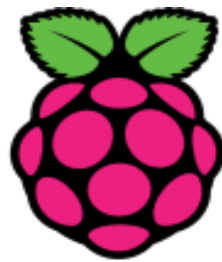


# Granja Web con



RaspberryPi

# Tabla de contenidos

[Introducción](#)

[Materiales](#)

[Preparando la Raspberry Pi](#)

[Instalación del sistema operativo](#)

[Preparación de la tarjeta SD.](#)

[Preparando el primer inicio de la raspberry pi](#)

[Instalación del sistema operativo](#)

[Configuración](#)

[IP estática](#)

[SSH](#)

[Raspberry Pi - Servidor](#)

[Instalación de APACHE](#)

[Instalación de PHP](#)

[Instalación de Mysql](#)

[Raspberry Pi - Balanceador](#)

[Ubicación final de las raspberry pi](#)

[Apertura de puertos](#)

[Configuración del Router](#)

[Rendimiento](#)

[OpenWebLoad](#)

[Seige](#)

[Replicación de Datos](#)

[Replicación de bases de datos MySQL](#)

[Configuración Maestro-Eslavo](#)

[Wordpress](#)

[Actualización automática de la IP](#)

[Instalación del cliente](#)

[Referencias](#)

# Granja Web con Raspberry Pi

## Introducción

En nuestro trabajo, vamos a simular la implementación de una Granja Web, usando para ello Raspberry Pi.

Para ésto configuraremos tres Raspberry como servidores y una como balanceador, para repartir el trabajo entre los distintos servidores.

Una vez la Granja Web esté funcionando, haremos distintas mediciones de rendimiento con Siege y OpenWebLoad. Realizaremos réplicas de datos y por último le daremos uso a la misma usando Wordpress.

## Materiales

- 2 Raspberry Pi 1
- 2 Raspberry Pi 3
- cables de alimentación
- cables ethernet
- cable hdmi
- mouse
- teclado
- pantalla
- 4 tarjetas SD 8G
- Router CG6640
- Router ARV7519RW22-A-LT VR9 1.2 (LIVEBOX)



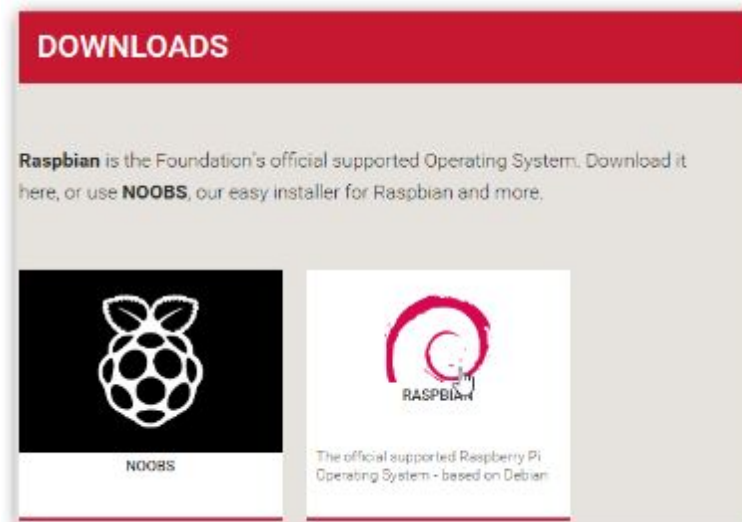
# Preparando la Raspberry Pi

## Instalación del sistema operativo

### 1. Preparación de la tarjeta SD.

En la preparación de la tarjeta hemos trabajado con el SO windows 10.

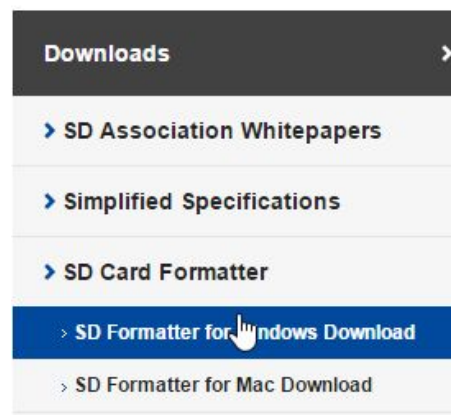
- a. Descargar el sistema operativo de la página oficial <https://www.raspberrypi.org>



- b. formatear la tarjeta con el programa descargado de la siguiente página <https://www.sdcard.org>

## Downloads

[HOME](#) > Downloads



- c. descomprimir el archivo descargado en la tarjeta SD

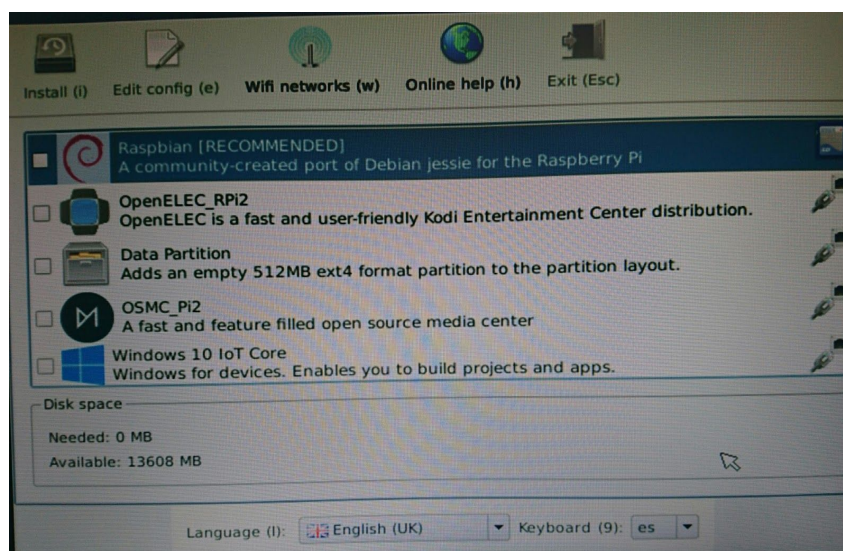
## 2. Preparando el primer inicio de la raspberry pi

- a. Insertamos la tarjeta SD que contiene el sistema operativo
- b. Conectamos el teclado, ratón, pantalla, router y por último a la fuente.

