

CSI602 – Banco de Dados I

Trabalho Prático

Emanuelle Ferraz Lima

Esdras Almeida Ávila

Mateus Henrique dos Santos

João Monlevade, MG

2024

Sumário

Etapa 1	3
Nome do SBD	3
Sobre o projeto	3
Requisitos	3
Etapa 2	5
Diagrama Conceitual ER	5
Etapa 3	6
Modelagem Lógica SBD	6
Etapa 4	7
Script Criação do Banco de Dados	7
Script Paciente	11
Script Agente	13
Script Médico	13
Script Doença	14
Etapa 5	15
Link para repositório GitHub	15



SICKSTEM
Sistema para Monitoramento de Doenças



UFOP
Universidade Federal
de Ouro Preto

Etapa 1:

<p>Nome do SBD</p> <p>Sistema para Monitoramento de Doenças por Bairro</p> <p>Desenvolvimento.: <i>Emanuelle Ferraz Lima – 22.1 8049</i> <i>Esdras Almeida Ávila – 22.1 8106</i> <i>Mateus Henrique dos Santos – 22.1 8109</i></p>
<p>Sobre o Projeto</p> <p><i>O sistema é uma aplicação web dedicada ao monitoramento de incidências de doenças por bairro em uma cidade específica. Utiliza dados provenientes de visitas domiciliares realizadas por agentes de saúde, bem como consultas médicas em postos de saúde. A finalidade principal é identificar e visualizar as prevalências de doenças em diferentes bairros, apresentando essas informações por meio de gráficos. O sistema armazena dados de agentes de saúde, médicos, pacientes, consultas, visitas e tipos de doenças para fornecer uma visão abrangente do panorama de saúde da comunidade.</i></p>
<p>Requisitos</p> <p><i>Para o funcionamento do Sistema Sickstem será necessário ter uma área (cidade) delimitada como um dos requisitos, uma vez que objetivo é monitorar as doenças por bairro.</i></p> <p>Outros Requisitos do Sistema:</p> <ol style="list-style-type: none">Cadastro de Doença: <i>Listar as informações necessárias para registrar uma doença, como nome, gravidade, condição, classificação, etc.</i>Listagem de Pacientes, Doenças, Médicos, Agentes, Consultas e Visitas: <i>Detalhar os critérios de pesquisa e filtragem para garantir uma busca eficiente e facilitar a localização de informações.</i>Editar ou Excluir Paciente, Médico, Agente e Doença: <i>Definir as permissões de edição/exclusão, e as ações específicas permitidas para cada usuário (administrador, médico, agente).</i>Autenticação e Autorização: <i>Implementar um sistema de login seguro com autenticação de usuário.</i> <i>Definir diferentes níveis de acesso para Agentes de Saúde, Médicos, Administradores e Visitantes.</i> <i>Garantir que apenas usuários autorizados tenham acesso a informações sensíveis.</i>



5. Monitoramento de Doenças por Região:

Implementar funcionalidades para monitorar a incidência de doenças em diferentes regiões.

Gerar relatórios e estatísticas específicas por cidade ou bairro para auxiliar na tomada de decisões.

Requisitos do Agente de Saúde:

1. Cadastro do Agente:

Incluir informações como nome, identificação, contato, senha para usar o sistema, entre outros detalhes relevantes.

2. Cadastro da Visita Domiciliar:

Definir os campos necessários para registrar informações sobre a visita, como data, hora, doença identificada etc.

3. Cadastro de Paciente:

Especificar os detalhes a serem registrados, como nome, data de nascimento, endereço, identificação etc.

Requisitos do Médico:

1. Cadastro Médico:

Incluir informações essenciais, como nome, especialização, número de registro profissional, contatos, etc.

