

Driven Soft

Projeto para diminuição da taxa de absenteísmo do
Hospital das Clínicas

Faculdade de Informática e Administração Paulista (FIAP)
Análise e Desenvolvimento de Sistemas

Nome da solução: HC Auxilia API

Integrantes:

- Max Hayashi Batista - RM563717
- Felipe Bezerra Beatriz - RM564723
- Henrique Cunha Torres - RM565119

São Paulo

2025

Sumário

Capa.....	Página 1
Sumário.....	Página 2
Descritivo.....	Página 3
Tabela de Endpoints (API).....	Página 4
1.1 Tabela de Endpoints (API).....	Página 5
Protótipo.....	Página 6
MER & Diagrama de Classe do Projeto.....	Página 7
Procedimentos para rodar a aplicação.....	Página 8

Descritivo

O avanço da tecnologia trouxe novas possibilidades para o setor de saúde, especialmente no atendimento e na comunicação entre instituições médicas e pacientes. Mesmo assim, muitos usuários ainda enfrentam dificuldades em receber informações importantes, como avisos, lembretes e orientações de acompanhamento, o que resulta em atrasos, esquecimentos e falta de retorno aos serviços médicos.

Pensando nesse cenário, o HC Auxilia foi desenvolvido como uma solução prática e acessível para centralizar e simplificar a comunicação entre o hospital e seus pacientes. A aplicação oferece uma interface leve e intuitiva, permitindo que usuários se inscrevam para receber notificações relevantes por SMS ou WhatsApp, garantindo que nenhum aviso importante seja perdido.

A solução é composta por uma API RESTful desenvolvida em Java, integrada ao banco de dados Oracle, responsável por armazenar e gerir as informações dos inscritos. O sistema aplica validações, mantém registros organizados e permite futuras expansões para módulos como consultas, receitas, exames e cadastro de pacientes.

Entre as funcionalidades implementadas, destacam-se:

- Cadastro de novos inscritos para recebimento de notificações
- Armazenamento de preferências de contato (SMS e/ou WhatsApp)
- Validação de dados enviados pelo usuário
- Registro automático da data de inscrição
- Possibilidade de consulta, listagem e exclusão de inscrições
- Operações seguindo o padrão REST (GET, POST, DELETE)
- Persistência das informações em banco Oracle

A arquitetura do sistema foi estruturada de forma modular, com pacotes organizados para modelos, controle, serviços e persistência, facilitando manutenção e evolução do código. Para garantir integração confiável com o banco de dados, a aplicação utiliza uma classe centralizada de conexão (ConnectionFactory), responsável por gerenciar o acesso ao Oracle.

Com isso, o HC Auxilia estabelece a base de um ecossistema escalável, capaz de evoluir para um sistema hospitalar completo, mantendo o foco na comunicação eficiente e no suporte ao paciente.

Tabela de Endpoints (API)

Tabela – ExameResource (/exames)

Método HTTP	URI	Descrição	Body (JSON)	Códigos de Resposta
GET	/exames	Lista todos os exames	Não	200 OK (lista de exames) 500 INTERNAL SERVER ERROR
GET	/exames/{id}	Busca um exame por ID	Não	200 OK (encontrado) 404 NOT FOUND (não encontrado) 500 INTERNAL SERVER ERROR
POST	/exames	Salva um novo exame	Sim (Exame)	201 CREATED 500 INTERNAL SERVER ERROR
PUT	/exames	Atualiza um exame existente	Sim (Exame)	204 NO CONTENT 500 INTERNAL SERVER ERROR
DELETE	/exames/{id}	Remove exame pelo ID	Não	204 NO CONTENT 500 INTERNAL SERVER ERROR
PUT	/exames/{id}/finalizar	Finaliza um exame	Não	204 NO CONTENT 500 INTERNAL SERVER ERROR
OPTIONS	/exames	Verifica opções e CORS	Não	204 NO CONTENT

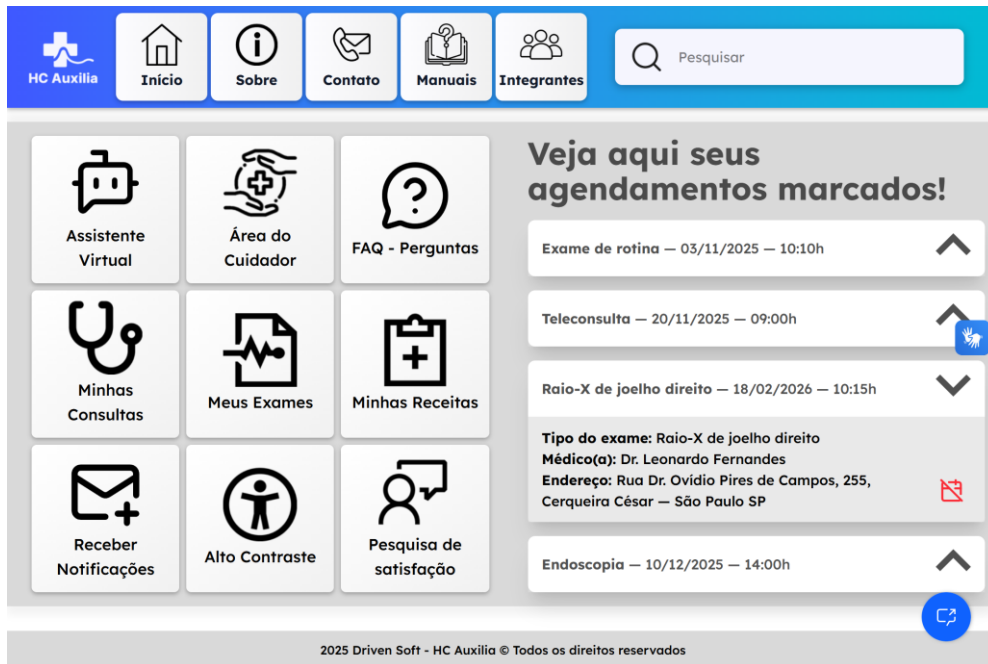
Tabela – FeedbackResource (/feedbacks)

