



**Tecnológico de Monterrey**  
**Escuela de Ingeniería y Ciencias**

### **3. Ejercicio de Modelación de Base de Datos Pokemon TCG**

- - -

Construcción de software y toma de decisiones (Gpo 501)

**Sergio Jiawei Xuan**

**A01784695**

Código de clase:

TC2005B.501

Profesor@:

Esteban Castillo Juarez

Fecha de entrega:

3 de mayo de 2025

Un jugador en Pokémon TCG Live mantiene una colección de cartas que agrupa todas las cartas que ha desbloqueado o adquirido. A partir de esa colección, puede construir múltiples decks, cada uno compuesto por exactamente 60 cartas, cumpliendo con las restricciones del formato vigente (p. ej. Estándar, Expandido, Histórico). Un deck registra además metadatos como su nombre, la fecha de creación, el formato al que pertenece y estadísticas propias (victorias/derrotas con ese mazo).

Cada carta dentro de la colección y de los decks cuenta con atributos clave:

- ID único (para rastreo y transacciones)
- Nombre (p. ej. “Pikachu”)
- Tipo (Pokémon, Entrenador, Energía)
- Subtipo (Básico, Fase 1, Fase 2, Objeto, Partidario...)
- Rareza (común, poco común, rara, ultra rara...)
- Edición/Formato (para saber si es legal en Standard, Expandido, etc.)

Las partidas (matches) se conforman siempre de dos jugadores y almacenan información como:

- Fecha y hora de inicio
- Formato de juego
- Decks utilizados por cada jugador
- Estado actual de la partida (en progreso, finalizada, interrumpida)
- Resultado: puede registrarse un ganador, un empate o quedar sin definición en caso de desconexión

Cuando la partida concluye, opcionalmente se guarda qué jugador resultó vencedor y bajo qué condiciones (por agotamiento de la baraja del oponente, por daño acumulado, etc.).

Por último, el sistema mantiene un historial de partidas para cada jugador, lo que permite generar estadísticas agregadas (total de partidas, porcentaje de victorias, rachas de victorias/derrotas) y activar reglas de emparejamiento o ligas clasificatorias.

## Entidades y atributos

Entidad	Atributos (dominio)
Player	player_id (PK, UUID), username (string), email (string), password (string), trainer_coins (int), trainer_credits(int), level(int), wins(int), created_at (date)
Deck	deck_id (PK), FK→Player(player_id), name (string), format (enum: A-H), created_at (date)
DeckCard	FK→Deck(deck_id), FK→Card(card_id), quantity (int)
Card	card_id (PK, string), name (string), rarity (enum), set_code (string), card_type (enum: Pokémon, Entrenador, Energía)
Collection	FK→Player(player_id), FK→Card(card_id), quantity (int)
PokemonCard	subtipo de Card FK→Card(card_id); health_points (int), stage (enum: Básico, Fase 1, Fase 2, V, VMAX...), pokemon_type (enum), ability (string), attack1_name (string), attack1_cost (text), attack1_dmg (int), attack2_name (string), attack2_cost (text), attack2_dmg (int), retreat_cost (string), weakness (string), resistances (string)
TrainerCard	subtipo de Card FK→Card(card_id); trainer_type (enum: Objeto, Partidario, Estadio...), text (string)
EnergyCard	subtipo de Card FK→Card(card_id); energy_type (enum: Fuego, Agua, Planta...), special_effect (string)
Match	match_id (PK), started_at (datetime), ended_at (datetime), FK→Player(player1_id), FK→Player(player2_id), FK→Player(winner_id, nullable)
GameState	match_id (PK, FK→Match(match_id)), player_id (PK, FK→Player(player_id)), turn_number (integer), active_card_id (string <<FK Card(card_id)>>), deck_count (int), hand_count (int), bench_count (int), prize_count (int), discard_count (int)

EnergyAttachment	FK→GameState(match_id,player_id), pokemon_slot (enum: Active, Bench1...5), FK→Card(energy_card_id)
------------------	--

## Relaciones

1. Player —\* Deck
  - a. Un jugador puede crear muchos mazos.
2. Deck — Card (vía DeckCard)
  - a. Un mazo contiene muchas cartas, y cada carta puede estar en muchos mazos.
3. Player — Card (vía Collection)
  - a. Modela la colección completa de un jugador.
4. Card — PokemonCard/TrainerCard/EnergyCard
  - a. Cada carta base (Card) puede especializarse en un solo subtipo.
5. Match —\* Player
  - a. Dos jugadores por partida; uno opcionalmente gana.
6. Match 1—\* GameState
  - a. Cada partida almacena el estado para cada jugador.
7. GameState — EnergyAttachment
  - a. Un estado de juego puede tener varias energías unidas a distintas ranuras de Pokémon.