## 

## FACULDADE ANHANGUERA EDUCACIONAL

UNIDADE 1

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

## CARLOS EDUARDO SOUSA SENSSULINI – RA 2228202803

ELIANE MELLI – RA 2257521103

FABIANO DE OLIVEIRA SARDINHA – RA 2301945302

FERNANDO DE BARROS DA SILVA – RA 2204624603

LEANDRO COSTA SILVA – RA 22828189

RICHARD PAULINO – RA 2258561003

PRODUÇÃO TEXTUAL INTERDISCIPLINAR

3°SEMESTRE

Disciplinas Norteadoras: Análise Orientada a Objetos I

Banco de Dados I

Organização de Computadores

Linguagens de Programação e Estrutura de Dados

Tutor (a) EAD: JOICE SIQUEIRA LIMA

**PRODUÇÃO TEXTUAL INTERDISCIPLINAR**

Trabalho de portfólio apresentado como requisito parcial para a obtenção de pontos para a média semestral.

Orientadora: Tutora EAD Joice Lima

SUMÁRIO

**1 INTRODUÇÃO**......................................................................................................................4

**2 DESENVOLVIMENTO**........................................................................................................5

Figura 1...........................................................................................................................5

**2 .1 Tarefa 1 – Análise Orientada a Objetos I**........................................................................6

2.1.1 Diagrama de Use Cases................................................................................................6

Figura 2...........................................................................................................................6

Figura 3...........................................................................................................................8

Figura 4...........................................................................................................................9

**2 .2 Tarefa 2 – Banco de Dados I**...........................................................................................10

Figura 5.........................................................................................................................10

**2 .3 Tarefa 3 – Arquitetura e Organização de Computadores**............................................14

**2 .4 Tarefa 4 – Linguagem de Programação de Estrutura de Dados**.................................15

Figura 6.........................................................................................................................15

Figura 7.........................................................................................................................16

**3 CONCLUSÃO**......................................................................................................................22

**REFERÊNCIAS**......................................................................................................................23

# **1 INTRODUÇÃO**.

Este trabalho tem como objetivo desenvolver, a partir da Produção Textual Interdisciplinar a criação de uma aplicativo de denúncia para combate à focos de propagação do mosquito da dengue, serão utilizados conceitos estudados no 3 º Semestre do curso de Análise e Desenvolvimento de Software.

Utilizaremos os conceitos de Análise Orientada a Objetos I, Banco de Dados I, Organização de Computadores, Linguagem de Programação e Estrutura de Dados.

O nosso aplicativo desenvolvido tem o nome de Zer@Dengue e sua funcionalidade consiste em denunciar e apontar locais onde há possíveis focos de dengue. Esta denúncia será analisada e tratada por especialistas de combate à dengue.

# **2 DESENVOLVIMENTO.**

O portfólio é uma atividade, que reúne todos os conhecimentos adquiridos ao longo do semestre, e para realizar seu desenvolvimento iremos responder cada questão separada pelo sua disciplina estudada.

