**Projeto Integrador 2º Semestre - DSM**

**Disciplinas:**

Banco de Dados

Desenvolvimento Web II

Engenharia de Software II

**Professores:**

Bruno

Nilton

Orlando

**Grupo(n) / Nome da Empresa:**

Sistema:

|  |
| --- |
| **Integrantes** |
| Darlan dos Santos Oliveira Júnior |
| Felipe Peliçari Candido |
| Maria Dulce Leão Marcicano |
| Pedro Rufino da Mata Neto |
| Stephan Mendes de Oliveira |

Fatec Araras

2024

**FICHA DE CONTROLE - PROJETO INTERDISCIPLINAR**

**DISCIPLINA CHAVE: Engenharia de Software II - PI II**

**PROFESSOR: Bruno Henrique de Paula Ferreira**

**GRUPO:** Nome do grupo **SEMESTRE:** Escolher um item.

**TÍTULO DO PROJETO: SSU – Agendamento de Saúde Único**

**DATA DA APRESENTAÇÃO: 25/06/2024**

**NOTA:**

**INTEGRANTES DO GRUPO:** Nome grupo

|  |  |
| --- | --- |
| **Nome** | **Nota Individual** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Araras, 25 de junho de 2024**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Professor Bruno Henrique de Paula Ferreira**

Sumário

Sumário

[1. Apresentação da Empresa 5](#_Toc161762654)

[1.1 Missão 5](#_Toc161762655)

[2. Escopo do sistema 6](#_Toc161762656)

[2.1 Objetivos do projeto 6](#_Toc161762657)

[2.2 Técnica de levantamento de requisitos 6](#_Toc161762658)

[2.3 Requisitos funcionais 6](#_Toc161762659)

[2.4 Requisitos não funcionais 6](#_Toc161762660)

[3. Documentação do Sistema 7](#_Toc161762661)

[3.1 Metodologia de Desenvolvimento 7](#_Toc161762662)

[3.2 Diagramas UML 7](#_Toc161762663)

[Diagrama de caso de uso 7](#_Toc161762664)

[Diagrama de classes 7](#_Toc161762665)

[Diagrama de sequência 7](#_Toc161762666)

[3.3 Modelo Conceitual 7](#_Toc161762667)

[4. Testes e Qualidade 8](#_Toc161762668)

[5. Considerações Finais 8](#_Toc161762669)

# Apresentação da Empresa

Somos a SmartCode, uma empresa de tecnologia dedicada a criar soluções de inovadoras através de softwares para atender às necessidades do mercado atual e de nossos clientes. Fornecemos serviços terceirizados em sua empresa ou também uma variedade de softwares disponíveis para venda.



## 

## 1.1 Missão

## Participar ativamente da comunidade de desenvolvimento de software, compartilhando conhecimento, contribuindo para projetos de código aberto e promovendo o avanço da tecnologia como um todo. Buscar sempre por novas tecnologias, metodologias e abordagens para garantir que os produtos desenvolvidos sejam de qualidade e atendam às necessidades do mercado em constante evolução.

1.2 Visão

Participar ativamente da comunidade de desenvolvimento de software, compartilhando conhecimento, contribuindo para projetos e promovendo o avanço da tecnologia como um todo.

1.3 Valores

Inovação: Buscar constantemente por novas ideias, tecnologias e abordagens para oferecer soluções criativas e diferenciadas.

Qualidade: Comprometer-se com altos padrões de qualidade em todos os produtos e serviços entregues, garantindo a satisfação dos clientes e a confiabilidade das soluções.

Colaboração: Fomentar um ambiente de trabalho colaborativo, onde a comunicação aberta, o trabalho em equipe e o compartilhamento de conhecimento são valorizados.

Foco no Cliente: Colocar as necessidades e expectativas dos clientes em primeiro lugar, buscando entender profundamente seus desafios e oferecer soluções que agreguem valor real aos seus negócios.

## 1.4 Link Repositório

<https://github.com/Jrdotan/Medcontrol-Proj-2SEM-Grupo-1>

# Escopo do sistema

# Criação de um sistema médico para uso dos hospitais e dos pacientes, que deve armazenar e gerenciar os prontuários de diversos pacientes, onde o diferencial será o compartilhamento das informações entre diferentes instituições a fim de facilitar o atendimento dos pacientes e o trabalho dos funcionários.

O sistema também poderá guardar exames e resultados recentes feitos pelos pacientes para eventuais consultas com outros médicos.

Com o armazenamento das informações do prontuário, o sistema deve ser capaz de separá-las e categorizá-las visando a criação de uma interface informativa com gráficos e tabelas mostrando os índices e a gravidade das doenças em diferentes regiões da cidade.

## Objetivos do projeto

Atualmente estamos passando por um surto grave de dengue, mas a população não está devidamente informada sobre a gravidade da situação e isso pode ocasionar em um aumento dos casos devido a falta de informações para que seja claro para a população a necessidade de uma intervenção e controle nos focos da doença.

