

Viva Bem Digital



Equipe DuoTech

Felipe da Silva Garcia Honores RM:563485

Rafael Siqueira Martins RM:564900

1TDSA

Sumario

- **Objetivo e Escopo do Projeto**
- **Solução Desenvolvida**
- **Destaque das Funcionalidades Implementadas**
- **Protótipo do Sistema**
- **Diagrama de Classes**
- **Tabela SQL**
- **Link GitHub**

Objetivo e escopo do projeto

Solução Desenvolvida

O projeto consiste em um aplicativo complementar ao sistema do HC, criado para reduzir as faltas em teleconsultas. A solução foi pensada para pacientes com dificuldade no uso de tecnologia, oferecendo acesso simples e inclusivo por meio de reconhecimento facial, interface intuitiva, suporte em áudio e chatbot para dúvidas frequentes.

Propósito e Limites

O principal propósito do sistema é aumentar a adesão às consultas e facilitar o acesso de pacientes que enfrentam barreiras digitais, garantindo inclusão e acessibilidade. O aplicativo não substitui o sistema principal do hospital, mas atua como apoio, limitado às funções essenciais de login, orientação ao paciente e suporte básico durante o agendamento e preparação das consultas.

Destaque das funcionalidades Implementadas

- **API VIA CEP**

Consiste em buscar o endereço pelo cep, e foi implementada alguns atalho de cep para facilidade sem ter que ficar digitando, voce escolhe a opção e ele ja mostra o endereço buscado pelo cep automaticamente

- **CRUD CONSULTA**

Consiste em um sistema que voce consegue, agendar a consulta, ver a consulta agendada, atualizar a consulta e deletar a consulta

- **CRUD LEMBRETE**

Consiste em um sistema que voce consegue, criar lembretes, ver os lembretes criado, atualizar os lembretes e deletar os lembretes

Protótipo do sistema



Tela de Boas-vindas

- **Função:** Apresentar o aplicativo e iniciar a navegação.
- **Interação:** O usuário toca no botão **Iniciar** para prosseguir para o login.

Tela de Login

- **Função:** Permitir o acesso ao sistema.
- **Interação:** O usuário pode informar o **CPF** ou utilizar o **reconhecimento facial** para autenticação. Caso tenha dificuldades, pode acionar o botão **Ajuda para acessar**.

Tela de Menu Principal

- **Função:** Exibir as principais opções do sistema.
- **Interação:** O usuário escolhe o que deseja fazer por meio de botões claros e com ícones, podendo:
 - **Agendar Consulta**

- **Iniciar Teleconsulta**
- **Visualizar Minhas Consultas**
- **Consultar Histórico Médico**
- **Acessar Ajuda e Suporte**
- **Configurações** (no rodapé da tela, para ajustes de acessibilidade).

Tabela de Endpoints (API Restful)

AgendamentoResource

@Path("/agendamentos")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

ConsultaResource

@Path("/consulta")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

EnfermeiroResource

@Path("/enfermeiro")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

MedicoResource

@Path("/medico")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

PacienteResource

@Path("/paciente")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

ProntuarioResource

@Path("/prontuario")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

SetorResource

@Path("/setor")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

CepResource

@Path("/viacep")

@GET / @POST / @PUT / @DELETE

@Produces("application/json")

@Consumes("application/json")