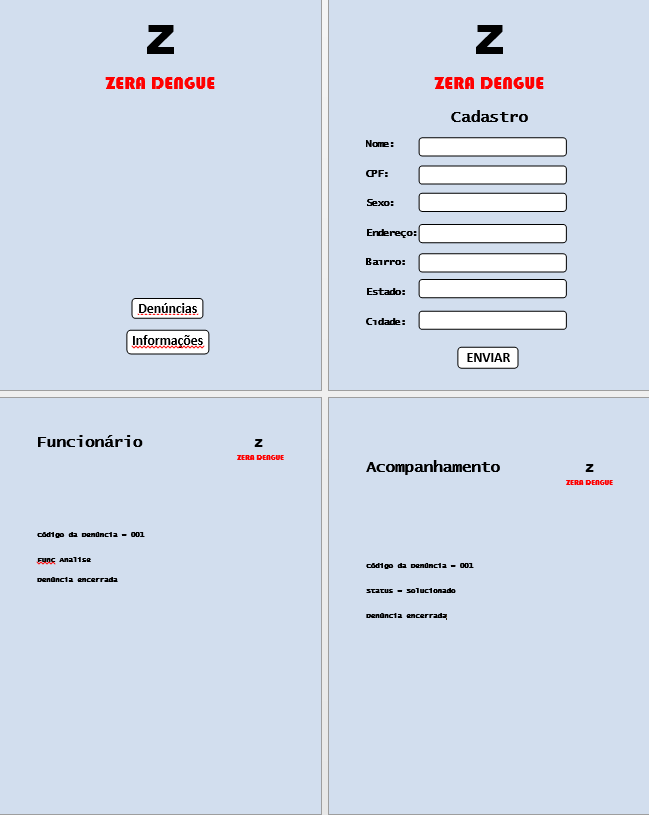


Figura 1 - Design esboço aplicativo

Questão 1.

**1) Análise Orientada a Objetos I**

Baseado no cenário proposto, elabore a modelagem da atividade de Análise de Sistemas em uma ferramenta CASE de modelagem que contemple a Unified Modeling Language (UML). Considere que a solução deve abranger as funcionalidades para realização do cadastro da pessoa física que faz a denúncia, cadastro da localização (estado, cidade e bairro), cadastro da denúncia e consulta de acompanhamento do status da denúncia, no mínimo. Seja criativo e complemente a sua solução com funcionalidades inovadoras!

A modelagem da atividade de Análise de Sistemas deve contemplar a seguinte especificação:

1. O Diagrama de Use Cases.
2. A documentação de todos os Use Cases, no formato numerado, com a descrição dos Cenários Principal e Alternativos.
3. O Diagrama de Classe.

**2.1 Tarefa 1 – Análise Orientada a Objetos I.**

2.1.1 – Diagrama de Use Cases

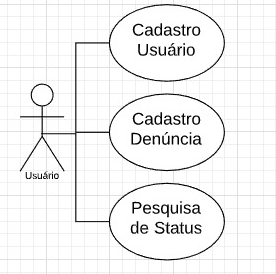


Figura-2

**Número:** 01 Use Case: Cadastro do Usuário Descrição: Candidato informa seus dados para realizar seu cadastro pessoal. Ator: Usuário

**Curso Normal:**

1. Usuário solicita seu cadastro pessoal. 2. Sistema exibe o cadastro de Usuário. 3. Usuário informa seu CPF. 4. Sistema verifica se CPF é válido. 5. Sistema verifica se não existe cadastro já associado ao CPF. 6. Candidato informa demais dados. 7. Sistema verifica se CEP é válido. 8. Sistema verifica se e-mail é válido. 9. Sistema atribui situação para “Ativo”. 10. Sistema atribui data de cadastro. 11. Usuário confirma o cadastro. 12. Sistema verifica se atributos requeridos foram preenchidos. 13. Sistema registra Candidato. 14. Sistema emite mensagem 01, informando que o usuário foi cadastrado com sucesso. 15. Sistema encerra o use case.

**Curso Alternativo 01:**

3. Sistema verifica que CPF não é válido. 3.1 Sistema emite mensagem 01, informando que o CPF não é válido. 3.2 Sistema permite informar novo CPF.

**Curso Alternativo 2:**

3. Sistema verifica que existe CPF associado ao usuário. 3.1 Sistema recupera dados do cliente associado ao CPF. 3.2 Sistema permite alteração dos dados. 3.3 Candidato informa dados (nome, endereço, bairro, cidade, estado, CEP, telefone, fax, e-mail, contato) a serem alterados. 3.4 Sistema verifica se e-mail é válido. 3.5 Candidato confirma alteração dos dados. 3.6 Sistema verifica se atributos requeridos foram preenchidos. 3.7 Sistema atualiza cadastro do cliente. 3.8 Sistema emite mensagem 01, informando que os dados do cliente foram alterados com sucesso. 3.9 Sistema encerra o use case.

