

## TRABALHO FINAL

### Implementação do Jogo Caça ao Tesouro

#### Objetivo

Desenvolver uma aplicação em Java que simule o jogo Caça ao Tesouro, jogado via terminal. A aplicação deverá permitir que o jogador explore coordenadas em uma ilha misteriosa e receba feedback sobre suas escavações, incluindo se encontrou tesouros, armadilhas ou apenas areia.

#### Descrição

Você deverá implementar o jogo Caça ao Tesouro em um mapa 8x8 que representa uma ilha misteriosa contendo 8 tesouros e 5 armadilhas posicionados aleatoriamente. O jogador poderá escavar posições no mapa até que todos os tesouros sejam encontrados ou tenha completado 25 tentativas de escavação. O jogo deve ser jogado via terminal e deve fornecer feedback ao jogador após cada escavação.

#### Requisitos

##### 1. Estrutura do Projeto:

- Utilize a linguagem Java somente com o conteúdo aprendido em sala de aula (unidade 1 até 7, matrizes e classe Random (dica utilizar a classe Random para posicionar os elementos no mapa)).
- O código deve ser bem documentado e seguir boas práticas de programação.

##### 2. Inicialização do Mapa:

- Crie um mapa 8x8 representado por uma matriz bidimensional de caracteres.
- Preencha o mapa com areia ('~') inicialmente.

Figura 1 - Exemplo de inicialização

0	1	2	3	4
0	~	~	~	~
1	~	~	~	~
2	~	~	~	~
3	~	~	~	~
4	~	~	~	~

##### 3. Posicionamento dos Tesouros e Armadilhas:

- Posicione aleatoriamente oito tesouros ('t') no mapa utilizando a classe Random do Java.
- Posicione aleatoriamente cinco armadilhas ('a') no mapa utilizando a classe Random do Java.
- Certifique-se de que tesouros e armadilhas não se sobreponham.
- As posições dos tesouros e armadilhas não devem ser reveladas para o jogador.

##### 4. Interação com o Jogador:

- Peça ao jogador para inserir coordenadas de escavação (linha e coluna).
- Valide as coordenadas inseridas (devem estar dentro dos limites do mapa).

##### 5. Feedback das Escavações:

- Informe ao jogador se a escavação encontrou um tesouro, uma armadilha ou apenas areia.
- Atualize o mapa visível com:
  - 'T' para tesouros encontrados
  - 'A' para armadilhas encontradas
  - 'O' para areia escavada (posições vazias)
- Mostre o mapa ao jogador após cada escavação, mas sem revelar as posições dos tesouros e armadilhas restantes.
- Mostre ao jogador se é uma posição que ele já escavou.
- Mostre ao jogador se ele inseriu uma posição inválida.
- **Sistema de Pontuação:**
  - +10 pontos por cada tesouro encontrado
  - -5 pontos por cada armadilha encontrada
  - Mostre a pontuação atual após cada jogada

Figura 2 - Exemplo de jogada com feedback

```
Digite as coordenadas para a escavação (linha e coluna, separadas por espaço):  
1 2  
Você errou  
  0 1 2 3 4 5 6 7 8|  
0 ~ ~ ~ ~ ~ ~ ~ ~  
1 ~ ~ 0 ~ ~ ~ ~ ~ ~  
2 ~ ~ ~ ~ ~ ~ ~ ~  
3 ~ ~ ~ ~ ~ ~ ~ ~  
4 ~ ~ ~ ~ ~ ~ ~ ~  
5 ~ ~ ~ ~ ~ ~ ~ ~  
6 ~ ~ ~ ~ ~ ~ ~ ~  
7 ~ ~ ~ ~ ~ ~ ~ ~  
8 ~ ~ ~ ~ ~ ~ ~ ~  
Pontuação atual: 0 pontos
```

##### 6. Condição de Parada:

- O jogo termina quando todos os tesouros forem encontrados (nesse caso ele vence) ou o jogador tenha completado um total de 25 escavações e não tenha encontrado todos os tesouros (nesse caso ele perde).
- Informe ao jogador se ele venceu ou foi derrotado e revele todas as posições dos tesouros e armadilhas.
- Mostre a pontuação final e classifique o desempenho:
  - 70+ pontos: "Explorador Lendário!"
  - 50-69 pontos: "Caçador de Tesouros Experiente!"
  - 30-49 pontos: "Aventureiro Iniciante"
  - Menos de 30 pontos: "Precisa de mais prática na exploração"

## Entregáveis

### 1. Código Fonte:

- Todo o código fonte do projeto em arquivos Java, junto com a documentação.
- O código deve ser entregue via AVA.

### 2. Documento:

- Descrição das funcionalidades implementadas.

### 3. Imagens:

- Capturas de telas demonstrando a funcionalidade do jogo, juntamente com o bloco de código responsável por aquela funcionalidade.
- As imagens devem ser enviadas todas juntas em um documento .PDF com a descrição de cada uma delas.
- Incluir exemplos de:
  - Inicialização do jogo
  - Encontro de tesouro
  - Encontro de armadilha
  - Tentativa de escavar posição já explorada
  - Tela de vitória
  - Tela de derrota com revelação do mapa completo

## Critérios de Avaliação

- **Correção:** o programa deve funcionar corretamente e seguir as regras do jogo.
- **Qualidade do Código:** o código deve ser bem estruturado, legível e seguir boas práticas de programação.
- **Interação com o Usuário:** a interação deve ser intuitiva, com mensagens de erro adequadas quando necessário.
- **Sistema de Pontuação:** implementação correta do sistema de pontuação e classificação do desempenho.
- **Documentação:** A descrição das funcionalidades deve ser completa e clara.
- **Imagens:** As imagens devem representar significativamente as funcionalidades do jogo.
- **Apresentação:** A apresentação para a turma (Documento e utilização do sistema).
- **Orientação à objetos:** Quem fizer a utilização de orientação à objetos na sua implementação terá uma pontuação bônus na nota do trabalho.