### **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:** As Front\_Furious

### **1. Introdução**

#### **1.1. Objetivo do Documento**

Este documento tem como objetivo descrever de forma clara os requisitos do sistema de controle acadêmico, suas funcionalidades, restrições e representações visuais, visando a criação de um software que facilite a gestão acadêmica em uma escola comunitária que oferece cursinhos gratuitos. O sistema automatiza os processos de matrícula, controle de frequência, planejamento de aulas e outras funcionalidades, tornando a gestão escolar mais organizada, prática e eficiente.

### 

### **2. Visão Geral do Sistema**

#### **2.1. Visão do Produto**

O **Acadêmico App** é uma aplicação web que visa melhorar a gestão acadêmica de uma escola comunitária que oferece cursinhos gratuitos. O sistema foi projetado para automatizar processos críticos, como matrícula de alunos, controle de frequência, planejamento de aulas e alocação de professores, que atualmente são feitos manualmente.

As principais funcionalidades do **Acadêmico App** incluem:

* **Cadastro de Alunos**: Permite registrar as informações pessoais e acadêmicas dos alunos, facilitando a matrícula e o controle de dados.
* **Registro de Frequência**: Os professores podem registrar facilmente a frequência dos alunos, controlando presenças e faltas para cada aula.
* **Cadastro de Professores**: Permite a gestão de professores voluntários, incluindo seus dados e a alocação nas turmas.
* **Cadastro de Planos de Aula**: Auxilia no planejamento das aulas, registrando datas, temas, objetivos e materiais de apoio.
* **Histórico de Aulas Ministradas**: Permite registrar todas as aulas ministradas, criando um histórico de atividades escolares.
* **Alocação de Professores**: Facilita a distribuição de professores nas turmas conforme sua disponibilidade e as disciplinas que ministram.

**Diferenciais**:

O **Acadêmico App** oferece uma solução centralizada para gestão escolar, eliminando a necessidade de múltiplos sistemas ou métodos manuais.

Possui uma interface intuitiva e acessível, especialmente desenhada para um público não técnico, como professores e gestores escolares.

O sistema facilita a transparência e o acompanhamento em tempo real do progresso acadêmico de cada aluno, algo que não é facilmente acessível com o controle manual.

#### **2.2. Justificativa**

O desenvolvimento do Acadêmico App é motivado pela necessidade de otimizar e automatizar os processos manuais que atualmente são feitos na gestão acadêmica da escola comunitária. Atualmente, a instituição enfrenta desafios significativos relacionados à organização e ao controle das informações dos alunos e professores, como:

* Falta de Organização: As matrículas, registros de frequência e planejamento de aulas são realizados manualmente, o que aumenta as chances de erros e desorganização.
* Processos Lentos e Ineficientes: O registro de informações em papel ou planilhas exige muito tempo e esforço, o que causa atrasos no acompanhamento do desempenho dos alunos e na alocação de professores.
* Dificuldade no Acompanhamento de Dados: Sem um sistema centralizado, o acompanhamento da frequência dos alunos e o histórico de aulas ministradas são dispersos, dificultando a análise de desempenho e o planejamento de ações pedagógicas.

Por esses motivos, o Acadêmico App visa centralizar e simplificar esses processos, oferecendo uma solução mais organizada e eficiente.

#### **2.3 Objetivos e Benefícios**

Objetivos do Sistema:

* Automatizar Processos Acadêmicos: O principal objetivo do Acadêmico App é substituir os processos manuais de matrícula, registro de frequência e planejamento de aulas, automatizando essas tarefas para garantir maior eficiência.
* Facilitar o Acompanhamento Acadêmico: O sistema visa proporcionar uma forma prática para gestores e professores acompanharem o desempenho dos alunos e o histórico das aulas de forma rápida e intuitiva.
* Centralizar a Gestão Escolar: Consolidar todas as informações acadêmicas em um único sistema, evitando a dispersão dos dados e facilitando a consulta, atualização e análise de informações.
* Facilitar a Comunicação: O Acadêmico App oferece ferramentas para uma comunicação clara entre professores, alunos e gestores, como a possibilidade de registrar e consultar avisos importantes.

