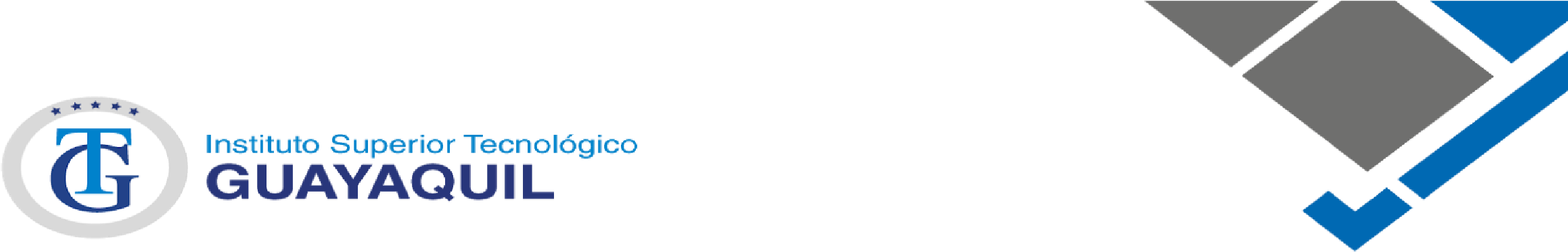
**CARRERA TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE**



**UNIDAD #3**

**ASIGNATURA:** PROGRAMACION VISUAL  **FECHA**:16/.AGOSTO/22 **TEMA**: GIT-GITHUB

**DOCENTE**: ING. CARLOS PAZMIÑO

**NOMBRE ESTUDIANTE:BRYAN ALEXI CANDO AGUAYO**

**PARALELO: 3 …**

**TRABAJO AUTÓNOMO**

**TAREA # 1**

Realizar la carga de un manual de Django a su repositorio Github y enviarme su cuenta al classrom para poder calificar.

El manual de Django debe realizarse de la siguiente manera:

1) En Word, Power point, donde se encuentre los logos del instituto

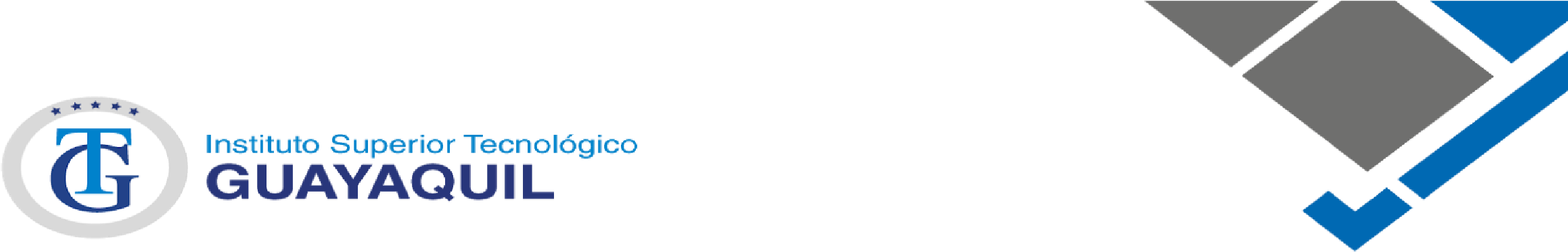
tecnológico superior Guayaquil.

2) En la primera hoja debe tener el nombre de la carrera, estudiante, curso,

nombre del docente y foto del estudiante.

3) En la siguiente hoja debe tener los temas y el número de páginas de donde se encuentra el tema desarrollado.

4) 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.

Flamework.- Es un conjunto de código desarrollado por

uno o muchos programadores para hacer más fácil la

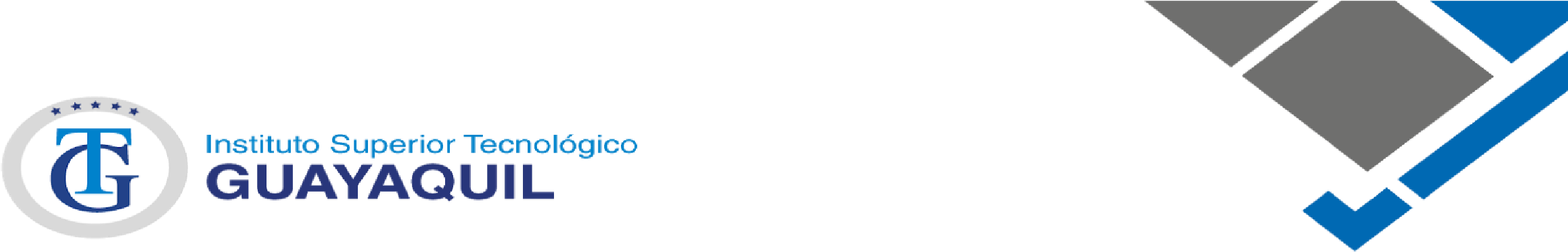
programación.

1. ¿Qué es la máquina virtual en Django?

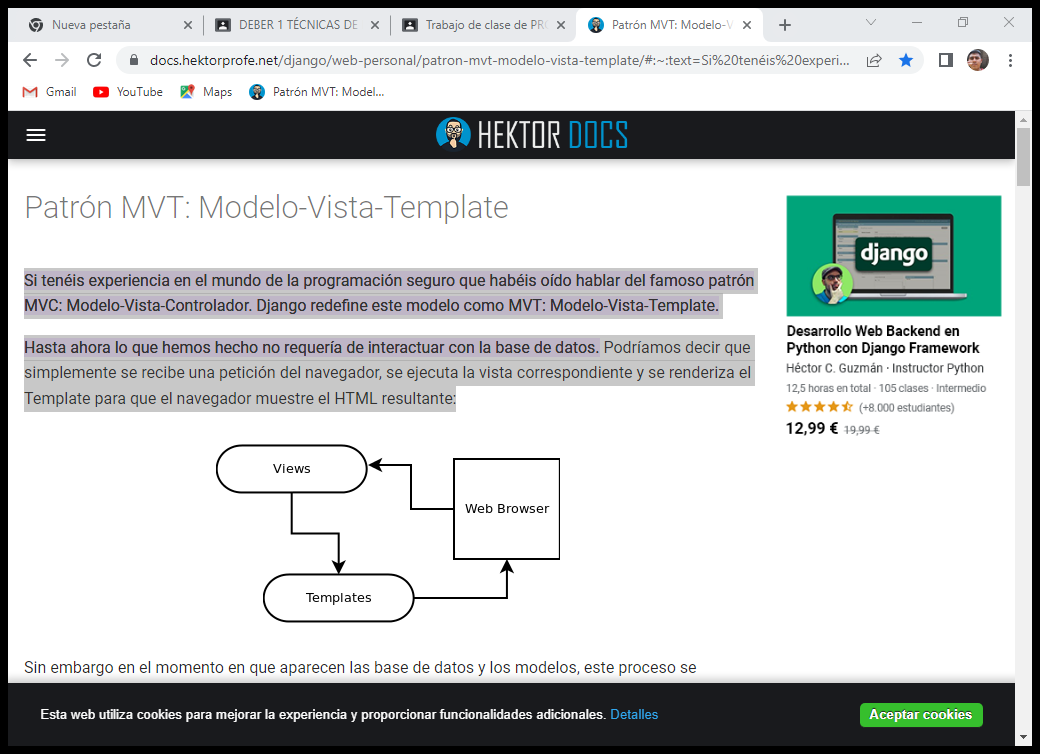
Un entorno virtual es un **entorno de Python aislado creado para un proyecto**. Este entorno tiene su propio intérprete de Python, bibliotecas y paquetes, lo que significa que el intérprete de Python y las dependencias instaladas pertenecen solo a este proyecto.



