

Taller

**APELLIDOS**:

Pinto Carpio

**NOMBRES:**

Emily Dayana

**CARRERA:**

Tecnología en desarrollo de software

Foto

DATOS DEL ALUMNO



**CURSO:**

3

**PARALELO:**

“H”

**MATERIA:**

Programación visual

**DOCENTE DE LA MATERIA:**

ING. Carlos Luis Pazmiño Palma

Contenido

[¿QUE ES DJANGO? 3](#_Toc111701597)

[¿QUE ES LA MAQUINA VIRTUAL EN DJANGO? 4](#_Toc111701598)

[¿QUE ES MVT EN DJANGO? 5](#_Toc111701599)

[¿CREAR UN PROYECTO CON LA MAQUINA VIRTUAL? 6](#_Toc111701600)

[¿DESCARGAR LOS INSTALADORES DE DJANGO AL PROYECTO? 7](#_Toc111701601)

[¿CREAR UN PROYECTO PARA PROGRAMAR EN DJANGO? 9](#_Toc111701602)

[¿EJECUTAR EL PROYECTO Y EL MENSAJE DE FELICITACIONES? 10](#_Toc111701603)

[¿CREAR UNA APPS CORE? 11](#_Toc111701604)

[¿QUE ES LA CARPETA TEMPLATES? 12](#_Toc111701605)

[¿QUE ES LA CARPETA STATIC? 12](#_Toc111701606)

[¿CREAR UN ARCHIVO BASE HTML EN LA APPS CORE 13](#_Toc111701607)

[¿COMO SE LLAMAN LOS CSS DESDE EL ARCHIVO BASE HTML? 14](#_Toc111701608)

[¿COMO CONSUME UN ARCHIVO HIJO HTML AL UTILIZAR LA HERENCIA DEL ARCHIVO BASE HTML? 14](#_Toc111701609)

[INTEGRAR LA APLICACIÓN APPS CORE AL PROYECTO PRINCIPAL 15](#_Toc111701610)

[CREAR LAS TABLAS DEL SISTEMA DE USUARIOS PARA UTILIZAR EL PANEL DE ADMINISTRACION 16](#_Toc111701611)

[¿CREAR UN USUARIO PARA PODER INGRESAR AL PANEL DE ADMINISTRACION? 17](#_Toc111701612)

[¿QUE ES UN MODELO EN DJANGO? 18](#_Toc111701613)

[¿CREAR UN MODELO EN DJANGO? 18](#_Toc111701614)

[MIGRAR EL MODELO A LA BASE DEL PANEL DE ADMINISTRACION 19](#_Toc111701615)

[INTEGRAR EL MODELO AL PANEL DE ADMINISTRACION 19](#_Toc111701616)

[INGRESAR INFORMACION AL MODELO POR EL PANEL DE ADMINISTRACION 20](#_Toc111701617)

[REALIZAR LA CONSULTA DE TODO LO INGRESADO EN EL MODELO DESDE EL VIEWS 20](#_Toc111701618)

[MOSTRAR LOS DATOS GUARDADOS EN EL MODELO HTML HIJO 21](#_Toc111701619)

# ¿QUE ES DJANGO?



**El  framework  Django es uno de los más populares para el desarrollo de aplicaciones web usando el lenguaje Python**

El  *framework*  Django es uno de los más populares para el desarrollo de aplicaciones web usando el lenguaje Python.

Al ser un *framework*, tiene ventajas como su uso de patrones de arquitectura de *software*, su flexibilidad, la alta calidad de su documentación y su amplia comunidad de desarrolladores.

Django además tiene características particulares que lo hacen una muy buena opción para desarrollar nuestras aplicaciones web. Entre ellas:

* Su ORM (*object relational mapping* o mapeadores de objetos relacionales), que nos permite interactuar con una base de datos de manera muy práctica.
* Su interfaz de administración, que nos otorga un lugar para manejar los datos de nuestra aplicación con muy poco trabajo adicional.
* La amplia selección de módulos y librerías, tanto oficiales como provistas por la comunidad, que hace posible incorporar todo tipo de funcionalidades, como proveer una API (*application programming interface* o interfaz de programación de aplicaciones).

# ¿QUE ES LA MAQUINA VIRTUAL EN DJANGO?

Es a lo que se denomina entorno de desarrollo y esta se hace a través de una instalación de Django en tu computadora local que puedes usar para desarrollar y probar apps Django antes de desplegarlas al entorno de producción.

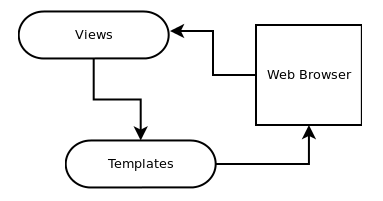
Las principales herramientas que el mismo Django proporciona son un conjunto de scripts de Python para crear y trabajar con proyectos Django, junto con un simple *servidor web de desarrollo* que puedes usar para probar de forma local (es decir en tu computadora, no en un servidor web externo) aplicaciones web Django con el explorador web de tu computadora.

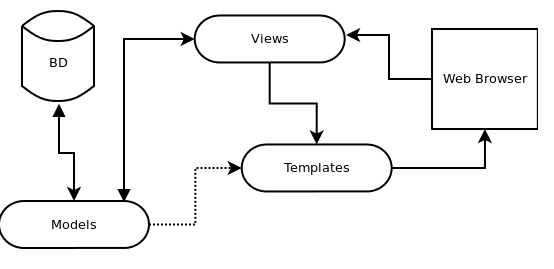
Django es extremadamente flexible en términos de cómo y dónde puede instalarse y configurarse. Django puede ser:

* instalado en diferentes sistemas operativos.
* ser usado con Python 3 y Python 2.
* instalado desde las fuentes, desde el Python Package Index (PyPi) y en muchos casos desde la aplicación de gestión de paquetes de la computadora.
* configurado para usar una de entre varias bases de datos, que pueden también necesitar ser instaladas y configuradas por separado.
* ejecutarse en el entorno Python del sistema principal o dentro de entornos virtuales Python separados.

