

**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS – CAMPUS SÃO JOÃO EVANGELISTA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

Felipe Fernandes Alves
Rodrigo Alexandre Alves

TRABALHO PRÁTICO

SÃO JOÃO EVANGELISTA
2022

Felipe Fernandes Alves
Rodrigo Alexandre Alves

Sistema de uma lanchonete

SUMÁRIO

1.	INTRODUÇÃO.....	4
	1.1. Objetivo Geral	4
	1.2. Objetivos Específicos	4
	1.3. Justificativa	4
2.	DESENVOLVIMENTO.....	4
	2.1. Sistema de uma lanchonete.....	5
	2.2. Implementação	6
3.	CONCLUSÃO	7

1. INTRODUÇÃO

O presente trabalho proposto pela disciplina de Linguagem de Programação I para o desenvolvimento de um sistema de uma lanchonete/restaurante. O trabalho consiste na elaboração da interface gráfica e na manipulação do banco de dados.

1.1. Objetivo Geral

Este trabalho tem como objetivo geral desenvolver uma aplicação de uma lanchonete, utilizando a linguagem de programação Java e o editor de código Netbeans.

1.2. Objetivos Específicos

Esse trabalho tem como objetivos específicos:

- Fixar conceitos sobre orientação a objetos e interface gráfica com swing (Java).
- Apresentar o diagrama entidade relacionamento
- Criar o pacote “Visual” contendo um formulário principal e uma tela de Cadastro e uma tela de Login o modelo do projeto.

1.3. Justificativa

Esse trabalho é justificado como principal trabalho da disciplina além de servir como uma atividade de fixação de toda a matéria vista durante o período.

2. DESENVOLVIMENTO

Inicialmente o código foi desenvolvido para que qualquer pessoa entenda apenas lendo os comentários.

Este software deverá disponibilizar as seguintes informações para cadastro: nome, número do CPF, código único de identificação, telefone para contato, salário e sexo.

Para acessar o sistema é necessário fazer um login no sistema, onde se o usuário digitar o login errado vai retornar uma mensagem de erro, caso contrário o usuário será redirecionado para a tela principal.

Os funcionários criam um pedido, o pedido contém as seguintes informações: um código de identificação do pedido, um código exclusivo do funcionário, o valor total e uma descrição.

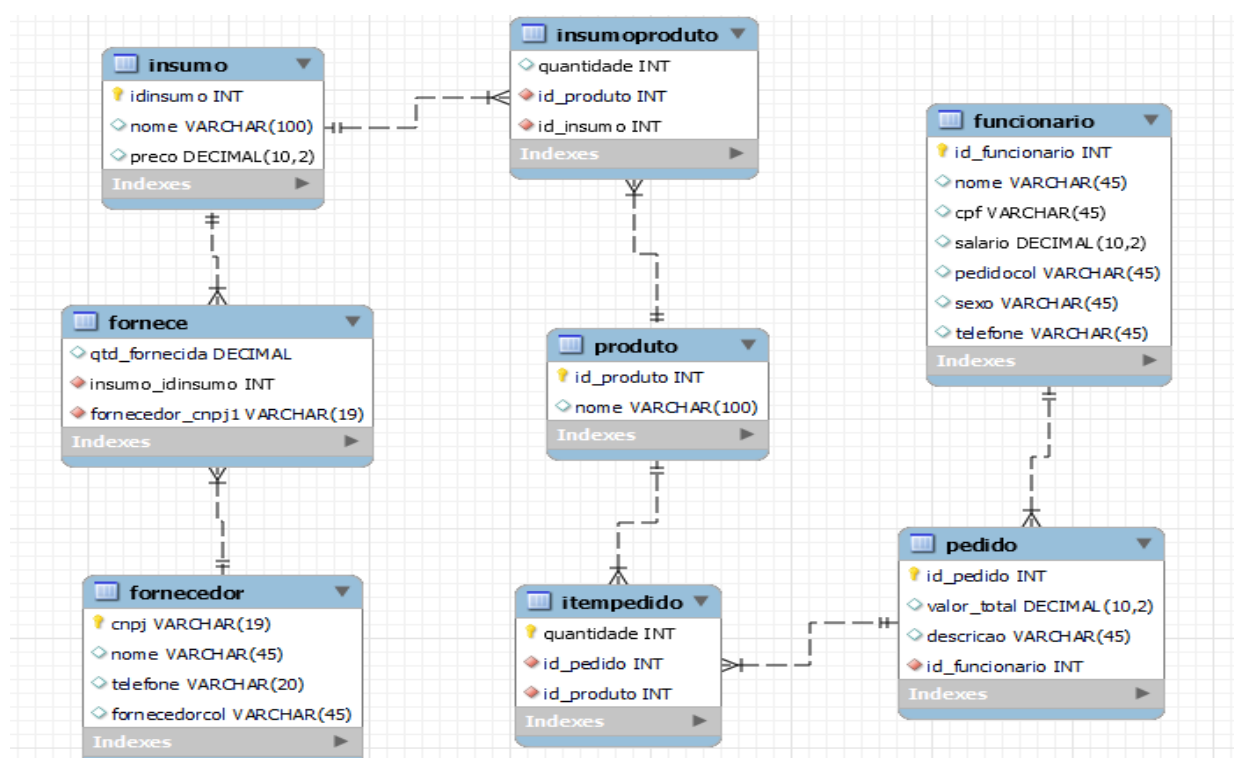
Para cada produto, deve ter vários itens para produção, a quantidade usada em cada item, o nome do item, a quantidade armazenada e o preço pago por ele também devem ser registrados. Além disso, devem armazenar o ID do item, a

