



**UNIVERSIDADE ANHANGUERA – UNIDERP**

POLO DE (Araraquara-SP)

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

(Jonas da Silva Pinto - 2470056201)

**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

Araraquara / SP 2020

## SUMÁRIO

<b>Introdução.....</b>	<b>pg. 3</b>
<b>Lógica Computacional.....</b>	<b>pg. 4</b>
<b>Análise e Modelagem de Sistemas.....</b>	<b>pg. 5 - 7</b>
<b>Algoritmos e Programação Estruturada.....</b>	<b>pg. 8 - 10</b>
<b>Modelagem Banco de Dados.....</b>	<b>pg. 11 - 14</b>
<b>Arquitetura e Organização de Computadores.....</b>	<b>pg. 15 - 16</b>
<b>Conclusão.....</b>	<b>pg. 17</b>
<b>Referencias bibliograficas.....</b>	<b>pg. 18</b>

## INTRODUÇÃO

A tecnologia evoluiu muito ao longo do tempo, trouxe soluções para diversos problemas em praticamente todas as áreas onde o homem atua.

Usar a tecnologia se tornou indispensável. A tecnologia além das características mencionadas, exerce a função de satisfazer necessidades dentre seus mais variados usuários.

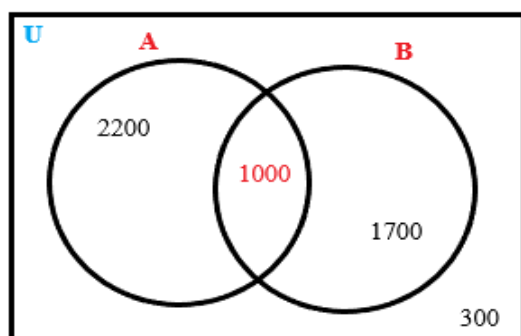
Dentro deste cenário tornou-se viável e indispensável o desenvolvimento de sistemas, aplicativos ou softwares para resolver problemas como por exemplo o de saúde como é proposto para este trabalho.

O tema escolhido (zer@dengue) faz menção ao problema da dengue, doença causada pelo mosquito que já causou muitas mortes e problemas de saúde, *aedes aegypti* é o nome deste mosquito transmissor da dengue, Zika e Chikungunya, doenças que podem gerar outras enfermidades como microcefalia e Guillain-Barré.

Tendo como base o uso da tecnologia para a solução do problema ao qual o tema refere-se, torna-se, portanto, importante e relevante no sentido de trazer conhecimento no âmbito acadêmico como também uma visão de responsabilidade que cada um pode exercer para com a sociedade.

A partir destas considerações, visa-se aplicar os conhecimentos adquiridos e chegar a uma solução satisfatória ao problema proposto.

## Lógica Computacional



Como pode ser visto acima, um conjunto universo contendo dois subconjuntos denominados (A e B), cadastro e denuncia respectivamente, uma intersecção entre A e B formada por pessoas que gostaram das duas funções, dentre outras informações:

300    pessoas não gostaram de nenhuma das duas funções

1000    pessoas gostaram das duas funções

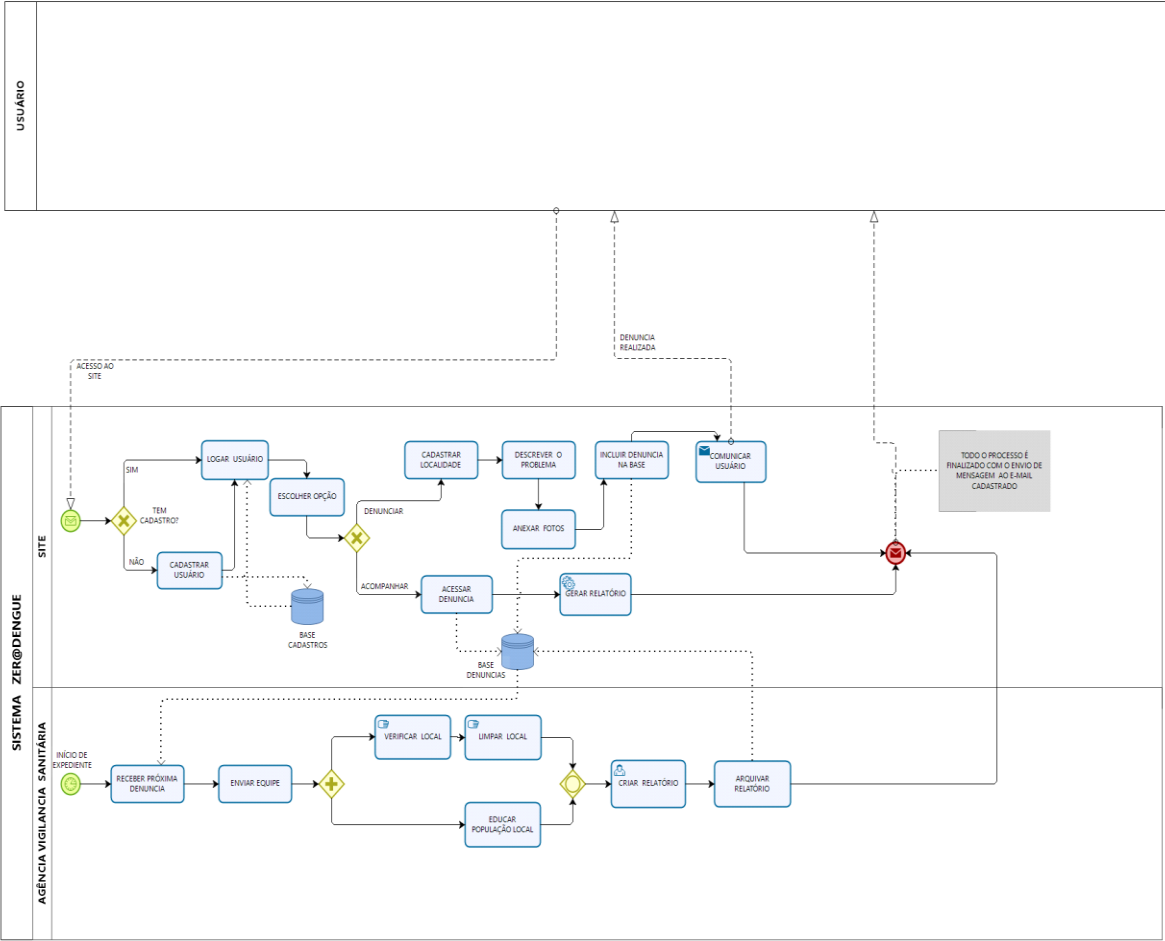
2200    pessoas **gostaram da função cadastro**

