**Desarrollo de software**

**Estudiantes:**

Joselyn Benites Villalta.

**Curso:**

3ero G.

**Docente:**

Ing. Carlos Pazmiño.



**Contenido**

[1) ¿Qué es Django? 3](#_Toc111587764)

[2) ¿Qué es la máquina virtual en Django? 3](#_Toc111587765)

[3) ¿Qué es MVT en Django? 3](#_Toc111587766)

[4) Crear un proyecto con la máquina virtual. 4](#_Toc111587767)

[5) Descargar los instaladores de Django al proyecto. 5](#_Toc111587768)

[6) Crear un proyecto para programar en Django 6](#_Toc111587769)

[7) Ejecutar el proyecto y el mensaje de felicitaciones. 7](#_Toc111587770)

[8) Crear una Apps core. 8](#_Toc111587771)

[9) ¿Qué es la Carpeta Templates? 15](#_Toc111587772)

[10) ¿Qué es la Carpeta stactic? 15](#_Toc111587773)

[11) Crear un archivo base html en la APPS core. 16](#_Toc111587774)

[12) Como se llaman a los CSS desde el archivo base html. 16](#_Toc111587775)

[13) Como consume un archivo hijo html al utilizar la herencia del archivo base html. 17](#_Toc111587776)

[14) Crear un view que llame al html hijo. 17](#_Toc111587777)

[15) Crear la urls que llame al views. 17](#_Toc111587778)

[16) Integrar la aplicación APPS core al proyecto principal. 18](#_Toc111587779)

[17) Crear las tablas del sistema de usuarios para utilizar el panel de administración. 19](#_Toc111587780)

[18) Crear un usuario para poder ingresar al Panel de Administración. 20](#_Toc111587781)

[19) Que es un modelo en Django. 21](#_Toc111587782)

[20) Crear un modelo en Django. 21](#_Toc111587783)

[21) Migrar el Modelo a la base del Panel de Administración. 21](#_Toc111587784)

[22) Integrar el Modelo al Panel de Administración. 22](#_Toc111587785)

[23) Ingresar información al modelo por el Panel de Administración. 22](#_Toc111587786)

[24) Realizar la consulta de todo lo ingresado en el modelo desde el views. 23](#_Toc111587787)

[25) Mostrar los datos guardados en el modelo al html hijo. 24](#_Toc111587788)

# ¿Qué es Django?

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

# ¿Qué es la máquina virtual en Django?

Es un sotfware que crea una capa independiente donde se emula el funcionamiento de un ordenador real con todos los componentes de hadware que necesita para funcionar.

# ¿Qué es MVT en Django?

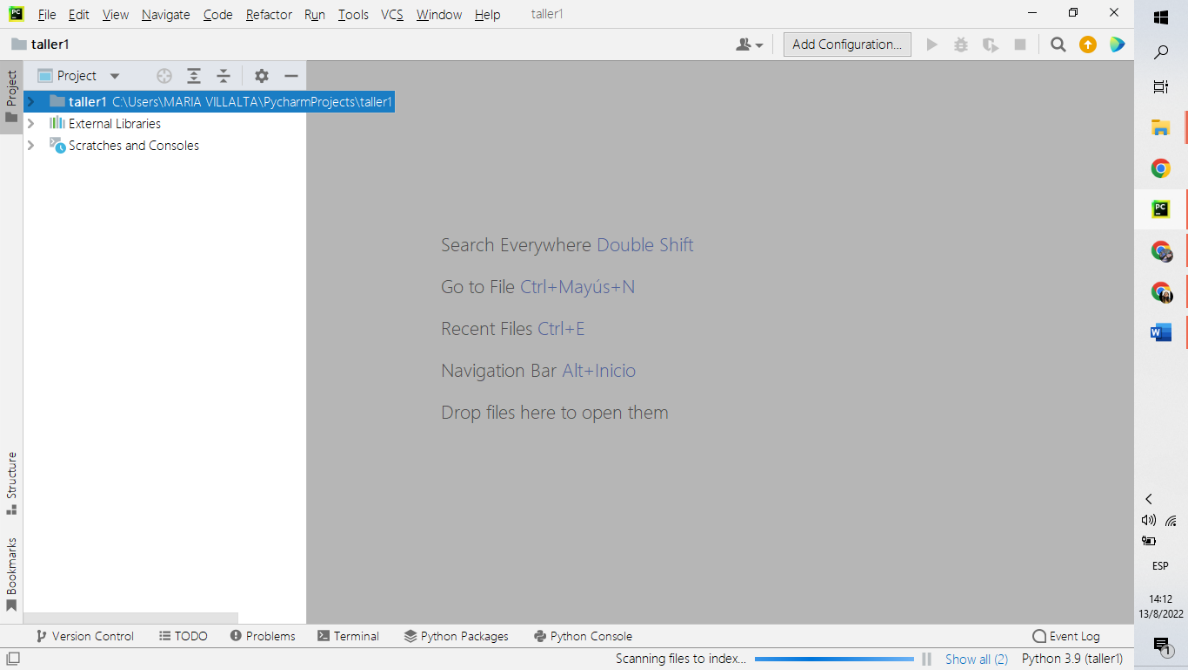
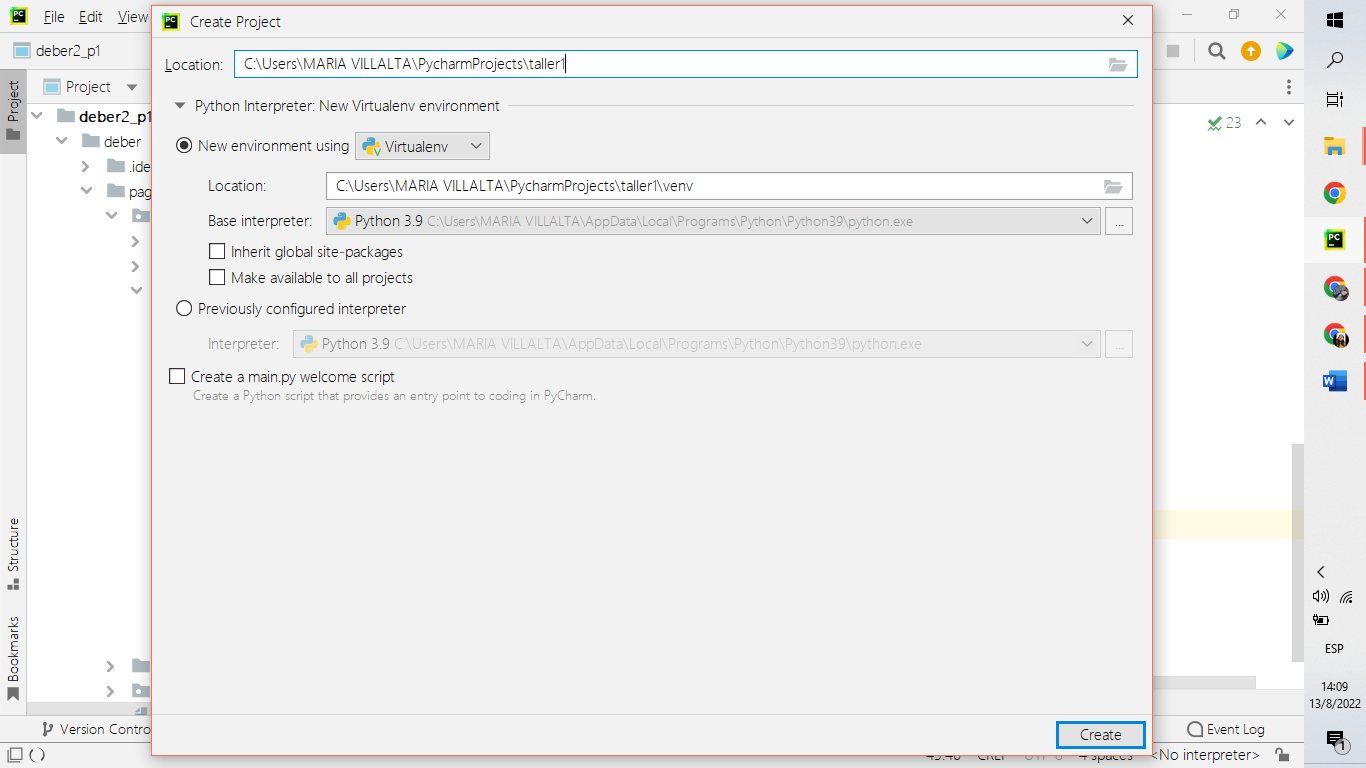
Las siglas en ingles *MTV* corresponden a *Model (Modelo)*, *Template (Plantilla)* y *View (Vista)*. En este post me referiré a las plantillas como templates, ya que su uso en el español es bastante común.

En la practica el patrón MTV es muy similar al MVC a tal punto que se puede decir que Django es un framework MVC. Realmente este no se desvía demasiado del patrón Modelo Vista Controlador, simplemente lo implementa de una manera distinta y para evitar confusiones es llamado MTV.

En Django, el controlador sigue estando presente, nada más que de una manera intrínseca, ya que todo el framework Django es el controlador.

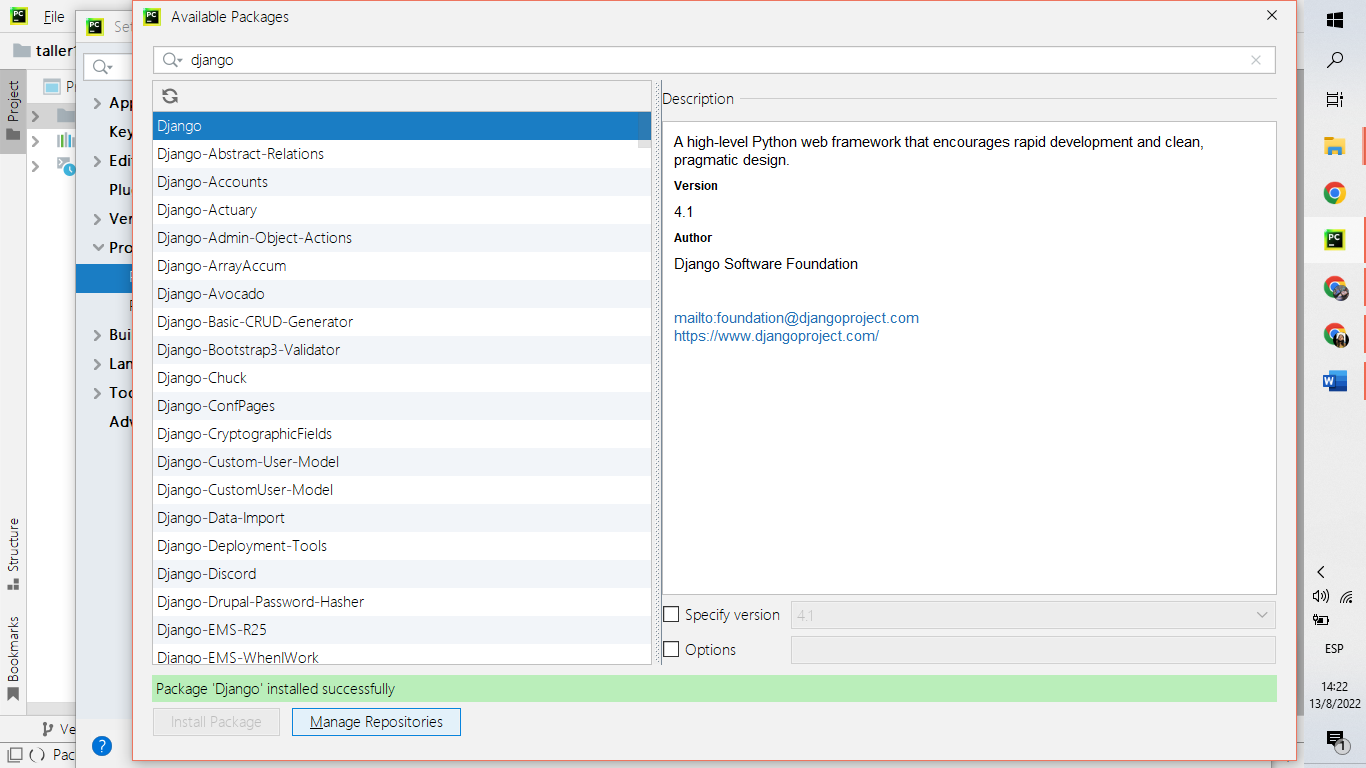
* **Modelo**: Maneja todo lo relacionado con la información, esto incluye como acceder a esta, la validación, relación entre los datos y su comportamiento.
* **Vista**: Es un enlace entre el modelo y el template. Decide qué información será mostrada y por cual template.
* **Template**: Decide como será mostrada la información.

