



Instituto Superior Tecnológico
GUAYAQUIL



Instituto Superior Tecnológico
GUAYAQUIL

Carrera:

Tecnología Superior en Desarrollo de Software

Alumno:

Adonis Alexander Merchan Guadamud

Paralelo:

3G

Docente

Carlos Luis Pazmiño Palma

Foto del estudiante:





Contenido

¿Qué es Django?	3
¿Qué es la máquina virtual en Django?	3
¿Qué es MVT en Django?	3
¿Qué es la Carpeta Templates?	3
¿Qué es la Carpeta static?	4
¿Cómo se llaman a los CSS desde el archivo base html?.....	4
¿Cómo consume un archivo hijo html al utilizar la herencia del archivo base html? ..	4
¿Qué es un modelo en Django?	5
Crear un proyecto con la máquina virtual.	5
Descargar los instaladores de Django al proyecto.....	6
Crear un proyecto para programar en Django.....	6
Ejecutar el proyecto y el mensaje de felicitaciones.	7
Crear una Apps core.	7
Crear un archivo base html en la APPS core	8
Crear un view que llame al html hijo.....	8
Crear la urls que llame al views.	9
Integrar la aplicación APPS core al proyecto principal.....	9
Crear las tablas del sistema de usuarios para utilizar el panel de administración. ...	10
Crear un usuario para poder ingresar al Panel de Administración	10
Crear un modelo en Django.....	11
Migrar el Modelo a la base del Panel de Administración.....	11
Integrar el Modelo al Panel de Administración.	12
Ingresar información al modelo por el Panel de Administración.....	12
Realizar la consulta de todo lo ingresado en el modelo desde el views.....	13
Mostrar los datos guardados en el modelo al html hijo.	13

Los temas a desarrollar son los siguientes:

¿Qué es Django?

Django es un framework de desarrollo web de código abierto, escrito en Python, soporta el desarrollo dinámico de sitios web, aplicaciones y servicios. Provee una serie de herramientas y funcionalidades que resuelven muchos problemas comunes asociados con el desarrollo web, tales como características de seguridad, acceso a bases de datos, sesiones, procesamiento de plantillas, ruteo de URLs, internacionalización, ubicación y mucho más.

Respeto el patrón de diseño conocido como modelo-vista-controlador (MVC).

¿Qué es la máquina virtual en Django?

Usar entornos virtuales no es obligatorio, pero es muy recomendado. Cuando se desarrollan sitios web o proyectos con Django, es muy común tener que instalar librerías externas para apoyar el desarrollo. Usando entornos virtuales, cada proyecto que desarrolles tiene su ambiente aislado, así que las dependencias no harán conflictos, y permite mantener en un ambiente local de proyectos que se ejecutan con distintas versiones de Django.

¿Qué es MVT en Django?

MVC: Modelo-Vista-Controlador. Django redefine este modelo como MVT: Modelo-Vista-Template.

¿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. 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.

¿Qué es la Carpeta static?

Esta carpeta contiene todos los archivos de estilos: css y scss, archivos de comportamientos JavaScript, imágenes incluidas dentro del theme y el archivo checkout.

¿Cómo se llaman a los CSS desde el archivo base html?

Asegúrese de que `django.contrib.staticfiles` esté incluido en su archivo `INSTALLED_APPS`.

En su archivo de configuración, defina `STATIC_URL`, por ejemplo:

```
STATIC_URL = 'static/'
```

En sus plantillas, use la `static` etiqueta de plantilla para crear la URL para la ruta relativa dada usando el archivo `STATICFILES_STORAGE`.

```
{% load static %}
```

```
<link rel="stylesheet" href="{% static 'css/estilo.css' %}">
```

Almacene sus archivos estáticos en una carpeta llamada `static` en su aplicación. `app/static/css/estilo.css` por ejemplo

¿Cómo consume un archivo hijo html al utilizar la herencia del archivo base html?

Una forma clásica de solucionar este problema es usar includes, insertando dentro de las páginas HTML a "incluir" una página dentro de otra. Es más, Django admite esta aproximación, con la etiqueta `{% include %}` anteriormente descrita. Pero la mejor forma de solucionar este problema con Django es usar una estrategia más elegante llamada herencia de plantillas.

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.

Usamos una etiqueta de plantilla aquí que no hemos visto antes: la etiqueta `{% block %}`. Todas las etiquetas `{% block %}` le indican al motor de plantillas que una plantilla hijo quizás sobrescriba esa porción de la plantilla.

Ejemplo:

```
{% extends "base.html" %}

{% block title %} Título {% endblock %}

{% block content %}

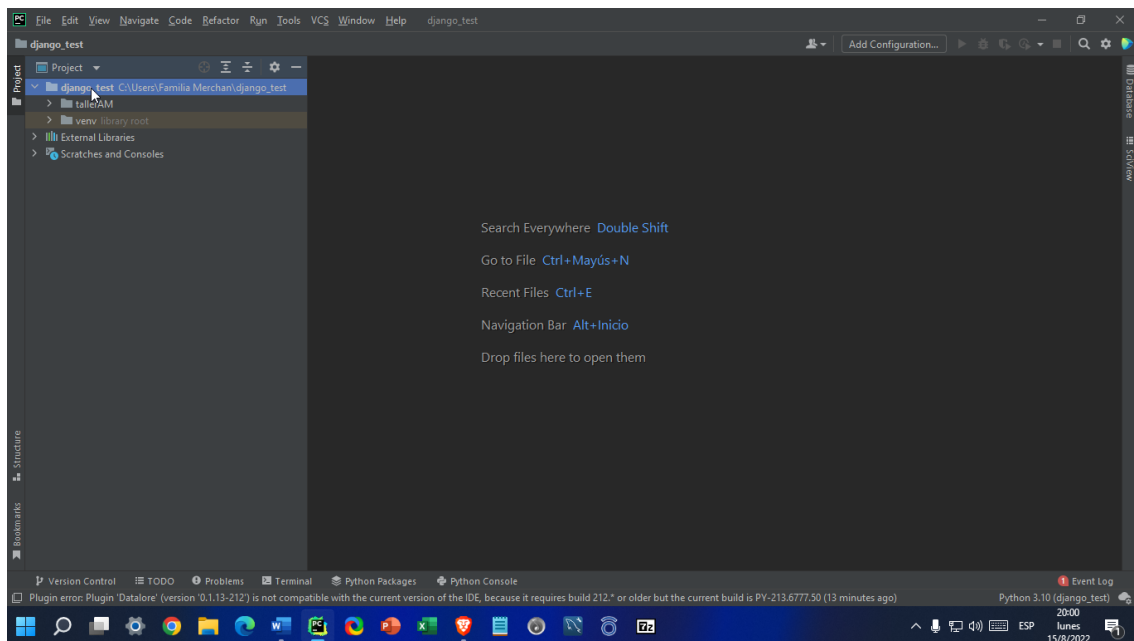
<p>Informacion.</p>

{% endblock %}
```