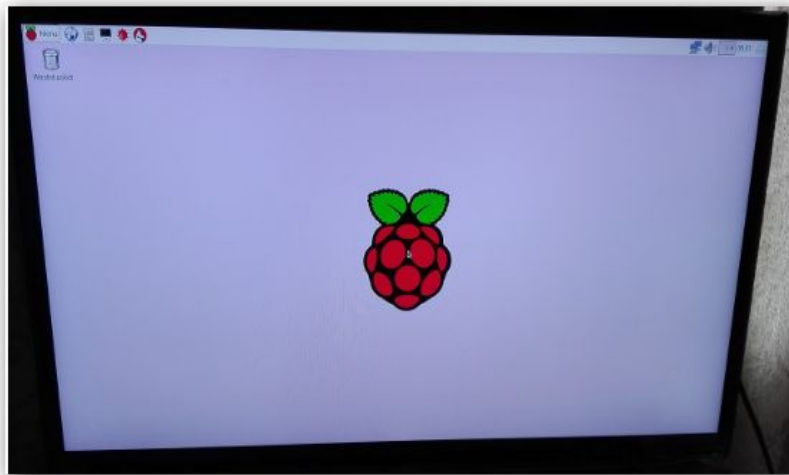


## 3. Instalación del sistema operativo

- a. Hemos descargado el archivo que está identificado como NOOBS, este viene cargado con distintos sistemas operativos, en esta ocasión para este proyecto escogemos Raspbian que es la recomendada.



- b. Una vez finalizada la instalación accedemos al escritorio.



## Configuración

### 1. IP estática

- a. Editamos el fichero `/etc/network/interfaces` para asignarle una ip estática

```
GNU nano 2.2.6      File: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpd
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

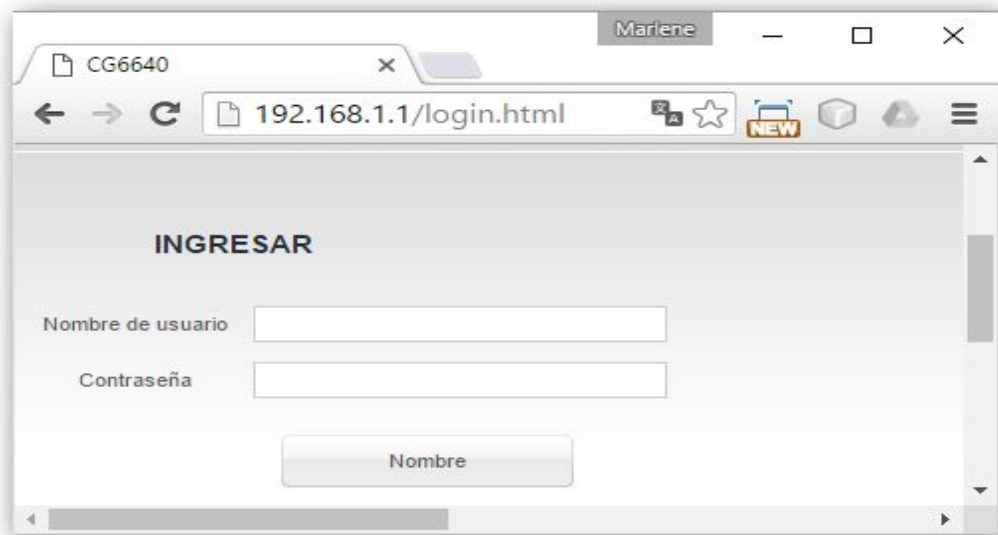
#####
iface eth0 inet static
address 192.168.1.129
netmask 255.255.255.0
gateway 192.168.1.1
#####

allow-hotplug wlan0

[ Read 21 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```



- b. En nuestro router asignamos a nuestras raspberry pi una ip estática
- i. Accedemos al panel de administrador que por defecto se encuentra en la dirección <192.168.1.1>



Dirección MAC	Dirección IP
B8:27:EB:B6:2D:DC	192.168.1.129
B8:27:EB:16:96:AD	192.168.1.131
B8:27:EB:5F:F7:34	192.168.1.130

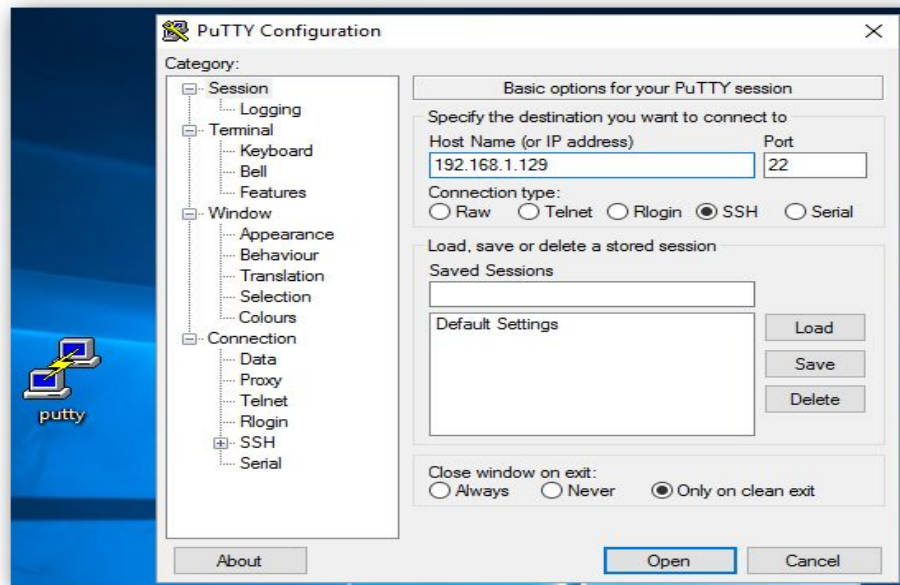
Una vez configuradas las ip de todas las raspberry pi continuaremos con la instalación de apache, php, mysql, phpmyadmin y nginx, para ello y por facilitar el trabajo usaremos ssh.



## 2. SSH

- a. Por defecto viene instalado en las raspberry pi, por lo que podremos acceder a ellas desde nuestro portátil con SO windows 10 y la aplicación [PUTTY](#).





```

pi@raspberrypi: ~
login as: pi
pi@192.168.1.129's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 12 14:17:08 2016

```

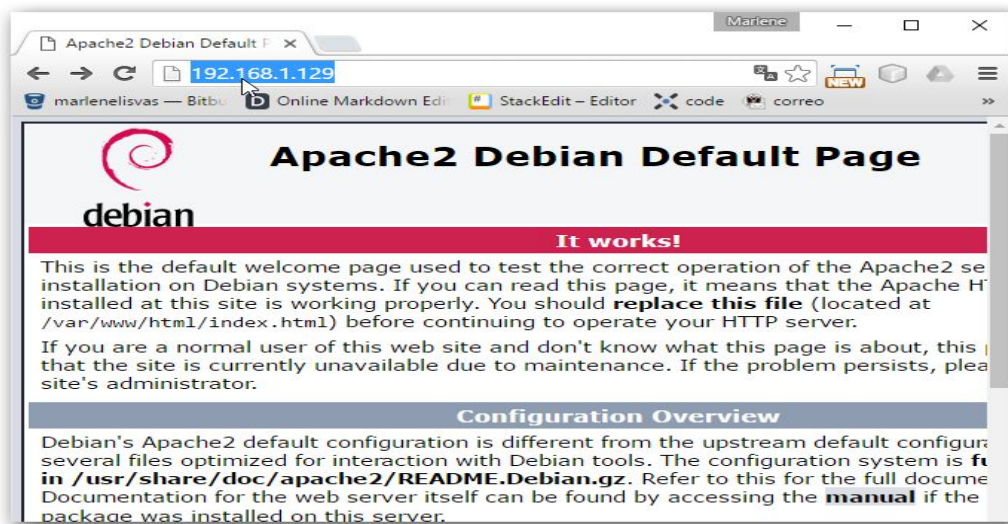
## Raspberry Pi - Servidor

### Instalación de APACHE

Instalamos apache con la siguiente orden en el terminal:

