TRABAJANDO CON DJANGO

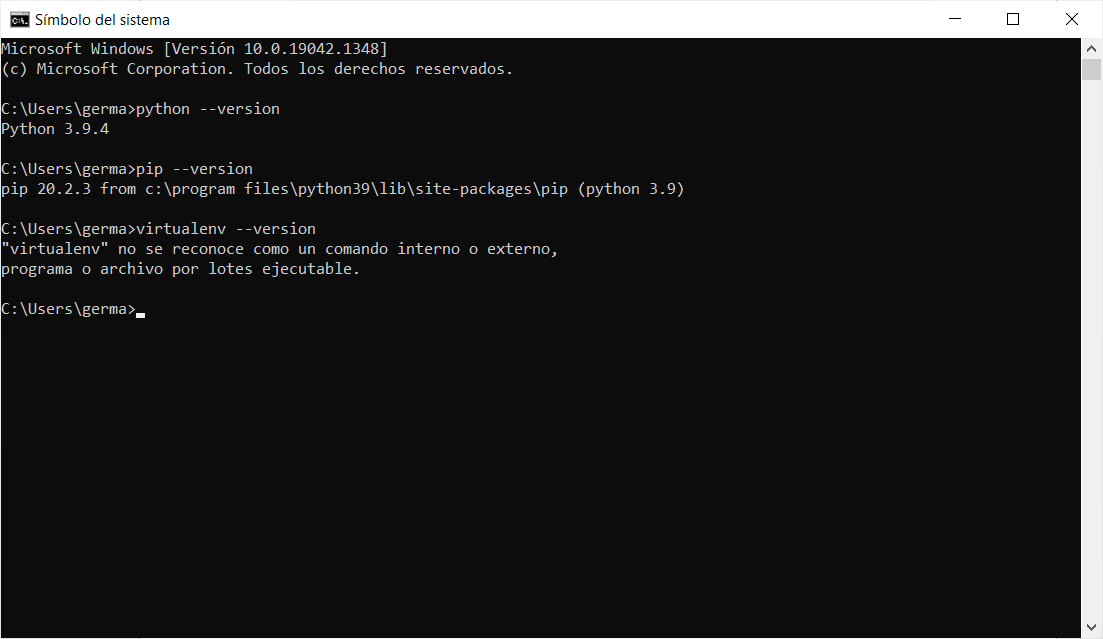
Documentación

Capturas de pantalla: German Gutiérrez Machicado

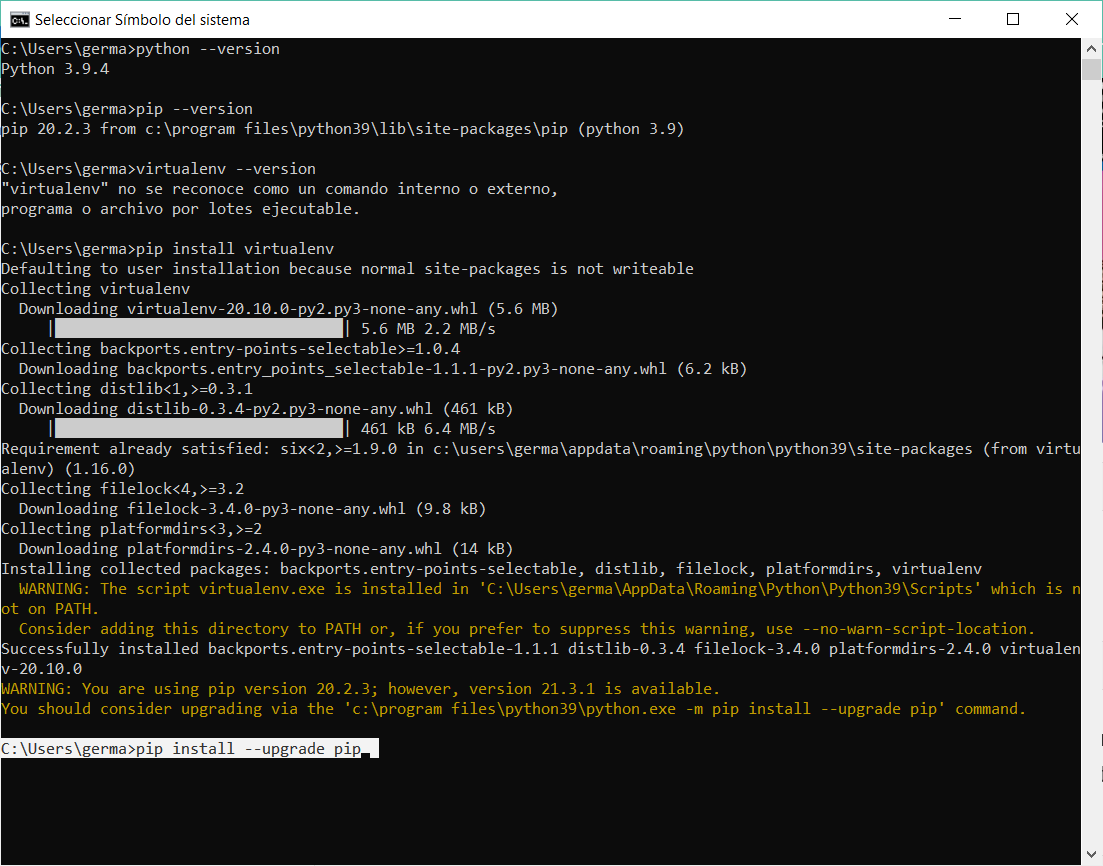
Diciembre de 2021

1. [Establecer el entorno de Django](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#establecer-el-entorno-de-django)
2. [Configurar proyecto Django](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#configurar-proyecto-django)
3. [Probar el servidor](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#probar-el-servidor)
4. [Actualizar el archivo settings.py](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#actualizar-el-archivo-settings.py)
5. [Crear un modelo](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#crear-un-modelo)
6. [Crear un formulario para cargar archivos](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#crear-un-formulario-para-cargar-archivos)
7. [Agregar vistas](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#agregar-vistas)
8. [Configurar el archivo urls.py](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#configurar-el-archivo-urls.py)
9. [Pruebas](https://www.delftstack.com/es/howto/django/django-upload-file-or-image/#pruebas)
10. **Establecer el entorno de Django**

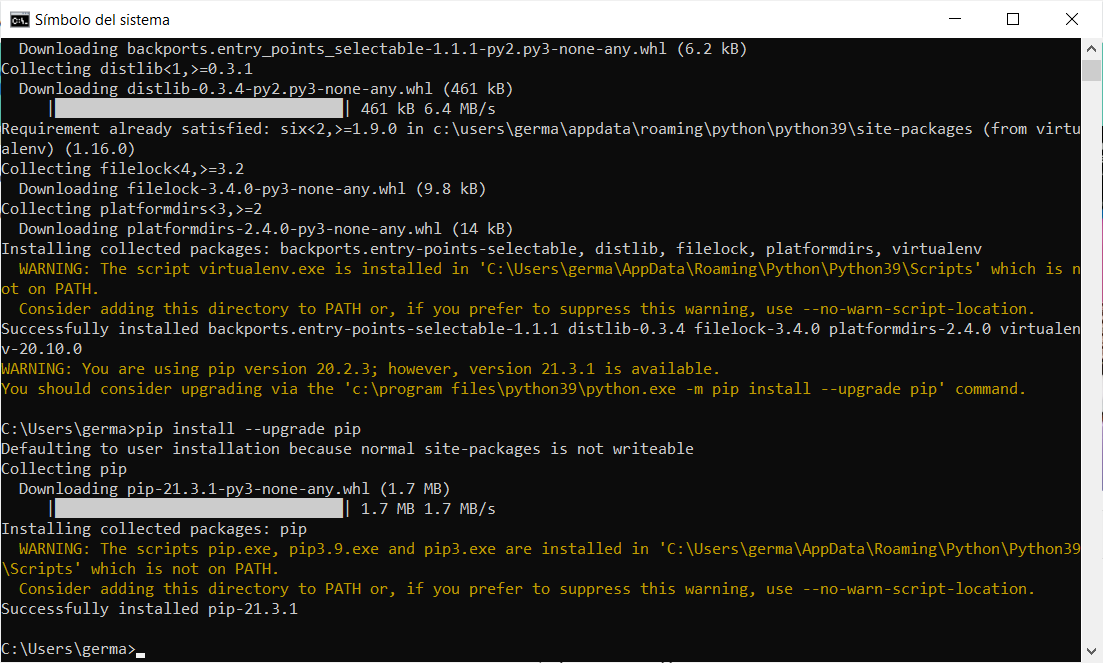
Verificamos la versión de nuestro “python” y nuestro “pip” si no están los descargamos



Establecemos un entorno virtual. Installamos “virtualenv”

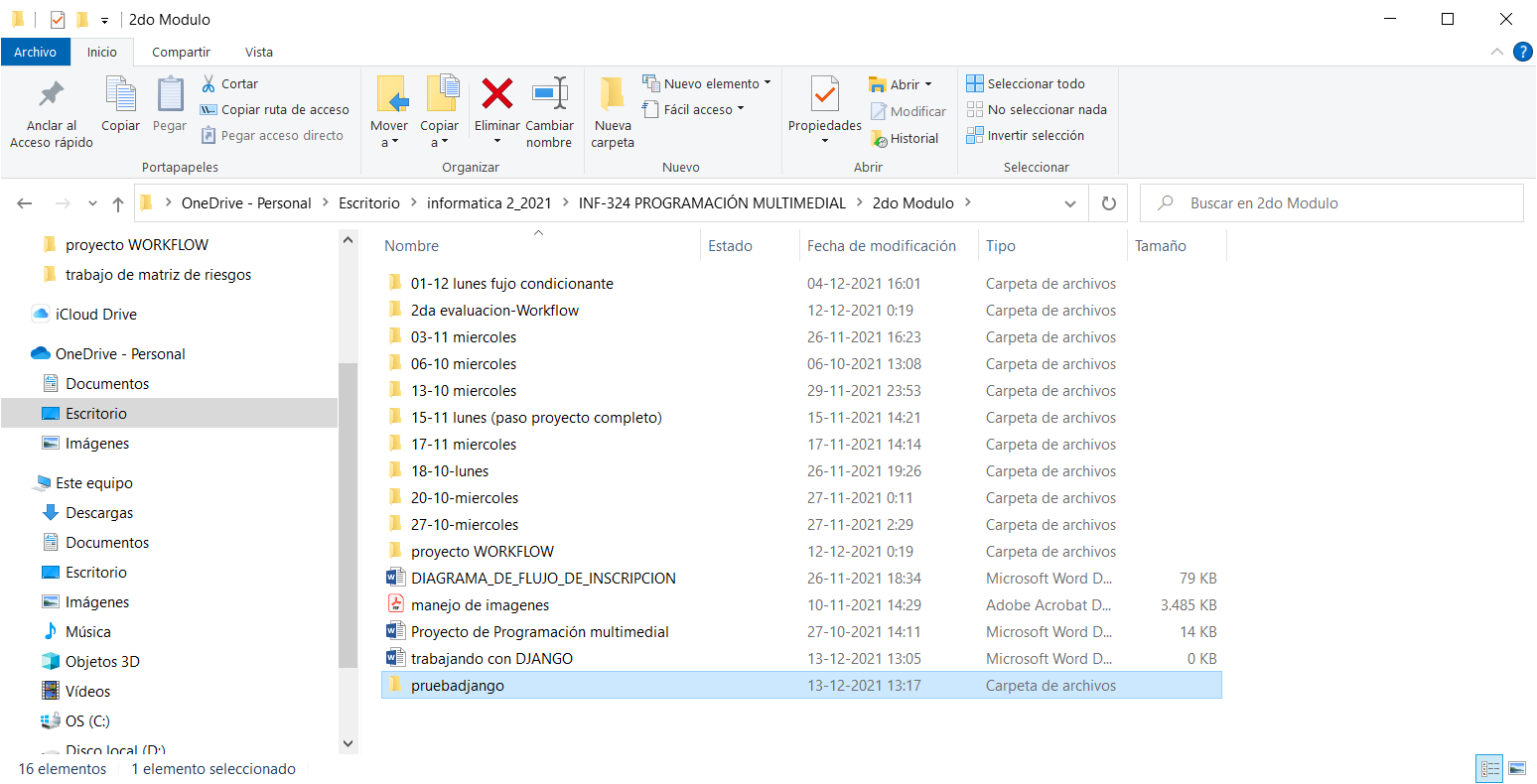


Nos pide actualizar. Lo actualizamos.



