

Invox

Bruno Ferreira
Gabriel Robertoni Padilha
Leonardo Aragaki Rodrigues

Sumário

1. Objetivo e Escopo do Projeto	2
2. Descrição das Funcionalidades	3
3. Tabela de Endpoints (API RESTful).....	5
4. Protótipo do Sistema	5
5. Modelo de Entidade-Relacionamento (MER)	6
6. Procedimentos para Rodar a Aplicação	7

1. Objetivo e Escopo do Projeto

Nosso projeto foi pensado para ajudar a reduzir as faltas em consultas online no Hospital das Clínicas, especialmente para pacientes mais idosos ou com algum tipo de deficiência. A ideia é oferecer um site simples e fácil de utilizar, que simplifique o acesso dos pacientes às teleconsultas. Nesse site, eles encontram um quiz interativo que verifica, de maneira prática e didática, se estão preparados para realizar uma consulta remota, avaliando pontos como conexão à internet, funcionamento da câmera e adequação do ambiente em termos de iluminação e silêncio.

Além do quiz, o sistema conta com um chatbot para responder dúvidas frequentes e com a possibilidade de redirecionamento para o WhatsApp, onde o paciente poderá conversar diretamente com um atendente humano, caso necessite de ajuda personalizada. Essa combinação garante que o usuário nunca esteja sozinho no processo, sendo acompanhado passo a passo até se sentir seguro para participar da consulta.

Nosso foco é tornar o processo mais acessível, promover a inclusão digital e dar mais autonomia e confiança aos pacientes, contribuindo para que estejam devidamente preparados antes de acessar a plataforma oficial de teleconsulta. Importante destacar que o sistema não substitui a ferramenta de consulta médica em si, mas atua como uma etapa de preparação e apoio, garantindo que o paciente chegue ao atendimento com segurança, confiança e autonomia.

2. Descrição das Funcionalidades

A lógica de negócio do nosso sistema é estruturada para garantir que o processo de agendamento e realização de teleconsultas seja simples, guiado e acessível, principalmente para pacientes com dificuldade em usar tecnologias digitais.

O sistema foi desenvolvido utilizando a linguagem Java, contemplando as seguintes funcionalidades principais:

1. **Gerenciamento de Pacientes**
 - Cadastro, consulta, atualização e exclusão de informações de pacientes.
 - Armazenamento de dados como nome, idade, telefone, e-mail e sexo.
2. **Gerenciamento de Acompanhantes**
 - Registro e associação de acompanhantes aos pacientes.
 - Controle de informações como nome, CPF, telefone e parentesco.
3. **Agendamento de Consultas**
 - Criação, atualização e exclusão de agendamentos.
 - Registro de prontuário, horário da consulta, status, especialidade e vínculo com paciente e atendimento.
4. **Gerenciamento de Especialidades Médicas**

- Cadastro e consulta das especialidades disponíveis no hospital.
- Controle de segmento, turno e grau de prioridade.
- 5. **Atendimento**
 - Registro de atendimentos realizados.
 - Associação com paciente, agendamento, médico e especialidade.
- 6. **Feedback**
 - Cadastro e consulta de feedback dos pacientes.
 - Registro de comentários, avaliações e referência ao atendimento realizado.
- 7. **Gerenciamento de Médicos**
 - Cadastro e manutenção de informações dos médicos.
 - Controle de CRM, nome completo, turno e e-mail.

3. Tabela de Endpoints (API RESTful)

1 -- O endpoint da entidade **Paciente** está disponível no caminho /paciente. Através dele, o sistema permite a criação, consulta, atualização e exclusão de registros de pacientes.

O método **GET** acessa os dados cadastrados e pode ser usado de duas formas: /paciente retorna todos os pacientes existentes, enquanto /paciente/{id} busca um paciente específico com base no seu identificador.

O método **POST** é utilizado para cadastrar um novo paciente, enviando um JSON com nome, idade, telefone, e-mail, sexo e senha.

O método **PUT** é responsável por atualizar um registro já existente, modificando as informações do paciente informado pela URI.

Por fim, o método **DELETE** remove permanentemente um paciente do banco de dados.

Os códigos de resposta esperados incluem **200** para sucesso, **201** para criação, **204** para exclusão bem-sucedida, **400** para erros de validação e **404** quando o registro não é encontrado.

2 -- O recurso referente à entidade **Médico** é acessado através do caminho /medico.

