

ABPro – Módulo 6: Proyecto Grupal 1

Como primer ejercicio, levantaremos un servidor con Python, para que imprima el mensaje “Bienvenidos a TeLoVendo” en un navegador. Realícelo solo utilizando Python.

- ◆ Creo un archivo server.py con el siguiente código y lo ejecuto desde la línea de comandos usando “python server.py”

```
import http.server
import socketserver

class MiManejador(http.server.SimpleHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(b'Bienvenidos a TeLoVendo')

puerto = 8080

with socketserver.TCPServer(("", puerto), MiManejador) as httpd:
    print("Servidor en el puerto", puerto)
    httpd.serve_forever()
```

- ◆ Esto iniciará el servidor en el puerto puerto 8080. Luego, pruebo el server abriendo el navegador web e ingresando a la dirección http://localhost:8080 para ver el mensaje "Bienvenidos a TeLoVendo".

Ahora cree un entorno virtual e instalen Django.

```
python -m venv .grupallm6
.grupallm6\scripts\activate.bat
pip install Django
```

Verifiquen la versión de Python y Django.

```
python --version (Resultado: Python 3.11.3 = 3.11.3 es la versión de Python).
python -m django --version (Resultado: 4.2.1 = Esa es la versión de Django).
```

Ahora desarrollaremos nuestro primer proyecto utilizando Django. El objetivo es que muestre el mensaje “Bienvenidos a TeLoVendo”. Esta vez utilizaremos la arquitectura que nos brinda Django, para luego complejizar la aplicación. En esta primera instancia desarrollaremos una vista, basándonos en la documentación de Django en la página web de Django Project.

- ◆ Crear un proyecto de Django y una app del mismo (dentro del directorio del proyecto):

```
django-admin startproject telovendo (proyecto)
django-admin startapp telovendoapp (aplicación)
```

- ◆ En el archivo "views.py" de la aplicación, modificamos la función de vista "mensaje" para que muestre el mensaje “Bienvenido a TeLoVendo”:

```
from django.http import HttpResponse

def mensaje(request):
    return HttpResponse("Bienvenidos a TeLoVendo")
```

- ◆ En el caso de "urls.py" al interior de la carpeta de la aplicación (telovendoapp) tuvimos un problema el cual fue que el archivo no se creo por sí solo, por lo que decidimos crearlo e insertar el siguiente código:

```
from django.urls import path
from . import views

urlpatterns = [ path('', views.mensaje, name='telovendo'), ]
```

- ◆ Otro problema que nos surgió fue que al no crearse el archivo mencionado anteriormente tuvimos que ingresar una línea de código y un import en el archivo `urls.py` a nivel proyecto:

```

import include
[...]
path('', include('telovendoapp.urls')),
[...]

```

- ◆ Después en el archivo `settings.py` añadimos nuestra app a la lista de aplicaciones instaladas:

```

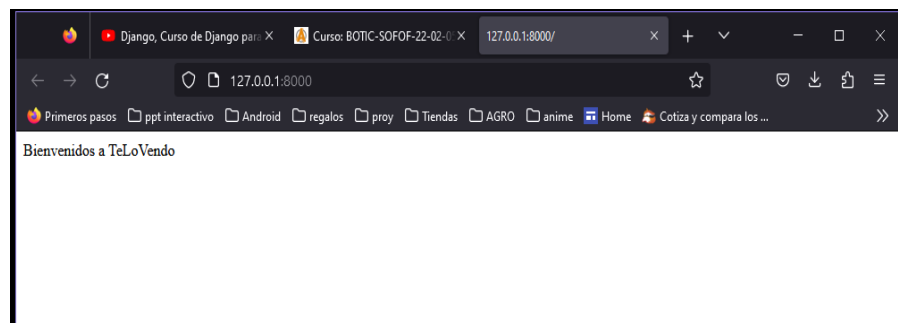
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'telovendoapp'
]

```

- ◆ Finalmente ejecutamos el servidor de desarrollo de Django con el siguiente comando:

```
python manage.py runserver
```

- ◆ Ahora, probamos lo realizado anteriormente ingresando a “`http://localhost:8000/`” en nuestro navegador web.



Identifique los diferentes archivos creados por Django al crear un proyecto y una aplicación. Describa en palabras simples la utilidad de cada script.

Los archivos creados por Django al iniciar un proyecto son:

- 1. manage.py:** Este es el archivo principal de administración de Django. Se utiliza para ejecutar comandos y realizar diversas tareas en el proyecto, como iniciar el servidor de desarrollo, crear migraciones, etc.
- 2. settings.py:** Contiene la configuración principal del proyecto Django. Aquí se definen aspectos como la base de datos a utilizar, las aplicaciones instaladas, las rutas de los archivos estáticos, entre otros.
- 3. urls.py:** Este archivo define las URL (direcciones web) que serán utilizadas por la aplicación. Aquí se mapean las URLs a las vistas correspondientes.
- 4. wsgi.py:** Este archivo es utilizado para desplegar el proyecto en un servidor web compatible con WSGI (Web Server Gateway Interface). WSGI es una especificación que define cómo un servidor web se comunica con una aplicación web.
- 5. asgi.py:** en Django se utiliza para configurar el servidor de aplicaciones ASGI (Asynchronous Server Gateway Interface) que permite manejar conexiones asíncronas y en tiempo real en aplicaciones web.

Luego, al crear una aplicación dentro del proyecto, se generan también los siguientes algunos archivos adicionales:

- 1. models.py:** Este archivo define los modelos de datos de la aplicación. Los modelos son representaciones de las tablas de la base de datos y se utilizan para almacenar y manipular los datos de la aplicación.
- 2. views.py:** Acá se definen las vistas de la aplicación. Las vistas son funciones o clases que procesan las solicitudes del usuario y generan una respuesta, que puede ser una página web, un archivo JSON, etc.
- 3. admin.py:** Este archivo permite registrar los modelos de la aplicación en el panel de administración de

Django. El panel de administración proporciona una interfaz para gestionar los datos de la aplicación de forma sencilla.

4. urls.py: Al igual que en el archivo de configuración del proyecto, este archivo define las URL específicas de la aplicación. Aquí se mapean las URLs a las vistas correspondientes dentro de la aplicación.

5. tests.py: El archivo tests.py se utiliza para escribir pruebas unitarias que verifiquen el correcto funcionamiento de los diferentes componentes de tu aplicación.

6. apps.py: aquí se define la configuración específica de la aplicación, como su nombre, configuración de la base de datos, de la aplicación de administración, de la internacionalización, etc.