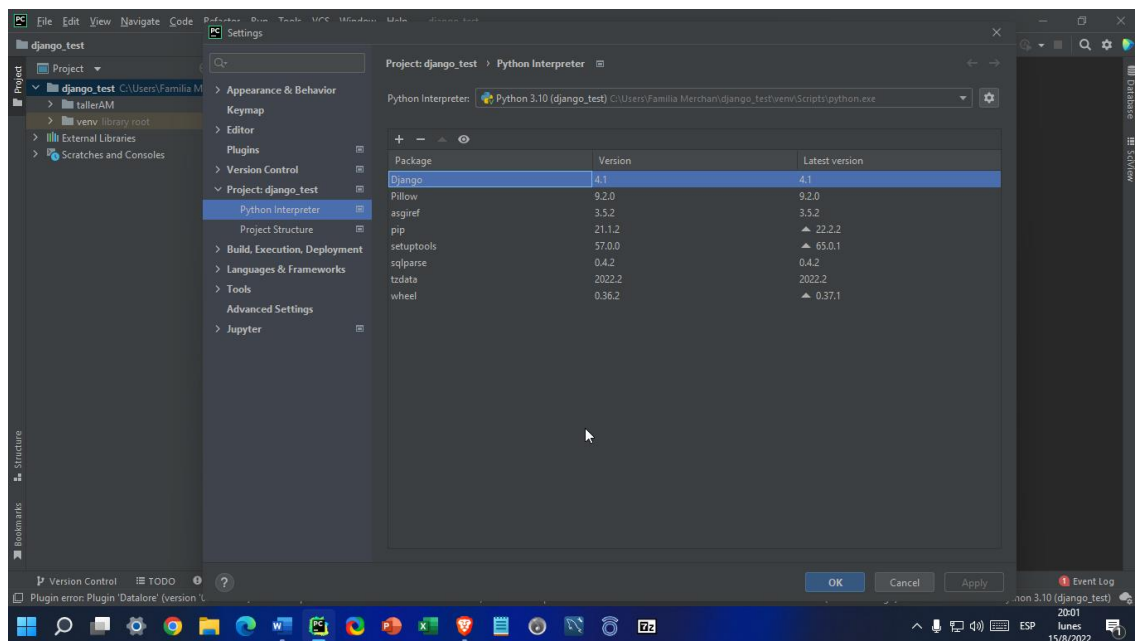
¿Qué es un modelo en Django?

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.

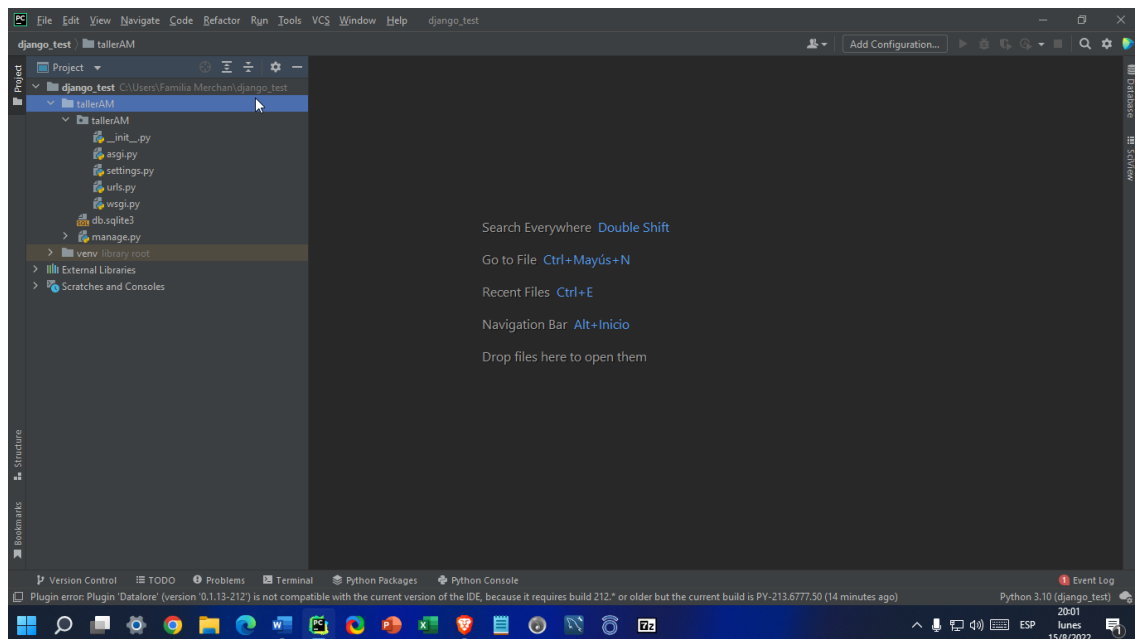
Crear un proyecto con la máquina virtual.



