

INF
INSTITUTO DE
INFORMÁTICA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

Projeto Final

Preparado para: **Acadêmicos da Disciplina POO – Turma A**

Elaborado por: **Prof. Dr. Sávio Salvarino Teles de Oliveira**, com apoio dos monitores **Willgnner Ferreira Santos e Vinícius Aguiar**

Semestre: **2024/1**

Adaptado do **Prof. Me. Olegário Corrêa da Silva Neto**

RESUMO EXECUTIVO

Objetivo

Demonstrar a compreensão dos conceitos de programação orientada a objetos (POO) através da implementação de um projeto em Python ou Java.

Trabalho em grupo

Você deve formar uma equipe de até 03 acadêmicos para o projeto. Elaborar uma matriz de responsabilidades, atribuindo responsabilidades individuais a cada membro da equipe. O produto final deve ser apresentado em sala de aula.

Solução

O *software* deve ser desenvolvido com o *back-end* na linguagem Java ou Python. A persistência dos dados será feita em arquivos de texto no formato CSV. O repositório do *front-end* estará disponível para que o aluno possa integrar o *back-end* e realizar melhorias na interface de acordo com suas preferências.

Estrutura do Projeto

O projeto está organizado em módulos, que funcionarão como etapas do mesmo. Cada módulo possui um conjunto de artefatos que descrevem os requisitos e estrutura do sistema de *software* a ser codificado.

Projeto Final

Você pode escolher entre desenvolver o sistema definido neste documento de Sistema de Controle de Pedidos ou propor um projeto próprio, caso apresente os conceitos de orientação a objetos. Independente do projeto escolhido, a implementação deve atender aos seguintes requisitos:

- **Classes e Objetos:** definir classes para representar as entidades do projeto, com atributos e métodos apropriados.
- **Encapsulamento:** utilizar modificadores de acesso para controlar a visibilidade dos atributos e métodos das classes.
- **Herança:** criar uma hierarquia de classes utilizando herança, se aplicável ao projeto.

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

- **Polimorfismo:** implementar métodos polimórficos para permitir diferentes comportamentos para objetos de diferentes classes.
- **Exceções:** utilizar tratamento de exceções para lidar com erros e situações inesperadas.
- **Testes:** implementar testes unitários para garantir o funcionamento correto do código.

Entregáveis

- **Código-fonte:** O código-fonte completo do projeto em Python ou Java.
- **Documentação:** Um documento descrevendo o projeto, as classes e os métodos implementados.
- **Apresentação:** Uma apresentação (10-20 minutos) para demonstrar o funcionamento do projeto.

Critérios de Avaliação

- **Correção:** O código funciona corretamente e atende aos requisitos do projeto.
- **Design:** O projeto utiliza os conceitos de POO de forma eficiente e elegante.
- **Documentação:** A documentação é clara, concisa e completa.
- **Apresentação:** A apresentação demonstra o funcionamento do projeto de forma clara e detalhada.

Dicas

- Planeje o design do seu projeto antes de começar a codificar.
- Utilize boas práticas de programação, como nomes de variáveis e métodos significativos, indentação consistente e comentários claros.
- Teste seu código regularmente para identificar e corrigir erros.
- Consulte a documentação e recursos online para Python ou Java se necessário.

Cronograma

A data de entrega estipulada para esse trabalho será no dia 04/07/2024 e está sujeita a ajuste conforme a discricionariedade do professor responsável pela disciplina. Caso haja necessidade de alteração na data de entrega previamente acordada, será

solicitada uma revisão considerando as circunstâncias pertinentes, sempre visando assegurar o melhor resultado possível para todos os envolvidos.

SISTEMA DE CONTROLE DE PEDIDOS - SCP

Caso de Uso

O diagrama a seguir apresenta as principais funcionalidades do SCP e delimita seu escopo.

Descrição de Cenários

Cadastrar Cliente - **Cadastrar** novo cliente no sistema.

Realizar Pedido - **Escolher** itens para compra, **Confirmar** os itens do pedido e **Listar** histórico de pedidos efetuados.

