

# João Atacadão

## A sua loja de Departamentos

Lucas de Almeida Bandeira Macedo, 19/0047089

João Pedro Felix de Almeida, 19/0015292

João Víctor Siqueira Araujo, 19/0031026

<sup>1</sup>Dep. Ciência da Computação – Universidade de Brasília (UnB)  
CiC 0197 - Técnicas de Programação 1 - Grupo 02

190047089@aluno.unb.br, 190015292@aluno.unb.br, 190031026@aluno.unb.br

**Abstract.** *This report covers all the construction phases of the Group 2 project for the discipline Programming Techniques 1, Class B, at the Universidade de Brasília in the semester 2020/1. Here, the initial phases of software planning, the definition of business rules and final implementation will be covered, as well as considerations about what was changed in the initial plan.*

**Resumo.** *O presente relatório aborda todas as fases de construção do projeto do Grupo 2 da disciplina de Técnicas de Programação 1, Turma B, da Universidade de Brasília no semestre 2020/1. Aqui serão abordadas tanto as fases iniciais de planejamento de software, definição das regras de negócios e implementação final, além de considerações acerca daquilo que foi alterado no planejamento inicial.*

## 1. Introdução

Lojas de departamento sempre foram uma peça chave na economia do país. Pensando nisso, nos inspiramos em algo que raramente as pessoas notam, mas que é essencial para o bom atendimento de qualquer consumidor: o *software* de caixa. Neste trabalhamos implementamos, a partir do uso do paradigma de Orientação a Objetos, um programa que simula um sistema de caixa, porém adicionando diversas outras funções como a de controle de estoque e de gerenciar funcionários da empresa. Um programa, diversas ferramentas.

### 1.1. Objetivos

Inicialmente nossos objetivos se centraram na criação de três blocos essenciais para o funcionamento do programa: uma estrutura hierárquica para a superclasse produto que possuiria diversas classes filhas que tornariam cada produto único, uma maneira prática de se trabalhar com um pseudo banco de dados em arquivo de texto e a interface com o usuário através de recursos gráficos Java Swing.

### 1.2. Recursos

Durante a implementação desse projeto utilizamos os seguintes recursos:

- *Dia Diagram Editor*;
- *Apache NetBeans IDE*;
- *GitHub*;

## 2. Desenvolvimento

### 2.1. Descrição do Problema

O programa que implementamos consiste basicamente em um gerenciador de tarefas para uma loja de departamentos. Nosso objetivo foi desenvolver um *software* capaz não apenas de processar os dados lidos de um banco de dados a partir de seu código de barras para processar uma compra mas também de oferecer facilidades tanto para os clientes quanto para os gestores do negócio. Para tal, foram criados recursos capazes de gerir os funcionários da empresa, clientes participantes do programa de fidelidade da loja, além de suporte total ao controle de estoque.

### 2.2. Regras de Negócio

Para o nosso programa, foram considerados as seguintes regras de negócio:

- Existem campos para o cadastro de Cliente, Funcionário e Produto;
- O programa só funcionará plenamente caso haja pelo menos 1 cliente, 1 gerente e 1 produto já cadastrados;
- Todo cliente necessita de um Cartão Fidelidade para realizar uma compra;
- Apenas um cliente é atendido por vez no caixa;
- O cliente pode pedir um desconto de 10% no valor da compra pela comemoração de seu aniversário;
- A compra só deve ser efetuada se o saldo no Cartão Fidelidade for suficiente para realizar o pagamento;
- O caixa deve ser operado por um funcionário qualquer por vez, podendo ser ou não um gerente;
- Gerente é um funcionário da loja, mas com uma senha de acesso;
- A exclusão de produtos do banco de dados ou do carrinho deve ser realizada somente sob autorização do gerente da loja;
- Qualquer funcionário pode ver os produtos salvos no banco de dados;
- O carrinho só contém produtos cadastrados e salvos no banco de dados, podendo conter repetições;
- Todo objeto possui uma identificação (para pessoas é o CPF e para produtos é o Código de Barras);
- Os códigos dos produtos seguem uma convenção para o primeiro caractere:
  - A ou a para Celulares.
  - B ou b para Computadores.
  - C ou c para Eletroeletrônicos.
  - D ou d para Filmes.
  - E ou e para Periféricos.
  - F ou f para Vestuário.
  - G ou g para Livros.

