



Agencia de
Aprendizaje
a lo largo
de la vida

DJANGO

Clase 9

Django: Introducción

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 09**Django: Introducción**

- Patrón MVT
- Estructura de directorios de Django
- Proyecto, configuraciones, aplicaciones, requerimientos
- Url Dispatcher

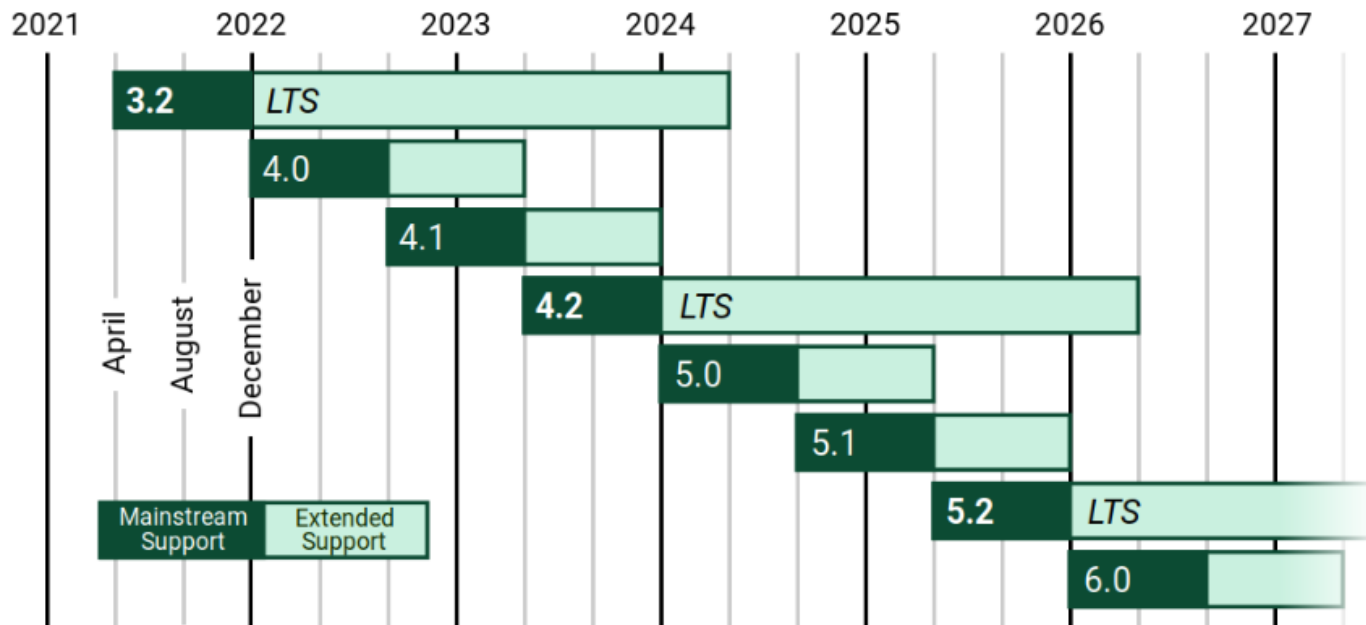
Clase 10**Django: Views – 1**

- Creación de rutas
- Múltiples vistas y URL
- HttpResponse
- Ejemplos con vistas genéricas

Versión de Python

Release	Released	Security Support	Latest
3.11	5 months ago (24 Oct 2022)	Ends in 4 years and 7 months (24 Oct 2027)	3.11.2 (07 Feb 2023)
3.10	1 year and 5 months ago (04 Oct 2021)	Ends in 3 years and 6 months (04 Oct 2026)	3.10.10 (07 Feb 2023)
3.9	2 years and 5 months ago (05 Oct 2020)	Ends in 2 years and 6 months (05 Oct 2025)	3.9.16 (06 Dec 2022)
3.8	3 years and 5 months ago (14 Oct 2019)	Ends in 1 year and 6 months (14 Oct 2024)	3.8.16 (06 Dec 2022)
3.7	4 years and 9 months ago (26 Jun 2018)	Ends in 3 months (27 Jun 2023)	3.7.16 (06 Dec 2022)
3.6	6 years ago (22 Dec 2016)	Ended 1 year and 3 months ago (23 Dec 2021)	3.6.15

