



Sistema de Ensino Presencial Conectado

Superior em Tecnologia em Análise e desenvolvimento de sistemas

ALUNO

**Sistema Zer@Dengue**

POLO – UF

2020

POLO – UF

2020

ALUNO

Utilizando tecnologias atuais e modernas para sistemas delivery

**Sistema Zer@Dengue**

Trabalho de produção textual interdisciplinar individual apresentado à Universidade Pitágoras UNOPAR, como requisito parcial para a obtenção de média semestral nas disciplinas de XXXXXXXXXX.

Orientador: XXXXXXXXX.

|  |
| --- |
|  |

|  |
| --- |
|  |

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc34473737)

[2 DESENVOLVIMENTO. 4](#_Toc34473738)

[2.1 Análise Orientada a Objetos I 4](#_Toc34473739)

[2.2 Banco de Dados I 7](#_Toc34473740)

[2.3 Arquitetura e Organização de Computadores 9](#_Toc34473741)

[2.4 Linguagem de Programação e Estrutura de Dados 9](#_Toc34473742)

[3 CONCLUSÃO 12](#_Toc34473743)

# INTRODUÇÃO

Esse tema proposto, a produção textual a seguir, tem como objetivo desenvolver um sistema para denúncia e monitoramento de focos de dengue.

Serão apresentados os diagramas, a concepção do banco de dados com seu script, levantamento operacional de equipamentos e a solução para a otimização das visitas da Equipe de Vistoria Zer@Dengue*.*

# DESENVOLVIMENTO.

## Análise Orientada a Objetos I

Durante o ciclo de vida de desenvolvimento de software, o desenvolvimento normalmente é dividido em estágios, conceitos abstratos, soltos, usados ​​para separar as atividades que ocorrem em cada fase do desenvolvimento. Muitas vezes, essas etapas podem incluir requerimentos, planejamento, design, e assim por diante.

No caso de metodologias rigorosas de desenvolvimento, como o método cascata, esses estágios são sequenciais e devem ser completamente separados um do outro. Assim, ao criar um aplicativo usando o método de cascata, é improvável que as descobertas feitas durante as fases de testes ou depuração podem afetar as decisões já tomadas durante a fase de planejamento. Essas limitações, juntamente com o rigoroso processo de preparo passo a passo dos modelos em cascata, levaram ao surgimento de modelos interativos como a análise orientada à objetos.

A Análise Orientada a Objetos é o procedimento para identificar requisitos de engenharia de software e desenvolver especificações de software em termos de modelo de objeto de um sistema de software, que compreende objetos em interação.

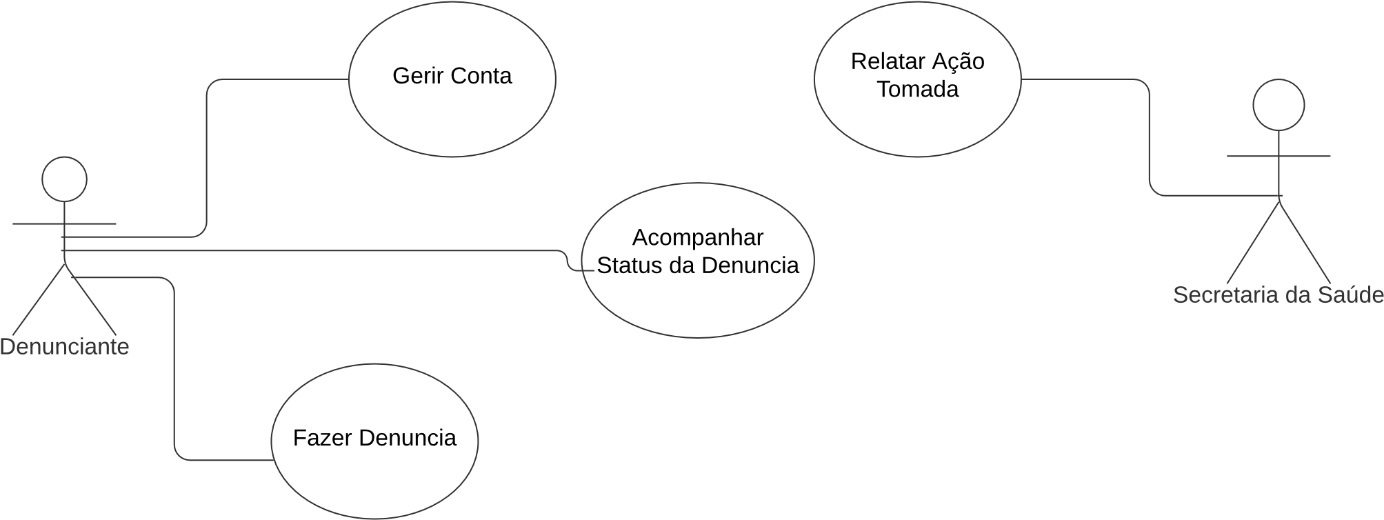
A principal diferença entre a análise orientada a objetos e outras formas de análise é que, na abordagem orientada a objetos, os requisitos são organizados em torno de objetos, que integram dados e funções. Eles são modelados a partir de objetos do mundo real com os quais o sistema interage. Nas metodologias tradicionais de análise, os dois aspectos, funções e dados, são considerados separadamente.

