

“Juego de Pokémon”

Integrantes:

Andrés Camacho Sánchez

Puko chacón Valverde

Sonia Umaña Piñar

Julián Madrigal Gómez

Brandon Ureña Porras

Universidad Fidélitas

SC-304: Estructura de datos

Profesor: Pablo Gómez Ramírez.

16 de noviembre del 2023

III Cuatrimestre, 2023



- **Recolección de requerimientos:**

Reunir los requisitos del proyecto es un paso crítico en el proceso de desarrollo de software porque sienta las bases para comprender las necesidades y expectativas de los usuarios, clientes y partes interesadas. Como una observación de parte del profesor fue poner un screenshot del Código. Los requisitos de colección deben ser específicos y detallados en el contexto del proyecto del juego Pokémon en el que estamos trabajando. Aquí damos ciertas indicaciones sobre algunas instrucciones para completar este paso:

Para el Proyecto de juegos de Pokémon implementamos todas las clases solicitadas y necesarias, así como ir desarrollando cada punto para que el Código logre correr con éxito, el Código cuenta con 23 clases, cada una con un nombre diferente y funcionalidad distinta.

- **Propuesta de manejo de interfaz gráfica:**

Para el diseño de la interfaz grafica

1. En la pantalla principal: se muestra la pila de entrenadores y sus Pokémon listos para la batalla, junto con el botón de iniciar una batalla nueva
2. En la pantalla de batalla: muestra la información sobre los entrenadores involucrados, como sus nombres y los Pokémon que han seleccionado la batalla. Por otro lado, hay una sección para mostrar los turnos de cada uno, los Pokémon en batalla y el estado de la batalla y un botón para realizar un nuevo turno.

Descripción de las interacciones de los elementos de interfaz

1. Pila de entrenadores: la pila se mostrará en una lista en la pantalla principal, al seleccionar un entrenador de la pila, se puede ver la información detallada sobre los Pokémon de ese entrenador. El botón de iniciar batalla, se iniciará la batalla entre el entrenador en la cima de la pila y un entrenador adversario.
2. Información de los entrenadores: se mostrarán los nombres de los entrenadores y los Pokémon seleccionados para la batalla, la sección de turno mostrará información sobre el turno actual y permitirá avanzar al siguiente turno. La sección de Pokémon en batalla muestra los pokemones en batalla, nombres, niveles y la vida restante. El botón de nuevo turno al darle clic se ejecuta un nuevo turno calculando y mostrando el resultado de ataques.
3. La interfaz se actualizará automáticamente para reflejar cualquier cambio en el estado de los pokemones durante la batalla.

- **Propuesta de manejo para la batalla:**

La parte de gestión de batallas Pokémon es una parte fundamental del proyecto. Para el Proyecto de juegos de Pokémon implementamos todas las clases solicitadas y necesarias, así como ir desarrollando cada punto para que el Código logre correr con éxito, el Código cuenta con 23 clases, cada una con un nombre diferente y funcionalidad distinta.

La clase es la encargada de manejar la lógica de las batallas entre un entrenador usuario y entrenador adversario. Para resumir las principales características de la clase:

La clase contiene un constructor que recibe un entrenador usuario y un entrenador adversario para iniciar la batalla. El método de iniciar batalla utiliza un while para continuar con los turnos hasta que los entrenadores se quedan sin Pokémon con vida, cada turno se elige un Pokémon para la batalla de cada entrenador y se llama al método realizar turno.

El método de realizar turno, se determina el ataque del usuario y un ataque aleatorio, se calcula el daño causado por cada ataque utilizando el método calcular daño, el método de calcular daño es para saber cuanto daño inflige un Pokémon a otro, utilizando un valor de daño base generado aleatoriamente y ajusta el daño según el ataque del Pokémon atacante y el tipo de Pokémon defensor. El método de obtener nombre ataque devuelve el nombre del ataque realizado por un Pokémon según el Código ataque. El método de determinar ganador, este método determina el ganador basándose en si el entrenador usuario tiene Pokémon con vida.

