

# UML - *Unified Modeling Language*

## Casos de Uso

Marcio E. F. Maia

**Disciplina: Engenharia de Software**

Professora: Rossana M. C. Andrade

Curso: Ciências da Computação

Universidade Federal do Ceará

24 de abril de 2008



# Agenda

- 1 **Introdução**
  - Conceitos básicos
- 2 **UML**
  - Características Gerais da UML
- 3 **Diagramas da UML**
  - Diagramas Estruturais
  - Diagramas Comportamentais
  - Ferramentas
- 4 **Diagrama de Casos de Uso**
  - Definições
- 5 **Exemplo**
  - Exercício 1



# Agenda

- 1 **Introdução**
  - **Conceitos básicos**
- 2 UML
  - Características Gerais da UML
- 3 Diagramas da UML
  - Diagramas Estruturais
  - Diagramas Comportamentais
  - Ferramentas
- 4 Diagrama de Casos de Uso
  - Definições
- 5 Exemplo
  - Exercício 1



# Por que modelar Software?

- Melhorar a comunicação entre desenvolvedores, arquitetos e clientes
- Diminuir inconsistências no projeto
- Diminuir o tempo total de desenvolvimento
- Aumentar legibilidade
- Melhorar documentação e facilitar manutenção
- ...



# Levantamento e análise de requisitos

Compreender as reais necessidades dos usuários

- Entrevistas

Análise dos requisitos

- Garantir que as necessidades foram bem compreendidas
- Auxílio de uma linguagem de modelagem
- Quanto mais cedo uma inconsistência for identificada, menor o custo para corrigi-la



# Prazos e custos

- Qual o real prazo de entrega
- Quantos profissionais deverão trabalhar?
- Qual o custo?
- Qual a complexidade?



# Prazos e custos

- Qual o real prazo de entrega
- Quantos profissionais deverão trabalhar?
- Qual o custo?
- Qual a complexidade?

**Uma boa modelagem pode auxiliar a responder essas perguntas**

- Tempo: difícil precisar tempo total de desenvolvimento
- Custo: diretamente relacionado ao escopo e ao tempo previsto de desenvolvimento



# Manutenção

Autores afirmam que representa entre 40% e 60% do custo total de desenvolvimento de um software

- Modelagem diminui erros de especificação e compreensão dos requisitos

Facilita a evolução do software

- Dinamicidade do ambiente
  - Direção da empresa
  - Leis ou alíquotas



# Manutenção

Autores afirmam que representa entre 40% e 60% do custo total de desenvolvimento de um software

- Modelagem diminui erros de especificação e compreensão dos requisitos

Facilita a evolução do software

- Dinamicidade do ambiente
  - Direção da empresa
  - Leis ou alíquotas



# Reuso

Novos projetos podem se beneficiar do conhecimento gerado pela empresa em projetos anteriores

- Modelagem facilita a difusão do conhecimento gerado nos projetos
- Melhora as estimativas em novos projetos
- Diminui o custo e o prazo de desenvolvimento através da reutilização de rotinas, funções, algoritmos, documentos, casos de testes, ...



# Agenda

- 1 Introdução
  - Conceitos básicos
- 2 **UML**
  - **Características Gerais da UML**
- 3 Diagramas da UML
  - Diagramas Estruturais
  - Diagramas Comportamentais
  - Ferramentas
- 4 Diagrama de Casos de Uso
  - Definições
- 5 Exemplo
  - Exercício 1



# UML - *Unified Modeling Language*

Linguagem visual utilizada para modelar sistemas computacionais

- Definir características do software
  - Requisitos
  - Comportamento
  - Estrutura lógica
  - Dinâmica dos processos
  - Arquitetura



# UML - *Unified Modeling Language*

Linguagem visual utilizada para modelar sistemas computacionais

- Definir características do software
  - Requisitos
  - Comportamento
  - Estrutura lógica
  - Dinâmica dos processos
  - Arquitetura



