# Laboratorio de Sistemas Operativos y Redes The Bomb Game

Integrantes: Maximiliano Díaz, Liza Chambi Orellana, Rosalí Zaracho Profesor: José Luis Di Biase

1er cuatrimestre 2016



# Introducción:

The Bomb Game es un proyecto dividido en dos partes, que tiene como propósito vincular los conocimientos adquiridos entre las materias Laboratorio de Redes y Sistemas Operativos e Introducción a la Programación de Microcontroladores con Tecnologías Libres. El proyecto se basa en un juego que tiene como objetivo desactivar una bomba, el cual se puede jugar de manera local o en red. Los resultados de las partidas en red son publicados en Twitter, como también en la tabla de posiciones de la página web del proyecto http://www.thebombgame.com.ar.

En esta parte del proyecto nos enfocaremos en la instalación y configuración de distintos programas que nos permitirán crear una página web, asociarla a una ip, tener un servidor de correo electrónico y utilizar una herramienta llamada Node-RED, que nos permitirá registrar los usuarios, actualizar los puntajes en la página web, publicar en twitter, como también avisar a los administradores del proyecto en caso de recibir un mensaje. A continuación explicaremos brevemente cada uno de los programas utilizados:

**Bind:** Es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, que nos permite asociar una ip con un dominio y viceversa.

**Node-RED:** Es una nueva herramienta de código abierto creada por el equipo de IBM, que nos provee de una interfaz basada en un navegador web que nos permite crear flujos de eventos e interconectar todos ellos a través de un ligero entorno de trabajo y desarrollo. Está construido en Node js.

WordPress: Es el CMS (Sistema de Manejo de Contenido) mas popular en internet. Permite que puedas configurar blogs y sitios web de manera flexible por encima de un backend con MySQL y procesamiento por PHP. WordPress es visto como una gran elección a la ahora de hacer que un sitio web que funcione rápidamente. Más del 20 % de la red utiliza esta tecnología.

**SquirreMail:** Es una aplicación webmail escrita en PHP. Puede ser instalado en la mayoría de servidores web siempre y cuando éste soporte PHP y el servidor web tenga acceso a un servidor IMAP y a otro SMTP.

# Características de la maquina utilizada:

Se utilizó una maquina virtual, el cual tiene un SO Linux Debian, pero previamente las instalaciones y configuraciones de los programas fueron hechos en un SO Linux Ubuntu 14.04 de manera local, de la cual se basa esta guía. Además registramos el dominio Thebombgame.com.ar a través de https://nic.ar/, para poder crear la página web del proyecto.

# Instalación y Configuración:

Antes de comenzar las instalaciones y configuraciones de los programas es recomendable acceder a la terminal como root (sudo su) para no tener problemas de permisos denegados.

**Bind:** Instalamos Bind9 para poder asociar la página web con la ip del servidor.

• Instalar servidor de DNS con Bind9:

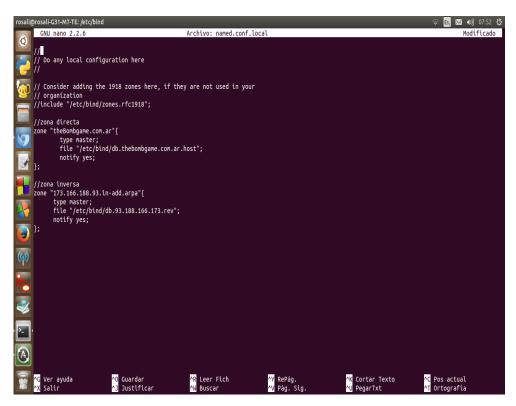
#### apt-get install bind9

• Configuración del servidor DNS: Nos ubicamos en el directorio del programa para editar los archivos de configuración necesarios:

#### cd /etc/bind/

Editamos el archivo nano named.conf.local. En este archivo especificaremos las zonas de búsqueda directa e inversa del servicio DNS. El dominio de nuestra zona directa y la subred del la zona inversa. También tendremos que incluir qué tipo de servicio es (maestro o esclavo) y en que archivos hará la búsqueda de nombres.

nano named.conf.local

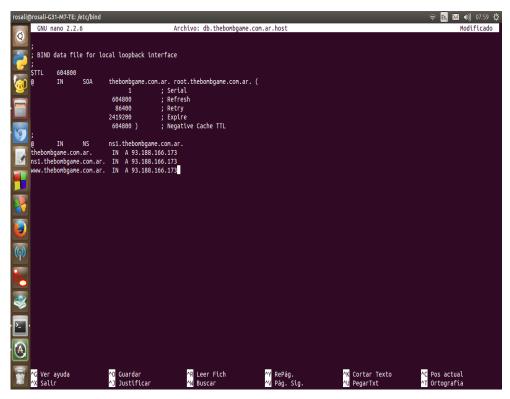


Podemos comprobar que no tenemos errores de sintaxis en el archivo, por medio del comando **named-checkconf**. Si no devuelve nada, significa que no hay errores. De lo contrario nos especificará cual es el error de sintaxis y en que linea se encuentra el error.

Creamos los archivos db.<br/>thebombgame.com.ar.host y db.93.188.166.173.rev con los comandos:

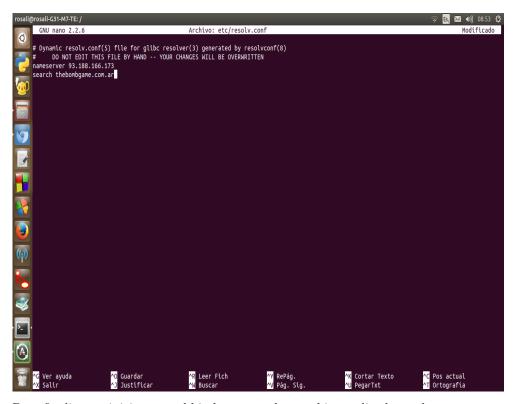
# cp db.local db.thebombgame.com.ar.host cp db.local db.93.188.166.173.rev

Editamos primero el archivo que contendrá las tablas de la búsqueda directa /etc/bind/db.thebombgame.com.ar.host. Las primeras líneas son unos parámetros relacionados con la actualización del DNS (número de serie y periodos de actuación). La siguiente línea indica quién es el servidor primario (NS = Name Server). Las siguientes líneas especifican las @IP's de los diferentes PC's componentes del dominio (A = Address). En la siguiente imagen podemos ver como quedaría el archivo:



Luego editamos el archivo /etc/bind/db.93.188.166.173.rev. Este archivo contiene las tablas de búsqueda inversa. Así quedaría el archivo:

También se comprobar que estos dos ficheros que contienen las tablas de búsqueda directa e inversa estén sintácticamente correctos con el comando named-checkzone thebombgame.com.ar /etc/bind/db.thebombgame.com.ar .host para el fichero de la zona directa y named-checkzone thebombgame.com.ar /etc/bind/db.93.188.166.173.rev para el fichero de la inversa. Si están correctos nos devolverá un **OK**. Una vez configurado nuestro servidor DNS, debemos indicar a nuestro PC que el servidor DNS es él mismo, se especifica editando el archivo /etc/resolv.conf. Indicando la @ip del servidor y el dominio de donde realizará las búsquedas.



