

**Faculdade de Informática e Administração Paulista**

**Projeto Vigia Saúde**

Adrian Satiro Sivilha

Jaci Teixeira Santos

Pedro Henrique Nobrega de Castro Paterno

Sabrina Faustinos do Prado

**Documentação Domain Driven Design**

São Paulo

2023

## Sumário

Índice.....	3
Diagrama de Classes - UML.....	4
Apresentação das telas (Páginas Web).....	5
Diagrama Entidade-Relacionamento.....	8
Endpoints .....	9
LoginResource:.....	9
MedicoResource: .....	9
TipoResource: .....	9
UnidadeResource:.....	10

## Índice

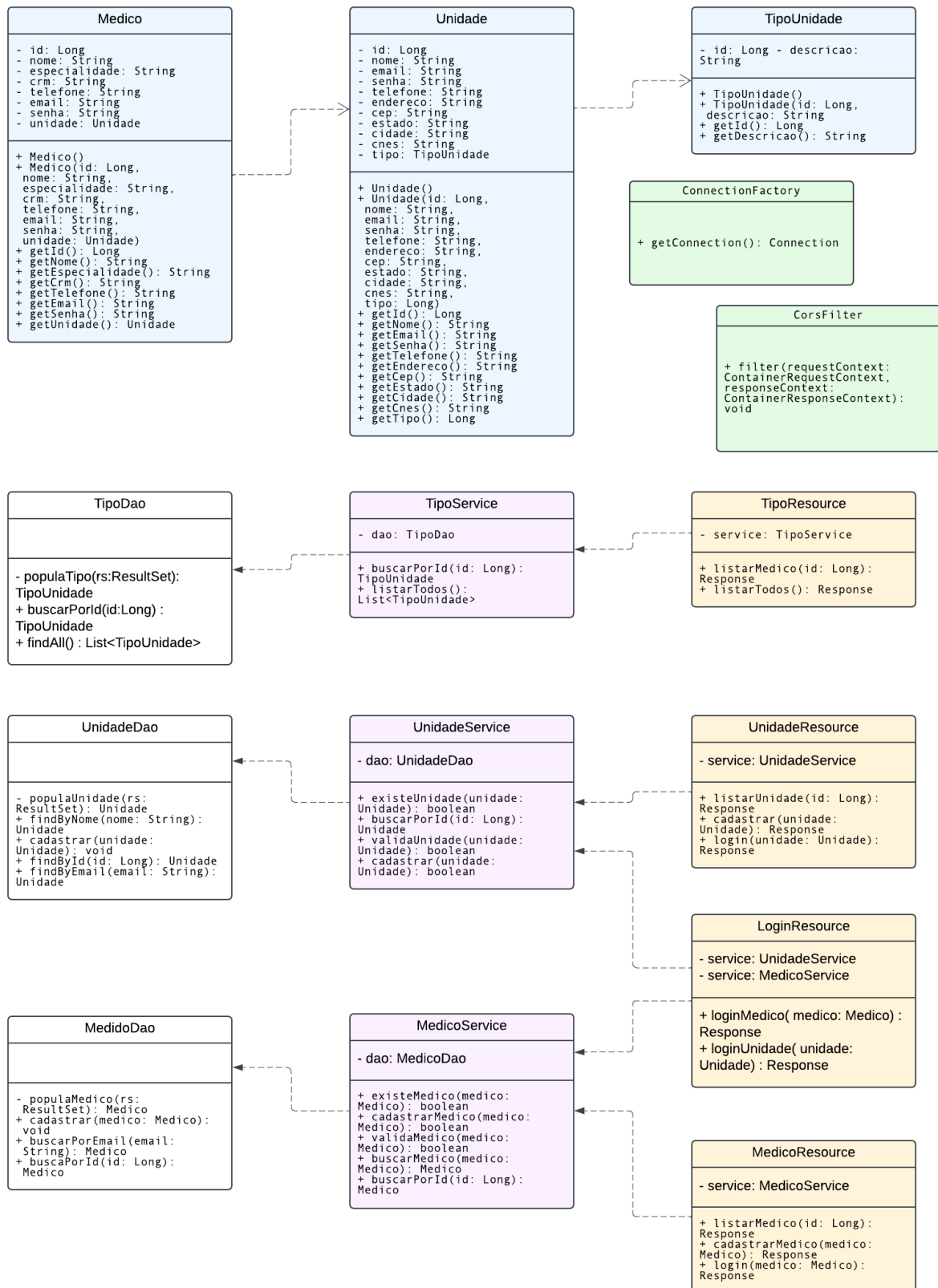
Estamos desenvolvendo uma aplicação dedicada ao monitoramento de casos de doenças tropicais muitas vezes negligenciadas no Brasil. Através desse sistema, pretendemos agir proativamente ao detectar um aumento significativo nos números de qualquer uma dessas doenças, gerando automaticamente alertas direcionados a hospitais, unidades de saúde, prefeituras e entidades públicas. O propósito é disponibilizar informações cruciais para que possam intensificar suas campanhas de prevenção e cuidados relacionados às doenças em questão.