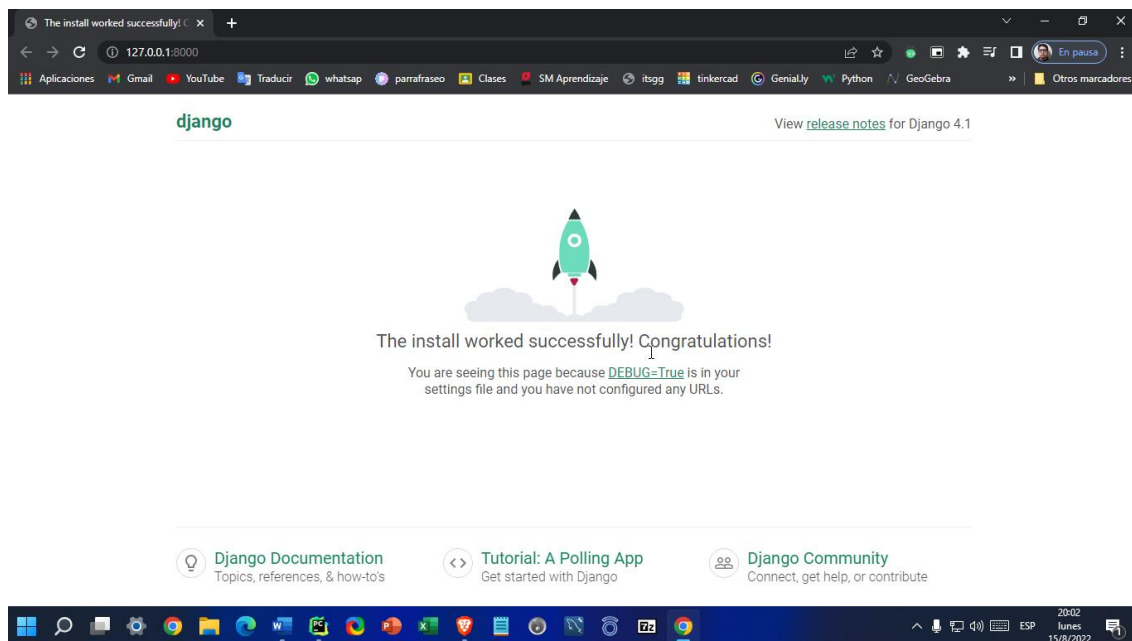
Descargar los instaladores de Django al proyecto



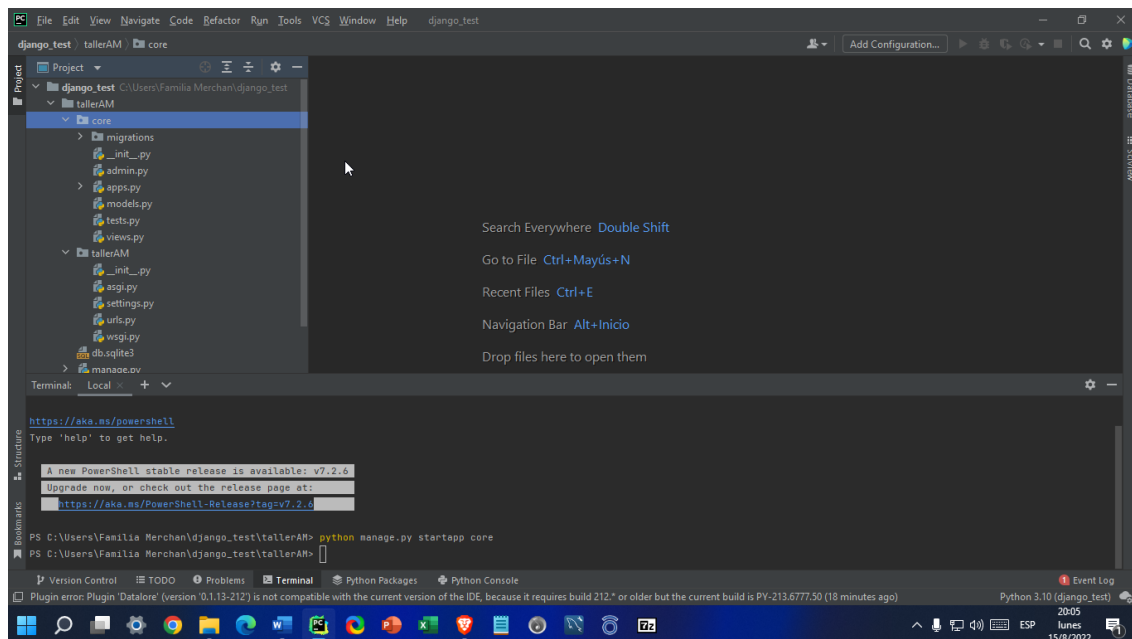
Crear un proyecto para programar en Django



