

**PROJETO INTERDISCIPLINAR**

**Controle de Vacinas**

ControleVacina

**Alunos:**

|  |  |
| --- | --- |
| **RGM** | **Nome** |
| 31465196 | Gabriel de Jesus Melo |
| 31088341 | Pedro Arthur Alves da Costa |
|  | Jeferson Castilho |

São Paulo

2022

**UNIVERSIDADE CRUZEIRO DO SUL**

**PROJETO INTERDISCIPLINAR**

**Controle de Vacinas**

ControleVacina

## Trabalho apresentado como parte do requisito para aprovação na Disciplina de Projeto Interdisciplinar do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul.

**Orientadores:** Prof. Cristiane e Prof. Jessica

São Paulo

2022**Sumário**

[1. Apresentação: 3](#_Toc48489109)

[1.1 Justificativa e Motivação 3](#_Toc48489110)

[1.2 Dados do Sistema. 3](#_Toc48489111)

[2 Requisitos de Técnica de Desenvolvimento de Algoritmos 3](#_Toc48489112)

[3 Requisitos de Programação Orientada a Objetos 3](#_Toc48489113)

[4 Consideração finais 3](#_Toc48489114)

[5 Referencias 3](#_Toc48489115)

[APENSO 1 – Cronograma de entrega de atividades. 4](#_Toc48489116)

# 1. Apresentação:

## 1.1 Justificativa e Motivação

O tema Controle de Vacinas foi escolhido com base na situação atual do país e mundo, onde vivemos uma pandemia e a produção e controle de vacinas foi de suma importância para diminuir e controlar a situação.

Trabalhar nesse projeto nos proporcionou desafios e descobertas através da pesquisa

## 1.2 Dados do Sistema.

O Sistema ControleVacina trabalha com 3 tipos de procedimentos, produção de vacinas, aplicação de vacinas e controle de dados de pacientes vacinados e também controle de eficácia e aplicação das vacinas produzidas. Com o programa podemos ter um controle de estoque onde são produzidas quantidades de vacina e também retiradas para distribuição.

# 2 Requisitos de Técnica de DESENVOLVIMENTO DE ALGORITMOS

No desenvolvimento do algoritmo de ControleVacina foram utilizados os conceitos de variáveis, estruturas de repetição e condição, funções e métodos e também vetores.

Foi utilizado variáveis de tipos primitivos tanto globais, quanto locais conforme print abaixo para armazenar valores, assim como vetores para armazenar informações de estoque e também de pacientes.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Foram criados menus para interação conforme print utilizando estrutura de repetição faça, enquanto.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

As opções descritas no menu encaminham o agente que irá controlar o programa para a área solicitada, que será executada através de métodos.

Um dos métodos utilizados para produção e adição de quantidade de vacinas no estoque foi o **producaovacina**(), conforme print abaixo foi utilizado estrutura de decisão para armazenar no vetor as informações de acordo com a opção selecionada no menu

Texto

Descrição gerada automaticamente

Os métodos abaixo foram utilizados para imprimir informações salvas no programa em Literal, conforme print:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

E por fim o método utilizado para aplicação e cadastro de pacientes, conforme informações digitadas pelo agente:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

**Abaixo descrito o código em alg:**

Algoritmo "ControledeVacinas"

//variáveis Globais

Literal agente,vacina,nome,dose,cpf,fabricado,fab[5],vac[5],paciente[200],cpfPaciente[200];

inteiro re,idade,entrada,saida,qtvac[5],idadePaciente[200];

real eficacia;

inicio

//Variaveis de controle

Literal resp = "";

inteiro op = 0,

// login

login();

//Menu Principal

faca {

escreva("Sistema de controle de Vacinas");

escreva("== Menu principal ==");

escreva("[1] - Aplicação de vacina");

escreva("[2] - Verificar Registro de Vacinados");

escreva("[3] - Cadastrar entradas Vacinas");

escreva("[4] - Cadastrar Saída de Vacinas");

escreva("[5] - Sair");

escolha(op){

caso 1:

aplicaVacina();

imprimePaciente();

pare;

caso 2:

imprimePaciente();

pare;

caso 3:

addEstoque();

imprimirE();

pare;

caso 4:

retEstoque();

imprimirE();

caso 5:

escreva("Finalizando programa...!");

pare

padrao:

escreva("Opção invalida");

}

} enquanto (op != 5)

}

escreva("Programa encerrado! até logo!")

fim

//Metodo login e senha (RE).

void login(literal agente, inteiro re);

inicio

escreva("Bem Vindo ao Controle De Vacinas");

escreva(" == lOGIN == ");

escreva("Digite o Nome do agente")

leia(agente);

escreva("Digite o RE do agente");

leia(re);

fim

void aplicaVacina()