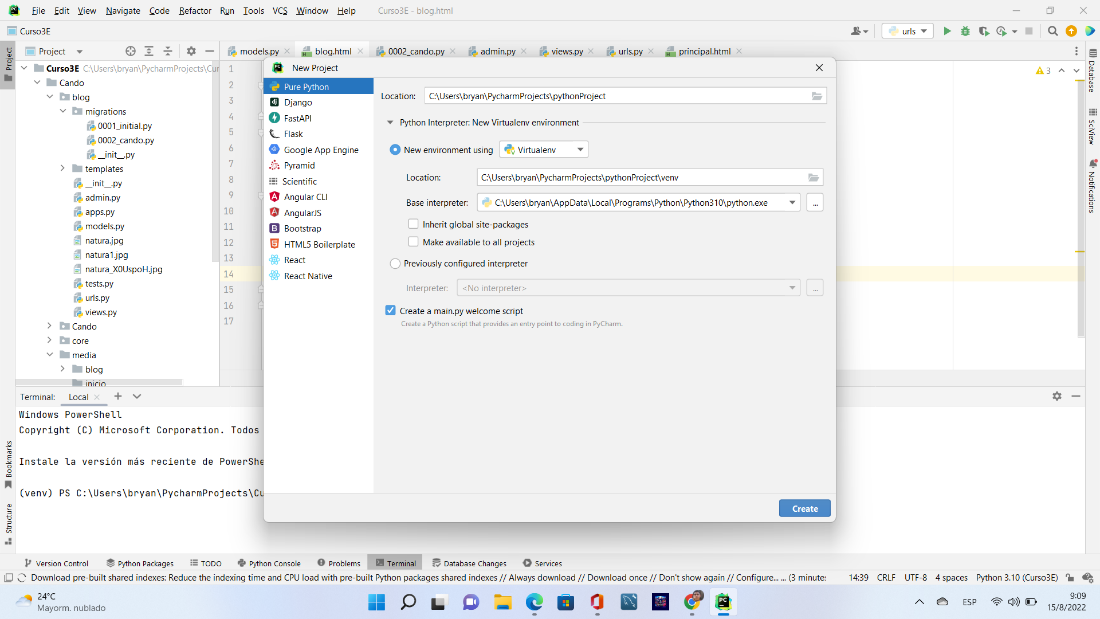
1. ¿ Qué es MVT en Django. ?



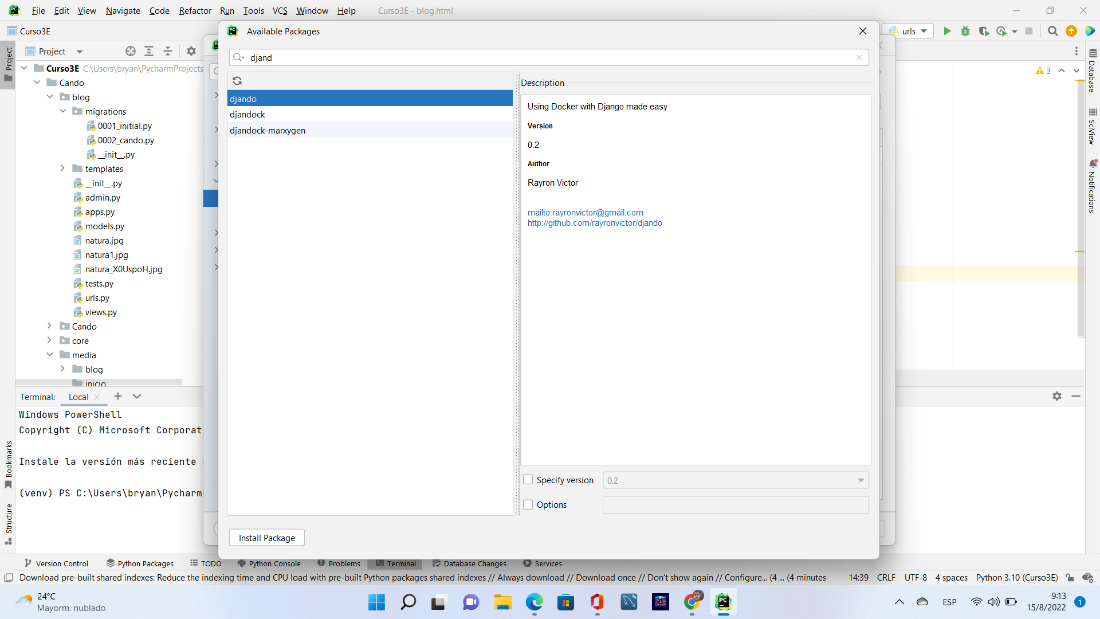
En el mundo de la programación el famoso patrón MVC: Modelo-Vista-Controlador. Django redefine este modelo como MVT: Modelo-Vista-Template.

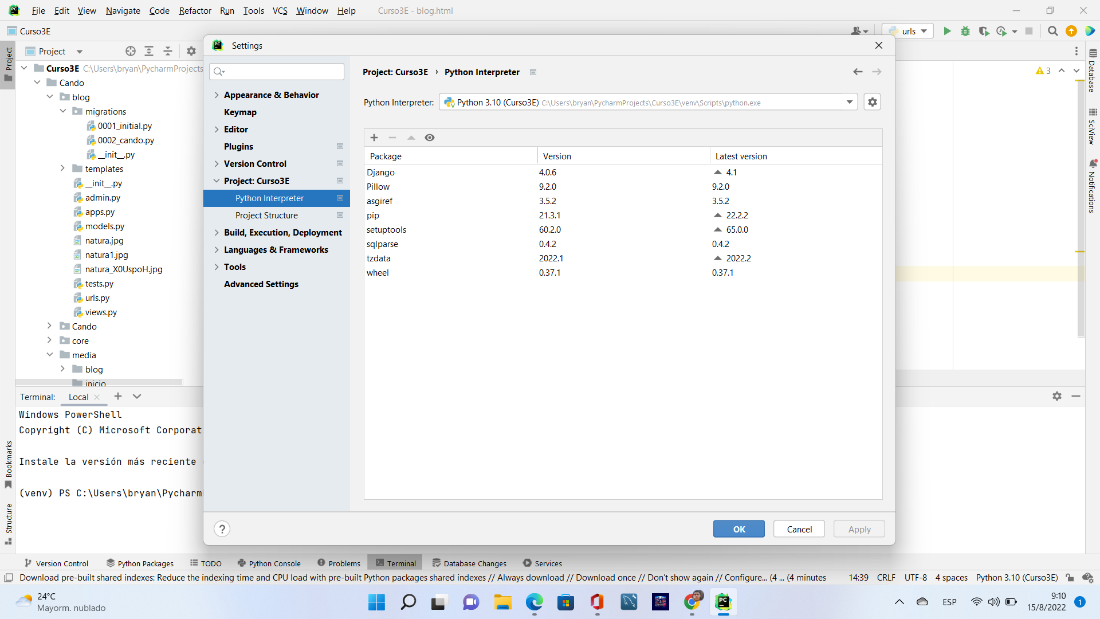
Hasta ahora lo que hemos hecho no requería de interactuar con la base de datos. Podríamos decir que simplemente se recibe una petición del navegador, se ejecuta la vista correspondiente y se renderiza el Template para que el navegador muestre el HTML resultante:

4) Crear un proyecto con la máquina virtual.