2. Cadastro de Consulta:

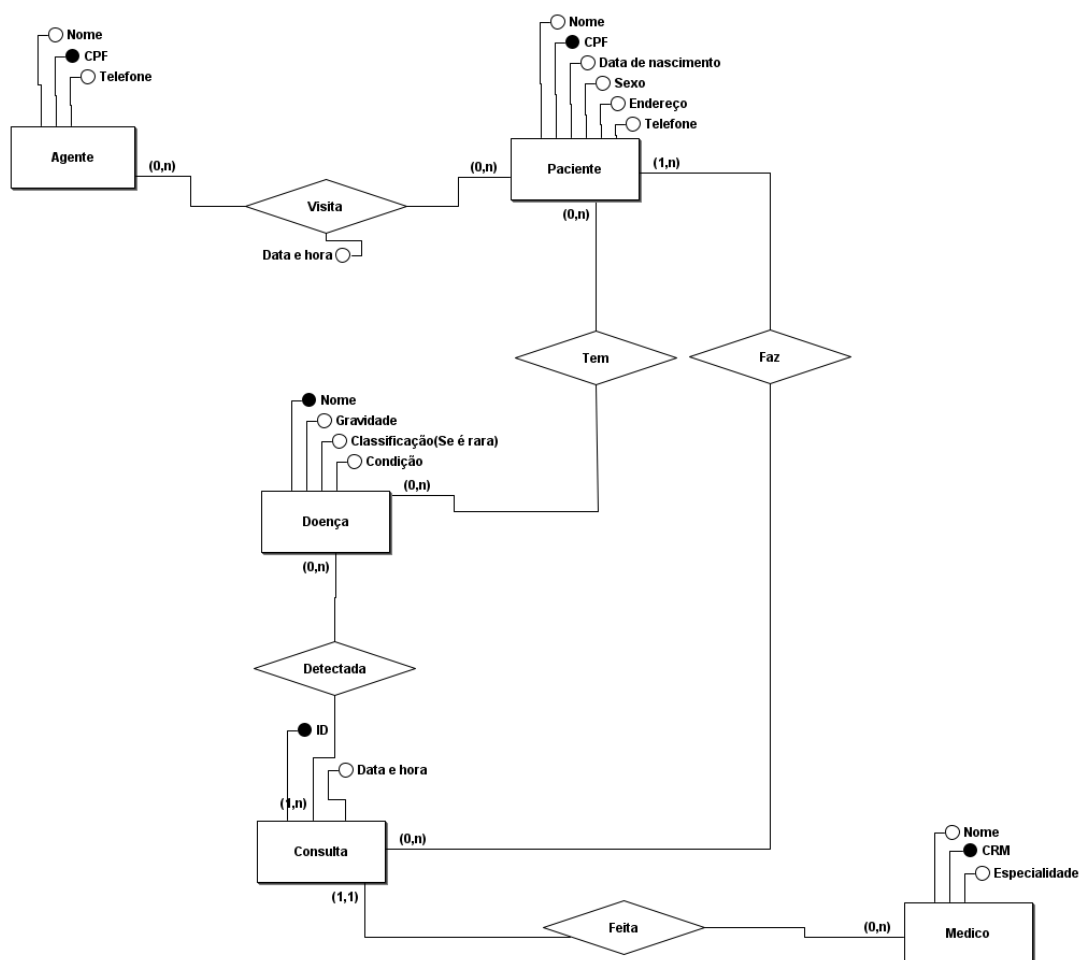
Definir os campos necessários para registrar detalhes sobre a consulta, como data, hora, paciente atendido, diagnóstico, número de consulta etc.

3. Cadastro de Pacientes:

Especificar os detalhes a serem registrados, como nome, data de nascimento, endereço, identificação etc.

