

**UNIVERSIDADE ANHANGUERA – UNIDERP**

POLO DE (Cidade-UF)

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas- TADS**

(NOME COMPLETO DO ACADÊMICO - RA)

**PTI - PRODUÇÃO TEXTUAL INTERDISCIPLINAR**

Disciplinas Norteadoras:

**Lógica Computacional**

**Análise e Modelagem de Sistemas**

**Algoritmos e Programação Estruturada**

**Modelagem Banco de Dados**

**Arquitetura e Organização de Computadores**

Tutor (a) EAD: IVO DUCATTI

CIDADE / UF 2020

SUMÁRIO

1 – INTRODUÇÃO.....................................................................................pg. x

2 - TAREFA 1 - **Lógica Computacional**.....................................................pg. x

3 - TAREFA 2 - **Análise e Modelagem de Sistemas**..................................pg. x

4 - TAREFA 3 - **Algoritmos e Programação Estruturada**..........................pg. x

5 - TAREFA 4 - **Modelagem Banco de Dados**...........................................pg. x

6 - TAREFA 5 - **Arquitetura e Organização de Computadores**................pg. x

7 – CONCLUSÃO.......................................................................................pg. x

8 – REFERÊNCIAS BIBLIOGRÁFICAS.....................................................pg. x

- INTRODUÇÃO

Como o mundo está cada vez mais informatizado e a tecnologia está cada dia mais em alta temos a necessidade de automatizar o máximo possíveis das nossas tarefas, tornando nosso dia a dia mais simples, e assim, agilizando nossa rotina.

Esse trabalho tem como objetivo mostrar o funcionamento do sistema de denúncia Zer@Dengue, esse Software permitirá ao usuário realizar denúncias sobre focos do mosquito transmissor da dengue utilizando a internet, o objetivo do sistema é ampliar e facilitar a fiscalização.

O Software Zer@Dengue é uma ferramenta extremamente necessária nos dias atuais, tendo em vista que mais da metade da população ainda não dá a devida importantância para os focos de fecundação do mosquito, essa ferramenta vem para possibilitar que as denúncias sejam realizadas online através de celulares, tablets ou computadores.

A dengue é conhecida no Brasil desde os tempos de colônia.

O mosquito Aedes aegypti tem origem africana. Ele chegou no Brasil junto com os navios negreiros, depois de um loga viajem de seus ovos dentro dos depósitos de água das embarcações.O primeiro caso da doença foi registrado em 1685. Recife (PE).Em 1692, a dengue provocou 2000 mortes de Salvador (BA), reaparecendo em novo surto em 1972. Em 1846, o mosquito Aedes aegypti tornou-se conhecido quando uma epidemia de dengue atingiu o Rio de Janeiro, São Paulo e Salvador. Entre 1851 e 1853 e em 1916, São Paulo foi atingida por epidemias da doença.

Em 1923, Niterói, no Estado do Rio de Janeiro, lutou contra uma epidemia em sua região oceânica.Em 1903, Oswaldo Cruz, então Diretor Geral de Saúde Pública, implantou um programa de combate ao mosquito que alcançou seu auge em 1909.

Em 1957, anunciou-se que a doença estava erradicada do Brasil, embora os casos continuassem ocorrendo até 1982, quando houve uma epidemia em Roraima.

Em 1986, foram registrados epidemias nos estados do Rio de Janeiro, de Alagoas e do Ceará. Nos anos seguintes, outros Estados brasileiros foram afetados.

Em 2006, o número de casos de dengue voltou a crescer no país.

Segundo dados do Ministério da Saúde, entre janeiro e setembro de 2006 foram registrados 279.241 casos de dengue o equivalente a 1 caso(não fatal) para cada 30 km² do território desse país. Apresenta crescimento de 27% em relação ao mesmo período em 2005. A região com maior incidência foi a sudeste. Fonte:(” https://www.otempo.com.br/hotsites/luta-contra-a-dengue/conheca-a-origem-da-doenca-1.775732”)

Números atualizados:

O Brasil registrou 1.544.987 casos de dengue em 2019, um aumento de 488% em relação a 2018, segundo dados do Ministério da Saúde. Desse total, 782 pessoas morreram em todo o país.

Juntando todos os casos de dengue, zika e chikungunya, houve um aumento de 248% no registro das doenças transmitidas pelo mosquito do Aedes aegypti em 2019.

Fonte: (“ https://g1.globo.com/ciencia-e-saude/noticia/2020/01/13/brasil-teve-aumento-de-488percent-nos-casos-de-dengue-em-2019.ghtml”)

Portanto, o nosso sistema será promissor e terá muito trabalho a fazer.

2 - **Lógica Computacional**

Para calcular o total de pessoas que participaram da entrevista é necessário realizar os cálculos dos conjuntos (A para a função Cadastro e B para a função Denúncia).

Em uma entrevista de emprego sobre a preferência de determinada função obtivemos as seguintes repostas: 250 pessoas responderam que preferem atuar na área de cadastro, 180 pessoas responderam que preferem exercer a função de denuncia, 80 pessoas disseram que tanto faz pois ambas funções são de sua preferência e 50 pessoas disseram que não gostam de nenhuma das funções. Com base nas informações obtidas determine:

1. Quantas pessoas participaram dessa entrevista?
2. Quantas pessoas gostam apenas da função denúcia?

2.1 - Resolução:

.170 pessoas gostaram da função “Cadastro.”

