

Base de datos Django
José Santorcuato Tapia
Chile 2017

Para utilizar base de datos prefiero PostgreSQL, la mayoría de los sistemas Cloud Computing vienen optimizados para trabajar con ella, sé que MySQL es favorita y que en ambiente de negocios prefieren MSSQL.

Instalación PostgreSQL

En mi repositorio comandos varios y de Raspberry encontrarás cómo instalar PostgreSQL en Linux y OSX, también está para Windows en el repositorio de Ruby Sura.

<https://github.com/joseSantorcuato>

En consola psql crearé la base de datos

Ingreso a PostgreSQL

```
psql postgres
```

```
postgres=# CREATE DATABASE usuarios WITH OWNER pguser;
```

Listamos Las base de datos

```
\l
```

Name	Owner	Encoding	Collate	Ctype	Access privileges
postgres	usuario	UTF8	C	UTF-8	
template0	usuario	UTF8	C	UTF-8	=c/usuario + usuario=CTc/usuario
template1	usuario	UTF8	C	UTF-8	=c/usuario + usuario=CTc/usuario
usuarios	pguser	UTF8	C	UTF-8	

Django

En el editor de código abre Accesible/settings.py

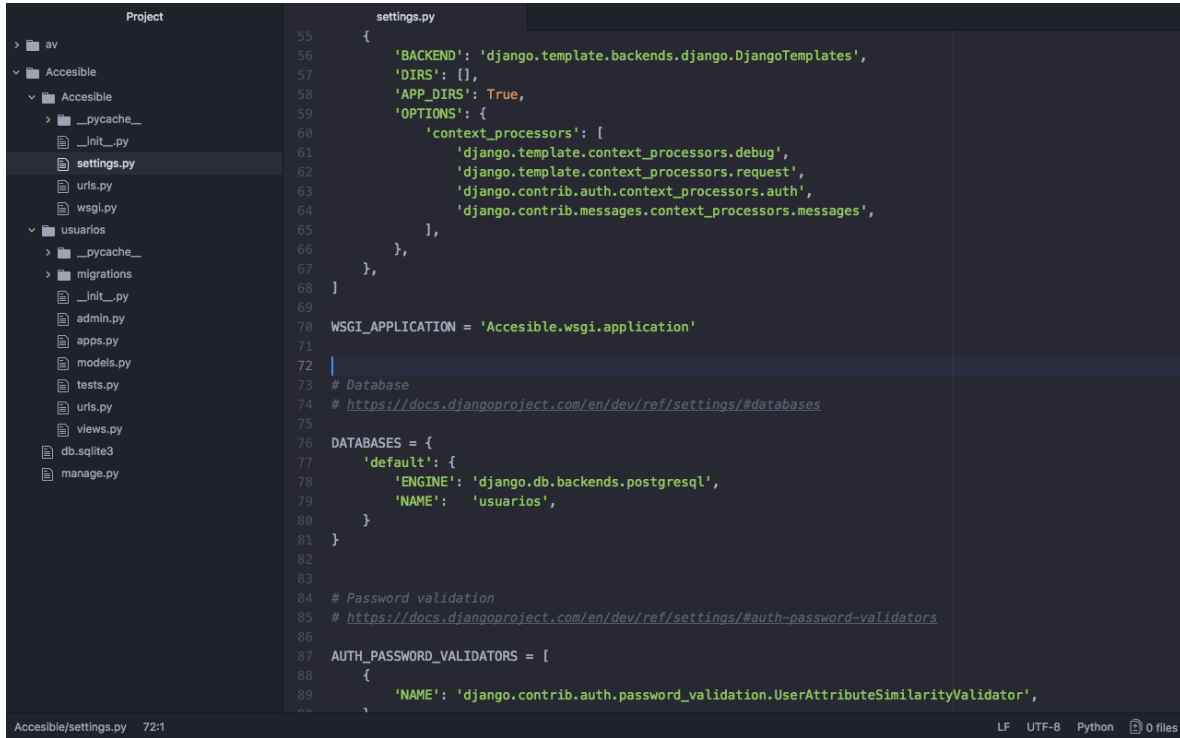
Django creará tablas básicas para funcionar, están en el mismo settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

Agregaremos nuestro motor de base de datos y nuestra base de datos

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'usuarios',  
    }  
}
```

Se verá así



