

Diseño de la Base de Datos: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

Análisis del contexto

Para realizar un correcto diseño de la base de datos, previamente a caracterizar éste a la tecnología a utilizar, es necesario hacer un análisis de toda la información necesaria a almacenar y sus respectivas relaciones.

Para ello, y recapitulando todas las características principales del proyecto, existen varias entidades básicas de las que tenemos que almacenar información:

- Usuarios
- Partidas
- Torneos

De los usuarios derivan: los jugadores, los gestores de eventos, la lista de amigos, las peticiones de amistad y el buzón.

De los torneos derivan: las participaciones de los jugadores en los torneos y las rondas de cada torneo.

Con esta información, podemos identificar las identidades y las relaciones que se dan en la base de datos, añadiendo también los campos asociados:

- **Usuarios (entidad):** nombre de usuario, correo, nombre, apellidos, fecha de registro, si es manager o no, y el avatar):
 - o **Jugadores (entidad):** nombre de usuario, wordles completados, partidas 1vs1 ganadas, torneos ganados, experiencia y categoría)
 - o **Gestores de eventos (entidad):** nombre de usuario, correo, nombre, apellidos, fecha de registro
- **Lista de amigos (relación):** nombre de usuario de un jugador, nombre de usuario de otro jugador.

- **Solicitudes de amigos (entidad):** nombre de usuario del remitente, nombre de usuario del destinatario, la fecha de creación y si es válida o no.
- **Buzón (entidad):** nombre de usuario asociado al buzón, texto asociado y link asociado. Como no existe una clave primaria, esta entidad es una entidad débil de *Jugador*
- **Partidas (entidad):** identificador de la partida, nombre de usuario del jugador 1, nombre de usuario del jugador 2, si el jugador 1 ha jugado, si el jugador 2 ha jugado, y quién es el ganador.
Nota: las partidas serán asíncronas, es decir, no es necesario que ambos jugadores estén al mismo tiempo jugando la partida, por lo que es necesario los campos de si cada jugador ha jugado la partida.
- **Torneos (entidad):** identificador del torneo, nombre, descripción y número máximo de jugadores.
- **Participaciones (relación Jugadores-Torneos):** identificador del torneo asociado, nombre de usuario del jugador asociado.
- **Rondas (relación):** identificador de la ronda, identificador del torneo asociado, número de la ronda, partida asociada a dicha ronda

En el análisis podemos notar lo siguiente:

- Las entidades **Jugadores** y **Gestores de eventos** son entidades heredadas de **Usuarios**, por lo que tendrán una relación de herencia *es-un*.
- La entidad **Gestores de eventos** resulta innecesaria al tener un subconjunto (y solamente dicho subconjunto) de los campos de la entidad **Usuarios**.
- Existen muchas más relaciones sin mencionar, pero que son innecesarias y no corresponden a la base de datos, pero sí se tendrán en cuenta en el diagrama E/R²⁹.

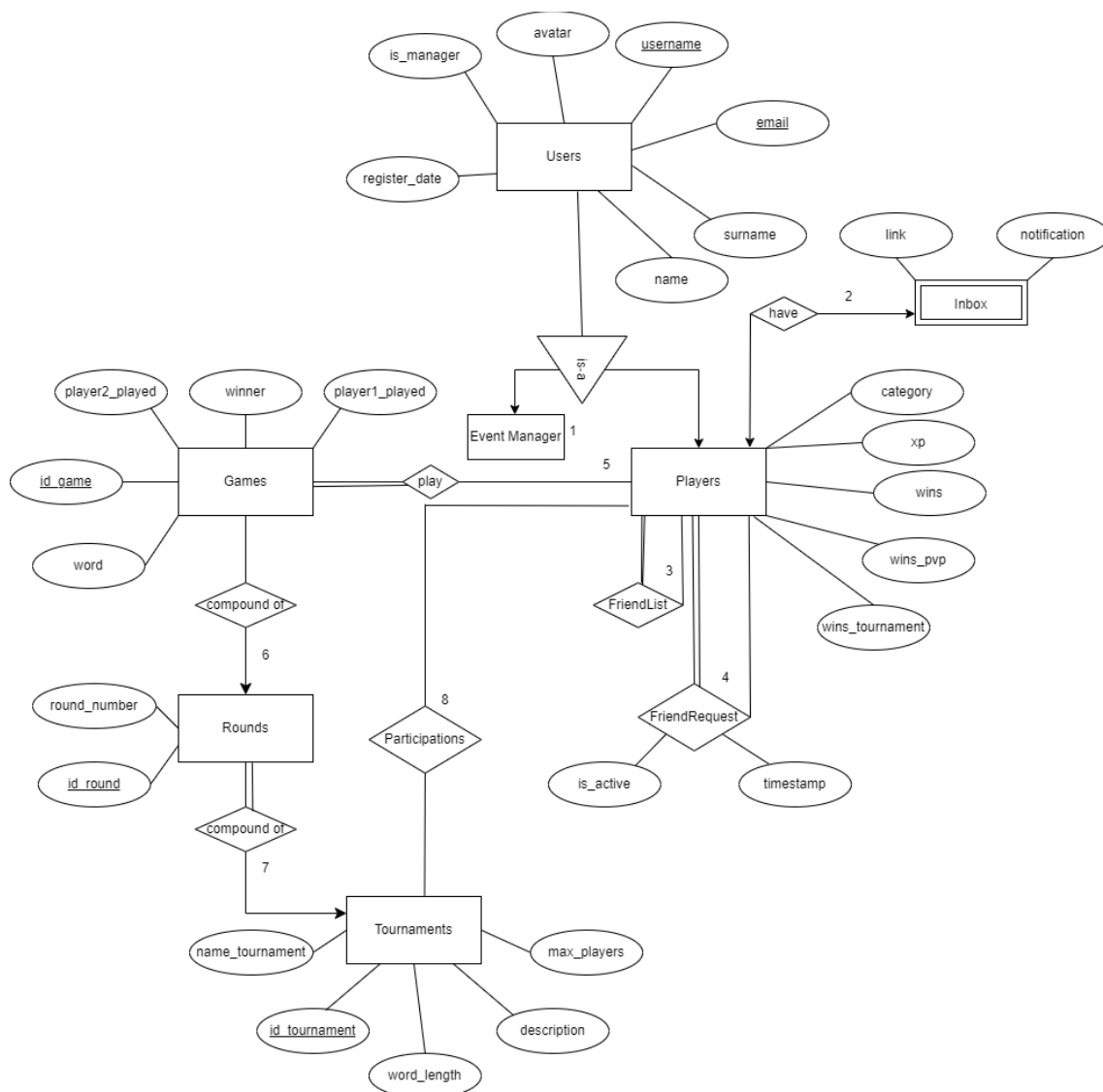
Estás relaciones pueden ser: *Jugador-Buzón* (tiene), *Jugador-Lista de amigos* (tiene), *Jugador-Solicitudes de amistad* (tiene), *Torneos-Rondas* (se componen de), *Jugador-Partidas* (juegan), *Rondas-Partidas* (se componen de).

- Existen varias entidades que dependen existencialmente de otras, es decir, la existencia de una entidad es obligatoria para la existencia de otra entidad.

Estas obligatoriedades se dan entre (la entidad de la izquierda es la que depende existencialmente de la otra): *Lista de amigos-Jugador*, *Solicitudes de amistad-Jugador*, *Buzón-Jugador*, *Partidas-Jugador*, *Ronda-Torneo*.

Diagrama Entidad-Relación

Teniendo en cuenta todos los aspectos destacados en el previo análisis, un posible diagrama Entidad-Relación sería el siguiente:



Algunas aclaraciones a tener en cuenta, numeradas en el diagrama, son:

1. Como se ha comentado anteriormente, la entidad *Event Manager* es una especialización de la entidad *Users* sin ningún atributo extra, por lo que añadiendo el campo *is_manager* a la entidad *Users* no es necesario crear una tabla nueva para *Event Manager*.
2. Un jugador solamente tiene un buzón, y dicho buzón solamente puede pertenecer a un usuario: cardinalidad 1:1. Además, la entidad buzón es débil al no tener una clave primaria propia. Se ha obviado la obligatoriedad entre estas entidades al ser *Inbox* una entidad débil, por lo que dicha condición ya se cumple. La entidad *Inbox* heredaría como clave externa y primaria la clave de la entidad *Player*.
3. Un jugador puede ser amigo de muchos jugadores, y también viceversa, por lo que presenta una cardinalidad N:M. Además, la tabla resultante de esta relación presenta una obligatoriedad con la entidad *Players* ya que su existencia depende de ésta. Esta relación heredaría las claves primarias de la entidad *Players*.
4. Relación similar a la anterior, pero añadiendo atributos extra. Un jugador puede tener varias peticiones de amistad, y una misma petición de amistad puede pertenecer a varios jugadores (N:M), pero en este caso solamente puede pertenecer a uno de los dos jugadores, pero no de forma simultánea. Esta condición deberá comprobarse antes de introducir valores a la tabla resultante. La tabla resultante de esta relación exige una obligatoriedad con la entidad *Players* con la misma justificación que el punto anterior.
5. Un jugador puede jugar muchas partidas, y esa partida puede ser jugada varias veces por los mismos jugadores (repetir una partida), por lo que la relación presenta una cardinalidad N:M, con obligatoriedad de las partidas hacia los jugadores, ya que sin jugadores las partidas no pueden ser jugadas.
6. Una ronda está compuesta de varias partidas, pero una partida de un torneo en concreto solamente puede pertenecer a una ronda, ya que en un torneo no se repiten enfrentamientos entre los mismos jugadores en diferentes momentos (en torneos eliminatorios), por lo que la relación presenta una cardinalidad (1:N). No se ha especificado obligatoriedad en este caso ya que se permiten que haya rondas sin partidas, ya que no es posible saber las partidas de las rondas siguientes sin los resultados de la ronda actual en ese momento.

7. Un torneo se compone de varias rondas, pero dicha ronda solamente puede pertenecer a un torneo en concreto, por lo que la relación presenta una cardinalidad (1:N). Además, existe obligatoriedad entre las rondas y los torneos, ya que una ronda no tiene sentido sin antes existir un torneo que la contenga.
8. Un jugador puede participar en muchos torneos, y en un mismo torneo pueden participar muchos jugadores, por lo que la relación presenta una cardinalidad N:M.

Paso a tablas

En el modelo entidad relación se observan muchas entidades y relaciones, pero no es necesario representar todas las relaciones y entidades expuestas en el paso a tablas, por lo que las más importantes y fundamentales son:

<i>Users</i>	<i>Players</i>	<i>FriendList</i>
username - PK	username - PK, FK (Users)	sender - FK (Players)
email - UNIQUE		receiver - FK (Players)
	wins	<i>PK (sender, receiver)</i>
name	wins_pvp	
surname	wins_tournament	
register_date	xp	
is_manager	category	
avatar		

<i>FriendRequest</i>	<i>Games</i>	<i>Inbox</i>
sender - FK (Players)	id_game - PK	nickname - FK (Players), PK
receiver - FK (Players)	player1 - FK (Players)	notification
<i>PK (sender, receiver)</i>	player2 - FK (Players)	link
timestamp	player1_played	
is_active	player2_played	
	winner	
	word	

<i>Tournaments</i>	<i>Participations</i>	<i>Rounds</i>
id_tournament - PK	player - FK (Players)	id_round - PK
	id_tournament - FK (Tournaments)	id_tournament - FK (Tournaments)
name_tournament	<i>PK (player, id_tournament)</i>	id_game - FK (Games)
description		<i>PK (id_round, id_tournament)</i>
max_players		<i>UNIQUE (id_round, id_game)</i>
word_length		
		round_number
		max_players
		word_length

Bibliografía y referencias: Wordle+.

-
- ¹ Wikipedia. *es.wikipedia.org*. 22 de febrero de 2023.
https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software.
- ² SA, ERIKA. *atlassian.com*. s.f.
<https://www.atlassian.com/es/agile/scrum/scrum-metrics> (último acceso: 2 de febrero de 2023).
- ³ Kabanize. *¿Qué es Kanban?* s.f. <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban#:~:text=Kanban%20es%20un%20m%C3%A9todo%20Lean,la%20eficiencia%20y%20mejorar%20continuamente>.
- ⁴ Unity. *¿Qué es CI/CD?* s.f. <https://unity.com/es/solutions/what-is-ci-cd#:~:text=La%20CI%2FCD%2C%20o%20integraci%C3%B3n,la%20entrega%20continua%20de%20c%C3%B3digo>.
- ⁵ Google. *developer.android.com*. s.f. <https://developer.android.com/studio> (último acceso: 25 de febrero de 2023).
- ⁶ Flutter. <https://flutter.dev/>. s.f. <https://flutter.dev/> (último acceso: 25 de febrero de 2023).
- ⁷ IBM. *www.ibm.com*. 9 de mayo de 2019. <https://www.ibm.com/es-es/cloud/learn/lamp-stack-explained>.
- ⁸ XenForo. *linux.org*. s.f. <https://www.linux.org/>.
- ⁹ Apache. *httpd.apache.org/*. s.f. <https://httpd.apache.org/> (último acceso: 25 de febrero de 2023).
- ¹⁰ Oracle. *mysql.com*. s.f. <https://www.mysql.com/> (último acceso: 25 de febrero de 2023).
- ¹¹ PHP. *php.net*. s.f. <https://www.php.net/> (último acceso: 25 de febrero de 2023).
- ¹² JavaScript. *javascript.com*. s.f. <https://www.javascript.com/> (último acceso: 25 de febrero de 2023).
- ¹³ MongoDB. *What is the MERN stack?* s.f. <https://www.mongodb.com/mern-stack#:~:text=MERN%20stands%20for%20MongoDB%2C%20Express,MongoDB%20%E2%80%94%20document%20database>.
- ¹⁴ Inc., MongoDB. *mongodb.com*. s.f. <https://www.mongodb.com/> (último acceso: 25 de febrero de 2023).
- ¹⁵ Platforms, Meta. *reactjs.org*. s.f. <https://es.reactjs.org/>.
- ¹⁶ Angular. *angular.io*. s.f. <https://angular.io/> (último acceso: 25 de febrero de 2023).

¹⁷ Foundation, OpenJS. *nodejs.org*. s.f. <https://nodejs.org/en/> (último acceso: 25 de febrero de 2023).

¹⁸ Dart. *dart.dev*. s.f. <https://dart.dev/> (último acceso: 25 de febrero de 2023).

¹⁹ Native, React. *https://reactnative.dev/*. s.f.

²⁰ Wikipedia. *Objective-C definition*. s.f.
<https://es.wikipedia.org/wiki/Objective-C>.

²¹ Inc, Apple. *swift*. s.f. <https://www.apple.com/es/swift/>.

²² Oracle. *java.com*. s.f. <https://www.java.com/es/> (último acceso: 25 de febrero de 2023).

²³ Ionic. *https://ionicframework.com/*. s.f.

²⁴ You, Evan. *vuejs.org*. s.f. <https://vuejs.org/> (último acceso: 25 de febrero de 2023).

²⁵ PostgreSQL. *postgresql.org*. s.f. <https://www.postgresql.org/> (último acceso: 25 de febrero de 2023).

²⁶ Python. *python.org*. s.f. <https://www.python.org/> (último acceso: 25 de febrero de 2023).

²⁷ Framework, Django Rest. *django-rest-framework.org*. s.f.
<https://www.django-rest-framework.org/> (último acceso: 25 de febrero de 2023).

²⁸ Docker. *docker.com*. s.f. <https://www.docker.com/> (último acceso: 25 de febrero de 2023).

²⁹ Wikipedia. *Modelo entidad-relacion*. s.f.
https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n (último acceso: 25 de febrero de 2023).