Guayaquil, 16 de agosto del 2022

MANUAL DE DJANGO



MATERIA: PROGRAMACION VISUAL

TALLER # 1

TEMA: GIT-GITHUB

CALIFICACION:2 PUNTOS

NOMBRE: ERIKA MUQUINCHE

DOCENTE: CARLOS LUIS PAZMIÑO PALMA

CURSO: TERCERO “H”

CARRERA: DESARROLLO DE SOFTWARE



***Los temas a desarrollar son los siguientes:***

1. **¿Qué es Django?**

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, un panel de

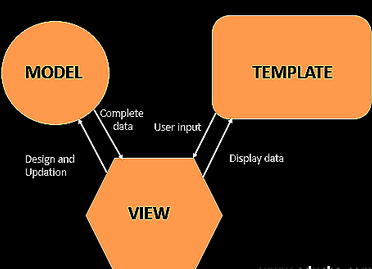
administración para tu sitio web, formularios, una forma de subir archivos, etc.

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

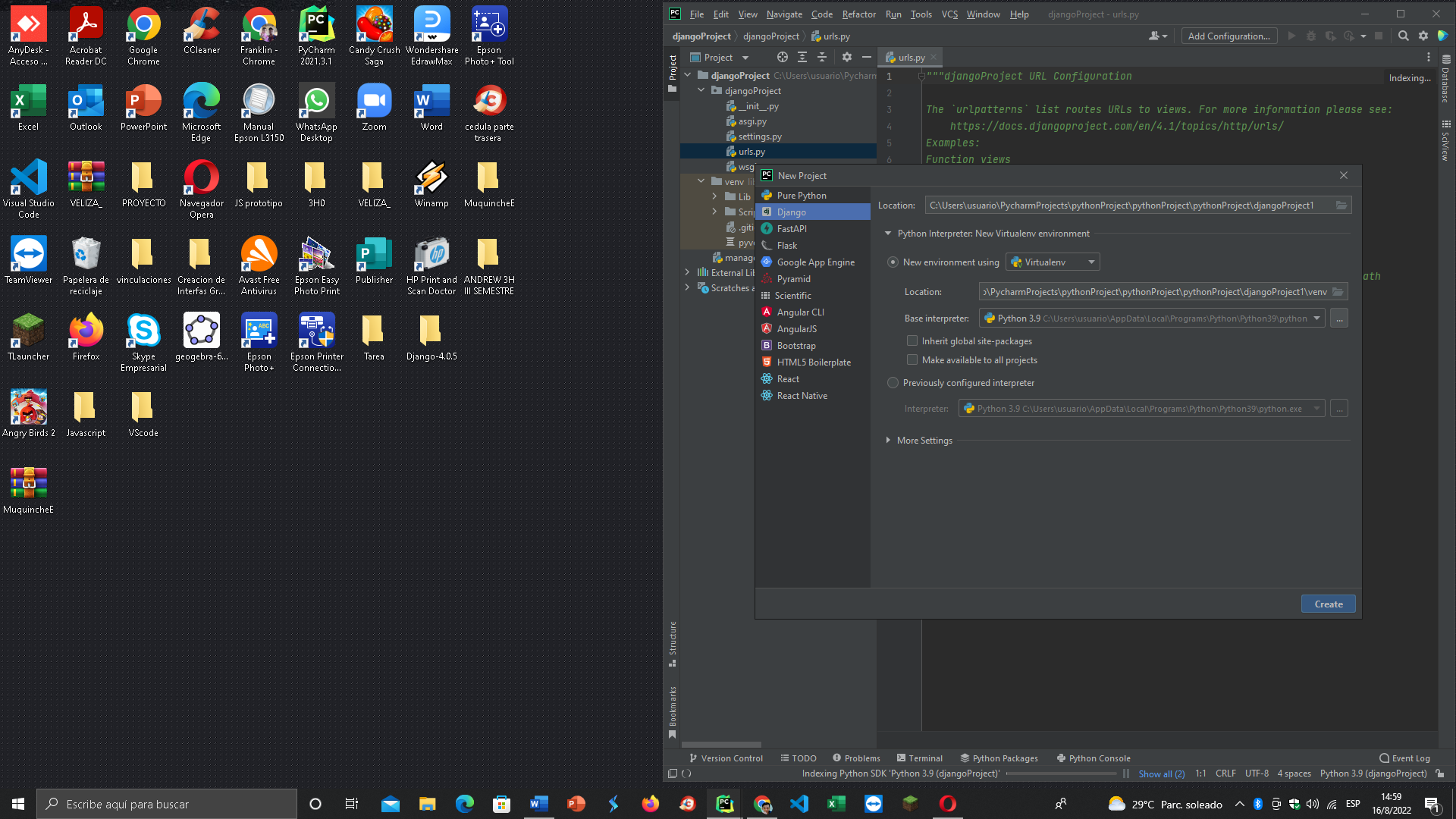
El entorno de desarrollo es una instalación de Django en tu computadora local que puedes usar para desarrollar y probar apps

