



Uleam
UNIVERSIDAD LAICA
ELOY ALFARO DE MANABÍ
EXTENSIÓN EL CARMEN

CARRERA

Tecnología de la Información

MATERIA

Sistemas Distribuidos

TEMA

Django con Aplicación Web con
Base de Datos y Envío de Correos

FECHA

22/06/2024

ESTUDIANTE

Oliver Alexander Chiriboga Mero

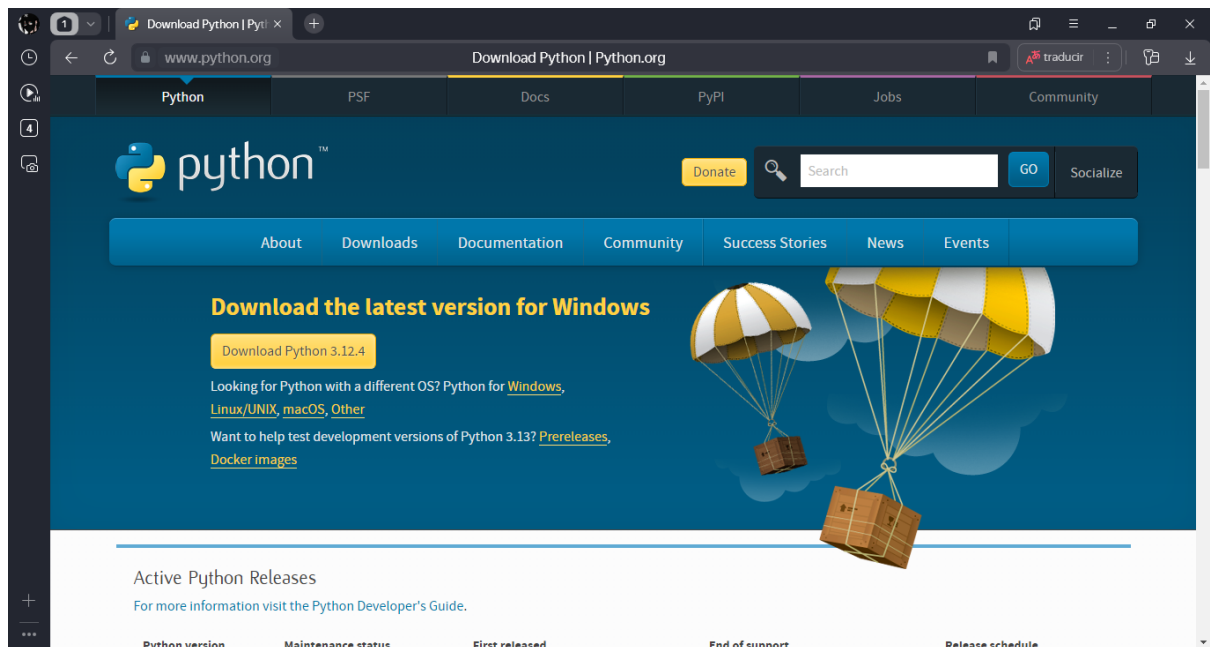
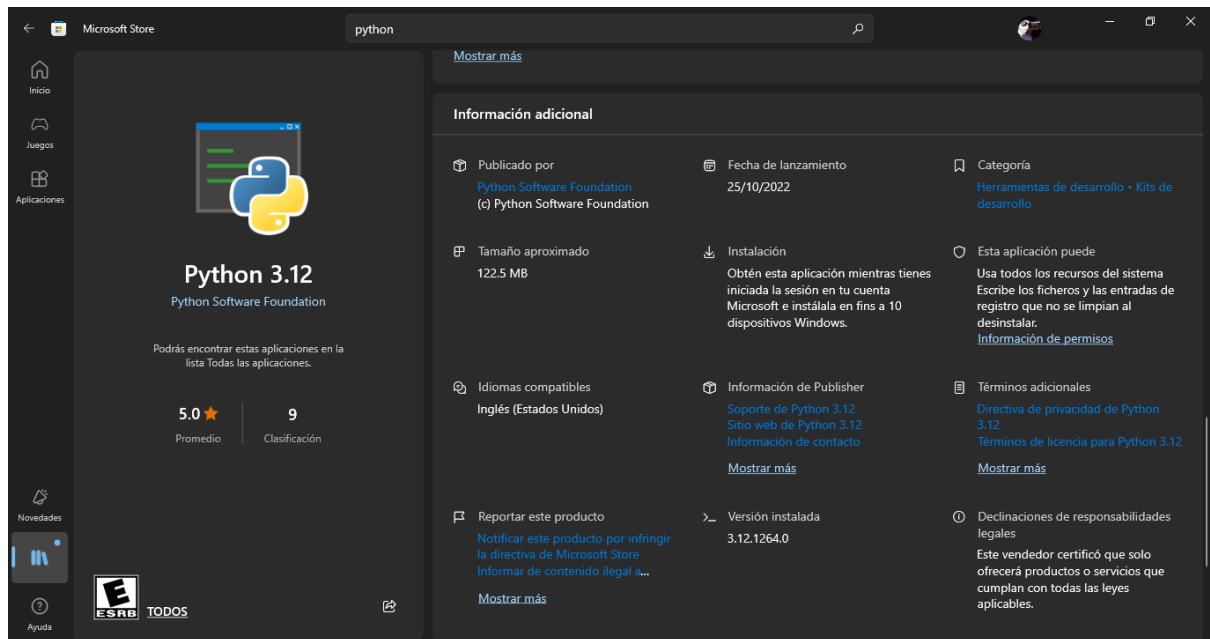
PARALELO

A

DOCENTE

Ing. Cesar Augusto Sinchiguano Chiriboga

Instalacion de Python



Primero se instala Python en nuestro computador, existen dos maneras, una es por medio de la tienda de Microsoft y la segunda es por medio del sitio web de Python el cual pueden acceder con el siguiente enlace: <https://www.python.org/downloads/>

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.4529]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Alexander>python --version
Python 3.12.4

C:\Users\Alexander>
```

Una vez que hayan instalado Python abren la consola de cmd y verifican que se haya instalado correctamente, usando el siguiente comando: **python --version**

Instalación de Django

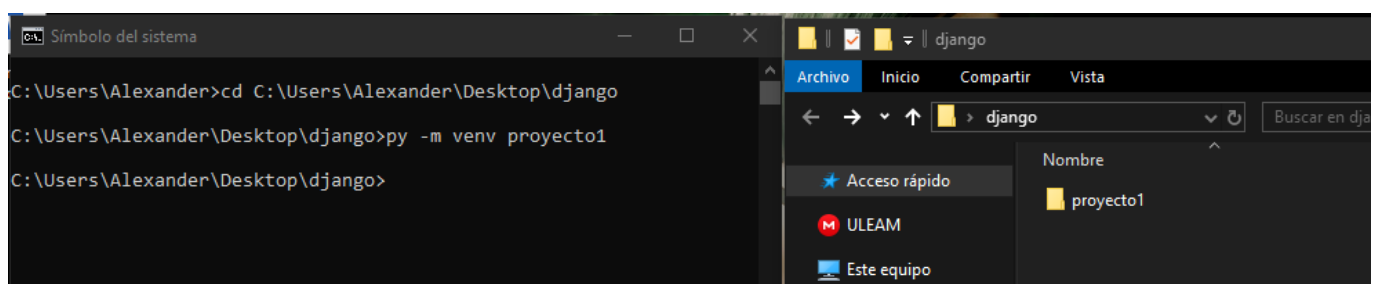


Creamos una carpeta en el escritorio con el nombre de **django**, luego abrimos el cmd y colocamos el comando: **cd** y damos un espacio, arrastramos la carpeta al cmd para obtener la ruta y presionamos enter.

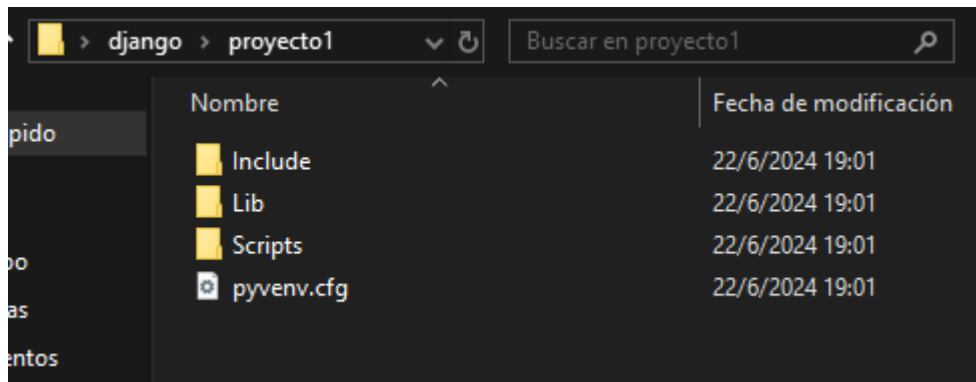
```
Símbolo del sistema

C:\Users\Alexander>cd C:\Users\Alexander\Desktop\django
C:\Users\Alexander\Desktop\django>_
```

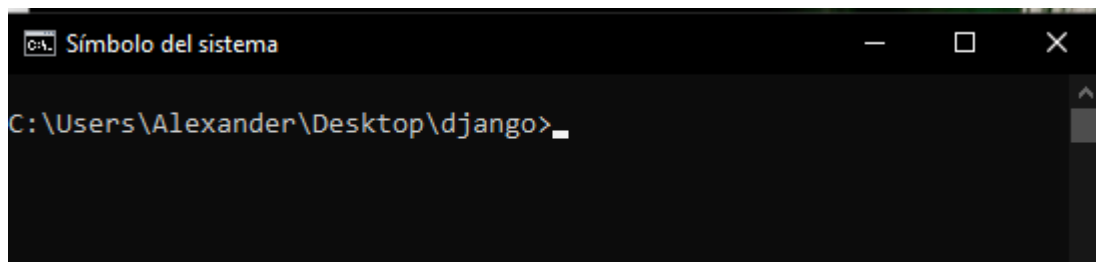
Una vez dentro, escribimos el siguiente comando: **py -m venv proyecto1** y presionamos enter.



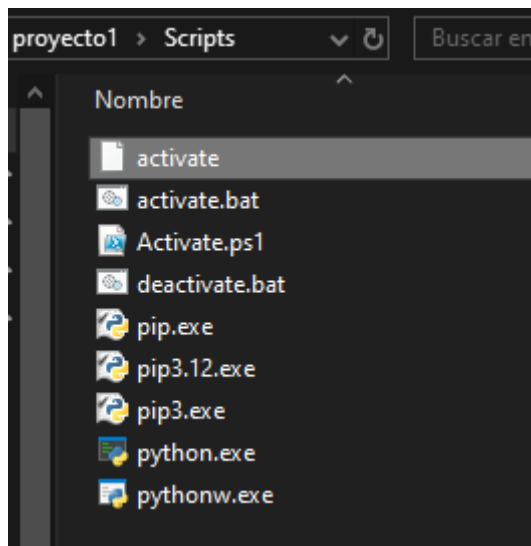
Si todo salió bien, dentro de la carpeta **proyecto1** se deben haber creado los siguientes archivos.



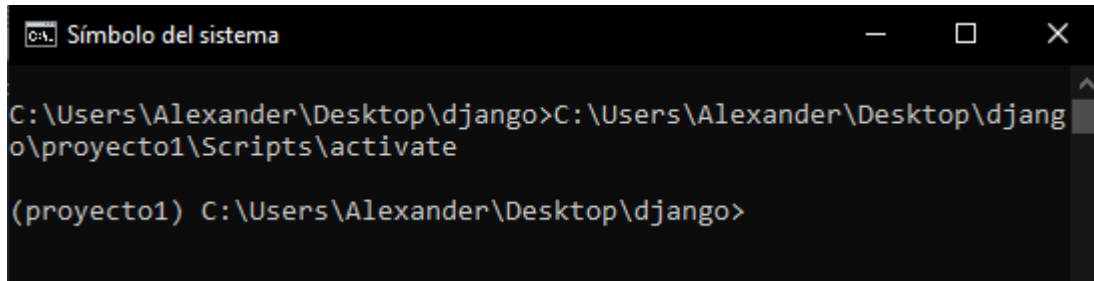
Ahora en la consola de cmd, escribimos el comando: **cls** para limpiarla.



Nos dirigimos a la carpeta **Scripts** y dentro de esta vamos arrastrar el archivo que dice **actíivate** a la consola de cmd y presionamos enter.

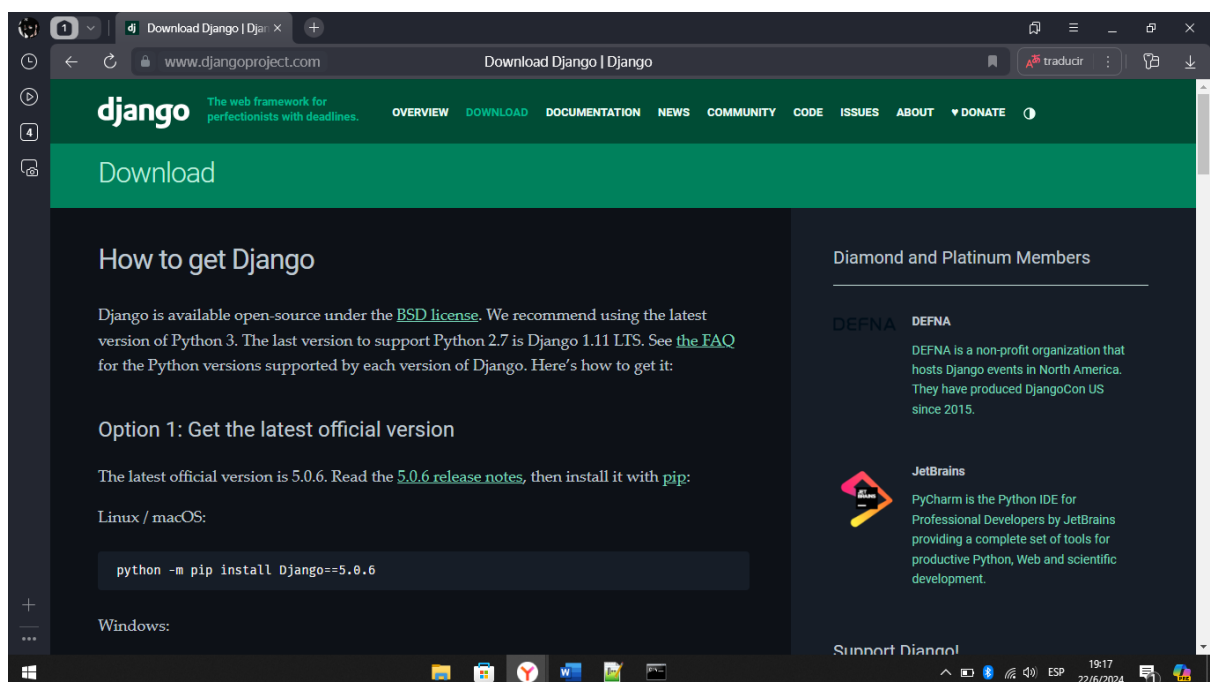


Si todo salió bien, se debe mostrar entre paréntesis (**proyecto1**) eso significa que se activó correctamente.



