



Instituto Superior Tecnológico
GUAYAQUIL



Nombre de la carrera:

Desarrollo del software

Estudiante: William

Alejandro Yagual Quimis

Curso: “3H”

Nombre del docente:

CARLOS LUIS PAZMIÑO

PALMA

Foto del estudiante.





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



1) ¿Qué es Django?

Django es un framework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables.

2) ¿Qué es la máquina virtual en Django?

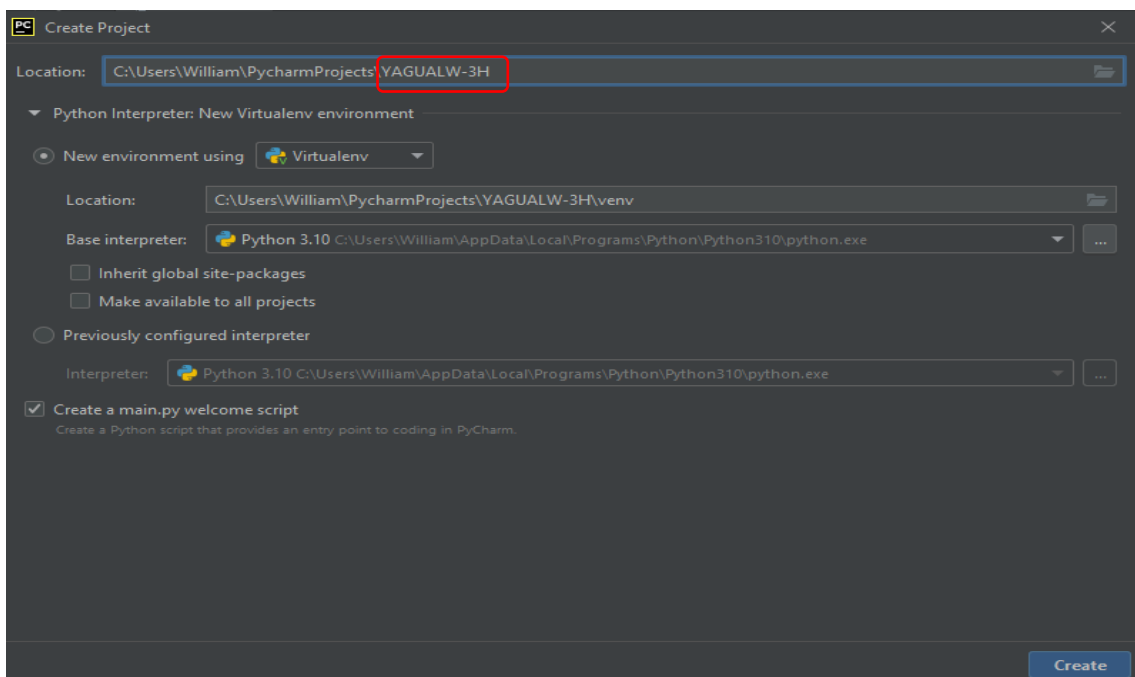
es un entorno de Python aislado creado para un proyecto. Este entorno tiene su propio intérprete de Python, bibliotecas y paquetes, lo que significa que el intérprete de Python y las dependencias instaladas pertenecen solo a este proyecto.

3) ¿Qué es MVT en Django?

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

4) Crear un proyecto con la máquina virtual.

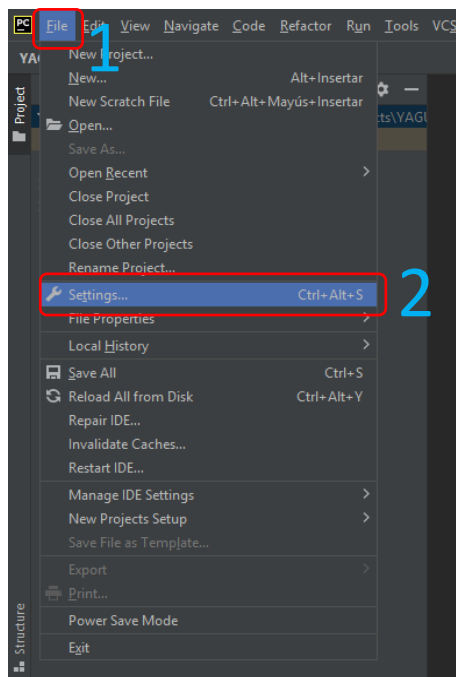
Colocamos el nombre que deseamos para nuestro proyecto en mi caso es lo que está en rojo