Versión de Django



ref: [Download Django | Django \(djangoproject.com\)](https://www.djangoproject.com/download/)

Django + Python

Django version	Python versions
3.2	3.6, 3.7, 3.8, 3.9, 3.10 (added in 3.2.9)
4.0	3.8, 3.9, 3.10
4.1	3.8, 3.9, 3.10, 3.11 (added in 4.1.3)
4.2	3.8, 3.9, 3.10, 3.11
5.0	3.10, 3.11, 3.12

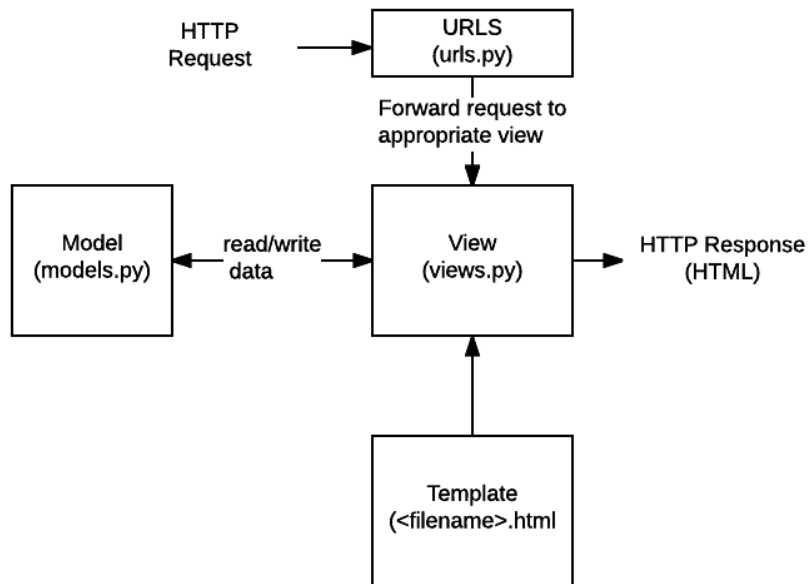
ref: [FAQ: Installation | Django documentation | Django \(djangoproject.com\)](#)

¿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.

Patrón MVT



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

Vista: la capa de la lógica de negocios..

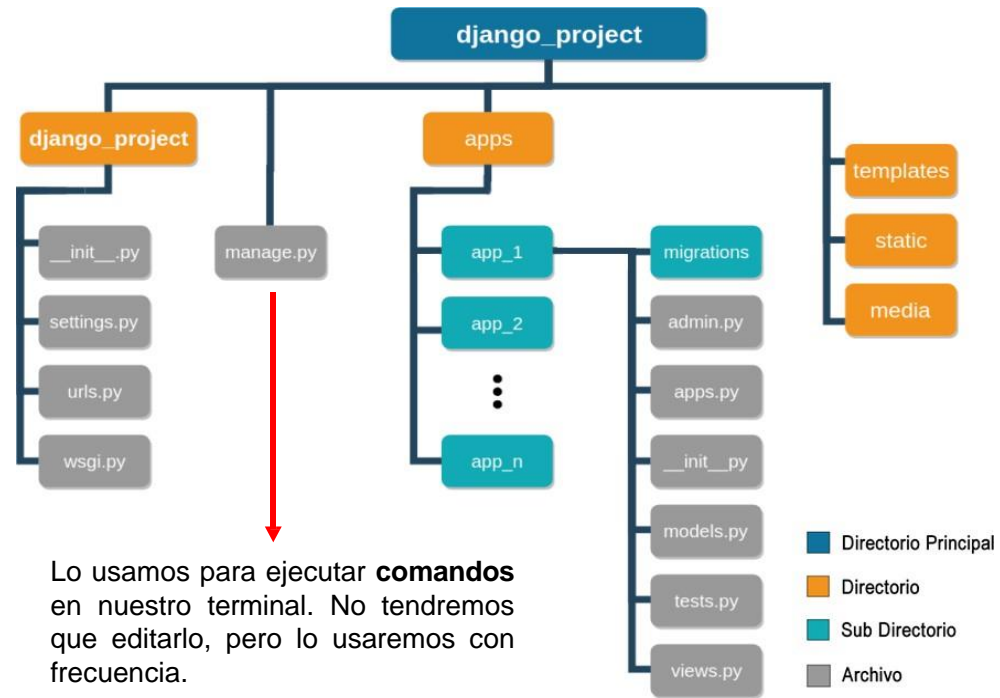
Template: (Plantilla), la capa de presentación.

