

# Manual de Informações e Instruções Pokemon v1.0

Adriano Henrique Rezende, Leonardo Deganello de Souza<sup>1</sup>

<sup>1</sup>RA70114, RA98995

## Requisitos

1. Java 9 - setado como default
2. Terminal
3. Banco de Dados Mysql com usuário root sem senha
4. Arquivos 'ataques.csv', 'Pokemons.csv', 'multiplicadoresAtaque.csv' e 'mysql.jar' na raiz do projeto
5. pasta GUI para a saída dos arquivos compilados
6. arquivo run.sh na raiz do projeto

## Compilando e executando

Para compilar e executar basta rodar o arquivo run.sh

## Orientação a Objetos

### Encapsulamento

Encapsulamento serve para controlar o acesso aos atributos e métodos de uma classe. Isso é feito através da classe Pokemon, em que seus atributos são declarados como private e para se ter acesso a algum atributo é necessário que se use o get ou setter respectivo do atributo caso contrário não será possível obter o dado armazenado naquele atributo para aquela instância.

### Herança

Herança é um mecanismo da Orientação a Objeto que permite criar novas classes a partir de classes já existentes, aproveitando-se das características existentes na classe a ser estendida. Isso é feito através da classe abstrata Ataque, em que ela é usada como base para cada tipo de ataque, tendo como o único método abstrato o Efeito, sendo esse implementado em cada classe que estende a class Ataque.

### Polimorfismo

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação, assinatura, mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. Isso é feito através da classe Ataque e suas subclasses em que uma instância da classe ataque pode ser usada para referenciar o método efeito implementado em cada uma de suas classes filhas.

## **Decisões de Projeto**

### **Classe de Controle do Projeto**

As classes que são responsáveis por fazer o controle do fluxo do programa são: a classe InitialScenario responsável por inicializar a interface gráfica e possibilitar a escolha de que tipo de jogador vai controlar cada time. A classe TimeScenario responsável pela criação do time com os respectivos pokemons e ataques. E por último a classe BattleScenario responsável pela execução da batalha em si.

### **Interação com o Usuário**

A interação com usuário ocorre através da interface gráfica criada utilizando JavaFx.

### **Tratamento de Exceções**

Foram feitas tratamento de exceções com relação a conexão do banco de dados para caso a conexão não seja possível ou o usuário/senha não esteja correto ou não seja encontrada a biblioteca para conexão. E além disso foram feitos tratamentos de exceções nas ações que o usuário pode interagir com a interface e por algum motivo gerar um erro é sempre exibido uma mensagem para o usuário informando que algo que ele realizou não foi uma ação válida.

### **Organização do Programa**

O programa foi organizado em 4 pacotes: gui, database, player e poke. Dentre esses, o pacote gui é o principal e responsável pela execução da interface gráfica, sendo ele subdividido em outros pacotes para o controle do fluxo da interface gráfica.

O pacote database, como o nome diz, é responsável pela implementação da conexão com banco de dados além de realizar algumas consultas.

O pacote player é responsável pela implementação do jogador.

E por último, o modulo poke é responsável pela implementação dos pokemons, especies e os ataques disponiveis no programa.