O software tem o objetivo de comunicar e conscientizar a população sobre focos de doenças próximos, para que assim inicie um movimento para combater e minimizar a gravidade nas regiões da cidade.

## 2.2 Técnica de levantamento de requisitos

Com base no tema designado, criamos um grupo no Trello e um repositório no Github para discutir qual programa será desenvolvido, e seus requisitos. Chegamos nesses requisitos através da modelagem do software usando diagramas e protótipos para deixar mais claro a visualização do escopo, seguindo o tema proposto (saúde).

## Requisitos funcionais

Sistema de cadastro/login para médicos e funcionários instituições;

Cadastro e gerenciamento de prontuários;

Consulta de prontuários;

Categorização das doenças;

Interface informativa exibindo gráficos e tabelas;

Filtro de índice de doenças;

Acesso a prontuários cadastrados;

Armazenamento de exames e resultados;

Integração com sistemas externos.

## 2.4 Requisitos não funcionais

Segurança e integridade das informações dos pacientes;

Acesso à internet para acessar informações em tempo real;

Suporte para mobile;

Desempenho e escalabilidade. O sistema deve suportar grande quantidade de dados e crescimento.

**2.5 Cronograma**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tarefas** | **05/03** | **12/03 a**  **19/03** | **26/03 a**  **02/04** | **03/11 a 9/11** | **10/11 a 16/11** | **17/11 a 23/11** | **25/06** |
| Definição Grupos | X |  |  |  |  |  |  |
| Criação Empresa |  | X |  |  |  |  |  |
| Escopo Sistema |  |  | X |  |  |  |  |
| Requisitos |  |  | X | X | X |  |  |
| Diagramas |  |  | X | X | X |  |  |
| Protótipo |  |  |  |  | X | X |  |
| Documentação |  | X | X | X | X | X |  |
| Entrega |  |  |  |  |  | X |  |
| Apresentação |  |  |  |  |  |  | X |

# Documentação do Sistema

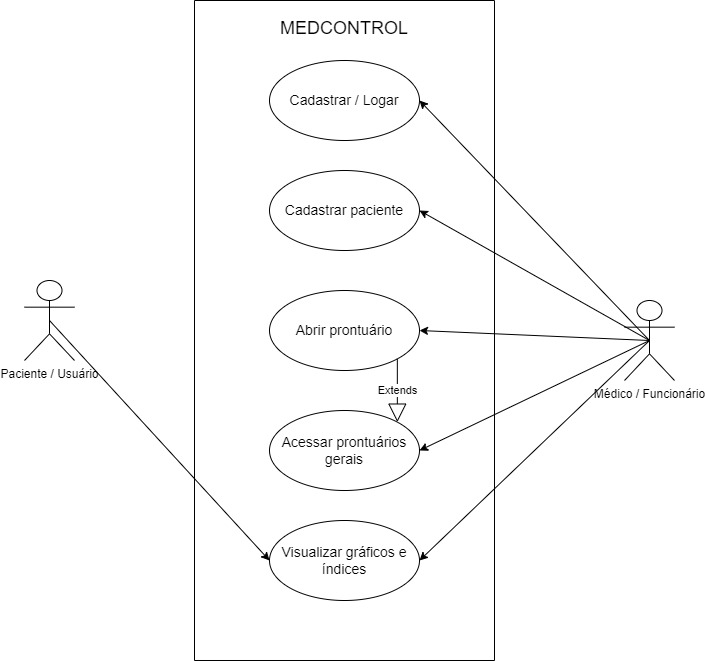
Neste capítulo será apresentado a metodologia e a engenharia de software usada para a criação do sistema, onde será abordado diagramas, textos e as metodologias ágeis.

## Metodologia de Desenvolvimento

Utilizamos uma metodologia voltada para o XP (Extreme Programming) com reuniões constantes para análise de requisitos e programação em pares. Focamos em um desenvolvimento mais eficiente e voltado para a funcionalidade do sistema para entregar um produto funcional o mais breve possível.

## 3.2 Diagramas UML

## Diagrama de caso de uso



Nesse diagrama de casos de uso temos as seguintes situações:

**Cadastro / Login:** Qualquer usuário pode acessar nosso programa para visualização dos gráficos, índices e mapas. Entretanto funcionários e médicos das instituições médicas deverão fazer um cadastro e realizar o login com uma conta própria para utilizar as demais funcionalidades do sistema referente ao prontuário hospitalar