Estructura de directorios Django

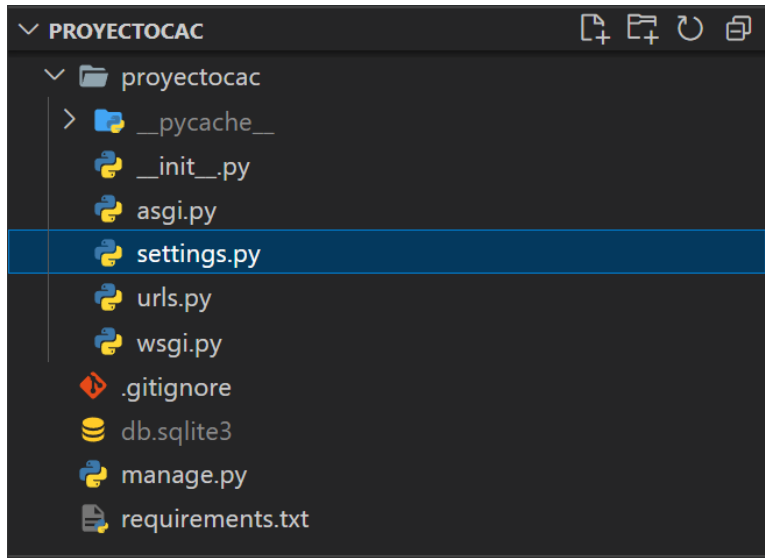
Contiene algunos **ajustes** de configuración importantes.
Es posible que deseemos cambiar algunos ajustes predeterminados de vez en cuando.

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

Activación entorno virtual

```
path\mientornovirtual\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

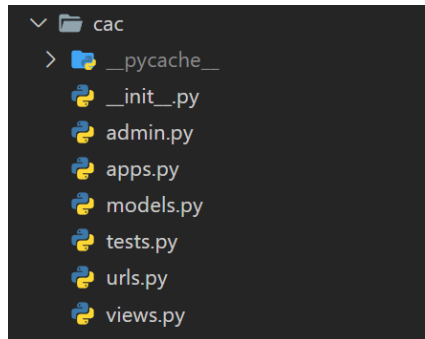
Django provee un servidor web local para desarrollo, en el cual podremos probar lo que vamos modificando en el código de nuestro proyecto.

```
cd .\“nombreproyecto”  
python manage.py runserver
```

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'  
]
```

Instalación y configuración

Creación de archivo requirements.txt

```
pip freeze > requirements.txt
```

Este comando nos generará un archivo requirements.txt que contendrá el listado de todas las librerías y sus versiones instaladas en nuestro virtual env para el proyecto. Esto es útil para que posteriormente podamos instalarlas directamente por medio de este archivo en cualquier otro entorno virtual donde llevemos al proyecto.

```
pip install -r requirements.txt
```

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.

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

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

**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