PROGRAMACIÓN VISUAL

TALLER 1

GIT-GITHUB

NOMBRE:

ALLISON PINCAY REYES



CURSO:

3G

PROFESOR:

CARLOS PAZMIÑO PALMA

CONTENIDO:

1. ¿Qué es Django?
2. ¿Qué es la máquina virtual en Django
3. ¿Qué es MVT en Django. ?
4. Crear un proyecto con la máquina virtual.
5. Descargar los instaladores de Django al proyecto
6. Crear un proyecto para programar en Django
7. Ejecutar el proyecto y el mensaje de felicitaciones.
8. Crear una Apps core.
9. ¿Qué es la Carpeta Templates?
10. ¿Qué es la Carpeta stactic?
11. Crear un archivo base html en la APPS core
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
15. Crear la urls que llame al views.
16. Integrar la aplicación APPS core al proyecto principal
17. Crear las tablas del sistema de usuarios para utilizar el panel de administración.
18. Crear un usuario para poder ingresar al Panel de Administración
19. Que es un modelo en Django
20. Crear un modelo 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.
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.
26. ¿Qué es Django?

es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Cuando construyes un sitio web, siempre necesitas un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

Los frameworks sirven para que no tengamos que reinventar la rueda cada vez y que podamos avanzar más rápido al construir un nuevo sitio.



1. ¿Qué es la máquina virtual en Django?

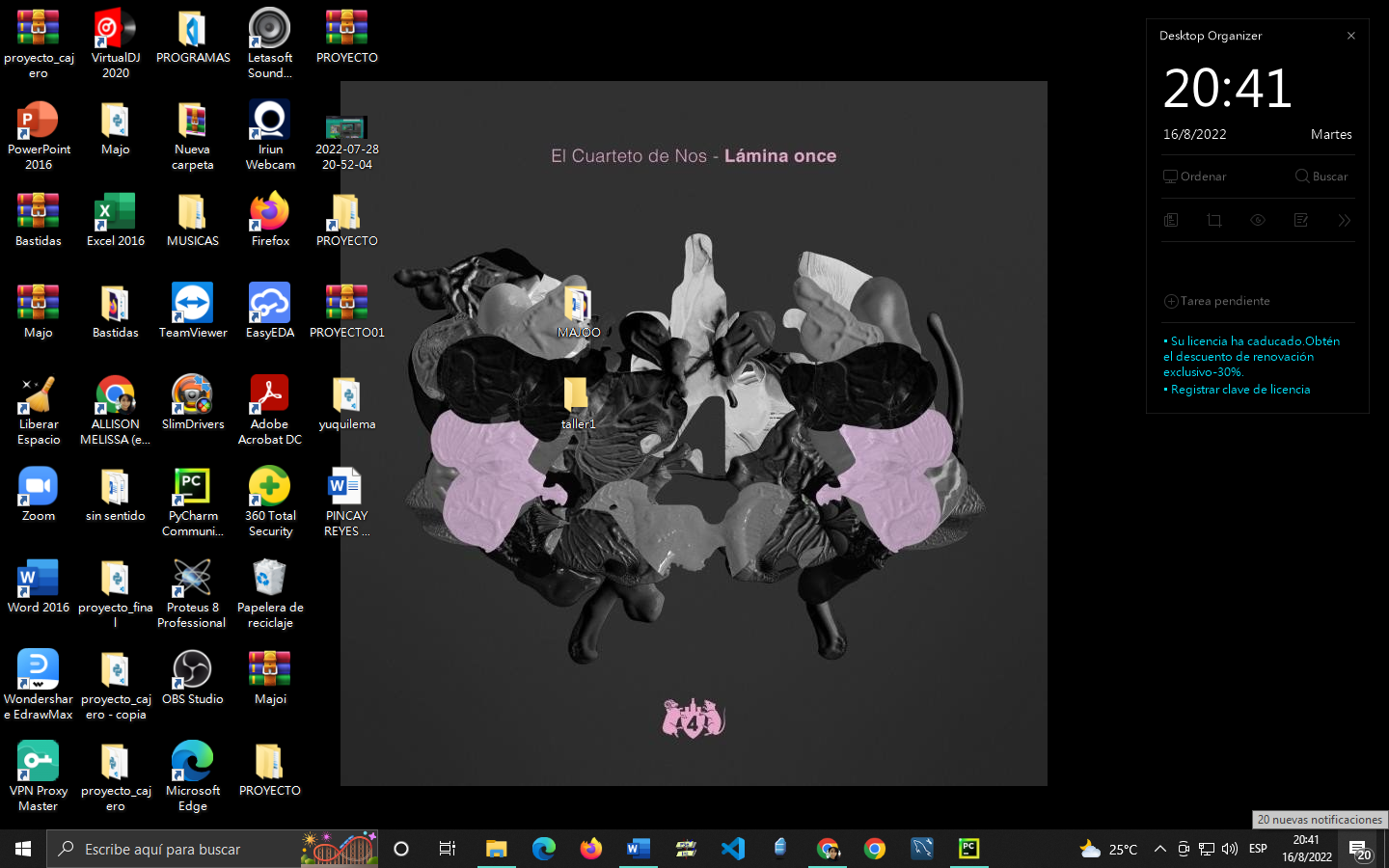
Es una instalación de Django en tu computadora local que puedes usar para desarrollar y probar apps Django antes de desplegarlas al entorno de producción.

1. ¿Qué es MVT en Django. ?

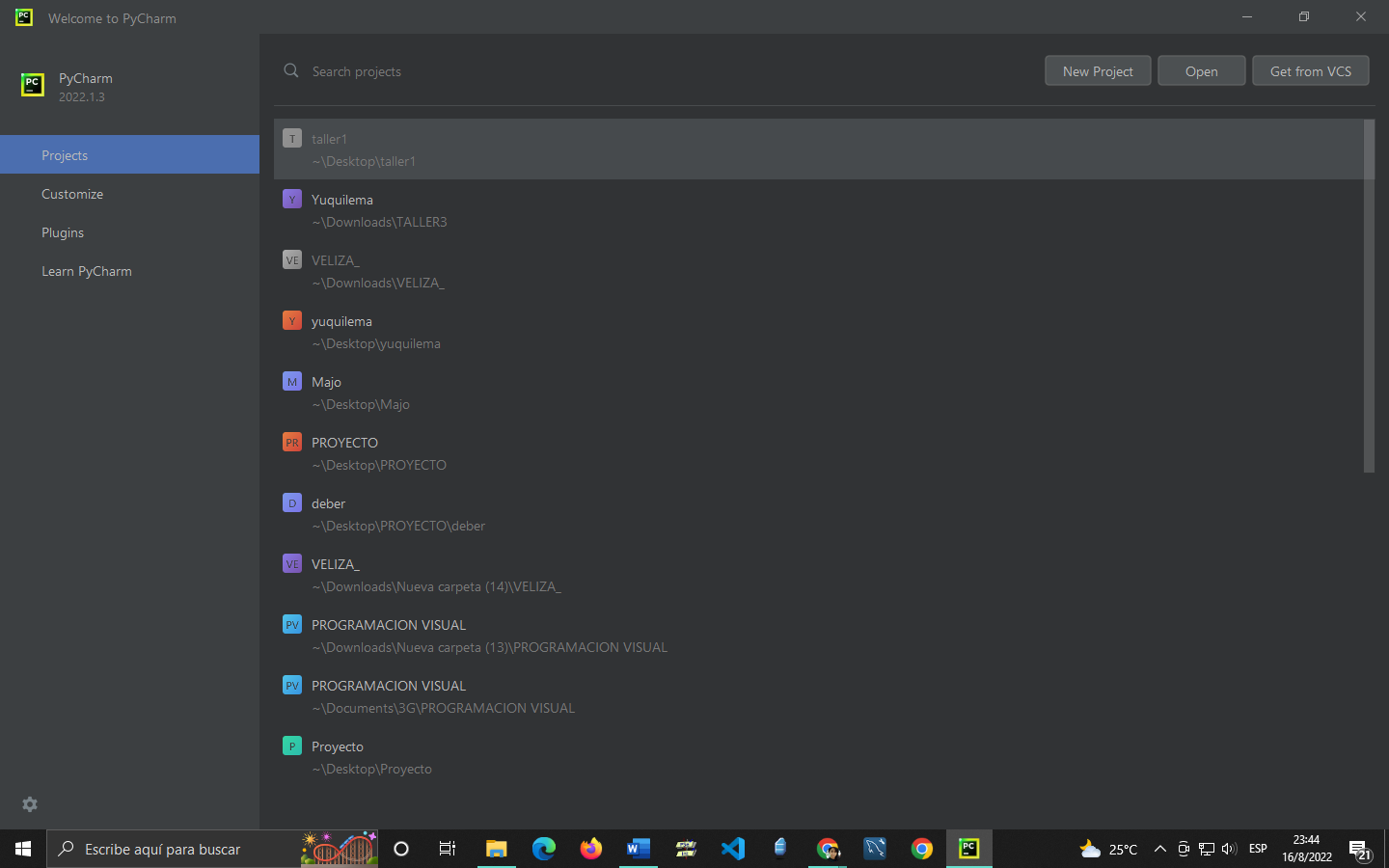
Django redefine este modelo como MVT: Modelo-Vista-Template.

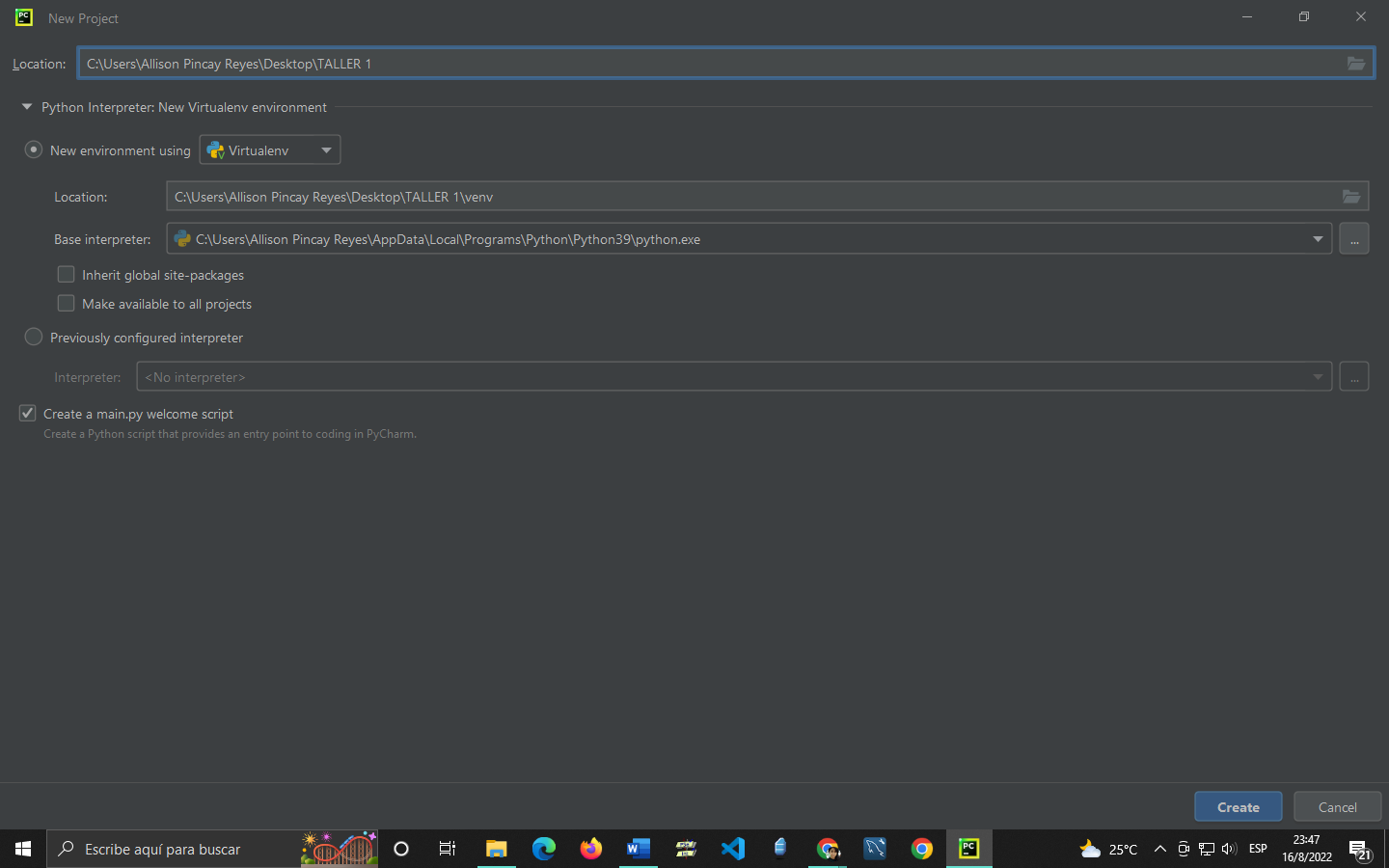
1. Crear un proyecto con la máquina virtual.