Ele permite o gerenciamento completo dos profissionais cadastrados no sistema, incluindo dados como número do CRM, especialidade, turno, nome completo e e-mail.

A requisição **GET** retorna todos os médicos existentes ou, quando acompanhada de um CRM específico (/medico/{nrCrm}), retorna apenas aquele profissional.

O método **POST** é utilizado para cadastrar um novo médico, enquanto o **PUT** serve para atualizar informações de um profissional já cadastrado, como o turno ou a especialidade.

O método **DELETE** remove o médico do banco de dados.
As respostas seguem o padrão HTTP com os códigos **200**, **201**, **204**, **400** e **404**, de acordo com o resultado da operação.

3 -- A entidade **AgendamentoConsulta** tem seu endpoint principal em /agendamento.

Nele, o método **GET** permite visualizar todos os agendamentos cadastrados no sistema, e a chamada /agendamento/{cdAgendamento} recupera um agendamento específico.

O método **POST** cria um novo registro, exigindo que o JSON inclua dados como o prontuário, a data e hora da consulta, o status inicial (que deve ser "Agendado"), e os identificadores do atendimento, paciente, médico e especialidade.

O método **PUT** atualiza o registro de um agendamento, podendo alterar informações como status ou data.

O **DELETE** é utilizado para remover um agendamento específico.
Os retornos padrão incluem **201** para criação bem-sucedida, **200** para atualização, **204** para exclusão, **400** para erro de validação e **404** para inexistência de registro.

4 -- O endpoint da entidade **Atendimento** é definido em /atendimento.

Essa rota é responsável por registrar e consultar os atendimentos realizados entre pacientes e médicos, vinculando-os aos agendamentos existentes.

A chamada **GET** lista todos os atendimentos cadastrados ou, ao utilizar /atendimento/{cdAtendimento}, exibe os dados de um atendimento específico.

O método **POST** realiza o cadastro de um novo atendimento, exigindo informações como nome do paciente, grau de prioridade, data do atendimento, código de agendamento, CRM do médico e especialidade.

O método **PUT** permite atualizar dados já registrados, como o status ou a prioridade do atendimento.

Já o **DELETE** remove permanentemente um atendimento da base.
Os códigos de retorno incluem **200** e **201** para sucesso, **204** para remoção, **400** para erro de validação e **404** quando o atendimento não é encontrado.

5 -- O recurso **Acompanhante** é acessado pelo caminho /acompanhante e gerencia os acompanhantes vinculados aos pacientes.

O método **GET** retorna a lista completa de acompanhantes cadastrados, enquanto **GET /acompanhante/{cdAcompanhante}** busca um registro específico.

A operação **POST** é usada para cadastrar um novo acompanhante, informando nome, CPF, telefone e grau de parentesco.

O método **PUT** permite atualizar as informações de um acompanhante já cadastrado, e o **DELETE** remove um registro do banco de dados.

O sistema retorna **201** quando um novo acompanhante é criado, **200** em atualizações, **204** para remoções, **400** em caso de erro e **404** quando o registro solicitado não é encontrado.

6 -- O endpoint de **Acompanhamento** está em /acompanhamento e é responsável por controlar o vínculo ativo entre o paciente e seu acompanhante.

Com **GET**, é possível listar todos os acompanhamentos registrados ou consultar um acompanhamento específico através de /acompanhamento/{idPaciente} e /acompanhamento/{idPaciente}/{cdAcompanhante}.

O método **POST** cria um novo acompanhamento, garantindo que o paciente e o acompanhante sejam corretamente associados.

O **PUT** atualiza o status do acompanhamento (por exemplo, de “Ativo” para “Encerrado”) ou altera o vínculo.

O **DELETE** encerra o acompanhamento de um paciente, removendo o vínculo entre ambos.

As respostas esperadas seguem o padrão **200**, **201**, **204**, **400** e **404**, de acordo com a operação executada.

7 -- O endpoint /feedback trata das avaliações registradas pelos pacientes após o atendimento.

O método **GET** retorna todos os feedbacks existentes ou, quando acompanhado de um identificador, exibe um feedback específico.

A operação **POST** permite registrar uma nova avaliação, contendo prontuário, data de envio, comentário, referência, avaliação e informações de vínculo com o atendimento, paciente e médico.

O método **PUT** atualiza comentários ou avaliações já existentes, enquanto o **DELETE** remove um feedback do sistema.