Considerando o estudo de caso do Sistema Zera Dengue, foi realizada a modelagem da atividade de Análise de Sistemas em uma ferramenta CASE de modelagem, contemplando a UML. Levamos em consideração as funcionalidades para realização do cadastro da pessoa física que faz a denúncia, cadastro da localização da denúncia, cadastro da denúncia e consulta de acompanhamento do status da denúncia.

1. O Diagrama de Use Cases:

Um diagrama de casos de uso é um diagrama dinâmico ou de comportamento na UML. Os diagramas de casos de uso modelam a funcionalidade de um sistema usando atores e casos de uso. Casos de uso são um conjunto de ações, serviços e funções que o sistema precisa executar. Nesse contexto, um "sistema" é algo que está sendo desenvolvido ou operado, como um site. Os "atores" são pessoas ou entidades que operam sob funções definidas no sistema.

Os casos de uso do sistema Zera Dengue foram colocados em uma única imagem para melhor definição, conforme segue:



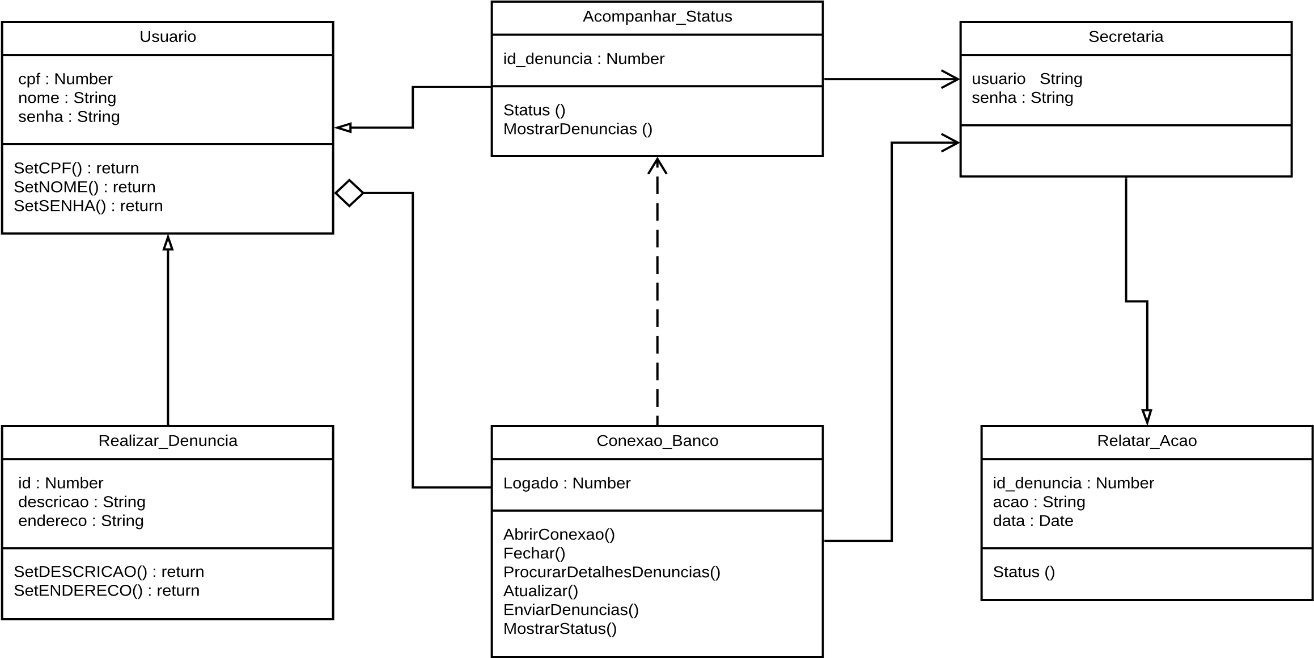
**Figura 1: Diagrama de Caso de Uso**

1. A documentação de todos os Use Cases, no formato numerado, com a descrição dos Cenários Principal e Alternativos:

|  |  |
| --- | --- |
| **CSU 01** - Gerir conta  **Pré-Condição:** Caso não tenha uma conta, então deve criar uma.  **Ator envolvido:** Denunciante | 1. Usuário cadastra os dados no sistema 2. Usuário visualiza os dados no sistema 3. Usuário pode ou não alterar os dados no sistema 4. Usuário encerra o caso de uso. |
| **CSU 02** - Fazer Denúncia  **Pré-Condição:** Ter uma conta e estar “logado” no sistema.  **Ator envolvido:** Denunciante | 1. Denunciante entra no sistema 2. Clica no botão fazer denúncia 3. Informa o endereço do local e a descrição do problema 4. Clica no botão fazer denúncia 5. Encerra o caso de uso |
