



Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

DAYVSON SOUZA DE MELO

GABRIEL FERREIRA DA SILVA

JOÃO ROBERTO ARAUJO DE SOUZA

MÁRCIA MIRELLA DE ALBUQUERQUE PINTO TINÔCO

MATHEUS DA SILVA BELO

MAYCON PAULINO DA SILVA

WASHINGTON CASTELO BRANCO FILHO

# **SISTEMA ZER@DENGUE**

MAIO 2020



Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

DAYVSON SOUZA DE MELO

GABRIEL FERREIRA DA SILVA

JOÃO ROBERTO ARAUJO DE SOUZA

MÁRCIA MIRELLA DE ALBUQUERQUE PINTO TINÔCO

MATHEUS DA SILVA BELO

MAYCON PAULINO DA SILVA

WASHINGTON CASTELO BRANCO FILHO

Trabalho de conclusão semestral apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como requisito parcial para a obtenção de média semestral nas disciplinas Lógica Computacional, Análise e Modelagem de Sistemas, Algoritmos e Programação Estruturada, Modelagem Banco de Dados e Arquitetura e Organização de Computadores.

Professores: Adriane Aparecida Loper, Iolanda Cláudia Sanches Catarino, Vanessa Matias Leite, Gilberto Fernandes Junior e Paulo Kiyoshi Nishitani.

MAIO 2020  
**SUMÁRIO**

1 - INTRODUÇÃO .....	03
2 - DESENVOLVIMENTO .....	05
2.1 - LÓGICA COMPUTACIONAL .....	05
2.2 - ANÁLISE E MODELAGEM DE SISTEMAS .....	06
2.3 - ALGORITMOS E PROGRAMAÇÃO ESTRUTURADA .....	28
2.4 - MODELAGEM BANCO DE DADOS .....	37
2.5 - ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES .....	41
3 - CONCLUSÃO .....	44
4 - REFERÊNCIAS BIBLIOGRÁFICAS .....	46

## 1 - INTRODUÇÃO

O objetivo deste trabalho é criar um sistema chamado de Zer@dengue, que tem por finalidade ser utilizado por cidadãos para combater a presença de focos de mosquitos da dengue em terrenos baldios, ou qualquer local onde esteja. Baseando-se nas disciplinas de Análise e Modelagem de Sistemas, Algoritmos e Programação Estruturada, Modelagem de Dados, Lógica Computacional e Arquitetura e Organização de Computadores, ministradas online e semi-presencialmente, respaldando-se em todo o conhecimento adquirido durante o semestre letivo, criou-se um sistema em linguagem C++ em parceria com o Ministério da Saúde capaz de cadastrar cidadãos em escala nacional, em um banco de dados, para que cada indivíduo possa enviar fotos do local, criar um texto explicativo sobre o problema a ser resolvido e realizar uma denúncia que será encaminhada à autoridade competente através do próprio sistema.

Os temas abordados serão os seguintes:

Na Análise e Modelagem de Sistemas são apresentadas as etapas iniciais para a criação de um software, desde o processo de software passando por processos de negócio, engenharia de requisitos até os métodos orientados a os objetos do sistema. Nos Algoritmos e Programação Estruturada são dispostos os fundamentos de algoritmos e linguagens de programação, passando pelas estruturas, funções, recursividades e estruturas de dados que compõem a programação do sistema. Na Modelagem de Dados, os modelos de banco de dados contribuem para a criação de relacionamentos entre dados e visualização estratégica dos mesmos no sistema. A Lógica Computacional é apresentada como as teorias e conjuntos da matemática e o estudo da lógica contribui para a elaboração de uma estrutura racional para o sistema. E finalmente, na Arquitetura e Organização de Computadores é exibido como a estrutura física do hardware atua como a base para a implementação e funcionamento do sistema como um todo.

A importância deste trabalho se dá ao fato de que ele atende a uma necessidade pública nacional, que é a vigilância sanitária nas cidades e comunidades, principalmente nas comunidades mais carentes onde não há investimento público suficiente para garantir a erradicação dessa moléstia

contribuindo para a piora das condições urbanas onde os mais pobres vivem, acarretando a proliferação de doenças e morte generalizada.

A finalidade, assim, é proporcionar mais recursos de comunicação entre a população e os agentes sanitários, e ser uma porta de ajudar a população a construir uma qualidade de vida melhor. Além disso, é uma oportunidade para que profissionalmente se desenvolva a consciência voltada para o apelo social, em que a ação individual produza um resultado coletivo, que beneficie a todos. É também um meio de pôr em prática o que foi estudado teoricamente, e observar os possíveis resultados da aplicação da tecnologia no dia a dia dos cidadãos, analisando seu impacto, bem como as mudanças sociais decorrentes de sua utilização como uma ferramenta de expressão democrática na sociedade contemporânea.

## 2 - DESENVOLVIMENTO

### 2.1 - LÓGICA COMPUTACIONAL

A Lógica Computacional está voltada para o desenvolvimento do pensamento lógico, racional do indivíduo, que o fará construir interpretações a partir de um determinado argumento.

Com o estudo da lógica conseguimos fundamentar nossos argumentos, entender e relacionar os fatos apresentados, construir softwares formulando corretamente um algoritmo.

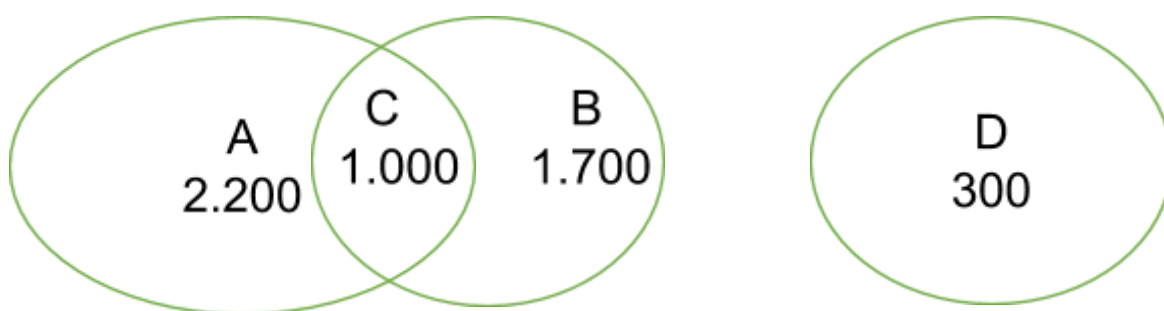
Aplicando seus conceitos de teoria de conjuntos no nosso trabalho podemos apresentar os dados colhidos através de uma entrevista sobre a funcionalidade na utilização do sistema Zer@Dengue pela população.

A pesquisa de funcionalidade do sistema, foi realizada com a participação de 5.200 usuários.

Após a finalização da pesquisa sobre o sistema em uso, captamos as seguintes informações:

- Grupo A - 3200 pessoas gostaram da função “Cadastro.”
- Grupo B - 2700 pessoas gostaram mais da função “Denúncia”.
- Grupo C - 1000 pessoas afirmaram que gostaram das duas funções.
- Grupo D - 300 pessoas não gostaram de nenhuma das duas funções.

Análise gráfica das informações obtidas:



Praticando o raciocínio lógico nas respostas obtidas na pesquisa realizada, inferimos:

- 2.200 usuários gostaram apenas da função cadastro.
- 1.700 usuários gostaram apenas da função denúncia.
- 1.000 usuários gostaram das duas funções – cadastro e denúncia.
- 300 usuários não gostaram de nenhuma das duas funções.

Totalizando uma quantidade de 5.200 usuários entrevistados.

## **2.2 - ANÁLISE E MODELAGEM DE SISTEMAS**

Com o assunto estudado em Análise e Modelagem de Sistemas conseguimos entender os aspectos que estão relacionados com o desenvolvimento de sistemas em seu todo – isto é Pessoas, Computadores - Hardware, Software, Engenharia de Processos e parte lógica – produção de um software.

Para seguirmos um padrão de desenvolvimento é necessário cumprir etapas no processo de Engenharia de Software como:

- Definição de Requisitos
- Projeto em si do sistema – modelagem
- Desenvolvimento de subsistemas
- Integração com sistemas existentes
- Implantação e instalação do sistema
- Acompanhamento da evolução do sistema
- Caso necessário desativação de um sistema.

Para modelar é preciso fazer um levantamento das atividades que precisam ser executadas e ter um diagrama facilita a documentação e entendimento dos colaboradores melhorando a comunicação durante a atividade.

Por meio de diagramas que mostre o fluxo do trabalho, os processos são graficamente apresentados dando uma visão do projeto do início ao fim, identificando possíveis problemas antes dos mesmos ocorrerem.

O Modelo de Processos de Negócio (BPM) tem como objetivos reduzir os desperdícios de tempo, ter uma visão ponta a ponta do processo, e gerar valor para o cliente.

