

# Trabajo Práctico - Algoritmos 3

Brasburg. Cano. Raik  
FIUBA

September 8, 2023

# Objetivos

Los objetivos de la primera parte del trabajo práctico son:

- Analizar el problema planteado correctamente.
- Diseñar un modelo general de solución utilizando correctamente los conceptos de programación orientada a objetos.
- Utilizar los conceptos de programación funcional cuando se requiera

## 1 Dominio del Trabajo Práctico

En los confines de un mundo digital, donde la creatividad se fusiona con la tecnología, y la pasión se enciende como un fuego eterno, nos adentramos en un viaje épico. Un viaje que nos llevará a través de la vastedad de un universo repleto de criaturas misteriosas y poderes inimaginables. Bienvenidos a la creación de un juego de batallas de Pokémon, donde el destino se forja en líneas de código y la estrategia es la clave para alcanzar la gloria.

En este desafío titanesco, no solo seremos programadores, sino también entrenadores de Pokémon, guiando a nuestras criaturas a la victoria en combates legendarios. Nuestro poder radicará en la habilidad para diseñar y desarrollar un mundo virtual donde la rivalidad se transforme en un arte, donde cada elección de programación sea un paso hacia la conquista.

A medida que enfrentemos obstáculos y desafíos en el camino, recordemos que somos los arquitectos de un sueño que une dos dimensiones: la del mundo real y la del mundo Pokémon. Nuestro deber es fusionar estas realidades, creando un juego que despierte la emoción en los corazones de nuestros jugadores y los transporte a un universo lleno de batallas, aventuras y amistad.

Así que, preparémonos para sumergirnos en la creación de un juego que trascienda las barreras del tiempo y el espacio, donde las criaturas se conviertan en héroes, los bytes en poder y la innovación en magia. ¡Que comience la leyenda de nuestro juego de batallas de Pokémon!

### 1.1 Introducción

El juego se juega entre 2 oponentes que turno a turno se atacan mutuamente hasta que uno alcance la victoria. Cada jugador tiene un máximo de 6 Pokémons que pueden utilizar a lo largo de la partida. El objetivo es

derrotar todos los Pokémons del adversario antes de quedarse sin Pokémons no debilitados.

## 1.2 Batalla

Inicialmente cada jugador elige su Pokémon inicial sin conocer el del adversario. El juego se desarrolla por turnos iniciando por el jugador que tiene en el campo de juego el Pokémon más **veloz**. Cada jugador en su turno puede hacer **una** de las siguientes acciones:

- Usar habilidad: Elegir una de las 4 habilidades del Pokémon. Ver más en la sección sobre habilidades.
- Usar un ítem: Elegir un ítem. Ver más en la sección sobre ítems.
- Cambiar el Pokémon: Elegir otro Pokémon que tenga disponible, osea que tenga vida aún y ponerlo en el campo de juego
- Rendirse: Se da por ganador al otro jugador.

En caso de que un Pokémon se quede sin vida disponible, el jugador tiene que remplazarlo inmediatamente, perdiendo su turno.

## 1.3 Pokémons

Cada Pokémon tiene un nombre, una historia, un nivel (número entre 1 y 100) y un tipo. Además, tienen un valor de velocidad, un valor de ataque y un valor de defensa. Por último, tienen una vida máxima que es su número de vida inicial. Por ejemplo:

- Nombre: Bulbasur
- Nivel: 5
- Tipo: Planta
- Historia: Este Pokémon nace con una semilla en el lomo, que brota con el paso del tiempo.
- Vida máxima: 100
- Velocidad: 20
- Defensa: 3
- Ataque: 10

## 1.4 Tipos

El tipo de un Pokémon puede favorecerlo o no en la batalla. Los tipos disponibles son:

- Agua
- Bicho
- Dragón
- Eléctrico
- Fantasma
- Fuego
- Hielo
- Lucha
- Normal
- Planta
- Psíquico
- Roca
- Tierra
- Veneno
- Volador

## 1.5 Habilidades

A su vez, los Pokémons pueden tener hasta 4 habilidades. Hay tres tipos de habilidades:

- De ataque: Estas habilidades tienen un tipo (que puede no coincidir con el tipo del Pokémon), un poder de ataque y una cantidad máxima de oportunidades que puede ser utilizada en batalla. Por ejemplo: La habilidad llamarada es de tipo fuego, tiene un poder de ataque de 50 y puede ser utilizada 10 veces.

- De modificación de estadística: Estas habilidades lo que hacen es restaurar parte de la vida del Pokémon o aumentar/disminuir alguna de las estadísticas del adversario o de sí mismo. Por ejemplo disminuir la defensa del Pokémon rival.
- De modificación de estado: Estas habilidades modifican el estado del Pokémon rival. En caso de que el otro Pokémon ya tenga un estado entonces esta habilidad no realizará ningún cambio.

