### **Documento de Especificação de Requisitos de Software (ERS)**

**Projeto:** Sistema de Controle Acadêmico  
**Disciplina:** Engenharia de Software I  
**Curso:** Análise e Desenvolvimento de Sistemas  
  
**Grupo:** G2M – Gabriel Henrique Menegon de Morais, Matheus da Silva Freitas e Miguel Honório dos Santos

### **Introdução**

Este documento tem como finalidade apresentar a especificação dos requisitos do sistema de controle acadêmico denominado Alunus, desenvolvido no contexto da disciplina de Engenharia de Software I, do curso de Análise e Desenvolvimento de Sistemas. Ele reúne de maneira estruturada todas as informações necessárias para o entendimento, desenvolvimento, validação e manutenção do sistema.

A introdução serve como ponto de partida para o leitor compreender o escopo do projeto, bem como sua relevância e o contexto em que está inserido. Através das seções seguintes, são detalhados os requisitos funcionais e não funcionais, os casos de uso, os diagramas de atividades, os protótipos de telas e demais elementos que contribuem para a definição precisa do comportamento esperado do sistema.

O documento é direcionado a todos os envolvidos no projeto, incluindo desenvolvedores, analistas, professores, stakeholders acadêmicos e qualquer parte interessada na implementação e uso do sistema. Ele proporciona uma base sólida para que o software seja desenvolvido de forma eficiente, coerente e alinhada com os objetivos definidos pela equipe.

#### **Objetivo do Documento**

O objetivo deste documento é descrever de forma clara, objetiva e detalhada os requisitos do sistema de controle acadêmico, abrangendo suas funcionalidades, restrições e representações visuais. Essa especificação busca estabelecer um entendimento comum entre os membros da equipe de desenvolvimento e os demais interessados, servindo como referência durante todo o ciclo de vida do software.

Além de definir o que o sistema deve fazer, o documento também registra os critérios que orientarão o processo de implementação e validação, assegurando que o produto final atenda às expectativas dos usuários. Ao formalizar os requisitos, este documento contribui diretamente para a qualidade, rastreabilidade e manutenibilidade do sistema, reduzindo ambiguidades e aumentando a eficiência do desenvolvimento.

### **2. Visão Geral do Sistema**

#### **2.1. Visão do Produto**

O Alunus é um sistema de controle acadêmico desenvolvido como um aplicativo mobile, voltado para instituições de ensino que desejam modernizar a gestão escolar de forma prática, acessível e eficiente. Pensado especialmente para escolas com infraestrutura tecnológica limitada, o sistema foi projetado para funcionar em dispositivos móveis com baixa capacidade de processamento, eliminando a dependência de registros físicos, como diários de classe em papel.

Entre as funcionalidades centrais do sistema, destacam-se o login com diferentes níveis de acesso para alunos, professores e coordenadores, o cadastro e a matrícula de alunos, a organização de turmas e o controle da frequência escolar. Os professores podem registrar o conteúdo ministrado em sala, enquanto os alunos têm acesso às suas notas, frequência e planejamento das disciplinas. A coordenação, por sua vez, pode inserir o plano de aula de cada disciplina, pode acompanhar o progresso das aulas ministradas, além de gerenciar os usuários do sistema, inclusive com a opção de restringir acessos quando necessário.

O principal diferencial do Alunus está na sua simplicidade de uso e na acessibilidade, sua interface foi projetada para ser intuitiva, dispensando a necessidade de treinamentos complexos. Além disso, por ser um sistema leve e funcional, pode ser utilizado em ambientes escolares com recursos reduzidos, oferecendo uma solução moderna que centraliza todas as informações acadêmicas em um único ambiente digital. Dessa forma, o Alunus se apresenta como uma ferramenta estratégica para facilitar o dia a dia escolar e tornar a administração educacional mais ágil, organizada e conectada.

#### **2.2. Justificativa**

Atualmente, muitas instituições de ensino, especialmente as de pequeno porte ou com infraestrutura tecnológica limitada, ainda dependem de registros físicos para organizar informações acadêmicas. Esse cenário resulta em diversos problemas, como a perda de documentos, lentidão nos processos de matrícula e controle de frequência, além de dificultar o acompanhamento do desempenho dos alunos por parte de professores, coordenação e até mesmo dos próprios estudantes.

Diante dessa realidade, torna-se evidente a necessidade de um sistema que centralize, digitalize e facilite a gestão escolar. A proposta do Alunus surge, portanto, como uma resposta direta a esses desafios. Ao oferecer uma plataforma digital acessível por dispositivos móveis, o sistema tem como objetivo substituir os registros manuais, automatizar processos e proporcionar um ambiente mais eficiente, seguro e organizado para o cotidiano acadêmico.

Além disso, o uso de um sistema intuitivo e de fácil operação elimina a necessidade de grandes investimentos em equipamentos ou capacitação técnica, permitindo que até mesmo instituições com poucos recursos tecnológicos possam se beneficiar da digitalização dos seus processos administrativos e pedagógicos.

#### **2.3 Objetivos e Benefícios**

O principal objetivo do sistema Alunus é otimizar a gestão acadêmica por meio da digitalização de atividades escolares essenciais, como o controle de matrícula, registro de frequência, lançamento de planos de aula e acompanhamento do conteúdo ministrado. O sistema também busca melhorar a comunicação entre os agentes da comunidade escolar, garantindo que coordenadores, professores e alunos tenham acesso rápido e seguro às informações relevantes.

Entre os benefícios esperados com a implantação do sistema, destacam-se a redução de erros manuais e retrabalhos, a agilidade nos processos internos, o aumento da transparência na gestão escolar e o fortalecimento do acompanhamento pedagógico por parte dos estudantes. Com o Alunus, os professores ganham mais autonomia e praticidade para registrar e consultar dados, a coordenação passa a ter uma visão mais ampla e detalhada do andamento das atividades escolares, e os alunos têm acesso facilitado ao seu próprio desempenho, promovendo maior engajamento com os estudos.

### **3. Requisitos do Sistema**

No âmbito da Engenharia de Software, requisitos são definições formais e detalhadas sobre o que um sistema deve fazer e como ele deve se comportar. Eles representam as necessidades, expectativas e restrições dos usuários finais, clientes, stakeholders e do próprio ambiente em que o software será inserido. Em outras palavras, os requisitos são a base fundamental de qualquer projeto de software, pois definem com clareza o problema a ser resolvido e os critérios que o sistema deve atender para ser considerado satisfatório.

A importância dos requisitos está no fato de que eles orientam todas as fases do desenvolvimento de software. Desde a concepção inicial até a validação final do sistema, os requisitos funcionam como um guia para os desenvolvedores, analistas, testadores e gestores do projeto. Um sistema que não possui requisitos bem definidos tende a apresentar problemas como retrabalho, atrasos, aumento de custos e, principalmente, insatisfação do cliente.

A definição dos requisitos ocorre nas etapas iniciais do processo de desenvolvimento e envolve a comunicação ativa entre os desenvolvedores e os stakeholders (clientes, usuários finais, gestores). Técnicas como entrevistas, questionários, observação direta, análise de documentos e reuniões de levantamento de requisitos são amplamente utilizadas. Em abordagens ágeis, como o Scrum, os requisitos são definidos como histórias de usuário no backlog do produto, o que facilita sua priorização e adaptação ao longo do projeto.

Uma vez definidos, os requisitos servem como documentação de referência para todas as próximas etapas do projeto. Durante a análise, os requisitos são estudados e refinados para definir a arquitetura do sistema. Na implementação, eles orientam o desenvolvimento do código-fonte. Nos testes, os requisitos são usados como base para criar os casos de teste e verificar se o sistema atende aos critérios especificados. Por fim, na fase de validação, os requisitos ajudam a garantir que o software entregue corresponde às expectativas do cliente.

Além disso, os requisitos desempenham um papel essencial na gestão de mudanças. À medida que o projeto evolui, novos requisitos podem surgir, e mudanças podem ser solicitadas. Ter um controle estruturado sobre os requisitos permite que essas mudanças sejam avaliadas com clareza, reduzindo o impacto negativo sobre o cronograma e a qualidade do produto final.

**3.1 Requisitos Funcionais**

Requisitos funcionais são descrições detalhadas das funções, comportamentos e ações que um sistema de software deve executar. Eles especificam o que o sistema deve fazer para atender às necessidades dos usuários, estabelecendo claramente as operações que ele deve ser capaz de realizar. Esses requisitos estão diretamente relacionados à lógica de negócio, às regras operacionais e às tarefas que o software deverá executar durante sua utilização.

Esses requisitos são fundamentais porque orientam as equipes de desenvolvimento sobre como o sistema deve se comportar em situações específicas. Eles são utilizados como base para o design da arquitetura do sistema, para a codificação de suas funcionalidades e, posteriormente, para a criação de casos de teste que permitirão validar o comportamento do software. Abaixo estão os requisitos funcionais do sistema:

RF01 – Registrar frequência dos alunos

RF02 – Registrar plano de aula

RF03 – Registrar conteúdo ministrado

RF04 – Cadastrar professores

RF05 – Cadastrar alunos

RF06 – Realizar matrícula dos alunos

RF07 – Dividir alunos em turmas

RF08 – Retirar acesso de alunos e professores

RF09 – Visualizar aulas ministradas RF10 – Visualizar notas

RF11 – Visualizar frequência

RF12 – Visualizar plano de aula

RF13 – Realizar login no sistema

#### **3.1.1 Product Backlog com Histórias de Usuário**

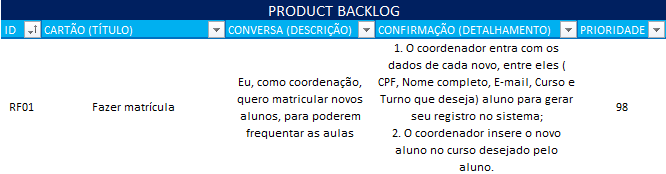
No desenvolvimento ágil de software, especialmente em metodologias como Scrum e XP, as histórias de usuário representam uma forma simples, prática e centrada no usuário de capturar os requisitos funcionais de um sistema. Elas descrevem, de maneira breve e informal, uma funcionalidade desejada sob a perspectiva do usuário final, destacando o valor ou benefício que essa funcionalidade proporciona.

Diferentemente das especificações tradicionais, que costumam ser longas e detalhadas, as histórias de usuário são curtas narrativas escritas em linguagem natural, com o propósito de facilitar a comunicação entre os membros da equipe e garantir o foco nas necessidades reais do usuário. Elas geralmente seguem um formato padrão, conhecido como modelo de três partes:

Como [tipo de usuário], eu quero [funcionalidade], para que [objetivo/benefício].

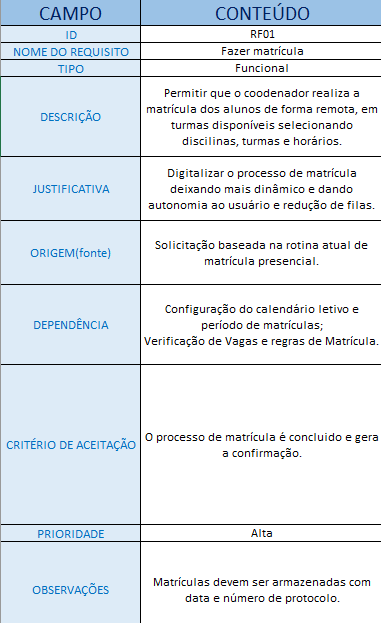
Esse formato oferece uma visão clara de quem está solicitando, o que está sendo solicitado e por que isso é importante. Essa abordagem promove um alinhamento maior entre o time técnico e os stakeholders, facilita a priorização das funcionalidades no backlog do produto e serve como base para a criação de critérios de aceitação e testes.

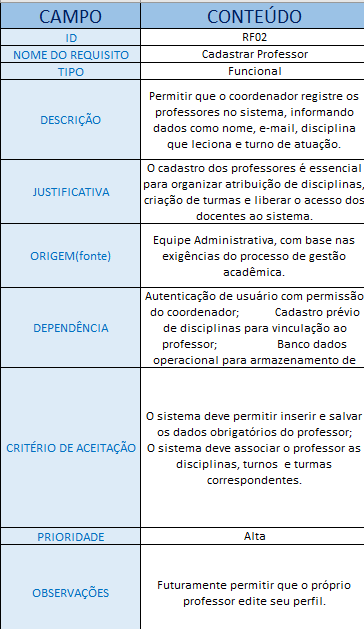
As histórias de usuário são frequentemente utilizadas em conjunto com reuniões de planejamento, onde são discutidas, refinadas e quebradas em tarefas menores durante os ciclos de desenvolvimento, conhecidos como sprints. Sua simplicidade torna o processo mais dinâmico, flexível e adaptável a mudanças, o que é essencial em projetos com requisitos evolutivos ou com alto grau de incerteza. Abaixo temos um exemplo de Product Backlog usado no desenvolvimento.

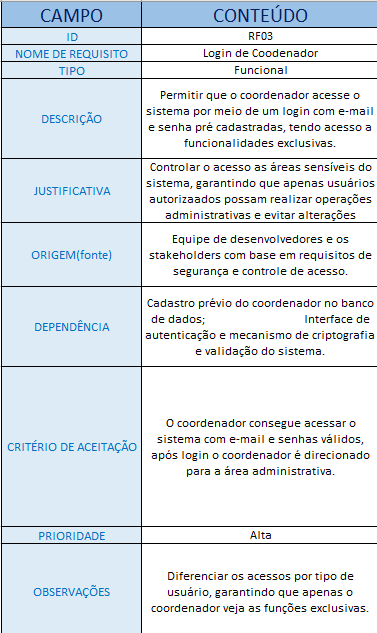


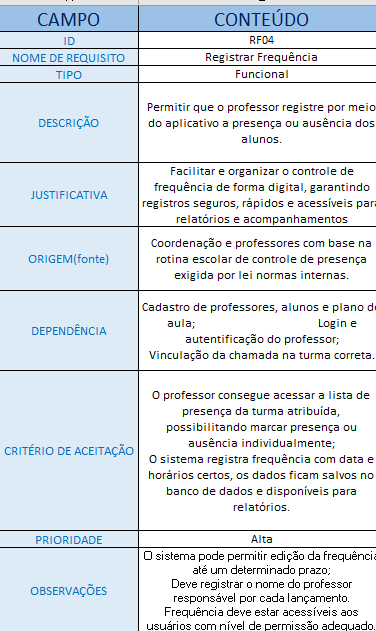
#### **3.1.2. Fichas de Requisitos**

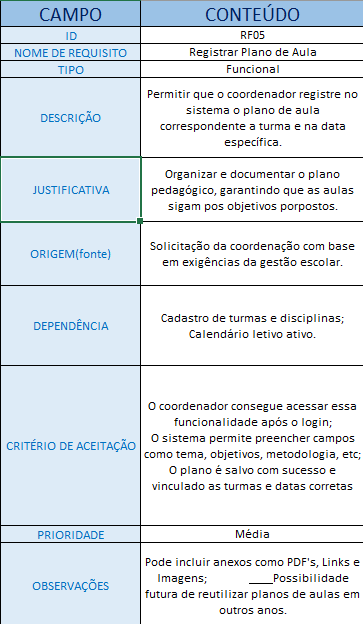
As fichas de requisitos são documentos estruturados que descrevem, de forma clara e padronizada, cada um dos requisitos identificados durante a fase de levantamento e análise de um sistema. Elas têm como principal objetivo organizar as informações relacionadas a cada funcionalidade do software, garantindo rastreabilidade, clareza e controle ao longo do ciclo de desenvolvimento. Nas tabelas abaixo temos a fichas dos requisitos funcionais do sistema:

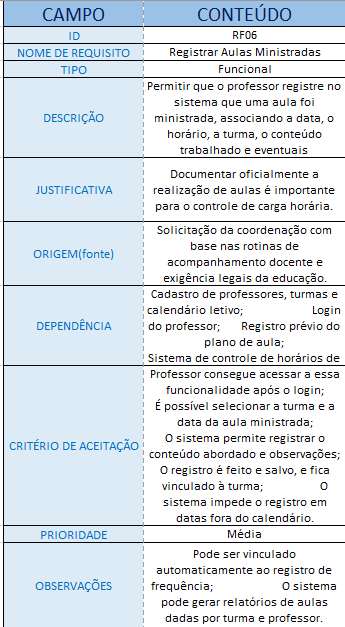
**

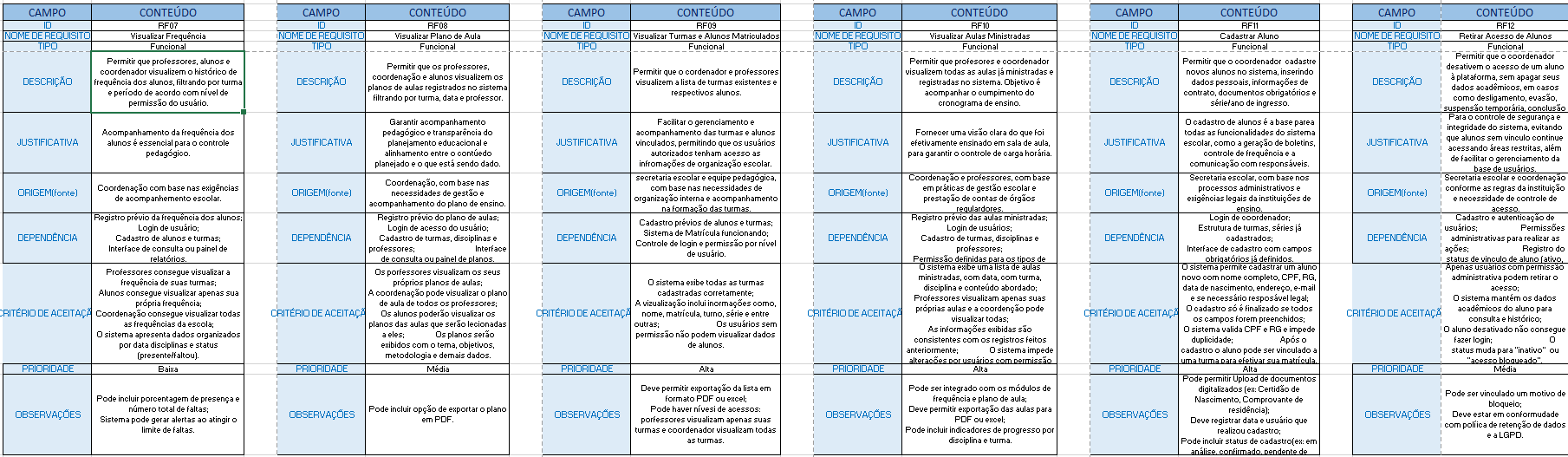
**

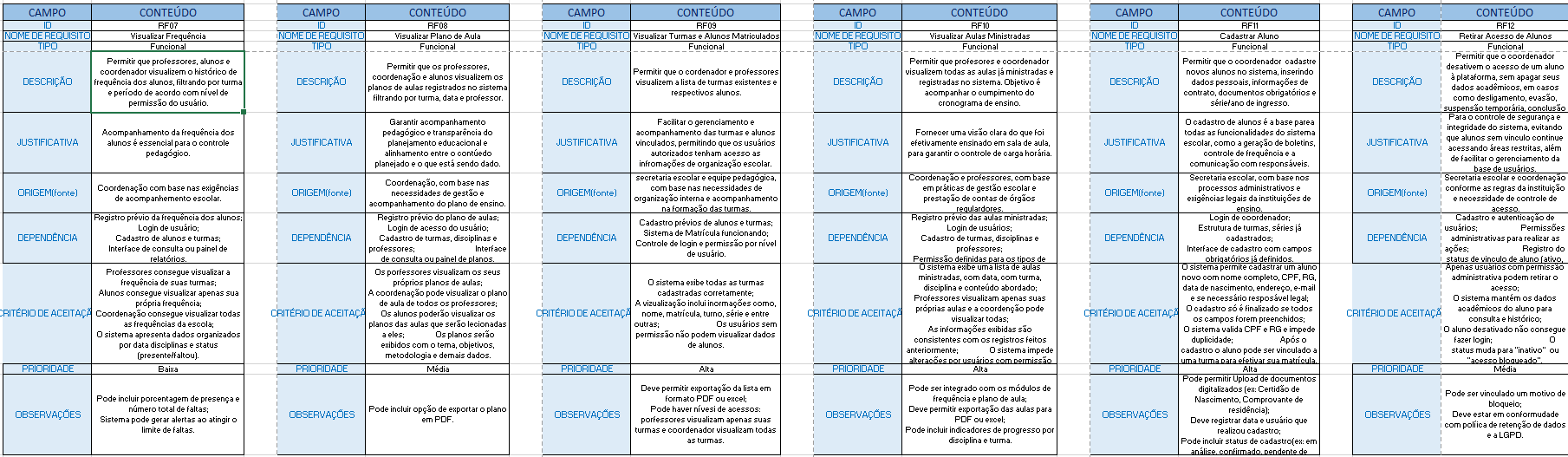
**

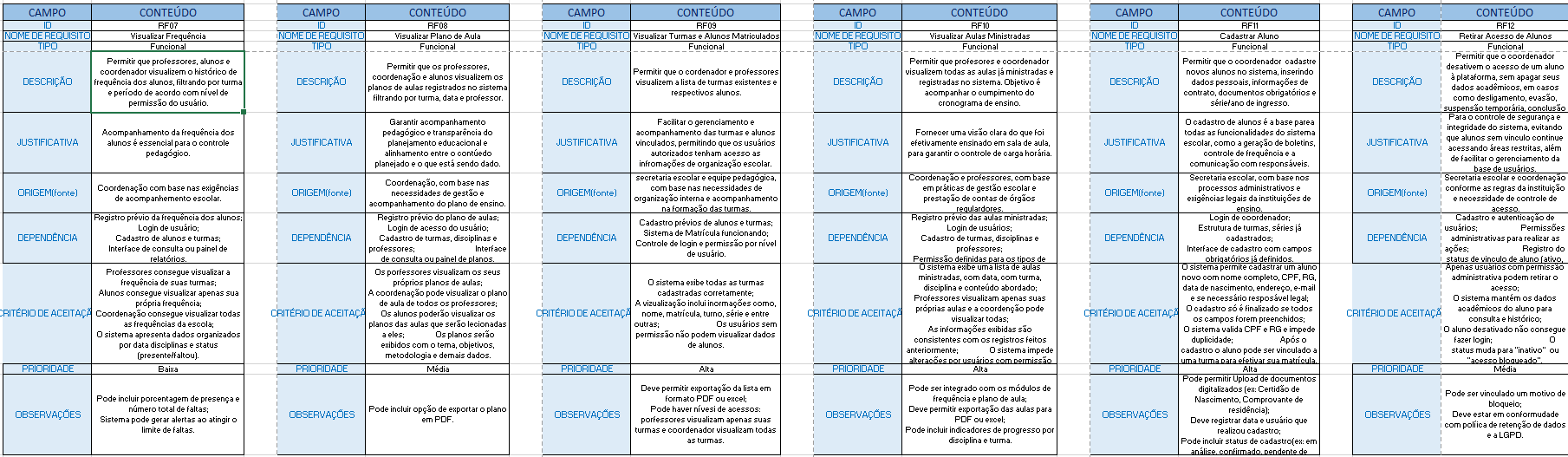
**

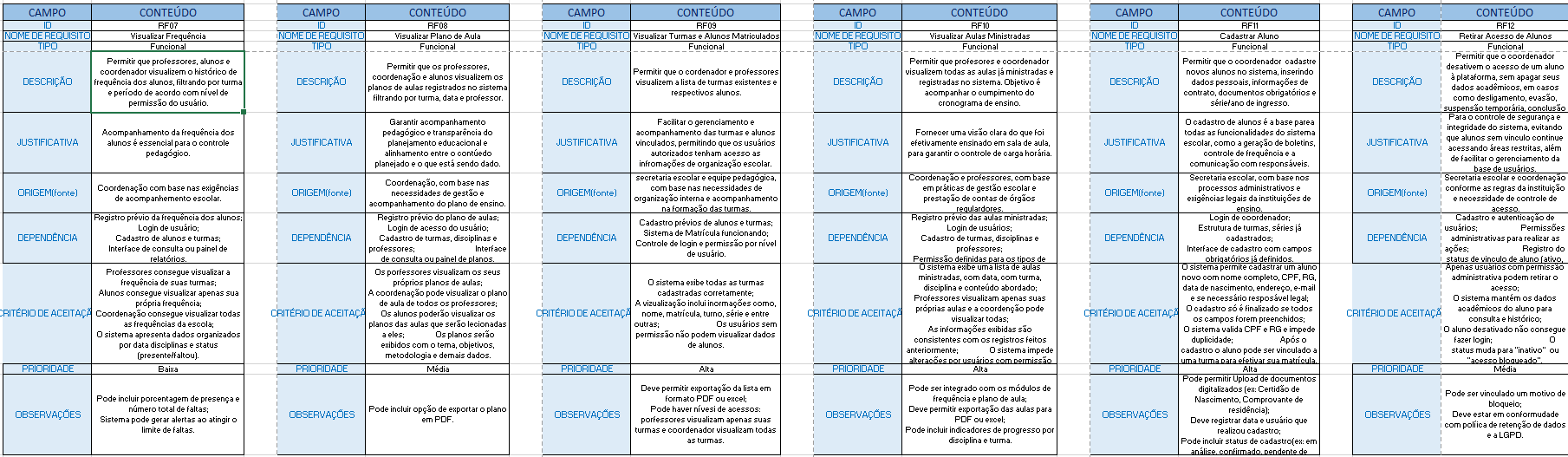
**

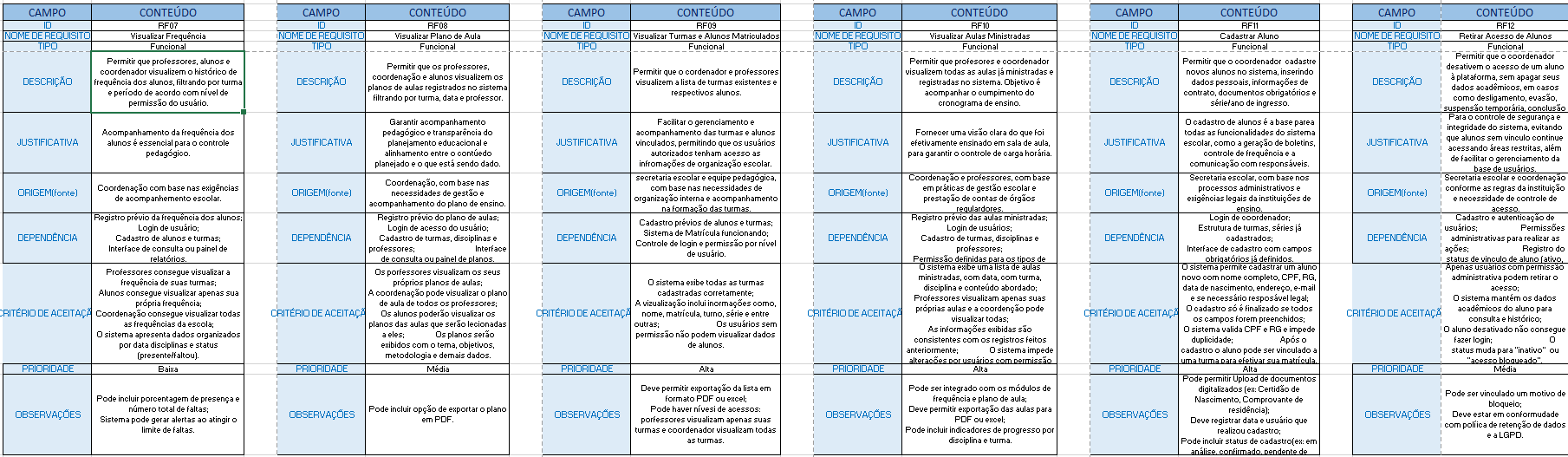
**

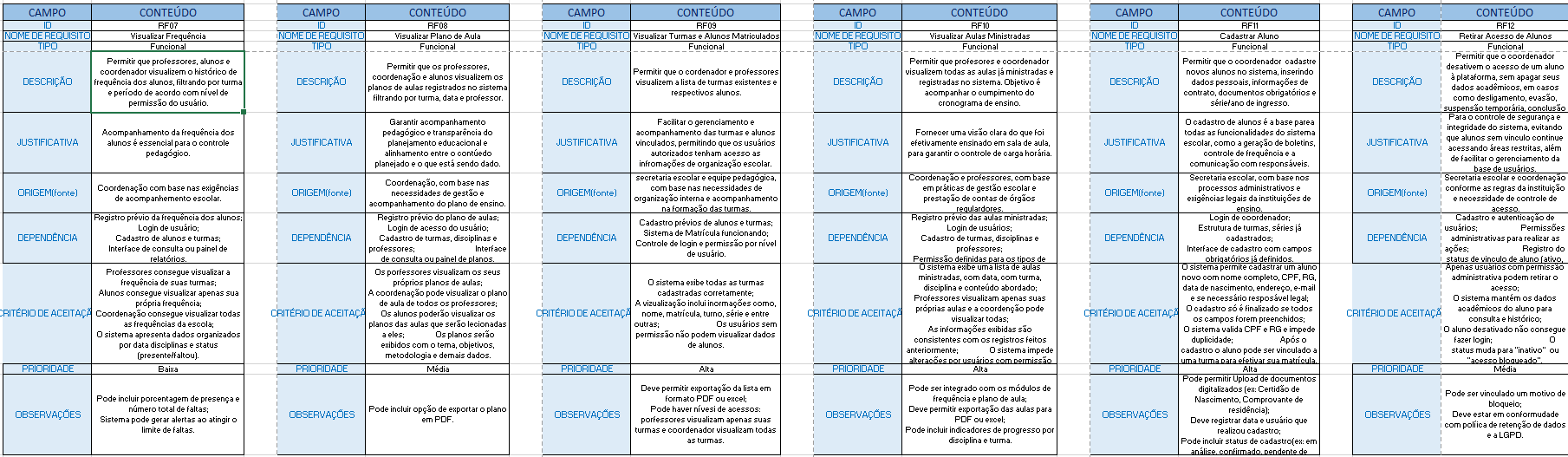
**

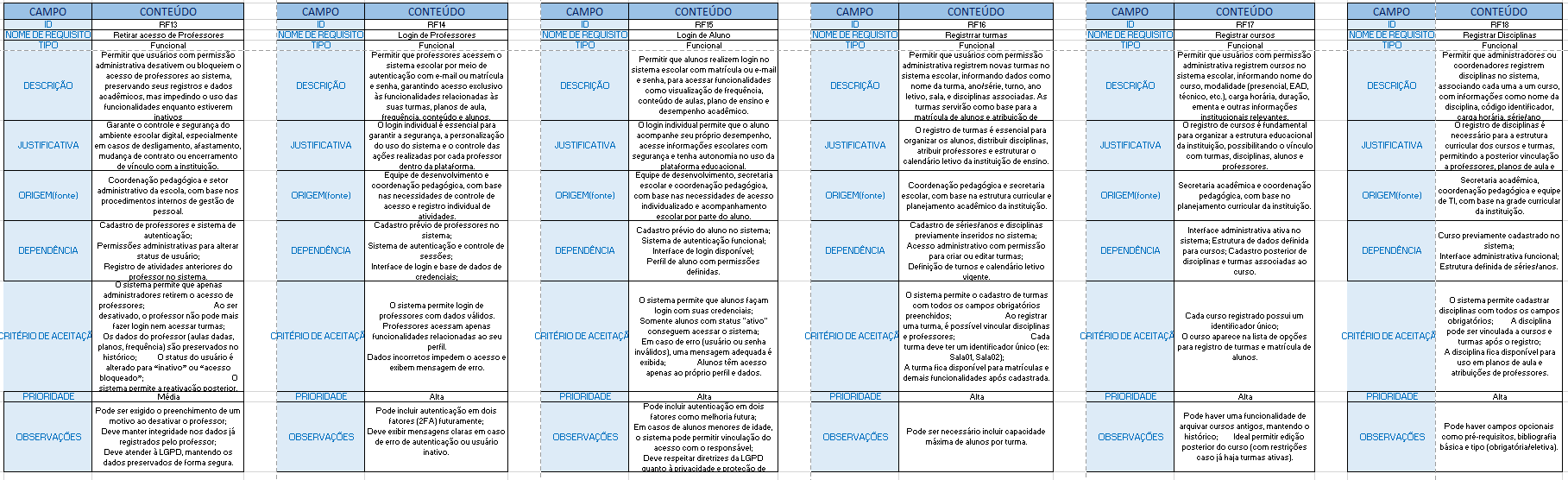
**

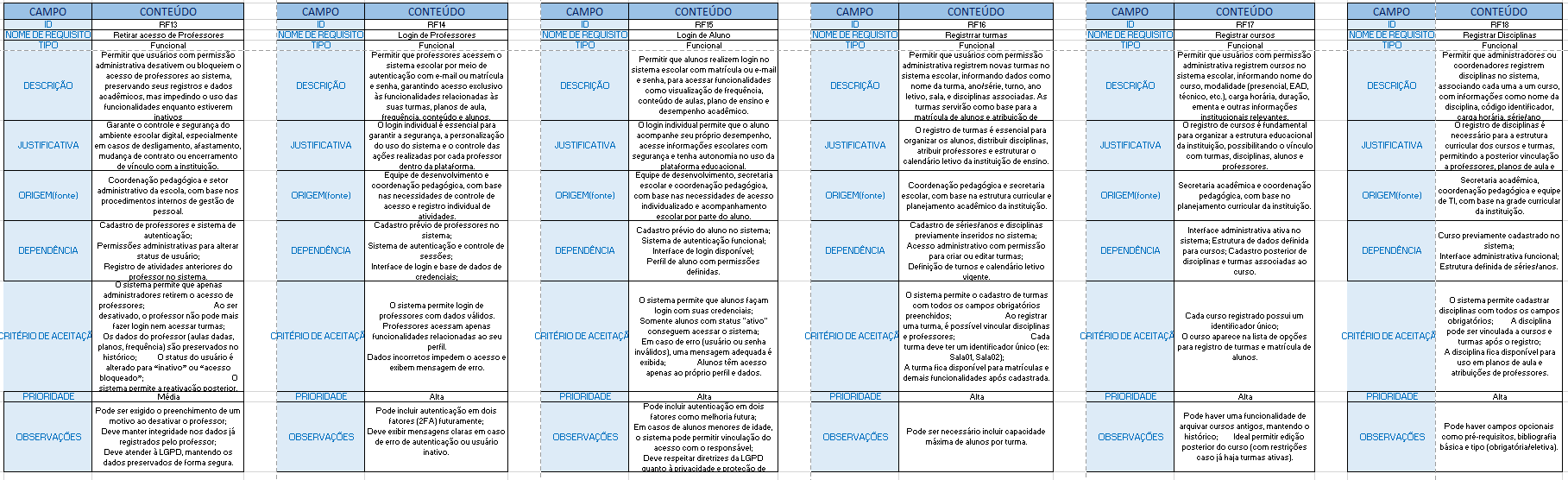
**

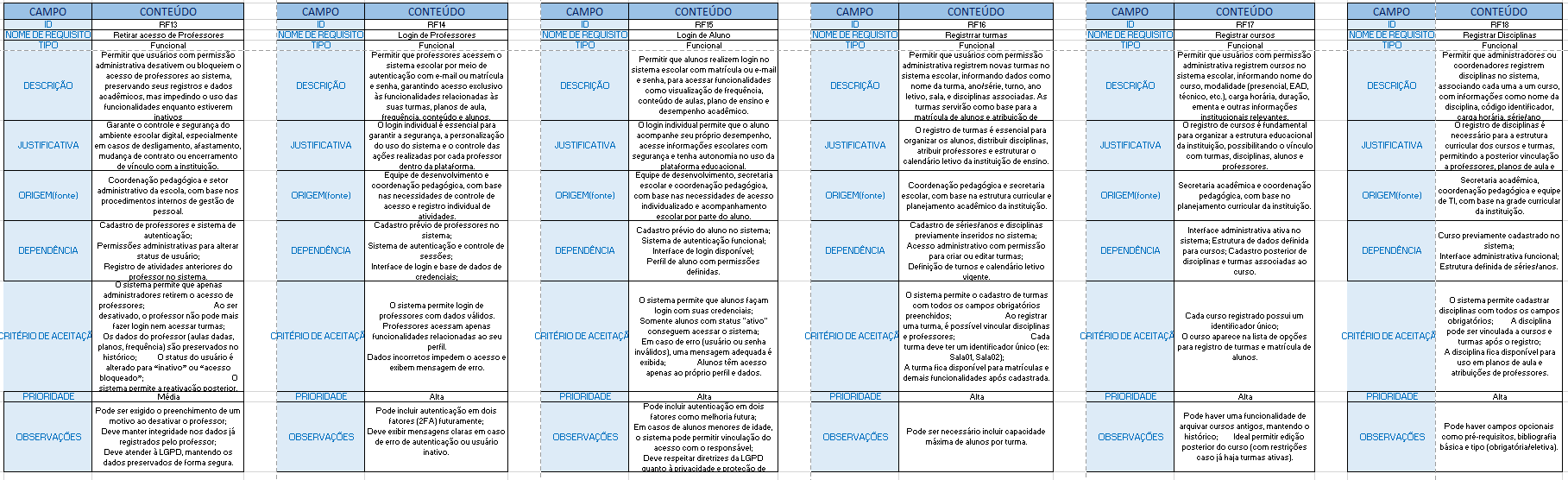
**

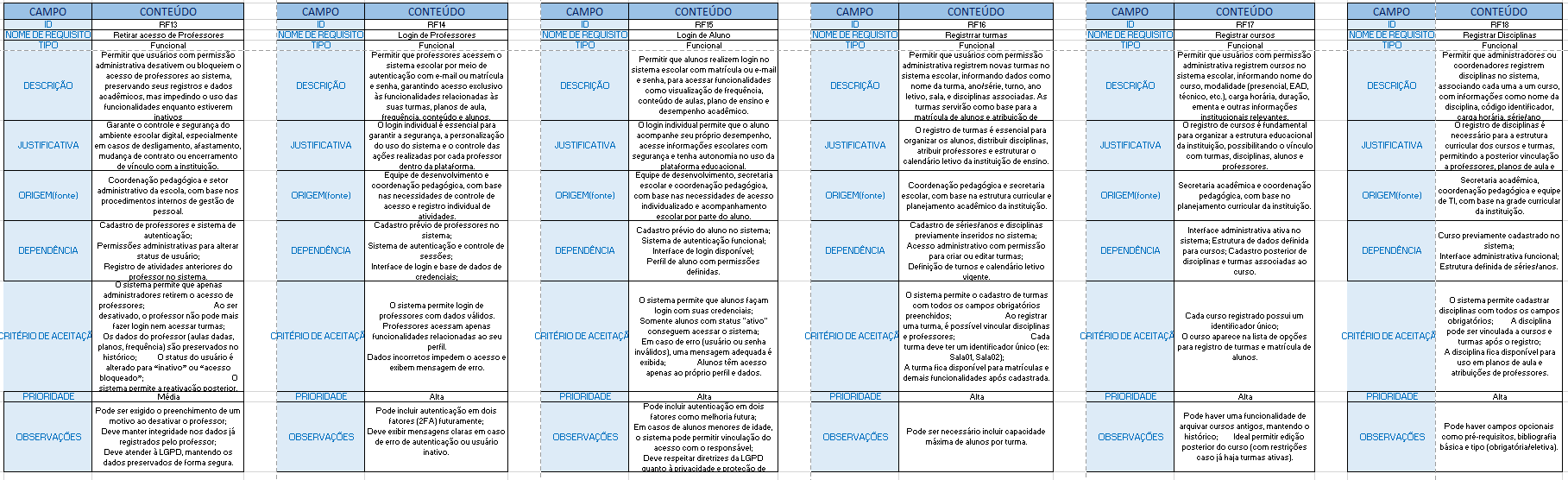
**

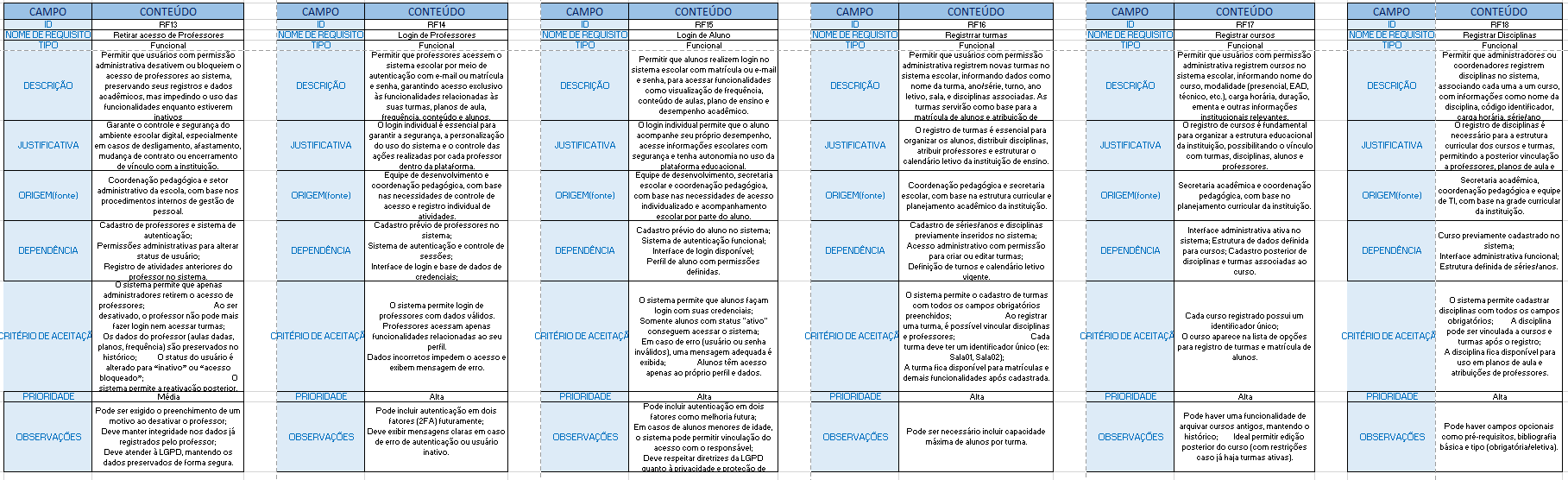
**

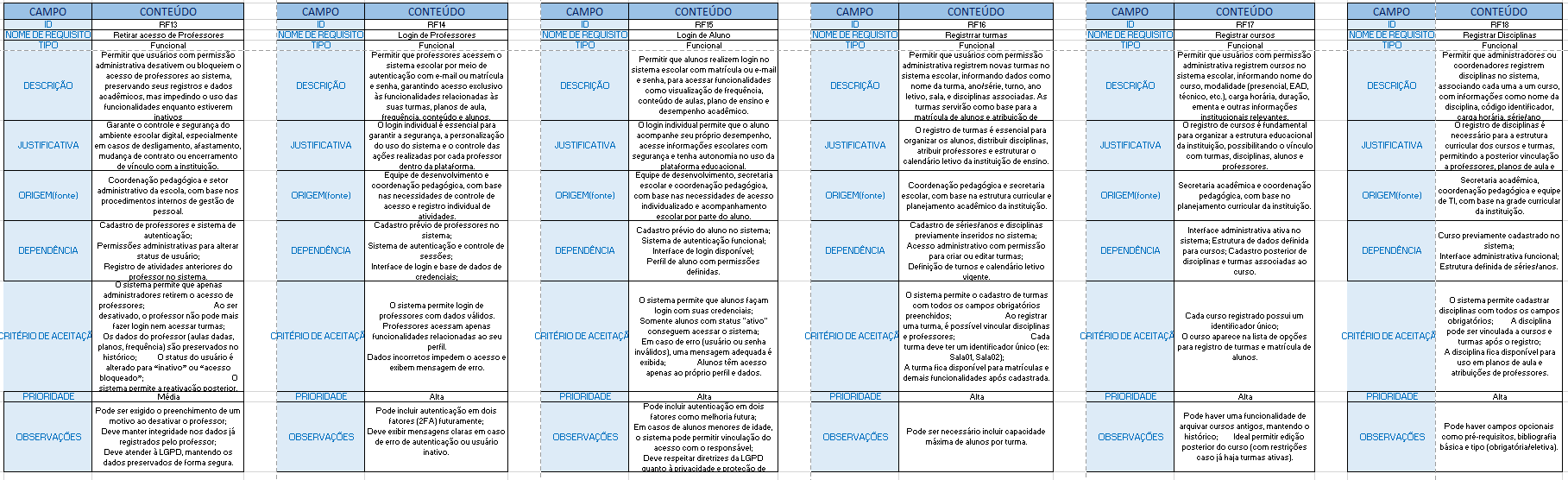
**

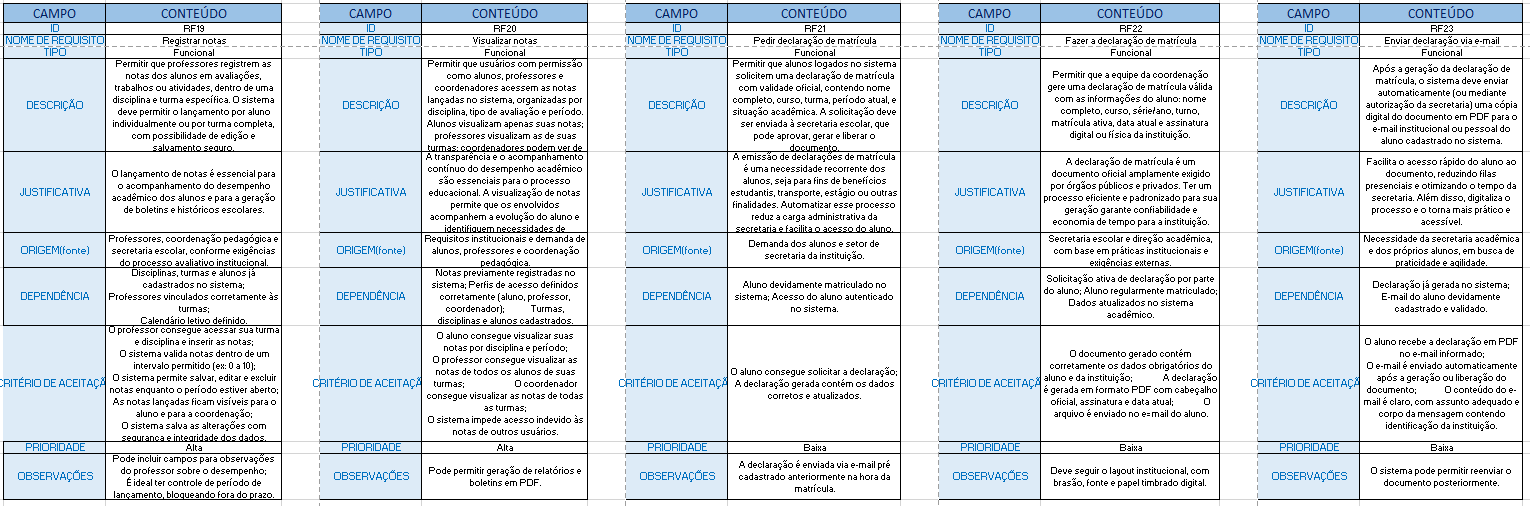
**

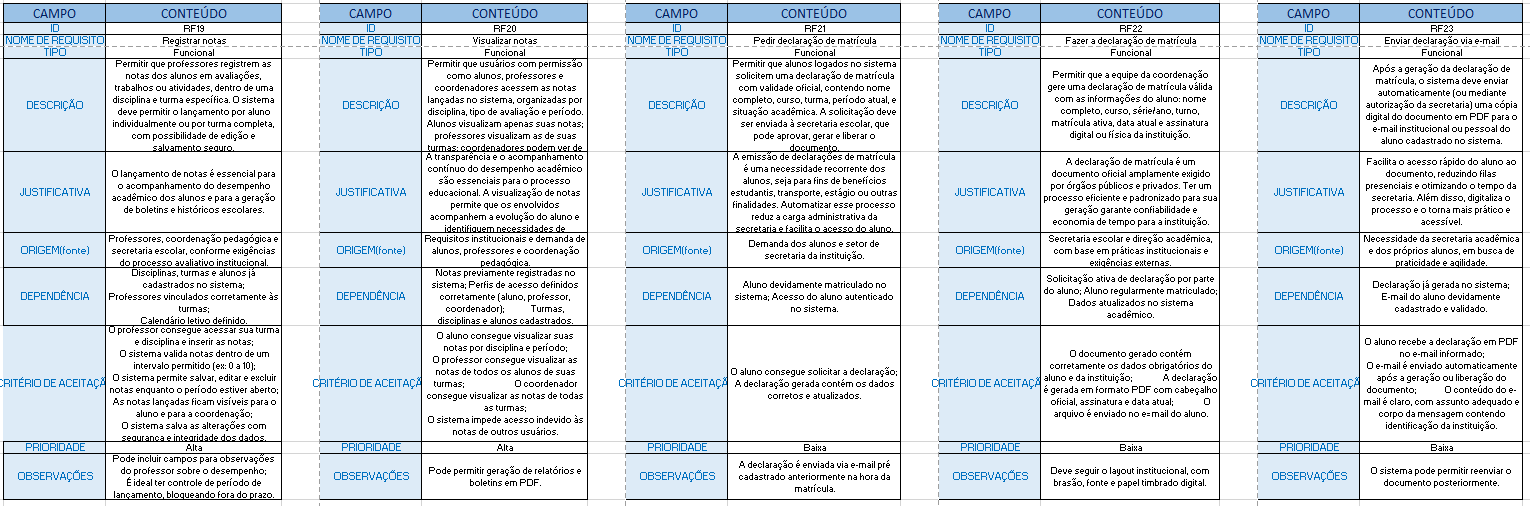
**

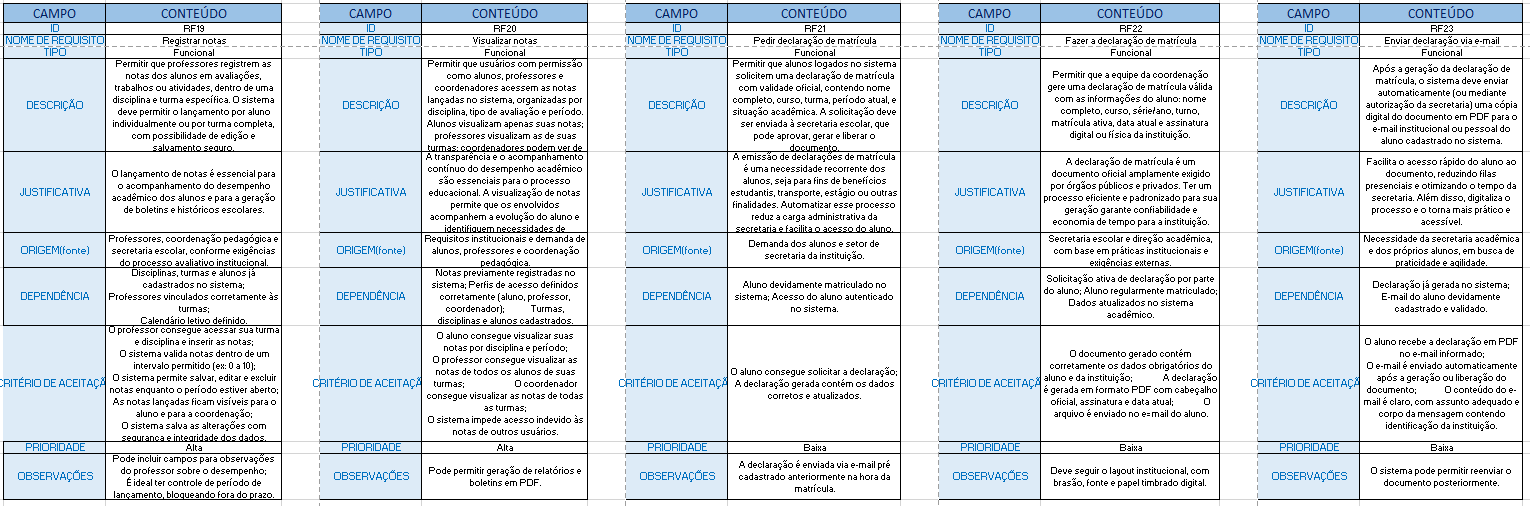
**

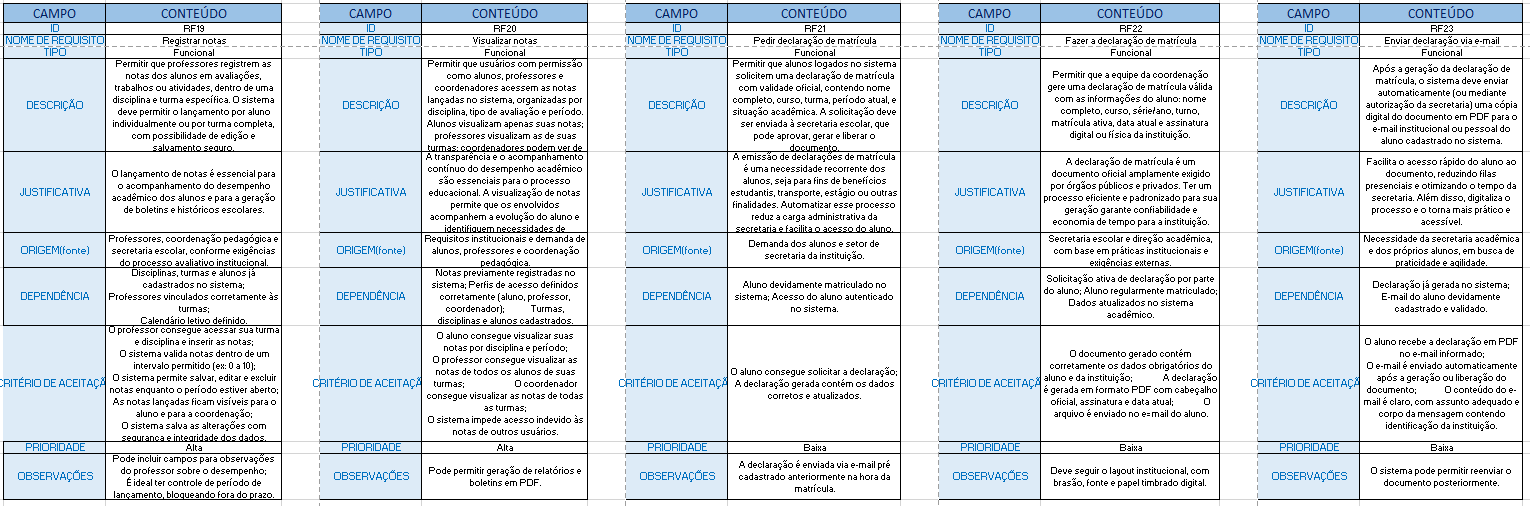
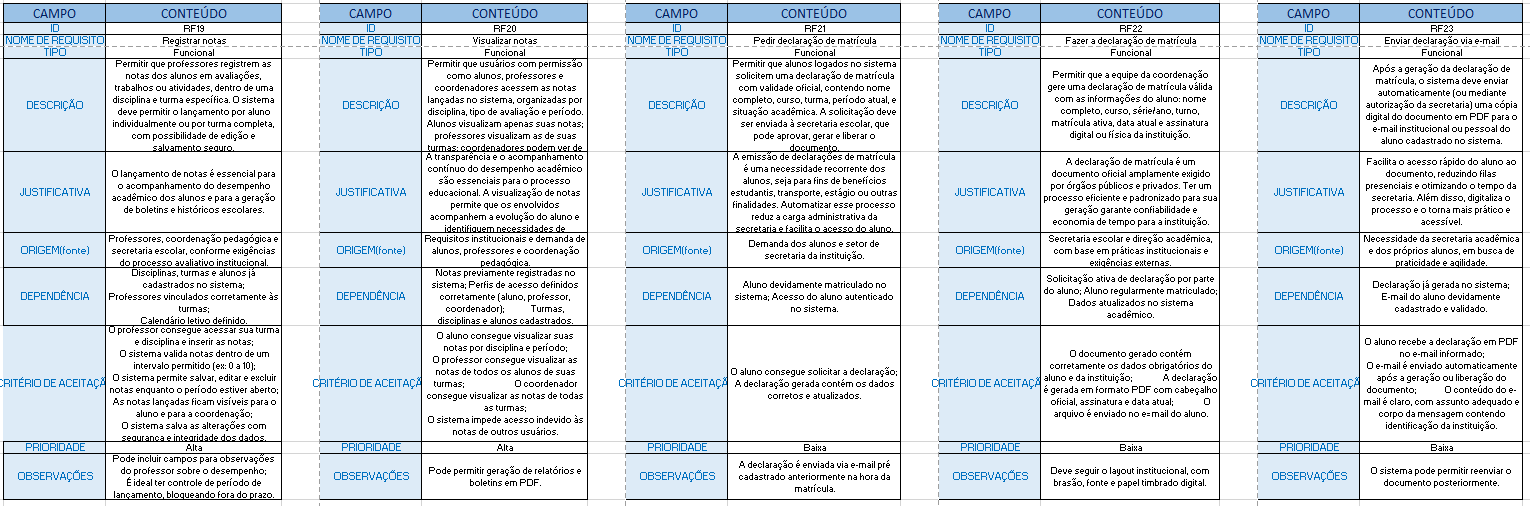
