



9 DE NOVEMBRO DE 2025

# REABILI – JAVA DOCUMENTATION

CCG

1TDSPW

GUILHERME RIBEIRO DA COSTA, CARLOS ANDRÉ SILVA, CAMILO MICHELETTO



## Sumário

Sumário.....	1
Objetivo.....	2
Escopo.....	2
Protótipo do Sistema (Figma).....	3
Descrição das funcionalidades.....	3
Diagrama MER.....	4
Diagramas de classes.....	5
Ferramentas utilizadas.....	5
Versões necessárias.....	6
Como executar.....	6
Importando o projeto.....	6
SQL.....	6
Tabela de endpoints.....	7

## Objetivo

Disponibilizar uma plataforma para centralizar o suporte para serviços do HC envolvendo teleconsultas.

Na plataforma devemos ter os seguintes recursos para usuários:

1. Pesquisa de dúvidas por texto e/ou voz.
  - a. Pode ter dúvidas exibidas em formato de texto
  - b. Pode ter dúvidas exibidas em formato de vídeo.
2. Interação com chatbot por texto e/ou voz.

Para usuários administradores temos os seguintes recursos:

Para essa seção entendemos conteúdo como textos, vídeos, imagens ou qualquer formato de mídia que é disponibilizado para as dúvidas de usuários, com exceção das respostas fornecidas via chatbot.

1. Pesquisa de dúvidas por texto e/ou voz.
  - a. Pode ter dúvidas exibidas em formato de texto
  - b. Pode ter dúvidas exibidas em formato de vídeo.
2. Interação com chatbot por texto e/ou voz.
3. Deleção de conteúdo na plataforma.
4. Alteração de conteúdo na plataforma.
5. Adição de conteúdo na plataforma.
6. Login para usuários admin

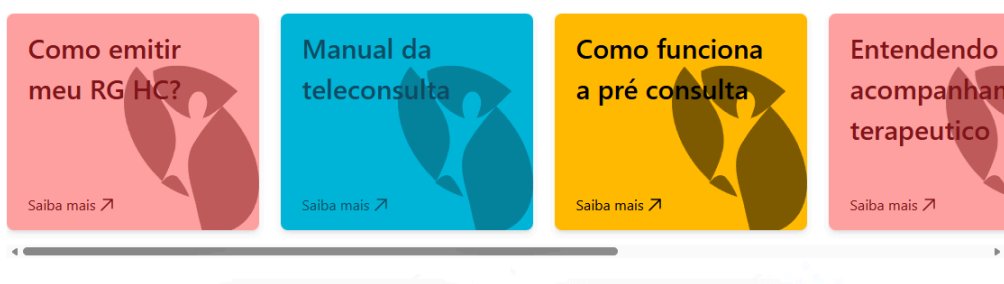
## Escopo

Este projeto visa apenas abranger fluxos relacionados a teleconsultas, não devendo ser responsável por fluxos externos, ao menos no início.

Através dos chatbots, e conteúdos alimentados organicamente por administradores dos sistemas, os usuários devem se informar, se o conteúdo no site não for suficiente, o usuário tem a possibilidade de falar com um atendente pelos canais de whatsapp/email.

## Protótipo do sistema (Prints)

### Dúvidas frequentes



Artigos/ Dúvidas servidos pelo backend através do endpoint “GET /articles”.



Botões “Editar Conteúdo”/”Deletar conteúdo” acessam os endpoints PUT /article/{id} e DELETE/article/{id}

## Descrição das funcionalidades

- Pesquisa por **texto**

O usuário pode realizar pesquisa por texto na caixa de pesquisa que está presente no app.

- Pesquisa por **voz**

O usuário pode clicar no botão de microfone para iniciar a pesquisa por voz, o que for dito deve ser traduzido em texto.

- Interação com chatbot por texto

O usuário deve digitar sua mensagem na caixa de diálogo do chat para que o bot responda.

- Interação com chatbot por voz

O usuário deve clicar no microfone presente na caixa de diálogo do chat para que o bot responda.

- Deleção de conteúdo na plataforma.

O usuário administrador pode deletar artigos na plataforma.

- Alteração de conteúdo na plataforma.