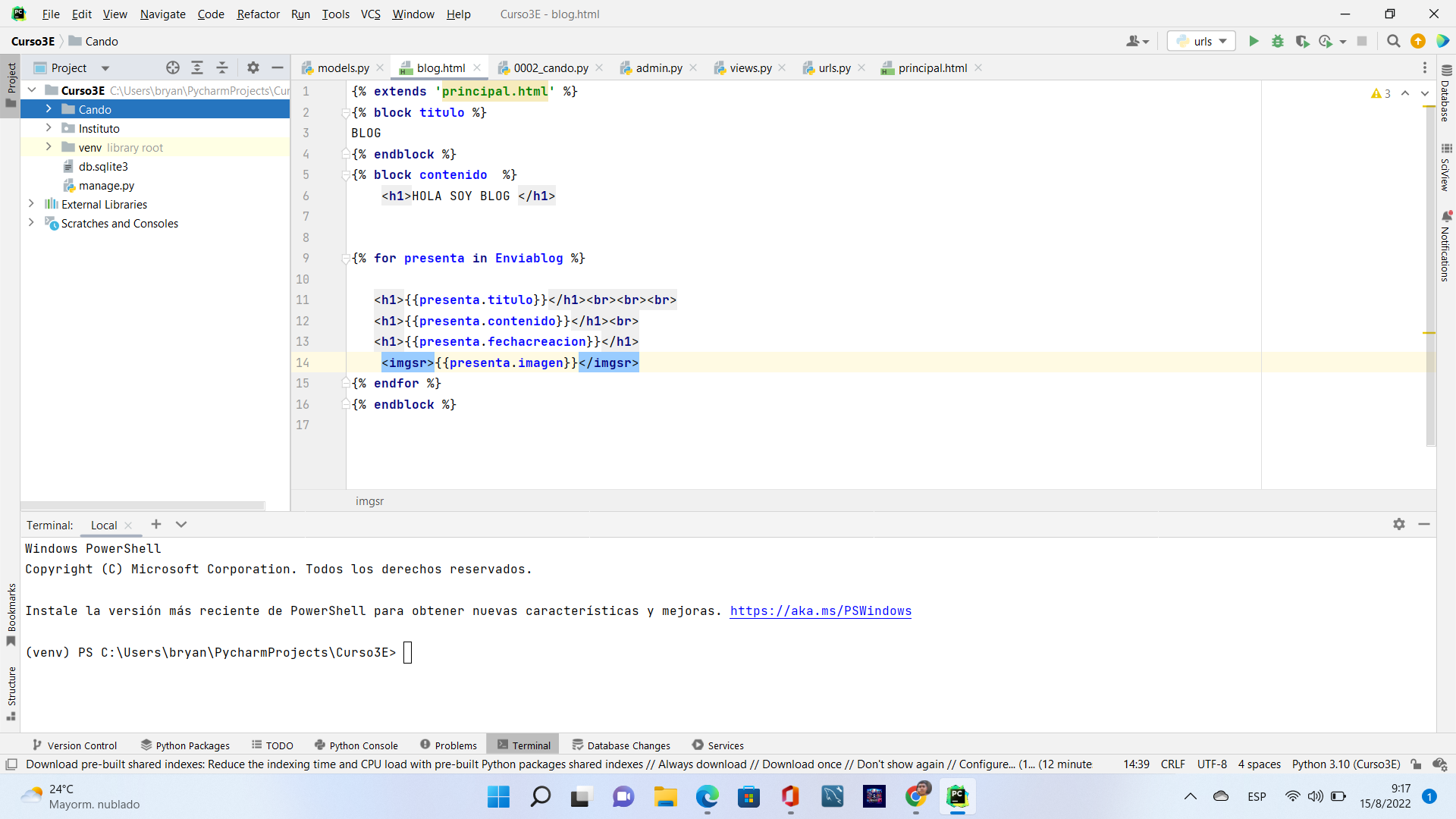
5) Descargar los instaladores de Django al proyecto





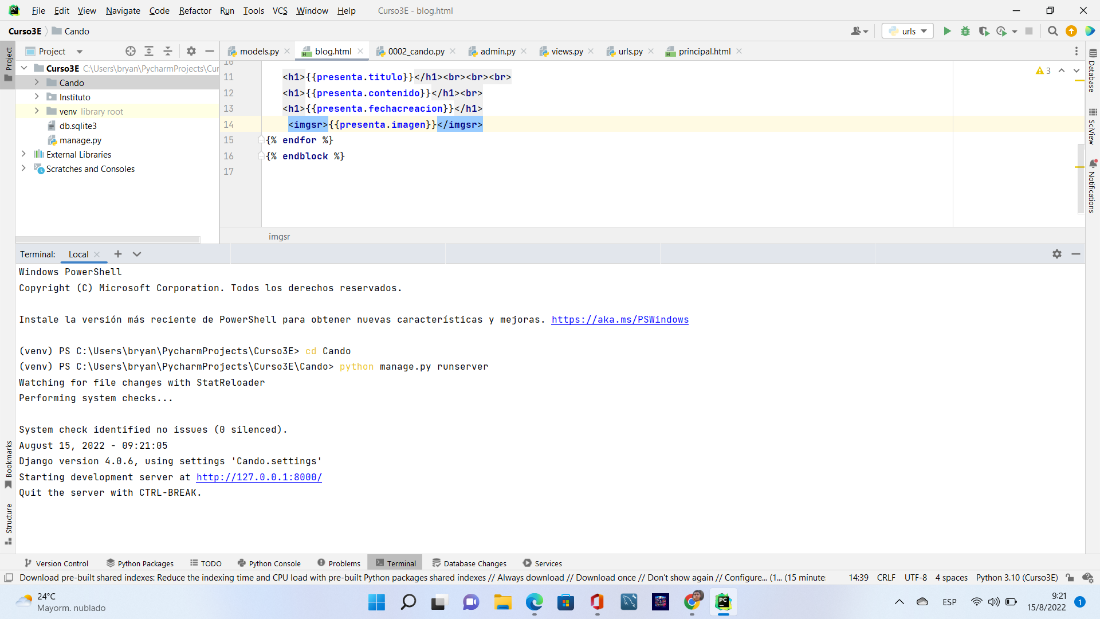
6) Crear un proyecto para programar en Django

Utilizamos este comando en el terminal para crear nuestro proyecto en el paychar , le podemos poner el nombre que deseemos yo le puse el siguiente django-admin startproject Cando



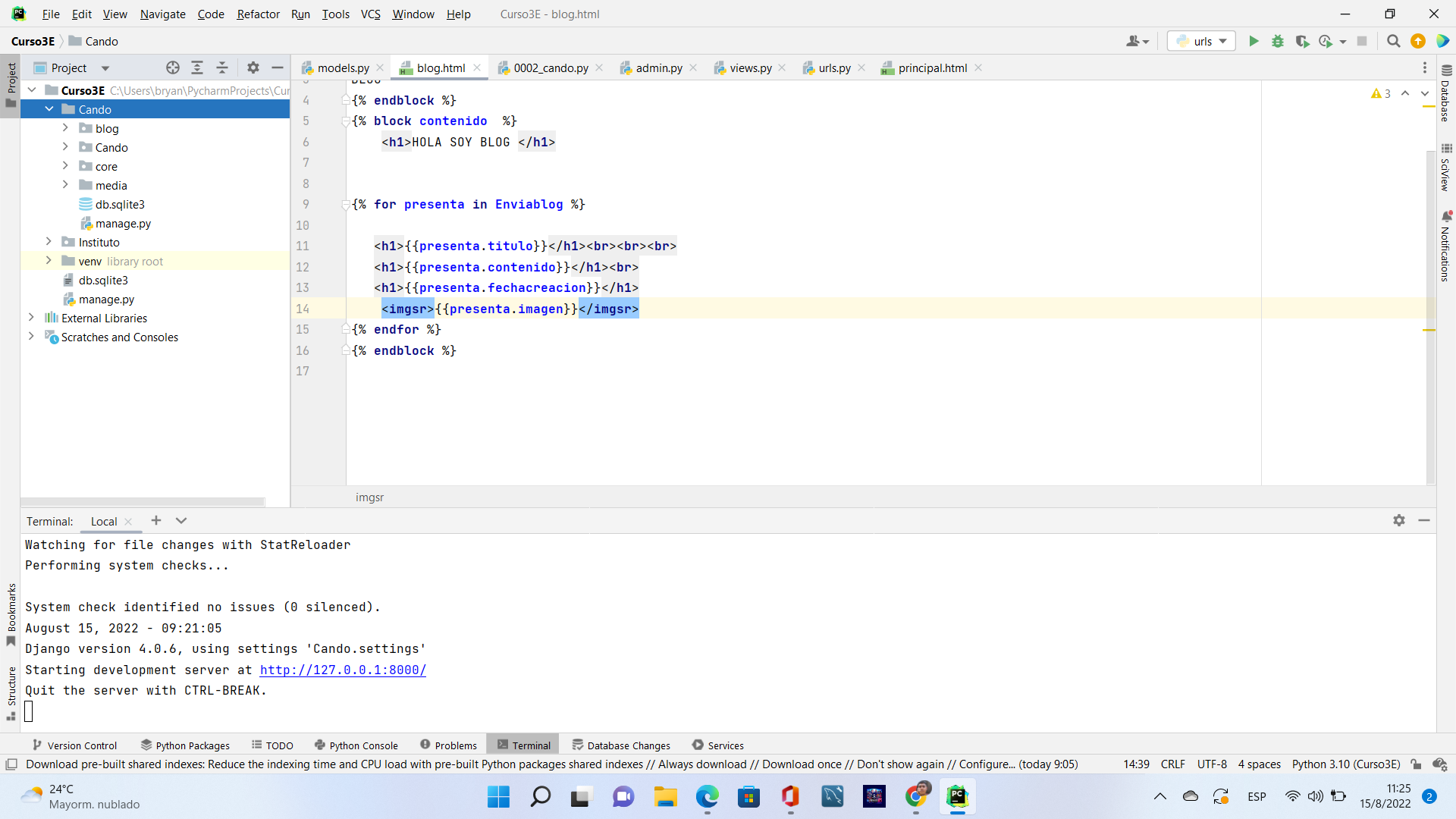
7) Ejecutar el proyecto y el mensaje de felicitaciones.

