

Nome: Luna de Carvalho Guimarães, RM: 562290 , Turma: 1TDSPH

Nome: Felipe Monte de Sousa, RM: 562019 , Turma: 1TDSPH

Nome: Aline Lourenço Carvalho, RM: 564538, Turma: 1TDSPH

Nome da Equipe: Lumi

Nome da solução: Sistema para Suporte de Saúde Digital

# Sistema para Suporte de Saúde Digital

# Objetivo e Escopo do Projeto

O principal objetivo deste trabalho será em desenvolver uma API RESTful em Java, que seja capaz de permitir a comunicação entre o nosso sistema back-end com outros sistemas de diferentes plataformas que no caso neste trabalho em específico, nosso API será utilizada para realizar a integração com o sistema do Java com a disciplina de front-end, onde estará utilizando as tecnologias de React, Vite e TypeScript para fazer a parte visível no qual os nossos usuários estarão interagindo com o sistema.

Seu principal propósito será de ajudar os usuários do aplicativo do HC ( Hospital das Clínicas) que possuem dúvidas sobre a plataforma de saúde, criando assim um ambiente de colaboração mútua entre os usuários, especialmente para atender um determinado grupo de pessoas que possuem dificuldades em utilizar o aplicativo principal do HC, como por exemplo idosos, pessoas com deficiência ou em alta vulnerabilidade social.

## Escopo:

O sistema desenvolvido implementa as seguintes funcionalidades:

- **Gestão dos usuários:** Cadastro, login, atualização dos dados da conta e sua exclusão;
- **Perguntas e respostas:** Sua criação, edição, exclusão e sua listagem, incluindo a associação entre perguntas e suas respectivas respostas;
- **Vídeos:** Criação de vídeos, edição, exclusão e listagem de todos os vídeos da plataforma;
- **Feedbacks:** Envio, edição, listagem e sua exclusão dos feedbacks vinculados ao próprio usuário;

## O que não faz parte do Escopo do Projeto:

Como esse projeto foca exclusivamente em criar um sistema para um suporte de aplicativo, as partes das funcionalidades médicas, administrativas ou que possuem alguma integração direta com o sistema principal do aplicativo do HC, ficarão de fora do escopo desse trabalho, como por exemplo, os seguintes itens:

- Agendar consultas reais diretamente nessa aplicação;
- Acessar aos Prontuários Médicos do paciente;
- Realizar vídeos chamadas de consultas;
- Ver receituário médico digital real;
- Lembretes de medicação para tomar e de consultas que estão agendadas;

# Breve descrição das próprias funcionalidades da solução

As funcionalidades da solução estão divididas em diferentes pacotes como a camada TO que representa as entidades de domínio, DAO que é responsável pela persistência de dados (CRUD) com o banco de dados e BO que fará a implementação das nossas regras de negócio, além da Resource que é onde ficarão nossos endpoints HTTP.

Abaixo contém a descrição das funcionalidades:

**Gestão de Usuários:** Permite o cadastro do usuário, seu login, atualização dos dados de sua própria conta e sua exclusão, controlando assim as informações básicas do acesso ao nosso sistema, garantindo assim que cada pessoa tenha uma conta individual para interagir com a plataforma, possuindo executar as seguintes operações:

- **Cadastro de usuário:** o usuário deve informar o seu nome completo, seu username, e-mail, senha e sua idade;
- **Login:** o usuário pode acessar sua conta informando ou o seu e-mail ou o seu username, junto com a sua senha, assim o sistema verifica se as credenciais estão certas e se estiverem, ele permite o acesso as outras funcionalidades do programa;
- **Atualização de dados:** permite que o usuário possa alterar suas informações da conta, como o nome, e-mail, senha ou sua idade;
- **Exclusão de conta:** o usuário pode solicitar a exclusão definitiva de sua conta;

Durante o cadastro, o sistema irá realizar validações para evitar que os usuários façam contas com os mesmos e-mails e usernames que já foram registrados lá no banco, garantindo que esses atributos específicos sejam únicos para cada usuário.

**Perguntas e Respostas:** Tem como objetivo permitir que os usuários publiquem, consultem, atualizem e excluam perguntas e respostas, essa funcionalidade possui o seguinte objetivo para tirarem dúvidas com os outros usuários sobre o uso do aplicativo da HC, criando um ambiente colaborativo para todos, onde possam

compartilhar dúvidas e ajudar uns aos outros, promovendo a troca de conhecimento dentro da comunidade, executando assim as seguintes operações:

- **Criação de pergunta:** o usuário informa o título e a descrição da dúvida que deseja publicar.
- **Listagem de perguntas:** o sistema mostra todas as perguntas que foram cadastradas na plataforma, podendo assim os usuários visualizá-las.
- **Edição de pergunta:** permite que somente o autor da própria pergunta possa atualizá-la caso queira mudar o título ou a descrição de uma pergunta que já foi publicada.
- **Exclusão de pergunta:** possibilita a remoção definitiva de uma pergunta feita somente pelo próprio usuário.

Além das perguntas, o nosso sistema também irá implementar a funcionalidade de responder essas perguntas, proporcionando a interação com outros usuários:

- **Criação de resposta:** o usuário pode responder a uma pergunta existente informando o conteúdo de sua resposta.
- **Listagem de respostas:** o sistema exibe todas as respostas vinculadas a uma pergunta específica, podendo assim os usuários visualizar todas as respostas daquela pergunta.
- **Edição de resposta:** o usuário pode editar o conteúdo de sua resposta que ele mesmo publicou em uma pergunta específica.
- **Exclusão de resposta:** permite ao usuário excluir uma resposta de sua autoria de uma pergunta que foi publicada.

