

## TRABALHO PRÁTICO – DESIGN PATTERNS

### Objetivos:

- Elaborar um programa que cumpra os pré-requisitos enumerados no enunciado.
- Comentários no código-fonte são encorajados e alvo de avaliação.
- Otimização é alvo de avaliação.
- É obrigatório o programa executar (sem erros de compilação).
- Deve ser usada linguagem Java para a resolução do presente projeto prático.
- Deve ser entregue na tarefa do Microsoft Teams. Pasta do Projeto comprimida (.zip) com o nome TP\_DPT\_(nome\_formando)

## EXERCÍCIOS

Após terminar o curso de **Software Developer**, foi contactado pelo **Parque Temático CESAEland ©** para desenvolver uma aplicação que permita carregar a sua base de dados de atrações e vendas, e efetuar uma série de análises. Para tal, desenvolva uma aplicação em Java que implemente o padrão **MVC**. Este parque tem uma série de atrações temáticas para as quais vende bilhete de adulto ou criança, variando o preço. Os registos das vendas são armazenados num ficheiro com a data da venda, contendo mês e ano, para posterior análise.

Deve conter um menu de login, que permita efetuar login como Administrador, Engenheiro de Manutenções ou Cliente.

O menu de **Cliente** pode ser acedido sem login e tem as seguintes funcionalidades:

- Consultar Atrações Disponíveis: Deve imprimir em forma de tabela, as atrações, preços dos bilhetes e duração da atração no formato min:seg.
- Consultar Atrações Favoritas: Deve imprimir a atração mais procurada por adultos e a atração mais procurada por crianças (que vendeu mais bilhetes no total, para o respetivo segmento de clientes).

O menu de **Engenheiro de Manutenção** deve ser acedido com login, validado com as contas permitidas guardadas num ficheiro. Sabemos que todas as atrações, por cada 50 bilhetes vendidos, devem ter uma revisão de segurança. E deve conter as seguintes funcionalidades:

- Consultar Próximas Revisões: Consulta as 3 próximas revisões a serem efetuadas e o número de bilhetes que faltam vender para a revisão ter de ser efetuada. Deve identificar o ID da Atração, Nome de Atração e Número de Bilhetes que Faltam.
- Consultar Histórico de Revisões: Consulta as 3 últimas revisões que foram efetuadas. Deve identificar o ID da Atração, Nome de Atração e Número de Bilhetes vendidos depois da última revisão.

O menu de **Administrador** deve ser acedido com login, validado com as contas permitidas guardadas num ficheiro. E deve conter as seguintes funcionalidades:

- Consultar o total de todas as vendas.
- Consultar o total de lucro.
- Consultar o total de vendas e lucro por mês, de forma tabelar.
- Consultar a atração mais procurada por adultos (número de bilhetes vendidos).
- Consultar a atração mais procurada por crianças (número de bilhetes vendidos).
- Consultar a atração mais procurada (número de bilhetes vendidos).
- Consultar a atração mais lucrativa (considere o período total).
- Consultar a atração menos lucrativa (considere o período total).
- Consultar a atração com melhor preço/tempo. Apresenta a atração que custa menos por segundo.
- Adicionar novo login: Permite adicionar um novo acesso válido, perguntando o tipo de acesso a criar. Deve validar se o nome do utilizador não é duplicado.

Os ficheiros devem ser lidos através de uma ferramenta, para um repositório.

Para tal, considere os seguintes ficheiros existentes e os seus cabeçalhos.

#### Cesaeland\_atracoes

- id: ID da Atração
- atracao: Designação da Atração
- precoAdulto: Preço do Bilhete para Adultos
- precoCrianca: Preço do Bilhete para Crianças
- duracaoSegundos: Duração da Atração, em segundos

#### Cesaeland\_vendas

- atracao: ID da Atração
- data: Mês e Ano da Venda do Bilhete
- tipoCliente: Tipo de Bilhete Vendido (adulto ou criança)

#### Cesaeland\_custos

- atracao: ID da Atração
- custoManutencaoBilhete: Custo de Manutenção por cada bilhete vendido
- custoFixoMes: Custo de Manutenção fixo por mês

#### Cesaeland\_logins

- Tipo de Acesso: Tipo de Acesso do Login (ADMIN para Administrador ou ENG para Engenheiro de Manutenções)
- Username: username de acesso do respetivo login
- Password: password de acesso do respetivo login

Desenvolva a estrutura de classes necessária para permitir o desenvolvimento do projeto em questão, assim como todas as assinaturas dos métodos. Deve implementar um login válido, tendo em conta o ficheiro de logins, assim como o acesso que cada conta deve ter.

Seguidamente, implemente uma funcionalidade para cada tipo de utilizador de sistema (excetuando as funcionalidades sublinhadas).

**Bom trabalho! ☺**