

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE – CÂMPUS PELOTAS - VISCONDE DA GRAÇA
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**HESTÍGIA
UM SISTEMA DE CONTROLE DE PRESCRIÇÕES PARA O ASILO DE
MENDIGOS DE PELOTAS**

**Bianca Maria Silva dos Santos
Lucas Gabriel Teles Martins**

Pelotas, 2024

Bianca Maria Silva dos Santos
Lucas Gabriel Teles Martins

Hestígia

Trabalho de Conclusão de Curso
apresentado como requisito na disciplina
Metodologia de Pesquisa II do curso
Técnico em Desenvolvimento de Sistemas,
do Instituto Federal Sul-Rio-Grandense –
Campus Pelotas - Visconde da Graça.

Orientador: Prof. Dr. Fernando Augusto
Treptow Brod

Pelotas, 2024.

SUMÁRIO

1. INTRODUÇÃO.....	4
2. TEMA.....	5
3. MOTIVAÇÕES.....	5
4. OBJETIVOS.....	6
4.1. OBJETIVO GERAL.....	6
4.2. OBJETIVOS ESPECÍFICOS.....	6
5. ESPECIFICAÇÃO DE REQUISITOS.....	6
5.1. MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS.....	7
5.2. REQUISITOS FUNCIONAIS.....	7
5.3. REQUISITOS NÃO FUNCIONAIS.....	8
6. MODELAGEM.....	9
6.1. MODELO DE CASOS DE USO.....	9
6.2. MODELAGEM CONCEITUAL DO BANCO DE DADOS.....	10
7. Modelagem Lógica do Banco de Dados.....	12
8. TECNOLOGIAS UTILIZADAS.....	13
9. DESCRIÇÃO DO SISTEMA.....	14
9.1 Tela de login.....	14
9.2 Tela Inicial - Administrador e Médicos.....	15
9.3 Tela de Cadastro de Residentes.....	15
9.4 Tela de Cadastro de Usuários.....	16
9.5 Tela de Cadastro de Responsáveis.....	17
9.6 Tela de Cadastro de Prescrições.....	17
9.7 Tela de Residentes.....	18
9.7.1 Tela de visualizar prescrição de um residente.....	19
9.8 Tela do Perfil do Residentes.....	19
9.8.1 Tela de Editar dados do residente.....	20
10. CONSIDERAÇÕES FINAIS.....	25
11. REFERÊNCIAS.....	26
12. APÊNDICE I - Instruções SQL para Criação da Base de Dados.....	27

1. INTRODUÇÃO

Em 1880, Pelotas vivia um de seus melhores momentos em questão de economia, isso se dava principalmente por conta da grande produção de charque, sendo Pelotas um dos principais centros desse tipo de produção no Brasil, e isso trouxe grandes mudanças para a cidade. No entanto, Pelotas enfrentou uma crise econômica que afetou significativamente a cidade, deixando inúmeros indivíduos sem emprego e em condição de pobreza e desamparo, muitas dessas pessoas eram ex-trabalhadores da indústria de charque.

O AMPel (Asilo de Mendigos de Pelotas) foi fundado em 23 de setembro de 1882 pelo jornalista e fundador do jornal Correio Mercantil, Antônio Joaquim Dias, com o seguinte apelo: “A redação do Correio Mercantil solicita de todos os habitantes desta cidade um donativo qualquer para ser aplicado à construção de um edifício destinado à mendicidade”¹. Inicialmente era destinado a cuidar da população em situação de abandono nas ruas, como veteranos da Guerra do Paraguai, que chegavam a Pelotas em situações críticas de saúde física e mental e pessoas em situação de vulnerabilidade buscando alguma ajuda. Porém, de algumas décadas pra cá, o asilo passou a ter como objetivo acolher, cuidar e abrigar residentes exclusivamente aposentados acima de 60 anos, tendo um máximo de 110 residentes. O asilo é uma entidade filantrópica, sem depender de verbas públicas - somente em cima de doações e uma parcela do salário e/ou benefícios dos residentes.

O Asilo de Mendigos de Pelotas conta atualmente com 46 residentes na ala masculina e 52 na ala feminina. As enfermeiras dão assistência e cuidados na rotina de cada um dos residentes, a assistência médica é realizada três vezes por semana, o atendimento de serviço social ocorre todos os dias e a assistência odontológica somente nas sextas-feiras à tarde por um projeto da UFPel.

Atualmente, o controle das prescrições dos residentes no AMPel é feita de modo totalmente manual, de forma escrita, organizados em pastas enormes que contém todo o histórico de prescrições para cada residente, no formato de fichas em folhas de papel e organizado por funcionário.

O sistema Hestígia tem como propósito auxiliar o controle de prescrições deste lar de residentes benficiente, integrando tecnologia digital na prática da enfermagem e gestão administrativa do Asilo, tendo como finalidade facilitar o controle de prescrição dos residentes.

A implementação do nosso sistema busca não apenas otimizar a organização dos dados, mas também disponibilizar informações essenciais de forma objetiva, auxiliando a equipe multiprofissional no acompanhamento das prescrições e no cuidado integral dos residentes.

2. TEMA

Sistema de controle de prescrições para residentes no Asilo de Mendigos de Pelotas.

3. MOTIVAÇÕES

O nome Hestígia foi escolhido para refletir a junção das qualidades que o nosso sistema busca incorporar para o Asilo de Mendigos de Pelotas. Este nome foi inspirado nas deusas da mitologia grega, Héstia e Hígia. A deusa grega Héstia simboliza o acolhimento e o lar, enquanto a deusa grega Hígia está associada ao bem-estar, à saúde e à cura. Por isso, Héstia e Hígia são perfeitas para representar essa combinação de acolhimento e bem-estar, pois, juntas, essas figuras mitológicas formam a junção entre um ambiente seguro e a manutenção da saúde dos residentes acolhidos no Asilo, que são princípios fundamentais que o sistema pretende oferecer.

Acreditamos que o desenvolvimento e a implantação deste sistema poderá contribuir como uma solução inovadora para controlar de forma eficaz as prescrições e os residentes envolvidos no asilo.

A efetivação deste sistema nos motiva também por sabermos dos problemas e limitações que tem no Asilo de Mendigos de Pelotas em termos de organização e controle de prescrição e de dados dos residentes. E é por estes motivos que acreditamos que a Hestígia promoverá um melhor acompanhamento das condições de saúde dos residentes.

Essa experiência não apenas fortalece nosso aprendizado acadêmico, mas também impactará positivamente no fluxo de rotina.

4. OBJETIVOS

Os objetivos do sistema Hestígia serão apresentados logo abaixo.

4.1. OBJETIVO GERAL

Desenvolver um sistema para controlar as prescrições médicas dos residentes situados no Asilo de Mendigos de Pelotas, facilitando a organização, acompanhamento e melhoria da assistência à saúde dos mesmos.

