

# Modelação e Design

## 01: Apresentação

Leonor Melo  
leonor@isec.pt

1

### Sumário

- Apresentação
- Apresentação da FUC
- Noção de Modelo
- Introdução ao UML

01 Apresentação

2

2

## Docentes

- Leonor Melo
  - responsável pela Unidade Curricular
  - teóricas + algumas práticas diurno + práticas pós-laboral
  - leonor@isec.pt
- Ivo Gonçalves
  - algumas práticas diurno
  - ivo.goncalves@isec.pt

01 Apresentação

3

3

## Objetivos

- Estabelecer contacto com a linguagem de modelação UML
- Utilizar o UML como ferramenta para a modelação de sistemas
- Adquirir conhecimentos em metodologias orientadas a objectos
- Compreender a evolução de paradigmas de análise e construção de software
- Desenvolver capacidades para analisar e projectar sistemas complexos

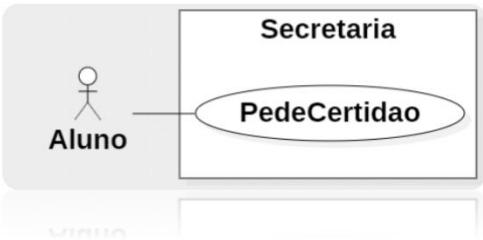
01 Apresentação

4

4

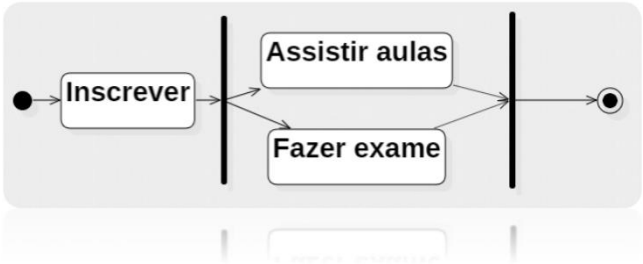
Programa - 1

- 1. Introdução ao UML
- 2. Casos de Uso
  - Serve para refletir e definir os requisitos que o sistema deve cumprir.
  - Que utilizadores o vão usar e que funcionalidades pretendem



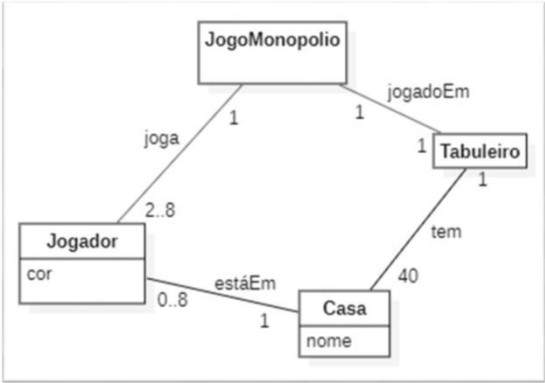
Programa - 2

- 3. Diagramas de Atividade
  - modelar os processos (processos de negócio ou processos de software). Mecanismo de controlo do fluxo de execução e do fluxo de dados



Programa - 3

- 3. Modelo do domínio
  - ferramenta de comunicação e análise do problema
  - dicionário visual do domínio do problema



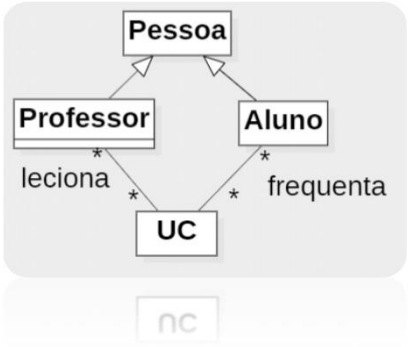
01 Apresentação

7

7

Programa - 4

- 4. Modelação Orientada a Objetos
- 5. Diagramas de Classes
  - teve origem no desenvolvimento orientado a objetos. Baseado nos conceitos de classe , generalização e associação