```
C:\Users\Alexander\Desktop\django>C:\Users\Alexander\Desktop\django\proyecto1\Scripts\activate

(proyecto1) C:\Users\Alexander\Desktop\django>
```



Ahora nos dirigimos a la pagina oficial de Django por medio del siguiente enlace: <https://www.djangoproject.com/download/> y vamos a copiar el siguiente comando para Windows.

Option 1: Get the latest official version

The latest official version is 5.0.6. Read the [5.0.6 release notes](#), then install it with [pip](#):

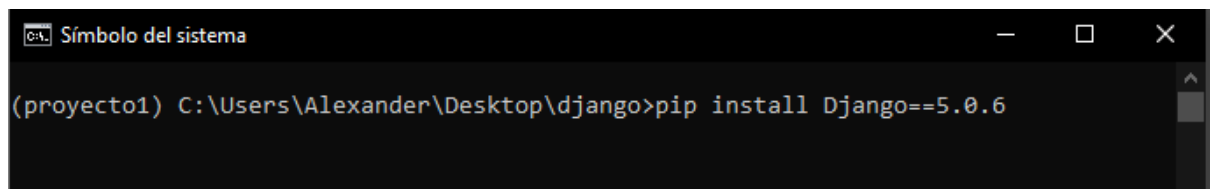
Linux / macOS:

```
python -m pip install Django==5.0.6
```

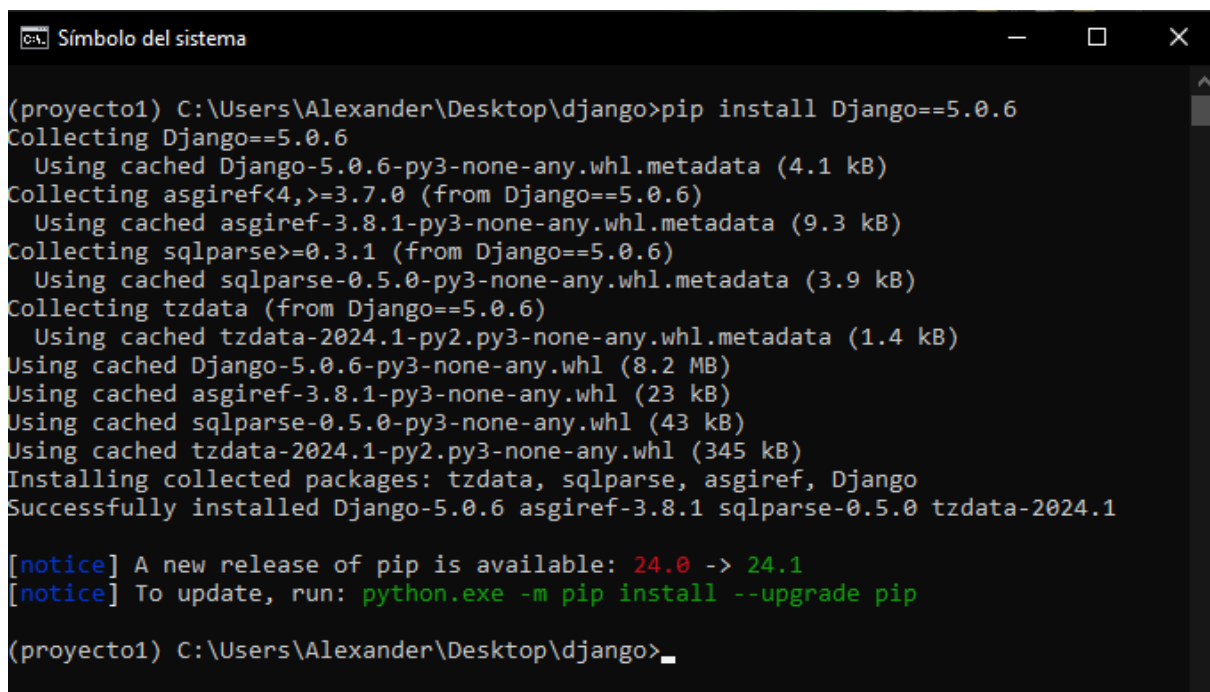
Windows:

```
py -m pip install Django==5.0.6
```

pip install Django==5.0.6 este comando lo vamos a pegar en la consola de cmd en la que ya está activado el proyecto1 y presionamos enter para que inicie la descarga e instalación.



```
(proyecto1) C:\Users\Alexander\Desktop\django>pip install Django==5.0.6
```



```
(proyecto1) C:\Users\Alexander\Desktop\django>pip install Django==5.0.6
Collecting Django==5.0.6
  Using cached Django-5.0.6-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.7.0 (from Django==5.0.6)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from Django==5.0.6)
  Using cached sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from Django==5.0.6)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.0.6-py3-none-any.whl (8.2 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.0-py3-none-any.whl (43 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-5.0.6 asgiref-3.8.1 sqlparse-0.5.0 tzdata-2024.1

[notice] A new release of pip is available: 24.0 -> 24.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(proyecto1) C:\Users\Alexander\Desktop\django>
```

Una vez que haya finalizado, comprobamos que se haya instalado correctamente con el siguiente comando: **pip freeze**

```
Selecionar Símbolo del sistema

[notice] A new release of pip is available: 24.0 -> 24.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(proyecto1) C:\Users\Alexander\Desktop\django>pip freeze
asgiref==3.8.1
Django==5.0.6
sqlparse==0.5.0
tzdata==2024.1

(proyecto1) C:\Users\Alexander\Desktop\django>
```

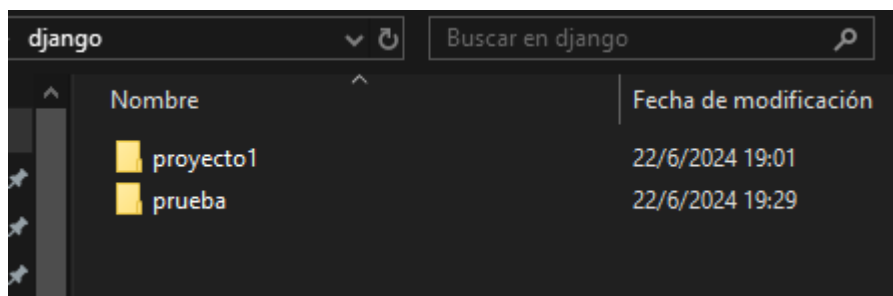
Se nos debe mostrar la versión de Django instalada, en este caso la versión 5.0.6

Creación de un proyecto en Django

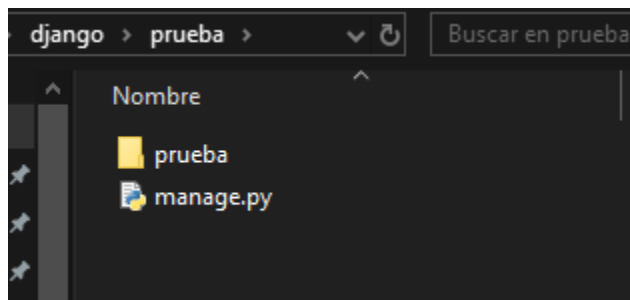
```
Símbolo del sistema

(proyecto1) C:\Users\Alexander\Desktop\django>django-admin startproject prueba
(proyecto1) C:\Users\Alexander\Desktop\django>
```

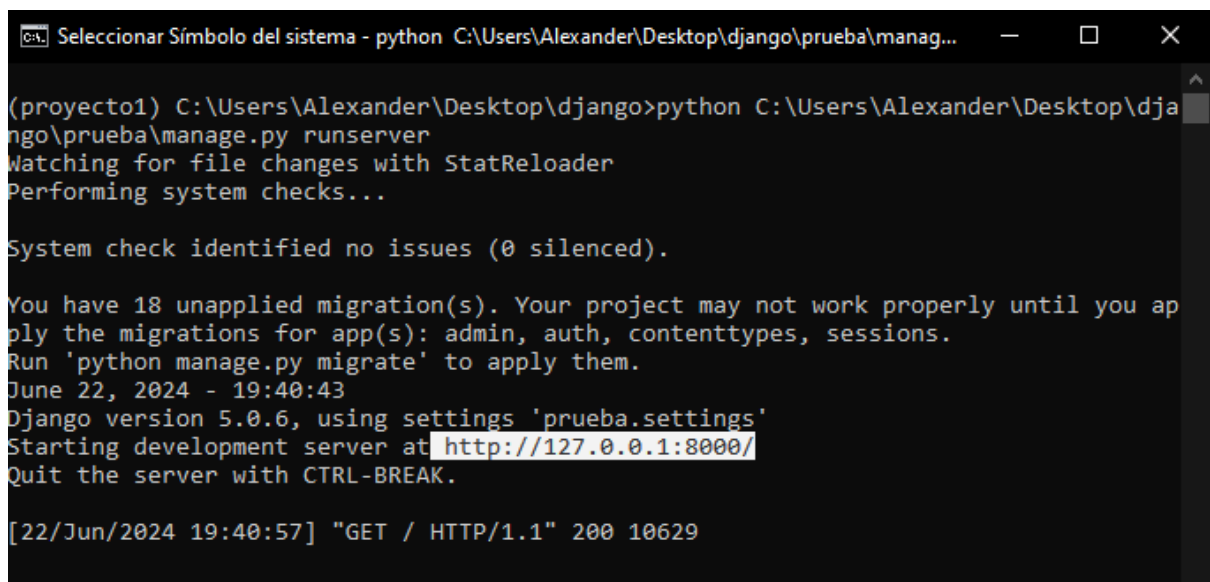
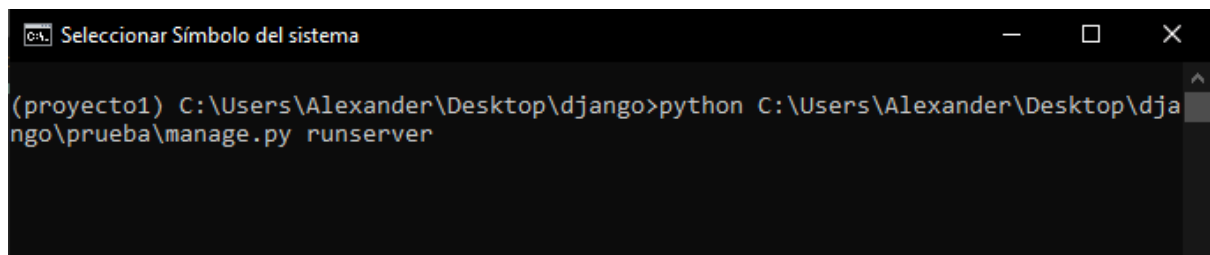
Limpiamos la consola y con el comando: **django-admin startproject prueba** creamos un nuevo proyecto (“**prueba**” es el nombre del proyecto, este puede ser cualquiera).



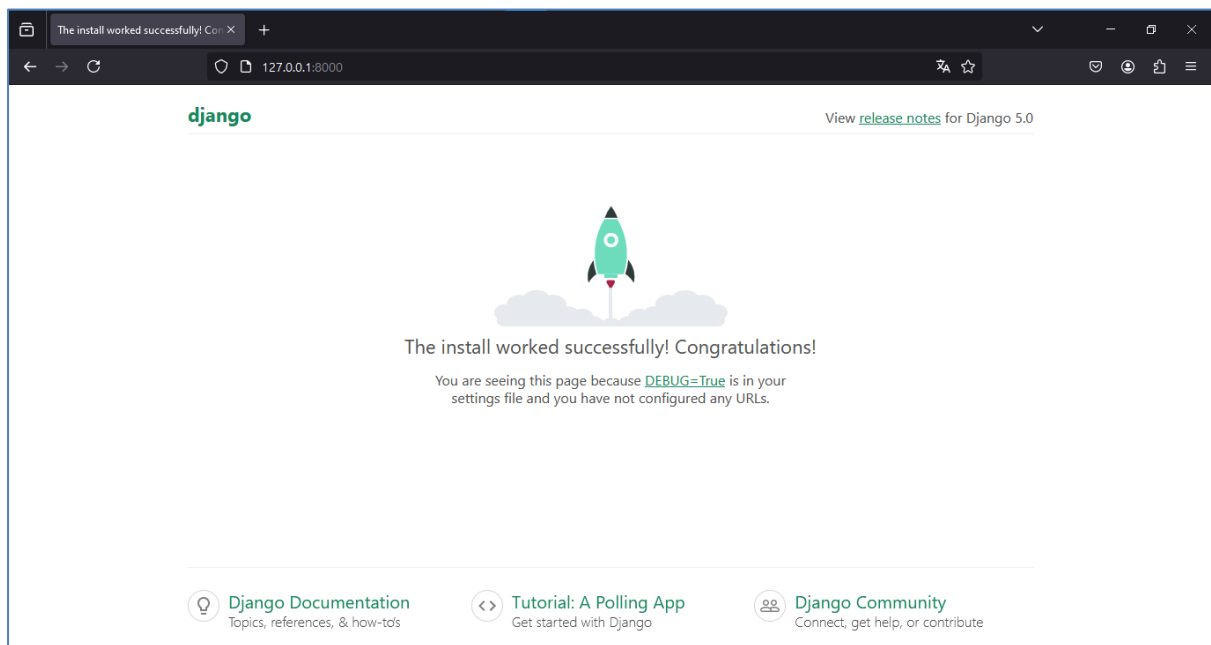
Se nos debe mostrar la carpeta prueba, y si ingresamos dentro de ella veremos los siguientes archivos.



Limpiamos la consola y colocamos el comando: **python ruta_manage.py runserver** (para obtener la ruta, solo hay que arrastrar el archivo manage.py al cmd) y presionamos enter.

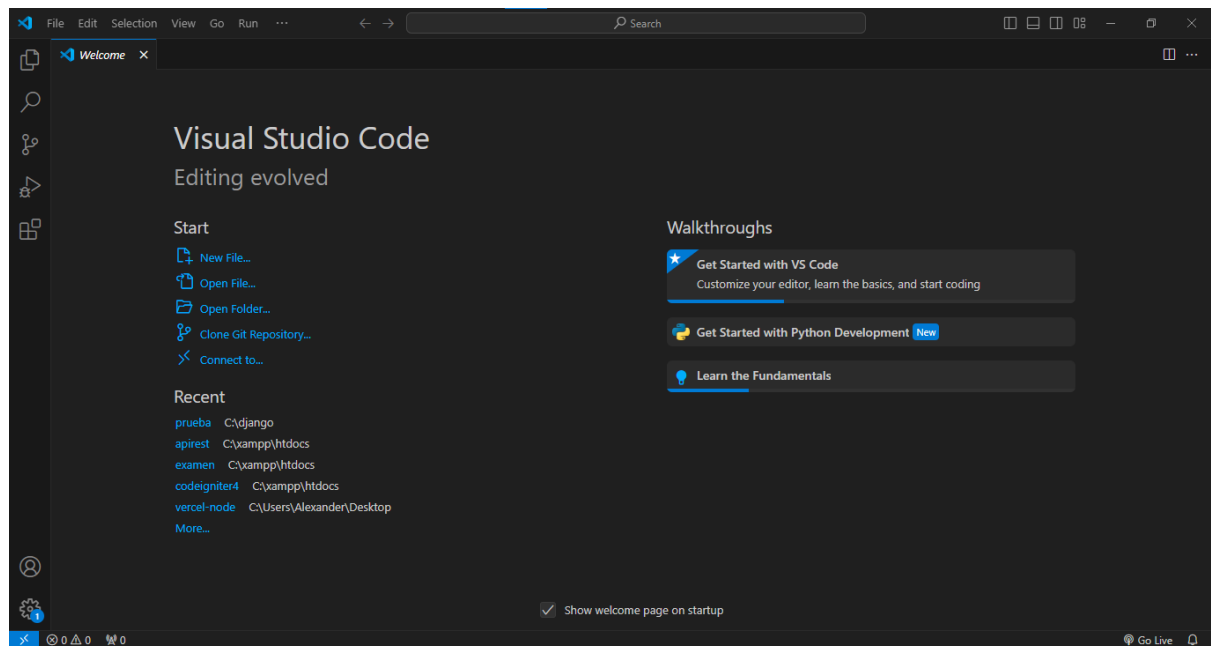


Si todo salió bien, se debe iniciar el servidor y copiamos la url y la pegamos en cualquier navegador.

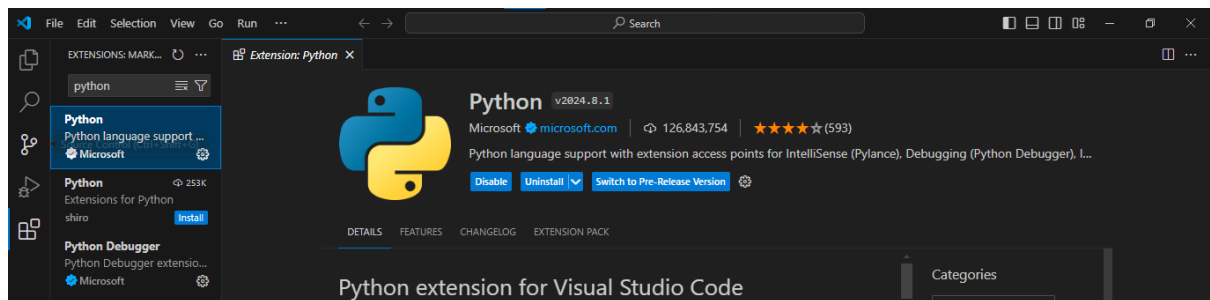


Se nos debe mostrar la página de inicio de django. Con esto ya quedo instalado Django correctamente. Para detener el servidor, solo presionamos la combinación de teclas: **control + c**

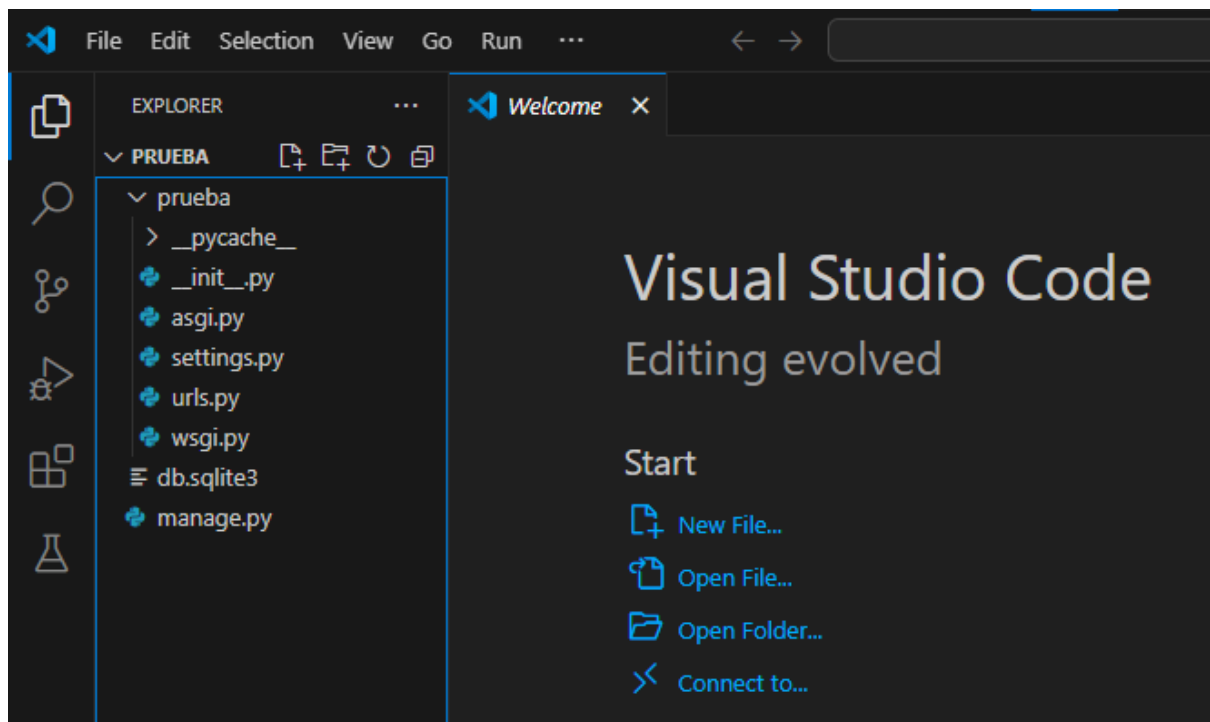
Programacion de la Aplicación



Vamos a necesitar un IDE de desarrollo compatible con Python (puede ser cualquiera), en este caso se usará Visual Studio Code.

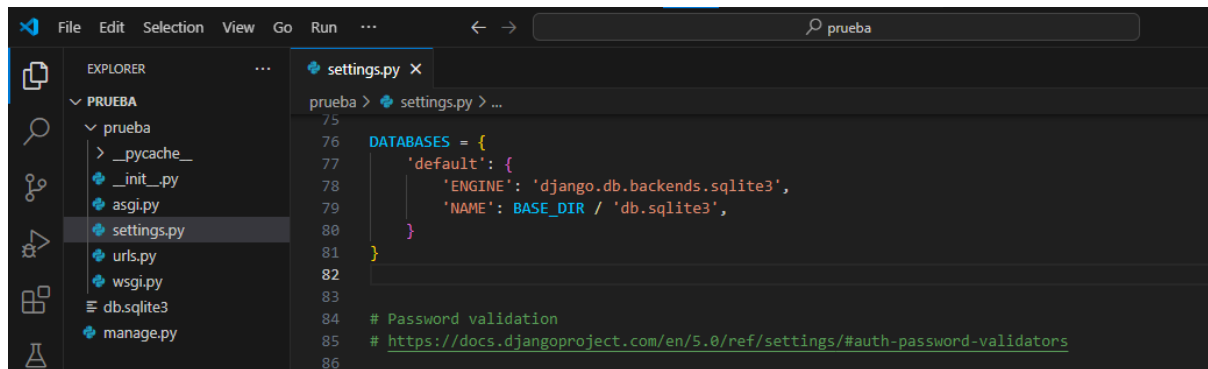


Nos situamos en el apartado de extensiones y buscamos Python y lo instalamos.



Procedemos a abrir la carpeta prueba con visual code. Y se nos debe mostrar la estructura de los archivos como se visualiza en la imagen.

Abrimos el archivo **setting.py** y procedemos a bajar hasta el apartado de **DATABASES**



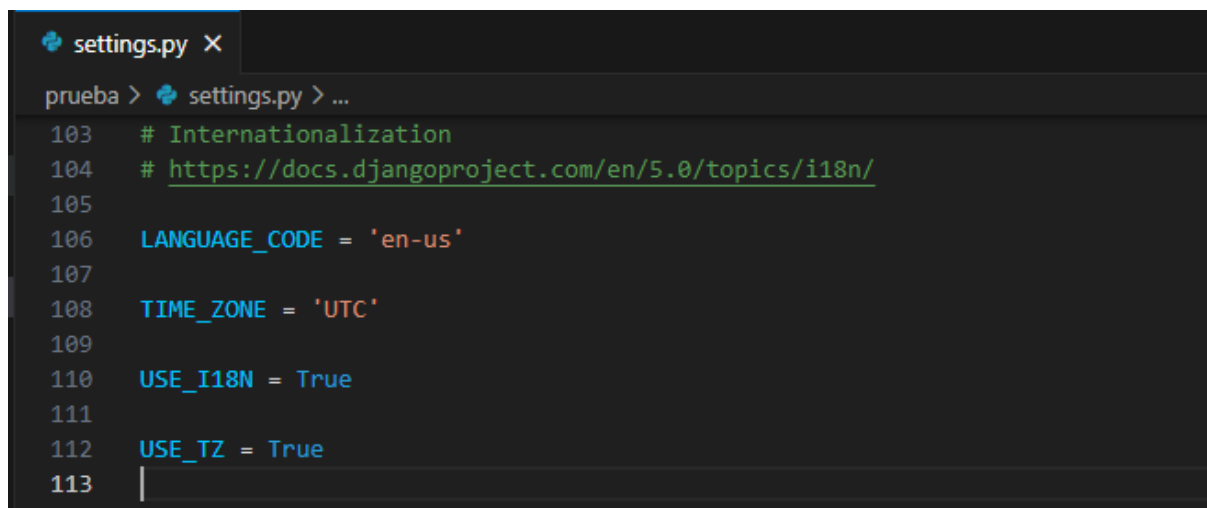
```
File Edit Selection View Go Run ...
prueba > settings.py > ...
75
76 DATABASES = {
77     'default': {
78         'ENGINE': 'django.db.backends.sqlite3',
79         'NAME': BASE_DIR / 'db.sqlite3',
80     }
81 }
82
83
84 # Password validation
85 # https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators
86
```

Vamos a trabajar con el motor de base de datos de SQLite.