| **CSU 03** - Relatar ação tomada  **Pré-Condição:** Denuncia ter sido realizada.  **Pós - Condição**: Denunciante deve acompanhar o status da denúncia e visualizar a ação tomada.  **Ator envolvido**: Secretaria da Saúde | 1. Secretaria da Saúde acessa o sistema 2. Verifica as denúncias 3. Verifica gravidade e proximidade 4. Realiza uma ação quanto a isso 5. Informa ação tomada no sistema 6. Encerra o caso de uso |
| **CSU 04** – Acompanhar status da denúncia  **Pré-Condição:** Denúncia ter sido realizada, estar “logado” no sistema  **Atores envolvidos:** Denunciante e Secretaria da Saúde | 1. Denunciante ou Secretaria da Saúde acessa o sistema 2. Verifica as denúncias em andamento 3. Verifica qual é o status da denúncia 4. Encerra caso de uso |

**Tabela 1: Descrição dos casos de uso**

1. O Diagrama de Classe:

Na engenharia de software, um diagrama de classes na UML é um tipo de diagrama de estrutura estática que descreve a estrutura de um sistema, mostrando as classes do sistema, seus atributos, operações, ou métodos, e os relacionamentos entre objetos.

**Figura 2: Diagrama de Classe**

## Banco de Dados I

Banco de dados é uma coleta sistemática de dados. Os bancos de dados suportam armazenamento e manipulação de dados. Os bancos de dados facilitam o gerenciamento de dados.

Uma lista telefônica on-line, por exemplo, definitivamente usaria o banco de dados para armazenar dados referentes a pessoas, números de telefone, outros detalhes de contato etc.

O sistema Zer@Dengue precisa armazenar diversas informações importantes e relevantes, como algumas informações pessoas dos usuários, locais dos focos de dengue, arquivos de mídia, etc.