Os retornos seguem o padrão HTTP, com **201** para criação, **200** para

atualização, **204** para exclusão, **400** para falhas e **404** quando o registro não é localizado.

8 -- O endpoint **/login** centraliza o processo de autenticação e é acessado tanto via navegador quanto via cliente REST.

O método **GET** exibe uma página HTML explicativa sobre o uso do endpoint, servindo como documentação interna da API.

O método **POST**, por sua vez, realiza a autenticação do usuário com base no e-mail e senha enviados no corpo da requisição JSON.

Em caso de sucesso, o sistema retorna o usuário autenticado com código **200**; caso contrário, responde com **401 (Unauthorized)** ou **400 (Bad Request)** se houver erro de validação.

Esse endpoint garante o controle de acesso e validação das credenciais cadastradas no banco de dados.

9 -- O endpoint **/remedio** foi utilizado como exemplo didático para implementação de CRUD básico e segue o mesmo padrão de estrutura das demais entidades.

O método **GET** lista todos os remédios cadastrados ou retorna um registro específico através de **/remedio/{codigo}**.

O **POST** cria um novo remédio, enviando dados como nome, preço, data de fabricação e validade.

O **PUT** permite atualizar essas informações, e o **DELETE** exclui o registro do banco.

Os status retornados seguem o padrão REST: **200** e **201** para sucesso, **204** para exclusão, **400** para erro de dados e **404** quando o item solicitado não é encontrado.

4. Protótipo do Sistema – Print das telas

Fechar

Pacientes

Médicos

Consultas

Atendimentos

Acompanhantes

Acompanhamentos

Remédios

Especialidades

Feedback

Sair

Acompanhamentos

Ocultar Lista

Paciente

Acompanhante

Status

Selecione o paciente...

Selecione o acompanhante...

Selecione o status...

Salvar Acompanhamento

Paciente	Acompanhante	Status	Ações
João Pereira	Maria da Silva	Aguardando documentos	<div>Editar</div> <div>Excluir</div>
Bruno Ferreira	Maria da Silva	Em análise médica	<div>Editar</div> <div>Excluir</div>

Acompanhantes

Ocultar Lista

Nome do Acompanhante

CPF

Digite o nome completo

Somente números

Telefone

Parentesco

(11) 99999-9999

Ex: Pai, Mãe, Irmão...

Salvar

ID	Nome	CPF	Telefone	Parentesco	Ações
25	Maria da Silva	12345678900	11998765432	Filha	<div>Editar</div> <div>Excluir</div>

Agendamento de Consultas

Ocultar Lista

Prontuário

Data e Hora

Paciente

Selecione...

dd/mm/aaaa --:--

Selecione...

Status

Médico (CRM)

Especialidade

Agendado

Selecione o CRM...

Selecione...

Salvar Consulta

ID	Prontuário	Data/Hora	Paciente	Status	CRM	Especialidade	Ações
23	Consulta de rotina - cardiologia	2025-12-18 14:30	1	Agendado	1	Clinico Geral	<div>Editar</div> <div>Excluir</div>
25	Retorno	2025-11-07 20:20	Bruno Ferreira	Realizado	23	Clinico Geral	<div>Editar</div> <div>Excluir</div>
27	Primeira consulta	2025-11-07 21:07	23	Agendado	26	Clinico Geral	<div>Editar</div> <div>Excluir</div>

Fechar

Pacientes

Médicos

Consultas

Atendimentos

Acompanhantes

Acompanhamentos

Remédios

Especialidades

Feedback

Sair

Atendimentos

Ocultar Lista

Paciente

Grau de Prioridade

Data do Atendimento

Selecione o paciente...

Selecione...

dd/mm/aaaa

Código do Agendamento

Médico (CRM)

Especialidade

Selecione...

Selecione o médico...

Selecione...

Salvar Atendimento

Código	Paciente	Prioridade	Data	Agendamento	CRM	Especialidade	Ações
76	João Pereira	Alta	27/12/2025	23	2	1	<div>Editar</div> <div>Excluir</div>
79	João Pereira	Alta	27/12/2025	23	2	1	<div>Editar</div> <div>Excluir</div>

Fechar

Pacientes

Médicos

Consultas

Atendimentos

Acompanhantes

Acompanhamentos

Remédios

Especialidades

Feedback

Sair

Feedbacks

Ocultar Lista

Paciente

Avaliação

Referência

Selecione o paciente...

