

# **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
IVANYLSON HONÓRIO GONÇALVES	201776002
DANIEL RIBEIRO LAVRA	201735042
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
• CLASSES.....	6
• ENTIDADE E RELACIONAMENTO .....	9
• DIAGRAMA DE CLASSE .....	10
• INTERFACES .....	11
• LEVANTAMENTO DE REQUISITOS .....	19
• TECNOLOGIAS UTILIZADAS .....	20

# 1. 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.

## 2. Classes

Para tal objetivo, será utilizada a implementação de classes e métodos. As classes presentes no projeto são:

- Pessoa;
- Pessoa Física (Herança da classe Pessoa);
- Pessoa Jurídica (Herança da classe Pessoa);
- Login;
- Estoque;
- Endereço;
- Venda;
- Venda a prazo (Herança da classe Venda);
- Venda à vista (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:

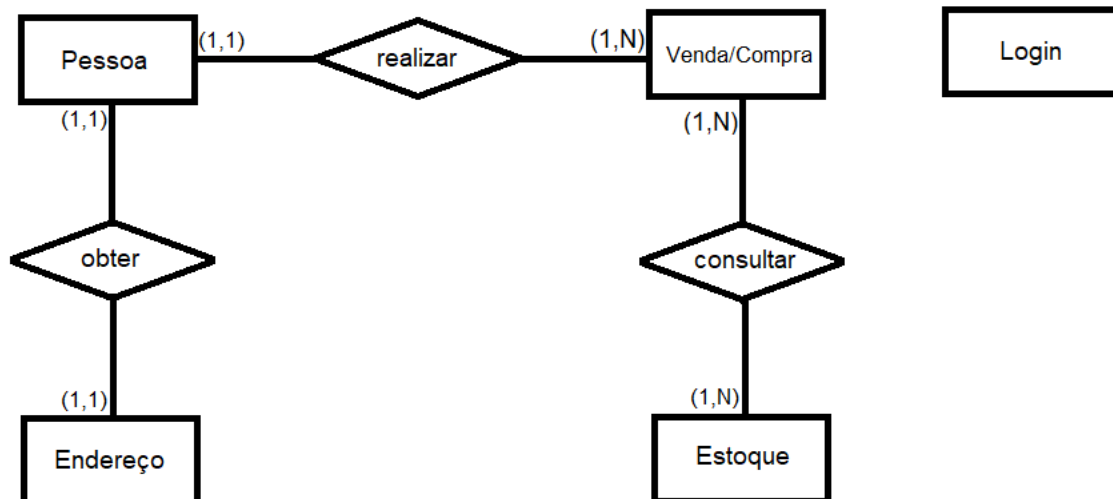
- Pessoa: A classe pessoa contém os atributos principais para o controle de clientes. Os atributos são:
  - Nome: atributo para guardar o nome do cliente;
  - Telefone: atributo para guardar o telefone do cliente;
  - Data de registro: atributo para guardar a data de entrada do cliente no sistema;
  - 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:
  - CPF: atributo para guardar o CPF do cliente;
  - Data de nascimento: atributo para guardar a data de nascimento do cliente;
  - 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:
  - CNPJ: atributo para guardar o CNPJ do cliente;
  - 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:
  - Usuário: atributo para guardar o usuário (nome do login) do sistema;
  - Senha: atributo para guardar a senha de entrada do sistema.
- Venda: A classe venda contém os atributos para o controle das vendas da loja. Os atributos são:
  - Quantidade: atributo para guardar quantas unidades do produto foi vendido;
  - Valor: atributo para guardar o valor de cada produto vendido (quantidade x valor unitário);
  - Pessoa: atributo da classe Pessoa para identificar qual cliente que realizou a compra;
  - Data: atributo para guardar a data da venda;
  - Estoque: atributo da classe Estoque para identificar qual produto foi vendido.
- Venda à vista: A classe venda à vista contém o atributo específico para a realização de uma venda à vista. O atributo é:
  - Desconto: atributo para, dependendo do cliente, calcular a quantidade de 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 podem ser realizadas na venda;
  - Valor: atributo para guardar o valor de cada parcela;
  - Juros: atributo para guardar o valor dos juros da parcela caso seja necessário.
- Endereço: A classe endereço contém os atributos para guardar o endereço dos clientes. Os atributos são:
  - Rua: atributo para guardar o nome da rua;
  - Bairro: atributo para guardar o nome do bairro;
  - Cidade: atributo para guardar o nome da cidade;
  - Estado: atributo para guardar o nome do estado;
  - País: atributo para guardar o nome do país;
  - Número: atributo para guardar o número da residência;

- Complemento: atributo para guardar o complemento da residência;
  - CEP: atributo para guardar o CEP da rua.
- Estoque: A classe estoque contém os atributos para guardar o estoque da loja. Os atributos são:
  - Quantidade: atributo para guardar a quantidade de um produto;
  - Valor: atributo para guardar o valor unitário de um produto;
  - Nome: atributo para guardar o nome do produto;
  - Tamanho: atributo para guardar o tamanho da peça de roupa;
  - Sexo: atributo para guardar o sexo da peça de roupa;
  - Data de registro: atributo para guardar a data de registro de um produto no estoque.



