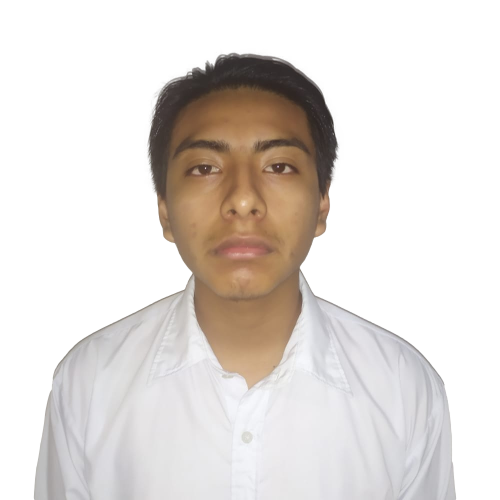
|  |
| --- |
| GIT-GITHUB  CARRERA: TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE  MATERIA: PROGRAMACION VISUAL  NOMBRE: BAQUE PILAY JEAN CARLOS  DOCENTE: CARLOS LUIS PAZMIÑO PALMA  NIVEL: 3 PARALELO: H  AÑO LECTIVO: 2022-2023 |
|  |
|  |

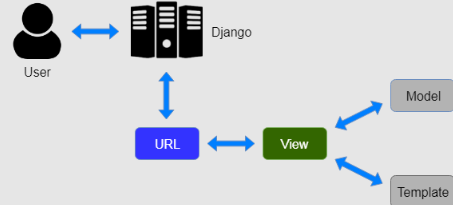
****



|  |  |
| --- | --- |
| Índice General | Pagina |
| 1) ¿Qué es Django? | 2 |
| 2) ¿Qué es la máquina virtual | 3 |
| 3) ¿Qué es MVT en Django? | 4 |
| 4) Crear un proyecto con la máquina virtual. | 5 |
| 5) Descargar los instaladores de Django al proyecto | 6 |
| 6) Crear un proyecto para programar en Django | 7 |
| 7) Ejecutar el proyecto y el mensaje de felicitaciones | 8 |
| 8) Crear una Apps core | 10 |
| 9) ¿Qué es la Carpeta Templates? | 14 |
| 10) ¿Qué es la Carpeta static? | 16 |
| 11) Crear un archivo base html en la APPS core | 17 |
| 12) Como se llaman a los CSS desde el archivo base html. | 18 |
| 13) Como consume un archivo hijo html al utilizar la herencia del archivo base html. | 19 |
| 14) Crear un view que llame al html hijo | 19 |
| 15) Crear la urls que llame al views. | 19 |
| 16) Integrar la aplicación APPS core al proyecto principal | 20 |
| 17) Crear las tablas del sistema de usuarios para utilizar el panel de administración. | 21 |
| 18) Crear un usuario para poder ingresar al Panel de Administración | 21 |
| 19) ¿Qué es un modelo en Django? | 23 |
| 20) Crear un modelo en Django. | 23 |
| 21) Migrar el Modelo a la base del Panel de Administración. | 24 |
| 22) Integrar el Modelo al Panel de Administración. | 25 |
| 23) Ingresar información al modelo por el Panel de Administración. | 26 |
| 24) Realizar la consulta de todo lo ingresado en el modelo desde el views. | 28 |
| 25) Mostrar los datos guardados en el modelo al html hijo. | 29 |

**¿Qué es Django?**Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.  
  
  
  
  
  
  
  
  
  
  
  
 **¿Qué es la máquina virtual en Django?**

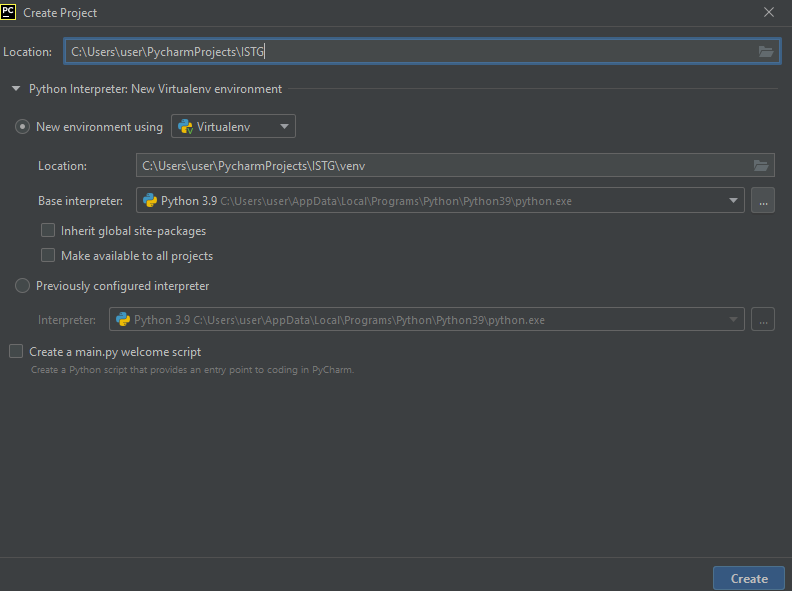
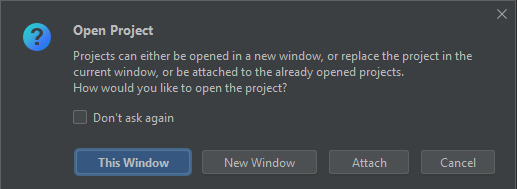
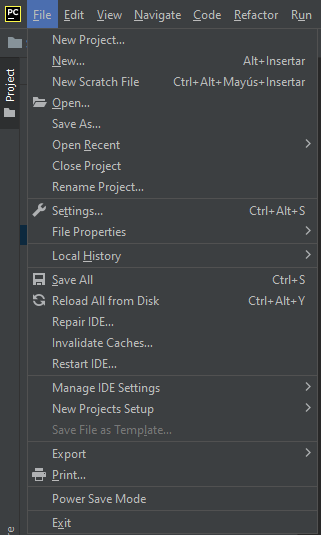
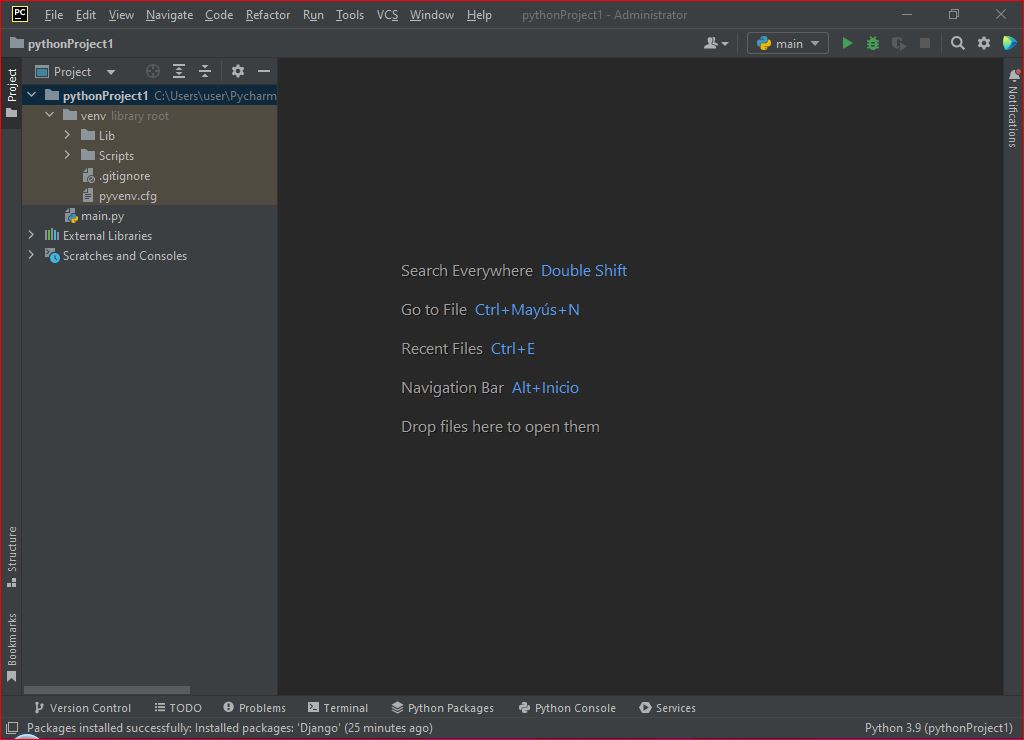
****El entorno virtual es el que nos permite que nuestro proyecto sea portable, eso significa que lo podemos trasladar sin problemas hace un nuevo computador y ahí va a estar toda la programación.El objetivo principal de los entornos virtuales es administrar la configuración y las dependencias de un proyecto en particular, independientemente de otros proyectos de python.

