

Manual instalación DJANGO

¿Por qué DJANGO?

DJANGO es un framework de desarrollo que utiliza el lenguaje Python.

¿Por qué?

Python es un lenguaje que tiene una amplia colección de módulos para HCI, Human-Computer Interaction.

¿Qué es un framework?

Un marco, que define lenguaje, componentes y características de un producto, sea digital o no. Podría ser un framework de negocios, procesos y/o servicios. Si lo quieres llevar a un grado más literario es una convención que utilizamos para que otros también puedan participar, al conocer el lenguaje, características y componentes de ese eventual producto.

En informática es un marco de trabajo, que define lenguaje, componentes y características que serán el soporte para los equipos de trabajo, es el esqueleto en términos conceptuales, lógicos y físicos que tiene un sistema.

Podemos encontrar lo **MVC** típicos, Visual Studio, Rails, Laravel y otros como **MPV**, Modelo , plantilla , vista.

Vamos a revisar la instalación de Django, para construir un sistema para facilitar la accesibilidad a personas con algún grado de discapacidad, auditiva principalmente.

¿Qué vamos a necesitar?, que tengas nociones básicas de base de datos, html, css y Python.

Lo primero es que conozcas la creación de ambientes virtuales en Python, permiten instalar cuantas versiones se requieran en un computador, además cada módulo se instala para x ambiente, de esa manera no tenemos problemas de dependencias entre módulos sin ocupar el Python nativo del computador(en el caso de Macos, Linux).

Para crear los ambientes virtuales puedes pinchar [acá](#)

Posterior a la instalación y creación de ambientes virtuales lo activamos y deberíamos tener algo como esto:

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

(amb36) es el ambiente en el que estoy trabajando, crearé un directorio para los proyectos

```
mkdir Django
```

Ingresaré al directorio

```
(amb36) MacBook-Pro-de-usuario:Django usuario
```

Instalamos Django

```
pip install Django==1.8.2
```

```
pip install --pre django #utilizaremos esta versión
```

*Si fallas puedes desinstalar con `pip uninstall Django==1.8.2` o solo `pip uninstall Django`

Creamos un proyecto llamado HolaMundo

```
django-admin startproject Accesible
```

Ingresamos al directorio HolaMundo

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

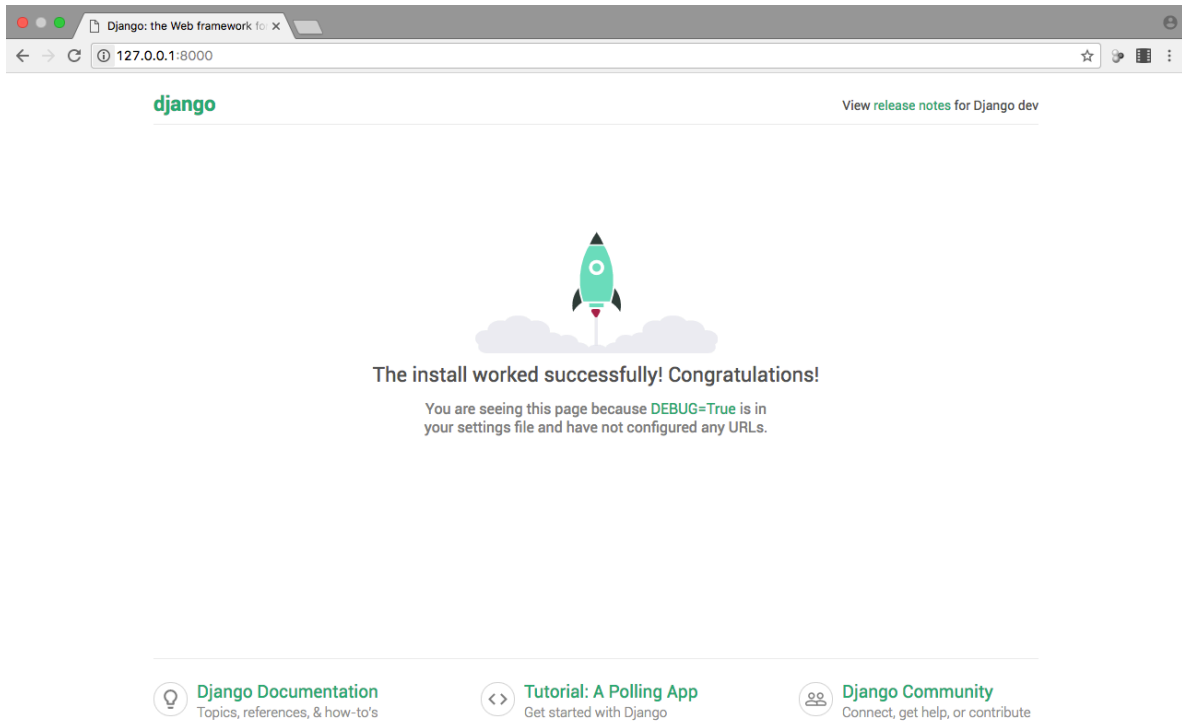
Ejecutamos la aplicación

```
python manage.py runserver
```