Ejecutamos este comando en el terminal para levantar el servidor en mi caso se levanta el servidor poniendo python manage.py runserver



8) Crear una Apps core.

Utilizamos este comando para crear una aplicación python manage.py startapp core.



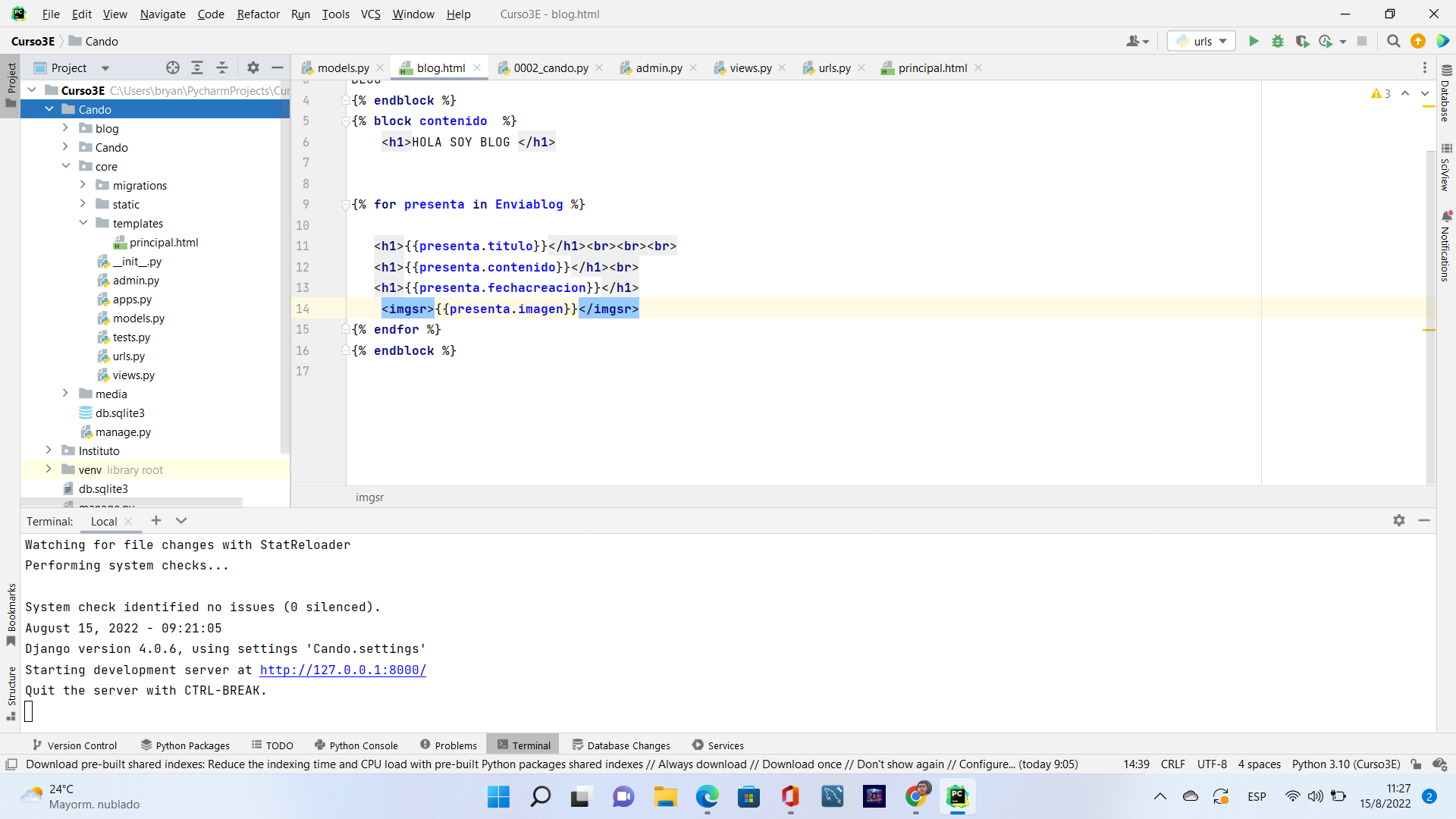
9) ¿Qué es la Carpeta Templates?

En la carpeta Templates, **las subcarpetas contienen las plantillas que se muestran en el cuadro de diálogo Crear nuevo archivo**(como una página HTML). Puede crear y guardar plantillas personalizadas en la carpeta Templates. Contiene plantillas para la creación de archivos. Se puede guardar el archivo activo como una plantilla.

10) ¿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



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

Las hojas de estilo en cascada (**CSS**, cascading style sheets) permiten crear páginas web atractivas. El archivo CSS se llama **stylesheet**.

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

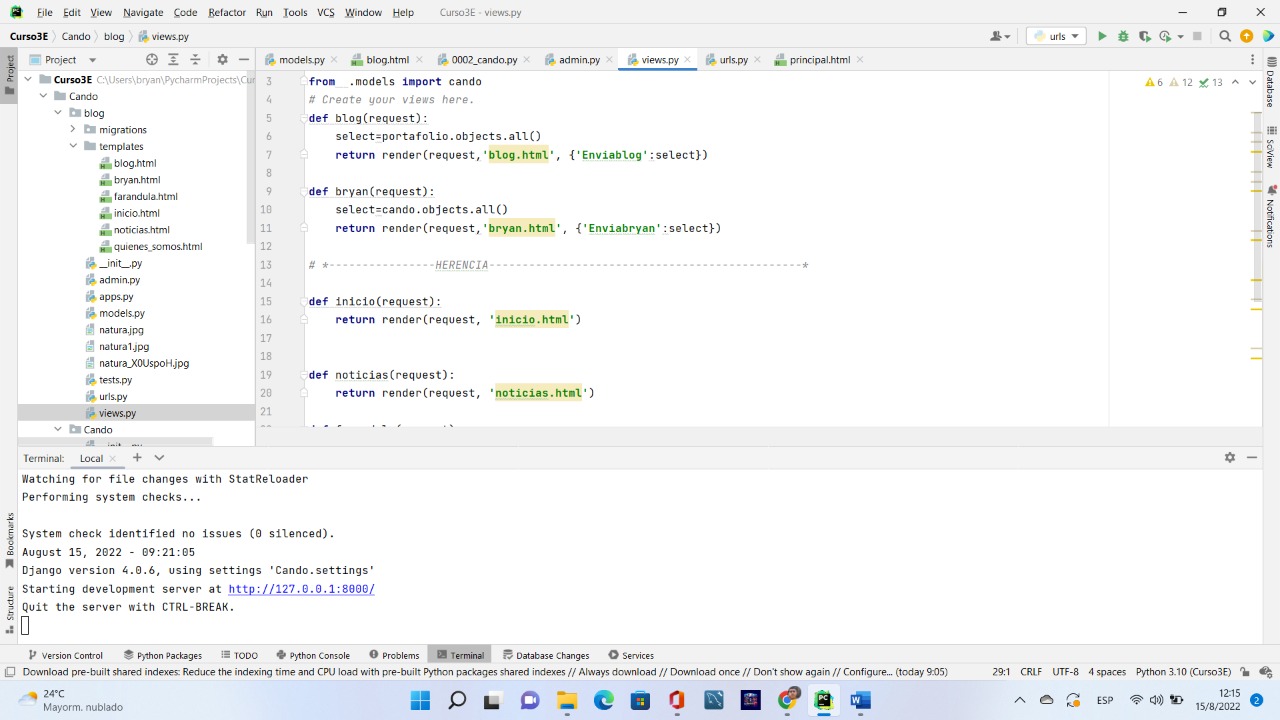
archivo base html.

Si usas **{% extends %}** en la plantilla, esta debe ser la primer etiqueta de esa plantilla. En otro caso, la **herencia** no funcionará. Generalmente, cuanto más etiquetas **{%** block **%}** tengas en tus plantillas, mejor.

