UML
Unified Modelling Language
(Parte 01)

Porque modelar ?

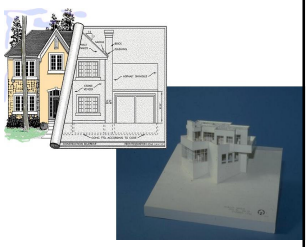
1) A analogia com as engenharias;
   a) Maquetes
   b) Modelos matemáticos
   c) Storyboards na indústria cinematográfica
   d) Modelos sociais
   e) Miniaturas de carros, aviões etc.

2) Comunicação com terceiros;

3) Participação dos 'clientes'

Porque modelar ?

- Da mesma forma que é impossível construir uma casa sem primeiramente definir sua planta, também é impossível construir um SW sem inicialmente definir sua arquitetura



- Desta forma, é extremamente importante ter uma representação visual de seu sistema antes que ele entre na etapa de implementação.

O que é um modelo ?

a) Modelo é a simplificação da realidade;

b) O modelo reduz a complexidade pela decomposição da realidade em elementos 'fáceis de entender'. (abstração)

c) Diferentes aspectos definem um sistema, portanto, é necessário vários modelos para representar cada um destes aspectos;

Objetivos do Projeto

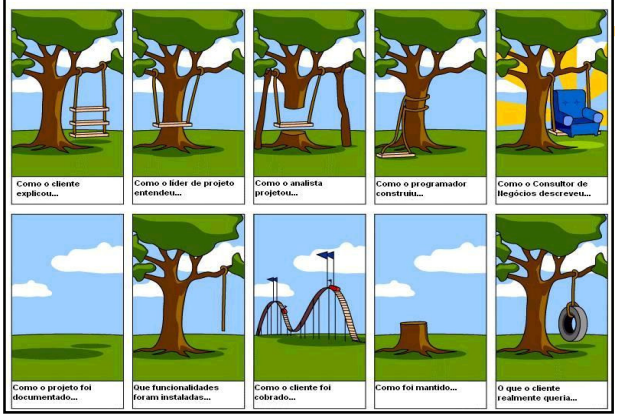
1) Entregar um produto ou serviço(Solução)

2) Esteja de acordo com os requisitos(problema)

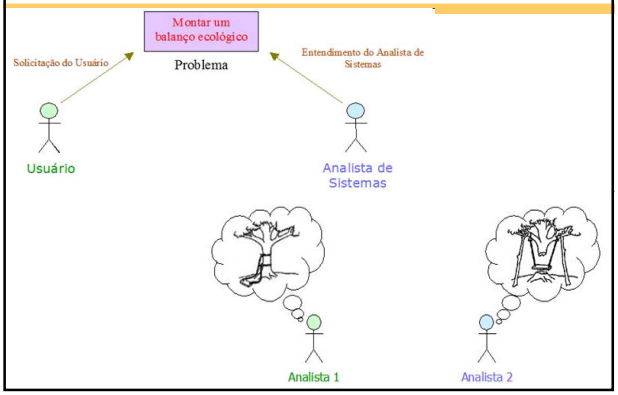
3) Tanto problema como solução estão num contexto(dominio da aplicação)

4) Entender da melhor maneira possivel o poblema , documenta-lo e disponibiliza-lo de uma forma compreensivel para o cliente.

Detalhes sobre a fase de Construção



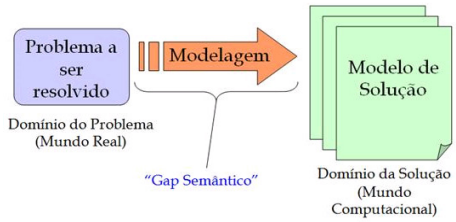
Balanço ecológico



O problema do Gap Semântico

Definido como "*a distância do que o usuário solicita para o que o analista de sistemas compreende*",

Ex"*Balanço ecológico*", surge como sendo um dos maiores fatores de insucesso de projetos de SW



8

Linguagem de Modelagem

- **A linguagem para definição de modelos deve ter:**
  - a) **Elementos do modelo: Conceito e semântica fundamentais ao modelo.**
  - b) **Notação dos elementos: representação visual dos elementos do modelo**
  - c) ***Guidelines*: guias de como e onde usar a linguagem.**

9

Princípios da Modelagem

1) A escolha de quais modelos criar tem uma profunda influência em como o problema é atacado e em como a solução é obtida.

