

## Aula 4

### Análise de Sistemas

Profª Adriana Bastos da Costa

1

### Conversa Inicial

2

### Análise de sistemas

- Projetar e construir um programa de computador não é uma tarefa trivial, pois envolve conhecer bem o problema que se quer resolver e traduzir a solução ideal para o problema em linhas de código, portanto, entender bem os requisitos ou o que o *software* precisa fazer para atender às necessidades do cliente é base de tudo
- Se os requisitos não forem bem analisados, fica mais difícil projetar uma solução adequada e eficiente para resolver o problema do cliente

3

### Engenharia de requisitos

- Na aula 4 vamos conversar sobre a modelagem UML, discutindo sobre a origem e a importância da UML na construção de *software*
- Vamos também compreender como é estruturado o diagrama de casos de uso

4

### Modelagem de processos de negócio

- Esta aula estará organizada em 5 grandes temas, sendo eles
  - O que é UML
  - Modelo orientado a objetos
  - Técnicas de construção do diagrama de casos de uso
  - Componentes de um diagrama de caso de uso
  - Analisando um exemplo de diagrama de caso de uso

5

### O que é UML

6

- A **Unified Modelling Language (UML)** é uma linguagem ou notação de diagramas para especificar, visualizar e documentar modelos de software desenvolvidos sob os preceitos da orientação por objetos
- Como a UML não é um método de desenvolvimento, ela não diz o que fazer primeiro, o que fazer depois ou como desenhar o sistema, mas ajuda a visualizá-lo e a comunicá-lo para os outros

7

- A notação UML começou a ser criada em outubro de 1994, quando James Rumbaugh se juntou a Grady Booch na Rational Software Corporation, uma grande empresa de *software* da época
- O constante crescimento da Rational fez com que surgisse a necessidade de organizar e definir as etapas de desenvolvimento adotadas em seus projetos, organizando assim um ciclo de vida para desenvolvimento de software. Foi concebido, assim, o processo de desenvolvimento unificado, denominado RUP (*Rational Unified Process*)

8

- A existência de um modelo visual facilita a comunicação e faz com que os envolvidos na construção do *software* tenham a mesma visão e conhecimento sobre o sistema
- Cada símbolo gráfico utilizado tem uma semântica bem definida, por isso são chamados de padrão
- A padronização facilita o uso e o entendimento sobre o modelo

9

### O que são modelos?

- Um modelo de processo de desenvolvimento de *software*, ou simplesmente modelo de processo, pode ser visto como uma representação, ou abstração dos objetos e atividades envolvidas no processo de *software*
- Dessa forma, um modelo é uma simplificação ou uma abstração da realidade que é construído para compreender melhor o sistema que será desenvolvido e codificado

10

### Tipos de modelos

- Na UML, os modelos expressam duas visões diferentes, porém complementares
  - Visão estrutural
    - ✓ Os modelos criados tentam capturar a estrutura do sistema, ou seja, quais elementos compõem a estrutura do sistema e como eles se relacionam

11

### ● Visão comportamental

- ✓ Os modelos criados tentam capturar a dinâmica do sistema, ou seja, como os elementos que compõem o sistema se comunicam e como se comportam e respondem aos diversos estímulos

12

### Modelos da UML

- A UML é composta por muitos elementos e por diferentes modelos que representam as partes de um sistema de *software*
- A junção de todos os modelos criados deve dar a visão completa de como o *software* será construído e funcionará
- Os elementos UML são usados para criar diagramas, que representam uma determinada parte, ou um ponto de vista do sistema
- Os diagramas mostram visualmente a solução definida para a construção do *software*

13

### Modelo Orientado a Objetos

14

### Orientação a objetos

- A orientação a objetos é um processo conceitual independente de uma linguagem de programação, pois tem como foco visualizar o domínio do problema a ser automatizado como uma coleção de objetos e métodos associados

15

- Todo objeto é identificável e deve ter um nome claro e direto, que identifique o real objetivo do objeto
- Ou seja, as coisas do mundo real são denominadas de objetos. Um objeto é, portanto, uma entidade real ou abstrata, que modela um conceito presente na realidade humana, ocupando espaço físico ou lógico

