

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

Fábio Fernandes Junior

**Dengue Monitoring**

**Vídeo de apresentação do Projeto Integrador**

<https://youtu.be/rOdwN4VahEU>

Dourado- SP  
2025

# **UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

## **Dengue Monitoring**

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Bacharelado em Tecnologia da Informação da Universidade Virtual do Estado de São Paulo (UNIVESP).

Dourado - SP  
2025

FERNANDES, Junior, Fábio. Dengue Monitoring. Relatório Técnico-Científico. Projeto Integrador para o curso de Bacharelado em Tecnologia da Informação– **Universidade Virtual do Estado de São Paulo**. Tutor: Luana Barbosa da Silva. Polo Dourado, 2025.

## **RESUMO**

O presente relatório refere-se ao desenvolvimento de um aplicativo para o monitoramento de pacientes suspeitos de dengue, com foco na coleta, armazenamento, manipulação e recuperação de dados de forma eficiente e segura. O projeto foi estruturado em torno da utilização de um banco de dados MySQL, elaborado por meio do MySQL Workbench, visando garantir a organização e o fácil acesso às informações dos pacientes. A interface do usuário foi construída com HTML e CSS, oferecendo uma experiência intuitiva e responsiva. A prática de controle de versão foi realizada por meio do GitHub, facilitando a colaboração entre os membros da equipe e o gerenciamento do código-fonte. O armazenamento foi realizado na nuvem AWS, garantindo escalabilidade e segurança dos dados. Para assegurar a qualidade do desenvolvimento, foram adotadas práticas de integração contínua, testes automatizados e análise de dados, com o objetivo de monitorar a evolução dos pacientes e gerar relatórios úteis para os profissionais de saúde. Este projeto visou melhorar o processo de gestão e acompanhamento de casos suspeitos de dengue, proporcionando uma solução eficiente e acessível..

**PALAVRAS-CHAVE:** Banco de dados; monitoramento; pacientes; Dengue.

## **LISTAS DE TABELAS**

**TABELA 1** - ENTREVISTA COM PROFISSIONAL DE SAÚDE.....21

**TABELA 2** - DIAGRAMA RELACIONAL.....22

## SUMÁRIO

### SUMÁRIO

INTRODUÇÃO.....	2
1 DESENVOLVIMENTO.....	3
1.1 Objetivos.....	3
1.2 Justificativa e delimitação do problema.....	5
1.3 Fundamentação teórica.....	7
1.4 Aplicação das disciplinas estudadas no Projeto Integrador.....	9
1.5 Metodologia.....	11
2 RESULTADOS: SOLUÇÃO FINAL.....	16
3 CONSIDERAÇÕES FINAIS.....	19
REFERÊNCIAS.....	20
ANEXOS.....	21

## **1 INTRODUÇÃO**

O desenvolvimento de um aplicativo para o monitoramento de pacientes suspeitos de dengue buscou otimizar o processo de coleta, armazenamento, manipulação e recuperação de dados de forma eficiente e segura. A escolha do tema específico reflete a necessidade de aplicar tecnologias modernas para o gerenciamento de dados de saúde, utilizando um banco de dados MySQL para o armazenamento de informações e um front-end responsivo em CSS para a interação do usuário. Além disso, o projeto contou com práticas de controle de versão no GitHub, armazenamento em nuvem via AWS, integração contínua e a aplicação de testes e análise de dados, garantindo eficiência e segurança durante o desenvolvimento e operação do aplicativo.

## **2 DESENVOLVIMENTO**

### **2.1 OBJETIVOS**

O objetivo geral:

Desenvolvimento de um aplicativo móvel integrado com um banco de dados para o monitoramento de pacientes com suspeita de dengue, que permita o armazenamento, manipulação e recuperação eficiente e segura dos dados relacionados aos pacientes, facilitando o acompanhamento médico e a tomada de decisões rápidas e precisas.

Objetivos específicos:

Elaboração do Banco de Dados no MySQL Workbench:

- Criação de um banco de dados relacional no MySQL Workbench para armazenar as informações dos pacientes, como dados pessoais, sintomas, exames e evolução do quadro clínico.
- Garantia de integridade e eficiência no armazenamento e recuperação dos dados, aplicando boas práticas de modelagem de banco de dados, como normalização e criação de índices.

