

3 Lab: Modelação do domínio com classes

3.1 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 Diagram?”](#)

Entrega

As atividades marcadas com  dão origem a entrega, para avaliação.

3.2 Parte A: análise por objetos de um domínio

3.2.1 Caso 1: estrutura de uma empresa



Considere a seguinte descrição de um problema:

“Trabalhamos com várias empresas: cada **empresa** tem sempre um NIF e está organizada em vários **departamentos** (e.g.: Marketing, Produção, Aprovisionamento, ...). Cada departamento tem um **conjunto de funcionários** atribuídos. Também existe, para cada departamento, um **gestor** atribuído. Conheço bem o Eng. Casimiro, empregado da HighTech SA, que é responsável do departamento de Produção!

Os funcionários têm um número mecanográfico, uma categoria profissional e um salário base. O Luís Pedro e a Zélia trabalham para o Eng. Casimiro; recebem 1800EUR.”

- A partir da informação disponível, represente, num diagrama (de classes):
 - Entidades sobre as quais é relevante representar informação e as suas características (atributos);
 - Associações entre esses conceitos, considerando a multiplicidade adequada.
- A partir do texto, dê exemplos de “classes” e “objetos”.

3.2.2 Caso 2: Conceitos associados ao funcionamento da Biblioteca



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

- “UML Distilled”, de Martin Fowler (3a edição).
- “Use case driven object modeling with UML”, de D. Rosenberg (2007).

Tendo presente a ficha com a descrição de cada livro:

- “As obras têm autores”.

Quais os conceitos (classes, tipos de “coisas”) associados a esta observação?
Quais as características essenciais (atributos de informação) que descrevem cada conceito?

Crie um diagrama UML para modelar os conceitos do domínio relativo ao funcionamento da biblioteca, incluindo o resultado das alíneas anteriores e o seguinte conhecimento da área

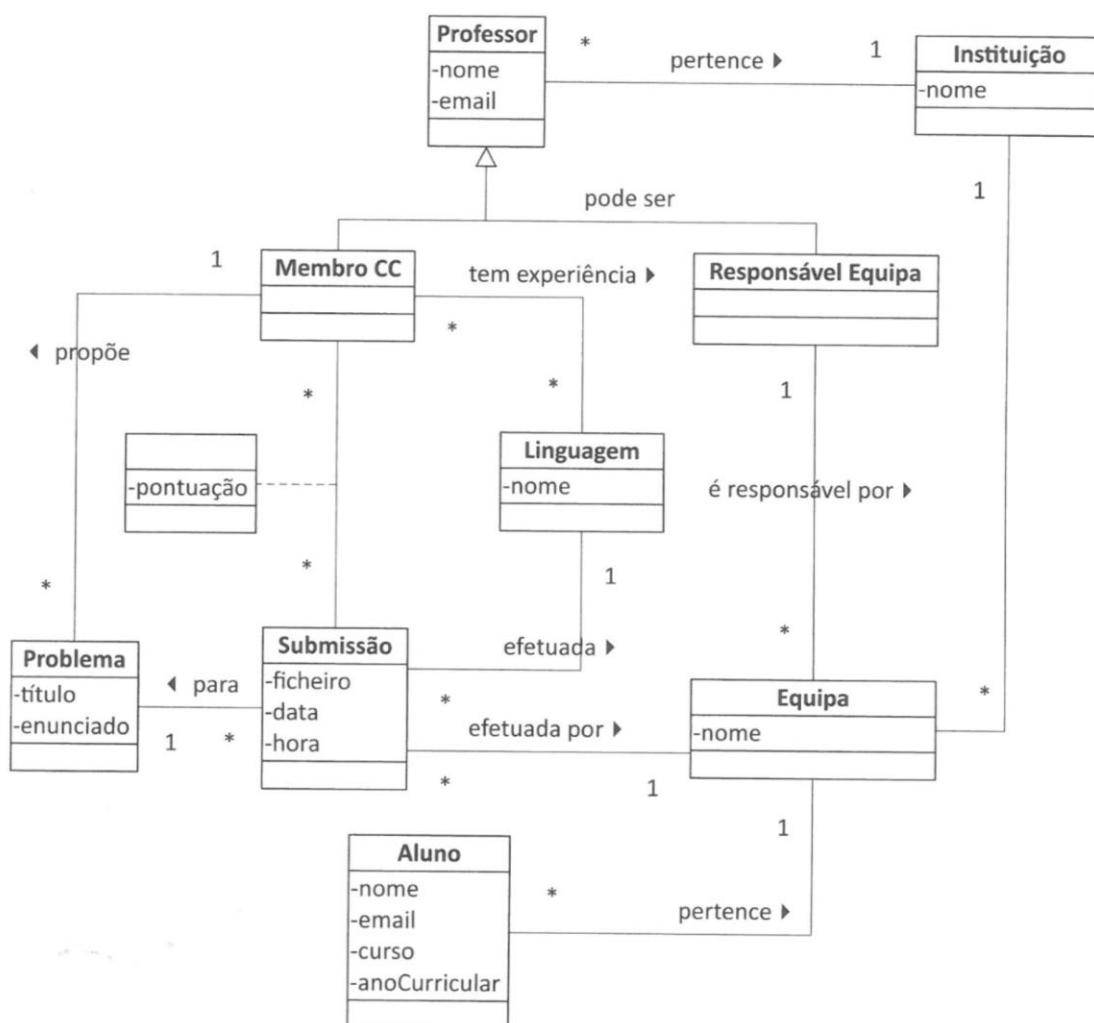
do problema:

