

10/6/2019



Manual Técnico Juego Pokémon.

Programación



Andrés Ceballos Rodríguez

Manual Técnico Juego Pokémon.

Programación

Índice

1. Introducción	2
2. Base de Datos	2
3. Librerías.....	5
4. Requisitos e Instalación.....	5
5. Clases Java:	7
7. Bibliografía.....	8

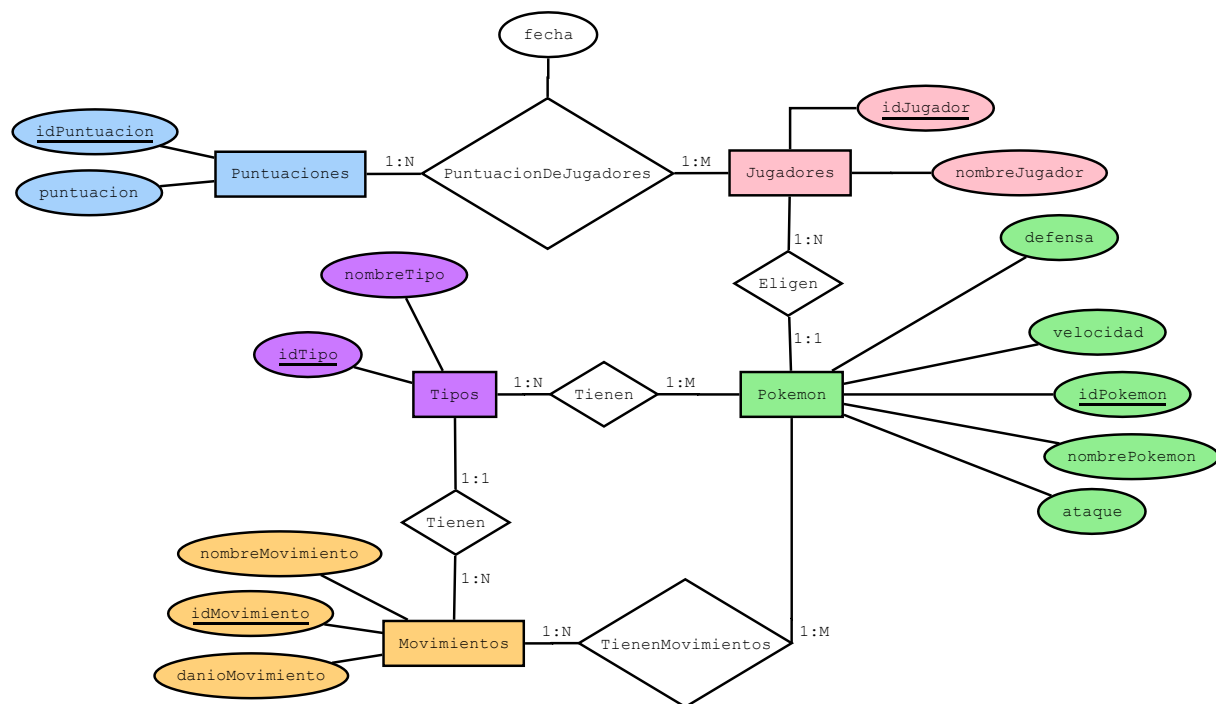
1. Introducción

En el siguiente manual, vamos a exponer los detalles técnicos de Juego Pokémon, así como los detalles de su base de datos (diagrama e-r, esquema relacional, sentencias sql...), como de las librerías externas que hemos usado para la realización del programa, los requisitos para poder utilizar el programa y la explicación de las clases Java que hemos desarrollado para la realización del mismo.

También, describiremos el proceso de la creación del ejecutable de nuestro programa.