Para finalizar, reiniciaremos el bind para que los cambios realizados en los archivos anteriores tengan efecto. Lo haremos con el comando: /etc/init.d/bind9 restart.

#### Node-RED:

• Instalar Node Js: Node Js es un intérprete Javascript del lado del servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que maneje decenas de miles de conexiones simultáneas en una sólo una máquina física.

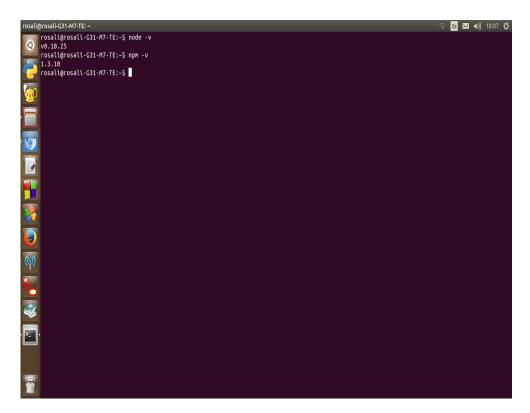
Aclaración: Node-RED soporta versiones de Node j<br/>s a partir de  $0.10.x\ y$  superiores.

## apt-get install nodejs

Luego instalamos el package manager, npm:

## ln -s /usr/bin/nodejs /usr/bin/node

Ahora deberíamos tener node js y npm instalados.



#### • Instalar Node-RED:

La manera mas fácil de instalar Node-RED es usando el comando npm anteriormente instalado:

# sudo npm install -g –unsafe-perm node-red

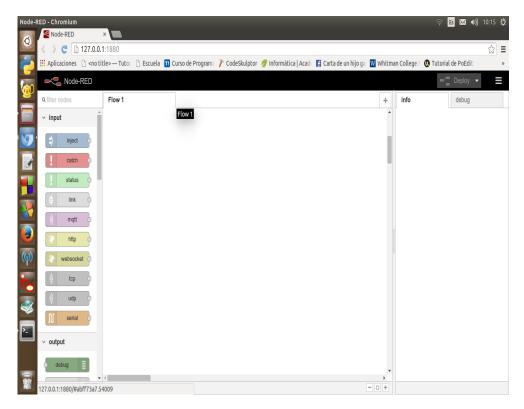
• Probar Node-RED:

```
rosalignosali-G31-M7-TE:-$ node-red

| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node-red
| cosalignosali-G31-M7-TE:-$ node
```

Ahora lo podemos probar desde el navegador:

http://localhost:1880.



• Instalar nodo MySQL: Vamos a necesitar instalar este nodo para que Node-Red pueda leer y escribir en bases de datos de mysql. Desde el directorio /.node-red escribimos:

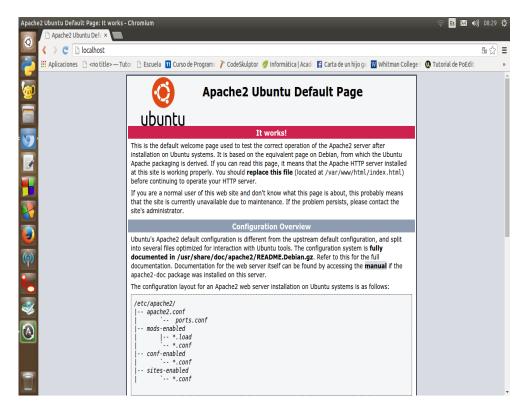
npm install node-red-node-mysql

#### WordPress:

• Instalar Apache2: Apache2 es un potente servidor HTTP de código abierto. Con el podemos montar un servidor web de una forma sencilla y segura. Para poder instalar un servidor de correo que sea accesible vía web, es indispensable instalar Apache.

# apt-get install apache2

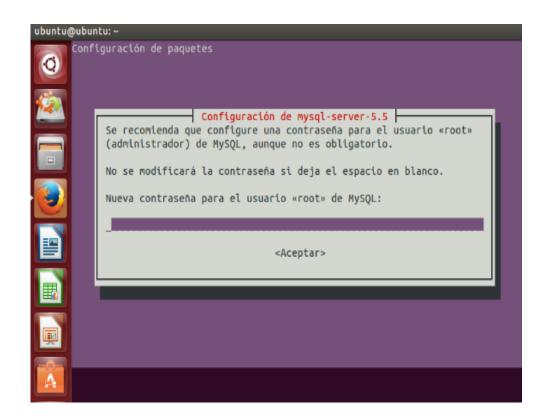
Si la instalación es correcta, abriendo un navegador e introduciendo localhost nos aparecerá la siguiente imagen:

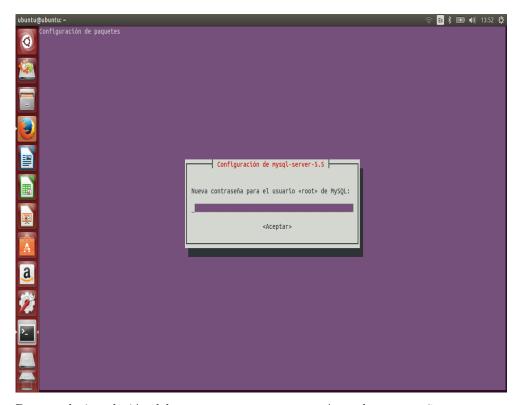


• Instalar MySQL: Una vez que tenemos el servidor web, el siguiente paso es instalar la base de datos. MySQL es un sistema de base de datos que nos va a permitir almacenar los datos de las páginas o aplicaciones web.

#### sudo apt-get install mysql-server php5-mysql

El paquete mysql-server instalará el servidor MySQL y el paquete php5-mysql dará el soporte de PHP a la base de datos. Nos mostrara la siguiente ventana en donde estableceremos una contraseña.





Durante la instalación del paquete se nos preguntará por la contraseña del superusuario (root) para acceder a la base de datos. Esta cuenta administrativa de MySQL tiene todos los privilegios de administración, para poner un ejemplo es similar a la cuenta de root del propio servidor, por tanto esta contraseña debe ser muy segura, siendo aconsejable el uso de un mínimo de ocho caracteres, con mayúsculas y minúsculas, números y caracteres especiales permitidos. Una vez la instalación ha sido completada, el siguiente paso es ejecutar algunos scripts adicionales que provee el paquete **mysql-server** para asegurar la base de datos.