- a) os utilizadores 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.

3.3 Parte B

3.3.1 Conceitos associados a um concurso de programação

- i) Faça uma leitura interpretativa do diagrama junto (explique, por palavras suas, que conceitos são representados e como se relacionam). E.g. “Cada Equipa tem um Professor responsável.”
- ii) Considerando o modelo representado no diagrama, explique se as seguintes afirmações têm ou não suporte no modelo, isto é, se são V/F face ao que está no diagrama.
 - a) Todas as Equipas precisam de indicar um Professor responsável.
 - b) Podem existir Professores que não coordenam nenhuma Equipa.
 - c) A Entrega (submissão) é feita por vários Alunos.
 - d) Uma Submissão é avaliada por um Membro do CC.
 - e) Uma Equipa poder ser composta por alunos de várias Instituições (i.e., a Equipa não é de uma Instituição).
 - f) Um Membro do CC só pode avaliar entregas resolvidas com linguagens de programação para as quais é especialista.
 - g) As Entregas de uma Equipa são sempre feitas pelo capitão da equipa.
 - h) As Entregas de uma Equipa relativa a um Desafio podem ser avaliadas por Docentes diferentes.



Modelo de classes relativo à organização de concursos de programação; CC: Conselho científico das provas. [In: Borges et al, "Modelação de Dados em UML: uma abordagem por problemas."]

3.3.2

Retome o exemplo considerado antes, na secção 3.2.1. Estenda esse modelo, considerando as necessidades de informação associadas à gestão de projetos de desenvolvimento de software:

- Os projetos têm uma duração prevista, com data de início e de fim bem definidas.
- O projeto tem um Project Leader atribuído (que é um funcionário) e um departamento responsável. O Eng. Casimiro está responsável pelo projeto "Stock-RFID+" dada a sua especialização em sistemas embebidos.
- Os funcionários devem fazer *check-in* e *check-out* para que se possa registar as suas horas de trabalho, ao longo dos dias.
- Cada projeto é organizado em várias tarefas (ou atividades), com uma duração prevista definida. As tarefas são sempre referentes a um projeto, no qual estão definidas.
- Um projeto vai ter uma equipa de funcionários atribuída, entre *developers*, especialistas de interação e especialistas de DevOps. No entanto, os funcionários podem estar associados a um projeto por um tempo inferior à sua duração, sendo importante saber o período em que cada um trabalhou nesse.
- Também é necessário saber quem esteve envolvido na realização de cada tarefa (uma tarefa pode envolver diferentes pessoas).
- Os projetos externos (ao contrário dos projetos internos), têm um cliente bem definido.
- Uma tarefa tem de ter uma descrição clara, para todos a entender de forma objetiva.

