

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

**PROJETO INTEGRADO**

Desenvolvimento de Sistema Gerenciador de Restaurante Vegano

SÃO JOÃO DA BOA VISTA, SP

ABRIL, 2023

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS

**PROJETO INTEGRADO**

Desenvolvimento de Sistema Gerenciador de Restaurante Vegano

SÃO JOÃO DA BOA VISTA, SP

ABRIL, 2023

MÓDULO: DESENVOLVIMENTO DESKTOP

Professor: Sidney Gitcoff Teles

ESTUDANTES:

Gabriel Mandes de Almeida - 1012023100249

Karla K G. G. Pinho - 1012022200042

Lucas Balbino Paulino- 1012023100315

Marcelo Augusto Camargo Souza Dias - 1012023100332

Neemias Francisco de Souza – 1012023100325

Thais de Moraes Vieira – 1012022100664

SÃO JOÃO DA BOA VISTA, SP

ABRIL, 2023

## Sumário

1. Introdução.....	7
2. Objetivo .....	9
3. Desenvolvimento.....	10
4. Banco de Dados .....	11
4.1 Modelo de Entidade Relacionamento .....	11
4.2 Modelo Lógico.....	13
4.3 Modelo físico .....	14
5. Linguagens para desenvolvimento .....	15
6. Arquitetura do Projeto .....	17
6.1 Model .....	17
6.2 View .....	19
6.2.1 Tela de login.....	20
6.2.2 Tela de Menu.....	21
6.2.3 Tela de cadastro de Cliente .....	22
6.2.4 – Tela de Cadastro de Produto .....	23
6.2.5 Tela de Consulta de Cliente .....	24
6.2.6 Tela de consulta de produto .....	25
6.3 Controller .....	26
7. Conexão com o banco.....	28
8. CONCLUSÃO.....	30

9 Referências .....31

## Lista de Figuras

<b>Figura 1 Exemplo de Diagrama de Entidade de Relacionamento .....</b>	<b>12</b>
<b>Figura 2 Modelo Lógico .....</b>	<b>13</b>
<b>Figura 3 Modelo físico.....</b>	<b>14</b>
<b>Figura 4 Apache NetBeans .....</b>	<b>16</b>
<b>Figura 5 Organização do Pacote Model .....</b>	<b>17</b>
<b>Figura 6 Exemplo de Get e Set da Class Cliente .....</b>	<b>18</b>
<b>Figura 7 Organização do pacote Views .....</b>	<b>19</b>
<b>Figura 8 Tela de Login.....</b>	<b>20</b>
<b>Figura 9 Tela de Menu.....</b>	<b>21</b>
<b>Figura 10 Tela de Cadastro de Cliente.....</b>	<b>22</b>
<b>Figura 11 Tela de Cadastro de Produto .....</b>	<b>23</b>
<b>Figura 12 Tela de Consulta de Cliente .....</b>	<b>24</b>
<b>Figura 13 Tela de Consulta de Produto.....</b>	<b>25</b>
<b>Figura 14 Fluxo do MVC .....</b>	<b>26</b>
<b>Figura 15 Exemplo de Controller .....</b>	<b>27</b>
<b>Figura 16 Organização do Pacote DataBase .....</b>	<b>28</b>
<b>Figura 17 Trecho do Script de Conexão com o Banco de Dados .....</b>	<b>28</b>
<b>Figura 18 Organização do Pacote Libraries .....</b>	<b>29</b>

# 1. Introdução

O Projeto Integrado do primeiro trimestre de 2023, conforme esclarecido em seu Plano Pedagógico, tem como objetivo desenvolver um projeto exploratório que pretende trazer inicialmente o conceito programação orientada a objeto e banco de dados. Estabelecido em nosso projeto para elucidar forma de pensamento e de solução de problema, aplicando etapas e processos, transformando informação bruta em hipótese para um intercepto e sintaxe de solução.

Nesse sentido, o referido plano pedagógico contextualiza o trabalho da seguinte forma:

Desenvolver um software para desktop, utilizando programação orientada a objeto e banco de dados que facilitasse processos dentro de uma empresa, passando assim de ser feito no papel para um sistema dedicado a isso.

