



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## SERUCE SISTEMA DE ESCANEEO DE RED Y USUARIOS PARA CENTROS EDUCATIVOS (12985 lectures)

Per **Alumnos 2CF\_ASI** del **IES Leopoldo Queipo**, **DES** (<http://fpiqueipo.hopto.org/moodle/>)  
Creado el 18/01/2006 21:43 modificado el 14/02/2006 21:19

*Sistema web dinamico basado en Perl+Apache+Shell para generar paginas html dependiendo de los usuarios del sistema (linux) y los hosts disponibles en la red que ofrezcan determinados servicios. Ademas incluye comportamiento diferente para profesores y alumnos de un centro educativo, con posibilidad para ambos de publicar sus paginas html y compartir ficheros por http, y con la opcion de mensajeria comun por nfs para profesores, cuya autentificacion es cifrada para http por el modulo digest (MD5). Es una practica de la asignatura DES (Desarrollo de funciones en el sistema informatico) de 2º Curso de Administracion de sistemas informaticos, ciclo formativo de grado superior, del IES Leopoldo Queipo, Melilla.*

### SERUCE SISTEMA DE ESCANEEO DE RED Y USUARIOS PARA CENTROS EDUCATIVOS

SERUCE para el I.E.S Leopoldo Queipo, Familia Profesional de Informatica.  
Ciclo Formativo Administracion de sistemas informaticos.  
Ciclo Formativo Explotacion de Sistemas Informaticos.

#### 0.INTRODUCCION

##### 1 ENUNCIADO DEL EJERCICIO

- 1.1 Informacion del alumno
- 1.2 Informacion del profesor
- 1.3 Informacion de las tareas de administracion

##### 2 INSTALACION/MANTENIMIENTO DE APACHE2

- 2.1 Explicacion de la seccion
- 2.2 Script utilizado
- 2.3 Muestra de ficheros plantilla utilizados

##### 3 TAREAS DE ADMINISTRACION

- 3.1 Creacion de los scripts alumnos.pl y profesores.pl para generar una pagina web de inicio con sus datos en cada home
- 3.2 Copia de seguridad del fichero mensajes.txt
- 3.3 Instalar y configurar un servidor dns para nuestra red de area local
- 3.4 Generar un script que ponga en marcha la instalacion del servidor apache con la configuracion ya establecida para nuestro servidor

##### 4 SECCION DE ALUMNOS

- 4.1 Explicacion de la seccion
- 4.2 Script utilizado
- 4.3 Muestra de ficheros plantilla utilizados

##### 5 SECCION DE PROFESORES

- 5.1. Script para la busqueda de ficheros en mensajes.txt
- 5.2. Estructura del fichero mensajes.txt



## 0. INTRODUCCION

### Sistema SERUCE

#### ¿En que consiste?

Es un sistema pensado para institutos aunque aplicable a cualquier sector, este sistema, mediante un simple servidor web, se encarga de diversas tareas. Hace un escaneo de red para detectar que ordenadores estan activos, y nos muestra los servicios que tiene activo. Tambien muestra una lista de los alumnos, solo pasandole un fichero de texto, hace lo mismo con la lista de profesores. A cada usuario se le asigna una carpeta para compartir por http y una pagina web personal. Los profesores tendran acceso a un sistema de mensajes.

#### ¿Que hace falta?

Obviamente, este sistema esta pensado para montarse sobre una maquina linux. En esta maquina debera tener instalado un servidor web, con capacidad de ejecutar scripts CGI de perl. Tambien necesitara tener instalado perl, nmap y corriendo el demonio cron. En nuestro caso, se ha montado el servidor apache, por ser el que mejor resultados nos ofrece y el mas difundido.

#### Ficheros a usar

##### Ficheros de alumnos

```
/var/www/perl/alumnos/alumnos.pl  
/var/www/perl/alumnos/alumnos.txt  
/var/www/images/f.jpg  
/var/www/images/m.jpg  
/var/www/perl/alumnos/opciones.pl
```

##### Ficheros de profesores

```
/var/www/perl/profes/profesores.pl  
/var/www/perl/profes/opciones_profe.pl  
/var/www/perl/profes/profesores.txt  
/var/www/perl/profes/passwd_digest  
/var/www/images/pf.jpg  
/var/www/images/pm.jpg
```

##### Ficheros de mensajes de profesores

```
/var/www/perl/profes/mensajes/anadir_mensaje.pl  
/var/www/perl/profes/mensajes/buscar_mensajes.pl  
/var/www/perl/profes/mensajes/copia_respaldo.pl  
/var/www/perl/profes/mensajes/mensajes.txt  
/var/www/perl/profes/mensajes/mostrar_mensaje.pl
```

##### Ficheros para los equipos

```
/var/www/perl/red/equipos.pl  
/var/www/perl/red/servicios.pl  
/var/www/perl/red/ordenatas  
/var/www/images/Firefox-logo.png  
/var/www/images/windows_logo.gif  
/var/www/images/ssh.jpg  
/var/www/images/Ftp.jpg  
/var/www/images/133t_DEV_nfs_mount.png  
/var/www/images/pc.jpg  
/var/www/images/impresora_ico.jpg
```