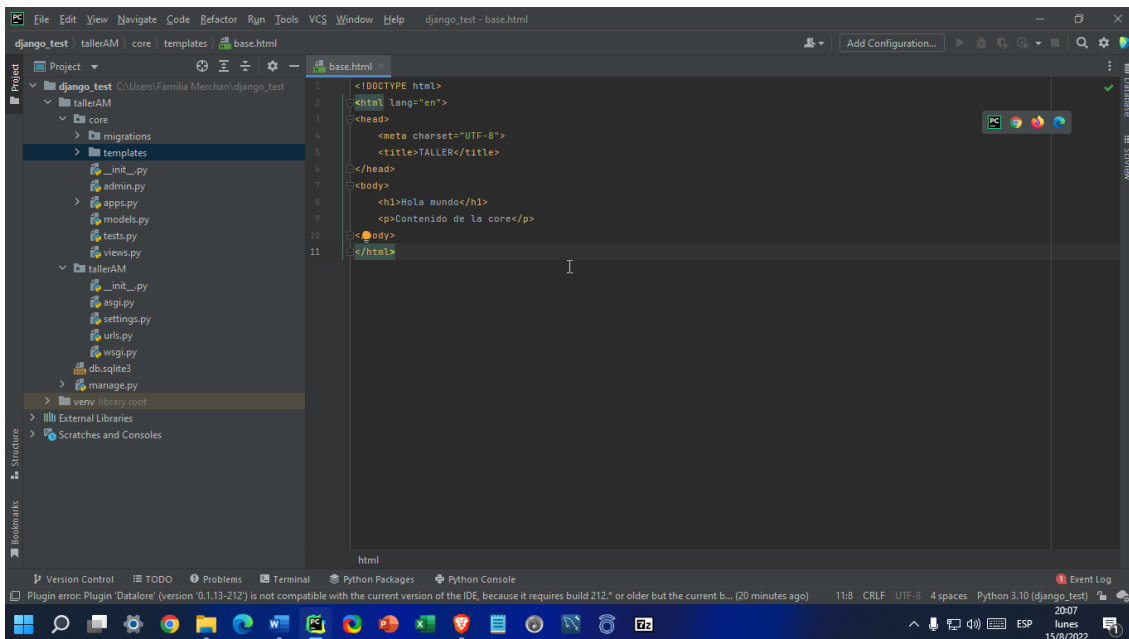
Ejecutar el proyecto y el mensaje de felicitaciones.



Crear una Apps core.

