

# Protótipo de Classes e Diagramas Detalhados

## Protótipo de Classes e Diagramas Detalhados

1. Identificação
2. Diagrama UML de Classes (Detalhado)
3. Diagramas de Interação (Sequência ou Atividade)
4. Protótipo de Código (Skeleton Classes)
5. Planejamento de Uso dos Conceitos de POO
6. Aplicação dos Padrões de Projeto
7. Persistência e Tratamento de Exceções (Planejamento Inicial)
8. Critérios de Sucesso desse Entregável

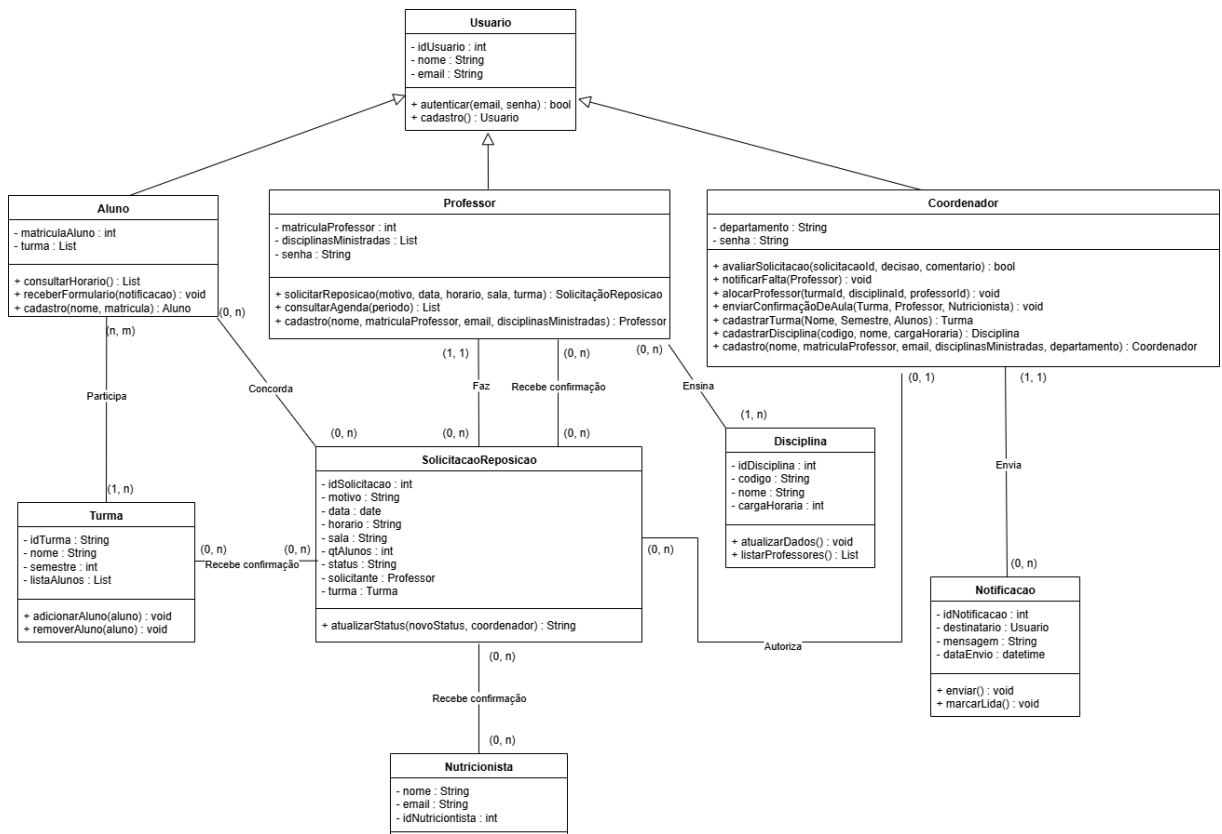
## Entrega

Este entregável tem como finalidade detalhar a modelagem do sistema e construir a estrutura inicial do código (Esqueletos de classes, interfaces e pacotes), permitindo validar se a arquitetura e os requisitos estão consistentes antes da implementação completa.

