

Como núcleo del código, la clase Main deberá ejecutar el Menú de juego, al llamar la clase con este mismo nombre a modo de constructor. En este Menú, se realizarán todos los despliegues de mensaje o texto a modo de información visual que acompañará la ejecución de los distintos programas (de las clases) que compondrán el sistema de juego.

Cada componente como elemento, entrenador, ataque, pokemon y gimnasio tendrán su propia clase, de forma que unas utilizarán a otras para sus atributos; séase mediante listas, por ejemplo.

En sí, la estructura consistirá en que, primero, se le deberá de indicar al usuario, la introducción de su nombre, mediante una entrada de tipo Scanner (estas, que serán utilizadas a lo largo del juego, serán utilizadas acompañadas, en todo momento, del manejo de errores que pueden suponer). Con este nombre, el jugador será agregado a la lista de entrenadores. Luego, a modo de método, se deberá de indicar al usuario la elección de tres pokemones, los cuales serán introducidos a la lista atributo del entrenador asociado al jugador. Esto dentro de un bucle, para garantizar la elección correcta de los pokemones. En segundo lugar, se le deberá de mostrar la lista de entrenadores del gimnasio al usuario, mediante un método, propio de la clase Gimnasio, que muestre de forma ascendente a los entrenadores de acuerdo con su nivel (dificultad).

Luego, se procederá con el primer combate. El sistema de combate deberá de hacer uso de la clase Batalla, la cual tendrá como función principal empezarBatalla, además de otras funciones para la gestión de los turnos de ambos entrenadores, de los pokemones, ataques, vida y el tiempo durante el combate, que respaldaran (funciones privadas) a la primera función. Una de las funciones que ayudará al cambio de pokemones, será un bucle que haga uso de la gestión de vida de los pokemones, pues si uno de los pokemones que se enfrentan pierde toda su vida (sin necesidad de ser negativa), el entrenador deberá de elegir su siguiente pokemon; deberá de haber otra condición sobre el atributo boolean del entrenador, pues si uno de los dos no es el jugador o usuario, se deberá de elegir autónomamente su pokemon de forma aleatoria. Si es por parte del usuario, a este deberá de mostrársele, mediante otra función, su lista de pokemones a modo de opciones, sin posibilidad de escoger el que fue derrotado, para continuar con el combate.

Siguiendo esta línea, la función principal de la batalla hará uso de la gestión de la cantidad de pokemones de los dos jugadores, en la que, si uno de los dos se queda sin pokemones, se detiene la batalla (y el bucle), y el jugador sin pokemones pierde. Principalmente, esta función recibirá siempre dos variables de tipo Entrenador, el

jugador y un entrenador. En caso de que el jugador gane, se deberá de utilizar esta información (¿return?) para continuar o, si pierde, para que pueda intentarlo o rendirse. Es sistema funcionará por for-each que, en primer lugar, iterará cada entrenador de la lista de entrenadores de Gimnasio, hasta el líder, para los combates, y en cada iteración deberá de haber un while en el que, si el jugador decide rendirse, no funcione más, y pueda dejar de ejecutarse el programa (aquí deberá de utilizarse una variable que rompa con el bucle principal del Menú). Aunque si este desea volver a intentarlo, el while debe mantener el programa en la misma iteración.

De esta forma el jugador, idealmente, deberá de ir ganando los combates hasta llegar al líder (entrenador con mayor nivel). Al final de cada iteración, luego de un while, deberá de haber un if que tenga como condición la información dada por la función empezarBatalla, que, en caso de entrar, registre la batalla ganada por parte del jugador. Si contra el líder el jugador gana el combate, deberá de mostrársele un mensaje de felicitación, además del historial de la partida; esto, mediante un if que posea como requerimiento el registro antes mencionado de las batallas ganadas. Luego, como condicional último, se le deberá de preguntar al jugador si desea volver a empezar otra partida o finalizar el juego, lo que romperá o no con el bucle principal.

Esto es con respecto al manejo de los combates, de forma general. Con respecto a los ataques, cada pokemon poseerá una lista de 4 ataques, por lo cual, en la clase de Batalla, habrá una función que permita, en la batalla (función antes mencionada), elegir entre los ataques, o tomar uno aleatoriamente (en caso de que el entrenador sea o no el jugador). Dependiendo del ataque tomado (por cualquiera de las dos partes), el atributo de daño en específico hará más o menos daño (restará más o menos vida) dependiendo del tipo del pokemon que recibe el daño, de la probabilidad de golpe crítico del pokemon que ataca, o de la precisión del ataque. En sí, el daño que recibe el pokemon será calculado en otra función privada dentro de la clase Batalla. La elección de los ataques, estarán también en función de los puntos de poder del pokemon.

Otra de las funciones importantes, dentro de esta misma clase Batalla, será la elección de quien ataca (turnos), la cual dependerá de la velocidad de los pokemones elegidos en el combate (la función que se realizó en clase para la simulación corta de una batalla pokemon puede servir).