**UNIVERSIDADE FEDERAL DE JUIZ DE FORA**

ICE - INSTITUTO DE CIÊNCIAS EXATAS DEPT. CIÊNCIA DE COMPUTAÇÃO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO DCC025 · ORIENTAÇÃO A OBJETOS

2° SEMESTRE DE 2019

**SISTEMA PARA LOJA DE ROUPAS EM JAVA**

**(CARIOCA MULTIMARCA)**

**Juiz de Fora 2019**



**UNIVERSIDADE FEDERAL DE JUIZ DE FORA**

BACHARELADO EM SISTEMAS DE INFORMAÇÃO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO DCC025 · ORIENTAÇÃO A OBJETOS

ORIENTADOR:

GLEIPH GHIOTTO LIMA DE MENEZES

COMPONENTES:

|  |  |
| --- | --- |
| NOME | MATRÍCULA |
| BEATRIZ CUNHA RODRIGUES | 201776038 |
| DANIEL RIBEIRO LAVRA | 201735042 |
| IVANYLSON HONÓRIO GONÇALVES | 201776002 |
| JAKSON DUARTE MARTINS JÚNIOR | 201435004 |
| MIKAELA CRISTINY QUITZ | 201676014 |

**Juiz de Fora 2019**

# Resumo

Este projeto tem o principal objetivo controlar e melhorar a organização da loja de roupas Carioca Multimarcas. Um programa básico com o intuito de organizar estoque, venda e clientes para um melhor funcionamento e mais precisão dos processos do ambiente.

Sumário

* **OBJETIVO 5**
* **ARQUIVOS 6**
* **INTERFACES 8**
* **CLASSES 19**
* **ENTIDADE E RELACIONAMENTO 22**
* **DIAGRAMA DE CLASSE 23**
* **LEVANTAMENTO DE REQUISITOS 24**
* **TECNOLOGIAS UTILIZADAS 25**
* **AMBIENTE COMPUTACIONAL 26**

# Objetivo

Carioca Multimarcas é uma loja virtual que tem como principal objetivo a venda de roupas em Juiz de Fora através das redes sociais. Para uma melhor organização, é realizado um programa básico para que haja controle dos processos da loja, incluindo controle de estoque, vendas e clientes. Além disso, como uma das principais vantagens, a informatização da mesma auxilia, principalmente, no controle da quantidade de vendas e estoque que a loja possui.

Para isso, o programa é desenvolvido em Java, onde será utilizada classes, objetos e interfaces para que o usuário (funcionário) tenha acesso a cadastro, alteração, exclusão e busca de dados. Para guardar os dados existentes, será utilizado o JSON. O programa utiliza conceitos fundamentais de Java, aprendidos em sala de aula na disciplina de Orientação a Objetos.

# Arquivos

Para a organização deste projeto, foram criados três pacotes: Registros, Frames e Arquivos.

No pacote Registros é possível encontrar as classes do projeto e seus métodos, contendo informações e dados sobre a mesma. No pacote Frames é possível encontrar as interfaces gráficas do projeto e, por fim, no pacote Arquivos se encontram as classes para utilização do JSON e uma lista contendo os dados das classes presentes no projeto.

O pacote arquivo contém:

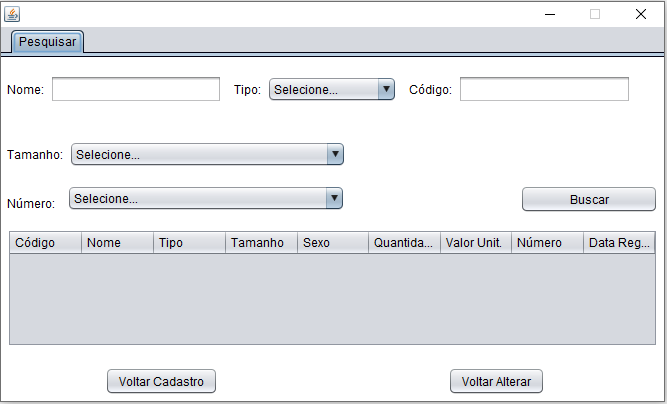
* + ArquivoCliente;
  + ArquivoEstoque;
  + ArquivoLogin;
  + ArquivoVenda.