Lo primero es indicarle a la base de datos que cree la estructura de directorios donde se almacenará la información. Esto se consigue ejecutando el siguiente comando:

#### sudo mysql\_install\_db

Además de esto, vamos a ejecutar un script de seguridad que eliminará algunas características por defecto peligrosas y limitará el acceso a la base de datos. Dicho script se ejecuta del siguiente modo:

#### sudo mysql\_secure\_installation

Necesitaremos introducir la contraseña que hemos puesto durante la instalación para la cuenta **root de MySQL**. Lo siguiente que nos preguntará

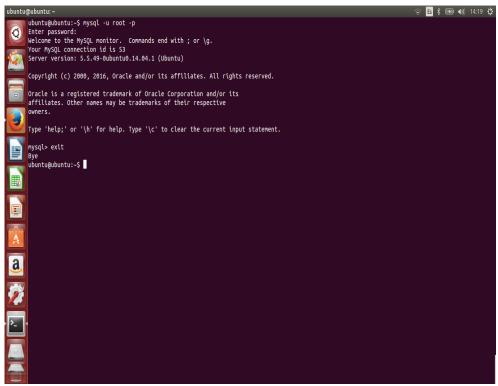
es si queremos cambiar la contraseña. Si la hemos definido bien, no habría necesidad de volver a modificarla, por tanto indicamos "no" en el prompt.

Para el resto de cuestiones, simplemente vamos a pulsar la tecla "ENTER" para dejar las opciones por defecto que ya aseguran la configuración. Esto eliminará los usuarios y bases de datos de ejemplo, deshabilitará el acceso remoto del usuario root, y cargará estas nuevas reglas en el servicio.

Una vez realizado todo esto, ya tendremos el servicio instalado y correctamente configurado. Para comprobar si podemos entrar a mysql escribimos el siguiente comando:

#### mysql -u root -p

Nos pedirá la contraseña que establecimos en el paso anterior.



• Instalar PHP: Php es el componente de la configuración que nos dará el soporte para cargar contenido dinámico en nuestras webs. Este puede ejecutar scripts, conectarse a la base de datos de MySQL para obtener información, y facilitar contenidos procesados hacia el servidor web para poder mostrarlos.

#### apt-get install php5 libapache2-mod-php5 php5-mysql

Con este comando estamos instalando todas las librerías necesarias para integrar php con el servidor de apache, luego haremos una prueba del

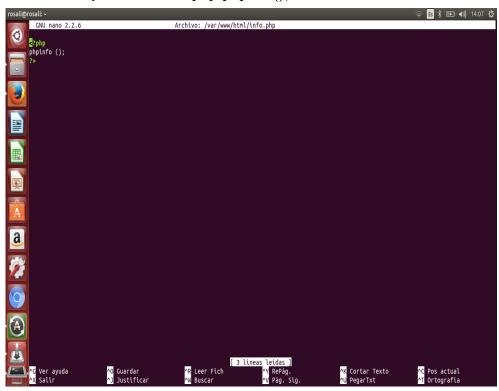
Procesador PHP en el Servidor Web para esto reiniciamos el servicio de apache:

# /etc/init.d/apache2 restart

Con el fin de probar que nuestro sistema se ha configurado correctamente para PHP, podemos crear un script PHP muy básico para comprobar el funcionamiento de PHP:

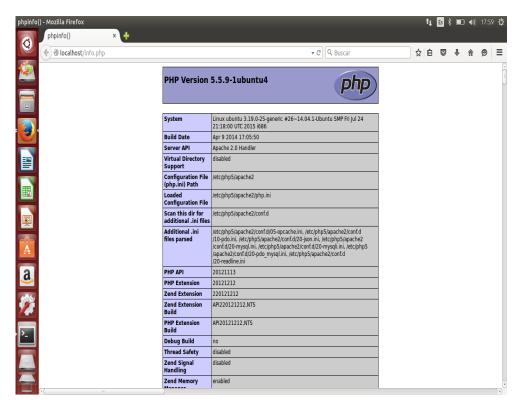
## nano /var/www/html/info.php

Dentro del script escrbimos: <?php phpinfo(); ?>



Comprobamos entrando a la dirección:

http://127.0.0.1/info.php



Esta página básicamente te da información sobre el servidor desde la perspectiva de PHP. Es útil para la depuración y para asegurarse de que los ajustes se están aplicando correctamente.

Si esto fue un éxito, entonces su PHP está funcionando como se esperaba.

Es posible que desees eliminar este archivo después de esta prueba, ya que en realidad podría dar información sobre el servidor a los usuarios no autorizados. Para ello, puede escribir lo siguiente:

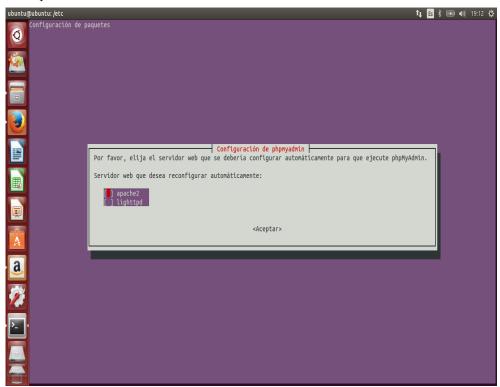
#### sudo rm /var/www/html/info.php

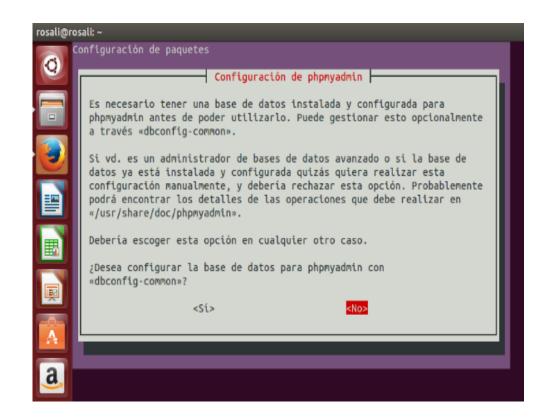
Siempre se puede volver a crear esta página si necesita acceder a la información nuevamente.

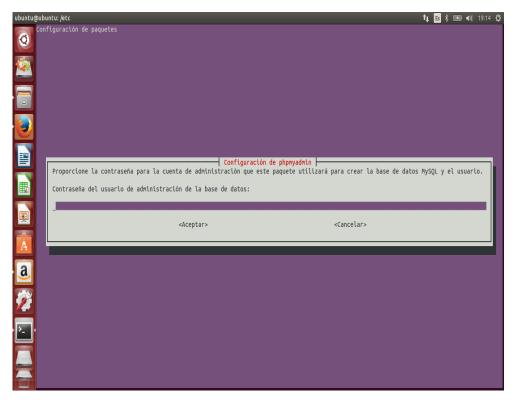
Instalar PhpMyAdmin (opcional): Es una herramienta escrita en PHP
con la intención de manejar la administración de MySQL a través de
páginas web, utilizando Internet. Actualmente puede crear y eliminar
Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir
campos, ejecutar cualquier sentencia SQL, administrar claves en campos,
administrar privilegios, exportar datos en varios formatos y está disponible
en 72 idiomas.