# Crear un proyecto con la máquina virtual.



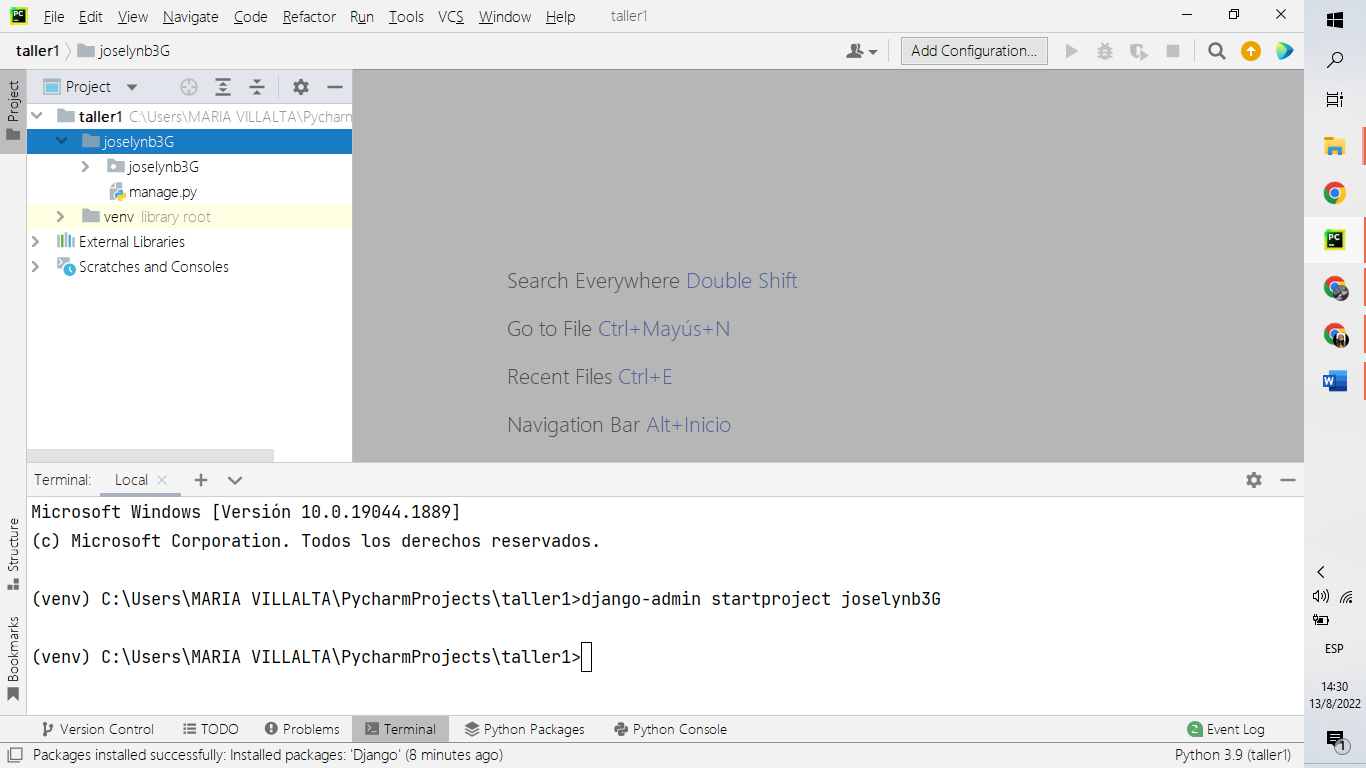
# Descargar los instaladores de Django al proyecto.

* Dentro de nuestro pycharm nos dirigimos a la parte superior de nuestra pantalla y hacemos clic en **file**.
* Luego a settings.
* En settings nos vamos a **Project: taller1** y luego **Python interpreter**
* Le damos clic en **+** y buscamos **Django**, procedemos a instalar el paquete.

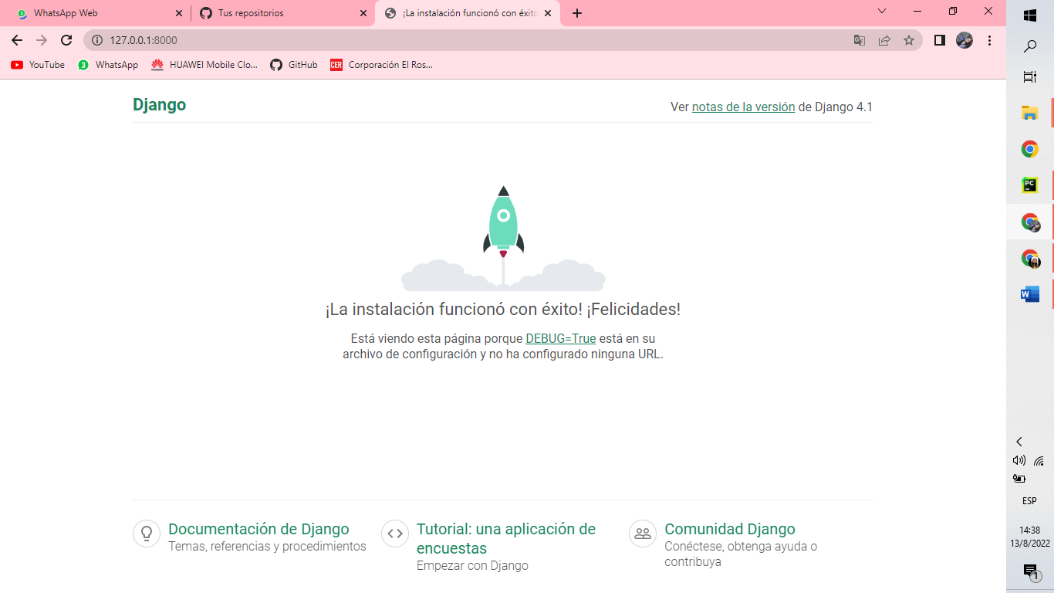


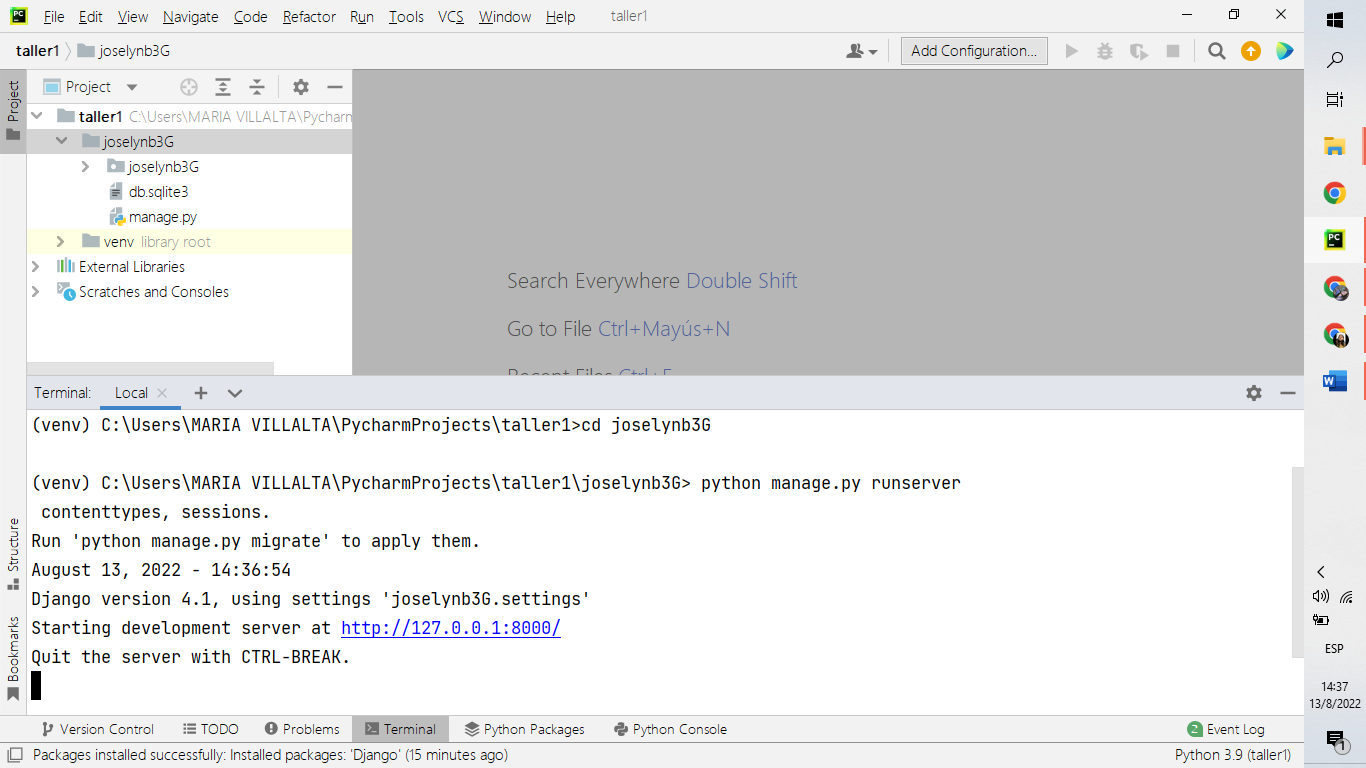
# Crear un proyecto para programar en Django

Para crear un proyecto en django ejecutamos con el comando **django-admin startproject** joselynb3G en el terminal de nuestro pycharm.



