

10 DE SEPTIEMBRE, 2024
UDC

PROYECTO DJANGO, DESARROLLO DE APLICACIONES

MATERIA: PROGRAMACION WEB 2
PROFESOR: JULIO CASCO
ALUMNO: CAMILA SIMON



Tecn. Universitaria Desarrollo de Software
Camila Simon

1. Configuración del Proyecto Django:

- Crear un entorno virtual para el proyecto.
- Configurar un nuevo proyecto Django llamado TableroMensajes.
- Crear una aplicación dentro del proyecto llamada **mensajes**.

Creo una nueva carpeta llamada **TP2**, donde se ubicará el entorno, el proyecto Django y la aplicación. Creo el entorno virtual **TP2venv** y loactivo.

```
(entornoVirtual) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2# python3 -m venv TP2venv  
(entornoVirtual) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2# source TP2venv/bin/activate
```

Como siguiente paso, instalamos **Django**.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2# pip install django  
Collecting django  
  Using cached Django-5.1.1-py3-none-any.whl (8.2 MB)  
Collecting sqlparse>=0.3.1  
  Using cached sqlparse-0.5.1-py3-none-any.whl (44 kB)  
Collecting asgiref<4,>=3.8.1  
  Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)  
Collecting typing-extensions>=4  
  Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)  
Installing collected packages: typing-extensions, sqlparse, asgiref, django  
Successfully installed asgiref-3.8.1 django-5.1.1 sqlparse-0.5.1 typing-extensions-4.12.2
```

Posteriormente, creo el proyecto Django **TableroMensajes**.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2# django-admin startproject TableroMensajes  
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2# ls  
TP2venv  TableroMensajes
```

Nos ubicamos en la carpeta **TableroMensajes**, y creamos la aplicación **mensajes**.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python3 manage.py startapp mensajes
```

Ya terminado el primer punto, vamos a seguir con **GitHub**. Inicializamos con **git init** en la carpeta principal **TableroMensajes**, este comando se utiliza para inicializar un nuevo repositorio Git en un directorio, al ejecutar Git crea un repositorio vacío y permite eventualmente ser gestionado bajo el control de versiones.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# git init  
hint: Using 'master' as the name for the initial branch. This default branch name  
hint: is subject to change. To configure the initial branch name to use in all  
hint: of your new repositories, which will suppress this warning, call:  
hint:  
hint:   git config --global init.defaultBranch <name>  
hint:  
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and  
hint: 'development'. The just-created branch can be renamed via this command:  
hint:  
hint:   git branch -m <name>  
Initialized empty Git repository in /home/ProgWebII/TP2/TableroMensajes/.git/
```



Agregamos todos los archivos con **git add .**, realizamos el commit con una breve descripción de lo que se hizo hasta el momento, en este caso seria “**creación del entorno virtual, proyecto django y aplicación mensajes**”.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# git add .
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# git commit -m "creacion del entorno virtual, proyecto django y aplicacion mensajes"
[master (root-commit) fd84ec5] creacion del entorno virtual, proyecto django y aplicacion mensajes
 15 files changed, 217 insertions(+)
 create mode 100644 TableroMensajes/__init__.py
 create mode 100644 TableroMensajes/_pycache__/__init__.cpython-310.pyc
 create mode 100644 TableroMensajes/_pycache__/settings.cpython-310.pyc
 create mode 100644 TableroMensajes/asgi.py
 create mode 100644 TableroMensajes/settings.py
 create mode 100644 TableroMensajes/urls.py
 create mode 100644 TableroMensajes/wsgi.py
 create mode 100755 manage.py
 create mode 100644 mensajes/__init__.py
 create mode 100644 mensajes/admin.py
 create mode 100644 mensajes/apps.py
 create mode 100644 mensajes/migrations/__init__.py
 create mode 100644 mensajes/models.py
 create mode 100644 mensajes/tests.py
 create mode 100644 mensajes/views.py
```

Siguentemente, creamos un repositorio en GitHub. Agregamos el repositorio remoto al repositorio local, con el siguiente comando: **git remote add origin url_del_repositorio**. Después, solo queda subirlo a GitHub, con: **git push -u origin master**.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# git
push -u origin master
Username for 'https://github.com': KORATcs
Password for 'https://KORATcs@github.com':
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (19/19), 4.55 KiB | 1.52 MiB/s, done.
Total 19 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/KORATcs/TP2-PROGWEB2/pull/new/master
remote:
remote: To https://github.com/KORATcs/TP2-PROGWEB2
 * [new branch]      master -> master
Branch 'master' set up to track remote br
```