Modelo Conceitual

O modelo conceitual apresenta as principais classes que compõem o SCP.

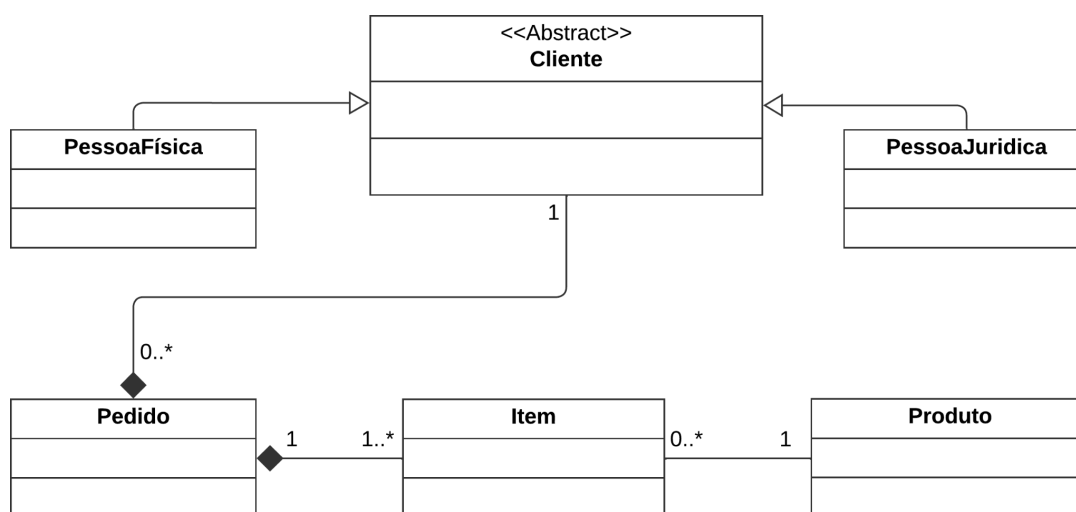
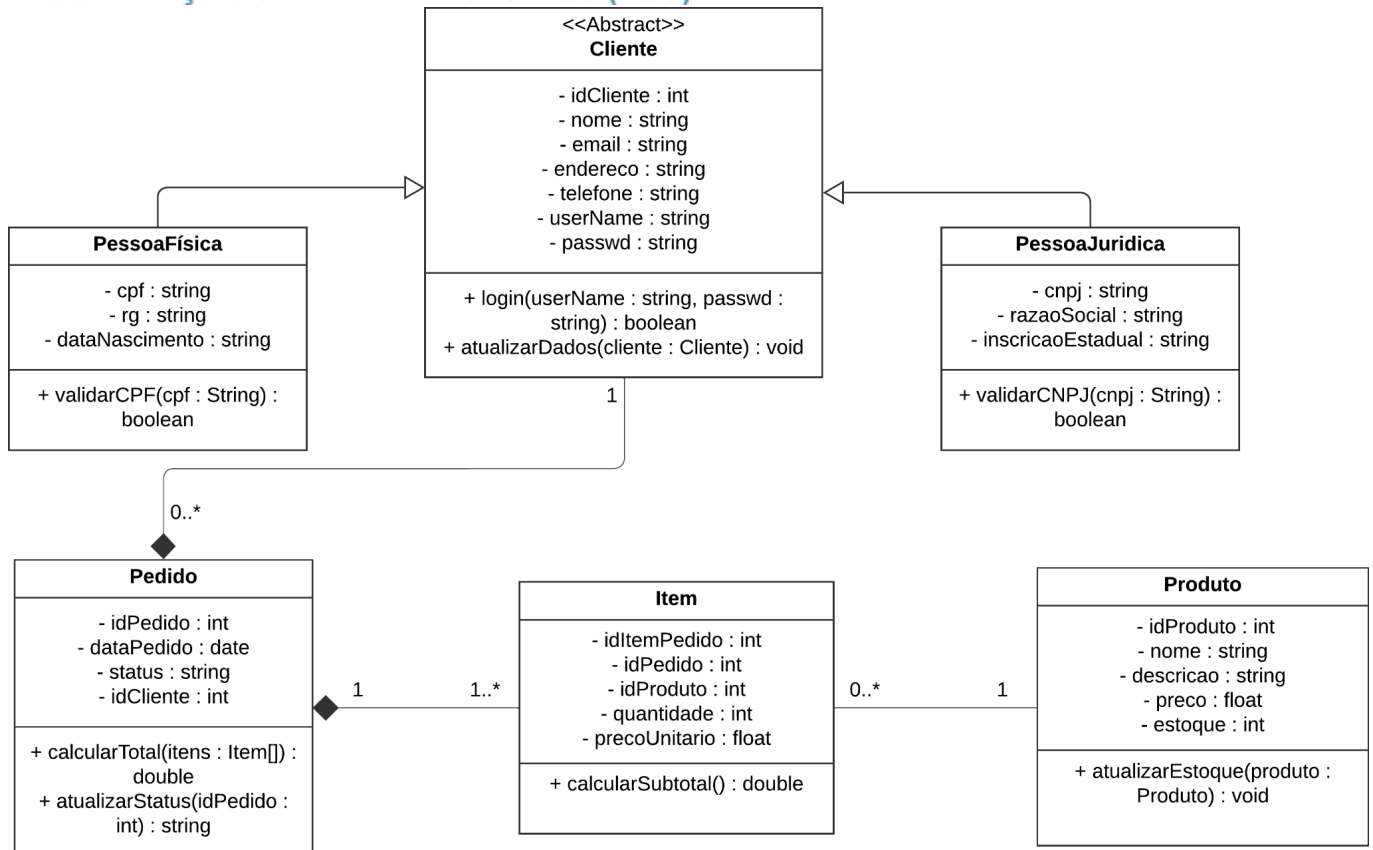