## Ficheros para instalacion

`/var/www/perl/instalacion/inst_alumnos.pl``/var/www/perl/instalacion/inst_profes.pl`

## Ficheros de index y documentacion

`/var/www/html/index.html``/var/www/images/logo.jpg``/var/www/html/scripts.html`

## Instalacion

## Modificaciones al httpd.conf

Para las imagenes de seruce, se crea un directorio `/var/www/images`; en `/var/www/images`. Es necesario indicar esto en el `httpd.conf` con la siguiente clausula.

```
Alias /images/ /var/www/images/
<Directory "/var/www/images">
AllowOverride None
Options -Indexes FollowSymLinks MultiViews
<IfModule mod_access.c>
Order allow,deny
Allow from all
</IfModule>
</Directory>
```

Sera necesario tener instalado el modulo `mod_userdir` para poder poner las directivas que nos daran acceso sin problemas a los directorios personales.

Las directivas que hay que poner son las siguientes

```
UserDir "/seruce/*/public_html"
```

```
<Directory /seruce/*/public_html>
AllowOverride None
Options MultiViews -Indexes Includes FollowSymLinks
<IfModule mod_access.c>
Order allow,deny
Allow from all
</IfModule>
</Directory>
```

```
<Directory "/seruce/*/public_html/publico">
Options Indexes
DirectoryIndex algo.sec
<IfModule mod_access.c>
Order allow,deny
Allow from all

</IfModule>
</Directory>
```

## CONTENIDO DE leeme.txt:

SERUCE  
SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE  
SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE SERUCE

## INSTALACION

-----



1. --- Deberas tener instalado apache, con apache-mod\_perl,mod\_userdir y perl-CGI.
2. --- Añadir las directivas que vienen en httpd.conf\_install
3. --- Ejecutar el script de instalacion

CONTENIDO DE instalar.sh

```
#!/bin/bash

tar -xvPf seruce_v0.1_mdk2k6.tar
perl /var/www/perl/instalacion/inst_alumnos.pl 2> /dev/null
perl /var/www/perl/instalacion/inst_profes.pl 2> /dev/null
chown apache:apache /var/www/ -R
chmod 775 /seruce/ -R

cat httpd.conf_install >> /etc/httpd/conf/httpd.conf
/etc/init.d/httpd restart
echo -e "0\t 23\t 1\t *\t *\t /admin/departamento/admin/scripts_shell/copia_mensaje.sh" >> /var/spool/cron/root

/etc/init.d/crond restart
```

## 1 ENUNCIADO DEL EJERCICIO

Desarrollo de funciones con Perl.

- 1.1 Informacion del alumno
- 1.2 Informacion del profesor
- 1.3 Tareas de administracion

Se pretende instalar y configurar un servidor web de intranet en un centro educativo, en el que el uso de ordenadores es muy frecuente. De ahí que se planteen tareas a resolver para un mejor entendimiento de la red local y aprovechamiento de sus recursos, que van a ser mejor visualizados mediante paginas web.

El sistema que albergara el servidor http (apache2) sera Linux Mandriva 2006, y que funcionara tambien como servidor de ficheros a los alumnos y profesores del centro. Para esta practica nos valdremos de scripts en perl para generar automaticamente paginas web dependiendo de la disposicion en ese instante de la red y sus recursos. Tambien pueden utilizarse scripts en shell, y programacion de tareas con cron. Evidentemente necesitaremos manejar codigo HTML dentro de los scripts en perl, pero de una manera basica.

La configuracion http forzara a que la informacion devuelta por el servidor varie segun el usuario logeado contra el servidor mediante login y contraseña. Se devolvera informacion atendiendo a si el usuario es alumno o profesor del centro.

La informacion a la que un alumno tendra acceso sera:

- Ordenadores conectados a la red en ese instante, mostrando tanto su nombre de dominio interno de DNS como su direccion IP.
- Impresoras de red conectadas, idem.
- Paginas web de todos los alumnos ubicados en /home del servidor, es decir en los directorios /home/usuarios/public\_html.



La informacion a la que un profesor tendra acceso sera:

- Toda la informacion a la que un alumno tiene acceso.
- Paginas web de todos los profesores ubicados en /admin del servidor, es decir en los directorios /admin/usuarios/public\_html.
- Un apartado de mensajes reservado solo a profesores para comunicaciones internas entre ellos. Este sistema de mensajes esta basado en enlaces simbolicos sobre ficheros montados mediante NFS ubicado tambien en el servidor. El formato de estos ficheros sera "usuario:fecha:asunto[:asunto]", y se pretende que se tenga acceso a los mensajes escritos atendiendo a diferentes busquedas tanto por usuario, por fecha, o por asunto.

