

COMPRA FÁCIL

INTRODUÇÃO

A tradicional folhinha “lista das compras” não é propriamente um exemplo de “tecnologia ao serviço do cidadão”. Usando uma aplicação, a lista pode ser “elaborada” sempre que surgem as necessidades (inserindo a nova compra a fazer) e “gerida” no decorrer do processo de compra, “picando” os produtos que se vão adquirindo.

Dadas as vantagens óbvias, pretende-se desenvolver uma aplicação que permita gerir as compras a realizar.

Como, tipicamente, se pode ir a vários supermercados e a outras lojas adquirir o que se necessita, pretende-se que a aplicação a desenvolver permita a existência de mais do que uma lista de compras. Assim, havendo várias listas, sempre que se verificar a falha de um artigo, basta adicioná-lo à lista pretendida. Depois, quando se for às compras, usa-se a lista apropriada e registam-se as aquisições (marca-se o produto - item da lista - como adquirido), sempre que se colocar no carrinho de compras.

A primeira vez que se pretender inserir um produto novo (através da indicação do nome) numa qualquer lista de compras, o produto será criado se o utilizador o entender, podendo então ser indicada a categoria a que pertence (ex. carne, peixe, congelados, enlatados, vegetais, frutas, etc.), a unidade de aquisição e o valor unitário, sendo depois o produto (item) acrescentado à lista de compras. Da próxima vez que se pretender adicionar esse produto a uma lista de compras, ele deverá ser logo inserido (pois já existe como produto).

Cada produto (item) da lista de compras é caracterizado pelos seguintes dados:

Nome; Quantidade e Estado (já adquirido ou não - o tradicional visto, quando se adquire o bem).

TAREFAS

1. Como podem concluir da descrição da aplicação atrás apresentada, além do ficheiro “Listas”, relativo às diversas listas existentes, do ficheiro “ItensLista”, os itens a adquirir registados em cada lista e aos produtos (ficheiro “Produtos”), haverá também um ficheiro “Categorias”, cada uma caracterizada por uma sigla, um nome e outros atributos que julgarem importantes.

Os dados existentes em todos estes ficheiros e num outro (“ItensCabaz”) que será necessário, resultante das funcionalidades solicitadas nas questões 3.a) e b), deverão ser lidos para as estruturas de dados em memória, aquando do arranque da aplicação e/ou a pedido do utilizador, devendo poder especificar-se o tipo de ficheiro a ler (se .txt ou binário).

Obs.: Se acharem pertinente, podem acrescentar dados ou características aos elementos atrás descritos. Será importante existir um dado (que irão depois conhecer como atributo ou campo) que permita “ligar” ou “relacionar” os elementos entre si. Por exemplo, no

Produto, pode-se incluir um dado adicional “Codigo” que permitirá depois “relacioná-lo”, de forma mais simples e fácil, com um elemento da entidade ListaCompras. Nesta situação, p. ex., poderá existir um produto de nome “Arroz Carolino (Ceifeira)”, cujo código é 123. Este produto, para ser registado no ItensLista, pode apenas ser “identificado” pelo respetivo código e não pelo nome. Vejam-se as vantagens: além de ocupar muito menos espaço para se registar cada item na Lista de Compras (2 ou 4 bytes usando o código vs. 15, 20 ou mais, para o nome), a pesquisa do produto será muito mais simples, pois não será necessário usar o nome, mas o código. Repare-se que, além do “tamanho” do dado ser muito mais pequeno, uma comparação de um tipo numérico é muito mais simples e rápida (em termos de execução) do um tipo “string” (em C, como é sabido, um vetor de caracteres).