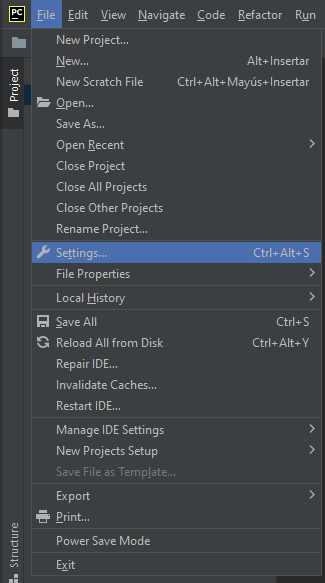
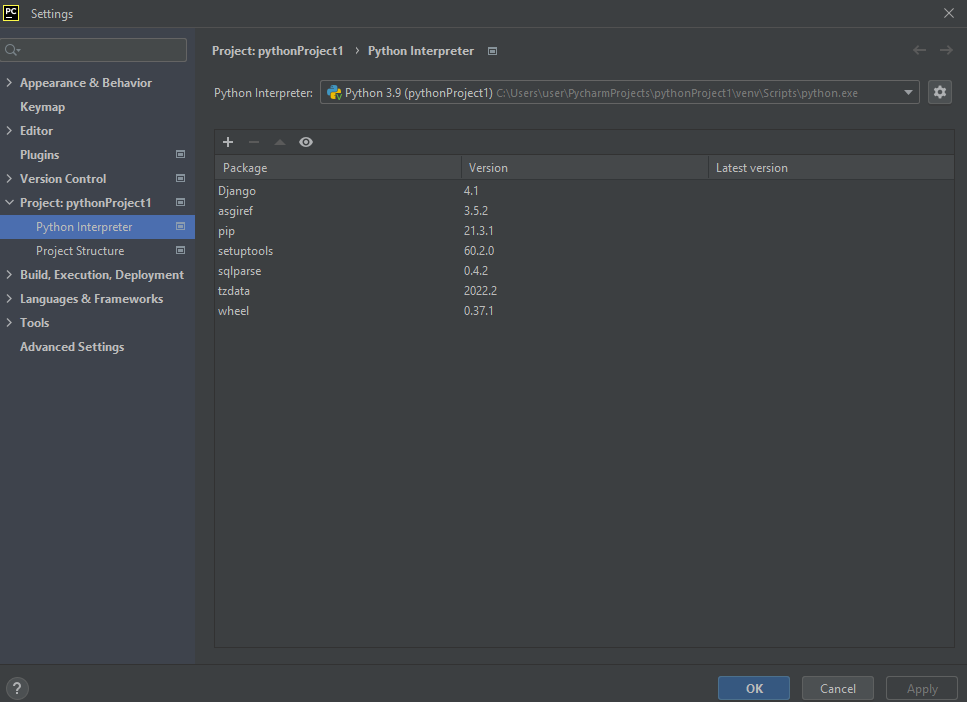
**¿Qué es MVT en Django?**Django se basa en la arquitectura MVT (Model-View-Template) . MVT es un patrón de diseño de software para desarrollar una aplicación web.

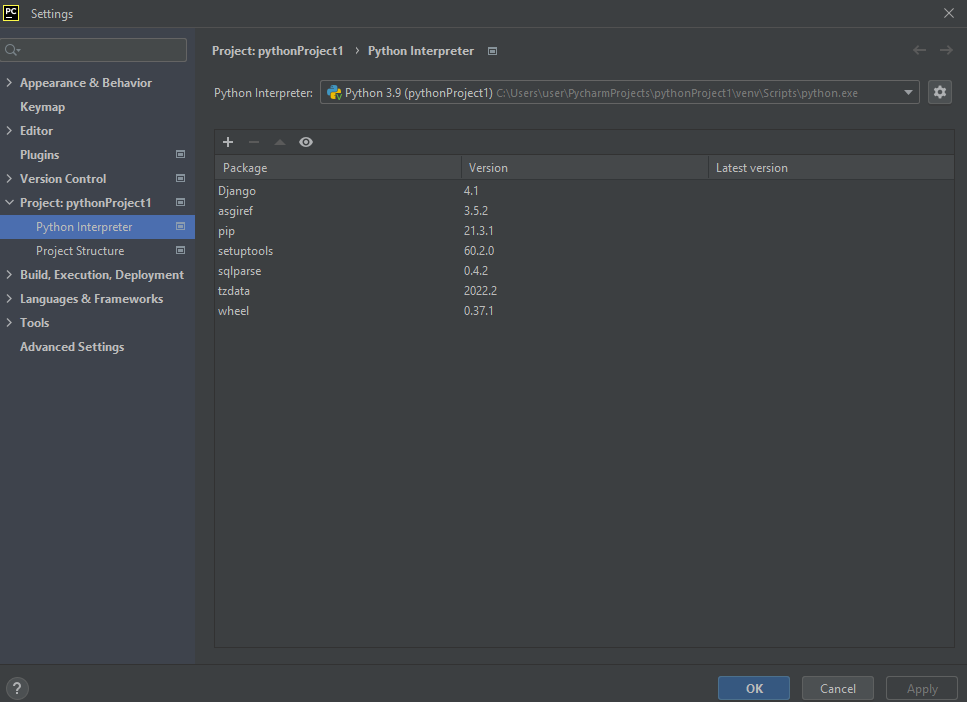
La estructura MVT tiene las siguientes tres partes:

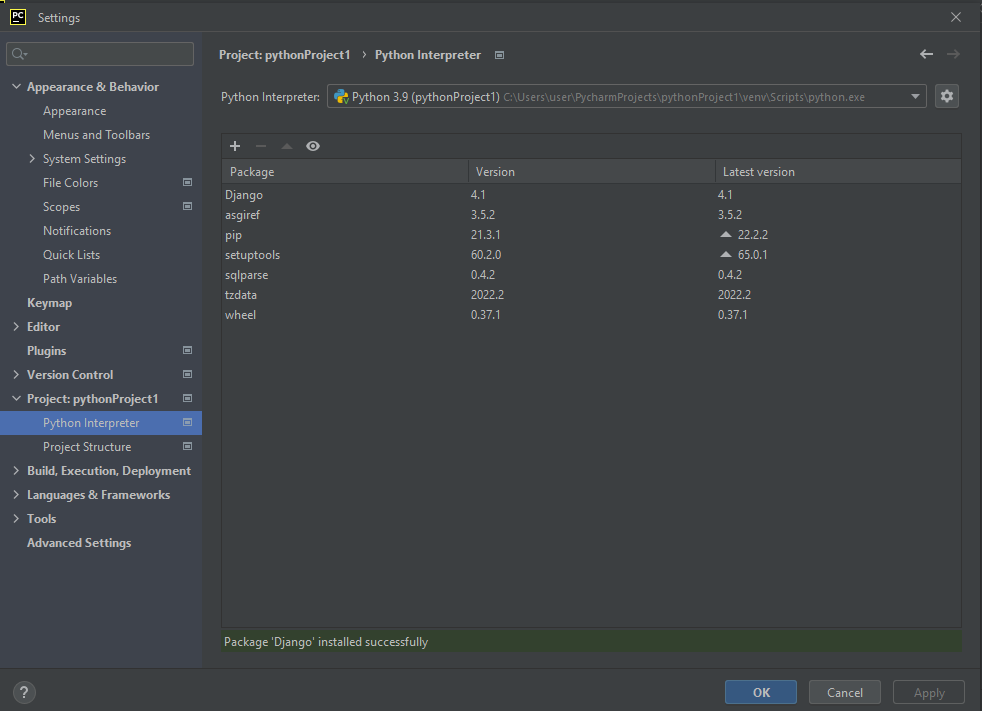
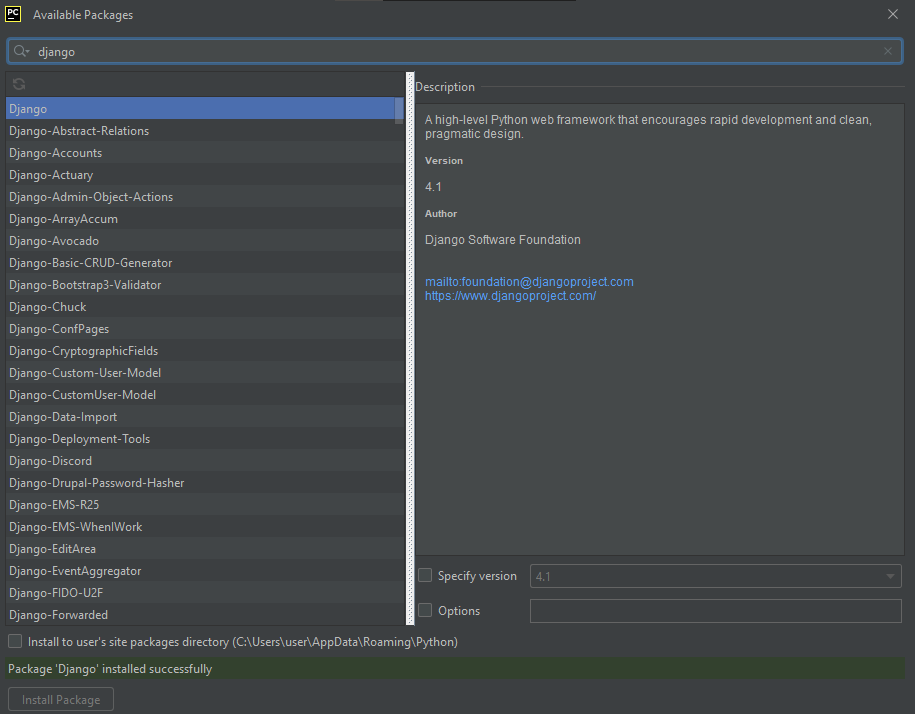
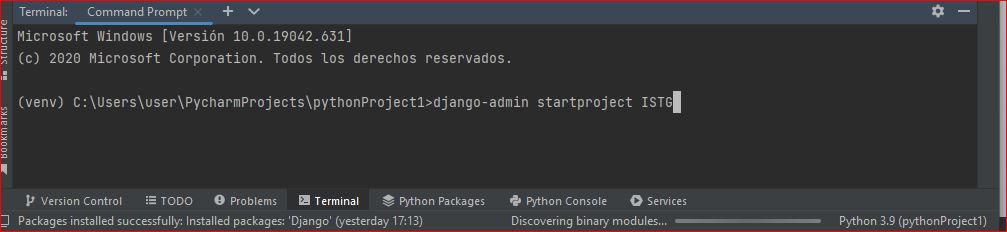
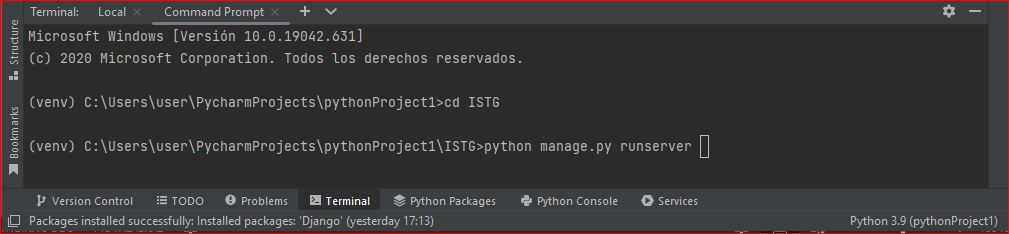
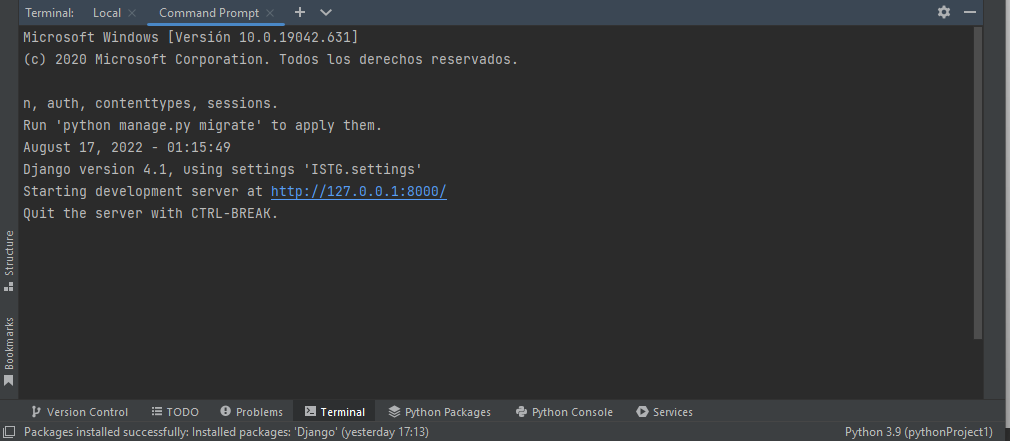
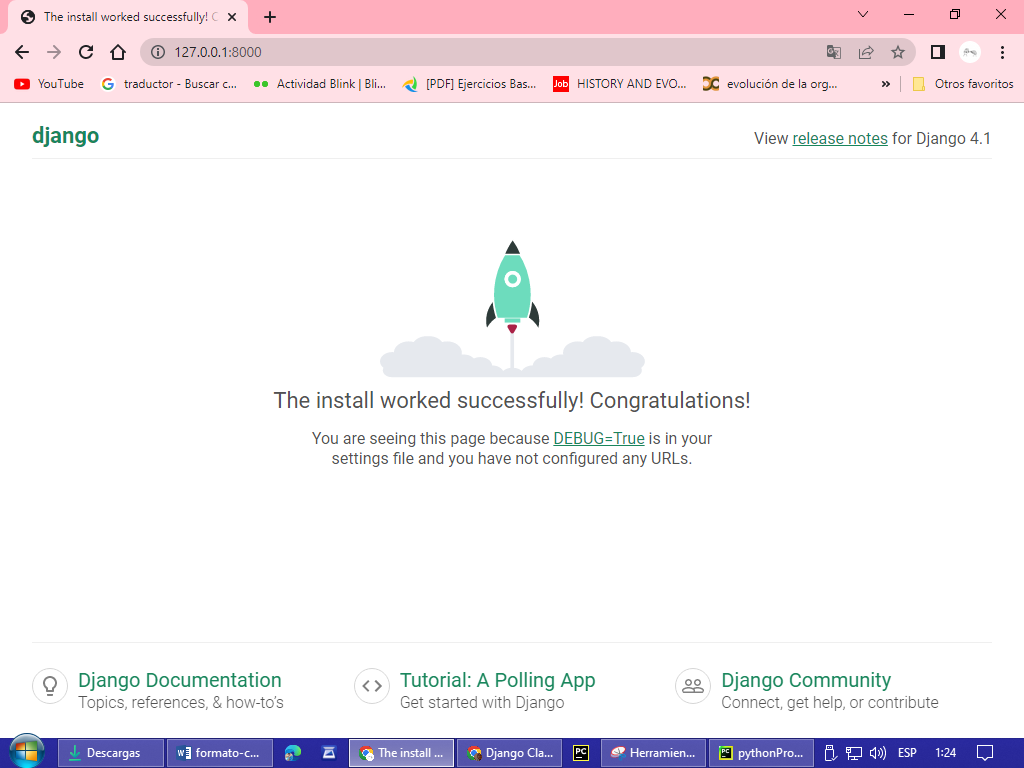
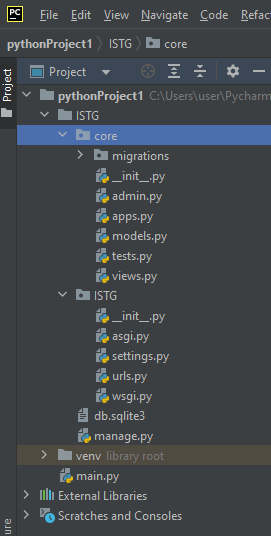
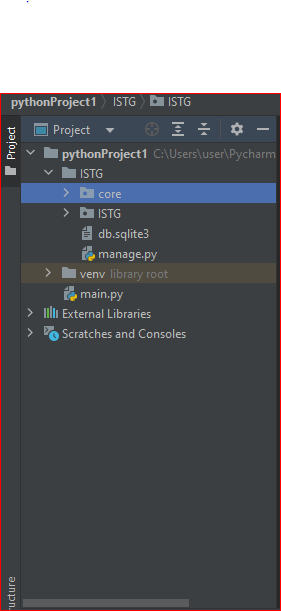
**Modelo**: El modelo actuará como la interfaz de sus datos. Es responsable de mantener los datos. Es la estructura de datos lógica detrás de toda la aplicación y está representada por una base de datos (generalmente bases de datos relacionales como MySql, Postgres). Para ver más, visite – Django Models.

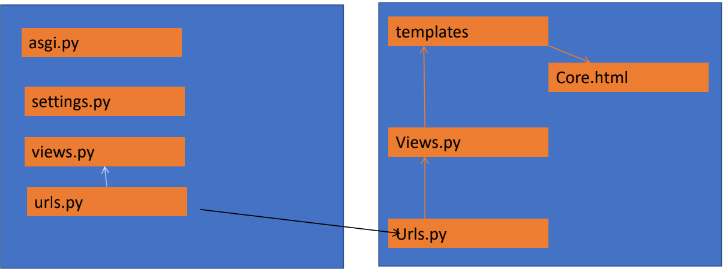
**Ver**: La vista es la interfaz de usuario: lo que ve en su navegador cuando muestra un sitio web. Está representado por archivos HTML/CSS/Javascript y Jinja. Para ver más, visite – Django Views.

**Plantilla**: una plantilla consta de partes estáticas de la salida HTML deseada, así como alguna sintaxis especial que describe cómo se insertará el contenido dinámico. Para ver más, visite – Plantillas de Django.  
  
  
 **Crea un proyecto con la máquina virtual (Pychram community)**1.Abrimos Pychram 2.Escribimos un nombre en mi caso fue ‘pythonproject1’   
community   
-Seleccionamos FILE -Clic en ‘Create’   
 clic en New proyect  
  
3.Clic en ‘New Window’  
  