Ademas de estas funciones desarrolladas con perl para controlar el acceso a la informacion del servidor web, se necesita realizar varias tareas de administracion del sistema usando scripts tanto en perl como shell, que son:

- Crear automaticamente una pagina web de inicio en cada /home/usuarios/public\_html/index.html y /admin/usuarios/public\_html/index.html, con la informacion justa pero decorosa en el estilo de diseño web. Para ello se dispondran de fotografias de tipo carnet de cada usuario, asi como de ficheros de texto con la siguiente sintaxis "nombre\_apellidos:login:descripcion:fichero\_foto".

Crear una tarea programada con cron para realizar copias de seguridad de los mensajes de profesores cada mes. Hacer retoques en el dominio creado para el centro del servidor DNS, para cumplir con la resolucion de la practica.

## 2. iNSTaLaCioN Y MaNTENiMieNTTo De aPaChE2

### 2.1 Explicacion de la seccion

### 2.2 Script utilizado

### 2.3 Muestra de ficheros plantilla utilizados

### 2.1 EXPLICACION DE LA SECCION

Para empezar, comenzaremos con la instalacion de Apache Server. Con el comando en una consola:

```
# urpmi apache
```

Despues termina la instalacion.

Probaremos el estado del demonio con le comando:

```
/etc/init.d/httpd status
```

Bueno ya tenemos terminada la instalacion de Apache Server en nuestro ordenador, para comprobar, lo que haremos es:

Abre tu explorador de Internet y teclea `http://localhost/`

A continuacion procederemos con la instalacion de perl. Con el comando en una consola:

```
Urpmi perl-CGI
```

Ejemplo:

Una vez completada instalacion y configuracion del servidor apache para que funcionen los scripts de perl, comenzaremos a crear nuestros scripts. Para que funcionen correctamente, deben de guardarse en la carpeta



/var/www/perl, que es el lugar donde se pueden ejecutar los scripts CGI de perl.

Hemos hecho un programa en shell que lo que hace es crear un grupo llamado profes y dentro le añade a todos los profesores que hay en el fichero profesores.txt y a su vez tambien crea una carpeta llamada /admin y le da unos permisos y propietario de manera recursiva de forma que solo pueden entrar los usuarios que haya dentro del grupo profes.

## 2.2 SCRIPT UTILIZADO

Script utilizado para dar de alta a los administradores altas\_administradores.sh:

```
#!/bin/bash
#altas_administradores.sh
#modo ejecucion: altas_administradores.sh /admin profesores.txt
#1? sino existe el directorio /admin lo creara y le pondra permisos 660 y propietarios root.profes
#2? Con el segundo fichero lo que hara es cortar el nombre de usuario e ir a?adienlo a esos nombres al grupo
profesores.

cat /etc/group | grep -q profes
if [ $? = 0 ]
then
echo "el grupo ya esta creado"
else
echo "el grupo aun no esta creado. se creara a continuacion..."
groupadd profes
fi

if [ -d /admin ]
then
echo "el directorio ya esta creado"
echo "aplicanco permisos..."
chown -R root.profes /admin
else
echo "el directorio aun no esta creado. se creara a continuacion..."
mkdir /admin
echo "aplicanco permisos..."
chown -R root.profes /admin
fi

fichero="/admin/departamento/admin/textos/profesores.txt"

while read linea
do
user=$(echo $linea | cut -f2 -d:)
useradd -G profes -d /admin/$user $user
done < $fichero

chmod -R 660 /admin
```

Otro script que hemos utilizado es el de altas.sh, que da de alta a una listas de usuarios existentes en un fichero pasandose lo como parametro:

```
#!/bin/bash

#llamada al script: ./altas fichero_usuarios
#fichero_usuarios tiene la siguiente sintaxis
#user:passwd_no_encrip:coment

while read linea
do
```



```

user=$(echo $linea | cut -f1 -d:)
clave=$(echo $linea | cut -f2 -d:)
coment=$(echo $linea | cut -f3 -d:)
useradd -c "$coment" $user
echo "$clave" | passwd $user --stdin
done < $1

```

### 2.3 MUESTRA DE FICHEROS PLANTILLA UTILIZADOS

Informacion adicional:

Para ejecutar el fichero para dar de alta a usuarios hemos rellenado ficheros con el nombre de usuario y contraseña de cada uno para que mediante el script los de de alta uno a uno. El fichero tiene este formato:

```

gs2_1:queipo:Administracion de Sistemas Informaticos
gs2_2:queipo:Administracion de Sistemas Informaticos
gs2_3:queipo:Administracion de Sistemas Informaticos
gs2_4:queipo:Administracion de Sistemas Informaticos
gs2_5:queipo:Administracion de Sistemas Informaticos
gs2_6:queipo:Administracion de Sistemas Informaticos
gs2_7:queipo:Administracion de Sistemas Informaticos
gs2_8:queipo:Administracion de Sistemas Informaticos
gs2_9:queipo:Administracion de Sistemas Informaticos
gs2_10:queipo:Administracion de Sistemas Informaticos
gs2_11:queipo:Administracion de Sistemas Informaticos
gs2_12:queipo:Administracion de Sistemas Informaticos
gs2_13:queipo:Administracion de Sistemas Informaticos
gs2_14:queipo:Administracion de Sistemas Informaticos
gs2_15:queipo:Administracion de Sistemas Informaticos
gs2_16:queipo:Administracion de Sistemas Informaticos
gs2_17:queipo:Administracion de Sistemas Informaticos
gs2_18:queipo:Administracion de Sistemas Informaticos
gs2_19:queipo:Administracion de Sistemas Informaticos
gs2_20:queipo:Administracion de Sistemas Informaticos
gs2_21:queipo:Administracion de Sistemas Informaticos
gs2_22:queipo:Administracion de Sistemas Informaticos
gs2_23:queipo:Administracion de Sistemas Informaticos
gs2_24:queipo:Administracion de Sistemas Informaticos
gs2_25:queipo:Administracion de Sistemas Informaticos
gs2_26:queipo:Administracion de Sistemas Informaticos
gs2_27:queipo:Administracion de Sistemas Informaticos
gs2_28:queipo:Administracion de Sistemas Informaticos
gs2_30:queipo:Administracion de Sistemas Informaticos