Benefícios Esperados:

* Eficiência Operacional: O sistema eliminará tarefas manuais demoradas, como preencher formulários em papel, gerando economia de tempo e trabalho.
* Aumento na Precisão: Ao automatizar o controle de frequência, cadastro de alunos e planejamento de aulas, o sistema minimiza erros humanos, garantindo maior precisão na gestão acadêmica.
* Melhoria na Transparência: Alunos, professores e gestores terão acesso a dados centralizados e atualizados em tempo real, promovendo maior transparência nas informações acadêmicas.
* Maior Agilidade no Acompanhamento Acadêmico: Com a digitalização dos registros, será possível gerar relatórios e acompanhar o progresso dos alunos de forma muito mais ágil.
* Redução de Custos Administrativos: Ao otimizar os processos administrativos, o Acadêmico App contribui para a redução de custos com materiais e recursos humanos.
* Melhoria no Planejamento Pedagógico: A possibilidade de registrar planos de aula e aulas ministradas de forma estruturada facilita a criação de um planejamento pedagógico mais eficiente e coerente com os objetivos educacionais da escola.

### 

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

#### *Explique o que são Requisitos no contexto da Engenharia de Software (importância para o projeto, como são definidos, como são utilizados no projeto).*

Os requisitos são as especificações e condições que o sistema deve atender para satisfazer as necessidades e expectativas dos interessados no projeto (clientes, usuários finais, desenvolvedores e outros envolvidos). Na Engenharia de Software, os requisitos são fundamentais porque eles definem as funcionalidades, comportamentos e restrições do sistema, servindo como base para o desenvolvimento, testes e validação do projeto.

* Importância para o Projeto: Os requisitos são essenciais porque orientam todo o ciclo de vida do desenvolvimento do software. Eles ajudam a garantir que o produto final atenda às necessidades do usuário e permita que a equipe de desenvolvimento saiba exatamente o que precisa ser feito.
* Definição: Os requisitos são definidos com base nas necessidades do cliente, nas expectativas dos usuários finais e nas restrições técnicas ou de negócios. Eles podem ser obtidos por meio de entrevistas, reuniões, análise de documentos existentes e observação direta dos processos.

**Utilização no Projeto**: Uma vez definidos, os requisitos são usados para elaborar o **Product Backlog**, definir as **histórias de usuário** e organizar o trabalho em sprints dentro de metodologias ágeis, como o Scrum. Eles servem como base para os testes de aceitação, garantindo que as funcionalidades estejam sendo implementadas corretamente.

**3.1 Requisitos Funcionais**

Os Requisitos Funcionais descrevem o comportamento específico do sistema. Eles especificam o que o sistema deve fazer, ou seja, as funcionalidades e os serviços que devem ser implementados. São os requisitos diretamente relacionados às funções que o sistema executará, como cadastro de usuários, registro de informações, cálculo de dados, entre outros.

Exemplos de Requisitos Funcionais podem ser:

* O sistema deve permitir que um professor registre a frequência de seus alunos.
* O sistema deve permitir o cadastro de alunos e professores.
* O sistema deve gerar relatórios sobre o histórico de aulas ministradas.

Esses requisitos ajudam a garantir que o sistema atenda ao que foi proposto, permitindo uma verificação e validação claras durante o desenvolvimento e testes.

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

As **Histórias de Usuário** são uma técnica usada em metodologias ágeis, como o Scrum, para capturar requisitos do ponto de vista do usuário. Elas descrevem a funcionalidade do sistema de maneira simples e focada nas necessidades dos usuários finais, utilizando a seguinte estrutura:

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

As histórias de usuário são pequenas e descritas de forma que qualquer membro da equipe de desenvolvimento consiga entender claramente o que deve ser implementado e qual problema ela resolve para o usuário. Elas são discutidas, detalhadas e refinadas conforme o progresso do projeto.

