



Algoritmos e Tipos Abstratos de Dados

Mini-Projeto 2

Ready-Wear Soft-Ware

1 Requisito básico

A aplicação terá de ser desenvolvida em linguagem C.

A entrega do programa, bem como a demonstração do seu funcionamento, terá de ser feita como projeto em Codeblocks.

2 Objetivo Pedagógico

Prática no uso de árvores binárias de pesquisa, listas e outros TADs, tais como *queues* e *stacks*.

3 Descrição do Problema

Uma loja de pronto-a-vestir pretende criar um registo dos eventos (visitas e compras) gerados pelos seus clientes, de modo a poder obter indicadores que lhe permitam compreender os hábitos dos clientes. Considera-se que o programa entrou em funcionamento no dia de abertura da loja.

Para além da produção de um relatório com um conjunto de indicadores sobre os hábitos dos clientes (descritos na secção 4.4), são pretendidas as seguintes funcionalidades: imprimir registos no ecrã e em ficheiro ordenados por nome de cliente em formato de árvore binária, imprimir lista de clientes ordenada por identificador em ficheiro e no ecrã e passar um cliente para o estado inativo sempre que não entre na loja após uma duração previamente fixada.

4 Desenvolvimento do trabalho

Para o desenvolvimento do trabalho será necessário implementar uma única aplicação de consola (interfaces visuais não são consideradas para a avaliação) que resolva o problema descrito acima e a escrita de um relatório, que são descritos nas secções seguintes. É da responsabilidade do aluno, seleccionar as abordagens de implementação (estáticas, semi-estáticas ou dinâmicas) mais eficientes para o problema em causa.

4.1 Implementação

Para o desenvolvimento do trabalho será necessário implementar uma árvore binária de pesquisa (organizada pelo nome do cliente) para representar os eventos gerados pelos clientes (compras e visitas) e uma lista para armazenar os clientes da loja, onde o *rank* da lista corresponde ao id do cliente (no *rank* *n* está o cliente com o id *n+1*). Considera-se assim que os clientes são representados pelo seu nome, que se considera único (e.g., manuel), pelo seu género (feminino ou masculino), e por um campo de identificação numérico (id.), atribuído automaticamente pelo programa (função *auto-increment*), pelo valor total acumulado de compras efetuadas e o estado do cliente (ativo ou inativo). O cliente passa a inativo sempre que não entre na loja após uma duração previamente fixada (considere 30 dias de duração para testar a aplicação) e o seu valor total de gastos é colocado a zero. Este cliente é removido da árvore de registos. Quando o cliente volta a efetuar uma compra, o seu estado volta a ativo, e um novo registo de eventos é adicionado à árvore.

As datas dos eventos devem ser implementadas com dia, mês e ano (e.g., 31/12/2016). Considera-se que a data de abertura de loja é 31/01/2016.



Algoritmos e Tipos Abstratos de Dados

Mini-Projeto 2



Ready-Wear Soft-Wear

4.2 Dados de Entrada

A aplicação receberá os dados de entrada através da interface com o utilizador e através de um conjunto de ficheiros.

A entrada de informação que será feita por intermédio de ficheiros diz respeito aos eventos que simulam a visita à loja e compra de artigos por parte dos clientes. Cada ficheiro diz respeito a uma única data, não sendo possível carregar ficheiros com datas iguais ou anteriores à data da aplicação. A data da aplicação é atualizada cada vez que um novo ficheiro de dados é carregado com a data do ficheiro. Estes ficheiros têm o seguinte formato:

- Primeira linha: data tem o formato dia/mês/ano; este campo ocupa a primeira linha e ocorre uma só vez;
- Nas linhas seguintes: em cada linha, um tuplo nome,género,gasto em que: nome representa univocamente o cliente; género é 'm' ou 'f'; gasto representa o total despendido em compras, havendo uma linha por visita ou compra efetuadas (um gasto de 0€ representa uma visita, e um gasto negativo representa uma devolução).

Exemplo do conteúdo de um ficheiro:

No exemplo, as três primeiras linhas correspondem a uma compra do manuel. Repara-se que houve uma devolução – a venda do segundo produto foi anulada. De modo a facilitar a execução do projeto, considera-se que a devolução aparece imediatamente após a compra respetiva. Note-se ainda que o manuel voltou à loja mais tarde para uma segunda compra. Portanto, o manuel visitou a loja duas vezes nesse dia.

```
31/12/2016
manuel,m,100
manuel,m,200
manuel,m,-200
alberto,h,0
diana,f,123
manuel,m,100
```

4.3 Interface com o Utilizador

A aplicação deverá disponibilizar o seguinte Menu:

- Carregar Ficheiro de Entrada**
- Gerar Relatório**
- Imprimir Árvore de Registos**
- Imprimir Lista de Clientes**
- Sair**



Algoritmos e Tipos Abstratos de Dados

Mini-Projeto 2

Ready-Wear Soft-Ware

- A. A opção “Carregar Ficheiro de entrada” deve pedir o nome do ficheiro de dados (especificado em 4.1) e processar os dados lidos.
- B. A opção “Gerar Relatório” deverá imprimir no ecrã e no ficheiro o relatório especificado em 4.4
- C. A opção “Imprimir Árvore” deverá imprimir a árvore de registo de eventos (no ficheiro *arvore.txt*) e no ecrã, onde em cada nó da árvore é impresso apenas o nome do cliente e o número total de eventos guardados para esse (ver 4.4).
- D. A opção “Imprimir Lista” deverá imprimir a lista de clientes (no ficheiro *lista.txt*) e no ecrã (ver 4.4).
- E. Esta opção deverá encerrar a aplicação.