**

**

**

**

**

** **

#### **3.3. Requisitos Não Funcionais**

Requisitos não funcionais são aqueles que descrevem as características de qualidade e os critérios de desempenho, segurança, usabilidade e restrições técnicas que o sistema deve atender, sem estarem diretamente relacionados às funcionalidades específicas que ele executa. Enquanto os requisitos funcionais definem o que o sistema deve fazer, os requisitos não funcionais definem como o sistema deve se comportar ao realizar essas funções.

Esses requisitos são fundamentais para garantir que o sistema seja eficiente, seguro, confiável, acessível e escalável. Muitas vezes, são eles que determinam a aceitação do sistema pelos usuários, pois estão diretamente ligados à experiência de uso, à robustez da aplicação e à adequação às condições reais de operação.

Os requisitos não funcionais são utilizados como parâmetros para orientar decisões de arquitetura, escolha de tecnologias, estratégias de implementação e, posteriormente, para a realização de testes de desempenho e validação da qualidade do software. Eles devem ser especificados de forma clara e mensurável, para que possam ser verificados ao longo do processo de desenvolvimento e após a entrega do sistema. No sistema Alunus os requisitos não funcionais são os seguintes:

RFN01: O sistema deve estar disponível para uso 24 horas por dia, garantindo o acesso contínuo a alunos, professores e coordenadores.

RFN02: O sistema deve ser compatível com dispositivos móveis de baixa performance, visando atender escolas com infraestrutura tecnológica limitada.

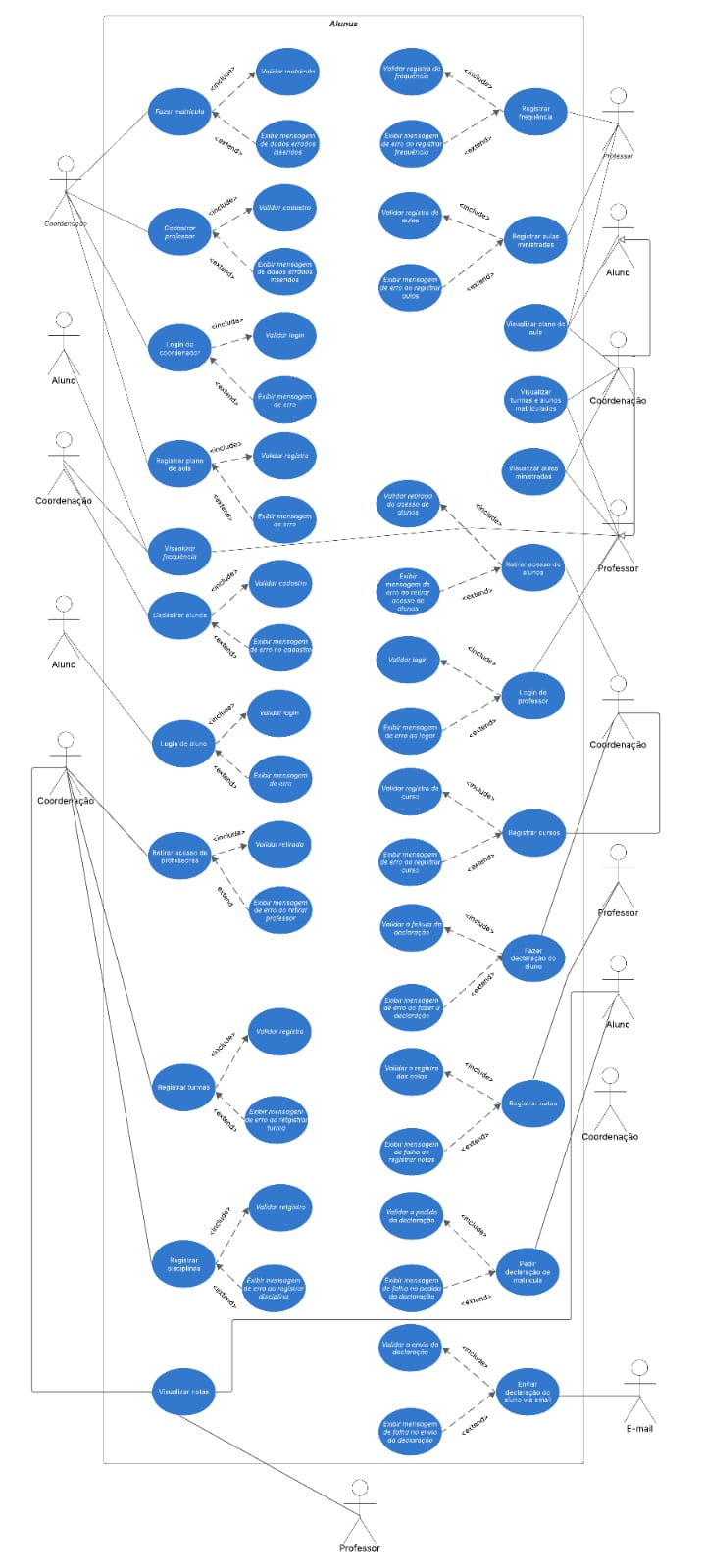
RFN03: As senhas dos usuários devem ser criptografadas, assegurando a proteção de dados sensíveis e a integridade das contas.