A seguir, estão as histórias de usuário do **Product Backlog** do **Acadêmico App**:

* + Como administrador, eu quero cadastrar os dados dos alunos para que eu possa manter um controle organizado das matrículas.
  + Como professor, eu quero registrar a frequência dos alunos por aula para que eu possa acompanhar a presença de cada aluno de forma rápida e eficiente.
  + Como administrador, eu quero cadastrar professores voluntários para que eu possa gerenciar sua atuação nas turmas de forma organizada e precisa.
  + Como professor, eu quero registrar um plano de aula para que eu possa planejar e organizar os conteúdos a serem abordados nas aulas de maneira eficaz.
  + Como professor, eu quero registrar as aulas já ministradas para que eu possa manter um histórico organizado das aulas que foram dadas.
  + Como administrador, eu quero alocar professores nas turmas conforme sua disponibilidade para que o planejamento das aulas seja eficiente e as turmas sejam bem distribuídas.

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

|  |  |
| --- | --- |
| Campo | Cadastro de Alunos |
| ID | 1 |
| Tipo | Funcional |
| Descrição | O sistema deve permitir ao administrador cadastrar os dados dos alunos (nome, CPF, telefone, e-mail, curso e turno). |
| Justificativa | Organizar e centralizar o processo de matrícula dos alunos. |
| Origem | Administrador |
| Dependências | Nenhuma. |
| Critério de Aceitação | O aluno é salvo corretamente no sistema e aparece na listagem de alunos. |
| Prioridade | Alta |
| Observações | Deve ser possível editar ou excluir um cadastro posteriormente. |

#### 

|  |  |
| --- | --- |
| Campo | Registro de Frequência |
| ID | 2 |
| Tipo | Funcional |
| Descrição | O sistema deve permitir ao professor registrar a frequência dos alunos por aula. |
| Justificativa | Facilitar o acompanhamento da presença dos alunos. |
| Origem | Professor |
| Dependências | Cadastro de alunos e aulas. |
| Critério de Aceitação | O professor marca presença/falta e os dados são salvos corretamente. |
| Prioridade | Alta |
| Observações | Deve ser possível visualizar frequência acumulada por aluno. |

|  |  |
| --- | --- |
| Campo | Cadastro de Professores |
| ID | 3 |
| Tipo | Funcional |
| Descrição | O sistema deve permitir ao administrador cadastrar professores com dados como nome, disciplina, dias e horários disponíveis. |
| Justificativa | Manter controle sobre os voluntários que atuam nas turmas. |
| Origem | Administrador |
| Dependências | Nenhuma. |
| Critério de Aceitação | Os dados são cadastrados corretamente e utilizados em outras funcionalidades como alocação. |
| Prioridade | Alta |
| Observações | Deve ser possível editar as informações posteriormente. |

|  |  |
| --- | --- |
| Campo | Cadastro de Plano de Aula |
| ID | 4 |
| Tipo | Funcional |
| Descrição | O sistema deve permitir ao professor criar planos de aula com data, tema, objetivos e materiais. |
| Justificativa | Planejar o conteúdo didático de forma estruturada. |
| Origem | Professor |
| Dependências | Cadastro da disciplina e turma. |
| Critério de Aceitação | Plano de aula salvo e acessível ao professor responsável. |
| Prioridade | Média |
| Observações | Deve ser possível editar e consultar planos anteriores. |

|  |  |
| --- | --- |
| Campo | Registro de Aulas Ministradas |
| ID | 5 |
| Tipo | Funcional |
| Descrição | O sistema deve permitir ao professor registrar as aulas ministradas com data, turma e descrição. |
| Justificativa | Manter histórico de aulas para consultas futuras. |
| Origem | Professor |
| Dependências | Plano de aula e turma cadastrados. |
| Critério de Aceitação | Aula registrada com sucesso e visível no histórico. |
| Prioridade | Média |
| Observações | Pode ser vinculado automaticamente ao plano de aula. |

|  |  |
| --- | --- |
| Campo | Alocação de Professores |
| ID | 6 |
| Tipo | Funcional |
| Descrição | O sistema deve permitir ao administrador alocar professores nas turmas com base em sua disponibilidade e disciplina. |
| Justificativa | Otimizar o uso dos professores e garantir cobertura das turmas. |
| Origem | Administrador |
| Dependências | Cadastro de professores, disciplinas e turmas. |
| Critério de Aceitação | Alocação feita com base nas preferências e exibida corretamente. |
| Prioridade | Média |
| Observações | Pode incluir sugestões automáticas de alocação. |

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

Os **requisitos não funcionais** descrevem as propriedades e restrições do sistema que não estão diretamente relacionadas com as funcionalidades, mas que influenciam **o desempenho, a confiabilidade, a usabilidade, a segurança e outras qualidades técnicas** do software. Eles não definem “o que” o sistema faz, mas sim “como” ele deve funcionar.