Diante deste desafio, foram iniciadas as interações entre os membros do Grupo 06, optando por fazer um sistema que facilitaria o dia a dia de um dos membros do grupo cuja tem um empresa que tem como nicho a venda de produtos veganos.

O método de pesquisa utilizado foi coletas de requisitos e dados para compreendermos o problema em si, bem como a discussão de sugestões de solução.

O estudo foi desenvolvido a partir de:

1. Pesquisas bibliográficas analisando os conceitos de Programação orientada a objeto e Banco de Dados, juntamente com os materiais oferecidos pelo professor nas aulas ministradas.
2. Pesquisa de campo: Reuniões realizadas juntamente com o cliente, com questões exploratórias para obtenção de requisitos que melhores resolveriam o problema em questão.

Utilizamos a plataforma BRModelo para a confecção do modelo Relacional do projeto. A plataforma utilizada para o desenvolvimento do projeto foi o NetBeans, utilizando a linguagem de programação Java na versão 20 se já para o gerenciamento do bando de dados SGBD (Sistema Gerenciador de Banco de Dados) MySQL, JDBC (Java Database Conector) e MYSQL-Connector-J, maiores detalhes destas ferramentas serão descritos nos próximos capítulos.

## 2. Objetivo

O objetivo deste trabalho foi explorar e apresentar dados dos clientes e produtos da empresa de forma mais rápida e visual. Optamos por fazer um sistema que gerenciasse essas informações.

Este sistema teria por finalidade possibilitar que a cliente pudesse cadastrar novos clientes e novos produtos, podendo também consultar as informações dos mesmos para obtenção de alguma informação, assim como fazer a atualização das informações caso ocorra alguma mudança.

Pensamos em fazer um interface que fosse simples e intuitiva assim facilitando todo o processo e não tomando tanto tempo para a inserção dos dados.

Assim, considerou-se para esse sistema os fatores mais importantes para que constassem no banco de dados e a melhor forma de exibição de dados nas interfaces. Nas reuniões com dados e análises foram decididas quais informações dos clientes e dos produtos seriam pertinentes armazenar.

### 3. Desenvolvimento

Como todo o processo de cadastro de cliente e produto é feito à mão, o torna trabalhoso e depende da disponibilidade da cliente para ser realizado.

Ao longo da idealização do projeto, e análise do problema a ser resolvido, chegamos à conclusão de quais informações seriam importantes conter no banco de dados, sendo elas:

Nos clientes as informações que deveriam ser armazenadas são: Nome, CPF, Telefone, Bairro, Logradouro e Complemento. Já nos produtos seriam armazenados o Nome, Descrição e Valor.

Partindo deste ponto iniciamos a criação do nosso banco de dados.

## 4. Banco de Dados

O Banco de Dados Armazena um conjunto de dados que se relacionam, os quais podem conter várias informações como nome, telefone, endereço, entre outros.

A construção do Banco de dados desse projeto foi feita em 3 etapas a partir da análise de requisitos, a saber, Modelo Relacional, Lógico e Físico.

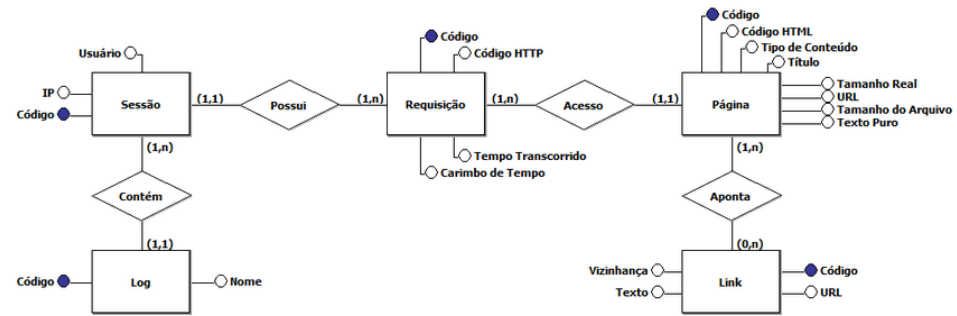
Descreveremos com mais detalhes essas etapas nos próximos tópicos

### 4.1 Modelo de Entidade Relacionamento

O Modelo Entidade-Relacionamento, também conhecido pela sigla **MER**, é um modelo conceitual que descreve os objetos envolvidos no domínio de um sistema de bancos de dados a ser construído, é uma maneira mais sistemática de descrever o processo lógico, simplificando o uso de tabelas e relacionamento.