#### apt-get install phpmyadmin

Preguntará para que servidor web configurar, elegir **apache2** marcándolo con la barra espaciadora y continuar. Luego pedirá configurar la base de datos con dbconfig-common elegir que **NO**. hacemos las configuraciones correspondientes:

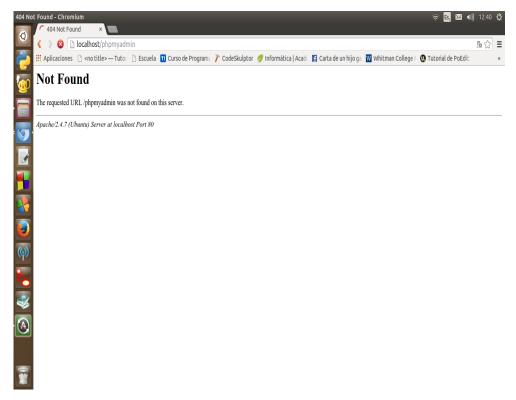




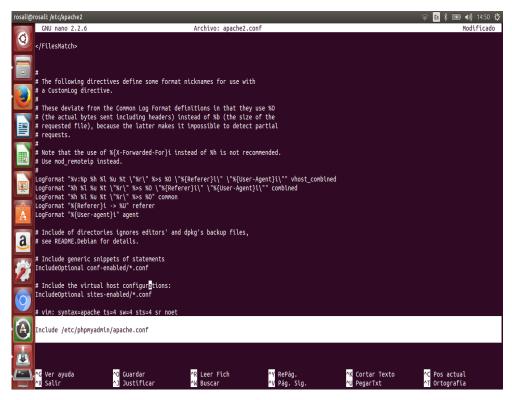


Comprobamos si funciona entrando a:

 $\rm http://127.0.0.1/phpmyadmin$ 



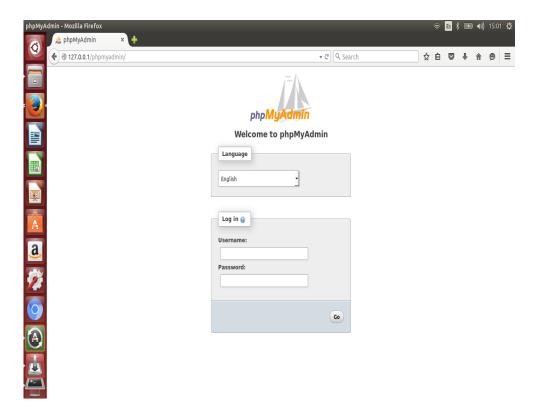
En caso de no funcionar hay que editar /etc/apache2/apache2.conf , Agregando: Include /etc/phpmyadmin/apache.conf



Luego reiniciamos Apache y listo. /etc/init.d/apache2 restart

Probamos nuevamente:

http://127.0.0.1/phpmyadmin



#### • Instalar WordPress:

Antes de instalar Wordpress debemos crear una base de datos y un usuario para su funcionamiento, ya que Wordpress utiliza una base de datos relacional para administrar y almacenar la información del usuario y el sitio. Para poder crearlos podemos hacerlo por medio de comandos ó a través de PhpMyAdmin. En este tutorial veremos las dos opciones.

Con comandos. Para empezar, ingresamos en la cuenta root (administrador) de MySQL usando el comando:

#### mysql -u root -p

Se te solicitará la clave que le diste a la cuenta root de MySQL cuando instalamos el programa. Tras esto aparecerá una consola de comandos de MySQL. Primero, podemos crear una base de datos separada que WordPress pueda controlar. Puedes llamarla como quieras, pero en la guía la llamaremos wordpress. Escribimos este comando para crear la base de datos:

#### CREATE DATABASE wordpress;

Toda sentencia de MySQL debe finalizar con punto y coma (;), así que asegúrate de que está presente si estás teniendo problemas. Ahora vamos a crear una cuenta de usuario de MySQL separada que usaremos exclusivamente para operar con nuestra nueva base de datos.

Una vez creada la base de datos, vamos a crear el usuario y password que utilizará esta base de datos para el wordpress.

# CREATE USER usuarioWordpress@localhost IDENTIFIED BY 'contrasenhaUsuario';

Ahora ya tenemos la base de datos y la cuenta de usuario, cada uno hechos específicamente para WordPress. Sin embargo, estos dos componentes no tienen relación alguna aún. El usuario no tiene acceso a la base de datos. Para solucionarlo vamos a darle acceso al usuario a nuestra base de datos con este comando:

# GRANT ALL PRIVILEGES ON wordpress.\* TO usuarioWordpress@localhost;

Ahora el usuario tiene acceso a la base de datos. Necesitamos hacer un flush a los privilegios para que la instancia de MySQL sepa acerca de los cambios que hemos realizado sobre los privilegios recientemente:

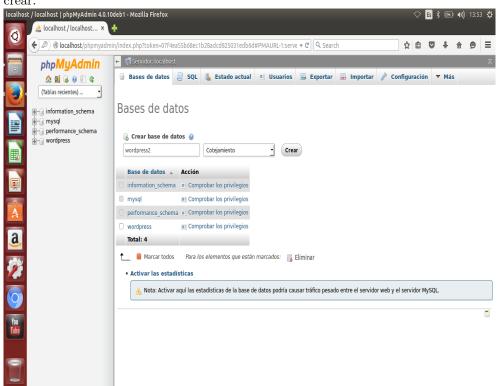
#### FLUSH PRIVILEGES:

Ahora ya tenemos todo listo. Podemos salir de la línea de comandos de MySQL escribiendo:

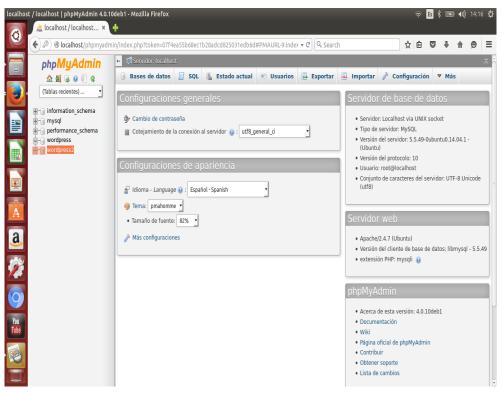
exit

 Con PhpMyAdmin: Ingresamos a localhost/phpmyadmin: Introducimos el usuario root y su contraseña.

