

Introdução

O presente relatório apresenta o projeto de desenvolvimento de um sistema de gerenciamento de vendas de smartphones, realizado pelo grupo composto por Lucas Henrique, Pedro Gonçalves e Gabriel Carvalho. O objetivo deste projeto é proporcionar às empresas uma ferramenta eficiente e integrada para controlar e analisar suas transações de venda no mercado de dispositivos móveis.

Nesse contexto, o sistema de gerenciamento de vendas de smartphones proposto oferece diversas funcionalidades essenciais. Através do cadastro detalhado de cada venda, com informações como identificador, nome do smartphone, quantidade, fabricante, ano de criação e descrição do produto, será possível registrar todas as transações de forma organizada e segura.

Além disso, o sistema permitirá o controle preciso do estoque, atualizando automaticamente as quantidades disponíveis de cada smartphone com base nas vendas registradas. Isso evitará erros de estoque, vendas duplicadas e possibilitará um melhor planejamento da empresa com o estoque real que possui. Será possível identificar quais modelos de smartphones estão no estoque, avaliar suas quantidades e tomar decisões estratégicas com base em dados concretos.

Adicionalmente, o sistema oferecerá recursos de pesquisa e filtragem, permitindo aos usuários encontrar smartphones específicos com base em critérios como fabricante, ano de criação, identificador ou preço do produto. Isso proporcionará uma experiência mais ágil e eficiente, tanto para os vendedores quanto para os clientes.

Ordem de leitura

A organização utilizada para manter a ordem dos dados armazenados no arquivo foi a seguinte:

Identificador: Um inteiro que funciona como uma forma de identificação única para cada produto. Cada produto possui um identificador exclusivo que facilita a organização geral dos dados (exemplo: 15).

Nome: Uma string que armazena o nome do produto. No caso de celulares, pode ser o modelo do celular (exemplo: iPhone 14 Pro Max).

Preço: Um número decimal (float) que representa o valor do produto. Isso permite a busca por preços específicos e facilita a compreensão do valor do produto pelo cliente (exemplo: 599,09).

Quantidade: Um inteiro que controla a quantidade de produtos de um determinado modelo em estoque. Isso mantém o estoque organizado e atualizado (exemplo: 3).

Fabricante: Uma string que identifica o fabricante do produto. Pode ser usada para filtrar a busca por fabricante, melhorando a experiência do usuário (exemplo: Apple).

Ano de Criação: Um inteiro que indica o ano em que o produto foi criado. Isso facilita a busca por modelos mais recentes, mais antigos ou de um ano específico (exemplo: 2020).

Descrição: Uma string que descreve as especificações do produto para fornecer informações detalhadas aos usuários (exemplo: O iPhone 14 é um smartphone poderoso, equipado com processador A14 Bionic, câmera dupla de 12MP, tela Super Retina XDR de 6,1 polegadas e suporte para 5G).

Essa estrutura de organização permite a armazenagem e manipulação eficiente dos dados dos produtos, facilitando a busca, filtragem e ordenação com base nos diferentes atributos.

Estruturas Utilizadas

Struct: Foi utilizada apenas uma estrutura "struct" para armazenar como vetor de registro todos os principais dados do vetor, que são: 3 inteiros, 1 string simples, 2 strings com espaço e um valor em ponto flutuante.

Modularização com void: O funcionamento do software é inteiramente baseado em chamadas de funções *void*. Dependendo das ações do usuário, certas funções são chamadas, o que facilita a legibilidade do código e o deixa mais organizado. Alguns exemplos de funções *void* são: exibir o menu, confirmar a saída, fazer a remoção de um item, alterar um item e outras similares.

Switch: A função `int main` é, em grande parte, um switch. Utilizado para seguir caminhos diferentes, baseados nas opções que o usuário seleciona. Além do switch da `main`, existem outros que também são utilizados com a mesma premissa: a de seguir por caminhos diferentes para atender às demandas do usuário (funcionando como um menu do sistema).

For: As estruturas "For" foram, em sua maioria, utilizadas para percorrer o vetor dos dados principais. Para aumentar a eficiência, essas estruturas só executam os comandos dentro delas pela quantidade necessária, ou seja, a quantidade de itens que foram incluídos no vetor de registro.

While: As estruturas While, juntamente com o `getline`, foram utilizadas para executar a leitura dos dados dos arquivos ".csv".