```

### 3. TaReaS De aDMiNiSTRaCioN

3.1 Creacion de los scripts alumnos.pl y profesores.pl para generar una pagina web de inicio con sus datos en cada home

3.2 Copia de seguridad del fichero mensajes.txt

3.3 Instalar y configurar un servidor dns para nuestra red de area local

3.4 Generar un script que ponga en marcha la instalacion del servidor apache con la configuracion ya establecida para nuestro servidor

#### 3.1 CREACION DE LOS SCRIPTS ALUMNOS.PL Y PROFESORES.PL PARA GENERAR UNA PAGINA WEB DE INICIO CON SUS DATOS EN CADA HOME

Vamos a crear una pagina web de inicio, en cada /home/nombre\_usuario/public\_html/index.html y /admin/nombre\_profesor/public\_html/index.html, con fotografia tipo carnet de cada usuario, y sacando la informacion de un fichero de texto que tiene que tener la siguiente sintaxis:



"nombre\_apellidos:login:descripcion:fichero\_foto"

Nos hemos creado dos scripts, en uno creamos los index.html de los profesores y en el otro el de los alumnos.

Descripcion del Script alumnos.pl

Comenzamos abriendo el fichero de texto con un manejador que contiene los nombres de los alumnos.

El fichero alumnos.txt esta compuesto por una serie de campos que son nombre y apellido,login de usuario,descripcion,fichero de la fotografia. Todos estos campos estan delimitados por ":".para luego poder sacarlos como variables en nuestro script.

Con un bucle hasta que el vector @lineas este vacio diferenciaremos los campos que queremos mostrar mas adelante, el delimitador seran los ":" y los campos que obtendremos seran: nombre, login, descripcion y el nombre de la foto. Para diferenciar los campos utilizaremos el comando split del siguiente modo:

```
@campos = split (/:,$lineas[$i]); -> siendo lineas el vector del fichero
```

Posteriormente crearemos la carpeta public\_html dentro del home del alumno y abriremos un fichero de escritura:

Sin salirnos del bucle vamos a generar dentro de los ficheros html toda la informacion del alumno de forma clara y decorosa.

script alumnos.pl :

```
#!/usr/bin/perl
# Carga el modulo CGI, para poder ejecutar los scripts por pagina güeb
use CGI qw/:standard/;
# Se abre el fichero de texto con el contenido de los alumnos
open(FILE,'/var/www/perl/alumnos/alumnos.txt');
@fichero=<FILE>;
close(FILE);
# Imprimimos la cabecera y el principio del documento de html
print header,start_html('Lista de Alumnos');

print "<table border=0 cellpadding=0 cellspacing=0 align=center>";
print "<tr><td><img src=../images/logo.jpg width=140 height=140 align=right></td>";
print "<td><em><font size=6>SERUCE <font size=3>SISTEMA DE ESCANEADO DE RED Y USUARIOS PARA
CENTROS EDUCATIVOS<br></em></font>";
print "<font size=3>SERUCE para el I.E.S Leopoldo Queipo, Familia Profesional de Informatica<br></font>";
print "<font size=3>Ciclo Formativo Administracion de sistemas informáticos.<br></font>";
print "<font size=3>Ciclo Formativo Explotación de Sistemas Informaticos</font></td></tr></table>";

print "<table cellspacing=6 border=2 align=center>";
# Variable contador para las tablas
$j=0;

$ruta="opciones.pl";
```



```

# Recorremos las lineas del fichero
foreach $linea (@fichero)
{
# Separamos los campos del fichero
($nombre, $usuario, $descripcion, $imagen) = split(/:/,$linea);
# Si una linea tiene curso1, imprime la cabecera
if ($nombre eq "curso1")
{
print "<tr><td colspan=5><div align=center> ...: GRADO SUPERIOR 1º CURSO ... </div> </td></tr>";
$j=0;
# Si una linea tiene curso2, imprime la cabecera
}elsif ($nombre eq "curso2"){
print "<tr><td colspan=5><div align=center> ...: GRADO SUPERIOR 2º CURSO ... </div></td></tr>";
$j=0;
}
# Si una linea tiene medio1, imprime la cabecera
elsif ($nombre eq "medio1"){
print "<tr><td colspan=5><div align=center> ...: GRADO MEDIO 1º CURSO ... </div></td></tr>";
$j=0;
}
# Si una linea tiene medio2, imprime la cabecera
elsif ($nombre eq "medio2"){
print "<tr><td colspan=5><div align=center> ...: GRADO MEDIO 2º CURSO ... </div></td></tr>";
$j=0;
}
# Si no es ninguna de las anteriores, es informacion sobre un alumno
else{
# Comprobacion para las tablas
if ($j % 2 eq 0 )
{
print "<tr><td>";
}
else{
print "<td>";
}
}

$j = $j + 1;
# imprime la informacion del alumno y las ruta para el enlace a sus paginas personales
print "<a href=$ruta?$usuario><img src='.././images/$imagen' height=62 width=62></a>", # Pone la imagen
"</td><td><div align=left>","<a href=$ruta?$usuario>", # Alinea el texto a la izq.
"<font size=2>",br,
# Crea un vinculo a info.pl pasandole como parametro la variable $ip
$nombre,br, # Muestra el nombre
$usuario,br,# Muestra el nombre de usuario
$descripcion, # Muestra la clase a la que asiste y su grupo
"</font></div></td>","</a>";

}
}
print "</td></tr></table>";