2) Cada modelo deve permitir diferentes níveis de precisão.

3) Os melhores modelos estão conectados à realidade.

4) Um único modelo não é suficiente. Geralmente um sistema é melhor descrito por um conjunto de modelos.

10

Benefícios da Modelagem

a) **Modelos comunicam a estrutura e o comportamento do aplicativo**

b) **Modelos permitem visualizar e controlar a arquitetura do aplicativo.**

c) **Modelos permitem entender melhor o aplicativo que estamos construindo, expondo oportunidades de simplificação e reusabilidade.**

d) **Modelos permitem gerenciar os riscos.**

11

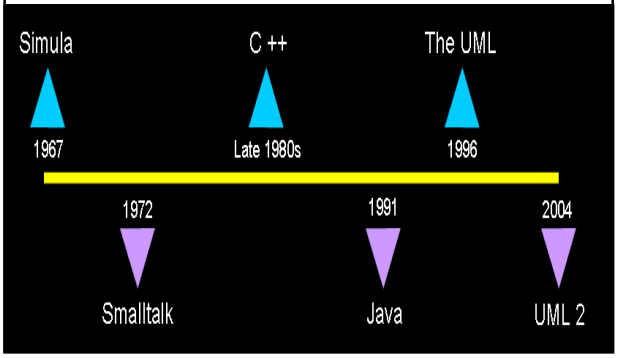
Modelagem [Tradicional x (OO)]

- **A visão tradicional do desenvolvimento de SW é baseada em uma perspectiva algorítmica, onde o principal bloco de construção é a função ou procedimento.**
- **A visão contemporânea do desenvolvimento de SW é baseada em uma perspectiva OO, onde o principal bloco de construção é o objeto ou sua classe.**

12

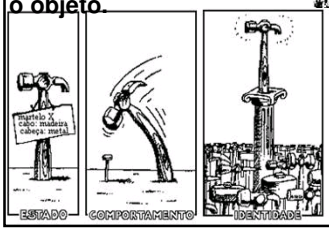
Evolução histórica de OO

Maiores marcos da tecnologia de objetos



Abstração de Procedimentos

Identifica somente os procedimentos relevantes de um objeto, retira a complexidade da operação exercida sobre o objeto.

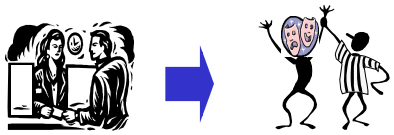


14

ETAPAS DO DESENVOLVIMENTO DE SISTEMAS EM UML

Unified Modeling Language (UML)

1) Análise de Requisitos
2) Análise
3) Projeto
4) Programação
5) Testes



15

1) Análise de Requisitos(UML)

a) Esta fase captura as intenções e necessidades dos usuários do sistema a ser desenvolvido através do uso de funções chamadas "use-cases“ [Barros, 2001].

- ✓ é identificar e documentar o que é realmente necessário,
- ✓ Comunicação (mais clara possível);
- ✓ Evitar ambiguidades;
- ✓ identificar riscos .

b) também pode ser desenvolvida baseada em Stmas de negócios,

16

2) Análise (UML)

a) preocupa-se com as primeiras abstrações (classes e objetos) e mecanismos presentes no contexto do problema [Larman, 2000].

b) Descrevemos as classes nos Diagramas de Classes e também para ajudar na descrição dos “use-cases”, podendo estas estar ligados uma nas outras através de relacionamentos.

c) Nesta fase modelamos somente as classes que pertencem ao domínio principal do problema, ou seja, classes técnicas mais detalhadas não estarão neste diagrama.

17

3) Projeto(UML)

a) cria uma representação do domínio de problema do mundo real e leva-a a um domínio de solução que é o SW [Pressman, 2005].

b) Serão adicionadas novas classes para oferecer uma infra-estrutura técnica tais como:
- ✓ interface do usuário,periféricos, BD, interação com outros sistemas, etc.

c) É feita uma junção das classes da fase da análise com as classes técnicas da nova infra-estrutura, podendo assim alterar tanto o domínio principal do problema quanto a infra-estrutura.