1700    pessoas gostaram da função denuncia

52000 pessoas **participaram dessa entrevista**

Análise e Modelagem de Sistemas

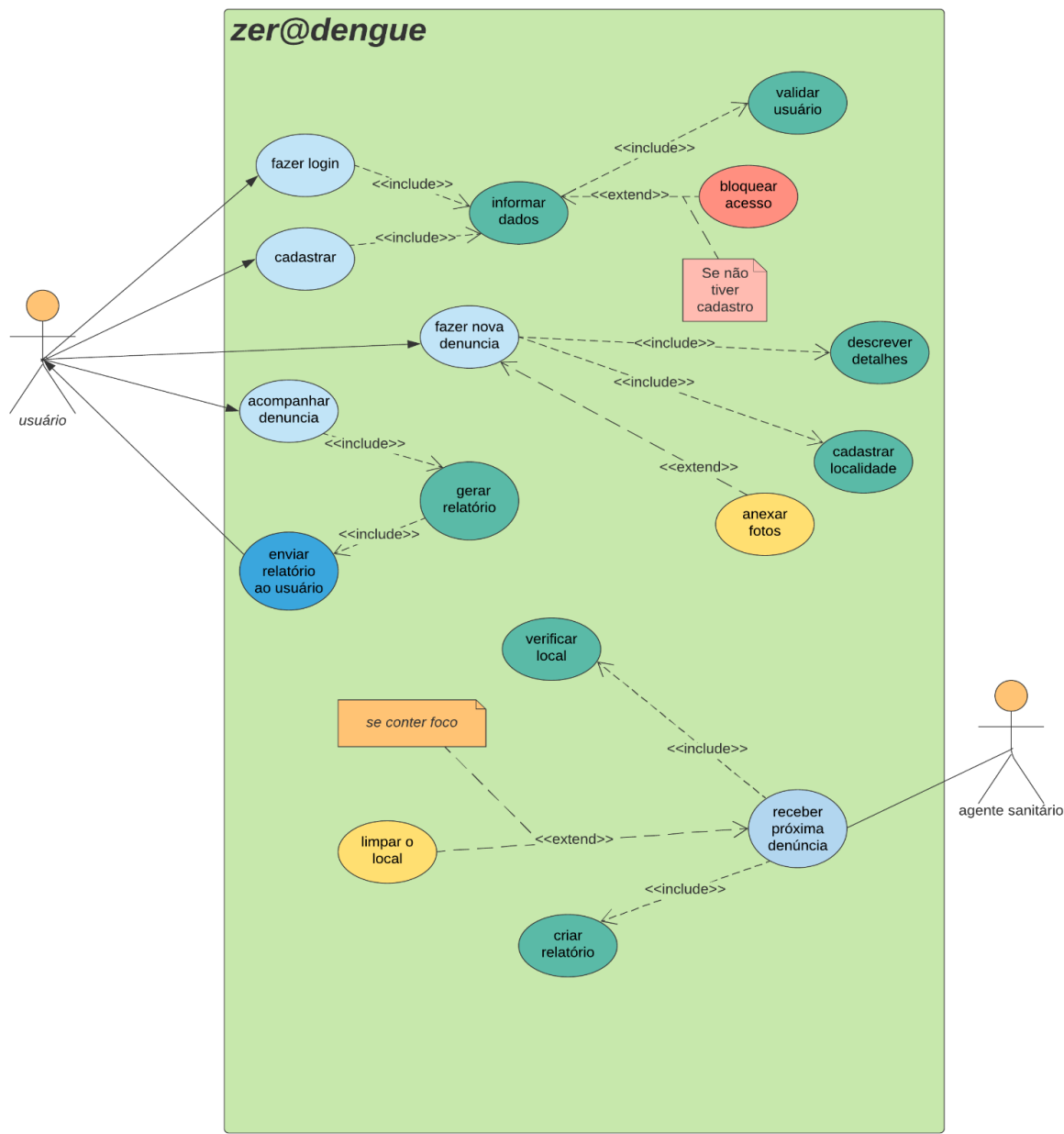
BPMN



**listagem, descrição e especificação dos requisitos funcionais e não funcionais**

REQUISITOS		FUNCIONAL	NÃO FUNCIONAL
<b>R001</b>	O sistema deve prover ao usuário a opção de cadastro	X	
<b>R002</b>	O sistema deve prover ao usuário a opção de login	X	
<b>R003</b>	O usuário apenas poderá efetuar login se estiver devidamente cadastrado		X
<b>R004</b>	O sistema deve disponibilizar a opção <i>fazer nova denúncia</i>	X	
<b>R005</b>	O sistema deve disponibilizar a opção <i>acompanhar denúncia</i>	X	
<b>R006</b>	O sistema deve possuir uma base de dados para armazenar cadastros e denúncias		X
<b>R007</b>	O sistema deve gerar relatório ao solicitar acompanhamento de denúncia	X	
<b>R008</b>	O sistema deverá enviar relatório atualizado com data hora e status ao email cadastrado pelo usuário solicitante		X
<b>R009</b>	O sistema deve prover fácil usabilidade e ser intuitivo		X
<b>R010</b>	O sistema deve ter a opção para cadastrar e armazenar (localidade, descrição e fotos) da nova denúncia	X	
<b>R011</b>	O sistema deve emitir mensagem de cadastramento ( <i>efetuado com sucesso</i> ) ao finalizar uma nova denúncia		X