### **4. Modelagem de Casos de Uso**

#### **4.1. Diagrama de Casos de Uso (UML)**

O Diagrama de Casos de Uso é uma representação gráfica da interação entre os atores (usuários ou sistemas externos) e o sistema em desenvolvimento. Ele é uma ferramenta fundamental da linguagem UML e tem como principal objetivo ilustrar, de maneira clara e objetiva, as funcionalidades oferecidas pelo sistema e a forma como os usuários se relacionam com elas.

No contexto do sistema Alunus, o diagrama de casos de uso desempenha um papel essencial na modelagem dos requisitos funcionais, pois permite visualizar, em alto nível, quais são as principais operações que podem ser realizadas por cada perfil de usuário e como essas operações estão conectadas entre si. Segue o caso de uso do sistema Alunus:



#### **4.2. Especificação dos Casos de Uso**

**Caso de Uso: Fazer matrícula**

● Atores: Coordenação, Sistema

● Fluxo de Eventos Principal:

1. Usuário coordenador realiza login no software

2. A coordenação insere os dados do aluno para nova matricula

3. A coordenação seleciona o curso desejado

4. A coordenação seleciona a turma disponível

5. A coordenação confirma a matrícula do aluno

● Fluxo Alternativo:

1.1. Login incorreto: Caso o login do usuário coordenador estiver errado emitir mensagem de erro.

3.1 Curso não disponível: Caso o curso selecionado não estiver disponível para matrícula emitir mensagem de curso indisponível.

4.1 Turma cheia: Caso a turma escolhida esteja com lotação máxima exibir mensagem de turma cheia.

4.2 Caso turma cheia, verificar se possui outra turma disponível, caso sim voltar na escolha de turma; caso não, entrar na lista de espera e encerrar programa.

**Caso de Uso: Login de aluno professor e coordenador (Cada um deles tem um nível diferente de administração do site)**

● Atores: Aluno, Professor, Coordenador, Sistema

● Fluxo de Eventos Principal:

1. O usuário entra para fazer login no sistema

2. O usuário insere seu e-mail e senha

3. O sistema valida as informações

4. O usuário entra no programa de acordo com seu nível de permissão

● Fluxo Alternativo:

2.1 Dados Incorretos: O usuário insere seus dados e caso haja um erro, recebe uma mensagem de dados incorretos

3.1 Usuário Coordenador: o sistema verifica os dados inseridos e verifica se é um coordenador

3.2 Usuário Professor: O sistema verifica os dados inseridos e verifica se é de um professor

3.3 Usuário Aluno: O sistema verifica os dados inseridos e verifica se é de um aluno

**Caso de Uso: Registrar frequência**

● Atores: Professor, Sistema

● Fluxo de Eventos Principal:

1. Professor faz login no software

2. O professor acessa a turma que deseja registrar chamada

3. O professor seleciona a data da aula

4. O professor registra a presença para o aluno

5. o professor verifica se a lista toda já foi registrada

6. O professor envia a frequência da turma

● Fluxo Alternativo:

1.1 Dados Incorretos: O usuário insere seus dados e caso haja um erro, recebe uma mensagem de dados incorretos

3.1 A data está dentro do intervalo informado, se não estiver volta para a ação anterior

4.1 Aluno ausente: caso o aluno não esteja presente selecione falta

5.1 Caso a lista não estiver completa volta para a ação registrar frequência para o aluno

**Caso de Uso: Registrar plano de aula**

● Atores: Coordenação

● Fluxo de Eventos Principal:

1. A coordenação acessa a plataforma

2. Seleciona a turma

3. Seleciona a turma que deseja inserir o plano de aula

4. Seleciona a data para inserir o Plano de Aula

5. Insere o Plano de Aula da Disciplina

6. A coordenação salva e confirma

● Fluxo Alternativo:

1.1 Erro de login: Caso erro de login emitir mensagem de erro

4.1 Caso a data não estiver dentro do intervalo informado, retornar a ação anterior

**Caso de Uso: Registrar aulas ministradas**

● Atores: Professor, Sistema

● Fluxo de Eventos Principal:

1. Professor faz login

2. O professor acessa o calendário, e seleciona a data

3. Professor seleciona a turma

4. Seleciona a disciplina

5. O professor insere o conteúdo ministrado

6. O professor, salva e confirma

7. O sistema valida e atualiza o calendário

● Fluxo Alternativo:

1.1 Caso login inválido emitir mensagem de erro

2.1 O professor seleciona a data, caso data fora do intervalo voltar para o passo anterior e selecionar a data correta.

6.1 Esquecer de salvar. Se o professor não salvar, o sistema irá exibir uma mensagem de erro, lembrando-o de salvar o documento

**Caso de Uso: Registrar turmas**

● Atores: Coordenação

● Fluxo de Eventos Principal:

1. Coordenação faz login

2. A coordenação acessa a lista de alunos matriculados

3. A coordenação separa os alunos em turmas

4. A coordenação salva as turmas

5. O coordenador confirma as turmas

● Fluxo Alternativo:

1.1 Caso login inválido emitir mensagem de erro

2.1. Número incorreto de alunos por sala e esquecer de salvar

4.1 Caso haja erro: emitir mensagem de erro e coordenador deve revisar as turmas

**Caso de Uso: Retirar acesso de alunos/professores**

● Atores: Coordenação, Sistema

● Fluxo de Eventos Principal:

1. Coordenação faz login

2. A coordenação acessa uma lista de aluno matriculados e professores cadastrados

3. A coordenação escolhe quem irá retirar

4. O sistema pergunta se a coordenação tem certeza da ação que será realizada (Sim/Não)

5. A coordenação clica sim

6. O sistema valida a escolha e atualiza a lista de alunos/professores

● Fluxo Alternativo:

1.1 Caso login inválido emitir mensagem de erro

4.1. A coordenação opta por não quando o sistema pergunta se é certeza

1. Se a escolha for não, o sistema não remove ninguém da lista.

2. O caso de uso retorna para o passo 3.

**Caso de Uso: Visualizar frequência / plano de aula / notas**

● Atores: Coordenação, Professor, Aluno

● Fluxo de Eventos Principal:

1.Usuário faz login

2.Usuário escolhe a turma desejada

3.Usuário escolhe a disciplina desejada

4.Usuário visualiza as notas escolhida

5.Usuário visualiza o plano de aula da disciplina

6.Usuário visualiza a frequência desejada

● Fluxo Alternativo:

1.1 Erro no login: Caso possua erro emitir mensagem

**Caso de Uso: Visualizar aulas ministradas**

● Atores: Coordenação, Professor, Sistema

● Fluxo de Eventos Principal:

1. Usuário faz login

2. O usuário abre o calendário de aulas

3. O sistema exibe todas as datas com as aulas ministradas de cada dia

● Fluxo Alternativo:

1.1 Erro no login: Caso possua erro emitir mensagem

**Caso de Uso: Cadastrar professor / aluno**

● Atores: Coordenador

● Fluxo de Eventos Principal:

1. Coordenação faz login

1. O usuário insere as informações exigidas para o cadastro

2. O usuário confirma os dados

3. O usuário envia o cadastro

● Fluxo Alternativo:

1.1 Erro no login: Caso possua erro emitir mensagem

3.1 Se os dados enviados estão incorretos ou fora do devido formato, o sistema exibe uma mensagem de erro nos dados

### **5. Diagrama de Atividades (UML)**

Diagrama de Atividades referente à FAZER MATRÍCULA:

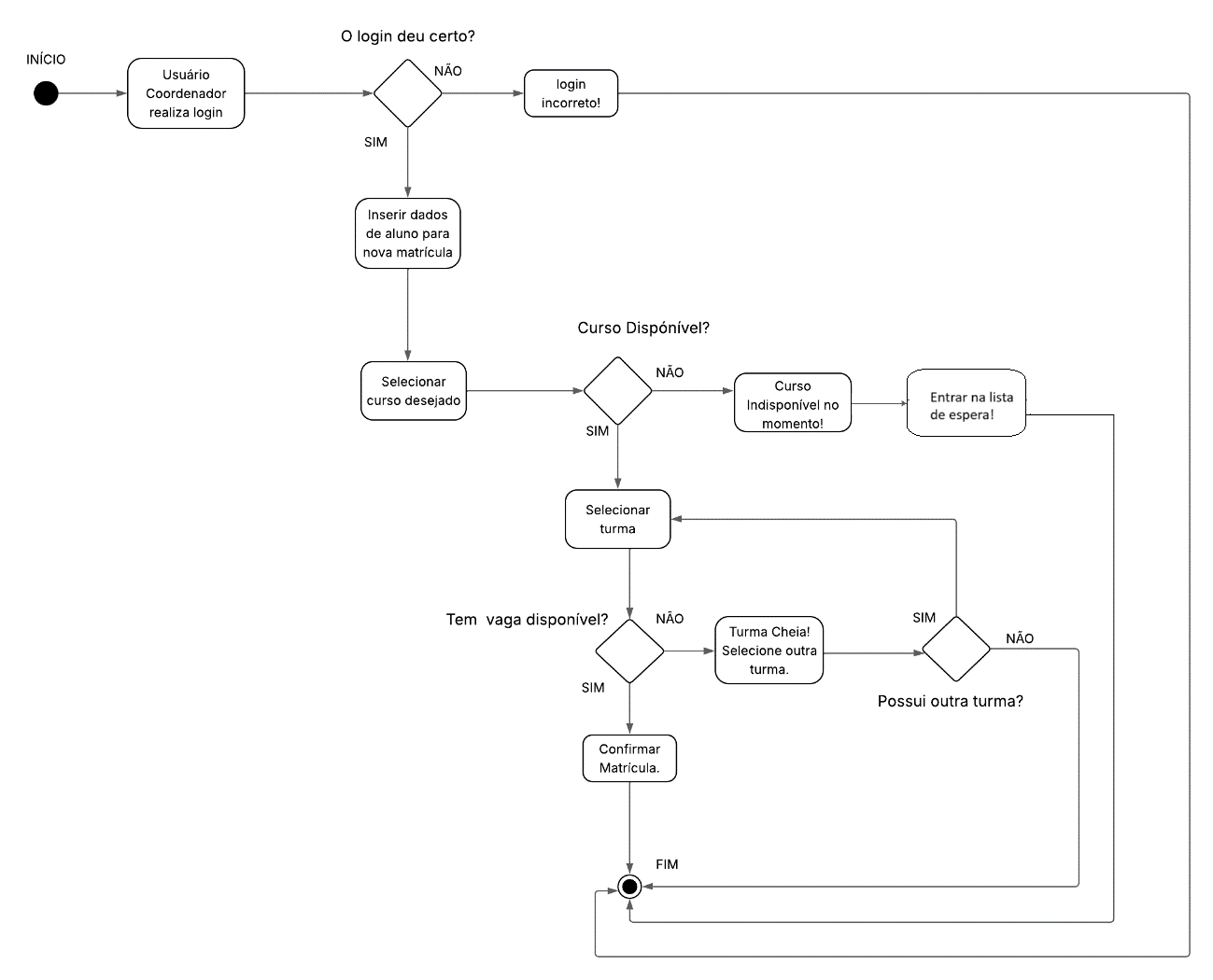


Diagrama de Atividades referente à LOGIN DE USUÁRIOS:

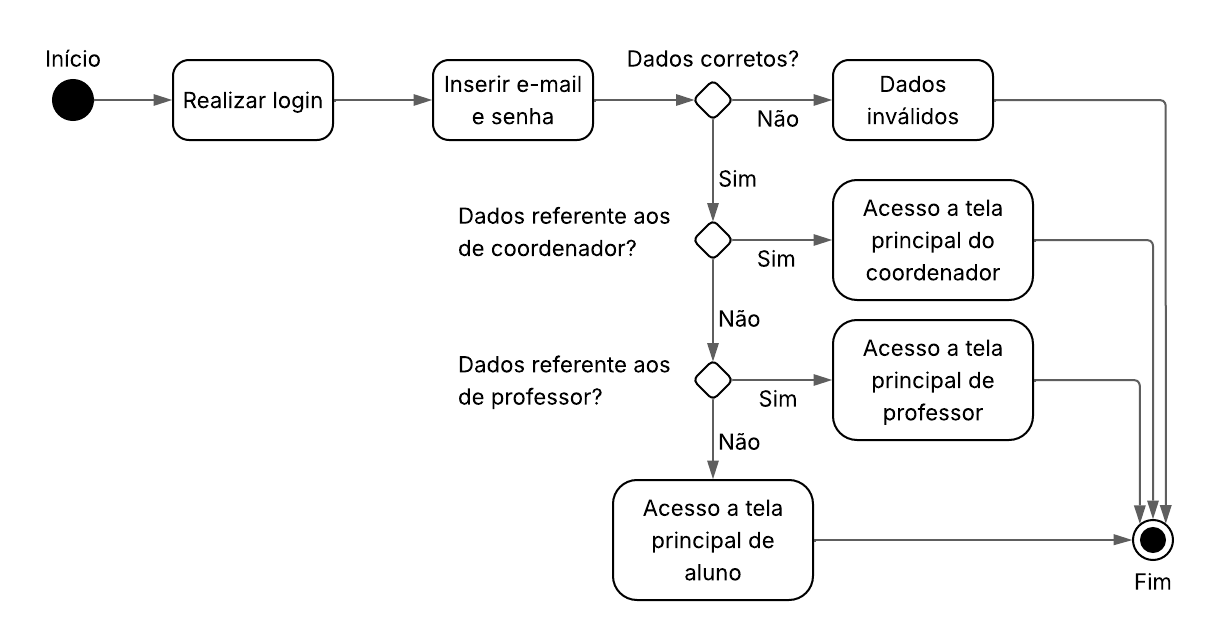


Diagrama de atividades referente à REGISTRAR FREQUÊNCIA:

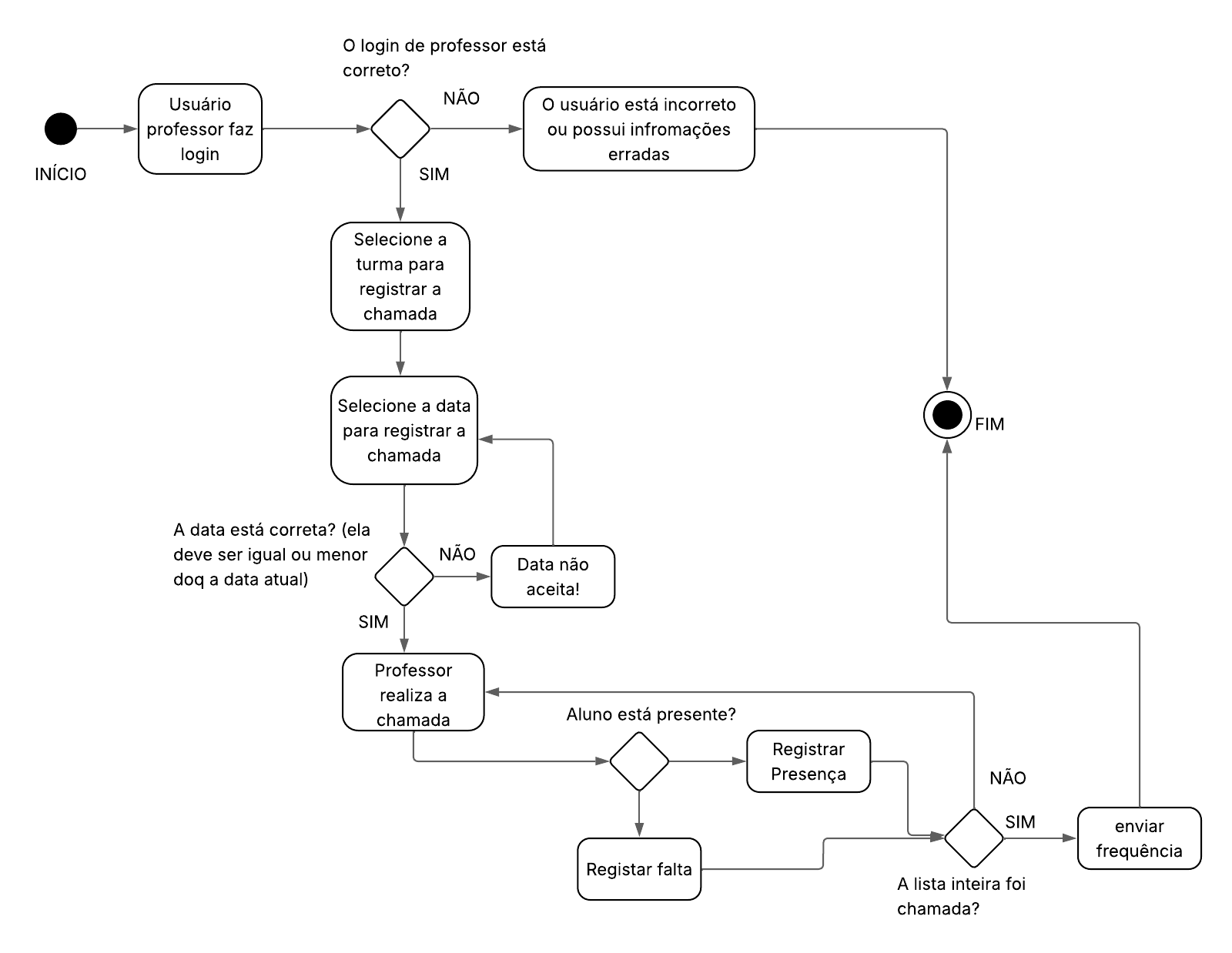


Diagrama de atividades referente à REGISTRAR PLANO DE AULA:

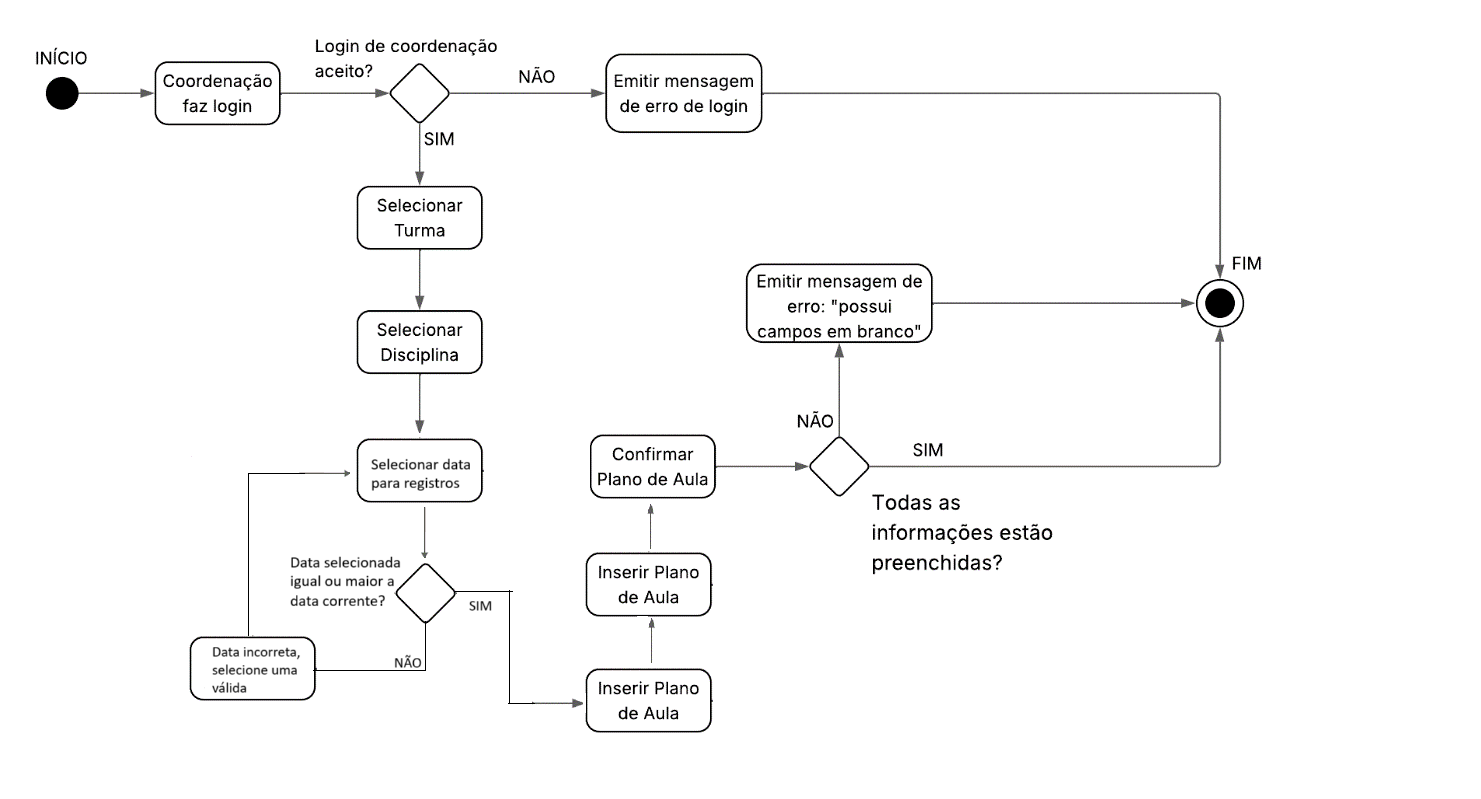


Diagrama de atividades referente à REGISTRAR AULAS MINISTRADAS:

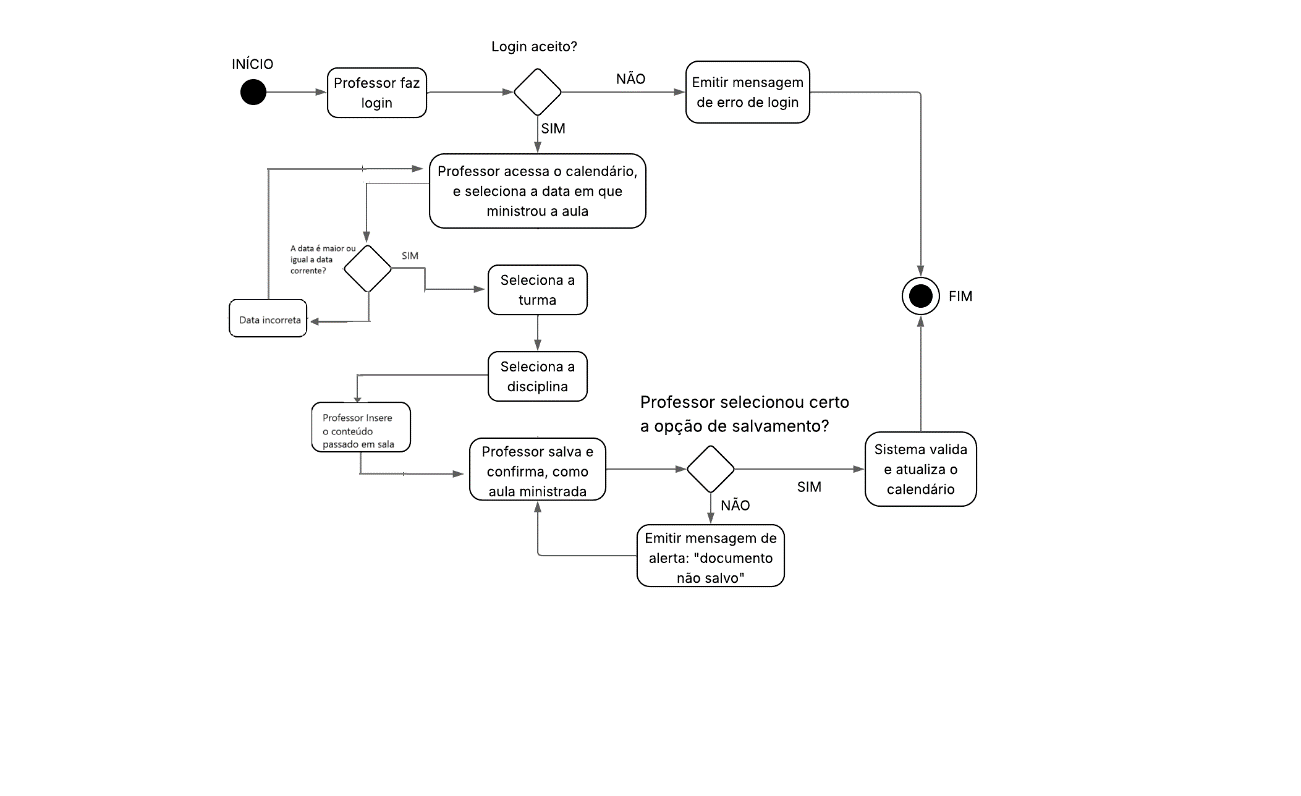


Diagrama de atividades referente à REGISTRAR TURMAS:

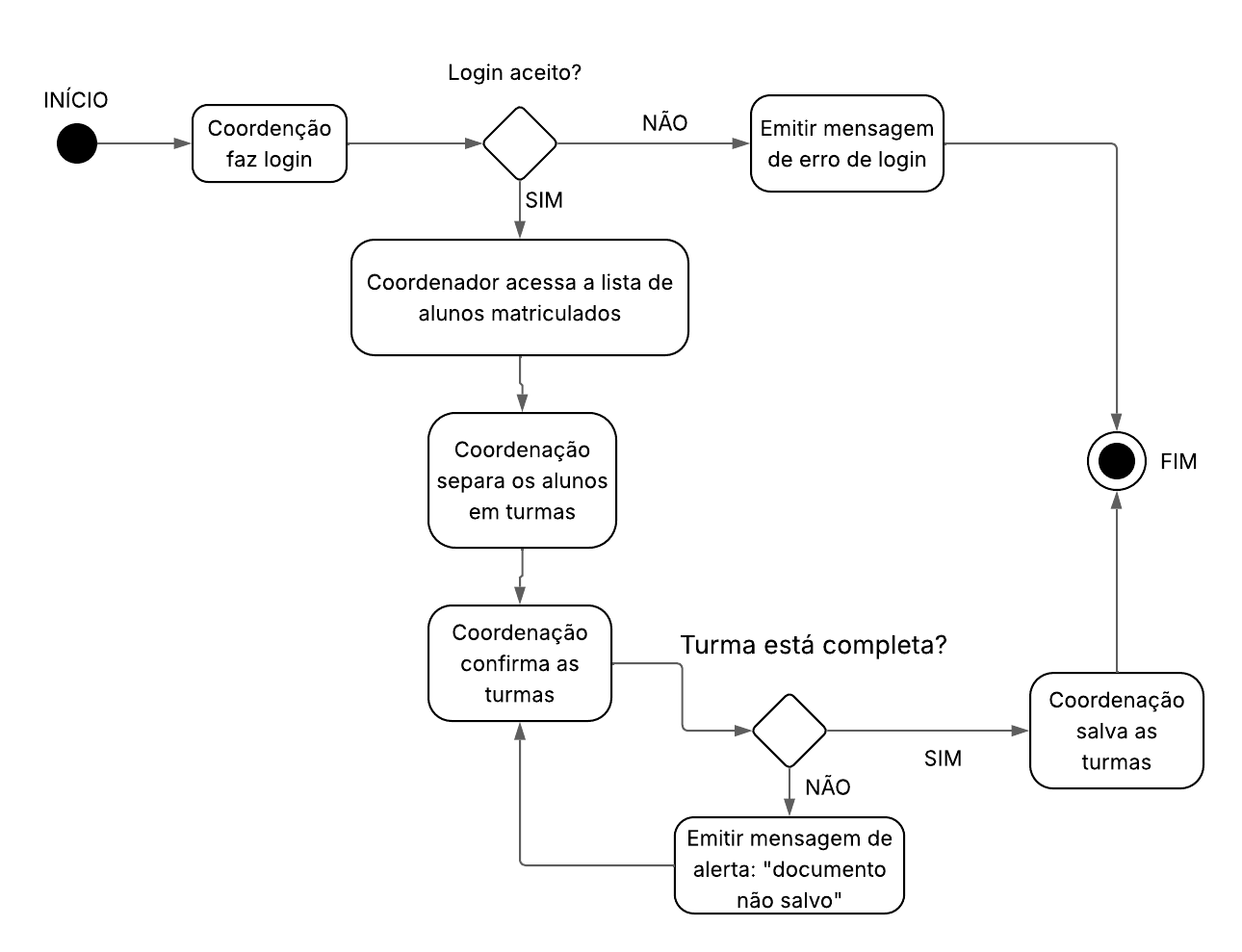


Diagrama de Atividades referente à RETIRAR ACESSO DE ALUNOS/PROFESSORES:

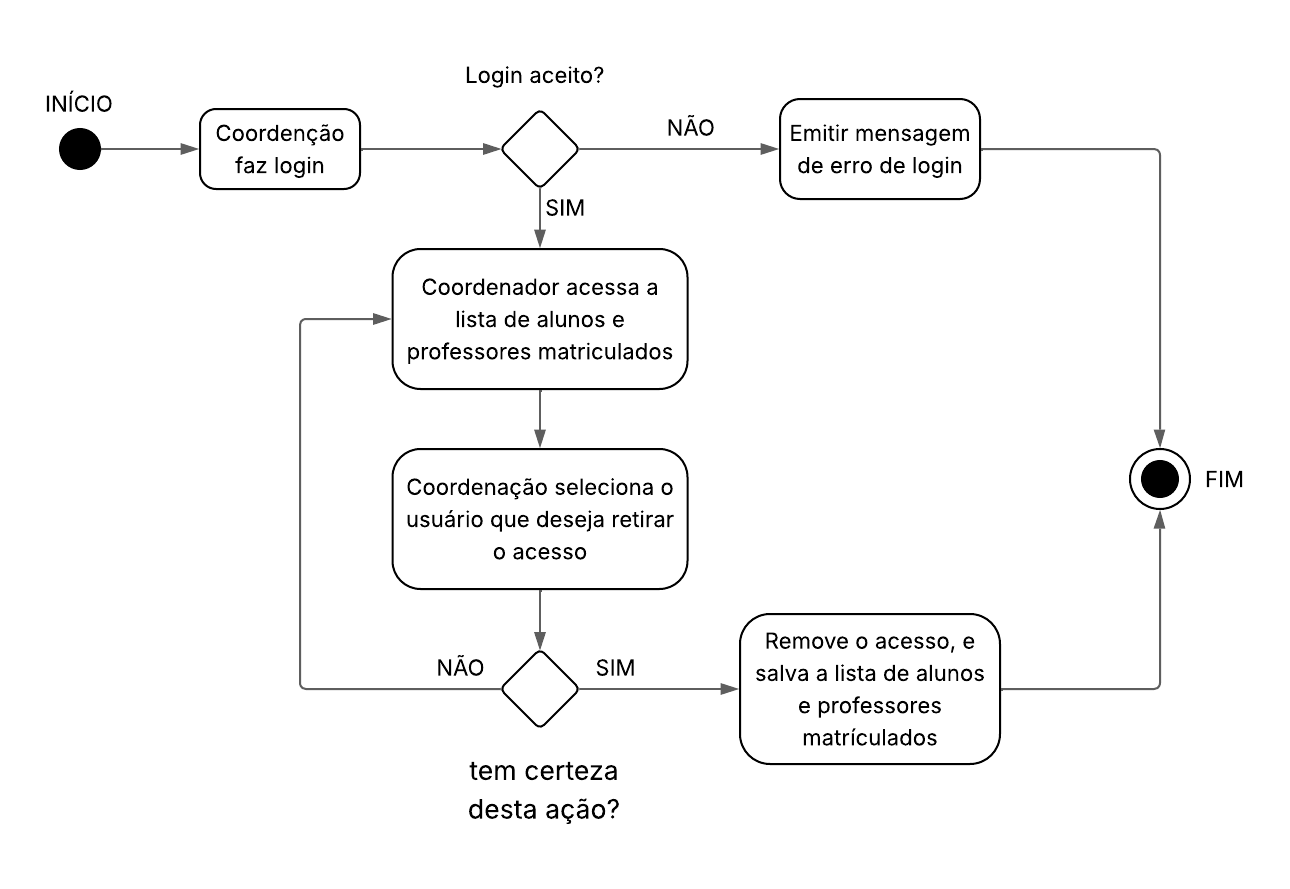


Diagrama de Atividades referente à VISUALIZAR FREQUÊNCIA / PLANO DE AULA / NOTAS:

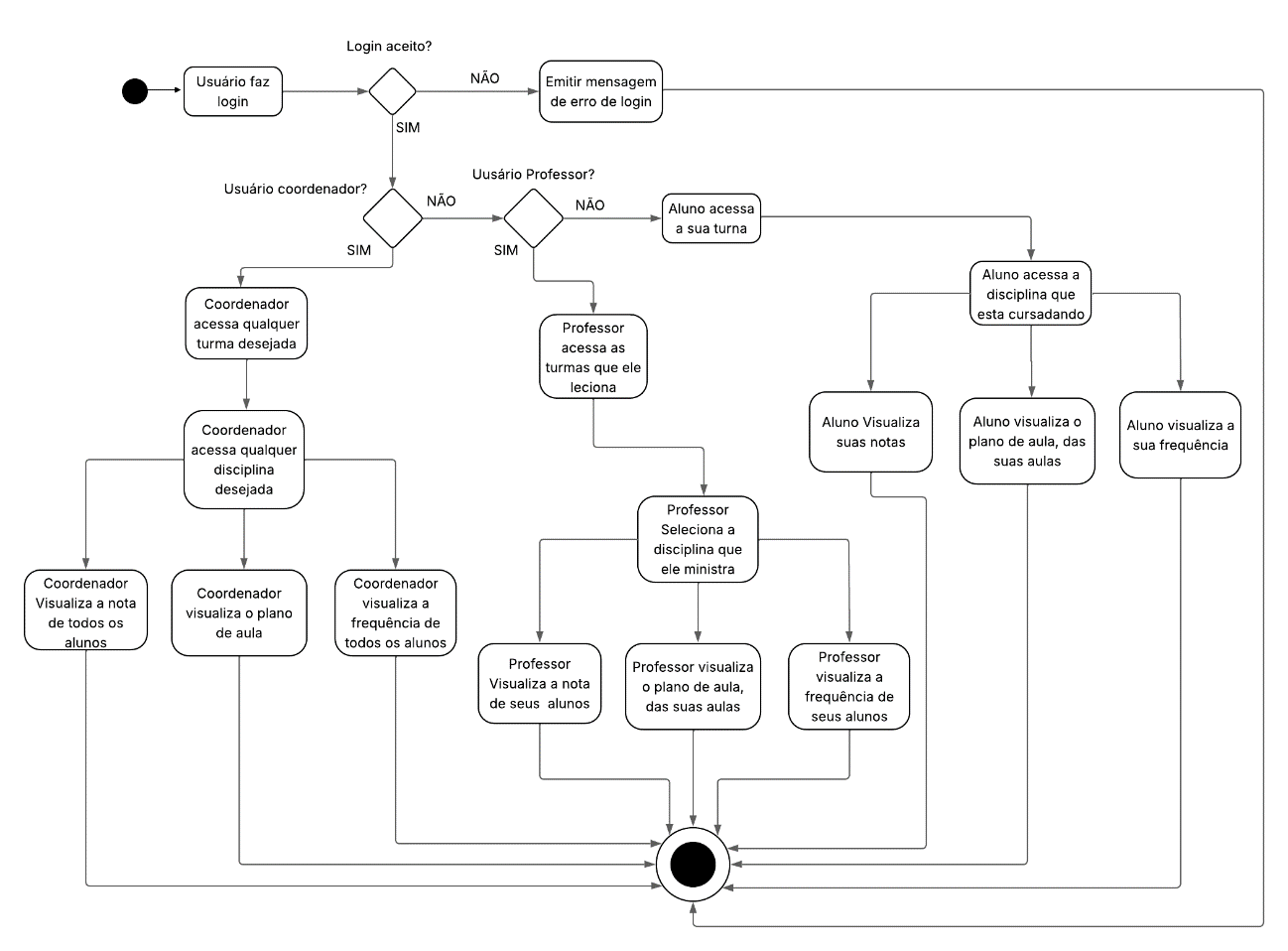


Diagrama de Atividades referente à VISUALIZAR AULAS MINISTRADAS:

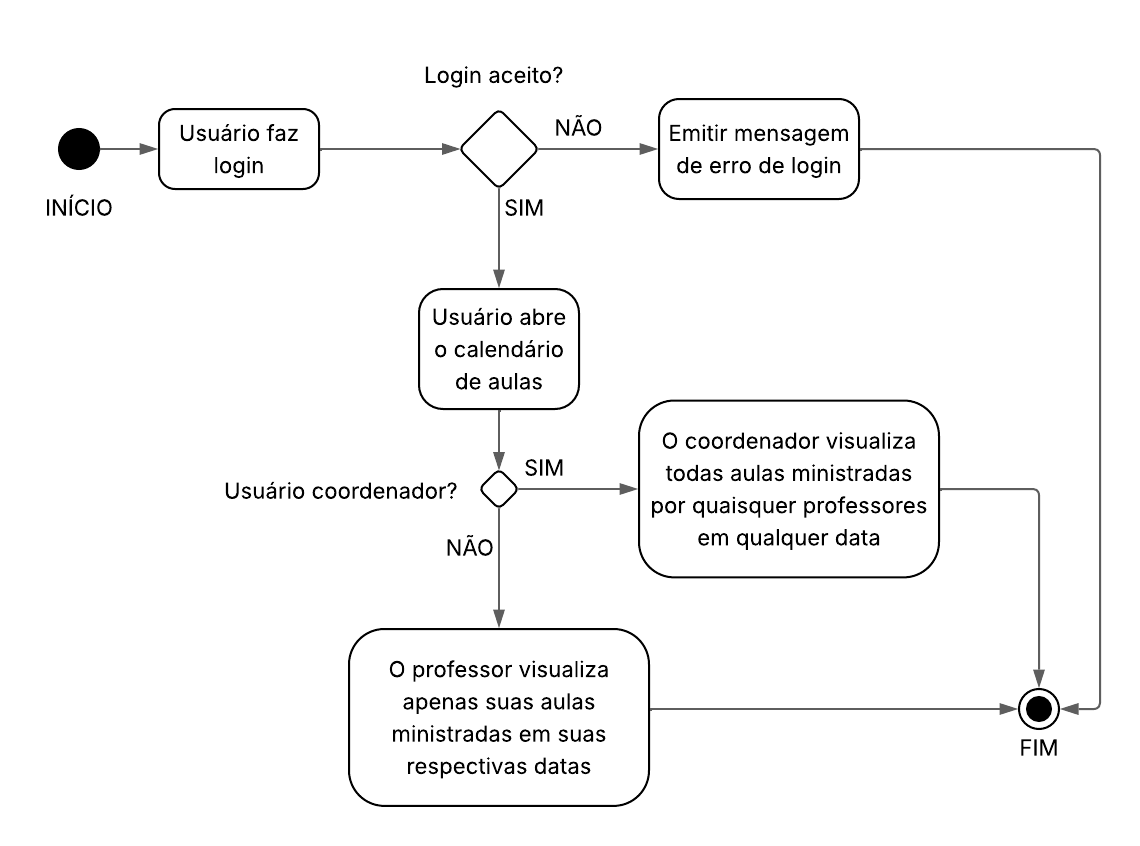
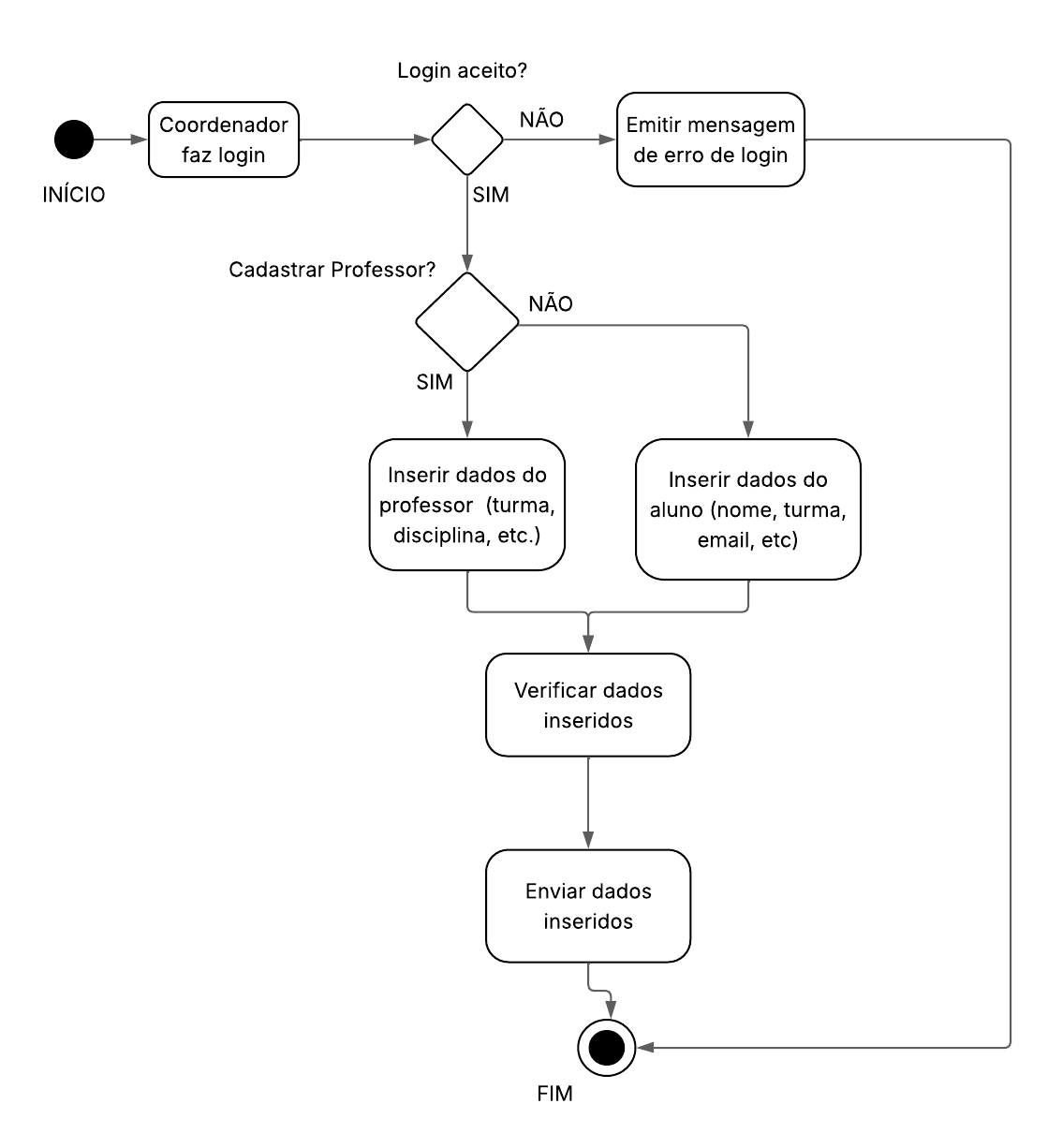


