

## **Trabalho Prático de Estruturas de Informação**

**1º Semestre**

**Ano Letivo 2014/2015**

Pretende-se com este trabalho desenhar e implementar um simulador de armazém disponibilizando diversas funcionalidades, desde a criação automática do armazém até ao armazenamento e expedição de produtos no armazém.

Considere que um armazém é constituído por vários depósitos, ou espaços de armazenagem com características diferentes, salientando-se dois tipos: depósitos frescos com refrigeração e depósitos normais. Ambos os tipos de depósitos são constituídos por várias paletes de armazenamento que podem ser acedidas diretamente. Nos depósitos frescos os produtos são guardados nas paletes por ordem de chegada e nos depósitos normais os produtos são empilhados nas paletes. Cada depósito é identificado por uma chave, sabe-se o número de paletes de armazenamento que contém, capacidade máxima das paletes e a área total que ocupa.

Os depósitos frescos, com  $N$  paletes de armazenamento todas com igual capacidade, os produtos são distribuídos pelas paletes por ordem de chegada e de modo a que todas tenham sensivelmente a mesma carga. A expedição de produtos destes depósitos é feita por ordem de chegada. Os depósitos normais também com  $N$  paletes de armazenamento, têm a particularidade das paletes pares (0,2,4,...) terem o dobro da capacidade das paletes ímpares, ou seja, a primeira paleta tem o dobro da capacidade da segunda, que por sua vez tem metade da capacidade da terceira e assim sucessivamente. Os produtos são armazenados de modo a preencher primeiro as paletes de maior capacidade e só depois destas estarem todas completamente cheias são preenchidas as paletes de menor capacidade. A retirada de produtos destes depósitos processa-se de modo inverso, ou seja, são primeiro retirados os produtos das paletes de menor capacidade e só após todas estas estarem vazias são retirados produtos das paletes de maior capacidade.

O simulador de armazém deverá permitir gerar um armazém de acordo com parâmetros previamente definidos como, número de depósitos, normais e frescos, para cada tipo de depósito o número de paletes de armazenamento e as distâncias entre os depósitos, assuma que cada depósito se liga apenas a outros dois depósitos distintos. Todos os parâmetros para definição do armazém devem ser gerados recorrendo a funções aleatórias parametrizáveis com valores máximos e mínimos. A estrutura do armazém deve ser guardada em ficheiro de texto.

O sistema deverá também disponibilizar funções para armazenamento e expedição de produtos no armazém, sendo possível manusear diferentes produtos por depósito e inserir e expedir vários produtos de uma só vez.

## Trabalho Prático de Estruturas de Informação

**1º Semestre****Ano Letivo 2014/2015****1ª Parte****Cotação: 40%**

Aproveitando as facilidades oferecidas pelo paradigma orientado a objetos, o trabalho deverá ser realizado de forma incremental — começando com as classes mais simples até se chegar ao sistema final. É importante testarem as diferentes classes à medida que as desenvolvem. Serão assim estabelecidos os seguintes objetivos intermédios a alcançar.

1. Apresente o diagrama de classes
2. Desenvolva as classes necessárias ao simulador de armazém acima especificado.
3. Para cada classe, para além dos construtores, destrutor, métodos para atualização/acesso aos membros de dados privados e sobrecarga dos operadores necessários ao bom funcionamento da classe, deverão também ser implementados os métodos específicos da classe.
4. Para os métodos específicos das classes devem apresentar o diagrama de sequências.

**Data Entrega até 26-Out-2014****2ª Parte****Cotação: 60%**

A distância entre os depósitos de armazenamento é muito importante na gestão de um armazém, pelo que pretende-se que implementem o armazém, gerado e guardado em ficheiro de texto com o sistema anteriormente desenvolvido, através de um grafo usando a classe `graphSt1` e na qual a gestão do grafo ocorra indistintamente ao nível dos depósitos frescos e dos depósitos normais.

A classe para além dos construtores e destrutor deverá disponibilizar funcionalidades para:

- Construir o grafo
- Apresentar todos os percursos possíveis entre dois depósitos
- Apresentar um percurso entre dois depósitos que envolva apenas um tipo de depósito (fresco/normal)
- Calcular o percurso mais curto entre dois depósitos

**Nota:** Para os percursos devem ser apresentados os nós que o constituem e as vias de ligação usadas.

A aplicação deve possuir uma interface simples que permita invocar todas as funcionalidades pedidas.

**Data Entrega até 8-Dez-2014**

## Trabalho Prático de Estruturas de Informação

**1º Semestre**

**Ano Letivo 2014/2015**

### Regras

- É obrigatório o trabalho ser realizado em **grupos de dois alunos**.
- Deve ser usada a ferramenta GIT de controlo de versões.
- O relatório do projeto deverá ser elaborado segundo o template de relatório já usado em LAPR2 e devem constar os elementos pedidos ao longo do enunciado: diagrama de classes, diagrama de sequências dos métodos específicos das classes e para estes devem também fazer a análise de complexidade temporal.
- A entrega da 1ª Parte (até **26-Out-2013**) envolve submeter o projeto em código C++ e o relatório.
- A avaliação da 1ª Parte do Trabalho Prático será feita principalmente em função da conformidade da solução proposta com o Paradigma da POO.
- O projeto entregue na 1ª Parte poderá ser corrigido na 2ª Parte de acordo com os comentários/sugestões do professor.
- A entrega da 2ª Parte (até **8-Dez-2014**) envolve a submissão do projeto final e do relatório.
- O projeto será avaliado de acordo com as funcionalidades implementadas **modularidade, organização, clareza e eficiência do código**.
- Cada parte do trabalho deverá ser submetida no moodle, na área da disciplina, até às **24 horas do dia indicado**. A partir das datas indicadas, a nota do trabalho será penalizada **10% por cada dia de atraso** e não se aceitam trabalhos após **dois dias** das datas indicadas.
- O projeto criado em código C++ e o relatório devem ser colocados num ficheiro ZIP, com a designação: **SIGLAPROFAULAPL-NºAluno-NºAluno.zip** por exemplo, **MFC-1101350-1101460.zip**
- A apresentação/avaliação do trabalho final será individual, em datas a fixar com o professor das aulas PL, na semana de **9-13 Dezembro**.