1. **¿Qué es MVT en Django?**

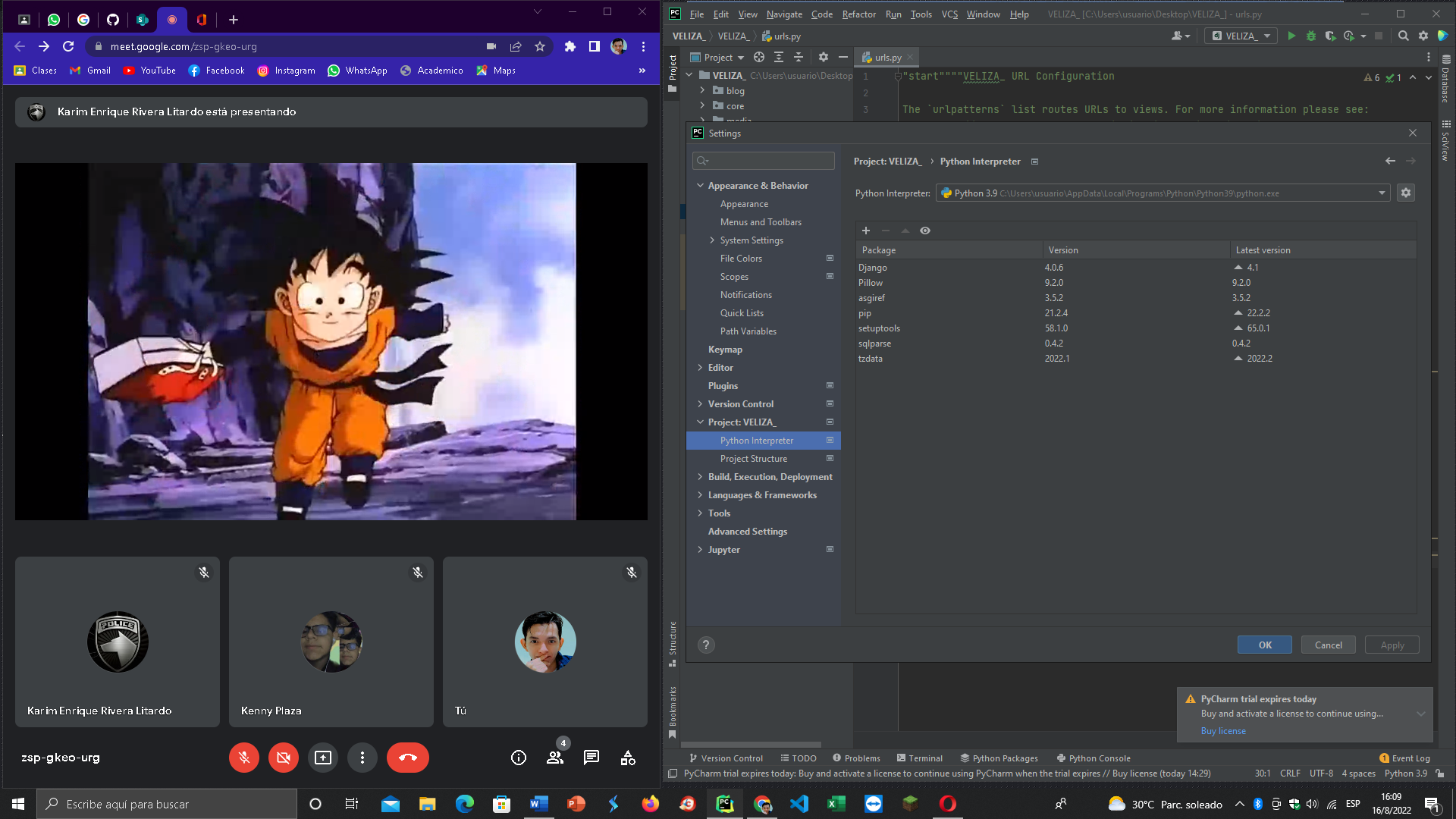
Es el Model, Views y templates



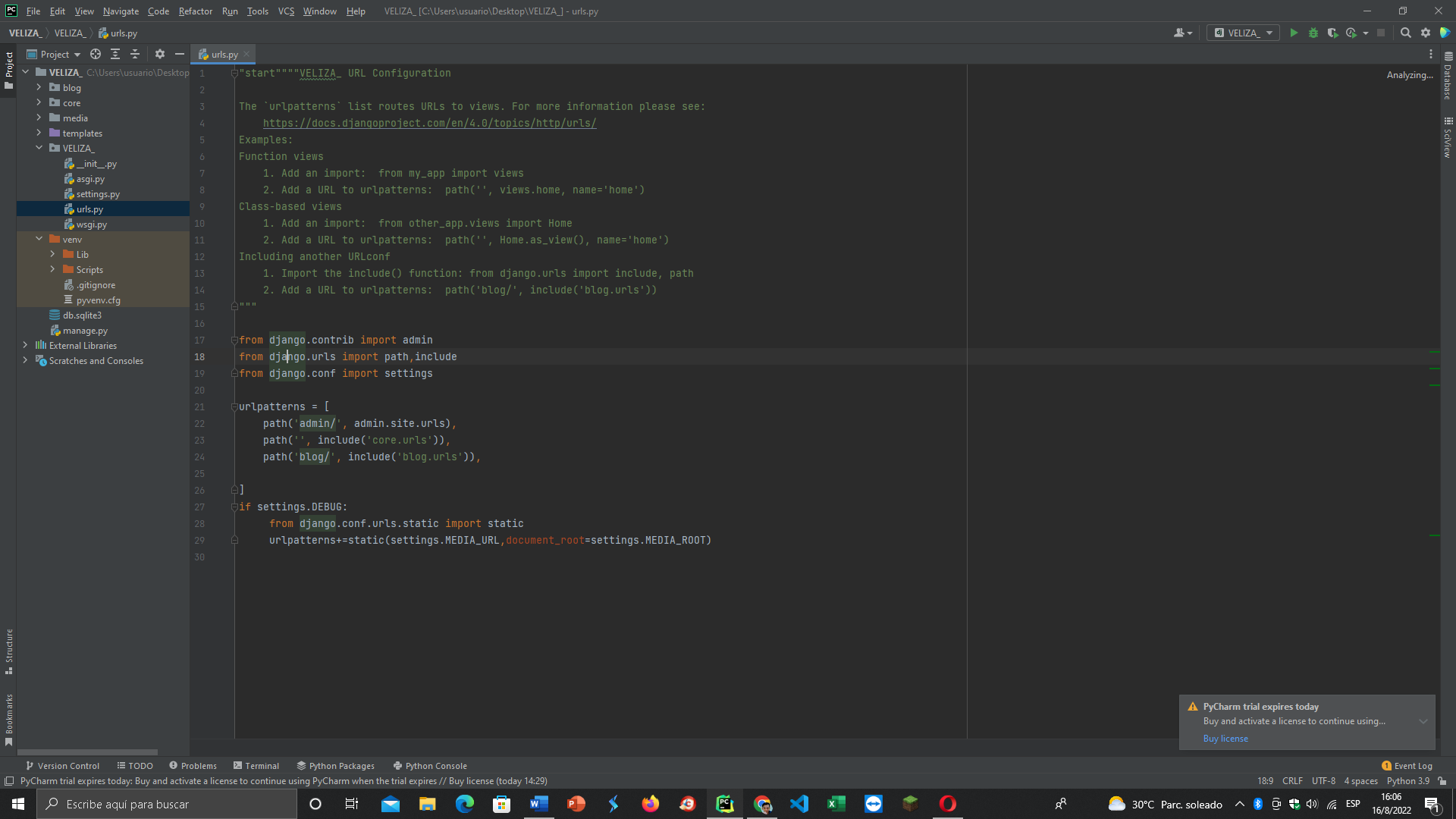
1. **Crear un proyecto con la máquina virtual**



