

## Supermercado “Vende++” - suporte para decisões informadas (versão com classes C++)

### INTRODUÇÃO

A competição entre supermercados é muito forte. O marketing para cativar clientes e aumentar a venda deve ser cuidadosamente elaborado. Com o objetivo de aumentar o sucesso, uma campanha publicitária deve ser informada, isto é, deve ser baseada em informação das preferências dos clientes recolhida previamente. A aplicação que vai desenvolver neste trabalho deve permitir a realização de publicidade personalizada, com recurso a um sistema de recomendação. O objetivo do trabalho prático é desenvolver um programa para preparar uma gestão informada das campanhas de promoção de um supermercado. Neste trabalho serão desenvolvidas as funcionalidades necessárias ao desenvolvimento da aplicação completa para uma campanha publicitária de sucesso.

### OBJETIVOS

Este trabalho prático tem como objetivo desenvolver um programa para preparar a aplicação de gestão informada das campanhas de marketing de um supermercado.

Para além dos conhecimentos treinados no primeiro trabalho, ao realizar este segundo, os estudantes terão oportunidade de aplicar os conhecimentos mais avançados de programação em C++ adquiridos até ao momento, e que incluem:

- definição e uso de classes;
- *overload* de funções;
- definição e *overload* de operadores;
- utilização de funções *Template*;
- utilização de iteradores;
- algoritmos de ordenação;
- uso de diversos tipos de estruturas de dados (*maps*, *multimaps*, etc);

### DESENVOLVIMENTO DO PROGRAMA

#### Funcionamento geral

O programa deve começar por ler o ficheiro de clientes, o ficheiro de produtos e o ficheiro das transações (compras) efetuadas pelos clientes. O ficheiro de clientes tem, para cada cliente, um identificador único, o seu nome, a data (DD/MM/AAAA) de adesão ao supermercado e o volume de compras efetuado por esse cliente. O ficheiro de produtos armazena o nome dos produtos disponíveis e o custo do produto. O ficheiro de transações tem, para cada transação, o identificador único do cliente, a data em que a transação foi efetuada e uma lista dos produtos comprados. O volume de compras de um cliente é a soma do valor de cada transação que realizou. O valor de cada transação é a soma do custo de cada produto transacionado. Assuma que o preço de cada produto é fixo e independente da quantidade. As transações devem “sobreviver” aos clientes. Isto é, se o cliente for removido (ou marcado como não existente), as suas transações NÃO devem ser removidas. É importante para fazer boas recomendações ter bastantes dados. O formato dos ficheiros será especificado adiante.

---

## Ficheiro de clientes, de produtos e de transações

Qualquer um dos ficheiros utilizados no programa (clientes, produtos ou transações) são ficheiros de texto que podem ser criados com um editor de texto simples como, por exemplo, o Notepad do Windows. O programa deve começar por perguntar ao utilizador qual o nome dos 3 ficheiros a usar, terminando com um erro caso algum dos ficheiros indicados não exista. Cada um dos 3 ficheiros tem sempre na primeira linha o número de “itens” armazenados nesse ficheiro, como se poderá ver nos exemplos que se seguem.

Apresenta-se a seguir um possível conteúdo de um ficheiro de clientes.

```
4
1 ; Rui Pereira ; 1/3/2008 ; 2.70
2 ; Maria de Lurdes ; 23/4/2009 ; 0.00
6 ; Joana Beires ; 15/10/2010 ; 7.50
16 ; Manuela Moura ; 31/3/2016 ; 8.30
```

*Exemplo de um ficheiro de clientes.*

A primeira linha do ficheiro indica o número de clientes guardados. Cada uma das linhas seguintes contém a seguinte informação acerca de cada cliente: o identificador único, o nome por extenso, a data de adesão (DD/MM/AAAA) e o montante total de compras efetuado pelo cliente. Os valores em cada linha estão separados entre si por “;”.

Um conteúdo possível de um ficheiro de produtos é de seguida mostrado.

```
4
laranja ; 1.5
banana ; 1.2
peixe ; 5.6
batatas fritas ; 0.4
```

*Exemplo de um ficheiro de produtos.*

Cada uma das linhas do ficheiro de produtos guarda o nome de um produto disponível no supermercado e o respectivo custo associado.

Apresenta-se a seguir um possível conteúdo do ficheiro de transações.

```
4
1 ; 1/01/2016 ; laranja, banana
1 ; 3/01/2016 ; peixe, carne, batatas fritas
16 ; 15/03/2016 ; banana, laranja, peixe
6 ; 20/05/2016 ; peixe, batatas fritas, laranja
```

*Exemplo de um ficheiro de transações.*

Cada uma das linhas do ficheiro de transações contém a seguinte informação: o identificador do cliente, a data em que a transação foi efetuada (dia/mês/ano) e a lista de produtos comprados em que os elementos são separados por vírgula.

O valor de compras de cada cliente é igual à soma do valor das transações que efetuou!

## Funcionalidades a implementar

A parte da aplicação que deve implementar neste trabalho inclui as seguintes funcionalidades:

1. Ler e guardar a informação de clientes, produtos (só leitura) e transações em ficheiros;
2. Gestão de clientes: criar, alterar, remover e listar os clientes por ordem alfabética do seu nome.
3. Gestão de transações: listar todas as transações num intervalo de tempo data; listar por ordem cronológica as transações de um cliente cujo nome foi dado;
4. Efetuar compras (adicionar transações). Ao efetuar uma compra deve actualizar o montante total de compras do correspondente cliente;

5. Visualizar de modo formatado informação: de um cliente ou de todos os clientes;
6. Visualizar de modo formatado as transações: de um cliente; de um dia; entre duas datas especificadas; ou todas as transações;
7. Calcular os **Bottom10** (piores) clientes. Um cliente é tanto pior quanto menor for o montante de compras efetuado.
8. Efetuar publicidade personalizada para um cliente, para os **Bottom10** clientes utilizando o sistema de recomendação.

