FI*AP*

Curso: Análise e Desenvolvimento de Sistemas 1TDSPJ

Felipe Maglio Filho

Mateus Granja dos Santos

Vitória Valentina Maglio

**FACILITAREABI**

Sistema de Teleconsulta e Gestão de Agendamentos

São Paulo – SP

2025

Sumário

**1** **Introdução**...................................................................................................................................3

**2** **Descritivo do Projeto**..................................................................................................................3

**2.1** **Objetivo e Escopo.**..............................................................................................................3

**2.2** **Funcionalidades do Sistema**..............................................................................................4

**3 Tabela de Endpoints.**.................................................................................................................5

4 **Protótipo das telas.**.....................................................................................................................6

5 **MER (Modelo Entidade Relacionamento** .............................................................................14

6 **UML (Diagrama de Classes)**....................................................................................................15

7 **Procedimentos para rodar a aplicação**...................................................................................16

8 **Considerações finais.**................................................................................................................17

# Introdução

Este documento tem como objetivo apresentar uma descrição do projeto FacilitaReabi, desenvolvido na linguagem Java, com foco na resolução do problema apresentado pelo Hospital das Clínicas aos alunos da FIAP. Através desse sistema, os usuários podem realizar cadastros, agendar consultas, receber notificações, e médicos do sistema conseguem listar todos os pacientes e consultas cadastrados no banco de dados.

# Descritivo do Projeto

## Objetivo e Escopo

O projeto FacilitaReabi tem como objetivo oferecer uma solução para pacientes em processo de reabilitação, especialmente aqueles com dificuldades no uso de tecnologias móveis, a fim de que possam utilizar as teleconsultas para melhorar sua saúde. Seu objetivo principal é permitir o cadastro de pacientes, consultas e usuários, além de oferecer recursos para gerenciar teleconsultas (como agendamentos, remarcações e cancelamentos), garantindo que pacientes em situação de vulnerabilidade recebam apoio.

O escopo do sistema se limita às seguintes atividades:

* Registro e gerenciamento de usuários, pacientes e consultas por meio de endpoints REST.
* Validação de dados básicos para evitar cadastros incorretos.
* Registro e acompanhamento de consultas com status definidos (agendada, remarcada, cancelada).
* Registro de motivos de ausência em consultas.
* Persistência dos dados em um banco relacional via DAO (CRUD).

O sistema não substitui sistemas médicos completos (como prontuários eletrônicos), mas atua de forma complementar, focado em reduzir faltas nas teleconsultas e facilitar a comunicação com pacientes.

O cadastro de pacientes inclui uma **triagem de aptidão para teleconsultas**, verificando se o paciente possui condições mínimas para realizar atendimentos online. Isso ajuda a reduzir o absenteísmo, garantindo que apenas pacientes aptos sejam agendados para teleconsultas.

O sistema também permite:

* Armazenar dados pessoais de usuários para acesso ao aplicativo.
* Registrar pacientes e consultas, associando-os corretamente.
* Enviar notificações sobre consultas com base na interação do paciente com o sistema.
* Coletar feedback de consultas e da aplicação para uso administrativo e melhoria contínua.

A arquitetura do sistema reflete o **contexto do projeto**, onde cada classe tem um papel específico no cuidado digital, na adesão ao tratamento e na promoção da inclusão.

## Funcionalidades do Sistema

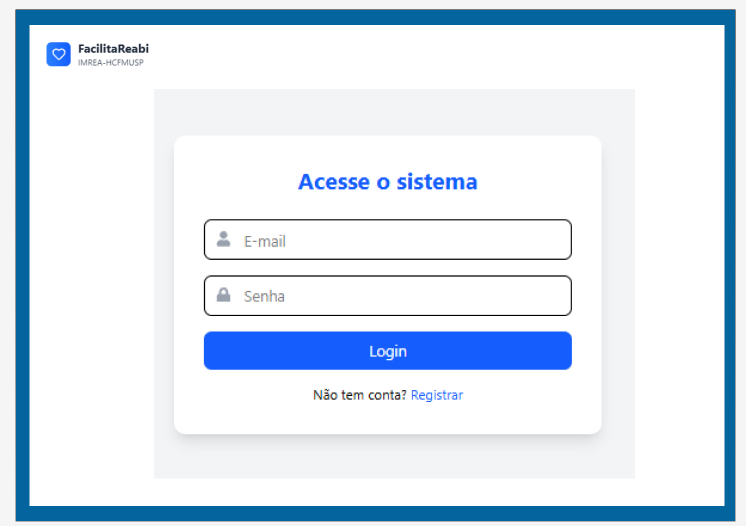
O projeto é composto pelas seguintes classes principais:

* **Cadastro de Pacientes**
  + Inserção de dados básicos (nome, CPF, telefone, e-mail, data de nascimento e vulnerabilidade).
  + Validação de dados para evitar cadastros incompletos.
  + Disponibilização de endpoints REST para **cadastrar, listar, buscar, atualizar e excluir pacientes**.
* **Cadastro de consultas**
  + Registro de consultas associadas a pacientes.
  + Definição de **data, status e motivo da falta** (quando aplicável).
  + Endpoints REST para **cadastrar, listar, atualizar e excluir consultas**.
* **Cadastro de Usuários**
  + Armazenamento de **login e senha criptografada** de administradores/médicos.
  + Endpoints REST para **cadastrar, autenticar, listar, atualizar e excluir usuários**.
* **Validação de aptidão pata teleconsultas**
  + Verifica se o paciente possui condições mínimas (como ausência de vulnerabilidades graves) para realizar a teleconsulta.
* **Remarcação de Consultas**
  + Permite remarcar uma consulta existente, alterando a data e registrando o motivo da ausência.
* **Cancelamento de Consultas**
  + Permite cancelar consultas já agendadas, com registro de motivo da falta.
* **Consulta a Instruções do Sistema (FacilitaReabi)**
  + Disponibiliza informações sobre como utilizar a teleconsulta e promove a acessibilidade digital para os usuários.
* **Persistência em Banco de Dados (DAO + CRUD)**
  + Inserção, atualização, exclusão e busca de registros de paciente, consulta e usuário.
* **Camadas bem definidas (MVC + Service + DAO)**
  + **Controller**: recebe entradas do usuário e dispara fluxos.
  + **Service**: contém as regras de negócio e validações.
  + **DAO**: executa o CRUD no banco de dados.
  + **Model**: representa os objetos do mundo real (Paciente, Consulta, Usuário, FacilitaReabi).

**3 Tabela de Endpoints**   
**Uris - Verbos Http - Código de status de resposta esperados**

|  |  |  |
| --- | --- | --- |
| **Recurso / Endpoint** | **Verbo HTTP** | **Código de Resposta Esperado** |
| /usuarios | POST | 201 Created / 400 Bad Request / 500 Internal Server Error |
| /usuarios/login | POST | 201 Created / 400 Bad Request / 500 Internal Server Error |
| /usuarios | GET | 200 OK / 500 Internal Server Error |
| /usuarios/{login} | GET | 200 OK / 404 Not Found / 500 Internal Server Error |
| /usuarios/{id} | PUT | 200 OK / 500 Internal Server Error |
| /usuarios/{login} | DELETE | 200 OK / 500 Internal Server Error |
| /pacientes | POST | 201 Created / 400 Bad Request / 500 Internal Server Error |
| /pacientes | GET | 200 OK / 500 Internal Server Error |
| /pacientes/{nome} | GET | 200 OK / 404 Not Found / 500 Internal Server Error |
| /pacientes/{id\_paciente} | PUT | 200 OK / 500 Internal Server Error |
| /pacientes/{nome} | DELETE | 200 OK / 500 Internal Server Error |
| /consultas | POST | 201 Created / 400 Bad Request / 500 Internal Server Error |
| /consultas/cadastradas | GET | 200 OK / 500 Internal Server Error |
| /consultas/{id} | PUT | 200 OK / 500 Internal Server Error |
| /consultas/{id} | DELETE | 200 OK / 500 Internal Server Error |

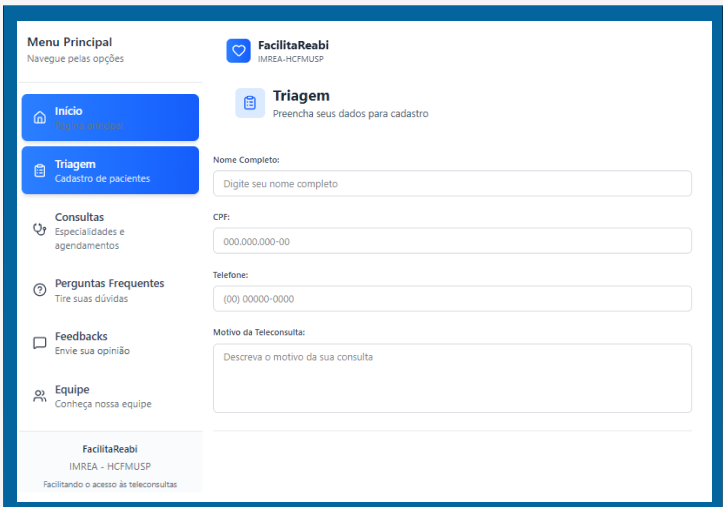
# 4 Protótipo das telas

A primeira página do site contém o cadastro do usuário, no qual os dados digitados são armazenados no banco de dados. 