Literal resp, inteiro op

inicio

enquanto(resp == 'n' || resp == 'N'){

escreva("Qual vacina será aplicada?");

escreva("[1] - Coronavac");

escreva("[2] - Astrazenica ");

escreva("[3] - Pfizer");

escreva("[4] - Gripe");

escreva("[5] - Poliomielite");

escreva("[6] - Sair");

escolha(op){

caso 1 :

escreva("Digite o nome do paciente");

leia(nome);

escreva("Digite a idade");

leia(idade);

escreva("Digite o CPF do Paciente")

leia(cpf)

vacina = "Aplicado : Coronavac"

se(op <10){

paciente[op] = nome;

idadePaciente[op] = idade;

cpfPaciente[op] = cpf;

} else{

escreva("Não é possivel realizar o cadastro falta espaço em disco ");

}

pare

caso 2 :

escreva("Digite o nome do paciente");

leia(nome);

escreva("Digite a idade");

leia(idade);

escreva("Digite o CPF do Paciente")

leia(cpf)

vacina = "Aplicado : Astrazenica"

se(op <10){

paciente[op] = nome;

idadePaciente[op] = idade;

cpfPaciente[op] = cpf;

} else{

escreva("Não é possivel realizar o cadastro falta espaço em disco ");

}

pare

caso 3 :

escreva("Digite o nome do paciente");

leia(nome);

escreva("Digite a idade");

leia(idade);

escreva("Digite o CPF do Paciente")

leia(cpf)

vacina = "Aplicado : Pfizer"

se(op <10){

paciente[op] = nome;

idadePaciente[op] = idade;

cpfPaciente[op] = cpf;

} else{

escreva("Não é possivel realizar o cadastro falta espaço em disco ");

}

pare

caso 4 :

escreva("Digite o nome do paciente");

leia(nome);

escreva("Digite a idade");

leia(idade);

escreva("Digite o CPF do Paciente")

leia(cpf)

vacina = "Aplicado : Gripe"

se(op <10){

paciente[op] = nome;

idadePaciente[op] = idade;

cpfPaciente[op] = cpf;

} else{

escreva("Não é possivel realizar o cadastro falta espaço em disco ");

}

pare

caso 5 :

escreva("Digite o nome do paciente");

leia(nome);

escreva("Digite a idade");

leia(idade);

escreva("Digite o CPF do Paciente")

leia(cpf)

vacina = "Aplicado : Poliomielite"

se(op <10){

paciente[op] = nome;

idadePaciente[op] = idade;

cpfPaciente[op] = cpf;

} else{

escreva("Não é possivel realizar o cadastro falta espaço em disco ");

}

pare

caso 6:

escreva("Até logo");

}

escreva("Deseja sair do programa (s/n): ")

leia(resp);

enquanto( resp != 'S' e resp != 's'e resp != 'N' e resp != 'n'){

escreva("Deseja sair do programa (s/n): ")

leia(resp);

}

fim

Literal toStringAplicaVacina(Literal);

inicio

retorne "Paciente : " + nome + "\nIdade" + idade + "\n" + vacina + "\n" + dose + "\n-------------------");

fim

void imprimePaciente()

inicio

para ( int i = 0 ; i<aplica.lenght ; i++) {

escreva("Aplicação de Vacina" + aplica[i].toStringAplicaVacina());

fim

Literal toStringEstoque(Literal)

inicio

retorne "EstoqueVacina qtvac=" + qtvac[] + ", vac=" + vac[] + ", fab="

+ fab[] + "]";

fim

void imprimeEstoque()

inicio

escreva("Estoque de Vacinas");

para ( int i = 0; i <vac.lenght; i++){

escreva("Vacina : " + vac[i].toString() + "\n" + fab[i].toString() + "\nQuantidade : " + qtvac[i] + "\n---------------");

}

fim

void ProducaoVacina()

inicio

para ( int i = 0 ; i < vac.lenght ; i++){

int resp = 0;

escreva ("Setor de Controle e Produção de vacinas"+"\n"+

"Qual Vacina deseja Depositar?" + "\n" +

"[1] - Coronavac" "\n" +

"[2] - Astrazeneca" +"\n" +

"[3] - Pfizer " + "\n" +

"[4] - Gripe " +"\n" +

"[5] - Poliomielite " + "\n" +

"[6] - Sair"));

se (resp ==1) {

vac[i] = "Coronavac";

fab[i] = "Fabricado por Instituto Butantan";

escreva("Digite a quantidade");

leia(entrada);

qtvac[i] = entrada;

} senao {

se ( resp == 2 ) {

vac[i] = "Astrazeneca";

fab[i] = "Fabricado por AstraZeneca-SKBio";

escreva("Digite a quantidade");

leia(entrada);

qtvac[i] = entrada;

} senao {

se ( resp == 3 ) {

vac[i] = "Pfizer";

fab[i] = "Fabricado por BioNTech ";

escreva("Digite a quantidade");

leia(entrada);

qtvac[i] = entrada;