16

### Conceitos da orientação a objetos

- Abstração
  - A abstração consiste em se concentrar nos aspectos essenciais, próprios de uma entidade e em ignorar suas propriedades acidentais
  - Ou seja, dar foco em aspectos relevantes para um determinado propósito
  - Esse conceito é muito importante para simplificar os conceitos complexos

17

### Encapsulamento

- Consiste na separação dos aspectos externos de um objeto, acessíveis por outros objetos, dos detalhes internos da implementação daquele objeto, que ficam ocultos dos demais objetos
- Ou melhor, significa separar o *software* em partes, o mais isoladas possível

18

- A ideia é tornar o *software* mais flexível, fácil de modificar e de criar novas implementações
- É uma forma de restringir o acesso ao comportamento interno de um objeto
- Cada objeto possui uma interface que é o que ele conhece e o que ele sabe fazer, sem descrever como é feito

19

## Herança

- É o compartilhamento de atributos e operações entre objetos com base em um relacionamento hierárquico
- Permite que a estrutura comum seja compartilhada por diversos outros objetos relacionados, sem redundâncias
- Cada objeto em um nível de hierarquia herda as características dos objetos nos níveis acima

20

## Técnicas de Construção do Diagrama de Casos de Uso

21

## Diagrama de caso de uso

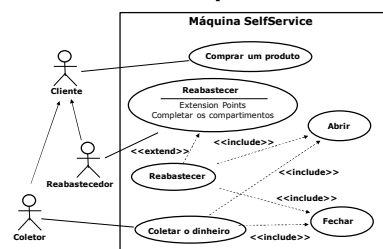
- O diagrama de caso de uso normalmente é o primeiro diagrama a ser construído, após o levantamento de requisitos
- Pois construir um diagrama de caso de uso é uma das formas de analisar o relacionamento de cada uma das funcionalidades que o *software* deve executar com os usuários que as executam

22

- O diagrama de casos de uso é bastante utilizado em projeto por vários motivos, tais como
- São utilizados para expressar a fronteira do sistema, mostram a visão estática do caso de uso, pois a visão dinâmica do caso de uso é especificada através da descrição dos cenários, além disso, por serem representações gráficas, permitem uma visão geral dos relacionamentos entre os casos de uso e os atores de um sistema

23

## Exemplo



24

- Os elementos que aparecem no diagrama de caso de uso do exemplo anterior são
- Os atores, que são representados pelos bonecos
- Os casos de uso, que são representados pelas circunferências
- Os relacionamentos, que são representados por linhas retas ou pontilhadas, dependendo do tipo de relacionamento

25

- Os atores que interagem com este sistema são o cliente em si, o profissional que reabastece a máquina (reabastecedor) e o profissional que coleta o dinheiro usado pelos clientes para comprar os produtos oferecidos (coletor)
- Os casos de uso identificados são os descritos nas elipses apresentadas
- O diagrama mostra ainda qual ator interage com cada caso de uso, e como os casos de uso interagem entre si

26

#### Características

- Utiliza um padrão fácil de ser compreendido mesmo por quem não é de TI
- O foco é o problema e não a solução computacional, por isso o diagrama de caso de uso representa os requisitos funcionais do *software*
- Permite registrar de fato o que o sistema deve fazer, e como as funcionalidades do *software* se relacionam entre si

27

- Por ser um modelo de análise, o diagrama de caso de uso é independente da abordagem de desenvolvimento
- Pode ser descrito com diversos níveis de detalhamento, facilitando a compressão no nível macro do *software* e também nos níveis de detalhe que permitam visualizar todo o relacionamento entre as funcionalidades identificadas para o *software*

28

#### Componentes de um Diagrama de Caso de Uso

- Os componentes de um diagrama de caso de uso também são chamados de elementos, e compõem a identidade visual do modelo
- Para entender um diagrama, é preciso entender a linguagem utilizada na sua construção
- A linguagem apresentada a seguir é a linguagem UML, estudada no tema 1 desta aula