```

Descripcion del script profesores.pl

Comenzamos abriendo el fichero de texto con un manejador que contiene los nombres de los profesores.



El fichero profesores.txt esta compuesto por una serie de campos que son nombre y apellido, login de usuario de ese profesor, descripcion, fichero de la fotografia. Todos estos campos estan delimitados por ":".  
Igualmente con un bucle hasta que el vector @lineas este vacio diferenciaremos los campos que queremos mostrar mas adelante, el delimitador seran los ya conocidos ":" y los campos que obtendremos seran el nombre, login, descripcion y el nombre de la foto.

Para diferenciar los campos utilizaremos el ya conocido comando split del siguiente modo:

```
@campos = split (/:,$lineas[$i]); -> siendo lineas el vector del fichero
```

Posteriormente crearemos la carpeta public\_html dentro de admin de cada profesor y abriremos un fichero de escritura:

Sin salirnos del bucle volveremos a generar dentro de los ficheros html toda la informacion del profesor de la misma forma que el script de alumnos.

script profesores.pl :

```
#!/usr/bin/perl
# Carga el modulo CGI, para poder ejecutar los scripts por pagina güeb
use CGI qw/:standard/;

# Abre el fichero con la informacion de los profesores
open(FILE,'/var/www/perl/profes/profesores.txt');
@fichero=<FILE>;
close(FILE);
# Cabecera y comienzo del html
print header,start_html('Lista de Profesores');

print "<table border=0 cellpadding=0 cellspacing=0 align=center>";
print "<tr><td><img src=../images/logo.jpg width=140 height=140 align=right></td>";
print "<td><em><font size=6>SERUCE <font size=3>SISTEMA DE ESCANEADO DE RED Y USUARIOS PARA
CENTROS EDUCATIVOS<br></em></font>";
print "<font size=3>SERUCE para el I.E.S Leopoldo Queipo, Familia Profesional de Informatica<br></font>";
print "<font size=3>Ciclo Formativo Administracion de sistemas informáticos.<br></font>";
print "<font size=3>Ciclo Formativo Explotación de Sistemas Informaticos</font></td></tr></table>";

print "<table cellspacing=6 border=2 align=center>";
# Contador para las tablas
$j=0;
$ruta="opciones_profe.pl";
print "<tr><td colspan=5><div align=center> ...: PROFESORES :... </div></td></tr>";
# Se recorre linea por linea el fichero
foreach $linea (@fichero)
{
# Se dividen los campos con la informacino de cada profesor
($nombre, $usuario, $descripcion, $imagen) = split(/:,$linea);

# Contador para la correcta vision de las tablas
if ($j % 2 eq 0 )
{
print "<tr><td>";
}
}
```



```

else{
print "<td>";
}

# Imprime la informacion del profesor
print "<a href=$ruta?$usuario><img src='.././images/$imagen' height=62 width=62></a>", # Pone la imagen
"</td><td><div align=left>","<a href=$ruta?$usuario>", # Alinea el texto a la izq.
"<font size=2>",br,
# Crea un vinculo a info.pl pasandole como parametro la variable $ip
$nombre,br, # Muestra el nombre
$usuario,br,# Muestra el nombre de usuario
$descripcion, # Muestra la clase a la que asiste y su grupo
"</font></div></td>","</a>";
$j = $j + 1;

}
print "</td></tr></table>";

```

### 3.2 COPIA DE SEGURIDAD DEL FICHERO MENSAJES.TXT

Editaremos un script en shell que genere una copia de seguridad mensual del fichero mensajes.txt utilizado por los profesores.

Este script consta de:

- La variable origen en la que se almacena el fichero a copiar.
- La variable copia que almacena el nombre del fichero origen mas la extension que indica que es una copia de seguridad y la fecha en la que se ha copiado.
- Luego simplemente copia origen en copia.