El daño base para cada ataque es de 80.

- **Propuesta de estructura de datos:**

Una elección adecuada de estructura de datos seleccionados para manejar eficientemente la información del juego.

Función de las estructuras en el programa: queremos una ejecución eficiente y sin problemas de la estructura de datos

Pilas:

Almacenamiento de Pokémon en pilas: Las pilas son ideales para el almacenamiento de los Pokémon de los entrenadores. Cuando un entrenador se queda sin su Pokémon actual en la batalla, el siguiente Pokémon disponible se toma de la parte superior de la pila, lo que simula la mecánica de cambio de Pokémon. Al eliminar un Pokémon de la batalla, este se "desapila", y el siguiente en la pila se convierte en el activo. Una vez que un Pokémon es derrotado, se puede "desapilar" de la pila y se elimina. Esto con la intención de que si la pila de entrenadores es nula y no contiene ningún elemento esto se le entrega como victoria al usuario.

Listas:

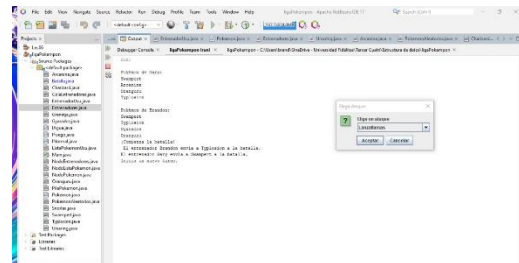
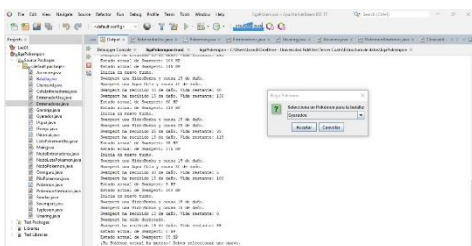
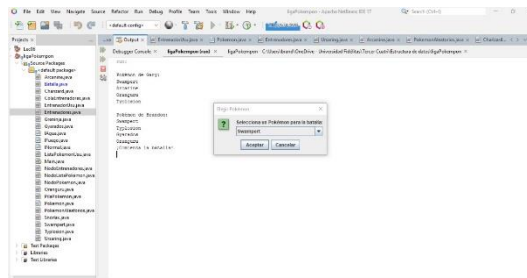
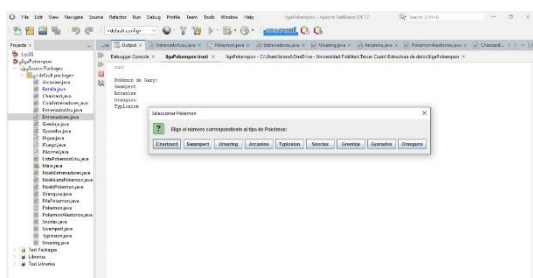
Manejo de Pokémon del usuario: Las listas son útiles para el usuario, ya que le permiten tener una colección de Pokémon de los cuales puede elegir cuando su Pokémon actual es derrotado en batalla. Esta lista le ofrece al usuario la flexibilidad de elegir el Pokémon que desee usar a continuación, dándole cierto control estratégico sobre su equipo en la batalla.

Colas:

Almacenamiento de entrenadores: Las colas son utilizadas para manejar la entrada y salida de los entrenadores en las batallas. Los entrenadores se agregan a la cola en orden de llegada al desafío. Una vez que un entrenador es derrotado, es el primero en salir de la cola, permitiendo que el siguiente entrenador en la fila sea el próximo desafiante. Esta lógica simula la dinámica de los desafíos en una secuencia de manera secuencial, donde el próximo oponente siempre es el que ha estado esperando en la cola.

Estas estructuras son vitales para el desarrollo del juego de Pokémon, ya que modelan con precisión el comportamiento esperado de los Pokémon, los entrenadores y las interacciones entre ellos, lo que garantiza una dinámica fluida y una experiencia de juego coherente con las reglas establecidas en la serie Pokémon.

- **Implementación de las propuestas:**



- Ejecución de Código