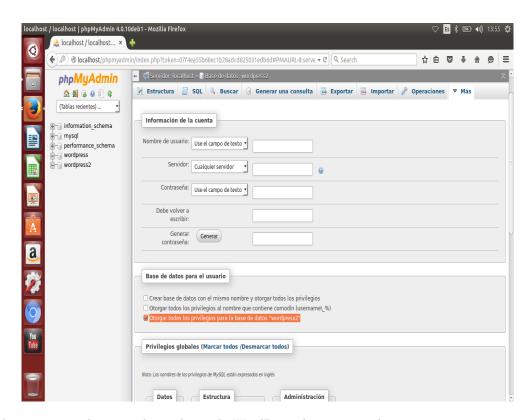
Del menú que nos aparece pulsamos sobre la opción **Bases de Datos**. En este punto le damos nombre a la base de datos y pulsamos en crear.



Una vez creada, vemos el nombre de la base de datos en lado izquierdo. Ingresamos a la base de datos.



Dentro de la base de datos elegimos la opción Mas> privilegios, para crear un usuario y contraseña. Creamos el usuario y damos check al "Darle privilegios a la base de datos recién creada".



Ahora vamos a descargar los archivos de WordPress de su sitio web:

#### cd

#### wget http://wordpress.org/latest.tar.gz

Esto descargará la versión en inglés, para obtener la versión en español deberíamos de entrar en la web de WordPress en español (http://es.wordpress.org/) y continuar hasta el link al archivo .tar.gz y copiar la url en la línea de comandos, usándola en lugar de la url en inglés.

Con esto habremos descargado el archivo comprimido con el contenido de WordPress en nuestra carpeta raíz. Podemos extraer los archivos para reconstruir el directorio de WordPress escribiendo:

#### tar xzvf latest.tar.gz

Esto creará un directorio llamado wordpress en nuestra carpeta raíz. Así mismo deberíamos de conseguir algunos paquetes más que necesitamos. Podemos obtenerlos directamente desde los repositorios por defecto de Ubuntu después de que hayamos actualizado nuestro índice local de paquetes:

## apt-get update

Ahora vamos a descargar el siguiente paquete si no lo tenemos aún instalado, ya que nos servirá para trabajar con imágenes y nos dejará instalar

plugins en nuestro sitio web.

#### apt-get install php5-gd libssh2-php

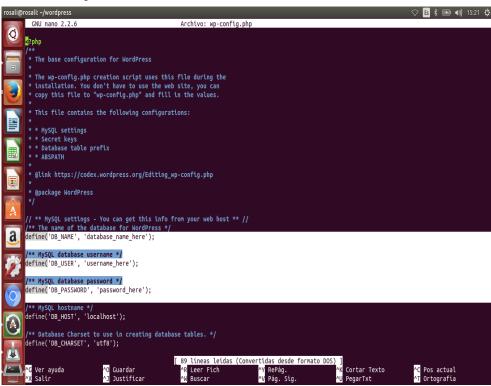
Por último para configurar WordPress primero tenemos que hacer una copia del fichero de configuración de ejemplo que viene en el directorio wordpress donde están los ficheros que descomprimimos anteriormente:

#### cp /wordpress/wp-config-sample.php /wordpress/wp-config.php

Este nuevo fichero lo editaremos para establecer la configuración de nuestra instalación:

#### nano /wordpress/wp-config.php

Cambiaremos los datos de la conexión a la base de datos por los que creamos en el paso anterior:



Guardamos los cambios y salimos del editor.

Movemos los ficheros que teníamos descomprimidos al directorio raíz del servidor web:

#### rsync -avP /wordpress/ /var/www/

Y le damos los permisos para poder subir contenido:

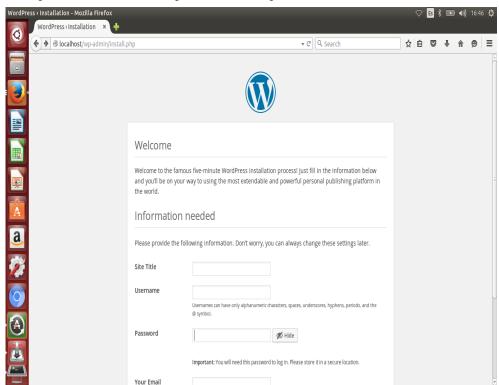
#### sudo chown -R:www-data

A partir de ahora se puede seguir configurando través del navegador web.

Solamente nos falta acceder a la página de instalación en nuestro servidor:

#### http://localhost/wp-admin/install.php

Nos aparecerá un formulario parecido a este que debemos rellenar:



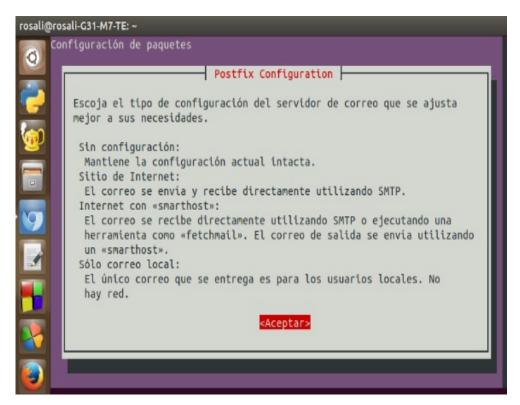
Una vez finalizada la instalación, ya podemos comenzar a crear nuestro sitio web.

#### Squirremail

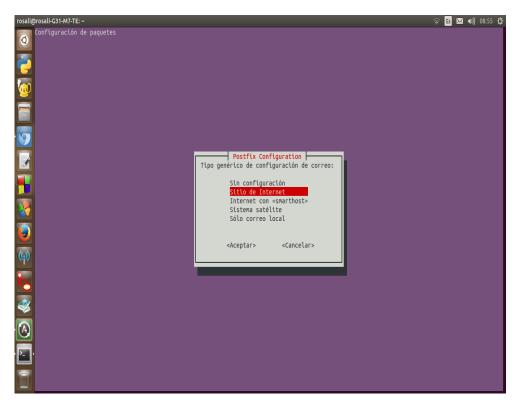
- Instalar Apache2: En nuestro caso ya lo tenemos instalado desde la instalación de WordPress.
- Instalar Postfix: Es una MTA (Mail Tranport Agent) que se encarga del enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado.