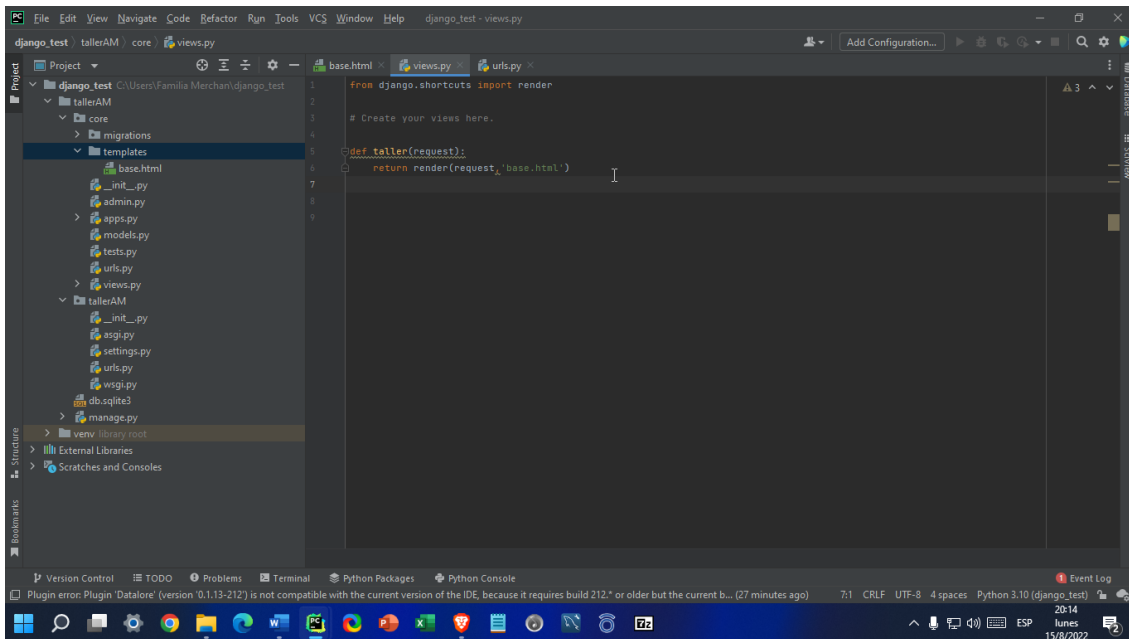


Crear un archivo base html en la APPS core



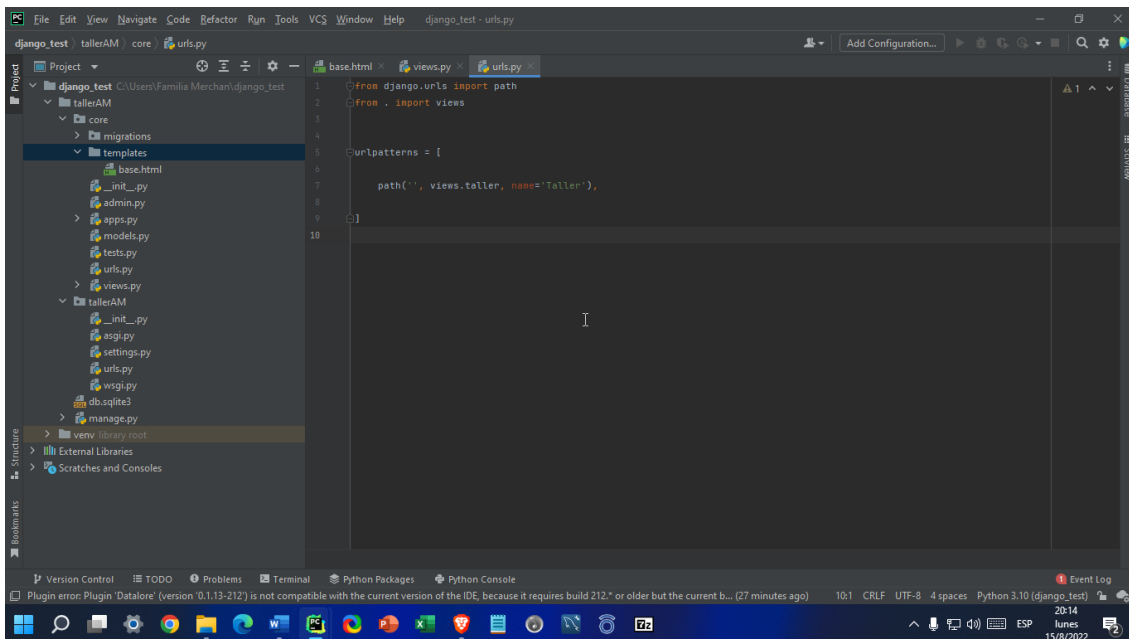
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>TALLER</title>
6 </head>
7 <body>
8 <h1>Hola mundo</h1>
9 <p>Contenido de la core</p>
10 </body>
11 </html>
```

Crear un view que llame al html hijo



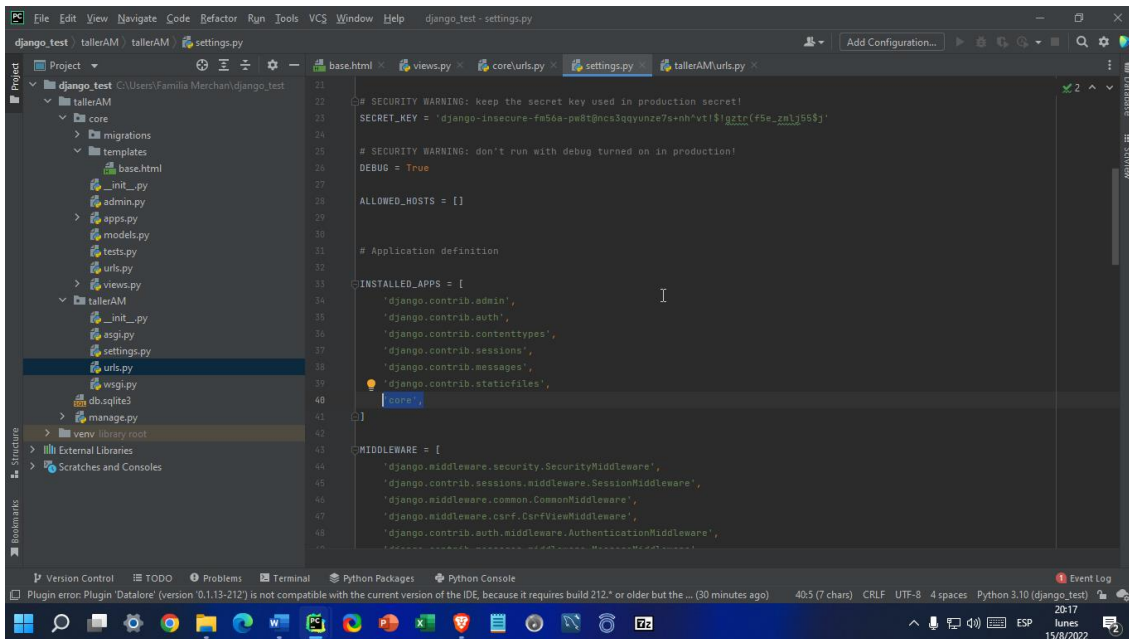
```
1 from django.shortcuts import render
2
3 # Create your views here.
4
5 def teller(request):
6     return render(request, 'base.html')
```


Crear la urls que llame al views.



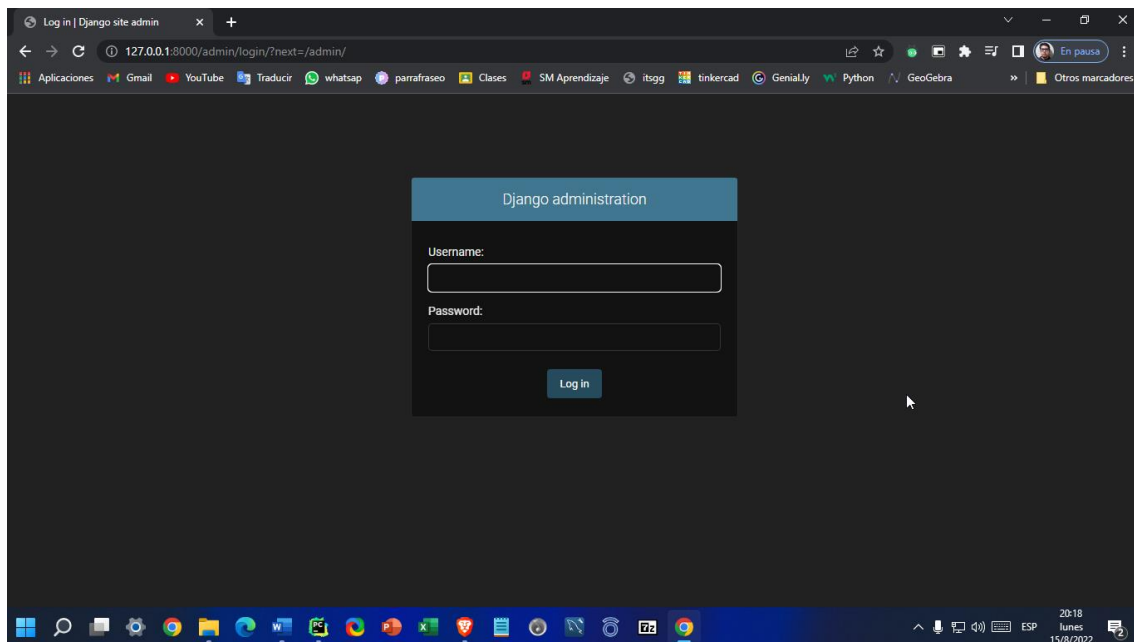
```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5
6     path('', views.taller, name='Taller'),
7
8 ]
```

Integrar la aplicación APPS core al proyecto principal

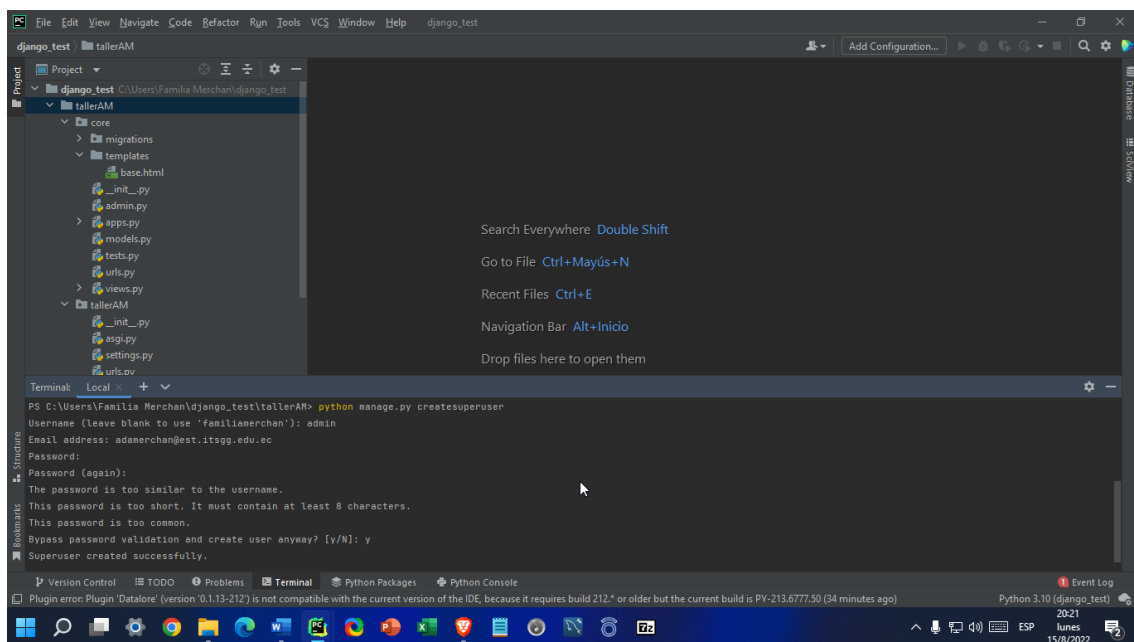