```
sudo apt-get install apache2
```

Si la instalación se ha realizado correctamente, en nuestro navegador escribimos la ip de la raspberry en la que se ha realizado la instalación y se nos mostrará la siguiente página.



## Instalación de PHP

Escribimos en el terminal

```
sudo apt-get install php5
```

Comprobamos si la instalación se ha realizado con éxito, para ello creamos un documento php cuyo contenido será lo siguiente, `<? php phpinfo(); ?>` lo guardamos en `/var/www/html`

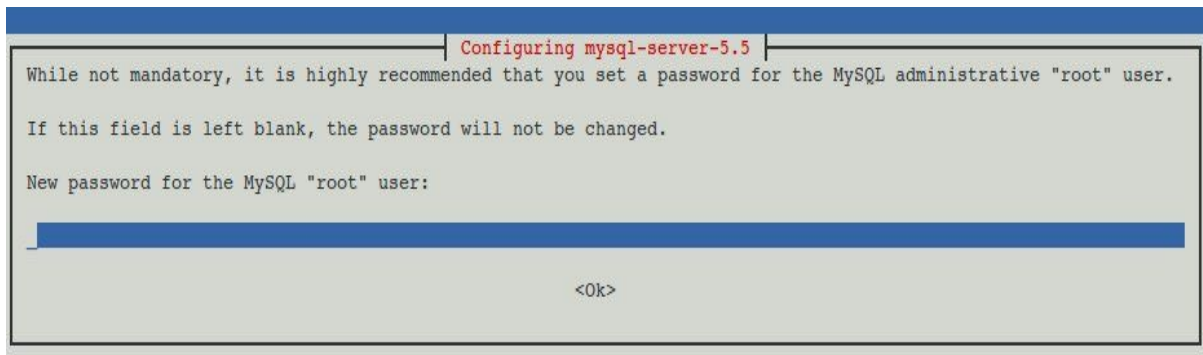


## Instalación de Mysql

Escribimos en el terminal

```
sudo apt-get install mysql php-mysql
```

escribimos la contraseña para el usuario root



Una vez finalizada las instalaciones anteriores reiniciamos apache.

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

## Raspberry Pi - Balanceador

1. Instalación del balanceador en este caso usamos NGINX.

```
sudo apt-get install nginx
```

2. Configuración

- a. Editamos el fichero de configuración

```
sudo nano /etc/nginx/sites-available/default
```

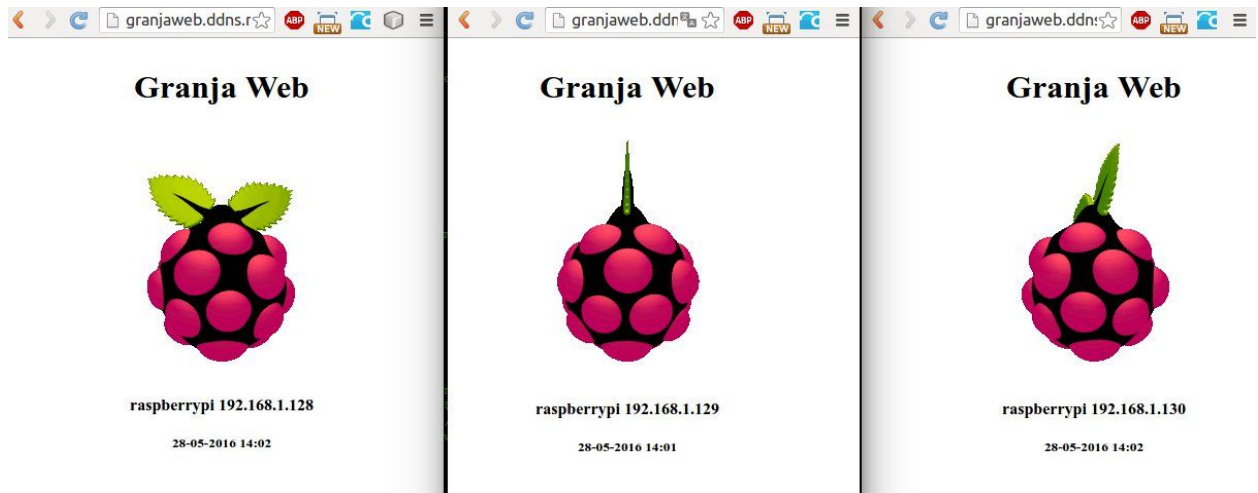
```
GNU nano 2.2.6      File: sites-available/default      Modified

upstream apaches{
    server 192.168.1.129;
    server 192.168.1.130;
}
server{
    listen 80;
    server_name balanceador;
    access_log /var/log/nginx/balanceador.access.log;
    error_log /var/log/nginx/balanceador.error.log;
    root /var/www/html;
    location /
    {
        proxy_pass http://apaches;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Al tener raspberry de distintas características el balanceo se realiza aplicando distintos pesos.

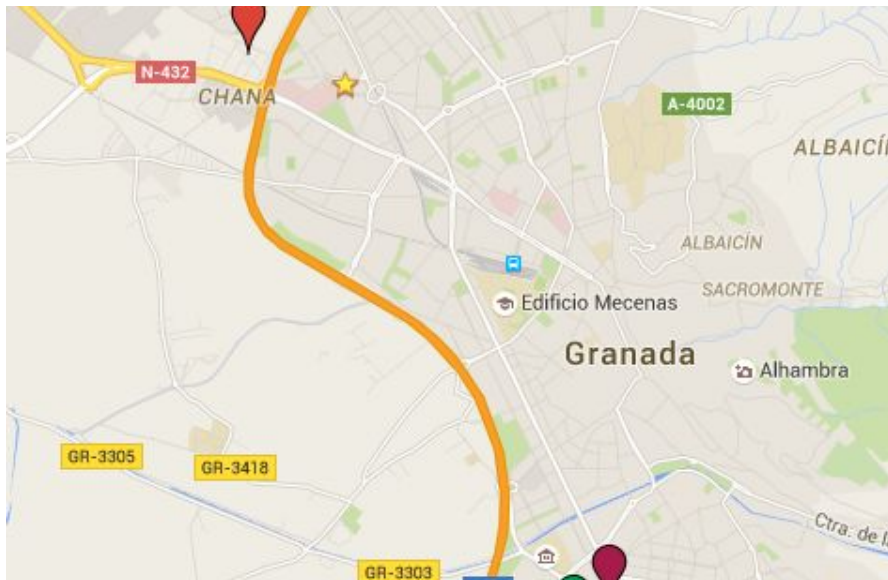
```
upstream apaches{
#       server 89.130.36.198      weight=2;#E
#       server 192.168.1.130     weight=6;#L-M
#       server 217.216.31.251    weight=5;#M
}
```

