



Projeto Final

Preparado para: **Acadêmicos da Disciplina CMP2105, Turmas A01, A05**

Preparado por: **Olegário Correa da Silva Neto**

26 de novembro de 2023

2023, 2º Semestre, 2º Bimestre

RESUMO EXECUTIVO

Objetivo

Desenvolver um sistema de software para controle de pedidos registrados por um usuário para clientes de uma empresa.

Metas

Compreender os requisitos especificados para o sistema de controle de pedidos. Definir uma equipe de até três acadêmicos para execução do projeto. Elaborar uma matriz de responsabilidades, designando a responsabilidade individual de cada membro da equipe. Codificar os módulos Manter Cliente, Manter Produtos e Manter Pedidos. Testar os módulos desenvolvidos. Apresentar o software ao professor.

Solução

O software deve ser codificado na linguagem de programação C. A persistência dos dados mantidos pelo sistema será realizada em arquivos texto no formato CSV e arquivos binários para dados estritamente numéricos. O sistema de software será desenvolvido, prioritariamente, para funcionamento na plataforma Linux.

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.

Artefatos:

- Diagrama de caso de uso
- Descrição de cenário de uso
- Diagrama de classe
- Descrição requisitos de implementação

Critérios de Avaliação

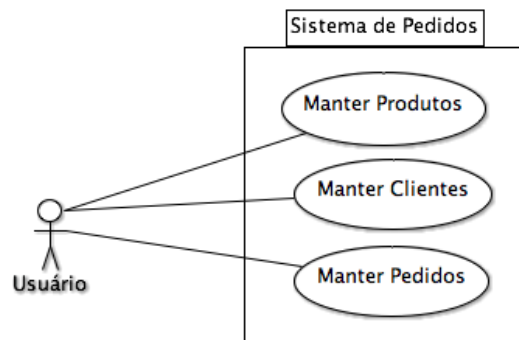
O resultado da avaliação deste projeto corresponderá à 7,0 pontos da média referente ao 2º bimestre*. Esta avaliação será realizada individualmente e ponderada de acordo com a matriz de responsabilidades definidas pela equipe.

* Média do 2º Bimestre = (7,0 pontos + 2,0 pontos + 1,0 ponto). 7,0 pontos do projeto final. 2,0 pontos de atividades de fixação. 1,0 ponto de Avaliação Interdisciplinar.

SISTEMA DE CONTROLE DE PEDIDOS - SCP

Caso de Uso

O diagrama abaixo apresenta as principais funcionalidades do Sistema de Controle de Pedidos e delimita seu escopo.



Descrição de Cenários

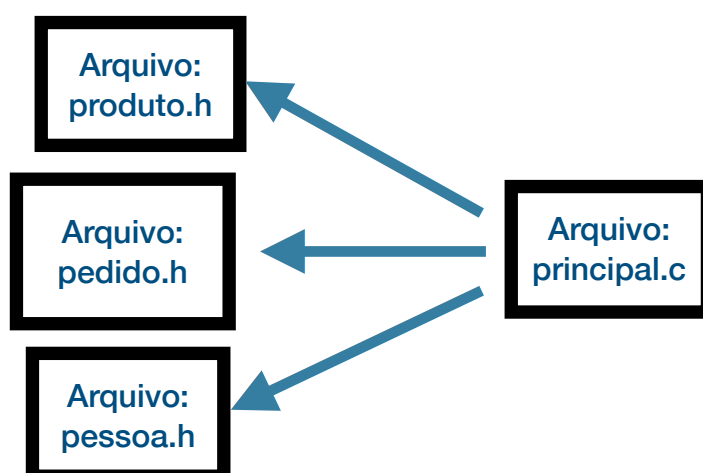
Manter Produtos - Cadastrar novo produto, Remover produto cadastrado, Consultar produto cadastrado e Listar produtos cadastrados.

Manter Clientes - Cadastrar novo cliente, Remover cliente cadastrado, Consultar cliente cadastrado e Listar clientes cadastrados.

Manter Pedidos - Cadastrar novo pedido de produto(s) cadastrado(s) para cliente cadastrado, Remover pedido cadastrado e seus respectivos itens, Consultar pedido cadastrado e Listar pedidos cadastrados.

Estrutura do Sistema

A estrutura do sistema apresenta as principais bibliotecas que compõem o software (SCP).



