



Tecnológico de Monterrey

Ejercicio de Modelación de Base de Datos Pokemon TCG

Amilka Daniela Lopez Aguilar - A01029277

Campus Santa Fe

Construcción de software y toma de decisiones

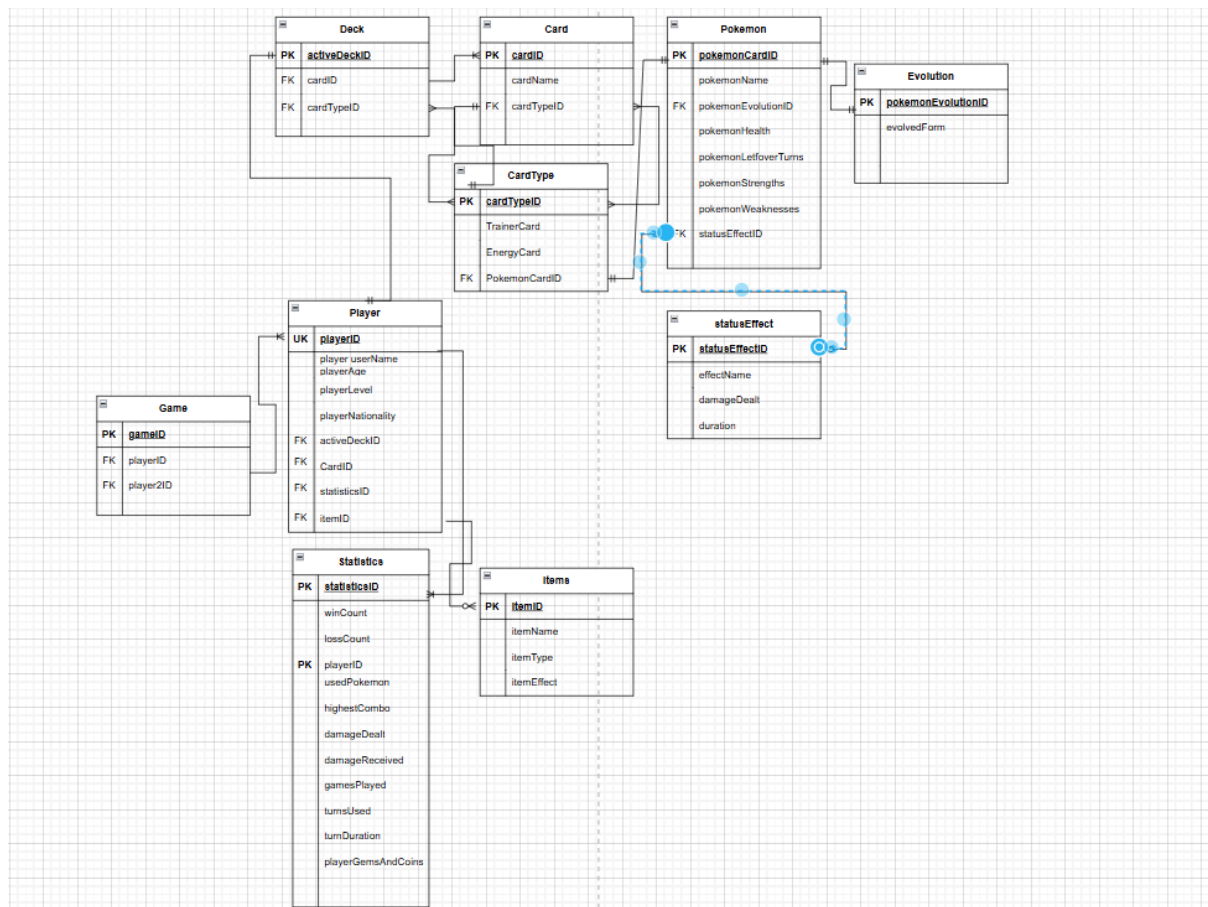
Octavio Navarro Hinojosa

Esteban Castillo Juarez

Gilberto Echeverría Furió

Martes 4 de marzo de 2025

Justificación de las clases que aparecen en el diagrama ER



El sistema de juego está organizado mediante varias tablas interrelacionadas que gestionan distintos aspectos del juego.

La tabla de juego es el “padre” de todas las demás entidades que podrían aparecer en el juego, modelando que una partida está compuesta por dos jugadores. Se relaciona con una cardinalidad de 1 a muchos, pues para un juego pueden existir dos perfiles de jugador y esos dos perfiles de jugador solamente participan en un juego. La tabla de jugadores almacena la información personal y de perfil de los participantes, como su nivel, edad y nombre de usuario, además de vincularse con las barajas que utilizan en el juego (y por ende, sus cartas), así como sus estadísticas de juego y artefactos recolectados.

Jugadores tiene una cardinalidad 1 a muchos con estadísticas porque para un jugador existen muchas estadísticas, como son sus resultados de partida, tiempo jugado, artefactos, monedas y gemas. Para 1 jugador pueden existir varias estadísticas según su número de

partidas. 1 jugador puede tener 1 o muchos artefactos opcionalmente dependiendo de cómo se desarrolle la partida.

Las barajas contienen las cartas del jugador, que a su vez están relacionadas con los Pokémon que estas representan. Cada Pokémon tiene atributos como su vida, fortalezas, debilidades, turnos restantes, y puede verse afectado por efectos de estado que modifican su rendimiento durante la partida. La tabla de evolución permite gestionar las transformaciones de los Pokémon, mostrando sus posibles formas evolucionadas y las relaciones entre ellas. Se entiende que una baraja puede contar con muchas cartas y que varias cartas pueden pertenecer a una misma baraja. Sin embargo, la cardinalidad entre carta, pokemon y evolución es de 1 a 1 porque una carta le pertenece solamente a un pokemon y un pokemon solamente tiene un tipo de evolución (si nos referimos al nombre específico que adquieren una vez que evolucionan). La tabla intermedia de Cardtype permite atribuir no solo cartas pokemon, sino de entrenador y energía a la baraja y a la carta. Una baraja cuenta con muchos tipos de cartas, pero una carta solamente tiene atribuida un tipo de la misma.

Por último, la entidad de statusEffect revela pokemones afectados por algún estado o combinación de cartas por parte del jugador opuesto, admitiendo los atributos de duración, nombre del efecto y daño. Se relaciona al pokemon para saber sus vitales en una cardinalidad de muchos opcionales, pues puede o no existir un efecto especial y acumularse.

Las relaciones entre las tablas permiten una gran flexibilidad en el juego, donde los jugadores pueden personalizar sus barajas, gestionar los efectos de estado en sus Pokémon y mejorar sus estrategias mediante el uso de artefactos. Los datos recopilados durante las partidas en la tabla de juego permiten analizar y comparar el desempeño de los jugadores, ofreciendo una visión detallada de cómo su elección de cartas, habilidades y tácticas influyen en el resultado final. Este sistema facilita no solo la gestión de las partidas, sino también la

evolución de los jugadores dentro del juego, creando un entorno competitivo y de alto interés para el jugador.

Las restricciones de integridad son fundamentales para garantizar que los datos en la base de datos se mantengan consistentes, precisos y coherentes. En primer lugar, se debe asegurar la integridad de entidad, lo que significa que cada tabla debe tener una clave primaria única y no nula, garantizando que cada fila represente una entidad única. En cuanto a la integridad referencial, es crucial que las relaciones entre tablas se validen mediante claves foráneas, asegurando que no existan referencias en el aire y que los datos en una tabla estén directamente relacionados con los de otras tablas.

Además, es importante respetar las restricciones de cardinalidad, lo que asegura que las relaciones entre entidades se mantengan dentro de los límites previstos. Por ejemplo, un jugador puede tener varias estadísticas, pero cada estadística solo se puede asociar a un jugador. De igual manera, una baraja puede contener varias cartas, pero cada carta debe pertenecer a una sola baraja. La relación entre cartas y Pokémon también debe ser de uno a uno, dado que cada carta representa un solo Pokémon y cada Pokémon tiene un solo tipo de evolución.

Las restricciones de unicidad también deben aplicarse, garantizando que los valores en ciertas columnas, como los nombres de usuario de los jugadores, sean únicos, evitando duplicados dentro de la base de datos. Además, las restricciones de integridad de dominio deben asegurarse de que los datos introducidos sean válidos según el tipo de dato esperado, como asegurarse de que la edad de un jugador sea un número entero positivo o que el daño de un estado sea un valor no negativo. Todos los nombres y descripciones serán strings, mientras que el resto de los datos serán integers positivos.

Finalmente, las restricciones de integridad respecto a la jugabilidad. Por ejemplo, un Pokémon solo debe poder evolucionar a una forma específica según las reglas definidas en la

base de datos. Del mismo modo, las cartas de Pokémon deben estar vinculadas exclusivamente a Pokémon existentes, evitando la creación de cartas sin referencia.

En resumen, la implementación de estas restricciones de integridad asegura que los datos sean consistentes, que las relaciones entre las entidades sean válidas y que el comportamiento del sistema se ajuste a las reglas definidas, creando un entorno de juego robusto y confiable.