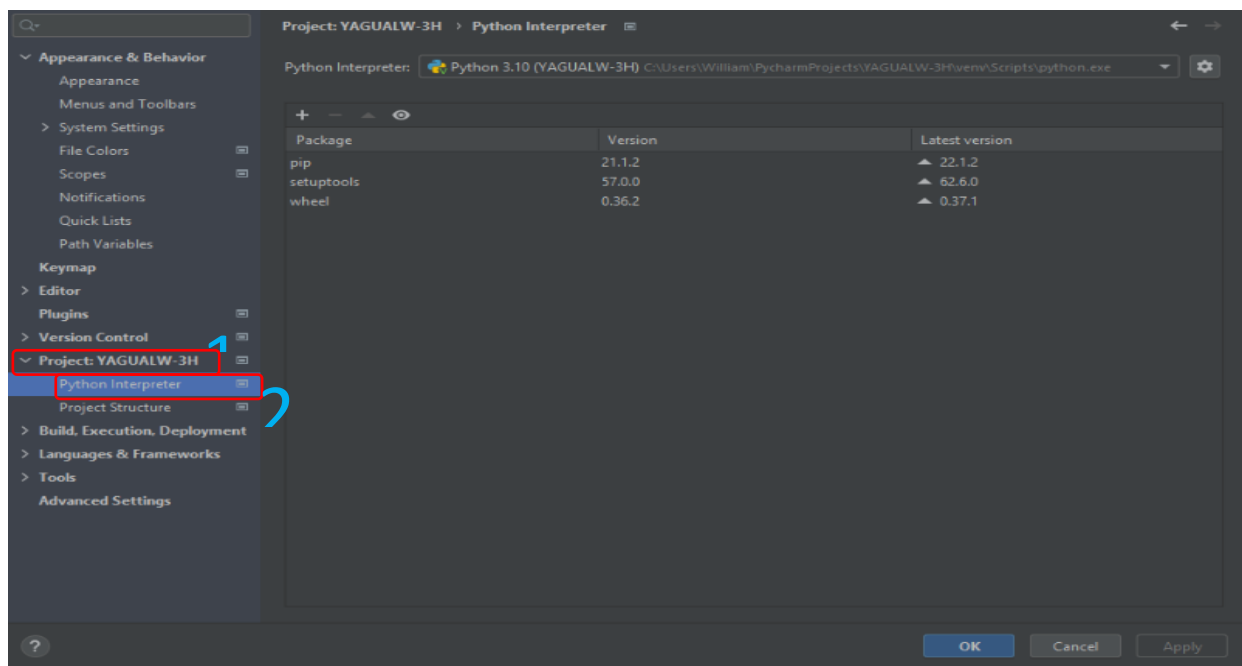


5) Descargar los instaladores de Django al proyecto

Le damos a file y buscamos la opción settings

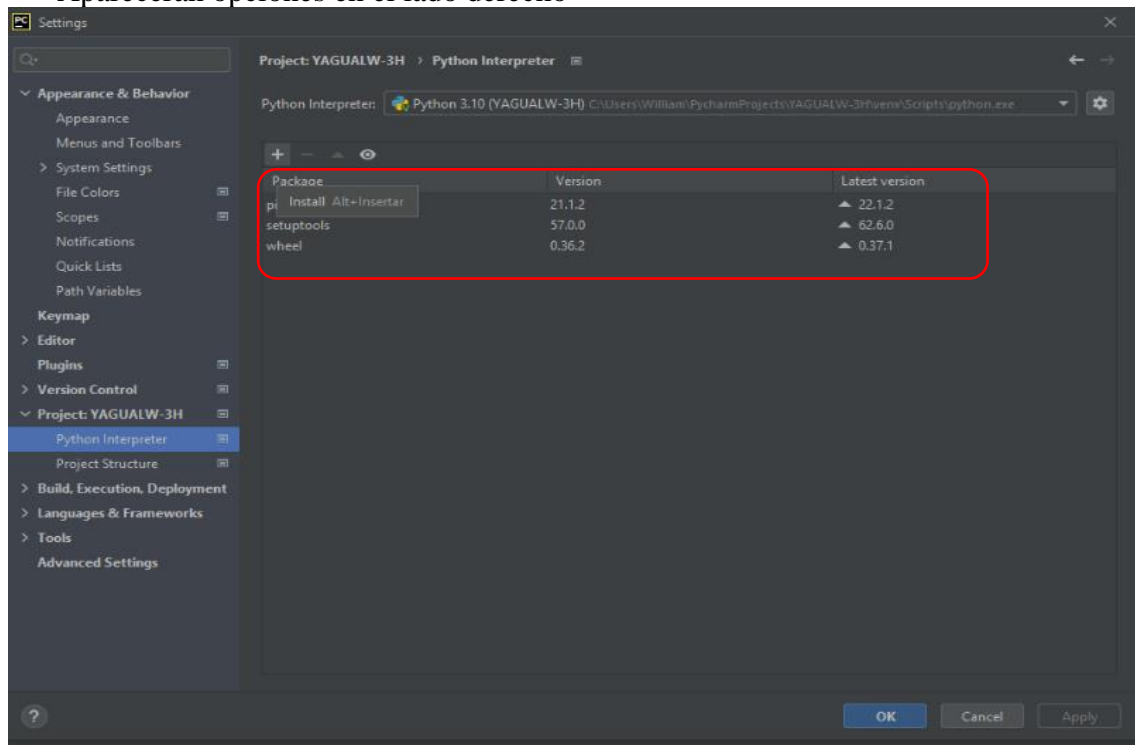


Nos aparecerá esta ventana en la cual buscaremos el nombre de nuestro proyecto y le damos en Python interprete

