

Relatório projeto de software MYFOOD

André Aparecido Monteiro de Araújo

As principais classes são Usuario, Empresa, Produto e Pedido, são responsáveis por representar essas entidades, armazenando suas informações e servindo como base para operações no sistema. Classes de controle, como ControleUsuarios, ControleEmpresa, ControleProdutos e ControlePedido, são responsáveis pela manipulação e validação dos dados dessas entidades, realizando operações como criação, consulta e edição. A classe XmlPersistencia gerencia a persistência de dados, salvando e carregando informações em arquivos XML.

Classe Usuario representa uma entidade genérica para todos os usuários do sistema. Sua função principal é servir como molde e fornecer informações sobre os usuários, como seus dados pessoais e tipo de usuário. É uma classe base abstrata para tipos específicos de usuários, como clientes ou donos de estabelecimento. Ela define os atributos comuns e métodos que serão herdados por subclasses específicas.

Atributos id, nome, email, senha, e seus respectivos métodos setter e getter e seu construtor. Além dos métodos setter e getter foi implementado metodos usuarioCliente(), usuarioDonoEmpresa() ambos tipo booleano e tem retorna falso para ser subscrito nas classes que herdaram da classe usuario, objetivos deste dois métodos é facilitar a validação de usuários quando necessária.

As Classes **Cliente** e **DonoEstabelecimento** são subClasses de **Usuario()**, esta classe herda os atributos e métodos de Usuario e adiciona um atributo específico e subscrito os métodos usuarioCliente(), usuarioDonoEmpresa() alterando seu valor para verdadeiro.

ControleUsuarios serve para gerenciar os dados dos usuários no sistema. Ela realiza a autenticação e validação das informações dos usuários antes de cadastrá-los, garantindo que dados essenciais como nome, email, senha e endereço sejam válidos. A classe permite a consulta de atributos específicos dos usuários registrados. Os principais métodos desta classe é criarUsuarios()

responsável pela criação de usuário tipo cliente e dono do estabelecimento, e o método **buscarAtributos()** responsável pelas buscas.

A classe **Empresa** representa uma entidade genérica para todas as empresas no sistema. Sua função principal é servir como molde e fornecer informações gerais sobre as empresas, como identificador, nome, dono, endereço e tipo de empresa. É uma classe base para tipos específicos de empresas, como **Restaurante**, **Mercado** e **Farmacia**. Os atributos são id(empresa), nome(Nome da empresa), donoID, endereco, tipoEmpresa e seu construtor e respectivos métodos setter e getter.

A classe **Restaurante** é uma subclasse de **Empresa** e representa uma empresa do tipo restaurante. Adiciona um atributo específico para armazenar o tipo de cozinha oferecido pelo restaurante. Pensando em implementações futuras foi criada as classes **Mercado** e **Farmacia** elas porém só recebe os atributos de **Empresa**.

ControleEmpresa é a classe responsável pela gestão das empresas no sistema. Ela realiza operações como criação, busca de empresas, além de garantir que as empresas sejam cadastradas corretamente de acordo com as regras estabelecidas. Ela realiza a autenticação e validação das informações dos usuários antes de cadastrá-los.

A classe **Produto** representa um item que pode ser vendido por uma empresa. Armazena informações sobre o produto, incluindo seu identificador, a empresa a qual pertence, nome, valor e categoria.. Sua função principal é servir como molde e fornecer informações sobre os produtos.

Os atributos são id(Identificador único do produto), empresaID(Identificador da empresa à qual o produto pertence), nome(nome do produto), valor e categoria e seus respectivos métodos getters, setters e construtor.

ControleProdutos é responsável pela gestão dos produtos no sistema. Ela realiza operações como criação, edição, consulta e listagem de produtos, garantindo que as informações sejam válidas e que não haja produtos duplicados.

seus métodos principais são criarProduto que é responsável por criar um novo produto com base nas informações fornecidas. Valida os dados e verifica se o produto já existe para a empresa especificada, editarProduto edita um produto existente e atualiza o produto no sistema, já getProduto realiza busca através de um atributo específico do produto com base no nome e na empresa listarProdutos lista

todos os produtos associados à empresa especificada, formatando a lista como uma string.