## 2.3. Implementação

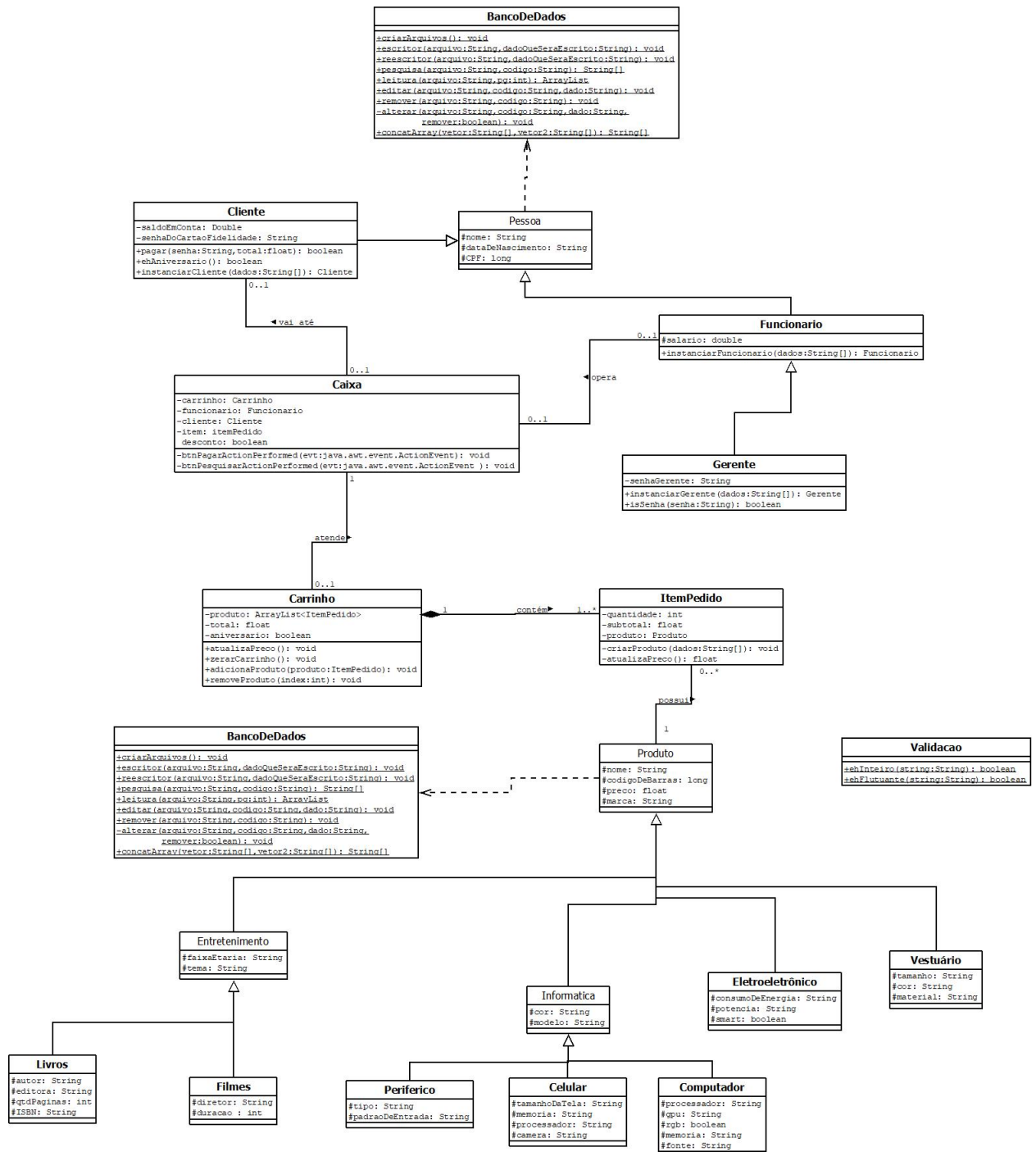


Figura 1. Diagrama de Classes

### 2.3.1. Banco de Dados

O banco de dados criado para o projeto não foi feito da maneira clássica (usando JDBC ou SQL), fizemos um banco apenas com arquivos textos (.txt). O banco de dados consiste em 9 arquivos, 7 arquivos para os departamentos de produtos (eletroeletrônico, vestuário,

etc), um arquivo para registro de cliente e o último para registro de funcionários. Criamos uma classe com a exclusiva finalidade de manipular nosso banco, que tem 7 métodos públicos (para uso nas outras classes) e 2 métodos privados (para uso interno da classe). Todos os métodos são estáticos, para que possamos manipular qualquer arquivo, em qualquer classe.

Todos os métodos públicos foram feitos para lidar diretamente com os arquivos textos. Temos um método para criar todos os arquivos (e não fazer nada, caso já existam), dois para escrever no banco (um para adicionar texto ao final, e o outro para reescrever completamente o arquivo), um para pesquisar determinado objeto em determinado banco, outro para ler todos os arquivos incondicionalmente (foi feito, inclusive, com paginação, de modo que os objetos são exibidos de 10 em 10), um método para alterar determinado elemento da tabela, e o último para deletar o objeto do banco.

Todo o banco de dados segue um padrão, parecido com o padrão JSON. Entre os atributos temos uma vírgula, entre o nome do atributo e o próprio atributo, temos dois pontos, e ao final do objeto, temos ponto e vírgula. Aqui está um exemplo simples:

Código:1234,

Nome:Mouse sem Fio,

Preço:12.50;

A partir dessa organização, conseguimos usar o método "useDelimiter" presente na Scanner para iterar ao longo do nosso banco, facilitando muito o trabalho.

