



Agencia de
Aprendizaje
a lo largo
de la vida

DJANGO

Reunión 33

Django: Introducción

Les damos la bienvenida

Vamos a comenzar a grabar la clase

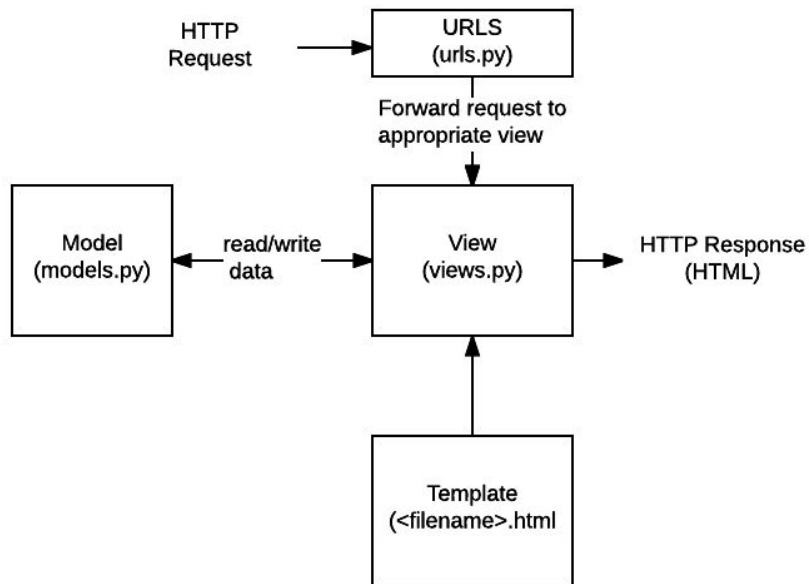
¿Qué es Django?

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles.

- **Gratuito.**
- **Código abierto.**
- **Gran comunidad y documentación.**
- **Incluye librerías extras.**
- **Escalable.**
- **Versátil.**

[Django Documentation](#)