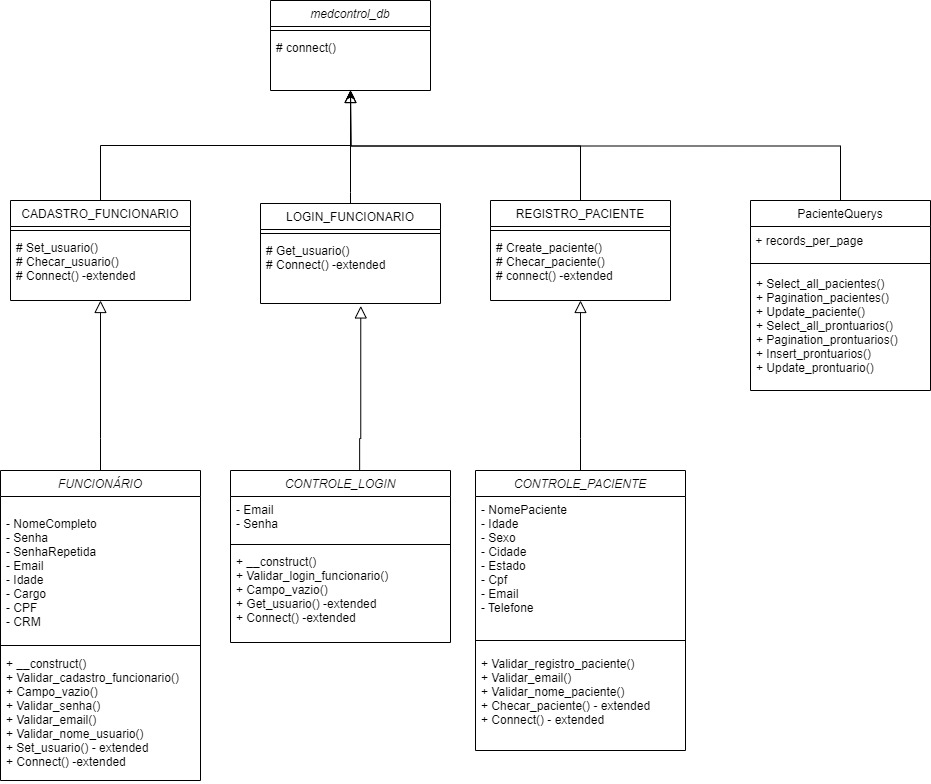
**Cadastrar paciente:** É a parte onde o funcionário da instituição irá criar/completar o cadastro de um paciente assim que ele chega no hospital.

**Abrir prontuário:** Após a realização dos exames com o médico, o mesmo terá uma aba para criar um prontuário, que irá receber as informações do cadastro do funcionário e os resultados do exame com as informações das doenças.

**Acessar prontuários gerais**: Uma aba apenas para as instituições para facilitar o gerenciamento e movimento dos prontuários e a troca de informações entre os hospitais, visando um atendimento mais rápido visto que um paciente pode ter informações já cadastradas em outras unidades.

**Visualizar índices e gráficos:** Área do programa onde exibe gráficos, índices e mapas com os focos de doenças e informações sobre surtos na região e no país, ajudando no controle de epidemias. O software deve processar as informações dos prontuários no banco de dados para gerar essas informações em tempo real.

## Diagrama de classes



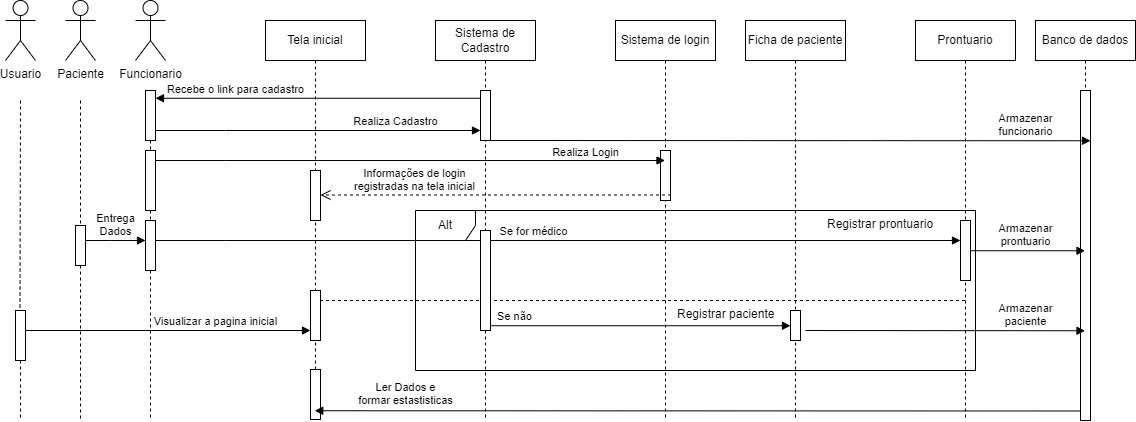
Nesse diagrama podemos ver relações de herança entre todas as classes do programa. A classe principal, e pai de outras 4 classes mais genéricas é a classe de conexão do banco de dados, MEDCONTROL\_DB, que possui uma única função que realiza a conexão do site com o banco de dados através do método connect, que é herdado nas próximas classes filhas.

As classes CADASTRO\_FUNCIONARIO, LOGIN\_FUNCIONARIO, REGISTRO\_PACIENTE E CADASTRO\_PRONTUÁRIO, são 4 classes filhas que herdam a conexão com o banco de dados. Elas possuem funcionalidades semelhantes e são usadas para abrir a conexão e criar as queries(SQL) com a funcionalidade de inserir ou checar os dados no banco. Elas estão ligadas com a função de validação das próximas classes.