Diagrama de Use Cases (casos de uso)



## Algoritmos e Programação Estruturada

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Node
```

```
{
```

```
    int dado;
```

```
    struct Node *next;
```

```
};
```

```
void insere_inicio(struct Node** inicio_ref, int new_dado)
```

```
{
```

```
    struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
```

```
    new_node->dado = new_dado;
```

```
    new_node->next = (*inicio_ref);
```

```
    (*inicio_ref) = new_node;
```

```
}
```

```
void insertAfter(struct Node* prev_node, int new_dado)
```

```
{
```

```
    if (prev_node == NULL)
```

```
    {
```

```
        printf("the given previous node cannot be NULL");
```

```
        return;
```

```
    }
```



```
struct Node* new_node =(struct Node*) malloc(sizeof(struct Node));
```

```
new_node->dado = new_dado;
```

```
new_node->next = prev_node->next;
```

```
prev_node->next = new_node;
```

```
}
```

```
void insere_fim(struct Node** inicio_ref, int new_dado)
```

```
{
```

```
    struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
```

```
    struct Node *ultimo = *inicio_ref;
```

```
    new_node->dado = new_dado;
```

```
    new_node->next = NULL;
```

```
    if (*inicio_ref == NULL)
```

```
    {
```

```
        *inicio_ref = new_node;
```

```
        return;
```

```
    }
```

```
    while (ultimo->next != NULL)
```

```
        ultimo = ultimo->next;
```

```
    ultimo->next = new_node;
```

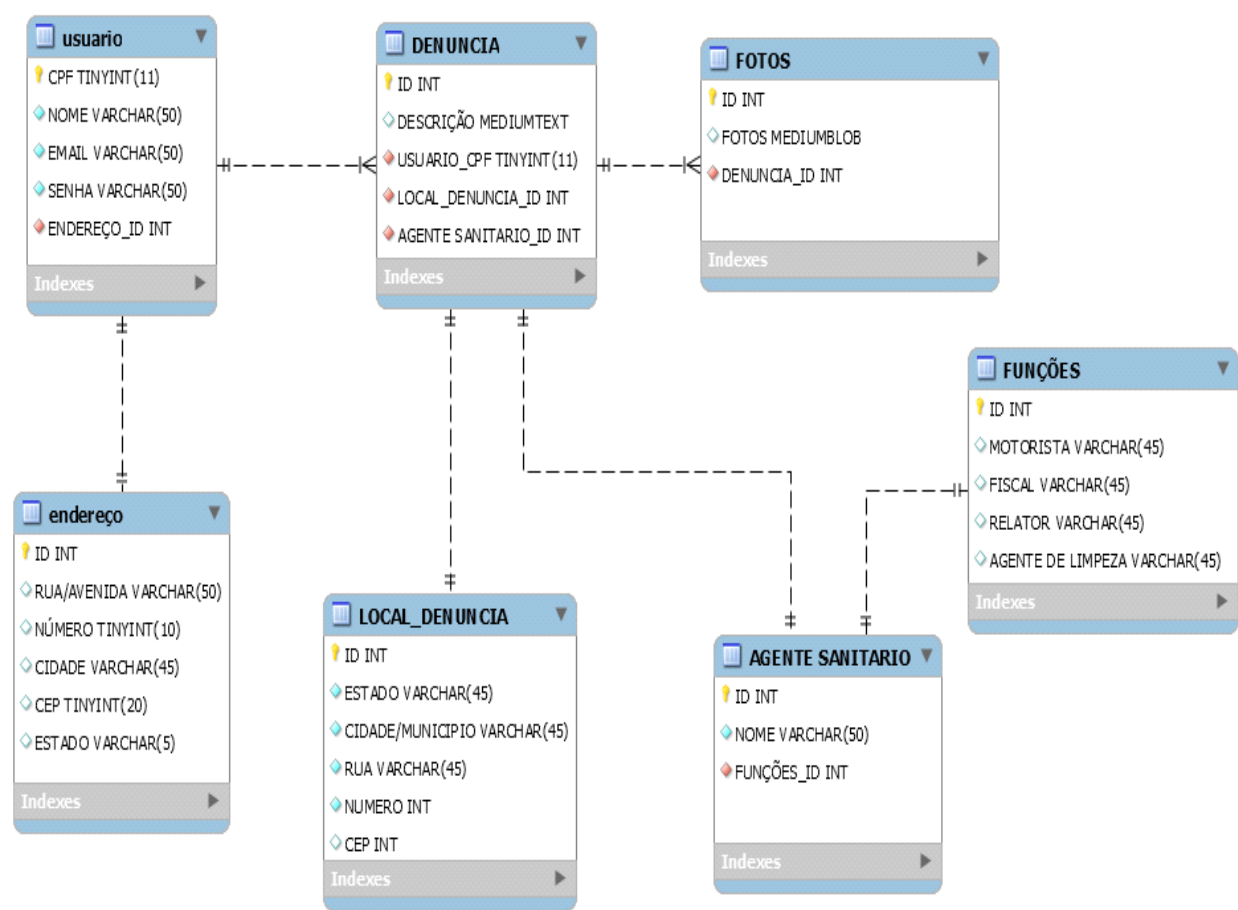
```
    return;  
}
```

