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 stactic? – 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 flamework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables.

* 1. ¿Qué es la máquina virtual en Django?

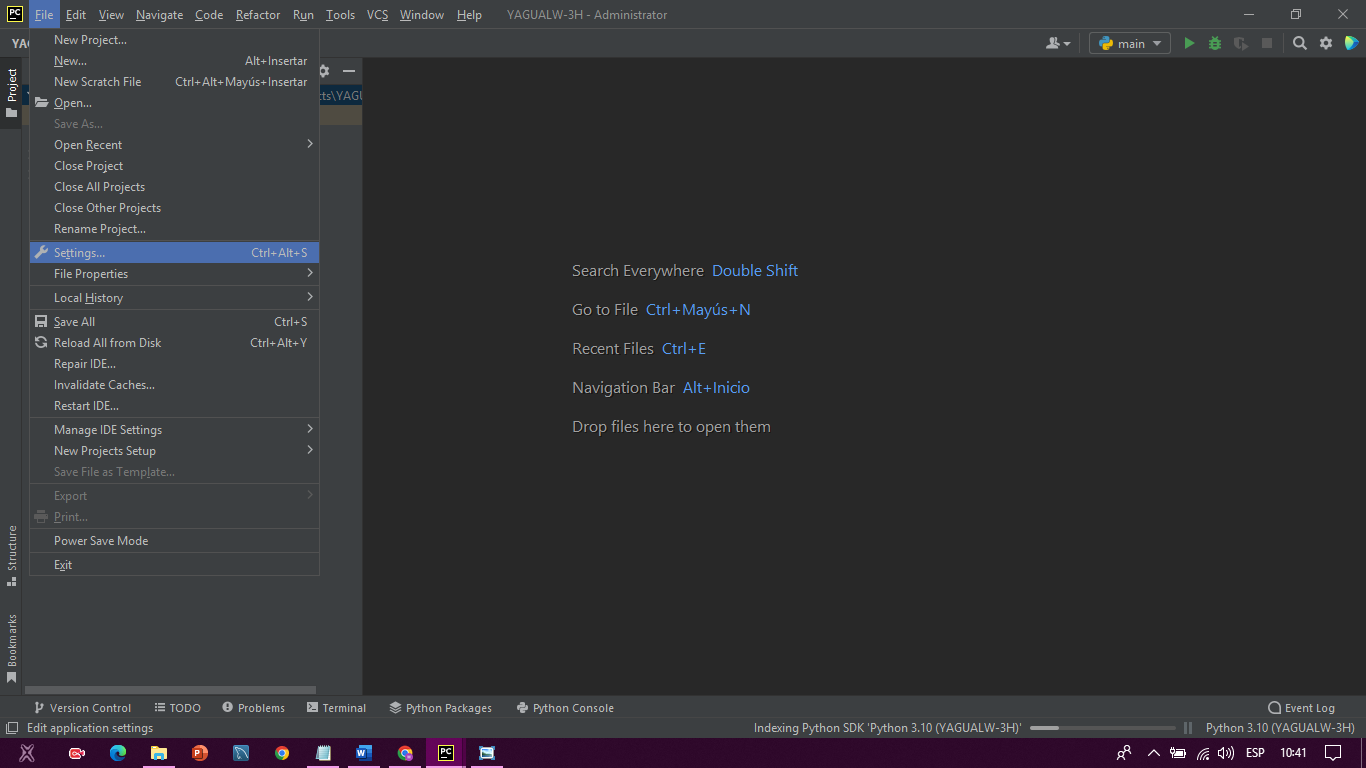
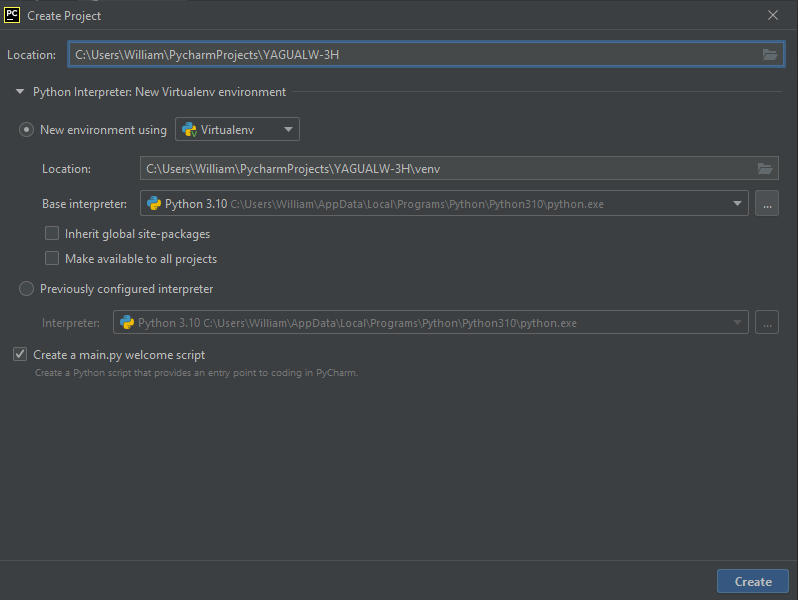
es un entorno de Python aislado creado para un proyecto. Este entorno tiene su propio intérprete de Python, bibliotecas y paquetes, lo que significa que el intérprete de Python y las dependencias instaladas pertenecen solo a este proyecto.

