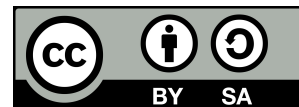


CURSO PYTHON

APLICACIONES WEB CON DJANGO



Autor: Jon Vadillo
www.jonvadillo.com



Contenidos

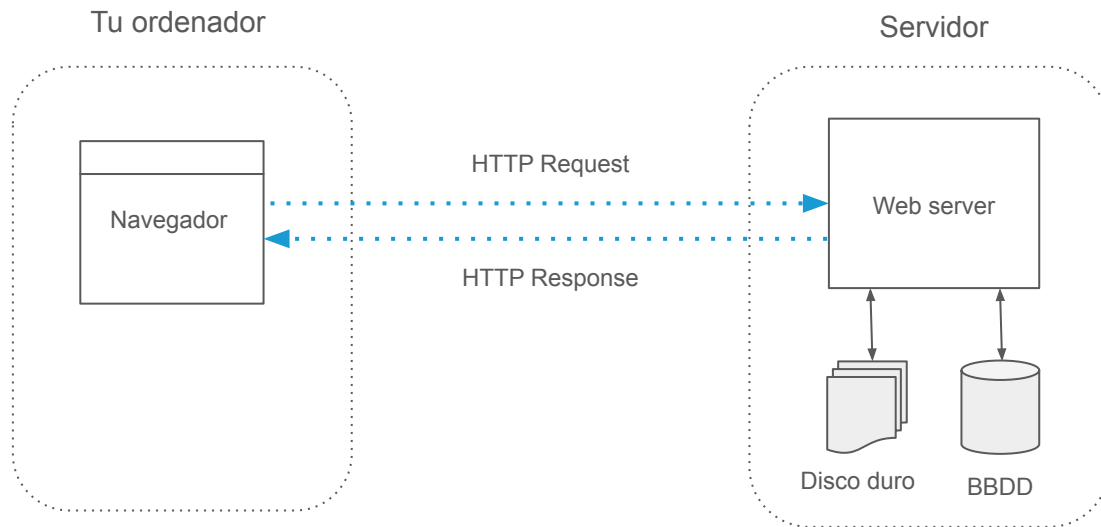
1. Introducción y **fundamentos básicos**
2. Crea tu primer **proyecto** en Django
3. Crea tu primera **aplicación** en django
4. El **modelo** en Django, **acceso a datos** y la aplicación de administrador
5. **Vistas** y **plantillas** en Django
6. **Vistas basadas en clases** (Class Based Views)
7. **Formularios** en Django

2. Crea tu primer proyecto en Django

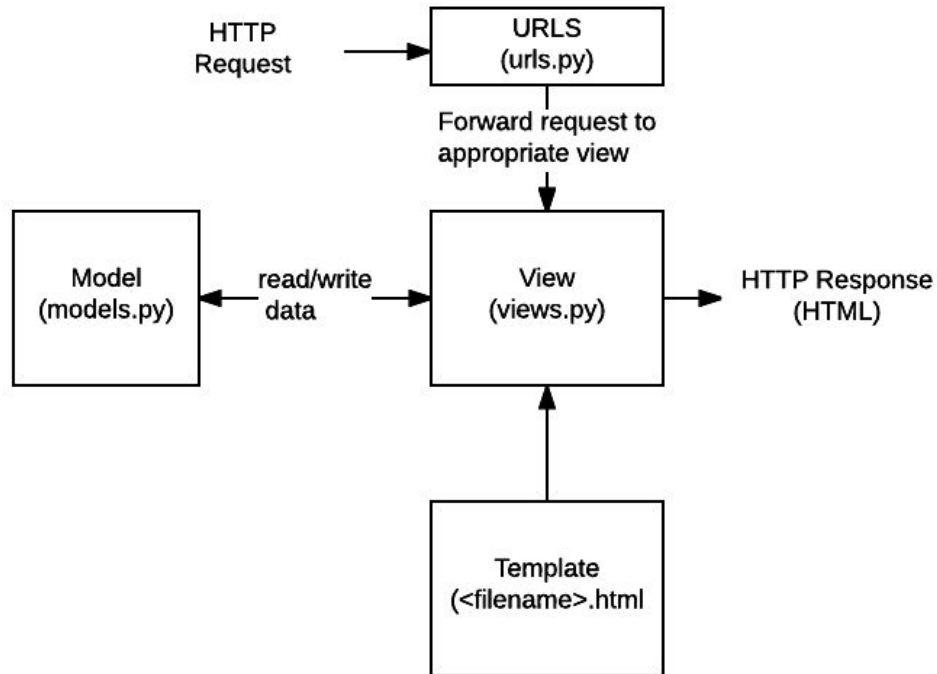
Funcionamiento de una aplicación web

1. La aplicación web **espera peticiones HTTP** del cliente (normalmente un navegador web)
2. Al recibir una petición la aplicación **extrae lo que se necesita** basándose en la URL y la información incluida en los datos POST o GET
3. **Realiza las operaciones necesarias** para satisfacer la petición (por ejemplo, leer/escribir en la base de datos)
4. La aplicación **devuelve una respuesta al cliente** (normalmente creando dinámicamente una página HTML para que el navegador la interprete y la muestre)

Funcionamiento de una aplicación web



Arquitectura de Django



Características principales

- **URLs:** El mapeador URL se usa para **redirigir las peticiones HTTP a la vista apropiada** basándose en la URL de la petición.
- **Vista (View):** **gestiona las peticiones HTTP que recibe y devuelve respuestas HTTP**. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formateo de la respuesta a las plantillas ("templates").

Características principales

- **Modelos (Models):** son objetos que **definen la estructura de los datos** de una aplicación y **proporcionan mecanismos** para gestionar (añadir, modificar y borrar) y consultar registros en la base de datos.
- **Plantillas (Templates):** una plantilla es un fichero de texto que **define una estructura de otro fichero**. Una vista puede crear dinámicamente una página usando una plantilla, rellenandola con datos de un modelo.

Crear un proyecto nuevo

- Ejecutar el comando: **django-admin startproject *mysite***
- Estructura creada:
 - `mysite`: Carpeta del proyecto, en este caso: mysite
 - `manage.py`: utilidad que permite interactuar con el proyecto
 - `__init__.py`: archivo vacío para que Python reconozca la carpeta como un Paquete y te permita usar sus objetos dentro de otras partes del proyecto
 - `settings.py`: archivo de configuración del proyecto
 - `urls.py`: define las URLs (puntos de entrada) de la aplicación
 - `wsgi.py`: se usa para ayudar a la aplicación a comunicarse con el servidor

Iniciar el servidor

- Para iniciar el servidor ejecutar: **python manage.py runserver**

```
>>> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

...

Django version 2.2.2, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Iniciar el servidor

- Comprobar que se accede correctamente desde `http://127.0.0.1:8000`

django

[View release notes for Django 2.2](#)



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



Django Documentation
Topics, references, & how-to's



Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

*Nota: se trata de un servidor ligero para pruebas, no se recomienda utilizarlo en entorno de producción.

Sources

- Documentación oficial: <https://www.djangoproject.com/>
- Mozilla MDN Web Docs: <https://developer.mozilla.org/>