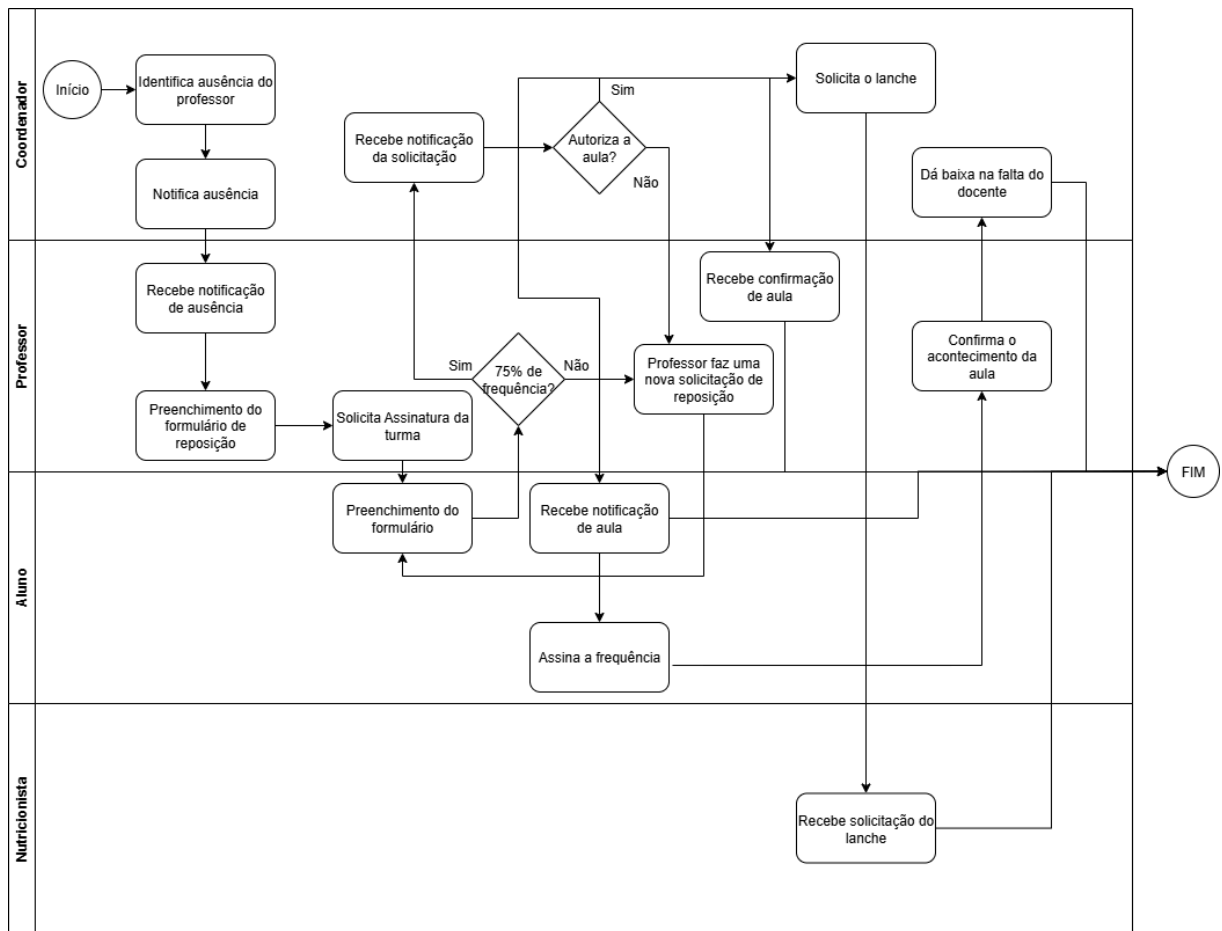
## **1. Identificação**

- Sistema de Reposição de Aulas - IFCE
- Equipe:
  - Rafael - Scrum Master
  - Marcos Barros - Desenvolvedor Fullstack
  - Erlano Benevides - Desenvolvedor Fullstack
- Data de entrega: 08/09/2025

