**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO – UFERSA**

**CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS – CMPF**

**ENGENHARIA DE SOFTWARE**

**DISCENTE: VERA CÉLIA GADELHA**

**DOCENTE: ALYSSON FILGUEIRA MILANEZ**

**DISCIPLINA: GERÊNCIA DE CONFIGURAÇÃO E MUDANÇAS**

**Plano de teste: O Desafio do Saber (Quiz Educativo ODS 4)**

**1. Objetivo dos testes**

O objetivo principal dos testes é garantir que a lógica de negócio do *Quiz Educativo* esteja funcionando conforme os Requisitos Funcionais (RFs) e que as restrições de qualidade (RNFs) sejam atendidas, especialmente a cobertura de código (RNF4).

**- Validar a Funcionalidade Central:** Assegurar que o sistema carrega o conteúdo corretamente, exibe as questões, processa as respostas e calcula a pontuação (RF1 a RF6).

**- Comprovar a Qualidade (RNF4):** Verificar se a lógica de cálculo de pontuação (Quiz.verificarResposta()) está correta e possui alta cobertura de testes de unidade (≥80%).

**- Garantir a Modularidade (RNF2):** Assegurar que as classes (Questao, Quiz, Dados) trabalham de forma independente e correta.

**2. Abordagens de testes utilizados**

As abordagens de teste detalham os métodos que serão aplicados no projeto.

|  |  |  |
| --- | --- | --- |
| **Tipo de Teste** | **O que está sendo verificado** | **Justificativa** |
| **Testes de Unidade** | Lógica de negócio de classes isoladas (Quiz, Questao) e métodos (ex: verificarResposta(), calcularTaxaAcerto()). | Essencial para atingir a meta de **cobertura de testes (≥80%)** (RNF4) e garantir a confiabilidade do *core* do jogo. |
| **Testes Funcionais** | Ações do usuário no sistema (RFs), como selecionar a categoria, responder e visualizar o relatório final. | Validar o fluxo de ponta a ponta na interface de **console** (RNF1). |
| **Testes de Integração** | Comunicação entre classes (ex: Quiz lendo dados via Dados) e a leitura do **arquivo de configuração (JSON/CSV)** (RF1). | Assegurar que a separação entre conteúdo e código está funcionando. |

**3. Escopo dos testes**

**3.1 Dentro do escopo**

A tabela a seguir lista as funcionalidades que serão testadas ativamente, com foco na lógica de negócio e nos RFs essenciais.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Funcionalidades** | **Objetivo do teste** | **Método de teste** |
| **CT-RF1.1** | Carregamento de Dados (RF1) | Validar que o sistema lê corretamente o arquivo JSON/CSV e popula a lista de Questaos. | Teste de Integração |
| **CT-RF4.1** | Processamento Correto (RF4/RF5) | Testar a função Quiz.verificarResposta() com uma entrada correta. | Teste de Unidade (JUnit) |
| **CT-RF4.2** | Processamento Incorreto (RF4/RF5) | Testar a função Quiz.verificarResposta() com uma entrada incorreta. | Teste de Unidade (JUnit) |
| **CT-RF6.1** | Relatório de Acerto (RF6) | Validar que Quiz.calcularTaxaAcerto() retorna 0% após um ciclo de 10 erros. | Teste de Unidade (JUnit) |
| **CT-RF6.2** | Relatório de Pontuação (RF6) | Validar que o relatório final exibe a pontuação máxima após um ciclo de 10 acertos. | Teste Funcional |
| **CT-RNF4.1** | Cobertura de Código | Medir a cobertura de código na classe Quiz (Jacoco/Cobertura). | Teste de Cobertura |

**3.2 Fora do Escopo**

As seguintes funcionalidades e aspectos não serão testados nesta fase inicial.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Funcionalidades** | **Objetivo do teste** | **Método de teste** |
| **CT-FE.1** | Testes de Interface Gráfica (UI) | Testar a aparência visual do console (cores, *layout*, etc.). | Manual (Não Aplicável) |
| **CT-FE.2** | Testes de Segurança | Verificar injeção de comandos ou vulnerabilidades do sistema. | Não Aplicável (App de Console) |
| **CT-FE.3** | Testes de Desempenho | Medir o tempo de resposta do quiz em diferentes máquinas. | Teste de Carga (Não prioridade) |

**4. Planejamento e Realização dos Testes**

Planejamento detalhado conforme o andamento e [Caso de teste](https://docs.google.com/document/d/1F5krIhzSGWnLDVA2nODA0R-wEgFVsMaM91SV_SjNz2A/edit?usp=sharing).

O planejamento detalhado e os casos de teste específicos serão documentados conforme o andamento do desenvolvimento.

**- Ferramenta de Teste:** JUnit 5(é a versão mais recente do framework de testes para a plataforma Java) será o *framework* principal para todos os Testes de Unidade.

**- Gerenciamento:** Os testes serão executados antes de cada *commit* do tipo feat ou fix, garantindo que o requisito de *commits semânticos* (RNF3) esteja ligado à qualidade do código.

**5. Conclusão**

Este plano de testes fornece a estrutura necessária para garantir a qualidade do *Quiz Educativo ODS 4*. O foco em Testes de Unidade (JUnit) e a alta cobertura de código garantem a robustez da lógica central, cumprindo o RNF4 e permitindo uma integração contínua e segura com o uso dos *Commits Semânticos* (RNF3).