29

30

### Ator

- O ator representa o papel executado por uma entidade que interage com o sistema em questão
- Um ator troca informações com o sistema, e pode ser um indivíduo ou outros sistemas, ou seja, o ator interage com o sistema, ele não faz parte do sistema
- Um usuário pode executar o papel de vários atores diferentes e um determinado ator pode ser representado por vários usuários

31

### Relacionamentos

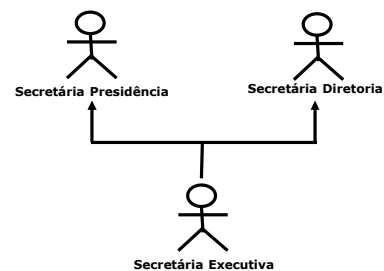
- Mostram a ligação entre os elementos de um diagrama de caso de uso, ou seja, mostram a ligação dos atores com os casos de uso e dos casos de uso entre si
- Mostram quais atores têm acesso a cada caso de uso e como os casos de uso se relacionam, apresentando a dinâmica de interação entre eles, focando no entendimento do funcionamento dos processos de negócio

32

### Relacionamento entre atores

- A generalização é identificada quando temos dois atores semelhantes, mas com um deles realizando algo a mais
- Ela deve ser identificada e validada com cuidado para não modelar relacionamentos desnecessários a um determinado ator, ou mesmo não demonstrar relacionamentos que devem ocorrer para garantir o bom funcionamento do *software*

33



34

### Relacionamento entre casos de uso

- Include => Um relacionamento include de um caso de uso A para um caso de uso B indica que B é essencial para o comportamento de A
- Pode ser dito também que B is\_part\_of A

35

- Extend => Um relacionamento extend de um caso de uso B para um caso de uso A indica que o caso de uso B pode ser acrescentado para descrever o comportamento de A (não é essencial)
- Ponto de extensão em um caso de uso é uma indicação de que outros casos de uso poderão ser adicionados a ele
- Quando o caso de uso for invocado, ele verificará se suas extensões devem ou não ser invocadas

36

## Analizando um Exemplo de Diagrama de Caso de Uso

37

## Estudo de caso

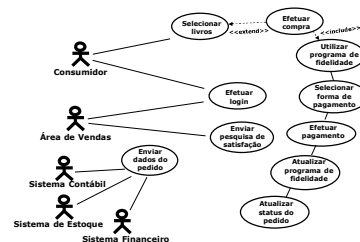
- O nosso estudo de caso será o seguinte
- "Fomos contratados pelo nosso cliente para modelar o processo de vendas on-line de livros. O nosso cliente tem uma livraria virtual, que vende produtos diretamente em um site próprio

38

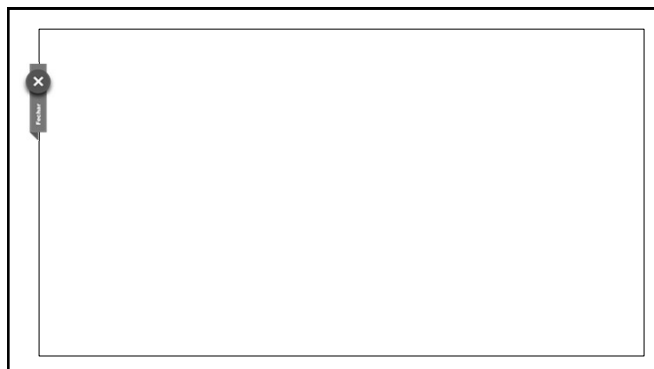
- O diferencial desta livraria é ter um estoque próprio, o que garante uma entrega mais rápida a seus clientes, e aceitar vários tipos de pagamento, como cartão de crédito, cartão de débito e boleto bancário. A livraria possui um programa de fidelidade, que permite desconto de 10% aos clientes que comprarem R\$ 500,00 ou mais em 1 ano"

39

## Diagrama de caso de uso



40



41