TÉCNICAS DE PROGRAMAÇÃO II

Descrição das Bibliotecas

Arquivo cliente.h

Dados do Cliente (cpf, nome, celular, telefone, email)

Funções:

- **Todas as funções necessárias para manipular os dados da struct, o arquivo clientes.csv e apresentar os respectivos resultados.**

Arquivo produto.h

Dados do Produto(codigo, descricao, valor)

Funções:

- **Todas as funções necessárias para manipular os dados da struct, o arquivo produtos.bin e apresentar os respectivos resultados.**

Arquivo pedido.h

Dados do Pedido(codigo, data, codigoProduto, qtdeProduto, valorTotal)

Funções:

- **Todas as funções necessárias para manipular os dados da struct, o arquivo pedidos.bin e apresentar os respectivos resultados.**

Arquivo principal.c

Funções:

- **short menuPrincipal()** // imprimir o menu principal: Clientes, Produtos, Pedidos, Sair
- **int main()** // Responsável por “orquestrar” as funções de menu e demais. apresenta os detalhes das principais classes a serem implementadas.

Descrição de Requisitos de Implementação

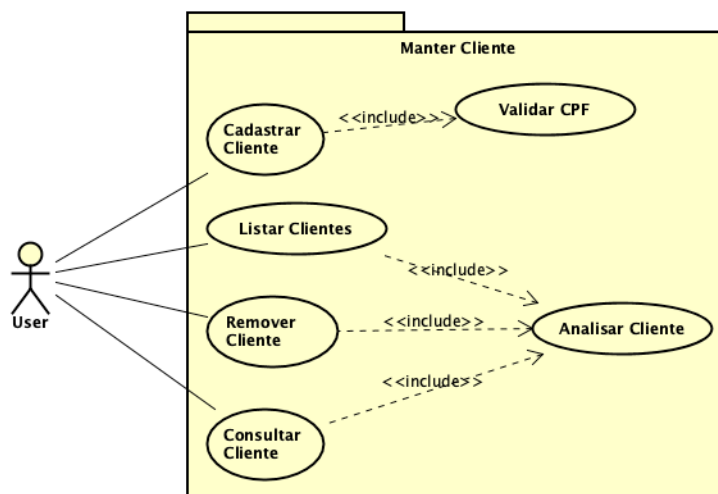
O código referente a estes requisitos funcionais devem contemplar os seguintes requisitos de implementação:

- Utilizar a linguagem C;
- Persistência:
 - + Os dados serão persistidos em seus respectivos arquivos de dados (clientes.csv, produtos.bin e pedidos.bin).
 - + O arquivo com extensão CSV será em formato texto, enquanto os arquivos com a extensão BIN serão em formato binário.
 - + Todos os dados numéricos deverão ser armazenados com tipos numéricos.

MÓDULO CLIENTE

Caso de Uso

Manter Cliente: Módulo especializado para manipulação e manutenção de todos os dados relativos ao cliente da empresa.



Descrição de Cenários

Cadastrar Cliente - Descrição do fluxo principal/alternativo

1. O usuário informa ao sistema o código identificador do cliente (CPF).
2. O sistema valida o número de cadastro do cliente.
 - I. Valida CPF - verifica se o número do CPF é válido (de acordo com o algoritmo definido pela Receita Federal);
3. O sistema analisa cliente (verifica se o CPF já encontra-se cadastrado no sistema). Caso já esteja cadastrado o sistema retorna uma mensagem de erro informando que o respectivo número de cadastro já encontra-se cadastrado e retorna o cursor para tela de cadastro posicionando o cursor sobre o campo do número de cadastro (CPF).
4. O usuário informa o nome completo, celular, telefone e email.
5. O usuário efetiva o cadastro após digitar o e-mail.
6. O sistema emite uma mensagem "Cliente Cadastrado"

Consultar Cliente - Descrição do fluxo principal/alternativo

1. O usuário informa o CPF do cliente, a ser consultado no sistema.
2. O sistema analisa o cliente, verifica se o respectivo CPF existe. Caso exista o sistema mostra os dados do respectivo cliente na tela. Caso não exista, o sistema retorna uma mensagem de erro informando que o cliente não existe.

Remover Cliente - Descrição do fluxo principal/alternativo

1. O usuário informa o CPF do cliente, a ser removido do sistema.

TÉCNICAS DE PROGRAMAÇÃO II

2. O sistema analisa o cliente, verifica se o respectivo CPF existe. Caso exista o sistema pede uma confirmação de exclusão do cliente ao usuário. Caso ele confirme o cliente é retirado do cadastro, caso não confirme o cliente é mantido. Caso não exista, o sistema retorna uma mensagem de erro informando que o cliente não existe.