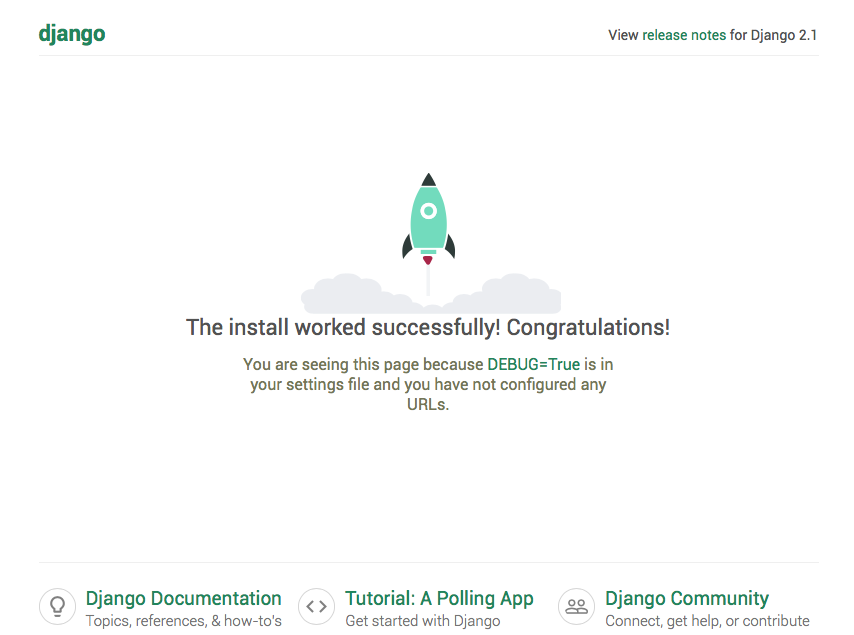
1. **Descargar los instaladores de Django al proyecto**



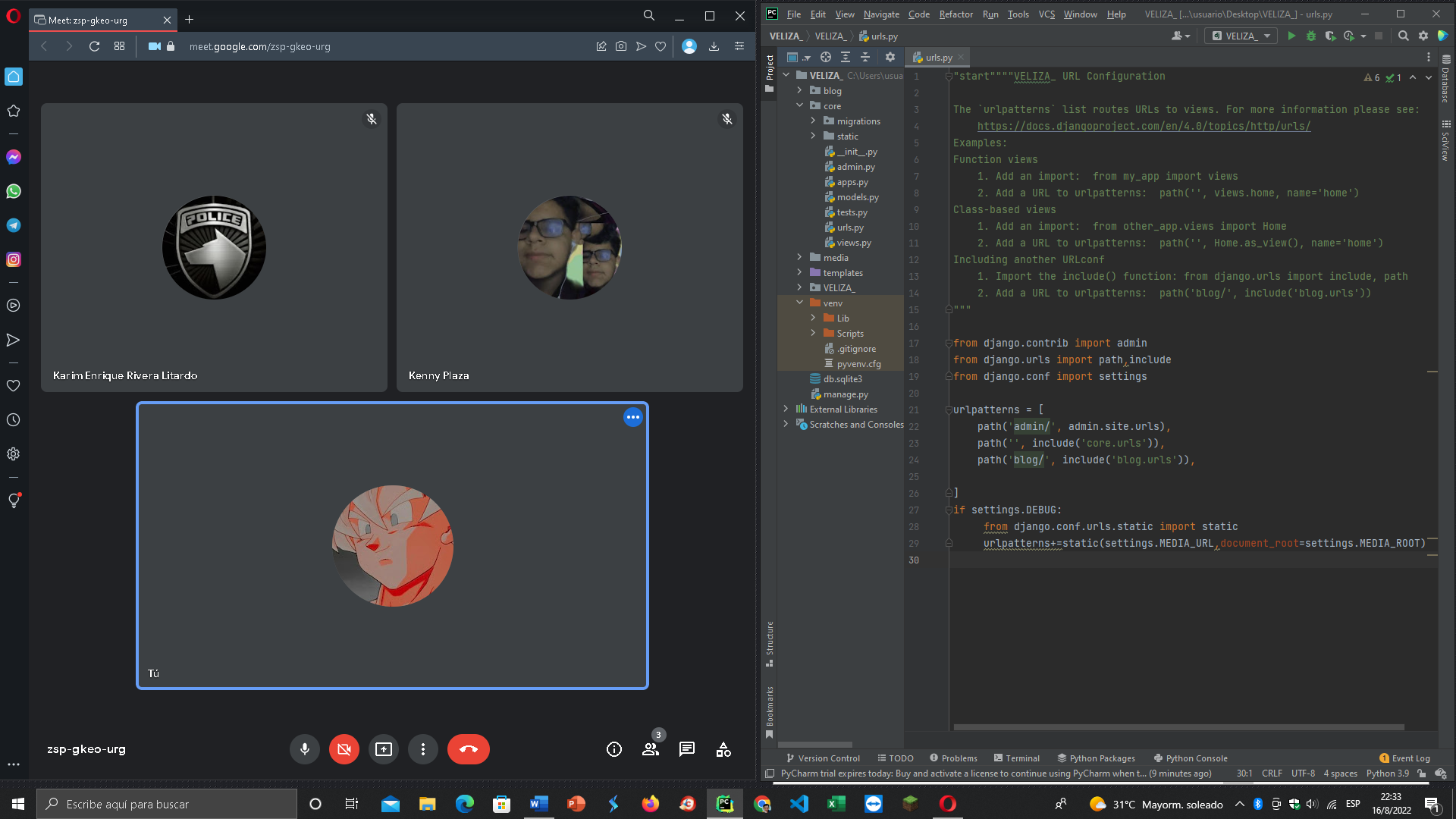
1. **Crear un proyecto para programar en Django**