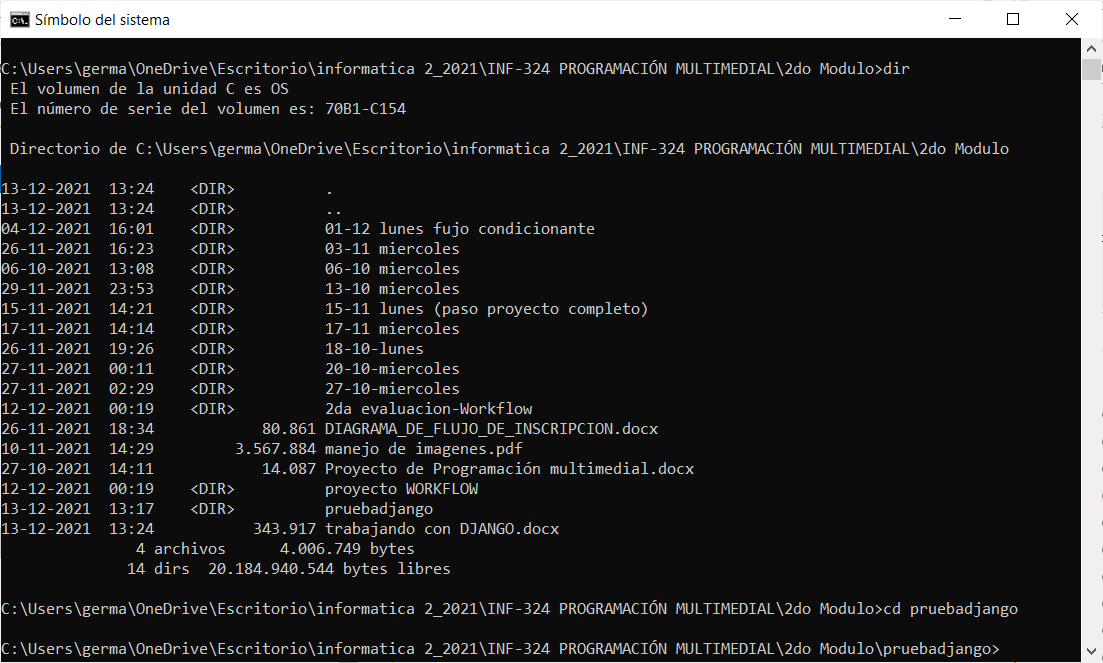
Se actualizó satisfactoriamente

Creamos una carpeta donde trabajaremos

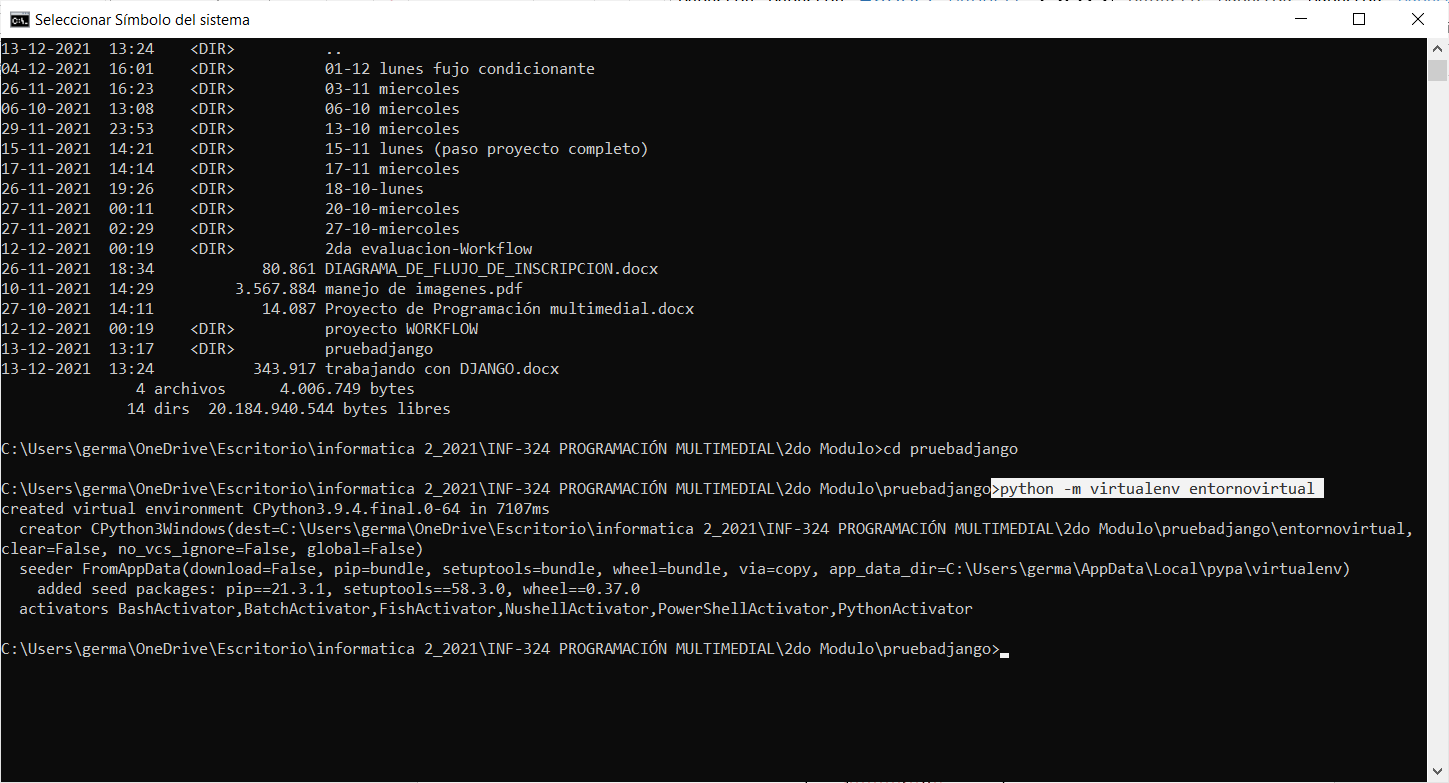


Y dentro de esta carpeta creamos nuestro entorno virtual con el comando:

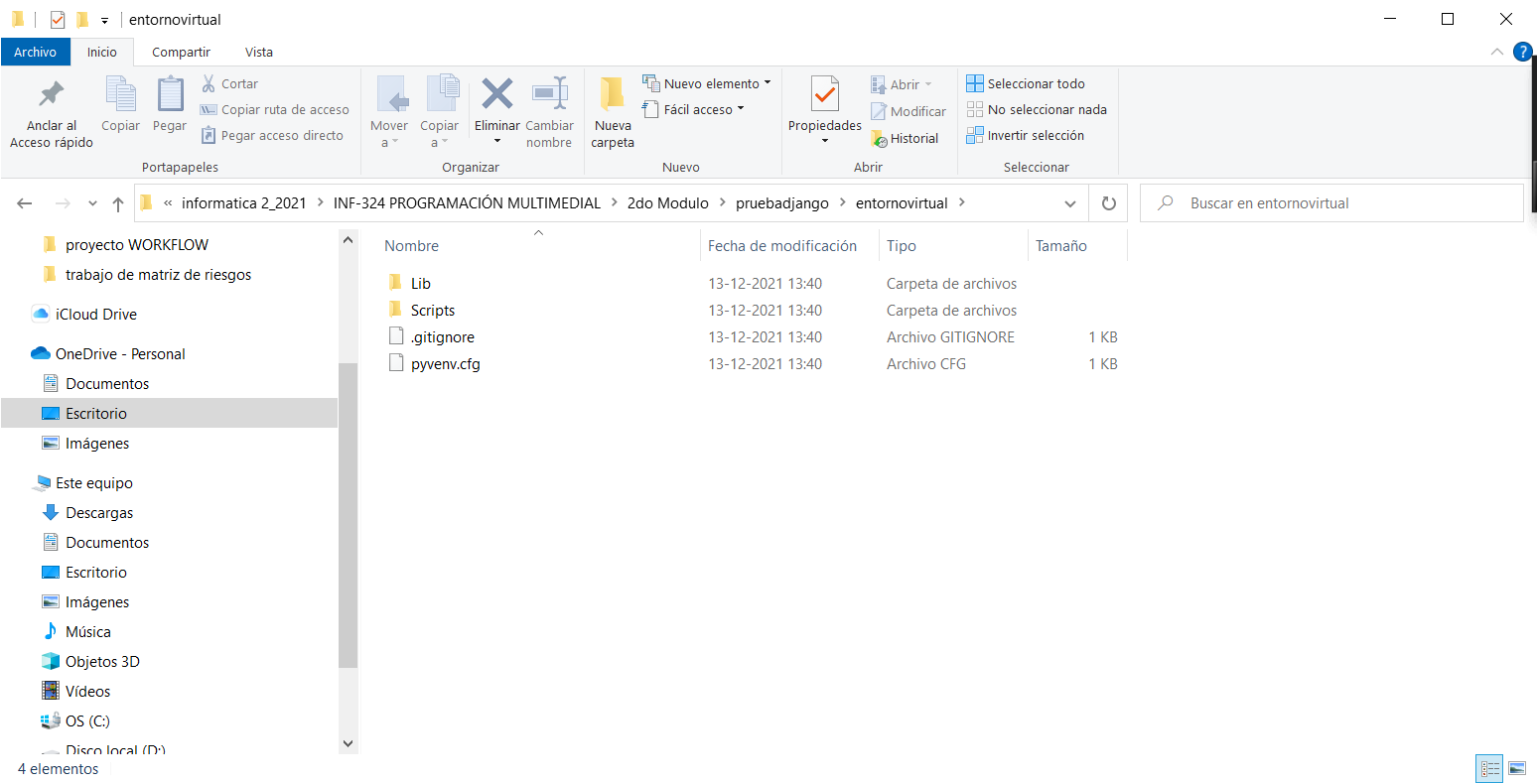
**>Python –m virtualenv nombre** (crea la carpeta “nombre” donde estará nuestro entorno. No hay que modificar esta carpeta)



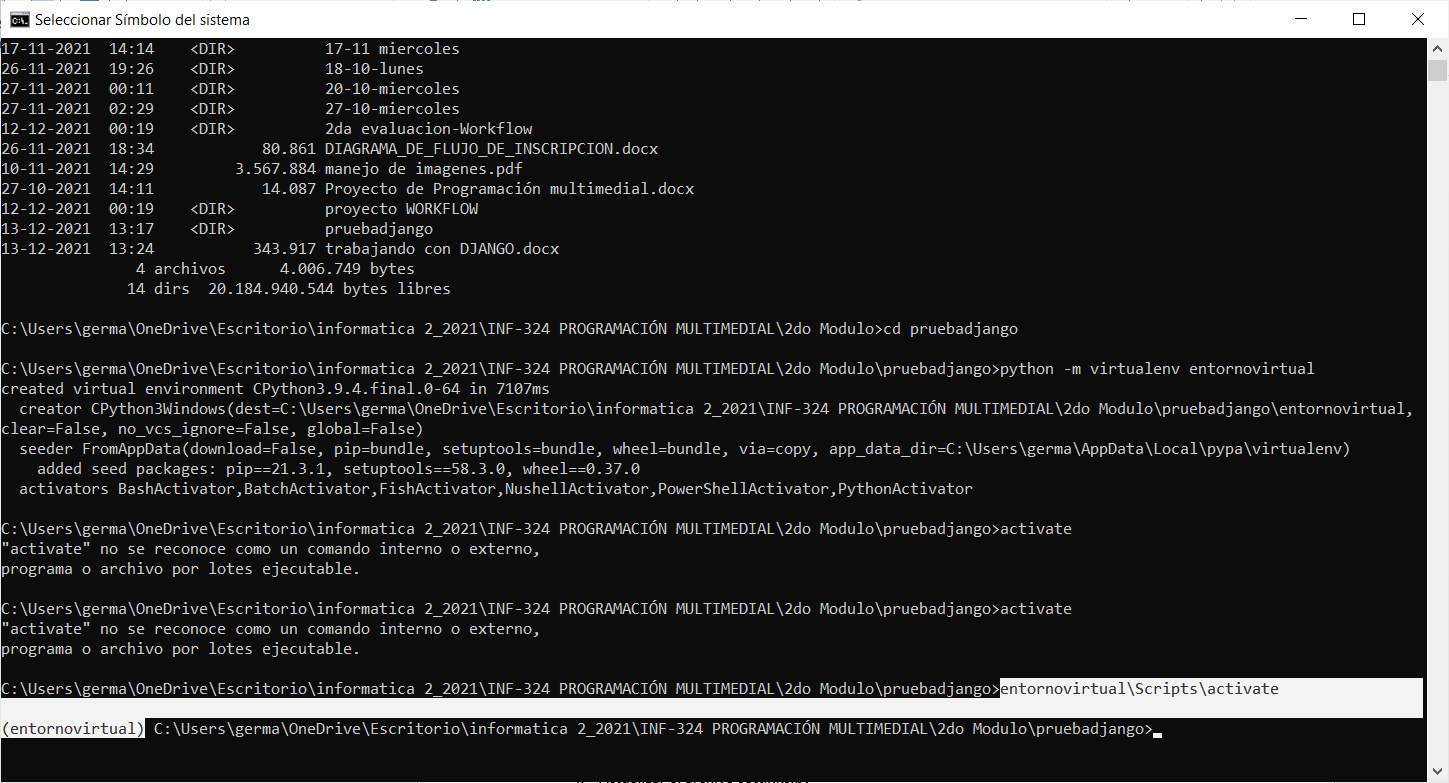
Estamos dentro de nuestra carpeta “pruebadjango”. Y ahí ejecutamos el comando citado anteriormente.



Se creó nuestra carpeta “entornovirtual” como se puede verificar abajo. No hay que tocar nada en estas carpetas.

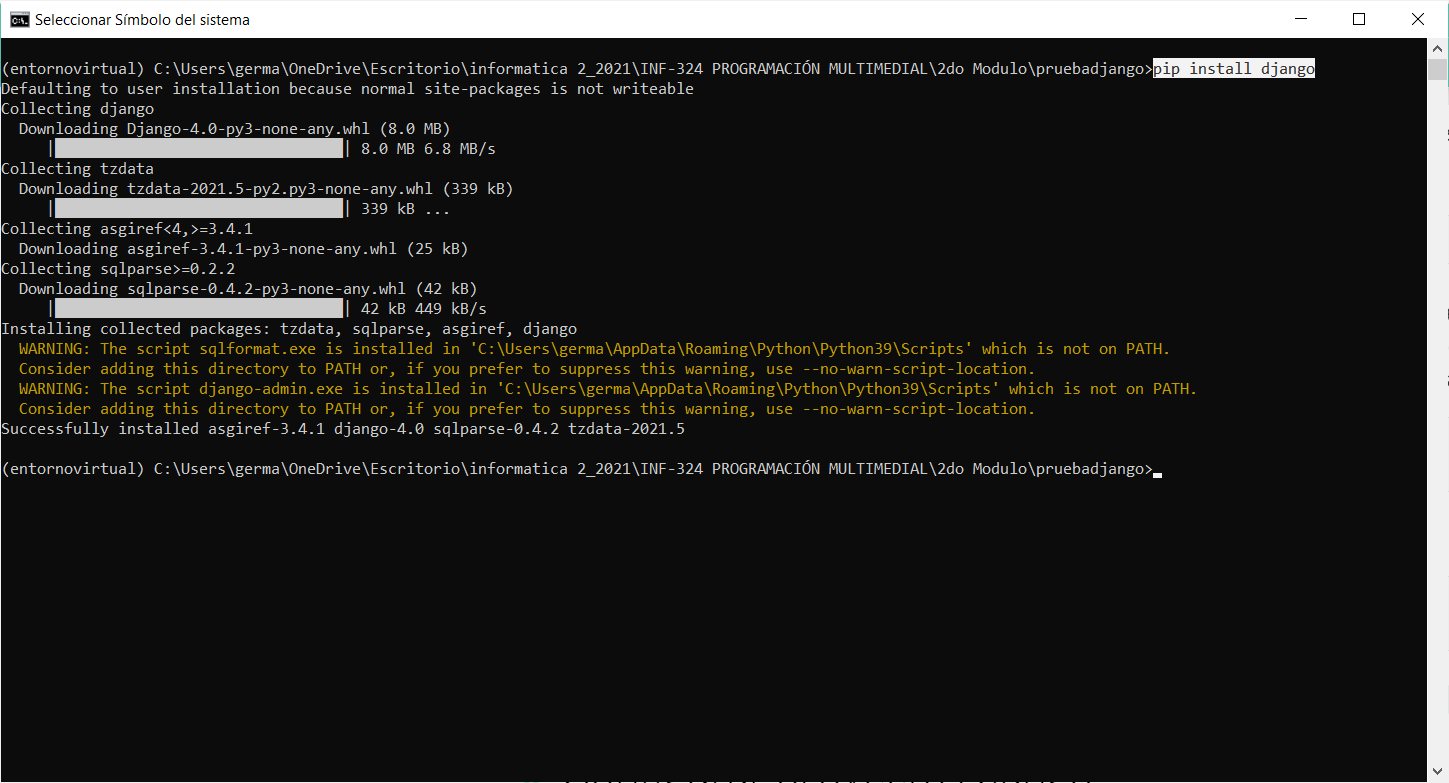


Ahora activamos nuestra carpeta “pruebadjango” con el comando: entornovirtual\Scripts\activate