.100 pessoas gostaram da função “Denúncia.”

. 80 pessoas gostaram das duas funções.

. 50 pessoas não gostaram de nenhuma das funções.

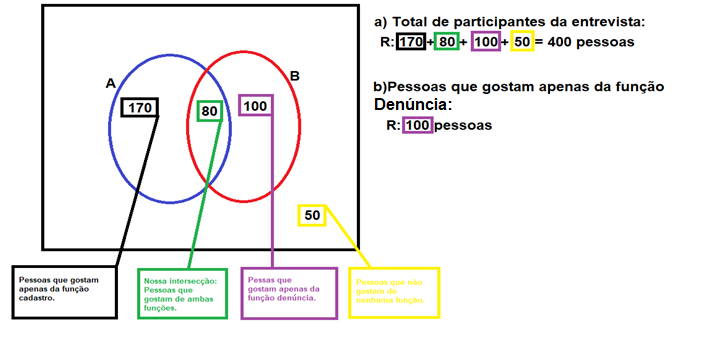
Com base nas informações obtidas determine:

2.1.1 - Quantas pessoas participaram dessa entrevista?

R: 400 pessoas participaram dessa entrevista.

* + 1. Quantas pessoas gostam apenas da função denúcia?

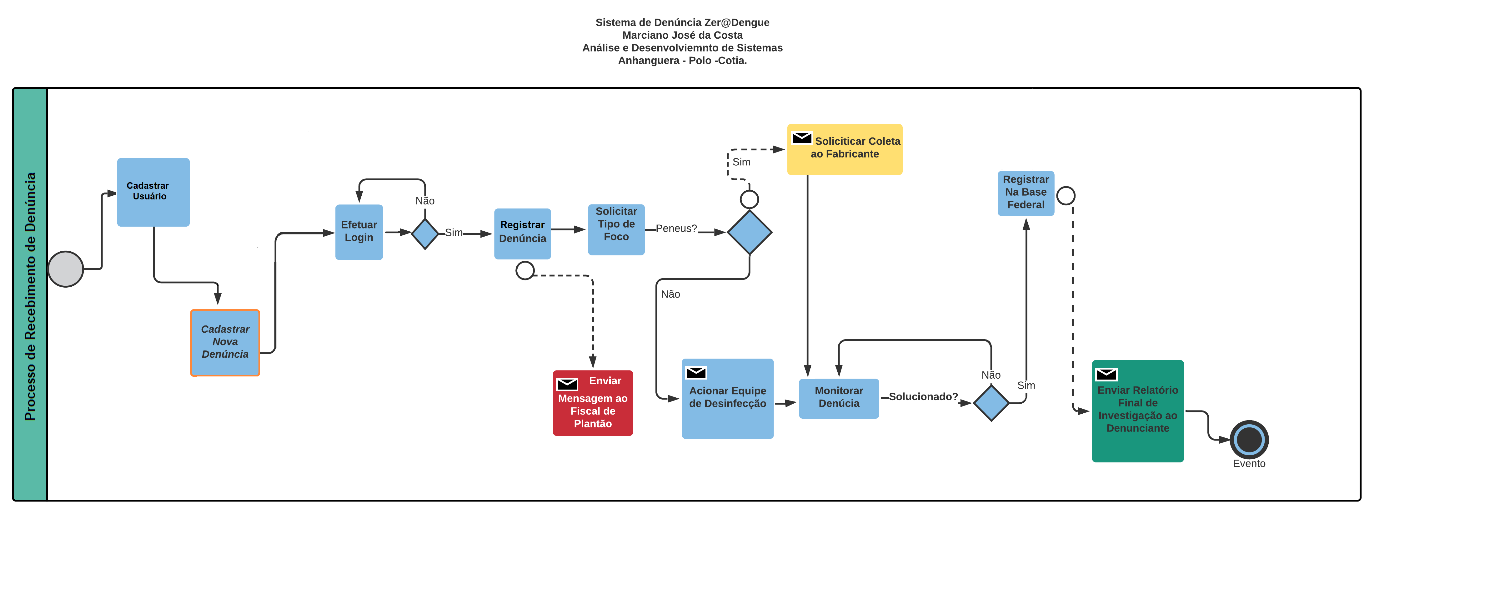
R: 100 pessoas gostam apenas da função denúncia.



3 - **Análise e Modelagem de Sistemas**

**3.1 - Modelo de Processos de Negócio do tipo fluxo de trabalho:**

O aluno deve apresentar o Diagrama de Processos de Negócio (BPD) ou um Fluxograma no formato Linear ou Funcional;



**3.2 - Documentação de todos os processos de negócio (primários, suporte e gerenciamento), descrevendo a Política de Negócio e o Procedimento Operacional Padrão (POP):**

O aluno deve apresentar a **Descrição da Política de Negócio** e o **Procedimento Operacional Padrão** de cada processo de negócio;

O Zer@Dengue é um Software que se compromete a controlar com eficiência a proliferação

do mosquito da “Dengue”, estabelecendo um controle eficiente e dinâmico do atendimento

das denúncias, sinalizando os desrespeitos, ampliando a fiscalização sanitária

epidemiológica, mapeando as zonas de riscos, sinalizando as possíveis ações de limpeza dos

focos, simplificando o atendimento e o acompanhamento da população atingida.