As classes FUNCIONARIO, CONTROLE\_LOGIN, CONTROLE\_PACIENTE, CONTROLE\_PRONTUARIO por fim, são as classes filhas das 4 classes de cadastro/registro mencionadas anteriormente, e que visam criar os objetos (FUNCIONARIO, LOGIN, PACIENTE E PRONTUÁRIO) com as informações fornecidas pelo usuário, validando cada uma delas com seus métodos de validação e após isso as informações são inseridas no banco com o método herdado pelas classes de cadastro/registro.

Por último, a classe PacienteQuery, que servirá para manipular os cadastros dentro do sistema, com funções para edição dos pacientes e criação dos prontuários. A principal funcionalidade dessa classe está ligada com as queries do banco de dados, envolvendo o CRUD. Na prática, a classe será instanciada para gerenciar os cadastros já existentes, fazendo alteração na sua estrutura ou criando e editando prontuários a partir deles.

## Diagrama de sequência



Esse diagrama apresenta toda a sequência de ações que podem e devem ser executadas para o funcionamento do sistema.

Qualquer usuário pode acessar o sistema e ver a tela inicial que contém os dados dos gráficos, índices e mapas dos focos e epidemias de doenças em tempo real.

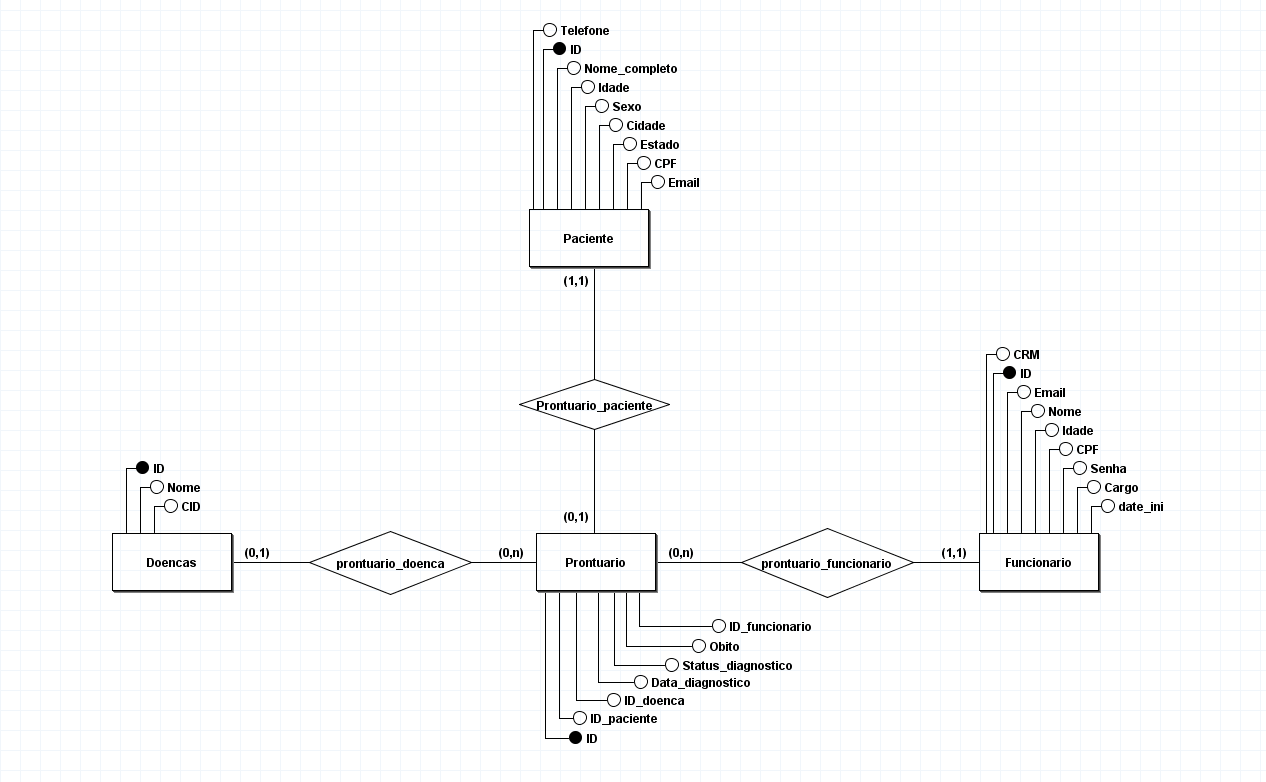
Quando estamos dentro da instituição hospitalar, os funcionários e médicos deverão realizar um cadastro e um login que ficarão armazenados no banco de dados previamente para assim poder manipular as informações dos pacientes e dos prontuários.

Logo o funcionário da recepção vai receber os dados do paciente e inserir no formulário de cadastro do programa, que ao ser processado pelo sistema, criará um cadastro do paciente no banco de dados.