- b. reiniciamos el servicio para que se haga efectivo el cambio.  
`sudo /etc/init.d/nginx restart`
- c. Comprobamos si se realiza el balanceo. Por defecto el balanceo se realiza mediante el algoritmo de Round Robin.



## Ubicación final de las raspberry pi

Hasta el momento he trabajado con las raspberry pi dentro de la red local. Como se observa en la imagen procederemos a ubicarlas en distintas localizaciones.



## Apertura de puertos

### Configuración del Router

Accedemos al panel de configuración del router donde hemos ubicado nuestra raspberry pi que tendrá como función de ser el **servidor principal**.



Vamos a utilizar ufw para restringir el acceso a nuestra raspberry pi ,solo permitiendo el acceso por el puerto 22 para el servicio ssh.

```
sudo apt-get install ufw
sudo ufw allow 22
sudo ufw enable
```

Par ver su estado

```
sudo ufw status verbose
```

Ahora vamos a instalar fail2ban que prohibirá automáticamente las direcciones IP que no están pudiendo entrar en nuestra raspberry pi demasiadas veces.

```
sudo apt-get install fail2ban
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Reiniciamos el servicio

```
sudo service fail2ban restart
```



Añadimos los puertos que necesitamos tenerlos abiertos para un correcto funcionamiento como son el puerto 80 y el puerto 21.

```
pi@raspberrypi:~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
21 ALLOW Anywhere
80 ALLOW Anywhere
22 ALLOW Anywhere (v6)
21 ALLOW Anywhere (v6)
80 ALLOW Anywhere (v6)
```

Comprobamos las restricciones

```
sudo iptables -L
```

```
Chain ufw-user-input (1 references)
target prot opt source destination tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
ACCEPT udp -- anywhere anywhere udp dpt:ssh
ACCEPT tcp -- anywhere anywhere tcp dpt:ftp
ACCEPT udp -- anywhere anywhere udp dpt:ftp
ACCEPT tcp -- anywhere anywhere tcp dpt:http
ACCEPT udp -- anywhere anywhere udp dpt:http
```

## Rendimiento

Para comprobar el rendimiento de nuestra granja web hemos realizado 2 tipos de benchmarks, comprobamos en una sola raspberry como servidor y en la raspberry con el balanceador de carga.

Para la realización hemos usado OpenWebLoad y Seige.

## OpenWebLoad

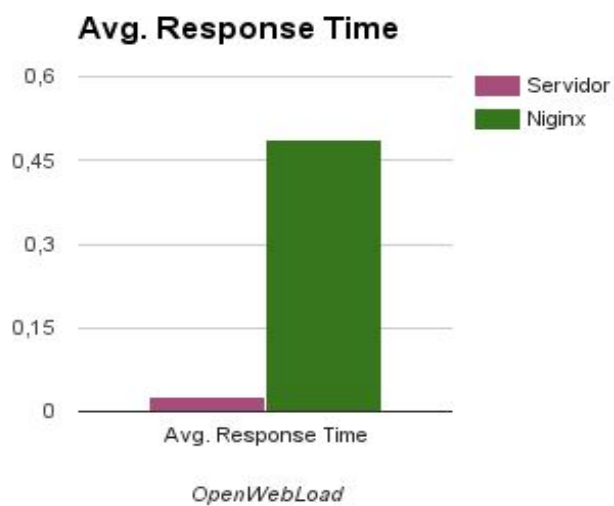
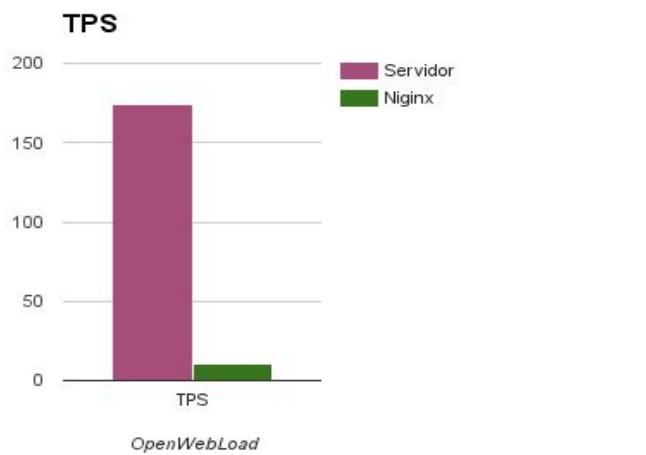
```
openload -l 60 http://192.168.1.128
```

```
openload -l 60 http://192.168.1.128
```

Servidor	1	2	3	4	5	6	7	8	9	10
TPS	176,23	170,07	173,61	174,71	175,26	173,91	174,77	174,26	174,38	175,99
Avg. Response Time	0,028	0,029	0,028	0,028	0,028	0,028	0,028	0,028	0,028	0,028
Nginx	1	2	3	4	5	6	7	8	9	10
TPS	7,74	8,89	8,69	10,15	12,53	9,95	11,45	10,39	10,93	11,50
Avg. Response Time	0,631	0,547	0,57	0,488	0,395	0,469	0,429	0,45	0,454	0,429



Servidor	Media	Desviación
TPS	174,32	1,085
Avg. Response Time	0,028	0,000
Nginx	Media	Desviación
TPS	10,22	1,138
Avg. Response Time	0,4862	0,058



## Seige