Viemos para fortalecer a proteção contra os riscos da dengue, prevenindo o contágio e a

reprodução do mosquito transmissor;

Através do apoio as políticas públicas, cumprindo os compromissos assumidos e definindo as

ações para que a população tenha voz ativa na participação do combate aos focos da Dengue.

O principal foco é o tornar o ambiente social mais sustentável, livre de riscos, garantido os direitos fundamentais da pessoa humana.

Buscamos a evolução continua do desempenho, incorporando e ouvindo todos os envolvidos

nos assuntos relacionados a saúde pública no combate a Dengue.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sistema de**  **Padronização** | **Procedimento Operacional Padrão** | | | | |  |
| **Número:**  **01** | **Localizador:** | **Revisão:** 1 | | **Folha: 1** |
| **Título: Realizar Denuncia** | | | | | |
| **Responsável:** Usuário Cadastrado | | | | **Data: 29/05/20** | |
| 1. **OBJETIVOS**   Realizar denúncias de Focos do mosquito transmissor da dengue através do Sistema de Denúncia Zer@Dengue. | | | | | | |
| **2. ALCANCE**  Esse Sistema se destina ao atendimento de todo cidadão em nível nacional, abrangendo os Estado e Municípios, segmentado por bairros e ruas. | | | | | | |
| 1. **MATERIAL NECESSÁRIO**   Para acessar o sistema o cidadão necessita de acesso à internet por um dispositivo móvel (celular, tablet), por um note book ou computador pessoal; | | | | | | |
| 1. **PASSOS** | | | | | | |
| 1. Acessar o Sistema via internet | | | | | | |
| 1. Realizar o cadastro de novo usuário no sistema informando dados pessoais e endereço | | | | | | |
| 1. Clicar em efetuar login | | | | | | |
| 1. Informar descrição da denuncia, clicar em Registrar Denúncia | | | | | | |
| 1. Clicar em Informar tipo do foco | | | | | | |
| 1. Sair do sistema | | | | | | |
| 1. **MANUSEIO DO MATERIAL**   1. Após Cadastrar denúncia, finalizar a seção no sistema.  2. Após sair do Sistema, e fechar a tela. | | | | | | |
| 1. **RESULTADOS ESPERADOS**   1. Realizar denuncia e cadastrar no sistema.  2. Informar empresa de coleta de pneus.  3. Acionar fiscal de plantão.  4.Registrar denúncia na base de dados do Governo Federal.  5. Realizar o acompanhamento das denúncias.  6. Informar o denunciate das ações tomadas. | | | | | | |

**3.3 - Documentação (listagem, descrição e especificação) dos requisitos funcionais e não-funcionais:**

O aluno deve apresentar uma relação (formato tabela) com a listagem e uma breve descrição dos requisitos funcionais e não-funcionais identificados no sistema;

**3.3.1 - Requisitos funcionais:**

Os requisitos funcionais descrevem as funcionalidades que cada tipo de usuário

deve ser capaz de acessar no sistema (SOMMERVILLE, 2011). Esses requisitos estão

relacionados às atividades que o sistema realiza.



**3.3.2 - Requisitos não-funcionais:**

Os requisitos não-funcionais são aqueles que não se relacionam diretamente com

as funções do sistema; no entanto, são restrições impostas aos serviços oferecidos pelo

sistema (SOMMERVILLE, 2011).



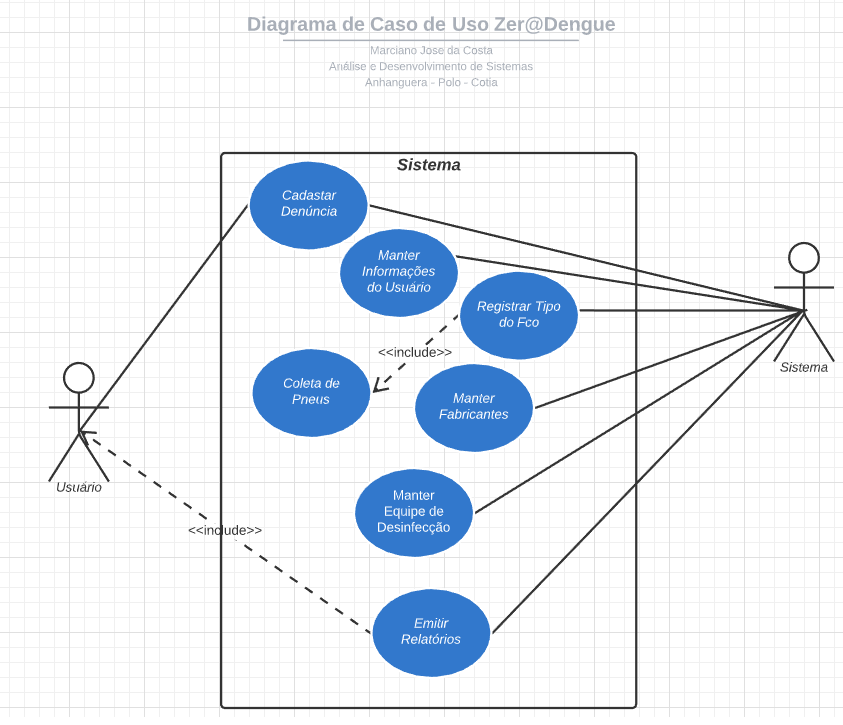
**3.4 - Diagrama de *Use Cases:***

O aluno deve apresentar o Diagrama de Use Cases consistente com a listagem dos **requisitos funcionais;**