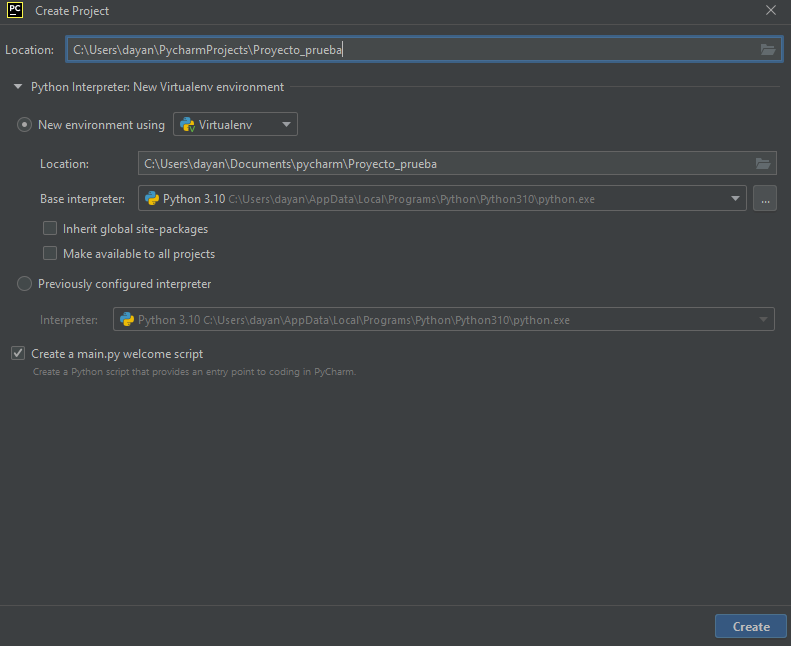
# ¿QUE ES MVT EN DJANGO?

en el mundo de la programación seguro que habéis oído hablar del famoso patrón MVC: Modelo-Vista-Controlador. Django redefine este modelo como MVT: Modelo-Vista-Template.

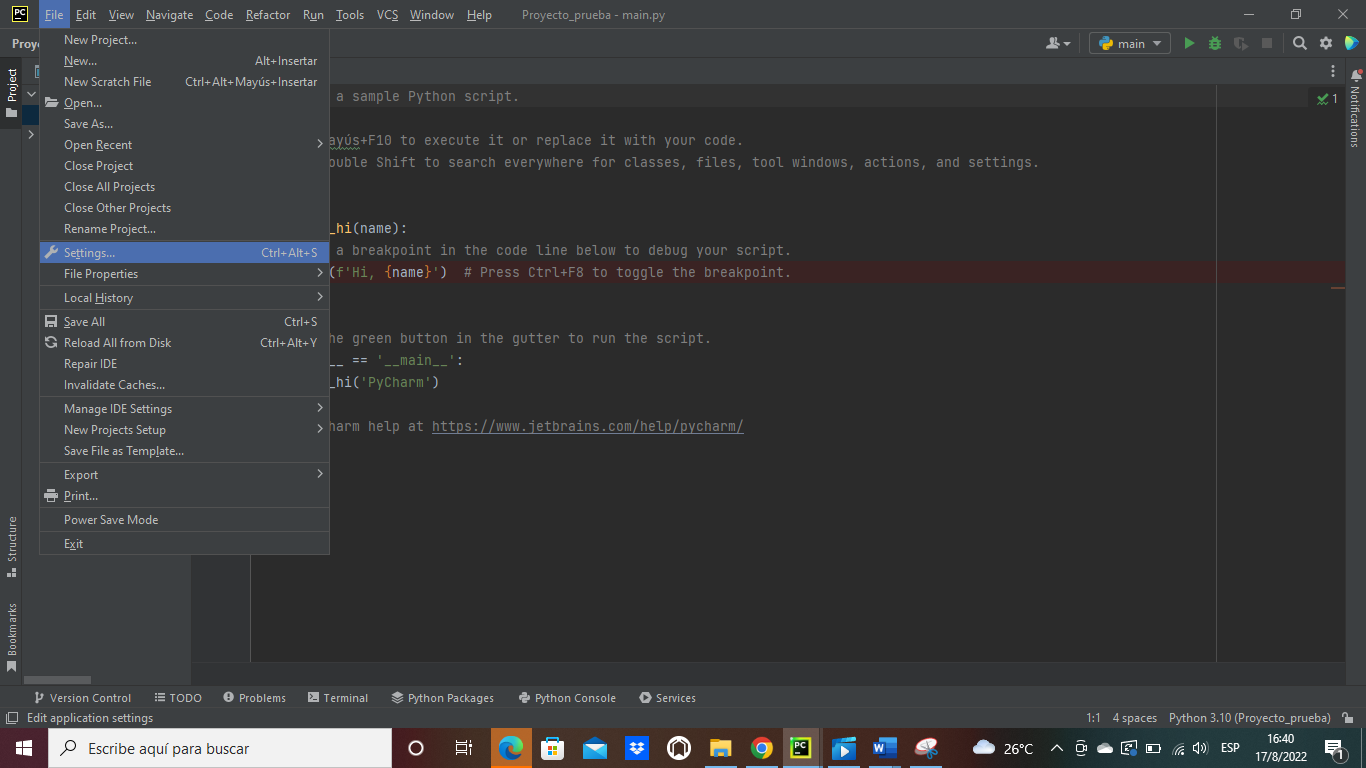
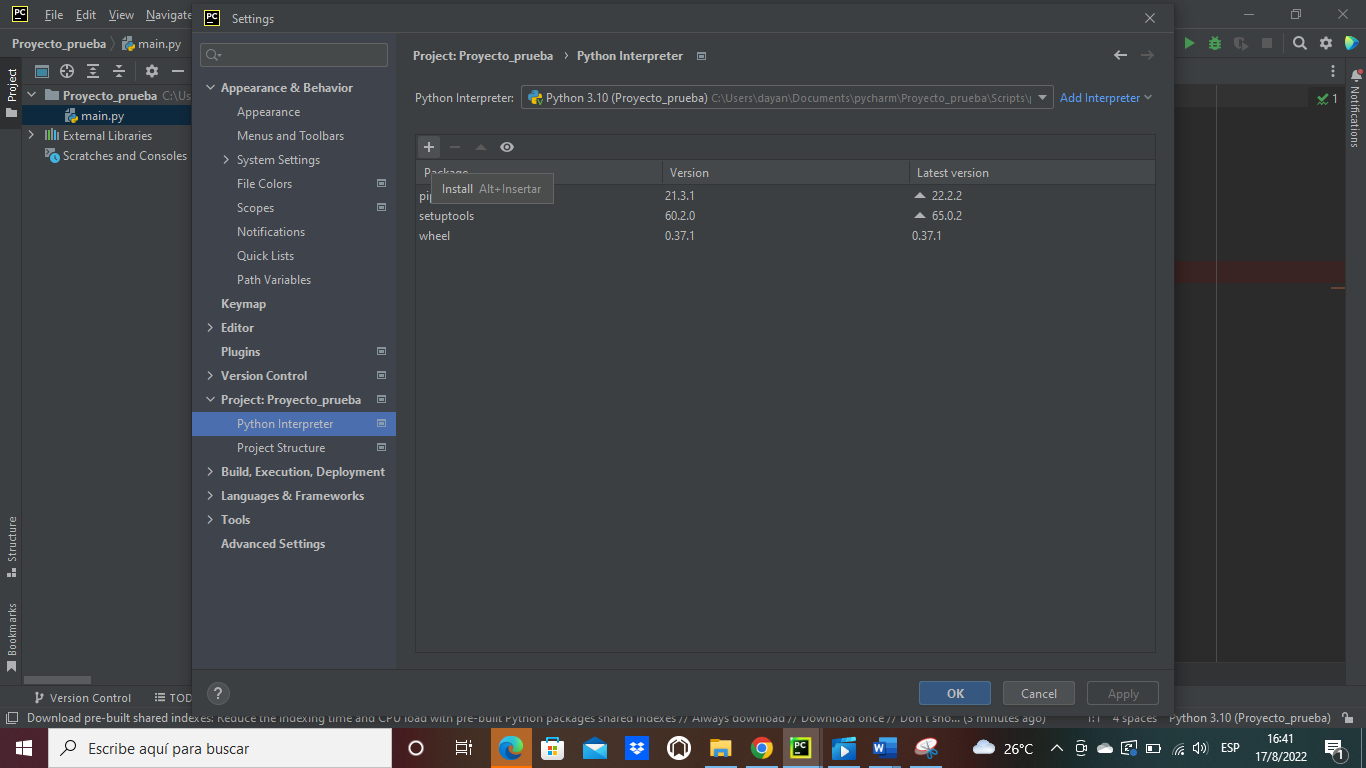
Hasta ahora lo que hemos hecho no requería de interactuar con la base de datos. Podríamos decir que simplemente se recibe una petición del navegador, se ejecuta la vista correspondiente y se renderiza el Template para que el navegador muestre el HTML resultante:

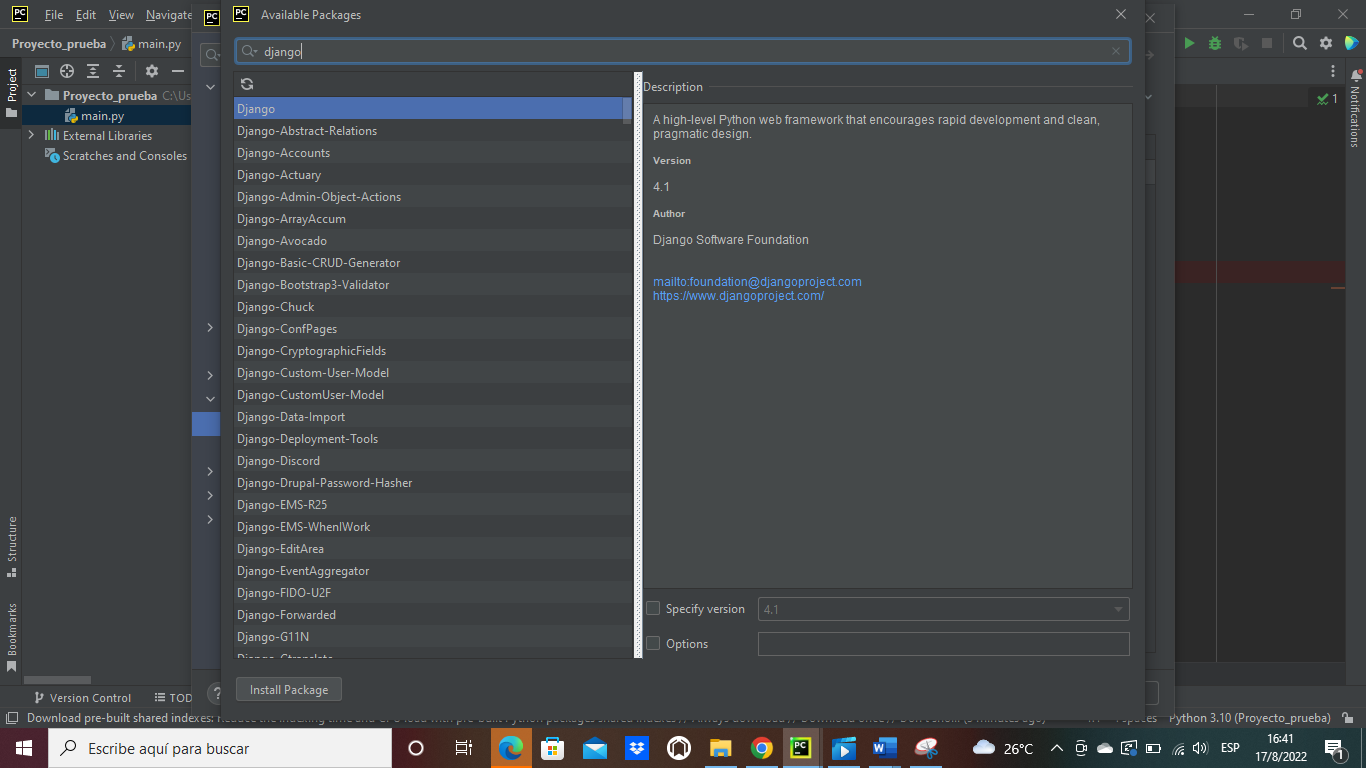
Sin embargo, en el momento en que aparecen las base de datos y los modelos, este proceso se extendiende. Ahora se recibirá la petición, se pasará a la vista, en la vista recuperaremos los datos del modelo correspondiente, y finalmente la renderizaremos el Template pero esta vez integrando los datos dinámicos recuperados del modelo, antes de enviar el HTML final al navegador:

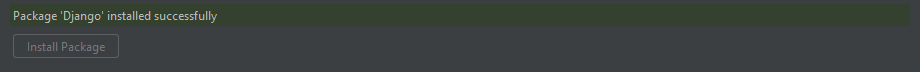
# ¿CREAR UN PROYECTO CON LA MAQUINA VIRTUAL?



# ¿DESCARGAR LOS INSTALADORES DE DJANGO AL PROYECTO?



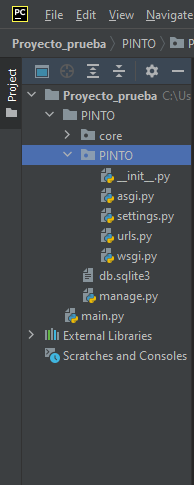




# ¿CREAR UN PROYECTO PARA PROGRAMAR EN DJANGO?

# ¿EJECUTAR EL PROYECTO Y EL MENSAJE DE FELICITACIONES?

# ¿CREAR UNA APPS CORE?



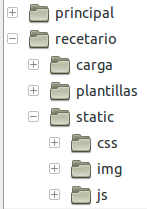
# ¿QUE ES LA CARPETA TEMPLATES?