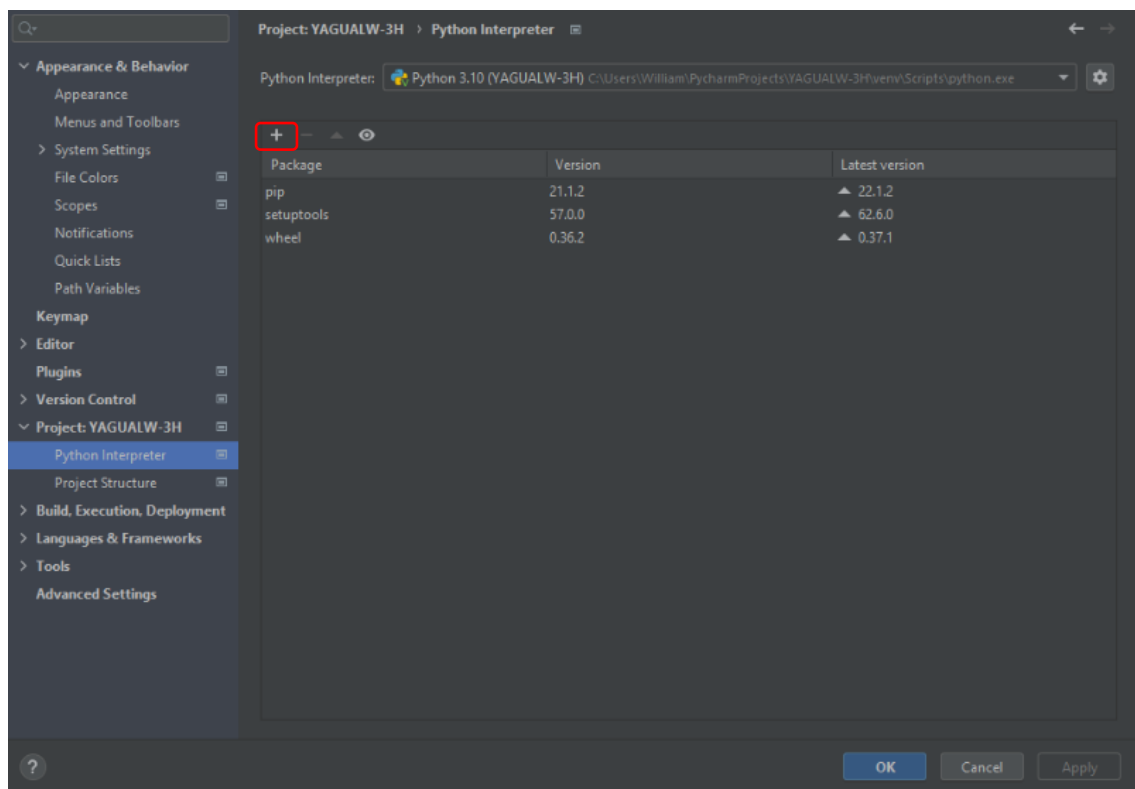




Aparecerán opciones en el lado derecho

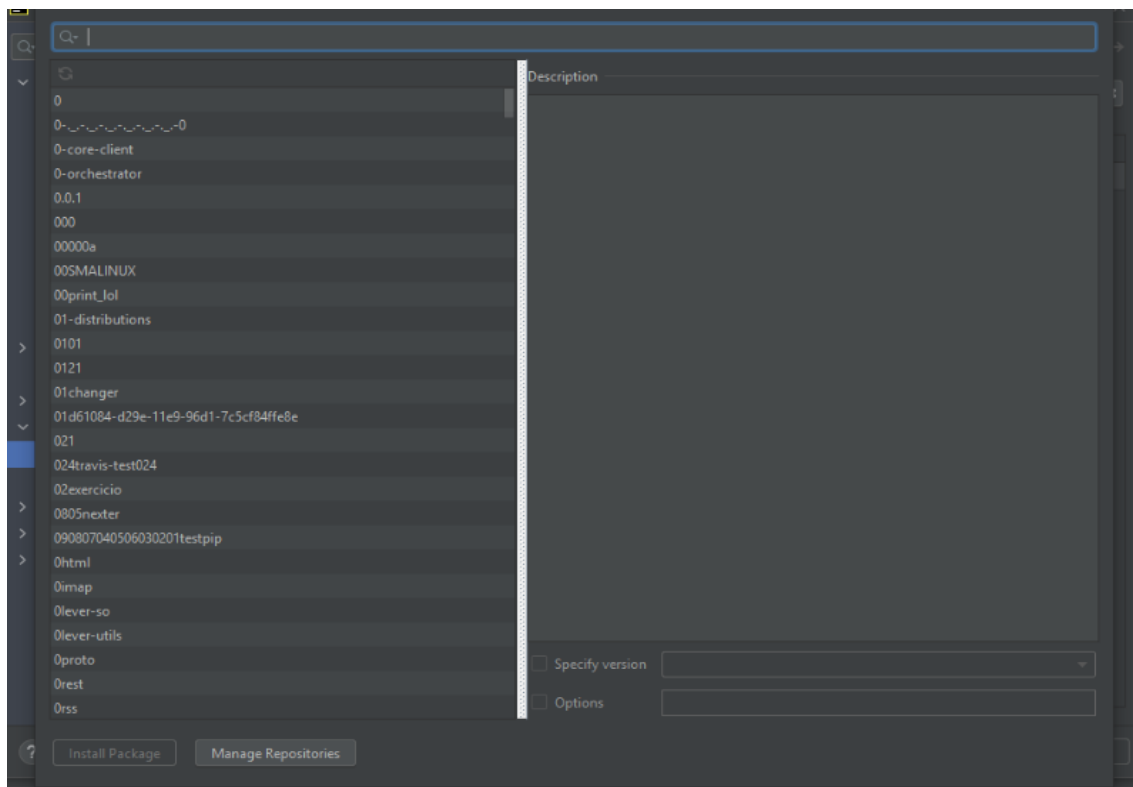


Y le daremos al mas

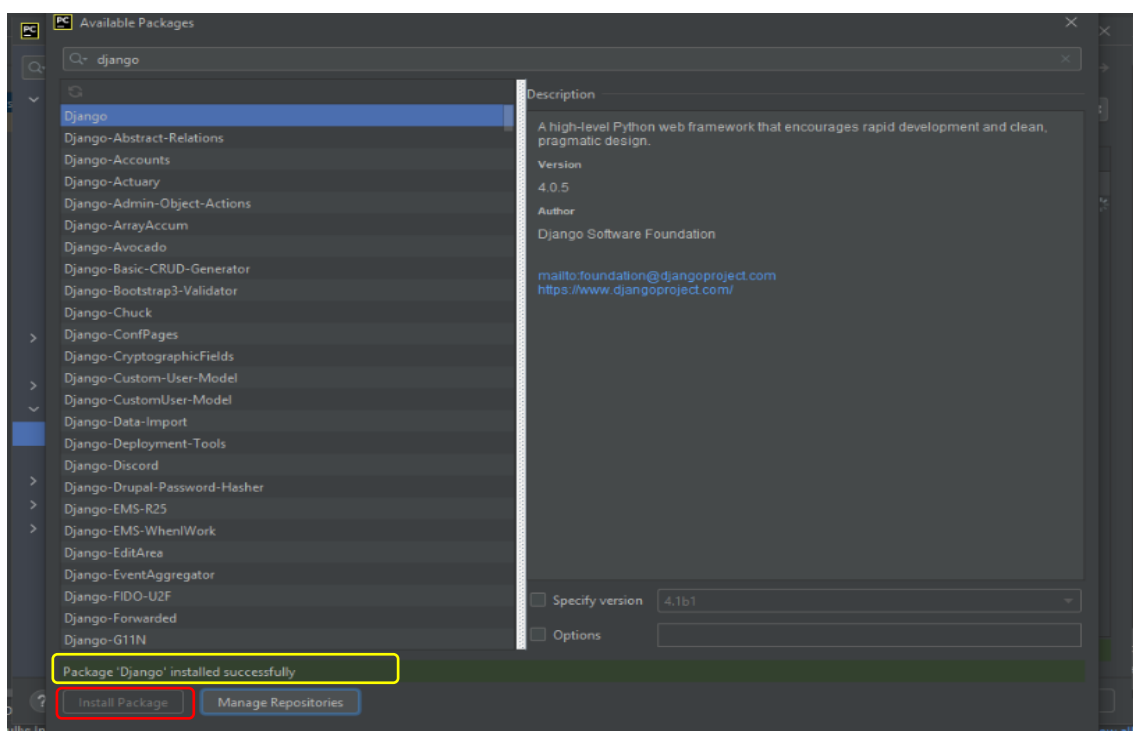




