

E80 PROYECTOS INFORMÁTICOS
INGENIERÍA INFORMÁTICA

Desarrollo de un ISP

1. Objetivos
2. Descripción general
3. Hitos más relevantes
4. Conclusiones

<http://www.bith.net/proyecto>
Héctor Castillo Andreu
18 de Julio del 2.003

1. Objetivos



- Determinar qué es hoy en día un ISP
 - Ver la evolución de Internet y del sector de los ISP
 - Analizar por qué a un ISP pequeño no le es rentable ofrecer acceso a Internet
 - Determinar qué servicios puede ofrecer el ISP en Internet que no sean acceso a Internet
- Desarrollar el ISP
 - Determinar los recursos necesarios
 - Configurar los sistemas
 - Desarrollar el software a medida para ofrecer los servicios del ISP

2. Descripción general



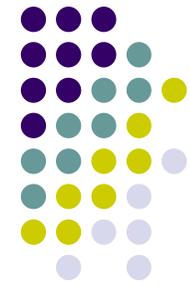
- El acceso a Internet
 - Rentable con grandes economías de escala
 - Pequeño ISP depende de otros: se convierte en revendedor
 - Acceso a Internet actualmente en manos grandes operadoras
- IPP = Internet Presence Provider
 - Ofrece alojamiento Web y correo
 - Desvinculamos acceso a Internet y los servicios que siempre tenía aparejados, como el correo
 - Pequeño ISP bien diseñado con costes similares a un gran operador en servicios Web y correo