Do While: As estruturas Do While foram utilizadas para garantir que o sistema execute pelo menos uma vez. Além disso, foram utilizadas como sistema de loop para que o sistema continue funcionando até que o usuário requisiute sua interrupção.

If Else: As estruturas condicionais foram utilizadas, em sua grande maioria, para que o usuário receba uma resposta de sua ação. Uma resposta positiva caso o comando seja executado com sucesso, ou uma resposta negativa caso algo seja cancelado ou não seja possível realizar a ação.

Getline: O "getline" foi utilizado para obter as strings com espaço que estão nos arquivos ".csv".

SystemClear: O "SystemClear" é um comando que utilizamos para limpar o terminal do usuário. Como a nossa interface gráfica é toda aplicada no terminal, a organização do mesmo é imprescindível.

Acertos e Erros Durante o Desenvolvimento do Projeto

Ao longo do desenvolvimento do projeto, diversos novos desafios apareceram, sendo preciso uma pesquisa direcionada a resolver cada um deles, alguns exemplos foram:

- Utilização de vetor normal em vez de utilizar struct:
O primeiro erro identificado foi a utilização de um vetor normal para armazenar os principais dados do sistema, em vez de utilizar uma estrutura de dados adequada, como a struct. Essa escolha compromete a organização e legibilidade do código, dificultando o acesso e manipulação dos dados.

Para corrigir esse erro, fizemos a substituição do vetor por uma struct, que agrupa os dados relacionados a uma única entidade, facilitando o gerenciamento dos mesmos.

- Esquecimento de apagar o último elemento do vetor:

Outro erro observado foi o esquecimento de apagar o último elemento do vetor após a remoção de algum telefone. A ausência dessa ação pode causar duplicatas de registros na memória, comprometendo a exatidão do sistema. Para solucionar esse problema, declaramos todos os dados desse elemento como -1, e quando vamos exportar em arquivos, fazemos um “filtro”, deixando ser exportado apenas cadastros com identificadores diferentes de -1.

- Leitura incorreta do arquivo binário:

Mais um erro observado foi na introdução da leitura binária, na qual tivemos um problema na hora da captação das informações dentro de vetores separados, uma parte por usarmos string, na qual corrigimos com a utilização de char. Também tivemos problemas com as informações entrando no vetor errado, como parte do nome indo parar na descrição, etc, viemos a corrigir isso com a utilização de uma estrutura de leitura dos dados até ocorrer a leitura de um “;”, pois foi assim que separamos as informações originalmente.

Mas, apesar de termos cometido esses erros, tivemos alguns pontos interessantes e que acertamos no nosso código:

- Modularização: Utilizamos funções para agrupar blocos de código com funcionalidades específicas. Isso torna o código mais legível, facilitando sua manutenção e reutilização. Evitando o uso excessivo de estruturas e obtendo resultados mais flexíveis.
- Utilização adequada das estruturas de controle: A utilização das estruturas de controle (switch, for, while, do while, if else) foi uma solução que criamos para conseguir fazer uma pré-interface para o usuário conseguir navegar na aplicação como se fosse telas. Isso possibilita ter um melhor entendimento do sistema, assim como também deixa melhor a experiência do usuário.
- Tratamento de erros: Testamos diversas estruturas separadas para encontrar possíveis erros e exceções que possam ocorrer durante a execução do sistema. Isso ajuda a evitar falhas inesperadas e fornecerá uma melhor experiência ao usuário, com mensagens sugestivas do que deve ser inserida no prompt de comando para que tenha uma boa execução do projeto.

Conclusão

Conseguimos, enfim, executar a finalização do projeto com excelência, atendendo a todos os requisitos referentes ao projeto, além de implementar algumas funcionalidades extras, visando um melhor aproveitamento das funcionalidades do sistema.

O projeto serviu de muito aprendizado para todos os integrantes do grupo, tanto de forma individual quanto no trabalho em equipe, realizando diversas reuniões presenciais e a distância, o que nos proporcionou uma ótima troca de conhecimentos e uma confecção do projeto bem executada.

Acreditamos que este projeto de gerenciamento de vendas de smartphones seja uma contribuição valiosa para as empresas do setor, proporcionando um maior controle e eficiência nas operações, além de fornecer informações estratégicas para impulsionar o crescimento do negócio.