

**MAC0242 - Laboratório de Programação II**  
**Prof. Dr. Alfredo Goldman**

**Projeto da disciplina - Etapa 1**

**Visão geral**

Nosso projeto da disciplina será a criação de um jogo de batalha, estilo Pokémon, utilizando **Python 3** com orientação a objetos para isso. Ele será composto de três etapas:

1. Modelagem da batalha em modo texto
2. Comunicação ponto a ponto via rede para batalhas
3. Inteligência artificial para batalhas

O objetivo é ao fim do semestre organizar um campeonato entre os grupos. Então desde o começo pensem na melhor modelagem possível para maximizar o aproveitamento de código entre as fases.

Em todas as etapas será esperada a manutenção de **diagramas de classe** descrevendo a modelagem do seu programa, bem como uma **cobertura de testes** completa.

**Tecnologias**

Obrigatoriamente, os programas devem ser desenvolvidos com Python 3. O uso de bibliotecas já existentes do Python é livre, mas devem ser claramente indicadas em um arquivo *requirements.txt* ([https://pip.pypa.io/en/latest/user\\_guide.html#requirements-files](https://pip.pypa.io/en/latest/user_guide.html#requirements-files)).

Também é recomendado o uso do pacote virtualenv (<http://virtualenv.readthedocs.org/en/latest/virtualenv.html#usage>) para isolar essas dependências.

## Etapa 1

Como dito antes, nesta etapa deve ser implementada uma batalha em modo texto entre dois pokémons. É nesta fase onde boa parte dos conceitos fundamentais será definida:

- Pokémon
  - Nome
  - Nível (LVL)
  - Atributos (valores inteiros de 0 a 255)<sup>1</sup>
    - Pontos de vida (HP)
    - Ataque base (ATK)
    - Defesa base (DEF)
    - Velocidade base (SPD)
    - Especial base (SPC)
  - Tipos 1 e 2<sup>2</sup> (entre)
    - 0: Normal
    - 1: Fighting
    - 2: Flying
    - 3: Poison
    - 4: Ground
    - 5: Rock
    - 6: Bird
    - 7: Bug
    - 8: Ghost
    - 9: Fire
    - 10: Water
    - 11: Grass
    - 12: Electric
    - 13: Psychic
    - 14: Ice
    - 15: Dragon
    - 16: Blank

---

<sup>1</sup> [http://bulbapedia.bulbagarden.net/wiki/Pok%C3%A9mon\\_base\\_stats\\_data\\_structure\\_in\\_Generation\\_I](http://bulbapedia.bulbagarden.net/wiki/Pok%C3%A9mon_base_stats_data_structure_in_Generation_I)

<sup>2</sup> O segundo tipo é opcional, nesse caso use o tipo *blank*

- Até quatro ataques (com atributos)<sup>3</sup>
  - Nome
  - Tipo (TYP)
  - Acurácia (ACU)
  - Poder (PWR)
  - Pontos (PP)

**Nota importante sobre os ataques:** Por simplicidade, neste projeto implementaremos ataques que causam apenas dano direto ao outro pokémon. Então não devem ser considerados ataques que modifiquem atributos e status (como *Tail Whip* e *Sleep Powder*), ataques que consumam dois turnos (como *Fly*). Não se esqueçam de implementar *struggle*<sup>4</sup>.

Finalmente, a batalha entre dois pokémons segue a seguinte dinâmica:

1. Pokémon com o maior atributo de velocidade base inicia o seu turno
2. São exibidos o nome, HP e indicação se está em seu turno para os dois pokémons
3. É exibida a lista de ataques deste pokémon com um número, nome e PP de cada ataque

- 1 - Ember (13/20)
- 2 - Tackle (20/20)
- 3 - Quick attack (0/20)
- 4 - Body slam (10/10)

4. O usuário entra com número do ataque que deseja usar pela entrada padrão.
5. O dano deste ataque é calculado de acordo com a fórmula em [http://bulbapedia.bulbagarden.net/wiki/Damage#Damage\\_formula](http://bulbapedia.bulbagarden.net/wiki/Damage#Damage_formula), com o dano sendo subtraído do HP do outro pokémon
6. O turno deste pokémon acaba e passa ao turno do outro pokémon
7. Repita 2 a 6 até que o HP de algum pokémon seja menor ou igual a zero

Assim, resta apenas definirmos de onde virão as informações sobre os pokémons. Ela se dará de forma linear pela entrada padrão, seguindo a ordem dos atributos acima para cada pokémon. O formato deve ser o seguinte para cada pokémon:

---

<sup>3</sup> [http://bulbapedia.bulbagarden.net/wiki/List\\_of\\_moves#Generation\\_I](http://bulbapedia.bulbagarden.net/wiki/List_of_moves#Generation_I)

<sup>4</sup> [http://bulbapedia.bulbagarden.net/wiki/Struggle\\_%28move%29#Generation\\_I](http://bulbapedia.bulbagarden.net/wiki/Struggle_%28move%29#Generation_I)

<NOME>  
<LVL>  
<HP>  
<ATK>  
<DEF>  
<SPD>  
<SPC>  
<TYP1>  
<TYP2>  
<NUM ATKS>  
<NOME>  
<TYP>  
<ACU>  
<PWR>  
<PP>  
...  
<NOME>  
<TYP>  
<ACU>  
<PWR>  
<PP>

**Curiosidade:** A geração I de pokémon (Red e Blue) foi desenvolvida para rodar em um hardware bastante modesto<sup>5</sup> que, entre outras características, estão meros 8Kb de memória. Isso obrigou os desenvolvedores originais a pensarem em diversas soluções um tanto quanto originais e interessantes. Nesse projeto não haverá restrição de uso de memória, mas tentem pensar em como seria ter apenas 8Kb de memória disponíveis. Será um belo exercício.

---

<sup>5</sup> [http://en.wikipedia.org/wiki/Game\\_Boy#Technical\\_specifications](http://en.wikipedia.org/wiki/Game_Boy#Technical_specifications)

## O que deve ser entregue

- Todo o código fonte compactado em um arquivo **zip** ou **tar.gz**, inclusive testes de unidade que devem cobrir o código implementado
- Diagrama de classes descrevendo a implementação nos formatos **pdf** ou **png**
- Relatório
  - Nome e número USP de todos os integrantes do grupo
  - Descrição das dificuldades e desafios

## Importante

- O trabalho é estritamente em grupos de no máximo 4 pessoas. Discussões entre os grupos serão mais do que bem vindas, mas plágio ou cola não serão tolerados. Veja a [política do Departamento de Ciência da Computação para casos de plágio ou cola](#).
- Escreva de forma clara e estruturada todos os seus códigos e relatórios. Organize e nomeie todos os arquivos entregues de forma que eles possam ser facilmente identificados. A avaliação levará em conta todas essas questões! Uma apresentação ruim, ou a falta de clareza, poderá prejudicar sua nota.
- O programa deve ser entregue por meio do sistema [Paca](#) em um arquivo zip ou tar.gz
- Apenas **um** integrante do grupo deve fazer a entrega
- Enquanto o prazo de entrega não expirar, você poderá entregar várias versões do mesmo exercício-programa. Apenas a última versão entregue será guardada pelo sistema. Encerrado o prazo, você perderá 1 ponto para cada hora de atraso. Não deixe para entregar seu exercício na última hora!
- Guarde uma cópia do seu exercício-programa pelo menos até o final do semestre.
- Caso existam problemas de relacionamento entre os grupos, o professor e os monitores devem ser comunicados prontamente, para que medidas possam ser tomadas a tempo.