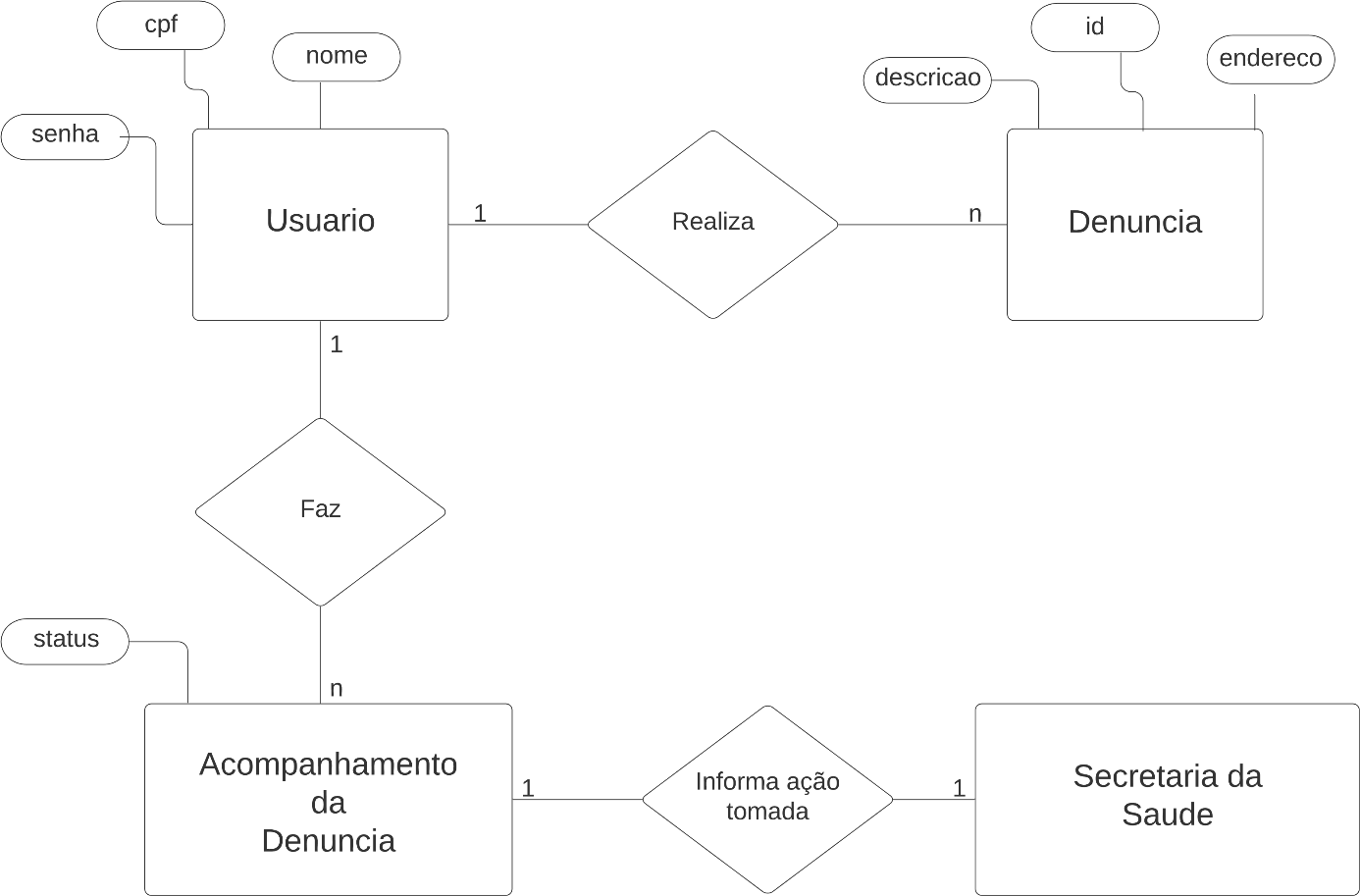
Dessa forma foram levantados os requisitos sobre os tipos de informação que serão armazenadas no banco de dados e criado um modelo entidade-relacionamento (MER).

