

”

**E-fólio A** | Folha de resolução para E-fólio

**UNIDADE CURRICULAR:** Laboratório de Programação

**CÓDIGO:** 21178

**DOCENTE:**

**A preencher pelo estudante**

**NOME:** Henrique Alexandre Cortez Melo

**N.º DE ESTUDANTE:** 2201211

**CURSO:** Licenciatura em Engenharia Informática

## 1.) Foi escolhido o Enunciado A da Atividade Formativa 2

Partindo do enunciado selecionado, o seguinte foi interpretado:

### Ficheiros CSV:

→ produtos.csv

- a)Código
- b)Designação
- c)Preço

→ vendas.csv

- a)Código do produto
- b)Quantidade vendida
- c)Data da venda

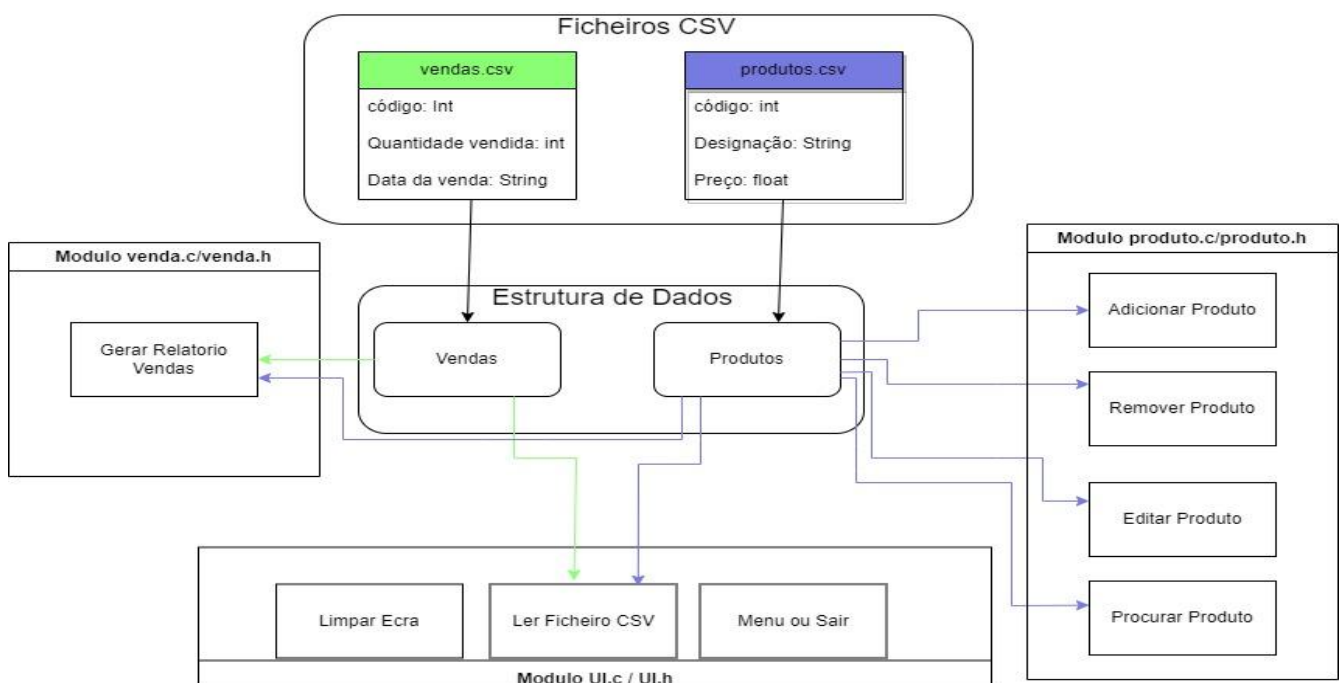
### Funcionalidades que o utilizador poderá usar:

- Adicionar Produtos
- Remover Produtos
- Editar Produtos
- Pesquisar Produtos por nome e código
- Gerar Relatório de Vendas

\*No enunciado encontra-se a seguinte frase: “e atualizar os dados dos produtos em memória de acordo com as vendas realizadas.” No entanto, não nos é apresentado nenhum dado em comum que possa ser atualizado com a quantidade de vendas realizadas, sendo o único dado em comum entre os dois ficheiros .csv o código do produto, mantendo-se este sempre igual.

Como o enunciado é específico e diz-nos exatamente os dados que cada ficheiro .csv terá, existe uma discrepância entre o que nos é dado e o que é pedido, numa situação real, isto seria resolvido contactando o cliente.

A partir desta interpretação foi criado o seguinte mapa visual, para auxílio do programador:



\*o programa feito terá sido escrito e compilado usado em Windows e gcc.

### a)

O programa está separado nos seguintes módulos:

- produto.h/.c → Funções relacionadas com produtos
- vendas.h/.c → Funções relacionadas com vendas
- UI.h/.c → Funções relacionadas com interface/ utilização do programa

Cada módulo contém as suas próprias funções e variáveis, sendo estas disponibilizadas para os restantes módulos.

### b)

O programa utiliza as seguintes duas estruturas de dados:

- Produto (Armazena Informações relacionadas com cada produto)
  - Código
  - Designação
  - Preço
- Venda (Armazena Informações relacionadas com cada venda)
  - Código
  - Quantidade de Venda
  - Data da venda

O programa usa arrays destas estruturas para armazenar os dados lidos dos ficheiros .csv.

### c)

A função principal (main) inicia o programa e fornece o menu principal. Lendo primeiramente os respetivos ficheiros .csv e preenchendo as estruturas de dados com as informações presentes nestes.

Em seguida, entra-se num loop infinito exibindo o menu principal e aguardando o 'input' do utilizador. Cada opção do menu chama uma função correspondente, que executa a função desejada.

O programa exibe o menu até que o utilizador escolha a opção para sair, liberando a memória alocada e terminando a sua execução.

## 3)

### a)

Algumas funções podiam ser reutilizadas sem alterações ou com alterações mínimas, tais como todas as funções do módulo **UI**. Também seria possível para as funções presentes no módulo **produto** fossem usadas para outros enunciados após serem adaptadas para as respetivas estruturas de dados, pois estas simplesmente inserem/removem/editam e procuram dados presentes em ficheiros .csv, dependendo das necessidades específicas de cada enunciado.

A única função que necessitaria de uma adaptação mais extensa seria `gerar_relatorio_vendas`.

**b)**

Para os enunciados B e C, será necessário as seguintes mudanças:

1. Adaptar a estrutura de dados Produto e Vendas para armazenar os diversos dados dos ficheiros .csv em questão. *Por exemplo, (Enunciado B) adaptar a estrutura Produto para incluir as equipas de futebol, incluindo o número de vitórias, derrotas, empates, etc.*
2. Criar funções novas, como a implementação da atualização de informações. *Por exemplo, atualização de número de vitórias e derrotas para as equipas ou número de quartos disponíveis para os hotéis.*
3. Atualizar o menu principal para incluir as novas funções.
4. Modificar módulos existentes e/ou criar módulos novos.