DEPARTAMENT DE LLENGUATGES I SISTEMES INFORMÀTICS
UNIVERSITAT JAUME I

E80
PROYECTOS INFORMÁTICOS

INGENIERÍA INFORMÁTICA
Curso 2002 – 2003

Memoria Técnica del Proyecto

Desarrollo de un ISP

Proyecto presentado por el Alumno

Héctor Castillo Andreu

Dirigido por **Manuel Mollar Villanueva**

Castellón, a 18 de julio del 2.003

Resumen

Este proyecto va orientado al diseño e implementación de un ISP, habiéndose abordado concretamente:

- El diseño de los sistemas informáticos que necesitará el ISP (incluyendo la conectividad y redes informáticas necesarias).
- El diseño y el desarrollo del software de gestión necesario para un ISP, entendiendo que no hay producto similar que permita la configuración automática de los servicios y su venta a los clientes.

Se ha incidido especialmente en el análisis, la planificación y el diseño de todas las etapas.

Palabras clave

Internet, redes, ISP, TCP/IP, Linux.

hector@bith.net

Contenido

Índice de ilustraciones

1 Introducción

Aquellas compañías de tamaño mediano o pequeño que nacieron como ISPs en España al calor de *Infovía* (cuyo negocio básico era ofrecer conexión a Internet a través de ellos) se han concentrado hoy en ofrecer servicios como el correo, alojamiento y diseño Web, cuando no han optado directamente por convertirse en ASP (compañías que sólo ofrecen contenidos y no servicios en Internet). Este abandono del negocio del acceso ha sido debido al empuje de las grandes operadoras, hoy por hoy las únicas que continúan en el sector del acceso a Internet.

Este nuevo tipo de compañías de servicios como el correo y el alojamiento se conocen como proveedor de soluciones en Internet, cuyas siglas en inglés coinciden con su anterior actividad: ISP.

En Castellón se pueden encontrar algunos ejemplos de esta situación, entre ellos *Csnet S.L.* (**www.csnet.es**) y *Stalker Comunicaciones S.L.* (**www.stalker.es**), ambas aparecidas al calor del fuerte crecimiento del acceso a Internet producido en los años 1.995 y 1.996.

Es lícito afirmar que el negocio para el que fueron creadas estas compañías (ofrecer acceso a Internet) ya no es viable económicamente, pero pueden ocupar un nicho rentable si se concentran en ofrecer aquellos servicios que antes vendía como de valor añadido o complemento al acceso (el alojamiento y el correo serán dos buenos ejemplos) a precios competitivos a los que las grandes operadoras que ahora copan el acceso a Internet no van a entrar. A estas empresas, por no ofrecer acceso, se les conoce a veces como IPP (*Internet Presence Provider*), ya que ofrecen a terceros presencia física en Internet a través de *hosting*, *housing* y similares, aunque en este proyecto se le llamará indistintamente ISP o IPP.

El proyecto pretende demostrar, a partir de mi experiencia de dos años en la gestión técnica de los servicios de un ISP de pequeño tamaño, la viabilidad de este modelo de negocio en el que nos dedicamos a dar servicios complementarios al acceso y no acceso propiamente, y todo ello partiendo desde cero, con una compañía que crearemos a tal efecto y cuya viabilidad se tratará de garantizar en el proyecto, acometiéndose todos los pasos que llevarían hasta la puesta en servicio. Todo ello sin necesidad de ofrecer contenidos (como haría un ASP, cosa harto difícil con el hundimiento del mercado publicitario en la Red, que sería la única forma de obtener ingresos, dadas las resistencias del consumidor a pagar por los contenidos, que luego analizaremos).

2 Objetivos

El objetivo de este proyecto será acometer todo el proyecto de creación de un proveedor de servicios de Internet (ISP) de tamaño medio y gasto reducidos, con productos como el correo o sitios Web, y orientado a la venta directa de dichos productos a través de la Web.

Para esto será necesario:

- Identificar los servicios que deben ofrecerse y la forma en que estos serán ofrecidos al cliente.

  Establecer qué servicios serán fáciles de gestionar e integrar en un software que permita la configuración por parte del propio cliente.

- Determinar los recursos necesarios para el ISP.

  Establecer la cantidad, localización y fisonomía de los sistemas necesarios, así como analizar el tipo y tamaño de la conectividad a Internet necesaria. Establecer los costes de manera global.

- Desarrollo del software a medida para dicha gestión.

  Proyecto de ingeniería completo en el que se crearán las herramientas necesarias para que el ISP pueda dedicarse a la venta directa.

  Utilizar el modelado UML para este proyecto de ingeniería.

- Lograr el máximo posible de automatismo en la gestión de los servicios y usuarios, para minimizar el personal que intervendrá.

  Los servicios ofertados habrán de disponer de un panel de control que permita al cliente solicitarlos, que se le facturen y que se creen minimizando la intervención humana, todo ello con las garantías de seguridad adecuadas.

- Configuración de los sistemas.

  Establecer la configuración que deberán tener los sistemas, así como tratar la seguridad de los mismos.

3 Estructura de la memoria

La memoria está estructurada en diez capítulos, entre los que destacan cuatro por su extensión: uno de antecedentes y tres más que corresponden a las tres fases de un proyecto informático (análisis, diseño e implementación).

El capítulo 4 (de Antecedentes) está dedicado a analizar la evolución de Internet desde sus inicios, es decir, su historia y evolución desde su aparición en los 60.

El objetivo de esta documentada y extensa introducción es tratar de equilibrar los contenidos eminentemente prácticos que cualquier proyecto informático tiene, con una descripción más generalista del entorno (tanto económico como social) que rodea a Internet y que no podemos ignorar y necesitamos explicar para entender qué es hoy día un ISP. Asimismo, en el capítulo de antecedentes se encuentran apartados dedicados a analizar aspectos de Internet necesarios para un ISP: seguridad, derecho legal y fiscal, etc.

El capítulo 5 está dedicado al análisis del software y proyecto de ingeniería planteados. A este capítulo le sigue el capítulo 6, dedicado al diseño del proyecto.

La implementación realizada ha sido descrita en el capítulo 7, y contiene una somera descripción de las principales funcionalidades del software desarrollado, así como ejemplos de la configuración del sistema en lo relativo a asegurar la seguridad y funcionalidad del mismo.

A continuación, en el capítulo 8, se ha incluido un estudio de viabilidad económica del proyecto tal y como si se partiera de cero en la creación del ISP.

Finalmente están las conclusiones y posibles mejoras, expuestas en el capítulo 9, y la bibliografía, que se puede consultar en el capítulo 10. La memoria finaliza con un índice de términos en el capítulo 11, con referencias a las páginas en las que éstos se tratan.

4 Antecedentes

Internet ha sido y es el mayor de los grandes cambios que la Informática ha aportado a la vida moderna. Por Internet hemos de entender que hablamos de la red que interconecta ordenadores de todo el mundo usando un protocolo ya bastante maduro conocido como TCP/IP. Hemos conseguido reducir el coste de acceso a la información hasta el extremo que, gracias a ella, la noticia es difundida simultáneamente al propio hecho que la acontece al otro lado del planeta. Si comparamos este hecho con algunos hitos históricos, veremos que hace dos mil quinientos años hicieron falta bastantes horas para que un soldado recorriera a pie los 45 kilómetros necesarios para que los habitantes de Atenas pudieran conocer su victoria en la batalla de Maratón.

Internet no puede entenderse sino como el paso último de la revolución de las Telecomunicaciones, que permiten la transmisión de la información a distancia, y que se inició en 1791 con el telégrafo mecánico, si dejamos de lado el uso que desde hace milenios se había hecho de señales de humo y el sonido.

El volumen de información que en Internet está al alcance de cualquier persona es astronómico: más de 800 millones de páginas Web alojadas en 150 millones de hosts (cifras que además no dejan de crecer), miles de millones de correos electrónicos y mensajes en boletines de noticias...

La o el Internet (ni el género del término está claro en castellano debido a lo realmente novedoso que sigue siendo) como red de comunicaciones constituye un servicio hoy día esencial, protegido ya en España al mismo nivel que la luz, la televisión, el agua y el gas como servicio público (es decir, con obligación por parte del Estado de que se facilite a cada ciudadano el acceso a este servicio si lo solicita) y que requiere (al igual que el gas, la luz o el agua) de un ISP que lo suministre al usuario final.

Esas compañías dedicadas a ofrecer el acceso a Internet, son conocidas como ISPs, acrónimo en inglés de *Internet Service Provider*, y aunque en los inicios de Internet las hubieron por millares en todo el mundo, en la actualidad sólo sobreviven las filiales de las compañías telefónicas tradicionales, por la dificultad que otras compañías tienen para desplegar redes propias para llegar hasta el cliente. La mayoría se han reconvertido para ofrecer otros servicios, del tipo de los que aquí en este proyecto se explican.

4.1 Historia de Internet

Hablaremos primero de cómo se ha llegado a la situación actual, siguiendo la historia de Internet desde sus orígenes en los años 60 hasta la actualidad, y desde una perspectiva mundial hasta acabar hablando de la situación y evolución concretas en España. Analizaremos con mayor detenimiento la historia en los Estados Unidos, al ser este país el que dio origen a Internet.

4.1.1 Desde ARPAnet hasta hoy[1]

Fue en Estados Unidos justamente donde nació Internet, y no en su origen no se buscaba mejorar el comercio o las comunicaciones: se ha dicho que la guerra ha contribuido a desarrollar la inmensa mayoría de las invenciones que luego resultaron útiles para la Humanidad. Internet también cumple este dicho, ya que se creó para una guerra, la Guerra Fría.

En los años 50 el Departamento de Defensa de los EE.UU. quiso que en caso de guerra nuclear hubiera un sistema de comunicaciones tolerante a ataques, que sobreviviera y permitiera que los diferentes centros de control militar siguieran comunicados. Esto le llevó a buscar formas de mantener la comunicación incluso en caso de destrucción de una parte de la infraestructura.

La primera idea para lograr el objetivo que se buscaba llegó de la mano de Paul Baran en 1.962, cuando propuso una red de malla con ordenadores interconectados entre sí sin jerarquía ni centralización, dónde la desaparición de parte de los nodos (es decir, cada uno de los ordenadores que se pretenden comunicar) no incapacitaba a priori la comunicación, ya que cada nodo decidiría qué ruta seguirían los datos que a él llegaran en forma de paquetes o fragmentos de entre los enlaces disponibles en ese nodo, competiendo de manera exclusiva al destinatario del mensaje el recomponerlo a partir de los trozos recibidos.

Esta solución incorporaba otro concepto previo pero revolucionario: el fragmentar la información y transmitir estos fragmentos de manera independiente, idea que había nacido a principio de los años 60 en la corporación RAND (www.rand.org), en la que trabajaba Leonard Kleinrock. Leonard publicó en julio de 1.961 el primer trabajo donde se habla sobre comunicación basada en algo que sea daría luego en llamar *conmutación de paquetes*. El Pentágono, a través de su Agencia de Proyectos de Investigación Avanzada o ARPA, financió la puesta en marcha del modelo experimental y finalmente en 1.969 se creó la red,

---

[1] Cronología de Internet, de Robert H. Zakon:
<http://www.zakon.org/robert/internet/timeline>

bautizada como *ARPAnet*. ARPA hoy sólo investiga ya en áreas militares, de ahí que sea DARPA su nombre actual (www.darpa.gov)

ARPAnet nació con sólo cuatro nodos en centros de investigación, dado su carácter experimental. Tres de esos nodos estaban en el área de Los Ángeles, California, en una zona que aún hoy concentra la mayoría de las empresas de Nuevas Tecnologías de los EE.UU. y al famoso Silicon Valley. El primer nodo creado fue el del campus de Los Ángeles de la Universidad de California (UCLA, en sus siglas en inglés), el 2 de Septiembre de 1.969. Otro nodo fue creado dos mes más tarde en otro campus de dicha Universidad en Santa Bárbara (UCSB). El tercer nodo se montó algo más al norte, en el Instituto de Investigaciones de Stanford (SRI), y apareció en octubre del mismo año. Este último centro era parte de la Universidad de Stanford, pero a partir de 1.977 se ha convertido en un instituto sin ánimo de lucro (figura que no está presente en Europa), autónomo con respecto a la Universidad que lo creó y que continúa en la misma línea de investigación.

El nodo más alejado se encontraba dentro de la Universidad de Utah en Salt Lake City, y debido a las dificultades que experimentaron con ATT para el tendido del cableado, no fue operativo hasta el 1 de diciembre. En realidad si nos fijamos en el esquema de la red que se ve en la ilustración, veremos que no cumplía demasiado bien lo de ser redundante y útil en caso de ataque: Salt Lake City sólo quedaba conectada por una única conexión.

Ilustración 4-1: Localización de los nodos de ARPAnet

La heterogeneidad de los sistemas que integraban esta red también era muy alta. De hecho se conserva un esquema que lo muestra, esquema encontrado en los papeles de los ingenieros que en aquellos años colaboraron en su construcción, y que ha pasado a la historia como la imagen más conocida del ARPAnet original (a falta de fotografías de los equipos y el cableado usado). En sólo cuatro nodos no había un solo protocolo, equipamiento o sistema operativo igual: máquinas PDP, servidores IBM 360, un Sigma T...

La primera comunicación que se produjo en esta red data del 29 de Octubre, cuando Charley Kline intento acceder de manera remota desde UCLA al servidor que estaba en SRI: valga decir que no lo consiguió, ya que la red se cayó cuando estaba escribiendo la letra G de la palabra LOGIN.

Poco a poco fueron solucionándose estos problemas y apareciendo otros muchos nodos de instituciones académicas y militares, conforme se estabilizaba el diseño del protocolo, y para 1.972 había 40 nodos interconectados. Aunque eso sí: todos dentro de los EE.UU., ya que al fin y al cabo estamos ante un proyecto financiado con dinero público y

muy rentable por cierto, ya que le valdría posteriormente tanto el controlar algunas de las instituciones con poder técnico sobre la Red una vez esta se ha universalizado a nivel planetario, como disfrutar de un tremendo crecimiento económico en los 90, vinculado a la Nueva Economía en la se engloba a Internet, pero para llegar a esto aún queda mucha historia.

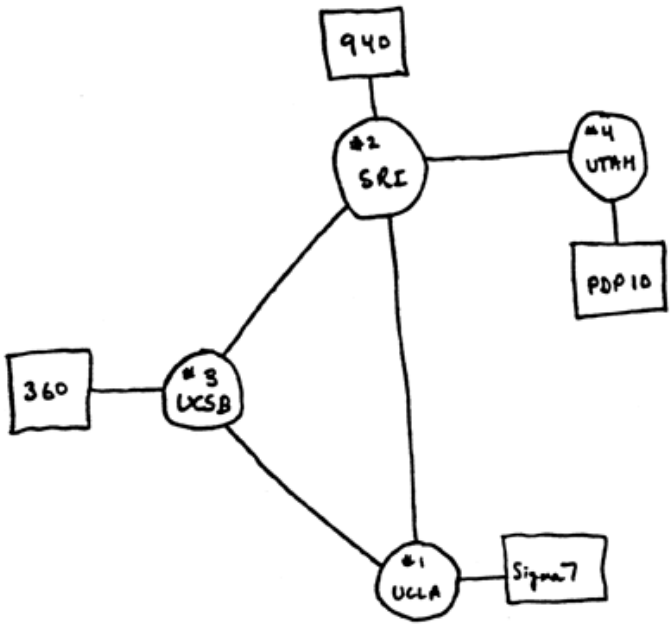


Ilustración 4-2: Sistemas que formaban la ARPAnet original

Esta primitiva red, en la que participaban distintos centros de investigación, empezó a servir muy pronto para algo realmente revolucionario: para intercambiar información a un coste muy bajo, para comunicarse. En 1.969 apareció en la Universidad de California en Los Ángeles el sistema de RFC (*Request for Commentaries*: petición de comentarios), que permitía a todos los participantes en un proyecto o discusión publicar las conclusiones asignándoles un número de índice. El sistema acabaría por convertirse en una forma de discutir los estándares que controlan la mayor parte de los aspectos de Internet, que además pueden ser posteriormente revisados y dejados obsoletos por nuevos documentos RFC: un RFC es publicado y aceptado tras constituirse un grupo de trabajo y un periodo de discusión sobre el mismo, y los hay de todo tipo de estándares o incluso sobre propuestas de diseño de estándares o redes que luego no llegan a aplicarse: el primer RFC, de Steve Crocker, detallaba los pormenores del protocolo de comunicación que en aquel remoto tiempo comunicaba los nodos de ARPAnet[2], y nunca llegó a usarse en su totalidad.

También la cultura es información que se puede transmitir por Internet, y pronto se vería esto en el nuevo medio: en 1.971 Michael

[2] Host Software, de *Steve Crocker* (abril de 1.969), RFC 1: <http://www.faqs.org/rfcs/rfc1.html>

Hart creaba el Proyecto Gutenberg, para crear y difundir textos electrónicos gratuitamente (el estándar ASCII databa de 1.968).

En 1.972 apareció el primer programa de correo electrónico, que pronto se convirtió en una de las aplicaciones más usada (y tres años después ya se discutía el problema de cómo bloquear el correo no autorizado, otro de los problemas más usuales en Internet). Fue Ray Tomlinson quien en 1.971 logró enviar el primer mensaje de correo electrónico, y el que estableció la letra arroba (@) usada en las direcciones de correo y que tanto se ha popularizado. Como curiosidad reseñar que este símbolo procede de la tradición sajona de usar esta abreviatura de la palabra latina AD, y que se empleaba en el comercio para separar el artículo y cantidad vendidos de su precio (por ejemplo, 3 sillas a 5$ la silla, sería 3 @ 5$), situación a la que llegó tras evolucionar desde la unidad de peso medieval original que era. Que se usara para estos menesteres aseguró su presencia en los teclados de las máquinas de escribir desde finales del siglo XIX, y de ahí llegó a los teclados de los ordenadores, pese a que en los años 70 del siglo XX casi nadie usaba este símbolo: la razón que llevó a Ray a usarla fue justamente su presencia en los teclados y el bajo uso que de esta tecla se hacía[3].

Mientras tanto, el primitivo proyecto ARPAnet se conectaba con otras redes que existían, apareciendo los primeros problemas de intercomunicación tanto por este hecho como porque con el crecimiento cada vez había más pilas del protocolo diseñadas por terceros y que no seguían al pie de la letra las especificaciones o que lo adaptaban a sus necesidades.

Conforme mas grande era la red más necesidad había un estándar que independientemente del sistema operativo empleado, de la topología de la red o de la conexión, pudiera entenderse con otro *host*, sentándose así las bases de la aparición de un protocolo más universal y que no lastrara el crecimiento de la red. Robert Kahn introdujo el término *arquitectura abierta* en 1.972 refiriéndose justamente a esta necesidad: en inglés se la llamó *Internetting*, porque servía para la relación entre redes. Y para relacionar esas redes nacería un protocolo básico en Internet: TCP/IP.

En 1.974 había aparecido un documento titulado *A Protocol for Packet Nertwork Internetworking* (y cuyos autores eran Vint Cerf, de la Universidad de Stanford, y el mismo Robert Kahn que acabamos de nombrar), documento en el que se ahondaba en esta necesidad de un protocolo común. Ese mismo año se crearía un protocolo que evolucionaría hasta su versión 4 en apenas ocho años, en 1.982. El protocolo era orientado a capas y validaba como estándar dos protocolos ya existentes, el *Transmisión Control Protocol* (TCP) y el *Internet Protocol* (IP), dando lugar a una pila de protocolos que se

---

[3] Origen y significado de la arroba, por Antonio Caravantes:
<http://www.caravantes.com/arti03/arroba.htm>

conocería por la unión de los nombres de los dos protocolos, TCP/IP. La versión del protocolo actualmente en uso es conocida como IPv4.

En 1.984 se forjaba *Well* (www.well.com), la primera comunidad comercial de usuarios. Ya en 1.975 apareció el primer ISP y la primera red comercial con usuarios de la banca financiera y del mercado de valores, TELENET, que años más tarde acabaría en manos de Verizon (www.verizon.com). La razón de este tímido despegue de usuarios no académicos o militares la hemos de buscar también en UUCP (que aparecería dos años más tarde, en 1.986, estandarizado en el RFC 976)[4], y las BBS, ordenadores a los que se conectaban directamente los usuarios para intercambiar información. Estamos aún en una etapa en la que los actuales usos de la Internet aún no eran los predominantes de la red (la excepción es el correo), y la información que los usuarios se encontraban en los BBS se parecía más a un panel de anuncios de la entrada de una facultad (cada usuario podía ver todo lo que los otros habían ido dejando y modificando).

El UUCP eran una serie de utilidades que permitían transferir datos entre ordenadores, fundamentalmente en Unix, de dónde le viene el nombre (UUCP significa *Unix to Unix Copy*), permitiendo también la ejecución y programación remota de aplicaciones. Fue el rey durante los años en los que (al igual que ocurría con el BBS) los usuarios conectaban entre ellos directamente de módem de usuario a módem de usuario, y no lo hacían a una red central como constituye Internet. Cuando la conexión del usuario se comenzó a realizar a Internet usando TCP/IP el UUCP dejó de tener sentido.

El principal impulso en la Red sigue siendo la ARPA, único organismo que hasta 1.979 financiaba Internet. Fue en aquel año cuando muchas universidades no vinculadas a proyectos de investigación militar se unieron con la Nacional Science Foundation (www.nsf.gov) para lanzar la *Computer Science Network* (CSnet), proyecto que se lanzaría en enero de 1.980 con un presupuesto inicial de cinco millones de dólares y el requisito de que fuera autosuficiente en cinco años. En su primer mes de funcionamiento la red conectó las tres universidades que comandaron el proyecto: Wisconsin-Madison, Purdue y Delaware.

En 1.983 el control estricto que ARPA ejerce sobre la red obliga a un divorcio entre la parte militar y las universidades, que se pasan a la red civil CSnet, aunque con una pérdida importante de infraestructura de acceso (que queda en manos de los militares). Fue también ese año de 1.983 cuando se creó el sistema de nombres de dominios, que prácticamente se ha mantenido hasta ahora.

---

[4] *UUCP Mail Interchange Format Standard* (febrero de 1.986), laboratorios Bell, RFC 976:

<www.rfc-editor.org/rfc/rfc976.txt>

La red pública CSnet tenía un problema de financiación que impedía equiparla en calidad a la parte militar, por lo que se ven obligados a facilitar las cosas con dos acciones importantes: la primera sería hacer de libre disposición el protocolo TCP/IP, incentivando además el desarrollo de las pilas de protocolo para las plataformas más habituales. La otra acción tomada fue permitir la conexión de otros países, siendo creado en febrero de 1.984 el primer nodo fuera de los EE.UU., en Israel, al que poco tiempo después comenzarían a seguirle Corea, Australia, Canadá, Francia, Alemania y Japón.

Tras la escisión, la parte militar se cierra al resto y pasa a llamarse MILnet. Sólo se dispuso una pasarela entre ambas redes para el intercambio de correo. En ese momento ya había más de 500 nodos interconectados a la red civil, que comienza a ser ya conocida como Internet.

MILnet desaparecería también posteriormente en 1.989, cuando las instituciones que en ella se encontraban (de la NASA al Departamento de Energía) prefirieron incorporar sus máquinas a Internet, que llevaba ya ese año camino de convertirse en red planetaria: el número de ordenadores en la red superaba los 100.000. Había habido también ya problemas con el ancho de banda disponible, escaso en ocasiones para satisfacer a tantos nodos, y ya en fechas tan tempranas como 1.987 el National Science Foundation se vio en la necesidad de crear una red de más capacidad, NSFnet, la cual conectaba siete de los nodos con más tráfico a través de enlaces de 1.5 Mbits, toda una revolución si tenemos en cuenta que muchos normalmente antes disponían de 56 Kbps. Posteriormente habría proyectos similares, como Internet2, hasta que a finales de los 90 se hizo innecesario esa subvención pública a la estructura de la Red, porque el sector privado ya dispone de infraestructura suficiente para soportar el tráfico.

Ese mismo año de 1.989 Tim Berners-Lee, investigador en el centro europeo CERN de Suiza, elaboró su propuesta de un sistema de información que se relacionaba con otros recursos: era el primer esbozo de la Web. De cualquier forma, el mérito de la idea de los hipervínculos no corresponde a Tim, sino a Ted Nelson, que en 1.974 publicó un trabajo en el que aparecían conceptos como hipertexto y links, y en el cual se basó Tim.

Pero Internet aún tuvo que dar un último salto con la revolución multimedia que se produjo a principios de los noventa en la informática: hasta que Tim Berners-Lee en 1.989 inventó el *World Wide Web* y desarrolló el protocolo pertinente para que dos equipos intercambiaran imágenes y texto vinculado (de ahí que *http* sea el acrónimo de *HyperText Transfer Protocol*: protocolo de transferencia de hipertexto), en Internet la gente intercambiaba mensajes, transfería ficheros y accedía a servidores remotos, pero no resultaba un medio atractivo ni interactivo.

La mayor interactividad que se podía lograr entonces estaba limitada a los servidores *gopher* (en los que nos encontrábamos con un índice o

menú que mejoraba algo el servicio de *ftp*) y en los BBS, por su estructura de tablón de anuncios, y con la Web lo que se lograría es una comunión entre imagen, sonido y texto interactivo que haría atractiva a cualquier persona la información.

Tim no desarrolló el primer cliente Web hasta 1.990 (aún mientras trabajaba en el CERN), y al que llamó justamente *WorldWideWeb*: Como inducía a confusión con el nombre del Web, lo renombró más tarde como *Nexus*, programado enteramente usando un equipo *NeXT*[5]. Durante el año 1.990 sigue el desarrollo, y no es hasta agosto de 1.991 cuando Tim publica en el grupo *alt.hypertext* la disponibilidad del programa: en principio se trata únicamente de la versión estable del navegador (el cliente Web) ya que el servidor Web por el momento es una única máquina a la que se conectan los miembros del CERN para descargar de ella los contenidos.

Pese a que el primer cliente Web fuera suyo, fue el equipo formado por Mac Andreessen y alumnos suyos del *National Center for Supercomputing Aplications* (www.ncsa.edu) el que creó el navegador más popular, *NCSA Mosaic*. Mac Andreessen fundaría luego junto a Jim Clark la compañía *Netscape* (www.netscape.com), cuyo negocio estuvo basado justamente en la venta del navegador por ellos desarrollado. El tal Jim era otro visionario, que para fundar *Netscape* dejó otra compañía informática rentable que él mismo había fundado, *Silicon Graphics* (www.sgi.com).

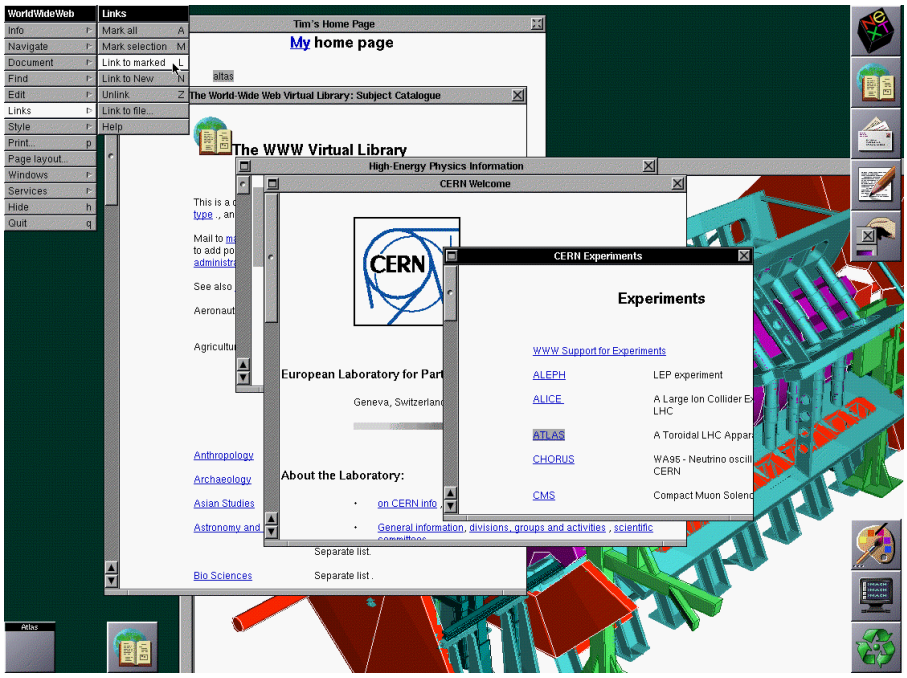


Ilustración 4-3: Aspecto del navegador original deTim en 1.990

La explosión fue terrible: en poco tiempo desterró a los servidores *gopher*, cuya idea era similar (vincular mediante enlaces) pero carecía de la capacidad multimedia de los gráficos y el sonido que la Web

[5] Biografía de *Tim Bernes-Lee*: <http://www.w3.org/People/Berners-Lee>

adquiriría[6]. *Gopher* era un protocolo también reciente (de 1.991 igualmente), en el que los usuarios conectaban a una dirección y se les mostraba un menú con enlaces, siempre en modo texto. Dado que suponía una mejora sustancial respecto del FTP, tuvo un rápido crecimiento y hasta finales de 1.993 siguió superando en éxito al Web. Pero cuando los ordenadores tuvieron la potencia suficiente para comenzar a servir gráficos, los usuarios prefirieron el Web.

De hecho podemos ver esta tendencia si analizamos el tráfico TCP segmentado por puertos que cruzaba el *backbone* de NFSnet, la entonces red troncal de Internet: a partir de mayo de 1.994, el porcentaje de tráfico dedicado al Web subiría de manera exponencial hasta suponer el protocolo dominante y hacer desaparecer a *Gopher*.

En 1.995 ya prácticamente todo el mundo era incapaz de entender Internet sin la *World Wide Web*, que es el nombre que se le quedó. Tim en 1.994 fue uno de los fundadores del W3C, el *World Wide Web Consortium*, un organismo que trataría de liderar el desarrollo de la Web buscando mantener un control sobre el protocolo y sobre el desarrollo de nuevos estándares relacionados, objetivo que pese a la guerra que años más tarde establecerían dos compañías desarrolladoras de navegadores (*Microsoft* con su *Internet Explorer* y *Netscape* con su *Navigator*) más a menos se ha logrado.

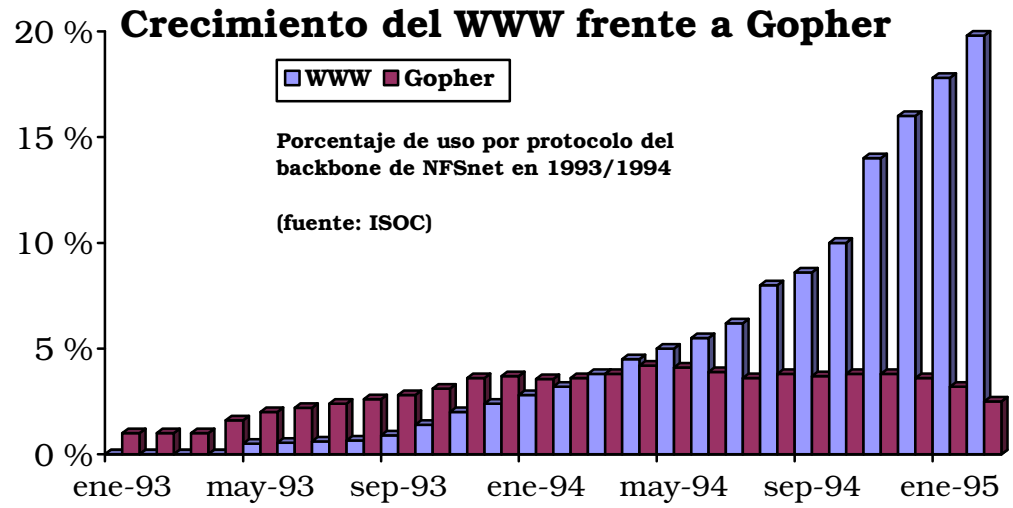


Ilustración 4-4: El crecimiento del Web frente a Gopher

En 1.992 –con más de un millón de servidores en la red– se creó la *Internet Society*, una especie de *autoridad* de Internet. Nacía como el lugar donde pactar los protocolos que harían posible la comunicación, aunque en esto siempre ha habido más árbitros que juego, y no siempre ha liderado los nuevos estándares (IEEE, W3C... son muchos los organismos que han creado estándares que han afectado en mayor o menor medida a Internet). Realmente el problema básico para estos organismos es que un medio como Internet carece de mando: los

[6] Bellver, Carles y Jordi Adell (1.995): *La evolución de Internet y el World Wide Web*. Revista Net Conexión 1. Págs. 88-92.

protocolos se convierten aquí realmente en protocolos cuando son adoptados por la mayoría de usuarios, y son estos los que realmente deciden qué tecnologías se aplican en Internet.

Con la extensión de los ordenadores personales y el lanzamiento del primer navegador comercial, *Mosaic*, en 1.993, ya había llegado la madurez del invento. En 1.994 se abre el primer banco que opera en Internet, el *First Bank* (www.firstbank.com), y comienzan a aparecer proyectos comerciales en la Web, abriendo un lustro de gasto desenfrenado por parte de grandes compañías que luchan por ser las primeras o las mejores en Internet en ofrecer cierto servicio.

En 1.997 ya hay 17 millones de servidores en la red. A partir de aquí las cifras se disparan debido a la popularización de Internet, aunque el crecimiento no se ha ido produciendo a igual ritmo en Europa (más lento) que en EE.UU., ni ha sido grande en aquellas capas sociales que no pueden costearse la conexión o en el llamado Tercer Mundo, muy por detrás de Europa.

El que Europa esté por detrás de las cifras estadounidenses de uso de la Red tiene algo de sentido si consideramos que han sido los EE.UU. los que han liderado la investigación y la adopción de las Nuevas Tecnologías en la sociedad. Una lástima sobretodo viendo que iniciativas en Internet las hubo, y algunas muy tempranas: el primer prototipo de red de conmutación de paquetes del mundo lo construyó en 1.968 el Laboratorio Nacional de Física del Reino Unido, pero el proyecto no fue luego continuado. Y aunque no hay que olvidar que la idea de la Web surgió en el CERN suizo, a finales de 1.994 el propio CERN decidió interrumpir su trabajo de desarrollo de la WWW por considerarlo fuera de sus objetivos como organización de investigación en Física Nuclear. En resumidas cuentas: los gobiernos europeos no supieron nunca darle el impulso necesario o ver las posibilidades tecnológicas que liderar este medio han supuesto luego a EE.UU.

Un factor (como posteriormente veremos en profundidad) que ha impedido un crecimiento de usuarios tan grande a partir del despegue definitivo de la red como el de Norteamérica ha sido el precio de las comunicaciones (en Estados Unidos y Canadá normalmente las llamadas locales, usadas por los particulares para conectarse a un ISP, son gratuitas) y la mala calidad de estas (en especial las conexiones a la red troncal: en EE.UU. la existencia de bastantes operadores de larga distancia dispuestos a invertir en grandes redes contrastaba con un panorama europeo en el que en cada país se desarrollaba una única red monopolística y que se interconectaba con otros países sólo por iniciativa política y no comercial), pero sobretodo por los monopolios.

Ilustración 4-5: Aspecto de la página de acceso de *Minitel*

Otras iniciativas de gran interés realizadas en Europa con intención de potenciar el desarrollo tecnológico se perdieron al intentar marcar diferencias con EEUU y no utilizar los mismos protocolos, optando por redes cerradas. Un ejemplo lo tenemos en *Minitel* (www.minitel.com), implantado en Francia en 1.981 por *France Telecom*. Consistía en un terminal que se conectaba a la línea telefónica con un protocolo propio, que creció hasta tener más de 15 millones de usuarios, pero al trabajar sin gráficos (de hecho trabaja de un modo similar al teletexto), y además no requerir de más aparatos que un televisor y un terminal que *France Telecom* distribuía sin opción de compra, ha supuesto paradójicamente un lastre tanto en el porcentaje de hogares con PC como en el desarrollo de la Internet en ese país.

Otro ejemplo similar lo encontramos en el propio modelo OSI, creado por el ISO con la idea de reemplazar al TCP/IP, objetivo que no se ha logrado. La razón es evidente: el protocolo TCP/IP se usaba ya ampliamente cuando se diseñó OSI, y cambiar a otra pila de protocolos suponía una inversión que pocos veían lógica. Había también otras razones que hicieron que OSI haya acabado siendo el modelo teórico de cualquier red (no solo TCP/IP), como sería su complejo diseño y redundancias de algunas funciones y capas, pero el principal problema para Europa fue que en su diseño los países europeos invirtieron mucho esfuerzo y recursos (siempre se vio TCP/IP como el protocolo americano de *ARPAnet*, y OSI parecía algo más europeo), con lo que en su fracaso por reemplazar a TCP/IP se consumieron recursos para inventar algo que ya existía y funcionaba, como era una pila de protocolos.

Y llegando a la actualidad, la última iniciativa relevante que nombraremos es *Internet2*, que propone crear un espacio aparte y de más calidad de comunicaciones para instituciones de investigación mediante conexiones de alta velocidad. Este proyecto se viene asociando en su implantación a IPv6, ya que ambos requieren cambios tecnológicos y en la pila de protocolos que no van a hacer fácil su generalización, tema que luego discutiremos con mayor profundidad.

Para finalizar este breve repaso a algunos de los nombres que en Internet han de ser recordados, evitaremos olvidarnos de *BBN Technologies* (**www.bbn.com**), hoy parte de *Verizon*, pero que en tiempos era un importante socio tecnológico del Departamento de Defensa de los EE.UU., al que se encargó el suministro de equipos que facilitaran el acceso a *ARPAnet*. Estamos hablando de los precursores de los routers: en 1.977 esta compañía sería la primera en diseñar un *gateway* (como entonces se les conocía normalmente al router, de ahí el equívoco que aún hoy existe entre *gateway* y router) que usaba la pila de protocolos TCP/IP. Poco después ARPA solicitó mejoras en la eficiencia de estos aparatos vitales para que la red funcionara, y fueron dos estudiantes de Stanford que colaboraban en este proyecto los que fundaron en 1.984 *Cisco Systems* (**www.cisco.com**), compañía que dedicándose al hardware que da soporte a Internet, concentra la mayor parte de este mercado y es la 95 compañía más importante de los EE.UU.[7].

[7] Ranking *Fortune 500* de las quinientas mayores compañías de los EE.UU., de la revista *Fortune*:

<http://www.fortune.com/fortune/fortune500>

4.1.2 Historia de Internet en España

En España Internet fue hasta hace muy poco cosa de Universidades, fundamentalmente por estar ligadas al proyecto *RedIRIS* (www.rediris.es)[8], de financiación estatal y entre cuyos fines (siempre con más propósitos que presupuesto) estaba el lograr el desarrollo tecnológico español.

En la prehistoria de la red en España está la creación, el año 1.988, por el Plan Nacional de Investigación y Desarrollo, de un programa para la *Interconexión de los Recursos InformáticoS* de los centros de investigación (de ahí lo de IRIS). Al principio lo gestionó *Fundesco* (Fundación para el Desarrollo de las Comunicaciones de Telefónica, hoy desparecida), pero en 1.989 pasó a manos de las propias Universidades y centros de investigación. IRIS fue el motor de Internet en España, pero como tal no constituía en su momento una gran red, con muchas conexiones RDSI de 64 Kbps. en los mejores casos.

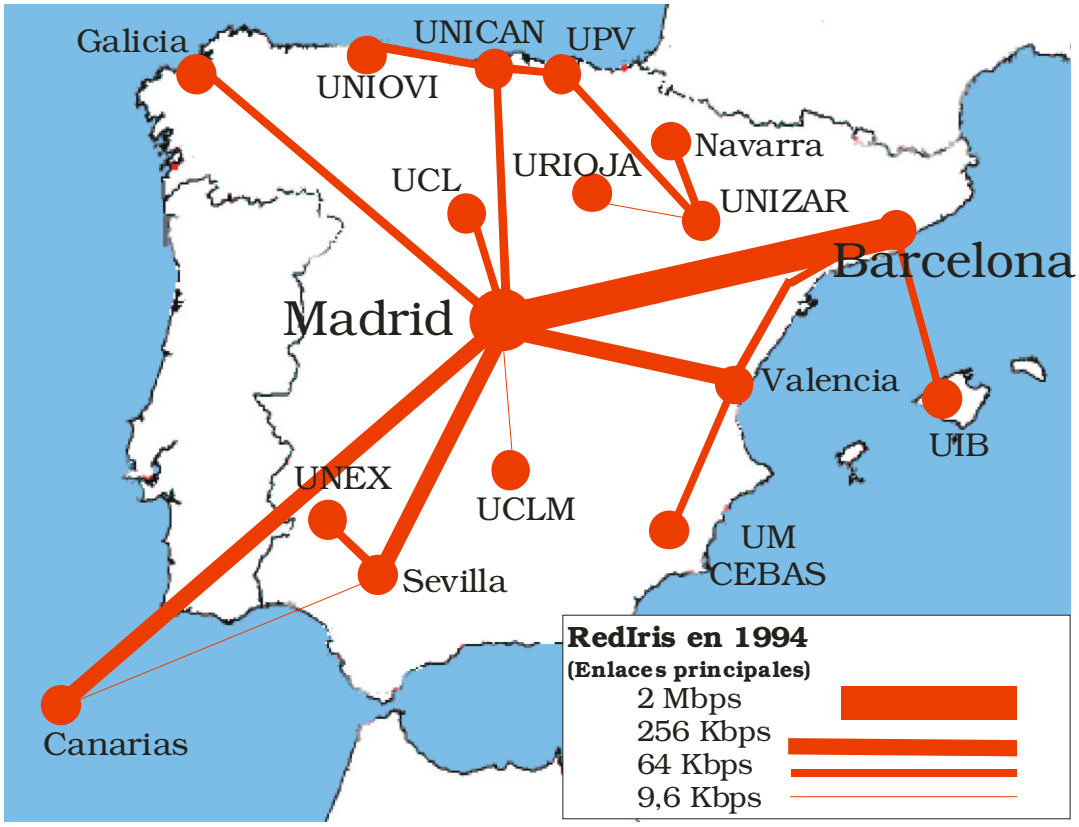


Ilustración 4-6: *Backbone* de RedIris en 1.994

[8] Acerca de *RedIris*: <http://www.rediris.es/rediris>

Martínez, Ignacio (1.992). *RedIRIS en la Internet: Una panorámica general de la Internet*. "Boletín de Red IRIS núm. 20-21.

Barberá, José (1.998). *Veinticinco años de Internet: una retrospectiva autobiográfica*. Boletín de *RedIRIS*, Págs. 23-34.

Pero IRIS como proyecto adolecía de una generalidad en sus objetivos que dio tal la sensación de carecer de rumbo, de tal forma que en el año 1.991 el Gobierno decidió darle un nuevo impulso y al calor de iniciativas similares en el resto de Europa invirtió en una red troncal de mayor tamaño que quedó bautizada como *RedIRIS*.

Desde enero de 1.994 está gestionada por el Consejo Superior de Investigaciones Científicas[9]. La asignación de los nombres dentro del TLD asignado a España, el ".es" era también exclusividad suya, aunque esta actividad ha sido en el 2.002 transferida por el Ministerio de Ciencia y Tecnología a la entidad pública empresarial *Red.es* (www.Red.es), quedando pues en la actualidad *RedIRIS* con la gestión de la red que intercomunica Universidades y centros de investigación. *RedIris* es actualmente miembro de Dante (www.dante.net), iniciativa europea que nace del deseo de crear una red troncal de investigación, y que en la actualidad es responsable de la implantación de *GÉANT* (antes conocida como *TEN-155*), formada por conexiones *Gigabit* entre 20 países europeos.

La etapa 1.993-1.995: primeros pasos

Desde la aparición de la red académica española en 1.989 la mayoría de centros universitarios y de investigación conectados incrementaban anualmente sus necesidades de ancho de banda a la par que Internet comenzaba a extenderse, necesidades que no siempre eran financiadas y entendidas desde el Ministerio correspondiente.

A estas trabas presupuestarias se unía también en España (y esto afectaba por igual a empresas que a organismos públicos) la existencia de un monopolio en las Telecomunicaciones que hacía que hasta las primeras modificaciones legales (introducidas a partir de 1.992) *Telefónica de España* fuera la empresa a la que iban a morir todos los proyectos que se pretendieran montar y que requerían Internet.

Un ejemplo lo tenemos en *Unión Fenosa*, una de las grandes eléctricas de este país. Por las mismas redes de alta tensión, ya mucho antes de las iniciativas para el desarrollo del PLC (Internet a través de la red eléctrica) se podían hacer circular datos a baja velocidad y voz, y la compañía *Unión Fenosa* así lo hacía para usos internos de la compañía hasta que en 1.992 fue demandada por *Telefónica*, prefiriendo contratar con *Telefónica* (y dejar de usar las suyas) que arriesgarse a perder debido a la legislación existente (que otorgaba a Telefónica la exclusividad sobre las comunicaciones). Una situación parecida se vivía en RENFE, el cual estaba posicionándose con una red de fibra óptica que discurría paralela al ferrocarril, y que no pudo alquilar ni utilizar hasta comenzar la liberalización de las telecomunicaciones.

---

[9] Sanz, Miguel Ángel (1.998). *Fundamentos históricos de la Internet en Europa y en España*. Boletín de Red IRIS núm. 45, Págs. 22-36.

En realidad la raíz de este problema nacía en 1.924, cuando fue creada la Compañía Telefónica Nacional de España (CTNE) por el Gobierno de Primo de Rivera, a partir de la fusión forzosa de todas las compañías previas y con un monopolio exclusivo sobre telefonía que nunca fue modificado. En EE.UU. el gobierno siempre dispuso de un órgano de control (la FCC) sobre la compañía mayoritaria y el mercado, pero en España, al igual que en Sudamérica y muchos otros países europeos, el Gobierno confiaba totalmente en la compañía (a cuya directiva además nombraba) y dejaba en sus manos cuantos reglamentos y decisiones técnicas creyesen necesarias.

Después de la Guerra Civil[10], el Decreto 31-10-1.946 establecía una separación de poderes que se mantuvo hasta los 90: Telefónica obtenía la exclusividad del servicio de retransmisión de datos y de transmisión de información (incluyendo la teleinformática, videotexto, teletexto, señales de telealarma y control, telemedida y telemando, según aclaró luego el Decreto de 26-10-1.978), mientras la Dirección General de Correos y Telecomunicación (Correos y Telégrafos) obtenía la exclusividad del servicio público de mensajes telegráficos y de *telex*. El tercer actor en esta función era la Televisión pública (RTVE), cuya red de radiodifusión era también monopolística hasta que la aparición de las televisiones autonómicas hizo necesaria la separación entre difusión y producción de señal audiovisual, y en 1.989 se creó *Retevisión*, compañía que sería privatizada años más tarde y se convertiría en la alternativa más firme a Telefónica.

Esta situación de monopolio o separación de poderes en cuanto a la comunicación perjudicó sobremanera el desarrollo de la Internet, no sólo por la falta de oferta alternativa, sino porque la propia compañía trababa cualquier iniciativa. Como ejemplo la creación del ISP *Arrakis*, el cual solicitó al monopolio presupuesto para un acceso de 2 *Mbps* mediante *frame-relay* en 1.993, que técnicamente nadie en Telefónica sabía cómo montar (los dos megas superaban con creces la mejor instalación del momento, que era en España de 512 *Kbps*). Hasta los comerciales dudaron tanto de su viabilidad que presentaron a los directivos de *Arrakis* la misma oferta que habían hecho por menor ancho de banda a IBM, sin preocuparse siquiera de reemplazar el nombre del cliente original[11].

Tampoco lo ha puesto *Telefónica* fácil en términos de coste de acceso al no haber nunca distinguido las llamadas de datos de las llamadas de voz. Durante años las llamadas de larga distancia han sido tan caras que durante la época en que existían en España algunos BBS (ordenadores con módem de hasta 9,6 *Kbps* a los que se conectaba

---

[10] Todas estas leyes y directrices aparecen mejor comentadas por Diego López Garrido en *La crisis de las telecomunicaciones*. Editorial Fundesco (1.989).

Castells, M. y otros (1.986). *Nuevas Tecnologías, Economía y Sociedad en España*. Alianza Editorial.

[11] Almirón, Nuria (2.001). *Cibermillonarios, la burbuja de Internet en España*. Editorial Planeta. Pág. 221.

directamente la gente) el coste de acceder a los mismos era astronómico (al tratarse de llamadas no locales). Incluso cuando comenzaron a operar los primeros operadores, *Cinet* (un operador catalán auspiciado por la *Fundació Catalana per la Recerca*, que también promovería posteriormente el portal *Olé*) le era imposible cuadrar las cuentas por ofrecer correo electrónico: el proveedor tenía un canal RDSI de 64 *Kbps* que le conectaba con Washington DC, por el que pagaba el coste de una llamada internacional. Obviamente el servidor de correo no estaba permanentemente conectado a Internet, sino que conectaba durante algunos preciosos minutos varias veces al día para intercambiar todo el correo acumulado en la cola.

La aparición de *Infovía* en 1.995

En esa situación apareció en 1.995 *Infovía*, una vez más forzándose su creación desde el Gobierno y al calor de los movimientos similares que se venían produciendo a nivel internacional. La situación era la siguiente: había en España muy pocos ISP que se arriesgaran a lanzar servicios dado que toda la infraestructura que llegaba a los clientes era de Telefónica, y era a esta a quién había que pagar peaje para llegar a los clientes, incluso para obtener caudal desde EEUU o Europa, porque no había otros operadores. En EEUU, por el contrario, existían operadores locales que ejercían el monopolio en su área, pero siempre había alternativas en las llamadas de larga distancia debido a la fuerte competencia que había en la larga distancia, o incluso varios operadores locales en la misma área metropolitana si estábamos en una ciudad importante.

En España incluso en el caso de que algún ISP se decidiera a lanzar algún servicio de acceso, el cliente final difícilmente se iba a lanzar a contratarlo si aparte del coste de acceso debía pagar el coste de las llamadas telefónicas: si el ISP se decidía a distribuir nodos por las capitales más importantes para ofrecer allí el acceso mediante llamada local, se iba a encontrar con que a su vez debería contratar con Telefónica la interconexión de todos sus nodos, e incrementar aún más sus costes.

Es con esta situación con la que el Gobierno se encuentra en 1.995 cuando se inicia la liberalización de las telecomunicaciones, y para permitir un despegue rápido de Internet el Gobierno opta por obligar al monopolio a crear *Infovía*.

Infovía consistía y consiste (porque aún está en funcionamiento) en facilitar a cualquier operador con licencia de tipo C1 (aquella que facultará en España a una empresa a vender acceso a Internet) puntos de acceso a precio de llamada local en toda España. Dichos puntos de acceso lo son en cantidad suficiente para cubrir la totalidad de la Red

Telefónica Básica (de ésta quedan fuera aquellos que usan telefonía no convencional en áreas rurales: la conocida como telefonía TRAC). La autenticación de los usuarios se hacía mediante un software de *Radius* que *Telefónica* distribuía a aquellos operadores que utilizan *Infovía*, lo cual no dejó de ser un avance, como luego veremos.

Con este esquema todos ganaban: los ISP se ahorraban desplegar una red de puntos de acceso, mientras *Telefónica* se aseguraba las llamadas de teléfono y algo más... Y es que en la mayoría de los ISP *Telefónica* hacía de *carrier* del ISP, por lo que el tráfico de los clientes del *Radius* entraba por la misma conexión por la que instantes después el tráfico volvía hacia Internet: el ISP era un salto más, en el que a ambos lados estaba *Telefónica*. Aunque el problema principal era otro: el tamaño. Todos los ISP aparecidos al calor de *Infovía* (cientos) carecían de tamaño suficiente como para sobrevivir si un gran operador lanzaba ofertas atractivas a sus clientes, dado su escasa capacidad de maniobra.

Por lo que tan rápido como aparecieron han desaparecido los ISP que no disponían de redes propias y que nacieron únicamente para trabajar a través de *Infovía*, debido a que cuando a Telefónica comenzaron a aparecerle competidores más serios (*Wanadoo*, *Eresmas*, BT...) que desplegaron red propia, y la operadora histórica contraatacó, la guerra de precios desplazó del mercado a aquellos que ofrecían lo mismo (acceso a Internet a través de nodos de acceso de Telefónica), más caro (porque tenían que pagar a Telefónica por partida doble: por el *Radius* y por el acceso) y de peor calidad (ya que el tráfico que estos pequeños ISP generaban o estaba por encima de su capacidad real, o en el mejor de los casos realizaba demasiado camino antes de alcanzar la Internet).

De hecho con la aparición de la banda ancha estos pequeños actores del mercado del acceso a Internet acabarán por desaparecer del todo, ya que la banda ancha necesita de una infraestructura mucho más costosa de lo que lo fueron en su momento los módems RTB, amén de que la concentración (y por tanto las economías de escala que dará un tamaño muy grande) son las únicas formas de que la actual competitividad permita sobrevivir en este mundo del acceso a Internet.

Pero no hemos comentado aún una de las ventajas que supuso *Infovía*, y que es la universalización de la autenticación PAP a través del *Radius* impuesto por Telefónica, desterrando el acceso UUCP[12], más común hasta 1995 y bastante costoso de mantener. Cualquier conexión de módem utiliza TCP/IP encapsulado sobre PPP de manera habitual, pero hay bastantes formas de autenticar y negociar el acceso a la Red.

Hay que tener en cuenta que estamos hablando del software que permite identificar y asignar recursos a los usuarios cuando conectan a

---

[12] Boletín número 50-51 (2.000). Ponencia de L. Álvarez, A. Ocón, E. Rubio y M. Galán: *Experiencias en la organización de un centro proveedor de servicios Internet*:

<http://www.rediris.es/rediris/boletin/50-51/ponencia8.html>

través de cualquiera de los puntos de acceso existentes en el país y que no eran pocos. La complejidad era por tanto bastante alta, ya que se trataba bien de tener una base de datos distribuida en cada servidor de acceso, o de poseer un ordenador centralizado para la autenticación. Desde luego cualquiera de las opciones era incompatible con una autenticación basada en usuario de *Unix* propia de los tiempos del UUCP. Y aunque la solución adoptada por Telefónica no fue mala, fue acompañada de dos exigencias a los ISP que trabajaran con *Infovía* que hicieron muy impopular a *Infovía*: el ISP no podía modificar el software del *Radius* ni obtener sus fuentes (muchos ISP tuvieron luego que apañárselas para parchear por su cuenta un software de *Radius* que tenía *bugs* y que Telefónica no mantenía), y lo que fue peor, no se ofrecían garantías de calidad de servicio, ni se ofreció en realidad buen servicio.

Esto condujo a un periodo comprendido entre 1.996 y 1.998 con una satisfacción de clientes e ISP tan baja que aún subsistieron gran cantidad de conexiones de Nodo Local con autenticación basado en usuario de sistema, conexiones que se diferenciaban de las conectadas mediante *Radius* que estaban controladas por el ISP en la medida que lo hacían mediante un software peor, pero conocido por él. Esta situación se acabaría a partir del 1 de diciembre de 1.999 con *Infovía Plus*, momento en que la operadora adquiere cada vez más el control de *Infovía* al desaparecer muchos ISP, y va convirtiendo cada vez más a *Infovía* en su red de acceso, distinta de las de otros operadores.

4.1.3 La burbuja especulativa de Internet

En el siglo XIX la aparición del ferrocarril supuso tal revolución en el transporte y comercio mundial que trajo consigo cambios en el modo de vida de todo el planeta, y generó tantas expectativas sobre sus posibilidades que una vez incumplidas estas, las compañías del ferrocarril acabaron bajo intervención estatal tras haber invertido cifras astronómicas en crear una red de ferrocarril con un alto valor estratégico y social, pero imposibles de rentabilizar en los plazos esperados.

Con Internet y la llamada Nueva Economía ha ocurrido algo similar a finales del siglo XX: en su momento álgido (antes del 2.000) nadie pensaba en conceptos financieros como el retorno de la inversión, sino únicamente en ser los primeros en conquistar este Nuevo Oeste que no existía una década antes (el Web como tal no apareció hasta 1.989).

Se llegó a hablar de revolución tecnológica, y profetas como *Nicholas Negroponte* (**http://web.media.mit.edu/~nicholas**) y la mayoría de analistas de bolsa contribuyeron a crear una burbuja especulativa que llevó a que cualquier empresa que tuviera relación con Internet se disparara en bolsa de manera ilógica, valorándose como positivo en la mayoría de ellas que no existieran beneficios. Como muestra tenemos la acción de *Yahoo!* (**www.yahoo.com**) que el enero del 2.000 alcanzó su valor máximo de 250 dólares y hoy día apenas cotiza a veinte dólares[13].

A esta burbuja financiera le ha seguido un batacazo aún mayor que anteriores situaciones similares en Bolsa debido a factores como la llamada democratización de la Bolsa (la masiva afluencia del ahorro familiar inexperto en sus tratos en la Bolsa), que contribuyó tanto al auge como al actual derrumbamiento por su falta de conocimiento y la irracionalidad de sus operaciones de compraventa.

Un ejemplo español: *Terra*

Pep Vallès y *Juan Villalonga* estaban en la Gran Vía madrileña el 17 de noviembre de 1.999 reunidos con la prensa ante la Bolsa de Madrid para culminar una de las mayores cortinas de humo financieras jamás lanzadas en España: *Terra*, que en sus mejores momentos en Bolsa tuvo mayor valor que su matriz, Telefónica de España.

Adquiriendo a golpe de talonario Telefónica logró crearse un nombre propio en Internet que no tenía, logrando su momento álgido en el 2.000 adquirir uno de los grandes nombres de Internet, *Lycos*.

Dada su importancia en España el nacimiento y ocaso de este (en teoría) ISP reconvertido en referente en Internet, merece ser comentado desde sus orígenes: *Terra* comenzó en *Cinet*, el ISP que la *Fundació*

---

[13] Cotización de *Yahoo* en la Web financiera *Yahoo! Finanzas*:
<http://es.finance.yahoo.com/q?s=YHOO&d=c&k=c1&a=v&p=s&t=5y&l=on&z=m&q=l>

Catalana de Recerca (organismo investigador público de la *Generalitat* catalana, **www.fcr.es**) lanzó en 1.994 y al que pertenecía Pep Vallès.

Este hombre fue uno de los principales impulsores dentro de la FCR del desarrollo de un buscador en catalán, con financiación pública. Con el proyecto acabado y el buscador ya bautizado como *Olé*, Pep Vallès y sus colaboradores fueron adquiriendo una presencia y nombre en Internet que atrajo el interés de Telefónica cuando esta descubrió que había perdido el tren de Internet.

Telefónica en aquellos momentos disponía únicamente de *Teleline*, su marca para Internet y su ISP, cuando decidió inyectar dinero y publicidad en lanzar lo que después se conocería como *Terra*, y dada la carencia de contenidos para ofrecer que el grupo padecía en esos momentos, para crear la marca tuvo que comprar las visitas allá donde más hubiera, siendo los buscadores un buen partido y *Olé* uno de los escogidos.

La vorágine hizo que desde el lanzamiento al parqué madrileño de aquel noviembre de 1.999 hasta el 2.000 *Terra* fuera adquiriendo todo aquel portal que le pudiera hacer sombra, descuidando cada vez más el negocio real que podía traerle ingresos y concentrándose en contenidos y publicidad, justo los dos capítulos que finalmente han resultado ser menos rentables. En el camino han ido siendo absorbidos portales como *Wired* (**www.wired.com**), *Angelfire* (**www.angelfire.com**), y sobretodo la última gran operación, la fusión con *Lycos* (el cual a su vez ya había adquirido otros portales de la talla de *Tripod*, **www.tripod.com**)[14].

La adquisición/fusión con *Lycos* le costó en mayo del 2.000 al grupo *Telefónica* la friolera de 12.500 millones de dólares (más de dos billones de pesetas de la época). Estas cifras iban lastrando un proyecto ya en números rojos por el ya desorbitado coste de *Olé* (18 millones de euros) y la creación de las filiales sudamericanas y estadounidense (*Terra* tiene entre sus objetivos en convertirse en la referencia en el mundo de habla hispana). Como muestra de esas cifras: 250 millones de euros para *Terra Networks Mexico* (**www.terra.com.mx**), 180 millones para *Terra Networks Brasil* (**www.terra.com.br**)... Se supone que en esta política de gasto sin control Telefónica gastó alrededor de 30.000 millones de euros.

La política llevada a cabo hasta el cambio del equipo gestor en noviembre del 2.000, aunque tiene referentes aún peores que han acabado incluso en bancarrota (*Worldcom*, sin ir más lejos), nos servirá aquí de muestra de la locura que imperó desde finales de los 90 y hasta marzo del 2.000 entre las compañías de Internet. Los nuevos gestores de Telefónica se encontraron con una compañía que no casaba en absoluto con sus objetivos (es incluso evidente que se perjudican entre ella y la propia Telefónica al ofrecer ambas los mismos productos de acceso a Internet) y comenzaron a surgir enfrentamientos con el equipo

[14] Portales propiedad de *Terra Networks*:
<http://www.terralycos.com/esp/about/au_1_3.asp>

gestor de la parte americana de la fusión, *Lycos*, y en especial con su presidente ejecutivo, *Robert J. Davis*, hoy fuera de la misma. El valor bursátil de *Terra* es casi despreciable, el valor de la marca e imagen creadas en estos años –aunque considerable– no justifica el gasto realizado, y sigue sin estar clara la política que tomará la compañía matriz *Telefónica* con respecto a esta filial, aunque de momento parece haberla abandonado a su suerte, diferenciándose claramente sus productos ADSL de los de Telefónica y dejando *Telefónica* de vender productos de su filial, siendo actualmente *Terra* un portal más del panorama español.

Los ASP y el mercado de la publicidad en la red

Aunque fue uno de los mejores soportes económicos para los portales de Internet hasta ahora hemos evitado hablar de esta área de negocio, descartándola indirectamente en nuestro proyecto al no nombrarla. Veremos ahora el por qué.

Los portales son las páginas que más visitas concentraban y concentran en la Red, y cuyo atractivo reside en acumular información atractiva al usuario, bien sean noticias de actualidad, descargas de programas, *chat* en línea, y sobretodo, los buscadores. Estas empresas podrían si sólo ofrecen contenidos ser denominadas ASP, nombre con el que se conocen a los proveedores de contenidos en Internet.

Aprovechándose de esta circunstancia fueron las primeras en obtener unos ingresos que no obligaban al pago de los servicios por parte del usuario vía la publicidad: *banners* de publicidad que se insertaban en las cabeceras del sitio Web, ventanas que se desplegaban por encima de la que el usuario solicitaba (anuncio conocido como *pop-up*), texto añadido a los *emails*, resaltarse en las búsquedas… las posibilidades de la Web son infinitas, y además permiten un seguimiento de la audiencia que ve la publicidad mayor que en los medios tradicionales, mediante el uso de *cookies*. De hecho existen compañías especializadas en publicidad como *DoubleClick* (**www.doubleclick.com**) o *RealMedia* (**www.realmedia.com**), cuya principal misión es gestionar los espacios dedicados a *banners* en los portales y venderlos a anunciantes.

Las inversiones en publicidad se dispararon hasta cifras astronómicas y muchas compañías basaron su negocio en esta área de manera exclusiva, coincidiendo su momento álgido con los meses previos a la caída de la burbuja especulativa. Sin ir más lejos valga de muestra el ejemplo de *Terra*, que durante la fusión alcanzó un acuerdo muy ventajoso con el grupo editorial y de medios alemán *Bertelsmann* que comprometía a la segunda a una inversión de mil millones de dólares en cinco años en publicidad en los portales de la segunda.

Los problemas llegaron tras la burbuja con una drástica caída de la inversión en publicidad: el castigo sufrido por la publicidad online fue excesivo: si de media antes de la caída de la inversión se pagaban 3 centavos de dólar por *banner* leído en la actualidad se pagan alrededor

de la décima parte de esa cantidad[15]. Esta reducción era en parte achacable a la endogamia presente en la publicidad de la red: la mayoría de los sitios Web ofrecían *banners* de otros portales o empresas de la red, con lo que la quiebra de una a su vez reducía los ingresos de la otra. Además se producía en un contexto de recesión económica en el que los gastos que primero recortan las empresas son justamente los de publicidad, sufriéndose en el año 2.001 una caída media del 15% de la publicidad incluso en los medios tradicionales.

Pero al margen de la recesión, Internet no es tratada por los anunciantes como otros medios publicitarios (radio, TV, prensa) en los que el anunciante paga por visionado, sino que se está evolucionando del pago por *banner* o *clic* en el mismo a la interactividad y las campañas publicitarias en las que el *banner* es el cebo que conduce a una Web del anunciante con contenidos, para tratar de que el mensaje impacte y sea recordado.

La razón de la caída es justamente ésta: el mensaje del anunciante no es captado por los clientes. La falta de diferenciación y la sobresaturación ante los cientos de páginas Web y los anuncios que en ellas encuentra hacen que el anunciante no encuentre rentable invertir en un medio en el que el impacto de su anuncio es bajo o directamente nulo, mientras que en los otros medios tradicionales tiene seguro un mayor impacto y seguimiento.

En resumidas cuentas lo que nos encontramos en la actualidad es en un mercado publicitario bajo mínimos, que ha llevado a la mayoría de los portales a replantearse su modelo de negocio. Esto implica casi siempre intentar cobrar por los contenidos ofertados, con una resistencia de los consumidores a pagar por ellos que ha llevado a la quiebra a bastantes de estas compañías.

Como ejemplos más recientes de esa resistencia del consumidor a pagar lo tenemos en los portales en Internet de diarios nacionales de España como EL PAÍS (**www.elpais.es**), cuya Web ha pasado de ser la cuarta Web española en Internet hasta el 2.003 en números absolutos, con 1.120.000 visitantes únicos por día, según el Estudio General de Medios de la Asociación de Investigación de los Medios de Comunicación (**www.aimc.es**), a salirse de este sistema de medición por el descalabro sufrido por su reciente cambio a una Web de acceso restringido a suscriptores que pagan unos 50 euros al año por acceder al diario en la Red[16].

---

[15] *El Marketing y Publicidad en Internet*. Estudio de la AGEMDI (2.002):

<www.aece.org/recursos.asp>.

La publicidad online no levanta cabeza, por ahora (2.001), artículo de *Baquia*:

<http://www.baquia.com/com/20011205/not00002.html>

[16] Estudio General de Medios (2.003), de la Asociación de Investigación de los Medios de Comunicación (**www.aimc.es**):

<http://download.aimc.es/aimc/datosegm/internet.pdf >

Aunque las causas habrían de ser analizadas más profundamente, nos interesa de momento únicamente los efectos de esta caída: resulta extremadamente arriesgado lanzarse a ofrecer contenidos en la Red, tanto por la nula rentabilidad de la publicidad como la resistencia de los potenciales consumidores de esos contenidos a pagar por ella. Y esta situación tardará aún bastantes años en cambiar

4.1.4 Internet en la actualidad

El acceso a Internet

Un estudio de una consultora norteamericana, *Jupiter Media Matrix*[17] cifraba en el 87% el porcentaje que representaría la banda ancha frente al acceso tradicional mediante módem de 56Kbps el mercado de los ISP en el año 2.004. La consultora erró en sus optimistas previsiones, dado el estancamiento de la banda ancha a nivel mundial, y con ella la inmensa mayoría de las consultoras, que daban por seguro que en la actualidad hasta las lavadoras estarían conectadas a Internet.

Las causas de ese estancamiento hay que buscarlas en la falta de contenidos en las redes que requieran conexiones de banda ancha: los clientes domésticos han descubierto que para consultar sus cuentas corrientes o leer el periódico no necesitan más que su módem de 56 *Kbps*, y la única razón que hace incrementar el uso del ancho de banda son usos no demasiado legítimos (descarga de películas o música). Se palpa un crecimiento en la necesidad del ancho de banda, pero no tan drástico como se llegó a pensar en su momento, y eso ha pesado en bastantes compañías que apostaron fuerte por lo contrario, situación que ha llevado a una caída de los precios del ancho de banda bastante grande, por exceso de oferta de acceso y la necesidad de estas operadoras de alcanzar beneficios.

El acceso en los Estados Unidos

Veamos de manera detallada el caso de los *EEUU*, donde la gratuidad de las llamadas locales es otro factor más, ya que los usuarios únicamente pagan a su *ISP* por el servicio de acceso a Internet. Situación aparentemente totalmente opuesta a la que tenemos en Europa, donde lo gratuito suele ser la conexión y se paga al operador telefónico por la llamada. Pero es sólo aparentemente: en Norteamérica las únicas llamadas gratuitas suelen ser las realizadas a clientes de la misma operadora en la misma ciudad, con lo que el servicio de Internet tiene que ofrecerse a nivel local en cada ciudad, siendo a fin de cuentas la propia compañía telefónica la única capaz de ofrecerlo de manera competitiva (un *ISP* puro debe desplegar un nodo de acceso en cada población y por cada operadora, u optar por establecer un número de acceso gratuito).

En *EEUU* la oferta de banda ancha se concentra fundamentalmente en las operadoras de cable (mucho mejor posicionadas que en Europa debido a la larga tradición de la televisión por cable existente en aquel país) y en el DSL, de reciente despegue pero con mejor futuro por su bajo coste de despliegue. En Norteamérica la tecnología DSL implantada

[17] Estudio de la Júpiter Media Matrix (enero del 2.001):
<http://www.jmm.com/xp/jmm/press/2001/pr_012501.xml>

es al igual que en Europa la conocida como ADSL, con lo que hay una asimetría entre el canal de subida y el de bajada sustancial, siendo el primero más reducido (aunque ciertos operadores vienen ofreciendo igual tasa en subida y bajada a precios significativamente mayores). Pero a diferencia de Europa los operadores han optado por dotar a sus conexiones de velocidades muy superiores a las que disfrutan los usuarios europeos, aprovechando que las centralitas norteamericanas disponen de un menor número de usuarios y por tanto hay más ancho de banda por abonado por la menor densidad.

También existe allí la posibilidad de acceder mediante módem con tarifa plana (aunque a diferencia de aquí allí es válida para cualquier hora del día) a un precio de 22 $/mes en la mayoría de las compañías (a fecha de marzo del 2.003), con lo que es de entender la resistencia en el pase a la banda ancha, cuyo coste está rondando los 40 $/mes, y limitado en muchas ocasiones a las grandes urbes de la costa oeste y de la este.

Para mostrar el grado de competencia en el mercado estadounidense, lo mejor es enumerar los mayores operadores existentes y sus ofertas a fecha de marzo del 2.002 para el mercado doméstico y empresarial, así como analizar la evolución de esas compañías, comenzando por la decana de todas, la *Bell*.

Bell y la America Telephone &Telegraph (www.att.com)[18]

El otrora monopolio telefónico norteamericano, la *Bell Telephone Company*, fue disgregado en 1.982 en 7 *RBOCs* (*Regional Bell Operating Companies*), más conocidas como *baby-bells*, quedándose siete corporaciones regionales (que a su vez poseían el control de las 22 compañías de telefonía local que la *Bell* poseía) y una compañía de telefonía de larga distancia, la *AT&T*. La separación de la compañía fue a raíz de una demanda judicial planteada por las demás compañías que había en el mercado y las cuales sólo controlaban la quinta parte del mismo. *General Telephone and Electronics* y otras muchas lograron que el juez federal *Harold Greene* sentenciara en 1.982 que la compañía ejercía un monopolio en el sector que dañaba al libre mercado, exigiendo su disgregación, efectiva dos años más tarde cuando el Tribunal Supremo refrendó esta sentencia.

En realidad desde la fundación en 1878 por *Alexander Graham Bell* y sus inversores, la *Bell Telephone Company* fue atacada en repetidas veces por el Departamento de Justicia o el de Comercio por su situación monopolística, pero hasta 1.982 y tras un largísimo proceso judicial que alcanzó incluso al Tribunal Supremo de los EE.UU., el gobierno federal no pudo cambiar esa situación. La *America Telephone & Telegraph* (AT&T) era una filial de la *Bell*, que por su carácter de intercambiador entre las filiales locales acabó por copar la telefonía de larga distancia,

[18] Historia de la Bell Telephony Company:
<http://www.bell.com>.

sólo atacada por *Microwave Communications, Inc.* (MCI) a partir de 1.969. Las operadoras locales no eran propiedad en su totalidad de *Bell*, ya que en su despliegue, las inversiones obligaron a contar con una gran cantidad de inversores y colaboradores locales, además de que incluso había compañías ajenas a *Bell* ofreciendo telefonía, pero en realidad poseía la mayor parte del tráfico y actuaba como un monopolio.

AT&T tras esta división mantuvo entre 1.982 y 1.995 una dura batalla en el mercado de la larga distancia y como *carrier*, mientras las *baby-bells* disfrutaban a pequeña escala del monopolio de acceso al cliente, hasta que en 1.996 la Ley de Liberalización de las Comunicaciones permitió finalmente a cualquier compañía competir por el bucle local, incluyendo a la misma AT&T que en 1.995 había vuelto a dividirse voluntariamente en tres compañías: *Lucent Technologies* (uno de los fabricantes de equipamiento de acceso y red más importantes en la actualidad), NCR (compañía informática de software y soluciones de gestión) y el negocio de telefonía e Internet que se mantuvo bajo el nombre AT&T. Ofrece conectividad vía módem RTB/RDSI a todo el país y con tarifa plana (entendida como la posibilidad de conectar a cualquier hora del día sin restricciones horarias ni temporales). Dispone de una filial de cable que opera en el Este (**www.attbroadband.com**), con tarifas por paquetes de televisión + Internet. Además, dispone de acceso ADSL de 768 Kbps/128 Kbps en muchas ciudades.

De las siete RBOCs en que fue escindida AT&T, el mercado ha permitido únicamente sobrevivir a tres de ellas después de múltiples fusiones entre ellas:

- *SBC Communications Inc.*, resultado de la compra de otras dos *RBOCs*: la *Southern New England Telephone* (SNET, **www.snet.com**), y *Ameritech* (**www.ameritech.com**). Integra además otras operadoras adquiridas con anterioridad, como la *Bell Atlantic*, y *Prodigy* **www.prodigy.com**. En acceso a Internet está ofreciendo en la actualidad tecnología ADSL con velocidades de 768 Kbps y 1.5 Mbps (ambas con 128 Kbps de canal de subida).

- Otras dos *RBOCs* fusionadas son *Bell South* (**www.bellsouth.com**) y *Qwest* (**www.qwest.com**), manteniéndose la marca de la segunda. La compañía ofrece por otro lado DSL (simétrico en canal de subida y de bajada) de 256 Kbps y de 640 Kbps. Tiene además un producto de conectividad de 7 Mbps a 275 $/mes que requiere contratar por separado el proveedor de Internet.

- La séptima *baby-bell*, la compañía *Cincinnati Bell* (**www.cincinnatibell.com**), fue adquirida por la compañía *Broadwind Inc.* (**www.broadwind.com**), y mantiene una

política agresiva de precios, con conexión RTB de tarifa plana y precios por debajo de los de la competencia.

Existen además otras compañías nacidas al calor de la liberalización del mercado, y que no existían hace dos décadas, con accesos vía DSL o incluso vía RTB, y de cobertura casi nacional en muchos casos:

- *Verizon Inc* (www.verizon.com), con accesos vía ADSL de 768 *Kbps* y de 1.5 *Mbps* (ambos con 128 Kbps de subida).
- *Sprint* (www.sprint.com), con acceso vía DSL simétrico de 512 *Kbps*.
- *WorldCom* (www.worldcom.com), protagonista de una de los más sonados escándalos financieros de los últimos tiempos, su dirección se lanzó durante el último lustro a una compra desenfrenada de otras operadoras, buscando crecer incluso falsificando estados financieros. Es propietaria de otra histórica compañía, como es MCI (www.mci.com). Ambas tienen dificultades financieras que les han obligado a renunciar a ofrecer DSL, limitándose su oferta al acceso y por bonos, sin entrar en la tarifa plana.

Los ISP americanos "puros"

Tanto AT&T como las otras compañías americanas nombradas hasta el momento son operadoras de telefonía: bien ofrecen nodos de acceso en las ciudades, bien utilizan un teléfono 1-800 (gratuitos para los usuarios) para facilitar el que sus clientes lleguen hasta ellos, pero siempre complementando su nicho real de mercado que son las llamadas telefónicas.

Aunque los ISP puros (aquellos que carecen de negocio de telefonía) parecen ver peligrar su negocio con la competencia fuerte de las operadoras telefónicas y de cable (como ocurre en Europa), aún hay margen debido a la diferencia en el marco regulatorio y a la clara diferenciación hecha por alguno de ellos.

El mejor ejemplo de capacidad de diferenciación de mercado es *America Online* (www.aol.com), el mayor ISP del mundo y con puntos de acceso internacionales para sus clientes en más de ochenta países, y que es capaz de captar clientes fieles. AOL son las siglas de *America Online*, y es caso de estudio por su evolución en muchas facultades de Economía. Hablar de AOL es hablar del éxito de *Steve Case*, su fundador en 1.985, cuando se llamaba *Quantum Computer* y ofrecía servicios a los jugadores online de los juegos *Commodore*. En 1.991 se rebautizó como *America Online*, y salió a bolsa en fechas muy tempranas, en 1.992.

Esta compañía absorbió en 1.998 al otro gran ISP histórico, *Compuserve* (www.compuserve.com), y también a otra compañía histórica en Internet como era *Netscape* (www.netscape.com), después

de su debacle tras ser barrido del mercado su navegador por *Microsoft* (www.microsoft.com). En enero del 2.000 se fusionó con *Time Warner* (www.aoltw.com), en una operación que fue más considerada una absorción por parte de AOL, una compañía que en aquel momento con una quinta parte de los trabajadores de *Time Warner* y generaba igual beneficio.

Dos años más tarde y con las cosas ya más calmadas, Steve Case ya no controla la compañía fusionada, pero se ha de admitir que en estos momentos AOL sigue siendo no sólo una referencia en Internet, sino una compañía con una capacidad tremenda para generar beneficios basándose en su diferencia con un ISP tradicional: AOL 8.0.

A diferencia de otros ISP los usuarios de AOL reciben algo más que un acceso a Internet, reciben el software para conectarse (AOL 8.0 es su última versión) y también un entorno de navegación cuidado y adaptado por AOL a las necesidades de sus suscriptores. Logra de esta manera unos usuarios finales integrales (que mantienen con AOL servicios que en otros ISP se buscan fuera) y cautivos, dado que los servicios ofrecidos son exclusivos para usuarios de AOL. Tanto es así que MSN para lograr arrancar clientes a AOL ha desarrollado una herramienta de migración de la configuración, *TrueSwitch* (www.trueswitch.com).

Naturalmente *America Online* limita bastante su oferta al mercado residencial, pero tiene además otra vía de ingresos más alta que la competencia: el que el software de acceso esté controlado por la operadora le permite además insertar publicidad y productos, y éstos estar activos en poco tiempo en todos sus usuarios, con lo que la publicidad y los programas ofrecidos a AOL tienen seguro un mercado de millones de lectores, circunstancia que AOL también aprovecha para obtener ingresos por esas licencias y publicidad. De hecho pese a haber adquirido *Netscape* y su navegador, aún pasó bastante tiempo antes de que abandonara el navegador de *Netscape*, no sólo por funcionalidad, sino por la mengua de ingresos.

Su hoy compañía filial *Compuserve* (www.compuserve.com), que adquirió en 1.998, mantiene su marca y actividad de manera disgregada, merece ser comentada si queremos ser justos y atribuir a quien se debe la idea en la que se basa el éxito de AOL: ofrecer Internet y "algo más", ya que fue Steve Case quien adquirió esta compañía tras haber copiado su idea y haber crecido a la sombra de este grande de Internet, al que tras superar acabó por absorber.

Compuserve es mucho más antigua que AOL: data de 1.969, cuando se creo en Ohio como compañía informática especialista en bases de datos de negocios. Su incorporación a Internet data de 1.978, cuando a sus clientes comenzó a ofrecerles correo electrónico, y ya en 1.983, disponía de una BBS de venta de productos de sus clientes. Cuando en 1.980 comenzó a colaborar con Associated Press (www.ap.org), la compañía estaría estableciendo las bases de lo que luego sería *Compuserve 1.0*, un software para que sus suscriptores pudieran

acceder de manera simple a noticias, correo y ficheros. *Compuserve* era ya desde los 80 un ISP si tenemos en cuenta que sus clientes accedían por módem a sus servidores, e hizo además hizo una decidida apuesta internacional desde el principio: ya en 1.987 estableció filiales en Europa y Asia (filiales que además no pertenecen hoy a AOL y que son ISPs relevantes, como el japonés *Nifty*, www.nifty.com).

Compuserve únicamente cometió un error: no abrirse a la banda ancha, que fue justamente lo que hizo de manera decidida Steve Case desde 1.997 y lo que le permitió seguir creciendo cuando *Compuserver* se estancó: únicamente en acceso residencial mediante RTB no hay viabilidad, tal y como demuestra *ExciteAtHome Corp*, un ISP puro que a principios del 2.002 suspendió pagos al ser incapaz de soportar los gastos de todos sus puntos de acceso (llamados *Point of Presence*, *POP*) le suponían frente al número de usuarios conectados.

La supervivencia para estos más de cien *ISPs* llega de la mano del DSL, ya que el organismo de regulación federal, la FCC, sigue favoreciendo su permanencia mediante una regulación que separa el transporte de señal DSL del servicio de acceso a Internet, que puede ofrecer un ISP distinto a la compañía que instala el DSL y que posee el par de cobre. Lo único necesario es disponer de caudal de acceso a Internet y estar registrado en alguno de los puntos de intercambio creados al efecto, teniéndose una situación similar a la que hemos nombrado con *Infovía* en España.

Al paraguas de esta situación pueden seguir existiendo operadores como *Dakotacom.net* (www.dakotacom.net), *Matrix Broadband* (http://www.dslbroadband.com), *First Link Technology, Inc.* (www.firstlink.com)…y así hasta más de cien compañías[19]. Por ejemplo *EarthLink* (www.earthlink.com) ofrece conexión DSL a través de algunas de las *baby-bells*, y conexión RTB mediante un teléfono 1-800 (de llamada gratuita) disponible en muchos condados de EE.UU.

Y también está el intento de *Microsoft* de crear un ISP a partir de su portal *MSN.com* (www.msn.com), tanto por conexión RTB como por conexión DSL cuya velocidad varía según la zona de conexión y el operador de DSL local al que haya que recurrir. El objeto real de *Microsoft* es competir directamente con *AOL*.

Los operadores de cable

Al igual que en la telefonía, existe una gran disgregación a nivel nacional en EE.UU. en el tema del cable, con casi una compañía por ciudad o área cableada. Pero en el último lustro y a base de fusiones y alianzas conducen a un gran grupo nacional. En concreto ese grupo es *Time Warner* (la filial de cable tiene página en cada área,

[19] Listado completo de los ISP de EE.UU.:
<http://www.isp.com/isps/NationWideISP.html>

como por ejemplo la de *Nueva York*, www.twcnyc.com y www.sicable.com). *Time Warner* podría ser considerado actualmente el ISP con mayor número de suscriptores a nivel planetario.

En EE.UU. hablar del cable es hablar de la televisión, debido a que (como contrapunto a la situación europea) en las zonas urbanas siempre ha sido preferido frente a la radiofrecuencia para la distribución de la televisión a gran escala, ya desde 1.972, en que fue regulado. Esta situación se da como norma general en los países de cultura sajona, en los que los núcleos urbanos son muy extensos por ser la vivienda adosada más común que en otras culturas europeas. En algunas culturas europeas, pese a ser común este tipo de vivienda, al ser países prácticamente sin relieves en las grandes urbes (Francia y Alemania) las antenas de radiodifusión son también viables. Pero en los EE.UU. el tamaño de las áreas urbanas y la orografía haría necesarias gran cantidad de antenas de repetición, siendo por tanto más común que la señal de televisión llegue a través del cable.

En cambio en Europa la aparición de redes de difusión terrestre por cable está despegando desde la última década como alternativa para introducir en el hogar todo un paquete de servicios (Internet, telefonía y televisión) de una manera alternativa a las operadoras telefónicas, aunque con un lento despegue por los costes.

El hecho de que en América estas compañías dispongan de una gran infraestructura (aunque vieja, y basada en cableado coaxial, con poco ancho de banda si lo comparamos con la fibra óptica) hace que deban ser tenidas en cuenta, habida cuenta si nos fijamos en que su producto de acceso a Internet se encuentra integrado en paquetes junto a la telefonía y la televisión, siendo por tanto de nuevo un conjunto diferenciador del resto de compañías.

La situación en Europa con los monopolios

Tras la aparición del teléfono en el siglo XIX y su posterior extensión, en EE.UU. se favoreció con la competencia la existencia de pequeñas compañías, frente al modelo tomado en la mayoría de países del mundo que era el crear una única empresa pública, lo que por otro lado garantizaba la solvencia y sobretodo el control sobre las comunicaciones. Para lograr un control sobre el mercado lo que se hizo fue crea un árbitro que velara porque se cumplieran tanto los derechos de los consumidores como las reglas de libre mercado, la FCC, creada en 1.934. Esta situación favoreció frente a Europa el desarrollo de la Internet en los años 90, al haber una mayor oferta en acceso telefónico y no estar en EE.UU. la totalidad del bucle local en manos de un único operador, trasladándose la competencia existente en la telefonía a Internet.

Pese a que Europa y muchos países del mundo han iniciado una corriente liberalizadora en Telecomunicaciones, si nos fijamos en España, ha sido un fracaso, ya que en España el acceso a Internet esta totalmente en manos de los operadores telefónicos: a diferencia de

EEUU en España el sector del ISP no existe ya ni podrá existir al margen de una operadora telefónica, y aún dentro de las operadoras, la dominante con diferencia es el antiguo monopolio.

En España cualquier operador con licencia de telefonía puede solicitar de Telefónica que las llamadas le sean desviadas a él. Pero las llamadas a Internet son tratadas en la factura como una llamada tradicional y por tanto, ningún cliente escapa de pagar el "peaje" al operador dominante. Si la línea no es de *Telefónica* la situación aún empeora: no hay obligación de permitir siquiera la interconexión hacia otras redes, y las operadoras de cable vienen bloqueado el acceso a los nodos de Internet desde sus redes telefónicas para hacer cautivos de su conectividad a sus clientes (operadoras como *Ono* sólo autorizan a sus clientes a marcar del rango 908 y 909 el número de la operadora).

En Norteamérica, gracias a la Ley de Telecomunicaciones de 1.996, el acceso a la última milla (el tramo entre el cliente y la centralita de telefonía) es posible para cualquier operadora de banda ancha que quiera conectar DSL a un abonado, u obtener del cliente el tráfico de todas las llamadas de larga distancia. En España la teoría dice lo mismo, pero no ha pasado de la experimentación el acceso de otras compañías a las centralitas de *Telefónica*, porque el precio de subrogar parcial o totalmente una línea de esta compañía impide competir con la propia operadora telefónica, y sigue existiendo una desigualdad muy fuerte entre la fuerza y tamaño del operador dominante y sus competidoras, demasiado pequeñas y disgregadas.