A partir disso é gerado um dia diagrama com todas as entidades, os atributos e os relacionamentos do banco de dados .

*Figura 1 Exemplo de Diagrama de Entidade de Relacionamento*



Fonte: [Modelo de entidade relacionamento - Bing images](#)

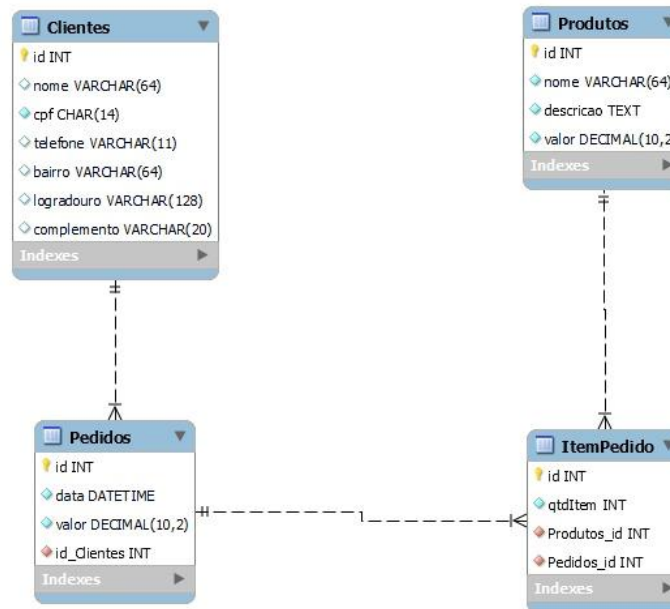
## 4.2 Modelo Lógico

No Modelo lógico colocamos como os dados serão armazenados no banco, assim como seus relacionamento, assim como padronizamos os dados, criamos as chaves primarias e estrangeiras e atributos únicos

Nesta fase trazemos o nosso banco de dados mais próxima de um banco de dados real, trazendo os relacionamentos entre as tabelas

A seguir temos uma foto do nosso modelo lógico:

**Figura 2 Modelo Lógico**



**Fonte: Autor**

### 4.3 Modelo físico

Nesta etapa fizemos a modelagem física do modelo de banco, trazendo o modelo lógico para um modelo real de banco de dados, nesta etapa realizamos os testes de execução para garantir que todos os relacionamentos estavam corretos.

A seguir temos uma foto do modelo físico do banco:

*Figura 3 Modelo físico*

```
Projeto Integrado - Restaurante - createDatabase.sql
1 DROP DATABASE IF EXISTS projetointegrador_restaurante;
2 CREATE DATABASE IF NOT EXISTS projetointegrador_restaurante;
3 USE projetointegrador_restaurante;
4
5 DROP TABLE IF EXISTS clientes;
6 CREATE TABLE IF NOT EXISTS clientes(
7 id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
8 nome VARCHAR(64) NOT NULL,
9 CPF CHAR(14) UNIQUE NOT NULL,
10 telefone CHAR(11) NOT NULL,
11 bairro VARCHAR(64) NOT NULL,
12 logradouro VARCHAR(128) NOT NULL,
13 complemento VARCHAR(20) NULL DEFAULT NULL
14 );
15
16 DROP TABLE IF EXISTS pedidos;
17 CREATE TABLE IF NOT EXISTS pedidos(
18 id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
19 dataPedido DATETIME,
20 valor DECIMAL(10,2),
21 statusPedido TINYINT NOT NULL,
22 idCliente INT NOT NULL,
23 FOREIGN KEY (idCliente) REFERENCES clientes(id)
24 );
25
26 DROP TABLE IF EXISTS produtos;
27 CREATE TABLE IF NOT EXISTS produtos(
28 id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
29 nome VARCHAR(64) NOT NULL,
30 descricao TEXT NOT NULL,
31 valor DECIMAL(10,2) NOT NULL
32 );
33
34 DROP TABLE IF EXISTS itemPedido;
35 CREATE TABLE IF NOT EXISTS itemPedido(
36 id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
37 qtddItem TINYINT NOT NULL DEFAULT 1,
38 idProduto INT NOT NULL,
39 idPedido INT NOT NULL,
40 FOREIGN KEY (idProduto) REFERENCES produtos(id),
41 FOREIGN KEY (idPedido) REFERENCES pedidos(id)
42 );
43
44
```