**Número:** 02 Use Case: Cadastro Denúncia. Descrição: Candidato informa os dados para realizar o cadastro da denúncia. Ator: Usuário

**Curso Normal:**

1. Usuário solicita da denúncia. 2. Sistema exibe o cadastro da denúncia. 3. Usuário informa denúncia. 4. Sistema verifica se denúncia é válida. 5. Usuário informa o local da denúncia 6. Sistema ativa a localização do GPS. 7. Sistema verifica se CEP da denúncia é válido. 8. Sistema solicita imagens e libera para usuário anexar a imagem. 09. Sistema libera para descrição da denúncia. 10. Usuário confirma a denúncia. 12. Sistema verifica se atributos requeridos foram preenchidos. 13. Sistema registra Protocolo. 14. Sistema emite mensagem 01, informando o número do protocolo. 15. Sistema emite mensagem 02, informando que a denúncia foi cadastrada com sucesso. 16. Sistema encerra o use case.

**Curso Alternativo 01:**

7. Sistema verifica que CEP não é válido. 7.1 Sistema emite mensagem 01, informando que o CEP não é válido. 7.2 Sistema permite informar novo CEP.

**Curso Alternativo 2:**

7.Sistema verifica que existe uma denúncia associado ao CEP. 7.1 Sistema recupera dados da denúncia associada ao CEP. 7.2 Sistema permite alteração dos dados. 7.3 Candidato informa dados a serem alterados ou acrescentado. 7.4 Candidato confirma alteração dos dados. 7.5 Sistema verifica se atributos requeridos foram preenchidos. 7.7 Sistema atualiza cadastro

do denúncia. 7.8 Sistema emite mensagem 01, informando que os dados do denúncia foram alterados com sucesso. 7.9 Sistema encerra o use case.

**Número:** 03 Use Case: Pesquisa de Status Descrição: Candidato informa os dados para realizar a pesquisa de status da denúncia. Ator: Usuário

**Curso Normal:**

1. Sistema solicita o preenchimento do protocolo ou CEP do usuário. 2. Sistema exibe faz a consulta do status. 3. Usuário Sistema emite mensagem 01, informando que o usuário ao status da denúncia pesquisada. 4. Sistema encerra o use case.

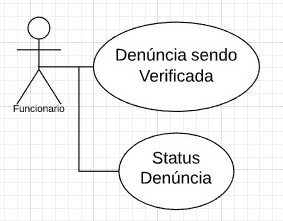


Figura-3

**Número:** 01 Use Case: Denúncia sendo Verificada Descrição: Funcionário acessa os dados para realizar a verificação. Ator: Funcionário.