**Diagrama de Casos de Uso:**

Considerando os requisitos descritos anteriormente, é possível modelar as funcionalidades do sistema proposto.

Para isso, é apresentado o diagrama de casos de uso do sistema com os dois atores envolvidos (Usuário, Sistema).



4 - **Algoritmos e Programação Estruturada**

#include <stdio.h>

#include <stdlib.h>

int cabec(){

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

printf("Marciano Jose da Costa \n");

printf("Analise e Desenvolvimento de Sistemas - 2020 \n");

printf("Anhanguera - Polo - Cotia\n");

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

printf("\CONTROLE DE SOLICITACAO DE USUARIOS ZER@DENGUE\n");

printf("-----------------------------------------------------------------------------------------\n");

return(0);

}

struct Atendimento{

int num;

struct Atendimento \*prox;

};

typedef struct Atendimento Atendimento;

int tam;

int menu(void);

void opcao(Atendimento \*FILA, int op);

void inicia(Atendimento \*FILA);

int vazia(Atendimento \*FILA);

Atendimento \*aloca();

void insere(Atendimento \*FILA);

Atendimento \*retira(Atendimento \*FILA);

void exibe(Atendimento \*FILA);

void libera(Atendimento \*FILA);

int main(void)

{

Atendimento \*FILA = (Atendimento \*) malloc(sizeof(Atendimento));

if(!FILA){

printf("Sem memoria disponivel !\n");

exit(1);

}else{

inicia(FILA);

int opt;

do{

opt=menu();

opcao(FILA,opt);

}while(opt);

free(FILA);

return 0;

}

}

int menu(void)

{

cabec();

int opt;

printf("Escolha a opcao\n");

printf("1. Adicionar Nova Solicitacao de Atendimento\n");

printf("2. Exibir Solicitacoes\n");

printf("3. Excluir todas Solicitacoes \n");

printf("4. Retirar Solicitacao de Atendimento\n");

printf("0. Sair\n");

printf("Opcao: "); scanf("%d", &opt);

return opt;

}

void opcao(Atendimento \*FILA, int op)

{

Atendimento \*tmp;

switch(op){

case 0:

libera(FILA);

break;

case 1:

insere(FILA);

break;

case 2:

exibe(FILA);

break;

case 3:

libera(FILA);

inicia(FILA);

break;

case 4:

tmp= retira(FILA);

if(tmp != NULL){

printf("Atendimento Finalizado: %3d\n\n", tmp->num);

libera(tmp);

}

break;

default:

printf("Comando invalido\n\n");

}

}

void inicia(Atendimento \*FILA)

{

FILA->prox = NULL;

tam=0;

}

int vazia(Atendimento \*FILA)

{

if(FILA->prox == NULL)

return 1;

else

return 0;

}

Atendimento \*aloca()

{

Atendimento \*novo=(Atendimento \*) malloc(sizeof(Atendimento));

if(!novo){

printf("Sem memoria disponivel!\n");

exit(1);

}else{

printf("Novo Atendimento: "); scanf("%d", &novo->num);

return novo;

}

}

void insere(Atendimento \*FILA)

{

Atendimento \*novo=aloca();

novo->prox = NULL;

if(vazia(FILA))

FILA->prox=novo;

else{

Atendimento \*tmp = FILA->prox;

while(tmp->prox != NULL)

tmp = tmp->prox;

tmp->prox = novo;

}

tam++;

}

Atendimento \*retira(Atendimento \*FILA)

{

if(FILA->prox == NULL){

printf("Fila ja esta vazia\n");

return NULL;

}else{

Atendimento \*tmp = FILA->prox;

FILA->prox = tmp->prox;

tam--;

return tmp;

}

}

void exibe(Atendimento \*FILA)

{

if(vazia(FILA)){

printf("Nenhuma Solicitacao de Atendimento!\n\n");

return ;

}

Atendimento \*tmp;

tmp = FILA->prox;

printf("Fila :");

while( tmp != NULL){

printf("%5d", tmp->num);

tmp = tmp->prox;

}

printf("\n ");

int count;

for(count=0 ; count < tam ; count++)

printf(" ^ ");

printf("\nOrdem:");

for(count=0 ; count < tam ; count++)

printf("%5d", count+1);

printf("\n\n");

}

void libera(Atendimento \*FILA)

{

if(!vazia(FILA)){

Atendimento \*proxAtendimento,

\*atual;

atual = FILA->prox;

while(atual != NULL){

proxAtendimento = atual->prox;

free(atual);

atual = proxAtendimento;

