

3 Lab: Modelação do domínio com classes

Enquadramento

Objetivos de aprendizagem



- Identificar conceitos/classes na descrição de um problema.
- Caracterizar as estruturas de dados de um problema como classes e associações.
- Construir e interpretar diagramas de classes (perspetiva do analista).
- Utilizar associações “simples”, agregações, composições e generalizações.

Preparação

- Informação tutorial: [“What is Class Diagrams?”](#)

Entrega

Cada grupo deve designar um “pivot” para o lab, que se encarrega de recolher os contributos dos colegas e fazer a entrega. Este papel é rotativo.

O exercício tem atividades para serem realizadas em **duas aulas** práticas. O grupo deve submeter uma entrega, com as respostas aos exercícios assinalados com  

A entrega é um breve relatório, identificando o lab e os autores, e destacando (sublinhado) o aluno que foi o pivot.

Exercício

E3.1

Pesquise os seguintes livros no [catálogo da Biblioteca da UA](#):

- “UML Distilled”, de Martin Fowler.
- “Use case driven object modeling with UML”, de D. Rosemberg.

Tendo presente os resultados da pesquisa, dê exemplos de “classes” e “objetos”.

E3.2

Crie um diagrama UML para modelar o domínio relativo ao funcionamento da biblioteca, incluindo o resultado do ponto anterior e o seguinte conhecimento da área do problema:

- a) os utilizadores da Biblioteca pesquisam obras por autor, título, ano, ou uma combinação desses elementos.
- b) para cada obra, podem existir vários exemplares, com cota e código de barras únicos, que podem ser levantados pelos utilizadores, em regime de empréstimo.
- c) existem multas para devoluções tardias, mas nem todos os utilizadores têm o mesmo tempo para reter os livros em empréstimo domiciliário. Há que distinguir entre utilizadores que são alunos, professores ou utilizadores externos. O tempo de empréstimo normal é de 15, 90 e 30 dias respetivamente. Todos os utilizadores têm um número mecanográfico alfanumérico.
- d) Para inscrever um utilizador externo, é necessário confirmar a sua identidade (contra a apresentação do cartão de cidadão) e a morada (com a apresentação de uma fatura, titulada ao utilizador, do fornecimento de eletricidade, água ou serviço similar).
- e) Os utilizadores podem também pedir a reserva de obras para utilização numa data futura

(sendo atribuído o primeiro exemplar disponível).

- f) Existem vários polos (e.g.: Biblioteca Campus Santiago, Mediateca, Biblioteca ESTGA,...) nos quais se encontram os exemplares. Cada polo tem o seu próprio horário de funcionamento.

E3.3

Considere a área da gestão de projetos, por exemplo, relativamente ao seguimento de projetos de desenvolvimento de software. Para concretizar, pode experimentar as funcionalidades da ferramenta de gestão de projetos Redmine, no [respetivo site](#), ou, melhor, no [code.ua.pt](#) (que é baseado no 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.

A partir do texto anterior (em caixa), identifique os principais substantivos e a partir daí, anote numa tabela os **conceitos e atributos candidatos** a serem incluídos no modelo do domínio.

Conceito candidato	Atributos candidatos
Projeto	Título, data de início [...]
...	...

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.

E3.4 [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, da Biblioteca.)

E3.5 Análise de categorias

Uma outra abordagem para fazer o levantamento dos conceitos de um domínio é utilizar uma lista de categorias para procurar conceitos (classes).

Utilizando a chave incluída no material suplementar (Tabela 1), procure identificar conceitos do domínio da gestão de projetos, para cada uma das categorias. Nota: poderá haver mais que um conceito em cada categoria, bem como poderá não existir nenhum.

Categoria conceptual	Conceito identificado (domínio da gestão de projetos)
e.g.: Lugares físicos	e.g.: Sala
...	...

E3.6

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 todos 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**, para o 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).

E3.7 (em conjunto com a alínea anterior)

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).

Material suplementar

Tabela 1: Algumas categorias em que podemos procurar conceitos de um domínio [Larman]

Conceptual Class Category	Examples
business transactions <i>Guideline:</i> These are critical (they involve money), so start with transactions.	<i>Sale, Payment</i> <i>Reservation</i>
transaction line items <i>Guideline:</i> Transactions often come with related line items, so consider these next.	<i>SalesLineItem</i>

product or service related to a transaction or transaction line item <i>Guideline:</i> Transactions are <i>for</i> something (a product or service). Consider these next.	<i>Item</i> <i>Flight, Seat, Meal</i>
where is the transaction recorded? <i>Guideline:</i> Important.	<i>Register, Ledger</i> <i>FlightManifest</i>
roles of people or organizations related to the transaction; actors in the use case <i>Guideline:</i> We usually need to know about the parties involved in a transaction.	<i>Cashier, Customer, Store</i> <i>MonopolyPlayer Passenger, Airline</i>
place of transaction; place of service	<i>Store</i> <i>Airport, Plane, Seat</i>
noteworthy events, often with a time or place we need to remember	<i>Sale, Payment MonopolyGame Flight</i>
physical objects <i>Guideline:</i> This is especially relevant when creating device-control software, or simulations.	<i>Item, Register Board, Piece, Die</i> <i>Airplane</i>
descriptions of things <i>Guideline:</i> See p. 147 for discussion.	<i>ProductDescription</i> <i>FlightDescription</i>
catalogs <i>Guideline:</i> Descriptions are often in a catalog.	<i>ProductCatalog</i> <i>FlightCatalog</i>
containers of things (physical or information)	<i>Store, Bin Board Airplane</i>
things in a container	<i>Item Square (in a Board) Passenger</i>
other collaborating systems	<i>CreditAuthorizationSystem</i> <i>AirTrafficControl</i>
records of finance, work, contracts, legal matters	<i>Receipt, Ledger</i> <i>MaintenanceLog</i>
financial instruments	<i>Cash, Check, LineOfCredit</i> <i>TicketCredit</i>
schedules, manuals, documents that are regularly referred to in order to perform work	<i>DailyPriceChangeList</i> <i>RepairSchedule</i>