Requisitos Funcionais da API REST:

**Cadastro de Usuário:**

A API deve permitir o cadastro de um novo usuário com informações como nome, e-mail e senha.

O cadastro deve ser feito através do método POST.

**Atualização de Usuário:**

A API deve permitir a atualização das informações de um usuário existente.

A atualização deve ser feita através do método PUT, informando o ID do usuário a ser atualizado.

**Exclusão de Usuário:**

A API deve permitir a exclusão de um usuário existente.

A exclusão deve ser feita através do método DELETE, informando o ID do usuário a ser excluído.

**Agendamento de Uso do Laboratório:**

A API deve permitir o agendamento do uso do laboratório, informando data, horário de início e horário de término.

O agendamento deve ser feito através do método POST.

**Atualização de Agendamento:**

A API deve permitir a atualização das informações de um agendamento existente.

A atualização deve ser feita através do método PUT, informando o ID do agendamento a ser atualizado.

**Exclusão de Agendamento:**

A API deve permitir a exclusão de um agendamento existente.

A exclusão deve ser feita através do método DELETE, informando o ID do agendamento a ser excluído.

Consulta de Agendamento:

A API deve permitir a consulta das informações de um agendamento específico.

A consulta deve ser feita através do método GET, informando o ID do agendamento desejado.

Requisitos Não Funcionais da API REST:

**Segurança:**

As senhas dos usuários devem ser armazenadas de forma segura, utilizando armazenamento seguro.

**Desempenho:**

A API deve ser eficiente e responsiva, suportando um grande número de requisições simultâneas.

Deve ser otimizada para minimizar o tempo de resposta e o consumo de recursos do servidor.

**Integração com Banco de Dados:**

A API deve ser capaz de se integrar a um banco de dados para armazenar e recuperar os dados dos usuários e agendamentos.

O banco de dados escolhido deve ser compatível com o Spring Framework e atender aos requisitos de segurança e desempenho.

**Protótipo do Aplicativo Flutter no Figma:**

O protótipo do aplicativo Flutter deve ser criado no Figma para demonstrar o comportamento do aplicativo, incluindo as telas de cadastro de usuário, login, agendamento do uso do laboratório e listagem dos agendamentos existentes.

**API REST com Integração ao Banco de Dados:**

A API REST deve ser desenvolvida utilizando o Spring Framework e se integrar a um banco de dados escolhido, como MySQL ou MariaDB.

A API deve disponibilizar os endpoints necessários para o cadastro, atualização, exclusão e consulta de usuários e agendamentos.

Os dados devem ser armazenados no banco de dados e recuperados conforme as requisições feitas pelo aplicativo Flutter.

**Aplicativo Flutter:**

O aplicativo Flutter deve ser desenvolvido para permitir o cadastro de usuários, agendamento do uso do laboratório, login e consulta dos agendamentos existentes.

O aplicativo deve consumir os endpoints da API REST para interagir com o banco de dados e realizar as operações desejadas.

Deve haver telas para a criação e atualização de usuários, agendamentos e login, bem como para exibir a lista de agendamentos existentes.

Tech Stack:

Spring Framework: Para o desenvolvimento da API REST, utilizando Java ou Kotlin.

Flutter: Para o desenvolvimento do aplicativo multiplataforma.

Banco de Dados: Pode ser escolhido entre MySQL, MariaDB ou outro banco de dados compatível com o Spring Framework e que atenda aos requisitos do projeto.