

***Carrera: Desarrollo de Software***

***Estudiante: Michelle Juliet Carreño Morales***

******

***Paralelo: 3 <H>***

***Profesor: CARLOS LUIS PAZMIÑO***

***TEMA: GIT-GITHUB***

***TALLER # 1***

Los temas a desarrollar son los siguientes:

1. ¿Qué es Django?

Django es un flamework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables.

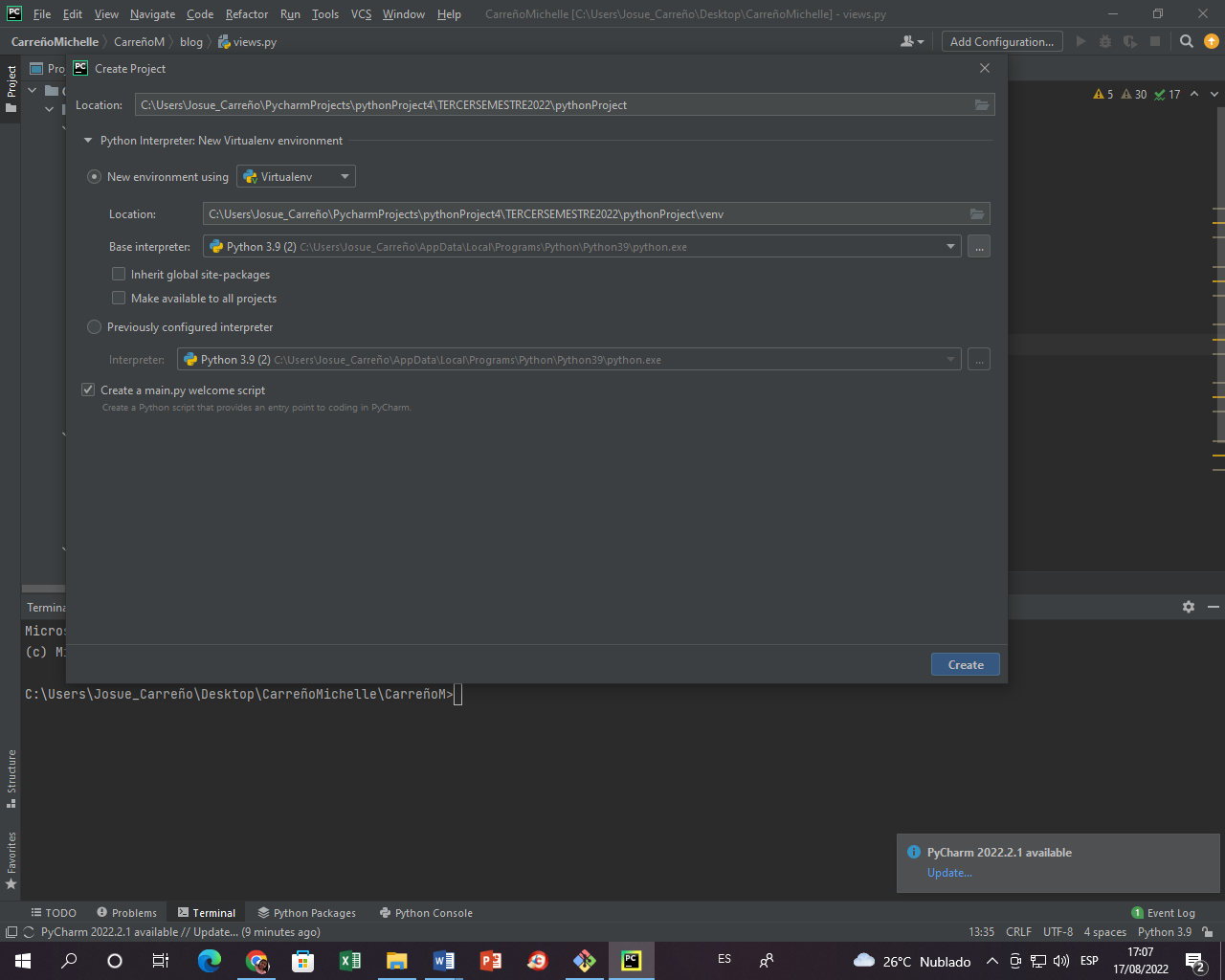
1. ¿Qué es la máquina virtual en Django

Es la que permite trabajar en el proyecto de Django

1. ¿Qué es MVT en Django. ?

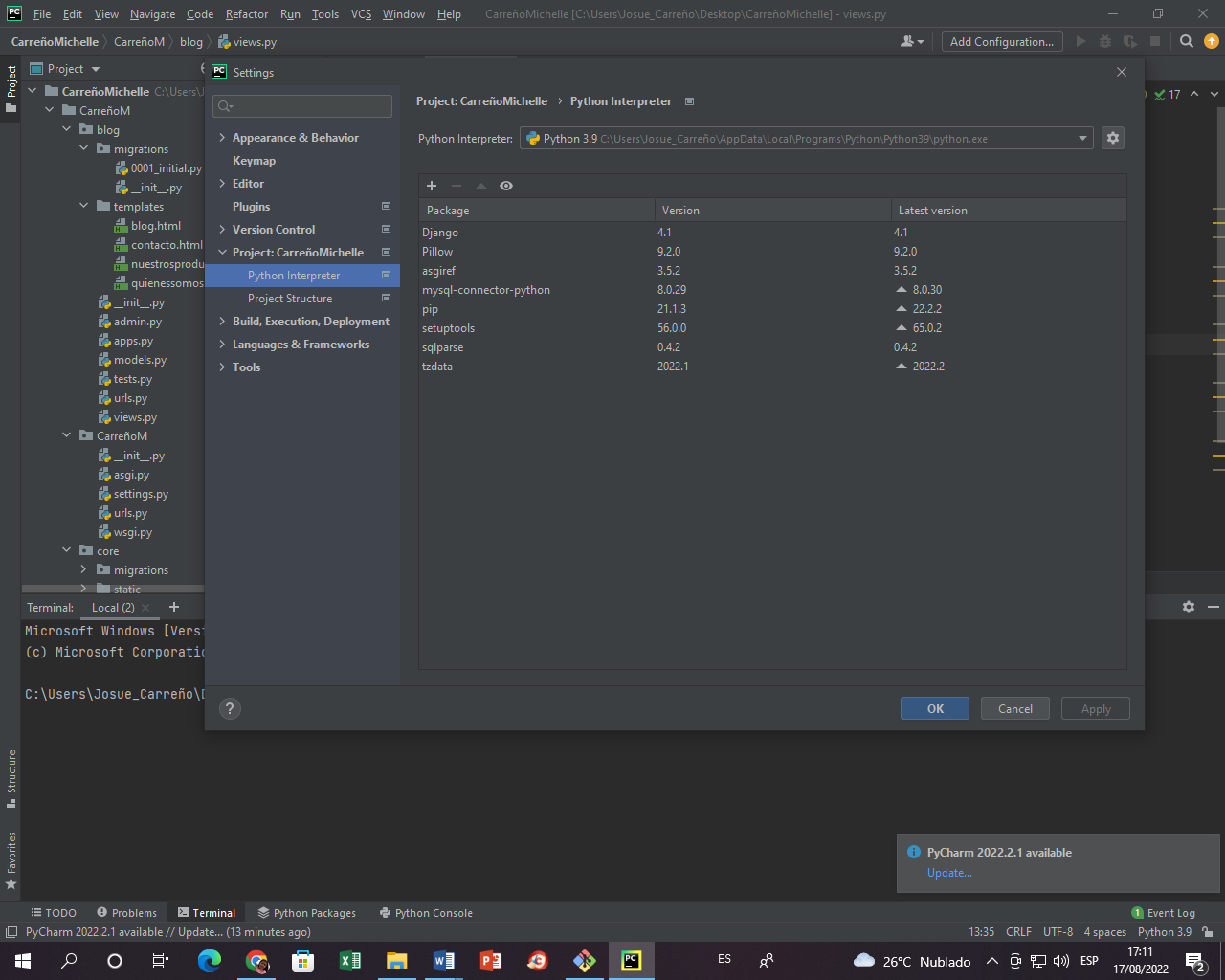
Es una forma de que esté conectado el directorio templates al views y al url del core

1. Crear un proyecto con la máquina virtual.



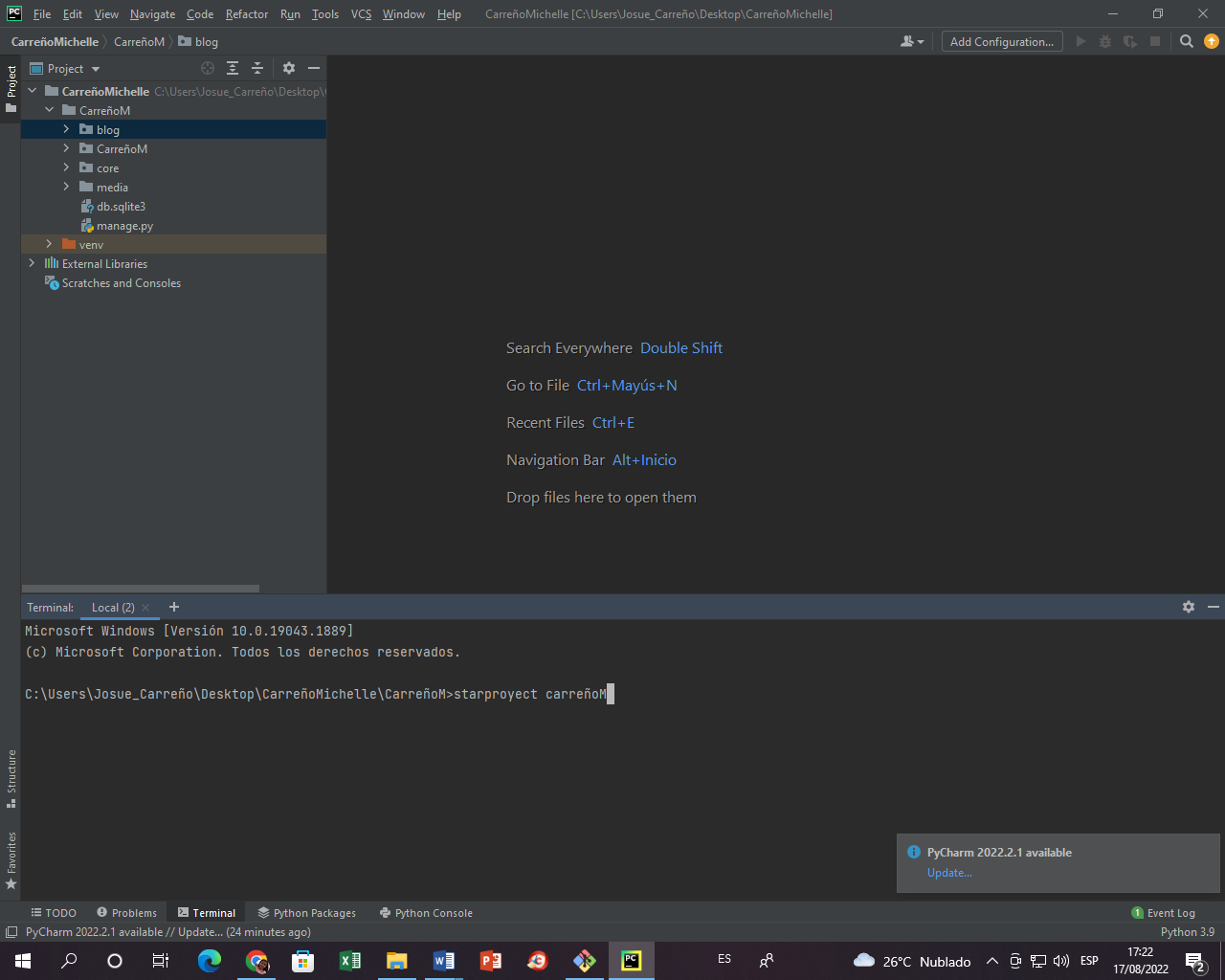
En esta pantalla se ve como creamos un nuevo proyecto con la máquina virtual