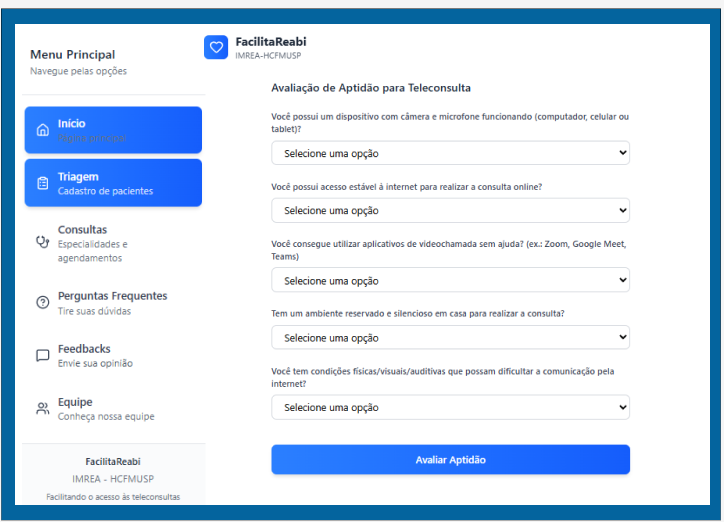
A segunda página do site contém uma explicação do FacilitaReabi e botões acessíveis que direcionam o paciente para outras telas do site.



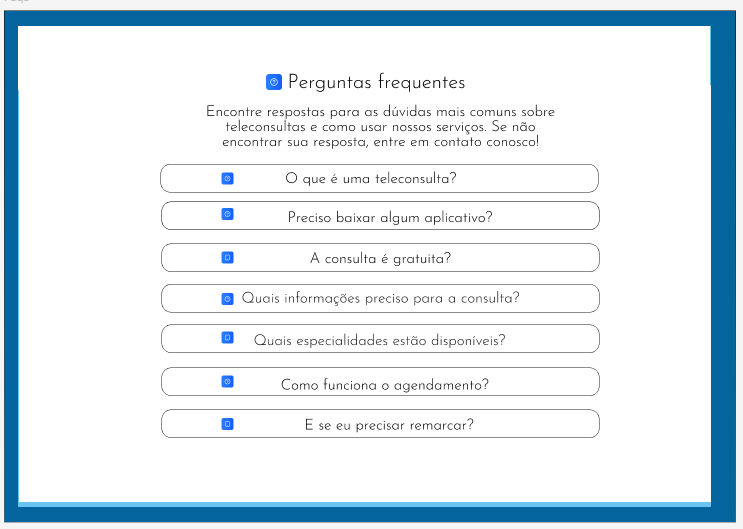
A terceira página do site contém o cadastro do paciente, no qual os dados digitados são armazenados no banco de dados e aqui é feita a identificação do paciente com disponibilidade para realizar teleconsultas.



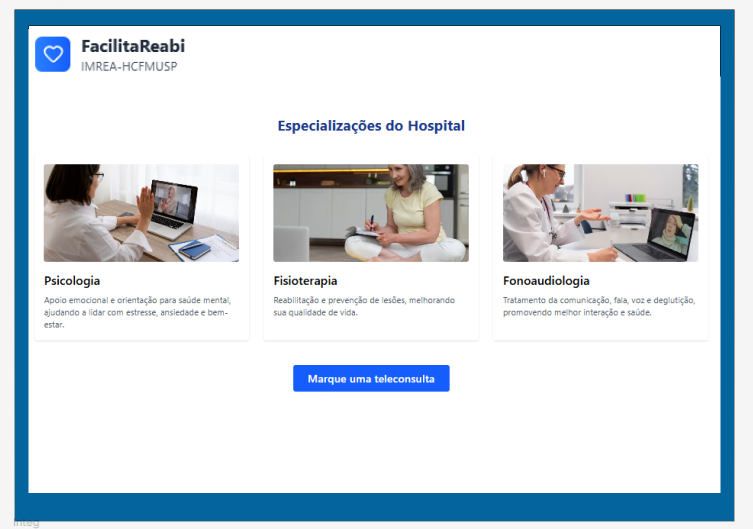
Continuação do cadastro do paciente. Nessa etapa, se o usuário se cadastrar com vulnerabilidade, ele não será indicado para realizar consultas on-lines, e no banco de dados, na tabela “paciente”, terá uma coluna “aptidao” com um valor preenchido com "Não apto", se o cadastro não possuir vulnerabilidade essa coluna será preenchida com "Apto".



