

DOMAIN DRIVEN DESIGN USING JAVA

Faculdade de Informática e Administração Paulista

CHALLENGE 2025

2º Semestre / Sprint 4

Equipe: PGR – 1TDSPW (FIAP)

Integrantes:

Pedro Henrique de Oliveira – RM562312

Guilherme Lisboa Silva – RM565187

São Paulo, 2025

SUMÁRIO

1. Descrição da Solução
2. Principais Funcionalidades
3. Destaques Implementados
4. Protótipo (capturas de tela)
5. Modelo de Entidade-Relacionamento (MER) — DDL e especificações
6. Diagrama de Classes
7. Tabela de Endpoints
8. Links Úteis

1. Descrição da Solução

1. Objetivo e Escopo do Projeto

O HC Teleconsulta é uma solução voltada a mitigar um problema clássico do Hospital das Clínicas: o absenteísmo em consultas. Quando o paciente não comparece, o hospital sofre com ociosidade, profissionais ficam com horários ociosos, filas aumentam e a percepção de qualidade diminui. A plataforma integra cadastro, agendamento tradicional e teleconsulta, além de rotinas de lembretes e feedback pós-atendimento, tudo sustentado em um backend Quarkus/Java 21 com banco Oracle FIAP.

Problemas enfrentados

- Alto índice de não comparecimento por esquecimento ou dificuldade de confirmar presença.
- Comunicação reativa entre hospital e paciente, sem lembretes automatizados.
- Dificuldade em gerir agendas e disponibilidades em tempo real.
- Recursos desperdiçados: profissionais, salas, equipamentos ociosos.
- Serviços digitais fragmentados, sem integração entre cadastro, agenda e teleconsulta.

Objetivos da solução

- Reduzir o absenteísmo ao enviar lembretes automáticos (24h, 3h e 1h antes) e facilitar cancelamentos.
- Ampliar o acesso com teleconsulta, atendendo pacientes que não podem ir ao hospital.
- Centralizar a agenda médica em um fluxo simples, com slots controlados e reserva automática.
- Garantir confiabilidade e rastreabilidade usando banco Oracle e logs detalhados.
- Permitir evolução do ecossistema (integrações, analytics, portais) via API REST.

Como funciona

1. **Cadastro validado:** Pacientes e profissionais passam por validação de CPF, e-mail, CRM e código interno (no caso de funcionários).
2. **Agenda baseada em disponibilidade:** cada profissional publica slots livres; o sistema impede marcações em horários incompatíveis.
3. **Consultas com link automático:** teleconsultas recebem um link gerado pelo backend, dispensando input do usuário.
4. **Regras de alteração:** mudanças de tipo avançam/limpam o link, cancelamentos liberam o slot.
5. **Lembretes e feedback:** configurados para reduzir ausências e medir satisfação (estruturas já prontas na base).
6. **Arquitetura Quarkus + Oracle:** API performática, com keep-alive e endpoint /warmup para mitigar cold start no Render.

Benefícios

- **Hospital:** agenda otimizada, indicadores confiáveis, menos tempo ocioso.
- **Profissional de saúde:** visão clara da agenda, menor incidência de encaixes vazios.

- **Paciente:** facilidade para agendar/cancelar, atendimento remoto, lembretes e feedback.

2. Principais Funcionalidades

1. **Login simples (POST /auth/login)** – valida credenciais e devolve usuariold, role, pacienteld ou profissionalld. Se o usuário não estiver vinculado, retorna 400 e loga o erro.
2. **Cadastro restrito de profissionais** – exige codigoFuncionario (config app.profissional.codigo-acesso). Evita que visitantes estabeleçam perfis corporativos.
3. **Agenda baseada em disponibilidades** – slots criados via /disponibilidades e reservados automaticamente quando há consulta. Fechado a cada profissional.
4. **Consultas atreladas aos slots** – obrigam disponibilideld, validam o par paciente/profissional e mantêm o horário livre ou reservado conforme status.
5. **Teleconsulta com link automático** – em POST/PUT /consultas, se tipoConsulta = TELECONSULTA, o backend gera linkAcesso com base em app.teleconsulta.base-url. O front apenas exibe o link retornado.
6. **Keep Alive / Warmup** – um scheduler interno pinga /hello periodicamente, e o endpoint /warmup faz select 1 from dual para forçar a conexão com o Oracle após um deploy.

3. Destaques Implementados

1. Enum Role apenas com PACIENTE e PROFISSIONAL.
2. DTOs fortemente tipados (Java Records).
3. DAO manual com JDBC puro.
4. CORS configurado e reforçado com CorsFilter personalizado.
5. Script [db/ddl/load_sample_data.sql](#) com pacientes/profissionais reais e datas futuras (2030).
6. Logs adicionais para facilitar diagnóstico (ex.: login sem vínculo, preflight 401).
7. Configurações externalizadas ([app.*](#)) via variáveis de ambiente.

4. Protótipo (capturas de tela)

Página Home

Cuidando de Você, Onde Estiver

Telemedicina de excelência do Hospital das Clínicas. Consultas online com os melhores especialistas.

[Agendar Consulta →](#)[Conhecer Mais](#)**10K+**

Pacientes Atendidos

50+

Médicos Especialistas

24/7

Atendimento

98%

Satisfação

Por que Escolher o **TeleSaúde HC**?

Página de Login

Acesse o TeleSaúde HC

Faça login ou crie sua conta para continuar.