4.2. OBJETIVOS ESPECÍFICOS

- Desenvolver um sistema capaz de gerenciar dados de residentes e seus respectivos responsáveis de maneira eficiente;
- Criar funcionalidades que facilitem o registro e a consulta de prescrições médicas;
- Proporcionar maior agilidade e organização nas demandas de saúde dos residentes do asilo;
- Garantir a segurança e a acessibilidade aos dados históricos, tanto de prescrições quanto de residentes inativos, em conformidade com a legislação vigente;
- Assegurar que as informações sobre medicamentos sejam detalhadas no momento da prescrição, incluindo horários e dosagens, para melhorar a precisão no cuidado;
- Fornecer relatórios e detalhes das prescrições, indicando o status (ativo ou inativo), o residente destinatário e o prescritor responsável.

5. ESPECIFICAÇÃO DE REQUISITOS

A especificação de requisitos foi feita para obter informações sobre o que será necessário ser implantado no sistema. Através dessas informações identificamos o que é necessário para o AMPel, além dos requisitos fundamentais para o bom funcionamento do Asilo de Mendigos de Pelotas e que eles devem ser mantidos e aprimorados.

5.1. MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS

Para saber os requisitos do sistema, foi utilizada o método Analítico, de Observação e de Conversação, com uma série de perguntas, tais como:

- Qual o principal objetivo do sistema?;
- Quais são os usuários do sistema?;
- História do Asilo de Mendigos?;
- Quantos residentes residem no Asilo?;
- Como são feitas as prescrições hoje? (conforme Figura 1);
- Como vocês querem estabelecer a função de cada utilizador do sistema?.

Figura 1: Prescrição médica feita atualmente

Prescrição médica			
<i>Medicação de uso contínuo.</i>			
1- Amplictil 100mg	1CP	20:00	
2- Glicage 500mg	2CPS	8:00	17:30
3- Glibenclamida 5mg	1CP	8:00	
4- Metropolol 25mg	1CP	20:00	
5- Hidroclorotiazida 25mg	1CP	8:00	
6- Enalopril 10mg	2CPS	8:00	20:00

Fonte: Autoria própria

5.2. REQUISITOS FUNCIONAIS

Conforme a Tabela 1 serão apresentados os requisitos funcionais. Que são aqueles que mostram a funcionalidade do sistema e o que pretende fornecer aos usuários.

Tabela 1: Requisitos Funcionais

Requisito	Nome	Descrição
REF 01	Realizar login	Os médicos e o administrador efetuarão login por meio de CPF e senha.
REF 02	Cadastrar residentes	Os médicos e o administrador irão cadastrar os dados dos residentes, como nome, endereço, e outros dados importantes.
REF 03	Cadastrar prescrições	Os médicos e o administrador poderão fazer a prescrição dos residentes no sistema. O tempo, horário, dosagem e o medicamento serão colocados em um campo texto.
REF 04	Cadastrar usuários	O administrador ou o médico poderão realizar cadastro de médicos ou outros administradores que forem atuar no asilo.
REF 05	Cadastrar Responsáveis	Os médicos e o administrador irão fazer o cadastro de responsáveis dos residentes, com os seguintes dados: Nome, endereço, contato(telefone), cidade e selecionar o residente a quem o responsável será cadastrado.
REF 06	Editar residentes	Os médicos e o administrador poderão editar os dados dos residentes
REF 07	Excluir Responsáveis	Os médicos e o administrador poderão excluir os responsáveis.
REF 08	Visualizar prescrições	O visualizador, o médico e o administrador verão as prescrições dos residentes.

REF 09	Visualizar dados dos residentes	O visualizador, o médico e o administrador verão os dados dos residentes.
REF 10	Alterar Status da Prescrição	O visualizador, o médico e o administrador poderão alterar o status da prescrição caso a prescrição que era ativa venha a se tornar inativa.

Fonte: Autoria própria.

5.3. REQUISITOS NÃO FUNCIONAIS

A seguir os requisitos não funcionais que são descritos com as características que o sistema deve apresentar, conforme a Tabela 2.

Tabela 2: Requisitos Não Funcionais

Requisito	Tipos	Descrição
RNF 01	Segurança	Realizar backups dos dados do sistema.
RNF 02	Segurança	O sistema deve garantir que o controle das prescrições não sejam perdidas após feita pelos médicos ou administrador.
RNF 03	Interoperabilidade	O sistema deverá se comunicar com o banco de dados SQL e gerenciado pelo SGBD MySQL
RNF 04	Éticos	O sistema não divulga os dados dos usuários sem autorização prévia.
RNF 05	Confiabilidade	O sistema deve ser confiável e atender aos requisitos do usuário.
RNF 06	Desenvolvimento	O sistema deverá ser implementado em uma linguagem Web.
RNF 07	Desenvolvimento	O sistema deverá ser desenvolvido com as linguagens HTML, PHP, CSS e JavaScript.
RNF 08	Proteção	Somente os usuários (administrador e médicos) terão acesso interno ao sistema Hestígia, salvo a exceção que será permitido a consulta das prescrições e

		dos dados dos residentes.
RNF 09	Disponibilidade	O sistema deve estar em funcionamento sempre que os usuários precisarem.
RNF 10	Usabilidade	O sistema deve ter uma aparência simples e acessível, permitindo que os usuários interajam com ele de forma eficiente.

Fonte: Autoria própria.