### 2.3.2. Produto

O propósito desse conjunto de classes formado a partir da superclasse produto é o de facilitar a manipulação dos dados dentro do contexto do processamento das compras no caixa.

Inicialmente definimos a superclasse produto como abstrata, afinal, ela por si só não apresenta características o suficiente para descrever um produto por completo. Em cada nova camada, isto é, a cada classe que é derivada da produto, adicionamos novos atributos que fazem com que cada uma possua traços únicos que as diferenciem entre si.

As últimas classes derivadas de produto, isto é, as classes folhas se visualizarmos toda a estrutura como uma espécie de árvore, são, finalmente, classes concretas e que portanto podem ser instanciadas. A herança nessa parte do código está muito mais atrelada a atributos em comum do que métodos especificamente, ainda sim fomos capazes de aproveitar das vantagens oferecidas pelos métodos abstratos, não necessitando por exemplo de realizar *castings* nas chamadas de métodos de um *ArrayList* de produtos.

### 2.3.3. Pessoa

Temos, no nosso programa, um total de três tipos de pessoa: um funcionário, um gerente, e o cliente. O funcionário servirá como a pessoa que estará associado ao caixa como operador, e cuidará do funcionamento deste. O gerente herdará o funcionário, mas com

um privilégio extra: a possibilidade cancelar um pedido de cliente. Caso um funcionário tente cancelar um produto, ele não conseguirá e deverá chamar um gerente para inserir suas credenciais.

Por último, o cliente é o mais diferente dos três. Como nosso estabelecimento funciona obrigatoriamente com um cartão fidelidade, é obrigatório que o cliente tenha um saldo e uma senha para o cartão. Sempre que o cliente fizer uma compra, o valor total do pedido será descontado diretamente do cartão.