[Entrar](#)[Criar conta \(Paciente\)](#)[Criar conta \(Profissional\)](#)

E-mail

Senha

[Entrar](#)

Logins de exemplo

Paciente - Ana: ana.paciente@hcc.com / 123456

Paciente - Bruno: bruno.paciente@hcc.com / 123456

Paciente - Carla: carla.paciente@hcc.com / 123456

Profissional - Henrique: henrique.prof@hcc.com / 123456

Profissional - Marina: marina.prof@hcc.com / 123456

Página de Cadastro de Paciente e Profissional

Acesse o TeleSaúde HC

Faça login ou crie sua conta para continuar.

[Entrar](#)
[Criar conta \(Paciente\)](#)
[Criar conta \(Profissional\)](#)

Nome

E-mail

Senha

CRM

Especialidade

Código interno

Use o código fornecido pela coordenação (HC-ACCESS).

Criar conta de profissional

Logins de exemplo

Paciente - Ana: ana.paciente@hc.com / 123456

Paciente - Bruno: bruno.paciente@hc.com / 123456

Paciente - Carla: carla.paciente@hc.com / 123456

Profissional - Henrique: henrique.prof@hc.com / 123456

Profissional - Marina: marina.prof@hc.com / 123456

Acesse o TeleSaúde HC

Faça login ou crie sua conta para continuar.

[Entrar](#)
[Criar conta \(Paciente\)](#)
[Criar conta \(Profissional\)](#)

Nome

E-mail

Senha

CPF

Sexo

Data de nascimento

Telefone

Cidade

Criar conta de paciente

Logins de exemplo

Paciente - Ana: ana.paciente@hc.com / 123456

Paciente - Bruno: bruno.paciente@hc.com / 123456

Paciente - Carla: carla.paciente@hc.com / 123456

Profissional - Henrique: henrique.prof@hc.com / 123456

Profissional - Marina: marina.prof@hc.com / 123456

Página do Painel de Administração do Profissional

Disponibilidades

De Até Novo horário

Adicionar disponibilidade

DOMINGO, 09/11

09:00 [Remover](#)
11:00 [Remover](#)

SEGUNDA-FEIRA, 10/11

10:00 [Remover](#)

QUINTA-FEIRA, 13/11

09:00 [Remover](#)
10:00 [Remover](#)
11:00 [Remover](#)
12:00 [Remover](#)
13:00 [Remover](#)
14:00 [Remover](#)
15:00 [Remover](#)
16:00 [Remover](#)
17:00 [Remover](#)

SEXTA-FEIRA, 14/11

09:00 [Remover](#)
10:00 [Remover](#)
11:00 [Remover](#)
12:00 [Remover](#)
13:00 [Remover](#)

SÁBADO, 15/11

09:00 [Remover](#)
10:00 [Remover](#)
11:00 [Remover](#)
12:00 [Remover](#)
13:00 [Remover](#)

DOMINGO, 16/11

09:00 [Remover](#)
10:00 [Remover](#)
11:00 [Remover](#)
12:00 [Remover](#)
13:00 [Remover](#)

Consultas marcadas

Paciente	Horário	Link	Status	Ações
Ana Paciente	10/11/2025, 09:00	Abrir reunião	CANCELADA	<div>Cancelada</div> <div>Detalhes</div>
Carla Paciente	10/11/2025, 10:00	Abrir reunião	REALIZADA	<div>Realizada</div> <div>Detalhes</div>
Ana Paciente	10/11/2025, 11:00	Abrir reunião	CANCELADA	<div>Cancelada</div> <div>Detalhes</div>
Carla Paciente	10/11/2025, 12:00	Abrir reunião	REALIZADA	<div>Realizada</div> <div>Detalhes</div>
Elisa Prado	10/11/2025, 13:00	Abrir reunião	REALIZADA	<div>Realizada</div> <div>Detalhes</div>
Ana Paciente	10/11/2025, 13:00	Abrir reunião	CANCELADA	<div>Cancelada</div> <div>Detalhes</div>

Página 1 de 8

Anterior

Próxima

Página do Painel do Paciente

Calendário de disponibilidades

Profissional

Dr. Henrique Souza

De

09/11/2025

Até

16/11/2025

DOMINGO, 09/11

09:00

Selecionar

11:00

Selecionar

SEGUNDA-FEIRA, 10/11

10:00

Selecionar

QUINTA-FEIRA, 13/11

09:00

Selecionar

10:00

Selecionar

11:00

Selecionar

12:00

Selecionar

13:00

Selecionar

14:00

Selecionar

15:00

Selecionar

16:00

Selecionar

17:00

Selecionar

SEXTA-FEIRA, 14/11

09:00

Selecionar

10:00

Selecionar

11:00

Selecionar

12:00

Selecionar

13:00

Selecionar

14:00

Selecionar

15:00

Selecionar

SÁBADO, 15/11

09:00

Selecionar

10:00

Selecionar

11:00

Selecionar

12:00

Selecionar

13:00

Selecionar

14:00

Selecionar

15:00

Selecionar

Consultas agendadas

Horário	Profissional	Link	Status	Ações
10/11/2025, 09:00	Thiago Cruz	Abrir reunião	REALIZADA	Detalhes Cancelar
10/11/2025, 09:00	Dr. Henrique Souza	Abrir reunião	CANCELADA	Detalhes
10/11/2025, 11:00	Thiago Cruz	Abrir reunião	CANCELADA	Detalhes
10/11/2025, 11:00	Dr. Henrique Souza	Abrir reunião	CANCELADA	Detalhes
10/11/2025, 12:00	Thiago Cruz	Abrir reunião	REALIZADA	Detalhes Cancelar