18

4) Programação(UML)

a) Se o desenho foi elaborado corretamente e com detalhes suficientes, a tarefa de codificação é facilitada [Furlan, 1998].

b) Para que uma fase de programação possa ter um bom desempenho, necessitamos de um projeto bem elaborado, consequentemente converteremos as classes da fase do projeto para o código da linguagem OO escolhida.

c) Em UML durante a elaboração dos modelos de análise e projeto, devemos evitar traduzi-los em códigos. Cabendo esta tarefa à fase de programação.

19

5) Testes (UML)

a) Na fase de teste executamos um programa com a intenção de descobrir um erro [Pressman, 2005].

b) Testamos cada rotina ou processo detalhadamente, bem como a integração de todos os processos e a aceitação

c) Testes de integração(classes, componentes): para verificar se estas classes estão realmente colaborando uma com as outras conforme especificado nos modelos.

d) Testes de aceitação (ALFA e BETA) é verificado se o sistema está de acordo com o especificado no diagramas de "use-cases".

20

MODELO DE PROCESSO DE SW

Artefatos de software

a) a informação susceptível à reutilização inclui a análise de requisitos, especificações do sistema, estruturas de desenho, e qualquer informação que seja necessária ao processo de desenvolvimento.

b) Estes produtos do desenvolvimento são chamados *artefatos de software*.

c) Um artefato é um sub-produto do desenvolvimento de software.

d) Podem ser manuais, arquivos executáveis, módulos etc.

21

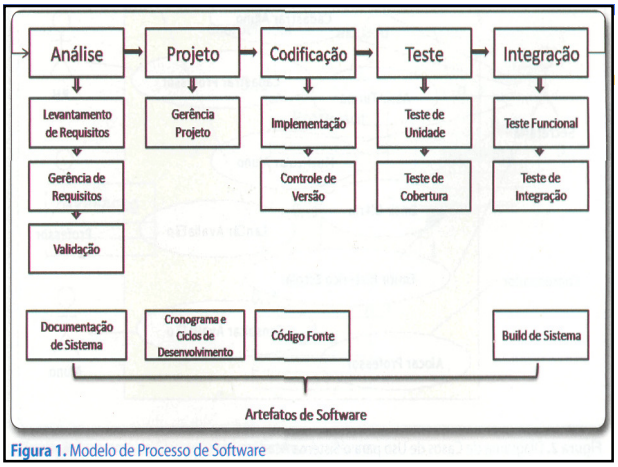
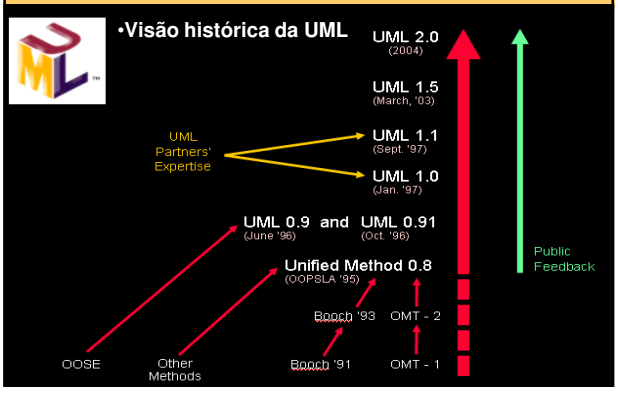


Figura 1. Modelo de Processo de Software

O que é UML



Evolução da UML

a) 1997
- ✓ Setembro: UML versão 1.1 em votação pela OMG (Object Manegement Group)
- ✓ Novembro: UML 1.1 é adotada pela OMG. A responsabilidade pela manutenção passa a ser da OMG RTF (Revision Task Force) por Cris Kobryn

b) 1998 – Versão 1.2
c) 1999 – Versão 1.3
d) 2001 – Versão 1.4
e) 2004 – Versão 2.0 (Abril)

24

Definição da OMG-UML

a) É uma linguagem para especificar, visualizar, construir e documentar sistemas através de modelos.

b) Não proprietária

c) Representa uma coleção de praticas de engenharia que comprovadamente se demonstraram eficientes na modelagem de sistemas complexos...