```
settings.py X
prueba > settings.py > ...
74 # https://docs.djangoproject.com/en/5.0/ref/settings/#databases
75
76 DATABASES = {
77     'default': {
78         'ENGINE': 'django.db.backends.sqlite3',
79         'NAME': BASE_DIR / 'MiUniversidad.db',
80     }
81 }
82
83
```

Vamos a crear una base de datos que se va a llamar **MiUniversidad.db**



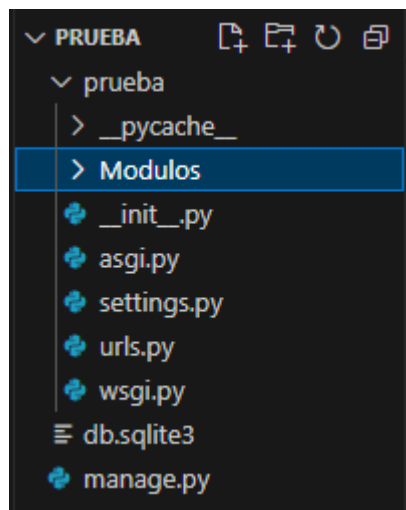
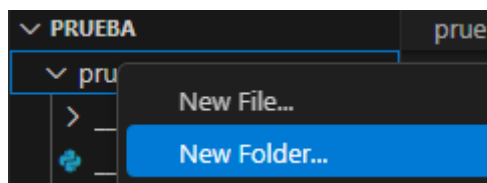
```
settings.py X
prueba > settings.py > ...
103 # Internationalization
104 # https://docs.djangoproject.com/en/5.0/topics/i18n/
105
106 LANGUAGE_CODE = 'en-us'
107
108 TIME_ZONE = 'UTC'
109
110 USE_I18N = True
111
112 USE_TZ = True
113 |
```

Seguimos bajando y ahora cambiamos el lenguaje de nuestra región, por defecto está en inglés, así que la cambiamos a español ecuador, no olvidar guardar los cambios con **Control + s**

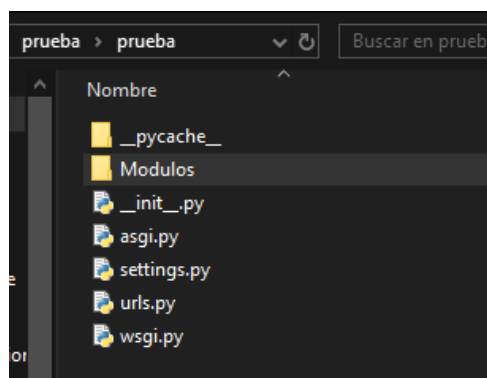
```
# Internationalization
# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE_CODE = 'es-ec'
```

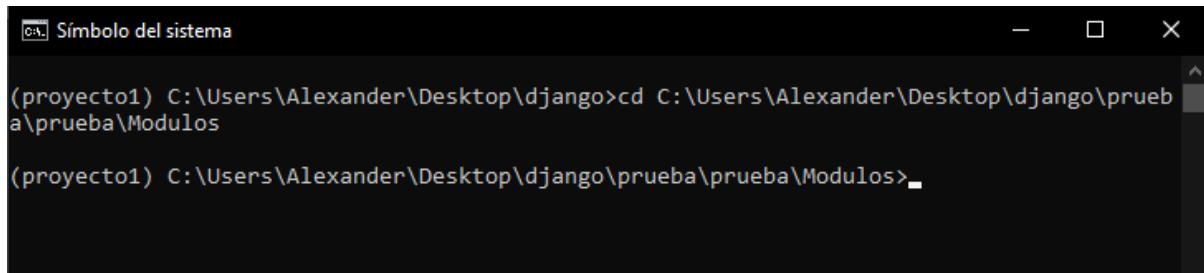
ahora creamos una nueva carpeta en la raíz, esta se va a llamar **Modulos**



Nos dirigimos a la carpeta de prueba y dentro debe estar la carpeta de Módulos que habíamos creado.



Vamos a ingresar a esa carpeta por medio de la consola de cmd, escribiendo el comando: **cd ruta_Modulos** (arrastramos la carpeta Módulos para obtener la ruta).

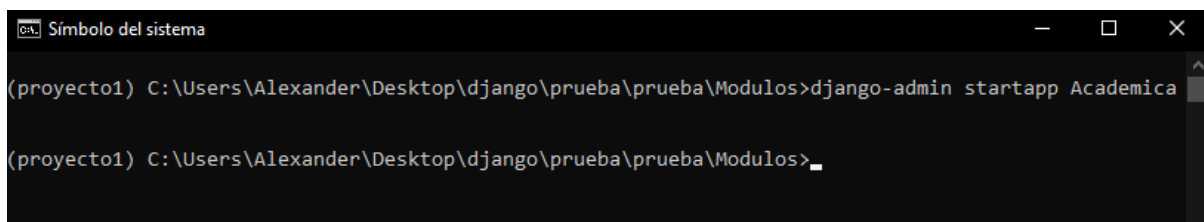


```
Símbolo del sistema

(proyecto1) C:\Users\Alexander\Desktop\django>cd C:\Users\Alexander\Desktop\django\prueba\prueba\Modules

(proyecto1) C:\Users\Alexander\Desktop\django\prueba\prueba\Modules>
```

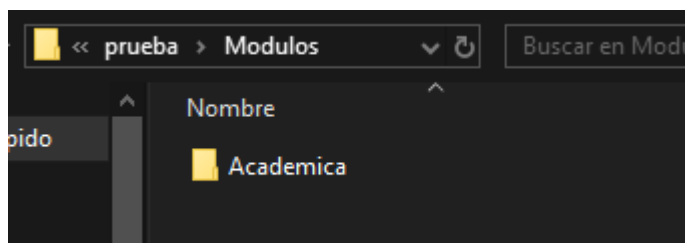
Una vez dentro, vamos a colocar el comando: **django-admin startapp Académica** y presionamos enter.



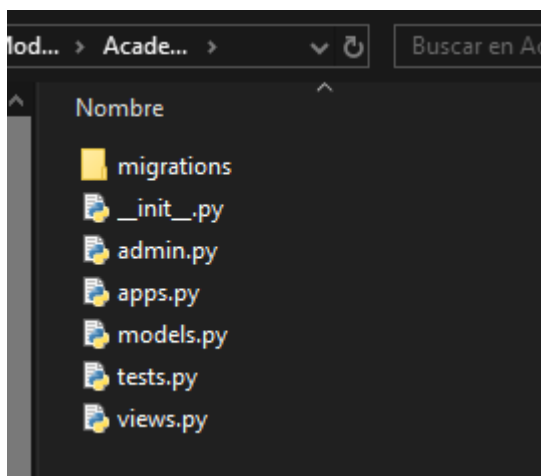
```
Símbolo del sistema

(proyecto1) C:\Users\Alexander\Desktop\django\prueba\prueba\Modules>django-admin startapp Académica

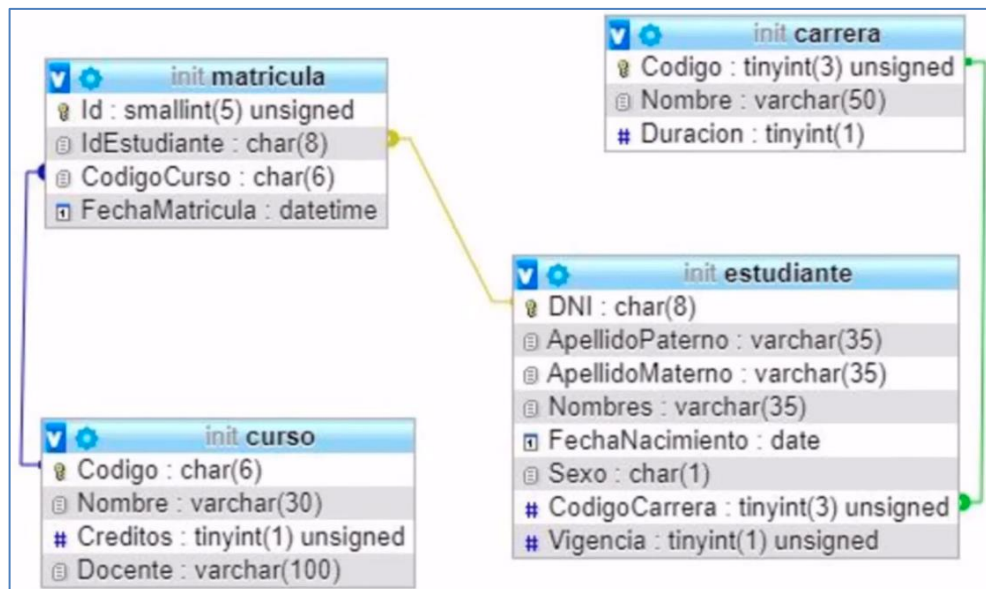
(proyecto1) C:\Users\Alexander\Desktop\django\prueba\prueba\Modules>
```



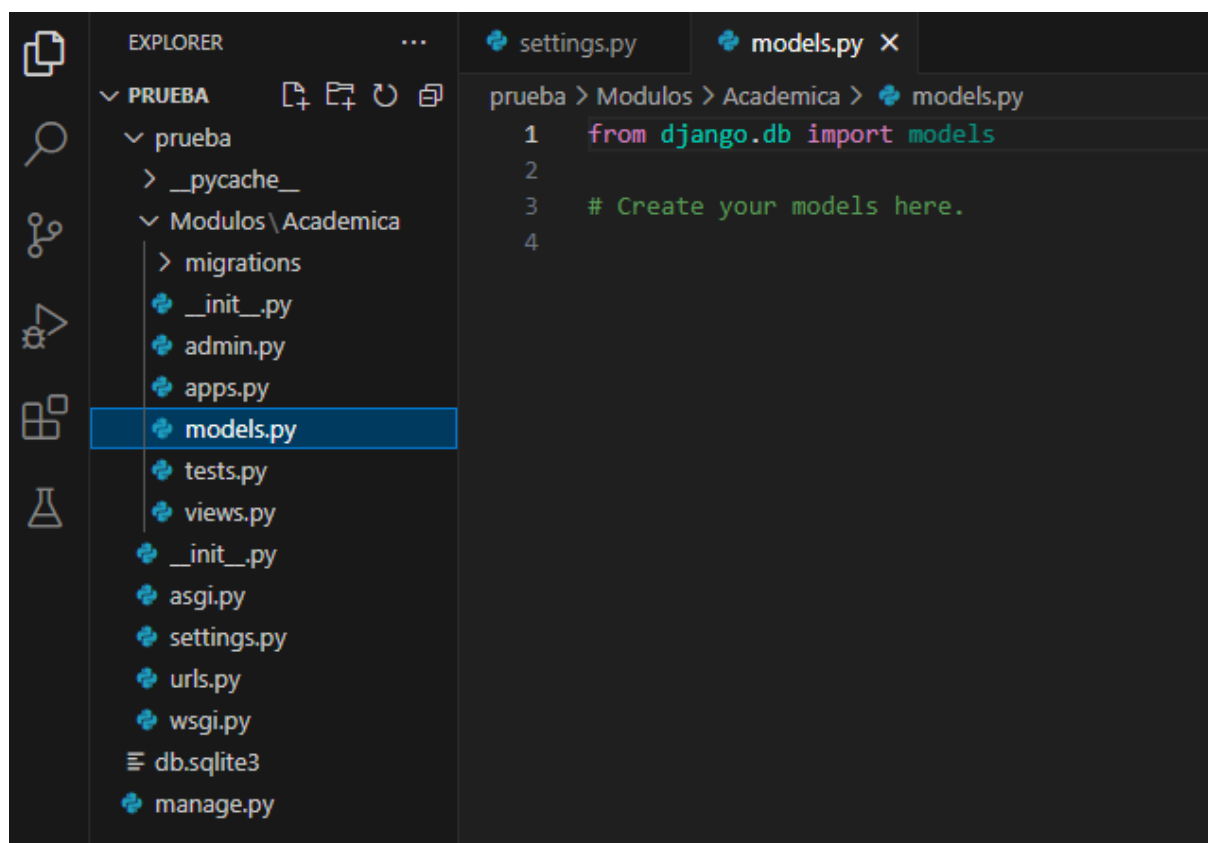
Si abrimos la carpeta Módulos, veremos que se habrá creado una carpeta llamada **Académica**, y si la abrimos vamos a ver varios archivos, que corresponden a la vista controlador.



Vamos a crear las siguientes tablas para la base de datos **MiUniversidad**.



Devuelta en visual studio code, vamos a abrir el archivo de **models.py** que se encuentra dentro de la carpeta Académica.



Procedemos a escribir la estructura de cada tabla junto con sus relaciones.