Este va a ser mi proyecto en la maquina virtual.

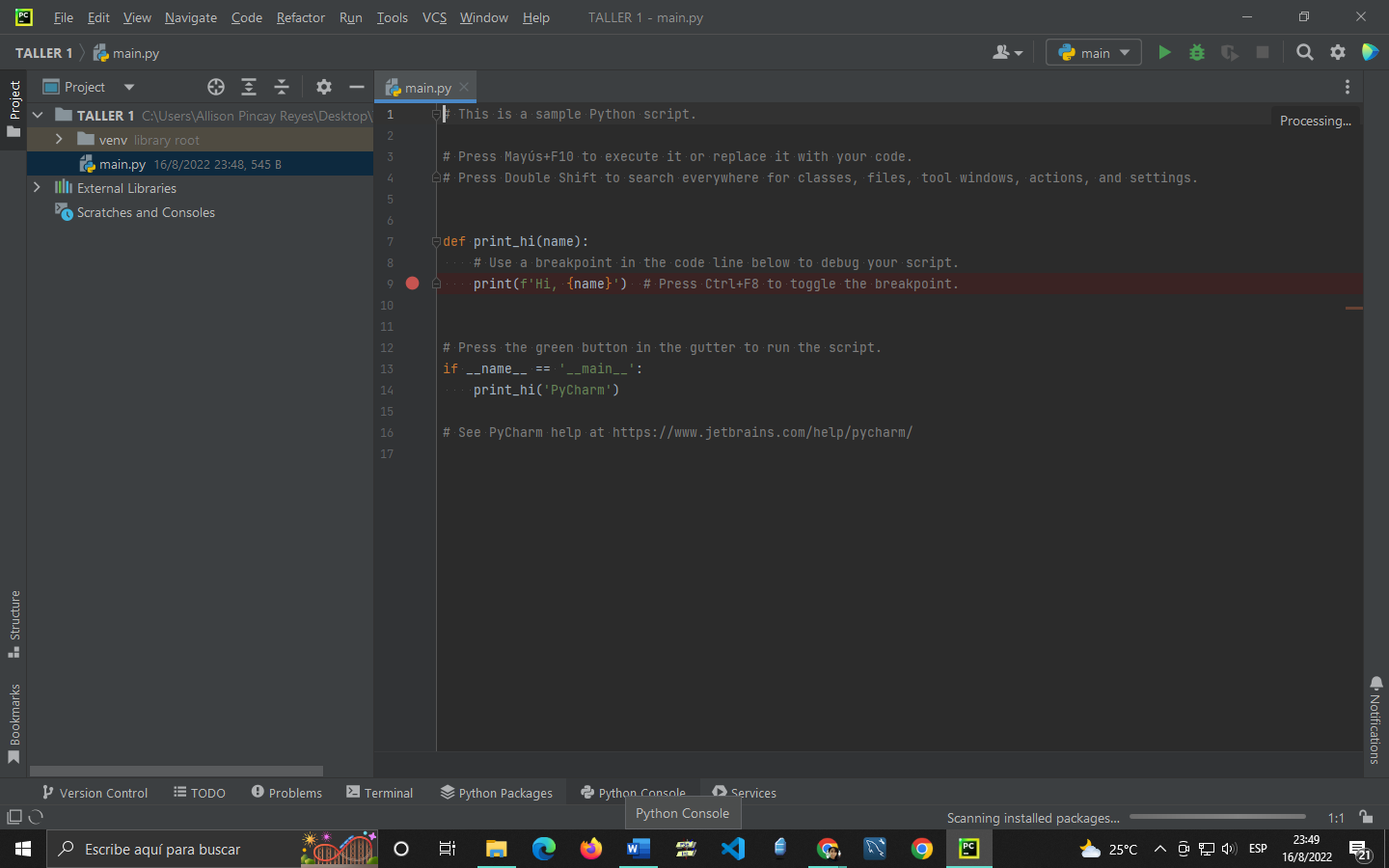


En pycharm nos vamos a la opcion New Proyect y buscamos la ubicación de nuestro archivo “taller 1”.