Para cada classe existe um construtor e os métodos input (criação e cadastro), output (saída do arquivo), alterar e excluir. Especificamente, cada classe tem o seguinte objetivo e atributos:

* ArquivoCliente: A classe ArquivoCliente funciona como controle de clientes e seus endereços no sistema. Contém listas e os métodos para utilização do JSON:
  + Lista de clientes físicos: lista de acesso aos atributos da classe PessoaFísica;
  + Lista de clientes jurídicos: lista de acesso aos atributos da classe PessoaJurídica;
  + Lista de endereço de clientes físicos: lista de acesso aos atributos da classe Endereco para apenas clientes físicos;
  + Lista de endereço de clientes jurídicos: lista de acesso aos atributos da classe Endereco para apenas clientes jurídicos.
* ArquivoEstoque: A classe ArquivoEstoque funciona como controle de produtos em estoque. Contém listas e os métodos para utilização do JSON:
  + Lista de blusas: lista de acesso aos atributos da classe Blusas;
  + Lista de calças: lista de acesso aos atributos da classe Calcas.
* ArquivoLogin: A classe ArquivoLogin funciona como controle de logins no sistema. Contém listas e os métodos para utilização do JSON:
  + Lista de logins: lista de acesso aos atributos da classe Login.
* ArquivoVenda: A classe ArquivoVenda funciona como controle de vendas no sistema. Contém listas e os métodos para utilização do JSON:
  + Lista de venda à vista: lista de acesso aos atributos da classe VendaAVista;
  + Lista de venda a prazo: lista de acesso aos atributos da classe VendaAPrazo.

# Interfaces

# Busca código produto



# Clientes: cadastro, alterar e pesquisar, respectivamente.

# 

# 

# 

# Estoque: cadastro, alterar e pesquisar, respectivamente.

# 

# 

# 

# Logar

# 

# Login: cadastro, alterar e pesquisar, respectivamente.

# 

# 

# 

# Principal:

# 

# Vendas: cadastro, alterar e pesquisar, respectivamente.

# 

# 

# 

# Classes

Para tal objetivo, será utilizada a implementação de classes e métodos. Encontradas no pacote Registro, as classes presentes no projeto são:

* + Pessoa(Abstrata);
  + PessoaFísica (Herança da classe Pessoa);
  + PessoaJurídica (Herança da classe Pessoa);
  + Login;
  + Produto(Abstrata);
  + Blusas;
  + Calças;
  + Endereço;
  + Venda(Abstrata);
  + VendaAPrazo (Herança da classe Venda);
  + VendaAVista (Herança da classe Venda).

Para cada classe existe um construtor e os métodos get’s e set’s para cada atributo. Especificamente, cada classe tem o seguinte objetivo e atributos:

* + Pessoa (Abstrata): A classe pessoa contém os atributos principais para o controle de clientes. Os atributos são:

o Nome: atributo para guardar o nome do cliente;

o Tipo: atributo para guardar o tipo do cliente (Físico ou Jurídico);

o Telefone: atributo para guardar o telefone do cliente;

* + - Data de registro: atributo para guardar a data de entrada do cliente no sistema;

o Endereço (objeto da classe Endereço): atributo da classe endereço para guardar o endereço dos clientes;

* + - Número de compras: atributo para guardar o número de compras que um cliente tem na loja.
  + Pessoa Física: A classe pessoa física herda atributos da classe Pessoa, porém, contém os atributos específicos para o controle de cliente físico. Os atributos são:

o CPF: atributo para guardar o CPF do cliente;

* + - Data de nascimento: atributo para guardar a data de nascimento do cliente;

o Sexo: atributo para guardar o sexo do cliente.

* + Pessoa Jurídica: A classe pessoa jurídica herda atributos da classe Pessoa, porém, contém os atributos específicos para o controle do cliente jurídico. Os atributos são:

o CNPJ: atributo para guardar o CNPJ do cliente;

o IE: atributo para guardar o IE (inscrição estadual) do cliente.