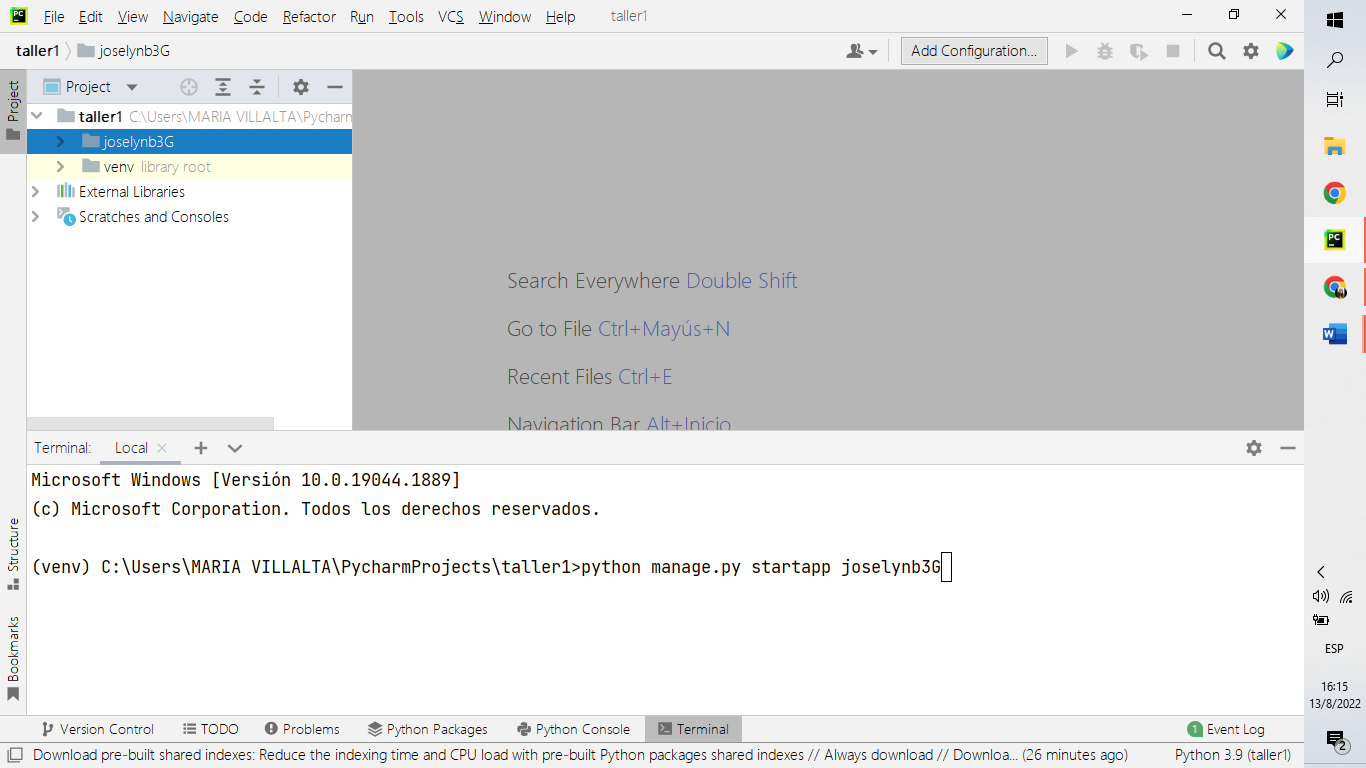
# Ejecutar el proyecto y el mensaje de felicitaciones.

Para ejecutar el proyecto utilizamos el comando **python manage.py runserver.**

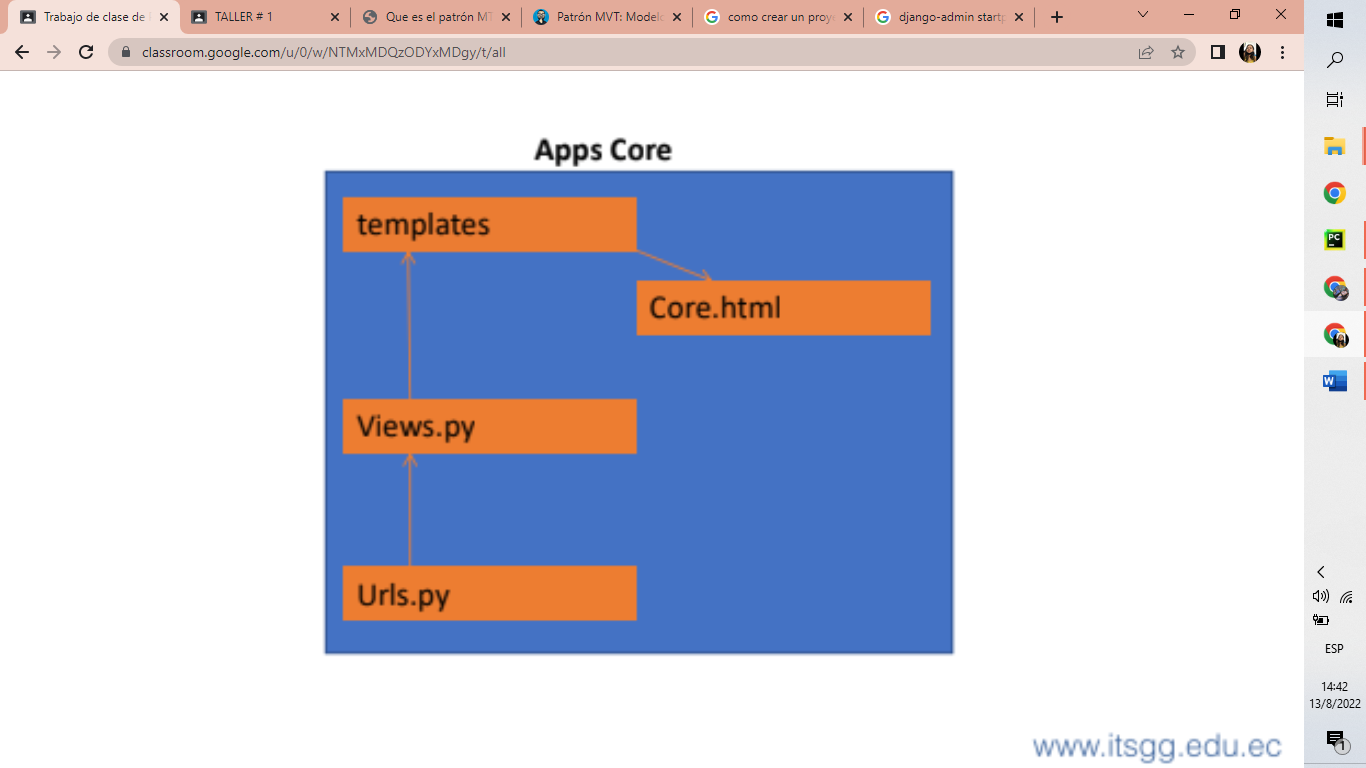


# Crear una Apps core.

\*En el terminal se escribe el código **python manage.py startapp** y el nombre del proyecto.

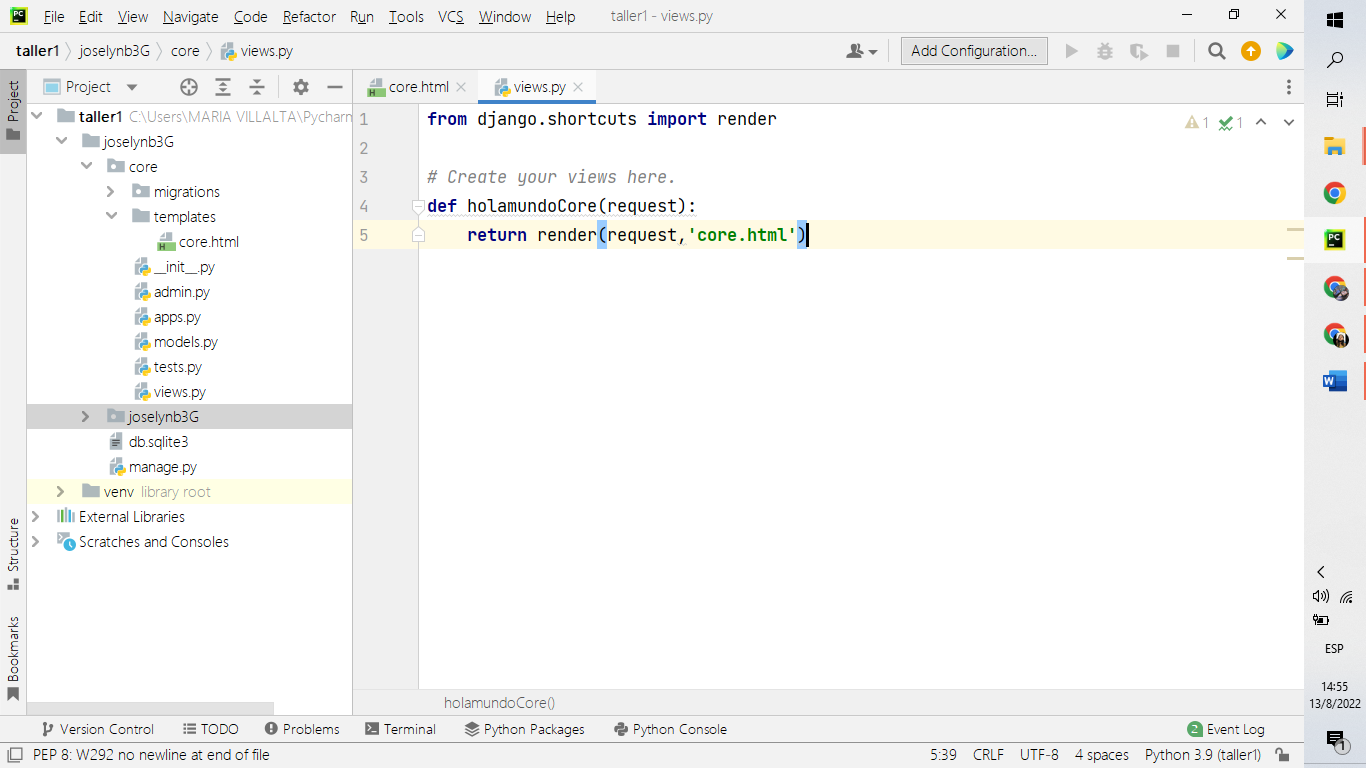


\* Debe tener creado los siguiente dentro de la core:



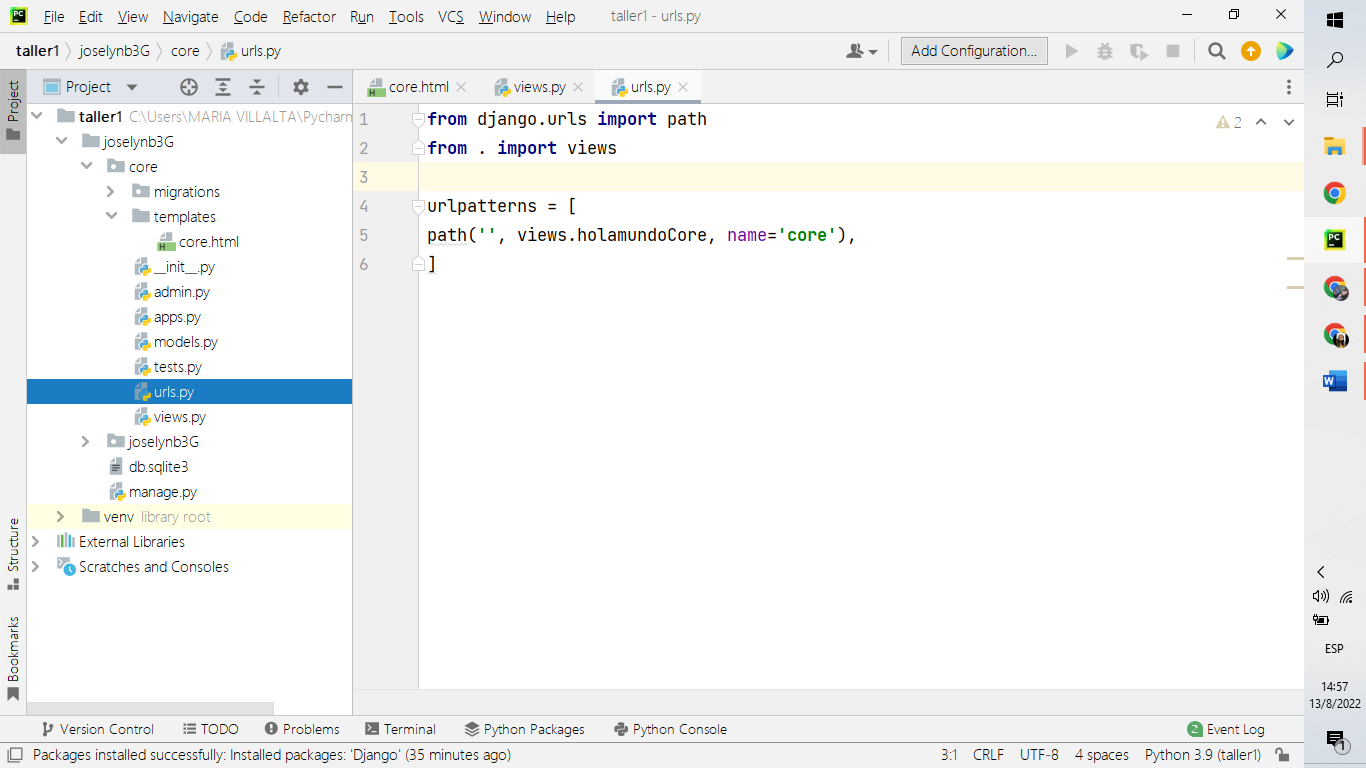
\* Dentro de nuestro Proyecto Core, creamos la carpeta **templates**, dentro de ella creamos un archivo html.



\* Dentro de nuestro **proyecto Core**, modificamos el **Views**, llamando a nuestro Html.

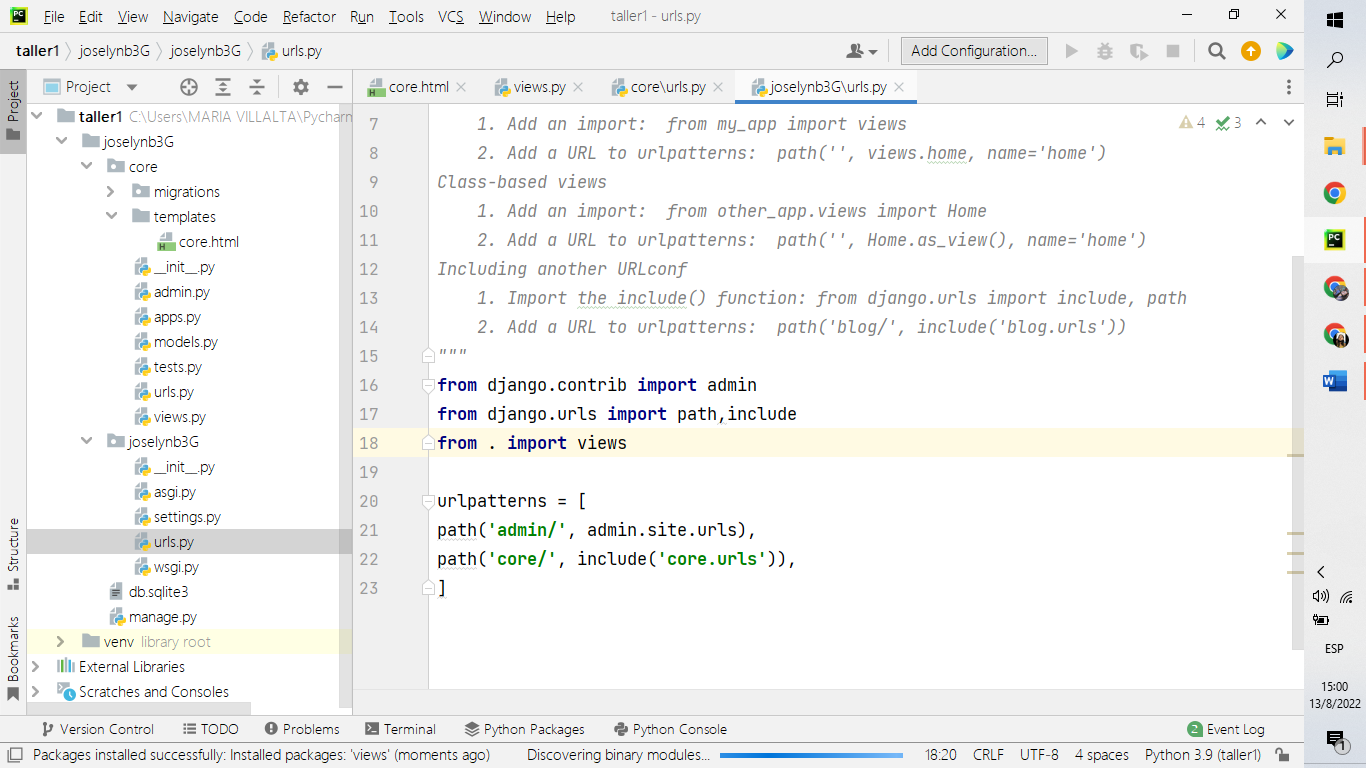
**def** holamundoCore(request):  
 **return** render(request,**'base.html'**)

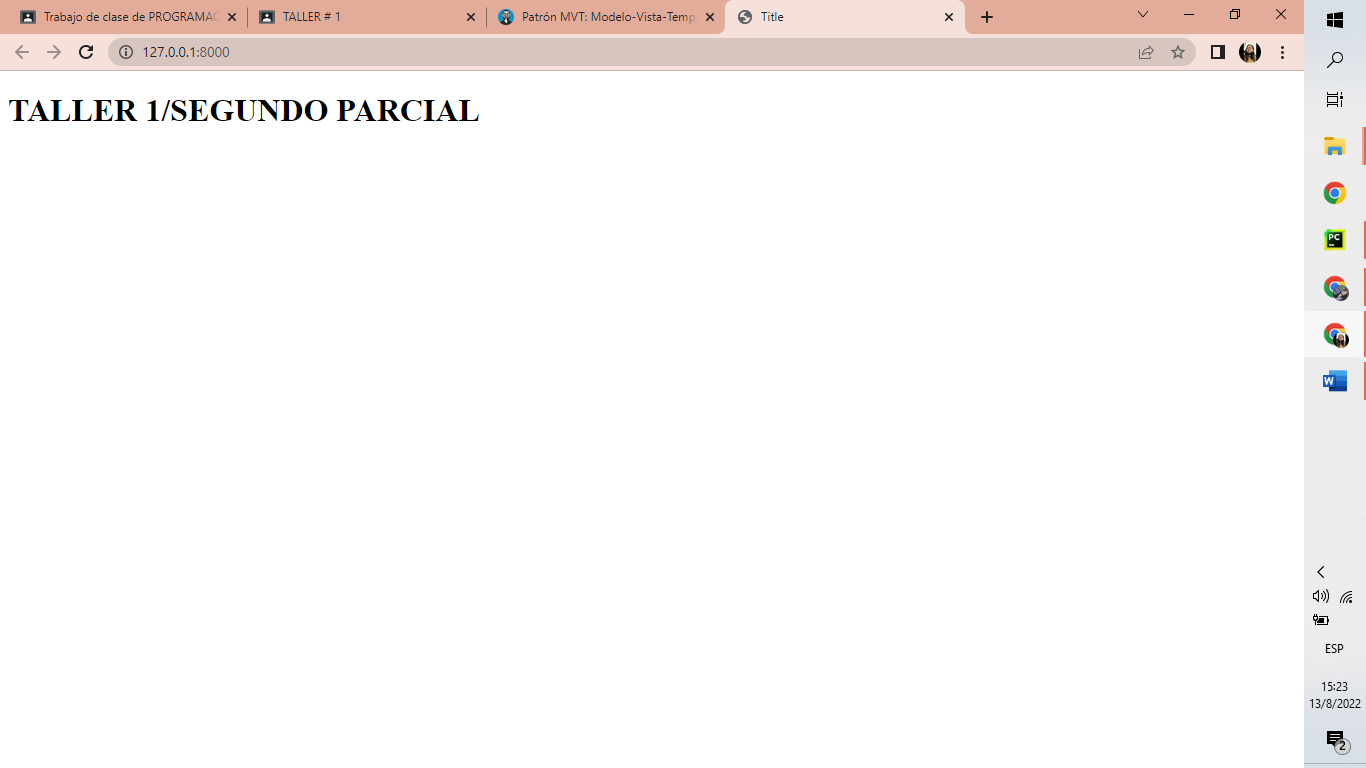
\* Dentro de nuestro **proyecto Core**, creamos el Urls.py

**from** django.urls **import** path  
**from** . **import** views  
urlpatterns = [  
path(**''**, views.holamundoCore, name=**'core'**),  
]

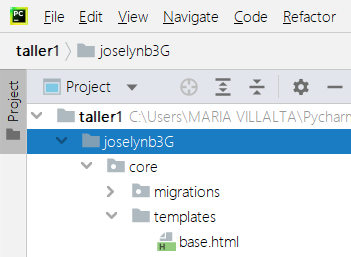
\* Dentro de nuestro proyecto principal(taller1) Urls.py debemos llamar al Urls.py de nuestra app creada core.