25

Metas de UML

a) Prover aos usuários uma linguagem de modelagem visual expressiva e pronta para uso.
b) Prover mecanismos de estensibilidade e especificação para ampliar os conceitos centrais
c) Ser independente de linguagem de programação e processos de desenvolvimento particulares.
d) Prover uma base formal para entendimento da linguagem de modelagem;
e) Suportar conceitos de desenvolvimento de nível mais alto, tais como colaborações, estruturas, modelos e componentes.
f) Integrar as melhores práticas.

26

UML

a) UML é uma linguagem para:
- ✓ visualização,
- ✓ especificação,
- ✓ construção e
- ✓ documentação de artefatos de um SW em desenvolvimento.

b) UML permite modelar:
1) elementos;
2) relacionamentos;
3) mecanismos de extensibilidade;
4) diagramas.

27

UML

1) Elementos:

1.a) estruturais
- ✓ classes, interfaces, componentes

1.b) comportamentais
- ✓ interações, máquinas de estado

1.c) grupos de elementos
- ✓ pacotes, subsistemas

1.d) outros
- ✓ anotações

28

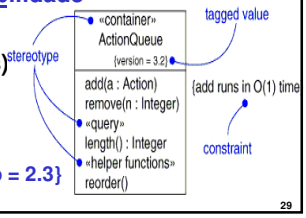
UML

2) Relacionamentos
- ✓ Dependências, Associações, Generalizações, Implementações (*realization*)

3) Mecanismos de Extensibilidade

3.a) Estereótipos
3.b) Regras (constraints)
3.c)Tagged value
(valores marcados)

{aluno = "placido neto"; versao = 2.3}



29

UML

4) Diagramas

a) Um modelo é uma descrição completa do sistema em uma determinada perspectiva.

b) Um modelo é representado por um ou mais diagramas.

c) Desta forma, um diagrama pode ser visto como uma visão dentro de um modelo.

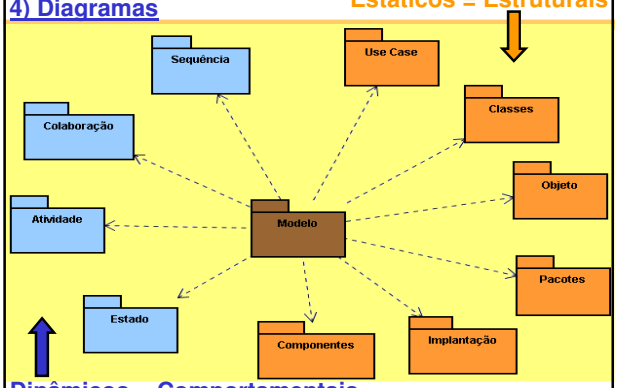
d) Um diagrama pode ser representado de várias formas, dependendo de quem irá interpretá-lo.

30

UML

4) Diagramas

Estáticos = Estruturais



Dinâmicos = Comportamentais

31

UML

Visões (4 +1) Vistas da Arquitetura

- ✓Um diagrama é uma visão sobre um modelo:

UML

a) UML não é uma metodologia:

- ✓ Não diz quem deve fazer o que, quando e como;
- ✓ UML pode ser usado segundo diferentes metodologias (XP, RUP, etc).

b) UML não é uma linguagem de programação.

c) Linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de OO

33

Propósitos da UML

1)Documentar

- ✓Modelo de arquitetura
- ✓Código-fonte
- ✓Modelo de análise
- ✓Protótipo

2) Visualizar

- ✓Visualização de todas a estrutura de um SW através de símbolos gráficos para representação de artefatos de SW

3) Especificar (Atende os requisitos)

- ✓desde a fase de análise até a fase de teste e implementação do Sma construído

34

Ferramentas de Modelagem(UML)

a) Estudo de Casos de Uso (Use Cases)