Desenvolvimento do Front-end em CSS:

- Criação de uma interface de usuário responsiva e amigável, utilizando CSS para garantir que o aplicativo seja acessível em dispositivos móveis e desktops.
- Desenvolvimento de telas de cadastro de pacientes, monitoramento de sintomas e

geração de relatórios.

#### Controle de Versão via GitHub:

- Utilização do GitHub para gerenciamento de versões do código-fonte, permitindo que todos os membros do grupo possam colaborar de maneira eficiente, mantendo o histórico de alterações e facilitando a reversão de mudanças, quando necessário.

#### Armazenamento em Nuvem com AWS:

- Utilização dos serviços de armazenamento em nuvem da AWS para hospedar o banco de dados e os arquivos do aplicativo, garantindo escalabilidade e segurança no acesso às informações.
- Implementação da infraestrutura necessária para garantir a alta disponibilidade e a proteção dos dados armazenados.

#### Integração Contínua (CI) e Testes:

- Implementação de um pipeline de integração contínua que permita realizar testes automatizados do aplicativo e do banco de dados durante o processo de desenvolvimento, garantindo que novas funcionalidades não comprometam o funcionamento do sistema.
- Realização de testes de desempenho, segurança e usabilidade para validar a eficiência do aplicativo e garantir uma boa experiência ao usuário.

#### Análise de Dados:

- Criação de funcionalidades para a geração de relatórios e análise dos dados coletados, como gráficos e resumos, para facilitar o acompanhamento dos pacientes pelos profissionais de saúde.



## 2.2 JUSTIFICATIVA E DELIMITAÇÃO DO PROBLEMA

A crescente incidência de casos de dengue, especialmente em regiões tropicais e subtropicais, destaca a necessidade de um sistema eficiente para o monitoramento e gestão de pacientes suspeitos dessa doença. A falta de um processo automatizado e integrado para registrar, armazenar e acessar dados de pacientes dificulta a tomada de decisões rápidas e assertivas, o que pode resultar em falhas no controle e na prevenção de surtos.

Este projeto tem como objetivo criar um aplicativo que inclua um banco de dados robusto para o monitoramento de pacientes suspeitos de dengue. A utilização do **MySQL Workbench** para a elaboração do banco de dados garantirá uma estrutura sólida e escalável, enquanto o **front-end em CSS** proporcionará uma interface amigável e responsiva para os profissionais de saúde. O uso do **GitHub** como sistema de controle de versão permitirá a colaboração eficiente entre os membros da equipe, e o armazenamento em nuvem na **AWS** assegurará que os dados sejam acessíveis de qualquer lugar com segurança.

Além disso, a **integração contínua** e a **realização de testes automatizados** garantirão a estabilidade e a qualidade do aplicativo, promovendo uma experiência segura e confiável para os usuários. A análise de dados coletados contribuirá para a geração de relatórios e insights, permitindo que as autoridades de saúde possam monitorar a evolução dos casos e realizar ações preventivas mais eficazes.

Portanto, a implementação deste sistema tem um impacto direto na melhoria da gestão de pacientes suspeitos de dengue, contribuindo para a rapidez no diagnóstico e tratamento, além de fornecer dados estratégicos para políticas públicas de saúde.

### **Delimitação do Problema**

O sistema proposto será restrito ao **monitoramento de pacientes suspeitos de dengue**, com foco nas seguintes funcionalidades:

**Cadastro de Pacientes:** O aplicativo permite o cadastro completo de pacientes, com informações como nome, idade, sexo, sintomas e histórico de viagens, essenciais para o

diagnóstico.

**Registro de Sintomas e Exames:** O sistema é capaz de registrar os sintomas e exames realizados, associando-os a cada paciente de forma clara e organizada.

**Relatórios e Análises:** O aplicativo gera relatórios com informações consolidadas sobre os casos suspeitos de dengue, permitindo a análise estatística e a visualização de padrões relacionados aos surtos da doença.

**Segurança dos Dados:** implementada a segurança necessária para proteger dados sensíveis dos pacientes, seguindo as normas de privacidade de dados, como a **Lei Geral de Proteção de Dados (LGPD)** no Brasil.