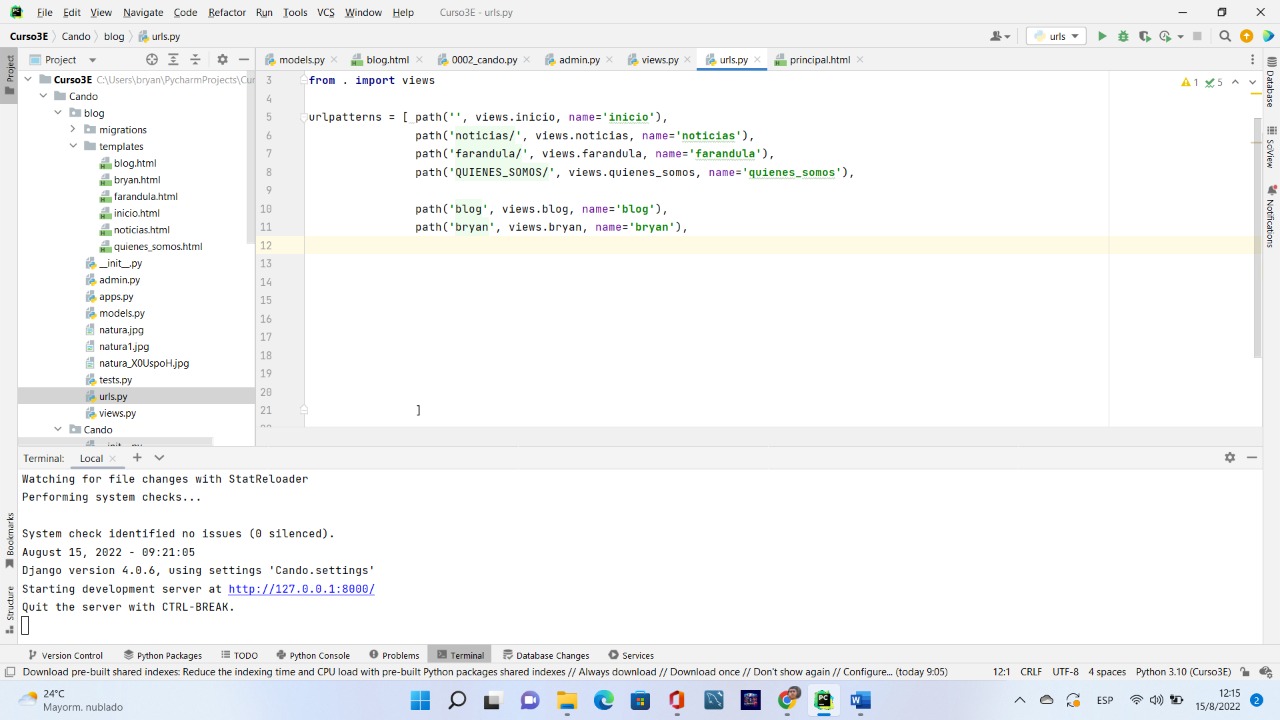
En esencia, la herencia de plantillas te deja construir una plantilla base "esqueleto" que contenga todas

las partes comunes de tu sitio y definir "bloques" que los hijos puedan sobrescribir.

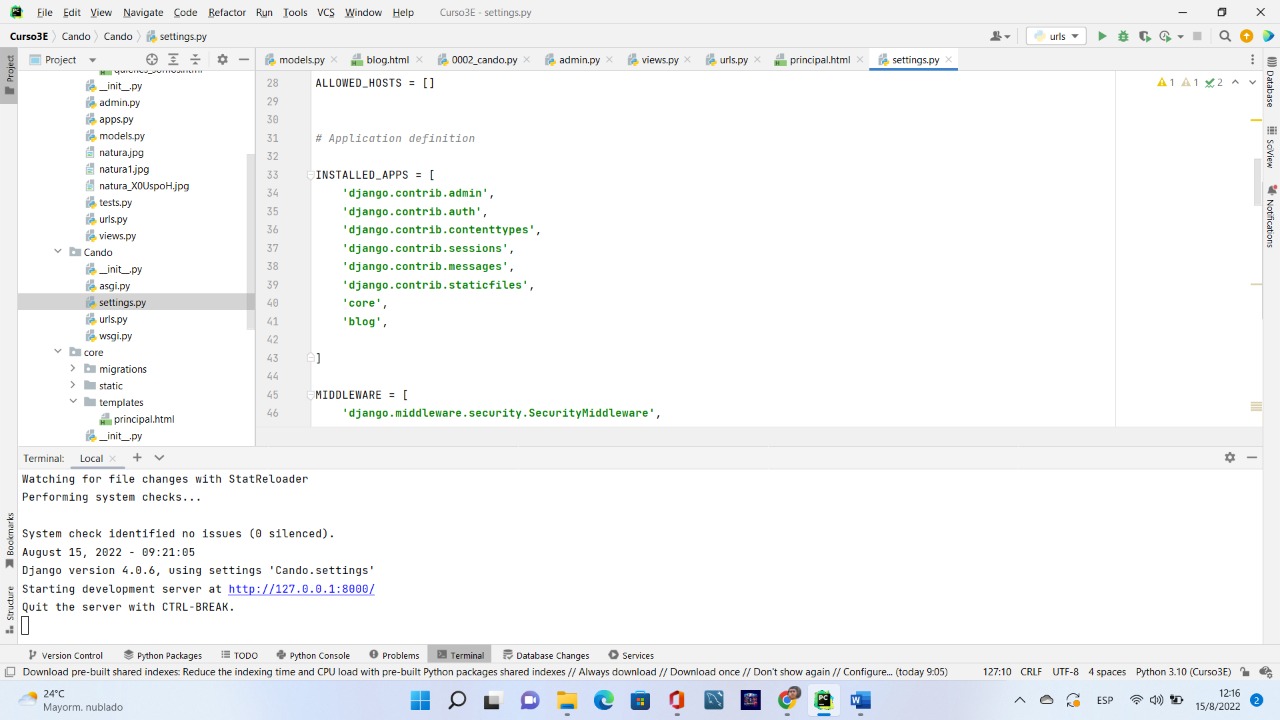
14)Crear un view que llame al html hijo



15)Crear la urls que llame al views.



16)Integrar la aplicación APPS core al proyecto principal

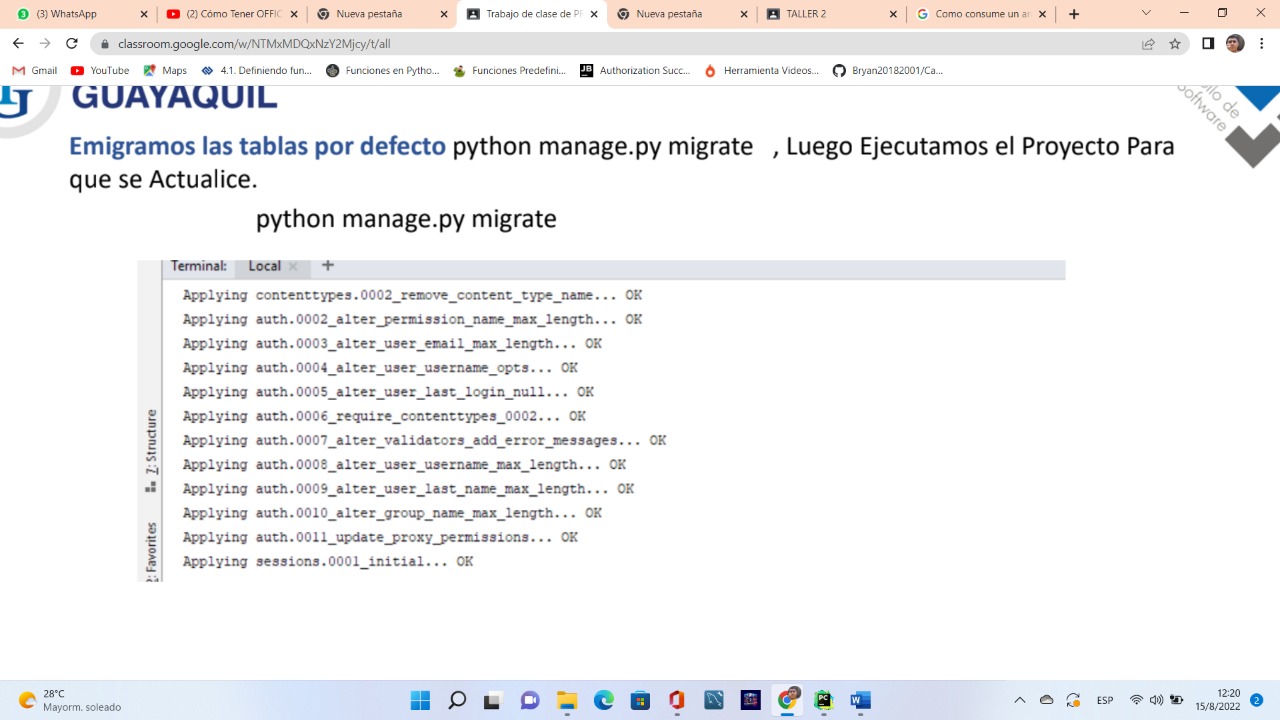


17)Crear las tablas del sistema de usuarios para utilizar el panel de

administración.