```
21 # SECURITY WARNING: keep the secret key used in production secret!
22 SECRET_KEY = 'django-insecure-fm56a-pw8t@nc3qayunze7s+nh*vt!$!gzt0(f9e_znlj55$'
23
24 # SECURITY WARNING: don't run with debug turned on in production!
25 DEBUG = True
26
27 ALLOWED_HOSTS = []
28
29 # Application definition
30
31 INSTALLED_APPS = [
32     'django.contrib.admin',
33     'django.contrib.auth',
34     'django.contrib.contenttypes',
35     'django.contrib.sessions',
36     'django.contrib.messages',
37     'django.contrib.staticfiles',
38     'core',
39 ]
40
41 MIDDLEWARE = [
42     'django.middleware.security.SecurityMiddleware',
43     'django.contrib.sessions.middleware.SessionMiddleware',
44     'django.middleware.common.CommonMiddleware',
45     'django.middleware.csrf.CsrfViewMiddleware',
46     'django.contrib.auth.middleware.AuthenticationMiddleware',
47 ]
```

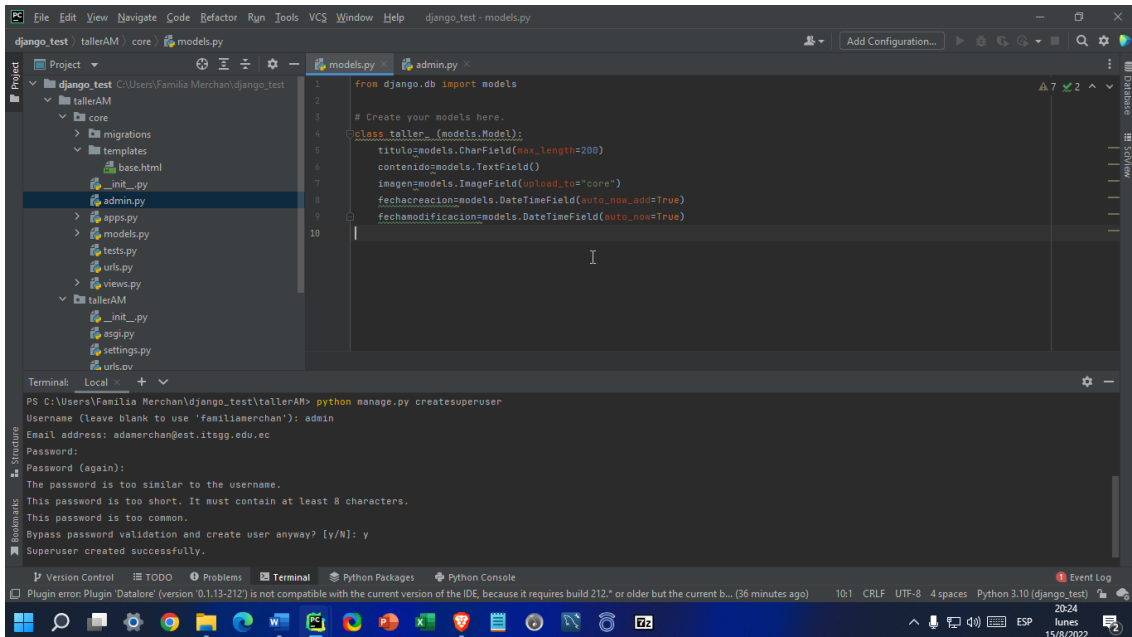
Crear las tablas del sistema de usuarios para utilizar el panel de administración.



Crear un usuario para poder ingresar al Panel de Administración



Crear un modelo en Django.



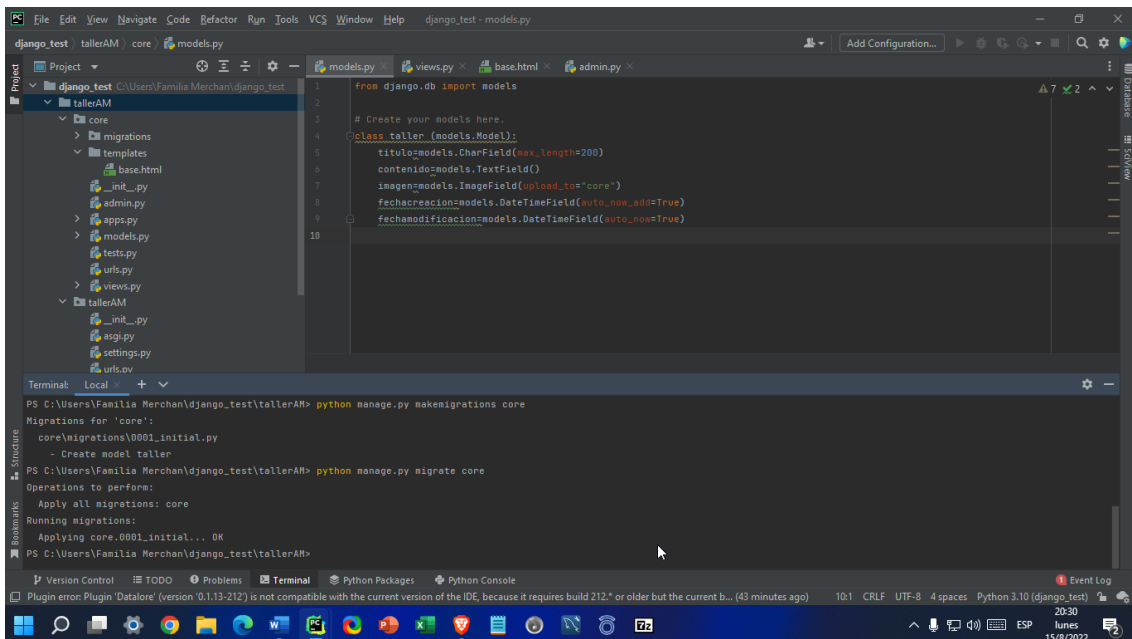
```
from django.db import models