**Fonte: Autor**

## 5. Linguagens para desenvolvimento

Linguagem tem por finalidade executar instruções que um determinado sistema irá fornecer, quando falamos de programação orientada a objeto temos algumas opções como C++, Java C# entre outras.

Linguagens com esse paradigma se baseia principalmente em dois conceitos:

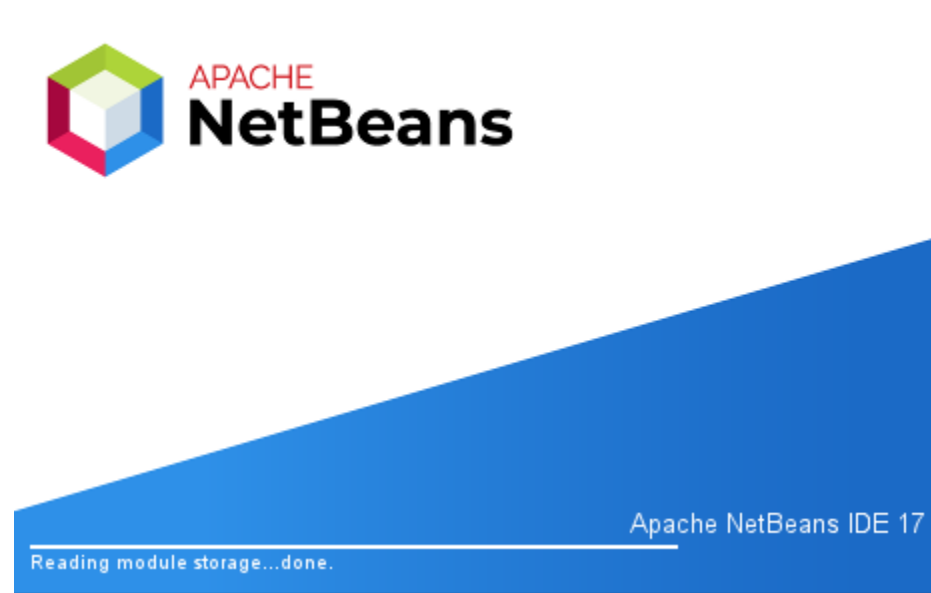
- Classes
- Objetos

Como visto ao decorrer das unidades a POO tem 3 principais características, sendo elas:

- Encapsulamento: Protege os dados para que o usuário não tenha acesso direto ao mesmo, permite que o programa possua menos erros.
- Herança: Torna possível a reutilização de classes, ou seja, criação de novas classes que podem reutilizar atributos e métodos da classe de origem.
- Polimorfismo: Quando usado qualquer classe que foi herdada da mesma classe de origem pode utilizar métodos que tem a mesma identificação, fazendo que os esses métodos possam ter comportamentos diferentes em cada classe.

Para o desenvolvimento deste projeto foi escolhida a linguagem Java, pois ela permite que possamos explorar os conhecimentos adquiridos de encapsulamento, herança e polimorfismo, vistos nesse trimestre, e foi usada juntamente com o IDE(Ambiente de desenvolvimento Integrado) NetBeans.

*Figura 4 Apache NetBeans*



**Fonte: Autor**

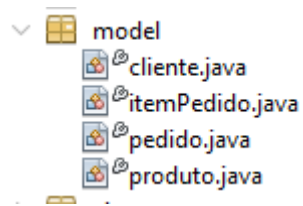
## 6. Arquitetura do Projeto

Já dentro do NetBeans, começa o desenvolvimento do sistema, seguimos o padrão de arquitetura MVC ( Model, View, Controller).

