SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DE VIÇOSA · UFV

CAMPUS FLORESTAL



**REFINAMENTO DA SPRINT 03**

**CSU01**

GUILHERME BROEDEL ZORZAL - 5064

ARTHUR FERNANDES BASTOS - 4679

Florestal - MG

Outubro - 2024

**SUMÁRIO**

[1. INTRODUÇÃO 3](#__RefHeading___Toc329_1017783847)

[2. CASO DE USO 3](#__RefHeading___Toc327_1017783847)

[CSU01: Jogar 4](#__RefHeading___Toc331_1017783847)

[3. ESTRUTURA GERAL 5](#__RefHeading___Toc325_1017783847)

[4. BACK-END 5](#__RefHeading___Toc323_1017783847)

[4.1 TextoJogavel 5](#__RefHeading___Toc335_1017783847)

[Figura 1 : Classe do Texto Jogável 6](#__RefHeading___Toc337_1017783847)

[5. BANCO DE DADOS 6](#__RefHeading___Toc333_1017783847)

# 

# **1. INTRODUÇÃO**

O objetivo do seguinte documento consiste em apresentar especificações relacionadas ao caso de uso abordado na terceira sprint, de acordo com o product backlog proposto com base no calendário.

Uma convenção importante com relação aos diagramas de classe: caso a representação da classe não possua nenhum método, deve-se entender que a classe precisa possuir getters e setters para seus atributos, a menos que tenham sido dadas outras instruções na descrição da classe presente no documento.

# **2. CASO DE USO**

O caso de uso abordado na Sprint atual ficou decidido como sendo o CSU01: Jogar. Sua especificação, proposta na documentação dos casos de uso em ESOF I, pode ser conferida abaixo. Note que os casos de usos sofreram poucas alterações em relação ao proposto originalmente, ou seja, o fluxo proposto inicialmente continua sendo o mesmo. Apenas algumas palavras foram atualizadas de forma a melhorar a compreensão.

Esses casos de uso são responsáveis por exibir o texto a ser escrito, computar a digitação do usuário e demonstrar a lógica de erros e acertos. A documentação do caso de uso pode ser encontrada abaixo.

|  |
| --- |
| CSU01: Jogar **Nome:** Jogar.  **Ator Primário:** Aluno.  **Sumário:** Ser capaz de iniciar o jogo e começar a contagem de pontos de acordo com o desempenho do aluno.  **Pré-condições:**   * O usuário deve ter criado seu personagem; * O texto deve ter sido selecionado pelo professor.   **Fluxo Principal:**   1. O usuário seleciona a opção "jogar"; 2. O sistema exibe a primeira frase do texto selecionado pelo professor; 3. O aluno digita; 4. O sistema avança o texto e computa o tempo e as entradas; 5. O aluno termina de digitar o texto; 6. O sistema dispara a ação “gerar e salvar relatório individual”; 7. Fim de caso de uso.   **Fluxo Alternativo (4): O texto não foi terminado.**   1. O sistema exibe a próxima frase do texto selecionado pelo professor; 2. Retorna ao passo 3.   **Pós-condições:**   * O usuário será capaz de visualizar sua pontuação. |

# 

# **3. ESTRUTURA GERAL**

Quando a tela de jogo carregar, o nome e o avatar do jogador devem ser exibidos. O acesso à digitação deve ser bloqueado até que o professor escolha um texto, que será carregado em pequenas partes: duas ou três frases por vez. Esse conteúdo estará salvo no banco de dados, e, conforme o jogador digita, o texto já escrito muda de cor, enquanto o que ainda resta para ser digitado permanece na cor branca. Um teclado ilustrativo também acompanha o jogador, acendendo a próxima letra a ser digitada a cada caractere. Ao término de cada frase, devem ser registrados o tempo gasto, a quantidade de erros e a frase seguinte é exibida. Quando o jogador concluir todo o texto, ele será redirecionado para uma tela com os resultados individuais.

# **4. BACK-END**

O back-end, na sprint atual, seguindo o diagrama de Classes disponível no arquivo docx na mesma pasta, ficará responsável pela implementação de algumas classes e relações envolvendo uma entidade do Texto jogável, uma controladora para o jogo, os endpoints para jogar e uma controladora de texto que será implementada apenas com funções específicas dessa sprint (será aprimorada com o caso de uso de Crud de texto).

### 4.1 TextoJogavel

A classe TextoJogavel vai servir para conter uma lista de frases que serão carregadas aos poucos no front-end, ou seja, uma lista de frases (cada uma contendo entre 40 e 55 caracteres) organizadas em ordem. A estrutura de dados a ser utilizada fica a critério dos desenvolvedores, desde que ela cumpra as funcionalidades especificadas.

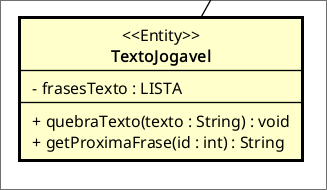
A função quebraTexto() recebe um texto e quebra ele em uma lista de frases (dentro da variável frasesTexto).

A função getProximaFrase() recebe um inteiro (o id da frase sendo utilizada atualmente) e retorna a próxima frase. Por exemplo, se foi recebido o id 2, retorna a frase na posição 3 da lista. Caso a posição que está se tentando acessar não exista, retorna uma string vazia. Além disso, caso o id do parâmetro seja -1, retorna a primeira.

A função dessa classe é prover as frases a serem digitadas no front-end de forma iterativa, para que seja possível realizar a digitação da forma mais fluida possível.

IMPORTANTE: as frases devem possuir um espaço “ “ no fim das frases, com exceção da última. Deve ser feito dessa forma, visto que o front-end irá apenas concatenar as frases de acordo com a ordem em que forem enviadas e, portanto, a concatenação não deve gerar espaços repetidos entre as frases (concatenar uma frase com um espaço no fim e um no início gera dois espaços consecutivos), bem como não deve retirar os espaços (concatenar uniria a última palavra da frase anterior com a primeira, o que é errado).

OBS.: Caso o id passado como parâmetro para a função getProximaFrase não fizer parte da lista, ou for o id da última frase, deve ser retornado uma string vazia “”.



###### **Figura 1 :** Classe do Texto Jogável

## **5. BANCO DE DADOS**

Para a sprint atual, a tarefa do banco de dados será uma extensão do caso de uso passado. O objetivo é modelar uma tabela **avatar** que contenha as colunas **id**, **nomeAvatar** e **avatar**. A coluna avatar deve ser capaz de armazenar os avatares dos alunos. Caso seja necessário adicionar outras colunas ao banco de dados para tornar possível a recuperação das imagens, sintam-se livres para criar.

Fica também a cargo de quem estiver modelando o banco de dados encontrar novas imagens e exemplos de avatares para complementar a ambientação do jogo. Tentem encontrar avatares que sejam razoavelmente parecidos com a ambientação do jogo.