#### apt-get install postfix

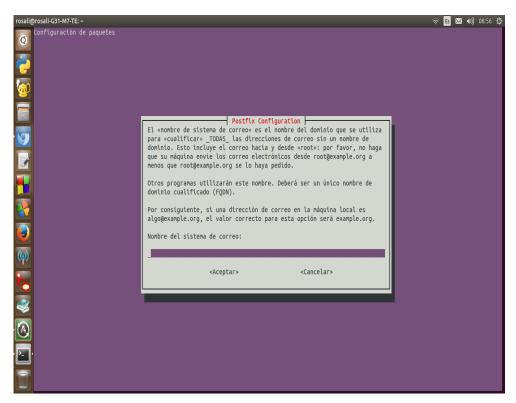
Nos mostrará la siguiente ventana de configuración, pulsaremos intro para continuar.



Durante la instalación se abrirá el agente de configuración. En la primera opción elegimos **Sitio de Internet** 



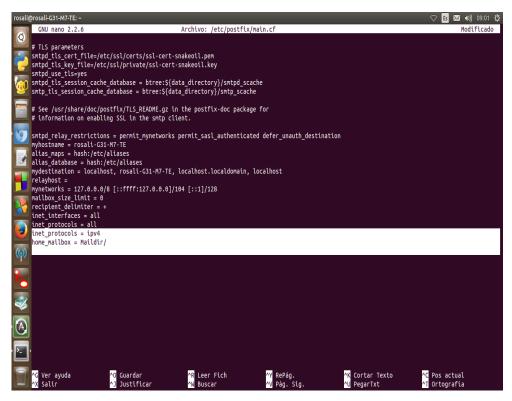
A continuación indicaremos el nombre de dominio que creamos anteriormente en el servidor DNS, en nuestro caso sería thebombgame.com.ar.



Ahora editaremos el fichero de configuración principal que se encuentra en /etc/postfix/main.cf con nano (ó cualquier otro editor):

## nano /etc/postfix/main.cf

Agregaremos al final del fichero una lineas de código donde indicaremos el directorio donde se guardaran los emails.



Para aplicar los cambios realizados reiniciaremos el servicio postfix:

#### /etc/init.d/postfix restart

• Instalar courier-pop y courier-imap: Para los servicios de recepción y lectura de correo se usará Courier, utilizando su servicio IMAP.

Para instalar courier-pop ejecutaremos el siguiente:

#### apt-get install courier-pop

Nos mostrará una ventana de configuración de courier-base, responderemos " ${
m NO}$ " a la pregunta.

Para instalar courier-imap ejecutaremos el siguiente comando:

#### apt-get install courier-imap

• Instalar mailx para enviar y recibir correo electrónico mediante la línea de comandos:

#### apt-get install mailutils

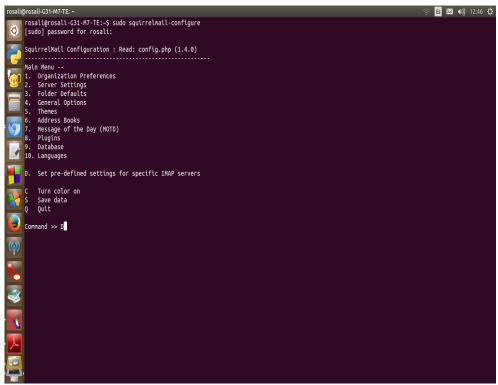
• Instalar SquirrelMail aplicación webmail:

#### apt-get install squirrelmail

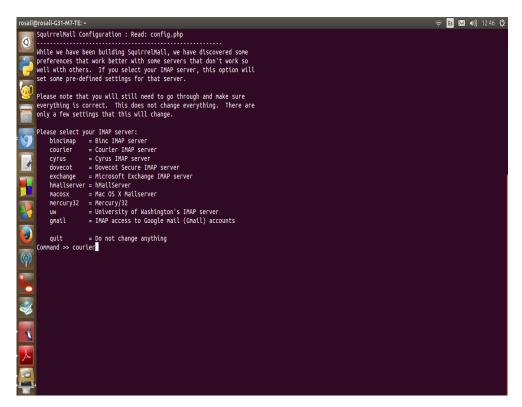
Tras la instalación de SquirrelMail lo configuraremos ejecutando el siguiente comando:

# ${\bf sudo} \ {\bf squirrel mail-configure}$

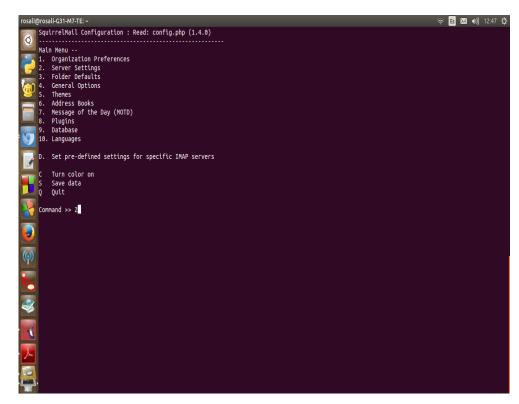
Escribimos a la opción de " ${\bf D}$ " para definir la configuración del servidor IMAP :



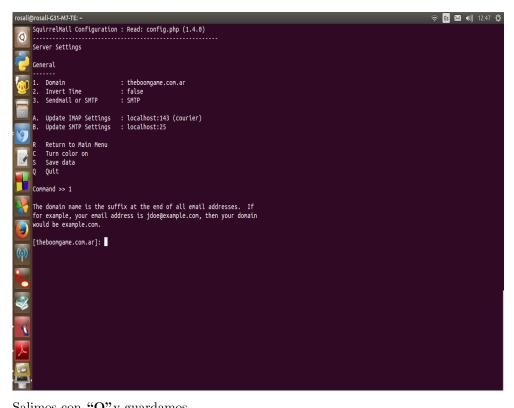
Ingresamos en el servidor IMAP Courier:



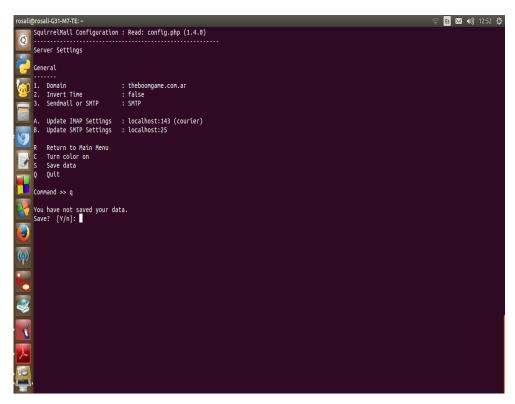
Volveremos al menú principal y elegiremos"2" para entrar en la configuración del servidor:



Entramos en la sección de Domain con "1" Y escribimos el nombre de nuestro dominio(thebombgame.com.ar):



Salimos con " ${f Q}$ "y guardamos.



