



# UNINASSAU

Centro Universitário Maurício de Nassau  
Análise e Desenvolvimento de Sistemas  
Uninassau - Caxangá

## **Documento de Requisitos do Sistema**

Aplicação de acesso e controle ao laboratório de informática

Versão 0.1



# UNINASSAU

Centro Universitário Maurício de Nassau  
Análise e Desenvolvimento de Sistemas  
Uninassau - Caxangá

## **Documento de Requisitos do Sistema**

Aplicação de acesso e controle ao laboratório de informática - 2023.1

Artur Andrade Barbosa de Lima  
Dafnys Hugo Nascimento dos Santos Pereira  
Guilherme Gomes Barbosa de Barros  
Igor Cavalcanti Moura  
José Henrique Freitas da Silva  
Rodrigo Silva do Nascimento

## 1 . Introdução

Este documento especifica os requisitos da “Aplicação de acesso e controle ao laboratório de informática”, fornecendo aos projetistas e desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

### 1.1. Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

- **Seção 2 Descrição geral do sistema:** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3 Requisitos funcionais:** Especifica brevemente os casos de uso do sistema.
- **Seção 4 Requisitos não funcionais:** cita e explica os requisitos não funcionais do sistema.
- **Seção 5 Especificação de requisitos do sistema (casos de uso):** descreve requisitos funcionais e não funcionais mais detalhadamente. No caso de requisitos funcionais, descreve os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- **Seção 6 Tech Stack:** Apresenta a seleção estratégica de tecnologias que trabalham em conjunto para criar a solução de software.

## 2. Descrição geral do sistema

A aplicação proposta oferece uma solução para o problema de agendamento de uso do laboratório de informática, que atualmente enfrenta dificuldades devido ao grande número de professores que necessitam utilizá-lo. A proposta consiste na criação de uma API REST e de um aplicativo em Flutter para android, no qual os docentes da instituição possam controlar e gerenciar o acesso ao laboratório.

Com a implementação da API REST e do aplicativo desenvolvido em Flutter, espera-se facilitar o agendamento do uso do laboratório de informática, proporcionando um controle mais eficiente e melhorando a experiência dos professores que utilizam o ambiente.

Para a implementação da API REST, será utilizada a tecnologia Spring Framework, que é uma plataforma amplamente utilizada para o desenvolvimento de aplicativos Java. O Spring Framework oferece suporte ao desenvolvimento de APIs RESTful e possui recursos avançados de segurança e integração com bancos de dados.

A API será responsável por gerenciar o acesso ao laboratório de informática, permitindo o cadastro de usuários, agendamento de uso do laboratório e autenticação de login. O aplicativo Flutter irá consumir a API e fornecerá uma interface amigável para os professores interagirem com o sistema.

### **3. Requisitos funcionais**

#### **3.1. Realizar cadastro do usuário.**

**Prioridade:** ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve permitir o cadastro de novos usuários, fornecendo endpoints para as operações POST, PUT, DELETE e GET relacionadas aos dados do usuário.

#### **3.2. Realizar login.**

**Prioridade:** ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve suportar um mecanismo de autenticação seguro, permitindo que os usuários façam login no aplicativo para acessar as funcionalidades disponíveis.

#### **3.3. Realizar agendamento do uso do laboratório.**

**Prioridade:** ☒ Essencial ☐ Importante ☐ Desejável

O sistema deve fornecer funcionalidades para agendar o uso do laboratório, permitindo a criação, atualização, exclusão e consulta de agendamentos por meio de endpoints correspondentes.

#### **3.4. Compatibilidade com dispositivos Android**

**Prioridade:** ☒ Essencial ☐ Importante ☐ Desejável

O aplicativo deve ser desenvolvido de forma a ser compatível e executar corretamente em dispositivos Android. Isso implica que o aplicativo deve ser projetado levando em consideração as diretrizes de design específicas do Android. Todas as funcionalidades e recursos do aplicativo devem estar disponíveis e acessíveis em dispositivos Android, sem quaisquer limitações ou restrições significativas.

O aplicativo deve ser testado em emuladores, para garantir a sua funcionalidade e compatibilidade. O objetivo é fornecer aos usuários Android uma experiência de uso do aplicativo sem problemas e atender às suas expectativas de desempenho e usabilidade em seus dispositivos.

## **4. Requisitos não funcionais**

### **4.1. Segurança**

A API deve garantir a segurança dos dados e das operações, implementando mecanismos de autenticação e autorização adequados.

### **4.2. Desempenho**

A API deve ser capaz de lidar com múltiplas solicitações simultâneas de forma eficiente, proporcionando uma resposta rápida aos usuários.

### **4.3. Escalabilidade**

A solução deve ser projetada levando em consideração a possibilidade de crescimento futuro, permitindo a adição de novos recursos e acomodando um aumento no número de usuários.

## **5. Especificação de requisitos do sistema (Caso de uso)**

### **5.1. Realizar cadastro do usuário.**

#### ***Fluxo de Eventos:***

- O usuário acessa a funcionalidade de cadastro de usuários.
- O ator preenche os dados do novo usuário (nome, e-mail, matrícula e senha).
- O sistema valida e armazena os dados do novo usuário.
- O sistema retorna uma confirmação de sucesso ao ator.