Página 1 de 6

Anterior

Próxima

Página de Detalhes da Consulta

TeleSaúde

HC

Home

Sobre

FAQ

Integrantes

Contato

Ana Paciente

Paciente

Meu painel

Sair

Consulta #83

Voltar

PACIENTE

Ana Paciente

PROFISSIONAL

Thiago Cruz

HORÁRIO

10/11/2025, 09:00

TIPO

Teleconsulta

STATUS

REALIZADA

LINK DE ACESSO

[Abrir reunião](#)

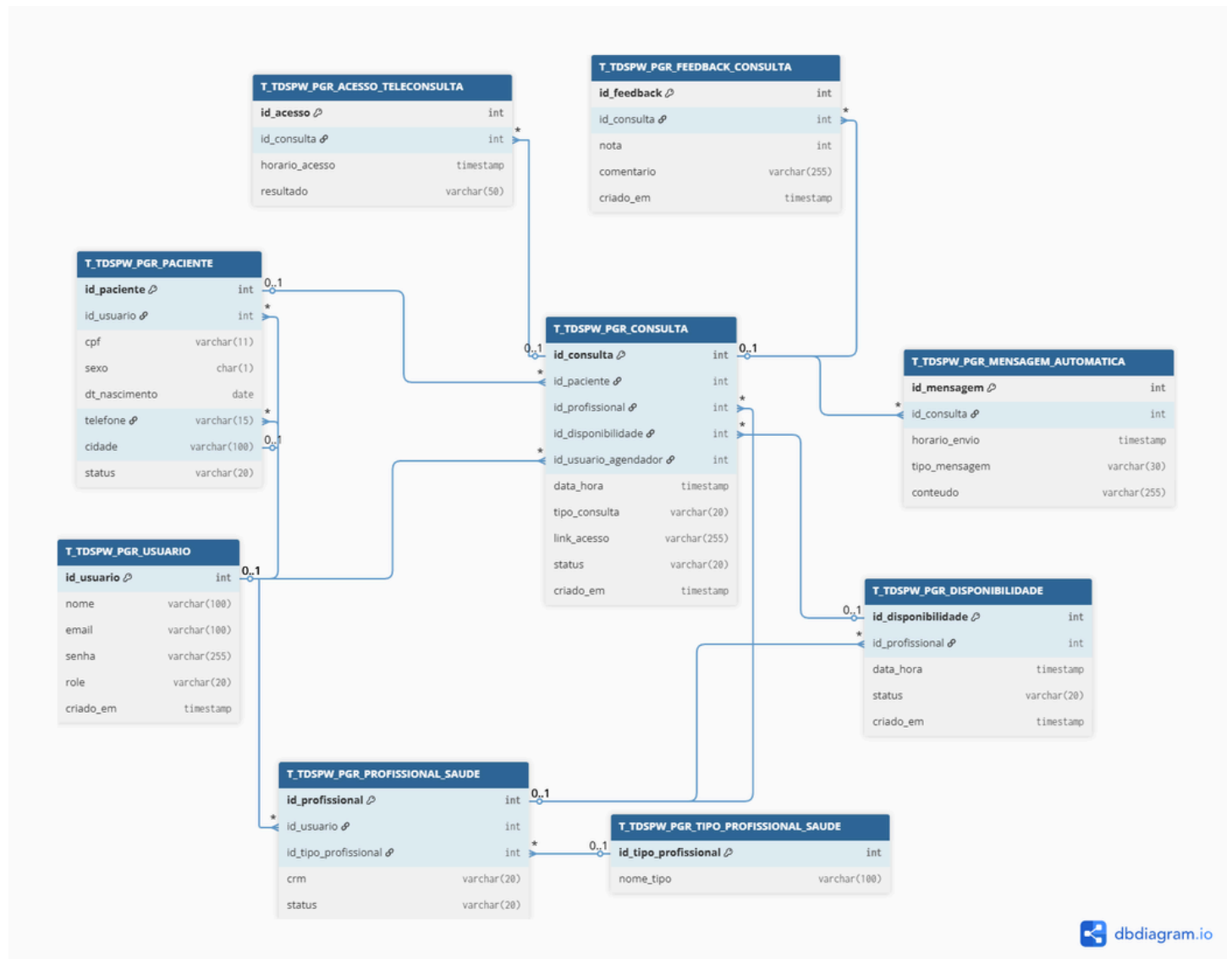
Atualizar status

Realizada

Salvar

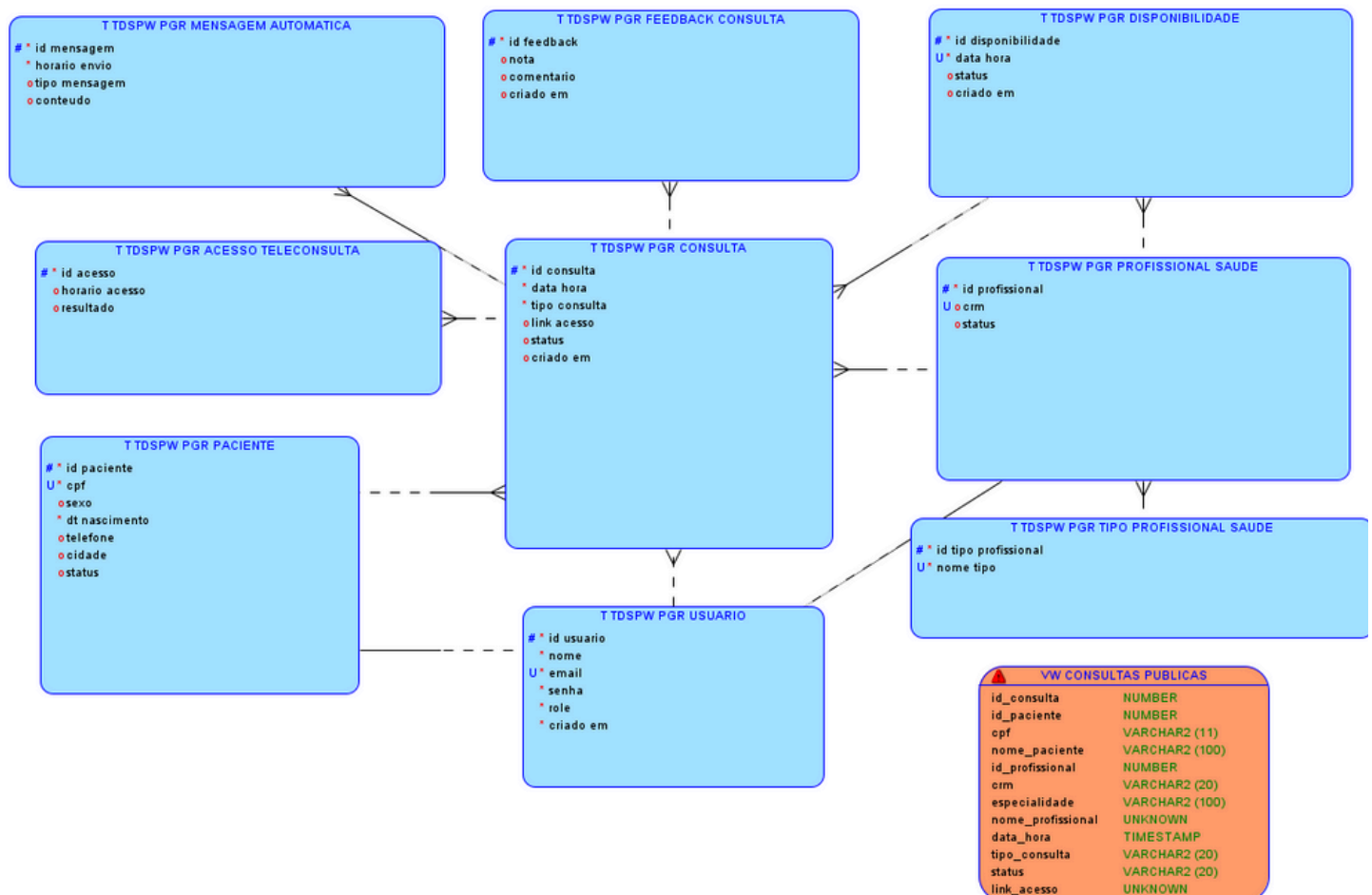
5. Modelo de Entidade-Relacionamento (MER) — DDL e especificações

5.1 Modelo Conceitual (DER)



- Entidades principais: Usuário, Paciente, Tipo Profissional, Profissional, Disponibilidade, Consulta, Mensagem, Acesso Teleconsulta, Feedback.
- Relacionamentos:
 - Usuário 1:1 Paciente / Usuário 1:1 Profissional.
 - Profissional 1:N Disponibilidades.
 - Disponibilidade 1:1 Consulta (via FK).
 - Consulta 1:N Mensagens, 1:N Acessos, 1:1 Feedback (max).
- Justificativa: cada contexto (comunicação, acesso, satisfação) é dependente da consulta, evitando duplicidade de FK espalhada.

5.2 Modelo Lógico