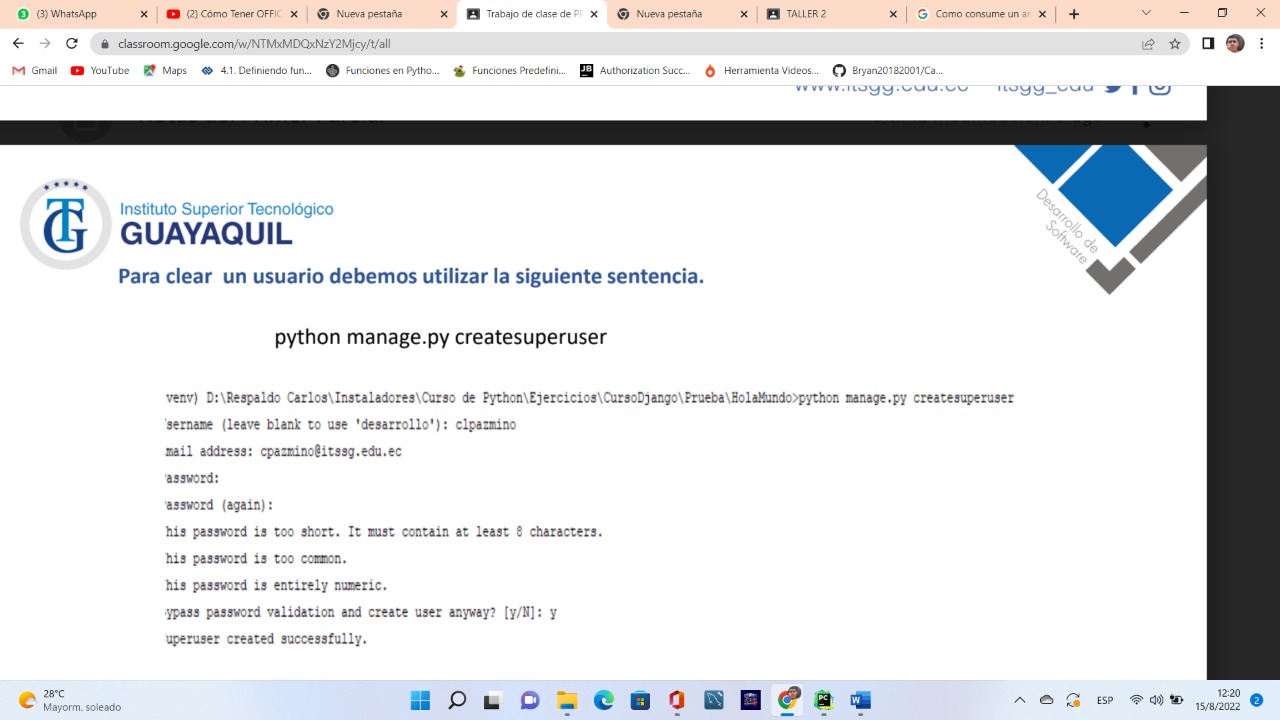
Emigramos las tablas por defecto python manage.py migrate , Luego Ejecutamos el Proyecto Para

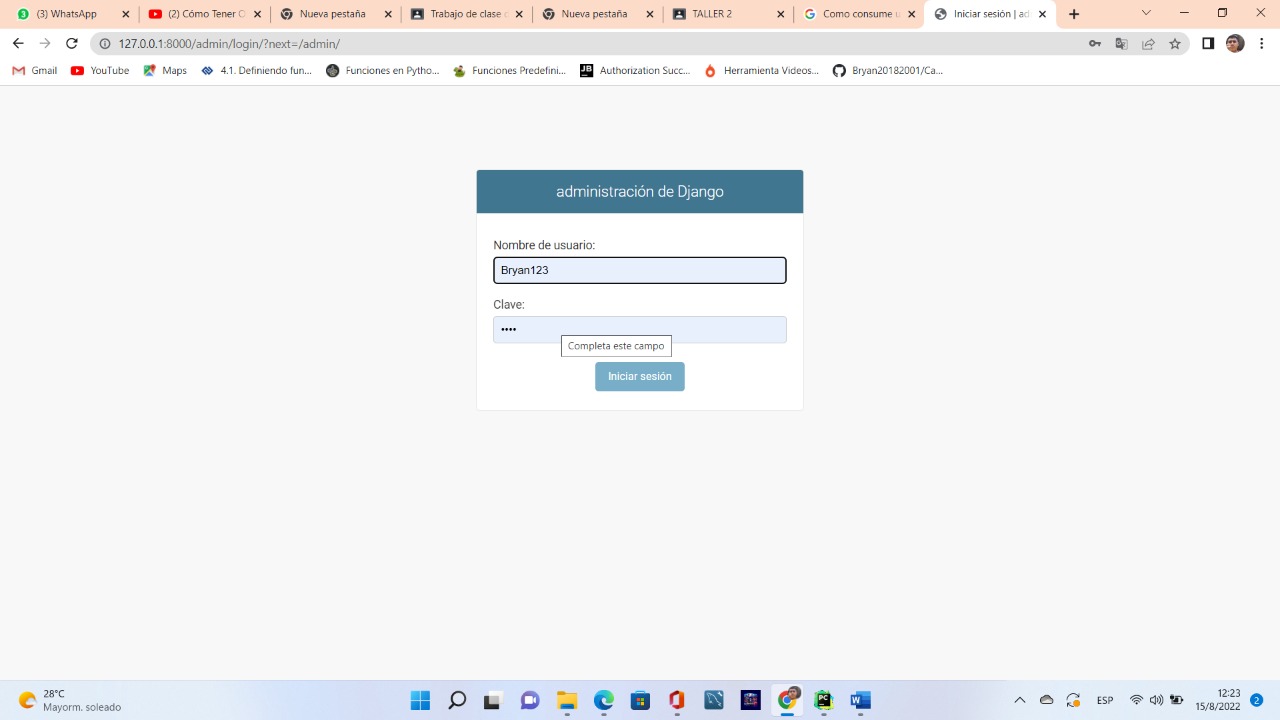
que se Actualice.