```
class Carrera(models.Model):
    codigo = models.CharField(max_length=3, primary_key=True)
    nombre = models.CharField(max_length=50)
    duracion = models.PositiveSmallIntegerField(default=5)

    def __str__(self):
        txt = "{0} (Duración: {1} año(s))"
        return txt.format(self.nombre, self.duracion)
```

```
class Estudiante(models.Model):
    dni = models.CharField(max_length=10, primary_key=True)
    apellidoPaterno = models.CharField(max_length=35)
    apellidoMaterno = models.CharField(max_length=35)
    nombres = models.CharField(max_length=35)
    fechaNacimiento = models.DateField()
    sexos = [('F', 'Femenino'), ('M', 'Masculino')]
    sexo = models.CharField(max_length=1, choices=sexos, default='F')
    carrera = models.ForeignKey(Carrera, null=False, blank=False, on_delete=models.CASCADE)
    vigencia = models.BooleanField(default=True)

    def nombreCompleto(self):
        txt = "{0},{1},{2}"
        return txt.format(self.apellidoPaterno, self.apellidoMaterno, self.nombres)

    def __str__(self):
        txt = "{0} / Carrera: {1} / {2}"
        if self.vigencia:
            estadoEstudiante = "VIGENTE"
        else:
            estadoEstudiante = "DE BAJA"
        return txt.format(self.nombreCompleto(), self.carrera, estadoEstudiante)
```

```
class Curso(models.Model):
    codigo = models.CharField(max_length=6, primary_key=True)
    nombre = models.CharField(max_length=30)
    creditos = models.PositiveSmallIntegerField()
    docente = models.CharField(max_length=100)

    def __str__(self):
        txt = "{0} ({1}) / Docente: {2}"
        return txt.format(self.nombre, self.codigo, self.docente)
```

```

class Matricula(models.Model):
    id = models.AutoField(primary_key=True)
    estudiante = models.ForeignKey(Estudiante, null=False, blank=False, on_delete=models.CASCADE)
    curso = models.ForeignKey(Curso, null=False, blank=False, on_delete=models.CASCADE)
    fechaMatricula = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        txt = "{0} matriculad{1} en el curso {2} / Fecha: {3}"
        if self.estudiante.sexo == "F":
            letraSexo = "a"
        else:
            letraSexo = "o"
        fecMat = self.fechaMatricula.strftime("%A %d/%m/%Y %H:%M:%S")
        return txt.format(self.estudiante.nombreCompleto(), letraSexo, self.curso, fecMat)

```

Una vez creadas las tablas con sus respectivas filas y columnas, procedemos a abrir el archivo de **admin.py** e importamos los módulos y los registros de las tablas.

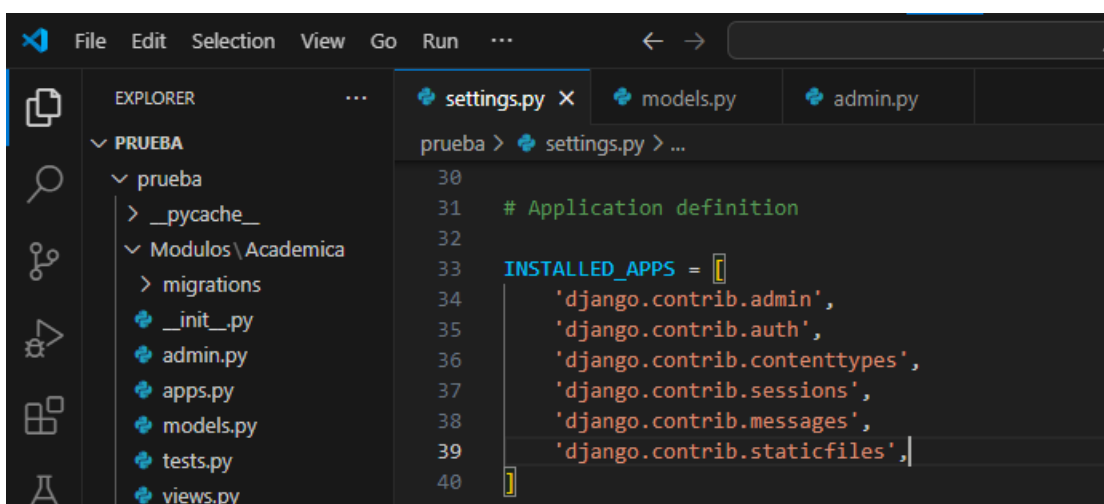


```

prueba > Modulos > Academica > admin.py
1  from django.contrib import admin
2
3  # Register your models here.
4
5  from Modulos.Academica.models import * #Se importan todos los modulos
6
7  admin.site.register(Carrera)
8  admin.site.register(Estudiante)
9  admin.site.register(Curso)
10 admin.site.register(Matricula)
11

```

Ahora nos dirigimos al archivo de **setting.py** y agregamos el nombre de nuestra app, que se encuentra en Módulos Académica.



```

File Edit Selection View Go Run ...
EXPLORER
  PRUEBA
    prueba
      __pycache__
    Modulos\Academica
      migrations
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      views.py
  settings.py
  models.py
  admin.py

prueba > settings.py > ...
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'Modulos.Academica',
41 ]

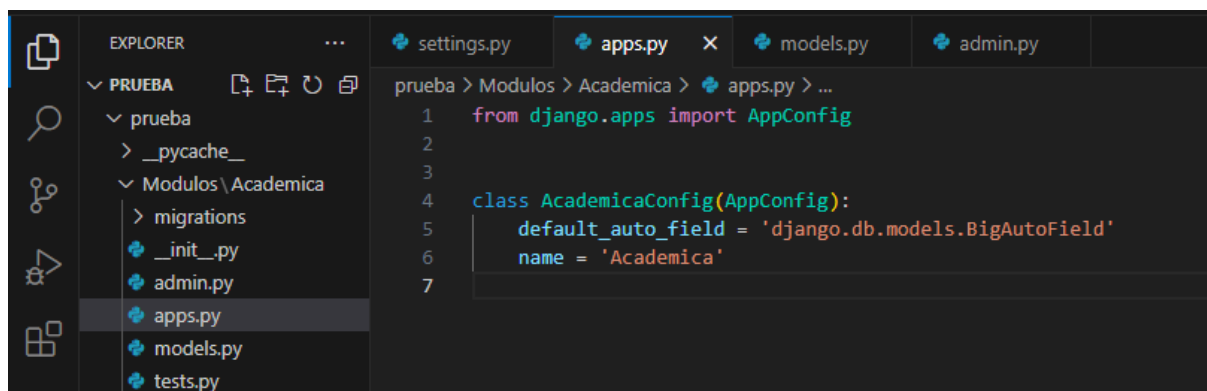
```


Nos debe quedar de la siguiente manera.

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'Modulos.Academica' #nombre de la apps
]
```

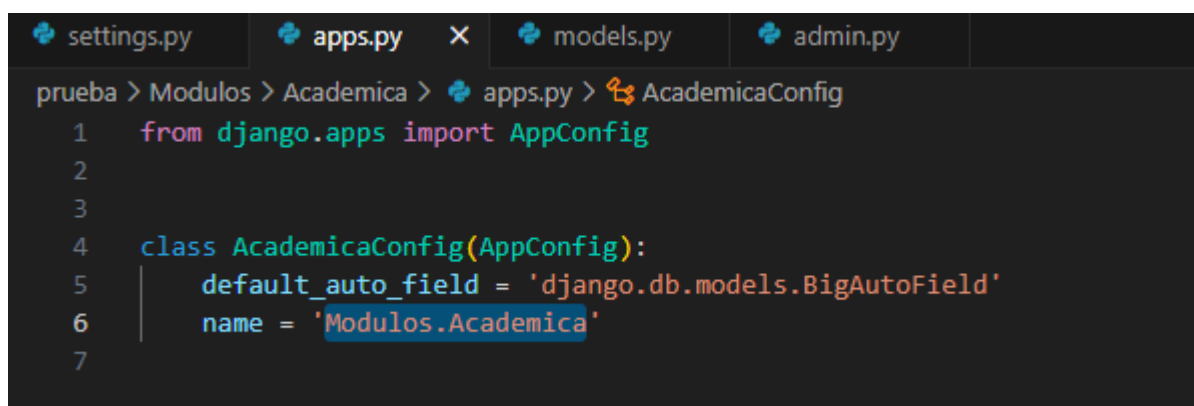
Luego abrimos el archivo de **apps.py** y en el **name** colocamos lo mismo.



The screenshot shows the VS Code interface. On the left, the Explorer pane displays the project structure: 'PRUEBA' (expanded) contains 'prueba' (expanded) and 'Modulos\Academica' (expanded). Under 'Modulos\Academica', there are 'migrations', '__init__.py', 'admin.py', 'apps.py' (selected), 'models.py', and 'tests.py'. The main editor shows the 'apps.py' file with the following code:

```
1 from django.apps import AppConfig
2
3
4 class AcademicaConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'Academica'
7
```

Nos debe quedar de la siguiente manera.



The screenshot shows the VS Code interface with the 'apps.py' file open. The Explorer pane on the left is the same as the previous screenshot. The main editor shows the 'apps.py' file with the following code:

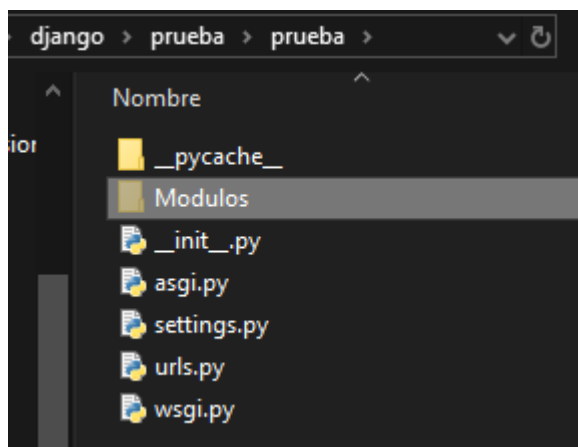
```
1 from django.apps import AppConfig
2
3
4 class AcademicaConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'Modulos.Academica'
7
```

Devuelta en la consola de cmd, vamos a regresar a la raíz de la carpeta prueba, con el comando **cd..** y nuevamente **cd..**.

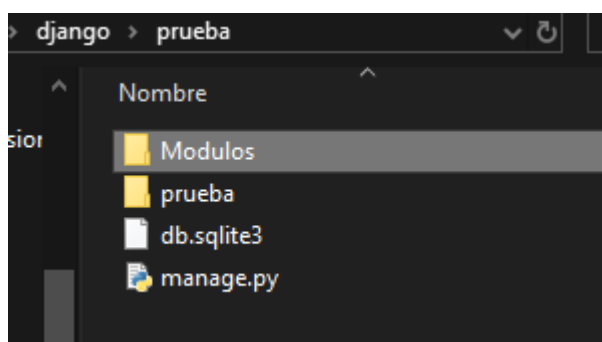
```
Símbolo del sistema

(proyecto1) C:\Users\Alexander\Desktop\django\prueba\prueba\Modulos>cd..
(proyecto1) C:\Users\Alexander\Desktop\django\prueba\prueba>cd..
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>_
```

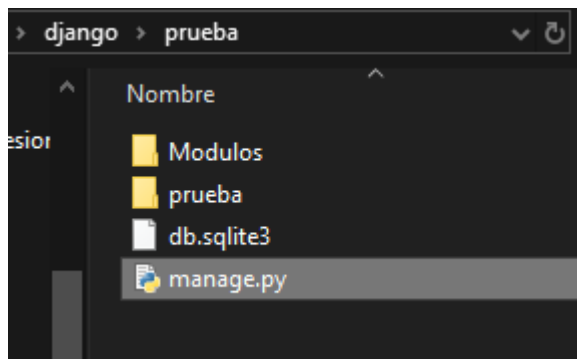
Ahora vamos a movemos la carpeta de Módulos a la raíz de la carpeta prueba.



Debe quedar de la siguiente manera.



Una vez en la raíz de la carpeta prueba, vamos a escribir el comando: **python ruta_manage.py migrate** (no olvidar que la ruta del archivo manage.py la obtenemos arrastrando el archivo a la consola)



```
Símbolo del sistema

(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py migrate
```

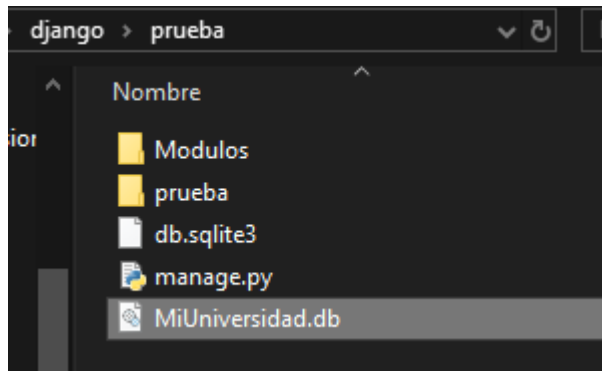
Al presionar enter, comenzara la migración de la base de datos.

```
Símbolo del sistema

(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\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

(proyecto1) C:\Users\Alexander\Desktop\django\prueba>
```

Nos aparecerá la base de datos de **MiUniversidad.db** en la raíz de la carpeta prueba.



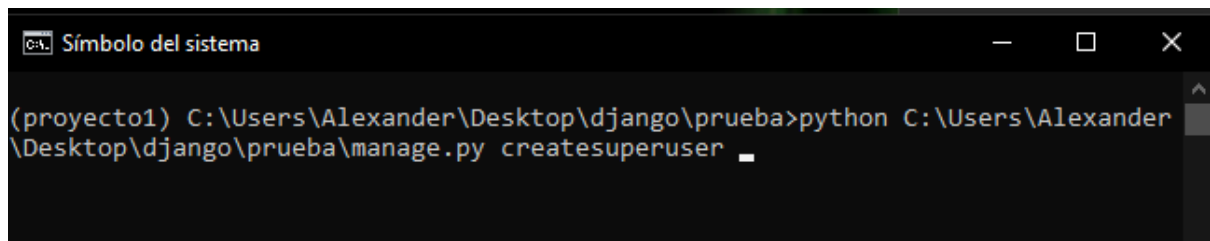
Una vez en la raíz de la carpeta prueba, vamos a escribir el comando: **python ruta_manage.py makemigrations** (no olvidar que la ruta del archivo manage.py la obtenemos arrastrando el archivo a la consola).