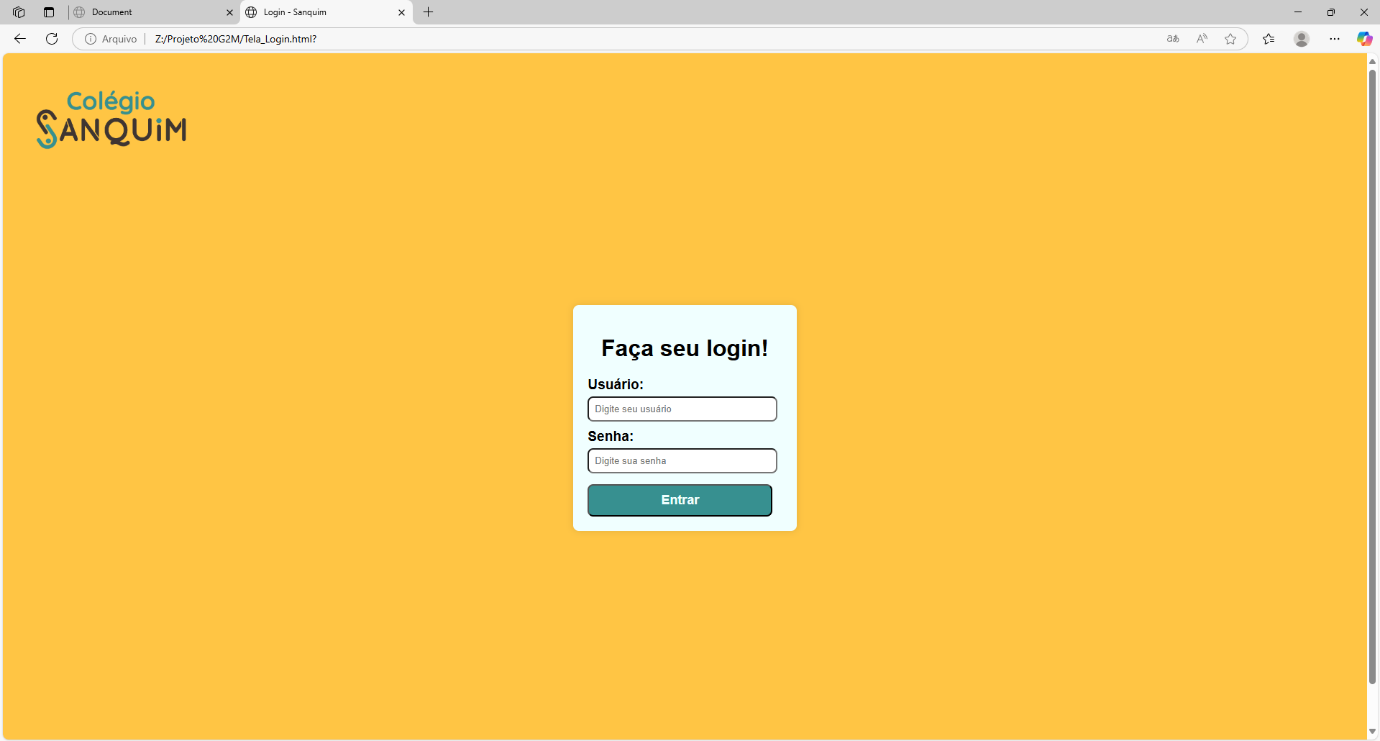
Diagrama de Atividades referente à CADASTRAR PROFESSOR / ALUNO:



### **6. Protótipos das Telas**

Os protótipos das telas têm como objetivo ilustrar visualmente a interface do usuário do sistema Alunus, permitindo uma melhor compreensão de como o sistema será utilizado na prática. Embora não representem ainda o produto final, os protótipos servem como uma ferramenta de comunicação entre os desenvolvedores, os stakeholders e os usuários, facilitando a validação da estrutura e da disposição dos elementos nas telas antes da implementação definitiva.

TELA DE LOGIN:



TELA PRINCIPAL COORDENADOR:



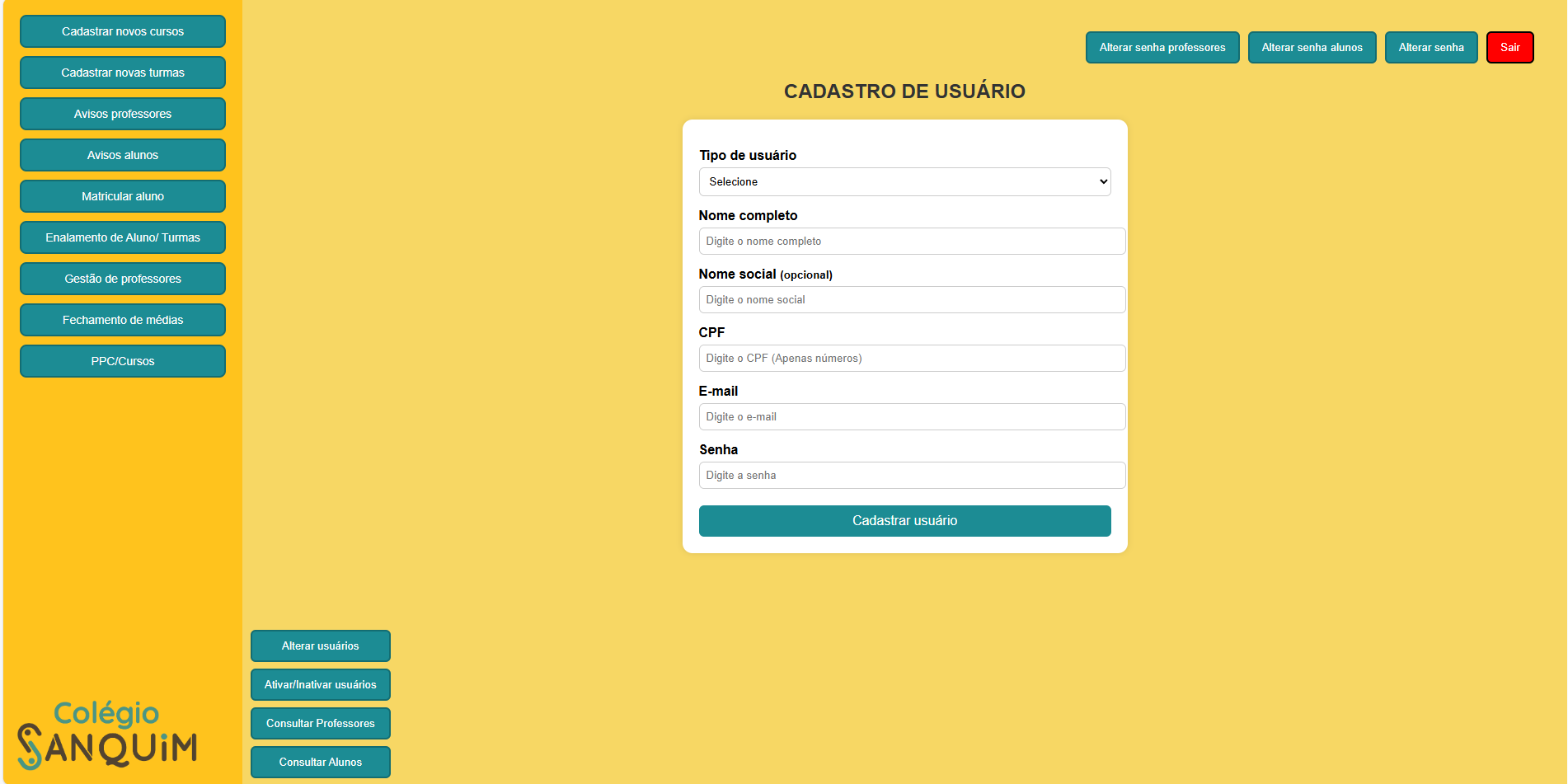
TELA PRINCIPAL PROFESSOR:



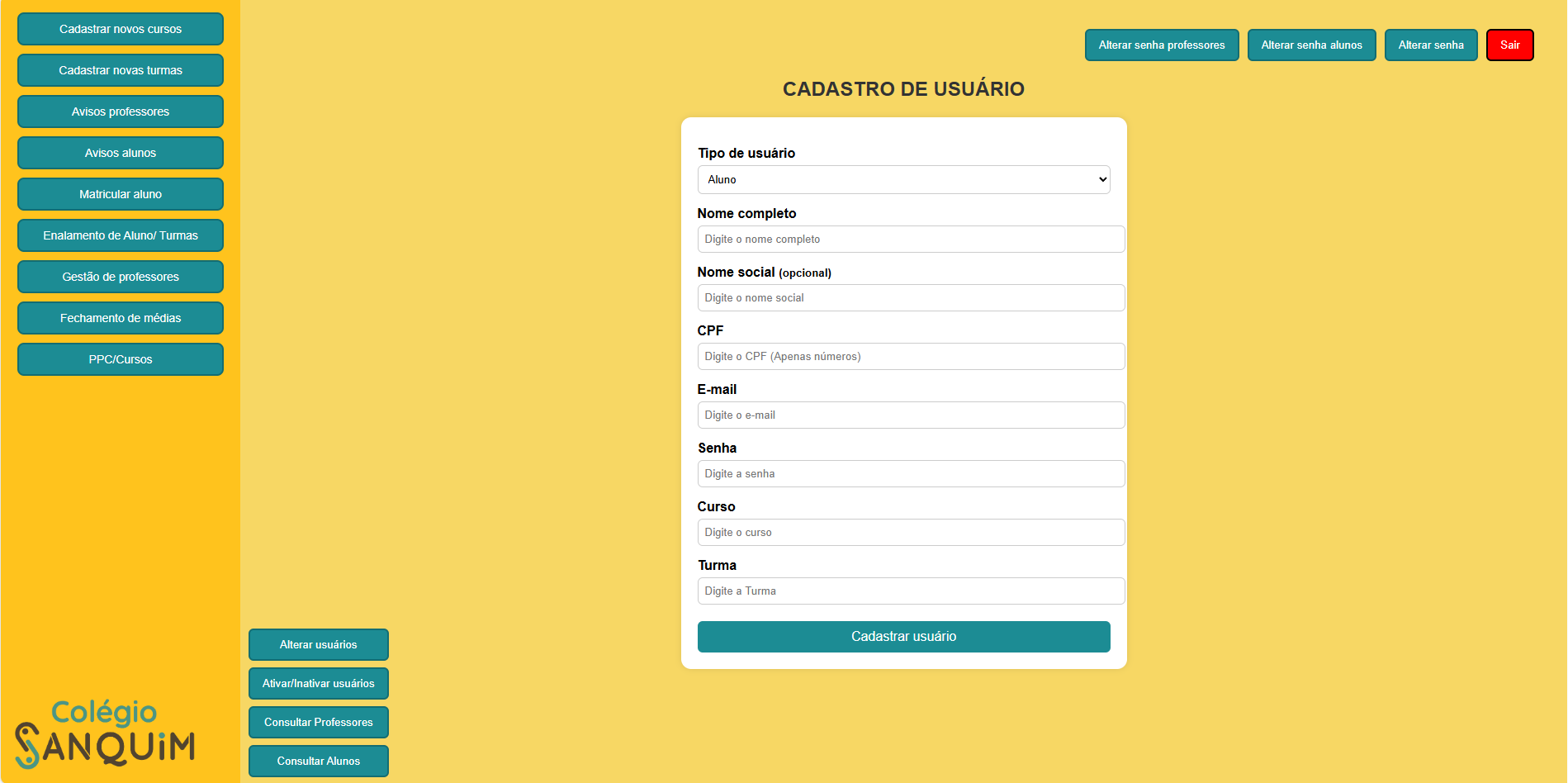
TELA PRINCIPAL ALUNO:



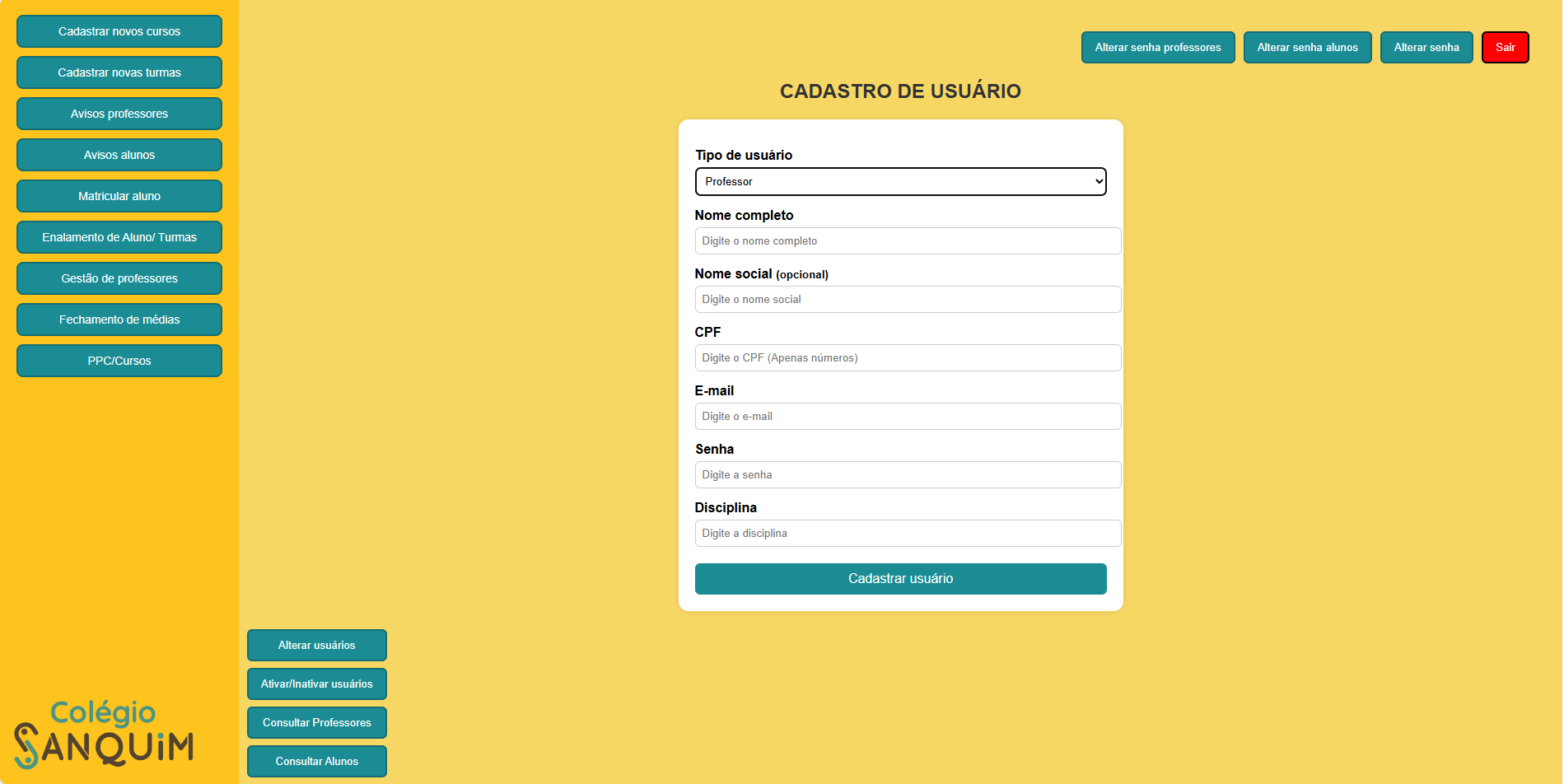
TELA DE CADASTRO USUÁRIO:



TELA DE CADASTRO ALUNO:



TELA DE CADASTRO PROFESSOR:



### **7. Considerações Finais**

Ao longo do desenvolvimento da especificação do sistema Alunus, aprendemos bastante sobre como organizar as ideias de um projeto de software e como registrar essas informações de forma clara. Foi importante entender melhor o papel dos requisitos e como eles influenciam todas as etapas do desenvolvimento. Também percebemos como os diagramas e os protótipos ajudam muito a visualizar o funcionamento do sistema antes mesmo de começar a programar.

Durante esse processo, enfrentamos alguns desafios, principalmente na hora de decidir o que era mais importante para o sistema e como adaptar tudo para funcionar bem em dispositivos mais simples. Outra limitação foi o tempo curto que tivemos para modelar todos os fluxos com mais profundidade, então alguns pontos ainda vão precisar ser ajustados mais pra frente.

Entre as decisões tomadas, optamos por deixar o sistema o mais simples e acessível possível, tanto em termos de uso quanto de tecnologia, já que o foco são escolas com pouca estrutura. Decidimos também separar bem as funções de cada tipo de usuário, como aluno, professor e coordenador, para facilitar o uso no dia a dia. É necessário realizar algumas mudanças e melhorias no futuro para que o sistema Alunus fique ainda melhor,