The screenshot shows a GitHub repository named 'TP2-PROGWEB2'. The repository is public and contains 2 branches (master and main) and 0 tags. The master branch is ahead of the main branch by 1 commit. A recent commit from the user 'KORATcs' is visible, titled 'creacion del entorno virtual, proyecto django y aplicacion mensajes'. Below this commit, there are other commits for 'TableroMensajes', 'mensajes', and 'manage.py'. At the bottom of the page, there is a section to 'Add a README'.



Realizaremos varios cambios, y desarrollaremos el modelo y vistas de la aplicación y el proyecto para finalizar la configuración.

Dentro de la carpeta **TableroMensajes**, en el archivo **settings.py**, ubicamos **INSTALLED_APPS** y agregamos la aplicación que creamos, después ubicamos **TEMPLATES** y agregamos la carpeta templates que es donde estarán las vistas (los archivos html).

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'mensajes' #agregamos la aplicacion mensajes en settings.py de TableroMensajes
]
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'], #agrego la carpeta templates
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

2. Implementación de Funcionalidades:

- *Crear Mensajes*: Implementar la funcionalidad que permita a los usuarios crear nuevos mensajes.
- *Ver Mensajes Recibidos*: Desarrollar una vista que muestre los mensajes recibidos por un destinatario.
- *Ver Mensajes Enviados*: Desarrollar una vista que muestre los mensajes enviados por un remitente.
- *Eliminar Mensajes*: Implementar la funcionalidad que permita a los usuarios eliminar mensajes.



Dentro de la carpeta **mensajes**, creamos el archivo **urls.py** que más adelante vamos a configurar. Ahora solo nos ubicamos en **models.py**, y desarrollaremos el modelo de la aplicación.

MODELO DE MENSAJES

```
1  from django.db import models
2  from django.utils import timezone
3
4  # Create your models here.
5
6  class Mensaje (models.Model):
7      texto = models.TextField()
8      remitente = models.CharField(max_length=1000)
9      destinatario = models.CharField(max_length=1000)
10     fecha_hora_envio = models.DateTimeField(default=timezone.now)
```

Realizamos las migraciones correspondientes una vez hecho el modelo.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python3 manage.py makemigrations
  Migrations for 'mensajes':
    mensajes/migrations/0001_initial.py
      + Create model Mensaje
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python3 manage.py migrate mensajes
  Operations to perform:
    Apply all migrations: mensajes
  Running migrations:
    Applying mensajes.0001_initial... OK
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes#
```

Utilizamos **makemigrations** para crear archivos de migración basados en los cambios que se hagan en los modelos, sirve para generar instrucciones que luego serán ejecutadas para modificar la estructura de la base de datos sin que tengas que escribir SQL manualmente. Utilizamos **migrate** para aplicar las migraciones que se han generado por el **makemigrations** a la base de datos. Django ejecuta los cambios en la estructura de la base de datos basándose en las instrucciones definidas en los archivos de migración.

Implementamos un formulario que capture la información del mensaje y la guarde en una base de datos.

```
mensajes > forms.py > ...
1  from django import forms
2  from .models import Mensaje
3
4  class MensajeForm(forms.ModelForm):
5      class Meta:
6          model = Mensaje
7          fields = ['remitente', 'destinatario', 'texto']
8
```



En el archivo **views.py** le vamos a crear todas las funciones necesarias para el correcto funcionamiento de la aplicación, para poder visualizar los mensajes enviados y recibidos, crearlos y eliminarlos.

```
from django.shortcuts import render get_object_or_404 redirect
from django.views import View
from models import Mensaje
from     import

class VerMensajesView View
    def get self request
        mensajes = Mensaje objects all
        return
    render request 'mensajes/verMensajes.html' 'mensajes' mensajes

def crearMensaje request
    if request == 'POST'
        form =
            request
        if form
            mensaje = form commit=False
            mensaje
            return redirect 'mensajes:verMensajes'
        else
            form =
    return render request 'mensajes/crearMensajes.html' 'form'
form

class VerMensajesPorUsuarioView View
    def get self request
        usuario = request 'usuario' None

        if usuario
            mensajes_recibidos =
Mensaje objects filter destinatario_icontains=usuario
            mensajes_enviados =
Mensaje objects filter remitente_icontains=usuario

        else
            return render request
'mensajes/verMensajesUsuario.html' 'mensajes_enviados' 'mensajes_recibidos'
```