De manera numérica se puede comprobar: a finales de 1.999 en *EEUU* las operadoras históricas alquilaban 117.000 líneas para servicios de DSL a otros ISP, mientras esas mismas operadoras tenían otras 386.000 líneas instaladas. Hablamos por tanto de menos de un 20% de penetración en 1.999, mientras que en la actualidad (con datos del 2.002) ese porcentaje ha subido hasta el 25 con medio millón de líneas alquiladas a terceros para acceso DSL[20].

En España el fracaso al respecto es total, con más del 95% de las líneas ADSL en manos del operador dominante. El operador histórico tiene un interés natural en resistirse a proporcionar el acceso desglosado a su bucle local y en discriminar a la competencia favoreciendo sus propios accesos, pero el órgano regulador, la CMT tiene parte de la culpa, por no restringir adecuadamente a *Telefónica* en sus políticas comerciales: es bastante común que el operador dominante lance ofertas puntuales para recuperar aquellos clientes que ha perdido, prácticas comerciales que siendo lícitas no benefician la libre competencia si ese operador mantiene cuotas de más del 90% en todos los campos.

Si bien es cierto que a largo plazo el desglose del bucle local puede atentar contra la inversión, tal y como los operadores dominantes

[20] Artículo de la Asociación de Usuarios de Internet (2.002):
<http://www.aui.es/biblio/documentos/telecomunicaciones/bucle/usa-dc-bucle.htm>

aducen (los nuevos operadores no invierten en redes propias, especializándose en vivir como rémoras del operador histórico) esta amenaza puede ser corregida con obligaciones de inversión asociadas a la concesión de las licencias (como ya se ha hecho con las operadoras de cable, aunque la crisis haya eliminado dichos compromisos ahora).

Al considerar que la comentada situación en España se está dando a mayor o menor nivel en los demás países europeos (a excepción del Reino Unido, con un mercado más liberalizado y próximo al estadounidense), no vamos a detallar la situación país por país, sino que nos limitaremos ahora a enumerar primero el panorama en líneas generales en Europa, y luego pasaremos a España. Nos limitaremos a nombrar algunos de los grandes operadores a nivel europeo en acceso, operadores que además están presentes en el mercado español, uno de los cinco más importantes a nivel europeo junto al francés, italiano, alemán e inglés:

- *British Telecom* (www.bt.com), que sufrió una fuerte crisis financiera por una política errónea de crecimiento sin control, era considerada la operadora histórica que mejor había sabido adaptarse al nuevo mercado en competencia. Tras tener el dudoso honor de ser la empresa inglesa con mayores pérdidas, su actual prioridad es reducir su deuda y abandonar todo aquel mercado que no le sea rentable. En temas de Internet dispone de una de las mejores redes de fibra óptica de Europa que gestiona desde su filial *BT Ignite*, orientada a la empresa. Aunque mantiene productos para el mercado doméstico a través de *BT OpenWorld* (www.btopenworld.com), se está concentrando claramente en el sector empresarial. En España adquirió en 1.999 el ISP que mayor cuota alcanzó sin tener detrás a operadora alguna, *Arrakis* (www.arrakis.com), que sigue ubicada en el parque tecnológico de la Cartuja y orienta también hacia el mercado empresarial.

- La otra gran compañía del Reino Unido es *Cable & Wireless* (www.cw.com), con más de 130 años de historia y muy bien posicionada en las antiguas colonias británicas.

- *France Telecom* (www.francetelecom.fr) es la operadora histórica francesa, casi una empresa pública por la condición de funcionarios de sus empleados y una fuerte intervención estatal. En 1.996 creó la filial de Internet que ha ido lanzando por distintos países europeos con mayor o menor éxito, *Wanadoo* (www.wanadoo.fr), que en concreto en España es ya el segundo en importancia como tal. En nuestro país la decidida apuesta por

Internet ha sido consecuencia del fracaso de la matriz en el otro segmento, la telefonía, donde su filial *Uni2*. En 1.999 también adquirió uno de los pocos ISP que tenían un servicio técnico muy bien valorado por los usuarios, como era *Centre Telematic Valencia, scp* (CTV).

- La operadora alemana *Deutsche Telekom AG* (www.telekom.de) y su filial de Internet llamada *T-Online* (www.tonline.com) son hoy por hoy dos pesos fuertes a nivel europeo. Siguiendo un poco el modelo español, su filial compite por la matriz y mantiene una política de gestión y adquisiciones propias, ralentizadas a partir del año 2.000. La filial de Internet posee en propiedad de un banco por Internet (www.comdirect.de), una librería virtual, y filiales en Austria, la República Checa, Francia y España (ya.com).

- *Tiscali* (www.tiscali.com) es un caso atípico en Europa: es un verdadero ISP nacido sin tener detrás a operadora telefónica alguna, y que está alcanzado fuerza con filiales que con la misma marca que la matriz, ya operan en la mayoría de países europeos, además de Brasil y Sudáfrica (estos dos debido a que en su política de adquisiciones se hizo con *World Online*). La empresa nació como tal en 1.996 en la ciudad de *Tiscali*, en la isla de Cerdeña, y hoy por hoy dispone de capacidad incluso como *carrier*.

En el camino han quedado otras grandes compañías, como *Telecom Italia*, que aunque no desaparecen abandonan sus planes de expansión y se concentran en su país de origen, o las filiales desplegadas por AOL en algunos países (entre ellos España), que han acabado por ser liquidadas por falta de mercado.

Vista la situación a nivel europeo, a nivel nacional hay que reseñar que están presentes gran parte de estas compañías listadas antes a través de filiales, pero a excepción de *Wanadoo* y tal vez *ya.com*, en este país la fuerza en Internet en el mercado doméstico y empresarial está en manos de la operadora tradicional, *Telefónica*, que ofrece Internet a través de múltiples filiales y marcas (*Terra*, *Telefónica Data*, *Telefonica.NET*...) y copa también el acceso a través del ADSL.

ISP puros e independientes en España ya no hay (fueron adquiridos por operadoras con mayor capacidad económica presentes en este país). Y además es significativo que ante el fracaso de la competencia en arrancar cuota de mercado, la reacción que ahora se advierte y potencia desde el Gobierno es la creación de un grupo fuerte, surgido de las fusiones de casi todas las operadoras de cable y el grupo *Retevisión*, como es *Auna*. Este grupo ha

vendido su filial de Internet generalista a *Wanadoo*, habiendo preferido la compañía invertir su esfuerzo en el área del cable.

Otras operadoras relevantes del mundo

Aunque con las compañías europeas y americanas poseen filiales por casi todos los países del mundo, aún hay unas operadoras que no hemos nombrado y que por volumen, dimensión o tradición convendría nombrar:

- La *Nippon Telegraph & Telephone Company* o NTT (www.ntt.co.jp) es el operador histórico en Japón, una corporación con presencia tanto en telefonía móvil como en Internet, y que cuenta con filiales en gran parte de los países de su entorno. En Japón el ISP más antiguo es *Nifty* (www.nifty.com), surgido en 1.986 como filial del *Compuserve* y hoy propiedad en su totalidad de la corporación *Fujitsu* (www.fujitsu.com). Existen otras operadoras como *Biglobe* (www.biglobe.ne.jp).

- En China, la apertura hacia Occidente se hace bajo tutela gubernamental, siendo el único país que exige no sólo el registro previo de los ISP y su autorización oficial, sino que obliga a permitir las escuchas gubernamentales y el control sobre los contenidos a los que acceden los usuarios (sencillo ya que todo el tráfico pasa por servidores gubernamentales, que filtran e impiden a discreción). Existen grandes ISP no vinculados a ninguna telefónica, como *Sing.net* (www.signet.com.sg), extendido desde Singapur. La telefónica más importante de China también está presente, la PCCW (www.pccw.com).

- En Brasil también está un ISP que aspiró en su momento a extenderse por todo el mundo y que pese a limitar su actividad a Latinoamérica, sigue siendo importante, como es UOL (www.uol.com.br).

El *GigaADSL* en España

En España el diseño de *Megavia ADSL* (que es como se conoce al acceso a través de ADSL implantado por el Gobierno y suministrado por *Telefónica*) no ha favorecido la apertura del mercado a otros operadores. Más bien al contrario ha acelerado la desaparición de los ISP alternativos.

Veamos primero qué opciones ofrece el operador dominante en la actualidad: cuando se lanzó el ADSL y se diseñó la red de *Megavía ADSL* se intentaba que la situación previa de *Infovía* se reprodujera en el ADSL: Telefónica ponía la red y comunicaba al y el ISP, y éste sólo se encargaba del acceso a Internet

Pero fue un fracaso, porque las conexiones de banda ancha requieren un buen caudal de acceso a Internet en comparación con los módems de 56 *Kbps*, y la mayoría de los ISP además no pudieron competir con la competencia de *Telefónica*, que era a la vez la que suministraba el servicio y la que a través de filiales, los ofertaba como otro ISP más a los usuarios. Otro factor fue la desastrosa gestión inicial por parte de los ISP de la atención técnica, y la dependencia del tramo inicial del servicio con *Telefónica* y su red, que era aprovechada además por el operador histórico: las incidencias de sus clientes siempre se resolvían antes que las de los de la competencia cuando afectaban a tramos de la comunicación que los ISP no podían controlar.

Otra razón de peso fue que en el diseño por parte del Gobierno de *Megavía ADSL*, se limitó terriblemente la oferta a tres velocidades concretas, con lo que no existía diferenciación entre las ofertas de Telefónica y de los ISP, siendo las velocidades del ADSL (asimétricas, con mayor caudal de descarga desde Internet que de subida hacia ella):

- 256 Kbps. de canal de descarga /128 Kbps. de subida
- 512 Kbps. de canal de descarga /128 Kbps. de subida
- 2048 Kbps. de canal de descarga /512 Kbps. de subida

Con esta tecnología, Internet llega al abonado a través del par de cobre tradicional, sobre el que se sitúa una señal cuyo espectro no interfiere la señal de voz

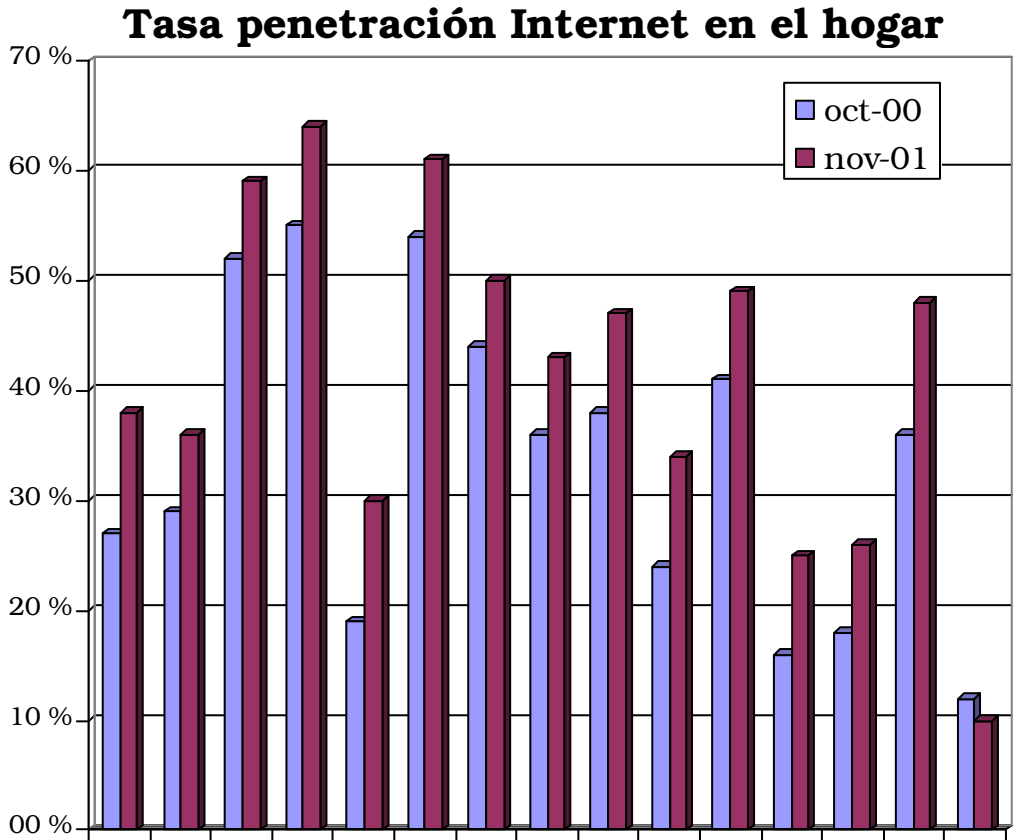
Desde enero del 2.002, con la entrada en vigor de la liberalización total, cualquier operador puede además instalar dentro de las centrales de Telefónica el equipamiento necesario para que todo el bucle de abonado sea controlado por él (a excepción del cable que va desde la central al abonado, que sigue siendo de Telefónica y por el que se paga una cuota para cubrir ese uso y el de espacio en la central), y no tenga por tanto que depender de *Telefónica*. Esta fue desde un principio la acción tomada en EE.UU., donde hoy una de cada cuatro conexiones de banda ancha está instalada por un operador distinto al propietario de la línea telefónica, mientras en España apenas suponen una de cada veinte.

En EE.UU. ha sido posible porque los ISP al acceder a las centralitas e instalar sus equipamientos, personalizan mucho más el producto (velocidades distintas a las tres posibles en España) y controlan mejor las incidencias.

Y además de tarde, la posibilidad de entrar a las centralitas de Telefónica tampoco ha tenido de momento mucho éxito: las pruebas se siguen sucediendo y dos años después apenas son algunos cientos las líneas alquiladas en todo el país, y la culpa no es sólo del operador histórico (que puede bloquear la apertura), sino también de los nuevos operadores que no ven muy claro el negocio teniendo en cuenta los costes de alquiler fijados por el Gobierno por ocupar las instalaciones del operador histórico.

Pese a todo lo comentado, ya hay un millón de conexiones ADSL en España, por lo que la falta de competencia no parece haber mermado el despegue de las conexiones. Pero para tratar de establecer alguna conclusión es mejor recurrir a los datos del EUROFLASH realizado por *Eurostat*[21], organismo de la Unión Europea encargado de las estadísticas. En este estudio de campo del 2.001 se analizó la penetración del Internet en el hogar (del 38% dentro del conjunto de los 15 países actualmente miembros de la Unión Europea).

En este estudio se analizó también la mejor oferta disponible en banda ancha en cada país para una conexión de un megabit tanto de tipo ADSL como de cable. Estos datos resultan de gran interés para ver cómo la situación de cada país de la oferta va pareja a la tasa de penetración.



	DE	BE	DK	NL	FR	SE	FI	LU	AT	IT	UK	ES	PT	EI	GR
oct-00	27	29	52	55	19	54	44	36	38	24	41	16	18	36	12
nov-01	38	36	59	64	30	61	50	43	47	34	49	25	26	48	10

Ilustración 4-7 Tasa de penetración de Internet en el hogar
Fuente: Estudio EUROFLASH de diciembre del 2.001

[21] FLASH EB Número 112: *Internet and the general public*. Realizado por *Gallup Europe* para el *Eurostat* (Diciembre del 2.001):

<http://europa.eu.int/information_society/eeurope/benchmarking/list/2001/index_en.htm>

De Grecia no se hacen constar datos de costes de la conexión, dado que sólo tiene una operadora pública y problemas técnicos le impiden ofrecer algo que no sea módem a 56 Kbps. en muchas áreas del país: con una tasa actual del 10% y un descenso respecto al año anterior es el país con peor situación. Hay además otros que no disponen de mucha oferta o incluso de alguna de las dos tecnologías comentadas (Luxemburgo e Italia carecen de operadoras de cable que conecten los hogares a Internet, y en Irlanda se da la situación contraria, sólo están las compañías inglesas de cable al margen de la operadora pública *Eircell*). Los datos vienen a demostrar que España tiene junto a Portugal, las conexiones de ADSL más caras, mientras que en el sector del cable esa situación no se da al no estar presente en ese mercado operador dominante alguno.

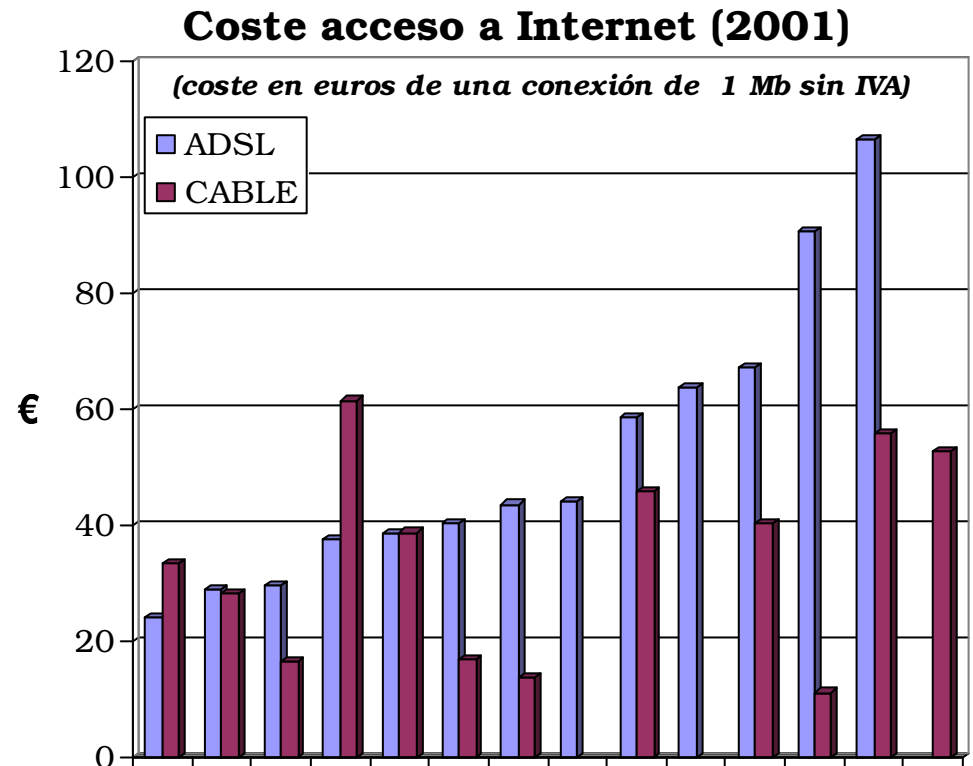
Los países aparecen ordenados en función del coste del ADSL e identificados por su código internacional de dos dígitos de país según ISO 3166[22] (que es el estándar escogido para asignar el TLD de cada país en el DNS).

Resulta paradójico pensar que una tecnología llamada a triunfar por su bajo coste de implantación resulte al cliente más cara casi siempre que el cable (dado que el ADSL aprovecha redes existentes, como son las telefónicas), mientras que el cable que requiere la apertura de zanjas y en Europa continental no disponía como Estados Unidos y las islas británicas de gran implantación tenga que partir de cero en redes.

[22] Listado de TLD correspondientes a cada país, según el ISO3166-1 aceptado por la IANA en el RFC 1591 en Marzo de 1.994:

<http://www.iana.net/cctld/cctld-whois.htm>

<http://www.iso.ch/iso/en/prods-services/iso3166ma >



	DE	BE	DK	NL	FR	SE	FI	LU	AT	IT	UK	ES	PT	EI
ADSL	24	29	30	38	39	40	44	44	59	64	67	91	106	
CABLE	34	28	17	62	39	17	14		46		40	11	56	53

Ilustración 4-8: Coste en euros del acceso a Internet

Los datos de este estudio para España respecto al cable son de cualquier forma criticables, ya que el estudio no tuvo en cuenta que la oferta tomada como referencia y que resulta especialmente barata (la conexión ofrecida por *Madritel*) tiene una limitación de descarga mensual irrisoria (120 MB/mes) a partir de la cual hay que pagar por cada *megabyte* transferido, siendo por tanto no comparable a un acceso (por ejemplo) de 34 euros/mes alemán y suministrado por *Ish GmbH* (www.ish.de).

Un ejemplo de mala regulación: Australia[23]

Australia tuvo al igual que los países occidentales una compañía que ofrecía servicios de Telecomunicaciones en régimen de monopolio hasta los 90, pero fue pionera en la liberalización del sector, iniciada en 1.992 y acabada en 1.997, año en el que en total competencia en todos los campos de las Telecomunicaciones existían más de 600 ISP en el país y 80 proveedores de larga distancia.

[23] Al respecto del estado de las telecomunicaciones en Australia, existe un completo artículo de Robert Clark en telecomasia.com de febrero del 2003 titulado *Australia's great broadband disaster*:

<http://telecomasia.net/telecomasia/article/articleDetail.jsp?id=53441>

El problema comenzó a partir del mismo momento de la total liberalización en 1.997 con el abuso de posición de Telstra (www.telstra.com), parte del antiguo monopolio OTC. El gobierno federal controla el 50.1% de la compañía, y en ningún momento evitó que esta compañía usara su tamaño para barrer a la competencia en cada uno de los sectores en los que se fue introduciendo.

En la actualidad, Telstra es la operadora de cable dominante, el operador de telefonía de larga distancia de mayor tráfico y el propietario de la mayoría de los bucles telefónicos locales del país. Controla además cadenas de televisión, la plataforma de pago del país y productoras de televisión y cine. Tanto es su dominio, que Alan Fels, responsable de la comisión federal que vela por la libre competencia de mercado en Australia (la Australian Competition and Consumer Commission, www.accc.gov.au) ha definido a Telstra como la compañía del mundo con mayor integración vertical y horizontal en un sector de las telecomunicaciones liberalizado.

Si nos fijamos en Internet, que es lo que nos afecta, menos del 2% de los australianos usan accesos a Internet de banda ancha, lo que sitúa al país entre las posiciones 30 y 40 del ranking a nivel mundial, bochornosamente por detrás de economías mucho peores como Estonia, y lo que es peor: bajando debido al elevado coste que supone una conexión de este tipo (un ADSL, por ejemplo, está sobre los 60 euros).

Las razones son políticas y técnicas: Australia es un continente aislado que ha necesitado grandes recursos en infraestructuras para estar unidos al resto del Planeta. En los 80 se pensó en incrementar la conexión con Europa y EE.UU. mediante el uso de satélites, debido a que hasta el posterior pleno desarrollo de la fibra óptica, se creía que el cableado submarino tendría menor ancho de banda que el satélite. Australia destinó casi 1.000 millones de euros de la época para poder desplegar tres satélites, y coincidiendo con la corriente liberalizadora de los 90 se decidió privatizar los satélites y a la propia operadora OTC.

Se crearon dos compañías: Aussat y Telstra. La primera se quedó con la red de larga distancia terrestre y los satélites, mientras la segunda dispuso de los abonados y las redes metropolitanas. La intención era que tras la liberalización todos pudieran acceder en igualdad de condiciones a las redes de Aussat. Una decisión política que luego resultó errónea.

Aussat no pudo mantener ingresos suficientes y acabó en manos de las compañías extranjeras Cable&Wireless y BellSouth, que la convirtieron en la segunda gran operadora del país para competir con Telstra, Optus (www.optus.com.au).

Telstra reaccionó desplegando su propia red de larga distancia con fibra óptica, mucho más evolucionada que la red original de OTC que Optus aún utiliza, y con un ancho de banda muy superior (aparte que le permite además transportar la señal de televisión de su operadora de cable). Mientras, los satélites de Optus quedaron fuera de escena para el transporte de información, ya que Cable&Wireless garantiza a su filial

una mayor conectividad mediante los cables submarinos de su propia red mundial.

La conclusión es que han confluido dos factores negativos en este proceso y que deberían ser tenidos en cuenta por los demás países: mala liberalización (se creó una gran empresa con todos los clientes e ingresos, Telstra, y otra que únicamente tenía la red y ningún ingreso si los operadores desplegaban su propia red) y mala regulación, ya que pese a que se siguió el modelo estadounidense de crear una autoridad independiente que velara por la libre competencia (la ya nombrada ACCC), ésta no ha sabido o podido limitar la fuerza del antiguo monopolio. Por el camino ha habido también escándalos políticos importantes (como el pase de directivos de Telstra a ocupar carteras ministeriales en relación con las Telecomunicaciones o directamente al ente regulador) y sobretodo, haber autorizado a Telstra licencias en cualesquier tipo de tecnología que fuera surgiendo (cable, telefonía, móviles...), licencias que además en muchos casos se denegaron a compañías extranjeras que podrían haber introducido algo de competitividad.

Por tanto, y al igual que puede acabar ocurriendo en muchos países europeos, Australia disfruta hoy de un monopolio de facto, con la diferencia respecto a los 80 que ahora el operador dominante es parcial o totalmente privado, lo que dificulta su control por parte del Estado.

Situación a nivel de *carriers*

Nos queda por hablar de las empresas que están por detrás de los ISP de tamaño medio o incluso de las grandes operadoras de cada país: los *carriers*, aquellas compañías propietarias de la mayor parte de las redes de fibra óptica.

Históricamente las redes habían sido incluso en Estados Unidos propiedad pública en mayor o menor medida, debido a los grandes costes de despliegue y su hasta entonces normal orientación hacia el mundo de la investigación.

Sin embargo, a partir de la eclosión en 1.995 del acceso a Internet en el hogar comenzó a incrementarse por un lado la necesidad de ancho de banda de los operadores (que esas redes públicas no iban a poder soportar) y por otro lado comenzaron a surgir nuevas oportunidades de negocio como es el ancho de banda bajo demanda (que se contrata para una situación puntual, como un congreso o feria), la facturación por diferentes franjas horarias (las de mayores consumos son las tarifas más caras)... etc. Es así como en una década se ha pasado de una capacidad de interconexión planetaria que no crecía al mismo ritmo que los usuarios, a otra en la que el *backbone* creado está infrautilizado, por ser demasiadas las empresas que se han dotado de red propia sin necesidad o esperando un crecimiento aún mayor del tráfico en la red.

Pero incluso hay otras compañías que se han especializado en crear redes dentro de las zonas comerciales e industriales más importantes de los cinco continentes. Como ha llegado un momento que la oferta ha

sido mayor que la demanda (sobretodo a raíz de la caída de la actividad a partir de la crisis desatada en el 2.000), todas estas compañías están provocando el hundimiento del precio del ancho de banda en Occidente, hasta hace unos años a precios ahora impensables: en términos de costes, en ciudades grandes de España es posible contratar anchos de banda de un *megabite* por menos de cuatrocientos euros al mes, cuando hace cuatro o cinco años, por cifras de ancho de banda de la décima parte se pagaba aún más. Han aparecido incluso espacios en Internet dedicados a este tipo de compra-venta, como **www.iber-x.com**, donde se pueden encontrar incluso ofertas de conexión entre diferentes capitales europeas de anchos de 155 *Mbps* a menos de un millón de pesetas/mes. Ante tanta bajada de precios las que no han podido aguantar sólo con esta actividad han comenzado a diversificarse hacia el alojamiento y el alquiler de espacio en sus instalaciones.

Pero convendría primero de todo explicar un poco cuál era la situación previa a una Internet en manos de las empresas privadas, recordando la época de NSFNET y otras iniciativas similares en Europa.

NSFnet y otros proyectos públicos

En 1.984, la NSF (*National Science Fundation*), empezó a diseñar un sucesor de alta velocidad para la ARPAnet, que se abriría a todos los grupos universitarios. Esta maravillosa red de alta velocidad usaba conexiones de 1.5 *Mbps* que superaba a los anteriores 56 Kbps. vía líneas serie (aún no existía el protocolo V.52 que años después nos dejaría conectar a esa velocidad creyéndola la más lenta del Universo) para conectar sus 6 centros de supercomputadoras.

La NSF financió también unas cuantas redes regionales (cerca de 20), que se conectaron a la red principal para permitir a universidades, laboratorios, etc. el acceso a cualquier supercomputadora. El conjunto de la red principal y estas 20 redes se dio en llamar NSFnet.

En diciembre de 1.991, el Congreso de Estados Unidos aprobó un documento que autorizaba a la NREN, la Red Nacional Educativa y de Investigación, como sucesora de investigación de la NSFnet, sólo que operando a velocidades de *gigabits*. La meta era una red nacional que funcionara a 3 *Gbps* antes del milenio, pero no hizo falta: en 1.995 la totalidad de la red fue vendida a *América Online* dado que ya había redes privadas de esa capacidad, y se consideró que el objetivo había sido claramente cumplido y que a partir de ese momento la NSF trabajaría con operadoras privadas. Había llegado la madurez de los *carriers* de Internet: ya no necesitaban fondos públicos.

Otros países y regiones estaban construyendo redes comparables a NSFnet, aunque siempre algo más limitadas al territorio de cada país, existiendo luego iniciativas que permiten la intercomunicación de estas diferentes redes. En concreto, la Unión Europea ha dispuesto siempre de una red paneuropea de investigación (hoy conocida como Dante y que acomete el proyecto *Géant*, **www.dante.net**) que actualmente opera también a nivel de *gigabits*. Siendo estas velocidades y ancho de bandas

bastante grandes, ya existen de todas formas proyectos que buscan superarlos, como por ejemplo la GTRN (*Global Terabit Research Network*, www.gtrn.net), que desde mayo del 2.002 busca con fibra óptica conectar capitales europeas a velocidades mil veces superiores: *terabits*.

En España la red académica se conoce como *RedIris*, y trata en estos momentos de implantar una red *gigabit* que supone una mejora sustancial, quedando el *backbone* con enlaces que en la mayoría de los casos son de al menos 622 *Mbps*. Esta red sigue siendo la más grande de España, aunque el conjunto de las redes de los *carriers* que operan en este país la supere en tamaño y tráfico: Su estructura y diseño además resulta loable por la redundancia que presentan sus conexiones entre nodos: cualquier red de fibra óptica comercial minimiza el gasto de tal forma que se dan sorpresivas faltas de servicio ante un daño en un simple cable de fibra óptica, que en una red mejor pensada permite desviar el tráfico a través de otras conexiones o rutas.

Ejemplos de estos fallos a nivel comercial hay bastantes: como fue el caso reciente de Telefónica ante un sabotaje en un cable situado en un punto concreto de la red que dejó sin servicio a media España en Junio del 2.002[24] o bien dos años antes, en Junio del 2.000, cuando la rotura del entonces único cable submarino que conectaba con Oceanía provocó un colapso del tráfico en Asia y Oceanía (por la gran dependencia que ambas subredes tenían entre sí en ciertos servicios).

En realidad *RedIris* únicamente aplica algo que debería ser común en Internet y no siempre lo es: que ciertos nodos considerados prioritarios deben poderse ver a través de diferentes conexiones, tanto para facilitar el tráfico en situaciones normales como para asegurarse de que en caso de caída de una de las líneas, el servicio pueda ser igualmente sostenido si usamos protocolos de enrutamiento medianamente automáticos (RIP, EIGRP, etc.).

[24] Cernuda, Olalla (14 de Junio del 2.002): *Telefónica denuncia el sabotaje de su red de fibra óptica*:

<http://www.el-mundo.es/navegante/2002/06/14/empresas/1024067053.html>

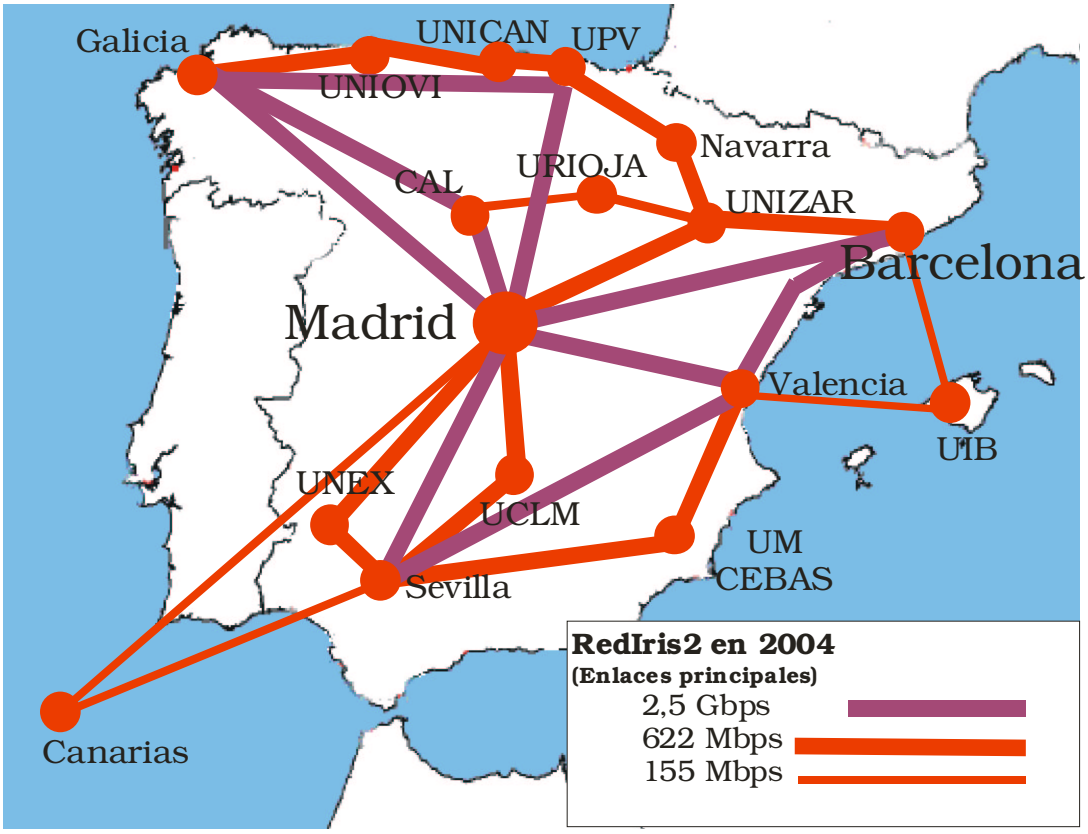


Ilustración 4-9: *Backbone* de RedIris en la actualidad

La práctica ha demostrado que esto no se da en redes más comerciales, en las que los enlaces tienen mejores características de ancho de banda, pero son demasiado sensibles a caídas. En concreto, la red *RedIris2* actualmente en implantación destaca por ser una malla con un anillo de conexión principal de 2,5 *Gbps* que se ha establecido entre Madrid (NACIONAL), Barcelona (CAT), Valencia (VAL) y Sevilla (AND), o bien el situado entre Santiago de Compostela (GAL), la Universidad del País Vasco (PAV) y la de Cantabria (CAL). Además incorpora por primera vez un SLA con las Universidades (acuerdo sobre el nivel de servicio se compromete a mantener) de menos de 8 minutos al mes de caída de un *PoP* (cada uno de los nodos indicados en la gráfica), así como una sustancial mejora en el acceso internacional al reemplazarse los enlaces STM-1 de Telefónica Data por dos enlaces de 2,5 *Gbps* (aunque por desgracia uno de ellos era con *KPNQWest* y tuvo que ser reemplazado a mediados del 2.002 por la quiebra de dicha compañía).

Por otro lado el hecho de que la gestión del *backbone* de Internet se haya privatizado, aparte de incrementar la oferta y reducir costes también ha traído problemas, como el incremento de la dificultad en el enrutamiento (demasiadas compañías) e imposibilidad de tener un acuerdo con cada una de ellas. Para solventar este detalle y agilizar el tráfico es por lo que nacieron los puntos neutros, algo así como unos intercambiadores de tráfico en el que cada operador instala un acceso a

su red y paga una cuota de mantenimiento del punto neutro. Por ejemplo en España existe uno en Madrid llamado *EspaNIX* (www.espanix.com) con más de seis años de historia, que gestiona en la actualidad un intercambio de tráfico de más de 6 gigabits/segundo. Además es uno de los miembros fundadores de Euro-IX, la asociación que agrupa a 20 de estos puntos neutros europeos.

A nivel de empresas privadas las mejores redes privadas disponibles en funcionamiento y en las que se pueden alquilar/comprar consumo son justamente las grandes operadoras como BT, Tiscali... Básicamente estaríamos repitiendo de nuevo en la mayor de los casos un listado de operadores que nos resultaría conocido. Por el contrario en EE.UU. la mayoría de las operadoras (a excepción de aquellas que trabajan en la larga distancia) apenas han invertido en ese aspecto, por lo que allí si que hay *carriers* exclusivos.

Los *carriers* y el tráfico intercontinental

Aunque en la actualidad existen una tecnología alternativa al cable submarino (el satélite) la capacidad de la fibra óptica ha hecho que la mayoría del tráfico intercontinental siga viajando bajo el agua y se reserve el satélite a sistema de respaldo o para ofrecer conectividad en aquellos puntos en los que desplegar una red terrestre supondría un coste mayor. Como luego veremos el satélite para la transferencia de datos y telefonía en Internet se usó más hasta la década de los 80 que hoy día (en aquellos tiempos los cables aún eran mayoritariamente de coaxial: lentos y costosos).

La historia del cable submarino se remonta a 1850, cuando se unió por primera vez Inglaterra y Francia. Hasta 1.985 básicamente estos cables estaban destinados a la telefonía y usaban señales eléctricas para la comunicación extremo a extremo: eran cables de tipo coaxial, caros de mantener, desplegar y con un consumo eléctrico y necesidad de repetidores cada poca distancia. Valga de ejemplo que SAT-1, cable que conectaba Sudáfrica con Europa a través de Portugal (a su vez conectado al Reino Unido y a Francia por idéntica tecnología) usaba para una longitud de 10.000 kilómetros de cable coaxial un total de 656 repetidores sumergibles (la señal eléctrica se degradaba y había que regenerarla) y 57 igualadores (dedicados a verificar la integridad del tramo), todo ello hasta llegar a la estación terminal de Ciudad del Cabo y sin tener en cuenta otro problema como era el intenso consumo eléctrico que realizaba el total de la red[25].

Con el desarrollo de la tecnología de fibra óptica, a partir de 1.983 esta situación ha cambiado notablemente, y ha permitido además la entrada de competidores frente a *AT&T*, que era hasta hace cuatro o cinco años propietaria de la mayor parte del *backbone* de Internet. Fue también esta compañía la pionera de la introducción de satélites de comunicaciones comerciales, con el lanzamiento en julio de 1.962 de

[25] Comunicaciones Eléctricas. Núm. 44/3 (1.969). *El sistema de cable submarino SAT-1*: <http://www.coit.es/museo/tecnolog/cable/sat1/sumario.pdf>

Telstar. En especial la explosión de redes entre Europa, Asia y Oceanía ha sido fuerte (ya que *AT&T* siempre priorizó la comunicación de cualquier destino hacia EE.UU. y no de estos entre sí, con lo que teníamos un mercado totalmente emergente).

Fue una consultora, la *Kessier Marketing Intelligence Corp* (KMI), dedicada a realizar el seguimiento de los proyectos de cableado submarino, la primera que vio posibilidades de negocio en competir con los grandes *carriers* de entonces en Internet, *AT&T* y *C&W*, creando redes alternativas, modernas y con precios más ajustados, comenzado en 1.997 el proyecto FLAG (*Fiberoptic Link Around the Globe*) con un banco saudita como inversor y dos operadoras como socio tecnológico, la japonesa KDD y *Nynex*. El proyecto FLAG comenzó por unir Londres y Oriente Medio a 8 *Gbps* y en la actualidad ya dispone de una red global tal y como publicitan a través de su Web (**www.flagtelecom.com**).

Ilustración 4-10: Red desplegada por el proyecto FLAG

Esta situación ha permitido a los fabricantes *Alcatel* y *Lucent Technologies*, líderes en el área de cableado submarino, incrementar su negocio y al mismo tiempo mejorar la velocidad de despliegue y reducir los costes. En concreto la primera compañía, la francesa *Alcatel*, dispone en estos momentos de una flota de ocho barcos[26] que sigue desplegando más y más conexiones de fibra: en concreto trece mil kilómetros de fibra sólo con el proyecto Apolo, destinado a triplicar la anterior capacidad de comunicación submarina entre Europa y EE.UU. Este despegue de esta tecnología ha dejado prácticamente fuera de la necesidad el uso de la que durante los 80 fue la estrella de las telecomunicaciones internacionales: el satélite (mucho más costoso de mantener y lanzar, y cada vez más restringido a su uso en la difusión de señales e Internet por parabólica comercial, antes que como *backbone* de Internet).

Otra compañía importante por el tamaño de su red es la británica *Cable & Wireless*, cuya red de fibra AS 3561 es considerada el mayor sistema autónomo del mundo (de hecho 3561 es justamente el identificador de ese sistema autónomo tal y como es usado para enrutamiento en redes con soporte de protocolos de enrutamiento

[26] Doré, Christophe (2.002): *Le Figaro Economie*. Consultable en la web de *Alcatel*: <http://www.alcatel.com/newslink/0204/pdf/neighborhood.pdf>

dinámico[27]). Su problema reside en que sus 460.000 kilómetros de fibra están desplegados, al igual que AT&T, en Europa y EE.UU. lo que le impide posicionarse en el emergente mercado asiático como debería.

Ilustración 4-11: Red de Cable&Wireless

No podemos dejar de nombrar la red troncal de la operadora española Telefónica, claramente regional y centrada en su área de servicio (la Península y Sudamérica). Dispone de presencia en EE.UU. y Europa a través de acuerdos con otras compañías para reducir la dependencia, ya que el tráfico de Internet está localizado mayormente en esta área, pero hoy día Telefónica sigue siendo la única compañía con cobertura global en toda América a excepción de Canadá.

Toda la red de fibra submarina y los enlaces entre distintos países de Telefónica está gestionada a través de *Emergia* (www.emergia.es), la filial en cuya Web se puede consultar el siguiente plano de su red:

Ilustración 4-12: Red de Telefónica

El por qué de la concentración de la red en Sudamérica y España es debido a motivaciones políticas: desde 1.956, en que se tendió el primer cable trasatlántico hasta Argentina, el TAT-1, Telefónica siempre ha buscado dominar un mercado con el que compartimos la lengua y en el que en la actualidad debe estar por el mero hecho de que la mayoría de estos países tienen filiales suyas. En concreto en 1.999, con la incorporación del cable Panamericano de 7.442 kilómetros de longitud, la red de Telefónica alcanzó los 89.000 kilómetros de longitud acumulada[28].

---

[27] *Whois* del ARIN *(American Registry for Internet Numbers)*: <http://ws.arin.net/cgi-bin/whois.pl?queryinput=a+3561>

[28] Web del Museo de las Telecomunicaciones de la Fundación Telefónica: <http://www.fundacion.telefonica.com/museo>

Otra de las compañías que más invirtió hasta el 2.001 en despliegue de una red intercontinental fue la norteamericana *QWest*, aunque la lista podría extenderse bastante más con compañías con un gran tamaño en la actualidad y cuya red es ya superior a la que toda Internet tenía apenas una década atrás. Aunque la compañía que más merece ser comentada por su crecimiento en Europa es la británica Colt Telecom Group plc (www.colt.es), fundada en 1.992 en Londres y que a partir de 1.995 se ha ido estableciendo en países europeos como Alemania, España, Italia, disponiendo en la actualidad de puntos de presencia en 32 ciudades de 13 países europeos, conectados por una red propia de más de 15.000 kilómetros de fibra óptica llamada *COLT EuroLAN*.

Ilustración 4-13: Red de Colt Telecom

En nuestro país hay una serie de compañías que han logrado hacerse un hueco en el alojamiento sin ser *carriers*: son compañías para las que su éxito les ha llegado tras una experiencia como ISPs tradicionales que no resultó rentable o dejó de serlo en cierto momento, actividad que han ido abandonando o mantienen ya a nivel residual, o bien han nacido directamente para ofrecer lo que están ahora ofreciendo, y que son las dos que ahora comentaremos, *Arsys* y *Acens*.

La primera es además la pionera en España, *Arsys* (www.arsys.es), y nació en La Rioja en 1.996 como otro cualquiera de los ISP que aparecieron al calor de *Infovía*. Tras un cambio de estrategia fundamental a partir de la aparición de *Infovía Plus*, abandonando casi totalmente el acceso a Internet, hoy día es la empresa informática más fuerte de esa zona, dedicada exclusivamente al alojamiento o *hosting*.

Un *data center* es una empresa que disponiendo de un caudal de acceso a Internet decente y dispone de varias conexiones (lo que en el argot se conoce como *multihomed*) ofrece a terceros su ancho de bando, sus instalaciones, su hardware, y su *know-how* para la configuración, mantenimiento y servicio de servicios de Internet, fundamentalmente correo y Web. También se les conoce como IPP (*Internet Presence Provider*), ya que ofrecen presencia en Internet a terceros.

Hasta que *Arsys* se lanzó a esta actividad, en España no había ninguna otra compañía capaz de ofrecerlo debido a los elevados precios del caudal por la inexistencia de competencia frente a *Telefónica*, que imponía su ley. La única solución pasaba por usar un operador virtual, con el servidor localizado en Estados Unidos para reducir los costes del caudal. Pero *Arsys* supo arriesgar y tras asegurarse un gran caudal se lanzó con gran éxito al *hosting* de páginas Web y correo, e incluso está hoy día presente en Europa en Francia (www.arsys.fr). Su éxito es haber logrado un tamaño lo suficiente grande como para lograr economías de escala en el alojamiento y en el acceso a Internet.

La otra compañía a comentar, *Acens* (www.acens.es), es una apuesta reciente por copiar el modelo de *Arsys*: Surgió a principios del 2.001 como una refundación de la versión española de *RapidSite* (www.rapidsite.com) que desde 1.997 venía ofreciendo lo mismo que *Arsys* pero sin lograr el éxito que los bajos precios de *Arsys* le han supuesto. Su éxito también ha desatado la competencia en el sector, que ahora pasa por su mejor momento.

Direcciones IPv4

Si queremos convertirnos en un ISP, necesitaremos estar en Internet y tener direcciones públicas. El actual espacio de direcciones viene condicionado por el empleo de un conjunto de protocolos conocidos como pila de protocolos TCP/IP, desarrollados para ARPAnet y que no podía prever la actual eclosión de Internet.

La capa 3 de esta pila de protocolos es la que contiene las direcciones de enrutamiento de los paquetes a nivel de todo Internet. Esta capa 3 es el protocolo IP, y la versión es la 4, de modo que se conoce como IPv4. Como se emplean únicamente 32 bits para estas direcciones de red, y además se da la situación de que el aprovechamiento del espacio de direcciones hasta tiempos recientes no había sido muy cuidado, el agotamiento de las mismas es casi total, y hace ya años que se viene desarrollando una nueva versión del protocolo llamada a reemplazar a IPv4 que se conoce como IPv6, pero cuya implantación supone cambios en el hardware y en el software usado en todos los dispositivos de Internet, con lo que se está haciendo bastante más difícil de lo esperado su implantación.

El organismo responsable a nivel mundial de la asignación de direcciones IP, tanto con el protocolo IPv4 como con IPv6, es el IANA (**www.iana.org**). Es también responsabilidad de la IANA la asignación de otros recursos de la red (por ejemplo la asignación de un determinado número de servicio dentro del llamado de puertos privilegiados, aunque en este caso las infracciones al RFC son bastante más numerosas de lo debido).

De los 32 bits de una dirección, se hicieron cuatro tipo de direcciones (caracterizadas por los dos primeros bits), que permitieron a la IANA otorgar bloques de diferente tamaño en función de cuál fuera el tamaño de la entidad que los solicitaba. Este organismo creado en los tiempos en que Internet era puramente académico o por lo menos no era comercial, actuaba con el principio de que quien pedía es porque necesitaba, y sin demasiados problemas otorgaba grandes cantidades de direcciones a organismos que no las empleaban demasiado, cobrando por ellas el precio simbólico de un centavo por cada IP.

La primera clase de direcciones, las de clase A, tienen su primer bit a 0 y contiene cada una de ellas la friolera de 16.777.214 direcciones IP dentro de cada red, o lo que es lo mismo, $2^{21} - 2$. Las 126 redes de clase A, cuyo primer octeto se identifica por estar entre 1 y 126, están ocupadas: fueron asignadas a grandes empresas o instituciones que nunca han hecho un aprovechamiento digno de las mismas (*Hewlet Packard*, *Apple*, IBM, etc.), pero una vez hecha la asignación el IANA difícilmente lo revoca.

Unas redes de tamaño algo menor son las de clase B, cuyos dos primeros bits son siempre 10, y cuyo primer octeto en valor decimal está comprendido entre el 128 y el 191. En total 16.382 redes que están también ya asignadas o dedicadas a otros menesteres, disponiendo

cada una de las organizaciones que poseen una de 65.534 *hosts* útiles (2^{14} – 2). Por último se creó un tipo C, de sólo 254 *hosts* por red y un total de 2.097.150 redes, y dos tipos D y E, el primero destinado al *multicast* y el segundo a fines experimentales. Las direcciones de tipo C son aquellas que comienzan por 192 hasta 223, las de tipo D entre 224 y 239, mientras que las de tipo E son las restantes 240 a 254. Ninguna de estas clases es asignada hoy día como tal a quien solicite una nueva, aunque se respeta la asignación original hecha, consultable en **www.iana.org/assignments/ipv4-address-space**, y encima hay rangos enteros destinados a otros menesteres:

- 127.0.0.0/0 está considerada el *loop* de prueba: cualquier equipo puede usar esta dirección como propia para hacer pruebas de funcionamiento de la pila de protocolos.
- 172.16.0.0/12, así como 10.0.0.0/8 y 192.168.0.0/16 son direcciones privadas, asignadas por lo general en las redes locales para equipos que puedan funcionar sin IP pública (aquellos equipos que no ofrecen servicios a Internet a equipos distintos de su espacio de direcciones privadas).
- 169.254.0.0/16 es el *Link Local Block*, otro bloque de direcciones especial usado por muchos equipos para la configuración por defecto de una IP.

Este derroche se ha pagado muy caro años más tarde: La razón es que si nos fijamos, de 2^{32}-2 (aprox. 4.300 millones) posibles *hosts* que los 32 bits de una dirección IP pueden dar de sí, el direccionamiento planteado inicialmente ya derrochaba 600 millones (ya que sólo 3.700 millones son útiles), y a la hora de asignar, era imposible que una compañía con necesidad de conectar 300 equipos pudieran asignársele dicha cantidad de IPs: se le daban directamente sesenta y cinco mil.

Para facilitar la distribución de las IPs apareció en 1.985 el *subnetting*, que consiste básicamente en utilizar máscaras de red distintas de los múltiplos de ocho, con lo que una organización que dispusiera de una clase C podría a su vez coger esos 254 *hosts* y cogiendo dos bits extras, generar dos subredes de 126 *hosts* cada una. De nuevo había que ir con cuidado en las divisiones ya que una división no demasiado cuidadosa conduciría a perder en la transformación parte de las direcciones. En 1.987 se dio un paso más allá con las máscaras de subred de longitud variable, reguladas por el RFC 1009, que permitían hacer *subnetting* sin obligar a que cada subred creada tuviera igual tamaño.

Pero la verdadera revolución llegó en 1.996 con el enrutamiento entre dominios sin clase o CIDR: se abandonaba sencillamente el uso de las clases tradicionales, se traspasaba la asignación de la IANA a organismos regionales, y se daba el poder pertinente a estos organismos sobre la asignación de aquellas clases pendientes de asignar, así como de cuanto tramo de rangos aún pudiera recuperarse, exigiéndose además que a partir de ese momento tuviera especial cuidado en dar las

IPs estrictamente necesarias. Todo ello para tratar de evitar un colapso en el crecimiento de la red, colapso que finalmente no se ha producido.

Se crearon tres registradores regionales (RIR, por sus siglas en inglés): APNIC, ARIN y el RIPE-NCC. Posteriormente, el ARIN se dividió y apareció un cuarto registrador, el LACNIC. Esta fragmentación en tres entidades de menor tamaño se hizo para facilitar el control sobre la asignación de IPs. Cada organización recibió unos tramos de los aún disponibles, y gracias a las técnicas desarrolladas con el CIDR se pueden otorgar incluso bloques de 16 IPs.

Realmente el problema que genera enrutar esto es bastante complejo en el *backbone* de una gran *carrier* de Internet: hasta ahora era un mismo destino el que nos permitía alcanzar una clase B de sesenta y cinco mil *hosts*, cuando ahora cuatro de esos *hosts* se pueden localizar en Asia y los demás en el extremo opuesto del planeta. La solución pasa por la asignación lo más territorial posible: dentro de los rangos que corresponden a Europa, las que se asignan a operadores españoles son consecutivas, para facilitar luego el enrutado entre y hacia ellas.

De todas formas lo que se han endurecido y mucho son las condiciones para la asignación, variando en cada registrador.

RIPE-NCC

El tramo correspondiente a Europa y regiones limítrofes (lo que incluye el Norte de África y parte de Asia) está formado por los siguientes rangos de tamaño equivalente a una clase A: 62.0.0.0/8, 80.0.0.0/8 hasta 82.0.0.0/8, el grupo 193-195.0.0.0/0, 212.0.0.0/8, 213.0.0.0/8, y por último el 217.0.0.0/8. Todas las IPs asignadas de momento dentro de este rango están localizadas en Europa y Asia, debido a la baja demanda que se produce en África.

El RIPE-NCC está formado por empresas del sector y otros organismos como universidades, centros de investigación y los propios NIC de cada país, Para ser miembro del mismo, la condición indispensable es, además de pagar las cuotas correspondientes, disponer o gestionar el equivalente a una red /20, es decir: 4.096 hosts. RIPE-NCC (www.ripe.net) está situado en Ámsterdam.

En lo que respecta a la asignación de IPs, no ofrece rangos de menos de 1024 IPs (una red /22, tal como se denota en la notación creada con la aparición del CIDR). Por lo que a diferencia de antes, ni los particulares ni los pequeños ISP pueden acudir directamente al RIPE para solicitar un conjunto de direcciones: para obtener la delegación de un rango menor hay que acudir a alguien que ya disponga.

Los espacios son asignados a miembros del RIPE (denominados *Local Internet Registries*), teniendo que justificar su necesidad para la asignación (para evitar el acaparamiento innecesario) para un mínimo de dos años y para la totalidad de las IPs, una por una. La asignación

se realizará en base a criterios de enrutamiento, por lo que además es requisito casi indispensable que la actividad del LIR sea un *carrier* o un gran ISP para poder justificar adquirir tantas IPs.

El RIPE mantiene en la actualidad sobre 14 millones de direcciones IP asignadas.

ARIN y APNIC

El ARIN (www.arin.net) se encarga de la asignación en América. No establece tasas, pero tiene un mínimo de asignación de una red de clase /20, y la obligación de uso inmediato de un 25% y un 50% a un año. El ARIN se encuentra dominado por EE.UU., viniendo a ser el organismo que ocupa tanto oficinas como personal del antiguo IANA. El funcionamiento es el mismo que para el RIPE: hay que solicitar el reconocimiento como LIR para poder trabajar con ellos directamente.

En Asia y Oceanía la situación es similar: se exige como mínimo la gestión de una red de /22 para ser miembro del APNIC (www.apnic.net), pero establece además precios bastante altos al trabajar por el principio de que quien quiere estas IPs bien puede pagarlas: las cuotas van desde los 1.250 dólares al año para un /22 (por lo que cada IP vendría a costar 1,22 $) hasta 40.000 dólares al año si se desea más de un /10. Es la más cara justamente por ser la que actualmente sufre una mayor expansión, al contener el APNIC el país de mayor crecimiento en todos los aspectos (demográficos y económicos): China.

LACNIC

Debido a que ARIN estaba tan claramente dominado por los EE.UU., en el año 2.000 los países latinoamericanos logran constituir un cuarto registrador regional, el LACNIC (www.lacnic.net), que ha asumido el control de sus propios rangos a partir de un subconjunto de los asignados al ARIN.

Había también otras razones para esta escisión: protegerse de la vorágine de EE.UU., donde en muy poco tiempo hubieran acabado con cualquier rango asignado al ARIN, dificultando por tanto el enrutamiento posteriormente si se hubiese tenido que comenzar a asignar una misma clase de direcciones en localizaciones tan distantes como Buenos Aires y Nueva York.

Esta nueva implementación de la pila de protocolos diseñada por el IETF (*Internet Engineering Task Force*, www.ietf.org) tiene el objetivo de satisfacer demandas de mayor seguridad y capacidad, esperándose que sea implantado justo cuando el actual espacio de direccionamiento proporcionado por los escasos 32 bits de la cabecera IP esté agotado.

La versión actual de este protocolo (IPv4) proporciona aproximadamente 4.600 millones de direcciones. Con una población mundial actual que ya supera dicha cantidad, el número de direcciones necesarias quedará pronto corto, sobretodo en el momento en que comiencen a despuntar en la red áreas como Asia. IPv6 proporciona miles de millones de direcciones adicionales al usar más bits para las direcciones (128 en total), tantas como que habrá casi como átomos en el planeta, y podrán lanzarse todo tipo de dispositivos con IPs propias[29].

La industria está pasando de la fase de pruebas de IPv6 a su utilización, pero hay una serie de resistencias a vencer. Entre estas resistencias la principal es el coste para las operadoras que supondrá el reemplazar todos los elementos hardware de la red en los que haga su presencia la capa 3 del protocolo TCP/IP, ya que el protocolo IP cambiará radicalmente tanto la forma de ser enrutado como la cabecera de los paquetes. Estamos por tanto ante un nuevo *efecto 2.000*.

La otra resistencia es el negocio que se ha creado en torno al espacio de direcciones IPv4: como si de un bien precioso se tratara, el disponer de direcciones de Internet válidas no deja de ser un activo de los que muchas empresas pueden vivir revendiendo a terceros. Dado que esto con IPv6 se acabaría y las operadoras no tendrían posibilidad de controlar la asignación de IPs (dada la escasez es difícil obtener la asignación directa de IPs), el alcanzar la implantación real de IPv6 es algo hoy por hoy utópico.

En este análisis del número de hosts presentes en Internet del *Internacional Software Consortium* (www.isc.org, organismo que aloja un servidor raíz de Internet, además de ser el responsable del desarrollo del software de servidor de nombres más popular en Internet, *bind*) hay datos referidos al número de IPs que contienen sitios Web. Es un estudio muy interesante porque ofrece datos de los servidores en activo conectados a Internet desde principio de 1.993 (momento en el que los hosts se contaban con la mano dada su reciente aparición) hasta la actualidad, con más de 170 millones de sitios Web activos. Esta gráfica tiene un crecimiento claramente exponencial, que de continuar así, obligará muy pronto a implantar IPv6 casi únicamente por la evidente falta de espacio de direccionamiento.

---

[29] Aunque es común oír que IPv6 tiene "más IPs que átomos el planeta" esto no es así si suponemos que el planeta tiene 10^{51}, lo que vendría a ser 2^{170}, mientras IPs tendríamos a lo sumo 2^{128}.

Hosts en uso en Internet

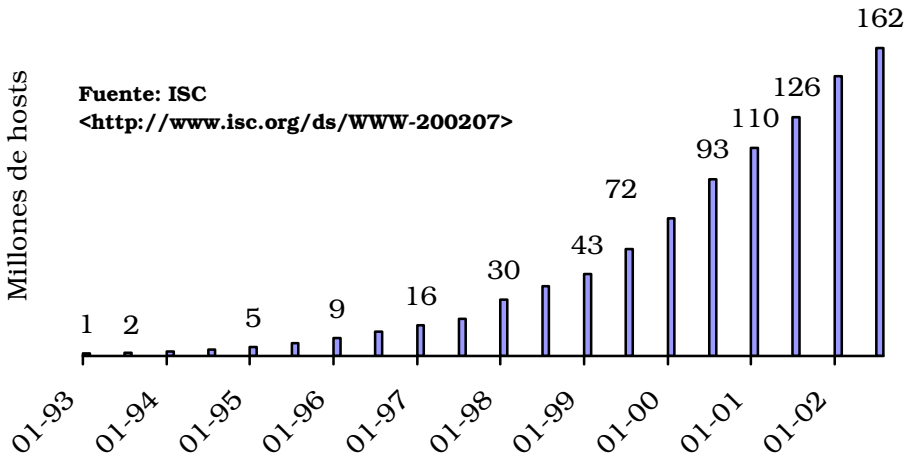


Ilustración 4-14: Número de IPs en uso en Internet

Fecha	Num. Hosts		
1-1993	1.313.000	1-1998	29.670.000
7-1993	1.776.000	7-1998	36.739.000
1-1994	2.217.000	1-1999	43.230.000
7-1994	3.212.000	7-1999	56.218.000
1-1995	4.852.000	1-2000	72.398.092
7-1995	6.642.000	7-2000	93.047.785
1-1996	9.472.000	1-2001	109.574.429
7-1996	12.881.000	7-2001	125.888.197
1-1997	16.146.000	7-2002	162.128.493
7-1997	19.540.000	1-2002	147.344.723

Dominios

Las direcciones en Internet basadas en su IP no son algo fácil de recordar, de ahí que en 1.983, como hemos visto anteriormente, se creara el sistema de dominios DNS, que permite asignar nombres más fáciles de recordar para el ser humano. Su diseño original apenas ha variado, consistiendo en base de datos distribuida en una jerarquía de servidores que se encargan de responder a las peticiones de los usuarios o de otros servidores que dependen de ellos en la jerarquía.

En sus comienzos la supervisión y control de las direcciones de las máquinas (y la asignación de nombres una vez fue creado el DNS), era competencia de la *Nacional Science Foundation* (**www.nsf.gov**), que la ejercía a través del SRI-NIC[30], organismo que seguía siendo una entidad pública y de recursos más o menos limitados.

A principios de 1.993 se privatizó esta tarea y el gobierno acabó por delegar la gestión de los registros y del *whois* (el directorio con la información de los registros) en una compañía privada llamada *Network Solutions* (**www.networksolutions.com**), para que fuera esta la que asumiera los retos que estaban suponiendo un crecimiento exponencial de los registros a partir de la expansión internacional de la Internet.

En realidad esta compañía privada sólo ostentó en régimen de monopolio la gestión de los gTLD (*generic Top Level Domain*): *com*, *net*, *org* y *edu*, mientras se mantenían bajo el control del gobierno de los EE.UU. tanto el *gov* (organismos gubernamentales de los EE.UU.) como el *mil* (militares de ese país), al mismo tiempo que ya al crearlos la IANA delegaba en los organismos NIC de cada país la gestión de los ccTLD (*country-code Top Level Domain*), como *es* o *fr*. Todos los ccTLD contienen únicamente dos caracteres, que provienen de la abreviación del nombre oficial del país en la lengua vernácula del mismo país con el alfabeto latino, según el estándar ISO3166 (que actualmente ya está en su versión 3, dado el crecimiento de países que en la pasada década de los 90 se ha producido tras la caída del bloque soviético). El ISO depende de las Naciones Unidas, y el mismo código está siendo utilizado por agencias de la ONU, organismos internacionales y el Comité Olímpico, por ejemplo. El documento con el listado del ISO3166 se puede consultar en la página Web del ISO (**www.iso.org**), pero resulta más interesante acudir a las asignaciones en vigor y ver qué organismo nacional de cada país (asignado por el gobierno de dicho país) ha asumido la gestión de su ccTLD y cómo lo gestiona, información que se puede consultar en **http://www.iana.org/cctld/cctld-whois.htm**. En realidad este organismo suele coincidir con el NIC (*Network Information Center*) de cada país, que se encarga de establecer otros protocolos o asignaciones, como podría ser la asignación de IPs, puertos, etc., aunque algunos países han privatizado la gestión de su ccTLD. Todo

[30] Funcionamiento del NICNAME/WHOIS, RFC 954 (Octubre 1.985): <http://www.rfc-editor.org/rfc/rfc954.txt>

este proceso para los ccTLD fue establecido en el RFC-1591 y ha permanecido inalterable con el paso de los años. Los únicos cambios que se producen son en la reasignación del organismo que gestiona un determinado ccTLD, como ha ocurrido con Afganistán, que estaba delegado a una persona en Londres[31].

Otra situación muy común es que un ccTLD tenga un conjunto definido de *SLDs* bajo los que ya se asignan comercialmente entradas a un tercer nivel, en el 3LD, por ejemplo como ocurre en Gran Bretaña, donde el ccTLD es *uk*, pero una persona no puede registrar suempresa.uk, sino que ha de colocarlo bajo **suempresa.co.uk**, o bien si se trata de una oficina gubernamental lo encontraría bajo *gov.uk*, etc.

Como estábamos comentado la privatización tuvo su punto final cuando en abril de 1.998 el contrato entre la NSF y *Network Solutions* caducó, momento que fue aprovechado para imprimir un poco más de liberalismo comercial en Internet y evitar dejar en manos de una única empresa todo el negocio. En octubre de 1.998 se creó la ICANN (**www.icann.org**), siglas de *Internet Corporation for Assigned Names and Numbers*, que desde su primer congreso celebrado en Singapur en 1.999 y con incluso delegados elegidos por votación electrónica y abierta, ha asumido la supervisión de los nombres de dominio, direcciones IP y los propios protocolos de Internet. Tantos aires de apertura resultan insólitos si tenemos en cuenta que los EE.UU. perdían así el control directo que hasta el momento venían ejerciendo, y efectivamente a partir del 2.001 los estamentos gubernamentales de este país han ido tumbando parte de las decisiones del ICANN y éste ha acabado por admitir únicamente consejeros delegados por los gobiernos y no votados por Internet, además de permitir delegar el control sobre el espacio de direccionamiento IPv4 en manos de los tres organismos regionales comentados anteriormente, con lo que el ARIN (controlado por los EE.UU.) recuperaba parte del poder.

El ICANN sólo supervisa: la gestión de los dominios pasó en su momento de manos de *Network Solutions* al CORE (*Council Of Registers*), **www.corenic.org**), organismo del que forman parte casi todos los registradores acreditados por el ICANN como tales. Lo que resulta más irónico es que la base de datos que afecta a los TLD *com*, *net*, *edu* y *org* continúa alojada en un servidor raíz que está en manos de *Verisign Global Registry Services*, que es lo mismo que decir *Network Solutions*, la compañía de la que se intentó sacar el control de estos TLD. Otra medida destinada a atacar el control de *Verisign* en los dominios se ha llevado a cabo este año 2.003, cuando el ICANN adjudicó el control del TLD *org* a *Public Interest Registry* (**www.pir.org**), produciéndose el reto técnico de hacer este traslado transparente a los usuarios, y que más adelante analizaremos en profundidad.

---

[31] *Request of Islamic Transitional Government of Afghanistan for Redelegation of the .af Top-Level Domain* (enero del 2.003, en inglés):

<http://www.iana.org/reports/af-report-08jan03.htm>

En realidad lo que hemos logrado tras la pérdida del monopolio de *Network Solutions* es competencia: ahora la creación de nuevos TLD queda en manos de un organismo, el ICANN, el cual delega la gestión de cualquier base de datos de los TLD existentes o nuevos a determinada compañía que albergue un servidor raíz (el resto replicarán de él la información) por un plazo de años determinado en un concurso público. Esta compañía (*Network Solutions* en el caso de los *com*, *net* y *edu*) está obligada a facilitar la operatividad de cualquier registrador convenientemente registrado en el ICANN (el registro es objetivo y basado en el cumplimiento de unos requisitos económicos de solvencia y otros técnicos de disponibilidad, servicio y operatibilidad con el resto de registradores). Cualquiera puede, cumpliendo los requisitos exigidos, convertirse en un registrador y acceder al *Shared Registration System* (SRS)[32]. Pero la competencia también ha generado algunos problemas.

La evolución del registro de dominios

El número de registradores a los que el ICANN deja participar en este negocio se ha disparado de manera astronómica desde que en 1.998 dejáramos de tener únicamente a *Network Solutions* como opción. Los precios han caído debido a la competencia, aunque posiblemente ya hemos alcanzado el suelo y ahora vayamos a ver una guerra de desgaste, con un resultado de concentración en el sector. Pero la pregunta sigue siendo la misma: ¿ha sido bueno o malo?

Desde el punto de vista del cliente, el cambio ha sido bueno: el precio ha caído desde los 50 ó 60 $ de la época de *Network Solutions* a menos de 10 $ en la actualidad. Técnicamente es más discutible: lo que antes era una base de datos única en manos de una única empresa que hacía sobre ella modificaciones, ahora disponemos de 192 empresas sólo para los gTLD (el listado completo de las empresas se puede obtener en **http://www.icann.org/registrars/accredited-list.html**). Lo peor en este caso ha sido que esta disgregación se ha visto acompañada de una falta de control: algunas de estas empresas mantienen registros incorrectos en sus bases de datos (incumpliendo en parte los requisitos que aceptó para ser registrador), o incluso han llevado a disputas en la asignación de nombres que han llevado a que hoy día en el propio *Council of Registrars* se hayan articulado tribunales de arbitraje entre los registradores.

Network Solutions como tal fue absorbida por otra compañía norteamericana en junio del 2.000, *Verisign*. Aunque ya no controla el registro de los dominios, sigue siendo un peso pesado y como tal es aún el primer registrador a nivel de los dominios no territoriales. En resumen: para calibrar si se ha ganado con el cambio o no, vamos a tratar de ver de cuánto es el negocio de que hablamos y sobretodo, su

---

[32] ICANN: documentos necesarios para la acreditación como registrador y pasos a realizar:

<http://www.icann.org/registrars/ra-agreement-17may01.htm>

<http://www.icann.org/registrars/accreditation-documents.htm>

crecimiento en los últimos años: si el mercado del registro de dominios supone una cifra considerable habremos ganado al disgregarlo, y sobretodo, si ha crecido más que los dominios territoriales.

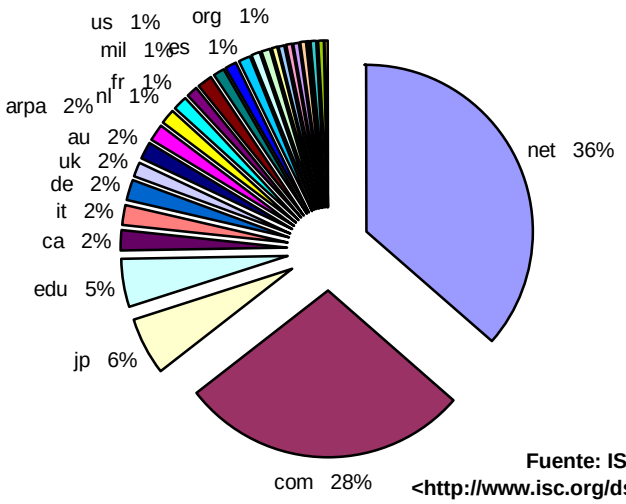
Algunos de los dominios territoriales han intentado en los últimos años copiar este modelo de permitir la competencia en el registro: *us*, *jp*, *ge*, *uk*, *cn*, *tv*... Otros por el contrario continúan utilizando el organismo único centralizado. El tiempo está dando la razón a los que optaron por disgregar el servicio, esto es lo que puede deducirse si vemos cuales son los TLD que están dando mejores resultados en cuanto a dominios en uso se refiere.

TLD	Dominios activos
net	56.646.014
com	43.814.657
jp	8.713.920
edu	7.381.306
ca	3.129.884
it	2.958.899
de	2.923.327

us	1.874.513
tw	1.814.090
es	1.682.434
org	1.238.739
se	1.187.942
mx	1.004.637
fi	986.285

uk	2.508.151
au	2.496.683
arpa	2.420.976
nl	2.150.379
fr	2.052.770
br	1.988.321
mil	1.918.954

dk	872.328
be	832.853
pl	731.371
at	720.587
gov	700.107
ch	667.509
no	634.098



Fuente: ISC (Internet Software Consortium)
<http://www.isc.org/ds/WWW-200207/dist-bynum.html>

Ilustración 4-15 Sitios Web activos en Internet

Aunque estas cifras no reflejan el número de hosts realmente registrados (dónde el líder es el TLD *com*) se ha estimado más conveniente el descartar aquellos dominios no activos (que es lo que refleja esta gráfica), es decir, aquello que no se usa o está inaccesible al público en general. Hay que tener en cuenta que no es lo mismo el número de hosts activos que los dominios, ya que un mismo servidor puede ofrecer a través de una IP múltiples dominios.

El número de dominios en servicio se puede descubrir gracias a que el *Internet Software Consortium* ha tenido a bien medirlo cada seis meses desde 1.993[33]. La forma de medirlo concreta es bastante simple: sólo contabiliza aquellos sitios registrados por cada TLD para los que encuentra alguna IP accesible (con lo que no considera dominios cuya situación técnica sea de reserva, caducados o estén mal configurados y no funcionen). Estos datos reflejaban a fecha de Julio del 2.002 que dentro del gTLD más popular (*com*) el número de dominios ocupados pero no en uso era del 54,5% (de los 96.316.392 había fuera de servicio 52.501.735). Se trata de pura y simple ocupación para especular o revender, que ha llevado a que ya en 1.999 de las 25.000 palabras de diccionario más populares sólo 1760 de ellas estuvieran libres (y a ese ritmo, ninguna en la actualidad). Es especialmente interesante ponerse a buscar nombres de poblaciones y luego comprobar quién las tiene registradas.

Como resultado, el negocio de registro ha movido y está moviendo cifras multimillonarias: A poco que se paguen 10 $ dólares por cada dominio, para un mercado estimado de 160 millones de dominios para el 2.003, estaríamos hablando de más de 1.000 millones de dólares.

Pero a partir de abril de 2.002 las cifras de negocio han dejado de crecer tanto como crecían hasta ahora: el registro bajo *com*, *net* u *org* (de acceso libre, a diferencia de *edu*, donde hay que demostrar alguna relación con temas educativos) se ha estancado bien por agotamiento de las combinaciones interesantes, bien porque la creación de tribunales ante los que dirimir disputas como la *ciberocupación* de sitios ajenos.

Ha sido el ICANN el que ha dispuesto una política unificada de resolución de disputas que han de aplicar de manera obligada todos los registradores, la *Uniform Domain Name Dispute Resolution Policy* (UDRP)[34], con cuatro organismos actualmente reconocidos como árbitros, entre ellos el WIPO (www.wipo.int). Todo esto en contraste con lo que pasaba tiempo atrás, cuando la única solución era iniciar una demanda judicial en EE.UU., territorio donde radicaba la compañía que gestionaba este asunto.

Aparte de la resolución de disputas también se ha creado sobre los dominios gTLD un periodo de gracia o redención para tratar de hacer reversible el borrado de un dominio por error, ya que en ese caso el dominio pasaba de nuevo a estar disponible y podía perderlo el cliente original.

Ahora ya no: los dominios al darse de baja pasan un engorroso proceso que retarda al menos 50 días el abuso sobre los mismos. Los dominios registrados poseen un estado (*status*) visible al consultarlos,

---

[33] *Internet Domain Survey*, realizado por el *Internet Software Consortium:* <http://www.isc.org/ds/WWW-200301/dist-bynum.html>

[34] Documentación sobre la UDRP en el ICANN: <http://www.icann.org/udrp/udrp.htm>

que por lo normal es ACTIVE. Si el registrador lo ha bloqueado, por impago o cualquier causa, el registro es modificado por éste a ONHOLD (momento que se aprovecha para desactivarle la Web únicamente, aunque lo demás suele permanecer activo). El registrador tiene hasta 20 días para transferir el registro de nuevo a ACTIVE o bien pasarlo a REDEMPTIONPERIOD. En este estado queda 30 días, durante los cuales únicamente el propietario puede recuperarlo, aceptando únicamente el sistema dejar el registro en el estado anterior pasando por un periodo transitorio en estado PENDINGRESTORE. Una vez transcurrido este periodo de gracia, el dominio pasa a PENDINGDELETE y en menos de 5 días es borrado definitivamente y pasa a estar de nuevo disponible para registro[35].

El sistema resulta muy caro: durante el periodo de gracia instaurado, recuperar un dominio cuesta siempre más de 200 dólares. Y esto además no evita otros abusos muy comunes: anular a sus clientes registros de dominios por los que el registrador esperaba cobrar mejores tarifas a otros clientes (beneficiando por tanto al mejor postor en lugar del primero en solicitar el registro, tal como el reglamento que dicho registrador firmó con el ICANN), quedarse el registrador con el dominio una vez vencido (lo que no es nada ético si tenemos en cuenta que ya hace tiempo que se lucha contra la *ciberocupación* para que ahora sean los registradores los que la practiquen), etc.

El ICANN espera ampliar la transparencia del sistema para evitar estas situaciones, haciendo que los registradores utilicen un sistema unificado de reservas en los nombres de dominio, pero resulta difícil evitarlo con tanto registrador autorizado.

Crecen los gTLD

Ante el agotamiento y el estancamiento de la demanda de nuevos dominios, los pasos de los registradores han ido encaminados a mantener vivo al mercado, como ahora veremos, fomentando el uso de otros TLD: bien presionando hasta lograr la creación de nuevos gTLD, bien logrando que los ccTLD de países minúsculos pero con un alto potencial (por ejemplo el *tv* de que dispone el archipiélago de *Tuvalu*) sean accesibles para miembros del *CORE*, o cedan directamente el control a determinada compañía.

El primer cambio a nivel de gTLD fue crear las terminaciones *info*, *name* y *biz*. Posteriormente, este año 2.003 se han incorporado la posibilidad de registrar otras más: *aero*, *coop*, *museum* y *pro*, aunque esta última aún no ha sido asignada a ninguna compañía para su gestión. Aunque parte de estos gTLD no son para registro libre (por ejemplo el *aero* sólo está disponible para compañías del sector aeronáutico, tal y como se puede consultar en **www.nic.aero**), lo cierto

[35] Explicación sobre la implantación del periodo de gracia, en la reunión de Febrero del comité pertinente del ICANN:

<http://www.icann.org/minutes/report-vgrs-rgp-consolidate-23feb03.htm#AppendixCtocomandnetRegistryAgreements-7A>

es que tanto movimiento nos obliga a resumir en forma de tabla la actual situación y clasificación de los TLD, para luego hablar de alguna de las amenazas y problemas aparecidos en los últimos años:

Genéricos (gTLD)	*com, org, edu (los tres de 1.984), net (añadido en 1.985) y los nuevos: aero, coop, museum, pro, info, name, biz*
Internacionales (*iTLD*)	*int (uso casi testimonial)*
Especiales (*sTLD*)	*arpa*
Uso exclusivo de EE.UU.	*mil, gov*
Regionales (ccTLD)	*es, uk, us, fr, de...*

Los dominios multilenguaje

Pero para los registradores también resultaba interesante que los usuarios pudieran usar caracteres no latinos del alfabeto inglés en las direcciones Web. Esta barbaridad es conocida como IDN (*Internationalized Domain Names*), y fue iniciada por *Verisign*[36] y permitida en su momento por el ICANN, pero por suerte su éxito ha sido bajo (achacable con toda probabilidad a los problemas de implantación que desde su aparición en noviembre de 2.000 ha producido), pese a que ya hace seis años que se venía presionando en este sentido.

El problema en este caso viene por el hecho de que estos dominios contienen caracteres que no va a ser posible que cualquier persona los escriba en cualquier teclado. Estamos hablando de caracteres *Unicode*, como la letra eñe, o el acento en una vocal, y sobretodo los alfabetos orientales (chino, japonés).

El sistema requiere además que el navegador sea compatible con esta sintaxis (por otro lado el único servicio en el que parece tener sentido usar dominios multilenguaje), con lo que los clientes han de actualizarse o tampoco podrán acceder. Afortunadamente, no ocurre así con el servidor DNS, ya que para la implantación se han decantado por un sistema de traducción de direcciones en las que una URL como **peñafiel.es** no es realizada directamente al servidor DNS de esperar, sino que se traduce por **bq--abygl4lbmzuwk3a.mltbd.net** (siempre un subdominio de mltbd.net, apareciendo codificada la petición original en dicho subdominio). Ni que decir tiene que los únicos navegadores que soportan de forma nativa esta característica son los de Microsoft, y sólo las versiones más recientes.

Además, la única forma que tendríamos de acceder fuera de nuestro país a un dominio multilenguaje cuyo carácter no está presente en

[36] Definición y preguntas más frecuentes sobre los *Internationalized Domain Names*:

<www.verisign-grs.com/idn>

nuestro teclado sería o bien registrar las combinaciones sin estos caracteres e informar de ello al usuario, o introducir en los navegadores la traducción que realmente ellos utilizan internamente para lograr la dirección IP (es decir, el bq--abygl4lbmzuwk3a.mltbd.net antes comentado). Resulta por tanto evidente que no interesa en ningún caso usar dominios multilenguaje, y eso que ni siquiera hemos llegado a comentar los precios de los mismos.

Las keywords

El primer problema que plantean ciertas mejoras practicadas por compañías privadas al margen de cualquier organismo es que no buscan crear un estándar, sino única y exclusivamente obtener un beneficio.

Esto es lo que ha ocurrido con el tema de las *Keywords* o *CommonNames* de Internet, otra acción comercial que ha fracasado. Básicamente la idea era ofrecer nombres cortos y fáciles de recordar a usar en lugar de las complejas URL, sin siquiera tener que recordar si la página acababa en *com*, *net* u *org*. El problema es que esta característica requiere que el navegador se comporte de manera diferente a como debería: al no detectar caracteres de puntuación en el nombre introducido ha de evitar realizar una petición de resolución DNS y solicitar a un servidor prefijado que le suministre una URL relacionada con la palabra o palabras introducidas, comportamiento que el navegador adquiere sólo con una actualización del mismo o bien porque esta característica está incluida como fruto de un acuerdo entre el fabricante del software y la compañía que cobra por registrar estas palabras clave.

Aparte de que los precios son altos, el problema es que en la práctica los navegadores que soportan esta característica son únicamente los de *Microsoft*, para el caso de *CommonName Inc* (www.commonname.com). *Netscape* usa sus propias *keywords* en sus navegadores, incompatibles con las primeras (otro problema añadido).

Lo que en cambio ha logrado asentarse con bastante mejor éxito ha sido el uso de estas palabras clave en entornos cautivos como son los usuarios de AOL, cuyo navegador (especial para sus clientes) incorpora también esta característica.

El sistema DNS a nivel técnico y el crecimiento de tráfico

El funcionamiento del DNS era y es jerárquico: tenemos ante nosotros un sistema de base de datos distribuida montado en 1.984 y basado en ideas de Jon Postel que se ha tenido que ir adaptando sobre la marcha, hasta ser convertido en parte troncal de la propia red.

En realidad los nombres son necesarios únicamente porque no podemos retener las direcciones IP de aquello a lo que deseamos acceder, ya que un servidor DNS está únicamente para traducir nombres en IPs, aquello que es realmente útil a la pila de protocolos IPv4 usada en Internet. Antes de 1.984 la manera de usar nombres y

que aún se puede utilizar tanto en Windows como en *Unix* (además de que tiene mayor prioridad) era introducir las equivalencias equipo por equipo en cada máquina en un fichero llamado *hosts*.

Un servidor DNS almacena y centraliza estas equivalencias que estaban en el fichero nombrado, pero de manera jerárquica, cogiendo de derecha a izquierda un nombre compuesto por caracteres latinos ingleses y el carácter de puntuación, siendo el carácter de puntuación el delimitador de cada uno de los niveles que conforman un nombre de dominio. Pero los ordenadores que están en Internet no usan todos ellos el mismo servidor DNS, sino que cada uno que lo considera crea unos y son compartidos por sus usuarios, siendo únicamente estos servidores los que a su vez cuando desconozcan algo tendrán que acudir al siguiente en la jerarquía de dominios para poder resolver una entrada.

Por TLD se entiende la palabra que no contiene punto que haya más a la derecha de un nombre (por ejemplo en **www.rediris.es** sería *es*), y la delegación de uno de estos TLD sobre un servidor implica que ese servidor se encargará de responder en el sistema DNS y a nivel global de cualquier entrada que acabe en dicho TLD, así como de gestionar las altas, modificaciones y bajas del mismo. Si seguimos leyendo el nombre del dominio, el siguiente campo separado por puntos que sigue a la izquierda del TLD sería el SLD (*Second Level Domain*), el siguiente el 3LD (*Third Level Domain*), etc., con lo que tendríamos siempre entradas del tipo <3LD>.<SLD>.<TLD>. Todo el sistema está explicado en los RFC 920 y RFC 921[37].

En la raíz de todo este sistema existen un número limitado de servidores de nombres que se encuentran en lo alto de la jerarquía, conocidos como servidores raíz. Estos trece servidores se pueden listar en **www.root-servers.net**, y como se podrá advertir al analizar su localización, se encuentran casi todos ellos en EE.UU. Los nombres que se han asignado a cada uno de los servidores son letras del alfabeto: **d.root-server.net**, **m.root-server.net**, etc.

El problema fundamental que acecha a este diseño jerárquico es la vulnerabilidad del sistema a la caída de esas trece máquinas, caída debida tanto a ataques intencionados como al incremento de tráfico. De hecho que hoy haya trece no quiere decir que los hubiera en el diseño original, ya que se han ido añadiendo más conforme se veía el crecimiento de la red desde los ocho originales, habiéndose ya creado tres de ellos fuera de los EE.UU. (en concreto son *I*, *K* y *M*, que están en Estocolmo, Londres y Tokio). Cada servidor es administrado por un organismo distinto, y en sus orígenes cuando había ocho, estos eran cada uno de los gestores de los TLD iniciales (el Departamento de Defensa de los EE.UU. como gestor del mil, las universidades a través

---

[37] Postel, Jon: *Domain Name System Implementation Schedule* (RFC 920) y *Domain Requirements* (RFC 921). 1.984, en inglés:

<http://www.rfc-editor.org/rfc/rfc920.txt>

< http://www.rfc-editor.org/rfc/rfc921.txt>

de la Universidad de *Maryland* por el *edu*...) y la propia *Network Solutions* como compañía que pasó a gestionar algunos de estos TLD (y que de hecho sigue siendo **a.root-server.net**). Un servidor raíz sirve hoy día unas 15.000 peticiones por segundo de media.

Uno de los casos más famosos de ataques contra estos servidores pasó el 21 de Octubre del 2.002, cuando durante un hora y quince minutos un se produjo un ataque de denegación de servicio distribuido simultáneo contra los trece servidores raíz. Este ataque es más conocido por sus siglas en inglés: *DDoS* (*Distributed denial of service*), y consiste en saturar a cada servidor de tráfico desde diferentes orígenes (para dificultar la identificación del atacante y ralentizar la respuesta de los administradores de red) con tramas ICMP (simples *pings* aunque tamaño superior, TCP SYN (paquetes de negociación de la conexión en el tráfico TCP, conexión que establecían y luego no usan, obligando al servidor a consumir recursos en la gestión de una conexión ficticia) y tráfico UDP. En resumen, durante esa hora y media estos servidores soportaron conjuntamente un tráfico de 900 *Mbits/sec* o 1.8 *Mpkts/sec* (millones de paquete por segundo), que causaron que cualquier intento de acceso normal a los servidores no tuviera respuesta[38]. Pese a la gravedad aparente del ataque (total parálisis durante horas), su impacto no llegó hasta ese extremo: el diseño jerárquico de la DNS hace que de las peticiones de resolución DNS de los usuarios la mayoría se resuelvan en los niveles inferiores, unido a que los servidores además cachean esas respuestas.

Pero esta situación descubrió un peligro latente en una Internet ya demasiado fundamental como para pararse durante horas, sobretodo si valoramos el hecho de que los *hackers* atacantes no actuaban con intención dolosa, sino meramente para demostrar el punto débil. De ahí que ahora comentaremos una serie de soluciones que se han ido implantando ya desde antes de ese ataque y que contribuirían a fortalecer el sistema. Todos estos retos han sido discutidos en el RSSAC (*Root Server Advisory Committee*), comité del ICANN encargado del asunto.