2. Base de Datos

Modelo ERD



Modelo E/R

JUGADORES (idJugador, nombreJugador)

PUNTUACIONES (idPuntuacion, puntuacion)

PUNTUACIONESDEJUGADORES (idPuntuacionDeJugador, **idPuntuacionFK**, **idJugadorFK**, fecha)

TIPOS (idTipo, nombreTipo)

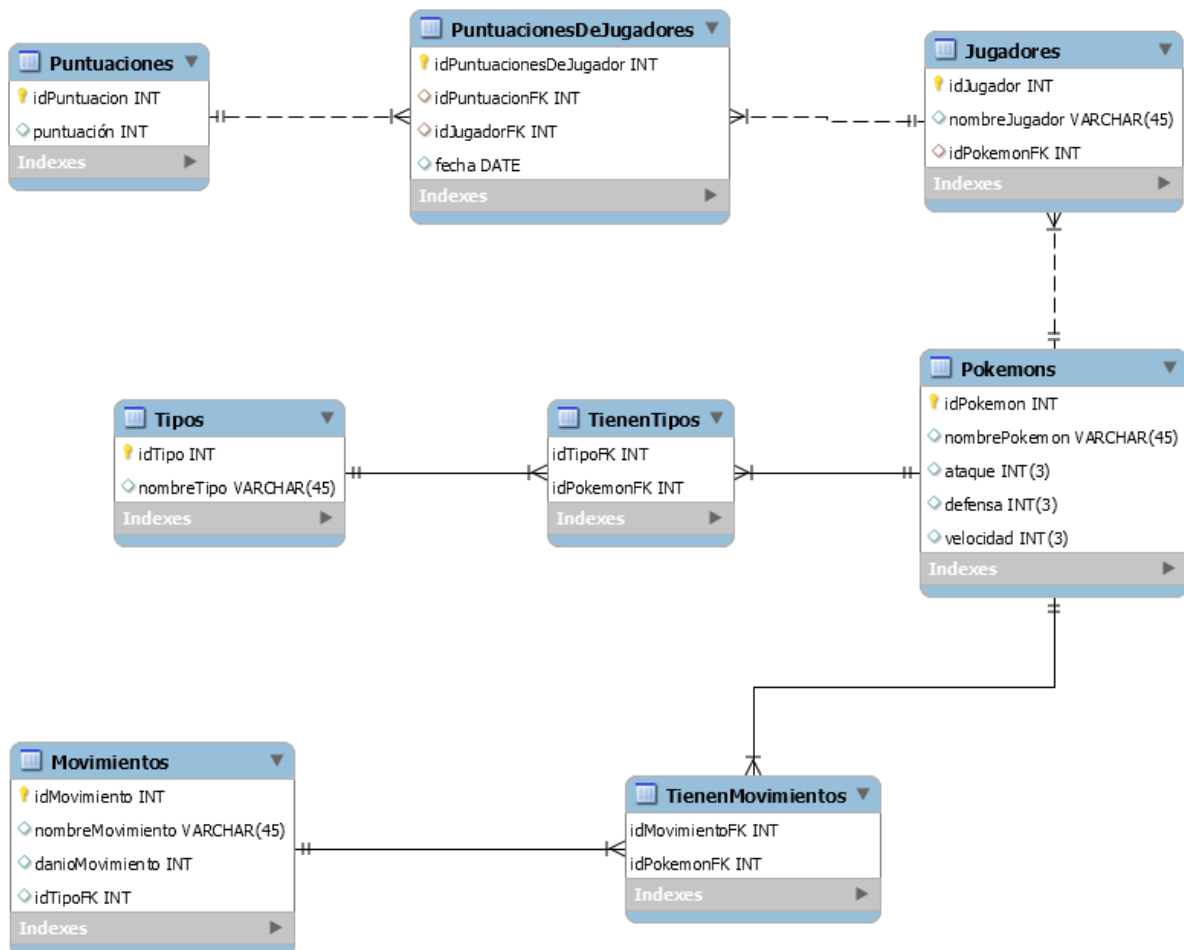
POKEMONS (idPokemon, nombrePokemon, ataque, defensa, velocidad, idTipoFK)

MOVIMIENTOS (idMovimiento, nombreMovimiento, danioMovimiento, idTipoFK)

TIENENTIPOS(idTipoFK, idPokemonFK)

TIENEN MOVIMIENTOS(idMovimientoFK, idPokemonFK)

Modelo Workbench



Sentencias SQL

Creación de la Base de datos:

```
create database JuegoPokemon char set utf8mb4 collate utf8mb4_spanish2_ci;
```

Creación de las Tablas:

Tipos:

```
CREATE TABLE `tipos` (
  `idTipo` int(11) NOT NULL AUTO_INCREMENT,
  `nombreTipo` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_spanish2_ci DEFAULT NULL,
  PRIMARY KEY (`idTipo`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_spanish2_ci;
```

Pokémons:

```
CREATE TABLE `pokemons` (
  `idPokemon` int(11) NOT NULL AUTO_INCREMENT,
  `nombrePokemon` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_spanish2_ci DEFAULT NULL,
  `puntosSalud` int(3) DEFAULT NULL,
  `ataque` int(3) DEFAULT NULL,
  `defensa` int(3) DEFAULT NULL,
  `idTipoFK` int(11) DEFAULT NULL,
  `imagen` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_spanish2_ci DEFAULT NULL,
  PRIMARY KEY (`idPokemon`),
  KEY `idTipoFK` (`idTipoFK`),
  CONSTRAINT `pokemons_ibfk_1` FOREIGN KEY (`idTipoFK`) REFERENCES `tipos` (`idtipo`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_spanish2_ci;
```

Movimientos:

```
CREATE TABLE `movimientos` (
  `idMovimiento` int(11) NOT NULL AUTO_INCREMENT,
  `nombreMovimiento` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_spanish2_ci DEFAULT NULL,
  `danioMovimiento` int(3) DEFAULT NULL,
  `idTipoFK` int(11) DEFAULT NULL,
  PRIMARY KEY (`idMovimiento`),
  KEY `idTipoFK` (`idTipoFK`),
  CONSTRAINT `movimientos_ibfk_1` FOREIGN KEY (`idTipoFK`) REFERENCES `tipos` (`idtipo`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_spanish2_ci;
```

Línea Movimientos:

```
CREATE TABLE `lineamovimientos` (
  `idMovimientoFK` int(11) NOT NULL,
  `idPokemonFK` int(11) NOT NULL,
  PRIMARY KEY (`idMovimientoFK`, `idPokemonFK`),
  KEY `idPokemonFK` (`idPokemonFK`),
  CONSTRAINT `lineamovimientos_ibfk_1` FOREIGN KEY (`idPokemonFK`) REFERENCES `pokemons` (`idpokemon`),
  CONSTRAINT `lineamovimientos_ibfk_2` FOREIGN KEY (`idMovimientoFK`) REFERENCES `movimientos` (`idmovimiento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_spanish2_ci;
```

Jugadores:

```
CREATE TABLE `jugadores` (  
  `idJugador` int(11) NOT NULL AUTO_INCREMENT,  
  `nombreJugador` varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_spanish2_ci DEFAULT NULL,  
  `idPokemonFK` int(11) DEFAULT NULL,  
  PRIMARY KEY (`idJugador`),  
  KEY `idPokemonFK` (`idPokemonFK`),  
  CONSTRAINT `jugadores_ibfk_1` FOREIGN KEY (`idPokemonFK`) REFERENCES `pokemons` (`idpokemon`)  
) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_spanish2_ci;
```

Puntuaciones:

```
CREATE TABLE `puntuaciones` (  
  `idPuntuacion` int(11) NOT NULL AUTO_INCREMENT,  
  `puntuacion` int(15) DEFAULT NULL,  
  `idJugadorFK` int(11) DEFAULT NULL,  
  PRIMARY KEY (`idPuntuacion`),  
  KEY `idJugadorFK` (`idJugadorFK`),  
  CONSTRAINT `puntuaciones_ibfk_1` FOREIGN KEY (`idJugadorFK`) REFERENCES `jugadores` (`idjugador`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_spanish2_ci;
```

3. Librerías

Para desarrollar el programa, hemos usado la librería externa siguiente:

MYSQL Connector 5.1.46 Utilizada para proporcionarnos las utilidades para conectarnos a la Base de Datos.

4. Requisitos e Instalación








Requisitos

Los requisitos para poder correr el programa son los siguientes:

- MySQL Server
- Java Versión 8
- Archivos de instalación de nuestro programa.