Se nos abrirá esta ventana y procedemos a buscar Django

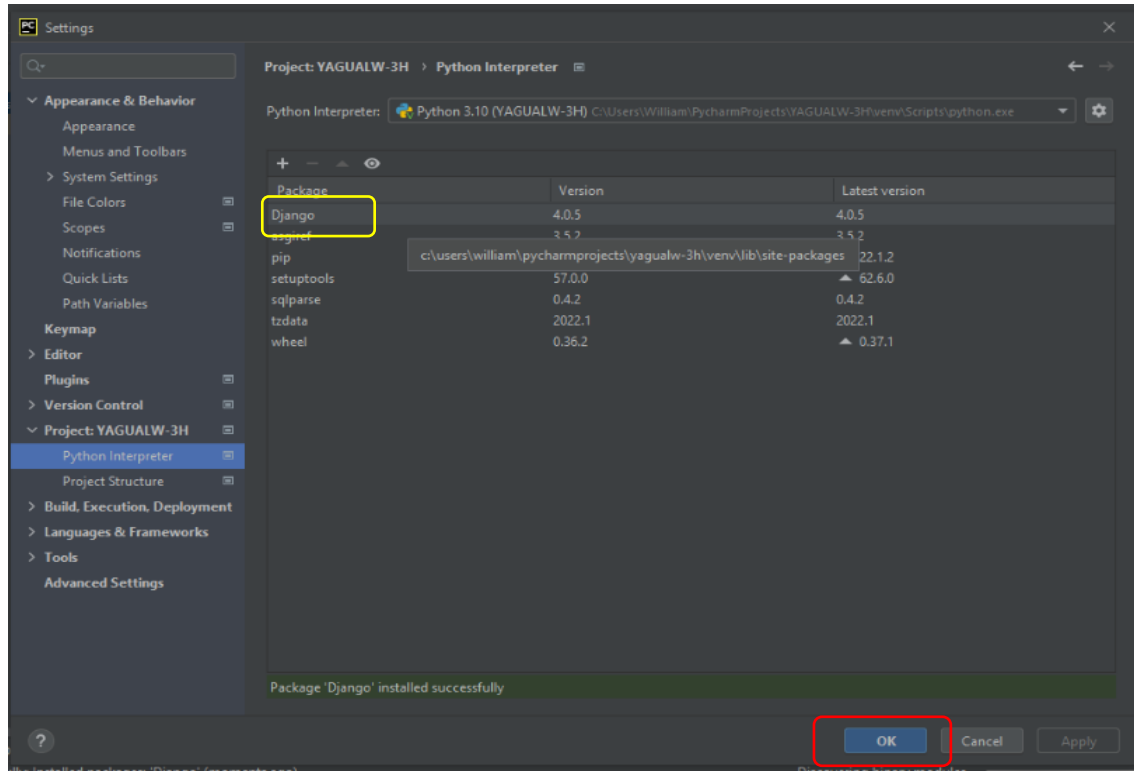


Después le damos a install package y una vez terminado nos aparecerá lo del rectángulo amarillo que significa que ya se completó la instalación