Los servidores raíz espejo

En lugar de crecer en número de servidores se ha estimado más oportuno comenzar a trabajar con servidores espejo. La razón es que un número mayor de servidores raíz dificulta la continua transferencia de información entre ellos (hay que tener en cuenta que como base de datos distribuida que es, los servidores raíz tienen que mostrar todos ellos la misma información). Un servidor espejo únicamente conecta con aquél de quien hace espejo, con lo que el coste de transferencia de información se restringe a sólo uno de los servidores raíz.

---

[38] Vixie, Paul (2.002): *The attack against root servers of 21-Oct-2.002*: <http://f.root-servers.org/october21.txt>

Este proyecto ha sido desarrollado por el *Internet Software Consortium* en el servidor raíz F (f.root-servers.net), que ellos gestionan. En la actualidad hay disponibles ya cinco espejos del servidor principal alojado en Palo Alto (California): tres están dentro de los Estados Unidos, y los otros en el extranjero, uno en *Hong Kong* y el otro en España.

Estos espejos están interconectados directamente con el servidor principal, y en caso de caída pueden mantener operativa la respuesta de dicho servidor en las áreas por ellos cubiertas. En realidad para que este sistema funcione los ISP tienen que colaborar de manera activa modificando el enrutamiento, ya que estos espejos tienen también la misma IP que el servidor raíz. Se prefirió esta alternativa a usar varias IPs por el hecho de que cualquier modificación en la asignación de las IPs sobre los servidores raíz supone cambios en las tablas estáticas que dentro de cada servidor DNS del mundo (que deben ser varios miles) y además, todos los registros asociados tienen unos tiempos de caducidad desmesuradamente altos (3.600.000 segundos), con lo que además la propagación total de la respuesta tardaría hasta casi 42 días en producirse. Aunque esto de cambiar no es tan poco habitual como puede parecer: por ejemplo en Noviembre del 2.002 fue actualizado por un cambio en la IP de j.root-servers.net.

En concreto el espejo español del nodo raíz que acabamos de comentar ha sido creado con la intención de que absorba el tráfico de España, Francia, Italia, Portugal y norte de África. Básicamente los 31 operadores presentes en el nodo neutro español *Espanix* resuelven gran parte del tráfico en este nuevo espejo creado.

El proyecto AS112[39]

Otro problema detectado tras un análisis estadístico es que los servidores raíz suelen responder a peticiones de resolución inversa con bastante frecuencia (es decir, a peticiones para obtener el nombre de una determinada IP), cosa que no tendría mucha importancia porque muchas de esas peticiones de resolución corresponden a rangos de redes privadas o reservadas, como son las redes 10.0.0.0/8, 172.16.0.0/12, 169.254.0.0/16 y la 192.168.0.0/16. Estas cuatro redes son las definidas en el RFC 1918[40], y por tanto muchas veces a estas resoluciones inversas sin sentido se les da el nombre de este RFC. La razón es evidente: estas redes no poseen de por si resolución directa o inversa pública, siendo responsabilidad de cada administrador de red el configurar una DNS para esas zonas. Si éste no lo hace, los hosts que haya dentro pueden llegar a preguntar al DNS cuál es el nombre del host, y esa petición irá escalando en la jerarquía hasta llegar a algún

[39] Página oficial del proyecto AS112: <www.as112.net>

[40] Y. Rekhter y otros (1.996): *Address Allocation for Private Internets*, RFC 1918: <http://www.faqs.org/rfcs/rfc1918.html>

servidor raíz, ya que ninguno de los servidores DNS (ni tan siquiera el raíz) serán capaces de dar una respuesta autoritativa para el mismo). El resultado es que el usuario percibe retardos (sus aplicaciones esperan inútilmente un nombre para una IP que es privada), la red tiene más tráfico del debido (aunque no mucho) y sobretodo, los servidores raíz se sobrecargan con peticiones innecesarias.

El proyecto AS112 busca que existan servidores separados para estas resoluciones inversas que además den respuesta autoritativa. Personalmente vería más conveniente que estas peticiones directamente no se realizaran, con lo que se eliminaría parte del tráfico DNS redundante. Además, las zonas comentadas no van a cambiar nunca de nombre (dado que no hay servicios en ellas) y el proyecto sigue necesitando que todos los servidores DNS modifiquen el software para que estas zonas las resuelvan en diferente localización a la que vienen utilizando.

El sistema resuelve dos de las pegas planteadas por las IPs de los rangos privados, como son la saturación de los servidores raíz (resolvemos este rango en diferente servidor), y el tiempo de retardo provocado al usuario (ya que la respuesta es autoritativa y por tanto más rápida, ya que en caso de no encontrarse en un determinado servidor éste a su vez la escalaría hacia arriba esperando encontrarla en el siguiente, cosa que no se produciría finalmente y acabaría llevándonos hasta un servidor raíz).

Para conocer cuál servidor podríamos usar para estas zonas basta preguntar al servidor **prisoner.iana.org** por las correspondientes IPs asociadas a **hostname.iana.org**, y en la actualidad de esa respuesta se deduce que existen tres servidores: 192.175.48.1, 192.175.48.6 y192.175.48.42.

El sTLD .arpa[41]

Hasta ahora no habíamos hablado de este *sTLD* (la "s" del nombre es de *special*), creado temporalmente en su momento creyéndose que sería útil nada más durante la transición de *ARPAnet* a Internet y que finalmente continúa en servicio.

El nombre viene de las siglas que conforman *Address and Routing Parameter Area*. Estamos ante un TLD usado para tareas de infraestructura, en concreto este TLD posee de manera única seis SLD, que son:

- *e164.arpa*, *uri.arpa* y *urn.arpa*, usado para el protocolo E164 y otros protocolos.
- *in-addr.arpa*, para la resolución inversa de IPv4.
- *ip6.arpa*, resolución inversa de IPv6.

---

[41] Huston, G (2.001)*: Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain*, en el RFC 3172:

<http://www.rfc-editor.org/rfc/rfc3172.txt>

La única verdaderamente útil es la resolución inversa: así como podemos obtener a partir de un nombre su equivalencia en IPv4, podemos hacer el paso contrario, sabiendo que para ello lo que realmente va a buscar el servidor DNS es el puntero de tipo PTR correspondiente a la entrada WW dentro de la zona X.Y.Z.in-addr.arpa, si el host era ZZ.YY.XX.WW. El único problema planteado por este sistema es la incapacidad para gestionar de manera individualizada la resolución inversa de una IP concreta o de todo aquello que no corresponda de manera inequívoca con una clase C, B o A, teniendo que recurrir a la delegación individual IP por IP.

La jerarquía presente dentro de los servidores raíz y la incorporación de nuevos TLD

Aunque realmente se ha dado a entender que los trece servidores raíz actúan como *primus inter pares*, la realidad no es esta sino otra muy distinta: la información entre ellos compartida es la misma, pero los cambios y el control de esa información se realiza desde un único punto dentro de la jerarquía.

En concreto tenemos siempre tres servidores para cada gTLD, que son sobre los que se refleja una nueva alta o una baja, y desde ahí se alcanzan a los restantes diez.

Tabla 4-1: Servidores raíz, localización y TLD que gestionan. El TLD *org* está alojado por ahora en los mismos servidores que *com*, pero al estar fuera ya de *Verisign* no se descarta que en el futuro pase a otros.

Servidor	Organismo que lo gestiona	Localización	TLD que controla	URL
a	Verisign	Herndon	com	http://www.internic.org
b	ISI	Marina del Rey	edu	http://www.isi.edu
c	PSInet	Herndon,	com	http://www.psi.net
d	UMD	College Park	edu	http://www.umd.edu
e	NASA	USA	mil, gov	http://www.nasa.gov
f	ISC	Palo Alto (USA)	com	http://www.isc.org
g	DISA	Vienna	mil, gov	http://nic.mil
h	ARL	Aberdeen	mil, gov	http://www.arl.mil
i	NORDUnet	Estocolmo (Suecia)	int	http://www.nordu.net
j	Verisign	Herndon		http://www.iana.org
k	RIPE	Londres (Reino Unido)	int	http://www.ripe.net
l	ICAAN	Colorado		http://www.iana.org
m	WIDE	Tokio (Japón)	int	http://www.wide.ad.jp

La gestión de estos servidores ha llevado bastantes discusiones, y existe abundante documentación al respecto, como por ejemplo los RFC 2010 y 2870, relativos a la operatividad de la comunicación entre ellos y requisitos mínimos[42].

[42] Vixie, Paul, Plfzak y otros (1.996): *Operational Criteria for Root Name Servers y Root Name Server Operational Requirements*:

<http://www.rfc-editor.org/rfc/rfc2870.txt> <http://www.rfc-editor.org/rfc/rfc2010.txt>

El traslado de *org*

Recientemente se han producido otro gran reto en el sistema DNS a nivel global: trasladar la gestión de un determinado gTLD de unos servidores a otros. Justamente ha sido con el gTLD *org* por tres razones: la primera era porque el contrato con *Verisign* (que actualmente lo gestionaba) vencía este mes de diciembre del 2.002, la segunda que se buscaba romper el actual dominio de *Verisign* en este campo, dado que justamente eran los gTLD que ella administra los que mayor nivel de ocupación registran y la tercera porque de entre los TLD existentes, era el que menos entradas tiene (sobre 2 millones frente a 26 millones del resto de combinaciones) y esto permitiría usarlo de cobaya:

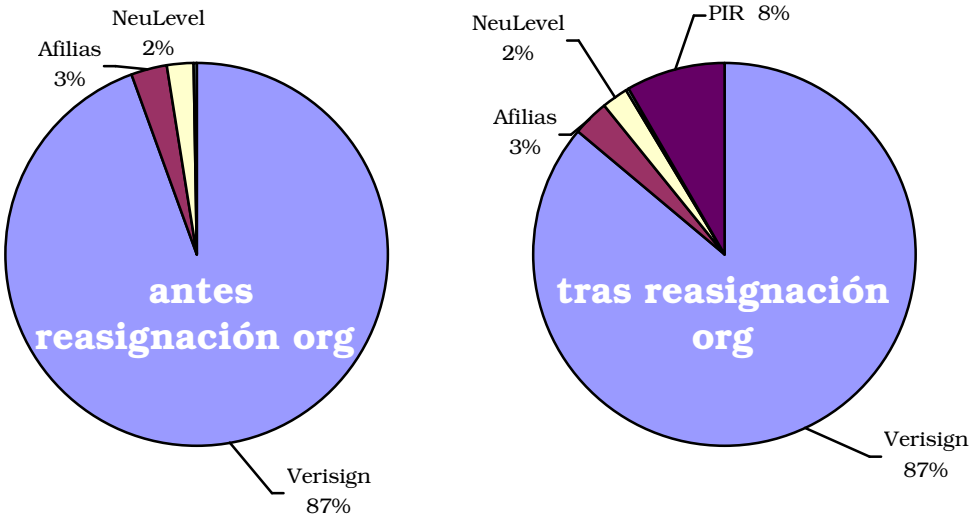


Ilustración 4-16: Reparto del mercado de los dominios

(Fuente: ICANN, http://www.iana.org/reports/org-report-09dec02.htm**)**

Se ha buscado por tanto lograr un cierto reequilibrio en el reparto, que ha sido asignado al *Public Interest Registry* (PIR, **www.pir.org**), el cual carece de fines económicos por ser una fundación sin ánimo de lucro. El PIR ha sido creado por *Internet Society* (ISOC, **www.isoc.org**), y se ha comprometido también a analizar con mayor rigor las peticiones de registro recibidas, por tratarse de un TLD reservado a cualquier organización tipo sindicato, ONG, etc. (cosa que hasta no se cumplía demasiado bien).

El reto venía más que nada por el hecho de que la base de datos debía trasladarse sin afectar a su función última, que es responder de manera continua en Internet. Para ello se han sacrificado algunas cosas por el camino, como los plazos (hasta el día 25 de enero no se hizo efectivo el traspaso de poderes, con lo que *Verisign* estuvo gestionándolo un mes más del esperado), pero pasado el periodo de transición, el ICANN se encuentra satisfecho por los resultados obtenidos.

4.2 Normativa y marco legal de Internet

Sin llegar a extremismos como los demostrados por el magnate de los periódicos *Robert Murdoch* (que afirmó durante una charla en Singapur, en enero de 1.999, que Internet destruiría más riqueza de la que crearía, dado que permite eliminar intermediarios tanto en la información como en el comercio) lo cierto es que, Internet al igual que el ferrocarril, alterará tanto el comercio mundial y la legislación tanto tributaria como penal, como ningún otro invento o hecho lo ha hecho hasta ahora en la Humanidad en tan poco tiempo (en menos de diez años, y todo ello eliminando fronteras porque son justamente las fronteras lo que Internet no respeta).

Básicamente, Internet no está cambiando las formas de hacer negocios en el mundo ni nuestro comportamiento o cultura, sino que como herramienta humana es parte ejecutora de un cambio que viene impuesto por la Globalización. Globalización económica de la economía de mercado sin protecciones sociales ni fronteras arancelarias, globalización cultural de la cultura occidental (mayoritaria en la red la anglosajona, especialmente si tenemos en cuenta que la mayoría de las páginas disponibles en la red son en inglés). En definitiva, Internet no es el culpable de estos cambios, sino una manifestación más de ellos.

La imposibilidad de controlarlo (pese a los intentos de países como China, donde todo el tráfico pasa por servidores bajo control del Gobierno chino) hará que tarde o temprano se den iniciativas y acuerdos a nivel internacional que regulen las lagunas que Internet ha generado en el comercio (ausencia de pago de impuestos y garantías al consumidor), protección de datos (cuando estos viajan entre países distintos), libertad de expresión, etc.

4.2.1 Legislación tributaria en Internet [43]

El comercio electrónico surge de la fusión entre la actividad económica e Internet, pero desde sus inicios ha estado al margen de la legislación tanto tributaria como comercial en su sentido más general por la falta de fronteras por un lado y la propia virtualidad del trato (no hay escritos firmados que permitan validar la existencia de esta relación comercial, tal como los habría en un trato más personal).

Han sido pues varios los retos que han tenido que superarse para regular una actividad económica como cualquier otra, que se produce en el marco de un nuevo medio:

---

[43] Fernández, Eleuterio. *Comercio electrónico y sujetos del e-commerce*: <http://www.alfa-redi.org/upload/revista/102501--7-3-COMERCIO1.doc>

Hyurley, Brian y Peter Birkwod (2.000). *Como hacer negocios en Internet*.

- Falta de legislación específica e imposibilidad de aplicar en muchas ocasiones la existente sin que sea adaptada.
- Ambigüedad sobre quién ha de regular las transacciones internacionales que se producen cuando comprador y vendedor están en diferente país, si la legislación del país de origen o la del país de destino.
- Resistencias por parte de consumidores y usuarios ante un medio que infunde respeto por el alarmismo provocado por la falta de seguridad en los pagos e identificación de la otra parte.
- Costes: aunque pueda parecer que Internet disminuye los costes al evitarse intermediarios y los costes de una tienda con presencia física (imagen, personal e inmueble), esto sólo ocurre con empresas con economías de escala muy grandes. A otros niveles, el ahorro en capítulos de personal queda equilibrado en muchas ocasiones por los gastos de transporte (mayores) y sobretodo, por los costes de la morosidad en tarjetas de crédito y las comisiones que de los pagos hay que descontar. Además Internet es un medio en el que la igualdad es total: todas las tiendas están igual de lejos, y ante eso el cliente únicamente mira el coste, entrándose en una guerra de precios o gastando mucho más en publicidad para atraer a los clientes.
- Fraude: el 1,2 % de las compras online son fraudulentas, perjudicando los resultados de los comerciantes tanto por los gastos de gestión extras (pedidos devueltos, denuncias, etc.), como porque la mayoría de ocasiones es el comerciante quien acaba pagando las operaciones con esas compras fraudulentas. Es también bastante habitual que una operación ya finalizada le sea retrocedida hasta dos meses más tarde por el cliente alegando cualquier excusa, mientras que el producto ya ha sido enviado y está en casa del consumidor.

En realidad, al nivel del derecho internacional, se ha dado en definir por comercio electrónico a aquella transacción que al menos cumpla tres condiciones:

- Existencia de una sociedad que sea destinatario del derecho como ordenamiento, lo que significa que haya unos sujetos que sean el medio a través de los cuáles se realiza esta actividad económica.
- Existencia de una regulación de las relaciones que sea el fin del derecho como ordenamiento (es decir: que haya una normativa a las que las partes puedan acogerse como base de la transacción).

- Existencia de una organización que sea el medio o instrumento del derecho como ordenamiento (que se pueda recurrir a estamentos arbitrales o judiciales en caso de conflicto).
- Necesidad de dar cobertura y validez legal a la firma electrónica, siendo como es la única firma posible cuando en la transacción ambas partes sólo están comunicadas a través de Internet.

En realidad a la legislación sobre el comercio electrónico aún le queda mucho camino por recorrer, y se apunta como única solución definitiva la creación de una legislación internacional que evite perjudicar este comercio por vía de la doble imposición normativa (tanto del país origen como del destino) o la excesiva regulación que se está dando en estos momentos. Como ejemplo de esa maraña de normativas, valga el ejemplo de que existiendo una Directiva de la Unión Europea., la Directiva 2000/31/CE, que regula el comercio electrónico y que fue emitida en el 2.000, hasta fechas recientes (finales del 2.002), no apareció en España la LSSI que afirma adaptar a nuestro país dicha directiva, con lo que mientras que en países de nuestro entorno ya hace algún tiempo se procedió a realizar esa adaptación y por tanto una transacción entre España y un tercer país europeo debía cumplir tanto la normativa que adaptaba la directiva, en España había de recurrirse a toda una serie de leyes y normas anteriores, como el Real Decreto 14/99 que regula el uso de la firma electrónica. Estamos hablando de la legislación que ofrecería garantías al comprador en temas de garantías y sus derechos como consumidor, como la que ofrece las mismas garantías al vendedor, ante morosos, estafas.

Otro tema aparte es el tratamiento fiscal, especialmente por la actual situación tributaria que se da en la mayoría de países occidentales.

En las dos últimas décadas las Haciendas de los países occidentales han ido reduciendo sus cargas fiscales directas en beneficio de las cargas indirectas, vía impuestos especiales como el IVA. Estos impuestos resultan especialmente perjudicados por aquellas transacciones internacionales, en las que no se paga IVA o bien resulta complicado su control y seguimiento, abiertas por tanto a un mayor fraude que los métodos tradicionales. En definitiva las Haciendas de los países temen una reducción de la recaudación.

En un artículo de Isabel Gómez Calleja[44] se habla de esta problemática y de su actual tratamiento al más alto nivel institucional en seno de la Organización Mundial del Comercio (www.wto.org): tanto esta abogada como los comités de la organización abogan por una clara segregación del mercado B2B y del B2C. B2B viene de *Business to Business*, denominándose así a las transacciones ocurridas entre

[44] Gómez Calleja, Isabel (2.001): La *tributación indirecta de las ventas online*. El Mercantil Valenciano, edición del 15 de abril del 2.001 en la sección *Bolsa y mercados*.

empresas (y donde por tanto no hay IVA porque conduciría a la doble imposición). Por otro lado, una operación entre una empresa y un consumidor final se denomina B2C, que viene de *Business to Consumer*. Esta diferenciación permitiría implantar mejores controles sobre estas transacciones por las Haciendas públicas.

En Europa además existe un espacio intracomunitario en el que el comportamiento es como si estuviéramos ante una economía nacional: cualquier operación entre ciudadanos de países miembros sigue exenta de gravámenes pero no de IVA, el cuál difiere además entre los distintos miembros de la Unión.

Por tanto ha sido la Unión la que mediante directivas y órdenes haya tenido que establecer a qué país compete el cobro del IVA y qué operaciones estarán sujetas a él:

- Si el operador de la prestación es de un tercer país y el comprador es europeo, la operación está sujeta al IVA del país del comprador (obligando además al operador a establecer una filial en Europa para gestionar estas transacciones).
- Si el operador es europeo y el comprador extracomunitario, la operación no devengará IVA.

Si la operación es entre países miembros, se distingue si el comprador es empresa o consumidor final:

- Si el operador comercia con otra empresa, ambas europeas, el lugar de prestación será el de la empresa compradora y devengará IVA en dicho país.
- Si el operador comercia con un particular, ambos europeos, el lugar de prestación será el del operador y el IVA el del su país.

Esta definición, introducida en la directiva 2000/31/CE antes comentada, aunque parezca lo contrario ha simplificado las cosas con respecto al comercio con terceros países extracomunitarios, donde a las consideraciones anteriores habríamos de añadir la legislación de dicho país (cuando aquí ya es suficiente con la directiva, al garantizarse que cada país la habrá aplicado respetando el espíritu de la misma).

La otra directiva relacionada con el comercio sería la directiva 2002/58/CE sobre la Privacidad y las Comunicaciones Electrónicas, que comentaremos posteriormente.

4.2.2 Regulación penal y social de Internet

Pese a que nos centraremos en la situación legal en España, la actual regulación de las telecomunicaciones es fruto de la adopción a nivel mundial por parte de los países desarrollados del modelo estadounidense, que aboga por la existencia de un mercado en libre competencia (con ausencia de monopolios en cualquiera de los sectores afectados: telefonía, Internet, etc.) y busca la autorregulación del propio sector a través de la creación de organismos independientes que arbitren y regulen este mercado.

En EE.UU. esta función de arbitraje y marco regulador es realizada por la *Federal Communications Commission* (www.fcc.gov), en un país en el que las telecomunicaciones han supuesto en los últimos años casi la totalidad del motor del crecimiento económico, y todo ello en un ambiente de plena competencia entre los operadores y tras una escisión en los 80 del monopolio telefónico que ya hemos comentado. La FCC existe desde 1.934, cuando fue creada durante el Gobierno de *Franklin Delano Roosevelt*, y su poder se extiende a cualquier aspecto de las telecomunicaciones, siendo de su exclusividad tanto el control del mercado como su regulación.

Todo ello nos lleva a entender mejor la actual situación en España: hasta esta pasada década existía un único operador, Telefónica de España (www.telefonica.es), y a partir de 1.992 con la aparición de nuevos operadores se hizo necesario un marco regulatorio, empujados además porque desde la propia Unión Europea se buscaba también la apertura: con reales decretos, directivas europeas, organismos de control, etc. Aunque no todos van a afectarnos, si que vamos a tratar de ver ahora qué organismos y leyes afectarán de manera directa a un ISP en España.

En primer lugar veamos los actores que nos afectan: desde 1.996 existe en España el Ministerio de Ciencia y Tecnología (www.mcyt.es), siendo las Telecomunicaciones tarea de su Secretaría de Estado para las Telecomunicaciones y la Sociedad de la Información (www.setsi.mcyt.es). Aparte de la Secretaría, responsable de la mayor parte de la actual legislación del sector, existe la entidad pública empresarial *Red.es* (www.red.es, cuya casi única labor viene siendo el control del registro de dominios bajo el TLD español) y la Agencia de Protección de Datos (www.agenciaprotecciondatos.org, cuya labor es velar por el correcto uso de la información privada de cada ciudadano).

Los delitos en Internet

Veamos también los distintos tipos de delitos informáticos por los que un ISP puede llegar a verse afectado: hay que tener en cuenta que como intermediarios o prestadores de servicios, hemos de evitar tanto que a través de nuestros servicios se cometan delitos por parte de nuestros clientes, como que nuestros clientes se conviertan en víctimas del delito informático. En ambos casos el operador se puede llegar a encontrar inmerso en un proceso judicial en el que, ante la falta de claridad sobre el responsable del delito, sea él el responsable subsidiario del mismo: se daría entonces la situación de que como intermediarios del servicio usado para cometer un delito acabemos siendo cómplices del mismo por una negligente gestión de ese servicio.

Un ISP necesita conocer todos aquellos delitos que puedan producirse a través de sus servicios, para prevenir el uso anónimo de los mismos y poder identificar con posterioridad en todo momento al responsable final del delito. Los más importantes que hemos de evitar vendrían a ser los siguientes:

- Ataques contra el derecho a la intimidad o contra la Ley de Protección de Datos: el poseer datos de carácter personal obliga a su comunicación a una agencia estatal y a cumplir los preceptos de la Ley Orgánica de Protección de Datos de Carácter Personal. En especial, es relevante que nos aseguremos de la legalidad y plena inscripción de aquellas bases de datos que sean propiedad de nuestros clientes pero que estén alojados en servicios ofrecidos por nuestra empresa, y que nos aseguremos de comunicar al cliente de la necesidad de su inscripción.
- Infracciones a la Propiedad Intelectual: dado que en Europa el software está acogido a patente intelectual en lugar de industrial, aquí se incluirá tanto el uso de software sin licencia para ello (piratería informática), como la existencia de contenidos en sitios Web cuya autoría no sea la reflejada o bien que hayan sido reproducidos sin la autorización del autor.
- Sabotajes informáticos, como los cometidos por clientes nuestros haciendo uso de nuestros servicios contra un tercero o contra nosotros mismos.
- Fraudes informáticos, por parte de clientes nuestros haciendo uso del correo electrónico por nosotros suministrado, o bien a través de un sitio Web que con nosotros tenga contratado.
- Amenazas, calumnias e injurias. Delito de relevación de secretos, pornografía infantil…todos aquellos delitos que se puedan producir en los medios de comunicación tradicionales tienen cabida en la Web, donde una página es también una publicación.

Se tratará en resumidas cuentas de establecer medidas preventivas como por ejemplo incluir cláusulas exculpatorias hacia nosotros en los contratos, que aunque no evitarán nunca que estos contratiempos nombrados se produzcan, si minimizarán el daño económico o de imagen para la empresa. Pasaremos ahora a enumerar alguna de estas medidas a tener en cuenta.

Consideraciones en la relación entre el operador y sus clientes

Los contratos entre el operador y sus clientes han de recoger la no responsabilidad por parte del operador del uso que haga el cliente de nuestros servicios, y tratar al mismo tiempo de evitar usos ilegítimos del servicio, para que posteriores delitos cometidos por clientes nuestros no puedan perjudicar al operador.

También es interesante proteger a la empresa por la calidad del servicio ofrecida: en Internet resulta complejo ofrecer garantías de calidad y disponibilidad. Aunque medir estos parámetros también resultara difícil para un posible cliente perjudicado que quisiera demandarnos, es importante que en el contrato entre operador y cliente se recoja una cierta relajación en la calidad del servicio ofertado, evitando que figuren por escrito cláusulas que vinculen a la empresa con plazos de reparación y tiempos máximos de no disponibilidad. Igualmente, es muy común que el contrato exima al operador de responsabilidad en los daños que el cliente pueda sufrir por no poder utilizar nuestros servicios.

El operador ha de disponer a nivel técnico de mecanismos de control que le permitan asegurar el uso legítimo de los servicios (contraseñas para el acceso a los mismos, es lo más usual). Es también igual de importante que el operador pueda determinar la identidad del cliente en todo momento, y que conserve registros de los accesos y actividades realizadas por sus clientes, para poder facilitar estas posteriormente a las autoridades ante actuaciones indebidas de nuestros clientes.

También incidiendo de nuevo en lo que respecta al tratamiento de datos, el cliente ha de poder acceder a los datos de carácter personal que sobre él disponga el operador, y tanto esa información como cualquier otra de que dispongamos, nuestra empresa ha de declararla ante la Agencia de Protección de Datos si fuera pertinente. Como además nuestros clientes pueden a su vez disponer de ficheros de carácter personal, es importante que en nuestro contrato recojamos que no somos responsables de los mismos, al margen de que luego informemos al cliente de su obligación de registro y declaración de dichos ficheros.

Consideraciones en las relaciones entre operador y sus proveedores

Así como con nuestros clientes trataremos de minimizar el daño por falta de servicio, con nuestro operador habremos justamente de buscar

lo contrario: hay que tener en cuenta que al actuar de intermediarios el daño producido por un operador que nos deja sin servicio temporalmente se reflejará en cientos de clientes nuestros afectados, y en el contrato de prestación que firmemos con el operador habremos de incorporar cláusulas que aseguren pagos económicos y tiempos de respuesta en esos casos.

Al igual que las caídas temporales del servicio, habremos de protegernos ante la quiebra de nuestros proveedores, o ante la ausencia prolongada del servicio: por un lado mediante unos plazos mínimos en la ruptura del contrato de servicios con nuestro operador, y por otro lado justo lo contrario, un plazo pequeño de falta de servicio que de cómo lugar a nuestro derecho a extinguir el contrato y además a exigir daños y perjuicios.

Consideraciones en cuanto a sus trabajadores

El operador ha de asegurarse de que el personal de su plantilla o de terceras partes que pueda tener acceso a datos de carácter personal de los clientes (bien en bases de datos, bien en comunicaciones producidas por el uso de nuestros servicios) sea obligado mediante el reglamento de personal o documentos similares a salvaguardar la confidencialidad de dichos datos, aún en el caso de que dicho personal cesara en su puesto de trabajo.

El trabajador ha de conocer y ha de figurar por escrito que consiente en que su acceso a datos de los clientes será por el tiempo imprescindible posible. Al igual que el apartado anterior el objetivo no es evitar que la empresa pueda ser luego responsable de las acciones de sus trabajadores, que lo será en cualquier caso, sino permitir a la empresa actuar a su vez contra el empleado por conducta improcedente, por cuanto había un reglamento o documento que no le permitía. Se trataría de una acción preventiva: el trabajador conoce de antemano qué puede implicar tal o cual acción, y además la empresa puede luego durante una demanda judicial aducir a su favor el reglamento, evitándose así la incertidumbre de que la responsabilidad se tenga que dirimir durante un juicio.

A su vez, el trabajador ha de hacer un uso adecuado de los recursos de la empresa, y se ha de poder controlar su actividad en la empresa. A este efecto es muy común en la actualidad que los trabajadores deban consentir por escrito que las comunicaciones por e-mail y el uso de la navegación Web sean controladas y usadas para acciones disciplinarias por uso inadecuado de las mismas.

Todas estas consideraciones con la plantilla nos llevan a la necesidad de que la empresa establezca un Manual de Operaciones o de un Reglamento Interno que recoja estas necesidades.

Propiedad intelectual

Aunque podría ser casi algo obvio, en España no lo es: una empresa que vaya a prestar servicios en Internet necesitará software y hardware,

y en el caso del software este no deberá ser ilegal. La piratería del software (que es como se conoce comúnmente la copia y uso sin licencia legal de programas informáticos) está en España en tasas del 57%, un índice elevadísimo según la organización de empresas de programación que lucha contra ellas a nivel mundial, la *Business Software Alliance* (www.bsa.org). Los derechos de autor en España están protegidos por la Ley de Propiedad Intelectual 1/1.996, y es esa ley junto a los artículos 270 al 272 del código penal los que usa la BSA en las 471 demandas judiciales que cada año pone en España (datos del 2.002). Es por tanto muy probable que la BSA requiera a nuestra empresa alguna vez sobre la posesión de licencias.

El software adquirido por la empresa habrá de ser acompañado de su correspondiente licencia de uso (que no tendrá porque ser siempre de pago, como luego veremos), y el control de las licencias y su renovación asignado a una persona, que además habrá de catalogarlas. Es igualmente importante que los trabajadores sean advertidos de que no pueden instalar software ajeno al existente sin permiso y conocimiento de la dirección.

La LSSI[45]

La ley 34/2002, de 11 de julio del 2.002, de servicios de la sociedad de la información y de comercio electrónico, más conocida como LSSI, es la iniciativa más relevante (y más reciente) en materia legislativa que Internet se ha hecho en España. Como tal merece ser comentada en profundidad, así como complementada con el resto de normativas legales que puedan afectarnos.

La ley nació para reglamentar el comercio electrónico a través de la red: sus siete títulos desarrollan este objetivo, incluso con sanciones administrativas, pero con tan mala fortuna que su interpretación estricta puede afectar a cualquier actividad (no sólo de comercio electrónico). Básicamente el problema es que con la ley se puede equiparar cualquier Web a un servicio a terceros, servicio que estará sujeto a control y regulación: en el caso de no ser empresas, deberán registrarse en el Registro Mercantil (por mecanismos además que permanecen sin desarrollar y que los propios registros desconocen), en todos los casos el autor de la página (y suministrador por tanto del servicio) deberá quedar identificado, etc.

De la gravedad del hecho se refleja que tras su aprobación algunas páginas reconocidas se trasladaron a servidores situados fuera de España (por miedo a verse afectadas) o incluyeron extensas notas legales en sus sitios Web (como por ejemplo **www.kriptopolis.com**), y la ley ha sido criticada incluso a nivel internacional en prensa (por ejemplo, en **www.wired.com**). En realidad lo que más asusta de esta

---

45 Ley de Servicios de la Sociedad de Información, Web del MCYT: <http://www.lssi.es>

ley es lo que no se sabe, es decir: aquellos artículos sobre los que ningún tribunal de rango superior (el Tribunal Supremo) ha sentado aún jurisprudencia, debido a que la deficiente precisión y excesiva ambigüedad de la redacción de la ley (en parte por las presiones sufridas por el Gobierno durante su redacción) han conducido por error o de manera intencionada a una ley que podría ser empleada de manera discrecional para cerrar cualquier Web haciendo una interpretación rígida de la misma.

Artículos relevantes de la LSSI

La necesidad de registrar la Web, el hacer a los ISP responsables de monitorizar y señalar a las autoridades los contenidos considerados ilícitos, al margen de que puedan o no limitar derechos fundamentales como son la libertad de expresión (discusión en que no entraremos), afectarán directamente a nuestra actividad, y trataremos ahora de detallar algunos de estos mejor para ver qué medidas tomaremos al respecto.

Cuando entró en vigor la ley, ni los Registros Mercantiles, ni el Colegio de Registradores ni el propio Ministerio de Ciencia y Tecnología sabían cómo proceder para registrar una Web comercial, el registro de las cuales se planteaba en su artículo 9 como una de las vías para adquirir la necesaria personalidad jurídica como tal. A igual ocurre con algunos puntos de la controvertida ley, que serán regulados por reglamentos posteriores que están todavía por desarrollar, pero en ningún caso está claro cuáles de entre los requisitos son de aplicación inmediata y cuáles no En realidad ya con la ley en vigor ninguna empresa está optando por este registro, ya que resulta ambiguo entender a quién afecta este artículo. El problema viene con aquellos sitios no estrictamente comerciales (páginas personales, sitios de ONG, sitios Web de información y noticias... careciendo todos ellos del registro mercantil que si que posee cualquier empresa dedicada a comerciar), pero en los que el simple acceso a los mismos pueda ser interpretado de manera estricta como una prestación de servicio, sujeta por tanto a regulación.

Por otro lado, el artículo 12 obliga a los ISPs a retener los datos de acceso (entendiendo por ello tanto navegación, como conexión a algún servidor, envío de correo, etc.) de los usuarios durante seis meses. Sin embargo, no consideraremos este artículo un problema para un ISP, básicamente porque esta información es igualmente útil para estadísticas. También resulta espinoso el tema de los contenidos de la red, haciéndose a los prestadores responsables de la comunicación los contenidos que pudieran resultar ilícitos.

Existe también para el ISP responsabilidad por realizar copias temporales de los datos solicitados por los usuarios: que se ha venido interpretando que afecta a los prestadores de un servicio de *proxy*.

Algunos proveedores de acceso lo utilizan para reducir su tráfico hacia Internet, y quizás se esté ofreciendo una página que no se halla

actualizada. Parece ser que la LSSI se refiere a este tipo de servicios pues la dicción expresa de la misma en su artículo 13 hace referencia a los prestadores de servicio que «transmitan por una red de telecomunicaciones datos facilitados por un destinatario del servicio y, con la única finalidad de hacer más eficaz su transmisión ulterior a otros destinatarios que los soliciten, los almacenen en sus sistemas de forma automática, provisional y temporal».

En realidad, un servicio de *proxy* lo que almacena no son datos facilitados por un destinatario, sino facilitados por un emisor de información a requerimiento de un destinatario. En este supuesto, la LSSI exonera de responsabilidad al ISP (cuando cumple funciones de *proxy*) siempre que cumpla con los siguientes requisitos:

- No modifique la información suministrada.
- Se permita el acceso a la información por parte únicamente de los destinatarios que cumplan condiciones impuestas a tal fin por el destinatario cuya información se solicita. Se desconoce qué quiere decir la LSSI con este requisito.
- Se respeten las normas generalmente aceptadas y aplicadas por el sector para la actualización de la información. Se hace referencia a las pautas de actualización del contenido grabado en el ordenador que sirve de *proxy*.
- No se interfiera en la utilización lícita de tecnología generalmente aceptada y empleada por el sector.
- Retire o haga imposible el acceso a la información siempre que ya no exista en el ordenador de origen, se haya imposibilitado el acceso a ella o un tribunal o una autoridad administrativa haya ordenado la retirada de la información o su acceso a la misma (este último párrafo resulta además escandaloso por cuanto puede afectar al derecho a la libre expresión que un órgano administrativo tenga capacidad para cerrar un diario digital).

Además existe responsabilidad de los prestadores de servicios que faciliten enlaces a contenidos o instrumentos de búsqueda, aunque distingue entre proveedores de servicios de enlaces (*Yahoo!*) y el de buscadores (*Google, Altavista, etc.*).

Con el hipertexto que estos sitios enlazan o indexan mediante motores de búsqueda, el derecho se plantea la licitud de las siguientes actividades:

- Señalar un enlace sin permiso del enlazado.
- Señalar un enlace a un sitio cuyo contenido es ilícito (páginas de contenidos pirata, pornografía infantil, como construir una bomba de neutrones…).

- Llamar a una página de otro autor dentro de un marco (es decir, listar contenidos ajenos sin permiso y dentro de nuestra página mediante el uso de *frames*).

En estas cuestiones, el Derecho mezcló dos conceptos, el primero relacionado con la propiedad intelectual del lugar enlazado y el segundo hacía referencia al Orden Público, esto es, la licitud de enlazar con un sitio ilícito. La LSSI admite tácitamente la licitud de los enlaces sin consentimiento previo del enlazado pero subordina dicha licitud a la legalidad de la página enlazada. Así, se establece que los directorios o los buscadores no serán responsables de la inclusión del enlace en dos supuestos:

- Que no tengan conocimiento efectivo de que la actividad o conformación a la que enlazan sea ilícita o pueda lesionar bienes o derechos de un tercero.
- Que, aun teniendo dicho conocimiento, actúen con diligencia para suprimir el enlace. Este artículo constituye una demostración más de la mala redacción de esta ley, en la que se deja a criterio judicial el calibrar la diligencia en suprimir el enlace

Se presume que se conoce efectivamente la ilicitud del enlace cuando una autoridad competente haya declarado la ilicitud de los datos y ordenado su retirada o haya ordenado la imposibilidad de acceso a los mismos. La presunción legal de que el prestador de servicios conoce la ilicitud del enlace depende de la declaración previa de la autoridad competente (de nuevo ambigüedad: no se sabe si un juez o la propia Administración). Además queda la duda acerca de la aplicabilidad en España de las resoluciones dictadas por un Tribunal extranjero que no sean directamente aplicables en territorio nacional.

Continuando en el siguiente artículo, se establece en el artículo 14 de la LSSI una exoneración de responsabilidad de los operadores de redes sobre los contenidos transmitidos, siempre y cuando el prestador de servicios no los hubiera originado, modificado o seleccionado. En este sentido esta ley supone una mejoría de la situación anterior, como ahora veremos.

En los medios de comunicación tradicionales, la responsabilidad pasa en orden del autor, luego al director por permitir su publicación, y al editor por último, por ser el representante legal de la compañía editora. Una de las primeras preguntas que Internet obligó a plantearse fue la de la traslación a las redes del concepto tradicional de responsabilidad, haciendo recaer en el ISP la antigua responsabilidad del editor.

De hecho en la justicia sajona se ha dado este caso con un resultado de condena contra el ISP, que actuaba de mero intermediario, sentando jurisprudencia, ya que en el derecho sajón la jurisprudencia es establecida directamente por los tribunales al dictar sentencia en

aquellos aspectos no cubiertos por las leyes. En 1.995 se presentó un famoso caso, el de Stratton Oakmonth Inc. contra Prodigy, visto en la Corte Suprema del Estado de Nueva York[46]. La demandante, la asesora de bolsa Stratton Oakmonth, había sido anteriormente condenada en firme por difundir en un foro noticias falsas sobre ciertas acciones para alterar la cotización de estas, y buscaba con la demanda hacer coautor del delito al ISP por el que Stratton se conectaba, cosa que logró con la sentencia. Afortunadamente, sentencias posteriores han ido corrigiendo esta jurisprudencia, achacable a un desconocimiento profundo de lo que implicaba Internet por parte de los jueces que emitieron esta sentencia.

La CMT

La Comisión del Mercado de Telecomunicaciones (www.cmt.es) es un organismo independiente creado en 1.996 por el Gobierno español para que actúe de regulador y árbitro a nivel estatal en el campo de las telecomunicaciones, asesorando al Gobierno en el marco legislativo (no tiene por tanto potestad legislativa completa) y vigilando a los operadores, ya que su objetivo es que los servicios prestados por los operadores a sus clientes lo sean en las condiciones adecuadas. Esta comisión surge como reacción a directivas europeas que abogaban por un mercado de las telecomunicaciones similar al americano, y realmente España fue una de las más tempranas en su implantación: iniciativas similares como la autoridad alemana *Regulierungsbehörde für Telekommunikation und Post* (www.regtp.de) datan de enero de 1.998, o la francesa *Autorité de Régulation des Télécommunications* (www.art-telecom.fr), que se creó en enero de 1.997. En todo el marco de la Unión Europea existen en la actualidad organismos nacionales independientes similares a la CMT.

La CMT posee capacidad sancionadora a nivel administrativo sobre los operadores, y su fuente de financiación exclusiva son las tasas creadas al efecto por el Estado, por lo que está sufragada de manera directa por los propios operadores que controla. De estas tasas unas se cobran por la concesión de autorizaciones, y las otras se establecen cada año como un porcentaje de los ingresos brutos de explotación de servicios de telecomunicaciones de las operadoras.

Su Consejo de nueve miembros está complementado por la plantilla propia del organismo (formada por técnicos, personal directivo y de administración), siendo potestad exclusiva del Consejo la toma de decisiones en cuanto a arbitraje y legislación. Los miembros del Consejo

---

[46] Sentencia del caso *Stratton Oakmonth Inc.* contra *Prodigy*, visto en la Corte Suprema del Estado de Nueva York (1.995):

<http://www.jmls.edu/cyber/cases/stratton.txt>

son elegidos por el Gobierno a propuesta del propio consejo por periodos de dos años, aunque es común que se renueven en el cargo.

La CMT concede además los títulos que habilitan para prestar servicios de telecomunicación. La otra función importante de esta comisión es el análisis económico anual del sector, para establecer a partir de los costes de los servicios unos precios regulados que serán de aplicación para aquellos operadores mayoritarios que controlan gran parte del mercado (lo que es conocido como *price cap*). Esta última labor no tendrá ningún interés para nuestro objetivo, ya que nos centraremos tras esta breve descripción en analizar qué licencias y requisitos habremos de disponer ante la CMT para poder operar en España como prestador de servicios en Internet.

En España es necesario disponer de una licencia para actuar como prestador de acceso a la información. De estas licencias existen 4 tipos fundamentales: La llamada A para la telefonía pública, B para la telefonía con arrendamiento a terceros, es decir, para intermediarios (B1 para telefonía fija y B2 para telefonía móvil) y las que nos pueden afectar, las de tipo C, para redes de comunicaciones que excluyan a la telefonía (Internet, por ejemplo).

Existe una licencia C1 para aquellos que no usen el espacio radioeléctrico, que será la que solicitaremos, y otra llamada C2, aplicable cuando el operador vaya a utilizar espacio radioeléctrico (del que además habrá que pagar por su ocupación). Las licencias C2 no son otorgadas por la CMT, pues al ser concursos son llevadas directamente por el Ministerio.

La agencia de calidad en Internet: la IQUA (www.iqua.net) es otra iniciativa de reciente aparición, y que ha surgido a iniciativa de la CMT y otros organismos reguladores en España para promover la autorregulación en Internet, buscando la implantación de un sello de calidad que avale el cumplimiento por parte de la empresa de dicha Web de un código deontológico y de buenas prácticas. La calidad se ha convertido en los últimos años en una necesidad en el mundo empresarial, y por tanto no dejaría de tener interés que nuestro ISP dispusiera de alguno de estos sellos, que vienen a suplir las carencias de la legislación de cada país: no todo es regulable, o bien la legalidad vigente queda superada por el avance de las tecnologías, y en estas situaciones un sello de calidad ofrece, desde el punto de vista de un cliente, la creencia en que el producto ofertado por nuestra empresa cumple más allá de lo que la ley obliga (en materia de protección de datos, reclamaciones de usuario y calidad de servicio en el caso de los ISP, por ejemplo).

Estas iniciativas no dejan de resultar interesantes, pero al ser de carácter privado carecen de validez legal (lo que abre el campo a la subjetividad de cada organismo, y al igual que ocurre con los sellos de calidad de asociaciones como AENOR (www.aenor.es), ofrecen prestigio e imagen que se

puede usar en campañas publicitarias, pero que quedarán fuera de este estudio, al considerarse que una compañía de tamaño pequeño o mediano como la que estamos tratando en este proyecto, tiene presupuestos que quedan fuera de la capacidad necesaria para estos sellos de calidad.

La protección de datos en España la LPD y la APD

En España, la Ley Orgánica 15/1.999 de 13 de diciembre, conocida como Ley de Protección de Datos de Carácter Personal (o bien Ley de Protección de Datos) derogó una ley previa que era la LORTAD (Ley Orgánica 5/1.992 de Tratamiento de Datos de Carácter Personal, de 29 de octubre de 1.992).

Su carácter de ley orgánica ya nos indica la importancia de ésta: afectan a un derecho fundamental reconocido por la Constitución, como es el derecho a la intimidad, implicando esto el conocer dónde están nuestros datos registrados, acceder a los mismos o cancelarlos cuando creamos oportuno. Hablamos en pocas palabras de un *habeas data*: el derecho a conocer en todo momento qué y quién sabe de él.

Internet es un medio publicitario más, y es justamente la publicidad la que más trabajo genera en el organismo independiente encargado de velar por el cumplimiento de esta ley, como es la Agencia de Protección de Datos (**www.agenciaprotecciondatos.org**), y será esta área de jurisprudencia (Internet) la que vamos a tratar de analizar ahora. Realmente hemos de tener en cuenta que no es Internet el objetivo real de esta ley ni tampoco el *marketing directo*, sino que hay aspectos regulados por esta ley cuya importancia es mucho mayor y que ni siquiera nombraremos aquí (protección de los datos del censo electoral, de los datos de filiación religiosa, política u orientación sexual, de los datos médicos incluyendo el acceso al mapa genético, etc.).

En realidad la única aplicación de esta ley en Internet será el marketing directo, ya que la recogida de información del usuario durante el acceso y como parte de la actividad de estadísticas está autorizada, además de ser necesaria: Los interesados a los que se soliciten datos personales deberán ser previamente informados de modo expreso, preciso e inequívoco de que los mismos van destinados a un fichero y que las respuestas que deban dar a las preguntas que les sean planteadas deberán indicar si son de carácter obligatorio o facultativo; deberá también hacerse figurar la identidad y la dirección del responsable del tratamiento de los datos o, en su caso, de su representante, de cara a hacerles exigibles las responsabilidades que contraigan.

Hablamos de un consentimiento inequívoco por parte del afectado, además de regular la comunicación de esos datos a terceros, autorizado únicamente para fines relacionados y que figuren en el ámbito del consentimiento que el usuario previamente ha dado. Para su seguimiento y aplicación, la Agencia antes nombrada, que actúa con independencia económica y funcional de la Administración pública, dispone de una batería de sanciones administrativas, que es el único ámbito sancionador posible. Realmente su función y estructura se asemeja mucho a la comentada para la Comisión del Mercado de las Telecomunicaciones antes nombrado.

Ventajas de la ley hay bastantes: obliga a diferencia de la LORTAD al consentimiento previo expreso, a diferencia de la situación anterior, en la que se daba únicamente el caso de aceptar la publicidad por le mero hecho de no haberse opuesto a él. Por el contrario, una laguna abierta por esta ley sería el considerar que la recopilación de direcciones y posterior reparto de publicidad, venta a distancia, prospección comercial o actividades análogas sólo podrán utilizar nombres y direcciones que figuren en fuentes accesibles al público, con lo que publicar nuestros datos de contacto en la Web daría justificación a que alguien nos bombardeara a *spam*, siempre y cuando los datos hayan sido facilitados por los propios interesados constando el consentimiento de éstos. En el ejercicio del derecho de acceso los interesados podrán conocer el origen de sus datos.

La venta a través de Internet en España

Cuando la relación entre clientes y empresas se realiza a través de medios no presenciales, estaremos ante un problema en cuanto a la validez de los contratos celebrados, teóricamente resuelto en España a través del Decreto Ley 14/1999, de 17 de Septiembre de 1.999.

Otro aspecto de dicha relación es su consideración de venta a distancia, regulada a través del Real Decreto 1133/1997 de 11 de Julio de 1.997, que fue posteriormente modificado por otro Real Decreto, el 1976/1998. De estos dos reales decretos se deduce la obligatoriedad de inscripción en el Registro General de Empresas de Venta a Distancia (conocido como REVA), de carácter único en todo el territorio nacional y dependiente del Instituto Nacional de Consumo (**www.inc.es**).

Para las transacciones electrónicas el Gobierno hizo el Real Decreto 1906/1999, que venía a desarrollar en él el artículo 5.3 de la Ley de Condiciones Generales de Contratación, 7/1998 del 13 de abril de 1.998.

El asentamiento de la firma electrónica, cubierto por el Decreto Ley 14/1999 y que deberá rubricar cualquier contrato a distancia, está aún por producirse por razones técnicas. Si bien hace ya bastantes años de esta Ley (y de hecho ya apareció tres años más tarde que la alemana, por ejemplo) la dificultad en lograr el no repudio y el mutuo reconocimiento de las autoridades emisoras de los certificados, complican su plena extensión.

Una firma electrónica se basa en cualquiera de los algoritmos de clave pública disponibles, existiendo una clave pública y otra privada, así como una autoridad emisora del certificado del que cada usuario dispone, autoridad que deberá ser reconocida por ambas partes.

4.3 El software de Internet

4.3.1 Los sistemas operativos que dan soporte a Internet

GNU es un proyecto potenciado desde la *Free Software Foundation* que inició en 1.984 el desarrollo de un sistema *Unix* libre y gratuito, y que ha desembocado entre otros, en *Linux*. Si la década de los 80 la mayoría del software era propietario, se espera que en los próximos años cada vez más el software libre gane terreno, y en la Red eso es donde más se nota. Sus armas son muy simples: es abierto al análisis de terceros, fomenta la cooperación en los proyectos ya existentes (todo programa con licencia GNU ha de estar abierto a que sea mejorado por terceros) y permite que el esfuerzo económico de una empresa se traslade a la investigación y formación en lugar de hipotecar esos recursos en adquisiciones de costosas licencias de uso y sacrificar (como se ha hecho muchas veces) la formación y la mejora de los productos.

Linux es software de referencia con licencia GNU. Comenzó a ser desarrollado en 1.991 por *Linus Torvalds*, y aún hoy día mantiene *Linus* el control del núcleo, que es la parte fundamental de este sistema. El resto de programas y librerías que se pueden encontrar dentro de cualquier distribución *Linux* difieren entre sí y no se han de considerar parte integrante del sistema operativo.

Frente a la estrategia de otra compañía de crear una versión distinta de su sistema según el perfil de uso (una familia para servidores, una versión para uso doméstico y otra orientada a la oficina) *Linux* siempre ha tenido una versión única, lo que ha hecho por un lado que el código sea más estable e integrado (por lo que respecta al *kernel*, sin entrar a valorar la existencia de diferentes distribuciones) pero sacrificando a cambio el mercado doméstico, donde la complejidad de *Linux* no atrae demasiado. Pese a que esto está cambiando últimamente (debido sobretodo a la aparición de escritorios y suites ofimáticas, y a que las distribuciones están simplificando la instalación) el caso que aquí nos ocupa, que es el de los ISP, siempre ha sido un mercado natural para esta evolución de *Unix*.

Unix nació en la década de los 60 a partir del tesón de *Ken Thompson* en diseñar un sistema de archivos. Cuando en 1.971 *Dennis Ritchie* y *Kerningham* crearon el lenguaje C el mismo *Ritchie* y *Thompson* acabaron rescribiendo en dicho lenguaje el sistema de archivos y ya puestos un sistema operativo multitarea para que fuera ejecutado por un PDP-11.

El salto a nivel comercial aún tardó en producirse, y se produjo en 1.977 cuando se mejora y lanza *UNIX System III*, que al llegar la Universidad de *Berkeley* es mejorado y bautizado como *Unix* 3BSD (que viene de *Berkeley Software Distribution*), que mejoró rápidamente hasta

llegar a 4.2BSD. Casi al mismo tiempo *AT&T* creaba su *System V* o SVID. En apenas un lustro prácticamente todas las grandes corporaciones acabaron creando un sistema operativo basado en alguno de estos dos: de tipo BSD la DEC desarrolló *Ultrix* y *Sun* creó *SunOS*, mientras HP lanzaba HP-UX con estructura similar al *System V* de *AT&T*. La diversidad de nombres se debe a que la palabra *Unix* es marca registrada de *AT&T*, y por tanto los demás fabricantes se veían obligados a buscar otras formas de llamar a lo que en esencia constituía un sistema *Unix*.

También hubo otros desarrollos: Microsoft llegó a desarrollar *Xenix*, aunque sus derroteros acabaron por otras aguas debido a NT, mientras IBM lanzaba otro *Unix* más, llamado AIX. En el mundo académico *Tanenbaum* creó en 1.983 *Minix*, con afán únicamente didáctico (el mismo que movió en 1.984 a Comer a crear *Xinu*).

De todos estos desarrollos muchos han sobrevivido e ido mejorando hasta que en la actualidad sólo presentan como característica común el continuar usando C como lenguaje para su desarrollo. Se ha llegado por tanto a un grado tal de heterogeneidad dentro de los sistemas basados en *Unix* que la reacción actual tiende a corregir esto con una convergencia de los sistemas.

De los grandes fabricantes IBM continúa montando AIX en parte de sus equipos y SUN ha transformado a un *Unix* estilo *System V* su sistema operativo *Solaris* (la evolución de *SunOS*). Por el camino han quedado los otros sistemas, mientras que han aparecido otros desarrollos nuevos que bajo licencia GNU tratan de ofrecer características presentes en alguno de esos proyectos abandonados.

Por ejemplo ahora que *Berkeley* ha anunciado tras el lanzamiento de 4.4BSD que no va a continuar desarrollando un sistema *Unix*, se dispone tanto de *NetBSD* como de *FreeBSD*, proyectos con características similares, sin nombrar al SCO *Unix* con *UnixWare*.

El hecho de usar plataformas *Unix* es pues una decisión de estabilidad: es escoger un sistema históricamente destinado a servidor, frente a Microsoft, que hasta la década de los 90 había sido un fabricante de sistemas destinados al usuario final y que luego se pasó al mercado de los servidores (primero con *Xenix*, luego con IBM y OS/2, y finalmente con la familia NT).

Pero el hecho de usar *Linux* (que consideraremos uno más dentro de la familia *Unix*) supone un paso más allá todavía: supone confiar en un sistema operativo abierto. Y nuestro proyecto va a buscar intencionadamente confiar en un sistema abierto por razones evidentemente de coste, pero también por considerar que no está justificado el que el sistema operativo propietario por excelencia vaya a aportarnos mejora alguna en nuestro nicho de actuación, que es el de las Redes, para las que UNIX parece hecho como anillo al dedo.

De hecho hasta hace muy poco, Microsoft carecía de cualquier herramienta de actualización automática de sus productos, mientras

esta herramienta estaba disponible hacía bastante tiempo en la mayoría de distribuciones *Linux*. Y también es un hecho que fabricantes de la talla de IBM o HP tardan muchísimo más tiempo en solucionar algunos de los *bugs* que a las pocas horas ya se encuentran parcheados en *Linux*.

Por tanto y aunque en última instancia siempre va a depender de la capacidad técnica de las personas que vayan a administrar una red, es estratégicamente necesario que el sistema operativo que ofrezca los servicios sea seguro, y hoy por hoy parece que el software libre tiene las de ganar.

Otro aspecto a tener en cuenta es la distribución del gasto: los sistemas abiertos ofrecen un coste nulo de licencias, pero a cambio exigen un nivel mayor para administrarlos (carencia de interfaces, fundamentalmente), y esto nos lleva a invertir más en formación. Pese a todo ello, los costes son inferiores al mismo proyecto llevado a cabo en sistemas propietarios y cerrados, y es por ello por lo que la informática podrá llegar a mercados a las que antes no llegaba por razones de costes. Y por esta razón tenemos que pensar en *Linux* como algo positivo: no atenta al modelo de negocio de la Informática, ya que el software se sigue vendiendo en la misma medida que antes y a los mismos precios, lo que ha ocurrido es que se ha extendido hasta capas y negocios que si hubieran tenido que pagarlo, no hubiera llegado.

Cuando comenzó a crecer, *Linux* no fue tenido en cuenta como amenaza: no era un sistema soportado por fabricante de hardware alguno ni tenía detrás a una compañía de software que asegurara su futuro. En la actualidad, no sólo no ha desaparecido, sino que tiene asegurado un puesto importante incluso en el segmento de los servidores, y por todo ello será muy tenido en cuenta en este proyecto.

4.3.2 Servidores de correo

El origen del servidor de correo electrónico hemos de buscarlo en UUCP, cuando los ordenadores se intercomunicaban directamente entre ellos sin usar red alguna, conectándose mediante módem los unos con los otros.

Era una situación en la que intercambiar los mensajes (lo cual se producía mediante la utilidad *mail*) se realizaba entre las distintas máquinas a las que se conectaba el servidor de manera puntual, con lo que el enrutamiento de los mensajes era complejo (en ausencia de una red era también complejo asegurar que la entrega del correo se había producido con éxito, ya que sólo disponíamos de sistemas conectados ocasionalmente vía módem).

La solución a esto fue un programa llamado *delivermail*, que apareció en 1.979 en los sistemas BSD 4.1. Pero en ARPAnet las cosas eran bastante más complejas de lo que parecía y *delivermail* se hace cada vez más y más complejo.

Esa complejidad hacía más inestable al sistema, y tras algunas discusiones, apareció en 1.982 el primer RFC que habla sobre un nuevo protocolo para transferencia de correo electrónico: SMTP (de *Simple Mail Transfer Protocol*)[47]. Este luego sería posteriormente extendido en otro RFC, en lo que se daría en llamar SMTP extendido o ESMTP[48].

Y así llegamos al nacimiento de *Sendmail*: fue Eric Allman, de la Universidad de Berkeley, el que modificó *delivermail* y lo convirtió en *sendmail* tras adaptarlo al protocolo SMTP. La primera versión pública de *sendmail* apareció en la distribución 4.1c del Unix BSD que esa Universidad distribuía. Esta fue una de las más famosas versiones del BSD, ya que también fue la primera en soportar el protocolo TCP/IP.

Desde entonces *Sendmail* ha crecido tanto en complejidad como en implantación: siete de las diez empresas de la lista FORTUNE 10 aún emplean *Sendmail* como sistema de correo, lo que da muestra de su fuerza.

Lo negativo de *Sendmail* reside en el hecho de que el ser tan antiguo su código y haberse ido adaptando poco a poco, lo hagan al tiempo uno de los códigos más seguros por lo verificado que se encuentra, y uno de los más complejos de modificar. Tan complejo resulta también configurar un servidor *Sendmail*, ya que el fichero de configuración es

---

[47] Jonathan B. Postel (1.982): *Simple Mail Transfer Protocol*, RFC 821:

<http://www.ietf.org/rfc/rfc821.txt>

[48] J. Klensin y otros (1.994): *SMTP Service Extensions* (RFC 1651), después obsoleto por el RFC 1869 y también por el más reciente RFC 2821, de abril del 2001:

<http://www.ietf.org/rfc/rfc1651.txt>

<http://www.ietf.org/rfc/rfc1869.txt>

<http://www.ietf.org/rfc/rfc2821.txt>

complejo (existen incluso programas y *scripts* encargados únicamente de generar la configuración).

La complejidad es algo innato a un agente de correo, debido a que en él se encuentra al mismo tiempo un agente de entrega de correo (MUA, *mail user agent*) y un agente de distribución de correo (MTA, *mail transfer agent*).

El MUA es el sistema encargado de la distribución local dentro del servidor (es decir, para las direcciones locales), mientras que el MTA es el encargado de encaminar el correo hacia localizaciones remotas, tras detectar que el destino no es local. Asimismo, el MTA tiene que ser capaz de reescribir las direcciones de manera que sean compatibles con el sistema remoto, y permitir el seguimiento de la ruta realizada por el correo, insertando las cabeceras adecuadas. El MTA y el MUA han de ser además capaces de soportar alias de direcciones de correo, y otras muchas características, como la autenticación del usuario conectado.

Existe además otro sistema, que es el que usan los clientes para recoger posteriormente su correo, que en la actualidad es necesario dado que la mayoría de usuarios han dejado de trabajar mediante consola, usando además protocolos distintos al comentado SMTP, como serían el POP3 y el IMAP, protocolos mucho más recientes[49]. Este sistema puede ser parte integrante del servidor de correo, aunque ni *Sendmail* ni muchos otros lo integran.

Es por eso por lo que en la actualidad, aunque sigue siendo mayoritaria la implantación de *Sendmail*, es interesante analizar otros agentes de correo mucho más simples, aunque para ello nos concentraremos antes en hablar sobre la autenticación del cliente y el propio protocolo SMTP.

El protocolo SMTP, MIME y las mejoras ESMTP

El protocolo SMTP que hemos estado comentado se usa en la actualidad es un protocolo relativamente simple, basado en un diálogo de comandos que se realiza únicamente a través de un puerto, el 25.

Los comandos disponibles son indicados por el propio servidor si se solicitan mediante HELP en muchas ocasiones, lo que da muestras de lo simple que resulta. Tan simple como que al ser heredado de la época del UUCP, adolece de algo tan básico como la capacidad para transmitir caracteres internacionales o código binario, y es incapaz en su implementación original de transmitir otros caracteres que no sean los ASCII imprimibles.

---

[49] J. Myers, Carnegie Mellon y M. Rose(1.996): *Post Office Protocol - Version 3*. RFC 1939, implementación más reciente de este protocolo:

<http://www.ietf.org/rfc/rfc1939.txt>

M. Crispin (1.996): *Internet Message Access Protocol – versión 4rev1*, protocolo IMAP definido en el RFC 2060:

<http://www.ietf.org/rfc/rfc2060.txt>

La solución tradicional pasaba por enviar una codificación UUENCODE del mensaje, con la que el mensaje original se trocea en líneas de 62 caracteres, pero sólo los imprimibles del juego ASCII de 7 bits, incluyendo el retorno de carro (es decir, hasta el valor #63 de la tabla de caracteres ASCII). Para hacer posible esta codificación, cada 3 bytes del mensaje binario original se transforma en 4 bytes, con lo que de entrada esta solución supone que el mensaje tenga un 33% extra de tamaño.

El primer comando necesario según el RFC es siempre el saludo inicial. A partir del momento en que introducimos un comando el servidor siempre responderá con un código numérico y un texto como resultado de nuestra petición, código que funciona de una manera muy similar a como funcionan los códigos devueltos por un servidor Web, tres dígitos en los que el primero indica que todo fue correctamente si fue un dos, error si comienza por cinco, etc.

La lista de los comandos disponibles es la siguiente:

- HELO, inicia el dialogo e identifica la maquina desde la que se establece la conexión. Lo habitual hoy día debería ser verificar este saludo (es decir, obtener la resolución inversa de la IP del servidor que se nos conecta y verificar que coincide). La IP desde la que se conecta también se utiliza a menudo para bloquear mediante alguna de las listas negras existentes. La respuesta del servidor a este comando (que ha de ser el primero) será devolvernos su identificación.
- MAIL FROM: <remitente> indica que vamos a enviar un mensaje, y que el origen es el indicado. Es también habitual verificar que esta dirección tiene un dominio de Internet válido, aunque los MTA más recientes llegan al extremo de verificar completamente la dirección haciendo una conexión al servidor correspondiente y sin llegar a enviar un correo, verificar que la dirección es reconocida.
- RCPT TO: <destinatario> indica la dirección de destino del correo. Pueden ser especificados diversos destinos, pero solo un remitente.
- DATA indica el comienzo del mensaje. Para finalizar la introducción de datos, se introduce una línea que comience por punto. En el caso de querer introducir una línea que comience por punto dentro del texto, lo haremos duplicando dicho punto.
- QUIT para desconectar del servidor.
- EXPN, aunque debería servir para indicar como se va a resolver la dirección de correo del RCPT que le indiquemos, lo normal es que este comando no esté implementado como tampoco lo está VRFY. La razón es porque junto al comando VRFY (que sirve para saber si el servidor va a aceptar o no una

dirección de correo) era usado en un antiguo truco de *hacker* para buscar usuarios en un sistema.

- VERB para pedirle que presente además de las respuestas numéricas que el servidor devuelve, un texto descriptivo o explicación. Normalmente esa respuesta ya devuelve los textos, por lo que tampoco es un comando muy útil hoy día.
- RSET para reiniciar y comenzar de cero.
- TURN es otro comando hoy día no usado y arcaico: indicaba que el cliente pasa a modo servidor, esperando por tanto que el servidor le enviara sus correos. Era habitual cuando UUCP y las conexiones telefónicas, pero casi ningún MTA lo utiliza hoy en día.
- ETRN fuerza el envío de correo dirigido a un determinado host o dominio en el servidor. Su implementación y uso es raro, por cuanto se espera que sea el servidor pueda decidir el enrutamiento correcto para el correo.

Como ya se ha comentado, la mayor limitación del protocolo SMTP era la imposibilidad de enviar mensajes de 8 bits. Es por eso por lo que en el diseño de las mejoras se incorporó esta característica, con lo que un servidor que soporte ESMTP es capaz de hacer ese envío. El problema sigue siendo que no podemos controlar qué soportarán los servidores por los que puede pasar el mensaje, con lo que sigue siendo necesario usar UUENCODE o métodos similares.

La manera de conectar a un servidor indicando que usamos y reconocemos las extensiones ESMTP pasa por usar la palabra EHLO en lugar del HELO en el saludo inicial. La respuesta del servidor también variará, ya que ahora nos indicará no sólo su identificación, sino las características extendidas que soporta (PIPELINING, el tamaño máximo del mensaje que va a aceptar en bytes, así como los métodos de autenticación soportados).

Pero aunque no tuviéramos el problema de tener que transmitir mensajes binarios, el protocolo SMTP, al estar claramente orientado al texto en plano, tenía otra dificultad que ha costado bastante superar, como es que no permitía transmitir ficheros adjuntos a un mensaje, firmas digitales, mensajes con contenido HTML o cualquiera de las cosas habituales hoy en el correo.

Todo esto hizo que se pasara a utilizar la codificación MIME en los mensajes, y es aquí cuando hablaremos de los clientes de correo electrónico (*Outlook Express*, *Netscape Mail*, etc.), ya que sin modificar el protocolo y dejando toda la responsabilidad de la codificación y posterior descodificación en el lado del cliente con MIME, se hizo posible todo lo que antes hemos comentado. El problema de descargar todo el problema en el MUA es que los en principio simples clientes de correo electrónico comenzaron a ser bastante más complejos, también en parte achacable a que comenzaron a incorporar la capacidad de escribir mensajes en formato HTML. Es por eso por lo que entre el comando

mail que en los ochenta se usaba para leer el correo y los actuales MUA existe un abismo mayor que entre las primeras versiones de un navegador y uno actual, y también que durante muchos años fuera difícil encontrar un MUA capaz de interpretar todos los tipos MIME posibles y existentes.

No describiremos aquí en profundidad el funcionamiento del MIME (que significa *Multi-Purpose Internet Mail Extensions*), pero tras su aparición en 1.993 apenas ha evolucionado[50] (sólo ha sido corregido en algunos aspectos), siendo la versión 1.0 la actual.

Se pueden indicar múltiples adjuntos (*attachments* en inglés), mediante el uso del *multipart-mixed*, mientras que si un mensaje contiene dos o más partes equivalentes pero en diferente formato (por ejemplo el mensaje en texto plano y el mismo en formato HTML) se emplearía *multipart-alternative*. Con estas y otras características cubrimos prácticamente todas las necesidades actuales.

La lucha contra el SPAM y la autenticación SMTP

El fenómeno del envío de mensajes de no solicitados de manera masiva, conocido como SPAM, es uno de los principales problemas del correo. Por SPAM nos referimos tanto al correo enviado de manera masiva sin objeto comercial concreto (UBE, del inglés *Unsolicited Bulk E-mail*) como al correo comercial que no siempre autorizamos (UCE, o *Unsolicited Commercial E-mail*). Este tipo de correo ha venido creciendo sin parar hasta la actualidad debido sobretodo a la falta de costes por mensaje transferido que tiene el correo electrónico y el diseño abierto del estándar SMTP, pensado para una época en la que Internet era algo reducido a ingenieros e investigadores que transferían poca información en cada mensaje.

Dentro de la categoría de UBE la mayoría de los correos no autorizados corresponden a mensajes destinados a confundir o estafar al destinatario, suplantando la personalidad de un tercero (simulando ser su banco para solicitarle el envío de contraseñas a una tercera dirección o a una determinada Web, por ejemplo). En esta categoría la más famosa estafa llevada a cabo es conocida como el fraude 4-1-9, y apareció en 1.994 en forma de cartas y fue posteriormente evolucionando hacia el e-mail a través de Internet. Básicamente en el mensaje que se enviaba el remitente aseguraba ser una persona próxima a un ministro del gobierno de Nigeria responsable de la *Nigerian National Petroleum Corporation* (NNPC), empresa real y tangible que en ningún momento tuvo relación con este fraude. El remitente aducía diversas razones para solicitar del destinatario su colaboración

---

[50] N. Borenstein y otros (1.993). *Multipurpose Internet Mail Extensions: Part One* (RFC 1521) y *Part Two* (RFC 1522), luego extendidos en los RFC 2045 al 2049:

<http://www.ietf.org/rfc/rfc1521.txt>

<http://www.ietf.org/rfc/rfc1522.txt>

<http://www.ietf.org/rfc/rfc2045.txt>

para transferir una cuantía elevada de dinero (entre diez y sesenta millones de dólares) a Occidente, prometiendo a cabo una sustanciosa comisión sobre el total, aunque lo que realmente acababa ocurriendo es que el incauto destinatario perdía el dinero que debía adelantar por cualquier excusa.

El fraude 4-1-9 se conoce así por ser el artículo 419 del Código Penal nigeriano (fraude o estafa) el usado para su persecución. En Junio de 1.995 un ciudadano de los EE.UU. fue asesinado en Lagos por unos estafadores cuyo método de captación fue justamente una acción similar a la comentada, usándose desde entonces este nombre para conocer al correo electrónico o postal que ha llegado a convertirse en el timo del tocomocho moderno[51].

Debido a que suplantar una dirección de correo es realmente sencillo en SMTP, o incluso el disponer de un dominio aparentemente perteneciente a un organismo oficial (**www.whitehouse.com**), no es raro que los usuarios del correo acaben por desconfiar de un sistema (el correo electrónico) ampliamente desprestigiado por el SPAM y los virus.

Históricamente siempre se había optado por autenticar únicamente la recogida del correo (es decir: solicitar usuario y contraseña o método similar únicamente si se deseaba acceder a su correo, permitiendo a cualquier persona enviar un correo a través de nuestra red sin restricción alguna). Pero con el crecimiento del *spam* se ha hecho necesario poner puertas al envío del correo, autorizando únicamente el envío de y para los usuarios locales (dado que si están conectados ya se han identificado previamente al enviar, y si está destinado a ellos aceptaremos en cualquier caso) y si son correos de y para usuarios remotos, sólo autorizaremos si se identifica como usuario del sistema.

La autenticación del cliente requiere de las extensiones SMTP. Lo habitual en la autenticación es usar de unas librerías desarrolladas por el proyecto *Cyrus* de la Universidad de Carnegie Mellon y conocidas como SASL (de *Simple Authentication and Security Layer*, **http://asg.web.cmu.edu/sasl**), que además permiten incorporar seguridad a la comunicación posterior, mediante el cifrado.

SASL en su nivel más bajo y si estamos en un sistema Unix HP, AIX, *Solaris* o Linux, acaba usando el módulo de autenticación del propio sistema, el PAM (*Plegable Authentication Module*). La función de SASL es independizar el MTA del proceso de autenticación, permitiendo los clásicos sistemas de *login* y *password*, u otros más complejos como CRAM-MD5, todo ello mediante una estructura modular ampliable.

Permite además definir un sistema propio de autenticación mediante ficheros separados, para así poder crear usuarios virtuales (que carecen de cuenta de usuario en la máquina), evitando así tener que recurrir a

---

[51] Existen múltiples páginas en Internet dedicadas al fraude 4-1-9:

Servicio Secreto de los EE.UU.: <http://www.treas.gov/usss/alert419.shtml>

Policía Metropolitana del Reino Unido: <http://www.met.police.uk/fraudalert/419.htm>

PAM. Se puede incluso autenticar contra un tercero, como un servidor LDAP[52].

Comparativa entre *sendmail* y otros MTA

Hasta ahora hemos nombrado únicamente un MTA, *Sendmail*. Existen otros muchos (tanto para Windows como para Linux), que como principal baza ofrecen una configuración más simple.

Al margen del producto de Microsoft para plataformas Windows, Exchange (**www.microsoft.com/exchange**), que ofrece como principal baza su alta integración con esta plataforma y un buen rendimiento, existe otro muy popular en esta plataforma, *MDaemon*, de *Alt-N Technologies* (**www.altn.com**). Ninguno de estos dos vamos a incluirlos en esta pequeña comparativa que a continuación haremos, porque no se han localizado estudios de rendimiento fiables ni objetivos en los que aparezcan estos dos MTA.

Por el contrario, Matthias Andree, de la Universidad de Dortmund, ha hecho una comparativa que vamos a resumir entre *sendmail* y los siguientes MTA[53]:

- *Qmail* (**http://cr.yp.to/qmail.html**), un proyecto que nació en 1.997 con la intención de reemplazar a *Sendmail*. Al igual que *Postfix* es modular. Su desarrollo original se debe a D. J. Bernstein (**http://cr.yp.to/djb.html**).
- *Postfix* (**www.postfix.org**), también conocido como *VMailer*, es un proyecto original de IBM para su plataforma AIX, que lanzó en 1.998 bajo el nombre de *IBM Secure Mailer*. El equipo de desarrollo original estaba y está dirigido por el holandés Witsie Zweitze Venaza (**http://www.porcupine.org/wietse**).
- *Exim* (**www.exim.org**) es el MTA de la Universidad de Cambridge, que ya va por su versión 4. En realidad el proyecto buscaba tanto simplificar *Sendmail* como hacer un MTA rápido y compacto, y de hecho lo es debido a que está realizado en C.

Existen además otros muchos MTA que no van a ser incorporados en la comparativa, pero que tampoco resultan relevantes por no estar muy extendidos.

La comparativa que aquí comentaremos se basó en someter en una red local a una misma máquina a entregas de correo con los diferentes MTA estudiados, analizando los tiempos utilizados por cada MTA. Aunque el estudio original contemplaba muchos casos dentro de cada MTA (por ejemplo iba jugando con el número de *threads* que había en

---

[52] Pérez Oñate, Borja y Pascual Pérez Sánchez (2.001): Autentificación de clientes SMTP mediante un servidor LDAP. Boletín 51 de RedIris.

[53] Matthias Andree (2.001), *MTA Benchmark*:

<http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/bench2.html>

ejecución listos para atender peticiones en el puerto 25), buscamos analizar las diferencias de rendimiento entre los MTA y no ver cuál es la mejor configuración de cada uno de ellos, por lo que sólo tendremos en cuenta los datos obtenidos con 20 *threads* y 1.000 correos inyectados al MTA, por tratarse de un volumen alto en el que ya se puede analizar el rendimiento del MTA en condiciones extremas.

Tabla 4-2: Estudio de Matthias Andree (http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/bench2.html)

MTA	Segundos para inyectar 1.000 correos	Segundos para entregar los 1.000 correos	Productividad (correos/segundo)
exim	56	262	3,8
postfix	49	62	16,1
qmail	138	310	3,2
sendmail	99	99	10,1
sendmail	97	97	10,3

En realidad los datos deben ser tomados con sumo cuidado y no extraer conclusiones precipitadas: este estudio no tuvo en cuenta la carga que supuso para el servidor la entrega del correo, sino únicamente el rendimiento y velocidad, que en muchas ocasiones no es relevante. En realidad *Postfix* ganará en este estudio, pero es también el MTA que más recursos de memoria consume (de ahí que al minimizar el uso del disco obtenga un mayor rendimiento), debido también a su estructura modular.

Pero de cara al usuario que *Postfix* sólo tardara 49 segundos en aceptar los 1.000 correos es un factor importante: el usuario advertirá una mayor velocidad aparente, de ahí que pueda resultar claramente interesante utilizar *Postfix* como MTA.

4.3.3 Servidores Web

Si nos remontamos a 1991, cuando Tim Berners-Lee puso la primera piedra de los tres estándares que el Web requiere: el protocolo HTTP, el lenguaje HTML y los URL y su estructura actual, veremos que en esos momentos iniciales ya tuvo que haber dos proyectos que dieran lugar a dos programas diferentes: el servidor Web por un lado, y los clientes que se conectaran a él y pudieran descargarse los contenidos. Hablaremos primero de los servidores Web.

El NCSA y el MIT fueron las dos instituciones estadounidenses que se unieron de manera clara en 1.991 al CERN para completar el desarrollo de los estándares de la Web que acabamos de comentar (aunque la regulación y evolución del HTML y las URL sea hoy cosa del ISOC), y que de hecho fueron reemplazando en esta tarea al CERN, que pronto perdería a Tim e iría abandonando el proyecto.

Fue el NCSA el que dio un impulso al Web creando ese mismo 1.991 el servidor Web más famoso de la historia y que aún hoy día sigue dominando el mercado a través de sus evoluciones: el NCSA Web Server. Del equipo desarrollador de este software su cabeza visible era Rob McCool.

A mediados de 1.994 Rob abandonó la NCSA. A partir de ese momento el proyecto de desarrollo del servidor HTTP de la NCSA se estancó debido a la falta de interés de la NCSA por continuar el proyecto, y comenzaron a quedar sin resolver algunos *bugs* del software.

Esta circunstancia obligó a que muchos administradores de sitios que utilizaban aquella aplicación tuvieran que desarrollar sus propias extensiones y corregir de forma individual los fallos de funcionamiento de la aplicación original.

Un pequeño grupo de aquellos administradores se unió con objetivo de coordinar y unificar sus trabajos de corrección y mejora de la aplicación original de NCSA. Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores en un ordenador instalado en California.

En febrero de 1995, ocho colaboradores de este proyecto decidieron organizarse y fundaron lo que fue conocido como Grupo Apache (**www.apache.org**), que es hoy día una fundación que desarrolla multitud de proyectos, todos ellos e incluyendo al propio servidor Web Apache, bajo la licencia GNU.

El nombre de la fundación y del proyecto (Apache), viene de un acrónimo de *A PAtCHEd web server*, que es en definitiva lo que estaban desarrollando, un conjunto de parches para el servidor NCSA original. Los ocho colabores miembros fundadores eran: Brian Behlendorf, Roy

T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau y Andrew Wilson.

Usando como base el HTTPD 1.3 de NCSA y aplicando los *patchs* desarrollados hasta el momento, lanzaron la primera versión oficial (0.6.2) del software de servidor de Apache en abril de 1995.

De hecho si NCSA Web Server continuó luego evolucionando fue a base de seguir los parches que el proyecto Apache fue creando, siendo la versión más reciente del NCSA *Web server* la 1.5.2a de Septiembre de 1.996 (**ftp://ftp.ncsa.uiuc.edu/Web/httpd/Unix/ncsa_httpd**).

Aquella primera versión -y sus sucesivas evoluciones y mejoras- alcanzaron una gran implantación como software de servidor -inicialmente solo para sistemas operativos UNIX- y se han convertido de facto en el servidor Web más implantado en la actualidad en cualquier plataforma (si exceptuamos el dominio de Microsoft y su IIS en las plataformas Windows).

Manteniendo el espíritu original del producto de la NCSA, Apache siempre se ha distinguido por una clara orientación modular que le ha permitido seguir siendo el preferido, aún incluso cuando el cliente *NCSA Mosaic* hace ya muchos años que dejó de existir.

Si hay que destacar alguno de los módulos disponibles (que de hecho son muchos), este sería el que permite activar la capa SSL, totalmente gratuito, mientras que cualquier otro servidor Web requiere ser adquirido por separado. SSL y el protocolo HTTPS es hoy día la forma más adecuada de cifrar las comunicaciones entre cliente y servidor, siendo por lo tanto usado de manera habitual en aquellas transacciones en las que se busca seguridad (otros protocolos como la iniciativa SET –de *Secure Electronic Transactions*– de VISA y MasterCard han quedado en el camino). Otra ventaja añadida de Apache es que dado que se trata de un proyecto GNU de código fuente accesible, se puede compilar, adaptar e instalar en cualquier plataforma, y eso ha llevado a que existan versiones del mismo en todos los sistemas y plataformas posibles.

Alternativas al software GNU comentado las hay en el mundo de los servidores Web, con lo que enumeraremos algunas de ellas:

- El Servidor de *Netscape* fue uno de los más importantes durante los años 96 al 98, debido a que también dominaba el mercado de los navegadores Web. Sigue siendo comercializado, pero hoy día no supone apenas el 1% de la cuota de mercado. Está disponible para multitud de plataformas, como SunOS, Solaris, Digital, OSF, HP-UX, IRIX, BSDI, AIX, aparte de plataformas Microsoft. Existe otro producto similar llamado Servidor para Comercio que incorpora la capa SSL.
- Microsoft es la única compañía que tiene un servidor Web que hoy por hoy hace sombra a Apache, con su *Internet Information Server.* Aunque su cuota de mercado sigue sin ser

muy alta siquiera en las plataformas Windows, su total integración con las plataformas Windows lo hacen ser el aconsejable en la actualidad si deseamos basar nuestros sistemas en arquitecturas Intel/Microsoft.

- *WebSite* era un desarrollo de O'Reilly & Associates, para plataformas Windows NT y Windows 95 que no pasa de ser un producto experimental y con claros tintes educativos. De cualquier forma su calidad es aceptable.
- *MacHTTP* es lo único que puede encontrarse en la actualidad para plataformas Macintosh. Su fabricante es BIAP Systems (**www.biap.com**), pero no tiene posibilidad de ser un futuro referente en cuanto a servidores Web dado que Macintosh dejó de ser hace años una arquitectura adecuada para montar servidores.
- SCO Unix también dispone de su propio servidor Web, aunque es una plataforma Unix en la que Apache funciona de manera óptima. En este caso el fabricante, SCO Unix (**www.sco.com**), desarrolló Global Access Supplement, un servidor Web con capa SSL incluida.

Por otro lado habíamos comentado la existencia de dos proyectos de software cuando se creó el Web. El segundo proyecto que hemos de abordar ahora es el navegador Web, el cliente capaz de conectar al servidor Web que el usuario requiera y descargarse desde él las páginas HTML que interpretará y mostrará al usuario.

Anteriormente también, habíamos hablado ya de Tim Berners-Lee y su primer navegador, pero el navegador por excelencia sería el que en el NCSA (**www.ncsa.uiuc.edu**) desarrollaría en 1.993, en un equipo en el que la cabeza visible del mismo era Marc Andreesen: *NCSA Mosaic*. La primera beta de este cliente Web salió en febrero de 1.993, aunque la versión 1.0 definitiva se retardó un poco más, hasta septiembre de ese mismo año, apareciendo simultáneamente para plataformas Windows, Macintosh y Unix/X-Windows.

En marzo del año siguiente Marc abandonaría el NCSA (que como ente público no compartía su visión comercial del futuro de un software como aquel) y junto a Jim Clark y gran parte del equipo de desarrollo del *Mosaic*, formaron Netscape Inc (**www.netscape.com**), que ni que decir cabe que tuvo como primer producto un navegador, el Netscape Navigator, el cual llegó en sus mejores momentos con la versión 3 a estar instalado en 9 de cada 10 equipos que navegaban por la Web.

Cuando en 1.997 Microsoft descubrió que estaba perdiendo el tren de Internet, una de sus primeras preocupaciones fue hacerse un sitio tanto entre los navegadores como entre los servidores. El servidor Web de Microsoft ya hemos comentado que es hoy día el segundo en implantación, aunque a una clara distancia de Apache. En cambio, por lo que respecta a los clientes Web, hoy todos los equipos de plataformas Windows usan el navegador Web de Microsoft, debido

fundamentalmente a su estrategia de regalar el navegador Internet Explorer junto a su sistema operativo. El como Microsoft logró darle la vuelta al mercado de los navegadores es una historia que se conoce como la "guerra de los navegadores", en la que ha habido procesos judiciales aún no cerrados del todo, debido al aparente abuso de posición dominante que siempre ha ejercido Microsoft. De cualquier forma la compañía de Redmond ha pagado un alto precio, y en dólares: para desplazar a los demás tuvo que comenzar comprando un navegador lo suficientemente avanzado como para no tener que partir de cero, y *NCSA Mosaic* fue el elegido, y a partir de ese momento se dedicaron ingentes recursos humanos a mejorarlo. La comentada guerra también tuvo por víctimas a los estándares existentes, ya que una manera de diferenciarse que Microsoft utilizó fue enriquecer las características que HTML 3.0 o bien 4.0 ofrece, creándose diferencias en cuanto al comportamiento que no siguen el estándar y por tanto perjudican a los usuarios.

Del navegador de Microsoft existen ejecutables tanto para plataformas Windows como para Macintosh, aunque ni que decir tiene que no existen ni existirán para otros sistemas que Microsoft considera rivales. Existen igualmente otros navegadores con menor cuota de mercado, que ahora nombraremos brevemente:

Amaya (**www.w3.org/pub/WWW/Amaya**) es un navegador bastante simple pero que al estar desarrollado por el propio consorcio encargado del protocolo HTML, el *W3 Consortium*, tiene la ventaja de cumplir a rajatabla el estándar de la actual versión del lenguaje HTML, la 4.0. Existe para plataformas Unix o Windows.

Opera (**www.operasoftware.com**) es una de las pocas alternativas comerciales serias que quedan en la actualidad, siendo el único que sigue buscando ingresos a través de la venta de la licencia de uso del mismo, aunque se licencia también a cambio de soportar publicidad en el cliente. Para plataformas Windows.

