**INSTALANDO DJANGO**

Para usar django en un proyecto, es necesario crear un entorno virtual y luego instalar django en el mismo:

py -3 -m venv .venv

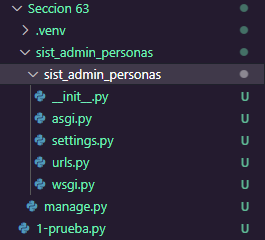
.venv\scripts\activate

python -m pip install django

**CREANDO UN PROJECTO**

django-admin startproject sist\_admin\_personas

Una vez ejecutado el proyecto este es el directorio que se genera:



Para ejecutar la app es necesario hacer un cd a la carpeta principal sist\_admin\_personas ya que ahí se encuentra el archivo **manage.py** que es el que permite ejecutar la aplicación en un servidor web.

**python** .\manage.py **runserver**

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

November 24, 2021 - 16:57:06

Django version 3.2.9, using settings 'sist\_admin\_personas.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

[24/Nov/2021 16:57:36] "GET / HTTP/1.1" 200 10697

[24/Nov/2021 16:57:37] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423

[24/Nov/2021 16:57:37] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692

[24/Nov/2021 16:57:37] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184

[24/Nov/2021 16:57:37] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876

Not Found: /favicon.ico

[24/Nov/2021 16:57:37] "GET /favicon.ico HTTP/1.1" 404 2123

Con CTRL+C finalizo el proceso. Cada vez que levante el sv necesito el command:

**python** .\manage.py **runserver**

**\_\_PY\_CACHE\_\_**

Cuando ejecuta un programa en python, el intérprete lo compila primero en bytecode (esto es una simplificación excesiva) y lo almacena en la \_\_pycache\_\_carpeta. Si observa allí, encontrará un montón de archivos que comparten los nombres de los archivos .py en la carpeta de su proyecto, solo sus extensiones serán .pyc o .pyo. Estas son versiones compiladas por bytecode y compiladas por bytecode de los archivos de su programa, respectivamente.

Como programador, puede ignorarlo en gran medida ... Todo lo que hace es hacer que su programa se inicie un poco más rápido. Cuando cambien sus scripts, se volverán a compilar, y si elimina los archivos o la carpeta completa y ejecuta su programa nuevamente, volverán a aparecer (a menos que suprima específicamente ese comportamiento)

Si está utilizando cpython (que es el más común, ya que es la implementación de referencia) y no desea esa carpeta, puede suprimirla iniciando el intérprete con el indicador -B, por ejemplo

**APP**

El concepto de app se define como una “parte” de una aplicación django que se encarga de “presentar distintas secciones”.

Por ejemplo, tener una app para mostrar la seguridad de la apliacacion, otra para mostrar el listado de personas, otra para configurar las rutas etc.

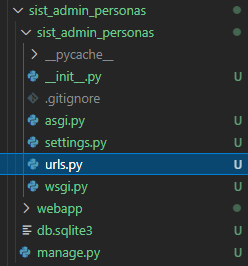
Para crear una app con el nombre webapp:

python manage.py startapp webapp

**CREANDO UNA VIEW**

Luego de haber creado la app, se define una **url** la cual se accederá para presentar esa “app”.

El archivo que configura las url está dentro de la carpeta principal



Dentro de urlpatterns se agrega una nueva ruta:

urlpatterns = [

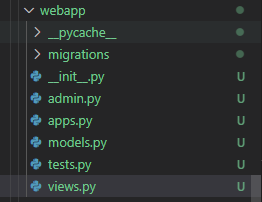
    path('admin/', admin.site.urls),

    path('webapp/', iniciar\_webapp)

]

En este caso la ruta será **webapp** e **iniciar\_webapp**  es un método que se definirá a continuación.

Para definir el método debemos ir al archivo **views** dentro de la nueva app creada:



Dentro de este comenzamos a definir el método que se llamará al ir a la url indicada:

def iniciar\_webapp(request):

    print(request)

    return HttpResponse(f"""<html>

    <head></head>

    <body>

        <h1>Hola mundo desde django</h1>

    </body>

    </html>""")

El parámetro request es información relacionada con la petición realizada al recurrir a la url indicada:

<WSGIRequest: GET '/webapp/'>

<class 'django.core.handlers.wsgi.WSGIRequest'>

El return es un objeto de clas HtppResponse que contiene el html de la pagina que el servidor mostrará

**PATH POR DEFECTO**

Para indicar el path por defecto:

    path('', iniciar\_webapp)