Guayaquil, 16 de agosto del 2022

\*MANUAL DE DJANGO\*



MATERIA: PROGRAMACION VISUAL

TALLER # 1

TEMA: GIT-GITHUB

CALIFICACION:2 PUNTOS

NOMBRE: MARIA JOSE PIN MUÑIZ

DOCENTE: CARLOS LUIS PAZMIÑO PALMA

CURSO: TERCERO “G”

CARRERA: DESARROLLO DE SOFTWARE



**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

escrito en Python. Un framework web es un conjunto de componentes que te

ayudan a desarrollar sitios web más fácil y rápidamente.

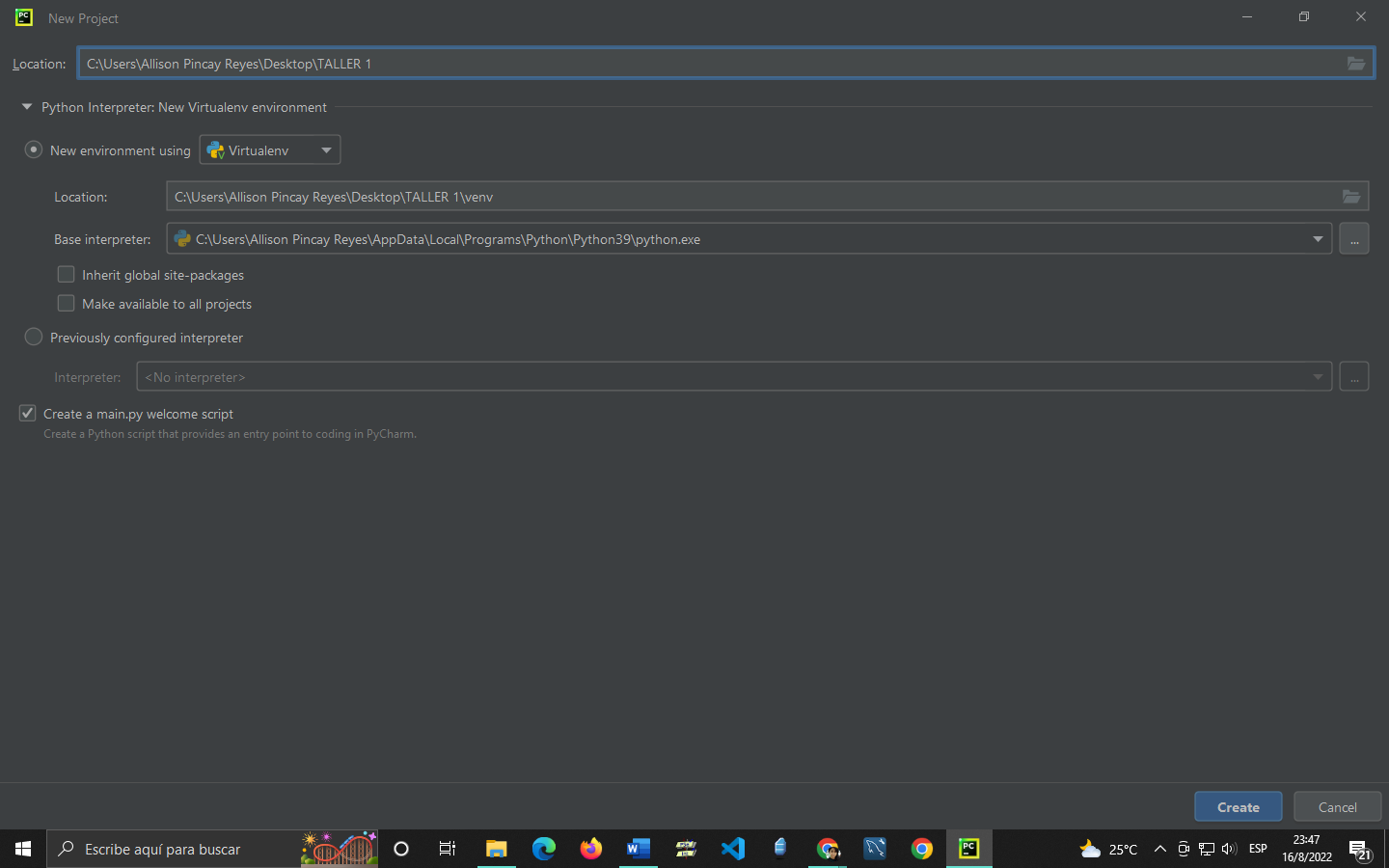
1. **¿Qué es la máquina virtual en Django**

El entorno de desarrollo es una instalación de Django en tu computadora local que puedes usar para desarrollar y probar apps

