

Granja de servidores

Sistema para consulta de películas

Autores:

- Álvaro Arévalo Toro
- Álvaro Muñoz García
- José Adrián Garrido Puertas
- Pablo Martínez Ruano
- Félix Ramírez García

Índice

Creación de noip	3
Servidor 1 - Álvaro Arévalo Toro	4
Servidor 2 - Pablo Martínez Ruano	5
Balanceador - Álvaro Muñoz García	7
Server SQL - José Adrián Garrido Puertas	10
Aplicación Web - Félix Ramírez García	12

Creación de noip

Para el correcto funcionamiento de nuestro proyecto debemos conectar a la página <https://www.noip.com/> donde nos registraremos, para tener acceso a sus servicios.

En el caso de windows será solo descargar la aplicación para poder ponerlo en funcionamiento siguiendo los pasos que nos indiquen, mientras que al utilizar linux el proceso es un poco más largo pero no por ello más complicado.

Para empezar abrimos el terminal y ejecutamos las siguientes órdenes:

- `cd /usr/local/src/`
- `wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz`
- `tar xf noip-duc-linux.tar.gz`
- `cd noip-2.1.9-1/`
- `make install`

Al utilizar la distribución de ubuntu server, es posible que no tengamos el compilador de c++ instalado, por lo que deberemos poner la siguiente orden:

- `sudo apt-get install gcc`

Seguidamente pedirá tanto el usuario como el password que tengamos para nuestra cuenta.

Para configurar el cliente, ejecutamos como root la siguiente orden

- `/usr/local/bin/noip2 -C` (el -C es para abrir el archivo de configuración)

Ahora ponemos nuestro usuario y la contraseña de No-IP y seleccionamos el subdominio que usaremos. Una vez instalado, para ejecutarlo usamos el siguiente comando.

- `/usr/local/bin/noip2`

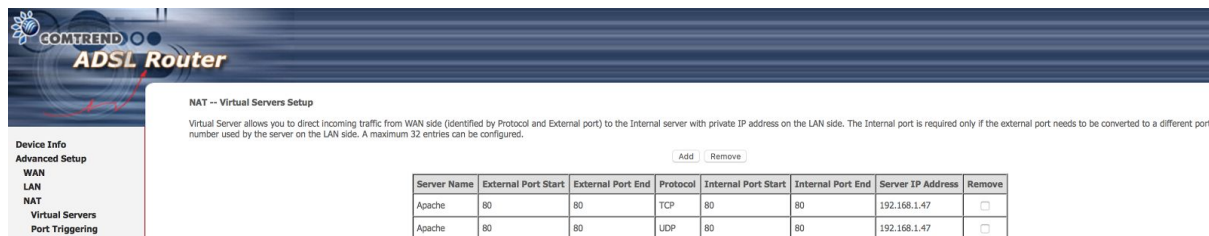
Si quiere ejecutarse automáticamente cuando la máquina es iniciada, entonces deberá poner el siguiente script en su directorio de inicio.

```
#!/bin/sh
#./etc/rc.d/init.d/functions
#descomente/modifique para su killproc
case "$1" in
  start)
    echo "Iniciando noip2..."
    /usr/local/bin/noip2
    ;;
  stop)
    echo -n "Apagando noip2..."
    killproc -TERM /usr/local/bin/noip2
    ;;
  *)
    echo "Uso: $0 {start|stop}"
    exit 1
esac
exit 0
```

Servidor 1 - Álvaro Arévalo Toro

El servidor 1 se ha instalado en un SO **Ubuntu Server 16.04**. La función de este servidor es alojar la aplicación y prestar el servicio. Sin embargo, la conexión a la base de datos se realizará mediante el servidor de base de datos.

En primer lugar, he establecido como IP privada fija la dirección **192.168.1.47** y he abierto el puerto **80** en el router para dicha IP.



Acto seguido he instalado **LAMP** (Apache, MySQL y PHP), junto con el cliente **No-IP** para tener un DNS dinámico que se actualiza cada vez que cambia mi ip pública. Mi dominio de No-IP es **swap001.ddns.net**.