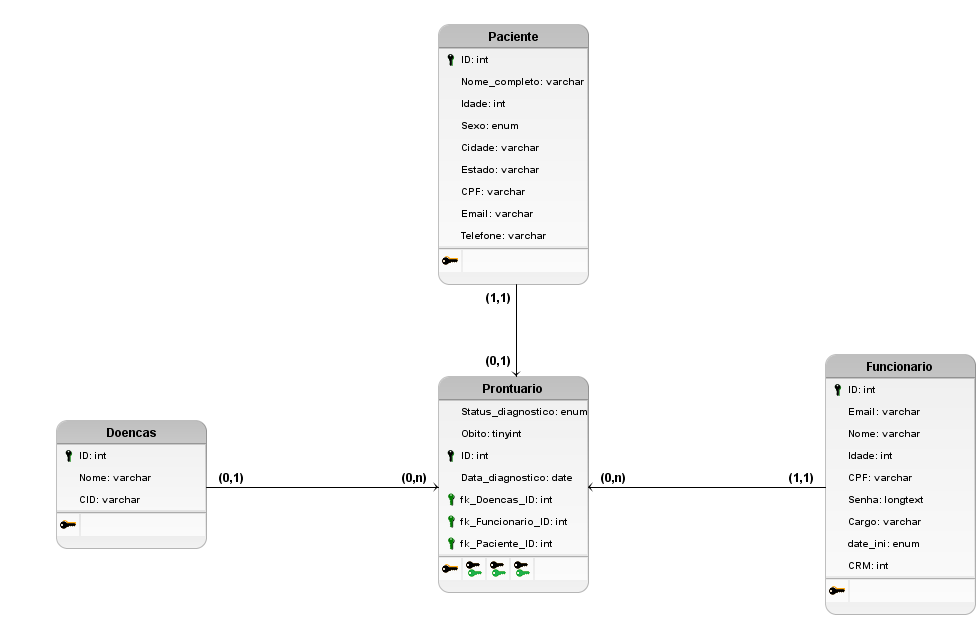
Esse paciente será encaminhado para a consulta com o medico e ao final, o medico abrira um prontuário no sistema contendo o cadastro do paciente e as informações adicionais com os resultados do exame e se caso houver, as doenças do paciente. O prontuário também será armazenado no banco de dados para uso futuro do próprio programa e de outras instituições medicas.

Com o armazenamento de todos esse prontuários no banco de dados, o sistema deve reunir todas essas informações, tanto do cadastro quanto em relação as doenças, a fim de processar esses dados e criar gráficos de gravidade das doenças, índices de focos e mortes registradas e mapas com os principais indicadores de epidemias e zonas de perigo nas diferentes áreas da cidade, estado ou do país.

## Modelo Conceitual



## Modelo lógico



## Modelo Físico

/\* medcontrol\_Logico: \*/

CREATE TABLE Doencas (

ID int PRIMARY KEY,

Nome varchar,

CID varchar

);

CREATE TABLE Funcionario (

ID int PRIMARY KEY,

Email varchar,

Nome varchar,

Idade int,

CPF varchar,

Senha longtext,

Cargo varchar,

date\_ini enum,

CRM int

);

CREATE TABLE Paciente (

ID int PRIMARY KEY,

Nome\_completo varchar,

Idade int,

Sexo enum,

Cidade varchar,

Estado varchar,

CPF varchar,

Email varchar,

Telefone varchar

);

CREATE TABLE Prontuario (

Status\_diagnostico enum,

Obito tinyint,

ID int PRIMARY KEY,

Data\_diagnostico date,

fk\_Doencas\_ID int,

fk\_Funcionario\_ID int,

fk\_Paciente\_ID int

);

ALTER TABLE Prontuario ADD CONSTRAINT FK\_Prontuario\_2

FOREIGN KEY (fk\_Doencas\_ID)

REFERENCES Doencas (ID)

ON DELETE SET NULL;

ALTER TABLE Prontuario ADD CONSTRAINT FK\_Prontuario\_3

FOREIGN KEY (fk\_Funcionario\_ID)

REFERENCES Funcionario (ID)

ON DELETE CASCADE;

ALTER TABLE Prontuario ADD CONSTRAINT FK\_Prontuario\_4

FOREIGN KEY (fk\_Paciente\_ID)

REFERENCES Paciente (ID)

ON DELETE CASCADE;

## Tecnologias Utilizadas

TRELLO para reuniões em grupo;

GITHUB para versionamento do software;

VSCODE para criação e edição dos códigos;

HTML, CSS e BOOTSTRAP para criação das interfaces do front-end;

JAVASCRIPT para a criação dos gráficos e dos mapas;

