



# **TCG POKEMON**

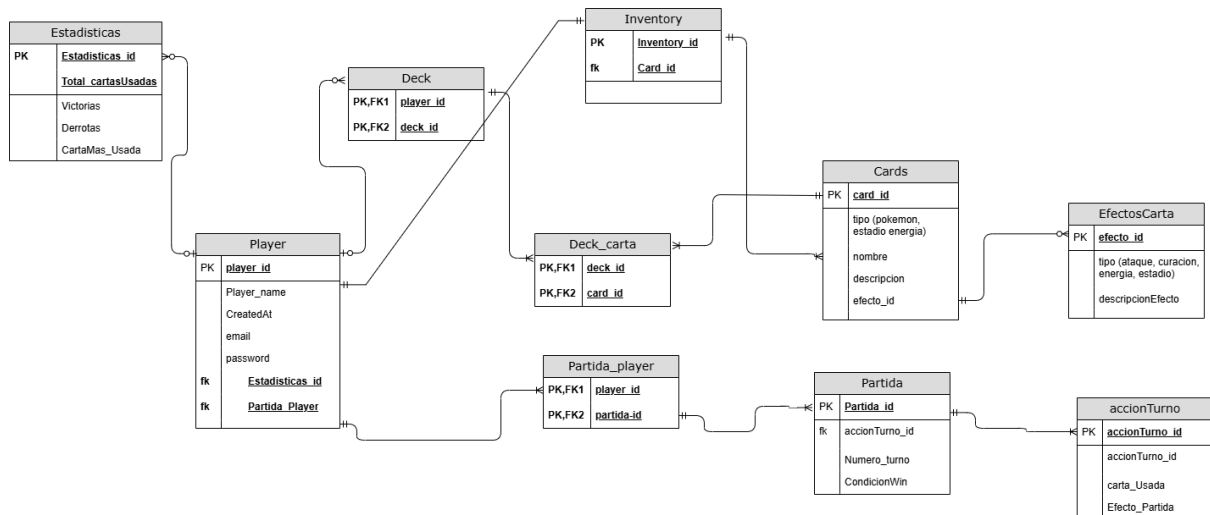
**Profesor:** Esteban Castillo Juárez

**Materia:** Construcción de Software para la toma de decisiones

**Luis Leonardo Rodriguez Galvez**

**a01029331**

Fecha: 3 de mayo del 2025



## Justificación del Modelo ER para Pokémon TCG

Este documento describe la motivación de cada tabla y atributo, sus dominios, restricciones de integridad, las cardinalidades entre entidades y la forma en que estos elementos reflejan la mecánica del juego Pokémon TCG.

### 1. Tabla: **Player**

**Descripción:** Representa a cada jugador registrado en la plataforma.

- **Atributos básicos:**

- **player\_id** (PK): Identificador único del jugador. Dominio: entero positivo.
- **player\_name**: Nombre de usuario. Dominio: cadena alfanumérica (mínimo 3, máximo 30 caracteres).
- **CreatedAt**: Fecha de creación de la cuenta. Dominio: fecha.
- **email**: Correo electrónico. Dominio: formato email.
- **password**: Contraseña cifrada. Dominio: hash seguro.
- **estadísticas\_id** (FK): Referencia a las estadísticas globales del jugador.

**Restricciones de integridad:**

- `player_name` y `email` deben ser únicos.
- `email` debe cumplir patrón de correo.
- `password` no nulo.
- `estadisticas_id` debe existir en `Estadisticas` (integridad referencial).

#### Cardinalidades:

- 1:N con `Partida_player`: un jugador puede participar en muchas partidas.
- 1:1 con `Estadisticas`: cada jugador tiene un conjunto agregado de estadísticas.
- 1:N con `Deck`: un jugador puede crear muchos mazos.

**Funcionalidad en el juego:** El jugador inicia sesión, elige mazos, participa en partidas y acumula estadísticas (victorias, derrotas, cartas usadas).

---

## 2. Tabla: `Estadisticas`

**Descripción:** Consolida el desempeño global de un jugador.

- **Atributos básicos:**
  - `estadisticas_id` (PK): Identificador único. Dominio: entero.
  - `Total_cartasUsadas`: Número total de cartas jugadas. Dominio: entero no negativo.
  - `Victorias`: Conteo de partidas ganadas. Dominio: entero no negativo.
  - `Derrotas`: Conteo de partidas perdidas. Dominio: entero no negativo.
  - `CartaMas_Usada`: Identificador de la carta más utilizada. FK a `Cards.card_id`.

#### Restricciones de integridad:

- Los contadores (`Total_cartasUsadas`, `Victorias`, `Derrotas`)  $\geq 0$ .
- `CartaMas_Usada`, si no es nulo, debe existir en `Cards`.

#### Cardinalidades:

- **1:1** con **Player**: vincula un conjunto de estadísticas a un jugador.

**Funcionalidad en el juego:** Permite mostrar reporte de desempeño y recomendaciones basadas en uso de cartas.

---

### 3. Tabla: **Deck**

**Descripción:** Define los mazos que crea cada jugador.

- **Atributos básicos:**
  - (**player\_id**, **deck\_id**) (PK compuesto): Identifica un mazo único por jugador.
  - **nombre\_deck**: Nombre descriptivo del mazo. Dominio: cadena.

**Restricciones de integridad:**

- Un mismo **player\_id** no puede tener dos mazos con el mismo **deck\_id**.
- **nombre\_deck** no nulo.

**Cardinalidades:**

- **N:M** con **Cards** a través de **Deck\_carta** (un mazo contiene muchas cartas y una carta puede estar en muchos mazos).
- **1:N** con **Player**: un jugador puede tener múltiples mazos.