1. Descargar los instaladores de Django al proyecto



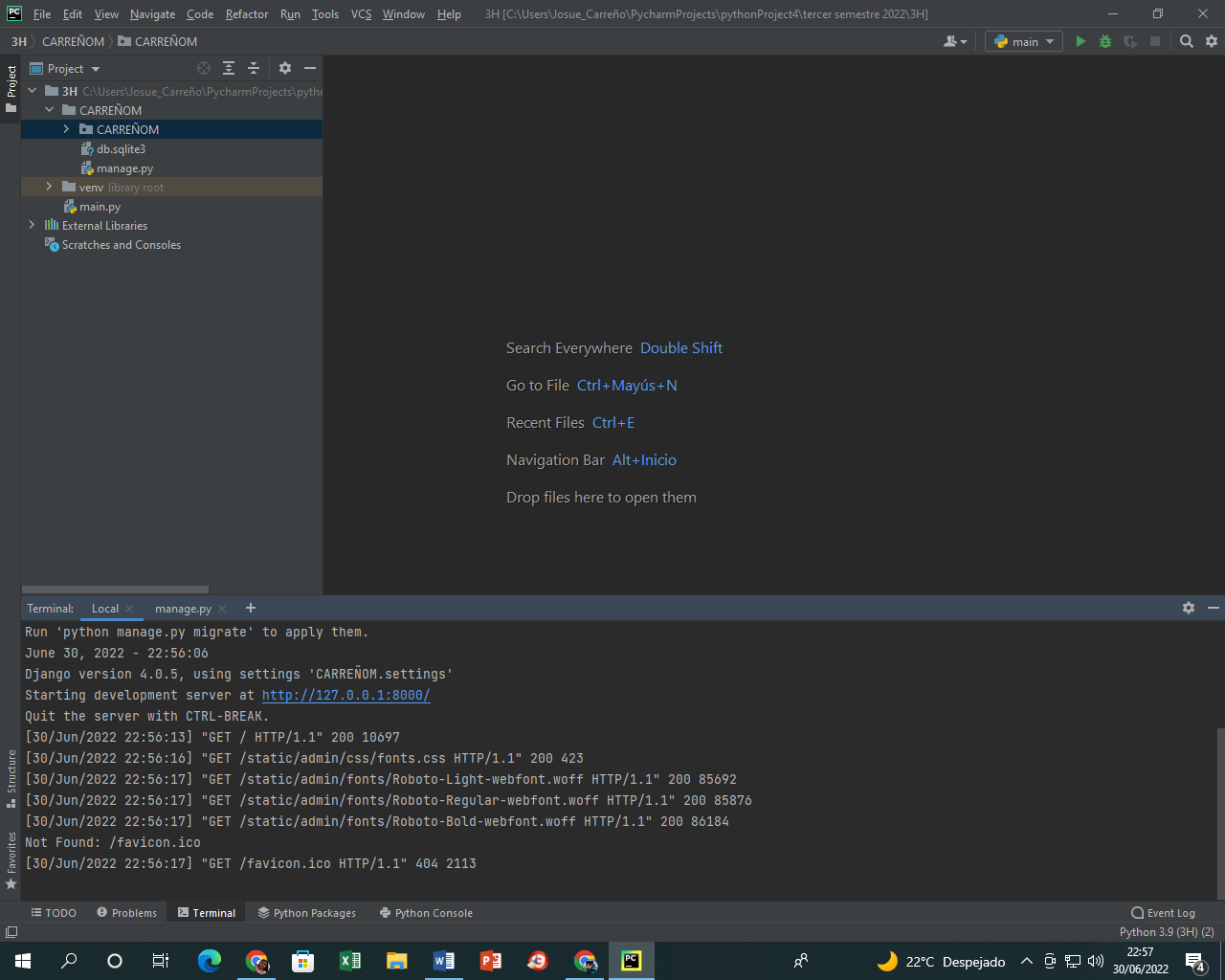
Se procede a descargar el paquete de Django y también instalar el paquete pillow el cual servirá para después

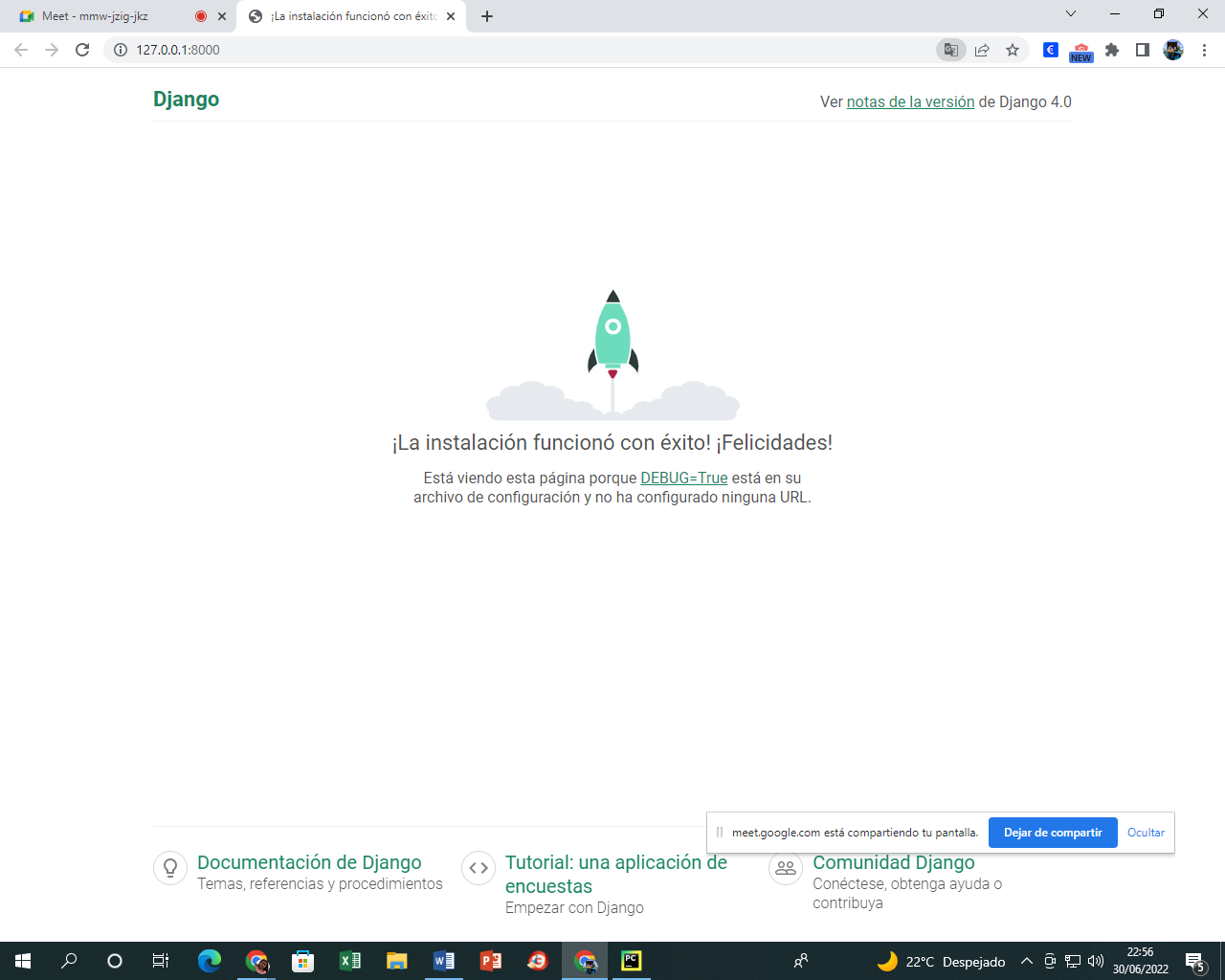
1. Crear un proyecto para programar en Django



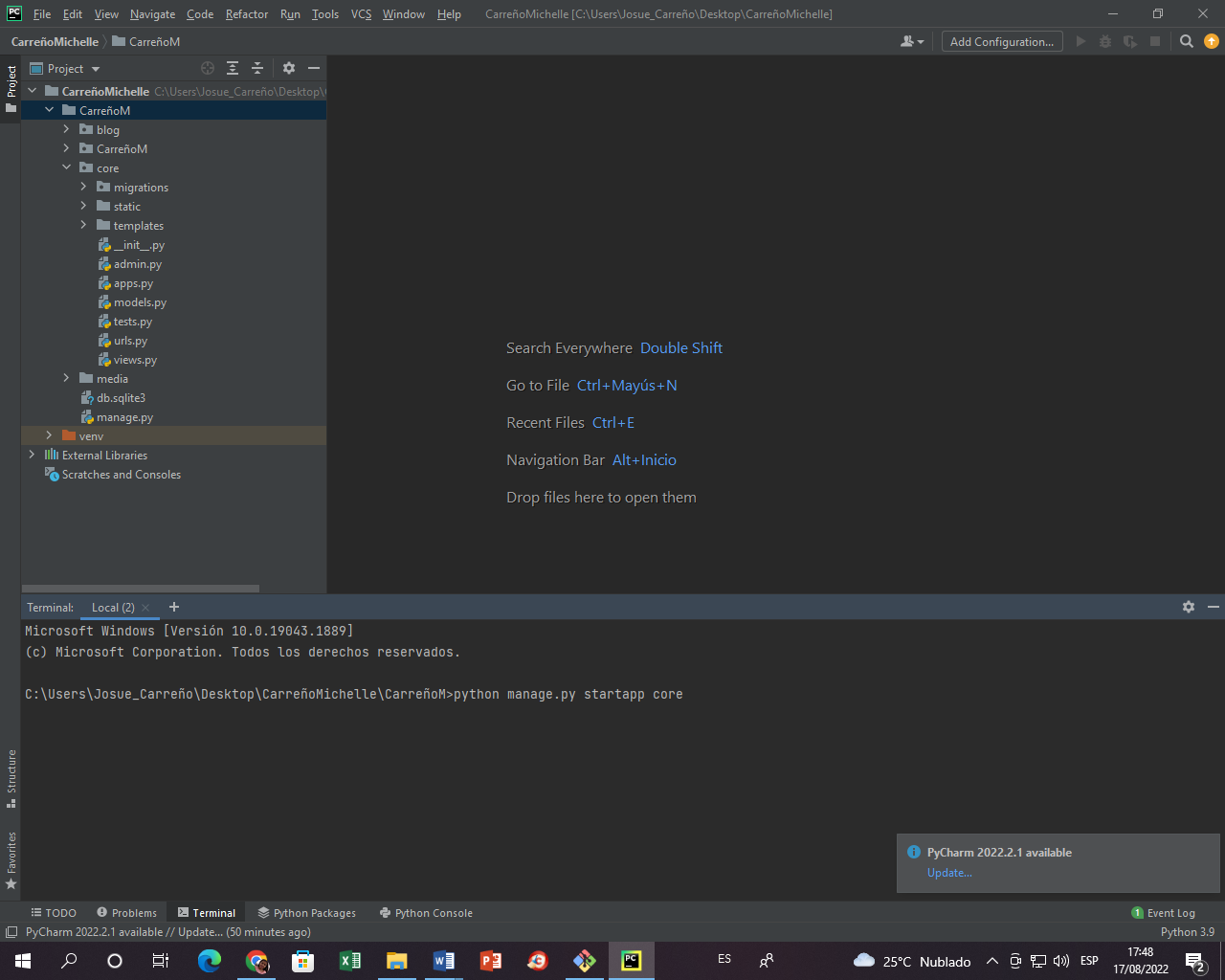
Para crear el proyecto se procede a crear otro proyecto dentro del original para eso utilizamos startproyect (el nombre que quieran poner a el proyecto)

1. Ejecutar el proyecto y el mensaje de felicitaciones.





1. Crear una Apps core.



Para crear una core apps se usa ese comando python manage.py startapp core

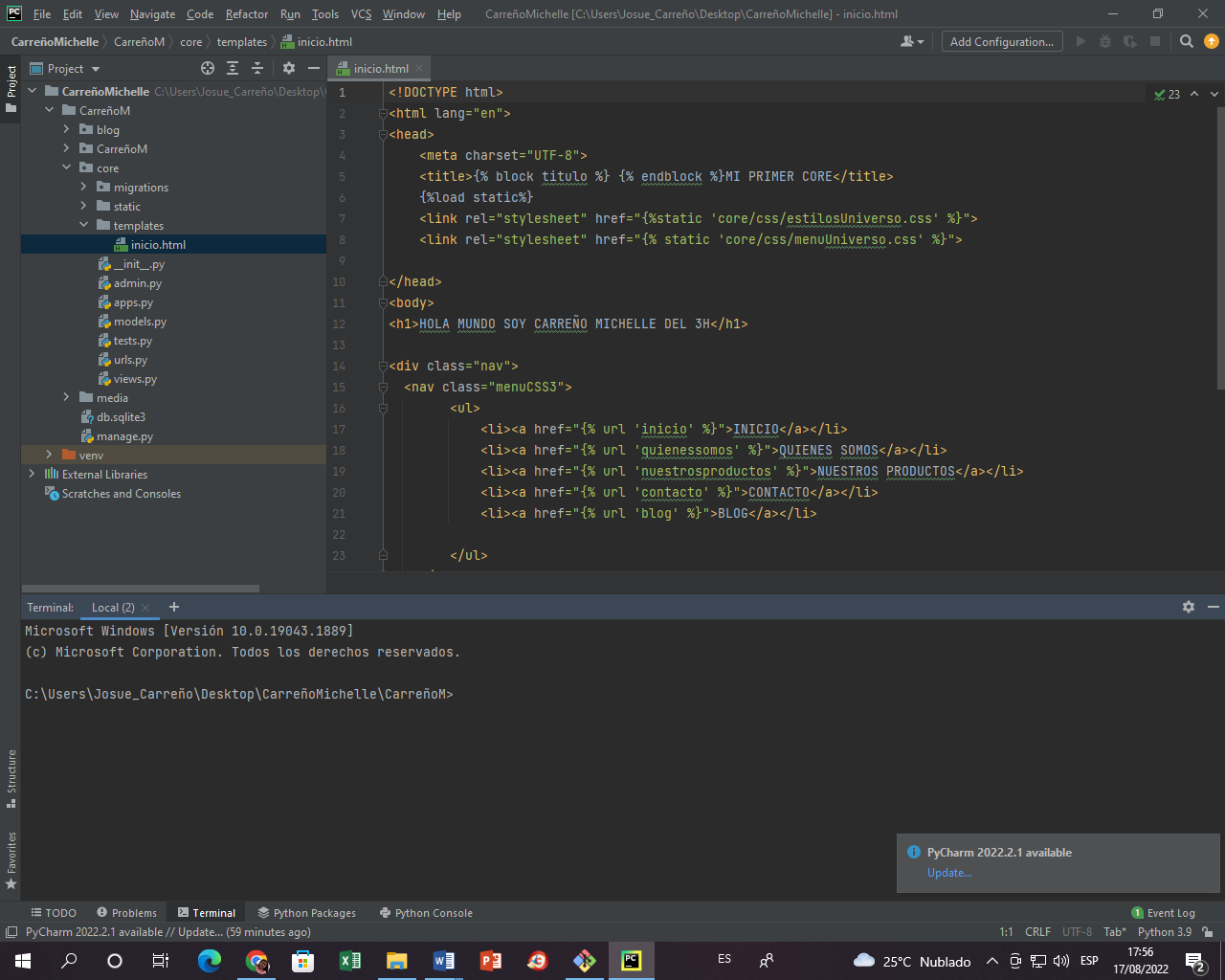
1. ¿Qué es la Carpeta Templates?