Mozilla es por último la evolución sufrida por Netscape y su *Netscape Navigator* tras el envite sufrido por Microsoft: para el desarrollo de la versión 6.0 del mismo, el fabricante se planteó crear una licencia que le permitiera aprovechar las ventajas del GNU y no perder el control del proyecto. Con el código de Netscape ocurrió que se formaron equipos de desarrollo externos a Netscape y de dentro de la comunidad de desarrollo GNU que mejoraron el navegador y continúan mejorándolo, habiéndose creado una versión nueva, *Mozilla*, que Netscape luego ha vuelto a modificar para cambiar ligeramente la interfaz y venderlo como la siguiente versión de su *Netscape Communicator*.

Ya dentro del software GNU, nos quedaría por último comentar la existencia de dos navegadores en cada uno de los entornos gráficos que más han triunfado en plataformas Unix, como son el navegador *Konqueror* dentro de KDE, y el *Galeon* en el caso del *Gnome*.

Si bien HTML es igual de importante que el protocolo HTTP, únicamente comentaremos el segundo para tratar de comprender el diálogo que se establece entre servidor Web y navegador a través del puerto 80. La versión actualmente en uso del HTTP es la 1.1, de 1.999[54]. Por tanto hablaremos también de URI en lugar de URL: mientras URL viene de *Universal Resource Locutor* y es la forma original de referirse a un recurso alojado en una localización electrónica remota, el URI (*Universal Resource Identifier*) ha sido establecido como una entidad superior de la que es subconjunto el URL, creándose así una especificación más global que es que hoy día se utiliza.

A diferencia de lo que ocurría con el servidor de correo, no se establece un verdadero diálogo en el puerto 80 entre servidor y cliente, sino que el servidor espera que el navegador introduzca la solicitud y todas las opciones pertinentes antes de devolverle el código de estado (respuesta que según su primer dígito supondrá un error si comienza por dos, etc.) y la página en sí solicitada. Para cada una de las partes que comprenden la página Web (imágenes, etc.) habrá de establecer un diálogo idéntico pero independiente.

La solicitud de una página deberá contener alguno de los siguientes métodos:

- GET y POST son los más usuales: sirven para solicitar una página del servidor, diferenciándose en la manera en la que se le pasan los valores (a través de la URL en el caso del GET).
- HEAD para solicitar únicamente la cabecera, es decir, el valor del código de estado y las cabeceras, usado habitualmente para verificar si ésta ha cambiado en los servidores *proxy* a través de la cabecera *Age*.
- PUT para subir información al servidor en lugar de obtenerla de él, en una determinada URI.
- DELETE para borrar un recurso URI del servidor.
- Y otros menos usados, como TRACE, CONNECT y OPTIONS.

Aparte de esta solicitud, que se colocará a continuación de la versión del protocolo que se va a usar (por ejemplo: *HTTP/1.1 GET xxx*) existen una serie de modificadores que habrían de ir en las siguientes líneas. El servidor sólo responderá cuando se encuentre una línea en blanco, momento en el que supondrá que hemos terminado. Hay que tener en cuenta que también el servidor nos responderá usando alguna de estas cabeceras antes de comenzar a devolvernos el documento.

---

[54] R. Fielding y otros (1.999): *Hypertext Transfer Protocol* – HTTP/1.1, que deja obsoleto el RFC 2068 (RFC 2016):

<http://www.ietf.org/rfc/rfc2016.txt>

<http://www.ietf.org/rfc/rfc2017.txt>

Existen hasta 47 cabeceras, entre las que destacaremos algunas:

- *Accept* (y todas las variantes de ésta, como *Accept-Language*, *Accept-Ranges*, etc.), que sirven para indicar al servidor los tipos MIME aceptados por el navegador cliente.
- *Age*: usada por el servidor para indicarnos la fecha de modificación, y usada en los *proxys* para determinar si ha de recuperar de nuevo el URI completo.
- *Host*: es de hecho la cabecera más relevante con diferencia, sobretodo desde que es común que un servidor Web contenga más de una página Web con múltiples dominios. Antes había que disponer de una IP por cada dominio a alojar, dado que en el momento de la conexión al servidor, el servidor desconoce a qué sitio Web desean conectarse, pero dado que existe la cabecera Host con el nombre del dominio solicitado, Apache incorporó la capacidad del servidor para alojar múltiples sitios Web haciendo que el servidor tuviera en cuenta esta cabecera al recibir una petición.
- *Referer*: es común su uso para hacer un seguimiento del sitio Web desde el que nos llegan al nuestro los clientes, ya que contiene el URI del sitio Web que nos estaba referenciado y a través del cuál han llegado hasta nosotros tras pinchar en un link.
- *WWW-Authenticate*: en conjunción con el valor de retorno 401 por el servidor, indican al navegador que ha de solicitar del cliente un usuario y contraseña que utilizará al pedir por segunda vez el URI solicitado.

Los códigos de estado devueltos por el servidor son muchos. Están formados por tres dígitos, el primero de los cuales identifica al grupo e indica la acción a tomar sobre esa respuesta:

- 1XX: en teoría son códigos de respuesta provisional, y desde HTTP/1.0 no se han vuelto a usar, incluso allí lo fueron con fines experimentales. De hecho sólo se definieron 2, el 100 (que indica que el cliente puede continuar) y el 101.
- 2XX: la petición es correcta y va a ser atendida.
    - 200 OK: es la más habitual.
    - 201 *Created* y 202 *Accepted*, son respuestas habituales ante un método PUT.
    - 203 *Non-Authoritative Information*
    - 204 *No Content*
    - 205 *Reset Content*
    - 206 *Partial Content*

- 3XX: error en la petición realizada subsanable (de hecho muchas veces lo corrige el propio navegador y realiza de nuevo la petición sin esperar respuesta del usuario): la petición ha de ser reformulada con un URI diferente:
    - 300 *Multiple Choices*
    - 301 *Moved Permanently*: se usa muy a menudo para hacer una redirección del cliente hacia otro sitio Web, cuyo nuevo URI se le indica en esta respuesta.
    - 302 *Found*: se suministra un URI temporal en el que se puede encontrar el recurso solicitado (es decir, se ha trasladado temporalmente).
    - 303 *See Other*: el recurso está en otra URI y debe ser vuelto a solicitar usando un GET.
    - 304 *Not Modified*: el contenido que se acaba de solicitar no ha cambiado, y por tanto no se le reenvía.
    - 305 *Use Proxy*
    - 307 *Temporary Redirect*
- 4XX: error en la petición no subsanable y que se ha de mostrar al usuario (o que requiere de datos extra, como es el caso del 401, donde se solicitará del cliente un usuario para permitirle el acceso):
    - 400 *Bad Request*
    - 401 *Unauthorized*
    - 402 *Payment Required*
    - 403 *Forbidden*: recurso no accesible por falta de permisos.
    - 404 *Not Found*: el recurso solicitado no existe.
    - 405 *Method Not Allowed*
    - 406 *Not Acceptable*
    - 407 *Proxy Authentication Required*
    - 408 *Request Timeout*
    - 409 *Conflict*
    - 410 *Gone*: el recurso solicitado estuvo en algún momento pero se ha trasladado y no se suministra URI alternativo alguno. De hecho no se emplea, y en su lugar es más común suministrar un error 404.
    - 411 *Length Required*
    - 412 *Precondition Failed*
    - 413 *Request Entity Too Large*

    - 414 *Request-URI Too Long*
    - 415 *Unsupported Media Type*
    - 416 *Requested Range Not Satisfiable*
    - 417 *Expectation Failed*
- 5XX: error en el servidor que impide siquiera atender la petición.
    - 500 *Internal Server Error*
    - 501 *Not Implemented*
    - 502 *Bad Gateway*
    - 503 *Service Unavailable*
    - 504 *Gateway Timeout*
    - 505 *HTTP Version Not Supported*

Extensiones del servidor Web: PHP

El PHP –acrónimo de *Hypertext Preprocessor*– suele definirse como un lenguaje interpretado, de alto nivel, cuyo código va insertado en páginas HTML y que es ejecutado en el servidor antes de responder a la petición, por lo que estamos ante un lenguaje embebido.

Es un lenguaje orientado a objetos pero de sintaxis simple y muy similar a C++. La gran diferencia con los otros lenguajes –C++ o *Java*– es que nació expresamente para ser parte de un servidor Web, con un tratamiento de cadenas y HTML es potente y simple. Dispone de muchos módulos y una interacción muy elevada con los sistemas *Unix*, además de que por la cantidad de módulos de que dispone es el sistema indicado en la actualidad si queremos desarrollar contenidos dinámicos y que usen bases de datos en plataformas no Windows (donde siempre estaría la opción de recurrir al código ASP de Microsoft).

4.4 Los ataques de denegación de servicio en Internet

La seguridad en Internet (su ausencia, en muchos casos) es uno de los temas que siempre han preocupado más a sus posibles usuarios y han retardado el crecimiento del comercio electrónico. Es por eso que el análisis y correcto diseño de las estrategias adecuadas para lograr la seguridad es fundamental para cualquier compañía con presencia en Internet.

Aunque el espectro de posibles ataques es muy amplio, vamos de momento a comentar únicamente los específicos de Internet y que más afectan al ISP, que son aquellos que están dirigidos contra nuestra presencia en Internet, es decir, que tienen como objetivo dejarnos fuera de la red.

Un servidor Web o e correo, al estar en Internet estará expuesto a ataques de denegación de servicio, encaminados a impedir el acceso a los usuarios legítimos del servicio, bien dañando la máquina o su software que ofrece los servicios, bien colapsándolo, deteniéndolo o cualquier método alternativo. Se trata de un ataque contra los recursos disponibles: ancho de banda, memoria, espacio en disco, tiempo de CPU o incluso energía si el ataque se produce durante un corte en el suministro eléctrico.

No contemplaremos de manera específica otros ataques igualmente graves pero que difícilmente se podrían dar, como serían la alteración de datos o la destrucción física del equipamiento (estos ataques los analizaremos en el apartado dedicado a los ataques genéricos).

En este tipo de ataques incluiremos también aquellos usos ilegítimos del servicio por parte de nuestros propios usuarios, como la existencia de ficheros de gran tamaño y la descarga abusiva de los mismos por parte de terceros, que podrían llevarnos a colapsar nuestro ancho de banda.

En todos estos casos, el impacto es tremendo si no se han tomado las medidas pertinentes.

4.4.1 Ataque *SYN flood*

Este ataque contra los recursos de memoria y CPU del servidor se aprovecha del tratamiento de las conexiones que se realiza en TCP/IP: las conexiones nuevas tienen una conexión inicial que a través de la negociación de tres vías aseguran que ambas partes van a poder a partir de ese momento comunicarse una vez han sincronizado unos números de secuencia para cada sentido del tráfico. Los atacantes en este caso se aprovecharían de esta característica y obligarían al servidor víctima a ir estableciendo nuevas conexiones que el servidor responde en vano y para cada una de las cuales el servidor tendrá que asignar recursos para gestionarlas.

Para que el servidor responda en vano lo más común es que el atacante emplee técnicas de *spoofing* que le permitirán insertar en el paquete TCP/IP una dirección IP que no es válida ni es la suya realmente, con lo que la respuesta SYN ACK del servidor va a ser en vano (destino inalcanzable) y encima va a ocupar recursos durante los 75 segundos que estará el temporizador de conexión antes de que se elimine dicha entrada por no encontrar la misma respuesta.

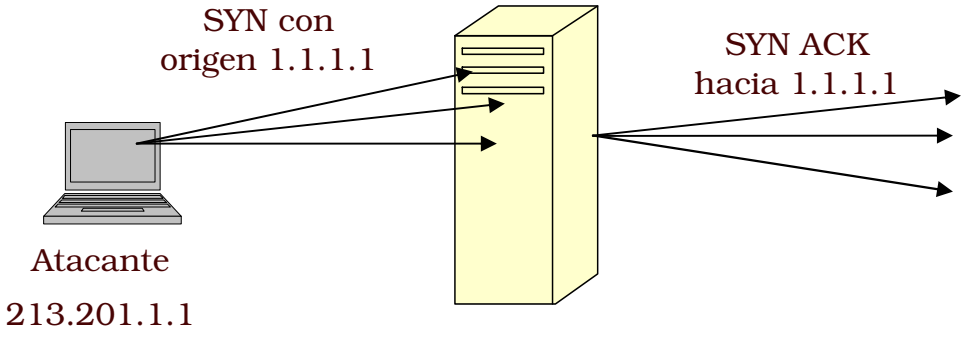


Ilustración 4-17: Esquema de un ataque *SYN flood*

El primer ataque de estas características es bastante antiguo: data de Septiembre de 1996, cuando el servicio de correo de *Panix*, un ISP del estado de Nueva York fue tumbado desde un simple módem RTB. El problema viene causado por el tamaño del *backlog* del servidor: la pila TCP/IP implementa una estructura en memoria en la que se guardan las peticiones de conexión no atendidas por cada *socket* TCP, entendiéndose como *backlog* el número de entradas de esa estructura. En concreto, en los sistemas más presentes en Internet, cuando se comenzó a dar este tipo de ataques estos valores eran bastante bajos.

Tabla 4-3: Entradas del *backlog* en los sistemas operativos más comunes

	entradas backlog	*gracia (exceso permitido sobre el backlog)*
SunOS 4.x.x e IRIX 5.2	*5*	*3*
Linux 1.2.x	*10*	*0*
FreeBSD 2.1.5	*12*	*8*
Windows NT 4.0	*6*	*0*

No por eso podemos pensar que se escogieron valores bajos de manera ilógica, sino que se hizo y se hace de hecho así por economía de recursos: de hecho en un sistema *Unix* existe también un límite de 10 descriptores de fichero por proceso y es suficiente. Dado que existe un *backlog* por cada *socket*, modificar este valor supondría disparar las necesidades de memoria en un sistema de una manera exponencial, además de retardar las respuestas.

Es justamente por la rapidez con la que un servidor atiende las peticiones de conexión por lo que este ataque sólo funciona si se une al uso del *spoofing*: sólo cuando el cliente no responde a la trama SYN ACK se produce una ocupación efectiva de la entrada en la tabla *backlog* del servidor, durante los 75 segundos que estará esperando el servidor a que el supuesto cliente que conectó le responda.

En situaciones normales, si un cliente no puede establecer una conexión, es seguro que al retransmitir el cliente la petición de conexión ya tendremos espacio en la cola de *backlog* para atenderla, ya que el proceso de conexión es rápido. El problema viene durante un ataque, cuando de manera ininterrumpida están entrando nuevas peticiones de conexión, encontrándose lleno el *backlog* también para las conexiones legítimas.

De manera numérica el número de entradas n ocupadas en el *backlog* si cada a segundos se produce una petición de conexión nueva vendría dado en función del instante t a partir del ataque por:

$$n = \min\left(\left\lfloor \frac{t}{a} \right\rfloor, \left\lfloor \frac{75}{a} \right\rfloor\right)$$

Ecuación 4-1

Suponiendo 75 como el valor del temporizador que haría que se eliminara esa entrada del *backlog* al no obtener respuesta, tendríamos que bastará con generar 8 peticiones de conexión por minuto con IP incorrecta para tumbar un servicio concreto (si es 10 el tamaño de esa tabla y 75 el temporizador, con 8 peticiones tendríamos una cada 7.5 segundos).

En el límite tendríamos 7.5 peticiones por minuto (una petición cada ocho segundos), valor que nos llevaría a que sólo una entrada de *backlog* estaría disponible para un cliente real:

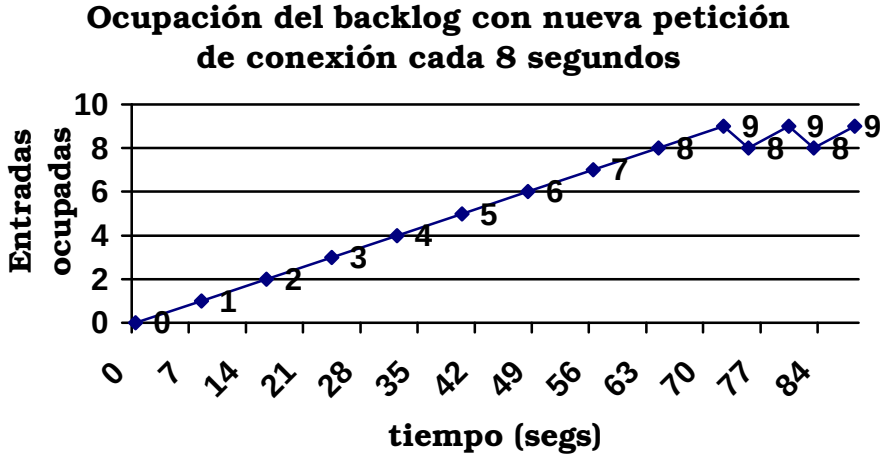


Ilustración 4-18: Saturación del *backlog* en un ataque *SYN flood*

En definitiva, se trata de un problema ya conocido hacía tiempo y de difícil tratamiento por cuanto afecta al diseño del protocolo[55]: se puede aumentar el tamaño del *backlog*, pero segaríamos expuestos a que incrementando el número de conexiones lograran su objetivo, o bien podríamos reducir el *timeout* de las conexiones, con el riesgo de que se perdieran alguna de las mismas.

Las soluciones más realistas son aquellas en las que los routers que llevan el tráfico hasta nuestro servidor filtran los ataques de *spoofing* (es decir, eliminan el tráfico cuya IP de origen no se encuentre en un rango válido). Esta solución es más efectiva si los routers que realizan el filtrado son los intermedios, en los que se puede saber de manera efectiva si la IP del paquete es la correcta, analizando si está llegándonos a través de la ruta correcta (por cuanto estos routers disponen de diferentes conexiones).

Pero la solución más eficiente y que podemos implementar en nuestros servidores pasa por el uso de las *SYN cookies*, una característica ya disponible en los últimos *kernels* de Linux pero que no viene activada por defecto.

En realidad de lo que se trata es de trasladar la asignación de recursos a un punto en el que no sea susceptible de ataque: la entrada se creará tras la respuesta al SYN ACK que cerrará la negociación de tres vías TCP y que por tanto nos asegurará que existe un cliente que realmente desea establecer una conexión, evitándose crear un recurso antes de que se cree del todo la conexión.

Para ello, el servidor generará una respuesta SYN ACK en la que su número de reconocimiento inicial contenga información suficiente para que cuando nos lo devuelva un cliente real podamos reconocerlo, es decir. Hasta ahora la necesidad del *backlog* nos la daba que en la negociación de tres vías, el número de reconocimiento que nos

[55] Garfinkel y Spafford (1.991): *Practical UNIX and Internet Security*, problema del tamaño del backlog tratado en la Pág. 778.

devolvería el cliente en la tercera fase, sería nuestro número incrementado en una unidad, con lo que necesitábamos guardarnos ese dato en algún lugar. Pero con esta técnica el valor de SYN que dará el servidor será un número generado por una función que sólo el servidor conoce, con lo que se podrá comprobar si fue generado por el cliente en la respuesta que nos dé posteriormente, al mismo tiempo que eliminamos esa molesta cola que consume recursos. *Eric Schenk* lo implementó para Linux en febrero de 1997, al calor del comentado ataque sufrido por *Panix*, con lo que resulta muy fácil defenderse de este tipo de ataques.

Esta técnica ha sido criticada una y otra vez, en primer lugar porque ofrece un nuevo campo de ataque, como sería el intento de un *hacker* de predecir qué números de secuencia serán generados por el servidor en un determinado momento, facultándole una vez más para establecer conexiones saltándose ahora incluso la negociación de tres vías.

Para evitarlo, la generación de ese número de secuencia de 32 bits se compone de una parte (los primeros 5 bits) cuyo valor es el temporizador del servidor que se incrementa cada 64 segundos módulo 32, más otra parte (los últimos 24) que combinan la IP del cliente y su puerto origen de la conexión. El modelo ha sido lo bastante estudiado como asegurar que está a salvo de criptoanálisis hasta un nivel razonable.

Pero también ha recibido ataques más duros, en los que se acusa a este sistema de incumplir el protocolo TCP, cosa que no es cierta: incluso el número de secuencia que ahora se genera sigue cumpliendo que crece poco en el tiempo (uno de los requisitos que se exigen en la implementación de las pilas TCP/IP).

4.4.2 Otros ataques de denegación

Ataques contra el espacio en disco

Cualquier servicio que escriba datos en disco es susceptible de servir para este tipo de ataques. Es particularmente peligroso aquello que genera cualquier tipo de *log* en el sistema, ya que por tratarse de texto su tamaño resulta mayor. La solución en este caso parece evidente: disponer de una gran cantidad de espacio de almacenamiento en disco, pero esto no es de por sí suficiente, porque nunca va ser infinito y en un determinado momento nuestro sistema puede quedarse igualmente sin espacio. De hecho el sistema operativo ya toma algunas medidas al respecto: en Linux el demonio responsable de los *logs*, el demonio *syslogd*, ya se encarga de retardar y analizar las líneas a insertar en los ficheros de *log* para detectar las repeticiones consecutivas en el mismo y únicamente informar de cuántas veces se repite el último mensaje (*"last message repeated X times"*), en lugar de escribir esos caracteres de nuevo.

Por nuestra parte, en primer lugar los programas en ejecución habrán de evitar usar de manera alegre el espacio en disco: grabar los

datos una vez confirmadas las transacciones o compras, y no informar de hechos redundantes, pero esto no basta. Lo interesante sería analizar formas de evitar que un cierto *log* o proceso en ejecución sature el espacio en disco: resultará para ello aconsejable que todos los servicios estén asignados a usuarios con cuota en disco asignada, lo que implica ya de entrada descartar el que haya servicios que usen al *root* como usuario para ejecutarse.

Aunque resulte tentador, hay que evitar como alternativa el que cada directorio sensible de ofrecer problemas (*/log*, */var*...) sea montado en una partición o disco diferente: el control que ofrece esta característica es el mismo que el uso de cuotas, pero nos puede llevar a situaciones desagradables si en un determinado momento necesitamos ampliar el tamaño de estas particiones.

Redirección ICMP

Existe entre los diferentes tipos de mensajes ICMP uno que permite indicar al host que lo recibe que existe una ruta más corta para determinado destino, circunstancia que puede ser aprovechada para tratar de desviar el tráfico entre dos hosts por un tercer punto en el que el atacante pudiera interceptarlo y analizarlo.

Dado que no es muy común actuar de *carriers* de Internet, y como de hecho hoy día la redirección ICMP ha quedado obsoleta como método de encaminamiento dinámico, lo común es filtrar este tipo de tráfico o hacer que sea ignorada esta petición de redirección, con lo que tampoco tendríamos grandes problemas. Aún en el caso de dispongamos de diferentes conexiones y podamos o necesitemos desviar el tráfico entre ellas, existen protocolos de enrutamiento lo suficientemente potentes como para no necesitar este tipo de mensajes.

El ping de la muerte y ataque *smurf*

El ping de la muerte se produce cuando nuestro servidor recibe grandes cantidades de tráfico ICMP y no puede dedicar sus recursos más que a responder estas peticiones de *echo*. Hay que tener en cuenta además que las tramas ICMP pueden tener un tamaño considerable, siendo factible un relleno de 64 *Kb*. en la trama que acabe por saturar nuestra conexión.

El otro tipo de ataque, similar al anterior, es el producido por el envío de tráfico ICMP en el que la dirección destino es la dirección *broadcast* de nuestra red, logrando así que cualquiera de los hosts de esa red le respondan teóricamente. Y remarco lo de teóricamente porque hoy día ya no es habitual esta situación, pero aunque así fuera, lo que haremos será evitar directamente cualquier solicitud de *ping*.

De esta manera perdemos herramientas que igualmente nos podrían ser útiles en un futuro para detectar problemas de configuración o funcionamiento de los servicios, pero será preferible tener que parar en esas circunstancias el *firewall* que tener continuamente la red expuesta a *pings* malintencionados.

5 Análisis

La compañía objeto de este estudio, al igual que la inmensa mayoría de los ISP que quedan en España de tamaño medio y pequeño, está abocada a perder el negocio del acceso a Internet por la presión ejercida por los grandes operadores. Muchos ISP van a optar por mantenerse en la otra actividad que más o menos ya venían llevando: el diseño Web y el alojamiento, dónde los grandes operadoras tienen más complicado barrer a las compañías más pequeñas. Se trata de una circunstancia muy común en las empresas que usan tecnología avanzada: a mayor tamaño de la empresa, mayores economías de escala y posibilidad de reducción del precio.

Pero también en el diseño Web y en el alojamiento existe la suficiente competencia como para que para la compañía objeto del estudio deba aprovechar la oportunidad y buscar diferenciarse de los demás para asegurar su supervivencia en el mercado. El proyecto trata únicamente del alojamiento y de la configuración de los sistemas y red de acceso necesaria para ello, descartando el diseño Web (que no sería objeto de estudio en un proyecto informático). Aquellos ISP que se dedican en exclusiva al alojamiento Web y servicios de correo electrónico se les conoce como IPP, de *Internet Presence Provider*, acrónimo que usaremos indistintamente al de ISP para referirnos a la compañía en estudio.

La diferenciación que esta compañía ofrecerá será el servicio personalizado, pero al mismo tiempo y para no disparar los costes, hemos de asegurar que el personal de la empresa no tiene que perder gran parte de su tiempo en tareas repetitivas y que se pueden perfectamente automatizar, para lo que se habrá de reducir el uso de la atención telefónica (un capítulo importante de los gastos) y diseñar los servicios para que sean configurables por el usuario directamente a través de un panel de control.

De esta forma se busca evitar que personal de la empresa haya que realizar manualmente dicha configuración (implicando esto tanto el registro de los dominios, como la configuración de las zonas DNS, gestión de correos, y otros muchos pasos repetitivos). Esta automatización permitirá además que las altas de los servicios se puedan producir en cualquier momento del día, sin estar restringida a las horas laborales de la empresa, como ocurriría si una simple reserva de un dominio se hubiera de realizar a mano.

Separaremos en el análisis las dos áreas a trabajar: el proyecto de software necesario para automatizar los servicios, y por el otro la configuración de los sistemas.

Para el proyecto de software necesitaremos poder identificar correctamente todos los elementos que habrá que desarrollar e implementar usamos el modelado UML, modelado que no será necesario por lo que respecta al otro segmento del proyecto, el relacionado con la configuración de los sistemas.

UML es un lenguaje de modelado visual[56] usado para especificar, construir y documentar el desarrollo de software, desde las fases iniciales hasta la implementación. El uso del modelado UML se hará presente también posteriormente en la fase de diseño e implementación. Usaremos diagramas de casos de uso para obtener los requisitos a satisfacer, e identificaremos los elementos a desarrollar a partir de los diagramas de colaboración y de clases en el modelo de análisis que se mostrará.

Para analizar el diseño y configuración de los sistemas y el acceso a Internet no usaremos modelos tan avanzados y confiaremos más bien en unas estimaciones menos precisas, debido fundamentalmente a que no van a consumir tanto tiempo ni recursos como el proyecto software.

Una vez identificados estos elementos del proyecto, se determinarán los costes mediante el modelo de COCOMO, así como se especificará una agenda de proyecto y los recursos que intervendrán (fundamentalmente humanos por lo que corresponde al proyecto software). Como conclusión del análisis se mostrará un resumen de costes y el diagrama de Gantt que muestre la distribución temporal de estos y de los recursos.

[56] El Object Management Group (OMG, www.omg.org) se creó en 1.989 por once compañías para la estandarización de la programación orientada a objetos, y es el responsable de UML. Para más información al respecto:

<http://www.uml.org>

5.1 Requisitos a satisfacer

5.1.1 El panel de control

Se busca hacer una interfaz a la que el usuario accederá dónde sea autosuficiente para solicitar y configurar la mayoría de los servicios por si mismo, descargando al personal del ISP de estas tareas y concentrándolo en resolución de incidencias.

La interfaz deberá ser un panel de control en el que las acciones posibles estarán restringidas a aquellas en las que intervengan los dominios y servicios adquiridos por el cliente, quedando ocultos el resto de usuarios y servicios de los mismos, por lo que la seguridad de este panel habrá de ser elevada.

El panel de control integrará también un área de gestión de los datos del cliente, para que acciones tan simples como un cambio de cuenta corriente no tengan que realizarse vía telefónica.

Además incluiremos en lo que llamamos panel de control la interfaz con la facturación de este panel de control y también la interfaz con el personal técnico de la empresa, que ha de mantener la capacidad de control sobre las mismas tareas que los clientes podrán realizar a través de la Web.

Este panel de control deberá ser desarrollado, ya que satisface unos requisitos que serán la diferenciación de nuestro ISP, y para el que por tanto no nos sirven otros proyectos similares que pudiera haber. Este bloque será además el que requerirá del uso del modelado UML, tanto en su análisis como posteriormente en el diseño e implementación.

5.1.2 Los sistemas a configurar y diseño de la Red

Por el contrario el resto de tareas contempladas (diseño de la red, y la selección y configuración de los sistemas de nuestro ISP, así como la integración final y la fase de pruebas) se planificarán con un enfoque distinto al no existir programación de software alguna. Se tratará por tanto de una parte del proyecto en el que, al no haber código a programar, el peso mayor se lo llevará la fase de diseño.

Las actividades necesarias para estas dos tareas serán tres: una de diseño, una segunda de implantación, y una fase final de prueba. Estas actividades serán previas al proyecto de desarrollo del panel de control, al considerase útil que el servidor anfitrión usado durante el desarrollo del proyecto tenga la misma configuración que se desea posteriormente usar como servidor definitivo.

5.1.3 Servicios que se considerarán en el proyecto

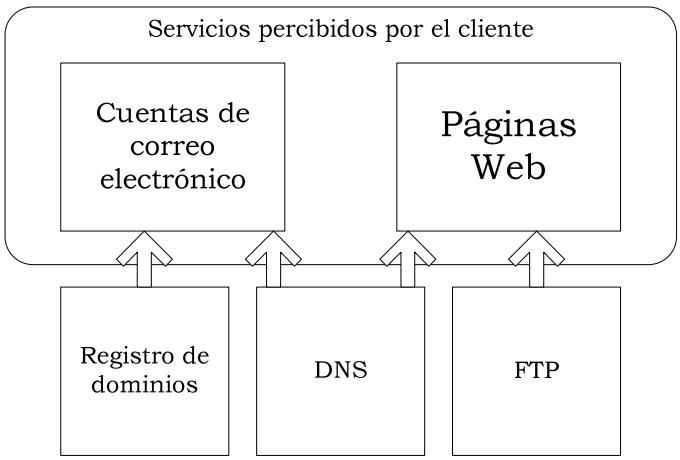
El ISP o IPP que proyectamos dispondrá de una serie de servicios que ofertará y habremos de determinar en esta fase inicial de análisis, para establecer cuáles de ellos serán parte del proyecto.

Dado que los servicios que hoy de manera más común son ofertados en un proveedor de presencia en Internet son el correo y el Web, serán estos los dos y el servicio DNS los que aquí tendremos en cuenta (el DNS no se entiende sino se suministra a la vez algo sobre el mismo, como el correo y el Web, por eso lo habitual será que hablemos más de los otros dos servicios por cuánto serán los que el cliente y usuario van a entender y desear contratar realmente, mientras que el DNS será el paso previo a la contratación de estos otros dos o bien se configurará de manera transparente al usuario en las modificaciones que deseen realizar sobre el servicio Web o de correo).

El diseño y la implementación habrán de permitir un futuro crecimiento en la cantidad de servicios, dado que en Internet existen otros servicios que se pueden ofertar y resultarían igualmente interesantes, pero que por la extensión del proyecto que su inclusión comportaría no han sido tenidos en cuenta. En el epílogo nombraremos algunos de estos servicios que se podrían haber tenido también en cuenta.

Por servicio entenderemos la activación del mismo para un cliente nuestro concreto, es decir: la acción de creación de las cuentas de usuario necesarias para que un cliente consulte su correo o tenga su espacio Web y éste esté disponible al público en Internet será considerado un servicio más que ofreceremos a dicho cliente.

El conjunto de los servicios de un mismo tipo deberán ser suministrados mediante un servidor, entendiendo por tal el software en ejecución en una o varias máquinas y responsable de dicho tipo de servicio.



Servicio Web

El servicio de página Web se entenderá en un sentido muy amplio como el conjunto de información que un cliente puede colocar en nuestros sistemas y que estará disponible al público en general a través del protocolo HTTP vía el puerto TCP número 80 (en el caso de tratarse de una Web normal) o bien del puerto TCP número 443 (en el caso de tratarse de una página que además utilice SSL, que también soportaremos). Para ello será necesario que cada cliente solicite el servicio y se le asigne un usuario y contraseña para acceder mediante protocolo FTP y colocar en su cuenta de usuario la información estática que el mismo desee (sean o no páginas HTML). Este mismo espacio será el que esté disponible a través del servidor Web, y podrá tener áreas de acceso restringido mediante usuario y contraseña, si así lo solicita el cliente.

Además del contenido estático se permitirá la colocación en ese mismo espacio de contenido dinámico, entendiendo por tal las páginas que contengan código dinámico, al estilo de PHP o ASP, y que puedan requerir o no bases de datos. Al igual que estas librerías opcionales (ASP, PHP, etc.) podrán acompañar al servidor aquellas otras que se consideren pertinentes, tal y como podría ser la inclusión de las extensiones FrontPage de Microsoft.

Si el cliente además lo solicita, habrá que ofrecerle otro usuario y contraseña para acceder a una o varias bases de datos relacionales de sintaxis SQL en las que dispongan de los privilegios suficientes para crear tablas, relaciones y aquellas características que el motor de bases de datos escogido soporte. Se consideran únicamente las bases de datos relacionales con sintaxis SQL por ser las comunes en la actualidad.

Tanto las bases de datos como el espacio Web tendrán limitado su tamaño máximo, necesitando el cliente la contratación adicional de espacio cuando alcance el mismo.

Igualmente estará limitada la transferencia máxima mensual que cada sitio Web alojado pueda originar, entendiéndose por tal el tráfico en bytes que el servidor Web ha tenido a causa de dicho sitio. Se suministrarán además estadísticas del mismo tanto al propio cliente como al departamento técnico, estadísticas que habrán de estar desglosadas tanto por días como por horas, mostrando la evolución del tráfico y los propios límites en vigor por cada sitio. Al margen de las estadísticas se guardará registro de todo el tráfico producido por el sitio, incluyéndose tanto las fechas como la identidad del usuario que accede, bien a través de su usuario y contraseña, bien a través de su IP.

Este servicio requerirá además de la posesión de un dominio de Internet por parte del usuario, pudiendo el usuario montar posteriormente sobre el mismo dominio más de un sitio Web.

Servicio de correo

Por servicio de correo electrónico entenderemos el suministrar a nuestro cliente un usuario y una contraseña que le permitan el envío de mensajes de correo electrónico que sean acordes al RFC 2.822[57], y a su vez la recepción de ese mismo tipo de mensajes a través de una dirección de correo electrónico única, dirección que estará formada por una combinación de letras escogida por el cliente más el dominio.

Podrá asimismo asociar más de una dirección de correo electrónico a la misma cuenta, asociar cualquier dirección de correo electrónico que dependa de un dominio de su propiedad a un buzón, y a su vez las operaciones contrarias, como sería hacer que del correo llegado a una misma dirección de correo electrónico se entreguen copias en tantos usuarios como el cliente desee.

La recepción de correo desde Internet por el servidor se hará mediante el protocolo SMTP a través del puerto TCP número 25, cumpliéndose el RFC que hemos anteriormente comentado en el capítulo de antecedentes. Para enviar correo a Internet, nuestro cliente podrá optar por este mismo protocolo autenticándose de manera fehaciente con usuario y contraseña (lo que se conoce como *SMTP autenticado*), o bien utilizar algún protocolo alternativo si este fuera finalmente implantando (IMAP), pero siempre autenticándose.

La recepción del correo que se haya entregado en una determinada cuenta de correo se hará al menos mediante el protocolo POP3 en el puerto TCP número 110, mediante usuario y contraseña, al margen de que se implante también el protocolo IMAP u otros similares.

El espacio máximo que el conjunto de correos por leer de un cliente existan en un buzón estará igualmente limitado, pudiendo el cliente contratar adicionalmente más espacio al respecto.

Tanto el correo entrante como el saliente deberán ser analizados en busca de virus. Además deberá garantizarse al mismo tiempo la inviolabilidad del contenido de los mensajes que nuestros clientes reciban o envíen y el registro de la existencia de los mismos, mediante ficheros de registros a conservar en los que quedará grabado al menos el origen y destinatario, la fecha y hora, así como un identificador del correo, identificador que además deberá ser parte del mensaje en forma de una cabecera que nuestro MTA deberá insertar en caso de no estar presente.

Se podrán añadir características especiales a nuestro servicio de correo, como filtrado o bloqueo de correo no autorizado (conocido como *spam*), mensajes de autorespuesta, envío programado de recordatorios, o la gestión y configuración de listas de correo electrónico.

---

[57] Resnick, P (2.001), miembro de Qualcomm Inc.: *Internet Mesesage Format*. RFC 2822 (que deja obsoleto el RFC822):

< http://rfc.sunsite.dk/rfc/rfc2822.html>

Servicio DNS

El servicio DNS existirá de manera transparente al cliente, ya que el cliente únicamente deseará crear una zona DNS de tercer nivel con objeto de asignarle un sitio Web, o bien deseará recibir correo bajo la misma. Tampoco nos será necesario configurar un servidor DNS que vaya a dar servicio más allá de los dominios delegados, ya que no habrá usuarios conectados a través nuestros que requieran de un servidor DNS recursivo.

Es por eso por lo que el servicio DNS ofrecido carecerá de interfaz Web y se trasladará la operativa de añadir subzonas, modificarlas o borrarlas a las áreas correspondientes de la gestión de cada uno de los servicios en los que la DNS intervenga (correo y Web por el momento).

El sistema únicamente habrá de tener un control sobre los dominios que se encuentran registrados, y permitir al cliente el registro de nuevos dominios sobre los que posteriormente montar nuevos servicios.

Este sistema habrá de ser capaz a su vez de automatizar la gestión de la petición del dominio ante un registrador que posea pasarela que así lo permita (vía aplicación Web, aplicación propietaria, o protocolo o mecanismo autenticado). Este registrador habrá de disponer de un servicio de atención al cliente para la resolución de las más que probables incidencias que en el uso del servicio se presentarán, sobretodo en lo que respecta al traslado de dominios.

Se habrá también de asegurar un cierto determinismo en las peticiones que se hagan sobre este registrador, estableciéndose los plazos para el registro efectivo del dominio y puesta en servicio tras su inclusión en los servidores raíz.

Se habrá también de permitir por parte del registrador escogido el registro de un rango amplio de dominios de primer nivel, tanto con los ccTLD (al menos el *.es*), como de todos los gTLD. La elección del registrador por tanto dependerá del número de TLD de los que soporte registro y de la existencia de una pasarela que permita el registro automático de dominios sin intervención humana, evitándose si es posible el tener que recurrir a más de un registrador, por lo complejidad que eso acarrearía en la implementación.

5.2 El panel de control

5.2.1 Casos de uso

Los actores que van a interactuar con nuestros sistemas son múltiples: uno de ellos será el usuario final de los servicios que vayamos a poner en marcha (es decir, el público en general que a través de Internet enviará un correo a un cliente nuestro, nuestro cliente cuando recoja su correo, alguien que consulte un sitio Web alojado con nosotros, etc.). A este actor lo nombraremos Internet, englobando cualquier comunicación que sobre un servicio en vigor llegue a Internet o esté destinada a ella. Otro actor será el propio cliente cuando gestione estos servicios, solicite de nuevos o elimine alguno de ellos: avanzaremos aquí algo que va ser evidente posteriormente en el diseño, que es que estas gestiones del cliente se producirán a través de una página Web autenticada que comúnmente se conoce como *panel de control*.

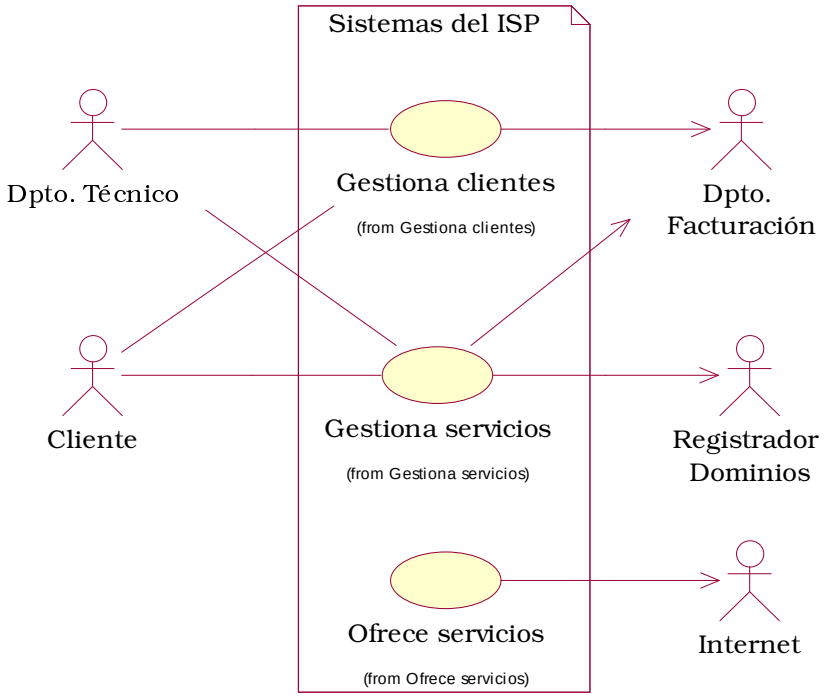


Ilustración 5-1: Diagrama de contexto de casos de uso

Intervienen, asimismo, otros tres actores: el primero sería nuestro departamento técnico, formado por el personal responsable del buen funcionamiento de los sistemas, y que también deberá poder crear, alterar o eliminar servicios. Otro actor relevante es el departamento de administración o facturación, encargado de facturar al cliente los servicios, y finalmente tendríamos un actor externo y que no controlamos directamente como sería el agente registrador a través del

cuál obtenemos los dominios en Internet, único agente externo con el que tendremos que asegurarnos de establecer un mecanismo que permita automatizar la solicitud de sus servicios de registro de dominios.

Con todos estos actores definidos, el diagrama de casos de uso de contexto nos permite identificar de entrada tres partes bien diferenciadas: la gestión a través del Web de los clientes (el caso de uso *Gestiona clientes*), la gestión de los servicios que estos clientes contratarán (el caso de uso *Gestiona servicios*) y por último los propios servicios en funcionamiento interactuando con los usuarios finales (el caso de uso *Ofrece servicios*). Este último caso de uso no va a ser apenas considerado, ya que el objeto de este proyecto no es desarrollar un servidor Web ni una MTA para gestionar el correo electrónico, sino un aplicativo que gestione estos servicios y a los propios clientes.

Serán por tanto los dos primeros casos de uso los que llevaremos íntegramente al modelo de análisis, olvidándonos de este último caso de uso la parte que constituye el propio servicio ofertado, ya que se considerará suficiente la funcionalidad del programa que escojamos en cada caso, y en ningún caso entraremos a modificar el código del servidor Web, sino únicamente determinaremos cuáles de entre los disponibles será el que utilizaremos. Únicamente de este caso de uso nos interesará obtener las estadísticas de uso del mismo para podérselas a ofrecer a los clientes:

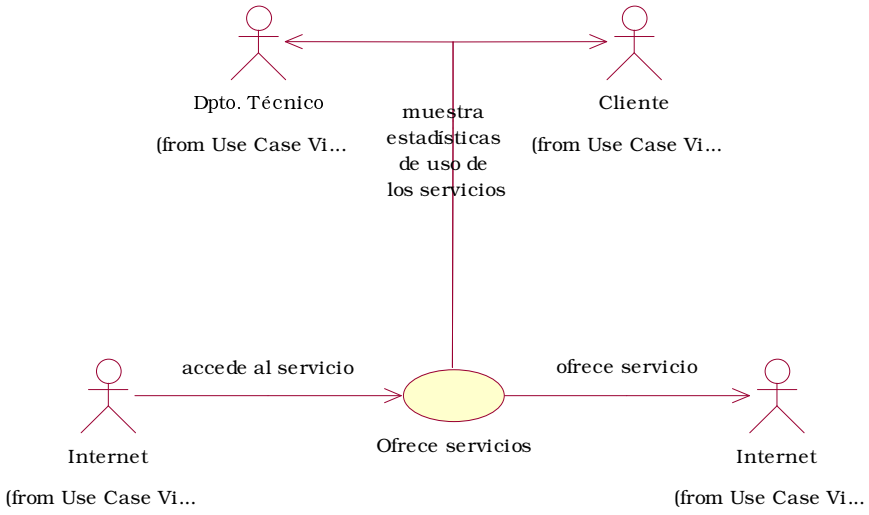


Ilustración 5-2 Diagrama de casos de uso de *Ofrece servicios*

Por lo que respecta a la gestión de los servicios que se realizará de manera prioritaria por el cliente hay que tener en cuenta que también el personal técnico deberá poder acceder a los mismos, fundamentalmente por consultas realizadas para resolver posibles incidencias que los clientes tengan, aunque también en ocasiones para modificar los servicios o bien cancelarlos o bloquearlos por las razones más variadas (aunque una de las más usuales podría ser la falta de pago).

Sólo el departamento técnico dispondrá de acceso a los sistemas que permitan configurar o modificar servicios, interviniendo únicamente el departamento de facturación como parte receptora de la información de esas modificaciones, sin posibilidad de alterarla.

El otro actor que aquí interviene será nuestro registrador de dominios, del que se habrá de obtener también información en lo que respecta a la disponibilidad o no de un determinado dominio, caso de uso aquí no concretado, pero que se daría en un nivel más detallado. A este nivel sólo nos interesa que habremos de establecer un mecanismo suficiente para saber en todo momento en qué situación se encuentra un dominio y controlarlo.

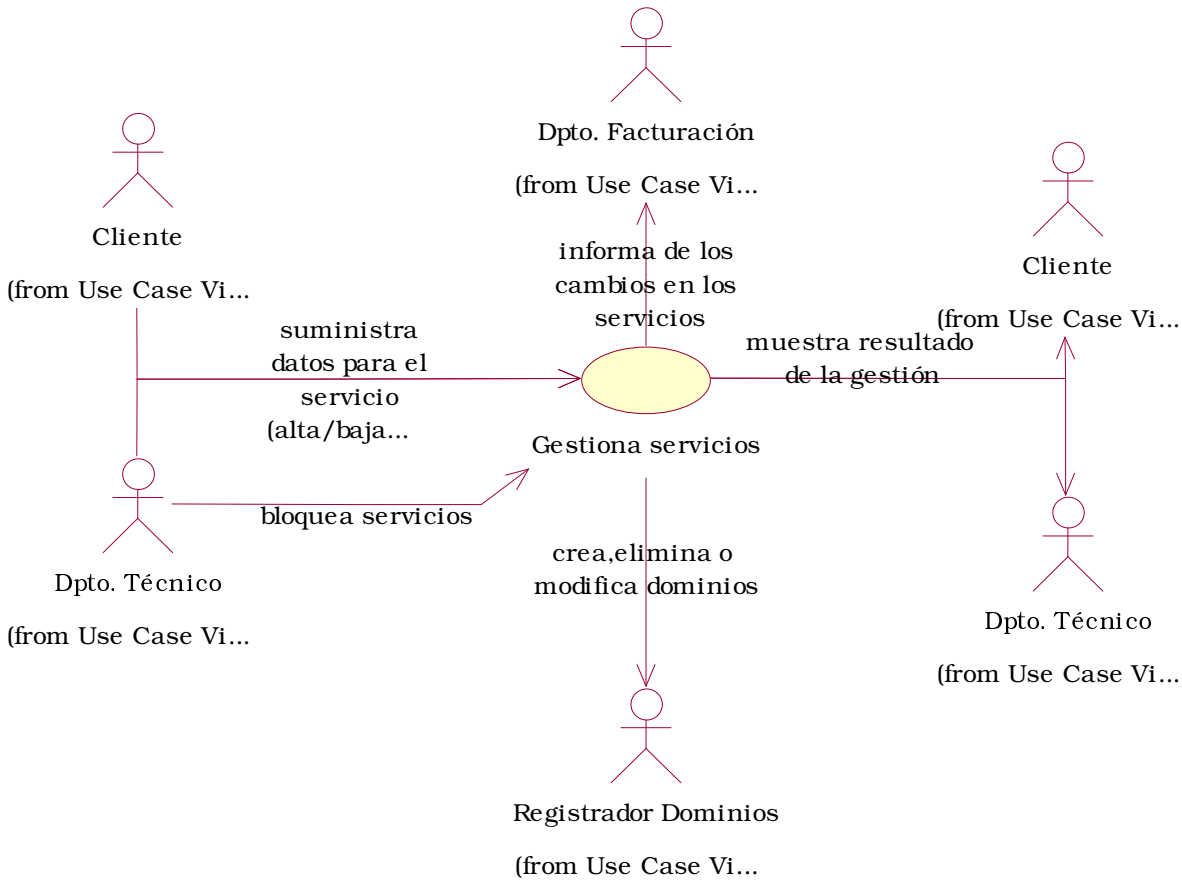


Ilustración 5-3 Diagrama de casos de uso de *Gestiona servicios*

En el diagrama de casos de uso *Gestión de clientes* se ha mostrado con mucho más detalle cuáles serían los casos en él presentes. El primero caso por orden cronológico al que un cliente se enfrentaría sería el de alta (con la pertinente solicitud de sus datos, la validación de los mismos, su almacenamiento y la generación de la información de acceso, es decir: de un usuario y una contraseña que a partir de ese momento el cliente deberá utilizar para acceder al sistema). La información que de los clientes se habrá de almacenar deberá ser la necesaria para la facturación de los servicios (es decir, de al menos una dirección postal, una cuenta corriente o método de pago alternativo,

como sería la tarjeta de crédito, y de un NIF o CIF a utilizar para la factura emitida), así como la que el ISP considere oportuna.

Para los clientes ya registrados en el sistema se ha de permitir la modificación de sus datos registrados, la validación de aquellas modificaciones, e incluso la baja de los clientes (una vez verificado que no le quedan servicios en activo y verificada dicha baja por el departamento de facturación para comprobar que no existan cuentas pendientes).

El último de los casos contemplados es el del acceso al sistema, ya que cada operación que el cliente realice deberá haber pasado antes por la introducción del usuario y contraseña que se le asignaron.

En caso de no recordar estos valores lo habitual es contactar telefónicamente, pero aquí también se ha previsto esa circunstancia y se espera que los clientes utilicen la opción de recordar contraseña, en la que esos valores serían enviados vía postal o por correo electrónico a las direcciones de contacto facilitadas durante el alta.

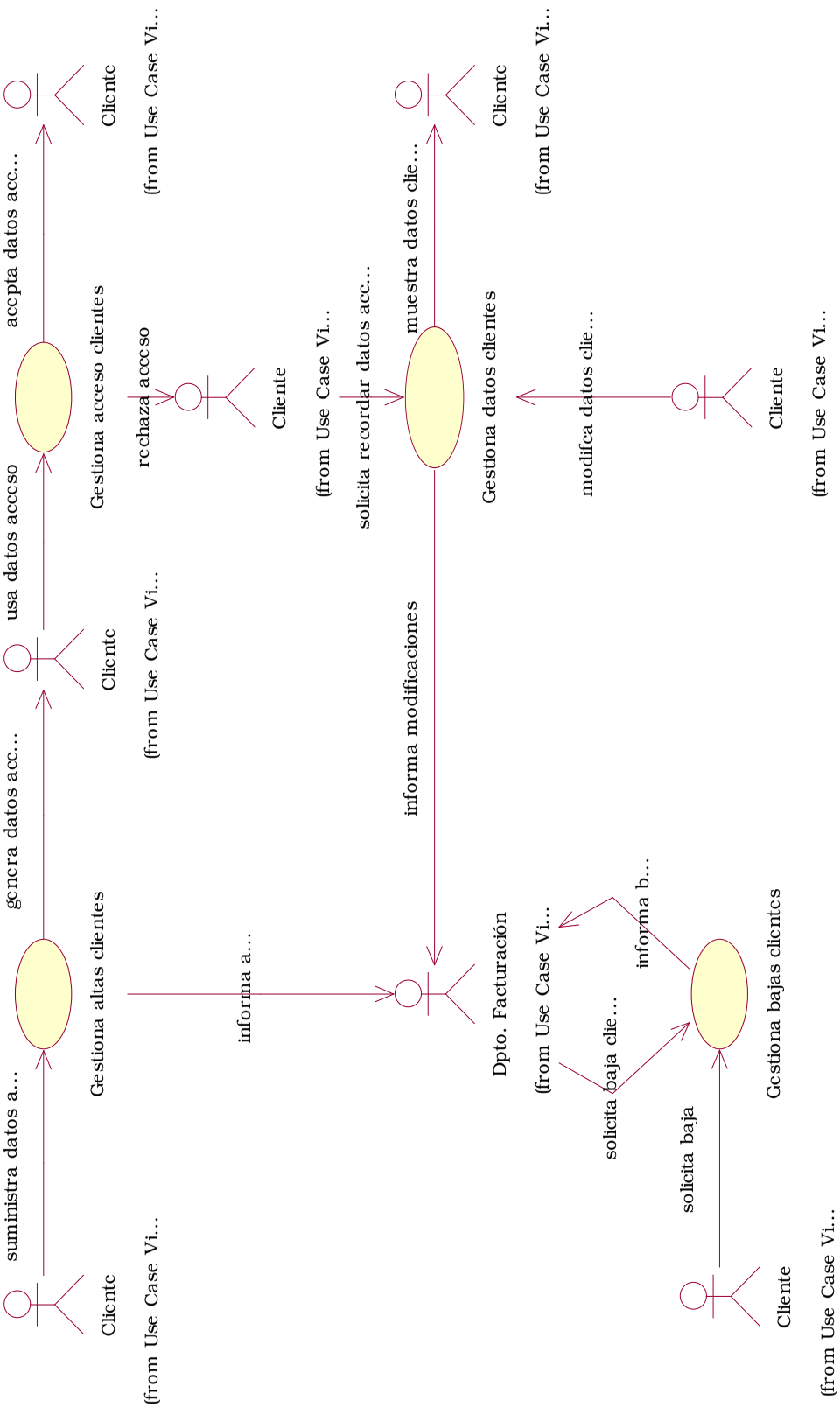


Ilustración 5-4: Diagrama de casos de uso de *Gestiona cliente*

5.2.2 Modelo de análisis

Con los requisitos y los casos de uso ya descritos, nuestro siguiente paso según el modelado UML será establecer un modelo de análisis en el que podamos ya reconocer las partes principales del sistema a desarrollar.

Se ha creído conveniente comentar primero el diagrama de contexto, dada su baja complejidad, y sobre él analizar una de las decisiones más relevantes que en esta fase de análisis se va a tomar.

En el diagrama de contexto se puede apreciar la baja importancia que se da al actor llamado Internet, que sería el usuario final de los servicios. Como ya se ha comentado, el objeto del proyecto no va a ser diseñar un servidor de correo, sino la aplicación que permita gestionarlos sin intervención de usuarios que trabajen desde la consola. Lo que antes era el caso de uso *Ofrece servicios*, queda por tanto reducido a la actividad de generación de estadísticas visibles dentro de la aplicación Web, por lo que a nuestro proyecto se refiere, y a partir de ahora se harán pocas o ninguna referencia a este caso de uso.

En cambio si que se da una cierta importancia es a la comunicación entre nuestro ISP y su registrador de dominios, ya que es crucial que entre la solicitud por parte del cliente y la puesta en servicio no transcurra mucho tiempo, y nuestro proyecto se habrá de dedicar de manera relevante a este respecto.

Lo que antes nombrábamos como una de las decisiones más relevantes es la determinación de qué tipo de gestión sobre los servicios se va a implementar, sobretodo teniendo en cuenta la necesidad de que esta gestión sea accesible tanto para el departamento técnico, que accederá a través de una interfaz de consola de comandos, conectando directamente a alguno de los sistemas del ISP, y estos mismos servicios deberán a su vez ser gestionados desde la Web por parte del propio cliente. Existe pues la posibilidad de desarrollar dos módulos de gestión o bien diseñar dos interfaces que accedan al mismo módulo de gestión, que es la opción considerada más óptima.

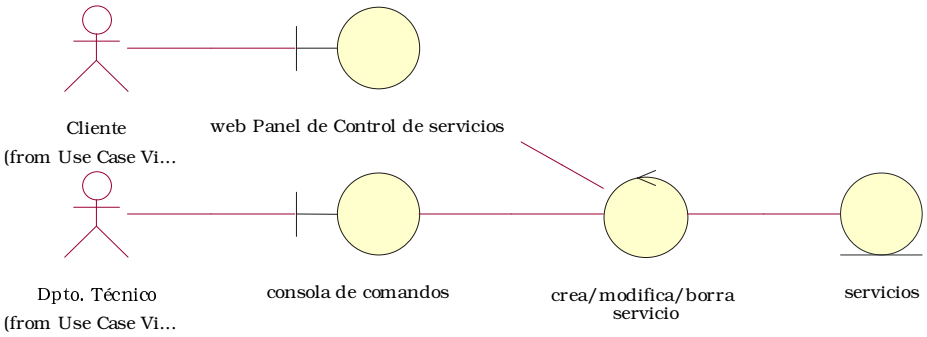


Ilustración 5-5: Detalle de la gestión de servicios del modelo de análisis

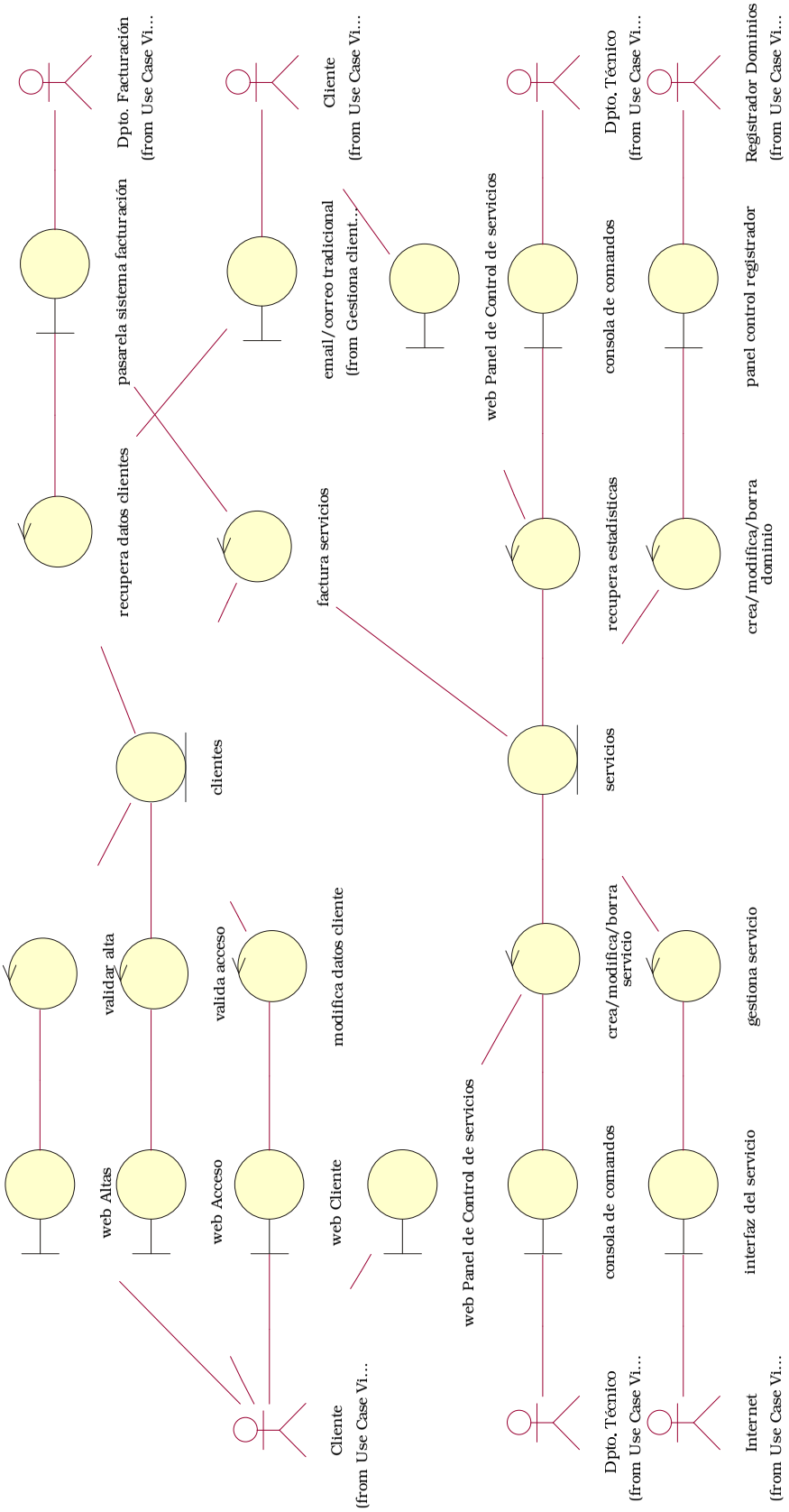


Ilustración 5-6: Diagrama de análisis

En realidad lo que haremos será realmente desarrollar una interfaz potente de cara al cliente, que será la que exija una fuerte comprobación de los valores por él introducidos, además de tener que resultar fácil de aspecto e intuitiva, mientras que la interfaz del departamento técnico podrá ser mucho más simple y diáfana.

De este diagrama de análisis global vamos a especificar algunos diagramas de colaboración más concretos, para mostrar algunos de los problemas con los que nos encontraremos en el diseño, en concreto en la gestión de usuarios y en la solicitud de un servicio concreto.

Área de gestión de clientes: alta de clientes

En el proceso de alta de un cliente se habrá de suministrar al cliente los datos de acceso generados por el sistema, formados por un usuario y una contraseña en cuya elección no interviene el usuario.

Se espera que esto incremente la seguridad del sistema de acceso, ya que de esta manera (con una contraseña aleatoria) se reducen las posibilidades de que un atacante pueda encontrar una combinación válida que le permita acceder de manera fraudulenta a una cuenta de un cliente (cuando el cliente escoge el mismo la contraseña ésta suele ser tan sencilla como su propio nombre de usuario seguido de una cifra, en la mayoría de los casos).

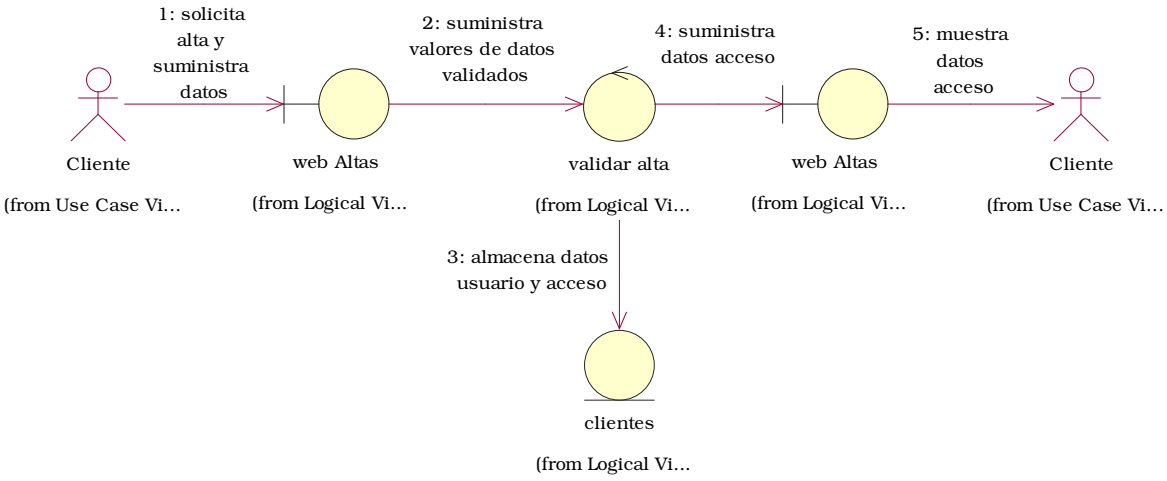


Ilustración 5-7: Diagrama de colaboración de alta de clientes

El usuario de acceso será el número del NIF o CIF del cliente (previamente validado durante el proceso de alta junto con todos los demás datos), evitándose además de esta forma la duplicidad de cuentas para un mismo cliente (el NIF o el CIF se consideran únicos). Se podría incrementa aún más la seguridad si pasados un número máximo de intentos fallidos con un cliente concreto, se bloqueara la cuenta, aunque esta situación no se considera oportuna ya que obligaría a una atención telefónica de esta incidencia que deseamos evitar.

El acceso de los clientes al panel de control o zona en la que pueden modificar servicios y datos personales se hará a partir de estos datos de identificación que se han generado en el proceso de alta.

Dado que son datos en los que no se ha permitido al cliente escoger la combinación de contraseña adecuada, es de prever que el cliente los olvide fácilmente, incluso aunque se le remarque la importancia de recordar los mismos.

Por tanto vamos a tener que establecer un mecanismo de recordar contraseña que permita al cliente entrar al sistema sin tener que solicitar ayuda telefónica.

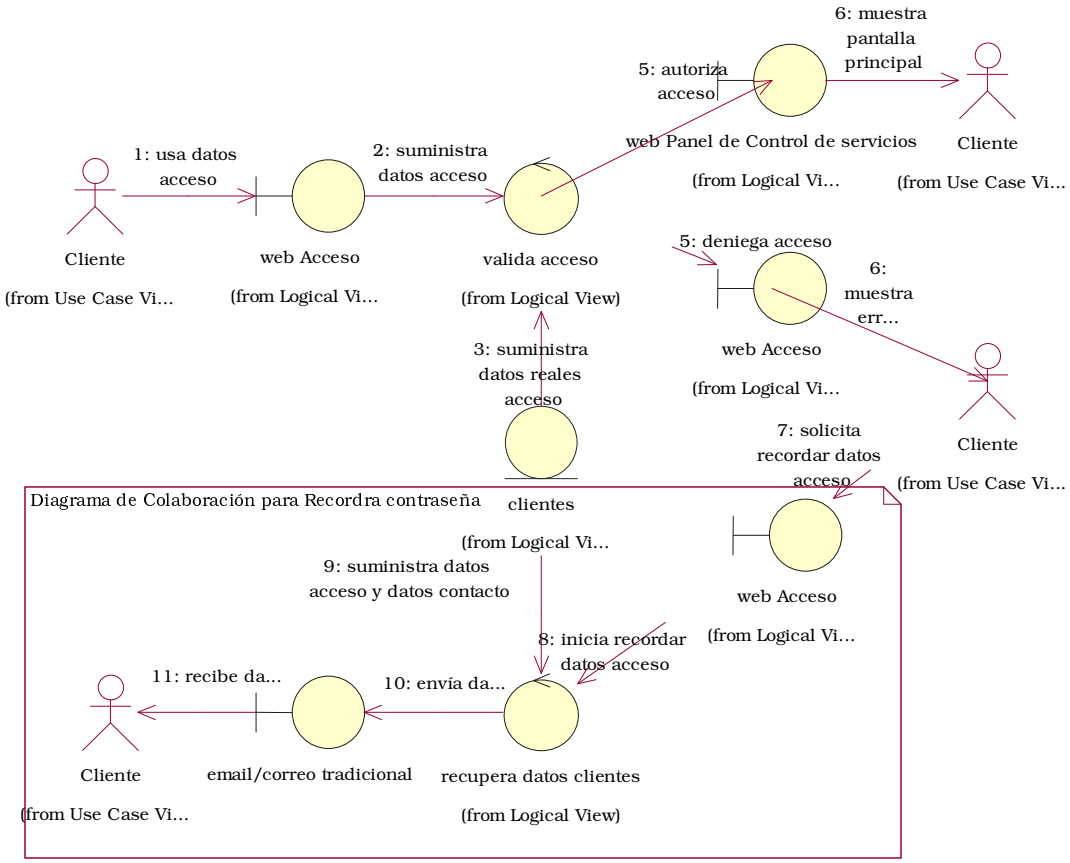


Ilustración 5-8: Diagrama de colaboración del acceso de clientes

El mecanismo de recordar contraseña consistirá en enviar los datos de acceso a alguna dirección que suministró el cliente durante el proceso de alta o figura en nuestras bases de datos como direcciones de contacto, pudiendo ser tanto el correo tradicional como un correo electrónico.

Si se utiliza el correo tradicional entra en juego la molestia causada al cliente por tener que esperar algunos días para acceder al sistema, por lo que es de prever que el cliente acabe llamándonos, además de que genera costes para el ISP en franqueo y papel.

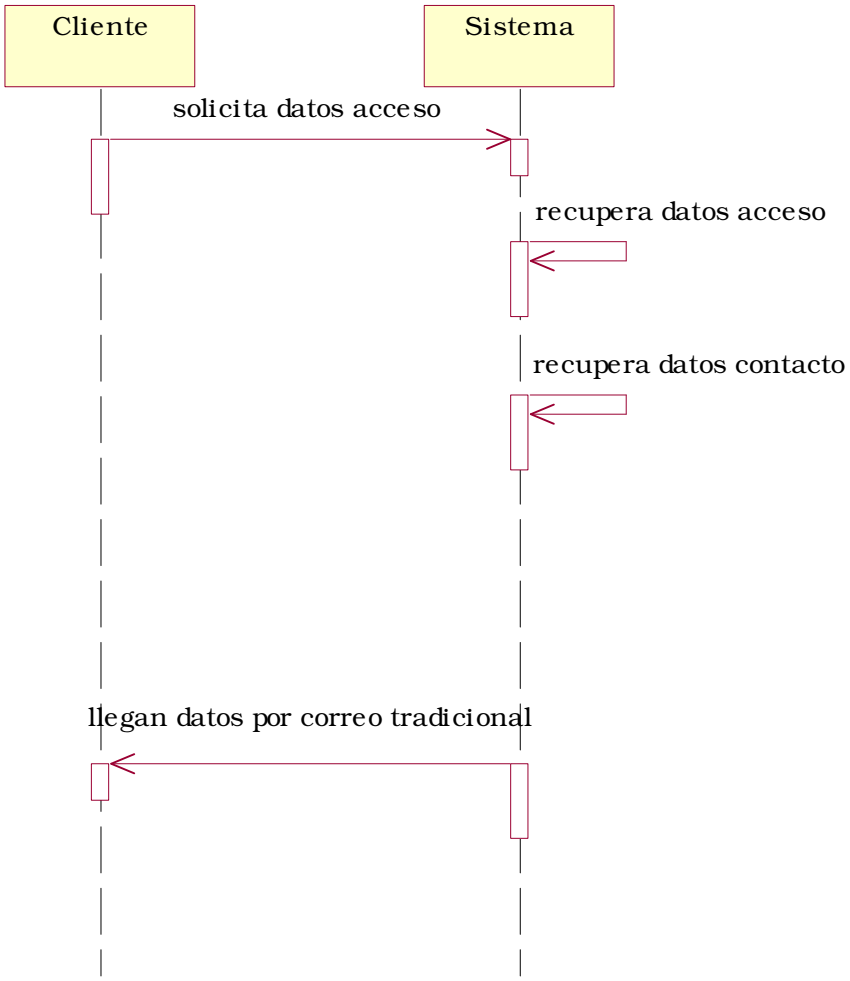


Ilustración 5-9 Diagrama de secuencia al recordar contraseña por correo

Por otro lado, si permitimos el envío de estos datos a una dirección de correo electrónico, aunque no haya retraso y evitemos los costes del correo tradicional, tenemos en nuestra contra dos factores: el primero sería que muchos clientes puedan no haber suministrado una dirección de correo electrónico en el proceso de alta, y el segundo sería la falta de seguridad en las notificaciones por correo.

Por lo que respecta a la ausencia de correo electrónico, la solución pasa por hacer este campo obligatorio durante el proceso y de alta, y además comprobar su validez mediante el envío de un correo a esa dirección que haya de ser respondido para completar el formulario de alta. No nos hemos de preocupar de que los clientes no dispongan de dirección de correo electrónico, ya que es altamente improbable que en las actuales circunstancias de desarrollo de Internet alguien carezca de una dirección electrónica aunque sea para usos privados (posteriormente si crean un servicio de correo como clientes nuestros podrían cambiar su dirección particular por otra que hayan creado en nuestros sistemas).

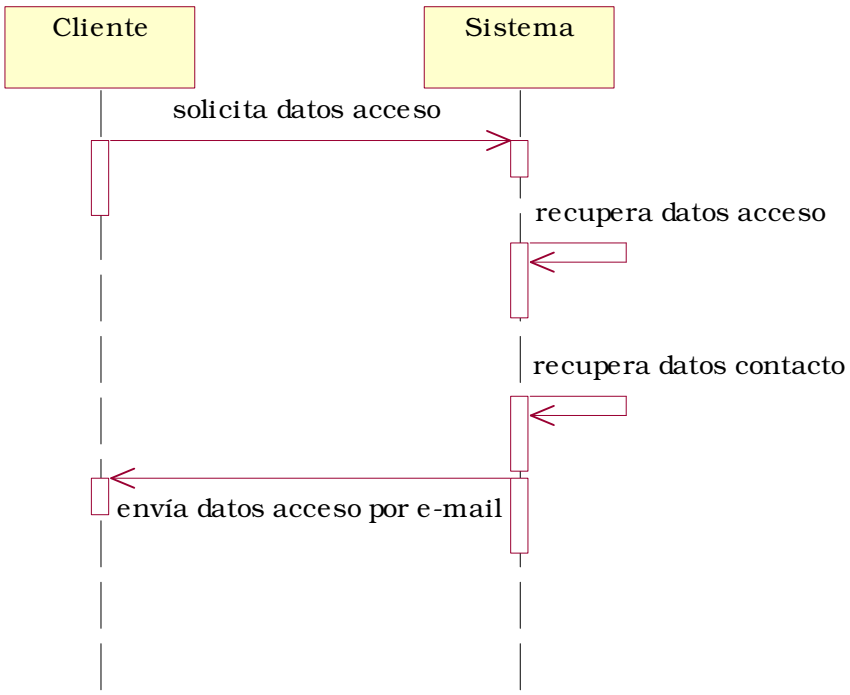


Ilustración 5-10: Diagrama de colaboración para contraseña por e-mail

Básicamente de esta manera garantizamos un canal de comunicación inmediato con el cliente, y con coste despreciable, canal que podría además ser usado para posteriores avisos o publicidad.

El otro problema comentado es la seguridad que ofrece este mecanismo, sobretodo si tenemos en cuenta lo vulnerable que son las cuentas de correo electrónico (podemos no haberle dejado escoger la contraseña de su cuenta de gestión, pero el cliente sí escoge la de su correo y esta puede ser mucho más simple).

La solución que tomaremos será el guardar la fecha y hora de conexión al sistema de gestión, así como la IP desde la que se produjo, datos que además se mostrarán en cada inicio de sesión para que el usuario pueda verificarlos y notificarnos cualquier anomalía detectada (como que asegure que no se conectó en el último mes y el sistema le informe que dos días antes sí se había conectado). Esto también requiere una oportuna concienciación del usuario de lo que implica poseer una contraseña y un usuario, haciéndole ver que ha de conservar las mismas con la misma seguridad que si de una cuenta bancaria se tratara.

Para reforzar esta idea de seguridad se habrá de implementar todo el sitio Web con cifrado SSL: aunque la seguridad que esto ofrezca no vaya a ser realmente mayor, si que incrementará la percibida por el cliente.

En el diagrama de análisis antes mostrado se integró en un mismo control la creación de un servicio, su modificación y su baja. Esto es así porque la secuencia que seguiría cada uno de ellos es la misma, y aquí procederemos a hacer algo similar: en el diagrama de colaboración sólo se mostrará la secuencia que correspondería a un alta de un nuevo servicio, que sería idéntica en su secuencia a la modificación o a la baja.

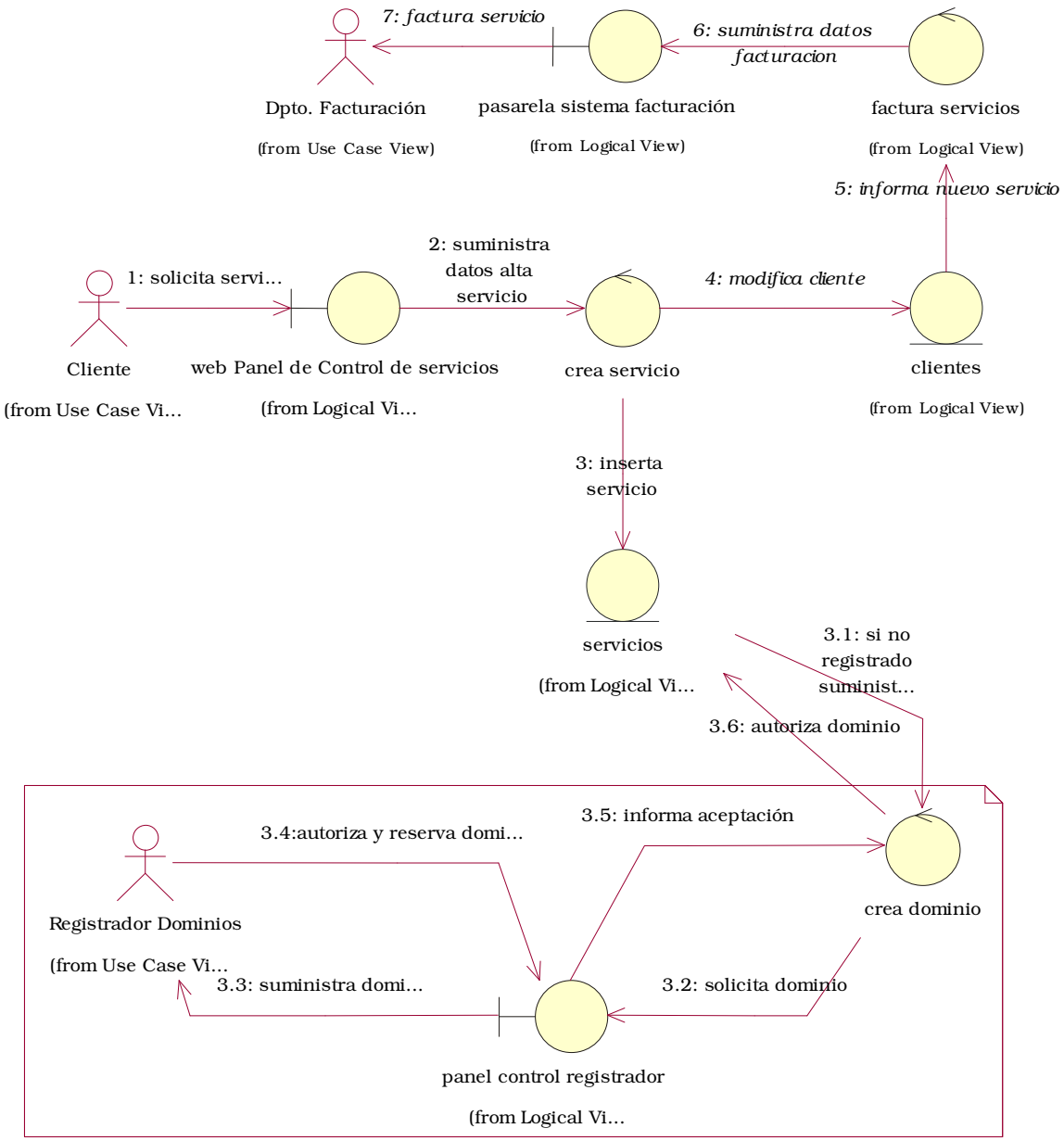


Ilustración 5-11: Diagrama colaboración para creación de un servicio

En el diagrama además se muestra una zona opcional que correspondería a la petición de un dominio: si el servicio que se desea

crear el cliente lo realiza sobre un dominio ya contratado y en servicio, este diagrama de colaboración no sería procedente.

En cambio si solicita un sitio Web para un dominio que desea tener y que aún no hemos adquirido, se activaría todo el proceso destinado a obtener la reserva y activación del mismo en nuestro registrador de dominios.

Hay que tener en cuenta que en este caso además se producirán también retrasos entre la petición del cliente y la plena disponibilidad del servicio, ya que los cambios sobre los servidores raíz de Internet no se propagan de manera inmediata: nuestro cliente tendría activo el servicio, pero desde Internet tardará hasta 48 horas en estar disponible.

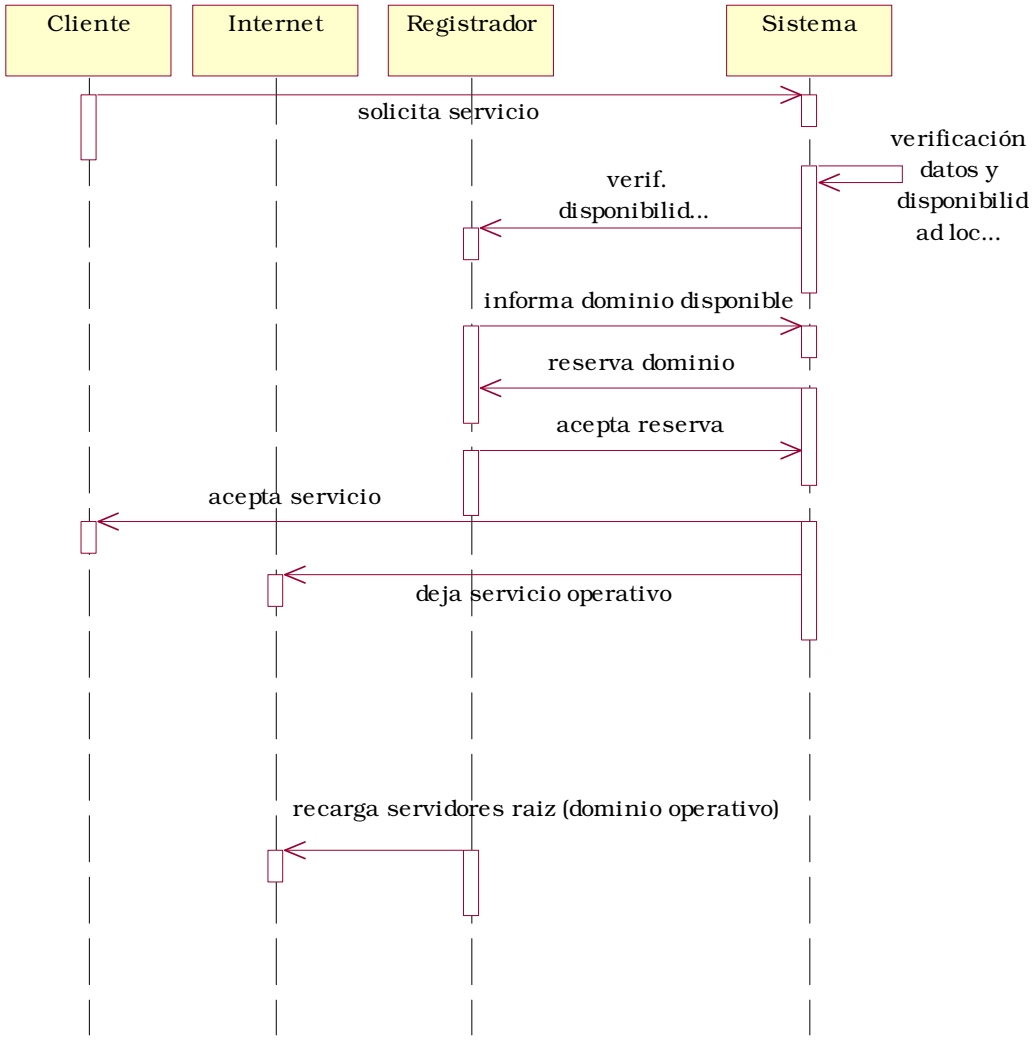


Ilustración 5-12: Diagrama de secuencia para creación de servicios

5.3 Determinación de las actividades

A partir del diagrama de análisis del proyecto del panel de control, se pueden establecer también las actividades que comprenderá, tal y como hemos hecho para la configuración del sistema. Estas actividades se pueden ver unidas de manera jerárquica en el diagrama WBS (*Work Breakdown Structure*), en el que también aparecen la actividad final de integración de los sistemas y el panel de control realizado, finalizando el proyecto tras una última actividad de prueba.

Todas estas actividades del WBS tienen unas dependencias que se reflejan en la tabla de precedencias, y de manera tabular, esas dependencias generarán la matriz de encadenamientos. Las actividades se han numerado con los caracteres del alfabeto latino para facilitar la construcción de la matriz de encadenamientos.

Actividad		Precedencias
A	Diseño de los sistemas del ISP	
B	Implantación del servicio Web del ISP	A
C	Implantación del servicio de correo del ISP	A
D	Implantación del servicio DNS del ISP	A
E	Implantación de la seguridad en los sistemas	A
F	Diseño del panel de control del ISP	A
G	Implementación de la gestión del servicio Web	B
H	Implementación de la gestión del servicio de correo	C
I	Implementación de la gestión del servicio DNS	D
J	Implementación de la gestión de clientes para el panel de control del ISP	F
K	Implementación del panel de control de servicios en la interfaz del ISP	G, H, I
L	Implementación de la interfaz con el registrador	G, H, I
M	Implementación de la interfaz con facturación	J, K
N	Integración final del panel de control	L, M
O	Diseño de la red de acceso	A
P	Implantación de la red de acceso	O
Q	Integración definitiva	E, N, P
R	Prueba final de funcionamiento y correcciones	Q

Tabla 5-1: Tabla de precedencias de las actividades

Las actividades que van desde la F hasta la N son la que estamos modelando usando UML, y que requieren de la formación de un equipo de desarrollo y van a suponer por tanto más tiempo hasta el inicio de actividad del ISP, que se logrará cuando completemos la actividad R o de prueba final de funcionamiento. Se espera que en esa actividad se verifique perfectamente la integridad del panel de control, de cuyo correcto funcionamiento depende gran parte de la credibilidad del ISP como tal. Se deberá realizar también en esta actividad una prueba de carga que verifique que el panel de control no ralentizará el servidor, incrementándose el hardware en ese caso.

Para obtener el camino crítico que nos permitirá determinar la planificación más adecuada nos es necesario fijarnos en la matriz de encadenamientos, que nos permitirá extraer la información necesaria para calcular el diagrama de Pert de nuestro proyecto.