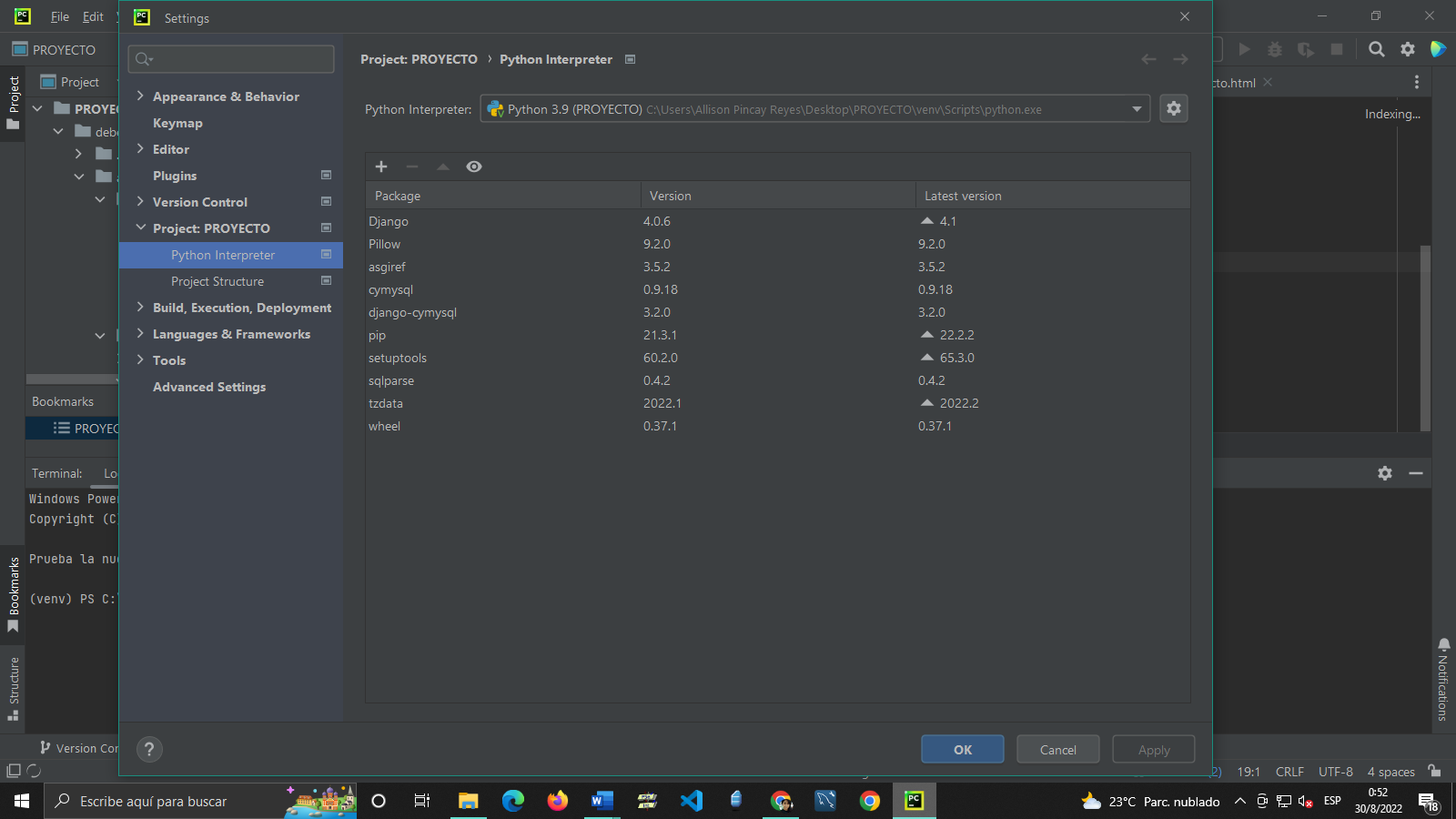
1. **¿Qué es MVT en Django?**

Es el Model, Views y templates

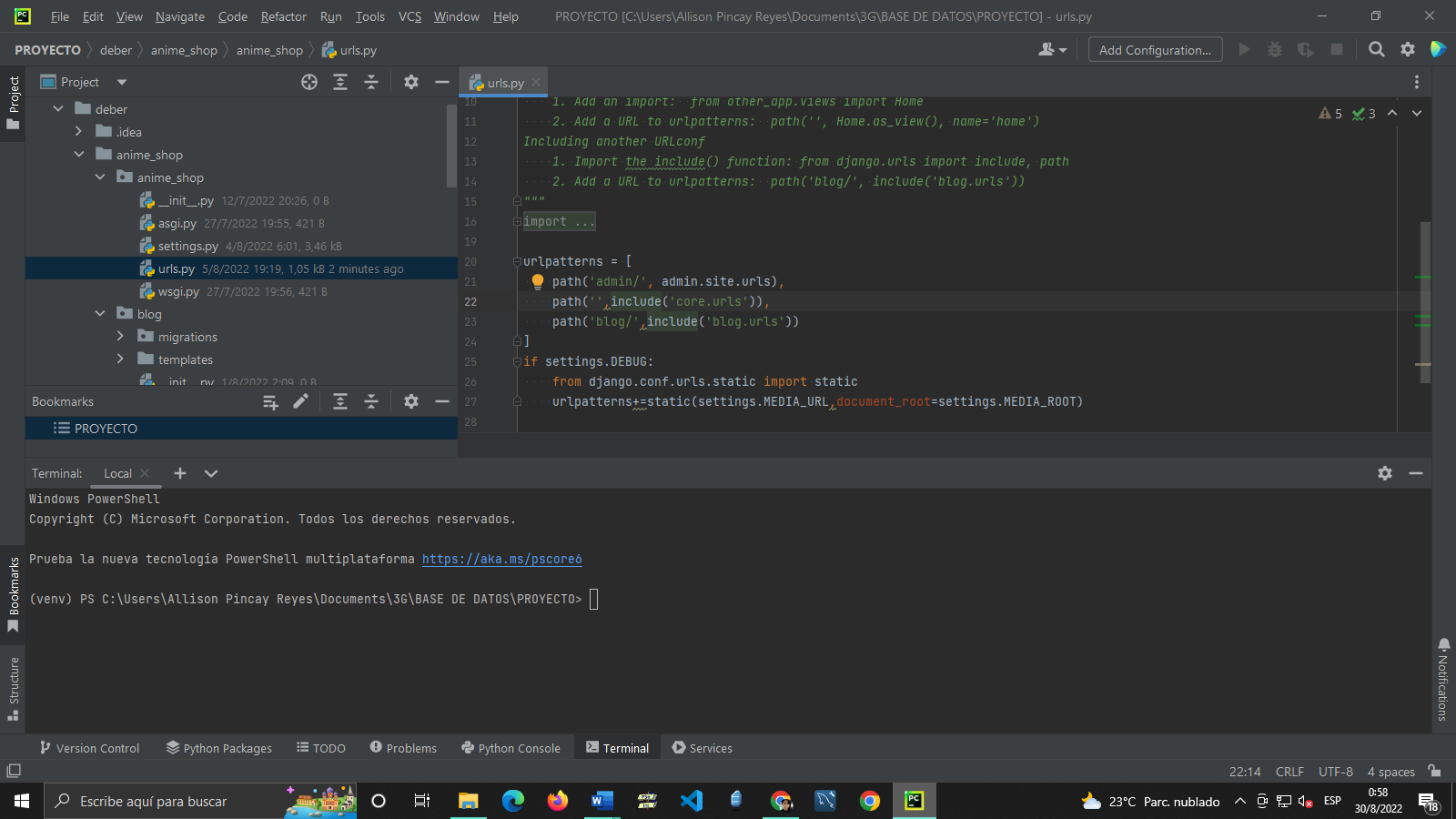
1. **Crear un proyecto con la máquina virtual**