Para realizar essa análise de maneira eficiente, utilizamos técnicas avançadas de treinamento de máquina, permitindo a leitura e identificação precisa das variações nos dados. Isso contribui para uma resposta mais ágil e direcionada diante das condições epidemiológicas observadas.

O objetivo central do nosso projeto é implementar um sistema de monitoramento baseado em inteligência artificial para detectar padrões anormais de incidência de doenças. Essa abordagem possibilitaria uma resposta mais rápida a possíveis surtos, contribuindo significativamente para o controle de epidemias.

Idealmente, buscamos integrar uma API de coleta de dados aos sistemas de hospitais, tanto públicos quanto privados. Essa API receberia em tempo real a entrada de incidência de casos de doenças, registrando as informações em bancos de dados. Os dados coletados, a princípio, incluiriam a doença, o local (estado) e a idade do paciente. É importante ressaltar que nossa API acessaria e coletaria APENAS esses dados, mantendo a solução alinhada com a Lei Geral de Proteção de Dados (LGPD). Dessa forma, não exigimos dos hospitais o envio de dados sensíveis dos pacientes, assegurando a conformidade com as normativas de privacidade.

## Diagrama de Classes - UML

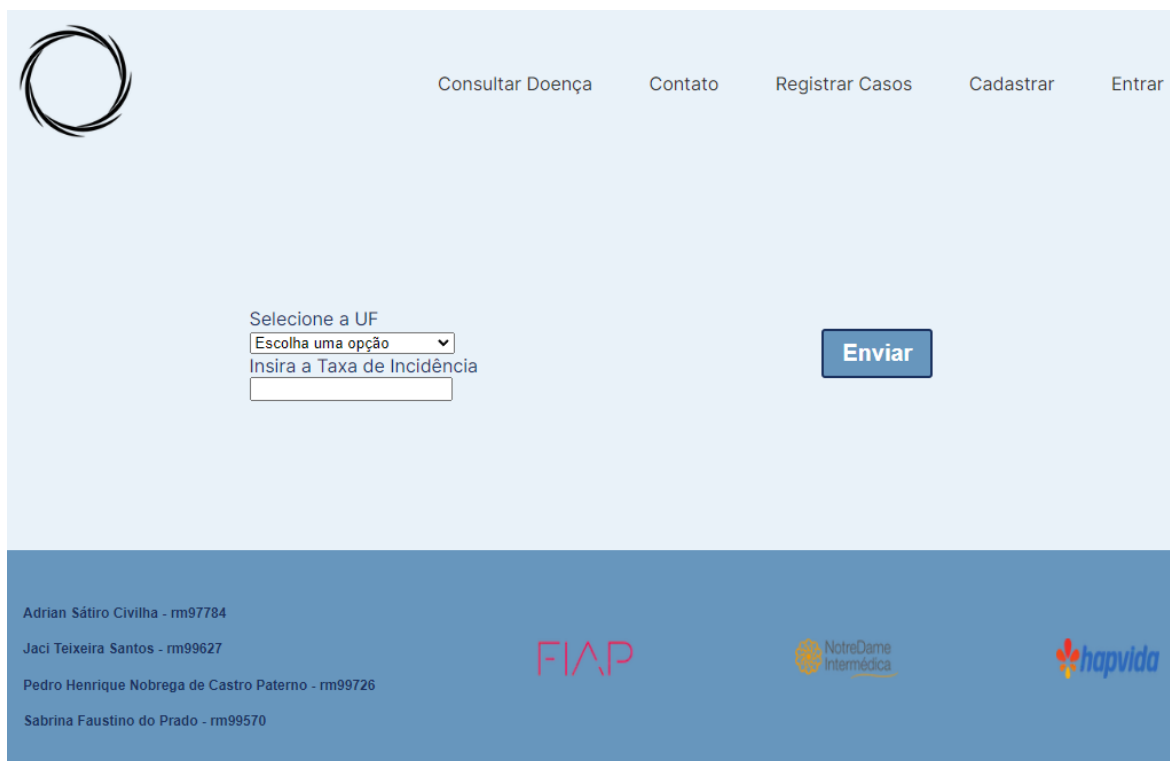


## Apresentação das telas (Páginas Web)


### Página inicial:



### Página para consultar doença:



### Página de contato:



Consultar DoençaContatoRegistrar CasosCadastrarEntrar

### Contate-nos

Nome:

E-mail:

Telefone:

Mensagem:

Adrian Sátiro Civilha - rm97784

Página de registro:



Consultar DoençaContatoRegistrar CasosCadastrarEntrar

### Proposta desta página

Segundo nossa ideia de projeto, que infelizmente não conseguimos implementar completamente em tempo hábil, esta seria a página onde médicos e administradores/sistemas automatizados de hospitais iriam dar entrada em novos de doenças (ex.: Tuberculose).