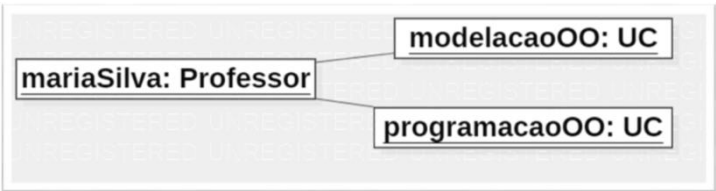
01 Apresentação

8

8

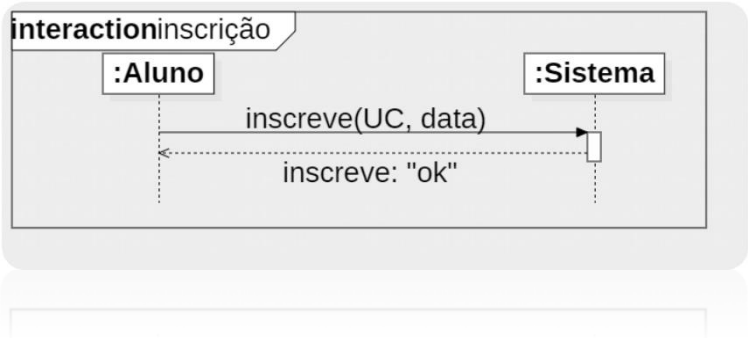
Programa - 5

6. Diagramas de Objetos
- retrato concreto do estado do sistema em determinado instante



Programa - 6

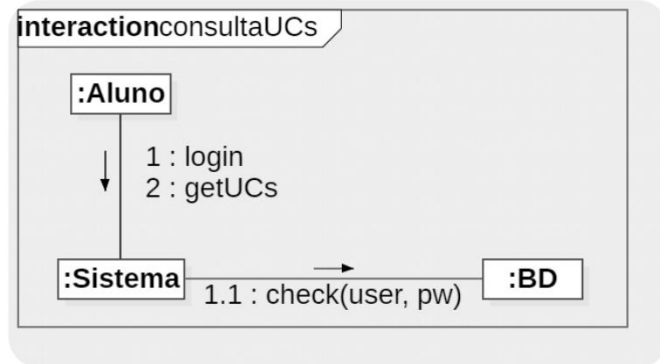
7. Diagramas de Sequencia
- Descreve as interações entre objetos de forma a realiza determinada tarefa.O foco é a ordem cronológica das mensagens trocadas



## Programa - 7

## 8. Diagramas de Comunicação

- Também descreve as interações entre objetos mas o foco é quem comunica com quem



01 Apresentação

11

11

## Programa - 8

## 9. Diagramas de Componentes

- Um componente é uma unidade de execução independente, que fornece serviços ou usa serviços de outros componentes. Podemos modelar duas perspectivas
  - blackbox: que representa a especificação do componente
  - whitebox: que representa a implementação do componente



01 Apresentação

12

12

Programa - 9

10. Diagramas de Instalação

- Representa a forma como o software é atribuído ao hardware (servidor, desktop,..) e ambientes de execução (sistema operativo, web server,...), e o modo como as partes comunicam



Programa - 10

11. Padrões de Software

12. Metodologias de desenvolvimento de software

## Bibliografia - 1

- UML distilled, Martin Fowler, 2004, Addison Wesley, 3ª edição
- UML: guia do usuário, Grady Booch, Ivar Jacobson, James Rumbaugh, 2006, Editora Campus
- Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development, Craig Larman, 2007, Prentice-Hall, 3ª edição
- Software Engineering, Ian Sommerville, 2011, Pearson, 9ª edição

01 Apresentação

15

15

## Bibliografia - 2

- UML, Metodologias e Ferramentas CASE, Alberto Silva, Carlos Videira, 2008
- Fundamental de UML, Mauro Nunes, Henrique O'Neil, 2004, FCA
- Slides das aulas teóricas

01 Apresentação

16

16



### Metodologias de ensino / aprendizagem

- Aulas teóricas
  - 2 aulas (de 1h) por semana
  - apresentação dos conceitos
  - ilustração dos conceitos através de exemplos
- Aulas práticas
  - 1 aulas (de 2h) por semana
  - aplicação dos conceitos através de exercícios guiados
  - eventual exposição de conceitos adicionais
- Desenvolvimento de um Trabalho Prático

01 Apresentação

17

17

### Avaliação

- Exame escrito - 10 valores
- Trabalho prático - 10 valores
  - fase 1: (25%)
  - fase 2: (25%)
  - fase 3: (50%)
  - defesa/feedback após cada fase entregue

01 Apresentação

18

18

### Alunos Erasmus

- Alunos Erasmus?
  - enviar mail para leonor@isec.pt

01 Apresentação

19

19

### Motivação

- Um modelo realça os aspetos importantes do software através de uma notação clara e tão simples quanto possível, e omite detalhes irrelevantes
- Diferentes tipos de modelos são usados para comunicar diferentes aspetos do que queremos construir.
  - Cada modelo apresenta apenas parte da informação e deve ser facilmente interpretada por quem o vai ler
- Os vários modelos devem ser consistentes entre si

01 Apresentação

20

20

## Sistemas

- Os modelos permitem descrever sistemas de forma eficiente
- Um sistema é algo constituído por componentes que estão relacionadas entre si de tal forma que podem ser percecionadas - do ponto de vista de determinada tarefa ou propósito - como uma unidade, e assim serem distintas do ambiente que as rodeia:
  - exemplo: um carro, um avião, um lago, uma floresta, um sistema informático...

