# Laboratorio 1 - temario C - Alejandro Sagui

Análisis

## Requisitos funcionales e indispensables para mi programa

* Que dos entrenadores se registren en la pelea
* Cada jugador o entrenador puede poder escoger 3 pals distintos
* Definir 4 batallas distintas en el combate
* Cada ronda debería de poder hacer:
  + Que los jugadores escojan un pal que no hayan usado antes
  + Que el player decida si su peleador haga un movimiento especial o ataque normal
  + Ver el efecto del ataque entre los peleadores como el fuego o la planta, y a eso sumarle 20 de efectividad, restarle 10 de debilidad o no hacer nada por neutral.
  + Ver si el número escogido cae en la probabilidad para que la habilidad especial se active
  + Si la habilidad especial se activa, se debería de sumar el efecto en la ronda actual y en la siguiente
  + Debería de poder diferenciar los puntajes entre el más alto para ver quien gana
  + Debería de definir el empate por si tienen los mismos puntos
* Llevar el conteo de quien gana la ronda y poder decir al final quien gano o si hubo un empate
* Mostrar un resumen del combate

## Clases y su propósito

* Tipo del pals
  + Dice si el pals es de agua, planta, fuego o electricidad
  + Facilita la efectividad de los ataques y sus cálculos
* Armamentoespecial
  + Modela la habilidad especial del pals
  + Tiene el nombre, tipo de efecto, su valor y su probabilidad para activarse
* Pal
  + El animal o criatura que cada entrenador escoge
  + Tiene nombre, tipo, ataque básico, su armamento especial y su defensa básica
* Entrenador
  + Es el jugador con su nombre y su lista de pals
  + Tiene las rondas que gana y controla la selección del pals antes de empezar la rona
* ronda
  + encapsula un combate entre dos pals
  + recibe a los dos pals seleccionados y saca a un ganador
* combate
  + tiene las 4 rondas entre los entrenadores
  + lleva el orden entre las rondas, las victorias, empates y derrotas

## Atributos de mis clases y su propósito

Tipo

* FUEGO, AGUA, PLANTA, ELÉCTRICO

ArmamentoEspecial

* String nombre: identifica la habilidad especial.
* Efecto tipoEfecto: indica si aumenta el ataque, aumenta la defensa o daña al pal rival directamente.
* int valor: magnitud del efecto.
* double probabilidad: probabilidad de activación expresada en 30%.

Pal

* String nombre: nombre del Pal.
* Tipo tipo: tipo elemental del Pal.
* int ataqueBase: valor basído del pal.
* int defensaBase: valor basíco del pal.
* ArmamentoEspecial armamento: habilidad especial.
* boolean especialActivoSiguiente: indicador si el efecto del armamento activo en la ronda anterior aplica a la siguiente.

Entrenador

* String nombre: nombre del entrenador.
* List<Pal> disponibles: lista de los 3 Pals seleccionados por el jugador.
* Set<Pal> usados: los Pals que ya usaron en las rondas anteriores.
* int rondasGanadas: contador de victorias de el entrenador.

Ronda

* Pal pal1, pal2: referencias a los Pals que van a pelear.
* int modificadorTipo1, modificadorTipo2: efecto calculado de tipo.
* boolean activado1, activado2: esto va a decirme si la habilidad especial se activa.

Combate

* Entrenador entrenador1, entrenador2: participantes del combate.
* int totalRondas: número total de rondas

## 4. Métodos por Clase

ArmamentoEspecial

* boolean seActiva(): calcula si la habilidad se activa basado en probabilidad.
* int aplicarEfectoAtaque(int ataque): da el nuevo ataque tras aplicar el efecto, según tipoEfecto y valor.

Pal

* int calcularModificadorTipo(Tipo rival): retorna +20 si efectivo, -10 si débil, 0 si neutral.
* int getAtaqueTotal(Pal rival): combina ataqueBase, modificador de tipo y el aumento de habilidad especial si es que hay.
* void registrarUsoEspecial(): marca que el armamento se va a usar en la siguiente ronda.

Entrenador

* Pal elegirPal(): solicita y retorna un Pal disponible que no esté en usados, añade a usados.
* void incrementarVictoria(): incrementa rondasGanadas en 1.

Ronda

* ResultadoSimulacion jugar(): ejecuta la ronda:
  1. Calcula modificadores de tipo.
  2. Mira si se activa alguna habilida especial.
  3. Calcula ataques totales.
  4. devuelve el resultado (ganador o empate).

Combate

* void iniciar(): ciclo de 4 rondas donde:
  1. Cada entrenador elige un Pal.
  2. Se crea un objeto Ronda con los Pals.
  3. Se llama a jugar() y se actualizan rondasGanadas.
* void mostrarResultados(): muestra ronda por ronda los resultados y anuncia el ganador final.

# Diseño

Diagrama

El contenido generado por IA puede ser incorrecto.