

Apoia+

Nome do projeto: Apoia+

Nome da solução (sistema desenvolvido): Plataforma Digital Acessível para Redução de Absenteísmo em Tele consultas.

Disciplina: Domain Driven Design using Java

Turma: 1TDSPO

Integrantes:

Enzo Monteiro Maciel - RM563734

Gabriel Cabral Mendes Mariano - RM563230

Matheus de Almeida Sousa - RM563557

Sumário

1.	Objetivo e Escopo do Projeto	3
2.	Principais Funcionalidades.....	3
3.	Tabela de Endpoints (API RESTful)	4
4.	Protótipo — Prints das Telas Implementadas	4
5.	Modelo Entidade-Relacionamento (MER)	6
6.	Diagrama de Classes	6
7.	Boas Práticas Aplicadas	6

1. Objetivo e Escopo do Projeto

O Apoia+ é uma API em Java 17 com Javalin 5 que integra o frontend ao backend para reduzir faltas (no-show) em teleconsultas. O escopo contempla recursos de Pacientes, Consultas e Lembretes; regras de negócio para confirmação/ausência e geração automática de lembretes em T-24h, T-2h e T-15min; armazenamento H2 em memória.

2. Principais Funcionalidades

- - CRUD completo de Pacientes e Consultas.
- - Confirmação de presença e marcação de ausência.
- - Geração automática de lembretes (T-24h, T-2h, T-15min).
- - Seed de dados e bootstrap automático do banco (H2).
- - Tratamento global de erros (400/404/500) e logs estruturados.
- - CORS habilitado para integração direta com o Front-End.
- - Health-check (/healthz) para monitoramento em produção (Render).

3. Tabela de Endpoints (API RESTful)

Recurso	Verbo	URI	Descrição	Status
Health	GET	/healthz	Verifica disponibilidade	200
Info API	GET	/api/	Informações da API	200
Pacientes	GET	/api/pacientes	Lista pacientes	200
Pacientes	POST	/api/pacientes	Cria paciente	201/400
Pacientes	GET	/api/pacientes/{id}	Busca paciente	200/404
Pacientes	PUT	/api/pacientes/{id}	Atualiza paciente	204/404
Pacientes	DELETE	/api/pacientes/{id}	Remove paciente	204/404
Consultas	GET	/api/consultas	Lista consultas	200
Consultas	POST	/api/consultas	Cria consulta	201/400
Consultas	GET	/api/consultas/{id}	Busca consulta	200/404
Consultas	PUT	/api/consultas/{id}	Atualiza consulta	204/404
Consultas	DELETE	/api/consultas/{id}	Remove consulta	204/404
Consultas	POST	/api/consultas/{id}/confirmar	Confirma presença	200/404
Consultas	POST	/api/consultas/{id}/ausente	Marca ausente	200/404
Lembretes	POST	/api/consultas/{id}/lembretes	Gera lembretes T-24h/T-2h/T-15min	200/404
Lembretes	GET	/api/consultas/{id}/lembretes	Lista lembretes da consulta	200/404

4. Protótipo — Prints das Telas Implementadas

The screenshot shows a web-based application interface for managing patients. At the top, there's a navigation bar with links for 'Apoia+', 'Pacientes', and 'Consultas'. The 'Consultas' link is underlined, indicating it's the active section. On the left, there's a sidebar with a '+ Novo Paciente' button. The main form area has fields for 'Nome' (Name) containing 'Maria Souza', 'E-mail' (Email) containing 'maria@email.com', and a dropdown for 'Acessível por ferramentas digitais' (Accessible via digital tools) set to 'Sim' (Yes). A blue 'Salvar' (Save) button is at the bottom left.