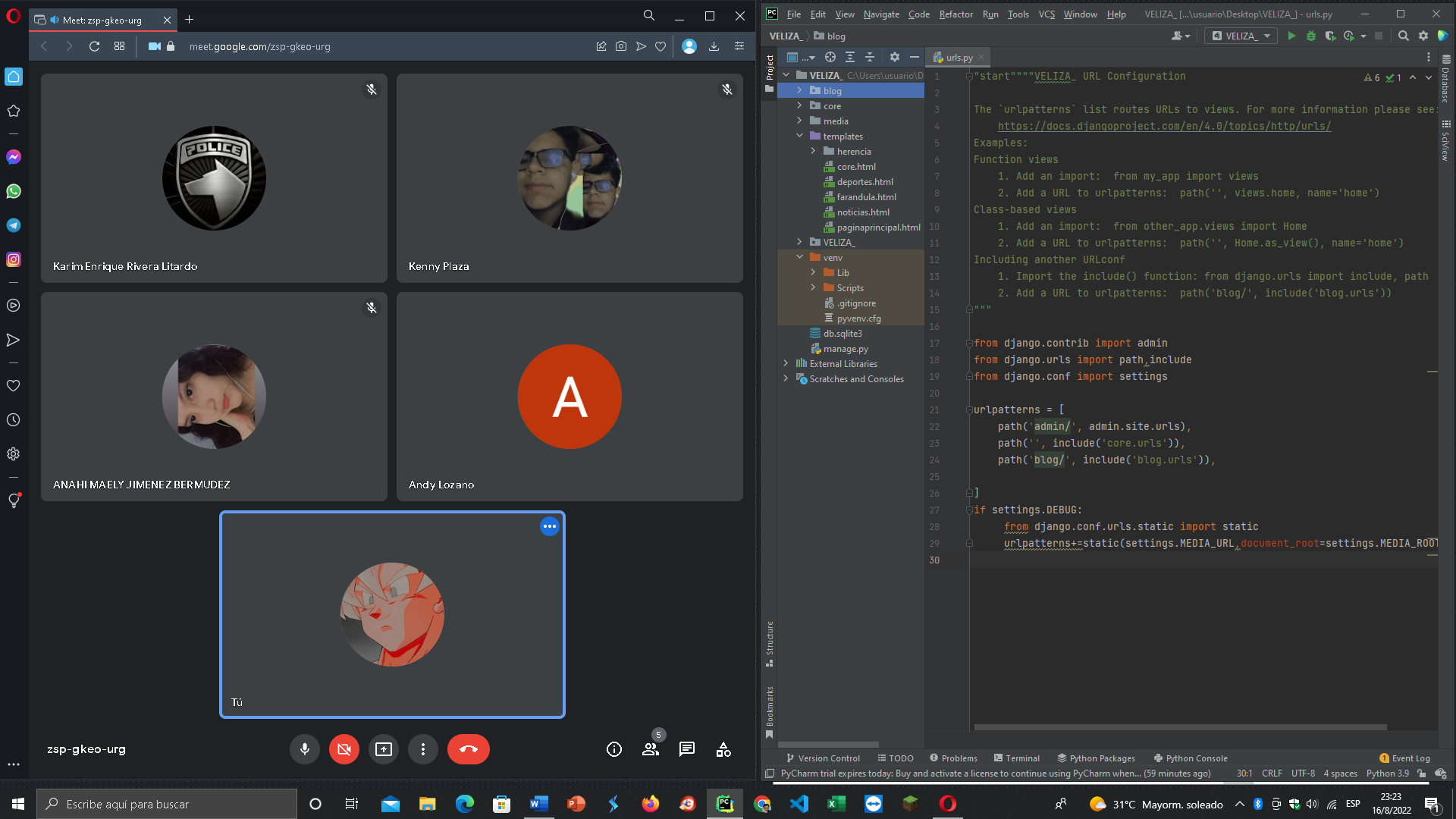
1. **Ejecutar el proyecto y el mensaje de felicitaciones.**