6. MODELAGEM

6.1. MODELO DE CASOS DE USO

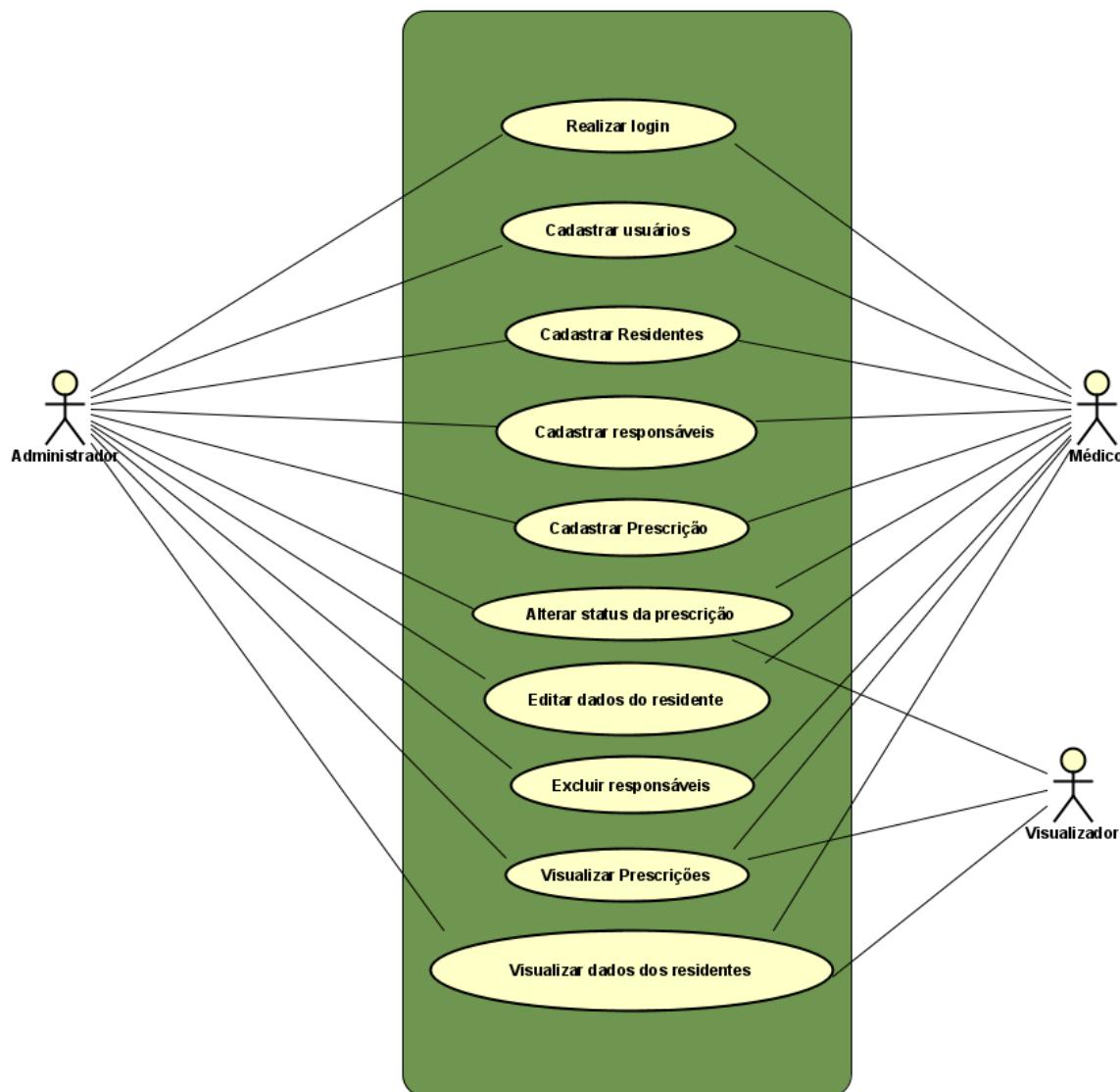
A modelagem de caso de uso serve para mostrar quem são os usuários e como eles interagem com o sistema.

No sistema de controle de prescrições, conforme a Figura 2, encontram-se três usuários, sendo eles: Administrador, Médicos e Usuário Visualizador.

O administrador e os médicos possuem acesso interno ao sistema, por meio de login. Ela possui funções como: Cadastrar prescrições médicas e os dados dos residentes, visualizar os dados dos residentes, cadastrar usuários, cadastrar e excluir responsáveis, visualizar prescrições.

O usuário visualizador não possui acesso interno ao sistema, somente será capaz de visualizar as prescrições e os dados dos residentes.

Figura 2: Diagrama de Caso de Uso



Fonte: Autoria própria.

6.2. MODELAGEM CONCEITUAL DO BANCO DE DADOS

A modelagem conceitual de banco de dados, serve para descrever as tabelas(entidades), dados e seus respectivos relacionamentos que serão armazenados no banco de dados.

A modelagem conceitual do sistema Hestígia, contida na Figura 3, apresenta um total de 5 entidades e 5 relacionamentos.

A entidade ‘usuários’ apresenta dados que fazem a distinção do usuário que estiver utilizando na hora, como: ‘cpf’ (chave primária), ‘crm’, ‘tipo_usuario’, ‘status_usuario’, ‘nome’, ‘data_nascimento’, ‘contato’ e ‘senha’ possui relacionamento de (1,n) com a tabela prescrição, um ou vários usuários realizam nenhuma ou várias prescrições e as prescrições podem ser feitas por um ou vários usuário.

Na entidade ‘prescrição’ é armazenado os dados da prescrição de um residente, como: ‘cod_prescrição’ (chave primária), ‘data_prescrição’, ‘receita’ e ‘status_receita’, fazendo relação de (1,n), como apresentado antes, com a entidade ‘usuários’. Além disso, possui relação de (0,n) com a entidade ‘residente’, onde nenhuma ou várias prescrições estão relacionadas a somente um residente, e um residente possui nenhuma ou várias prescrições.

A entidade ‘residente’ contém os principais dados dos residentes do Asilo, tais como: ‘cod_residente’ (chave primária), ‘cpf_residente’, ‘nome’, ‘data_nascimento’, ‘data_entrada’, ‘cidade’, ‘gênero’, ‘status_residente’, ‘observações’ e ‘laudo’, tendo relação de (1,1) com a entidade ‘prescrição’, no qual somente um residente se relaciona com uma ou várias prescrições e uma ou várias prescrições faz relação com um e somente um residente. A entidade ‘residente’ também se relaciona com a entidade ‘responsável’, com cardinalidade de (0,n), onde nenhum ou vários residentes possuem nenhum ou vários responsáveis, e um ou mais responsáveis faz relação a um ou vários residentes. A entidade ‘residente’ se relaciona também com a entidade ‘cidade’, com a cardinalidade (0,n) residentes pertencem a uma e somente uma cidade, e uma cidade pertence a nenhuma ou vários residentes.

A entidade ‘cidade’ contém os dados de onde os residentes são, tendo os seguintes atributos: ‘cod_cidade’ (chave primária) e ‘descrição’. E como apresentado anteriormente, ela tem relação com o residente, se relaciona de (1,1) onde somente uma cidade pertence a nenhum ou vários residentes.

A entidade ‘responsável’ contém os dados dos responsáveis dos moradores, tendo como atributo os seguintes: ‘cod_responsável’ (chave primária), ‘contato’, ‘nome’, ‘endereço’, ‘cidade’ e como apresentado anteriormente, se relaciona de (0,n) com a entidade ‘residente’, onde nenhum ou vários responsáveis se relaciona a um e somente um residente, da mesma forma que um residente se relaciona a nenhum ou vários responsáveis.