4.Asi queda nuestro proyecto creado con Pychram community

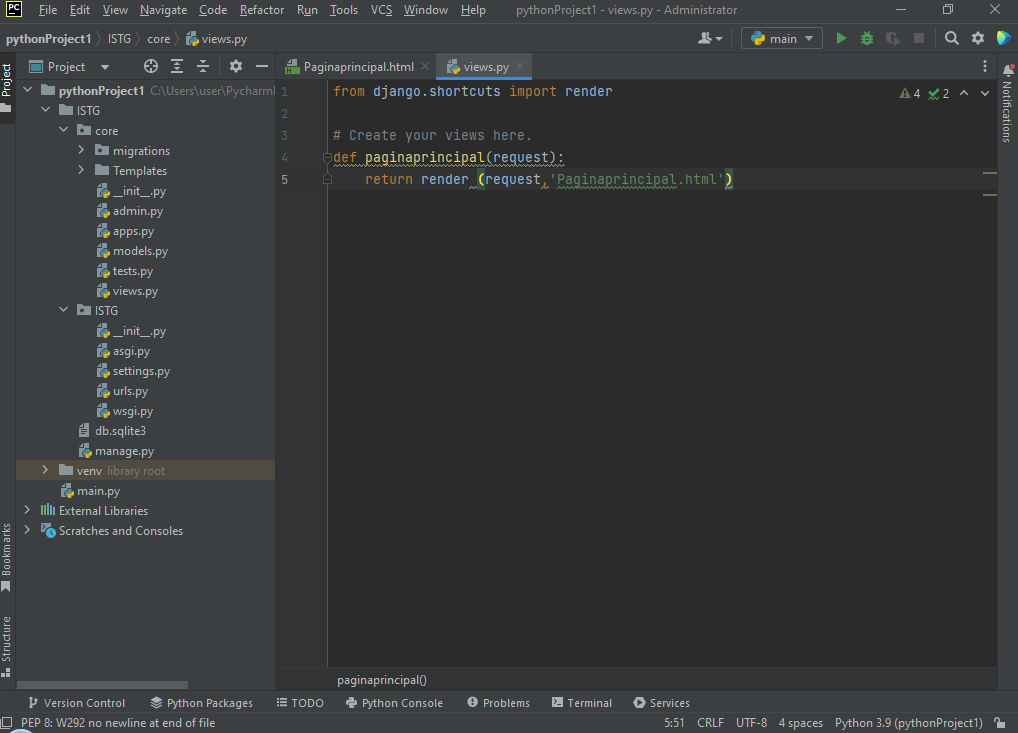
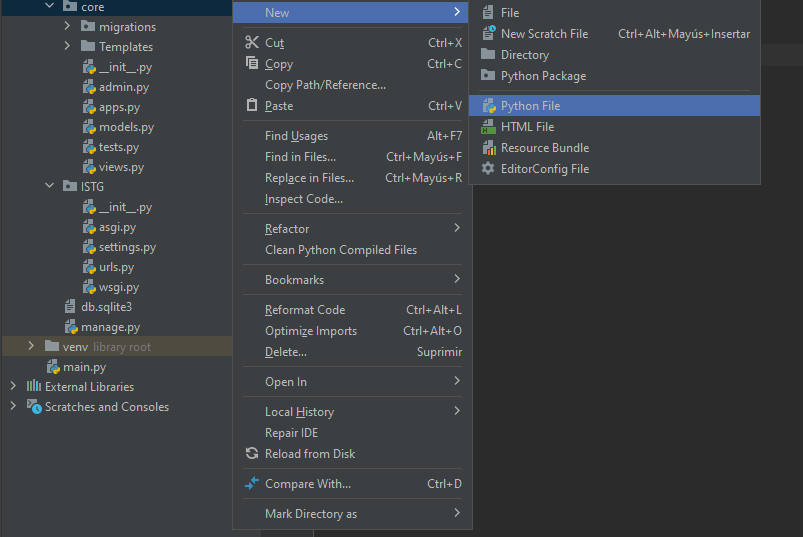
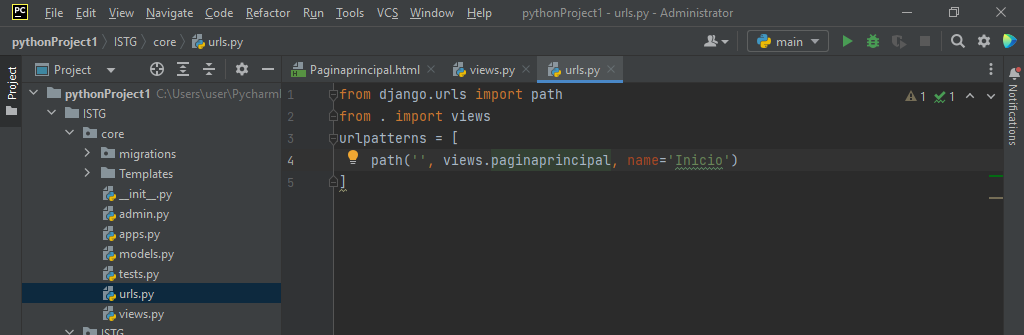
**Descargar los instaladores de Django al proyecto**  
-Seleccionamos ‘FILE’, luego ‘SETTINGS’ - Y buscamos ‘Pyhton interpreter’  
  


  
  
  
  
  
  
  
  
  
  
  
  
  
  
-Damoc clic en el signo’+’

-Esribimos en la lupa la palabra ‘Django’, y luego damos clic en ‘Install Package’  
  
-Al final verificamos que ya se instalo ‘Django’ y damos clic en ‘OK’ **Crear un proyecto para programar en Django**-Nos dirijimos al terminal y escribimos el comando:  
‘django-admin startproject **ISTG** ( Aqui puedes escribir el nombre que tu quieras para tu proyecto, en mi caso fue ISTG)’  
  
 **Ejecutar el proyecto y el mensaje de felicitaciones**  
-Ahora con dirijimos dentro de nuestro proyecto creado con un :  
‘cd **ISTG**(nombre de nuestro proyecto )’  
-Una vez dentro de nuestro proyecto  
-Escribimos el comando:  
‘manage.py runserver’ o ‘python manage.py runserver ’  
-Depende de cada computador o version de python  
  
  
  
  
  
  
  
  
 -Excelente ahora damos clic en el enlace que nos muestra   
-Asi nos muestra el mensaje de ‘CONGRATULATIONS! ’ felicitaciones  **Crear una Apss Core**-Para crear una Apss core no dirigimos a terminal y escribimos el siguiente código:  
****‘python manage.py startapp core ’  
****

  
  
-  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Debemos configuarar ‘CORE’ Y ‘ISTG’ ,como se muestra en la siguiente imagen   
**ISTG APPS CORE**

-Dentro de nuestro Proyecto Core, Modificamos el Views, llamamos a nuestro Html  
  
def holamundoCore(request):

return render(request,'core.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 (**ISTG**) Urls.py debemos llamar al Urls.py de nuestra  
apps Creada core de la siguiente manera:  
from django.contrib import admin