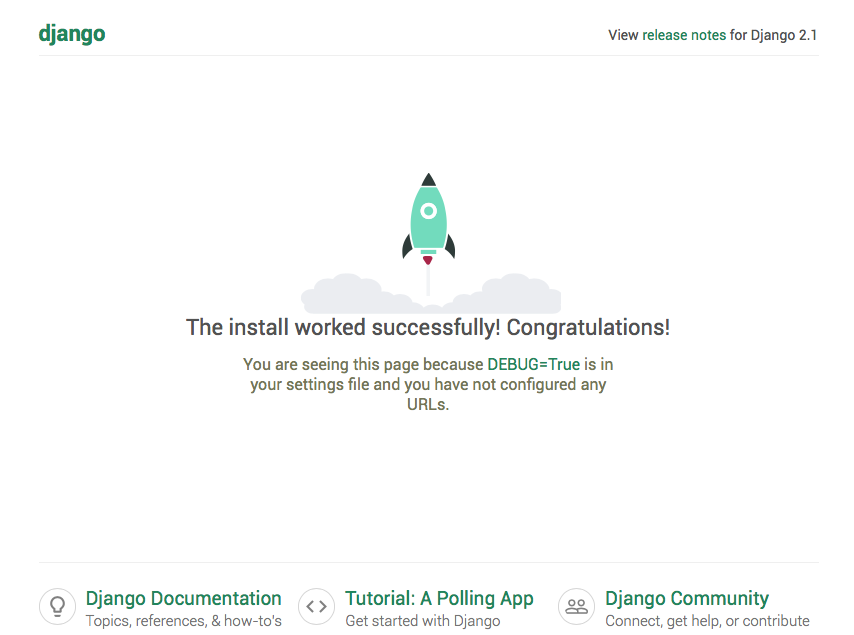
1. **Descargar los instaladores de Django al proyecto**



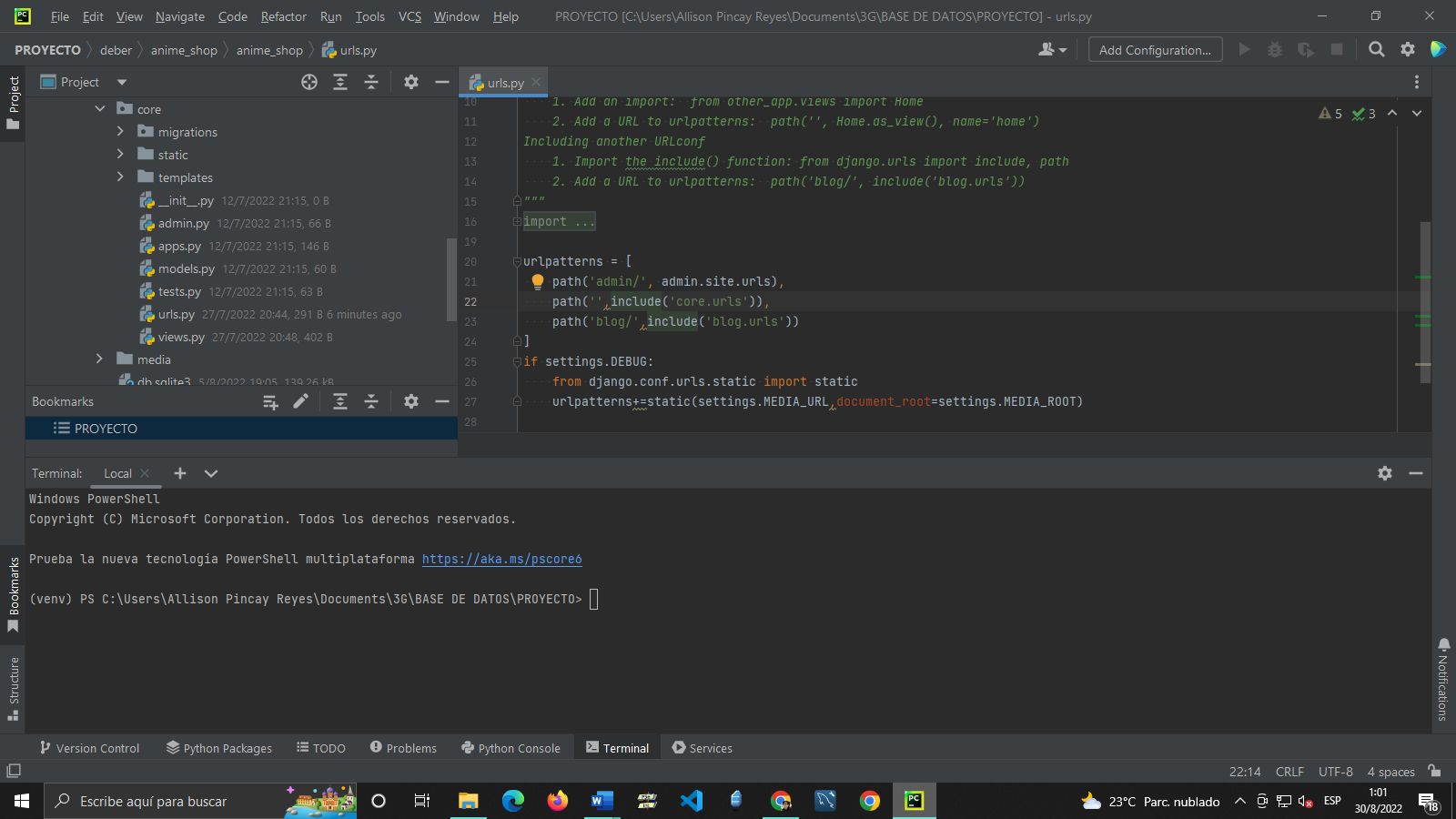
1. **Crear un proyecto para programar en Django**