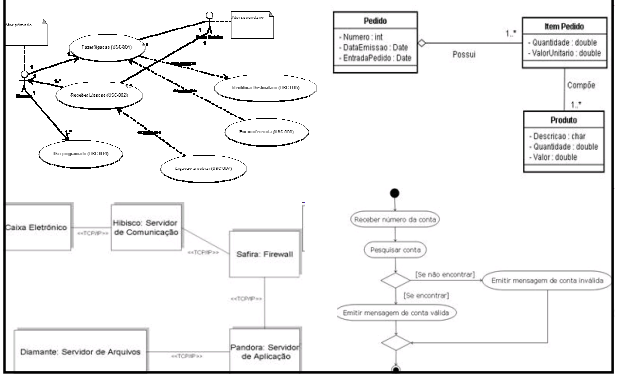
b) Diagramas:

1) Casos de uso (USC)
2) Classes
3) Seqüência
4) Objetos
5) Máquina de estados
6) Atividade
7) Interação
8) Comunicação
9) Componente, Implantação, Estrutura Composição (UML2)

35

O que é UML

Documentação:



Diagramas - 1) Caso de uso

a) **Fase:** Análise de requisitos

b) **Modela o problema a ser solucionado.**

c) **descreve um cenário - funcionalidades do Stma (ponto de vista do usuário).**

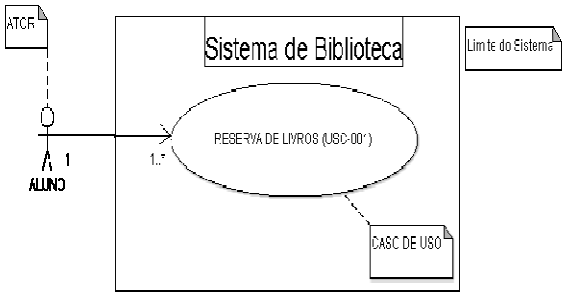
d) **O cliente deve ver no diagrama de Use Cases as principais funcionalidades de seu sistema.**

❑ **Feita a partir de várias discussões entre as pessoas envolvidas com o sistema:**

- ✓ **Clientes, Usuários, Desenvolvedores**

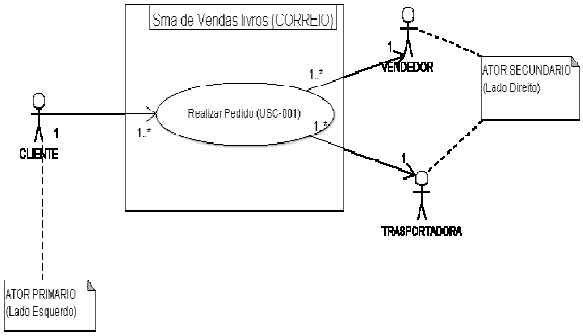
37

Exemplo (Notação)



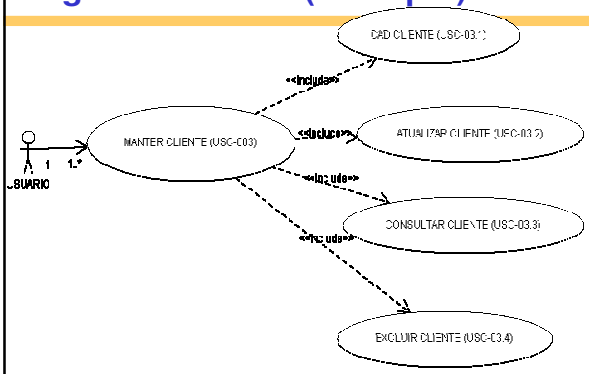
38

Exemplo (Notação)



39

Diagrama de USC (Exemplo)



40

Diagramas - 1) Caso de uso

Principais objetivos

a) **objetivo de auxiliar a comunicação entre os analistas e o cliente.**

b) **Visão funcional de tudo que o SW deverá fazer**

c) **base para todo o processo de desenvolvimento**

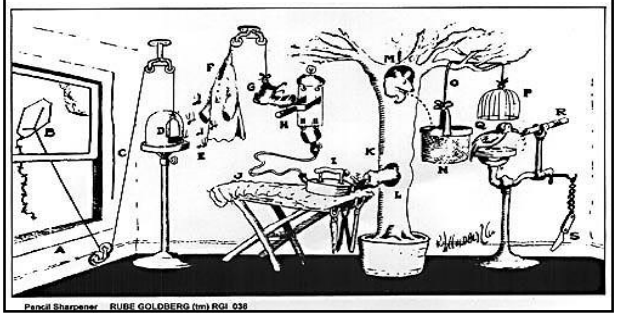
✓ **Deverá ser aplicado para os testes de validação:**

