

## **MANUAL TECNICO**

### **Introducción.**

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que necesitan de la instalación de Java para su correcto funcionamiento. Java es rápido, seguro y fiable. Con Java se pueden realizar desde aplicaciones de portátiles, centros de datos consolas para juegos, teléfonos móviles, internet, etc.

En este proyecto se realizó con Java una simulación de un combate doble del famoso juego Pokemon. Se programó la aplicación para ser ejecutada dentro de un cmd, con menús y validaciones simples. Se hizo uso de arreglos para almacenar los datos necesarios y métodos para realizar tareas específicas para su correcto funcionamiento.

### **Plataforma de Ejecución.**

La plataforma de Java es el nombre del entorno de computación creado por Sun Microsystems. Esta plataforma es capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java. En este caso, la plataforma de ejecución no es un hardware o sistema operativo, sino que es una máquina virtual que se encarga de la ejecución de las aplicaciones, y un conjunto de bibliotecas que ofrecen una funcionalidad común.

La plataforma java se compone de varias tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real. Las aplicaciones Java pueden usarse de varias formas, como por ejemplo colocándolas en una página web. Para el desarrollo de aplicaciones Java se utilizan un conjunto de herramientas conocidas como JDK.

El Java Development Kit (JDK) es un software que provee herramientas de desarrollo para la creación de programas en Java.

Lo principal de la plataforma Java es el concepto de un procesador “virtual” que ejecuta programas escritos en Java. En concreto, ejecuta el código resultante de la compilación del código fuente, conocido como bytecode (0s y 1s). Este “procesador” es la máquina virtual de Java, que se encarga de compilar el bytecode en instrucciones nativas de la plataforma destino.

En java existen las bibliotecas, que es una cantidad de código para simplificar una tarea de programación. Este código toma la forma de un conjunto de bibliotecas dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Lo que hace la plataforma de Java es ofrecer un conjunto de estas bibliotecas, que contienen mucha de las funciones reutilizables disponibles en los sistemas operativos actuales.

## Diccionario de Métodos Usados

- **asociarImgs.** Este método asociaba los valores Strings de las imágenes a una matriz.
- **pkIniciales.** Este método asignaba los primeros 6 pokemon que los jugadores iban a tener para escoger.
- **encabezado.** Este método imprime el encabezado de la aplicación, el cual es el logo de Pokemon.
- **menuPrincipal.** Este método contiene el menú principal con las opciones “Ingresar al Sistema” y “Salir”.
- **submenu.** Este método contiene el menú que le permitía al usuario ingresar como administrador, jugar o regresar al menú principal.
- **menuAdmin.** Este método contiene el menú de administrador que contiene las opciones “Lista de Pokemon”, “Agregar Pokemon”, “Editar Pokemon”, “Reportes” y “Regresar al menú principal”.
- **mostrarLista.** Este método mostraba la lista de los Pokemon que pueden usar los jugadores para jugar. Mostraba el número asignado a cada Pokemon, la imagen, nombre, los puntos de salud y ataque.
- **agregarPk.** Este método agregaba un nuevo Pokemon hecho por el usuario. Solicita que seleccione una imagen, escriba un nombre y asigna valores aleatorios a los puntos de salud y ataque del Pokemon agregado.
- **mostrarImg.** Este método muestra la galería de imágenes de Pokemon disponibles.
- **crearPk.** Este método asigna los valores establecidos en agregarPk a la matriz de Pokemon.
- **editarPk.** Este método le permite al usuario editar a los Pokemon que se encuentran dentro de la matriz de Pokemon. Le pide que seleccione al Pokemon a editar, que escoja la nueva imagen que le asociara, nombre y asigna nuevos valores aleatorios para los puntos de salud y ataque.
- **llenarPk.** Este método ingresa dentro de la matriz de Pokemon los datos solicitados en editarPk. Unicamente en la posición del Pokemon que el usuario deseaba editar.
- **menuReportes.** Muestra el menú de Reportes con las opciones “Registro de Partidas” y “Regresar al menú principal”.
- **registro.** Este método imprime los datos de todas las partidas que se han jugado.
- **menuJugar.** Muestra el menú para jugar donde están las opciones “Batallar” y “Regresar”.
- **encabezadoBatalla.** Imprime un encabezado para los combates.
- **solicitarDatosBatalla.** Este método le pide a los usuarios sus nombres y los Pokemon que desean usar. Esto para la batalla.
- **mostrarPkB.** Este método muestra los nombres de cada Pokemon que se puede usar en batalla junto con su número asociado.
- **asignarValores.** Este método asigna los valores, solicitados en solicitarDatosBatalla y los valores encontrados en la matriz de Pokemon, a variables para su fácil manejo dentro de la batalla.

- **turno.** Este método genera el turno aleatorio.
- **j1empieza.** Este método se inicia si el turno aleatorio nos dice que el jugador 1 es el que iniciará la batalla.
- **j2empieza.** Este método se inicia si el turno aleatorio nos dice que el jugador 2 es el que iniciará la batalla.
- **j1ataca.** Este método realiza el ataque de j1 hacia algún Pokemon de j2. Dentro de este método también se realizan validaciones por si alguno de los Pokemon que atacará o ataca ya se encuentra en estado “Muerto”.
- **j2ataca.** Este método realiza el ataque de j2 hacia algún Pokemon de j1. Dentro de este método también se realizan validaciones por si alguno de los Pokemon que atacará o ataca ya se encuentra en estado “Muerto”.
- **finalizador.** Este método asigna todos los valores necesarios a una matriz que nos servirá para el registro de las partidas.
- **validarVida.** Este método realiza validaciones con la vida de los Pokemon usados tanto por jugador 1, como con jugador 2. Si la vida de estos es menor o igual a 0 les asigna un valor de estado “Muerto”.
- **imprimir.** Este método imprime los datos que se muestran durante la batalla.
- **finalizarCombate.** Este método finaliza el combate y muestra las opciones a los usuarios de “Jugar otra vez” o “Regresar al menú principal.”