Selecione...

Selecione...

Comentário

Salvar Feedback

Código	Paciente	Avaliação	Referência	Comentário	Ações
39	João Pereira	Ótimo	Excelente atendimento	O médico foi muito atenciosos e explicou claramente os resultados.	<div>Editar</div> <div>Excluir</div>
44	João Pereira	Ótimo	Excelente atendimento	O médico foi muito atenciosos e explicou claramente os resultados.	<div>Editar</div> <div>Excluir</div>
47	João Pereira	Ótimo	Excelente atendimento	O médico foi muito atenciosos e explicou claramente os resultados.	<div>Editar</div> <div>Excluir</div>
52	Bruno Ferreira	Ótimo	Consulta de rotina	okkk	<div>Editar</div> <div>Excluir</div>

Especialidades

Ocultar Lista

Segmento

Turno

Grau de Prioridade

Digite o nome da especialidade

Selecione o turno...

Selecione...

Salvar Especialidade

Código	Segmento	Turno	Prioridade	Ações
1	Clinico Geral	Manhã	Média	<div>Editar</div> <div>Excluir</div>
2	Cardiologia	Tarde	Alta	<div>Editar</div> <div>Excluir</div>
3	Ortopedia	Noite	Baixa	<div>Editar</div> <div>Excluir</div>
4	Dermatologia	Manhã	Média	<div>Editar</div> <div>Excluir</div>



Acessar o Sistema

E-mail

Digite seu e-mail

Senha

Digite sua senha

Entrar

Não tem conta? Cadastre-se

Fechar

Pacientes

Médicos

Consultas

Atendimentos

Acompanhantes

Acompanhamentos

Remédios

Especialidades

Feedback

<+ Sair

Fechar

Pacientes

Médicos

Consultas

Atendimentos

Acompanhantes

Acompanhamentos

Remédios

Especialidades

Feedback

<+ Sair

Fechar

Pacientes

Médicos

Consultas

Atendimentos

Acompanhantes

Acompanhamentos

Remédios

Especialidades

Feedback

<+ Sair

Pacientes

Ocultar Lista

Nome

Telefone

E-mail

Digite o nome completo

(11) 99999-9999

exemplo@email.com

Sexo

Idade

Selecione...

Ex: 35

Salvar

ID	Nome	E-mail	Telefone	Idade	Sexo	Ações
19	João Pereiraa	joao.pereira@gmail.com	11999887766	65	Masculino	<div>Editar</div> <div>Excluir</div>
21	Bruno Ferreiraa	bruno@gmail.com	1194049-1088	25	Masculino	<div>Editar</div> <div>Excluir</div>
25	Gabriel	joao.pereira@gmail.com	11999887777	13	Masculino	<div>Editar</div> <div>Excluir</div>

Médicos

Ocultar Lista

Nome Completo

CRM

Código da Especialidade

Digite o nome completo

Ex: 123456

Selecione...

Segmento

Turno

E-mail

Selecioneado automaticamente

Selecione...

exemplo@email.com

Salvar

CRM	Nome	E-mail	Turno	Segmento	Especialidade	Ações
2	Dra. Ana Costa	ana.costa@hospital.com	Manhã	Cardiologia	1	<div>Editar</div> <div>Excluir</div>
23	Dra. Ana Costa	ana.costa@hospital.com	Manhã	Cardiologia	1	<div>Editar</div> <div>Excluir</div>
24	Dr zezinho	ze@gmail.com	Tarde	Clinico Geral	1	<div>Editar</div> <div>Excluir</div>

Remédios

Ocultar Lista

Código

Nome do Remédio

Ex: 1

Ex: Paracetamol 750mg

Preço (R\$)