✓ **Facilitar a transformação dos requisitos funcionais em classes e operações reais do SW**

41

Documentação dos casos de uso

- **A descrição do modelo deve ser mantida no nível mais simples possível...**



Diagramas - 1) Caso de uso

- O Stma se parece com uma caixa preta que oferece funcionalidades
- Contexto de como será o funcionamento do Stma, sem se preocupar com a implementação do mesmo

Notação:

1) Atores;
2) Use case;
3) Relacionamento entre elementos;
   - 3.a) Associação entre atores e casos de uso;
   - 3.b) Generalização entre atores;
   - 3.c) Generalizações entre casos de uso;
     - ✓Extends e Includes

43

Diagramas - 1) Caso de uso

1) Ator

- é representado por um boneco e um rótulo com o nome do ator.
- é um usuário do Stma, que pode ser um usuário humano ou um outro Stma computacional.

Formas de representar os atores e relacionamento entre atores

Diagramas - 1) Caso de uso

Categorias de Atores:

a) Pessoas: Usuário, secretária, aluno, professor, administrador, empregado, Cliente, Gerente, Almoxarife, Vendedor, etc;.

b) Dispositivos: impressoras, máquina ou equipamentos ,atuadores/sensores, etc.

c) HW: (Leitora de Código de Barras, Sensor, etc.)

d) organizações (Empresa Fornecedora, Agência de Impostos, Administradora de Cartões, etc);

e) outros sistemas (Sistema de Cobrança, Sistema de Estoque de Produtos, etc).

f) Eventos Externos: Controladores de tempo - Cron

45

Como identificar os Atores ?

a) Quem utilizará a funcionalidade principal do Stma?

b) Quem precisará de suporte do Stma para fazer suas tarefas diárias ?

c) Quem necessita administrar e manter o Stma funcionando ?

d) Quais dispositivos de HW o Stma precisará manipular ?

e) Com que outros Stmas o Stma precisará interagir ?

f) Quem tem interesse nos resultados que o Stma irá produzir ?

46

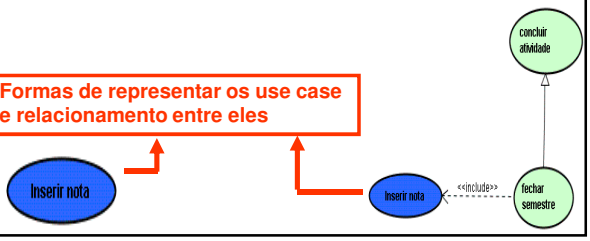
Diagramas - 1) Caso de uso

2) Use case:

a) Um *use case* é representado por uma elipse e um rótulo com o nome do *use case*.

b) Um *use case* é uma funcionalidade do Stma.

Formas de representar os use case e relacionamento entre eles



Diagramas - 1) Caso de uso

3) Relacionamentos:

a) Ajudam a descrever os *use cases*
b) Entre um ator e um *use case*
c) Define uma funcionalidade do Stma do ponto de vista do usuário.

Formas de Relacionamentos

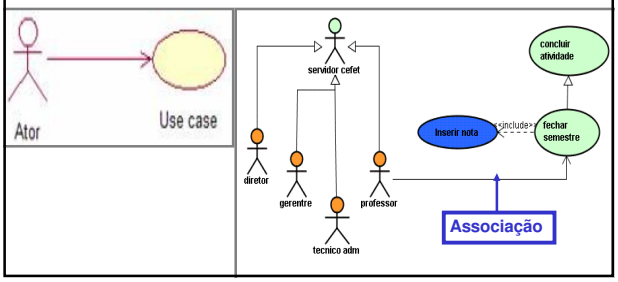
- 3.a) Associação entre atores e casos de uso;
- 3.b) Generalização entre atores;
- 3.c) Generalizações entre casos de uso;
  - Extends e Includes

48

Diagramas - 1) Caso de uso

3.a) Associação entre atores e casos de uso;

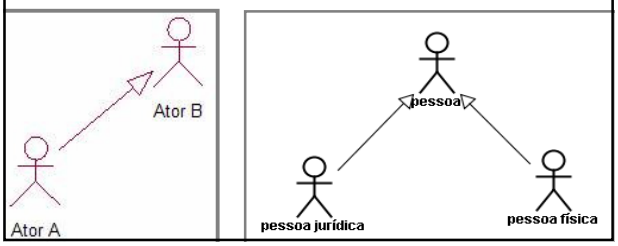
Define uma funcionalidade do Stma do ponto de vista do usuário.