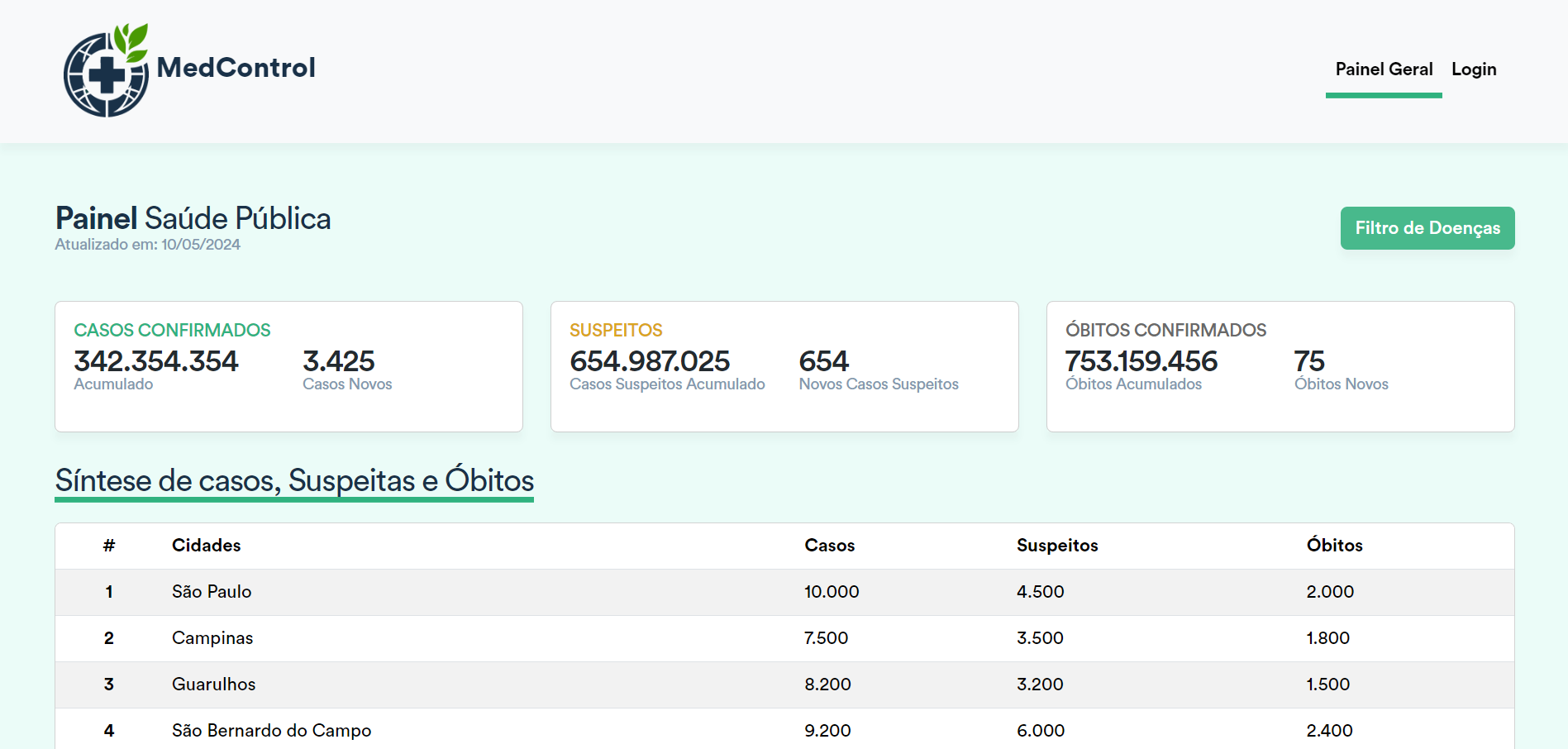
PHP para o funcionamento do back-end e a integração do programa com o banco de dados;

MYSQL para a criação do banco de dados, das queries e das funções;

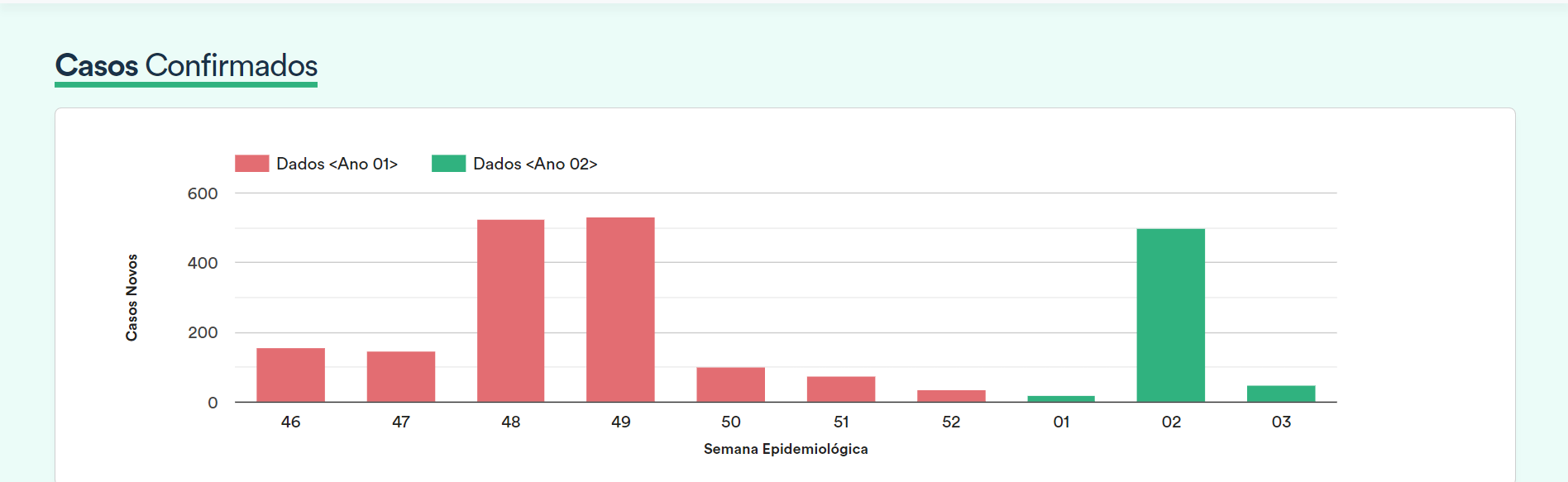
DRAW.IO para criação dos diagramas e artefatos de software;