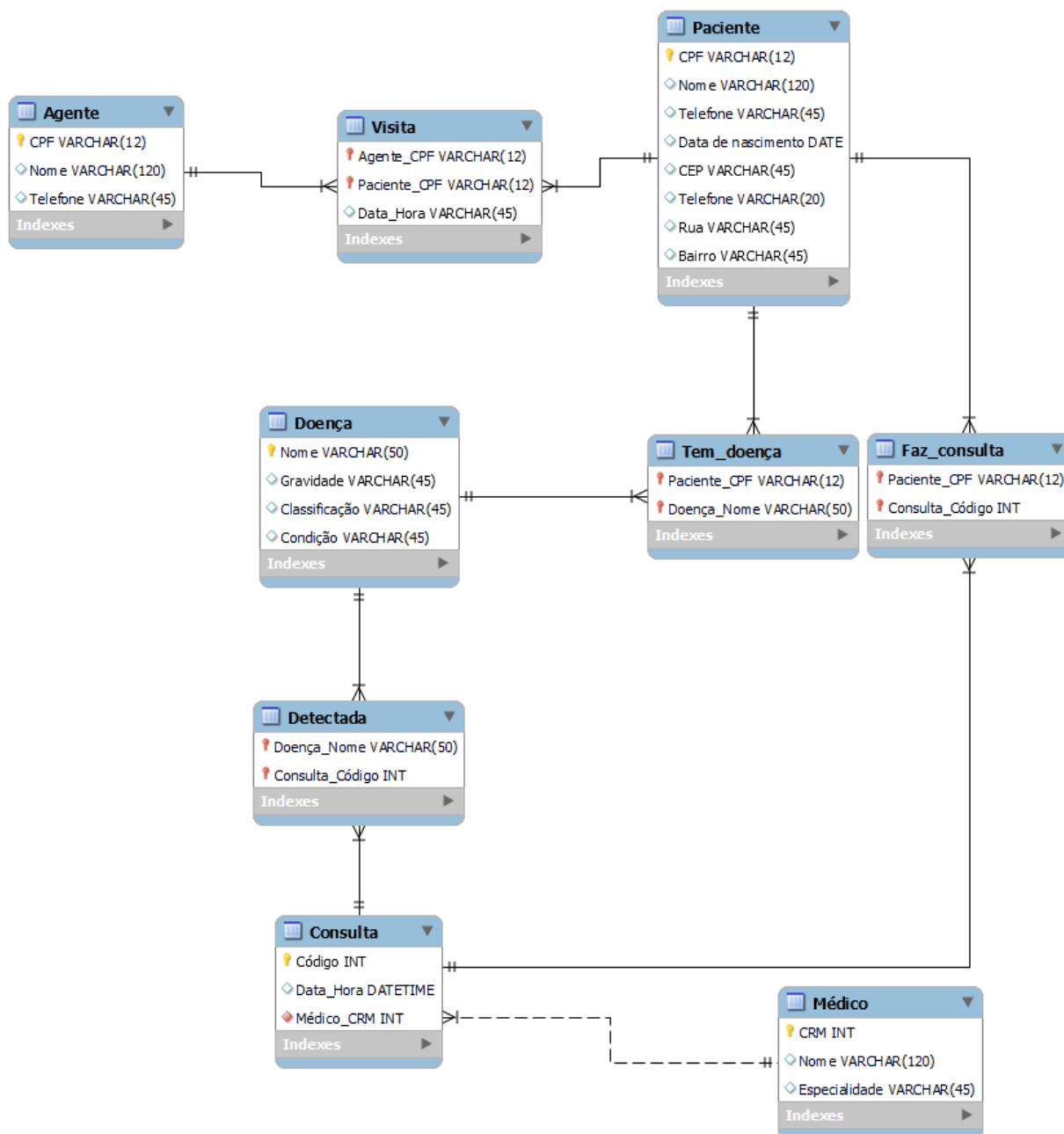
Etapa 2:

Diagrama Conceitual ER:



Etapa 3:

Modelagem Lógica do SBD:





UFOP
Universidade Federal
de Ouro Preto

Etapa 4:

Scripts de Criação e Povoamento do Banco de Dados:

Script Criação do Banco de Dados:

```
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Agente (  
  CPF VARCHAR(12) NOT NULL,  
  Nome VARCHAR(120) NULL,  
  Telefone VARCHAR(45) NULL,  
  Senha VARCHAR(120) NULL,  
  PRIMARY KEY (CPF))  
ENGINE = InnoDB;
```

```
-----  
-- Table controle_doencas_bairro.Paciente  
-----
```

```
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Paciente (  
  CPF VARCHAR(12) NOT NULL,  
  Nome VARCHAR(120) NULL,  
  Data_de_nascimento DATE NULL,  
  CEP VARCHAR(45) NULL,  
  Sexo VARCHAR(45) NULL,  
  Telefone VARCHAR(20) NULL,  
  Rua VARCHAR(45) NULL,  
  Bairro VARCHAR(45) NULL,  
  PRIMARY KEY (CPF))  
ENGINE = InnoDB;
```



-- Table controle_doencas_bairro.Doença

```
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Doenca (  
  Nome VARCHAR(50) NOT NULL,  
  Gravidade VARCHAR(45) NULL,  
  Classificacao VARCHAR(45) NULL,  
  Condicao VARCHAR(45) NULL,  
  PRIMARY KEY (Nome))  
ENGINE = InnoDB;
```

-- Table controle_doencas_bairro.Médico

```
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Medico (  
  CRM VARCHAR (15) NOT NULL,  
  Nome VARCHAR(120) NULL,  
  Especialidade VARCHAR(45) NULL,  
  Senha VARCHAR(120) NULL,  
  PRIMARY KEY (CRM))  
ENGINE = InnoDB;
```

