

# PYTHON

**BATTLE ZONE**

**Iván Báscones Cubillo**



# IVÁN BASCONES CUBILLO

RESPONSABLE DEL DESARROLLO DE LA APLICACIÓN

Fecha: XX, MARZO, 2025



# ÍNDICE DE CONTENIDOS

<b>DESCRIPCIÓN GENERAL DEL PROYECTO</b> .....	3
<b>OBJETIVOS Y ALCANCE DEL PROYECTO</b> .....	5
<b>STACK TECNOLÓGICO Y ALTERNATIVAS EVALUADAS</b> .....	6
<b>MODELO DE DATOS</b> .....	7
<b>REQUISITOS DE LA APLICACIÓN</b> .....	11
<b>MANUAL DE INSTALACIÓN</b> .....	13
Instalación de SQLite3 .....	13
Instalación de Python .....	16
Instalación del Proyecto .....	18
Preparación del proyecto .....	20
Comandos Django .....	26
<b>CONCLUSIONES</b> .....	27
<b>EVOLUTIVOS DEL PROYECTO</b> .....	28



## DESCRIPCIÓN GENERAL DEL PROYECTO

**BattleZone** es una innovadora aplicación web diseñada para organizar y gestionar torneos de videojuegos, brindando una experiencia fluida y accesible para los jugadores. Esta plataforma está pensada para adaptarse a diferentes tipos de eventos, permitiendo la participación de un número de jugadores que oscila entre los 4 y los 32 participantes. Este rango garantiza una competencia sana y emocionante, desde pequeños encuentros hasta eventos de mayor escala.

La aplicación ha sido diseñada con un estilo de torneo bien estructurado que mantiene la emoción y la competitividad en cada ronda. Para garantizar un torneo equilibrado y justo, **BattleZone** se enfoca en una dinámica basada en puntos. Cada jugador se enfrentará en varias rondas, y dependiendo del resultado de cada enfrentamiento, se asignarán puntos de la siguiente manera:

### Sistema de Puntos

- **Victoria:** El jugador que gane el enfrentamiento recibirá un total de 3 puntos. Esta victoria le proporciona una ventaja significativa en la clasificación y lo acerca más al objetivo de convertirse en campeón del torneo.
- **Empate:** Si ambos jugadores terminan el enfrentamiento sin un ganador claro (por ejemplo, si las condiciones del juego permiten empates), cada jugador recibirá 1 punto. Este sistema asegura que, aunque no haya un ganador directo, ambos jugadores tengan una pequeña recompensa por su esfuerzo en la ronda.
- **Derrota:** En caso de derrota, el jugador perdedor no obtiene puntos. El jugador ganador, en cambio, recibe los 3 puntos que le permiten avanzar en la clasificación.

### Determinación del Ganador del Torneo

El ganador del torneo se determinará por la mayor cantidad de puntos acumulados a lo largo de todas las rondas y enfrentamientos. Al final del torneo, el jugador con más puntos será coronado como el campeón.



## Experiencia del Usuario

La experiencia del usuario en BattleZone está pensada para ser intuitiva y atractiva. Desde la inscripción hasta la visualización de los resultados, los jugadores se sentirán respaldados en cada paso del proceso. Además, los administradores de los torneos tienen acceso a herramientas avanzadas para personalizar las reglas del torneo, controlar los enfrentamientos y gestionar la puntuación de manera eficiente.



## OBJETIVOS Y ALCANCE DEL PROYECTO

### Objetivos

Los objetivos principales que tiene que lograr el proyecto son los siguientes:

- **Creación y gestión de usuarios:** En este apartado nos centraremos en la creación de los usuarios (jugadores) que van a participar en los torneos. En este objetivo incluimos el sistema de login del usuario.
- **Zona Home:** Este apartado es donde el usuario puede ver sus estadísticas, los últimos juegos (partidos) que ha participado y los resultados que ha obtenido. Además de que si es necesario el usuario pueda actualizar sus datos si es necesario.
- **Creación y gestión de torneos creados por gestores:** Creación y gestión de torneos que solamente tienen permisos o administradores. Dichas cuentas no podrán participar en los torneos, pero si crearlos o borrar los torneos que todavía no han empezado.
- **Sistema de puntos del torneo:** Creación del matchmaking de los jugadores basado en la cantidad de puntos que ha conseguido dichos jugadores y que se cumplan las recompensas que hay indicadas en el apartado de descripción
- **Sistema de ranking:** Crear un apartado en la web para que todos los usuarios y las personas que consulten la web vean los 5 mejores jugadores y sus estadísticas de manera pública.

### Alcance

El alcance (Limites) impuesto en este proyecto se compone de dos puntos principalmente:

- **Sistema de ranking:** En el caso de que no sea posible lo propuesto el objetivo anterior por tiempo o demasiada complejidad se descartaría este objetivo ya que es uno que no es importante para el proyecto
- **Zona Home:** En este caso se simplificaría los datos mostrados y se quitaría la vista de los últimos partidos mostrados en el caso de que sea demasiado complejo



# STACK TECNOLÓGICO Y ALTERNATIVAS EVALUADAS

## STACK TECNOLÓGICO

Las tecnologías utilizadas en este proyecto y por qué se ha decidió usar éstas y no otras:

- **Django (Python):** Es un framework de Python muy conocido por su versatilidad para crear aplicaciones web, además de que ya facilita el uso de BBDD y viene preparado para usar **SQLite3** directamente como BBDD.
- **HTML5:** Es el lenguaje de marcas usado para desarrollo de las páginas webs.
- **Bootstrap5 (CSS3):** un framework de CSS3 que es muy conocido y sencillo para decorar tus páginas webs.
- **SQLite3:** Es usada como nuestra BBDD para este proyecto por la sencillez de implementación ya como lo he comentado antes el framework **Django** ya tiene implementado esta tecnología

## Alternativas Evaluadas

Las tecnologías que han sido evaluadas como posibles sustitutos a las comentadas anteriormente son las siguientes:

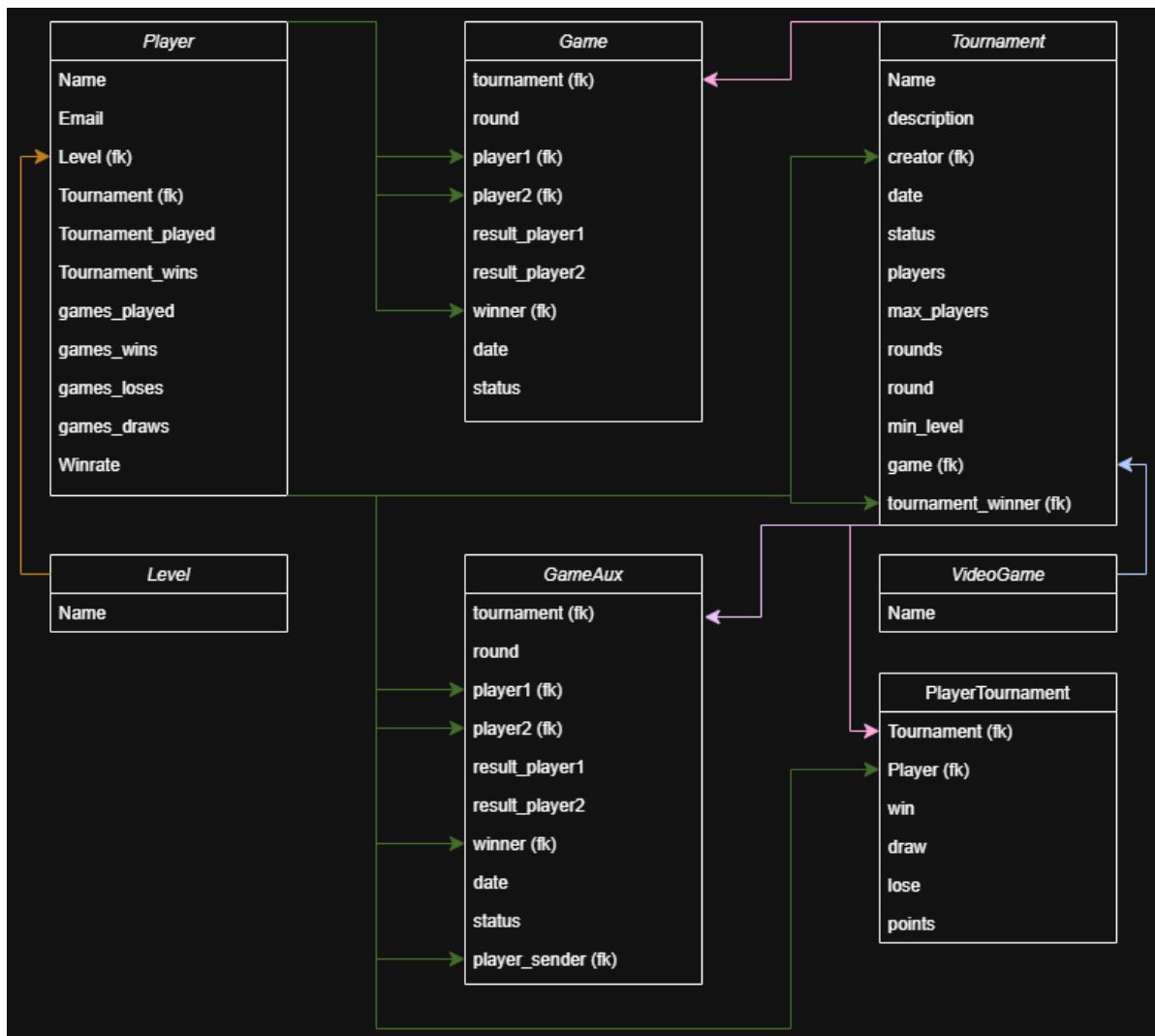
- **PostgreSQL:** Como alternativa a **SQLite3** como tecnología utilizada para gestionar la base de datos de este proyecto. Pero se puede actualizar a un futuro a esta tecnología si es el caso de que siga adelante el proyecto.
- **Flask (Python):** Este framework de Python se podía haber empleado para el hosting de la aplicación web, pero debido a que está basado en una tecnología obsoleta se ha dejado de lado.

# MODELO DE DATOS

## SQLite3

Se ha escogido **SQLite3** como base de datos debido a que el framework **Django** ya viene implementado con la integración a **SQLite3** y sería menos complicado realizar la creación, las migraciones y la integración de la BBDD al proyecto que, usando otros sistemas como **PostgreSQL**, **MySQL** o **MariaDB**.

## Esquema de la BBDD







La base de datos esta compuestas de un total de 7 tablas y todas, aunque no se indica tienen un campo id que es la clave primaria. Las

### Tablas Principales

En esta categoría son las tablas donde se guardará de manera definitiva los datos que se modifiquen, añaden en la BBDD. Son un total de 3 tablas:

- **Player:** Aquí se guarda toda la información relacionada del usuario.
  - **Name:** Nombre del usuario
  - **Email:** Correo del usuario
  - **Level (FK):** Nivel del jugador
  - **Tournament:** En el id del torneo
  - **Tournaments\_played:** Numero de los torneos que han finalizado
  - **Tournaments\_wins:** Numero de torneos ganados
  - **Games\_played:** Juegos que han participados en todos los torneos
  - **Games\_wins:** Juegos que han ganado
  - **Games\_draws:** Juegos que ha quedado empate
  - **Games\_loses:** Juegos que se ha perdido
  - **Winrate:** Porcentaje de juegos ganados comparado a los jugados
- **Game:** Aquí se guarda todos los registros de las partidas realizadas en cualquier torneo que se ha empezado.
  - **Tournament:**
  - **Round:** Ronda que se realiza el partido
  - **Player1 (FK):** Jugador 1
  - **Player2 (FK):** Jugador 2
  - **Result\_player1:** Resultado del primer jugador
  - **Result\_player2:** Resultado del segundo jugador
  - **Winner (FK):** Ganador del partido
  - **Date:** Fecha que se ha empezado el partido
  - **Status:** Indicamos el estado del juego (0 empezado y 1 terminado)



- **Tournament:** Esta tabla se almacena todos los torneos realizados en la aplicación web
  - **Name:** Nombre del torneo
  - **Description:** Descripción del torneo
  - **Creator (FK):** Creador del torneo
  - **Date:** Fecha de inicio del torneo
  - **Status:** Estado del torneo (0 publicado, 1 iniciado y 2 finalizado)
  - **Players:** Cantidad de jugadores apuntados
  - **Max\_players:** máximo de jugadores que se pueden apuntar
  - **Rounds:** Máximo de rondas que se va a jugar
  - **Round:** Ronda actual que se encuentra el torneo
  - **Min\_level (FK):** Nivel que deben tener los jugadores para participar
  - **Game (FK):** Videojuego que se va a usar para el torneo
  - **Tournament\_winner (FK):** Jugador ganador del torneo

### Tablas Categorías // Dificultad

Esta categoría de tablas se almacenará los datos de los videojuegos que la empresa quiera usar para sus torneos y los diferentes niveles que tiene los usuarios. Son un total de 2 tablas:

- **VideoGame:** Hay que guarda los juegos que el cliente va a usar para enfrentar a los usuarios en el torneo.
  - **Name:** Nombre del juego
- **Level:** se almacenará los diferentes niveles que separan a los jugadores para que participen de manera equitativa.
  - **Name:** Nombre de la dificultad

## Tablas Auxiliares

En esta categoría las tablas se almacenarán de manera provisional datos para no alterar las tablas principales hasta que sea de manera definitiva. Todos los datos cuando ya no tienen utilidad serán borrados. Son un total de dos tablas:

- **GameAux:** En esta tabla se almacenará los resultados que envíen los jugadores al gestor del torneo y se borrarán cuando se registró el resultado final del juego.
  - **Tournament (FK):** Id del torneo que se ha realizado el juego
  - **Round:** Ronda que se ha realizado el torneo
  - **Player1 (FK):** Jugador que ha participado en el juego
  - **Player2 (FK):** Jugador que ha participado en el juego
  - **Result\_player1:** Resultado del jugador 1
  - **Result\_player2:** Resultado del jugador 2
  - **Winner (FK):** Jugador que ha resultado ganador del encuentro
  - **Date:** Fecha de inicio del partido
  - **Status:** Estado del partido (0 empezado y 1 terminado)
  - **Player\_sender (FK):** Jugador que ha enviado el reporte al gestor del partido
- **PlayerTournament:** En esta tabla se almacena los resultados de los jugadores en un torneo para no modificar los datos finales de la tabla Player. Los datos son borrados cuando un torneo tiene a un ganador y cuando toda la información recogida es pasada a la tabla **Player**.
  - **Tournament (FK):** Id del torneo que está participando el jugador
  - **Player (FK):** Id del jugador participante
  - **Win:** Juegos que ha ganado el jugador
  - **Draw:** Juegos que ha empatado el jugador
  - **Lose:** Juegos que ha perdido el jugador
  - **Points:** Total de puntos acumulados



# REQUISITOS DE LA APLICACIÓN

## Requisitos Funcionales

Los requisitos funcionales que tiene que cumplir la aplicación son los siguientes:

- **Sistema de registro y control de acceso:** Todos los participantes u personas que quieren participar en cualquier torneo será necesaria la creación de una cuenta que pide como datos, el nickname o username que vaya a utilizar el participante, su correo electrónico, indicar que nivel está el participante y una contraseña. Estos usuarios creados no tendrán acceso a la BBDD de la web y cuando estén participando solo modificarán las tablas auxiliares indicadas en el apartado anterior.
- **Interfaz de Usuario:** La interfaz de usuario cuando inicia sesión se ve sus estadísticas de todas las partidas y torneos que ha jugado con éxito. Además de visualizar las ultimas partidas y el resultado obtenido contra el jugador que te enfrentaste en ese enfrentamiento. Además de que disponen de un apartado para ver el ranking de los 5 jugadores con más participaciones, más ratio de victoria y más torneos ganados. Todos los usuarios pueden ver los torneos que son de su mismo nivel que marcaron cuando crearon su cuenta de usuario. Solo se podrán apuntar a torneos que estén sin empezar y solamente a uno.
- **Torneo:** El torneo tiene dos vistas principales uno por parte de administrador y otro por parte de jugador. En ambas vistas tendrán disponible la tabla que se ira actualizando la información y se sabrá en todo momento en que posición está el jugador y la del resto de jugadores. Las características que tiene cada vista son las siguientes:
  - **Administrador:** El administrador ve la lista de todos los juegos de la ronda en la que se encuentra el torneo e ira recibiendo los reportes enviados por parte de los usuarios y en el caso de que no se concuerden los resultados enviados por los jugadores el administrado puede decidir el resultado de la partida después de confirmar cuál de los dos resultados enviados es el correcto. Es el encargado de iniciar el torneo, gestionar las rondas y finalizar el torneo cuando todo se ha terminado.
  - **Jugador:** Cuando el administrador inicie el torneo entrará a la zona del torneo que vera cuando el administrador cuál será su contrincante y podrá enviar el resultado de la partida.



- **Creación de torneos:** Solo los usuarios con permisos de administrador estarán capacitados para crear y dirigir los torneos que haya creado.

## Requisitos no Funcionales

Los requisitos no funcionales que tiene que cumplir la aplicación son las siguientes:

- **Seguridad:** Cuando un usuario crea una cuenta de usuario la contraseña que pone en la base de datos está cifrada. Además del uso de tablas temporales para no afectar a los datos originales que tiene la BBDD y posibilidad de perder datos por culpa de cualquier fallo humano.
- **Mantenimiento:** En el caso de que sea necesario el mantenimiento o implementar mejoras al proyecto. La aplicación esta dividida en tres módulos (main, tournaments, users). Cada parte tiene las funciones, vistas y tablas relacionadas al nombre del módulo: tournaments = torneos, users = usuarios y main = página principal. En el caso de actualizar la vista o la Base de datos en cada módulo tienen los siguientes archivos models.py (Tablas de la Base de datos) y views.py (Funciones de las vistas). En el caso que se quiera actualizar la vista en cada módulo tienes la carpeta templates que contiene todo el código **HTML5**.
- **Usabilidad:** La web tiene todos los menús y opciones en sitios que son habituales a los usuarios interactuar. Como los botones de login y crear cuenta a la derecha arriba o el menú con las diferentes opciones que tienes cuando una cuenta está conectada.

# MANUAL DE INSTALACIÓN

## Instalación de SQLite3






Para poder modificar la BBDD necesitamos instalar SQLite3 en nuestro sistema operativo. Para instalarlo correctamente hay que hacer los siguientes pasos:

1. Tenemos que ir a la web oficial de y descargamos la siguiente opción en el caso de que tu ordenador tenga instalado Windows y con una estructura de x64 bits. [Enlace descarga](#)

### Precompiled Binaries for Windows

<a href="#">sqlite-dll-win-x86-3490100.zip</a> (1.02 MiB)	32-bit DLL (x86) for SQLite version 3.49.1. (SHA3-256: 1dac243f154828327d6a250b74cf115e4f6a4cbfe83dbc488d45510b59e7a157)
<a href="#">sqlite-dll-win-x64-3490100.zip</a> (1.28 MiB)	64-bit DLL (x64) for SQLite version 3.49.1. (SHA3-256: ec8fb7976d9c4bc495c4a142da05e97c4d6dc6c1205877adcce0bd5b191026e2)
<a href="#">sqlite-tools-win-x64-3490100.zip</a> (6.12 MiB)	A bundle of command-line tools for managing SQLite database files, including (1) t (SHA3-256: f80860a651026229c8d6ba76cd8c3af74e114cecf71ad9fc5bf938adcd4ca0b9)

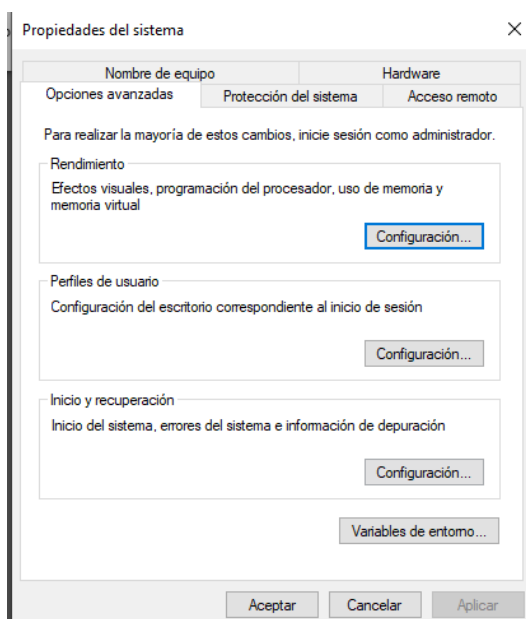
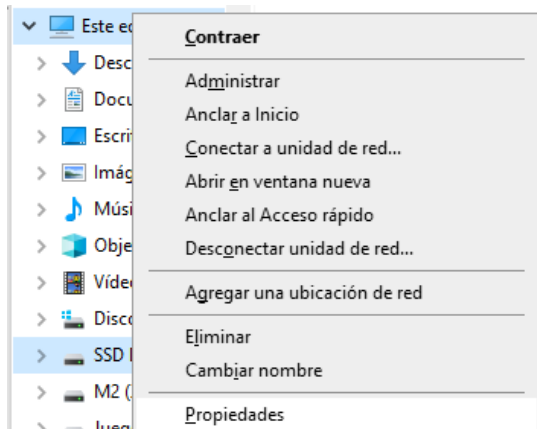
2. Descomprimos el zip donde queramos tener SQLite3 y tiene que contener los siguientes tres ejecutables. Los archivos def y dll se crearán cuando todo este bien instalado

Este equipo > M2 (X:) > SQLite3					Buscar en SQLite3	
Nombre	Fecha de modificación	Tipo	Tamaño			
 sqldiff.exe	08/10/2024 17:42	Aplicación	3.043 KB			
 sqlite3.def	08/10/2024 17:41	Archivo DEF	9 KB			
 sqlite3.dll	08/10/2024 17:41	Extensión de la ap...	3.108 KB			
 sqlite3.exe	08/10/2024 17:42	Aplicación	3.686 KB			
 sqlite3_analyzer.exe	08/10/2024 17:42	Aplicación	4.286 KB			

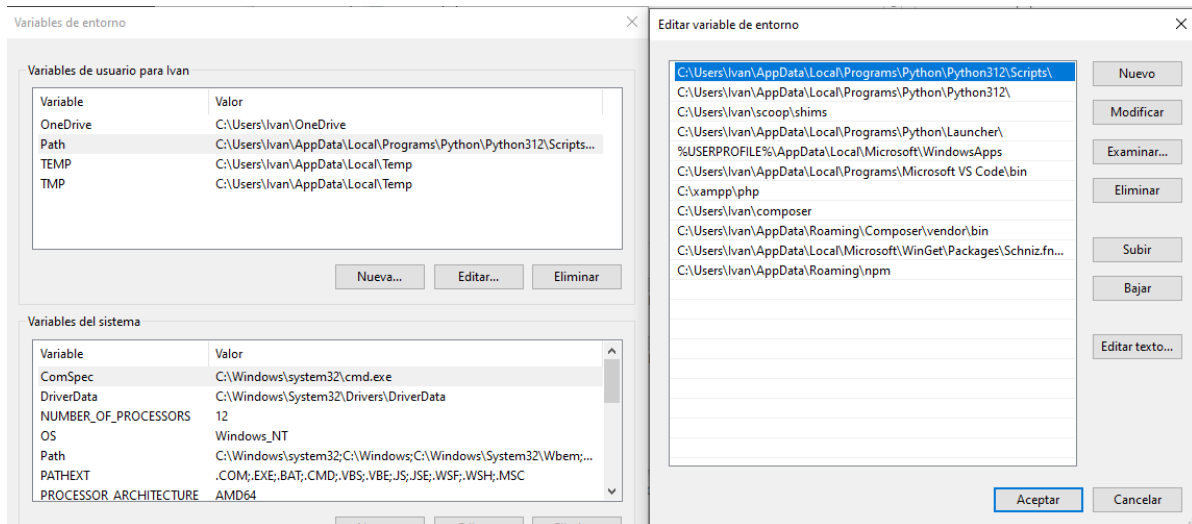
- Después tenemos que crear una variable de entorno PATH, para ello nos dirigiremos a Este Equipo -> Propiedades -> Configuración Avanzada del sistema -> Variables de Entorno.

## Acerca de

### Opciones de configuración relacionadas

[Configuración de BitLocker](#)[Administrador de dispositivos](#)[Escritorio remoto](#)[Protección del sistema](#)[Configuración avanzada del sistema](#)[Cambiar el nombre de este equipo \(avanzado\)](#)

4. Cuando entremos en variables de Entorno tenemos que entrar en las variables PATH y nos sale la siguiente ventana



5. Le damos a nuevo y lo que tenemos que insertar la ruta donde descomprimos el zip que nos instalamos en el paso uno

X:\SQLite3

6. Reinicia el ordenador y comprobamos si funciona usando el comando sqlite3 en el cmd

```
C:\> Símbolo del sistema - sqlite3

Microsoft Windows [Versión 10.0.19045.5247]
(c) Microsoft Corporation. Todos los derechos reservados.

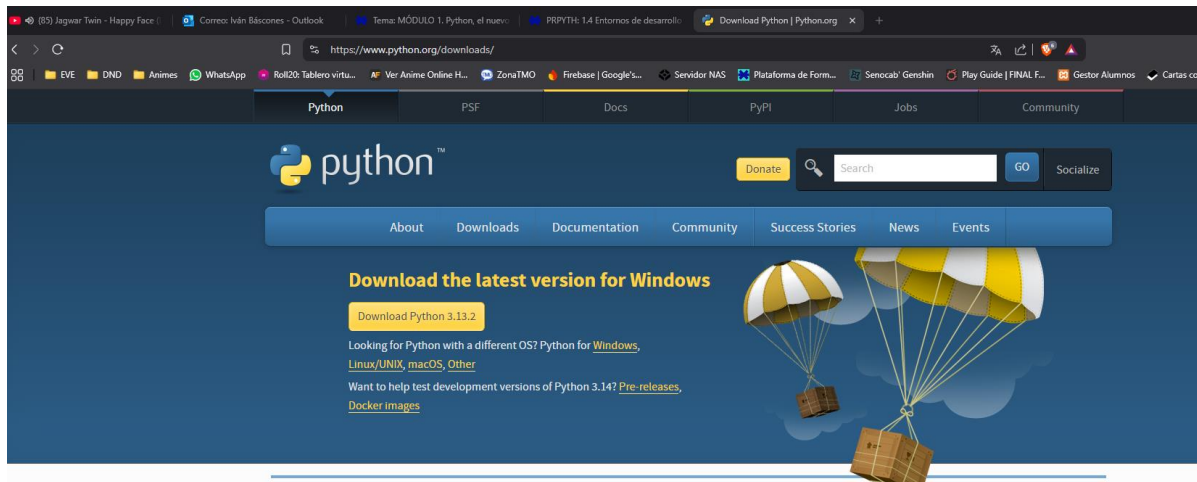
C:\Users\Ivan>sqlite3
SQLite version 3.46.1 2024-08-13 09:16:08 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> _
```



## Instalación de Python

Para instalar Python en el dispositivo tenemos que seguir los siguientes pasos de instalación:

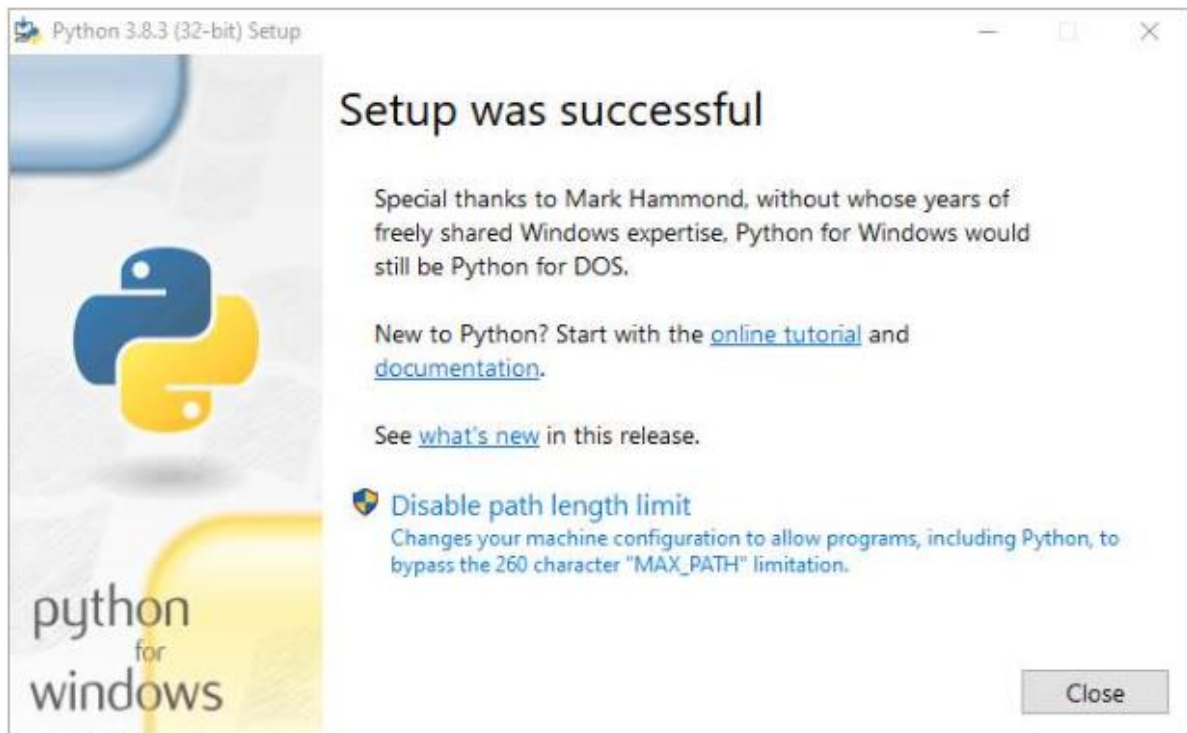
1. Primero de todo tenemos que ir a su página oficial y descargamos la última versión de Python. [Enlace Descarga](https://www.python.org/downloads/)



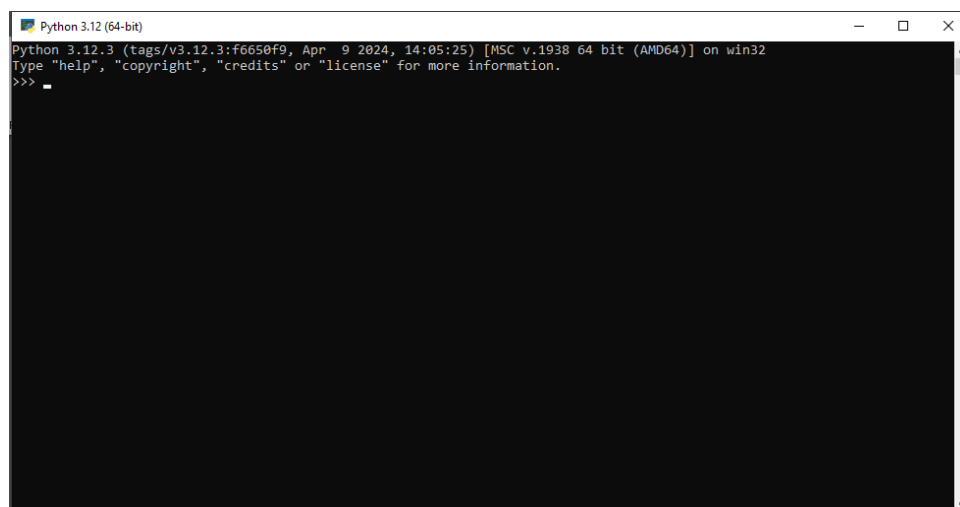
2. Ejecutamos el instalador y damos a las opciones "Install launcher for all users (recommended)" y "Add Python 3.x to PATH" y pulsamos sobre Install Now



- Después de eso empezara a instalar el intérprete de Python. Nos saldrá la venta de que se ha instalado correctamente Python, tras lo cual desactivaremos la longitud máxima de las rutas en Windows, pulsando sobre "Disable path lenght limit", que nos aparece en esa misma ventana, y, por último, pulsaremos sobre el botón "Close".

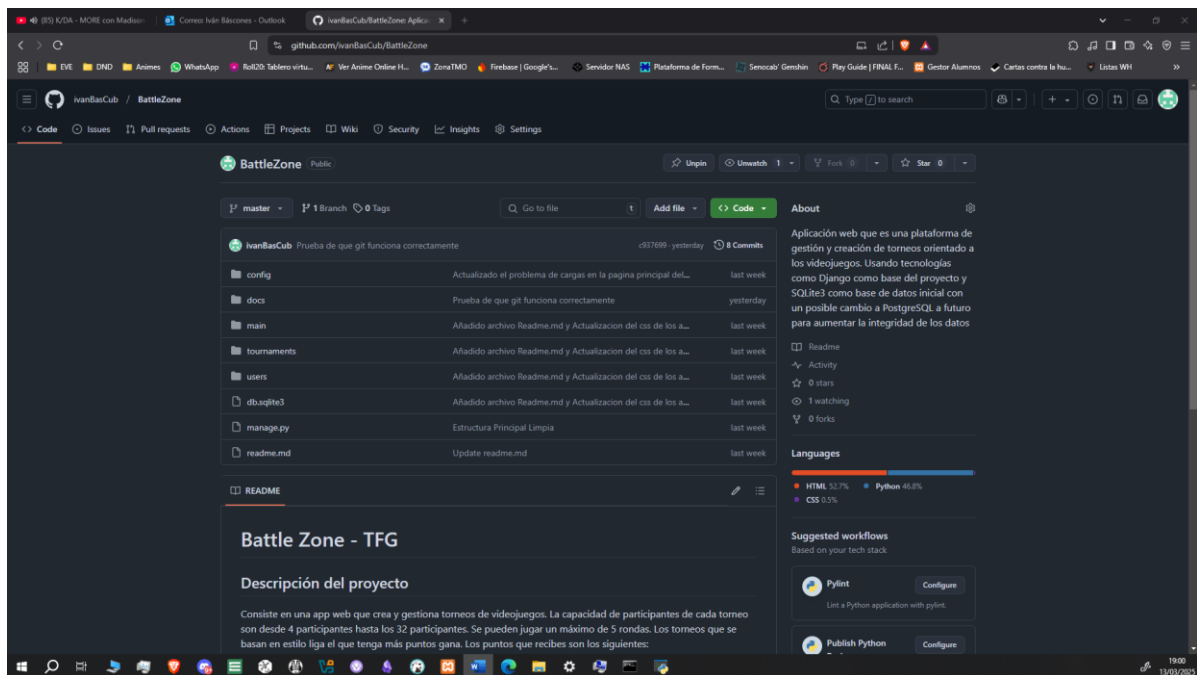


Con esto ya tendremos instalado el intérprete y el Python Shell que nos permitirá empezar programar en Python correctamente.

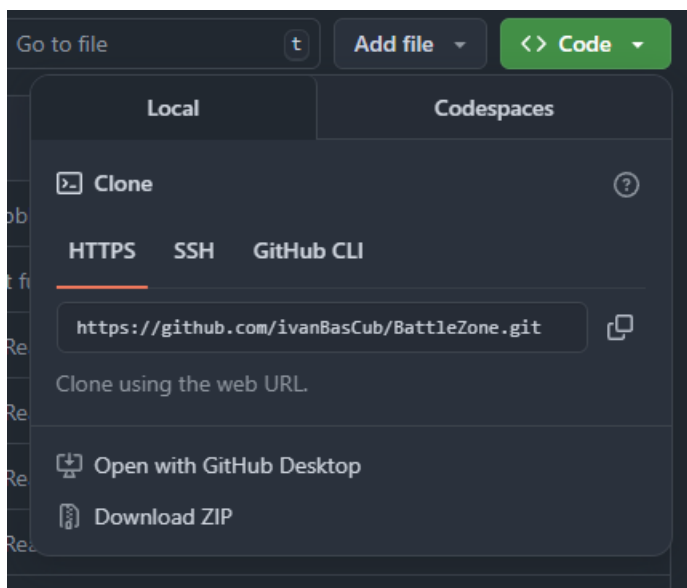


## Instalación del Proyecto

Para poder instalar el proyecto tienes que ir al siguiente repositorio de GitHub [BattleZone](https://github.com/ivanBasCub/BattleZone). En este repositorio podrás descargar todo el contenido de la aplicación de manera gratuita

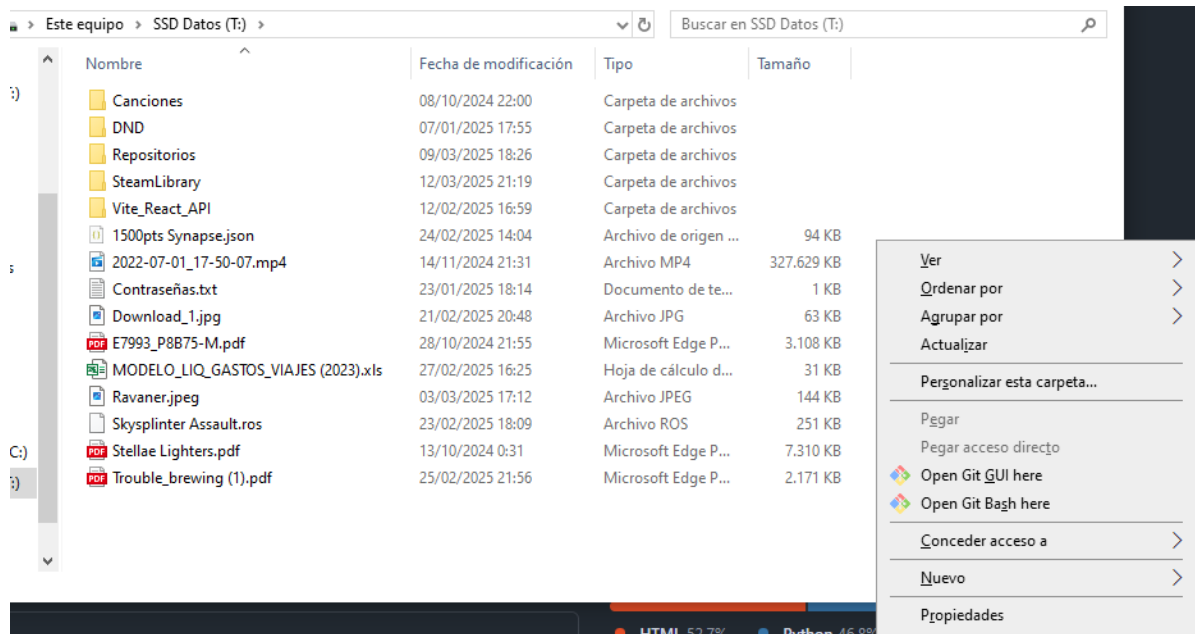


En la opción de "code" puedes clonar el repositorio si quieres mantener actualizado y solamente quieres tenerlo das a la opción de "download ZIP"



En el caso de que has seleccionado la opción de "download ZIP" simplemente tienes que descomprimir el zip donde lo vallas a utilizar. En el caso de que decidas clonar el repositorio necesitaras tener instalado en tu equipo GIT y configurado.

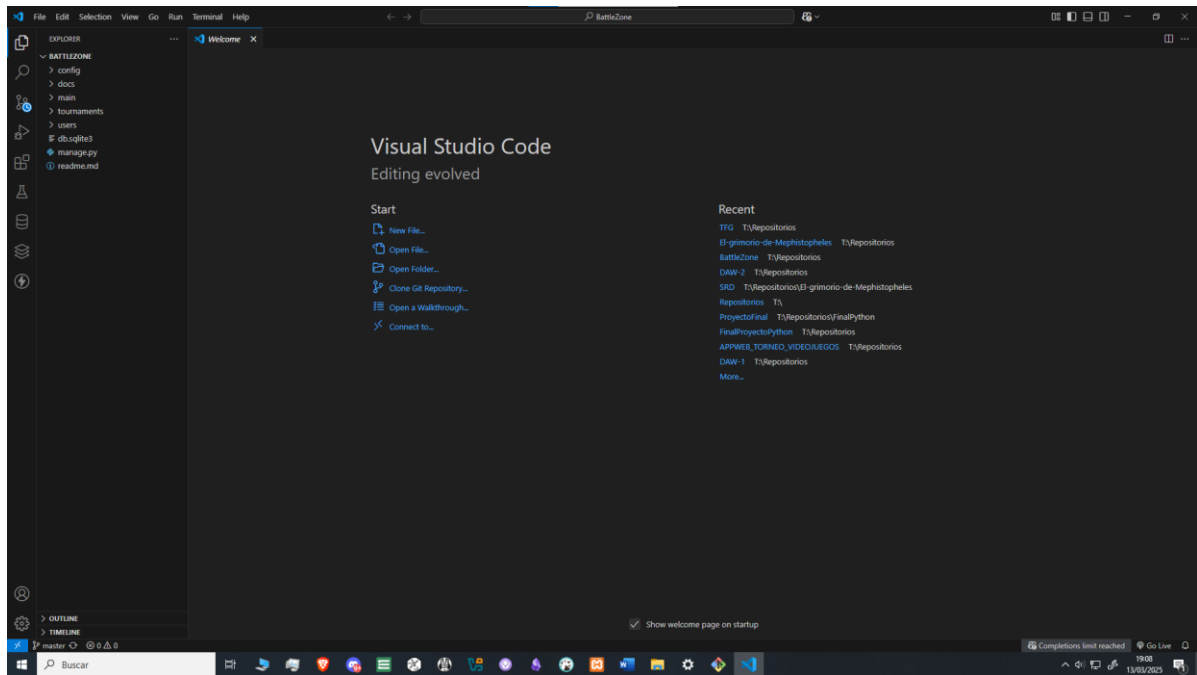
En la carpeta que vayas a clonar el repositorio necesitas abrir una consola git bash y poner el siguiente comando



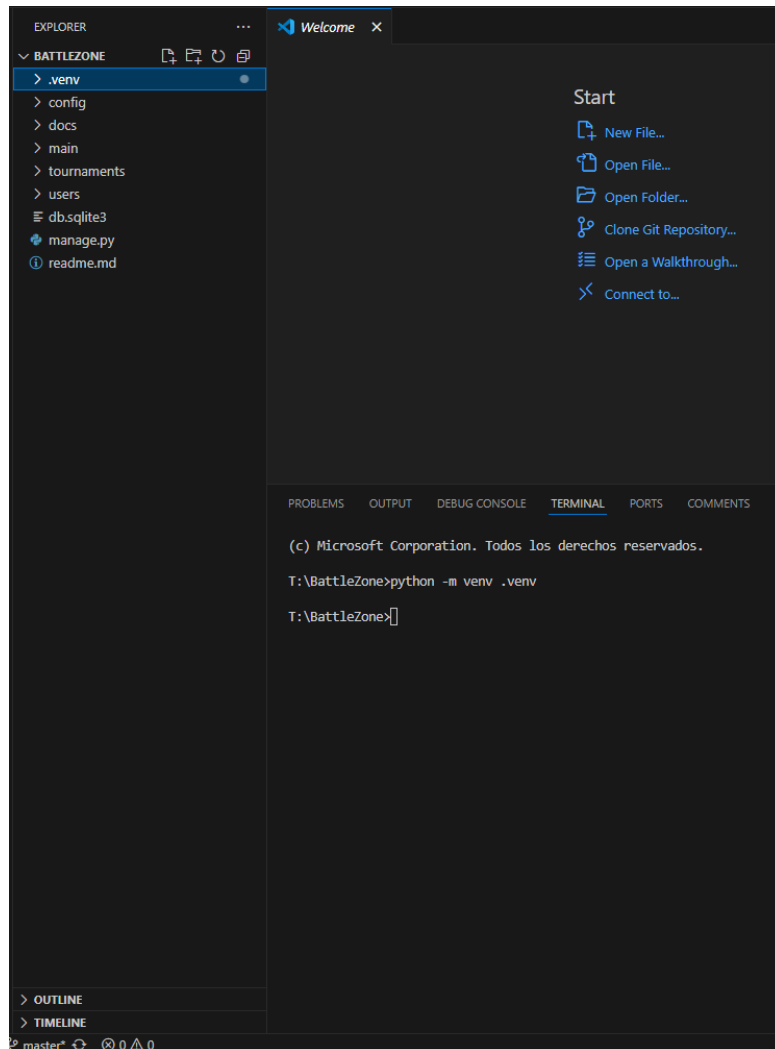
```
Ivan@DESKTOP-L036KKG MINGW64 /t
$ git clone https://github.com/ivanBasCub/BattleZone.git
```

## Preparación del proyecto

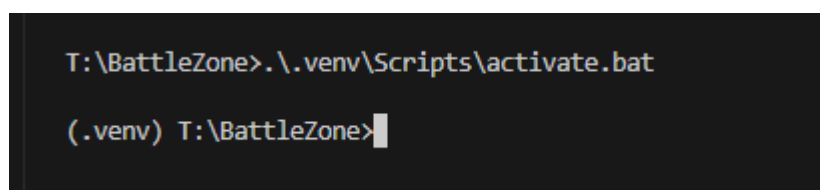
Después de descargar el proyecto lo abrimos con nuestro entorno de desarrollo de preferencia (VSCode, Pycharm, Anaconda, ...). En mi caso será VSCode. Nos tendría que quedar el Workspace de la siguiente forma:



Abrimos una terminal de comandos y escribimos el siguiente comando:



Este comando nos permite crear un entorno de virtualización para que no se mezcle los framework instalados en Python y no nos causen problemas de compatibilidad. Después de eso usamos el siguiente comando para activar el entorno virtual. Que se nos indicara con un (.venv) en la terminal



Ya que tenemos instalado el entorno virtual instalamos el único framework que necesita el proyecto **Django** con el siguiente comando:

```
(.venv) T:\BattleZone>pip install django  
Collecting django
```

Al final de la instalación tendrá que poner lo siguiente:

```
Installing collected packages: tzdata, sqlparse, asgiref, django  
Successfully installed asgiref-3.8.1 django-5.1.7 sqlparse-0.5.3 tzdata-2025.1  
  
[notice] A new release of pip is available: 24.0 -> 25.0.1  
[notice] To update, run: python.exe -m pip install --upgrade pip  
  
(.venv) T:\BattleZone>
```

Tenemos casi todo listo para empezar. Lo único que nos queda es migrar la BBDD y podemos ya empezar usamos el siguiente comando

```
(.venv) T:\BattleZone>python manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, auth, contenttypes, main, sessions, tournaments, users  
Running migrations:  
  Applying contenttypes.0001_initial... OK  
  Applying auth.0001_initial... OK  
  Applying admin.0001_initial... OK  
  Applying admin.0002_logentry_remove_auto_add... OK  
  Applying admin.0003_logentry_add_action_flag_choices... OK  
  Applying contenttypes.0002_remove_content_type_name... OK  
  Applying auth.0002_alter_permission_name_max_length... OK  
  Applying auth.0003_alter_user_email_max_length... OK  
  Applying auth.0004_alter_user_username_opts... OK  
  Applying auth.0005_alter_user_last_login_null... 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 main.0001_initial... OK  
  Applying sessions.0001_initial... OK  
  Applying users.0001_initial... OK  
  Applying tournaments.0001_initial... OK
```



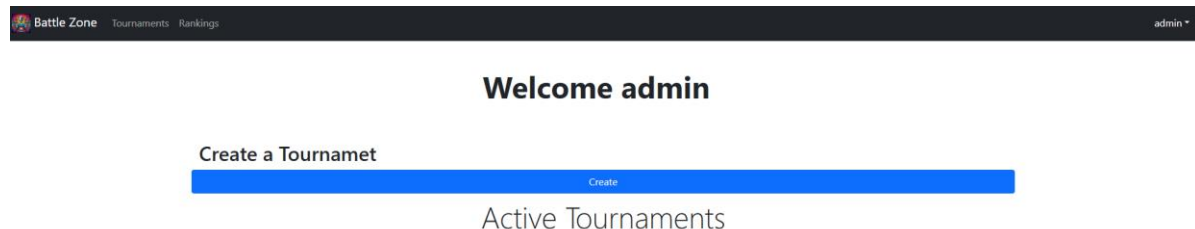
Para que todo funcione bien hay que crear una cuenta administradora usamos el siguiente comando:

```
(.venv) T:\Repositorios\BattleZone>python manage.py createsuperuser
Username (leave blank to use 'ivan'): admin
Email address: a@a.es
Password:
Password (again):
Superuser created successfully.
```

Después arrancamos el servidor para iniciar sesión con la cuenta administrador con el comando `python manage.py runserver`

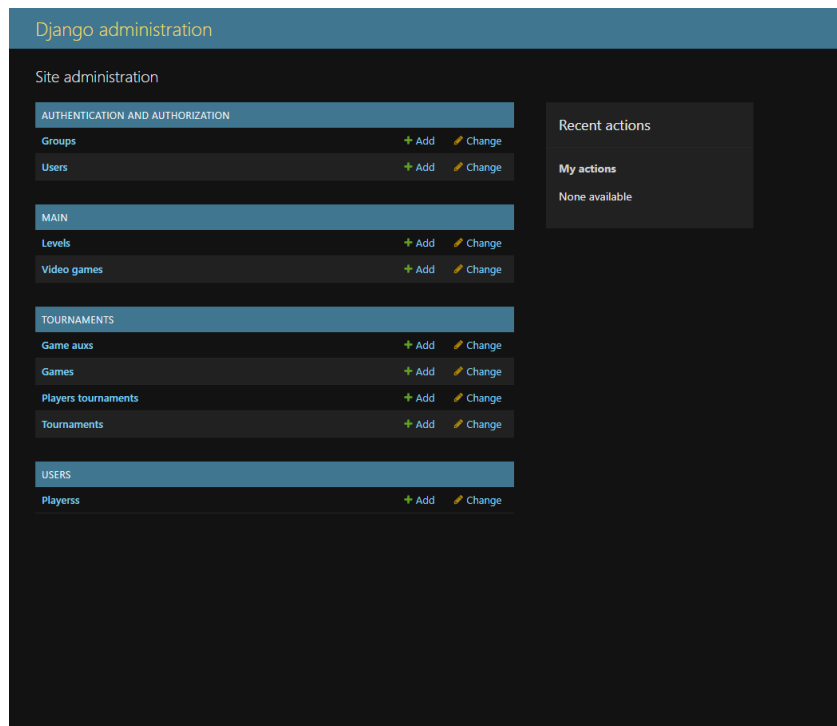
```
(.venv) T:\Repositorios\BattleZone>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 24, 2025 - 18:05:47
Django version 5.1.3, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

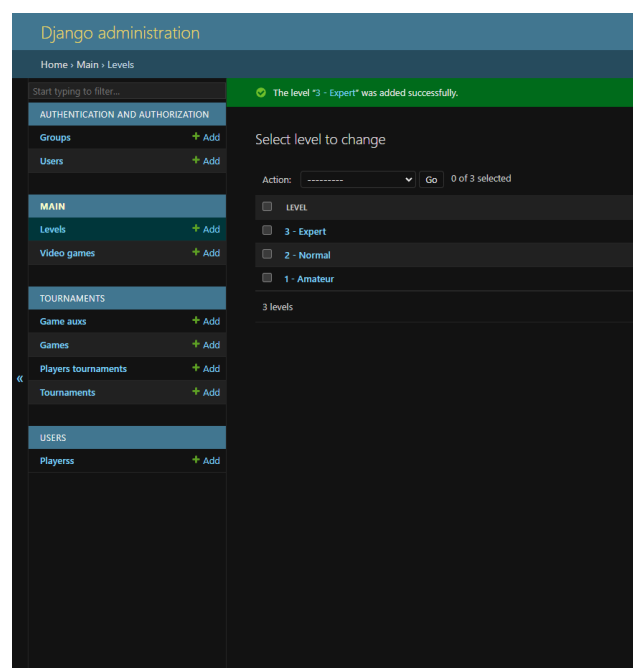




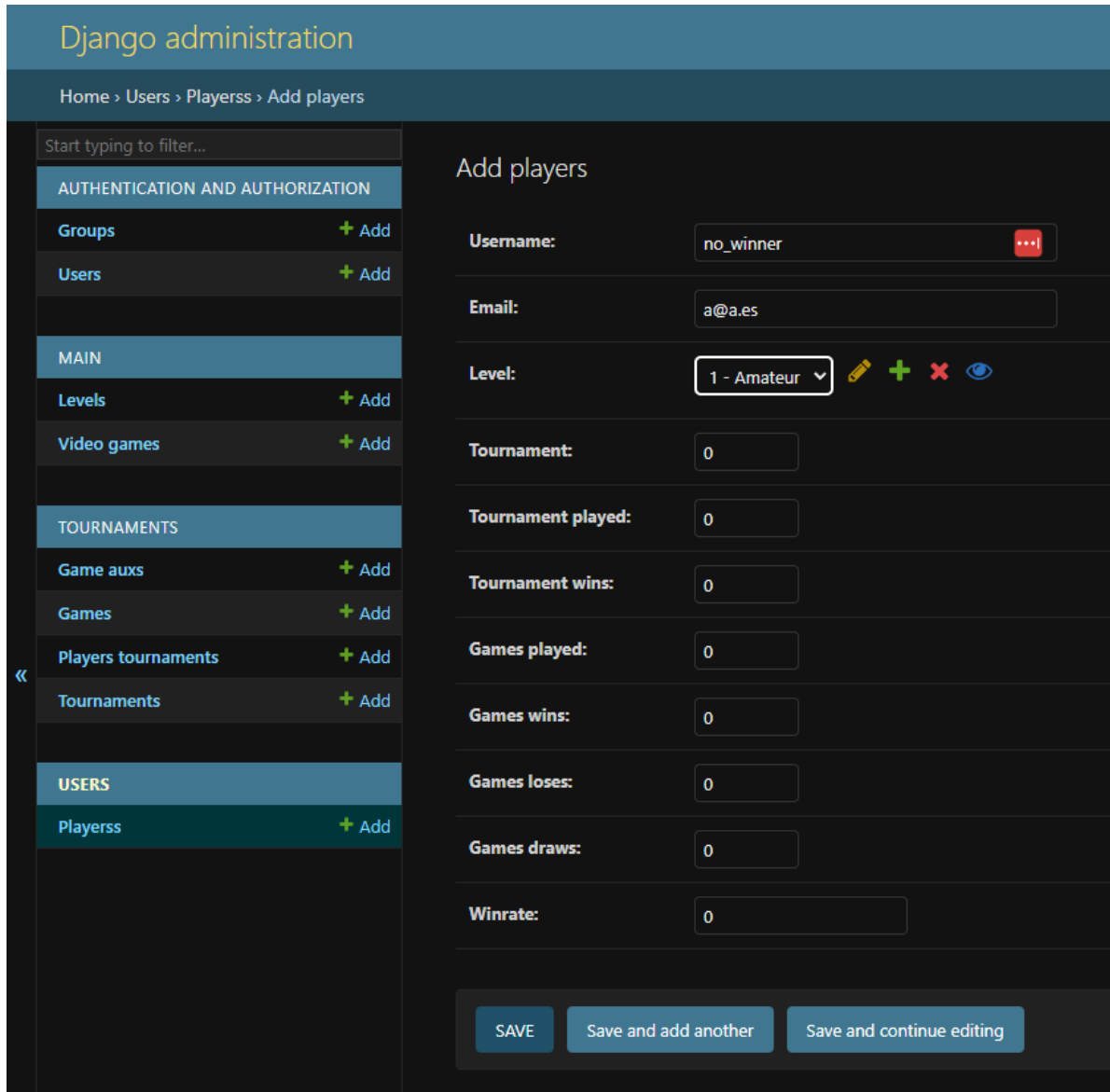
Nos vamos a la zona de administrador y aquí podemos ver las tablas del proyecto. Vamos a añadir datos en tres. Levels, Videogames y Players.



Empezamos con Levels introduciremos los diferentes niveles que diferenciarian a los jugadores y la dificultad de los torneos dándole al botón de add:



La siguiente tabla que vamos a introducir datos es en la tabla Players. Vamos a crear un usuario llamado "no\_winner" con el objetivo de que es un usuario vacío que se usara para crear los torneos e indicar el estado de empate en los resultados de los juegos



The screenshot shows the Django administration interface for a system called 'Battle Zone'. The left sidebar contains a navigation menu with categories: AUTHENTICATION AND AUTHORIZATION (Groups, Users), MAIN (Levels, Video games), TOURNAMENTS (Game auxs, Games, Players tournaments, Tournaments), and USERS (Playerss). The 'Playerss' link is highlighted. The main content area is titled 'Add players' and contains a form with the following fields: Username (no\_winner), Email (a@a.es), Level (1 - Amateur), Tournament (0), Tournament played (0), Tournament wins (0), Games played (0), Games wins (0), Games loses (0), Games draws (0), and Winrate (0). At the bottom of the form are three buttons: 'SAVE', 'Save and add another', and 'Save and continue editing'.

Django administration

Home > Users > Playerss > Add players

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

- Groups + Add
- Users + Add

**MAIN**

- Levels + Add
- Video games + Add

**TOURNAMENTS**

- Game auxs + Add
- Games + Add
- Players tournaments + Add
- Tournaments + Add

**USERS**

- Playerss + Add

**Add players**

Username: no\_winner

Email: a@a.es

Level: 1 - Amateur

Tournament: 0

Tournament played: 0

Tournament wins: 0

Games played: 0

Games wins: 0

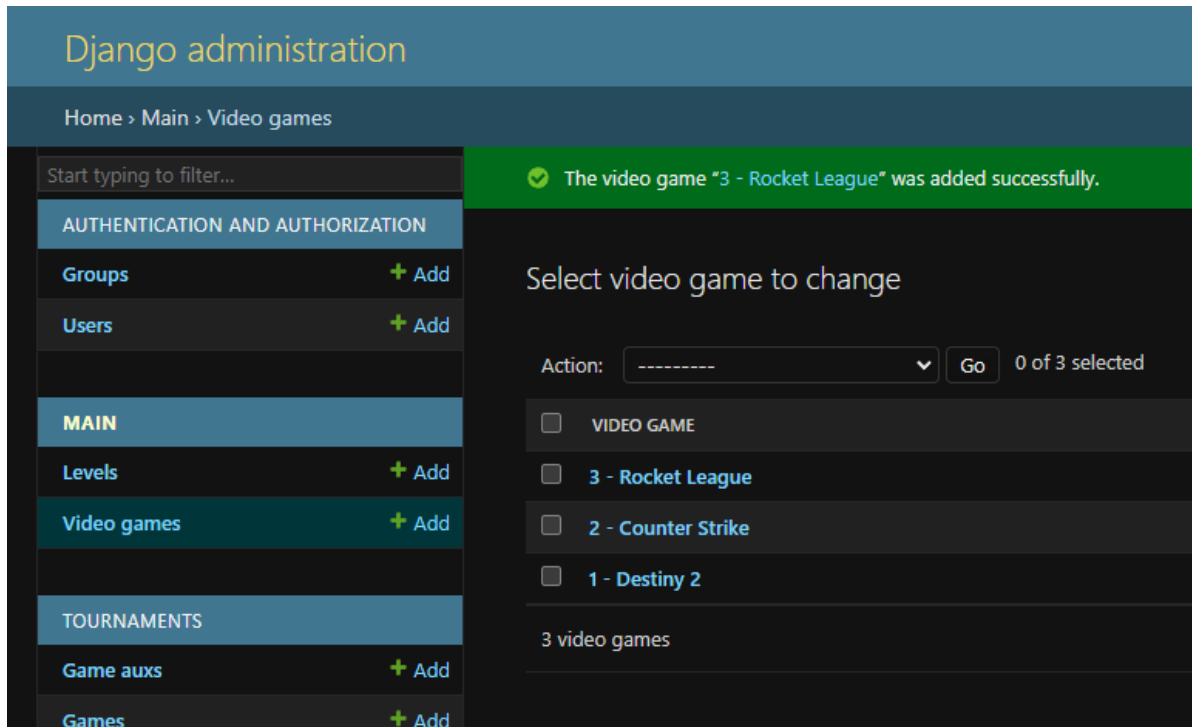
Games loses: 0

Games draws: 0

Winrate: 0

SAVE Save and add another Save and continue editing

La última tabla es Video Games que añadiremos los juegos que se van a usar para los torneos:



Así tendríamos todo el proyecto preparado para funcionar

## Comandos Django

Los comandos que vas a necesitar saber para poder usar el proyecto son los siguientes:

- `python manage.py runserver`: Este comando te permite arrancar el servidor y empezar a funcionar
- `python manage.py createsuperuser`: Este comando te permite crear un usuario administrador para acceder a la BBDD
- `python manage.py makemigrations`: Este comando en el caso de que modifiques la BBDD es para guardar los datos en una nueva migración
- `python manage.py migrate`: Este comando aplica todos los cambios almacenados en las migrations y los aplica a la BBDD



# CONCLUSIONES

## Conclusiones Generales

Ha sido un proyecto interesante tanto por su dificultad como el uso por primera vez de Python para un proyecto de estas características. Las mayores complejidades de este proyecto han sido tanto la composición de las tablas de la BBDD y la creación del sistema para poder registrar los partidos de los jugadores. Estas dificultades me han permitido aprender nuevas cosas y desarrollarme como programador de Python.

## Aprendizajes y experiencias adquiridas

Los puntos más importantes que he aprendido con este proyecto son los siguientes:

- **Uso de Django:** He aprendido a usar y programar en este framework de Django.
- **Uso de HTML twig:** Primera vez que creo un proyecto donde puedes meter condiciones que, si se cumplen o no, modifican la vista del usuario.
- **Búsqueda de Información:** Debido a mi desconocimiento con este framework, tras pasar veces con un estrés alto debido a mi desconocimiento aprendes a manejar la información que necesitas y filtrar cual es la información que te sirve más para el proyecto.



## EVOLUTIVOS DEL PROYECTO

Las mejoras que se pueden implementar a futuro en este proyecto son los siguientes:

- **Mejora de diseño gráfico:** Diseño gráfico pobre y poco llamativo.
- **Mejora en el sistema de reporte:** Mejorar el sistema de reporte de los resultados de los jugadores y como le llega al gestor del proyecto
- **Mejora de la creación de cuentas gestoras de proyectos:** Ahora se usa a las cuentas con privilegios como cuentas que tengan los permisos para crear los proyectos.
- **Implementación de actualizado automático:** Mejer un sistema que actualice la información de manera automática cuando un jugado envíe un reporte sin problemas.