

ACTIVIDAD 2. CONFIGURACIÓN DE UN SERVIDOR.

DESPLIEGUE DE APLICACIONES WEB

JAIME JUÁREZ FERNÁNDEZ

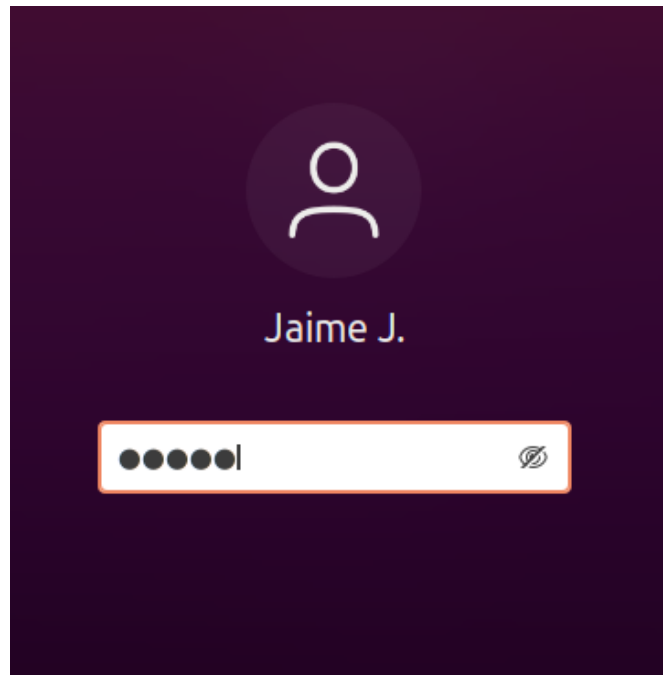
Índice

INSTALACIÓN JAVA	2
INSTALACIÓN APACHE	3
INSTALACIÓN TOMCAT	5
INSTALACIÓN OPENSSE	7
INSTALACIÓN MARIADB	8

En la siguiente guía se a desarrollar el proceso de creación, instalación y configuración de un servidor en la máquina virtual con ubuntu.

Se hará una explicación de cada comando utilizado y una captura para ir viendo las salidas que da la terminal con cada comando.

Para comenzar se arrancará la máquina virtual ya instalada previamente con el usuario que se creó en la instalación.



1.- Verificación e instalación de java

Se comprueba si está instalado java para ello se utiliza el comando **java - versión**

```
jaine@ubuntu:~$ java --version
Command 'java' not found, but can be installed with:

sudo apt install openjdk-11-jre-headless # version 11.0.10+9-0ubuntu1~20.04, or
sudo apt install default-jre             # version 2:1.11-72
sudo apt install openjdk-13-jre-headless # version 13.0.4+8-1~20.04
sudo apt install openjdk-14-jre-headless # version 14.0.2+12-1~20.04
sudo apt install openjdk-8-jre-headless  # version 8u282-b08-0ubuntu1~20.04

jaine@ubuntu:~$
```

Como se puede apreciar no está instalado java. Para ello se utilizarán los comandos proporcionados por la propia terminal de Ubuntu.

Con el comando `sudo apt install default-jre` se instala el jre por defecto.

```
jaime@ubuntu:~$ sudo apt install default-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre
  openjdk-11-jre-headless
Suggested packages:
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
```

Con el comando `which java` verificamos donde se ha instalado

```
jaime@ubuntu:~$ which java
/usr/bin/java
jaime@ubuntu:~$
```

Verificamos la carpeta donde se ha instalado java

2.- Verificación e instalación de Apache.

Para iniciar la instalación del servidor apache, primero actualizo todos los paquetes y luego instalo con `sudo apt install apache2`.

```
jaime@ubuntu:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 1.819 kB of archives.
```

Una vez instalado vemos la lista de los perfiles disponibles con el comando `sudo ufw app list`, Solo habilitaremos el Apache con el puerto 80, ya que si no se van a hacer conexiones seguras no es necesario activar Apache full ya que te abre el puerto 80 y el 443 que es por dónde se accede de forma segura.