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

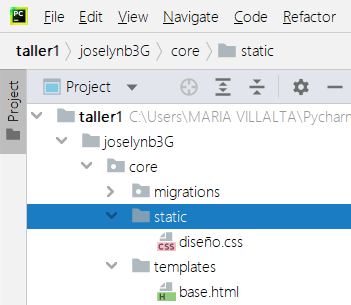


\*Volvemos a levantar el servidor con el comando **python manage.py runserver** y listo.

# ¿Qué es la Carpeta Templates?

En un archivo de texto que determina la estructura o diseño de un archivo (como una página HTML), con marcadores usados para representar el contenido real. **Django** automáticamente buscará plantillas en un directorio llamado '**templates**' de su aplicación.

# ¿Qué es la Carpeta stactic?

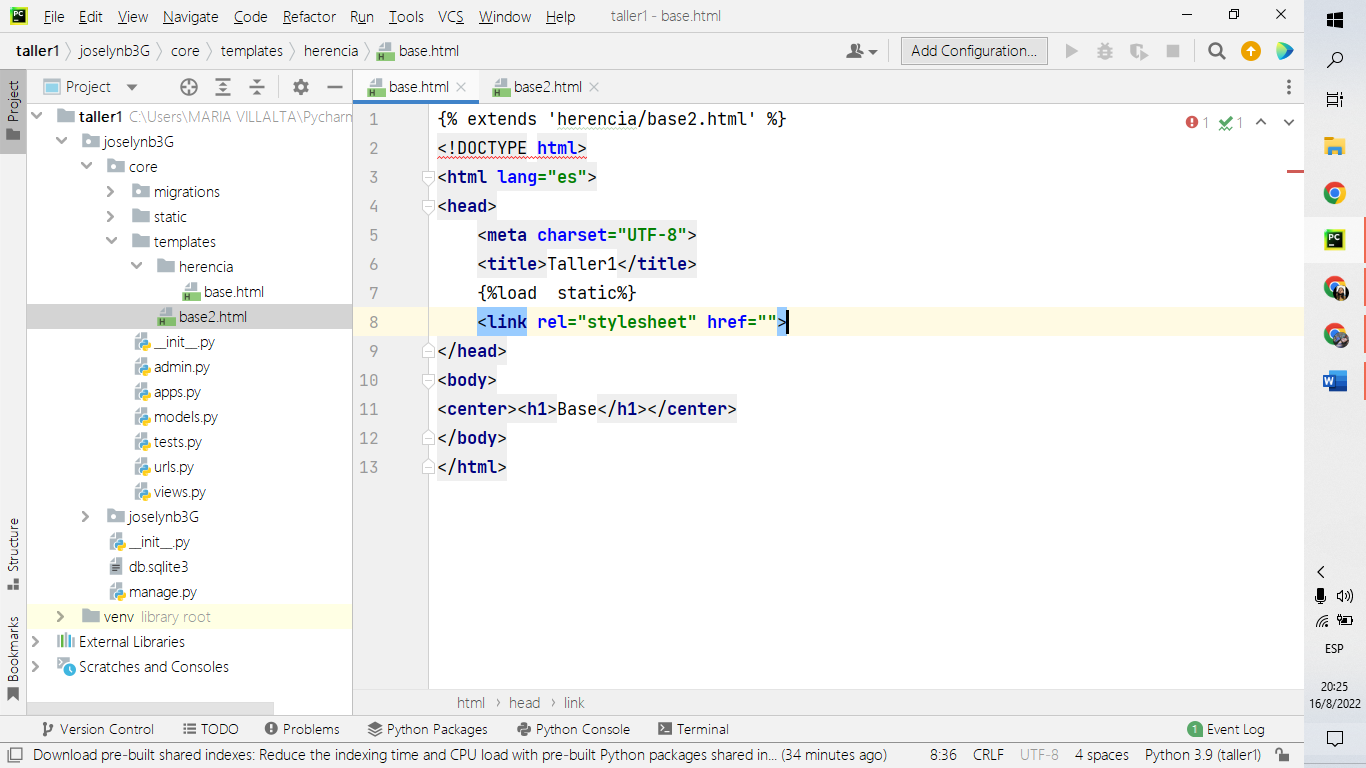
Esta carpeta contiene todos los archivos de estilos: css, etc.

# Crear un archivo base html en la APPS core.

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

Se lo llama con:

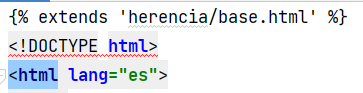
{%load static%}  
<**link rel="stylesheet" href=""**>

Y se lo ubica en el head:

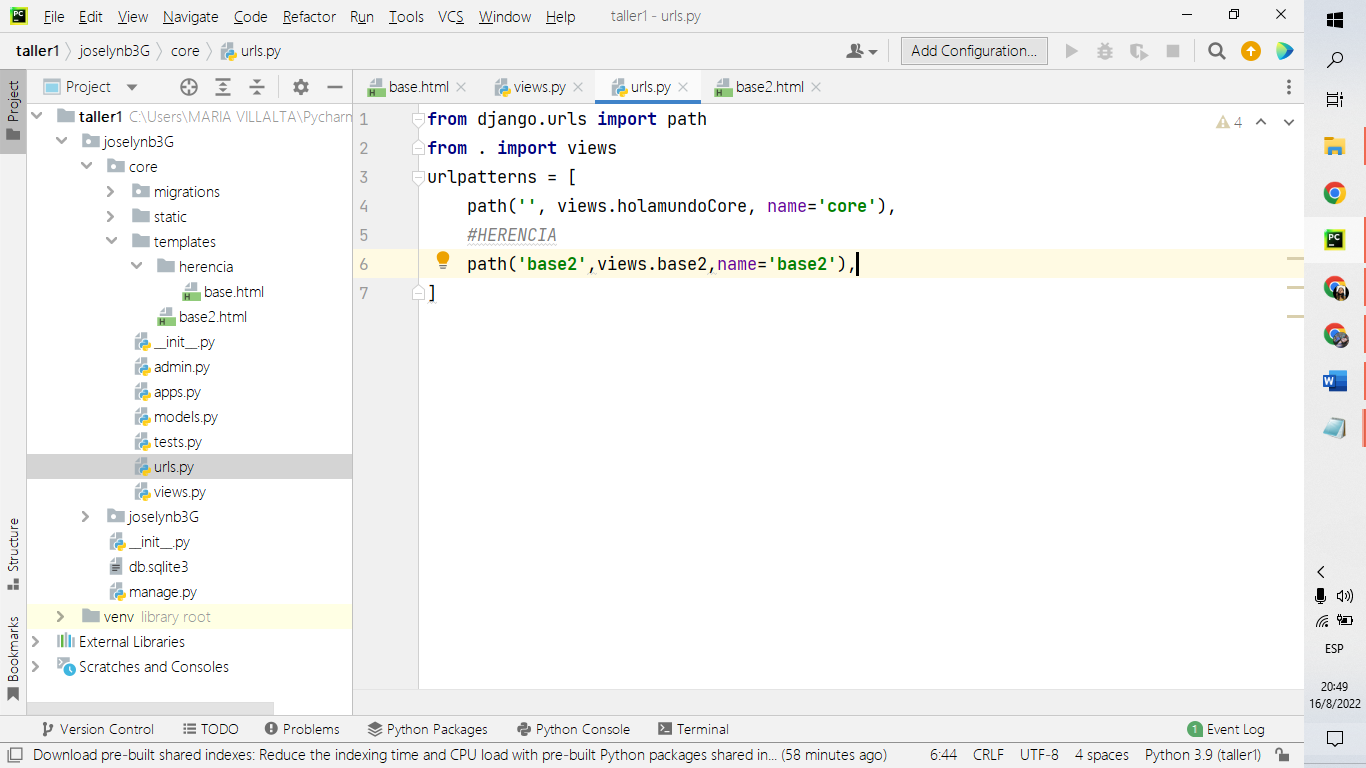
# Como consume un archivo hijo html al utilizar la herencia del archivo base html.

El hijo recibe la herencia directa del archivo padre mediante el comando:

**{% extends 'carpeta\_de\_la\_herencia/archivo\_padre\_herencia.html' %}**

****

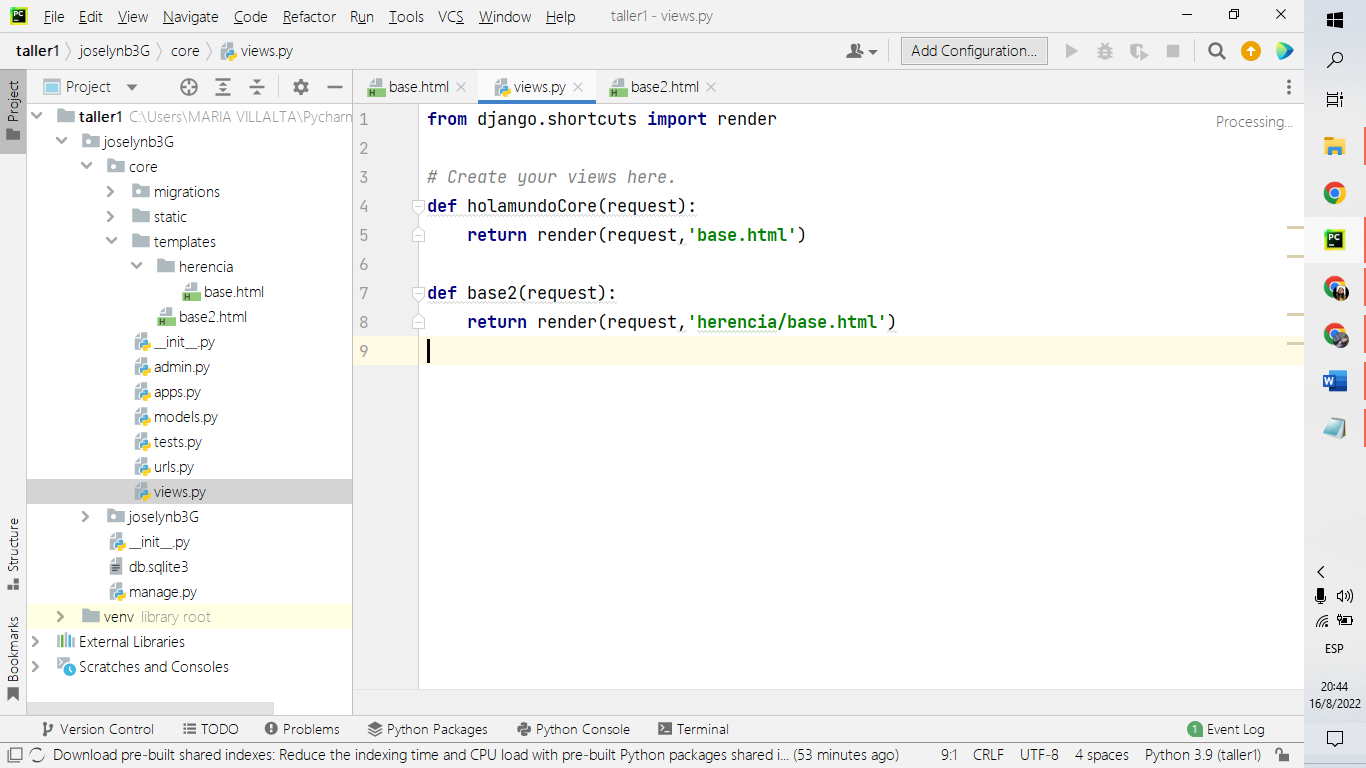
# Crear un view que llame al html hijo.