**Integração com Armazenamento em Nuvem (AWS):** O sistema foi hospedado na AWS (Amazon Web Services), garantindo escalabilidade, segurança e fácil acesso aos dados.

**Limitação Geográfica e Temporária:** Inicialmente, o aplicativo será implementado para uso em uma área geográfica específica, com a possibilidade de expansão conforme a evolução do projeto e a adaptação da infraestrutura.

**Foco em Usabilidade:** O aplicativo foi desenvolvido com uma interface simples e intuitiva, para garantir que os profissionais de saúde possam utilizar o sistema com eficiência, sem a necessidade de treinamento extensivo.

O sistema não abordará, neste primeiro momento, a automação do diagnóstico de dengue ou integração com outros sistemas de saúde externos. O foco está na melhoria do processo de coleta e armazenamento de dados, garantindo acesso rápido e seguro à informação para os profissionais de saúde.

## 2.3 FUNDAMENTAÇÃO TEÓRICA

### **Banco de Dados MySQL para Armazenamento de Informações**

O **MySQL** é um Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) amplamente utilizado devido à sua estabilidade, escalabilidade e desempenho. É ideal para aplicações que exigem integridade e confiabilidade no tratamento de grandes volumes de dados, como sistemas de saúde. De acordo com Elmasri e Navathe (2011), bancos de dados relacionais oferecem mecanismos robustos para o controle de integridade, normalização e recuperação eficiente de dados estruturados, como o prontuário clínico de pacientes.

Como salientam Date (2004) e Silberschatz et al. (2013), a modelagem adequada de um banco relacional, por meio de normalização e uso eficiente de chaves, é fundamental para garantir a consistência dos dados e a redução de redundâncias.

### **Front-End Responsivo com HTML e CSS**

A camada de apresentação do sistema é fundamental para a experiência do usuário. O uso de **CSS3** e **HTML5** permite a criação de interfaces responsivas e acessíveis. Nielsen (1993), referência em usabilidade, defende que uma interface eficiente deve ser intuitiva, reduzir a carga cognitiva do usuário e garantir acessibilidade em diversos dispositivos.

Complementarmente, Wroblewski (2011) reforça o conceito de "mobile first", destacando que a adaptação a telas menores deve ser prioridade no design de sistemas modernos. O uso de **media queries** permite adaptar o layout dinamicamente, o que é essencial para contextos como o da saúde pública, onde muitos profissionais utilizam dispositivos móveis.

### **Controle de Versão com Git e GitHub**

Ferramentas de controle de versão são indispensáveis no desenvolvimento colaborativo. O **Git**, aliado ao **GitHub**, facilita o gerenciamento do histórico de alterações, integração entre membros da equipe e automação de testes. Segundo Chacon e Straub (2014), o Git permite que diferentes desenvolvedores trabalhem simultaneamente sem sobrescrever o trabalho dos colegas, o que aumenta a produtividade e segurança do projeto.

### **Armazenamento em Nuvem com AWS**

A adoção de **computação em nuvem**, como os serviços oferecidos pela **Amazon Web Services (AWS)**, proporciona alta disponibilidade, escalabilidade e redução de custos operacionais. Linthicum (2017) destaca que a nuvem permite o provisionamento dinâmico de recursos, o que é essencial para aplicações com variações na demanda, como no caso de monitoramento de surtos de dengue.

### **Integração Contínua e Testes Automatizados**

A **Integração Contínua (CI)**, combinada com testes automatizados, garante que cada nova versão do software seja testada automaticamente antes de ser integrada ao sistema. Segundo Fowler (2006), essa prática minimiza o risco de erros não detectados e aumenta a confiabilidade do produto final. Beck (2002), um dos idealizadores da metodologia ágil, também defende a automação como pilar da entrega contínua de software de qualidade.

### **Análise de Dados e Suporte à Decisão**

A análise dos dados coletados permite identificar padrões e prever surtos. Davenport e Harris (2007) afirmam que o uso de **Business Intelligence (BI)** e **Data Analytics** transforma dados brutos em informações estratégicas, facilitando a tomada de decisões na gestão da saúde. Graham (2016) reforça que sistemas orientados por dados têm um impacto significativo em áreas como epidemiologia e saúde pública.