- Cada entidade conceitual virou uma tabela física conforme a documentação do DDL.
- Todos os relacionamentos fortes são implementados por FKs obrigatórias, exceto id_usuario_agendador (opcional).
- Restrições de domínio foram mapeadas com CHECK (role, status, tipo_consulta, etc).
- O padrão de identidades substitui sequences em Oracle 21c+, conforme create_pgr_sprint4.sql.

5.3 Modelo Relacional (Resumo)

);

CREATE TABLE T_TDSPW_PGR_PACIENTE (

id_paciente NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

id_usuario NUMBER NOT NULL UNIQUE,

cpf VARCHAR2(11) NOT NULL UNIQUE,

sexo CHAR(1) DEFAULT NULL CHECK (sexo IN ('M','F')),

dt_nascimento DATE NOT NULL,

telefone VARCHAR2(15),

cidade VARCHAR2(100),

status VARCHAR2(20) DEFAULT 'ATIVO' CHECK (status IN ('ATIVO','INATIVO')),

CONSTRAINT fk_paciente_usuario FOREIGN KEY (id_usuario)

REFERENCES T_TDSPW_PGR_USUARIO(id_usuario)

);

CREATE TABLE T_TDSPW_PGR_TIPO_PROFISSIONAL_SAUDE (

id_tipo_profissional NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nome_tipo VARCHAR2(100) NOT NULL UNIQUE

);