Página que busca educar o usuário com perguntas e respostas sobre teleconsultas.



Página para explicação sobre as especializações médicas disponíveis em teleconsultas.

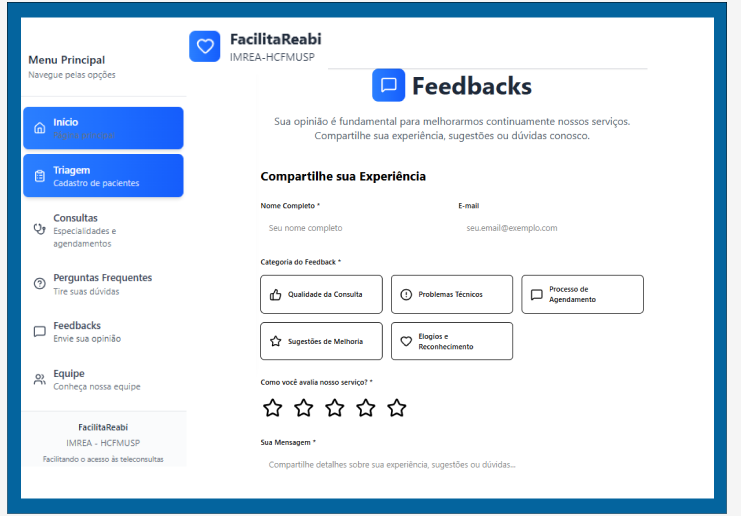


Página na qual o paciente insere os dados sobre sua consulta, e esses dados são cadastrados no banco.

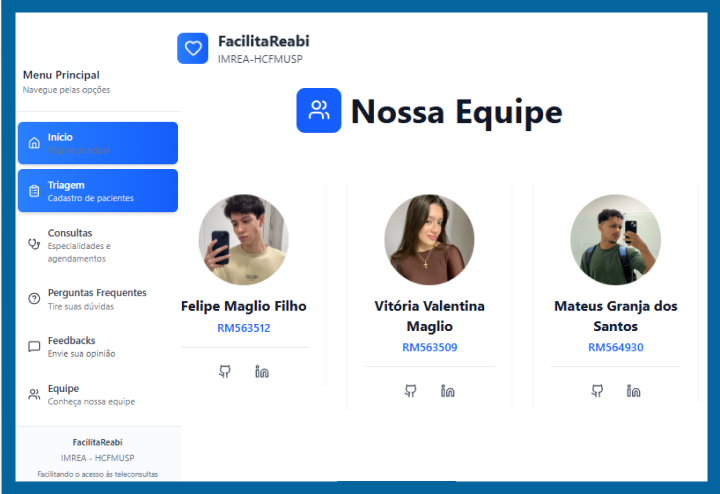
Página de reagendamento de consultas, na qual o paciente deve digitar o sua nova data de consulta, a data da consulta cancelada e motivo da sua falta, para identificar no banco e substituir a data e para que o Hospital possa tratar esse dado e otimizar seu sistema.

Página de cancelamento de consultas, na qual o paciente deve digitar o motivo da sua falta, para que o Hospital possa tratar esse dado e otimizar seu sistema.

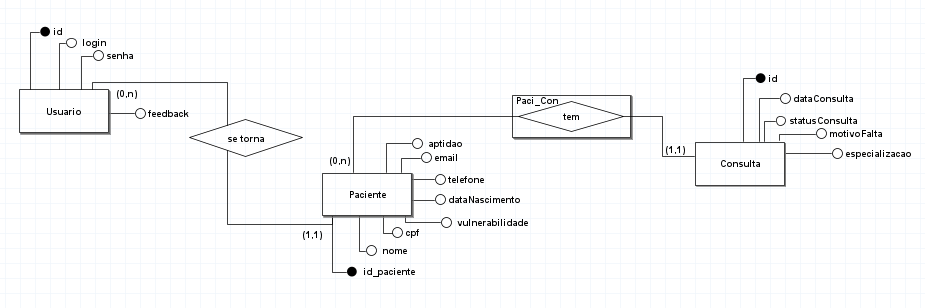
Página para usuários colocarem suas opiniões sobre a acessibilidade do site e dicas de como torná-lo mais inclusivo.