**Atores:** Usuário Administrador

**Entradas:** Dados do novo usuário (nome, e-mail, matrícula e senha).

**Saídas:** Confirmação de sucesso ou mensagem de erro.

## 5.2. Realizar login.

### Fluxo de Eventos:

- O ator (usuário) acessa a funcionalidade de login.
- O ator fornece suas credenciais (e-mail e senha).
- O sistema verifica a validade das credenciais fornecidas.
- Se as credenciais forem válidas, o sistema autentica o usuário.
- O sistema retorna uma confirmação de sucesso ao ator.

**Atores:** Usuário

**Entradas:** Credenciais do usuário (e-mail e senha)

**Saídas:** Confirmação de sucesso ou mensagem de erro.

## 5.3. Realizar agendamento do uso do laboratório.

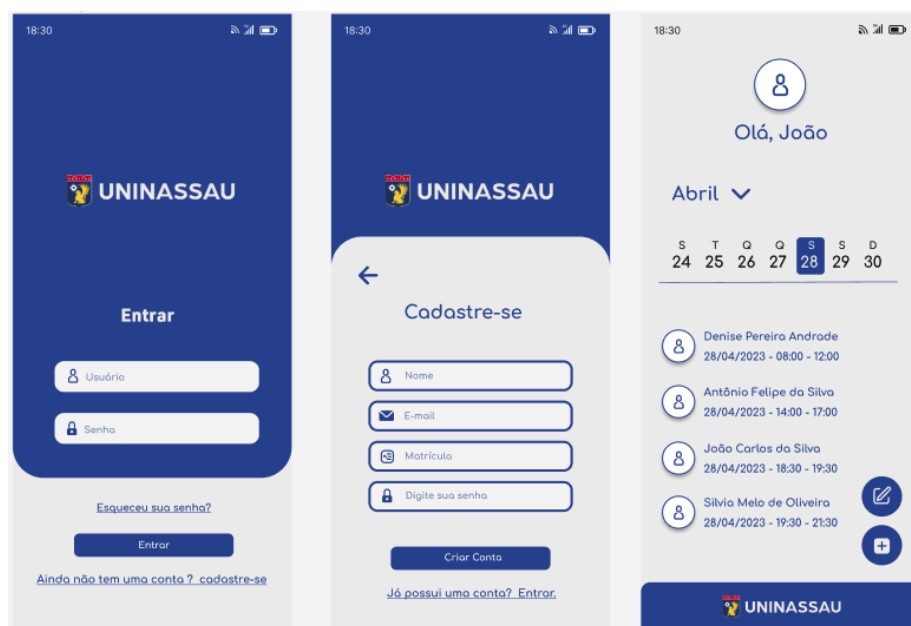
### Fluxo de Eventos:

- O ator (usuário professor) acessa a funcionalidade de agendamento.
- O ator seleciona a data e o horário desejados para o uso do laboratório.
- O sistema verifica a disponibilidade do laboratório na data e horário selecionados.
- Se o laboratório estiver disponível, o sistema registra o agendamento.
- O sistema retorna uma confirmação de sucesso ao ator.

**Atores:** Usuário Professor

**Entradas:** Data e horário do agendamento

**Saídas:** Confirmação de sucesso ou mensagem de erro.



## **6. Tech Stack do App**

Refere-se ao conjunto de tecnologias, ferramentas e frameworks utilizados para o desenvolvimento deste aplicativo. Essa combinação de tecnologias foi selecionada com base nos requisitos do projeto e nas necessidades do negócio, englobando as diferentes camadas e componentes do sistema, incluindo frontend (interface do usuário), backend (lógica de negócios e manipulação de dados) e banco de dados.

### **6.1. Frontend**

O aplicativo é desenvolvido utilizando o framework Flutter, que permite a criação de interfaces de usuário nativas para Android e iOS a partir de um único código-base. Flutter utiliza a linguagem de programação Dart e fornece uma vasta biblioteca de widgets personalizáveis para a construção da interface do usuário.

### **6.2. Backend**

O backend do aplicativo é desenvolvido utilizando o Spring Framework, que é um framework popular para o desenvolvimento de aplicativos Java. O Spring Framework oferece recursos abrangentes para a criação de APIs RESTful, incluindo o suporte a endpoints, manipulação de solicitações HTTP, autenticação e autorização.

### **6.3. Banco de Dados**

O banco de dados MySQL é escolhido como a solução de armazenamento de dados para o aplicativo. O MySQL é um sistema de gerenciamento de banco de dados relacional amplamente utilizado, conhecido por sua confiabilidade, desempenho e suporte à linguagem SQL. Ele permite a criação de tabelas, consultas e operações de manipulação de dados.

### **6.4. Outras Tecnologias**

#### **6.4.1. Dart**

A linguagem de programação Dart é utilizada para o desenvolvimento do aplicativo Flutter. Dart é uma linguagem moderna, orientada a objetos, com suporte a tipagem estática e assincronismo, otimizada para o desenvolvimento de aplicativos móveis.

#### **6.4.2. API RESTful**

O backend do aplicativo é projetado como uma API RESTful, seguindo as melhores práticas para a comunicação entre o frontend e o backend. A API RESTful permite que o aplicativo realize operações CRUD (Create, Read, Update, Delete) no banco de dados MySQL, fornecendo endpoints para cada funcionalidade.