3. Hitos más relevantes

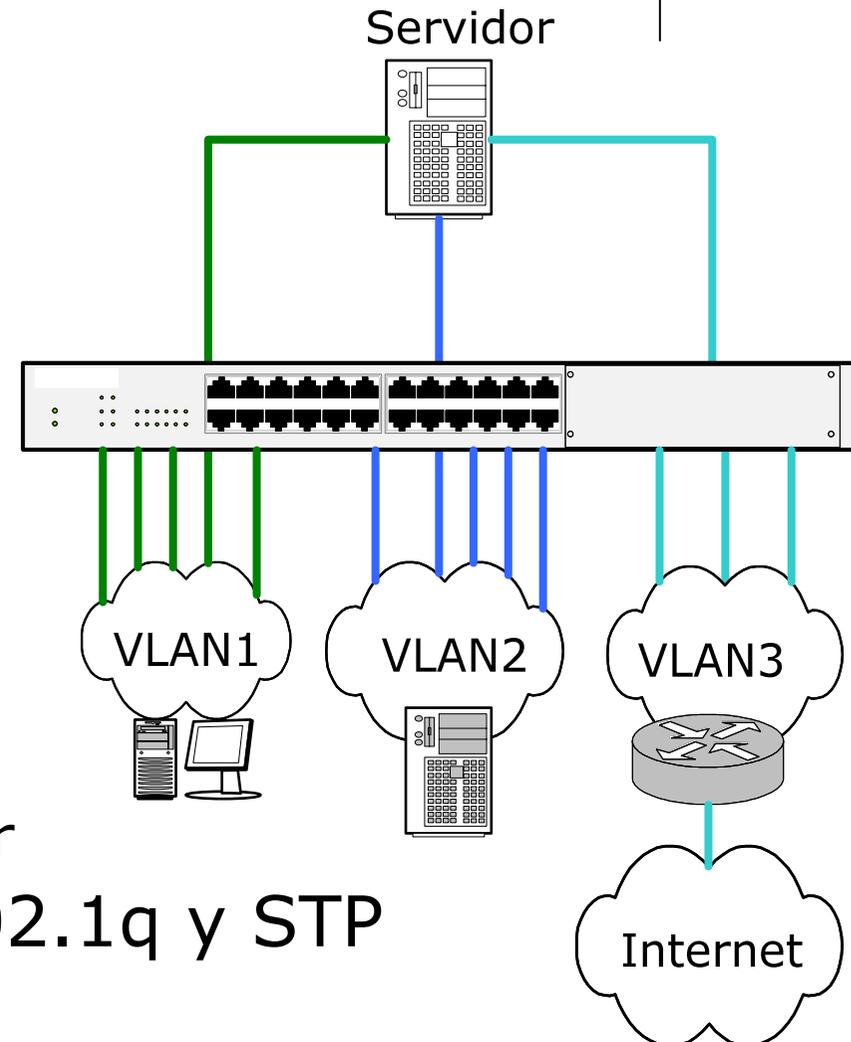


- Diseño del ISP
 - 3.1. Red local usando VLAN
 - 3.2. Enrutamiento con múltiples conectividades
 - 3.3. DNS y vistas múltiples
 - 3.4. La lucha contra el SPAM
- Software de gestión
 - 3.5. COCOMO y UML
 - 3.6. La interfaz Web
 - 3.7. Documentación adecuada

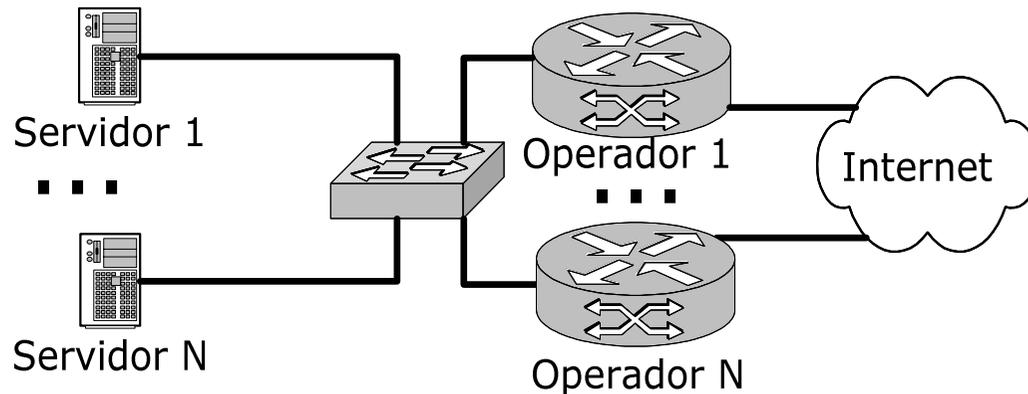
3.1. Red local usando VLAN



- Con sólo un switch dar servicio a tres redes distintas
- Puertos del switch asignados a cada red
 - Seguridad (no se ven entre sí)
 - Escalabilidad y eficiencia (stack de switches)
 - El servidor hace de router
- RFC 3069: protocolo 802.1q y STP



3.2. Enrutamiento con múltiples conectividades



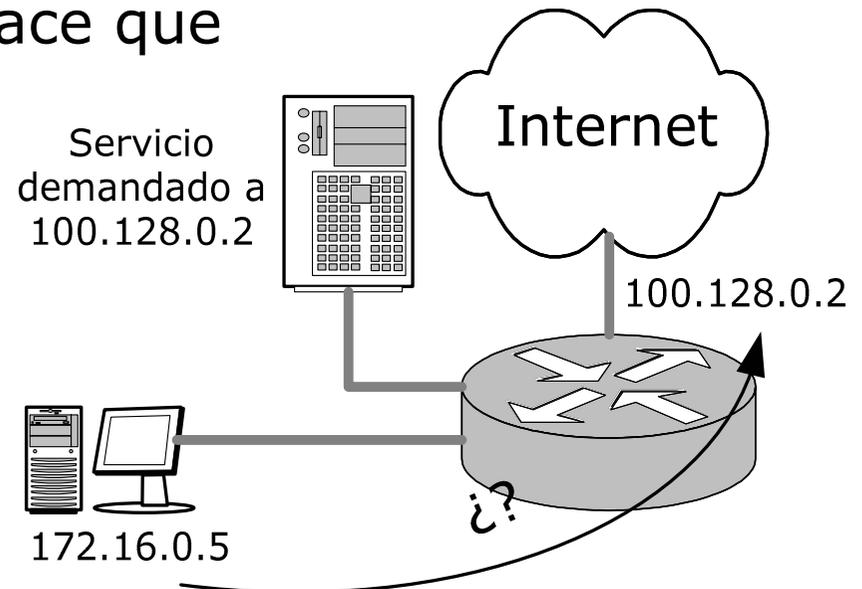
- Mismo host tendrá múltiples IPs
- DNAT en algunas conexiones

