



MINISTERIO DEL TRABAJO

Configurar Proyecto Web Django + Python + MySQL

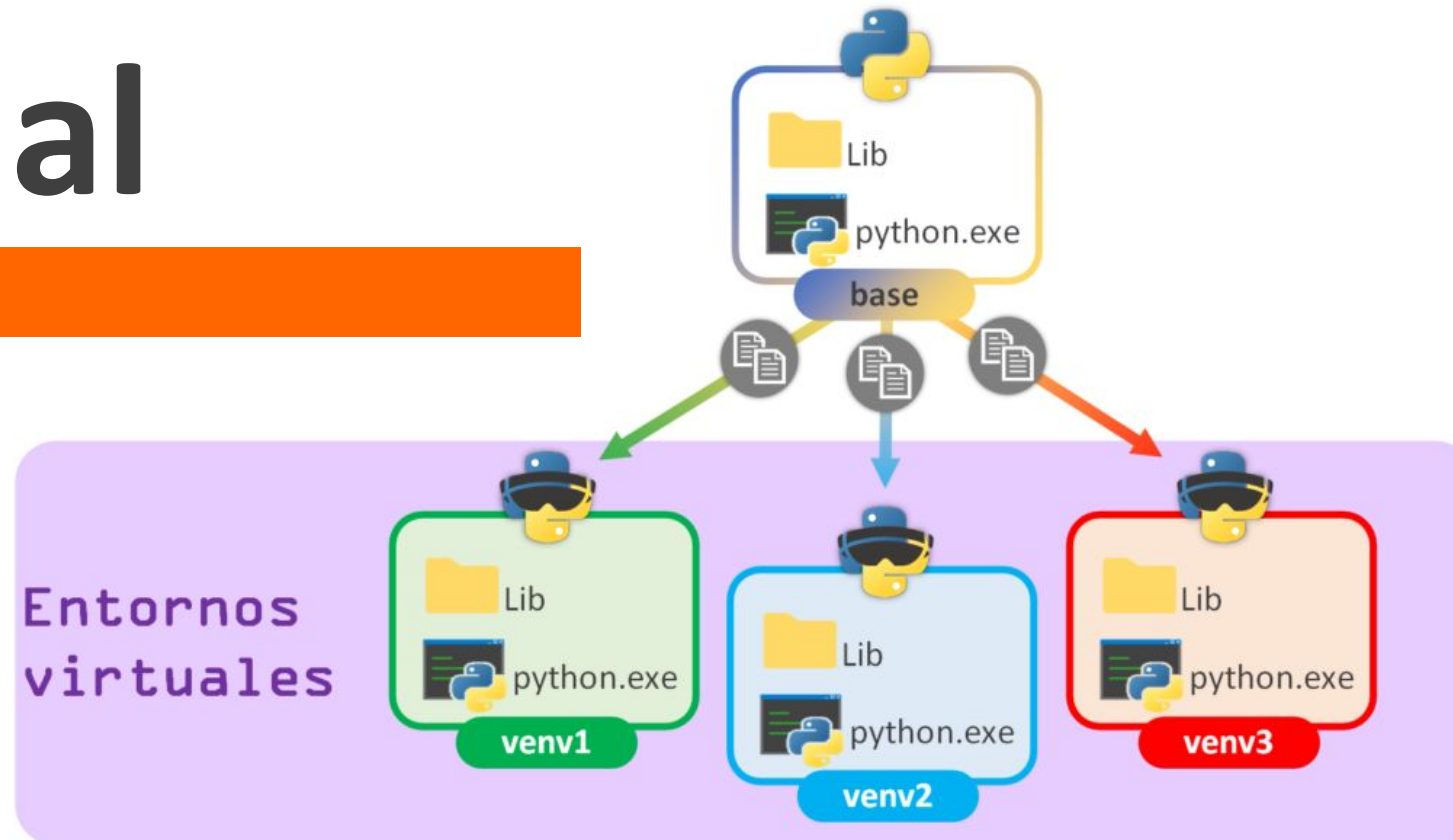
django



@SENAcomunica

www.sena.edu.co

Configurar el entorno virtual



Configurar entorno virtual

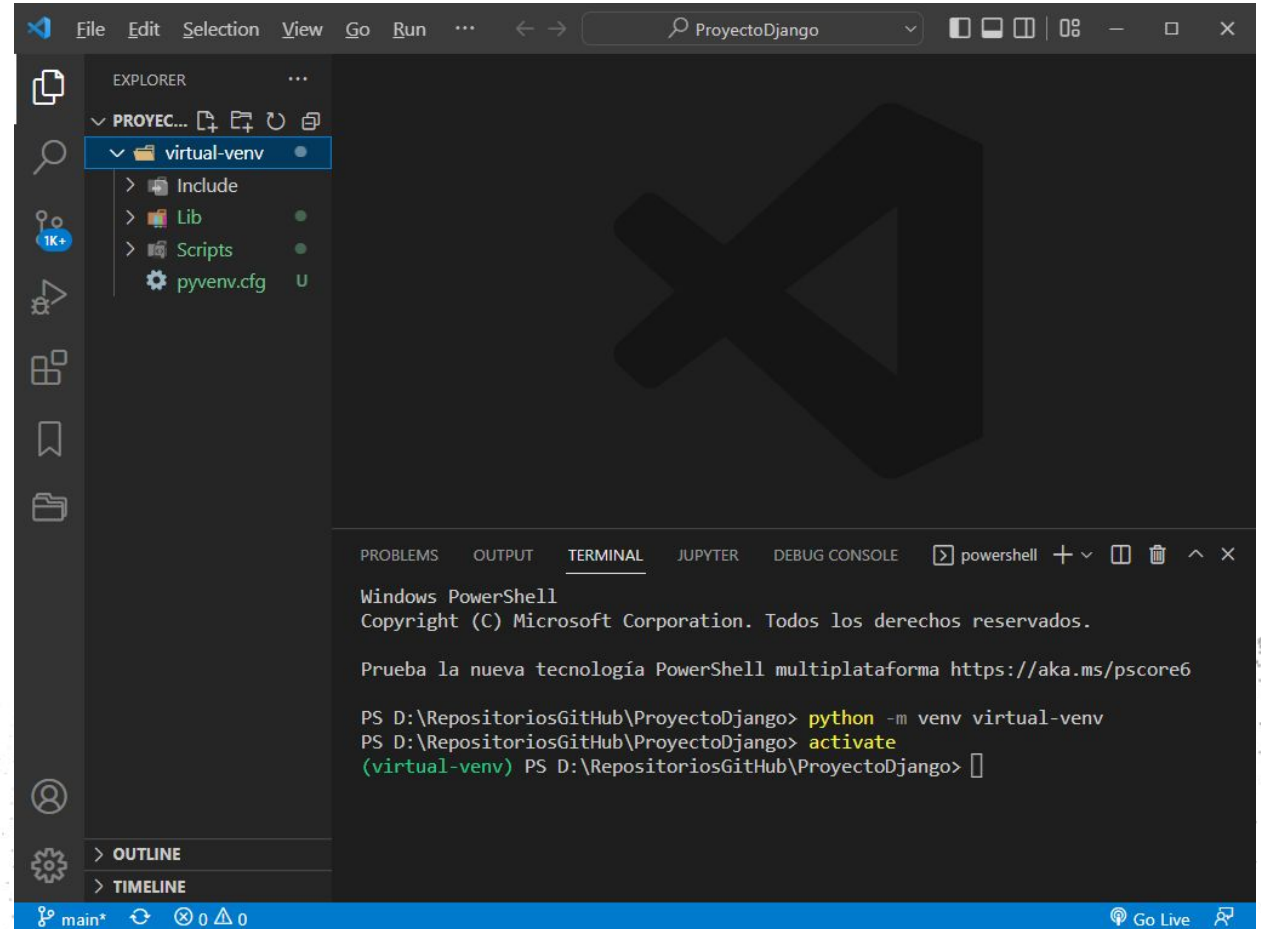
Ctrl + ñ: abrir la terminal

- Crear entorno virtual por primera vez
python -m venv virtual-venv
- Agregar al path de las variables de entorno de script del entorno virtual que se realizó:
 - Abrir la carpeta virtual-venv
 - Clic derecho sobre Scripts
 - Copiar path
 - Editar las variables de entorno de Windows
 - Pegar el path copiado en el path de Windows
 - Aceptar los cambios
- activar el entorno virtual:

activate

Nota: en caso de no tener instaladas las librerías de Django, escriben en la terminal:

pip install django



```
File Edit Selection View Go Run ... < -> ProyectoDjango
EXPLORER
PROYEC...
virtual-venv
  Include
  Lib
  Scripts
  pyvenv.cfg
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE powershell + - X
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\RepositoriosGitHub\ProyectoDjango> python -m venv virtual-venv
PS D:\RepositoriosGitHub\ProyectoDjango> activate
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango>
```

Configurar entorno virtual

Si saca error se van a:

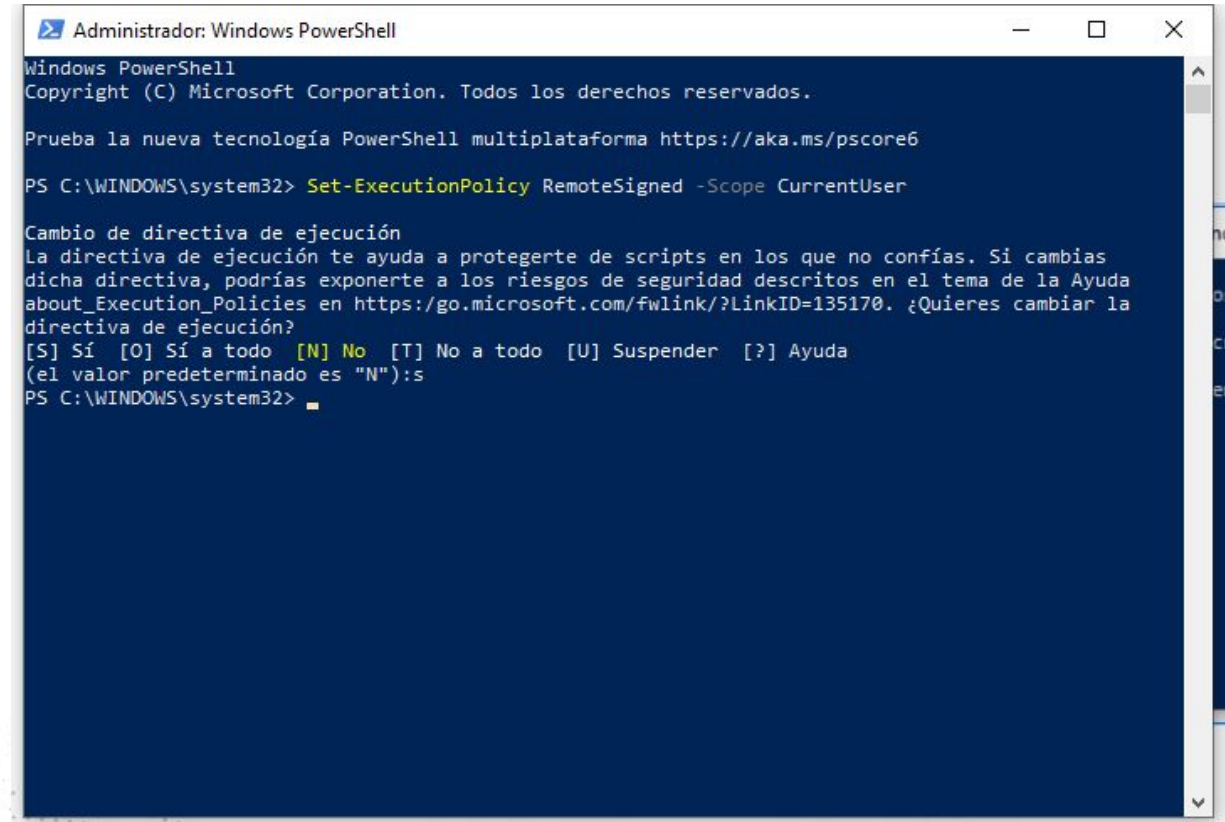
Inicio de Windows->PowserShell->clic derecho ejecutar como administrador
escribir:

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Nota: sin espacio al final
escribir la letra **S** y **enter**

Si no funciona, revisar que se tenga configurada las variables de entorno de python en Windows:

- Variable de Python en PATH variables de entorno de Windows.

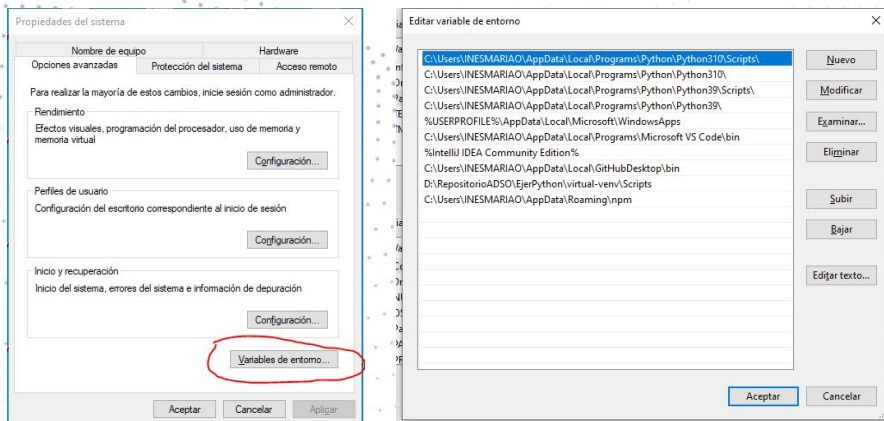


```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

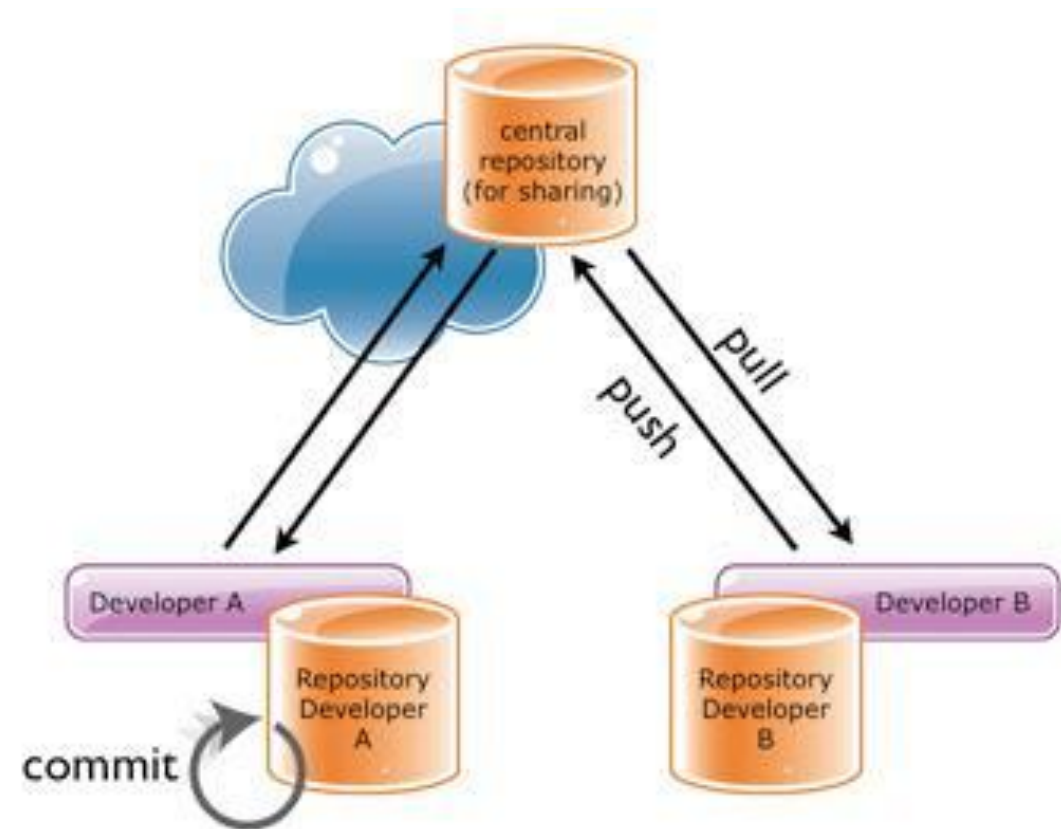
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda
(el valor predeterminado es "N"):s
PS C:\WINDOWS\system32>
```