* + Login: A classe login contém os atributos para o acesso do usuário no sistema. Os atributos são:

o Usuário: atributo para guardar o usuário (nome do login) do sistema;

* + - Senha: atributo para guardar a senha de entrada do sistema.
    - Tipo: atributo para guardar o tipo do usuário (Administrador ou Funcionário).
  + Venda (Abstrata): A classe venda contém os atributos para o controle das vendas da loja. Os atributos são:
    - Código: atributo para guardar o código da venda;
    - Quantidade: atributo para guardar a quantidade do produto vendido;
    - Valor: atributo para guardar o valor total da venda;
    - Tipo da Venda: atributo para guardar o tipo da venda realizada (vista ou á prazo);
    - Tipo da Pessoa: atributo para guardar o tipo do cliente que a venda foi realizada (físico ou jurídico);
    - CPF: atributo para guardar o CPF do cliente para qual a venda foi realizada;
    - CNPJ: atributo para guardar o CNPJ do cliente para qual a venda foi realizada;
    - Data: atributo para guardar a data da venda;
    - Código do Produto: atributo para guardar o código do produto vendido;
    - Tipo do Produto: atributo para guardar o tipo do produto vendido;
    - Cliente: atributo da classe ArquivoCliente para acessar a lista de clientes e seus atributos;
    - Estoque: atributo da classe ArquivoEstoque para acessar a lista de produtos e seus atributos.
  + Venda à vista: A classe venda à vista contém o atributo específico para a realização de uma venda à vista. O atributo é:

o Desconto: atributo para guardar o desconto que um cliente pode ter ao realizar a compra à vista.

* + Venda a prazo: A classe venda a prazo contém os atributos específicos para a realização de uma venda a prazo. Os atributos são:
    - Parcelas: atributo para guardar a quantidade de parcelas que foi dividida a compra;
    - Valor da Parcela: atributo para guardar o valor de cada parcela;
  + Endereço: A classe endereço contém os atributos para guardar o endereço dos clientes. Os atributos são:
    - Rua: atributo para guardar o nome da rua;

o Bairro: atributo para guardar o nome do bairro;

* + - Cidade: atributo para guardar o nome da cidade;

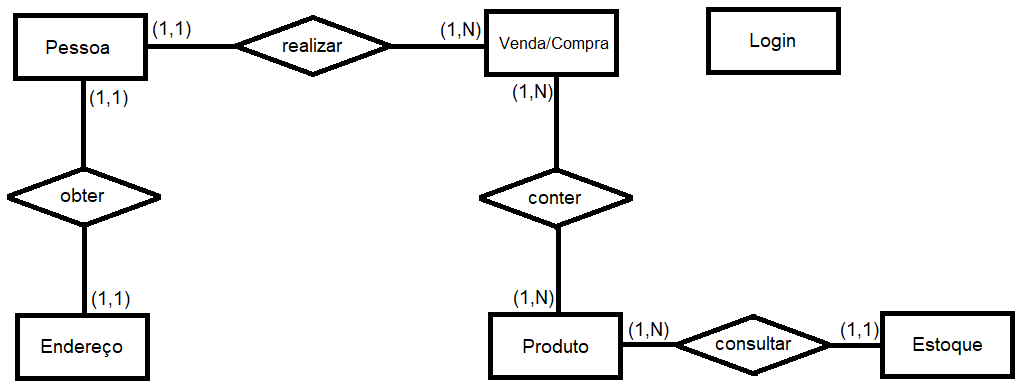
o Estado: atributo para guardar o nome do estado;

o País: atributo para guardar o nome do país;

o Complemento: atributo para guardar o complemento da residência;

* + - CEP: atributo para guardar o CEP da rua;
    - Número: atributo para guardar o número da residência.
  + Produto (Abstrata): A classe endereço contém os atributos para guardar o os produtos registrados na loja. Os atributos são:
    - Código: atributo para identificação única do produto;
    - Quantidade: atributo para guardar a quantidade de um produto em estoque;
    - Tipo: atributo para guardar o tipo do produto (blusas ou calças);
    - Valor: atributo para guardar o valor unitário do produto;
    - Nome: atributo para guardar o nome do produto;
    - Sexo: atributo para guardar para qual sexo o produto é direcionado;
    - Data registro: atributo para guardar a data que o produto foi registrado no sistema.
  + Blusas: A classe blusas contém o atributo específico para a entrada de blusas (camisas/camisetas) no sistema. O atributo é:
    - Tamanho: atributo para guardar o tamanho do produto.
  + Calças: A classe calças contém o atributo específico para a entrada de calças (calças/bermudas/shorts) no sistema. O atributo é:
    - Número: atributo para guardar o número do produto.