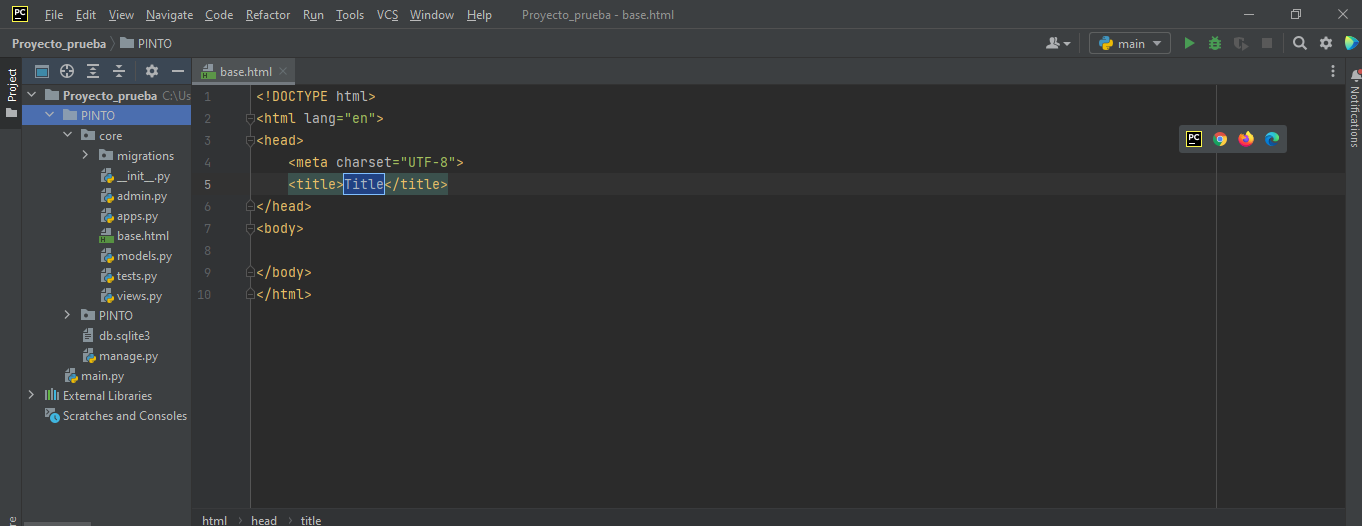
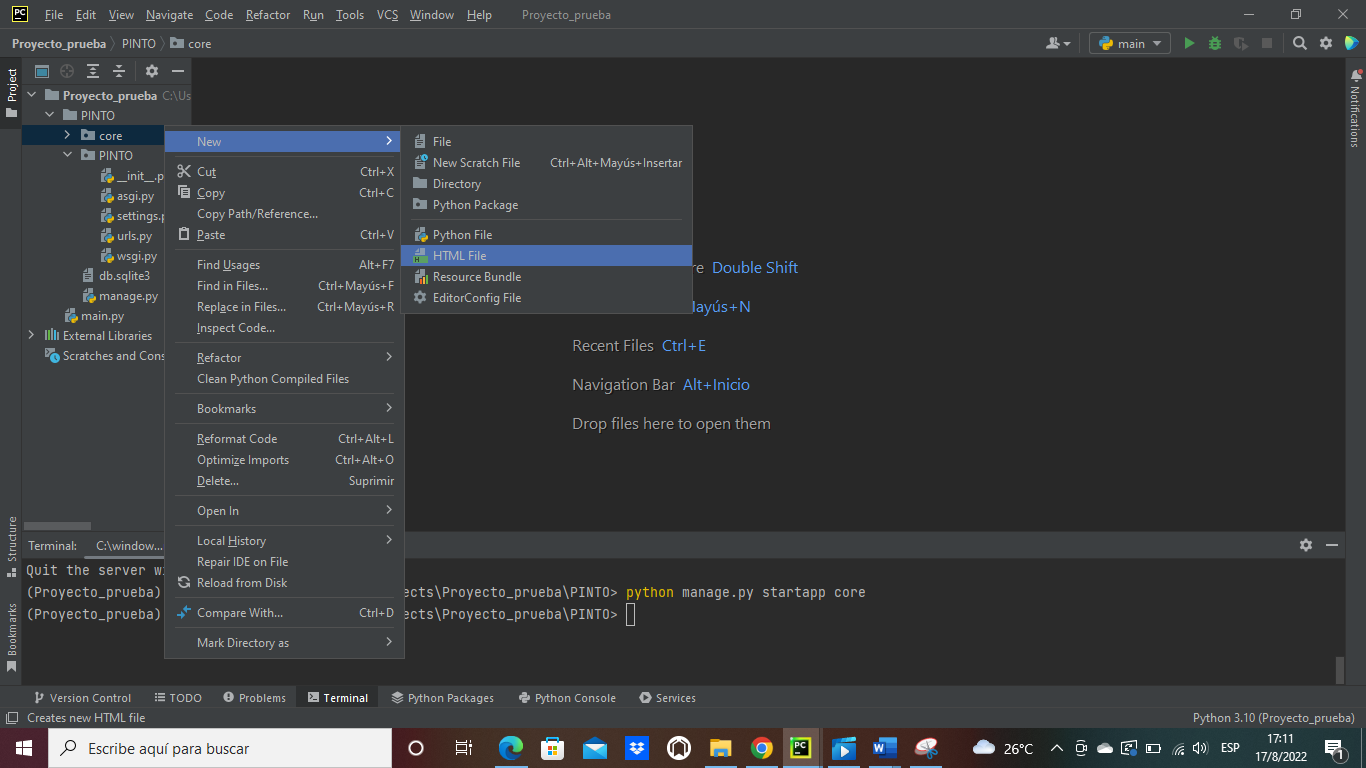
Al ser un marco web, Django necesita una forma conveniente de generar HTML dinámicamente. El enfoque más común se basa en plantillas. Una plantilla contiene las partes estáticas de la salida HTML deseada, así como alguna sintaxis especial que describe cómo se insertará el contenido dinámico

# ¿QUE ES LA CARPETA STATIC?

al crear el directorio: **static**, este se debe hacer dentro del directorio del proyecto, al mismo nivel que: carga y plantillas. Dentro del directorio static, debemos tener una carpeta por cada tipo de contenido estático, que deseemos incluir. Para el ejemplo tendremos tres subdirectorios en static: **css, img y js**.

Dentro de cada uno de estos directorios, debemos incluir nuestro contenido estático

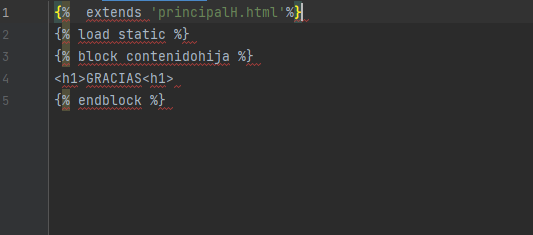
[](http://www.maestrosdelweb.com/images/2012/06/django-static-dirs.png)