**Vídeos:** A funcionalidade de vídeos é responsável por permitir que os usuários da plataforma possam criar, visualizar, editar e excluir vídeos educativos relacionados em dar um suporte ao uso do aplicativo do Hospital das Clínicas (HC).

Diante dessa funcionalidade, o sistema permite executar as seguintes operações:

- **Cadastro dos vídeos:** o usuário informa o título do vídeo, sua descrição, seu link, duração e sua categoria.
- **Listagem dos vídeos:** Mostra aos usuários todos os vídeos cadastrados na plataforma.
- **Edição de vídeo:** o usuário que criou o vídeo pode alterar suas informações, como o título, descrição, duração, categoria ou link, somente o próprio usuário que criou o vídeo pode editá-lo.
- **Exclusão de vídeo:** permite somente aos próprios autores excluírem seus próprios vídeos.

**Gestão de Feedbacks:** Essa funcionalidade é responsável por permitir que todos os usuários possam enviar suas avaliações a respeito do uso da plataforma, tendo como objetivo principal coletar todas percepções dos usuários que usam o nosso programa para melhorar a experiência do usuário e da usabilidade do sistema de suporte do aplicativo da HC.

Dentro dessa funcionalidade a Gestão de Feedbacks possui as seguintes operações:

- **Envio de feedback:** Para enviar um feedback o usuário deve informar um comentário sobre a plataforma e em seguida uma avaliação numérica, representando sua satisfação com a plataforma.
- **Listagem dos feedbacks:** Mostra todos os feedbacks cadastrados pelo próprio usuário.
- **Edição do feedback:** O usuário pode editar um feedback que ele mesmo enviou, atualizando o conteúdo do seu comentário ou sua nota de avaliação.
- **Exclusão de feedback:** O usuário tem a opção de excluir permanentemente um feedback que ele próprio enviou.

# Tabela de Endpoints (API RESTful)

UsuarioResource:

URI	Verbo HTTP	Descrição	Código de Status
/java/usuario/login	POST	Autentica um usuário no sistema.	200 (OK) e 401 (Não Autorizado)
/java/usuario	POST	Cadastra um novo usuário.	201 (Created) e 400(BadRequest)
/java/usuario	GET	Lista todos os usuários.	200 (OK) e 404 (Not Found)
/java/usuario/{id}	GET	Busca um usuário pelo ID.	200 (OK) e 404 (Not Found)
/java/usuario/{idUsuarioLogado}	DELETE	Deleta um usuário.	204 (No Content) e 404 (Not Found)
/java/usuario/{idUsuarioLogado}	PUT	Atualiza os dados do usuário.	200 (OK) e 400 (Bad Request)

VideoResource:

URI	Verbo HTTP	Descrição	Código de Status
/java/video	GET	Lista todos os vídeos.	200 (OK) e 404 (Not Found)
/java/video/{id}	GET	Busca um vídeo pelo ID.	200 (OK) e 404 (Not Found)
/java/video/usuario/{idUsuarioLogado}	POST	Salva um novo vídeo.	201 (Created) e 400 (Not Found)
/java/video/{idVideo}/usuario/{idUsuarioLogado}	DELETE	Deleta um vídeo.	204 (No Content) e 404 (Not Found)
/java/video/{idVideo}/usuario/{idUsuarioLogado}	PUT	Atualiza um vídeo.	200 (OK) e 400 (Bad Request)

PerguntaResource:

URI	Verbo HTTP	Descrição	Código de Status
/java/pergunta	GET	Lista todas as perguntas.	200 (OK) e 404 (Not Found)
/java/pergunta/{id}	GET	Busca uma pergunta pelo ID.	200 (OK) e 404 (Not Found)
/java/pergunta/usuario/{idUsuarioLogado}	POST	Salva uma nova pergunta.	201 (No Content) e 400 (Bad Found)
/java/pergunta/{idPergunta}/usuario/{idUsuarioLogado}	DELETE	Deleta uma pergunta.	204 (No Content) e 404 (Not Found)
/java/pergunta/{idPergunta}/usuario/{idUsuarioLogado}	PUT	Atualiza uma pergunta.	200 (OK) e 400 (Bad Request)

RespostaResource:

URI	Verbo HTTP	Descrição	Código de Status
/java/resposta	GET	Lista todas as respostas.	200 (OK) e 404 (Not Found)
/java/resposta/{id}	GET	Busca uma resposta pelo ID.	200 (OK) e 404 (Not Found)
/java/resposta/pergunta/{idPergunta}	GET	Busca respostas pelo ID da pergunta.	200 (OK) e 404 (Not Found)

/java/resposta/usuario/{idUsuarioLogado}	POST	Salva uma resposta.	201 (Created) e 400 (Bad Request)
/java/resposta/{idResposta}/usuario/{idUsuarioLogado}	DELETE	Deleta uma resposta.	204 (No Content) e 404 (Not Found)
/java/resposta/{idResposta}/usuario/{idUsuarioLogado}	PUT	Atualiza uma resposta.	200 (OK) e 400 (Bad Request)

FeedbackResource:

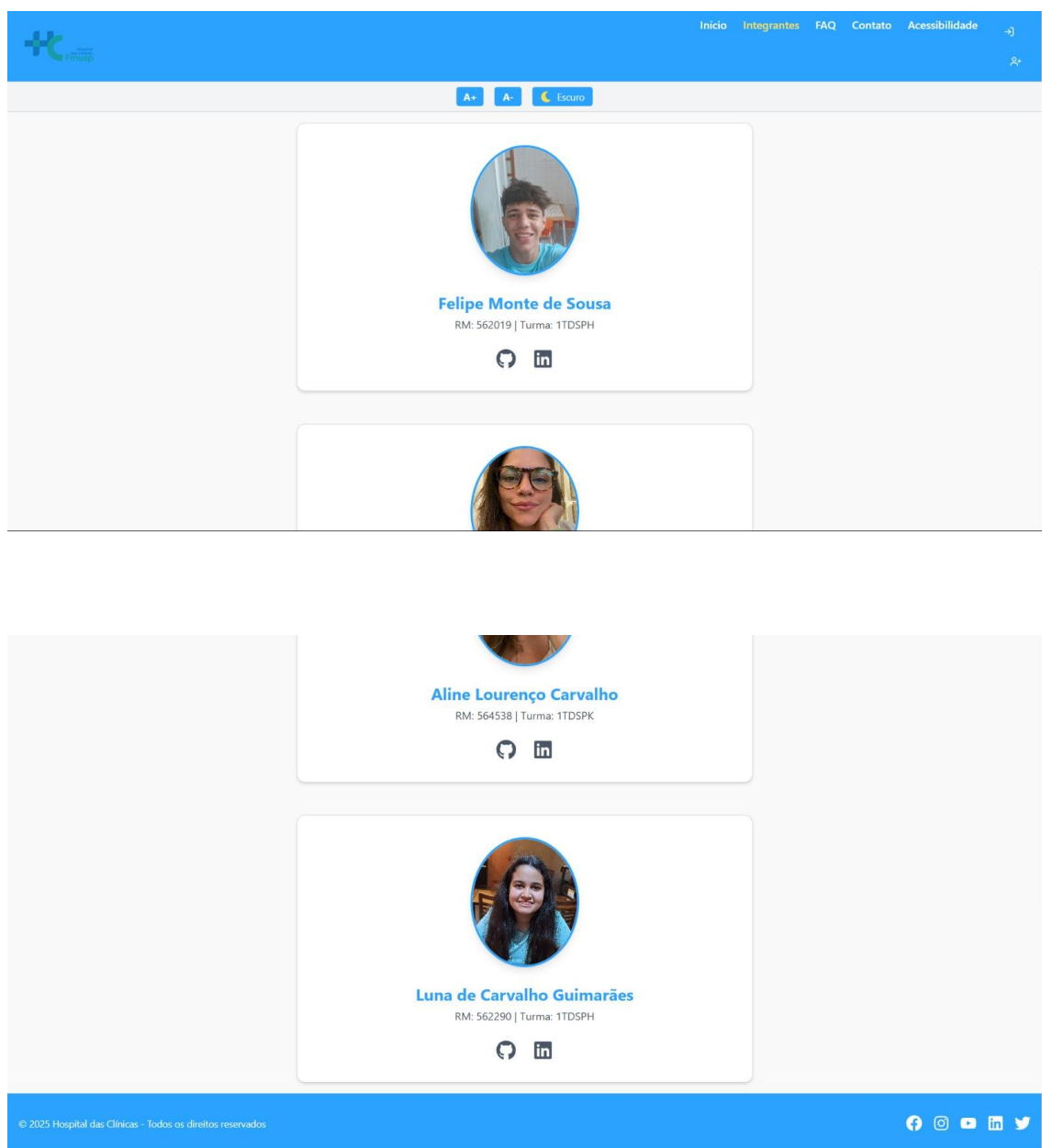
URI	Verbo HTTP	Descrição	Código de Status
/java/feedback	GET	Lista todos os feedbacks.	200 (OK) e 404 (Not Found)
/java/feedback/usuario/{idUsuario}	GET	Busca feedbacks pelo ID do usuário.	200 (OK) e 404 (Not Found)
/java/feedback/usuario/{idUsuarioLogado}	POST	Salva um novo feedback.	201 (Created) e 400 (Bad Request)
/java/feedback/{idFeedback}/usuario/{idUsuarioLogado}	DELETE	Deleta um feedback.	204 (No Content) e 404 (Not Found)
/java/feedback/{idFeedback}/usuario/{idUsuarioLogado}	PUT	Atualiza um feedback.	200 (OK) e 400 (Bad Request)