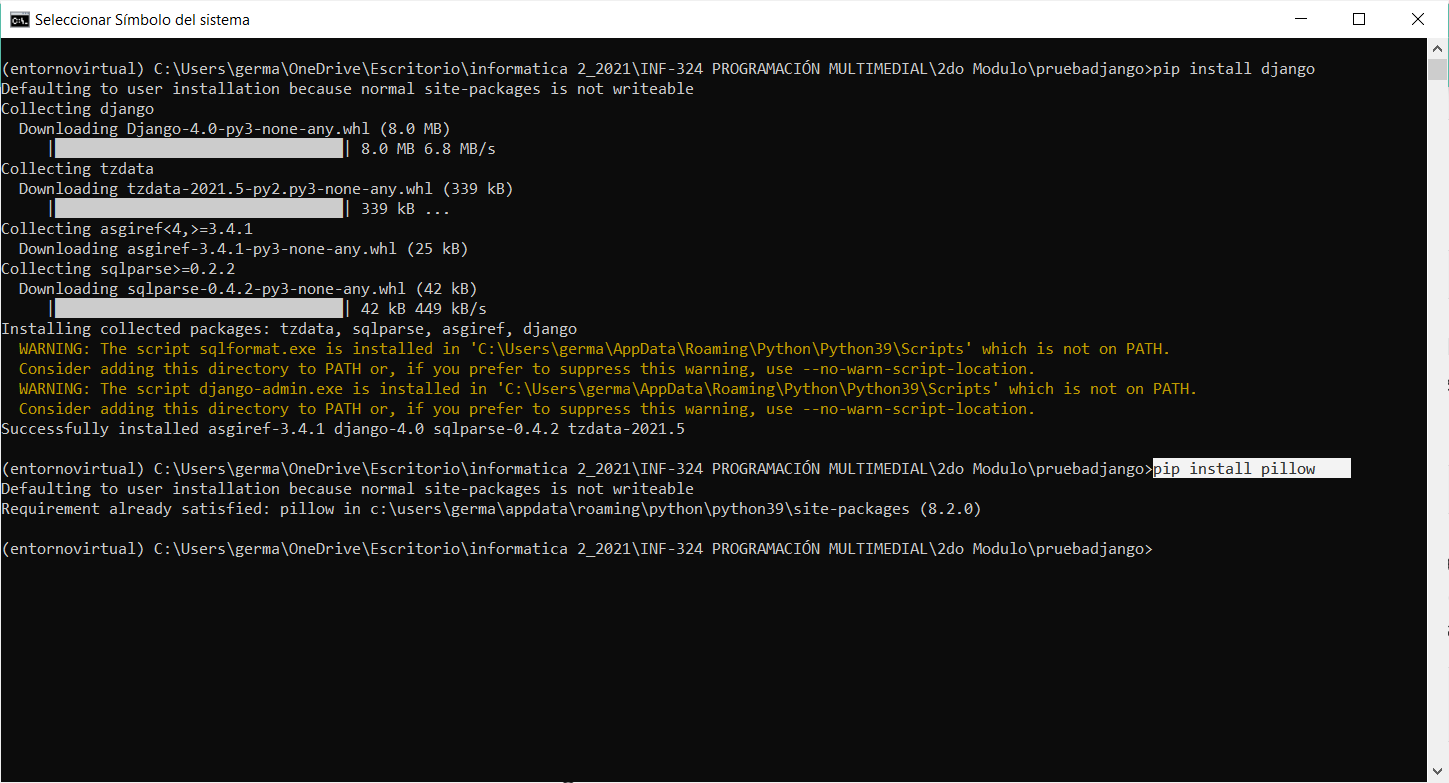
Ya estamos en el entorno virtual. Y aquí dentro instalamos django con el comando:

>pip install django



Instalamos además el paquete “pillow” para manejar las imágenes con el comando:

>pip install pillow

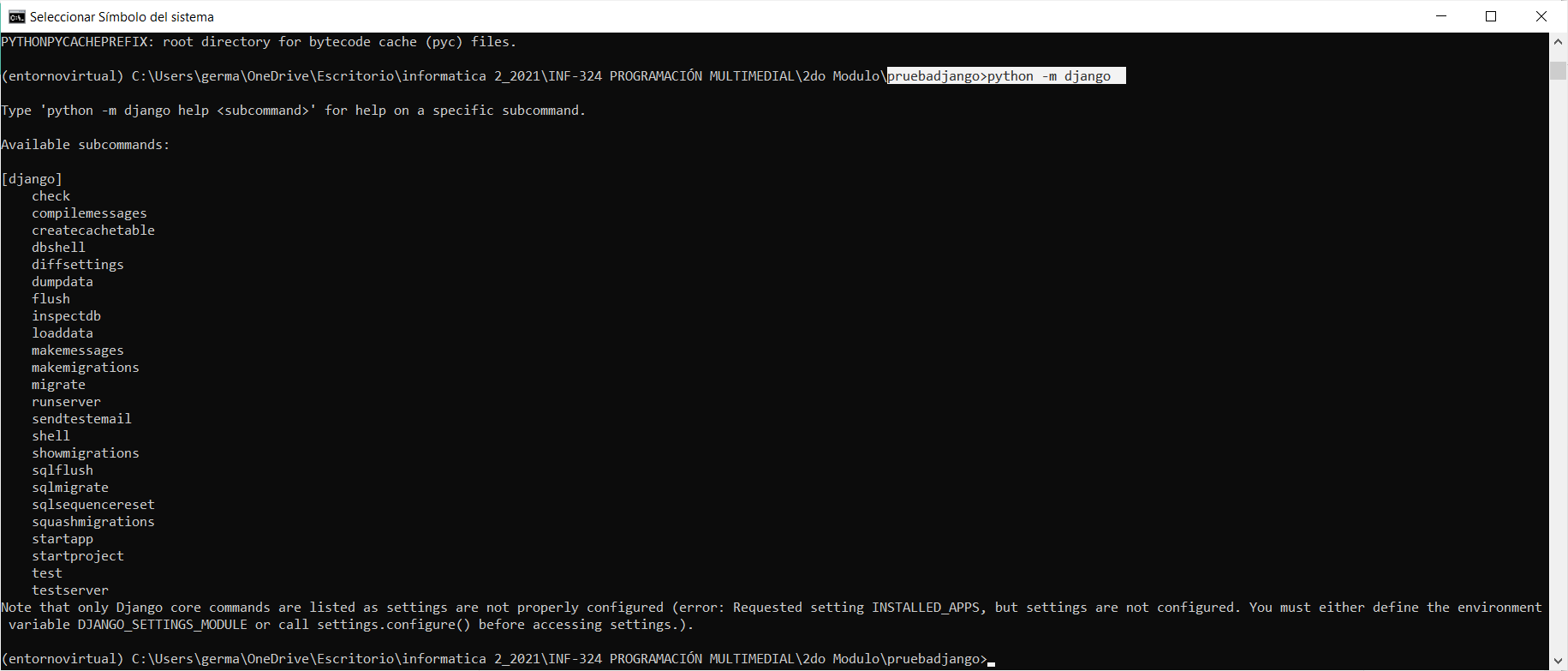


Con todo esto ya hemos establecido el entorno de DJANGO, ahora pasamos a la fase de configuración de proyectos.

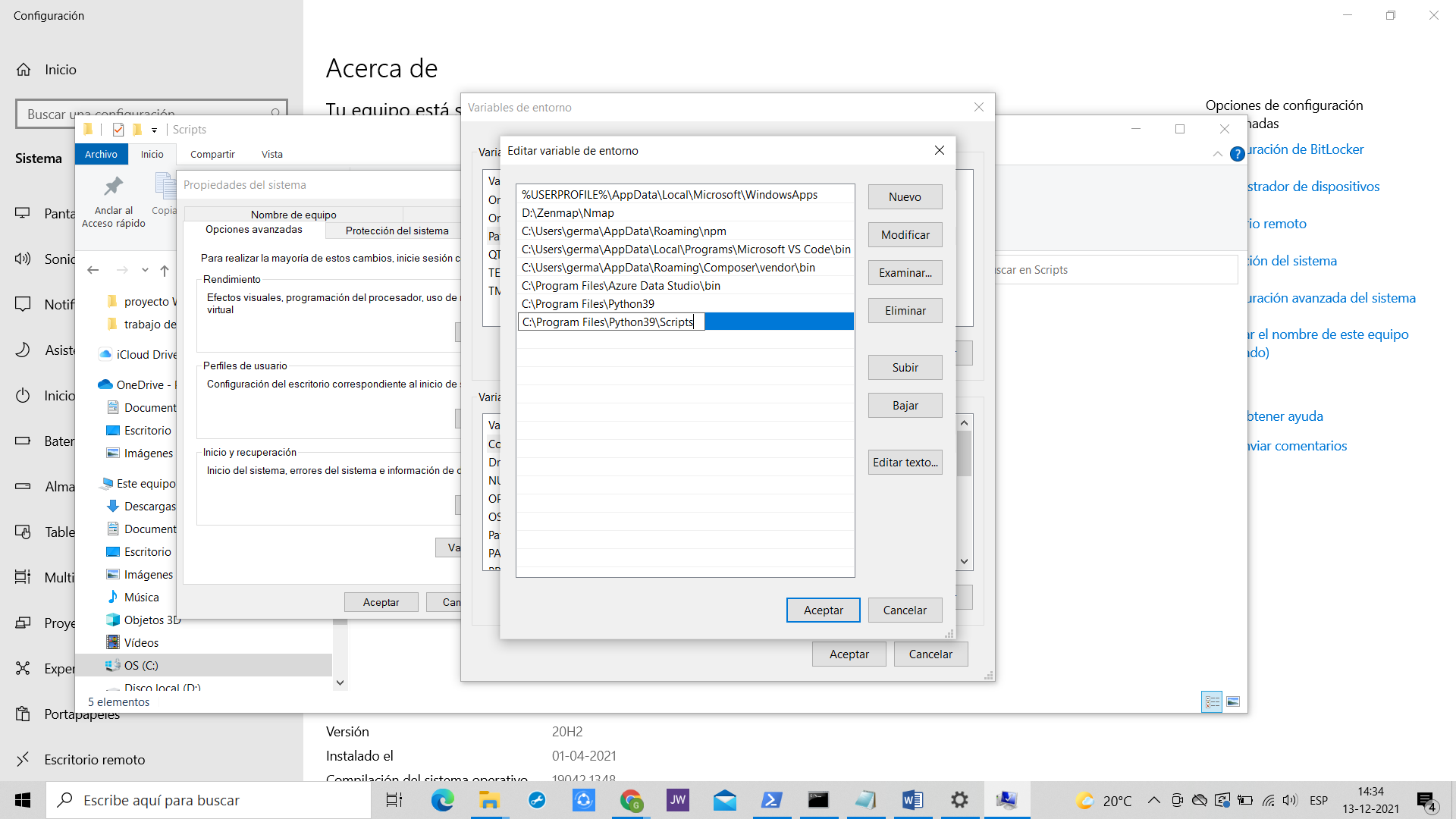
1. **Configurar proyecto Django**

Así como esta en la guía de DJANGO o como está en el video que uso como guía no me corre. El comando: **django-admin** , que debería mostrarme los comandos que puedo ejecutar con DJANGO no me corre. Pero encontré la forma de hacerlo, es el comando:

>**Python –m django**



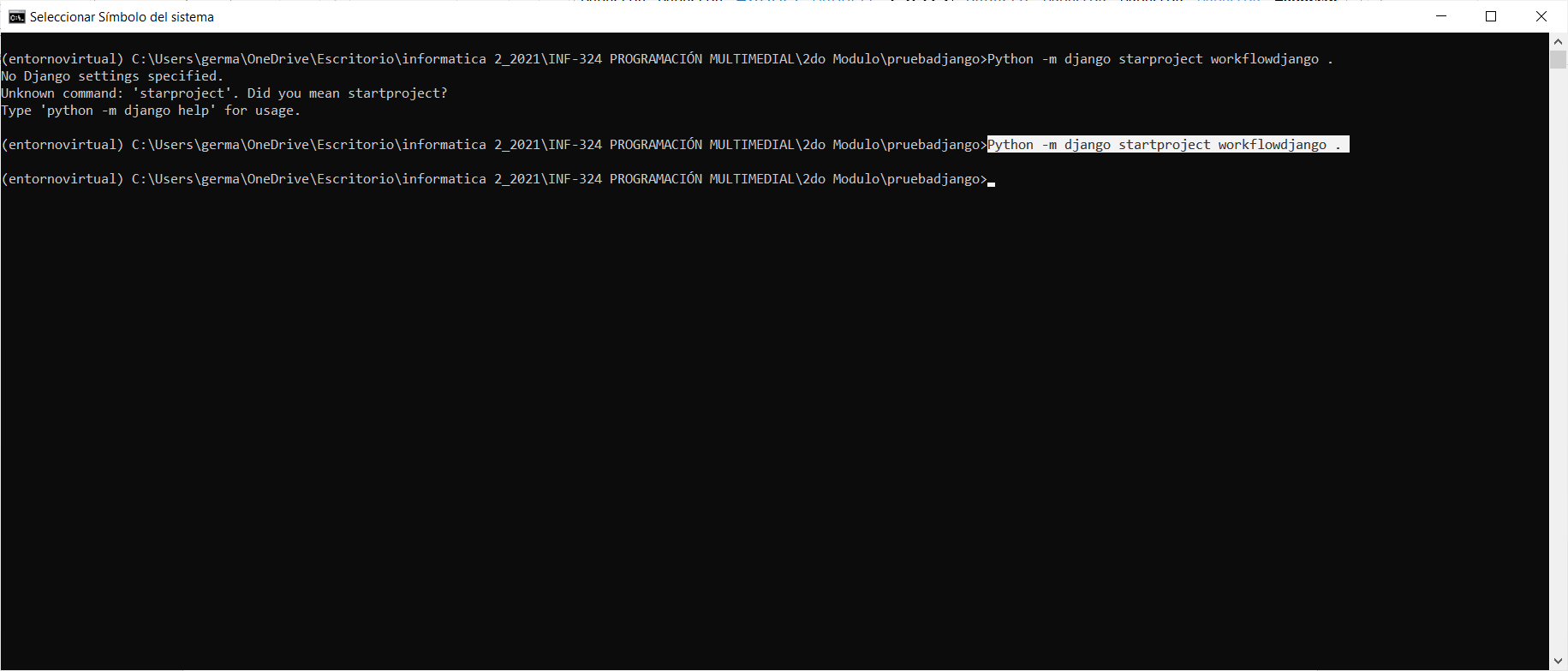
Intento resolver los errores de PATH que aparecen en las advertencias (esto guiándome con el video que encontré)



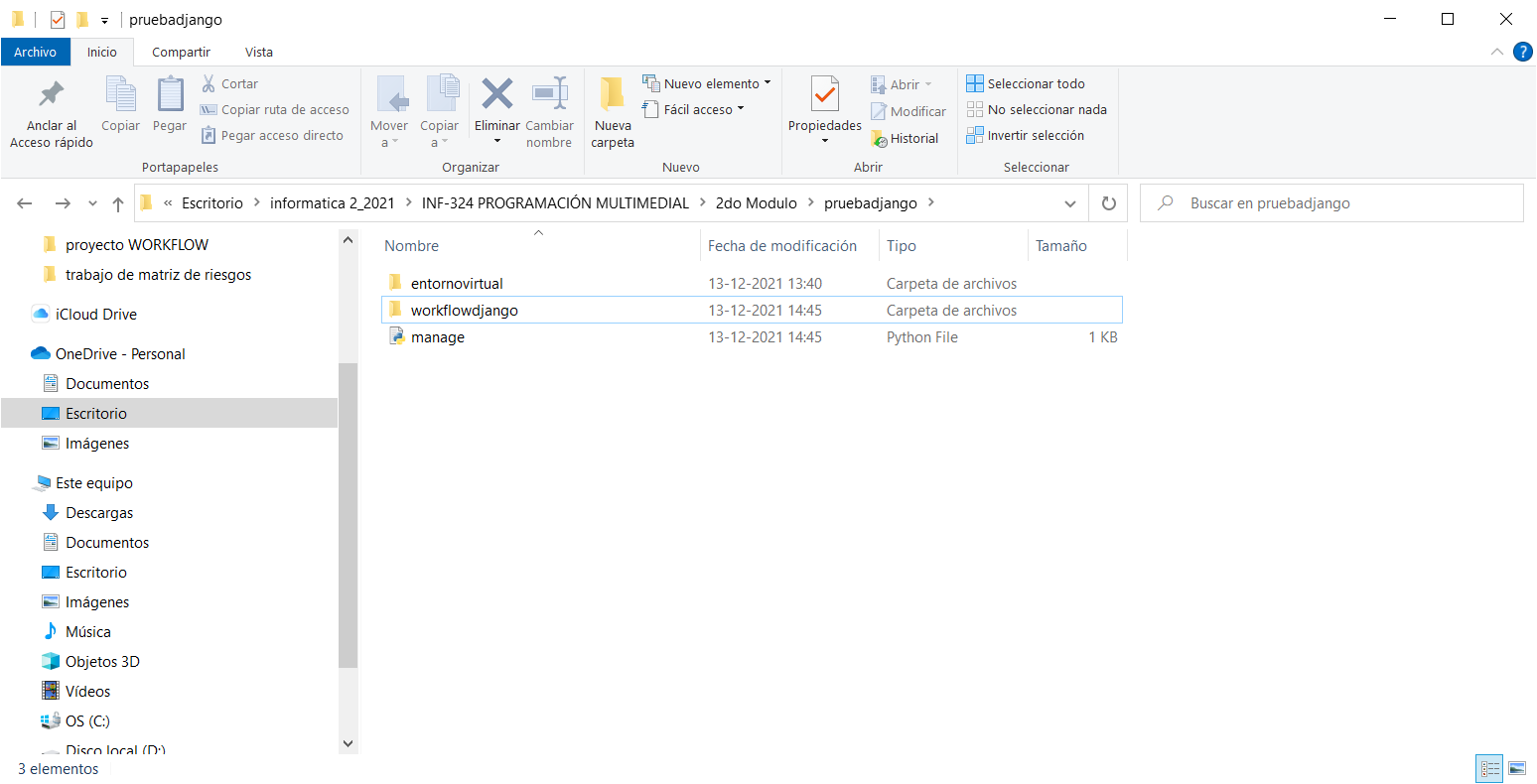
Pero aun nada. No me corre el comando: django-admin