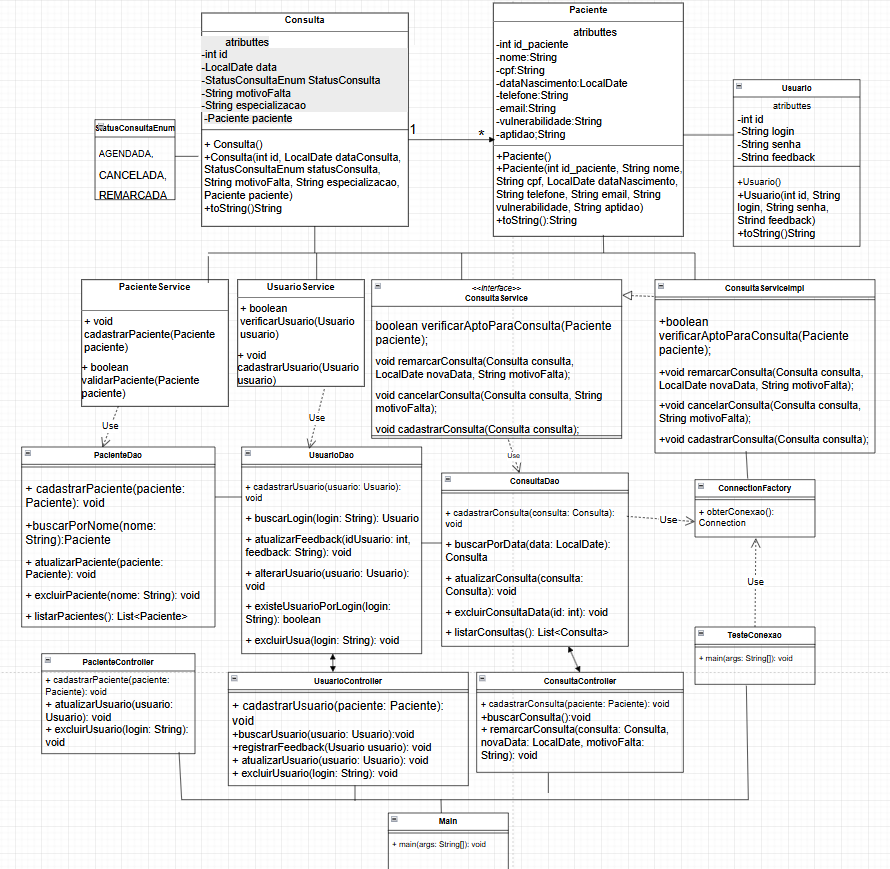
Página obrigatória em outras disciplinas para identificação dos integrantes do grupo.



# MER (Modelo Entidade Relacionamento)



# 5 UML (Diagrama de Classes)



# 6 Procedimentos para rodar a aplicação

### **Ferramentas utilizadas**

* IDE: IntelliJ IDEA, Eclipse, NetBeans ou VS Code
* Banco de dados: Oracle Database (pode ser Oracle XE)
* Driver JDBC: ojdbc11.jar
* Controle de versão: Git (opcional)

### **Versões necessárias**

* Java JDK 17 (ou superior)
* Oracle Database 19c ou Oracle XE
* ojdbc11.jar compatível com sua versão do JDK

### **Passo a passo da execução**

1. Instale o JDK 17 e configure o JAVA\_HOME.
2. Instale o Oracle Database e adicione as credenciais Nome: rm563509 e Senha: 140607 para criar as tabelas e realizar a conexão entre Java e o banco de dados.
3. Execute os scripts SQL para criar as tabelas necessárias (paciente, consulta, usuario, etc.).
4. Baixe ou clone o projeto para sua máquina.
5. Abra o projeto em uma IDE (IntelliJ, Eclipse, NetBeans ou VS Code).
6. Configure a biblioteca ojdbc11.jar no classpath do projeto.
7. No arquivo ConnectionFactory, configure a URL, usuário e senha do banco.
8. Rode a classe Main para iniciar a aplicação.
9. Interaja com o sistema via console (menu com Scanner).

### **Importar o projeto (se necessário)**

* **IntelliJ**: File -> Open -> selecione a pasta do projeto.
* **Eclipse**: File -> Import -> Existing Projects into Workspace.
* **NetBeans**: File -> Open Project.

# 7 Considerações finais

Este documento contém detalhes sobre a solução FacilitaReabi, desenvolvido na linguagem Java, com foco na resolução do problema apresentado pelo Hospital das Clínicas aos alunos da FIAP.