#### **2.3.4. Caixa**

O caixa é onde o pagamento é feito, é onde temos a famosa tela em que vemos todos os mercados, onde mostra o "Total", a lista de produtos registrados até o momento, etc. O caixa funciona obrigatoriamente associado a um funcionário, que operará o caixa, e um cliente, que está sendo atendido no momento.

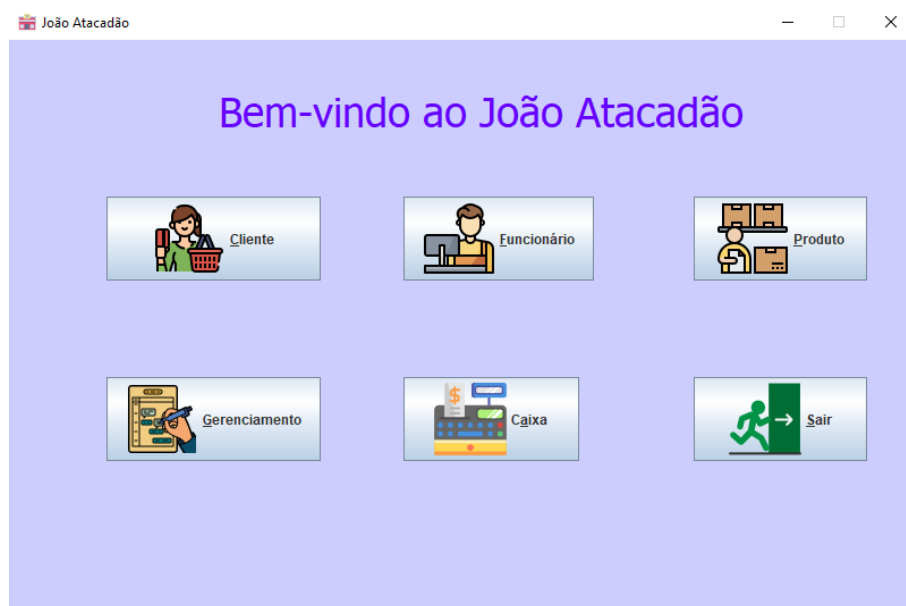
Ambos devem informar seus respectivos CPF's no começo do programa e, em seguida, o sistema irá pesquisar os usuários no banco de dados. Com ambos logados no sistema, o funcionário agora passará a registrar todos os produtos escolhidos pelo cliente pelo código, e o sistema extrairá eles do banco de dados. Todos os produtos estarão em uma classe chamada "ItemProduto", que guarda o produto e a quantidade de produtos, e todos os esses estarão armazenados em um "Carrinho", que contém um ArrayList de "ItemProduto"s. Este carrinho está instanciado no caixa.

A partir daí, o cliente terá algumas opções de ação a se executar. Ele poderá pedir a remoção de um produto do Carrinho, mas precisará do login e da senha de um gerente para conseguir remover. Poderá, também, requisitar um desconto, que será aprovado caso o dia de hoje seja o dia do aniversário do cliente (10%). Obviamente, o cliente também poderá cancelar completamente sua compra.

Finalmente, o cliente pode finalizar seu pedido e pagar a conta. Assim que o pagamento for requisitado, o cliente deverá inserir a senha do seu cartão fidelidade (onde seu saldo se encontra). Se a senha for incorreta, ou não houver saldo no cartão, o pagamento será cancelado e o programa voltará para o estado anterior (com os produtos armazenados no carrinho). Caso haja saldo e a senha esteja correta, o pagamento será efetuado. O saldo do cartão do cliente será diminuído no valor da compra, e seu registro no banco de dados será alterado com o novo valor.

#### **2.3.5. Telas**

Em nosso programa, existem no total 7 telas para interação com o usuário. A primeira delas é a tela do menu principal, que pode ser vista abaixo:



**Figura 2. Menu principal do programa**

Essa tela servirá de conexão principal com todas as demais telas. Como pode ser visto na imagem, essa tela apresenta 6 botões. Os botões de "Cliente", "Funcionário" e "Produto" dão acesso para as telas de Cadastro dos respectivos objetos. O botão de "Gerenciamento" abre a tela de Gerenciamento de Produtos, onde será possível pesquisar um produto, excluir um produto e ver todos os produtos salvos no Banco de Dados. Temos também o botão do "Caixa", que dá acesso à Caixa Registradora, onde será feita toda a operação comercial. Por fim, temos também o botão "Sair", que, uma vez acionado, encerra o programa.