Diagramas - 1) Caso de uso

3.b) Generalização entre atores;

- **Os *use cases* de B são também *use cases* de A**
- **A tem seus próprios *use cases***



Diagramas - 1) Caso de uso

3.c)Generalizações entre casos de uso;

- **Especialização / Generalização**
- ✓ **Generalização ou Especialização (é_um) *Use case* B é_um *use case* A (A é uma generalização de B, ou B é uma especialização de A).**
- ✓ **Um relacionamento entre um use case genérico para um mais específico, que herda todas as características de seu pai.**
- ✓ **é representada por uma seta que une ao Caso de Uso Geral (para onde a seta aponta)**

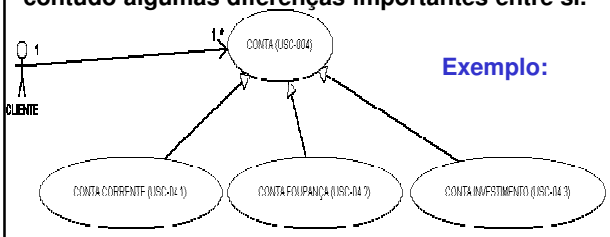
51

Diagramas - 1) Caso de uso

3.c)Generalizações entre casos de uso;

✓ **É uma forma de Associação entre casos de uso na qual existe dois ou mais casos de uso com muitas características semelhantes, apresentando contudo algumas diferenças importantes entre si.**

Exemplo:



52

Associação, inclusão e extensão

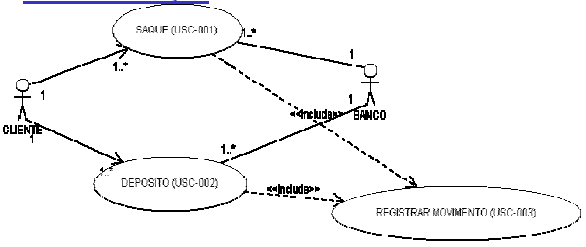
3.c.1) Inclusão (<<Include>>):

- ✓ **Relacionamento de inclusão indicam uma obrigatoriedade**
- ✓ **Utilizada quando existe um serviço, situação ou rotina comum a mais de um USC**
- ✓ **Essa rotina é colocada em um USC específico para que todos outros USC utilizem-se desse serviço**
- ✓ **Evitando-se descrever uma mesma seqüência de passos em vários USC**

53

3.c.1) Associação Inclusão

- ✓ **A execução do 1ro USC obriga a execução do 2do USC**
- ✓ **Pode ser comparado à chamada de uma sub-rotina ou função**



54

Associação, inclusão e extensão

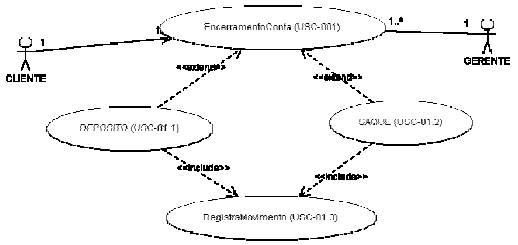
3.c.2) Extensão (<<Extend>>):

- ✓ para mostrar comportamentos opcionais,
- ✓ ou que só ocorrem mediante determinadas condições ou diferentes sequências que dependem da escolha do utilizador
- ✓ Os USC estendido descrevem cenários que somente ocorrerão em um situação específica, se uma determinada condição for satisfeita.