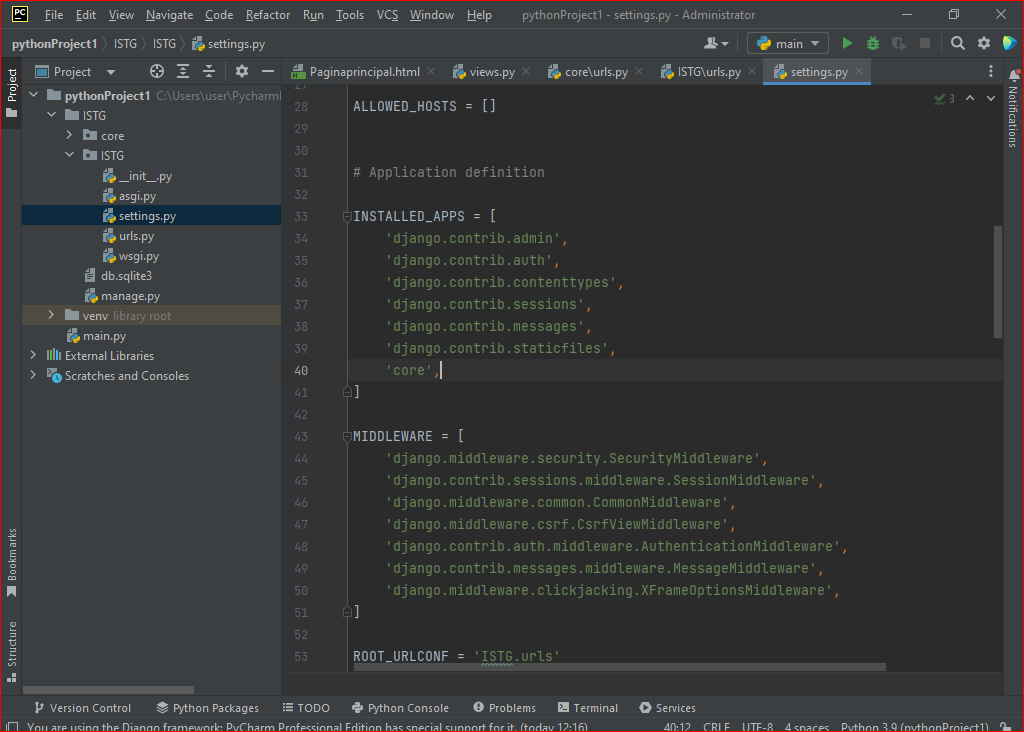
from django.urls import path,include

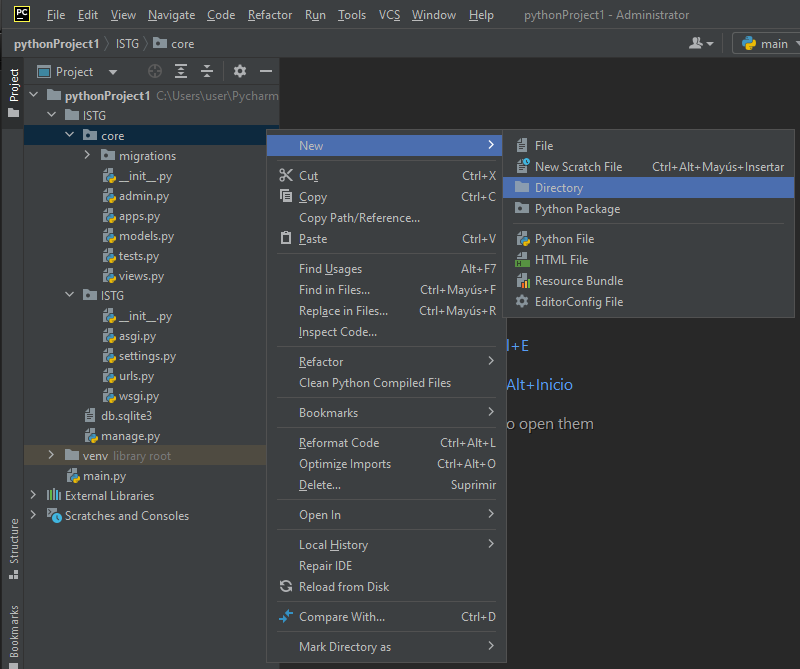
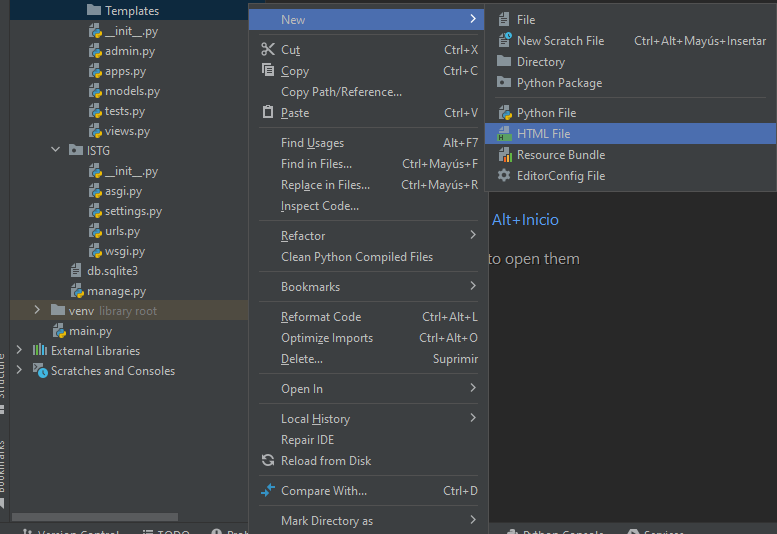
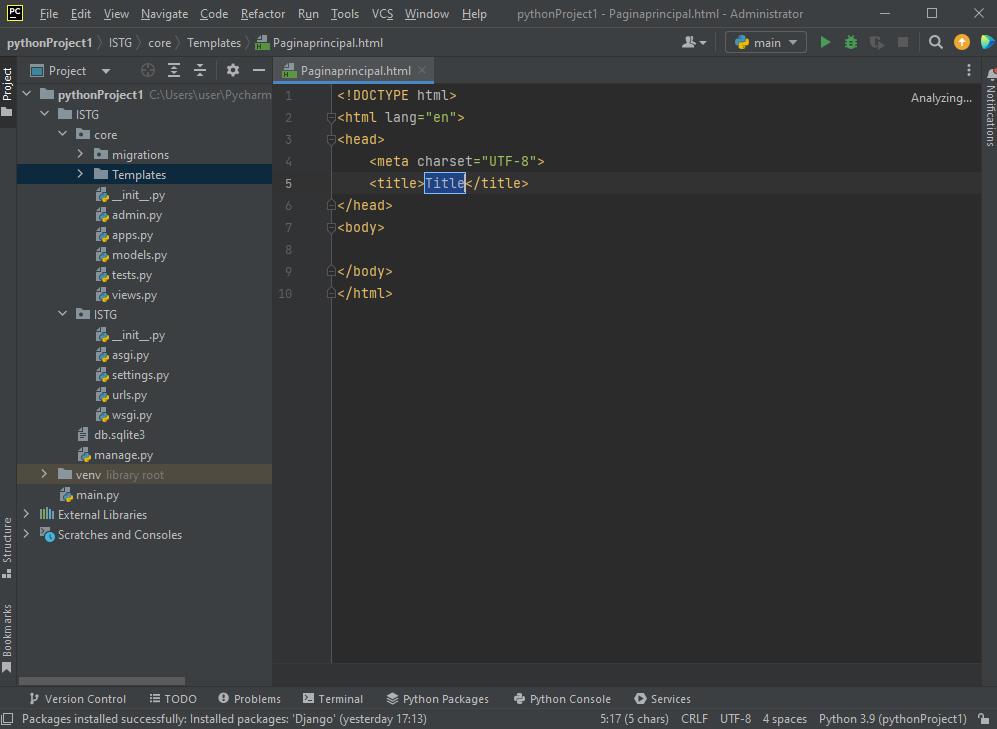
from . import views

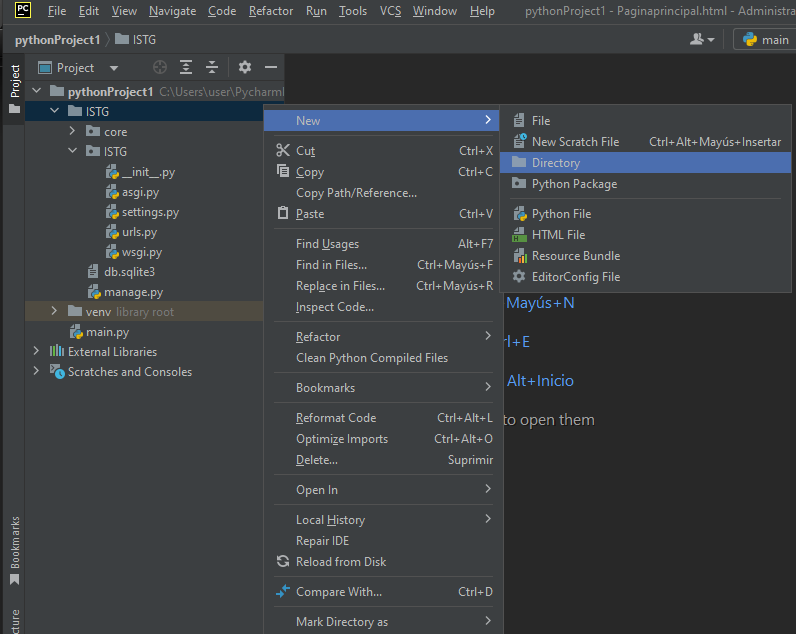
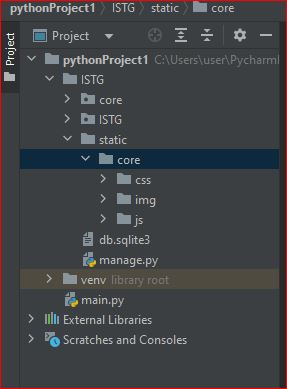
urlpatterns = [