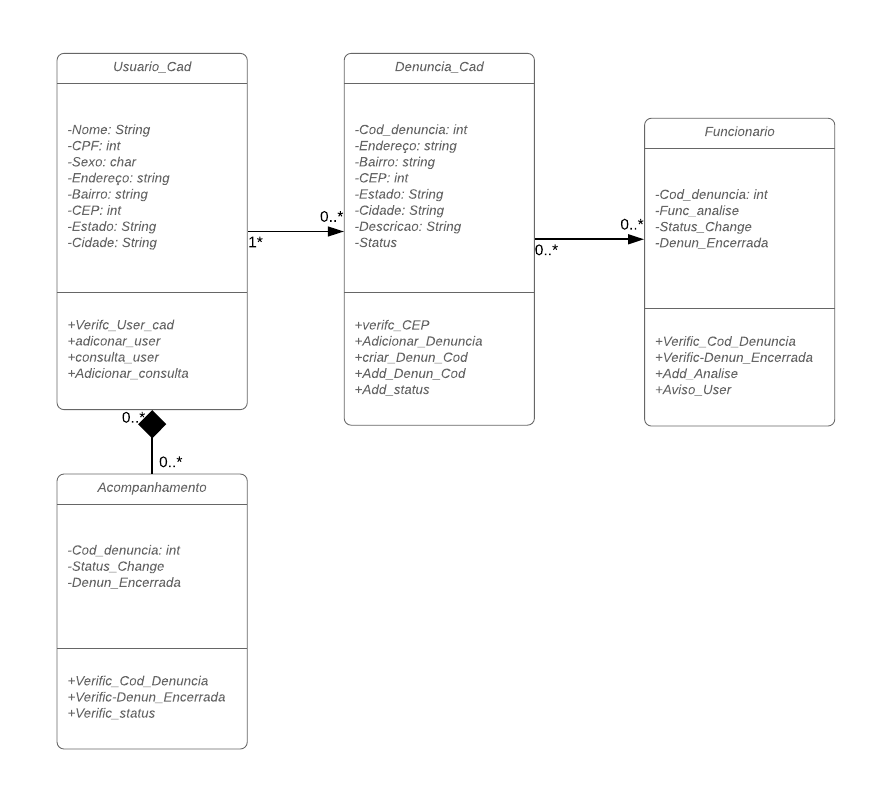
**Curso Normal:**

1.Funcionário solicita acesso a denúncia. 2. Sistema exibe a denúncia. 3. Funcionário solicitada alteração o status para em análise. 4. Sistema altera o status. 5. Sistema solicita preenchimento da análise do funcionário. 5. Sistema emite uma mensagem, informando que o status foi alterado. 6. Sistema emite uma mensagem pergunto se a denúncia será finalizada 7. Sistema encerra o use case.

Curso Alternativo:

6.Sistema verifica que status não foi finalizado pelo funcionário. 6.1 Sistema libera um novo campo para preenchimento da informações da nova visita realizada 6.2 Funcionário preenche as novas informações. 6.3 Sistema emite uma mensagem, informando que a ação foi salva com êxito. 6.4 Sistema emite uma mensagem, pergunto se a denúncia pode ser finalizada.

Diagrama de Classe:

Figura-4

Questão 2.

**2) Banco de Dados I**

O sistema Zer@Dengue precisa armazenar várias informações importantes e relevantes ao

processo de controle da dengue. Os usuários do sistema precisarão de um cadastro, sendo

necessário armazenar suas informações pessoais, bem como de login no ambiente. Além disso, como parte da principal funcionalidade do sistema, cada usuário poderá indicar locais de possíveis focos de dengue, enviando fotos, localização, etc. Sendo assim, precisamos de um banco de dados para guardar todos estes dados com segurança e eficiência. Sua tarefa então é fazer um levantamento de requisitos inicial dos tipos de informação que serão armazenadas no banco e criara partir disto um modelo entidade-relacionamento (MER) descrevendo como estas informações estarão relacionadas dentro do banco. Inclua no MER as entidades, os relacionamentos (com suas respectivas cardinalidades) e atributos.

Em um segundo momento, após a criação do MER, faça um script simples em SQL para criar todas as tabelas necessárias, em conformidade com o MER.

**2.2 Tarefa 2 – Banco de Dados I.**

O banco de dados escolhido para o complemento da geração das informações é o SQL SERVER utilizando a plataforma MICROSOFT AZURE serviço Cloud de alta disponibilidade e redundância. Abaixo todo o detalhamento de nosso diagrama da plataforma e o detalhamento do script de criação do banco de dados.