-- Table controle_doencas_bairro.Consulta

```
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Consulta (  
  Codigo INT NOT NULL,  
  Data_Hora DATETIME NULL,  
  Medico_CRM VARCHAR(15) NOT NULL,  
  PRIMARY KEY (Codigo),
```




```
INDEX fk_Consulta_Medico_idx (Medico_CRM ASC),
CONSTRAINT fk_Consulta_Medico
    FOREIGN KEY (Medico_CRM)
    REFERENCES controle_doencas_bairro.Medico (CRM)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE = InnoDB;

-----

-- Table controle_doencas_bairro.Detectada
-----

CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Detectada (
    Doenca_Nome VARCHAR(50) NOT NULL,
    Consulta_Codigo INT NOT NULL,
    PRIMARY KEY (Doenca_Nome, Consulta_Codigo),
    INDEX fk_Doenca_has_Consulta_Consulta1_idx (Consulta_Codigo ASC),
    INDEX fk_Doenca_has_Consulta_Doenca1_idx (Doenca_Nome ASC),
    CONSTRAINT fk_Doenca_has_Consulta_Doenca1
        FOREIGN KEY (Doenca_Nome)
        REFERENCES controle_doencas_bairro.Doenca (Nome)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_Doenca_has_Consulta_Consulta1
        FOREIGN KEY (Consulta_Codigo)
        REFERENCES controle_doencas_bairro.Consulta (Codigo)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table controle_doencas_bairro.Visita
```



```
-----  
  
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Visita (  
  Agente_CPF VARCHAR(12) NOT NULL,  
  Paciente_CPF VARCHAR(12) NOT NULL,  
  Data_Hora DATETIME NULL,  
  PRIMARY KEY (Agente_CPF, Paciente_CPF),  
  INDEX fk_Agente_has_Paciente_Paciente1_idx (Paciente_CPF ASC) ,  
  INDEX fk_Agente_has_Paciente_Agente1_idx (Agente_CPF ASC) ,  
  CONSTRAINT fk_Agente_has_Paciente_Agente1  
    FOREIGN KEY (Agente_CPF)  
      REFERENCES controle_doencas_bairro.Agente (CPF)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT fk_Agente_has_Paciente_Paciente1  
    FOREIGN KEY (Paciente_CPF)  
      REFERENCES controle_doencas_bairro.Paciente (CPF)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table controle_doencas_bairro.Tem_doença  
-----
```

```
CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Tem_doenca (  
  Paciente_CPF VARCHAR(12) NOT NULL,  
  Doenca_Nome VARCHAR(50) NOT NULL,  
  PRIMARY KEY (Paciente_CPF, Doenca_Nome),  
  INDEX fk_Paciente_has_Doenca1_Doenca1_idx (Doenca_Nome ASC) ,  
  INDEX fk_Paciente_has_Doenca1_Paciente1_idx (Paciente_CPF ASC) ,  
  CONSTRAINT fk_Paciente_has_Doenca1_Paciente1  
    FOREIGN KEY (Paciente_CPF)  
      REFERENCES controle_doencas_bairro.Paciente (CPF)
```



```
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT fk_Paciente_has_Doenca1_Doenca1
FOREIGN KEY (Doenca_Nome)
REFERENCES controle_doencas_bairro.Doenca (Nome)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table controle_doencas_bairro.Faz_consulta
-----

CREATE TABLE IF NOT EXISTS controle_doencas_bairro.Faz_consulta (
  Paciente_CPF VARCHAR(12) NOT NULL,
  Consulta_Codigo INT NOT NULL,
  PRIMARY KEY (Paciente_CPF, Consulta_Codigo),
  INDEX fk_Paciente_has_Consulta_Consulta1_idx (Consulta_Codigo ASC),
  INDEX fk_Paciente_has_Consulta_Paciente1_idx (Paciente_CPF ASC),
  CONSTRAINT fk_Paciente_has_Consulta_Paciente1
  FOREIGN KEY (Paciente_CPF)
  REFERENCES controle_doencas_bairro.Paciente (CPF)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT fk_Paciente_has_Consulta_Consulta1
  FOREIGN KEY (Consulta_Codigo)
  REFERENCES controle_doencas_bairro.Consulta (Codigo)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Script Paciente:

```
INSERT INTO paciente(CPF, Nome, Data_de_nascimento, CEP, Sexo, Telefone, Rua,
Bairro)
VALUES ('000000000', 'Alberto Kalazans Ferraz', '2003-02-12', '35930-010',
```



UFOP

Universidade Federal
de Ouro Preto

'Masculino', '(31) 95123-4659', 'Brasília', 'Carneirinhos'),
(012556771-00', 'Antoniela Amaral Lima', '1995-06-25', '35931-014', 'Masculino',
'(31) 94746-0225', 'Trinta', 'Loanda'),
(023876234-08', 'Janaina Almeida de Oliveira', '1998-12-20', '35930-478', 'Feminino',
'(31) 98723-1122', 'Avenida Gentil Bicalho', 'Carneirinhos'),
(034123567-09', 'Maria Ferreira Pinheiro', '1974-02-02', '35930-020', 'Feminino', '(31)
99911-8765', 'Rua Maria Beatriz', 'Carneirinhos'),
(075424678-02', 'Camilo Bandeira Nobre', '1995-09-18', '35930-003', 'Masculino',
'(31) 98544-0976', 'Avenida Getúlio Vargas', 'Carneirinhos'),
(076434987-06', 'Tânia Rosa Brito', '1978-05-12', '35931-058', 'Feminino', '(31)
99967-1234', 'Rua Cinquenta e Cinco', 'Cruzeiro Celeste'),
(076454196-04', 'Cecília Aguiar Ribeiro', '1998-12-05', '35932-135', 'Feminino', '(31)
99923-0986', 'Rua Ayrton Sena', 'Cruzeiro Celeste'),
(095434196-03', 'Bruno Santos Rebouças', '2000-08-09', '35931-063', 'Masculino',
'(31) 98799-0976', 'Rua Alumínio', 'Cruzeiro Celeste'),
(098123876-07', 'Thiago Fernandes Lima', '1980-11-30', '35930-008', 'Masculino',
'(31) 98512-0012', 'Rua Esmeraldas', 'Carneirinhos'),
(111444555-20', 'Kemelly Santos Brito', '2000-12-03', '39935-000', 'Feminino', '(33)
98765-0087', 'Avenida Armando Fajardo', 'Loanda'),
(123432678-01', 'Amanda Dias Costa', '1980-03-30', '35930-003', 'Feminino', '(31)
91239-0043', 'Avenida Getúlio Vargas', 'Carneirinhos'),
(124354668-55', 'Almir Costa Pereira', '1960-03-15', '35931-011', 'Masculino', '(31)
95342-2541', 'Av. Cândido Dias', 'Loanda'),
(159357456-21', 'Beraci Santos Pousa', '1990-05-10', '35930-164', 'Masculino', '(31)
93146-5731', 'São Lourenço', 'Rosário'),
(274297513-81', 'Lucas Silva Feliciano', '2003-02-24', '35930-168', 'Masculino', '(31)
98546-7312', 'Sérgio Pôrto', 'Rosário'),
(345666732-22', 'Maria Madalena Gonçalves', '1987-04-20', '31935-071', 'Feminino',
'(31) 99335-0010', 'Rua Manoel Almeida', 'Cruzeiro Celeste'),
(346735149-11', 'Carlos Almeida Nascimento', '2000-03-12', '35930-167', 'Masculino',
'(31) 99856-4685', 'Manoel Bandeira', 'Rosário'),
(573164982-64', 'Nilma Figueiredo da Silva', '1980-09-01', '35931-012', 'Masculino',
'(31) 93494-5567', 'Vinte e Quatro', 'Loanda'),
(649754124-02', 'Luisa Ávila Cristina', '2001-07-20', '35930-170', 'Feminino', '(31)
7544-9968', 'Dr. Gastão Veloso', 'Rosário'),
(784596354-12', 'Vitor Gomes Matos', '2000-06-25', '35930-165', 'Masculino', '(31)
986495-2349', 'Barra Mansa', 'Rosário'),
(813464567-71', 'Lídia Clemente Gasparini', '1999-12-25', '35930-169', 'Masculino',



UFOP

Universidade Federal
de Ouro Preto

('31) 9864-1241', 'Apólo XI', 'Rosário'),
('888999023-11', 'Mateus Henrique dos Santos', '1999-01-01', '31935-073',
'Masculino', '(31) 99919-0012', 'Avenida Armando Fajardo', 'Cruzeiro Celeste'),
('890045765-05', 'Diogo Alves Santana', '2000-04-22', '35931-111', 'Masculino', '(31)