Pincha aqui para ver el script...

El nombre del fichero resultante tendra este formato:

```
mensajes.txt.bak_07-12-05-18:54
```

Ahora queremos que la copia de seguridad del fichero mensajes se realice una vez al mes. Para ello editaremos el cron para que lo haga una vez al mes a las 23:00.

Ejecutamos en la terminal crontab -e para editar el cron en el insertaremos esta linea:

```
0,23 * * * /admin/departamento/admin/scripts_shell/copia_mensajes.sh
```

contenido del fichero equipos.pl :

```

#!/usr/bin/perl
# Carga el modulo CGI, para poder ejecutar los scripts por pagina güeb
use CGI qw/:standard/;

# Abre un fichero, se le pasa el manejador y la ruta del fichero, se almacena el contenido del fichero
# en el manejador.
open(FILE, '/var/www/perl/red/ordenatas');

```



```

# Pasamos el contenido del fichero a un vector.
@fichero = <FILE>;

# Cerramos el fichero.
close(FILE);

# Escribimos la cabecera de la pagina con la funcion "header" y el titulo y comienzo de la pagina con "start_html"

# HEADER: <!DOCTYPE html
# PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
# "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
# <!--html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US"-->
#
# START_HTML: <!--head-->
# <!--title-->miau<!--/title-->
# <!--meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /-->
# <!--/head-->
# <!--body-->
print header, start_html('miau');

print "<table border=0 cellpadding=0 cellspacing=0 align=center>";
print "<tr><td><img src=../images/logo.jpg width=140 height=140 align=right></td>";
print "<td><em><font size=6>SERUCE <font size=3>SISTEMA DE ESCANEEO DE RED Y USUARIOS PARA
CENTROS EDUCATIVOS<br></em></font>";
print "<font size=3>SERUCE para el I.E.S Leopoldo Queipo, Familia Profesional de Informatica<br></font>";
print "<font size=3>Ciclo Formativo Administracion de sistemas informáticos.<br></font>";
print "<font size=3>Ciclo Formativo Explotación de Sistemas Informaticos</font></td></tr></table>";

print "<table cellspacing=6 border=2 align=center>";
$j = 0;
# El bucle recorre todo el vector, para ir leyendo linea por linea
foreach $linea (@fichero)
{
# Usamos split para separar los campos divididos entre dos puntos
($ip, $nombre, $h, $m, $s) = split(/:/$, $linea);

# Mediante un contador para ir haciendo las columnas de la tabla
if ($j % 3 eq 0 )
{
print "<tr><td>";
}elseif($j % 3 eq 1){
print "<td>";
}else{
print "<td>";
}
$hora = "$h:$m:$s";

if ($ip eq "192.168.0.220"){
print "<a href=servicios.pl?dir=$ip>","<img src='../images/impresora_ico.jpg' height=32 width=42>"," # Pone la
imagen
"</td><td><div align=left>"," # Alinea el texto a la izq.
"<font size=2>"," <a href=servicios.pl?dir=$ip>"," $ip,br,
# Crea un vinculo a servicios.pl pasandole como parametro la variable $ip
" OLIVETTI",br, # Muestra el nombre
"Hora: ", $hora, # Muestra la hora
"</font></div></td>","</a>";
}
elseif ($ip eq "192.168.0.201"){

```



```

print "<a href=servicios.pl?dir=$ip>","<img src='.././images/impresora_ico.jpg' height=32 width=42>"," # Pone la
imagen
"</td><td><div align=left>"," # Alinea el texto a la izq.
"<font size=2>"," <a href=servicios.pl?dir=$ip>"," $ip,br,
# Crea un vinculo a servicios.pl pasandole como parametro la variable $ip
" OKI C5300",br, # Muestra el nombre
"Hora: ", $hora, # Muestra la hora
"</font></div></td>","</a>";
}
else {
# Juntamos la hora, ya que split nos lo ha separado en horas,minutos y segundos.
print "<a href=servicios.pl?dir=$ip>","
"<img src='.././images/pc.jpg' height=52 width=42>"," # Pone la imagen
"</td><td><div align=left>","<a href=servicios.pl?dir=$ip>"," # Alinea el texto a la izq.
"<font size=2>"," $ip,br,
# Crea un vinculo a servicios.pl pasandole como parametro la variable $ip
" ", $nombre,br, # Muestra el nombre
"Hora: ", $hora, # Muestra la hora
"</font></div></td>","</a>";
}
}
$j = $j + 1;
}
print "</table>";