## 2.4 APLICAÇÃO DAS DISCIPLINAS ESTUDADAS NO PROJETO INTEGRADOR

### **Banco de Dados**

**Contribuições:** Modelagem do banco de dados relacional (MySQL) com tabelas para pacientes, endereço, sintomas, resultado de sorologia (positivo/negativo). Criação de consultas SQL otimizadas para extração de informações relevantes. Garantia de integridade dos dados (chaves primárias, estrangeiras, normalização).

### **Programação de Computadores**

**Contribuições:** Desenvolvimento da lógica de funcionamento do sistema (back-end e scripts de controle). Implementação da comunicação entre o banco de dados e a interface do usuário usando Python. Tratamento de erros e testes de funcionalidades essenciais.

### **Fundamentos de Internet e Web**

**Contribuições:** Estruturação da arquitetura cliente-servidor do sistema. Compreensão do funcionamento de protocolos como HTTP/HTTPS, que garantem a comunicação segura entre usuário e servidor.

### **Algoritmos e Programação de Computadores**

**Contribuições:** Criação de algoritmos eficientes para análise dos dados (ex: identificar padrões nos sintomas). Otimização do desempenho das funções do aplicativo. Aplicação de lógica condicional.

### **Infraestrutura para Sistemas de Software**

**Contribuições:** Utilização de ambientes em nuvem (como AWS) para hospedar o sistema e garantir escalabilidade e versionamento com GitHub.

## **Segurança da Informação**

**Contribuições:** Criptografia de dados sensíveis (como dados pessoais dos pacientes). Autenticação e controle de acesso.

## **Integração entre as disciplinas**

A força do projeto está na **integração entre as áreas**: a segurança protege os dados do banco que foi modelado pela disciplina de Banco de Dados; a interface construída com fundamentos de Web consome os dados através da lógica desenvolvida com base em Programação e Algoritmos, enquanto a Infraestrutura garante que tudo funcione de forma contínua e escalável.

## METODOLOGIA

A metodologia adotada foi estruturada conforme os passos indicados pela UNIVESP. A seguir, descrevemos como o grupo seguiu essa metodologia e os resultados alcançados em cada etapa do processo.

### 2.1. Ouvir

O primeiro passo da metodologia foi entender o problema e ouvir as necessidades dos usuários potenciais, que no caso são os profissionais de saúde, especificamente os Agentes Comunitários de Saúde. Para isso, o grupo realizou as seguintes atividades:

- **Entrevistas com profissionais de saúde:** Entendemos as dificuldades enfrentadas no acompanhamento de pacientes suspeitos de dengue, o que inclui a gestão manual de dados, o risco de erros humanos e a necessidade de informações rápidas e seguras.
- **Análise de sistemas existentes:** Estudamos aplicativos e sistemas de monitoramento de saúde para identificar boas práticas, funcionalidades essenciais e pontos de melhoria.
- **Levantamento de requisitos:** Definimos as principais funcionalidades do aplicativo, como o cadastro de pacientes, monitoramento de sintomas, acompanhamento de exames e tratamentos, e a necessidade de relatórios gerenciais para as equipes de saúde.

#### Resultado da Etapa "Ouvir":

- Definição clara das funcionalidades necessárias para o aplicativo.
- Identificação das principais dificuldades enfrentadas pelos profissionais de saúde.
- Definição dos requisitos essenciais para a solução: segurança, eficiência e acessibilidade.

### 2.2. Criar

Após ouvir os profissionais de saúde e levantar os requisitos necessários, o próximo passo foi a fase de **criar**, onde o grupo projetou as soluções para o problema identificado.

### 2. 2. 1. Criação do Banco de Dados

O banco de dados foi projetado no **MySQL Workbench** para garantir uma estrutura relacional eficiente para o armazenamento das informações dos pacientes. As principais tabelas criadas incluem:

**Tabela de Pacientes:** Armazena os dados pessoais dos pacientes (nome, idade, endereço, etc.).

- **Tabela de Sintomas:** Registra os sintomas apresentados pelos pacientes ao longo do tempo.
- **Tabela de Histórico Clínico:** Contém informações sobre diagnósticos, exames realizados e tratamentos prescritos.
- **Tabela de Monitoramento:** Registra a evolução do quadro clínico dos pacientes.
- O modelo de banco de dados foi normalizado para garantir a integridade e eficiência das consultas e operações.

