

Bases de Datos

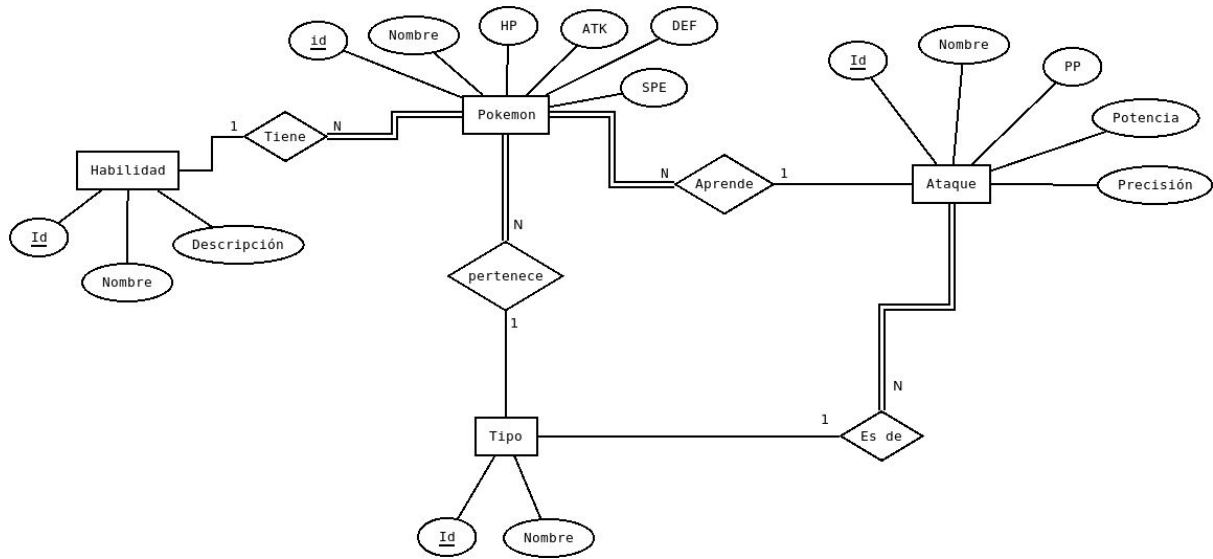
Avanzadas

Trabajo tutelado: Aplicación en SQL embebido en C

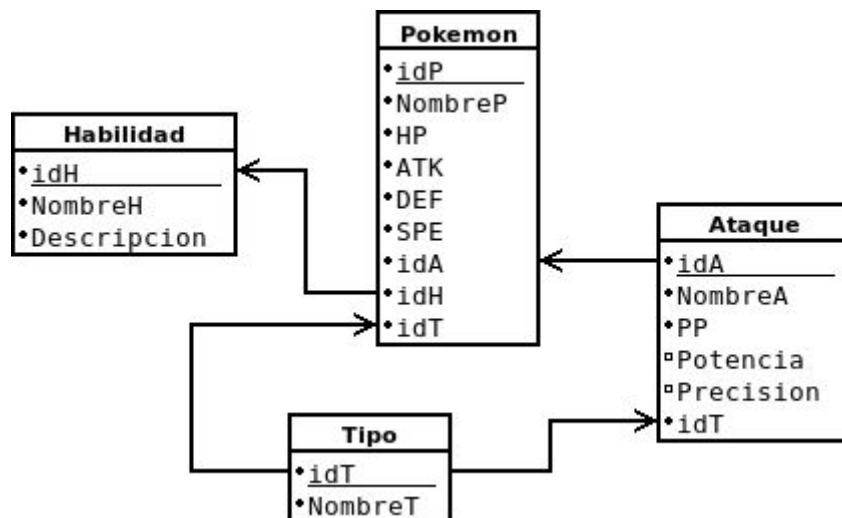
Curso 2018/2019

Nombre	DNI	email
Francisco Javier Manuel García	44918789L	francisco.j.manuelg@udc.es
David García Gondell	48119280Z	david.ggondell@udc.es

1. Modelización conceptual



2. Lógica de la base de datos



3. Diccionario de datos

=====

Tabla Pokemon: tabla para definir a cada Pokémon

=====

- **Atributos:**

- int **idP** : identificador del Pokémon.
- String **nombreP** : nombre del Pokémon.
- int **HP** : puntos de vida del Pokémon.
- int **ATK** : puntos de ataque del Pokémon.
- int **DEF** : puntos de defensa del Pokémon.
- int **SPE** : puntos de velocidad del Pokémon.
- int **idT** : identificador del tipo del Pokémon.
- int **idH** : identificador de la habilidad del Pokémon.
- int **idA** : identificador del ataque aprendido por el Pokémon.

- **Restricciones:**

- **Clave primaria** : idP
- **not null** : ningún atributo de la tabla Pokemon puede ser nulo, porque cada atributo es totalmente necesario para definir a cada Pokémon
- **Clave foránea** : idT(Tabla Tipo), idH(Tabla Habilidad), idA(Tabla Ataque)

=====

Tabla Ataque: tabla para definir el ataque que puede usar cada Pokémon

=====

- **Atributos:**

- int **idA** : identificador del ataque. Clave primaria de la tabla Ataque.
- String **nombreA** : nombre del ataque.
- int **PP** : número de veces que se puede usar el ataque en combate.
- int **Potencia** : potencia de ataque como medida numérica. Una potencia nula quiere decir que el ataque no hace daño.
- int **Precision** : probabilidad del ataque de acertar. Una precisión nula quiere decir que el ataque del Pokémon acertará siempre, independientemente de las condiciones del combate
- int **idT** : identificador del tipo del ataque.