1. **Ejecutar el proyecto y el mensaje de felicitaciones.**



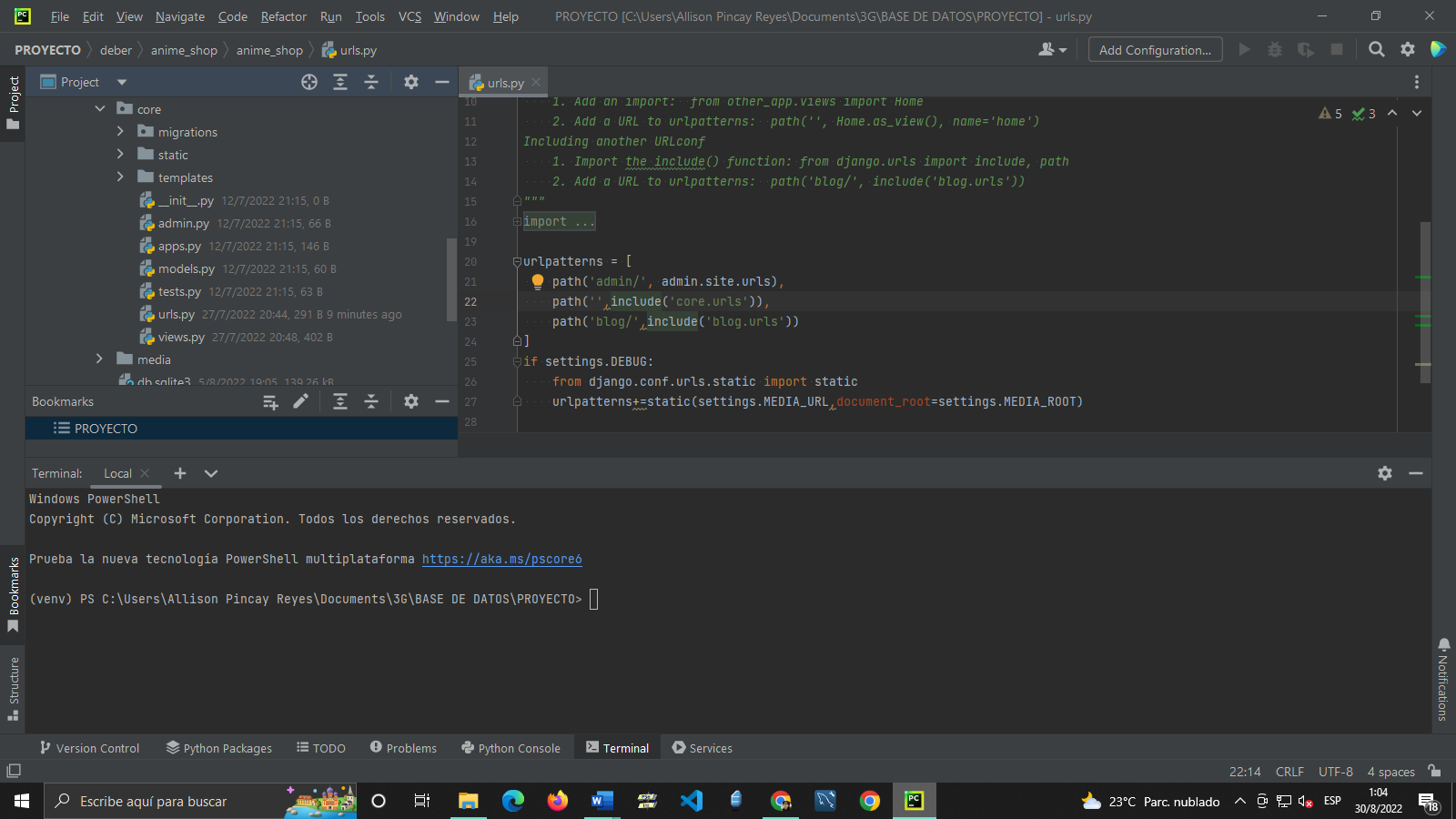
1. **Crear una Apps Core**



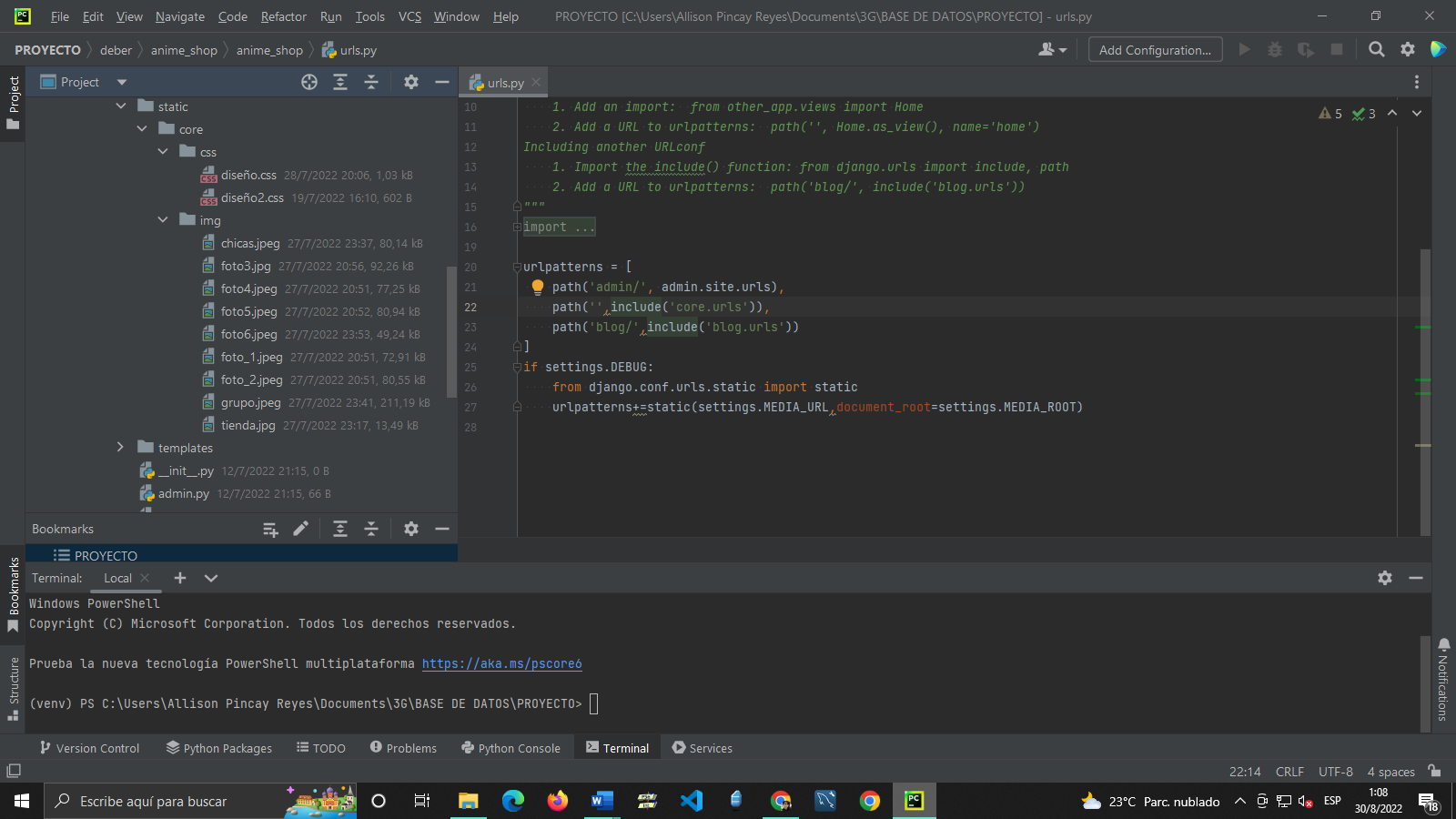
Usamos comando: “Python manage.py startapp core”