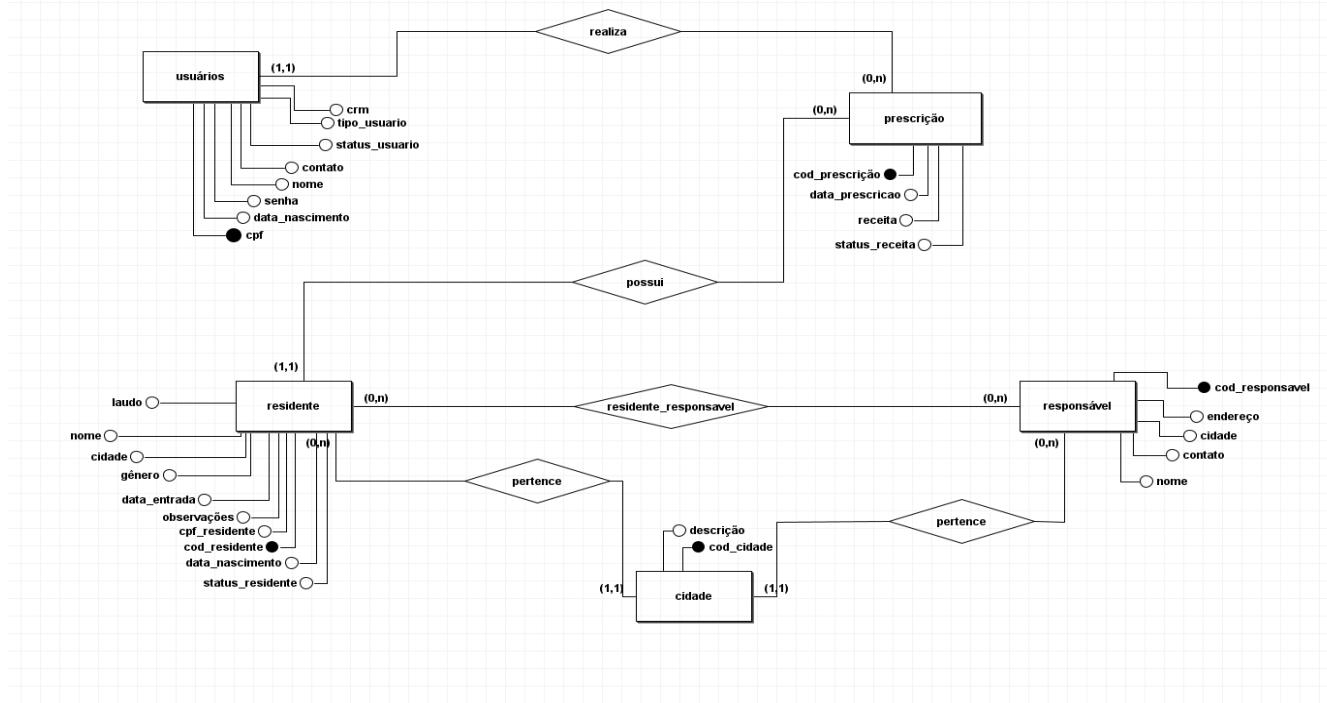


Figura 3: Modelagem Conceitual

7. Modelagem Lógica do Banco de Dados

Na modelagem lógica do sistema Hestígia, apresenta-se ao todo 4 entidades, sendo 4 delas compostas por chaves estrangeiras.

A entidade ‘prescrição’ possui 2 chaves estrangeiras. Uma delas é vinculada à entidade ‘residente’ que informa para qual residente é a prescrição e a outra é vinculada à entidade ‘usuário’ que informa quem foi o(a) prescritor(a).

Na entidade ‘residente’ possui apenas 1 chave estrangeira, ligando-a à entidade ‘cidade’ que informa de qual cidade o residente é.

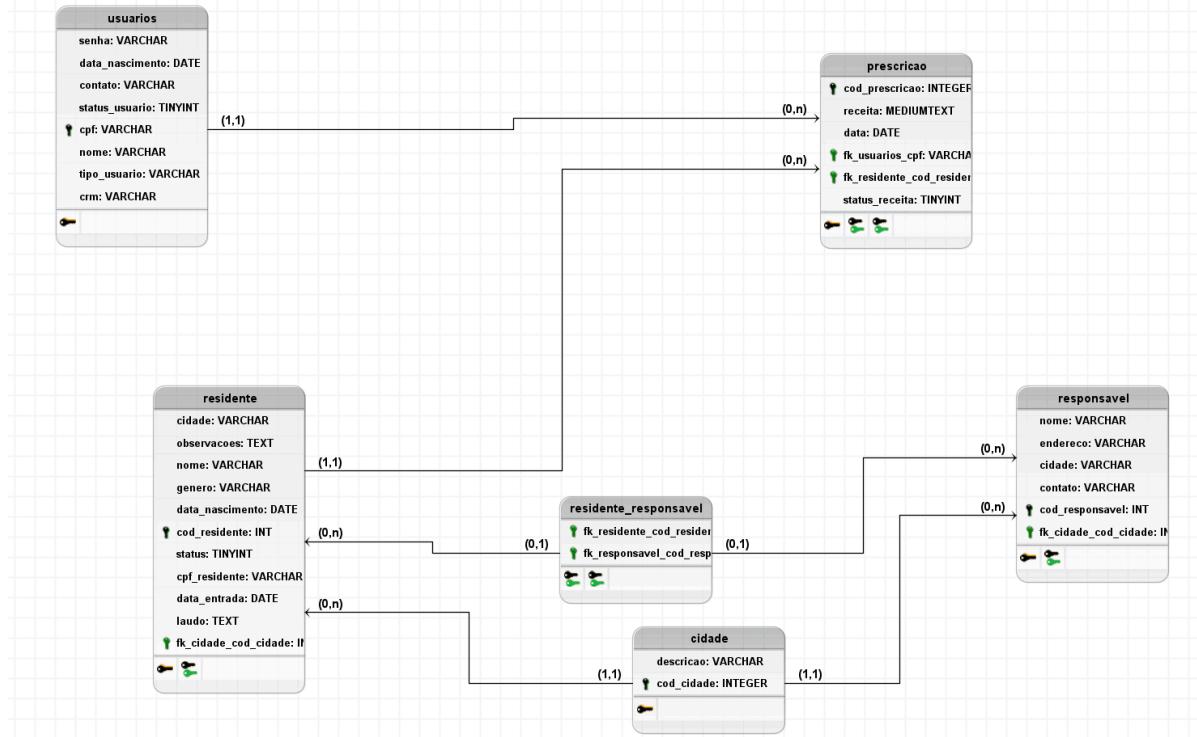
A entidade ‘responsável’ possui apenas 1 chave estrangeira, ligando-a à entidade ‘cidade’ que informa de qual cidade o responsável é.

A entidade ‘residente_responsavel’, gerada por outras duas entidades, possui 2 chaves estrangeiras, referente à entidade ‘residente’ e ‘responsável’.

Finalizando a descrição do modelo lógico de banco de dados do sistema Hestígia, finaliza-se também as descrições das modelagens utilizadas no desenvolvimento do sistema Hestígia.

Para mais informações sobre as chaves primárias, chaves estrangeiras e criação de novas tabelas, verifique as instruções sql do Apêndice I neste documento, assim como as modificações tidas entre a modelagem lógica e modelagem física (SQL).