CREATE TABLE T_TDSPW_PGR_PROFISSIONAL_SAUDE (

id_profissional NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

id_usuario NUMBER NOT NULL UNIQUE,

id_tipo_profissional NUMBER NOT NULL,

crm VARCHAR2(20) UNIQUE,

status VARCHAR2(20) DEFAULT 'ATIVO' CHECK (status IN ('ATIVO','INATIVO')),

CONSTRAINT fk_profissional_usuario FOREIGN KEY (id_usuario)

REFERENCES T_TDSPW_PGR_USUARIO(id_usuario),

CONSTRAINT fk_profissional_tipo FOREIGN KEY (id_tipo_profissional)

REFERENCES T_TDSPW_PGR_TIPO_PROFISSIONAL_SAUDE(id_tipo_profissional)

);

CREATE TABLE T_TDSPW_PGR_DISPONIBILIDADE (

id_disponibilidade NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

id_profissional NUMBER NOT NULL,

data_hora TIMESTAMP NOT NULL,

status VARCHAR2(20) DEFAULT 'LIVRE' CHECK (status IN ('LIVRE','RESERVADA')),

criado_em TIMESTAMP DEFAULT SYSTIMESTAMP,

CONSTRAINT fk_disponibilidade_profissional FOREIGN KEY (id_profissional)

REFERENCES T_TDSPW_PGR_PROFISSIONAL_SAUDE(id_profissional),

CONSTRAINT uk_disponibilidade_profissional_data UNIQUE (id_profissional, data_hora)

);

CREATE TABLE T_TDSPW_PGR_CONSULTA (

id_consulta NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

id_paciente NUMBER NOT NULL,

id_profissional NUMBER NOT NULL,

id_disponibilidade NUMBER NOT NULL,

id_usuario_agendador NUMBER,

data_hora TIMESTAMP NOT NULL,

tipo_consulta VARCHAR2(20) NOT NULL CHECK (tipo_consulta IN ('PRESENCIAL','TELECONSULTA')),

link_acesso VARCHAR2(255),

status VARCHAR2(20) DEFAULT 'AGENDADA' CHECK (status IN ('AGENDADA','REALIZADA','CANCELADA','FALTOU'));

```

criado_em      TIMESTAMP DEFAULT SYSTIMESTAMP,

CONSTRAINT fk_consulta_paciente FOREIGN KEY (id_paciente)

REFERENCES T_TDSPW_PGR_PACIENTE(id_paciente),

CONSTRAINT fk_consulta_profissional FOREIGN KEY (id_profissional)

REFERENCES T_TDSPW_PGR_PROFISSIONAL_SAUDE(id_profissional),

CONSTRAINT fk_consulta_disponibilidade FOREIGN KEY (id_disponibilidade)

REFERENCES T_TDSPW_PGR_DISPONIBILIDADE(id_disponibilidade),

CONSTRAINT fk_consulta_agendador FOREIGN KEY (id_usuario_agendador)

REFERENCES T_TDSPW_PGR_USUARIO(id_usuario)

);

```

```

-----

CREATE TABLE T_TDSPW_PGR_MENSAGEM_AUTOMATICA (

    id_mensagem  NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

    id_consulta  NUMBER NOT NULL,

    horario_envio  TIMESTAMP NOT NULL,

    tipo_mensagem VARCHAR2(30) CHECK (tipo_mensagem IN

('LEMBRETE_24H','LEMBRETE_3H','LEMBRETE_1H','CONFIRMACAO','AGRADECIMENTO')),

    conteudo      VARCHAR2(255),

    CONSTRAINT fk_mensagem_consulta FOREIGN KEY (id_consulta)

REFERENCES T_TDSPW_PGR_CONSULTA(id_consulta)

);

```

```

-----

CREATE TABLE T_TDSPW_PGR_FEEDBACK_CONSULTA (

    id_feedback  NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

    id_consulta  NUMBER NOT NULL,

    nota         NUMBER CHECK (nota BETWEEN 1 AND 5),

    comentario   VARCHAR2(255),

```

```
criado_em    TIMESTAMP DEFAULT SYSTIMESTAMP,  
  
CONSTRAINT fk_feedback_consulta FOREIGN KEY (id_consulta)  
  
REFERENCES T_TDSPW_PGR_CONSULTA(id_consulta)  
  
);  
-----
```

```
CREATE TABLE T_TDSPW_PGR_ACESSO_TELECONSULTA (  
  
    id_acesso  NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
  
    id_consulta NUMBER NOT NULL,  
  
    horario_acesso TIMESTAMP DEFAULT SYSTIMESTAMP,  
  
    resultado  VARCHAR2(50),  
  
    CONSTRAINT fk_acesso_consulta FOREIGN KEY (id_consulta)  
  
        REFERENCES T_TDSPW_PGR_CONSULTA(id_consulta)  
  
);  
-----
```

```
CREATE OR REPLACE VIEW VW_CONSULTAS_PUBLICAS AS
```

```
SELECT
```

```
    c.id_consulta,  
  
    p.id_paciente,  
  
    p.cpf,  
  
    pu.nome AS nome_paciente,  
  
    pr.id_profissional,  
  
    pr.crm,  
  
    tp.nome_tipo AS especialidade,  
  
    pru.nome AS nome_profissional,  
  
    c.data_hora,  
  
    c.tipo_consulta,  
  
    c.status,  
  
    CASE WHEN c.tipo_consulta = 'TELECONSULTA' THEN c.link_acesso END AS link_acesso
```



```

FROM T_TDSPW_PGR_CONSULTA c

JOIN T_TDSPW_PGR_PACIENTE p ON p.id_paciente = c.id_paciente

JOIN T_TDSPW_PGR_USUARIO pu ON pu.id_usuario = p.id_usuario

JOIN T_TDSPW_PGR_PROFISSIONAL_SAUDE pr ON pr.id_profissional = c.id_profissional

JOIN T_TDSPW_PGR_USUARIO pru ON pru.id_usuario = pr.id_usuario

JOIN T_TDSPW_PGR_TIPO_PROFISSIONAL_SAUDE tp ON tp.id_tipo_profissional = pr.id_tipo_profissional;