Es llamado de la siguiente manera:

**def** base2(request):  
 **return** render(request,**'herencia/base.html'**)

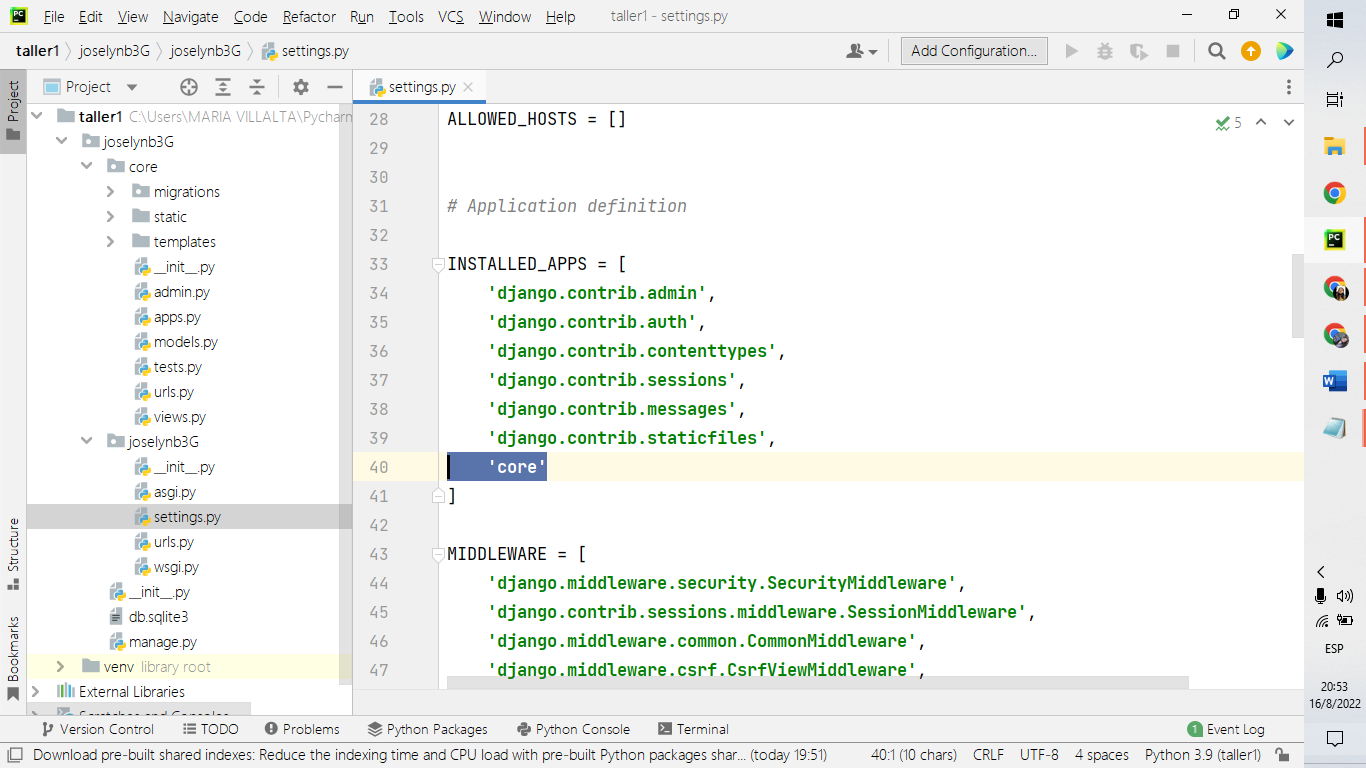
# Crear la urls que llame al views.

Llamado de la siguiente manera:

path(**'base2'**,views.base2,name=**'base2'**),

# Integrar la aplicación APPS core al proyecto principal.

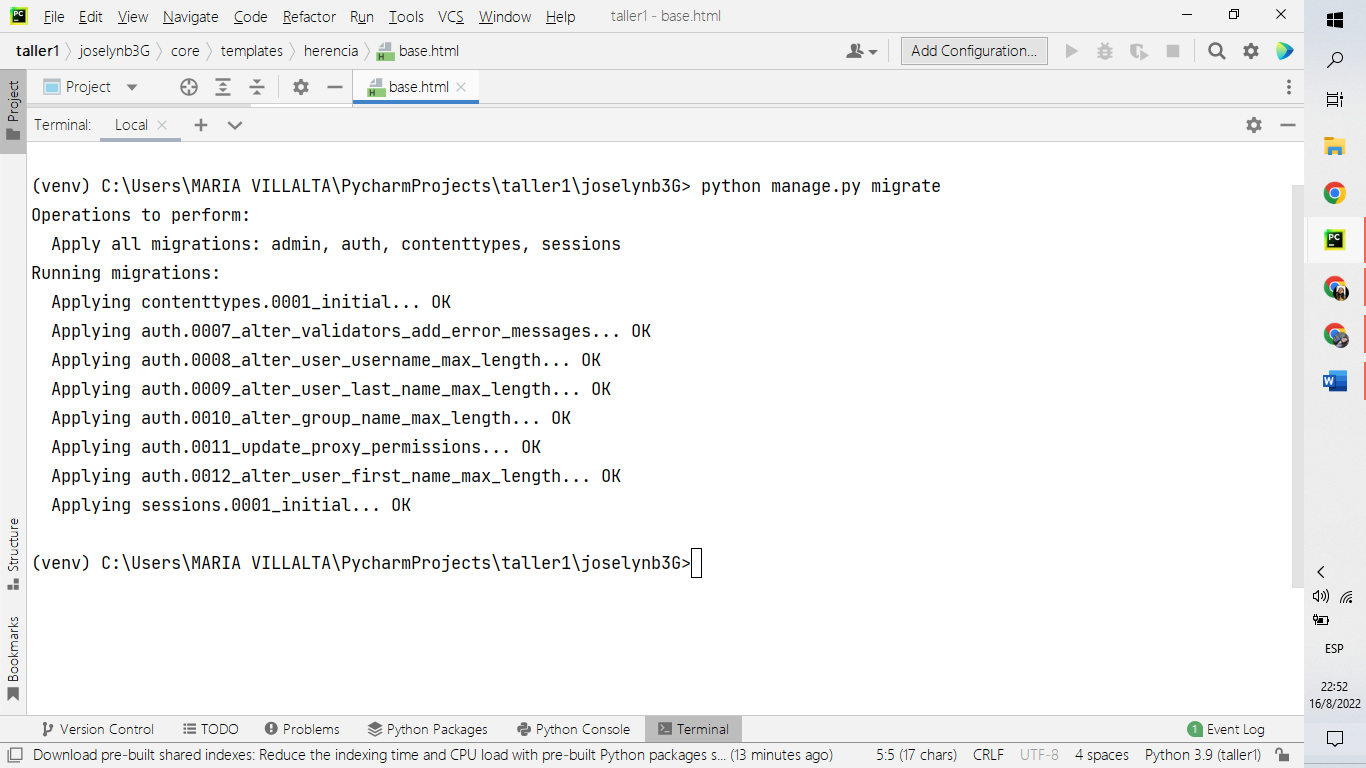
\*Nos dirigimos al **settings.py** en la carpeta principal para poner nuestro core y que Django lo reconozca como app.



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

Primero que todo migramos las tablas que Django necesita para subir toda nuestra información a la base de datos.

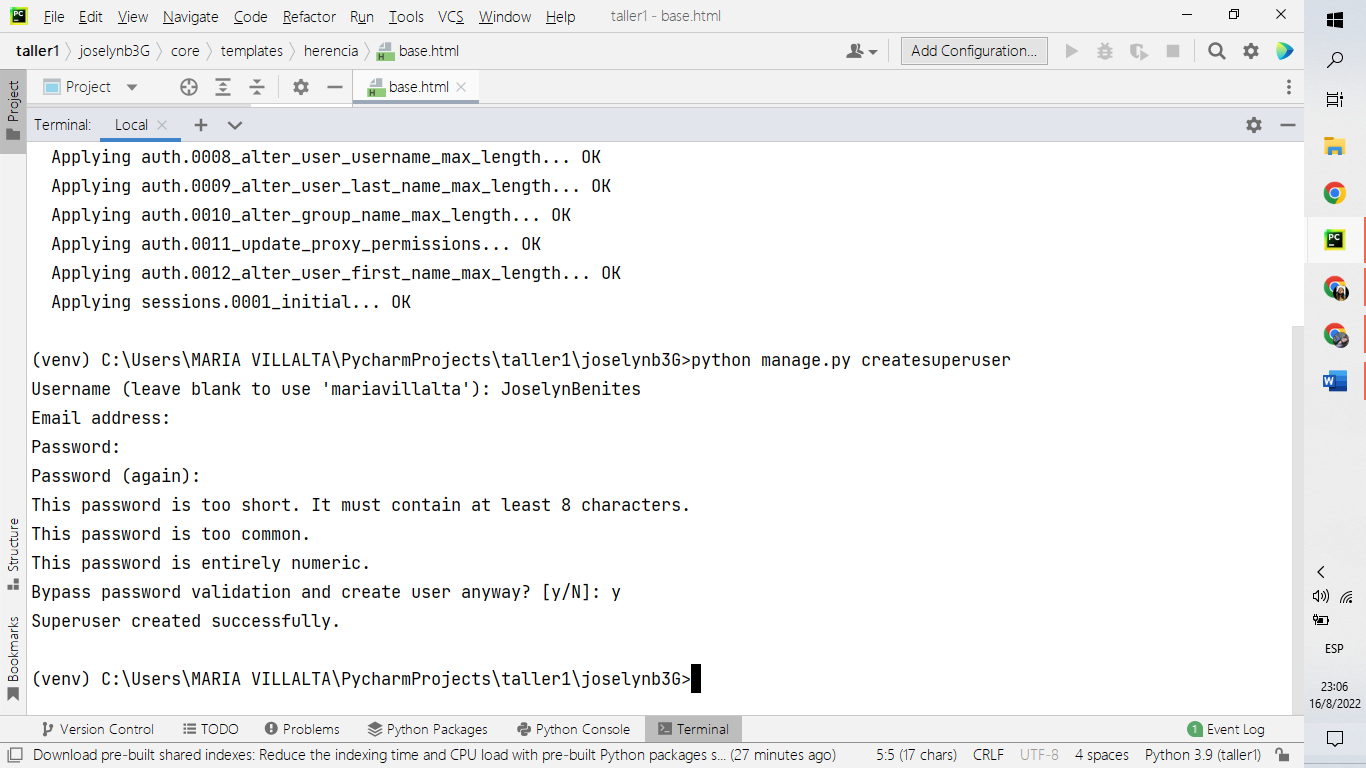
El comando es: **python manage.py migrate**