Ahora para activar SquirrelMail y poder probarlo vía web, accederemos a la carpeta /var/www con el comando:

## cd /var/www

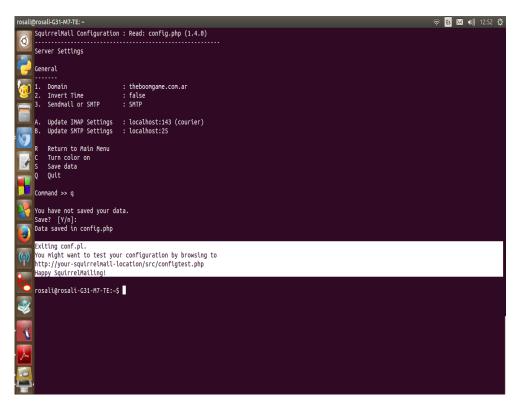
Y ejecutaremos el comando:

# sudo ln -s /usr/share/squirrelmail webmail

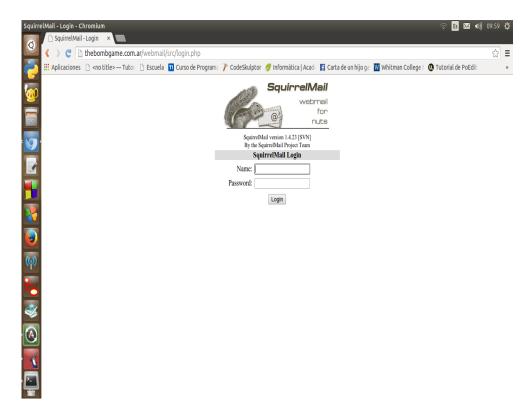
A continuación reiniciaremos Apache con el comando:

# sudo /etc/init.d/apache2 restart

Nos devolverá:



Desde un navegador comprobamos que funciona nuestro servicio de email:

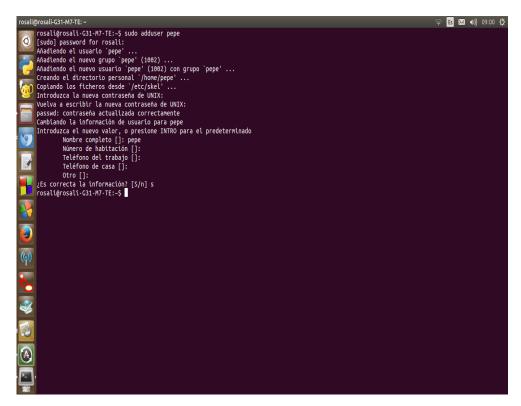


• Creación de usuarios (buzones) para servidor de correo electrónico.

Creamos los usuarios (buzones) que queramos, serán las cuentas de email: Para ello ejecutaremos el siguiente comando en una ventana de terminal, en nuestro caso crearemos un usuario llamado "pepe".

sudo adduser pepe

E Iremos introduciendo los datos que nos vaya pidiendo el asistente de creación de usuario :



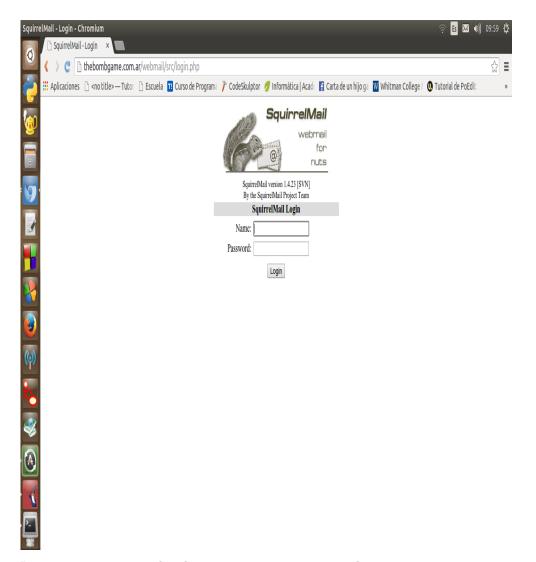
Luego creamos la carpeta "mails" que será la carpeta donde se guardarán los datos del buzón del usuario.

## sudo maildirmake /home/pepe/Maildir

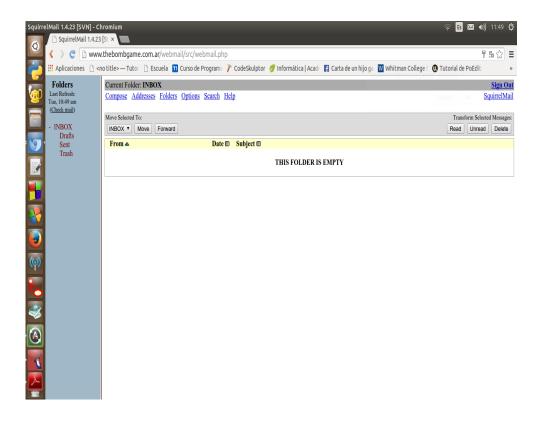
El comando anterior crea la carpeta "mails" y las subcarpetas "cur", "new", "temp". Ahora daremos permisos a la carpeta "mails" y a las subcarpetas que contiene para el usuario y grupo "usuarioNuevo" con el comando:

#### sudo chown pepe:pepe /home/pepe/Maildir -R

 Probar Squirrelmail: Luego de hacer todas las instalaciones y configuraciones de los pasos anteriores ya estamos listo para enviar y recibir mails.
 Nuevamente desde algún navegador escribimos el dominio elegido anteriormente seguido de /webmail.



Ingresamos nuestro nombre de usuario y contraseña y si todo es es correcto podemos acceder a nuestra cuenta en Squirremail:

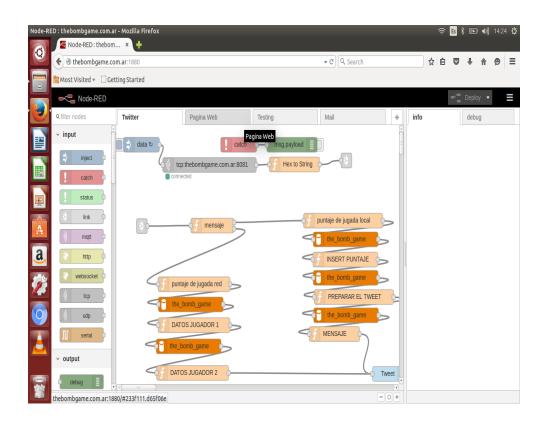


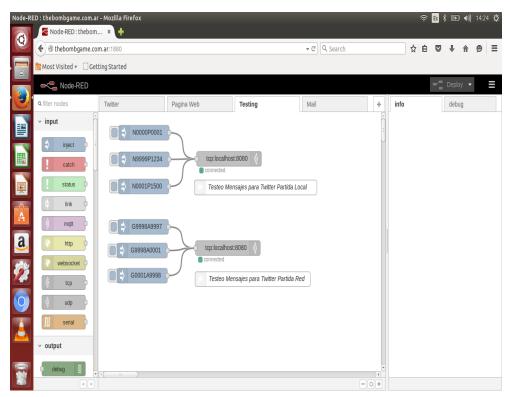
# Uso

Una de las características de nuestro proyecto es que los jugadores pueden registrase en el sitio web del proyecto y visualizar los puntajes obtenidos tanto en la tabla de posiciones, como en la cuenta de Twitter del proyecto. Para lograr esto hicimos uso de la herramienta Node-Red dividiendo en el trabajo en distintas solapas como lo vemos a continuación.

