

## Projeto Final (Introdução)

---

### Composição dos grupos:

O trabalho deve ser realizado em grupos de **no máximo 2 elementos**. É permitido que os grupos sejam formados por alunos de turmas diferentes. No caso de terem professores diferentes terão que optar por um deles para a defesa e avaliação do trabalho.

### Enunciado do trabalho:

São propostos vários níveis de dificuldade (I, II e III, para 1, 2 e 3 valores, respetivamente), cabendo ao aluno, ou grupo de alunos, escolher aquele que pretende implementar.

### Cotação do trabalho:

O trabalho final é cotado para um máximo de 3 valores. Contudo, é dada uma bonificação de até 0.5 valores aos trabalhos que implementem funcionalidades extra (ver enunciado).

### Material a entregar:

Cada grupo deve entregar um ficheiro compactado (.zip ou .rar) contendo o seguinte:

- O código fonte do programa (executável e código fonte em C++);
- Um ficheiro com um pequeno relatório sobre o trabalho (cerca de uma página A4), com uma descrição da solução implementada, que apresente a justificação das opções em termos de estruturas de dados e algoritmos utilizados.

### Prazo e local de entrega:

Cada grupo de alunos pode optar por entregar o trabalho na época normal ou na época de recurso, como descrito no InforEstudante.

### Defesa do trabalho:

O trabalho terá de ser defendido oralmente por ambos os elementos do grupo. **A não comparência no ato da defesa do trabalho implica a anulação do mesmo.** As datas das defesas serão definidas no InforEstudante, após a entrega do trabalho (depois das aulas terminarem, em período de exames).

### Fraudes:

**Qualquer tentativa de fraude, como por exemplo, a utilização dissimulada de código fonte disponível na Internet, ou a utilização de terceiros na realização do projeto, será punida com a anulação do projeto!** Aceita-se que os alunos troquem impressões/sugestões entre si, desde que qualquer colaboração seja explicitamente mencionada no relatório do trabalho ou no início da apresentação oral. A utilização de código obtido na Internet deve também ser mencionada.

### Avaliação:

Na avaliação deste trabalho, será dada ênfase aos algoritmos utilizados para resolver os diversos problemas bem como às estruturas de dados adotadas para manipular informação no programa. Será valorizada uma implementação orientada a objetos.

---

# MINI-PROJETO

## Programa de Gestão do Inventário de uma Empresa

---

### Descrição Sumária:

Uma empresa pretende que se desenvolva um *software* para gerir informação minuciosa acerca do seu inventário. O inventário é uma enumeração dos bens que pertencem ou pertenceram à empresa.

### Descrição Detalhada:

O programa manipula várias listas:

1. Lista de tipologias de bens

Diferentes tipologias de bens que podem ser inventariados

- código (código numérico da tipologia constituído por 3 dígitos)
- designação curta (*string* contendo até 20 carateres, ex. “mobiliário”)
- descrição (*string* contendo até 255 carateres, ex. “mesa redonda de escritório”)

2. Lista de espaços

Espaços onde se localizam bens pertencentes ao inventário

- código espaço (código numérico de espaço constituído por 4 dígitos)
- edifício (*string* contendo até 50 carateres, ex. “Escritórios da administração”)
- andar
- designação (*string* contendo até 50 carateres, ex. “Sala de reuniões”)
- área (m<sup>2</sup>)

3. Lista de pessoas responsáveis

Pessoas responsáveis por bens pertencentes ao inventário

- número mecanográfico (código numérico constituído por 5 dígitos)
- nome
- cargo (ex. “chefe de serviço”)
- local habitual de trabalho (código espaço)
- telefone

4. Lista de bens

Informação sobre bens pertencentes ao inventário

- código (código numérico do item inventariado constituído por 10 dígitos)
- tipologia (código numérico da tipologia constituído por 3 dígitos)
- descrição (*string* contendo até 255 carateres)
- Data de incorporação
- Espaço onde se encontra (código espaço)
- Valor do bem em €
- número mecanográfico do responsável (código numérico constituído por 5 dígitos)
- Data de abate (se já não pertencer à empresa; indicado por uma *flag*)

O programa deve apresentar um menu com as seguintes opções:

1. Edita tipos de bens: permite inserir, eliminar e listar tipologias. Apenas é possível eliminar tipologias que nunca tenham sido associadas a bens do inventário.

2. Edita Lista de Espaços: permite inserir, eliminar e listar espaços. Apenas é possível eliminar espaços que não estejam associados a bens do inventário ou pessoas.
3. Edita Lista de Pessoas: permite inserir, eliminar e listar pessoas. Apenas é possível eliminar o registo de pessoas que não estejam associadas a bens do inventário.
4. Seleciona Espaço: o programa lista os espaços existentes e o utilizador seleciona um deles para as operações realizadas através da opção 5.
5. Gerir Inventário: o programa lista todo o inventário do espaço selecionado e depois permite inserir ou abater bens no inventário desse espaço.
6. Listar inventário: o programa lista todo o inventário, ordenado por espaços.
7. Alienar espaço [nível II]: permite alienar um espaço, o que implica abater no inventário todos os bens localizados nesse espaço. Pede a confirmação do utilizador.
8. Mudança de espaço [nível II]: o programa apresenta uma listagem dos bens existentes no espaço selecionado através da opção 4 e permite ao utilizador selecionar um ou mais bens desta lista para serem mudados para outro espaço também selecionado pelo utilizador. Opcionalmente, o utilizador pode modificar o responsável pelos bens mudados para outro espaço.
9. Outras consultas ao inventário [nível III]: o programa permite ao utilizador: i) listar o inventário de uma tipologia específica; ii) listar o inventário de um espaço; iii) listar o inventário dos bens à responsabilidade de uma determinada pessoa. Em qualquer dos casos, o programa deve calcular o valor do total dos bens, apresentando os subtotais para bens ativos e para os já abatidos no inventário.
10. Determinação de espaços críticos [nível III]: o programa determina o valor dos bens inventariados (ativos) em todos os espaços, para apresentar uma listagem do valor existente nos espaços, ordenados por ordem decrescente. A listagem permite assim identificar os espaços mais críticos em termos de segurança anti-roubo.
11. Sair. Pede a confirmação do utilizador.

### **Especificações para nível II (2 valores)**

O programa faz uso de variáveis dinâmicas, com alocação e libertação de memória dinâmica.

### **Especificações para nível III (3 valores)**

É guardada de forma persistente (em ficheiros) toda a informação manipulada pelo programa durante a sua execução, não podendo haver perda de dados. Sempre que o programa seja executado novamente, é recuperada toda a informação a partir dos ficheiros.