97898-0965', 'Rua Berlim', 'Cruzeiro Celeste'),
('987098123-10', 'João Antônio Vieira do Nascimento', '1970-03-15', '35930-007',
'Masculino', '(31) 99977-8888', 'Rua Luiz Ferreira', 'Carneirinhos'),
('998246531-54', 'Guilherme Sales Smith', '2000-09-23', '35931-072', 'Masculino', '(31)
99574-4407', 'Trinta e Quatro', 'Loanda');

Script Agente:

INSERT INTO agente(CPF, Nome, Telefone, Senha)

VALUES ('123321231-01', 'Camila Ribeiro Lima', '(31) 98733-3434', md5('camila123')),
('456654564-02', 'Julia da Costa Silva', '(31) 99999-8778', md5('julia456')),
('789987897-03', 'Lucas Almeida de Souza', '(31) 98763-3241', md5('lucas789')),
('890098908-04', 'Alana Soares Brito', '(31) 99456-8798', md5('alana123')),
('098890980-05', 'Maria do Carmo Oliveira', '(31) 97654-9898', md5('maria456')),
('987789897-06', 'Joaquim Rodrigues Pinheiro', '(31) 99715-2233',
md5('joaquim789')),
('654456564-07', 'Vitória Andrade Camargo', '(31) 97456-0987', md5('vick123')),
('321123213-08', 'Daniela Carvalho Pinto', '(31) 98512-2334', md5('dani456')),
('231123456-09', 'Carlos Emanuel Silva', '(31) 99912-8750', md5('carlos789')),
('546789509-10', 'Barbara Aguiar Santos', '(31) 97045-9966', md5('barbie123')),
('076543218-11', 'Ana Maria Sousa Miranda', '(31) 98908-2134',
md5('aninha456')),
('109876543-12', 'Gabriela Gomes Ribeiro ', '(31) 99976-3456', md5('gabie789'));

Script Médico:

INSERT INTO medico(CRM, Nome, Especialidade, Senha)

VALUES ('111111/MG', 'Eduardo Andrade Costa', 'Clínico Geral', md5('edu123')),
('222222/MG', 'Karla Ribeiro Gonçalves', 'Clínico Geral', md5('karla234')),
('333333/ES', 'Fernanda Machado de Oliveira', 'Obstetrícia', md5('fer456')),
('444444/MG', 'Danilo Lopes Braga', 'Pediatria', md5('danilo789')),
('555555/BA', 'João Carlos Soares de Souza', 'Clínico Geral', md5('jc123')),



UFOP

Universidade Federal
de Ouro Preto

('666666/MG', 'Júnia Santos Amaral', 'Ortopedia', md5('junia456')),
('777777/MG', 'Camilo Eduardo Almeida', 'Clínico Geral', md5('camilo789')),
('888888/RJ', 'Maria Fernanda Ferraz Oliveira', 'Ginecologia', md5('mafe123')),
('999999/MT', 'Gabriel Santos Braga', 'Infectologia', md5('gabri456')),
('123456/MG', 'Stella Cabral de Souza', 'Endoscopia', md5('stella789'));

Script Doenças:

INSERT INTO doenca (Nome, Gravidade, Classificacao, Condicao)

VALUES

('Gripe', 'Moderada', 'Viral', 'Contagiosa'),
('Diabetes', 'Alta', 'Metabólica', 'Não contagiosa'),
('Hipertensão', 'Moderada', 'Cardíaca', 'Não contagiosa'),
('Asma', 'Moderada', 'Respiratória', 'Não contagiosa'),
('Enxaqueca', 'Moderada', 'Neurológica', 'Não contagiosa'),
('Dengue', 'Alta', 'Viral', 'Contagiosa'),
('Artrite', 'Baixa', 'Autoimune', 'Não contagiosa'),
('Osteoporose', 'Baixa', 'Óssea', 'Não contagiosa'),
('Alergia Alimentar', 'Moderada', 'Imunológica', 'Não contagiosa'),
('Depressão', 'Alta', 'Mental', 'Não contagiosa');

Observação:

As demais tabelas foram povoadas através da aplicação (Site do Sickstem).

Etapa 5:

Código Fonte no GitHub:

Link do repositório: <https://github.com/MateusSaantos/SICKSTEM>