### Diagrama Relacional

#### 1. Tabela: Pacientes

- **id\_paciente** (PK) - Identificador único do paciente.
- **nome** - Nome do paciente.
- **idade** - Idade do paciente.
- **endereco** - Endereço do paciente.
- **telefone** - Número de telefone do paciente.
- **data\_nascimento** - Data de nascimento do paciente.
- **sexo** - Sexo do paciente.

#### 2. Tabela: Sintomas

- **id\_sintoma** (PK) - Identificador único do sintoma.
- **descricao** - Descrição do sintoma (ex.: febre, dor de cabeça, etc.).
- **data\_ocorrencia** - Data em que o sintoma foi reportado.



### 3. Tabela: Histórico Clínico

- **id\_historico** (PK) - Identificador único do histórico clínico.
- **id\_paciente** (FK) - Relacionamento com a tabela Pacientes.
- **diagnostico** - Diagnóstico dado ao paciente.
- **exames\_realizados** - Lista de exames realizados.
- **data\_diagnostico** - Data do diagnóstico.

### 4. Tabela: Monitoramento

- **id\_monitoramento** (PK) - Identificador único do monitoramento.
- **id\_paciente** (FK) - Relacionamento com a tabela Pacientes.
- **id\_sintoma** (FK) - Relacionamento com a tabela Sintomas.
- **intensidade** - Intensidade do sintoma (leve, moderado, grave).
- **observacao** - Observações adicionais feitas pelos profissionais de saúde.
- **data\_monitoramento** - Data do acompanhamento.

### Relacionamentos

- **Pacientes** tem um relacionamento 1:N com **Histórico Clínico** (um paciente pode ter múltiplos históricos).
- **Pacientes** tem um relacionamento 1:N com **Monitoramento** (um paciente pode ser monitorado várias vezes).
- **Sintomas** tem um relacionamento N:M com **Monitoramento** (um sintoma pode ser monitorado por vários pacientes e um paciente pode ter vários sintomas monitorados). Esse relacionamento é mediado pela tabela **Monitoramento**.

#### 2.2.2. Desenvolvimento do Front-end

A interface do usuário foi criada utilizando **HTML, CSS e JavaScript**, com o objetivo de ser simples, intuitiva e responsiva para garantir o acesso ao sistema em diversos dispositivos, como desktops e smartphones.

Principais telas desenvolvidas:

- **Tela de Cadastro de Pacientes:** Permite que os profissionais de saúde insiram dados do paciente.
- **Tela de Monitoramento de Sintomas:** Exibe e permite a atualização dos sintomas dos pacientes ao longo do tempo.
- **Tela de Histórico Clínico:** Exibe informações detalhadas sobre os exames realizados e tratamentos indicados.
- **Tela de Relatórios:** Gera relatórios sobre a evolução do quadro clínico dos pacientes, facilitando a tomada de decisão pelos profissionais de saúde.

### 2.2.3. Controle de Versão com GitHub

Para garantir o controle de versão do código-fonte, foi utilizado o **GitHub**. O código foi dividido em branches para facilitar o trabalho colaborativo, com cada membro da equipe responsável por diferentes partes do desenvolvimento, como o banco de dados, o front-end e a integração com a AWS.

### Resultado da Etapa "Criar":

- Conclusão da modelagem do banco de dados e do front-end básico.
- Estabelecimento do fluxo de trabalho no GitHub para garantir a colaboração eficiente da equipe.
- Desenvolvimento das telas principais do aplicativo.

## 2.3. Implementar

Na fase de **implementar**, o grupo começou a colocar em prática as soluções desenvolvidas. Isso envolveu a implementação real do banco de dados no MySQL, a configuração do ambiente de desenvolvimento, a integração do front-end com o banco de dados, e a hospedagem na nuvem AWS.

### 2.3.1. Implementação do Banco de Dados

O banco de dados foi criado no MySQL e conectado ao front-end do aplicativo para garantir que os dados dos pacientes possam ser armazenados e recuperados de forma eficiente. A

implementação incluiu:

- **Criação de tabelas:** Implementação das tabelas de pacientes, sintomas, histórico clínico e monitoramento.
- **Criação de Stored Procedures e Triggers:** Para automatizar certos processos no banco de dados, como atualizações de dados e verificações de consistência.

