

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



**Design CSU01 e CSU06**

**Bytecraft - 5º ano**

Florestal - MG

2025

## Sumário

|                   |          |
|-------------------|----------|
| <b>CSU01.....</b> | <b>2</b> |
| Frontend.....     | 3        |
| <b>CSU06.....</b> | <b>5</b> |
| Backend.....      | 7        |
| Frontend.....     | 10       |

# CSU01

## Selecionar Tipo Usuário (CSU01)

**Sumário:** Permite que o usuário selecione seu perfil de acesso ao sistema entre duas opções, “Aluno” e “Professor”.

**Ator primário:** Usuário.

### Precondições:

1. O usuário deve ter iniciado a aplicação.

### Fluxo Principal

1. O sistema exibe dois botões na tela: “Aluno” e “Professor”.
2. O usuário clica em uma das opções disponíveis, escolhendo seu tipo de perfil.
3. O sistema registra o tipo de usuário selecionado.
4. O sistema redireciona automaticamente o usuário para o fluxo correspondente, e o caso de uso termina.

### Fluxo de Exceção(2): Falha no redirecionamento

- a. O usuário clica em um botão, mas ocorre uma falha na navegação.
- b. O sistema exibe a mensagem “Erro ao carregar a próxima etapa. Tente novamente”.
- c. O sistema permanece na tela atual, permitindo nova tentativa de seleção.

### Pós-condições

- a. (Sucesso): O sistema registra corretamente o tipo de usuário selecionado e inicia o fluxo apropriado.
- b. (Falha): O sistema informa o erro de navegação e não altera a tela até que a seleção seja realizada novamente com sucesso.

### Regras de Negócio

**RN01:** O usuário deve obrigatoriamente selecionar um perfil de acesso para prosseguir a navegação.

## Frontend

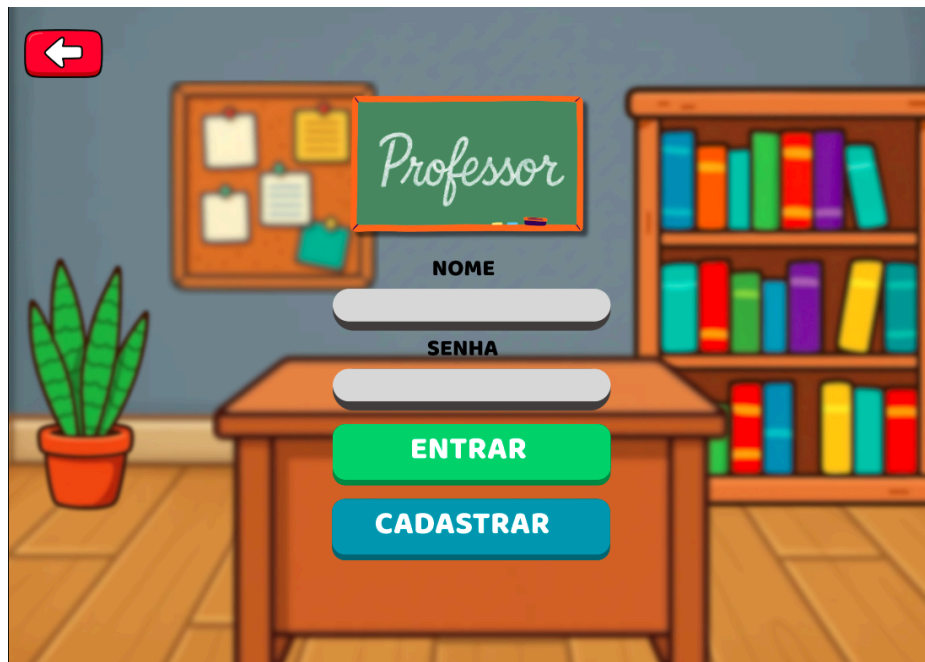
Devem ser criadas 3 páginas: página de seleção de tipo de usuário, página inicial do aluno e página inicial do professor. Devem ser seguidas como base de implementação as figuras abaixo.



Figura 1: Seleção de tipo de usuário.



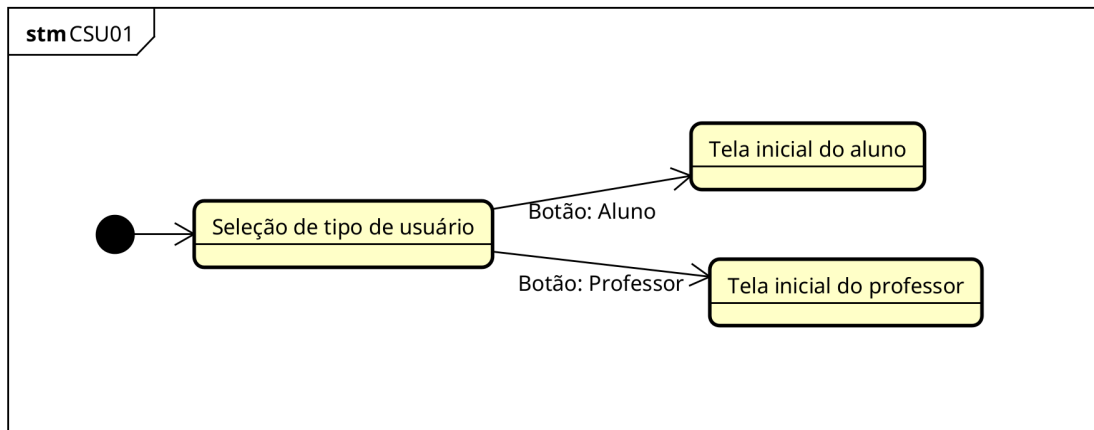
Figura 2: Tela inicial do aluno.



**Figura 3:** Tela inicial do professor.

O fluxo das telas deve ser o seguinte, como pode ser visualizado na Figura 4:

- **Tela Selecionar Tipo Usuário**
  - O botão “Aluno” deve registrar o perfil escolhido e redirecionar para a tela inicial de Aluno.
  - O botão “Professor” deve registrar o perfil escolhido e redirecionar para a tela inicial de Professor.
  - Em caso de falha no redirecionamento, o sistema deve exibir a mensagem “Erro ao carregar a próxima etapa. Tente novamente.” e manter o usuário nesta tela.
- **Tela Aluno inicial**
  - Representa o início do fluxo específico para o perfil de aluno.
  - Não é necessário detalhar funcionalidades adicionais neste caso de uso, apenas a confirmação de que o acesso ocorreu corretamente.
- **Tela Professor inicial**
  - Representa o início do fluxo específico para o perfil de professor.
  - Assim como a tela de aluno, não é necessário detalhar funcionalidades adicionais dentro deste caso de uso.



**Figura 4:** Máquina de estados representando o fluxo de telas de CSU01.

Espera-se que o sistema seja capaz de registrar corretamente o perfil escolhido, redirecionar para o fluxo correspondente e tratar erros de navegação de forma clara, garantindo que o usuário possa realizar uma nova tentativa de seleção caso ocorra alguma falha.

## CSU06

### Navegar para a tela de níveis (CSU06)

**Sumário:** Permite que o aluno acesse a tela de níveis após se vincular à sala, onde poderá escolher entre os modos de níveis de dificuldade disponíveis.

**Ator primário:** Aluno

#### Precondições:

1. O aluno já deve estar autenticado e corretamente vinculado a uma sala existente.

#### Fluxo Principal

1. O aluno é redirecionado para a tela de níveis, que exibe os botões: Fácil, Médio e Difícil
2. O aluno seleciona a opção desejada.
3. O sistema registra o nível selecionado.
4. O sistema redireciona automaticamente o aluno para a tela de Fases.

#### Fluxo de Exceção(2): Falha no redirecionamento

1. Se o usuário clica em um botão, mas ocorre uma falha na navegação:
  - a. O sistema exibe a mensagem “Erro ao carregar a Tela de Fases. Tente novamente”
  - b. O sistema permanece na tela atual, permitindo nova tentativa de seleção.

#### Pós-condições

- a. (Sucesso): O sistema registra corretamente o tipo de nível selecionado e inicia o fluxo apropriado.
- b. (Falha): O sistema informa o erro de navegação e não altera a tela até que a seleção seja realizada novamente com sucesso.

#### Regras de Negócio:

-

## Backend

A seguir está o modelo de classes de especificação para este caso de uso. Na Figura 4 é possível visualizar a estrutura de pacotes e as relações entre as classes. Serão criadas duas classes de Entidade, que estarão no pacote *model*: **Aluno** e **Sala**. No pacote *enum* está a classe de enumeração **NivelDificuldadeEnum**, que representa os níveis de dificuldades disponíveis. O pacote *repository* representa as classes que fazem comunicação direta com o banco de dados, sendo que **AlunoRepository** fará comunicação com a tabela **Aluno** exclusivamente. Já o pacote *controller* representa as classes de controle que receberão requisições REST e serão usadas pelo frontend para se comunicar com o backend, nele está a classe **AlunoController** que é responsável pelas requisições relacionadas ao aluno.

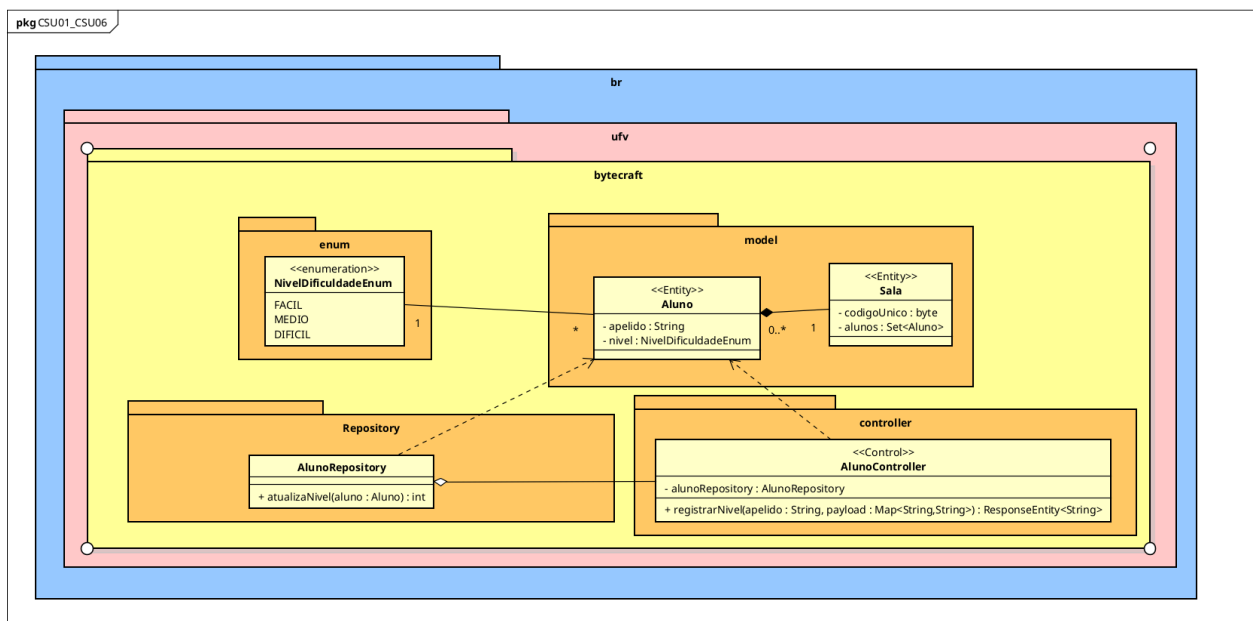
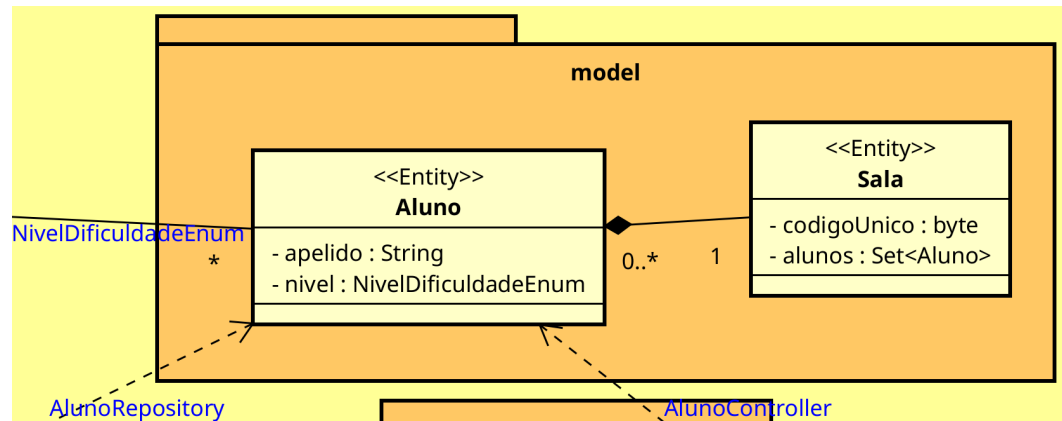


Figura 5: Modelo completo de classes.

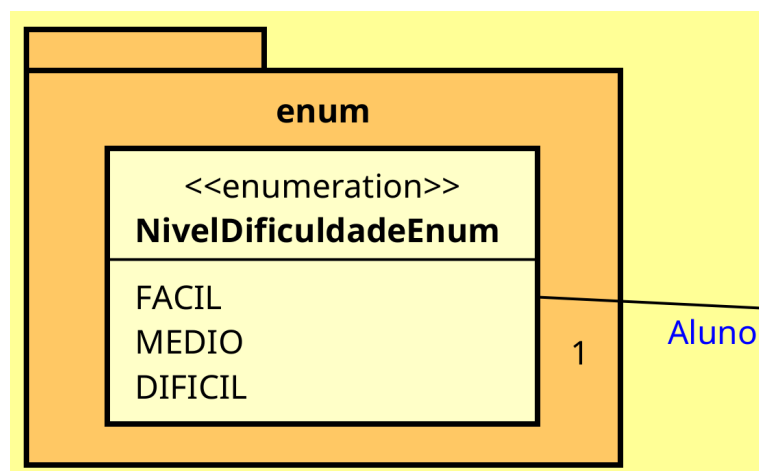
A classe **Aluno** deve ter dois atributos privados com seus respectivos getters e setters: **apelido**, do tipo **String**, e **nivel**, do tipo **NivelDificuldadeEnum**. Já a classe **Sala** representa a relação entre alunos e a sala única do professor, ela deve possuir os atributos privados **codigoUnico**, do tipo **byte** (escolhido pelo motivo do código único não possuir muitos dígitos), e o atributo **alunos**, do tipo **Set<Aluno>**, que representa os alunos ligados à sala. As classes de entidade tem uma relação de composição entre **Sala** e **Aluno**, devido ao fato que todo aluno deve



estar obrigatoriamente relacionado unicamente a uma única sala e uma sala pode estar relacionada com vários alunos.



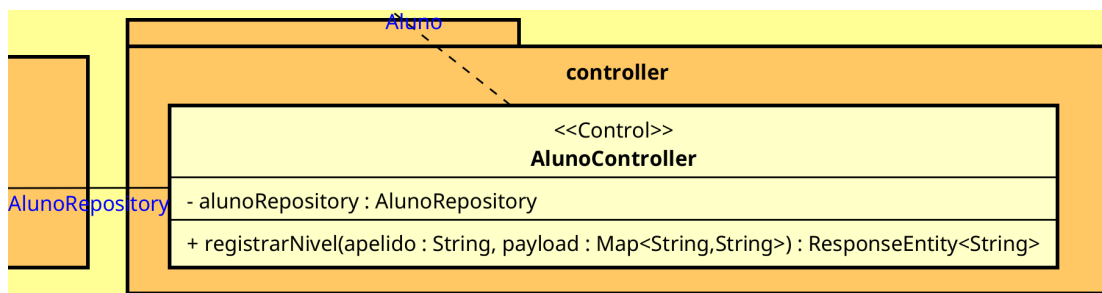
**Figura 6:** Classes de entidade.



**Figura 7:** Classes de enumeração.

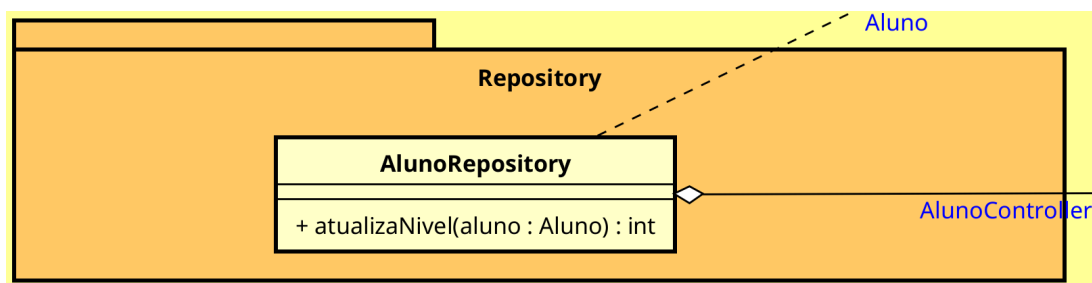
A classe **AlunoController** é uma classe de controle que deve possuir as anotações `@RestController` e `@RequestMapping("/api/alunos")` do springframework. O caminho `"/api/aluno"` representa o caminho que deve ser utilizado pelo frontend para qualquer requisição relacionada a aluno. Além disso, a classe deve possuir um atributo privado e **final** chamado `alunoRepository` do tipo **AlunoRepository**, que será criado pelo construtor dessa classe, que por sua vez deve possuir a anotação `@Autowired`. Esta classe deve implementar o método **registrarNivel**, que retorna o tipo **ResponseEntity<String>** com uma mensagem de sucesso ou erro; o método deve ter a anotação `@PostMapping("/{apelido}/registrarNivel")` e deve receber