Diagrama de Classes

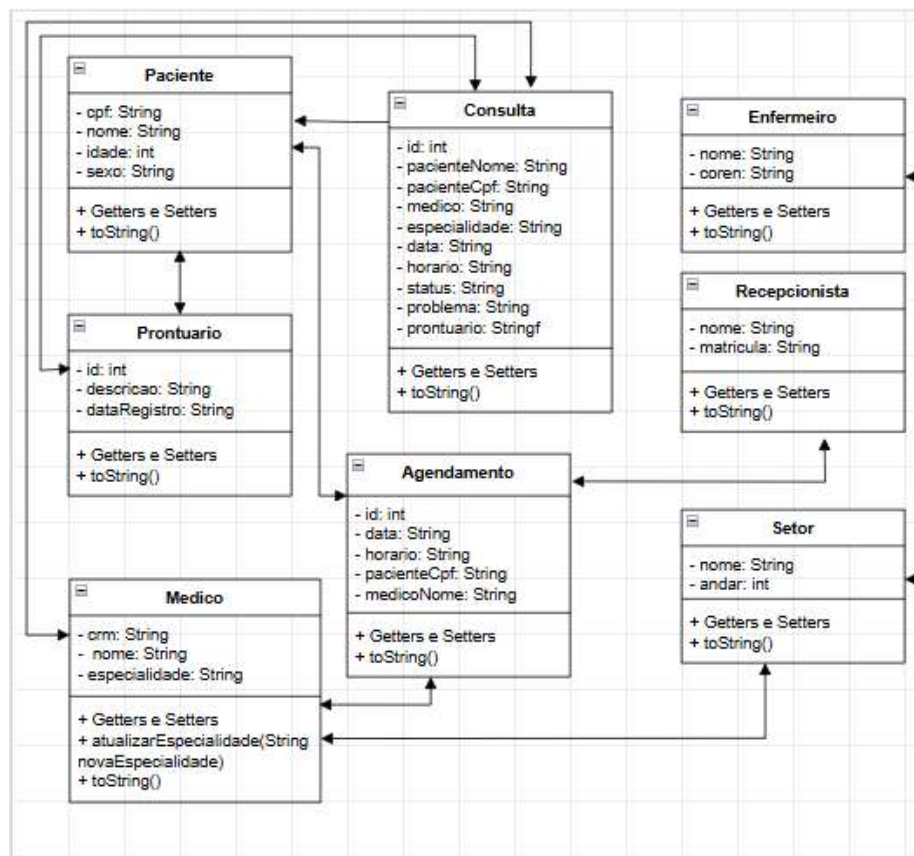


Tabela SQL

```

CREATE TABLE AGENDAMENTOS (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    PACIENTE_CPF VARCHAR2(20),
    MEDICO_NOME VARCHAR2(100),
    DATA VARCHAR2(10) NOT NULL,
    HORARIO VARCHAR2(5) NOT NULL
);

CREATE TABLE CONSULTAS (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    PACIENTE_NOME VARCHAR2(100) NOT NULL,
    PACIENTE_CPF VARCHAR2(20),
    MEDICO VARCHAR2(100),
    ESPECIALIDADE VARCHAR2(100),
    DATA VARCHAR2(10) NOT NULL,
    HORARIO VARCHAR2(5) NOT NULL,
    PROBLEMA VARCHAR2(200)
);

CREATE TABLE ENFERMEIROS (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    COREN VARCHAR2(20) NOT NULL,
    SETOR VARCHAR2(100)
);

```

```

CREATE TABLE MEDICOS (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    CRM VARCHAR2(20) NOT NULL,
    ESPECIALIDADE VARCHAR2(100)
);

CREATE TABLE PACIENTES (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    CPF VARCHAR2(20) NOT NULL,
    NOME VARCHAR2(100) NOT NULL,
    IDADE NUMBER,
    PROBLEMA VARCHAR2(200),
    PRONTUARIO VARCHAR2(200)
);

CREATE TABLE PRONTUARIOS (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    PACIENTE_NOME VARCHAR2(100) NOT NULL,
    OBSERVACOES VARCHAR2(300),
    MEDICO_NOME VARCHAR2(100),
    CID VARCHAR2(20)
);

CREATE TABLE SETORES (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    ANDAR VARCHAR2(10),
    RESPONSVEL VARCHAR2(100)
);

```

Link GitHub

https://github.com/Felipesgh/Sprint4_Java

https://github.com/Felipesgh/Challenge_Sprint4_Front

Explicação do Projeto

A integração com front end eu usei apenas a

api via cep a parte de integração do crud eu fiz
com backend de python