```

contenido de servicios.pl :

```

#!/usr/bin/perl
# Carga el modulo CGI, para poder ejecutar los scripts por pagina güeb
use CGI qw/:standard/;

# Se guarda en la variable lo que se le pasa a la pagina como parametro, se recupera del vector ENV, indicandole
# como indice QUERY_STRING
$parametro = $ENV{QUERY_STRING};

# Separamos el parametro, guardamos en dir la ip, y en nada la palabra dir, la variable se desecha.
($nada,$dir) = split(/=/,$parametro);

# Examina los puertos de la ip y guarda lo que devuelve
$contentido = `nmap $dir`;

# Se guarda en la variable especial el resultado de nmap, lo hacemos para no tener que repetir en
# las condiciones la comparacion con la variable
$_ = $contentido;

# Imprimimos la cabecera y el comienzo de la pagina
print header, start_html;

# Hace la busqueda del protocolo http en el puerto 80
print "<table cellpadding=30 border=2>
<tr><td colspan=5><div align=center><font size=6>SERVICIOS DE LA MAQUINA</div></font></td></tr>
<tr><td>";

print "<table border=3 cellpadding=5> <tr><td><div align=center> HTTP </div></td></tr>
<tr><td><img src='.././images/Firefox-logo.png' height=143 width=143></td></tr>";
if ( /80Vtcp/ )
{

```



```

# Informa que tiene el servicio activo y pone el enlace
print "<tr><td><div align=center><a href=http://$dir> Activo </div> </a></td></tr></table>";
}else{
print "<tr><td><div align=center>No activo</div></td></tr></table>";
}
print "</td><td>";

# Hace la busqueda del protocolo samba
print "<table border=3 cellspacing=5> <tr><td><div align=center> SAMBA </div></td></tr>
<tr><td><img src=../../images/windows_logo.gif height=143 width=143></td></tr>";
if ( /microsoft-ds/ )
{
# Informa que tiene el servicio activo y pone el enlace
print "<tr><td><div align=center><a href=smb://$dir> Activo </div> </a></td></tr></table>";
}else{
print "<tr><td><div align=center>No activo</div></td></tr></table>";
}
print "</td><td>";

# Hace la busqueda del protocolo ssh
print "<table border=3 cellspacing=5> <tr><td><div align=center> SSH </div></td></tr>
<tr><td><img src=../../images/ssh.jpg height=143 width=143></td></tr>";
if ( /ssh/ )
{
# Informa que tiene el servicio activo y pone el enlace
print "<tr><td><div align=center><a href=fish://$dir> Activo </div> </a></td></tr></table>";
}else{
print "<tr><td><div align=center>No activo</div></td></tr></table>";
}
print "</td><td>";

# Hace la busqueda del protocolo ftp
print "<table border=3 cellspacing=5> <tr><td><div align=center> FTP </div></td></tr>
<tr><td><img src=../../images/Ftp.jpg height=143 width=143></td></tr>";
if ( /ftp/ )
{
# Informa que tiene el servicio activo y pone el enlace
print "<tr><td><div align=center><a href=ftp://$dir> Activo </div> </a></td></tr></table>";
}else{
print "<tr><td><div align=center>No activo</div></td></tr></table>";
}
print "</td></tr><tr><td colspan=1>";

# Hace la busqueda del protocolo nfs
print "<table border=3 cellspacing=5> <tr><td><div align=center> NFS </div></td></tr>
<tr><td><img src=../../images/l33t_DEV_nfs_mount.png height=143 width=143></td></tr></table></td><td
colspan=3>";
if ( /nfs/ )
{
# Usamos showmount para ver los recursos compartidos de la maquina
@recursos = `showmount --exports $dir`;
# Lo imprimimos por pantalla
foreach $linea (@recursos)
{
($recurso,$restriccion) = split(/ /,$linea);
#print "$recurso $restriccion",br;
print "$linea",br;
}
}

```



```

}else{
print "No tiene recursos compartidos por nfs";
}
print "</td></tr></table>";

```

### 3.3 INSTALAR Y CONFIGURAR UN SERVIDOR DNS PARA NUESTRA RED DE AREA LOCAL

Lo primero que hemos de hacer es instalar el servicio, sino lo esta ya instalado, para instalarlo abrimos una consola y nos ponemos como root (superusuario).

Una vez hecho esto, tecleamos "urpmi bind" que es el servicio de DNS para Linux, esto nos dara una lista de archivos para descargar, aceptamos y esperamos que se nos instale el paquete.

Una vez instalado el paquete, nos disponemos a configurar el servicio, para ello hemos de ir a el fichero:

```
/etc/named.conf.
```

En este fichero, es donde se definiran las zonas de nuestro servidor DNS, el aspecto de una zona es mas o menos este:

```

Zone "fpi.ies" in {
Type master;
file "fpi.ies"
allow-updates {
127,0,0,1;};
};

```

Una pequeña explicacion de la zona:

Lo primero que nos encontramos es el nombre de la zona.

En nuestro caso sera de tipo primario.

El fichero donde se le dan los nombres a las maquinas.

Las maquinas o IP's que pueden actualizar el servidor DNS.

El formato del fichero donde se definen todas los nombres de las maquinas es, y se guarda en:

```
/var/named/fpi.ies:
```

Una cabecera donde se definen la caducidad y los tiempos de refresco del servidor.

Las siguientes lineas definen el nombre de la maquina con la @ son los nombres que se le daran a la maquina local

A continuacion vendrian todos los equipos de nuestra red.

Y ya tenemos configurado el servidor dns para nuestra red local.