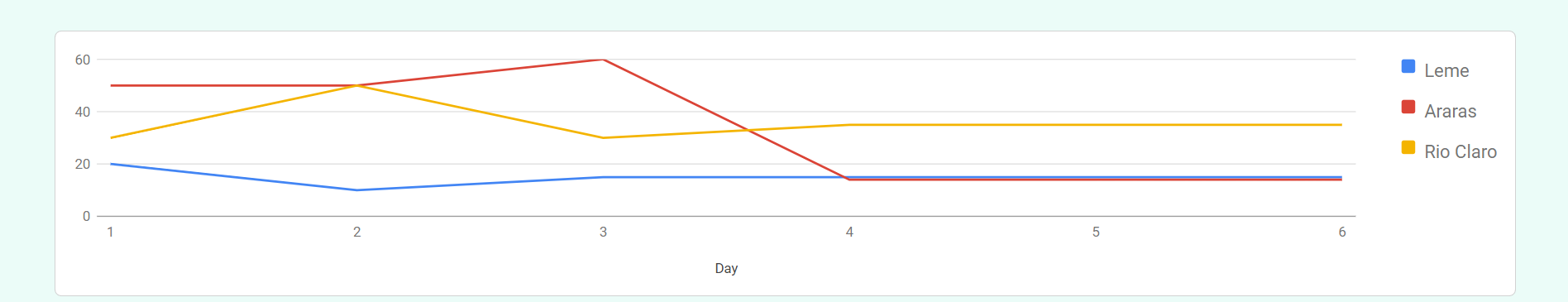
OFFICE para a documentação e engenharia do software.

## Interface do usuário



Página inicial com os principais índices, botões de login/registro e filtro dos casos