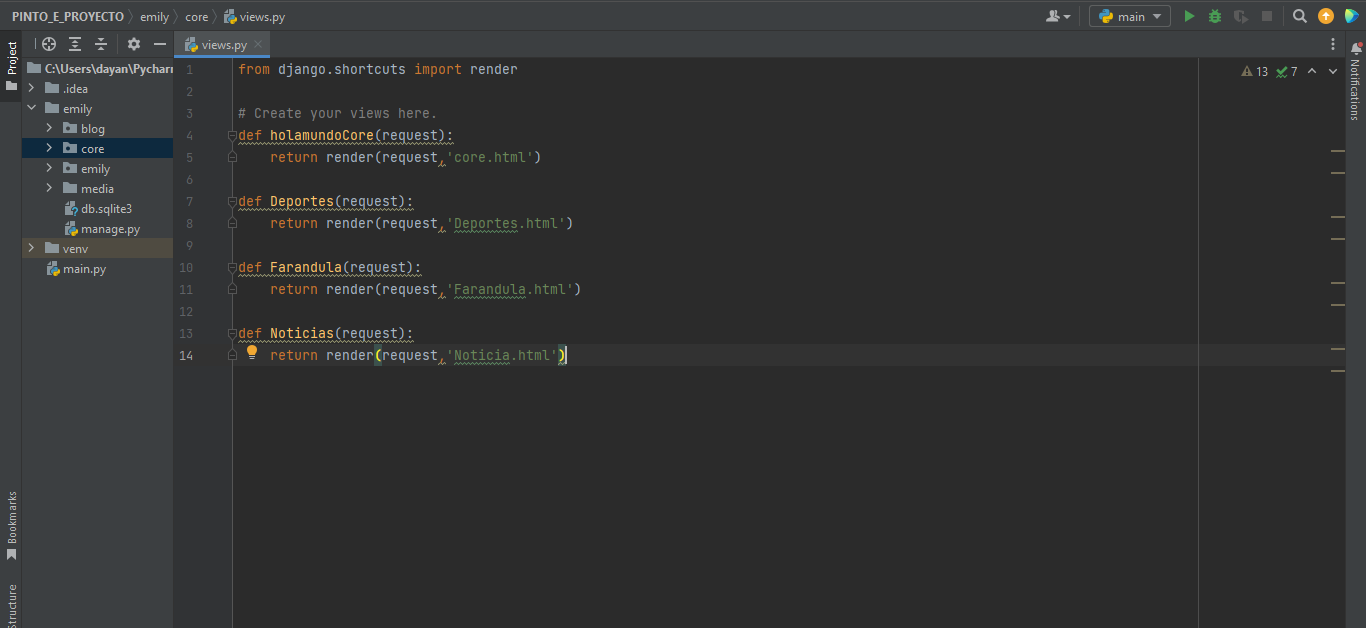
¿CREAR UN ARCHIVO BASE HTML EN LA APPS CORE***?***

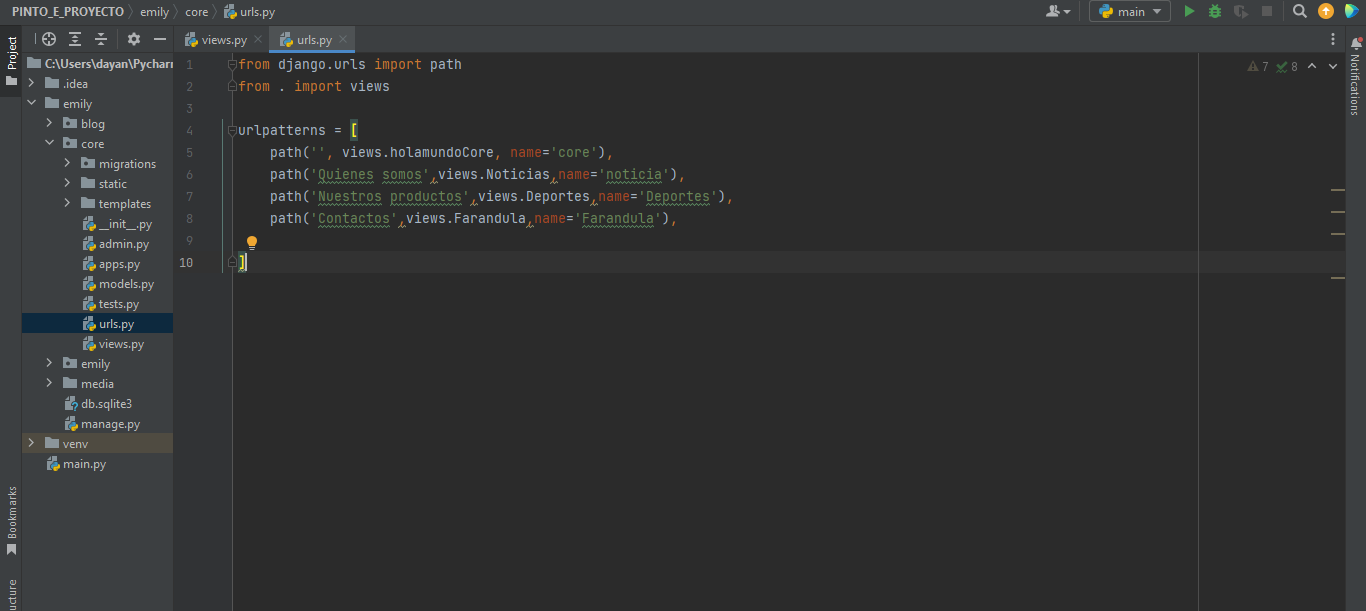
# ¿***COMO SE LLAMAN LOS CSS DESDE EL ARCHIVO BASE HTML?***

A este tipo de archivo se le llama static ,con los css podemos dar una presentación a la pagina ya sea poniendo márgenes, color de fondo, colores para cada menú entre otros cosas que queramos ponerle un estilo llamativo a la pagina

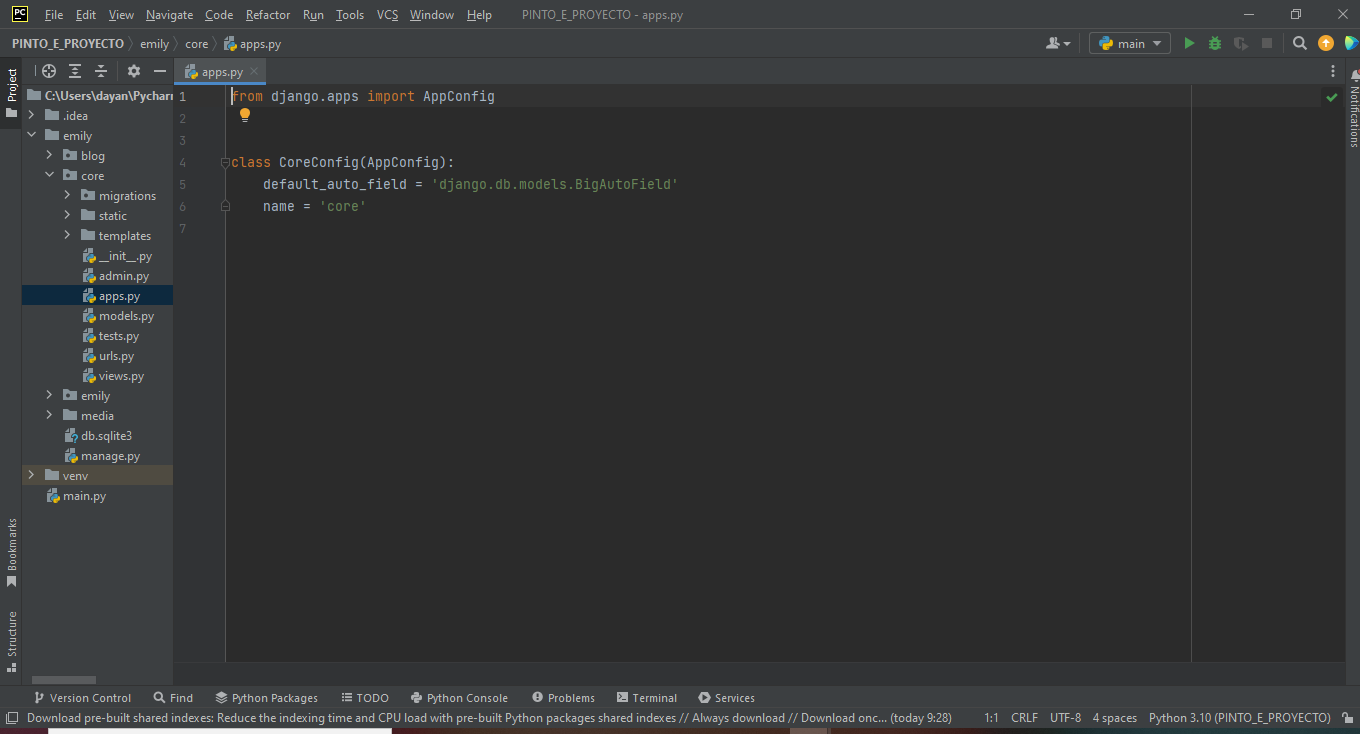
# ¿***COMO CONSUME UN ARCHIVO HIJO HTML AL UTILIZAR LA HERENCIA DEL ARCHIVO BASE HTML?***