Crear repositorio en GitHub para alojar el proyecto

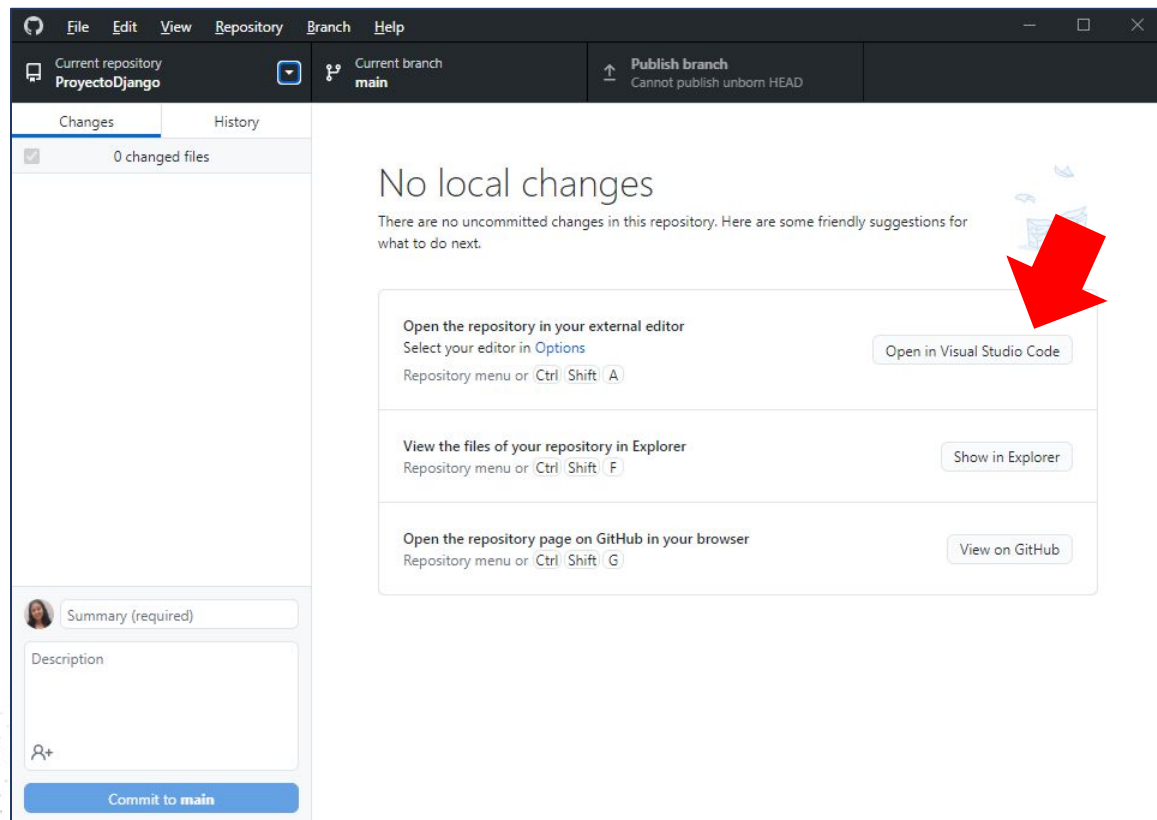
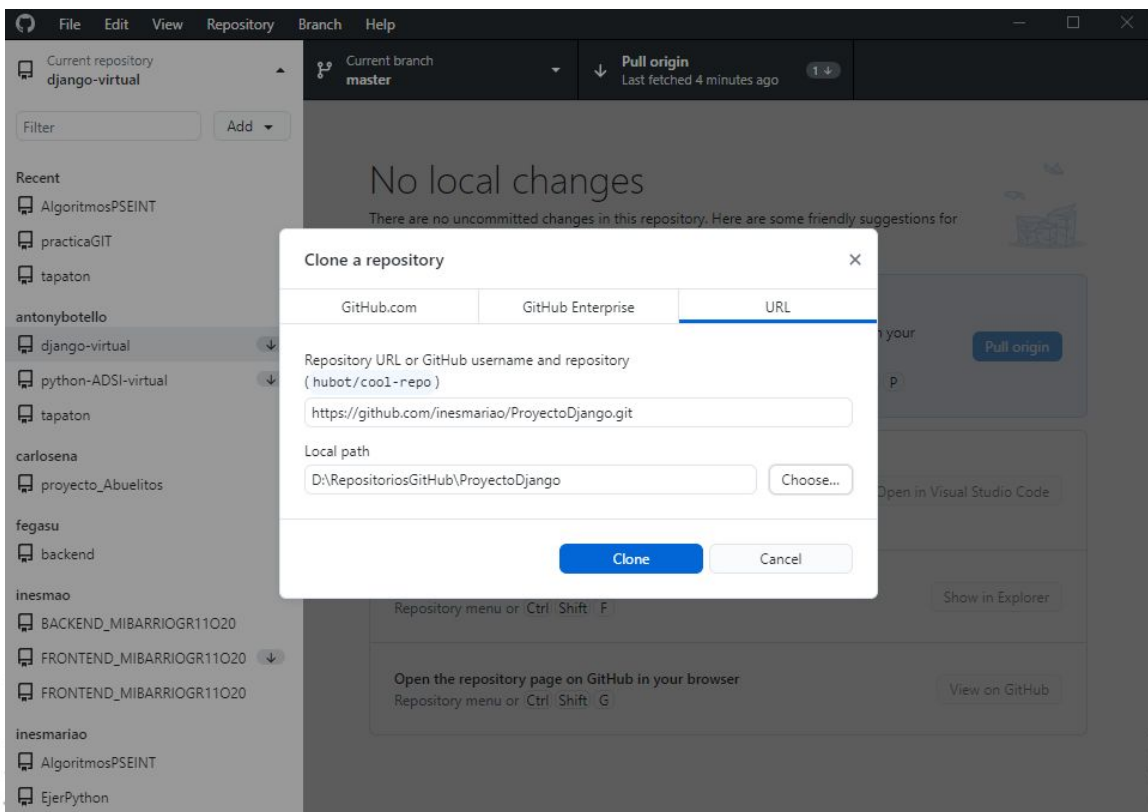


Crear el repositorio del proyecto en GitHub

A screenshot of a GitHub repository page for 'inesmariao / ProyectoDjango'. The repository is marked as 'Public'. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area features a 'Quick setup' section with options to 'Set up in Desktop', 'HTTPS', or 'SSH', and a URL 'https://github.com/inesmariao/ProyectoDjango.git'. Below this, there is a section titled '...or create a new repository on the command line' with a code block containing the following commands:

```
echo "# ProyectoDjango" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/inesmariao/ProyectoDjango.git
git push -u origin main
```

Clonar el repositorio en GitHub Desktop





Configurar el proyecto en Django



Abrir el proyecto en Visual Studio Code

- activar el entorno virtual:

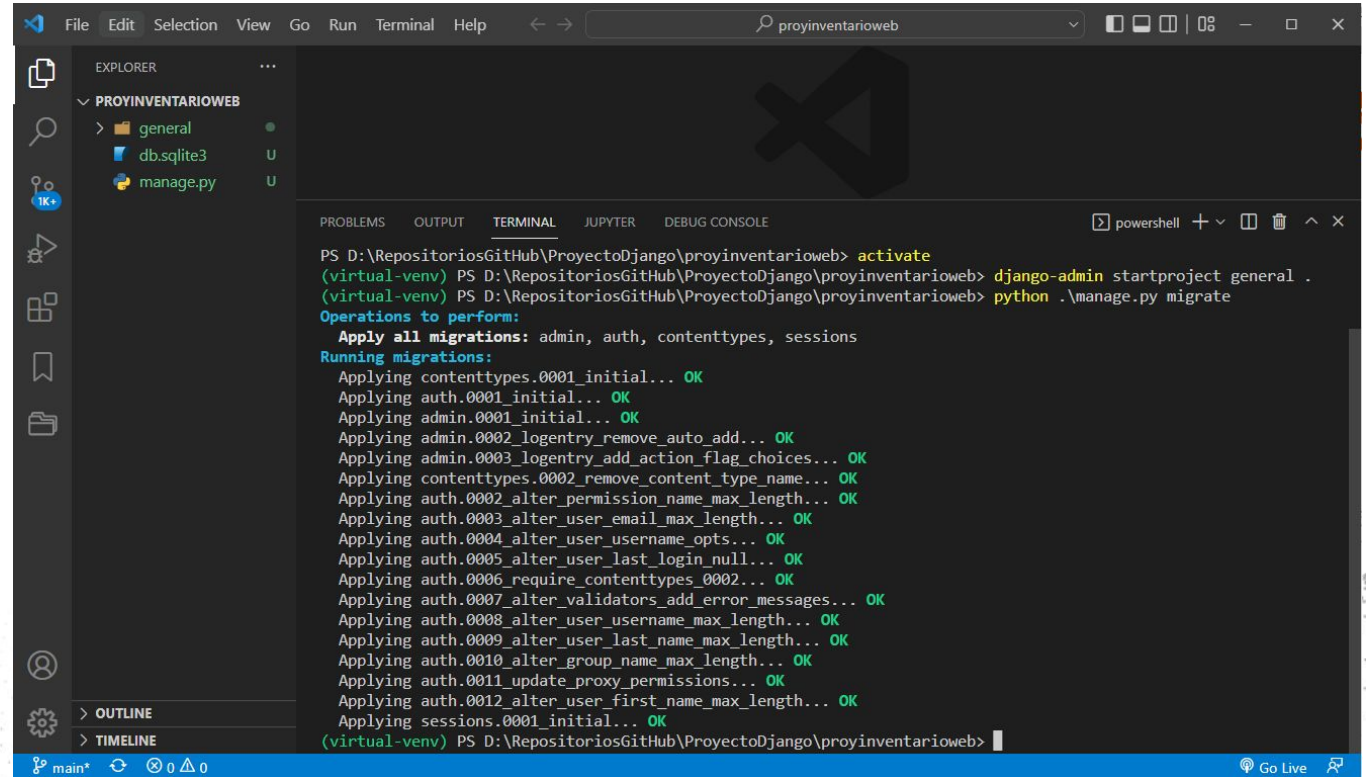
activate

- Iniciar el proyecto con el nombre de la carpeta principal de su proyecto.

django-admin startproject base .

Nota: escribir el punto al final y pueden cambiar **base** por otro nombre sin espacios.ej: (main, general, base, principal)

- Ejecutar el comando: **python .\manage.py migrate** el cual genera la base de datos inicial.



The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure: PROYINVENTARIOWEB, general, db.sqlite3, and manage.py. The Terminal panel at the bottom shows the following commands and output:

```
PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb> activate
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb> django-admin startproject general .
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb> python .\manage.py migrate

Operations to perform:
  Apply all migrations: admin, auth, contenttypes, 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
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb>
```



Iniciar Proyecto

- Iniciar el servidor. `python .\manage.py runserver`

Para abrir el sitio web del proyecto presionan Ctrl y hacen clic en <http://127.0.0.1:8000> que es la URL donde está alojado el proyecto.

```
File Edit Selection View Go Run Terminal Help
proyinventarioweb

EXPLORER
PROYINVENTARIOWEB
  general
  db.sqlite3
  manage.py

TERMINAL
powershell
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
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb> python .\manage.py runserver
>>
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced)
September 30, 2022 - 19:42:05
Django version 4.1.1, using settings 'general.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Estructura de Carpetas del Proyecto

A solid orange horizontal bar is positioned below the word "Proyecto" in the title.

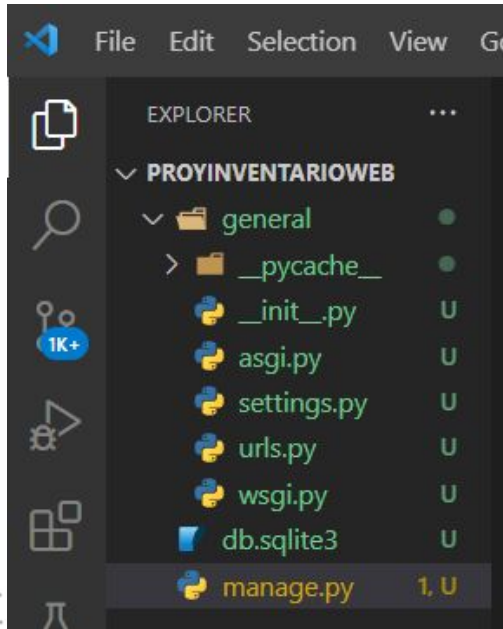
Estructura General de carpetas del proyecto



The screenshot displays the Visual Studio Code interface for a project named 'proyinventarioweb'. The Explorer sidebar on the left shows the project's file structure: a 'general' folder containing '__pycache__', '__init__.py', 'asgi.py', 'settings.py', 'urls.py', 'wsgi.py', 'db.sqlite3', and 'manage.py'. The 'manage.py' file is selected and open in the main editor. The code in 'manage.py' is a Django management utility script. It starts with a shebang line, followed by a docstring, imports for 'os' and 'sys', and a 'main()' function. The 'main()' function sets the default Django settings module to 'general.settings' and attempts to execute from the command line using 'django.core.management'. If an 'ImportError' occurs, it prints a message asking the user to verify Django is installed and the environment is set up. The status bar at the bottom indicates the current line is 14, column 18, with 4 spaces, UTF-8 encoding, CRLF line endings, and the Python interpreter selected.