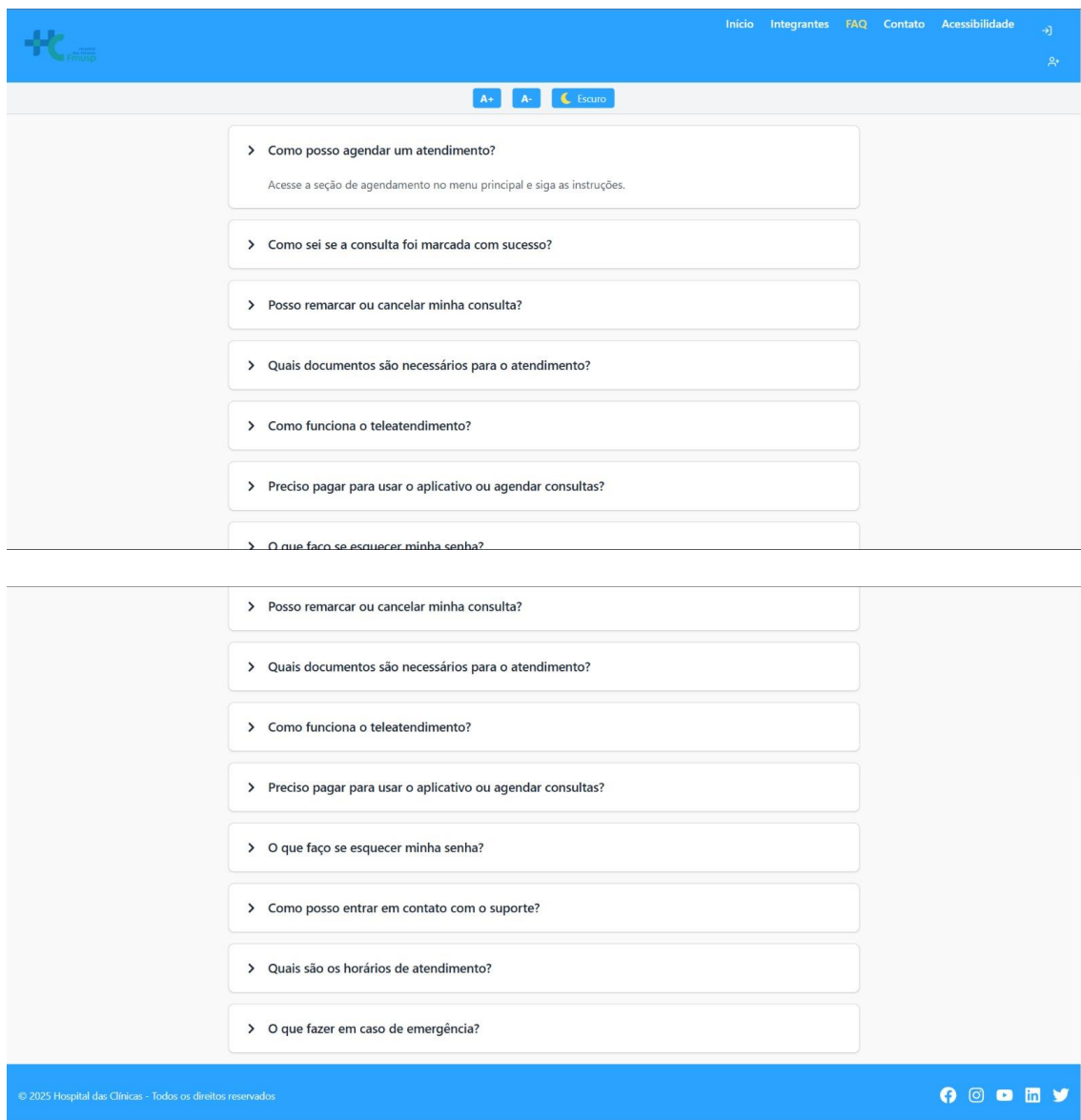
# Protótipo – Prints das telas implementadas



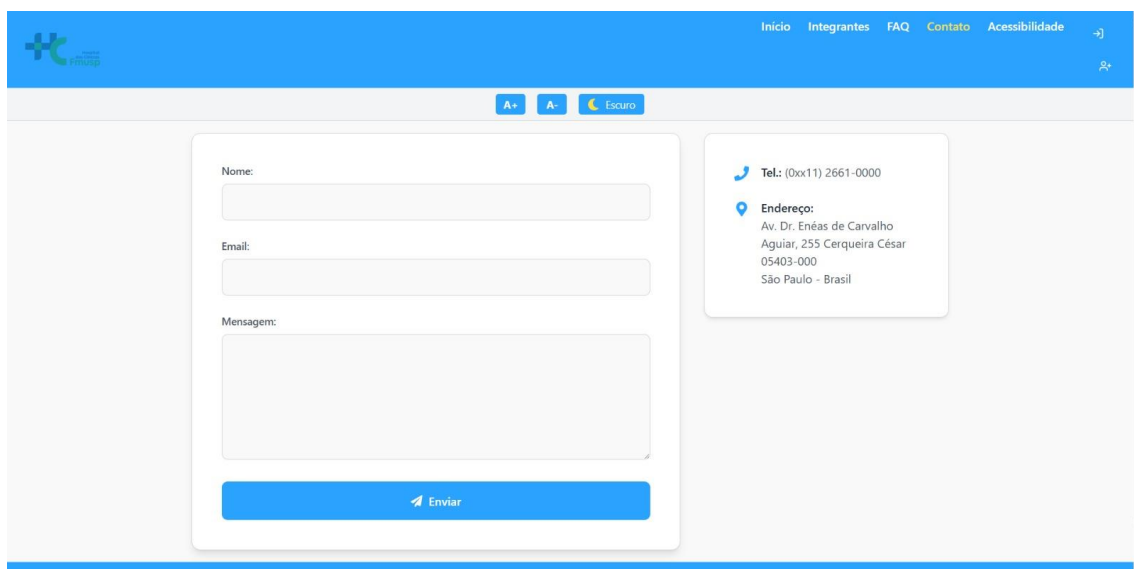
Página inicial do sistema no Frond-End, nela os usuários já podem ver as seguintes partes do programa como a área de Vídeos Tutoriais, área de Duvidas e do Fale Conosco, além de podem ver a parte dos integrantes que fizeram todo o sistema, acessar o FAQ, a área de Contato da parte da Acessibilidade.



Área do site onde contém todas as informações dos integrantes que colaboraram com o trabalho, além de seus perfis do GitHub e do LinkedIn.



Área sobre as perguntas frequentes do programa de suporte da HC.