Intentaré seguir trabajando con el otro comando para poder arrancar el proyecto (startproject). Para ello usare el comando:

>**Python –m django startproject nombreproyecto . (el punto hace que se cree el proyecto en la misma carpeta)**



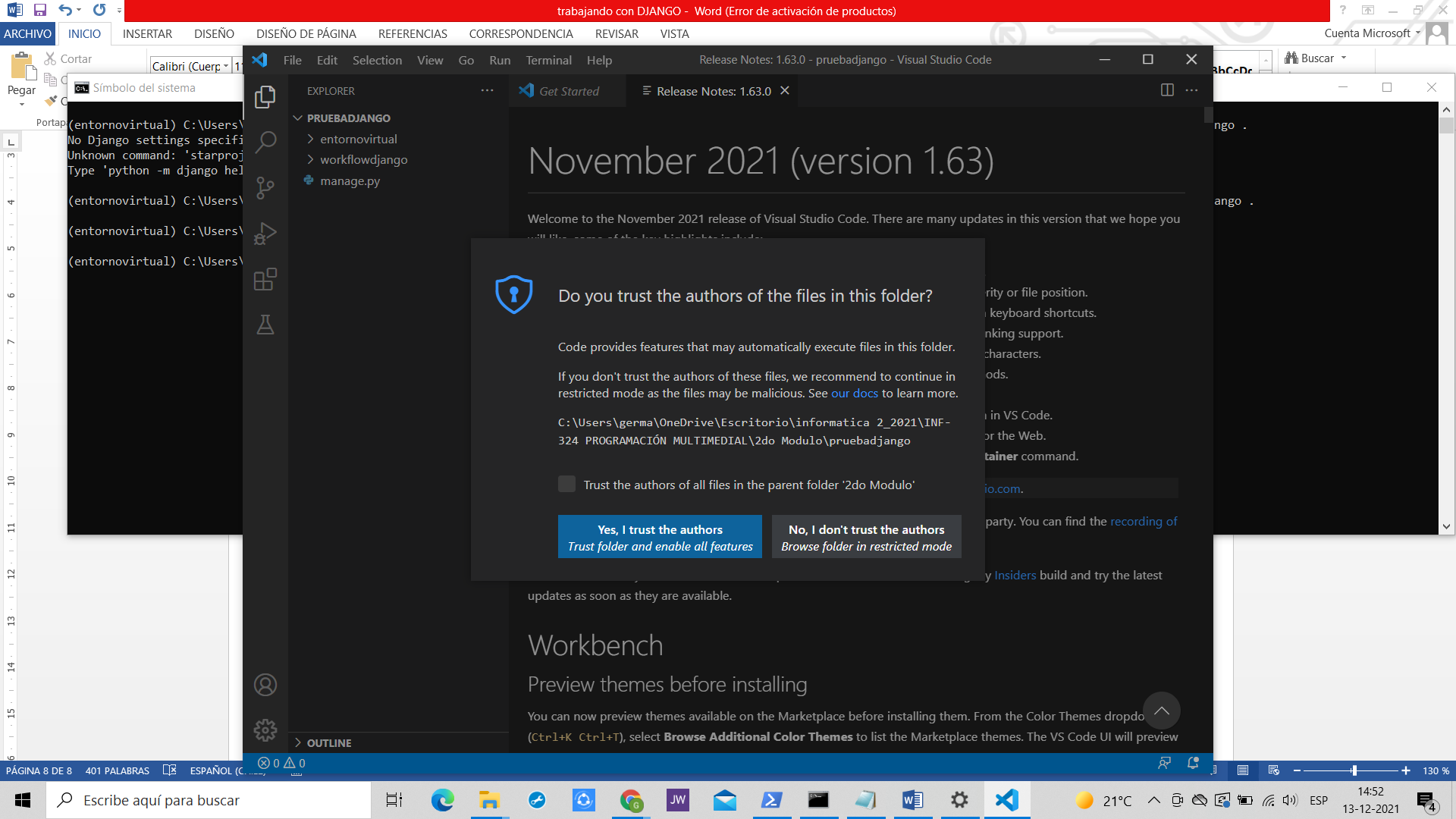
Verificamos… y si, si me creo otra carpeta de nombre “workflowdjango” y además nos aparece el archivo “manage.py”

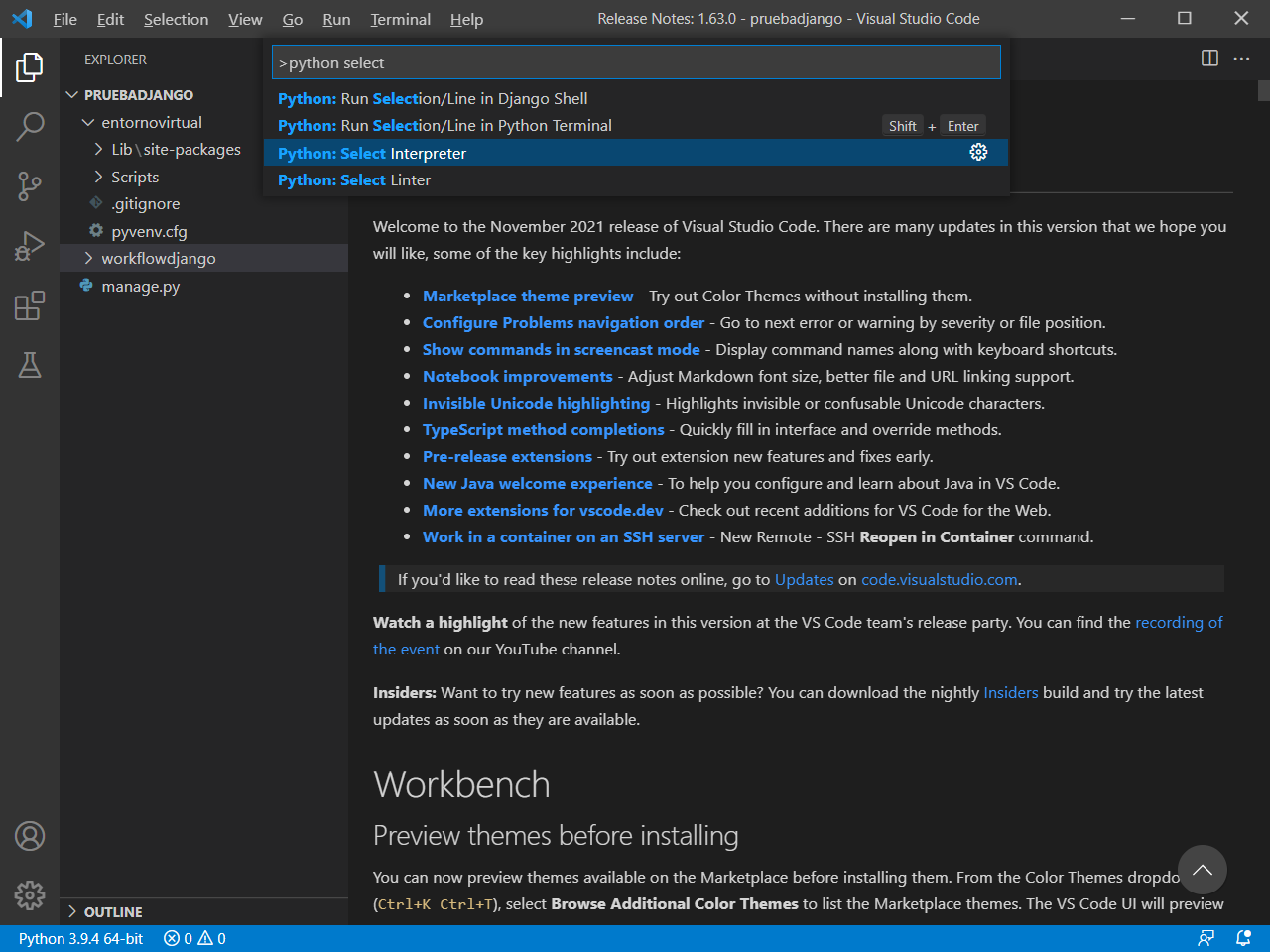


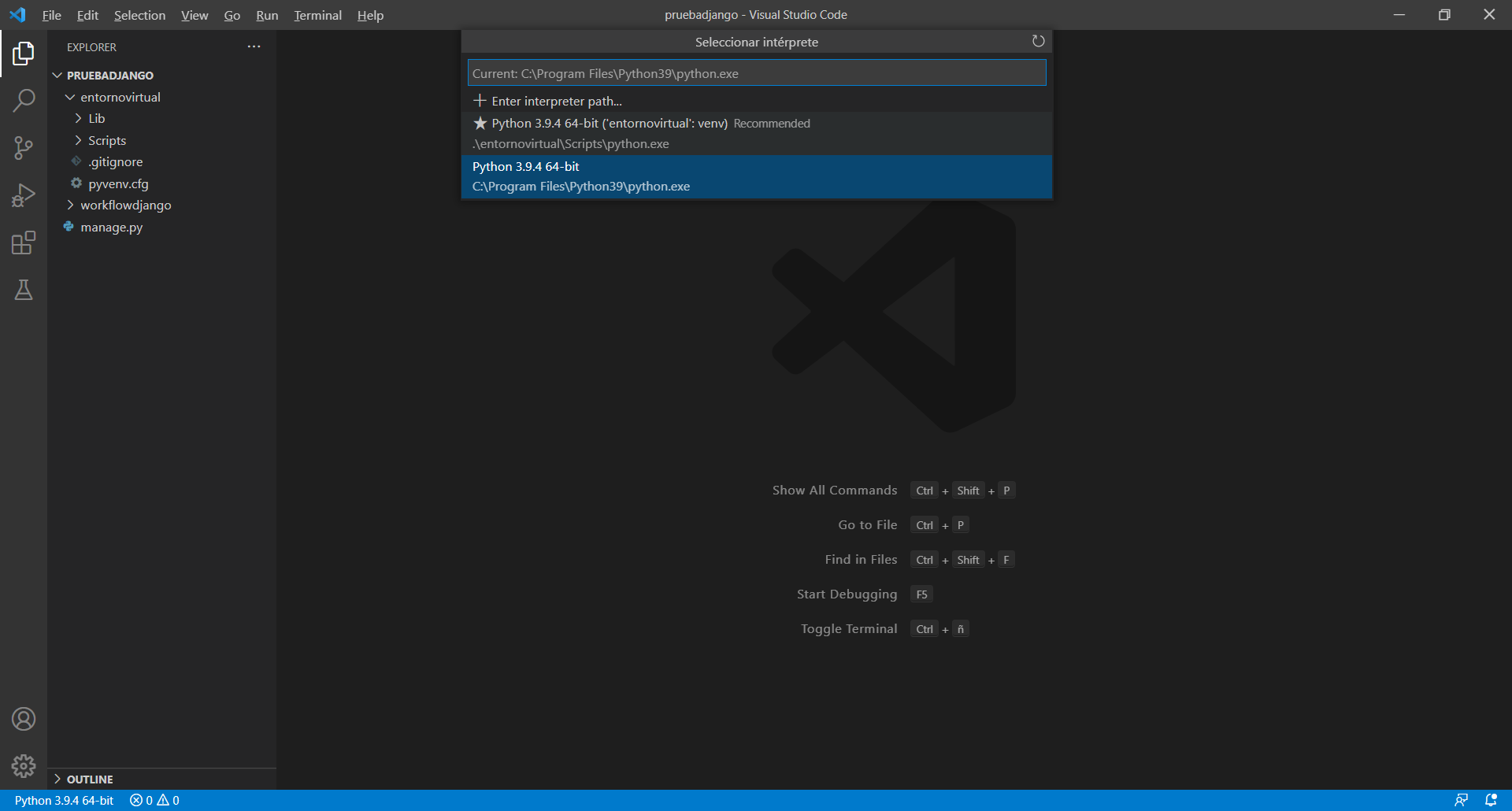
Ahora lo abrimos en visual estudio CODE. Hay 2 maneras: 1) que se arrastre la carpeta a la ventana de Visua estudio y 2) abrirla desde el CMD con el comando: **code .**



Y sí, lo abre. Esto es para que visual studio code se de cuenta de que estamos trabajando en un entorno virtual y no en la instalación por defecto de Python







No puedo cambiar al entorno virtual. No me deja.

Intentaremos seguir con lo que tenemos.

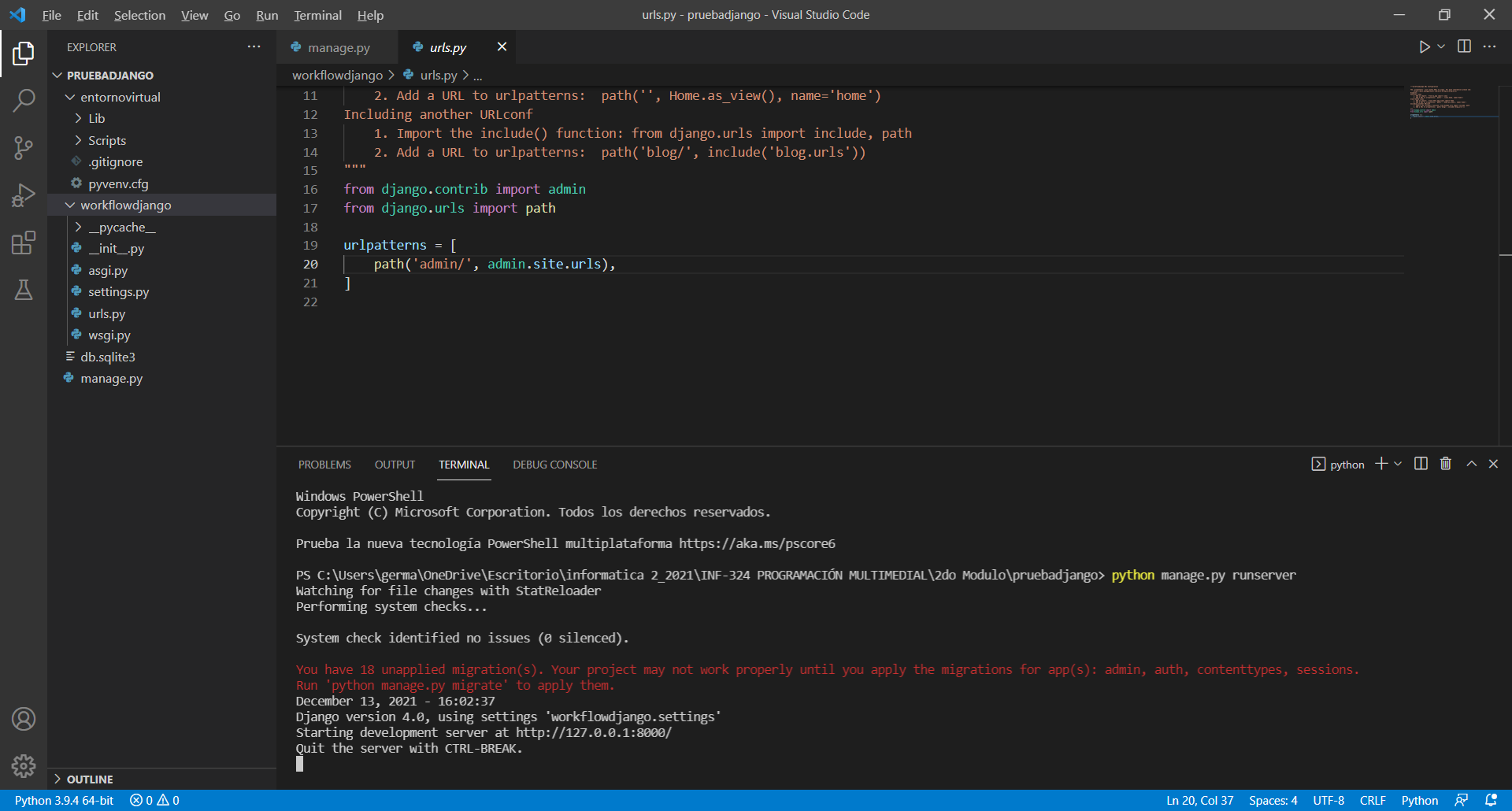
De los archivos que se tienen, estos son los más importantes:

* + manage.py (para ejecutar el proyecto y generar pequeñas porciones de código)
  + settings.py (para configuraciones de todo nuestro proyecto, como la BD por ejemplo)
  + urls.py (permite crear urls para nuestra aplicación)

1. Probar el servidor

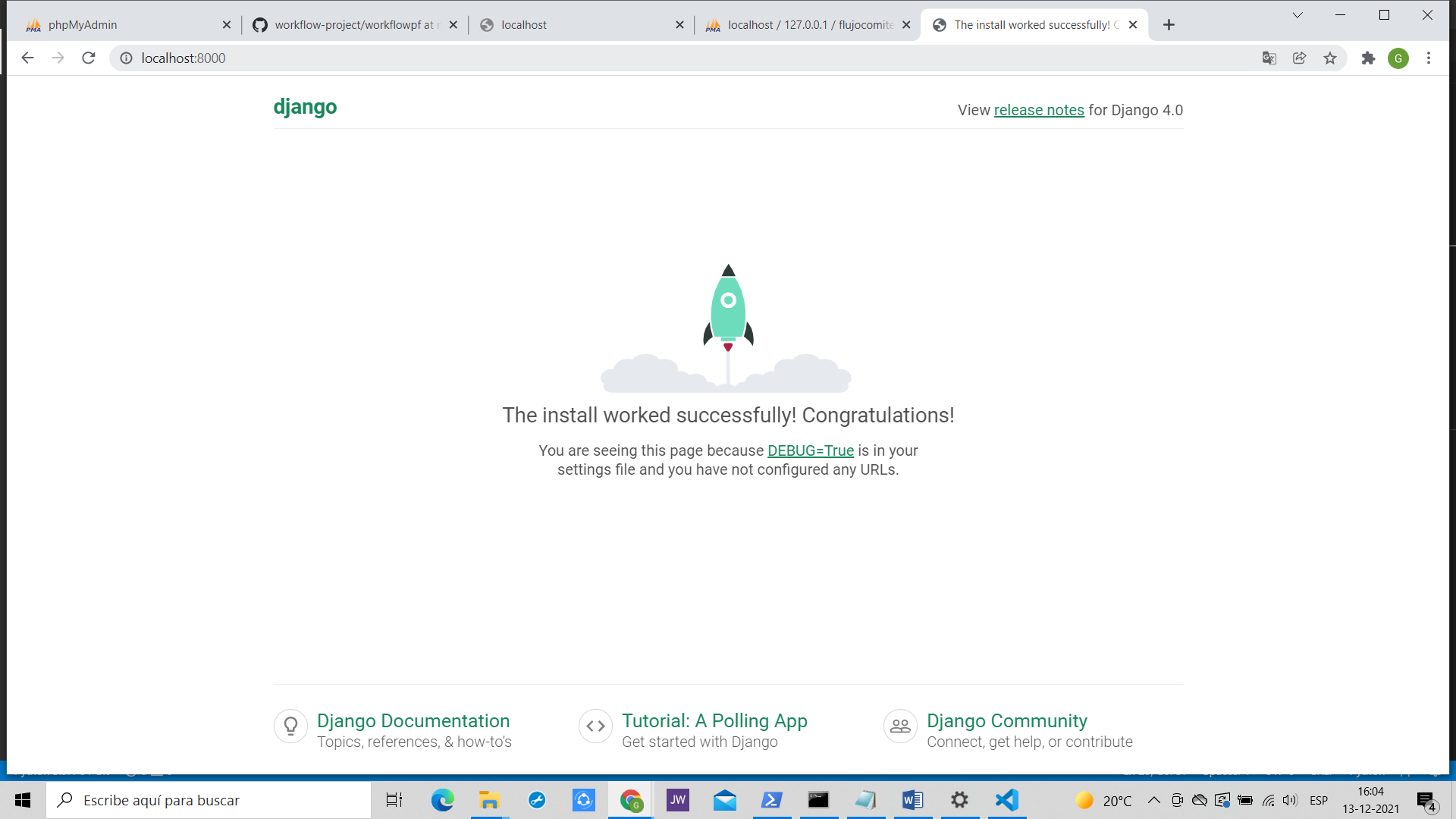
Ahora podemos probar nustro servidor con el comando:

**python manage.py runserver** (el comando “runserver” lo que hace es ejecutar un servidor de este Proyecto)

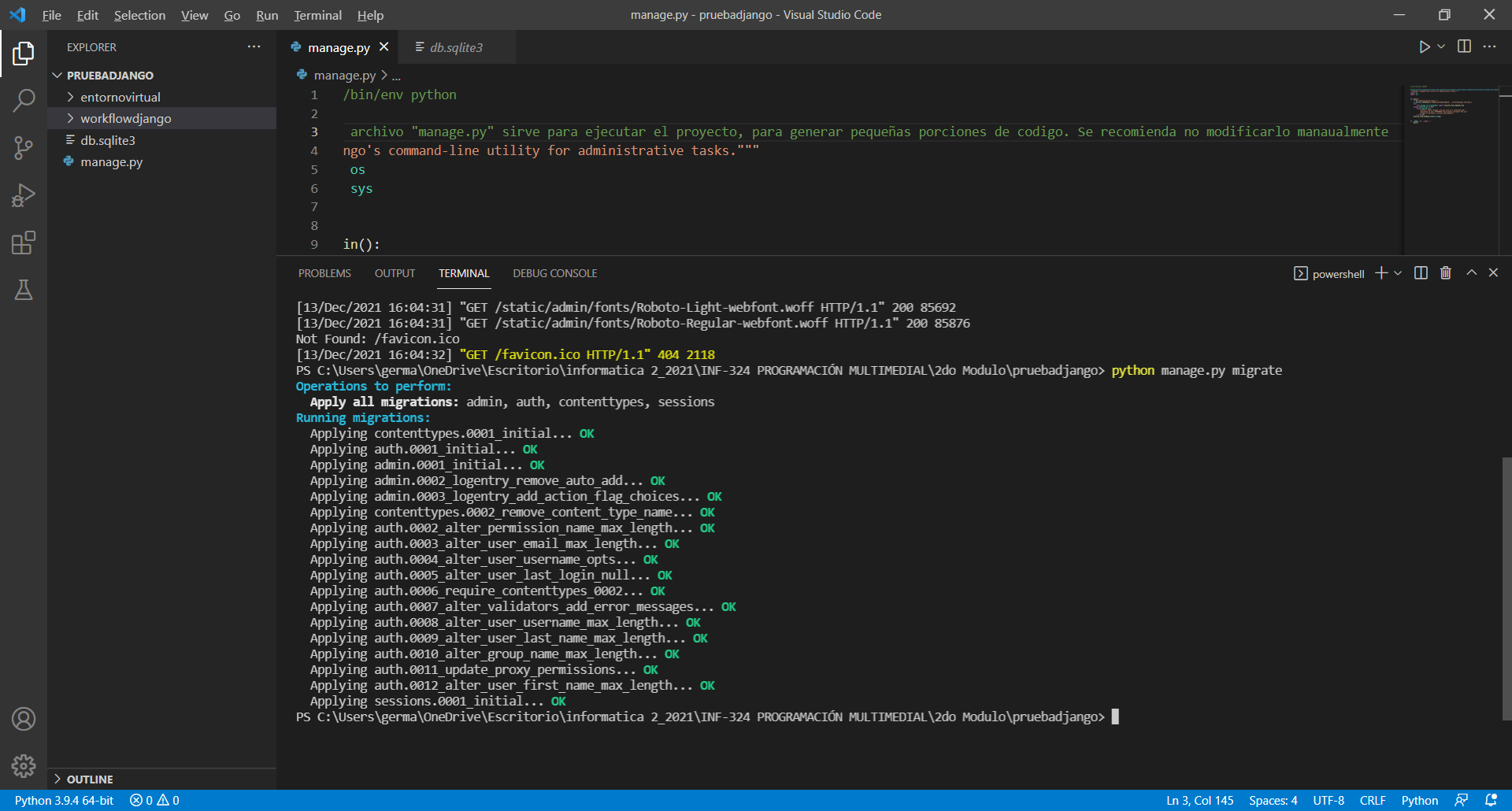


Nos vamos a nuestro localhost:8000

Y está funcionando



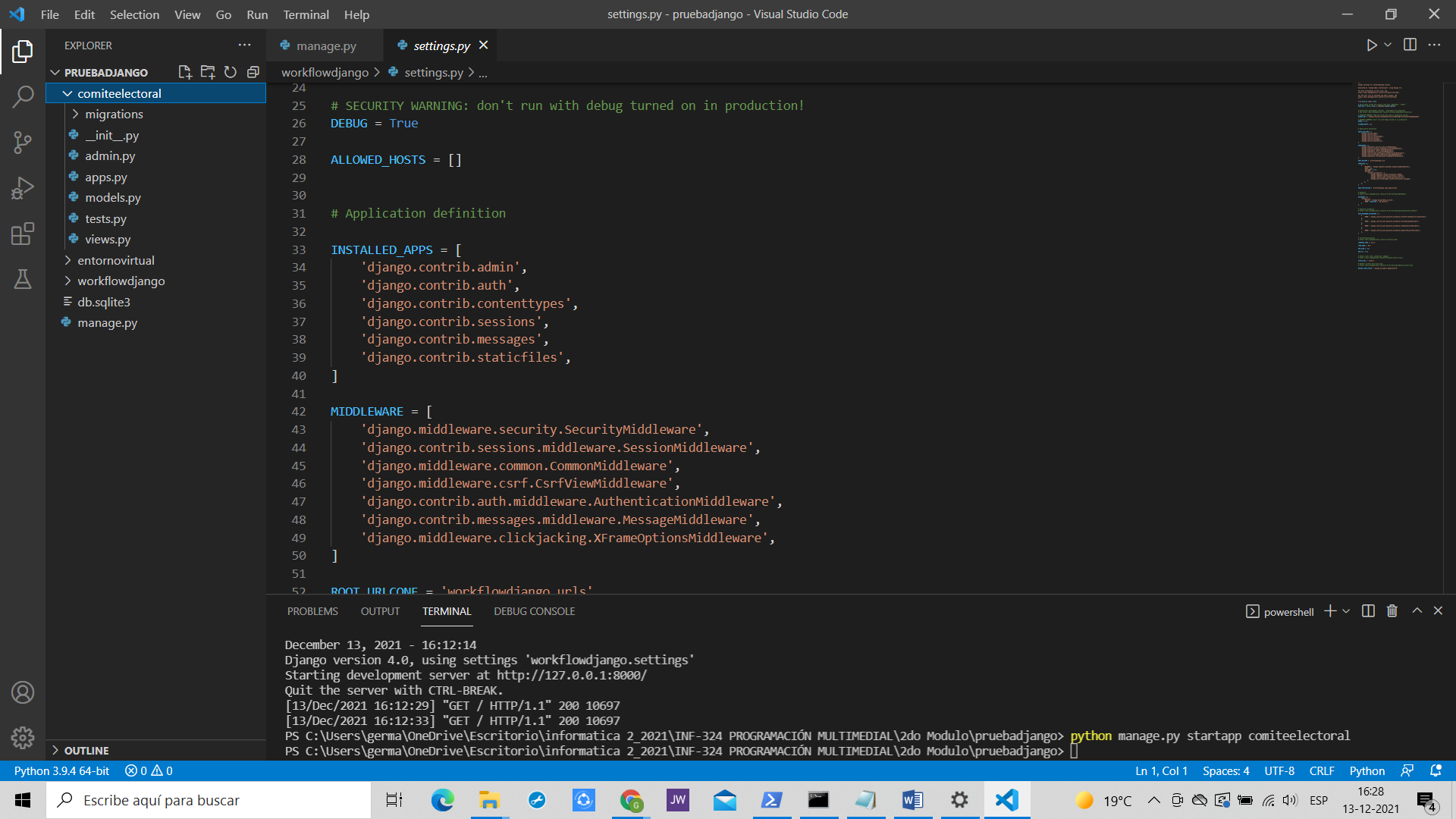
Nos han pedido la migración. Hacemos eso con el comando:> **Python manage.py migrate**



Ahora vamos a **crear una aplicación**. El concepto de aplicación nos ayuda a dividir nuestro proyecto en varias partes. Quizás queremos una aplicación para el administrador, otra aplicación para el usuario, etc.

Para crear una aplicación usamos el comando:

Python manage.py startapp nombrecarpeta (para nuestro caso, el nombre será: comiteelectoral)



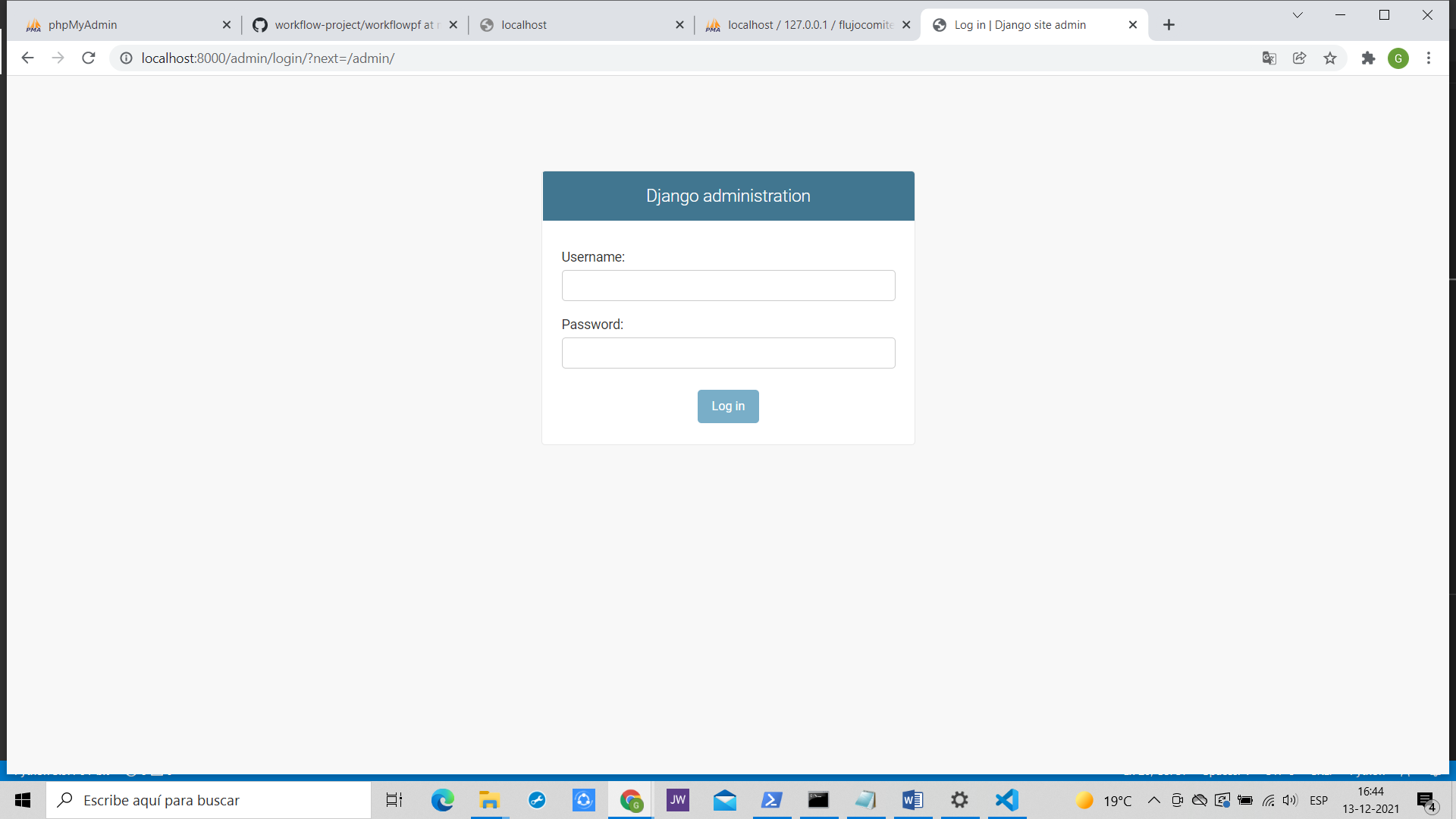
Se creó la carpeta “comiteelectoral” con todos los archivos correspondientes.