Por último, he puesto en el directorio **/var/www/html** los archivos pertenecientes a la página web. En el archivo **php/consulta.php** introducimos los datos para la conexión con el servidor de la base de datos, que en este caso es **swap007.ddns.net**.

Para realizar un clonado perfecto de todos los archivos del directorio **/var/www/html** del servidor 2 (con dominio **swap002.ddns.net**) usamos el siguiente comando (importante tener abierto el **puerto 22**):

```
rsync -avz --delete -e ssh swap002.ddns.net:/var/www/html /var/www/html
```

Para realizar el acceso por ssh sin contraseña seguimos los pasos realizados en la práctica 2:

```
ssh-keygen -b 4096 -t rsa
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@swap002.ddns.net
```

Para terminar, hacemos `sudo crontab -e` y le añadimos la siguiente línea:

```
0 * * * * rsync -avz --delete -e ssh swap002.ddns.net:/var/www/html /var/www/html
```

Servidor 2 - Pablo Martínez Ruano

El servidor 2 se ha instalado en un SO **Ubuntu Server 16.10**. La función de este servidor es alojar la aplicación y prestar el servicio. Sin embargo, la conexión a la base de datos se realizará mediante el servidor de base de datos.

En primer lugar, debido a la incapacidad de conectar el servidor por cable a internet he tenido que implementar la versión Wi-Fi. Para ello primero descargamos y movemos los drivers de la tarjeta de red vía USB. Una vez realizado instalamos el script con `make all`. Seguidamente para buscar la interfaz de red, ejecutamos el comando `iwconfig`.

```
pablo@Pablo: ~  
pablo@Pablo:~$ iwconfig  
lo        no wireless extensions.  
  
enp3s0    no wireless extensions.  
  
wlp2s0    IEEE 802.11abgn  ESSID:"MOVISTAR_8D9C"  
Mode:Managed  Frequency:2.437 GHz  Access Point: 8C:0C:A3:40:8D:9C  
Bit Rate=58.5 Mb/s   Tx-Power=15 dBm  
Retry short limit:7   RTS thr:off   Fragment thr:off  
Power Management:off  
Link Quality=57/70  Signal level=-53 dBm  
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0  
Tx excessive retries:0 Invalid misc:145 Missed beacon:0  
  
pablo@Pablo:~$
```

En este caso, me sale ESSID porque previamente hice este paso. Este campo me tendría que salir vacío. El último paso consiste en abrir el fichero para editar nuestra red. Este fichero se abre ejecutando `sudo nano /etc/network/interfaces`. Por último lo editamos con los datos de nuestra interfaz de red, SSID y la contraseña.

```
pablo@Pablo: ~  
GNU nano 2.5.3      Archivo: /etc/network/interfaces  
  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto wlp2s0  
iface wlp2s0 inet dhcp  
    wpa-ssid MOVISTAR_8D9C  
    wpa-psk lYzblE8nVF0IJ6N9gYiE  
  
[ 14 líneas leídas ]  
^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar Tex ^J Justificar ^C Posición  
^X Salir      ^R Leer fich.^_ Reemplazar  ^U Pegar txt  ^T Ortografía ^_ Ir a línea
```

También he establecido como IP privada fija la dirección **192.168.1.138** y he abierto el puerto **80** en el router para dicha IP.

Acto seguido he instalado **LAMP** (Apache, MySQL y PHP), junto con el cliente **No-IP** para tener un DNS dinámico que se actualiza cada vez que cambia mi ip pública. Mi dominio de No-IP es **swap002.ddns.net**.

Por último, he puesto en el directorio **/var/www/html** los archivos pertenecientes a la aplicación web. En el archivo **php/consulta.php** introducimos los datos para la conexión con el servidor de la base de datos, que en este caso es **swap007.ddns.net**.

