



CENTRO UNIVERSITÁRIO DA FUNDAÇÃO EDUCACIONAL INACIANA
PADRE SABÓIA DE MEDEIROS

ARQUITETURA DE SOFTWARE E PROGRAMAÇÃO ORIENTADA A OBJETOS

PROJETO FEIFOOD

GUILHERME SILVA XAVIER – 24.224.025-7

NOVEMBRO/2025



Projeto: PROJETO FEIFOOD – baseado no iFood

Título: Projeto orientado na construção de um executável em linguagem Java simulando aplicativo de comércio de comida

Ref: Produto final – 13/11/2025

Resumo: Este relatório visa a compreensão dos métodos, funcionamento da lógica e o programa, que surgem para qualificar o aluno mediante as aulas ministradas da disciplina **ARQUITETURA DE SOFTWARE E PROGRAMAÇÃO ORIENTADA A OBJETOS**, em objetivo de capacitar o aluno a desenvolver soluções baseadas nos conceitos de orientação a classes. Igualmente abordando conceitos de banco de dados, lógica de programação e estrutura de software.



Sumário

1. INTRODUÇÃO	4
2. FUNDAMENTAÇÃO TEÓRICA	6
3. RESULTADOS	Erro! Indicador não definido.
4. CONCLUSÕES	Erro! Indicador não definido.
5. REFERÊNCIAS	Erro! Indicador não definido.

Uma breve salva neste projeto a minha futura esposa e meu amor de todos os dias, eu te amo Wemilly Nunes...



1. INTRODUÇÃO

Este relatório refere-se ao último entregável definido no escopo da disciplina de computação estabelecida no Centro Universitário FEI-SBC para realizar o processo de construção de software FEIFOOD – um simulador de uma plataforma de comércios digitais – compreendendo o objetivo de integrar uma proposta realista às habilidades desenvolvidas em aula.

O objetivo do projeto é a construção completa de um sistema para resolver a finalidade principal de comércios. Para isso, os critérios principais são em solucionar questões básicas listadas.

- Login
- Cadastro
- Buscar por Alimentos
- Listar Informações de Alimentos
- Criar um Pedido
- Deletar Pedido
- Adicionar, remover alimento da lista de pedido

Deste modo, é necessário aplicar os conceitos de diagrama de classe, heranças, polimorfismo e mesmo normalização de banco de dados. Diagrama de classes abaixo ilustrando as classes:



Diagrama de Classes

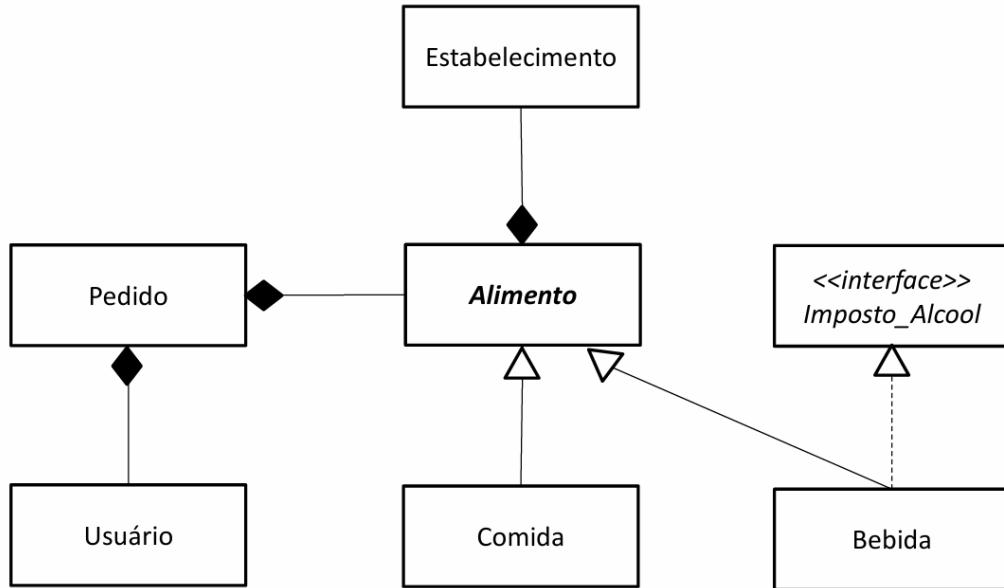


Imagen 1

Assim sendo o escopo do projeto:

Testar as capacidades do aluno em desenvolver sistema simulando uma proposta real baseado em problemas sólidos atuais.

Aplicar os conhecimentos teórico compreendidos em aula para solução de problemáticas na construção de aplicações algorítmicas.

Cumprir os requisitos solicitados em uma possível proposta de cliente.

Integração de outros conhecimentos em diferentes disciplinas para complementar o processo sólido de construção de algoritmos.



2. FUNDAMENTAÇÃO TEÓRICA

Os tópicos principais concebidos para o desenvolvimento do projeto regem partes básicas de qualquer projeto.

