

Tarea para DAW01 de José Luis Comesaña Cabeza.

La arquitectura Web es un modelo compuesto de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?

- 1. Capa de presentación** es la encargada de la navegabilidad, validación de los datos de entrada, formateo de los datos de salida, presentación de la web, etc.; se trata de la capa que se presenta al usuario.
Comprende las responsabilidades de:
 - Navegabilidad del sistema
 - Validación de datos de entrada
 - Formateo de los datos de salida
 - Internacionalización
 - Renderizado de presentación
 - etc
- 2. Capa de negocio** es la que recibe las peticiones del usuario y desde donde se le envían las respuestas; en esta capa se verifican que las reglas establecidas se cumplen.
Comprende las responsabilidades de:
 - Lógica de negocio (o dominio) del sistema.
 - Resultado del análisis funcional (*conjunto de reglas de negocio que abstraen el mundo real*)
 - Capa de presentación y capa de negocio han de ser independientes entre sí en la medida de lo posible
- 3. Capa de acceso a datos** o capa de persistencia, es la formada por determinados gestores de datos que se encargan de la lógica de persistencia de las entidades que maneja el sistema en desarrollo:
 - Inserción
 - Eliminación
 - Actualizaciones
 - Búsquedas
 - etcNo tiene por qué tratarse necesariamente de una base de datos relacional

Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web; destacan dos plataformas web, LAMP y WISA. Explica en qué consiste cada una de ellas.

La plataforma LAMP consiste en un conjunto de herramientas de software libre que al trabajar juntas nos permite disponer de un servidor web. Las siglas de este tipo de plataforma provienen de los términos:

- ✓ **L**inux: Sistema operativo.
- ✓ **A**pache: Servidor web.
- ✓ **M**ySQL: Gestor de bases de datos.
- ✓ **P**HP: Lenguaje interpretado PHP, aunque a veces se sustituye por Perl o Python.

Son de las más apreciadas por los proveedores de servicios web, ya que tienen como ventajas:

- ➔ Compuesta por software libre
- ➔ El código de todos sus componentes está disponible, con lo que podemos realizar las modificaciones que creamos oportunas.

Por su parte, la plataforma WISA está basada en tecnologías desarrolladas por la compañía Microsoft; se trata, por lo tanto, de software propietario. La componen los siguientes elementos:

- ✓ **W**indows: Sistema operativo.
- ✓ **I**nternet Information Services: servidor web.
- ✓ **S**QL Server: gestor de bases de datos.
- ✓ **A**SP o ASP.NET: como lenguaje para scripting del lado del servidor.

A parte del consabido coste económico que requiere la utilización de este tipo de plataforma, tiene como puntos fuertes:

- ➔ Más robusta si se usa para aplicaciones web que funcionen en una intranet o que posean un gran nivel de actividad.
- ➔ Posee multitud de aplicaciones desarrolladas por terceros (*aunque la mayoría son de pago*)

Dispones de una máquina que cuenta con el sistema operativo Ubuntu 12.04 LTS recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:

Instalar el servidor web Apache desde terminal.

Accedemos a una pantalla de terminal pulsando `Ctrl + Alt + T` y nos logamos como usuario root tecleando `sudo su` y poniendo la contraseña de root que tengamos.

Ahora tecleamos la orden:

```
apt-get install apache2
```

Nos avisa del número de paquetes que se instalarán y el espacio necesario para ello.

Si pulsamos “s” a la pregunta de ¿Desea continuar (s/n)? se comenzará la instalación de forma automática hasta finalizar teniendo instalado el servidos Apache.

```
root@Jflicmini:/home/joseluis# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1
 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
 apache2-doc apache2-suexec apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
 apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 9 se instalarán, 0 para eliminar y 1 no actualizados.
Necesito descargar 1.817 kB de archivos.
Se utilizarán 5.220 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? [s]
```

Comprobar que está funcionando el servidor Apache desde terminal.

Para comprobar el funcionamiento de apache desde terminal, estando como root teclearemos:

```
/etc/init.d/apache2 status
```

tras lo que nos informará que se está ejecutando y el número de proceso que tiene.

```
root@Jflicmini:/home/joseluis# /etc/init.d/apache2 status
Apache2 is running (pid 1258).
root@Jflicmini:/home/joseluis#
```

Comprobar que está funcionando el servidor Apache desde navegador.

Para comprobar el funcionamiento del servidor desde el navegador, abrimos éste, por ejemplo Mozilla Firefox y en su barra de direcciones tecleamos `localhost` o `127.0.0.1` tras lo cual nos aparecerá la pantalla por defecto del servidor “**It works!**”.

It works!
This is the default web page for this server.
The web server software is running but no content has been added, yet.

Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82.

