

# Ficha Prática #07

## 7.1 Objectivos

1. Praticar a modelação com **Diagramas de Classe** e **Diagramas de Sequência**.
2. Praticar o refinamento da arquitectura para prever a ligação à Camada de Dados.
3. Relacionar estes diagramas com a implementação que eles representam.

## 7.2 Exercícios

Para os exercícios abaixo propostos analise os enunciados e responda às questões criando os respectivos diagramas.

### 7.2.1 Parques de Estacionamento

Relembre o Exercício 6.2.2 da Ficha #06. Tendo como referência o Diagrama de Classes que desenvolveu, responda às seguintes questões:

1. Utilize as capacidades de geração de código do Visual Paradigm para gerar código Java a partir do Diagrama de classes.
2. Utilizando o código da alínea 1. com ponto de partida, escreva o código Java correspondente aos diagramas da alínea 2.
3. Identifique as entidades a persistir e que tabelas seriam necessárias para as mapear numa Base de dados relacional.
4. Refaça o Diagrama de Classes utilizado, por forma a considerar a implementação de persistência.

5. Refaça as alíneas 3. a 6. (do Exercício 6.2.2 da Ficha #06).
6. Desenvolva o **Diagrama de Sequência** para uma operação que dado o id de uma viatura, de um parque e duas datas, insere um novo registo no sistema.

### 7.2.2 Mediateca

Considere que lhe é pedido o desenvolvimento de um Sistema de Informação para uma Mediateca Digital (uma biblioteca multimédia). O sistema deverá permitir registar informação relativa aos vários tipos de documentos. Todos os documentos deverão possuir um número de registo que os identifica, bem como um nome, uma descrição, uma classificação etária (>0, >6, >12, >16, >18) e o endereço do ficheiro que contém o documento.

O sistema deverá ainda permitir acesso aos documentos (para tal todos os documentos deverão possuir um método play que permita a sua visualização/audição). Nesta fase vão apenas ser considerados três tipos de documento (mp3 para documentos áudio; png para documentos visuais; mpeg para documentos audiovisuais). A solução proposta deverá facilitar a adição de novos tipos de documento em fase posterior.

Para permitir referências cruzadas entre documentos, cada documento deverá possuir uma lista de referências para outros documentos no sistema (para ter a possibilidade de registar documentos associados ou semelhantes).

O sistema deverá contemplar dois tipos de utilizadores: utentes e administradores. Todos os utilizadores possuem um nome de utilizador e uma senha de acesso. Para cada utente deverá ser registado o seu nome e morada, bem como a sua data de nascimento.

Os utentes utilizam o sistema para procurar e consultar documentos. A procura é efectuada indicando uma frase chave, a consulta é efectuada indicando o número do registo do documento (esse número poderá ser obtido através de uma procura). No entanto, o acesso a um documento só é permitido se a classificação etária do documento não for incompatível com a idade do utente.

Os utentes podem ainda propor novos documentos para inclusão na Mediateca. Nesse caso a proposta é registada e fica pendente de aprovação por um administrador.

Os administradores fazem a gestão da informação no sistema, pelo que podem adicionar e remover utentes, bem como adicionar e remover documentos. É ainda incumbência dos administradores aceitar (ou não) os documentos propostos pelos utentes. Caso o documento seja aceite, o administrador deverá atribuir-lhe uma classificação etária.

Responda às seguintes questões (note que os diagramas pedidos são interdependentes):

1. Sabendo que cada Use Case identificável acima dará origem a um método na *Facade* da camada de negócio do sistema, proponha um **Diagrama de Classe**, o mais completo possível, para essa camada (inclua as variáveis e métodos que considerar mais relevantes).
2. Proponha um **Diagrama de Sequência** para o método que devolve uma lista com os números de registo de todos os documentos que satisfazem uma dada frase chave: `public List<String> procura(String chave)`
3. Uma das possibilidades para representar os vários tipos de documento no Diagrama de classes é ter uma classe Documento com sub-classes para os vários tipos de documento, outra é ter apenas a classe Documento e incluir nela uma variável de instância que indica o tipo do documento. Justifique a escolha que fez na resolução da questão anterior.
4. Utilize o Visual paradigm para gerar código Java a partir do Diagrama de Classes desenvolvido.
5. Escreva o código relativo ao Diagrama de Sequência da alínea 2.
6. Modifique o Diagrama de Classe anteriormente desenvolvido para prever a ligação com a camada de dados.
7. Refaça as alíneas 4. e 5.