```
1  #!/usr/bin/env python
2  """Django's command-line utility for administrative tasks."""
3  import os
4  import sys
5
6
7  def main():
8      """Run administrative tasks."""
9      os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'general.settings')
10     try:
11         from django.core.management import execute_from_command_line
12     except ImportError as exc:
13         raise ImportError(
14             "Couldn't import Django. Are you sure it's installed and "
15             "available on your PYTHONPATH environment variable? Did you "
16             "forget to activate a virtual environment?"
17         ) from exc
18     execute_from_command_line(sys.argv)
19
20
21 if __name__ == '__main__':
22     main()
23
```


Estructura General de carpetas y archivos del proyecto



- Carpeta **general**: contiene el núcleo de nuestro proyecto.
- Archivo **manage.py**: administra los comandos especiales de Django, por ejemplo crear usuarios, ejecutar el servidor, etc.
- Archivo **wsgi.py y asgi.py**: es para configuración de servidores y otras funciones, pero no los vamos a modificar.
- Archivo **settings.py**: si lo vamos a modificar y a trabajar, aquí tenemos todas las configuraciones del funcionamiento de nuestro proyecto. Aquí configuramos la ruta de los *templates* de nuestra aplicación, la conexión a la base de datos, validar contraseñas, lenguaje, zona horaria, referencias a las carpetas de estilos CSS, Javascript, imágenes, organización de Apps (módulos del proyecto), etc.
- Archivo **urls.py**: en este archivo se vinculan las vistas con el navegador.

Crear superusuario



python .\manage.py createsuperuser

1. Escribir el usuario
2. Escribir el correo
3. Escribir un password, no van a ver lo que se digita, se pide confirmar el password.

A screenshot of the Visual Studio Code editor interface. The Explorer pane on the left shows a project named 'PROYINVENTARIOWEB' with a 'general' folder containing files like '_pycache_', '_init_.py', 'asgi.py', 'settings.py', 'urls.py' (highlighted), 'wsgi.py', 'db.sqlite3', and 'manage.py'. The main editor window shows the 'urls.py' file with a docstring for 'general URL Configuration'. The bottom panel is split into 'PROBLEMS', 'OUTPUT', 'TERMINAL', 'JUPYTER', and 'DEBUG CONSOLE'. The 'TERMINAL' tab is active, showing a PowerShell session where the command 'python .\manage.py createsuperuser' has been executed. The terminal output shows the prompts for username, email, and password, with error messages indicating the password is too similar to the email, too short, and too common. The user has chosen to bypass validation, and the superuser was created successfully.

```
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb> python .\manage.py createsuperuser
Username (leave blank to use 'inesmariao'): inesmaoh
Email address: inesmao@misena.edu.co
Password:
Password (again):
The password is too similar to the email address.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: n
Password:
Password (again):
Superuser created successfully.
(virtual-venv) PS D:\RepositoriosGitHub\ProyectoDjango\proyinventarioweb>
```

Crear carpeta Static

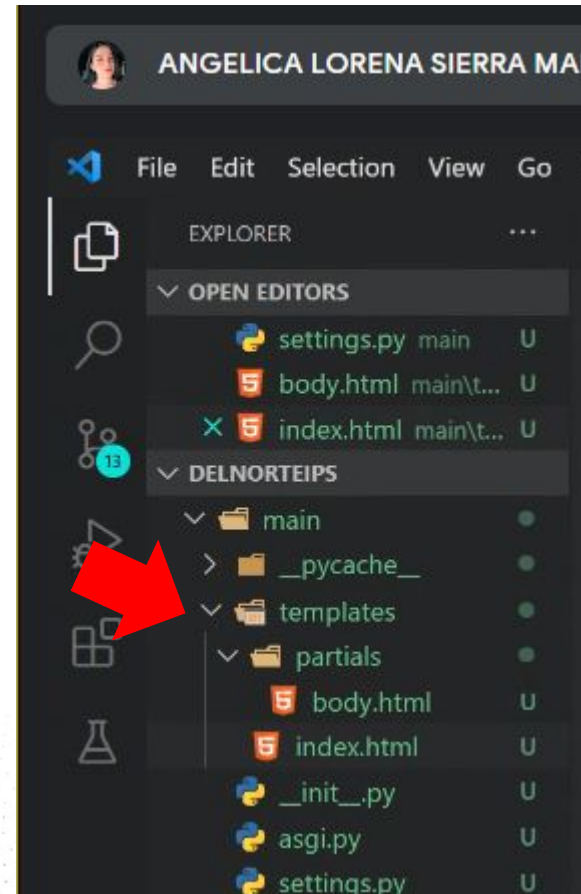
1. Crear una nueva carpeta llamada **static** por fuera de la carpeta **main**.
2. Abrir archivo settings.py
3. Borrar la línea 118 y escribir el siguiente código:

```
STATIC_URL = '/static/'  
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, "static"),  
]  
STATIC_ROOT= "/static"
```

1. Importar la librería **os** clic en el bombillito para importarla, ó escribir en la línea 13 **import os**
2. Línea 106: cambiar `LANGUAGE_CODE = 'es'`
3. `TIME_ZONE = 'America/Bogota'`

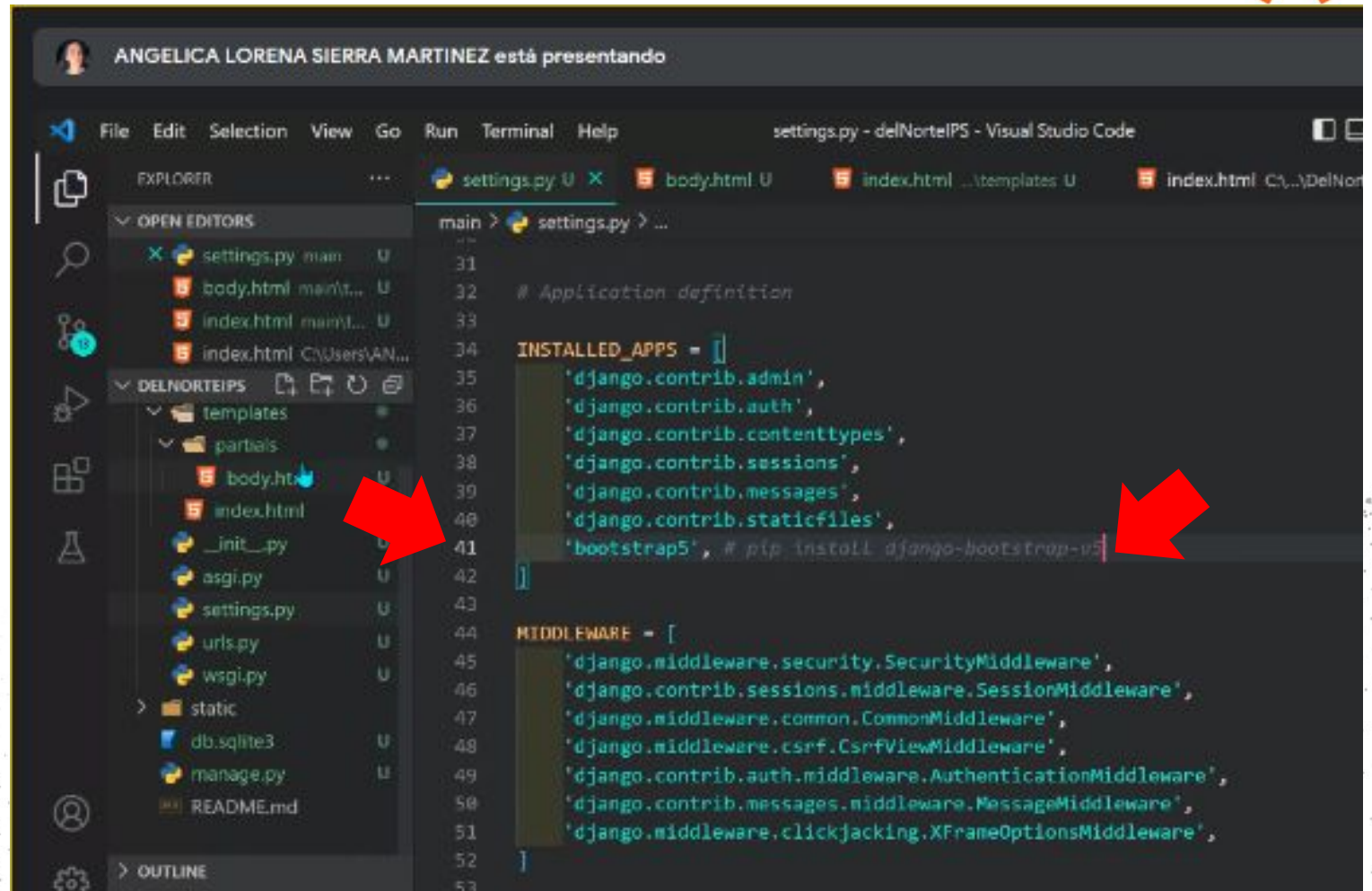
Carpeta Templates

1. Crear una nueva carpeta llamada **partial**.
2. Entrar a la nueva carpeta **partial** y crear un nuevo archivo llamado **body.html**
3. En la carpeta **templates** crear un nuevo archivo llamado **index.html**



Instalar Bootstrap

1. En la terminal escribir:
pip install django-bootstrap-v5
1. Abrir la carpeta **settings.py** y después de la línea 40 escribir el comando que indica que vamos a trabajar con Bootstrap.
'bootstrap5', # pip pip install bootstrap-v5

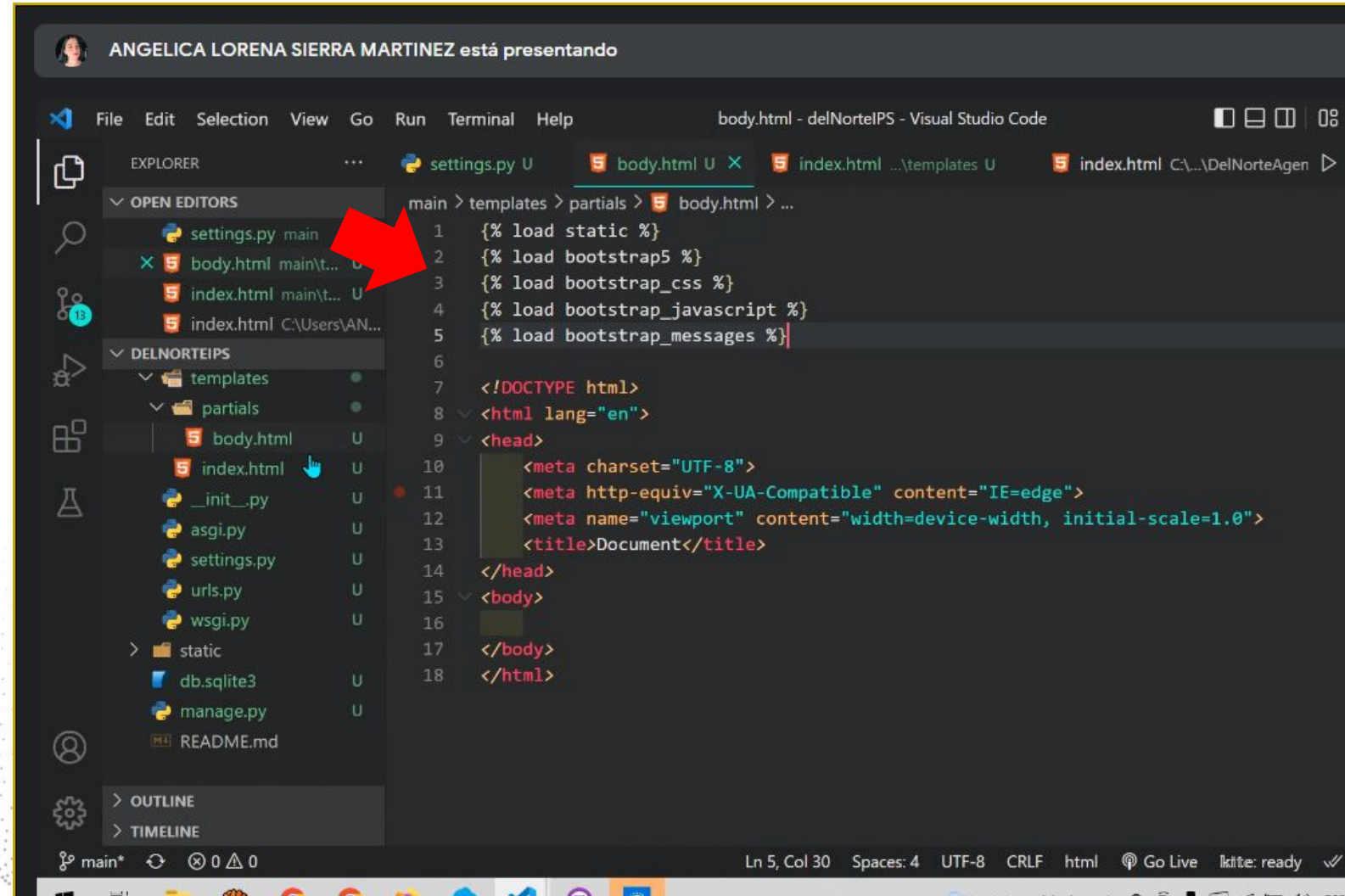


```
main > settings.py > ...
31
32 # Application definition
33
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'bootstrap5', # pip pip install bootstrap-v5
42 ]
43
44 MIDDLEWARE = [
45     'django.middleware.security.SecurityMiddleware',
46     'django.contrib.sessions.middleware.SessionMiddleware',
47     'django.middleware.common.CommonMiddleware',
48     'django.middleware.csrf.CsrfViewMiddleware',
49     'django.contrib.auth.middleware.AuthenticationMiddleware',
50     'django.contrib.messages.middleware.MessageMiddleware',
51     'django.middleware.clickjacking.XFrameOptionsMiddleware',
52 ]
53
```

Configurar la carga de Bootstrap, CSS y Javascript

1. En el archivo **body.html** insertar los siguientes comandos, desde la línea #1:

- {% load static %}
- {% load bootstrap5 %}
- {% bootstrap_css %}
- {% bootstrap_javascript %}
- {% bootstrap_messages %}



The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with a red arrow pointing to the **body.html** file in the **partials** directory. The code editor shows the content of **body.html** with the following code:

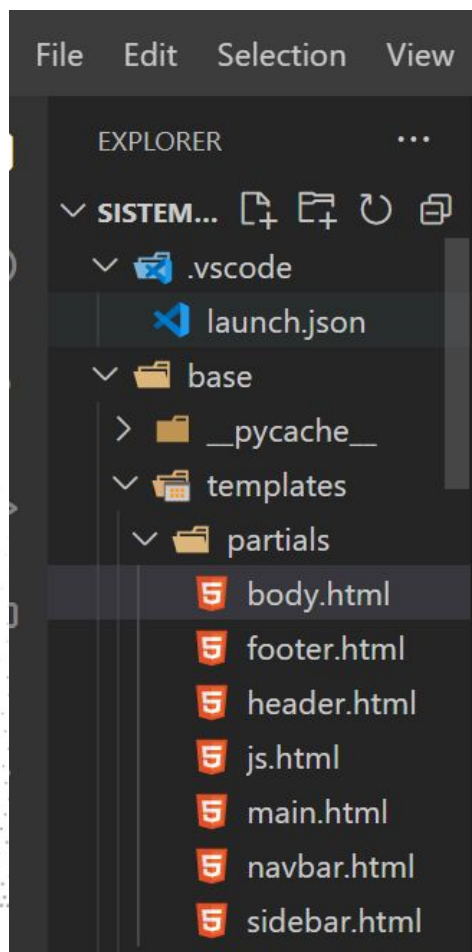
```
1 {% load static %}
2 {% load bootstrap5 %}
3 {% load bootstrap_css %}
4 {% load bootstrap_javascript %}
5 {% load bootstrap_messages %}
6
7 <!DOCTYPE html>
8 <html lang="en">
9 <head>
10 <meta charset="UTF-8">
11 <meta http-equiv="X-UA-Compatible" content="IE=edge">
12 <meta name="viewport" content="width=device-width, initial-scale=1.0">
13 <title>Document</title>
14 </head>
15 <body>
16
17 </body>
18 </html>
```

The status bar at the bottom indicates the current position is Ln 5, Col 30, with 4 spaces, UTF-8 encoding, CRLF line endings, and the file type is html.

Configurar los archivos .html del proyecto



Se deben crear los archivos html del proyecto organizados de acuerdo a cada parte del sitio web, dentro de la carpeta partial.



Módulos del proyecto





Crear los módulos del proyecto

En la terminal escribir:

```
python .\manage.py startapp usuarios
```

```
python .\manage.py startapp facturas
```

```
python .\manage.py startapp productos
```

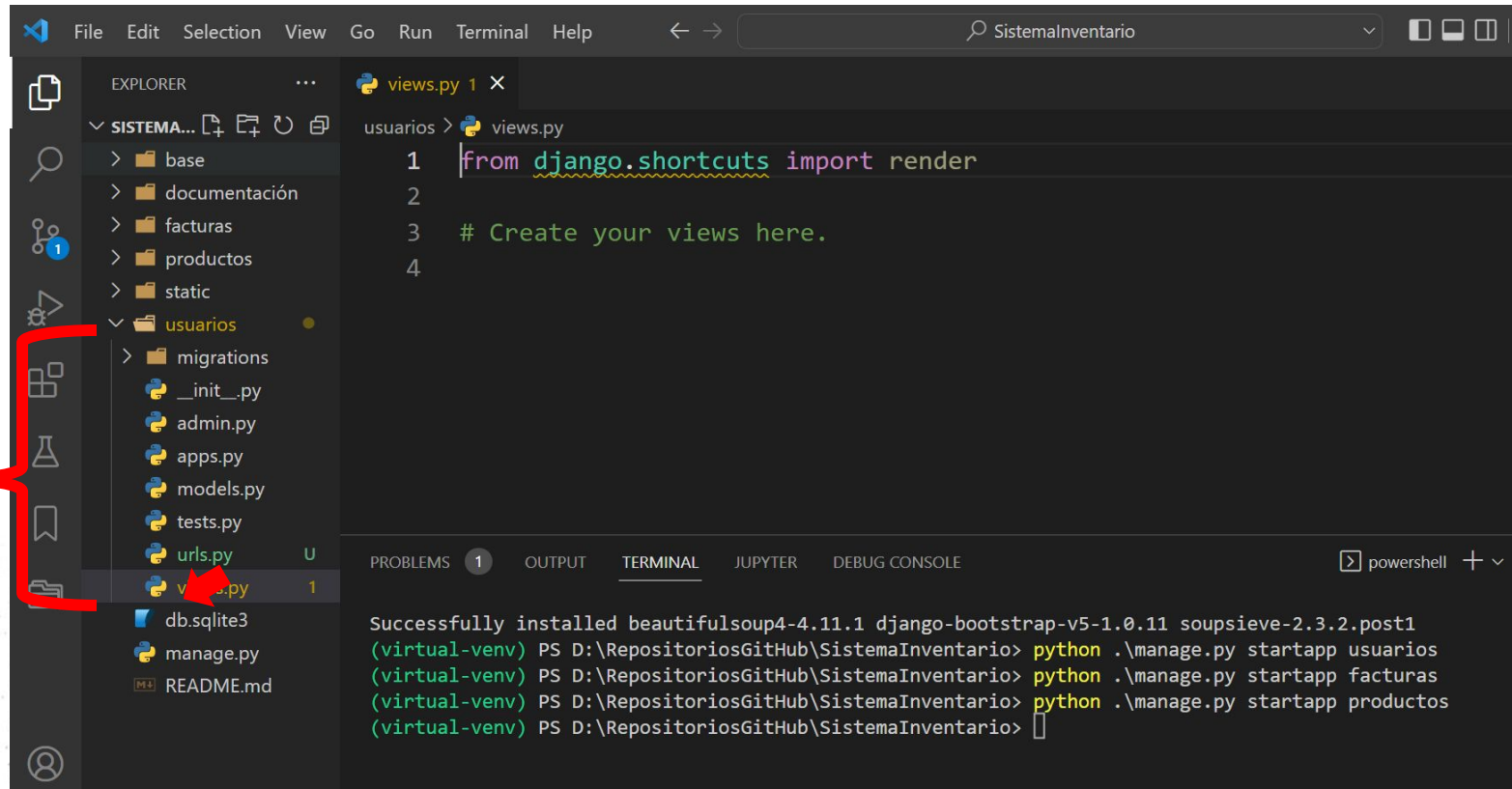
Nota: las palabras en azul son los nombres de los módulos, se deben crear los de sus proyectos.

Vamos a trabajar en los archivos:

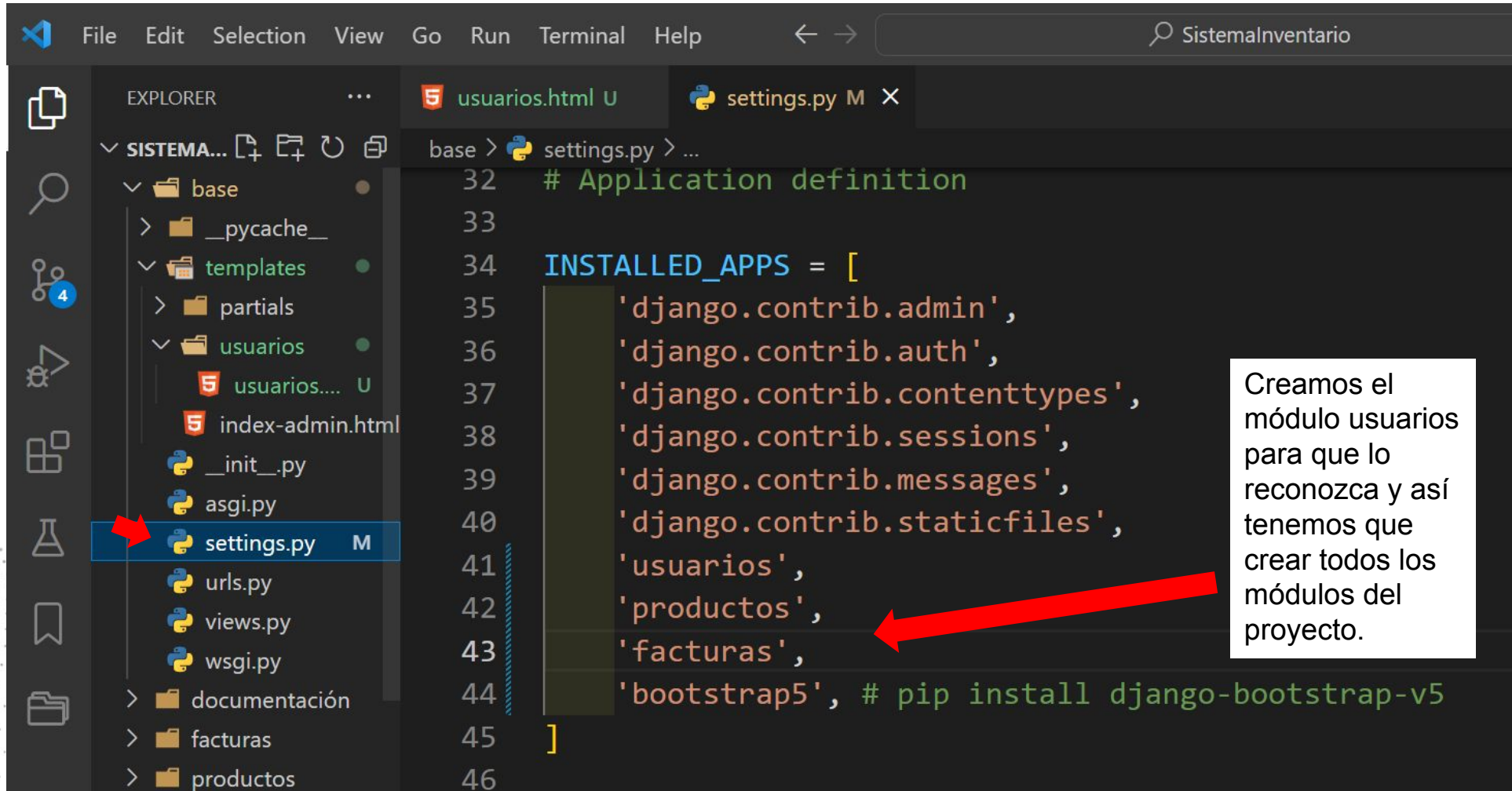
- **models.py**
- **views.py**

Creamos un nuevo archivo dentro de cada módulo llamado **urls.py**

Esta es la estructura de cada módulo creado



Configurar en el archivo `settings.py` del proyecto los módulos creados



The screenshot shows the Visual Studio Code editor with the `settings.py` file open. The `INSTALLED_APPS` list is visible, and the `'usuarios'` module has been added to the list. A red arrow points to the `'usuarios'` entry in the list. A text box explains that the `'usuarios'` module is added so the project recognizes it and all its modules.

```
32 # Application definition
33
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'usuarios',
42     'productos',
43     'facturas',
44     'bootstrap5', # pip install django-bootstrap-v5
45 ]
46
```

Creemos el módulo usuarios para que lo reconozca y así tenemos que crear todos los módulos del proyecto.

Configurar el archivo `urls.py` del módulo `usuarios`

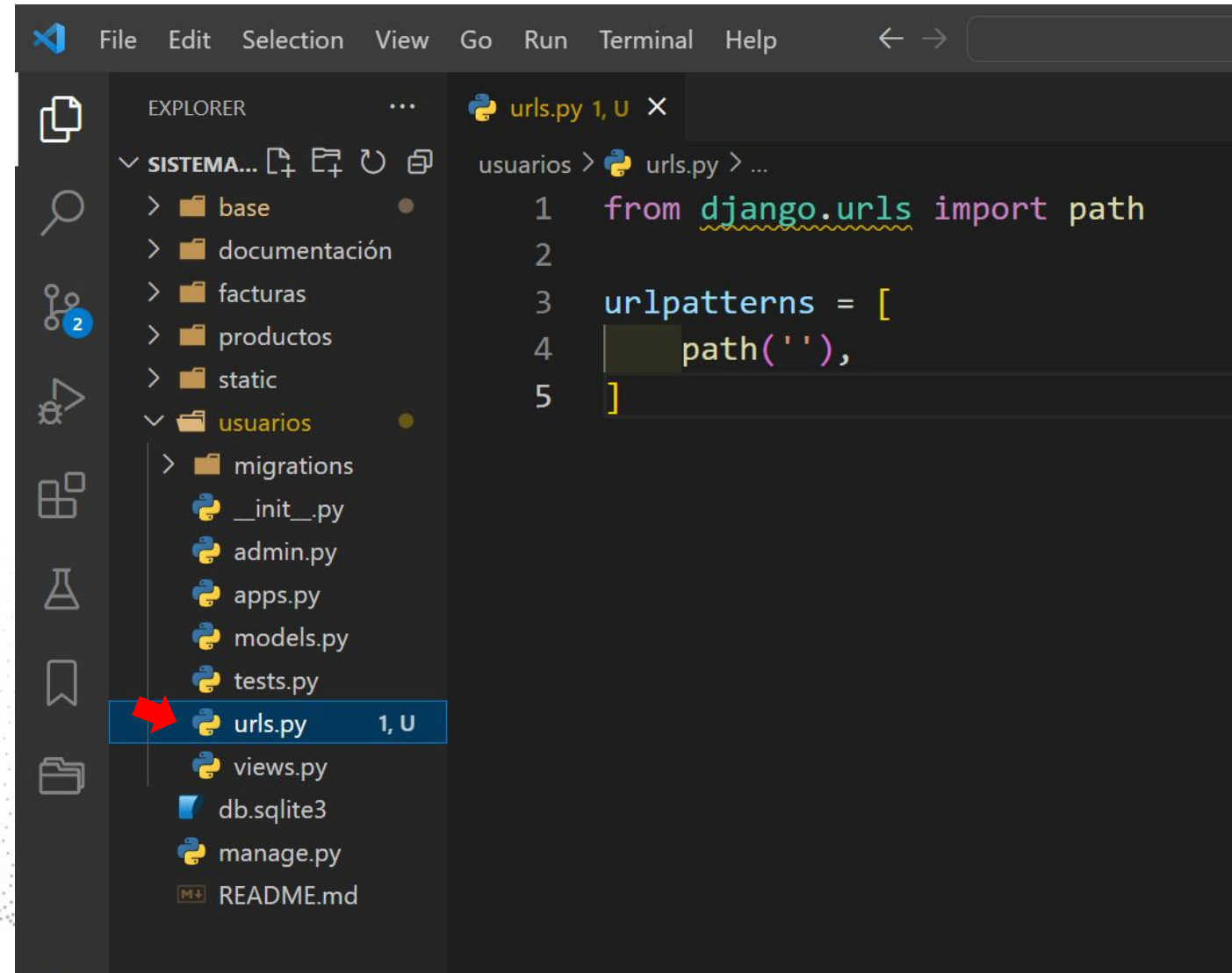
Importar el path de urls:

from django.urls import path