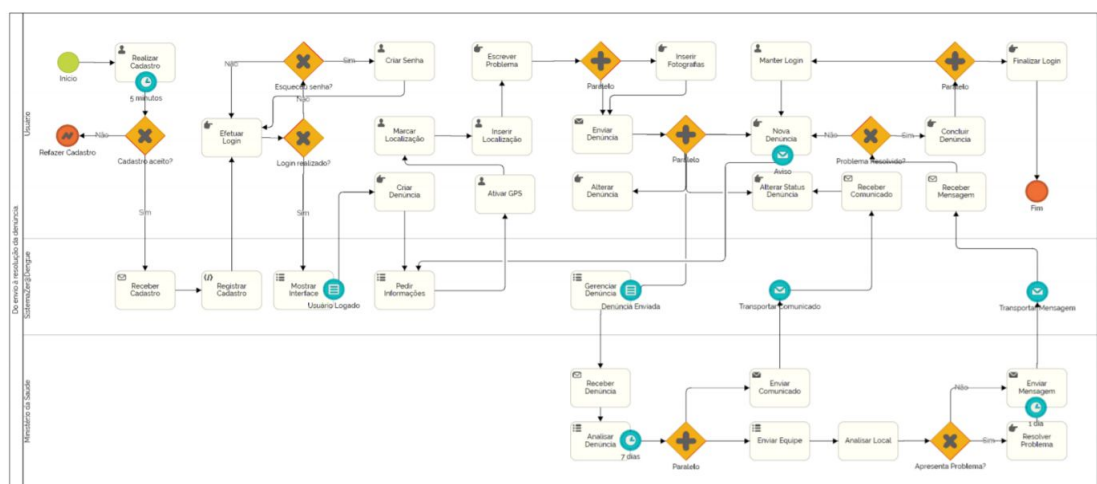
E com isso, é fundamental que cada empresa elabore o seu POP ( Procedimento Operacional Padrão) – que trará uma padronização na realização do processo evitando erros e desvios, e cada passo estará descrito devidamente, garantindo a qualidade do procedimento por qualquer colaborador que vier executar a tarefa.

O sistema Zer@dengue é um sistema idealizado a partir da orientação a objetos. Sendo assim, ele foi projetado visando atender a uma demanda humana, que parte da abstração de uma realidade local, a saber, o lugar onde se encontra o foco do mosquito da dengue.

Segundo Sommerville:

Os objetos incluem dados e operações para manipulá-los. Portanto eles podem ser entendidos e modificados como entidades independentes. Como os objetos estão associados a coisas, muitas vezes há um mapeamento claro entre as entidades do mundo real (como os componentes de hardware) e os objetos que as controlam no sistema.(2019,p. 177)

## Modelo do Processo de Negócio - Tipo Fluxo de Trabalho



Fonte - Elaborado pelos alunos



## Descrição do BPM ZER@DENGUE

### **[Usuário]**

**Início:** O usuário acessa o sistema.

**Tarefa (Realizar Cadastro):** O usuário preenche o formulário de cadastro.

**Intermediário Temporizador (5 minutos):** O usuário tem 5 minutos para se cadastrar no sistema.

**Desvio Exclusivo (Cadastro aceito?)**

**Não - Erro (Refazer Cadastro):** O usuário refaz o formulário de cadastro.

### **[Sistema Zer@Dengue]**

**Sim - Tarefa (Receber Cadastro):** O sistema recebe os dados do cadastro realizado pelo usuário.

**Tarefa (Registrar Cadastro):** O sistema cadastra o usuário de acordo com os dados inseridos.

### **[Usuário]**

**Tarefa (Efetuar Login):** O usuário preenche o formulário de login.

**Desvio Exclusivo (Login realizado?)**

**Não - Desvio Exclusivo (Esqueceu senha?)**

**Não - Tarefa (Efetuar Login):** O usuário preenche o formulário de login.

**Sim - Tarefa (Criar Senha):** O usuário tem a opção de registrar uma nova senha.

**Tarefa (Efetuar login):** O usuário preenche o formulário de login.

**Desvio Exclusivo (Login realizado?)**

### **[Sistema Zer@Dengue]**

**Sim - Tarefa (Mostrar Interface):** O sistema faz com que o usuário entre em contato com a sua interface.

**Intermediário Condicional (Usuário Logado):** O sistema mostra a interface apenas se o usuário estiver logado.

**[Usuário]**

**Tarefa (Criar Denúncia):** O usuário cria uma denúncia.

**[Sistema Zer@Dengue]**

**Tarefa (Pedir Informações):** O sistema mostra um formulário pedindo ao usuário informações referentes à denúncia.

**[Usuário]**

**Tarefa (Ativar GPS):** O usuário ativa o GPS pelo dispositivo utilizado.

**Tarefa (Marcar Localização):** O usuário seleciona no mapa e insere o lugar do foco da dengue.

**Tarefa (Inserir Localização):** O usuário insere os dados geográficos no qual está localizado.

**Tarefa (Escrever Problema):** O usuário descreve o problema no formulário.

**Desvio Paralelo**

**(1) - Tarefa (Inserir Fotografias):** O usuário tem a opção de inserir fotos armazenadas ou tirar fotos a partir da câmera do dispositivo utilizado.

**(2) - Tarefa (Enviar Denúncia):** O usuário finaliza o formulário e envia a denúncia.

**Desvio Paralelo**

**(1) - Tarefa (Alterar Denúncia):** O usuário tem a opção de alterar o formulário.

**(2) - Tarefa (Alterar Status Denúncia):** O usuário tem a opção de alterar o status da denúncia enviada.

**(3) - Tarefa (Nova Denúncia):** O usuário cria uma nova denúncia.

**Intermediário Mensagem (Aviso):** Aviso que o usuário pode enviar no máximo 20 solicitações.

#### [Sistema Zer@Dengue]

(4) - Tarefa (Gerenciar Denúncia): O sistema registra a data e hora que o usuário fez a denúncia e transporta os dados da denúncia para o Ministério da Saúde.

Intermediário Condicional (Denúncia Enviada): O sistema apenas poderá gerenciar a denúncia se a mesma tiver sido enviada.

#### [Ministério da Saúde]

Tarefa (Receber Denúncia): O Ministério da Saúde recebe a denúncia do usuário.

Tarefa (Analisar Denúncia): O Ministério da Saúde analisa a denúncia do usuário.

Intermediário Temporizador (7 dias): O Ministério da Saúde tem um prazo de 7 dias para atender a denúncia.

#### Desvio Paralelo

(1) - Tarefa (Enviar Comunicado): O Ministério da Saúde envia um comunicado para o usuário de que o status da denúncia foi alterado para: Em Andamento.

#### [Sistema]

Intermediário Mensagem (Transportar Comunicado): O sistema transporta o comunicado do Ministério da Saúde para o usuário.

#### [Usuário]

Tarefa (Receber Comunicado): O usuário recebe o comunicado do Ministério da Saúde.

Tarefa (Alterar Status Denúncia): O usuário tem a opção de alterar o status da denúncia enviada.

#### [Ministério da Saúde]

(2) - Tarefa (Enviar Equipe): O Ministério da Saúde envia uma equipe até o local do foco da dengue.
Tarefa (Analisar Local): A equipe do Ministério da Saúde analisa o local do foco da dengue.
Desvio Exclusivo (Apresenta Problema?): Local do foco da dengue apresenta problemas?
Sim - Tarefa (Resolver Problema): O Ministério da Saúde resolve o problema presente no local do foco da dengue.
Tarefa (Enviar Mensagem): O Ministério da Saúde envia uma mensagem de que o problema foi resolvido.
Não - Tarefa (Enviar Mensagem): O Ministério da Saúde envia uma mensagem de que o problema foi resolvido.
Intermediário Temporizador (1 dia): O Ministério da Saúde deve enviar uma resposta em até 1 dia depois de atender a denúncia.

#### [Sistema Zer@Dengue]

Intermediário Mensagem (Transportar Mensagem): O Sistema transporta a mensagem do Ministério da Saúde para o usuário.

#### [Cliente]

Tarefa (Receber Mensagem): O usuário recebe a mensagem do Ministério da Saúde informando que o problema foi resolvido.

Desvio Exclusivo (Problema Resolvido?)

Não - Tarefa (Nova Denúncia): O usuário cria uma nova denúncia.

Sim - Tarefa (Concluir Denúncia): O usuário altera o status da denúncia para: Concluído.

Desvio Paralelo

(1) - Tarefa (Manter Login): O usuário mantém o login.

Tarefa (Nova Denúncia): O usuário cria uma nova denúncia.

(2) - Tarefa (Finalizar Login): O usuário finaliza o login.

Término (Fim)

O POP ( Procedimento Operacional Padrão) é ordenado por uma sequência de elementos:

1. Título do procedimento;
2. Local de aplicação;
3. Setores responsáveis pela tarefa elencados por cargos,
4. Profissional pela elaboração do POP;
5. Informação quanto aos materiais envolvidos na tarefa;
6. Organização da documentação de referência,
7. Descrição de Siglas, caso necessário,
8. Descrição do passo a passo do procedimento, incluindo etapas e sequência de realização;
9. Apresentação de Gráficos e fluxogramas, no caso necessário;
10. Perspectiva de revisão - o que deve ser feito pelo menos anualmente.