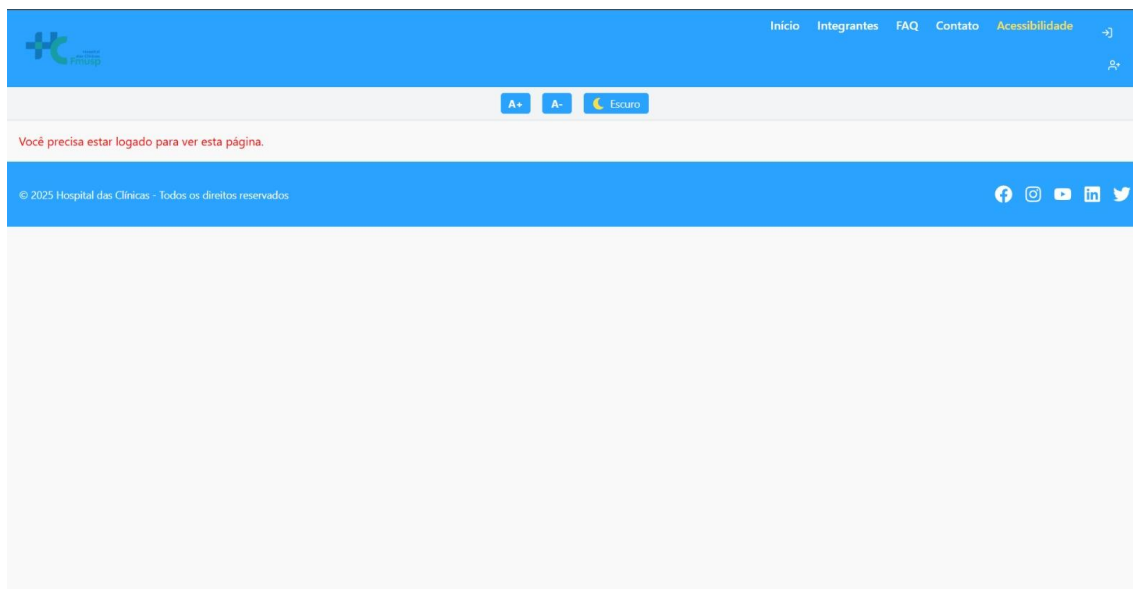
Área para contato ao suporte técnico se houver necessidade ao usuário.

The screenshot shows the 'Cadastro' (Registration) form on the Hospital das Clínicas website. The form is titled 'Cadastro' and includes the instruction 'Crie sua conta' (Create your account). It contains the following fields: 'Nome' (Name), 'Username' (with a placeholder '@usuario'), 'Email', 'Idade' (Age), and 'Senha' (Password). At the bottom of the form, there are two buttons: 'Voltar' (Back) and 'Cadastrar' (Register). The website's header is blue and contains the logo on the left and navigation links (Início, Integrantes, FAQ, Contato, Acessibilidade) on the right. Below the header, there are accessibility controls (A+, A-, and Escuro) and a search icon.

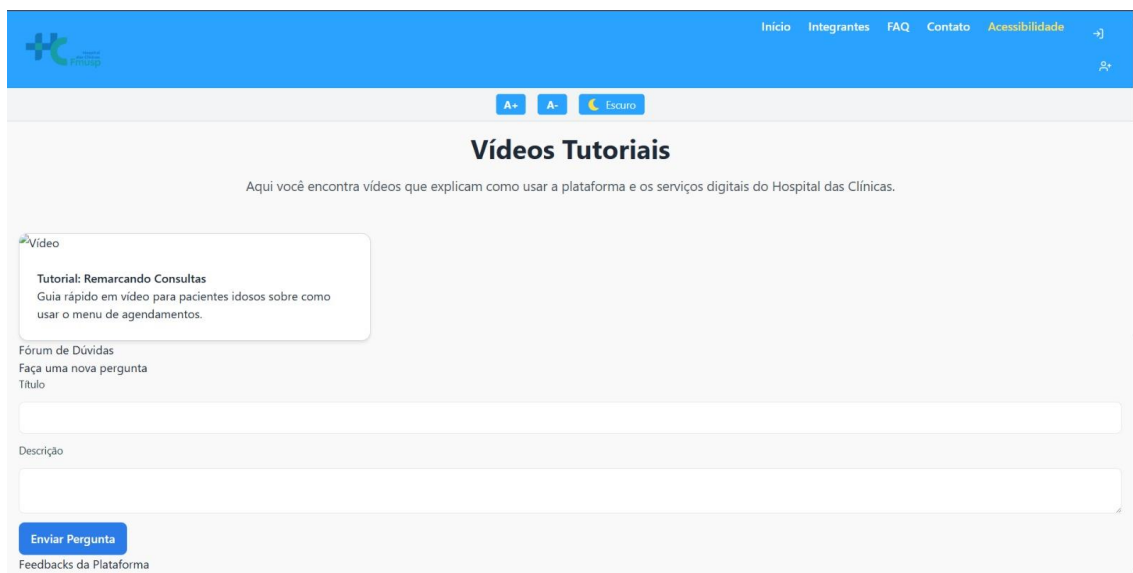
Área do Cadastro para os usuários criarem suas contas para usarem o programa.

The screenshot shows the 'Entrar' (Login) form on the Hospital das Clínicas website. The form is titled 'Entrar' and includes the instruction 'Acesse sua conta' (Access your account). It contains the following fields: 'Email (Login)' and 'Senha' (Password). At the bottom of the form, there are two buttons: 'Cadastrar' (Register) and 'Entrar' (Login). The website's header is blue and contains the logo on the left and navigation links (Início, Integrantes, FAQ, Contato, Acessibilidade) on the right. Below the header, there are accessibility controls (A+, A-, and Escuro) and a search icon. The footer is blue and contains the copyright notice '© 2025 Hospital das Clínicas - Todos os direitos reservados' on the left and social media icons (Facebook, Instagram, YouTube, LinkedIn, Twitter) on the right.