#### Twitter:

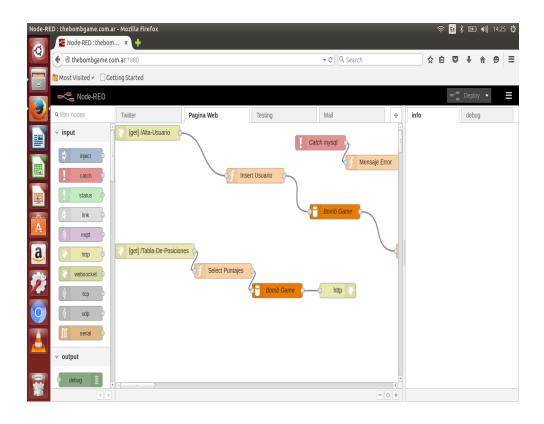
Node-RED utiliza el puerto 8081 para comunicarse con el servidor Node js para hacer las publicaciones en Twitter. En caso de haber algún mensaje para publicar, convierte el mensaje a String, ya que el mensaje recibido esta en hexadecimal. Luego con el mensaje ya convertido a String distinguimos el tipo de mensaje, ya que se pueden realizar partidas locales, como partidas entre dos jugadores. Para finalizar a partir del mensaje recibido, Node-Red busca dentro de la base de datos de los jugadores registrados el o los nombres de los jugadores y su cuenta de Twitter para etiquetarlo en caso de tenerlo y lo manda a Twitter, quien se encarga de publicar. Adicionalmente se creo una solapa para testear distintos tipos de mensajes para probar como se publican en Twitter.

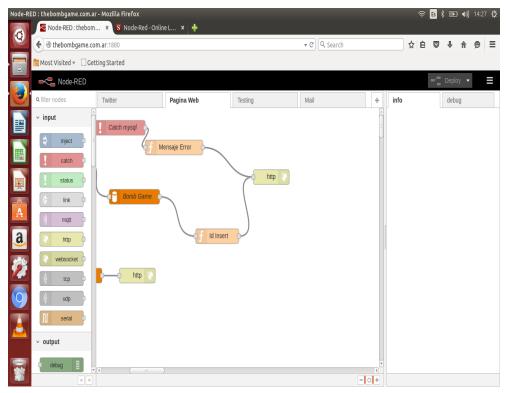




#### WordPress:

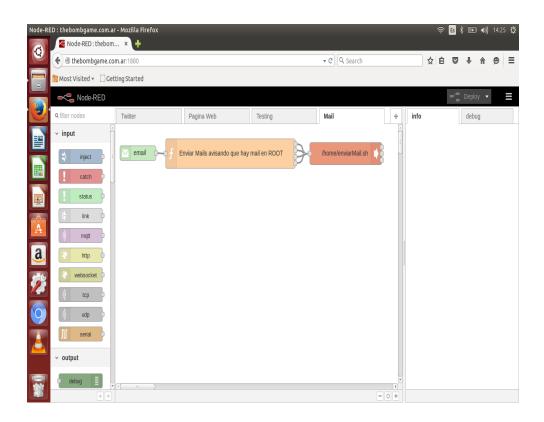
Node-RED responde a dos páginas web. En la primera se carga dar de alta a los usuarios. Cada vez que un usuario quiere registrase en la página web del proyecto, Wordpress le envía a Node-Red una solicitud a través Ajax utilizando JasonP. Node-RED guarda los datos dentro de la base de datos y le envía a Wordpress el id del usuario con el que se inserto. Este id es el que usará el jugador para asociar su usuario con el juego cada vez que quiera iniciar una partida en red. En la segunda página se tiene una tabla de puntajes, que cada vez que Wordpress le pide a Node-RED actualizar su tabla de puntajes, Node-RED consulta la base de datos y le envía a WordPress los datos actualizados los cuales aparecen en la página web.





#### Mails:

Node-Red se encarga de revisar la cuenta de root y en caso de ver un mail les avisa a los administradores del proyecto que tienen un mail.



# Problemas que surgieron:

Configuración del DNS: Intentamos seguir varios tutoriales para la configuración del dns pero no conseguíamos asociar la página web con la ip del servidor, hasta que probamos seguir este tutorial http://www.webebre.net/instalacion-y-configuracion-de-un-servidor-dns/y logramos vincular el dominio de la página web con la ip del servidor.

Envío de emails con Node-RED: No se podían enviar mails desde Node-Red porque necesita de certificados ssl, un cerificado ssl es un protocolo que garantiza la seguridad de los datos que se mandan a través de Internet, por lo tanto para poder solucionarlo utilizamos un nodo exec, que ejecuta comandos por consola y creamos un script que recibe por parámetro el origen, el destino y el mensaje.

Permisos en SquirreMail: Tuvimos problemas de permisos con la carpeta Maildir, lo solucionamos modificando los permisos de acceso a los usuarios.

WordPress con Node-RED: Ajax no permite la conexión a otros puertos del equipo, lo solucionamos usando conexiones Ajax JasonP.

# Conclusión:

Logramos mostrar el funcionamiento de todos los servicios y que los mismos interactúen entre ellos a partir de Node-Red, que sin dudas ha demostrado que es una herramienta de gran potencial.

# Referencias:

#### Node-RED:

- http://nodered.org/docs/getting-started/installation.html
- http://nodered.org/docs/getting-started/running
- https://www.npmjs.com/package/node-mysql

#### Wordpress:

- http://www.vicente-navarro.com/blog/2009/02/22/crear-los-certificados-ssl-para-nuestro
- http://cookyourweb.com/como-instalar-wordpress-en-ubuntu-14-04/
- https://openwebinars.net/como-instalar-wordpress-en-ubuntu-1404/
- https://openwebinars.net/como-instalar-linux-apache-mysql-y-php-lamp-en-ubuntu-1404/

#### SquirreMail:

- https://enredesao.wordpress.com/tutoriales-servicios-de-red/servidor-de-correo/servidor-de-correo-en-ubuntu/
- http://www.interorganic.com.ar/josx/SquirrelMail.pdf

## DNS:

• http://www.webebre.net/instalacion-y-configuracion-de-un-servidor-dns/