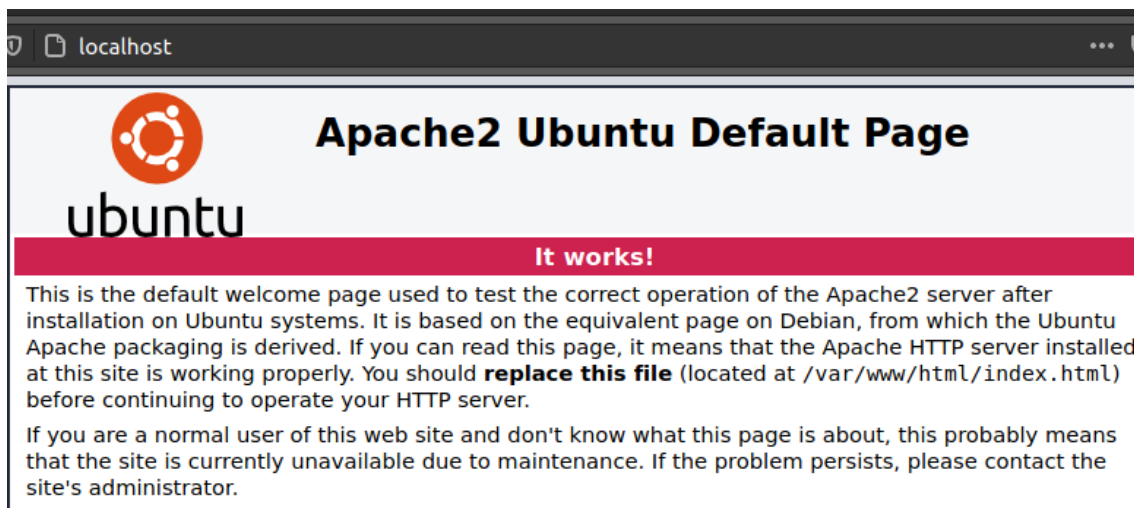
```
jaime@ubuntu:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
jaime@ubuntu:~$
```

Habilitamos esta opción para poder entrar

Para habilitar esa configuración se hace con el siguiente comando `sudo ufw allow 'Apache'`, en la siguiente imagen veremos cómo se realiza.

```
jaime@ubuntu:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
jaime@ubuntu:~$
```

Como podemos apreciar en la siguiente imagen metiendo en la barra de dirección la url de la consola local nos deja entrar en apache y poder trabajar con él.



Con el comando `netstat -tlnp` vemos la lista de puertos que tenemos a la escucha y muestra como el puerto 80 está a la escucha.

```
jaime@ubuntu:~$ netstat -tlnp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN
-
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN
-
tcp6       0      0 0 :::80                 :::*                    LISTEN
-
tcp6       0      0 0 :::1:631              :::*                    LISTEN
-
jaime@ubuntu:~$
```

Si se quiere comprobar el estado del apache, con el comando `systemctl status apache2` lo podemos verificar

```
jaime@ubuntu:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enable
   Active: active (running) since Mon 2021-02-22 04:51:47 PST; 36min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 9704 (apache2)
      Tasks: 55 (limit: 2281)
     Memory: 6.1M
    CGroup: /system.slice/apache2.service
            └─9704 /usr/sbin/apache2 -k start
              └─9705 /usr/sbin/apache2 -k start
                └─9706 /usr/sbin/apache2 -k start

feb 22 04:51:46 ubuntu systemd[1]: Starting The Apache HTTP Server...
feb 22 04:51:47 ubuntu apachectl[9703]: AH00558: apache2: Could not reliably determine
feb 22 04:51:47 ubuntu systemd[1]: Started The Apache HTTP Server.
```

3.- Verificación e instalación de Tomcat.

Para comenzar con la instalación del servidor de aplicaciones web Tomcat lo haremos con el siguiente comando `sudo apt install -y tomcat9 tomcat9-admin` esto nos instalará los paquetes principales al núcleo de tomcat y sus aplicaciones administrativas.

```
jaime@ubuntu:~$ sudo apt install -y tomcat9 tomcat9-admin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libecj-jdt-core-java libtcnative-1 libtomcat9-java tomcat9-common
Suggested packages:
  tomcat9-docs tomcat9-examples tomcat9-user
The following NEW packages will be installed:
  libecj-jdt-core-java libtcnative-1 libtomcat9-java tomcat9 tomcat9-admin
  tomcat9-common
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 12,3 MB of archives.
After this operation, 14,8 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libecj-jdt-core-  
libtcnative-1 libtomcat9-java tomcat9 tomcat9-admin tomcat9-common
```

Posteriormente si se quiere también se pueden instalar la documentación y algunos ejemplos de este servidor con los comandos `sudo apt install tomcat9-doc` y `sudo apt install tomcat9-examples`.