Las habilidades pueden ser utilizadas un número finito de veces por partida. Cada habilidad se define por su nombre, tipo de habilidad, el poder y la cantidad de veces que puede ser utilizada.

## 1.6 Cálculo de ataque

Para calcular la cantidad de daño que genera una habilidad de ataque se utilizará la siguiente fórmula:

$$damage = \left( \frac{2 * nivel * critico * poder * A / D}{50} + 2 \right) * mismoTipo * tipo * random$$

Donde:

- nivel: Es el nivel del Pokémon atacante
- critico: 2 si es crítico y 1 si no lo es. La probabilidad de que un ataque sea crítico es 0.9
- poder: Es el poder de la habilidad utilizada
- A: Es la estadística ataque del Pokémon atacante
- D: Es la estadística defensa del Pokémon que es atacado
- mismoTipo: 1.5 si el tipo de la habilidad coincide con el tipo del Pokémon atacante, 1 si no hay coincidencia.
- tipo: Es la efectividad del tipo de la habilidad utilizada contra el tipo del Pokémon atacado. La efectividad se puede ver en la siguiente tabla:

Efectividad		Tipo del Pokémon del oponente														
		AGUA	BICHO	DRAGÓN	ELÉCTRICO	FANTASMA	FUEGO	HIELO	LUCHA	NORMAL	PLANTA	PSÍQUICO	ROCA	TIERRA	VENENO	VOLADOR
TIPO DEL ATAQUE PROPIO	AGUA	1/2		1/2			x2				1/2		x2	x2		
	BICHO					1/2	1/2		1/2		x2	x2			x2	1/2
	DRAGÓN			x2												
	ELÉCTRICO	x2		1/2	1/2						1/2			0		x2
	FANTASMA					x2				0		0				
	FUEGO	1/2	x2	1/2			1/2	x2			x2		1/2			
	HIELO	1/2		x2				1/2			x2			x2		x2
	LUCHA		1/2			0		x2		x2		1/2	x2		1/2	1/2
	NORMAL					0							1/2			
	PLANTA	x2	1/2	1/2			1/2				1/2		x2	x2	1/2	1/2
	PSÍQUICO								x2			1/2			x2	
	ROCA		x2				x2	x2	1/2					1/2		x2
	TIERRA		1/2		x2		x2				1/2		x2		x2	0
	VENENO		x2			1/2					x2		1/2	1/2	1/2	
	VOLADOR	x2		1/2					x2		x2		1/2			

Figure 1: Tabla de efectividad de cada tipo

- random: random se realiza como una multiplicación por un número entero aleatorio uniformemente distribuido entre 217 y 255 (inclusive), seguido de una división entera por 255.

## 1.7 Estado

Las habilidades de modificación de estado, generan un cambio de estado en el Pokémon rival. Estados posibles:

- Dormido: El Pokémon no puede atacar mientras este dormido. El jugador aún tiene que seleccionar una habilidad para usar. El Pokémon puede despertarse en cada turno con probabilidad  $0.25 + N * 0.25$  siendo N la cantidad de turnos que ya perdió estando dormido. Al cuarto se despertaría seguro.
- Paralizado: El Pokémon no realizará la habilidad seleccionada con probabilidad 0.5.
- Envenenado: El Pokémon pierde un 5% (de la cantidad máxima de vida) de vida luego de cada turno.

## 1.8 Items

Hay distintos tipos de items:

- De modificación de estadística: Estos items lo que hacen es aumentar alguna de las estadísticas de un Pokémon propio. Los items de este tipo son:
  - Ataque X: aumenta un 10% el ataque del Pokémon en el campo de batalla.
  - Defensa X: aumenta un 10% la defensa del Pokémon en el campo de batalla.
- De restauración de vida: Cura un Pokémon propio. No se pueden curar Pokémons sin vida disponible. Los items disponibles son: Poción (cura 20), Mega Poción (cura 50) e Hiper Poción (cura 100)
- CuraTodo: Se utiliza para sacar al Pokémon del estado actual. Por ejemplo lo desenvenena. No se puede curar un Pokémon que no tiene ningún estado.
- Revivir: Revive a un Pokémon debilitado (obviamente no puede ser aplicado al Pokémon en el campo de juego)
- Se permite que el grupo invente sus propias categorías y agregue items a las categorías ya existentes

Aclaración: Cada item se puede usar una vez pero se permite al jugador tener mas de una unidad de un item. Por ejemplo es valido que un jugador disponga de 3 pociones.

## 2 Desarrollo

El sistema, inicialmente, tendrá como única interfaz con el usuario la consola. El juego preguntará por los nombres de los jugadores. Los nombres no pueden tener mas de 50 caracteres. Cada jugador tendrá que seleccionar su Pokémon inicial, y en base al Pokémon más veloz se definirá quien comienza. Luego el juego se administrará por turnos, alternando entre los dos jugadores.