Listar Cliente - Descrição do fluxo principal/alternativo

1. O usuário solicita ao sistema uma listagem de todos os clientes cadastrados.
2. O sistema mostra na tela a listagem com todos os dados dos clientes cadastrados.

Biblioteca Cliente

O software deve ser codificado na linguagem de programação C.

Arquivo cliente.h

Struct Cliente(cpf, nome, celular, telefone, email)

Funções:

- **short menuCliente()** // Incluir, Excluir e Consultar Cliente, Voltar ao Menu Principal.
- **short eValido(char* cpf)** // Verifica se CPF é válido
- **Cliente* buscarCliente(char* cpf)** //Abre o arquivo cliente.csv, busca o cpf informado e retorna o ponteiro para o referido struct.
- **short gravarCliente(Cliente* novo)** //Grava no arquivo cliente.csv, os dados que compoem o struct Cliente.
- **void imprimirCliente(Cliente* cliente)** //Apresenta na tela os dados do Cliente.
- **void listarClientes()** //Lista todos os clientes cadastrados

Descrição de Requisitos de Implementação

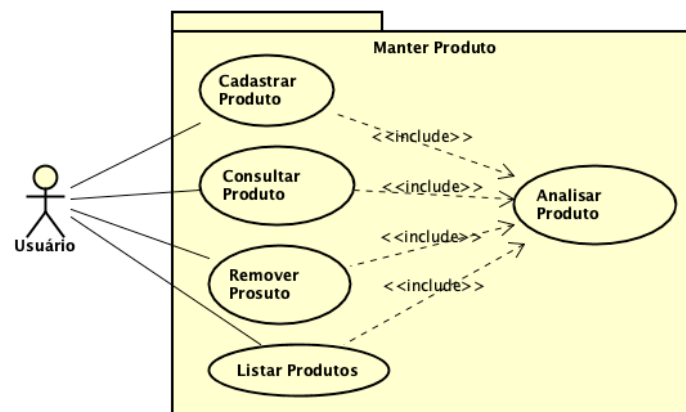
O código referente a estes requisitos funcionais devem contemplar os seguintes requisitos de implementação:

- Todas as operações cadastrais (cadastrar, consultar, remover e listar) serão realizadas sobre seus respectivo arquivo (cliente.csv).
- A struct Cliente deve contemplar todos os seguintes dados: CPF, nome, celular, telefone e email.
- O arquivo deve atender ao padrão de arquivo CSV, onde os dados são separados por ponto vírgula.

MÓDULO PRODUTO

Caso de Uso

Manter Produtos: Módulo especializado para manipulação e manutenção de todos os dados relativos aos produtos comercializados pela empresa.



Descrição de Cenários

Cadastrar Produto - Descrição do fluxo principal/alternativo

1. O usuário informa ao sistema o código identificador do produto
2. (Analisa Produto) O sistema verifica se o respectivo código já existe. Caso exista o sistema retorna uma mensagem de erro informando que o código já existe e retorna para tela de cadastro com o cursor no campo de código para ser preenchido novamente.
3. O usuário informa a descrição e preço do respectivo produto.
4. O usuário efetiva o cadastro após digitar o preço do produto.

Consultar Produto - Descrição do fluxo principal/alternativo

1. O usuário informa o código o produto, a ser consultado no sistema.
2. O sistema analisa o produto, verifica se o respectivo número de cadastro existe. Caso exista o sistema mostra os dados do respectivo produto na tela. Caso não exista, o sistema retorna uma mensagem de erro informando que o produto não existe.

Remover Produto - Descrição do fluxo principal/alternativo

1. O usuário informa o código do produto, a ser removido do sistema.
2. O sistema analisa o produto, verifica se o respectivo número de produto existe. Caso exista, o sistema verifica se existe algum pedido que contenha este produto. Caso exista, o sistema informa ao cliente que tal produto não poderá ser excluído por constar em pedidos de clientes. Caso não exista pedidos com o respectivo produto, o sistema pede uma confirmação de exclusão do produto ao usuário. Caso ele confirme o produto é retirado do cadastro, caso não confirme o produto é mantido. Caso não exista o produto com o respectivo código, o sistema retorna uma mensagem de erro informando que o produto não existe.