Ahora en el navegador

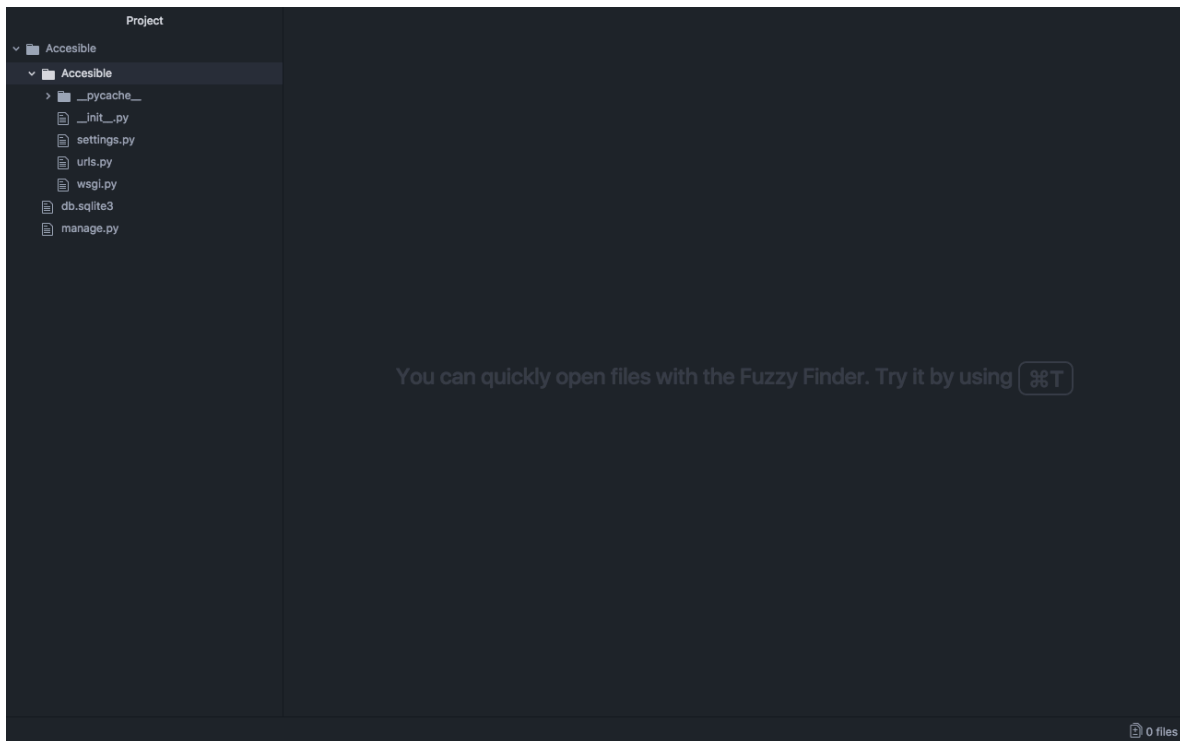
<http://127.0.0.1:8000/>

Se verá así



Django está funcionando

Para trabajar con la base de datos, te recomiendo usar PostgreSQL, encontrarás los manuales de instalación en mis repositorios. Un proyecto en Django tiene la siguiente estructura:



El nivel superior Accesible se puede cambiar, es solo un contenedor.

___init___py le indica a Python que este directorio se debe considerar un paquete Python

settings.py tiene la configuración de todo el proyecto Django

url.py la configuración de todas las url de tu sitio

wsgi.py una interfaz universal para conectar tu aplicación en servidores

https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface

Por defecto Django crea el servidor en el puerto 8080

```
python manage.py runserver
```

Puedes cambiarlo así:

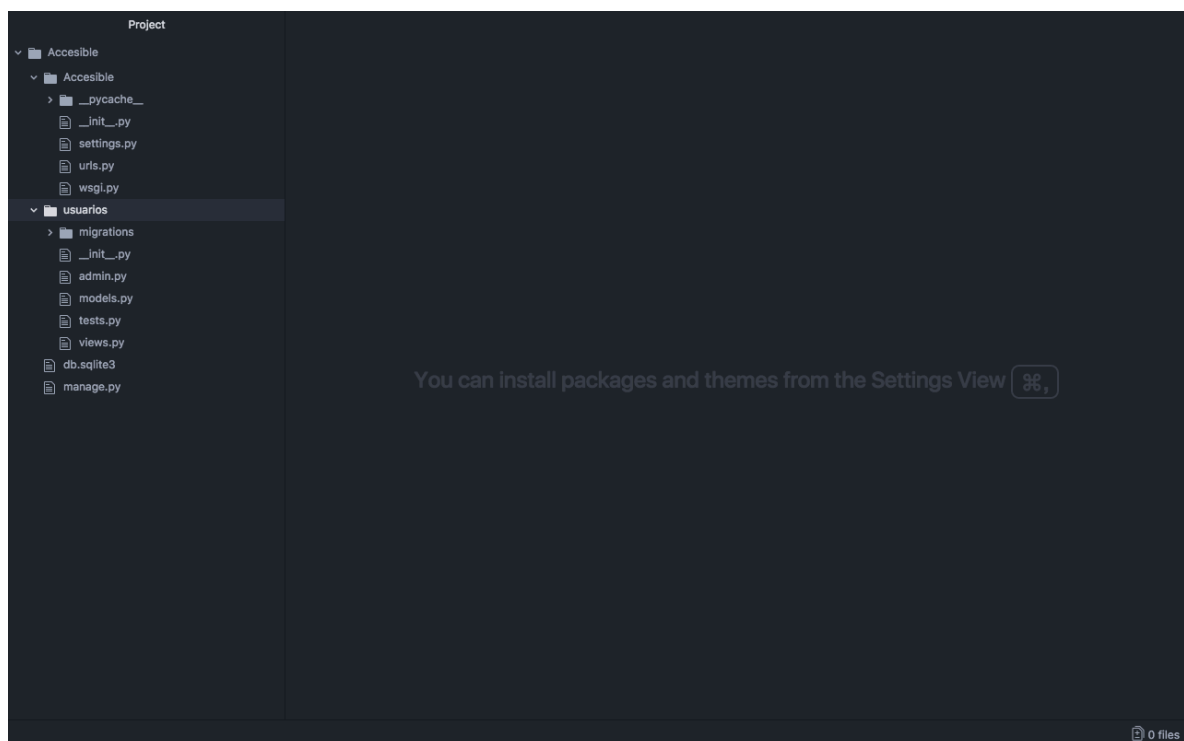
```
python manage.py runserver 0:8000 #El 0 es una abreviación de 127.0.0.1
```

APPS

Vamos a crear una aplicación, un proyecto puede tener muchas apps, que en general son los módulos de nuestro proyecto.

```
python manage.py startapp usuarios
```

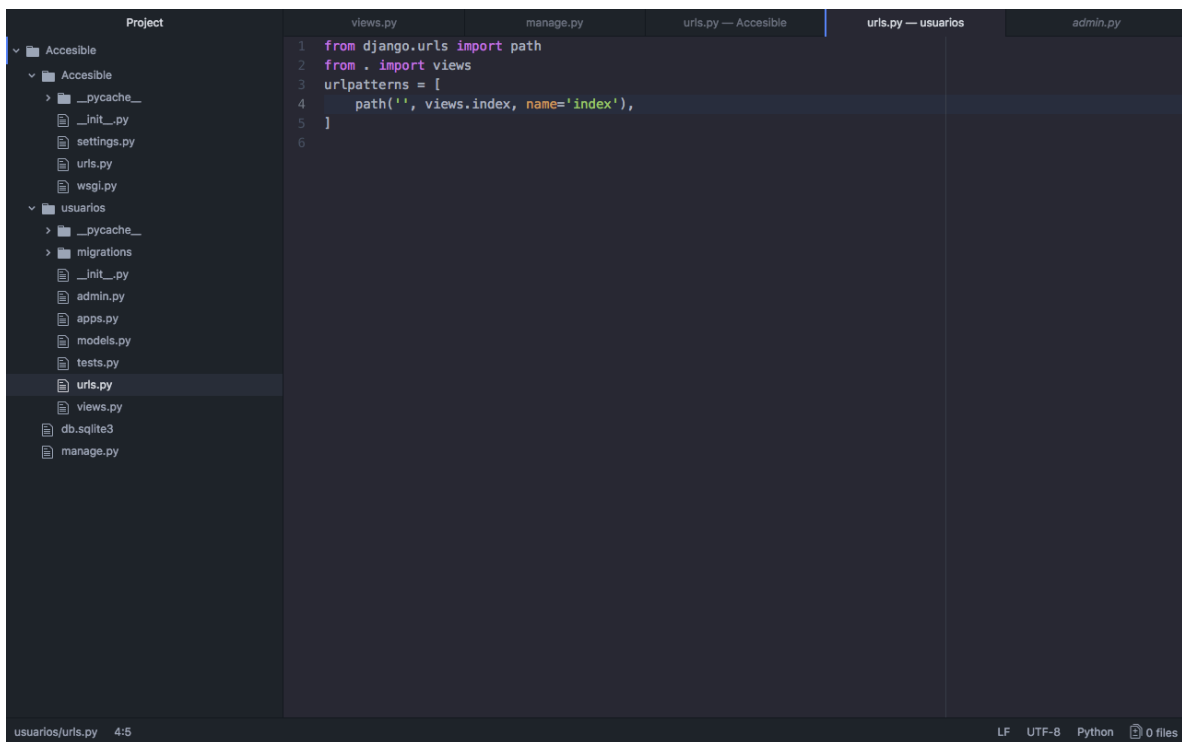
Esto creará automáticamente la aplicación, con admin, modelos, tests, views y su propio manage.