```

        return render request
'mensajes/verMensajesUsuario.html'  'mensajes_enviados' mensajes_enviados
'mensajes_recibidos' mensajes_recibidos

class EliminarMensaje View
    def post self request mensaje_id
        mensaje = get_object_or_404 Mensaje pk=mensaje_id
        mensaje delete
        return redirect 'mensajes:verMensajes'

```

3. Estructura de Plantillas:

- Crear una estructura de plantillas reutilizable mediante el uso de herencia y reutilización de plantillas.
- Implementar el uso del Django Template Language (DTL) para renderizar datos en las plantillas.

TEMPLATES

Realizamos los archivos **.html**. Una plantilla base que va a contener lo principal: **mensajes.html**, y que va a servir como guía para las demás plantillas. Luego tendremos las plantillas: **verMensajes.html**, **crearMensajes.html**, **eliminarMensajes.html** y **crearMensajesUsuario.html**. Estas cuatro últimas heredan propiedades de **mensajes.html**.

mensajes.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{% block title %}Mensajes{% endblock %}</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
    <style>
        body {
            background-color: #212529; /* Gris oscuro más oscuro */
            color: white; /* Color del texto en el cuerpo */
        }
        .content {
            padding-bottom: 80px; /* Espacio para los botones */
        }

```



```

.btn-custom {
    background-color: #6f42c1; /* Color púrpura */
    border: none; /* Sin borde */
}
.btn-custom:hover {
    background-color: #5a31a8; /* Color púrpura más oscuro al
pasar el mouse */
}
.button-container {
    background-color: black; /* Fondo negro para la barra de
botones */
    padding: 10px 10px; /* Espaciado vertical */
    position: absolute; /* Posición absoluta para que cubra
solo desde los botones hacia abajo */
    bottom: 0; /* Alinear en la parte inferior */
    width: 100%; /* Cubrir todo el ancho */
}

```

</style>

</head>

<body class="d-flex flex-column min-vh-100">

<div class="d-flex justify-content-end p-3">

<a href="{% url 'mensajes:verMensajes' %}" class="btn
btn-custom">Inicio

</div>

<div class="content flex-fill">

<h1 class="text-center my-5">{% block header %}{% endblock
%}</h1>

{%
 block content %}{%
 endblock %}
</div>

<div class="button-container text-center">

<div class="container">

<a href="{% url 'mensajes:crearMensaje' %}" class="btn
btn-custom">Crear Mensaje

<a href="{% url 'mensajes:verMensajesUsuario' %}" class="btn
btn-custom">Ver Mensajes por Usuario

<a href="{% url 'mensajes:verMensajes' %}" class="btn
btn-custom">Ver Mensajes

</div>

</div>



```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIds1K
1eN7N6jIeHz" crossorigin="anonymous"></script>
</body>
</html>

```

verMensajes.html

```

<div class "container">
    <ul class "list-group">

        <li class "list-group-item d-flex
justify-content-between align-items-start">
            <div class "flex-grow-1">
                <strong>    </strong>
<br>
                <strong>      </strong>
<br>
                <span>          </span>
            </div>
            <div class "text-end" style "min-width: 120px;">
                <small class "text-muted">
                    </small>
            </div>
            <form action "{% url 'mensajes:eliminarMensaje'
mensaje.id %}" method "post" style "display:inline;">

                <button type "submit" class "btn btn-danger
btn-sm">
                    </button>
                </form>
            </li>

            <li class "list-group-item">
</li>

    </ul>
</div>