Patrón MVT

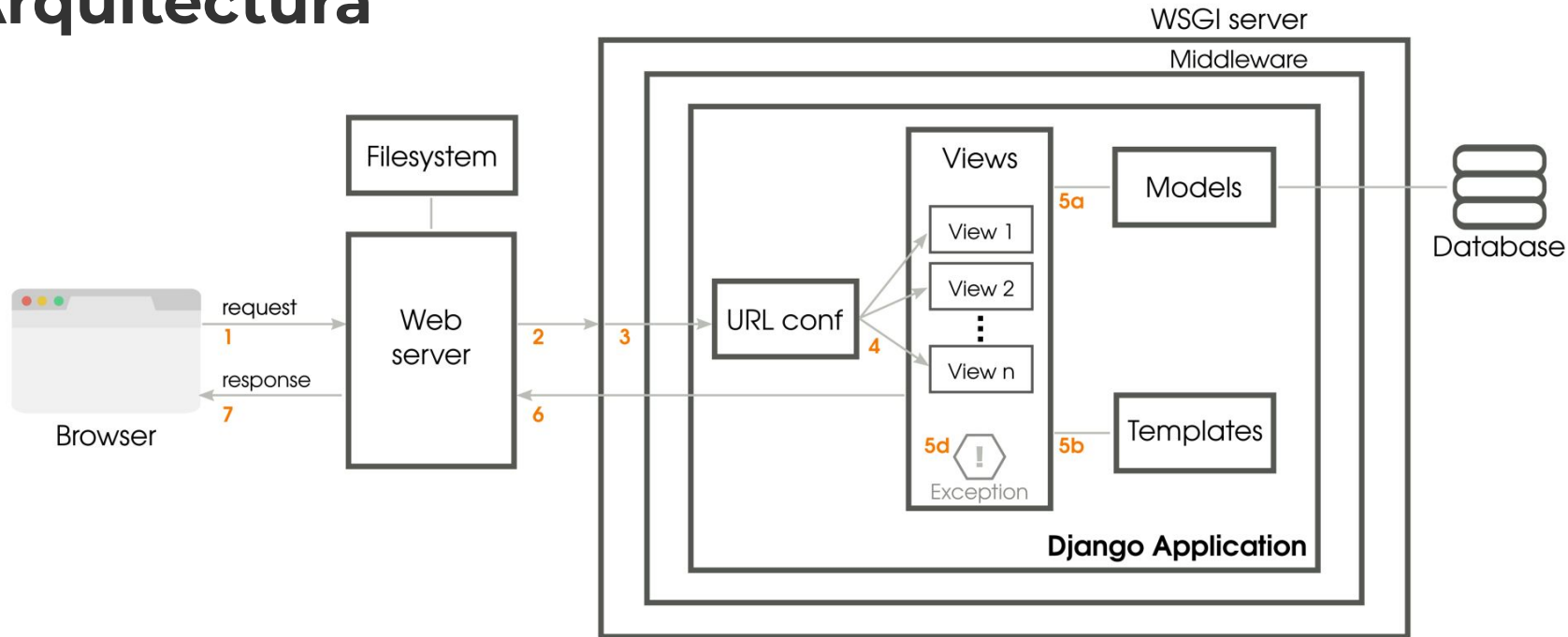


Modelo: la capa de acceso a la base de datos (ORM).

Vista: la capa de la lógica de negocios..

Template: (Plantilla), la capa de presentación.

Arquitectura



Instalación y configuración

Creación entorno virtual

```
python -m venv mientornovirtual
```

Activación entorno virtual

```
path\mentornovirtual\Scripts\activate
```

Instalación de Django

```
pip install Django==3.2.14
```

Verificación de Instalación de Django

```
python  
>>> import django  
>>> print(django.get_version())
```

Instalación y configuración

Creación del proyecto

```
django-admin startproject "nombreproyecto"
```

Verificación de creación del proyecto

Django provee un servidor web local para desarrollo, en el cual podremos probar lo que vamos modificando en el código de nuestro proyecto.

```
cd .\"nombreproyecto\"  
python manage.py runserver
```

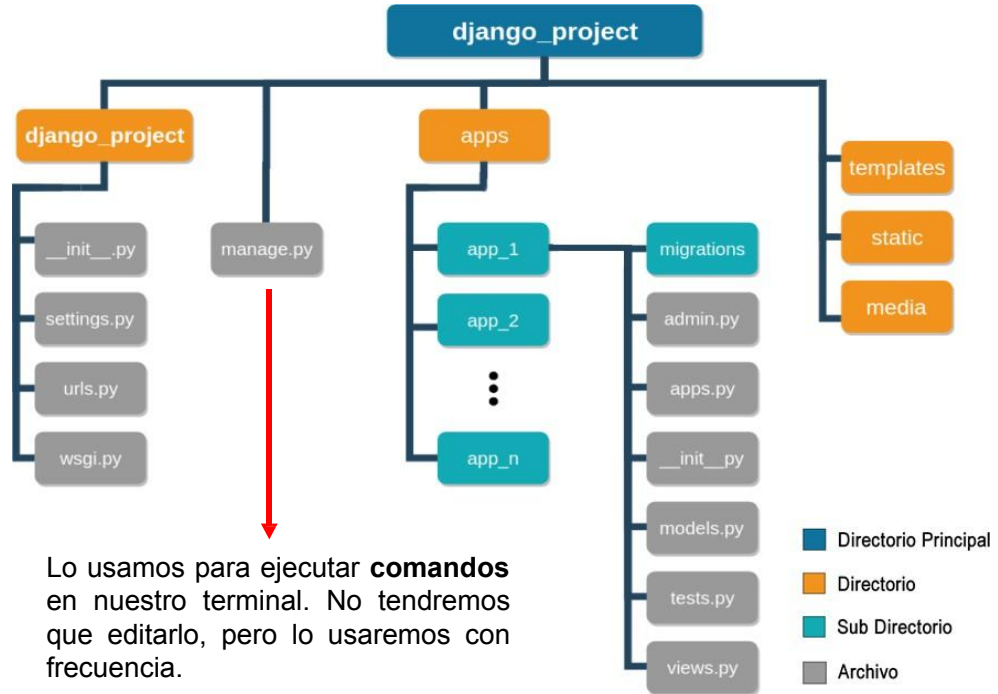
Estructura de directorios Django

Contiene algunos **ajustes** de configuración importantes.