Para cambiar el puerto de escucha de Apache hemos de editar el fichero `ports.conf` que se encuentra en `etc/apache2`, por lo que en un terminal como root tecleamos:

```
gedit /etc/apache2/ports.conf
```

Y modificamos las líneas que hacen referencia al puerto **80** que es el que deseamos cambiar, es decir, cambiaremos:

```
NameVirtualHost *:80
Listen 80
```

por estas otras:

```
NameVirtualHost *:82
Listen 82
```

Y como nuestro servidor va a ofrecer acceso a páginas web, tendremos que

```
root@Jflicmini:/home/joseluis# gedit /etc/apache2/ports.conf
root@Jflicmini:/home/joseluis# gedit /etc/apache2/sites-enabled/000-de
fault
root@Jflicmini:/home/joseluis# /etc/init.d/apache2 restart
* Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified do
main name, using 127.0.1.1 for ServerName
... waiting apache2: Could not reliably determine the server's fully
qualified domain name, using 127.0.1.1 for ServerName
[ OK ]
root@Jflicmini:/home/joseluis#
```

modificar también el puerto de las páginas que sirvamos, por lo que desde el terminal editamos el fichero:

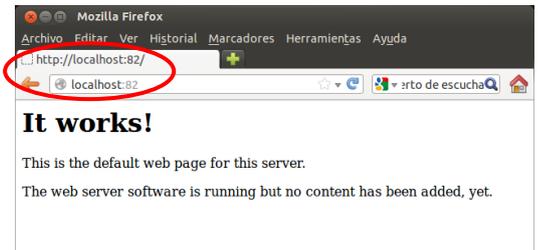
```
gedit /etc/apache2/sites-enabled/000-default
```

Y modificamos `VirtualHost *:80` por `VirtualHost *:82`

Tras ello hemos de reiniciar el servidor tecleando:

```
/etc/init.d/apache2 restart
```

A partir de este momento, si en un navegador tecleamos `localhost` nos saldrá una página de error, pero si ponemos `localhost:82` nos aparecerá la consabida página de **"It Works!"**

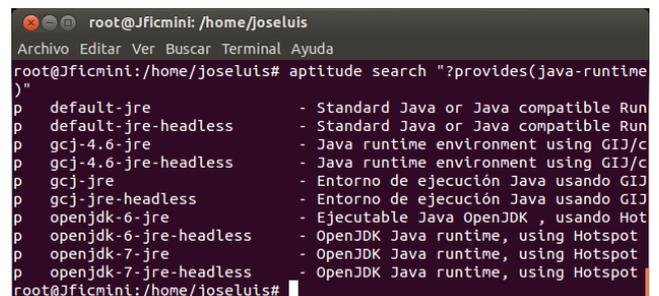


Instalar el servidor de aplicaciones Tomcat.

Lo primero que hemos de hacer es instalar el JDK de Java, ya que se usará un conector proporcionado por Java para redirigir las peticiones que se hagan a Apache, a Tomcat.

Para realizar esto buscaremos el paquete Java que más nos interese e instalamos para ello `aptitude` como buscador de versiones tecleando `apt-get install aptitude` y cuando lo tengamos instalado tecleamos:

```
aptitude search "?provides(java-runtime)"
```



Decidimos instalar la primera versión que aparece, por lo que tecleamos:

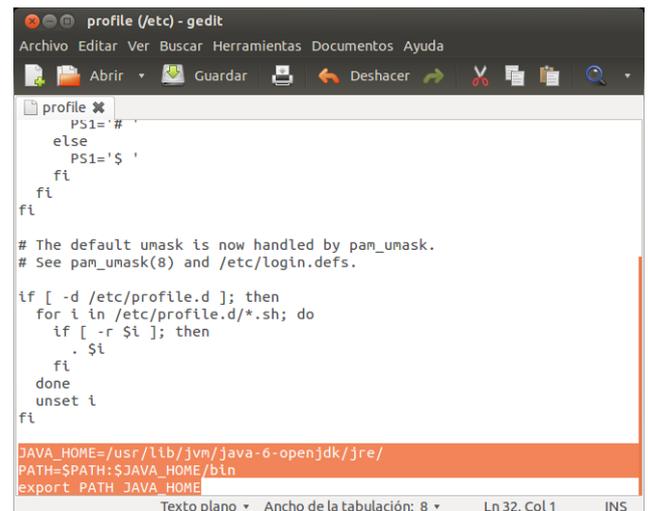
```
apt-get install default-jre
```

Ahora, y para que esté disponible desde cualquier parte del sistema, tenemos que añadir la dirección donde se encuentra instalado, a la variable de entorno PATH, por lo que editamos como root el fichero `profile` tecleando:

```
gedit /etc/profile
```

y añadimos al final del archivo las líneas:

```
JAVA_HOME=/usr/lib/jvm/java-6-openjdk/jre/
PATH=$PATH:$JAVA_HOME/bin
export PATH JAVA_HOME
```



Lo grabamos y actualizamos las variables de entorno tecleando en el terminal:

```
source /etc/profile
```

Ya tenemos Java preparado y es hora de descargar la última versión estable de Tomcat, para lo cual vamos a la dirección <http://apache.rediris.es/tomcat/tomcat-6/> y vemos que la última es la 6.0.36. Vamos a su carpeta bin y nos fijamos en el nombre del fichero comprimido que tendremos que descargar, para volver al terminal de root y proceder a dicha descarga tecleando:

```
wget http://apache.rediris.es/tomcat/tomcat-6/v6.0.36/bin/apache-tomcat-6.0.36.tar.gz
```

Cuando finalizamos su descarga lo descomprimos tecleando:

```
tar xvzf apache-tomcat-6.0.36.tar.gz
```

y ahora lo movemos a la carpeta donde procederemos a su instalación:

```
mv -v apache-tomcat-6.0.36 /usr/local/
```

Hacemos un enlace para realizar las actualizaciones de forma más cómoda:

```
ln -s /usr/local/apache-tomcat-6.0.36/ /usr/local/tomcat
```

Y para finalizar arrancamos el servidor Tomcat tecleando:

```
sh /usr/local/apache-tomcat-6.0.36/bin/catalina.sh start
```

Como todo ha ido bien tecleando en el navegador `localhost:8080` accederemos a la página inicial de Tomcat.