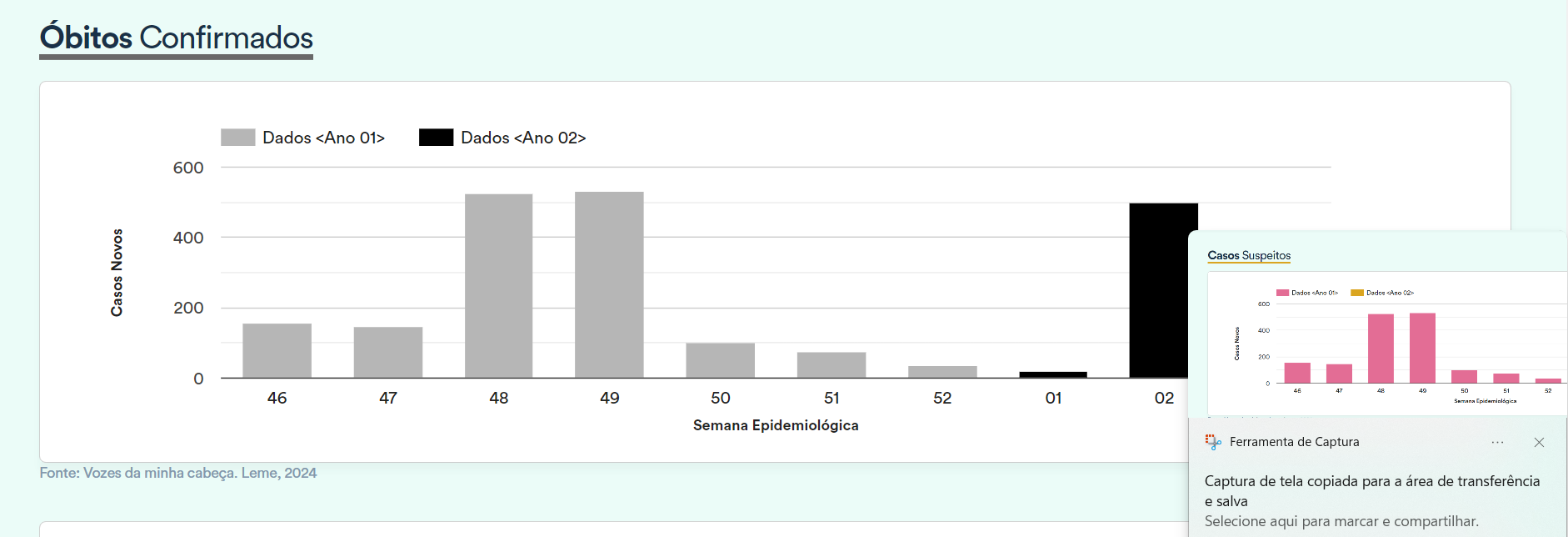
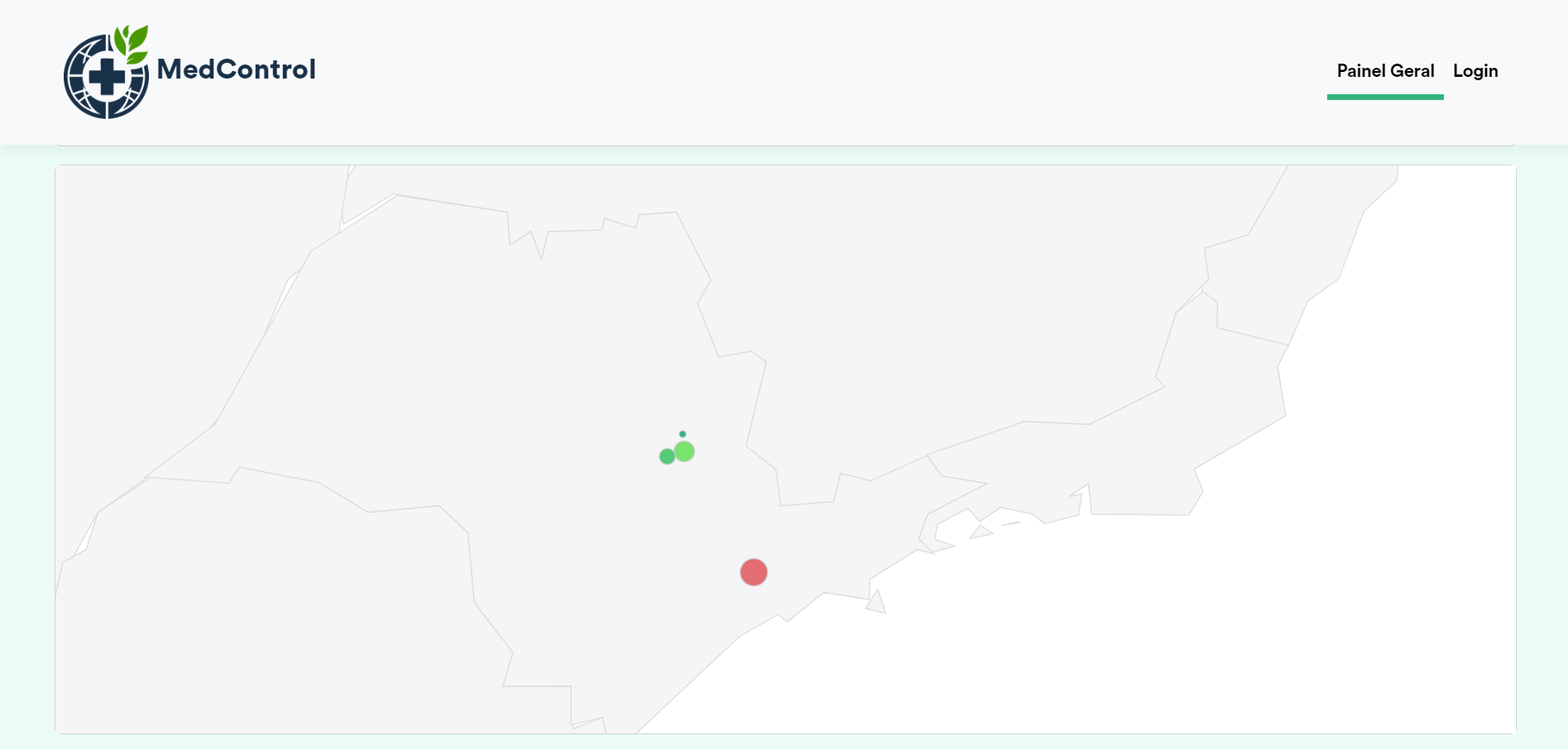


Gráfico com o acompanhamento dos casos e óbitos registrados



Mapa com os principais focos das doenças e controle das epidemias

## 3.8 Funcionalidades Implementadas

# Testes e Qualidade

4.1 Estratégia de Testes: Descrever a estratégia de testes adotada

4.2 Resultados dos Testes: Apresentar os resultados dos testes realizados

4.3 Garantia da Qualidade: Descrever as práticas adotadas para garantia da qualidade

4.4 Requisitos mínimos de hardware e software para o sistema

4.5 Contrato para desenvolvimento de software

# Considerações Finais

Conclusão: Concluir o trabalho e destacar aprendizados

Contribuições Individuais: Descrever as contribuições individuais de cada membro da equipe

Referências: Listar todas as fontes consultadas durante o trabalho