Diagrama de Classe

O diagrama a seguir apresenta os detalhes das principais classes a serem implementadas.

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)



Descrição de Requisitos de Implementação

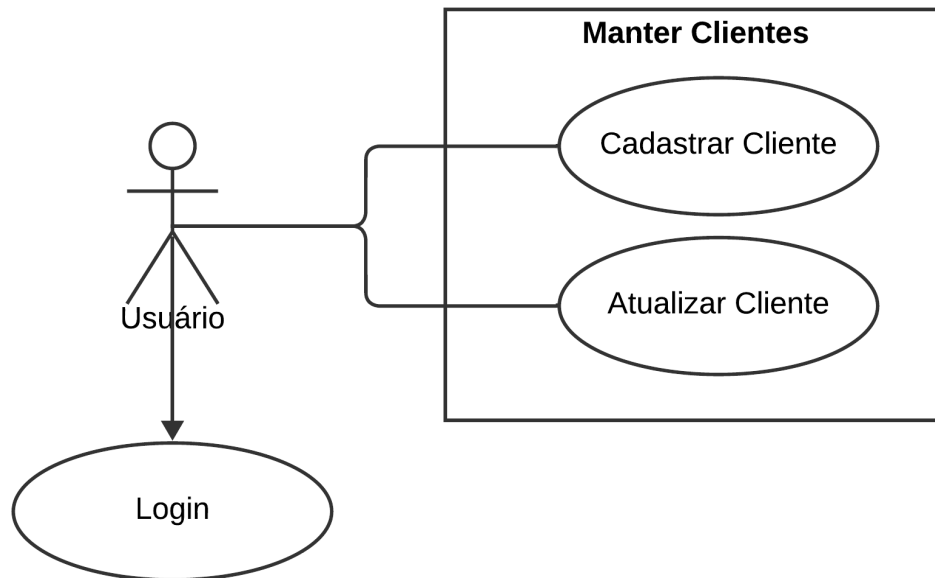
Os códigos referentes para estes requisitos funcionais devem contemplar as seguintes condições de implementação:

- Utilizar a linguagem Java ou Python para o *back-end*.
- **Persistência:**
 - Ao iniciar o sistema, ele carregará todos os dados contidos nos respectivos arquivos de dados (Produtos.csv e Pedidos-Realizados.csv) em arquivo.
 - Ao finalizar o sistema, ele deverá sobrescrever os dados atuais pelos novos, caso necessário seja feita alguma alteração, por exemplo, no registo de compra do produto.