1. **¿Qué es la Carpeta Templates?**

Template almacena todo los HTML que usaremos en nuestro proyecto

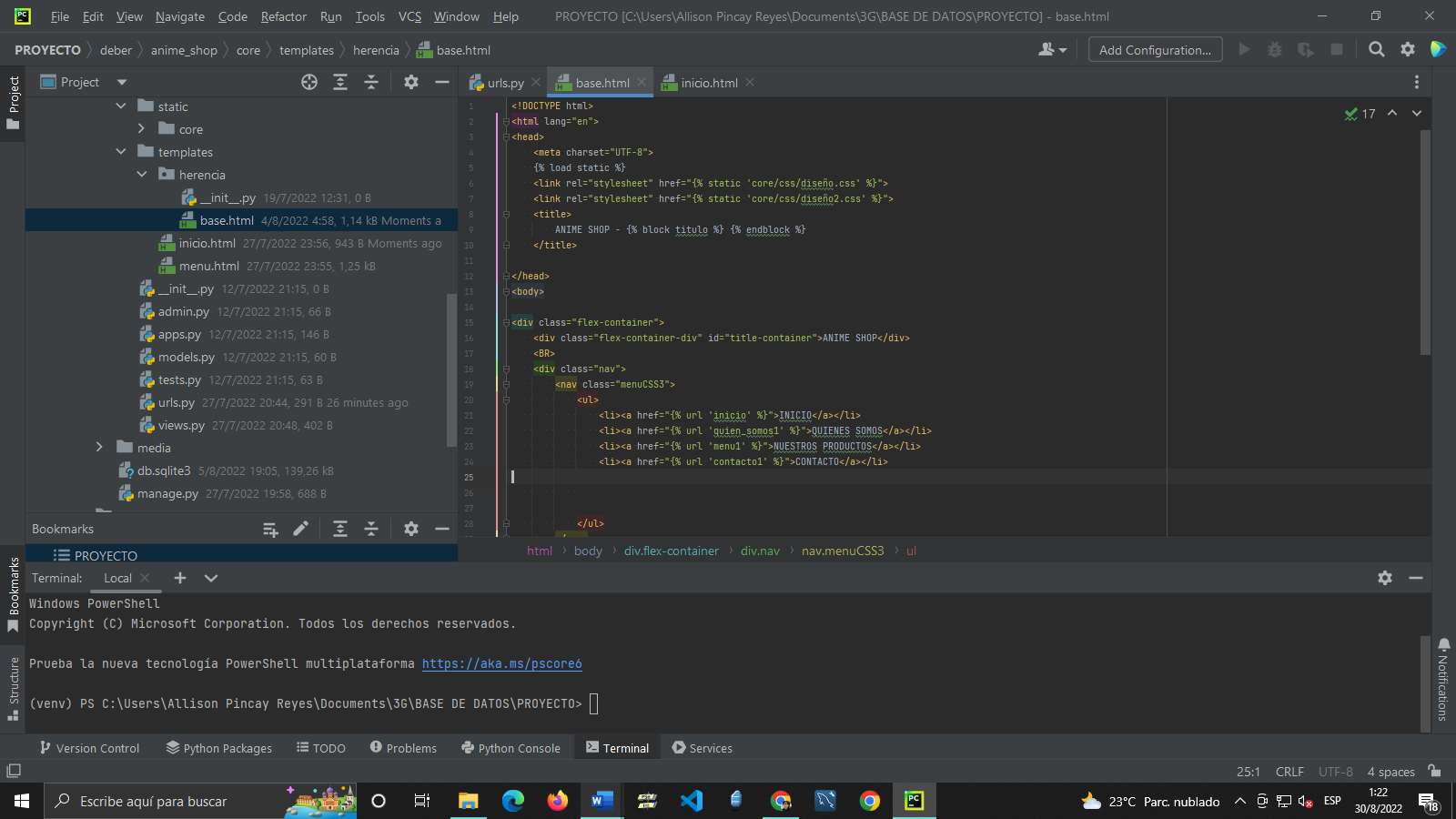


1. **¿Qué es la Carpeta stactic?**



En Static creamos nuestras carpetas css, imágenes, entre otros archivos de estilo que deseamos que se mantengan en nuestro proyecto.

