

## Projeto: Simulador de Batalha de Criaturas Místicas

### Visão Geral

Crie um sistema (incluindo diagrama de classe e sequência) de batalha por turnos entre criaturas místicas, onde cada criatura possui habilidades especiais, tipos elementais e características únicas. Vocês deverão implementar o sistema de batalha, criar testes unitários com JUnit e usar Mockito para simular comportamentos complexos.

### Requisitos do Projeto

#### 1. Sistema de Criaturas Místicas

- Cada criatura tem: nome, pontos de vida (HP), pontos de ataque (ATK), pontos de defesa (DEF) e tipo elemental (Fogo, Água, Terra, Ar, Luz, Trevas)
- Sistema de vantagens/desvantagens baseado em tipos elementais
- Habilidades especiais com efeitos únicos (ex: queimar, congelar, curar)

#### 2. Sistema de Batalha por Turnos

- Sistema de iniciativa baseado em velocidade
- Cálculo de dano considerando vantagem elemental
- Aplicação de efeitos de status entre turnos
- Condições de vitória/derrota

#### 3. Itens e Equipamentos

- Itens que podem ser usados durante a batalha
- Equipamentos que modificam atributos

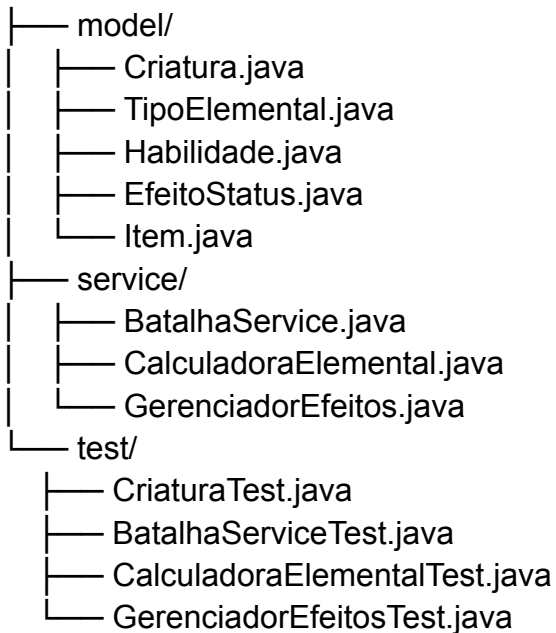
### Desafios de Implementação

1. **Sistema Elemental:** Implementar as relações de vantagem entre os 6 tipos elementais
2. **Habilidades Especiais:** Criar pelo menos 3 habilidades com efeitos únicos cada
3. **Efeitos de Status:** Implementar sistema de estados alterados (queimado, envenenado, etc.)
4. **Sistema de Inventário:** Gerenciar itens usáveis durante a batalha

## Exemplo de Estrutura de Classes (sugestão)

text

src/



## Dicas

1. Usem mocks para simular dependências complexas como banco de dados ou serviços externos
2. Criem testes para casos (valores limites) e cenários excepcionais
3. Documentem os testes com nomes descritivos que expliquem o que está sendo testado

## Próxima aula (dias 25/09 e 02/10): Apresentação para o professor - 1 ponto

**Execução e Análise:** Executar os testes e mostrar quais *bugs* foram encontrados.

1. **Apresentação:** A dupla deve apresentar o projeto ao professor:
  - Uma visão geral do projeto e o entendimento da arquitetura.
  - A lista de cenários de teste que eles criaram.
  - Exemplos de testes **com JUnit e Mockito**, explicando o motivo de cada *mock* e asserção.
  - Os *bugs* que foram identificados e como os testes os revelaram.
  - Ações propostas para corrigir os problemas encontrados no código.
  - Mostrar o **diagrama de classes e sequência** que criaram.  
Explicar quais dependências eles mockaram e **por quê**.  
Debater sobre **o que não faz sentido mockar** (reflexão crítica).

2. **Debate:** Após a apresentação, a dupla deve estar preparada para responder a perguntas e debater sobre as decisões de teste que tomaram.

## **Dia 09/10: Apresentação para a turma - 1 ponto**

---

### **1. Pitch de Apresentação**

Cada dupla terá **5 a 10 minutos** para apresentar seu jogo, como se estivesse em um evento de startups/games. Sejam criativos, organizados, a ideia é vender que seu projeto é o melhor! O pitch deve conter:

#### **1. Nome e Identificação do Jogo**

Criar um nome original e criativo para o jogo.

Elaborar um slogan curto

Criar um logo simples (pode ser feito no Canva ou à mão).

#### **2. Contexto do Jogo**

- Qual é o universo ou história por trás?  
Por que batalhas de criaturas místicas são interessantes?

#### **3. Diferencial**

- O que torna o jogo diferente de outros jogos de batalha?

#### **4. Público-Alvo**

- Quem jogaria esse jogo?  
Jovens? Fãs de RPG? Pessoas que gostam de estratégia?

#### **5. Exemplo de Gameplay**

- Explicar rapidamente como funciona uma batalha (passo a passo do turno).  
Mostrar um print/fluxograma ou log de teste do sistema funcionando.

#### **6. Demonstração Técnica**

- Mostrar que o jogo **foi testado** com JUnit + Mockito.

- Explicar um caso em que o Mockito ajudou a simular algo complexo (ex: congelamento, veneno).

## **6. Considerações Finais**

Falar dos desafios e dos aprendizados ao longo do processo com a experiência