### 6.1 Model

Este pacote é responsável pela manipulação de dados, nele é feito a leitura e escrita de dados, validações, e é onde fica todos os Get e Set das classes utilizadas no projeto, como mostrado a seguir:

*Figura 5 Organização do Pacote Model*



**Fonte: Autor**

*Figura 6 Exemplo de Get e Set da Class Cliente*

```
public class cliente {
    private int id;
    private String nome;
    private String CPF;
    private String telefone;
    private String bairro;
    private String logradouro;
    private String complemento;
    private String urlProfilePicture = "https://imgur.com/gallery/3M4hA2T";

    public int getId() {
        return id;
    }

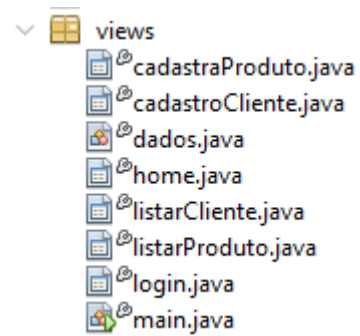
    public void setId(int id) {
        this.id = id;
    }
}
```

**Fonte : Autor**

## 6.2 View

Este é o pacote onde ficam todas as interfaces do nosso sistema, os Jframes e JPainel, são responsáveis pela interação e exibição de dados para o usuário, como mostrado a seguir:

**Figura 7 Organização do pacote Views**

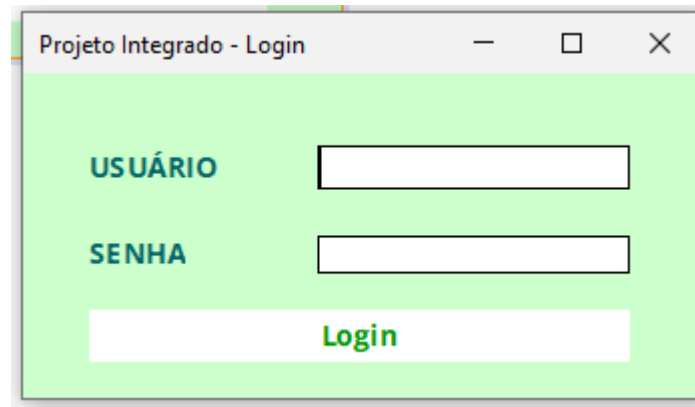


**Fonte: Autor**

### 6.2.1 Tela de login

Nesta tela é pedido a senha e o usuário, para melhor demonstração eles já são definidos no projeto como: Usuário: projeto, Senha: integrado.

**Figura 8 Tela de Login**



Projeto Integrado - Login

USUÁRIO

SENHA

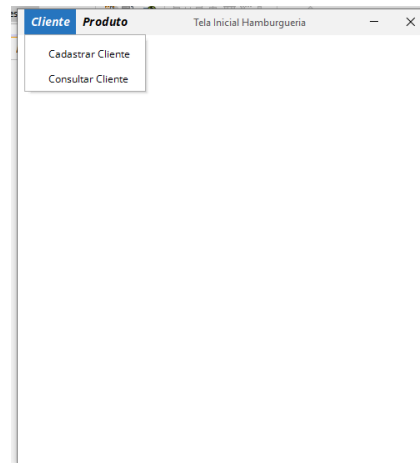
Login

**Fonte: Autor**

### 6.2.2 Tela de Menu

Após efetuar o login o usuário visualiza a tela de menu onde tem a possibilidade de escolher se vai cadastrar um nome cliente, um novo produto, ou se irá consultar um cliente ou um produto.