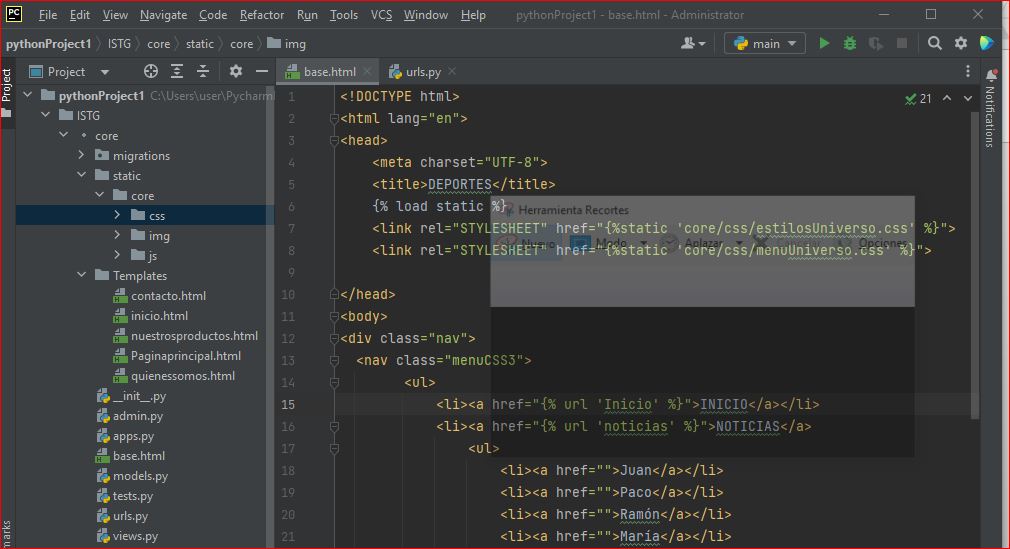
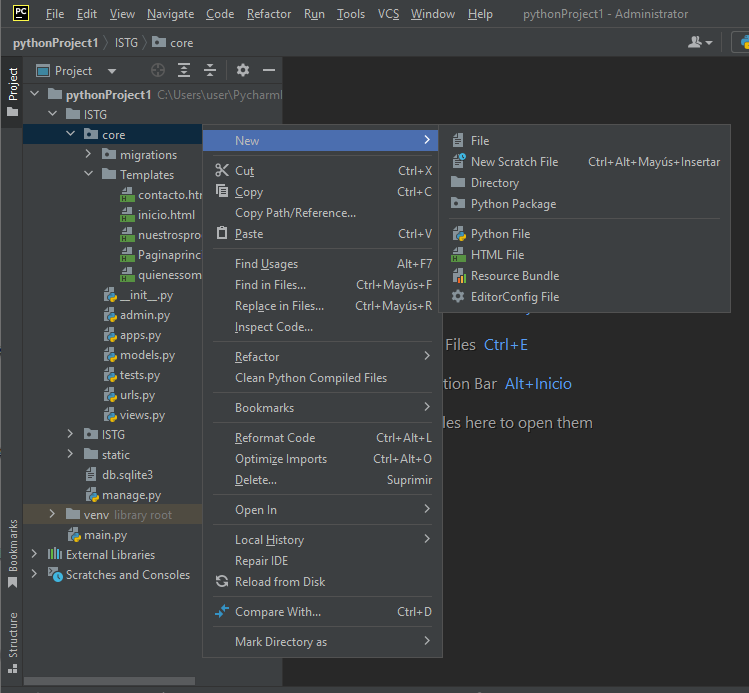
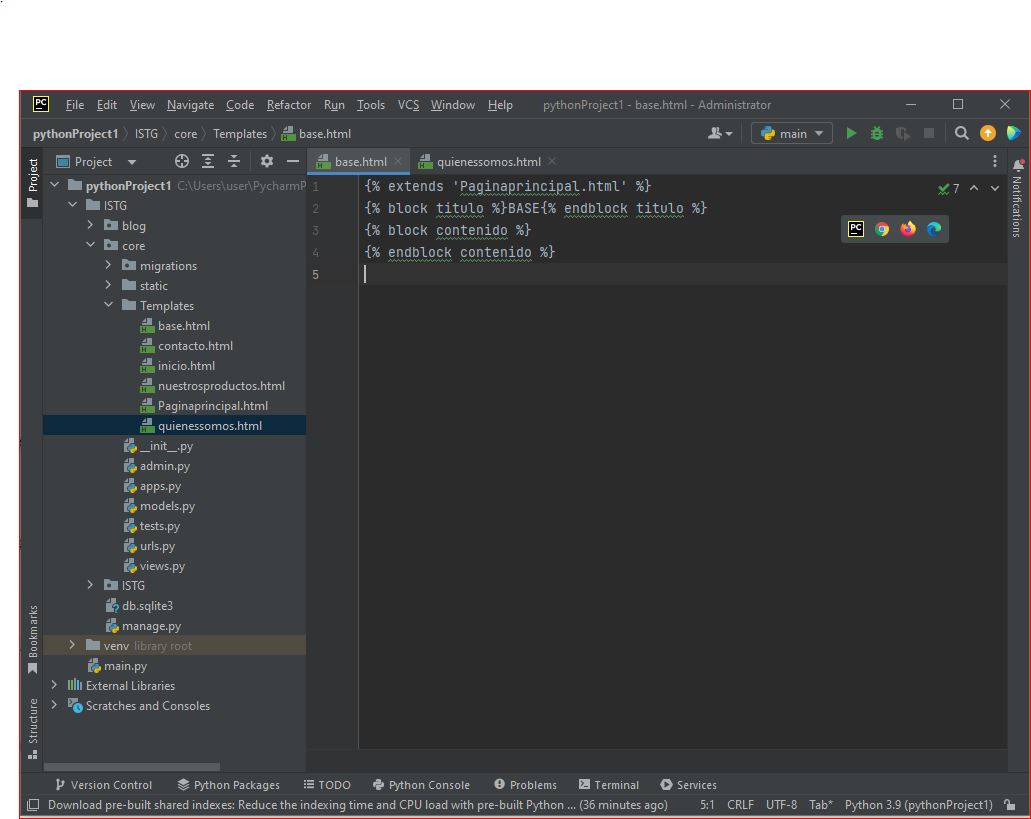
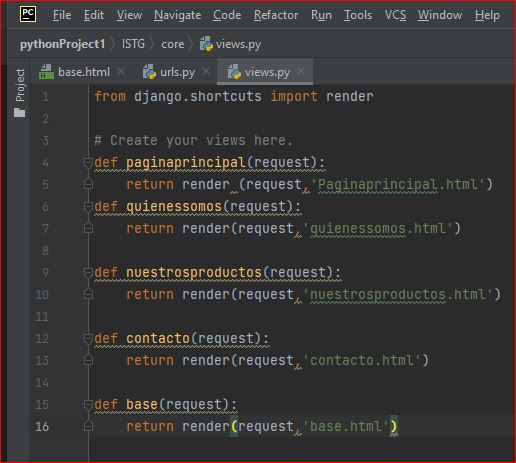
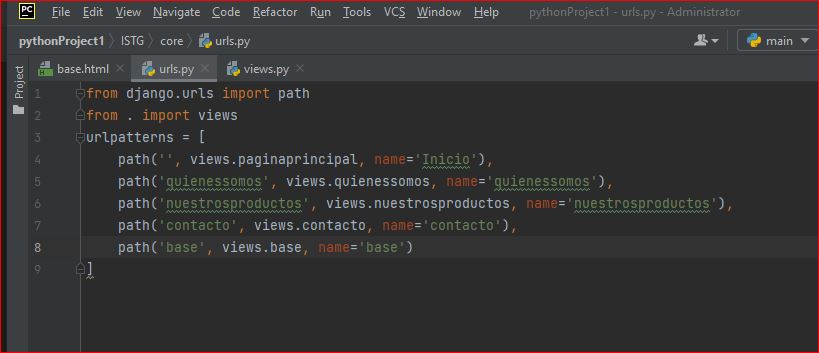
path('admin/', admin.site.urls),

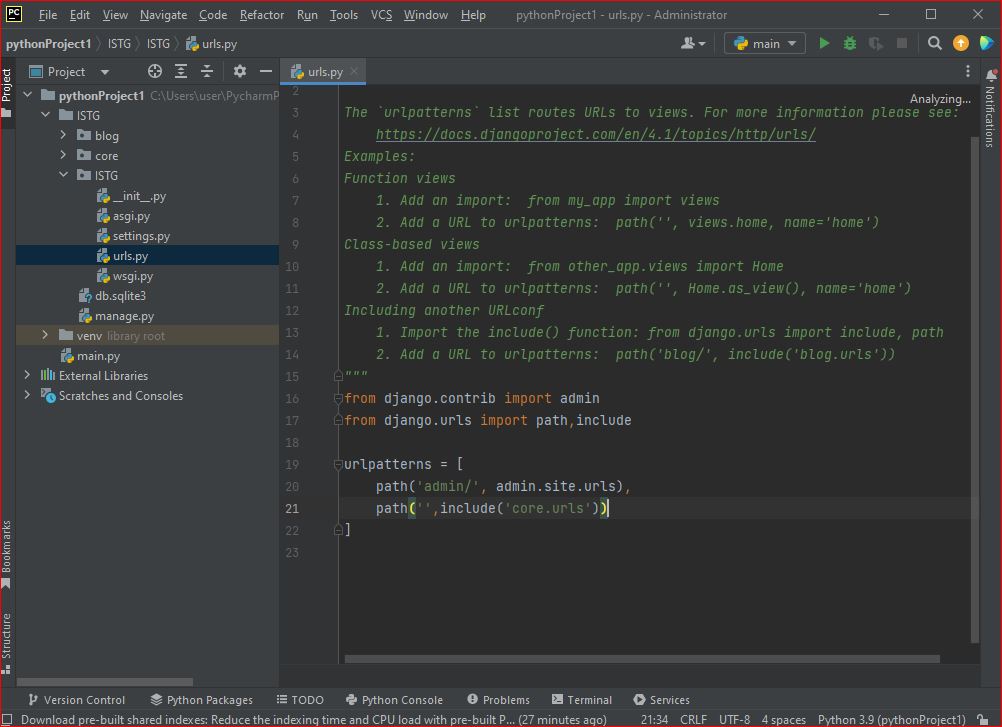
path(‘Inicio/', include('core.urls')),

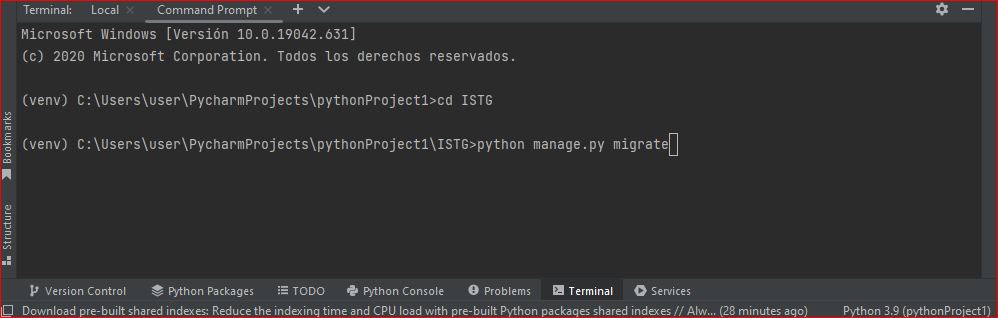
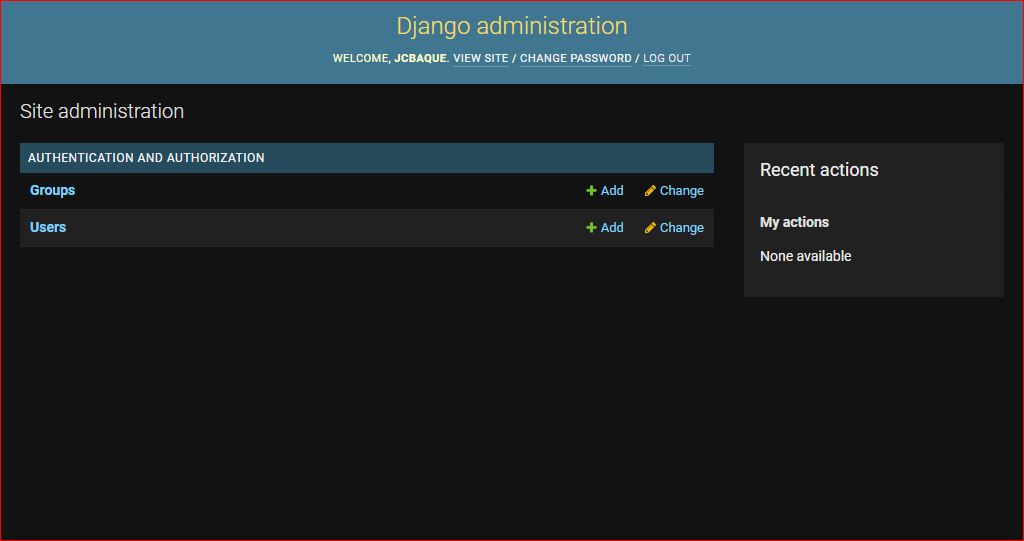
]  
  
-Ahora desde nuestro proyecto principal (**ISTG**), nos dirigimos a ‘settings’ , y escribimos ‘core’ y guardamos.  