Dessa forma, aqui seria uma área acessível apenas para usuários logados, sendo que cada tipo de usuário (médico(a) ou estabelecimento de saúde) seria capaz de registrar os casos de maneira diferente.

Médicos poderiam registrar casos individuais, informando Doença, Data do Diagnóstico e UF.

Já estabelecimentos, poderiam enviar múltiplos casos de uma vez, informando os mesmos parâmetros, mas através de planilhas, ou outros tipos de arquivos adequados, que seriam capturadas no front-end, interpretadas no backend e os dados enviados ao nosso banco de dados que guarda e contabiliza os casos.

Essa abordagem exigiria, ainda, que houvesse conexão entre nosso banco de dados e nosso modelo de treinamento para que, a cada nova entrada de um caso de uma doença, o modelo pudesse ser adaptado àquela nova amostra, recalculando as taxas de incidência consideradas acima ou abaixo do normal.

Adrian Sátiro Civilha - rm97784  
Jaci Teixeira Santos - rm99627  
Pedro Henrique Nobrega de Castro Paterno - rm99726  
Sabrina Faustino do Prado - rm99570



Página de cadastro para médicos e para unidade de saúde:

### Faça o seu cadastro

Categoria  
**Estabelecimento de Saúde**

Nome:

E-mail:

Telefone:

Tipo:  
**Escolha um tipo**

CNES:

CEP

Estado

Cidade

Endereço

Senha:

Confirmar Senha:

**Enviar**

Já possui uma conta? [Entrar](#)

### Faça o seu cadastro

Categoria  
**Médico(a)**

Nome:

E-mail:

Telefone:

CRM:

Especialidade:

Unidade onde trabalha:  
**Escolha uma unidade**

Senha:

Confirmar Senha:

**Enviar**

Já possui uma conta? [Entrar](#)

Página login:



Consultar DoençaContatoRegistrar CasosCadastrarEntrar

### Bem-vindo(a) de volta

**Escolha uma categoria**

Não tem uma conta ainda?[Cadastrar-se](#)