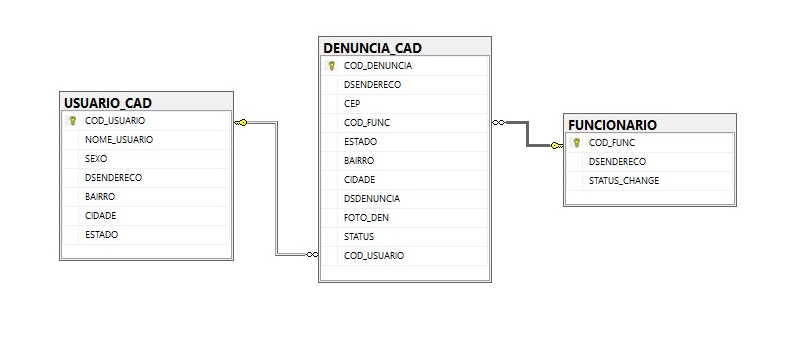


Figura 5

Na figura acima demostramos a construção da MER com suas entidade e também seus atributos, utilizamos o atributo COD\_USUARIO como chave primaria dentro da entidade USUARIO\_CAD, pois será uma forma de também identificar os usuários por seu código. Já na entidade DENUNCIA\_CAD o atributo COD\_DENUNCIA é sua chave primaria mais também COD\_USUARIO é chave estrangeira assim obedecendo a cardinalidade de 1:Infinito entre as entidade USUARIO\_CAD e DENUNCIA\_CAD, e a entidade FUNCIONARIO está com chave primário COD\_FUNC onde somente um funcionário por código pode acessar infinitas denúncias. Assim teremos que a entidade DENUNCIA\_CAD e FUNCIONARIO tem sua cardinalidade de infinito:1.

SCRIPT SQL.

USE [master]

GO

CREATE DATABASE [ZERADENGUE]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'ZERADENGUE', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\ZERADENGUE.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'ZERADENGUE\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\ZERADENGUE\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

GO

USE [ZERADENGUE]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[USUARIO\_CAD](

[COD\_USUARIO] [varchar](10) NOT NULL,

[NOME\_USUARIO] [varchar](40) NOT NULL,

[SEXO] [varchar](1) NOT NULL,

[DSENDERECO] [varchar](50) NULL,

[BAIRRO] [varchar](10) NULL,

[CIDADE] [varchar](10) NULL,

[ESTADO] [varchar](10) NULL,

CONSTRAINT [PK\_USUARIO\_CAD] PRIMARY KEY CLUSTERED

(

[COD\_USUARIO] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[FUNCIONARIO](

[COD\_FUNC] [varchar](10) NOT NULL,

[DSENDERECO] [varchar](50) NULL,

[STATUS\_CHANGE] [varchar](1) NULL,

CONSTRAINT [PK\_FUNCIONARIO] PRIMARY KEY CLUSTERED

(

[COD\_FUNC] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

USE [ZERADENGUE]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[DENUNCIA\_CAD](

[COD\_DENUNCIA] [nchar](10) NOT NULL,

[DSENDERECO] [varchar](50) NULL,

[CEP] [varchar](8) NULL,

[COD\_FUNC] [varchar](10) NULL,

[ESTADO] [varchar](10) NULL,

[BAIRRO] [varchar](10) NULL,

[CIDADE] [varchar](10) NULL,

[DSDENUNCIA] [varchar](100) NULL,

[FOTO\_DEN] [varbinary](max) NULL,

[STATUS] [char](1) NULL,

[COD\_USUARIO] [varchar](10) NULL,

CONSTRAINT [PK\_DENUNCIA\_CAD] PRIMARY KEY CLUSTERED

(

[COD\_DENUNCIA] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[DENUNCIA\_CAD] WITH CHECK ADD CONSTRAINT [FK\_DENUNCIA\_CAD\_FUNCIONARIO] FOREIGN KEY([COD\_FUNC])

REFERENCES [dbo].[FUNCIONARIO] ([COD\_FUNC])

GO

ALTER TABLE [dbo].[DENUNCIA\_CAD] CHECK CONSTRAINT [FK\_DENUNCIA\_CAD\_FUNCIONARIO]

GO

ALTER TABLE [dbo].[DENUNCIA\_CAD] WITH CHECK ADD CONSTRAINT [FK\_DENUNCIA\_CAD\_USUARIO\_CAD] FOREIGN KEY([COD\_USUARIO])

REFERENCES [dbo].[USUARIO\_CAD] ([COD\_USUARIO])

GO

ALTER TABLE [dbo].[DENUNCIA\_CAD] CHECK CONSTRAINT [FK\_DENUNCIA\_CAD\_USUARIO\_CAD]

GO

USE [ZERADENGUE]

GO

**2.3 Tarefa 3 – Arquitetura e Organização de Computadores.**

Questão 3.