Tabla 5-2: Matriz de encadenamientos

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
A																		
B	■																	
C	■																	
D	■																	
E	■																	
F	■																	
G		■																
H			■															
I				■														
J						■												
K							■	■	■									
L							■	■	■									
M										■	■							
N											■	■	■					
O	■																	
P															■			
Q					■									■		■		
R																	■	

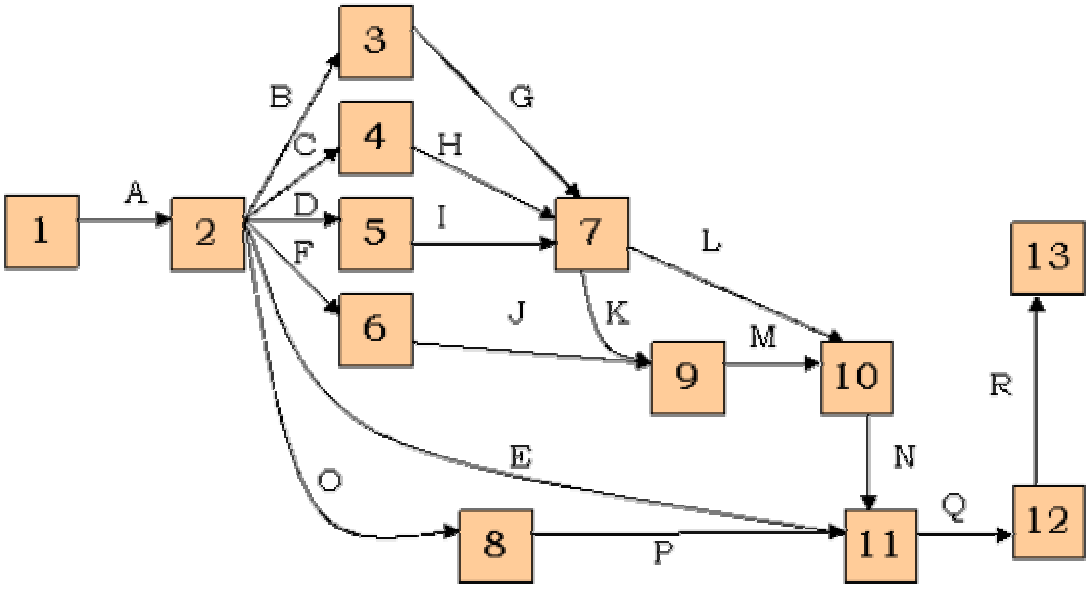


Ilustración 5-13: Diagrama Pert

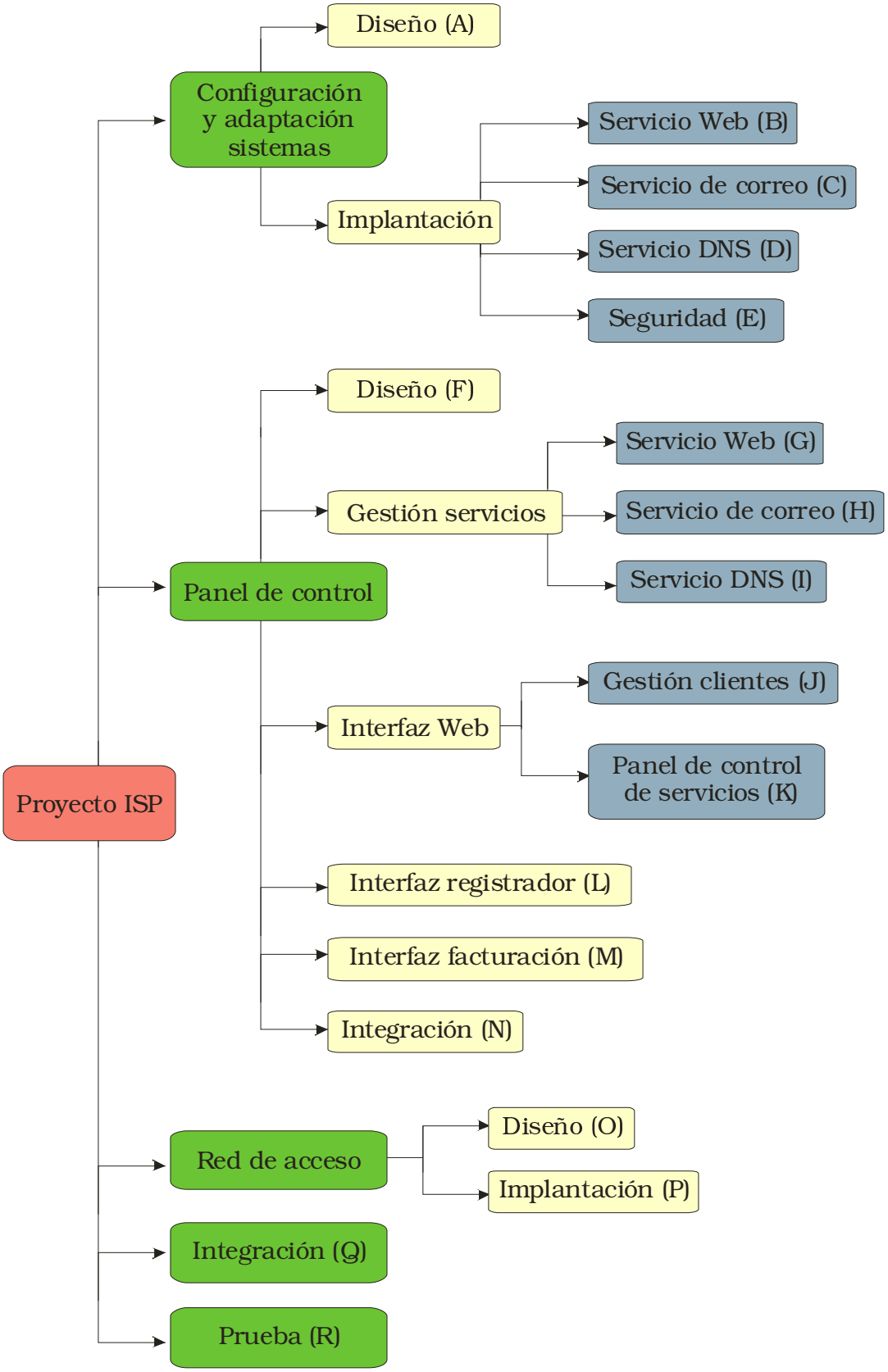


Ilustración 5-14: Diagrama de actividades

Dado que el grafo está por nivelar, y tras aplicar el algoritmo de Demoucron, obtendremos un grafo de Pert nivelado que nos demuestra que las actividades del proyecto se estructuran en nueve fases.

Tabla 5-3: Tabla algoritmo Demoucron

	1	2	3	4	5	6	7	8	9	10	11	12	13												
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
2	0	0	1	1	1	1	0	1	0	0	1	0	0	6	6	6	5	5	5	4	4	3	2	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0				
4	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0				
5	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0				
6	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0						
7	0	0	0	0	0	0	0	0	1	1	0	0	0	2	2	2	2	1	0						
8	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0								
9	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0							
10	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0								
11	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0									
12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0										
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
														13	12	11	8 10	9	6 7		3 4 5			2	1

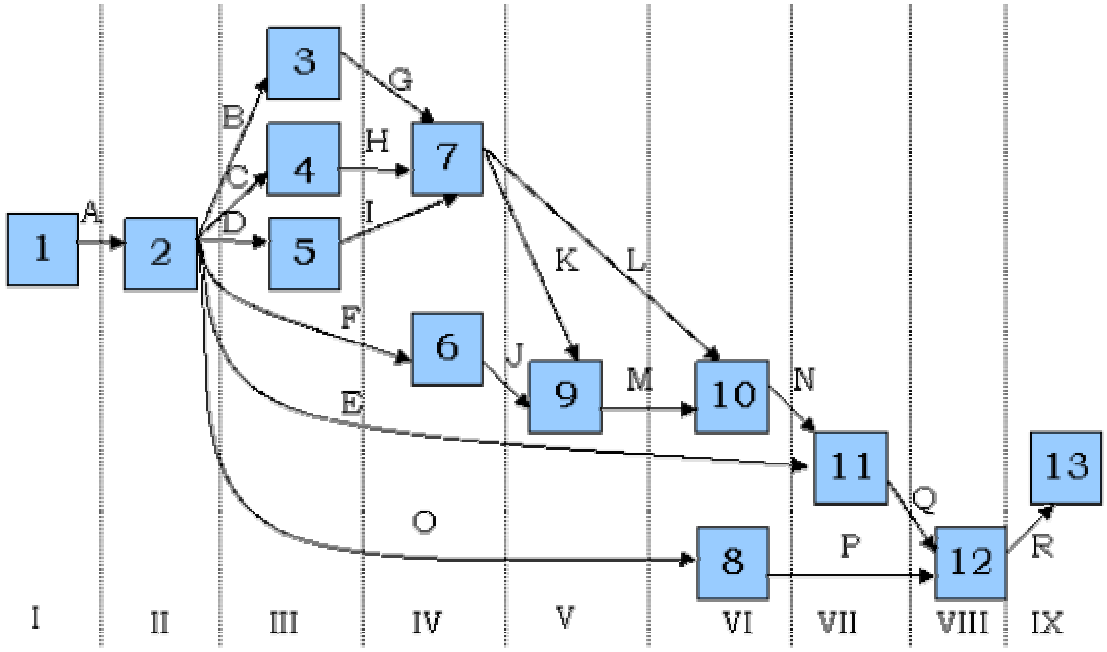


Ilustración 5-15: Diagrama Pert nivelado

De las nueve fases será posible la simultaneidad en el desarrollo de algunas, en concreto aquellas relacionadas con los servicios ofertados, debido a que DNS, Web y correo se consideran independientes entre ellas, tal y como mostraba la tabla de dependencias.

5.4 Análisis de costes temporales

5.4.1 Desarrollo del panel de control

Para la estimación de los costes de las actividades desde la F hasta la N se va a utilizar el modelo de estimación COCOMO, que data de 1.981 y cuya autoría se debe a Barry W. Boehm[58]. COCOMO significa COnstructive COst MOdel, y de entre los modelos que integra, vamos a escoger el llamado COCOMO Intermedio, por ofrecer un mayor detalle en sus cálculos que el básico.

El uso del modelo COCOMO abarcará desde el diseño hasta la integración y prueba, quedando explícitamente fuera el análisis del mantenimiento del software diseñado o de la futura modificación de los sistemas. Lo primero que necesitaremos para el modelo será un cálculo del número de líneas de código a programar.

Para establecer un número aproximado de líneas de código trataremos de establecer una aproximación para cada uno de los módulos a desarrollar, que coincidirán con las actividades hasta ahora comentadas..

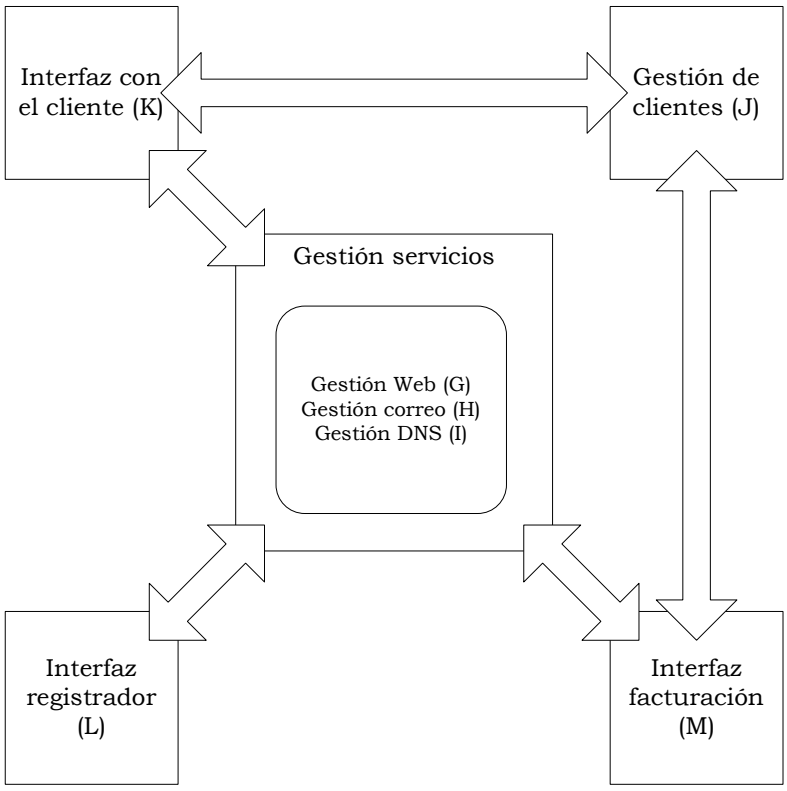


Ilustración 5-16: Comunicación entre los módulos

[58] COCOMO II es una actualización del modelo original, que se hizo en 1.990 y que puede ser consultada en su estado actual en:

<http://sunset.usc.edu/research/COCOMOII>

De estás areas a desarrollar se establecerán una estimación optimista (O) en miles de líneas de código, otra estimación media (M) y por último una pesimista (P). De estos tres valores y por la fórmula de Pert, se extrae el valor esperado (E), valor que da mayor peso a la estimación media (un 66%):

$$E = \frac{O + 4M + P}{6}$$

En esta estimación se ha dado más importancia a la interfaz con el cliente, debido a que es básico que resulte atractiva y sencilla y eso supondrá un mayor coste de desarrollo.

Actividad	KLDC estimadas			
	Optimista	Medio	Pesimista	Esperado
Gestión servicios	4	5,3	6,7	5,32
· Gestión servicio Web (G)	1,5	2	2,5	2,00
· Gestión servicio correo (H)	1,5	1,8	2,2	1,82
· Gestión servicio DNS (I)	1	1,5	2	1,50
Gestión de clientes (J)	3,5	4,8	6,1	4,80
Panel de control de servicios (K)	6	7	8	7,00
Interfaz registrador (L)	1,5	2	2,5	2,00
Interfaz facturación (M)	1,5	2	2,5	2,00
TOTAL	**16,5**	**21,1**	**25,8**	**21,12**

Tabla 5-4: Líneas de código estimadas para cada actividad / módulo

A falta de datos históricos, estableceremos una comparativa global con un proyecto similar llamado phpMyAdmin[59]. Se trata de un proyecto de desarrollo destinado a ofrecer una interfaz Web para administrar el sistema de bases de datos MySQL. Este proyecto concretamente, en su versión 2.4.0 de Febrero del 2.003 contaba con 196 ficheros y 28.656 líneas de código entre lenguaje PHP, SQL y scripts de shell. Se trataría de un proyecto muy similar que el nuestro, que sólo aspira a tener 21,12 KLDC frente a 28,67 que tiene phpMyAdmin.

[59] phpMyAdmin es un proyecto de software libre alojado en SourceForge desde marzo del 2.001. Actualmente está soportado por diez desarrolladores:

<http://sourceforge.net/projects/phpmyadmin>

El siguiente paso será obtener el esfuerzo (E) y la duración (D) del proyecto a partir de las ecucaciones del modelo COCOMO Intermedio, ecuaciones en las que se presentan unas constantes (*a,b,c,d*), con posiblidad de escoger valores para un proyecto orgánico, semi-acoplado y empotrado. Por extensión y complejidad estamos ante un proyecto semi-acoplado, que requerirán distintos niveles de experiencia para enfrentarse a requerimientos poco y medio rígidos:

$$E = a\,\mathrm{KLDC}^{b}\,\mathrm{FAE}$$

$$D = c\,E^{d}$$

$$a = 3{,}0 \quad b = 1{,}12$$

$$c = 2{,}5 \quad d = 0{,}35$$

El otro parámetro presente en la ecuación anterior es el factor de ajuste del esfuerzo o FAE, para cuya determinación necesitamos establecer una serie de pesos a cada atributo conductor del esfuerzo, tal y como está detallado en el modelo COCOMO intermedio. Existen cinco posibles pesos dentro de cada atributo, desde MUY BAJO a ALTO, con unos valores numéricos asignados. El modelo establece el FAE a partir del productorio de los valores dados a estos atributos.

Tabla 5-5: Valor de los atributos para este proyecto

Confiabilidad requerida.	ALTO
Tamaño de la base de datos.	NOMINAL
Complejidad del producto.	NOMINAL
Atributos del sistema (limitaciones impuestas por el hardware y el sistema operativo donde el proyecto va a funcionar).	
Restricciones de tiempo de ejecución.	BAJO
Restricciones de memoria principal.	BAJO
Volatilidad del sistema	NOMINAL
Tiempo de respuesta de la computadora.	ALTO
Atributos de personal (nivel de habilidades que tiene el personal. Son habilidades profesionales generales, habilidad de programación, experiencia con el desarrollo y familiaridad con el dominio del proyecto).	
Capacidad del analista.	NOMINAL
Experiencia en aplicaciones.	NOMINAL
Capacidad del programador.	NOMINAL
Experiencia con el sistema	NOMINAL
Experiencia con el lenguaje de programación.	ALTO
Atributos del proyecto (restricciones y condiciones bajo las cuales el proyecto se desarrolla).	
Prácticas modernas de programación	NOMINAL
Uso de herramientas de software.	NOMINAL
Calendario de desarrollo requerido.	ALTO

En los atributos se han resaltado aquellos que hoy día se tienen en cuenta para la determinación del esfuerzo, ya que es muy común no considerar los quince atributos sino únicamente los cinco resaltados. De los atributos que hemos descartado algunos ni siquiera eran relevantes para nuestro proyecto (volatilidad del sistema o tiempo de respuesta no son críticos excepto en tiempo real).

MUY BAJO	0,75
BAJO	0,88
NOMINAL	1
ALTO	1,15
MUY ALTO	1,4

Tabla 5-6: Pesos de los atributos en el modelo de COCOMO intermedio

El atributo al que se ha dado mayor peso es el de la confiabilidad del sistema: se espera una aplicación segura, debido a que por diseño un error de seguridad grave podría suponer que cualquiera pudiera crear cuentas o modificar el sistema, un hecho de extrema gravedad para la supervivencia del ISP. El resto de los atributos tienen valor NOMINAL o como el caso del tiempo de ejecución o memoria principal necesaria, BAJO debido a que la carga que se espera suponga para un servidor nuestra aplicación será baja (estamos ante una interfaz para el cliente, no diseñando el MTA o el servidor Web).

De esta forma el valor de FAE queda en 1,354, y el esfuerzo global en de 123,73 personas/mes, con una duración lineal para el panel de control estimada en 13,50 meses.

Módulos	Esfuerzo (E)	Duración (D)
Gestión servicios	31,15	3,40
· Gestión servicio Web (G)	11,72	1,28
· Gestión servicio correo (H)	10,64	1,16
· Gestión servicio DNS (I)	8,79	0,96
Gestión de clientes (J)	28,12	3,07
Panel de control de servicios (K)	41,01	4,47
Interfaz registrador (L)	11,72	1,28
Interfaz facturación (M)	11,72	1,28
TOTAL	123,73	13,50

Tabla 5-7: Esfuerzo y duración según COCOMO Intermedio

5.4.2 Diseño e implantación de los sistemas y la red

La estimación relativa a la configuración de los sistemas y diseño de la red la haremos por juicio experto usando como unidad de medición la persona/mes, al carecer esta parte de programación no son modelables mediante las técnicas que acabamos de emplear. Además, no van a suponer un porcentaje alto del total del proyecto y será escaso el riesgo asumido por tomar un juicio basado en experiencias previas.

Pese a que puedan parecer valores bajos que luego vayan a incumplirse, además de que ya se ha comentado que el volumen reducido de los sistemas a configurar ayudará, esta estimación por juicio experto se basa también en las siguientes aseveraciones:

- El diseño de los sistemas y de la red buscarán implantar sistemas y hardware estándar, lo que facilitará y acelerará su implantación.
- El software a implantar, pese a ser no propietario, dispone de manuales y documentación suficiente para poder asegurar que no van a consumir tiempo excesivo, tanto en lo que compete a la configuración del sistema operativo como lo relativo al diseño, y a la implantación de medidas que incrementen la seguridad global del sistema.
- La integración definitiva se supone de corta duración por esperarse que el detallado análisis realizado evite complicaciones excesivas.

Tabla 5-8 Costes para la implantación de los sistemas

Las unidades de esfuerzo son personas/mes, y la duración en meses.

	Actividad	Esfuerzo	Duración
A	Diseño de los sistemas del ISP	1,25	1
B	Implantación del servicio Web del ISP	0,25	0,2
C	Implantación del servicio de correo del ISP	0,25	0,2
D	Implantación del servicio DNS del ISP	0,1	0,2
E	Implantación de la seguridad en los sistemas	0,5	1
F	Diseño del panel de control del ISP	2	1
N	Integración final del panel de control	2	1
O	Diseño de la red de acceso	2	1
P	Implantación de la red de acceso	0,3	0,2
Q	Integración definitiva	1	0,5
R	Prueba final de funcionamiento y correcciones	3	1
	TOTAL	12,65	7,3

5.4.3 Resumen de costes temporales

Estos valores de esfuerzo y duración estimados por COCOMO los uniremos ahora a la estimación que del resto de actividades hemos realizado por el método de juicio experto (ya que no había desarrollo de código posible que nos pudiera servir para usar COCOMO).

Tabla 5-9 Costes por actividad

Las unidades de esfuerzo son personas/mes, y la duración en meses.

El fondo gris corresponde a las actividades que se han estimado mediante COCOMO, mientras las otras lo fueron por juicio experto.

Actividad		Esfuerzo	Duración
A	Diseño de los sistemas del ISP	1,25	1
B	Implantación del servicio Web del ISP	0,25	0,2
C	Implantación del servicio de correo del ISP	0,25	0,2
D	Implantación del servicio DNS del ISP	0,1	0,2
E	Implantación de la seguridad en los sistemas	0,5	1
F	Diseño del panel de control del ISP	2	1
G	Implementación de la gestión del servicio Web	11,72	1,28
H	Implementación de la gestión del servicio de correo	10,64	1,16
I	Implementación de la gestión del servicio DNS	8,79	0,96
J	Implementación de la gestión de clientes para el panel de control del ISP	28,12	3,07
K	Implementación del panel de control de servicios en la interfaz del ISP	41,01	4,47
L	Implementación de la interfaz con el registrador	11,72	1,28
M	Implementación de la interfaz con facturación	11,72	1,28
N	Integración final del panel de control	2	1
O	Diseño de la red de acceso	2	1
P	Implantación de la red de acceso	0,3	0,2
Q	Integración definitiva	1	0,5
R	Prueba final de funcionamiento y correcciones	3	1
	TOTAL	136,37	20,8

5.5 Recursos

En el capítulo de recursos necesarios para el proyecto no se analizará el acceso a Internet ya que éste únicamente se hará necesario contratarlo al iniciar la actividad, luego no afecta a los costes de desarrollo del proyecto ni tampoco se requiere una planificación. Únicamente se hará, en la fase de diseño del proyecto, un estudio de las diferentes opciones disponibles. Por tanto los únicos recursos considerados para el análisis del proyecto serán el hardware, software y recursos humanos necesarios, que es lo que pasaremos a analizar.

Durante el desarrollo del producto el ISP deberá estar conectado a Internet por razones de pruebas de funcionamiento y la propia operativa de cualquier compañía en la actualidad, pero el Internet necesario para esto se considera un servicio básico que, al igual que ocurre con el teléfono, luz y agua, cualquier compañía ha de tener al margen de que existan proyectos en desarrollo o no.

5.5.1 Hardware

El ISP requerirá al menos de dos servidores, uno que actuará de máquina principal y cuyas capacidades técnicas habrán de ser por tanto superiores, y una segunda máquina que actúe de servidor de respaldo como medida de precaución, así como de DNS secundario de las zonas alojadas en el servidor principal.

De un servidor se espera que esté en marcha las 24 horas del día, con lo que será importante que disponga de capacidades de tolerancia a fallos (discos RAID, fuente de alimentación redundante, etc.). Por las características del proyecto, además, no es necesaria una potencia descomunal: los servicios que se pretenden ofrecer no van a alojar grandes bases de datos, por estar el negocio orientado hacia alojamientos de páginas estándar y no a medida o de grandes corporaciones, que antes optarían por una fórmula de *housing*. Sólo exigiremos de los servidores un espacio en disco elevado y cierta capacidad de tolerancia a fallos, mientras la potencia quedará en un segundo plano.

El servidor de respaldo hará su función de manera íntegra si además de estar para suplantar las tareas fundamentales del servidor principal es capaz también de suplir estos servicios en caso de fallo de la conexión principal con nuestra empresa. Por servicios fundamentales entenderemos el *relay* del correo y el mantenimiento en operativo de la resolución de nombres, quedando a merced del diseño del proyecto lograr además que este servidor de respaldo sea capaz incluso de ofrecer las páginas Web alojadas cuando el servidor principal no lo haga.

Para lograr este objetivo el servidor de respaldo debería estar alojado en una localización diferente, y conectado mediante otra conexión. Este

hecho nos llevaría a incrementar sensiblemente el gasto y plantea además el problema de determinar dónde alojar dicho servidor (ya que la oficina y sede del ISP va a ser la que aloje el servidor principal y los equipos de trabajo, y según lo planteado evitaremos que esté junto a dicho servidor).

Como alojar el equipo en el domicilio de alguien no parece lo más adecuado, buscaremos contratar un alojamiento en *housing* de nuestro servidor de respaldo en otra compañía, de manera que durante los esperados pocos periodos de caída del servicio a través del servidor principal, haya otro servidor en Internet para seguir asegurando nuestra presencia y que no nos suponga un coste elevado.

Se optará por la fórmula del *housing* frente a otras más económicas como las plataformas de servidores compartidos o dedicados porque se desea mantener el control total del software y servicios que en dicho equipo se van a instalar. Además, dado que tenemos previsto no cursar mucho tráfico a través del servidor, estipularemos un coste único mensual de 100 euros por alojamiento del servidor en *housing*, con IP fija para el servidor, cuotas por consumo de caudal y consumo eléctrico incluidos. El mantenimiento del servidor correrá por cuenta del personal técnico de nuestro ISP, y se hará vía conexión remota. El equipo de respaldo será adquirido por 1.500 euros con una vida útil esperada de 30 meses. Si se opta por adquirir un servidor a medida suministrado por el operador que nos ofrezca el *housing* se espera que éste suponga un incremento de no más de 50 €/mes en la factura.

El coste mensual del *housing* no va a ser tenido en cuenta durante el desarrollo del proyecto ya que también se estima necesario únicamente a partir de la puesta en marcha del ISP. El servidor de respaldo por el contrario será adquirido desde el principio y configurado totalmente aunque no esté aún alojado en su lugar definitivo, para que antes del inicio de la actividad del ISP ya esté convenientemente configurado.

Se estimará un gasto de 3.000 € para el servidor principal, con una vida útil esperada de 30 meses. Por tanto el coste quedará en 100 €/mes si lo computamos sobre esos 2 ½ años de vida antes del recambio del mismo. Se muestra este valor para permitir su comparativa con el servidor de respaldo en *housing*.

Existe asimismo otro hardware a analizar: el diseño e implantación de la red local y del acceso a Internet, que se podrá producir bien a través de routers de los proveedores contratados, bien a través de routers adquiridos por el ISP en propiedad. El ISP necesitará habilitar en sus instalaciones un armario de distribución principal (conocido comúnmente por sus siglas en inglés: MDC) y de un punto de presencia o POP. Se denomina MDC al armario o zona en la que se dispone el hardware necesario para dar soporte a la red, mientras el POP es el área destinada a contener el hardware que da acceso a Internet (y que normalmente cede la operadora con la que contratamos el mismo). Se hará una estimación global por capítulos bastante baja, ya que escasamente disponemos de dos servidores y no vamos a necesitar por

tanto una gran cantidad de conexiones de red, ni tampoco un ancho de banda inmenso ya que no habrá muchos usuarios trabajando en la red local. Además, desde Internet siempre será menor el ancho de banda disponible que la peor de las conexiones de red local posibles.

La red será de tipología en estrella y de tipo *Ethernet* 802.3 de 100 Mbps., elección que no discutiremos en el diseño porque es de lejos la configuración dominante en la industria (y por tanto más sencilla de mantener y de adquirir el hardware pertinente). Deberán existir puntos de conexión suficientes a la misma para los servidores actuales y al menos tres más, así como para un mínimo de veinte puestos de trabajo, cifras totalmente arbitrarias pero que dejan un margen suficiente para crecer al menos hasta el doble de tamaño que el inicial, considerado un margen suficiente.

Tabla 5-10: Costes de hardware del proyecto

Servidor principal	3.000 €
Habilitación y montaje del MDC	1.000 €
Habilitación y montaje del POP	1.000 €
Acometidas eléctricas y cableado de la totalidad de las instalaciones del ISP	1.000 €
Habilitación de la zona de servidores y puestos de trabajo	2.000 €

5.5.2 Software

Los servidores utilizarán un sistema operativo orientado a redes, no distribuido (ya que el servidor de respaldo se espera sea útil justamente cuando el principal falle). En cualquier caso el sistema operativo tendrá que tener implementada la pila de protocolos TCP/IP en su versión 4, y se valorará la disponibilidad de la futura IPv6. En el hardware no se ha tenido en cuenta la disponibilidad o no de IPv6 para la adquisición de routers y otros aparatos en los que fuera relevante, debido a la casi nula disponibilidad de este tipo de tecnología en el hardware, y el alto coste que supondría.

El sistema operativo no podrá ser propietario para incrementar los costes, reduciéndose la terna de candidatos a dos posibilidades a priori: Linux y FreeBSD. Se valora como más importante el mayor conocimiento que del primero tendrá el personal que haya de ser contratado para decantarse por el mismo.

Al margen del sistema operativo se necesitará también un servidor Web, un MTA y un software de servidor DNS. Aunque en la fase de diseño nos plantearemos las diferentes alternativas existentes para cada uno de ellos, y analizaremos cuál utilizar, serán todas ellas software GNU, por lo que tampoco se contemplarán costes de licencia para estos programas.

La apuesta por el software no propietario podría ser arriesgada ante la falta de un fabricante de software que lo haya desarrollado y se encargue de ofrecer el soporte del mismo, pero no constituye en la

actualidad ya un riesgo para el proyecto ni para el servicio posterior del ISP por cuanto se buscará formar un equipo de programación con un nivel medio/alto sobre estos sistemas, al igual que con el futuro equipo técnico del ISP. Incluso para el desarrollo se usarán las herramientas de ingeniería del software y los entornos de desarrollo no propietarios existentes.

De todo lo comentado hasta el momento la conclusión es que el coste de licencias para la empresa va a ser nulo en todos los casos, tanto en el desarrollo del proyecto como en el posterior funcionamiento del mismo.

5.5.3 Personal

Según lo estimado en la Tabla 5-9 de estimación de costes del apartado 5.4.3 de este capítulo, el esfuerzo total previsto es de 136,37 personas/mes, con una duración del proyecto de 20,8 meses lineales, implicándose estas personas tanto en el desarrollo del software a diseñar, como en la implementación de los sistemas y de la red del ISP.

El cociente de estos dos valores arroja unas necesidades medias de personal para el proyecto de siete personas (6,56 concretamente). Pero si analizamos cada actividad de manera individual el personal necesario en algunas fases del proyecto será mayor, de hasta nueve personas, como luego veremos.

Una vez finalizado el proyecto el ISP necesitará personal técnico que atienda las incidencias que se presenten, y de hecho se espera que en el futuro crezcan el número de servicios disponibles, por lo que será necesario desarrollar o modificar software, y resultará interesante que dicho desarrollo lo produzca el mismo equipo o parte de él.

Por tanto buscaremos formar un equipo humano parte del cual vaya a continuar posteriormente en el ISP como personal técnico. Dado el tamaño que tiene la empresa objeto del estudio (más bien pequeña) únicamente dos de los programadores y el jefe de proyecto continuarán una vez finalizado el proyecto.

Se estimarán pues tres perfiles de personas a contratar: los programadores del software cuya función futura será de técnicos (dos personas que daremos en llamar programador *senior*), los programadores contratados únicamente para desarrollar el proyecto (siete personas que llamaremos programador *junior*), y un jefe de proyecto que además sea el responsable del diseño y de la coordinación. Esta segunda persona tendrá un coste laboral superior, y estará previsto que una vez finalizado el proyecto sea el jefe de sistemas del ISP.

Tabla 5-11: Costes de personal

	Coste/hora
Jefe de proyecto	30 €
Programador *senior*	20 €
Programador *junior*	15 €

El número de programadores necesarios se ajustará según la planificación temporal de cada actividad, buscando la simultaneidad en el desarrollo de alguna de estas tareas siempre que los recursos humanos máximos previstos (diez personas) lo permitan. Dentro de cada actividad se ha estimado de manera arbitraria la dedicación que habrá de dedicarle el jefe de proyecto a tareas de coordinación, excepto en aquellas tareas de diseño, que le competen de manera exclusiva. En la Tabla 5-12, el porcentaje representa el tiempo de su jornada que el trabajador dedicará a la actividad, mientras que de los programadores se muestra el número total necesario de cada tipo en cada momento.

Habrá momentos con infrautilización de recursos humanos, debido a que las actividades requieren menos esfuerzo que el personal disponible y no habrá actividad que en ese momento se pueda acometer. No nos interesará en esos momentos prescindir del personal si un mes más tarde hay que contratarlo de nuevo, por lo que ya asumimos que del personal a contratar, tanto el jefe de proyecto como los programadores señor lo serán desde el primer momento, pese que a que haya momentos en que no sean útiles al 100%.

De los datos de esta tabla se extraen un periodo en el que se habrá de coordinar a un total de nueve programadores, que irá desde la actividad G hasta la M, y corresponde a la implantación del panel de control. Este es el momento en el que se contratará a los programadores *junior*, periodo al que precederán otros dos en los que únicamente nos hagan falta los dos programadores que luego continuarán en el proyecto.

Tabla 5-12: Asignación recursos humanos (2/62,5% representa que de las dos personas, usarán sólo el 62,5% del tiempo a esa actividad, pudiéndose simultanear con otra

Actividad	Esfuerzo (pers /mes)	D (meses)	Personas (E/D)	Programadores. (número y % tiempo)		Jefe proyecto (% tiempo)
				senior	*junior*	
A. Diseño de los sistemas del ISP	1,25	1	1,25	2 50%		100,00%
B. Implantación del servicio Web del ISP	0,25	0,2	1,25	2 62,5%		0,00%
C. Implantación del servicio de correo del ISP	0,25	0,2	1,25	2 62,5%	0	0,00%
D. Implantación del servicio DNS del ISP	0,1	0,2	0,50	2 0,25%		0,00%
E. Implantación de la seguridad en los sistemas	0,5	1	0,50	0		50,00%
F. Diseño del panel de control del ISP	2	1	2,00	2 0,50%		100,00%
G. Implementación de la gestión del servicio Web	11,72	1,28	9,16			
H. Implementación de la gestión del servicio de correo	10,64	1,16	9,17			
I. Implementación de la gestión DNS	8,79	0,96	9,16			
J. Implementación de la gestión de clientes	28,12	3,07	9,16	2 100%	7 100%	25,00%
K. Implementación del panel de control de servicios	41,01	4,47	9,17			
L. Implementación de la interfaz con el registrador	11,72	1,28	9,16			
M. Implementación de la interfaz con facturación	11,72	1,28	9,16			
N. Integración final del panel de control	2	1	2,00	2 100%		25,00%
O. Diseño de la red de acceso	2	1	2,00	2 100%		100,00%
P. Implantación de la red de acceso	0,3	0,2	1,50	2 25%	0	100,00%
Q. Integración definitiva	1	0,5	2,00	2 50%		50,00%
R. Prueba final de funcionamiento	3	1	3,00	2 100%		100,00%
TOTAL	136,37	20,8	9,17			

5.6 Agenda

5.6.1 Técnica CPM

A partir de los datos de coste temporal por cada actividad de la Tabla 5-9, y de la Ilustración 5-15 del diagrama de Pert nivelado, vamos a establecer qué actividades tendrán mayor prioridad usando la técnica CPM (*Critical Path Method*), que nos determinará el camino crítico de este proyecto.

Un retraso en una tarea del camino crítico implica un retraso en la fecha de terminación del proyecto. Esta información es útil justamente para dar una mayor holgura a las actividades que no estén en el camino crítico: es posible retrasar actividades que no pertenecen al camino crítico sin atrasar el proyecto. No hemos de confundir camino crítico con las actividades más importantes técnicamente, sino aquellas que por retrasar otras actividades generarían mayores problemas, de ahí la importancia de conocer cuál es el camino crítico.

INI	FIN	Actividad	Duración (meses)	Tiempos (mes desde inicio)				Holguras (meses)		
				ES(i)	EF(i,j)	LS(i,j)	LF(j)	T	L	I
1	2	A	1	0	1	0	1	0	0	0
2	3	B	0,2	1	1,2	1	1,2	0	0	0
2	4	C	0,2	1	1,2	1	1,2	0	0	0
2	5	D	0,2	1	1,2	1	1,2	0	0	0
2	11	E	1	1	2	17,94	18,94	16,94	17,94	0
2	6	F	1	1	2	4,03	5,03	3,03	0	-3,03
3	7	G	2,62	1,2	3,82	1,2	3,82	0	0	0
4	7	H	2,62	1,2	3,82	1,2	3,82	0	0	0
5	7	I	2,62	1,2	3,82	1,2	3,82	0	0	0
6	9	J	7,55	2	9,55	5,03	12,58	3,03	3,03	0
7	9	K	8,76	3,82	12,58	3,82	12,58	0	0	0
7	10	L	5,36	3,82	9,18	12,58	17,94	8,76	8,76	0
9	10	M	5,36	12,58	17,94	12,58	17,94	0	0	0
10	11	N	1	17,94	18,94	17,94	18,94	0	0	0
2	8	O	1	1	2	18,24	19,24	17,24	0	-17,24
8	12	P	0,2	2	2,2	19,24	19,44	17,24	17,24	0
11	12	Q	0,5	18,94	19,44	18,94	19,44	0	0	0
12	13	R	1	19,44	20,44	19,44	20,44	0	0	0

Tabla 5-13: Tiempos y plazos según técnica CPM

De la tabla de tiempos los valores asignados en meses de duración a cada actividad lo fueron durante el análisis con COCOMO o bien por juicio experto, mientras el inicio y el fin indican el estado del que parte

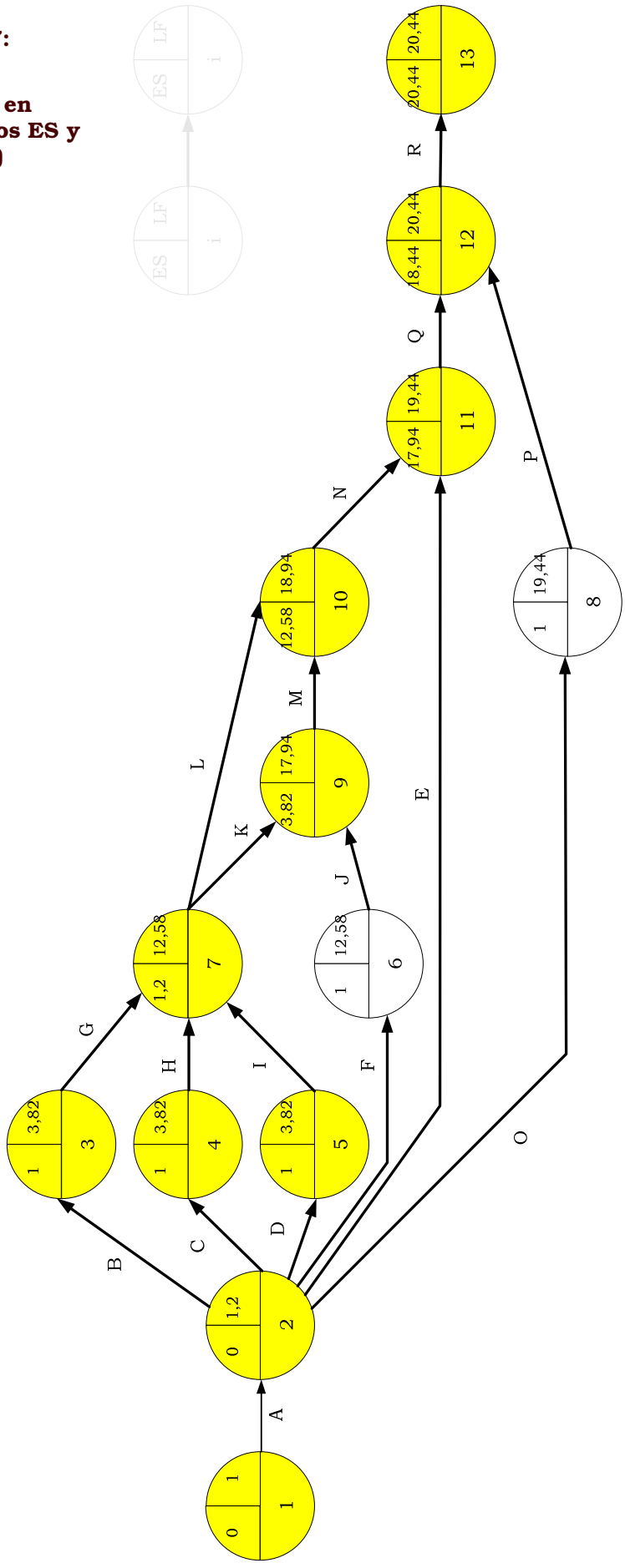
esa actividad y al que llega en el diagrama Pert nivelado que se encuentra en el Apartado 5.3 de este capítulo durante la determinación de las actividades (Ilustración 5-15). Los tiempos que se han calculado son:

- *Early Start* (ES), lo más pronto que puede comenzar la actividad tras finalizar todas las previas, resultado del máximo tiempo EF de las actividades de las que depende
- *Early Finish* (EF), lo más pronto que puede finalizar la actividad, resultado de sumar al tiempo ES la duración de esta actividad.
- *Last Start* (LS), lo más tarde que puede comenzar una actividad sin que afecte a aquellas de las que depende, que calculamos como el mínimo tiempo LF de sus sucesoras inmediatas.
- *Last Finish* (LF), lo más tarde que puede acabar una actividad, resultado de añadir al tiempo LS la duración de la actividad.

Con todos tiempos sólo nos queda por determinar las holguras para construir el grafo CPM, en el que se ha resaltado el camino crítico resultante, aquel en el que la flotación (*slack*, diferencia entre tiempo LF y tiempo EF) será cero.

Ilustración 5-17:
Diagrama CPM

(Camino crítico en amarillo, tiempos ES y LF en segundos)



5.6.2 Diagrama de Gantt

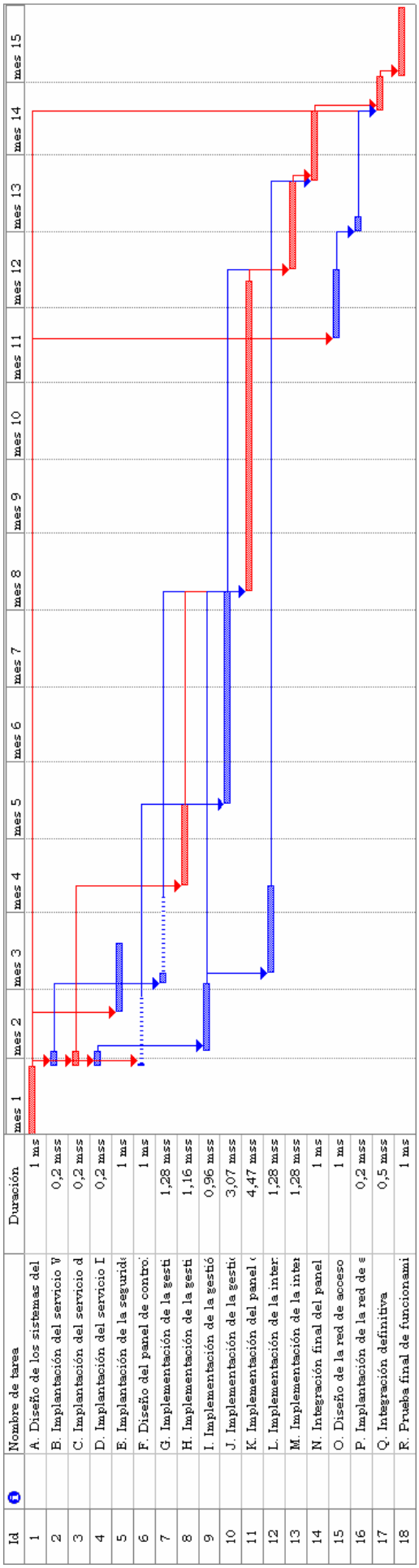
Distribuiremos ahora las actividades buscando la mayor economía posible (de los nueve programadores, gran parte de ellos sólo serán necesarios en las fases intermedias del proyecto) y partiendo de la disponibilidad de tres recursos: programadores *senior*, *junior* y de un jefe de proyecto, estando la asignación de actividades repartida según esta restricción y el orden en el que se han de realizar las actividades y el camino crítico que acabamos de obtener.

Para la representación se ha partido del supuesto que el proyecto comienza el día 1 de enero del 2003, y se han tenido en cuenta los días festivos de cara al cálculo de los plazos de ejecución. Las actividades han sido etiquetadas con la misma letra que lo fueron durante el cálculo de los costes temporales y figuran en el diagrama de Pert.

A diferencia de lo previsto mediante CPM, lograremos completar el proyecto en sólo quince meses (inicialmente eran 20,44) debido al mayor uso de recursos humanos que haremos durante el periodo central del proyecto, en el que habrá un total de diez personas colaborando en el mismo. Será durante los estados que hemos marcado en el diagrama CPM como 3, 4, 5, 7, 9 y 10, que además de estar en el camino crítico son los que permiten una mayor simultaneidad con otras tareas, especialmente del jefe del proyecto y coordinador.

La fuerza humana necesaria inicialmente y tras ese periodo de mayor esfuerzo será sensiblemente menor, de sólo tres personas. En la planificación resultante siguen siendo útiles las holguras calculadas mediante la técnica CPM, y que serán tenidas en cuenta de cara al seguimiento del proyecto, holguras que ya figuran en la tabla de tiempos convertidas en fechas concretas (recordemos que lo que hemos simulado es que el proyecto comenzaba el 1 de enero del 2003, luego finalizará a más tardar el 31 de marzo del 2004, justo 15 meses después). En el diagrama de Gantt aparecen además marcados en rojo las actividades que se encuentran en el camino crítico para hacer más comprensible cuál será el flujo prioritario de control sobre el proyecto que deberemos seguir.

Ilustración 5-18: Diagrama Gantt



Id	Nombre de tarea	Duración
1	A. Diseño de los sistemas del	1 ms
2	B. Implantación del servicio W	0,2 mss
3	C. Implantación del servicio d	0,2 mss
4	D. Implantación del servicio I	0,2 mss
5	E. Implantación de la seguridi	1 ms
6	F. Diseño del panel de contro	1 ms
7	G. Implementación de la gesti	1,28 mss
8	H. Implementación de la gesti	1,16 mss
9	I. Implementación de la gestió	0,96 mss
10	J. Implementación de la gesti	3,07 mss
11	K. Implementación del panel	4,47 mss
12	L. Implementación de la inter	1,28 mss
13	M. Implementación de la inter	1,28 mss
14	N. Integración final del panel	1 ms
15	O. Diseño de la red de acceso	1 ms
16	P. Implantación de la red de s	0,2 mss
17	Q. Integración definitiva	0,5 mss
18	R. Prueba final de funcionami	1 ms

Tabla 5-14: Programación prevista por cada actividad

Actividad	Fecha comienzo	Fecha fin	Límite comienzo	Límite fin	Demora aceptable
A. Diseño de los sistemas del ISP	Miércoles 01/01/03	Martes 28/01/03	Miércoles 01/01/03	Martes 28/01/03	0 meses
B. Implantación del servicio Web del ISP	Miércoles 29/01/03	Lunes 03/02/03	Viernes 07/03/03	Jueves 13/03/03	0 meses
C. Implantación del servicio de correo del ISP	Miércoles 29/01/03	Lunes 03/02/03	Miércoles 29/01/03	Lunes 03/02/03	0 meses
D. Implantación del servicio DNS	Miércoles 29/01/03	Lunes 03/02/03	Viernes 11/04/03	Jueves 17/04/03	0 meses
E. Implantación de la seguridad en los sistemas	Miércoles 19/02/03	Miércoles 19/03/03	Miércoles 03/03/04	Miércoles 31/03/04	13,48 meses
F. Diseño del panel de control del ISP	Martes 28/01/03	Martes 25/02/03	Miércoles 04/06/03	Miércoles 02/07/03	0 meses
G. Implementación de la gestión del servicio Web	Lunes 03/03/03	Lunes 07/04/03	Miércoles 09/04/03	Miércoles 14/05/03	1,35 meses
H. Implementación de la gestión del servicio de correo	Viernes 11/04/03	Miércoles 14/05/03	Viernes 11/04/03	Miércoles 14/05/03	0 meses
I. Implementación de la gestión DNS	Lunes 03/02/03	Lunes 03/03/03	Jueves 17/04/03	Miércoles 14/05/03	0 meses
J. Implementación de la gestión de clientes	Miércoles 14/05/03	Viernes 08/08/03	Miércoles 17/09/03	Jueves 11/12/03	4,47 meses
K. Implementación del panel de control de servicios	Viernes 08/08/03	Jueves 11/12/03	Viernes 08/08/03	Jueves 11/12/03	0 meses
L. Implementación de la interfaz con el registrador	Viernes 07/03/03	Viernes 11/04/03	Martes 16/12/03	Miércoles 21/01/04	10,13 meses
M. Implementación de la interfaz con facturación	Martes 16/12/03	Miércoles 21/01/04	Martes 16/12/03	Miércoles 21/01/04	0 meses
N. Integración final del panel de control	Miércoles 21/01/04	Miércoles 18/02/04	Miércoles 21/01/04	Miércoles 18/02/04	0 meses
O. Diseño de la red de acceso	Martes 18/11/03	Martes 16/12/03	Martes 30/12/03	Martes 27/01/04	0 meses
P. Implantación de la red de acceso	Jueves 01/01/04	Martes 06/01/04	Jueves 12/02/04	Miércoles 18/02/04	1,51 meses
Q. Integración definitiva	Miércoles 18/02/04	Miércoles 03/03/04	Miércoles 18/02/04	Miércoles 03/03/04	0 meses
R. Prueba final de funcionamiento	Miércoles 03/03/04	Miércoles 31/03/04	Miércoles 03/03/04	Miércoles 31/03/04	0 meses

Los costes del proyecto se calcularon para los gastos de personal que habíamos indicado en la Tabla 5-11 durante el análisis de los recursos humanos. A partir de esos datos y de las necesidades de personal de cada actividad, se obtiene un coste total de 323.898,48 € para las 4.801,43 horas de trabajo que serán necesarias, a lo que habrá que sumar los costes fijos que habíamos calculado para los recursos humanos, que en este diagrama Gantt no hemos estimado.

Tabla 5-15: Resumen costes laborales por actividad

Actividad	Horas trabajo	Coste
A. Diseño de los sistemas del ISP	240 horas	8.000,00 €
B. Implantación del servicio Web del ISP	36 horas	1.119,98 €
C. Implantación del servicio de correo del ISP	20,17 horas	806,40 €
D. Implantación del servicio DNS	8 horas	320,00 €
E. Implantación de la seguridad en los sistemas	80 horas	2.400,00 €
F. Diseño del panel de control del ISP	2,48 horas	99,10 €
G. Implementación de la gestión del servicio Web	56,03 horas	4.163,24 €
H. Implementación de la gestión del servicio de correo	234,8 horas	21.376,07 €
I. Implementación de la gestión DNS	186,95 horas	17.462,27 €
J. Implementación de la gestión de clientes	908,83 horas	67.506,63 €
K. Implementación del panel de control de servicios	1.439,85 horas	102.627,45 €
L. Implementación de la interfaz con el registrador	419,88 horas	29.924,61 €
M. Implementación de la interfaz con facturación	433,08 horas	30.400,61 €
N. Integración final del panel de control	192 horas	17.760,00 €
O. Diseño de la red de acceso	262,28 horas	9.468,60 €
P. Implantación de la red de acceso	35,08 horas	1.083,48 €
Q. Integración definitiva	80 horas	2.800,00 €
R. Prueba final de funcionamiento	166 horas	6.580,00 €
TOTAL	4.801,43	323.898,44 €

Este resumen de costes por actividad muestra el cómputo de gastos laborales para cada actividad a partir de la suma de las horas dedicadas por cada grupo de personal que hemos tenido en cuenta (jefe de proyecto, programador *senior* y programador *junior*). Para ver concretamente cuántas horas dedica cada persona a cada actividad y en qué fechas, se ha incluido la Tabla 5-16.

Tabla 5-16: Asignación de recursos detallada por actividad

Actividad	Horas	Duración	Comienzo	Fin
A. Diseño de los sistemas del ISP	240	1 mes	01/01/03	28/01/03
Jefe de proyecto	160		01/01/03	28/01/03
Programadores *senior*	80		01/01/03	28/01/03
B. Implantación del servicio Web del ISP	36	0,2 meses	29/01/03	03/02/03
Jefe de proyecto	32		29/01/03	03/02/03
Programadores *senior*	4		29/01/03	29/01/03
C. Implantación del servicio de correo del ISP	20,17	0,2 meses	29/01/03	03/02/03
Programadores *senior*	20,17		29/01/03	03/02/03
D. Implantación del servicio DNS	8	0,2 meses	29/01/03	03/02/03
Programadores *senior*	8		29/01/03	03/02/03
E. Implantación de la seguridad en los sistemas	80	1 mes	19/02/03	19/03/03
Jefe de proyecto	80		19/02/03	19/03/03
F. Diseño del panel de control del ISP	2,48	1 mes	28/01/03	25/02/03
Jefe de proyecto	0		28/01/03	28/01/03
Programadores *senior*	2,48		29/01/03	25/02/03
G. Implementación de la gestión del servicio Web	56,03	1,28 meses	03/03/03	07/04/03
Jefe de proyecto	6,43		03/03/03	06/03/03
Programadores *senior*	19,05		03/03/03	07/04/03
Programadores *junior*	30,55		03/03/03	07/03/03
H. Implementación de la gestión del servicio de correo	234,8	1,16 meses	11/04/03	14/05/03
Jefe de proyecto	8		11/04/03	21/04/03
Programadores *senior*	41,2		11/04/03	18/04/03
Programadores *junior*	185,6		11/04/03	14/05/03
I. Implementación de la gestión DNS	186,95	0,96 meses	03/02/03	03/03/03
Jefe de proyecto	0		03/02/03	03/02/03
Programadores *senior*	33,35		04/02/03	10/02/03
Programadores *junior*	153,6		04/02/03	03/03/03
J. Implementación gestión de clientes	908,83	3,07 meses	14/05/03	08/08/03
Jefe de proyecto	77,47		14/05/03	08/07/03
Programadores *senior*	340,17		14/05/03	14/07/03
Programadores *junior*	491,2		14/05/03	08/08/03
K. Implementación del panel de control de servicios	1.439,85	4,47 meses	08/08/03	11/12/03
Jefe de proyecto	145,45		08/08/03	18/11/03
Programadores *senior*	579,2		08/08/03	18/11/03
Programadores *junior*	715,2		08/08/03	11/12/03
L. Implementación de la interfaz con el registrador	419,88	1,28 meses	07/03/03	11/04/03
Jefe de proyecto	18,28		07/03/03	20/03/03
Programadores *senior*	196,8		07/03/03	10/04/03
Programadores *junior*	204,8		07/03/03	11/04/03
M. Implementación de la interfaz con facturación	433,08	1,28 meses	16/12/03	21/01/04
Jefe de proyecto	23,48		16/12/03	01/01/04
Programadores *senior*	204,8		16/12/03	21/01/04
Programadores *junior*	204,8		16/12/03	21/01/04

Actividad	Horas	Duración	Comienzo	Fin
N. Integración final del panel de control	192	1 mes	21/01/04	18/02/04
Jefe de proyecto	32		21/01/04	12/02/04
Programadores *junior*	160		21/01/04	18/02/04
O. Diseño de la red de acceso	262,28	1 mes	18/11/03	16/12/03
Jefe de proyecto	102,28		18/11/03	05/12/03
Programadores *senior*	160		18/11/03	16/12/03
P. Implantación de la red de acceso	35,08	0,2 meses	01/01/04	06/01/04
Jefe de proyecto	32		01/01/04	06/01/04
Programadores *senior*	3,08		01/01/04	06/01/04
Q. Integración definitiva	80	0,5 meses	18/02/04	03/03/04
Jefe de proyecto	40		18/02/04	03/03/04
Programadores *senior*	40		18/02/04	03/03/04
R. Prueba final de funcionamiento	166	1 mes	03/03/04	31/03/04
Jefe de proyecto	6		03/03/04	26/03/04
Programadores *senior*	160		03/03/04	31/03/04

Tal y como está previsto, tanto los programadores *senior* como el jefe del proyecto se espera continúen en el interior de la empresa por tiempo indefinido con tareas más de mantenimiento y atención técnica a los clientes, mientras que el conjunto de los siete programadores *junior* será contratado de manera puntual para cubrir las actividades de la G hasta la N y por tanto este coste laboral se limitará a las fechas de calendario en que se encuentren estas actividades, lo que implica sólo un año del total de los quince meses previstos.

Tabla 5-17: Resumen costos laborales

Tipo trabajador	Horas totales	Coste total	Fecha inicio	Fecha fin
Jefe de proyecto	763,42	22.902,66 €	01/01/2003	31/03/2004
Programadores *senior*	3.784,56	75.691,49 €	01/01/2003	31/03/2004
Por programador (2)	*1.892,28*	*37.845,75 €*		
Programadores *junior*	15.020,25	225.304,28 €	03/03/2003	18/02/2004
Por programador (7)	*2.145,75*	*32.186,33 €*		

Tabla 5-18: Cronograma de asignación de recursos

Id	Nombre de tarea	mes 1	mes 2	mes 3	mes 4	mes 5	mes 6	mes 7	mes 8	mes 9	mes 10	mes 11	mes 12	mes 13	mes 14	mes 15
1	A. Diseño de los sistemas del	240h														
	Jefe de proyecto	160h														
	Programadores senior	80h														
2	B. Implantación del servicio V	28h	8h													
	Jefe de proyecto	24h	8h													
	Programadores senior	4h														
3	C. Implantación del servicio d	15,12h	5,03h													
	Programadores senior	15,12h	5,03h													
4	D. Implantación del servicio D	6h	2h													
	Programadores senior	6h	2h													
5	E. Implantación de la seguridi		29,27h	50,73h												
	Jefe de proyecto		29,27h	50,73h												
6	F. Diseño del panel de control	2,48h	0h													
	Jefe de proyecto	0h														
	Programadores senior	2,48h	0h													
7	G. Implementación de la gesti			56,03h	0h											
	Jefe de proyecto			6,43h												
	Programadores senior			19,05h	0h											
	Programadores junior			30,55h												
8	H. Implementación de la gesti				156,25h	78,55h										
	Jefe de proyecto				8h											
	Programadores senior				41,2h											
	Programadores junior				107,05h	78,55h										
9	I. Implementación de la gestió		185,35h	1,6h												
	Jefe de proyecto		0h													
	Programadores senior		33,35h													
	Programadores junior		152h	1,6h												
10	J. Implementación de la gesti					219,25h	378h	269,83h	41,75h							
	Jefe de proyecto					24,37h	42h	11,12h								
	Programadores senior					97,45h	168h	74,72h								
	Programadores junior					97,45h	168h	184h	41,75h							
11	K. Implementación del panel d								284,05h	396h	414h	276,85h	68,95h			
	Jefe de proyecto								31,57h	44h	46h	23,88h				
	Programadores senior								126,25h	176h	184h	92,95h				
	Programadores junior								126,25h	176h	184h	160h	68,95h			
12	L. Implementación de la inter			289,98h	129,92h											
	Jefe de proyecto			18,28h												
	Programadores senior			135,85h	60,95h											
	Programadores junior			135,85h	68,95h											
13	M. Implementación de la inter												204,85h	228,23h		
	Jefe de proyecto												22,77h	0,73h		
	Programadores senior												91,05h	113,75h		
	Programadores junior												91,05h	113,75h		
14	N. Integración final del panel													77,8h	114,2h	
	Jefe de proyecto													15,57h	16,43h	
	Programadores junior													62,25h	97,75h	
15	O. Diseño de la red de acceso											134,08h	128,2h			
	Jefe de proyecto											67,05h	35,25h			
	Programadores senior											67,05h	92,95h			
16	P. Implantación de la red de s													35,08h		
	Jefe de proyecto													32h		
	Programadores senior													3,08h		
17	Q. Integración definitiva														62,25h	17,75h
	Jefe de proyecto														31,12h	8,88h
	Programadores senior														31,12h	8,88h
18	R. Prueba final de funcionami															166h
	Jefe de proyecto															6h
	Programadores senior															160h

5.7 Resumen de costes y plazos

El plazo de ejecución del proyecto son 15 meses a contar desde el día 1 de enero del 2.003, por lo que éste finalizará el 31 de marzo del 2.004. Los costes totales estimados se distribuyen entre costes fijos (hardware, fundamentalmente) y costes laborales.

Los costes de adquisición del hardware se habrán de producir en el primer mes desde el inicio del proyecto tras la actividad de diseño del mismo.

Tabla 5-19: Resumen de gastos

Concepto	Gastos durante el proyecto
Gastos iniciales en hardware y acondicionamiento del ISP (detallados en el apartado 5.5.1)	8.000 €
Gastos del personal fijo (2 programadores *senior* y el jefe de proyecto)	98.594,15 €
Gastos del personal contratado para las actividades de la G a la N	225.304,28 €
	331.898,43 €

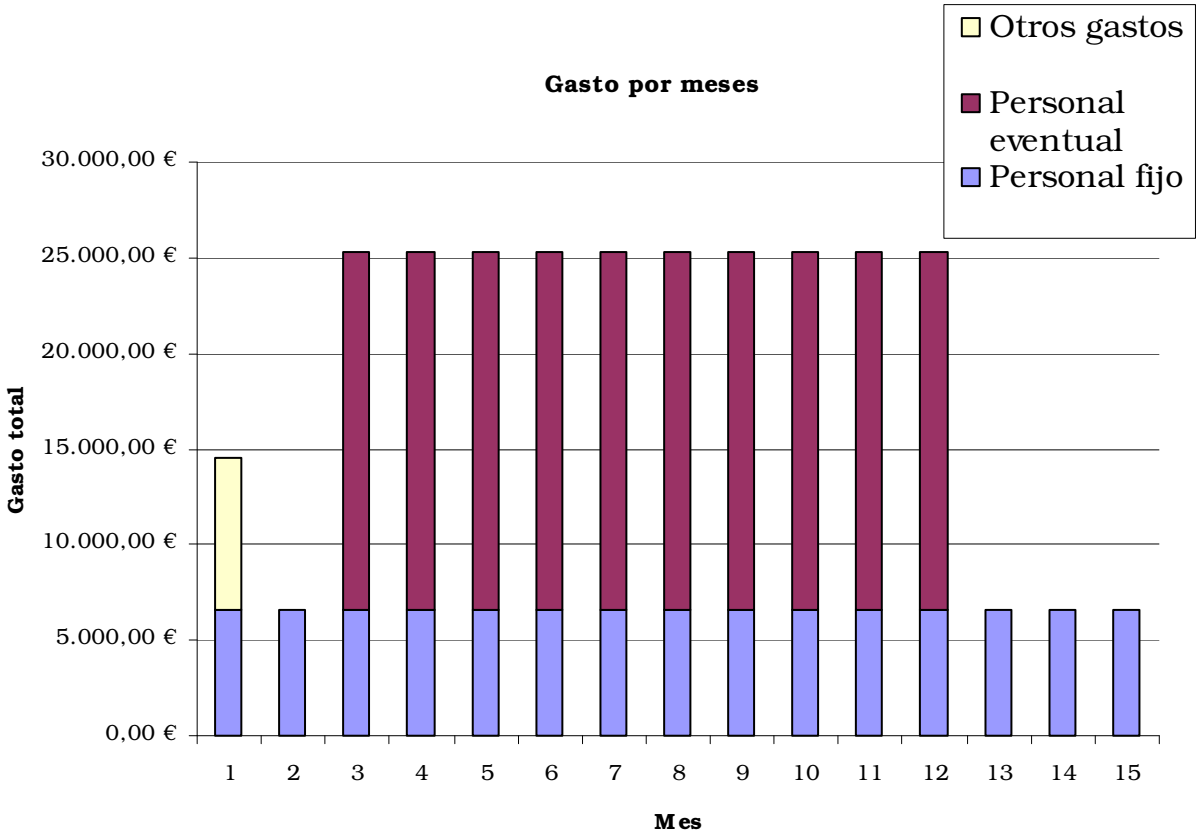


Ilustración 5-19: Distribución por meses del gasto

Evidentemente toda esta planificación hace que el proyecto total, contando todas las partes que se habrían de desarrollar, exceda la capacidad de este Proyecto de Fin de Carrera que se trata en la presente memoria.

Habrá por tanto que hablar de los objetivos mínimos que se cumplirán al finalizar el proyecto, que serán las tareas de análisis y diseño tanto del ISP como del panel de control que se pretende desarrollar. Respecto a la implementación del Proyecto, esta no será posible más que con la puesta en marcha y adquisición de los sistemas y conectividad previstos.

Es por ello por lo que en la fase de implementación del proyecto se pueden encontrar detalladas las configuraciones que se realizarían en los sistemas, que es uno de los objetivos del proyecto, pero sin adjuntar los listados y configuraciones completas, aunque en el sistema anfitrión en el que se desarrolla se han probado y verificado.

Por otro lado, por lo que respecta a la implementación del panel de control, se realizará ésta de modo parcial, sin llegar a implementar todas las pantallas y comandos para la totalidad de los servicios, concentrándonos por tanto únicamente en los servicios de correo electrónico y DNS (por requerirlo el correo).

Tampoco se realizará la implementación de la interfaz con el registrador de dominios ni la integración con el sistema de facturación, por la misma razón esgrimida.

Planteándose sólo como objetivo de este proyecto la implementación de ciertas fases, así como el análisis y el diseño completos, el diagrama de Gantt quedaría como sigue:

Actividad	Duración	Comienzo	Fin
A. Diseño de los sistemas del ISP	1 mes	01/01/03	28/01/03
B. Implantación del servicio Web del ISP	0,2 meses	29/01/03	03/02/03
C. Implantación del servicio de correo del ISP	0,2 meses	29/01/03	03/02/03
D. Implantación del servicio DNS	0,2 meses	29/01/03	03/02/03
E. Implantación de la seguridad en los sistemas	1 mes	04/02/03	04/03/03
F. Diseño del panel de control del ISP	1 mes	28/01/03	25/02/03
H. Implementación de la gestión del servicio de correo	1,16 meses	07/03/03	10/04/03
I. Implantación de la gestión del servicio de DNS	0,96 meses	11/04/03	10/05/03
O. Diseño de la red de acceso	1 mes	11/05/03	12/06/03
P. Implantación de la red de acceso	0,2 meses	13/06/03	19/06/03
Q. Integración	0,5 meses	20/06/03	5/07/03
TOTAL	7,42 meses		

Será por tanto ya abarcable como objetivo de este Proyecto de Fin de Carrera el realizar el conjunto de actividades descritas en la tabla superior, quedando el resto de actividades planificadas sujetas a una futura puesta en práctica de la totalidad del proyecto.

6 Diseño

6.1 Diseño de la red

El diseño de la red no consiste sólo en la simple interconexión de los servidores y los equipos de trabajo, sino que va más allá, buscando evitar posibles futuros problemas que resultarían muy problemáticos, especialmente por nuestra condición de ISP: colisiones, congestión, seguridad en la red, control del ancho de banda, etc. Aunque el control del ancho de banda y la seguridad pueden ser consideradas tareas relacionadas con la configuración de los sistemas informáticos que vayamos a usar, serán tratadas en este apartado en la medida en la que esta seguridad y control vayan a ser implantadas en el equipamiento de la red (routers) y no en los servidores.

Los objetivos del diseño serán tanto la funcionalidad de la red (proporcionar conectividad a los clientes con una velocidad y fiabilidad razonables), como la escalabilidad (permitir posteriores crecimientos en el número de equipos implicados) y la adaptabilidad (evitando limitar la implantación de futuras tecnologías, como IPv6).

En cualquier caso lo que no va a ser discutido es el protocolo a utilizar en nuestra red, ya que IPv4 es por ahora la única posibilidad hasta que se produzca una popularización mayor de IPv6, protocolo ya bastante maduro y en funcionamiento en muchas áreas de Internet mediante *gateways* que enlazan estas redes IPv6 con las tradicionales IPv4. Tampoco va a resultar elegible la topología de la red a instalar, ya que en ningún caso rentaría tomar una implementación que no sea Ethernet, por razones de coste (Ethernet se encuentra tan ampliamente disponible a nivel de hardware, software y conocimientos que supera con creces la desventaja de ser una red de acceso compartido). Dentro de las posibles tecnologías Ethernet a utilizar, se usará 100BaseTX, tanto en la red de servidores como en la red de equipos de trabajo, por su bajo coste y considerarse que el ancho de banda necesario no será superado con este tipo de instalación.

En la terminología habitual en el diseño de redes hay dos términos que usaremos habitualmente y deberemos definir antes de continuar:

- POP (que viene del inglés *Point of Presence*) corresponde al término usado para hablar del armario hasta el que llega el cableado de los operadores que nos suministran nuestro ancho de banda, y en el que puede haber o no hardware de dichos operadores. En nuestro caso, dado el tamaño del ISP que vamos a crear, este armario concentrará toda nuestra conectividad, con independencia del operador, sin dedicar un POP a cada conexión.

- MDF es el acrónimo en inglés del armario de distribución principal. Al disponer de una única oficina y de pequeño tamaño el MDF será el único punto de conectividad junto al POP, estando situados ambos dentro de la propia zona de servidores. En redes de mayor tamaño es común que el MDF constituya un área independiente y haya además armarios secundarios (IDF), pero no va a ser el caso dadas las futuras dimensiones de nuestra red.

El único servidor que hemos estimado necesario para el ISP estará situados dentro del propio MDF, dentro de un área de captación que dará lugar a una red de difusión en el que todos sus elementos dispondrán de IP pública en Internet (uno por ahora, pero no limitaremos un futuro crecimiento en el número de servidores, bien sea por realizar *housing* de otros servidores, bien por necesitar otros servidores para mantener el servicio).

Los equipos de trabajo del personal de la empresa constituirán la otra red que habremos de diseñar, con IPs privadas y que estarán situados en otra área de captación diferenciada en el MDF. Esta red de ordenadores de trabajo será la que denominaremos red local, red cuyo dominio de difusión estará diferenciado del que usaremos para la red de servidores: aunque de momento haya un único servidor es importante que el tráfico de Internet de nuestros clientes sea separado del generado por la propia actividad de nuestro personal, bien físicamente (lo que supondría un coste elevado en más equipamientos y conectividades) o bien mediante una acotación del mismo, que es lo que haremos: Habrá por tanto un área de servidores en el MDF, pero conectaremos a la red de trabajo de manera indirecta, a través de esta red de servidores.

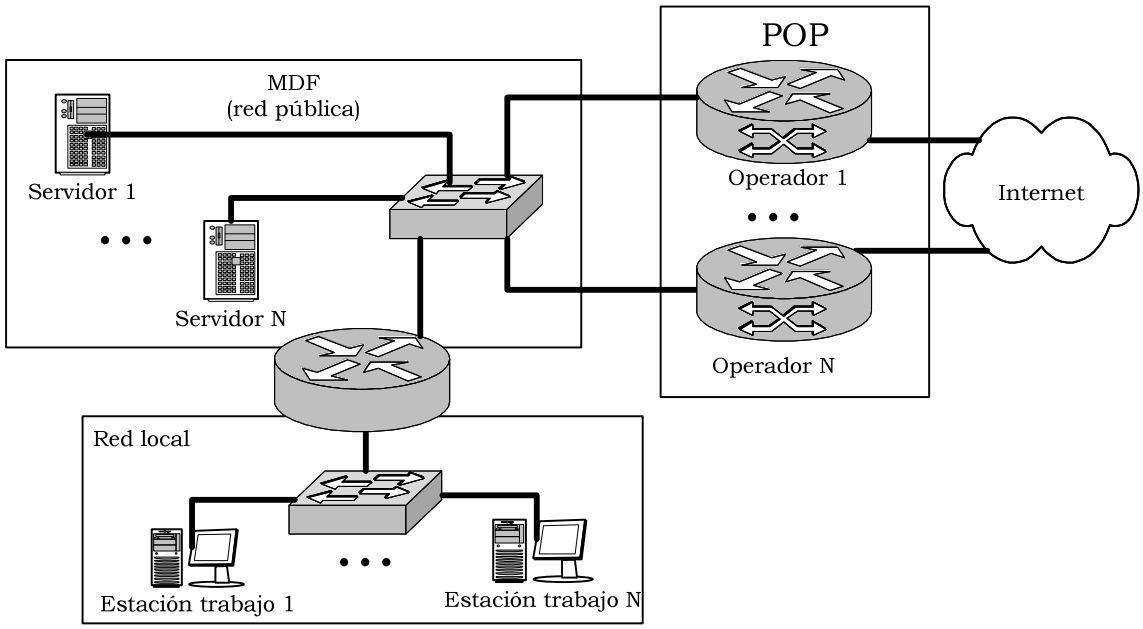


Ilustración 6-1: Esquema lógico de la red propuesto

6.1.1 Conectividad a Internet

El primer aspecto es decidir el tipo, calidad y número de accesos a Internet de qué dispondrá el ISP, para lo que necesitamos antes estimar el ancho de banda que nos hará falta, y que variará en función del número de clientes que tengamos.

Tradicionalmente en el acceso a Internet las empresas han buscado un único proveedor, buen ancho de banda en el mismo y una fiabilidad razonable, pagándose un sobrecoste para lograr esa supuesta calidad. Hablamos de supuesta por el hecho de que una conexión de teórica mejor calidad también falla, y en nuestra condición de ISP habremos también de seguir la tónica generalizada entre los proveedores de acceso y de contenidos de reducir este riesgo diversificando su conectividad entre varios operadores.

En nuestra condición de pequeño operador no podemos esperar por otro lado que en caso de fallo de un operador podamos redirigir el tráfico al otro mediante protocolos de enrutamiento dinámicos: más bien se busca que los servicios se distribuyan de una manera ordenada entre diferentes conexiones y el ISP no se vea sometido a un aislamiento total en caso de fallo.

Además, cualquiera que sea el proveedor al que acudamos, habremos de tener capacidad de crecer en el mismo cuando nuestra empresa lo haga, pudiéndose por tanto iniciar la actividad con un menor caudal de acceso.

A partir de las cifras consideradas de sitios Web alojados y usuarios de correo, calcularemos de forma aproximada cuál es el ancho de banda mínimo que habremos de contratar, distinguiéndolo entre tráfico destinado a correo y tráfico destinado a Web, con un margen de seguridad en ambos casos para otros usos del ancho de banda, ancho de banda que se calculará para los momentos de máxima concentración de tráfico, que es cuando debemos garantizar un funcionamiento adecuado del ISP. Esta franja de tráfico elevado coincide con las horas diurnas que van desde las siete de la mañana hasta las nueve o diez de la noche, e incluso dentro de esta franja horaria, el mayor número de accesos estarán concentrados por la mañana: básicamente el comportamiento del consumo de ancho de banda coincide con el comportamiento del consumo eléctrico.

Esta ecuación nos dará un valor: el consumo medio esperado por cada servicio y usuario activo, que multiplicado por nuestra previsión de usuarios iniciales nos dará el ancho de banda que habremos de contratar en un primer momento. Dado que no habrá sobrecoste alguno por ampliar la conectividad de un operador, el que la previsión inicial de ancho de banda resultara baja en la realidad sería fácil de solventar.

Distinguiremos entre los sitios Web de gran tráfico y aquellos con poco o nulo. La distinción entre ambos tipos de sitios será muy simple: por Web grande entenderemos aquellas que van a tener al menos 100 accesos únicos al día (unos cuatro accesos a la hora), mientras que Web pequeña serán las demás.

Aunque parezca escasa la cifra utilizada, hay que tener en cuenta que no pretendemos alojar en nuestros servidores sitios con un tráfico muy alto, y en especial se busca atraer a aquellos clientes que actualmente no disponen de Web porque no la ven rentable a los precios que se encuentra el mercado: estamos ante clientes que si se les da la oportunidad montarán un sitio Web de poca entidad y que no va a suponer demasiado tráfico, sitios Web que además pesarán en total entre uno y cinco megabytes, y que por tanto no van a suponer un gran tráfico:

Tabla 6-1: Previsión en tamaño y accesos por sitio Web

	Media de accesos por día	Peso de la Web
Web grande	200 visitas/día	5 MB + base de datos
Web pequeña	40 visitas/día	1 MB

Podríamos ya estimar una cifra aprox. a partir del tamaño de los sitios Web que alojaremos y el número de accesos supuestos. Pero este valor no es de por sí suficiente para decidir, porque no reflejaría la realidad del consumo, el cuál está concentrado en la franja que va desde las ocho de la mañana a las diez de la noche (y con valles en las horas de las comidas), variando además en función del día de la semana, tal y como muestra esta gráfica extraída de un sitio Web de tamaño medio y con elevado número de accesos alojado en el ISP en el que actualmente desarrollo mi actividad profesional. Por todo ello lo que haremos será suponer que las esperadas cifras de visitas medias por día y cantidad de *bytes* descargados en cada caso (que sería un 5% del peso de la Web, ya que cada cliente no se descarga la Web completa) están todas ellas concentradas en las horas de mayor tráfico (día laborable y dentro del día durante la mañana o la tarde):

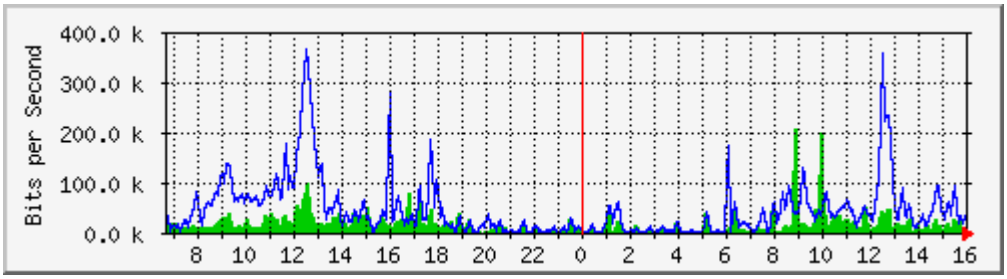


Ilustración 6-2: Ejemplo de gráfica con el tráfico en un día laboral

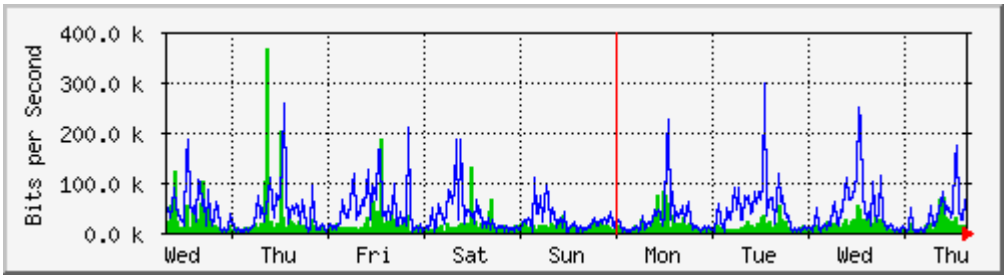


Ilustración 6-3: Ejemplo de gráfica con el tráfico semanal

Estas gráficas están realizadas con un programa bastante famoso: el MRTG, con licencia igualmente GNU y que posteriormente emplearemos nosotros mismos en nuestro propio software de gestión para extraer gráficas de consumo.

De ellas se extraerían los siguientes valores de consumo para cada sitio Web a partir del producto entre los accesos esperados y el 5% del peso (supondremos que cada acceso producido descarga un parcial muy bajo del total del espacio de la Web). Se ha considerado que el consumo se realiza en las 12 horas consideradas punta según lo comentado anteriormente, con lo que los 250 MB se descargarían en esas 12 horas y por tanto incrementan el ancho de banda necesario, según lo reflejado en la Tabla 6-2.

Tabla 6-2: Costes de ancho de banda del acceso Web

		Total consumo diario	Ancho de banda
Web grande	200 visitas/día * 0,25 MB descargados	50 MB	9,48 *Kbits*/segundo
Web pequeña	40 visitas/día * 0,25 MB descargados	4 MB	0,76 *Kbits*/segundo

Antes de continuar conviene recordar la convención que se sigue en conectividad para facilitar distinguir entre *bytes* (unidad habitual en la Informática) y bits (unidad de medida por excelencia del ancho de banda): cuando tengamos *KB*, la letra be mayúscula indicará que se trata de bytes, mientras *Kb*, por ser la letra be minúscula, significará bits.

La controversia con los valores de consumo supuestos vendrá por no haberse considerado los probables picos que se van a producir en el servicio, es decir: que en determinado instante coincidan peticiones que hagan que el ancho de banda sea insuficiente. Esto no es del todo cierto: el acceso a los sitios Web no va a seguir una curva continua en el tiempo ni habrá simultaneidad en el acceso a dos sitios de manera continuada, y por tanto todo o casi todo el ancho de banda disponible va a estar ahí en un instante de tiempo para servir al cliente una página solicitada, de ahí que no consideremos relevante analizar de manera concreta los picos de consumo, aunque para mayor seguridad contrataremos un excedente de al menos el 25% sobre el ancho de

banda considerado necesario, para utilizarlo no sólo como margen de seguridad, sino como ancho de banda dedicado para el servicio DNS, que no vamos a analizar de manera particular por considerarlo de bajo consumo, pero para el que aunque poco, tenga que haber un ancho de banda.

Una alternativa hubiera sido la incorporación a la gráfica usada para el análisis, de valores que mostraran cuáles eran los máximos de consumo durante los periodos de medición, de una manera similar a lo que muestra en la Ilustración 6-4 los valores en rojo y en verde oscuro, como son el *maximal* de consumo entrante y saliente (los picos de consumo).

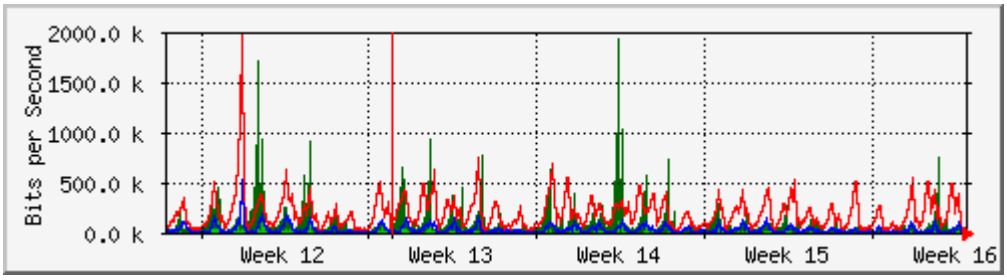


Ilustración 6-4: Ejemplo de gráfica de consumo con datos de maximales

Los picos son puntuales casi siempre, y alcanzan su máximo de 2 *Mbits* (que es el ancho de banda de esta conexión) en repetidas ocasiones. Si nuestra suposición es cierta esos picos corresponderían a un único usuario que sería el que estaría en ese instante ocupando la mayor parte del canal, y cuando acudiera el siguiente cliente el canal estaría de nuevo disponible, sin merma del ancho de banda.

Ancho de banda para correo

Para el correo la suposición de que existen dos tipos de usuarios no va a ser necesaria: a menos que la cuenta esté inactiva es bastante probable que tanto en un entorno empresarial como en un entorno doméstico el usuario del correo reciba más o menos igual número de correos (el primero por sus relaciones laborales, el segundo por estar suscrito a listas, etc.).

Por tanto en este caso si que vamos a tratar de establecer las necesidades directamente a partir de estadísticas previas de las que dispongamos, considerando además en las mismas no sólo los correos entregados a los usuarios, sino el tráfico de correos rechazados y cualquier actividad que genere tráfico y que esté relacionada con el correo (como por ejemplo sería la posterior recuperación del correo a través de POP3 o protocolos como IMAP e incluso desde el mismo *WebMail*).

En concreto, vamos a ayudarnos de los datos disponibles en el mismo ISP en el que trabajo, cuyo sistema de correo soporta una media de 30.000 mensajes / día (de los que pueden considerarse aceptados por el servidor un 30% de media si filtramos esos correos, ya que la

mayoría de las veces el *spam* es incapaz siquiera de pasar reglas de comprobación como que la dirección del remitente sea la correcta).

Los datos corresponden a una jornada normal, y al igual que ocurre con el servicio Web, se ve claramente una bajada hasta casi anularse a partir de las diez de la noche. De las gráficas, las dos primeras mostrarían el tráfico saliente desglosado según el puerto de origen y que tiene relación con el correo, mientras que la última representaría el tráfico entrante, siendo la media en esas doce horas de 118 *Kbps* para el tráfico entrante y de 72 para el saliente. He aquí una de las principales distinciones con respecto al tráfico Web: mientras en el Web todo el tráfico considerado es saliente (enviamos una página a un posible cliente), en el caso de correo necesitaremos ancho de entrada (para recibir el correo destinado a nuestros usuarios y aquel que estos enviarán a Internet) y ancho de salida (para entregar correo a terceros o a nuestros clientes mediante protocolo POP, IMAP o *Webmail*). Este tráfico lo orientaremos hacia las conexiones de menor coste para la empresa.

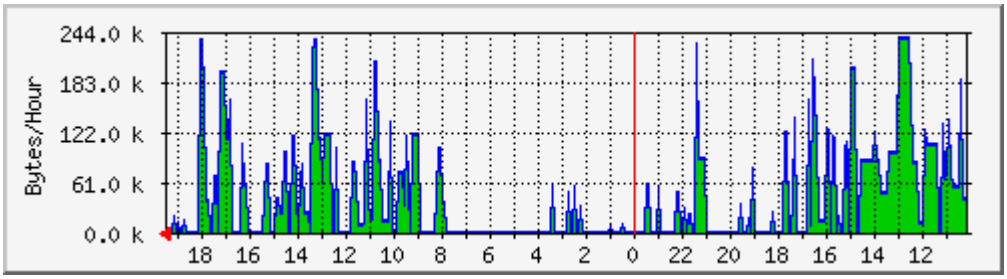


Ilustración 6-5: Ejemplo de tráfico saliente generado por POP e IMAP

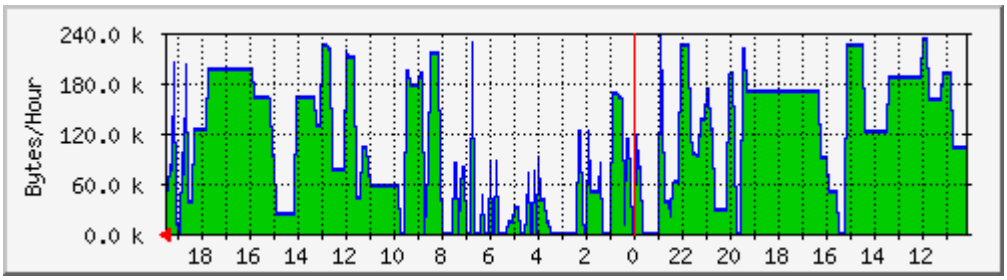


Ilustración 6-6: Ejemplo de tráfico saliente generado por SMTP

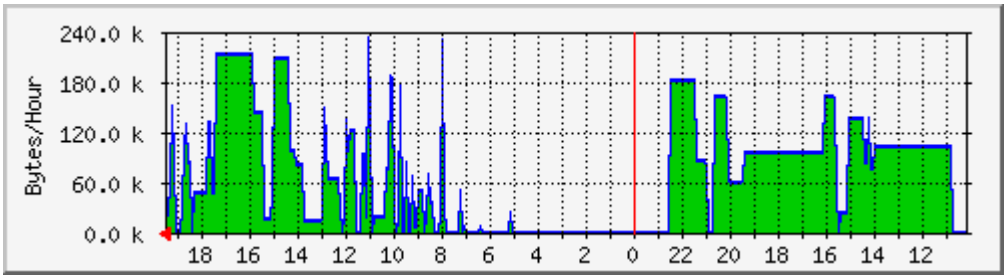


Ilustración 6-7: Ejemplo de tráfico saliente generado por SMTP

Con estos valores, y supuesto un nicho de 400 usuarios vivos del correo (es decir: cuya cuenta tiene tráfico diariamente) consideraremos

que para el servicio de correo nos hará falta 0,39 *Kbps* por usuario de ancho de bajada y 0,18 de saliente.