- Múltiples tablas enrutamiento
 - Cada IP requiere una tabla
 - Servidor Linux hace de router: puede gestionar tráfico de otros hacia conexión más adecuada
 - Requiere iproute2
- Al ser router, aplicaremos también QoS (calidad del servicio) mediante tc

3.3. DNS y vistas múltiples



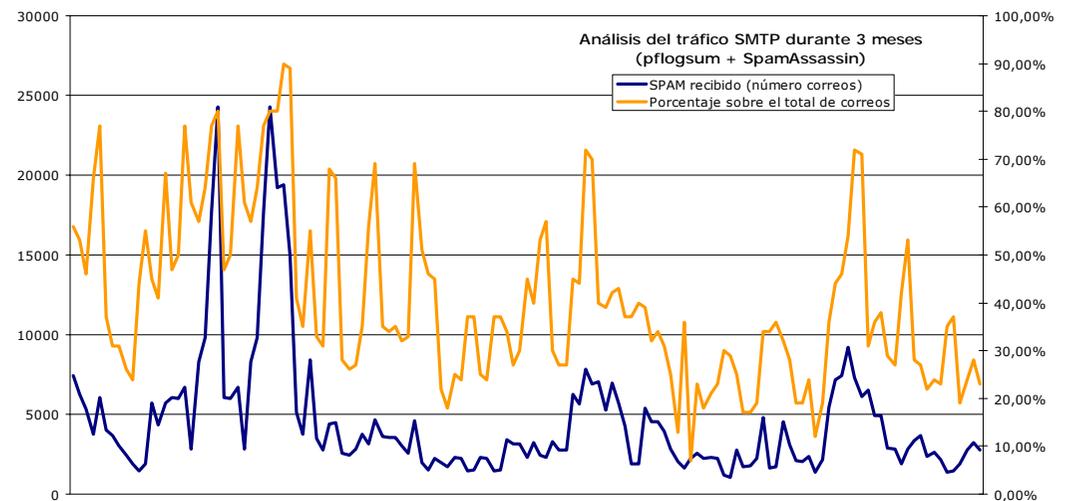
- Servicio sobre IP con DNAT hace que no podamos acceder a esa IP pública desde la propia red privada
- Soluciones:
 - Configurar esa IP en el servidor también
 - Válido si todos los servicios van al mismo servidor
 - Hay que reenrutar antes de que llegue a router que hace DNAT
 - Crear dos vistas de las zonas DNS: una para Internet y otra interna
 - No afecta a accesos desde Internet
 - Evita configurar dos IPs, aunque a cambio dos ficheros de zona



3.4. La lucha contra el SPAM



- El 90% de los usuarios recibe al menos un correo considerado SPAM al día pese a los filtros
- Es un fenómeno creciente
- Supone gran parte del tráfico SMTP
- Difícil de eliminar: las técnicas son superadas por los spammers

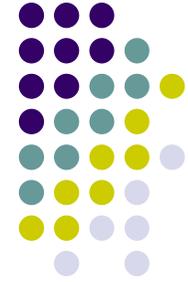


3.5. COCOMO y UML



- Estimación con técnica COCOMO
 - Estimación inicial cumplida
 - Área estratégica en el mundo empresarial
- Modelado UML del software
 - Mayor coste de desarrollo
 - Menor riesgo de errores de diseño
 - Identificación precisa de los requisitos

3.6. La interfaz Web



- Integrar en ella tanto gestión servicios como facturación
- Misma funcionalidad que desde consola pero limitada a los servicios del cliente
- Características:
 - Seguridad
 - Funcionalidad
 - Simplicidad
 - Interacciona con programas que gestionan servicios desde consola