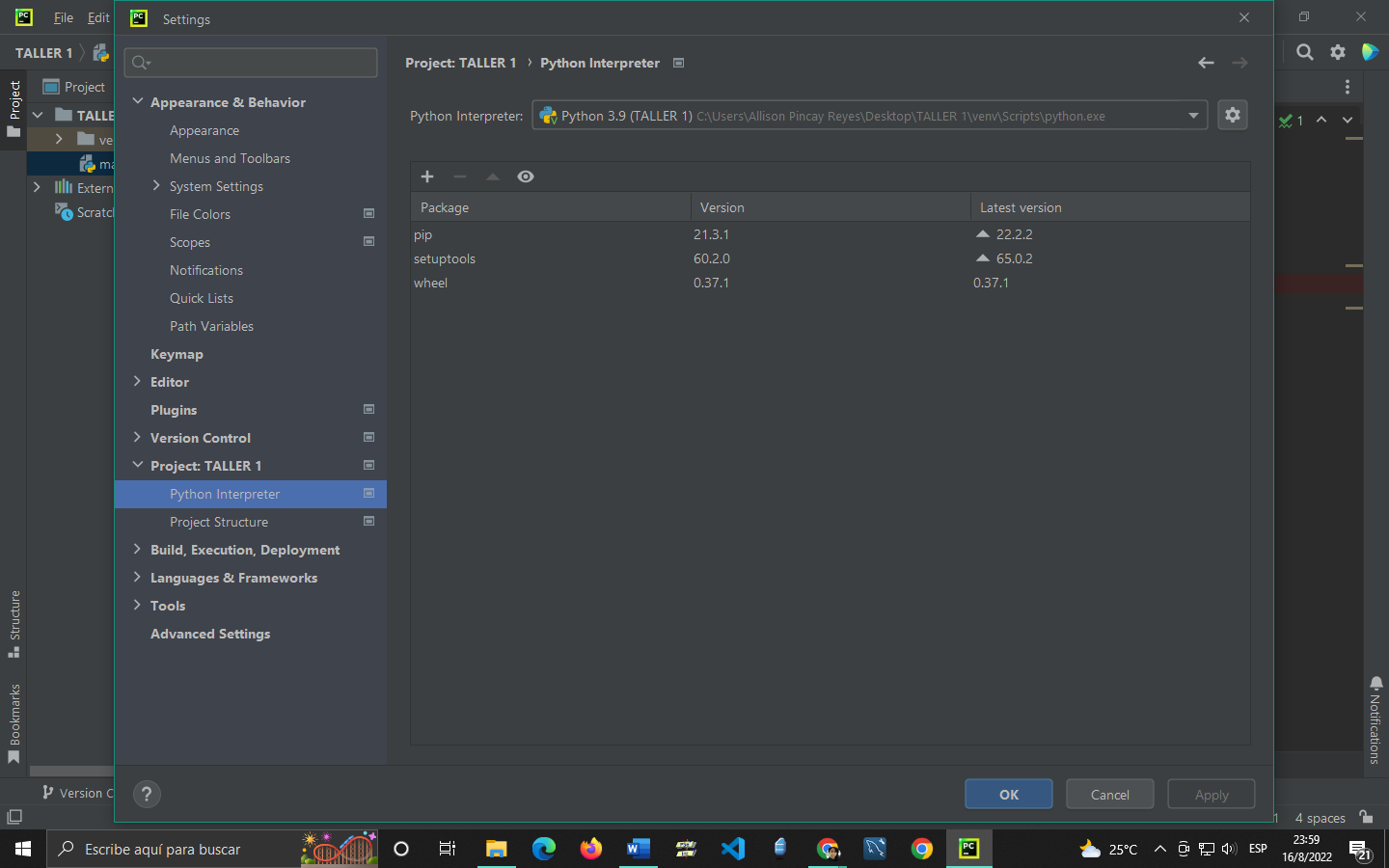


Y esta listo

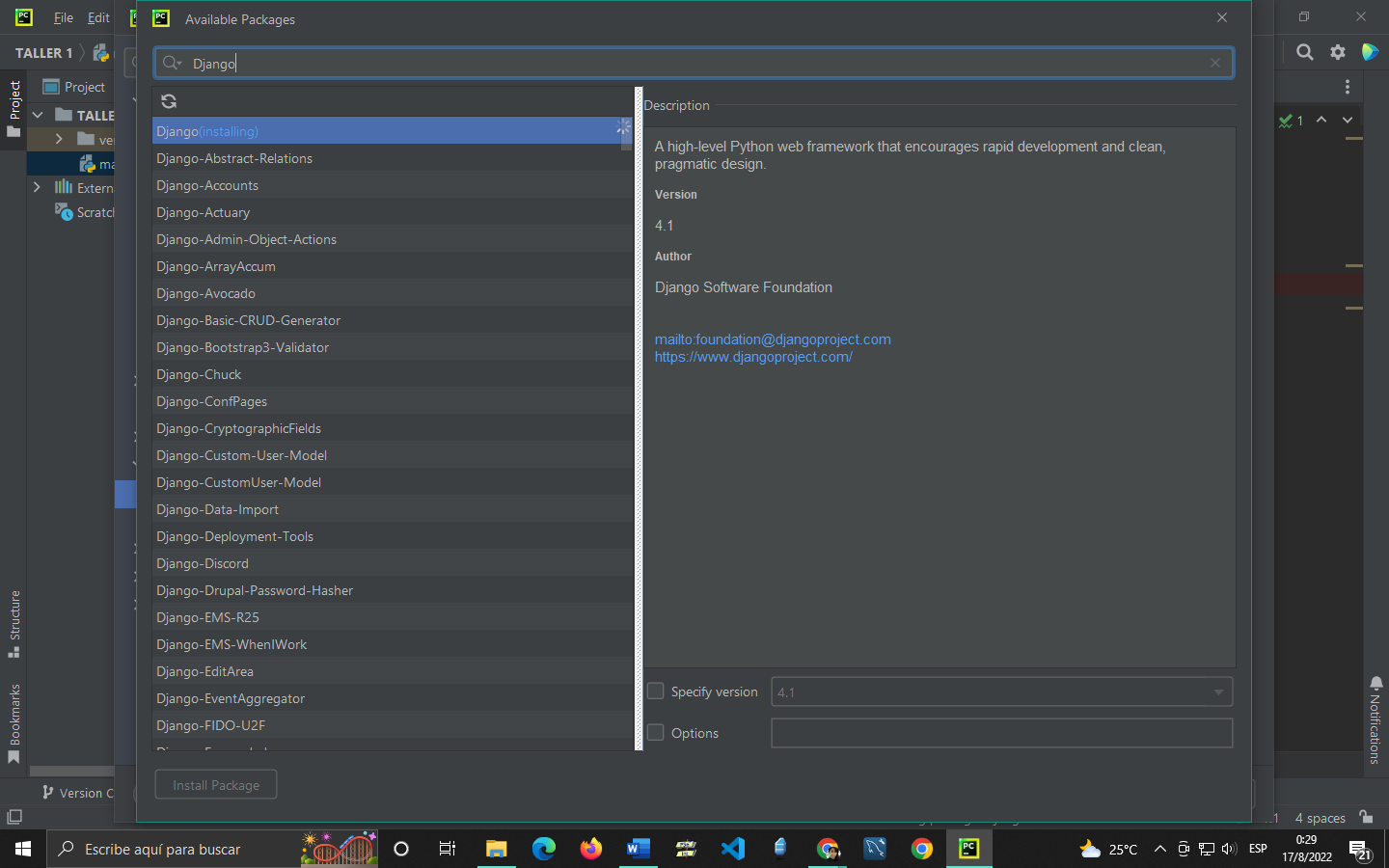


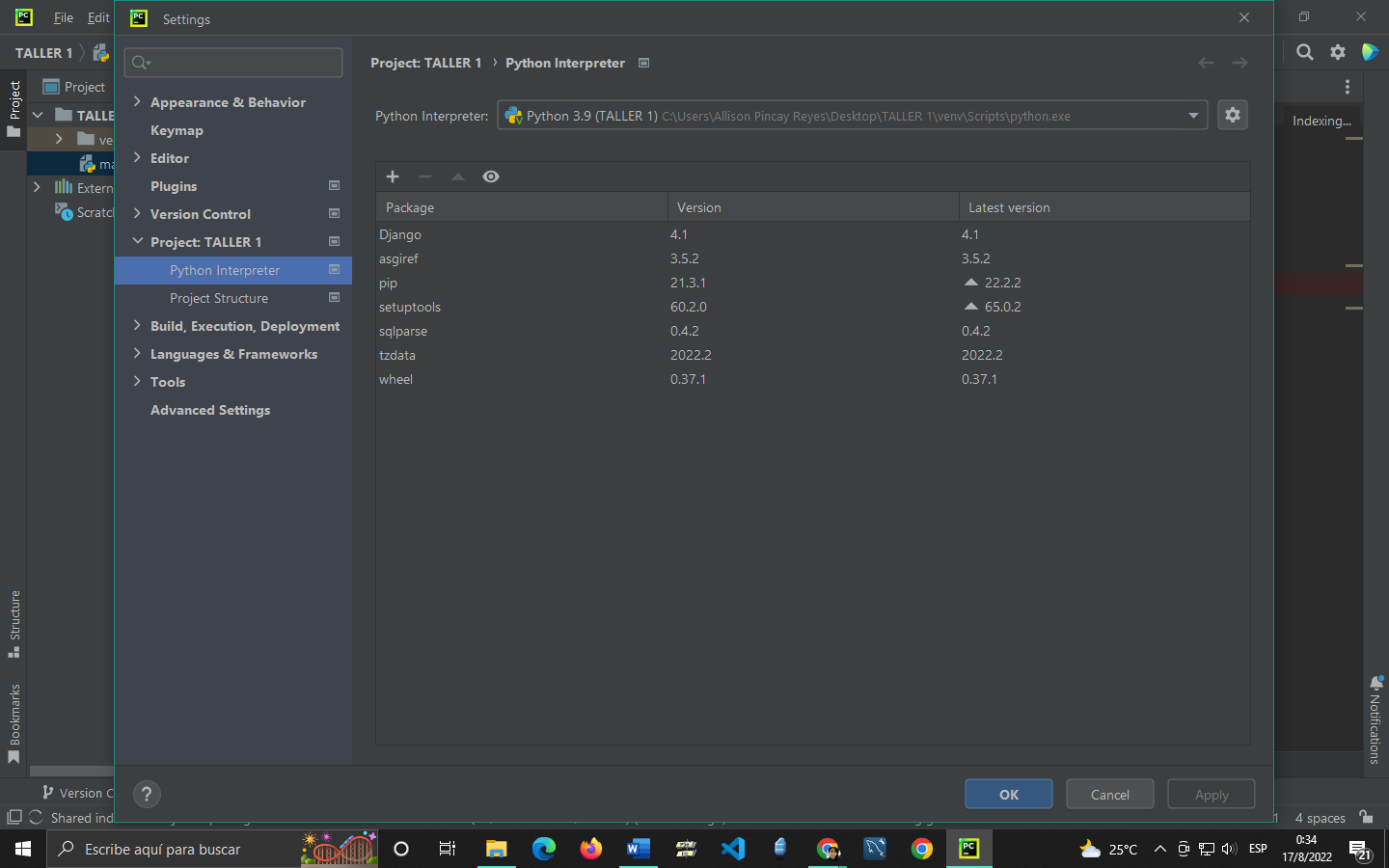
1. Descargar los instaladores de Django al proyecto

Dentro de lo que es Settings, en la parte que dice “Python Interpreter” nos vamos al simbolo “+”



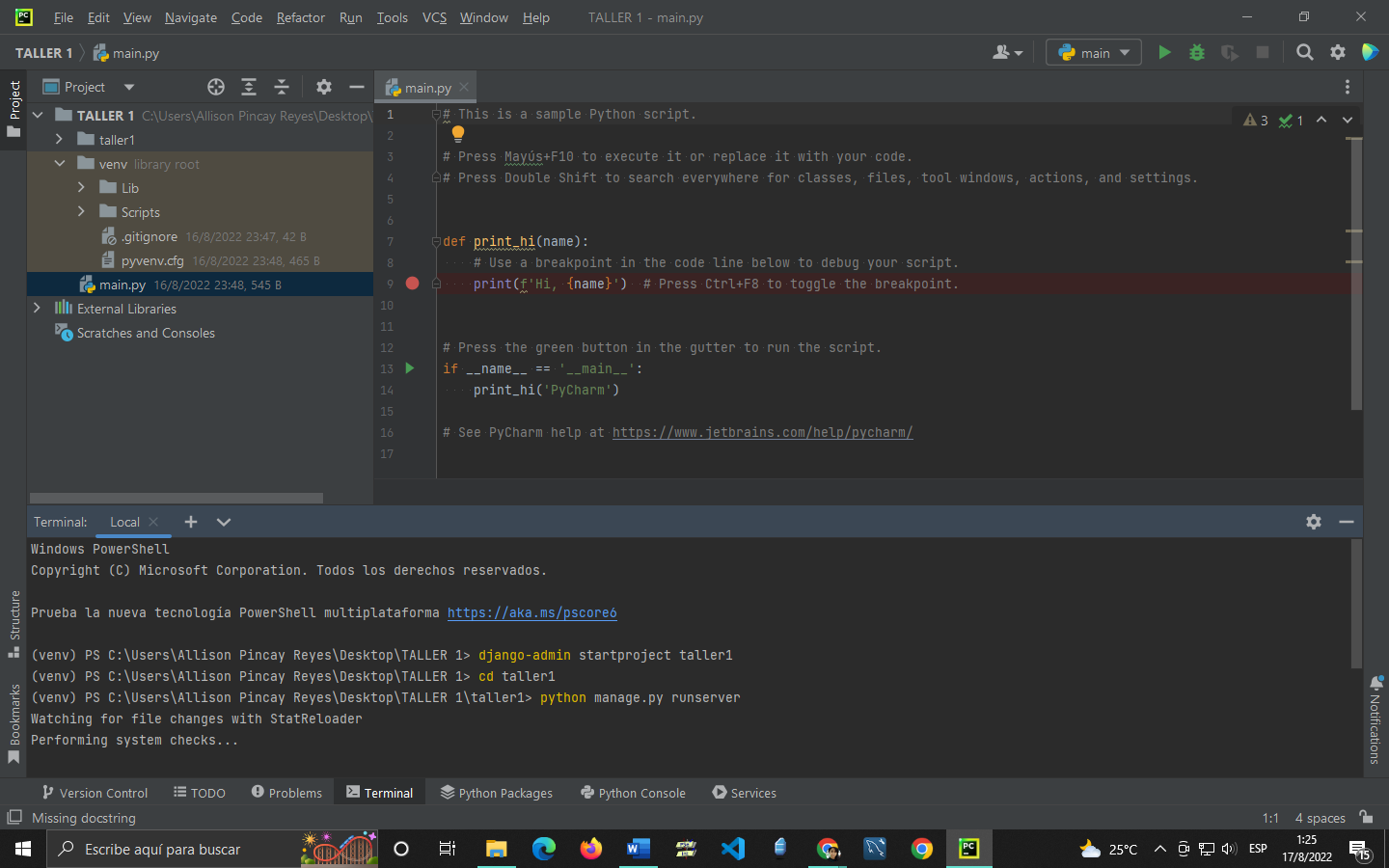
Buscamos Django y procedemos a instalarlo