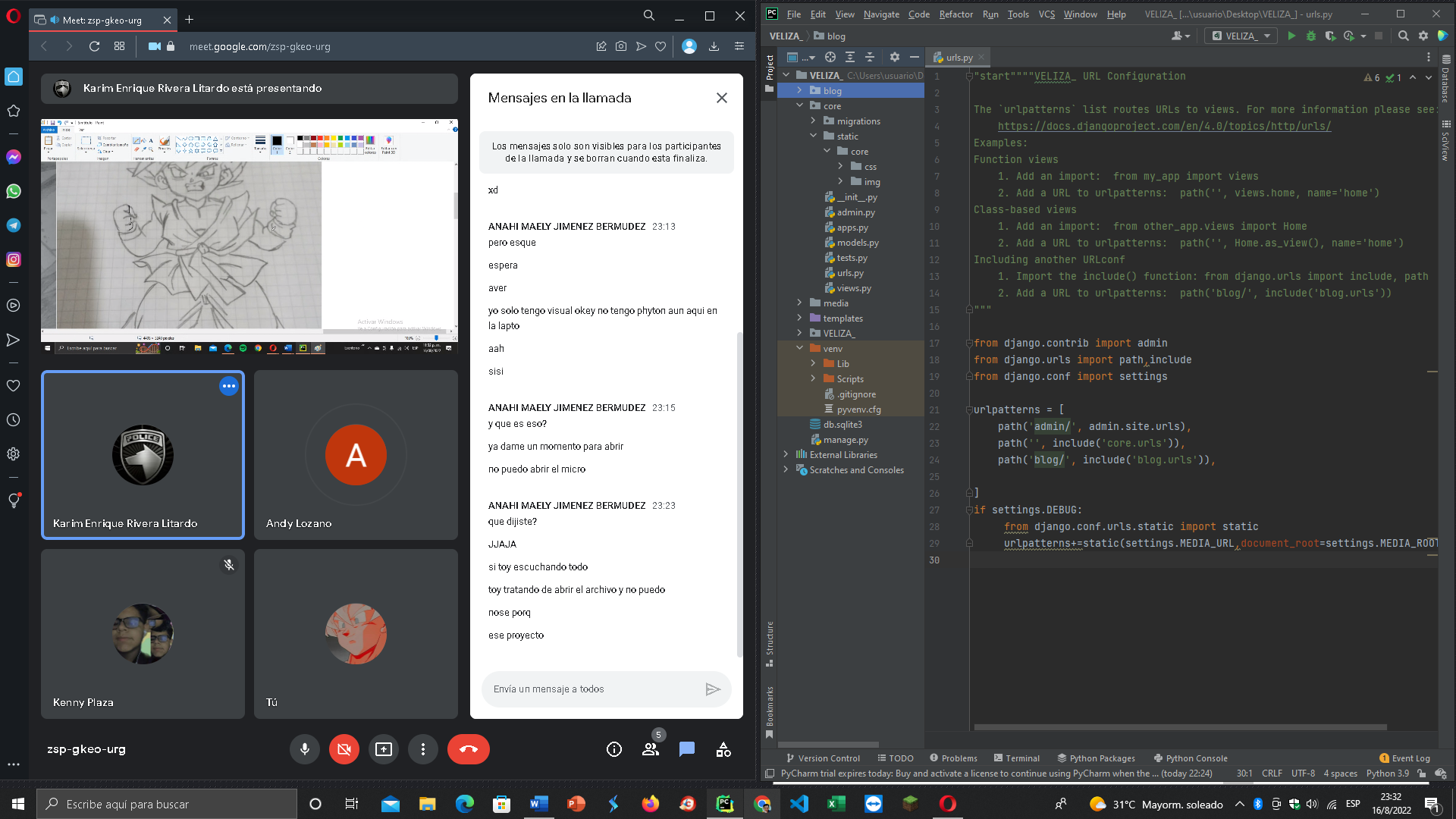
1. **Crear una Apps Core**

la creación de Core se creo mediante terminal con el código “Python manage.py startapp core”

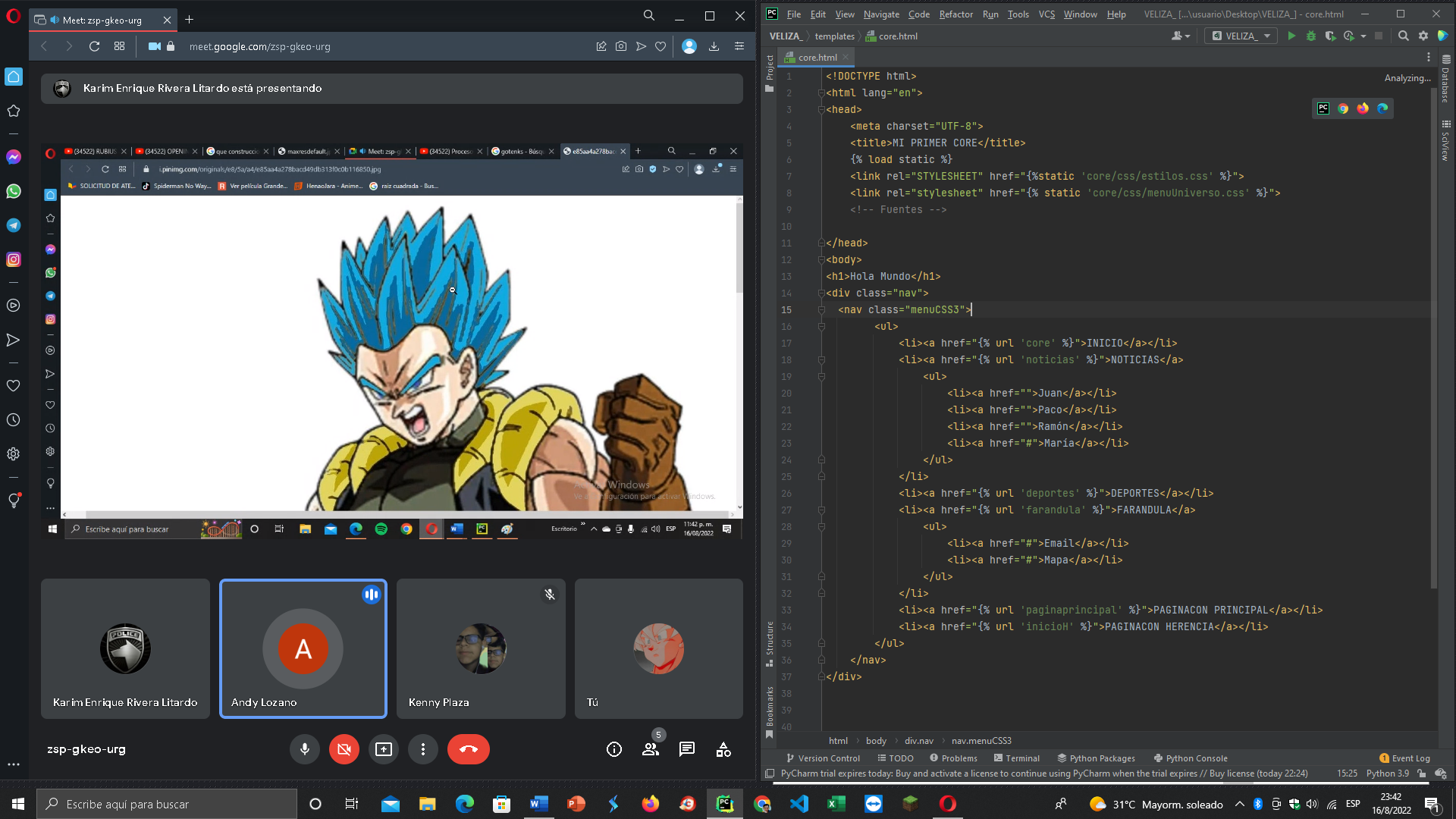
1. **¿Qué es la Carpeta Templates?**

nuestro template almacena todo los html que usaremos en nuestro proyecto

1. **¿Qué es la Carpeta stactic?**

en esta carpeta nosotros creamos nuestras subcarpetas donde almacenaremos nuestras imágenes y los estilos css