3.7. Documentación adecuada



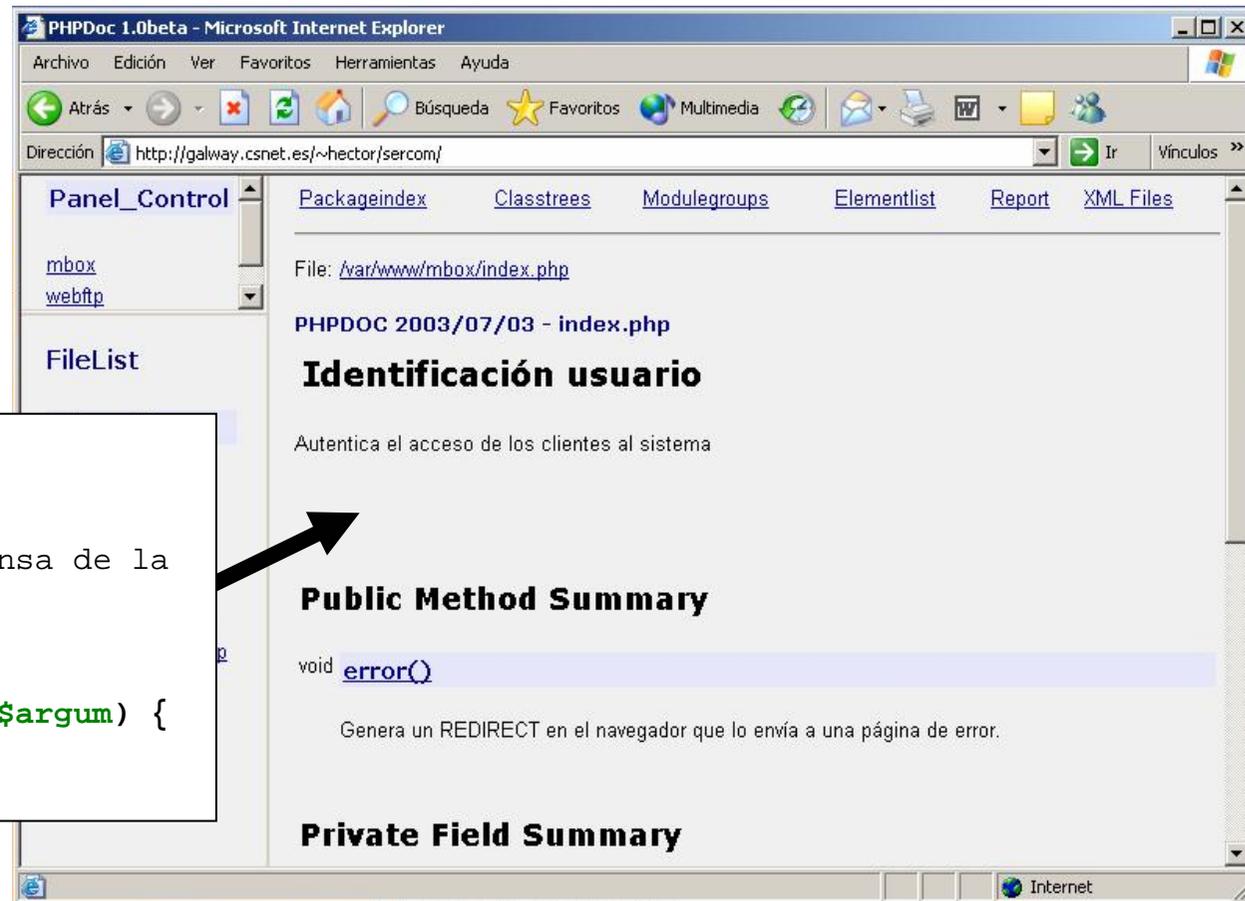
- Herramientas auto-documentación del software
 - Generan HTML con estructura de los objetos, funciones y comentarios de los programadores
 - Pydoc
 - PHPDoc
- Documentación de los sistemas
 - Manual y subjetiva

3.7. Documentación con PHPDoc



- Beta desde el 2.000 (<http://www.phpdoc.de>)
- No es oficial
- Comentarios en el código
- Genera HTML navegables