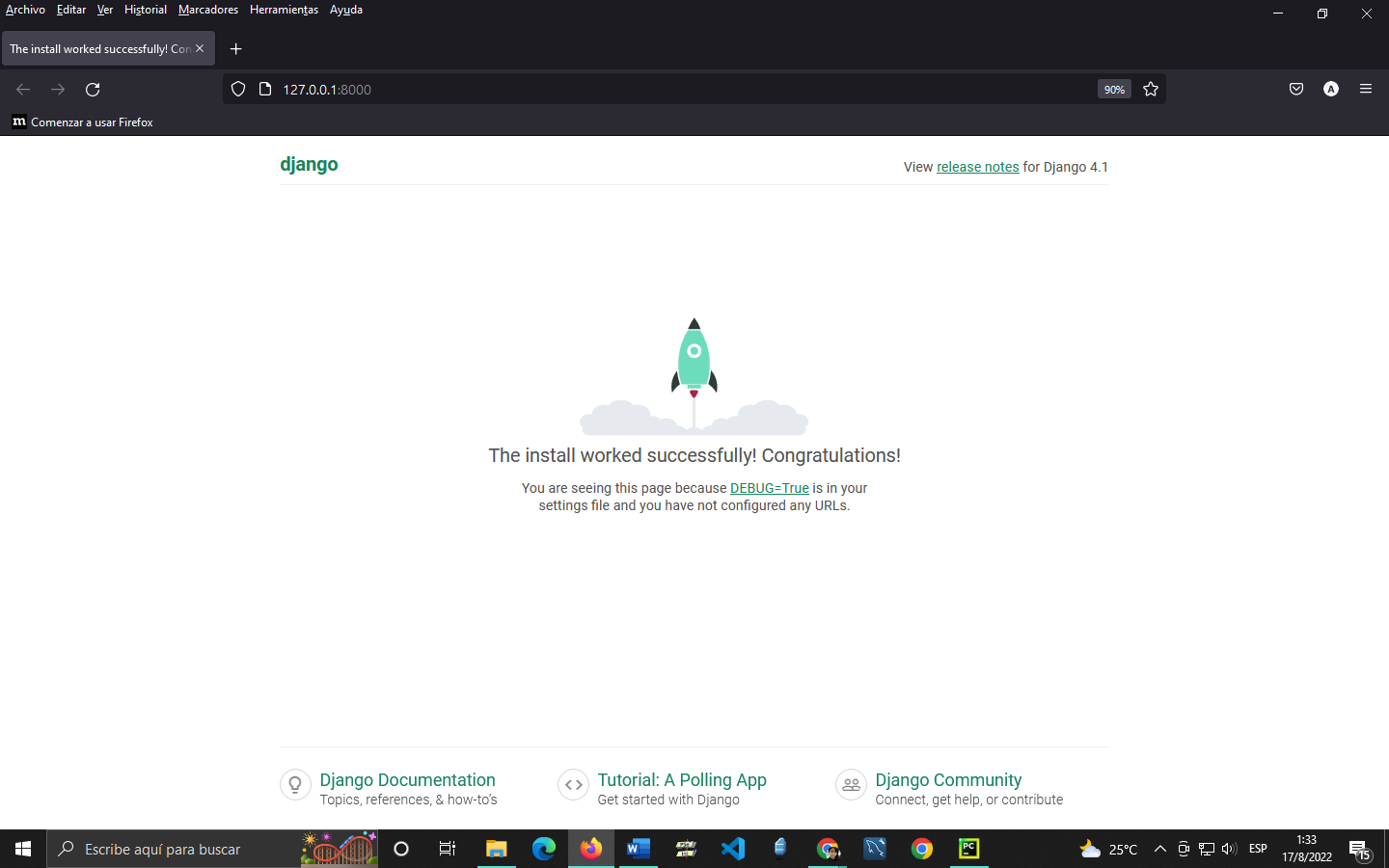


1. Crear un proyecto para programar en Django

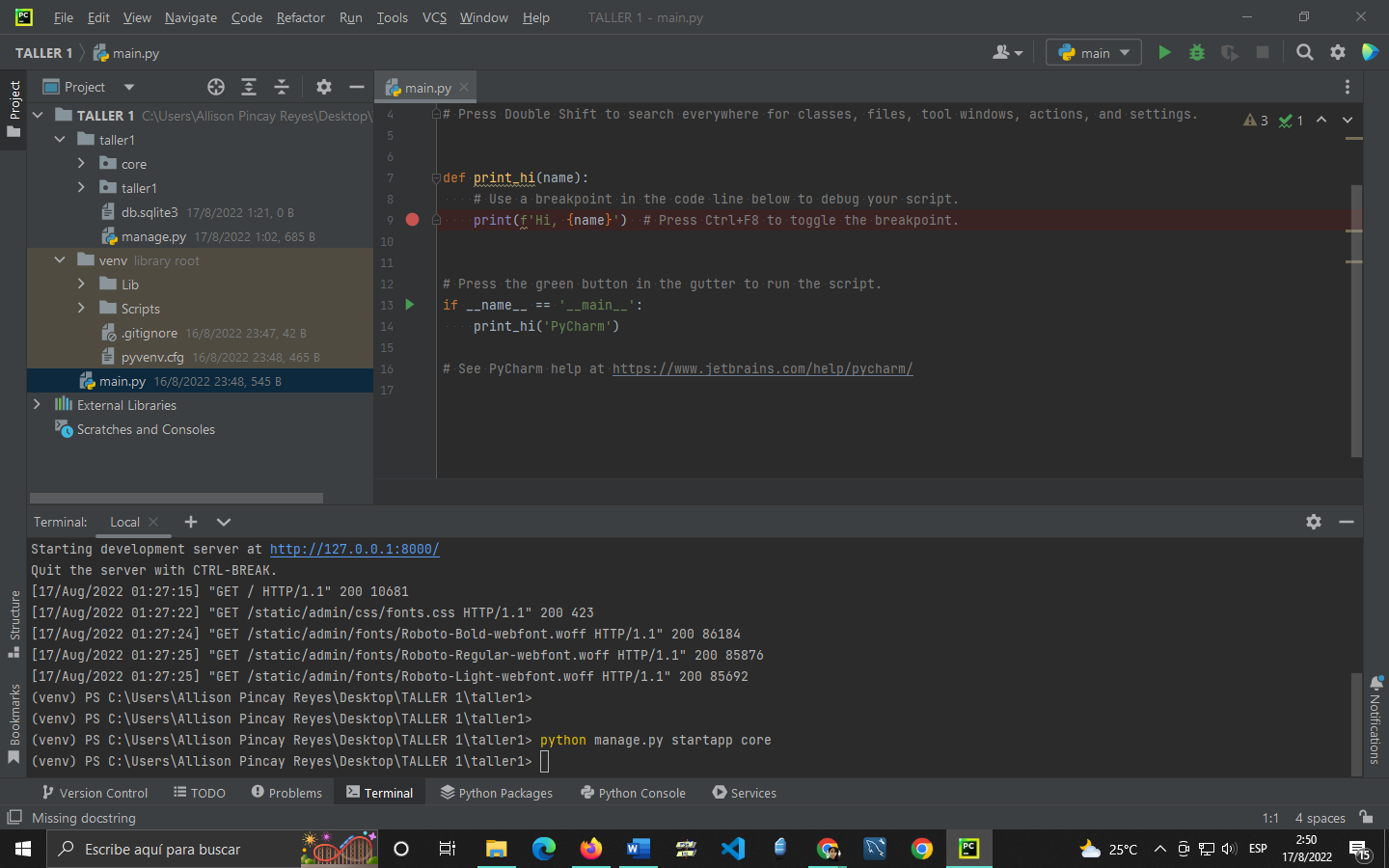
Mi proyecto va a tener el nombre “taller1”



1. Ejecutar el proyecto y el mensaje de felicitaciones.



1. Crear una Apps core.



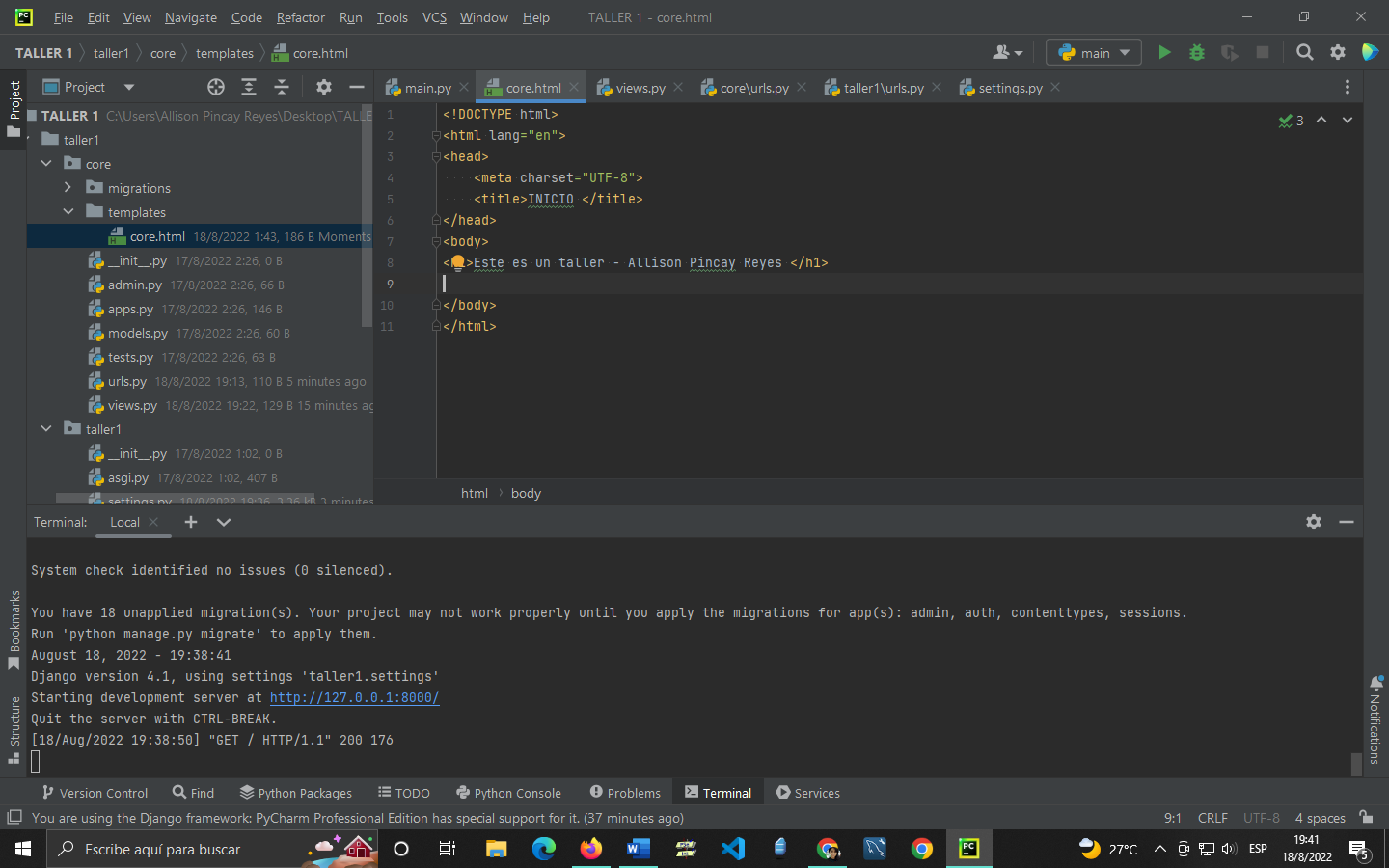
1. ¿Qué es la Carpeta Templates?

