

# Programação de Soluções Computacionais

- ENUNCIADO A3 -

Prof. Anderson Domingues ([anderson.domingues@animaeducacao.com.br](mailto:anderson.domingues@animaeducacao.com.br))

**Objetivo:** Exercitar as habilidades e competências desenvolvidas ao longo da UC através da simulação de um projeto de software, com foco na modelagem de algoritmos e utilização da linguagem de programação como ferramentas para a construção de soluções computacionais.

**Grupos de trabalho e avaliação:** Os projetos serão desenvolvidos em grupos de trabalho de 3 a 4 alunos. O tema será o mesmo para todos os grupos, cfe. discutido abaixo. Não serão aceitos trabalhos em outras linguagens senão aquelas trabalhadas em aula. Cada aluno será avaliado individualmente cfe. os critérios abaixo:

- Organização do projeto (2 pontos): Inclui indentação e comentários no código, correta utilização de nomes para as variáveis e outras estruturas de código, separação das múltiplas unidades de código em diferentes arquivos, correta utilização de sistema de versionamento e documentação contendo orientações para compilar/executar o projeto.
- Domínio dos paradigmas de programação estruturado e orientação a objetos (3 pontos): Inclui a correta modelagem e organização do projeto em classes, funções e outras estruturas; a correta decomposição do projeto em múltiplas unidades de código, a utilização correta utilização de estruturas de dados abstratas (ex.: pilha, fila, listas, vetores) e outras abstrações oferecidas pela linguagem de programação e suas bibliotecas padrão. Correta utilização de herança, polimorfismo, encapsulamento (modificadores de acesso) e outros aspectos de OOP. Os alunos precisam demonstrar domínio de herança, polimorfismo e estruturas abstratas de dados.
- Implementação dos requisitos e regras de negócio da aplicação (3 pontos): Correta implementação das regras de negócio da aplicação, conforme instruções abaixo.
- Apresentação do projeto e execução da aplicação (1 ponto): Aqui, são incluídos a execução e apresentação da aplicação para a turma, a ser executado presencialmente pelos membros do grupo e utilizando o projetor como recurso. Os alunos devem demonstrar a organização do projeto e como atenderam cada um dos outros critérios.
- Persistência de dados em arquivos (1 ponto): Tratamento de persistência de dados utilizando arquivos de texto e/ou binários, ou banco de dados.

**Pontos Extras:** Os seguintes itens, se implementados corretamente, garantem um ponto extra cada (total de 2 pontos).

- Interface gráfica (1 ponto EXTRA): Inclui a construção de uma interface gráfica utilizando uma biblioteca de interface gráfica. Não serão aceitas interfaces construídas em ferramentas WYSIWYG.
- Banco de dados (1 ponto EXTRA): Inclui o tratamento da persistência de dados utilizando um SGBD. Este extra substitui a implementação de persistência em arquivos (isto é, você recebe 2 pontos pela implementação com banco de dados).
- GitHub: (1 ponto EXTRA): Inclui a correta configuração do repositório, versionamento e compartilhamento do projeto com o grupo.

**Entregáveis:** Nenhum. O trabalho deverá ser apresentado em aula, com data previamente acordada com o professor. Durante a apresentação, os membros de cada grupo serão questionados acerca da construção do projeto, de seu funcionamento e das tecnologias e técnicas utilizadas.

**Tema:** Os grupos deverão implementar uma releitura do jogo Puzzle Quest! Para mais informações sobre o jogo, consulte a página da Wikipedia ([https://en.wikipedia.org/wiki/Puzzle\\_Quest](https://en.wikipedia.org/wiki/Puzzle_Quest)) e vídeos no Youtube (<https://www.youtube.com/watch?v=0rP17CeNtRs>). Em caso de dúvidas sobre as regras, consulte o professor. Para obtenção da nota máxima no critério *regras de negócio*, os seguintes itens deverão ser implementados.