01 Apresentação

21

21

## Propriedades dos modelos - 1

- Um modelo é uma representação de um sistema reduzida ao essencial
  - aspetos específicos e individuais são removidos deixando apenas as facetas fundamentais e caraterizadoras do sistema
- Normalmente o sistema é descrito não apenas por um modelo mas por um conjunto de "vistas" que em conjunto produzem uma imagem coerente e completa

01 Apresentação

22

22

## Propriedades dos modelos - 2

- Um modelo deve:
  - Omitir detalhes irrelevantes
  - Usar uma notação simples e intuitiva
  - Ser rigoroso
  - Ajudar a prever características não triviais do sistema
  - Ser mais simples de criar que o sistema

01 Apresentação

23

23

## Modelos podem ser usados para - 1

- Modelos como rascunho:
  - usados para comunicar de forma simples certas ideias ou aspetos parciais do sistema. Usados para clarificar raciocínios ou simplificar a discussão com outros membros da equipa
- Modelos como "plantas":
  - usados para sistematizar a informação e estabelecer a base a partir da qual se podem começar a desenvolver e implementar. Normalmente parciais mas com detalhe suficiente para permitir simulações, implementação, ou exploração de uma das suas sub-partes

01 Apresentação

24

24

### Modelos podem ser usados para - 2

- Modelos para desenvolvimento automático de software:
  - Modelos especificados de forma tão precisa que permitem a construção automática de código a partir deles. Uma área ainda em investigação com os seus defensores e oponentes

01 Apresentação

25

25

### UML - 1

- Unified Modeling Language
  - linguagem de modelação muito usada para a modelação de conceitos orientados a objetos (e não só)
- Permite representar vários aspetos de um sistema de sw
  - requisitos
  - estruturas de dados
  - fluxos de informação
  - fluxo de dados

01 Apresentação

26

26

## UML - 2

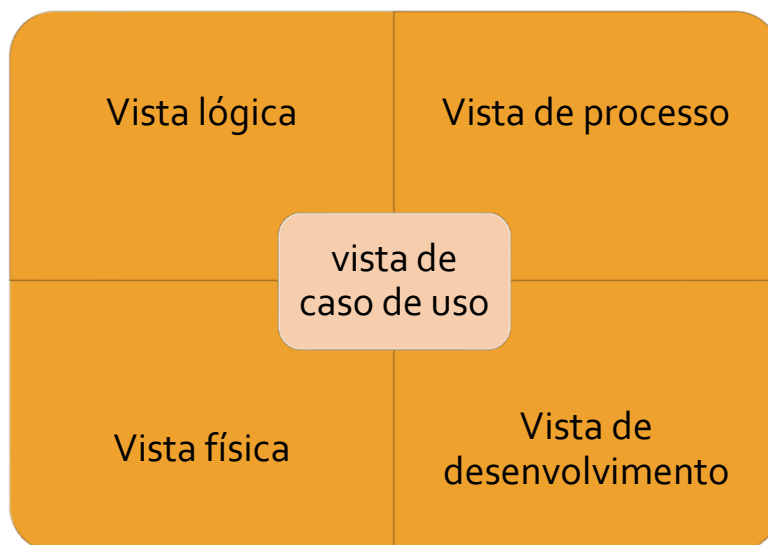
- Não está ligada a nenhuma linguagem de programação nem metodologia de desenvolvimento de software
  - mas é particularmente útil para abordagens iterativas e incrementais

01 Apresentação

27

27

## Perspetivas do modelo



01 Apresentação

28

28

### Perspetiva de caso de uso

- Perspetiva de caso de uso
  - Descreve as funcionalidades do sistema do ponto de vista do mundo exterior.
  - É necessária para descrever o que o sistema deve fazer.
  - Todas as outras perspetivas dependem desta
  - Inclui entre outros Diagramas de Caso de Uso e Especificação de Caso de Uso

01 Apresentação

29

29

### Perspetiva de processo

- Descreve os processos dentro do sistema
  - Particularmente útil para visualizar o que deve acontecer no sistema
  - Inclui Diagramas de Atividade

01 Apresentação

30

30

### Perspetiva lógica

- Descreve de forma abstrata as várias partes do sistema
  - Usada para modelar as partes que constituem o sistema e como é que elas interagem
  - Inclui os diagramas de classe, de objeto, de máquinas de estado e diagramas de interação

01 Apresentação

31

31

### Perspetiva de desenvolvimen to

- Descreve como as parte do sistema estão organizadas em módulos e componentes
  - Util para gerir as camadas da arquitetura do sistema
  - Esta vista inclui diagramas de componentes

01 Apresentação

32

32



Perspetiva física

- Descreve como o design do sistema, descrito nas perspetivas anteriores, se deve mapear num sistema real
  - Inclui os diagramas de instalação

Diagramas UML