A screenshot of a Windows Command Prompt window titled 'Símbolo del sistema'. The prompt shows the command: `(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py makemigrations`. The cursor is at the end of the command line.

Al presionar enter, se deben crear las migraciones de nuestras tablas de la base de datos.

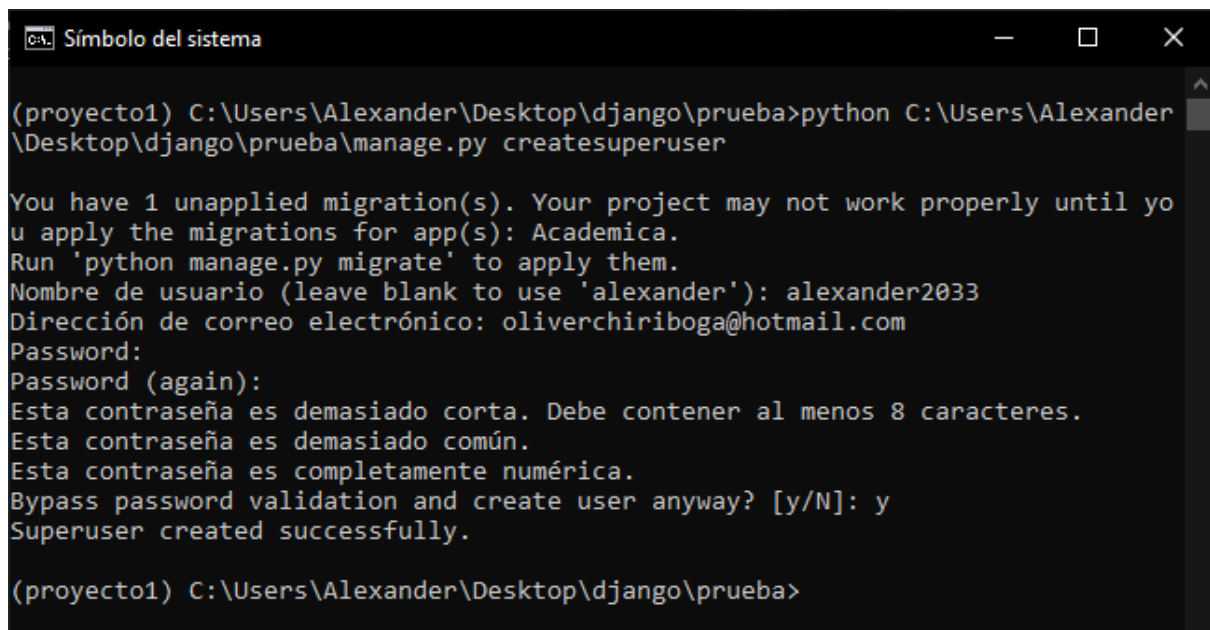
A screenshot of a Windows Command Prompt window titled 'Símbolo del sistema'. The prompt shows the command: `(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py makemigrations`. The output of the command is displayed: `Migrations for 'Academica':` followed by an indented list: `Modulos\Academica\migrations\0001_initial.py`, `- Create model Carrera`, `- Create model Curso`, `- Create model Estudiante`, and `- Create model Matricula`. The prompt then shows the cursor on a new line: `(proyecto1) C:\Users\Alexander\Desktop\django\prueba>_`.

Creación del super usuario y contraseña.



```
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py createsuperuser
```

En la raíz de la carpeta prueba, vamos a escribir el comando: **python ruta_manage.py createsuperuser** (la ruta del archivo manage.py la obtenemos arrastrando el archivo a la consola).



```
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py createsuperuser

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): Academica.
Run 'python manage.py migrate' to apply them.
Nombre de usuario (leave blank to use 'alexander'): alexander2033
Dirección de correo electrónico: oliverchiriboga@hotmail.com
Password:
Password (again):
Esta contraseña es demasiado corta. Debe contener al menos 8 caracteres.
Esta contraseña es demasiado común.
Esta contraseña es completamente numérica.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

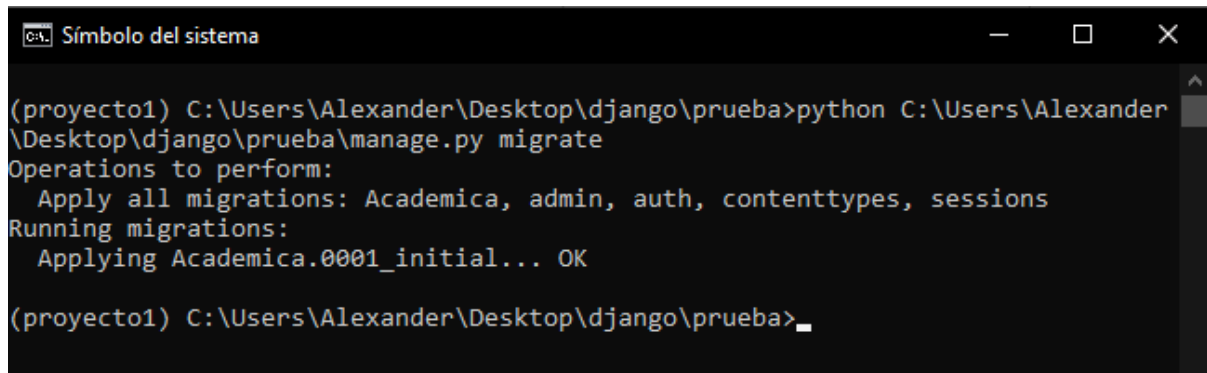
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>
```

Agregamos un nombre de usuario, correo electrónico y la contraseña.

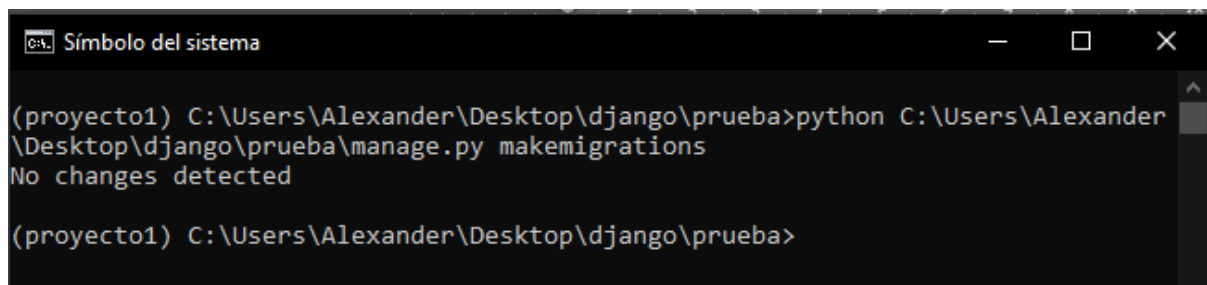
Es importante que, al momento de hacer algún cambio, ya sea de usuario o de alguna modificación en la base de datos o archivos de configuración, se deba usar los comandos:

python ruta_manage.py migrate

python ruta_manage.py makemigrations



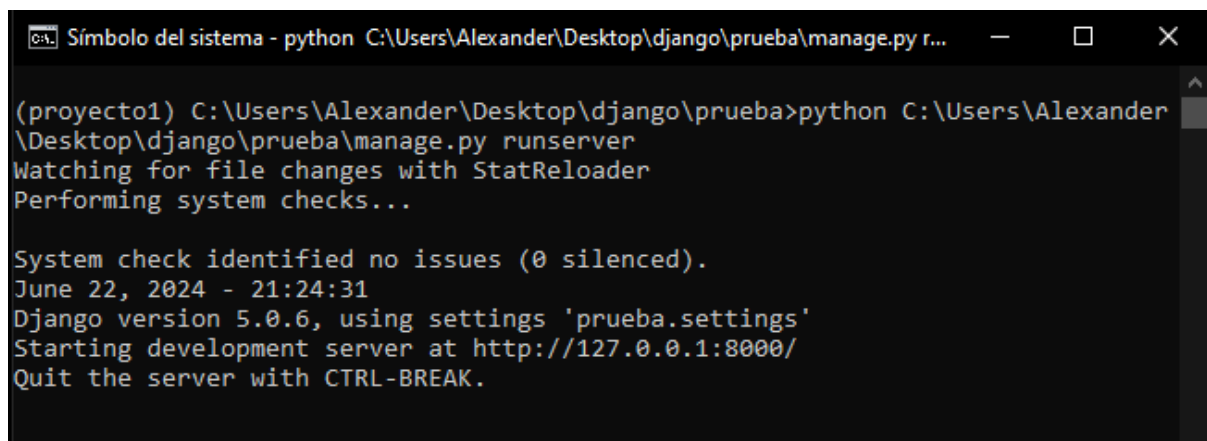
```
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py migrate
Operations to perform:
  Apply all migrations: Academica, admin, auth, contenttypes, sessions
Running migrations:
  Applying Academica.0001_initial... OK
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>
```



```
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py makemigrations
No changes detected
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>
```

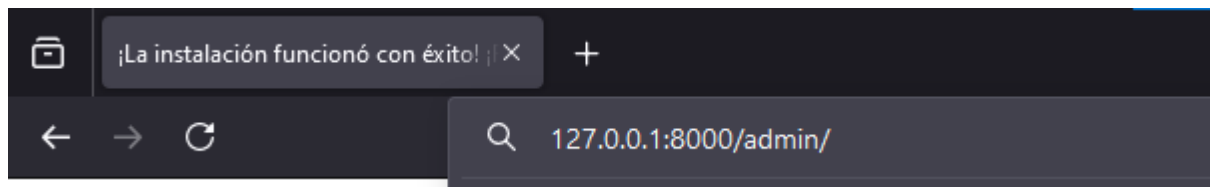
Ahora vamos a iniciar el servidor para ver como se visualiza nuestro proyecto.

Limpiamos la consola y colocamos el comando: **python ruta_manage.py runserver** (para obtener la ruta, se arrastrar el archivo manage.py al cmd) y presionamos enter.