MER é uma abordagem gráfica para o design de banco de dados que usa Entidade/Relacionamento para representar objetos do mundo real.

Uma entidade é uma coisa ou objeto no mundo real que é distinguível do ambiente circundante. Por exemplo, cada funcionário de uma organização é uma entidade separada. Algumas das principais características das entidades são:

* Uma entidade tem um conjunto de propriedades;
* As propriedades da entidade podem ter valores.

Dito isto, será mostrado na sequência o MER e um simples script SQL para auxiliar na criação das tabelas necessárias para o banco de dados.

**Figura 3: MER**

|  |
| --- |
| CREATE TABLE Usuario(  usr\_cpf number(2),  usr\_nome varchar2(100),  usr\_senha varchar2(8),  CONSTRAINT usr\_pk PRIMARY KEY(usr\_cpf)  );  CREATE TABLE Denuncia(  den\_id number(2),  den\_endereco varchar2(200),  den\_descricao varchar2(300),  CONSTRAINT den\_pk PRIMARY KEY(den\_id)  );  CREATE TABLE Acompanhamento(  aco\_id number(2),  aco\_status varchar2(100),  CONSTRAINT aco\_pk PRIMARY KEY(aco\_id)  ); |

**Quadro 1: script SQL**

## Arquitetura e Organização de Computadores

Para atender a demanda do Sistema Zer@Dengue foi selecionado um PC Desktop completo, com configurações ideais e valor acessível:

* Computador Completo com Monitor LED HDMI 19.5" Intel Core i5 4GB HD 500GB.
* Dupla saída de vídeo: HDMI Full HD e VGA, 6 Conexões USB: 4x traseiras e 2x frontais, 5 Conexões de áudio HD: 3x traseiras e 2x frontais, Conexão de rede: Rede Gigabit de alta velocidade 10/100/1000 RJ45 e Conexão PS2: Para a instalação de mouse e teclado.
* Monitor de LED com 19.5" HQ
* Brilho: 200cd
* Tempo de resposta: 5 ms
* Resolução máxima / recomendada: 1440 x 900
* Suporte de cores: Maior que 16 milhões
* Conexão: VGA e HDMI
* Valor: R$ 1.519,05

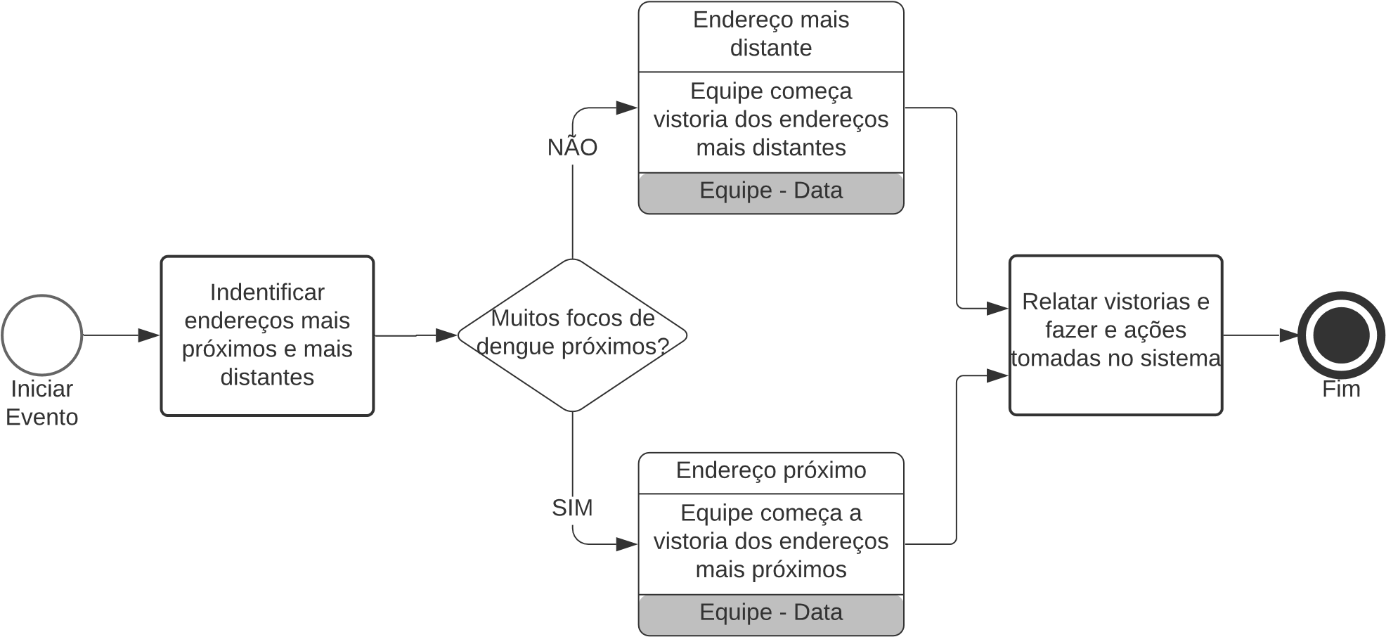
## LINGUAGEM DE PROGRAMAÇÃO E ESTRUTURA DE DADOS

