

PROJETO PRÁTICO DE AEDA

Tema 2 - Transações Online (Parte 2)

Turma 4 - Grupo 3:

João Marinho (up201905952@fe.up.pt)

Miguel Rodrigues (up201906042@fe.up.pt)

Tiago Silva (up201906045@fe.up.pt)

DESCRIÇÃO DO PROBLEMA

Aspetos centrais sobre o funcionamento de uma loja *Online*.

IMPLEMENTAÇÃO DE UMA LOJA ONLINE

- A **empresa** *BuyNow* decidiu implementar um sistema de apoio ao **cliente**:
 - Através deste sistema é possível estabelecer o contacto entre Empresa-Clientes e vice-versa.
- A *BuyNow* decidiu também introduzir um sistema de entregas ao domicílio:
 - O **cliente** tem a opção de escolher qual o método de envio;
 - As novas encomendas devem ser carregadas para as carrinhas com menor volume disponível, estas só devem sair para a distribuição se a sua ocupação for superior a 90% ou então até decisão do administrador.
- Os **fornecedores** são caracterizados por um **nome**, um **NIF** e um **conjunto de produtos**.
- Para cada **produto** vendido na loja passará a existir um conjunto de fornecedores.
- Este sistema deve manter o acompanhamento dos **stocks** dos **produtos**:
 - Permitir a **reposição** de **stock** da **loja online** a partir das **lojas físicas** (se possível);
 - Caso o ponto acima não seja exequível, então reabastecer os **stocks** através do fornecedor.

DESCRIÇÃO DA SOLUÇÃO

Algoritmos e operadores relevantes.

IMPORTÂNCIA DOS ALGORITMOS

- **Pesquisa nas BST's:**

- A Árvore Binária de Pesquisa é uma estrutura de dados que permite a pesquisa de elementos com uma complexidade temporal da ordem $O(\log n)$;
- Estas requerem que o **operator<** se encontre implementado para que possa haver comparações entre os diferentes elementos.
- Na prática, se nos encontrarmos num determinado nó, todos os elementos à sua esquerda são de menor e em contrapartida todos os elementos à sua direita são de maior valor, isto comparando com o valor do próprio nó.

- **Filas de Prioridade:**

- As filas de prioridade são baseadas em *Heaps* – árvores binárias completas e que podem ser representadas em vetores (de modo a evitar o uso de apontadores).
- Por definição os *Heaps* da STL são *max-heaps*, pelo que a implementação do **operator<** deve ser efetuada de forma cuidadosa.
- Pelo ponto anterior, no topo da fila está o maior elemento sendo que a ordenação dos mesmos é feita com recurso ao *HeapSort* – complexidade temporal na ordem $O(n * \log n)$;
- Por último, a complexidade temporal média de inserção e remoção é da ordem $O(\log n)$;

IMPORTÂNCIA DOS ALGORITMOS

- **Tabelas de Dispersão:**

- As tabelas de dispersão permitem a procura por um elemento seja da ordem $O(1)$, ou seja, constante.
- Para isso é necessário definir uma função de *hashing* que permite a identificação dos elementos de uma forma única.
- Uma boa função de *hashing* é fundamental de forma a evitar um elevado número de colisões, o que obriga à resolução das mesmas por métodos como a sondagem quadrática ou linear.
- Na STL, esta estrutura de dados é representada pelo contentor ***unordered_set***.

```
int operator()(ClientMessage * msg) const
{
    int v = 0;
    for(auto elem : msg->getMsg())
    {
        v += 37*elem;
    }
    for(auto elem : msg->getEmail())
    {
        v += 37*elem;
    }
    return v;
}
```

Figura 2 – Implementação da função de *hashing*.

ESTRUTURA DOS FICHEIROS

Processamento dos dados da empresa.