### 3. Entidades e Relacionamento

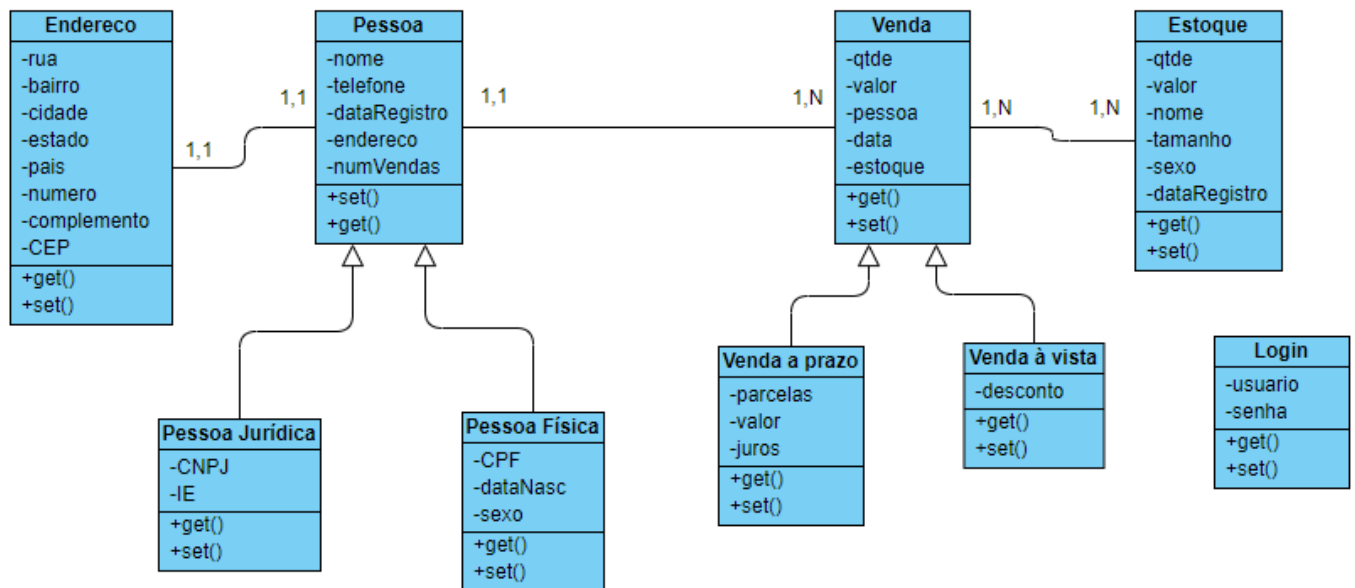


Uma pessoa (física ou jurídica) pode comprar um ou mais produtos, porém, uma venda pode ser realizada para uma pessoa por vez. No momento da compra ou venda, é possível consultar um ou mais produtos em estoque assim como um produto pode ser consultado para uma ou mais vendas.

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.

## 4. Diagrama de Classe



## 5. Interfaces

- PRINCIPAL



- PESSOA (FÍSICA/JURÍDICA)

The screenshot shows a window titled "PESSOA (FÍSICA/JURÍDICA)". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area is divided into two sections. The top section has three tabs: "Cadastro", "Alterar", and "Pesquisar". The "Cadastro" tab is selected. Below the tabs, there is a "Cliente:" label followed by a dropdown menu with the text "Selecione...". Below this, there are several input fields: "Nome:" followed by a text box, "CNPJ:" followed by a text box, "Telefone:" followed by a text box, "IE:" followed by a text box, "CPF:" followed by a text box, "Data Nascimento:" followed by a text box, "Sexo:" followed by a dropdown menu with the text "Selecione...", and "Data Registro:" followed by a text box. At the bottom of the form, there are four buttons: "Cadastrar Endereço", "Limpar", "Enviar", and "Voltar".