De estos archivos, los importantes son:

* + apps.py (aquí está el nombre y la configuración básica de la app)
  + models.py (aquí especificamos que queremos guardar en la base de datos)
  + views.py (para decirle que funciones podemos ejecutar cuando visite nuestra aplicación)

Probamos otra cosa. Hacemos correr el “runserver” y luego nos vamos al localhost y nos vamos a la url “admin”

Localhost:8000/admin



Este login es para ingresar a una parte especial de DJANGO

Nuestro runserver está corriendo.

Ahora pasamos a trabajar en esta nuestra aplicación “comiteelectoral”

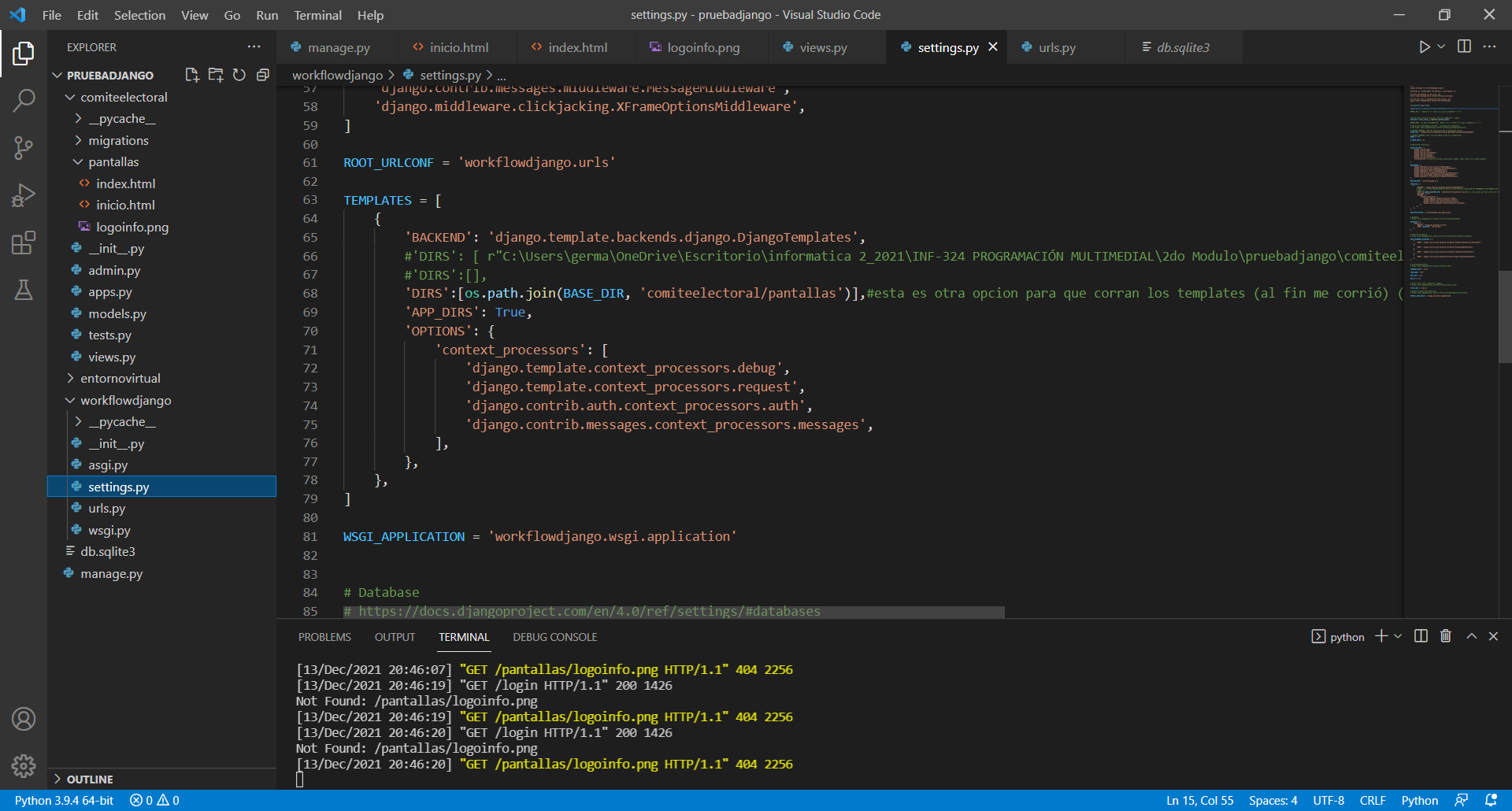
(Antes que nada debemos modificar ‘settings,py’ añadiendo nuestra carpeta de trabajo: comiteelectoral)

1. Creamos una subcarpeta para manejar los archivos html: “pantallas” (que serán nuestros templates). Para que django reconozca estos templates, tenemos que escribir, en settings.py, la ruta de nuestra carpeta que contendrá los templates en la parte que dice TEMPLATES ….. ‘DIRS’ :[…….],
2. En **views.py** creamos una función que llame a nuestro archivo html
3. En **urls.py** creamos un path para nuestra función que está en views.py (en el primer campo está el nombre que irá en la url (este nombre es indistinto, es lo que aparecerá en el url) , en el segundo campo llamamos la función) y antes de eso importamos nuestro archivo ‘views’ desde ‘comiteelectoral’.

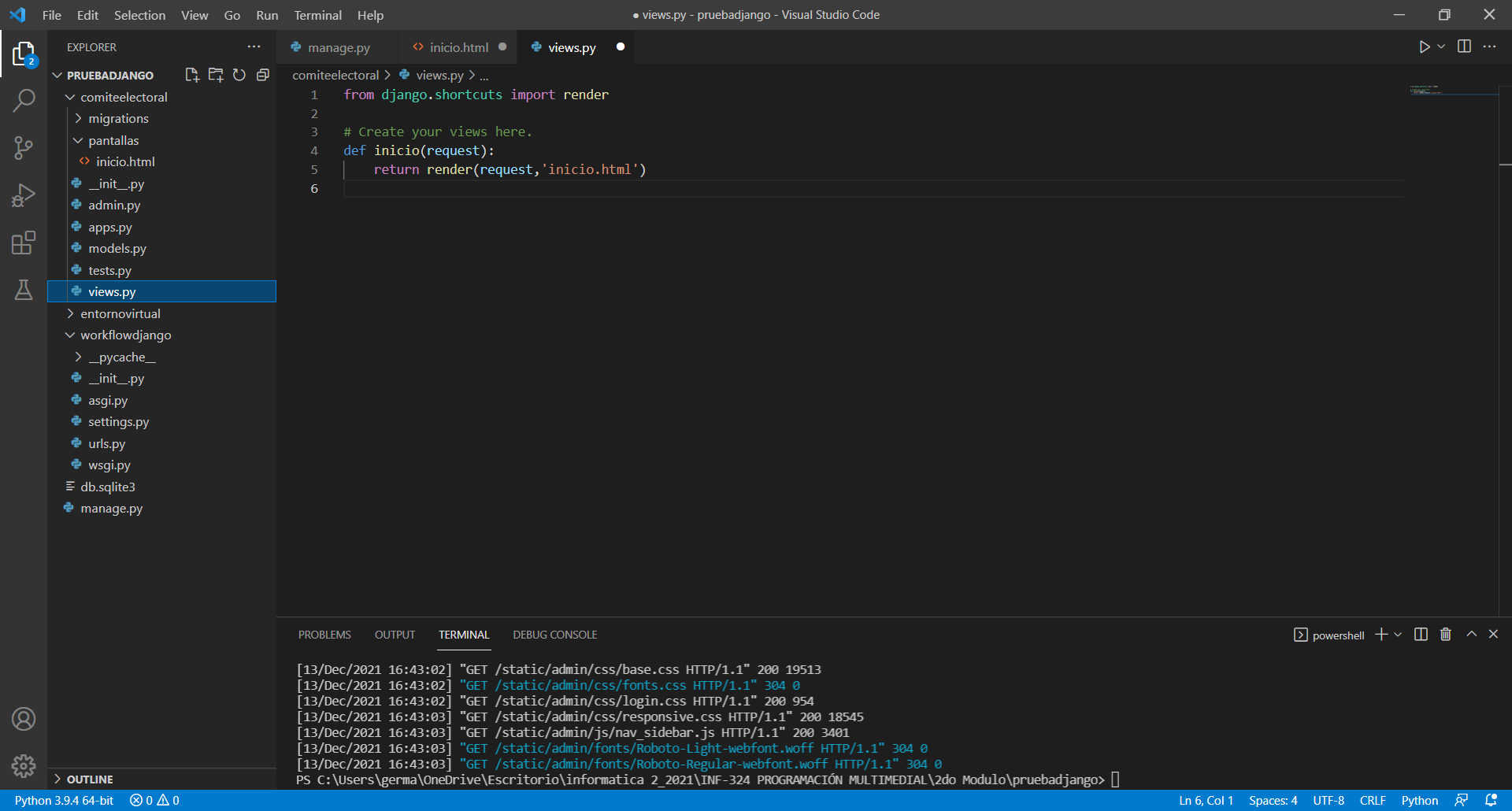
(antes de todo, en settings.py tenemos que adicionar la carpeta con la que estamos trabajando. En nuestro caso: ‘comiteelectoral’ )



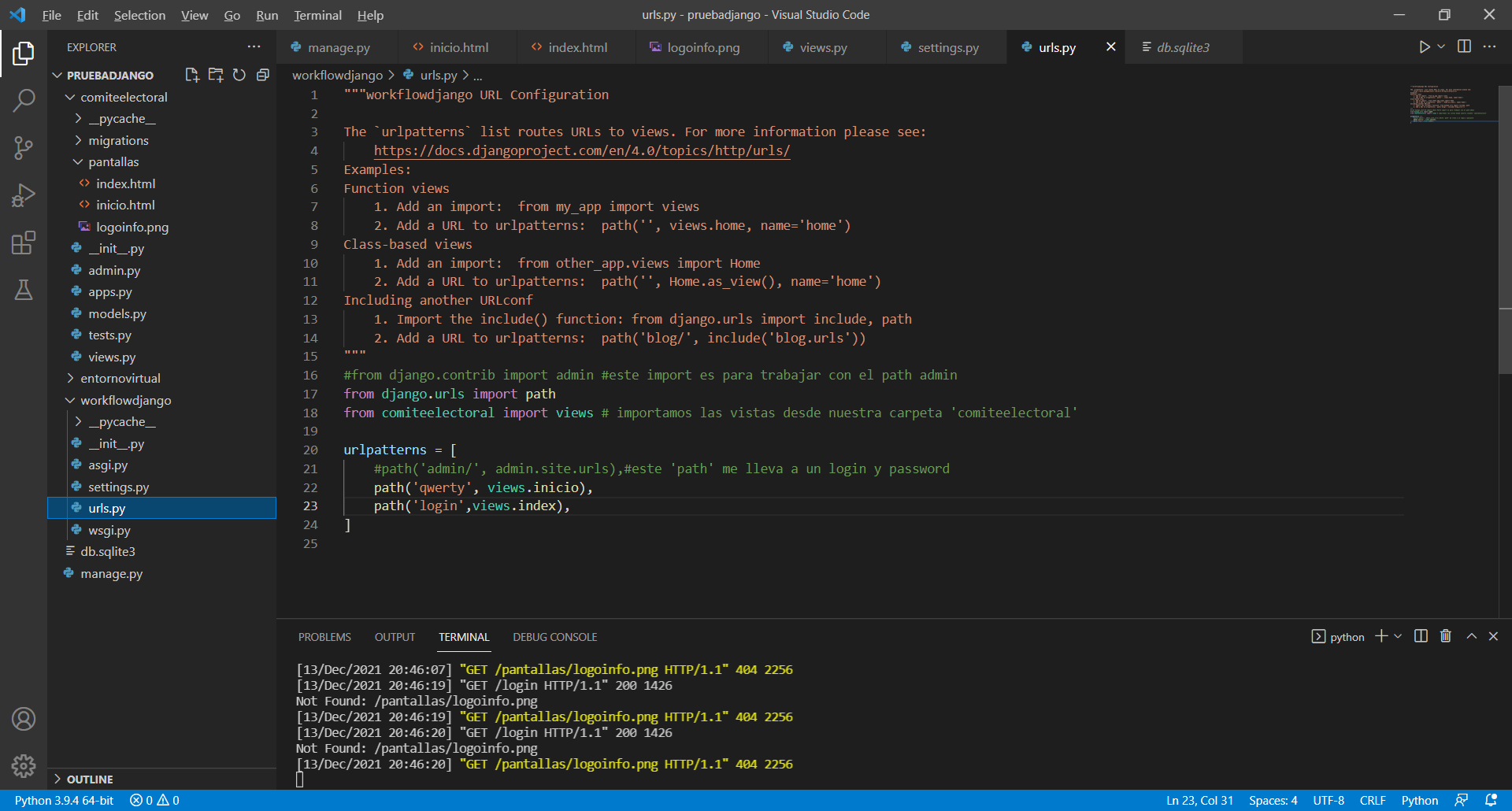
1. Creamos una subcarpeta para manejar los archivos html: “pantallas” (que serán nuestros templates). Para que django reconozca estos templates, tenemos que escribir, en settings.py, la ruta de nuestra carpeta que contendrá los templates en la parte que dice TEMPLATES ….. ‘DIRS’ :[…….],