**3) Arquitetura e Organização de Computadores**

Faça uma pesquisa de computadores disponíveis no mercado e indique a configuração para os possíveis equipamentos que farão parte da Empresa, levando em conta os itens citados:

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

2. Capacidade de memória RAM.

3. Tamanho em bytes do disco rígido.

4. Se acompanha teclado e mouse.

5. O tipo e o tamanho de monitor.

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

7. O preço encontrado para esse computador.

Neste item elaboramos um infraestrutura para atender a necessitado da aplicação, onde fora feito um orçamento visando o conteúdo de Arquitetura e Organização de Computadores.

1. Processador Intel® Xeon® E-2224 3.4GHz, 8M cache, 4C/4T, turbo (71W)2.
2. Memória RAM 8GB UDIMM DDR4 de 2666 MT/s.
3. Disco rígido 1TB SATA cabeado, 6 Gbps, 7200 RPM e 3,5".
4. Não acompanha teclado e mouse, mas foi adicionado na lista de compra (Teclado e Mouse Wireless Dell KM636)
5. Monitor Dell de 19.5" P2018H ,LCD com retro iluminação LED
6. Não acompanha acessório de som, jogo e placa gráfica.
7. Preço R$ 7.280,00.

**2.4 Tarefa 4 – Linguagens de Programação e Estrutura de Dados.**

**Questão 4.**

**4) Linguagens de Programação e Estrutura de Dados**

Você sabe que a estrutura de dados traz os conceitos e a compreensão de como os dados devem ser armazenados e recuperados, sendo assim deve prever uma solução para que a Equipe de Vistoria Zer@Dengue efetue suas visitas aos locais suspeitos de foco do mosquito de forma otimizada. Prevendo que necessite de uma funcionalidade que permita o gestor elaborar uma programação (roteiro) com os respectivos endereços de visitação, desenvolva:

a) Escolha uma alternativa, baseada em LISTAS, de como ficará o sequenciamento das visitas. Para facilitar o entendimento prepare um diagrama que represente a alternativa escolhida.

b) Fundamente a vantagem e a justificativa pela alternativa proposta.

Para fundamentar a escolha iremos utilizar o conceito de FILA, onde e uma estrutura de dados do tipo FIFO, que tem o conceito o primeiro elemento a ser inserido, será o primeiro elemento a sair da Fila como exemplo uma fila bancaria de atendimento.

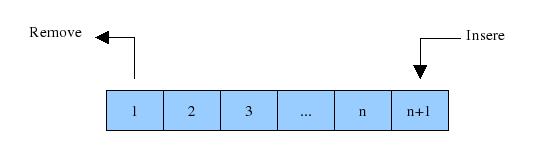


Figura -6

Por se tratar de denúncias nosso aplicativo registra cada denuncia, por um código numérico como um protocolo e este entraria em uma fila sequencial, que o analista responsável iria acessar o conteúdo da fila e atenderia o primeiro. Após a conclusão do chamado o primeiro item da Fila seria excluído e sequencialmente o segundo termo do vetor assumiria seu lugar.