- Carregamento e salvamento. A implementação apresentada por seu grupo deve conter um menu inicial, onde o usuário poderá escolher entre iniciar uma nova partida ou restaurar o progresso de uma partida anterior. Nesse caso, uma lista de jogos salvos deverá ser mostrada. Ainda, deverá existir uma opção para apagar jogos salvos. Os jogos deverão ser salvos em arquivo ou, caso o grupo queria implementar o extra, em banco de dados.
- Turnos. O grupo deverá implementar a mecânica de turnos. Em outras palavras, o jogo deverá ser "jogável" por dois jogadores. Assim que um jogador executar sua jogada, o próximo jogador deverá receber o controle do tabuleiro. A partida acaba quando os pontos de vida de um dos jogadores chega a zero. Quando o jogador consegue encaixar 4 esferas de uma mesma cor, ele recebe um turno extra.
- Jogada: A cada turno, cada jogador deverá escolher uma célula do tabuleiro e uma direção. Se a direção e a célula escolhida forem válidas, o conteúdo da célula escolhida e da célula adjacente na direção escolhida deverão trocar de lugar. Se uma das duas posições escolhidas estiver fora do tabuleiro, o jogo deverá notificar o erro e permitir que o jogador faça uma nova escolha.
- Pontuação: **PRESTE ATENÇÃO AQUI!** Seu tabuleiro deverá conter esferas de 7 cores distintas. No jogo original, as esferas são (i) caveiras, (ii) elementos coloridos - fogo, gelo, raio ou natureza, (iii) ouro, ou (iv) experiência. Aqui, você deverá implementar as seguintes esferas:
  - Caveiras: possuem o mesmo efeito que no jogo original. Cada caveira reduz a vida do adversário em 1 ponto.
  - Esfera vermelha: Aumenta os pontos de vida do jogador, 1 ponto por esfera.
  - Esfera azuis: Um jogo de esferas azuis transforma todas as esferas vermelhas em caveiras.

- Esfera verde: Transforma todas as caveiras em esferas vermelhas.
- Esfera amarela: Esvazia o ouro do inimigo.
- Ouro: Adiciona 1x ouro. Quando atingir 10x de ouro, você causará dano dobrado no seu próximo turno e seu ouro se tornará zero.
- Experiência (roxa): A cada 10x de experiência, a vida máxima do seu inimigo é permanentemente reduzida em 10 pontos (e sua experiência é zerada).

## Implementação do Tabuleiro

O tabuleiro implementado pelo seu grupo deverá imitar o tabuleiro do jogo mostrado na Figura 1. Porém, se optar por não implementar o ponto extra da interface gráfica, deverá representar o tabuleiro utilizando caracteres ASCII ou UNICODE, como mostra a Figura 2. O tabuleiro deverá possuir 8x8 de tamanho (64 células) e, caso não seja possível executar jogadas, o mesmo deverá ser embaralhado automaticamente (random). Note que, o tabuleiro gerado deve conter jogadas válidas, porém não deve possuir 3x esferas alinhados em linha ou coluna.

**Figura 1:** Tabuleiro do jogo Puzzle Quest.



**Figura 2:** Representação de tabuleiro em texto.

1	S	G	T	P	G	Y	B	C	S - Skull
2	P	G	B	T	T	B	S	T	G - Green
3	G	T	Y	S	S	B	G	R	R - Red
4	Y	G	T	Y	G	R	P	Y	B - Blue
5	R	P	S	T	G	S	G	B	P - Purple
6	B	R	Y	P	R	T	Y	G	T - Treasure (gold)
7	B	T	R	P	G	P	T	Y	Y - Yellow
8	Y	P	S	G	S	B	R	S	

**Outras informações relevantes:**

- Cópias de outros trabalhos (cola) e uso de ChatGPT (e outras IAs) serão consideradas plágio e receberão nota **zero**.
- O trabalho será apresentado em aula, em duas oportunidades. Organizem-se para que o grupo esteja **todo** presente em ao menos uma das aulas. Pontos poderão ser creditados aos alunos que participarem de forma construtiva na apresentação de outros grupos.
- O trabalho deverá ser escrito em Java. **Não** serão aceitos trabalhos em outras linguagens.
- Casos não tratados neste documento devem ser esclarecidos com o professor **em aula e em tempo hábil**.