En la carpeta Templates, **las subcarpetas contienen las plantillas que se muestran en el cuadro de diálogo Crear nuevo archivo**. Puede crear y guardar plantillas personalizadas en la carpeta Templates.

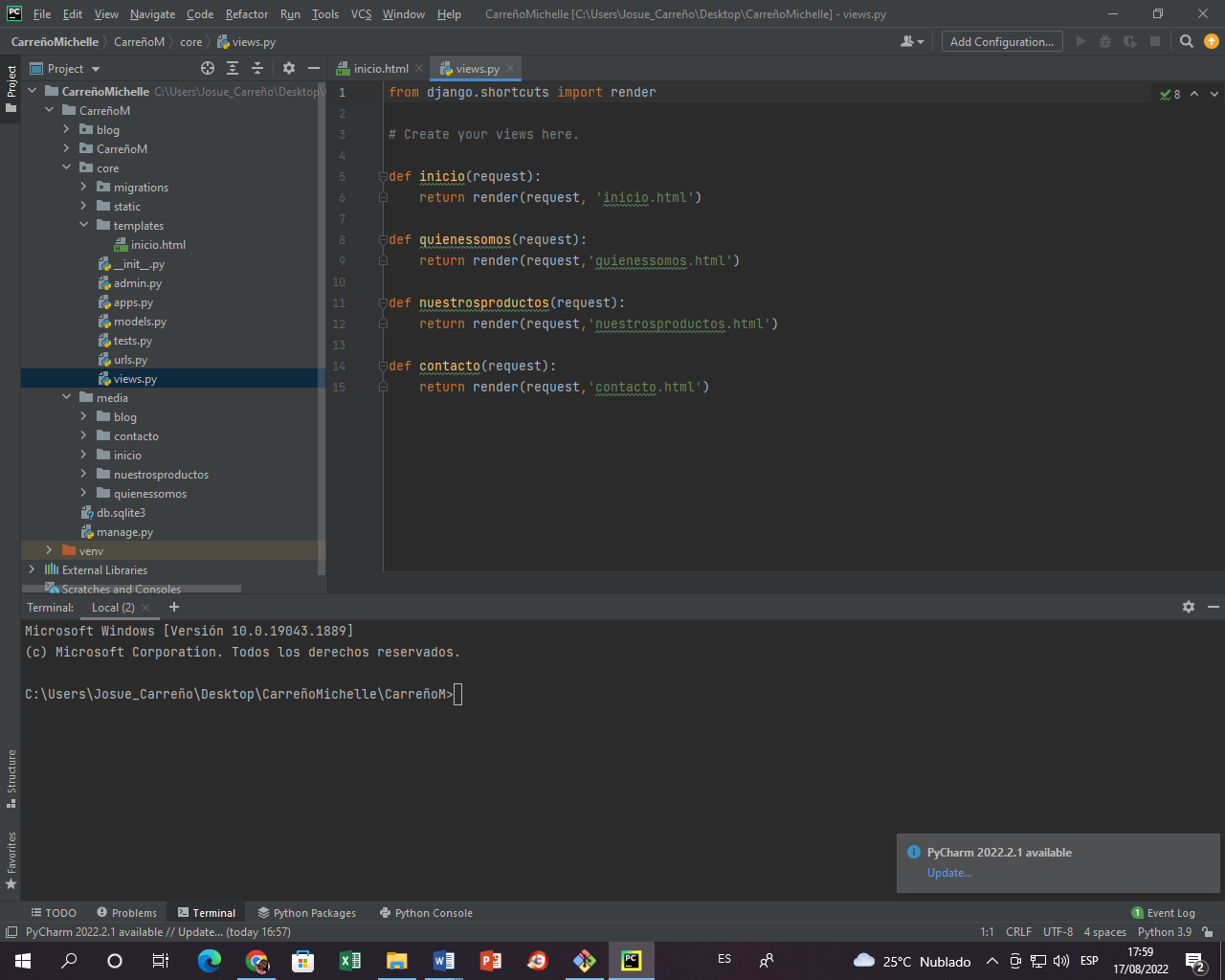
1. ¿Qué es la Carpeta stactic?

Esta carpeta **contiene todo los archivos de estilos: css y scss, archivos de comportamientos JavaScript, imágenes incluidas dentro del theme y el archivo checkout**.

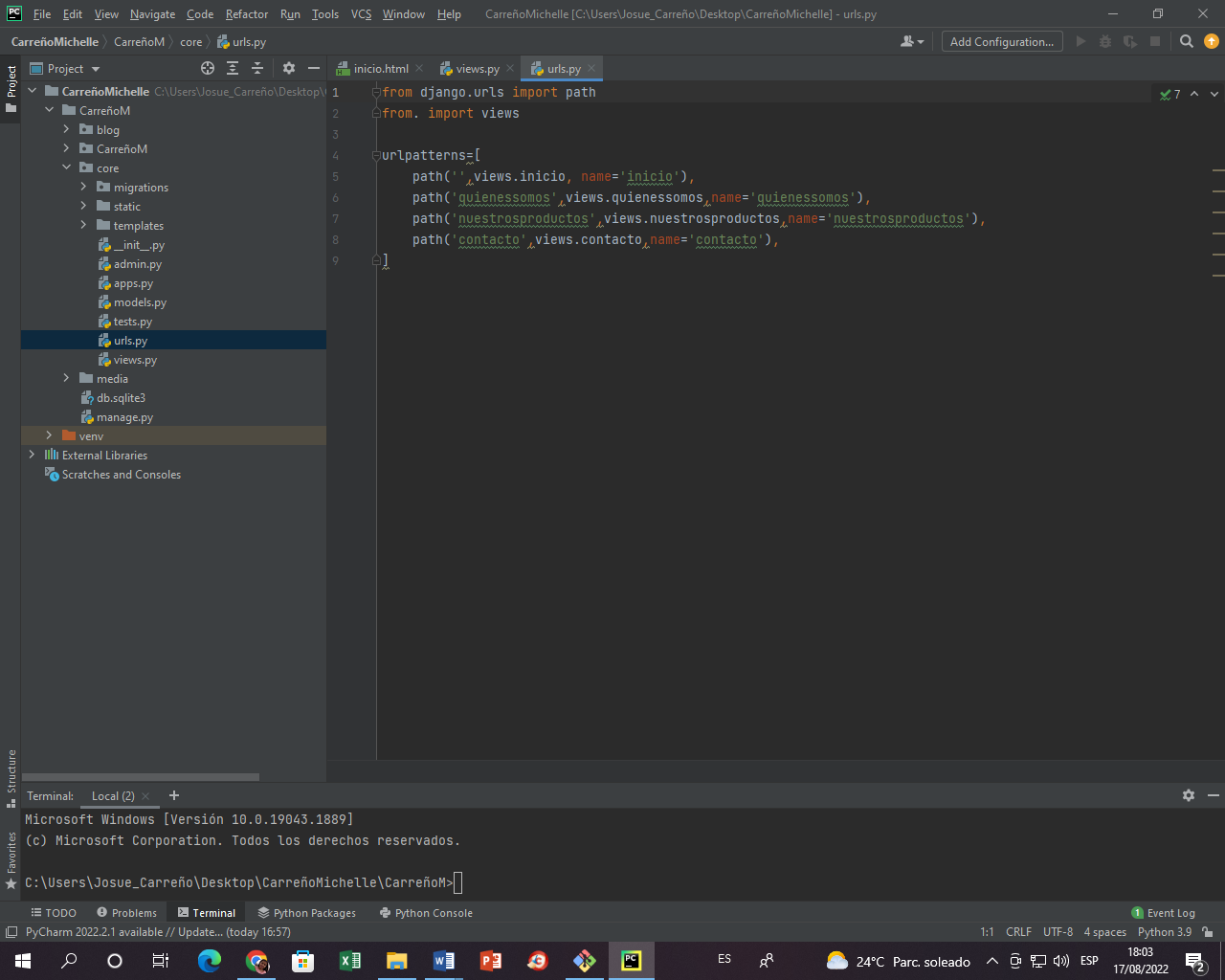
11) Crear un archivo base html en la APPS core



Dentro de nuestro Proyecto Core, creamos la carpeta templates dentro de ella nuestro Archivo html