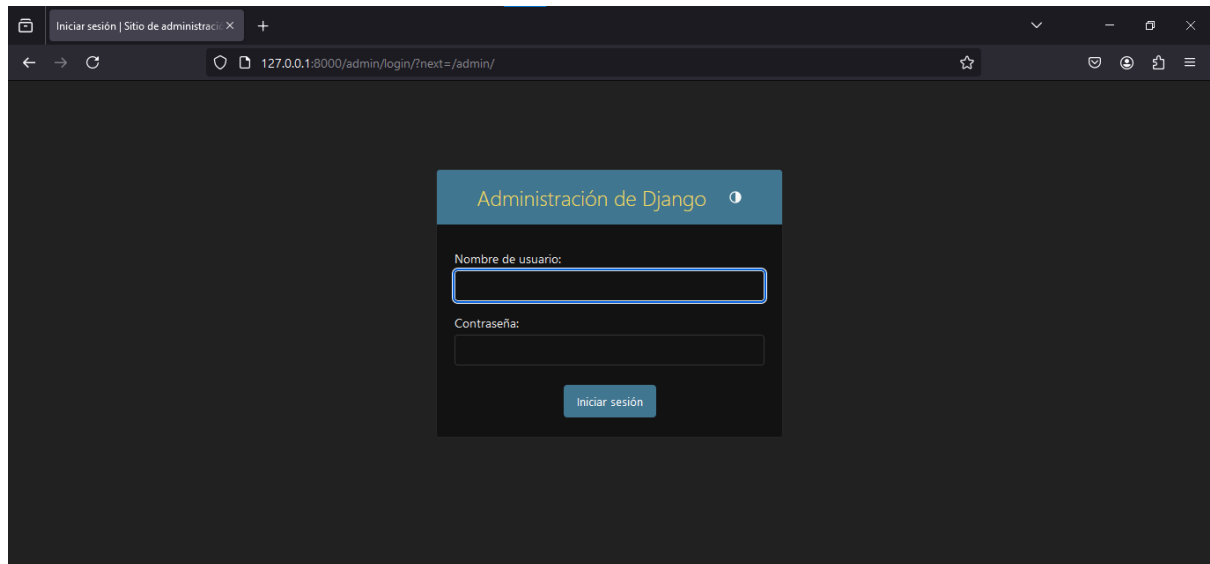


```
(proyecto1) C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prueba\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

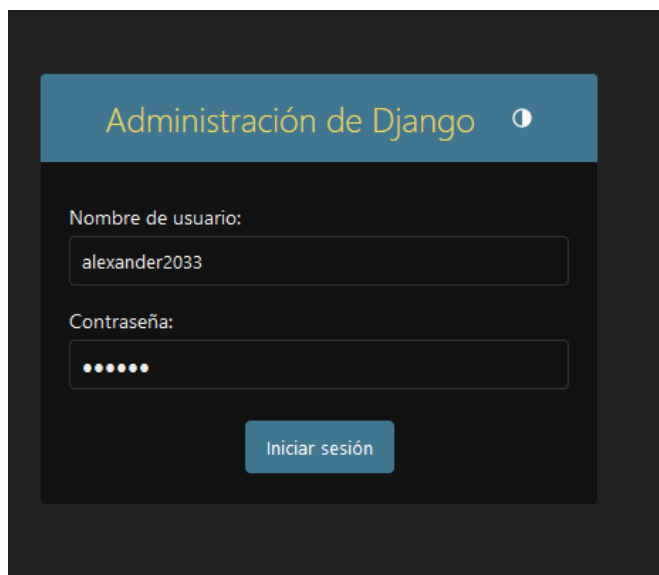
System check identified no issues (0 silenced).
June 22, 2024 - 21:24:31
Django version 5.0.6, using settings 'prueba.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

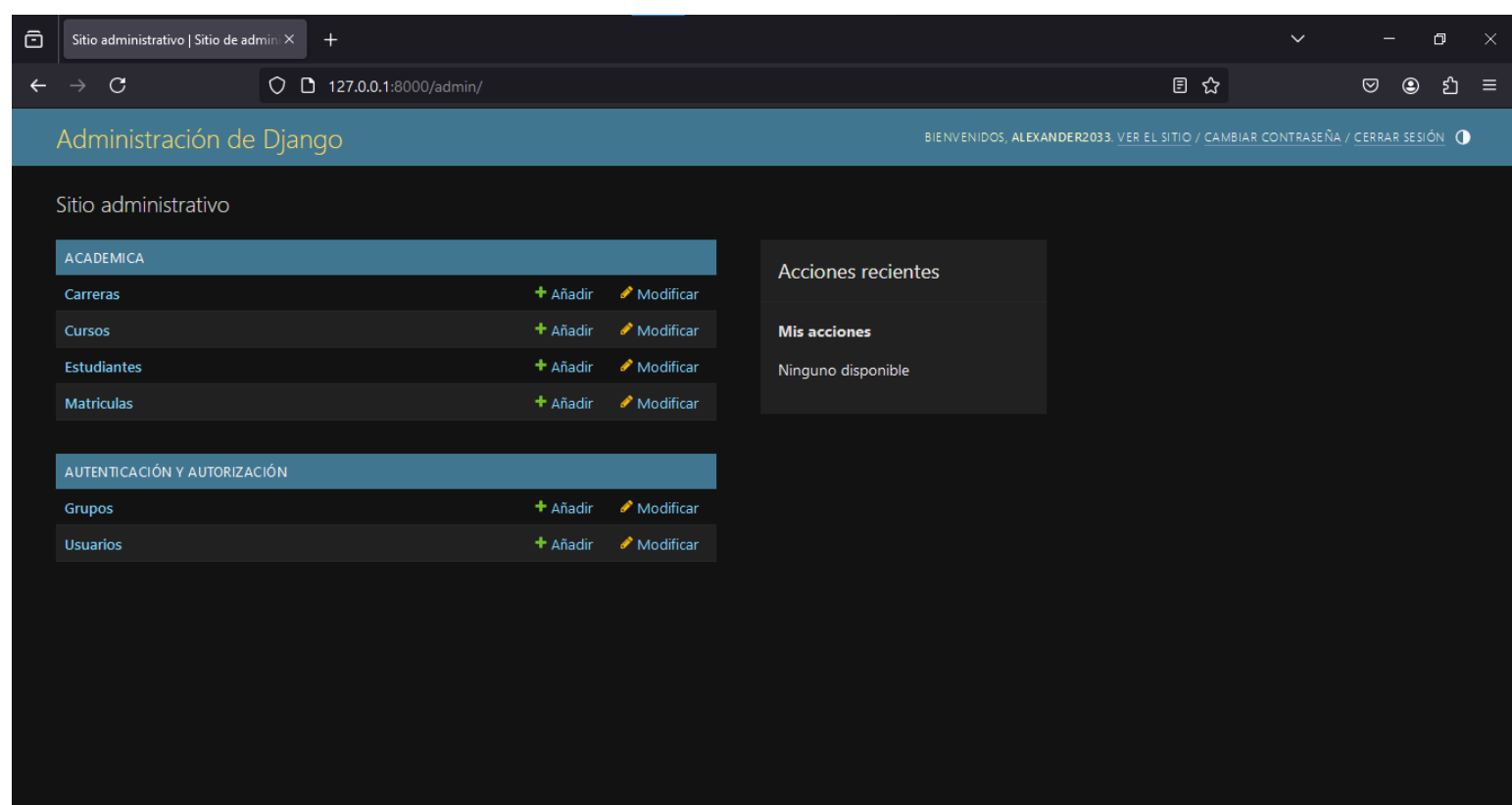


En la URL de la ruta del servidor agregamos admin

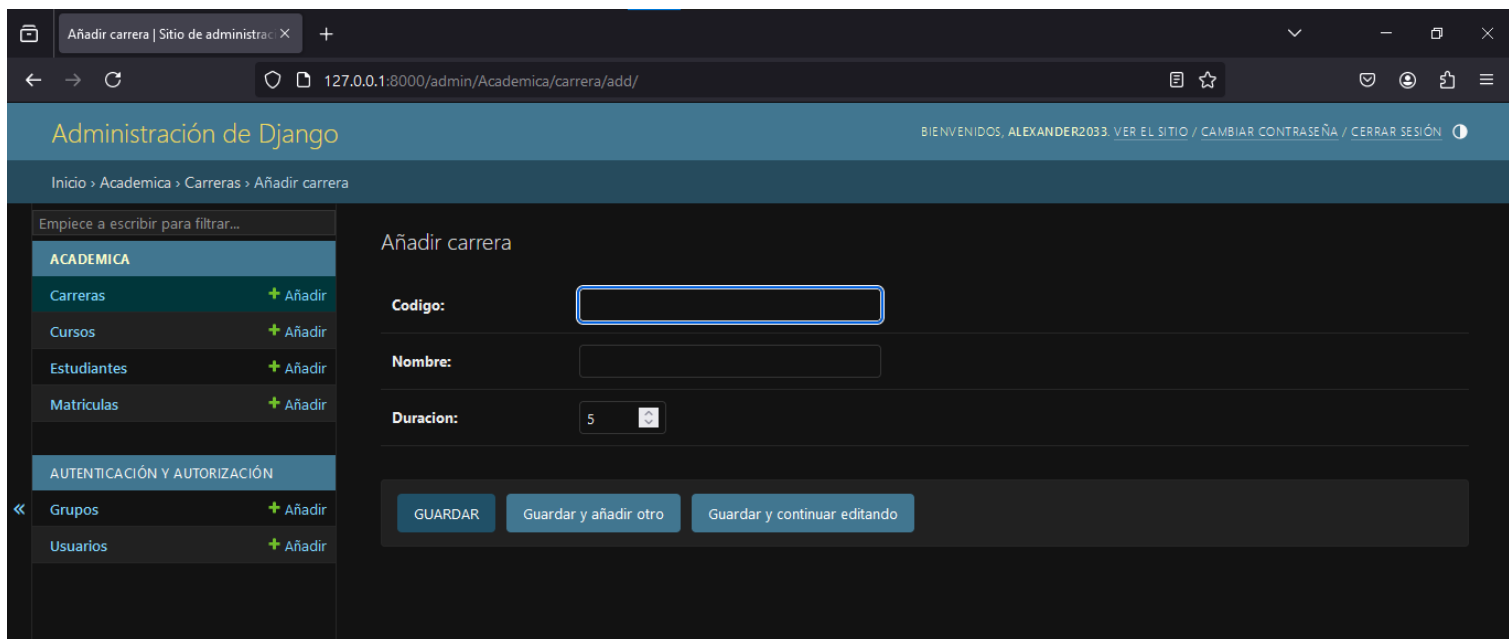


Iniciamos sesión con el usuario que habíamos creado y su contraseña.





Aquí ya podemos ir agregando información a cada tabla.



The screenshot shows the Django Admin interface for adding a student. The browser address bar shows the URL `127.0.0.1:8000/admin/Academica/estudiante/add/`. The page title is "Administración de Django". The breadcrumb trail is "Inicio > Academica > Estudiantes > Añadir estudiante".

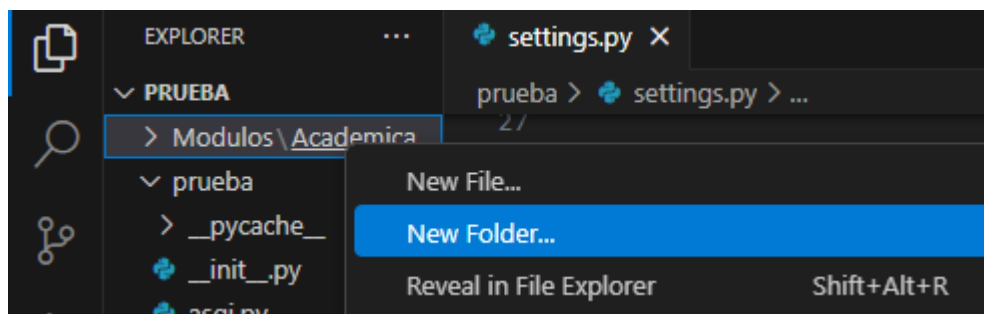
On the left, there is a sidebar with a search bar and two main sections: "ACADEMICA" and "AUTENTICACIÓN Y AUTORIZACIÓN". Under "ACADEMICA", there are links for "Carreras", "Cursos", "Estudiantes", and "Matrículas", each with a "+ Añadir" button. Under "AUTENTICACIÓN Y AUTORIZACIÓN", there are links for "Grupos" and "Usuarios", also with "+ Añadir" buttons.

The main content area is titled "Añadir estudiante". It contains several form fields:

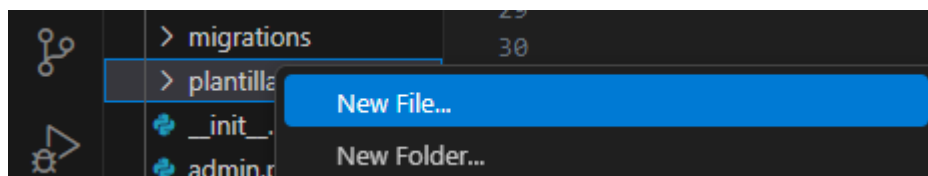
- Dni:** A text input field.
- ApellidoPaterno:** A text input field.
- ApellidoMaterno:** A text input field.
- Nombres:** A text input field.
- FechaNacimiento:** A date picker with a calendar icon and a "Hoy" button. Below it, a note says: "Nota: Usted va 5 horas por detrás de la hora del servidor."
- Sexo:** A dropdown menu with "Femenino" selected.
- Carrera:** A dropdown menu with a search icon, a "+", and an eye icon.
- Vigencia:** A checkbox that is checked.

At the bottom, there are three buttons: "GUARDAR", "Guardar y añadir otro", and "Guardar y continuar editando".

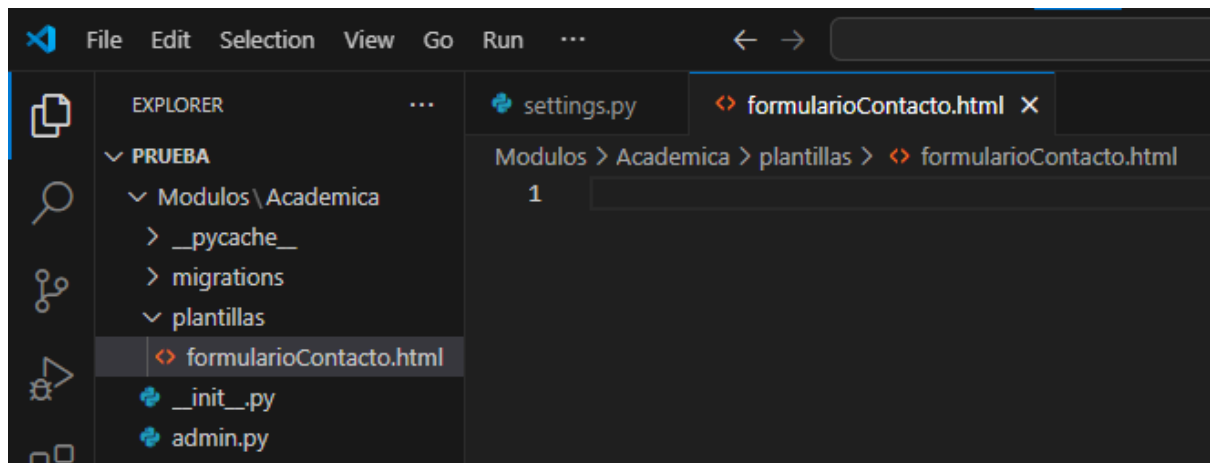
Conexión de correo electrónico por medio de Django



Dentro de la carpeta **Académica** vamos a crear una nueva carpeta llamada **plantillas**.

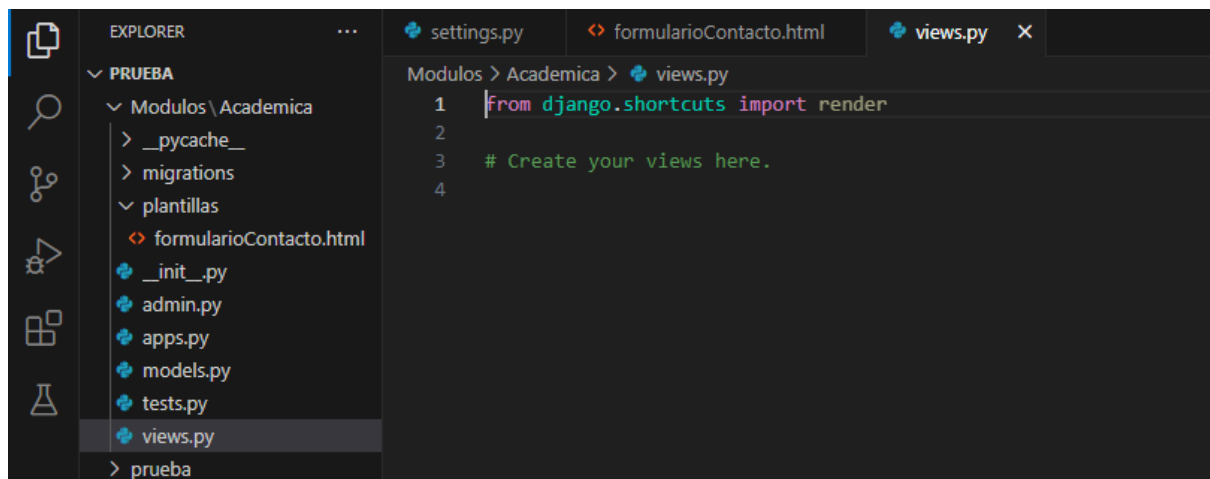


Ahora vamos a crear un archivo llamado **formularioContacto.html** dentro de la carpeta **plantillas**.

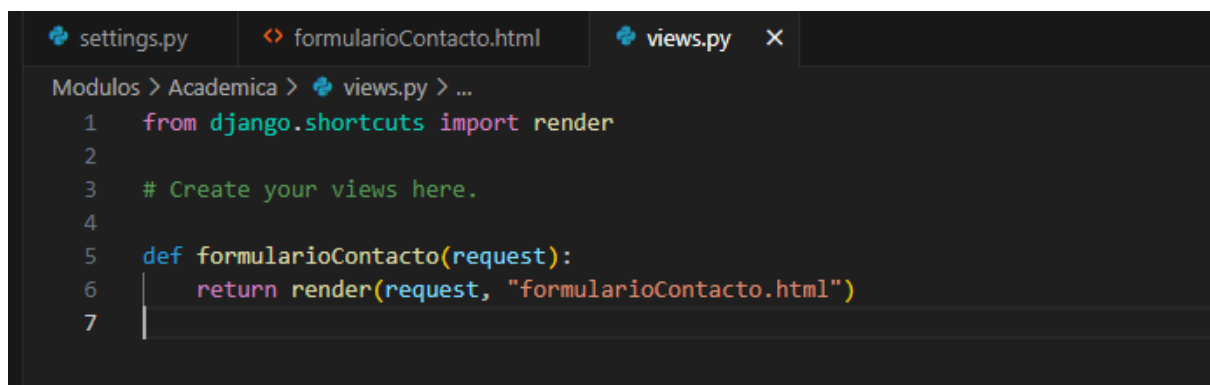


