

7507 Algoritmos y programacion III

# Trabajo Práctico

**Integrantes:**

Alumno	padron
Martin, Mariano Gabriel	97768
Blanco, Sebastian Ezequiel	98539
Marinero, Santiago Jose	97969
Perrone, Patricio Nahuel	98230

**Año:** 2016

**Cuatrimestre:** 2

**Fecha de entrega:** 14 de Diciembre

**GitHub:** <https://github.com/BlancoSebastianEzequiel/Algomon>

# Índice

<b>1. Supuestos</b>	<b>1</b>
<b>2. Modelo del dominio</b>	<b>2</b>
2.1. Entidades del modelo	2
<b>3. Diagramas de clase</b>	<b>3</b>
3.1. Clase Pokemon	3
3.2. Clase Ataque	3
3.3. Clase Elemento	4
3.4. Clase Ataque	4
3.5. Clase Entrenador	4
3.6. Clase Modo	5
3.7. Clase Juego	6
<b>4. Detalles de implementacion</b>	<b>7</b>
<b>5. Escepciones</b>	<b>8</b>

# 1. Supuestos

1. Cuando un ataque cuya potencia es impar genera daño reducido (por haber impactado contra un Pokémon resistente al tipo del ataque) se reduce de los puntos de vida del enemigo la mitad de la potencia del ataque redondeando hacia abajo (ya que no le vemos sentido a un ataque que, por ejemplo, quite 3.5 puntos de vida). De esta manera, un ataque que normalmente quita un punto de vida pasará a quitar 0 puntos de vida si impacta contra un Pokémon resistente al tipo del ataque.
2. Cuando un Pokémon pierde sus puntos de vida, éste podrá permanecer en el campo. Pero ya no será capaz de provocarle daño alguno al oponente con sus ataques, ni tampoco podrá sufrir más daño de los ataques enemigos.

## 2. Modelo del dominio

### 2.1. Entidades del modelo

- **Pokemon:** modela a un Pokémon genérico capaz de atacar y recibir ataques.
- **Subtipos de Pokemon:** son clases que heredan de Pokémon y modelan a los distintos tipos que plantea la consigna: Fuego, Agua, Hierba y Normal. Cada subtipo tiene redefinidos los métodos recibirAtaque correspondientes para lograr representar la ventaja de tipos
- **Pokemones concretos:** representan a los Pokémon que el usuario final de la aplicación podrá utilizar para pelear
- **Ataque:** modelan las técnicas de combate de los Pokémon. Se ejecutan con el método correspondiente y le mandan un mensaje al objetivo para que decremente sus puntos de vida según corresponda.
- **Subtipos de ataque:** modelan los distintos tipos de ataques que contempla la consigna: Fuego, Agua, Hierba y Normal.
- **Ataques concretos:** modelan los ataques que el usuario final podrá ordenar realizar a los Pokémon.
- **Entrenador:** modela al jugador que participa de la batalla. Tiene un conjunto de Pokémon y de elementos a su disposición. Posee además los métodos para hacer uso de dichos elementos, darle órdenes de ataque a sus Pokémon o hacer cambios dentro del equipo.
- **Elemento:** esta clase modela a los elementos que un entrenador puede usar durante la batalla. Tienen diversos efectos como recuperar puntos de vida o curar al Pokémon de estados adversos.
- **Modo:** esta clase modela el estado en que se encuentra un Pokémon. Heredan de esta clase, entre otras, las clases ModoDañoPermanente y ModoInhabilitadoPor1Turno, las cuales se encargan de modelar estados adversos. En el caso de ModoDañoPermanente, por ejemplo, el comportamiento del Pokémon es modificado de manera que, en cada turno, pierde una cantidad de puntos de vida equivalente al 10 % de sus puntos de vida originales.
- **Juego:** esta clase es la encargada de interactuar con la interfaz gráfica recibiendo las órdenes que ésta le pase y ejecutándolas en el modelo.

### 3. Diagramas de clase

A continuación presentamos algunos diagramas de clase que muestran on algo más de detalle las entidades anteriormente descriptas