quantidade que foi fornecida e a pessoa que o forneceu, devem também armazenar o CNPJ, o nome e o telefone do fornecedor.

Para ter total controle sobre cada componente do pedido, é necessário salvar o código de identificação do pedido, o código de identificação do produto e a quantidade do produto.

2.1. Diagramas

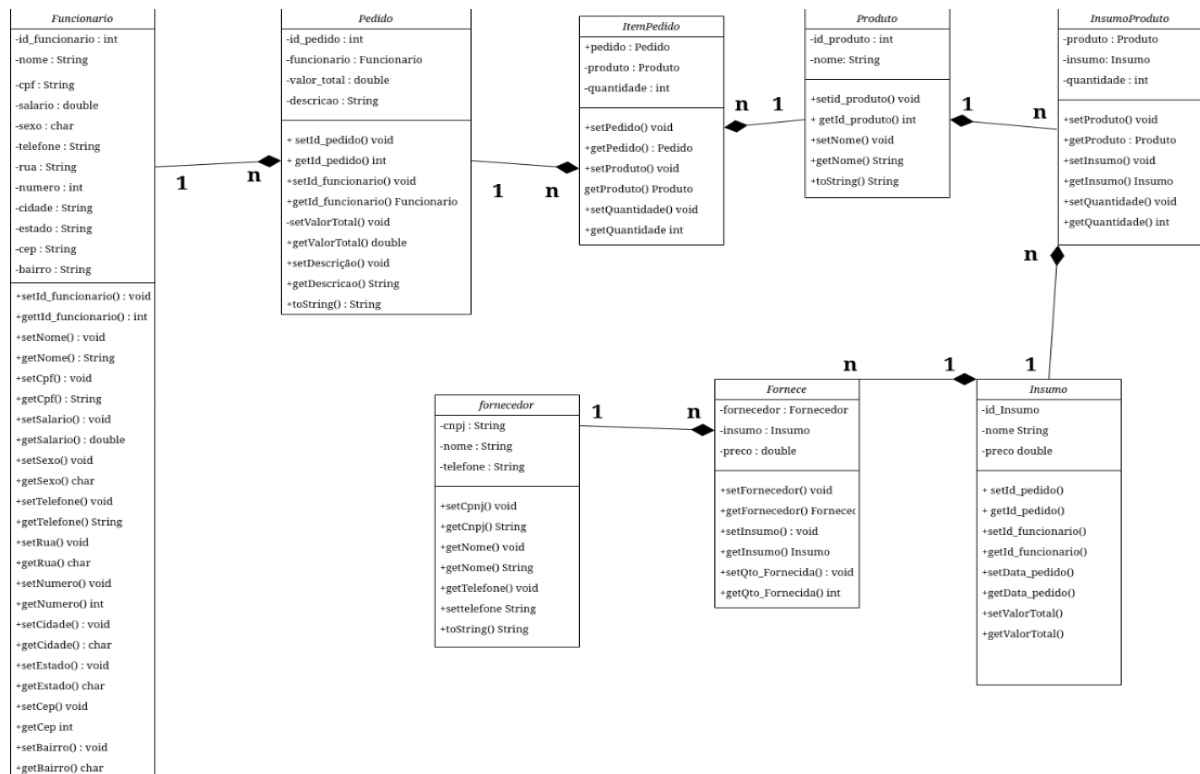
Figura 1 – Diagrama de entidade e relacionamento



A Figura 1 mostra as tabelas do banco de dados para a aplicação de desktop. A tabela "pedido" armazena informações sobre os pedidos que foram feitos e, adicionalmente, possui uma chave estrangeira de "funcionário" que identifica qual funcionário fez um determinado pedido. A tabela "itempedido" é uma tabela intermediária entre "Produtos" e "Pedidos", e além de conter a quantidade de produtos de um pedido, também é utilizada para identificar quais produtos estão em um determinado pedido. A tabela "produto" é utilizada para armazenar informações sobre os produtos finais que serão vendidos. A tabela Funcionários armazena os dados mais relevantes dos funcionários. A tabela intermediária "insumoproduto" serve para saber quantos e quais insumos compõe um produto final. A tabela "Fornecedores" é utilizada para armazenar as informações dos fornecedores de insumos da lanchonete. A tabela intermediária

“fornece” armazena a quantidade de produtos fornecida pelos “fornecedores”, além servir para identificar os produtos que foram fornecidos por um mesmo provedor.