2. A aplicação a desenvolver deverá responder aos seguintes requisitos:

- a) Permitir a alteração/eliminação de Produtos, ListaCompras (produtos numa qualquer lista) e ItensCabaz (produtos existentes num dado cabaz);
- b) Permitir a inserção/alteração/eliminação de Categorias e Listas;
- c) Permitir a inserção de novos itens (produtos) a adquirir, numa qualquer ListaCompras (observando os requisitos mencionados nas questões 3.a) e b));
- d) Consultar uma determinada lista de compras, com total de custo, devendo os produtos ser separados por categoria e dentro de cada categoria, estarem ordenados alfabeticamente, permitindo assim que, quando se está a efetuar as compras num supermercado, não se tenha de ir várias vezes à mesma secção, poupando o tão precioso tempo;
- e) Registar os produtos adquiridos (marcar o produto como adquirido), operação a efetuar quando se está em “modo compras”, devendo depois o item da lista de compras ser apresentado de forma diferente dos itens ainda não adquiridos;
- f) Transferir produtos de uma lista de compras para outra, à escolha do utilizador;
- g) Consultar os produtos de uma categoria ou de todas, ordenados por ordem alfabética crescente de nome;
- h) Permitir a eliminação de todos os produtos já adquiridos da lista de compras corrente (quando se está em “modo compras”);
- i) Permitir gerar um fich. txt com uma lista de compras, onde as primeiras linhas (as necessárias), indicarão o nome da Lista e a data de geração do ficheiro, sendo depois listados os produtos da lista de compras, agrupados por categoria e ordenados alfabeticamente crescente em cada categoria;
- j) Os dados da aplicação devem ser gravados em ficheiros binários, ao terminar a execução da aplicação.

3. Há duas funcionalidades muito úteis que devem também ser atendidas:

- a) Como os produtos que se adquirem acabam por se repetir, pretende-se que, para cada lista de compras, exista um denominado “cabaz” (ItensCabaz) que tem os produtos que já foram alguma vez inseridos nessa lista. Sempre que se marque um produto como “adquirido” numa dada lista de compras, produto esse que não exista ainda no cabaz respetivo, a aplicação irá incluí-lo, de forma automática, no cabaz dessa lista.
- b) Quando se estiver a registar novos produtos (itens) para uma lista, devem ser mostrados os produtos do cabaz (os ItensCabaz), mas só aqueles que ainda não estiverem na lista de compras, agrupados por categoria, em que cada produto é sinalizado por um número sequencial, devendo permitir-se a seleção de um deles (especificando esse número), que, assim, passa diretamente como produto (novo item) inserido na lista das compras. Claro que, se o produto pretendido não existir no “cabaz”, tem-se o requisito da questão 2.c).

CONSTITUIÇÃO DOS GRUPOS

Os grupos de trabalho deverão ser constituídos, no máximo, por 4 alunos, devendo as inscrições ser feitas, através de mensagem para jfialho@estgv.ipv.pt.

OBSERVAÇÕES

- Deverá criar as estruturas de dados necessárias;
- Na implementação deverá usar ficheiros;
- A validação de dados será valorizada;
- O trabalho deve ser implementado em linguagem C e deve apenas usar os conhecimentos apreendidos nas aulas de Algoritmos e Programação;
- A funcionalidade do programa é, nesta unidade curricular, muito mais importante que os aspetos estéticos. Será valorizada a clareza e simplicidade do código. Deverá assim usar funções e estruturar o programa de modo a torná-lo simples, bem estruturado, e sem repetições desnecessárias de código. Será ainda valorizada a conveniente indentação do código e a inclusão de comentários, sempre que tal for interessante do ponto de vista da legibilidade;
- Possíveis melhorias incluídas no programa, devidamente fundamentadas e enquadradas com o programa desta unidade curricular, serão consideradas e valorizadas;
- Deve ser elaborado um relatório que descreva o trabalho produzido e as funcionalidades implementadas;
- Não serão admitidos plágios, mesmo que parciais - trabalhos copiados terão nota ZERO (tanto quem copiou como quem deixou copiar);
- Caso se revele necessário, poderão ser feitas atualizações ou alterações a este enunciado, pelo que os alunos deverão estar atentos a esta eventualidade. Qualquer atualização ou alteração será devidamente anunciada no Moodle desta unidade curricular.

ENTREGA

O trabalho a entregar deve consistir num único ficheiro comprimido com o formato ZIP, devendo terem em atenção as seguintes indicações gerais:

1. O relatório solicitado deve estar em formato pdf;
2. A submissão deve ser realizada por um dos elementos do grupo, no link apropriado existente na página da unidade curricular no Moodle. Este ficheiro comprimido deve conter todos os ficheiros com o código fonte da aplicação, bem como os ficheiros de dados necessários à execução, o relatório e um ficheiro de texto com a identificação dos alunos (nome e número). Em último recurso, o trabalho pode ser enviado por e-mail, diretamente para jfialho@estgv.ipv.pt.

APRESENTAÇÃO E DEFESA

A defesa dos trabalhos, com a presença de todos os elementos do grupo de trabalho, é obrigatória em data a indicar oportunamente.