Creamos el archivo usuarios/urls.py y agregamos el siguiente código

```
from django.urls import path
from . import views
urlpatterns = [
    path("", views.index, name='index'),
]
```

Se verá así



En usuarios/views.py vamos a escribir

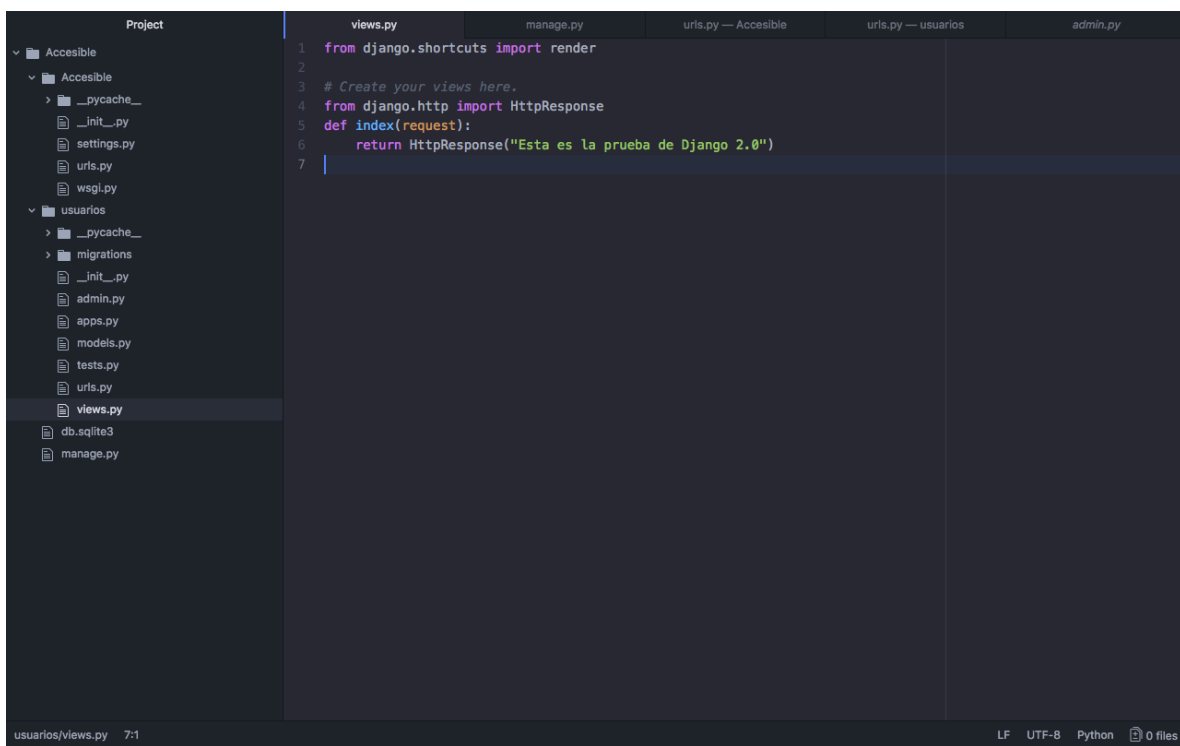
from django.http **import** HttpResponse

def **index**(request):

return HttpResponse("Este es nuestro proyecto Accesible!.")