```
seige -b -t60s -v http://192.168.1.128/index.php
```

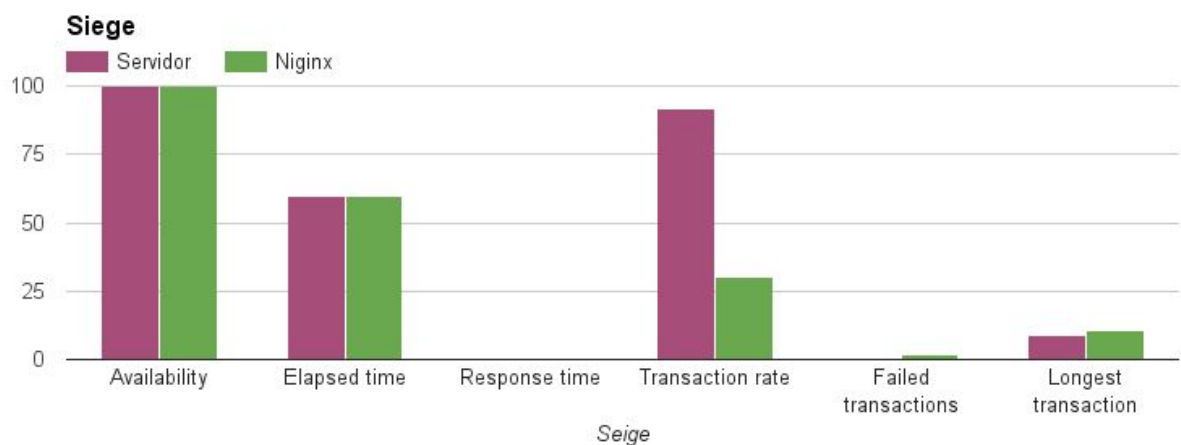
```
seige -b -t60s -v http://217.216.10.88/index.php
```

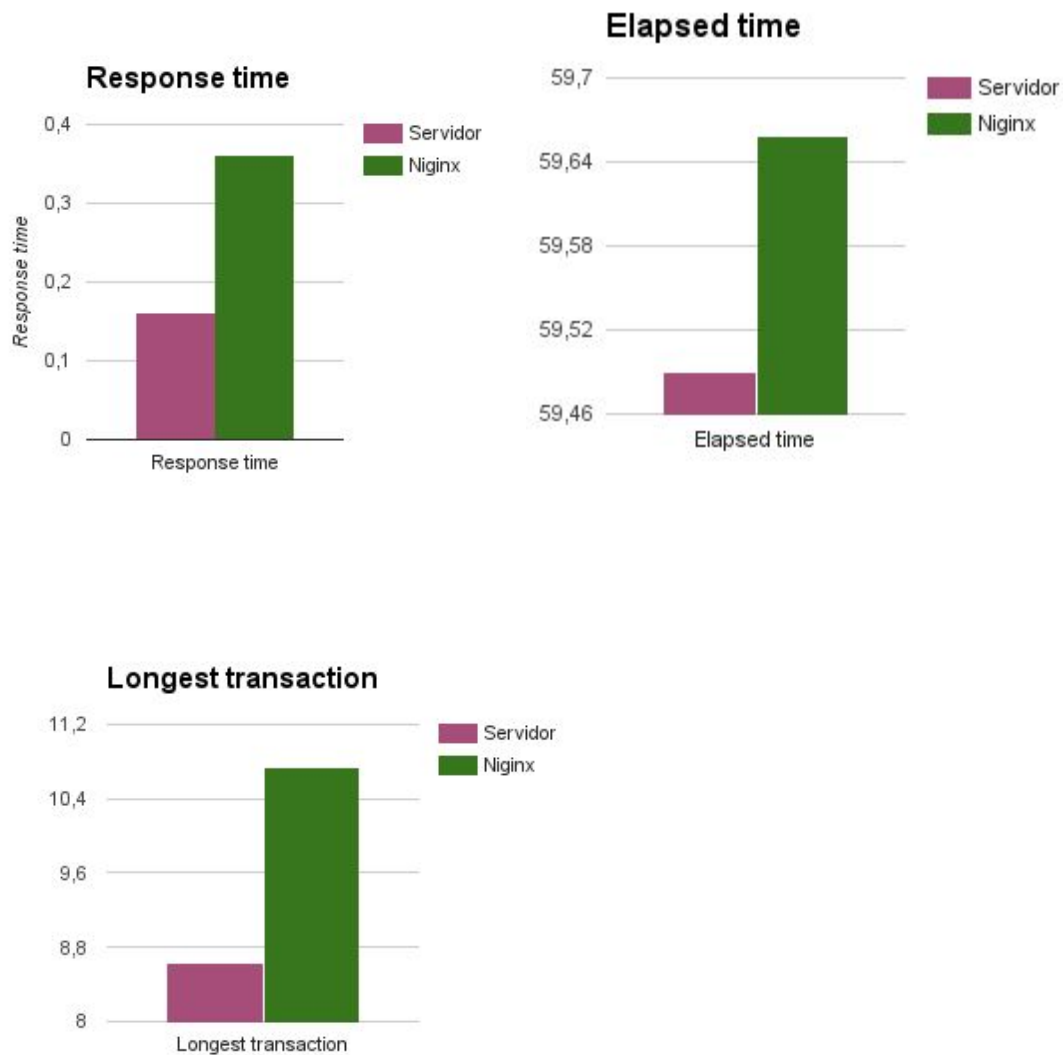
Servidor	1	2	3	4	5	6	7	8	9	10
Availability	100	100	100	100	100	100	100	100	100	100
Elapsed time	59,19	59,12	59,70	59,70	59,51	59,72	59,91	59,36	59,26	59,43
Response time	0,15	0,15	0,15	0,15	0,16	0,17	0,17	0,16	0,17	0,18
Transaction rate	99,39	101,01	96,68	96,68	92,98	87,09	86,18	89,05	86,25	82,08
Failed transactions	0	0	0	0	0	0	0	0	0	0
Longest transaction	9,08	9,06	9,06	6,13	9,09	9,06	7,58	9,08	9,09	9,08

Nginx	1	2	3	4	5	6	7	8	9	10
Availability	99,89	99,84	99,72	99,89	99,89	99,89	99,84	99,89	99,95	99,78
Elapsed time	59,94	59,61	59,31	60,00	59,06	59,89	59,85	59,88	59,97	59,07
Response time	0,41	0,40	0,34	0,37	0,39	0,42	0,40	0,43	0,42	0,04
Transaction rate	30,45	30,48	30,55	30,22	30,46	30,46	30,46	30,59	30,73	30,24
Failed transactions	2	3	0	2	2	2	3	2	1	2
Longest transaction	9,20	12,21	9,20	9,24	12,12	9,22	12,18	9,74	12,11	12,24

Servidor	media	Desviación
Availability	100	0
Elapsed time	59,49	0,22
Response time	0,16	0,01
Transaction rate	91,74	5,61
Failed transactions	0	0
Longest transaction	8,63	0,71

Nginx	media	Desviación
Availability	100	0
Elapsed time	59,66	0,32
Response time	0,36	0,07
Transaction rate	30,46	0,10
Failed transactions	2	1
Longest transaction	10,75	1,43





## Replicación de Datos

Para realizar la replica los datos utilizaremos la herramienta rsync que por defecto viene instalado.

Activamos la cuenta de root para trabajar con todos los permisos sobre todos los ficheros. Para ello ejecutamos:

```
sudo passwd root
```

Generamos un par de claves pública y privada en las raspberry pi secundarias para ello ejecutamos:

```
ssh-keygen -t dsa
```

```

root@raspberrypi:/home/pi# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
/root/.ssh/id_dsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
52:57:32:08:a3:26:d7:b1:67:e8:f8:fb:84:12:f4:64 root@raspberrypi
The key's randomart image is:
+---[DSA 1024]-----+
|      +. .o .      |
|      o =. +      |
|      . = E + .      |
|      = * + .      |
|      o + S      |
|      o o      |
|      . o .      |
|      . o      |
|      ...      |
+-----+
root@raspberrypi:/home/pi#

```

Procedemos a copiar la clave pública en el servidor principal.