### 3.4 GENERAR UN SCRIPT QUE PONGA EN MARCHA LA INSTALACION DEL SERVIDOR APACHE CON LA CONFIGURACION YA ESTABLECIDA PARA NUESTRO SERVIDOR

Este trozo de la documentacion va a tratar de como instalar el programa entero.

Para instalarlo simplemente tienes que poner en una consola para ejecutar el script ./seruce.sh y el solo te lo descomprime todo en sus respectivas carpetas.

El script tiene tres lineas, la primera linea es la que descomprime todos los archivos en sus carpetas respectivas.

```
tar -xPf seruce.sh
```

La segunda linea es la que copia la configuracion del apache sin confirmacion.



```
cp httpd.conf /var/www/seruce.sh
```

La tercera linea es la que reinicia el servidor apache.

```
service httpd restart
```

Pincha Aqui para Ver el script que pone en marcha el servidor seruce.

Y asi concluye la instalacion y puesta en marcha del servidor.

#### 4. SeCCioN De aLuMNoS

##### 4.1 Explicacion de la seccion

##### 4.2 Script utilizado

##### 4.3 Muestra de ficheros plantilla utilizados

#### 4.1 EXPLICACION DE LA SECCION

El primer script que haremos, sera el que se encarga de coger de un fichero la informacion de las maquinas, como la ip, nombre dns y hora en la que se contacto con ese equipo, le llamaremos carga.pl. El primer paso, sera abrir ese fichero. Una vez abierto, tendremos que ir leyendo linea por linea el contenido, y separando los datos que contiene cada linea. Todo esto, se ira mezclando con etiquetas de html para que la salida salga formateada y bonita. Despues solo habra que ir mostrando los datos con un enlace hacia la pagina con la informacion de los servicios.

El segundo script sera el que da la informacion sobre los servicios de una maquina, el fichero se llamara info.pl, solo tendremos ver que ip ha sido pasada como parametro, y hacer un nmap para buscar los puertos abiertos de esa maquina. Luego recogemos la informacion de los servicios activos y mostramos por pantalla un enlace hacia esos servicios.

Ahora le toca el turno al script alumnos.pl, que se encarga de la lista de los alumnos y profesores, este caso es igual que el primero, se recoge la informacion de un fichero alumno.txt, en el cual se recogerá nombres, apellidos, curso, nombre de usuario y el nombre de la fotografía. Luego solo hay que ir mostrando estos datos.

El siguiente fichero, seria como info.plim pero solo tendra dos hipervinculos hacia la pagina personal y los ficheros personales que el usuario comparte, a este fichero le hemos llamado direc.pl.

Para que quede bonito, hemos añadido unas imagenes, aparte de las fotografias de los usuarios, estas se guardaran en /var/www/images, y cuando hagamos un vinculo a algunos de los datos de esta carpeta, tendra que hacerse ../images, para que funcione correctamente. Y habra que hacer un alias para la carpeta images, pero eso es otra historia.

Por ultimo, tendremos un indice que nos ira llevando a cada una de las paginas anteriores.

#### 4.2 SCRIPT UTILIZADO

Fichero carga.pl en PDF.

Fichero alumnos.pl en PDF.

Fichero info.pl en PDF.

#### 4.3 MUESTRA DE FICHEROS PLANTILLA UTILIZADOS

Los ficheros plantillas que se han utilizado en esta parte del trabajo han sido los de los nombres de los alumnos y los nombres de los profesores.

Alumnos.txt

... y los de los profesores...



## 5. SeCCioN De PRoFeSoReS

5.1. Script para la búsqueda de ficheros en mensajes.txt

5.2. Estructura del fichero mensajes.txt

Como se muestra en el enunciado del trabajo, entre las cosas que tiene que hacer la seccion de alumnos, hay que realizar un script en perl que realice una búsqueda en el fichero mensajes.txt a partir del usuario, fecha o asunto.

El fichero mensajes.txt tiene que tener un formato asi:

fecha:usuario:asunto o asuntos

Ya estan listos los ficheros en el servidor. (Por ejemplo).

[Pincha aqui para ver el script en perl...](#)

Una vez realizado el Script para buscar en el fichero mensajes.txt, hemos realizado otro que genere la lista de todas las cabeceras (como enlaces) que encuentre en el archivo y pinchando en dichas cabeceras, nos mostrara una nueva pagina con la cabecera y el texto correspondiente.

[Pincha aqui para ver el script en perl...](#)

-----  
Para descargar el proyecto seruce version 0.1 ve a la URL siguiente:

<http://fpiqueipo.hopto.org/moodle/mod/resource/view.php?id=20>

(en el archivo comprimido estan todos los scripts)

En sucesivas revisiones se mejorara el articulo, con un tutorial con screenshots de la instalacion y funcionamiento. Se prevee que a finales de marzo estara disponible la version 0.2

---

E-mail del autor: alesfrancis\_ARROBA\_gmail.com

**Podrás encontrar este artículo e información adicional en:** <http://bulma.net/body.phtml?nIdNoticia=2271>