

# Instrucciones para implantar la plataforma de Prácticas

*Versión 1.0 2017/09/15*

## Paquetes necesarios

El software instalado en los laboratorios es Ubuntu 14.04, con los siguientes paquetes relevantes:

- postgresql
- apache2
- apache2-mpm-prefork
- php5
- php5-pgsql
- php5-xdebug
- tora
- phppgadmin
- pgadmin3
- google-chrome-stable
- firefox
- xsltproc
- libxml2-utils

Los pasos necesarios para instalarlos:

- Como tu usuario de escritorio, ejecuta:

```
$ sudo apt-get install postgresql apache2 apache2-mpm-prefork \  
php5 php5-pgsql php5-xdebug tora phppgadmin libqt4-sql-psql \  
google-chrome-stable pgadmin3 firefox xsltproc libxml2-utils
```

(si el paquete google-chrome da problemas, se puede descargar de la web de google)

Se recomienda el uso de las herramientas de desarrollador Web que tanto Firebug como Chrome tienen incluidas

## Configurar PostgreSQL

- Para que podamos utilizar el usuario "alumnodb", es necesario editar el fichero /etc/postgresql/9.3/main/pg\_hba.conf para que contenga las siguientes líneas adicionales. **Es importante que estas líneas estén situadas delante de otras directivas de autenticación (por ejemplo: al principio del fichero).**

```
local    all        alumnodb          md5
host     all        alumnodb      127.0.0.1/32    md5
```

También se puede usar 'trust' en lugar de 'md5' como método alternativo de autenticación (no se pide contraseña).

- Asimismo, conviene editar el fichero /etc/postgresql/9.3/main/postgresql.conf, buscar las líneas con los valores comentados de autovacuum\_vacuum\_threshold, autovacuum\_analyze\_threshold y establecer los valores:

```
autovacuum_vacuum_threshold = 5000000
autovacuum_analyze_threshold= 5000000
```

- Tras esto, es necesario reiniciar el servicio ejecutando:

```
$ sudo service postgresql restart
```

- Ahora debemos crear el "rol" alumnodb en postgres. Para ello, iniciamos sesión como usuario "postgres" y creamos el rol con los siguientes comandos:

```
$ sudo su - postgres
```

- y después:

```
$ createuser -s alumnodb
```

- Para dar de alta el lenguaje plpgsql en todas las nuevas bases de datos, como usuario postgres, ejecutar (normalmente no es necesario porque ya está dado de alta):

```
$ createlang plpgsql template1
```

- Para acceder a la base de datos, y asignar una contraseña (necesaria para acceder desde phpPgadmin; introducir 'alumnodb' como contraseña)

```
$ psql
\password alumnodb
...
\q
```

- Por último, para comprobar que todo está correcto, creamos una base de datos. Como nuestro usuario de inicio de sesión, ejecutamos:

```
$ createdb -U alumnodb bdat
```

- 

También conviene instalar el programa schemaSpy, <http://schemaspy.sourceforge.net>, (y schemaSpyGUI, <http://sourceforge.net/projects/schemaspygui>) para obtener diagramas de las bases de datos desarrolladas.

## Configurar Apache

Ahora vamos a configurar el servidor web Apache para publicar la carpeta personal public\_html en la ruta <http://localhost/~usuario/>. Para ello:

- Crear el directorio "public\_html" dentro de \$HOME (carpeta personal: /home/usuario)

```
$ mkdir ~/public_html
```

- Ejecutar este mandato:

```
$ sudo a2enmod userdir
```

- Tras esto, es necesario reiniciar el servicio ejecutando:

```
$ sudo service apache2 restart
```

- Ya se puede acceder al contenido de ésta en <http://localhost/~usuario/>, donde "usuario" es el que ha iniciado la sesión.

## Configurar PHP

Para poder ejecutar páginas php en el directorio de usuario es necesario editar el fichero

```
/etc/apache2/mods-enabled/php5.conf
```

y comentar las siguientes líneas

```
# <IfModule mod_userdir.c>
#     <Directory /home/*/public_html>
```

```
#           php_admin_value engine Off
#         </Directory>
#       </IfModule>
```

Para poder hacer uso del depurador de PHP en Netbeans es necesario añadir las siguientes líneas:

```
; PHP debugger en Netbeans
xdebug.remote_enable=on
xdebug.remote_handler=dbgp
xdebug.remote_host=localhost
xdebug.remote_port=9000
```

al final del archivo:

```
/etc/php5/apache2/php.ini
```

Después hay que rearrancar Apache:

```
$ sudo service apache2 restart
```

Para probar el correcto funcionamiento de PHP, crear el fichero [phpinfo.php](#) en el directorio public\_html del usuario con el siguiente contenido:

```
<?php
    phpinfo();
?>
```

y acceder a la página: <http://localhost/~usuario/phpinfo.php>

## Añadir plugin de PHP a NetBeans

Abrir NetBeans->Tools->plugins y seleccionar en la lista de 'AvailablePlugins' PHP.

Para hacer uso del debugger de PHP, activar PHP con Netbeans->Herramientas->Complementos->Instalados y marcar PHP.

## Trucos de PHP

Para revisar errores:

- Revisar

```
$ cat /var/log/apache2/error.log
```
- Cambiar las opciones en /etc/php5/apache2/php.ini:

```
display_errors = On
```

```
error_reporting = E_ALL & ~E_NOTICE
```

- Ejecutar intérprete PHP en la línea de mandatos:  

```
$ php5 mipagina.php
```
- Usar Netbeans (indica errores de sintaxis) y su depurador de PHP

## Trucos de PostgreSQL

Se recomienda el uso de las herramientas de gestión:

- `$ tora`
- `$ pgadmin3`
- `http://localhost/phppgadmin/`

Para entrar en el entorno interactivo de psql y obtener ayuda:

```
$ psql -U alumnodb bdat
```

```
bdat=#help
```

```
bdat=#\help
```

```
bdat=#\?
```

```
bdat=#\q
```

Para evitar tener que introducir usuario y contraseña:

```
$ export PGUSER=alumnodb
```

```
$ export PGPASSWORD=alumnodb
```

Para ejecutar un script:

```
$ cat miscript.sql | psql -U alumnodb bdat
```

Para crear una base de datos:

```
$ createdb -U alumnodb dbname
```

Para volcar el contenido (y estructura) de una base de datos:

```
$ pg_dump -U alumnodb bdat > outputfile.sql
```

El fichero resultante es un script que se puede usar para recrear la base de datos, ejecutándolo tal y como se indica más arriba.

Para realizar una carga masiva de datos se puede usar el mandato COPY. Ver el contenido del script de salida del mandato pg\_dump.

```
create table profesor (id int, nombre char(20), apellido char(20));  
COPY profesor from stdin using delimiters '|' ;  
1|Pedro|Pascual  
2|Julia|Díaz  
\.
```

Para ver los logs de postgresql:

```
$ sudo su - postgres  
$ ls -al /var/log/postgresql/  
$ cat /var/log/postgresql/postgresql*.log
```

## Enlaces de referencia

- <http://www.w3schools.com>
- <http://www.php.net>
- <http://www.postgresql.org>