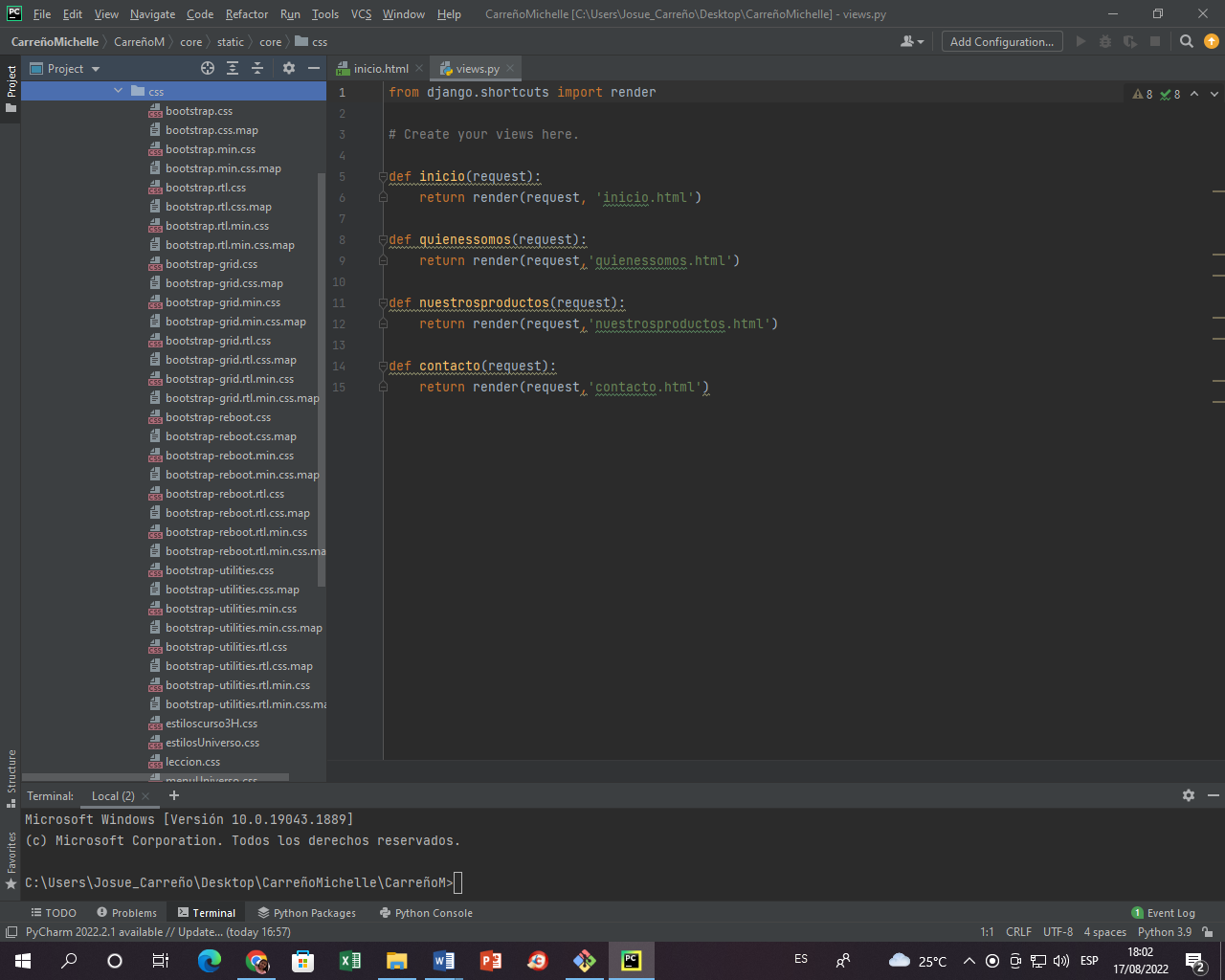


Dentro de nuestro Proyecto Core, Modificamos el Views, llamamos a nuestro Html

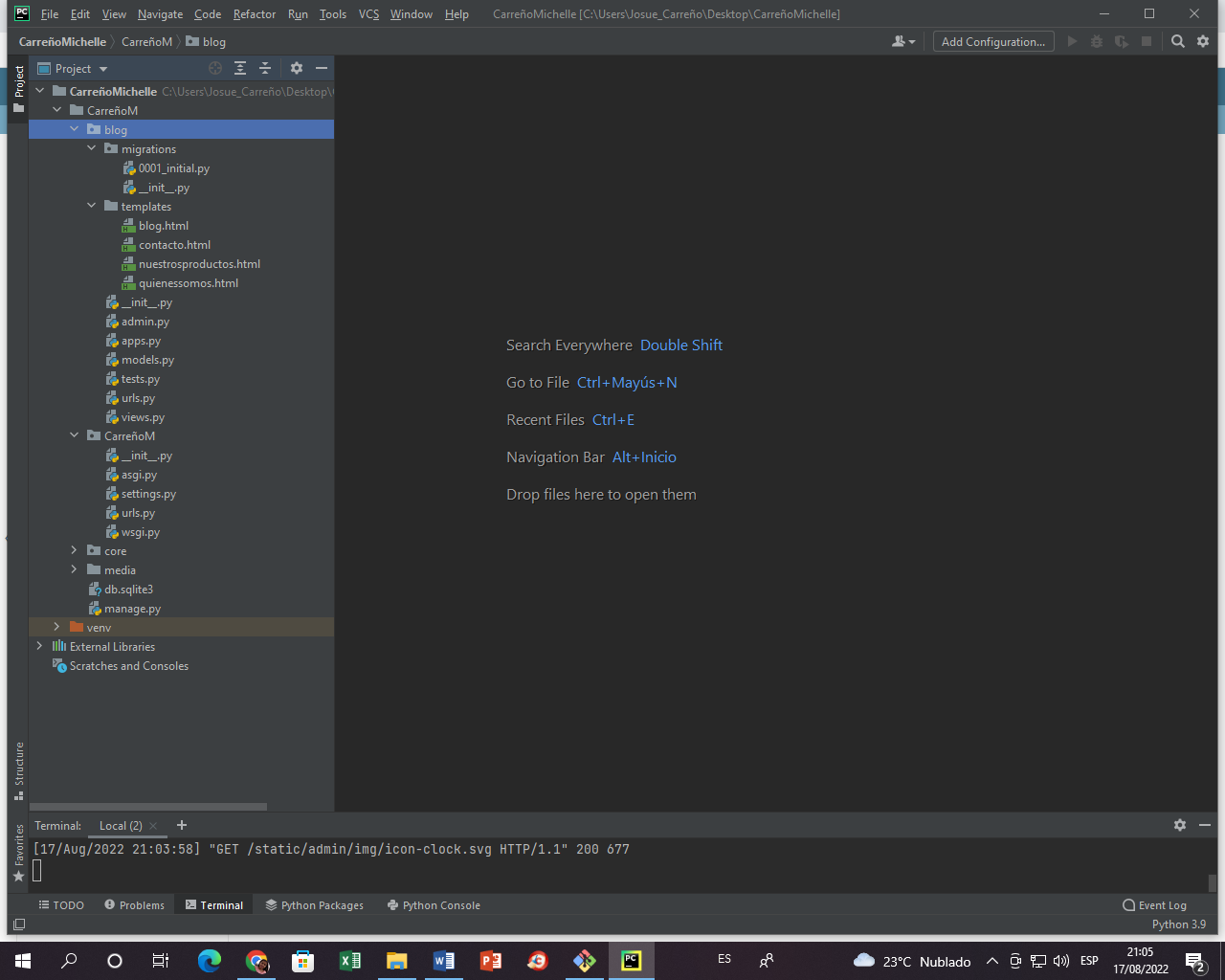


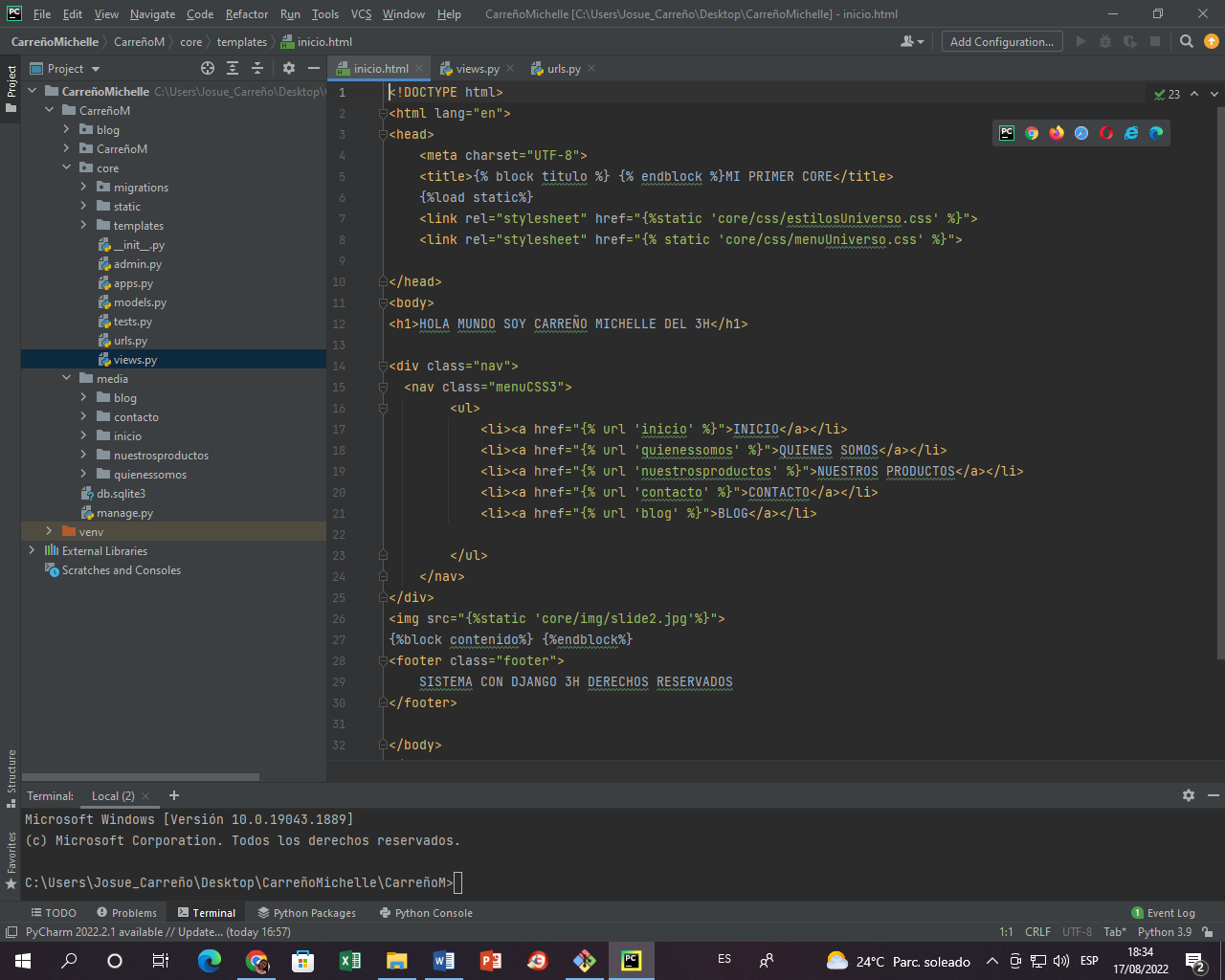
Dentro de nuestro Proyecto Core, Creamos el Urls.py

12) Como se llaman a los CSS desde el archivo base html.

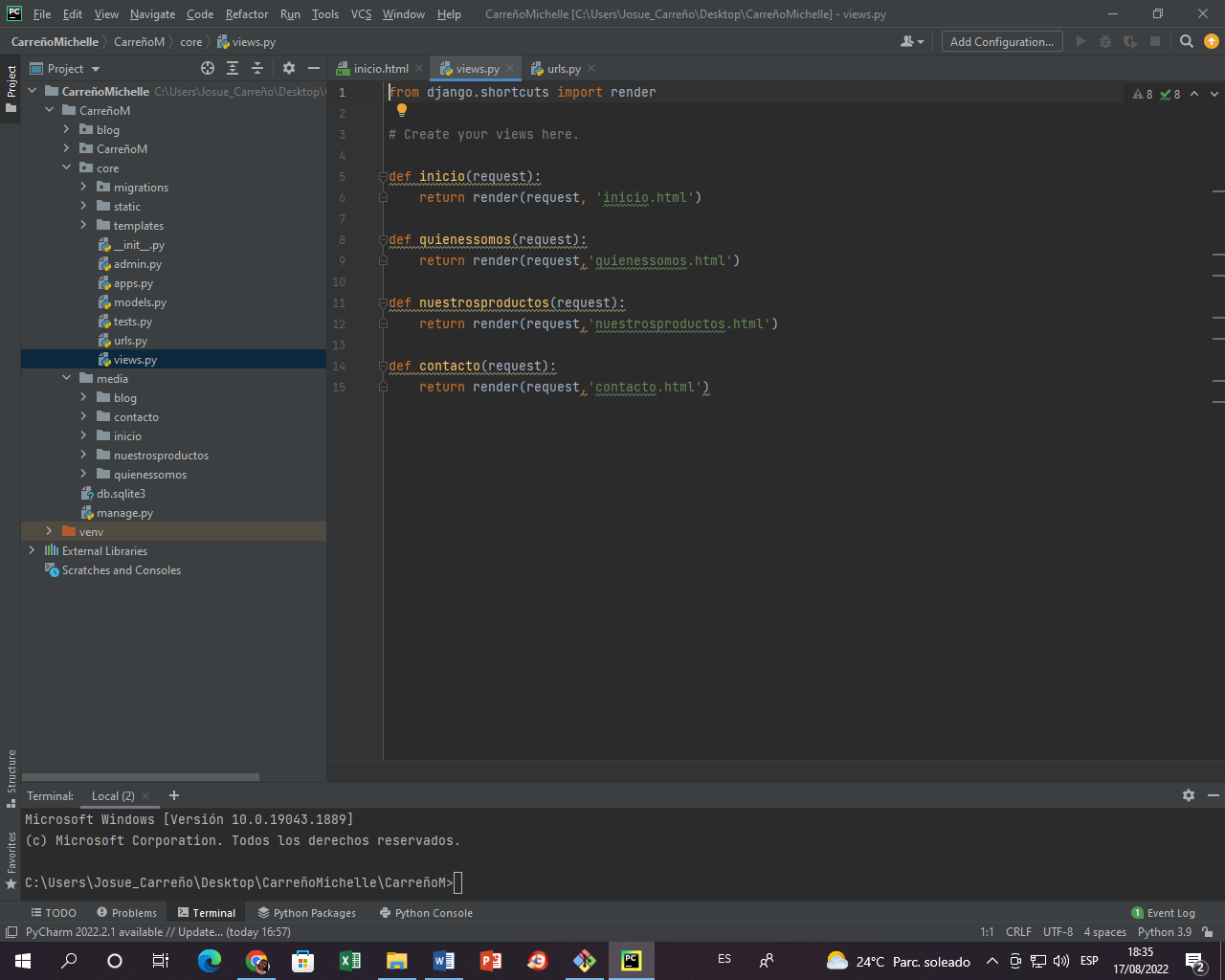


13) Como consume un archivo hijo html al utilizar la herencia del archivo base html.