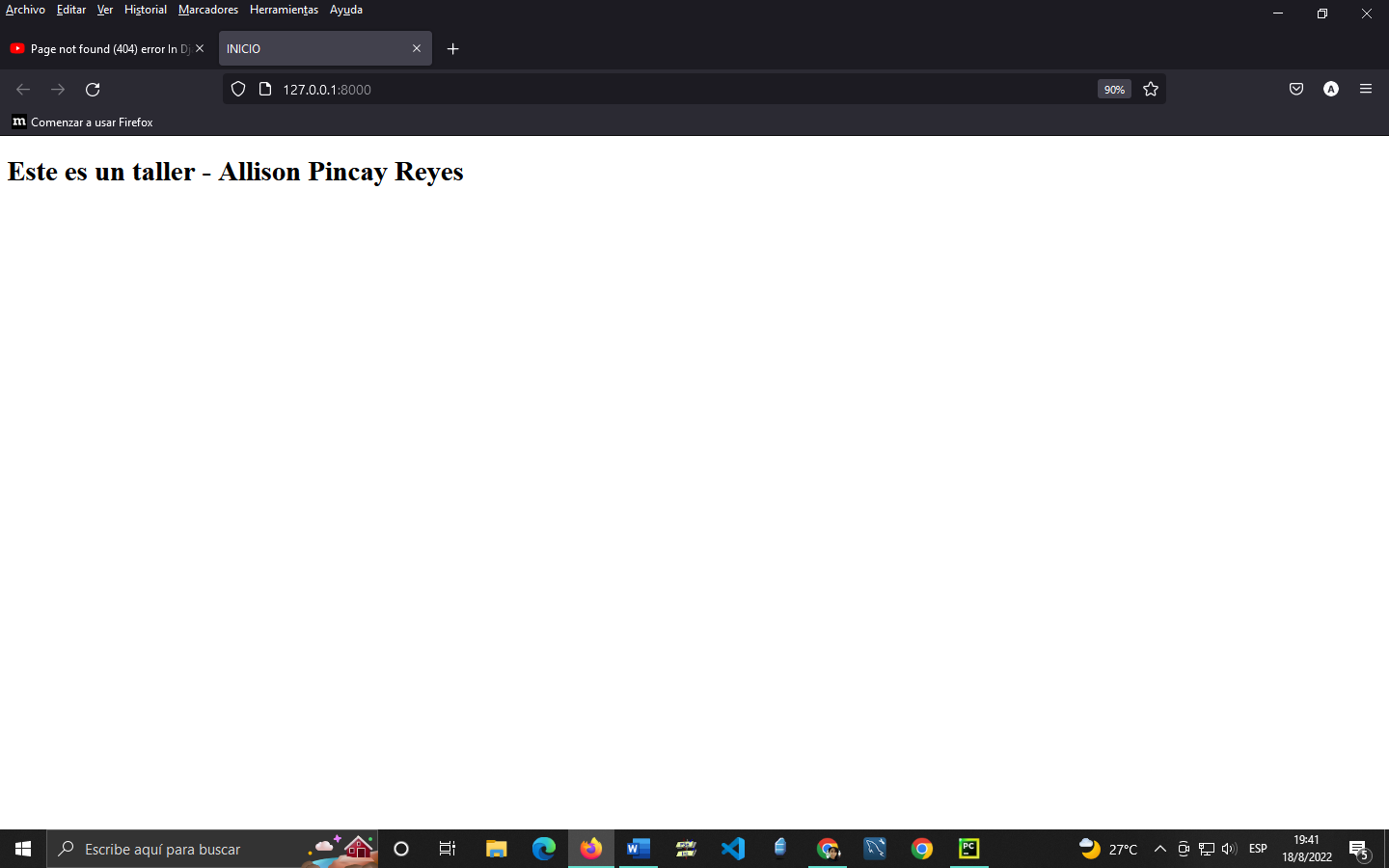
Al ser un marco web, Django necesita una forma conveniente de generar HTML dinámicamente. El enfoque más común se basa en plantillas. Una plantilla contiene las partes estáticas de la salida HTML deseada, así como alguna sintaxis especial que describe cómo se insertará el contenido dinámico.

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.scss.tpl**.

1. Crear un archivo base html en la APPS core





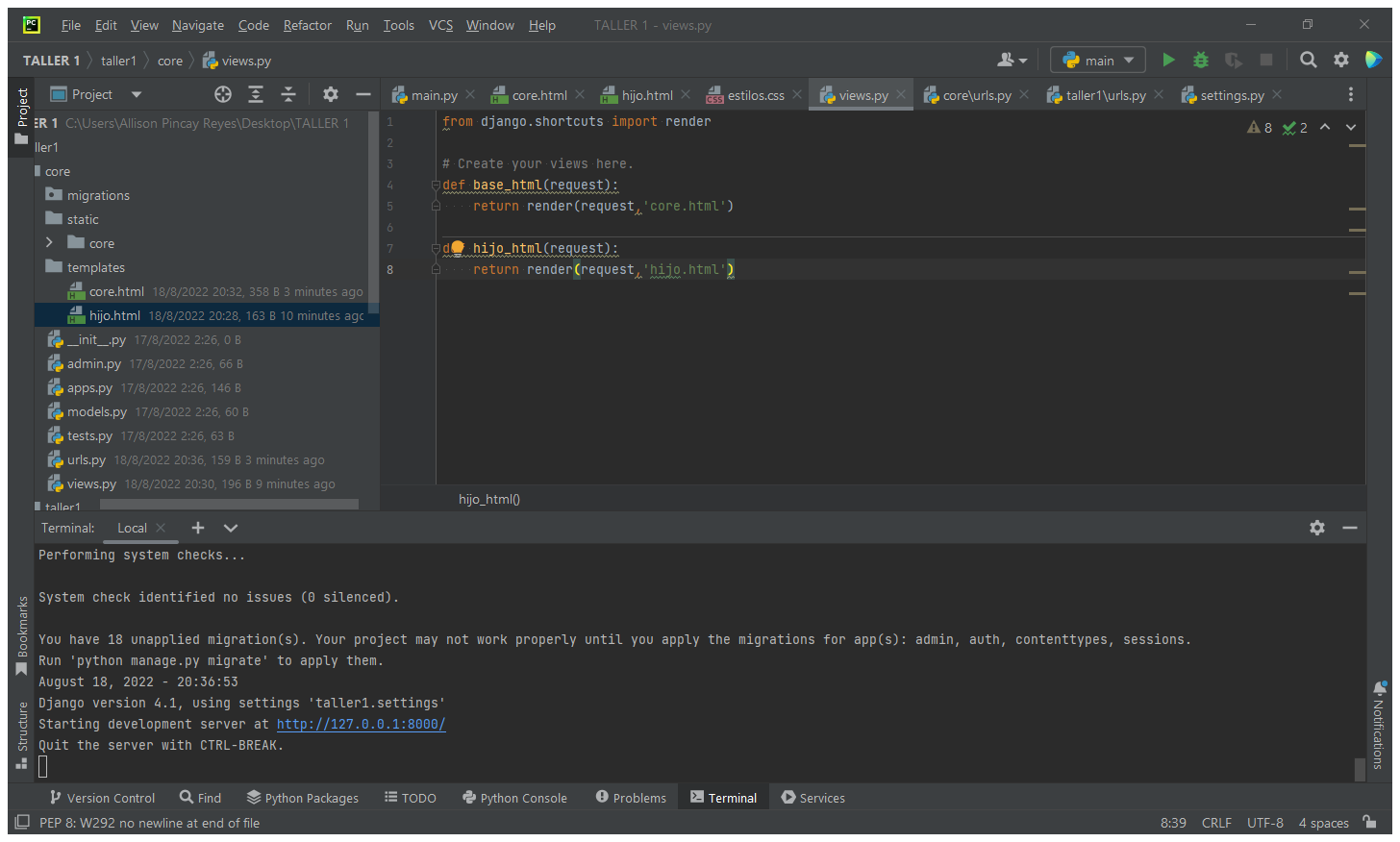
1. Como se llaman a los CSS desde el archivo base html.

<link rel="STYLESHEET" href="{%static 'core/css/estilos.css' %}">

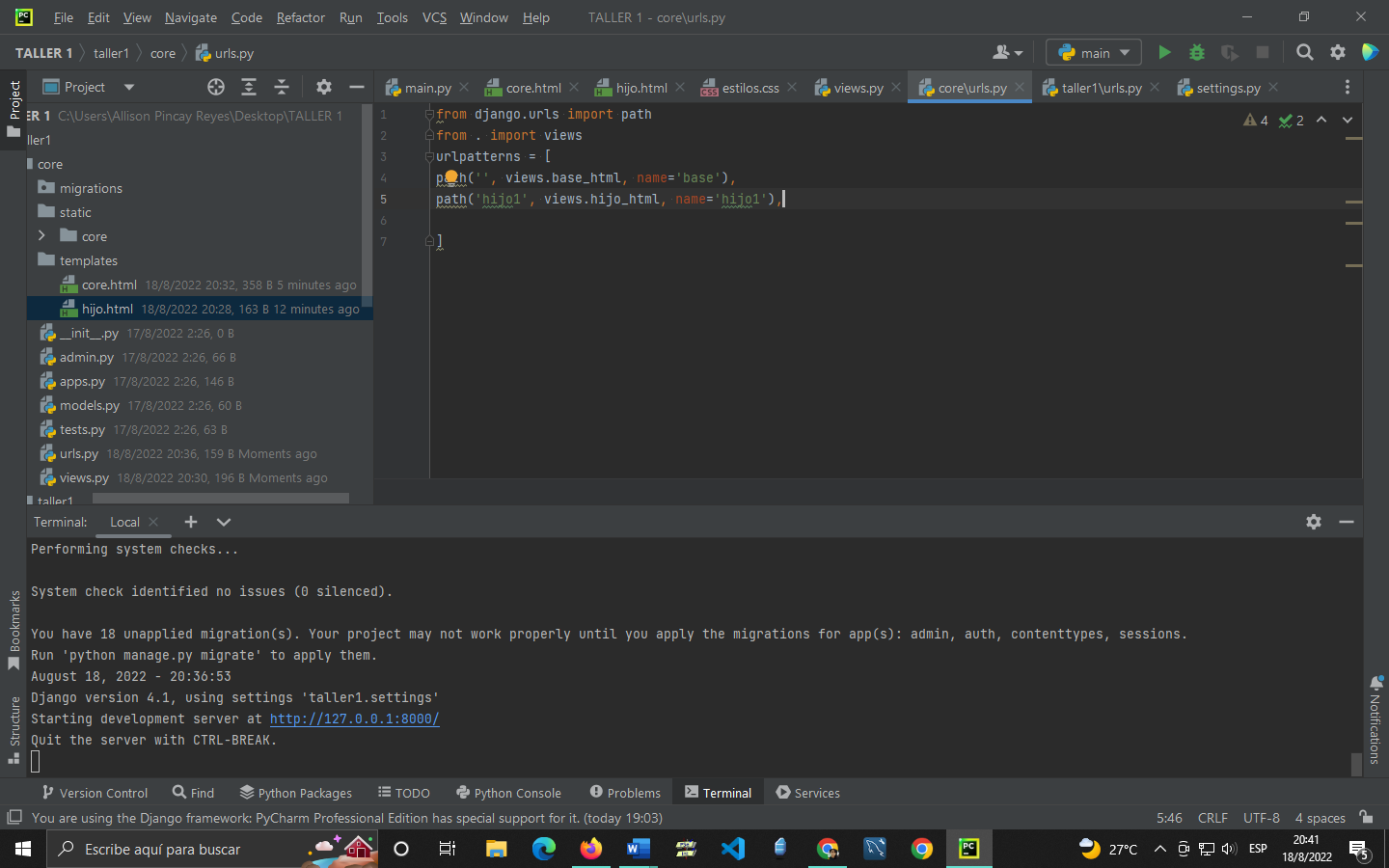
1. Como consume un archivo hijo html al utilizar la herencia del archivo base html.