De ese valor para el tráfico entrante y el saliente no necesitaremos una total disponibilidad del mismo tal y como ocurría con la Web, ya que el *relay* de correo entre los servidores de correo puede ofrecerse con anchos de banda menores sin que ello sea apreciado por los clientes más que en un imperceptible retraso en la entrada de un correo nuevo que le hayan enviado. Estamos por tanto hablando de dos tipos de tráfico de correo: el que nuestro servidor realiza directamente con nuestro cliente, y cuyo canal de acceso no puede sufrir saturación, y el canal por el que luego el servidor intercambiará con otros ese correo y que no nos es tan prioritario.

Posteriormente, cuando hablemos de las características del ancho de banda, veremos que de estos 0,39 *Kbps* de bajada y 0,18 *Kbps* nos harán falta no más de la mitad de cada valor garantizado.

Tabla 6-3: Anchos de banda estimados del correo

(por cuenta de correo)	Con el cliente	Con Internet
Ancho de bajada	0,19 *Kbps*	0,23 *Kbps*
Ancho de subida	0,09 *Kbps*	0,12 *Kbps*

En realidad no es tan simple y habría que matizar estas cifras: estamos ante unos servicios que por el mero hecho de estar en Internet, van a ser constantemente atacados por *spammers* y virus, lo que nos conduce a que con independencia del tamaño del ISP y del número de clientes alojados, sufriremos un gasto de ancho de banda con el correo que no guardará relación con los usuarios, y que será rechazado en muchas ocasiones o no, pero supondrá consumo de ancho de banda, de ahí que el ancho en Internet haya sido incrementado en un 25%, para acometer esta. De hecho este consumo inevitable de ancho de banda que se produce en un intercambiador de correo resulta más doloso para un ISP de tamaño pequeño, ya que proporcionalmente no crece con los usuarios reales del servidor en igual proporción: aunque un operador de gran tamaño enfoque la lucha contra el SPAM como algo estratégico (porque daña su imagen hacia los clientes) el tráfico que ese correo basura les generaría es menor que el que a un ISP de tamaño pequeño le puede suponer un solo de esos *spammers*.

Como el SPAM trataremos de evitarlo filtrando y comprobando la mayor parte de las cabeceras y analizando el contenido, esta situación se reducirá en parte, por lo que una vez más contrataremos por encima de los valores previstos para obtener un margen de seguridad con el que acometer estos posibles problemas.

Tras analizar los anchos de banda que por cliente esperamos nos hagan falta consideraremos también que podemos ofrecer estos servicios a través de conexiones diferentes, con diferentes parámetros de calidad, considerando además que habrá otros consumos de ancho de banda que hasta ahora no hemos previsto (DNS, conexión remota para administración, etc.).

Es ahora momento de nombrar dos parámetros importantes a considerar para contratar una conectividad: CIR y PIR. El primero significa *Commited Information Rate*, y representa el porcentaje del ancho de banda contratado que contiene una garantía por parte del operador de que en cualquier momento al que se use el mismo estará disponible para nuestro ISP. El PIR o *Peak Information Rate*, representa la máxima tasa de transferencia a la que los paquetes pueden ser transferidos por nuestra conexión, es decir: cuál será el pico más alto del consumo de ancho de banda que el operador garantiza que el circuito no va a cortar.

Un valor CIR de cero es justamente lo que los operadores vienen ofreciendo en las conexiones a Internet particulares: se trata de conexiones cuyo ancho de banda no viene respaldado por la existencia de una cantidad similar por cliente en las instalaciones del operador que las suministra. En cambio, cuando un operador ofrece una conexión con CIR se espera de él que tenga dedicado ese ancho de banda a la conexión que con él contrataremos, garantizándonos que no va a ser usada por cualquier otro cliente del operador.

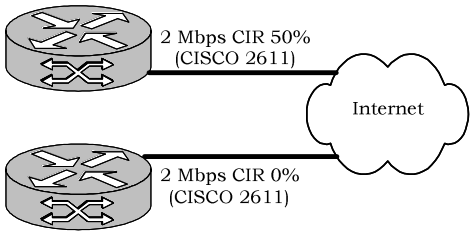
El reutilizar el ancho de banda es muy común: hay que tener en cuenta que en este sentido Internet tiene un comportamiento similar al que tiene una compañía de aguas o energética: el sistema funciona mientras todos sus miembros no accedan de manera simultánea a los servicios, momento en que el servicio se colapsaría. El controlar que nuestro ancho de banda sea siempre creciente cuando las circunstancias así lo requieran para que no se presenten estos colapsos será una de las actividades cruciales, y en esta fase de diseño será nuestra responsabilidad asegurarnos de que tampoco nos pasamos.

Estos parámetros de CIR y PIR se presentarán en el SLA que el ISP habrá de firmar con sus proveedores. El SLA o *Service Level Agreement* es el documento que definirá los compromisos y garantías sobre estos a que el proveedor se compromete por escrito, y hoy por hoy constituyen la única forma de negociar una calidad mínima en Internet.

Es cierto que tratándose de caudal de ancho de banda éste es difícil de medir en su totalidad, y que la mayoría de la letra escrita que figura en un SLA constituye papel mojado en caso de fallo del operador si operador e ISP cliente no se ponen de acuerdo posteriormente sobre el grado de incumplimiento de este SLA que se ha producido. Por otro lado tampoco al ISP le supondría consuelo alguno cualquier posterior compensación económica que el operador hiciera si ésta no va a cubrir

nunca el valor intangible de haber estado horas y horas fuera de Internet. Pero no es menos cierto que fuera del SLA (que es parte del contrato) no existe forma alguna de exigir al operador una calidad mínima, y en los últimos años los SLA se han hecho muy comunes en los contratos celebrados por los operadores.

Otro parámetro exigible en el SLA es la disponibilidad de la conexión: en este sentido hay que ser tajantes, tanto en las conexiones de menor coste y fiabilidad como en las otras, y obtener del operador que las facilite una disponibilidad del 99.9% en todos los casos, lo que querría decir que en plazo acordado (por mes o trimestre, varía según el operador) debe existir servicio activo al menos el 99.9% de ese periodo de tiempo, lo que supondrá que a partir de unas horas de corte se incumpliera esta disponibilidad y se puedan exigir responsabilidades al operador. La disponibilidad no implica ancho de banda o tiempo de respuesta alguno (otro parámetro ya más difícil de exigir en el SLA), pero al menos garantiza que haya un tiempo máximo de resolución de averías.



La única forma que tendremos de garantizar que incluso en caso de incumplimiento del operador de su propio SLA no vamos a sufrir de manera excesiva va a ser dotando a nuestro ISP de al menos dos conexiones con operadores distintos, para evitar que caigan el 100% de los servicios. Además, nuestro objetivo es tener un servidor de backup conectado a través de otro operador y alojado en distinta localización, que sirva para reducir la falta de presencia en Internet durante las caídas. Sería un error grave confiar todo en manos de un único operador.

El ISP por tanto contratará al menos estos dos accesos:

- Una conexión con ancho de banda garantizado (PIR y CIR elevados) para DNS, backup y *relay* de correo. Esta conexión llevará además aparejado una única dirección pública en Internet, conformándonos con un CIR del 50% de la conectividad, que será de 2 Mbps (el ancho de banda de subida y el de bajada además serán coincidentes), suficientes para dar un servicio inicial a un número de entre 200 y 300 clientes de sitios Web de tamaño medio y de 50 a 100 sitios Web de tamaño grande, tal y como acabamos de calibrar el tráfico que esos sitios nos supondrán. Esta conectividad podrá ser perfectamente entregada por el operador directamente con señal Ethernet, dado que sólo solicitamos del operador una IP pública, pero dado que esperamos en un futuro disponer de más de un servidor en nuestras instalaciones se va a prever con antelación el que se contraten en el futuro más IPs sobre esta conexión y por tanto se necesitará un router. Por las características de garantía solicitadas el mejor tipo de conectividad hoy día disponible sería *frame-relay*.

- Luego contrataremos una conexión con ausencia de compromiso en la disponibilidad de caudal. Esta conexión dispondrá de una única IP tanto ahora como en el futuro, pero dado que esperamos usarla para dar servicio a varios equipos necesitaremos un router con capacidad NAT. Esta conexión sin garantía a cambio ofrece mejores anchos de banda, como las ofertas ahora disponibles para accesos a Internet mediante tecnología LMDS de varios Mbits por segundo simétricos, que será la opción escogida en detrimento del ADSL o módem de cable.

6.1.2 Red local (equipos de trabajo)

Los equipos de trabajo y la actividad que estos generen no deberían interferir sobre el ancho de banda disponible en Internet por parte del ISP. Para que esto sea posible se va a diseñar una red local cuyo acceso a Internet no perjudique a los servicios del ISP, bien dedicándole una conexión independiente o bien acotando su parte del ancho de banda total disponible. El objeto de esta separación es evitar que en el día a día de la empresa haya usuarios que por desconocimiento o negligencia puedan abusar de tal manera de la conexión a Internet disponible que haya pérdida de calidad de servicio en las Web alojadas (que usarían la misma conexión a Internet).

Como diferenciar con diferentes conexión y equipamiento a estos usuarios de la conectividad propia de un ISP supondrá un coste elevado, lo que se hará será dotar a esta red de usuarios de acceso a Internet mediante la propia red de servidores, usando nuestro por ahora servidor único para que dedique una parte acotada de la conexión a esta red interna. La conexión sin garantías de caudal será la que utilizaremos para dotar de acceso a Internet a esta red local, siendo el router de esa conexión quién se encargue del NAT necesario.

Cada equipo de trabajo dispondrá de una configuración asignada de manera dinámica mediante DHCP. Se impedirá el acceso a Internet desde aquellos equipos que no utilicen las IPs asignadas mediante DHCP, notificándose además cualquier asignación de IPs al administrador, para detectar cualquier uso indebido de la red local. Las IPs serán privadas y del rango de direcciones 172.16.0.0/24, siendo la primera de ellas (172.16.0.1) la que asignaremos al router que realizará el NAT.

La siguiente IP disponible, 172.16.0.2, estará asignada a nuestro servidor, que estará además entre nuestra red y el router que le permite acceder a Internet. Este salto extra que supondrá el que nuestro servidor conmute el tráfico entre el router y la red local se hará mediante *proxy-arp*, para que sea totalmente transparente.

Del total de direcciones privadas posibles en el rango 172.16.0.0/12 sólo se usará la primera subred de 254 hosts, que será la que hemos especificado (127.16.0.0/24), pudiéndose soportar el crecimiento futuro

del ISP (no se esperan tantos equipos cliente en la red de trabajo, y aunque se llegaran a superar los 254 de la red inicialmente prevista, se pueden usar cualquiera de las otras subredes posibles).

Los equipos de la red local usarán al servidor principal como servidor DNS, siendo este al mismo tiempo el encargado de proveer las IPs dinámicas a la red local mediante el protocolo DHCP, y de controlar el ancho de banda que consumen estos equipos, además de realizar el *firewall* de esta red local.

Aunque el pase del tráfico de la red local por el servidor principal es innecesario para funcionar, permite el control sobre el ancho de banda (estamos ante un router software, ya que va a ser el servidor Linux el responsable del enrutamiento), y nos permitirá además no limitar futuras adaptaciones (el hardware específico como un router hubiera estado más limitado para añadírseles nuevas características que por software bastarán por lo general se obtendrán actualizando los programas).

Del conjunto de conexiones a Internet disponibles en el ISP, el servidor dispondrá de todas ellas, por lo que al ser el servidor también quien dota de acceso a Internet a esta red privada, será fácil redirigir ésta tráfico hacia otra conexión futura sin readaptar siquiera el espacio de direccionamiento.

6.1.3 Red de servidores

El área de captación dedicada a los servidores estará por delante del área de equipos de trabajo que acabamos de comentar, aunque ambas serán redes idénticas de medio compartido, Fast Ethernet cableadas con latiguillos UTP Cat5e para 100BaseTX (aunque la calidad del cableado permitiría un posterior crecimiento hoy por hoy innecesario hacia redes Gigabit). Se busca una segmentación elevada, de ahí que se usen en todos los casos switches.

A esta red de servidores corresponderán espacios de direccionamiento dispares, ya que los servicios que vamos a ofrecer lo pueden ser mediante diferentes operadores, existiendo para cada uno de ellos un router, switch o hardware análogo que entregue en la red local que habremos diseñado el tráfico de Internet. De momento sólo se han previsto dos conectividades distintas, una de cable y otra con una mayor garantía de ancho de banda.

El hardware de cada operador puede ser sustituido por equipamiento del propio ISP, pero no va a ser necesario: el número actual de servidores previstos será únicamente uno, y la seguridad será diseñada internamente en el propio servidor, por lo que no poseer el control de los routers no supondrá una mayor desprotección o falta de poder de limitación en el propio router.

Otra razón que podría habernos decantado por un router propio frente a utilizar los colocados por la operadora hubiera sido controlar mejor el tráfico y poder obtener estadísticas del mismo con independencia de las que nos pueda suministrar el operador. Una vez más, como sólo tenemos un servidor estas estadísticas se obtendrán dentro del mismo.

Supone esta decisión una apuesta arriesgada: si en el futuro el ISP alojara un número más elevado de servidores, o se decidiera por el *housing* de máquinas de otros clientes, el control individual y la seguridad ofrecida por un *firewall* harían necesaria la instalación de un servidor dedicado a enrutamiento y control del tráfico, ya que será totalmente desaconsejable dejar en manos del que configure cada equipo la correcta implantación de la seguridad y del control del ancho de banda, o esperar que todos los operadores lo hagan sobre sus routers, por eso nuestro servidor será el futuro router.

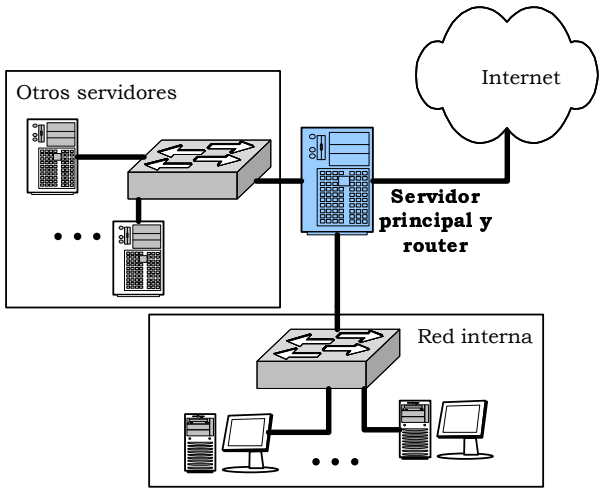


Ilustración 6-8

Usar Linux y un equipo para realizar por software la conmutación de paquetes asignada a un router dedicado (tanto para dar acceso a Internet a nuestra red de trabajo privada, como a los posibles futuros otros servidores), no supone un riesgo por cuanto a las capacidades del servidor: el ancho de banda que se habría de transferir entre las diferentes interfaces de red del servidor no va a ser lo suficiente elevado para comprometer sus recursos, y hemos de tener también en cuenta que la frecuencia de trabajo de los procesadores actuales es muy superior al ancho de banda (10 Mbps no es demasiado para un Pentium IV, como además viene a demostrar que cualquier router Cisco contiene en la actualidad microprocesadores de la gama 680XX de Motorola, que trabajan a frecuencias mucho más bajas, y no disponen tampoco ni de la décima parte de los 1024 MB de memoria RAM con los que dotaremos a nuestro servidor).

6.1.4 El servidor de backup

Hasta el momento hemos hablado del diseño de la red del ISP olvidándonos por completo del otro servidor que hemos previsto alojar en un operador externo mediante la fórmula de *housing*.

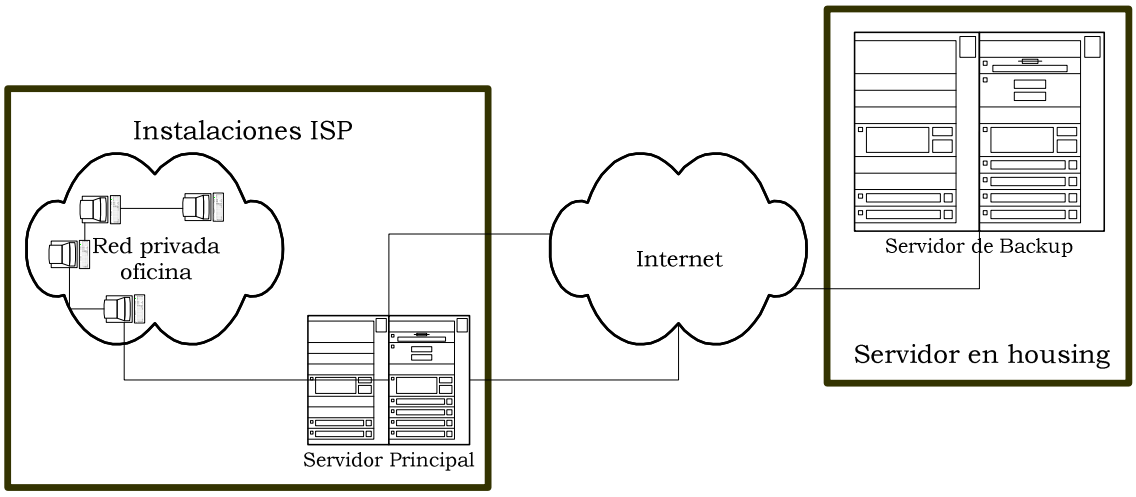


Ilustración 6-9: Posición del servidor de backup en Internet

Respecto dónde estará alojado este servidor, será exigible para determinar el proveedor:

- El servidor dispondrá de una IP pública.
- El caudal de acceso a Internet de que dispondrá el servidor no deberá estar garantizado: exigirlo supondría un sobrecoste difícil de asumir dado el bajo uso que se va a dar a este servidor. Las fórmulas de alojamiento en las que se tarifa en función del consumo mensual total serán mucho más interesantes.
- La IP asignada no estará filtrada o detrás de un *firewall* que pueda recortar nuestra capacidad de decisión respecto a qué servicios montamos en dicho servidor.

- El acceso a Internet no será mediante alguna de las dos compañías con las que contrataremos el acceso a Internet para el servidor principal del ISP, para garantizarse que en caso de que un fallo del operador no nos deje simultáneamente a ambos servidores fuera de servicio.

Como las tres primeras condiciones se cumplen en cualquier IPP como Arsys (www.arsys.es), Acens (www.acens.es) y similares, la elección sólo deberá estar en sintonía con la última exigencia de que el acceso se localice en un proveedor que no emplee a los operadores que vayamos a contratar.

Dado que el servidor principal y el de backup van a estar localizados en diferentes puntos, habría que analizar una forma de garantizar que el acceso al mismo es facilitado mediante algún mecanismo.

En concreto estamos hablando de utilizar tunelado, de forma que la IP privada 172.16.0.3 esté asignada a este servidor de backup, tunelado que en un lado sería responsabilidad del servidor principal y en el otro del propio servidor de backup, permitiéndonos a los administradores realizar tareas administrativas sobre este servidor sin conocer ni tan siquiera la IP pública del mismo, lo que además facilitaría una posterior migración a otro operador del servidor de backup.

6.1.5 Diseño de la red física con VLANs

A nivel lógico hemos hasta el momento hablado de tres redes distintas: una red de servidores, otra red de trabajo y una tercera red para la conexión de los diferentes routers, redes que necesitarían cada una de ellas un switch para su conexión, aunque de momento podríamos prescindir del switch para la red de servidores, dado que no tenemos previsto un uso inmediato del mismo.

Pero existe una alternativa más interesante y modular, consistente en usar switches que dispongan de capacidad para generar VLANs, es decir, con un único switch físico poder crear tres dominios de difusión distintos, con una serie de ventajas inmediatas:

- Con un solo dispositivo hardware necesario reduciremos el coste global del hardware de la red.
- Es una solución escalable, como luego veremos: siempre tendremos una cantidad de puertos de conexión en los switches similar a la de los hosts, sin apenas desaprovechamiento de los mismos.
- Es compatible con el *proxy-arp* que vamos a utilizar en el servidor principal.
- Agrega seguridad, al permitir crear más subredes sin necesidad de adquirir más hardware y de una manera rápida.

Antes de continuar detallaremos en qué consiste usar VLAN implementadas mediante un switch que da este soporte: con todos los

hosts conectados al mismo switch, éste es capaz de usando el protocolo 802.1q aislar los hosts del mismo en dos redes virtuales distintas.

Lógicamente el switch que ofrece este soporte tiene un coste mayor, pero con un único switch de 24 puertos podremos comenzar nuestro ISP, mientras que de la otra forma hubieran sido al menos tres y de igual tamaño, ya que por debajo de 24 puertos hubieran quedado obsoletos en poco tiempo.

Hemos comentado además que es una solución escalable: el modelo de switch que usaremos, la gama Catalyst 2900, permite usar sus dispositivos en un *stack*, al cual se irían incorporando nuevos switches conforme crecieran las necesidades.

Físicamente un *stack* es una pila de switches, los cuales por diseño están preparados para, mediante un puerto Gigabit interconectarse para conmutar el tráfico que a sus puertos Fast Ethernet llegue.

De esta forma y con sólo un switch de 24 puertos al principio, crearemos las 2 VLAN de la red de acceso a Internet y la red local mientras haya pocos equipos de trabajo en la misma. Si el total de puertos resultara insuficiente o hubiera que crear también la red de servidores porque se va a necesitar otro servidor, bastaría adquirir otro switch con las mismas características e interconectarlos, o bien usar estos dos switches por separado.

Un switch de 24 puertos cubrirá al menos dos años de vida del ISP, por no estar previsto que haya un número elevado de servidores a corto plazo, ni más de 4 o 5 usuarios estables en la red local de trabajo. Mientras dure la fase de diseño y haya necesidad de más estaciones de trabajo se podrá usar un simple hub conectado a uno de los puertos del switch para poder trabajar, aunque con una evidente pérdida de prestaciones.

Por esta modularidad y capacidad de crecimiento el modelo escogido será el Cisco 2912XL, que trabaja en el modo de conmutación *cut&throug*. Cisco será también la marca que usaremos para los routers de acceso a Internet.

Al existir dos conexiones a Internet, en cada una de ellas nos hará falta un router que bien puede ser suministrador por el operador, bien puede ser propiedad nuestra, pero en cualquiera de los casos mostraremos su configuración como si nuestros fueran, escogiéndose para ello el modelo 2611 de Cisco en ambos casos, por disponer de conexión auxiliar y ranuras de expansión que nos permitirán adaptarlo a los diferentes tipos de señales de nuestros operadores (LMDS suministrará directamente Ethernet, pero en el caso de la conexión *frame-relay* nos hará falta un router que disponga de esta capacidad, lo que cumpliría este modelo con la correspondiente ampliación).

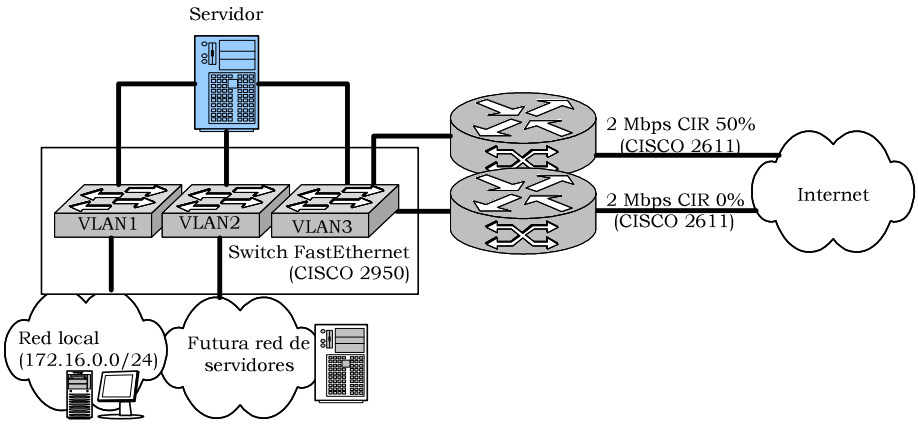


Ilustración 6-10: Esquema de la red diseñada con VLAN

El escoger una única marca para todo el hardware de red (routers y switches) obedece a sinergias de funcionamiento: una única marca supone un único modelo de configuración y administración de los diferentes elementos (en concreto, el sistema operativo IOS que esta marca utiliza), un único modelo de evaluación y monitorización (el protocolo de descubrimiento de Cisco ayuda casi tanto como el uso de SNMP a la hora de monitorizar el estado de una red). Lo único que cabría discutir es si el haber escogido esta marca concreta ha sido acertado por costes, pero no así por rendimiento y funcionalidad, donde Cisco por tradición es líder (el 80% del tráfico de Internet sigue estando en la actualidad transferido a través de dispositivos de esta marca).

6.2 Diseño de los sistemas y servicios ofrecidos

6.2.1 Diseño de los servidores

Hardware

Existirán dos servidores: uno alojado en las instalaciones del ISP, y otro alojado mediante una fórmula de *housing* o *collocating* de manera externa. En el primer caso se adquirirá una serie de equipamientos de red que acabamos de ver en el apartado anterior, además de los equipos cliente (de cuyas características no comentaremos nada, por no ser relevante). Por el contrario, en el equipo de backup no será necesario adquirir hardware alguno aparte de la propia máquina.

El servidor principal y el secundario tendrán como principales característica hardware el disponer de RAID 0, que permita redundancia en el almacenamiento de información en disco (al menos dos discos contendrán la misma información grabada en paralelo: en caso de fallo de uno de los dispositivos el servicio puede mantenerse en uso al mantener el disco en funcionamiento la totalidad de la información). El RAID puede ser montado en la actualidad tanto sobre dispositivos SCSI como sobre dispositivos IDE, al existir controladoras RAID con esta capacidad. Dada la diferencia de coste, se optará por la solución IDE basada completamente en hardware, con probablemente peor rendimiento en los accesos, pero a un coste mucho menor.

Hablamos de solución hardware debido a que se instalará una tarjeta PCI para evitar tener que simular mediante software el RAID. Estas tarjetas tienen soporte para el modo Ultra-DMA 66 más reciente, como ocurre con los modelos del fabricante BossLan (**www.bosslan.com**), con las siguientes características:

La tarjeta RAID incorpora un conector IDE de entrada, que se conecta a la placa, y dos conectores IDE de salida, dirigidos a conectar dos discos. Con un DIP switch se puede elegir entre RAID 0 y RAID 1 (como buscamos redundancia en los datos será RAID 1 lo que nosotros buscaremos), que se realiza de manera transparente al sistema operativo instalado. Debido a que montar un RAID SCSI supone un coste económico mayor y casi idéntico en configuración, elegir este tipo de tarjetas IDE están ya sobradamente justificado.

Los discos tendrán una capacidad de 120 GB en ambos casos, en un sólo disco (en realidad al usar RAID siempre tendremos el doble de discos, pero la vista lógica será una única unidad de 120 GB por servidor, existiendo físicamente dos unidades con idénticas características).

Del resto de características físicas de los servidores, no hay interés alguno por disponer de un hardware especialmente avanzado,

básicamente porque tal y como hemos analizado el tipo de servicio que van a ofrecer no requieren una especial potencia. Básicamente, cualquier configuración basada en Pentium IV o AMD Athlon que disponga de al menos 1 GB de memoria RAM, suficientes para garantizar tanto la conmutación de paquetes de nuestra red local que tendrá el servidor principal, como su función de servidor Web, de correo y de DNS (que se dará tanto en el servidor principal como en el servidor de backup.

En ambos servidores harán falta tarjetas de red *FastEthernet*: tres en el caso del servidor principal (una interfaz hacia la red local, otra hacia la red de futuros servidores, y otra hacia Internet) y una única tarjeta en el servidor de backup. Para asegurar una integración óptima exigiremos sean también de marca Cisco.

Sistema operativo

A nivel de software, el sistema operativo será tanto para el servidor principal como para el de backup, Linux. Las razones de esta elección son simplemente el buscar con ello reducir costes de adquisición de software, y también el hecho de que es en la actualidad el sistema operativo más adecuado para un entorno de redes en las que se desea utilizar un servidor para tantas y tan diferentes cosas como se esperan del servidor principal.

Linux permite una conmutación de paquetes con un control tal que no es capaz en la actualidad ningún router de hardware específico: permite no sólo transferir entre las interfaces los paquetes, sino que además permite filtrar el tráfico, monitorizarlo o limitar su ancho de banda mediante software de calidad de servicio (*QoS*, que viene del inglés *Quality of Service*). Existen multitud de pequeñas distribuciones de Linux adaptadas para estos propósitos y capaces de ser incluidas dentro de pequeños ordenadores destinados a hacer de router (**http://www.linuxrouter.org**).

Por otro lado, es también Linux el sistema operativo en el que se encuentran MTAs, servidores Web y servidores DNS de código abierto, y totalmente robustos y testeados como *Postfix*, *Apache* y *Bind*, que serán los servidores que utilizaremos.

Dentro de las diferentes distribuciones de Linux existentes, escogeremos *Debian*. El discutir el acierto o fallo de esta distribución con respecto a otras supondría entrar en una enumeración de ventajas e inconvenientes que nos desviarían del objeto de esta memoria técnica, que es lograr un servicio adecuado. El servicio será posible de igual manera con esta distribución que con otras, además de que todos los programas que vamos a utilizar estarán igualmente disponibles en otras distribuciones distintas. *Debian* es el escogido únicamente por ser de entre los más implantados, el menos dado a primar aspectos como la parte gráfica o la interfaz en detrimento del esmerado cuidado que se tiene con la seguridad de los programas que incorpora en su versión

estable, de ahí que se confíe más en esta seguridad que en disponer siempre de las últimas versiones de los diferentes programas.

Otra característica necesaria en el servidor principal será disponer la capacidad de usar *proxy-arp*. Aunque inicialmente no habilitado, la razón de ser de este servicio será la futura existencia de más servidores que deban conectarse a través de nuestro actual servidor-router (ya que su función actual será realmente una conjunción de ambas, servidor Web, correo y DNS, y al mismo tiempo router de la red privada y en el futuro de otros servidores).

Debido a que sería posible que en el futuro no controláramos alguno de los routers instalados por los ISP, y nos fuera por tanto imposible reconfigurar las tablas de enrutamiento de éstos, se daría el caso de que con *proxy-arp* podríamos "engañar" a estos routers, haciéndoles creer que en la zona de captación se encuentra un host que en la realidad el router solo vería porque nuestro servidor principal responde con su dirección MAC, estando realmente situado este servidor en un área de captación diferente.

A nivel lógico, nuestro servidor principal dispondrá de tres interfaces de red, que en la terminología usada para nombrar estas interfaces en Linux, serían:

- **eth0**, conectada al POP, y en la que situaremos todo el conjunto de IPs públicas de que dispongamos. En concreto vamos a suponer que disponemos de dos IPs, ya que en este diseño hemos establecido que sólo habrá dos conexiones distintas inicialmente, cada una con su IP: 128.100.0.2/32 sería primera, y 128.200.0.2/32. En ambos casos también se encuentra delegada la resolución inversa de esa IP, y uno de los dos operadores nos garantiza que el rango 128.100.0.0/28 se encuentra reservado para nuestro futuro crecimiento, con lo que las IPs comprendidas entre la 128.100.0.3 y la 128.100.0.7 se podrían asignar a otros futuros servidores.

- **eth1**, conectada a la red local de las estaciones de trabajo. Sobre esta interfaz se permitirá la resolución DNS recursiva (actuaremos de servidor DNS de nuestra red), se hará al servidor responsable de la asignación dinámicas de las IPs mediante DHCP, y se realizará enrutamiento y control del tráfico (mediante *QoS* para garantizar que el consumo de los hosts situados a ese lado no supera un determinado margen).

  El servidor no hará NAT de esta red (la 172.16.0.0/24), ya que esta función estará configurada en el propio router que dota esta conexión (aquel con IP 128.200.0.2) y que se encuentra en la red alcanzada por eth0, luego también en eth0 será necesario configurar la IP 172.16.0.2 que tendremos en eth1. Mediante *proxy-arp* se logrará que la puerta de enlace de los hosts de esta red (el router 172.16.0.1) sea alcanzable.

- **eth2**, conectada en el futuro a una hipotética red de servidores. Por ahora ni tan siquiera habremos adquirido el equipamiento de red necesario para ello (el switch necesario a este lado), pero ya dispondremos la tarjeta de red necesaria. El tráfico hacia esta red será el generado hacia las IPs 128.100.0.0/28, y la misma IP 128.200.0.2 tendremos que configurarla aquí.

El uso de las características que hemos ido incorporando al router van a incrementar de manera relevante el tiempo que el servidor habrá de dedicar a tareas de enrutamiento y control de calidad del servicio. Estas tareas (insistiremos una vez más) no van a suponer una ralentización en el servidor, aunque hay que admitir que supondrán incrementar el número total de saltos que el tráfico hacia y desde Internet realice cuando en el futuro tengamos una red de servidores, ya que la red de servidores habrá sido configurada de manera que pase siempre por este servidor.

Este salto innecesario nos hace falta para garantizar el control centralizado del tráfico: el ISP necesita *firewall*, monitorización y calidad de servicio, y el abandonar estas características a la configuración de cada uno de los servidores es un error, ya que en nuestra red lo que un servidor pueda tener mal configurado afectaría a todos los restantes.

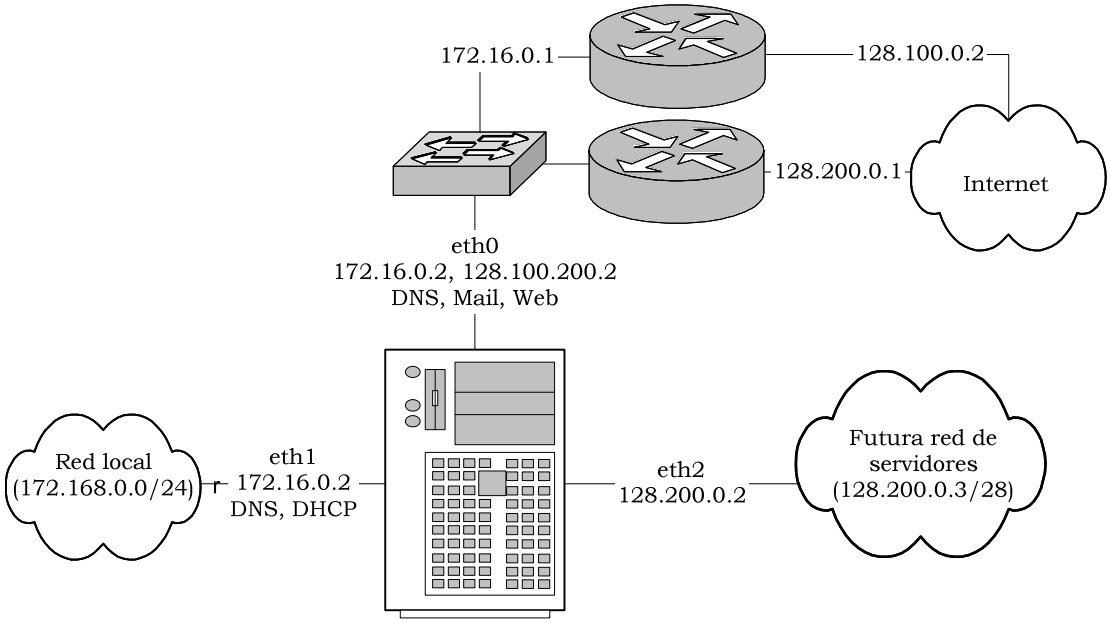


Ilustración 6-11: Interfaces en el servidor principal

En realidad si en un futuro se detectara un uso elevado de recursos por existir demasiados hosts en la red de servidores para que estos sean manejados por un router Linux, la solución pasaría por migrar esta solución a una configuración en la que el servidor que actúa de router no tuviera más servicios, o incluso una solución hardware dedicada, pero es importante que siempre se asegure el ISP de que todo su conectividad pasa por un único punto en el que pueda controlar en un determinado instante qué está sucediendo.

A nivel lógico, el diseño propuesto y los servicios que en cada interfaz se esperan sean necesarios serían los mostrados en la Ilustración 6-11.

Seguridad

La seguridad en Internet será crucial: aunque hay que establecer un seguimiento continuo de los servidores, no es lógico que tengamos que pasarnos las 24 horas delante del mismo monitorizando las conexiones que se produzcan, por lo que ya en el diseño de la configuración de los servidores minimizaremos los futuros problemas cerrando todo aquello que no será imprescindible.

Veremos ahora qué medidas tomaremos en cada caso y ante cada uno de los problemas de seguridad que nos puedan surgir.

Ataques de denegación de servicio

Nuestros servidores al estar en Internet estarán expuestos a ataques de denegación de servicio, encaminados a impedir el acceso a los usuarios legítimos del servicio, bien dañando la máquina o su software que ofrece los servicios, bien colapsándolo, deteniéndolo o cualquier método alternativo. Se trata de un ataque contra los recursos disponibles: ancho de banda, memoria, espacio en disco, tiempo de CPU o incluso energía si el ataque se produce durante un corte en el suministro eléctrico.

No contemplaremos de manera específica otros ataques igualmente graves pero que difícilmente se podrían dar, como serían la alteración de datos o la destrucción física del equipamiento (estos ataques los analizaremos en el apartado dedicado a los ataques genéricos).

En este tipo de ataques incluiremos también aquellos usos ilegítimos del servicio por parte de nuestros propios usuarios, como la existencia de ficheros con acceso reiterado y la descarga abusiva de los mismos por parte de terceros. En todos estos casos, el impacto es tremendo si no se han tomado las medidas pertinentes.

Ataque *SYN flood*

Este ataque contra nuestros recursos de memoria y CPU del servidor se aprovecha del tratamiento de las conexiones que se realiza en TCP/IP, y como ya comentamos anteriormente en los antecedentes, una buena solución contra ellos pasaría por usar *SYN cookies*, números de secuencia que puedan posteriormente ser verificados (es decir, validados como generados por el servidor) sin necesidad de asignar recursos hasta que se produce la transferencia de información, no durante la negociación.

Activar esta característica es posible en cualquier sistema Linux, desde que *Eric Schenk* lo implementara en febrero de 1997, al calor del comentado ataque sufrido por *Panix*.

Ataques contra el espacio en disco

Cualquier servicio que escriba datos en disco es susceptible de servir para este tipo de ataques. Es particularmente peligroso aquello que genera cualquier tipo de *log* en el sistema, ya que por tratarse de texto

su tamaño resulta mayor. La solución en este caso parece evidente: disponer de una gran cantidad de espacio de almacenamiento en disco, pero esto no es de por sí suficiente, porque nunca va ser infinito y en un determinado momento nuestro sistema puede quedarse igualmente sin espacio. De hecho el sistema operativo ya toma algunas medidas al respecto: en Linux el demonio responsable de los *logs*, el demonio *syslogd*, ya se encarga de retardar y analizar las líneas a insertar en los ficheros de *log* para detectar las repeticiones consecutivas en el mismo y únicamente informar de cuántas veces se repite el último mensaje (*"last message repeated X times"*), en lugar de escribir esos caracteres de nuevo.

Por nuestra parte, en primer lugar los programas en ejecución habrán de evitar usar de manera alegre el espacio en disco: grabar los datos una vez confirmadas las transacciones o compras, y no informar de hechos redundantes, pero esto no basta. Lo interesante sería analizar formas de evitar que un cierto *log* o proceso en ejecución sature el espacio en disco: resultará para ello aconsejable que todos los servicios estén asignados a usuarios con cuota en disco asignada, lo que implica ya de entrada descartar el que haya servicios que usen al *root* como usuario para ejecutarse.

Aunque resulte tentador, hay que evitar como alternativa el que cada directorio sensible de ofrecer problemas (*/log*, */var*...) sea montado en una partición o disco diferente: el control que ofrece esta característica es el mismo que el uso de cuotas, pero nos puede llevar a situaciones desagradables si en un determinado momento necesitamos ampliar el tamaño de estas particiones.

Redirección ICMP

Existe entre los diferentes tipos de mensajes ICMP uno que permite indicar al host que lo recibe que existe una ruta más corta para determinado destino. Dado que no actuaremos de *carriers* de Internet, que podrían si necesitar cambiar sus tablas de enrutamiento de manera continúa (y aún en ese caso existen protocolos de enrutamiento lo suficientemente potentes como para no necesitar este tipo de mensajes), y como estos paquetes podrían ser utilizados para obligarnos a enviar el tráfico a través de otras localizaciones, filtraremos en servidor este tráfico.

Evitar el *spoofing*

Siempre se ha de evitar en la medida de lo posible el *spoofing*. Aunque ya se ha comentado que es un problema más sencillo de tratar en los routers intermedios o incluso en los routers del ISP a través del cuál se realiza el ataque (por cuanto allí aún es probable que la IP pueda ser verificada a partir de la interfaz por la que ha llegado, mientras que en nuestro host de destino lo normal es que todo el tráfico nos llegue a través de una interfaz).

Lo que sí podemos es evitar el *spoofing* de direcciones privadas, de aquellas que entren desde Internet y usen nuestras propias IPs (es decir, si al disponer de dos interfaces eth0 y eth1, nos llega por eth0 tráfico cuya IP de origen pertenece a hosts que únicamente están en eth1:

También se descartarán los paquetes que parezcan venir o ir a *loopback* desde una interfaz que no sea la interfaz de *loopback*. Asimismo, se descartarán los paquetes de difusión mal formados y los paquetes de difusión múltiple de clase *D* que tengan dicha dirección como origen, así como aquellos con direcciones IP reservadas de clase E y las direcciones definidas como reservadas por el IANA. El listado de IPs a denegar será:

- 127.0.0.0/8 cuando no venga de la propia interfaz de *loopback*.
- 10.0.0.0/8, 172.127.0.0.0/8, 192.168.0.0/16, por ser espacios de direccionamiento privados. Del rango 172.16.0.0 únicamente permitiremos el tráfico proveniente de 172.16.0.0/24 y el que afecte a 172.16.254.0/24, siendo denegado el restante destinado a 172.16.0.0/12.
- Denegaremos también aquel tráfico de difusión inesperado, como el destinado a 255.255.255.255 o a 0.0.0.0.
- Las direcciones de difusión de clase D: 224.0.0.0/4, y las reservadas de clase E: 240.0.0.0/5.
- Existen además luego direcciones definidas como reservadas por el IANA dentro de muchos y dispares rangos, como serían las que casaran con X.0.0.0/8, donde X sería igual a 1, 2, 5, 7, 23, 27, 31, 37, 39, 41, 42, 58, 60, 80, 95, y de la 112 a la126.

Evitar ping de la muerte y ataque *smurf*

El ping de la muerte se produce cuando nuestro servidor recibe grandes cantidades de tráfico ICMP y no puede dedicar sus recursos más que a responder estas peticiones de *echo*. Hay que tener en cuenta además que las tramas ICMP pueden tener un tamaño considerable, siendo factible un relleno de 64 *Kb*. en la trama que acabe por saturar nuestra conexión.

El otro tipo de ataque, similar al anterior, es el producido por el envío de tráfico ICMP en el que la dirección destino es la dirección *broadcast* de nuestra red, logrando así que cualquiera de los hosts de esa red le respondan teóricamente. Y remarco lo de teóricamente porque hoy día ya no es habitual esta situación, pero aunque así fuera, lo que haremos será evitar directamente cualquier solicitud de *ping*.

De esta manera perdemos herramientas que igualmente nos podrían ser útiles en un futuro para detectar problemas de configuración o funcionamiento de los servicios, pero será preferible tener que parar en

esas circunstancias el *firewall* que tener continuamente la red expuesta a *pings* malintencionados. Las únicas IPs que autorizaremos serán aquellas de la red local (172.16.0.0/24).

Las puertas traseras son algo también habitual. Dado que el servidor sólo podrá tener servicios en ciertos puertos conocidos, podríamos activar el seguimiento de las conexiones que se ha incorporado como módulo en las distribuciones más recientes a *iptables*. Esta característica permite que el servidor sólo acepte el tráfico que inicia una conexión sobre un puerto privilegiado conocido y esperado (los demás estarán también restringidos), o bien aquel que está relacionado con una conexión previamente establecida.

6.2.2 Servicio Web

Usuarios

El servicio Web estará constituido por los espacios Web al que los usuarios podrán acceder mediante protocolo FTP para colocar los ficheros. Como es de esperar que la mayoría de usuarios tengan algún contenido dinámico, se creará por defecto una base de datos para cada usuario con espacio Web.

Los usuarios tendrán cuentas físicas en el sistema, cuentas protegidas mediante contraseñas que el usuario no podrá escoger, para evitar que sean usadas contraseñas fácilmente predecibles. La contraseña será modificable a través de la página Web desde el propio panel de control del cliente, panel al que llegará mediante otro usuario y contraseña válidos únicamente para administrar todo el conjunto de servicios que el cliente haya contratado.

La cuenta física de cada espacio Web permitirá que hacia ella apunten diferentes dominios, todo ello configurable desde el panel de control vía Web. Lo que no se permitirá es que dos dominios compartan el mismo espacio Web, ficheros y directorios.

De hecho dentro del directorio de usuario de cada espacio Web existirá por defecto un único directorio *www* que contendrá los ficheros visibles a través de **www.sudominio.com** y de cualquier otro dominio que esté asociado a esta cuenta física. Lo que si que se permitirá es que desde el panel de control se creen nuevas subzonas dentro de su dominio que irían a parar a otro directorio distinto (por ejemplo, **test.sudominio.com** implicaría que el panel de control vía Web creara una nueva entrada DNS si es pertinente, otra entrada en el servidor Web, y un directorio *test* bajo su directorio de usuario en el que irían los ficheros para esta entrada).

En un directorio, el servidor Web cogerá por defecto el fichero *index.php* en caso de que este se encuentre. Si este fichero no existiera buscaría en segundo lugar un fichero *index.html* y en tercer lugar *index.htm*. Si ninguno de estos tres ficheros está disponible, el servidor devolverá un mensaje de error. No se permitirá el listado de directorios

que puedan dejar al descubierto la estructura de la página Web en ningún caso, por lo que los usuarios tendrán que acceder por FTP si desean listar los contenidos de una Web. Todos los directorios de una Web estarán dentro de su carpeta de usuario y serán legibles para el usuario y el grupo, así como ejecutables para todos en general.

Al igual que también implantaremos herramientas de calidad de servicio, será interesante que en el caso concreto de las Web usemos uno de los módulos disponibles en Apache para garantizar que los diferentes sitios Web no superan una determinada tasa de transferencia total: *mod_throttle*. Esta opción permite asignar tanto una transferencia total mensual en *bytes* transferidos, como un ancho de banda máximo para que en ningún momento un sitio Web consuma la totalidad de nuestro acceso a Internet. Para esta segunda característica es para lo que usaremos este módulo, ya que por QoS únicamente esperábamos controlar que el ancho de banda dedicado a Web estuviera garantizado que no supusiera ralentizar otros servicios, pero ahora podemos incluso fijar distinciones entre unos clientes y otros (en función del tipo de sitio Web y su facturación, por ejemplo). Cada cliente podrá contratar un caudal garantizado mínimo que su sitio Web espera que consuma. El comportamiento de este módulo puede ser peligroso para la imagen de la empresa, ya que alcanzados los límites fijados, el módulo deja de permitir el acceso a ese sitio con un mensaje de servidor ocupado.

Otra herramienta adicional será la reescritura de URLs, disponibles a través del módulo *rewrite_module*. Estas herramientas serán usadas en conjunción con las estadísticas que se realizarán de cada sitio Web para que ningún cliente pueda acceder a su página Web usando el nombre canónico de nuestro servidor y su nombre de usuario en lugar del *VirtualHost* que para él hemos configurado, por ejemplo:

http://www.midominio.com está asignado al usuario físico de la máquina *web_midominio*, sitio Web que si no lo evitamos podría llegar a ser accedido mediante otra dirección distinta en la que probablemente no se computaría ese consumo de ancho de banda, **http://nombre.del.servidor.es/~web_midominio**. Con el mismo método (reescritura de peticiones URL) evitaremos que los usuarios puedan usar el lenguaje PHP en sus sitios Web si su producto no lo permite. En ambos casos la página a la que estas peticiones deben conducir es a la página principal corporativa del ISP, que es a dónde además acabarán por ir ante cualquier mensaje de error (página no encontrada, por ejemplo).

Herramientas y lenguajes disponibles en el servidor

El servidor Web tiene como tarea fundamental suministrar contenidos en formato HTML, y para dar más versatilidad al mismo estará disponible y sin limitaciones de ningún tipo el uso del lenguaje PHP tanto dentro de los ficheros con extensión *html* y *htm* como en los propios ficheros *php*.

Al crearse el espacio Web no se colocará fichero alguno en la Web excepto un fichero *index.php* que muestre una página por defecto diseñada para que sea visible el nombre del ISP donde está alojada y un mensaje al estilo de "página en construcción" o "próximamente". Tras un tiempo preestablecido en esa página en construcción (10 segundos) saltará a la Web corporativa del ISP.

Dado que PHP viene con librerías suficientes para hacer cualquier tipo de cosa, se instalarán las librerías de PHP más comunes:

- *PHPlib* (mejoras al lenguaje estándar).
- *LibGD* (tratamiento de imágenes).
- Interfaces hacia *MySQL* y *PostgreSQL* (interfaz con estos sistemas de bases de datos).

Los usuarios no verán limitadas sus capacidades para colocar en el Web ficheros de tipo PHP, aunque estarán prohibidos cualquier otro tipo de ejecuciones en el servidor por parte del cliente, quedando por tanto fuera las CGI basadas en *shell*, *Perl* o directamente en ficheros ejecutables. Si alguna de estas debiera ser utilizada, lo habría de ser en un directorio separado de las Web de usuario y sólo tras su análisis profundo por parte del ISP. Esta característica será posible con las compilaciones que se hacen de Apache en las que el programa que lanza las ejecuciones está *chrooteado* para que sólo trabaje sobre */var/lib/cgi-bin* o directorios concretos.

Acceso FTP

Cada vez que el usuario desee modificar algún fichero de la página Web, habrá de acceder mediante su usuario y contraseña mediante FTP y reemplazar los archivos que en cada directorio concreto haya colocado previamente.

Del acceso FTP se registrará tanto la fecha y hora como el usuario e IP desde la que se produjo. No se permitirá el listado de los directorios que no dependan del directorio del usuario, siendo su directorio de usuario lo único que podrán ver al acceder por FTP.

Los únicos ficheros no visibles a los usuarios y que puedan estar en su propio directorio de usuario serán aquellos que comiencen por punto, por tratarse la mayoría de veces de los ficheros para suministrar seguridad en el acceso a los directorios. Si se hubieran de activar restricciones de acceso a determinado directorio, ni el panel de control vía Web ni por FTP podrán realizarlo los clientes, necesitándose la intervención humana para esto.

No será válido el mismo usuario para otro servicio distinto del Web y base de datos. Estas cuentas tendrán totalmente restringido otros accesos como el acceso a correo o a consola de comandos, por no ser este su objetivo. Todos los usuarios de correo pertenecerán al mismo grupo de usuarios, el grupo "*webftp*", con independencia del cliente al que pertenezcan, y se establecerán restricciones en el uso del disco, permitiéndose únicamente que dispongan de espacio de

almacenamiento en el directorio */home/webftp*, donde estarán sus directorios de usuario y por tanto también las páginas.

Las restricciones de uso de espacio en nuestros sistemas se implementarán mediante la versión 2 del sistema disponible en UNIX para sistemas de archivos ext2 (conocida como vfsv0), en el que existen dos tipos de límites (del que sólo estableceremos el límite real, sin margen de gracia o similares). Además existe la posibilidad de restringir el número de ficheros, pero nosotros sólo limitaremos el total de espacio que puede ocupar el usuario.

Cada usuario tendrá una cuota de espacio máximo diferente dependiendo del producto contratado. Para facilitar el trabajo tanto la cuota en uso como el identificador del cliente al que pertenece este espacio Web serán parte integrante de la descripción del usuario, de manera que un cliente cuyo identificador es el 37, y tiene asignados 5 MB de espacio en disco, tendría por descripción "37-5mb-XXXXXX", siendo XXXXXX cualquier otro texto que pueda ayudar a describir mejor a este buzón.

Bases de datos

El usuario y la base de datos que se creen, junto con el espacio Web, no serán accesibles a través de Internet para consultas o modificaciones, por razones de seguridad. Los datos de esta base de datos serán accesibles únicamente por usuario y contraseña en el propio servidor y desde algún fichero de PHP como parte de una ejecución encaminada a mostrar esos valores a través de Internet.

Se usará el mismo usuario pero diferente contraseña para usuario físico con acceso FTP al espacio Web, para evitar que una pérdida de una contraseña suponga poner en compromiso los datos o al revés.

Los datos se podrán cargar de manera masiva en la base de datos usando el propio panel de control, de la siguiente forma: se suministrará un control para subir un fichero de texto o comprimido, que contendrá sentencias SQL únicamente y de cuya ejecución y resultado el usuario de la página Web tendrá constancia por la pantalla del navegador igualmente.

La base de datos por defecto creada será en *MySQL*, un producto algo simple si lo comparamos con otros sistemas de bases de datos relacionales como *PostgreSQL* u *Oracle*, pero suficiente para satisfacer las necesidades de los clientes que esperamos tener.

MySQL permite usuarios, por lo que no habremos de preocuparnos por lo que respecta al aislamiento de los datos de un cliente con respecto al otro: en ambos casos el usuario creado tendrá acceso a la base de datos con el mismo nombre creada. No se permitirá que un mismo usuario comparta el acceso a varias bases de datos-

6.2.3 Servicio de Correo

Usuarios

El servicio de correo estará constituido por un abanico amplio de protocolos, y no únicamente el servicio POP3 y el SMTP. En concreto se permitirá al cliente final usar tanto POP3 y SMTP como IMAP4 para que escoja cuál de estos protocolos desea usar.

Los usuarios tendrán cuentas físicas en el sistema, cuentas protegidas mediante contraseñas que el usuario no podrá escoger, para evitar que sean usadas contraseñas fácilmente predecibles. La contraseña será modificable a través de la página Web desde el propio panel de control del cliente, panel al que llegará mediante otro usuario y contraseña válidos únicamente para administrar todo el conjunto de servicios que el cliente haya contratado.

Se implantará también un servicio de *Webmail* que permita el uso a través de la Web del correo por parte de los clientes, servicio que estará alojado bajo el propio dominio del cliente en "*webmail*" (por ejemplo una entrada válida sería **http://webmail.bith.net**).

No será válido el mismo usuario para otro servicio distinto del correo (como podría ser el Web). Por tanto estas cuentas tendrán totalmente restringido otros accesos como el acceso FTP o a consola de comandos, por no ser este su objetivo. Todos los usuarios de correo pertenecerán al mismo grupo de usuarios, el grupo "*mbox*", con independencia del cliente al que pertenezcan, y se establecerán restricciones en el uso del disco, permitiéndose únicamente que dispongan de espacio de almacenamiento en el directorio */var/spool/mail*, donde estarán sus correos pendientes de ser leídos.

Las restricciones de uso de espacio en nuestros sistemas se implementarán mediante la versión 2 del sistema disponible en UNIX para sistemas de archivos ext2 (conocida como vfsv0), en el que existen dos tipos de límites (del que sólo estableceremos el límite real, sin margen de gracia o similares). Además existe la posibilidad de restringir el número de ficheros, pero nosotros sólo limitaremos el total de espacio que puede ocupar el usuario.

Cada usuario tendrá una cuota de espacio máximo diferente dependiendo del producto contratado. Para facilitar el trabajo tanto la cuota en uso como el identificador del cliente al que pertenece este buzón de correo electrónico serán parte integrante de la descripción del usuario, de manera que un cliente cuyo identificador es el 37, y tiene asignados 5 MB de espacio en disco, tendría por descripción "37-5mb-XXXXXX", siendo XXXXXX cualquier otro texto que pueda ayudar a describir mejor a este buzón.

La autenticación de usuarios será vía usuario y contraseña al usar el protocolo POP3 o el protocolo IMAP, y mediante autenticación LOGIN o PLAIN en el caso de los clientes que deseen enviar correo.

Finalmente se facilitarán a los clientes características añadidas a su cuenta de correo, como la posibilidad de configurar mensajes de autorespuesta. Para ello obligaremos al MTA a entregar a *procmail* los mensajes de correo electrónico destinados a los usuarios locales, y usaremos un fichero de configuración de dicho programa para cada usuario donde se insertará un patrón predefinido que responda con el contenido de un fichero que podrá editarse desde la Web, cuando se active el mensaje de autorespuesta.

Servidor SMTP

El MTA escogido será el servidor *Postfix*, en detrimento de *Sendmail*. Ya en los antecedentes justificamos las razones que nos llevaban a considerar mejor el rendimiento y la configuración de este MTA, luego no repetiremos aquí ese análisis.

El servicio de correo estará autenticado y permitirá únicamente el *relay* de correo para las direcciones locales, abriéndose el *relay* para cualquier destino únicamente si el usuario previamente se ha autenticado ante *Postfix* con el mismo usuario y contraseña que se le solicita para recibir su correo. Los correos que envíen estos usuarios quedarán registrados, al igual que los recibidos, a partir de la fecha y hora de entrada, su identificador único, y el remitente o destinatario en cada caso. Tanto en el caso de que el *email* sea entregado como en el que sea rechazado, en el registro debe figurar esta circunstancia. Estos registros serán guardados para cumplir la legislación vigente durante al menos 6 meses. El fichero que contiene toda esta información será */var/log/mail.log*, que semanalmente será copiado de manera rotatoria sobre */var/log/mail.log.1*, */var/log/mail.log.2*, etc. hasta que llegue a la posición 24 en que será eliminado. Se necesitará un espacio de almacenamiento elevado para esta actividad de registro del tráfico de correo de nuestro servidor.

Por lo que respecta al servidor de backup que tendremos alojado en el exterior de las instalaciones del ISP, éste se limitará a permitir el *relay* de los dominios alojados en el servidor principal, únicamente permitiendo el tráfico en el puerto 25.

Ambos servidores de correo tendrán configurados los tiempos habituales en Internet en lo que respecta al tratamiento de los errores temporales (aquellas respuestas numéricas del rango 4XX que los *relays* de Internet ofrezcan a un correo que nuestro servidor de Internet intente entregarles). Estos mensajes son encolados de nuevo para su posterior intento de entrega por el MTA tras un periodo habitual de 2 horas, tiempo que usaremos. El estado en el que estos mensajes quedan en la cola de *Postfix* es *deferred*. El MTA tendrá además configurado que realice un máximo de reintentos, espaciados cada vez más en el tiempo, hasta que tras dos días devuelva (*bounce*) al remitente el mensaje.

El correo no solicitado

En la actualidad el correo no solicitado o SPAM constituye uno de los mayores retos para un administrador de redes, siendo una batalla que periódicamente resurge de nuevo conforme las diferentes barreras que se habilitan para luchar contra él van cayendo.

El *spamming* (la acción de inundar un servidor de correo con correo no solicitado) puede suponer al ISP gran cantidad de ancho de banda y perjuicios en la imagen (los clientes son muy sensibles a verse inundados por correo que no han solicitado y perder tiempo en su eliminación). Por *spam* hablaremos también de la suplantación de personalidad o falsificación de identidad, cosa harto fácil con el protocolo SMTP.

Como cambiar el protocolo SMTP no es una solución, ni tampoco lo está siendo la regulación legal y penal que en algunos países se lleva a cabo, la lucha contra el *spam* es tarea de cada ISP y de un correcto diseño de su servicio de correo. Iremos ahora viendo las diferentes técnicas que usaremos, partiendo de las más elementales y antiguas hasta comentar las últimas soluciones al respecto.

Lo primero que habremos de considerar si vamos a utilizar un filtrado *antispam* en nuestro MTA es su posible impopularidad ante los usuarios finales de nuestro servicio: el identificar SPAM puede conducir a un porcentaje de falsos positivos (mensajes válidos que nuestro MTA marca como SPAM) que habremos de asumir, por muy bajo que pueda éste resultar, en beneficio del buen funcionamiento del sistema.

Verificación de las direcciones de correo suministradas en el envoltorio del mensaje

Antes de transmitir el contenido de un mensaje, el servidor de correo remitente y el de destino intercambian las direcciones del remitente y del destinatario, direcciones correspondientes al envoltorio del mensaje y que serán utilizadas durante la entrega del mensaje. Este filtrado inicial es la mejor forma de tratar el SPAM, ya que evita un consumo de tiempo y recursos elevado (sólo se llegan a intercambiar unos pocos bytes en lugar de todo el mensaje).

Lo normal hasta ahora había sido relajar tanto la negociación entre el MTA origen y destino del mensaje (no exigiendo el saludo HELO o EHLO inicial que el estándar define) como evitar comprobar que estas direcciones del envoltorio fueran o no reales. Será pues este la primera norma que modificaremos: nuestro MTA exigirá un saludo por parte del MTA remitente, saludo que además habrá de corresponder a un nombre de un dominio válido en Internet (se realizará una resolución inversa para verificarlo), rechazándose en caso de no poderse resolver de manera inversa el nombre suministrado durante el saludo inicial.

El nombre suministrado por los clientes de correo electrónico en el caso de que quien conecte a nuestro MTA no sea otro MTA sino un cliente nuestro que desea enviar un correo, tampoco ha de

preocuparnos: los clientes de correo electrónico usan el dominio del correo electrónico del remitente, con lo que este no será denegado por nuestro servidor. Para el caso de nuestros clientes exigiremos además la autenticación durante el envío mediante alguno de los métodos a los que daremos soporte en nuestro MTA (DIGEST-MD5, NTLM, LOGIN, PLAIN, y CRAM-MD5, verificando además que funciona de manera correcto para los clientes de correo más habituales).

Si se trata de un cliente nuestro que desea enviar le exigiremos además que configure su cuenta con la dirección de correo principal correspondiente al usuario en cuestión, por ejemplo: la cuenta con nombre de usuario *hector* y dirección **hector@bith.net** sólo podrá enviar correo a través de nuestro servidor si en el envoltorio del mensaje tras su autenticación coincide esta dirección con este usuario, impidiéndosele que pueda usar nuestro servidor para enviar con otra dirección cualesquiera.

Sobre la dirección del remitente del mensaje se realizará además siempre la resolución inversa de su dominio: en el caso del correo generado por nuestros clientes hacia Internet esto supondrá un retraso extra y un paso innecesario, pero en cambio permite evitar gran cantidad del SPAM, que muchas veces usa direcciones de envío totalmente falsas, aunque esta técnica está perdiendo cada vez más su efectividad, conforme los *spammers* van descubriendo que deben usar direcciones reales.

Listas negras (*blacklists*)

Consiste básicamente en mantener una base de datos de IPs tras las cuales en algún momento ha habido *relays* abiertos. Por *relay* abierto hemos de entender cualquier MTA que acepta procesar correo que no esté originado o destinado a un usuario local. Nosotros por usar la autenticación SMTP evitaremos esta circunstancia, pero si alguien no lo evita es un posible candidato a ser usado por un *spammer* para inundar a terceros ISP (entre ellos nosotros) con sus correos. Por existir ese riesgo potencial se descarta cualquier correo proveniente de dicha IP.

La manera de mantener esta base de datos centralizada es mediante el DNS y la resolución inversa una vez más: el MTA destinatario, en cuanto se conectan a él, lo que hace es obtener una resolución para la IP del remitente: si esta respuesta es positiva (existe un nombre asignado a dicha entrada) se aceptará el correo si no incumple alguna otra regla. Por el contrario, si está presente en la base de datos de *relays* abiertos, se producirá una respuesta negativa y el MTA sabrá pues que deberá rechazar este correo, informando además de manera adecuada al MTA remitente.

Estamos pues ante un método de lucha contra el SPAM muy polémico: de manera arbitraria una IP marcada por una de estas listas como *relay* abierto tiempo atrás puede provocar rechazos para MTAs legítimos instalados posteriormente (luego una de nuestras primeras acciones una vez conozcamos las IPs asignadas por nuestros

proveedores será verificar que están libres de marcado en lista alguna). Además existen en la actualidad listas que directamente incluyen cualquier IP asignada a conexiones de dial-up en sus listas negras (por lo que cualquier ADSL o conexión de cable con IP dinámica quedaría bloqueada para el usuario de dichas listas).

Además, una lista negra no garantiza que el remitente sea un *relay* abierto, sino que en algún momento lo fue, e incluso puede serlo y darse una respuesta positiva debido a que nadie hasta ahora haya comprobado que la IP del remitente era un *relay* abierto.

Otro aspecto polémico viene dado por la forma en que se introducen a los hosts en dichas listas y se borran de las mismas: un MTA puede ser incluido nada más a instancias de una denuncia efectuada por un tercero, tras lo que un sistema automático suele hacer una serie de verificaciones conectando al MTA. Si el resultado es positivo, se informa al administrador de dicho MTA un número predeterminado de veces (dos o tres) y tras un tiempo de espera, se incluye al MTA en la lista negra tras una segunda comprobación de que no han solucionado el problema. Luego existen métodos más o menos estandarizados para solicitar la baja de una de estas listas, pero básicamente consiste en repetir de nuevo el *test* anteriormente incumplido para verificar que han solucionado el problema que causaba que el MTA fuera considerado un *relay* abierto.

El uso de las listas negras fue muy útil durante aquellos años en los que se pasó de MTA totalmente abiertos al tráfico a la situación actual en la que es complicado encontrar un MTA que no exija autenticación SMTP o por lo menos, métodos como el *pop-before-smtp.*, que nosotros directamente no ofreceremos (consiste en abrir el acceso para enviar correo a aquellas IPs que acaban de autenticarse para descargar el correo, abriéndolo durante un cierto periodo de tiempo, como por ejemplo diez minutos).

Pero en la actualidad el SPAM ha pasado de esa fase en la que usaban *relays* abiertos a una fase en la que directamente ataca desde múltiples IPs, IPs que además se van modificando de manera constante por tratarse de conexiones de dial-up. Como cerrar a cualquier conexión de dial-up el acceso es sencillamente una barbaridad (dejaríamos fuera a mucho correo legítimo), y pese a que sí emplearemos las listas negras para bloquear a los *relays* abiertos, no subiremos tanto el escalón como para cerrar cualquier tráfico SMTP desde IPs marcadas como conexiones de dial-up.

Las listas negras más famosas y de mayor prestigio son la *Open Relay Database* (**www.ordb.org**), cuya resolución DNS para detectar un *relay* abierto (**relays.ordb.org**) usaremos. Importante destacar que no es lo mismo usar la comentada **relays.ordb.org** que **relays.ordb.com**: esta segunda corresponde a uno de los detractores de estas listas y directamente ofrece respuestas positivas con independencia de la IP sobre la que se realice la petición. Cuando un mensaje sea rechazado por esta causa, en el texto de error habrá de aparecer el mensaje en

inglés "550 Email rejected due to sending server misconfiguration - see http://www.ordb.org/faq/#why_rejected", para facilitar información que permita exigir al cliente remitente del mensaje que el MTA que está usando corrija esta situación.

Listas blancas

Suponen una evolución de las listas negras, pero son hoy por hoy inviables. Por lista blanca se entiende que el ISP dispondría de una base de datos de hosts de confianza, es decir: no consiste en denegar el correo por IP de origen, sino en aceptar únicamente el correo proveniente de los otros ISP conocidos.

Básicamente es inviable por dos motivos: una su arbitrariedad, ya que a estas alturas es difícil decidir quién es ISP y quién no, así como qué ISP aplica de modo honesto una política *antispam* que evite que al resto de ISP llegaran correos generados por terceros a través de él. La otra razón es su rigidez: los ISP deberían publicar las IPs de sus MTA y no modificarlas so pena de dejar de poder enviar correo a otros ISP.

Lo que sí que resulta viable es hacer que nuestro MTA no sea quién decida sino delegar en un tercero o terceros MTA, y dejar que nuestro MTA sólo acepte dos tipos de correo: el originado por las conexiones autenticadas con usuario y contraseña por nuestros clientes, y el originado por el MTA que haría de *relay* para nuestros dominios y en cuya política de filtrado se confía. Dado que deseamos diseñar un ISP autónomo a este respecto con respecto de otros ISP, esta opción es inviable.

Listas grises (*greylisting*)

Son una de las últimas evoluciones en la lucha contra el correo no solicitado, con un uso muy limitado de recursos en el MTA, al rechazar el correo a partir de la información disponible en el envoltorio.

Aunque está aún en fase de testeo (hasta el momento sólo se ha podido probar en sistemas de pequeña escala), la idea es bastante original: el MTA mantiene una base de datos automática compuesta por tripletes formados por:

- La IP desde la que se remite el mensaje.
- Dirección remitente del correo presente en el envoltorio (dirección MAIL FROM).
- Dirección destinataria del mensaje presente en el envoltorio (dirección RCPT TO).

Este triplete es utilizado para determinar si es la primera vez que se produce un correo con esa combinación. Si no es la primera vez se acepta, pero en caso contrario se produce un mensaje de error temporal del servidor, del rango numérico 4XX.

¿Qué pasa ante un error de este tipo? Si el mensaje ha sido originado por un *spammer*, lo normal es que el programa que ha

utilizado directamente no intente de nuevo la entrega, dado el elevado coste en recursos que supondría para él (habría de mantener una cola de mensajes devueltos y un tratamiento para estos, lo que supondría consumo de disco y CPU).

Como ejemplo de la situación que se daría en la negociación entre nuestro MTA y uno externo, veamos cuál es el comportamiento normal al entrar un mensaje de correo (las flechas permiten distinguir las respuestas del servidor, marcadas como ←, de los mensajes que enviaría un cliente, →):

```
→ HELO smtp.mta.origen.es
← 250-smtp.mta.destino.es
→ MAIL FROM: <origen@dominio.es>
← 250 2.1.0 Sender ok
→ RCPT TO: <destino@dominio.es>
← 250 2.1.5 Recipient ok
→ DATA
← 354 Enter email
→ (contenido del mensaje)
→ .
```

Pues bien, lo que pretendemos es hacer que la primera vez que se nos entregue un correo desde una dirección desconocida, la respuesta sea la siguiente:

```
→ HELO smtp.mta.origen.es
← 250-smtp.mta.destino.es
→ MAIL FROM: <origen@dominio.es>
← 250 2.1.0 Sender ok
→ RCPT TO: <destino@dominio.es>
← 451 4.7.1 Intente de nuevo en 1 hora.
```

Lo normal es que los programas empleados para hacer *spamming* acepten en su entrada un mensaje, una lista de destinatario por otro lado, y se limiten a entregar los mensajes, sin esperar la aceptación del mensaje por el MTA destinatario del mismo. En cambio un MTA real, ante este mensaje de error temporal lo que hará será encolar el correo y tras una espera de una o dos horas, reintentar la entrega de nuevo. Será entonces cuando aceptemos realmente el correo. Durante los testeos que los diseñadores de este sistema de bloqueo de *spam* hicieron se pudo comprobar que existía una efectividad del 95%[60], aunque estos índices bajarían considerablemente si el remitente comenzara a usar un MTA correctamente configurado (ya que al reintentarlo el correo sí que pasaría), pero en este caso el *spammer* se vería obligado a involucrar un elevado número de recursos que pueden hacerle desistir.

[60] Harris, Evan (2.003): *The Next Step in the Spam Control War: Greylisting* <http://projects.puremagic.com/greylisting>

Por supuesto el método tiene dos inconvenientes principales, que hacen difícil su implantación generalizada inmediata:

- Todo el correo legítimo sufre un retraso en su entrega al fallar la primera entrega, si estamos ante tripletes que nunca se han producido antes (es decir, comunicaciones por parte de remitentes no previamente conocidos hacia destinatarios de nuestros sistemas). Esta característica es inevitable porque en ella reside justamente la fuerza del sistema, pero es difícil de explicar y justificar frente a los clientes, por lo que es importante que se dediquen recursos a explicar de manera adecuada al cliente que estos sistemas de bloqueo redundan en su beneficio y en el de todos.
- Aunque nuestro MTA no sufra un incremento de uso de los recursos más allá del mantenimiento de la base de datos, en el caso de los MTA que se nos conectarán para tratar de enviarnos correo necesitarán encolar más correo del que usarían si no estuviera en uso este tipo de listas, ya que el MTA tendrá que encolar de nuevo el mensaje para su posterior entrega.

A cambio el sistema no requiere mantenimiento por parte del administrador de la red, y además es mucho menos arbitrario que las listas negras, ya que no hay ninguna dirección IP a la que se imposibilite el envío.

En el diseño de este sistema para *Postfix* aún no se ha llegado a una versión definitiva, pero es bastante simple por diseño: a partir de los tres datos comentados anteriormente, devuelve un único valor lógico, que será la aceptación o no de ese mensaje.

El sistema ha de mantener una serie de tiempos asociados al triplete en la base de datos:

- La fecha en la que el triplete fue visto por primera vez (fecha de creación).
- La fecha en la que el bloqueo inicial establecido al crearse la entrada caducará (se exigirá pues un tiempo mínimo de por ejemplo 1 hora entre la primera aparición y la aceptación, de forma que si el remitente reintenta un minuto después del primer intento no sea aceptado por el sistema).
- La fecha en la que el triplete será eliminado de la base de datos (con lo que si es baja los remitentes que se comuniquen de manera ocasional pasarían de nuevo por el engorro de tener que sus mensajes ser retrasados una hora en su entrega).
- El número de intentos de entrega que han sido bloqueados.
- El número de mensajes que cumplían con el triplete dado y que han sido aceptados.

Por diseño este sistema permite además si se realiza a posteriori un control de las entradas presentes, detectar las IPs que habitualmente vienen siendo bloqueadas de manera automática. También es posible (y recomendable) lo contrario: mantener una lista de las IPs conocidas de los MTA de los ISP de gran tamaño y de los que habitualmente vendrá la mayoría del tráfico desde Internet hacia nosotros, eliminando para dichas IPs el inconveniente causado por el *greylisting* (estas IPs de confianza podrían evitar ser filtradas a través de este sistema, y sus correos pasar directamente a la entrega en todos los casos.

La metodología básica que ha de seguir el programa que sobre *Postfix* ha de filtrar los correos es el siguiente:

1. Si la IP remitente está listada en la lista blanca de excepciones, admitir en todos los casos.
2. Si el triplete no está presente, se crea un nuevo registro y se devuelve como respuesta al MTA remitente un fallo temporal 451.
3. Si el triplete está presente y la fecha de caducidad aún no se ha alcanzado, denegar de nuevo con error 451.
4. Si el triplete está presente y la fecha de caducidad ya ha sido superada, pasar el correo.

Pero el sistema tiene dos fallos principales que nos obligarán a cambiar un poco el comportamiento general del sistema:

- El primer problema será la existencia de ISP en los que se utilice un pool de servidores de correo para sacar a Internet el correo, gestionado mediante *round-robin* o similar, con lo que un correo remitido por la misma persona nos puede llegar cada vez desde una IP distinta, y por cada IP se produciría un retraso.
- Otro problema vendrá por el lado de las direcciones únicas que en algunos mensajes pueden aparecer, generadas por las listas de correo o las respuestas automáticas en las que parte de la dirección remitente es un identificador aleatorio o similar (del tipo **bounce-3249234@respuestas.bith.net**). Estas direcciones supondrían un triplete nuevo en cada caso, por lo que la base de datos crecería mucho y los retrasos se producirían siempre, no únicamente ante la primera aparición de ese correo.

Para evitar estos dos problemas de manera simple, el triplete cambiará: en lugar de fijarnos en la dirección remitente, nos fijaremos únicamente en el dominio remitente, con lo que evitaremos que las direcciones únicas vayan a suponer un coste más elevado de procesamiento en este sistema (se comprobaría únicamente el dominio **@respuestas.bith.net** en lugar de toda la dirección). Para evitar el otro problema la solución es similar: normalmente los ISP que usan un pool de servidores de correo suelen usarlos para descargar la carga y no por

razones de eficiencia en el ancho de banda, con lo que al no usarse conexiones distintas todos los servidores están en un mismo rango consecutivo de direcciones. Por tanto con usar la clase C equivalente en lugar de la IP concreta nos evitará que cada servidor del pool necesite pasar por la fase extra de rechazo inicial.

Ni que decir tiene que el sistema de *greylisting* y en general todos los sistemas de bloqueos hasta ahora vistos deben ser implementados por igual tanto en el servidor principal como en el previsto servidor de correo (no así el filtrado basado en el contenido que ahora veremos): necesario sobretodo porque que si el *relay* secundario admitiera estos correos luego el principal no podría saber la IP real que los originó de una manera tan simple como hasta ahora, además de que es muy común que los *spammers* ataquen tanto el *relay* principal de un dominio como cualquier otro *relay* listado.

Análisis del contenido del mensaje

Este método supone ya un considerable consumo del ancho de banda del ISP: se trata de la técnica o conjunto de técnicas destinadas a descubrir mensajes considerados SPAM a través de un análisis léxico o sintáctico del mismo, comparando el mensaje con patrones predefinidos. Por tanto esta técnica acepta totalmente el correo previamente a su análisis, evitándose únicamente la entrega al cliente.

Acarrea también otro problema: el correo rechazado en este análisis puede ser devuelto al remitente (duplicándose de nuevo el consumo de ancho de banda), o bien ser directamente eliminado. Si optamos por esta segunda característica los falsos positivos que el sistema genere darán lugar a que el remitente crea realmente haber entregado algo que finalmente no se colocó en el buzón del destinatario.

Es por ello por lo que optaremos por no devolver al remitente los mensajes marcados por *spam*, pero tampoco los borraremos: todo el correo que el sistema marque como *spam* se colocará en un buzón destinto al correo legítimo. Este buzón, accesible a través del Webmail podrá ser consultado posteriormente por el cliente para que él mismo verifique que mensajes fueron desviados hasta allí y poder recuperarlos.

Para que todo esto sea posible usaremos *Spamassassin* en conjunción con *Amavis*, que actuará de filtro tras la entrega de los mensajes al MTA. Todo correo que pase a través de nuestro sistema de correo será entregado al filtro *Amavis* (posteriormente veremos que este filtro nos servirá además para facilitar el análisis en búsqueda de virus.

La forma de funcionar de *Spamassassin* es la siguiente: el mensaje que le es entregado es analizado partiendo de una tabla de características y patrones que se pueden encontrar en la mayoría de los mensajes considerados *spam* (palabras como free, sex, etc.), características que tienen asignado un peso. El programa se limita a sumar los pesos de las características aparecidas y devolver el mensaje con una nueva cabecera llamada “X-Spam-Status”. Esta cabecera tiene por valor la puntuación asignada (más cercana a cero denota una

menor presencia de características de *spam*, mientras un valor elevado supone lo contrario).

Podríamos dejar que directamente *Amavis* y *Spamassassin* borraran o evitaran la entrega si el mensaje entrante supera un determinado umbral definido a nivel del sistema. Pero preferiremos dejar que sea cada cliente el que a través de la página Web personalice su propio umbral.

Desde la página Web se podrá introducir un valor numérico que denotará para ese buzón el grado de aceptación de *spam*, informando de que el sistema tiene un comportamiento arbitrario con lo que a más elevado sea el valor introducido más riesgo de obtener falsos positivos (mensajes legítimos que por cierto patrón sean tratados como *spam*).

Para acabar de personalizar al gusto del usuario el sistema, se permitirá que cada usuario pueda además desde la Web introducir valores para *Spamassassin* (direcciones que desea rechazar, y palabras concretas que desea sean tenidas en cuenta como provenientes de *spammers*), al existir en este programa la opción de que exista un fichero de configuración por usuario aparte del global.

De cualquier forma hay que indicar que *Spamassasin* y *Amavis* serán útiles mientras los patrones y cadenas a buscar sean actualizados de manera permanente, ya que los *spammers* van adaptándose a estos filtros conforme van detectando qué palabras reservadas y cadenas son las que causan que sus mensajes sean marcados como *spam*.Por defecto además este filtrado estará totalmente desactivado y será el propio cliente el que en función del volumen de correo no solicitado que reciba podrá decidir activarlo o no.

La resolución inversa en el correo

La mayoría de las técnicas comentadas en la lucha contra el SPAM se basan en el uso de la resolución inversa, lo que implica en muchas ocasiones un retardo de tiempo.

En el caso de que el correo esté siendo originado por otro MTA, el incremento de tiempo debido a esta resolución inversa no va a ser apreciado, pero si el correo está siendo originado por nuestro cliente y su programa de correo electrónico el retardo producido puede ser interpretado como una mala calidad de nuestro ancho de banda disponible.

Como nuestro ISP carece de una red propia tras la cual se encuentren los usuarios autorizados a enviar correo con nosotros, no podemos por tanto distinguir el tráfico SMTP originado por nuestros clientes del recibido de Internet, a diferencia de lo que ocurre en una red corporativa como la de la Universitat Jaume I (los únicos hosts autorizados a enviar a cualquier destino son aquellos situados dentro del rango 150.128.0.0/16).

La única solución efectiva para evitar que el correo originado por los clientes pase por los mismos retrasos y filtrados que el causado desde

Internet hacia nuestros clientes pasaría por utilizar un puerto distinto al definido en el estándar, el puerto TCP 25. Esta solución no garantizaría tampoco el que no se produjeran accesos indebidos al servicio de correo destinado a clientes, por lo que la autenticación seguiría existiendo, pero permitiría que sobre este otro MTA distinto al disponible en el puerto 25 se usara una política diferente y similar a la que hoy día se viene aceptado para las redes corporativas, donde el MTA al que conectan las IPs de confianza aceptan todo el tráfico de correo en el menor tiempo posible, para realizar posteriormente y ya sin tener esperando al remitente el análisis del correo. Estos sistemas permiten que el usuario del mismo perciba una gran velocidad en el servicio, aunque a fin de cuentas se vaya a realizar posteriormente el mismo análisis sobre cada uno de los correos enviados.

Esta solución no es viable en nuestro caso: ni tenemos un rango de IPs de confianza ni tampoco deseamos complicar la configuración para el envío del correo a nuestros clientes haciéndoles configurar un puerto distinto. Esta solución es viable en aquellas circunstancias en las que prima la velocidad frente al consumo de ancho de banda (si aceptamos todo inicialmente, consumiremos más ancho de banda tanto en posteriores devoluciones que podían haberse evitado al rechazar durante el envío), y este no es el caso.

Explicar por tanto al cliente el funcionamiento del servicio, y hacerle comprender que el correo es analizado de manera minuciosa será por tanto necesario para que este no advierta una velocidad aparentemente baja debida fundamentalmente a la resolución inversa.

El filtrado del correo en busca de virus

Antes de la generalización de Internet, la vía de contagio de virus informáticos era habitualmente los disquetes. Desde hace ya años los virus entran en los equipos a través de Internet, y muy especialmente a través del correo, por lo que una primera barrera para luchar contra éstos es necesaria, y qué sitio mejor que el propio servidor de correo electrónico.

No sólo redundará en una mayor calidad del servicio apreciable por el cliente, sino que protege indirectamente al propio ISP, ya que cualquier cliente con una infección en sus equipos supondrá que el ancho de banda consumido por este cliente crezca de manera apreciable: los equipos infectados se convierten en difusores del virus, y se reenvían habitualmente a través del correo a todos las direcciones de correo electrónico que el virus puede localizar en el equipo infectado. Las últimas oleadas de virus conocidas (*Nimda* en el verano del 2.001, *Khlez* este pasado año, y *Tanatos* en los inicios del 2.003) han demostrado que es bastante sencillo saturar las redes sin necesidad de recurrir a gusanos, bastándose para ello que todos los ordenadores infectados de las redes corporativas se pongan al tiempo a enviar.

La infección a través del correo electrónico tiene además la característica de que una vez infectado el equipo no requiere ninguna

acción humana para su propagación: la infección se produce tras la entrada en el equipo de un virus con un adjunto especializado para pasar inadvertido al destinatario y aprovecharse al mismo tiempo de alguno de los innumerables fallos conocidos de los clientes de correo electrónico, pero para propagarse a Internet a través del correo electrónico, y ante la evidente falta de control sobre las conexiones por parte de los clientes (que no son capaces de advertir el incremento de uso de la red), es fácil que en un solo día un PC infectado pueda dar lugar a 15.000 correos infectados si permanece 24 en marcha, a una media de 10 correos infectados al minuto (perfectamente viable en una conexión de ADSL), lo que con un tamaño medio por correo infectado de 200 KB, supondría que en un solo día un cliente hubiera generado en el ISP un tráfico de 24 Gbits, todo un desastre si este tráfico es simultáneo a varios clientes infectados.

Dado que *Postfix* va a ser nuestro MTA y lo soporta, utilizaremos un filtro dentro del servicio de correo. Este filtro validará cualquier correo que haya sido aceptado por el servidor de correo en busca de adjuntos que contengan infecciones (ya que los virus únicamente viajarán como ficheros adjuntos). Esto va a suponer incrementar sensiblemente la carga y actividad en el servidor de correo, pero es necesario. Una vez analizado el correo el filtro entregará el correo para su entrega efectiva al buzón destinatario del mismo o al MTA para que sea transferido a Internet, pero en caso de encontrarse en el filtrado algún elemento sospechoso el correo deberá ser devuelto al origen, aunque la devolución no será total, ya que únicamente se remitirá la cabecera del mensaje en el que se encontró el virus y una notificación de la causa de no entrega del mismo.

La devolución al remitente del mensaje es hoy día un elemento de discusión más: lo habitual es que los virus falsifiquen no suministren una dirección de envío válida o directamente usen una dirección conocida pero que no coincide con la del usuario o máquina en la que se encontraba realmente la infección. Es por eso por lo que las devoluciones de estos mensajes constituyen a veces un problema añadido (vamos a devolver a un tercero un mensaje sobre algo que él no ha provocado), generando falsas preocupaciones en busca de virus en personas que lo más probable es que no sean las que hayan enviado el mensaje infectado.

Por otra parte, no notificar la no entrega de un mensaje incumple claramente el RFC 821 que define el protocolo SMTP, por lo que no podemos dejar de emitir estas notificaciones. La solución más adecuada pasa por permitir al filtro antivirus el notificar al remitente la infección, pero de manera que al filtrado antivirus sólo lleguen aquellos mensajes que hayan pasado antes por un filtrado que elimine aquellos que tengan direcciones incorrectas o tengan otras características que los hagan ser eliminados por el filtrado *antispam* (al fin y al cabo estos mensajes infectados constituyen correo no solicitado).