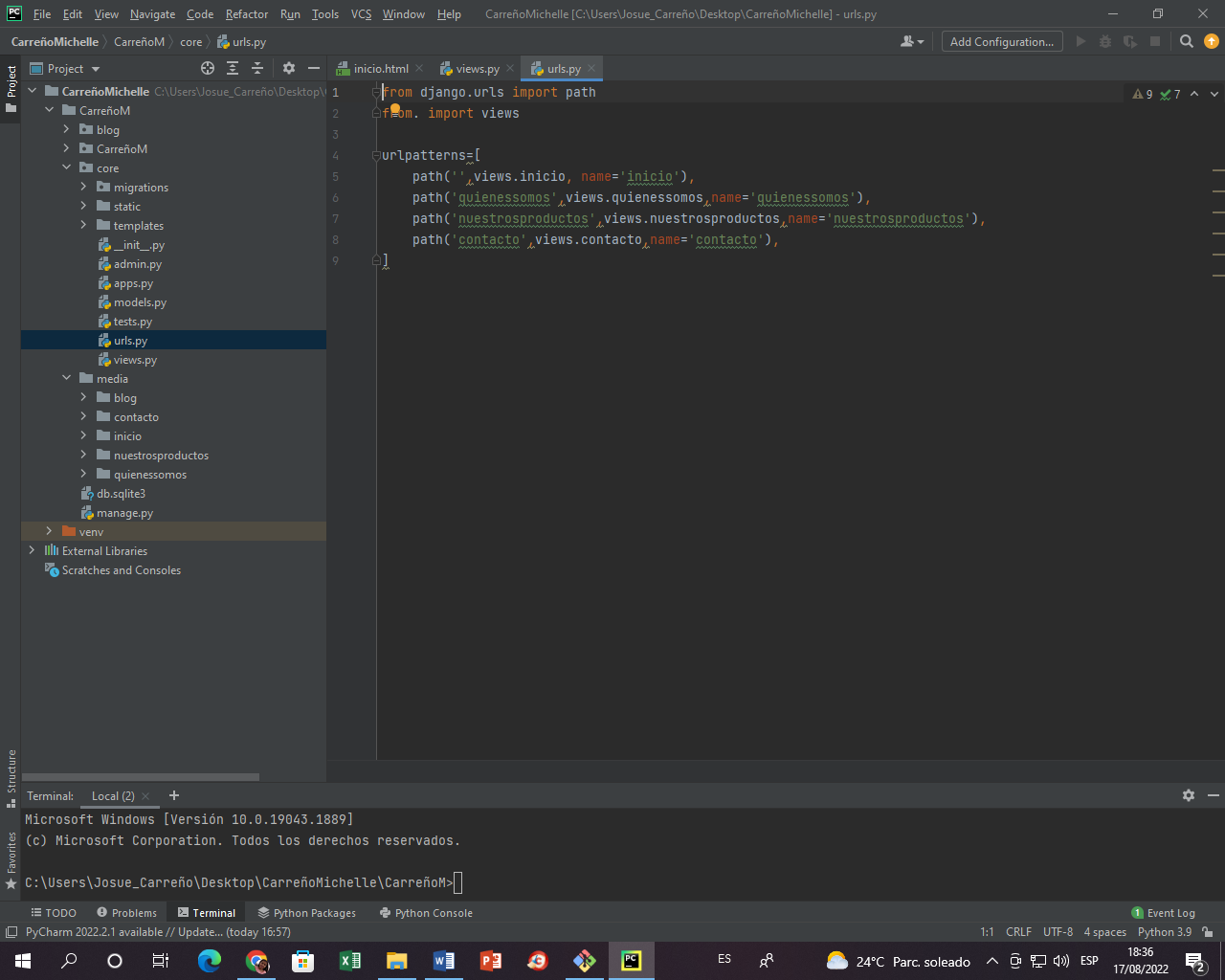


14) Crear un view que llame al html hijo



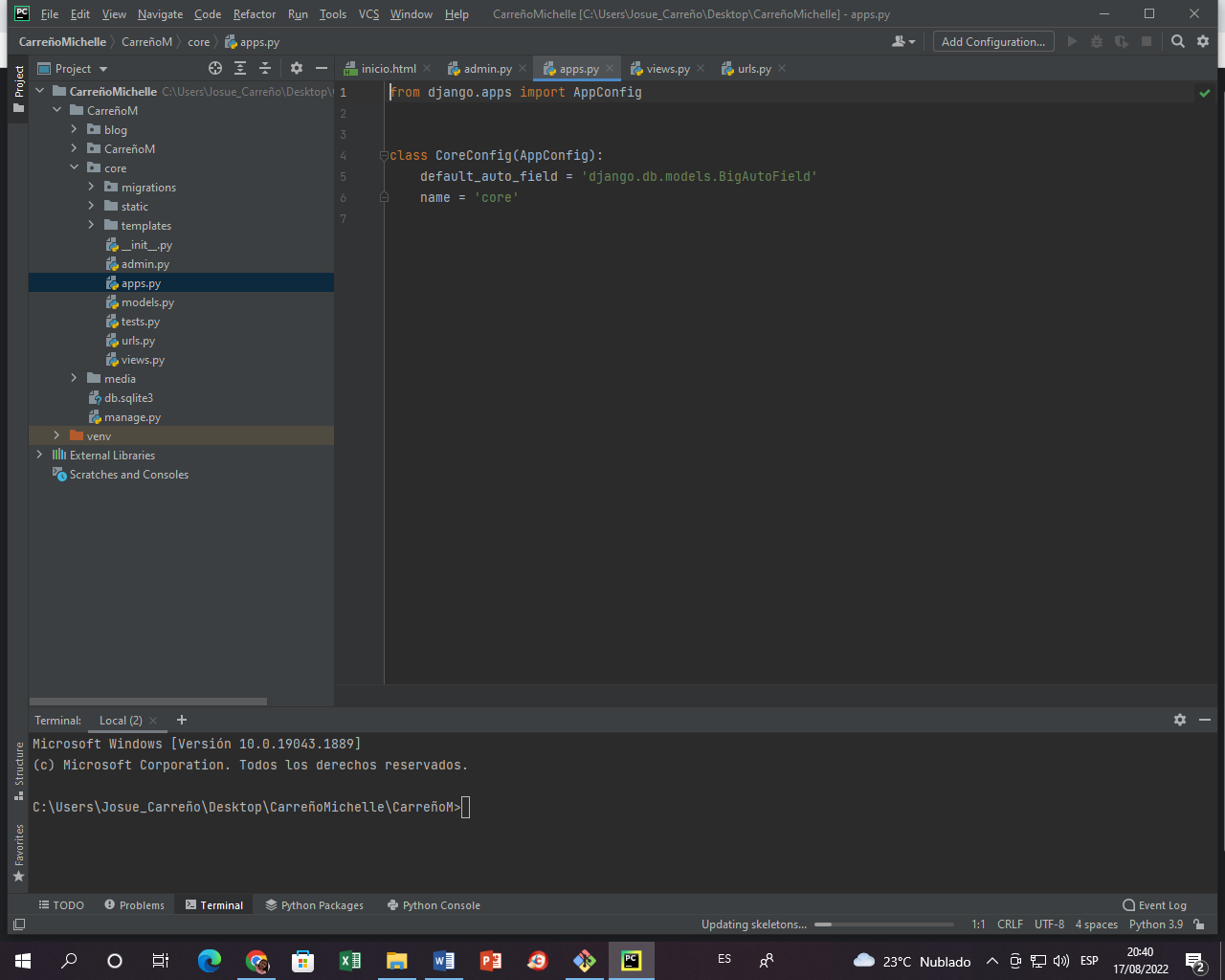
Creamos un views para que llama a todos los html hijo que eh creado

15) Crear la urls que llame al views.

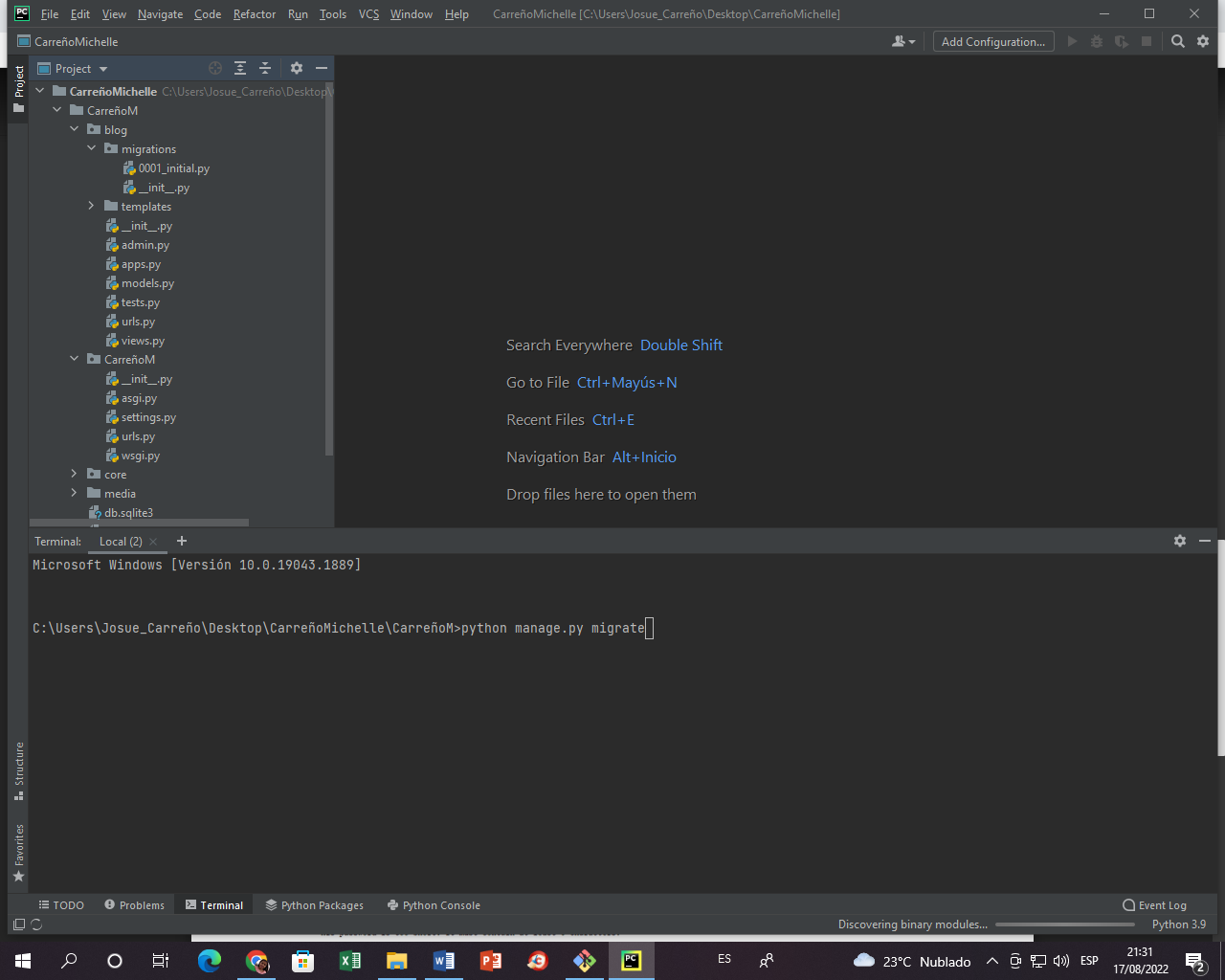


Creamos el urls para que llame a el views

16) Integrar la aplicación APPS core al proyecto principal

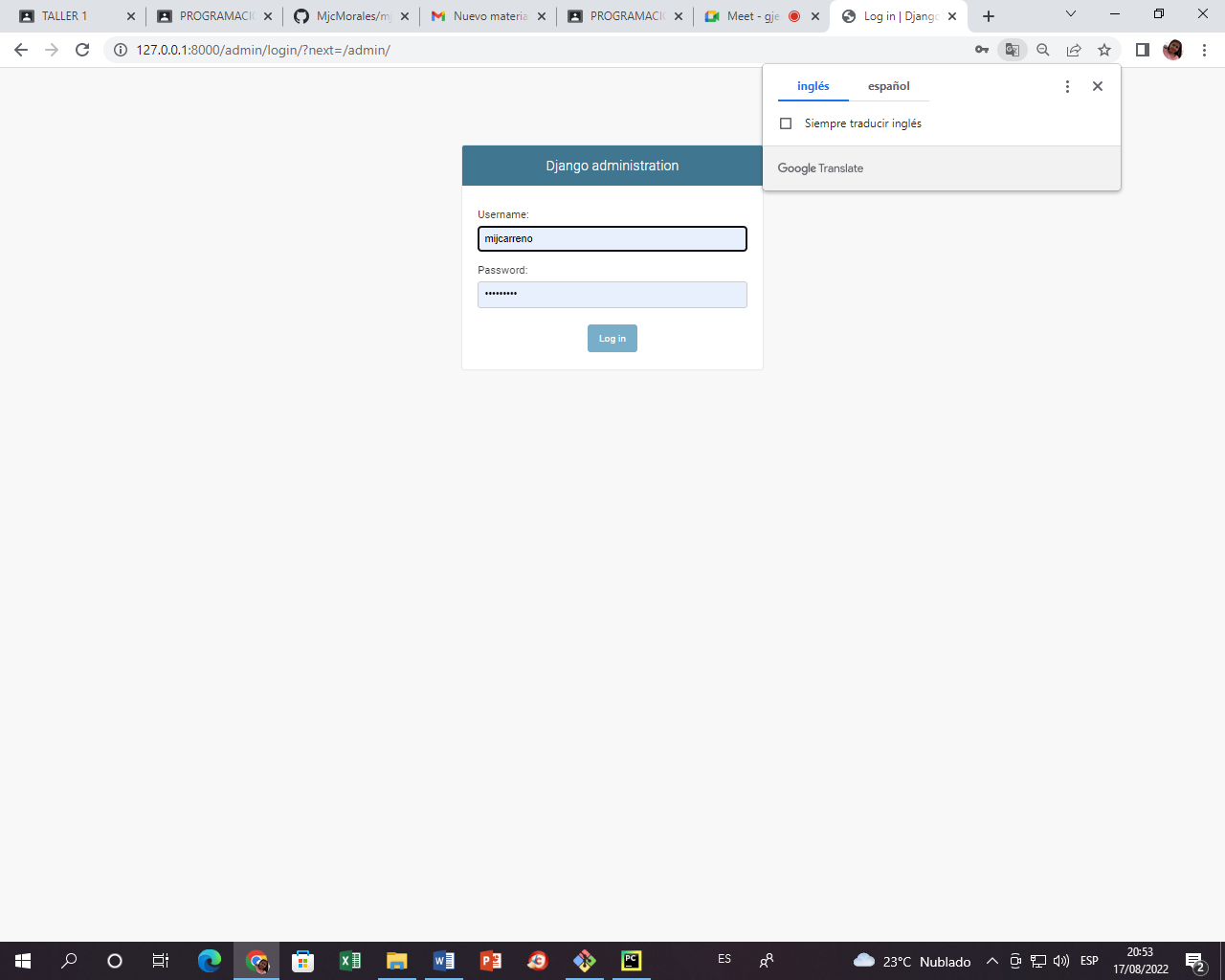


17) Crear las tablas del sistema de usuarios para utilizar el panel de administración.