Para verificar el estado del tomcat a través del comando `systemctl status tomcat9` o `systemctl status tomcat9.service` podemos observar que se ha iniciado de forma correcta y está corriendo

```
jaine@ubuntu:~$ systemctl status tomcat9
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enable
   Active: active (running) since Mon 2021-02-22 05:17:40 PST; 6min ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Main PID: 12459 (java)
    Tasks: 35 (limit: 2281)
   Memory: 151.4M
    CGroup: /system.slice/tomcat9.service
           └─12459 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=

feb 22 05:17:50 ubuntu tomcat9[12459]: Deployment of deployment descriptor [/etc/tomcat
feb 22 05:17:50 ubuntu tomcat9[12459]: Deploying deployment descriptor [/etc/tomcat9/Ca
feb 22 05:17:50 ubuntu tomcat9[12459]: The path attribute with value [/manager] in depl
feb 22 05:17:51 ubuntu tomcat9[12459]: At least one JAR was scanned for TLDs yet contain
feb 22 05:17:51 ubuntu tomcat9[12459]: Deployment of deployment descriptor [/etc/tomcat
feb 22 05:17:51 ubuntu tomcat9[12459]: Deploying web application directory [/var/lib/to
feb 22 05:17:53 ubuntu tomcat9[12459]: At least one JAR was scanned for TLDs yet contain
feb 22 05:17:53 ubuntu tomcat9[12459]: Deployment of web application directory [/var/li
```

Volvemos a comprobar los puertos abiertos para saber si está a la escucha el puerto 8080 que es por el que tenemos que entrar en el tomcat. Con el comando `sudo netstat -plnt` se puede ver que está a la escucha y que el programa que lo usa es java.

```
jaine@ubuntu:~$ sudo netstat -plnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      634/systemd-resolve
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      722/cupsd
tcp6       0      0 :::8080                 :::*                   LISTEN      12459/java
tcp6       0      0 :::80                   :::*                   LISTEN      9704/apache2
tcp6       0      0 :::1:631                :::*                   LISTEN      722/cupsd
udp        0      0 0.0.0.0:5353            0.0.0.0:*               *          719/avahi-daemon: r
udp        0      0 0.0.0.0:47643           0.0.0.0:*               *          719/avahi-daemon: r
udp        0      0 0.0.0.0:631             0.0.0.0:*               *          799/cups-browsed
udp        0      0 127.0.0.1:53            0.0.0.0:*               *          634/systemd-resolve
udp6       0      0 :::5353                 :::*                   *          719/avahi-daemon: r
udp6       0      0 :::41683                :::*                   *          719/avahi-daemon: r
jaine@ubuntu:~$
```

Se abre el navegador de internet y para entrar al tomcat se puede verificar tecleando localhost.8080, si se abre el tomcat es que podemos empezar a trabajar con el.



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally by clicking [here](#).

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP by clicking [here](#).

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. One is [webapp](#) and the [host-manager webapp](#).

4.- Instalación de Open SSH

El openssh sirve para realizar conexiones de forma remota y otras comunicaciones especiales de red de forma más segura.

Para instalar ssh desde los repositorios del sistema, primero es aconsejable actualizar todos los paquetes y una vez actualizados comenzar la instalación.

A través de este comando `sudo apt install -y ssh` instalamos ssh en nuestra máquina virtual.

```
jai@me@ubuntu:~$ sudo apt install -y ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 693 kB of archives.
After this operation, 6.130 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-sftp-server amd64 1:8.2p1-4ubuntu0.1 [51,5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-server amd64 1:8.2p1-4ubuntu0.1 [377 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 ssh all 1:8.2p1-4ubuntu0.1 [5.080 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal/main amd64 ncurses-term all 6.2-0ubuntu2 [249 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal/main amd64 ssh-import-id all 5.10-0ubuntu1 [10,0 kB]
```

Una vez instalado se puede comprobar el estado del servicio para corroborar su correcta instalación esta parte se haría a través de este comando `systemctl status ssh`

```
jai@me@ubuntu:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-02-22 05:49:29 PST; 1min 41s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 13829 (sshd)
      Tasks: 1 (limit: 2281)
     Memory: 1.4M
    CGroup: /system.slice/ssh.service
            └─13829 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

feb 22 05:49:29 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
feb 22 05:49:29 ubuntu sshd[13829]: Server listening on 0.0.0.0 port 22.
feb 22 05:49:29 ubuntu sshd[13829]: Server listening on :: port 22.
feb 22 05:49:29 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
```