}

}

}

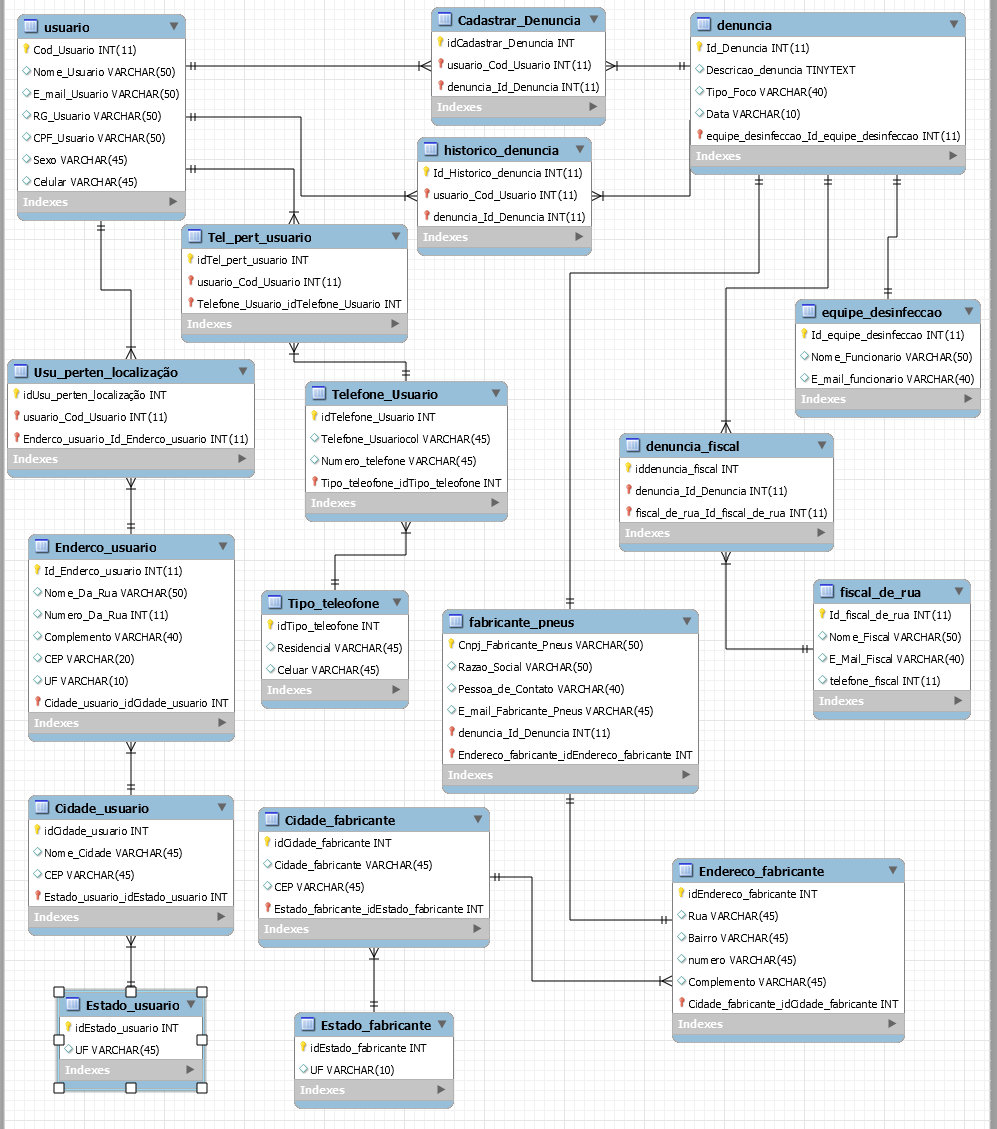
5 - **Modelagem Banco de Dados**

O aluno deve apresentar a modelagem do banco de dados do projeto por meio de uma representação gráfica (MER);

5.1 - O MER foi desenvolvido através da ferramenta Case MySQL Workbench:

Dada a natureza da aplicação desenvolvida, é necessário implementar um banco

de dados para armazenamento de informações sobre os usuários do sistema, das denúncias cadastradas, dos fabricantes de pneus, da equipe de desinfecção e dos fiscais de rua.



5.2 - Script em SQL: Tabelas em conformidade com o MER.

O desenvolvimento do banco de dados e a estruturação das tabelas foram feitas utilizando o Software MySQL WorkBench:

CREATE DATABASE IF NOT EXISTS `zerardengue` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `zerardengue`;

-- MySQL dump 10.13 Distrib 5.6.23, for Win32 (x86)

--

-- Host: localhost Database: zerardengue

-- ------------------------------------------------------

-- Server version 5.5.5-10.4.10-MariaDB

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `cadastrar\_denuncia`

--

DROP TABLE IF EXISTS `cadastrar\_denuncia`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `cadastrar\_denuncia` (

`idCadastrar\_Denuncia` int(11) NOT NULL,

`usuario\_Cod\_Usuario` int(11) NOT NULL,

`denuncia\_Id\_Denuncia` int(11) NOT NULL,

PRIMARY KEY (`idCadastrar\_Denuncia`,`usuario\_Cod\_Usuario`,`denuncia\_Id\_Denuncia`),