```

I

- Arquivo: drop_pgr_sprint4.sql.

```
-----
```

```
-- DROP
```

```
-----
```

```

DROP VIEW VW_CONSULTAS_PUBLICAS;

DROP TABLE T_TDSPW_PGR_ACESSO_TELECONSULTA CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_FEEDBACK_CONSULTA CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_MENSAGEM_AUTOMATICA CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_CONSULTA CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_DISPONIBILIDADE CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_PROFISSIONAL_SAUDE CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_TIPO_PROFISSIONAL_SAUDE CASCADE CONSTRAINTS;

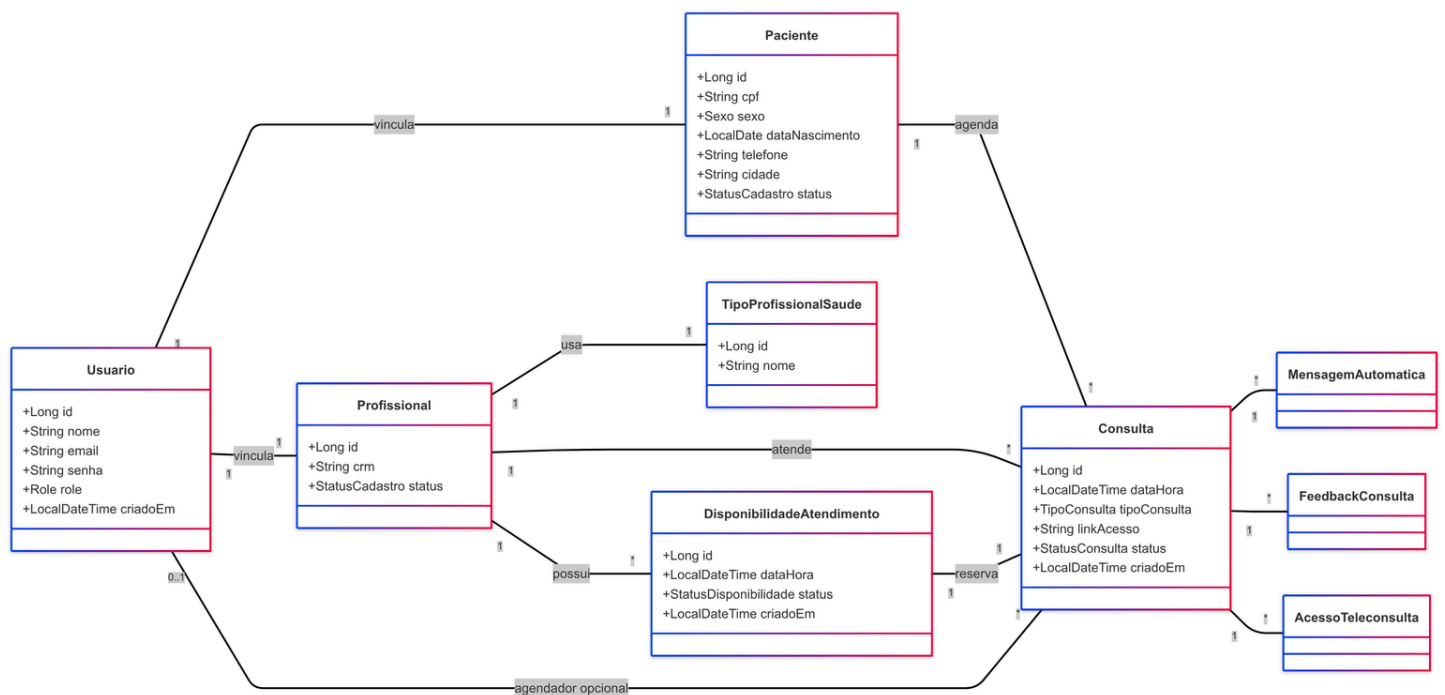
DROP TABLE T_TDSPW_PGR_PACIENTE CASCADE CONSTRAINTS;

DROP TABLE T_TDSPW_PGR_USUARIO CASCADE CONSTRAINTS;

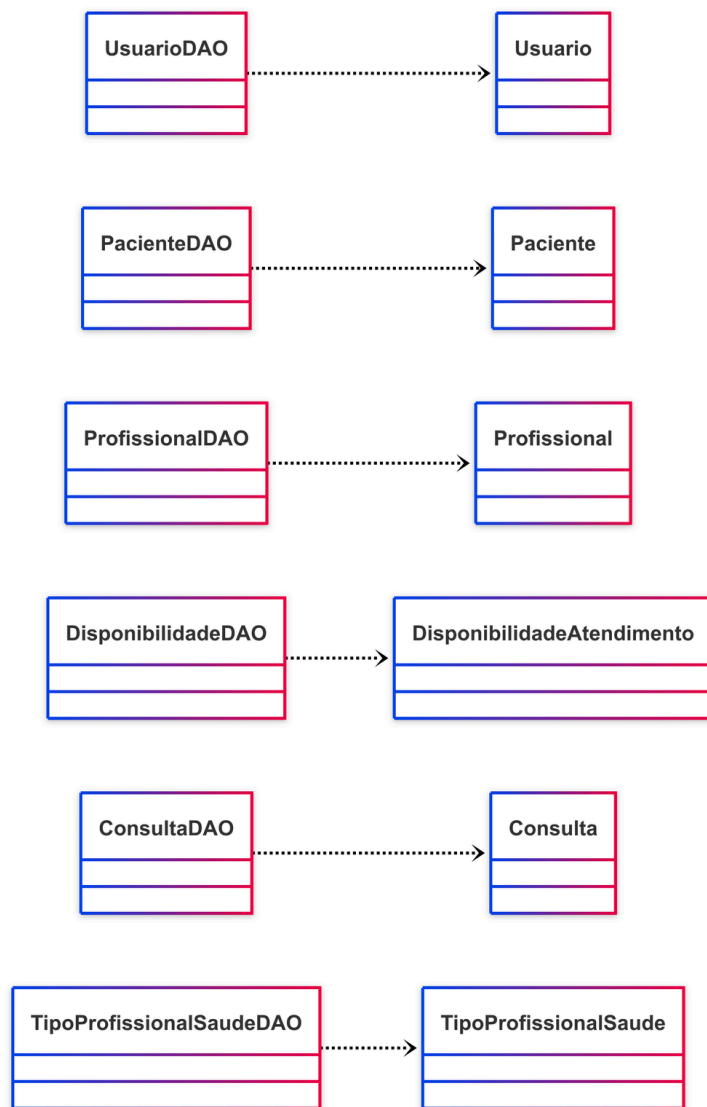
```