## Exemplo de Modelo de Aplicação do POP ( Procedimento Operacional Padrão)



**pitágoras**

PROCEDIMENTO OPERACIONAL PADRÃO – POP

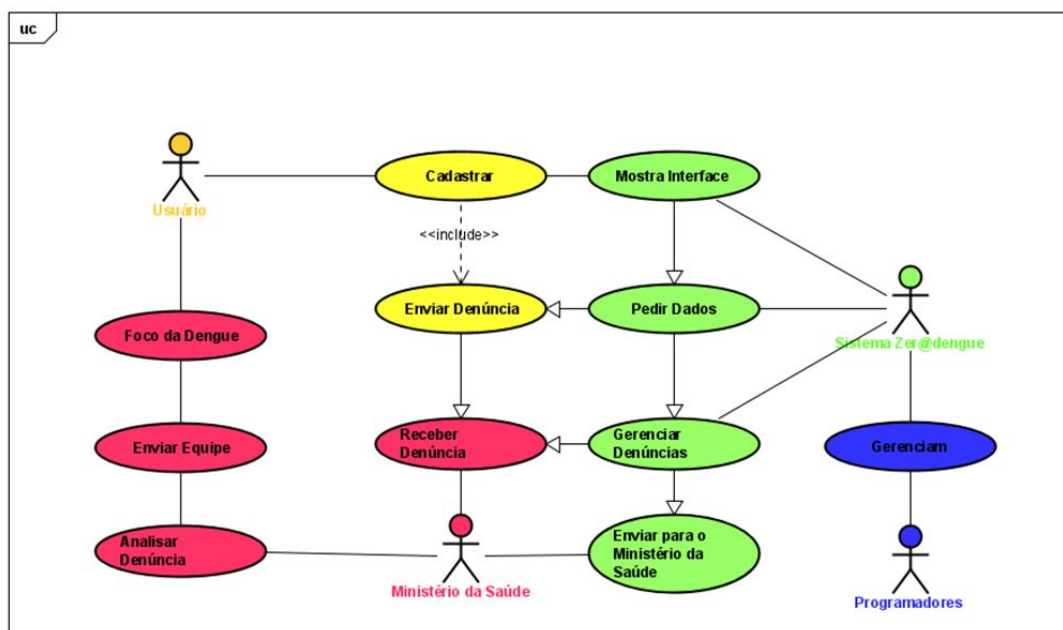
Versão – 001

Data Emissão - MAIO/2020 Data de Vigência - 10/MAIO/2020 Próxima Revisão - MAIO/2020

Área – Análise e Desenvolvimento de Sistemas

Programa ZER@DENGUE

### INFORMAÇÃO DO FLUXOGRAMA NO SISTEMA ZER@DENGUE





**pitágoras**

PROCEDIMENTO OPERACIONAL PADRÃO – POP

Versão – 001

Data Emissão - MAIO/2020 Data de Vigência - 10/MAIO/2020 Próxima Revisão - MAIO/2020

Área – Análise e Desenvolvimento de Sistemas

Programa ZER@DENGUE

### **INFORMAÇÃO AO USO DO SISTEMA ZER@DENGUE**

O sistema Zer@dengue tem por finalidade ser utilizado por cidadãos para combater a presença de focos de mosquitos da dengue em terrenos baldios, ou qualquer local onde esteja.

Criou-se um sistema em linguagem C em parceria com o Ministério da Saúde capaz de cadastrar cidadãos em escala nacional, em um banco de dados, para que cada indivíduo possa enviar fotos do local, criar um texto explicativo sobre o problema a ser resolvido e realizar uma denúncia que será encaminhada à autoridade competente através do próprio sistema.

#### **Utilização do aplicativo Zer@dengue:**

Destina-se a qualquer cidadão que possa denunciar ao Ministério da saúde possíveis locais de foco de dengue e com isso contribuir com o monitoramento e prevenção da saúde da comunidade que vive.

## COMO É FEITO O ACESSO AO SISTEMA ZER@DENGUE

- 1.O usuário se Cadastra no sistema.
- 2.O usuário faz Login e entra no sistema.
- 3.O usuário entra em contato com a interface do sistema.
- 4.O usuário escolhe se vai enviar fotos ou não.
- 5.O usuário escreve um texto relatando o problema.
6. Ativa o GPS.
- 7.Marca no mapa o local que escolher.
- 8.O usuário envia a sua localização.
- 9.O usuário envia a denúncia.
- 10.O Ministério da Saúde recebe a denúncia.
- 11.O Ministério da Saúde analisa a denúncia.
- 12.O Ministério da Saúde envia uma equipe até o local da denúncia;
- 13.A equipe resolve o problema;
- 14.O Ministério da Saúde informa ao usuário que o problema foi resolvido.
- 15.O usuário finaliza a denúncia;
- 16.O usuário mantém o cadastro.
- 17.O usuário pode enviar novas denúncias.
- 18.Fim do processo.





**pitágoras**

PROCEDIMENTO OPERACIONAL PADRÃO – POP

Versão – 001

Data Emissão - MAIO/2020 Data de Vigência - 10/MAIO/2020 Próxima Revisão - MAIO/2020

Área – Análise e Desenvolvimento de Sistemas

Programa ZER@DENGUE

### **RESPONSÁVEIS PELO POP**

Alunos do 1.semestre do curso superior de Análise e Desenvolvimento de Sistemas da Faculdade Pitágoras - 2020.1

DAYVSON SOUZA DE MELO

GABRIEL FERREIRA DA SILVA

JOÃO ROBERTO ARAUJO DE SOUZA

MÁRCIA MIRELLA DE ALBUQUERQUE PINTO TINÔCO

MATHEUS DA SILVA BELO

MAYCON PAULINO DA SILVA

WASHINGTON CASTELO BRANCO FILHO

### **REVISÃO, APROVAÇÃO E PUBLICAÇÃO:**

- Deverá ser realizada anualmente a partir da data de emissão.



**pitágoras**

PROCEDIMENTO OPERACIONAL PADRÃO – POP

Versão – 001

Data Emissão - MAIO/2020 Data de Vigência - 10/MAIO/2020 Próxima Revisão - MAIO/2020

Área – Análise e Desenvolvimento de Sistemas

Programa ZER@DENGUE

### **PASSO A PASSO DA UTILIZAÇÃO DO SISTEMA**

Ao acessar o sistema Zer@dengue, o usuário deve criar um cadastro de acesso na interface do sistema.

Serão solicitados para o cadastro do usuário os seguintes dados:

- Nome
- E-mail
- Telefone
- CEP
- CPF
- Data de Nascimento
- Localização
- Criação de Login e Senha

Após cadastro e realizado o login, terá acesso a interface e nela o usuário poderá realizar a denúncia quanto aos focos de mosquito da dengue encontrado, inserir fotos do foco do mosquito da dengue, enviar dados geográficos com o intuito do ministério da saúde averiguar o problema denunciado.

Com os dados enviados e a localização, o ministério da saúde providenciará a resolução do problema.

Os requisitos funcionais, segundo WERLICH, FEDOZZI e SAMUEL GONÇALVES (2020 ,p. 118) eles estabelecem de forma clara e precisa as funções específicas que o sistema tem que executar, de forma que todos os pedidos do cliente sejam satisfeito quando o software estiver concluído.

Já nos requisitos não-funcionais, restrições são estabelecidas sobre as funcionalidades do sistema, bem como limites específicos de desenvolvimento, tempo de resposta, plataforma e restrições de acessos. Nele, vê a qualidade que o sistema pretende alcançar quanto ao seu alcance de serviços e a forma como são prestados.

Usa-se a sigla RF (Requisito Funcional) seguida de uma numeração para indicar o requisito, e a sigla RNF para (Requisito Não-Funcional).