- **Restricciones:**

- **Clave primaria** : idA
- **not null** : idA, nombreA, PP, idT
- **Clave foránea** : idT(Tabla Tipo)

=====

Tabla Tipo

=====

- **Atributos:**

- int **idT** : identificador del tipo. Clave primaria de la tabla Tipo.
- String **nombreT** : nombre del tipo.

- **Restricciones:**

- **Clave primaria** : idA
- **not null** : idT, nombreT

=====

Tabla Habilidad: tabla para definir la habilidad de cada Pokémon

=====

- **Atributos:**

- int **idH** : identificador de la habilidad. Clave primaria de la tabla Habilidad.
- String **nombreH** : nombre de la habilidad.
- String **descripción**

- **Restricciones:**

- **Clave primaria** : idA
- **not null** : idH, nombreH, descripción

=====

Justificación del diseño

=====

- **Control de errores:**

- **Insertar Pokémon:** Para insertar un Pokémon comprobamos que todos los valores introducidos cumplen las restricciones, esto quiere decir que los atributos que tengan la restricción NOT NULL y el usuario intente insertar un nulo para ese atributo se le mostrará un mensaje de error referente a que no se puede insertar nulos en ese atributo. En cambio, si todos los valores son correctos, pero la clave primaria (idP) introducida por el usuario es igual a una ya existente en la tabla Pokemon se le mostrará al usuario un mensaje de error de que ya existe un Pokémon con ese ID. Por último, si todo es correcto se introducirá el Pokémon nuevo a la tabla Pokemon.

- **Insertar ataque:** Para insertar un ataque comprobamos que todos los valores introducidos cumplen las restricciones, esto quiere decir que los atributos que tengan la restricción NOT NULL y el usuario intente insertar un nulo para ese atributo se le mostrará un mensaje de error referente a que no se puede insertar nulos en ese atributo. En cambio, si todos los valores son correctos, pero la clave primaria (idA) introducida por el usuario es igual a una ya existente en la tabla Ataque se le mostrará al usuario un mensaje de error de que ya existe un ataque con ese ID. Por último, si todo es correcto se introducirá el ataque nuevo a la tabla Ataque.
- **Eliminar Pokémon:** Para eliminar un Pokémon solicitamos al usuario que introduzca el ID del Pokémon que quiere eliminar. En el caso de que el ID no corresponda con ningún Pokémon se le mostrará a usuario un mensaje de error de que no existe ningún Pokémon con ese ID en la tabla Pokemon. Si el ID corresponde a un Pokémon se eliminará esa fila de la tabla Pokemon.
- **Eliminar ataque:** Para eliminar un ataque solicitamos al usuario que introduzca el ID del ataque que quiere eliminar. En el caso de que el ID no corresponda con ningún ataque se le mostrará a usuario un mensaje de error de que no existe ningún ataque con ese ID en la tabla Ataque. Si el ID corresponde a un ataque, pero ese ataque ya ha sido aprendido por un Pokémon se le mostrará al usuario un mensaje de error. Por último, si el ID corresponde a un ataque y no lo ha aprendido ningún Pokémon se eliminará de la tabla Ataque.
- **Buscar Pokémon:** Para buscar un Pokémon el usuario tendrá que introducir el ID del Pokémon. Si el Pokémon no existe un Pokémon con ese ID se le mostrará al usuario un mensaje de error de Pokémon inexistente. Si ese ID corresponde a un Pokémon se le mostrará el detalle de ese Pokémon.

- **Buscar Pokémon por tipo:** El usuario introducirá el ID del tipo (idT) para encontrar a los Pokémon correspondientes a ese tipo. Si para ese tipo no hay ningún Pokémon asignado se mostrará que Pokémon encontrados que corresponden a ese tipo es 0. En cambio, si ese tipo tiene Pokémon asignados se mostrará todos los Pokémon de ese tipo.
 - **Actualizar PP:** El usuario introducirá el ID del ataque (idA) al que quiere actualizar los PP y el nuevo valor de los PP. Si el ataque no existe con ese ID se mostrará un mensaje de error. En cambio, si existe se actualizarán los PP del ataque con el valor introducido por el usuario.
 - **Aumentar porcentaje ATK:** El usuario introducirá el ID del Pokémon (idP) al que quiere actualizar un porcentaje el ATK y el porcentaje que quiere incrementar. Si el Pokémon no existe con ese ID se mostrará un mensaje de error. En cambio, si existe se actualizará el ATK.
 - **Intercambiar estadísticas:** El usuario introducirá el ID del Pokémon (idP) y después introducirá el ID de otro Pokémon con el que quiere intercambiar las estadísticas (HP, ATK, DEF, SPE). Si uno de los dos Pokémon o los dos no existe con ese ID se le mostrará al usuario un mensaje de error y no se realizan los updates. En cambio, si todo es correcto se realizará el intercambio.
- **Transacciones:**
 - **Transaction Read Only:** hemos usado este nivel de transacción para las operaciones de búsqueda de Pokémon por ID y por tipo, ya que estas operaciones son de lectura, por lo que conviene impedir la escritura dentro de la transacción para evitar errores innecesarios.

- **Serializable:** hemos usado este nivel de aislamiento de la transacción en las operaciones de actualización, es decir, actualizar PP, aumentar ATK e intercambiar estadísticas. Gracias a ello en dichas operaciones impedimos la lectura sucia , pérdida de actualización y aparición de inconsistencias en la base de datos, ya que cualquier actualización en la base de datos supone la espera de las demás que afecten a los mismos datos hasta que se termine la transacción.