El filtro más adecuado por su modularidad es *Amavis*, que ya íbamos a usar para el filtrado *antispam*, de ahí que resulte doblemente

interesante. Este filtro modular es capaz de interactuar con una gran cantidad de productos antivirus que están disponibles en entornos Linux (que es el entorno en el que estamos trabajando), trabajando además con múltiples de estos al tiempo. El soporte para antivirus es además bastante amplio:

- NAI antivirus
- H+BEDV antivirus
- Clam antivirus
- Sophos Anti Virus
- KasperskyLab AntiViral Toolkit (AVP)
- F-Secure Antivirus
- CyberSoft VFind
- CAI InoculateIT
- GeCAD RAV Antivirus 8
- ESET Software NOD32
- Command AntiVirus para Linux
- VirusBuster
- Symantec CarrierScan
- Sophie (Sophos SAVI)
- Trophie (Trend API)
- FRISK F-Prot
- Panda Antivirus para Linux (este producto es gratuito y está creado a partir de su equivalente en plataformas Windows, que es realmente el nicho de clientela de Panda Software)
- CentralCommand Vexira
- OpenAntiVirus ScannerDaemon
- DrWeb Antivirus para Linux
- MkS_Vir para Linux, aunque sea una beta cuyo uso aún no está recomendado.
- Norman Virus Control
- Trend Micro FileScanner

Otra ventaja añadida de *Amavis* es que no necesitamos realmente un filtro antivirus para el correo como tal (lo que se conoce como *milter*, de la unión de las palabras *mail* y *filter*): *Amavis* es capaz de interaccionar tanto con los *milters* como con productos destinados a escanear archivos, ya que en esos casos lo que se hace es llamar al antivirus para que analice cada uno de los adjuntos que previamente *Amavis* ha separado y colocado en un fichero distinto.

Un *milter* suele resultar más eficiente en cuanto a tiempo total de evaluación del correo, pero resulta más peligroso por cuanto el MTA entrega a este filtro el correo, y el comportamiento esperado es que el *milter* entregue el correo ya analizado, o entregue un correo con el mensaje a enviar al remitente original y la causa de la denegación de dicho correo original. Si el *milter* está mal programado y causa un error en su ejecución, el resultado es que como resultado de nuestro análisis no se producirá salida alguna, por lo que el correo directamente desaparecerá, y esto ocurre en algunas circunstancias. Estos fallos

inesperados en el filtro antivirus no son muchas veces debidos más que a nuevos virus diseñados de manera específica para hacer caer algunos de los antivirus más conocidos, pero en cambio sería desastroso que un adjunto con un ejecutable totalmente legítimo desapareciera por este comportamiento inesperado del filtro antivirus.

En cambio, *Amavis*, a cambio de un considerable mayor consumo de CPU y disco (especialmente disco, ya que como ventaja del *milter* tenemos que únicamente consumimos memoria) se encarga de asegurar que ningún correo desaparecerá: por diseño este filtro únicamente entrega copias del correo a analizar a cada uno de los antivirus que se han configurado para ser usados: por diseño también del resultado de ese análisis en paralelo (ya que podemos estar al mismo tiempo usando más de un antivirus) se extrae una devolución del mensaje (si alguno de los antivirus detectó algún virus en alguno de los adjuntos), una aceptación del mismo o un fallo temporal en la entrega.

Cualquier MTA basa su comportamiento en el estándar fijado en el RFC 821, y tal y como ya hemos descrito cuando comentábamos el tratamiento del correo con técnicas como el *graylisting*, es posible hacer que el correo que acaba de entrar se vuelva a encolar para que pasado el tiempo predefinido en el MTA (4 horas en nuestro caso, usaremos) se vuelva a intentar su entrega al filtro antivirus.

De esta manera nos aseguramos de que el correo no va a ser eliminado debido a un fallo de programación en el filtro (por diseño de *Amavis* se garantiza que el código responsable de estas tareas está testeado y no ofrece tantas problemáticas como los *milters*).

El antivirus de Kaspersky Labs

De entre el abanico de antivirus posibles para *Amavis*, escogeremos el creado por el ruso Eugene Kaspersky y su compañía Kaspersky Labs (**www.kaspersky.com**). Desde 1.989 cuando lo normal era que el virus dibujara virguerías en la pantalla hasta hoy día, cuando lo normal es que el virus dañe más a las redes del operador o a servidores conocidos en Internet que al propio usuario infectado, ha demostrado ser un producto que ha sabido adaptarse a la evolución en los virus.

En la actualidad existen un número elevadísimo de virus conocidos y esta cifra se incrementa de manera considerable cada día. De un buen antivirus se espera que sea capaz de analizar un fichero en un tiempo razonable (un analizador que busque un conjunto de patrones tan elevado es complejo) como que sea capaz de reconocer nuevos virus de manera rápida. Las actualizaciones a través de Internet del sistema antivirus son necesarias, y diariamente Kaspersky suministra estas actualizaciones de manera eficaz. Existiendo otros productos con iguales características, y sin entrar a discutir las diferencias, el bajo precio de este producto de *Kaspersky* es la otra razón que nos hace decidirnos por él. Al ser el filtro *Amavis*, no nos va a ser necesario adquirir el *milter* de que dispone *Kaspersky*, y el producto orientado a

los servidores y capaz de analizar ficheros tiene un coste bastante inferior a otros equivalentes. Las actualizaciones serán diarias.

6.2.4 Servicio DNS

El servicio DNS estará disponible tanto en el servidor principal como en el servidor de backup, siendo ambos servidores respuesta autoritativa para los dominios que aquí se configuren.

Se permitirá cualquier TLD soportado por nuestro registrador, evitándose mantener en el servidor dominios que no hayan sido facilitados por nuestro registrador: si un dominio ha de ser alojado en nuestros servidores antes ha de ser trasladado a nuestro registrador, para facilitar la gestión integrada de estos servicios DNS.

El servidor DNS será en ambos casos *Bind*, por ser un software robusto y universalmente usado. Al existir diferentes IPs que alcancen al servidor principal (una por cada conexión, luego dos IPs), y una tercera IP que tendrá el servidor de backup, se configurarán tres entradas para el DNS:

dns1.nuestroisp.es (una de las IPs en el servidor principal)
dns2.nuestroisp.es (la IP del servidor de backup)
dns3.nuestroisp.es (la otra IP del servidor principal)

El crecimiento en conectividades de nuestro ISP supondrá asimismo ir colocando de manera consecutiva dns4, dns5, etc., posibilitando que las peticiones DNS se repartan entre la totalidad de las conexiones y permitiendo pues una mayor tolerancia a fallos.

La jerarquía de todos estos servicios DNS que ofreceremos estará basada en que el servidor principal actúe de maestro de la zona DNS y el servidor de backup de esclavo de la misma. El panel de control y nuestro software habrán de ser capaces de crear la zona DNS en ambos servidores en una sola acción, sin que deba haber intervención humana (al igual que en la petición del dominio).

De los parámetros de configuración de cada zona DNS, los tiempos de vida de las zonas serán los más relevantes: se asignarán periodos de 5 horas para el tiempo de vida, considerado un valor medio suficientemente bajo como para que si una conexión falla de manera grave pueda usarse el DNS para modificar el enrutamiento del tráfico hacia otra conexión.

Por otro lado, se definirán en las zonas DNS valores elevados de tiempos de expiración para que en caso de que falle el servidor principal el servidor de backup (que actuará de esclavo por lo que respecta a la DNS) no deje de responder con su copia autoritativa de la zona.

La resolución DNS en nuestra red local

Los servidores DNS (ambos) no serán recursivos, ya que su única función es dar respuesta a las zonas DNS allí configuradas. La única

red para la que se permitirá la recursividad es para la red local de trabajo, ya que ésta necesitará Para la red local de ordenadores de trabajo se realizará la resolución DNS en el propio router, el cuál además no será accesible desde Internet.

En cambio al usar NAT en alguna de las conectividades que contrataremos, esto nos producirá un pequeño efecto colateral que no habíamos previsto y que ahora vamos a comentar.

NAT es la abreviatura del inglés *Network Address Translator*, y consiste en lograr que un conjunto de hosts con espacio de direccionamiento privado puedan obtener información de Internet a través de una única IP pública. Se utiliza por tanto para permitir que los equipos de trabajo de nuestra red local, sin disponer de direcciones públicas por razones de costo y justamente porque no van a ofrecer servicio alguno a Internet, puedan acceder sin embargo a éste.

El router con capacidad NAT sustituye en todas las peticiones a Internet la IP privada por la pública de que él dispone. El router usará un mecanismo muy simple para conocer a qué hosts concreto de la red privada ha de remitir la respuesta que de Internet llegue (ya que las respuestas contendrán también la misma IP pública, no la privada que es realmente quien espera recibir el paquete): divide el conjunto de posibles puertos TCP y UDP entre el conjunto de hosts con direccionamiento privado, de forma que un determinado puerto estará mapeado a un determinado host, puerto que no ha de coincidir con el que usó el host privado para realizar la petición. Otra opción consiste en que el router realice un *tracking* de todo el tráfico saliente a Internet para luego conocer en el entrante quién es el solicitante real, pero esta opción supone muchos más recursos y no evita además el otro problema que el NAT tiene, que es que no puede funcionar con un elevado tráfico y para un número elevado número de hosts (el número total de puertos que el protocolo IP soporta está limitado a 65.000, y por tanto es inviable que el NAT sirva más allá de redes corporativas de a lo sumo cien o doscientos hosts).

Pero también queremos utilizar de manera útil esta conexión con NAT para los propósitos del ISP, que es ofrecer en Internet servicios. NAT permite de manera adicional asignar a qué hosts concretos dentro de nuestra red privada se transmitirán las peticiones que lleguen a puertos concretos de nuestra IP pública. En concreto los puertos privilegiados objeto de algún uso por parte del ISP (Web, correo, DNS...) serán mapeados para que sea posible que desde Internet y con dicha IP pública se alcance a nuestro servidor principal.

El efecto colateral que comentábamos se va a producir por el uso de NAT al respecto de la resolución DNS en nuestra red local de trabajo y en el propio servidor: en ambos casos, las peticiones de resolución DNS que tanto el propio servidor como la red local realicen sobre aquellos dominios configurados sobre la IP pública en la que usamos NAT nos devolverán una IP a la que nos será imposible conectar.

Veámoslo con un ejemplo: si tenemos asignado a router la IP pública 128.100.0.2, y desde esa IP se ha mapeado hacia el host con dirección privada 172.16.0.2 el servicio Web: tanto ese propio host como todos los de nuestra red privada (172.16.0.3 y demás) no podrán conectar a la IP 128.100.0.2 porque el router hacia el que se dirigirán sus peticiones van a localizar en Internet dicha IP.

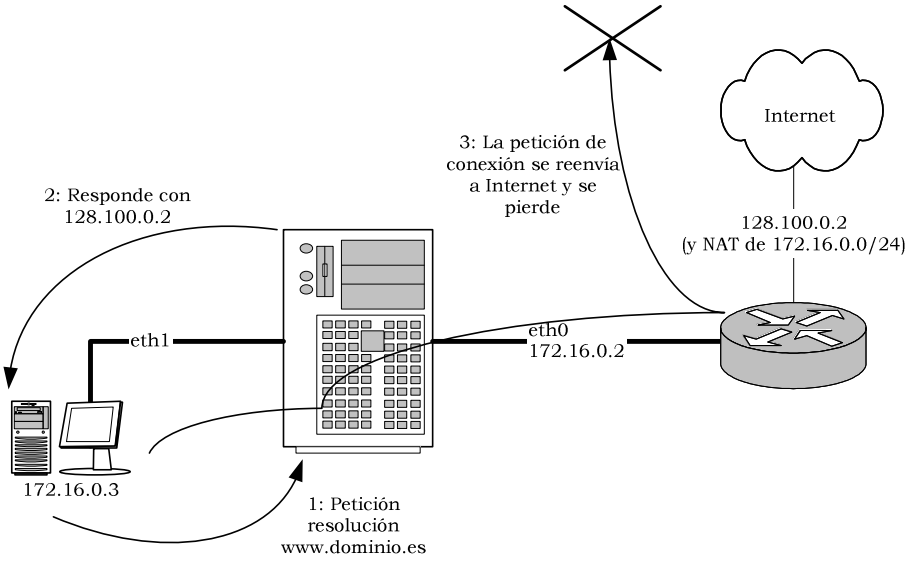


Ilustración 6-12: Secuencia ejemplo de un acceso Web a la IP en la que se realiza NAT

Esto dará lugar a que si el host 172.16.0.3 desea acceder a una Web configurada dentro de la IP 128.100.0.2, y que realmente estará alojada dentro de 172.16.0.2, lo que hemos de lograr es que la DNS responda de manera diferente (con la IP privada) o bien directamente hemos de configurar la IP pública en el servidor (lo que tampoco supondría ningún problema, ya que todo el tráfico le llega realmente al servidor hacia la IP 172.16.0.2).

Pese a que la opción de configurar la IP en el servidor parece más óptima nos decantaremos por configurar el DNS para que mediante vistas responda de manera distinta en un lado y en otro, informando a la red 172.16.0.0/24 de que cualquier resolución que afecte a un dominio que esté en la IP pública mapeada debe resolverse con 172.16.0.2.

Si hubiéramos optado por la otra solución habría hecho falta sólo a nivel de Web que por cada sitio Web se crearan en la configuración dos sitios virtuales diferentes (o *VirtualHost*, que es como en Apache se conocen a las entradas que van a ser mostradas a través de la misma IP), lo que claramente supone un engorro aún mayor.

Para que la respuesta sea distinta, la única zona cuya vista variará será la principal del ISP (su propio dominio), en la que existirá una entrada que será usada como CNAME por todos los restantes dominios. Ese CNAME resuelto será el que será 172.16.0.2 para aquellos hosts cuya IP de origen sea 172.16.0.0/24, y 128.100.0.2 para los restantes, si seguimos con la numeración hasta ahora empleada.

6.3 Diseño del panel de control

6.3.1 Gestión de los servicios

Para los servicios será necesario que los técnicos sean capaces de crearlos con las mismas características y detalles que si lo hubieran hecho a partir de la Web.

Una opción podría ser que los administradores los crearan directamente en la Web en una zona especial de administración donde pudieran ver la totalidad de las configuraciones. Esta opción será descartada por dos razones: la primera que los administradores no necesitan una interfaz tan cuidada y rica en filtrados como los clientes, pudiendo perfectamente realizar su actividad a través de la consola de comandos. La segunda razón está relacionada con el entorno en el que operarán: si lo hacen desde la Web cada una de las tareas que puedan realizar deberán estar convenientemente implementadas en un panel de control especial que haríamos desde cero (lo que supondría más trabajo) y además no estaría disponible la plena funcionalidad que podemos considerar tendrá que esos administradores trabajen desde consola de comandos, donde en un momento dado pueden ejecutar cualquier comando disponible en el sistema operativo.

La intención es que los administradores usen por tanto en su trabajo diario de monitorización, control y modificaciones sobre los servicios la consola de comandos, bien a través de la red local de trabajo, bien desde una localización remota, usando siempre un protocolo seguro de comunicaciones como es el SSH. La limitación que este modo de trabajo tiene es la incapacidad de poder ver gráficas de estado y similares, que si deberán ser colocadas en un sitio Web de acceso convenientemente restringido.

La totalidad de acciones que los clientes van a poder realizar desde la Web serán procesos que una vez convenientemente acotados serán diseñados como comandos dentro de los servidores. Es tarea de estos comandos comunicar al servidor de backup la creación de un nuevo servicio de DNS, o la necesidad de autorizar el *relay* de correo para un dominio. Estos comandos serán orientados a argumento, con lo que la totalidad del proceso necesario para crear un servicio o cuenta consistirá en una línea de argumentos de cuyo valor devuelto por la ejecución del mismo se deducirá si se ha creado, modificado o borrado correctamente (valor de retorno es cero) o si hubo algún error (valor distinto de cero, sin importarnos cuál ha sido el valor devuelto).

Como parte de esos procesos necesitarán mostrar el resultado del mismo más allá de conocer si ha habido o no errores, será necesario además que el comando devuelva un texto convenientemente formateado, siguiendo un formato que cumplirá la siguiente sintaxis:

- Toda la información que los comandos devolverán será texto ASCII estándar, sin acentos ni caracteres no imprimibles.
- El texto devuelto estará siempre compuesto por líneas en las que no existirán el carácter dos puntos excepto como separador entre el nombre de la variable o información mostrada y su valor. Si hay múltiples valores el separador a usar será la coma. Si alguno de estos caracteres se ha de utilizar, se tendrá que escapar.

  Por ejemplo, si estamos mostrando la lista de direcciones de correo de un usuario, sería este el formato a utilizar:

  email: **uno@dos.es**, **tres@cuatro.es**
- Los comandos dispondrán de una ayuda disponible a través de la opción –h. Todos los comandos habrán de soportar tanto argumentos cortos como argumentos largos.
- Todos los valores necesarios para completar el comando podrán ser suministrados mediante línea de argumentos. El comando además puede estar facultado para leer estos valores directamente desde la consola de comandos (que será como actúe el administrador). Para que se dé la circunstancia de que se lea desde la entrada estándar en lugar de como opción del comando, no deberá ser introducido ningún argumento.

Esta lista de requisitos sobre los comandos se plantea para que una vez diseñados los comandos no tengan que volverse a implementar de nuevo para realizar la segunda parte, que sería la interfaz con el cliente a través de la Web. Al mismo tiempo, esta lista de requisitos obligará a que todos los comandos sean implementados dentro del mismo lenguaje y utilicen al menos una librería que sea la interfaz que todos han de cumplir, llamando todos a las mismas funciones para realizar tareas como imprimir por pantalla un resultado de la acción, leer un valor de la línea de argumentos o similares. El lenguaje a utilizar será *Python*, por su versatilidad en el tratamiento de cadenas (que será usado de manera reiterada), y por poner a nuestra disposición la mayoría de funciones que en Linux podemos encontrar.

La interfaz Web que el cliente podrá ver sólo alcanzará a crear servicios vinculados al identificador del cliente, y sólo deberá poder ver asociados a ese cliente.

La interfaz Web será programada en PHP y necesitará de una base de datos, que será *MySQL*, dada la simplicidad de los datos que contendrá (las vinculaciones entre los servicios y los clientes). De hecho, también la pasarela con facturación será implementada usando PHP y *MySQL* por las mismas razones.

La interfaz Web programada en PHP ejecutarán los mismos comandos físicos que utilizan los administradores mediante un sistema de SUDO o similar, autorizándose únicamente la ejecución de estos comandos a los administradores por un lado y al usuario con el que se

ejecute el panel de control vía Web, usuario que será único y distinto al de cualquier otra Web alojada. Como la llamada a este comando no estará verificada para comprobar que el cliente que la ejecuta está realmente autorizado para crear ese servicio, será responsabilidad de la interfaz Web filtrar convenientemente la creación de servicios, comprobando que el cliente no ha alcanzado aún sus límites.

Al nivel de entidades de diseño, la programación necesaria implica únicamente tres entidades relevantes, a saber:

- Buzón de correo electrónico
- Sitio Web
- Zona DNS

Nosotros necesitaremos una identidad debido a que cada uno de estos servicios se configurará para ser ofrecidos a clientes, y esta entidad será justamente el cliente.

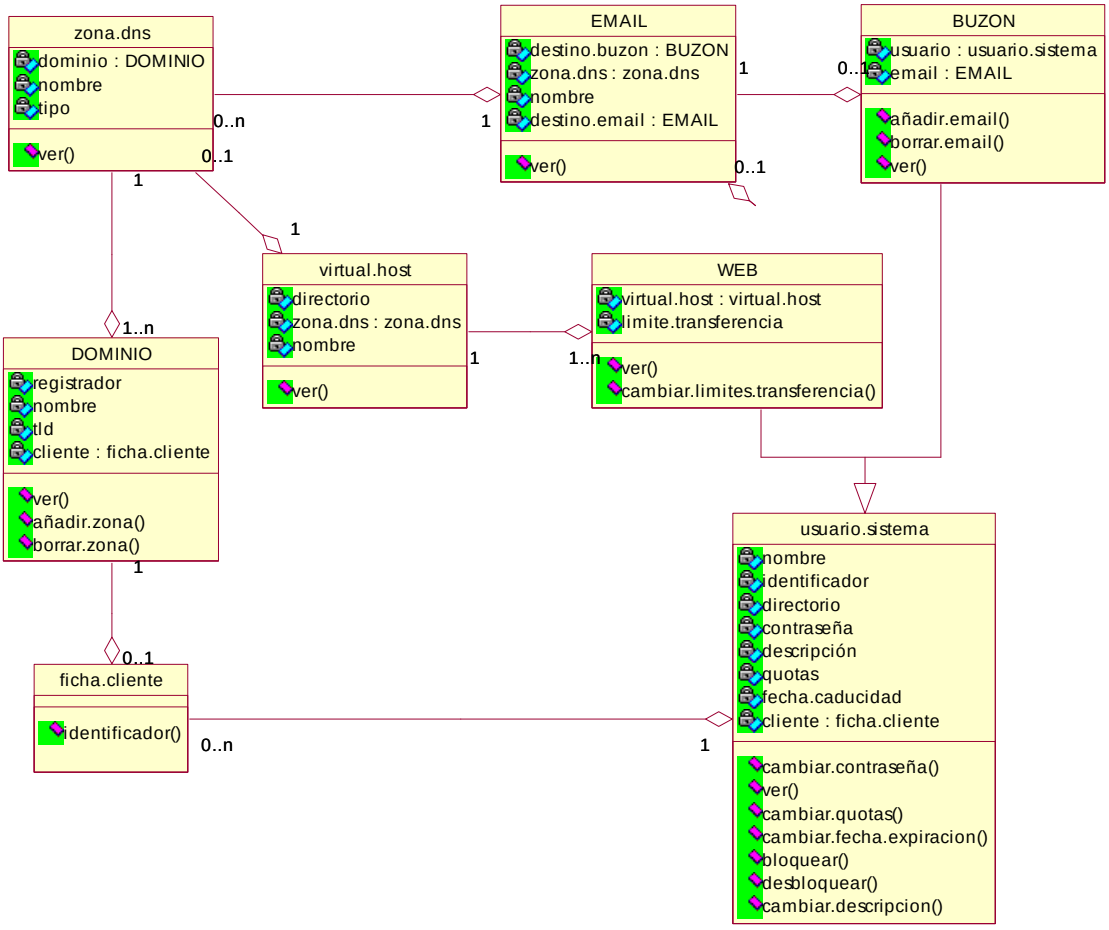


Ilustración 6-13: Diagrama de diseño de contexto

Estas identidades se pueden disgregar a un nivel inferior: por ejemplo el buzón de correo electrónico dispondrá de una o varias direcciones de correo electrónico, porque sobre un mismo usuario se pueden definir más de un correo. De esta forma extraemos dos entidades: el usuario del sistema Linux y la dirección de correo

electrónico, cuya cardinalidad y relación entre sí se muestra en la Ilustración 6-13.

La lista de comandos necesarios para gestionar todas estas entidades que hemos ido obteniendo sería:

- Servicio de correo
  - Añadir/borrar/modificar cuentas de correo (descripción, nombre del usuario, límites de tamaño de los buzones, etc.)
  - Modificar la contraseña del usuario.
  - Añadir/borrar/modificar direcciones de correo a las cuentas existentes.
- Servicio de Web
  - Añadir/borrar/modificar cuentas de Web (descripción, nombre del usuario, límites de tamaño y de descarga, etc.)
  - Modificar la contraseña del usuario.
  - Añadir/borrar/modificar los dominios que este usuario aloja.
- Servicio de DNS: no accesible para el usuario. Estos comandos serán ejecutados desde el servicio de correo o desde el servicio Web, y consistirán en crear una zona DNS nueva, borrarla, o modificarla.´

6.3.2 Gestión de usuarios

Tanto la gestión de los usuarios como la interfaz con los sistemas de facturación no requieren la dualidad de modos de trabajo que se requiere de la gestión de los servicios. Para la gestión de usuarios se requiere únicamente una interfaz Web que verifique a nivel básico los datos introducidos por el usuario (nombre, dirección, correo electrónico, etc.). Dado que la modificación de estos datos puede ser necesaria por parte del propio ISP será necesario que existan usuarios con acceso de modificación sobre la totalidad de la información de los usuarios.

Esta información del cliente será independiente de los servicios en vigor, con lo que alterar el nombre del cliente no puede afectar a sus cuentas en servicio. El acceso a esta zona de la Web se hará mediante un usuario y una contraseña que generará el sistema. La información invariante en cualquier caso será un identificador de cliente que habrá generado el sistema de facturación.

Una vez disponga de este usuario y contraseña vinculado a su identificador de cliente, desde este panel de control podrá visualizar todos los servicios en vigor, que serán aquellos que estén asociados en el sistema a este identificador de cliente.

Entre la información que el propio cliente no podrá modificar se encontrarán limitadores del total de servicios que puede activar, límites modificables en función de los productos que vaya contratando.

La interfaz será programada en PHP y necesitará de una base de datos, que será *MySQL*.

6.3.3 Gestión de dominios

Todos los servicios de alojamiento y correos se ofrecerán partiendo de la base de que los clientes además desean utilizar dominios para alojar estos servicios. Tendremos por tanto que considerar también estos costes y además la forma de gestionar estos dominios.

Dado que vamos a manejar un volumen alto de dominios, no interesa que este proceso sea manual y requiera cada alta intervención humana para registrar el dominio, y por tanto habremos de acudir a algún registrador que ofrezca al tiempo descuentos por volumen y aplicaciones que permitan registrar los dominios automáticamente, aplicación que además deberá ejecutarse en la plataforma que tenemos previsto utilizar, *Linux*. Aunque se pueden obtener mejores precios acudiendo a otros registradores, estos no ofrecen ni paneles de control ni una garantía en cuanto a la atención telefónica, cosa que nos interesará ya que en temas de traslado de dominios de un registrador a otro suelen aparecer bastantes problemas de difícil solución.

La lista de registradores puede obtenerse directamente desde la Web en **http://www.icann.org/registrars/accredited-list.html** para los

TLDs genéricos. Los TLD regionales requerirán de intervención humana para aquellos para los que el registrador que escojamos no posea capacidad de registro, como a buen seguro será el caso del ".es" español, que luego trataremos con detalle, ya que hablaremos primero de los TLD genéricos.

Obtener un buen tiempo de respuesta, y la posibilidad de trabajar de manera automática en el registro se cumplen con registradores que ofrecen precios por dominio que rondan los 10 euros para los TLD genéricos, sujeto al pago por adelantado de bloques de dominios que posteriormente serán usados, por lo que realmente el ahorro en el precio se ve penalizado con un adelanto económico al registrador. En concreto nos hemos fijado en *Bulk Register* (**www.bulkregister.com**) y en *eNom* (**www.enom.com**), aunque este segundo resultaría bastante más caro de lo que nos hemos presupuesto. Aunque siempre cabe la posibilidad de negociar directamente con un registrador, esto no parece lógico hasta que lleguemos a un punto en que nuestro TLD posea una base amplia de dominios con los que ejercer presión sobre los precios, por lo que de momento parece oportuno aprovechar las ofertas disponibles en la red para registros al por mayor.

Capítulo aparte son los TLD regionales, que como ya hemos comentado no serán ofrecidos directamente por un mismo registrador, además de ser diferente tanto los criterios de admisión de un registro como el procedimiento a seguir. En realidad nos concentraremos en los TLD genéricos y sólo ofreceremos la posibilidad de registrar bajo ".es", pero esta segunda opción conllevará un *sobrecoste* muy alto, dado que de momento todo el proceso necesario para registrar un dominio bajo este TLD está sujeto al intercambio de correos con *Red.es* siguiendo un protocolo cerrado que requerirá siempre intervención y control humanos.

Todos los gastos relativos a los dominios van a ser anuales, ya que pagaremos siempre por periodos de un año a menos que el cliente firme y pague por adelantado contratos de mayor periodicidad. Como esto raramente se va a producir, es preferible asumir un coste de 10 euros por dominio que pagar 40 por cinco años y luego perderlos al marcharse el cliente. Otra amenaza al respecto será la evolución de la paridad euro/dólar, ya que escojamos el registrador que escojamos tanto el registrador como nosotros nos veremos obligados a trabajar en dólares y no en euros, debido a que las tasas que todos los registradores pagan a su vez por registros también se realizan en esa moneda (en realidad, habremos de asumir que el dólar es la moneda de pago en cualquier transacción que llevemos a cabo en Internet).

La programación relacionada con la gestión de dominios se realizará en *Python*, al estar ya en ese lenguaje la mayor parte del desarrollo de la gestión de los servicios del ISP, y por tanto resultar más sencillo para el equipo de programación utilizar este lenguaje. La implantación partirá de un estándar que vendrá definido por el registrador de dominios, consistente en la especificación de qué protocolo y a través de qué mecanismos se comunicarán las altas y bajas de dominios. Lo habitual

es utilizar unas librerías que facilitan los propios registradores, librerías que disponen de comandos para crear los dominios sin necesidad de programar código que realice la conexión remota a los sistemas del registrador. Dispondremos pues de una de estas librerías y lo que en *Python* desarrollaremos será únicamente una interfaz que verifique antes de registrar que no está previamente registrado por nosotros y cree los registros DNS tanto en el servidor principal como en el servidor de backup.

6.3.4 Integración de la facturación

Tanto la gestión de los usuarios como la interfaz con los sistemas de facturación no requieren la dualidad de modos de trabajo que se requiere de la gestión de los servicios.

Si la información sobre los usuarios y los servicios ofrecidos a estos no está integrada con la información de los clientes y los servicios a estos facturados, ambos sistemas presentarán incongruencias (clientes con servicios por los que no pagan o servicios facturados y no servidos)

Los servicios que se facturen a los clientes deberán guardar relación con la información disponible en los servidores que ofrecen esos servicios. La solución que adoptaremos será definir quién creará los identificadores que en ambos sistemas (facturación y los servidores del ISP) vayan a identificar de manera única al cliente y sus servicios. Por versatilidad, los identificadores de cliente serán generados por el sistema de facturación (es decir, que cuando tengamos un cliente nuevo la primera acción deberá ser obtener de la facturación un identificador para ese cliente), identificador que será usado como parte de la descripción sobre cada uno de los servicios que se vayan creando. Los identificadores de estos servicios creados serán generados por los sistemas del ISP y suministrados posteriormente al sistema de facturación.

El sistema será congruente mientras todas las acciones en ambos casos puedan ser simultaneadas en todo momento, lo que nos va a exigir que la facturación resida también en nuestro propio servidor del ISP. Esto supone un riesgo mayor por lo que respecta a la seguridad tanto en que los datos estén siempre a buen recaudo como en que los datos de facturación no vayan a ser perdidos, pero supone la mejor alternativa posible a disponer de otro sistema para la facturación.

Para disminuir parte de este riesgo se copiarán de manera diaria los datos de facturación sobre el servidor de backup a través de Internet, y se habilitarán controles que permitan además verificar la congruencia de los datos de facturación y los datos de usuarios y servicios activos, para detectar aquellas operaciones que pueden haber dejado incompleto un borrado (dejando el usuario activo mientras en facturación está eliminado) bien una creación (el servicio es facturado pero no existe en los servidores).

7 Implementación

7.1 Implementación de la Red

7.1.1 Conectividad a Internet

Por lo que respecta a la conectividad a Internet, se ha optado por integrar en el proyecto la configuración de cada uno de los aparatos.

Del hardware presente en la red, los dos routers es previsible que sean suministrados por la operadora y no sea por tanto necesaria su configuración, pero en previsión de que esto no sea así, se ha optado por mostrar la configuración necesaria en dos routers Cisco 2611. Como ya se comentó en el diseño, escoger una única marca para la red reducirá los costes de mantenimiento posteriores.

De las tres áreas de captación posibles, la correspondiente a los servidores no será de momento tenida en cuenta ya que nuestro único servidor no estará en ella, al ser su función además la de router de dicha red: se reservará esta red para el futuro y por tanto no será necesaria de momento su configuración.

La configuración del único switch es sencilla: dispondrá de una IP de gestión que haremos sea del rango privado 172.16.0.0/24 (en concreto usaremos 172.16.0.254), permitiendo el acceso SNMP que usaremos para realizar monitorización del tráfico.

Las configuraciones mostradas han sido generadas para versiones de IOS mayores de la 11.1 (IOS es el nombre del sistema operativo que montan todos los componentes de red diseñados por Cisco). Se ha activado además de manera adicional listas de acceso que filtren parte del tráfico, con independencia de que luego el servidor principal haga de nuevo de *firewall* principal.

Espacio de direccionamiento

172.16.0.0/24: Red privada de trabajo

254 hosts posibles, realizándose además resolución inversa sobre este rango en nuestro servidor DNS. El espacio de direccionamiento se considera suficiente para cualquier crecimiento futuro.

La asignación inicial de IP será la siguiente:

- 172.16.0.1: Router responsable de la conexión con IP 128.100.0.2, y que hará NAT de dicha IP desde y hacia nuestra red privada.

- 172.16.0.2: Servidor principal de la empresa.
- 172.16.0.3: IP para el servidor de backup, asignada mediante un tunelado que se realizará en el servidor principal.
- 172.16.0.17–172.16.0.253: Asignación dinámica desde el servidor principal mediante DHCP. Dejamos libre el rango 172.16.0.4–172.16.0.16.
- 172.16.0.254: IP de administración del switch *Catalyst*.

128.100.0.2/32: Red pública LMDS

IP única obtenida del operador que nos suministra la conectividad mediante LMDS. Imposible el crecimiento por las características del acceso contratado (no orientado a su uso para dar servicios, la IP carecerá además de resolución inversa).

De esta IP se realiza NAPT en su router, redirigiéndose todos los puertos de servicios conocidos (Web, DNS...) hacia el servidor principal, para que éste pueda ofrecer servicios en Internet desde dicha IP, al mismo tiempo que los equipos de la red local pueden acceder a Internet a través de este router.

128.200.0.0/28: Red pública *frame-relay*

Subred de 8 IPs reservadas para el ISP por el operador que nos suministre la conectividad mediante *frame-relay*. Las IPs se obtendrán del proveedor, el cual mediará para ello ante el RIPE, organismo europeo encargado de la asignación de IPs en Europa dentro del rango CIDR, único disponible en la actualidad.

Nuestro ISP no podrá obtener directamente las IPs del RIPE, ni tampoco utilizar las que actualmente tenga contratado con otro operador, por razones de funcionamiento del enrutamiento dentro del rango CIDR: el espacio de direccionamiento mínimo a solicitar (y justificar) ante el RIPE de manera directa es una red /22 (de 1024 hosts, por tanto), que supera ampliamente nuestras actuales posibilidades. Para aquellas necesidades de direccionamiento de menor cuantía hay que obtenerlas directamente del espacio de direccionamiento que cada operador tiene asignado, aún debiendo ser justificadas de manera debida.

Por tanto, del proveedor lograremos la red 128.200.0.0/28 para nuestro ISP: la situación es totalmente simulada, ya que de entrada este rango está asignado por el ARIN (el organismo norteamericano), y ya se encuentra en uso, tal y como ocurre con el 128.100.0.0/16.

Suponiendo que este rango nos fuera asignado, el reparto que de él haríamos sería el siguiente:

- 128.200.0.1: Router.
- 128.200.0.2: Servidor principal.

El resto de IPs (128.200.0.3–128.200.0.7) permanecerán inutilizadas mientras no nos haga falta otro servidor, pero optaremos por pagarlas para tenerlas en reserva.

De este rango se obtendrá del operador la resolución inversa delegada en nuestro servidor principal y en el de backup.

128.300.0.1/32: Servidor de backup

Por último, nuestro servidor de backup alojado en otro operador tendrá que disponer de una IP. De esta IP también se realizará la resolución inversa.

Tanto en este caso como en el anterior, el lograr la resolución inversa podrá suponernos un pequeño inconveniente, al ser direcciones que por sus características corresponden a redes de clase C: por diseño el protocolo no permite la delegación de una red distinta a la /24, por lo que los operadores se verán obligados a delegarnos una a una las IPs que forman la red, y nosotros a disponer de un fichero de zona de resolución inversa para cada una de estas IPs delegadas.

Configuración de los routers

Router del operador que nos delega 128.100.100.0/28

El primer router a configurar será el que permitirá acceder a Internet a los equipos de la red de trabajo, mediante NAT y usando su única IP pública. Como además queremos ofrecer servicios hacia Internet, deberemos configurar también NAPT para que los puertos que nos interesen sean mapeados de manera estática hacia el servidor principal.

De la configuración completa del router, nos quedaremos con las partes más interesantes: en primer lugar la configuración del NAPT, y luego la definición del *firewall* básico que protegerá, junto al del servidor principal, tanto a este servidor como a la red local.

De las dos posibles interfaces de red local del router Cisco 2611 usaremos Ethernet0, y luego Serial0 para la conexión LMDS. Mostraremos la configuración fuera del contexto, explicando únicamente los pasos para configurar las interfaces, y siempre partiendo de que nos encontramos en el router en el modo de configuración global:

```
interface Ethernet 0/0
 no shutdown
 description red.local
 ip address 172.16.0.1 255.255.255.0

! por cada puerto que se desee mapear hacia el servidor principal:
! (en este ejemplo el 25, del SMTP, faltarían al menos
!  el 110, 80, 53, todos TCP)
 ip nat inside source static tcp 172.16.0.2 25 128.100.0.2 2

 ip inspect Ethernet_0_0 in
 ip access-group 100 in
```

```
 keepalive 10
nat inside source list 1 interface Serial 0/0.1 overload
```

Pese a que ya hemos remarcado que habrá un *firewall* en el servidor principal, *firewall* que también afectará a este tráfico, habilitaremos de manera extra aquí la inspección del tráfico que Cisco puede realizar y listas de acceso que impidan además la mayor parte del tráfico, exceptuando el acceso legítimo de los usuarios a Internet y los puertos mapeados hacia el servidor.

```
ip inspect tcp synwait-time 30
ip inspect tcp finwait-time 5
ip inspect tcp idle-time 3600
ip inspect udp idle-time 30
ip inspect dns-timeout 5
ip inspect one-minute low 900
ip inspect one-minute high 1100
ip inspect max-incomplete low 900
ip inspect max-incomplete high 1100
ip inspect tcp max-incomplete host 50 block-time 0
ip inspect name Ethernet_0_0 ftp
ip inspect name Ethernet_0_0 smtp
ip inspect name Ethernet_0_0 tcp
ip inspect name Ethernet_0_0 udp
no access-list 1
access-list 1 permit 172.16.0.0 0.0.0.255
no access-list 100
access-list 100 permit udp any eq rip any eq rip
access-list 100 permit tcp host 172.16.0.2 any range 20 21
access-list 100 permit tcp host 172.16.0.2 any eq 80
access-list 100 permit icmp host 172.16.0.2 any
access-list 100 permit tcp host 172.16.0.2 any eq 25
access-list 100 permit tcp host 172.16.0.2 any eq 22
access-list 100 permit udp host 172.16.0.2 any eq domain
access-list 100 deny ip host 172.16.0.2 any
access-list 100 deny ip host 172.16.0.3 any
access-list 100 permit udp any any eq domain
access-list 100 permit tcp any any range 20 21
access-list 100 permit tcp any any eq 80
access-list 100 permit icmp any any
access-list 100 permit tcp any any eq 119
access-list 100 permit tcp any any eq 25
access-list 100 permit tcp any any eq 22
access-list 100 permit tcp any any eq 23
```

Aunque ya hemos definido las reglas para activar NAPT, nos quedaría la configuración para habilitar NAT dinámico y que permita acceder a Internet al resto de equipos:

```
ip nat translation timeout 86400
ip nat translation tcp-timeout 86400
ip nat translation udp-timeout 300
ip nat translation dns-timeout 60
ip nat translation finrst-timeout 60
```

Sobre la conexión Serial0 se ha supuesto que el DLCI era 123, desconociéndose si la conexión LMDS tendrá esta característica o no. Si llegado el caso no fuera así o la señal fuera entregada de otra forma habría que reemplazar la tarjeta de expansión por otra adecuada.

Incluso podría darse el caso de que el operador suministrara directamente la señal en *Ethernet*, con lo que nos bastaría con usar las dos interfaces *Ethernet* que posee este modelo de router, ya que de momento Serial1 permanece sin uso alguno:

```
interface Serial 0/0.1 point-to-point
 no shutdown
 description connected to Internet
 ip address 128.100.0.2 255.255.255.252
 ip nat outside
 ip access-group 101 in
 frame-relay interface-dlci 123
```

Router del operador que nos delega 128.200.0.0/28

Por otro lado tendremos la configuración para un router Cisco 2611 que se encargará de la conexión *frame-relay*. De este router no mostraremos apenas datos sobre su configuración, ya que es básicamente la misma que para el anterior pero eliminando la configuración relativa al NAPT y NAT, aquí innecesario. Se implementa igualmente un *firewall* sobre esta conexión:

```
access-list 101 deny ip 128.200.0.0 0.0.0.15 any
access-list 101 permit udp any 128.200.0.0 0.0.0.15 eq domain
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 range 20 21
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 80
access-list 101 permit icmp any 128.200.0.0 0.0.0.15
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 143
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 220
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 25
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 110
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 3306
access-list 101 permit tcp any 128.200.0.0 0.0.0.15 eq 22
```

En este caso no procede filtrar demasiado tráfico sobre el servidor: únicamente sobre el tráfico entrante vigilaremos que no se salga del os protocolos habituales (en concreto en este listado se ha autorizado el tráfico Web, FTP, POP3, SMTP, IMAP, DNS y que permite conectar al *MySQL* de manera remota, aunque luego podría ser restringido de manera específica, esta tarea se realizará ya en el servidor principal mediante las herramientas que para ello dispone Linux). Faltaría por otro lado asignar la lista anterior sobre la interfaz Serial0 en dirección entrante, para completar la configuración del *firewall*.

Switch *Catalyst* 2912XL

En el switch se han creado dos redes virtuales de momento, a falta en un futuro de que sea necesario crear la red de servidores. La administración de la VLAN es estática por asignación directa de los puertos a una u otra VLAN, de forma que los primeros cinco puertos se han asignado a la VLAN por defecto, y luego cuatro a las dos VLAN creadas, *acceso_internet* y *red_local*. De momento esta configuración podrá cubrir hasta doce hosts conectados, más que suficiente para nuestra configuración inicial.

Dado que las VLAN requieren una división de los puertos para que estos sean asignados a una u otra red virtual, y como de momento sólo vamos a necesitar 2 VLAN, la política de conexión a los puertos del switch se basará en ir asignándolos a cada VLAN desde los extremos opuestos: de esta forma si el switch *Catalyst* 1924XL tiene 24 puertos dispuestos en 2 líneas o *slots* (la inferior se nombra como 0 y la superior como 1), quedando en cada *slot* 12 puertos, la VLAN1 asignada a la red de trabajo comenzará en el *slot* 0 hacia la derecha y la VLAN2 de la red de interconexión en el *slot* 1 hacia la izquierda, tal y como muestra la Ilustración 7-1.

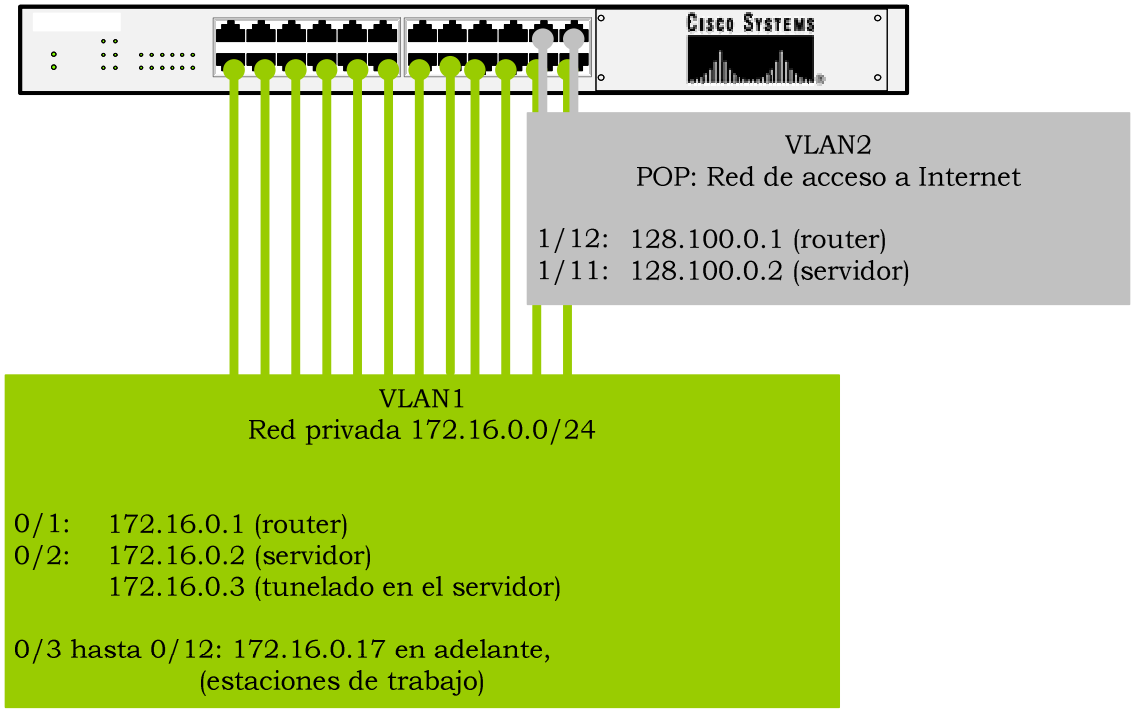


Ilustración 7-1: Asignación de puertos en el switch

Primero crearíamos las VLAN dándoles un número identificador (este modelo concreto soporta hasta 1024 VLAN, aunque dudamos que vayan a ser necesarias tantas:

```
# vlan database
(vlan)# vlan 10 name red.local
(vlan)# vlan 20 name red.acceso
```

Y ahora, en cada puerto del switch bastaría con especificarle a qué red virtual pertenece (este en concreto es el puerto del router 172.16.0.1):

```
interface Fast Ethernet 0/0
 no shutdown
 switchport mode access
 switchport access vlan 10
```

Hay que tener en cuenta que pese a no ser un router, el switch dispone de la misma CLI que un router, y ejecuta igualmente IOS, por lo que la configuración es similar en cuanto a proceso y comandos a la de un router. Veamos ahora la configuración global, en la que hemos

habilitado SNMP y el algoritmo *Spanning Tree Algorithm* para evitar *loops*.

```
set password hector
set enablepass password
set prompt Switch1
set length 24 default
set logout 20
!
#system
set system baud  9600
set system modem disable
set system name  BDS-1-5500
set system location ISP_MDF1
set system contact  Hector Castillo
!
#snmp
set snmp community read-only      public
set snmp community read-write     password
set snmp community read-write-all password
set snmp rmon enable
set snmp trap enable  module
set snmp trap enable  chassis
set snmp trap enable  bridge
set snmp trap enable  repeater
set snmp trap enable  vtp
set snmp trap enable  auth
set snmp trap 10.0.0.10   bdsltd
!
#ip
set interface sc0 2 172.16.0.254 255.255.255.0 172.16.0.255

set interface sl0 0.0.0.0 0.0.0.0
set arp agingtime 1200
set ip redirect   enable
set ip unreachable   enable
set ip fragmentation enable
set ip route 0.0.0.0         172.16.0.1     1
set ip alias default         0.0.0.0
!
set span 2/2  2/12 both
set span disable
!
set spantree enable         1
set spantree fwddelay 15    1
set spantree hello    2     1
set spantree maxage   20    1
set spantree priority 32768 1
set spantree portcost 2/11-12 100
set spantree portpri  2/11-12 32
set spantree portfast 2/11-12 disable
set spantree portcost 3/11-12 100
set spantree portpri  3/11-12 32
set spantree portfast 3/11-12 disable
set spantree portcost 4/11-12 100
set spantree portpri  4/11-12 32
set spantree portfast 4/11-12 disable
set spantree enable         2
set spantree fwddelay 15    2
set spantree hello    2     2
set spantree maxage   20    2
```

```
set spantree priority 1     2
set spantree portcost 2/1-2 100
set spantree portpri  2/1-2 32
set spantree portfast 2/1-2 disable
set spantree portcost 3/1-2 100
set spantree portpri  3/1-2 32
set spantree portfast 3/1-2 disable
set spantree portcost 4/1-2 100
set spantree portpri  4/1-2 32
set spantree portfast 4/1-2 disable
```

Las garantías contractuales del ancho de banda

Cuando se requiere contratar caudal de acceso a Internet las ofertas con las que nos encontraremos harán una distinción entre caudal CIR y caudal PIR, además de otros aspectos a considerar, como compensaciones por ausencia de calidad o servicio, etc. Veremos ahora por qué tipo de conexiones optaremos.

El PIR es el acrónimo de *Peak Information Rate* y es el máximo que puede alcanzarse de manera nominal con la conexión contratada, mientras que el CIR significa *Committed Information Rate*, y representa el ancho de banda al que se compromete el proveedor del servicio. Hay bastante diferencia pues entre ambos términos, y siempre habremos de buscar la mejor relación posible (incluso 1:1) entre ambos valores, el de CIR y el de PIR, pero puede no interesarnos del todo.

Hasta ahora hemos desgranado unos anchos de banda por cada sitio Web y por cada cuenta de correo, y hemos comentado la necesidad de que aparte de ese ancho de banda, se contratara de manera adicional como margen de seguridad. En relación al Web consideraremos necesario que las cifras de ancho de banda por Web supuestas se ofrezcan con caudal CIR, mejorándose de manera adicional en condiciones normales el rendimiento de la Web si además existe un valor de PIR mayor que el CIR.

Pero respecto al correo en el apartado anterior ya hemos distinguido entre el tráfico que nuestro ordenador hacía con el cliente (que también ofreceremos con ancho de banda CIR) y el tráfico con otros intercambiadores de correo de Internet, que por ser menos estratégico (no es visible al cliente final), vamos a ofrecer mediante caudal PIR en su totalidad, dado que el precio de este caudal contratado es bastante menor (fundamentalmente porque no hay garantías de que vaya a estar disponible, aunque sea así en el 99% de los casos).

SLA o garantía de nivel de servicio

Adicionalmente existen unas garantías de nivel de servicio que muchos proveedores pueden fijar por contrato y que obligan a éste a cumplir una serie de parámetros en la conexión de manera continua, reduciéndose la factura mensual si estos parámetros caen por debajo de los umbrales fijados en el contrato. Estas garantías se denominan SLA, acrónimo de *Service Level Agreement*, y aunque su objetivo es muy loable (miden parámetros objetivos con los que regir las posibles

disputas entre proveedor y el ISP cliente ante la calidad del servicio ofrecido), su problema reside en que al figurar en contrato, cualquier discrepancia o exigencia de resolución puede acabar llevándonos a los tribunales.

Además, el que la empresa reduzca la factura ese mes por habernos dejado sin servicio más allá de cuatro horas nos es indiferente, cuando nuestro ISP debe estar 24h en servicio para que no pierda clientes. Por todo ello y aunque analizaremos algunos de los parámetros que la SLA puede y debe reflejar, su presencia en nuestro contrato no tendrá como objetivo mas que facilitar una posible ruptura justificada de nuestro contrato en caso de falta o deficiencia de servicio, más que de obtener una satisfacción económica tras un corte en la conexión. Además de que estas garantías necesitarían de una tercera parte no implicada (que no fuera el cliente o el proveedor) para que fueran monitorizadas, cosa que difícilmente se va a permitir y que además supondría un coste extra.

Para solventar la falta de servicio procuraremos repartir la conexión entre diferentes operadores, de manera que al menos parte del servicio pueda continuar activa ante la caída de uno de ellos.

Algunas de las garantías que por contrato pueden y deben aparecer serán:

- Fecha de disposición en servicio propuesta (FDP): será la fecha que en el contrato firmaremos como fecha de puesta en servicio de la conexión, y cuyo incumplimiento por parte del proveedor suele acarrear un descuento en la cuota de instalación.
- Garantía de disponibilidad de la red: suele ser del 99.9% y puede excluir en muchos casos las interrupciones del servicio por tareas de mantenimiento. Habremos de evitar o negociar este tipo de cláusulas.
- Garantía de disponibilidad del circuito: referido al cableado, siempre suele ser del 99.9% y dado que no va a haber mantenimiento sobre el mismo a menos que ampliemos o reduzcamos la instalación, no parece un parámetro preocupante.
- Garantía de resolución de incidencias: habitualmente de 4h en la mayoría de proveedores, indica el tiempo máximo que puede estar abierta una incidencia en el servicio antes de que ésta sea resuelta.
- Latencia y pérdida de paquetes: algunos proveedores permiten por contrato establecer una latencia máxima en el tráfico

En realidad estamos hablando de garantías en un sector en el que se incumple diariamente todo por parte de todos: nadie disfruta de un ancho de banda del 100% durante el 99.9% del tiempo ni puede estar a salvo de interrupciones prolongadas, lo cual no deja de ser un

problema, ya que pese a que por contrato nunca haremos constar la disponibilidad plena con el cliente, la caída del servicio puede afectar a nuestra imagen y hacernos perder más de un cliente.

Podríamos evitar esta situación si utilizáramos una fórmula de *housing* con nuestros servidores, alojándolos en algún *data center* situado en Madrid o ya a nivel internacional, porque lo que si que es cierto es que los riesgos de caída del servicio se incrementan conforme nos alejamos de POP (puntos de presencia) de los grandes *carriers* de la red, dónde por haber gran cantidad de servidores y operadores, sale más rentable hacer cumplir estas garantías para el propio proveedor, además de ser mejores las instalaciones que los albergan. Aunque lo normal es disponer del servicio de Internet en la práctica totalidad de las ocasiones, no hay contrato que no incluya salvaguardas casi leoninas para el proveedor del servicio, incluso en estas fórmulas de *housing*.

Ofertas de ancho de banda

Los precios respecto al ancho de banda han sufrido una sustancial bajada en los últimos tiempos. Aunque este tema ya se ha tratado en la introducción, ahora para estipular un coste del ancho de banda recurriremos de nuevo a *Iber-X* (www.iber-x.com), de dónde se puede extraer esta gráfica que muestra la evolución del coste de un 1 *Mbps* de acceso a Internet desde el año 2.000 hasta nuestros días en dólares:

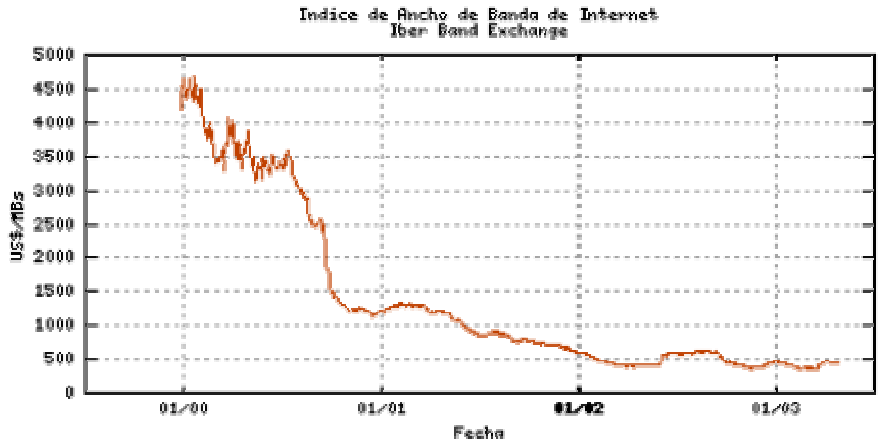


Ilustración 7-2: Coste del ancho de banda
Fuente: Iber-X
(http://www.iber-x.com/index.php?action=go&section=33)

Aunque aquí no se especifica el tipo de conexión, calidad o puntos de entrega de este tipo de conexiones, nos servirá como base para establecer un precio por *Mbps* de ancho de banda CIR y otro para el mismo producto con un índice 2:1 de PIR (es decir: una conexión en la que sólo el 50% del ancho se garantiza por contrato, siendo el restante 50% ancho de banda PIR).

En realidad, el precio varía bastante si la entrega del ancho de banda se ha de realizar en una capital como Madrid con respecto a una ciudad más pequeña y alejada, e incluso a nivel internacional, el coste

del ancho de banda en Madrid es un 33% más caro que en Londres, lo que además favorece que las grandes urbes acaparen cada vez más el alojamiento de servidores y los puntos de presencia en Internet (POP).

Nuestro ISP va a estar situado en una ciudad pequeña, y como tal tendrá bastante más limitada la oferta disponible, pero en líneas generales esperamos aspirar a obtener anchos de banda de ½ *Mbps* de CIR y 1 *Mbps* de PIR por 400 euros/mes. Aunque hasta el momento no lo hayamos indicado de manera explícita, estos caudales se componen de la misma cantidad de ancho de banda de subida que de bajada en canales independientes y no compartidos. Además supondremos difícil obtener caudales por debajo de esta cantidad, por lo que a la hora de tratar los costes siempre se trabajará con bloques múltiplos de ½ *Mbps* de CIR. Esta conectividad se presupone está ofrecida mediante *Frame Relay* o circuito *ATM* equivalente, e incluirá al menos una IP, que es lo todo lo que realmente esperamos necesitar (a menos que a su vez decidamos lanzarnos al *housing*). No aspiramos a necesitar en ningún momento un circuito T1 o STM-1, sino que aspiramos a un tamaño y volumen de negocio más bien modesto.

Hemos hablado de accesos alternativos que permitan lograr un poco de redundancia en el acceso a Internet ante la caída de uno de ellos: en realidad habremos de matizar esto, ya que lo realmente haremos será instalar una conexión de gran calidad (la explicada en el apartado anterior de tipo *ATM* o *Frame Relay*).

El resto de circuitos que contrataremos no contendrán una relación 1:2 entre CIR y PIR, y serán de calidad menor, por ejemplo mediante tecnología DSL o inalámbrica, considerada menos fiable pero de menor coste. En concreto la conectividad mediante LMDS (tecnología inalámbrica de acceso a la red operativa a nivel comercial en nuestro país mediante al menos un operador en cada demarcación) puede costarnos 300 euros por circuito de 4 *Mbps* con un CIR del 20% del mismo, mientras que mediante DSL el único producto disponible en España a nivel nacional sigue siendo el ADSL, con el cual estaríamos contratando anchos de banda de sólo 256/128 *Kbps* (su carácter asimétrico diferencia entre caudal de bajada –mayor– y de subida), pero con un CIR de sólo el 10% a 40 euros/mes.

No tendremos en cuenta ningún coste de instalación o despliegue porque la mayoría de proveedores ofrecen en la actualidad fórmulas para suplir estas cuotas de alta mediante contratos con vinculación de mantenimiento a largo plazo, de forma que la firma de una cláusula que asegure el sostenimiento del contrato más allá de dos años permite que sea el operador quien asuma ese coste inicial a cambio de una mayor seguridad en nuestra continuidad como clientes. Tampoco tendremos en cuenta el equipamiento para habilitar esta conexión, ya que el coste del mismo suele incluirse dentro del coste de instalación, que ya hemos explicado no va a darse.

Calidad del acceso y tecnología	Ancho de banda bajada/subida PIR y porcentaje CIR (en *Kbps*)		Coste mensual en euros
CIR 50% *Frame-Relay*	1024/1024	50%	400
CIR 12,5% LMDS	4096/4096	12,5%	300
CIR 10% ADSL	256/128	10%	40

Comenzaremos trabajando con una *frame-relay* y un circuito LMDS, descartando de momento usar ADSL. Usaremos la *frame-relay* para dar soporte al correo con los clientes, y la DNS y el correo, apoyándonos en el circuito LMDS para el resto de casos (correo con intercambiadores), y siendo mixto el tratamiento de los sitios Web (las grandes usarán el circuito LMDS y el de *frame-relay* de manera indistinta, suponiendo un resultado de un 50% de los accesos a uno y el 50% al otro, mediante el uso de ambas IPs en las respuestas DNS, mientras los sitios Web pequeños sólo requieren usar el circuito *frame-relay*).

7.1.2 Red local

El servidor, aparte de disponer de servicios, será a su vez router de acceso a la red local. Aunque se ha optado por dotar al equipo de varias interfaces de red esto no es necesario ya que podría hacerse con una única tarjeta.

Pero lo que realmente interesa es saber cómo nos permitiremos el lujo de lograr que sea nuestro servidor principal el que decida cómo realizar la conmutación de paquetes, existiendo dos posibilidades, el acceso repartido y el acceso balanceado. Aunque se optará por el primero de ellos, siempre cabe la posibilidad de aplicar el segundo.

Acceso repartido (*split access*)

Es la solución más simple: por cada conexión disponible tendremos una tabla de enrutamiento distinta, con su propio *gateway* configurado y todo el tráfico destinado/originado por la IP correspondiente se enruta en base a la tabla que le corresponde.

```
echo “201    Operador1” >> /etc/iproute2/rt_tables
echo “202    Operador2” >> /etc/iproute2/rt_tables


ip route add 128.100.0.1/24 dev eth1 src 1.2.3.4 table Operador1
ip route add default via 1.2.3.254 table Operador1

ip route add 5.6.7.8/24 dev eth1 src 5.6.7.8 table Operador2
ip route add default via 5.6.7.254 table Operador2
```

Por lo que respecta a la tabla principal, únicamente tendremos que asegurarnos de que cada red de nuestro operador sea accedida con la dirección de origen correspondiente, y para ello usaremos también una regla de enrutado.

```
ip route add 1.2.3.4/24 dev eth1 src 1.2.3.4
ip route add 5.6.7.8/24 dev eth1 src 5.6.7.8

ip rule add from 1.2.3.4 lookup table Operador1
ip rule add from 5.6.7.8 lookup table Operador2
```

Esta solución no permite lograr un balanceado del tráfico saliente generado en esta máquina, por lo que el tráfico de la red local saldría necesariamente siempre por uno de los operadores. Sólo sería útil si habiendo una página Web alojada en este servidor (o cualquier otro servicio), el DNS está configurado de manera que las respuestas que ofrece se distribuyen entre las dos IPs propuestas, lo que se lograría colocando la siguiente configuración el fichero de zona del dominio en cuestión si estamos usando *bind* para ello:

```
www     IN     A      1.2.3.4
www     IN     A      5.6.7.8
```

Esto lleva a *bind* a responder con toda la lista de IPs (porque puede haber más) asociadas a *www*, pero colocando cada vez una en primer lugar y que será la que utilice un cliente para acceder. Esta estrategia fue utilizada tiempo atrás en www.microsoft.com, aunque allí lo que había eran diferentes servidores la idea es la misma, pese a que al final la máquina sea la misma es ofrecida cada vez desde un operador distinto.

Otra ventaja de esta configuración es su mayor tolerancia: en caso de fallo de uno de los operadores el servicio se mantiene en el 50% de los casos en este ejemplo, ya que un 50% de las peticiones seguirán siendo ofrecidas desde la IP que continúa en servicio.

Hasta ahora el problema venía de que *bind* si había varias IPs asociadas al mismo nombre, no hacía realmente un reparto cíclico con el algoritmo *round-robin*, sino que daba una respuesta basada en la aleatoriedad. Desde la versión 9 esto no es así ya, y además podemos incluso definir qué porcentaje de las respuestas serán asociadas a cada uno de las IPs configuradas (en este caso desviaremos el 75% del tráfico por la IP 5.6.7.8):

```
www     IN     A      1.2.3.4      25
www     IN     A      5.6.7.8      75
```

Pero esta característica de *bind* no es en la realidad un balance de carga, ya que aunque estadísticamente en el tiempo si que debiera ser cierto que con un número de accesos muy altos el DNS si que respondiera en el 75% de los casos con una IP y en el 25% restante con la otra, la realidad no es esta debido a que los servidores DNS cachean nuestra respuesta.

Si esta respuesta cacheada (en la que por ejemplo se ha respondido con 1.2.3.4 en primer lugar) lo es por artemis.ttd.net, uno de los mayores servidores DNS más usados dentro de las redes de Telefónica de España, lo que tendremos es que todos los usuarios de la red de Telefónica (que son mayoría en este país) si acceden a esta Web lo

harán a través de la IP que supuestamente sólo debiera utilizarse en el 25% de los casos.

Esta será la estrategia que tomaremos en nuestro servidor principal (el servidor de backup dispone de una única conexión y por tanto no será necesario plantearse sistema similar.

Acceso balanceado (*load balancing*)

Para que el tráfico saliente pudiera ser balanceado de una manera equitativa, y no por servicios, como hasta ahora hemos planteado, hay que optar por rutas de múltiple camino (*multipath routes*), lo que se lograría añadiendo a la tabla de enrutamiento principal la siguiente orden:

```
ip route add default scope global \
            nexthop via 1.2.3.254 dev eth1 weight 1 \
            netxhop via 5.6.7.254 dev eth1 weight 1
```

De esta manera, y jugando con el peso (que en ambos coincide con 1) se podría también hacer que se usara de manera más intensa una determinada ruta, pero nos encontraríamos con el mismo problema que en el caso del balanceo del tráfico entrante usando el DNS, y que es el caché (para determinado destino las rutas son cacheadas y serían usadas de manera mayor), y sobretodo el hecho de que lo que se balancea es el uso de una ruta para acceder a determinado destino, no el tráfico desde allí o hacia allí enrutado, con lo que se daría el caso de que una determinada IP hacia la que se lleva/trae la mayor parte del tráfico siempre fuera enrutada a través del mismo acceso.

La solución implantada en el servidor principal es la primera, basada en el uso de la DNS para equilibrar entre las dos conexiones las peticiones.

7.1.3 Red de servidores

Esta red no va a ser implementada de momento, ya que como hemos comentado no nos hará falta debido a la ausencia de servidores a los que dar servicio de momento.

Por esta circunstancia la única implementación que de momento se realizará será a nivel de software dentro del servidor para permitir en un futuro el *proxy-arp* que engañe a los routers haciéndoles creer que lo que hemos diseñado como dos áreas de captación diferenciadas se comporte como un único dominio de difusión y no dos.

Debido a que la manipulación de las tablas ARP es muy simple en Linux optaremos por usar una pequeña *script* que se encargue de publicar en la interfaz adecuada la IP de la que deseamos hacer *proxy-arp.*, simplemente teniendo que ejecutar esa script la siguiente orden:

```
arp -s <ip> <mac.servidor> pub -i eth0
```

Donde:

- `<ip>` es la IP que está en eth1 o eth2 de la que haremos *proxyarp*.
- `eth0` es la interfaz en cuya red haremos *proxyarp.*
- `<mac.servidor>` es la dirección MAC del servidor que en eth0 será usada para responder a las peticiones de resolución ARP sobre `<ip>`.

Como además podemos querer determinar en el router a partir de cuál de las interfaces de red e IPs disponibles enviamos este tráfico (naturalmente si la IP de origen es privada o de un rango no enrutable por esa conexión no funcionaría a menos que hiciéramos NAT):

```
ip ro ad <ip> dev eth2 table <tabla.a.usar>
```

A partir de ese momento la IP indicada dejará de usar la tabla por defecto de enrutamiento y usará `<tabla.a.usar>`. Esta característica requerirá de un *kernel* reciente de Linux con NETLINK. El problema vendría en el momento en que hubiera demasiadas IPs para gestionarlas de manera manual, introduciéndolas a mano una a una en la tabla ARP, por lo que llegado a ese punto la solución más lógica pasaría por dejar que el servidor principal hiciera *proxy-arp* de manera inteligente sobre todas las IPs, actuando como un puente puro, para lo que únicamente haría falta activarse para cada interfaz mediante la orden:

```
echo 1 >>  /proc/sys/net/ipv4/conf/eth0/proxy_arp
```

7.2 Implantación de los sistemas y servicios ofrecidos

7.2.1 Servidores

Configuración de las interfaces

Debido a que habrá múltiples IPs será necesario disponer de Iproute2. Además, como la configuración de las interfaces de red en *Debian* resulta algo característica (debido a que se utilizan los programas *if-up*/*if-down*), mostraremos ahora como se configuraría una interfaz en el servidor principal en el fichero */etc/network/interfaces*.

Para cada dirección de red que debamos añadir en dicha interfaz podremos darle un nombre cualesquiera a la interfaz virtual que la gestionará. La IP más relevante será asignada a eth0 con una tabla de enrutamiento propia:

```
iface eth0 inet static
      address   128.100.100.2
      netmask   255.255.255.224
      network   128.100.100.0
      broadcast 128.100.100.15
      up        ip ru ad from 128.100.100.2 lookup Red1
      up        ip ru ad fwmark 5 lookup Red1
```

Mientras una cualquiera de las otras IPs iría en este mismo fichero como:

```
iface eth0:Red2 inet static
      address   172.16.0.2
      netmask   255.255.255.0
      network   172.16.0.0
      broadcast 172.16.0.255
      up        ip ru ad from 172.16.100.2 lookup Red2
      …
```

Firewall corporativo

Dado que el servidor principal es aquel por el que pasa todo el tráfico del ISP (a excepción del destinado u originado por el servidor de backup, que se encuentra alojado en otras instalaciones), es en este servidor donde configuraremos el *firewall*.

Ya en los routers hemos definido unas reglas, pero sólo a nivel genérico y con el objetivo de que sirvan únicamente de "primera barrera" del ISP: es ahora cuando por software (usando *iptables*) vamos a disgregar ya a nivel individual el tráfico autorizado del no autorizado, aprovechando además la ocasión para realizar gráficas del consumo de

ancho de banda. En el servidor de backup se configurarán de manera similar, con la única diferencia de que no hay tráfico destinado a otros hosts, como existe en el servidor principal.

Iptables permite la administración del filtrado de paquetes TCP/IP y la realización de NAT que se realiza a nivel de *kernel* en Linux, dentro del módulo que antes hemos, NETLINK. Sin pretender explicar ahora cuál es el modo de funcionamiento de *iptables*, veremos que será necesario distinguir entre el tráfico destinado a este servidor (gestionado en las cadenas INPUT y OUTPUT) y el tráfico destinado a la red local y del que el servidor sólo es responsable de su conmutación hacia dicha red (cadenas FORWARD). Dado que no hay que hacer NAT, y del tunelado hablaremos posteriormente, todas las cadenas a tratar serán las de la tabla *filter*, quedando vacías las tablas *nat* y *mangle*).

A la hora de definir un *firewall*, existen dos estrategias de filtrado que pueden usarse: la primera (y que será la que tomaremos) consistirá en denegar todo, e ir autorizando posteriormente el tráfico legítimo. Aunque es más difícil de configurar permite asegurar que no nos dejamos ningún servicio desprotegido:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -i lo -p 1 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -p 6 --dport 20:21 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -p 6 --dport 22 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 111 -j ACCEPT # portmap (nfs)
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 25 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 53 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -p 17 --dport 53 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 80 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 110 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 113 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 143 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -i lo -p 6 --dport 389 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 443 -j ACCEPT
iptables -A INPUT -s 128.200.0.2/255.255.255.255 \
       -d 0.0.0.0/0.0.0.0 -p 17 --dport 517:518 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -p 6 --dport 993 -j ACCEPT
```

```
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -i lo -p 17 --dport 1812:1813 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -i lo -p 6 --dport 3306 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -p 1 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -p 6 --dport 1024:65535 -j ACCEPT
iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 \
      -p 17 --dport 1024:65535 -j ACCEPT

# por último, nos queda lograr que la red local tenga acceso a
# Internet (sólo correo, Web y DNS), y esta vez,
# para acortar el número de órdenes, usaremos
# el módulo que permite la especificación de más de un puerto:
iptables -A INPUT –s 172.16.0.0/24 –d 0.0.0.0/0.0.0.0 –m multiport \
      -p 6 --dports 80,53,110,25 –j ACCEPT
```

En el listado se han utilizado en lugar de los nombres de los protocolos, los identificadores numéricos de los mismos, que el propio sistema traducirá desde */etc/protocols*, de modo similar a como convierte los nombres de los puertos conocidos a sus números correspondientes desde */etc/services*.

La otra estrategia que habíamos comentado que se podía tomar consiste en realizar justo lo contrario: denegar todo el tráfico que consideremos ilegítimo, y autorizar por defecto el resto. La configuración que permitiría esto debería ser algo similar a lo siguiente (no se ha detallado tanto, porque en este caso hay más normas que especificar, ya que hemos de denegar cada uno de los servicios que no queremos ofrecer, mientras que antes bastaba con autorizar un subconjunto de los mismos):

```
iptables -P INPUT ACCEPPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -A INPUT -s 10.0.0.0/255.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -i eth0 -j DROP
iptables -A INPUT -s 127.0.0.0/255.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -i eth0 -j DROP
iptables -A INPUT -s 172.16.0.0/255.255.0.0 -d 0.0.0.0/0.0.0.0  \
      -i eth0 -j DROP
#iptables -A INPUT -s 192.168.0.0/255.255.0.0 -d 0.0.0.0/0.0.0.0  \
      -i eth0 -j DROP
iptables -A INPUT -s 213.201.48.163/255.255.255.255  \
      -d 0.0.0.0/0.0.0.0 -i eth0 -j DROP
```

Una vez finalizado el *firewall*, existen escaneadores de puertos que permiten validar de una manera rápida aquellos puertos que han quedado abiertos, como *nmap*. También hay herramientas de análisis de seguridad más avanzadas, como *Nessus*.

El mismo *iptables* nos permitirá realizar de una manera simple estadísticas, ya que NETLINK contabiliza tanto el tamaño como el número de los paquetes que verifican una u otra regla.

Por otro lado, aquel tráfico ilegítimo puede ser registrado en el *log* del sistema, mediante la acción LOG:

```
# tráfico ilegítimo:
iptables -A INPUT -s 10.0.0.0/255.0.0.0 -d 0.0.0.0/0.0.0.0 \
       -i eth0 -j LOG
# [si la política por defecto es denegar no hay además que rechazarlo
#  lo de manera explícita]
```

Pero la parte más interesante sería el registrar gráficamente el uso del ancho de banda: *mrtg* permite justamente esto, un comando muy simple que ejecutado de manera repetitiva espera recibir de una programa los valores que posteriormente registra en una gráfica PNG. La utilidad de esta gráfica es obtener una representación viva del tráfico, pudiéndose incluso mejorar esta si la conjuntazos con *RRDtool*, con la que podríamos dibujar múltiples series de datos en la misma gráfica, herramienta realizada al igual que la anterior por Tobías Oetiker (http://people.ee.ethz.ch/~oetiker).

Usaremos una cadena especial en *iptables* por la que haremos pasar todo el tráfico que deseamos analizar. Las cadenas creadas por nosotros no tienen efecto alguno sobre el tráfico, y de las cadenas que creemos vuelve el flujo de análisis de *iptables* de manera directa a la cadena desde la que se realizó el salto.

Como nuestro objetivo no es rechazar tráfico alguno, en esta cadena que creamos las reglas simplemente tendrán como acción un RETURN.

Por ejemplo, si deseamos dibujar la totalidad del tráfico generado por nuestra red local de trabajo, tráfico que además pasa a través nuestro, crearíamos en primer lugar la cadena:

```
iptables -N trafico_local
iptables -J trafico_local -A FORWARD
iptables –I trafico_local -i 172.16.0.0/24 -j RETURN
iptables –I trafico_local -o 172.16.0.0/24 -j RETURN
```

Con la tabla creada, y la totalidad del tráfico pasando por la misma, podemos pensar que hubiera sido más sencillo evitarse usar una cadena distinta y haber operado directamente sobre la cadena FORWARD donde estábamos. Esto no hubiera sido correcto porque en ese caso hubiéramos tenido que forzosamente aceptar o denegar el tráfico, mientras que de esta forma no estamos aún decidiendo el destino de estos paquetes, que podrá ser posteriormente rechazado o

aceptado en función de otros parámetros en la cadena de la que venimos.

Con una frecuencia de minutos, *mrtg* necesitará acceder a esa información y borrar los contadores para que vuelvan a partir de cero:

```
Target[red_local]: `/sbin/iptables -L trafico_local -Z -v -x |
                              tail -2 | /usr/bin/awk '{ print $2; }'`
Title[red_local]: Tráfico global
MaxBytes[red_local]: 500000
PageTop[red_local]: <h1>Tráfico red_local</h1>
Options[red_local]: growright,absolute
Legend1[red_local]: Tráfico interno
Legend2[red_local]: Tráfico a Internet
LegendI[red_local]: interno
LegendO[red_local]: a Internet
```

El resultado sería una página Web que mostraría el tráfico entrante y saliente (ya que nos hemos preocupado de que haya dos reglas en la cadena que distinguen ambos casos).

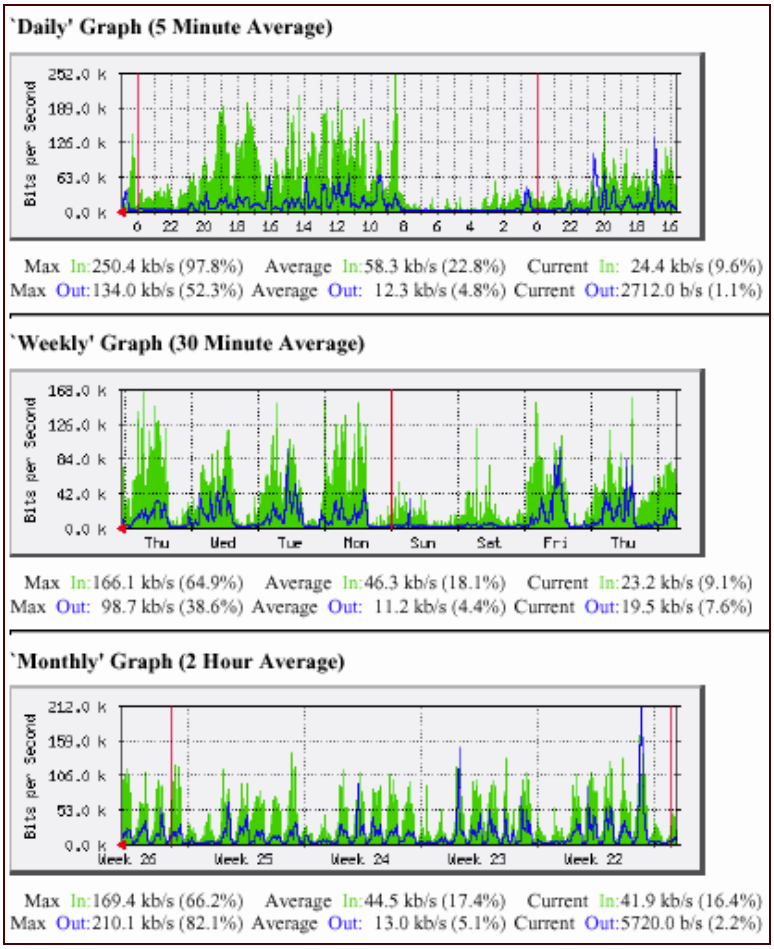


Ilustración 7-3: Ejemplo de un reporte generador por MRTG

Dado que el mismo programa tiene soporte además para SNMP, usaremos *mrtg* también para realizar estadísticas similares sobre el tráfico en el switch y en los propios routers. Existe además la posibilidad de usar *ipaccount*, un programa que trabajado también con *iptables*, facilita bastante las cosas al realizar un agrupamiento del tráfico y reportes semanales.

Calidad del servicio

Uno de los primeros intentos para ofrecer *QoS* aparece en el propio protocolo IP, donde el campo TOS (*Type of Service*) fue y es utilizado en ocasiones como política de enrutamiento, asignando mayor prioridad a los paquetes marcados de una manera u otra. Pero este método no da suficiente flexibilidad.

Otro planteamiento fue *DiffServ*, que utilizaba el campo DS de la cabecera IP, un único bit que viene en la propia cabecera del paquete, por lo que si es conocido el método usado para implantar la calidad del servicio el ISP tendría problemas para hacer efectiva esta calidad, .

Pero el problema realmente estaba mal planteado desde un principio, ya que el principal problema al que nos hemos de enfrentar para garantizar la calidad del servicio es la falta de diferenciación en las tramas ACK que el host cliente intercambia con el servidor de la información y que viajan por el mismo canal.

En ausencia de mecanismos de *QoS* estas tramas ACK se enrutan de la misma manera que las tramas de datos, y pese a su tamaño minúsculo en un momento de saturación sufrirán ellas mismas la congestión.

```
iptables -I OUTPUT -t mangle -p tcp --dport 22 -j TOS \
        --set-tos Minimize-Delay
iptables -I OUTPUT -t mangle -p tcp --dport telnet -j TOS \
        --set-tos Minimize-Delay
iptables -I OUTPUT -t mangle -p tcp --tcp-flags ACK ACK -j TOS \
        --set-tos Minimize-Delay
iptables -I OUTPUT -t mangle -p tcp --dport ftp -j TOS \
        --set-tos Minimize-Delay
iptables -I OUTPUT -t mangle -p tcp --dport ftp-data -j TOS \
        --set-tos Maximize-Throughput
```

Una vez claro que con estas técnicas comentadas no es suficiente, veamos una configuración mínima que garantice al menos que en caso de saturación exista un ancho de banda garantizado mínimo para cada uno de los servicios en uso, aunque en este ejemplo no se esté aplicando esa configuración a ninguna interfaz concreta del servidor:

```
tc qdisc add dev eth0 root handle 1:0 cbq bandwidth 10Mbit cell 8 \
 avpkt 1000 mpu 64
tc class add dev eth0 parent 1:0 classid 1:1 cbq bandwidth \
 10Mbit rate 100kbit avpkt 1000 prio 5 allot 1514 weight 10.0kbit \
 maxburst 5000 bounded
tc filter add dev eth0 parent 1:0 protocol ip prio 5 handle \
 101 fw flowid 1:1
iptables -I OUTPUT -t mangle -p tcp --sport 110 -j MARK \
 --set-mark 10
```

Control efectivo sobre los hosts de la red local

Para facilitar la administración, todas las estaciones de trabajo de nuestra red local deberán aceptar la asignación de la IP mediante DHCP.

Para evitar también contratiempos usaremos *arpwatch*: es una utilidad que realiza un seguimiento sobre la tabla ARP, detectando cualquier cambio de equivalencias entre dirección MAC y dirección IP y notificándolo al administrador. Es por tanto útil para: detectar aquellos hosts de la red local cuya configuración no sea la correcta (usen una IP estática, básicamente). El único inconveniente de esto es que generará falsas alarmas tras cada instalación de un sistema operativo como Windows, en el que es habitual que durante la misma configuración el sistema operativo se asigne una IP del *Link Local Block* (169.254.0.0/16), con intención de autoconfigurar la interfaz.

Por lo que respecta a la configuración del DHCP, este es muy simple: se realiza sobre el fichero */etc/dhcpd.conf*, indicándose cuál es el rango que se desea asignar o las equivalencias concretas entre MAC e IP. Dado que no vamos a tener un número elevado de estaciones de trabajo, no será relevante usar una IP dinámica, sino preferible que en la misma configuración figuren las MAC de todos los equipos y reciban siempre la misma IP, para que además tenga sentido usar *arpwatch*.

```
subnet 172.16.0.0 netmask 255.255.255.0 {
  range 172.16.0.17 172.16.0.253;
  option domain-name-servers bb.172.16.0.2;
  option routers 172.16.0.1;
  option subnet-mask 255.255.255.0;
  option broadcast-address 172.16.0.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

Y tal y como hemos dicho, por cada estación de trabajo concreta indicaríamos cuál es la IP que va a recibir:

```
host pc-direccion {
  hardware ethernet 08:00:07:26:c0:a5;
  fixed-address 172.16.0.25;
}
```

7.2.2 Gestión de los servicios

Del conjunto de servicios a implementar, la totalidad de ellos cumplirá una forma de trabajar común: todos los comandos están en *Python*, y son llamados a través del comando *super* (un programa al estilo de sudo, con el que autorizaremos la ejecución del mismo a los administradores y al usuario bajo el que se ejecuta el servidor Web).

Los servicios serán configurados desde esos comandos, y uno de los servicios, el Web, será además la interfaz visible a los clientes, interfaz que además dispondrá de una base de datos contra la que los clientes se identificarán, y donde residirá la información de los clientes necesaria para la facturación de los servicios.

Hablaremos ahora del tratamiento de esa interfaz y de esos comandos (que son los elementos tratados como paquetes dentro del diagrama de implementación que se puede ver en la Ilustración 7-4).

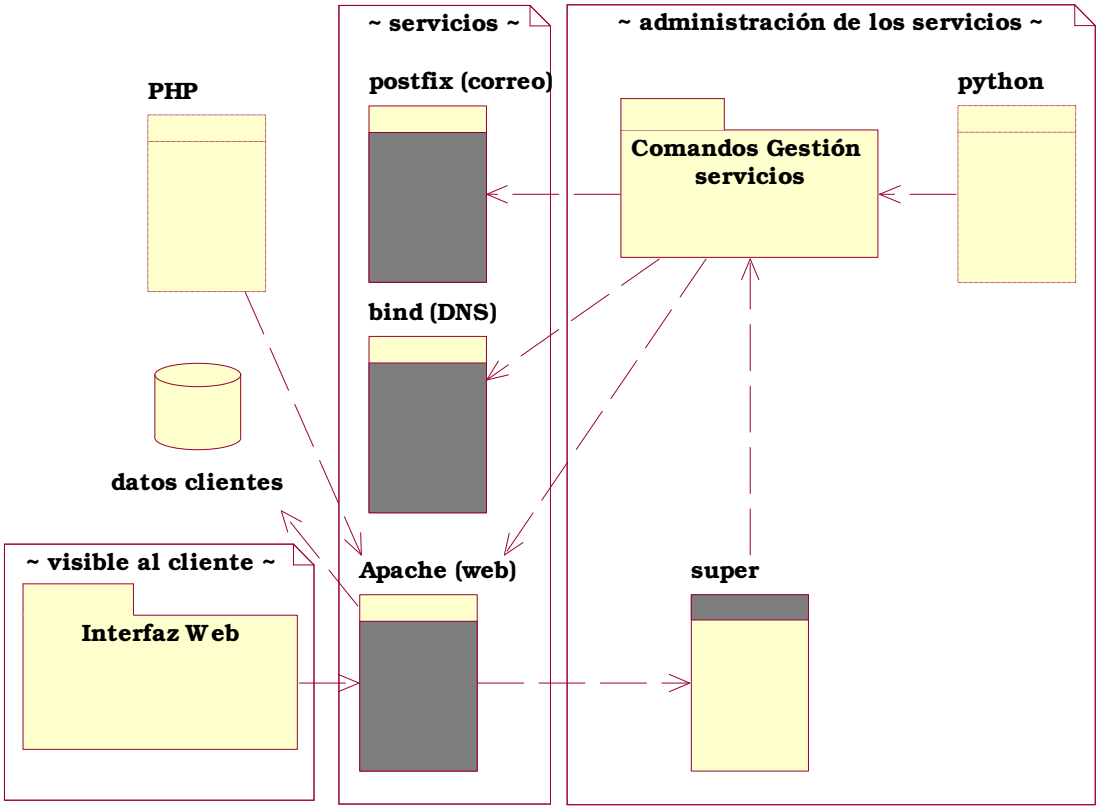


Ilustración 7-4: Diagrama de implementación de contexto

En lo que respecta a los comandos de gestión de los servicios desarrollados en Python, la totalidad de los ficheros involucrados se pueden ver en la Ilustración 7-5.