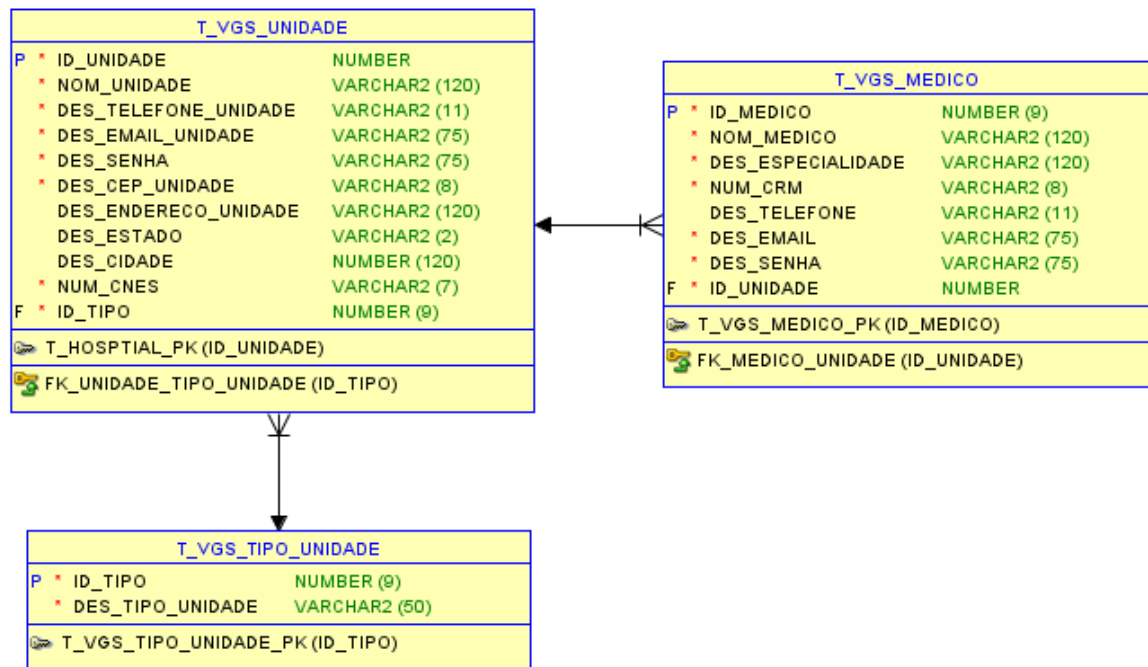
Adrian Sátiro Civilha - rm97784  
Jaci Teixeira Santos - rm99627  
Pedro Henrique Nobrega de Castro Paterno - rm99726  
Sabrina Faustino do Prado - rm99570

FIAP

Notre Dame Intermédica

hapvida

## Diagrama Entidade-Relacionamento





## Endpoints

### LoginResource:

Endpoint: `/login/medico`

- Método: POST
- Descrição: Autentica um médico.
- Possíveis Respostas:
- 200 OK: Autenticação bem-sucedida.
- 401 Unauthorized: Credenciais inválidas.

Endpoint: `/login/unidade`

- Método: POST
- Descrição: Autentica uma unidade de saúde.
- Possíveis Respostas:
- 200 OK: Autenticação bem-sucedida.
- 401 Unauthorized: Credenciais inválidas.

### MedicoResource:

Endpoint: `/medico/{id}`

- Método: GET
- Descrição: Lista informações de um médico por ID.
- Possíveis Respostas:
- 200 OK: Retorna informações do médico.
- 404 Not Found: Médico não encontrado.

Endpoint: `/medico/cadastro`

- Método: POST
- Descrição: Cadastra um novo médico.
- Possíveis Respostas:
- 200 OK: Médico cadastrado com sucesso.
- 400 Bad Request: Requisição inválida (por exemplo, dados ausentes).
- 409 Conflict: Email já em uso.

### TipoResource:

Endpoint: `/tipo`

- Método: GET
- Descrição: Lista todos os tipos de unidades.
- Possíveis Respostas:
- 200 OK: Retorna a lista de tipos.

- 404 Not Found: Nenhum tipo encontrado.

Endpoint: /tipo/{id}

- Método: GET
- Descrição: Lista informações de um tipo por ID.
- Possíveis Respostas:
- 200 OK: Retorna informações do tipo.
- 404 Not Found: Tipo não encontrado.

UnidadeResource:

Endpoint: /unidade

- Método: GET
- Descrição: Lista todas as unidades.
- Possíveis Respostas:
- 200 OK: Retorna a lista de unidades.
- 404 Not Found: Nenhuma unidade encontrada.

Endpoint: /unidade/{id}

- Método: GET
- Descrição: Lista informações de uma unidade por ID.
- Possíveis Respostas:
- 200 OK: Retorna informações da unidade.
- 404 Not Found: Unidade não encontrada.

Endpoint: /unidade/cadastro

- Método: POST
- Descrição: Cadastra uma nova unidade.
- Possíveis Respostas:
- 200 OK: Unidade cadastrada com sucesso.
- 400 Bad Request: Requisição inválida (por exemplo, dados ausentes).
- 409 Conflict: Email já em uso.

Link canal onde o video do funcionamento da Aplicação foi hospedado:

[https://youtube.com/@tutorialscraft9531?si=iW\\_2umHEEa9J1iMP](https://youtube.com/@tutorialscraft9531?si=iW_2umHEEa9J1iMP)