Área de login onde os usuários que já possuem conta podem entrar no sistema.



Caso o usuário preferir entrar em alguma funcionalidade do programa sem estar logado irá aparecer a seguinte mensagem de erro “Você precisa estar logado para ver esta página”.



Descrição

Enviar Pergunta

Feedbacks da Plataforma

Deixe seu feedback

Avaliação

Comentário

Enviar Feedback

O que outros usuários estão dizendo

"O processo de cadastro e login é muito rápido e intuitivo!"

ID Usuário: 42 | Data: 05/11/2025

© 2025 Hospital das Clínicas - Todos os direitos reservados

Área de Tutoriais de vídeos, Perguntas, Respostas e Feedbacks, onde os usuários podem criar e acessar vídeos de outros usuários, além de poderem escrever suas dúvidas e repostas a essas respectivas perguntas, e também avaliarem o uso da plataforma com suas opiniões.

HOSPITAL  
das Clínicas

Início

Integrantes

FAQ

Contato

Acessibilidade

A+

A-

Escuro

Vídeos Tutoriais

Aqui você encontra vídeos que explicam como usar a plataforma e os serviços digitais do Hospital das Clínicas.

[Teleatendimento] Orientação

Tutorial 1: [Teleatendimento] Orientação aos Pacientes

Acessar teleconsulta (Android)

Tutorial 2: Como acessar uma teleconsulta? (Celular Android)

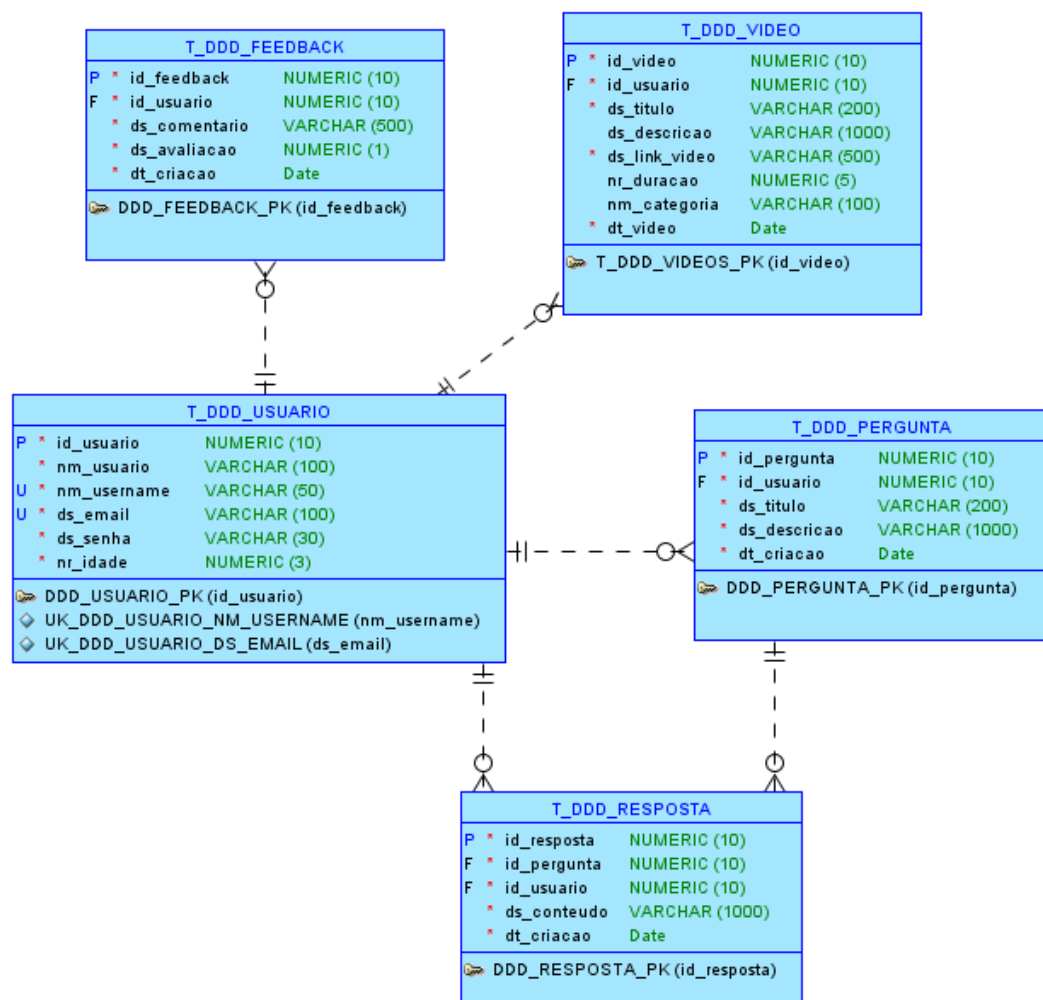
Acessar teleconsulta (iOS)

Tutorial 3: Como acessar uma teleconsulta? (Celular iOS)