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

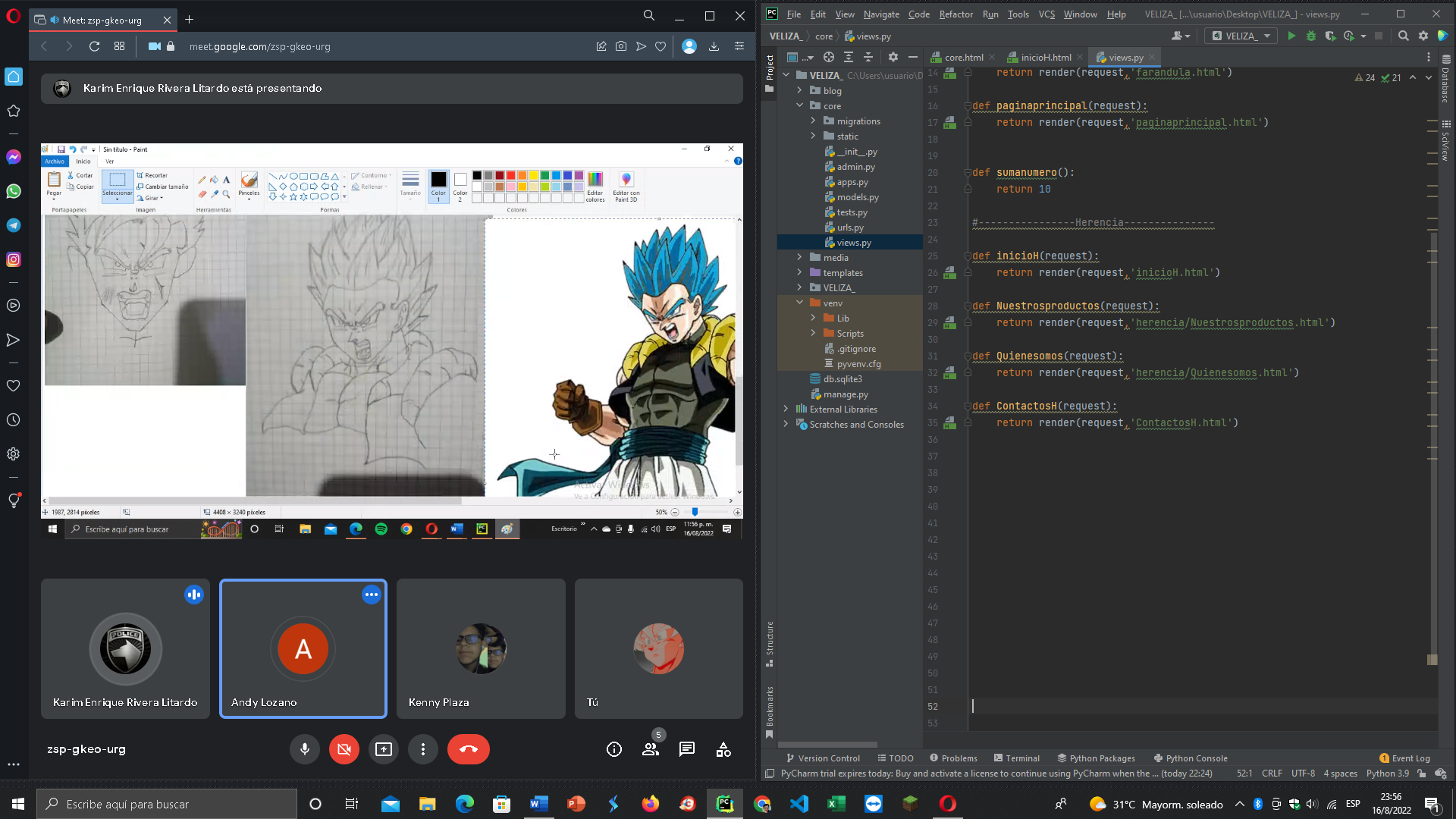
Por nuestro link llamando a la carpeta static donde contiene nuestros estilos

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

{% extends 'principalH.html' %}  
{% load static %}  
{% block titulo %}  
 3H  
{% endblock %}  
{% block contenidohija %}  
<h1>Gracias<h1>  
  
{% endblock %}

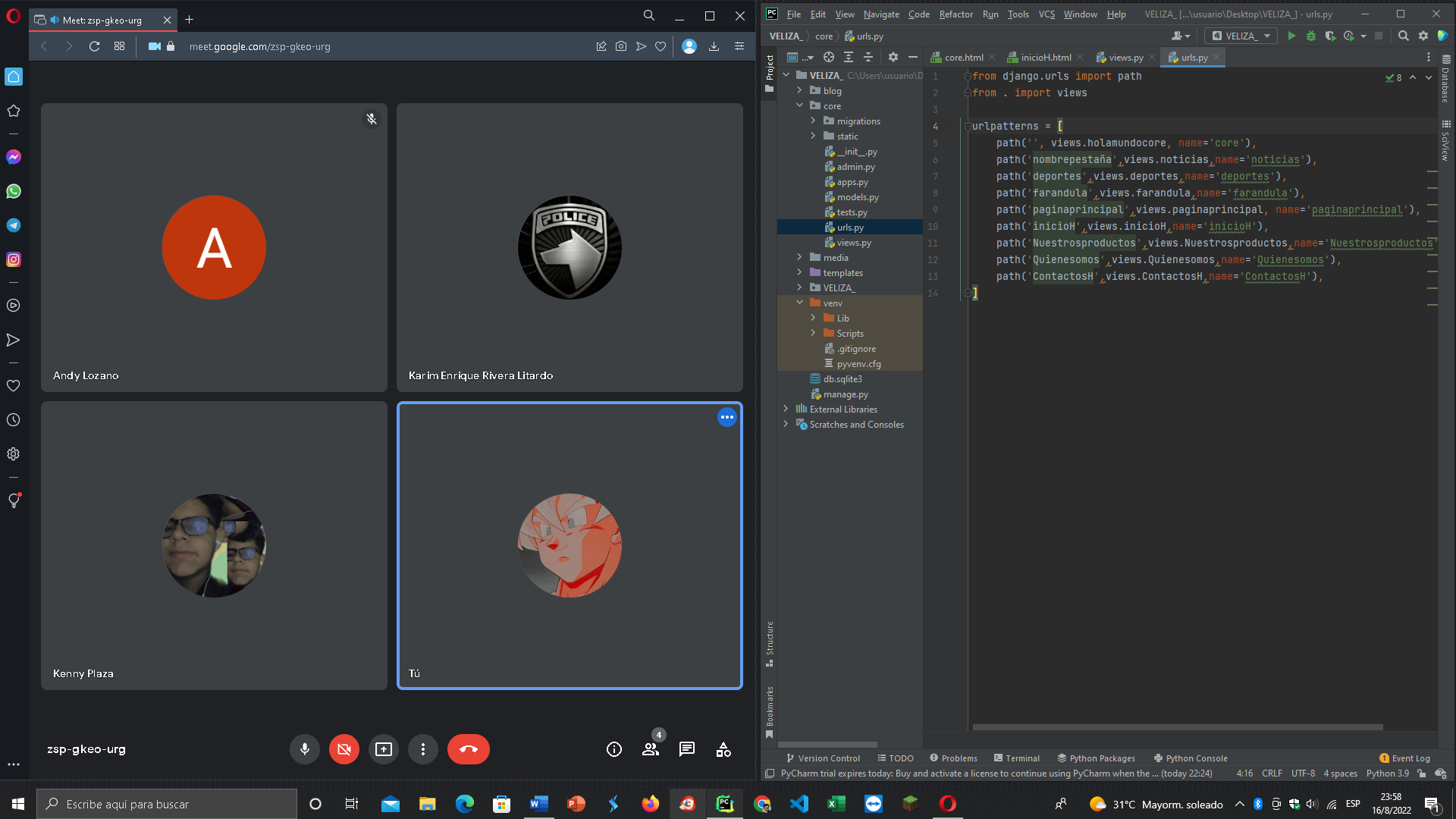
Un archivo hijo HTML a nuestro HTML base nos da un llamado que nuestro HTML