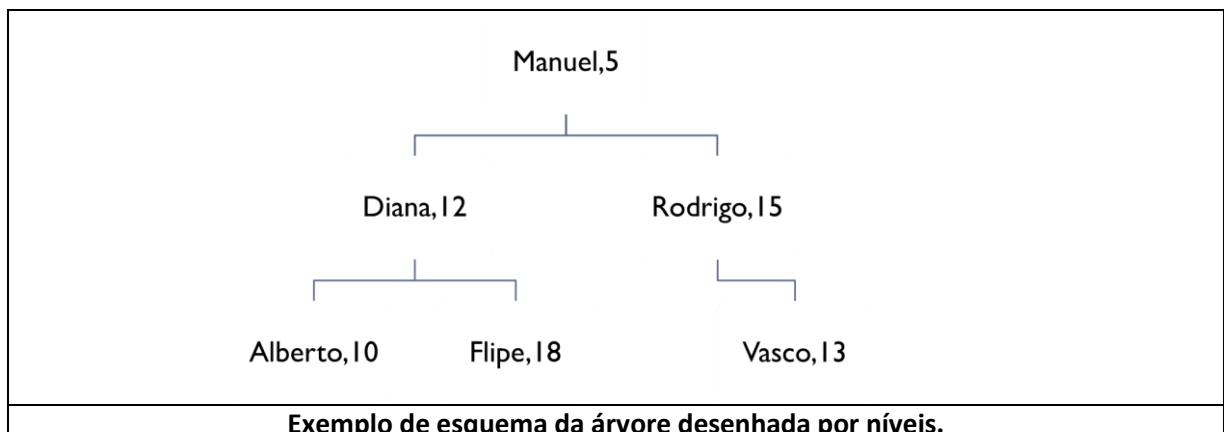
4.4 Dados de Saída

A opção B, “Gerar Relatório”, deverá produzir um ficheiro com o seguinte formato:

```
Número de clientes que visitaram a loja:valor
Número de clientes que efetuaram pelo menos uma compra:valor
Número de clientes que só visitaram a loja:valor
Valores de consumo totais
    dia/mês/ano :
        nomeCliente,valorGasto
        nomeCliente,valorGasto
    dia/mês/ano :
        nomeCliente,valorGasto
        nomeCliente,valorGasto
Valores de consumo médios:
    nomeCliente,valorMedio
    nomeCliente,valorMedio
Género que mais visita a loja: genero
Cliente que mais gastou na loja: id,nomeCliente,genero
```

Formato geral do relatório

A opção C deverá “desenhar” a árvore por níveis, onde cada nó da árvore é representado no formato: *nomeCliente, totalEventos*.



Exemplo de esquema da árvore desenhada por níveis.



Algoritmos e Tipos Abstratos de Dados

Mini-Projeto 2



Ready-Wear Soft-Wear

A opção D deverá “desenhar” uma lista de clientes, aparecendo um cliente por linha.

```
1 - Manuel, masculino, 2050Euros, estado ativo
2 - Rodrigo, masculino, 0Euros, estado inativo
3 - Diana, feminino, 25Euros, estado ativo
4 -...
```

5 Relatório do Projeto

No relatório deverão constar as seguintes secções:

- Índice.
- Descrição dos tipos abstratos de dados usados e explicação de qual a abordagem utilizada para a sua implementação.
- Análise da complexidade dos algoritmos de:
 - Inserção de um novo registo na árvore.
 - Remoção de um registo na árvore.
 - Inserção de uma nova visita à loja de um cliente.
 - Determinação do género que mais visita a loja.
- Conclusões e Limitações.



Algoritmos e Tipos Abstratos de Dados

Mini-Projeto 2

Ready-Wear Soft-Ware

6 Critérios de Avaliação

Usar definições recursivas a par com implementações iterativas, abordando o assunto de forma crítica no relatório.

Uso diversificado dos TADs estudados. Sendo este o último projeto da UC, nele culmina a oportunidade de uso prático das matérias estudadas.

A interface do programa será o tópico que menos se valorizará. Ou seja, não prejudica o aluno desenvolver uma interface minimalista.

Descrição	Cotação
Implementação da árvore binária e da lista	2,5 valores
Carregar ficheiros com dados de entrada	3,5 valores
Produção correta do relatório com indicadores	4 valores
Impressão da árvore/lista	3,5 valores
Uso correto de outros TADS lecionados.	1 valores
Remoção de clientes antigos e devoluções	2,5 valores
Relatório	3 valores

7 Regras e Instruções

O não cumprimento das seguintes regras implica uma penalização na nota do trabalho. Situações não previstas devem ser comunicadas à Prof^a. Patrícia Macedo, responsável pela UC, atempadamente.

Regras:

- O Mini-Projeto deverá ser elaborado por dois alunos do mesmo docente.
- A nota do Mini-Projeto será atribuída individualmente a cada um dos elementos do grupo após a discussão. As discussões poderão ser orais e/ou com perguntas escritas. As orais poderão ser feitas aos elementos do grupo presentes em simultâneo ou individualmente.
- A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos com semelhanças; original e cópias.
- No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente a que se destina.
- O trabalho deverá ser submetido no Moodle, no link do respetivo docente de laboratórios criado para o efeito, até às 23h55 do dia 6 de Junho de 2016. Para tal terão de criar uma diretoria com o nome nomeAluno1_númeroAluno1-nomeAluno2_númeroAluno2 onde colocarão o ficheiro do relatório em formato pdf e a diretoria com o projeto CodeBlocks da implementação da aplicação desenvolvida. Os alunos terão de submeter essa diretoria compactada no formato zip no Moodle. Apenas será permitido submeter um ficheiro.
- Não serão aceites trabalhos que não cumpram na íntegra o ponto anterior.
- As datas das discussões serão publicadas após a entrega dos trabalhos.