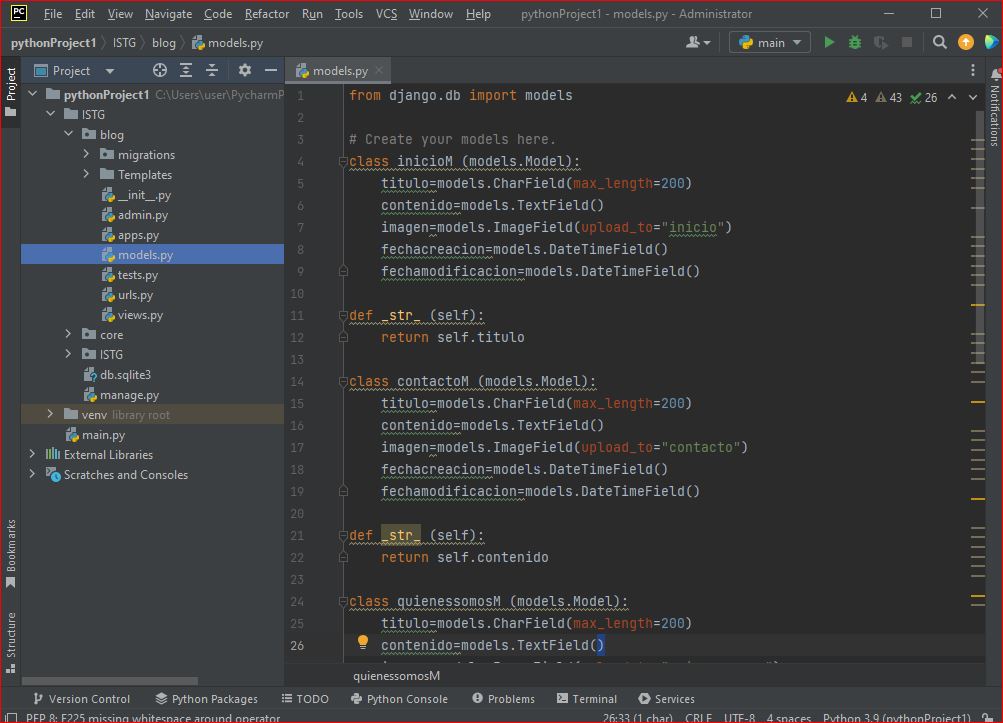
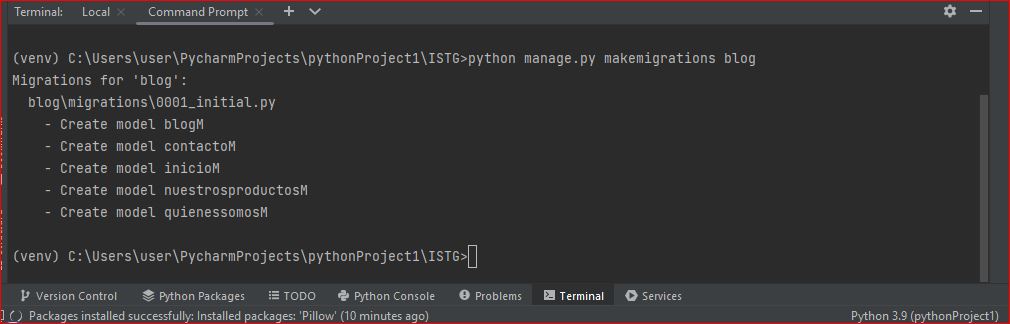
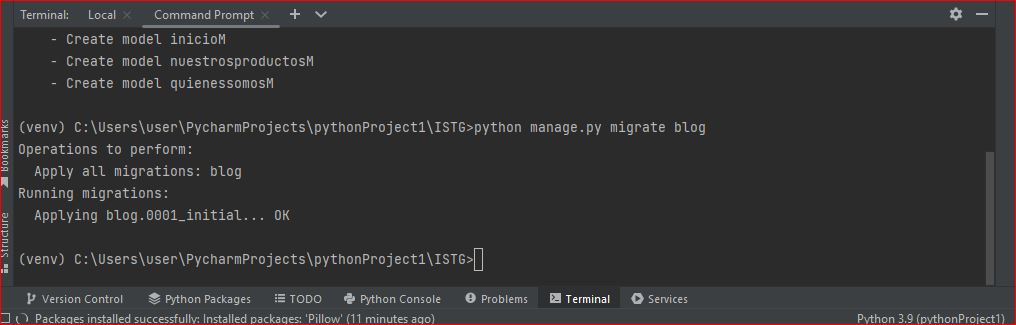
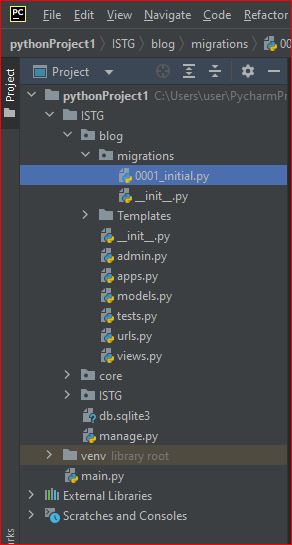
**¿Qué es la carpeta ‘Templates’?**Una plantilla es un archivo de texto que determina la estructura o diseño de un archivo (como una página HTML), con marcadores usados para representar el contenido real. Django automaticamente buscará plantillas en un directorio llamado 'templates' de su aplicación  
-Desde ‘CORE’, ‘NEW’, ‘DIRECTORY’ , y como nombre ‘Templates’.  
  
  
   
-Desde el directory ‘Templates’, ‘NEW’, ‘HTML FILE’ , con el nombre (que ustedes prefieran en mi caso fue PaginaPrincipal’)  
  
  
  
  


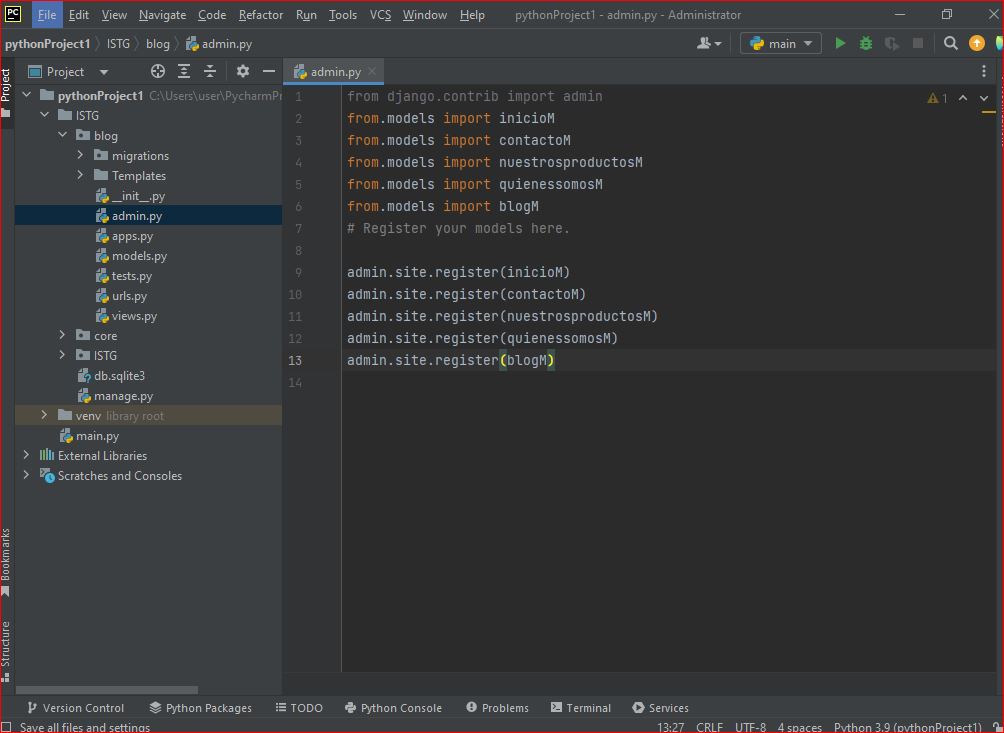
**¿Qué es la carpeta ‘Static’?**Cuando se desarrolla una aplicación web se manejan recursos estáticos como logos, videos, archivos javascript y css ,los cuales se agregan directamente a las plantillas html generadas, estos, normalmente son almacenados en un directorio llamado “Static”.  
-Desde nuestro proyecto principal (ISTG), ‘NEW’,’DIRECTORY’, con nombre ‘Static’  
  
-Dentro de ‘STATIC’,’NEW DIRECTORY’, como nombre ‘CORE’ desde ‘CORE’, ’NEW’,’DIRECTORY’,como nombre’CSS’, colocamos los css que necesitamos   
desde ‘CORE’, ’NEW’,’DIRECTORY’,como nombre’IMG’, colocamos las imagenes que necesitemos  
desde ‘CORE’, ’NEW’,’DIRECTORY’,como nombre’JS’, colocamos los javascript que necesitemos

