

Laboratorio 1:

II. Análisis

2.1 Requisitos funcionales

- El programa debe permitir crear dos entrenadores diferentes (dos usuarios)
- Cada entrenador debe poder escoger 4 Pokémon diferentes al inicio
- Cada Pokémon debe tener un nombre, tipo, ataque, defensa y una habilidad especial
- Cada una de las habilidades especiales debe tener un nombre, un efecto y una probabilidad de ser activado
- Al inicio de cada ronda, cada entrenador debe escoger un Pokémon sin repetir
- Se debe escoger un numero aleatorio y compararlo con la probabilidad de la habilidad especial para saber si se puede usar o no
- Si la habilidad se activa, su efecto se suma al ataque actual y al siguiente.
- El ganador de la ronda es quien tenga más ataque total
- La defensa de un Pokémon se le resta al ataque del otro
- Se debe poder comparar los tipos de Pokémon
- El juego tiene 4 rondas y gana quien gane más rondas
- Puede haber empates si los ataques finales son iguales

2.2 Clases

Clase: Entrenador			
Propósito:	Representa a un jugador que participa en la batalla. Cada entrenador tiene 4 Pokémon, lleva el conteo de rondas ganadas y gestiona si la habilidad especial está activa.		
Atributos:			
Atributo:	Descripción:	Tipo de dato:	Visibilidad:
nombre:	Identifica a los entrenadores	String	Private
pokemones:	Lista los pokemones disponibles de cada uno de los entrenadores (4 pokemones)	Lista	Private
efectoActivo:	Indica si la habilidad especial está activa	Boolean	Private

	por segunda ronda consecutiva.			
rondasGanadas:	Cuenta cuántas rondas ha ganado el entrenador		Private	
Métodos:				
Método:	Descripción:	Tipo de retorno:	Visibilidad:	Parámetros:
Entrenador:	Constructor	-	Public	Nombre, pokemones
seleccionarPokemon():	Devuelve un Pokemon disponible para la ronda	Pokemon	Public	-
ganarRonda():	Aumenta el valor de rondas ganadas por 1	void	Public	-
getNombre():	Devuelve el nombre del entrenador	String	Public	-
habilidadActiva():	Indica si la habilidad especial sigue activa	Boolean	Public	-
activarHabilidad():	Activa la habilidad esta ronda y la siguiente	Void	Public	-
reiniciarHabilidad():	Desactiva la habilidad después de dos rondas	Void	Public	-

Clase: Pokemon			
Propósito:	Representa a los pokemones que participan en cada una de las rondas con sus atributos		
Atributos:			
Atributo:	Descripción:	Tipo de dato:	Visibilidad:
nombre:	Nombre del Pókemon	String	Private
tipo:	Tipo elemental del pokemon (fuego, agua, planta, eléctrico)	String	Private
ataque:	Valor del ataque base	Int	Private
defensa:	Valor de defensa base	Int	Private
habilidad:	Objeto de “habilidadEspecial” asignada	HabilidadEspecial	Private

Métodos:				
Método:	Descripción:	Tipo de retorno:	Visibilidad:	Parámetros:
Pokemon():	Constructor (tiene 16 opciones para que el entrenador escoga sus pokemones)	-	Public	String nombre, String tipo, int ataque, int defensa, HabilidadEspecial habilidad
Pokemon():	Copia del constructor con solo los pokemones escogidos	-	Public	Pokemon original
calcularPoder():	Calcula el ataque total, considerando ventaja de tipo y si hay habilidad activa	int	Public	String tipoEnemigo, boolean habilidadActiva
getTipo():	Devuelve el tipo del Pokémon	String	Public	-
getNombre():	Devuelve el nombre del pokemon	String	Public	-
getHabilidad():	Devuelve un objeto tipo HabilidadEspecial	HabilidadEspecial	Public	-

Clase: HabilidadEspecial					
Propósito:		Guarda los valores y detalles de la habilidad especial			
Atributos:					
Atributo:		Descripción:	Tipo de dato:	Visibilidad:	
nombre:		Nombre de la habilidad	String	Private	
efecto:		Tipo de efecto de la habilidad (aumenta ataque, defensa o daña al enemigo)	String	Private	
valor:		Valor numérico del efecto de la habilidad	Int	Private	
probabilidad:		Probabilidad de que se active (1-100%)	Int	Private	
Métodos:					
Método:		Descripción:	Tipo de retorno:	Visibilidad:	Parámetros:

HabilidadEspecial():	Constructor	-	Public	String nombre, String efecto, int valor, int probabilidad
Activar():	Compara un numero aleatorio con la probabilidad para determinar si se puede usar la habilidad especial o no	Boolean	Public	-
getNombre():	Devuelve el nombre de la habilidad especial	String	Public	-
getEfecto():	Devuelve el tipo de efecto	String	Public	-
getValor():	Devuelve el valor del efecto	Int	Public	-

Clase: Batalla				
Propósito:	Controla el flujo de las 4 rondas y determina al ganador			
Atributos:				
Atributo:	Descripción:	Tipo de dato:	Visibilidad:	
entrenador1:	Primer jugador	Entrenador	Private	
entrenador2	Segundo jugador	Entrenador	Private	
Métodos:				
Método:	Descripción:	Tipo de retorno:	Visibilidad:	Parámetros:
Batalla():	Constructor	-	Public	Entrenador entrenador1, Entrenador entrenador2
iniciarBatalla():	Ejecuta las 4 rondas y determina el ganado	Void	Public	-
jugarRonda():	Ejecuta la ronda de una ronda	Void	Private	-
calcularVentaja():	Devuelve modificador de la diferencia de los tipos de pokemon	int	private	String tipo1, String tipo2

III. Diseño