Instalación del programa

Para la instalación del programa, deberemos tener la carpeta de nuestro programa con los siguientes archivos:

	imagenes	10/06/2019 9:06	Carpeta de archivos	
	AyudaJuego.chm	07/06/2019 9:57	Archivo de Ayuda ...	313 KB
	instalador.bat	10/06/2019 17:37	Archivo por lotes ...	2 KB
	JuegoPokemon.jar	10/06/2019 9:15	Executable Jar File	1.016 KB
	JuegoPokemon.sql	10/06/2019 9:38	SQL Text File	8 KB
	Manual Tecnico Juego Pokemon.pdf	10/06/2019 18:27	Chrome HTML Do...	369 KB
	Manual Usuario Juego Pokemon.pdf	10/06/2019 18:27	Chrome HTML Do...	369 KB

En esta carpeta, hemos guardado todo lo necesario para el correcto funcionamiento del programa, también hemos creado un instalador.bat en el que tenemos creadas las líneas de código para poder instalar el programa.

```

@echo off
REM Habilitar tildes y ñ
chcp 65001
echo Bienvenido al programa de instalación
REM Crear Ubicaciones
mkdir "C:\Program Files\JuegoPokemon"
mkdir "C:\Program Files\JuegoPokemon\imagenes"
REM Mover los ficheros
copy "C:\Users\Andres Ceballos\Desktop\JuegoPokemon\*.*)" "C:\Program Files\JuegoPokemon"
copy "C:\Users\Andres Ceballos\Desktop\JuegoPokemon\imagenes\*.*)" "C:\Program Files\JuegoPokemon\imagenes"

cd "C:\Program Files\JuegoPokemon"
REM Montar la BD
echo Montando la base de datos: Introduzca clave de root
mysql -u root -p -e "CREATE DATABASE JuegoPokemon CHARACTER SET utf8 COLLATE utf8_spanish2_ci;"
echo Importando la base de datos: Introduzca clave de root
mysql -u root -p JuegoPokemon<JuegoPokemon.sql
echo Creando el usuario: Introduzca clave de root
mysql -u root -p -e "CREATE USER 'usuarioJuego'@'localhost' IDENTIFIED BY 'Stadium2018;'"
echo Dando permisos al usuario: Introduzca clave de root
mysql -u root -p -e "GRANT ALL ON JuegoPokemon.* TO 'usuarioJuego'@'localhost'"
REM Iniciar
echo Iniciando el programa
JuegoPokemon.jar
  
```

Para instalar el programa deberemos pulsar sobre el instalador.bat y a continuación seguiremos los pasos que nos va indicando el instalador.

Cuando tengamos creado el instalador, nos dirigiremos a la carpeta donde hayamos instalado el programa y lo ejecutaremos.

5. Clases Java:

Lo siguiente que vamos a exponer es una breve explicación de cada clase Java que hemos desarrollado que componen nuestro programa.

Ayuda.java: En esta clase, ejecutamos la ayuda de nuestro Juego.

BaseDatos.java: En esta clase, tenemos todas las funciones necesarias en las que hacemos inserts, selects, etc.

EstadisticaMovimiento.java: En esta clase, podemos ver las estadísticas de un movimiento de un Pokémon elegido anteriormente.

EstadisticasPokemon.java: En esta clase, podemos ver las estadísticas del Pokémon.

Funciones.java: Aquí encontramos la función del cálculo del daño de un movimiento.

MenuPrincipal.java: Esta clase es la clase principal en la que encontramos el menú principal.

NuevaPartida.java: En esta clase, elegimos los nombres de los jugadores.

Partida.java: Esta es la clase en la que transcurre la batalla.

Seleccion1.java: En esta clase, el primer jugador seleccionará el Pokémon con el que va a jugar.

Seleccion2.java: En esta clase, el segundo jugador seleccionará el Pokémon con el que va a jugar.

Top10.java: En esta clase, creamos el Top 10 de mejores jugadores.

El código de las clases java lo podremos consultar en la carpeta adjuntada con el proyecto llamada clases.

7. Bibliografía

Temario Tema 9 Programación. (10 de 06 de 2019). Obtenido de Temario Tema 9 Programación:
http://aulastudium.com/pluginfile.php?file=%2F11026%2Fmod_resource%2Fcontent%2F2%2FPR-TEMA9%20Acceso%20a%20Bases%20de%20Datos.pdf