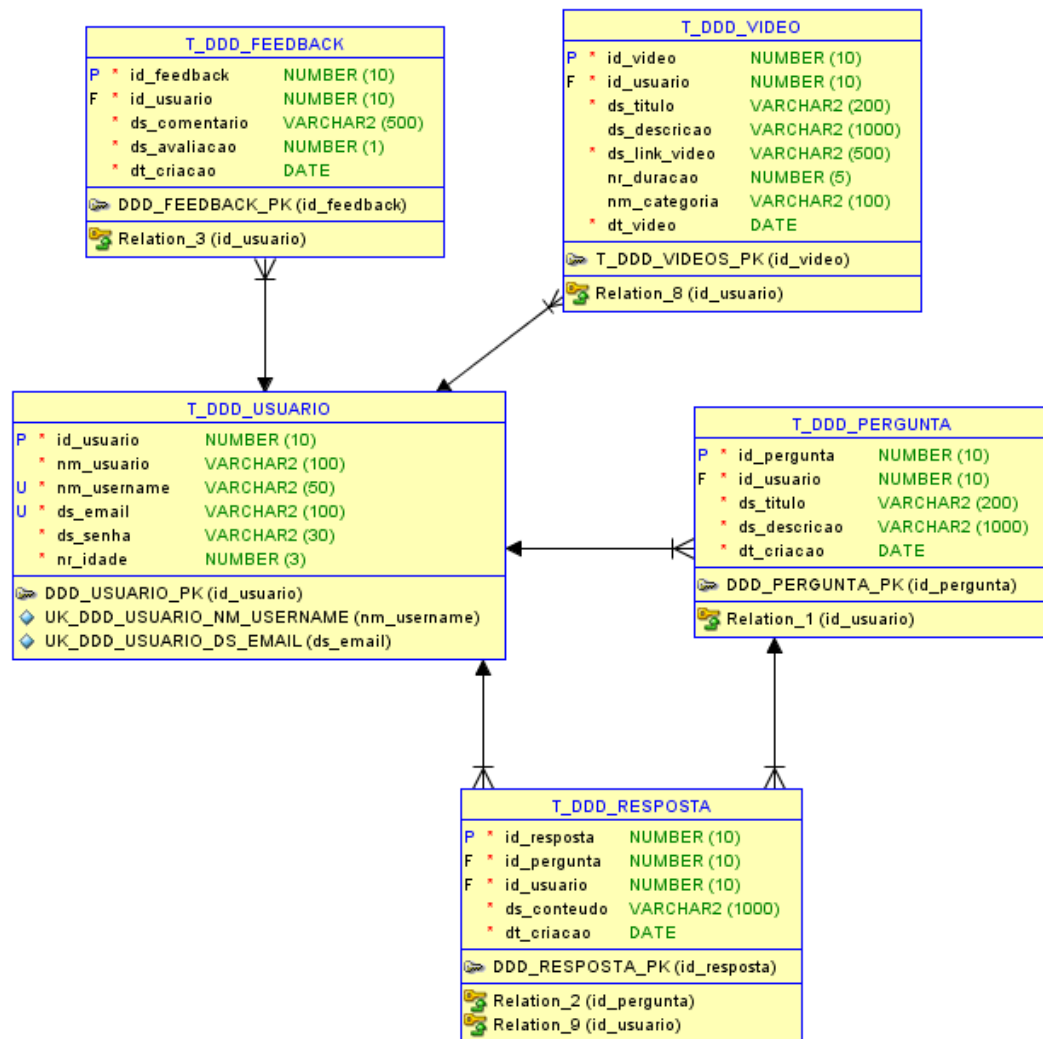
Área de Acessibilidade onde os usuários podem encontrar vídeos de como usar a plataforma e os serviços digitais do Hospital das Clínicas.

# Modelo de Entidade-Relacionamento (MER)

- Modelagem Lógica Relacional:

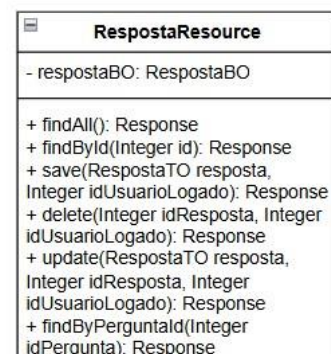
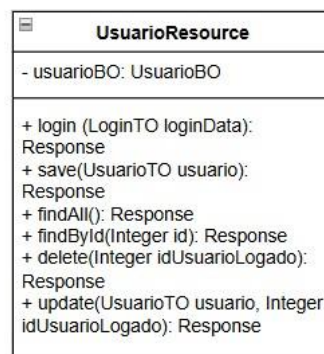
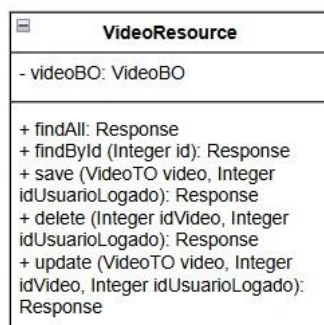
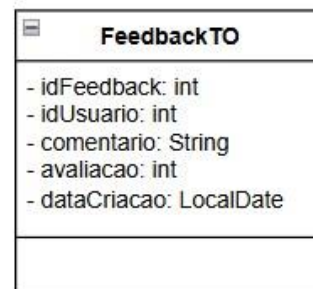
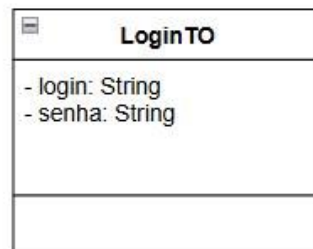
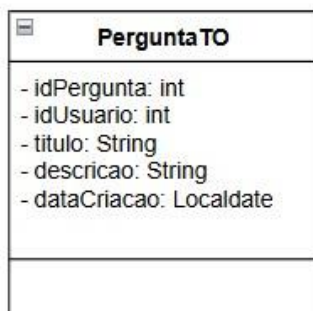
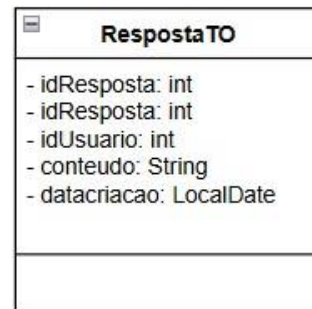
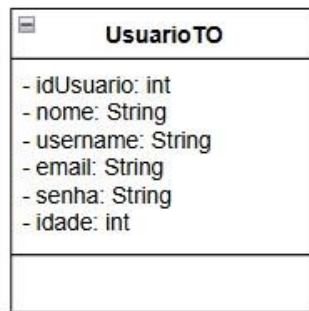