Para realizar un clonado perfecto de todos los archivos del directorio **/var/www/html** del servidor 1 (con dominio **swap001.ddns.net**) usamos el siguiente comando (importante tener abierto el **puerto 22**):

```
rsync -avz --delete -e ssh swap001.ddns.net:/var/www/html /var/www/html
```

Para realizar el acceso por ssh sin contraseña seguimos los pasos realizados en la práctica 2:

```
ssh-keygen -b 4096 -t rsa
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@swap001.ddns.net
```

Para terminar, hacemos `sudo crontab -e` y le añadimos la siguiente línea:

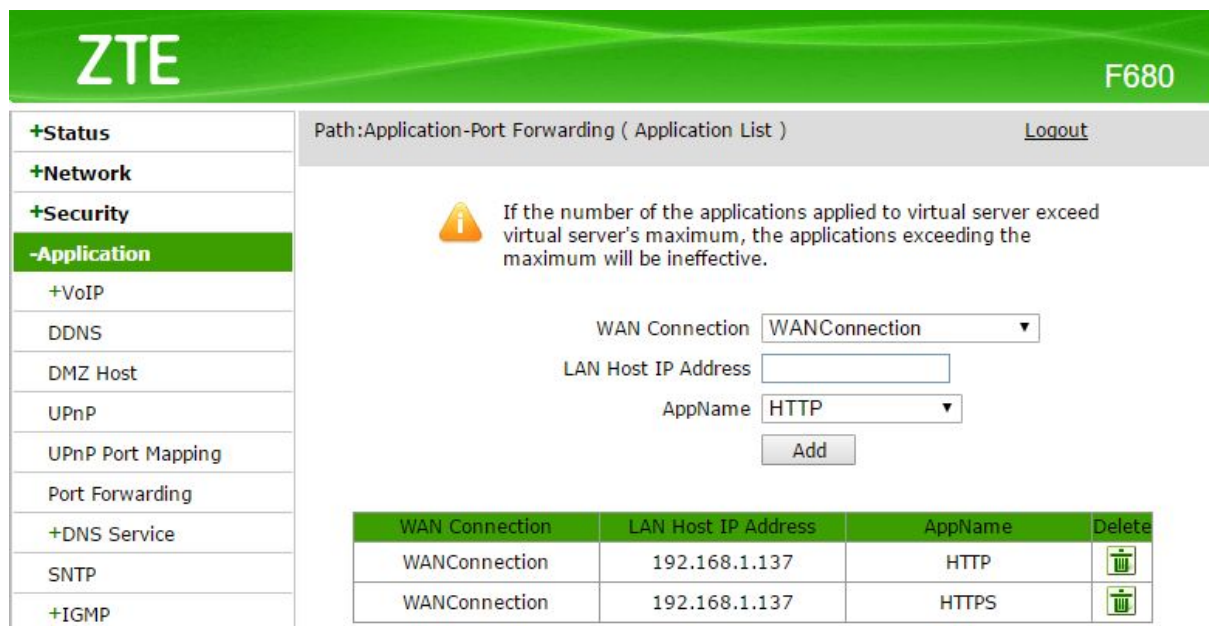
```
0 * * * * rsync -avz --delete -e ssh swap001.ddns.net:/var/www/html /var/www/html
```

Balanceador - Álvaro Muñoz García

En la máquina balanceadora he instalado el SO Ubuntu Server 16.10 utilizado en las prácticas de la asignatura ya que este sistema solo va a ejecutar la aplicación que redirige el tráfico y por tanto no se necesita interfaz gráfica.

Como nuestra granja web está realizada con dominios reales se han tenido que abrir los puertos 80 y 443 en nuestro router para que las máquinas puedan atender las peticiones HTTP y HTTPS. Además se ha instalado el cliente No-IP para que el dominio sea accesible aunque la IP de nuestra máquina cambie.