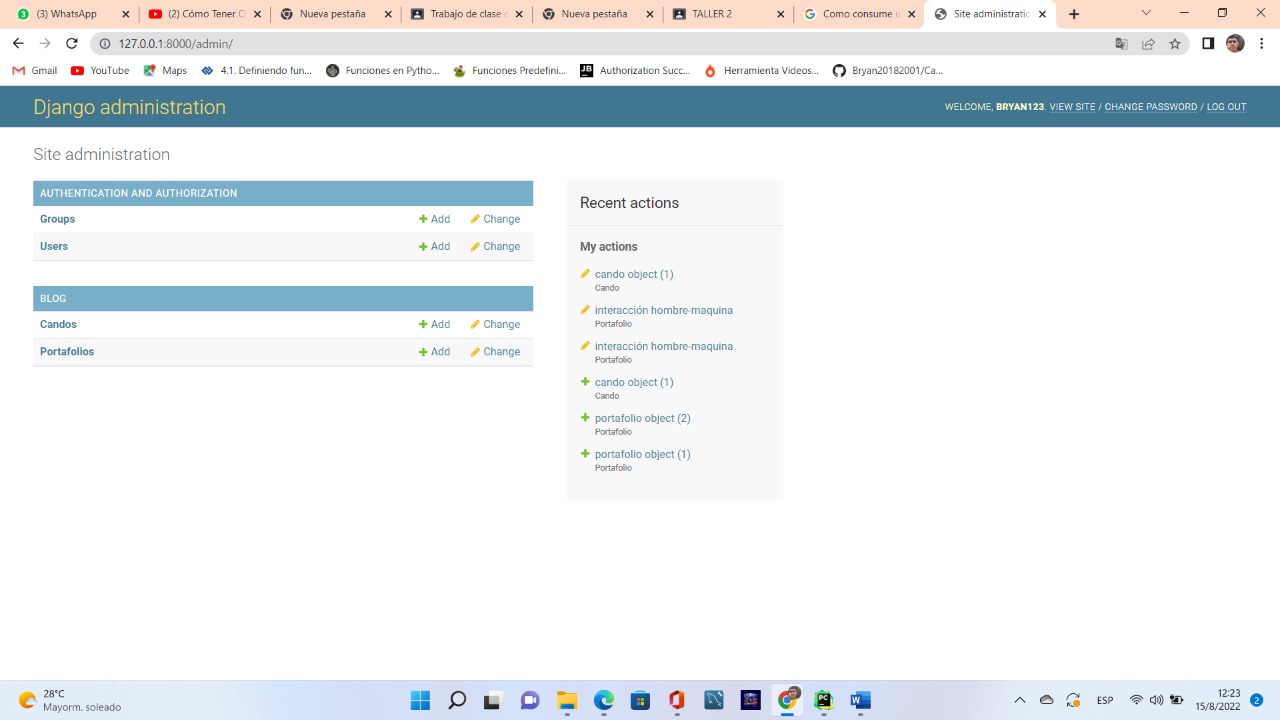


18)Crear un usuario para poder ingresar al Panel de Administración

Creamos el usuario mediante el siguiente comando: paython manage.py createsuperuser para asi poder ingresar al panel de administración



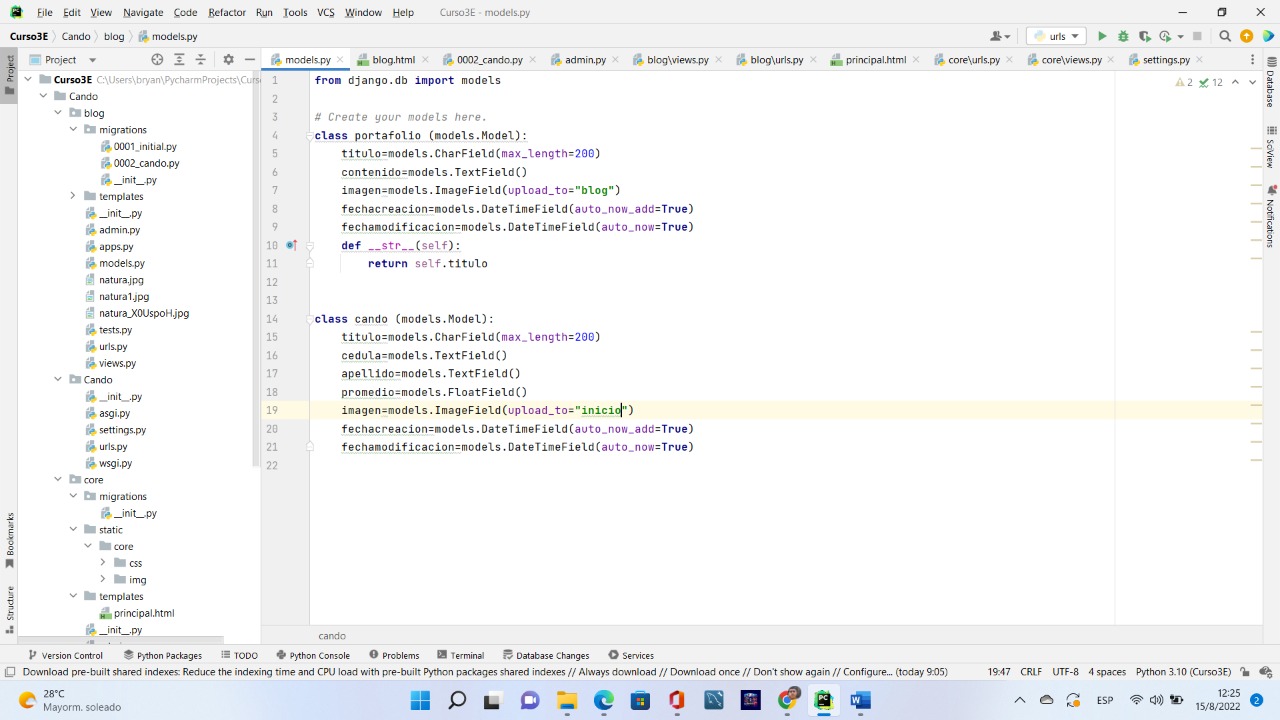




19)Que es un modelo en Django

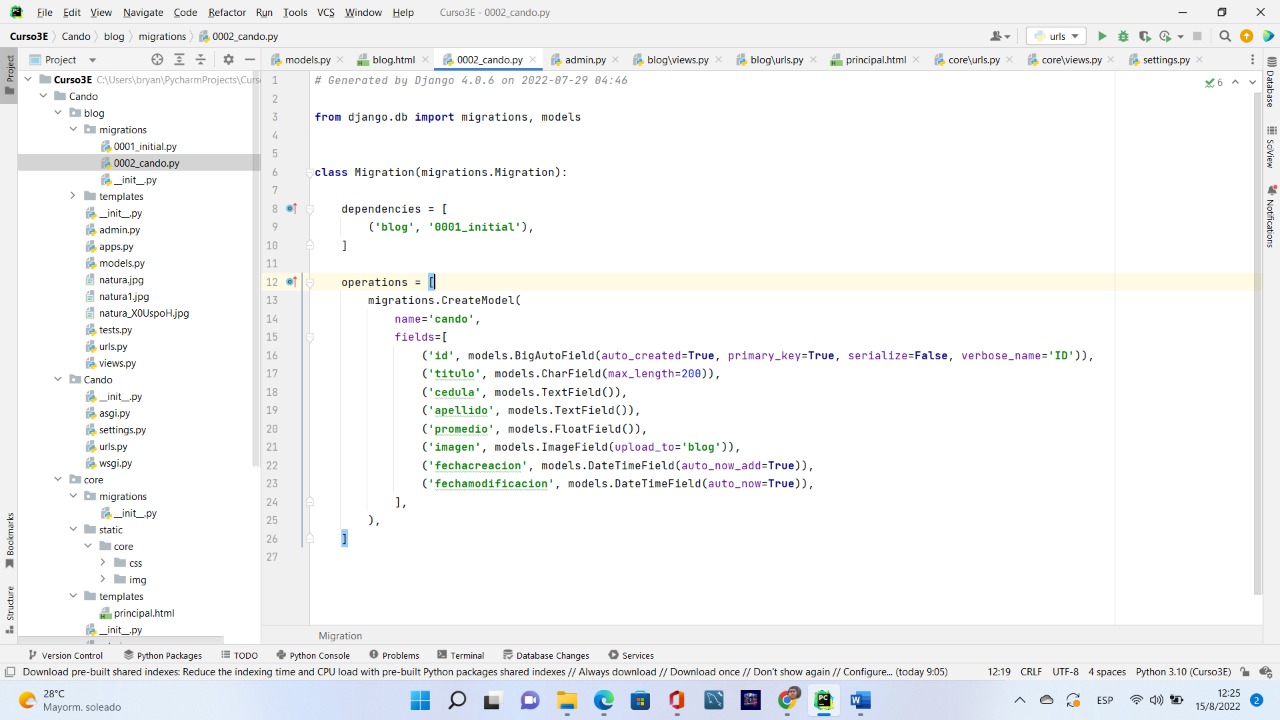
Un modelo en Django **es un tipo especial de objeto que se guarda en la base de datos** . Una base de datos es una colección de datos. Es un lugar en el cual almacenarás la información sobre usuarios, tus entradas de blog, etc. Utilizaremos una base de datos SQLite para almacenar nuestros datos.

20)Crear un modelo en Django.