#### 3.1. Clase Pokemon

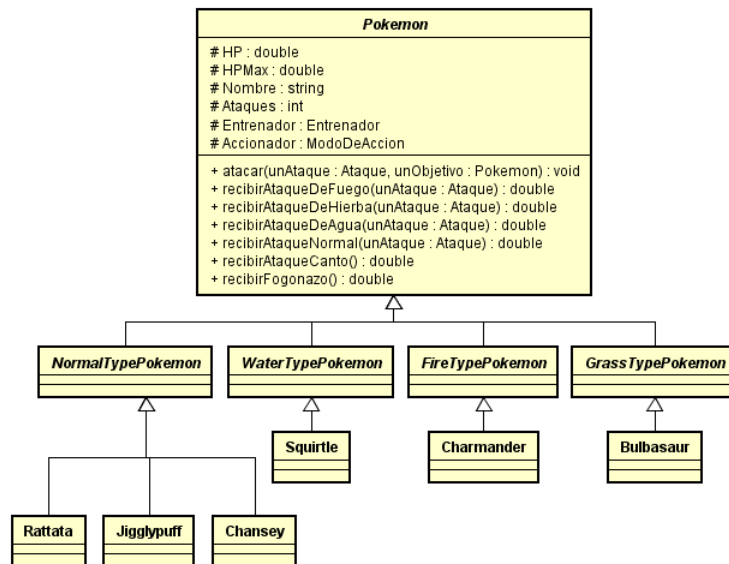


Figura 1: Pokemon

#### 3.2. Clase Ataque

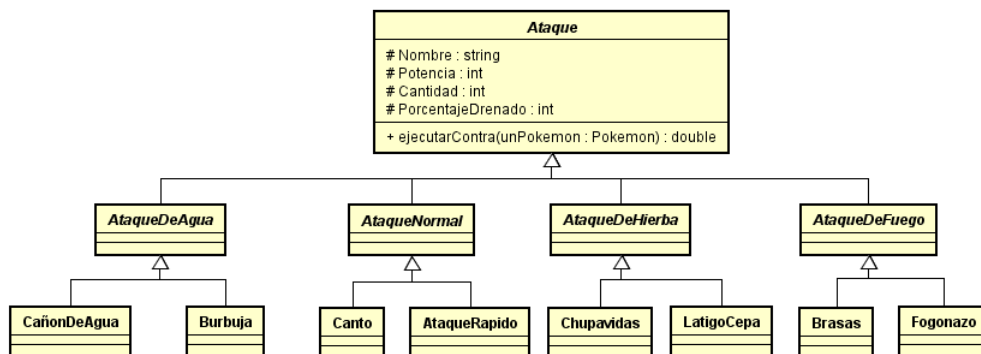


Figura 2: Ataque

### 3.3. Clase Elemento

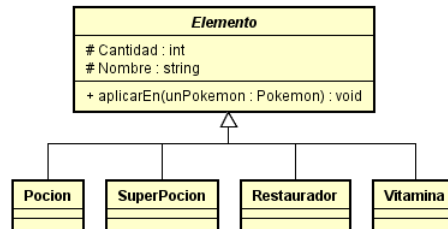


Figura 3: Elemento

### 3.4. Clase Ataque

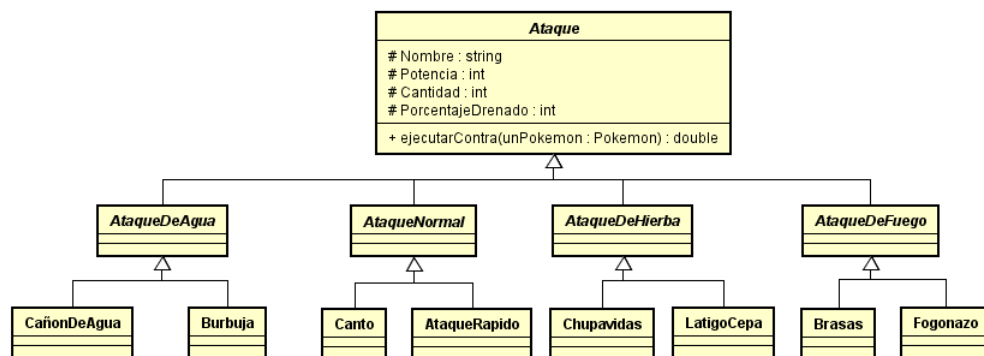


Figura 4: Ataque

### 3.5. Clase Entrenador

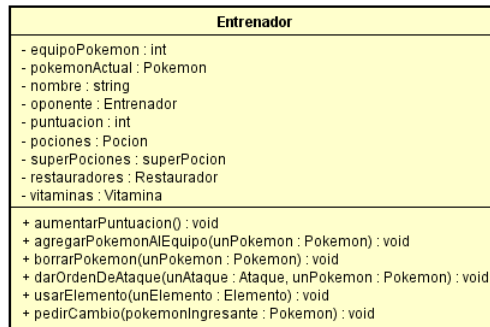


Figura 5: Entrenador

### 3.6. Clase Modo

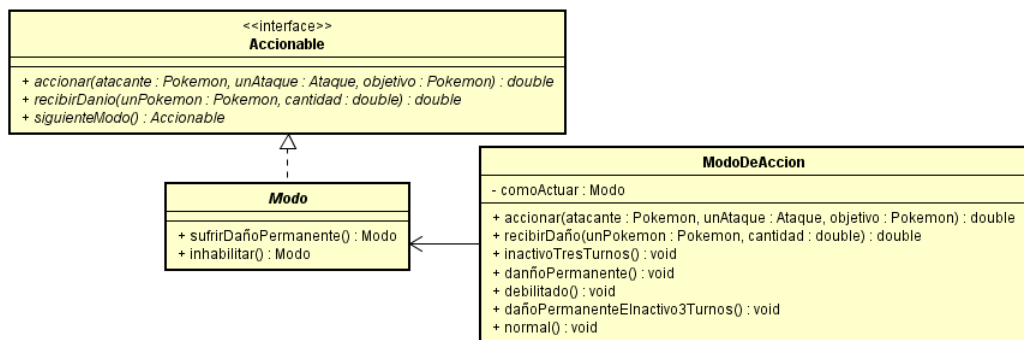


Figura 6: Modo

Son herederas de Modo las clases:

- ModoInhabilitadoPor1Turno
- ModoInhabilitadoPor2Turnos
- ModoInhabilitadoPor3Turnos
- ModoDañoPermanente
- ModoDañoPermanenteEInhabilitadoPor1Turno
- ModoDañoPermanenteEInhabilitadoPor2Turnos
- ModoDañoPermanenteEInhabilitadoPor3Turnos
- ModoDebilitado

### 3.7. Clase Juego

Juego
- EntrenadorActual : Entrenador
+ comenzarBatalla(unEntrenador : Entrenador, otroEntrenador : Entrenador) : void + cambioDeTurno() : void + atacar(unAtaque : Ataque, unPokemon : Pokemon) : void + usarElemento(unElemento : Elemento) : void + cambiarPokemon(nombrePokemon : string) : void

Figura 7: Juego



## 4. Detalles de implementacion

Implementamos los métodos de un Pokémon genérico aplicando el patrón double dispatch. Todas las instancias de la clase Pokémon tienen métodos para recibir ataques de diversos tipos pero, dependiendo de la subclase a la que pertenezca cada instancia, el efecto de los ataques que se reciban varía.

A su vez, todos los Pokémon pueden lanzar ataques (recibiendo previamente una instancia del ataque en cuestión y un objetivo como parámetro). Dependiendo del subtipo de la clase `Ataque` al que pertenezca la instancia recibida, ejecutar el método atacar resultará en enviarle un mensaje distinto al enemigo. Atacar con `Brasas`, por ejemplo, resultará en enviarle el mensaje `recibirAtaqueDeFuego()` al enemigo. Por ejemplo, Squirtle y Charmander tienen el método `recibirAtaqueDeHierba()` al igual que todo Pokémon.

Pero `Squirtle` pertenece además al subtipo `WaterTypePokemon`, por lo que `recibirAtaqueDeHierba()` le debitará a sus puntos de vida el doble de la potencia del ataque recibido. Por otra parte `Charmander` pertenece al subtipo `FireTypePokemon`, lo que significa que tiene el método `recibirAtaqueDeHierba()` implementado de manera diferente. En cuestión, lo tiene implementado para debitar de sus puntos de vida solamente la mitad de la potencia del ataque recibido. De esta manera, siguiendo este patrón podemos modelar el sistema de ventaja de tipos entre Pokémon.

Los Pokémon también pueden sufrir estados adversos que modifiquen su comportamiento durante la batalla. Para eso tienen un atributo accionador que lleva dentro una instancia de Modo. De esta manera, algunos de los ataques que reciban pueden cambiar el Modo que llevan por Inhabilitado o quemado por ejemplo.

Estos Pokémon serán comandados luego por un entrenador que llevará tres instancias dentro de su atributo equipoPokémon.

Por encima de los entrenadores está la clase juego, que lleva como atributo dos instancias de entrenador, es decir, a los dos jugadores.

El juego recibirá las órdenes del jugador a través de la interfaz gráfica y le ordenará al entrenador del turno actual ejecutar dicha acción. Entre una acción y la otra el juego irá cambiando los turnos para que cada jugador pueda jugar.

## 5. Excepciones

Las excepciones que contempla el modelo son las siguientes:

- **NoPuedeRealizarElAtaqueException:** Esta excepción se lanza cuando un Pokémon intenta efectuar un ataque que ya consumió por completo. La excepción es atrapada por el método atacar del mismo Pokémon y, de esta manera, conseguimos que un ataque no se puede usar más veces de las contempladas por la consigna.
- **PokemonSeDebilitoException:** Esta excepción es lanzada cuando, producto de un ataque, un Pokémon se queda sin puntos de vida. La excepción es lanzada y asciende hasta llegar al entrenador que dio la orden de ataque que resultó en el debilitamiento de dicho Pokémon. El entrenador atrapa la excepción y se suma un punto a su puntaje (pues derrotó a uno de los Pokémon del enemigo). Esta excepción se lanza en otros ámbitos también como, por ejemplo, cuando un entrenador intenta cambiar a su Pokémon actual por otro de su equipo que ya no tiene puntos de vida. En este caso la misma interfaz se encarga de atrapar la excepción y realizar las acciones que correspondan.
- **EquipoCompletoException:** Esta excepción se lanza cuando un entrenador intenta agregar a su equipo de tres Pokémon un cuarto. Esta excepción es atrapada luego por la interfaz que se encarga de realizar las acciones que correspondan.
- **GanadorException:** Cuando un entrenador se anota un tercer punto a su puntaje, significa que derrotó a los tres Pokémon del enemigo. Entonces lanza esta excepción que luego la interfaz captura declarando ganador al entrenador que lanzó esta excepción.