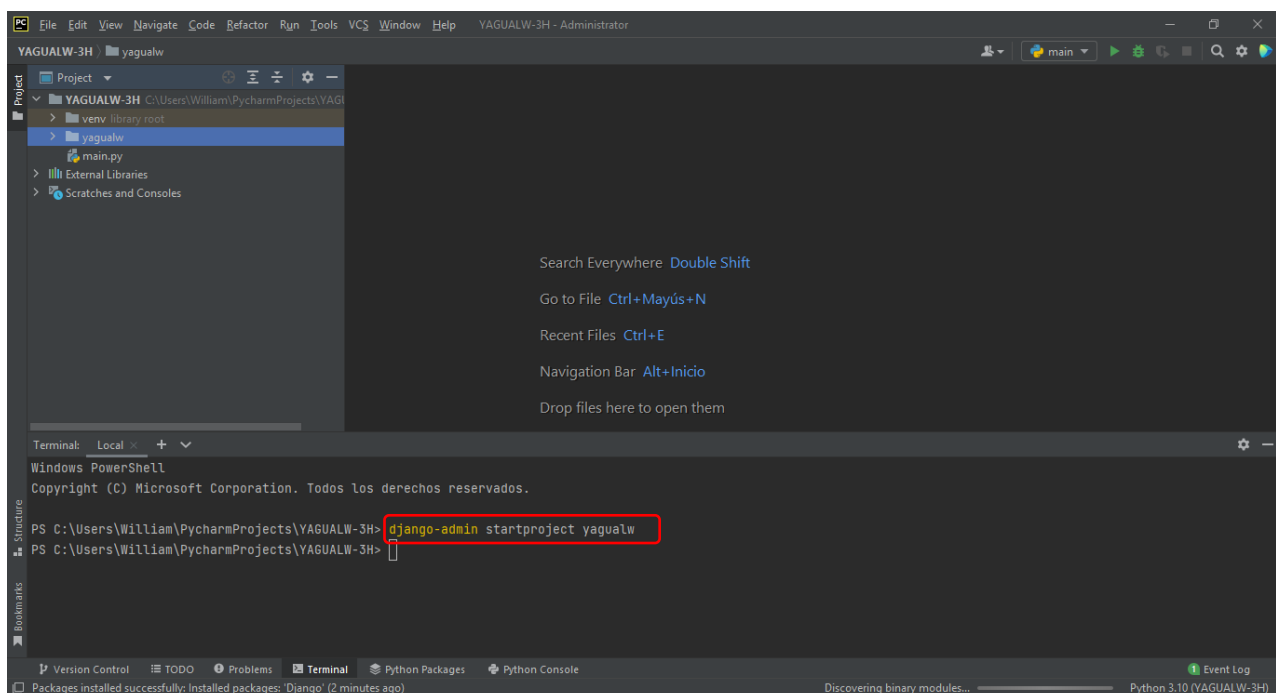


Le damos a la X y vemos que nos aparece Django ya instalado y le damos a OK



6) Crear un proyecto para programar en Django

Para administrar Django y crear la carpeta de nuestro proyecto utilizamos la sentencia del rectángulo rojo y colocamos el nombre que deseamos





Utilizamos el `cd` y el nombre que deseamos por ej: yagualw para ubicarnos en la carpeta del proyecto creada

```
YAGUALW-3H | yagualw
Project
  YAGUALW-3H C:\Users\William\PycharmProjects\YAG
  venv library root
  yagualw
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
    manage.py
    main.py
  External Libraries
  Scratches and Consoles

Search Everywhere Double Shift
Go to File Ctrl+Mayús+N
Recent Files Ctrl+E
Navigation Bar Alt+Inicio
Drop files here to open them

Terminal Local +
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\William\PycharmProjects\YAGUALW-3H> django-admin startproject yagualw
PS C:\Users\William\PycharmProjects\YAGUALW-3H> cd yagualw
PS C:\Users\William\PycharmProjects\YAGUALW-3H\yagualw>

Version Control | TODO | Problems | Terminal | Python Packages | Python Console
Packages installed successfully: Installed packages: 'Django' (3 minutes ago) Python 3.10 (YAGUALW-3H)
```

Utilizamos un `dir`

```
YAGUALW-3H | yagualw
Project
  YAGUALW-3H C:\Users\William\PycharmProjects\YAG
  venv library root
  yagualw
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
    manage.py
    main.py
  External Libraries
  Scratches and Consoles

Search Everywhere Double Shift
Go to File Ctrl+Mayús+N
Recent Files Ctrl+E
Navigation Bar Alt+Inicio

Terminal Local +
PS C:\Users\William\PycharmProjects\YAGUALW-3H> cd yagualw
PS C:\Users\William\PycharmProjects\YAGUALW-3H\yagualw> dir

Mode                LastWriteTime         Length Name
----                -
d-----          1/7/2022   10:45             yagualw
-a----          1/7/2022   10:45             685 manage.py

PS C:\Users\William\PycharmProjects\YAGUALW-3H\yagualw>

Version Control | TODO | Problems | Terminal | Python Packages | Python Console
Packages installed successfully: Installed packages: 'Django' (4 minutes ago) Python 3.10 (YAGUALW-3H)
```




7) Ejecutar el proyecto y el mensaje de felicitaciones.

