POLITÉCNICO DE PORTALEGRE

Escola Superior de Tecnologia, Gestão e Design

**DEPARTAMENTO DE TECNOLOGIAS**

***Programação Orientada a Objetos***

**Trabalho 2** – 2024/2025 – 1º Ano – Engenharia Informática

**Data:** 06/maio/2025 – **Docente**: Pedro Miguel da Silva Roque

---

Com o 2º trabalho pretende-se a implementação de um jogo de estratégia com vários tipos de personagens. A implementação deve ser feita em Java e com recurso a técnicas de programação orientada a objetos, herança e polimorfismo.

Este trabalho deve ser desenvolvido individualmente ou em grupos de 2 alunos.

Até ao dia **12 de junho de 2025** deve ser entregue através do **PAE**, um **ficheiro comprimido (.zip)** que contenha o **diagrama de classes** em PDF e todo o **código fonte** implementado para o desenvolvimento do trabalho.

Nas aulas de **13 de junho** decorrerão as **apresentações** dos trabalhos conforme a distribuição dos turnos. A apresentação é **obrigatória** e a não presença na mesma implica a anulação do trabalho, assim como qualquer indício de que o trabalho não tenha sido da autoria de quem o está a apresentar, ou de que o mesmo tenha sido gerado por inteligência artificial.

## Jogo de Estratégia

Implementação de um jogo de estratégia que permita ao utilizador criar e gerir diferentes tipos de unidades militares e simular combates entre estas. Estas unidades podem ser guerreiros, arqueiros ou feiticeiros, e deverão possuir características específicas que influenciam o seu desempenho em combate.

A implementação deve ser feita com recurso a técnicas de POO e utilizando herança e polimorfismo, de forma a maximizar a reutilização de código.

### Funcionalidades a implementar:

- Todas as unidades militares devem ter: nome, custo de produção, pontos de vida, pontos de ataque, pontos de defesa, alcance, posição (coordenadas x e y) e velocidade (valor a aumentar às coordenadas x e/ou y em cada passo da simulação;
- Os guerreiros devem ter também: pontos de blindagem (pontos adicionais de defesa);
- Os arqueiros devem ter também: tipo de armamento (arco e flecha ou besta) com o respetivo aumento de pontos de ataque;
- Os feiticeiros devem ter também: número máximo de feitiços que podem lançar por simulação;
- Todas as unidades militares devem ter o seu método de conversão para string;
- Deve ser possível adicionar, listar e remover unidades militares;

- Deve ser possível simular o combate entre dois exércitos:
    1. O utilizador cria pelo menos duas unidades militares;
    2. O utilizador cria dois exércitos com base nas unidades militares anteriormente criadas e com um limite de custo de produção total (escolhido pelo utilizador) para cada exército;
    3. O utilizador define o tamanho do "tabuleiro" de jogo, através do valor máximo para as coordenadas x e y;
    4. O utilizador posiciona manualmente ou aleatoriamente todas as unidades militares dos 2 exércitos no tabuleiro;
    5. O utilizador executa a simulação indicando o número de passos que deseja executar.
        - Em cada passo da simulação, todas as unidades militares se devem mover aleatoriamente no tabuleiro de acordo com a sua velocidade e interagir (defender e atacar) umas com as outras se terminarem em posições cujo alcance lhes permita interagir;
        - Quando duas (ou mais) unidades militares interagirem, cada uma delas deve perder primeiramente os pontos de defesa e posteriormente os pontos de vida, correspondentes à soma dos pontos de ataque de todas as outras unidades militares inimigas cujo alcance lhes permita atacar;
    6. Após cada passo deve ser listado no terminal o estado de cada unidade militar de cada exército;
    7. A simulação termina quando um dos exércitos já não tiver unidades militares com pontos de vida, ou quando o utilizador não quiser simular mais passos.
- Toda a interação com o utilizador deve ser validada de forma a evitar erros durante a execução do programa. Quando necessário deverá recorrer ao tratamento de exceções;
- O utilizador pode adicionar ou remover unidades militares antes de iniciar uma nova simulação;
- Deve existir uma opção para terminar o programa.

---

**Avaliação do trabalho** – para a avaliação do trabalho vai ser considerada:

- 6 valores – Correta utilização do paradigma da programação orientada a objetos;
- 6 valores – Correta utilização de herança e polimorfismo;
- 1 valor – Funcionamento amigável;
- 0,5 valores – Código bem estruturado;
- 0,5 valores – Código corretamente indentado;
- 1 valor – Comentários adequados;
- 3 – Tratamento de exceções;
- 2 valores - Diagrama de classes.

- **A não comparência na apresentação do trabalho corresponderá à anulação do mesmo, assim como qualquer evidência de partilha de resoluções entre alunos, ou do uso de código gerado por inteligência artificial.**