### 2.3.2. Armazenamento em Nuvem com AWS

O banco de dados foi configurado para ser hospedado no **Amazon RDS**, que fornece escalabilidade e segurança. Além disso, o front-end do aplicativo foi armazenado no **Amazon S3**, garantindo que a aplicação possa ser acessada de forma rápida e eficiente de qualquer local.

### 2.3.3. Integração Contínua e Testes

Foi configurado um pipeline de **Integração Contínua (CI)** utilizando o **GitHub Actions**. Isso permite que o código seja automaticamente testado a cada commit, garantindo que os erros sejam detectados precocemente.

### Resultado da Etapa "Implementar":

- Banco de dados configurado e integrado com o front-end.
- Implementação do sistema de armazenamento em nuvem na AWS.
- Configuração da integração contínua e testes automatizados para garantir a qualidade do código.
- O banco de dados está funcional e devidamente integrado ao front-end.
- O aplicativo está em funcionamento básico, permitindo o cadastro de pacientes e a entrada de dados de sintomas.
- O sistema de controle de versão no GitHub está em funcionamento, permitindo colaboração eficaz entre os membros da equipe.
- O armazenamento em nuvem foi implementado, e o sistema está acessível pela AWS.

### 3 RESULTADOS: SOLUÇÃO FINAL

O projeto propõe a criação de um **aplicativo web responsivo** para o **monitoramento de pacientes suspeitos de dengue**, com foco na eficiência, segurança da informação e integração entre tecnologias modernas.

#### Componentes da Solução:

##### Interface do Usuário (Front-End):

Desenvolvida em HTML5, CSS3 e JavaScript. Design responsivo, acessível por dispositivos móveis e desktops. Tela de cadastro e acompanhamento de pacientes, sintomas e notificações.

PACIENTES
NOME
ENDEREÇO
TELEFONE
CPF

Fonte : elaborada pelos autores

##### Lógica do Sistema (Back-End):

Utilização de linguagem de programação como Python (Flask ou Django) ou PHP. Comunicação segura com o banco de dados MySQL via APIs RESTful. Geração de alertas com base na gravidade dos sintomas registrados.

Fonte : elaborada pelos autore

SINTOMAS
DOR NO CORPO
FEBRE
DOR DE CABEÇA
NAUSEAS
FALTA DE APETITE
DIARRÉIA
MANCHAS AVERMELHADAS
SANGRAMENTO



RISCO
DIABETES
HIPERTENSÃO ARTERIAL
OBESIDADE
MAIOR DE 60 ANOS



SOROLOGIA
POSITIVO
NEGATIVO

### **Banco de Dados (MySQL):**

Estrutura relacional com tabelas normalizadas. Armazenamento de dados dos pacientes, sintomas, grupos de risco, e resultados de sorologia.

### **Infraestrutura e Hospedagem:**

Utilização de serviços AWS (como EC2, S3 ou RDS) para hospedagem e escalabilidade do sistema. Armazenamento em nuvem para facilitar acesso e backups automatizados. Armazenar estes dados, pode futuramente garantir o levantamento do histórico de sintomas do paciente, bem como informar se já passou por esses quadro no passado e como foi a reação de seu organismo.

### **Controle de Versão e Integração Contínua:**

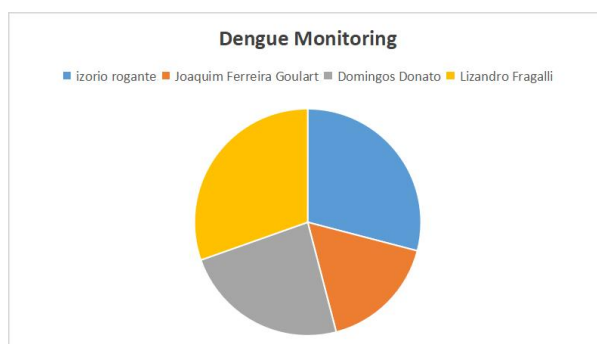
Histórico de alterações e colaboração entre desenvolvedores.