```
from django.urls import path

urlpatterns = [
    path(''),
]
```

Inicialmente se realiza vacío
mientras se configura el archivo
views.py del módulo de `usuarios`



Configurar el `views.py` del módulo `usuarios`

Se crean las vistas de los módulos del proyecto:

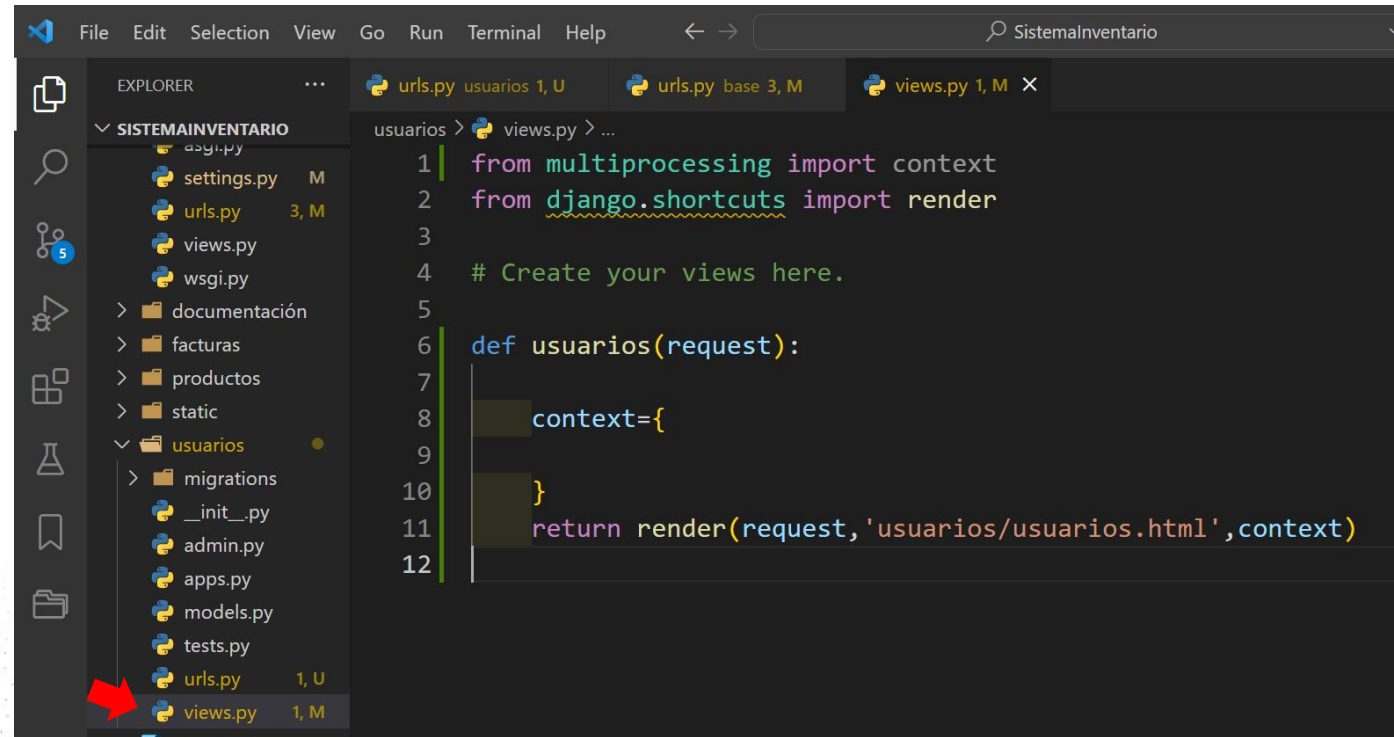
```
from multiprocessing import context
from django.shortcuts import render

# Create your views here.

def usuarios(request):

    context={

    }
    return
    render(request,'usuarios/usuarios.html',context)
```

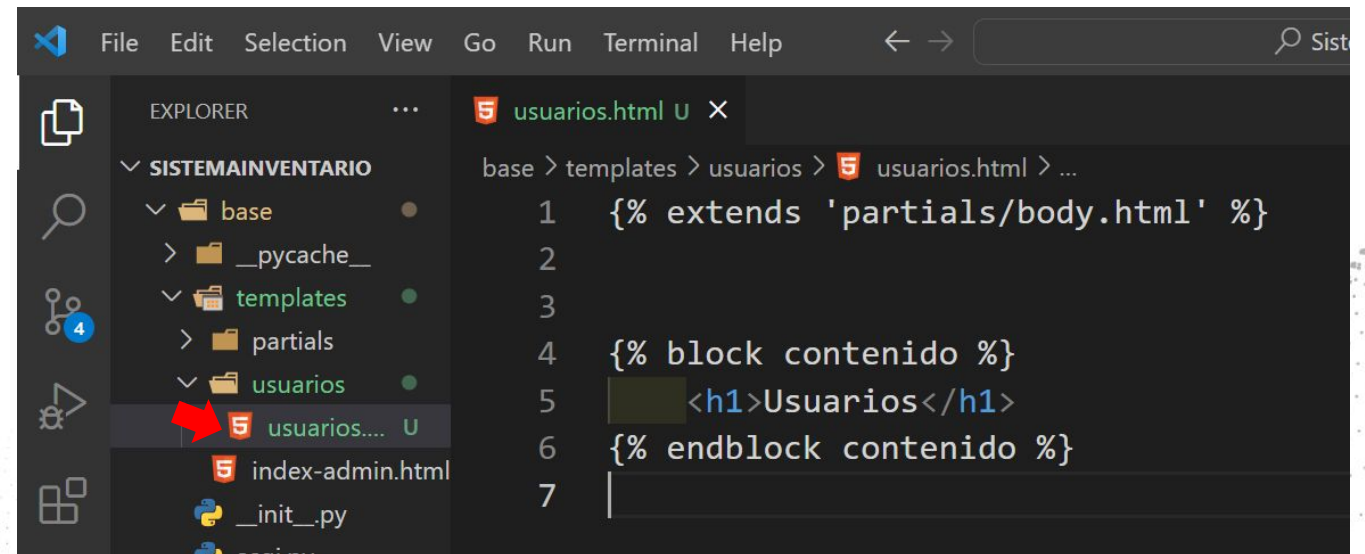


Crear la carpeta para usuarios en la carpeta templates del proyecto

- Se crea una nueva carpeta dentro de la carpeta **templates** llamada **usuarios**.
- Dentro de la carpeta **usuarios**, se crea un archivo llamado **usuarios.html**
- En el archivo **usuarios.html** se usa la base que ya tenemos de la vista, con **extends**.