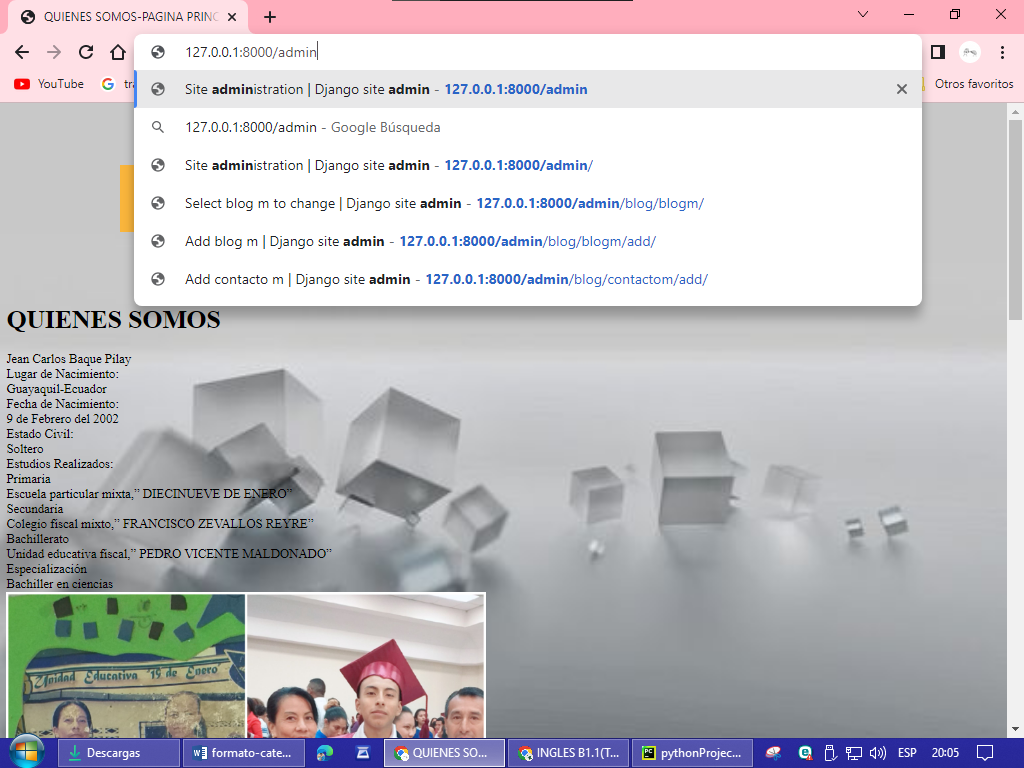
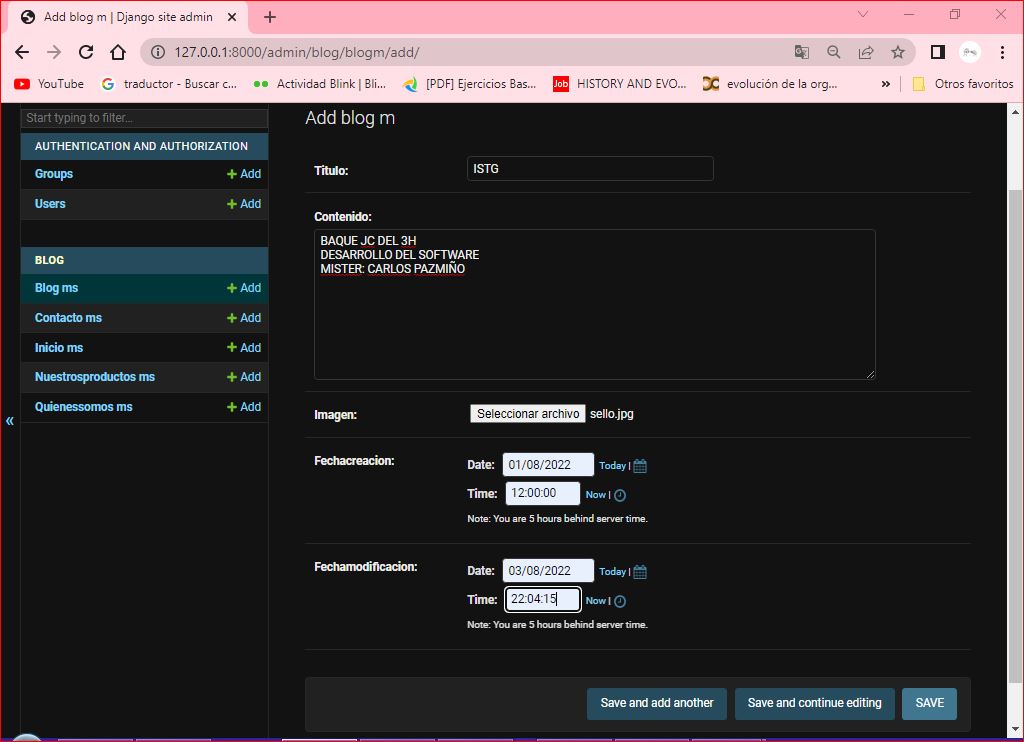
**Crea un archivo base.html en la Apps Core**-Desde core , new, directory,html como nombre base  
  
  
**Como se llaman los css desde el archivo base html  
  
Como consume un archivo html al utilizar la herencia del archivo base.html  
  
  
  
  
Crear un view que llame al html hijo  
****Crear las urls que llame al views**

 **Integrar las Apps core al proyecto principal**

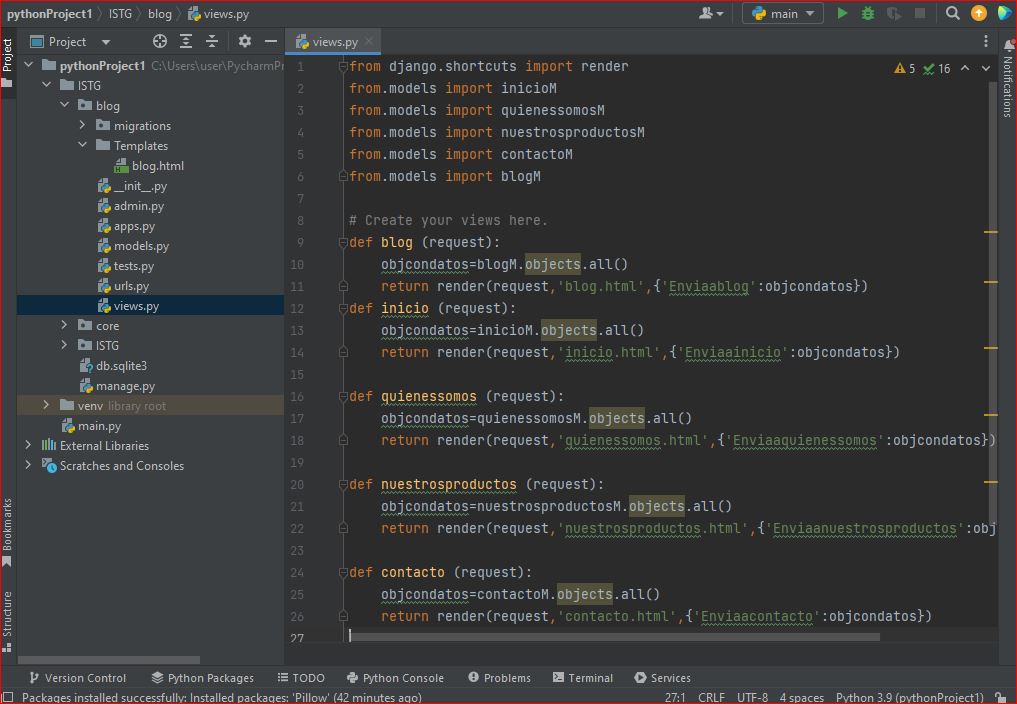
**Crear las tablas del sistema de usuarios para utilizar el panel de administración.**-Para crear las tablas del sistema de usuarios primero debes migrar las tablas todas las tablas que por defecto tiene Django.  
 -Desde terminal ejecutamos el código:  
“python manage.py migrate”  
  
  
  
  
**Crear un usuario para poder ingresar al panel de administración**-Luego desde el mismo terminal escribimos el código:  
“python manage.py createsuperuser”  
-Nos pedirá un username, un email y un password  
-Levantamos el servidor con el codigo: “python manage.py runserver ”  


**¿Qué Es un modelo en Django?**Los modelos están definidos, normalmente, en el archivo models.py de la aplicación. Son implementados como subclases de django.db.models.Model, y pueden incluir campos, métodos y metadata. El fragmento de código más abajo muestra un modelo "típico"  
**Crear un modelo en Django**  
-Para crear un modelo en Django lo hacemos por ‘models.py’, creando los diferentes modelos   
ejemplo:  
class **contactom** (el nombre que queramos, en mi caso fue contectom) (models.model):  
 titulo=models.charfield(max\_length=200)  
 contenido=models.textfield( )  
 imagen=models.Imagefield(upload\_to=”contacto”)  
 fechacreacion=models.Datetimefiels()  
 fechamodificacion=models.Datetimefiel()  
  
def \_str\_ (self):  
 return self.contenido  
**Migrar el modelo a la base del panel administración**-Para migrar al panel de administración debemos escribir el siguiente código en la terminal:  
‘python manage.py makemigrations blog’  
****  
****

**Integrar el modelo por el panel de administración** -Desde ‘admin.py’ from.models e importamos las clase del models de la pagina#23  
from.models import iniciom   
from.models import **contactom**   
from.models import nuestrosproductosm   
from.models import quienessomosm   
from.models import blogm   
  
admin.site.register(iniciom)  
admin.site.register(contactom)  
admin.site.register(nuestrosproductosm)  
admin.site.register(quienessomosm)  
admin.site.register(blogm)

**Ingresar información al modelo por el panel de administración.**-Levantamos el servidor desde terminal con:  
‘python manage.py runserver’  
-Luego escribimos’/admin’  
  
  
  
  
  
  
  
  
  
  
  
  
  


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



**Mostar los datos guardados en el modelo al html hijo.**