```
void printList(struct Node *node)  
{  
    while (node != NULL)  
    {  
        printf(" %d ", node->dado);  
        node = node->next;  
    }  
}
```

```
int main()  
{  
    struct Node* inicio = NULL;  
  
    insere_fim(&inicio, 6);  
  
    insere_inicio(&inicio, 7);  
  
    insere_inicio(&inicio, 1);  
  
    insere_fim(&inicio, 4);  
  
    insertAfter(inicio->next, 8);  
  
    printf("\n Lista Criada: ");  
    printList(inicio);  
  
    return 0; }
```

# Modelagem Banco de Dados

## DER



## SCRIPT SQL

Schema zer@dengue

```
CREATE SCHEMA IF NOT EXISTS `zer@dengue` DEFAULT CHARACTER
SET utf8;
USE `zer@dengue`;
```

Table `zer@dengue`.`agente sanitario`

```
CREATE TABLE IF NOT EXISTS `zer@dengue`.`agente sanitario` (
  `ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `NOME` VARCHAR (45) NOT NULL,
  `funções_ID` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`ID`),
  INDEX `fk_agente sanitario_funções1_idx` (`funções_ID` ASC) VISIBLE,
  CONSTRAINT `fk_agente sanitario_funções1`
    FOREIGN KEY (`funções_ID`)
    REFERENCES `zer@dengue`.`funções` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

Table `zer@dengue`.`denúncia`

```
CREATE TABLE IF NOT EXISTS `zer@dengue`.`denúncia` (
  `ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `DESCRICAO` MEDIUMTEXT NULL DEFAULT NULL,
  `usuario_cpf` TINYINT UNSIGNED NOT NULL,
  `local_denuncia_ID` INT UNSIGNED NOT NULL,
  `agente sanitario_ID` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`ID`),
  INDEX `fk_denuncia_usuario_idx` (`usuario_cpf` ASC) VISIBLE,
  INDEX `fk_denuncia_local_denuncia1_idx` (`local_denuncia_ID` ASC)
  VISIBLE,
  INDEX `fk_denuncia_agente sanitario1_idx` (`agente sanitario_ID` ASC)
  VISIBLE,
  CONSTRAINT `fk_denúncia_usuario`
    FOREIGN KEY (`usuario_cpf`)
    REFERENCES `zer@dengue`.`usuário` (`cpf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_denuncia_local_denuncia1`
    FOREIGN KEY (`local_denúncia_ID`)
    REFERENCES `zer@dengue`.`local_denúncia` (`ID`)
```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_denúncia_agente sanitario1`
FOREIGN KEY (`agente sanitario_ID`)
REFERENCES `zer@dengue`.`agente sanitario` (`ID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