{% extends 'core.html' %}  
{% load static %}  
{% block titulo %}  
 Hijo  
{% endblock %}  
{% block content %}  
<h1>Mensaje del archivo hijo</h1>  
  
{% endblock %}

1. Crear un view que llame al html hijo



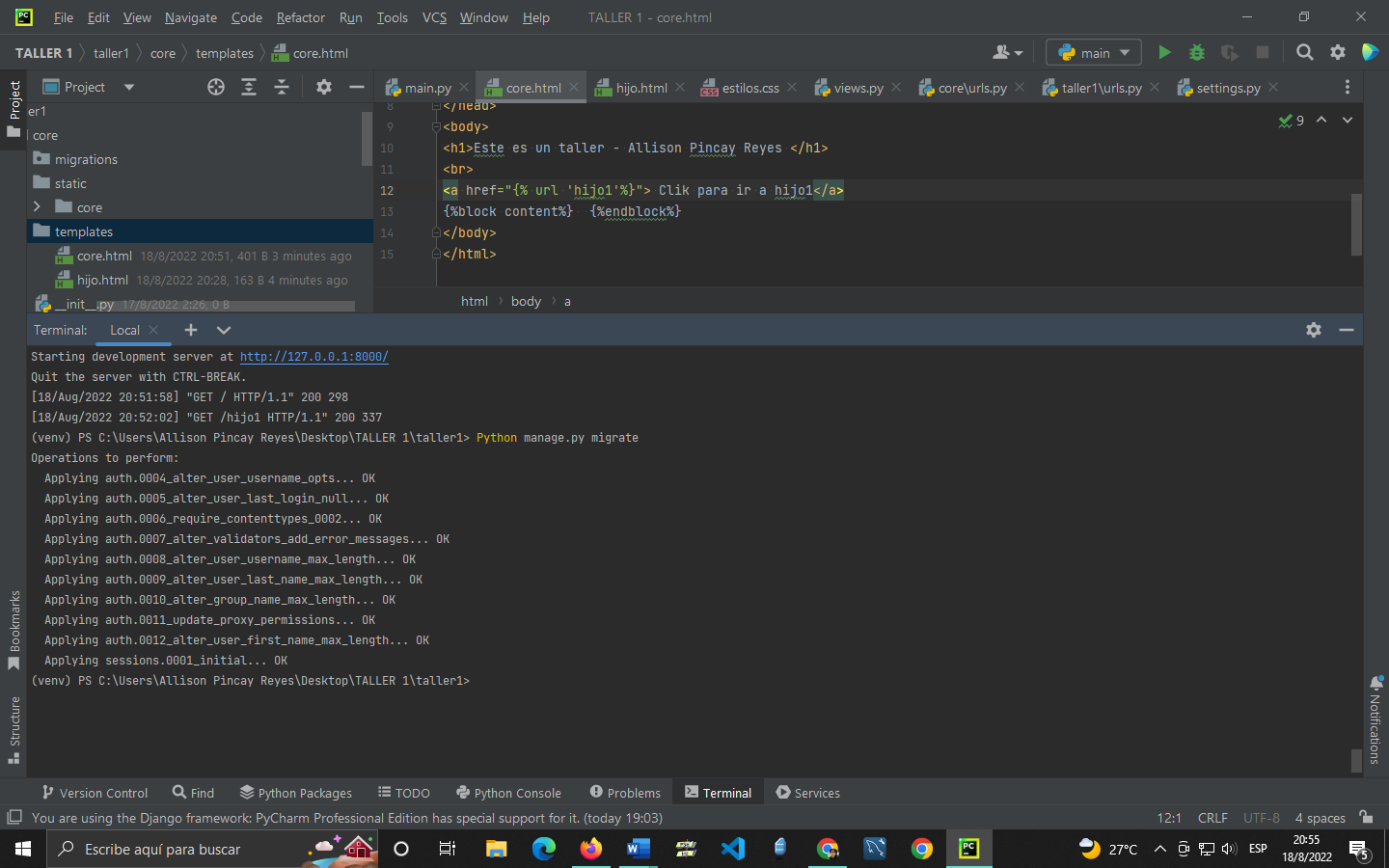
1. Crear la urls que llame al views.



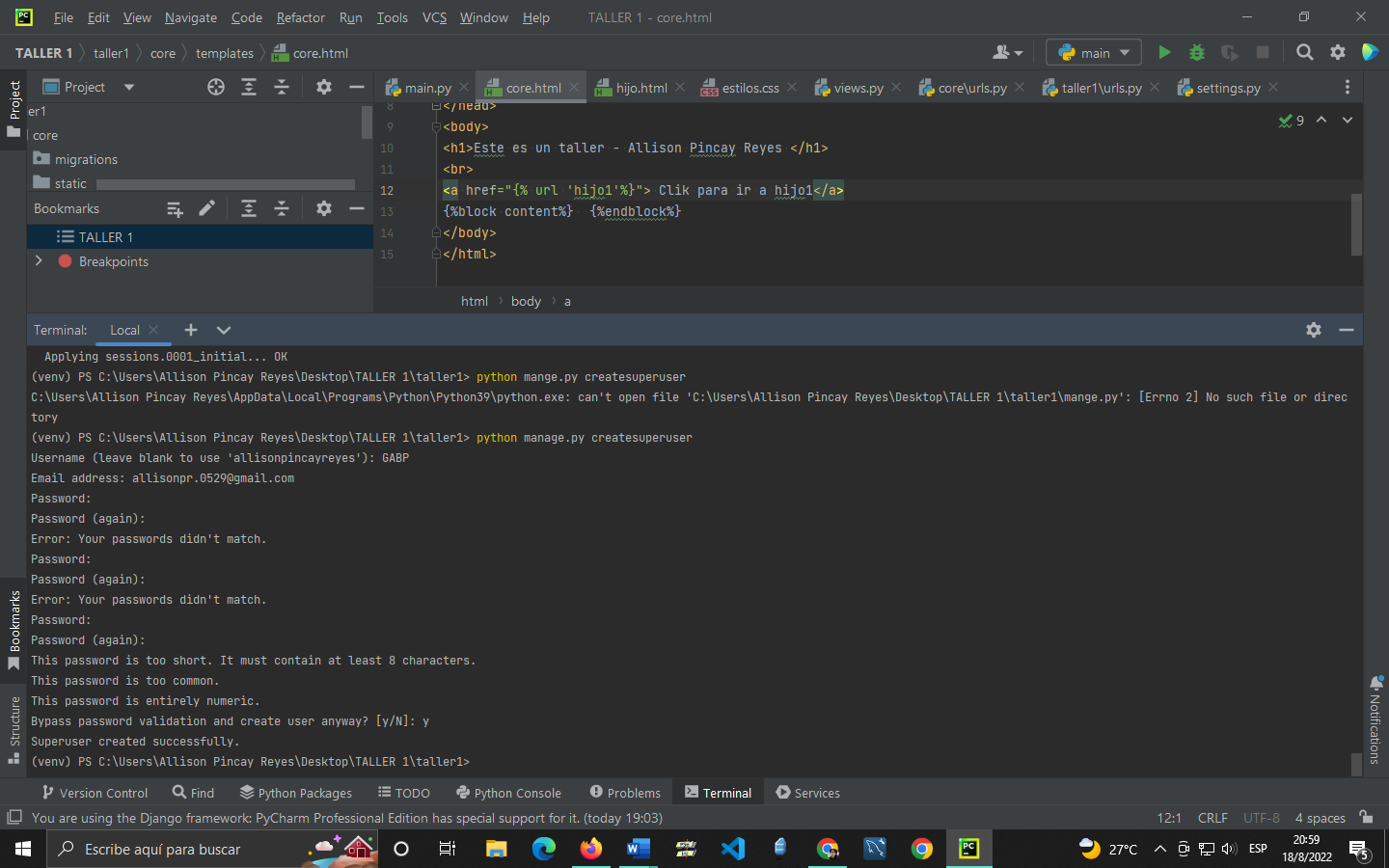
1. Integrar la aplicación APPS core al proyecto principal

from django.contrib import admin  
from django.urls import path,include  
  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', include('core.urls')),  
]

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



1. Crear un usuario para poder ingresar al Panel de Administración



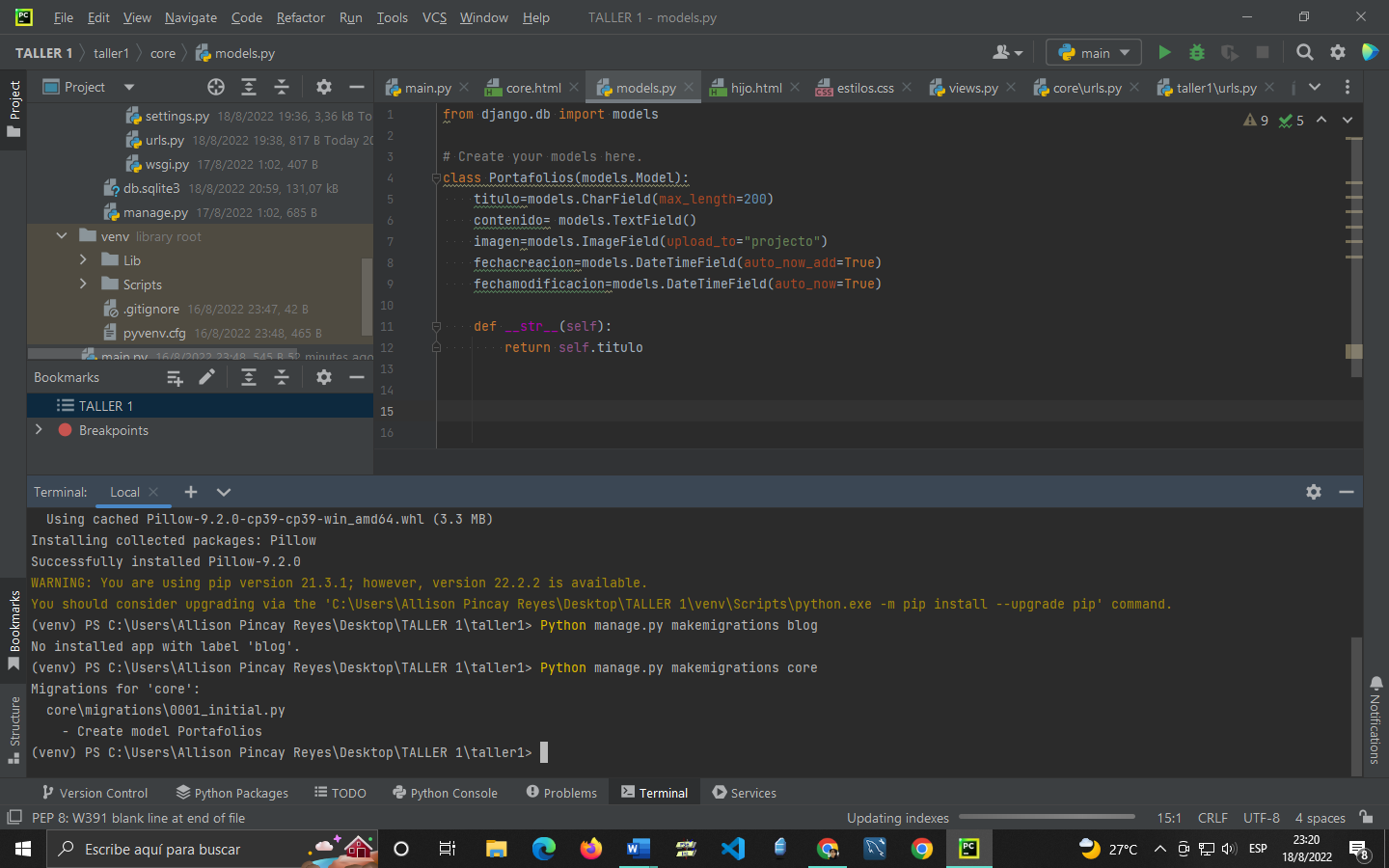
1. Que es un modelo en Django

Un modelo en Django es un tipo especial de objeto que se guarda en la base de datos. Una base de datos es una colección de datos.