A classe **Pedido** representa um pedido feito por um cliente a uma empresa. Armazena informações sobre o pedido, incluindo identificador, cliente, empresa, estado, lista de produtos e valor total. Os atributos são pedidoID: (Identificador único do pedido), clienteID(Identificador do cliente que fez o pedido), empresaID: (Identificador da empresa para a qual o pedido foi feito), estado(Estado do pedido ex: aberto, preparando fechado), produtos(Lista de produtos incluídos no pedido) e valor(Valor total do pedido). os métodos são os Construtor que Inicializa todos os atributos da classe, e os métodos getter e setter.

A classe **ControlePedido** gerencia os pedidos no sistema, realizando operações como criação, adição e remoção de produtos, fechamento de pedidos, e consulta de informações dos pedidos. Tem como métodos o Construtor que Inicializa os atributos e define o contador de IDs com base no número de pedidos existentes, gerarNovoId gera um novo identificador único para o pedido, criarPedido Cria um novo pedido para o cliente e empresa especificados, verificando se já existe um pedido em aberto para a mesma empresa, getNumeroPedido e getPedidos responsáveis por buscas em pedidos, adicionarProduto adiciona um produto ao pedido especificado. Verifica se o pedido está aberto e se o produto pertence à empresa do pedido, fecharPedido Altera o estado de um pedido para "preparando", indicando que o pedido está sendo preparado, e removerProduto é responsável por remover um produto do pedido especificado, se o pedido estiver aberto. Atualiza o valor total do pedido após a remoção do produto, atualizarValorPedido calcula e atualiza o valor total do pedido com base nos produtos incluídos.

ControleDados é a classe que gerencia o armazenamento e recuperação de dados do sistema, incluindo usuários, empresas, produtos e pedidos. Possui Atributos, usuarios(mapa de IDs de usuários para instâncias de Usuario), empresas (mapa de IDs de empresas para instâncias de Empresa), produtos (Mapa de IDs de produtos para instâncias de Produto), pedidos(Mapa de IDs de pedidos para instâncias de Pedido) e os métodos de persistência e carregamento para usuários, empresas, produtos e pedidos, utilizando a classe **XmlPersistencia** para salvar e carregar dados de arquivos XML.

XmlPersistencia é a classe responsável pela persistência de dados do sistema em arquivos XML. Ela oferece métodos para salvar e carregar informações sobre usuários, empresas, produtos e pedidos. Após a leitura dos arquivos XML, eles são limpos para evitar reproprocessamento dos dados. A classe inclui tratamento de erros para lidar com exceções durante a leitura e escrita dos arquivos. Suas principais funcionalidades são **salvarUsuários** (grava detalhes dos usuários (ID, nome, email, senha) em XML, adicionando informações extras conforme o tipo de usuário), **carregarUsuários** (onde se lê e cria objetos de usuário a partir de XML, diferenciando entre Cliente e DonoEstabelecimento), **salvarEmpresas** (Armazena informações sobre empresas (ID, nome, ID do dono, tipo) em XML, com detalhes específicos para cada tipo de empresa), **carregarEmpresas** (responsável por ler e cria objetos de diferentes tipos de empresas a partir de XML), **salvarProdutos** (grava detalhes dos produtos (ID, nome, valor, categoria) em XML, vinculados a uma empresa), **carregarProdutos** (lê e cria objetos de produto a partir de XML), **Salvar Pedidos** (armazena informações sobre pedidos (ID do cliente, da empresa, estado, valor total, produtos) em XML), **carregarPedidos** (lê e cria objetos de pedido a partir de XML, incluindo todos os produtos associados).

A classe **login** verifica se o email e a senha fornecidos correspondem a um usuário registrado. Retorna o ID do usuário se as credenciais estiverem corretas, caso contrário, lança uma exceção. A classe **Autenticar** tem como função validar e verificar alguns dados relacionados à autenticação e criação de entidades. Na classe Facade é usada basicamente para instanciar as outras classes e para os métodos **encerrarSistema** e **ZerarSistema**.