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



Creamos el llamado a nuestro html hijo

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



Llamamos nuestro views

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

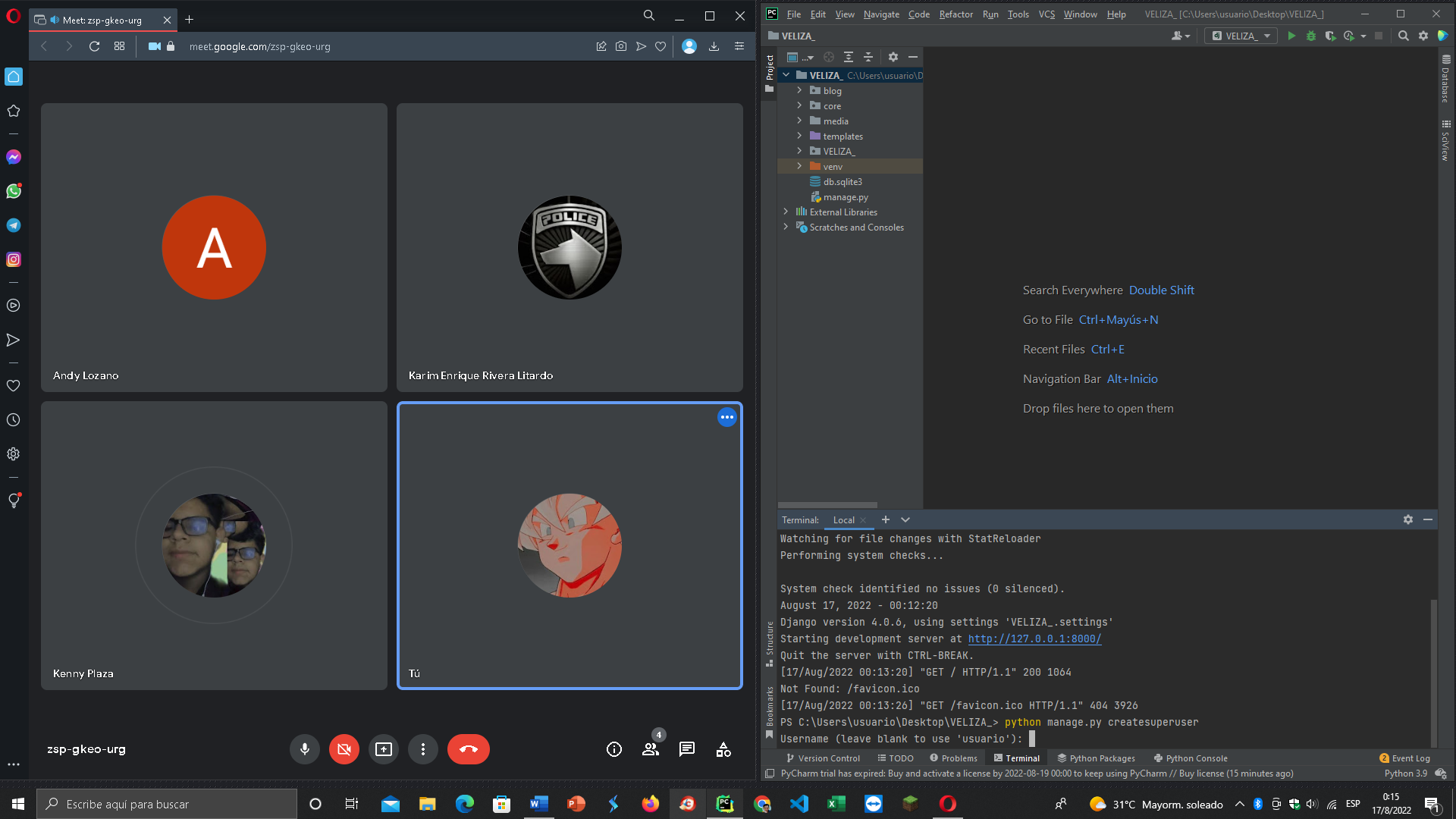
from django.contrib import admin  
from django.urls import path,include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', include('core.urls')),

integramos nuestra Core a mi proyecto principal

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

Para crear nuestra tabla lo hacemos mediante código para la creación de nuestra base de datos **“Python manage.py migrate”**,y se nos crea nuestra base de datos automáticamente

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



Mediante código **“Python manage.py createsuperuser”** para darle un nombre, crear contraseña y llenar los datos que nos pide.