1. Arrays, entrada de dados, classes e objetos

Para este projeto foi utilizado ferramentas como ‘ArrayList’ para armazenar valores em índices e acessa-los em um contador e de acordo com os métodos de GETs e SETs

Imagen 2

```
resultadoBusca = statement.getResultSet();
ArrayList<alimentos> alimentos = new ArrayList<>();
principal.getTelai().setText("");
while(resultadoBusca.next()){
    alimentos.add(new Comida(resultadoBusca.getString("nome"), resultadoBusca.getString("local"),
    resultadoBusca.getDouble("preco"),null));
```

No ilustrado, é um exemplo da integração entre declarações de objeto, uso de vetor para armazenamento e métodos de GET.

Métodos GET e SET são utilizados para obter e segurar entrada de valores para atribuir em atributos e acessa-las em outras classes.

2. Heranças, classes abstratas e diagrama de classes

A diante com os conhecimentos tratados em aula, se estende para classes abstratas que são objetos que não são declarados diretamente mas seus atributos são passados em herança para polimorfismo.

Imagen 3

```
public class Comida extends alimentos{
    private String tipoComida;

    public String getTipoComida() {
        return tipoComida;
    }

    public Comida(String nome, String local, double preco, String tipoComida) {
        super(nome, local, preco);
        this.tipoComida = tipoComida;
    }

}
```

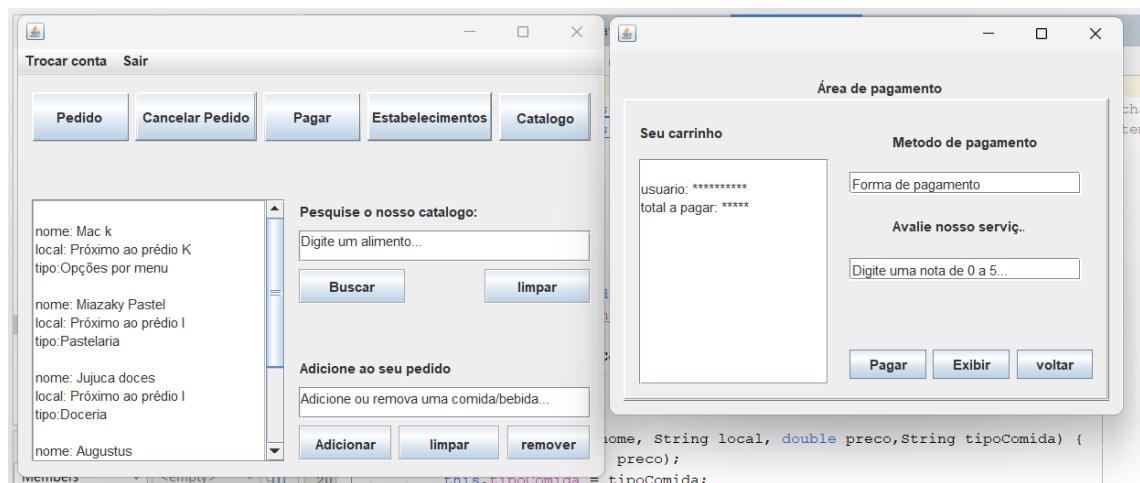


No exemplo, comida vai estender da classe alimento (abstrata) e seus atributos são utilizáveis em diferentes classes.

3. Interface gráfica (GUI) e arquivos DAO

Por fim aplicando os últimos conhecimentos para construir interfaces para o usuário conseguir compreender o sistema e utiliza-lo adequadamente, os JFrame form são capazes de trazer objetos como botões e painéis para facilitar o desenvolvimento do programador e o uso do cliente sobre o software conforme a imagem 4. Os arquivos DAO por sua vez são responsáveis por diretamente consultarem o banco de dados com os comandos SQL.

Imagen 4



Arquivos DAO vão precisar de biblioteca de ‘STATEMENTS’ para executar as chamadas ‘querys’ de consulta de dados das tabelas no banco de dados. Conforme a imagem 5.



```
public ResultSet obterDadosAlimentos(String alimento, String nomeUsuario) throws SQLException{
    String sql = "select * from alimentos;";
    PreparedStatement statement = conexao.prepareStatement(sql);
    statement.execute();
    resultadoBusca = statement.getResultSet();
    while(resultadoBusca.next()){
        if(alimento.equals(resultadoBusca.getString("nome"))){
            produto = resultadoBusca.getString("nome");
            estabelecimento = resultadoBusca.getString("local");
            preco = resultadoBusca.getDouble("preco");
            usuario = nomeUsuario;
        }
    }
    return resultadoBusca;
}
```

Imagen 5

4. Resultados e conclusões

Por fim, cabe a conclusão em dizer sobre o projeto da disciplina avalia as capacidades básicas em solucionar montantes de problemas que vão ocorrer em situações não-esporádicas. Adicionalmente, é explorado a capacidade do aluno em buscar soluções além das problemáticas entendidas em aulas em situações de eventualidades, moldando a perspectiva do aluno em combinar resoluções, eficiência, versatilidade e usabilidade do sistema.

5. Referências bibliográficas

1. **SITE OFICIAL JAVA.** *Java.com/pt-BR/*. Disponível em: <https://www.java.com/pt-BR/>
Acesso em: 29 set. 2025.
 - o **Usado para:** Buscar como funcionam os métodos e heranças.