Nos requisitos funcionais, o sistema deve:

[RF0001] Permitir que o usuário insira, altere ou remova dados: nome, e-mail, telefone, CEP, CPF, localização e data de nascimento;(essencial)

[RF0002] Pedir login do usuário;(essencial)

[RF0003] Pedir senha do usuário;(essencial)

[RF0004] Apresentar opção “esqueci a senha”(importante)

[RF0005] Cadastrar usuários;(requisito essencial)

[RF0006] Receber fotos ou imagens: (importante)

[RF0007] Permitir a criação de textos;(importante)

[RF0008] Ativar a localização/GPS;(essencial)

[RF0009] Enviar dados para o Ministério da Saúde;(essencial)

[RF00010] Receber dados do Ministério da Saúde;(essencial)

[RF00011] Registrar a data e hora que o usuário faz a denúncia;(desejável)

[RF00012] Permitir que o Status da denúncia seja alterado;(essencial)

Figura 1.0 - Tabela de Requisitos Funcionais

<b>Requisitos Funcionais</b>	<b>Descrição</b>	<b>Classificação</b>
<b>[RF0001] Cadastrar usuário</b>	Ao entrar no sistema o usuário deve criar um novo cadastro para acessar a interface do sistema.	Essencial
<b>[RF0002] Pedir dados completos do Usuário</b>	Dados como: nome, e-mail, telefone, CEP, CPF, localização, data de nascimento e senha. Serão usados para identificar cada usuário.	Essencial
<b>[RF0003] Pedir Login</b>	2- É o nome que o usuário vai usar para entrar no sistema, sempre que quiser.	Essencial
<b>[RF0004] Pedir Senha</b>	3- É a senha que o usuário vai usar para entrar no sistema, sempre que quiser.	Essencial
<b>[RF0005] Apresentar Opção "Esqueci a senha"</b>	4- é a opção para recuperar a senha, caso o usuário esqueça a mesma.	Essencial
<b>[RF0006] Receber Fotos ou Imagens</b>	5- Opção de o usuário inserir fotos armazenadas ou tirar foto a partir da câmera do smartphone.	Essencial
<b>[RF0007] Permitir a criação de textos</b>	6- Local onde o usuário pode escrever detalhes ou reclamações sobre a sua denúncia;	Essencial
<b>[RF0008] Ativar a localização/GPS</b>	7-O usuário envia os dados geográficos de onde está e seleciona no mapa o lugar do foco da dengue.	Essencial
<b>[RF0009] Enviar dados para o Ministério da Saúde</b>	8-O usuário envia suas informações para o Ministério da Saúde averiguar o problema.	Essencial
<b>[RF0010] Receber dados para o Ministério da Saúde</b>	9-O sistema recebe a resposta do Ministério da Saúde sobre a resolução do problema.	Essencial
<b>[RF0011] Alterar Status da Denúncia</b>	10-O status pode ser classificado em: não concluído, em análise, e concluído.	Essencial
<b>[RF0012] Registrar a data e hora que o usuário faz a denúncia</b>	11-Gravar o dia e hora da denúncia para saber o tempo de resposta, do dia da solicitação até a resolução do problema.	Importante

<b>[RF0013] Permitir que o usuário insira, altere ou remova seus dados</b>	Em eventuais mudanças o usuário altera seus dados pessoais.	Essencial
--	---	-----------

Fonte: elaborado pelos alunos.

E para os Requisitos não-funcionais:

[RNF0001] O usuário tem 5 minutos para se cadastrar no sistema; (importante)

[RNF0002] As fotos devem ter no máximo 4 mb; (desejável)

[RNF0003] Os textos devem ter no máximo 1000 palavras; (desejável)

[RNF0004] Se o usuário não ativar a localização, o sistema não funcionará corretamente; (essencial)

[RNF0005] O ministério da Saúde tem um prazo de 7 dias para atender a denúncia; (importante)

[RNF0001] O ministério da Saúde deve enviar uma resposta de até 1 dia(24h) depois de atender a denúncia; (importante)

[RNF0001] O ministério da Saúde recebe mensagens até às 22:00, de segunda-feira á sexta-feira; (importante)

[RNF0001]O sistema deve enviar uma mensagem pedindo que o usuário altere o status, assim que a denúncia for atendida; (importante)

[RNF0001] O usuário pode enviar no máximo 20 solicitações; (essencial)

[RNF0001] A letra do texto deve ter fonte padrão arial 12.(desejável)

[RNF0001] O sistema não tem restrição de idade; (importante)

[RNF0001] O sistema deve ser implementado em linguagem de programação C++; (essencial)

[RNF0001] O sistema deve ser disponibilizado de forma gratuita; (essencial)

[RNF0001] O Sistema tem suporte apenas para Android, a partir do 6.0; (desejável)

[RNF0001] O tamanho do sistema deve ser no máximo 50 Mbs; (desejável)

[RNF0001] O sistema requer permissão de armazenamento; (essencial)

[RNF0001] O sistema requer permissão para acessar a câmera; (essencial)

[RNF0001] O sistema requer conexão com a internet; (essencial)

Figura 2.0 - Tabela de Requisitos Não Funcionais

Requisitos Não Funcionais	Descrição	Classificação
<b>[RNF0001]</b> <b>Confiabilidade:</b>	O sistema se comporta de modo confiável, sem apresentação de erros e inconsistências no processamento das informações.	Essencial
<b>[RNF002]</b> <b>Usabilidade:</b>	O Sistema possui boa usabilidade e visibilidade, com telas claras, fáceis de usar, com recursos de interface interessantes, atendendo até mesmo, quesitos de acessibilidade.	Essencial
<b>[RNF003]</b> <b>Eficiência:</b>	O sistema responde pelas tarefas de modo adequado e suficiente, desobrigando o usuário de buscar outras ferramentas para o cumprimento de suas tarefas.	Importante

<b>[RNF004]</b> <b>Portabilidade:</b>	Possibilidade do sistema ser utilizado em diferentes plataformas (tanto sistemas operacionais, quanto de hardware diferenciadas)	Essencial
<b>[RNF005]</b> <b>Manutenibilidade:</b>	Capacidade de (ser fácil) manutenção. facilidade de alterações de suas estrutura.	Essencial
<b>[RNF006]</b> <b>Armazenamento:</b>	O sistema realizará backup diário( fazer backup dos dados todos os dias em um determinado horário x).	Essencial
<b>[RNF007]</b> <b>Cadastro e mudança códigos de segurança:</b>	O sistema permitirá o usuário cadastrar suas senhas com no mínimo 8 caracteres (deverá permitir que o usuário cadastre uma senha com no mínimo 8 caracteres, a senha deve conter letras maiúsculas, minúsculas, numeração ou caracteres especiais).	Essencial
<b>[RNF008]</b> <b>Segurança:</b>	Não somente voltado para questões de login de usuário, mas também, na integridade das informações, consistência das informações.  (o sistema deverá ter um dispositivo de segurança e proteção para os dados do usuário).	Essencial
<b>[RNF009]</b> <b>Identidade visual :</b>	Logotipo do Zer@dengue em todas as telas (o sistema devera ter a logo do Zer@dengue em todas as telas).	Essencial

<b>[RNF010]</b> <b>Custo:</b>	Restrições orçamentárias para a realização do projeto(só será possível o gasto do orçamento do projeto até um valor x).	Essencial
<b>[RNF011]</b> <b>Performance:</b>	Capacidade de processamento de informações no decurso do tempo.(o sistema deverá emitir uma resposta a solicitação de troca de senha, alteração de e-mail, informações sobre o processo de validação do cadastro e etc, em x minutos).	Essencial
<b>[RNF012]</b> <b>Cores:</b>	O sistema deverá ter uma paleta de cores correspondentes as da logotipo do Zer@dengue(incluindo a marca d'agua em cada página que direcionar o usuário).	Desejável
<b>[RNF013]</b> <b>Legislação:</b>	O sistema precisa atender a lei de proteção aos dados pessoais (garantir a proteção dos dados disponibilizados pelos usuários, sejam documentos pessoais ou de localização).	Essencial

Figura 2.0 - | Configuração Básica da Documentação da Especificação de Requisitos

<b>Identificador</b>	<b>RF002</b>					
<b>Nome</b>	Pedir dados completos do usuário					
<b>Módulo</b>	Cadastro					
<b>Data de Criação</b>	01/01/2020	<b>Autor:</b>	Alunos			



<b>Data da última alteração</b>	02/01/2020	Autor:	Alunos			
<b>Versão</b>	1.0	Prioridade:	Essencial			
<b>Descrição</b>	Dados como: nome, e-mail, telefone, CEP, CPF, localização, data de nascimento e senha. Serão usados para identificar cada usuário.					

Fonte: elaborado pelos alunos

É necessário classificar os requisitos segundo a ordem de importância de cada um para o sistema.

A classificação se dá entre: essencial, importante e desejável.

\* Essencial: é o requisito que é de extrema importância para o software funcionar, sendo fundamental em sua execução e funcionamento, sem o qual o sistema não funcionaria perfeitamente ou nem iniciaria.

A tabela acima mostra um exemplo de como as informações geradas a partir do levantamento de requisitos podem ser usadas de forma organizada para descrever apenas um requisito do sistema, seja funcional ou não-funcional.

Sendo assim, “a documentação gerada da Especificação de Requisitos não segue um padrão pré-estabelecido e as empresas adaptam os formulários da documentação de acordo com suas necessidades internas de desenvolvimento.” (WERLICH,FEDOZZI e SAMUEL GONÇALVES, 2020 ,p. 154)

“O sistema não poderá ser implantado ou concluído caso um requisito essencial não esteja totalmente realizado.” (WERLICH,FEDOZZI e SAMUEL GONÇALVES, 2020 , p. 115-116).