Usamos este comando (python manage.py migrate) utilizando este codigo hace que se crean las tablas para que se crean las bases de datos

18) Crear un usuario para poder ingresar al Panel de Administración



En esta imagen se ve creando un usuario

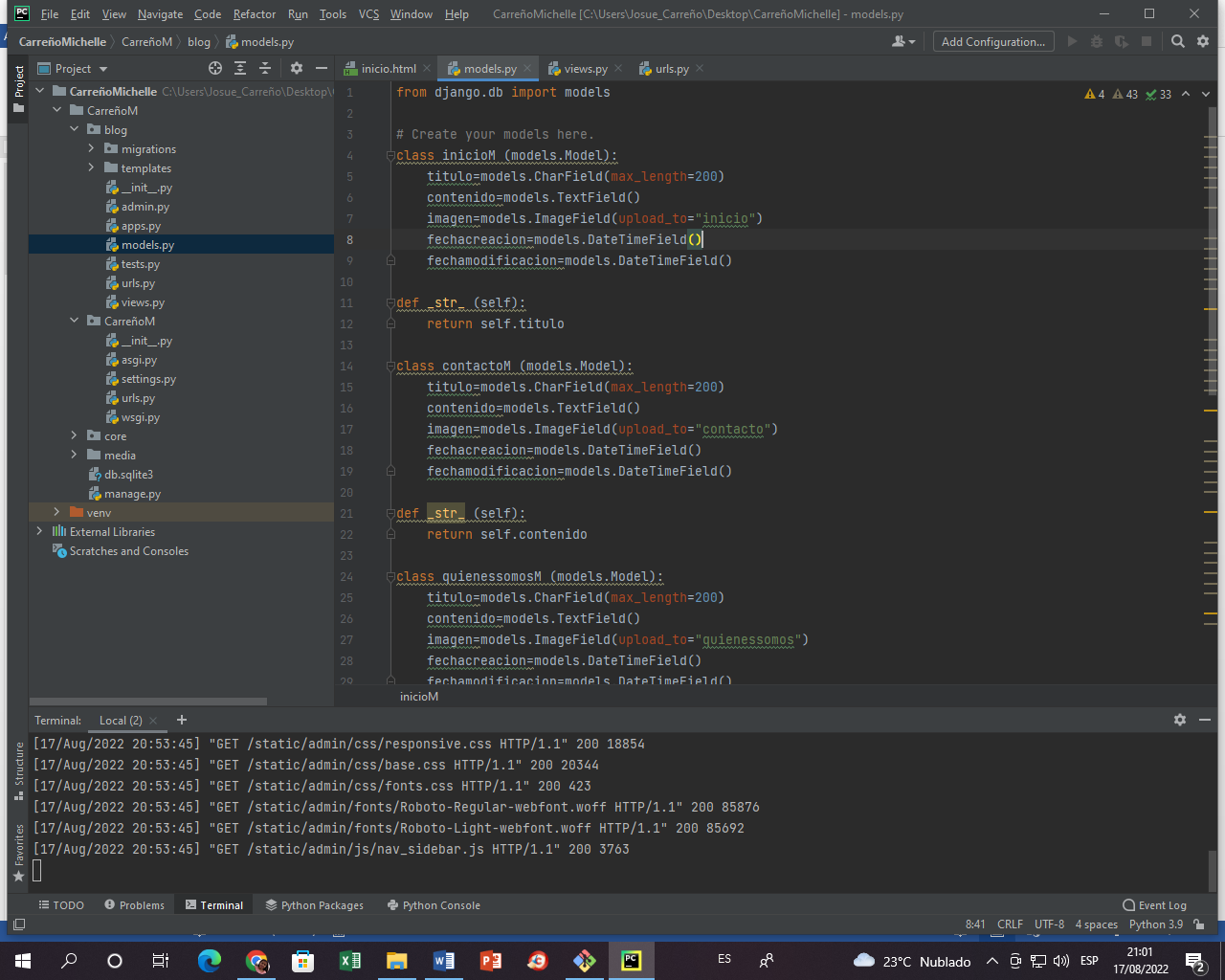
19) Que es un modelo en Django

Los modelos están definidos, normalmente, en el archivo models.py de la aplicación. Son

implementados como subclases de django.db.models.Model, y pueden incluir campos,

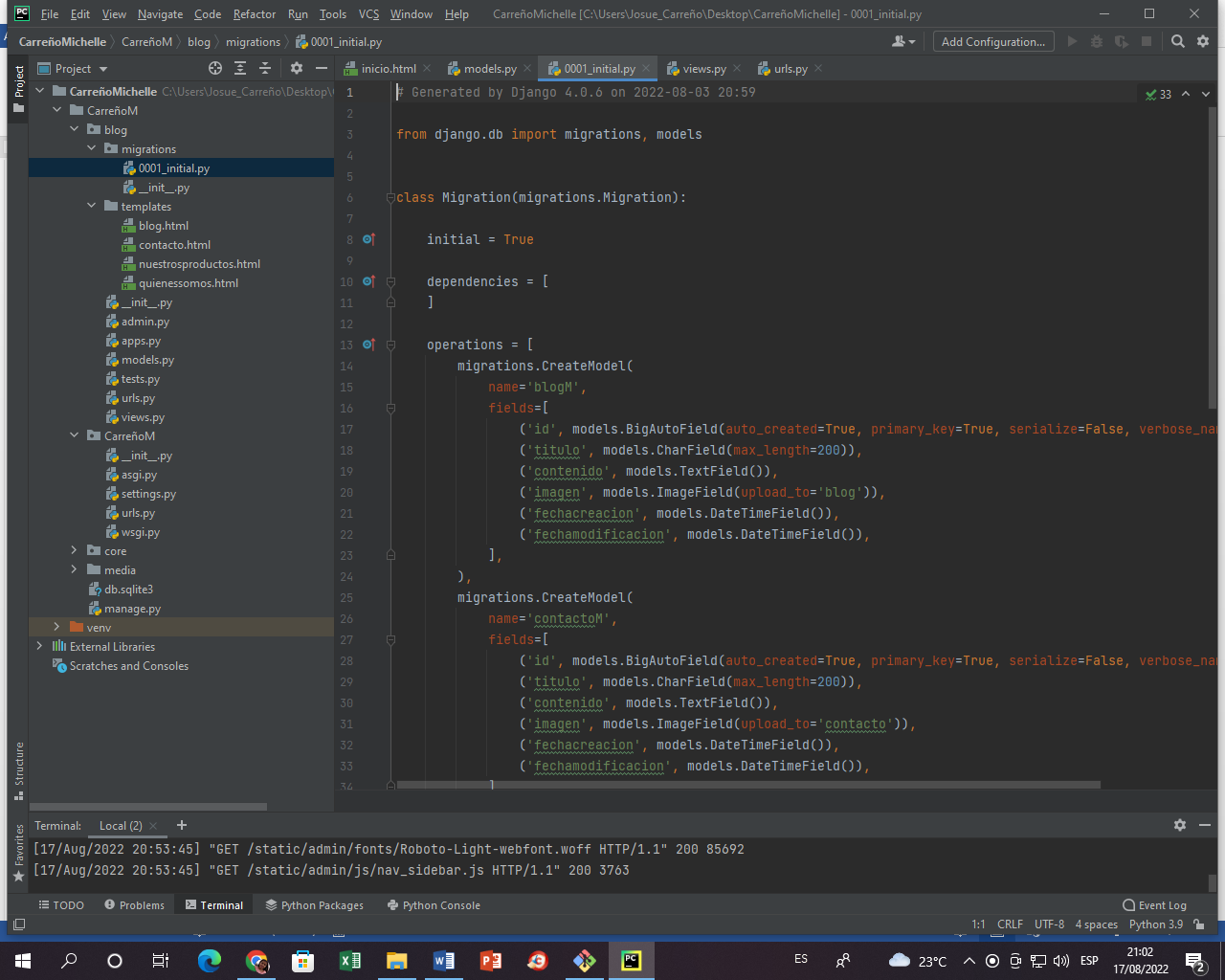
métodos y metadata. El fragmento de código más abajo muestra un modelo "típico"

20) Crear un modelo en Django.

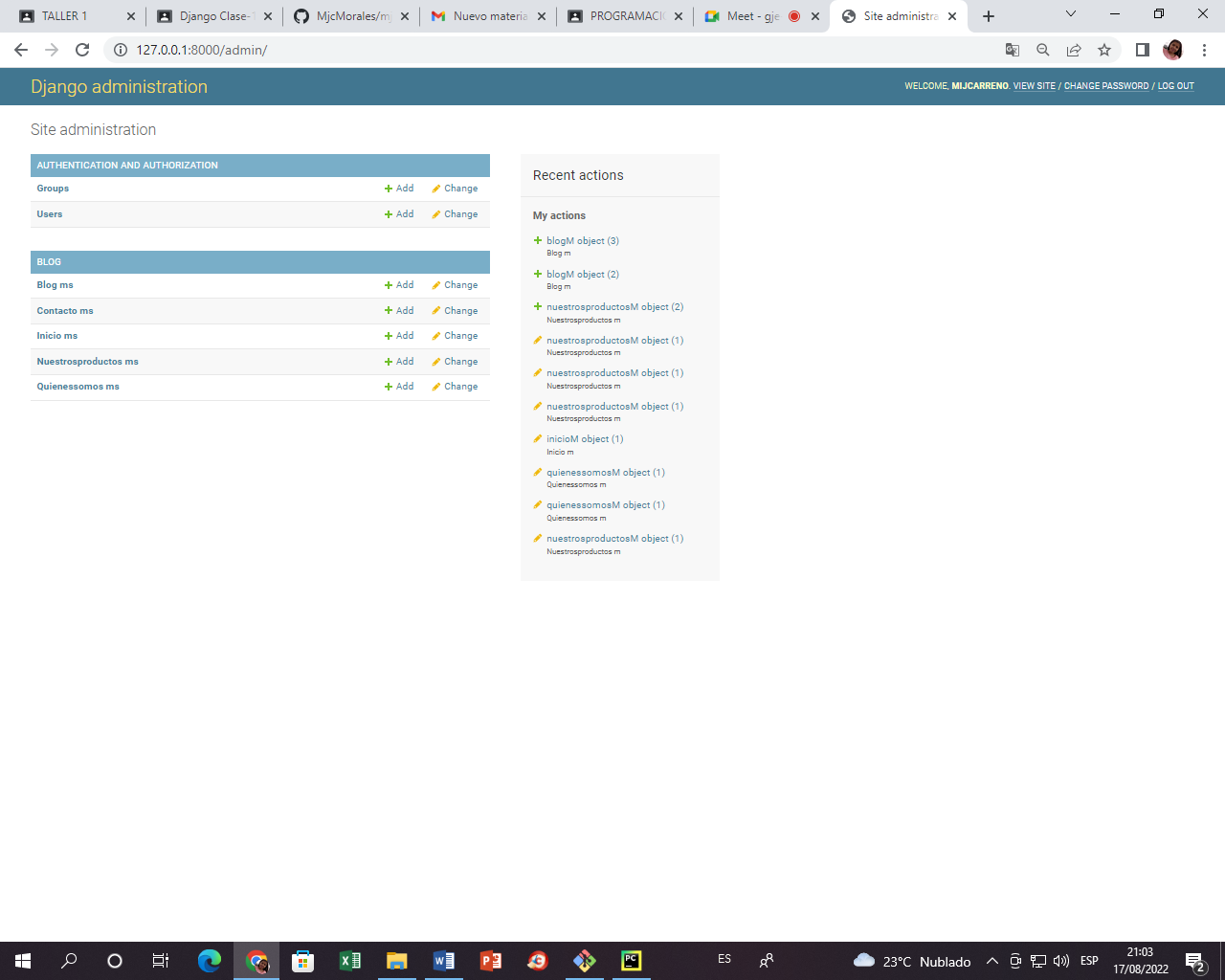


En esta imagen están todos los modelos que usamos en django

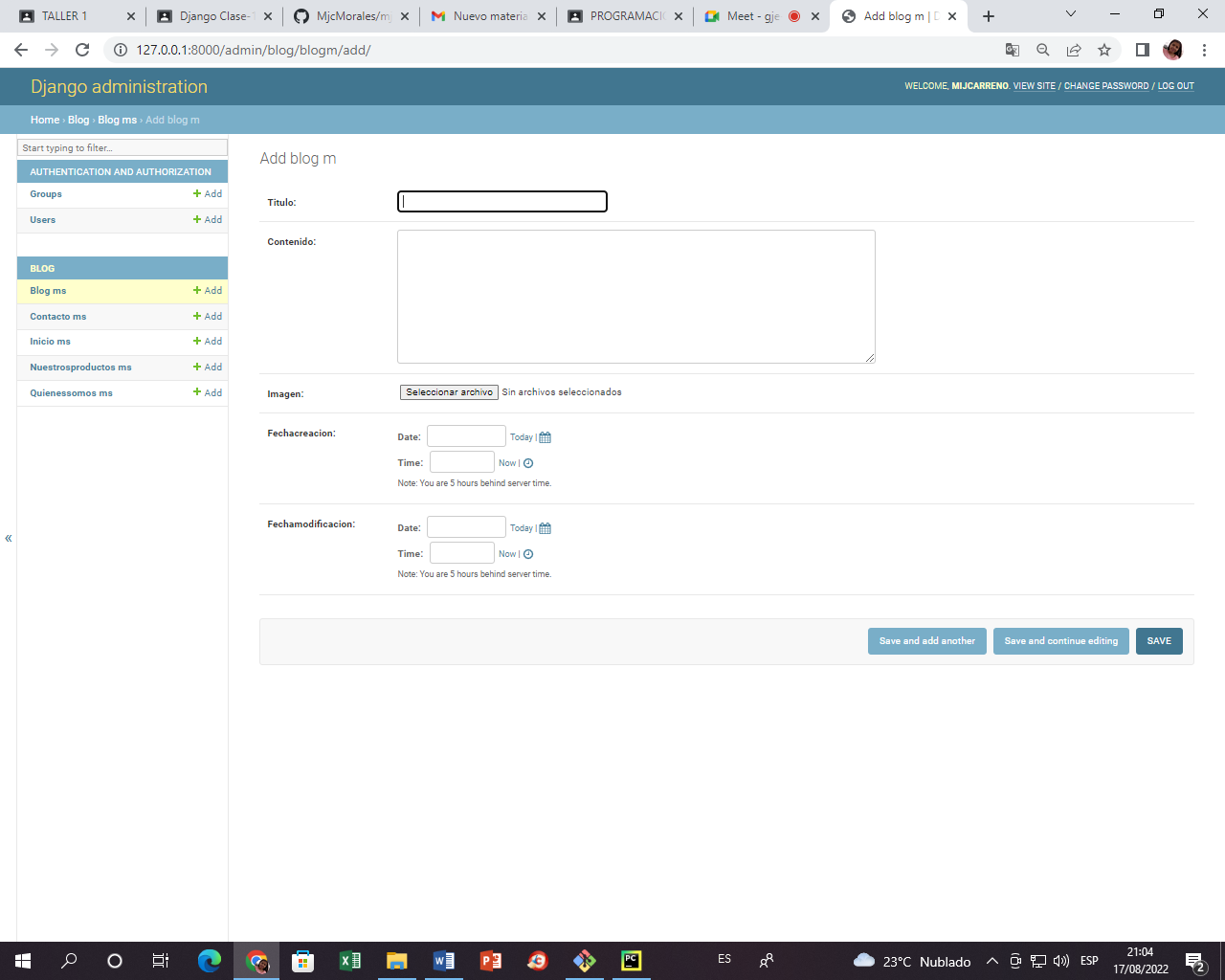
21) Migrar el Modelo a la base del Panel de Administración.