# Create your models here.
class taller(models.Model):
    titulo=models.CharField(max_length=200)
    contenido=models.TextField()
    imagen=models.ImageField(upload_to='core')
    fecha_creacion=models.DateTimeField(auto_now_add=True)
    fecha_modificacion=models.DateTimeField(auto_now=True)
```

Terminal: Local

```
PS C:\Users\Familia Merchan\django_test\tallerAM> python manage.py createsuperuser
Username (leave blank to use 'familia_merchan'): admin
Email address: adminmerchan@est.itsg.edu.ec
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Migrar el Modelo a la base del Panel de Administración.



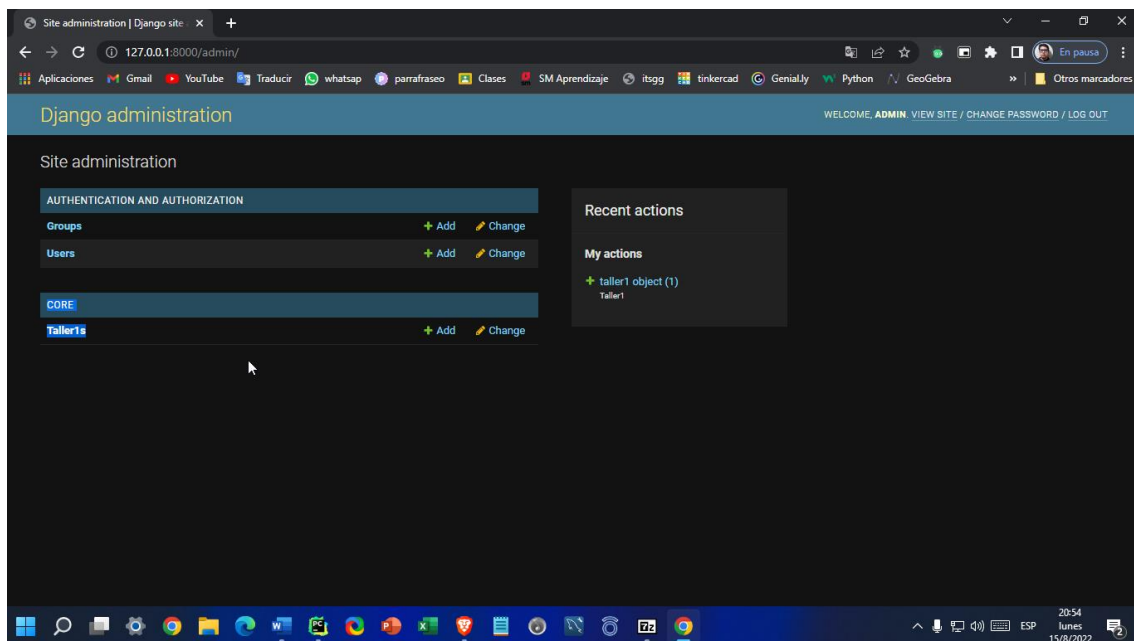
```
from django.db import models

# Create your models here.
class taller(models.Model):
    titulo=models.CharField(max_length=200)
    contenido=models.TextField()
    imagen=models.ImageField(upload_to='core')
    fecha_creacion=models.DateTimeField(auto_now_add=True)
    fecha_modificacion=models.DateTimeField(auto_now=True)
```