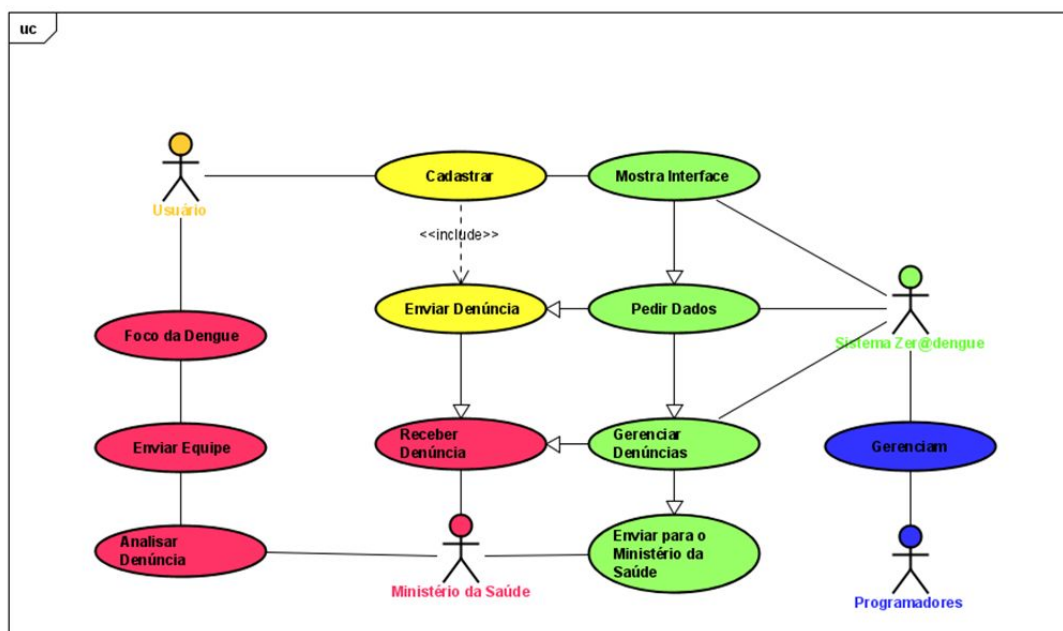
\*Importante: ele não é fundamental, e pode ficar em segundo plano.

“São requisitos desejáveis, porém não imprescindíveis.” (WERLICH,FEDOZZI e SAMUEL GONÇALVES, 2020 , p.116)

Desejável: é um requisito que não é essencial para o software estar concluído, podendo ser eliminado antes mesmo de o software ficar pronto.

“Esse é um requisito que, sendo classificado como desejável, pode eventualmente nem ser desenvolvido caso haja atrasos e seja necessária a implantação do software.”( WERLICH,FEDOZZI e SAMUEL GONÇALVES, 2020 , p.116)

Diagrama de Use Cases, conforme os requisitos funcionais:



Fonte: captura de tela da ferramenta Astah elaborada pelos alunos.

### Diagrama de Caso de Uso (Use Case)

Segundo (WERLICH, FEDOZZI e SAMUEL GONÇALVES, 2020, p. 205) “O diagrama de caso de uso fornece uma visão geral dos objetivos que os usuários (os atores) desejam alcançar utilizando o sistema.” Sendo assim, os elementos mais importantes do diagrama são os atores, relacionamentos e fluxo de eventos.

A UML (Unified Modeling Language – linguagem de modelagem unificada), “faz uso de uma linguagem gráfica, o que nos permite visualizar com mais facilidade os objetos e suas interações (relacionamentos), bem como construir, especificar e

documentar os artefatos gerados por um software.” (WERLICH, FEDOZZI e SAMUEL GONÇALVES, 2020, p. 202).

O diagrama de caso de uso é derivado da UML, e está inserido sob a estrutura de um diagrama de comportamento. Ele é usado para representar graficamente todos os atores, relacionamentos e fluxos de trabalho que o sistema realiza para funcionar corretamente.

Conforme VAZQUEZ e SIMÕES (2016 , p. 196) "um diagrama é uma representação visual, em geral composto de figuras, de um requisito ou conjunto de requisitos." Desde elementos simples, às subcamadas do sistema, o seu objetivo é ser um documento ilustrativo para que todos os stakeholders envolvidos na construção do software consigam entender qual é a lógica de funcionamento, bem como ser um meio de verificar possíveis inconsistências em seus processos, os quais podem ser alterados para corrigir os erros e proporcionar o correto sequenciamento dos processos que o compõem. Segundo SOMMERVILLE ( 2018 , p. 179 ) “ cada um desses casos de uso deve ser descritos em linguagem natural estruturada.” Pois assim, os objetos do sistema são mais facilmente identificados e melhora-se a compreensão do que o sistema irá fazer.

#### Descrição dos elementos de um Use Case

Fluxo de Eventos (Use Cases - Casos de Uso): segundo Werlich, Fedozzi e Samuel Gonçalves (2020, p. 157), todos os Casos de Uso (cada elipse) são detalhados e especificados com orientações sobre sua funcionalidade, além disto, um Caso de Uso pode interagir com um ou mais Casos de Uso.

Ator (boneco): Os atores são usuários ou grupos de usuários que interagem com o sistema. A interação se dá através de relacionamentos ou associações entre atores e os outros elementos, que criam vários processos em modo cíclico, os quais representam a movimentação lógica do sistema, durante sua execução. Segundo (WERLICH, FEDOZZI e SAMUEL GONÇALVES, 2020, p. 157) “Um Ator pode somente fornecer informações para o sistema ou somente receber informações do sistema, ou ainda, fornecer e receber informações para o sistema. Num diagrama pode haver vários Atores.”.

a) Usuário: é o ator responsável por tomar a iniciativa de interagir com o sistema. Ele é o objeto principal, para o funcionamento do projeto, pois sem ele o projeto não tem finalidade prática. Realiza o cadastramento, e prestação de informações pessoais, em como descreve de maneira subjetiva, o porquê de utilizar o serviço de comunicação e acompanhamento do caso do foco da dengue.

b) Sistema Zer@dengue: é o software ou sistema, capaz de ser um ambiente de interação com o usuário, através da exibição de interface gráfica, o qual coleta todos os dados necessários fornecidos pelo usuário, os armazena e gerencia.

c) Ministério da Saúde: é o órgão responsável por receber as denúncias criadas por um usuário, através do sistema Zer@dengue. O Ministério da Saúde tem uma conta única no sistema, a qual receberá todas as denúncias, para fim de análise e tomada de decisões corretivas. Recebida a denúncia, o Ministério da Saúde envia uma equipe de prevenção ou correção até o foco do mosquito da dengue para destruí-lo.

d) Programadores: são os estudantes de análise e desenvolvimento de sistemas, responsáveis por criar, configurar, editar ou remover quaisquer funções do sistema, que acharem necessário, em como gerenciar todos os dados e informações que o sistema receber dos usuários, visando garantir a integridade, confiabilidade e inalterabilidade dos mesmos.

e) Foco da Dengue: É o local no qual está o foco do mosquito da dengue ou um local propenso a se tornar um. O usuário deve identifica-lo e descrevê-lo, para que informe o seu endereço ao sistema, e o mesmo possa marcar no mapa onde é o local, podendo ser em uma casa, rua, terreno, ou qualquer localidade, para posterior investigação e início de ação preventiva ou corretiva.

f) Equipe: são os profissionais do Ministério da Saúde, que irão até o foco do mosquito da dengue para agir preventiva ou corretivamente. Eles devem resolver ou não o problema, dependendo das circunstâncias e do local, e fazer um relatório da ação, passando as informações para o Ministério da Saúde.

#### Descrição das Setas

a) Generalização (linha com seta): É a associação entre casos de uso. Segundo WERLICH, FEDOZZI e SAMUEL GONÇALVES (2020 ,p. 157) “representam a herança entre componentes (atores ou casos de uso)”.

- b) Extend (seta com linha tracejada): é a associação de funcionalidade opcional, usada quando um entre dois diagramas será usado a depender de um dos atores, podendo ser utilizado ou não.
- c) Simples (associação):segundo WERLICH,FEDOZZI e SAMUEL GONÇALVES (2020 ,p. 157). “é o relacionamento (a ligação) entre os Atores e outros Casos de Uso.”
- d) Include: é a associação de funcionalidade incondicional. Segundo WERLICH,FEDOZZI e SAMUEL GONÇALVES (2020 ,p. 157) “o tipo de relacionamento entre dois Casos de Uso implica na obrigatoriedade da execução do Caso de Uso que está sendo incluído.”
- e) Fronteira : é o limite do diagrama. Tudo o que estiver dentro desse limite, que é um retângulo com setas pretas, faz parte do diagrama o contrário, não compõe o caso de uso.
- f) Assunto: é o assunto sobre o qual o diagrama se trata, ou seja, o tema.