Esses requisitos são essenciais para garantir que o sistema seja eficiente, seguro, acessível e compatível com as necessidades operacionais e tecnológicas da instituição.

A seguir, os requisitos não funcionais definidos para o sistema:

1. O sistema deve estar disponível 24 horas por dia, 7 dias por semana.
2. O tempo de resposta das funcionalidades não deve ultrapassar 2 segundos.
3. A interface do usuário deve ser compatível com dispositivos móveis.
4. A autenticação dos usuários deve ser feita por login e senha.

### 

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

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

#### https://documents.lucid.app/documents/9e189b5d-ab47-4352-ac75-631cecc1c030/pages/0_0?a=334&x=-720&y=-223&w=567&h=1346&store=1&accept=image%2F*&auth=LCA%20385f0244530e266fb5d44a87cef697f30e386dedac2e7204e0a1110e4c9c6780-ts%3D1747838861

#### **Especificação dos Casos de Uso**

* **Caso de Uso 1: Registrar Frequência**

**Atores**: Professor, Sistema

**Fluxo de Eventos Principal:**

1. O professor acessa a turma no sistema.
2. O professor seleciona a data da aula.
3. O sistema apresenta a lista de alunos matriculados em ordem alfabética.
4. O professor marca os alunos presentes.
5. O sistema salva os dados de frequência.

**Fluxo Alternativo**:

* 1. Aluno ausente:

1. No passo 4, se o aluno estiver ausente, o professor registra a falta.
2. O caso de uso retorna para o passo 4 para continuar o registro dos demais alunos.

* **Caso de Uso 2: Realizar Matrícula**

**Atores**: Secretaria, Sistema

**Fluxo de Eventos Principal**:

1. A secretaria acessa o sistema com seu login administrativo.
2. A secretaria seleciona a opção de “Cadastrar novo aluno”.
3. O sistema exibe um formulário solicitando dados pessoais (nome, CPF, telefone, e-mail) e informações acadêmicas (curso e turno).
4. A secretaria preenche os dados do aluno e confirma o cadastro.
5. O sistema armazena as informações e exibe uma mensagem de matrícula realizada com sucesso.

**Fluxo Alternativo**:

4.1. Dados incompletos ou inválidos:

1. O sistema alerta que todos os campos obrigatórios devem ser preenchidos corretamente.
2. A secretaria revisa os dados e volta ao passo 4.

* **Caso de Uso 3: Cadastrar Professor**

**Atores**: Administrador, Sistema

**Fluxo de Eventos Principal**:

1. A secretaria acessa a interface de cadastro de professor.
2. O sistema exibe o formulário com campos para nome, disciplina e disponibilidade.
3. A secretaria preenche os dados do professor.
4. A secretaria confirma e salva o cadastro.
5. O sistema armazena os dados e exibe uma mensagem de confirmação.