1. **Crear un archivo base html en la APPS core**



Nuestro HTML base de la core

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

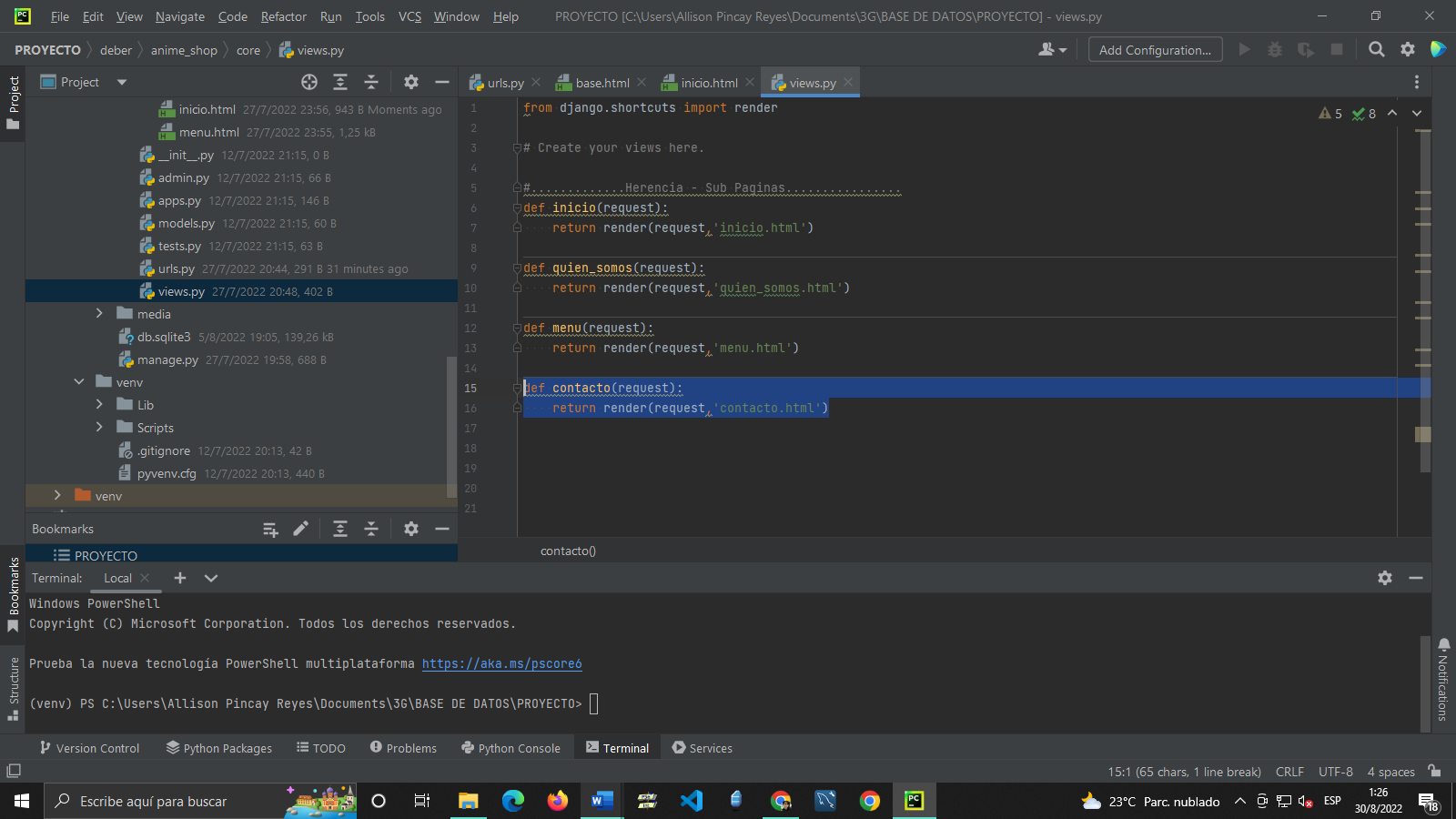
<link rel="stylesheet" href="{% static 'core/css/diseño.css' %}">  
<link rel="stylesheet" href="{% static 'core/css/diseño2.css' %}">

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

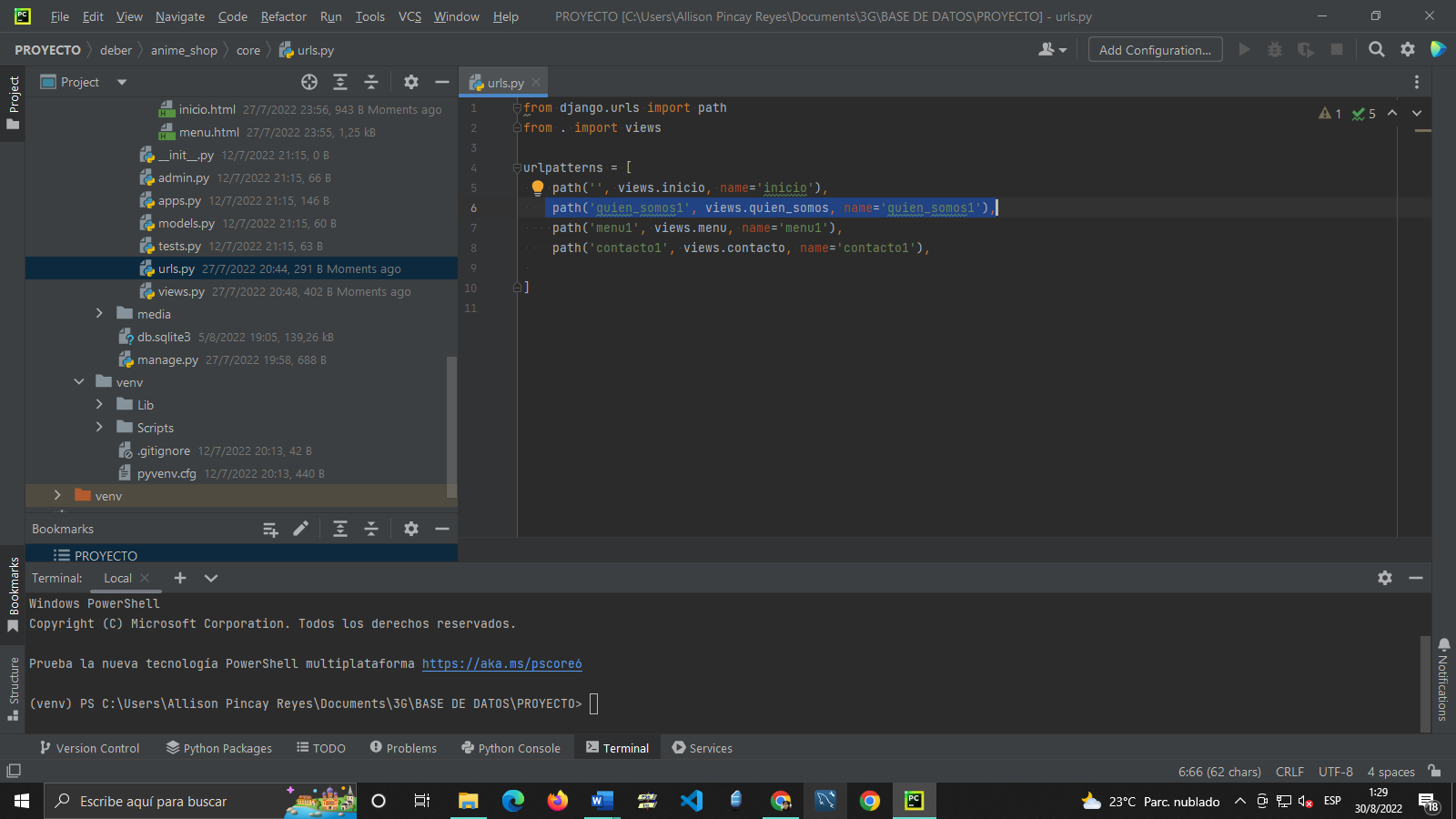
{% extends 'herencia/base.html'%}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{% block titulo %} Inicio {% endblock %}</title>  
 {% load static %}  
 <link rel="stylesheet" href="{% static 'core/css/diseño.css' %}">  
 <link rel="stylesheet" href="{% static 'core/css/diseño2.css' %}">  
</head>  
<body>  
{% block body %}  
<div class="flex-container-img">  
 <div>  
 <h1>TIENDA</h1>  
 <img src="static/core/img/tienda.jpg" height="350" width="600px">  
 </div>  
 <div>  
 <h1>FIGURAS</h1>  
 <img src="static/core/img/foto6.jpeg" height="350" width="600px">  
 </div>  
 <div>  
 <h1>MONEDEROS</h1>  
 <img src="static/core/img/foto\_2.jpeg" height="350" width="600px">  
 </div>  
 <div>  
 <h1>PINES</h1>  
 <img src="static/core/img/foto3.jpg" height="350" width="600px">  
 </div>  
</div>  
{% endblock %}  
</body>  
  