## **2.3 - ALGORITMOS E PROGRAMAÇÃO ESTRUTURADA**

Após estudo sobre algoritmos podemos inferir que existem várias formas de realizá-los, eles são independentes, porém com a mesma finalidade – executar uma tarefa em sequência linear, ordenada de passos.

Seu estudo é realizado a partir de três partes – Entrada ( dados) , Processamento (execução) , Saída ( solução - objeto atingido).

Utilizamos no sistema Zer@dengue, a linguagem C++, com a estrutura de dados no conceito FIFO( first in, first out) – a informação que entra primeiro, será a primeira a sair. Com isso, a ordem de chegada das denúncias será respeitada e primeiramente deverão ser atendidas. E ainda cada usuário só poderá realizar no máximo 20 solicitações simultâneas.

Através da linguagem C++ podemos verificar algumas características como;

- Variáveis – endereços de memórias que guardam o valor utilizado. Identificação de uma Informação.
- Atribuição – Indica os valores para as variáveis.

- Constantes – Valor armazenado - nunca será alterado.

### **Código Zer@dengue**

```
/*  codigo fonte para o APP Zer@aDengue  */

#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <string.h>
#include <time.h>
#include <queue>
using namespace std;

char nome,x,endereco,foco,email,texto,denuncia,imagem;
int idade,telefone,CPF;

void mostrar (queue <int> f )
{
    while( !f.empty())
```

```

    {
        cout<<f.front()<<" ";
        f.pop();
    }
}

```

```

int main(int argc, char** argv) {          // Nessa Função será inserido os dados do
usuário

```

```

//
-----
-----//

printf("\n\n");

```

```

printf("*=====
=====*\n");

printf("|| _____ || \n");

printf("|| |XXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXX|
|XXXXXXXXXXXXXXXX| || \n");

printf("|| |XXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXX|
|XXXXXXXXXXXXXXXX| || \n");

printf("||      XXXX  |XXX|      |XXX|      XXXX  |XXX  |XXX|
|| \n");

printf("||      XXXX  |XXX|      |XXX|      XXXX  |XXX  |XXX|
|| \n");

printf("||      XXXX  |XXX|_____ |XXX|      XXXXX  |XXX  |XXX|
|| \n");

printf("||      XXXX  |XXXXXXXX|      |XXX|      XXXXX  |XXX  |XXX|
|| \n");

```

```

printf("||   xxxx   |xxxxxxxx|   |xxxxxxxxxxxxxxxxx
|xxxxxxxxxxxxxxxxx|
|| \n");

printf("||   xxxx   |xxx|   |xxxxxxxxxxxxx
|xxxxxxxxxxxxxxxxx|
|| \n");

printf("||   xxxx   |xxx|   |xxx|  xxxx   |xxx   xxx|
|| \n");

printf("||   xxxx   |xxx|   |xxx|  xxxx   |xxx   xxx|
|| \n");

printf("||  xxxx_____ |xxx|_____ |xxx|  xxxx   |xxx
xxx|
|| \n");

printf("|| |xxxxxxxxxxxxxxxxx| |xxxxxxxxxxxxxxxxx| |xxx|   xxxx   |xxx
xxx|
|| \n");

printf("|| |xxxxxxxxxxxxxxxxx| |xxxxxxxxxxxxxxxxx| |xxx|   xxxx   |xxx
xxx|
|| \n");

printf("||
|| \n");

printf("||
|| \n");

printf("||
|| \n");

printf("|| _____
_____ || \n");

printf("|| |xxxxxxxxxxxxxxxxx   |xxxxxxxxxxxxxxxxx| |xxx|   |xxx|
|xxxxxxxxxxxxxxxxx| |xxx|   |xxx|   |xxxxxxxxxxxxxxxxx|   || \n");

printf("|| |xxxxxxxxxxxxxxxxx   |xxxxxxxxxxxxxxxxx| |xxxxx   |xxx|
|xxxxxxxxxxxxxxxxx| |xxx|   |xxx|   |xxxxxxxxxxxxxxxxx|   || \n");

printf("|| |xxx   xxxx   |xxx|   |xxx xxx   |xxx|   |xxx|
|xxx|   |xxx|   |xxx|   || \n");

printf("|| |xxx   xxxx   |xxx|   |xxx| xxx   |xxx|   |xxx|
|xxx|   |xxx|   |xxx|   || \n");

printf("|| |xxx   xxxx   |xxx|_____ |xxx| xxx   |xxx|   |xxx|
_____ |xxx| |xxx| |xxx|_____ || \n");

printf("|| |xxx   xxxx   |xxxxxxxxx|   |xxx| xxx   |xxx|   |xxx|
xxxxxxxxx| |xxx| |xxx| |xxxxxxxxx|   || \n");

printf("|| |xxx   xxxx   |xxxxxxxxx|   |xxx| xxx   |xxx|   |xxx|
xxxxxxxxx| |xxx| |xxx| |xxxxxxxxx|   || \n");

printf("|| |xxx   xxxx   |xxx|   |xxx| xxx   |xxx|   |xxx|
xxx| |xxx|   |xxx| |xxx|   || \n");

```



```

printf("|| |XXX    XXXX |XXX|      |XXX|    XXX |XXX| |XXX|
XXX| |XXX|    |XXX| |XXX|      || \n");

printf("|| |XXX    XXXX |XXX|      |XXX|    XXX XXX| |XXX|
XXX| |XXX|    |XXX| |XXX|      || \n");

printf("|| |XXX    XXXX |XXX|_____ |XXX|    XXXXXX|
|XXX|_____XXX| |XXX|_____|XXX| |XXX|_____ || \n");

printf("|| |XXXXXXXXXXXXXXXXXXXX |XXXXXXXXXXXXXXXXXXXX| |XXX|    XXXXXX|
|XXXXXXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXXXXXX| || \n");

printf("|| |XXXXXXXXXXXXXXXXXXXX |XXXXXXXXXXXXXXXXXXXX| |XXX|    XXXXXX|
|XXXXXXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXXXXXX| |XXXXXXXXXXXXXXXXXXXX| || \n");

printf("||

|| \n");

```

```

printf("*=====
=====*\n\n");

```

```

printf("===== Ola , seja bem vindo ao APP ZeraDengue
===== \n \n");

```

```

printf("===== Ajude-nos a combater esse mosquito
===== \n \n");

```

```

printf("+++++++ Sua informacao vale muito para nos
+++++++ \n \n");

```

```

printf("===== Primeiramente, insira o seu nome:
===== \n \n");

```

```

printf("NOME: ----- ");

```

```

scanf("%[^\\n]s", &nome);

```

```

    getchar();

    printf("\n\n");

    printf("
===== Agora diga-nos a sua idade:

scanf("%d" , &idade);
getchar();

if (idade<=17)
{
    printf("
Peca para o seu responsavel se cadastrar, Apenas maiores de 18 sao
permitidos. \a\n\n");

}else{
    printf("\n continue... ");
}

printf("\n\n");

printf("
===== agora insira o seu endereco com o numero:

printf("ENDERECO -----");

scanf("%[^\\n]s", &endereco);
getchar();

printf("\n Insira seu Email: ");
scanf("%[^\\n]s", &email);
getchar();

```

```

printf("\n Insira seu Telefone: ");
scanf("%[^\n]s", &telefone);

    getchar();

                printf("\n Insira seu CPF: ");
scanf("%[^\n]s", &CPF);

    getchar();


printf("\n\n");

printf("                ===== agora insira endereco do local da ocorrencia
com o numero: =====\n\n");


printf("LOCAL DO FOCO DE DENGUE -----");
scanf("%[^\n]s", &foco);

getchar();

printf("\n\n");


printf("*=====
=====*\n");


printf("\n\n");
printf("Selecione o tipo de ocorrencia: \n\n");
printf("\n Digite a letra que representa o tipo de ocorrencia:\n\n");
printf("a. poca de agua\n");
printf("t. tampa\n");
printf("p. pneu\n");
printf("\n");

```

```

printf("Digite sua opcao:");

scanf("%c", &x);

getchar();

printf("\n");


switch(x)
{
system("cls");

case 'a':

printf("Voce escolheu a \n\n (Poca de agua).\n\n Procure saber se e um vazamento de agua
e se caso for avise a companhia de abastecimento antes de nos encaminhar uma ocorrencia \n");


break;

case 't':

printf("Voce escolheu t \n\n (Tampa de caixa de agua) aberta. \n\n Procure o responsavel do
imovel para avisar ou solicitar que o mesmo corrija o problema, caso nao seja resolvido envie-nos a
ocorrencia. \n");


break;

case 'p':

printf("Voce escolheu p \n\n ()Pneu) acumulando agua. \n\n Se nao for possivel desvirar ou
descartar o pneu por motivo de o mesmo estar em area de acesso perigoso, entao prossiga com a
abertura de ocorrencia. \n");


break;


default:

printf("opcao invalida\n");

}

{

printf("\n Insira uma Imagem do Local do Foco do Mosquito da Dengue: \n");

scanf("%[^\n]s", &imagem);