- Modelagem Física Relacional:





# Diagrama de Classes Atualizado



PerguntaResource
- perguntaBO: PerguntaBO
+ findAll: Response + findById(Integer id): Response + save(PerguntaTO pergunta, Integer idUsuarioLogado): Response + delete(Integer idPergunta, Integer idUsuarioLogado): Response + update(PerguntaTO pergunta, Integer idPergunta, Integer idUsuarioLogado): Response

FeedbackResource
- feedbackBO: FeedbackBO
+ findAll: Response + findByUsuario(Integer idUsuario): Response + save(FeedbackTO feedback, Integer idUsuarioLogado): Response + delete(Integer idFeedback, Integer idUsuarioLogado): Response + update(FeedbackTO feedback, Integer idFeedback, Integer idUsuarioLogado): Response

VideoDAO
+ save(VideoTO video): VideoTO + delete(Integer idVideo): boolean + update(VideoTO video): VideoTO + findAll(): ArrayList<VideoTO> + findByCodigo(Integer idVideo): VideoTO + findByUsuario(Integer idUsuario): ArrayList<VideoTO>

UsuarioDAO
+ save(UsuarioTO usuario): UsuarioTO + delete(Integer idUsuario): boolean + update(UsuarioTO usuario): UsuarioTO + findAll(): ArrayList<UsuarioTO> + findByCodigo(Integer idUsuario): UsuarioTO + findByEmail(String email): UsuarioTO + findByUsername(String username): UsuarioTO + findByLogin(String login, String senha): UsuarioTO

RespostaDAO
+ save(RespostaTO resposta): RespostaTO + delete(Integer idResposta): boolean + update(RespostaTO resposta): RespostaTO + findAll(): ArrayList<RespostaTO> + findByCodigo(Integer idResposta): RespostaTO + findByPergunta(Integer idPergunta): ArrayList<RespostaTO>

PerguntaDAO
+ save(PerguntaTO pergunta): PerguntaTO + delete(Integer idPergunta): boolean + update(PerguntaTO pergunta): PerguntaTO + findAll(): ArrayList<PerguntaTO> + findByCodigo(Integer idPergunta): PerguntaTO + findByUsuario(Integer idUsuario): ArrayList<PerguntaTO>

FeedbackDAO
+ save(FeedbackTO feedback): FeedbackTO + delete(Integer idFeedback): boolean + update(FeedbackTO feedback): FeedbackTO + findAll(): ArrayList<FeedbackTO> + findByCodigo(Integer idFeedback): FeedbackTO + findByUsuario(Integer idUsuario): ArrayList<FeedbackTO>

ConnectionFactory
- connection: Connection
+ closeConnection(): void + getConnection(): Connection

VideoBO
- videoDAO: VideoDAO
+ findAll(): ArrayList<VideoTO> + findByUsuarioId(Integer idUsuario): ArrayList<VideoTO> + findById(Integer id): VideoTO + save(VideoTO video): VideoTO + delete(Integer idVideo, Integer idUsuarioLogado): boolean + update(VideoTO video, Integer idUsuarioLogado): VideoTO

UsuarioBO
- usuarioDAO: UsuarioDAO
+ findAll(): ArrayList<UsuarioTO> + findById(Integer id): UsuarioTO + save(UsuarioTO usuario): UsuarioTO + delete(Integer id): boolean + update(UsuarioTO usuario): UsuarioTO + login(String login, String senha): UsuarioTO

RespostaBO
- respostaDAO: RespostaDAO
+ findAll(): ArrayList<RespostaTO> + findById(Integer id): RespostaTO + save(RespostaTO resposta): RespostaTO + delete(Integer idResposta, Integer idUsuarioLogado): boolean + update(RespostaTO resposta, Integer idUsuarioLogado): RespostaTO + findByPerguntaId(Integer idPergunta): ArrayList<RespostaTO>

PerguntaBO
- perguntaDAO: PerguntaDAO
+ findAll(): ArrayList<PerguntaTO> + findByUsuarioId(Integer idUsuario): ArrayList<PerguntaTO> + findById(Integer id): PerguntaTO + save(PerguntaTO pergunta): PerguntaTO + delete(Integer idPergunta, Integer idUsuarioLogado): boolean + update(PerguntaTO pergunta, Integer idUsuarioLogado): PerguntaTO

FeedbackBO
- feedbackDAO: FeedbackDAO
+ findAll(): ArrayList<FeedbackTO> + findByUsuarioId(Integer idUsuario): ArrayList<FeedbackTO> + findById(Integer id): FeedbackTO + save(FeedbackTO feedback): FeedbackTO + delete(Integer idFeedback, Integer idUsuarioLogado): boolean + update(FeedbackTO feedback, Integer idUsuarioLogado): FeedbackTO