</html>

Un archivo hijo HTML a nuestro HTML base nos da un llamado que nuestro HTML

1. **Crear un view que llame al html hijo**



1. **Crear la urls que llame al views.**



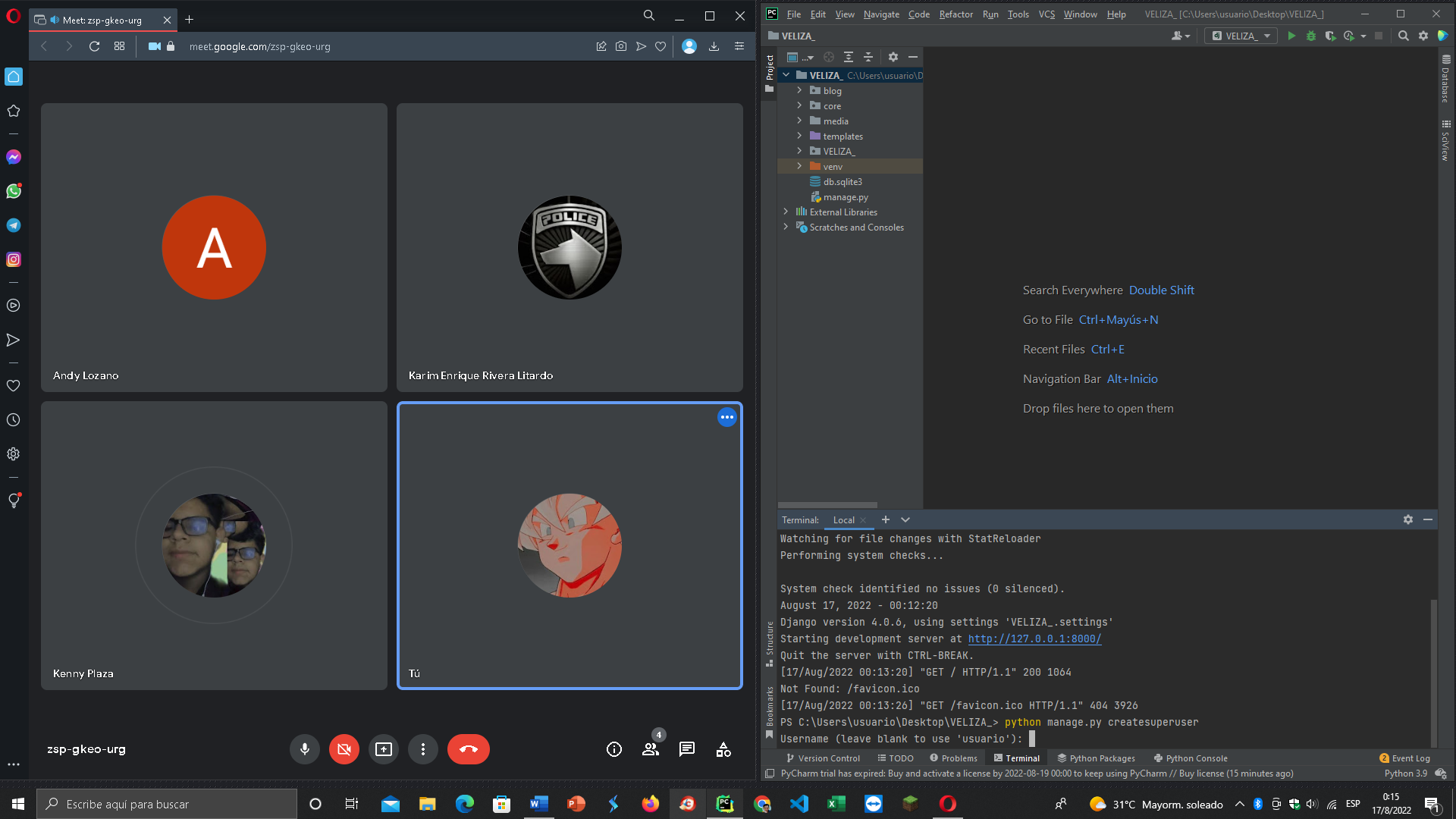
1. **Integrar la aplicación APPS core al proyecto principal**