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 tkinter import image\_names  
from django.db import models  
  
# Create your models here.  
class portafolio(models.Model):  
 titulo=models.CharField(max\_length=200,verbose\_name="Titulos")  
 contenido=models.TextField()   
 image=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.**

Procedemos a codificar nuestro modelo e integrar a mi panel de administración

**ADMIN**

from django.contrib import admin  
from .models import portafolio  
from .models import inicio2  
  
# Register your models here.  
  
admin.site.register(portafolio)  
admin.site.register(inicio2)

**URLS**

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('blog/', views.blog, name='blog'),

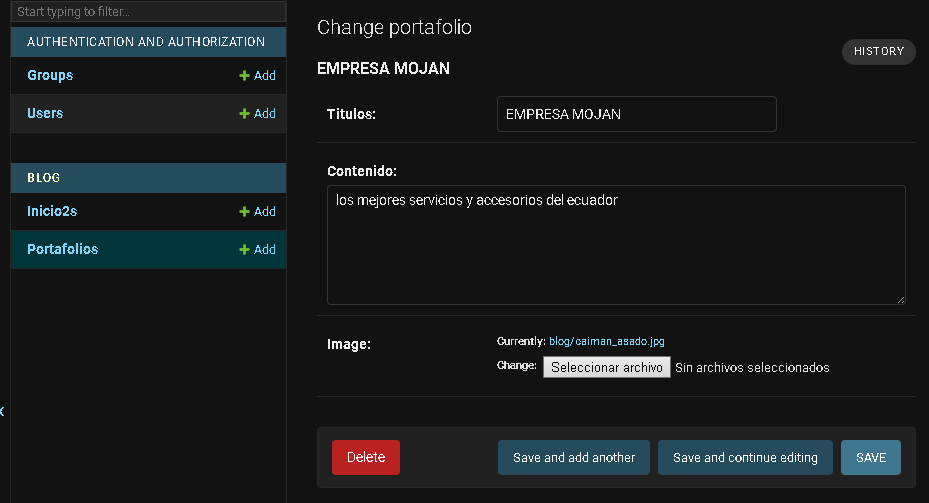
**VIEWS**

from django.shortcuts import render  
from .models import portafolio  
from .models import inicio2  
# Create your views here.  
  
def blog (request):  
 objCondatos= portafolio.objects.all()  
 return render(request, 'inicioH.html',{'Enviablog':objCondatos})

**RESULTADO**

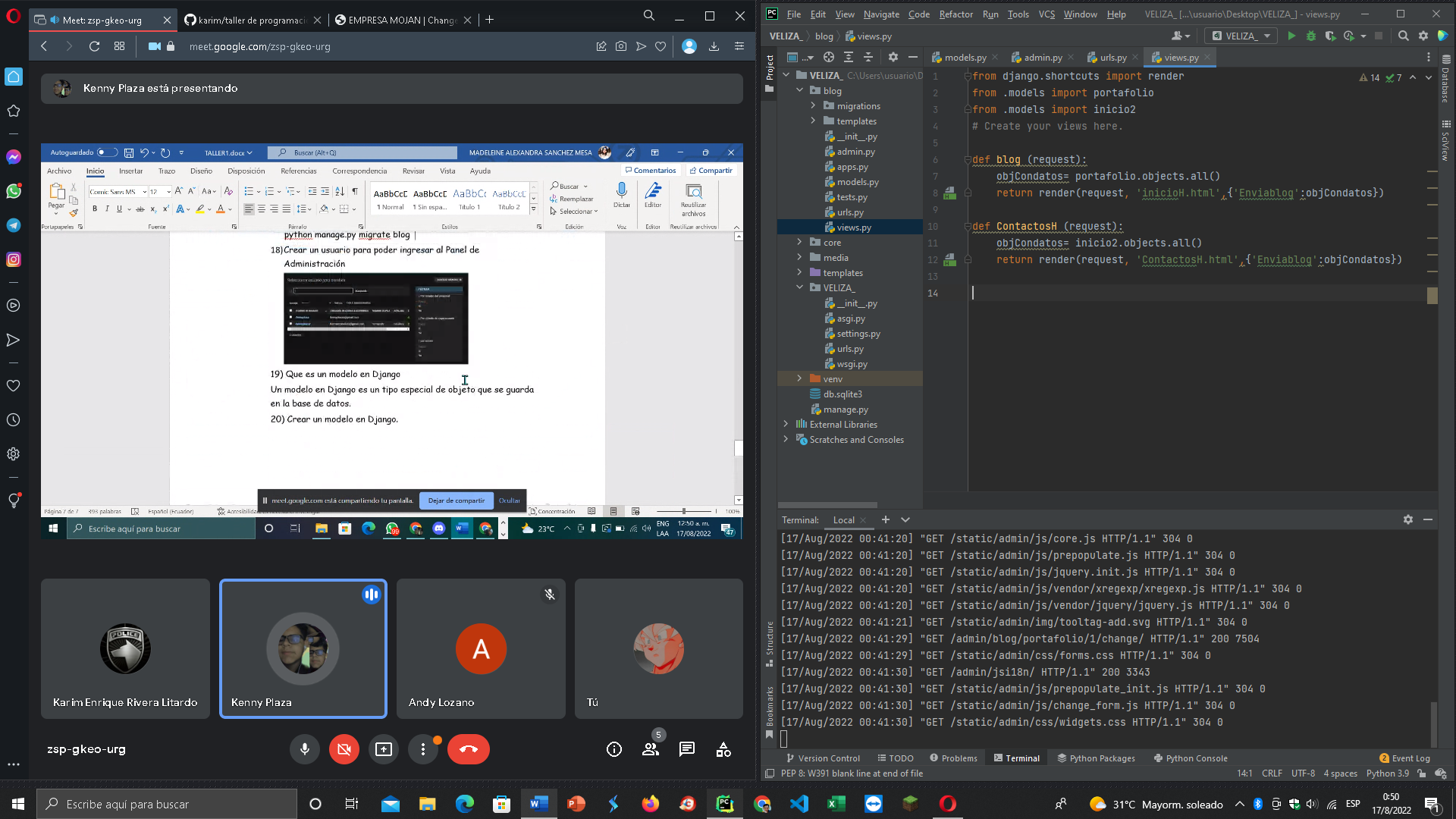


1. **Ingresar información al modelo por el Panel de Administración**

****

Guardamos información añadiendo una imagen en nuestro panel de administración creado con anterioridad

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



Todo lo consultado se registra en mi views

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

**MI HTML HIJO**

{% extends 'principalH.html' %}  
{% load static %}  
{% block titulo %}  
 3H  
{% endblock %}  
{% block contenidohija %}  
<h1>Gracias<h1>  
  
 {% for presenta in Enviablog %}  
  
 <h1>{{ presenta.titulo }}</h1><br><br>  
 <h1>{{ presenta.contenido }} </h1><br>  
 <h1>{{ presenta.fechacreacion }} </h1>  
 <h1><img src="{{presenta.image.url}}"></h1>  
  
{% endfor %}  
  
{% endblock %}

**MI MODELO**