Data de Fabricação

Data de Validade

Ex: 12.50

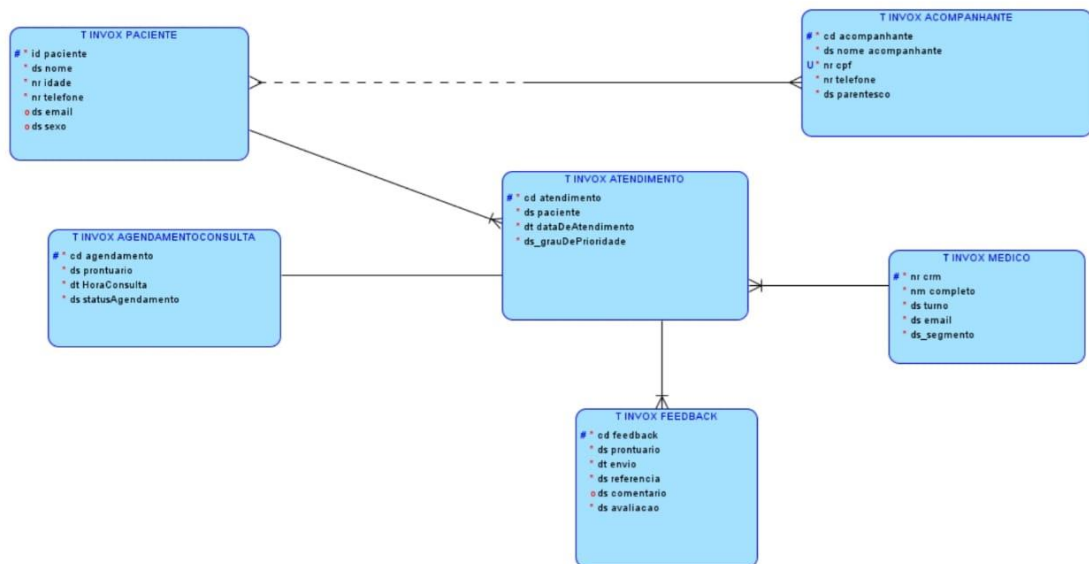
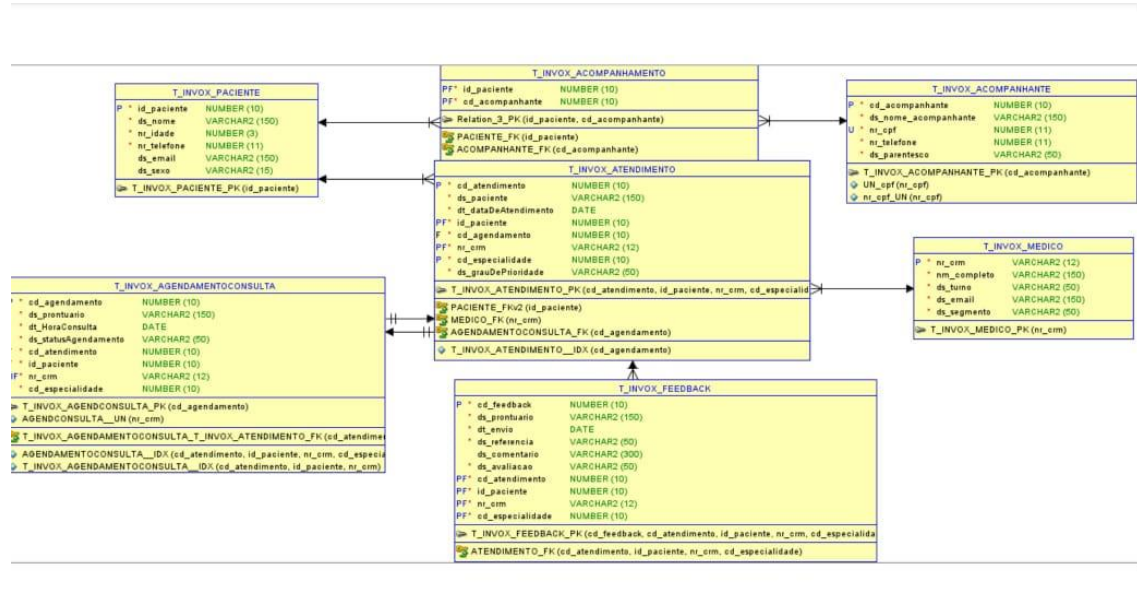
dd/mm/aaaa

dd/mm/aaaa

Salvar

Código	Nome	Preço (R\$)	Fabricação	Validade	Ações
68	Amoxicilina	18.75	2025-11-06	2026-01-10	<div>Editar</div> <div>Excluir</div>

5. Modelo de Entidade Relacionamento (MER)



6. Procedimentos para Rodar a Aplicação

Ferramentas necessárias:

- IntelliJ IDEA
- Java JDK 17

Passo a passo:

1. Instalar o IntelliJ IDEA e Java JDK 17

2. Importar o projeto Java no IntelliJ.
3. Configurar a conexão com o banco no arquivo ConnectionFactory.java.
4. Rodar a aplicação no IntelliJ.