**Figura 9 Tela de Menu**

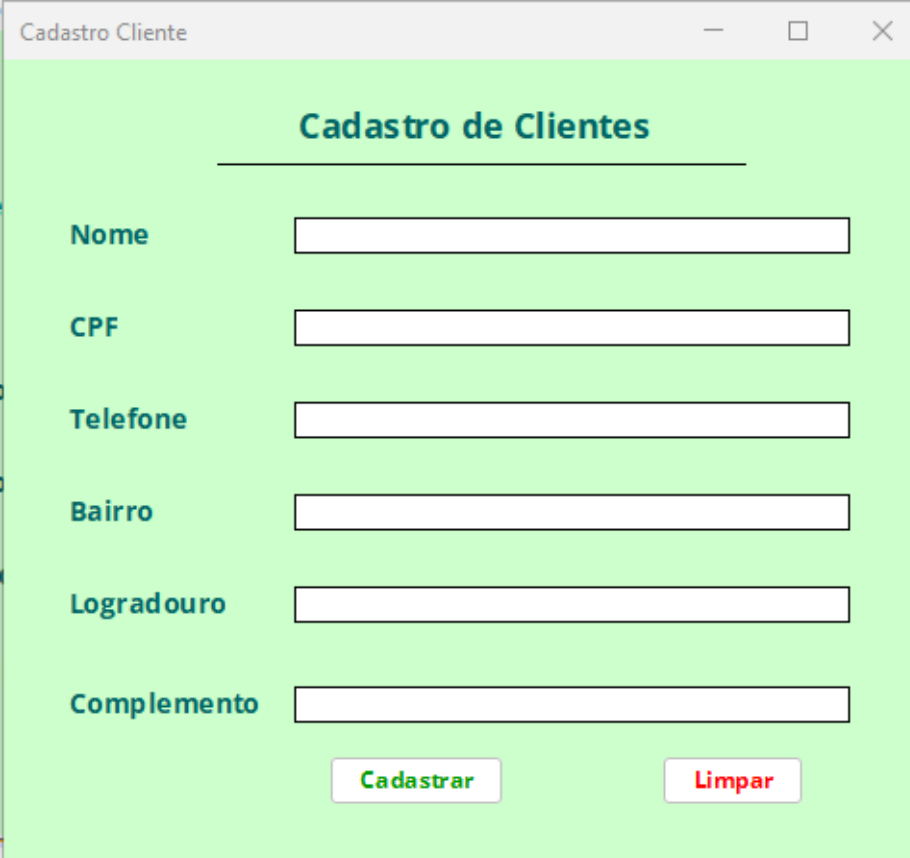


**Fonte: Autor**

### 6.2.3 Tela de cadastro de Cliente

Caso seja selecionada a opção de cadastrar clientes na tela de menu, essa tela será aberta onde o usuário insere os dados do cliente com o nome, cpf, telefone, bairro, logradouro e complemento, e apertando em cadastrar o cliente é inserido no banco de dados.

*Figura 10 Tela de Cadastro de Cliente*



Cadastro Cliente

### Cadastro de Clientes

Nome

CPF

Telefone

Bairro

Logradouro

Complemento

Fonte: Autor

#### 6.2.4 – Tela de Cadastro de Produto

Caso seja selecionada a opção de cadastrar produto na tela de menu essa tela será aberta onde o usuário insere os dados do produto como o nome, descrição e valor, e ao clicar em cadastrar o produto é inserido no banco.

*Figura 11 Tela de Cadastro de Produto*



The image shows a software window titled "Cadastro Produto" with standard window controls (minimize, maximize, close). The main content area has a light green background and is titled "Cadastro de Produtos" in bold blue text. Below the title is a horizontal line. There are three input fields: "Nome" (a single-line text box), "Descrição" (a multi-line text area), and "Valor" (a single-line text box). At the bottom, there are two buttons: "Cadastrar" (green text) and "Limpar" (red text).

**Fonte: Autor**

### 6.2.5 Tela de Consulta de Cliente

Caso o usuário escolha consultar um cliente, será aberta essa tela onde pelo telefone ou CPF é feita a busca do mesmo e retornado para o usuário, caso haja alguma alteração a ser feita, basta o usuário alterar o campo necessário e ao finalizar clicar em alterar dados que as novas informações serão armazenadas no respectivo cliente.

*Figura 12 Tela de Consulta de Cliente*

Listar Cliente

CPF (XXX.XXX.XXX-XX) ou Telefone (XXXXXXXXXX) | Buscar Cliente

**Nome do Cliente**

**CPF**

**Telefone**

**Bairro**

**Logradouro**

Rua do Cliente, Número da Residência

**Complemento**

Alterar dados | Limpar dados

Fonte: Autor

### 6.2.6 Tela de consulta de produto

Caso o usuário escolha consultar um produto, será aberta essa tela onde pelo nome é feita a busca do mesmo e retornado para o usuário, caso haja alguma alteração a ser feita, basta o usuário alterar o campo necessário e ao finalizar clicar em alterar dados que as novas informações serão armazenadas no respectivo produto.