# Entidades e Relacionamento



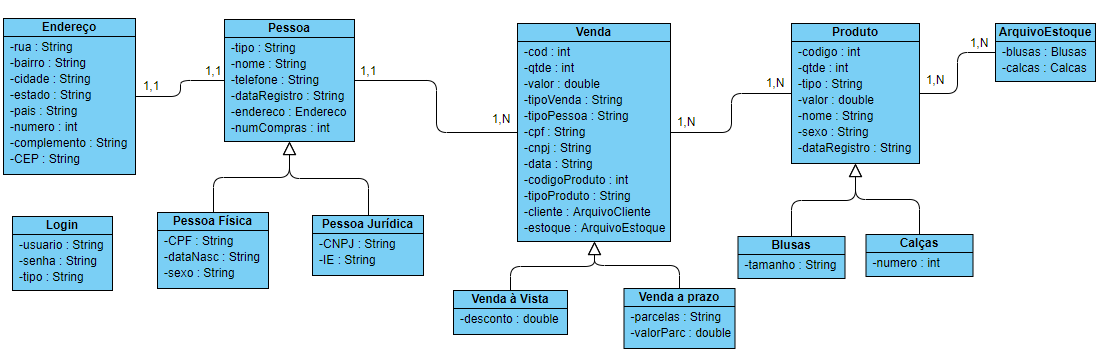
Uma pessoa (física ou jurídica) pode comprar um ou mais produtos, porém, uma venda só pode ser realizada para uma pessoa por vez. Para que seja realizada a venda/compra, é necessário conter o código de um ou mais produtos.

É possível consultar os produtos em estoque tanto no momento da compra/venda, quanto fora. Um ou mais produtos podem ser consultados em um estoque assim como o estoque pode ser consultado por um ou mais produtos.

Uma pessoa pode obter um endereço e um endereço pode ser gravado para uma pessoa.

O login é uma classe a parte onde sua única função é o acesso ao sistema pelo usuário.

# Diagrama de Classe



# Levantamento de requisitos

O sistema tem como objetivo assumir todo o controle dos processos necessários para uma loja virtual, trazendo a inovação através de um projeto completamente informatizado, facilitando e organizando o trabalho.

Com a ideia de montar uma loja informatizada, pesquisamos o funcionamento de uma loja, e desenvolvemos um software focado em controle de clientes e estoque de vendas.

* + Análise de Requisitos:
    - Requisito: controle de estoque.

Atualmente, a loja não possui um controle efetivo do estoque dos produtos presentes e em falta. Com esse sistema, será possível uma melhor organização da entrada e saída de produtos assim como a falta do mesmo.

Portanto, o programa terá um controle de entrada e saída de produtos no sistema, assim, controlando as vendas do mesmo e sua entrada em estoque.

* + - Requisito: controle de clientes.

A loja, atualmente, não tem um controle efetivo dos clientes. Com o sistema, será possível cadastrar/alterar/consultar clientes e até gerar descontos de acordo com o número de compras.

* + - Requisito: controle de vendas.

Para um melhor funcionamento, será implementado um sistema de vendas onde o usuário terá controle de todas as vendas já realizadas, clientes e produtos envolvidos na mesma.

# Tecnologias utilizadas

* + Java: para este projeto, a linguagem de programação utilizada será Java. O Java, atualmente, é uma das linguagens mais utilizadas no mundo. Com a vantagem de ser multiplataforma, tem capacidade de ser utilizado em vários sistemas operacionais. Além disso, através do NetBeans, é possível a criação de interfaces para a utilização do projeto pelo usuário.
  + JSOM: Para guardar os dados em arquivo, utilizamos o JSOM. Com uma fácil sintaxe, o JSOM é perfeito para mudança de dados entre aplicativos, além de ter uma facilidade de desenvolvimento e alto desempenho em suas aplicações.
  + NetBeans IDE: será utilizado o NetBeans como ambiente de desenvolvimento do projeto. É um dos mais utilizados e possui um conjunto de ferramentas que auxiliam a programação de tarefas comuns de implementação.

# Ambiente computacional

# Para este sistema, os seguintes ambientes computacionais foram utilizados:

# Windows 10 - 8GB, processador Intel (R) Core (TM) i5-5200U CPU 2.20GHz.

* + Unix - Distribuição Linux Lubuntu DDR2 2GB, processador Celeron 1.5Ghz 2 cores, 2 threads.

# Windows 10 - 6GB, processador Intel (R) Core (TM) i5-33337U CPU 1.80GHz.