FICHEIROS E A SUA UTILIDADE

- Para o armazenamento dos dados relativos a transações, listas de clientes e produtos foram usados ficheiros de texto.
- O programa, quando é iniciado, efetua a leitura dos conteúdos do ficheiro armazenando-os na estrutura de dados apropriada. Por outro lado, a escrita dos dados para o ficheiro de texto ocorre no quando o utilizador encerra o programa.
- Com a introdução destas novas funcionalidades adicionámos novos ficheiros de texto, relevantes no contexto deste problema.

- **Nome da Empresa**
 - Localização das lojas físicas
- **Lista de Clientes**
 - Clientes Não Registrados
 - Clientes Registrados
 - E-Mail e métodos de pagamento
 - Mensagens trocadas com os clientes
- **Lista de Produtos**
 - Descrição e Preço
 - Referências de Stock para cada loja
 - Stock mínimo
 - Stock disponível
- **Lista de Transações**
 - Para efeitos estatísticos

SUPPLIERS.TXT

- **Lista de Fornecedores**
 - Dados do Fornecedor
 - Nome e NIF
 - Lista de Produtos Vendidos
 - Descrição, preço e quantidade

FLEET.TXT

- **Lista de Veículos**
 - Matrícula da viatura
 - Capacidade total de carga
 - Capacidade restante de carga
 - Carga carregada

FUNCIONALIDADES

Facilidade de uso na ótica do utilizador.

FUNCIONALIDADES

- O programa permite, ao seu utilizador, de uma forma simples e intuitiva a execução de várias operações de forma a tornar mais fácil a sua utilização.
- **CRUD** (*Create, Read, Update, Delete*) - (OK):
 - Nesta componente inserem-se todos os objetos das novas classes criadas: *Supplier*, *Van* e ainda *Message*.
- **Listagem** - (OK):
 - É possível ao utilizador e ao gerente de loja de verem todo o seu registo de mensagens trocadas.
 - Ao gerente da loja também é dada a possibilidade de ver o estado de toda a sua frota e também de listar todos os fornecedores de um dado produto, seja por nome ou preço.
- **Pesquisa** - (OK):
 - O administrador pode fazer de uma forma simples e intuitiva uma pesquisa de um produto sobre um conjunto de fornecedores.
 - O administrador do sistema pode, por exemplo, ver estatísticas de vendas num determinado período de tempo à sua escolha, para tal, foram implementados algoritmos de pesquisa sobre as estruturas de dados que armazenam as Transações.

FUNCIONALIDADES EXTRA

- Como foi dito anteriormente o programa procura trazer ao utilizador uma plataforma simples e de utilização amigável para compras online.
- Como tal, uma das ferramentas mais úteis ao administrador é as reposições de stock automáticas. Otimizando a eficiência e os custos para a companhia.
 - Primeiro, quando se atinge um esgotamento de stock na loja *online*, procura-se nas lojas físicas se há stock suficiente para repor na loja *online*.
 - Caso não seja possível essa reposição através das lojas físicas, então adquire-se produto ao fornecedor – aquele que o utilizador considere a melhor opção.
 - Esta funcionalidade acabou por ser adaptada com a introdução das *BST's*.

DIFICULDADES

Quais as maiores desafios enfrentados pelo grupo?

DISTRIBUIÇÃO DAS TAREFAS

- Neste projeto, coube a cada elemento do grupo implementar uma das três novas funcionalidades propostas.
- Além disso a coordenação e distribuição equitativa das tarefas entre todos os elementos do grupo mostrou-se, desde o início, crucial para que fossem atingidos os objetivos esperados.
- **João Marinho (1/3)**
 - Reimplementação do sistema de *Stocks* com recurso às BST's;
 - Aprimoramento da interface (visível para o administrador).
- **Miguel Rodrigues (1/3)**
 - Implementação do sistema de entregas ao domicílio usando uma fila de prioridade;
 - Aprimoramento da interface (visível para o administrador e para o cliente).
- **Tiago Silva (1/3)**
 - Implementação do sistema de apoio ao cliente com recurso a tabelas de dispersão;
 - Aprimoramento da interface (visível para o administrador e para o cliente).