**Figura 13 Tela de Consulta de Produto**

A imagem mostra uma janela de software com o título "Listar Produto". No topo, há uma barra de busca com o campo de texto "Nome do Produto" e um botão "Buscar Produto". Abaixo, o formulário é dividido em seções com fundo verde claro:

- Nome:** Um campo de texto vazio.
- Descrição:** Um campo de texto grande e vazio.
- Valor:** Um campo de texto com o prefixo "R\$" e o texto de ajuda "Utilize . (ponto) ao invés da , (virgula) para valores decimais".

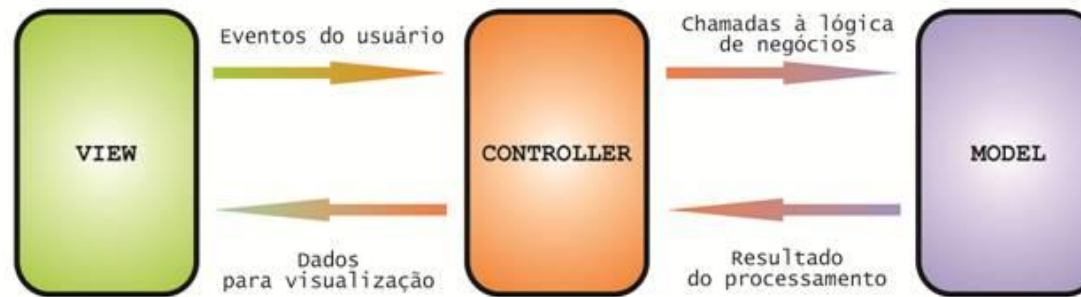
Na base da janela, há dois botões: "Alterar Produto" e "Limpar dados".

**Fonte: Autor**

### 6.3 Controller

Este Pacote é responsável por receber todas as requisições do usuário, seus métodos chamados Actions controlam qual model usar e qual será a view mostrada para o usuário

*Figura 14 Fluxo do MVC*



Fonte: fluxo do mvc - Bing images

*Figura 15 Exemplo de Controller*

```
public cliente listarCliente(cliente cliente){
    String sqlQuery = String.format(format: "SELECT * FROM clientes WHERE CPF = '%s' OR telefone = '%s'", arg

    cliente.setNome(nome: null);
    cliente.setCPF(cpf: null);
    cliente.setTelefone(telefone: null);
    cliente.setBairro(bairro: null);
    cliente.setLogradouro(logradouro: null);
    cliente.setComplemento(complemento: null);

    try (PreparedStatement stmt = connection.prepareStatement(string: sqlQuery)){
        ResultSet rs = stmt.executeQuery();

        while(rs.next()){
            cliente.setId(id: rs.getInt(string: "id"));
            cliente.setNome(nome: rs.getString(string: "nome"));
            cliente.setCPF(cpf: rs.getString(string: "cpf"));
            cliente.setTelefone(telefone: rs.getString(string: "telefone"));
            cliente.setBairro(bairro: rs.getString(string: "bairro"));
            cliente.setLogradouro(logradouro: rs.getString(string: "logradouro"));
            cliente.setComplemento(complemento: rs.getString(string: "complemento"));
        }
        stmt.close();

        return cliente;
    }catch(SQLException e){
        throw new RuntimeException(cause: e);
    }
}
```

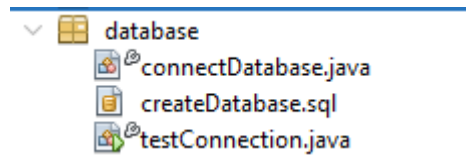
**Fonte: Autor**

## 7. Conexão com o banco

Para a conexão com o banco de dados para armazenar os dados inseridos nas interfaces, foram feitos 2 pacotes no projeto, sendo eles:

- **Database:** Este Pacote contém o script para a conexão do sistema com o banco de dados, a criação da Database e o teste de Conexão, para verificação se de fato o sistema está conectado com o banco.

