

# Manual de Instalación del Sistema TDS para Ambientes de Desarrollo Linux

Luis Crespo, María José Ramos, Daniel Azocar

23 de febrero de 2014

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Instalación de Dependencias</b>	<b>4</b>
2.1. Instalación de Git . . . . .	4
2.2. Instalación de Python . . . . .	4
2.3. Instalación de Pip . . . . .	5
2.4. Instalación de Django . . . . .	5
2.5. Instalación de PostgreSQL . . . . .	6
2.6. Instalación de Extensiones para Django . . . . .	6
<b>3. Configuración de la Base de Datos</b>	<b>7</b>
<b>4. Instalación de la Aplicación</b>	<b>8</b>

# 1. Introducción

Este manual contempla la explicación de la instalación y configuración de cada uno de los componentes necesarios para la ejecución del sistema TDS en un ambiente de desarrollo Linux.

Muchos de los paquetes de instalación se encuentran en los repositorios correspondientes a cada distribución de Linux. En el presente documento se utilizará el manejador de repositorios *yum* para ambientes basados en Red Hat Linux, y el manejador de repositorios *apt-get* para ambientes basados en Debian Linux.

Es importante destacar que la mayoría de los procedimientos para la instalación de los componentes necesarios, requieren de privilegios de super usuario (administrador), por lo tanto, se deben ejecutar las instrucciones correspondientes como usuario *root*; o simplemente precediendo cada instrucción con el modificador *sudo*.

## 2. Instalación de Dependencias

### 2.1. Instalación de Git

El código fuente de la aplicación se encuentra almacenado en un repositorio remoto en GitHub, para interactuar con dicho repositorio y manejar el control de versiones se utilizará *Git*.

Para instalar *Git*, abra una línea de comandos y ejecute la siguiente instrucción:

Ambientes Debian:

```
$ apt-get install git
```

Ambientes Red Hat:

```
$ yum install git
```

### 2.2. Instalación de Python

Es común que los sistemas de operación Linux tengan Python instalado por defecto. Para chequear si ya posee Python, y obtener la versión instalada, abra una línea de comandos y ejecute:

```
$ python --version
```

Si no se encuentra el comando, o la versión es menor a *2.7.x*, ejecute las siguientes instrucciones:

Ambientes Debian:

```
$ apt-get update  
$ apt-get install python
```

Ambientes Red Hat:

```
$ yum update  
$ yum install python
```

## 2.3. Instalación de Pip

Para instalar ciertas dependencias de Python es necesario el manejador de paquetes *pip*. Para instalarlo abra una línea de comandos y ejecute las siguientes instrucciones:

Ambientes Debian:

```
$ apt-get install python-pip
```

Ambientes Red Hat:

```
$ yum install python-pip
```

## 2.4. Instalación de Django

Para la instalación del framework Django, se debe utilizar el manejador de paquetes de Python instalado anteriormente. A continuación, abra una línea de comandos y ejecute la siguiente instrucción:

```
$ pip install django
```

Para verificar que la instalación ha sido exitosa, ejecute:

```
$ python
```

Se ejecutará el interpretador de Python, allí ejecute:

```
>>> import django
>>> django.VERSION
```

Debería aparecer en pantalla un resultado similar a:

```
(1, 6, 2, 'final', 0)
```

Ésto indica la versión de Django instalada actualmente, verificando así la correcta instalación del framework.