**Figura 3. Tela de cadastro de cliente**

Na tela de Cadastro de Cliente, temos campos textos onde deverão ser inseridos os dados do cliente (Nome, Data de Nascimento e CPF) e além dos dados do Cartão

Fidelidade do cliente (Limite e Senha do Cartão). Nessa tela, temos também dois botões, "Salvar", que salva os dados do Cliente no banco de dados, e o botão "Cancelar" que fecha a tela.

A janela 'Cadastrar Funcionário' possui o seguinte layout:

- Campos de texto para: Nome, Data de Nascimento, CPF, Salário e Senha do Gerente.
- Controles de Gerente: Radio Button 'Sim' (desselecionado) e 'Não' (selecionado), seguido de um botão 'Ok' com uma seta verde.
- Botões de Ação: 'Salvar' (com ícone de disquete) e 'Cancelar' (com ícone de proibido).

**Figura 4. Tela de cadastro de funcionário**

A tela de Cadastro de Funcionário é muito similar com a tela de Cadastro de Cliente. Nela temos campos de texto para inserção dos dados do funcionário (Nome, Data de Nascimento, CPF e Salário), mas ao contrário da tela de Cadastro de Cliente, nessa tela temos um Radio Button para determinar se o funcionário cadastrado é um gerente ou um funcionário normal. Caso seja um gerente, deve-se selecionar o botão "Sim" e clicar em "Ok" e com isso o campo para a inserção da Senha do Gerente será habilitado. Os botões "Salvar" e "Cancelar" tem a mesma funcionalidade vista anteriormente.


Ao selecionar o botão "Produto" no Menu Principal, será aberta a tela de seleção de produtos, como pode ser visto a seguir:

A janela 'Cadastro de Produto' apresenta a seguinte interface:

- Texto orientador: 'Selecione o tipo de produto que será cadastrado:'.
- Sete botões de seleção com ícones e rótulos: Livros, Filmes, Periféricos, Celulares, Computadores, Eletroeletrônicos e Vestuário.

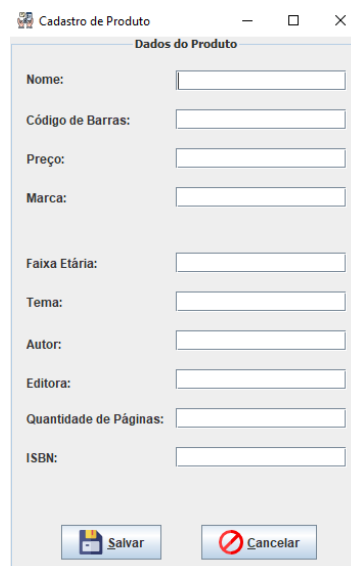
**Figura 5. Tela de seleção de produto**

No total temos 7 botões, cada botão abre uma janela para o Cadastro do Produto do tipo selecionado. Por exemplo, caso o botão "Computadores" seja acionado a seguinte tela surgirá:



**Figura 6. Cadastro de Produtos: Computador**

Assim como as outras telas de cadastro, essa tela também possui campos para inserção dos dados do produto cadastro, além dos botões "Salvar" e "Cancelar" vistos nas telas anteriores. Porém, ao contrário das telas anteriores, essa tela se modifica dependendo do produto selecionado na janela de seleção. No caso da tela abaixo, foi pressionado o botão "Livros" ao invés de "Computadores".



**Figura 7. Cadastro de Produtos: Livro**

Comparando as duas imagens, podemos ver que, com exceção dos 4 primeiros campos, os dados que deverão ser incluídos para o cadastro são diferentes (na primeira tínhamos radio button, enquanto que na segunda esses botões não estão presentes, por exemplo), isso acontece para todos os 7 tipos de produtos que a loja comercializa.