```

```

getchar();

printf("\n Digite aqui um Texto relatando o Problema: \n");
scanf("%[^\n]s", &texto);
getchar();

printf(" \n Digite 'Sim' para Enviar a Denuncia: \n");
scanf("%[^\n]s", &denuncia);
getchar();

printf("\n Denuncia 1 enviada com Sucesso! \n");

printf("\n Obrigado por usar o Sistema Zer@dengue!Seu pedido foi encaminhado
para o Ministerio da Saude.Em breve,você receberá uma notificação sobre a ação tomada
para corrigir o problema de sua denuncia.Fique ligado!");
}

{

//ordem de chamada das denuncias

printf("\n\n");
printf("\n Denuncias a serem Atendidas:");

queue<int> fila;
fila.push(1); // 1ª denuncia da lista
fila.push(2); //2ª denuncia da lista
fila.push(3); // 3ª denuncia da lista
fila.push(4); // 4ª denuncia da lista
fila.push(5); // 5ª denuncia da lista
fila.push(6); // 6ª denuncia da lista

```

```
fila.pop(); // retira o valor 1
//fila.pop(); // retira o valor 2
//fila.pop(); // retira o valor 3 e assim por diante. Retirar significa que a denuncia foi atendida
e descartada,logo a proxima denuncia sera atendida na ordem crescente.

mostrar(fila);
}

return 0;

}
```

## **2.4 - MODELAGEM BANCO DE DADOS**

O sistema Zer@dengue utiliza um banco de dados para registro de todos os cadastros realizados no sistema incluindo usuários e denúncias. Com isso, o acesso à informação é constante independente do período que foi inserido o dado no sistema.

Tabela de como usar o sistema Zer@dengue:

- 1.O usuário se Cadastra no sistema.
- 2.O usuário faz Login e entra no sistema.
- 3.O usuário entra em contato com a interface do sistema.
- 4.O usuário escolhe se vai enviar fotos ou não.
- 5.O usuário escreve um texto relatando o problema.
6. Ativa o GPS.
- 7.Marca no mapa o local que escolher.
- 8.O usuário envia a sua localização.
- 9.O usuário envia a denúncia.
- 10.O Ministério da Saúde recebe a denúncia.
- 11.O Ministério da Saúde analisa a denúncia.
- 12.O Ministério da Saúde envia uma equipe até o local da denúncia;
- 13.A equipe resolve o problema;
- 14.O Ministério da Saúde informa ao usuário que o problema foi resolvido.
- 15.O usuário finaliza a denúncia;
- 16.O usuário mantém o cadastro.
- 17.O usuário pode enviar novas denúncias.
- 18.Fim do processo.

Fonte: elaborado pelos alunos

### **Como usar o sistema Zer@dengue:**

- i. O usuário informa seu nome completo, CPF, e-mail, telefone, endereço, e CEP para se cadastrar no sistema;
- ii. Em seguida, cria um login e uma senha para entrar no sistema.
- iii. Depois disso, o usuário entra no sistema.
- iv. Ao entrar no sistema ele entra em contato com a interface do mesmo. Nela, haverá disponibilizado o acesso a História da dengue no Brasil; os Sintomas; a Prevenção e o Tratamento.
- v. Ele pode também escolher a opção Denúncia. Com essa escolha o usuário é direcionado as seguintes opções: 1- enviar foto do local(opcional); 2 - Criar texto explicativo sobre a denúncia(obrigatório);3 - Enviar localização do local do foco do mosquito(obrigatório).
- vi. Após isso, ele clica em Enviar Denúncia.

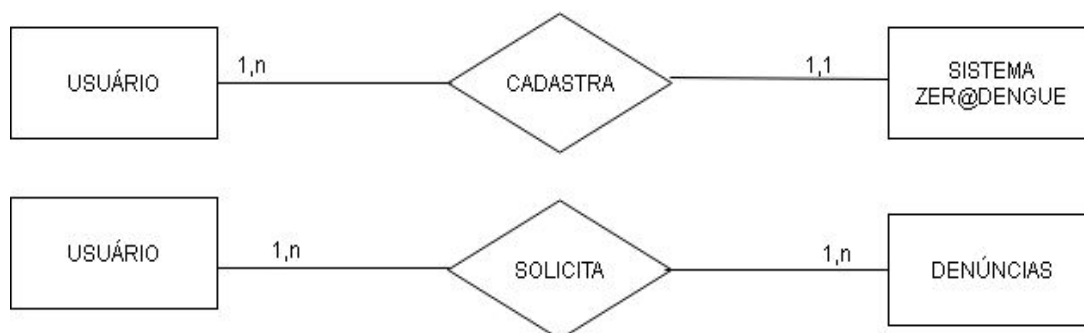
- vii. E a denúncia será encaminhada ao e-mail do Ministério da Saúde.
- viii. Após isso, o usuário deve esperar a resposta do Ministério da Saúde.
- ix. O Ministério da Saúde enviará uma resposta ao e-mail do usuário informando se o problema foi resolvido ou não.
- x. Se resolvido, o usuário entra no sistema e marca a opção "resolvido", 1 dentre 3 opções (não resolvido, resolvido, e em processo), que estará disponível para ele sobre a situação da denúncia. Assim deve ser feito caso a situação não seja resolvida ou esteja em processo.

Exemplo de Tabela Usuário com atributos do sistema:

Nome	CPF	Cidade	CEP	E-mail	Telefone	Data Nascimento	Title 2	Title 3
Matheus	123.456.789-11	Jaboatão dos G.	12345-678	matheus@hotmail.com	1234678	01/01/2001	Value 2	Value 3
Value 4	Value 5	Value 6	Value 4	Value 5	Value 6	Value 4	Value 5	Value 6
Value 7	Value 8	Value 9	Value 7	Value 8	Value 9	Value 7	Value 8	Value 9
Value 10	Value 11	Value 12	Value 10	Value 11	Value 12	Value 10	Value 11	Value 12

Fonte: elaborado pelos alunos

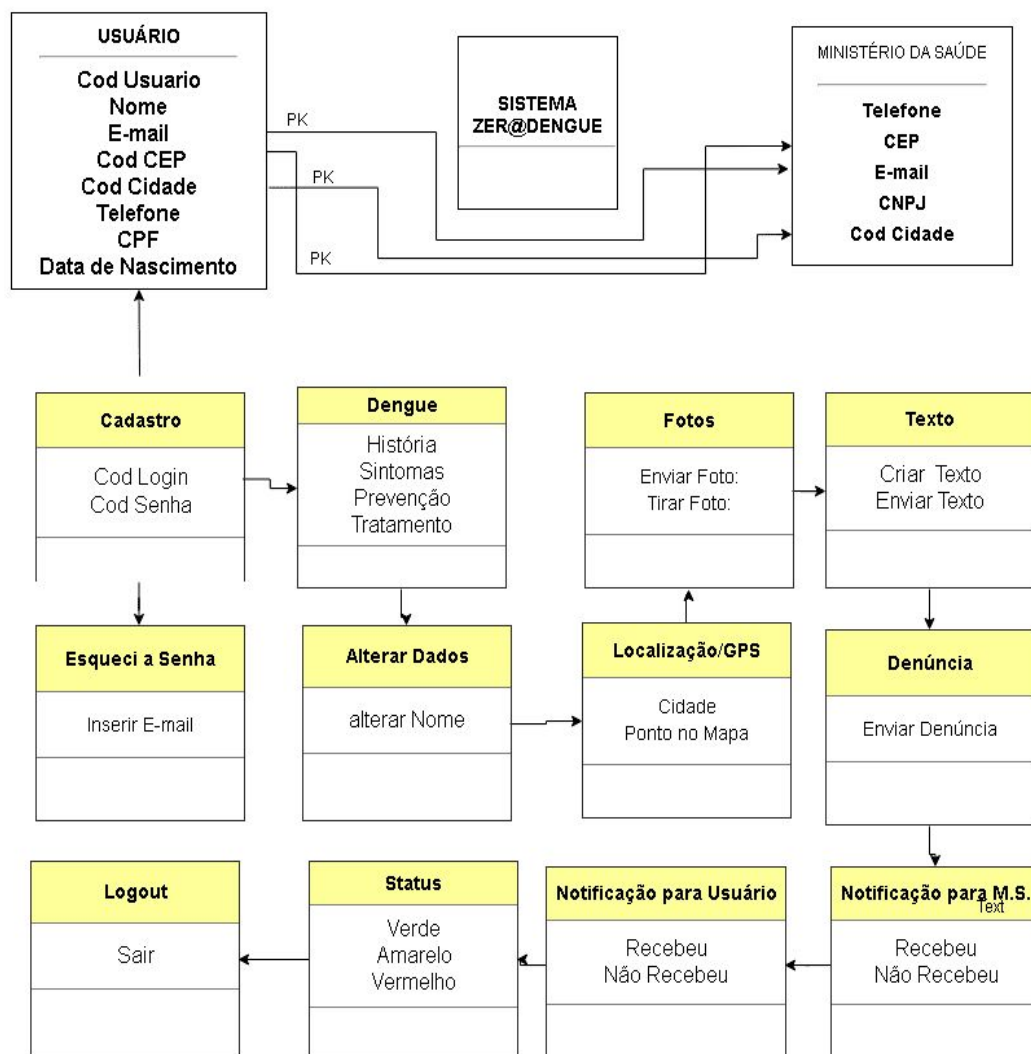
Exemplo de Cardinalidades e Relacionamentos:







## MER





1. O Sistema Zer@dengue (aplicação 1) envia as informações para o banco de dados;
2. Existe um SGBD (Sistema Gerenciador de Banco de Dados) entre o sistema Zer@dengue e o sistema do Ministério da saúde (aplicação 2);
3. O SGBD serve para proteger as informações do banco de dados, impedindo que alguém ou algo roube, altere, ou apague as mesmas indevidamente, e protege contra falhas de softwares ou hardware;
4. O sistema do Ministério da Saúde acessa os dados e informações que estão no banco de dados do sistema Zer@dengue, após ser aprovado pelo SGBD.

