Tablas de la Base de Datos: Wordle+.

Alumno: David Correa Rodríguez
Tutor: Juan Manuel Fernández Luna

Aclaraciones previas

El modelado de base de datos en Django (tecnología escogida para desarrollar el *backend* junto con PostgreSQL) es muy similar a un modelado convencional en SQL o similares. Sin embargo, Django ofrece algunas facilidades, como los tipos de datos Image o ManyToMany que son necesarios destacar, y que se utilizarán en el modelado de las tablas de la aplicación.

- **ManyToManyField**: es un campo que se utiliza para las relaciones Muchos-A-Muchos, en el que se incluye la tabla a la que hace referencia. En el modelado básico, susituiría a una clave primaria compuesta de dos claves, en donde cada una de ellas se puede repetir de forma independiente, pero de forma conjunta no es posible que existan instancias repetidas. Dicho de otra manera, el campo ManyToMany representa una **lista de claves primarias** de una tabla.

  Un ejemplo de esto puede ser una tabla “Pizza” que contiene varios *toppings* almacenados en la tabla “Toppings”. De esta manera, la tabla “Pizza” contendría una lista de *toppings*:

  ```
  class Pizza(models.Model):
      toppings = models.ManyToManyField(Toppings)
  ```

- **ImageField**: hereda de la clase FileField, que representa el almacenamiento de un fichero en un directorio en concreto. De forma interna, representa una cadena de caracteres con la ruta del fichero en cuestión. Particularizando en ImageField, añade además varios parámetros opcionales como la altura y anchura de la imagen.

Esta información ha sido extraída de la referencia de modelado de Django: https://docs.djangoproject.com/en/4.1/ref/models/fields/

Con el actual modelado de tablas de la base de datos no es necesario una tabla “Gestores de Eventos”, su información puede representarse en la tabla “Usuarios” teniendo en cuenta el atributo *is_manager.*

En cambio, al tener los jugadores más atributos que los gestores de eventos no tienen, sí es necesario, en este caso, crear una tabla aparte para los Jugadores, pero su información en común con los Gestores de eventos (como nombre, apellidos, etc) se almacenará en la tabla Usuarios para evitar redundancia e inconsistencias en los datos.

Modelado

Usuarios (Users):

Campo	Tipo	Descripción
username	String	Alias de usuario, CP.
email	mail	Correo. Único.
name	String	Nombre del usuario
surname	String	Apellido del usuario
register_date	Date	Fecha de registro del usuario
is_manager	Boolean	Indica si es jugador (False) o GE (True)
avatar	Image	Avatar del jugador

Jugadores (Players):

Campo	Tipo	Descripción
username	String	Nombre de usuario, único. CE de Usuarios. CP
wins	Integer	Total de Wordles completados
wins_pvp	Integer	Total de partidas 1vs1 ganadas
wins_tournament	Integer	Total de torneos ganados
xp	Integer	Experiencia del jugador
category	String	Rango del jugador

Lista Amigos (FriendList):

La lista de amigos representará una lista de claves primarias a la tabla de Jugadores

Campo	Tipo	Descripción
username	String	CP. CE de Jugadores (username).
friend_list	ManyToMany(Jugadores)	Lista de claves primarias de la tabla Jugadores (username).

Solicitudes Amigos (FriendRequests):

Para tener un sistema de gestión de amigos más realista, también existirá una tabla para almacenar las solicitudes de amistad. Las solicitudes de amistad podrán aceptarse/rechazarse/cancelarse, por lo que es el inidcador *is_active* será necesario.

Campo	Tipo	Descripción
sender	String	CE de Jugadores (username).
receiver	String	CE de Jugadores (username).
timestamp	Date	Fecha y hora de la solicitud
is_active	Boolean	Indica si está activa (True) o inactiva (False)

Partidas (Games):

Almacena las partidas jugadas, tanto 1vs1 como las partidas de los torneos. Al ser las partidas asíncronas, es decir, no es necesario que ambos jugadores estén jugando a la vez, es necesario almacenar qué jugador ha jugado la partida.

Campo	Tipo	Descripción
id_game	UUID	CP. Creada por defecto
player1	String	CE de Jugadores (username).
player2	String	CE de Jugadores (username).
player1_played	Boolean	True si el jugador 1 ha jugado. False si no ha jugado
player2_played	Boolean	True si el jugador 2 ha jugado. False si no ha jugado
winner	Boolean	True si ha ganado el jugador 1, False si ha ganado el jugador 2

Buzón (Inbox):

El buzón es una forma de notificar a los Jugadores de las interacciones con el resto de jugadores. De esta manera, a los jugadores se les notificará de una nueva petición de amistad o de que forman parte de un nuevo torneo.

Campo	Tipo	Descripción
id_notification	UUID	Clave primaria. Creada por defecto
nickname	String	CE de Usuario.
notification	String	Texto de la notificación
link	String	Enlace a la notificación

Torneos (Tournaments):

Campo	Tipo	Descripción
id_tournament	UUID	Clave primaria. Creada por defecto
name_tournament	String	Nombre del torneo
description	String	Descripción del torneo
max_players	Integer	Máximo de jugadores permitidos

Participaciones (Participations):

Es la tabla que contiene los jugadores inscritos en el torneo identificado por *id_tournament*.

Campo	Tipo	Descripción
id_tournament	UUID	CE de Torneo
players_list	ManyToMany	Lista de jugadores que participan en el torneo

Rondas (Rounds):

La tabla Rondas representa las rondas de un torneo. De forma interna, es una lista de partidas.

Campo	**Tipo**	**Descripción**
id_round	UUID	CP. Creada por defecto
id_tournament	UUID	CE de Torneos
round_number	Integer	Número de la ronda
games_list	ManyToMany	Lista de partidas de dicha ronda