Accedemos a la configuración del router y para la IP privada de la máquina en cuestión habilitaremos dichos puertos:



The screenshot shows the ZTE F680 router's web interface. The left sidebar contains a menu with options: +Status, +Network, +Security, -Application (highlighted), +VoIP, DDNS, DMZ Host, UPnP, UPnP Port Mapping, Port Forwarding, +DNS Service, SNTP, and +IGMP. The main area is titled 'Path: Application-Port Forwarding (Application List)' and includes a 'Logout' link. A warning icon and text state: 'If the number of the applications applied to virtual server exceed virtual server's maximum, the applications exceeding the maximum will be ineffective.' Below this, there are input fields for 'WAN Connection' (set to 'WANConnection'), 'LAN Host IP Address' (empty), and 'AppName' (set to 'HTTP'). An 'Add' button is present. At the bottom, a table lists existing applications:

WAN Connection	LAN Host IP Address	AppName	Delete
WANConnection	192.168.1.137	HTTP	
WANConnection	192.168.1.137	HTTPS	

En cuanto a la aplicación elegida me he decantado por haproxy ya que como vimos en la práctica 3 de la asignatura ofrecía mejor rendimiento que nginx. Se puede apreciar en esta tabla comparativa en la que se resumen las prestaciones de ambos servicios con los resultados obtenidos en Apache Benchmark:

	nginx	haproxy
peticiones/segundo	3198.56	4777.87
tiempo (s)	312.641	209.298
tasa de transferencia (Kb/s)	840.12	1259.79

El proceso de instalación y configuración es bastante sencillo. Para instalarlo ejecutamos el siguiente comando: `sudo apt-get install haproxy`

Modificamos el archivo de configuración `/etc/haproxy/haproxy.cfg` e incluimos las directivas indicadas en la práctica 3 incluyendo los dominios a los que se va a redirigir el tráfico:

```
global
    daemon
    maxconn 256

defaults
    mode http
    timeout 4000
    clitimeout 42000
    srvtimeout 43000

frontend http-in
    bind *:80
    default_backend servers

backend servers
    server m1 swap001.ddns.net:80 maxconn 32
    server m2 swap002.ddns.net:80 maxconn 32
```

Una vez que hayamos guardado los cambios reiniciamos el servicio con: `sudo service haproxy restart`

Como queremos que las conexiones que se hagan al balanceador sean seguras (por HTTPS) he instalado el certificado Let's Encrypt como relato a continuación.

Instalamos la herramienta git y clonamos el repositorio de letsencrypt en /opt:

```
sudo apt-get install git bc

sudo git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt
```

Para obtener el certificado debemos asegurarnos de que el puerto 80 no está en uso, para ello podemos ejecutar la siguiente orden y si no hay ningún tipo de salida es que dicho puerto está libre: `netstat -tulpn | grep :80`

Una vez hecha esta comprobación, nos situamos en el directorio `/opt/letsencrypt/` y ejecutamos el plugin que genera nuestro certificado:

```
cd /opt/letsencrypt

./letsencrypt-auto certonly --standalone
```

Se nos pedirá nuestro email, aceptar las condiciones y el nombre de nuestro dominio y una vez que hayamos facilitado estos datos nos aparecerá un mensaje que nos proporciona información como cuando expira nuestro certificado o el directorio en el que se ha generado. Comprobamos que en dicha carpeta tenemos

los ficheros cert.pem, chain.pem, fullchain.pem y privkey.pem con el siguiente comando: `sudo ls /etc/letsencrypt/live/swap003.ddns.net`

Para que haproxy utilice este certificado SSL deberemos incluirlo en su fichero de configuración. Lo primero que debemos hacer es combinar los archivos fullchain.pem and privkey.pem en uno solo. En este caso creamos el directorio /etc/haproxy/certs para almacenar el archivo resultante y le damos permisos:

```
sudo mkdir -p /etc/haproxy/certs
```

```
DOMAIN='swap003.ddns.net' sudo -E bash -c 'cat /etc/letsencrypt/live/$DOMAIN/fullchain.pem /etc/letsencrypt/live/$DOMAIN/privkey.pem > /etc/haproxy/certs/$DOMAIN.pem'
```

```
sudo chmod -R go-rwx /etc/haproxy/certs
```

