



# Manual Técnico

Juego en Consola

José Rafael Morente González

201801237 Sección C

# Contenido

Clase ControladorBatalla .....	2
Clase ControladorPokemon.....	3
Clase IngresoDatos.....	3
Clase Menu .....	4
Clase Pokemon y BatallaPokemon .....	5

# Juego en Consola

## Clase ControladorBatalla

`agregarBatalla(String jugador1, Pokemon pokemon1, Pokemon pokemon2, String jugador2, Pokemon pokemon3, Pokemon pokemon4)`

Este método se encarga de agregar las batallas en un vector de tipo batalla. Los parámetros que solicita es el nombre del primer jugador y del segundo jugador, pokemon1 y pokemon2 son los dos Pokémon que utiliza el primer jugador y Pokémon 3 y Pokémon 4 son los Pokémon del segundo jugador.

`mostrarBatallas()`

Este método se encarga de mostrar el historial de todas las batallas Pokémon.

`empezarBatalla(BatallaPokemon batalla)`

Empezar batalla es un método el cual se encarga de realizar la batalla entre los dos jugadores, hasta que los dos Pokémon de cualquiera de los jugadores llegue a vida cero.

`topGanadores()`

Este método imprime a los 5 jugadores que han ganado una o más partidas.

`topPerdedores()`

Este método imprime a los 5 jugadores que han perdido una o más partidas.

`pokemonMasElegido()`

Este método imprime a los Pokémon que más veces han sido elegidos.

`pokemonMenosElegido()`

Este método imprime a los Pokémon que no han jugado ni una partida.

`validarNumero()`

Este método valida que se haya ingresado un número correctamente de lo contrario tira un mensaje de error y repite el método para ingresar la cantidad numérica.

`valorAleatorio(int n)`

Este método devuelve un valor aleatorio menor o igual al número que se le manda como parámetro.

`getInstancia()`

Este método es un patrón de diseño que realiza una instancia única de la clase.

## Clase ControladorPokemon

`cargaInicial(String[] args)`

Este método se encarga de ingresar a los Pokémon desde un inicio a la aplicación.

`agregarPokemon(String nombrePokemon, String imagenPokemon)`

Este método agrega a los Pokémon en un vector.

`valorAleatorio(int a, int b)`

Este método retorna un valor aleatorio siempre y cuando este dentro del intervalo cerrado de a y b.

`mostrarPokemones()`

Este método retorna un listado de todos los Pokémon con sus atributos asignados.

`mostrarListaPokemones()`

Este método retorna una lista de todos los Pokémon con su nombre.

`buscarPokemon(int id)`

Este método busca un Pokémon dentro del vector por su posición si no lo encuentra retorna null.

`actualizarPokemon(int id, String nombrePokemon, int vidaPokemon, int ataquePokemon)`

Este método actualiza a los Pokémon en un vector.

`cargarImagenes()`

Este método carga una serie de String como imagen dentro de un vector de imágenes.

`getInstancia()`

Este método es un patrón de diseño que realiza una instancia única de la clase.

## Clase IngresoDatos

`Ingreso()`

Este método realiza una lectura de entrada de datos desde el teclado haciendo uso de la clase `BufferedReader`.

`getInstancia()`

Este método es un patrón de diseño que realiza una instancia única de la clase.

## Clase Menu

### `mostrarMenu()`

Este método muestra el menú principal de la aplicación con la cual interactúa el usuario final en el cual puede realizar lo siguiente:

1. Administrar Pokémon
2. Administrar Batallas
3. Nueva Partida

### `mostrarMenuPokemon()`

Este método muestra el menú de los Pokémon en los cuales se puede realizar lo siguiente:

1. Ver lista de los Pokémon
2. Editar Pokémon
3. Lista de los Pokémon más elegidos
4. Lista de los Pokémon menos elegidos

### `mostrarMenuPartidas()`

Este método muestra el menú de las batallas en la cual se puede realizar lo siguiente

1. Ver registro de partidas
2. Top 5 Jugadores Ganadores
3. Top 5 Jugadores Perdedores

### `mostrarMenuPartida()`

Este método nos retorna a una nueva partida Pokémon en la cual se crea la nueva batalla y se procede a jugar.

### `validarPokemon()`

Este método valida si el Pokémon seleccionado al crear una batalla es correcto de lo contrario pide de nuevo los datos.

### `validarNumero()`

Este método valida que se haya ingresado un número correctamente de lo contrario tira un mensaje de error y repite el método para ingresar la cantidad numérica.

### `getInstancia()`

Este método es un patrón de diseño que realiza una instancia única de la clase.

# Clase Pokemon y BatallaPokemon

## Getter's & Setter's

Los Setter's & Getter's son métodos de acceso lo que indica que son siempre declarados públicos, y nos sirven para dos cosas

- Getters: Obtener información
- Setters: nos sirve para asignar un valor al atributo.