Figura 4: Modelagem Lógica do Banco de Dados



Fonte: Autoria própria

8. TECNOLOGIAS UTILIZADAS

Para o projeto do sistema Hestígia, foram utilizadas as seguintes tecnologias:

- **AstaUML**: Linguagem gráfica utilizada para os diagramas de caso de uso;
- **BrModelo**: Ferramenta de modelagem de banco de dados, utilizado para desenvolver a modelagem conceitual, lógica e física;

Para o desenvolvimento do sistema foram utilizados as seguintes tecnologias:

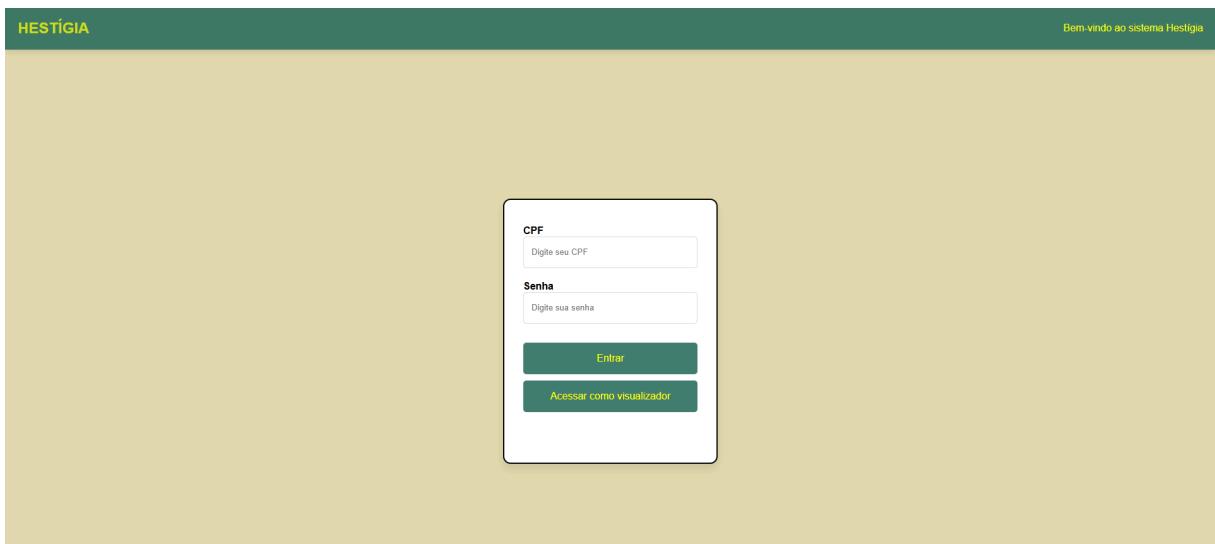
- **Visual Studio Code:** Editor de Código fonte, utilizado para o desenvolvimento dos códigos do sistema;
- **XAMPP:** Um pacote com os principais servidores de código aberto do mercado, utilizado para usar o sistema de forma local;
- **HTML:** Linguagem de formulários, utilizada para o desenvolvimento dos formulários e layout do sistema.
- **CSS:** Linguagem de estilização, utilizado para estilizar o sistema;
- **JavaScript:** Uma linguagem de programação de alto nível, utilizada para manipular comportamentos de páginas web;
- **PHP:** Uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML;

9. DESCRIÇÃO DO SISTEMA

9.1 Tela de login

A primeira tela do sistema é a “Tela de Login” (Figura 5). Nela os médicos e o administrador farão login por meio de CPF e senha, tanto o CPF e a senha estarão cadastrados no banco de dados. Caso os campos sejam preenchidos de forma errada, o sistema mostrará uma mensagem de erro, caso contrário, serão direcionados às suas respectivas telas iniciais. Além disso, esta tela apresenta um botão para os usuários visualizadores do sistema, portanto, não terão cadastro.

Figura 5: Tela de Login



Fonte: autoria própria

9.2 Tela Inicial - Administrador e Médicos

Após a realização do login, tanto o médico quanto o administrador serão direcionados para esta tela inicial (Figura 6). No cabeçalho temos uma breve navegação para acesso à lista de residentes e prescrições, no corpo principal temos as funções que ambos (administrador e médicos) poderão fazer.

Figura 6: Tela Inicial



Fonte: autoria própria

9.3 Tela de Cadastro de Residentes

Nesta tela de cadastro de residentes (Figura 7) é onde será feita a inserção de residentes no nosso sistema, colocando os dados necessários.

Figura 7: Cadastro de residentes (parte 1)

Fonte: autoria própria

Figura 7 : Cadastro de residentes (parte 2)



Cidade
Selecione uma cidade

Status: Ativo Inativo

Laudo

Observações

Cadastrar residente

Fonte: autoria própria

9.4 Tela de Cadastro de Usuários

Nesta tela de cadastro de usuários (Figura 8), será realizado o cadastro de médicos e administradores, sendo que, o CRM só se aplica aos médicos, portanto, não é um campo obrigatório.

Figura 8: Cadastro de usuários



HESTÍGIA

Inicio Residentes Prescrição Bem-vindo, Felipe Sair

Cadastro de Usuários

Nome

CPF

CRM

Número de Telefone

Data de Nascimento

dd/mm/aaaa

Senha

Cadastrar Usuário

Fonte: autoria própria

9.5 Tela de Cadastro de Responsáveis

Na tela de cadastro de responsáveis (Figura 9), será inserido alguns dados dos responsáveis para inseri-lo no sistema, e, quando cadastrado, já vincula automaticamente com o residente selecionado.

Figura 9: Cadastro de responsável



A imagem é uma captura de tela de uma interface web. No topo, há uma barra verde com o logo 'HESTÍGIA' em amarelo. À direita da barra, há links para 'Início', 'Residentes' e 'Prescrição', e uma sessão de usuário com o nome 'Bem-vindo, Felipe' e uma saída ('Sair'). O formulário principal, intitulado 'Cadastro de Responsável', está centralizado. Ele contém campos para 'Seleção de Residente' (menu suspenso com placeholder 'Selecione um residente'), 'Nome do Responsável' (campo de texto), 'Endereço' (campo de texto), 'Contato (Telefone)' (campo de texto), 'Cidade' (menu suspenso com placeholder 'Selecione uma cidade') e um botão verde 'Cadastrar Responsável'.

Fonte: autoria própria

9.6 Tela de Cadastro de Prescrições

A tela “Cadastro de prescrições” (Figura 10) é uma das principais do sistema, nela que serão inseridas as prescrições de determinado residente, o campo receita é onde será inserido tudo sobre a prescrição, como medicamento, dosagem e duração. É muito importante que o prescritor já seja definido de acordo com o login que estiver no sistema.

Figura 10: Cadastro de prescrições

Início Residentes Prescrições

Bem-vindo, Felipe Sair

Cadastro de Prescrição

Selecionar o Residente:

Receita:

Prescritor(a):

Felipe

Salvar Prescrição

Fonte: autoria própria

9.7 Tela de Residentes

Caso o usuário selecione a opção “Residentes” (Figura 11), que encontra-se no cabeçalho, ele será direcionado a tela onde lista todos os residentes cadastrados no momento no sistema. Nesta tela, é possível filtrar entre os ativos e inativos e o gênero, para uma visualização mais específica e rápida.