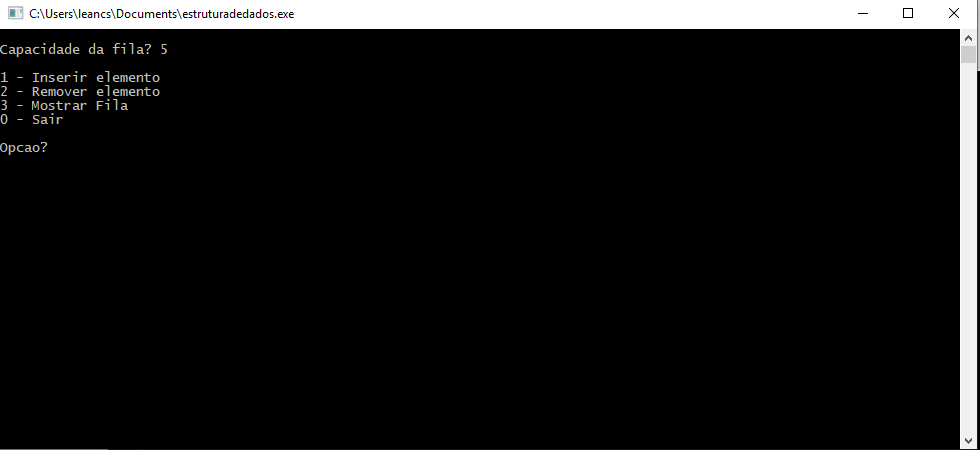


Figura 7 – esboço sistema de fila.

Algoritmo:

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

struct Fila {

int capacidade;

float \*dados;

int primeiro;

int ultimo;

int nItens;

};

void criarFila( struct Fila \*f, int c ) {

f->capacidade = c;

f->dados = (float\*) malloc (f->capacidade \* sizeof(float));

f->primeiro = 0;

f->ultimo = -1;

f->nItens = 0;

}

void inserir(struct Fila \*f, int v) {

if(f->ultimo == f->capacidade-1)

f->ultimo = -1;

f->ultimo++;

f->dados[f->ultimo] = v; // incrementa ultimo e insere

f->nItens++; // mais um item inserido

}

int remover( struct Fila \*f ) { // pega o item do comeÃ§o da fila

int temp = f->dados[f->primeiro++]; // pega o valor e incrementa o primeiro

if(f->primeiro == f->capacidade)

f->primeiro = 0;

f->nItens--; // um item retirado

return temp;

}

int estaVazia( struct Fila \*f ) { // retorna verdadeiro se a fila estÃ¡ vazia

return (f->nItens==0);

}

int estaCheia( struct Fila \*f ) { // retorna verdadeiro se a fila estÃ¡ cheia

return (f->nItens == f->capacidade);

}

void mostrarFila(struct Fila \*f){

int cont, i;

for ( cont=0, i= f->primeiro; cont < f->nItens; cont++){

printf("%.2f\t",f->dados[i++]);

if (i == f->capacidade)

i=0;

}

printf("\n\n");

}

int main () {

int opcao, capa;

float valor;

struct Fila umaFila;

// cria a fila

printf("\nCapacidade da fila? ");

scanf("%d",&capa);

criarFila(&umaFila, capa);

// apresenta menu

while( 1 ){

printf("\n1 - Inserir elemento\n2 - Remover elemento\n3 - Mostrar Fila\n0 - Sair\n\nOpcao? ");

scanf("%d", &opcao);

switch(opcao){

case 0: exit(0);

case 1: // insere elemento

if (estaCheia(&umaFila)){

printf("\nFila Cheia!!!\n\n");

}

else {

printf("\nValor do elemento a ser inserido? ");

scanf("%f", &valor);

inserir(&umaFila,valor);

}

break;

case 2: // remove elemento

if (estaVazia(&umaFila)){

printf("\nFila vazia!!!\n\n");

}

else {

valor = remover(&umaFila);

printf("\n%1f removido com sucesso\n\n", valor) ;

}

break;

case 3: // mostrar fila

if (estaVazia(&umaFila)){

printf("\nFila vazia!!!\n\n");

}

else {

printf("\nConteudo da fila => ");

mostrarFila(&umaFila);

}

break;

default:

printf("\nOpcao Invalida\n\n");

}

}

}

**3 CONCLUSÃO.**

Diante do que foi proposto no trabalho de Produção Textual Interdisciplinar, visamos atender em sua completitude as etapas propostas, nas quais desenvolvemos as competências e a compreensão embasadas em análise de desenvolvimento de sistemas.