} senao {

se ( resp == 4 ){

vac[i] = "Gripe";

fab[i] = "Fabricado por Instituto Butantan ";

escreva("Digite a quantidade");

leia(entrada);

qtvac[i] = entrada;

} senao {

vac[i] = "Poliomielite";

fab[i] = "Fabricado por Bio-Manguinhos ";

escreva("Digite a quantidade");

leia(entrada);

qtvac[i] = entrada;

}

fim

void DistribuicaoVacina()

inicio

inteiro resp = 0;

para ( int i = 0; i < vac.lenght ; i++) {

escreva("Setor de Controle e Produção de vacinas"+"\n" +

"Qual Vacina deseja retirar?"+ "\n" +

"[1] - Coronavac "+"\n" +

"[2] - Astrazeneca" "\n" +

"[3] - Pfizer " +"\n" +

"[4] - Gripe " + "\n" +

"[5] - Poliomielite" + "\n" +

"[6] - Sair ");

se (resp == 1) {

vac[i] = "Coronavac";

fab[i] = "Fabricado por Instituto Butantan";

escreva("Digite a quantidade");

leia(saida)

qtvac[i] = qtvac[i] - saida;

} senao {

se (resp == 2) {

vac[i] = "Astrazeneca";

fab[i] = "Fabricado por AstraZeneca-SKBio";

escreva("Digite a quantidade");

leia(saida)

qtvac[i] = qtvac[i] - saida;

} senao {

se (resp == 3) {

vac[i] = "Pfizer";

fab[i] = "Fabricado por BioNTech ";

escreva("Digite a quantidade");

leia(saida)

qtvac[i] = qtvac[i] - saida;

} senao {

se (resp == 4) {

vac[i] = "Gripe";

fab[i] = "Fabricado por Instituto Butantan ";

escreva("Digite a quantidade");

leia(saida)

qtvac[i] = qtvac[i] - saida;

} senao {

se (resp == 5) {

vac[i] = "Poliomielite";

fab[i] = "Fabricado por Bio-Manguinhos ";

escreva("Digite a quantidade");

leia(saida)

qtvac[i] = qtvac[i] - saida;

} senao {

System.exit(0);

}

}

}

}

}

}

}

# 3 Requisitos de Programação Orientada a Objetos

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

O programa “ControleVacina” compõe de 3 classes, uma super classe abstrata chamada “*ControleVacina”* , duas subclasses que herdam de ControleVacina chamadas de AplicacaoVacina e EstoqueVacina e também a classe UsaControleVacina com método main para execução.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Os métodos get e set foram adicionados conforme a necessidade das classes, e todas possuem o método toString para auxiliar o método abstrato “Imprimir” conforme exemplo abaixo :

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Foram criados objetos polimorfos conforme a necessidade e a interação do programa conforme print abaixo :

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Cada objeto executa uma ação programada nos métodos e imprime de acordo com sua classe com o método abstrato preenchido “imprimir”.

Para armazenamento de dados foi utilizado vetores e Arraylist, assim como estrutura de repetição para preencher o Arraylist e estrutura de decisão para preencher os vetores

Criação dos vetores, foram determinados 5 indices devido a quantidade de vacinas que foram pré estabelecidas.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Foi utilizado Arraylist para armazenar as informações sobre a aplicação e dados do paciente.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Informações salvas tanto nos vetores, quanto no Arraylist serão imprimidas no método imprimir sempre que forem feitas modificações.

O Menu principal foi criado utilizado switch case para encaminhar para a opção desejada pelo ” Agente “. Ao selecionar a opção o programa irá encaminhar e chamar o metodo conforme a opção desejada, foi adicionado o cast para que o objeto chame o metodo da classe específica e abaixo a impressão dos dados.



# 

# 4 Consideração finais

Destacar as facilidades e dificuldades durante a elaboração do projeto e outros comentários julgados pertinentes.

Desenvolver um sistema de Controle de Vacinas foi um desafio, onde tivemos facilidade em buscar informações e também desenvolver parte do código, vetores e arraylist foram um desafio que depois de muita pesquisa e exemplos nas aulas conseguimos utilizar.

Após o desenvolvimento desse projeto, temos mais pratica em como funcionam os conceitos das matérias utilizadas.

# 5 BIBLIOGRAFIA

Curso programação orientada a objetos Profa Cristiane.

Curso Técnica de desenvolvimento de algoritmos Profa Jessica.

# Apêndice 1 – Cronograma de entrega de atividades.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Descrição | Data | | Prazo do cronograma em semanas | | | | | | | | | | | | | | | | | |
| Início | Término |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 1 | Definição do tema e planejamento inicial |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Entrega do projeto final e apresentação |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**OBS:**

1. **P =** previsto; **R =** realizado