## 2.5. Instalación de PostgreSQL

El manejador de base de datos utilizado por la aplicación es PostgreSQL. Para instalarlo, abra una línea de comandos y ejecute la siguiente instrucción:

Ambientes Debian:

```
$ apt-get install postgresql
```

Ambientes Red Hat:

```
$ yum install postgresql-server
```

Luego, para hacer más sencillo el manejo y la configuración de bases de datos con PostgreSQL, se puede instalar pgAdmin3, para ésto simplemente ejecute la siguiente instrucción:

Ambientes Debian:

```
$ apt-get install pgadmin3
```

Ambientes Red Hat:

```
$ yum install pgadmin3
```

## 2.6. Instalación de Extensiones para Django

Para utilizar la base de datos a partir del framework, es necesario un paquete de Python, llamado *psycopg2*, que funciona como interfaz entre el lenguaje y el manejador de bases de datos de PostgreSQL. Para instalarlo simplemente abra una línea de comandos y ejecute:

```
$ pip install psycopg2
```

También serán necesarias otras extensiones de Django, para instalarlas ejecute:

```
$ pip install django_extensions  
$ pip install pillow  
$ pip install pisa  
$ pip install reportlab  
$ pip install html5lib
```

### 3. Configuración de la Base de Datos

Haciendo uso del programa instalado anteriormente, pgAdmin3, a continuación se realizará la configuración de la base de datos que será utilizada por la aplicación.

Abra pgAdmin3. Es posible que la conexión al servidor de PostgreSQL no esté establecida. Para hacerlo, diríjase a *File > Add Server*. Ésto abrirá una ventana en donde debe llenar los siguientes datos:

Name: <nombre que desee>  
Host: *localhost*  
Port: *5432*  
Service: <dejarlo vacío>  
Maintenance DB: *postgres*  
Username: *postgres*  
Password: <dejarlo vacío>  
Store password: <desmarcarlo>  
Group: *Servers*

Presione *Ok*. Podrá observar que se ha añadido el servidor en la lista de servidores ubicada en el panel izquierdo de la ventana. Al expandir el servidor añadido, que podrá ubicar con el nombre que le colocó, haga click derecho en la lista llamada *Databases* y luego haga click en *New database*. En la ventana que se abre coloque los siguientes datos:

Name: *cmsb*  
Owner: *postgres*

## 4. Instalación de la Aplicación

Para obtener el código fuente de la aplicación, el mismo debe ser descargado desde el repositorio en GitHub al cual debe haber sido invitado como colaborador. Si no ha sido invitado como colaborador, debe solicitar dicha autorización al equipo de desarrollo de la aplicación.

Si ya tiene acceso a dicho repositorio, puede clonarlo en su computadora abriendo una línea de comandos y ejecutando:

```
$ git clone git@github.com:figueira/TDScopia.git
```

De esta manera, se creará una carpeta llamada *TDScopia*, ubicada en el directorio en el que se encuentra actualmente, que contiene el código fuente de la aplicación. A partir de ahora, nos referiremos a la ruta de dicha carpeta como *home\_tds*.

Ahora es necesario realizar ciertas configuraciones para el correcto funcionamiento de la aplicación. Para ésto, debe crearse el archivo *settings.py* dentro de la carpeta *home\_tds/AM/*. El contenido de este archivo debe ser copiado del archivo *home\_tds/recursos/configuracion/settings.py.template* y luego modificar los siguientes datos:

En la opción llamada *DATABASES*, asigne a la entrada *ENGINE* el valor:

```
‘django.db.backends.postgresql_psycopg2’
```

Luego, asigne a la entrada *NAME* el valor:

```
‘cmsb’
```

A la entrada *USER* asígnele el valor:

```
‘postgres’
```

Por último, asigne a la entrada *PASSWORD* la contraseña del usuario *postgres*. Si no la ha cambiado, el valor de este campo debe ser vacío (‘’).

A continuación, para realizar la sincronización de la base de datos, abra una línea de comandos, ubíquese en *home\_tds* y ejecute:

```
$ python manage.py syncdb
```



Acepte la creación de un super usuario para la aplicación e ingrese un nombre de usuario, email y contraseña.

Una vez sincronizada la base de datos, se podrá ejecutar la aplicación ejecutando la siguiente instrucción:

```
$ python manage.py runserver
```

Luego, podrá acceder a la aplicación desde cualquier navegador a través del url:

```
localhost:8000
```