### **Segurança da Informação:**

Criptografia de dados sensíveis no banco (ex: AES, bcrypt). Autenticação de usuários com níveis de acesso (paciente, profissional de saúde, administrador). Proteção contra vulnerabilidades web (SQL Injection, XSS, CSRF).

### **Análise e Visualização de Dados:**

Geração de gráficos e relatórios com base nos dados coletados. Painel de controle com indicadores (ex: quantidade de casos por região, gravidade dos sintomas). Possibilidade de integrar a sistemas de saúde pública no futuro.



Fonte : elaborada pelos autores

Gráficos como este, sinalizam ao profissional visualizar as áreas com maior numero de casos, intensificando o trabalho na busca de focos eliminando criadouros e diminuindo a transmissão e o aumento no numero de pacientes positivos de Dengue.

#### Esclarecimento:

Durante todo processo de levantamento de dados, idealização, prototipação e testes, os profissionais de saúde (Agentes Comunitários) participaram ativamente, dando sugestões e mostrando muita empatia com o projeto. A pedido destes profissionais, este relatório não contém imagens e dados reais contidos no aplicativo, o objetivo é preservar dados de saúde de pacientes, já que estes dados são confidenciais e mantidos sobre sigilo. A base de todo trabalho é o respeito e a ordem, portanto toda e qualquer imagem contida neste relatório será fictícia, apenas para ilustrar resultados.

#### Feedback:

- O banco de dados está funcional e devidamente integrado ao front-end.
- O aplicativo está em funcionamento básico, permitindo o cadastro de pacientes e a entrada de dados de sintomas.
- O sistema de controle de versão no GitHub está em funcionamento, permitindo colaboração eficaz entre os membros da equipe.
- O armazenamento em nuvem foi implementado, e o sistema está acessível pela AWS.

## Considerações Finais

O desenvolvimento deste projeto integrador demonstrou como a aplicação de conhecimentos adquiridos em diferentes disciplinas permite criar uma solução tecnológica eficaz, segura e útil para a área da saúde pública. A proposta do aplicativo vai além do registro de informações: ele visa **antecipar riscos, facilitar o trabalho de profissionais de saúde e contribuir para o controle da dengue.**

A integração entre Banco de Dados, Programação, Segurança, Infraestrutura e Web permitiu o desenvolvimento de um sistema completo, que pode ser expandido com novas funcionalidades, como geolocalização dos casos, alertas automáticos para autoridades sanitárias e envio de mensagens para os pacientes.

Além disso, a adoção de boas práticas de desenvolvimento, como controle de versão, testes automatizados e uso da nuvem, garante a **escalabilidade, confiabilidade e manutenção contínua** do sistema.

Este projeto representa não apenas uma solução técnica, mas também um exemplo concreto de como a tecnologia pode impactar positivamente a saúde pública, especialmente em regiões vulneráveis a surtos de dengue.

## REFERÊNCIAS

BECK, Kent. *Test-Driven Development: By Example*. Boston: Addison-Wesley, 2002.

CHACON, Scott; STRAUB, Ben. *Pro Git*. 2. ed. Berkeley: Apress, 2014.

DAVENPORT, Thomas H.; HARRIS, Jeanne G. *Competing on Analytics: The New Science of Winning*. Boston: Harvard Business Review Press, 2007.

DATE, C. J. *An Introduction to Database Systems*. 8. ed. Boston: Pearson, 2004.

ELMASRI, R.; NAVATHE, S. B. *Fundamentals of Database Systems*. 6. ed. Boston: Pearson, 2011.

FOWLER, Martin. *Continuous Integration: Improving Software Quality and Reducing Risk*. Boston: Addison-Wesley, 2006.

GRAHAM, Richard. *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. Sebastopol: O'Reilly Media, 2016.

KUMAR, A.; SINHA, S.; SAHU, A. *Performance evaluation of MySQL and MongoDB*. International Journal of Advanced Research in Computer Science and Software Engineering, 2016.

LINTHICUM, David. *Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide*. Boston: Addison-Wesley, 2017.

MCLUHAN, Marshall. *Understanding Media: The Extensions of Man*. Cambridge: MIT Press, 2015.

NIELSEN, Jakob. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1993.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Database System Concepts*. 6. ed. New York: McGraw-Hill, 2013.