Utilizamos la sentencia `Python manage.py runserver` para ejecutar el proyecto

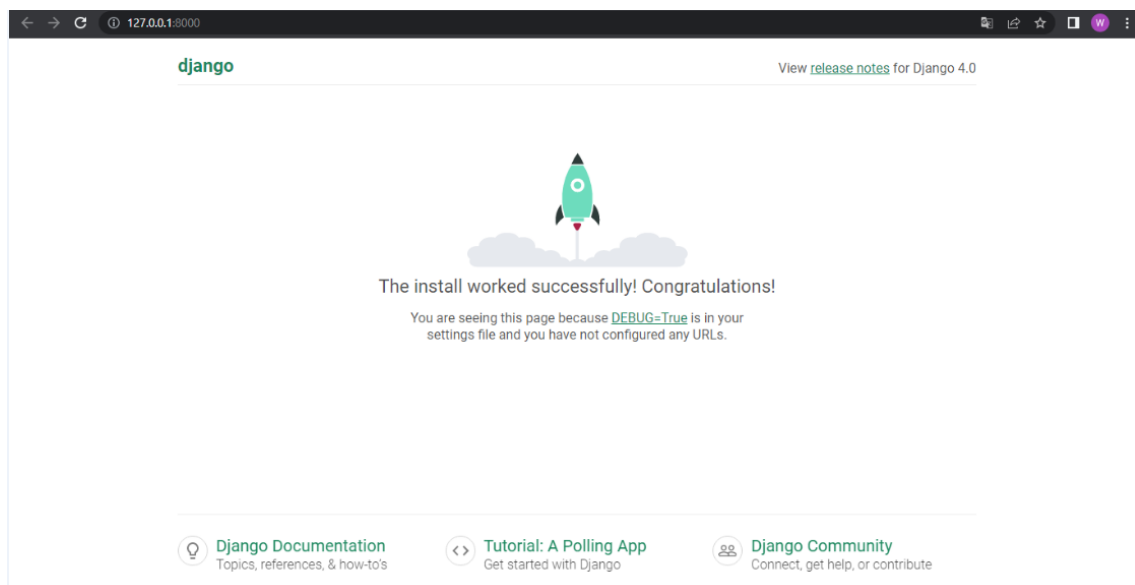
Nos aparecerá lo del rectángulo amarillo y le damos clic lo que nos aparece de azul

```
PS C:\Users\William\PycharmProjects\YAGUALW-3H\yagualw> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
July 01, 2022 - 10:48:11
Django version 4.0.5, using settings 'yagualw.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

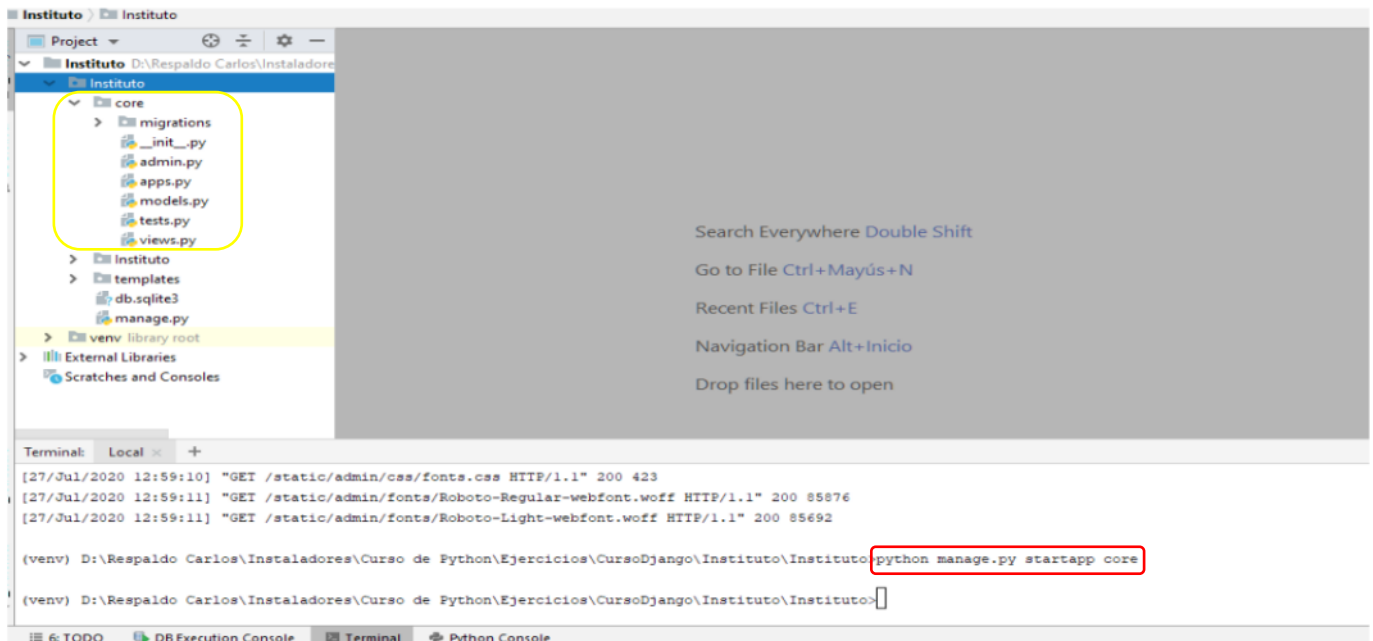
Una vez dado clic nos llevara al Mensaje de felicitaciones como veremos a continuación





8) Crear una Apps core.

Utilizamos la sentencia **Python manage.py startapp core** como aparece en el rectángulo rojo le damos enter y nos aparecerá lo del cuadro amarillo



9) ¿Qué es la Carpeta Templates?

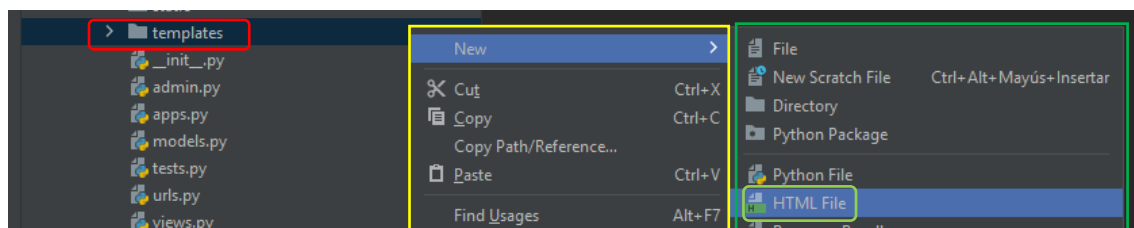
Es la subcarpeta que contiene plantillas que se muestran en el cuadro de diálogo Crear nuevo archivo. Puede crear y guardar plantillas personalizadas en la carpeta Templates. Contiene plantillas para la creación de archivos. Se puede guardar el archivo activo como una plantilla.

10) ¿Qué es la Carpeta static?

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

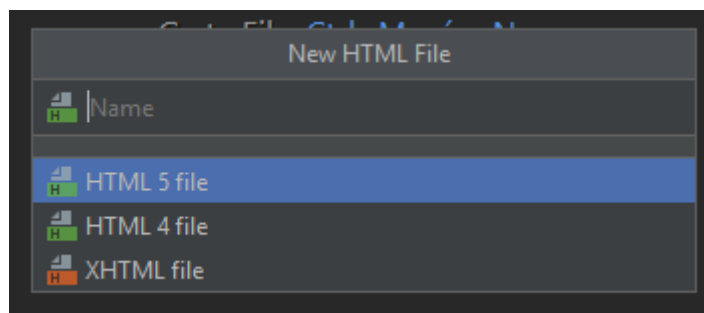
11) Crear un archivo base HTML en la APPS core

Nos vamos a la carpeta **templates** le damos clic derecho y nos aparecerá lo del cuadrado amarillo le damos a **New** y luego nos aparece lo del cuadrado verde y le damos en **HTML File**





Luego nos aparecerá este cuadradito donde nos pedirá el nombre para el HTML le ponemos el nombre y le damos enter



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

Utilizamos: `<link rel="STYLESHEET" href="{%static 'core/css/estilo1.css' %}">` lo de rojo es la ubicación del archivo css y arriba colocamos un `{% load static %}` como en el ejemplo que vemos a continuación