```
ssh-copy-id -i .ssh/id_dsa.pub root@ip-servidor-principal
```

Finalmente automatizamos la tarea de copia de ficheros del directorio `/var/www/html/` editando el fichero `/etc/crontab` y añadiendo:

```
*/1 * * * * root rsync -avz -e ssh root@217.1.1.251:/var/www/html/ /var/www/html/
```

De esta forma se replicarán los ficheros en intervalos de 1 minutos.

<pre> pi@raspberrypi:/var/www/html \$ ls -lai total 624 1707182 drwxr-xr-x 2 root root 4096 May 27 22:22 . 1707173 drwxr-xr-x 3 root root 4096 May 27 18:38 .. 1704355 -rw-r--r-- 1 root root 346 May 28 09:47 index.php 1716157 -rw-r--r-- 1 root root 620124 May 12 12:12 ras.gif 1716164 -rw-r--r-- 1 root root 27 May 12 19:14 test.php pi@raspberrypi:/var/www/html \$ sudo nano index.php pi@raspberrypi:/var/www/html \$ date Sat 28 May 09:48:52 UTC 2016 pi@raspberrypi:/var/www/html \$ ls -lai total 624 1707182 drwxr-xr-x 2 root root 4096 May 27 22:22 . 1707173 drwxr-xr-x 3 root root 4096 May 27 18:38 .. 1704355 -rw-r--r-- 1 root root 345 May 28 09:48 index.php 1716157 -rw-r--r-- 1 root root 620124 May 12 12:12 ras.gif 1716164 -rw-r--r-- 1 root root 27 May 12 19:14 test.php pi@raspberrypi:/var/www/html \$ </pre>	<pre> pi@raspberrypi:/var/www/html \$ ls -lai total 624 545978 drwxr-xr-x 2 root root 4096 May 27 22:22 . 545977 drwxr-xr-x 4 root root 4096 May 12 11:39 .. 537180 -rw-r--r-- 1 root root 346 May 28 09:47 index.php 545523 -rw-r--r-- 1 root root 620124 May 12 12:12 ras.gif 524961 -rw-r--r-- 1 root root 27 May 12 19:14 test.php pi@raspberrypi:/var/www/html \$ date Sat May 28 09:48:32 UTC 2016 pi@raspberrypi:/var/www/html \$ ls -lai total 624 545978 drwxr-xr-x 2 root root 4096 May 27 22:22 . 545977 drwxr-xr-x 4 root root 4096 May 12 11:39 .. 536822 -rw-r--r-- 1 root root 345 May 28 09:48 index.php 545523 -rw-r--r-- 1 root root 620124 May 12 12:12 ras.gif 524961 -rw-r--r-- 1 root root 27 May 12 19:14 test.php pi@raspberrypi:/var/www/html \$ sudo nano /etc/crontab pi@raspberrypi:/var/www/html \$ </pre>
--	---

## Replicación de bases de datos MySQL

Vamos a crear una base de datos y realizar su réplica.

- Accedemos al terminal de mysql  

```
sudo su mysql -uroot
```
- Creamos la base de datos.  

```
create database wordpress;
```
- Creamos una tabla de prueba.  

```
use wordpress;  
create table datos_prueba(nombre varchar(100), tlf int);
```

## Configuración Maestro-Esclavo

En el servidor que será el **maestro** realizamos la siguiente configuración:

- Editamos el fichero de configuración ***/etc/mysql/my.cnf***
  - `#bind-address 127.0.0.1`
  - Establecemos el identificador del servidor **server-id =1**
  - Indicamos donde almacenar el log de errores  

```
log_error = /var/log/mysql/error.log
```
- Reiniciamos el servicio  

```
/etc/init.d/mysql restart
```

Ahora pasamos al servidor esclavo realizamos los pasos anteriores realizados en el maestro pero en el identificador del servidor será **server-id = 2**

Volvemos al maestro y creamos el usuario que tenga permisos de acceso para la replicación.

```

pi@raspberrypi: ~
root@raspberrypi:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.5.44-0+deb8u1-log (Raspbian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE USER esclavo IDENTIFIED BY 'esclavo';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'esclavo'@'%' IDENTIFIED BY 'esclavo';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH TABLES;
Query OK, 0 rows affected (0.09 sec)

mysql> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      501 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

En la máquina esclava damos los datos del maestro.

```

pi@raspberrypi: ~
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.44-0+deb8u1-log (Raspbian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CHANGE MASTER TO MASTER_HOST='217.216.31.251', MASTER_USER='esclavo', MAS
TER_PASSWORD='esclavo', MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=501,
MASTER_PORT=3306;
Query OK, 0 rows affected (0.17 sec)

mysql> START SLAVE;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW SLAVE STATUS\G;