TÉCNICAS DE PROGRAMAÇÃO II

Listar Produto - Descrição do fluxo principal/alternativo

1. O usuário solicita ao sistema uma listagem de todos os produtos cadastrados.
2. O sistema mostra na tela a listagem com todos os dados dos produtos cadastrados.

Biblioteca Produto

O software deve ser codificado na linguagem de programação C.

Arquivo produto.h

Struct Produto(codigo, descricao, preço)

Funções:

- **short menuProduto()** // Incluir, Excluir e Consultar Produto, Voltar ao Menu Principal.
- **Produto* buscarProduto(char* codigo)** //Abre o arquivo produto.bin, busca o código informado e retorna o ponteiro para o referido struct.
- **short gravarProduto(Produto* novo)** //Grava no arquivo produto.bin, os dados que compoem o struct Produto.
- **void imprimirProduto(Produto* produto)** //Apresenta na tela os dados do Produto.
- **void listarProdutos()** //Lista todos os produtos cadastrados

Descrição de Requisitos de Implementação

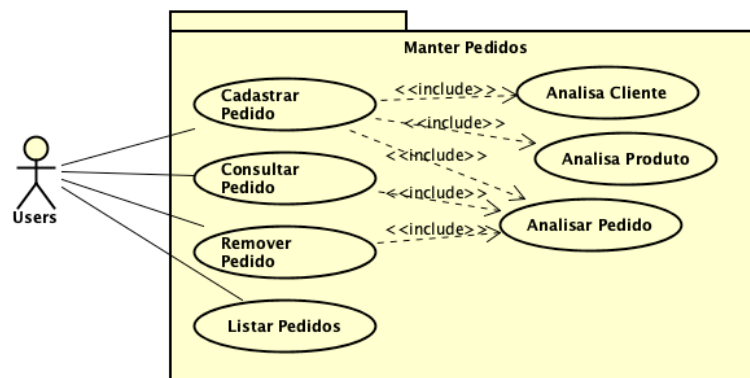
O código referente a estes requisitos funcionais devem contemplar os seguintes requisitos de implementação:

- Todas as operações cadastrais (cadastrar, consultar, remover e listar) serão realizadas sobre o arquivo produto.bin.
- A struct Produto deve contemplar todos os seguintes dados: código, descrição, preço.
- O arquivo produto.bin deve ser em formato binário.

MÓDULO PEDIDO

Caso de Uso

Manter Pedidos: Módulo especializado para manipulação e manutenção de todos os dados relativos ao Pedidos de Produtos da empresa.



Descrição de Cenários

Cadastrar Pedido - Descrição do fluxo principal/alternativo

1. O usuário informa ao sistema o código identificador do pedido
2. (Analisa Pedido) O sistema verifica se o respectivo código já existe. Caso exista o sistema retorna uma mensagem de erro informando que o código já existe e retorna para tela de cadastro com o cursor no campo de código para ser preenchido novamente.
3. O usuário informa o código do cliente do referido pedido.
4. (Analisa Cliente) O sistema verifica se o respectivo cliente já existe. Caso não exista, o sistema retorna uma mensagem de erro informando que o cliente não existe e retorna para tela de cadastro com o cursor no campo de código do cliente para ser preenchido novamente.
5. O usuário informa o código do produto associado ao pedido.
6. (Analisa Produto) O sistema verifica se o produto referenciado pelo código existe na lista de produtos. Caso não exista, o sistema retorna uma mensagem de erro informando que o produto não existe e retorna para tela de cadastro com o cursor no campo de código do produto do respectivo pedido para ser preenchido novamente.
7. O sistema deve consultar o preço do produto e mostrá-lo.
8. O usuário informa a quantidade do respectivo produto.
9. O sistema deve calcular o valor total do pedido e preenche-lo.
10. O usuário efetiva o cadastro do produto do respectivo pedido após digitar a quantidade.
11. O sistema define preenche automaticamente a data do sistema, calcula o valor total do pedido e preenche-o automaticamente.
12. O sistema pergunta se o usuário deseja incluir novo pedido. Caso deseje, ele volta ao item 1 do respectivo requisito.

TÉCNICAS DE PROGRAMAÇÃO II

13. O sistema inclui o respectivo pedido no arquivo pedidos.bin.

Consultar Pedido - Descrição do fluxo principal/alternativo

1. O usuário informa o código do pedido, a ser consultado no sistema.
2. O sistema analisa o pedido, verifica se o respectivo código de cadastro existe. Caso exista o sistema mostra os dados do respectivo pedido na tela. Caso não exista, o sistema retorna uma mensagem de erro informando que o pedido não existe.