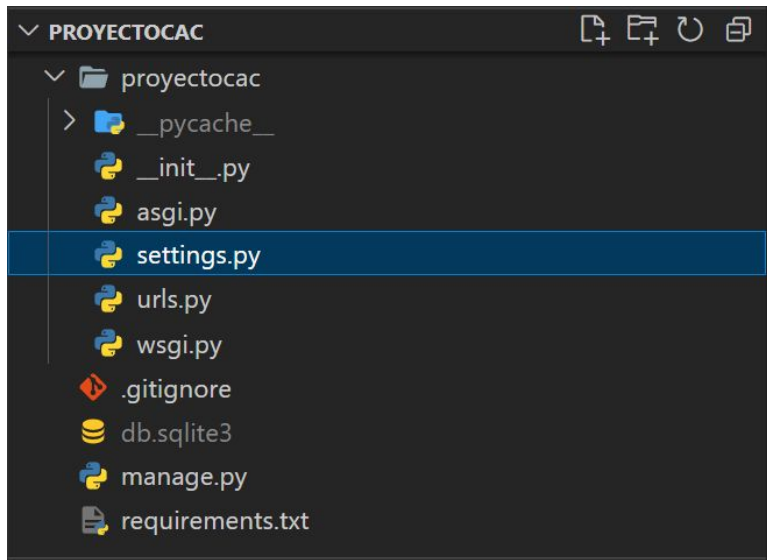
Es posible que deseemos cambiar algunos ajustes predeterminados de vez en cuando.

Contiene instrucciones sobre a dónde se debe dirigir a los usuarios después de navegar a una determinada **URL**.

Lo usamos para ejecutar **comandos** en nuestro terminal. No tendremos que editarlo, pero lo usaremos con frecuencia.



Estructura de directorios Django



manage.py: Una utilidad de la línea de comandos que le permite interactuar con este proyecto Django de diferentes formas.

__init__.py: Un archivo vacío que le indica a Python que este directorio debería ser considerado como un paquete Python.

settings.py: Ajustes/configuración para este proyecto Django.

urls.py: Las declaraciones URL para este proyecto Django.

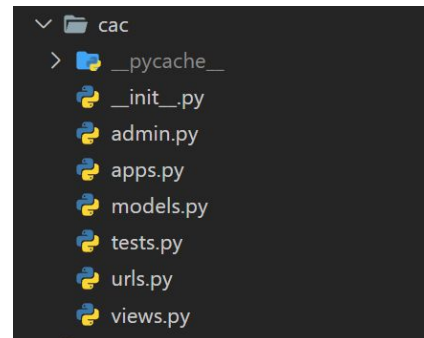
wsgi.py: Un punto de entrada para que los servidores web compatibles con WSGI puedan servir su proyecto.

asgi.py: Un punto de entrada para que los servidores web compatibles con ASGI puedan servir su proyecto.

Instalación y configuración

Creación de una aplicación

python manage.py startapp "cac"



Registrar aplicación en settings.py

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

Url dispatcher

Un esquema de URL limpio y elegante es un detalle importante en una aplicación web de alta calidad. Django te permite diseñar URL como quieras, sin limitaciones.

```
"""URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.urls import path

urlpatterns = [
]
```

**No te olvides de completar la
asistencia y consultar dudas**

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.

TODO EN EL AULA VIRTUAL