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

Para la creación de usuarios utilizamos el siguiente comando en la terminal:

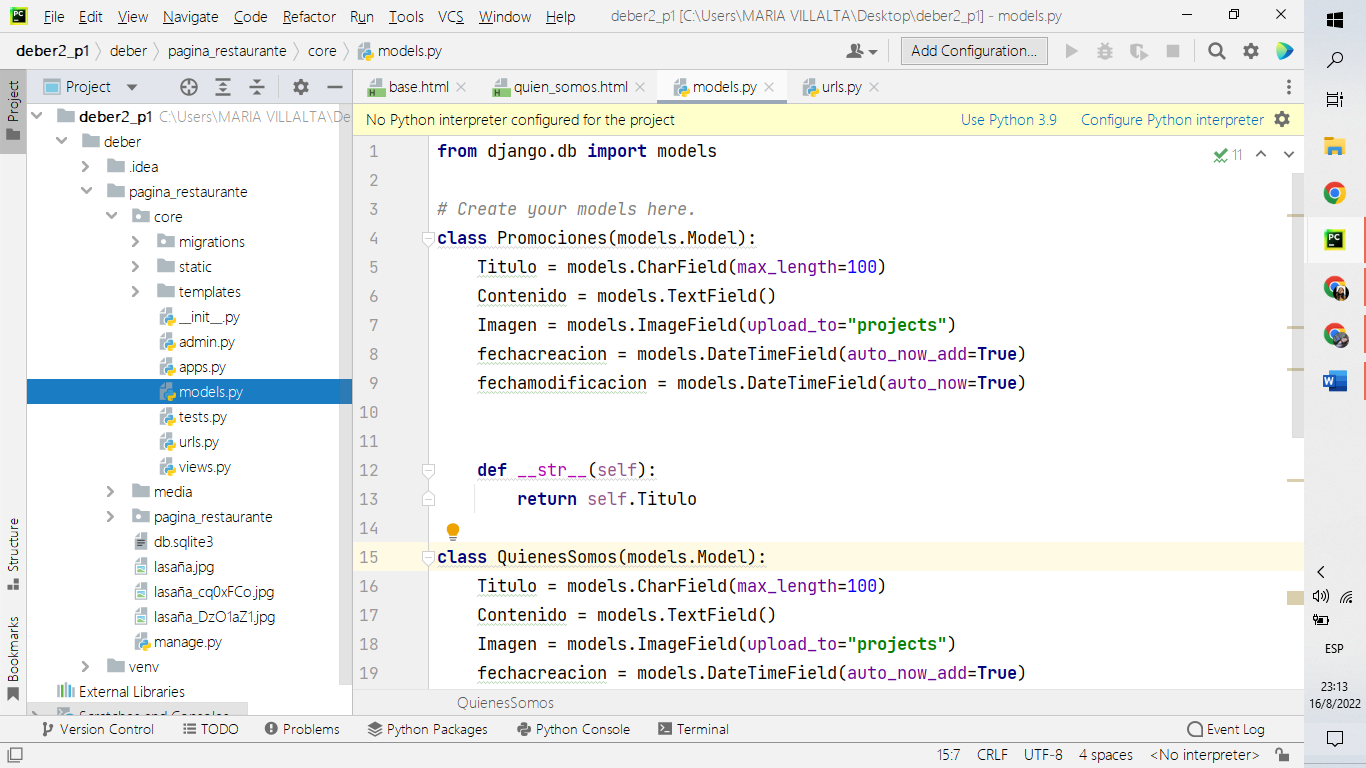
**python manage.py createsuperuser**



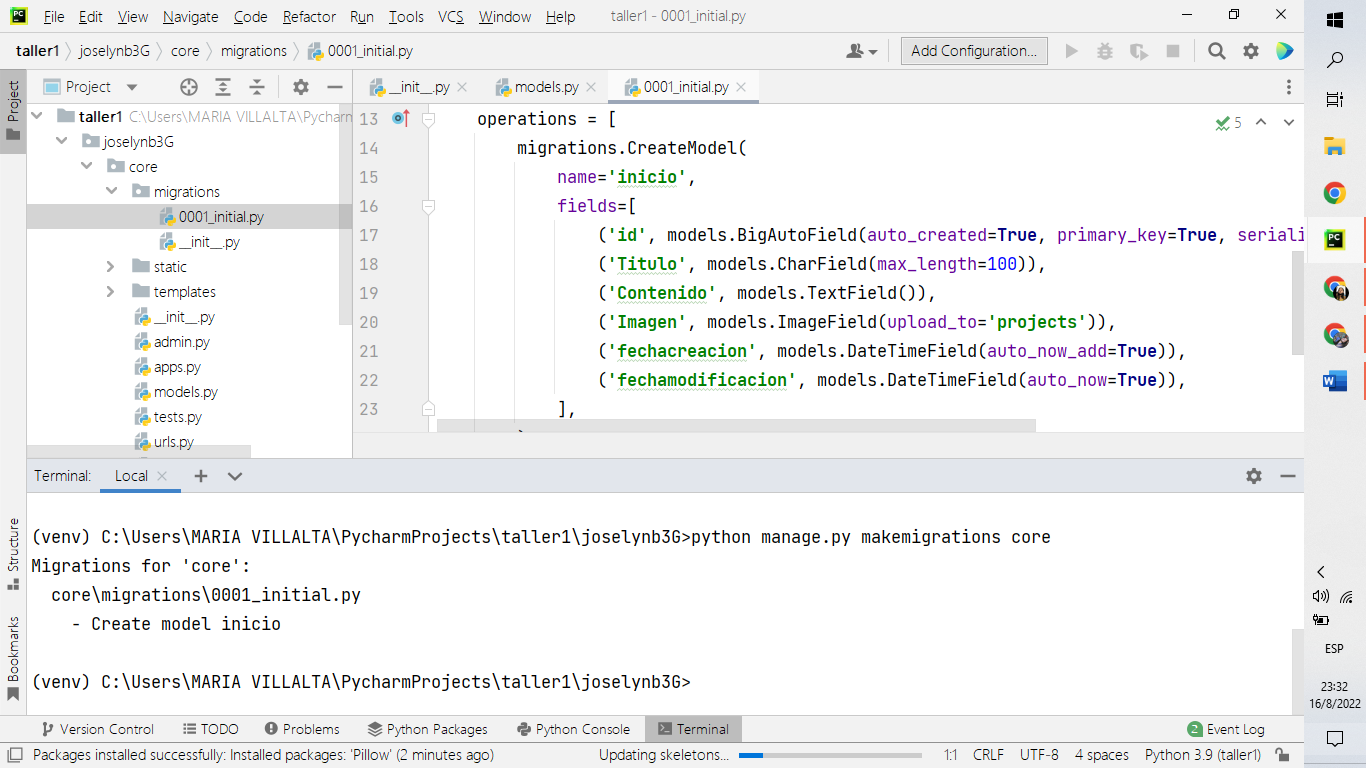
# Que es un modelo en Django.

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.

# Crear un modelo en Django.

Necesitamos crear una clase y agregarles los atributos.

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

Para migrar los modelos utilizamos el siguiente comando en la terminal:**python manage.py makemigrations core**

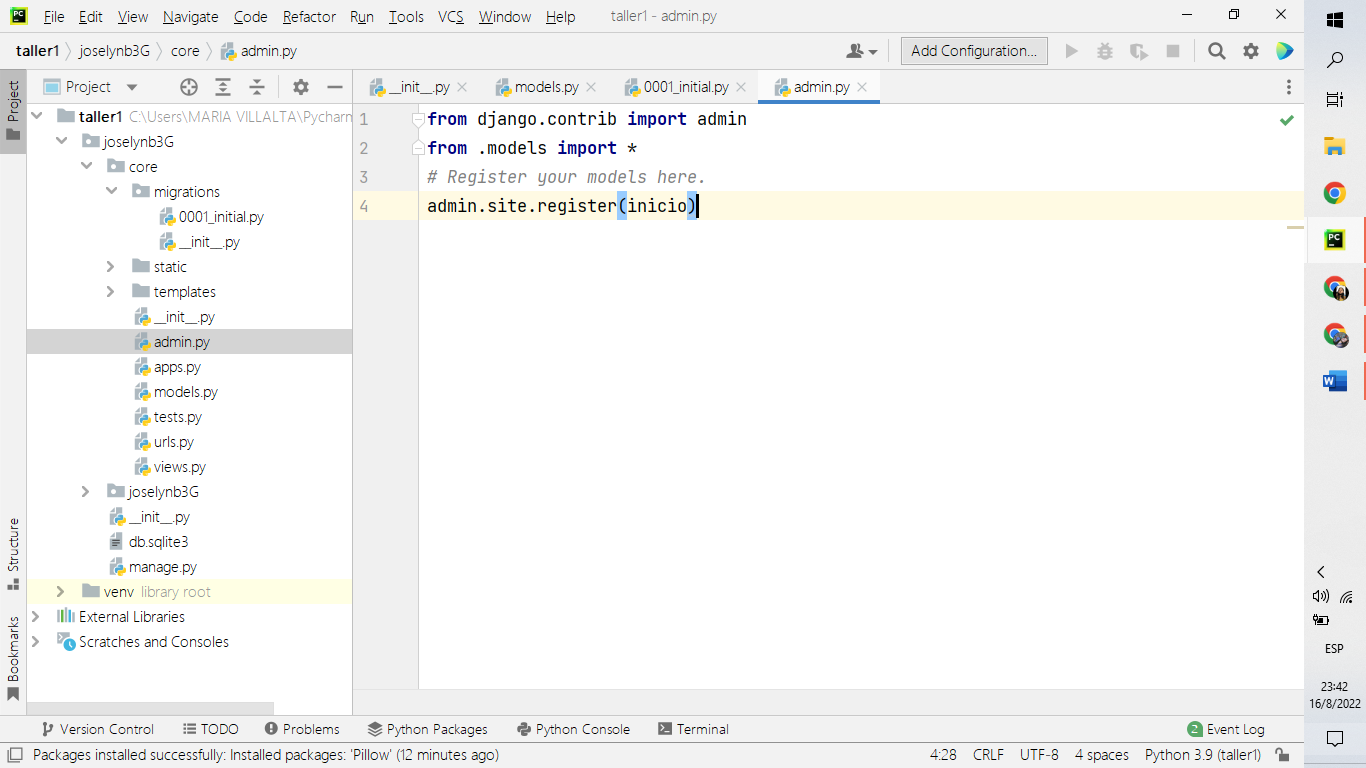
# Integrar el Modelo al Panel de Administración.