22) Integrar el Modelo al Panel de Administración.

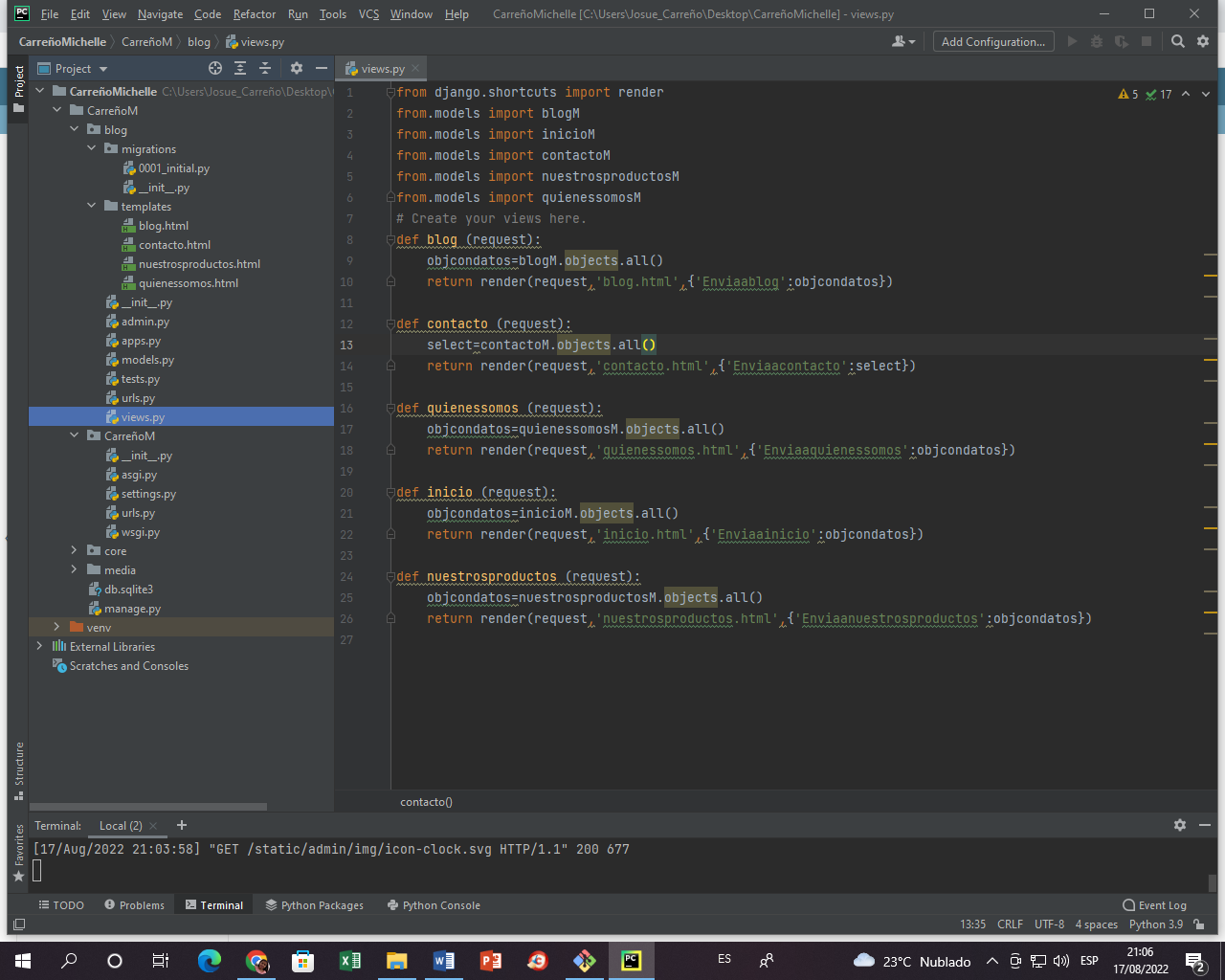


23) Ingresar información al modelo por el Panel de Administración.

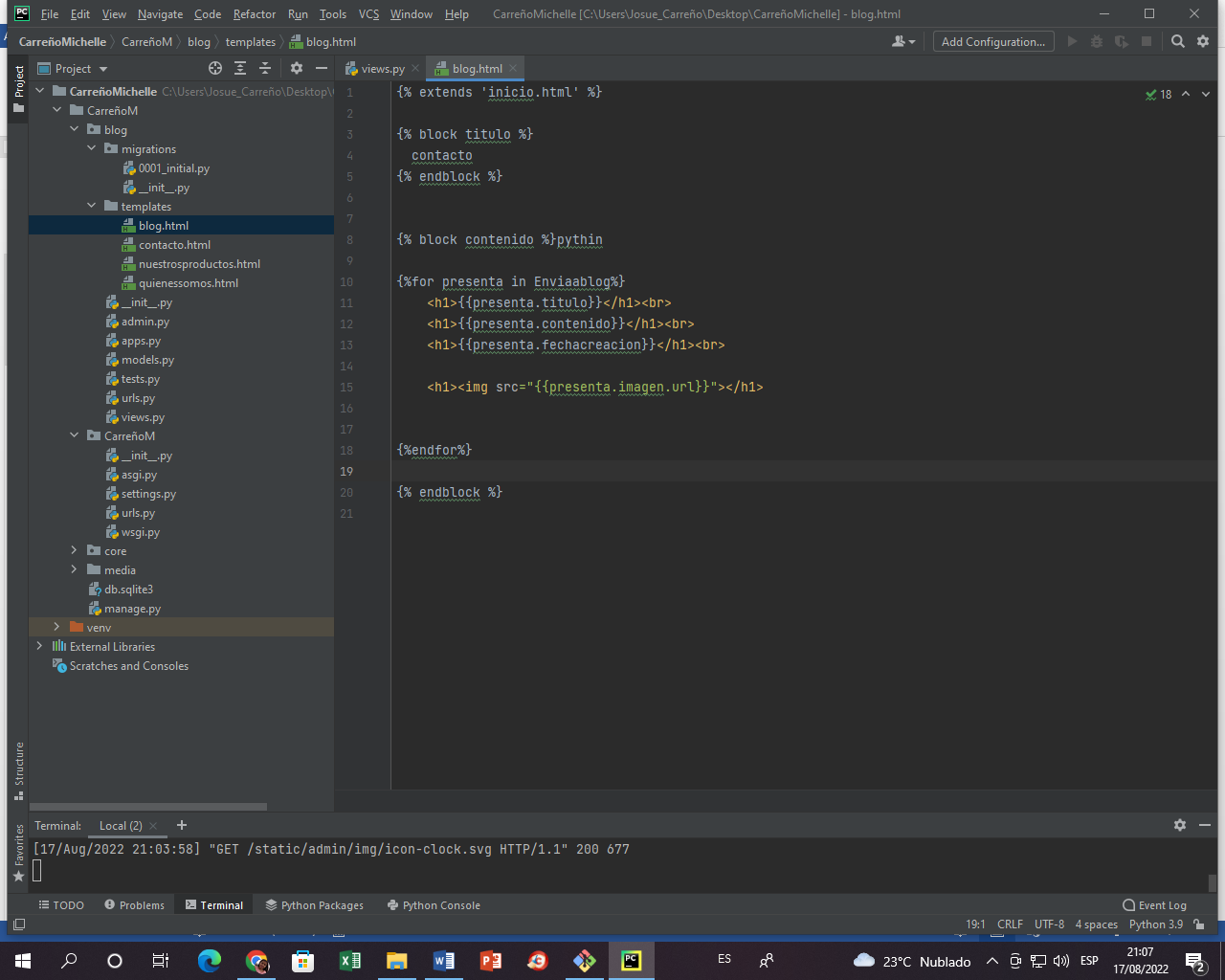


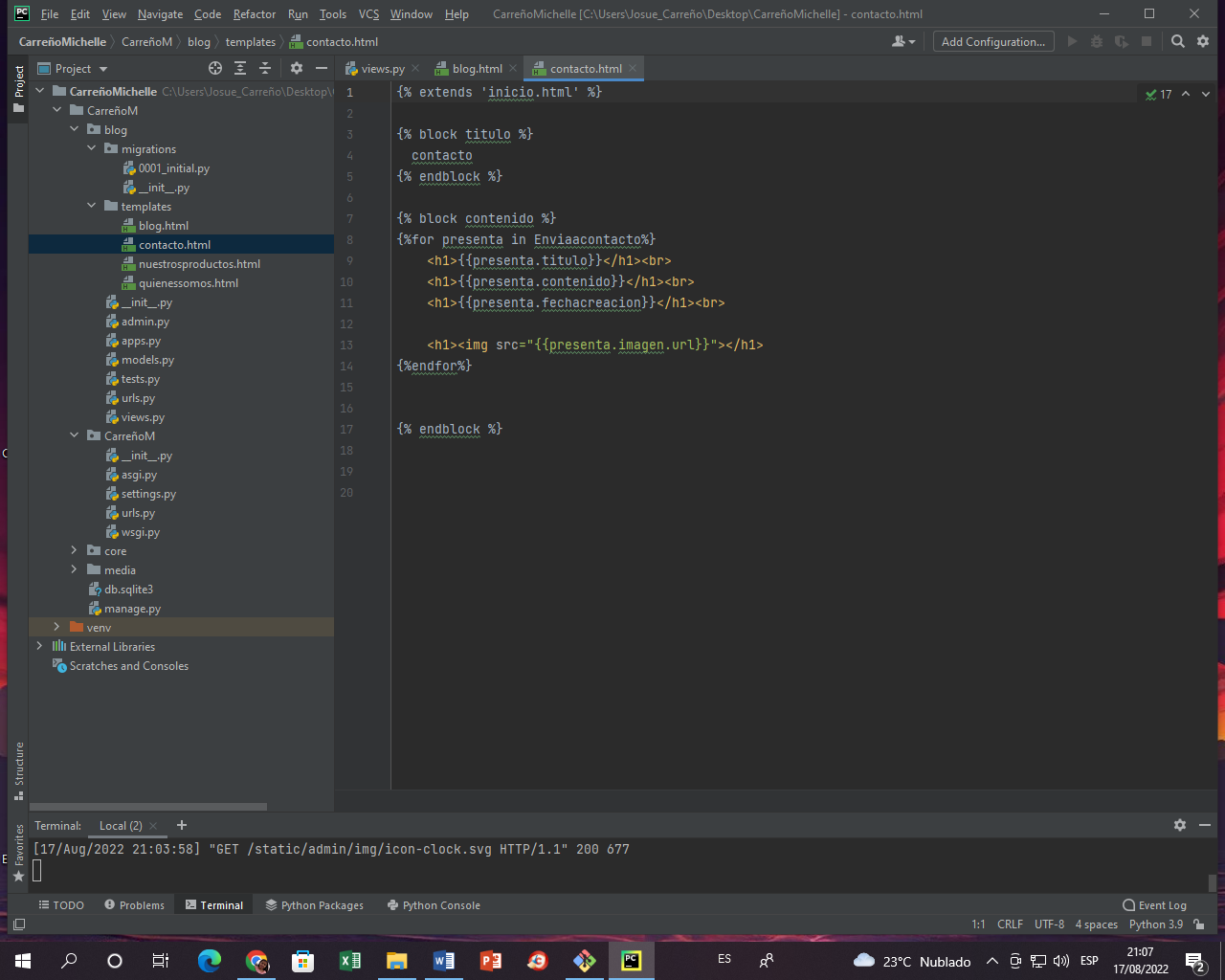
Aquí se puede ingresar o aumentar información en el panel