Podemos ver que se ha habilitado el puerto 22 que es por el cual se entra con seguridad al servidor web. Si comprobamos de nuevo los puertos a la escucha veremos habilitado el puerto 22.


```

jaime@ubuntu:~$ sudo netstat -plnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      634/systemd-resolve
tcp        0      0 0.0.0.0:22            0.0.0.0:*               LISTEN      13829/sshd: /usr/sb
tcp        0      0 127.0.0.1:631         0.0.0.0:*               LISTEN      722/cupsd
tcp6       0      0 :::8080               :::*                    LISTEN      12459/java
tcp6       0      0 :::80                 :::*                    LISTEN      9704/apache2
tcp6       0      0 :::22                 :::*                    LISTEN      13829/sshd: /usr/sb
tcp6       0      0 :::631                :::*                    LISTEN      722/cupsd
udp        0      0 0.0.0.0:5353          0.0.0.0:*               *          719/avahi-daemon: r
udp        0      0 0.0.0.0:47643         0.0.0.0:*               *          719/avahi-daemon: r
udp        0      0 0.0.0.0:631          0.0.0.0:*               *          799/cups-browsed
udp        0      0 127.0.0.53:53         0.0.0.0:*               *          634/systemd-resolve
udp6       0      0 :::5353               :::*                    *          719/avahi-daemon: r
udp6       0      0 :::41683              :::*                    *          719/avahi-daemon: r
jaime@ubuntu:~$

```

5.- Instalación de MariaDB

Para la instalación de MariaDB no se instalará ningún entorno de desarrollo como workbench o xampp, se hará todo por consola y mostraremos que ponemos meter alguna query a través de comandos en el terminal.

Para comenzar la instalación actualizamos todos los paquetes de Ubuntu con el comando `sudo apt update` y luego con el siguiente comando `sudo apt install mariadb-server` instalamos el server de la BBDD.

```

jaime@ubuntu:~$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
  libdbd-mysql-perl libdbi-perl libfcgi-perl libhtml-template-perl libreadline5 libsigsegv2
  libsnappy1v5 libterm-readkey-perl mariadb-client-10.3 mariadb-client-core-10.3
  mariadb-common mariadb-server-10.3 mariadb-server-core-10.3 socat
Suggested packages:
  gawk-doc libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx mariadb-test tinyca
The following NEW packages will be installed:
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
  libdbd-mysql-perl libdbi-perl libfcgi-perl libhtml-template-perl libreadline5 libsigsegv2
  libsnappy1v5 libterm-readkey-perl mariadb-client-10.3 mariadb-client-core-10.3
  mariadb-common mariadb-server mariadb-server-10.3 mariadb-server-core-10.3 socat
0 upgraded, 21 newly installed, 0 to remove and 6 not upgraded.
Need to get 20,1 MB of archives.
After this operation, 167 MB of additional disk space will be used.

```

Con siguiente comando `sudo mysql_secure_installation` haremos que MariaDB sea más segura y nos pida una contraseña para poder realizar el inicio de sesión.

```

jaime@ubuntu:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):

```

De momento no le metemos ninguna contraseña para poder acceder como root sin contraseña y también dejamos la base de datos de prueba para poder mostrar que se accede y está disponible desde el exterior.

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] n
... skipping.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
```

Abrimos mariadb para configurar un usuario y una contraseña. Abrimos MariaDB con el siguiente comando `sudo mariadb` y nos mete dentro del servidor.

```
Thanks for using MariaDB!
jaime@ubuntu:~$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 54
Server version: 10.3.25-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Se procede a crear un usuario con privilegios de root y una contraseña

`GRANT ALL ON *.* TO 'jaime'@'localhost' IDENTIFIED BY 'jaime' WITH GRANT OPTION`

```
MariaDB [(none)]> GRANT ALL ON *.* TO 'jaime'@'localhost' IDENTIFIED BY 'jaime' WITH GRANT OP
TION;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Vaciamos los privilegios para que garantizar que se guarden y estén disponibles en la sesión actual con `FLUSH PRIVILEGES`

Cerramos Mariadb con `exit` y probamos la conexión con el usuario y la password configurados anteriormente. Finalmente comprobamos el estado de MariaDB con el siguiente comando `systemctl status mariadb` y podemos comprobar que esta activo y funciona correctamente.

```
jaime@ubuntu:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.3.25 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-02-22 09:00:11 PST; 22min ago
     Docs: man:mysql(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 2826 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 31 (limit: 2281)
    Memory: 65.6M
   CGroup: /system.slice/mariadb.service
           └─2826 /usr/sbin/mysqld
```

*Nótese que está subida la actividad al repositorio de github:
<https://github.com/jjuarez1986/Act-2-Despliegue-Aplicaciones.git>

Despliegue de aplicaciones web