**BACK END DRUPAL**

Modificar el archivo /etc/apt/sources.list y agregamos la línea para poder instalar una versión previa de PostgreSQL:

deb http://ftp.debian.org/debian/ wheezy-backports main

Actualizar

#aptitude update

Instalar los paquetes necesarios.

#apt-get –t wheezy-backports install postgresql-9.0

#apt-get install apache2 drush php5-pgsql php5-gd drupal7

Php5-pgsql es el paquete para el soporte de php5 para postgres y php5-gd es el soporte de php5 para manejar imágenes. Drush es una herramienta para facilitar el manejo de Drupal por línea de comando.

* **Configuración de base de datos postgres**

Cambiamos al usuario postgres:

su postgres

Creamos un script para crear la base de datos de nuestro sitio:

DROP DATABASE IF EXISTS drupalDB;

DROP USER IF EXISTS drupal;

CREATE USER drupal WITH ENCRYPTED PASSWORD 'hola123,';

CREATE DATABASE drupalDB WITH OWNDER drupal;

GRANT ALL ON DATABASE drupalDB TO drupal;

\q

Después ejecutamos:

$psql < backendDB

* **Descargar Drupal**

Como root nos cambiamos al directorio /var/www. Utilizando drush, descargamos drupal:

#cd /var/www

#drush dl drupal-7.0

#mv drupal-7.0 drupalsite

Agregamos el sitio de drupal en el directorio de apache como un virtualhost:

#nano /etc/apache2/sites-available/drupalsite

Y le añadimos lo siguiente:

<VirtualHost \*:80>

ServerName drupalsite

ServerAdmin mail@drupalsite.com

DocumentRoot /var/www/drupalsite

<Directory />

Options FollowSymLinks

AllowOverride All

</Directory>

<Directory /srv/www/drupalsite>

Options Indexes FollowSymLinks Multiviews

Order allow,deny

Allow from all

</Directory>

</VirtualHost>

Creamos los archivos necesarios para la instalación y cambiamos sus permisos:

#cd /var/www/drupalsite/sites/default

#cp default.settings.php settings.php

#mkdir files

#chown www-data:www-data /var/www/drupalsite/sites/default/settings.php

#chown -R www-data:www-data /var/www/drupalsite/sites/default/files

Habilitamos el sitio en apache:

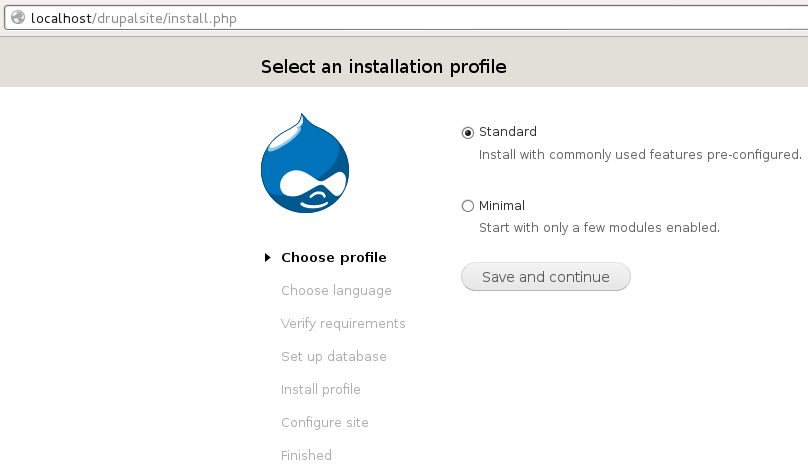
a2ensite drupalsite

Reiniciamos el servidor web:

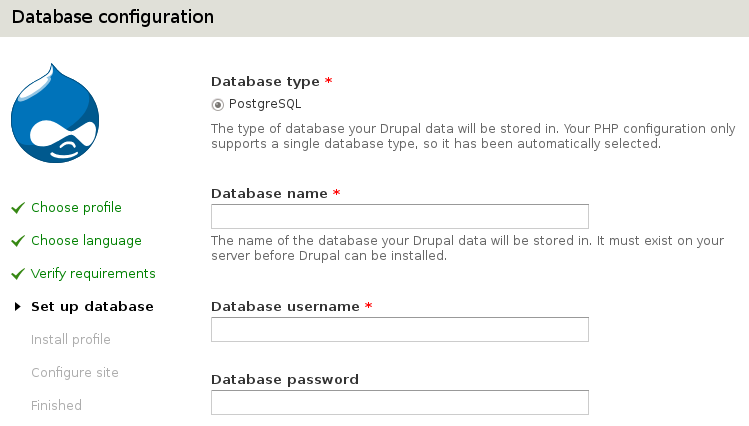
/etc/init.d/apache2 restart

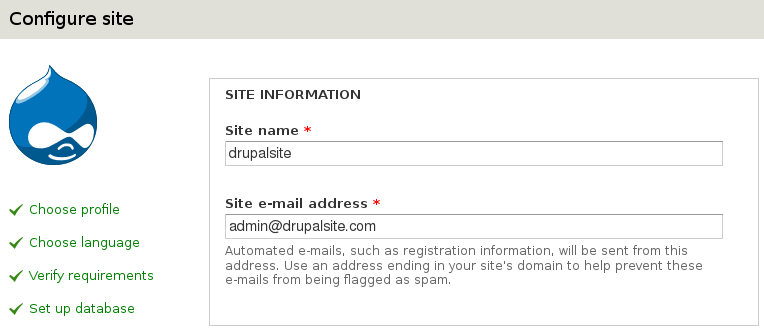
* **Instalar Drupal**

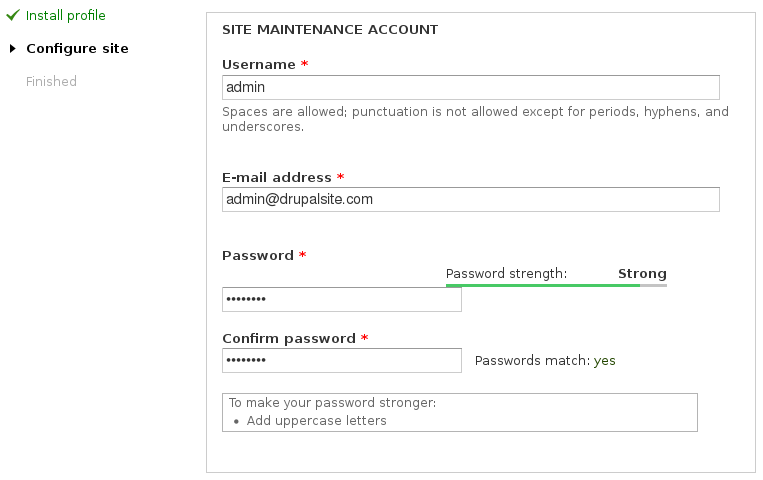
Abrimos en el navegador http://localhost/drupalsite/install.php y seguimos los pasos.



PostgreSQL es detectado por defecto







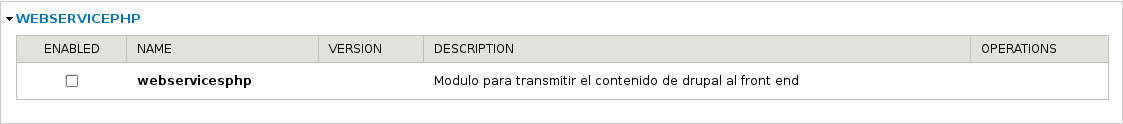
En la base de datos de drupal las tablas reacionadas con el contenido del blog son:

* Node: tipo de contenido (artículo), título,
* File\_managed: información sobre las imágenes. Nombre de la imagen, uri, tamaño.
* Field\_data\_body: contenido del artículo?
* Field\_revision\_body, node\_revision: encabezado del artículo?
* Search\_index, search\_total, taxonomy\_term\_data = información de los tags
* Node\_revision: título del artículo

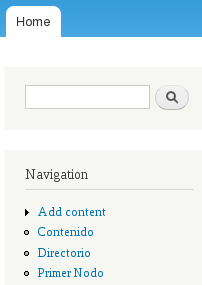
Creamos el directorio de nuestro módulo en /var/www/drupalsite/sites/all/modules y dentro, los siguientes archivos:

* webservicephp.info: contiene la información del módulo.
* webservicephp.module: Es el módulo en PHP.

Para comprobar que el módulo se ha creado correctamente, abrimos en el navegador la URL http://localhost/drupalsite y en el menú superior seleccionamos la opción “Modules”. El módulo y la información que acabamos de agregar deben ser visibles en la parte inferior de la página.



Habilitamos el módulo y, en la sección “Navigation” de la página principal de Drupal ya podremos observar los links a las opciones que habilitemos con el módulo.



**¿Cómo obtener datos de Drupal desde un servidor remoto?**

En el servidor remoto instalamos apache2 y php5. A continuación creamos un archivo llamado directorio.php en /var/www/ con el siguiente contenido:

<?php

//URL de la página habilitada por el módulo

$url = "http://ip\_servidor\_drupal/drupalsite/?q=webservicesphp/directorio";

//Opciones de la petición HTTP, en este caso le indicamos que use el método POST

//Y que el tipo de contenido es una aplicación datos de tipo nombre/valor

$options = array(

'method' => 'POST',

'header' => "Content-Type: application/x-www-form-urlencoded\r\n",

);

//Crea el contexto de flujo de datos con las opciones especificadas en $options

$context = stream\_context\_create();

//Se transmite el archivo o contenido de la $url a la cadena $result

$result = file\_get\_contents($url, false, $context);

//Se imprime el contenido de $result

echo $result;

?>

Y al abrir en el navegador <http://localhost/directorio.php> se mostrarán los datos obtenidos del servidor de Drupal.

**Referencias**

<http://digital-madness.in/blog/2013/github-for-beginners-debianubuntulinux/>

<http://askubuntu.com/questions/137165/git-ssh-permission-denied-publickey>

<http://guides.railsgirls.com/github/>

<http://mmc.geofisica.unam.mx/acl/linux/paquetes.txt>

<http://backports.debian.org/Instructions/>

<http://tubasededatoslibre.org/site/instalacion-de-postgresql-en-debian-gnulinux-squeeze/>

<http://www.redmamba.es/hola-mundo-programar-drupal-7-2011>

<http://hackurbano.net/2012/06/28/drupal-7-en-debian-squeeze-usando-postgres/>

<https://drupal.org/files/menu-link-save-1338282-30-d7.patch>

<https://drupal.org/node/1414412>

<http://adellefrank.com/blog/drupal-database-tables-files-backup-migrate>

<https://drupal.org/node/1848348> //db\_select

<https://drupal.org/node/43512> //formato de fecha

<https://drupal.org/node/1472886> //joins múltiples

<http://stackoverflow.com/questions/4286677/show-image-using-file-get-contents> //mostrar imagen obtenida a partir de su ruta

<http://stackoverflow.com/questions/5647461/how-do-i-send-a-post-request-with-php> //petición http desde el cliente