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

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

**Usamos comando “Python manage.py migrate”**

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



Mediante código **“Python manage.py createsuperuser”** para darle un nombre,

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 quienes\_somosM(models.Model):  
 titulo=models.CharField(max\_length=200)  
 contenido=models.TextField()  
 imagen=models.ImageField(upload\_to="blog")  
 fechacreacion=models.DateTimeField(auto\_now\_add=True)  
 fechamodificacion=models.DateTimeField(auto\_now=True)  
 def \_\_str\_\_(self):  
 return self.titulo  
  
class contactanosM(models.Model):  
 titulo=models.CharField(max\_length=200)  
 contenido=models.TextField()  
 imagen=models.ImageField(upload\_to="blog")  
 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.**

Al migrar la base al panel de administración usamos dos códigos **“Python manage.py makemigrations blog” / “Python manage.py migrate blog”**

1. **Integrar el Modelo al Panel de Administración.**

from django.contrib import admin  
from .models import contactanosM, quienes\_somosM  
# Register your models here.  
  
admin.site.register(contactanosM),  
admin.site.register(quienes\_somosM)

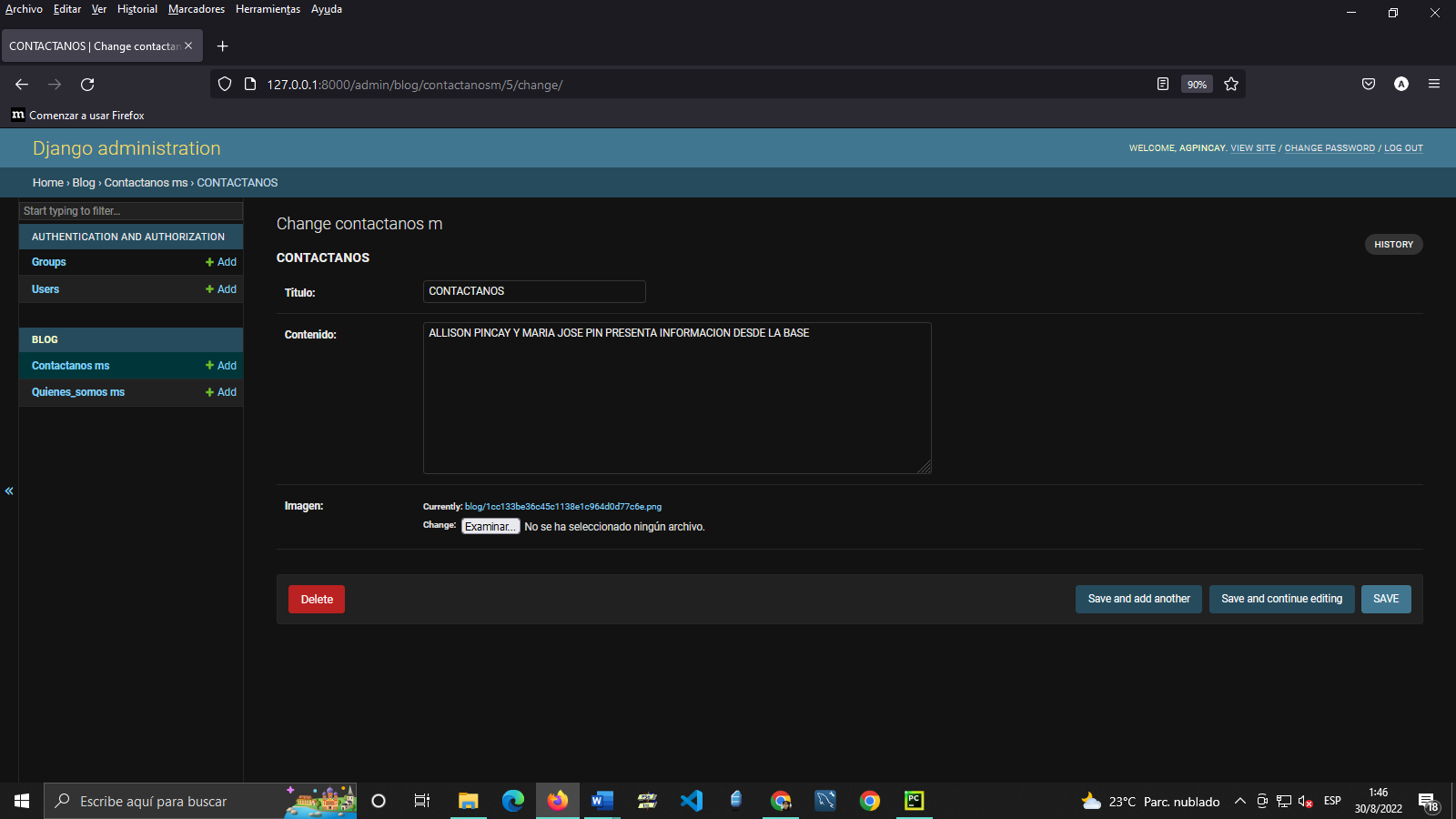
**URLS**

from django.urls import path  
from . import views  
  
urlpatterns = [  
  
 path('contacto1', views.contacto, name='contacto1'),  
 path('quien\_somos1', views.quien\_somos, name='quien\_somos1'),  
   
]

**VIEWS**

from django.shortcuts import render  
from .models import quienes\_somosM  
from .models import contactanosM  
# Create your views here.  
#def blog (request):  
# select=portafolio.objects.all()  
# return render(request,'blog.html',{'Enviablog':select})  
  
def quien\_somos(request):  
 objCondatos=quienes\_somosM.objects.all()  
 return render(request,'quien\_somos.html',{'Enviablog':objCondatos})  
  
def contacto(request):  
 objCondatos=contactanosM.objects.all()  
 return render(request,'contacto.html',{'Enviablog':objCondatos})

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



1. **Mostrar los datos guardados en el modelo al html hijo.**