KEY `fk\_Cadastrar\_Denuncia\_usuario\_idx` (`usuario\_Cod\_Usuario`),

KEY `fk\_Cadastrar\_Denuncia\_denuncia1\_idx` (`denuncia\_Id\_Denuncia`),

CONSTRAINT `fk\_Cadastrar\_Denuncia\_denuncia1` FOREIGN KEY (`denuncia\_Id\_Denuncia`) REFERENCES `zeradengue`.`denuncia` (`Id\_Denuncia`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Cadastrar\_Denuncia\_usuario` FOREIGN KEY (`usuario\_Cod\_Usuario`) REFERENCES `zeradengue`.`usuario` (`Cod\_Usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `cadastrar\_denuncia`

--

LOCK TABLES `cadastrar\_denuncia` WRITE;

/\*!40000 ALTER TABLE `cadastrar\_denuncia` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `cadastrar\_denuncia` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `cidade\_fabricante`

--

DROP TABLE IF EXISTS `cidade\_fabricante`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `cidade\_fabricante` (

`idCidade\_fabricante` int(11) NOT NULL,

`Cidade\_fabricante` varchar(45) DEFAULT NULL,

`CEP` varchar(45) DEFAULT NULL,

`Estado\_fabricante\_idEstado\_fabricante` int(11) NOT NULL,

PRIMARY KEY (`idCidade\_fabricante`,`Estado\_fabricante\_idEstado\_fabricante`),

KEY `fk\_Cidade\_fabricante\_Estado\_fabricante1\_idx` (`Estado\_fabricante\_idEstado\_fabricante`),

CONSTRAINT `fk\_Cidade\_fabricante\_Estado\_fabricante1` FOREIGN KEY (`Estado\_fabricante\_idEstado\_fabricante`) REFERENCES `estado\_fabricante` (`idEstado\_fabricante`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `cidade\_fabricante`

--

LOCK TABLES `cidade\_fabricante` WRITE;

/\*!40000 ALTER TABLE `cidade\_fabricante` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `cidade\_fabricante` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `cidade\_usuario`

--

DROP TABLE IF EXISTS `cidade\_usuario`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `cidade\_usuario` (

`idCidade\_usuario` int(11) NOT NULL,

`Nome\_Cidade` varchar(45) DEFAULT NULL,

`CEP` varchar(45) DEFAULT NULL,

`Estado\_usuario\_idEstado\_usuario` int(11) NOT NULL,

PRIMARY KEY (`idCidade\_usuario`,`Estado\_usuario\_idEstado\_usuario`),

KEY `fk\_Cidade\_usuario\_Estado\_usuario1\_idx` (`Estado\_usuario\_idEstado\_usuario`),

CONSTRAINT `fk\_Cidade\_usuario\_Estado\_usuario1` FOREIGN KEY (`Estado\_usuario\_idEstado\_usuario`) REFERENCES `estado\_usuario` (`idEstado\_usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `cidade\_usuario`

--

LOCK TABLES `cidade\_usuario` WRITE;

/\*!40000 ALTER TABLE `cidade\_usuario` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `cidade\_usuario` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `denuncia\_fiscal`

--

DROP TABLE IF EXISTS `denuncia\_fiscal`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `denuncia\_fiscal` (

`iddenuncia\_fiscal` int(11) NOT NULL,

`denuncia\_Id\_Denuncia` int(11) NOT NULL,

`fiscal\_de\_rua\_Id\_fiscal\_de\_rua` int(11) NOT NULL,

PRIMARY KEY (`iddenuncia\_fiscal`,`denuncia\_Id\_Denuncia`,`fiscal\_de\_rua\_Id\_fiscal\_de\_rua`),

KEY `fk\_denuncia\_fiscal\_denuncia1\_idx` (`denuncia\_Id\_Denuncia`),

KEY `fk\_denuncia\_fiscal\_fiscal\_de\_rua1\_idx` (`fiscal\_de\_rua\_Id\_fiscal\_de\_rua`),

CONSTRAINT `fk\_denuncia\_fiscal\_denuncia1` FOREIGN KEY (`denuncia\_Id\_Denuncia`) REFERENCES `zeradengue`.`denuncia` (`Id\_Denuncia`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_denuncia\_fiscal\_fiscal\_de\_rua1` FOREIGN KEY (`fiscal\_de\_rua\_Id\_fiscal\_de\_rua`) REFERENCES `zeradengue`.`fiscal\_de\_rua` (`Id\_fiscal\_de\_rua`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `denuncia\_fiscal`

--

LOCK TABLES `denuncia\_fiscal` WRITE;

/\*!40000 ALTER TABLE `denuncia\_fiscal` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `denuncia\_fiscal` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `endereco\_fabricante`

--

DROP TABLE IF EXISTS `endereco\_fabricante`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `endereco\_fabricante` (