como parâmetro o **apelido**, do tipo **String** (esse parâmetro deve possuir a anotação **@PathVariable**, que conecta o apelido presente no caminho da requisição) e **payload**, do tipo **Map<String,String>** que deve ser enviado como um JSON que deve conter o par chave-valor “nivel” : “<difículdade do nível>”, sendo que <difículdade do nível> deve respeitar a forma de escrita do enum **NivelDificuldadeEnum**. Por fim, este método mencionado deve realizar o registro do nível do aluno no banco de dados.



**Figura 8:** Classes de controle.

A classe **AlunoRepository** deve fazer a comunicação entre a tabela aluno e a classe controladora e deve possuir a anotação **@Repository**. Ela deve implementar o método **atualizaNivel** que retorna um inteiro representando se a operação foi bem sucedida. Esse método deve receber como parâmetro um aluno e deve ser responsável exclusivamente por atualizar a coluna nivel do aluno em questão, usando o apelido único do aluno para realizar a operação.



**Figura 9:** Classes de comunicação com o banco de dados.

## Frontend

Devem ser criadas 2 páginas: página de seleção de níveis e página de seleção de fases. Devem ser seguidas como base de implementação as figuras abaixo.



Figura 10: Seleção de níveis.



Figura 11: Seleção de fases.

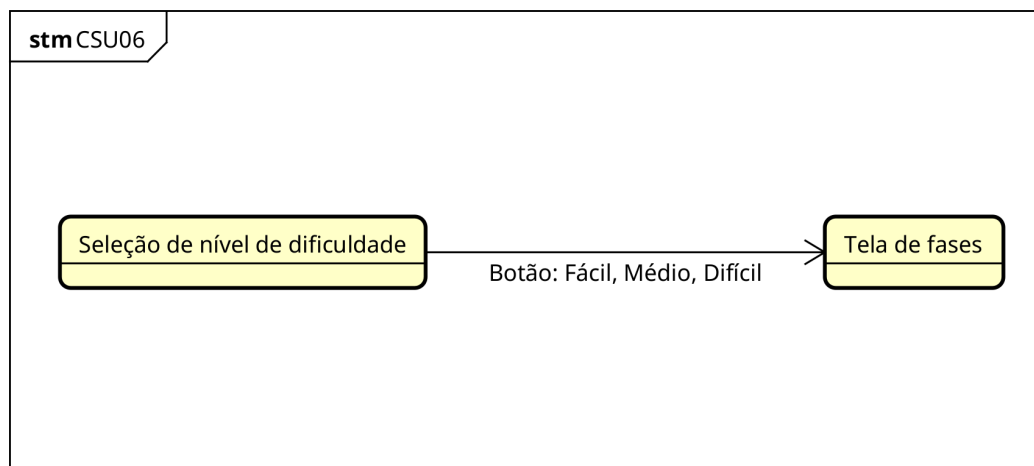
O fluxo das telas deve ser o seguinte:

- **Tela de Níveis**

- Exibe os botões “Fácil”, “Médio” e “Difícil”.
- Ao clicar em um dos botões, o sistema deve registrar o nível de dificuldade escolhido e redirecionar o aluno para a tela “Fases”. A operação de registrar deve realizar uma requisição ao backend, enviando para o caminho “/api/alunos/{apelido}/registrarNível” o apelido e o nível selecionado pelo aluno.
- Em caso de falha no redirecionamento, o sistema deve exibir a mensagem “Erro ao carregar a Tela de Fases. Tente novamente.” e manter o aluno nesta tela, permitindo outra tentativa de seleção.

- **Tela de Fases**

- Representa o início do fluxo específico de fases, de acordo com o nível escolhido pelo aluno.
- Não é necessário detalhar funcionalidades adicionais neste caso de uso, apenas garantir que o acesso tenha ocorrido corretamente.



**Figura 12:** Máquina de estados representando o fluxo de telas de CSU06.

Espera-se que o sistema seja capaz de registrar a escolha do aluno, direcioná-lo para a tela de fases correspondente e lidar com erros de navegação de forma clara, permitindo que o usuário tente novamente em caso de falha.