## 2. Diagrama UML de Classes (Detalhado)



### 3. Diagramas de Interação (Atividade)



### 4. Protótipo de Código (Skeleton Classes)

<https://github.com/MarcosBarros1/Sistema-de-reposicao-de-aulas>

### 5. Planejamento de Uso dos Conceitos de POO

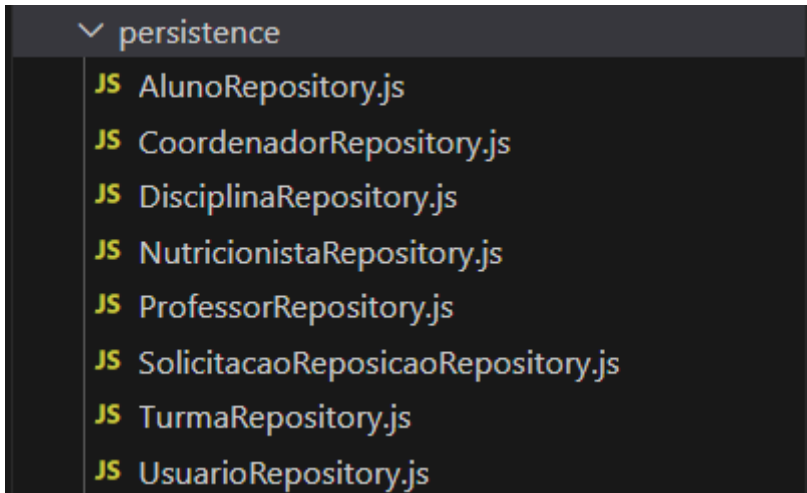
- Usamos herança com superclasse sendo usuário e classes filhas sendo professor, aluno e coordenador e sobrescrita no método de cadastro dessas três classes. Além disso, criamos um enum para o status da solicitação

### 6. Aplicação dos Padrões de Projeto

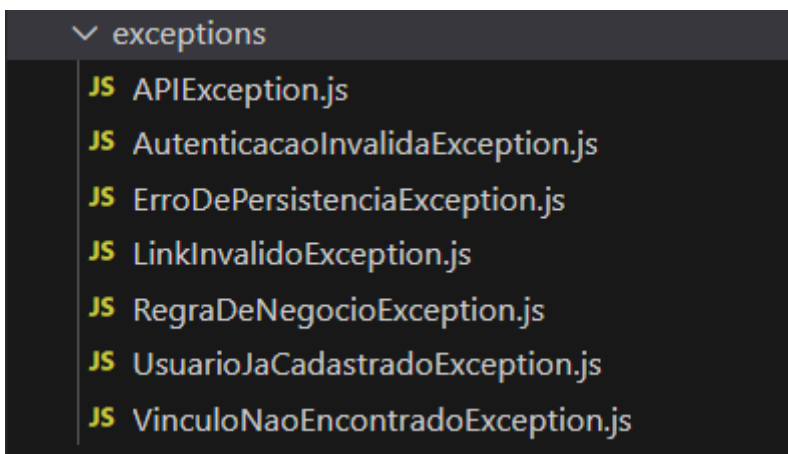
- **Singleton.** Garantindo que haverá apenas uma conexão aberta (db.js), evitando o consumo excessivo de recursos.
- **Repository.** É um padrão de design que isola a lógica de acesso a dados. Aplicado através da pasta persistence

## 7. Persistência e Tratamento de Exceções (Planejamento Inicial)

- Camada de acesso aos dados:



- Criar exceções customizadas (sem lógica ainda), ex.:



## 8. Critérios de Sucesso desse Entregável

- Estrutura inicial do código reflete corretamente a arquitetura definida.
- UML de classes **detalhado e consistente** com os requisitos.
- Pelo menos **dois diagramas de interação** (sequência ou atividade).
- Classes esqueleto organizadas em pacotes no GitHub.
- Uso inicial de conceitos de POO (mesmo que apenas planejado com comentários).
- Evidência da aplicação de pelo menos **dois padrões de projeto**.

# Entrega

## 1. Documento em PDF contendo:

- UML de classes detalhado.
- Diagramas de sequência/atividade.
- Explicação dos pontos acima.

## 2. Repositório GitHub com:

- Estrutura inicial do projeto.
- Classes esqueleto (com comentários e assinaturas).