

— Teste Teórico —

## Arquitecturas Aplicacionais

MEI - EA - AA

04/06/2019

Duração máxima: 1h30

### Grupo I / Parte Teórica

1. Um dos patterns mais conhecidos e utilizados é o *Abstract Factory*. Explique por palavras suas quais as situações em que faz sentido utilizá-lo e quais são os seus principais componentes.
2. Dê um exemplo de uma aplicação em que poderá ser útil a utilização do *Abstract Factory*. Apresente uma descrição da situação, o diagrama de classes e código (ou pseudo-código) que o permite operacionalizar.
3. Ao longo do semestre abordou-se a construção de aplicações complexas com recurso a uma lógica multi-camada (ou multi-tier). Refira quais lhe parecem ser as vantagens que encontra na construção de aplicações segundo este princípio. Seja sucinto e objectivo.
4. Considera que o pattern *MVC* é uma boa solução para promover a separação de camadas? Justifique a sua resposta, fornecendo, se possível, modelos arquitecturais que ajudem a perceber o sentido da resposta.

### Grupo II / Parte Prática

5. Considere que se pretende criar uma plataforma Web que ofereça uma solução de aluguer temporário de moradias (inspirado no AirBnB, HomeAway, etc.). Nessa plataforma os actores principais são os Proprietários, os Clientes e as Agências Imobiliárias. Os proprietários possuem um conjunto de imóveis que pretendem alugar, mas pretendem ter sempre a última palavra no processo de aluguer. Isto é, o proprietário deverá aprovar o pedido do cliente individual e no final deverá atribuir um rating ao aluguer (para garantir que consegue ter uma história dos alugueres efectuados). O cliente pesquisa por imóveis, quer através de localização quer através de pesquisa por imóveis de um proprietário específico.

Neste contexto existem também as agências imobiliárias, que agregam imóveis de diferentes proprietários e fazem o processo de promoção dos imóveis. Para os imóveis que estão no catálogo de uma agência não é necessário fazer o processo de aprovação via proprietário, na medida em que as agências já o acordaram com o cliente.

Para cada imóvel regista-se a sua designação, a sua tipologia, o preço por dia, o proprietário ou a agência em que está registado, bem como se mantém um histórico de anteriores alugueres.

A solução deverá permitir:

- (a) registar um novo proprietário
- (b) registar um novo cliente
- (c) registar uma nova agência imobiliária
- (d) registar um novo imóvel por parte de um proprietário (que o vai gerir autonomamente)
- (e) listar os imóveis por proprietário, por localização e por agência
- (f) listar os processos de aluguer de um imóvel (este use case está apenas disponível para os proprietários ou para as agências)
- (g) registar um pedido de aluguer por parte de um cliente
- (h) aprovar ou rejeitar, por parte de um proprietário, de um pedido de aluguer
- (i) consultar o top10 de clientes que mais alugam imóveis.

Responda às seguintes questões:

- (a) desenvolva o diagrama de classe, o PSM e eventualmente o PIM, da solução que propõe. Seja o mais descritivo possível por forma a identificar os diferentes papéis das diferentes classes.
- (b) desenvolva a servlet e beans necessários às funcionalidades que permitem a consulta dos imóveis de um proprietário e ao aluguer de imóvel por parte de um cliente.
- (c) como faria o deployment da solução? Que arquitectura utilizaria, que software e como lidaria com o aumento da carga aplicacional.