Método HTTP	URI	Descrição	Body (JSON)	Códigos de Resposta
POST	/feedbacks	Registra um novo feedback	Sim (Feedback)	201 CREATED 400 BAD REQUEST (dados inválidos) 500 INTERNAL SERVER ERROR
GET	/feedbacks	Lista todos os feedbacks	Não	200 OK 500 INTERNAL SERVER ERROR
DELETE	/feedbacks/{codigoHash}	Remove um feedback pelo identificador	Não	204 NO CONTENT 500 INTERNAL SERVER ERROR
OPTIONS	/feedbacks	Verifica opções disponíveis	Não	204 NO CONTENT

Tabela – InscritoNotificacaoResource (/contato/notificacoes)

Método HTTP	URI	Descrição	Body (JSON)	Códigos de Resposta
POST	/contato/notificacoes	Registra e-mail para notificações	Sim (InscritoNotificacao)	201 CREATED 500 INTERNAL SERVER ERROR

Protótipo



Agendamentos - (Após a conexão do frontend com a API hospedada no Render, a tela inicial faz uma requisição aos endpoints de /exames para carregar os exames do paciente e exibir em forma de lista ou cards. Caso ele queira excluir um exame, o frontend envia uma requisição DELETE para a API, e ao receber o retorno de sucesso, os dados são atualizados automaticamente na interface.)

Inscreva-se para receber notificações de agendamentos!

Ao se inscrever no nosso sistema de notificações, você será avisado(a) de todos seus agendamentos e teleconsultas marcados ao menos uma semana antes e um dia anterior.

Nome

Telefone (apenas números)

☐ Receber via SMS

* As notificações via SMS funcionam sem internet

☐ Receber via WhatsApp

Inscrever-se

Pesquisa de Satisfação

Conte-nos mais!

Digite seu nome:

Digite seu e-mail:

Qual seu nível de satisfação com o site?

1 2 3 4 5 6 7 8 9 10

Qual sua sugestão?

Enviar

Voltar

Notificações - (Ao cadastrar um e-mail ou telefone para receber alertas, o frontend envia um POST /contato/notificacoes para a API.)

Feedbacks - (Quando o usuário envia uma avaliação dentro do sistema, o frontend envia os dados para o backend utilizando POST /feedbacks.)