Além das telas de cadastro, temos também a tela de Gerenciamento de Produtos, que pode ser visualizada abaixo:

Gerenciamento de Produtos

### Pesquisa de Produtos

Código:

Departamento

Código	Nome	Preço	Marca
B-005	PC GAMER	50000	Dell

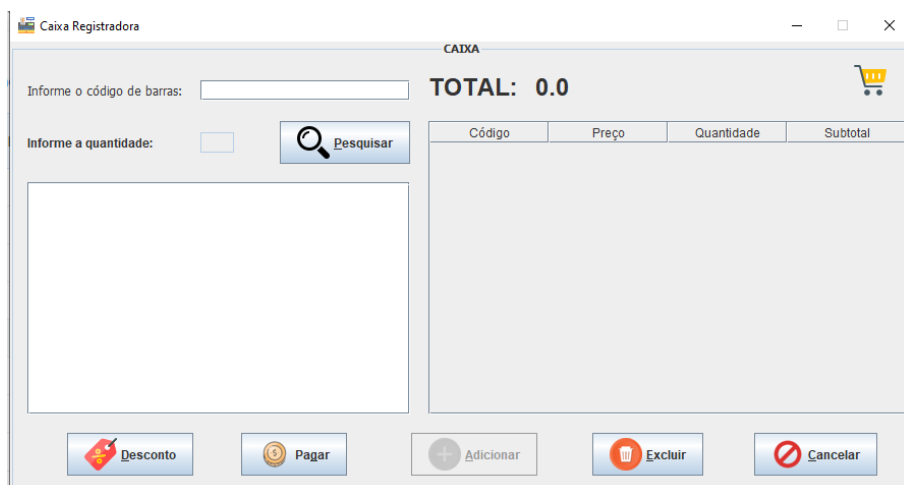
**Figura 8. Gerenciamento de Produtos**

Nessa tela, temos um campo de texto onde deve ser inserido um código de produto. Ao clicar no botão "Pesquisar" buscaremos o código digitado em nosso banco de dados, caso exista um produto com o código informado, ele aparecerá na tabela. Temos também o botão "Listar Todos" esse botão funciona em conjunto com a Combo Box. Na Combo Box, temos as opções de todos os tipos de produtos comercializados pela loja, ao selecionar um departamento (equivalente ao tipo de produto) e clicar em "Listar Todos", todos os produtos desse departamento serão inseridos na tabela.

Contudo, a tabela só mostra 10 produtos de uma vez, por isso temos os botões "Seta Esquerda" e "Seta Direita" que permitem navegar pela tabela. Ao clicar em "Seta Direita" os próximos 10 produtos do mesmo departamento serão colocados na tabela, enquanto que ao clicar em "Seta Esquerda" serão os 10 anteriores que serão inseridos.

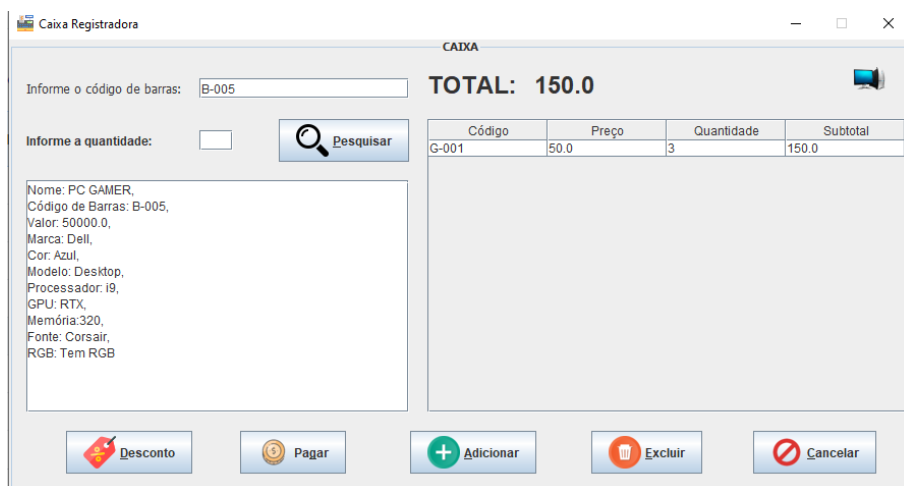
Por último, temos o botão "Excluir" que, ao selecionarmos um produto da tabela, ele irá excluir o dado produto do Banco de Dados, mas para que essa operação aconteça é necessário a autorização do gerente, por isso toda vez que esse botão for acionado será pedido que o gerente insira a sua senha para que então o produto seja de fato excluído do Banco de Dados.