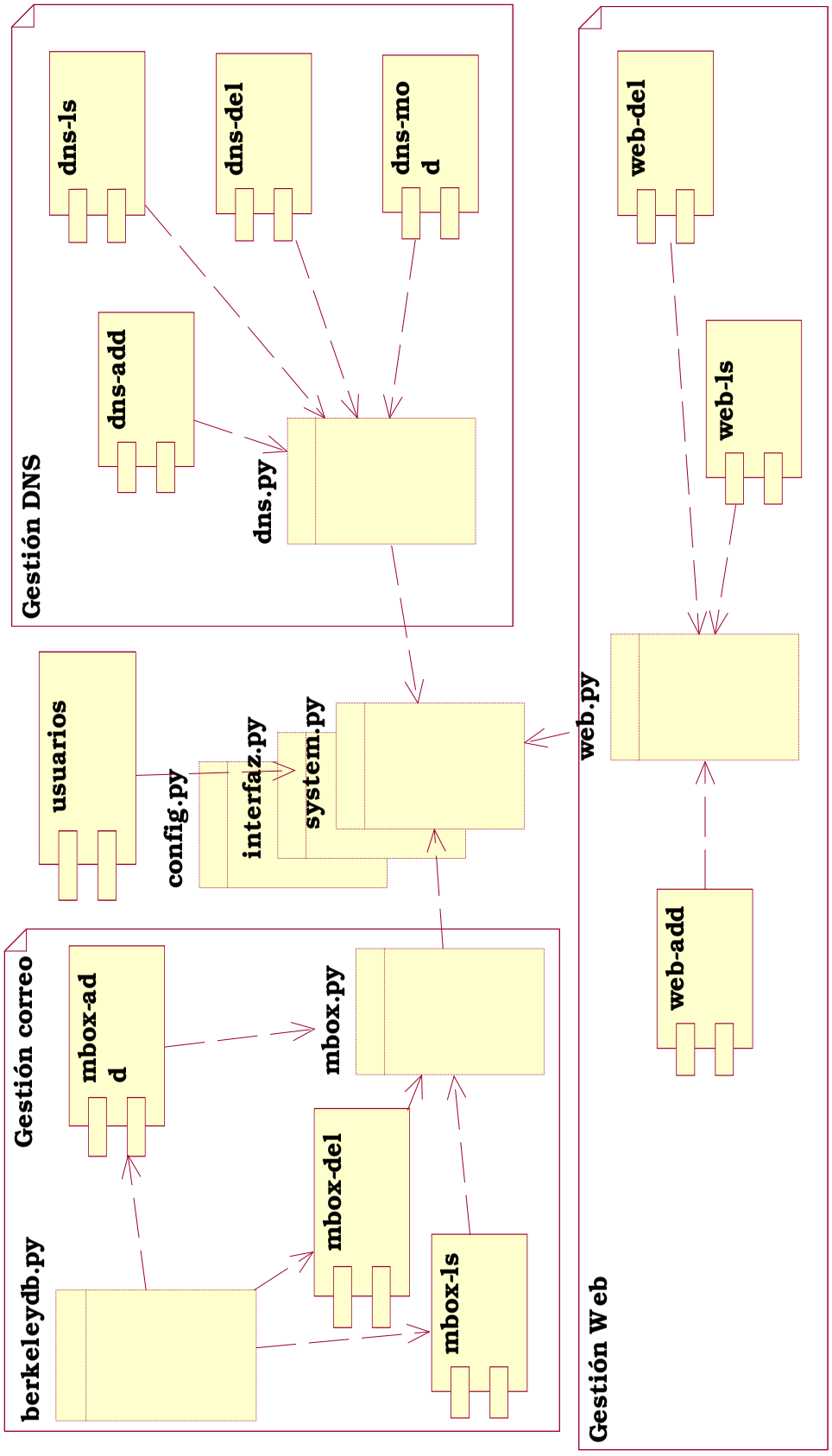


Ilustración 7-5: Diagrama de implementación de la gestión de servicios

Se han agrupado los ficheros en función de a qué servicio ofrecen sus funciones y si son o no librerías.

Las librerías comunes a todos los comandos desarrollados son tres: `config.py`, `interfaz.py` y `system.py`, aunque existen más librerías específicas, una por cada servicio. Para que desde Python las librerías puedan ser cargadas, durante la instalación se ha de colocar bajo el directorio `site-packages` que hay dentro del directorio con las librerías de Pitón una entrada para nuestro software de gestión. Por ejemplo, si tenemos *Python* 2.2 instalado en nuestro servidor, sería `/usr/lib/python2.2/site-packages/scripts_isp`, de forma que luego deberían ser cargadas las librerías mediante una cláusula del tipo `import scripts_isp.nombre_libreria`. Estas librerías, de forma ya más detallada, son las siguientes:

- `config.py`: contiene la configuración del sistema de gestión de servicios (directorios, ficheros de configuración a modificar, etc.). Los ficheros de configuración que los comandos esperan encontrar estarán todos dentro de `/etc/local` o el directorio que se especifique en este fichero de configuración.
- `interfaz.py`: ofrece todo un abanico de funciones destinadas a mostrar por pantalla textos formateados de la manera que luego describiremos y obtener de la línea de argumentos opciones. Como está orientado a que se les llame también desde la Web, este apartado es importante, ya que será esta librería la encargada de hacer que toda la salida consista en texto plano que luego desde PHP pueda ser formateado convenientemente.
- `system.py`: es la librería que ofrece desde el bloqueo de la ejecución de los comandos mediante semáforos durante la modificación de los ficheros de configuración, hasta la función responsable del cambio de contraseña del usuario.

Existen luego un módulo por cada servicio, con las partes concretas de cada servicio, y una más, `berkeleydb.py`, responsable de manejar los ficheros con formato *Berkeley*, usados para contener información a la que se accede repetidamente y no puede estar en formato texto, información que se almacena en este tipo de ficheros indexada y optimizada para posteriores consultas.

La gestión de los servicios requiere que establezcamos ahora un estándar acerca del tipo de salida que pueden generar los comandos.

Tal y como se expresó en el diseño, los comandos deberán devolver una salida estandarizada, para que posteriormente desde la página Web se pueda

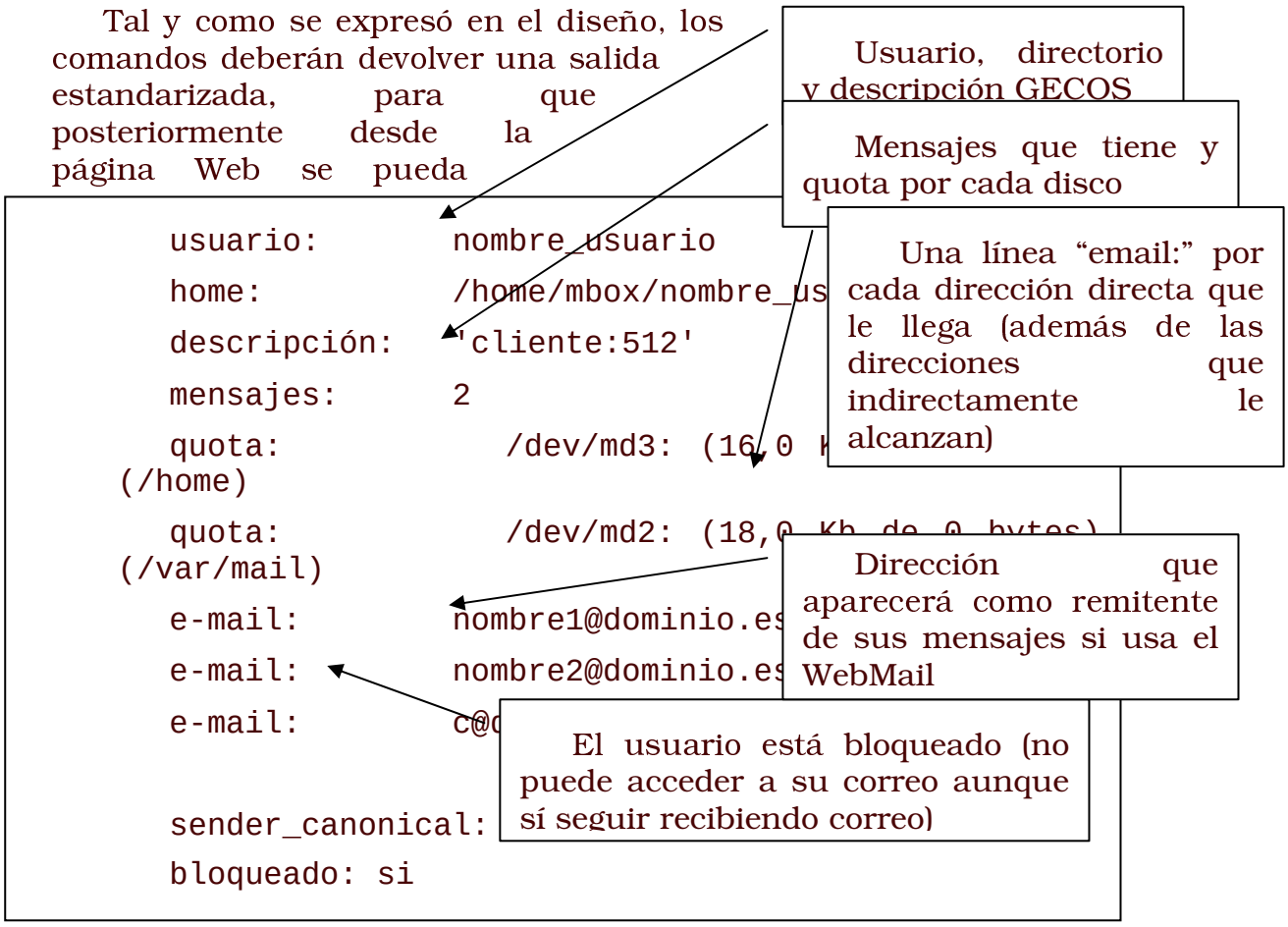


Ilustración 7-6: Salida esperada del comando mbox-ls

llamar a estos comandos y entender las respuestas que ofrecen a nuestras acciones (el valor de retorno de los comandos también se usa como parte de esta información, para detectar si ha habido o no algún fallo).

En concreto el ejemplo mostrado en corresponde a la salida producida por el comando responsable de mostrar los datos de un buzón de correo electrónico.

Toda la gestión se lleva a cabo usando los comandos disponibles a través del comando *super* desde la línea de comandos, autorizándose dicho comando para cualquier usuario de consola que pertenezca al grupo *ispadmin*. Es similar a cualquiera de las variantes de sudo disponibles, y se configura mediante el fichero `/etc/super.tab`. En concreto, las líneas necesarias en ese fichero para hacer funcionar todos los comandos que nos harán falta son las siguientes:

```
mbox-{add,ls,del}   /usr/ispadmin/*.py \
                    info="Gestión del correo" \
                    umask=0022 uid=root gid=root :ispadmin :www-data
```

```
web-{add,ls,del,mod} /usr/ispadmin/*.py \
                     info="Gestión de las páginas web" \
                     umask=0022 uid=root gid=root :ispadmin :www-data

dns-{add,mod,del,ls} /usr/ispadmin/*.py \
                     info="Añadir una zona DNS" \
                     umask=0022 uid=root gid=root :ispadmin :www-data

proxy-{add,ls,del}  /usr/ispadmin/*.py \
                    info="Gestión del servidor proxy" \
                    umask=0022 uid=root gid=root :ispadmin :www-data

network          /usr/ispadmin/network.py \
                 info="Gestión de los equipos autorizados a trabajar" \
                 umask=0022 uid=root gid=root :adm :ispadmin :www-data

servidor         /usr/ispadmin/servidor.py \
                 info="Gestión del servidor (backup,shutdown,etc)" \
                 umask=0022 uid=root gid=root :adm :ispadmin :www-data

usuarios         /usr/ispadmin/usuarios.py \
                 info="Gestión de usuarios" \
                 umask=0022 uid=root gid=root :ispadmin :www-data
```

La script *super* actúa de un modo similar al SUDO pero resulta más compacta su especificación, autorizando la ejecución de determinados comandos a usuarios/grupos concretos.

La gestión a la que se llama desde *super* está enteramente desarrollada en *Python*, y se encuentra bien en */usr/ispadmin* (con las librerías que ejecuta en */usr/ispadmin/lib*), o bien en */usr/ispadmin/bin* los ejecutables necesarios.

7.2.3 Servicio Web

El servicio Web se gestiona desde tres comandos, uno capaz de buscar y mostrar información acerca de las configuraciones existentes, otro responsable de añadir nuevos servicios Web, y un último comando responsable de su eliminación.

web-ls

```
Descripción:

    Mostrar información sobre el sistema web.

Uso:

    web-ls.py [Argumentos]

Argumentos:

    user     <usr_name>                 -> cuentas web
    vhost    <server_name> <ip> <port>  -> sevidores virtuales
    redirect <url_destine>              -> redirecciones
    find     <pattern>                  -> buscar información

    -u, --user=USER_NAME  -> solo servidores virtuales del usuario web
    -l, --list            -> forzar el formato lista

  (c) 2002 by hector@bith.net
```

web-add

```
Descripción:
    Crear cuentas de web y servidores virtuales basados en nombre.
Uso:
    web-add.py [Opciones] [Argumentos]
Argumentos:
    user     <user_name>                                -> cuenta web
    vhost    <server_name> <ip> <port>                  -> servidor virtual
    redirect <server_name> <ip> <port> <URL_destine>    -> redirección
Opciones:

    -u, --user=USER_NAME          -> nombre de usuario
    -p, --password=USER_PASSW     -> contraseña
    -c, --comment=USER_DESC       -> descripción
    -t, --type=USER_TYPE          -> tipo de usuario
    -q, --quota                   -> limita el espacio de disco
    -e, --expire=AAAA-MM-DD       -> fecha de desactivación
    -d, --documentroot=DIRECTORY  -> directorio principal de documentos

  (c) 2002 by hector@bith.net
```

web-del

```
Descripción:
    Borrar cuentas web y servidores virtules basados en nombre.
Uso:
    web-del.py [Opciones] [Argumentos]
Argumentos:
    user     <user_name>               -> cuenta web
    vhost    <host_name> <ip> <port>   -> servidor virtual
    redirect <url_destine>             -> redirección
Opciones:
    -f, --forzar  -> borrar sin conprobación
  (c) 2002 by hector@bith.net
```

7.2.4 Servicio Correo

Al igual que ocurre con la gestión Web, tampoco aquí está previsto un comando capaz de modificar un servicio. La razón es simple: modificar una dirección de correo electrónico asociada a un buzón consiste realmente en eliminar la antigua e insertar la nueva, que es justamente lo que haremos.

Mostraremos ahora algunos ejemplos de uso del servicio de correo. Las acciones más habituales se gestionarán desde los siguientes comandos:

(a) Crear un buzón de correo electrónico nuevo:

```
super mbox-add buzon [nombre_buzon]
```

(b) Borrar un buzón de correo electrónico existente junto con sus direcciones

```
super mbox-del buzon [nombre_buzon]
```

(c) Añadir una dirección de correo electrónico a un buzón existente:

```
super mbox-add email [email],[usuario]
```

(d) Añadir una dirección de correo que hará que el correo se entregue en otra dirección de correo externa a nosotros:

```
super mbox-add email [email],[email]
```

(e) Hacer que todo el correo de un dominio se entregue en un único buzón:

```
super mbox-add email @dominio.com,hector
```

(Es decir, sin indicar la parte anterior a la arroba)

(f) Borrar direcciones:

```
super mbox-del email [email],[usuario|email]
```

Por ejemplo:

```
super mbox-del email hector@bith.net,hector
```

O bien:

```
super mbox-del email hector@bith.net,
```

(Sin indicarle la parte de la derecha, y nos pedirá confirmación, pero ya seleccionará todas las combinaciones en las que aparezca **hector@bith.net**)

También deja:

```
super mbox-del email ,hector
```

(Para borrar todas las direcciones que tiene un usuario)

(g) Hacer que la dirección que aparece al enviar correo desde el Webmail sea una diferente a la que tiene el usuario por defecto (que al listar aparece como sender-canonical):

```
super mbox-add sender-canonical usuario,email
```

(h) Crear un mensaje de autorespuesta para un usuario:

```
super mbox-add autoresponder usuario <<"eof"
    contenido del mensaje que se mostrará
"eof"
```

(i) Borrar un mensaje de autorespuesta:

```
super mbox-del autoresponder usuario
```

(j) Activar / eliminar el filtrado anti-spam para el usuario:

```
super [ mbox-add / mbox-del ] antispam usuario
```

(k) Listar los mensajes que tiene un usuario (con opción de modificarlos):

```
super mbox-ls ver usuario_correo
```

(l) Listar las direcciones que tiene un usuario concreto:

```
super mbox-ls email usuario
Formato del resultado:
"direccion@email.es       >     buzón_entrega"
```

(m) Buscar algo (devuelve todo lo que contenga esa palabra):

```
super mbox-ls <palabra>
```

(n) Ver información sobre una cuenta de correo:

```
super mbox-ls buzon nombre_usuario
```

Se ha de suministrar el nombre del buzón concreto, si se desconoce primero buscar con mbox-ls <elemento_busqueda>).

(o) Modificar la contraseña de un usuario de correo:

```
super usuarios contraseña <usuario>
```

(p) Cambiar algo de un usuario (descripción Gecos, cuotas en disco, etc). Es interactiva (sólo hay que dejarse llevar):

```
super usuarios modificar <usuario>
```

(q) Bloquear/desbloquear un usuario (aparecerá "bloqueado: si" posteriormente al ejecutar `super mbox-ls buzon <usuario>`):

```
super usuarios bloquear <usuario>
```

```
super usuarios desbloquear <usuario>
```

(r) Ver correos que hoy han sido bloqueados por infectados:

```
super servidor correo infectados
```

En realidad, estos comandos disponen de una ayuda en línea propia visible cuando en la línea de argumentos se usa la opción –help o –h, que es lo que mostraremos ahora:

mbox-add

```
Creación de cuentas de correo para hosts virtuales y usuarios en general, o
    añadir direcciones virtuales (parejas email,usuario o bien email,email).
  Con --noenviaremail evitaremos que se informe por email de los cambios al
    usuario.

  Uso: mbox-add.py buzon < nombre_cuenta > < "descripción" > < email1..emailN >
          alias < email,nombre_cuenta | email,email >
          sender_canonical  < nombre_cuenta,email >
          relay < dominio >
          autoresponder < usuarios >
          antispam < usuarios >

  [alias|virtual] -> Añadir una dirección de correo sobre una cuenta existente
                     o que apunte a otra dirección de correo.
  [buzon|cuenta]  -> Para añadir cuentas de correo
    Usar siempre las comillas para delimitar la descripción. Por defecto se
      asume que tendrá cuota y se toma la predeterminada para un usuario de
      correo. La lista de emails ha de contener al menos un email.
    -q, --quota               Limita el espacio que puede usar el usuario al
                               modelo (por defecto)
    -n, --noquota             Sin cuota para el usuario
    -d, --disco=/dev/hda1,2Mb Especifica para un disco la cuota de ese
                               usuario
    -p, --password=PASS       Clave para el usuario (codificada o no). Si no
                               aparece la opción recibe una inválida
    -F, --from=EMAIL          Entrada para el sender_canonical
    -e, --expire=AAAA-MM-DD   Fecha en que se desactivará la cuenta
    -M, --masivo              En modo interactivo es mucho más parco en las
                               preguntas: sólo pregunta el email y genera
                               él TODO lo demás (contraseña aleatoria,
                               usuario,etc) -> sólo útil en modo interactivo
                              Además CUALQUIER ARGUMENTO será interpretado
                               como un email, y el primero usado para
                               generar un nombre de usuario
  [sender_canonical] -> Forzar dirección origen en usuarios interactivos
                        (de consola) y que no corresponda a usuario@maquina
```

```
[relay] -> Permitir el relay de correo en esta máquina (la cuál no es el
           mejor MX ni contiene ningún buzón para este dominio)
[autoresponder] -> Hay que indicar el usuario, y además el mensaje a colocar
           se habrá de indicar por <stdin> (en caso de que no lo haya en
           stdin se tomará el que ya hubiera o una página en blanco)
[antispam] -> Activar el filtrado anti-spam para usuarios concretos
```

mbox-ls

```
Mostrar información sobre una cuenta de correo. Si se introduce un nombre
  de usuario te dice todo lo que te interesa sobre él. Si das un email lo
  que te dice es quién recibe el correo de ese email.

Uso: mbox-ls.py [OPCIONES] buzon < elem > -> ver información sobre una cuenta
                                  sin args muestra todos los buzones
                    alias < elem > -> ver qué va a un usuario / email
                                  (según qué le des, pero si no sabes qué
                                   primero búscalo)
                                  sin args muestra todos los pares
                                  email/buzón
                    sender_canonical < elem > -> mira tabla sender_canonical
                                  y busca concordancias con el elemento
                                  dado, sea usuario / email
                                  sin args muestra todos los pares
                    ver < elem >    -> Ver el correo de ese usuario
                    relay           -> Ver sobre qué ips hacemos ahora relay
                    log             -> Muestra el log de hoy (filtro sed)
                    < patrón >      -> sin opción hace una búsqueda de algo
                                  (acepta ? y *)
                    autoresponder <elem > -> ver por stdout el mensaje que
                                       se enviaría al activarlo


Opciones de búsqueda:
-n,--norecursivo Impide a la búsqueda ser recursiva
-e,--emails      Limita los resultados a los emails si muestra
                   algo del virtual
-u,--usuarios    Ídem pero limitando a usuarios
-r,--regex       Usa una expresión regular como elemento de búsqueda
-1               Ordenar por la primera columna alfabéticamente
-2               Ordenar por segunda columna
-d               El criterio de ordenación no es alfabético, sino que
                   ordena por dominios y luego alfabéticamente
                   (a partir @ ó _)


(c) 2002 by hector@bith.net
```

mbox-del

```
Borrado de cuentas de correo o de alias (direcciones que apuntan a una
  cuenta o a otro email)

Uso: mbox-del.py [OPCIONES] buzon <item>     -> borra un usuario
                   alias <email,usuario>   -> borrar un email que apunta
                                              a un usuario
                   sender_canonical <item> -> borra una entrada del fichero
                                              sender_canonical
                   autoresponder <usuario> -> desactivar el mensaje de
                                              autorespuesta para este usuario
                   antispam <usuario> -> desactiva el filtrado anti-spam

--nobackup no hace una copia de seguridad de los datos del usuario que
           estamos borrando (sin efecto cuando borramos alias)

--force    no confirma el borrado

Una vez borrada la cuenta es irrecuperable la información sobre usuario
y password, sólo se mantendrán los emails (a menos que uses --nobackup)
y las correspondencias que tenía en el virtual.

(c) 2002 by hector@bith.net´
```

7.2.5 Servicio DNS

Respecto a la gestión DNS si que será necesario que implementemos un comando para modificar el servicio aparte de los hasta ahora tres habituales: añadir, borrar y buscar/mostrar información.

La razón es que en el servicio DNS distinguiremos dos situaciones bien distintas: por un lado estará la acción de añadir y borrar una zona DNS (gestionadas a través de los comandos que permiten añadir o borrar dicha zona), y por otro lado estará el comando que permitirá dentro de esa zona DNS crear entradas nuevas o borrar las existentes.

Para facilitar al usuario el trabajo, se ha optado por autogestionar el propio comando el número de serie de la zona DNS que figura en el fichero de configuración que *Bind* usará posteriormente: dicho número de serie estará formado por la fecha en formato AAAAMMDDXX, donde los cuatro primeros dígitos representan el año, luego el mes, el día, y por último dos números, que comenzando por cero, se van incrementando si es necesario que una zona DNS se modifique más de una vez en un día.

Ahora mostraremos algunos ejemplos:

- Crear una subzona (no crea el fichero de subzona, hay que hacerlo aparte):

```
dns-mod dominio.es add \
  subdominio:NS:dns1.isp.es.,dns2.isp.es.
```

- Borrarla (borra todas las entradas NS de la subzona):

```
dns-mod dominio.es del subdominio:NS
```

- Establecer los servidores DNS esclavos (en los que se creará la entrada si no existe) o bien al revés, establecer los maestros (si damos un nombre en lugar de la IP pertiennte, lo resuelve el propio programa):

```
dns-add dominio.es masters 213.201.48.162
dns-add dominio.es slaves \
 215.98.43.25,servidor.principal.es
```

- Reemplazar los MX del dominio principal:

```
dns-mod dominio.es add \
  set @:MX:relay0,relay1,relay2
```

- Si se quiere especificar la prioridad en los relays de correo:

```
dns-mod dominio.es add \
  set @:MX:0+relay0,0+relay1
```

  (De esta manera ambos tienen igual prioridad)

- Otra ventaja es que no hay porqué especificar todo al añadir:

  `dns-mod dominio.es add www:servidor.web.es`

  (El programa ya añadirá después el CNAME pertinente)

- Tampoco hay que especificarlo todo al borrar, basta indicar parte:

  `dns-mod dominio.es dns-mod dominio.es  del www:`

  (Borrará todas las entradas de 'www')

  `dns-mod dominio.es del www:CNAME`

  (Borrará todos los CNAME de 'www')

  `dns-mod dominio.es del www:CNAME:www.isp.es`

  (Borrará esa entrada y dejará el resto)

  `dns-mod dominio.es del www:www.isp.es`

  (El programa ya adivinará que es de tipo CNAME)

dns-add

```
añadir nuevas zonas DNS en BIND

uso: dns-add.py [OPCIONES]  master < nombre > -> añadir una zona maestra
                                                 (hay que indicar esclavos)
                    slave < nombre >  -> añadir una zona esclava
                                                 (hay que indica el master)

si creamos una zona maestra se aceptarán estos argumentos:
    <www> <mx> <ftp> <esclavos>
Y aceptaremos definir los siguientes valores para la SOA:
    -r,--reintento=valor
    -R,--refresco=valor
    -t,--ttl=valor
    -c,--caducidad=valor
    -e,--email=valor
Además con --slaves podremos definir aquellas máquinas en las que está
    instalado ispadmin y también hay que realizar los cambos

Si creamos una zona esclava se aceptará como argumento únicamente
    la IP que actuará de master.
```

dns-del

```
Mostrar información sobre las zonas DNS configuradas en este servidor
para funcionar con BIND.

Uso: dns-ls.py [OPCIONES]  zonas < elem > -> sin args verás un listado con
                                        todas las zonas, con argumento
                                        verás aquellas que coincidan
                                        con la dada
                    zona  < elem > -> ver una y sólo una zona DNS
                                        segundos y posteriores argumentos
                                         serán para mostrar sólo cierta
                                         información:
                                           nºserie (también serial)
                                           tipo (master|slave)
                                           www (ver todas sus entradas)
                                           www,mx (ver sólo sus relays)
                                           soa (nº serie,refresco,email...)
Opciones de búsqueda:
-r|--regex       Usa una expresión regular como elemento de búsqueda

(c) 2002 by hector@bith.net
```

dns-mod

```
Añadir nuevas zonas DNS en BIND
Uso: dns-mod.py < zona > [ACCION]

Si la zona a modificar es MASTER, las acciones pueden ser
(usar --ejemplos para ver un listado algo más amplio):
    add  | del | set  < entrada >
        add - añade a las entradas que ya haya
        del - borra esa entrada (puede no estar completa)
        set - reemplaza a las que ya hubiera
        < entrada > será
    email < valor >      con cualquiera de estos, el valor
    ttl < valor >        suministrado reemplazará al que
                         hubiera previamente
    masters < valor >    define unos nuevos servidores maestros
                         de los que atender notificaciones
    también reintento | refresco | caducidad < valor >
Si creamos una zona esclava se aceptará únicamente:
    add | del | set < ip >
    (para añadir/borrar ips de servidores considerados master)
Dado lo complejo del DNS, la script NO es interactiva
```

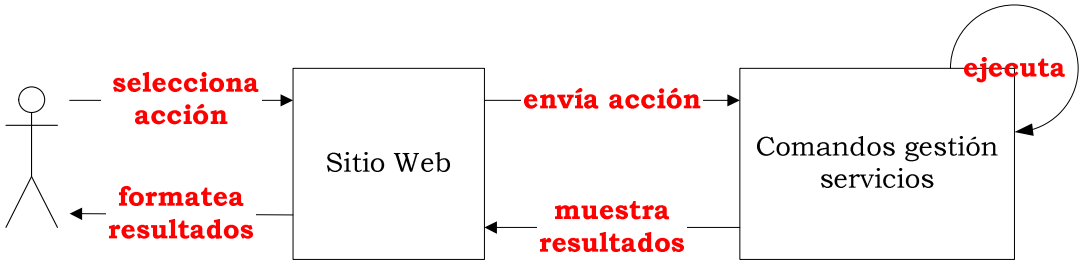
dns-ls

```
Mostrar información sobre las zonas DNS configuradas en este servidor
para funcionar con BIND.
Uso: dns-ls.py [OPCIONES]  zonas < elem > -> sin args verás un listado con
                                    todas las zonas, con argumento
                                    verás aquellas que coincidan
                                    con la dada
                    zona  < elem > -> ver una y sólo una zona DNS
                                    segundos y posteriores argumentos
                                     serán para mostrar sólo cierta
                                     información:
                                       nºserie (también serial)
                                       tipo (master|slave)
                                       www (ver todas sus entradas)
                                       www,mx (ver sólo sus relays)
                                       soa (nº serie,refresco,email...)
Opciones de búsqueda:
-r|--regex       Usa una expresión regular como elemento de búsqueda
```

7.3 Implementación del panel de control

Todo el conjunto de comandos antes comentados son accesibles a través de la Web para que los clientes puedan usarlos directamente, que era el objetivo último de este proyecto (lograr automatizar parte de la gestión de los servicios).

El sitio Web diseñado consiste en realidad únicamente en una interfaz que interactúa con los comandos anteriormente diseñados. Esto genera unos costes computacionales y tiempos de respuesta mayores de lo que hubiera supuesto que todo el conjunto hubiera sido diseñado directamente para ser utilizado desde la Web, con lo que hubiera resultado mucho más optimizado.



Este paso extra nos permite como ya hemos explicado obtener una ventaja que supera las penalizaciones temporales (que por otro lado pueden ser resueltas con mejoras en las prestaciones del servidor): esta ventaja es la unificación en el tratamiento de los servicios: únicamente existirá una forma de crear un buzón, y será a través de un comando que estará disponible tanto desde la consola de comandos para el administrador de sistemas y el personal técnico, como para el cliente a través de una Web debidamente autenticada.

De cómo trabaja la interfaz Web desarrollada en PHP no hay grandes diferencias respecto al panel de control: cada una de las acciones que se podían ejecutar mediante comandos están aquí reproducidas mediante pantallas que filtran de manera adecuada para que el cliente pueda únicamente modificar sus secciones.

Está además el tema de la base de datos: debido a que a este nivel tenemos que lograr la autenticación de los clientes, para que puedan gestionar todos los servicios y sus usuarios, la manera de hacer esto pasará por usar usuarios virtuales que únicamente existirán en la base de datos de clientes.

Los clientes además tendrán en esa base de datos un identificador que provendrá del sistema de facturación, identificador que será siempre el utilizado para controlar los servicios activos, vinculándose servicios y clientes en otra tabla que figurará en la base de datos, y también mediante la inserción del identificador dentro del comentario que se puede definir en el fichero `/etc/passwd` sobre cada usuario.

Se muestra a modo de ejemplo algunas de las pantallas que conforman esta interfaz vía Web. La primera de ellas es la pantalla de identificación ante el sistema:

Ilustración 7-7: Pantalla de acceso al sistema de gestión vía Web

Una vez realizada la autenticación, el cliente puede acceder a su pantalla de gestión de servicios, donde podrá listar y modificar sus cuentas de correo, las contraseñas de estas, etc.

El único servicio que se muestra con diferente nombre y nivel de administración es el DNS: en la Web se ha de permitir por un lado registrar dominios (cosa que no es posible desde los comandos de consola implementados), y por otro lado ha de permitir crear nuevas entradas de una manera mucho más simple que desde los comandos, eliminándose la mayoría de opciones (como configurar los servidores DNS, o los *relays* de correo, ya que no se espera que el cliente tenga conocimientos lo suficientemente avanzados como para comprender estos aspectos, por lo que desde la Web sólo se pueden añadir nuevas subzonas y su IP.

Ilustración 7-8: Pantalla de gestión de buzones de correo

Ilustración 7-9: Pantalla de gestión de direcciones de correo

8 Análisis de viabilidad económica del proyecto

Aunque excede ya los objetivos de este tipo de proyectos, no podemos acabar sin explicar antes qué posibilidades de negocio tiene nuestro ISP. Se trata básicamente de demostrar, de la manera más concisa posible, que un proyecto como este puede tener conexión con la realidad empresarial, aunque no se trate de realizar una puesta en práctica real del mismo.

Un ISP es una empresa de servicios: el nicho de clientes para los servicios de alojamiento Web y correo será nuestra fuente de ingresos, y es la determinación del número de clientes que podremos obtener lo que nos permitirá por un lado saber qué gastos mensuales habrá de soportar el ISP, y por otro lado qué beneficios quedarían en función de los precios que daremos a los servicios.

8.1.1 Estudio de mercado

A partir de los dos productos básicos que deseamos ofertar, alojamiento Web y correo, se van a dar tres tipos de clientes. En primer lugar, aquellos que contraten ambos productos y constituyan los grandes clientes, con sitios Web de gran tamaño y un número elevado de cuentas de correo electrónico (estos constituirán lo que hemos dado en llamar grandes sitios Web, con un consumo más elevado).

En segundo lugar estarán los clientes que disfruten de ambos servicios pero a menor escala, con sitios Web de menor consumo de ancho de banda y menor número de cuentas de correo electrónico, que se verá reflejado también en un precio más ajustado. De hecho al analizar el ancho de banda que necesitaríamos ya diferenciamos a este nivel, distinguiendo los sitios Web en función del tráfico que soportarían.

Finalmente nos encontraremos con algunos clientes que sólo deseen correo, y a los que se ofrecerá ese servicio de manera aislada al otro producto. No se concibe la existencia de sitios Web que no se vean

acompañados de buzones de correo electrónico, por eso no lo tendremos en cuenta, y no se concibe justamente porque en cualquier sitio Web habrá de haber una manera de contactar con el propietario de la misma.

Aquellos clientes que sólo deseen correo lo harán bajo un dominio propio igualmente, ya que no es de esperar que alguien desee una cuenta de correo bajo un dominio genérico siendo gratuita su obtención en cualquiera de los portales de operadores de acceso a Internet. Queda por tanto claro que el dominio interviene en todos los tipos de cliente, y que habrá un dominio registrado con cada producto (obviamente si un cliente posee dos dominios o más en cada dominio tendrá que tener también cuentas y espacio Web).

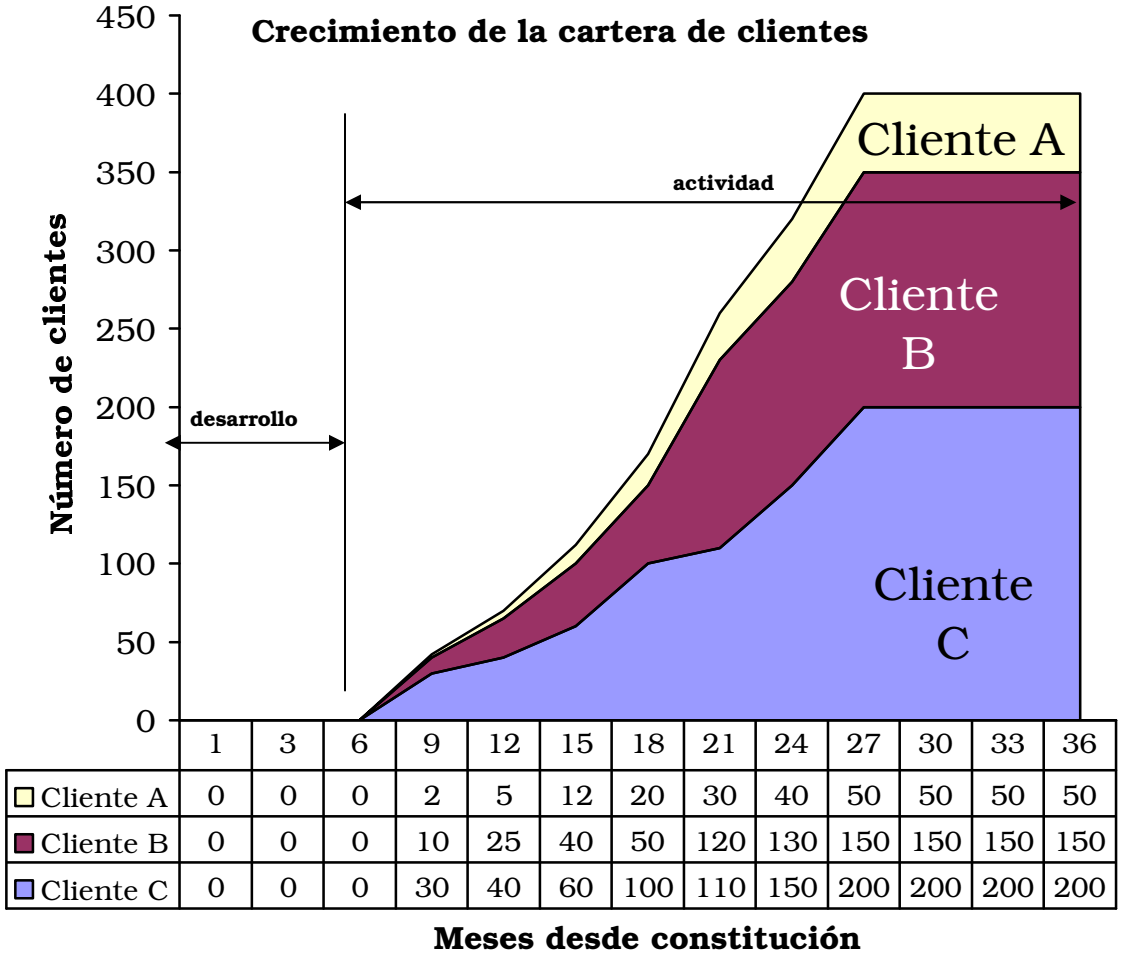
Con el cliente objetivo de nuestro ISP totalmente segmentado ahora deberíamos buscar la forma de captar a estos clientes, que será mediante paquetes de productos que incluyan el número de cuentas de correo y el espacio Web a la medida de sus necesidades. Determinar la combinación de cuentas de correo y tamaño del espacio Web es una tarea que competería a la política comercial y aquí no se realizará, sino que el precio estará ajustado al coste anteriormente estimado de ese conjunto de cuentas y espacio Web.

El número de clientes que esperamos poder captar con este planteamiento será de 400, que segmentaremos en función del tamaño de la Web y el uso del correo que vayan a hacer, tal y como arriba se ha explicado:

	Número de clientes	Uso medio esperado por cliente
Web de elevado consumo (Cliente A)	100	1 espacio Web + 40 buzones de correo
Web media (Cliente B)	150	1 espacio Web + 10 buzones de correo
Sólo correo (Cliente C)	200	5 cuentas de correo

El objetivo sería cumplir estas cifras a partir del año y medio desde el inicio de la actividad, con un crecimiento exponencial en el número de clientes, dada la dificultad inicial de entrar en el mercado (novedad, desconocimiento hacia la marca del ISP, etc.). Por tanto el estudio de viabilidad va a tener en cuenta este factor de crecimiento exponencial en su análisis, crecimiento que será más resistente en el caso de los sitios Web de gran tamaño (de estos que esperamos captar al menos 100 clientes).

Dado que estamos hablando de lograr captar estas cifras de clientes en 18 meses, el estudio de viabilidad va a tener en cuenta también el tiempo estimado de ingeniería necesarios para el presente proyecto, que era de quince meses, pero sin esperar a la finalización del mismo: supondremos que a partir del sexto mes de actividad de desarrollo seremos capaces de disponer de unos sistemas configurados en el ISP con los que ofrecer servicios, aunque sin el panel de control terminado.



	1	3	6	9	12	15	18	21	24	27	30	33	36
Cliente A	0	0	0	2	5	12	20	30	40	50	50	50	50
Cliente B	0	0	0	10	25	40	50	120	130	150	150	150	150
Cliente C	0	0	0	30	40	60	100	110	150	200	200	200	200

Ilustración 8-1: Crecimiento esperado de la cartera de clientes

La curva de crecimiento se estanca una vez cumplido el objetivo de clientes (en el mes 27 desde la constitución, con la carencia de seis meses del proyecto de desarrollo). Vamos a suponer que a partir de ese momento no hay crecimiento de clientes. Esto no será así de manera textual: sino más bien se espera que entremos en una fase de consolidación con algunas bajas (lógico en el mundo empresarial), y por tanto es preferible hacer el estudio a 36 meses sin estar creciendo siempre.

8.1.2 Viabilidad económica

Conviene destacar una vez más que la viabilidad del negocio analizada está supeditada únicamente al alojamiento, sin considerarse los más que probables ingresos extra sobrevenidos por el diseño de los sitios Web que pretendemos alojar (que a su vez redundarían en más gastos de personal, ya que no se ha previsto ningún diseñador ni programador Web dedicado a esta tarea exclusiva): este proyecto está basado únicamente en ofrecer servicios de datos y conectividad.

Con las cifras anteriores de número de clientes esperados, y los gastos ya calibrados, se puede plantear ahora de manera numérica la viabilidad económica de este proyecto antes de comenzar siquiera a diseñarlo de manera concreta.

Ingresos

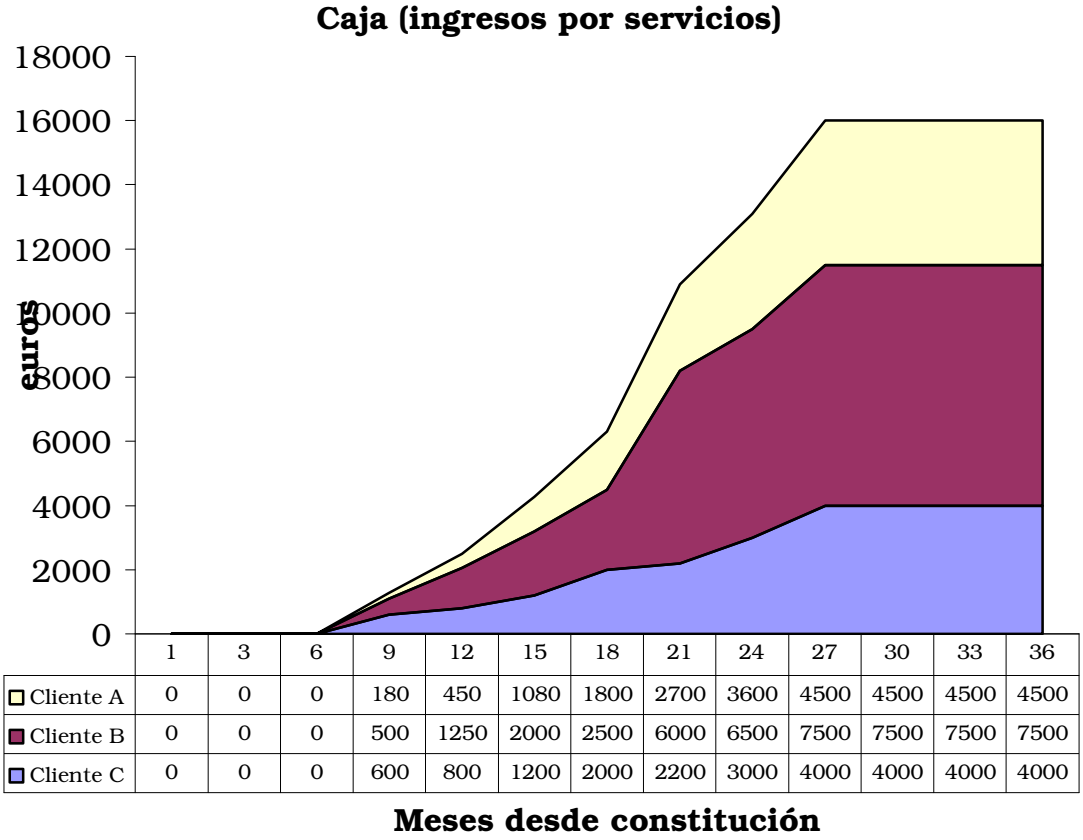
Aunque pueda parecer extraño, comenzaremos hablando de los ingresos previstos en función de la evolución de los clientes que vayamos adquiriendo, en lugar de tratar los costes. La razón es muy simple: con un periodo inicial de seis meses sin ingresos (lo que durará el desarrollo del proyecto para permitir la gestión del ISP) y unos meses más en los que los gastos van a seguir siendo mayores que los ingresos, necesitamos saber en qué momento se espera alcanzar el equilibrio entre ingreso y gasto, para determinar así hasta qué mes desde la constitución de la empresa nos hará falta el dinero del crédito, y por tanto poder determinar su importe.

Como ya existen caracterizados tres tipos de clientes, y la evolución prevista de los mismos, veremos tras asignarles el precio al producto que cada cliente tipo contratará cuál será el balance de caja. El precio de los productos ha sido asignado en función de una comparativa con productos similares existentes en el mercado. Se han analizado los paquetes de *housing* y correo ofrecidos por compañías como *Arsys* (**www.arsys.es**), *Acens* (**www.acens.es**), *InterHost* (**www.interhost.com**) y otros. Dado que seremos desconocidos en el mercado, la única forma de atraer clientes será mediante una rebaja sobre los precios de los productos de estas compañías, para que podamos introducirnos en el mercado y atraer a clientes:

Tabla 8-1: Precios asignados a los productos ofertados

	Número de clientes	Ingresos por cliente
Web de elevado consumo (Cliente A)	100	90 €
Web media (Cliente B)	150	50 €
Sólo correo (Cliente C)	200	20 €

Suponiendo estos valores, y a partir de las suposiciones de clientes hechas en la Ilustración 8-1, podremos obtener los ingresos que aspiramos a lograr. Hasta llegar a los 16.000 euros de ingresos mensuales que se supone tendremos a partir del mes 27, según mostramos en la Ilustración 8-2, hemos de asegurarnos que los gastos mensuales no nos ahogan, sobretodo durante los primeros meses de actividad en los que aún no hayamos adquirido suficientes ingresos.



	1	3	6	9	12	15	18	21	24	27	30	33	36
Cliente A	0	0	0	180	450	1080	1800	2700	3600	4500	4500	4500	4500
Cliente B	0	0	0	500	1250	2000	2500	6000	6500	7500	7500	7500	7500
Cliente C	0	0	0	600	800	1200	2000	2200	3000	4000	4000	4000	4000

Ilustración 8-2: Crecimiento esperado de los ingresos

Como posteriormente analizaremos, la mejor fórmula al respecto será una financiación inicial que incluya no sólo el gasto inicial, sino todos los gastos hasta que podamos cubrirlos con los ingresos.

Gastos

Los gastos se repartirán entre unos gastos iniciales y otros gastos fijos mensuales que habremos de cubrir mediante un préstamo los seis primeros meses en los que no habrá actividad y por tanto clientes con cuyos ingresos podamos contar.

Los gastos provienen de los cálculos realizados en la fase de análisis, aunque ha habido que considerar otros gastos no técnicos, como la constitución de la empresa y otros similares. Aunque aquí los gastos han sido clasificados en función a su naturaleza mensual o puntual: es decir, por gastos iniciales se computa la constitución de la empresa, la adquisición de hardware y software, y el personal temporalmente necesario para el desarrollo del programa, que luego no continuaría en este ISP. Esto implica retirar 98.594,15 € de los 331.898,43 € en que fue valorado el proyecto al completo, ya que éstos corresponden a los gastos mensuales fijos que se han detallado a continuación.

También en los gastos mensuales se ha tenido en cuenta la amortización del hardware (capítulo importante, ya que su tiempo de vida es inferior a cualquier otro elemento mobiliario de una empresa), y también los gastos provocados por los servicios que la empresa ha de contratar (luz y agua, por ejemplo), destacando entre ellos la conectividad del ISP.

De las dos conectividades que se tienen previsto contratar hay que darle una valoración en función del *Kilobit* de ancho de banda, porque el ISP necesitará contratar con un operador que sea capaz de venderle un ancho de banda garantizado y ampliable si las circunstancias lo aconsejan, y para eso la tendencia mayoritaria es orientar el cobro al ancho de banda disponible en lugar de la transferencia total mensual realizada.

Si el operador nos factura por ancho de banda disponible, existirá un precio por *Kilobit* por segundo disponible, que figura en la tabla como *Kbps* PIR (dado que está garantizada su disponibilidad por el ISP).

El problema que tendremos para ampliar ancho de banda es que la contratación del ancho de banda se realiza por bloques, sin posibilidad de que nosotros ampliemos el ancho de banda con la granularidad deseada. Por tanto supondremos que el bloque mínimo de ampliación de ancho de banda es de ½ *Mbps* en cualquiera de los dos tipos de conexión previstos.

El margen de seguridad de ancho de banda sin usar para evitar posibles saturaciones puntuales quedará sobradamente cubierto por el porcentaje de ancho de banda no garantizado de que dispondremos, que además será útil para la navegación en Internet y operativa en general de la oficina (es decir, no hará falta otra conexión para navegar desde la oficina).

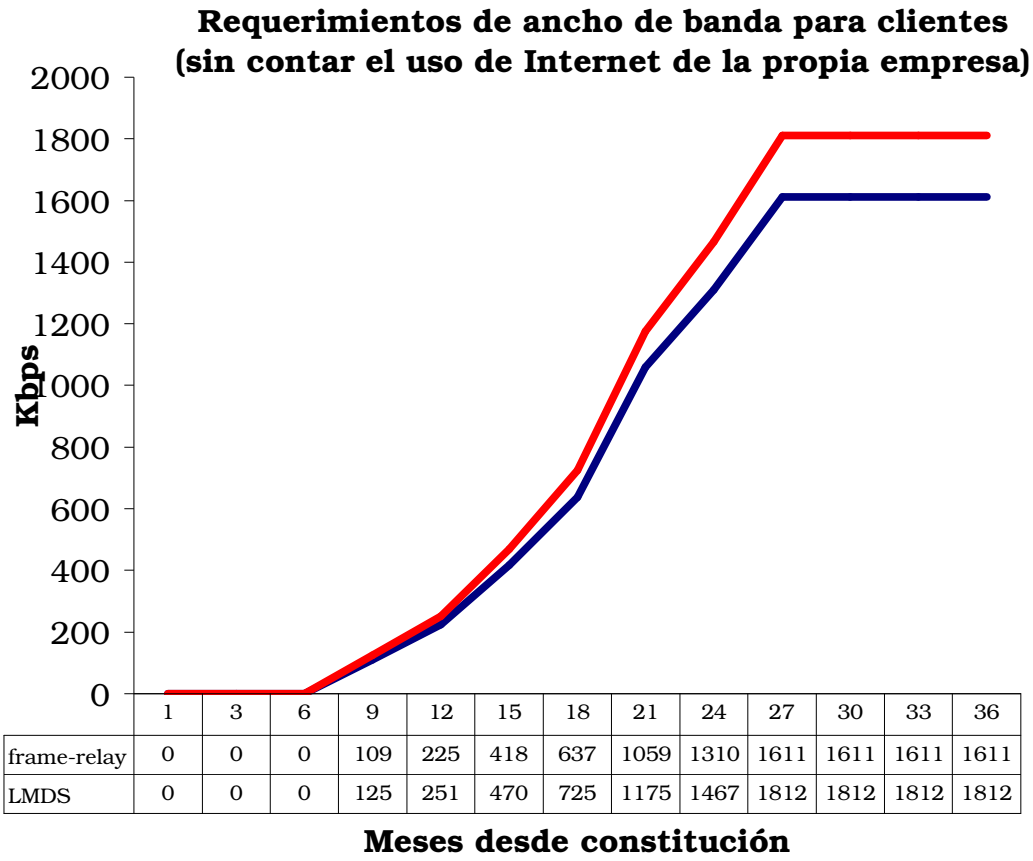
Tabla 8-2: Gastos del ISP

Gastos iniciales (a cubrir mediante préstamos)	
Equipamiento informático y acondicionamiento local	8.000,00 €
Constitución de la empresa y gastos del préstamo	700,00 €
Desarrollo del proyecto de ingeniería descrito en esta memoria	225.300,00 €
Otros gastos mensuales fijos no cubiertos por ingresos	39.000,00 €
	273.000,00 €
Gastos mensuales fijos (desde constitución hasta mes número 36)	
Amortización servidor y equipamiento redes	100,00 €
Equipamiento informático: *renting* 3 PCs	100,00 €
Housing del segundo servidor	200,00 €
Luz, agua y alquiler local	900,00 €
Personal	4.000,00 €
	5.300,00 €
Gastos mensuales de caudal y dominios (a partir inicio actividad)	
Circuito *Frame-Relay* (coste *Kbps* PIR)	0,40 €
Circuito LMDS (coste *Kbps* PIR)	0,07 €
Espacio Web grande (9.48 *Kbps*, ½ LMDS ½ *frame*)	2,22 €
Espacio Web medio (0.76 *Kbps* por *frame-relay*)	0,30 €
Cuenta correo (0,35 *Kbps* LMDS, 0,28 *frame-relay*)	0,14 €
Dominio	0,83 €
Cliente A (Web grande + 40 buzones + dominio)	8,65 €
Cliente B (Web medio +10 buzones + dominio)	2,53 €
Cliente C (5 buzones + dominio)	1,53 €

En lo que respecta a los gastos mensuales fijos, si se computan únicamente 4.000 euros destinados al capítulo de personal es debido a que los costes laborales asociados al desarrollo del proyecto se han tenido en cuenta en los 225.300 € detallados entre los gastos iniciales: aunque no se trate de un gasto inicial mismamente, el desarrollo y los costes que el mismo comporte tendrán fecha de finalización, a menos que la planificación no pueda ser cumplida.

De los gastos generados por el caudal se necesita ahora y también a partir de los datos que figuran en la Ilustración 8-1 calcular cuál será el gasto que supongan: el crecimiento en la necesidad de ancho de banda será una curva creciente, pero en cambio el gasto que comportará no, debido a que el operador que nos suministre el ancho de banda no nos va a permitir aumentar un circuito *frame-relay* en unos pocos *Kilobytes* únicamente, sino que nos exigirá hacerlo en unos bloques que hemos considerado serán como mínimo de ½ Mbps por lo que respecta a la *frame-relay*.

Tabla 8-3: Ancho de banda requerido

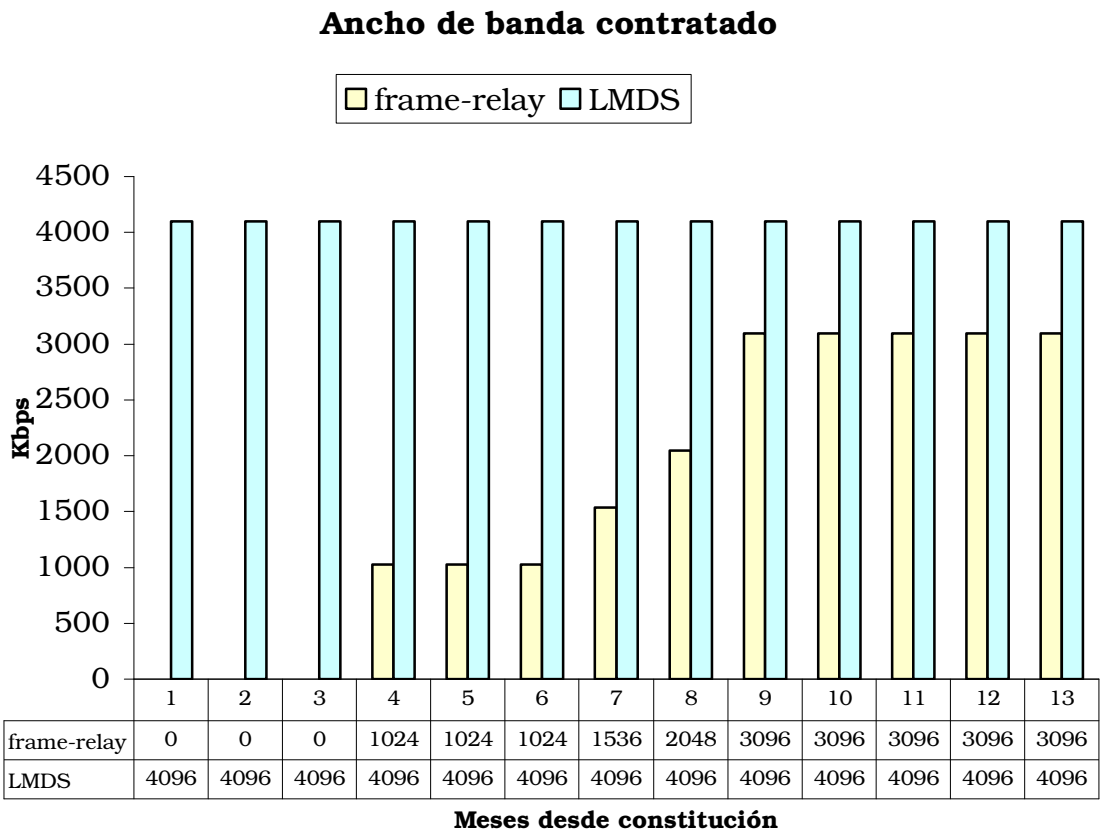


	1	3	6	9	12	15	18	21	24	27	30	33	36
frame-relay	0	0	0	109	225	418	637	1059	1310	1611	1611	1611	1611
LMDS	0	0	0	125	251	470	725	1175	1467	1812	1812	1812	1812

Por lo que respecta a la conexión LMDS, dado que el circuito contratado tiene un ancho de banda PIR de hasta 4 *Mbps* no se nos planteará la necesidad de ampliarlo, pero en el caso del circuito de *frame-relay* hará falta ampliarlo en repetidas ocasiones, para evitar tener que partir con un ancho mayor del necesario.

Si la conexión *frame-relay* dispone de un CIR de 512 *Kbps*, cuando la proyectada necesidad de dicho canal supere esa cantidad solicitaremos la ampliación de ½ *Mbps* antes comentada, de manera que pasaríamos de 1 *Mbps* con 512 *Kbps* de CIR a 1,5 *Mbps* con 768 *Kbps* de CIR, y así consecutivamente, por lo que ahora necesitaremos saber cuánto nos va a costar realmente esto en función de cuál será en cada instante el ancho de banda contratado.

Sólo incumpliremos esta premisa cuando se logre el objetivo de haber captado los clientes previstos, cuando el caudal de *frame-relay* necesario será de 1611 *Kbps*, y que para cubrir con CIR habríamos de tener un circuito de 3,5 *Mbps*, mientras que nos conformaremos con uno de 3 *Mbps* (ya que el exceso no es excesivo):



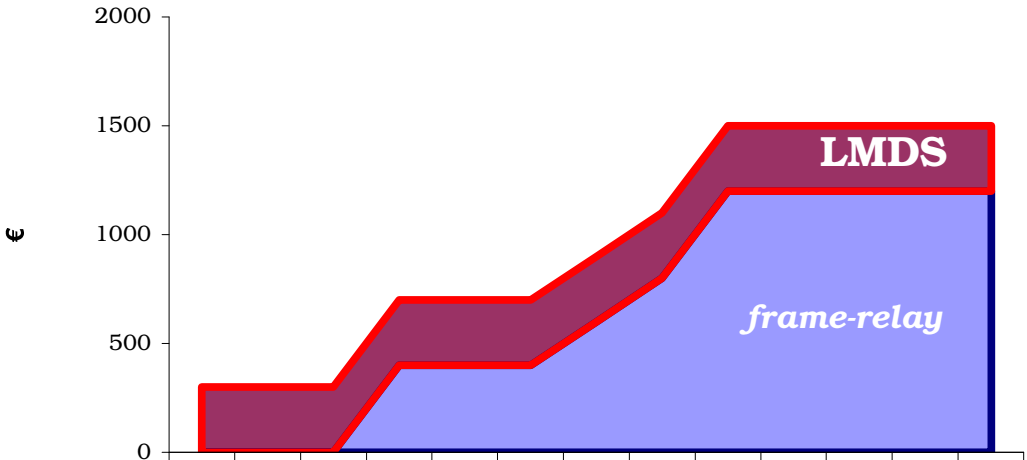
	1	2	3	4	5	6	7	8	9	10	11	12	13
frame-relay	0	0	0	1024	1024	1024	1536	2048	3096	3096	3096	3096	3096
LMDS	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096

Ilustración 8-3: Evolución del ancho de banda contratado

Con respecto al ancho de banda contratado se va a hacer una excepción para permitir la actividad y las pruebas durante el periodo inicial de seis meses de desarrollo sin servicios reales ofertados al cliente, usando la línea LMDS. La conexión de *frame-relay* se contratará una vez comencemos a generar ingresos, en el sexto mes.

Ilustración 8-4: Evolución temporal del coste ancho de banda del ISP

Coste del ancho de banda contratado



	1	3	6	9	12	15	18	21	24	27	30	33	36
Total	300	300	300	700	700	700	900	1100	1500	1500	1500	1500	1500
LMDS	300	300	300	300	300	300	300	300	300	300	300	300	300
frame-relay	0	0	0	400	400	400	600	800	1200	1200	1200	1200	1200

Meses desde constitución

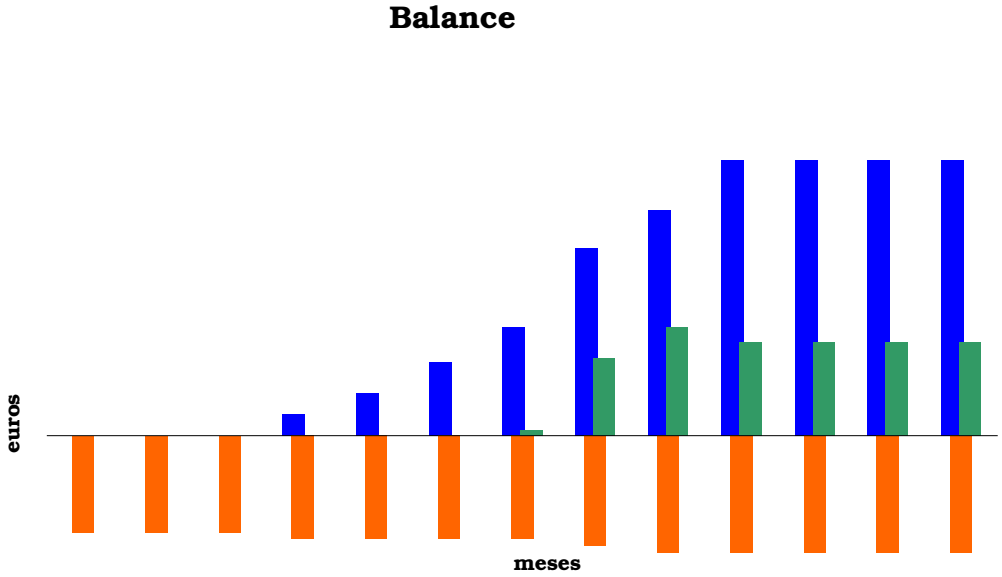
Como ya se dejó intuir al principio de este estudio de viabilidad, se ha previsto cubrir los gastos hasta la obtención de ingresos suficientes mediante una ampliación del crédito inicial. Estos ingresos llegarán a partir del mes 18 desde el comienzo del proyecto de desarrollo, tal y como figura en la Ilustración 8-5.

El ISP será en ese momento autónomo financieramente para cubrir los previstos 5.300 euros de gastos fijos y los gastos derivados del ancho de banda (hasta 700 euros más cuando contratemos todo el ancho).

Por tanto el crédito que solicitaremos cubrirá los gastos iniciales de la empresa y estos gastos de funcionamiento del primer año y medio, incluyendo además el préstamo un periodo de carencia en el pago de 18 meses. Se ha considerado que la obtención de este crédito a un interés del 5% TAE, y con plazo de amortización de 36 (a partir de los 18 meses de carencia) no es descabellado con los debidos avales y la correspondiente negociación con los bancos,

pudiéndose suplir por otros mecanismos de financiación que se estimen oportunos, ya que en muchas ocasiones la aportación de capital de la gerencia de la empresa haría innecesario parte de este crédito.

Ilustración 8-5: Evolución del balance contable del ISP



	1	3	6	9	12	15	18	21	24	27	30	33	36
Ingresos	0	0	0	1.280	2.500	4.280	6.300	10.900	13.100	16.000	16.000	16.000	16.000
Gastos	-5.600	-5.600	-5.600	-6.000	-6.000	-6.000	-6.000	-6.400	-6.800	-6.800	-6.800	-6.800	-6.800
Beneficio	0	0	0	0	0	0	300	4.500	6.300	5.433	5.433	5.433	5.433

El préstamo

El préstamo a obtener será por un importe de 343.000 €, incluyendo tanto los 273.000 € considerados necesarios como inversión inicial como los casi 70.000 € restantes para cubrir los gastos de funcionamiento hasta el mes 18, cuando los ingresos superarán a los gastos y podremos hablar ya de beneficios.

La cuantía del préstamo es considerable, y su amortización supondrá todos los beneficios existentes durante los 36 meses hasta los que hemos llevado el análisis de viabilidad que estamos realizando. Por tanto el ISP habrá logrado el equilibrio sólo tras tres años de actividad completa.

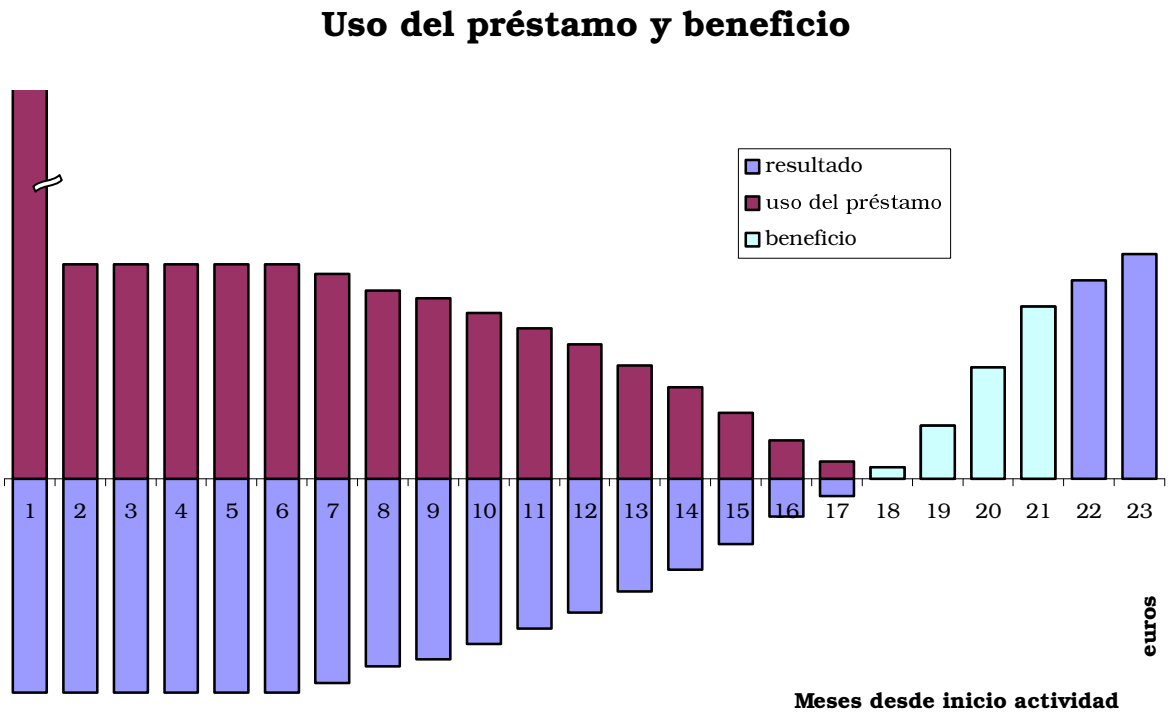
Además, dado que entre la solicitud del préstamo y la obtención de ingresos el ISP no podrá pagar el empréstito, habrá que por tanto solicitar un periodo de carencia de al menos 21 meses a contar desde su obtención. Hablamos de 21 meses en lugar de 18 debido a que estos tres meses no podremos tampoco cubrir con los

beneficios obtenidos la cuota mensual de amortización de préstamo que el ISP deberá realizar.

Este estudio de viabilidad no entra a valorar aspectos contables como el equilibrio entre activo y pasivo que el ISP como empresa debería tener: es evidente que el empréstito solicitado superará todos los posibles activos de la empresa, por lo que habrá que buscar mecanismos financieros que eviten la inviabilidad financiera del préstamo, ya que excede las capacidades de un proyecto de Ingeniería Informática.

Entre el mes 18 y el mes 21 disfrutaremos por tanto de mayores beneficios que los meses inmediatamente siguientes, tal y como muestra el siguiente gráfico, debido a que hasta ese mes 21 no pagaremos los 3.766,84 € de cuota fija del crédito. Por uso del préstamo se entiende la distribución por meses del uso del dinero obtenido por el préstamo.

Ilustración 8-6: Evolución del resultado económico del ISP



En el caso de que las previsiones de ingresos y gastos no se cumplieran, y el crecimiento fuera menor, se disponen de tres meses (los comprendidos entre el 18 y el 21) en los que los 4.580 euros obtenidos de beneficio pueden ser utilizados, porque aún no son necesarios para ningún crédito (ya que se pagará a partir del mes 21). Una vez alcanzado el mes 21, la única solución pasaría

por renegociar con el banco un periodo de carencia mayor para el crédito, siempre que los gastos excluyendo el crédito no superaran a los ingresos, que supondría el fracaso del negocio.

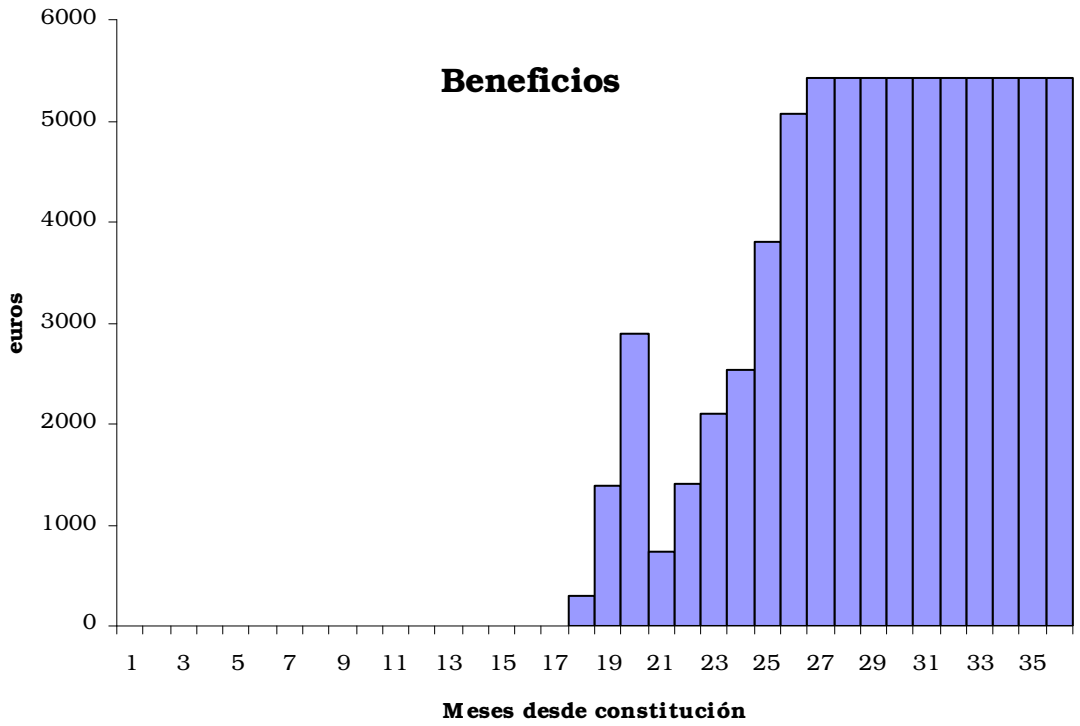
Rendimiento previsto

El otro aspecto no valorado hasta el momento es el sueldo del gerente de la compañía, que se estima sujeto a la existencia de beneficios.

Si consideramos que a partir del mes 18 ya existen beneficios del negocio, y que en este no ha sido necesario aportar capital alguno, se puede estimar un rendimiento aceptable tras los 36 meses analizados (o incluso directamente en los 54 meses desde la constitución, momento en el que dejaríamos de pagar el crédito, una vez transcurridas las 36 cuotas del mismo y los 18 meses de carencia iniciales), siendo por tanto lógico que la gerencia viva exclusivamente del rendimiento del negocio.

El rendimiento se estima transcurridos los 36 meses en un total de 74.560,49 €, con lo que excluyendo el hecho de que hasta el mes 18 no llegan los beneficios, y haciendo media, se esperan obtener un beneficio de 2.071,12 € al mes.

Ilustración 8-7: Evolución de los beneficios previstos en el ISP



Aunque la cifra pueda parecer baja, no es así si tenemos en cuenta que a partir del mes 54 se dejará de pagar el crédito, y que además no se ha analizado el crecimiento que a partir del 27 podría tener el negocio, siendo por tanto más que razonables las cifras de beneficio presentadas.

El objetivo es plantearse un ISP operativo, rentable y limpio de cargas en 3 años desde su puesta en funcionamiento.

9 Epílogo

9.1 Conclusiones

El proyecto ha sido realizado en los plazos previstos y cumpliendo los objetivos marcados: se ha especificado el diseño e implementación que deberá tener un ISP de tamaño medio, y sobretodo se ha desarrollado el software que permitirá gestionarlo.

De los logros obtenidos el software de gestión es la parte más relevante, dada su doble funcionalidad: por un lado los clientes del ISP podrán realizar por ellos mismos tareas que antes ocupaban tiempo al personal del ISP, mientras que por otro lado este mismo personal continuará pudiendo controlar y crear esos servicios, y además lo hará con los mismos controles y procedimientos que los clientes, con lo que además se estandariza en el ISP los servicios y su gestión.

Respecto a la tarea de diseño del ISP a nivel de conectividad y sistemas, se ha realizado también un diseño con bastantes detalles destacables: el principal es el carácter abierto del diseño (abierto tanto a un crecimiento futuro en el tamaño del ISP como a nuevos protocolos como IPv6 y nuevos servicios que puedan ser implantados). De entre las características que permiten esto, la principal puede que sea el uso de VLAN. Otra característica relevante es la seguridad, en la que se ha incidido de manera especial por ser un ISP el objetivo predilecto de los ataques de *hackers*, dado que sus sistemas permanecen expuestos a cualquier intrusión las 24 horas del día.

Dentro de esta búsqueda de seguridad, hay dos aspectos a destacar, como son la lucha contra el SPAM (tratada ésta de manera muy detallada) y los virus, involucrados estos dos con el servicio de correo. Equilibrar seguridad y funcionalidad es difícil en estos casos, debido a que un planteamiento inadecuado de los mismos puede abortar la entrega de correos legítimos si nuestro sistema los clasifica como virus o *spam*.

Todas estas tareas de configuración de los sistemas constituyen una labor de ingeniería que se sale del estilo tradicional, basado en el desarrollo de software concreto. Este proyecto ha tratado de rescatar este tipo de tareas para que sean valoradas como merecen (sobretodo de análisis y diseño), valiéndose para ello en un enfoque más global.

El proyecto ha sido provechoso por la batería de conocimientos adquirida, dada mi orientación laboral hacia el área de Redes: los conocimientos técnicos que intervienen en la gestión de un ISP son extensos.

En un ISP de tamaño reducido como el que se ha planteado en este proyecto no se puede delegar cada servicio en un departamento o servidor diferente, luego el abanico de tecnologías que el administrador de sistemas de un pequeño ISP tiene que conocer es amplio, reto que hemos tenido que asumir para poder desarrollar el proyecto.

Como además de la gestión del ISP, se ha trabajado sobre el diseño de software para el mismo, y también sobre el propio diseño de los sistemas del ISP, los campos sobre los que hemos trabajado han sido muy amplios: UML, planificación de recursos, programación en lenguajes diversos (PHP, *Python*, *shell*, etc.), configuración de servicios y de Redes, etc.

Igualmente importante es el grado de conocimiento adquirido sobre el sector de las Nuevas Tecnologías basadas en Internet, conocimientos demostrados en el capítulo 4 de Antecedentes: es un sector cambiante y que ha sufrido una fuerte crisis a partir del 2.000 que no se podía pasar sin describir.

Esta crisis ha hecho que el número de ISP se haya ido reduciendo a pasos agigantados, llevando a pensar que en poco tiempo sólo quedarían las grandes operadoras. Pensar que esto es debido a que con sólo un tamaño elevado del ISP se puede mantener los sistemas y las conectividades que un ISP necesita es un error, ya que el problema ha sido más bien los excesos cometidos por la mayoría de empresas del sector.

9.2 Futuros servicios

Los servicios implementados en el proyecto constituyen habitualmente los servicios que podemos encontrar en cualquiera de los ISP e IPP de nuestro alrededor.

Si el ISP fuera capaz de implementar además algún servicio distinto a estos que llamaremos “tradicionales”, la diferenciación que con ello obtendría le valdría una imagen y crecimiento a buen seguro importantes. De ahí que una acción recomendable sería que el departamento de Sistemas del ISP disponga en su jornada laboral de tiempo y recursos para investigar en estas líneas, una vez implantados los servicios esenciales que en el proyecto y hasta este momento habíamos ido nombrando (DNS, Web y correo electrónico).

Por todo ello, y aunque sólo sea brevemente, veremos ahora alguno de los posibles servicios que podría ofrecer este ISP, servicios que ya están disponibles en muchos casos en otros operadores.

VoIP

Voz sobre IP (*Voice over IP*) es el significado de VoIP, usado para referirse a un estándar ampliamente testeado y que cada día se puede encontrar en más redes corporativas, que usan Internet para transmitir mediante paquetes IP la señal de voz digitalizada que de otra manera debería utilizar y pagar las redes telefónicas.

Las mismas operadoras de voz en la actualidad usan fibra óptica para las comunicaciones de larga distancia, usando por tanto en muchas ocasiones conmutación de paquetes para transmitir la información a través de esos canales de fibra de manera mucho más optimizada que si lo hicieran mediante conmutación de circuitos. Pero con una Internet generalizada en todos los puntos del planeta es fácil comunicar directamente con el destinatario, sin pasar por la operadora.

Siendo este tipo de servicios más adecuados para operadores, a nosotros como ISP dedicado a ofrecer únicamente servicios y no conectividad nos puede interesar suministrar a los clientes servicios como centralitas telefónicas software, basadas en la interacción entre un servidor conectado directamente a la línea telefónica y terminales de VoIP conectados a la red local, todo a través del cable Ethernet.

Otra posibilidad sería ofrecer al cliente un buzón de voz alojado en uno de nuestros servidores y un número telefónico al que la gente llamaría. Estos mensajes podrían ser luego retransmitidos mediante el correo electrónico, o recuperados posteriormente desde un sitio Web, e incluso con un software de reconocimiento de voz extraer el texto y mandarlo por correo. Tanto en este caso como en el anterior, el servidor necesitaría de una tarjeta de sonido para digitalizar la voz a grabar o a transmitir mediante VoIP, y habría que partir de un hecho aún hoy por hoy difícil de explicar al cliente: cuando salimos de la red Ethernet la

calidad cae de forma tal que es muy probable que en un momento dado se produzcan cortes.

Servicio de fax

Existen en la actualidad servicios que permiten realizar a través de Internet el envío de faxes a un coste mucho menor que el envío directo a través de la línea telefónica. Básicamente la idea de funcionamiento es la misma que para la VoIP, pero lo que se remitiría por correo o se guardaría en un sitio Web sería el fax recibido.

Servicio de base de datos

Hasta ahora se ha previsto crear sitios Web en los que de manera opcional pueda haber bases de datos. Una opción sería lo contrario: que para algún cliente lo opcional sea diseñar un sitio Web. En concreto estaríamos hablando de la posibilidad de que un cliente solicite de nosotros la configuración de una base de datos que luego él utilice de manera remota.

Servicio de copias de seguridad

Existiendo como existe un horario valle (aquel en el que no hay apenas actividad), que va desde las doce de la noche hasta las ocho de la mañana, horario en el que ningún servidor registra apenas tráfico, resultaría interesante encontrar la forma de mejorar nuestra eficiencia buscando actividades que el ISP pueda realizar en esos horarios, y que no se limiten a tareas administrativas internas del propio ISP.

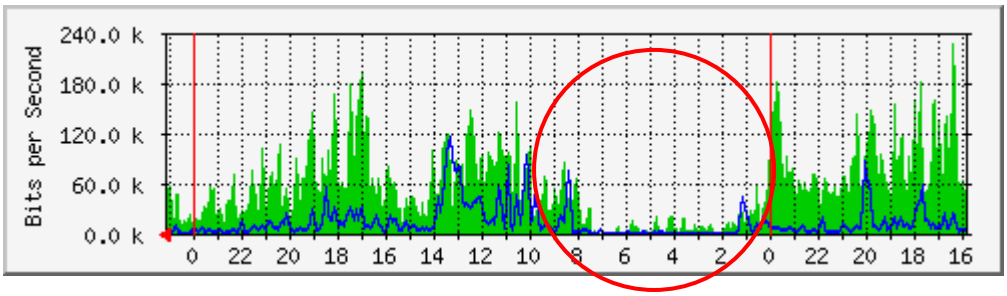


Ilustración 9-1: Horas valle en el acceso a Internet

Una posibilidad para esto es usar nuestra conectividad para facilitar a los clientes la colocación en una localización remota de una copia de sus datos más vitales, que permita que en caso de catástrofe grave (como un incendio) no se pierdan todas las copias de la información (que de manera habitual muchas veces se guarda en el mismo lugar, sin pensar que las copias de seguridad pueden también dañarse en la misma acción que destruye la fuente real de la información).

Estas copias requerirían de la instalación de un programa en el cliente que las controle, o bien simplemente un servidor FTP al que únicamente esté permitido entrar en este horario valle de tráfico que buscamos rentabilizar. Las copias deberían estar protegidas mediante

contraseña, para de esta forma incrementar la confianza del cliente en el servicio.

Data Center

En el diseño de la red del ISP se ha tenido en cuenta que en el futuro pueda haber más de un servidor. Si bien esta opción estaba pensada para alojar más servidores propiedad de la empresa, es perfectamente viable utilizar esta red de servidores opcional para ofrecer servicios de alojamiento de servidores a los clientes, controlando siempre el ancho de banda y sobretodo filtrando el tráfico hacia esas máquinas desde nuestro servidor principal.

Un Data Center pone a disposición del cliente el control de una máquina independiente, aunque manteniéndose siempre la posibilidad de monitorizar y controlar dicho servidor por parte del ISP. Existen tres posibles servicios que nuestro ISP podría suministrar:

- *Hosting compartido*, en un servidor suministrado por el ISP y en el que hay alojados múltiples clientes bajo la misma IP. Del ISP es la configuración y mantenimiento del servidor.
- *Hosting dedicado,* en el que la máquina sigue siendo del ISP, y los servicios suministrados también, pero el servidor es dedicado a un cliente en concreto.
- *Housing* (también conocido como *collocation*), con servidores suministrados por el cliente, cumpliendo unos requisitos mínimos que el ISP habrá de especificar.

Estas fórmulas responden a la tendencia entre las empresas de exteriorizar la gestión de sus equipos informáticos, y de hecho nuestro ISP será usuario de estos servicios, por contratarlos de un tercero para alojar su servidor de backup.

La tendencia también es a que los costes de la fórmula nombrada que en principio es más cara (el *hosting* dedicado) sean cada vez menores debido a la existencia de tecnologías que permiten en un solo equipo físico tener en ejecución diferentes. Esta tecnología se basa en simuladores en los servidores, y mediante el uso de servidores privados virtuales[61] lograr así que las ejecuciones de múltiples sistemas operativos en un mismo servidor sean totalmente independientes entre sí.

[61]Gélinas, Jacques. *Virtual Private Servers and security contexts* (2.003): <http://www.solucorp.qc.ca/miscprj/s_context.hc?s1=2&s2=0&s3=0&s4=0&full=0&prjstate=1&nodoc=0>

10 Bibliografía

Al margen de las notas al pie de página del presente documento, existen una serie de obras que han sido consultadas de manera reiterada para elaborar esta memoria, y que citaremos en orden alfabético, incluyéndose aquí tanto obras técnicas o libros de referencias de lenguajes y protocolos concretos, como los ensayos de tipo histórico o legal que han sido usados en el Capítulo 4 de Antecedentes.

Almirón, Nuria (2.001): *Cibermillonarios, la burbuja de Internet en España.* Editorial Planeta (ISBN 84-08-03675-0).

Boehm, Barry W (2.000): *Software cost estimation with COCOMO II.* Editorial Prentice Hall (ISBN 0-130266-92-2).

Boney, James (2.001): *Cisco IOS.* Editorial O'Reilly (ISBN 1-56592-942-X)

Consentino, Christopher (2.001): *Guía Esencial PHP.* Editorial Prentice Hall/Pearson (ISBN 84-205-3326-2).

Jones, Christopher A. (2.002): *Python & XML.* Editorial O'Reilly & Associates (ISBN 0-596001-28-2).

García Moreno, María Antonia (1.999): *De la teledocumentación a Internet.* Editorial Fragua.

Garfinkel y Spafford (1.991): *Practical UNIX and Internet Security.* Editorial O'Reilly (ISBN 0-937175-72-2).

González Encinar, José Juan (2.000): *Derecho de la comunicación.* Capítulos B (Las Telecomunicaciones) y C (Internet). Editorial Ariel (ISBN 84-34430-08-8).

Graham, Ian S. (1.995): *The HTML Sourcebook.* Editorial John Wiley and Sons (ISBN 0-471-11849-4).

Gudavaram, Shishir (2.002): *A guide to CGI programming*. Editorial O'Reilly (ISBN 1-56592-183-6).

Hunt, Craig (2.002): *TCP/IP Network Administration*. Editorial O'Reilly (ISBN 0-596-00297-1).

Jacobson, Ivar, Grady Booch y James Rumbaugh (2.000): *El Proceso Unificado de Desarrollo de Software*. Editorial Addison-Wesley. (ISBN 84-7829-036-2).

López Garrido, Diego (1989): *La crisis de las telecomunicaciones:* El fenómeno desregulador en Estados Unidos, Japón y Europa. Editorial Fundesco (ISBN 84-86094-49-6).

Martelli, Alex (2.003): *Python in a Nutshell*. Editorial O'Reilly. (ISBN 0596001886).

Muñoz Machado, Santiago (2.000): La regulación de la red, poder y derecho en Internet. Editorial Taurus (ISBN 84-306-0415-4).

Nombela, Juan José (1.996): Seguridad Informática. Editorial Paraninfo (ISBN 84-283-2341-1).

Pearce, Ferry y otros (2.002): *La Transformación Empresarial en la era de Internet*. Editorial Paidos Ibérica (ISBN 9501210871).

Shah (2.001): *Manual de administración de Linux*. Editorial Osborne/McGraw Hill (ISBN 84-481-2892-3).

Stein, Lincoln (1.995): *How to Setup and Maintain a World Wide Web Site: The Guide for Information Providers*. Editorial Addison-Wesley (ISBN 0-201-63389-2).

Tanenbaum, Andrew S. (4ª edición, 2.003): *Computer Networks*. Editorial NJ Pearson Education (ISBN 0-133-94248-1).

11 Índice

En este índice se han unido tanto los términos correspondientes a tecnologías, como los nombres de empresas y personas citados a lo largo de la memoria, encontrándose ordenados por orden alfabético.