WROBLEWSKI, Luke. *Mobile First*. New York: A Book Apart, 2011.

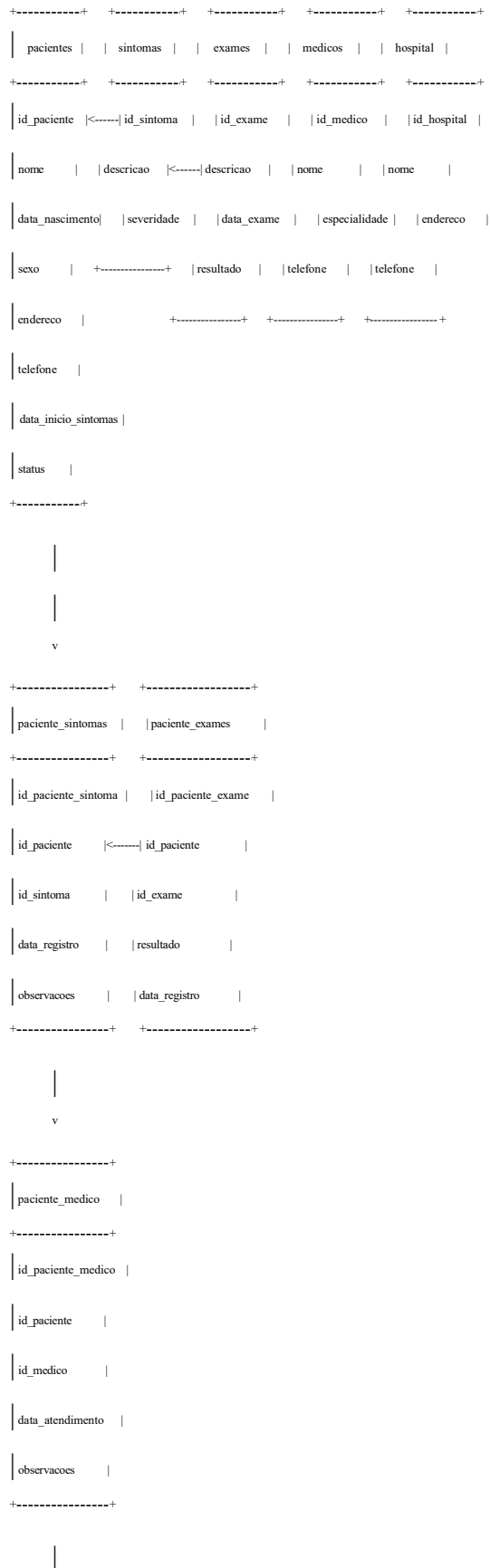


## ANEXOS

### 1. Entrevista com profissional de saúde:

Entrevistada: Juliana Godoi Função: Agente Comunitário de Saúde Unidade: PSF III Cidade: Dourado-SP	
Apresentação	Meu nome é Juliana Godoi, sou agente comunitário de saúde a 17 anos, nesta unidade.
Composição da equipe	Nossa equipe conta com: 2 clínicos gerais 1 enfermeira 2 técnicas de enfermagem 1 dentista 4 agentes de saúde 1 serviços gerais
Estrutura e equipamentos	Nossa unidade conta com amplo espaço físico, acesso a Internet, 6 computadores, 3 impressoras, telefone fixo, e nós agentes de saúde não contamos com tablets ou celulares da unidade, todos os nossos relatórios são elaborados e armazenados em papel, quando necessário fazemos uso de nossos celulares pessoais para manter contato com pacientes.
Principais demandas	Nossa principal demanda no momento é justamente o armazenamento de dados, especificamente no controle de pacientes suspeitos e positivos de dengue, nosso foco no momento.
A criação de um app, voltado para essa demanda, facilitaria o trabalho de vocês?	Sem dúvidas, além de facilitar o trabalho, resolveria questões de armazenamento e acesso de dados, diminuiria a quantidade de papel utilizado, gerando inclusive economia de material, seria ótimo.

## 2. Diagrama Relacional:



v	
+-----+	
internacao	
+-----+	
id_internacao	
id_paciente	
id_hospital	
data_internacao	
data_alta	
+-----+	