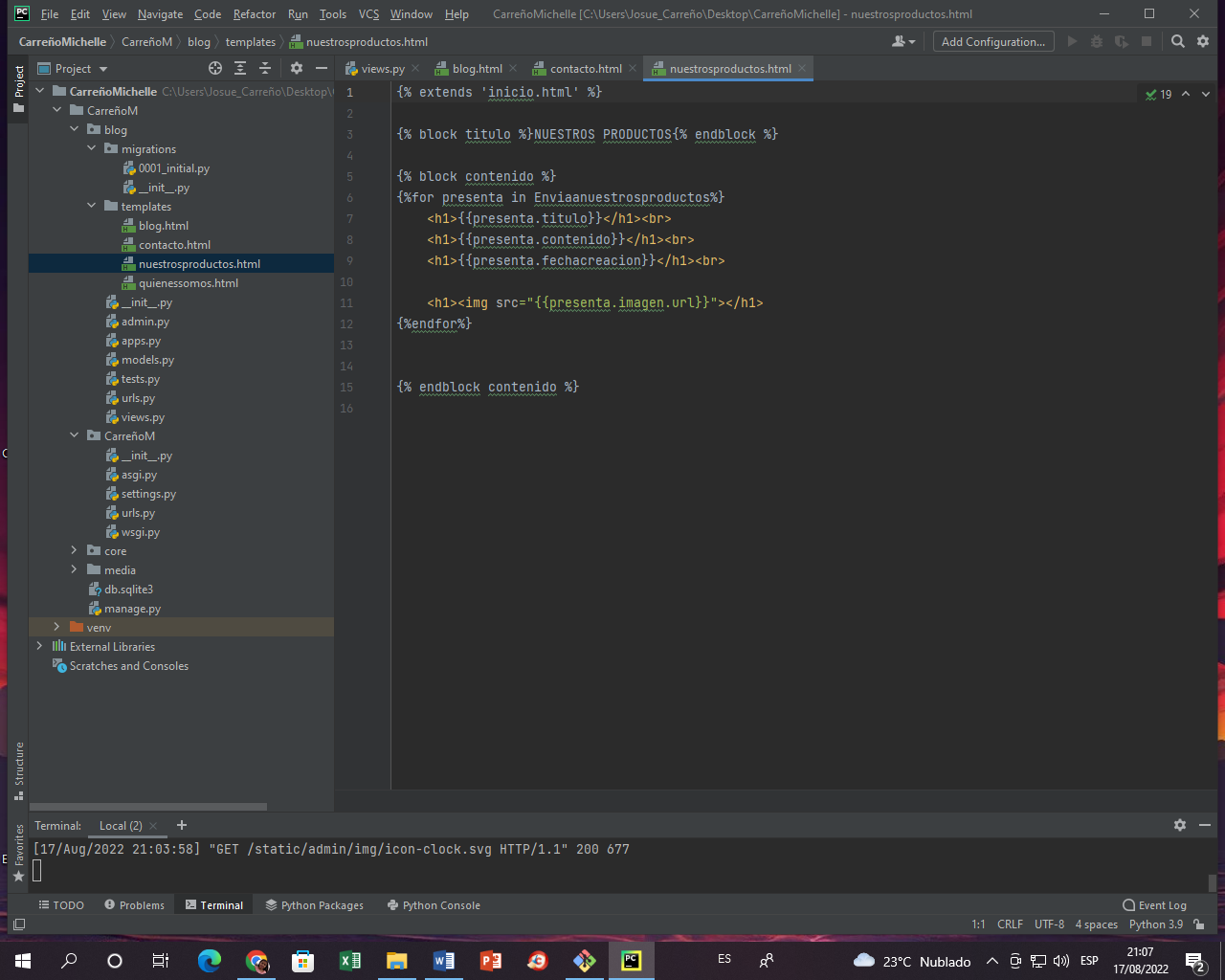
24) Realizar la consulta de todo lo ingresado en el modelo desde el Views.

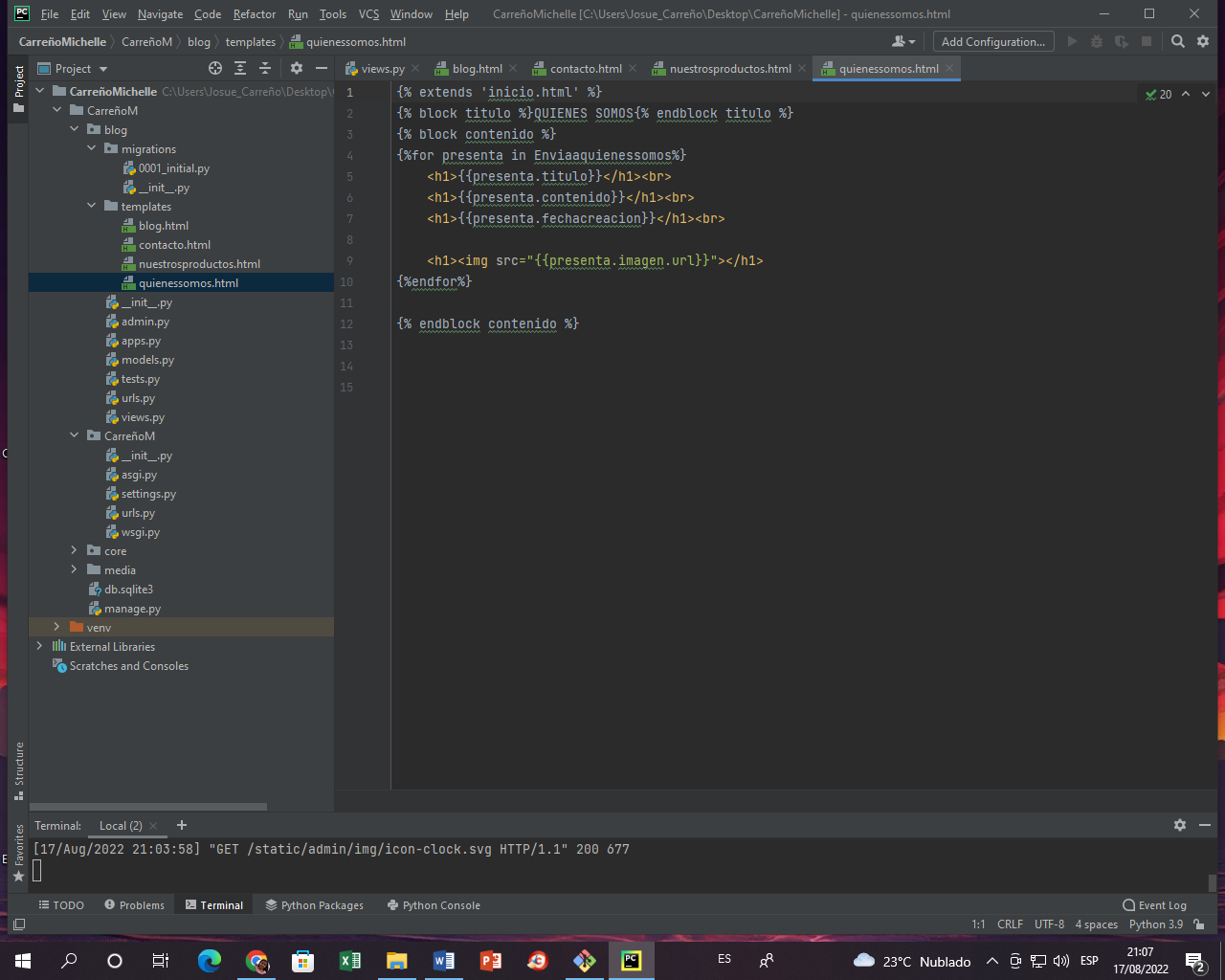


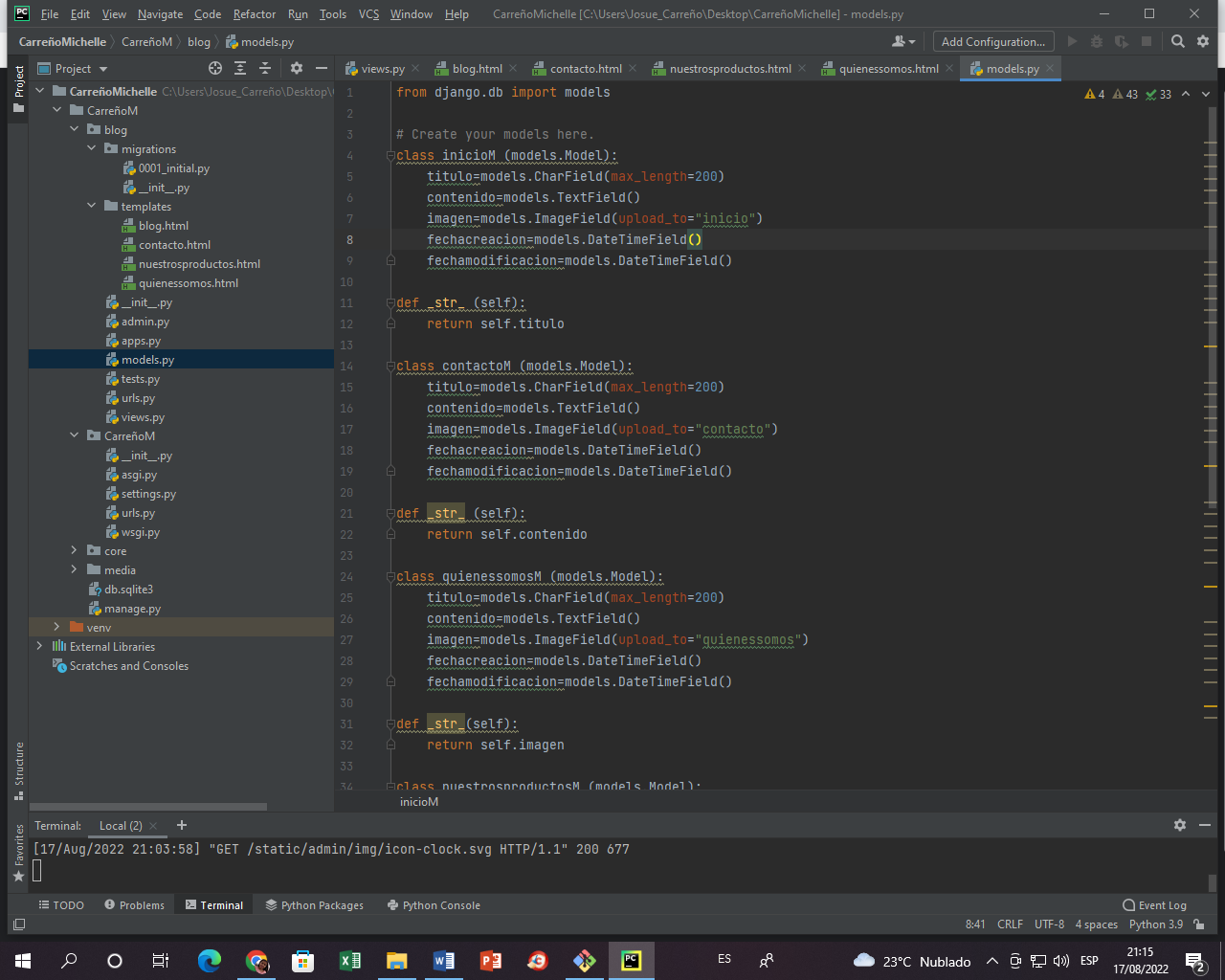
25) Mostrar los datos guardados en el modelo al html hijo.

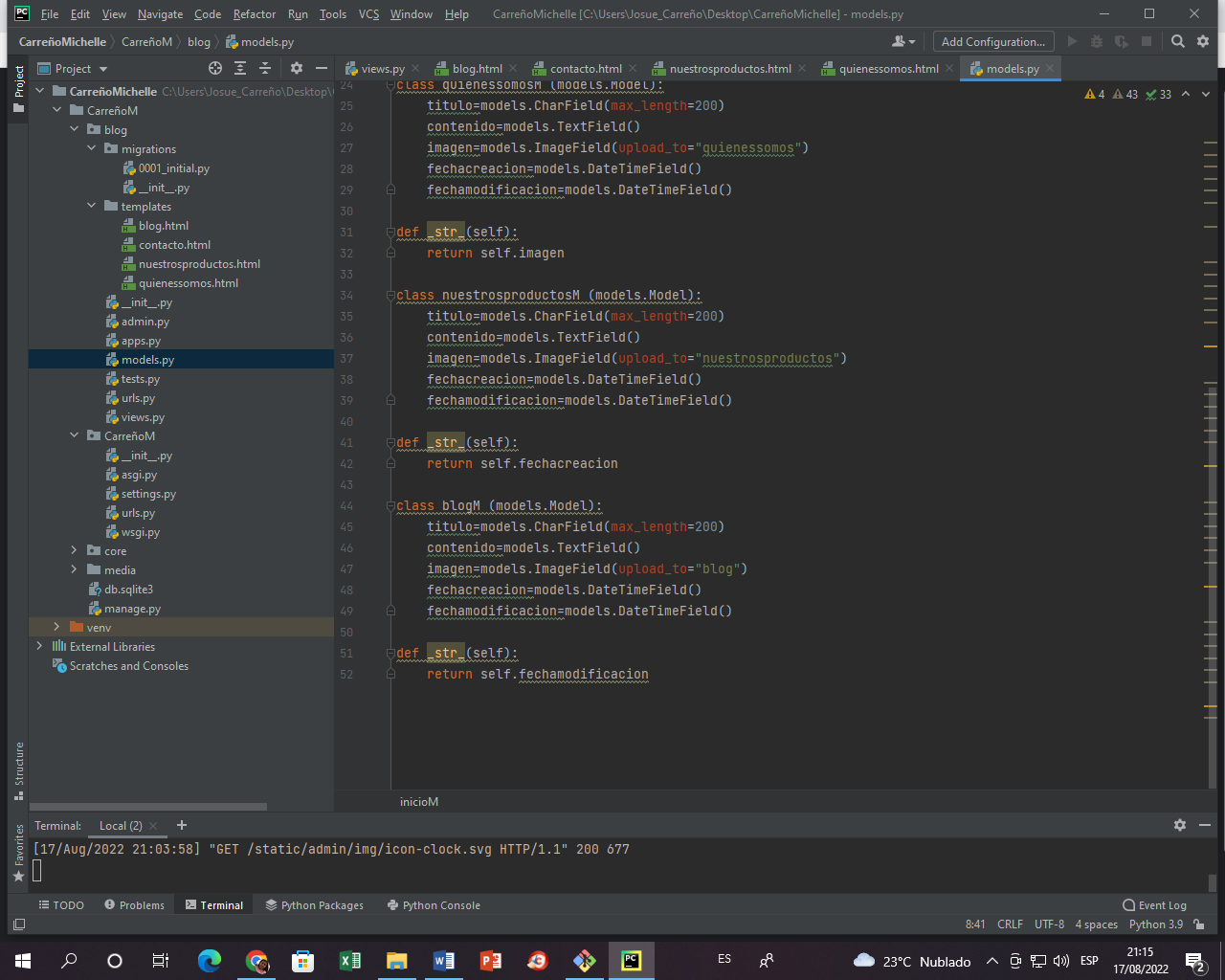












Estas imágenes son los datos guardados en el modelo hijo