```



El dato que nos interesa del resultado `SHOW SLAVE STATUS\G;` es “Seconds\_Behind\_Master” si es distinto de null, todo está funcionando correctamente.

```

Relay_Log_Space: 410
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
1 row in set (0.00 sec)

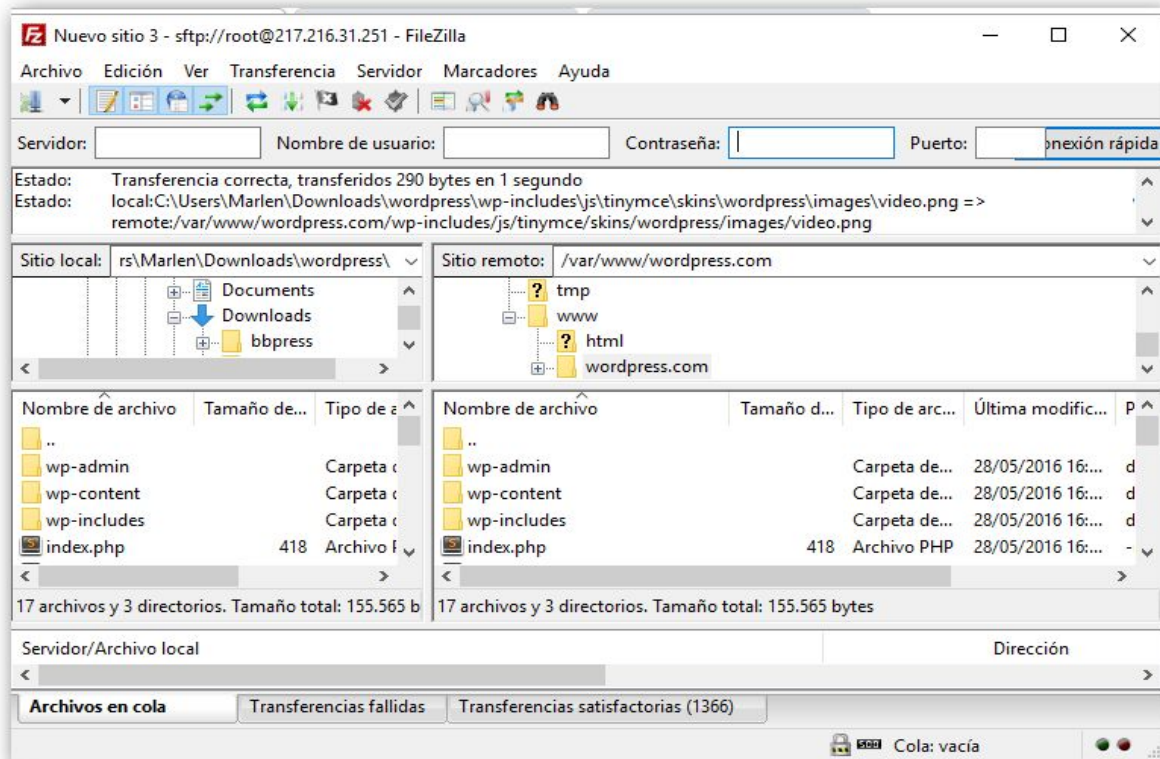
```

## Wordpress

Para darle uso a nuestra granja web con raspberry pi vamos a instalar wordpress.

The screenshot shows the WordPress.org website in Spanish. The header includes the WordPress logo, "WORDPRESS.ORG", and "Español". Navigation links include Inicio, Temas, Plugins, Soporte, Colabora, Contacto, Blog, Descargas, and Meetups. The main content area features a welcome message "¡Bienvenida! ¡Bienvenido!" and a description of WordPress as a software for creating websites, blogs, or applications. It mentions that WordPress is free and open-source, and that it is used by millions of people. A sidebar on the right shows a preview of the WordPress dashboard. At the bottom, there is a "Descargar" (Download) section with a button to "Descargar WordPress 4.5.2" and links to download the .zip (8.5 MB) or .tar.gz (7.8 MB) files.


Procedemos a subir los ficheros descargados a nuestras raspberry pi .



Para realizar la instalación en nuestro navegador escribimos la ip de cada una de las raspberry.



Rellenaremos los datos de la base de datos creada anteriormente:



A continuación deberás introducir los detalles de conexión a tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web.

Nombre de la base de datos	<input type="text" value="wordpress"/>	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	<input type="text" value="nombre_de_usuario"/>	El nombre de usuario de tu base de datos.
Contraseña	<input type="text" value="contraseña"/>	La contraseña de tu base de datos.
Servidor de la base de datos	<input type="text" value="localhost"/>	Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

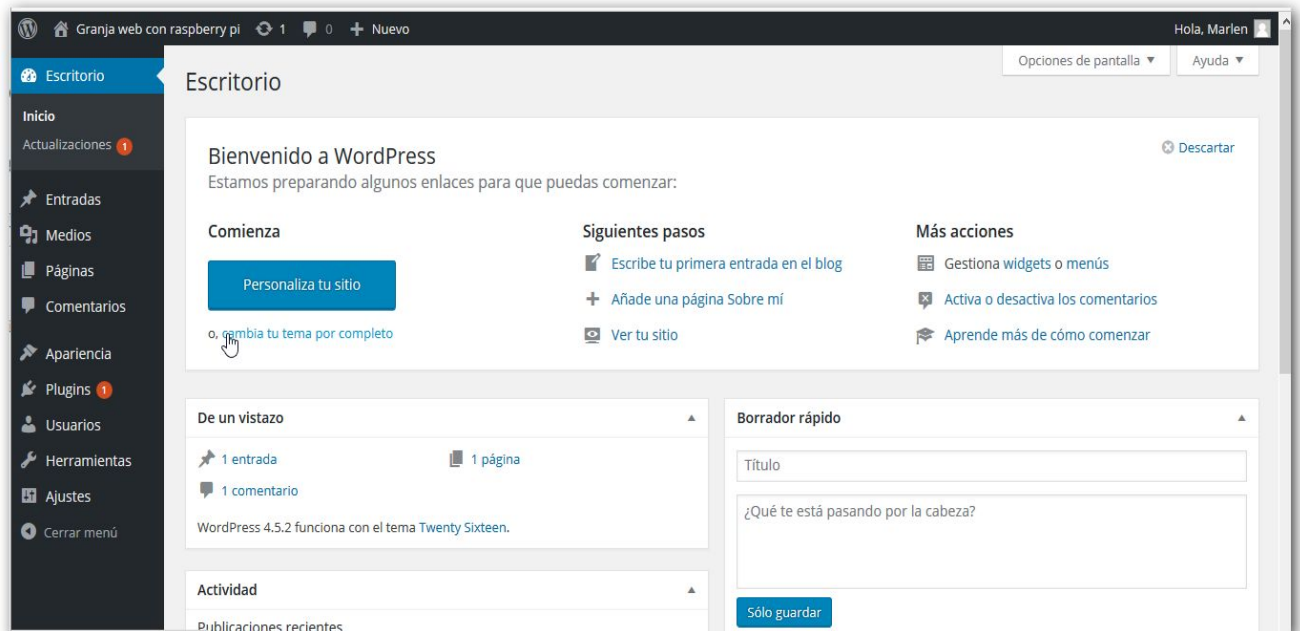
Una vez ingresado los datos necesarios procedemos con la instalación.



¡Todo bien amigo! Ya has terminado esta parte de la instalación. Ahora WordPress se puede comunicar con tu base de datos. Si estás listo es hora de...

Al finalizar la instalación tendremos que rellenar un formulario con los datos para la cuenta de usuario.