Remover Pedido - Descrição do fluxo principal/alternativo

1. O usuário informa o código do pedido a ser removido do sistema.
2. O sistema analisa o pedido, verifica se o respectivo código de pedido existe. Caso exista o sistema pede uma confirmação de exclusão do pedido ao usuário. Caso ele confirme o pedido é retirado do arquivo pedido.bin, caso não confirme o pedido é mantido. Caso não exista um pedido com o respectivo código, o sistema retorna uma mensagem de erro informando que o pedido não existe.

Listar Pedido - Descrição do fluxo principal/alternativo

1. O usuário solicita ao sistema uma listagem de todos os pedidos cadastrados.
2. O sistema mostra na tela a listagem com todos os pedidos cadastrados com seus respectivos itens.

Biblioteca Pedido

O software deve ser codificado na linguagem de programação C.

Arquivo pedido.h

Struct Pedido(codigo, data, codigoProduto, qtdeProduto, valorTotal)

Funções:

- **short menuPedido()** // Incluir, Excluir e Consultar Pedido, Voltar ao Menu Principal.
- **Produto* buscarPedido(char* codigo)** //Abre o arquivo pedido.bin, busca o código informado e retorna o ponteiro para o referido struct.
- **short gravarPedido(Pedido* novo)** //Grava no arquivo pedido.bin, os dados que compoem o struct Pedido.
- **void imprimirPedido(Pedido* pedido)** //Apresenta na tela os dados do Pedido
- **void listarPedidos()** //Lista todos os pedidos cadastrados

Descrição de Requisitos de Implementação

O código referente a estes requisitos funcionais devem contemplar os seguintes requisitos de implementação:

- Todas as operações cadastrais (cadastrar, consultar, remover e listar) serão realizadas sobre o arquivo pedidos.bin.
- O arquivo pedidos.bin deve estar em formato binário.
- A struct Cliente deve contemplar os seguintes dados: codigo, data, produto, quantidade, valor.

CRONOGRAMA

Descrição do Cronograma de Implementação

Este cronograma deve ser seguido pela equipe definida no início do projeto.

Relação de Atividades do Projeto de Desenvolvimento do Sistema de Controle de Pedidos				
Item	Pontuação	Módulo	Atividade	Data Entrega
1	1,00	Cliente	Definir struct Cliente	30/11/2023
2	1,00		Implementar menuCliente()	30/11/2023
3	1,00		Implementar a função de validação de CPF eValido()	04/12/2023
4	1,00		Implementar o caso de uso Cadastrar Cliente	04/12/2023
5	1,00		Implementar o caso de uso Listar Clientes	07/12/2023
6	1,00		Implementar o caso de uso Buscar Cliente	07/12/2023
7	1,00		Implementar o caso de uso Analisar Cliente	07/12/2023
1	1,00	Produto	Definir struct Produto	30/11/2023
2	1,00		Implementar menuProduto()	30/11/2023
3	1,00		Implementar a função gravarProduto()	04/12/2023
4	1,00		Implementar o caso de uso Consultar Produto	04/12/2023
5	1,00		Implementar o caso de uso Remover Produto	07/12/2023
6	1,00		Implementar o caso de uso Listar Produto	07/12/2023
7	1,00		Implementar o caso de uso Analisar Produto	07/12/2023

TÉCNICAS DE PROGRAMAÇÃO II

Item	Pontuação	Módulo	Atividade	Data Entrega
1	1,00	Pedido	Definir struct Pedido	30/11/2023
2	1,00		Implementar a função menuPedido()	30/11/2023
3	1,00		Implementar o caso de uso Cadastrar Pedido	04/12/2023
4	1,00		Implementar o caso de uso Consultar Pedido	04/12/2023
5	1,00		Implementar o caso de uso Remover Pedido	07/12/2023
6	1,00		Implementar o caso de uso Analisar Pedido	07/12/2023
7	1,00		Implementar o caso de uso Listar Pedidos	07/12/2023

Cronograma das Etapas de Entrega e Avaliação						
30/11/2023						
	30/11/2023					
		04/12/2023				
			04/12/2023			
				07/12/2023		
					07/12/2023	
						07/12/2023

Matriz de Responsabilidades

Nome do Acadêmico	Item	Item	Item	Item	Item	Item	Item
Pontuação (7,0 pontos)	1,00	1,00	1,00	1,00	1,00	1,00	1,00