- Como o Ministério da Saúde usa essas informações?

Quando as informações são recebidas através das fotos dos diversos locais de foco de mosquito, e dos textos dos usuários relatando o problema, o ministério da saúde tem o conhecimento do endereço e localização do foco facilitando a intervenção.

## **2.5 - ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

Atualmente temos no mercado tecnológico alguns modelos de computadores com configurações disponíveis para a criação de um sistema. Decidimos para o Zer@dengue o seguinte modelo:

1. Tipo de processador, sua velocidade e quantidade de núcleos.

2 INTEL XEON E5-2660 V4

14 NÚCLEOS / 28 THREADS

2,00 GHZ NORMAL / 3,20 GHZ EM TURBO

2. Capacidade de memória RAM.

4 X 32GB RAM

3. Tamanho em bytes do disco rígido.

2 HD 600GB

4. Se acompanha teclado e mouse.

NÃO

5. O tipo e o tamanho de monitor.

NÃO

6. Se acompanha algum tipo de acessório de som, jogo ou de placa gráfica.

NÃO

7. O preço encontrado para esse computador.

R\$32.980

Para o estudo de dimensionamento do servidor, foram usadas simulações com o APP já em desenvolvimento completo, onde o Hardware deverá comportar uma considerável quantidade de acessos simultâneos e diários, tendo em vista a enorme quantidade de municípios distribuídos por todo o país.

Na simulação, levando em conta que um usuário ao realizar o seu cadastro e envio dos dados para o Ministério da Saúde gerará um máximo de 350 KB de dados e o mesmo receberá cerca de 20 KB de informações, sendo essas informações como: número de protocolo, previsão de data para solução, entre outras .

Assim, o servidor deverá estar pronto para um pico de demanda na fila de espera de usuários simultâneos na ordem dos 16710 usuários e de 55700 usuários diários, gerando os dados descritos na tabela abaixo.

DADOS  
IMPORTANTES  
:

MUNICIPIOS		5570	
escrita	leitura		total
(KB)	(KB)		(KB)

TRÁFEGO DE  
DADOS  
ESTIMADO P/  
USUÁRIO

50                  20                  70

### SIMULAÇÃO

	USUÁRIO S	TOTAL USUARI O	TOTAL EM ESCRITA (KB)	TOTAL EM ESCRIT A (MB)	TOTAL EM LEITURA (KB)	TOTAL EM LEITURA (MB)	TRÁFEG O GLOBAL (GB)
ACESSO SIMULTANEO P/ CIDADE	3	16710	835500	835,5	334200	334,2	1,17
ACESSO DIÁRIO P/ CIDADE MÉDIA	10	55700	278500 0	2785,0	1114000	1114,0	3,90

ARMAZENAMENT  
O DIÁRIO  
ESTIMADO (GB)  
ARMAZENAMENT  
O MENSAL  
ESTIMADO (GB)

2,79

83,55

Havendo também uma folga no processamento (28 Núcleos e 56 Threads), tendo em mente futuras implementações no App, ou até mesmo o uso combinado de Sistemas rodando no mesmo servidor.

### 3 - CONCLUSÃO

A criação de sistemas computacionais é de suma importância para a sociedade, visto que através de seus usos ocorre interação, desenvolvimento, aprendizado, descobertas e principalmente a resolução de problemas no meio social. Com os avanços tecnológicos no dia a dia, bem como as diferentes consequências de um mundo cada vez mais conectado, se observa a constante necessidade de integrar o cidadão com o mundo digital. É através do acesso à tecnologia que o indivíduo está ativo e participativo no meio que vive, melhorando a sua qualidade de vida, agregando valores e inovações a sua rotina diária.

Na elaboração do sistema Zer@dengue observamos que a praticidade de um aplicativo proporciona ao cidadão uma ferramenta de fácil acesso e manipulação para informar uma situação recorrente no nosso meio, como a presença de locais propícios a focos de dengue, as autoridades responsáveis e consequentemente facilita a fiscalização e ação nos locais denunciados.

Através da lógica computacional podemos entender um novo modo de pensar, estruturando ideias e pensamentos para a solução de um problema. É com ela que criamos um raciocínio lógico para provar ou não a veracidade de um argumento. Entretanto, observamos que existe uma grande dificuldade do indivíduo em raciocinar de forma lógica devido a não estimulação dessa habilidade desde a infância dificultando o encontro de soluções em problemas presentes no cotidiano.

Hoje tendo a tecnologia presente nas nossas atividades diárias essenciais não podemos ignorar a necessidade de estimular essa habilidade de pensar com organização de ideias para alcançar uma solução. Com a habilidade do raciocínio lógico presente podemos usá-lo de maneira que facilite o entendimento das etapas da engenharia de software, onde abrange os aspectos relacionados ao desenvolvimento e produção de um software.

Assim, com os métodos disponíveis de análise e modelagem de sistemas, as empresas podem solicitar criações de softwares conforme suas necessidades e especificações. Porém, alguns softwares para seu desenvolvimento demandam alto custo inviabilizando a sua produção. Para a criação de um software é essencial

compreender que o Algoritmo é uma sequência finita de passos que leva a execução de uma tarefa, e é desenvolvido em alguma linguagem de programação.

A linguagem que implementamos nesse trabalho foi a linguagem C++, com a estrutura de dados no conceito FIFO( first in, first out) – a informação que entra primeiro, será a primeira a sair. Com isso, a ordem de chegada das denúncias será respeitada e primeiramente deverão ser atendidas. Observamos desvantagens na programação estruturada por ainda focar em como a tarefa deve ser feita e não em o que deve ser feito. Já em relação ao banco de dados é fundamental que seja criado um cadastro de usuários para que o mesmo tenha acesso ao ambiente e o ministério da saúde tenha a possibilidade de enviar um feedback da informação recebida.

Com isso constata-se que o aprendizado obtido por meio desse trabalho, é devido ao fato de que ele proporciona uma visão mais ampla dos assuntos estudados e seus inter-relacionamentos, além da importância de cada um para a implementação de um sistema, pois a falta de conhecimento sobre um tema específico à estrutura, impactará no funcionamento adequado e isso deixará vulnerável a sua qualidade e fidelidade ao pedido do cliente. Também ensina como a tecnologia pode ser usada benéficamente pela população para promover mudanças sociais, tendo em vista que estará presente nas atividades comuns e necessárias do dia a dia.

Por fim, a relevância deste assunto para a Ti – Tecnologia da Informação – se mostra na eficácia que o sistema computacional traz para a sociedade, sendo utilizado como instrumento de inclusão social e acessibilidade, pois mobiliza a sociedade como um todo – pobres e ricos, a participarem ativamente do processo de transformação geográfica do meio ambiente em que se vive. Isso evidencia que os recursos tecnológicos são bens de consumo relevantes para promoverem mudanças sociais, atuando como serviços acessíveis a qualquer pessoa, de qualquer idade, que queiram usá-los como meio de exercer a sua responsabilidade e ética social.

#### **4 - REFERÊNCIAS BIBLIOGRÁFICAS**

WERLICH, Claudia, FEDOZZI, Regina; DA SILVA, Samuel Gonçalves. Análise e modelagem de sistemas. Londrina: Editora, Editora e Distribuidora Educacional S.A , 2020.

ANTERO, Marcio Aparecido, SCHEFFER, Vanessa Cadan. Algoritmos e lógicas de programação. Londrina: Editora e Distribuidora Educacional S.A, 2018.

SOMMERVILLE, Ian. Engenharia de Software. 10. ed. São Paulo: Editora Pearson , 2018.

VAZQUEZ, Carlos E. ; SIMÕES, Guilherme S. ; Engenharia de Requisitos : software orientado ao negócio. 1. ed. Rio de Janeiro : Brasport Livros e Multimídia Ltda , 2016.