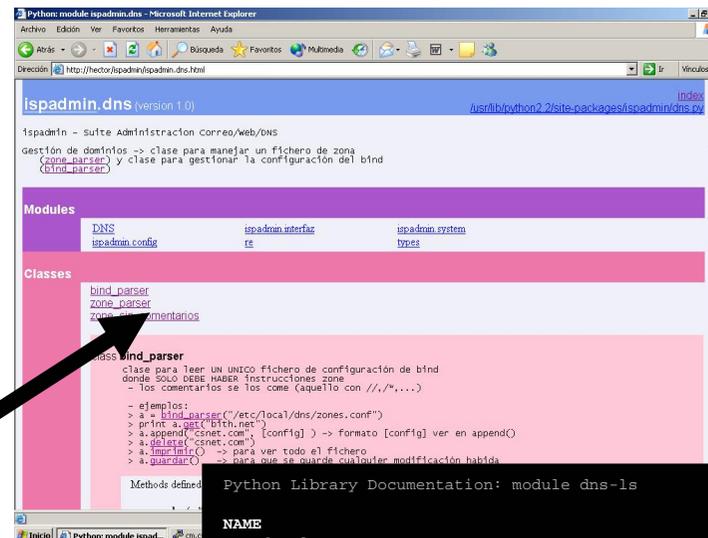
```
/**
 * Descripción
 *
 * Descripción extensa de la
 * función.
 * .....
 */
function prueba(int $argum) {
    .....
}
```



3.7. Documentación con Pydoc



- Totalmente oficial e integrado en Python
- Comentarios en el código
- Genera HTML, páginas man...
- Permite buscar



```
__doc__ = """
descripción extensa del
módulo
"""
def prueba(argum1, argum2):
    """
    descripción de la función
    + argum1 ...
    + argum2 ...
    """
```

```
Python Library Documentation: module dns-ls
NAME
  dns-ls
FILE
  /home/ispadmin/bin/dns-ls.py
DESCRIPTION
  ispadmin - Suite Administración Correo/Web/DNS
  hector@csnet.es
  ---
  administración de la dns
  - ver zonas dns disponibles en esta máquina
  - ver de una zona dns información sobre sus entradas, etc
FUNCTIONS
  main(argv, env)
    lista las zonas y las entradas dns de una zona concreta
DATA
  __ayuda__ = '\n Mostrar información sobre las zonas DNS confi... u...
  :
```

3.7. Documentación Pydoc



info(elem)
obtiene información sobre una cuenta de correo 'elem'
devuelve None si no existe o no es del grupo de los usuarios de correo,
el resultado es la siguiente tupla en caso contrario:
[nombre, grupos, descrip, home, bloqueo, fecha_desactivacion,
cuotas, emails, sender_canonical]

num_emails(user)
devuelve un entero que representa el número de mensajes que hay en el
buzón especificado, o 0 en caso de no existir éste o ser cero

show(elem, recursivo=1, nueva=None)
muestra por pantalla información de la cuenta de correo 'usuario'. si
se indica en 'nueva' una configuración no pregunta al sistema la real,
sino que muestra ésta.
con 'recursivo' muestra además otras direcciones de correo desde las
que llega el email a este usuario de manera indirecta

el array 'nueva' tiene que seguir este formato:
[nombre, grupos, descrip, home, bloqueo, fecha_desactivacion, cuotas, emails]
si alguno de los argumentos es None entonces NO se muestra esa parte

sugerencia_usuario(email)
sugiere un nombre para el usuario a partir de su dirección de correo

Data

```
__er_dominio__ = <_sre.SRE_Pattern object>  
__er_email__ = <_sre.SRE_Pattern object>  
__file__ = 'usr/lib/python2.2/site-packages/ispadmin/mbox.pyc'  
__name__ = 'ispadmin.mbox'  
__version__ = '1.0'  
__namepat__ = <_sre.SRE_Pattern object>  
relay = <ispadmin.mbox.relay_file instance>  
sender_canonical = <ispadmin.berkeleydb.fichero instance>  
virtual = <ispadmin.berkeleydb.fichero instance>
```

Las funciones aparecen agrupadas por clases. Los ficheros están vinculados de manera que la navegación es sencilla.

Aunque es imposible documentar una constante, al menos las enumera

4. Conclusiones



- Gestión técnica de un ISP muy compleja
 - Sector ISP muy cambiante: tecnologías evolucionan rápidamente
 - Estándares y protocolos regidos por las leyes de mercado
 - Seguridad es prioritaria
 - El factor 24x7x365: vigilancia continua consumirá tiempo y recursos
- ISP requiere software a medida
 - Software para gestionar, controlar y facturar al mismo tiempo los servicios
- Hay espacio para ISP pequeños