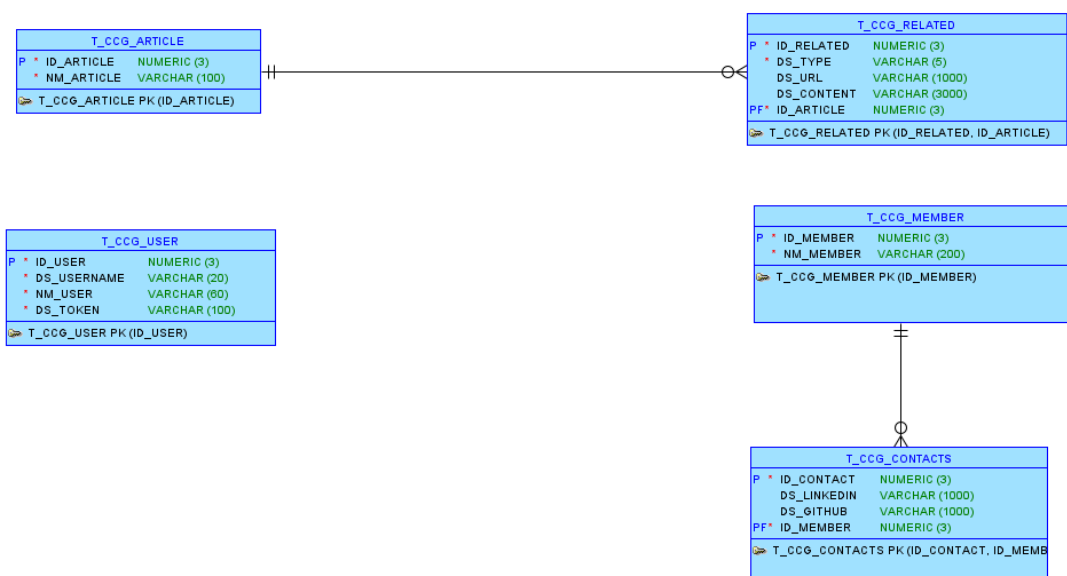
O usuário administrador pode gerenciar artigos na plataforma.

- Adição de conteúdo na plataforma.

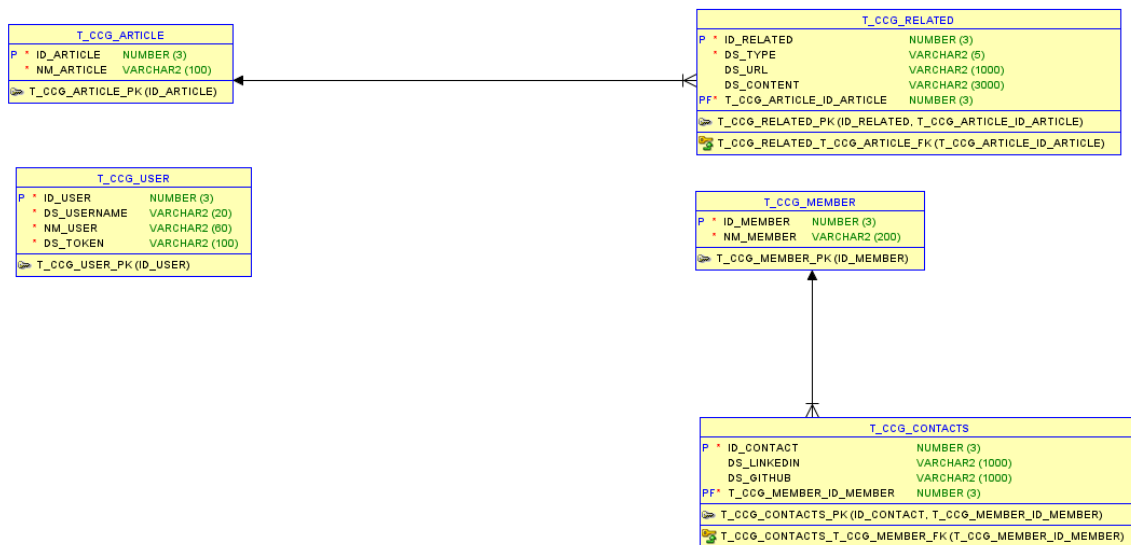
O usuário administrador pode adicionar artigos na plataforma.

