

## 6 Lab: Modelo do domínio, parte 2

### Tópicos

O modelo do domínio é uma representação dos objetos (conceitos) da área do problema. Este **mapa visual** é construído utilizando o **diagrama de classes** da UML. No contexto do modelo do domínio, as classes representam entidades naturais no universo de discurso do problema e não classes de programação ou tabelas de uma base de dados (modelo do domínio  $\neq$  modelo de dados da BD). O modelo do domínio mostra os conceitos, os seus atributos e as associações entre eles (normalmente, sem mostrar as operações).

### Tarefas

#### E6.1

Considere novamente o contexto do exercício 4.5, referente a uma solução de gestão de projetos. Retome, se necessário, os passos sugeridos para a exploração da ferramenta (Redmine).

A título exploratório, experimente (pelo menos) os seguintes passos, no contexto de uma equipa:

- Criar um projeto.
- Configurar os módulos que se pretende utilizar no projeto; incluir o módulo Gantt. Quanto aos Trackers, pode-se aceitar a predefinição.
- Configurar a equipa, adicionando os respetivos membros (Settings > Members). Note que os membros podem ter papéis diferenciados.
- Adicionar uma nova tarefa ao projeto (Issue X), relativa, por exemplo, "Protótipo da página de pesquisa de filmes por género".
- Configure a tarefa (Issue X) definindo, pelo menos, a descrição, prioridade, data de início e de finalização. Atribua a tarefa a um responsável (Assignee) e envolva mais pessoas no acompanhamento do progresso (Watchers).
- Verifique no cronograma (Gantt) o posicionamento da tarefa.
- Adicione agora uma segunda tarefa (Issue Y), como fez para a anterior, fazendo variar as características (prioridade, datas, assignee, watchers, etc).
- Volte à listagem de Issues e aceda ao detalhe do Issue X (o primeiro). Mude o estado para "Em curso"; atualize reportando trabalho feito: preencha a secção de Log time, descrição e anexe um screenshot (por hipótese, relacionados com a tarefa). Atualize também a % Done.
- Volte ao cronograma e verifique as alterações. Experimente filtrar a informação no cronograma para um responsável específico (Assignee).
- Experimente livremente alterar o estado das tarefas, reportar trabalho, alterar a duração, etc.

## E6.2

Identifique os principais substantivos na descrição anterior. A partir daí, anote numa tabela os conceitos candidatos a serem incluídos no modelo do domínio.

Nota: nem todos os substantivos revelados na análise textual serão relevantes. Para além disso, a linguagem natural terá repetições e ambiguidades, que é preciso filtrar.

Conceito candidato	Atributos candidatos
...	...

## E6.3

Utilizando a informação que obteve nas alíneas anteriores, crie um modelo do domínio da gestão de projetos.

O seu modelo de domínio deve ter a capacidade expressiva para permitir memorizar a **informação suficiente para suportar os resultados vistos** no Redmine (lista geral de *issues* e o seu estado, ficha com os detalhes do *Issue*, atribuição de tarefas a membros da equipa, Gantt, etc.).

Procure criar um modelo **completo**, mas limitado ao âmbito que foi experimentado.

Nota: o processo normal, será construir o modelo do domínio antes de haver sistema. Neste caso, estamos a fazer ao contrário (abstraindo a partir de um sistema real).

## E6.4

A partir do resultado da alínea anterior, introduza as alterações necessárias para considerar ainda os seguintes requisitos:

- Um membro pode fazer parte da equipa do projeto com papéis diferentes, ao longo do tempo (e pretende-se memorizar quais).
- Um colaborador pode observar o progresso de uma tarefa (*Watcher*) durante períodos de tempo discretos, e não necessariamente durante toda a tarefa (e pretende-se memorizar quais).

## E6.5 [Opcional]

A ferramenta VisualParadigm (edição Profissional) permite fazer análises textuais para identificar elementos de modelação a partir de descrições de requisitos.

Aplique a técnica de *textual analysis* neste problema, seguindo a informação tutorial disponível (em Help > UML Modeling > Textual Analysis, ou, alternativamente, [na Web](#)).

Edite o *problem statement*, marque no texto classes candidatas, e visualize-as num Diagrama de classes. (Para *problem statement*, pode adaptar o texto do exercício 1.)

## E6.6 [Para entrega]

Modele a informação que se pode depreender da seguinte descrição de um sistema de gestão de reservas de passagens aéreas.

- As companhias aéreas oferecem vários vôos, com um código associado (e.g.: TP701)
- Um vôo pode estar aberto para receber reservas, ou fechado, por ordem da companhia que o opera.

3. Um cliente pode reservar um ou mais vôos e para passageiros diferentes.
4. Cada reserva diz respeito a um único voo e a um único passageiro.
5. Depois de ter sido criada, uma reserva pode ser cancelada ou confirmada.
6. Um voo tem um aeroporto de partida e um aeroporto de chegada.
7. Um voo tem um dia e hora de partida, e um dia e hora de chegada.
8. Um voo pode envolver escalas intermédias em vários aeroportos.
9. Uma escala tem uma hora de chegada e uma hora de partida.
10. Cada aeroporto serve uma ou mais cidades.

A partir desta base, construa um modelo (estático) do domínio. Adicione aos conceitos identificados os atributos que considere serem essenciais neste domínio (mesmo que não tenham sido explicitados nas frases acima).