En cada turno el jugador podrá realizar algunas de las siguientes opciones:

- Rendirse: El programa imprime un mensaje felicitando al ganador utilizando el nombre. Al apretar cualquier tecla el programa termina.

- Ver el campo de batalla. Se muestra el Pokémon del rival y el propio, mostrando nombre, nivel, estado (en caso que no sea el normal, por ejemplo dormido) y la vida restante.
- Intercambiar Pokémon. Muestra los Pokémon del jugador. Al seleccionar uno cambia el Pokémon del campo de juego por el seleccionado. Si bien se muestran todos los Pokémon del jugador, solamente se pueden seleccionar un Pokémon que tenga vida restante, en caso de seleccionar uno sin vida mostrar mensaje de error. Debe haber una opción de cancelar operación y volver al menú de acciones. Luego de cambiar el Pokémon del campo de batalla se termina el turno, se debe mostrar un mensaje que muestre la acción realizada para el otro jugador.
- Aplicar item. Se muestra la lista de items disponibles con su nombre y cantidad restante. Si el usuario selecciona un item, se debe mostrar los Pokémon del usuario para que se seleccione el Pokémon al aplicar el item (se pueden aplicar items a Pokémons que no están en el campo de juego). Si el item no se puede aplicar al Pokémon seleccionado por que no generaria ningun cambio (no se puede curar Pokémon que tienen toda su vida disponible por ejemplo) se debe mostrar un mensaje de error. El usuario tiene que tener una forma de cancelar la operación. Luego de aplicar el item, se debe mostrar un mensaje que muestre la acción realizada para el otro jugador.
- Ataque. Se muestra los ataques del Pokémon con su nombre y la cantidad de veces restantes que se puede utilizar. El usuario debe poder cancelar la acción. Una vez seleccionada, se debe mostrar un mensaje mostrando la acción realizada y el resultado. El resultado depende de la acción, por ejemplo si cambió la vida, mostrar la vida restante, si el estado cambió mostrar un mensaje acorde. Luego de esta acción se finaliza el turno

En caso que un jugador se quede sin Pokémon con vida, el sistema debe comportarse como si se hubiera rendido.

Se puede ver un ejemplo de la dinámica en el siguiente video: <https://www.youtube.com/watch?v=PiCaWmrS4U4>

### 3 Implementación

Los mensajes de información y de error quedan para ser definidas por los alumnos. Se recomienda utilizar una libreria como JLine(<https://github.com>.



[com/jline/jline3](#)) para manejar el input del usuario y mostrar listas de opciones para limitar la cantidad de errores posibles. Como por ejemplo:

Listing 1: Python example

1. A
2. B
3. C

Ingrese numero de la opcion deseada: 1

Seleccionado: A

Recomendación: Tener un modulo que ayude con este tipo de opciones que reciba una lista de opciones y retorna la seleccionada para tener la impresión por consola en un solo lugar.

## 4 Datos del sistema

Naturalmente, la partida constará de dos sets de Pokémons y dos sets de items (uno por jugador). Para este primer TP, los datos deberán estar incluidos dentro del código donde parezca mas adecuado. Recomendación: Tener una clase que provea estos datos en lugar de tenerlos distribuidos por todos lados, principalmente por lo que luego este punto cambiará y será mas difícil de adaptar si no se sigue la recomendación.

## Informe

Se debe realizar un informe que contenga **al menos**:

- Caratula con el nombre de la cátedra, titulo de la entrega e integrantes.
- Aclaraciones e hipótesis necesarias sobre el enunciado.
- Diagrama de clases completo siguiendo las prácticas vistas en UML.
- Explicación y justificación de las decisiones tomadas para el diseño.
- Conclusiones

## Formato de entrega

Para realizar la entrega se debe mandar un mail a [algoritmos3.fiuba@gmail.com](mailto:algoritmos3.fiuba@gmail.com) indicando los integrantes del grupo adjuntando el informe e indicando cual es

el branch y el repositorio de la entrega. El repositorio tiene que ser privado. El trabajo práctico debe realizarse utilizando Git y Github. Para realizar la entrega se debe crear un branch llamado "tp-x" (en este caso tp-1) y no se debe modificar mas luego de enviado el mail. Se recomienda utilizar un branch común de trabajo, por ejemplo "main", y crear el branch una vez que se decida hacer la entrega. Si se quiere seguir trabajando en próximas entregas, continua el branch común de trabajo. Se debe dar permisos en el repositorio a los 3 docentes del curso. Se debe contar con un README que explique como correr el programa "desde cero", como instalar las dependencias y como correr el juego.

## **Fecha de entrega**

La fecha de entrega es 06/10/2023.

## **Criterios de aprobación**

Para aprobar el trabajo práctico se deben cumplir con los objetivos mencionados previamente y ademas cumplir con todos los puntos mencionados en el enunciado.