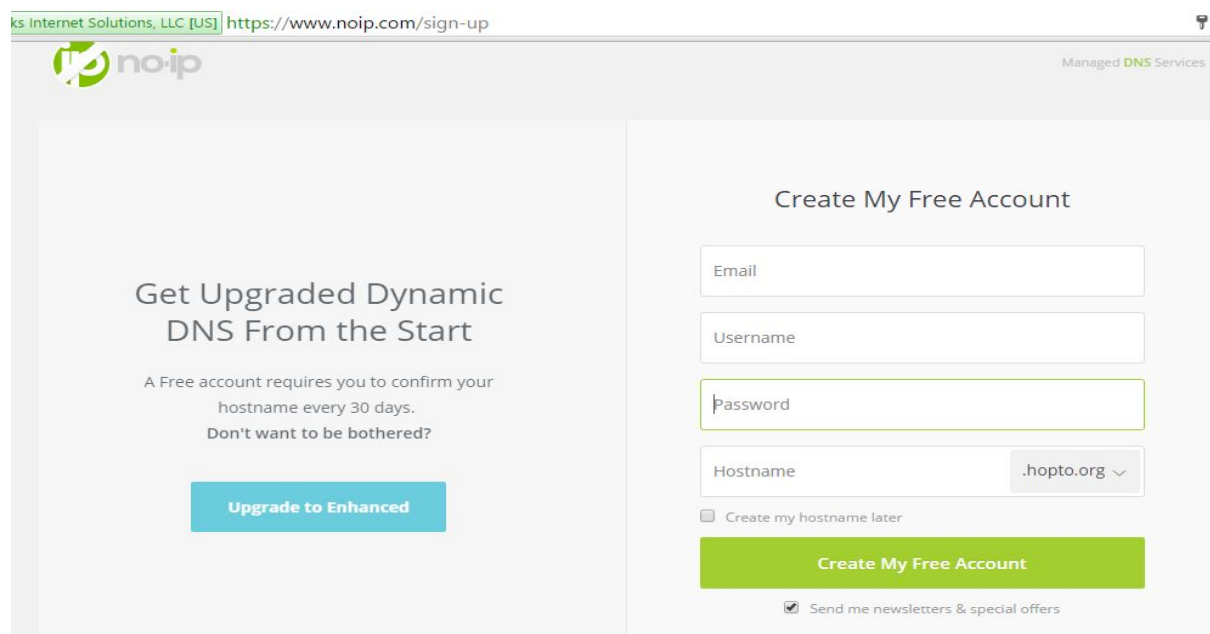
Accedemos a nuestra cuenta.



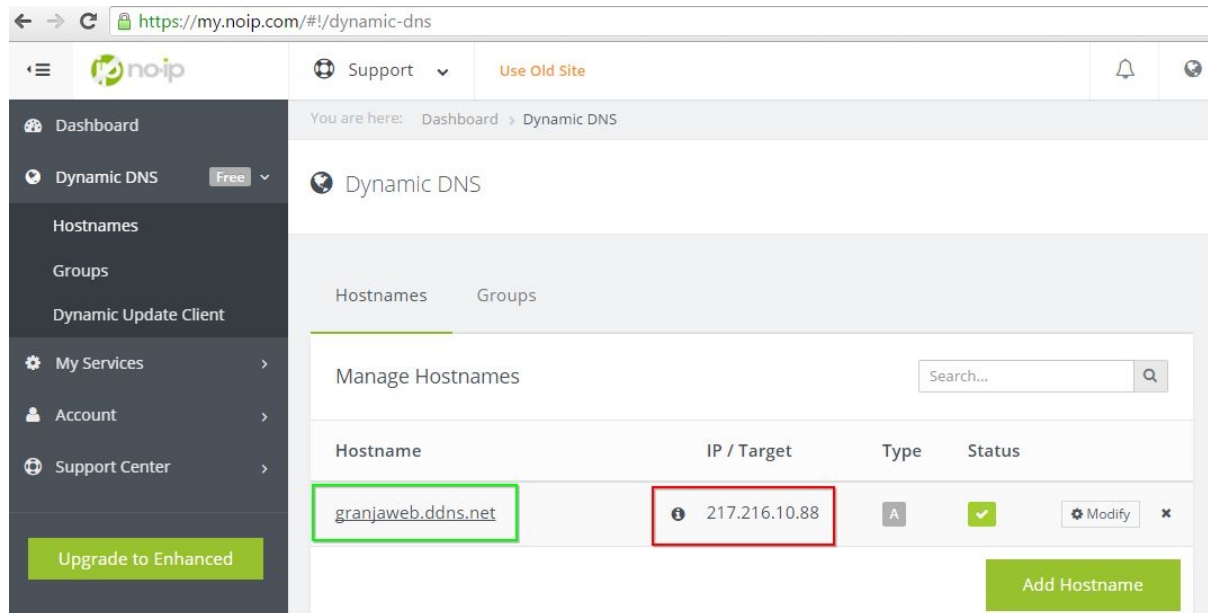
Como vamos a acceder a nuestro blog a través de la ip que tiene asignada por el router de nuestro proveedor, es posible que esté asignada dinámicamente.

## Actualización automática de la IP

Para ello usamos una herramienta gratuita, nos creamos una cuenta en [NO-IP](https://www.noip.com/sign-up)



Accedemos al panel de administración y asociamos nuestra ip.



## Instalación del cliente

La instalación del cliente lo realizamos en la raspberry con el balanceador.

Los pasos siguientes tenemos que hacerlo con el usuario root.

- Creamos la carpeta noip  
`mkdir noip`
- nos ubicamos en ella  
`cd noip`
- descargamos los ficheros para proceder con la instalación.  
`wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz`
- descomprimir  
`tar xzf noip-duc-linux.tar.gz`
- Nos ubicamos en la carpeta no-ip-2.1.9  
`cd no-ip-2.1.9`
- ejecutamos  
`make`
- Una vez finalizado el make  
`make install`

Durante la instalación nos pedirá el nombre de usuario la cuenta, password y el hostname.

Y con esto podremos acceder a nuestro sitio a través de la siguiente dirección <http://granjaweb.ddns.net>.



## Referencias

- Guiones de prácticas de la asignatura Servidores Web de Altas Prestaciones.
- <https://www.raspberrypi.org/>
- <http://sobrebites.com/montar-un-servidor-casero-con-raspberry-pi-parte-1-instalar-raspbian-en-una-tarjeta-sd/>
- <https://developer.ubuntu.com/en/snappy/start/raspberry-pi-2/>
- <http://www.rpi.uroboros.es/servers.html>
- <https://geekytheory.com/tutorial-raspberry-pi-crear-servidor-web/>
- <http://diymakers.es/raspberry-pi-como-servidor-web/>
- <https://www.raspberrypi.org/documentation/remote-access/ssh/>
- <http://comohacer.eu/guia-noobs-raspberry-pi/>
- <https://www.noip.com/>
- <http://www.modmypi.com/blog/tutorial-how-to-give-your-raspberry-pi-a-static-ip-address>
- <http://blog.self.li/post/63281257339/raspberry-pi-part-1-basic-setup-without-cables>