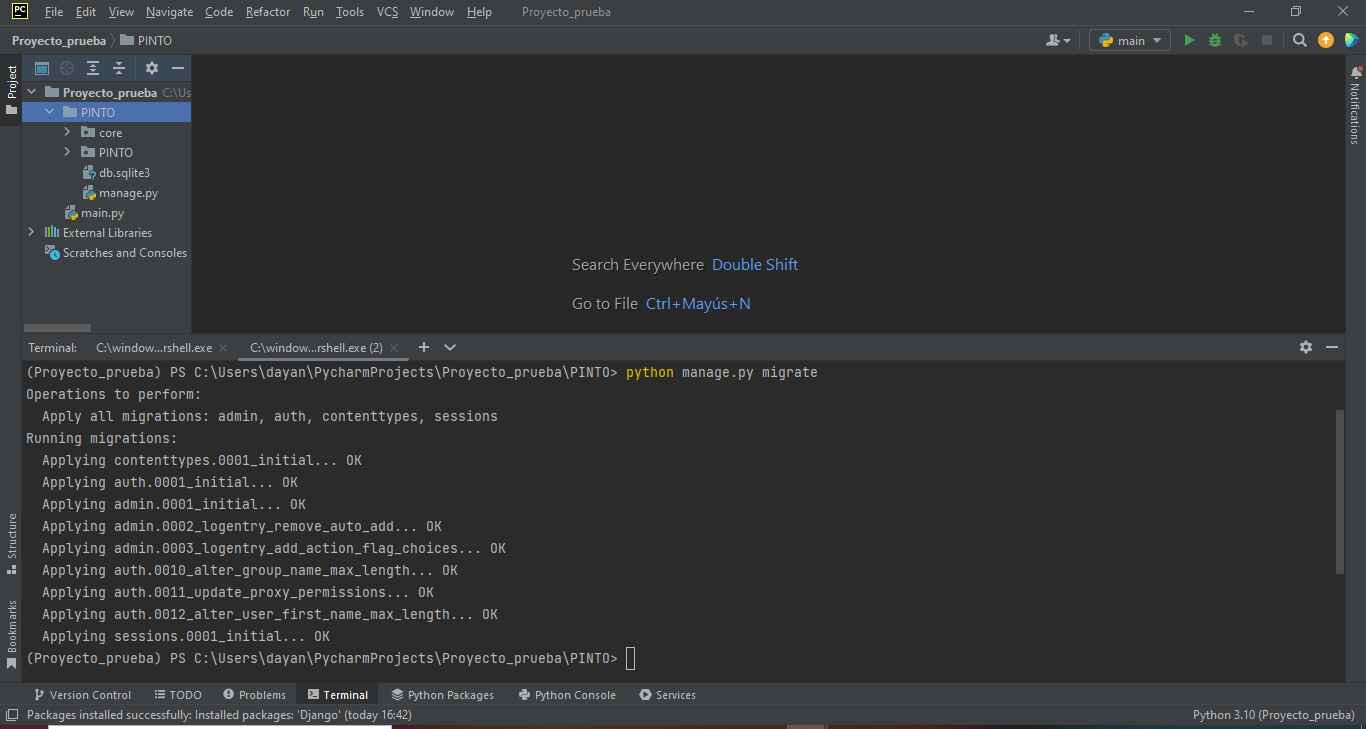
***¿***CREAR UN VIEW QUE LLAME AL HIJO?

***¿***CREAR LA URLS QUE LLAME AL VIEWS?

# INTEGRAR LA APLICACIÓN APPS CORE AL PROYECTO PRINCIPAL



# CREAR LAS TABLAS DEL SISTEMA DE USUARIOS PARA UTILIZAR EL PANEL DE ADMINISTRACION



# ¿CREAR UN USUARIO PARA PODER INGRESAR AL PANEL DE ADMINISTRACION?

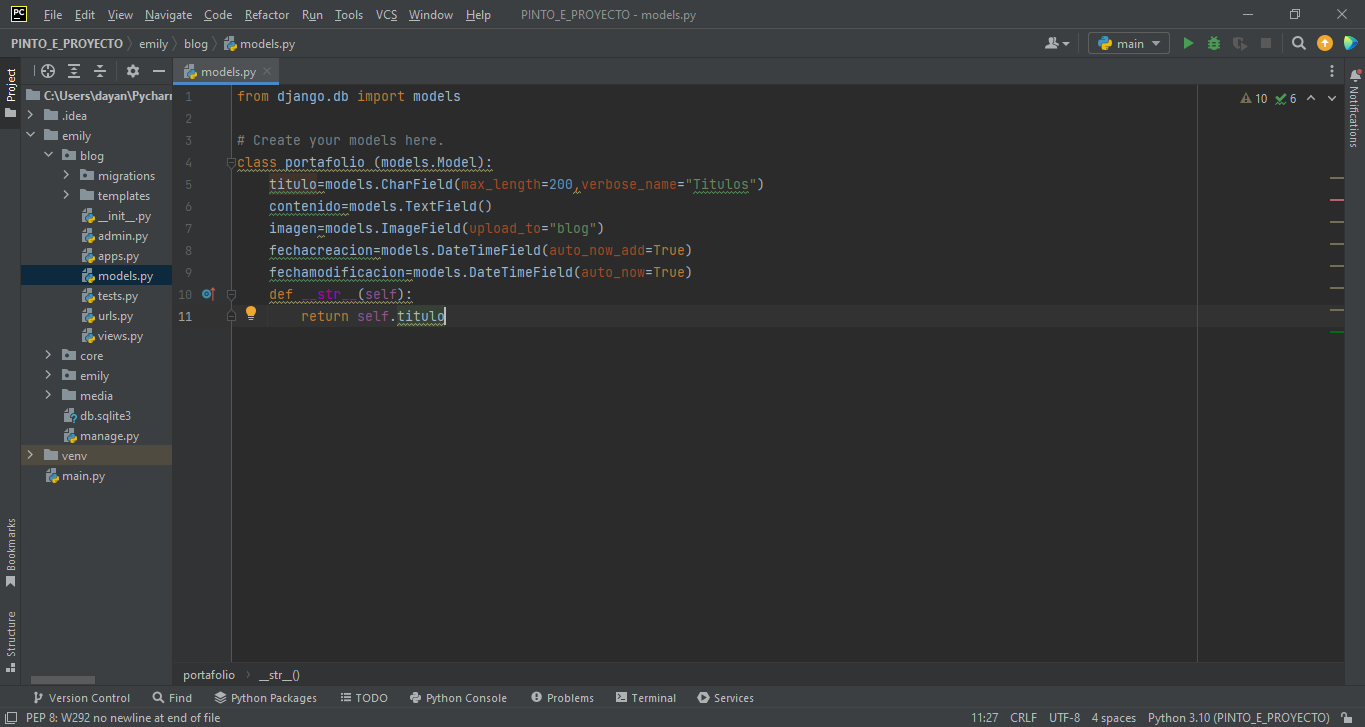
# ¿***QUE ES UN MODELO EN DJANGO***?