Por último, editamos nuevamente el archivo /etc/haproxy/haproxy.cfg para incluir la configuración de nuestro certificado y de tráfico HTTPS. Quedaría de esta forma:

```
global
    daemon
    maxconn 256
    tune.ssl.default-dh-param 2048

defaults
    mode http
    timeout 4000
    clitimeout 42000
    srvtimeout 43000
    option forwardfor
    option http-server-close

frontend http-in
    bind *:80
    reqadd X-Forwarded-Proto:\ http
    default_backend servers

frontend https-in
    bind *:443 ssl crt /etc/haproxy/certs/swap003.ddns.net.pem
    reqadd X-Forwarded-Proto:\ https
    acl letsencrypt-acl path_beg /.well-known/acme-challenge/
    use_backend letsencrypt-backend if letsencrypt-acl
    default_backend servers

backend servers
    redirect scheme https if !{ ssl_fc }
    server m1 swap001.ddns.net:80 check
    server m2 swap002.ddns.net:80 check

backend letsencrypt-backend
    server letsencrypt 127.0.0.1:54321
```

Una vez hayamos guardado los cambios y reiniciado el servicio tendremos el balanceador configurado de forma segura.

Server SQL - José Adrián Garrido Puertas

En mi caso, he tenido que abrir los puertos **80**, **443** y **3306**, para lo cual he tenido que conectar con el portal alejandra al tener la conexión con movistar. A continuación dejo una foto de como se hace la configuración:

ABRIR PUERTOS

Para que ciertas aplicaciones funcionen, necesitas abrir puertos. Un caso típico es el de los juegos online: P2P, Xbox, PS3... A continuación, te facilitamos un listado de puertos predeterminados, de uso frecuente, para que no tengas que añadirlos de forma manual.

Elige un formato

¿Qué regla de puertos vas a abrir?

☐ Predeterminado ☒ Manual

☒ Puerto único ☐ Rango de puertos

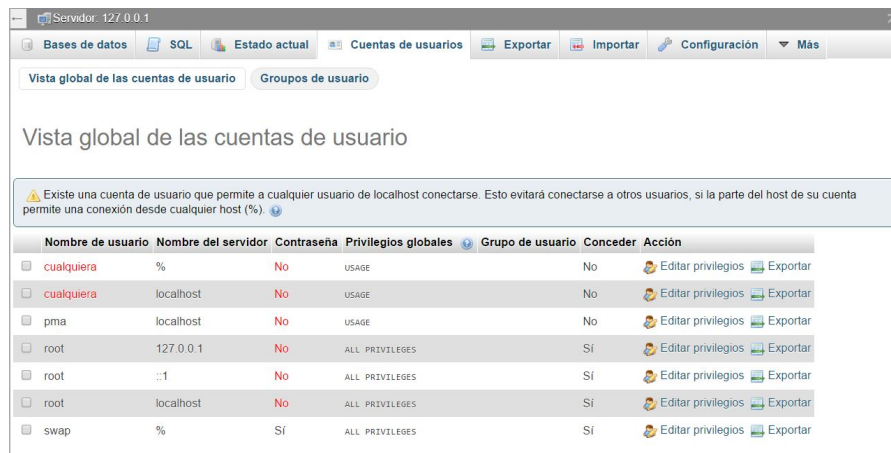
Dirección IP *	Puertos *	Protocolo	
<input type="text" value="192.168.1.139"/>	<input type="text" value="3306"/>	<input type="text" value="TCP+UDP"/>	<input type="button" value="Añadir"/>
Dirección IP			

Lo primero que haremos será la creación de la bd cuando ya tengamos los puertos abiertos. En mi caso, se ha realizado todo mediante **XAMPP** en windows 10, por lo abriremos XAMPP e iniciaremos MySQL y apache, y hecho esto entraremos desde el navegador en localhost/phpmyadmin.

Una vez dentro, solo seguiremos los pasos para la creación de la bd, en nuestro caso, una bd llamada "*peliculas*" con una tabla "*pelicula*", en la que se almacenan los datos referentes a:

- ID de la película
- Género
- Edad recomendada
- Director
- Sinopsis
- Nombre de la película

Cuando tenemos la clave, para facilitar la conexión externa crearemos un usuario al que llamaremos swap, con los permisos para acceder a la tabla.