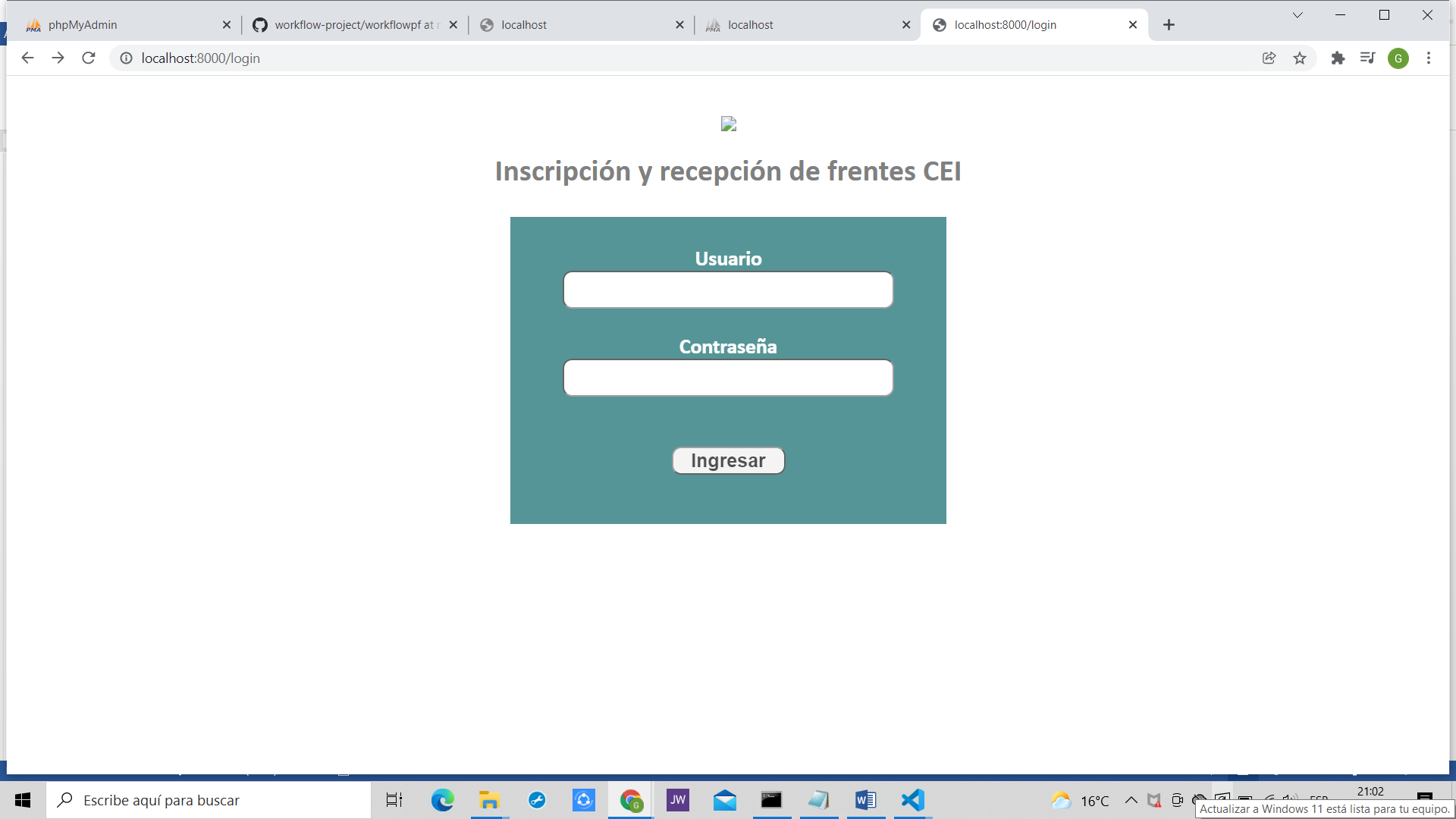
1. En views.py creamos una función que llame a nuestro archivo html



1. En urls.py creamos un path para nuestra función que está en views.py (en el primer campo está el nombre que irá en la url, en el segundo campo llamamos la función) y antes de eso importamos nuestro archivo ‘views’ desde ‘comiteelectoral’.



Finalmente lo conseguimos. (Después de mucho intentar)



Sgsgsgsgsggs

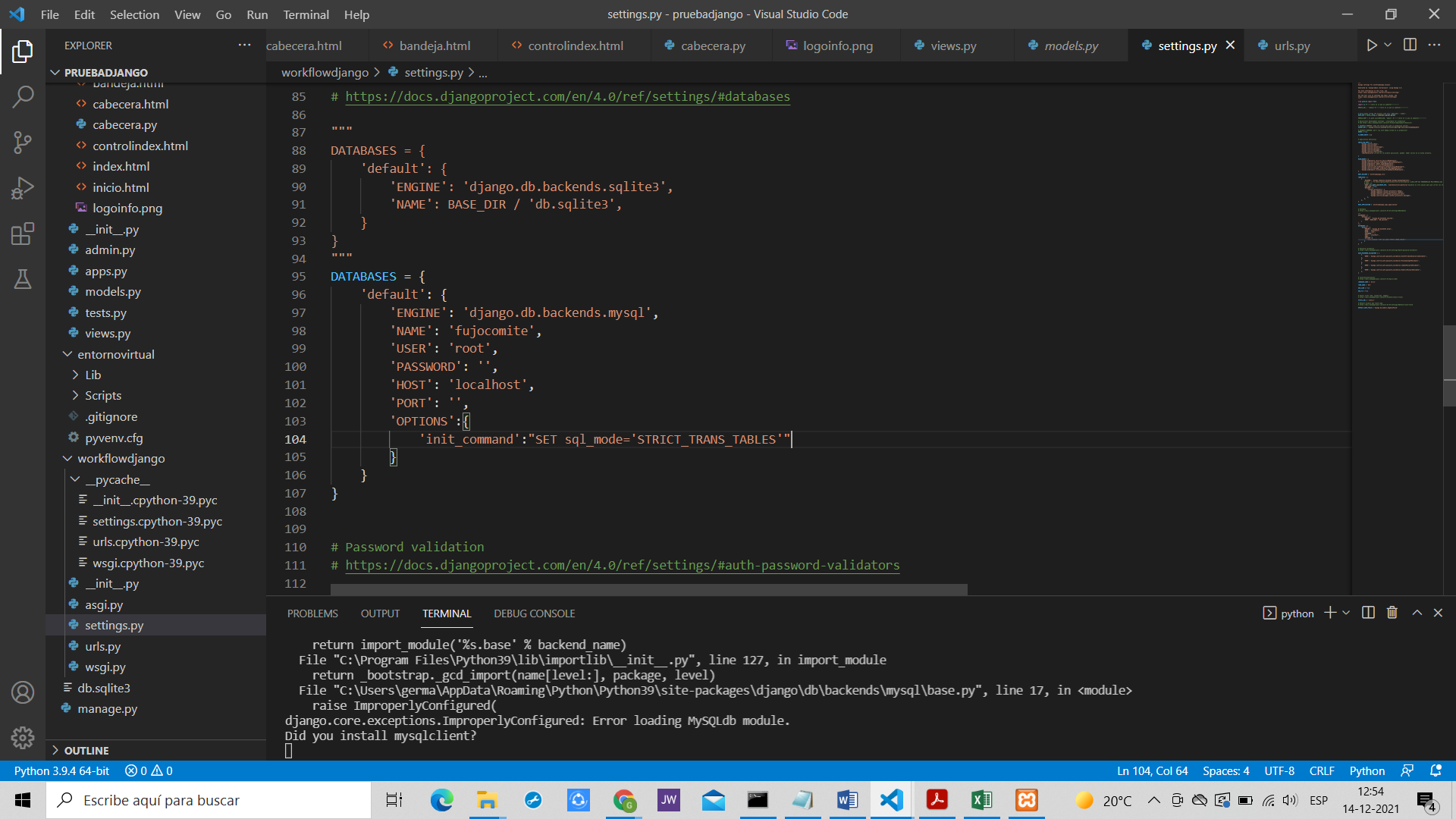
1. Actualizar el archivo settings.py

Ya hicimos las actualizaciones en los pasos anteriores

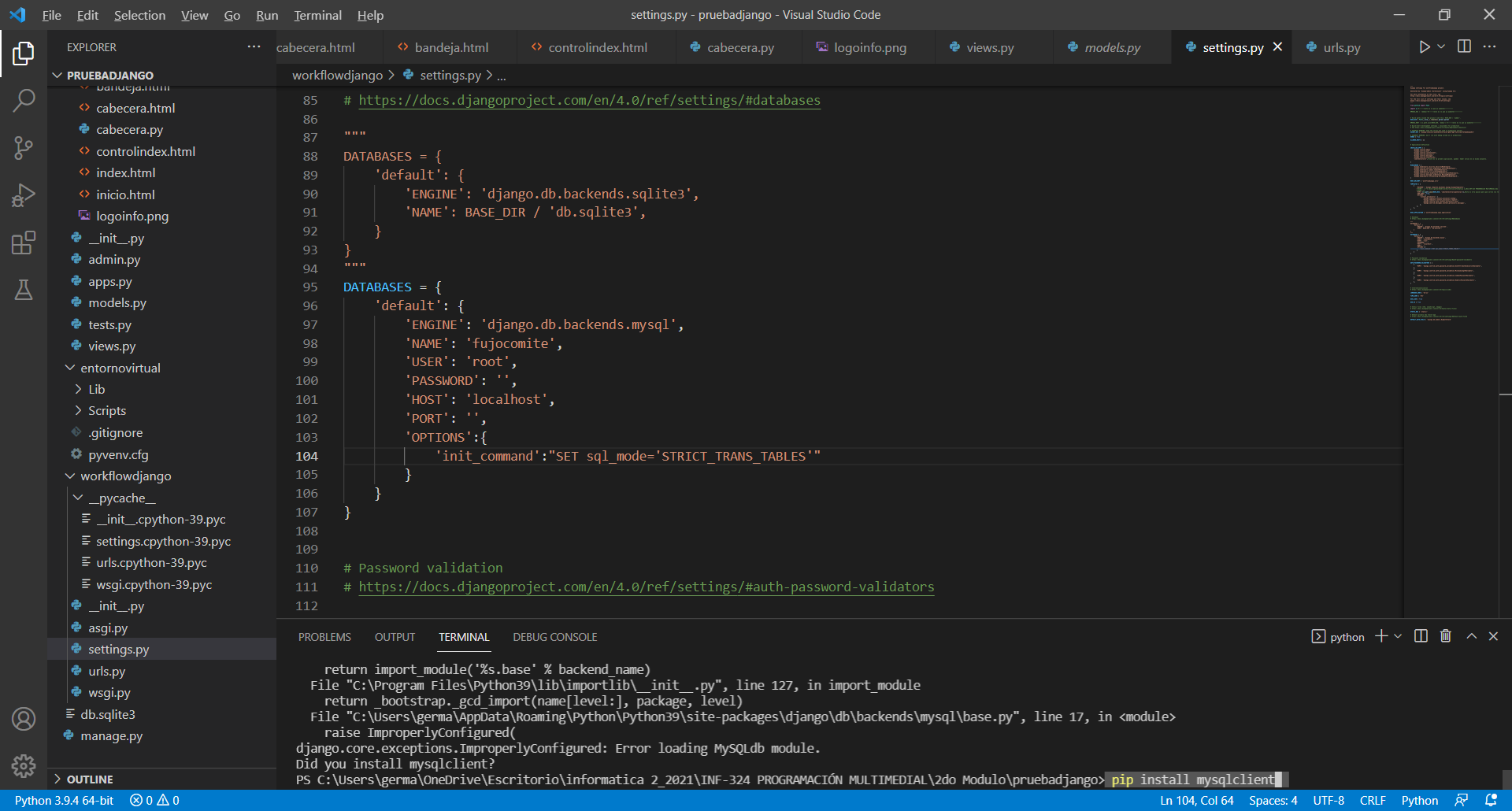
1. Crear un modelo
2. Crear un formulario para cargar archivos
3. Agregar vistas
4. Configurar el archivo urls.py
5. Pruebas

Haremos la conexión a la base de datos, para poder realizar el inicio de sesión.