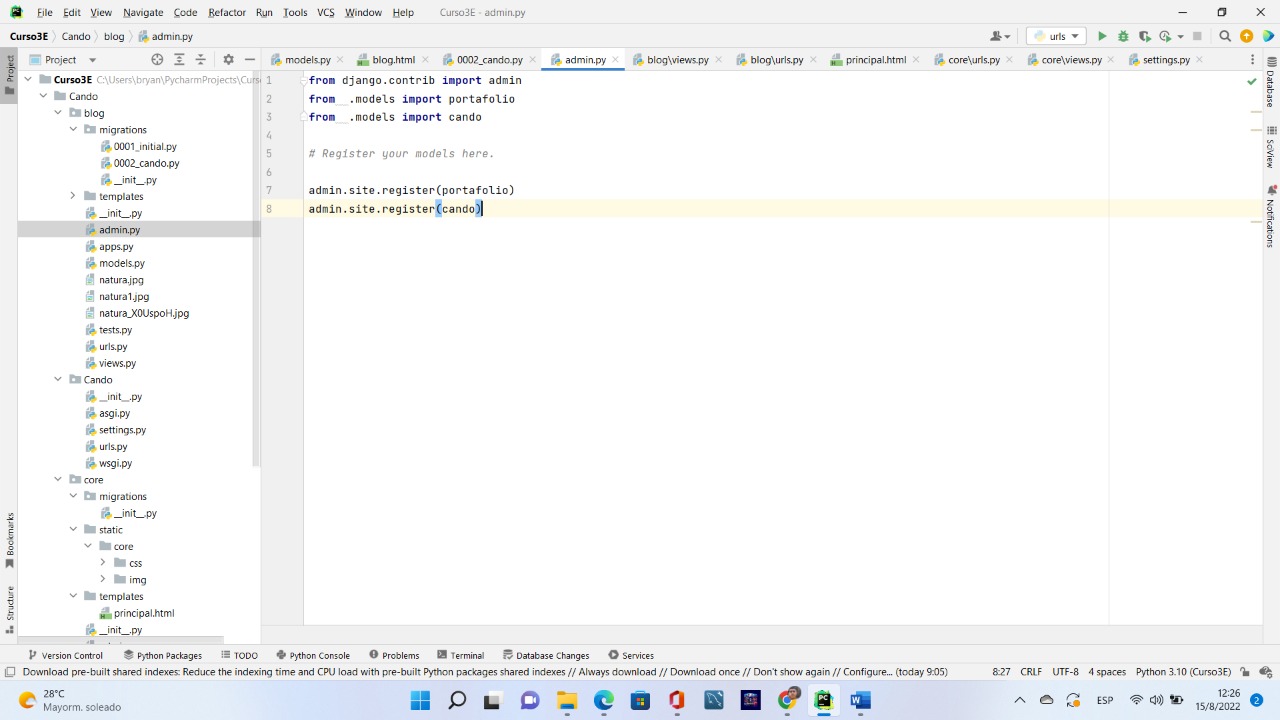


21)Migrar el Modelo a la base del Panel de Administración.

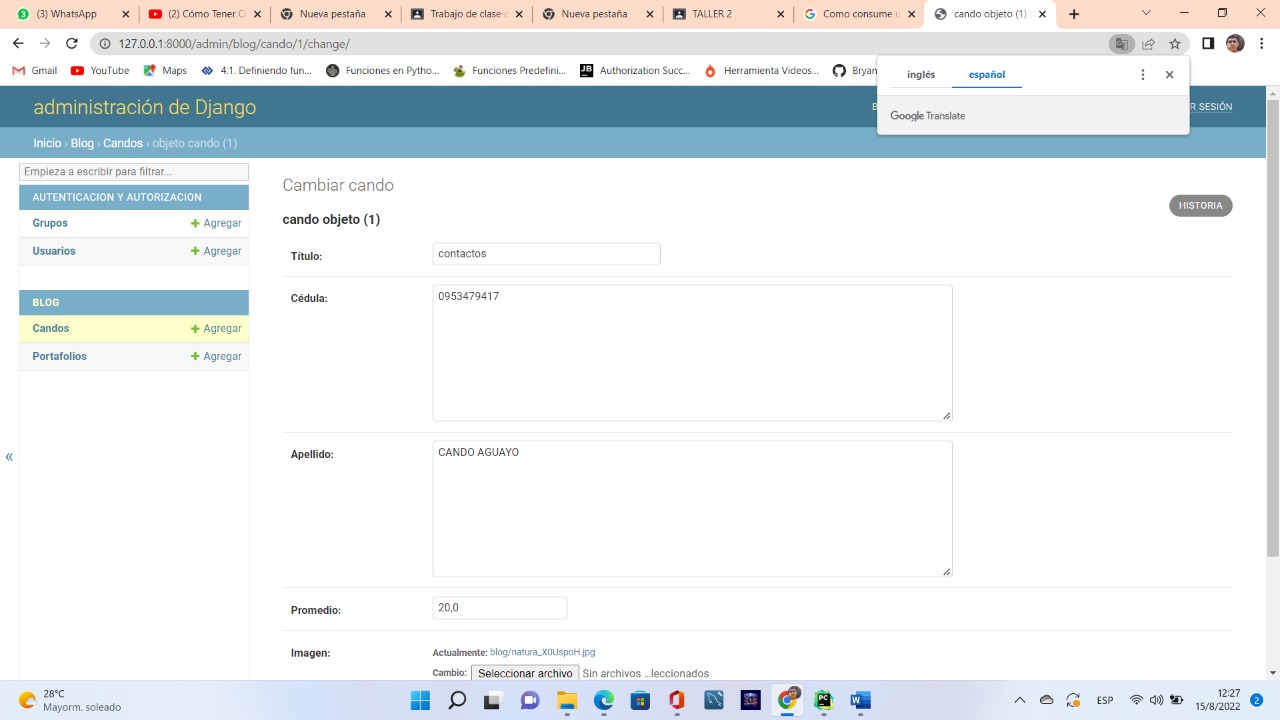
Migramos el modelo mediante el siguiente comando: paython manage.py migrations Cando



22)Integrar el Modelo al Panel de Administración.

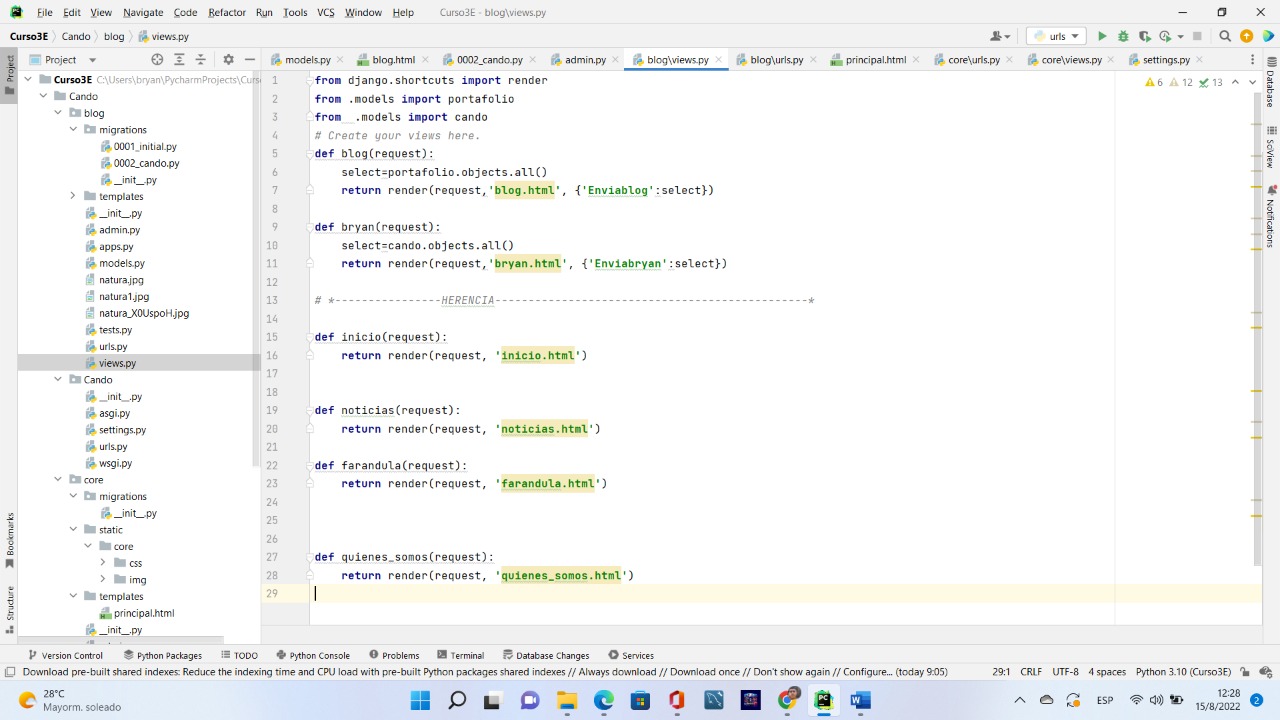


23)Ingresar información al modelo por el Panel de Administración.



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.