MER

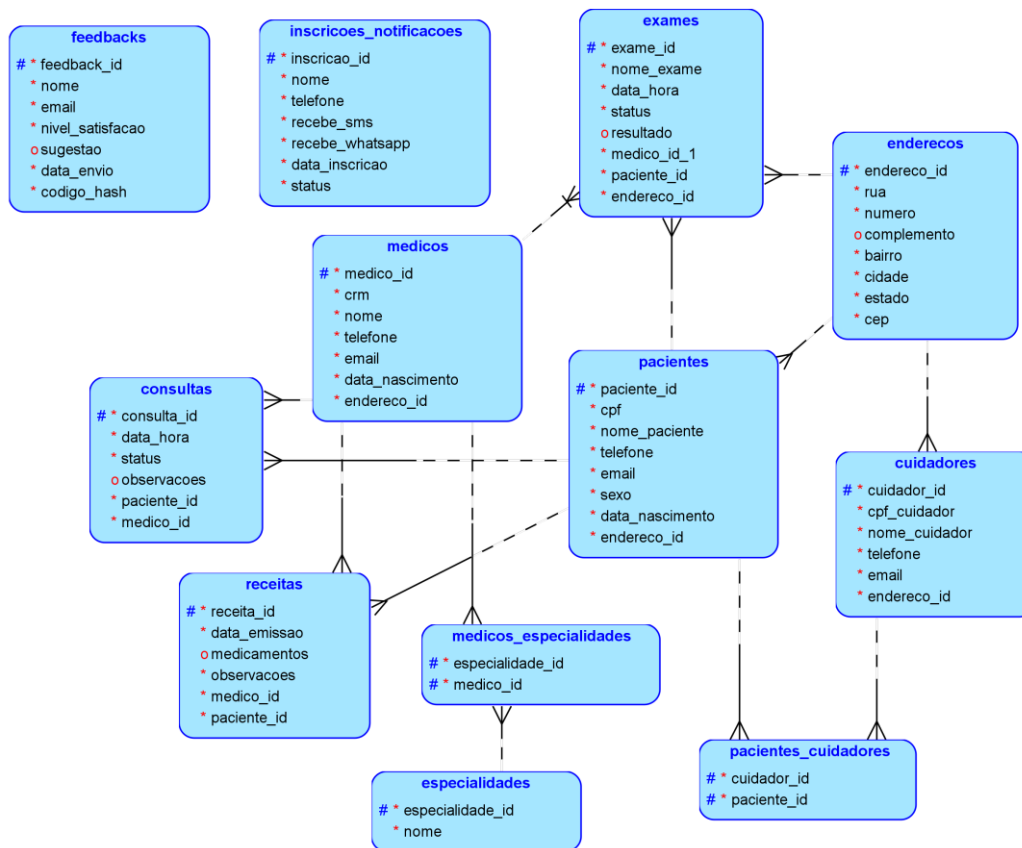
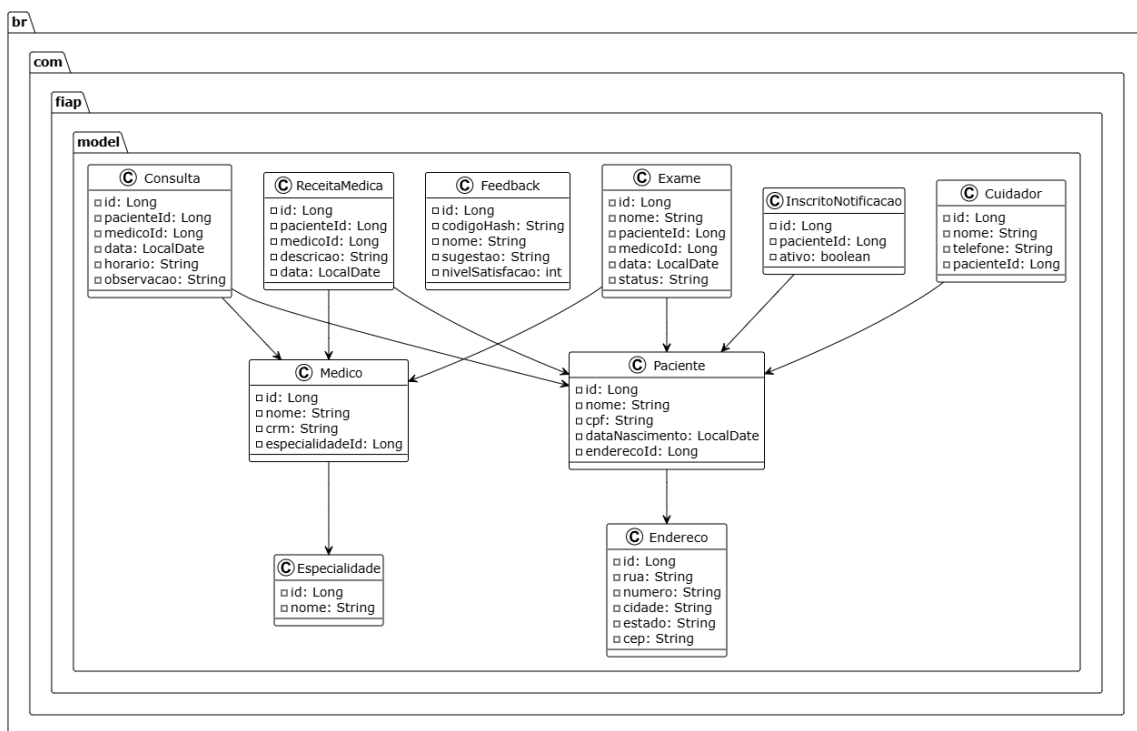


Diagrama de Classes do Projeto



Procedimentos

1. Ferramentas utilizadas: IntelliJ, Oracle Developer, Postman, Git e Render
2. Tecnologias utilizadas: Quarkus, Java, Maven, Rest Jackson e Oracle JDBC
3. Versões: Java JDK(17) & Oracle SQL Developer(19)
4. 1º Deve-se clonar o projeto feito no repositório do GitHub: `git clone https://github.com/Driven-Soft/HC-Auxilia-API`

2º Após isso deve-se entrar dentro dele: `cd HC-Auxilia-API`

3º Dentro do projeto é necessário inicializar o Maven através do comando: `mvn quarkus:dev`

4º Após isso dentro do link: <http://localhost:8080>, será possível fazer alterações no banco de dados com o Postman.

5º Por fim, A aplicação está organizada em 5 pacotes principais dentro de `br.com.fiap`:
 - `resource` → disponibiliza os endpoints da API
 - `model` → classes de domínio e entidades
 - `business` → regras de negócio, validações e verificações
 - `repository` → conexão com o banco e operações CRUD
 - `view` → objetos de transferência (DTOs) utilizados nas respostas