Para que el modelo lo reconozca nuestro panel de administración, debemos añadirlo a nuestro archivo admin.py

Primero importamos nuestro modelo con el comando:

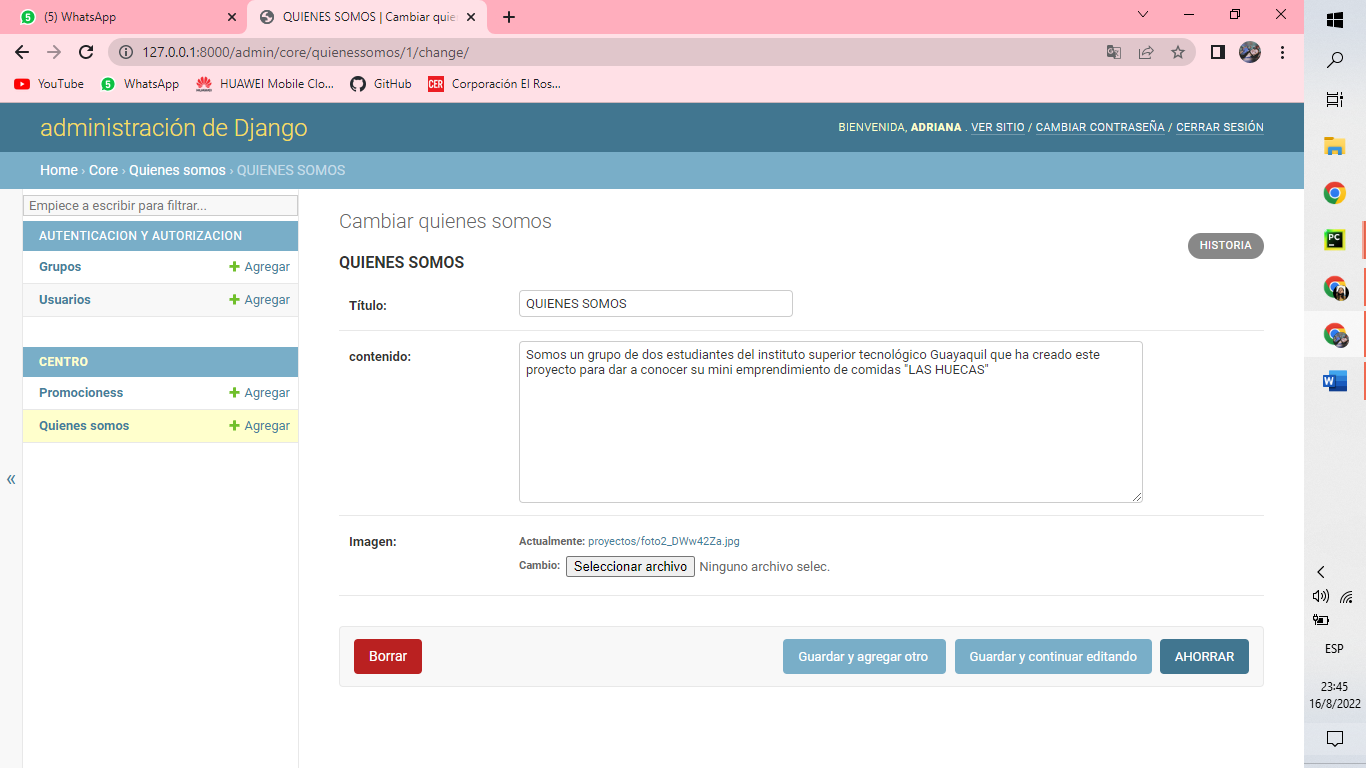
**from .models import \***

Luego utilizamos la siguiente sentencia y ponemos el nombre de nuestro modelo en el paréntesis:

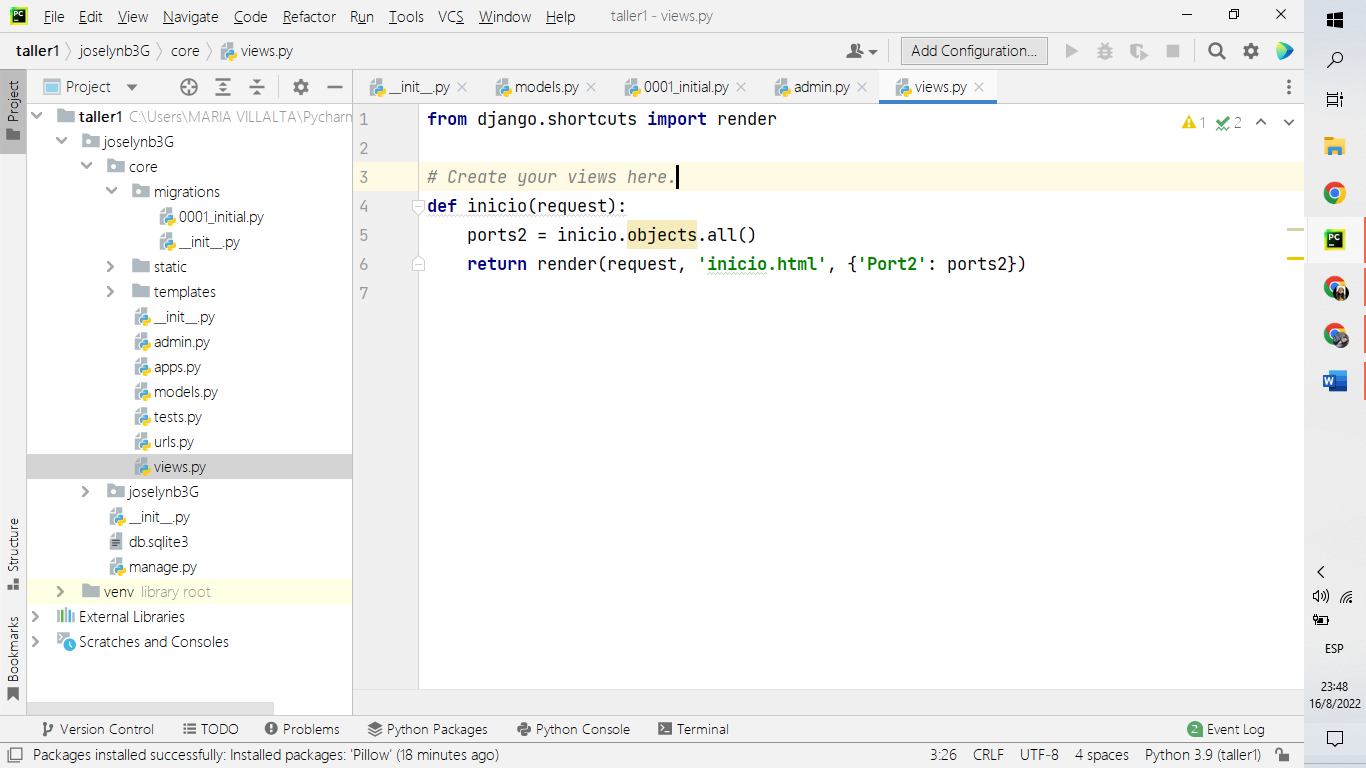
**admin.site.register(inicio)**

# Ingresar información al modelo por el Panel de Administración.

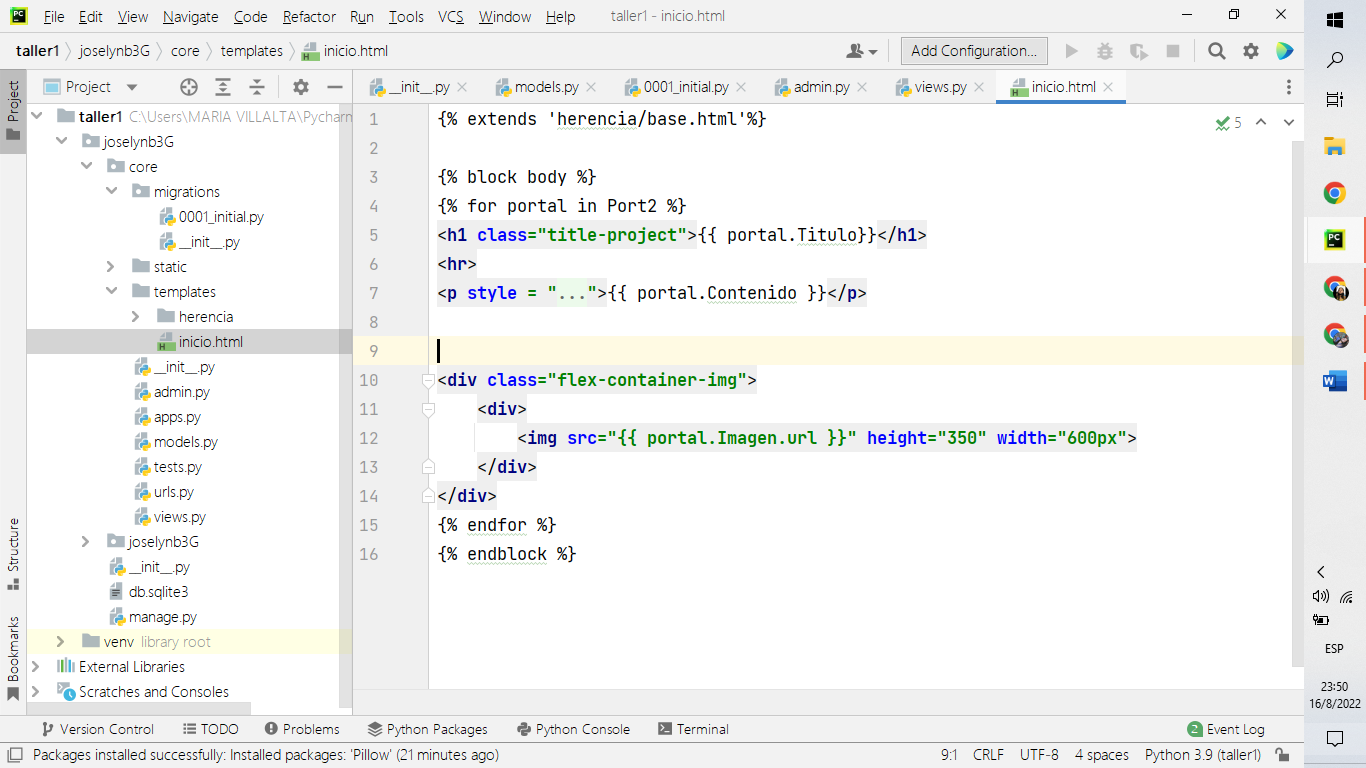
Con el usuario y contraseña, ingremos a nuestro admin y ingresamos información en el modelo.



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

Desde la views la podemos ver de la siguiente manera:

Para que se muestres en la página la agregamos también en el html:



# Mostrar los datos guardados en el modelo al html hijo.

Los datos ingresados se mostrarán de la siguiente manera: