



# Tecnológico de Monterrey

**Ejercicio de creación de una base de datos relacional en MySQL para  
el juego de cartas Pokemon TCG**

Paolo Antonio Pires Cano

A01749355

28 de Abril del 2024

Jugador: Esta entidad es completamente necesaria ya que es la que se utiliza para poder guardar la información básica sobre los jugadores como su ID, su nombre de usuario, su correo, cuándo se unieron al juego y su récord en las partidas. Además tiene una relación de uno a muchos con la entidad Baraja, ya que un jugador puede tener múltiples barajas, pero una baraja solo pertenece a un jugador.

Carta\_Pokemon, Carta\_Energía, Carta\_Entrenador: Al principio consideré crear una sola tabla de cartas, pero luego me di cuenta de todas las diferencias que tenían cada una de ellas. Las cartas de los pokémones tienen fases de evolución, el PS, tipo de ataque, mientras que las de energía y entrenador tienen más que nada los diferentes tipos que pueden llegar a ser como energía tipo rayo o entrenador tipo poción. Es por eso que decidí crear tres diferentes entidades para las cartas.

Baraja: Esta entidad fue creada para poder saber qué jugador tiene qué baraja, cuándo fue la fecha de creación y el tamaño (cantidad de cartas en la baraja).

Relación\_Cartas\_Baraja: Las tres entidades que creé (Relación\_CartaEntrenador\_Baraja, Relación\_CartaEnergía\_Baraja, Relación\_CartaPokemon\_Baraja), fueron para poder normalizar la relación que tenían los tres tipos de cartas con la entidad de baraja. Estas relaciones tenían una cardinalidad de muchos a muchos, es por eso que se crearon las tablas intermedias para poder trabajar con uno a muchos. Esto es porque una baraja puede tener múltiples cartas, y una carta puede estar en múltiples barajas.

Partida: Fue una entidad necesaria para guardar registro de las partidas, incluye su ID, la moneda que se utilizó en la partida y la fecha en la que la partida se llevó a cabo.

Moneda: Simplemente contiene el ID de una moneda específica que se puede utilizar en una partida. Tiene una relación con cardinalidad de uno a muchos con la entidad de partida

ya que una partida tiene una sola moneda, pero una moneda puede participar en múltiples partidas.

Turno: Esta entidad funciona para saber lo que pasa dentro del turno de un jugador específico dentro de una partida específica, el estado (si le toca al jugador o no), el daño dado en el turno, el daño recibido, el daño acumulado, al igual que cuánto duró el turno en segundos y la cantidad de cartas jugadas en el turno (sean Pokemon, Entrenador o Energía). Tiene relación con cardinalidad de uno a muchos con jugador, porque un turno solo tiene un jugador, pero un jugador puede tener muchos turnos E igual uno a muchos con una partida porque un turno pertenece a solo una partida, pero una partida puede tener varios turnos.

#### Justificación de forma Normal:

Está en primera forma normal porque, se tienen atributos atómicos (el campo Email por ejemplo, no puede contener muchos Emails), ninguna de las llaves primarias pueden tener atributos nulos, no hay ninguna variación en el número de columnas y los atributos que se guardan son genuinamente “indivisibles”. También está en segunda forma normal porque aparte de esto, las llaves primarias están formadas por una única columna con valor indivisible, en todas las tablas. E igualmente está en tercera forma normal porque sumado a todo esto, en todas las tablas se puede observar que las columnas que no forman parte de la llave primaria dependen únicamente de esa llave y no de otra columna.