Nos residentes listados, temos dois botões: um para editar o perfil em questão e outro para visualizar as prescrições daquele residente. Além do mais, clicando no nome do residente, irá para a página de visualização do perfil (Figura XX) dele, com os dados cadastrados.

Figura 11: Tela de listagem dos residentes

Residente	Prescrição
Bianca Maria Silva dos Santos	Visualizar Prescrição
Andréia Rodriguês	Visualizar Prescrição

Fonte: autoria própria

9.7.1 Tela de visualizar prescrição de um residente

Ao clicar no botão “Visualizar Prescrição” na tela de listagem dos residentes (Figura 11), abre uma nova tela que mostra todas as prescrições do residente selecionado.

Figura 12: Tela de prescrição de residente selecionado

Prescrições Médicas				
Status	Residente	Prescritor(a)	Data	Receita
Ativo	Bianca Mar	Felipe	05-12-2024	oooooooi... Ver Mais
Ativo	Bianca Mar	Felipe	03-12-2024	receita teste 3... Ver Mais
Ativo	Bianca Mar	Felipe	17-11-2024	Tomar Cefaliv de 6h/6h por 2 dias. ... Ver Mais
Ativo	Bianca Mar	Felipe	17-11-2024	teste de limite teste de limite teste ... Ver Mais

Fonte: Autoria própria

9.8 Tela do Perfil do Residentes

Ao selecionar em algum residente, o sistema leva para a página do perfil dele (Figura 12), onde contém os dados que foram cadastrados e o responsável

correspondente (podendo ser mais de um), conta também com um botão de editar perfil (Figura XX) e um botão para voltar à tela anterior.

Figura 13:Tela de perfil do residente



Fonte: autoria própria

9.8.1 Tela de Editar dados do residente

Nesta tela os usuários (Administrador e médicos) poderão editar os dados do usuário selecionado na tela de listagem dos residentes cadastrados no sistema. Nesta tela, os dados vêm preenchidos de acordo com o que tem no banco de dados e o usuário poderá alterar qualquer campo, caso queira. Além disso, também tem a opção de excluir algum responsável do residente, mas caso queira adicionar um para ele, terá que ir na tela de cadastrar o responsável e selecionar o residente desejado.

Figura 14: Tela de editar dados do residente (parte 1)



Nome:

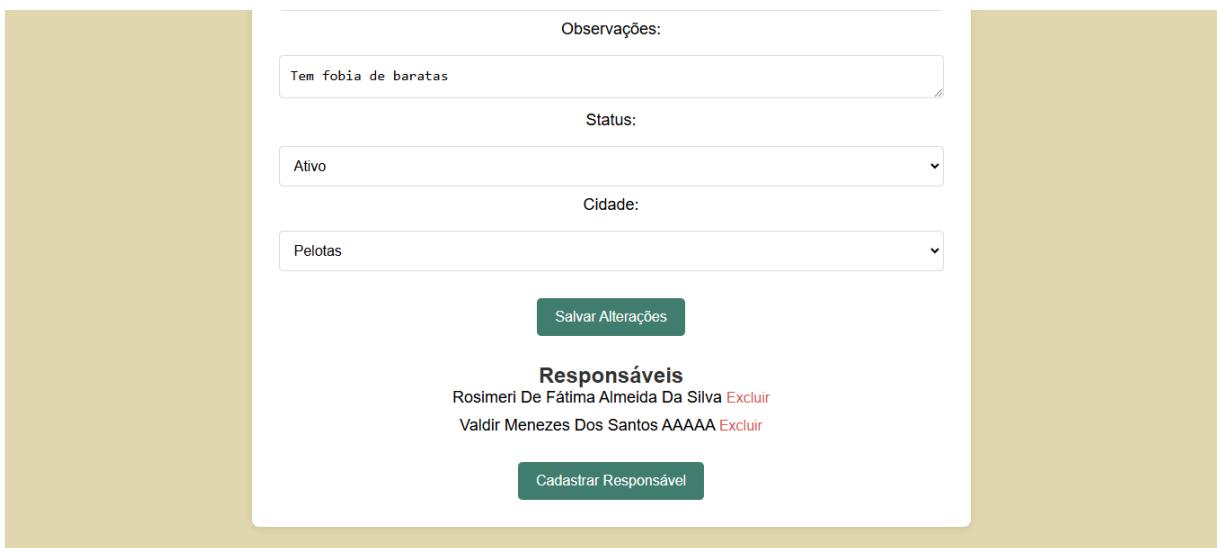
CPF:

Data de Nascimento:

Gênero:

Fonte: Autoria própria

Figura 14: Tela de editar dados do residente (parte 2)



Observações:

Status:

Cidade:

Responsáveis

Rosimeri De Fátima Almeida Da Silva [Excluir](#)

Valdir Menezes Dos Santos [AAAAA](#) [Excluir](#)

[Cadastrar Responsável](#)

[Salvar Alterações](#)

Fonte: Autoria própria

9.9 Tela de Visualizar Prescrição

Podemos visualizar a prescrição específica de um residente selecionado, nesta tela (Figura 15), temos todas as prescrições já feitas para a residente.

Figura 15: Tela de visualizar prescrições gerais

Status da receita	Residente	Prescritor(a)	Data	Receita
Ativo Alterar	Bianca Maria Silva dos Santos	Felipe	17-11-2024	Tomar Cefaliv de 6h/6h por 2 dias... Ver Mais
Ativo Alterar	Bianca Maria Silva dos Santos	Felipe	17-11-2024	teste de limite teste de limite teste ... Ver Mais

Fonte: autoria própria

9.10 Tela do Usuário Visualizador

Na tela do login (Figura 4), ao clicar em “Acessar como visualizador”, será acessado esta tela (Figura 16), nota-se que, como o usuário visualizador não tem as permissões de médico e administrador, o corpo da página vai contar somente com as opções de “residentes” (Figura 17) e “prescrição” (Figura 19).