```
settings.py  formularioContacto.html X
Modulos > Academica > plantillas > formularioContacto.html > html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Formulario de Contacto</title>
7
8  </head>
9  <body>
10     <div class="contact-form-container">
11         <h2>Contacto</h2>
12         <form action="/contactar/" method="POST">{% csrf_token %}
13             <label for="subject">Asunto</label>
14             <input type="text" id="subject" name="subject" required>
15
16             <label for="email">Email</label>
17             <input type="email" id="email" name="email" required>
18
19             <label for="message">Mensaje</label>
20             <textarea id="message" name="message" rows="5" required></textarea>
21
22             <button type="submit">Enviar</button>
23         </form>
24     </div>
25 </body>
26 </html>
```

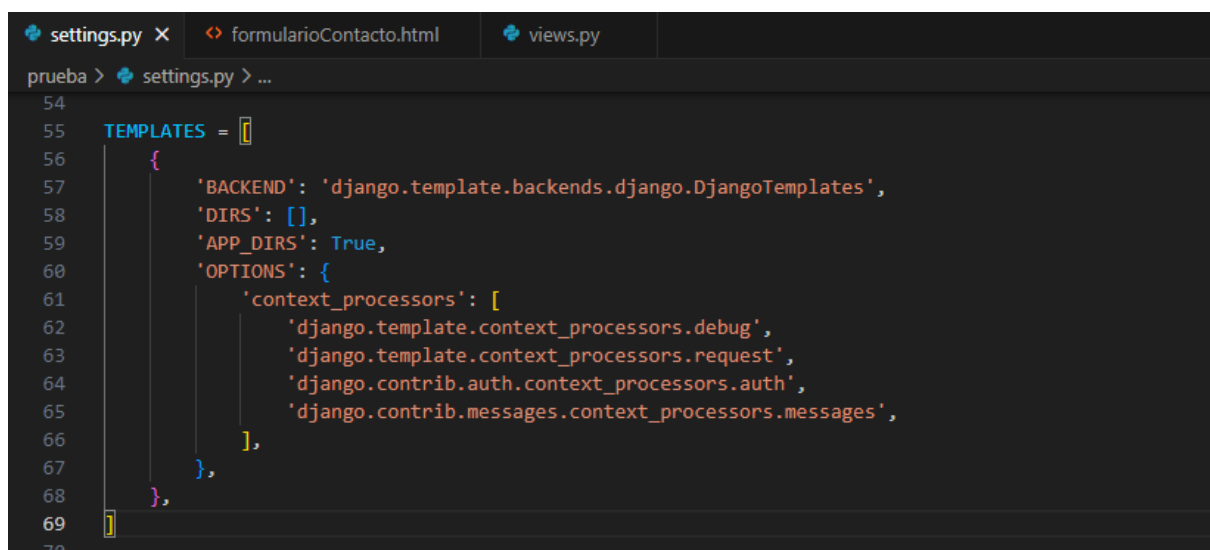
Es importante que en el formulario se agregue `{% csrf_token %}` (es una medida de seguridad para prevenir ciertos tipos de ataques web y proteger la integridad de las solicitudes enviadas a través de formularios en aplicaciones web).



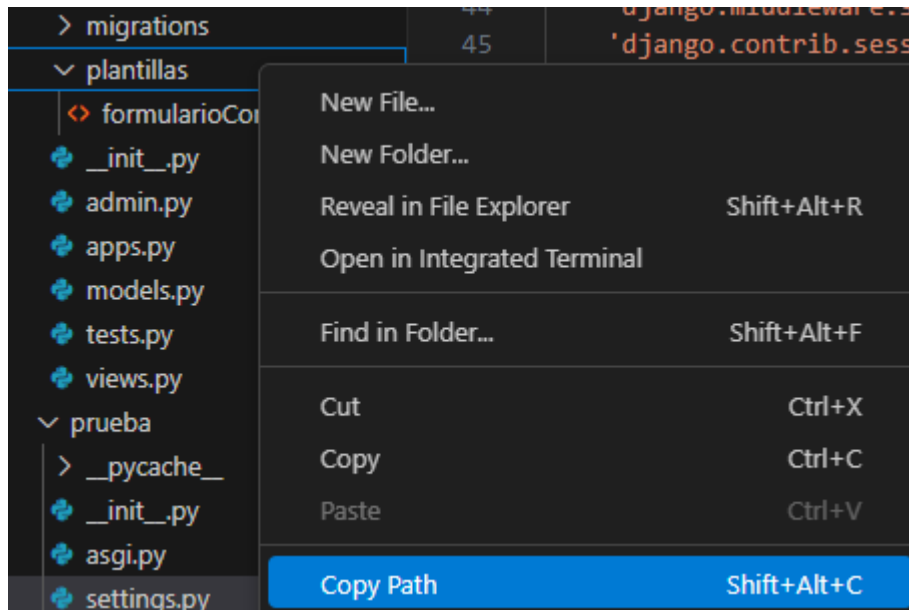
Ahora abrimos el archivo de **views.py** para agregar la ruta del formulario de contacto.



Abrimos el archivo de **setting.py** y buscamos el apartado de **TEMPLATES**.



Vamos a copiar la ruta en donde se encuentra la carpeta de plantillas, damos con clic derecho en la carpeta plantillas y copiamos la ruta de acceso.



Ahora en el apartado de TEMPLATES en DIRS, pegamos la ruta que habíamos copiado.

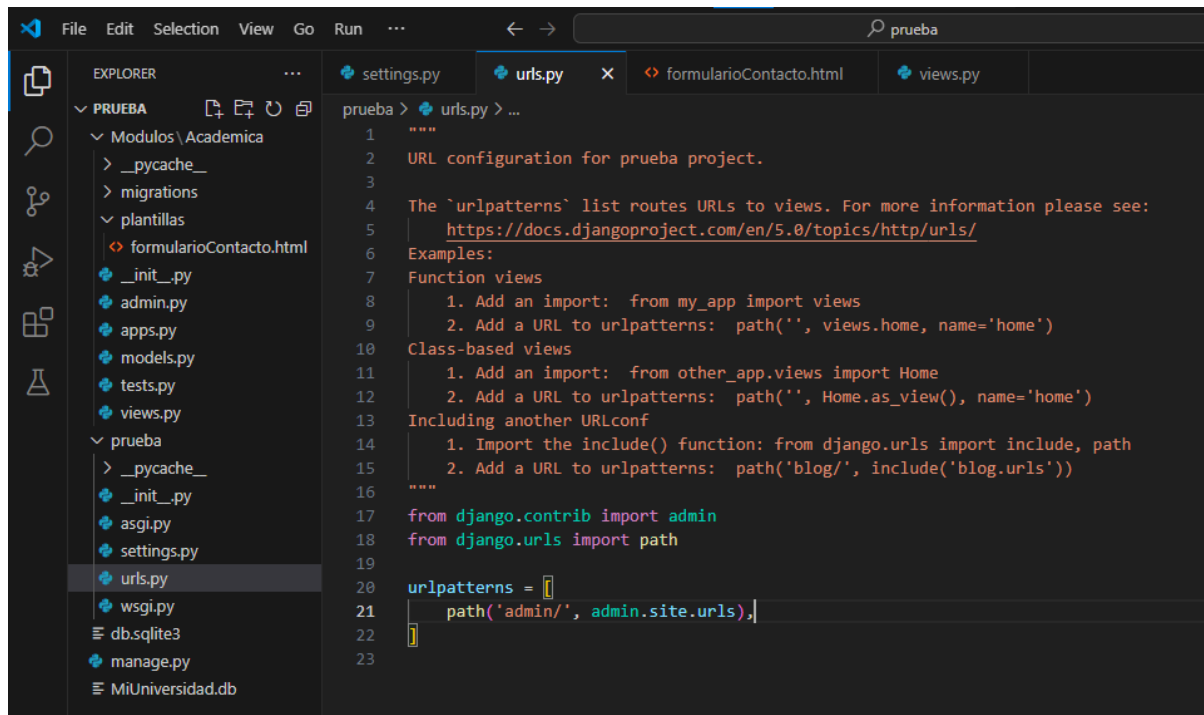
```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': ['C:\Users\Alexander\Desktop\django\prueba\Modulos\Academica\plantillas'], #Ruta  
        'APP_DIRS': True,  
        'OPTIONS': {
```

Ahora solo debemos cambiar el \ por /

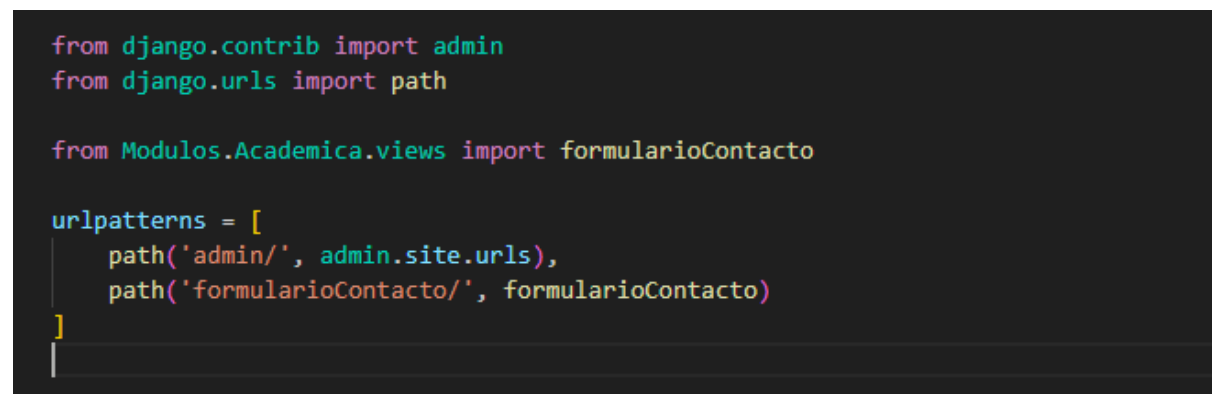
```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': ['C:/Users/Alexander/Desktop/django/prueba/Modulos/Academica/plantillas'], #Ruta  
        'APP_DIRS': True,  
        'OPTIONS': {
```

Nos debe quedar de esta manera.

Procedemos a abrir el archivo de **urls.py**.



Aquí vamos a agregar la url para acceder al formulario contacto, además de la vista de este. También se agrega el **import** del formulario contacto.



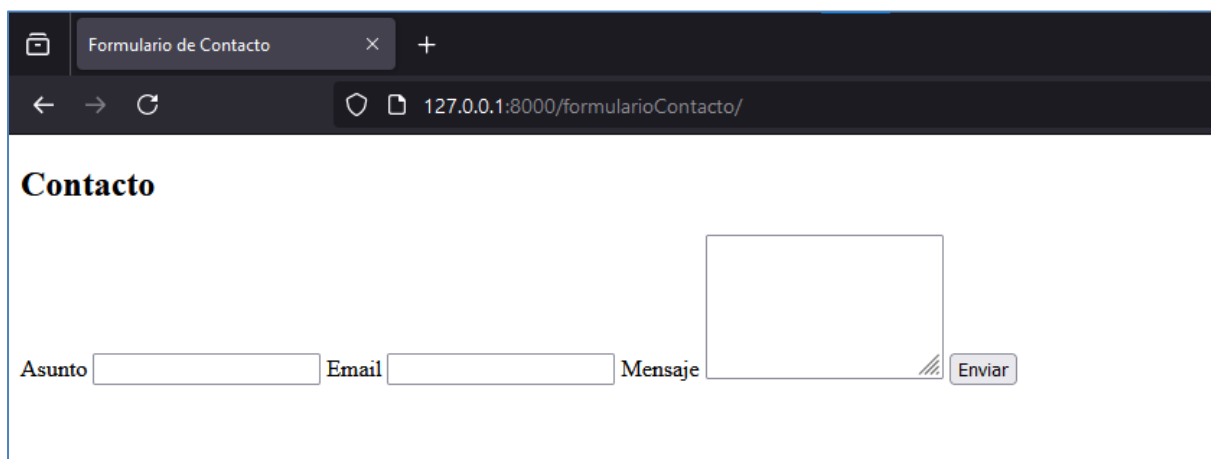
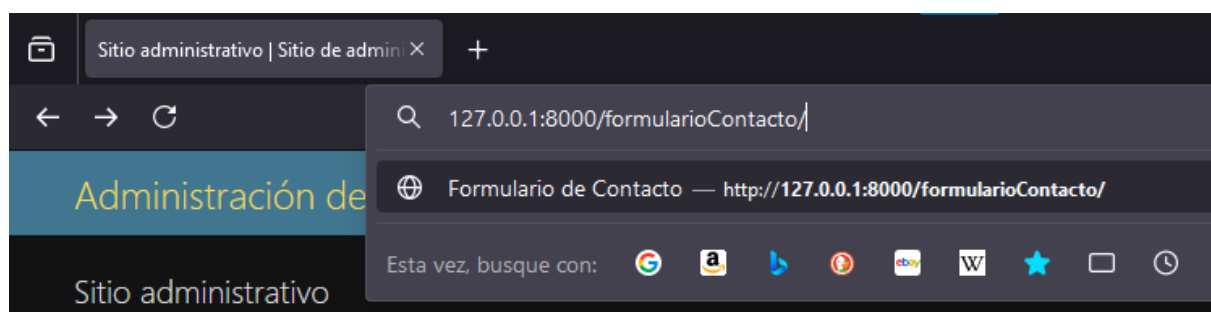
Arrancamos nuevamente el servidor.

```
Símbolo del sistema - python C:\Users\Alexander\Desktop\django\prueba\manage.py runserver

C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prue
ba\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 23, 2024 - 09:35:44
Django version 5.0.6, using settings 'prueba.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

En el navegador accedemos al formulario contacto



Se nos visualiza de esta manera, pero también podemos agregar CSS para tener un mejor diseño del formulario, para ello, nos dirigimos al formulario contacto y debajo de `</html>` agregamos los estilos.

```
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
  }

  .contact-form-container {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    max-width: 400px;
    width: 100%;
  }

  .contact-form-container h2 {
    margin-bottom: 20px;
    font-size: 24px;
    text-align: center;
  }

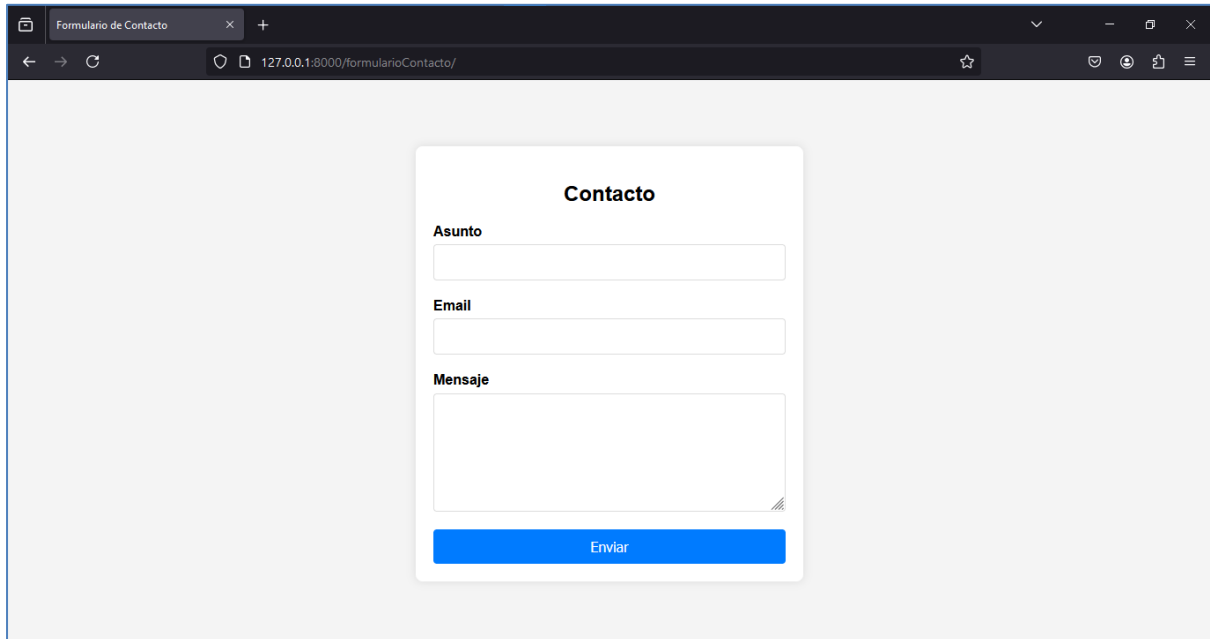
  .contact-form-container label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
  }

  .contact-form-container input[type="text"],
  .contact-form-container input[type="email"],
  .contact-form-container textarea {
    width: 100%;
    padding: 10px;
    margin-bottom: 20px;
    border: 1px solid #ddd;
    border-radius: 4px;
    box-sizing: border-box;
    font-size: 16px;
  }

  .contact-form-container button {
    width: 100%;
    padding: 10px;
    background-color: #007BFF;
    color: #fff;
    border: none;
    border-radius: 4px;
    font-size: 16px;
    cursor: pointer;
  }

  .contact-form-container button:hover {
    background-color: #0056b3;
  }
</style>
```

Ahora solo recargamos la pagina nuevamente y nos debe mostrar un mejor aspecto.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/formularioContacto/'. The page content is a contact form titled 'Contacto'. The form has three input fields: 'Asunto' (Subject), 'Email', and 'Mensaje' (Message). Below the 'Mensaje' field is a blue button labeled 'Enviar' (Send). The form is centered on a light gray background.

Nos situamos en el archivo de settings.py y bajamos hasta el final.

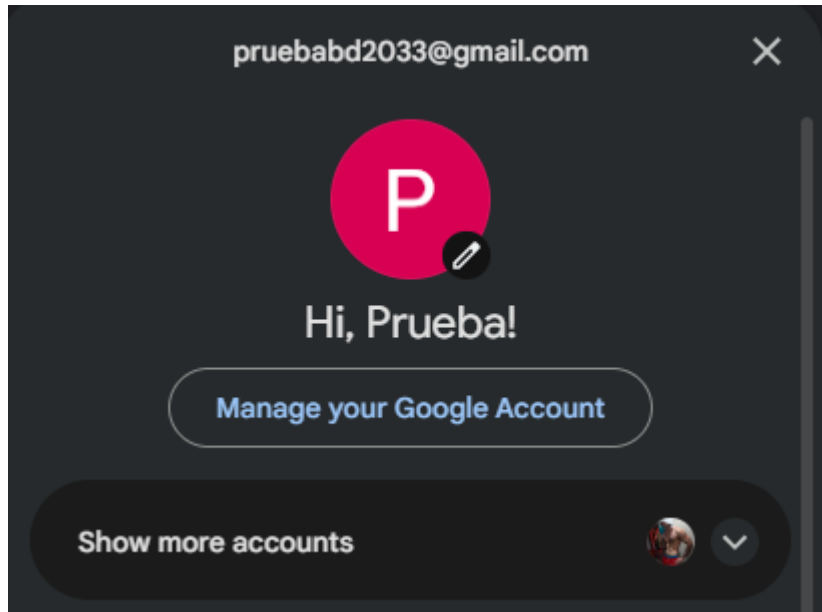
```
settings.py x  urls.py  formularioContacto.html  views.py
prueba > settings.py > ...
116  # Static files (CSS, JavaScript, Images)
117  # https://docs.djangoproject.com/en/5.0/howto/static-files/
118
119  STATIC_URL = 'static/'
120
121  # Default primary key field type
122  # https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field
123
124  DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
125
```

Y procedemos agregar las siguientes líneas para la conexión del protocolo de correo smtp.

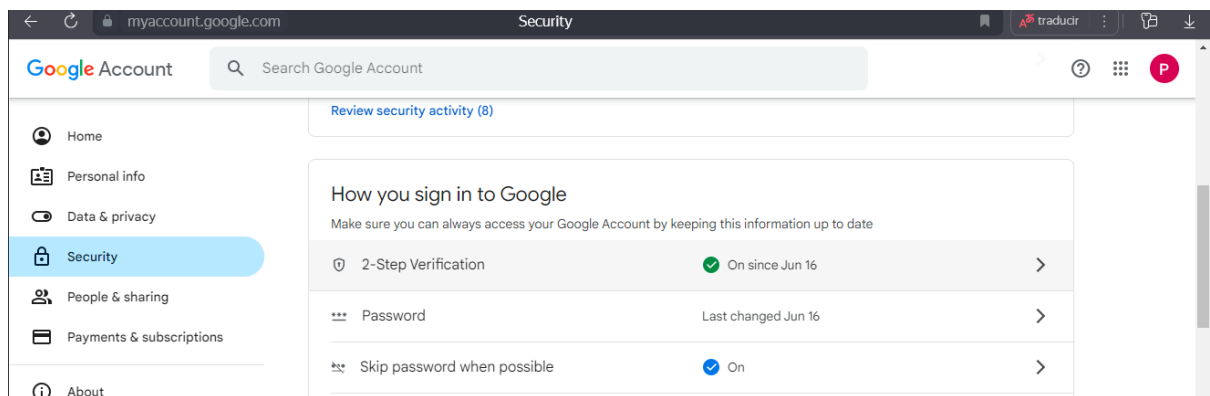
```
123
124  DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
125  💡
126  EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
127  EMAIL_HOST = 'smtp.gmail.com'
128  EMAIL_USE_TLS = True
129  EMAIL_PORT = 587
130  EMAIL_USE_SSL = False
131  EMAIL_HOST_USER = 'pruebabd2033@gmail.com'
132  EMAIL_HOST_PASSWORD = 'vxlw wuvl thjk fona'
```


En el apartado de:

EMAIL_HOST_USER deberán agregar un correo de Gmail. Por medidas de seguridad se recomienda usar un correo nuevo. Así que se dirigen a Google y crean un correo nuevo.