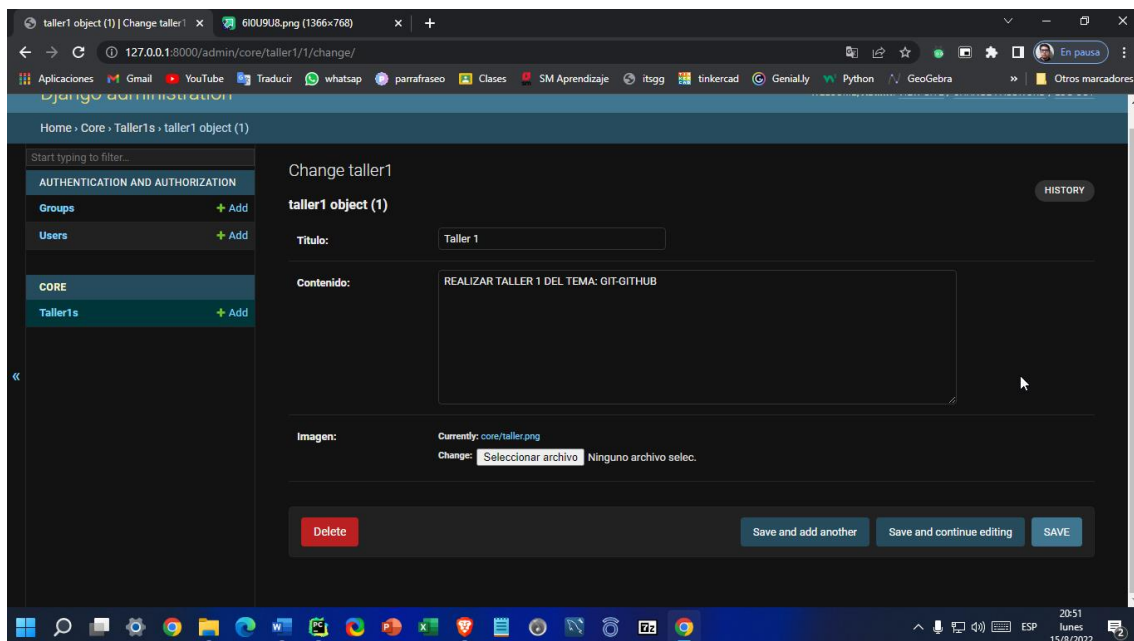
Terminal: Local

```
PS C:\Users\Familia Merchan\django_test\tallerAM> python manage.py makemigrations core
Migrations for 'core':
  core\migrations\0001_initial.py
  - Create model taller
PS C:\Users\Familia Merchan\django_test\tallerAM> python manage.py migrate core
Operations to perform:
  Apply all migrations: core
Running migrations:
  Applying core.0001_initial... OK
PS C:\Users\Familia Merchan\django_test\tallerAM>
```

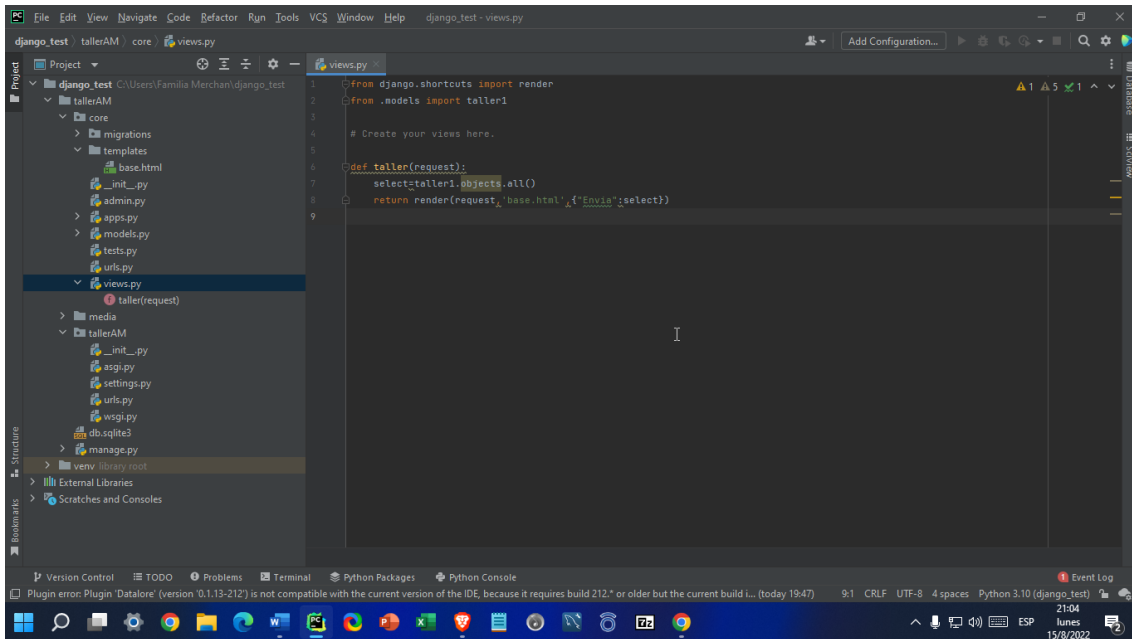
Integrar el Modelo al Panel de Administración.



Ingresar información al modelo por el Panel de Administración.



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



```
1 from django.shortcuts import render
2 from .models import taller1
3
4 # Create your views here.
5
6 def taller(request):
7     select=taller1.objects.all()
8     return render(request, 'base.html', {'Envia': 'select'})
9
```

Mostrar los datos guardados en el modelo al html hijo.