Apoia+ Pacientes Consultas

Pacientes

+ Novo Paciente

Nome

Maria Souza

E-mail

maria@email.com

Acessível por ferramentas digitais

Sim

Salvar

Pacientes

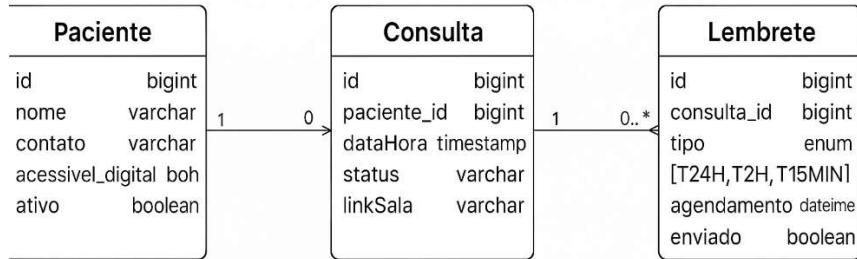
ID	Nome	Contato	Acessível p/ Digital	Ações
1	Maria Souza	maria@email.com	Sim	Editar Excluir
2	João Silva	joao@email.com	Não	Editar Excluir
3	Ana Costa	ana@email.com	Não	Editar Excluir
4	Pedro Lima	pedro@email.com	Sim	Editar Excluir
5	Jullana Oliveira	jullana@email.com	Sim	Editar Excluir
6	Carlos Pereira	carlos@email.com	Não	Editar Excluir
7	Renata Santos	renata@email.com	Sim	Editar Excluir
8	Lucas Almeida	lucas@email.com	Não	Editar Excluir
9	Fernanda Rocha	fernanda@email.com	Sim	Editar Excluir
10	Rafael Martins	rafael@email.com	Sim	Editar Excluir

Editar Paciente

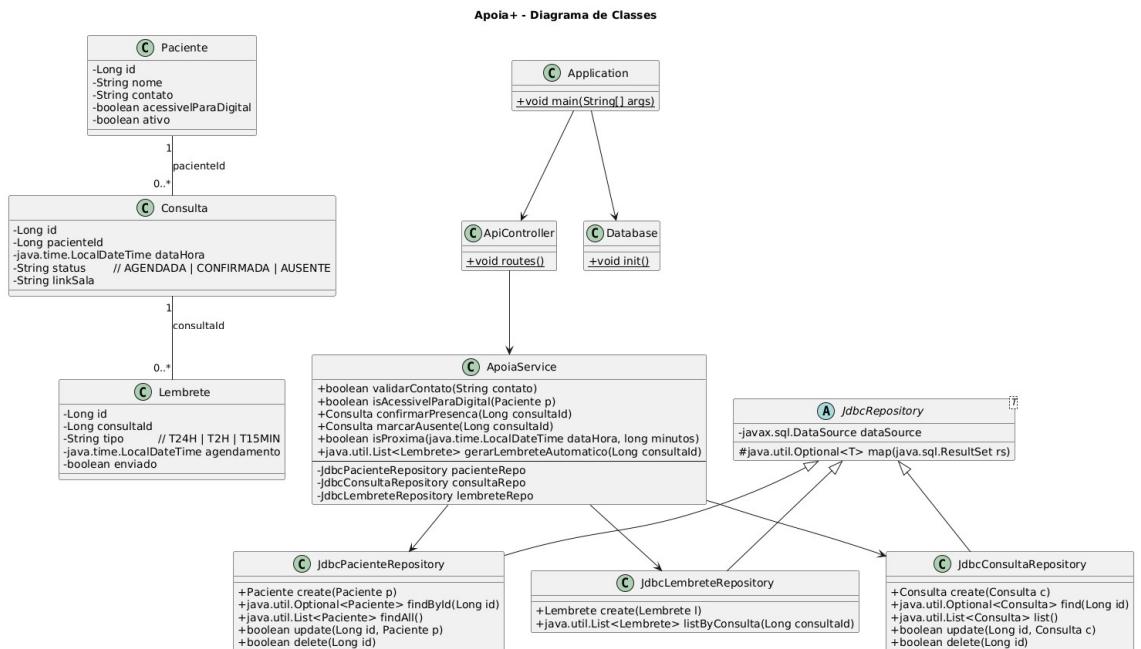
Nome Contato Acessível p/Digital [Cancelar](#)[Salvar](#)

5. Modelo Entidade-Relacionamento (MER)

MER simplificado do H2 (homologação). Chaves estrangeiras e cardinalidades conforme repositórios JDBC



6. Diagrama de Classes



7. Boas Práticas Aplicadas

- Padrão em camadas: controller → service → repository → model.
- Status HTTP consistentes e handler global de exceções.
- Validações de entrada no service/controller.
- Logs via SLF4J e endpoint de health-check.

- CORS habilitado para o front.
- Seed/H2 para homologação rápida.
- Empacotamento Maven + Dockerfile (Render).