Figura 2 – Diagrama de classes



A Figura 2 representa o diagrama de classes que segue a implementação do programa. A classe "Fornece" é utilizada para armazenar a quantidade de produtos fornecidos por cada fornecedor. A classe "Pedido" armazena as informações que um pedido deve conter e também possui uma chave estrangeira de "Empregado" para identificar qual funcionário fez um determinado pedido. A classe "Produto" armazena um id e um nome de produto para identificar exclusivamente o produto. A classe "Funcionário" representada é usada para armazenar as informações mais importantes sobre o funcionário. A classe "ItemPedido" é utilizada para armazenar a quantidade do produto no pedido. A classe "Fornecedor" é usada para armazenar informações do fornecedor. A classe "InsumoProduto" é usada para armazenar a quantidade de entrada para cada produto. A classe "Insumo" é usada para arquivar informações de produtos adquiridos de fornecedores.

2.2. Implementação

Para implementação das interfaces gráficas foi utilizado o pacote Javaw.SWING.

Para o desenvolvimento das telas foram utilizados Form JFrame e Form JPanel.

Para realizar a conexão com o banco de dados foi criado um pacote "Conexao".

Para armazenar as informações no banco de dados, as classes do pacote "DAO" devem ser modificadas. No método "getLista" existe uma variável para armazenar a consulta SQL executada no banco de dados, um ArrayList para retornar a lista de dados do banco de dados, uma variável do tipo PreparedStatement para armazenar a conexão com o banco de dados e uma última variável de ResultSet para armazenar os resultados da consulta. As duas últimas variáveis são criadas dentro do bloco try catch, pois se alguma exceção for tratada, desde que haja dados na tabela no banco de dados, isso será feito por um tempo dentro deste bloco, e existe uma variável "obj" nela. O tipo criado tem acesso ao método que irá armazenar os dados do banco de dados, ao final desta variável é adicionada à lista, e a lista é retornada ao final do método.

A consulta SQL é modificada de acordo com a tabela que está sendo acessada, e o tipo da variável "obj" também é modificado de acordo com o pacote DAO em que ela está. O método "incluir" possui uma consulta SQL que realiza o insert, cria uma variável do tipo PreparedStatement no bloco try que recebe a conexão com o banco de dados e em seguida adiciona o valor digitado pelo usuário ao banco de dados. O método "atualizar" é muito parecido com o método anterior, a única diferença é que ele possui uma consulta SQL que realiza a atualização. O método "salvar" recebe como parâmetro uma variável do tipo Object que é convertida em uma variável do tipo encontrada na classe. Este método apenas compara se a chave primária é igual a "null", se sim, significa que os dados estão sendo inseridos, e se não for igual a "null", significa que os dados estão sendo atualizados. O método "localizar" também tem uma consulta SQL e nela é procurado um valor específico que é definido de acordo com a necessidade do usuário. O método "remover" contém uma consulta SQL que realiza um delete na tabela. O método "localizarEspecifico" no bloco try ao verificar se existe próximo e isso for falso a função retorna "null" ao invés de apresentar uma mensagem com no método "localizar", isso teve de ser feito porque não estava sendo possível preencher as tabelas intermediárias do modo convencional.

3. CONCLUSÃO

O objetivo principal deste trabalho era o desenvolvimento da aplicação desktop para o gerenciamento de uma lanchonete. Nesse sentido, foi posto em prática o conteúdo visto nas aulas de Linguagem de Programação 1, ministrada pelo professor Eduardo Trindade, não somente isso, mas também buscando conhecimento por fora e discussão entre colegas.

Podemos dizer que este trabalho foi composto por duas partes, interfaces gráficas, em que foi necessário a utilização do pacote Swing, este que é a biblioteca gráfica padrão do Java. E também a implementação de um banco de dados, em que foi utilizado SQL (Structured Query Language), é uma linguagem padrão para trabalhar com bancos de dados relacionais.

Foram apresentados alguns erros da classe “formPedido”, em que não deu tempo suficiente de corrigi-los, de uma forma geral, o trabalho foi concluído.