Tal estudo fora realizado tendo por base pesquisas nas seguintes matérias: Análise Orientada a Objetos, Banco de Dados, Arquitetura e Organização de Computadores e Linguagem de Programação e Estrutura de Dados, chegando-se ao desenvolvimento do aplicativo Zer@dengue.

Dessa forma, com o trabalho ora finalizado, pudemos perceber a importância da matéria estudada, a qual impacta diretamente no dia a dia da população, inclusive, como no caso em questão, podendo ajudar no enfrentamento de problemas de saúde pública.

**REFERÊNCIAS**

VENTURA, Plínio. Entendendo o Diagrama de Classes da UML. **ateomomento**, 2020. Disponível em: < https://www.ateomomento.com.br/uml-diagrama-de-classes/ >. Acesso em: 24, Abril de 2020.

ORIENTAÇÃO a Objetos com Java – Professor Titular Clovis Fernandes – Professor Colaborador Eduardo Guerra – Local: Instituto Tecnológico da Aeronáutica, 2020. 21:39 – Disponível em:<https://pt.coursera.org/lecture/orientacao-a-objetos-com-java/diagrama-de-classes-uml-classe-associacao-e-multiplicidade-7adrt>. Acesso em: Abril de 2020.

SpaceProgrammer. Compreendendo multiplicidade e os tipos de Associação **spaceprogrammer**, 2020. Disponível em: < http://spaceprogrammer.com/uml/compreendendo-multiplicidade-e-os-tipos-de-associacao/>. Acesso em: 24, Abril e 2020.

GUEDES, Gilleanes T. A. UML: Uma abordagem prática. São Paulo: Novatec, 2006.

Hallan. Documentação de Casos de Uso. **hallanmedeiros**, 2010. Disponível em: < https://blog.hallanmedeiros.com/2010/03/29/documentacao-de-casos-de-uso/>. Acesso em: 28, Abril e 2020.

DEVMEDIA. Guia Completo de SQL. **devmedia**, 2020. Disponível em: < https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>. Acesso em: 27, Abril e 2020.

**PHPMYADMIN**. PhpMyadmin. **db4free**, 2020. Disponível em: <https://www.db4free.net/phpMyAdmin/db\_structure.php?server=1&db=bancofac>. Acesso em: 27, Abril e 2020.

**DELL: dell**, 2020. Disponível em: < https://www.dell.com/pt-br/work/shop/serviceselection/pe\_t140\_13161?selectionState=eyJPQyI6InBlX3QxNDBfMTMxNjEiLCJNb2RzIjpbeyJJZCI6MTUyMCwiT3B0cyI6W3siSWQiOiJHQ1ExVlBBIn0seyJJZCI6IkcxVEFQV0oifV19LHsiSWQiOjE1NTAsIk9wdHMiOlt7IklkIjoiRzFaRVlEUiJ9XX0seyJJZCI6MTYwMCwiT3B0cyI6W3siSWQiOiJHN0lIVzZFIn1dfSx7IklkIjo5MDAzLCJPcHRzIjpbeyJJZCI6IjU4MC1BRElPIn0seyJJZCI6Ik1fUDIwMThIQiIsIkNvTW9kcyI6W3siSWQiOjEsIk9wdHMiOlt7IklkIjoiR0s3MlVSNSJ9XX1dfV19XSwiVGkiOiIiLCJEaSI6IiJ9&cartItemId=>. Acesso em: 02, Maio e 2020.

**DELL:** **dell**, 2020. Disponível em: < https://www.dell.com/pt-br/work/shop/cty/pdp/spd/poweredge-t140/pe\_t140\_13161?configurationid=2d1f85f8-cf08-4d5f-8b93-f957bf379e54>. Acesso em: 02, Maio e 2020.

ESTRUTURA de Dados e Algoritmos – Professor Ricardo Farias: UFRJ, 05/2020. – Disponível em:< https://www.cos.ufrj.br/~rfarias/cos121/filas.html>. Acesso em: Maio de 2020.