6. Diagrama de Classes

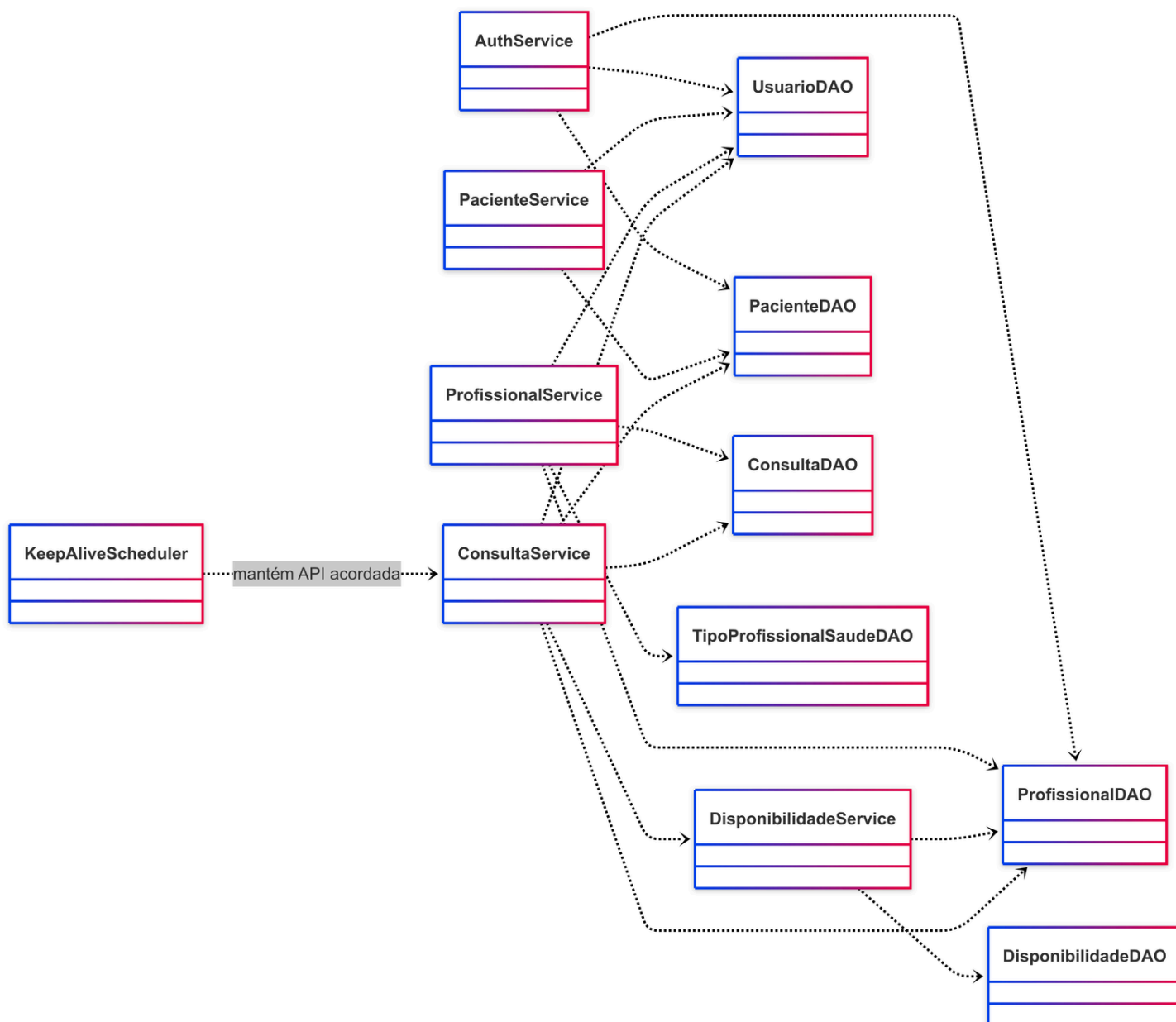
1. Entidades de Domínio



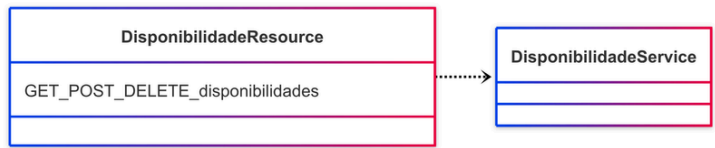
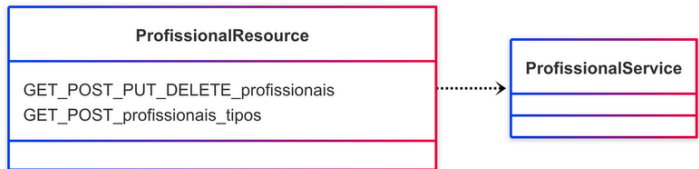
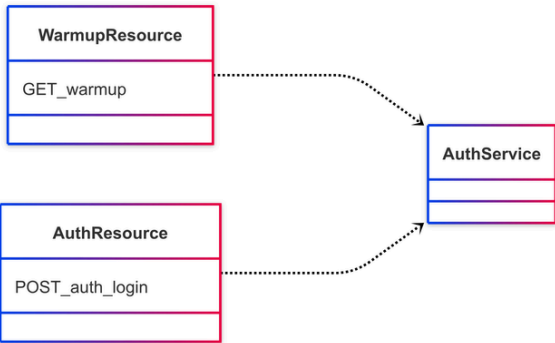
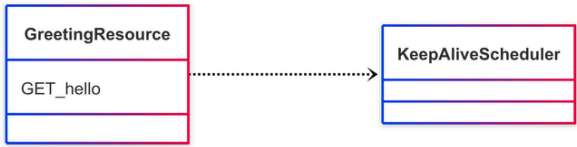
2. DAO → ModelosDAO → Modelos



3. Serviços



4. Resources (REST)



7. Tabela de Endpoints

Método	URI	Descrição	Status esperados
GET	/hello	Health check simples	200
GET	/warmup	Inicializa datasource (select 1 em dual)	204, 503
POST	/auth/login	Login com email/senha	200, 400
GET	/pacientes	Lista pacientes	200
GET	/pacientes/{id}	Detalha paciente	200, 404
POST	/pacientes	Cria paciente	201, 400
PUT	/pacientes/{id}	Atualiza paciente	200, 400, 404
DELETE	/pacientes/{id}	Remove paciente (se sem consultas)	204, 400
GET	/profissionais	Lista profissionais	200
GET	/profissionais/{id}	Detalha profissional	200, 404
POST	/profissionais	Cria profissional (exige codigoFuncionario)	201, 400
PUT	/profissionais/{id}	Atualiza profissional	200, 400, 404
DELETE	/profissionais/{id}	Remove profissional (se sem consultas)	204, 400
GET	/profissionais/tipos	Lista tipos de profissional	200
POST	/profissionais/tipos	Cria tipo de profissional	201, 400
GET	/disponibilidades?profissionalId=...	Lista slots livres futuros (filtro obrigatório)	200, 400
POST	/disponibilidades	Cadastra disponibilidade	201, 400