Ahora crearemos el archivo urls.py adentro de nuestra aplicación

Se verá así



The screenshot shows a code editor with a dark theme. On the left is a file explorer showing the project structure. The 'usuarios' app is expanded, showing files like __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, and views.py. The 'views.py' file is selected and its content is displayed in the main editor area. The code in views.py is as follows:

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 from django.http import HttpResponse
5 def index(request):
6     return HttpResponse("Esta es la prueba de Django 2.0")
7
```

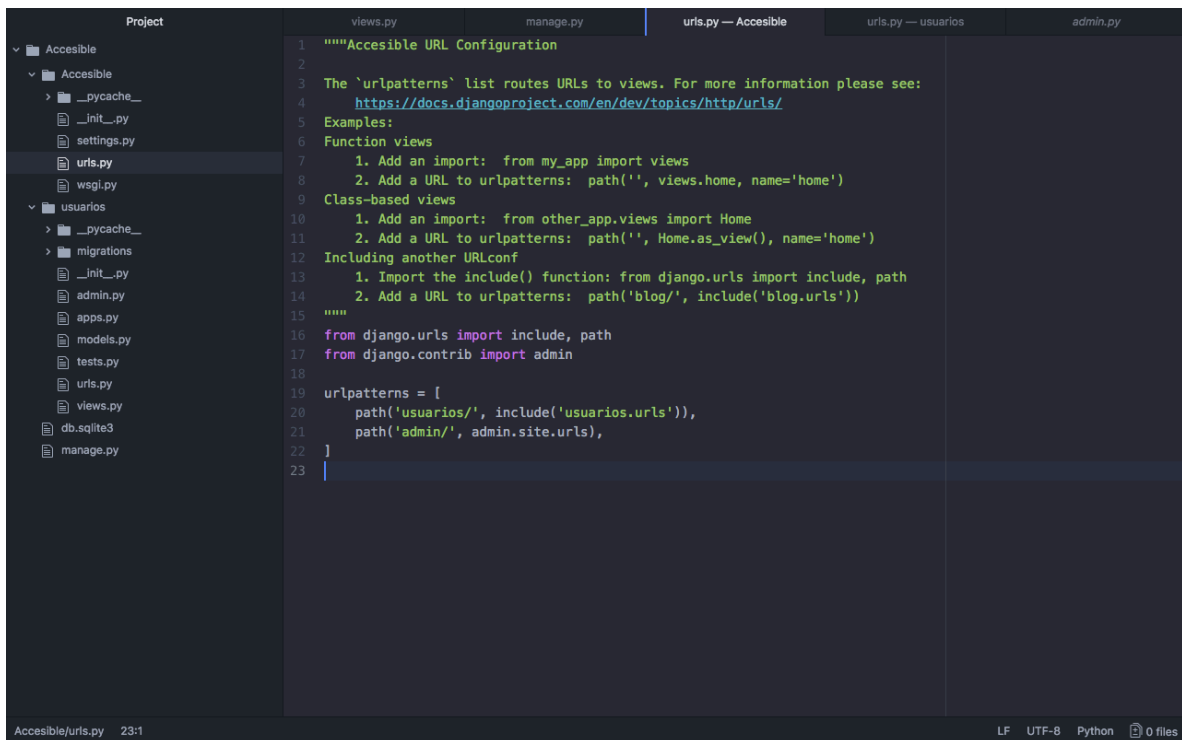
At the bottom of the editor, there is a status bar showing 'usuarios/views.py 7:1', 'LF', 'UTF-8', 'Python', and '0 files'.

En Accesible/urls.py

```
from django.urls import include, path
from django.contrib import admin
```

```
urlpatterns = [
    path('usuarios/', include('usuarios.urls')),
    path('admin/', admin.site.urls),
]
```

Se verá así



The screenshot shows a code editor with a project explorer on the left and a code editor on the right. The project explorer shows the following structure:

- Accesible
 - Accesible
 - __pycache__
 - __init__.py
 - settings.py
 - urls.py (selected)
 - wsgi.py
 - usuarios
 - __pycache__
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
 - db.sqlite3
 - manage.py

The code editor shows the following code in the file `urls.py`:

```
1 """Accesible URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/dev/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.urls import include, path
17 from django.contrib import admin
18
19 urlpatterns = [
20     path('usuarios/', include('usuarios.urls')),
21     path('admin/', admin.site.urls),
22 ]
23
```

The status bar at the bottom shows: Accesible/urls.py 23:1 LF UTF-8 Python 0 files

Corremos el servidor e ingresamos a

<http://127.0.0.1:8000/usuarios/>



Ya tenemos nuestra primera aplicación en Django!