*Figura 16 Organização do Pacote DataBase*



Fonte: Autor

*Figura 17 Trecho do Script de Conexão com o Banco de Dados*

```
package database;
import java.sql.*;

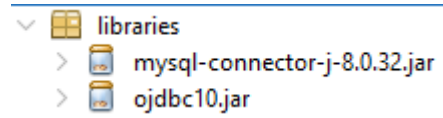
public class connectDatabase {
    // Declarada a variavel conexão que possui o tipo Connection. Essa variável será responsável pela nossa conexão
    public Connection connection = null;

    // Declaro de forma separada as variáveis responsáveis pelo Driver, Nome da Database, URL para o banco de dados:
    private final String DRIVER = "com.mysql.cj.jdbc.Driver";
    private final String DBNAME = "projeto_hamburgueria";
    private final int DBPORT = 3306;
    private final String URL = String.format("jdbc:mysql://localhost:%d/%s?autoReconnect=true&useSSL=false",
    private final String LOGIN = "root";
    private final String SENHA = "";
```

Fonte: Autor

- Libraries: Pacote que contém o MySQL Connector J e o Ojdbc10 responsáveis pela conexão do sistema com o banco

*Figura 18 Organização do Pacote Libraries*



**Fonte: Autor**

## 8. CONCLUSÃO

Este projeto tinha como objetivo inicial criar um sistema de gerenciamento de cliente e produto que tornasse mais fácil esta atividade que antes era realizada a mão. O sistema conta com várias funcionalidades e tem a vantagem de permitir que o usuário tenha as informações de seus clientes e seus produtos em questão de segundos.

Outra vantagem do sistema é a fluidez e simplicidade com que pode ser realizado os cadastros e consultas.

Ao decorrer do projeto encontramos alguns problemas como versão do Java diferentes, problemas com a IDE e conexão do banco com o sistema, mas após muita pesquisa os problemas foram resolvidos, resultando assim num sistema fiel ao proposto pelo Pi e que foi idealizado no começo.

O sistema se encontra como o previsto, porém é possível implantar várias melhorias como controle de venda, de estoque, fornecedores entre outros,

Ao decorrer do projeto podemos colocar em prática todos os conhecimentos adquiridos de POO como herança, polimorfismo, encapsulamento, o conceito de classe e objeto entre outros conceitos, assim como os conhecimentos adquirido em Banco de dados como, análise de requisitos, criação de modelos, aplicação de chave primeira e relacionamentos entre as tabelas.

Podemos desenvolver também na comunicação com os colegas de grupo, trabalho em equipe, desenvolver comprometimento com as tarefas, visão sistêmica

## 9 Referências

### Works Cited

Bauru, Viven Do. “Quais São as Linguagens de Programação Orientada a Objetos?” *Vivendobauru.com.br*, 7 Sept. 2022, [www.vivendobauru.com.br/quais-sao-as-linguagens-de-programacao-orientada-a-objetos/](http://www.vivendobauru.com.br/quais-sao-as-linguagens-de-programacao-orientada-a-objetos/). Accessed 3 Apr. 2023.

*Modelagem de Bancos de Dados: Conceitual, Lógica E Física*. 19 Sept. 2017, [spaceprogrammer.com/bd/introducao-ao-modelo-de-dados-e-seus-niveis-de-abstracao/](http://spaceprogrammer.com/bd/introducao-ao-modelo-de-dados-e-seus-niveis-de-abstracao/).

“POO: O Que é Programação Orientada a Objetos?” *Alura*, [www.alura.com.br/artigos/poo-programacao-orientada-a-objetos](http://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos).

Reis, Fábio dos. “O Modelo Entidade-Relacionamento: Introdução.” *Bóson Treinamentos Em Ciência E Tecnologia*, 11 Oct. 2020, [www.bosontreinamentos.com.br/modelagem-de-dados/o-modelo-entidade-relacionamento-introducao/](http://www.bosontreinamentos.com.br/modelagem-de-dados/o-modelo-entidade-relacionamento-introducao/).