Figura 16: Tela inicial do Visualizador

Bem-vindo ao Painel do Visualizador

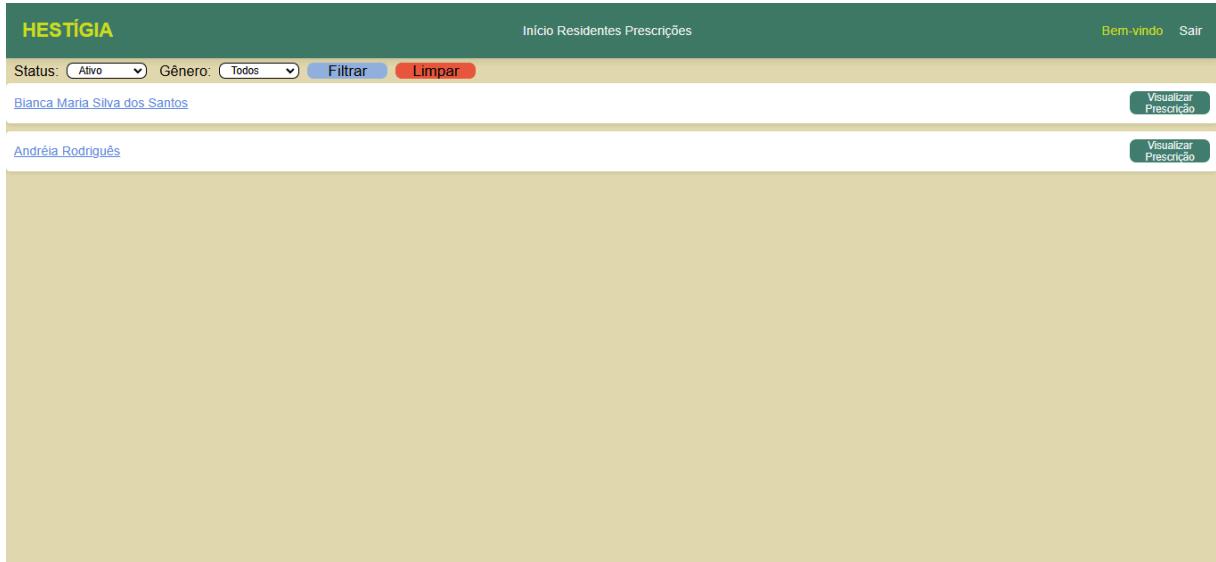
[Residentes](#) [Prescrição](#)

Fonte: autoria própria

9.11 Tela de Residentes para Visualizador

Conforme a Figura 17, a tela de residentes conta com apenas a visualização de prescrição, o botão de editar não permanece nesta tela. Aqui, o visualizador ainda poderá filtrar a busca, e selecionar o residente desejado.

Figura 17: Tela de listagem dos residentes para visualizador

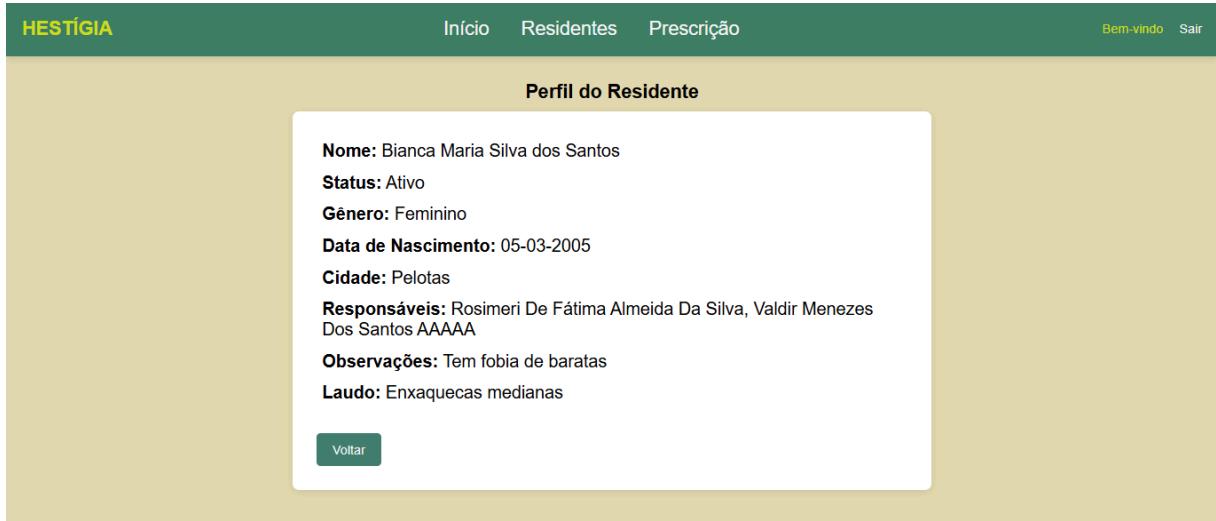


Fonte: autoria própria

9.12 Tela de Perfil de Residentes para Visualizador

O visualizador pode ver os dados dos residentes na tela de perfil dos residentes (Figura 18), que conta também com um botão para voltar à tela anterior.

Figura 18: Tela de perfil do residente para visualizador



Fonte: autoria própria

9.13 Tela de Prescrição para Visualizador

O residente pode ver a prescrição cadastrada dos residentes (Figura 19), que conta também com botão para alterar a prescrição caso ela não seja mais ativa, e um botão de ver mais que abre um *pop-up* para ler toda a prescrição. Além dos filtros.

Figura 19: Tela de Prescrição Visualizador

Status da receita	Residente	Prescritor(a)	Data	Receita
Ativo Alterar	Bianca Maria Silva dos Santos	Felipe	17-11-2024	<p>Tomar Cefaliv de 6h/6h por 2 dias. . .</p> Ver Mais
Inativo Alterar	Bianca Maria Silva dos Santos	Felipe	17-11-2024	<p>teste de limite teste de limite teste . . .</p> Ver Mais

Fonte: autoria Própria

10. CONSIDERAÇÕES FINAIS

Antes de tudo, gostaríamos de agradecer imensamente à professora Verlani Timm Hinz por nos ajudar e por apresentar a possibilidade de criar este TCC. Seu apoio foi fundamental para a realização deste trabalho.

O desenvolvimento do sistema Hestígia foi uma experiência enriquecedora, que uniu aprendizado técnico e impacto social, demonstrando o potencial da tecnologia em melhorar a rotina de serviço dos servidores do asilo.

Este trabalho não termina aqui, pois poderá ser transformado em um projeto contínuo no curso de Desenvolvimento de Sistemas, com melhorias e atualizações para os próximos anos. Fica o compromisso de continuar aprimorando o sistema, contribuindo para que ele cumpra sua missão de atender melhor os residentes do asilo e a equipe envolvida no cuidado deles. Alguns projetos para aprimoramento do sistema são:

1. Tela de “Esqueceu a Senha?”;
2. Editar o perfil do Usuário (Administrador e Médicos);
3. Adicionar todos os estados e cidades do Brasil, não apenas as de RS;
4. Adicionar imagem dos residentes e também do usuário;

Concluímos este documento com a satisfação de ter desenvolvido um sistema como trabalho de conclusão de curso, destacando o conhecimento adquirido no meio acadêmico e o crescimento alcançado, tanto no âmbito pessoal quanto profissional.

11. REFERÊNCIAS

CARTWRIGHT, Mark. **Hestia**. Enciclopédia da História Mundial. Disponível em: <<https://www.worldhistory.org/trans/pt/1-16141/hestia>>. Acesso em: 10 ago. 2024.