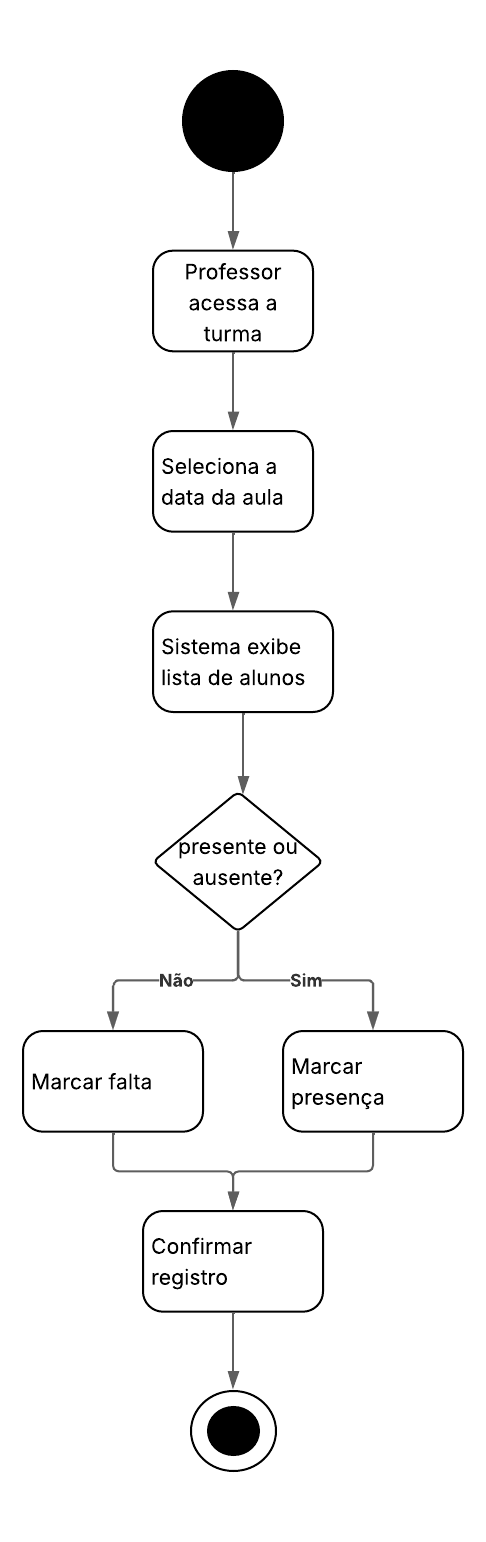
**Fluxo Alternativo**:

**3.1**. Informações inválidas:

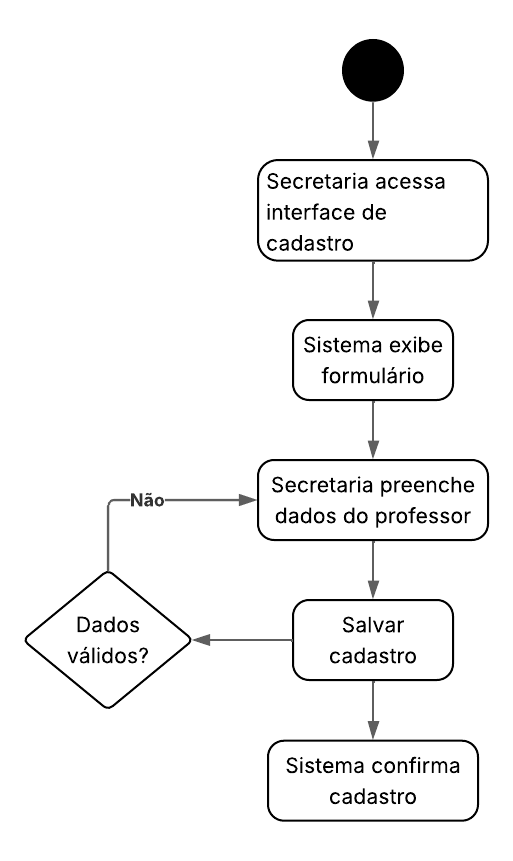
1. O sistema informa os campos obrigatórios que estão incorretos ou faltando.
2. O administrador corrige os dados e retorna ao passo 4.

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

* Registrar Frequência



* Cadastrar Professor



### **6. Protótipos das Telas**

*Apresentar imagens que representam as principais telas do sistema (os protótipos podem ser feitos com ferramentas digitais ou desenhados à mão.*

### **7. Considerações Finais**

*Descrever aprendizados da equipe, limitações ou decisões tomadas durante a especificação e próximos passos para a finalização do sistema iniciado no 1o semestre.*

### **8. Anexos (se necessário)**

*Documentos complementares, glossário ou referências utilizadas.*