## Diagrama MER

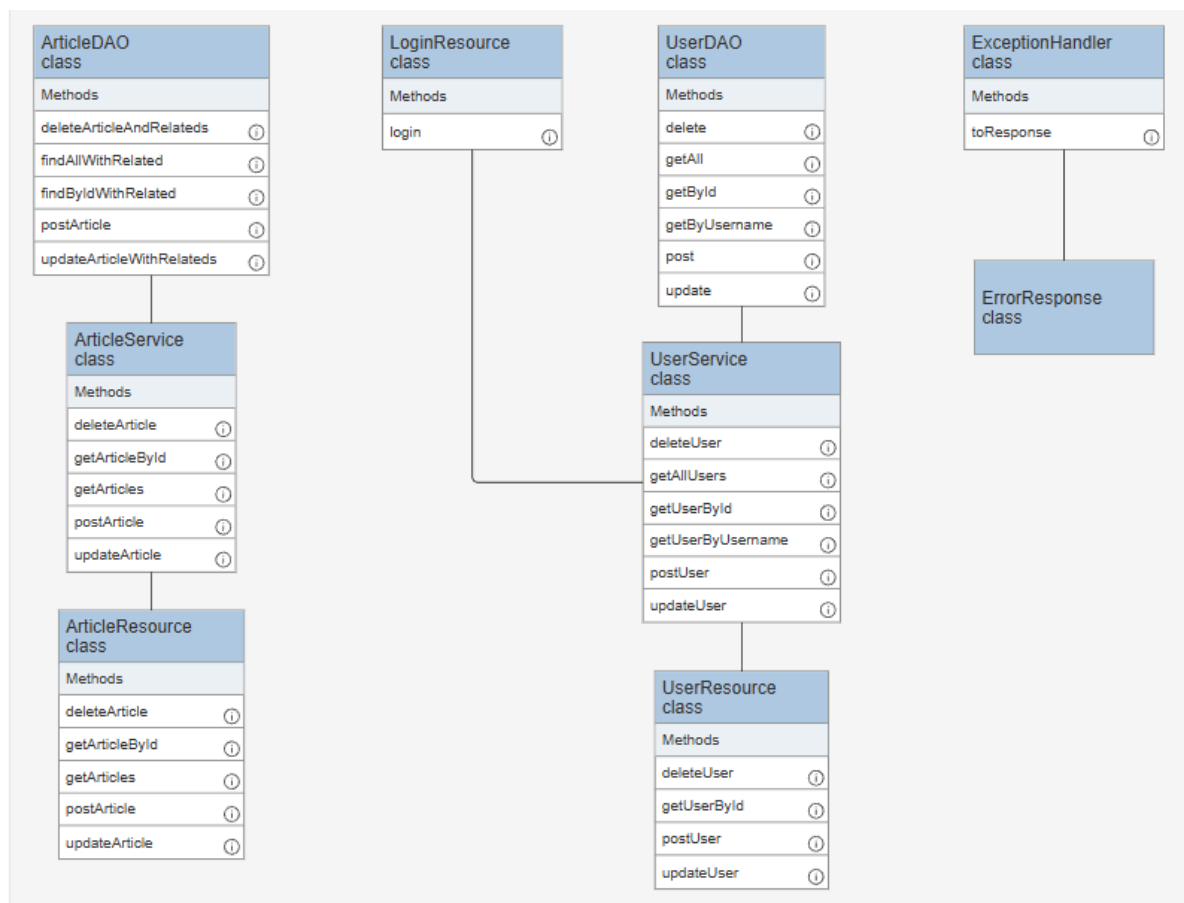
### Modelo lógico



## Modelo Relacional



## Diagramas de classes



## Ferramentas utilizadas

IntelliJ IDEA - Editor de código

Maven - Gerenciador de dependências

BrModelo - Modelagem Conceitual

Oracle Data Modeler - Modelagem lógica/física

Oracle SQL Developer - Edição de código SQL

Trello - Organização de tarefas

Figma - Diagramação das telas

Postman - Requisições HTTP

Draw.io - Diagramas de classe

## Versões necessárias

JDK - 21

Maven - 3.9.9 (Deve funcionar com 3.6.3 ou superior, mas dê preferência para 3.9.9)

IntelliJ IDEA - 2025.2

## Como executar

1. Abrir IntelliJ IDEA
  - a. No projeto, execute o comando **mvn clean install**
  - b. Acessar a aba de configurações de execução -> editar configurações -> clicar no botão “+” para adicionar nova execução -> maven
  - c. Já no painel de nova execução, em **Run**, adicione o seguinte comando “**quarkus:dev**”
  - d. Em **Java Options** desmarque a opção “Inherit from settings” e na opção de JRE anexe uma versão compatível com **21.X.X**
  - e. Em seguida, execute sua nova configuração de execução.
2. Abrir Postman
  - a. Selecionar a opção **import** no seu workspace
  - b. enviar arquivo JSON disponível no zip.
  - c. realizar requisições.
  - d. na variavel da collection `{{baseUrl}}` adicione **localhost:8080**

## Importando o projeto

1. ZIP
  - a. Descompacte o arquivo recebido
  - b. Selecione a pasta referente ao projeto
  - c. Abra a pasta como um projeto na sua IDE de preferência
2. GIT
  - a. Em sua IDE de preferência ou no terminal realize o seguinte comando (certifique-se de estar na organização)  
  
`git clone https://github.com/Challenge-2025-Equipe-07/hc-reabilli-backend.git`

Caso tenha problemas na branch **main** utilize **feat\_quarkus\_implementation**.

## SQL

O arquivo **.sql** estará disponível no .zip caso você queira utilizar em outro schema.

## Tabela de endpoints

- Article
  - GET /article
    - 200, 404, 500
  - POST /article
    - 201, 500
  - GET /article/{id}
    - 200, 404, 500
  - PUT /article/{id}
    - 200, 404, 500
  - DELETE /article/{id}
    - 204, 404, 500
- Login
  - POST /login
    - 200, 404, 500
- User
  - POST /user
    - 201, 404, 500
  - GET /user/{id}
    - 200, 404, 500
  - PUT /user/{id}
    - 200, 404, 500
  - DELETE /user/{id}
    - 204, 404, 500