`idEndereco\_fabricante` int(11) NOT NULL,

`Rua` varchar(45) DEFAULT NULL,

`Bairro` varchar(45) DEFAULT NULL,

`numero` varchar(45) DEFAULT NULL,

`Complemento` varchar(45) DEFAULT NULL,

`Cidade\_fabricante\_idCidade\_fabricante` int(11) NOT NULL,

PRIMARY KEY (`idEndereco\_fabricante`,`Cidade\_fabricante\_idCidade\_fabricante`),

KEY `fk\_Endereco\_fabricante\_Cidade\_fabricante1\_idx` (`Cidade\_fabricante\_idCidade\_fabricante`),

CONSTRAINT `fk\_Endereco\_fabricante\_Cidade\_fabricante1` FOREIGN KEY (`Cidade\_fabricante\_idCidade\_fabricante`) REFERENCES `cidade\_fabricante` (`idCidade\_fabricante`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `endereco\_fabricante`

--

LOCK TABLES `endereco\_fabricante` WRITE;

/\*!40000 ALTER TABLE `endereco\_fabricante` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `endereco\_fabricante` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `estado\_fabricante`

--

DROP TABLE IF EXISTS `estado\_fabricante`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `estado\_fabricante` (

`idEstado\_fabricante` int(11) NOT NULL,

`UF` varchar(10) DEFAULT NULL,

PRIMARY KEY (`idEstado\_fabricante`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `estado\_fabricante`

--

LOCK TABLES `estado\_fabricante` WRITE;

/\*!40000 ALTER TABLE `estado\_fabricante` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `estado\_fabricante` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `estado\_usuario`

--

DROP TABLE IF EXISTS `estado\_usuario`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `estado\_usuario` (

`idEstado\_usuario` int(11) NOT NULL,

`UF` varchar(45) DEFAULT NULL,

PRIMARY KEY (`idEstado\_usuario`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `estado\_usuario`

--

LOCK TABLES `estado\_usuario` WRITE;

/\*!40000 ALTER TABLE `estado\_usuario` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `estado\_usuario` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tel\_pert\_usuario`

--

DROP TABLE IF EXISTS `tel\_pert\_usuario`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tel\_pert\_usuario` (

`idTel\_pert\_usuario` int(11) NOT NULL,

`usuario\_Cod\_Usuario` int(11) NOT NULL,

`Telefone\_Usuario\_idTelefone\_Usuario` int(11) NOT NULL,

PRIMARY KEY (`idTel\_pert\_usuario`,`usuario\_Cod\_Usuario`,`Telefone\_Usuario\_idTelefone\_Usuario`),

KEY `fk\_Tel\_pert\_usuario\_usuario1\_idx` (`usuario\_Cod\_Usuario`),

KEY `fk\_Tel\_pert\_usuario\_Telefone\_Usuario1\_idx` (`Telefone\_Usuario\_idTelefone\_Usuario`),

CONSTRAINT `fk\_Tel\_pert\_usuario\_Telefone\_Usuario1` FOREIGN KEY (`Telefone\_Usuario\_idTelefone\_Usuario`) REFERENCES `telefone\_usuario` (`idTelefone\_Usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Tel\_pert\_usuario\_usuario1` FOREIGN KEY (`usuario\_Cod\_Usuario`) REFERENCES `zeradengue`.`usuario` (`Cod\_Usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tel\_pert\_usuario`

--

LOCK TABLES `tel\_pert\_usuario` WRITE;

/\*!40000 ALTER TABLE `tel\_pert\_usuario` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `tel\_pert\_usuario` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `telefone\_usuario`

--

DROP TABLE IF EXISTS `telefone\_usuario`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `telefone\_usuario` (

`idTelefone\_Usuario` int(11) NOT NULL,

`Telefone\_Usuariocol` varchar(45) DEFAULT NULL,

`Numero\_telefone` varchar(45) DEFAULT NULL,

`Tipo\_teleofone\_idTipo\_teleofone` int(11) NOT NULL,

PRIMARY KEY (`idTelefone\_Usuario`,`Tipo\_teleofone\_idTipo\_teleofone`),

KEY `fk\_Telefone\_Usuario\_Tipo\_teleofone1\_idx` (`Tipo\_teleofone\_idTipo\_teleofone`),

CONSTRAINT `fk\_Telefone\_Usuario\_Tipo\_teleofone1` FOREIGN KEY (`Tipo\_teleofone\_idTipo\_teleofone`) REFERENCES `tipo\_teleofone` (`idTipo\_teleofone`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `telefone\_usuario`

--

LOCK TABLES `telefone\_usuario` WRITE;

/\*!40000 ALTER TABLE `telefone\_usuario` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `telefone\_usuario` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tipo\_teleofone`

--

DROP TABLE IF EXISTS `tipo\_teleofone`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tipo\_teleofone` (

`idTipo\_teleofone` int(11) NOT NULL,

`Residencial` varchar(45) DEFAULT NULL,

`Celuar` varchar(45) DEFAULT NULL,

PRIMARY KEY (`idTipo\_teleofone`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tipo\_teleofone`

--

LOCK TABLES `tipo\_teleofone` WRITE;

/\*!40000 ALTER TABLE `tipo\_teleofone` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `tipo\_teleofone` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `usu\_perten\_localizaÃ§Ã£o`