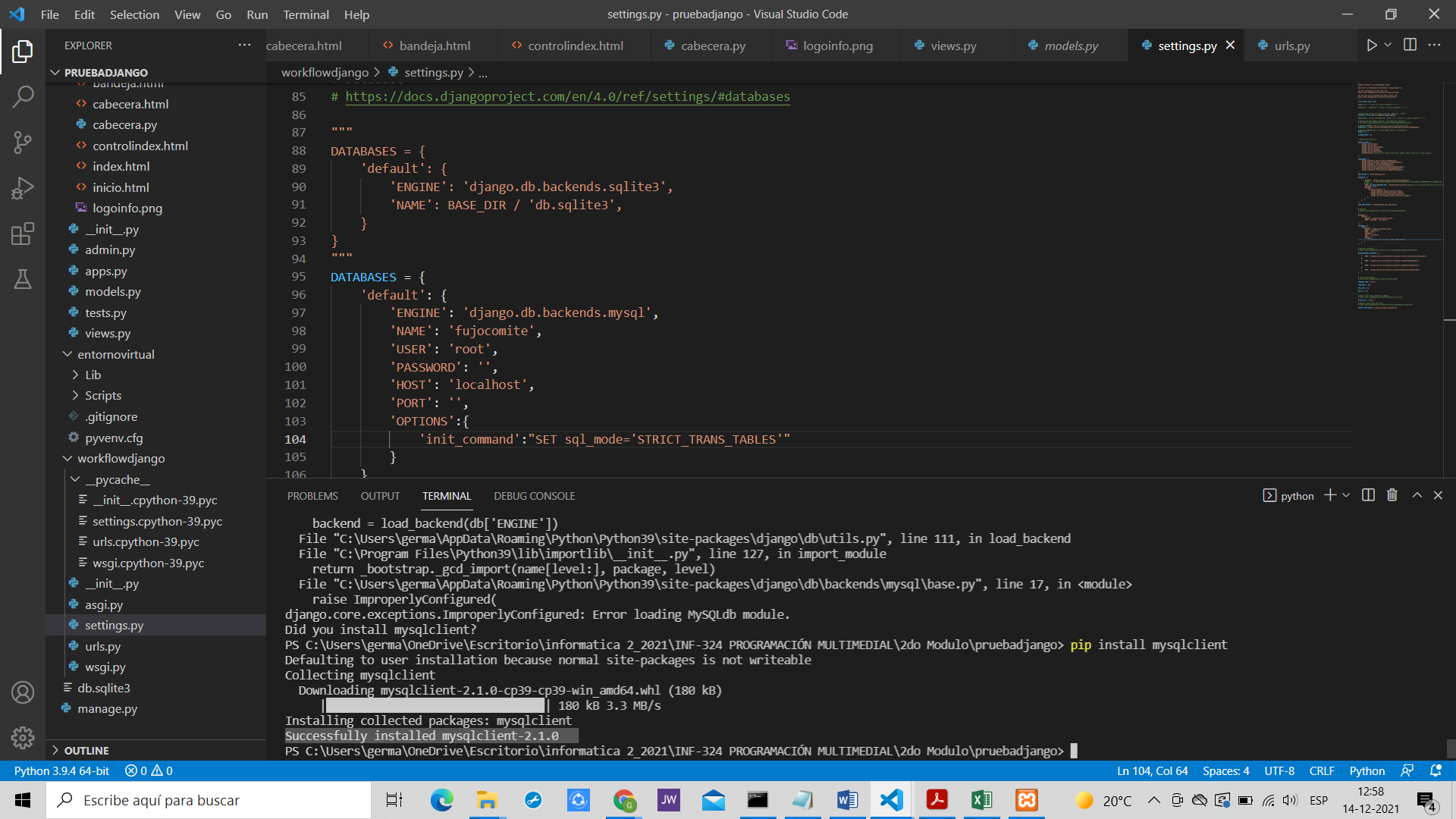
Trabajaremos con mysql php my admin



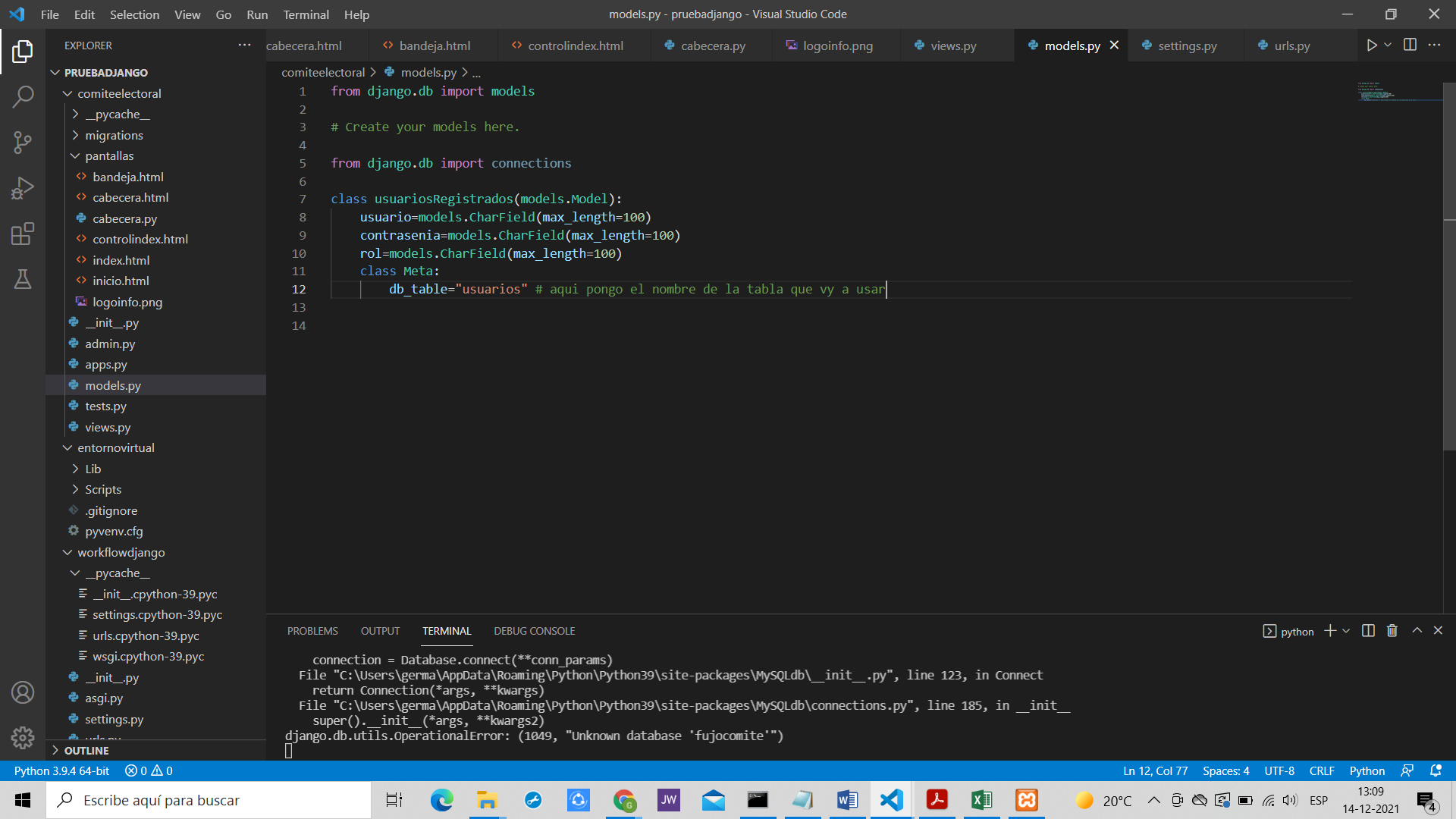
Nos regunta si istalamos mysqlclient. Lo hacemos con el comando: >pip install mysqlclient



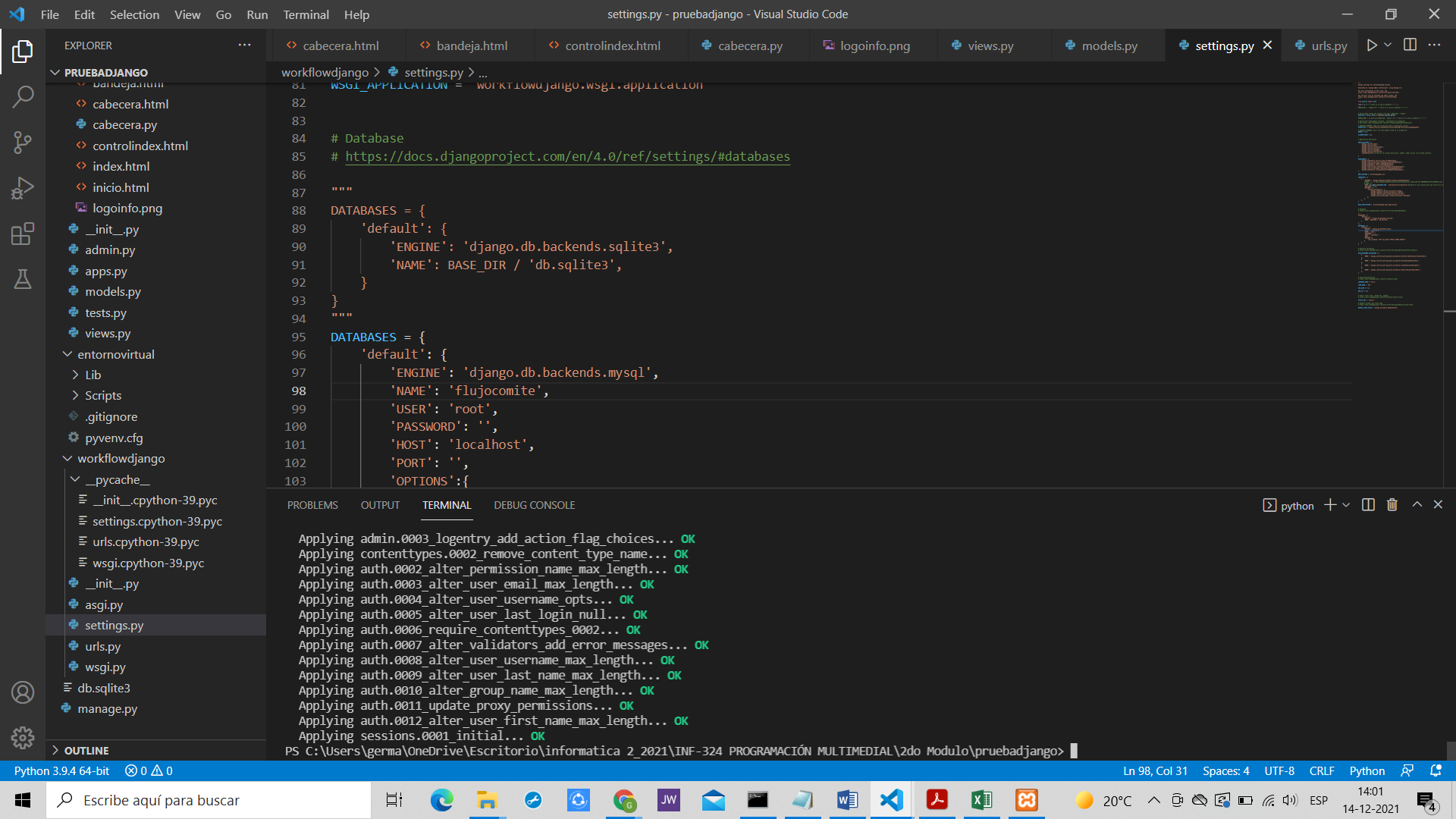
Se instaló correctamente



Ahora modificamos el archivo models.py (ponemos una clase que pueda sacar datos de la BD)

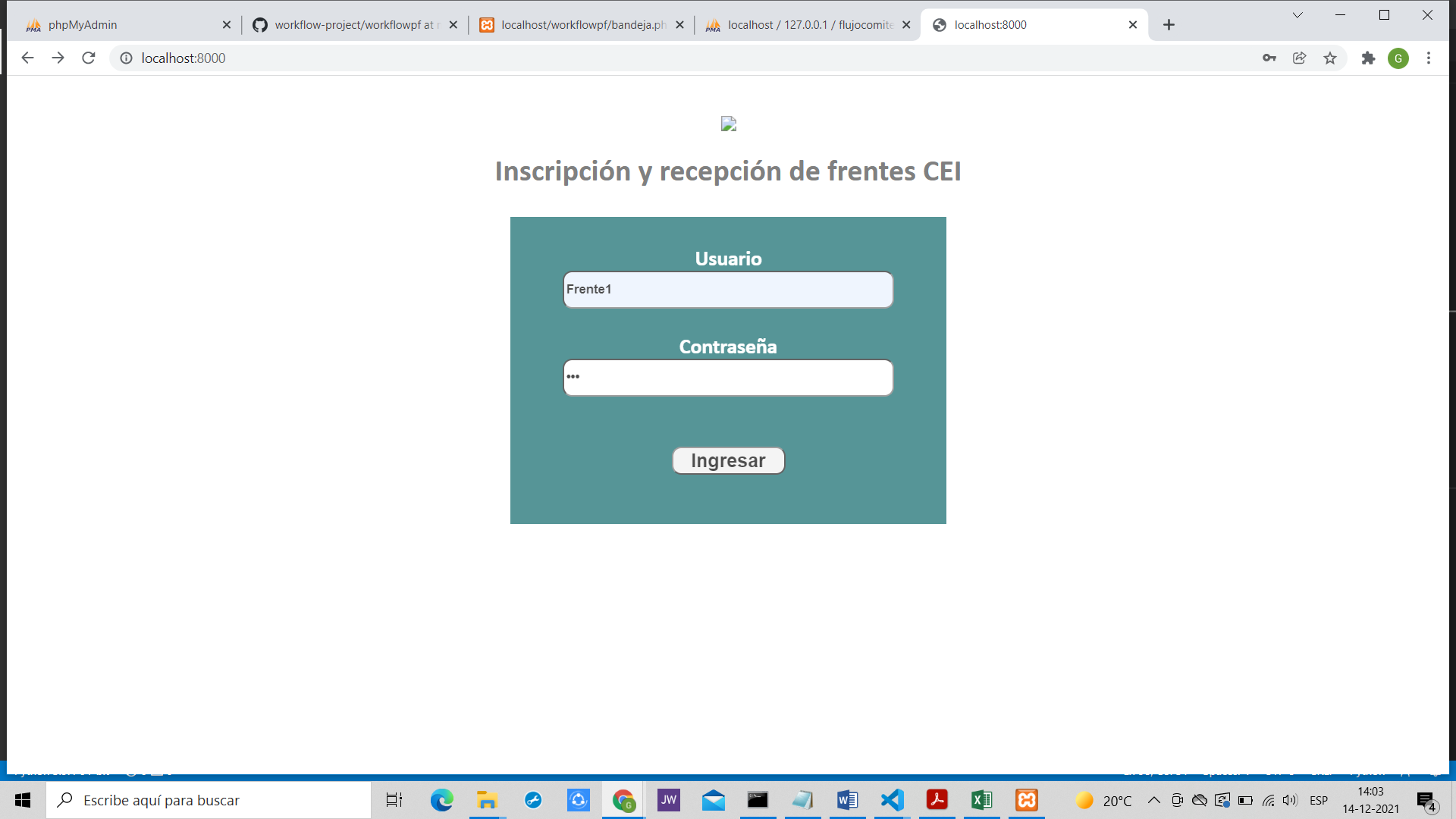


Hice todo el proceso de obtener los datos del formulario, ingresar a la base de datos, sacar datos, comparar, hacer las redirecciones correspondientes y todo eso. Una vez hecho todo eso hay que migrar con el comando: > python manage.py migrate



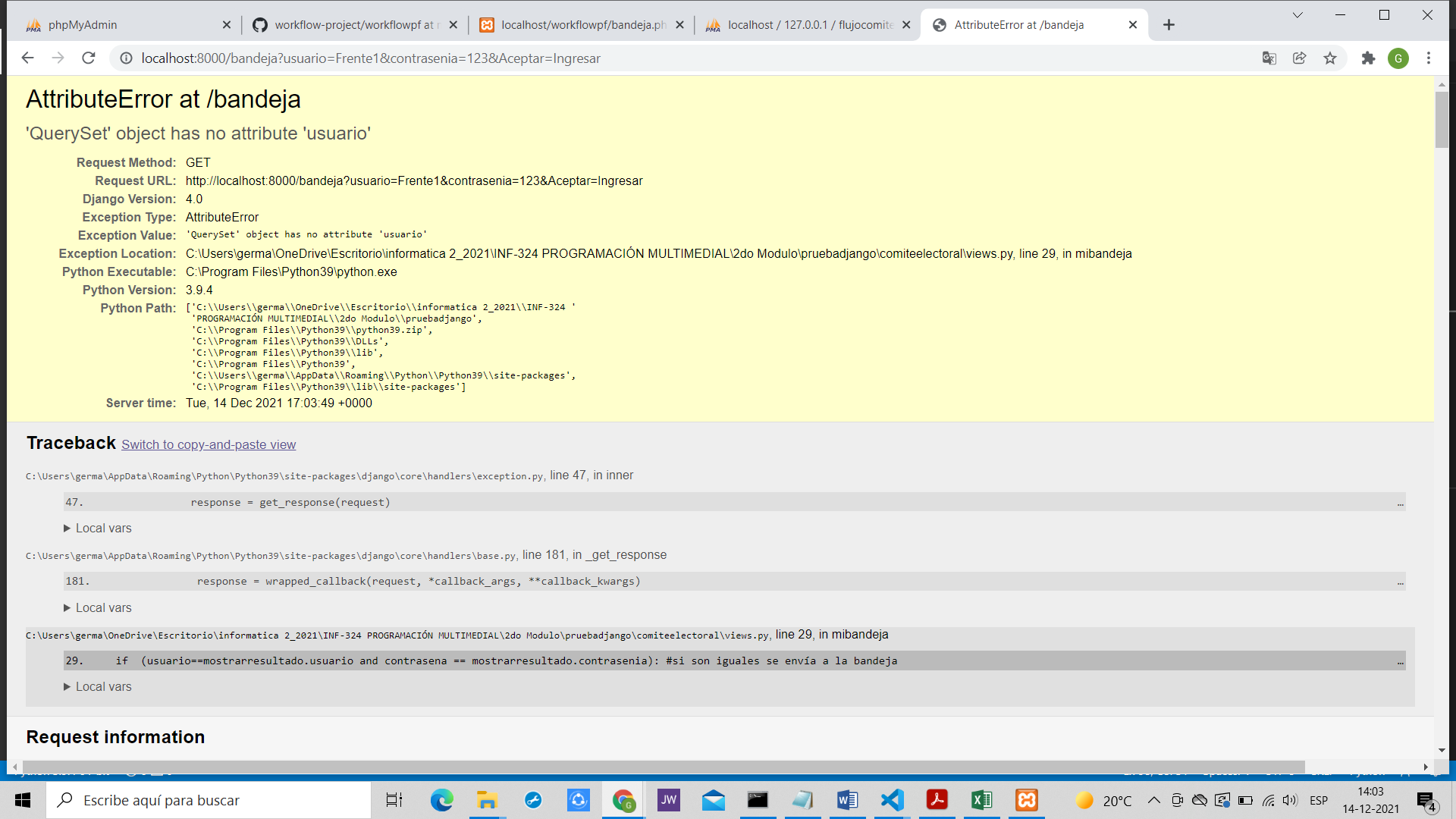
Volvemos a ejecutar el comando: >Python manage.py runserver

Y vemos que tal



Tenemos al menos la página de inicio.

Pero al momento de ingresar nos muestra este error



Tenemos un error que del que no encuentro solución

Tenemos esto que va corriendo