Para além disso, tem uma data de início, de fim, e uma duração esperada. Uma tarefa pode ser dividida em outras tarefas, que a compõe, se ajudar no planeamento e monitorização do projeto.

- i) No contexto de uma tarefa, há lugar a vários registos de intervenção, i.e., um *log* das atualizações/ações que cada funcionário fez no contexto daquela tarefa (e.g.: refinar modelo de casos de uso, versão inicial do UI,...). Em cada intervenção, o funcionário explica o que foi feito e o nr de horas gastas.
- j) As tarefas evoluem de acordo com estados bem definidos: “Em aberto”, “Em progresso”, “Concluída”, ou “Cancelada”.
- k) O Eng. Casimiro está a analisar o desempenho da equipa no projeto “Stock-RFID+”; concluiu que os tempos médios, para cada estado, foram: “Em aberto”, 2dias; “Em progresso”, 4.5 dias. 60% das tarefas encontram-se concluídas.

Construa um modelo para representar o mapa de informação que se depreende do texto.

Concretize tipos de dados para os atributos. Para além das associações, procure oportunidades para usar: enumerados, atributos derivados, operações (se relevantes para a fase de análise).

3.4 Parte C: modelo do domínio da encomenda *online* de comida

Este “atividade-projeto” deve dar origem a um relatório e ser submetido no Moodle.
[template disponível no Moodle]

3.4.1

Considere a área das encomendas de comida online.

Sugere-se, para o efeito, focar a análise num serviço concreto, possivelmente um que já lhe seja familiar.

Desenvolva um modelo do domínio para o caso de estudo que escolheu. O seu modelo do domínio deve ter **a capacidade expressiva suficiente para permitir captar/memorizar a informação necessária aos processos de encomenda e entrega de comida.**

Explore o seu caso de estudo e procure desenvolver um mapa completo da informação necessária. Veja, por exemplo, em um ou mais sites a informação envolvida. Para além de acrescentar o detalhe do caso concreto que escolheu, represente que:

- a) Os clientes pesquisam a oferta de menus/opções online e compõem o seu pedido.
- b) A oferta pode envolver diferentes restaurantes parceiros, que é possível pesquisar de forma integrada. (Embora um pedido concreto deva ser confeccionado por um único restaurante.)
- c) O pedido (encomenda) origina um pagamento e uma entrega que é assegurada por um estafeta.
- d) Os clientes podem seguir o progresso do seu pedido.
- e) Os menus oferecidos pelos restaurantes parceiros mudam; a própria lista de restaurantes parceiros muda.
- f) O preço dos menus pode mudar de acordo com promoções limitadas no tempo.

No modelo do domínio estamos à procura de “categorias de coisas”, ou conceitos, que são representadas como classes. Para cada categoria vamos encontrar alguns atributos (dados que devem ser memorizados no sistema de informação). Nesta fase, o analista não está preocupado em representar métodos/funções nas classes (como é próprio da programação).

Os conceitos estão relacionados entre si, formando uma rede de conceitos, ligados por associações. O Diagrama de Classes fornece os elementos de modelação para construir esse mapa, seguindo a técnica de análise por objetos.

3.4.2

Verifique se, no seu modelo, há situações em que seja oportuno mostrar a evolução do estado associado a um conceito, i.e., se existem entidades com um ciclo de vida associado. Modele, para essas entidades, a máquina de estados associada (com um Diagrama de estados)

No caso do problema dos cheques-dentista, abordado em outros exemplos, há um ciclo de vida associado: começa por ser Emitido; depois, entre Em Utilização, em várias consultas; depois, evolui ara Utilizado (na última consulta); depois, fica Pendente para pagamento (ao dentista), e a seguir Pago. Mas, depois de Emitido, também pode transitar para o estado Cancelado.

3.5 Exercícios complementares

[Estes exercícios destinam-se sobretudo a aprofundar a matéria e não são pedidos para submissão.]

i) **Análise textual para encontrar conceitos do domínio**

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.)