```
55 {
56     'BACKEND': 'django.template.backends.django.DjangoTemplates',
57     'DIRS': [],
58     'APP_DIRS': True,
59     'OPTIONS': {
60         'context_processors': [
61             'django.template.context_processors.debug',
62             'django.template.context_processors.request',
63             'django.contrib.auth.context_processors.auth',
64             'django.contrib.messages.context_processors.messages',
65         ],
66     },
67 },
68 ]
69
70 WSGI_APPLICATION = 'Accesible.wsgi.application'
71
72 # Database
73 # https://docs.djangoproject.com/en/dev/ref/settings/#databases
74
75 DATABASES = {
76     'default': {
77         'ENGINE': 'django.db.backends.postgresql',
78         'NAME': 'usuarios',
79     }
80 }
81
82 # Password validation
83 # https://docs.djangoproject.com/en/dev/ref/settings/#auth-password-validators
84
85 AUTH_PASSWORD_VALIDATORS = [
86     {
87         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
88     },
89 ]
```

Ahora ejecutamos el siguiente comando en terminal(adentro del proyecto que creamos en el pdf anterior 1djangoInstalacion.pdf).

(amb36) MacBook-Pro-de-usuario:Accesible usuario

python manage.py migrate

Arrojará

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

Running migrations:

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying sessions.0001_initial... OK
```

(amb36) MacBook-Pro-de-usuario:Accesible usuario

Se habrán creado las tablas, ahora crearemos la de nuestro proyecto en usuarios/models

```
from django.db import models
```

```
class Usuario(models.Model):
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    ocupacion = models.CharField(max_length=100)

class Discapacidad(models.Model):
    beneficiario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    discap = models.CharField(max_length=100)
    anoingreso = models.DateField()
    evaluacion = models.IntegerField()
```

Adentro de usuarios/migrations/apps tendremos esto

```
from django.apps import AppConfig
```

```
class UsuariosConfig(AppConfig):
    name = 'usuarios'
```

Ahora agregamos lo siguiente a settings.py

```
INSTALLED_APPS = [
    'usuarios.apps.UsuariosConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Ejecutamos nuestro modelo

```
python manage.py makemigrations usuarios
```

Migrations for 'usuarios':

```
usuarios/migrations/0001_initial.py
- Create model Discapacidad
- Create model Usuario
- Add field beneficiario to discapacidad
```

Se ha creado nuestro modelo y tablas

Ahora ejecutamos

```
python manage.py migrate
```

Operations to perform:

```
Apply all migrations: admin, auth, contenttypes, sessions, usuarios
```

Running migrations:

```
Applying usuarios.0001_initial... OK
```

Lo siguiente es crear al administrador, Django crea el sistema automáticamente, lo básico.