O objetivo do sistema de recomendação a usar para a publicidade personalizada é recomendar ao cliente alvo produtos que clientes com hábitos de compras “semelhantes” a ele tenham comprado mas ele ainda não comprou. Dois clientes são muito “semelhantes” se o histórico de compras revela que compram praticamente os mesmos produtos. A utilização de um sistema de recomendação para personalizar a publicidade a enviar a um cliente alvo é realizada do seguinte modo.

1. Escolha o cliente alvo.
2. Crie uma matriz de booleanos com um número de linhas igual ao número de clientes e número de colunas igual ao número de produtos. Inicialize a matriz com **false**.
3. Estabeleça uma associação entre o **id** de cada cliente e uma linha da matriz de modo a aceder à linha do cliente dado o seu **id**.
4. Percorra a lista de transações e, para cada transação, identifique o cliente e, na linha desse cliente na matriz coloque a **true** os produtos registados nessa transação.
5. Usando a linha do cliente alvo como referência identifique todos os clientes que satisfazem as seguintes duas condições: compraram o maior número de produtos comprados pelo cliente alvo; e compraram pelo menos um produto não comprado pelo cliente alvo.
6. Sugira ao cliente alvo o produto mais frequente da lista que recolheu no ponto anterior e que ele não tenha comprado.

Exemplo de lista simplificada das transações:

- 1 ; arroz, farinha, massa, pipocas
- 2 ; farinha, arroz, batatas, pipocas
- 3 ; massa, farinha, arroz, pipocas
- 4 ; sumo, arroz, farinha, massa

Considere que o cliente alvo comprou: arroz, farinha e massa.

Os clientes mais “semelhantes” são o 1, 3 e o 4 que compraram os 3 produtos que o cliente alvo comprou. Os produtos que os clientes 1, 3 e 4 compraram e que o cliente alvo não comprou são: pipocas e sumo. Como pipocas é o produto mais frequente, o supermercado deve publicitar as pipocas ao cliente alvo.

Para a campanha dos Bottom10 comece por reunir os produtos comuns aos 10 piores clientes. Encontre os outros clientes (*clientes interessantes*) que tenham comprado pelo menos todos os produtos comuns aos Bottom10 + 1. Faça um histograma dos produtos comprados pelos *clientes interessantes* e que não foram comprados por todos os **Bottom10** (mas podem ter sido comprados por alguns dos **Bottom10**). Percorra o histograma do produto mais frequente para o menos frequente. Proponha como promoção o primeiro produto que encontrar e que não tenha sido comprado por nenhum dos **Bottom10**. Se não houver nenhum produto nessas condições proponha o produto mais frequente entre os clientes interessantes e menos frequente entre os **Bottom10**.

Utilize um **map** para associar nome de um produto ao seu índice no vector de produtos e use um **multimap** para associar um cliente à sua lista de transações.

## MELHORIAS OPCIONAIS

Apresentam-se a seguir alguns desafios para desenvolvimentos adicionais:

- Melhorar as estruturas de dados de modo a tornar a aplicação mais eficiente.

## NOTAS SOBRE O DESENVOLVIMENTO

### Funcionamento geral do programa

O funcionamento geral do programa é o seguinte:

- Ler o conteúdo dos ficheiros de clientes, produtos e transações e guardar numa estrutura de dados adequada.
- Começar por exibir um menu onde aparecem as opções que implementam as funcionalidades: gestão de clientes; visualização de informação (clientes, produtos e transações); efetuar compras (transações); e personalizar publicidade.
- No final, gravar a informação dos clientes, e das transações caso tenha havido alterações.

Devem ser tomadas as precauções necessárias para evitar que o programa deixe de funcionar corretamente devido a entradas inválidas do utilizador, nomeadamente, valores fora da gama admissível.

### Escrita do código

- Na escrita do código devem ser respeitadas as indicações dadas nas aulas, nomeadamente, em relação aos seguintes aspetos:
- Escolha das estruturas de dados mais adequadas para representar os dados do programa.
- Escolha adequada dos identificadores de tipos, variáveis e funções.
- Estrutura modular do código.
- Escolha adequada das classes
- Separação em dois ficheiros (.h e .cpp) da definição de cada classe e da implementação dos seus membros função
- Comentários ao código.

### ENTREGA DO TRABALHO

- Criar uma pasta com o nome **TxGyy**, em que **x** representa o número da turma e **yy** representa o número do grupo de trabalho – por exemplo, T2G05, para o grupo 5 da turma 2 - e copiar para lá o código-fonte do programa (apenas os ficheiros com a extensão .cpp e .h). Incluir também um ficheiro ReadMe.txt (em formato de texto simples) indicando o estado de desenvolvimento do trabalho, isto é, se foram cumpridos todos os objetivos ou, caso contrário, quais os que não foram cumpridos, ou ainda que melhorias foram implementadas, se for esse o caso. Tudo o que não estiver especificado pode ser decidido pelos elementos do grupo de trabalho, sendo as decisões adoptadas descritas no mesmo ficheiro ReadMe.txt.
- Compactar o conteúdo desta pasta num ficheiro **Gxx.zip** ou **Gxx.rar** e submeter este ficheiro na página da disciplina de Programação, no Moodle da FEUP. Não serão aceites entregas por outras vias.
- Os estudantes deverão informar o docente da turma TP respetiva qual a constituição dos grupos de trabalho, até 30/abril; a lista dos grupos constituídos será oportunamente publicada no Moodle.
- Data limite para a entrega: 20/maio/2016 (às 23:55h).