Por fim, temos a tela mais importante do nosso programa, que é a tela do "Caixa". O acesso a essa tela por meio do Menu Principal só acontece se for informado o CPF de um funcionário e um cliente cadastrados anteriormente, para que as Regras de Negócio sejam mantidas. Uma vez que foram inseridos os CPF válidos, a seguinte tela surgirá:



**Figura 9. Caixa Registradora**

Na tela da Caixa Registradora, temos um campo onde será inserido um código de Produto, e ao clicar em "Pesquisar", o produto será buscado no Banco de Dados e, caso seja encontrado, seus dados aparecerão na Área de abaixo. Além disso, o campo de quantidade será habilitado permitindo que o funcionário informe quantas unidades estão sendo compradas. Também é habilitado o botão de "Adicionar" que insere o produto pesquisado no Carrinho do cliente e, com isso, o produto é colocado na tabela e o valor de "Total" é atualizado. A imagem a seguir pode ilustrar esse processo:



**Figura 10. Caixa Registradora pesquisa de um computador**

Um detalhe interessante de se notar é que, quando pesquisado o produto, o ícone do canto superior direito muda de acordo com o tipo de produto, então ele alterou de um Carrinho de Compras para o ícone de um Computador.

Além do botão "Adicionar", temos outros 4 botões nessa tela. O botão "Pagar" abrirá um campo para que o cliente insira a senha do seu Cartão Fidelidade, caso a senha esteja correta e o saldo seja suficiente, valor mostrado em total será debitado do saldo do cartão e com isso finaliza a compra. Temos também o botão "Desconto", o cliente pode

pedir um desconto pelo seu aniversário, ao clicar em "Desconto", o sistema verificará se a data de hoje e a Data de Nascimento do cliente coincidem, se sim, o cliente ganha um desconto de 10% do total a pagar.

Por fim, temos o botão "Excluir" que, selecionado o produto na tabela, excluía aquele produto do Carrinho do cliente, mas para que isso ocorra é necessária a autorização do gerente, por isso é solicitado o CPF do Gerente e a senha dele, para só então o produto ser removido do Carrinho. Há também o botão "Cancelar" que fecha a tela caso pressionado.

### **3. Conclusão**

O paradigma de orientação a objetos traz diversos benefícios e praticidades na hora de implementar um código, facilitando muito a manipulação de recursos gráficos por exemplo.

Para realizar a implementação final desse projeto foi necessário muita pesquisa, o que definitivamente contribuiu para ampliar nosso conhecimento sobre o paradigma de orientação a objetos e da linguagem de programação java.

Já no que se refere a implementação propriamente dita, as maiores dificuldades se mostraram em se manter fiel ao planejamento inicial. Ainda sim, fomos capazes de cumpri-lo em sua integridade, além de adicionar novas ferramentas que inicialmente não estavam presentes como o banco de dados. No geral, a implementação ocorreu sem muitos problemas e, sem sombra de dúvidas, a organização e a divisão de tarefas internamente entre o grupo contribuiu muito para alcançar tal objetivo.

### **Referências**

Ícones: <https://www.flaticon.com/br/>

Dúvidas em geral: <https://stackoverflow.com/>