```
(amb36) MacBook-Pro-de-usuario:Accesible usuario$
```

```
python manage.py createsuperuser
```

```
Username (leave blank to use 'usuario'): djangoadmin
Email address: pruebas@mail.com
Password: tucontraseña
Password (again):
Superuser created successfully.
```

Corremos nuestro servidor

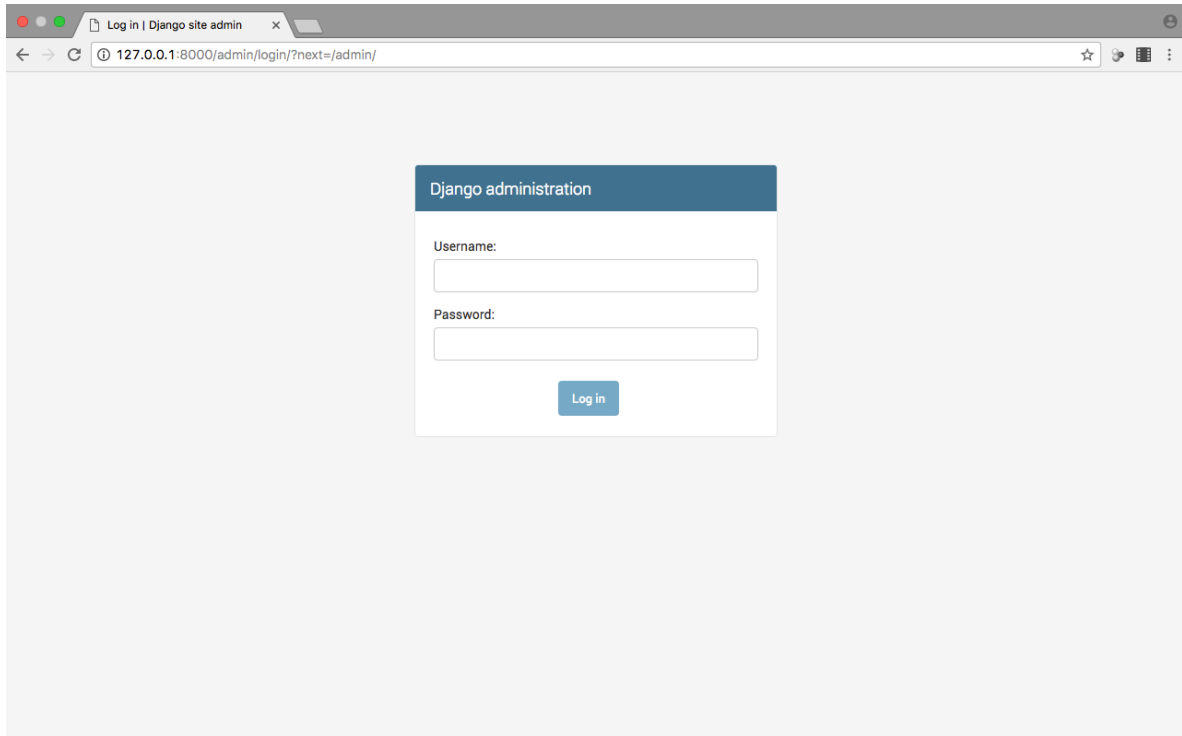
```
(amb36) MacBook-Pro-de-usuario:Accesible usuario$ python manage.py runserver
Performing system checks...
```

```
System check identified no issues (0 silenced).
October 28, 2017 - 01:13:57
Django version 2.0b1, using settings 'Accesible.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Ingresamos a

<http://127.0.0.1:8000/admin>

Se verá así



The screenshot shows a web browser window with the title "Log in | Django site admin". The address bar displays the URL "127.0.0.1:8000/admin/login/?next=/admin/". The main content area features a "Django administration" login form. This form includes a header, a "Username:" label with a text input field, a "Password:" label with a text input field, and a "Log in" button.

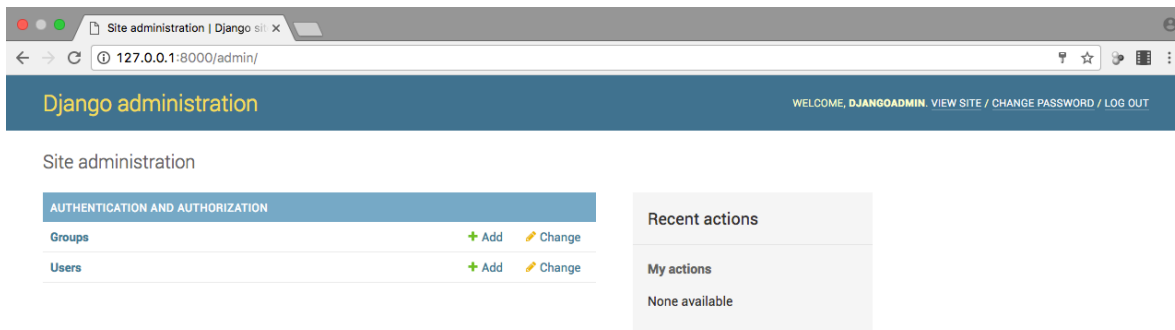
Django administration

Username:

Password:

Log in

Ingresamos nuestras credenciales y aparecerá

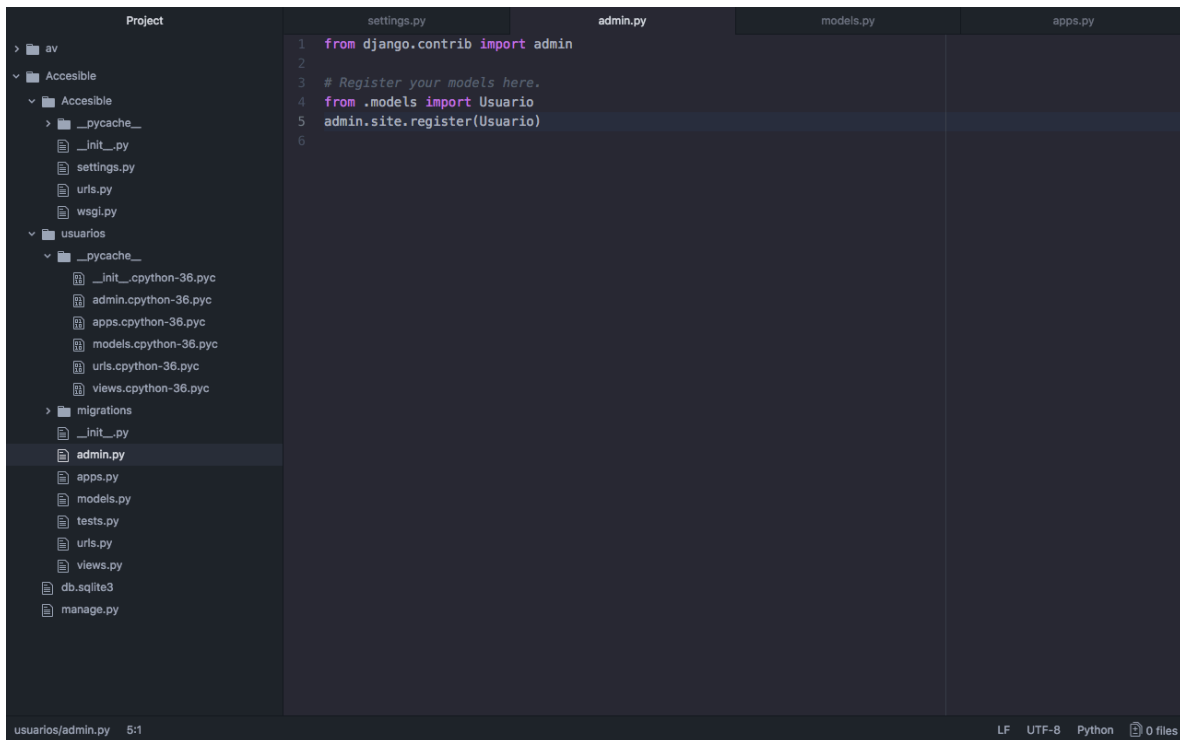


¿Pero y nuestra aplicación?

En el archivo migrations/admin.py agregamos

```
from .models import Usuario
```

```
admin.site.register(Usuario)
```

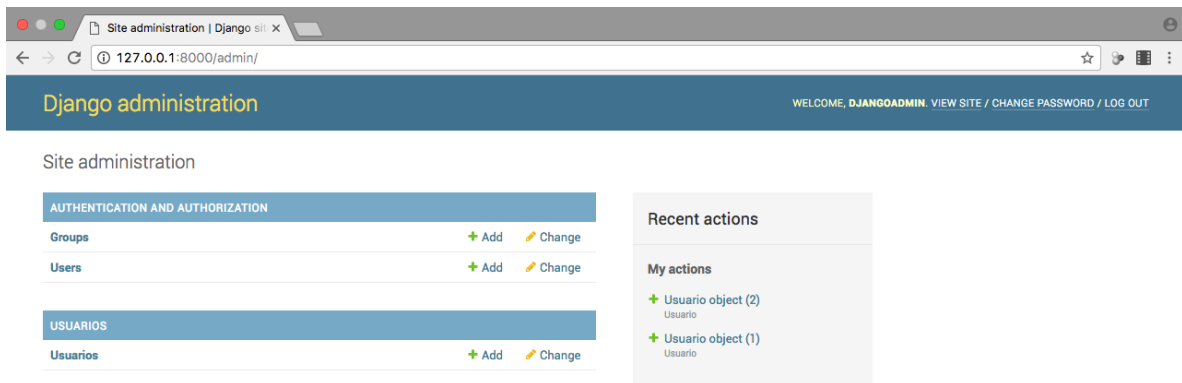


Ejecutamos python manage.py runserver

Ejecutamos

<http://127.0.0.1:8000/admin>

Ahora visualizamos nuestra app



En el próximo pdf agregaremos más tablas y más relaciones