Table `zer@dengue`.`endereço`

```

CREATE TABLE IF NOT EXISTS `zer@dengue`.`endereço` (
  `ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `RUA/AVENIDA` VARCHAR(45) NOT NULL,
  `NÚMERO` TINYINT NOT NULL,
  `CIDADE` VARCHAR (45) NOT NULL,
  `CEP` TINYINT UNSIGNED NOT NULL,
  `ESTADO` VARCHAR (5) NOT NULL,
  PRIMARY KEY (`ID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

Table `zer@dengue`.`fotos`

```

CREATE TABLE IF NOT EXISTS `zer@dengue`.`fotos` (
  `ID` INT NULL DEFAULT NULL,
  `FOTOS` MEDIUMBLOB NULL DEFAULT NULL,
  `denúncia ID` INT UNSIGNED NOT NULL,
  CONSTRAINT `fk_fotos_denuncia1`
FOREIGN KEY (`denuncia_ID`)
REFERENCES `zer@dengue`.`denúncia` (`ID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

Table `zer@dengue`.`funções`

```

CREATE TABLE IF NOT EXISTS `zer@dengue`.`funções` (
  `ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `MOTORISTA` VARCHAR (45) NULL DEFAULT NULL,
  `FISCAL` VARCHAR (45) NULL DEFAULT NULL,
  `RELATOR` VARCHAR (45) NULL DEFAULT NULL,

```

```
`AGENTE DE LIMPEZA` VARCHAR (45) NULL DEFAULT NULL,  
PRIMARY KEY (`ID`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

Table `zer@dengue`.`local denúncia`

```
CREATE TABLE IF NOT EXISTS `zer@dengue`.`local_denúncia` (  
  `ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `ESTADO` VARCHAR (45) NOT NULL,  
  `CIDADE/MUNICIPIO` VARCHAR (45) NOT NULL,  
  `RUA` VARCHAR (45) NOT NULL,  
  `NÚMERO` INT UNSIGNED NOT NULL,  
  `CEP` INT UNSIGNED NULL DEFAULT NULL,  
  PRIMARY KEY (`ID`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