55

3.c.2) Associação - Extensão

Indicam a necessidade de um teste para determinar se é necessário executar ou não o USC estendido



56

Lista de Exercicios

a) Elaborar um diagrama de Caso de uso para um usuário (cliente de uma loja) que vai efetuar o pagamento. A loja tem a opção, pagamento a vista, cheque, cartão de credito

b) Elaborar um diagrama de Caso de uso para um usuário (aluno e professor) para um sistema de Biblioteca que tem as opções DEVOLVER LIVRO e RESERVAR LIVRO

c) Elaborar um diagrama de Caso de uso para um usuário (escritor) que utiliza um editor de texto com as opções imprimir, corrigir texto, e salvar.

d) Elaborar um diagrama de Caso de uso para um usuário (cliente de um banco) que deseja realizar as seguintes operações(transação) Realizar Saque, Obter Extrato e Realizar Transferência. Lembrando que para cada operação (transação) o sistema solicita a identificação do mesmo.

57

Ex.: Características do sistema:

1) O sistema irá permitir o acesso sem restrições para qualquer usuário da empresa , não havendo portanto controle de acesso.
2) Não existe qualquer interface com outros sistemas Existentes
3) Deverão ser cadastrados os seguintes dados de um cliente: Nome , endereço, Cidade , Cep , Telefone , Email e Observações.
4) O nome do cliente , endereço , cidade , cep e email são obrigatórios e deverão ser sempre informados

Lista de ator(es):

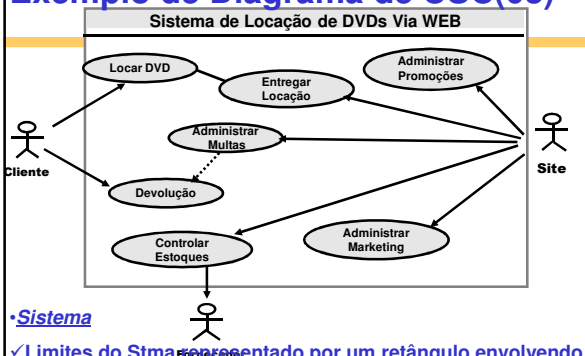
- ❑ Usuário (funcionário da empresa)

Casos de uso identificados

1. Exibir Clientes Cadastrados
2. Efetuar Manutenção de clientes

58

Exemplo de Diagrama de USC(03)



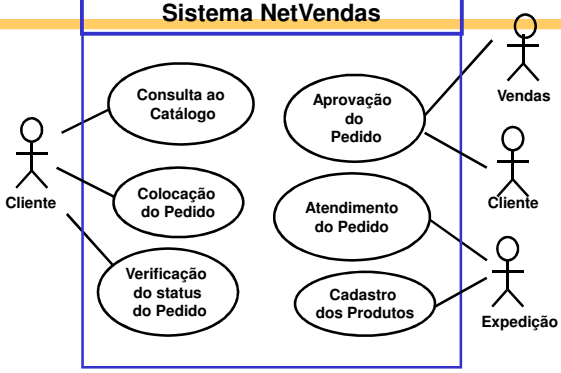
- *Sistema*

✓Limites do Stma representado por um retângulo envolvendo os use cases que compõem o sistema.

✓Nome do Stma: Localizado dentro do retângulo.

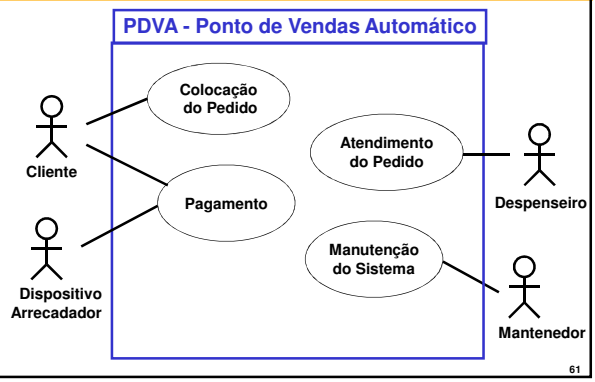
59

Exemplo de Diagrama de USC(04)



60

Exemplo de Diagrama de USC(05)



61

Referências

- Princípios de Analise e Projeto de sistema com UML. Eduardo Bezerra. Editora Campus, 2003
- Exercitando modelagem em UML. Ana Cristina Melo. Brasport. 2006
- Desenvolvendo SW com UML 2,0 definitivo. Ernani Medeiros. Editora Pearson-Makron Books, 2004
- UML 2,0 Do requisito a Solução, Editora Erica 2005. Adilson da Silva Lima

62