* 1. ¿Qué es MVT en Django?

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

* 1. Crear un proyecto con la máquina virtual.

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



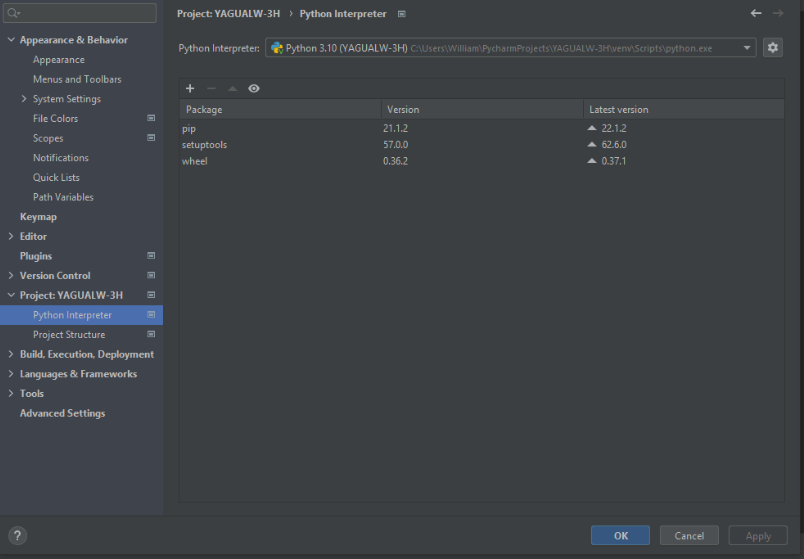
1

2

* 1. Descargar los instaladores de Django al proyecto

Le damos a file y buscamos la opción setings

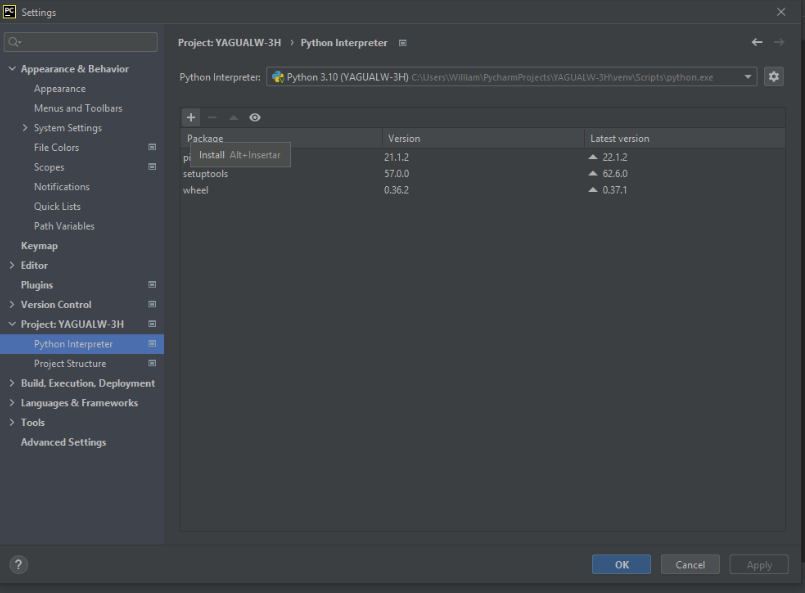
Nos aparecerá esta ventana en la cual buscaremos el nombre de nuestro proyecto y le damos en Python interprete



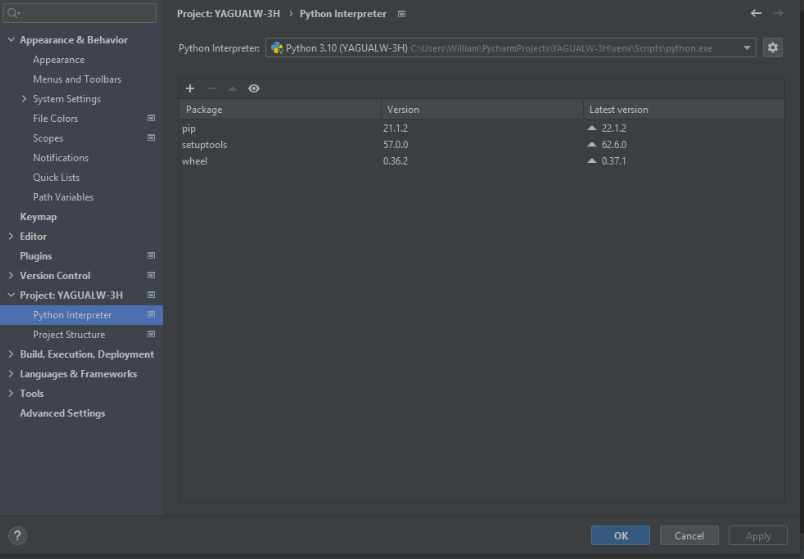
1

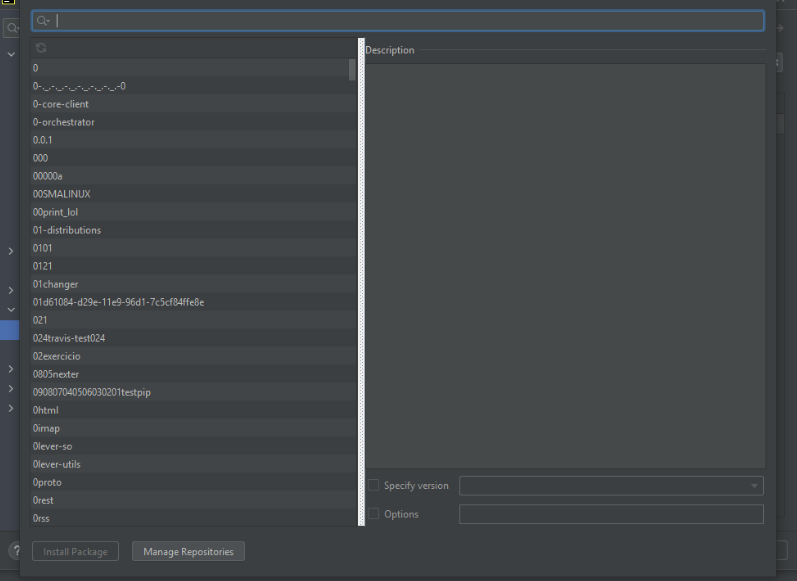
2

Aparecerán opciones en el lado derecho

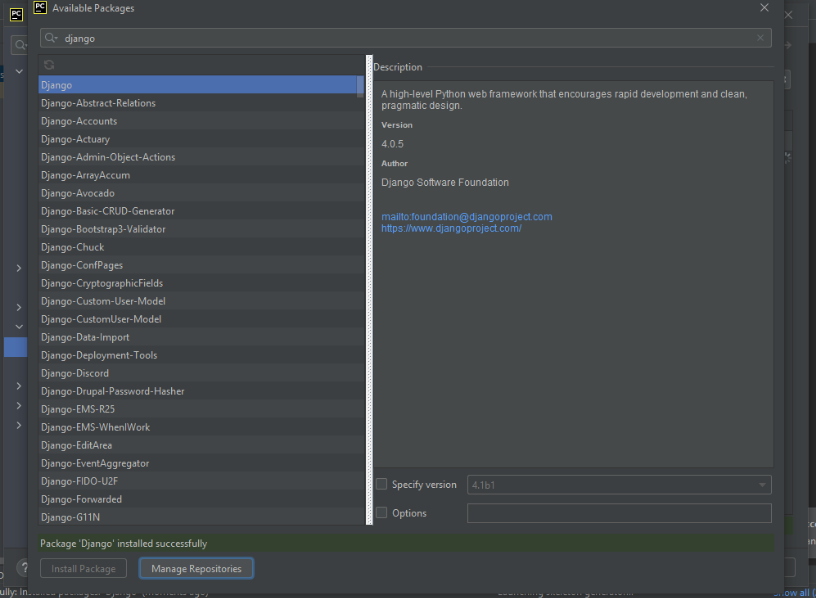


Y le daremos al mas

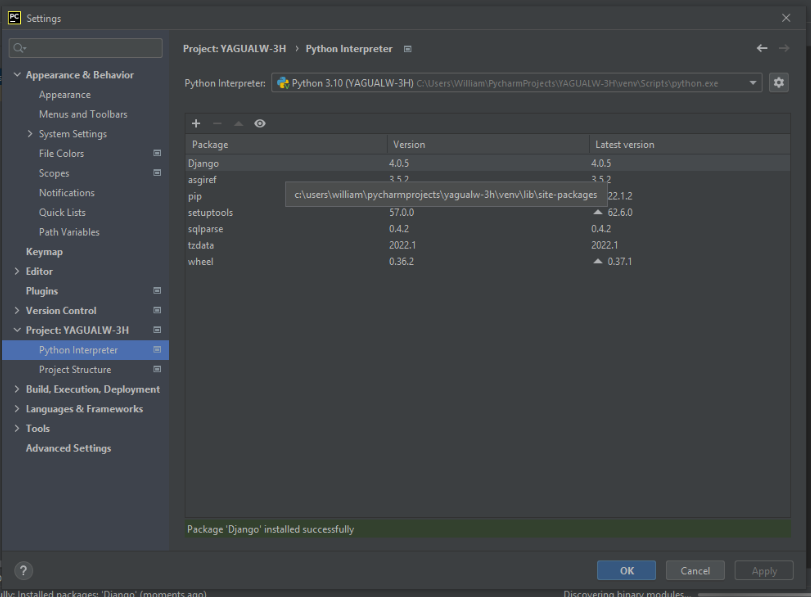


Se nos abrirá esta ventana y procedemos a buscar Django

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

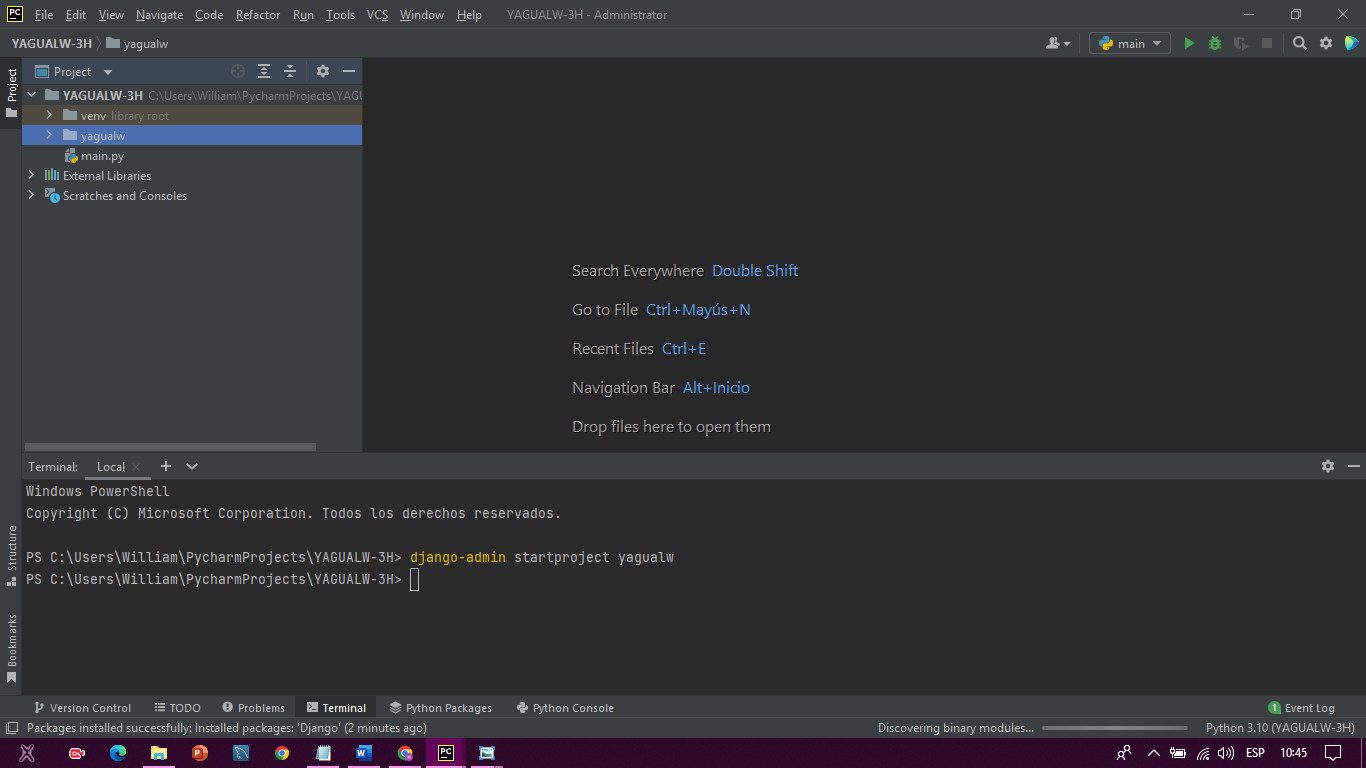


Le damos a la X y vemos que nos aparece Django ya instalado y le damos a OK



* 1. Crear un proyecto para programar en Django

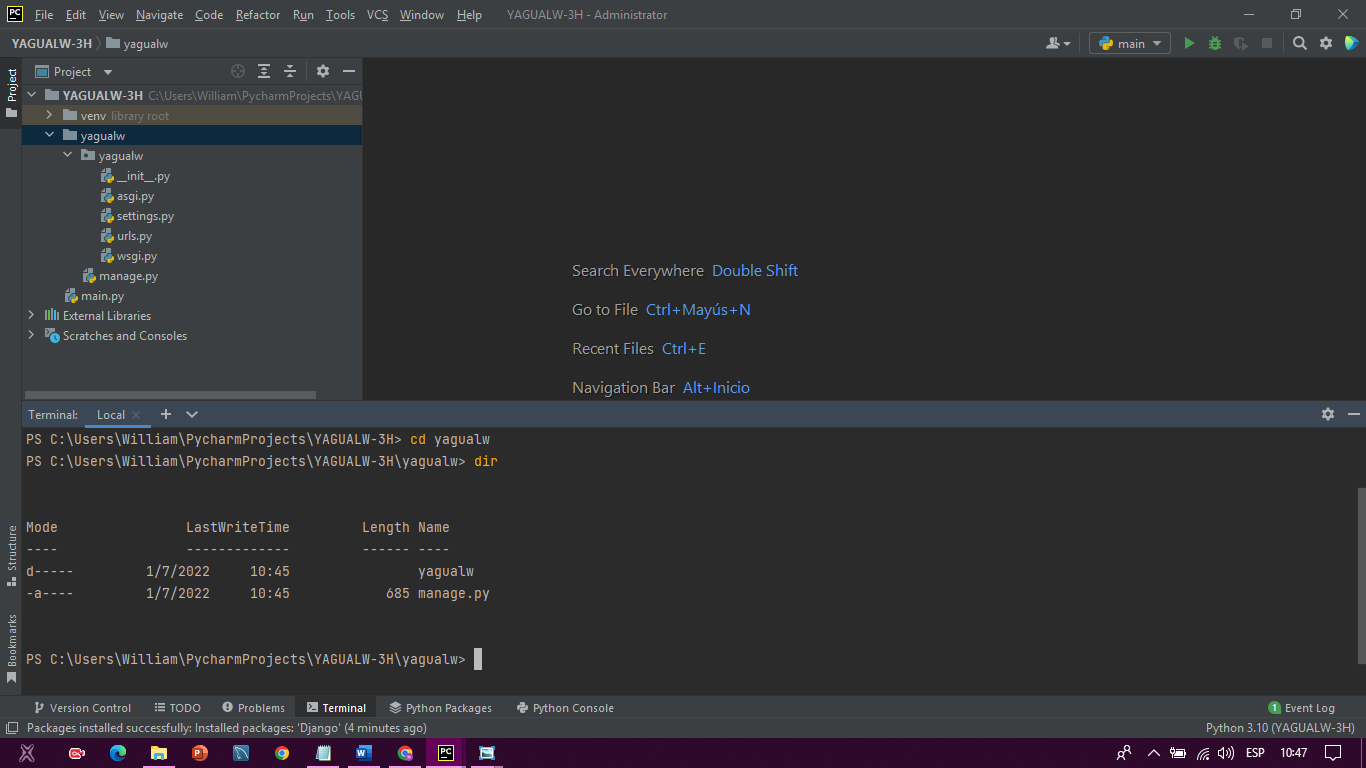
Para administrar Django y crear la carpeta de nuestro proyecto utilizamos la sentencia del rectángulo rojo y colocamos el nombre que deseaos



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



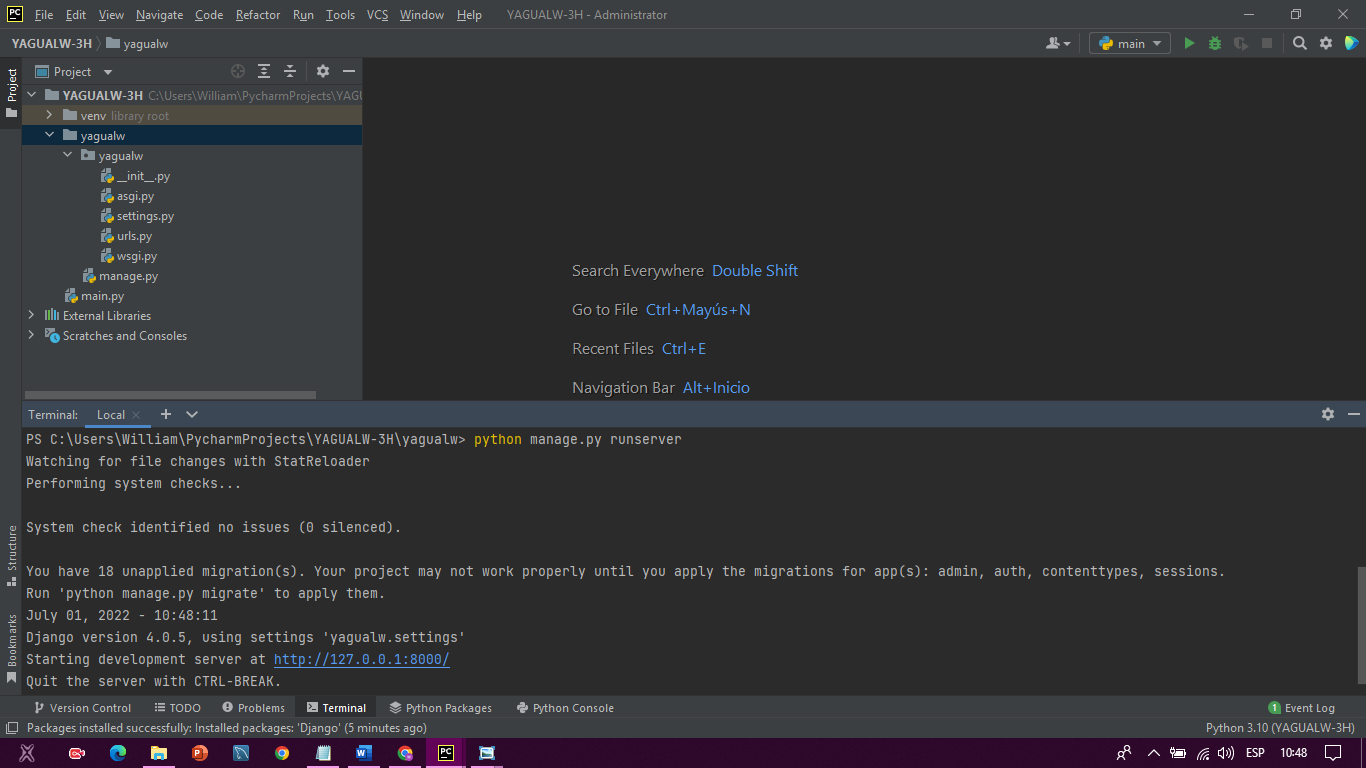
Utilizamos un dir



* 1. Ejecutar el proyecto y el mensaje de felicitaciones.

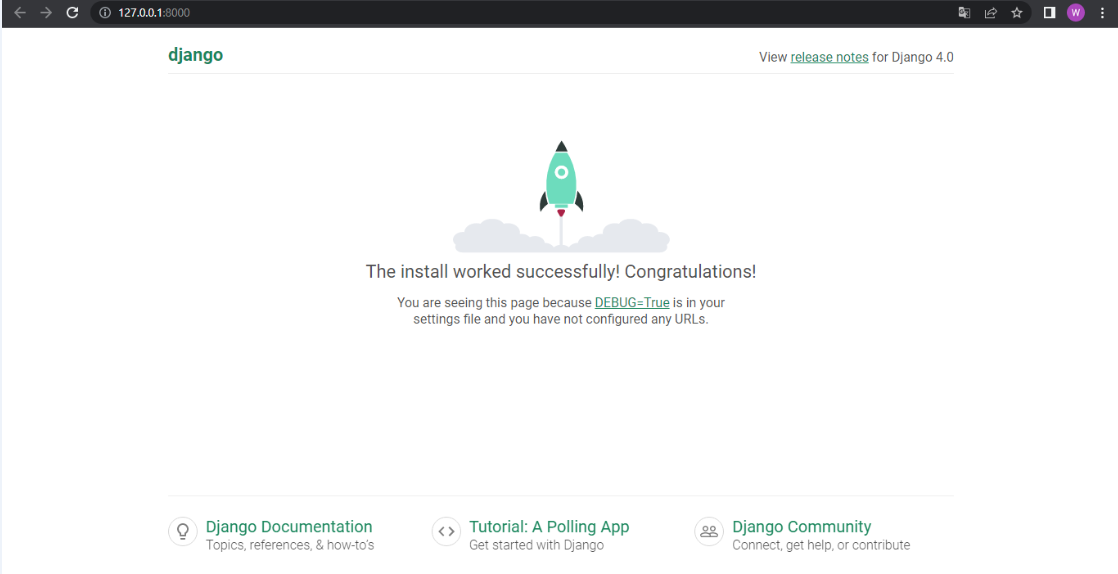
Utilizamos la sentencia Python manage.py runserver para ejecutar el proyecto

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



clic

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



* 1. 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



* 1. ¿Qué es la Carpeta Templates?

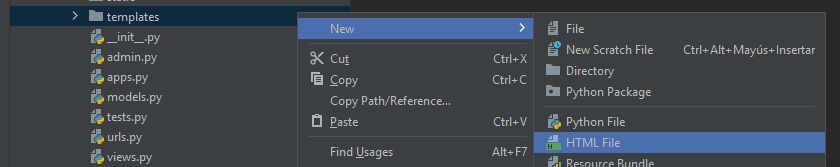
Es la subcarpeta que contiene plantillas que se muestran en el cuadro de diálogo Crear nuevo archivo. Puede crear y guardar plantillas personalizadas en la carpeta Templates. Contiene plantillas para la creación de archivos. Se puede guardar el archivo activo como una plantilla.

* 1. ¿Qué es la Carpeta static?

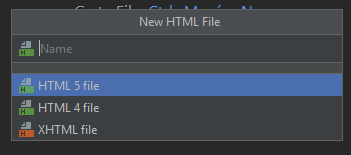
Es la carpeta que contiene todos los archivos de estilos: css y scss, archivos de comportamientos JavaScript, imágenes incluidas dentro del theme y el archivo checkout.

* 1. 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



* 1. Como se llaman a los CSS desde el archivo base HTML.

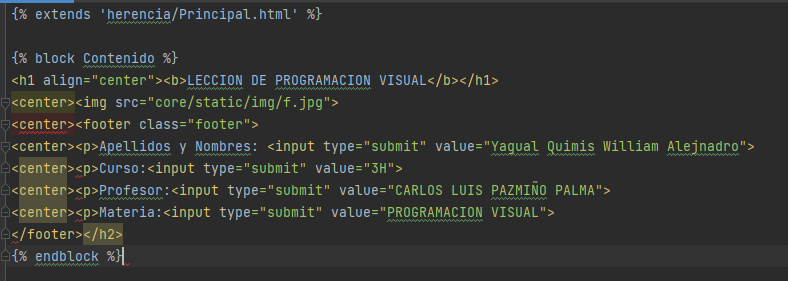
Utilizamos: **<link rel="STYLESHEET" href="{%static 'core/css/estilo1.css' %}">** lo de rojo es la ubicación del archivo css y arriba colocamos un {**% load static %}** como en el ejemplo que vemos a continuación



* 1. 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:

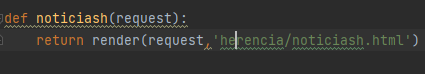


* 1. Crear un view que llame al HTML hijo

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:



* 1. 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:

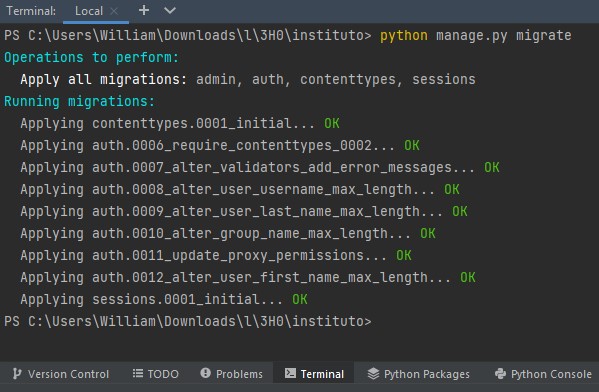


* 1. Integrar la aplicación APPS core al proyecto principal

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

* 1. 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

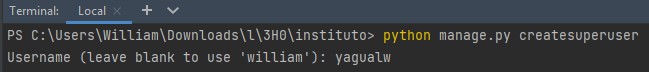


* 1. Crear un usuario para poder ingresar al Panel de Administración

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



Nos preguntara el nombre de usuario y colocamos el que deseamos



Después

nos

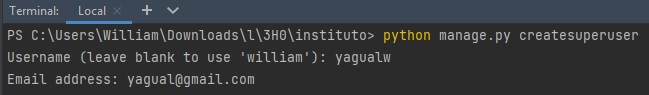
preguntara el

Email

y

lo

colocamos

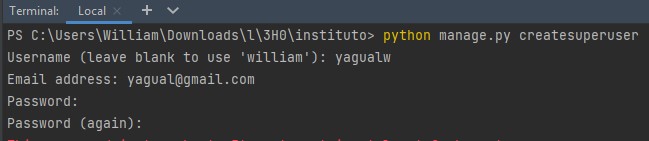
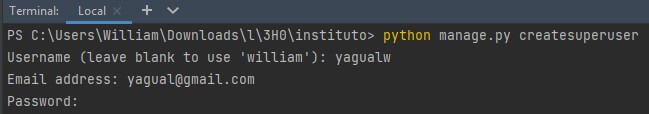


Después nos pide la contraseña y colocamos la de nuestra preferencia

P

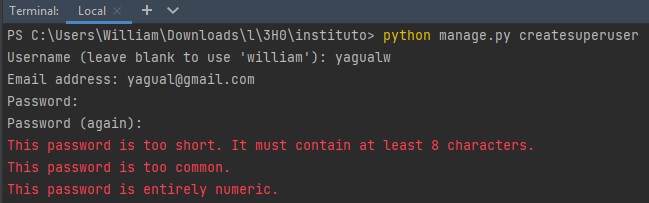
edirá

que la confirmes

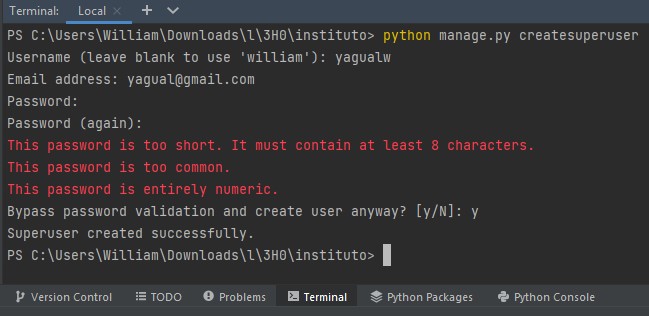


Si tu contraseña es muy corta te dirá que es corta y que debe contener al menos 8 caracteres

Que es muy común y puede ser alfanumérica



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

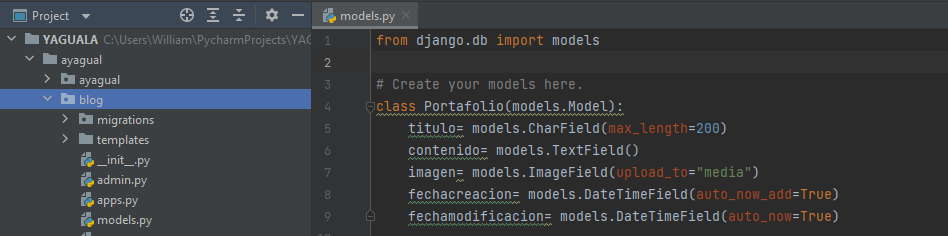


* 1. Que es un modelo en Django

Un modelo en Django es un tipo especial de objeto que se guarda en la base de datos.

* 1. Crear un modelo en Django.

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



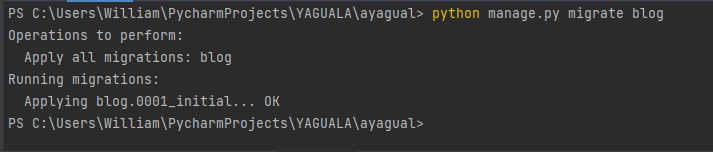
* 1. Migrar el Modelo a la base del Panel de Administración.

Migramos el modelo y observamos que se creo



* 1. Integrar el Modelo al Panel de Administración.

Migramos:

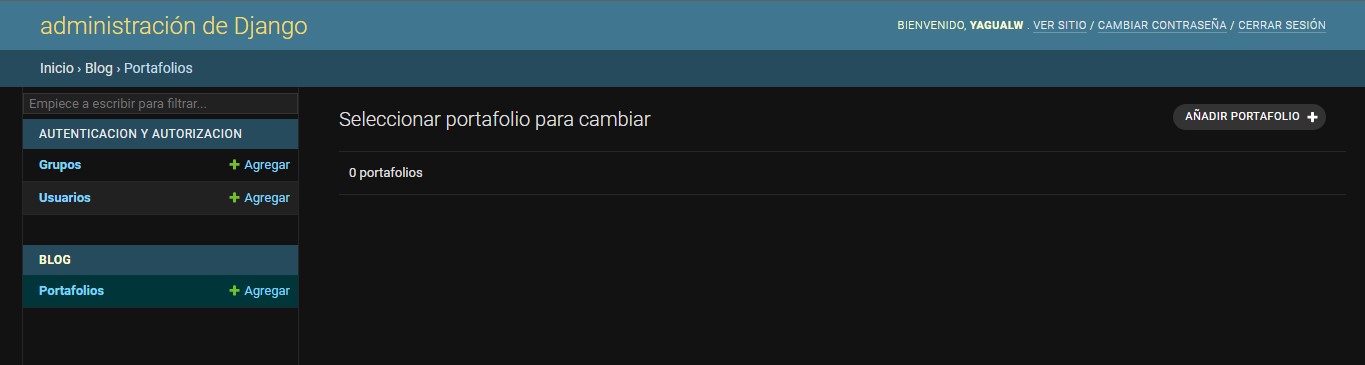


Y vemos que ya se creó:

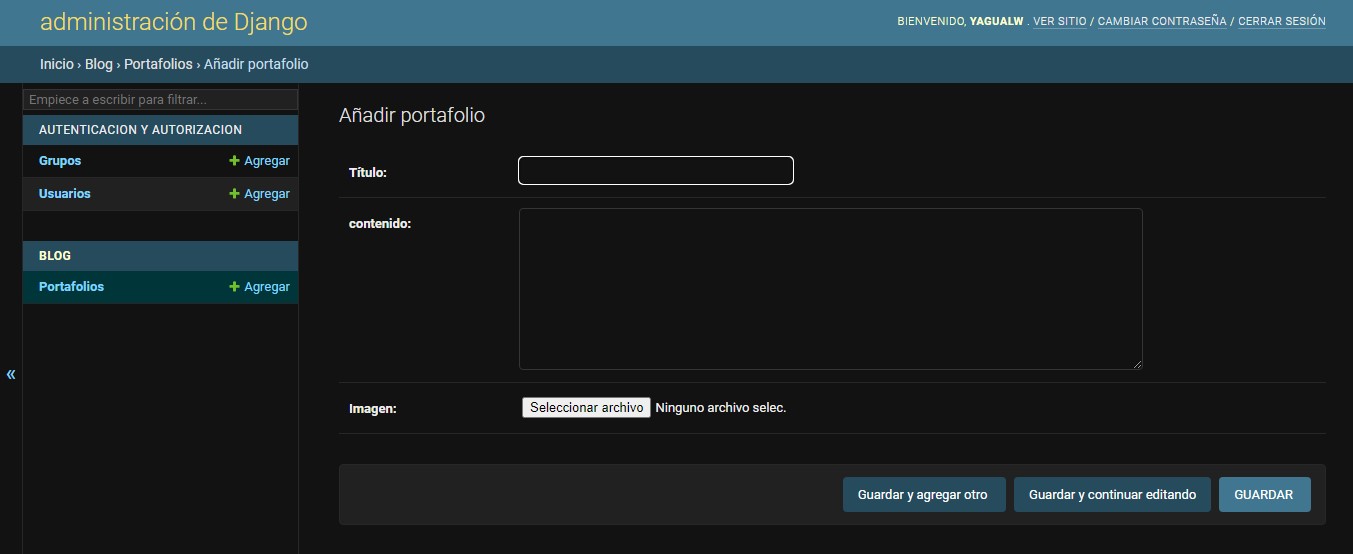


* 1. Ingresar información al modelo por el Panel de Administración.

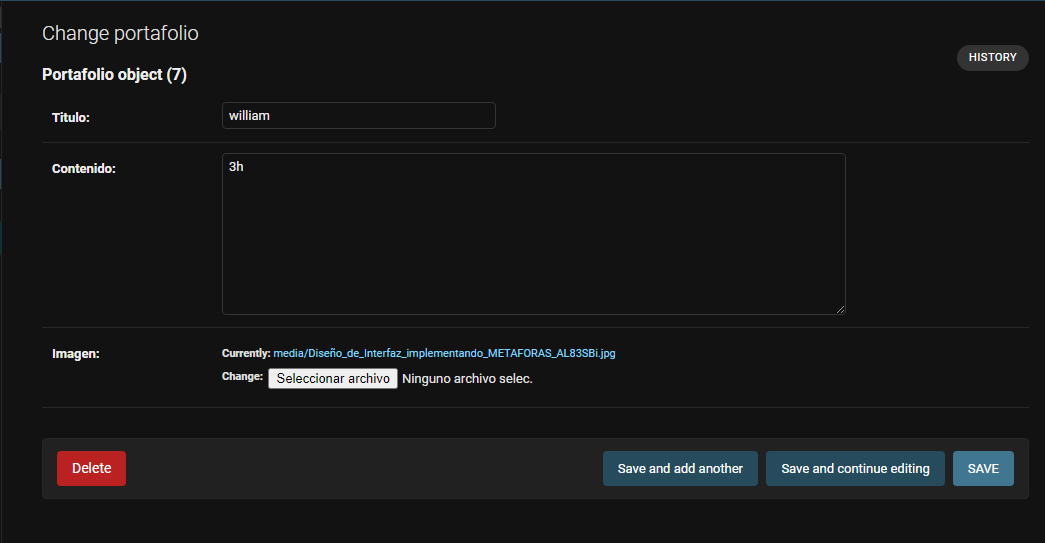
Le damos en **Añadir Portafolio**



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



Una vez llenado le damos a guardar



* 1. 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



* 1. Mostrar los datos guardados en el modelo al HTML hijo.