1. Crear un modelo en Django.

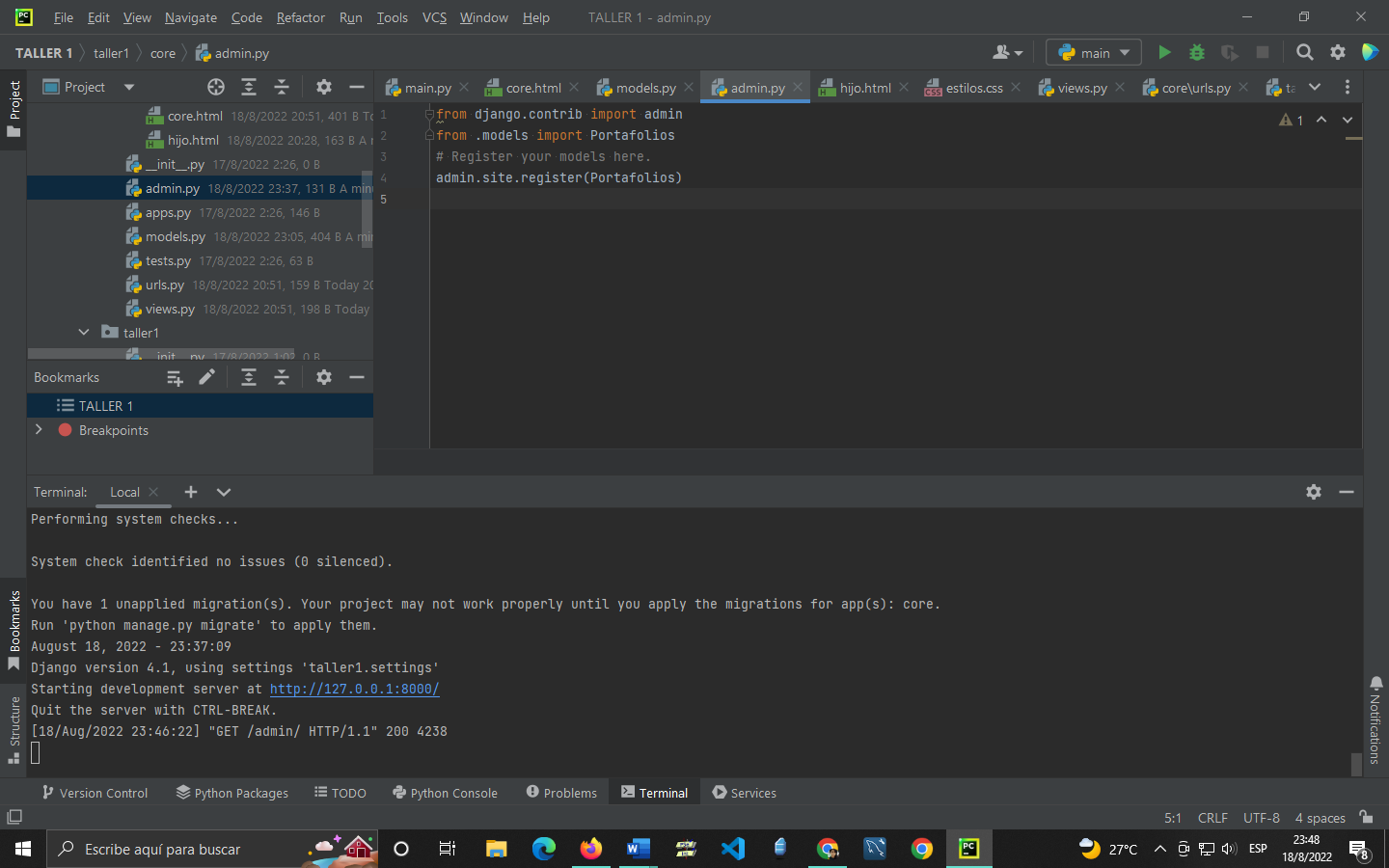
from django.db import models  
  
# Create your models here.  
class Portafolios(models.Model):  
 titulo=models.CharField(max\_length=200)  
 contenido= models.TextField()  
 imagen=models.ImageField(upload\_to="projecto")  
 fechacreacion=models.DateTimeField(auto\_now\_add=True)  
 fechamodificacion=models.DateTimeField(auto\_now=True)  
  