Un modelo es la fuente única y definitiva de información sobre sus datos. Contiene los campos y comportamientos esenciales de los datos que está almacenando. En general, cada modelo se asigna a una sola tabla de base de datos.

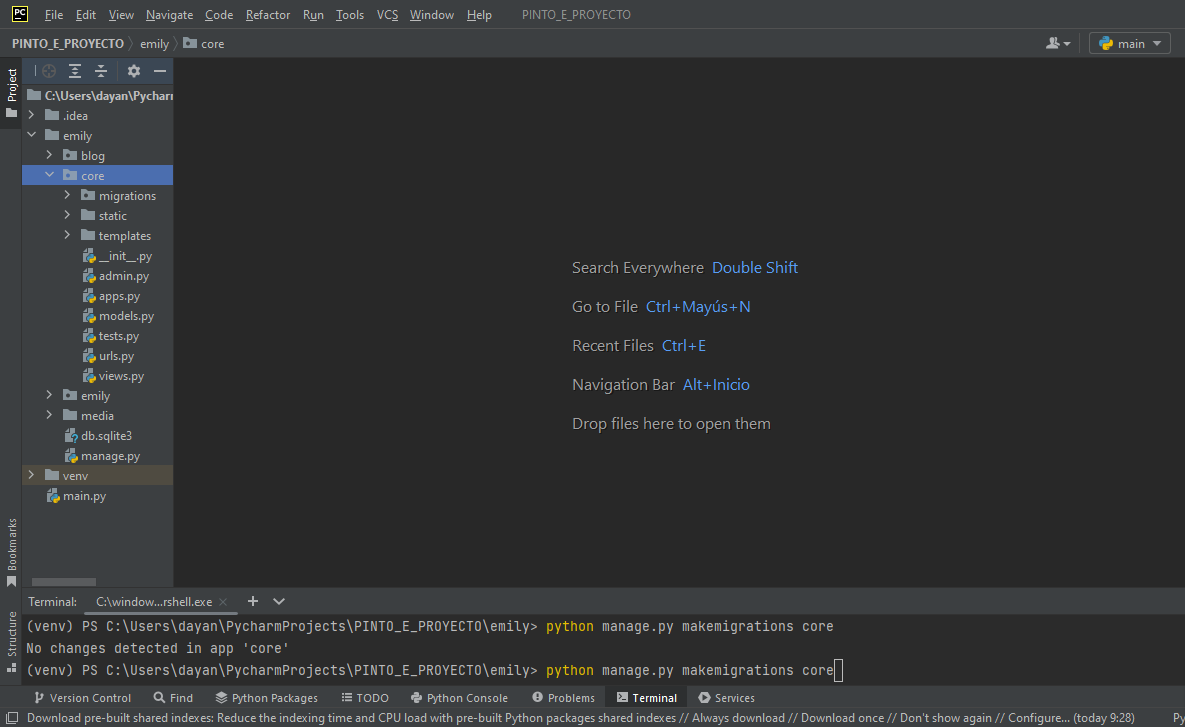
Lo básico:

* Cada modelo es una clase de Python que subclasifica [django.db.models.Model](https://runebook.dev/es/docs/django/ref/models/instances" \l "django.db.models.Model).
* Cada atributo del modelo representa un campo de la base de datos.
* Con todo esto, Django le ofrece una API de acceso a la base de datos generada automáticamente

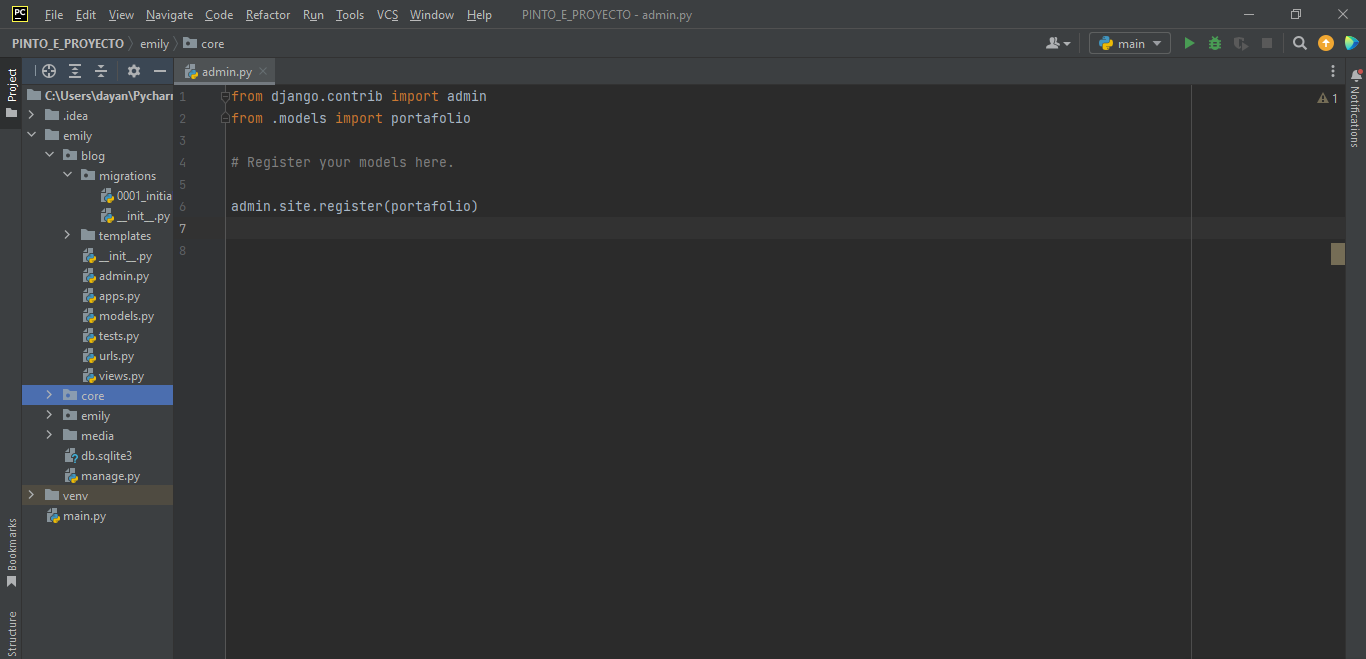
# ¿CREAR UN MODELO EN DJANGO?