```
{% load static %}
<link rel="STYLESHEET" href="{%static 'core/css/estilo1.css' %}">
```

13) Como consume un archivo hijo HTML al utilizar la herencia del archivo base HTML.

Utilizamos `{% extends 'herencia/Principal.html' %}` lo de rojo es donde se encuentra el archivo base.

Después utilizamos `{% block Contenido %}` ponemos todo el contenido que deseamos y para cerrar utilizamos `{% endblock %}`. ejemplo:

```
{% extends 'herencia/Principal.html' %}

{% block Contenido %}
<h1 align="center"><b>LECCION DE PROGRAMACION VISUAL</b></h1>
<center>
<center><footer class="footer">
<center><p>Apellidos y Nombres: <input type="submit" value="Yagual Quimis William Alejandro">
<center><p>Curso:<input type="submit" value="3H">
<center><p>Profesor:<input type="submit" value="CARLOS LUIS PAZMIÑO PALMA">
<center><p>Materia:<input type="submit" value="PROGRAMACION VISUAL">
</footer></h2>
{% endblock %}
```



14) Crear un view que llame al HTML hijo

Nos vamos al view de la core.

En el rectángulo rojo colocamos el nombre de nuestro archivo

En el rectángulo amarillo colocamos donde se encuentra el HTML hijo ejemplo:

```
def noticiash(request):  
    return render(request, herencia/noticiash.html)
```

15) Crear la urls que llame al views.

Nos vamos a la urls de la core colocamos **path('noticiash', views.noticiash, name= 'noticiash')**, y en lo que esta con rojo cambiamos por el nombre de nuestro archivo hijo ejemplo:

```
path('noticiash', views.noticiash, name= 'noticiash'),
```

16) Integrar la aplicación APPS core al proyecto principal

Utilizamos **manage.py makemigrations Core** para integrarla al proyecto principal

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

Utilizamos **Python manage.py migrate** para crear las tablas y poder utilizar el panel de administración

```
Terminal: Local x + v  
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, auth, contenttypes, sessions  
Running migrations:  
  Applying contenttypes.0001_initial... OK  
  Applying auth.0006_require_contenttypes_0002... OK  
  Applying auth.0007_alter_validators_add_error_messages... OK  
  Applying auth.0008_alter_user_username_max_length... OK  
  Applying auth.0009_alter_user_last_name_max_length... OK  
  Applying auth.0010_alter_group_name_max_length... OK  
  Applying auth.0011_update_proxy_permissions... OK  
  Applying auth.0012_alter_user_first_name_max_length... OK  
  Applying sessions.0001_initial... OK  
PS C:\Users\William\Downloads\l\3H0\instituto>
```



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

Utilizamos **Python manage.py createsuperuser** para crear el usuario

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
```

Nos preguntara el nombre de usuario y colocamos el que deseamos

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
Username (leave blank to use 'william'): yagualw
```

Después nos preguntara el Email y lo colocamos

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
Username (leave blank to use 'william'): yagualw
Email address: yagual@gmail.com
```

Después nos pide la contraseña y colocamos la de nuestra preferencia

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
Username (leave blank to use 'william'): yagualw
Email address: yagual@gmail.com
Password:
```

Pedirá que la confirme

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
Username (leave blank to use 'william'): yagualw
Email address: yagual@gmail.com
Password:
Password (again):
```



Si tu contraseña es muy corta te dirá que es corta y que debe contener al menos 8 caracteres

Que es muy común y puede ser alfanumérica

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
Username (leave blank to use 'william'): yagualw
Email address: yagual@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
```

Pero también te da la opción de crear el usuario de todos modos diciendo **yes[y] / no[N]**, y una vez ya teniendo tu contraseña ya tendrías tu usuario creado

```
Terminal: Local x + v
PS C:\Users\William\Downloads\l\3H0\instituto> python manage.py createsuperuser
Username (leave blank to use 'william'): yagualw
Email address: yagual@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS C:\Users\William\Downloads\l\3H0\instituto> |
```




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.

20) Crear un modelo en Django.

Nos vamos a **models.py** creamos el modelo como vemos a continuación en el cuadro amarillo:

```
1 from django.db import models
2
3 # Create your models here.
4 class Portafolio(models.Model):
5     titulo= models.CharField(max_length=200)
6     contenido= models.TextField()
7     imagen= models.ImageField(upload_to='media')
8     fechacreacion= models.DateTimeField(auto_now_add=True)
9     fechamodificacion= models.DateTimeField(auto_now=True)
```

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

Migramos el modelo y observamos que se creo