 def \_\_str\_\_(self):  
 return self.titulo

1. Migrar el Modelo a la base del Panel de Administración.

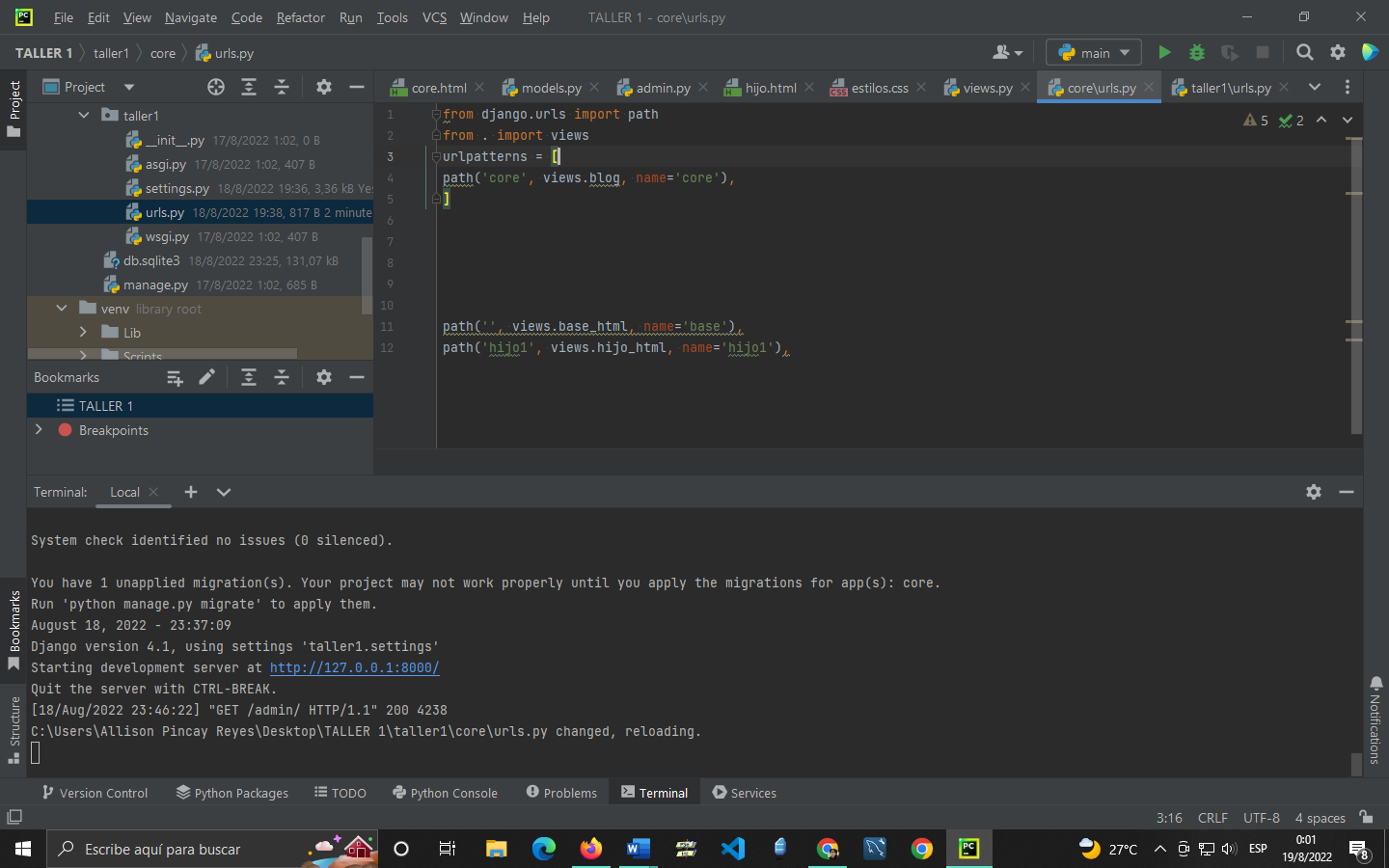


1. Integrar el Modelo al Panel de Administración.

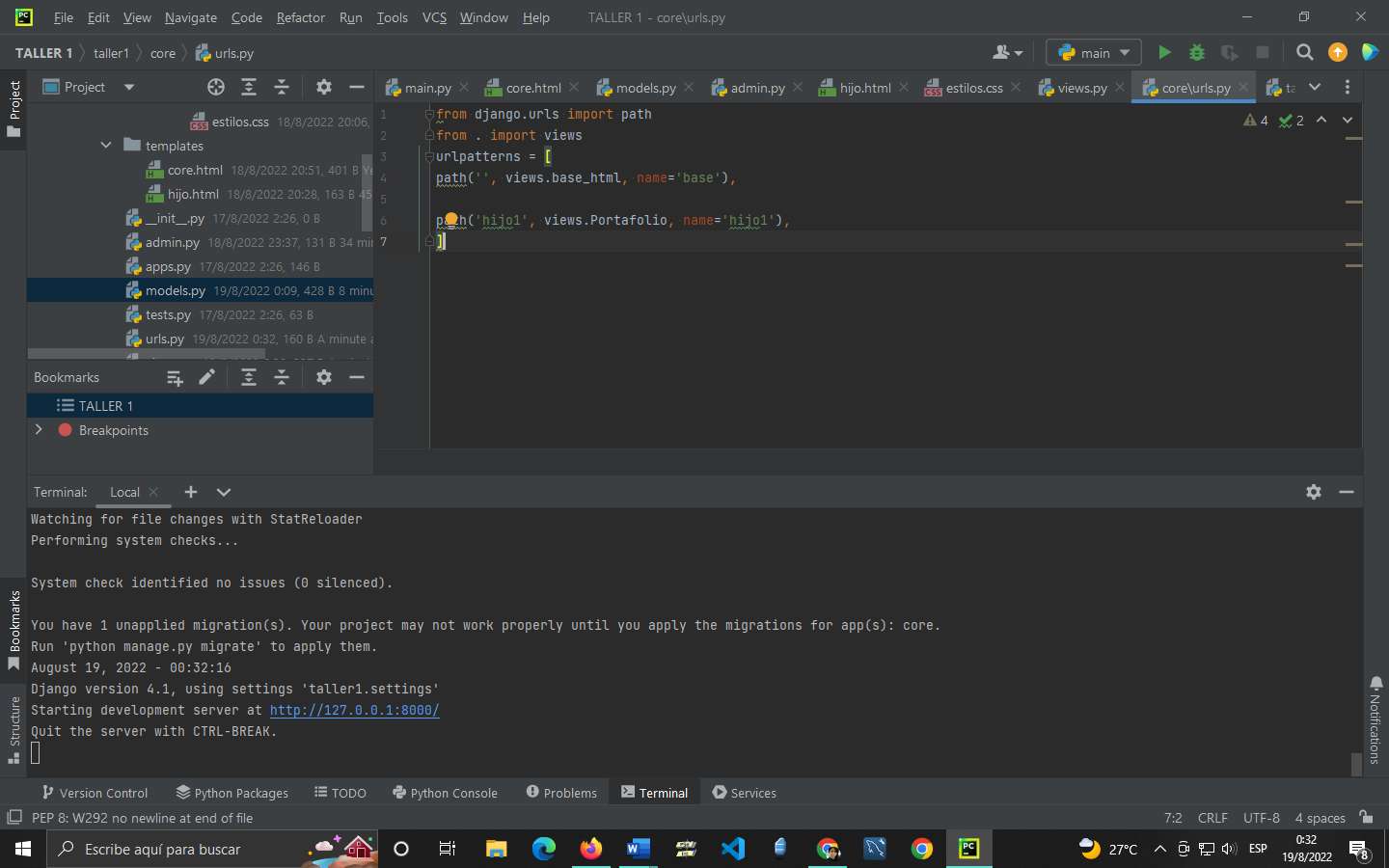
ADMIN



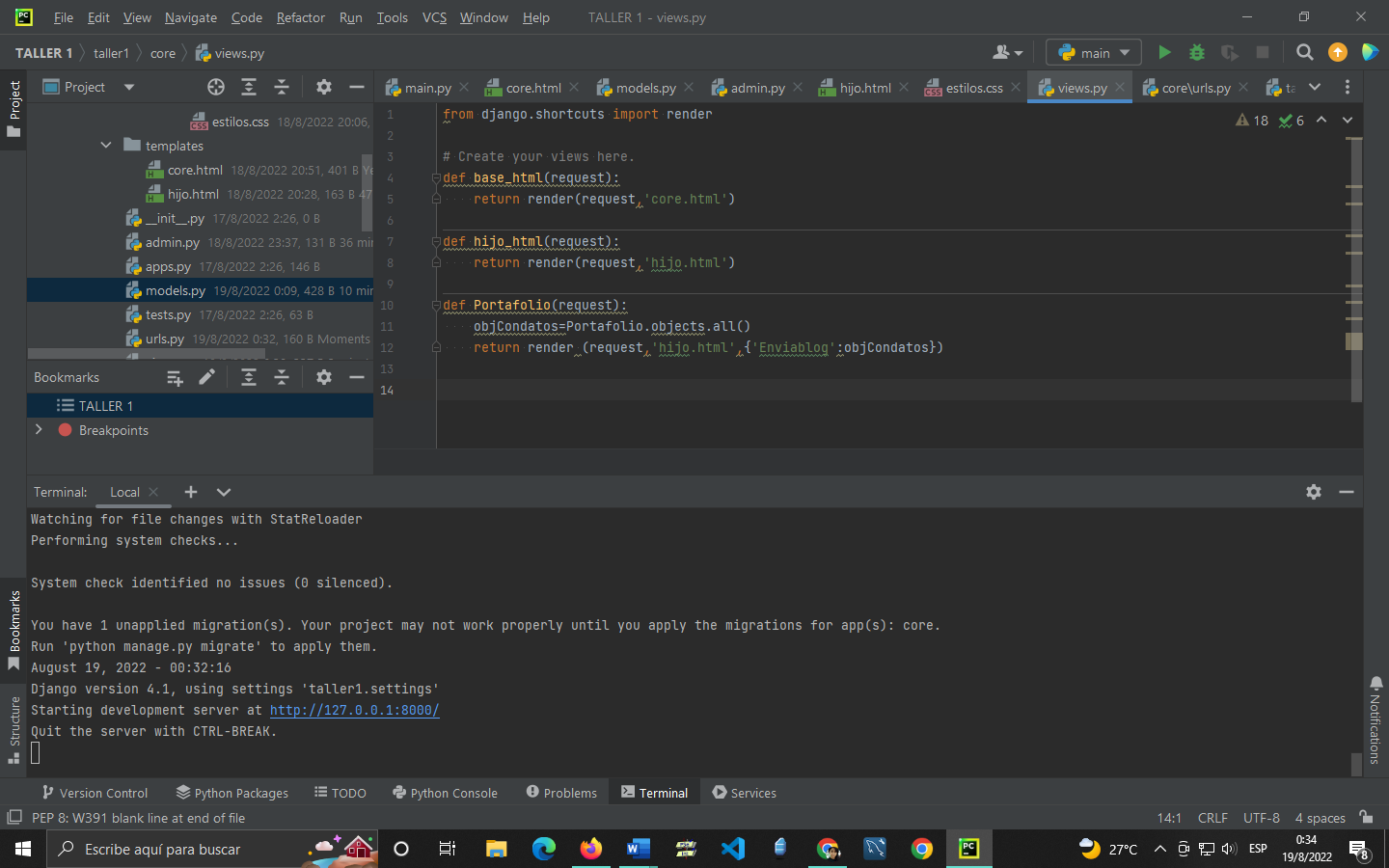
URLS



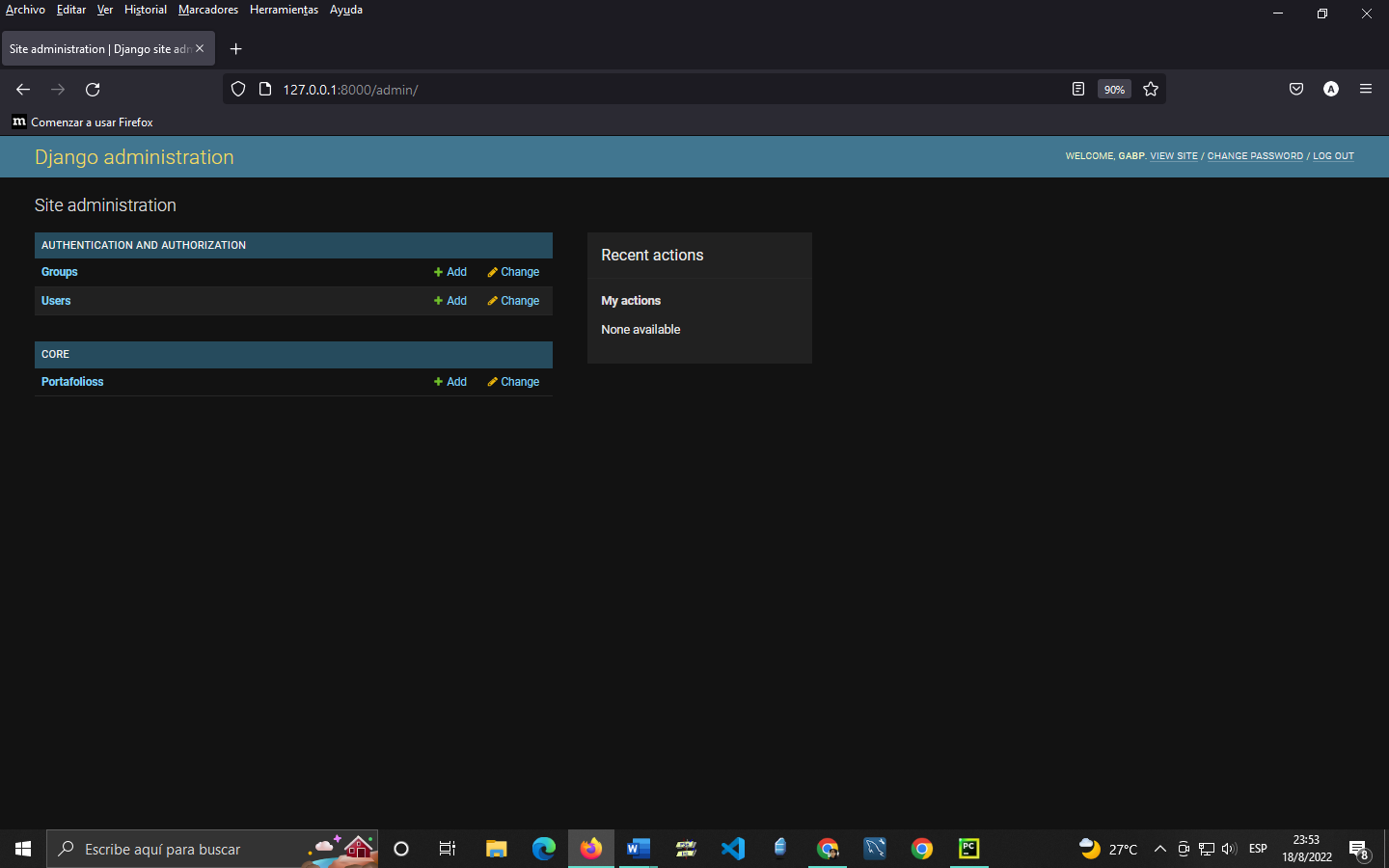
URLS



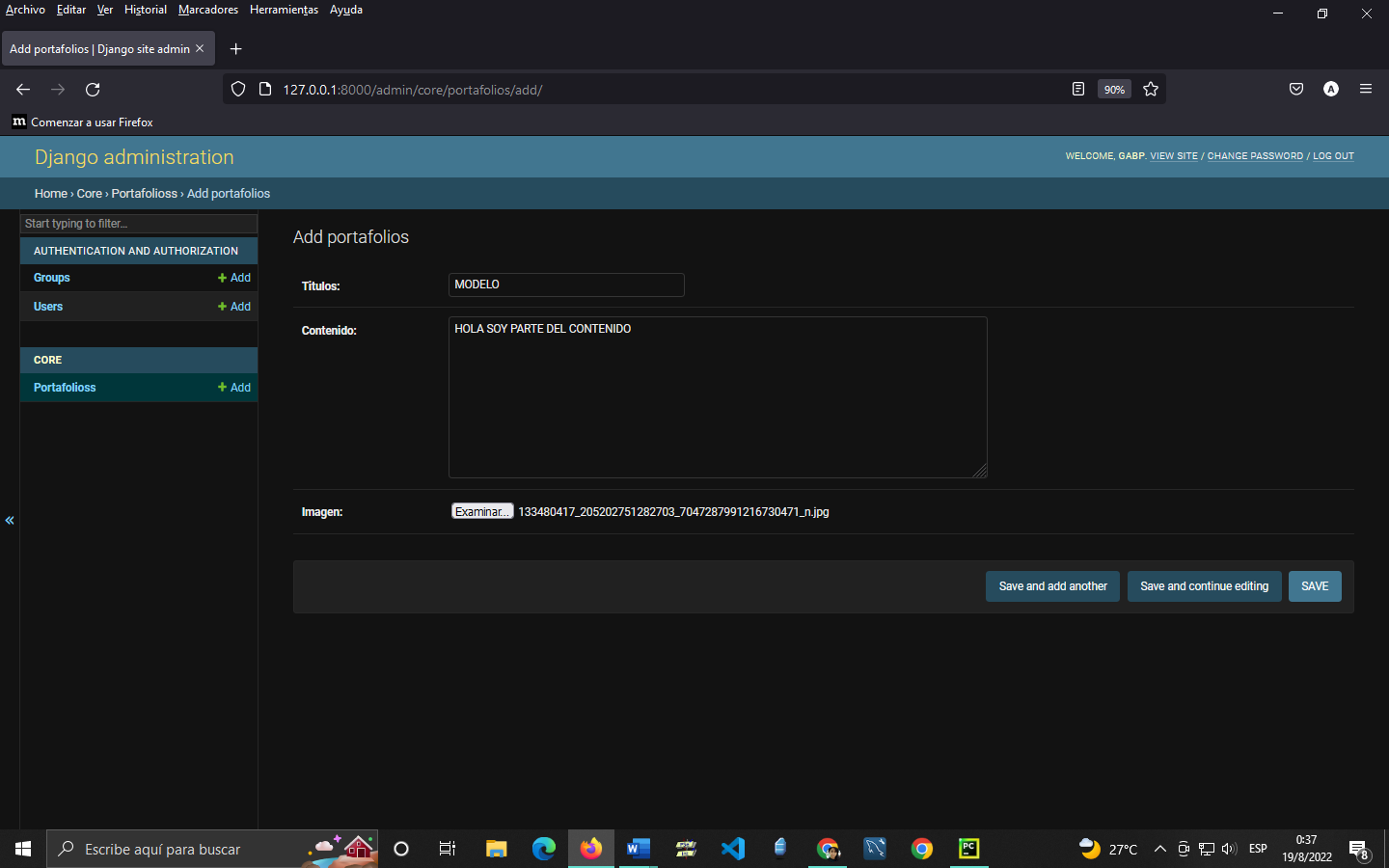
VIEWS



Resultado:



1. Ingresar información al modelo por el Panel de Administración.



1. Realizar la consulta de todo lo ingresado en el modelo desde el views.