Hecho esto, conectaremos la aplicación de no-ip para activar el dominio **swap007.ddns.net**, que es el que cree para poder conectarme.

Si toda la configuración es correcta, no habrá problema a la hora de conectar con el servidor de base de datos.

A continuación añado una foto con los datos insertados en la base de datos.

		id_Pelicula	genero	anio	edad_Recomendada	director	sinopsis	nombre
<input type="checkbox"/>	Editar Copiar Borrar	1	Terror	2000	13	Patrick Lussier	Cuando, tras cien años de confinamiento, Dracula e...	Dracula 2000
<input type="checkbox"/>	Editar Copiar Borrar	2	Animacion	2000	1	Kunihiko Yuyama, Michael Haigney	Segunda entrega del famoso videojuego japonés. Ash...	Pokémon 2: El poder de uno
<input type="checkbox"/>	Editar Copiar Borrar	3	Ciencia Ficción	2000	18	Roger Spottiswoode	Cuando el piloto de helicóptero Adam Gibson, despu...	El 6º día
<input type="checkbox"/>	Editar Copiar Borrar	4	Comedia	2000	13	Jay Roach	Greg Focker (Ben Stiller) es un enfermero algo pat...	Los padres de ella
<input type="checkbox"/>	Editar Copiar Borrar	5	Fantasia	2001	1	Peter Jackson	En la Tierra Media, el Señor Oscuro Saurón ordenó ...	El señor de los anillos: La comunidad del anillo
<input type="checkbox"/>	Editar Copiar Borrar	6	Fantasia	2002	1	Peter Jackson	Tras la disolución de la Compañía del Anillo, Frodo...	El señor de los anillos: Las dos torres
<input type="checkbox"/>	Editar Copiar Borrar	7	Fantasia	2003	1	Peter Jackson	Las fuerzas de Saruman han sido destruidas, y su f...	El señor de los anillos: El retorno del rey

Aplicación Web - Félix Ramírez García

La interfaz de la aplicación es la siguiente:

ELIGE TU PELICULA

Genero:

Drama

Año estreno:

Edad recomendada:

Buscar

Ha seleccionado películas del genero Accion , del año 2014 , y para una edad de 13 años o mas

ID	GENERO	AÑO	EDAD	DIRECTOR	SINOPSIS	NOMBRE
1	Accion	2014	13	steven spielberg	Ocho personas metidas 3 horas en una habitacion.	Los odiosos ocho
2	Accion	2014	13	Christopher Nolan	Otras tres horas de pelicula	Interstellar

En ella se encuentra un formulario con 3 campos que se envían con el método POST a esta misma página y un botón de búsqueda.

En la parte inferior se encuentra la información cargada de la base de datos, mostrándose únicamente en el caso de que se haya enviado el formulario y se haya encontrado alguna película con tales características. Toda esta consulta se realiza en el archivo **php/consulta.php**, como muestra el siguiente código:

```
<?php
    if(isset($_POST['submit']))
    {
        include_once("php/consulta.php");
    }
?>
```

El archivo **php/consulta.php** contiene el código necesario para la conexión:

```
<?php
// Conectando, seleccionando la base de datos
$link = mysql_connect('swap007.ddns.net:3306', 'swap', '1234')
    or die('No se pudo conectar: ' . mysql_error());
echo 'Conectado correctamente a la base de datos';
mysql_select_db('peliculas') or die('No se pudo seleccionar la base de datos');
```

Para la consulta a la base de datos e imprimir las filas de la base de datos, hacemos lo siguiente:

```
$query = "SELECT * FROM pelicula WHERE genero LIKE '%{$genero}%' and anio LIKE '%{$anio}%'  
and edad_Recomendada LIKE '%{$edad}%';"  
$result = mysql_query($query) or die('Consulta fallida: ' . mysql_error());
```

La aplicación se compone de una estructura de archivos simple, en el nodo raíz se encuentran, además de la pagina principal **index.php**, las carpetas **css**, **js**, y **php**.

Estas carpetas, como su nombre indica, contienen el código css, php, etc... para la funcionalidad de la aplicación.