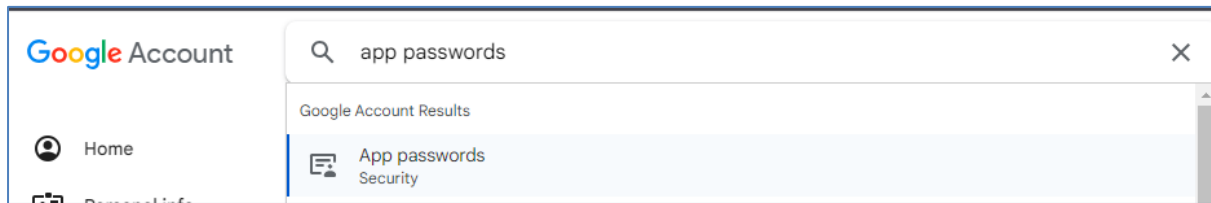


Una vez creado, nos vamos a dirigir a la configuración de la cuenta.

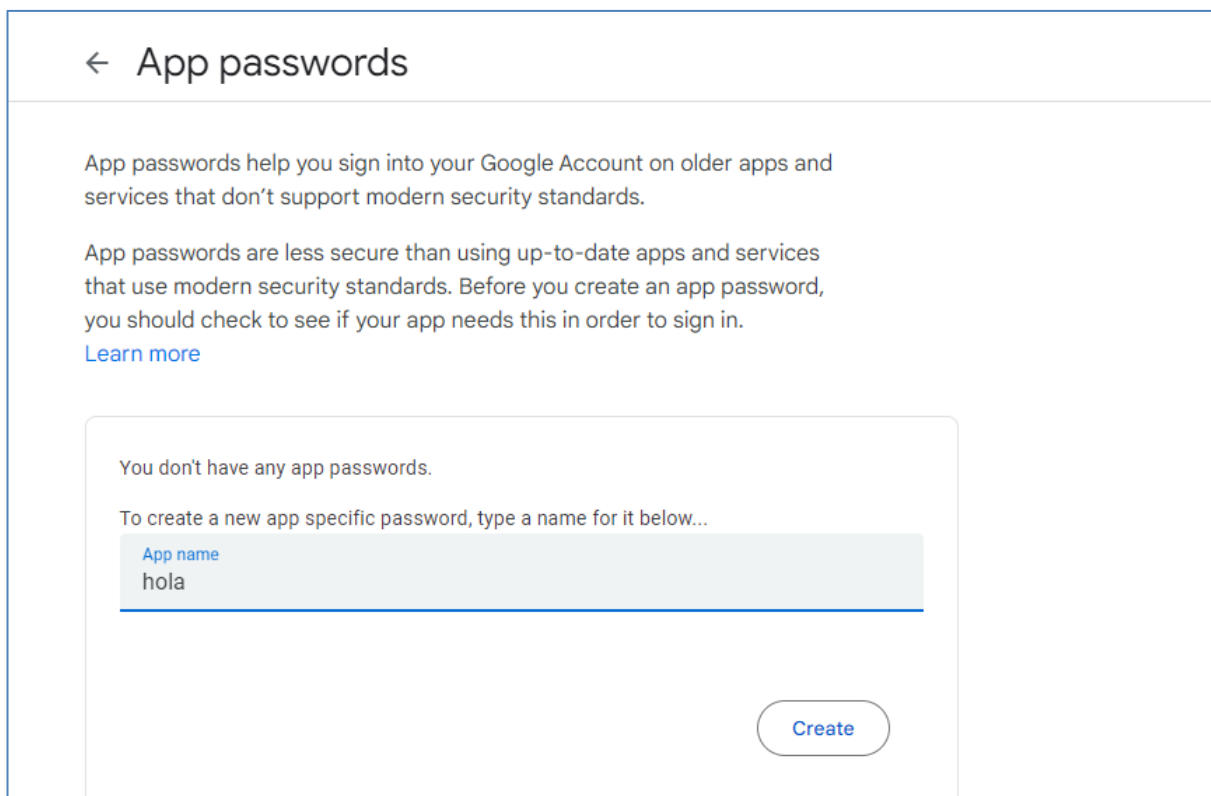


En el apartado de seguridad, vamos a activar la verificación de dos pasos, allí solo siguen las indicaciones hasta completar la verificación y una vez completada debe aparecer de color verde (es muy importante realizar esta acción, para que Google nos permita generar la contraseña para apps).

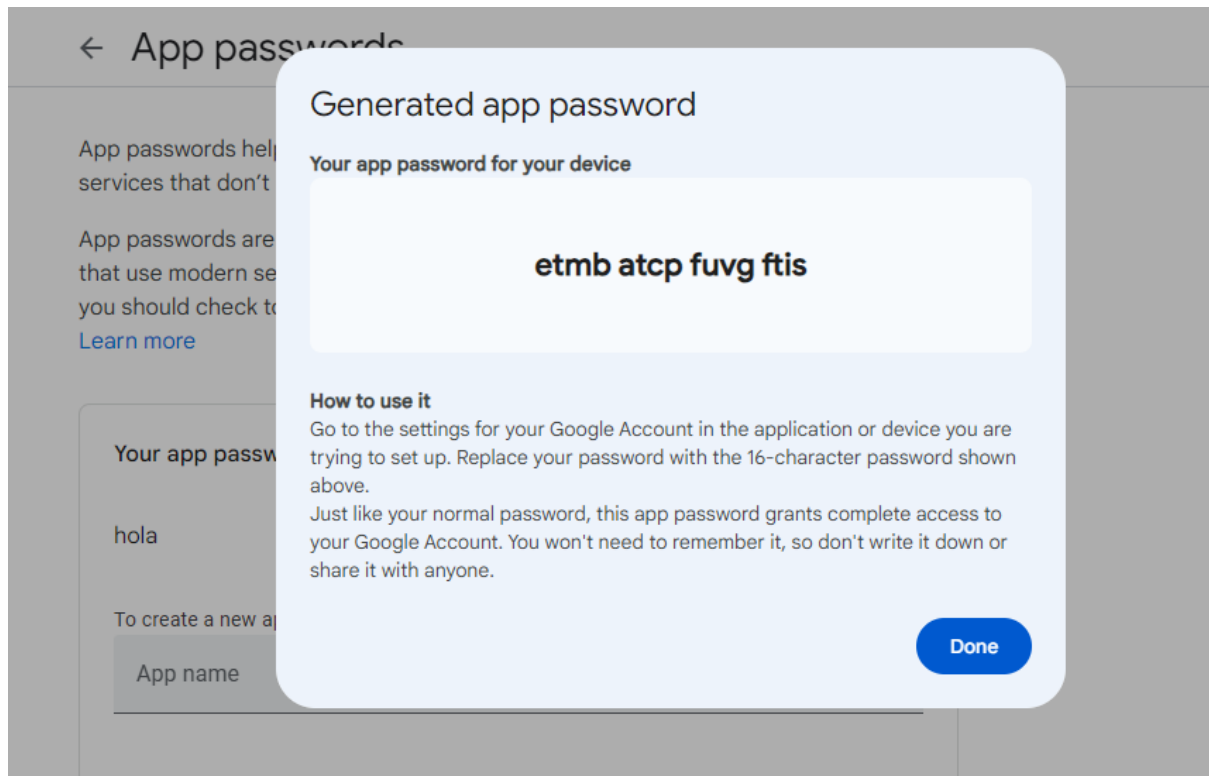
Ahora buscamos en la lupa **app passwords** e ingresamos, cabe recalcar que si no completaron la verificación de dos pasos este apartado nos les aparecerá.



Aquí solo agregamos un nombre para nuestra app, puede ser cualquiera. Y damos clic en créate.



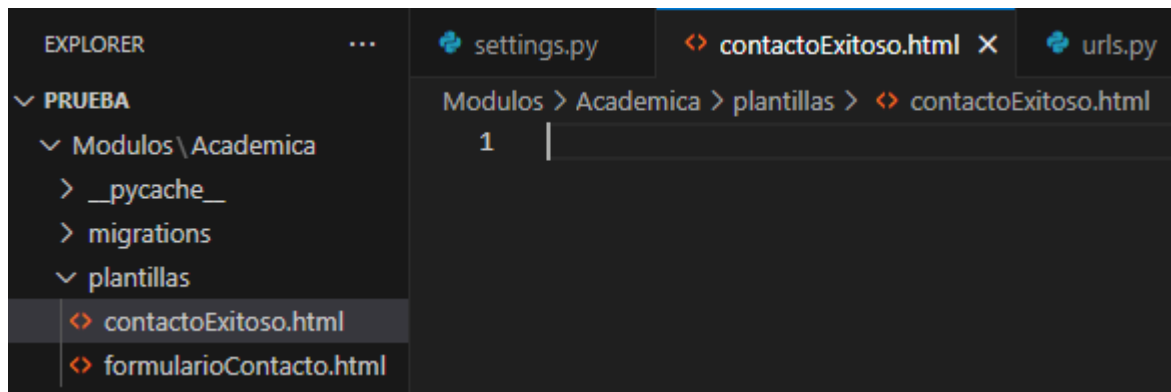
Aquí se nos va a generar una serie de caracteres el cual es la contraseña de la app de nuestra cuenta de Google, tan solo lo copiamos.



Este código que se nos genero lo vamos a pegar en: EMAIL_HOST_PASSWORD

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_USE_SSL = False
EMAIL_HOST_USER = 'pruebabd2033@gmail.com'
EMAIL_HOST_PASSWORD = 'etmb atcp fuvg ftis'
```

Ahora en la carpeta de plantillas, vamos a crear un nuevo documento html con el nombre de **contactoExitoso.html**.



```
settings.py  contactoExitoso.html  urls.py  formularioContacto.html  views.py

Modulos > Academica > plantillas > contactoExitoso.html > html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Mensaje Enviado</title>
7
8  </head>
9  <body>
10     <div class="container">
11         <h1>Mensaje de Contacto Enviado con Éxito</h1>
12         <p>Gracias por contactarnos. Nos pondremos en contacto contigo lo antes posible.</p>
13         <a href="#" class="button">Volver</a>
14     </div>
15 </body>
16 </html>
```

Devuelta en **views.py** importamos settings y send_mail.

```
settings.py  contactoExitoso.html  views.py  urls.py

Modulos > Academica > views.py > ...
1  from django.shortcuts import render
2
3  from django.core.mail import send_mail
4  from django.conf import settings
5
```

Agregamos la función de contactar. Y dentro de esta establecemos el método y las variables que van a obtener la información de las cajas de textos del formulario. Así como la dirección de correo a la cual vamos a enviar un mensaje (**email_para**). Caso de que se haya enviado correctamente nos muestre el html de contacto exitoso.

```
settings.py | contactoExitoso.html | views.py | urls.py
Modulos > Academica > views.py > contactar
1  from django.shortcuts import render
2
3  from django.core.mail import send_mail
4  from django.conf import settings
5
6  # Create your views here.
7
8  def formularioContacto(request):
9      return render(request, "formularioContacto.html")
10
11  def contactar(request):
12      if request.method == "POST":
13          asunto = request.POST["subject"]
14          mensaje = request.POST["message"] + " / Email: " + request.POST["email"]
15          email_desde = settings.EMAIL_HOST_USER
16          email_para = ["oliverchiriboga@gmail.com"]
17          send_mail(asunto, mensaje, email_desde, email_para, fail_silently=False)
18          return render(request, "contactoExitoso.html")
19
20      return render(request, "formularioContacto.html")
21
```

Ahora en el archivo de **urls.py** vamos a agregar la ruta de contactar y de su import.

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

from Modulos.Academica.views import formularioContacto, contactar

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

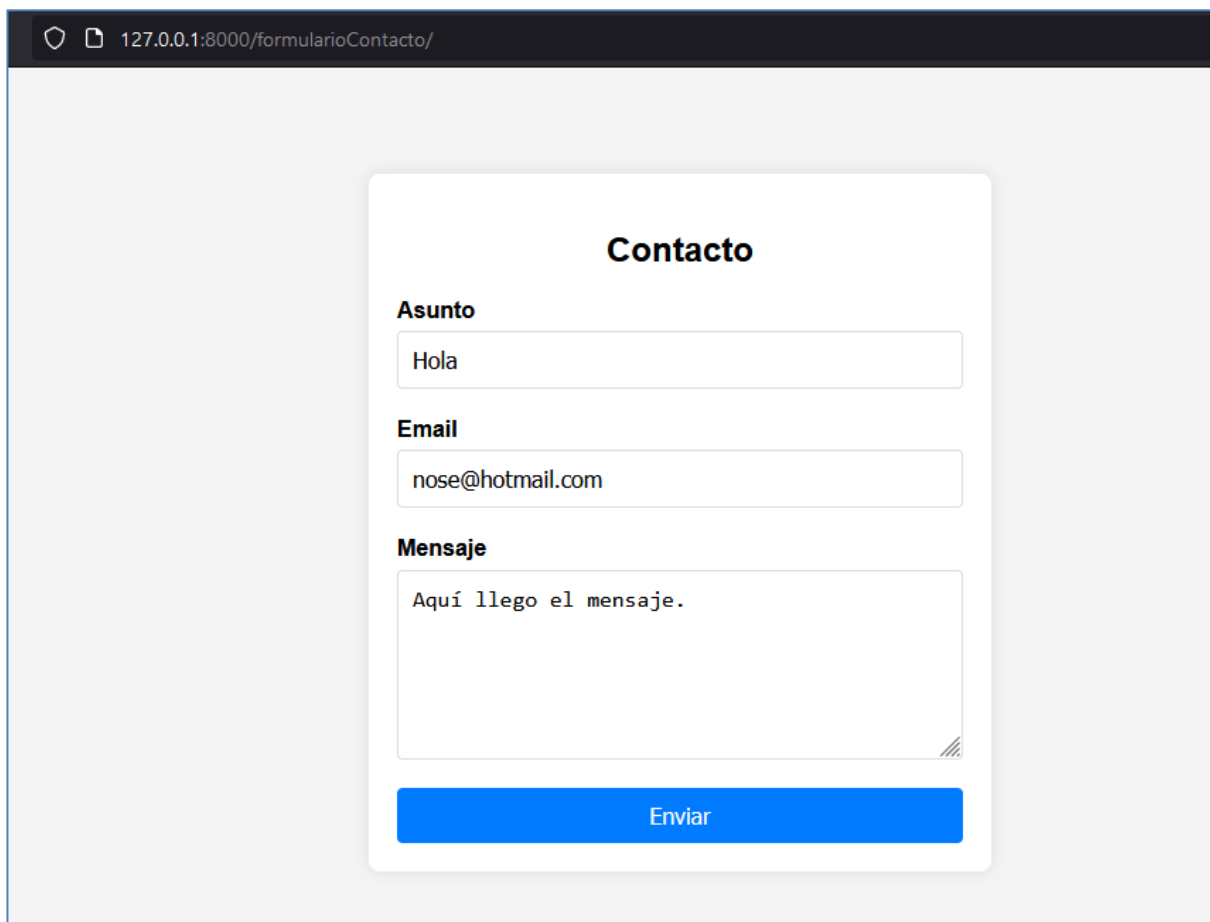
Finalmente detenemos el servidor con **control + c** y lo volvemos a iniciar.

```
Símbolo del sistema - python C:\Users\Alexander\Desktop\django\prueba\manage.py runserver

C:\Users\Alexander\Desktop\django\prueba>python C:\Users\Alexander\Desktop\django\prue
ba\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 23, 2024 - 10:36:15
Django version 5.0.6, using settings 'prueba.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Llenamos el formulario y enviamos el mensaje.



127.0.0.1:8000/formularioContacto/

Contacto

Asunto

Email

Mensaje

Enviar

Si todo salió correctamente se nos mostrara de esta manera.



Si revisamos la bandeja de entrada del correo que establecimos en **email_para**, veremos que nos habrá llegado el mensaje que enviamos desde el formulario.