--

DROP TABLE IF EXISTS `usu\_perten\_localizaÃ§Ã£o`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `usu\_perten\_localizaÃ§Ã£o` (

`idUsu\_perten\_localizaÃ§Ã£o` int(11) NOT NULL,

`usuario\_Cod\_Usuario` int(11) NOT NULL,

`Enderco\_usuario\_Id\_Enderco\_usuario` int(11) NOT NULL,

PRIMARY KEY (`idUsu\_perten\_localizaÃ§Ã£o`,`usuario\_Cod\_Usuario`,`Enderco\_usuario\_Id\_Enderco\_usuario`),

KEY `fk\_Usu\_perten\_localizaÃ§Ã£o\_usuario1\_idx` (`usuario\_Cod\_Usuario`),

KEY `fk\_Usu\_perten\_localizaÃ§Ã£o\_Enderco\_usuario1\_idx` (`Enderco\_usuario\_Id\_Enderco\_usuario`),

CONSTRAINT `fk\_Usu\_perten\_localizaÃ§Ã£o\_Enderco\_usuario1` FOREIGN KEY (`Enderco\_usuario\_Id\_Enderco\_usuario`) REFERENCES `zeradengue`.`enderco\_usuario` (`Id\_Enderco\_usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Usu\_perten\_localizaÃ§Ã£o\_usuario1` FOREIGN KEY (`usuario\_Cod\_Usuario`) REFERENCES `zeradengue`.`usuario` (`Cod\_Usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `usu\_perten\_localizaÃ§Ã£o`

--

LOCK TABLES `usu\_perten\_localizaÃ§Ã£o` WRITE;

/\*!40000 ALTER TABLE `usu\_perten\_localizaÃ§Ã£o` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `usu\_perten\_localizaÃ§Ã£o` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2020-05-30 0:51:13

6 - **Arquitetura e Organização de Computadores**

Pesquisa realizada na Internet: Descrever as configurações e os valores (custos) dos equipamentos.

**6.1 - Pequisas sobre as características de 03 Note Books Apple:**



**6.1.2- Pequisas sobre as características de 03 Note Books Dell:**



**6.1.3- Pequisas sobre as características de 03 Note Books Samsung:**



**6.1.4- Pesquisa sobre processadores:**





**6.1.5 - Pesquisa sobre memórias:**



**6.1.6 - Pesquisa sobre armazenamento em SSD:**



**6.1.7 - Hard Disk:**



**6.1.8 - Placas de vídeo:**



**6.1.9 - Placa de rede:**



**7 - CONCLUSÃO**

O trabalho atingiu o objetivo principal de apresentar uma abordagem

sobre o desenvolvimento do sistema de denúncia Zer@Dengue e as metodologias

aplicadas nas fases de desnvolvimento do software, focando na resolução das

atividades conforme os enunciados apresentados e as disciplinas estudadas

durante o semestre.

Com a conclusão deste, fica claro como esse assunto é dinâmico, o que demonstra

a real necessidade de se ter uma ferramenta para auxiliar nas denúncias apliando a

fiscalização e o controle dos focos de proliferação do mosquito Aedes aegypti.

**8 - REFERÊNCIAS BIBLIOGRÁFICAS**

THIAGO PINHEIRO FELIX DA SILVA E LIMA, Lógica Computacional - Editora e Distribuidora Educacional S.A, 2020.

JUCILANDIO DIAS DE SOUZA, O Raciocínio Lógico Argumentativo - Clube de Autores, 2012.

CLAUDIA WERLICH, REGINA FEDOZZI SAMUEL GONÇALVES DA SILVA, Análise e Modelagem de Sistemas - Editora e Distribuidora Educacional S.A, 2020.

ROGER S. PRESSMAN, PD. D, Engenharia de Software, 7ª Edição – Amgh Editora Ltda, 2011.

SOFTEX, Melhoria do Processo de Software Brasileiro, 2020.

IAN SOMMERVILLE, Engenharia de Software, 9ª Edição – Pearson Education do Brasil, 2011.

AARON AT TENENBAUM, YEDIDYAHN LANGSAN, MOSHE J. AUGENSTEIN, Estruturas de Dados Usando C, MAKRON Books do Brasil Editora Ltda., 1995.

KLEBER RICARDI ROVALI, MARCIO APARECIDO ARTERO, VANESSA CANDAN SCHEFFER, Algoritmo e Programação Estruturada – Editora Educacional , 2020.

HEBERT SCHILDT, C Completo e total Terceira Edição, Makron Books,1997.

DR. ANDRE BACKES, Linguagem C Programação Descomplicada - <https://www.youtube.com/watch?v=GiCt0Cwcp-U>;

PROFESSOR GUSTAVO GUANABAR, Algoritmo e lógica de Programação - <https://www.youtube.com/watch?v=8mei6uVttho>;

CLAUDIA WERLICH, Modelagem de Dados, Edutora Educacional,2018.

CARLOS ALBERTO HEUSER, Projeto de banco de Dados, Digital Source 1998

LEONARDO GUIMARÃES TANGON, ROGERIO CARLOS DOS SANTOS Arquitetura e organização de computadores, Editora e Distribuidora Educacional S.A., 2016.