Para otimizar as visitas da Equipe de Vistoria Zer@Dengue aos locais suspeitos de foco do mosquito, foi selecionado uma das mais utilizadas soluções baseadas em lista quando se está tratando de segmentação e organização: Busca Binária.

A busca binária é um eficiente algoritmo para encontrar um item em uma lista ordenada de itens. Ela funciona dividindo repetidamente pela metade a porção da lista que deve conter o item, até reduzir as localizações possíveis a apenas uma.

A ideia consiste em basicamente dividir repetidamente uma lista previamente ordenada de endereços com base na proximidade e gravidade do caso, ambas podendo resultar em um devido peso, dividimos essa lista e aplicamos a busca binária até encontrarmos o menor peso, dessa forma podemos saber quais locais atender primeiro seguindo do mais grave para o menos grave, ou vice-versa.

Um exemplo:

**Figura 4: Diagrama**

Consideremos os nós dessa árvore de busca binária devidamente balanceada como os casos de dengue, agora consideremos os números dentro desses nós como sendo os “pesos” baseados na distância e gravidade designados aos nós. Conseguimos identificar claramente que organizando um roteiro uma equipe pode começar do caso mais “leve” e alcançar posteriormente o caso mais “grave”, ou vice-versa.

Com uma organização logística adequada, essa forma torna-se viável e tangível diante de tudo que foi abordado até então.

# CONCLUSÃO

Após a análise e levantamentos dos dados, o projeto foi iniciado falando um pouco sobre análise orientada a objetos e sua importância. Foi preciso utilizar banco de dados para armazenar os dados sensíveis do sistema Zer@Dengue e produzir um script para facilitar sua criação.

Através de uma pesquisa sobre computadores disponíveis no mercado foi possível identificar a configuração necessária para os equipamentos que farão parte da empresa.

Por fim, foi apresentada uma solução para que a Equipe de Vistoria Zer@Dengue efetue suas visitas aos locais suspeitos de foco do mosquito de forma otimizada.

###### REFERÊNCIAS

FABRIS, P. P. G.; MORAES, E. M.; HISATOMI, M.. **Análise orientada a objetos I**. Londrina: Editora e Distribuidora Educacional S.A., 2018. Disponível em < https://biblioteca-virtual.com/detalhes/livro/1076> acesso em 06 mar. 2020.

PRESSMAN, R.; MAXIM, B. **Engenharia de software: uma abordagem profissional**. 8 ed. Porto Alegre: AMGH, 2016. Disponível em < https://integrada.minhabiblioteca.com.br/books/9788580555349> acesso em 06 mar. 2020.

SANTANA, Gisele Alves. **Linguagens de programação e estruturas de dados**. Londrina: Editora e Distribuidora Educacional S.A. 2018.