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

CAMPUS FLORESTAL



**REFINAMENTO DA SPRINT 02**

**CSU04 e CSU05**

GUILHERME BROEDEL ZORZAL - 5064

ARTHUR FERNANDES BASTOS - 4679

Florestal - MG

Outubro - 2024

**SUMÁRIO**

[1. INTRODUÇÃO 3](#__RefHeading___Toc445_1146780828)

[2. CASO DE USO 3](#__RefHeading___Toc443_1146780828)

[CSU04: Autenticar Professor 5](#__RefHeading___Toc447_1146780828)

[CSU05: Cadastrar Professor 6](#__RefHeading___Toc449_1146780828)

[3. EXPLICAÇÕES GERAIS E ESCLARECIMENTOS 7](#__RefHeading___Toc441_1146780828)

[4. BACK-END 7](#__RefHeading___Toc439_1146780828)

[4.1 Professor 7](#__RefHeading___Toc451_1146780828)

[5. BANCO DE DADOS 8](#__RefHeading___Toc437_1146780828)

# 

# **1. INTRODUÇÃO**

O objetivo do seguinte documento consiste em apresentar especificações relacionadas ao caso de uso abordado na segunda 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 de 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 CSU04: Autenticar Professor e o CSU05: Cadastrar Professor. Sua especificação, proposta na documentação dos casos de uso em ESOF I, pode ser conferida abaixo. Note que os casos de usos quase não sofreram 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 realizar o cadastro e o login do professor. A documentação do caso de uso pode ser encontrada abaixo.

|  |
| --- |
| CSU04: Autenticar Professor **Nome:** Autenticar professor.  **Ator Primário:** Professor  **Sumário:** Ser capaz de autenticar o professor previamente cadastrado.  **Pré-condições:**   * O usuário deve ter uma conta registrada no sistema. * O sistema deve estar conectado ao banco de dados de usuários. * O sistema deve estar operacional e acessível   **Fluxo Principal:**   1. O usuário seleciona a opção "Entrar como professor" 2. O sistema chama o serviço de autenticação pedindo login e senha 3. O usuário digita o respectivo login e senha 4. O sistema acessa o banco de dados e verifica se a autenticação é válida 5. Autenticação é validada e o sistema exibe as opções do menu do professor 6. Fim do caso de uso   **Fluxo Alternativo (3): Usuário selecionou a opção cadastrar professor**  a. O usuário seleciona a opção cadastrar professor.  b. O sistema dispara a ação para cadastrar o professor.  c. Retorna ao passo 2  **Fluxo Alternativo (4): Erro na autenticação**  a. O sistema pede para o professor digitar novamente o usuário e a senha  b. O professor digite novamente o usuário e a senha  **Fluxo de Exceção (4): Falha no acesso ao sistema**  a. O sistema tenta acessar o banco de dados que não se encontra presente, exibe para o usuário a falha e exibe um botão para voltar ao menu principal.  b. O usuário seleciona o botão de voltar.  **Pós-condições:**   * O usuário estará autenticado e capaz de realizar suas funções |

# 

|  |
| --- |
| CSU05: Cadastrar Professor **Nome:** Cadastrar Professor.  **Ator primário:** Professor.  **Sumário:** O professor deverá ser capaz de criar uma conta no sistema, que ficará associada a seus textos e relatórios.  **Fluxo principal:**   1. O sistema pede que o usuário preencha as seguintes informações: um email, um nome, uma senha, e o iToken 2. O usuário preenche as informações solicitadas e pressiona um botão de confirmação; 3. O sistema cria a conta de professor e direciona o usuário para a tela de login   **Fluxo de Exceção (2): campo faltando**   1. O sistema notifica o usuário de que um campo está faltando; 2. O caso de uso recomeça do passo 1.   **Fluxo de Exceção (2): confirmação da senha não é igual a senha**   1. O sistema notifica o usuário de que a senha e sua confirmação não são iguais; 2. O caso de uso recomeça do passo 1. |

# 

# **3. EXPLICAÇÕES GERAIS E ESCLARECIMENTOS**

Uma explicação importante acerca do funcionamento do sistema é como ocorrerá o uso do sistema. No sistema original, desenvolvido em ESOF I, a ideia seria uma aplicação de rede. Em outras palavras, existiriam vários alunos utilizando o computador. Todos eles fariam requisições para um mesmo computador, o do professor, e a partir dele seriam recuperados os dados dos usuários, textos, etc.

Após a redução de escopo, o fluxo continua parecido, com a exceção de que agora haverá apenas um único computador, que será utilizado tanto pelo aluno quanto pelo professor. Isso facilitará o uso do sistema e o desenvolvimento do mesmo

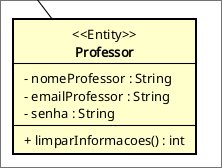
# **4. BACK-END**

O backend, na sprint atual, seguindo o diagrama de Classes disponível no arquivo de docx na mesma pasta, ficará responsável pela implementação de algumas classes e relações envolvendo o Professor, além de criar uma busca de validação com base no email do professor no banco de dados (caso seja encontrado, o email deve ser armazenado esse email na classe professor).

### 4.1 Professor

A classe professor é uma classe relativamente simples. Sua principal função é armazenar os dados do professor uma vez que ele esteja logado no sistema. Os atributos de professor só estarão preenchidos caso o professor esteja logado no sistema.

Essa classe deverá possuir getters e setters para os atributos, além de um método limparInformacoes() que remove as informações do professor da classe. Isso é feito para impedir que, uma vez que o professor esteja logado no sistema, o aluno seja capaz de acessar a área do professor



**Figura 1 : Classe Professor**

# 5. BANCO DE DADOS

Para essa sprint, a única coisa que será exigida do banco de dados é uma tabela Professor, que contenha as colunas nome, email e senha. Todos os três campos devem ser strings (a escolha do tipo específico, como varchar, text e outras alternativas ficam a critério do desenvolvedor). A chave primária deverá ser o email.

Após o desenvolvimento do diagrama relacional, é necessário gerar o script SQL correspondente no MySQL Workbench. Os produtos finais devem ser: o diagrama relacional (o arquivo de edição do workbench), a imagem do diagrama produzido e o script SQL (arquivo .sql).

OBS.: Para gerar o SQL, existe a opção de *forward engineering* dentro do MySQL Workbench. Em caso de dúvidas, consultem os Dev Seniors.