

UNIVERSIDADE VEIGA DE ALMEIDA
GRADUAÇÃO EM SISTEMA DE INFORMAÇÃO

ESTRUTURA AVANÇADA DE DADOS E ANÁLISES
A1 - ATIVIDADE AVALIATIVA VIRTUALIZADA

LUCAS QUINTINO GOMES ARAUJO – 1220200047

Rio de Janeiro – RJ
2024

1. Introdução:

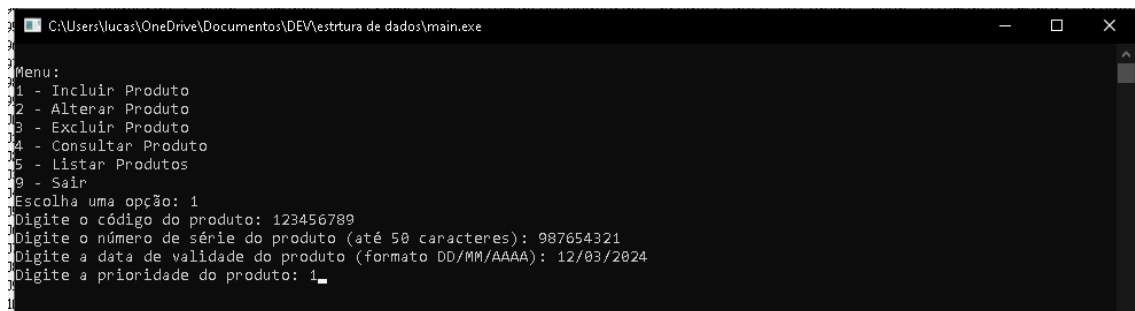
O dono de uma empresa comercial de pequeno porte tem se mostrado preocupado com a falta de informação adequada para a venda dos seus produtos havendo perdas de produto em razão da data de validade do produto. Para amenizar o problema e, considerando que ele não possui condições financeiras para o desenvolvimento de um sistema, foi solicitado o desenvolvimento de um programa que permitisse a consulta e atualização referentes a estes produtos pelo dono da empresa, considerando que a cada venda realizada, deve ser exibido o número de série do produto que deverá ser vendido, tendo como base a data de validade mais recente. A quantidade de produtos existentes na loja é indefinida, uma vez que os estoque são repostos sempre que houver condições financeiras para isso O programa deverá ser iniciado pela manhã e ficará ativo até o final do dia, com isso toda manhã ele irá fornecer informações sobre todos os produtos existentes no final do dia anterior, ao longo do dia ele fará as atualizações que forem necessárias e ao final do dia ele irá listar as informações existentes naquele momento para que sirva de base para o dia seguinte.

2. Aplicação:

Código no GitHub: <https://github.com/QuintinoPro/A1-Estrutura-de-dados>

- Incluir:

```
110 void inserir(MinHeap *heap, Produto produto, int prioridade) {
111     if (heap->tamanho == heap->capacidade) {
112         printf("Erro: Heap cheia.\n");
113         return;
114     }
115
116     int indice = heap->tamanho;
117     heap->elementos[indice].produto = produto;
118     heap->elementos[indice].prioridade = prioridade;
119     heap->tamanho++;
120
121     // Reajusta a heap para manter a propriedade de heap mínimo
122     while (indice > 0 && strcmp(heap->elementos[(indice - 1) / 2].produto.data_validade, heap->elementos[indice].produto.data_validade) > 0) {
123         // Troca os elementos
124         HeapNode temp = heap->elementos[indice];
125         heap->elementos[indice] = heap->elementos[(indice - 1) / 2];
126         heap->elementos[(indice - 1) / 2] = temp;
127
128         indice = (indice - 1) / 2;
129     }
130 }
```