— □ ×

Cadastro Alterar Pesquisas

Cliente: Seleccione... ▼

Nome:  CNPJ:

Telefone:  IE:

CPF:  Data Nascimento:

Sexo: Seleccione... ▼ Data Registro:

Alterar Endereço Limpar Alterar Voltar

— □ ×

Cadastro Alterar Pesquisas

Cliente: Seleccione... ▼

CPF:  CNPJ:

Buscar

Voltar

- ENDEREÇOS

A screenshot of a web application window titled 'Cadastro' (Registration). The window has a title bar with standard minimize, maximize, and close buttons. Below the title bar is a tabbed interface with two tabs: 'Cadastro' (selected) and 'Alterar' (Change). The main content area contains several input fields for address information: 'CEP:' (postal code), 'Número:' (number), 'Rua:' (street), 'Estado:' (state, a dropdown menu with 'Selecione...' selected), 'Bairro:' (neighborhood), 'País:' (country), 'Cidade:' (city), and 'Complemento:' (complement). At the bottom of the form are three buttons: 'Enviar' (Send), 'Limpar' (Clear), and 'Voltar' (Back).

A screenshot of the same web application window, but with the 'Alterar' (Change) tab selected. The layout and input fields are identical to the 'Cadastro' tab, including 'CEP:', 'Número:', 'Rua:', 'Estado:' (dropdown), 'Bairro:', 'País:', 'Cidade:', and 'Complemento:'. At the bottom, the buttons are 'Alterar' (Change), 'Limpar' (Clear), and 'Voltar' (Back).

- ESTOQUE

This screenshot shows the 'Cadastro' (Registration) window of a stock management system. The window has a title bar with standard minimize, maximize, and close buttons. Below the title bar is a tabbed interface with three tabs: 'Cadastro' (selected), 'Alterar', and 'Pesquisar'. The main area contains several input fields: 'Nome:' (Name), 'Quantidade:' (Quantity), 'Tamanho:' (Size), 'Valor unitário:' (Unit value), 'Sexo:' (Gender) with a dropdown menu showing 'Selecione...' (Select...), and 'Valor total:' (Total value). There is also a 'Data de Registro:' (Registration date) field. At the bottom, there are three buttons: 'Enviar' (Send), 'Limpar' (Clear), and 'Voltar' (Back).

This screenshot shows the 'Alterar' (Edit) window of the same stock management system. The window has the same title bar and tabbed interface as the first screenshot, but the 'Alterar' tab is now selected. The input fields for 'Nome:', 'Quantidade:', 'Tamanho:', 'Valor unitário:', 'Sexo:', 'Valor total:', and 'Data de Registro:' are present. At the bottom, the buttons are 'Alterar' (Edit), 'Limpar' (Clear), and 'Voltar' (Back).

Nome:

Buscar

Voltar

- **LOGAR**

**CARIOCA MULTIMARCAS**

Usuário:

Senha:

Entrar!

- LOGIN

Usuário:

Senha:

Confirmação de Senha:

Usuário:

Senha:

Confirmação de Senha:



Usuário:

Buscar

Voltar

- **VENDAS**

Tipo de Pagamento:

Produto:

Cliente:

Quantidade:

CPF:

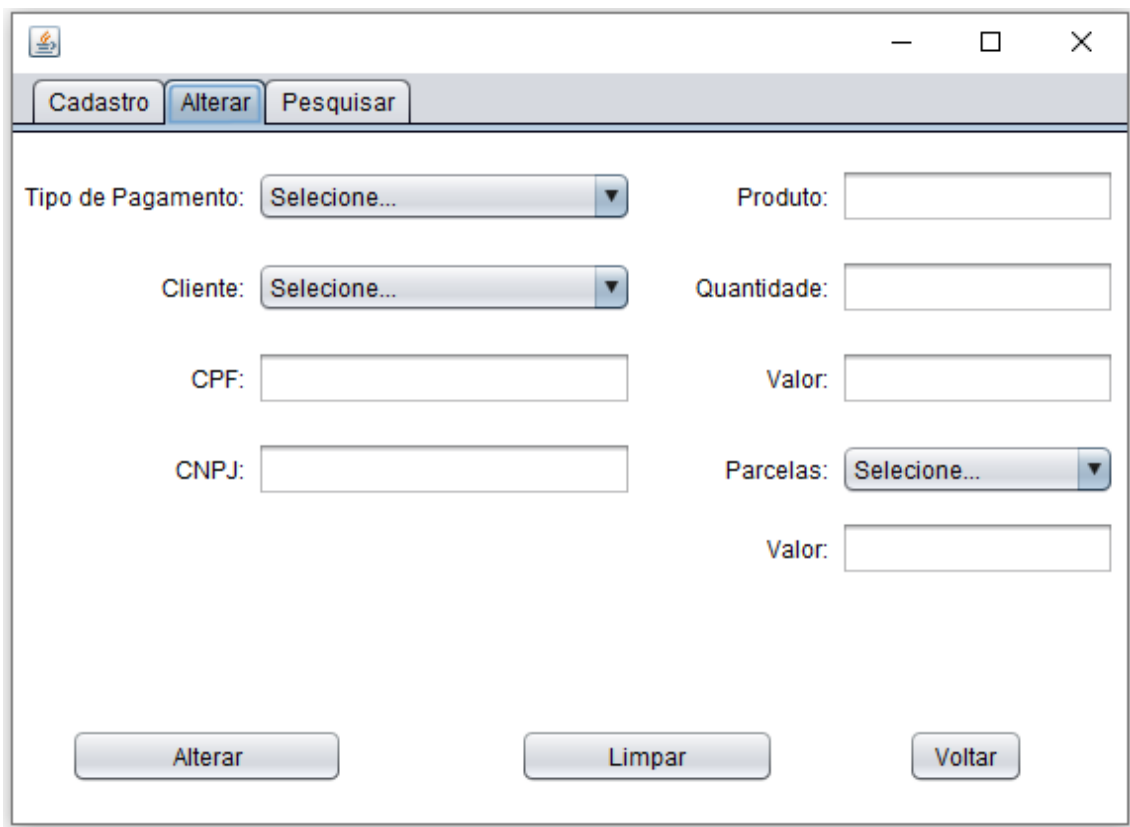
Valor:

CNPJ:

Parcelas:

Valor:

Enviar Limpar Voltar



Tipo de Pagamento:

Produto:

Cliente:

Quantidade:

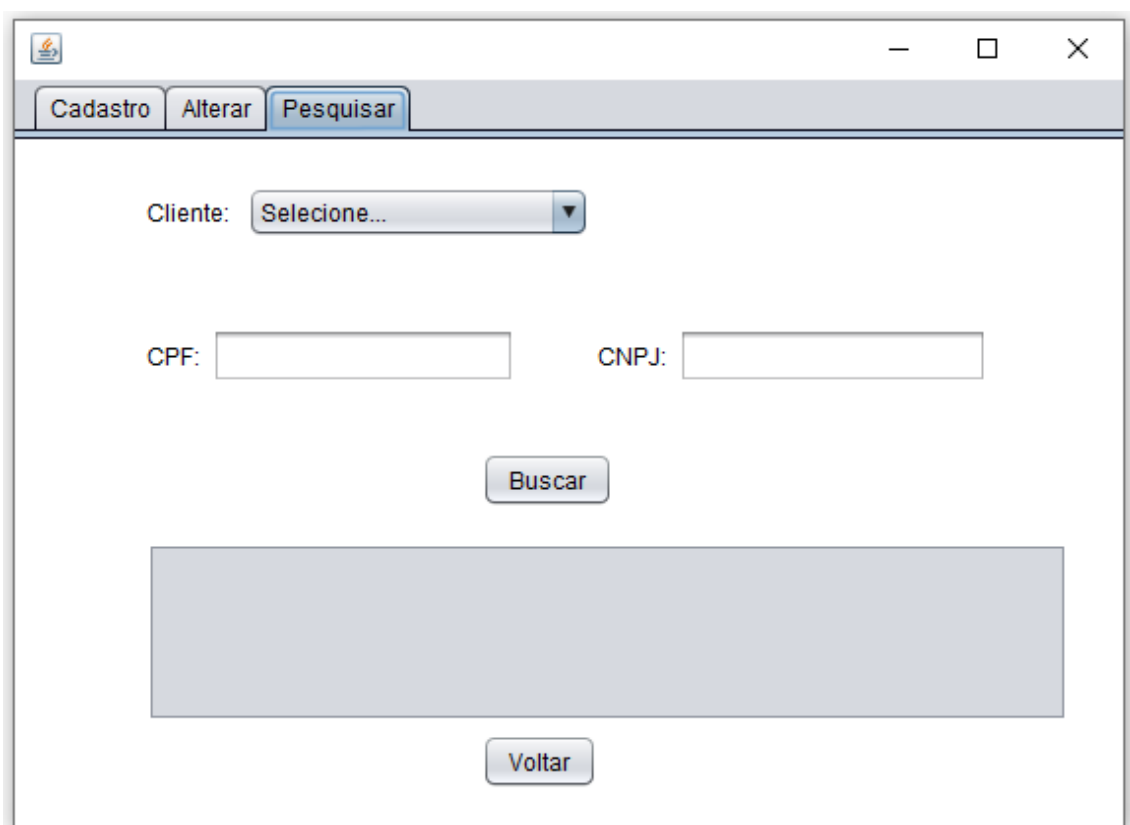
CPF:

Valor:

CNPJ:

Parcelas:

Valor:



Cliente:

CPF:

CNPJ:

## 6. 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.

## 7. 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.