Table `zer@dengue`.`usuário`

```
CREATE TABLE IF NOT EXISTS `zer@dengue`.`usuário` (  
  `cpf` TINYINT UNSIGNED NOT NULL,  
  `nome` VARCHAR (50) NOT NULL,  
  `email` VARCHAR (50) NOT NULL,  
  `senha` VARCHAR (50) NULL DEFAULT NULL,  
  `endereço_ID` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`cpf`),  
  INDEX `fk_usuario_endereço1_idx` (`endereço_ID` ASC) VISIBLE,  
  CONSTRAINT `fk_usuario_endereço1`  
    FOREIGN KEY (`endereço_ID`)  
    REFERENCES `zer@dengue`.`endereço` (`ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## Arquitetura e Organização de Computadores

Os equipamentos que farão parte da empresa são

**Tablets** - serão utilizados pelas equipes de vigilância sanitária

preço : +- R\$ 900,00  
marca : Samsung  
modelo : smt295  
memoria int/ : 32 GB  
memoria ram : 2 GB  
processador : Quad core  
velocidade : 2 GHz  
so : Android

**Servidor** - será utilizado para controle de usuários, armazenar dados, fazer backups, dentre outras funções

preço: +- R\$ 1600,00  
Processador  
Marca: Intel  
Modelo: G4930  
Geração: Nona  
Núcleos: 2  
Threads: 2  
Clock máximo: 3.20Ghz  
Cache: 2MB

Placa de vídeo  
Marca: Intel  
Modelo: UHD 610  
Frequência: 1050mhz  
Suporte a 4K: Sim  
Resolução máxima: 4096x2304  
DirectX: 12  
OpenGL: 4.5  
Telas suportadas: 3

Placa mãe  
Chipset: Intel H Series  
Slots: 1x PCI-e x16 e 1x PCI-e x1  
USB: 2x USB 3.0 e 4x USB 2.0 (2 frontais)  
Áudio: HD de alta definição com 5.1 canais  
Áudio conexões: entrada de som, saída de som (+1 frontal), entrada de microfone (+1 frontal)  
Rede: Gigabit 10/100/1000 + Wifi de alto ganho 802.11  
Saídas de vídeo: HDMI, DVI e VGA

Memória

Capacidade: 8GB  
Geração: DDR4

Armazenamento  
Tipo: HD  
Capacidade: 3TB

Gabinete  
Tipo: Slim  
Cor: Preto  
Conexões: 2x USB e 2x áudio  
Fonte: Bivolt  
Dimensões: 35cm altura x 9,7cm largura x 28cm profundidade  
so: Windows 10

## **Monitor**

preço: +- R\$ 350,00  
Marca: LG  
Tamanho da tela (Polegadas): 19,5"  
Formato: 16:9 Widescreen  
Contraste: 5.000.000:1  
Tempo de Resposta: 5ms (GTG)  
Brilho: 200 cd/m<sup>2</sup>  
Resolução Máxima: 1366 x 768  
Pixel Pitch: 0.3177 (H) x 0.307 (V) (mm)  
Suporte de cores: 16,7 M  
Ângulo de Visão: H:90° / V:65°  
Frequencia Horizontal: 30 ~ 61 kHz  
Frequencia Vertical: 56 ~ 75 Hz  
Revestimento da Tela: Anti- Glare, Hard Coating (3H)  
Tipo: LED  
Consumo de energia  
Normal: 20W (Típico)

## **Teclado e mouse**

preço: +- R\$ 40,00  
Mouse: usb  
Resolução: 1000dpi

Teclado: usb  
10 teclas multimídia  
Padrão abnt2



## **CONCLUSÃO**

**O presente trabalho abordou questões relacionadas ao problema de saúde causado em diversos lugares pelo mosquito da dengue e como a tecnologia pode ajudar a solucioná-lo.**

**Neste trabalho, buscou-se esboçar alguns tópicos de relevante questão no procedimento de desenvolvimento de software em contraste com o problema proposto.**

**Foi feito um levantamento sobre o conceito e a evolução da tecnologia, concluindo-se que os métodos de produção de software sofreram grandes e evolutivas mudanças ao longo do tempo, possibilitando e proporcionando padrões que pudessem servir como forma de resolver a qualquer tipo de problema do cotidiano.**

**Todo o procedimento de pesquisa foi feito com base na qualidade da produção de software acima de qualquer outra questão. Porém, não deixando passar despercebido o grande aprendizado que fora absolvido em função de tais pesquisas.**

## **REFERÊNCIAS BIBLIOGRÁFICAS**

### **Lógica Computacional**

Seminário da Prática Iv: Lógica Matemática e Conjuntos: Profa. Debora Cristiane Barbosa Kirnev - 11/11/2017 - Módulo 4, 2º Semestre de 2017, Aula 1, Matutino.

### **Análise e Modelagem de Sistemas**

PRESSMAN, R.; MAXIM, B.. Engenharia de software: uma abordagem profissional. 8 ed. Porto Alegre

VALLE, R.; OLIVEIRA, S. B. (Org.). Análise e modelagem de processos de negócio: foco na notação BPMN (Business Process Modeling Not.. São Paulo: Atlas, 2013

### **Algoritmos e Programação Estruturada**

Ta4 - Algoritmos e Programação Estruturada 09/05/20

### **Modelagem Banco de Dados**

Curso completo de Bancos de Dados com MySQL

[https://www.youtube.com/playlistlist=PLucm8g\\_ezqNrWAQH2B\\_0AnrFY5dJcgOLR](https://www.youtube.com/playlistlist=PLucm8g_ezqNrWAQH2B_0AnrFY5dJcgOLR)

### **Arquitetura e Organização de Computadores**

Americanas.com

KaBuM

Mercado Livre

Pichau

Submarino