MÓDULO CLIENTE

Caso de Uso

Manter Cliente: módulo especializado para manipulação e manutenção de todos os dados relativos ao cliente que irá realizar a compra. Este cliente pode ser classificado como Pessoa Jurídica ou Pessoa Física.



Descrição de Cenários

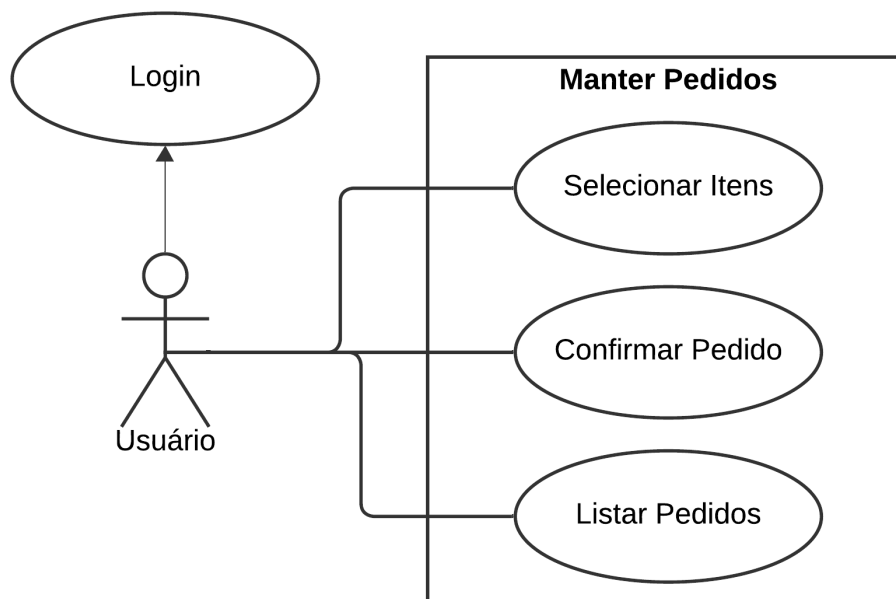
1. O cliente acessa a página de cadastro da plataforma e-commerce e o sistema exibe um formulário de cadastro solicitando informações do cliente, como:
 - a. Nome completo;
 - b. Endereço de e-mail;
 - c. Senha;
 - d. Data de nascimento (opcional);
 - e. CPF/CNPJ (opcional);
 - f. Endereço de entrega;
 - g. Telefone para contato.
2. O cliente preenche o formulário com suas informações.
3. O sistema valida as informações preenchidas:
 - a. Verifica se o e-mail já está cadastrado.
 - b. Verifica se os demais campos obrigatórios estão preenchidos.

4. Se houver algum erro de validação, o sistema exibe uma mensagem de erro e solicita ao cliente que corrija as informações.
5. Se as informações forem válidas, o sistema cria um novo cadastro para o cliente.

MÓDULO PEDIDO

Caso de Uso

Este módulo é especializado na manipulação e manutenção dos dados referentes aos pedidos realizados pelos clientes da empresa.



Descrição de Cenários

1. Selecionar itens do pedido

- a. O cliente navega pelo catálogo de produtos.
- b. O cliente adiciona os itens desejados ao carrinho de compras.

2. Confirmar pedido

- a. O cliente acessa o carrinho de compras e verifica os itens adicionados.
- b. O cliente confirma o pedido.

3. Listar histórico de pedidos

- a. O cliente acessa a área de histórico de pedidos para visualizar os pedidos realizados, com detalhes como data, itens, valor total e status.