```



verMensajesUsuario.html

```
<div class "container mt-5">

    <!-- Formulario de búsqueda -->
    <form method "GET" action "{% url 'mensajes:verMensajesUsuario'
%" class "mb-4">
        <div class "input-group">
            <input type "text" name "usuario" class "form-control
rounded mr-2" placeholder "Buscar por usuario" style "border-radius:
20px;">
            <div class "input-group-append">
                <button class "btn btn-primary"
type "submit">      </button>
            </div>
        </div>
    </form>

    <!-- Sección de mensajes -->
    <div class "row">
        <!-- Columna de Mensajes Enviados -->
        <div class "col-md-6">
            <h2>          </h2>
            <ul class "list-group">

                <li class "list-group-item d-flex
justify-content-between align-items-center">
                    <div>
                        <strong>  </strong>
                    <br>
                        <strong>    </strong>
                    <br>
                        <span>
                        </span>
                    </div>
                </li>
            </ul>
        </div>
    </div>
```



```

                <form action="{% url
'mensajes:eliminarMensaje' mensaje.id %}" method="post"
style="display:inline;">

                    <button type="submit" class="btn
btn-danger btn-sm">      </button>
                </form>
            </li>

            <li class="list-group-item">
        </li>

    </ul>
</div>

<!-- Columna de Mensajes Recibidos --&gt;
&lt;div class="col-md-6"&gt;
    &lt;h2&gt;          &lt;/h2&gt;
    &lt;ul class="list-group"&gt;

        &lt;li class="list-group-item d-flex
justify-content-between align-items-center"&gt;
            &lt;div&gt;
                &lt;strong&gt;    &lt;/strong&gt;
            &lt;br&gt;
                &lt;strong&gt;    &lt;/strong&gt;
            &lt;br&gt;
                &lt;span&gt;
                &lt;/span&gt;
            &lt;/div&gt;
            &lt;form action="{% url
'mensajes:eliminarMensaje' mensaje.id %}" method="post"
style="display:inline;"&gt;

                &lt;button type="submit" class="btn
btn-danger btn-sm"&gt;      &lt;/button&gt;
            &lt;/form&gt;
        &lt;/li&gt;

        &lt;li class="list-group-item"&gt;
    &lt;/li&gt;

    &lt;/ul&gt;
</pre>

```



```
        </div>
    </div>

</div>
```

eliminarMensajes.html

```
<h2>          </h2>
<p>
    </p>

<form method "post">

    <button type "submit">      </button>
    <a href "{% url 'mensajes:ver_mensajes' %}">      </a>
</form>
```

crearMensaje.html

```
<div class "d-flex flex-column justify-content-center
align-items-center" style "height: 80vh; margin-top: 20px;">
    <div class "form-container w-100 mw-md-500 p-4 border rounded
shadow bg-light">
        <h2 class "text-center mb-4">          </h2>
        <form method "post">

            <button type "submit" class "btn btn-primary w-100">
```



```

        </button>
    </form>
</div>
</div>

<style>
    textarea input type="text"
        transition           0.3s /* Transición suave para el
cambio de color */

    textarea:hover input type="text" :hover
        border-color white /* Cambia el color del borde al pasar
el cursor */
        box-shadow 0 0 5px rgba 255 255 255 0.5 /* Añade un
efecto de sombra */

/* Establecer un tamaño fijo para la ventana */
html body
    height 100vh /* Altura fija para la ventana */
    overflow hidden /* Evita el desplazamiento */

</style>

```

En la terminal, utilizamos el comando **python manage.py shell** para ingresar los mensajes a la base de datos, y con **from mensajes.models import Mensaje** nos aseguramos de ingresar correctamente todo.

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python manage.py shell
Python 3.10.12 (main, Jul 29 2024, 16:56:48) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from mensajes.models import Mensaje
```

Algunos de los mensajes ingresados:

```
>>> mensaje1 = Mensaje(texto= 'Hola, como estas? Ayer te vi en el Shopping pero me dio verguenza saludarte :(', remitente='J.Balvin', destinatario='Rosalia')er te vi en el Shopping pero me dio verguenza saludar>>> quit
Use quit() or Ctrl-D (i.e. EOF) to exit
>>> quit()
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python manage.py shell
Python 3.10.12 (main, Jul 29 2024, 16:56:48) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from mensajes.models import Mensaje
>>> mensaje1 = Mensaje(texto= 'Hola, como estas? Ayer te vi en el Shopping pero me dio verguenza saludarte :(', remitente='J.Balvin', destinatario='Rosalia')
>>> mensaje1.save()
>>> mensaje2 = Mensaje(texto= 'Holaa, que pena! Me hubieras saludado, tenia ganas de verte un ratito', remitente='Rosalia', destinatario='J.Balvin')
>>> mensaje2.save()
>>> mensaje3 = Mensaje(texto= 'Genial, entonces para la proxima. Te invito un helado en Bonke', remitente='J.Balvin', destinatario='Rosalia')
```

Verificación de que se ingresaron correctamente:



```

mensaje in mens>>> for mensaje in mensajes:
...     print(f'Remitente: {mensaje.remitente}, Contenido: {mensaje.texto}, Destinatario: {mensaje.destinatario}')
...
Remitente: J.Balvin, Contenido: Hola, como estas? Ayer te vi en el Shopping pero me dio verguenza saludarte :, Destinatario: Rosalia
Remitente: Rosalia, Contenido: Holaa, que penal! Me hubieras saludado, tenia ganas de verte un ratito, Destinatario: J.Balvin
Remitente: J.Balvin, Contenido: Genial, entonces para la proxima. Te invito un helado en Bonke, Destinatario: Rosalia
Remitente: Conn, Contenido: cheee, estaria genial que juguemos unas partidas de DBD, Destinatario: Ruben
Remitente: Ruben, Contenido: nooo, sos muy manco, si no sabes jugar competitivo no >:, Destinatario: Conn
Remitente: Conn, Contenido: malo rubencito :((, Destinatario: Ruben
Remitente: Ñoki, Contenido: meaw, meawww, meawww, Destinatario: Perlita
Remitente: Perlita, Contenido: meaw :000, Destinatario: Ñoki
>>
>>

```

Una vez hecho esto, levantamos el servidor para que se puedan visualizar las vistas del proyecto en un navegador Web. Navegamos hasta la carpeta de nuestro proyecto (donde se encuentra `manage.py`) y ejecutamos: **python manage.py runserver**. Esto inicia el servidor y estará disponible para verse ya.

```

(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, mensajes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 10, 2024 - 02:54:59
Django version 5.1.1, using settings 'TableroMensajes.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

```

Veremos a continuación cómo quedaron las vistas con los mensajes:

4. Repositorio GitHub:

- Crear un repositorio público en GitHub para el proyecto TableroMensajes.
- Asegurarse de incluir todos los archivos relevantes, incluyendo `requirements.txt` y un archivo `README.md` con la documentación del proyecto.

Ya tenemos el repositorio de GitHub creado para el proyecto. Y ahora con el siguiente comando: **pip freeze > requirements.txt**. Con esto, creamos un archivo que nos especifica las dependencias que el proyecto necesita para funcionar correctamente. Es una lista de los paquetes y sus versiones que deben ser instalados en el entorno virtual para garantizar que el código funcione de manera consistente en diferentes entornos.



```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# pip freeze > requirements.txt
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# ls
TableroMensajes  db.sqlite3  manage.py  mensajes  requirements.txt
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes#
```

Automáticamente se agrega a nuestro editor de código, si abrimos el archivo desde Visual Studio Code, nos muestra lo siguiente:

The screenshot shows the Visual Studio Code interface. On the left is a tree view of the project structure:

- apps.py
- forms.py
- models.py
- tests.py
- urls.py
- views.py
- TableroMensajes
 - _pycache_
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
- db.sqlite3
- manage.py
- requirements.txt

On the right is the content of the requirements.txt file:

```
1 asgiref==3.8.1
2 Django==5.1.1
3 sqlparse==0.5.1
4 typing_extensions==4.12.2
5
```

Luego le instalamos todas las dependencias:

```
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# cd requirements.txt
-bash: cd: requirements.txt: Not a directory
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes# pip install -r requirements.txt
Requirement already satisfied: asgiref==3.8.1 in /home/ProgWebII/TP2/TP2venv/lib/python3.10/site-packages
 (from -r requirements.txt (line 1)) (3.8.1)
Requirement already satisfied: Django==5.1.1 in /home/ProgWebII/TP2/TP2venv/lib/python3.10/site-packages
 (from -r requirements.txt (line 2)) (5.1.1)
Requirement already satisfied: sqlparse==0.5.1 in /home/ProgWebII/TP2/TP2venv/lib/python3.10/site-packages
 (from -r requirements.txt (line 3)) (0.5.1)
Requirement already satisfied: typing_extensions==4.12.2 in /home/ProgWebII/TP2/TP2venv/lib/python3.10/site-packages
 (from -r requirements.txt (line 4)) (4.12.2)
(TP2venv) root@DESKTOP-4IMSG6T:/home/ProgWebII/TP2/TableroMensajes#
```

Link del repositorio:

<https://github.com/KORATcs/TP2-PROGWEB2.git>