- Alterar:

```
168 void alterarProduto(MinHeap *heap) {
169     int codigo;
170
171     printf("Digite o código do produto que deseja alterar: ");
172     scanf("%d", &codigo);
173
174     // Procurar o índice do produto com o código fornecido
175     int indice = -1;
176     for (int i = 0; i < heap->tamanho; i++) {
177         if (heap->elementos[i].produto.codigo == codigo) {
178             indice = i;
179             break;
180         }
181     }
182
183     if (indice == -1) {
184         printf("Produto com código %d não encontrado.\n", codigo);
185         return;
186     }
187
188     // Solicitar ao usuário as novas informações para o produto
189     printf("Digite o novo número de série do produto (até 50 caracteres): ");
190     scanf("%s", heap->elementos[indice].produto.numero_serie);
191     printf("Digite a nova data de validade do produto (formato DD/MM/AAAA): ");
192     scanf("%s", heap->elementos[indice].produto.data_validade);
193
194     printf("Produto com código %d alterado com sucesso.\n", codigo);
195 }
```

```

Menu:
1 - Incluir Produto
2 - Alterar Produto
3 - Excluir Produto
4 - Consultar Produto
5 - Listar Produtos
9 - Sair
Escolha uma opção: 2
Digite o código do produto que deseja alterar: 2
Digite o novo número de série do produto (até 50 caracteres): 1
Digite a nova data de validade do produto (formato DD/MM/AAAA): 10/02/2024
Produto com código 2 alterado com sucesso.

```

- **Excluir:**

```

197 void excluirProduto(MinHeap *heap) {
198     int codigo;
199
200     printf("Digite o código do produto que deseja excluir: ");
201     scanf("%d", &codigo);
202
203     // Procurar o índice do produto com o código fornecido
204     int indice = -1;
205     for (int i = 0; i < heap->tamanho; i++) {
206         if (heap->elementos[i].produto.codigo == codigo) {
207             indice = i;
208             break;
209         }
210     }
211
212     if (indice == -1) {
213         printf("Produto com código %d não encontrado.\n", codigo);
214     } else {
215         // Remover o produto do heap
216         heap->elementos[indice] = heap->elementos[heap->tamanho - 1];
217         heap->tamanho--;
218
219         printf("Produto com código %d excluído com sucesso.\n", codigo);
220     }
221 }
222 }
223

```

```

Menu:
1 - Incluir Produto
2 - Alterar Produto
3 - Excluir Produto
4 - Consultar Produto
5 - Listar Produtos
9 - Sair
Escolha uma opção: 3
Digite o código do produto que deseja excluir: 2
Produto com código 2 excluído com sucesso.

```

- **Consultar:**

```

224 void consultarProduto(MinHeap *heap) {
225     int codigo;
226
227     printf("Digite o código do produto que deseja consultar: ");
228     scanf("%d", &codigo);
229
230     // Procurar o índice do produto com o código fornecido
231     int indice = -1;
232     for (int i = 0; i < heap->tamanho; i++) {
233         if (heap->elementos[i].produto.codigo == codigo) {
234             indice = i;
235             break;
236         }
237     }
238
239     if (indice == -1) {
240         printf("Produto com código %d não encontrado.\n", codigo);
241     } else {
242         printf("Produto encontrado:\n");
243         printf("Código: %d\n", heap->elementos[indice].produto.codigo);
244         printf("Número de Série: %s\n", heap->elementos[indice].produto.numero_serie);
245         printf("Data de Validade: %s\n", heap->elementos[indice].produto.data_validade);
246     }
247 }
248 }

```

```

Menu:
1 - Incluir Produto
2 - Alterar Produto
3 - Excluir Produto
4 - Consultar Produto
5 - Listar Produtos
9 - Sair
Escolha uma opção: 4
Digite o código do produto que deseja consultar: 1
Produto encontrado:
Código: 1
Número de Série: 1
Data de Validade: 1

```

- **Listar:**

```

250 void listarProdutos(MinHeap *heap) {
251     printf("Listando todos os produtos:\n");
252
253     for (int i = 0; i < heap->tamanho; i++) {
254         printf("Código: %d, Número de Série: %s, Data de Validade: %s\n", heap->elementos[i].produto.codigo, heap->elementos[i].produto.numero_serie, heap->elementos[i].produto.data_validade);
255     }
256 }

```

```

Menu:
1 - Incluir Produto
2 - Alterar Produto
3 - Excluir Produto
4 - Consultar Produto
5 - Listar Produtos
9 - Sair
Escolha uma opção: 5
Listando todos os produtos:
Código: 1, Número de Série: 1, Data de Validade: 1

```

```

Menu:

```