**Funcionalidad en el juego:** El jugador arma estrategias eligiendo cartas y creando diferentes combinaciones de mazo.

---

### 4. Tabla: **Deck\_carta**

**Descripción:** Tabla intermedia que relaciona mazos y cartas.

- **Atributos básicos:**

- (`deck_id`, `card_id`) (PK compuesto): Par mazo–carta.
- `cantidad`: Número de copias de la carta en el mazo. Dominio: entero [1..4].

**Restricciones de integridad:**

- `deck_id` y `card_id` deben existir en `Deck` y `Cards`, respectivamente.
- `cantidad` entre 1 y 4 (regla oficial de TCG).

**Cardinalidades:**

- N:1 hacia `Deck` y N:1 hacia `Cards`.

**Funcionalidad en el juego:** Controla cuántas copias de una carta incluye un mazo.

---

## 5. Tabla: `Cards`

**Descripción:** Catálogo de todas las cartas disponibles.

- **Atributos básicos:**
  - `card_id` (PK): Identificador único de carta. Dominio: entero.
  - `tipo`: Tipo de carta (Pokémon, Estadio, Energía). Dominio: lista enumerada.
  - `nombre`: Nombre de la carta. Dominio: cadena.
  - `descripcion`: Descripción de los efectos de la carta. Dominio: texto.
  - `efecto_id` (FK): Referencia a la tabla `EfectosCarta`.

**Restricciones de integridad:**

- `tipo` debe ser uno de los valores permitidos.
- `nombre` único.
- `efecto_id`, si no nulo, debe existir en `EfectosCarta`.

**Cardinalidades:**

- 1:N con `Deck_carta`: una carta puede pertenecer a muchos mazos.

- 1:N con **Partida** a través de eventos de uso.

**Funcionalidad en el juego:** Define las propiedades estáticas de cada carta y enlaza su lógica de efectos.

---

## 6. Tabla: **EfectosCarta**

**Descripción:** Detalla cada tipo de habilidad o efecto de cartas.

- **Atributos básicos:**
  - **efecto\_id** (PK): Identificador único. Dominio: entero.
  - **tipo** (ataque, curación, energía, estadio). Dominio: enumerado.
  - **descripcionEfecto**: Texto explicativo del efecto.

**Restricciones de integridad:**

- **tipo** restringido a valores del juego.
- **descripcionEfecto** no nulo.

**Cardinalidades:**

- 1:N con **Cards**: un efecto puede aplicarse a múltiples cartas.

**Funcionalidad en el juego:** Centraliza la descripción y clasificación de habilidades y efectos.

---

## 7. Tabla: **Partida**

**Descripción:** Registro de cada partida jugada.

- **Atributos básicos:**
  - **partida\_id** (PK): Identificador de partida. Dominio: entero.

- `accionTurno_id` (FK): Última acción de turno registrada.
- `numero_turno`: Contador de turnos actuales. Dominio: entero.
- `condicionWin`: Estado de victoria/derrota. Dominio: booleano o enumerado.

**Restricciones de integridad:**

- `condicionWin` debe ser `en curso`, `gana_jugadorA` o `gana_jugadorB`.
- `accionTurno_id` debe existir en `accionTurno`.

**Cardinalidades:**

- 1:N con `accionTurno`: una partida tiene muchos eventos de turno.
- 1:N con `Partida_player`: una partida involucra dos o más jugadores.

**Funcionalidad en el juego:** Almacena la secuencia de juego y la condición final.

## 8. Tabla: `Partida_player`

**Descripción:** Asocia jugadores con partidas.

- **Atributos básicos:**
  - (`player_id`, `partida_id`) (PK compuesto): Par jugador–partida.

**Restricciones de integridad:**

- Ambos IDs deben existir en sus tablas padre.
- No permite duplicados (un jugador no se registra dos veces en la misma partida).

**Cardinalidades:**

- N:M entre `Player` y `Partida`.

**Funcionalidad en el juego:** Permite saber quién participa en cada partida.

## 9. Tabla: **accionTurno**

**Descripción:** Eventos o acciones tomadas en cada turno de la partida.

- **Atributos básicos:**
  - **accionTurno\_id** (PK): Identificador de la acción.
  - **carta\_Usada** (FK): Carta jugada en este turno.
  - **efecto\_partida**: Descripción del efecto aplicado durante el turno.

**Restricciones de integridad:**

- **carta\_Usada** debe existir en **Cards**.
- **efecto\_partida** no nulo.

**Cardinalidades:**

- **N:1** con **Partida**: cada acción pertenece a una partida.
- **N:1** con **Cards**: una acción involucra una carta.

**Funcionalidad en el juego:** Registra secuencialmente cada jugada, permitiendo reproducir la partida.

---

## 10. Tabla: **Inventory**

**Descripción:** Inventario de cartas compradas o coleccionadas por el jugador.

- **Atributos básicos:**
  - **inventory\_id** (PK): Identificador único.
  - **player\_id** (FK): Dueño del inventario.
  - **card\_id** (FK): Carta en el inventario.
  - **cantidad**: Número de unidades en colección. Dominio: entero  $\geq 0$ .

**Restricciones de integridad:**



- `player_id` y `card_id` deben existir.
- `cantidad`  $\geq 0$ .

**Cardinalidades:**

- N:M entre `Player` y `Cards` (un jugador posee muchas cartas y una carta es poseída por muchos jugadores).

**Funcionalidad en el juego:** Gestiona la colección de cartas disponibles para construir mazos.

---