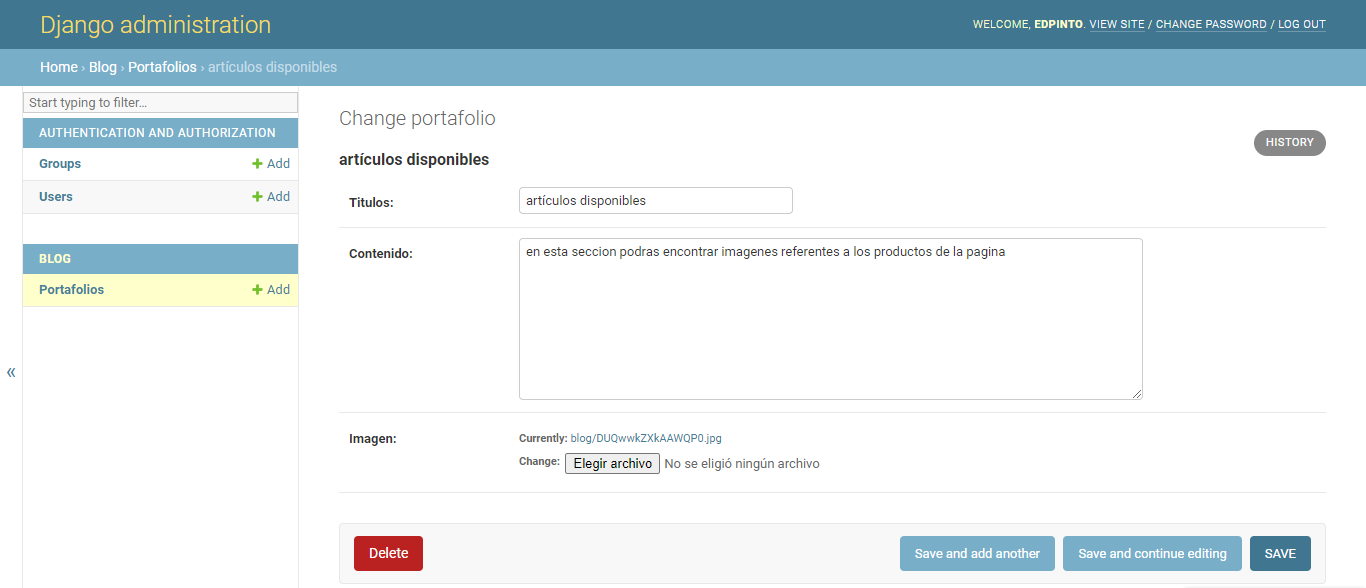
# MIGRAR EL MODELO A LA BASE DEL PANEL DE ADMINISTRACION



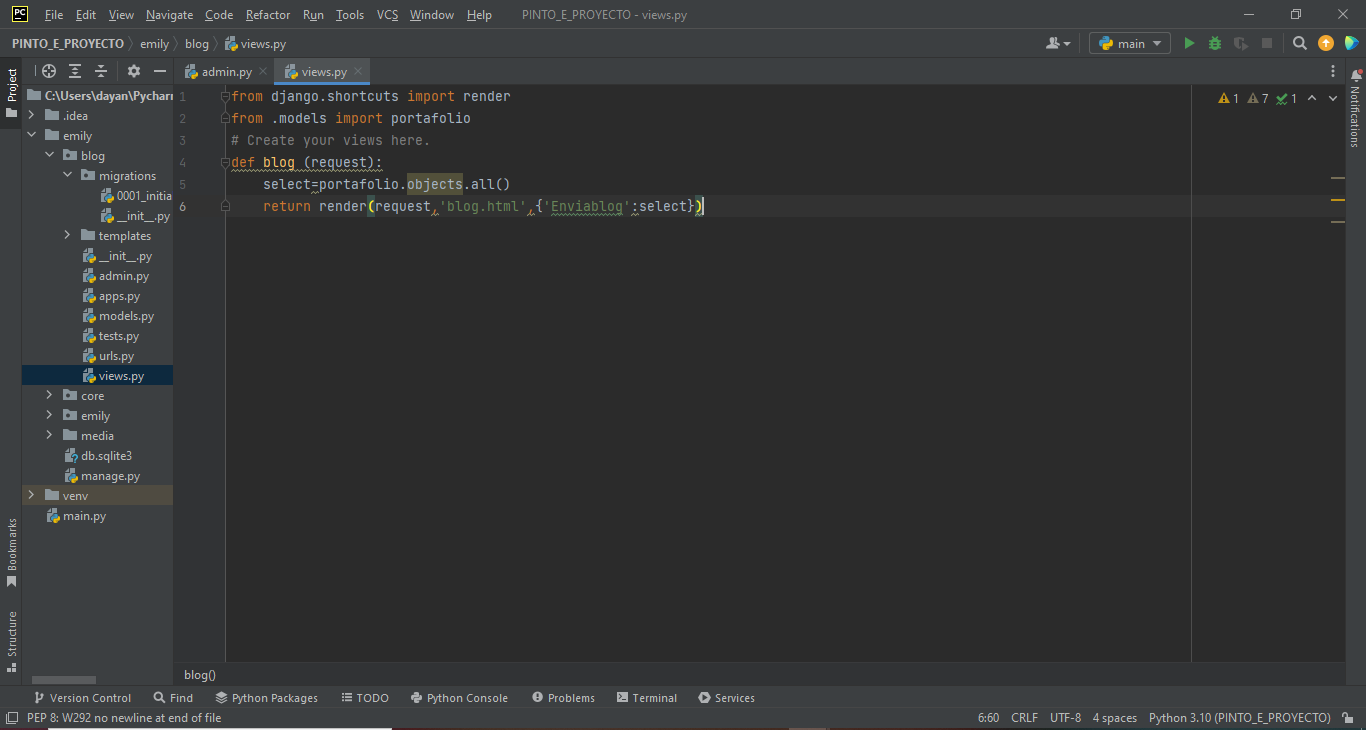
# INTEGRAR EL MODELO AL PANEL DE ADMINISTRACION



# INGRESAR INFORMACION AL MODELO POR EL PANEL DE ADMINISTRACION



# REALIZAR LA CONSULTA DE TODO LO INGRESADO EN EL MODELO DESDE EL VIEWS



# MOSTRAR LOS DATOS GUARDADOS EN EL MODELO AL HTML HIJO