```
10 dependencies = [
11 ]
12
13 operations = [
14     migrations.CreateModel(
15         name='Portafolio',
16         fields=[
17             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
18             ('titulo', models.CharField(max_length=200)),
19             ('contenido', models.TextField()),
20             ('imagen', models.ImageField(upload_to='proyecto')),
21             ('fechacreacion', models.DateTimeField(auto_now_add=True)),
22             ('fechamodificacion', models.DateTimeField(auto_now=True)),
23         ],
24     ),
25 ]
```

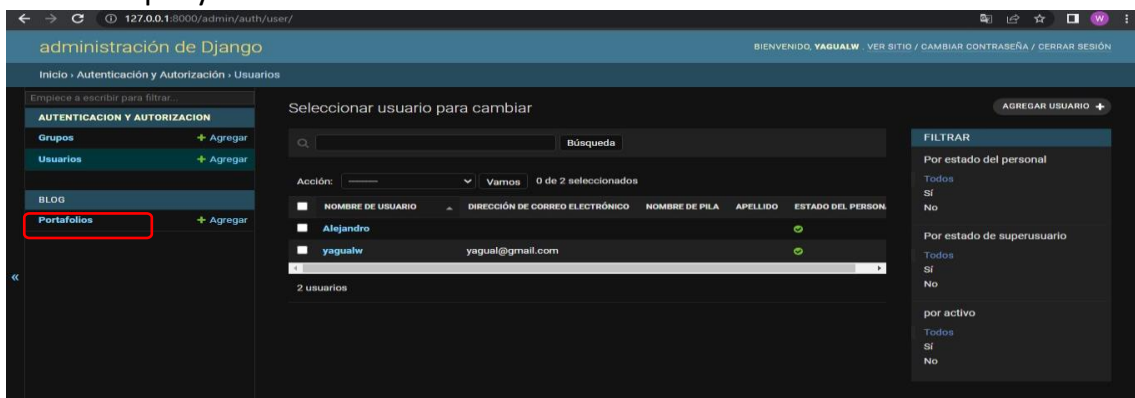


22) Integrar el Modelo al Panel de Administración.

Migramos:

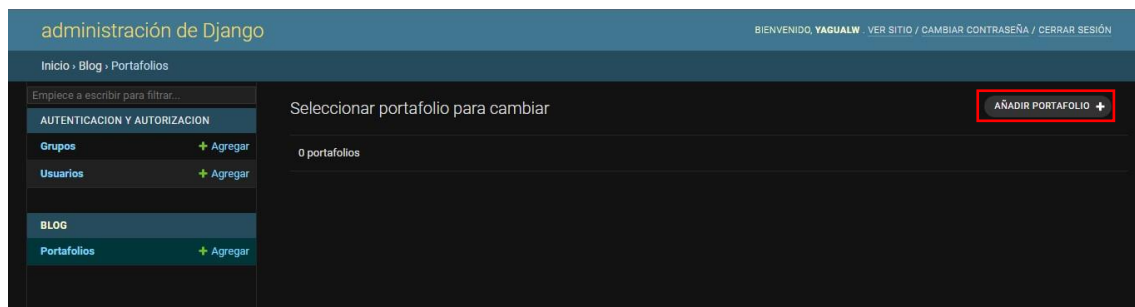
```
PS C:\Users\William\PycharmProjects\YAGUALA\ayagual> python manage.py migrate blog
Operations to perform:
  Apply all migrations: blog
Running migrations:
  Applying blog.0001_initial... OK
PS C:\Users\William\PycharmProjects\YAGUALA\ayagual>
```

Y vemos que ya se creó:

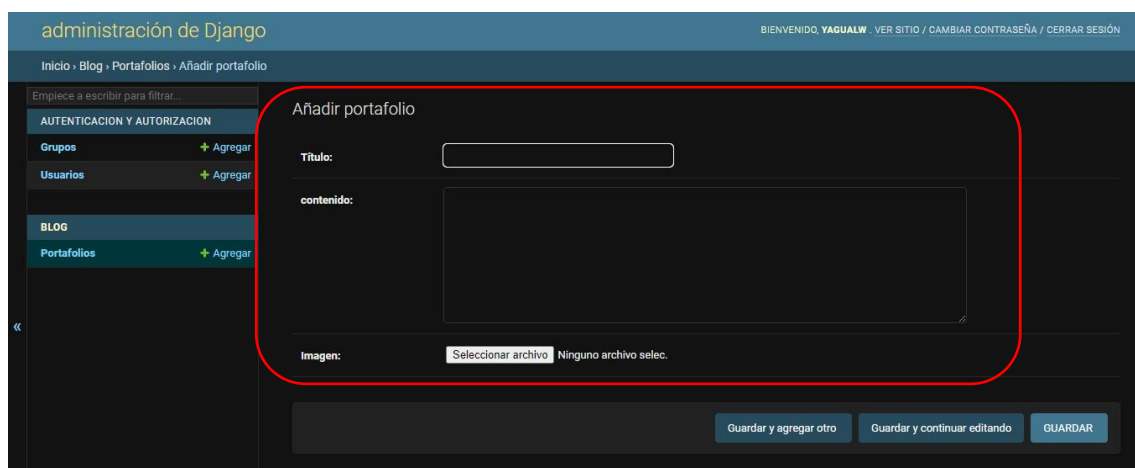


23) Ingresar información al modelo por el Panel de Administración.

Le damos en **Añadir Portafolio**



Se nos abrirá lo siguiente y procedemos a llenar con lo que deseamos





Una vez llenado le damos a guardar

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

Primero importamos el modelo como vemos en el rectángulo rojo y después creamos el def con un select como vemos en el rectángulo amarillo

```
from django.shortcuts import render
from . models import Portafolio
from . models import contacto1
# Create your views here.
def blog(request):
    select=Portafolio.objects.all()
    return render(request, 'blog.html', {'Enviar_blog': select})
```



25)Mostrar los datos guardados en el modelo al HTML hijo.

```
1  {% extends 'herencia/Principal.html' %}
2
3  {% block titulo %}
4      Contactos
5  {% endblock %}
6
7  {% block Contenido %}
8  <center><h1>Nombre: William Yagual</h1><footer class="footer"> <p>&#x2706; Telefono: 0993456791</p> <p>&#x2709; Correo:
9  <p>Asunto: <input type="text" name="Asunto" size="40"></p>
10 <p>Mensaje:<input type="text" name="Mensaje" size="40"></p>
11 <input type="submit" value="ENVIAR">
12 <input type="submit" value="CANCELAR">
13
14  {% endblock %}
15
```

arte.com

INICIO SIN HERENCIA INICIO QUIENES SOMOS NUESTROS PRODUCTOS CONTACTO BLOG

Nombre: William Yagual

☎ Telefono: 0993456791
✉ Correo: Juan@gmail.com

Asunto:

Mensaje:

ENVIAR CANCELAR