

Projeto "Boss Arena" (QA/Teste)

1. Objetivo

O objetivo deste trabalho é aplicar na prática os conceitos fundamentais de Qualidade (QA) e Teste de Software no ciclo de vida do desenvolvimento de um jogo digital.

Vocês devem projetar, gerenciar e validar uma demo de jogo funcional, utilizando ferramentas e metodologias profissionais de QA para garantir que o produto final atenda a requisitos específicos, desde a documentação (backlog) até os testes formais (unitários e de aceitação).

2. Descrição do Projeto (O Produto)

A equipe deverá desenvolver uma **DEMO jogável** de uma arena de batalha contra um chefe (Boss Fight). O escopo do jogo está estritamente definido pelas seguintes características:

- **Gênero:** Plataforma 2D.
- **Escopo:** Uma única arena onde o jogador enfrenta um Boss.
- **Estilo de Arte:** Pixel art, com restrição de *tiles* e *sprites* em 16x16 pixels e paleta de cores **monocromática** (ex: preto e branco, ou tons de verde, etc.).
- **Jogador:**
 - **Movimentação:** Correr (esquerda/direita), Pular (altura fixa ou variável), Dash (avanço rápido).
 - **Ataque:** O jogador deve ter **uma** forma de ataque (à escolha da equipe):
 - *Melee* (curto alcance, ex: espada) OU
 - *Projétil* (longo alcance, ex: disparo de energia).
- **Sistemas:** O jogo deve possuir um sistema de vida funcional (para o jogador e para o Boss), com feedback visual (ex: barra de vida, *flash* ao tomar dano).

3. Requisitos de Qualidade e Teste (O Processo)

O foco principal desta disciplina é o *processo* de garantia de qualidade. Portanto, além do jogo, os seguintes itens são obrigatórios:

A. Backlog e Critérios de Aceitação (AC)

A equipe deve organizar um **Product Backlog** (ex: Trello, Jira, Asana) detalhando todas as funcionalidades do jogo (ex: "Movimento de Pulo", "Sistema de Vida do Jogador", "IA do Boss - Ataque 1").

Fundamental: Cada item do backlog (seja uma *Feature* ou *User Story*) deve obrigatoriamente conter **Critérios de Aceitação (AC)** claros e testáveis.

Exemplo de Funcionalidade e seus Critérios de Aceitação:

- **Funcionalidade (Story):** "Como jogador, eu quero poder pular para desviar dos ataques do Boss."
- **Critérios de Aceitação (AC):**
 1. **Dado que** o jogador está no chão **E** pressiona o botão [Pulo], **Então** o personagem deve iniciar um movimento vertical para cima.
 2. **Dado que** o jogador está no ar (pulando), **Então** pressionar o botão [Pulo] novamente não deve resultar em um segundo pulo (sem pulo duplo).
 3. **Dado que** o jogador inicia o pulo, **Então** a animação de "pulo" deve ser ativada.
 4. **Dado que** o jogador está em movimento ascendente **E** colide com uma plataforma acima, **Então** seu movimento vertical deve ser interrompido.

B. Gestão de Testes de Aceitação

Todos os Critérios de Aceitação definidos no backlog devem ser traduzidos em **Casos de Teste (Test Cases)** formais.

A equipe deve utilizar uma ferramenta profissional de gestão de testes (ex: **TestLodge**, **TestRail**, **Jira + Zephyr**, etc.) para:

1. Documentar os Casos de Teste (Passos, Resultado Esperado).
2. Executar os testes na demo.
3. Registrar os resultados (Passou, Falhou, Bloqueado).

C. Testes Unitários

O código-fonte do jogo (scripts) deve ser modularizado e conter **Testes Unitários** para as lógicas críticas, independentemente da engine utilizada (Unity, Godot, GameMaker, etc.).

Exemplos de lógicas que devem ser testadas unitariamente:

- Cálculo de dano (ataque do jogador, ataque do boss).
- Lógica do sistema de vida (função `ReceberDano()`).
- Verificação de *cooldown* do Dash ou do Ataque.

4. Entregáveis

O trabalho será entregue em duas partes:

1. Demo do Jogo:

- Build executável do jogo (PC/Mac/Web).
- Link para o repositório do código-fonte (com os testes unitários).
- Link de acesso (leitura) ao Backlog (Trello/Jira).
- Link de acesso (leitura) à ferramenta de Gestão de Testes (TestRail/TestLodge).

2. Apresentação Final

- Tempo: 15-20min.
- Demonstração do gameplay ao vivo.
- Apresentação do Backlog e dos Critérios de Aceitação.
- Demonstração da ferramenta de gestão de testes (TestRail, etc.), mostrando os ciclos de teste e os resultados (relatórios de execução).
- Demonstração da execução dos testes unitários (no console ou na IDE).
- Relatório de bugs encontrados (se houver) e "lições aprendidas" durante o processo de QA

5. Critérios de Avaliação

- **Aderência ao Escopo do Jogo (25%)**: O jogo cumpre todas as especificações (Plataforma 2D, Arte 16x16, Jogador, Boss, Vidas)?
- **Processo de QA - Backlog e AC (25%)**: A qualidade do backlog e a clareza/cobertura dos Critérios de Aceitação.
- **Processo de QA - Testes (30%)**: A qualidade e cobertura dos Testes de Aceitação (na ferramenta) e dos Testes Unitários (no código).
- **Apresentação e Relatórios (20%)**: A clareza da apresentação, a demonstração dos relatórios de teste e a capacidade de explicar o processo de QA.