```
{% extends 'partials/body.html' %}

{% block contenido %}
    <h1>Usuarios</h1>
{% endblock contenido %}
```

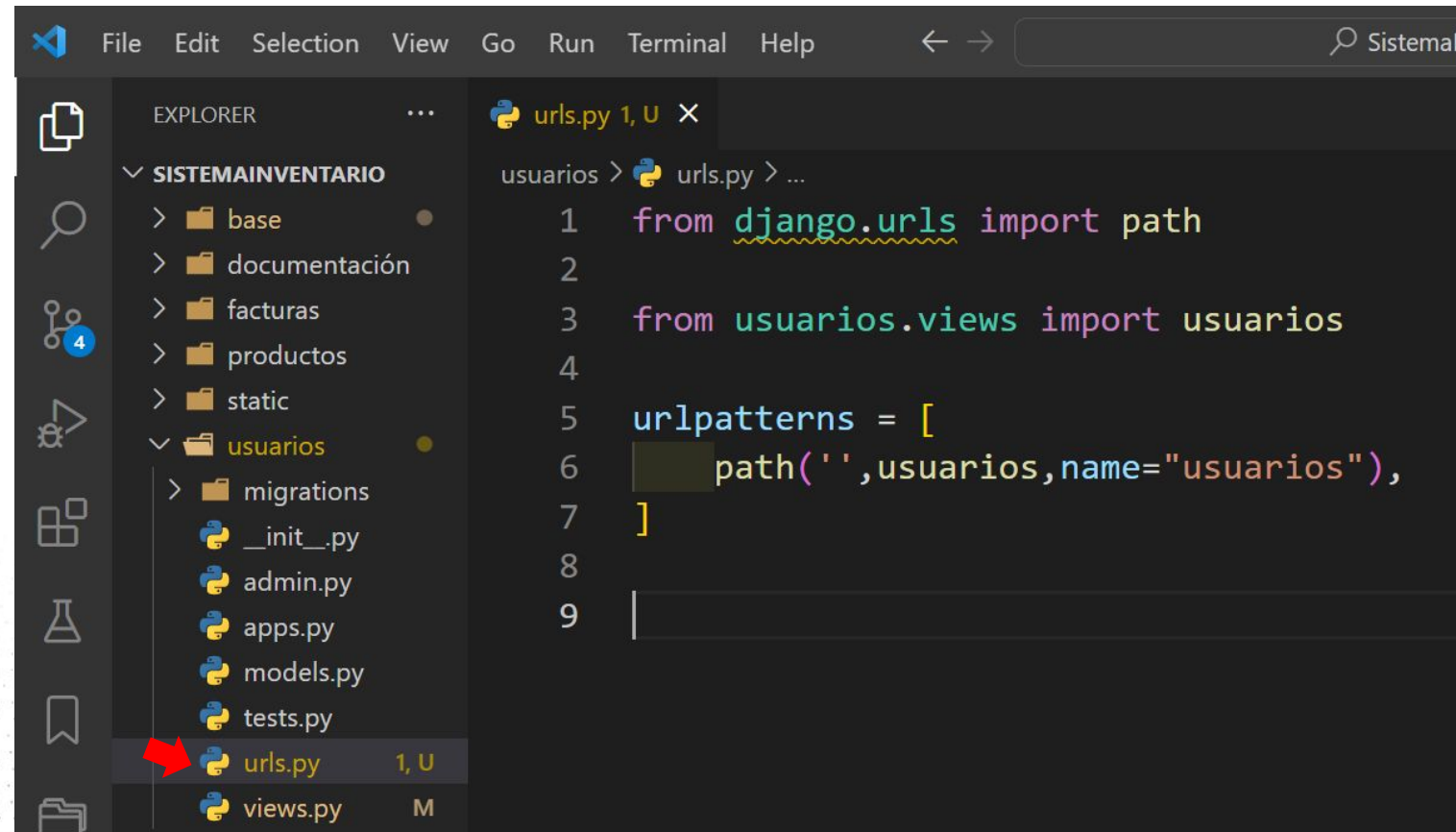


Asociar en el archivo `urls.py` del módulo `usuarios`

```
from django.urls import path

from usuarios.views import usuarios

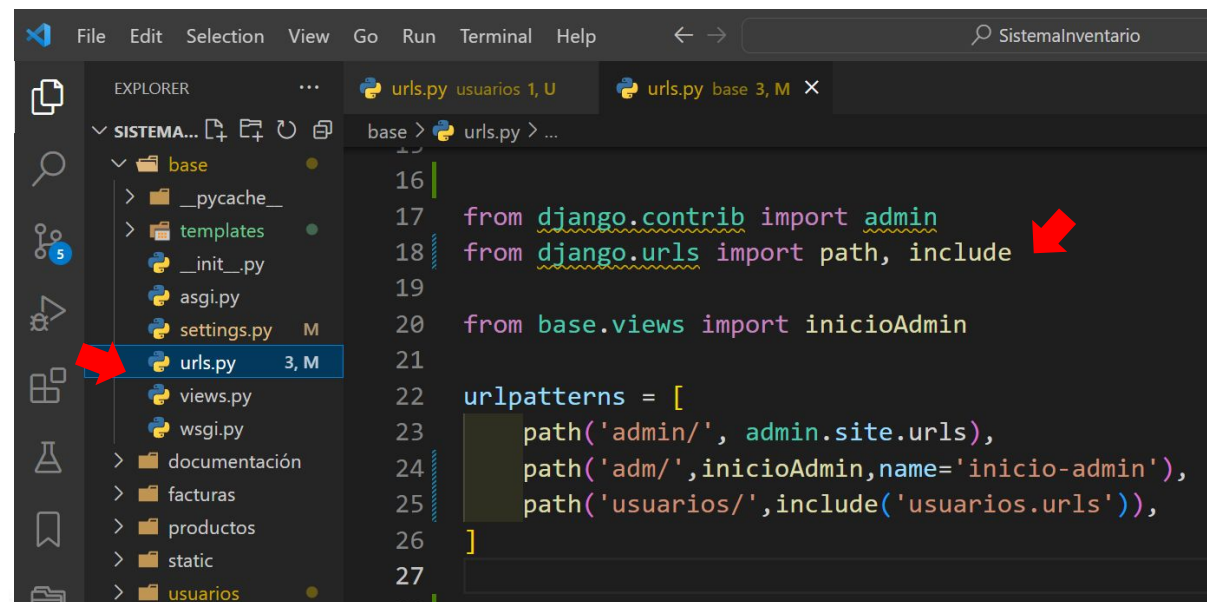
urlpatterns = [
    path('', usuarios, name="usuarios"),
]
```



Incluir el path del módulo `usuarios` en el archivo `urls.py` del proyecto



```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('adm/', inicioAdmin, name='inicio-admin'),  
    path('usuarios/', include('usuarios.urls')),  
]
```



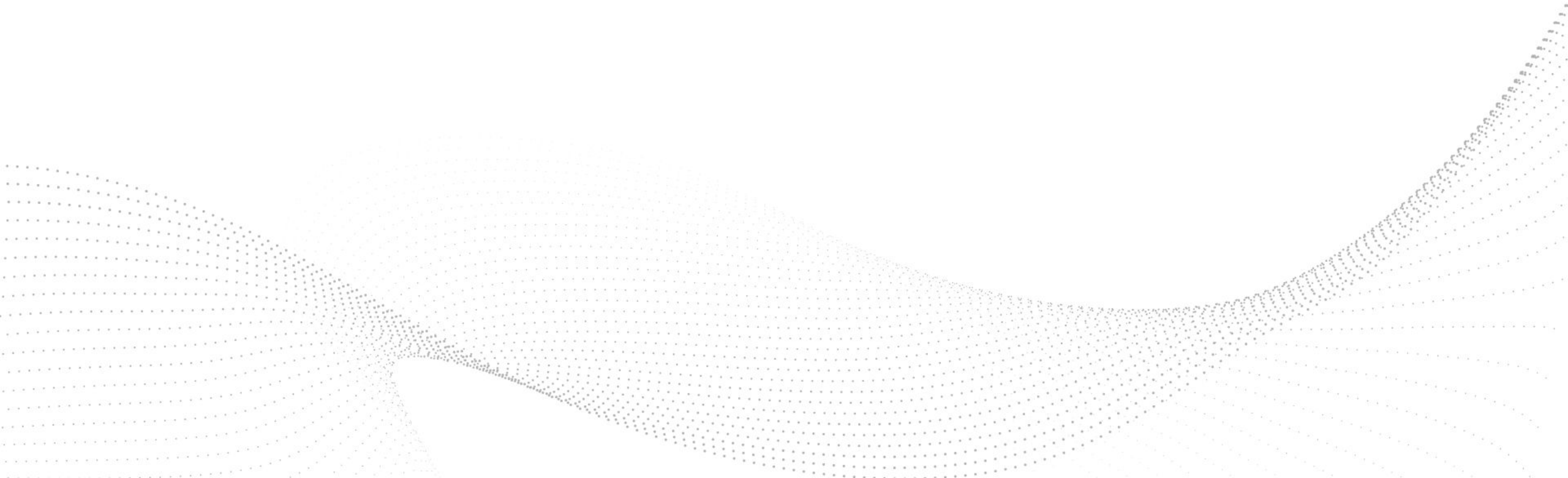
Incluir el load static a todas las páginas html que usen static

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named "SISTEMAINVENTARIO" with a file tree containing ".vscode", "base", "__pycache__", "templates", and "partials". The "body.html" file is selected in the "partials" folder. The main editor window shows the content of "body.html" with line numbers 1 through 12. A red arrow points to line 1, which contains the code "{% load static %}". The breadcrumb at the top of the editor indicates the path: "base > templates > partials > body.html > html".

```
base > templates > partials > body.html > html
1 {% load static %}
2 {% load bootstrap5 %}
3 {% bootstrap_css %}
4 {% bootstrap_javascript %}
5 {% bootstrap_messages %}
6
7
8 <!DOCTYPE html>
9 <html lang="es">
10
11 <head>
12 <meta charset="UTF-8">
```


Configurar el entorno

A solid orange horizontal bar is positioned below the title "Configurar el entorno".





Comandos rápidos de teclado

- Abrir terminal: **Ctrl + ñ**
- Activar portapapeles de Windows: **Tecla Windows + V**

Título

Textos en Calibri, en gris oscuro. Puede reemplazar la imagen por una fotografía alusiva al tema que se está tratando. Al reemplazar la foto tenga cuidado de colocarlo debajo del logo del SENA, que debe ir en la esquina superior derecha en blanco.



Título



50%

Texto corto descriptivo
a 2 o 3 líneas



GRACIAS

www.sena.edu.co

Si su presentación requiere incluir el logo de una marca externa, deberá ubicarlo cómo se indica en las siguientes diapositivas:



Título Presentación

Marca externa



El tamaño del logo de la marca externa no debe superar el tamaño del logo SENA.

Marca externa

Título

Textos en Calibri, en gris oscuro. Puede reemplazar la imagen por una fotografía alusiva al tema que se está tratando. Al reemplazar la foto tenga cuidado de colocarlo debajo del logo del SENA, que debe ir en la esquina superior derecha en blanco.



Marca externa

El tamaño del logo de la marca externa no debe superar el tamaño del logo SENA.

Marca externa



Para las diapositivas con fondo de color naranja, es recomendable colocar la versión blanca del logo de la marca externa.

Marca externa