MOTA, P.H. **Hígia, quem foi? Origem e papel da deusa na mitologia grega** - Segredos do Mundo. Segredos do Mundo. Disponível em: <<https://segredosdomundo.r7.com/higia/>>. Acesso em: 10 ago. 2024.

RIBEIRO, Roberto. **Pelotas: Asylo de Mendigos é a melhor instituição do gênero do Rio Grande do Sul** | Jornal Tradição Regional. Disponível em: <<https://jornaltradicao.com.br/pelotas/geral/pelotas-asylo-de-mendigos-e-a-maior-instituicao-do-genero-do-rio-grande-do-sul/>>. Acesso em: 10 ago. 2024.

Requisitos funcionais e não funcionais: o que são? Mestresdaweb.com.br. Disponível em: <<https://www.mestresdaweb.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>>. Acesso em: 11 ago. 2024.

12. APÊNDICE I - Instruções SQL para Criação da Base de Dados

```
-- Banco de dados: `hestigia_bd`  
--  
CREATE DATABASE `hestigia_bd` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;  
USE `hestigia_bd`;  
  
CREATE TABLE `cidade` (  
    `descricao` varchar(200) DEFAULT NULL,  
    `cod_cidade` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
CREATE TABLE `prescricao` (  
    `cod_prescricao` int(11) NOT NULL,  
    `receita` mediumtext DEFAULT NULL,  
    `data_prescricao` date DEFAULT NULL,  
    `fk_residente_cod_residente` int(11) DEFAULT NULL,  
    `fk_usuario_cpf` varchar(11) DEFAULT NULL,  
    `status_receita` tinyint(1) DEFAULT 1  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
CREATE TABLE `residente` (  
    `cidade` varchar(255) DEFAULT NULL,  
    `observacoes` text DEFAULT NULL,  
    `nome` varchar(200) DEFAULT NULL,  
    `genero` varchar(10) DEFAULT NULL,  
    `data_nascimento` date DEFAULT NULL,  
    `cod_residente` int(11) NOT NULL,  
    `cpf_residente` varchar(14) DEFAULT NULL,  
    `data_entrada` date DEFAULT NULL,  
    `fk_cidade_cod_cidade` int(11) DEFAULT NULL,  
    `status_residente` tinyint(1) NOT NULL,  
    `laudo` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
CREATE TABLE `residente_responsavel` (  
    `fk_residente_cod_residente` int(11) DEFAULT NULL,  
    `fk_responsavel_cod_responsavel` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
CREATE TABLE `responsavel` (
```

```

`nome` varchar(200) DEFAULT NULL,
`endereco` varchar(300) DEFAULT NULL,
`cidade` varchar(100) DEFAULT NULL,
`contato` varchar(50) DEFAULT NULL,
`cod_responsavel` int(11) NOT NULL,
`fk_cidade_cod_cidade` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE `usuarios` (
`senha` varchar(100) DEFAULT NULL,
`data_nascimento` date DEFAULT NULL,
`contato` varchar(50) DEFAULT NULL,
`status_usuario` tinyint(1) DEFAULT NULL,
`nome` varchar(200) DEFAULT NULL,
`tipo_usuario` varchar(20) NOT NULL,
`cpf` varchar(11) NOT NULL,
`crm` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

-- 
-- Acionadores `usuarios`
-- 
DELIMITER $$

CREATE TRIGGER `before_insert_usuarios` BEFORE INSERT ON `usuarios` FOR EACH ROW BEGIN
    IF NEW.crm IS NULL OR NEW.crm = " THEN
        SET NEW.tipo_usuario = 'administrador';
    ELSE
        SET NEW.tipo_usuario = 'medico';
    END IF;
END
$$
DELIMITER ;

```

```

-- 
-- Índices para tabelas despejadas
-- 

```

```

-- 
-- Índices de tabela `cidade`
-- 

```

```

ALTER TABLE `cidade`
    ADD PRIMARY KEY (`cod_cidade`);

```

```

-- 
-- Índices de tabela `prescricao`
-- 

```

```

ALTER TABLE `prescricao`

```

```
ADD PRIMARY KEY (`cod_prescricao`),
ADD KEY `FK_prescricao_3` (`fk_residente_cod_residente`);

-- Índices de tabela `residente`
-- 
ALTER TABLE `residente`
ADD PRIMARY KEY (`cod_residente`),
ADD KEY `FK_residente_2` (`fk_cidade_cod_cidade`);

-- Índices de tabela `residente_responsavel`
-- 
ALTER TABLE `residente_responsavel`
ADD KEY `FK_residente_responsavel_1` (`fk_residente_cod_residente`),
ADD KEY `FK_residente_responsavel_2` (`fk_responsavel_cod_responsavel`);

-- Índices de tabela `responsavel`
-- 
ALTER TABLE `responsavel`
ADD PRIMARY KEY (`cod_responsavel`),
ADD KEY `FK_responsavel_2` (`fk_cidade_cod_cidade`);

-- Índices de tabela `usuarios`
-- 
ALTER TABLE `usuarios`
ADD PRIMARY KEY (`cpf`);

-- AUTO_INCREMENT para tabelas despejadas
-- 

-- AUTO_INCREMENT de tabela `cidade`
-- 
ALTER TABLE `cidade`
MODIFY `cod_cidade` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT de tabela `prescricao`
-- 
ALTER TABLE `prescricao`
MODIFY `cod_prescricao` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT de tabela `residente`
-- 
ALTER TABLE `residente`
```

```

MODIFY `cod_residente` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT de tabela `responsavel`
-- ALTER TABLE `responsavel`
  MODIFY `cod_responsavel` int(11) NOT NULL AUTO_INCREMENT;

-- Restrições para tabelas despejadas
-- Restrições para tabelas `residente`
-- ALTER TABLE `residente`
  ADD CONSTRAINT `FK_residente_2` FOREIGN KEY (`fk_cidade_cod_cidade`)
  REFERENCES `cidade` (`cod_cidade`) ON DELETE CASCADE;

-- Restrições para tabelas `residente_responsavel`
-- ALTER TABLE `residente_responsavel`
  ADD CONSTRAINT `FK_residente_responsavel_1` FOREIGN KEY
  (`fk_residente_cod_residente`) REFERENCES `residente` (`cod_residente`) ON
  DELETE SET NULL,
  ADD CONSTRAINT `FK_residente_responsavel_2` FOREIGN KEY
  (`fk_responsavel_cod_responsavel`) REFERENCES `responsavel`(`cod_responsavel`)
  ON DELETE SET NULL;

-- Restrições para tabelas `responsavel`
-- ALTER TABLE `responsavel`
  ADD CONSTRAINT `FK_responsavel_2` FOREIGN KEY (`fk_cidade_cod_cidade`)
  REFERENCES `cidade` (`cod_cidade`) ON DELETE CASCADE;
  COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
  CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```