DELETE	/disponibilidades/{id}	Remove disponibilidade livre	204, 400,

			404
GET	/consultas?pacientId=...	Lista consultas de um paciente	200, 400
GET	/consultas?profissionalId=...	Lista consultas de um profissional	200, 400
GET	/consultas/{id}	Detalha consulta	200, 404
POST	/consultas	Cria consulta (link gerado automaticamente em teleconsulta)	201, 400
PUT	/consultas/{id}	Atualiza consulta (gera/zera link conforme tipo)	200, 400, 404
PUT	/consultas/{id}/status	Atualiza status (libera slot se cancelada)	200, 404
DELETE	/consultas/{id}	Remove consulta	204, 404

8. Links Utéis

- API em produção: <https://hc-teleconsulta-api-java-1.onrender.com>
- Health check (/hello): <https://hc-teleconsulta-api-java-1.onrender.com/hello>
- Warmup (inicializa datasource): <https://hc-teleconsulta-api-java-1.onrender.com/warmup>
- Swagger/OpenAPI: <https://hc-teleconsulta-api-java-1.onrender.com/q/swagger-ui>
- Repositório da API: <https://github.com/challenge-pmg/hc-teleconsulta-api-java>
- Front-end em produção: <https://hc-teleconsulta-pg.vercel.app>
- Scripts SQL: db/ddl/create_pgr_sprint4.sql, db/ddl/drop_pgr_sprint4.sql, db/ddl/load_sample_data.sql

Comandos para validação rápida:

* curl -I -X OPTIONS <https://hc-teleconsulta-api-java-1.onrender.com/auth/login> -H "Origin: http://localhost:5173"

* curl -X GET <https://hc-teleconsulta-api-java-1.onrender.com/warmup>