# UML - *Unified Modeling Language*

Linguagem visual utilizada para modelar sistemas computacionais

- Definir características do software
  - Requisitos
  - Comportamento
  - Estrutura lógica
  - Dinâmica dos processos
  - Arquitetura

## Definição

Linguagem visual para especificar, visualizar, construir e documentar os artefatos gerados durante o processo de desenvolvimento de um sistema de software



# Histórico UML

Primeira versão criada em 1996 a partir da união de três metodologias

- Booch
- OMT - *Object Modeling Technique*
- OOSE *Object-Oriented Software Engineering*

Contou com o amplo apoio da Rational Software

Adotada em 1997 pela OMG (*Object Management Group*)  
como a linguagem padrão de modelagem



# Histórico UML

Primeira versão criada em 1996 a partir da união de três metodologias

- Booch
- OMT - *Object Modeling Technique*
- OOSE *Object-Oriented Software Engineering*

Contou com o amplo apoio da Rational Software

Adotada em 1997 pela OMG (*Object Management Group*) como a linguagem padrão de modelagem

Recentemente lançada a versão 2.0

Melhoramento e criação de novos diagramas



# Para que servem os diagramas?

Cada diagrama apresenta uma visão diferente do sistema.  
Cada diagrama é útil em um determinado estágio de desenvolvimento do sistema.

- Alguns enfocam uma visão mais geral como os Diagramas de Caso de Uso
- Enfoques mais específicos como os Diagramas de Seqüencia

Outros níveis de detalhes apresentados pelos diagramas

- Organização estrutural do sistema
- Comportamento de um processo específico
- Necessidades físicas do sistema



# Agenda

- 1 Introdução
  - Conceitos básicos
- 2 UML
  - Características Gerais da UML
- 3 **Diagramas da UML**
  - Diagramas Estruturais
  - Diagramas Comportamentais
  - Ferramentas
- 4 Diagrama de Casos de Uso
  - Definições
- 5 Exemplo
  - Exercício 1



# Diagramas Estruturais e Comportamentais

## Diagramas Estruturais

- Diagrama de Classes
- Diagrama de Objetos
- Diagrama de Componentes
- Diagrama de Implantação
- Diagrama de Pacotes

## Diagramas Comportamentais

- Diagrama de Casos de Uso
- Diagrama de Seqüência
- Diagrama de Colaboração
- Diagrama de Gráfico de Estados
- Diagrama de Atividades



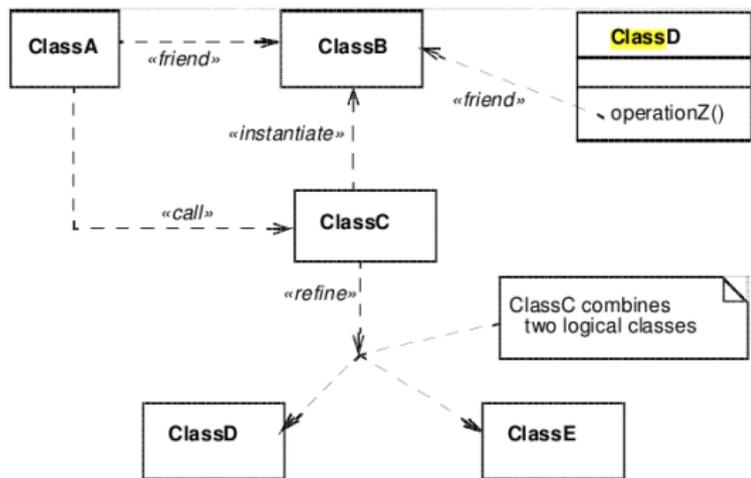
# Diagrama de Classes

- Diagrama mais utilizado e importante da UML
- Define a estrutura das classes do sistema
- Determina atributos e métodos de cada classe, além de mostrar como as classes se relacionam



# Diagrama de Classes

- Diagrama mais utilizado e importante da UML
- Define a estrutura das classes do sistema
- Determina atributos e métodos de cada classe, além de mostrar como as classes se relacionam



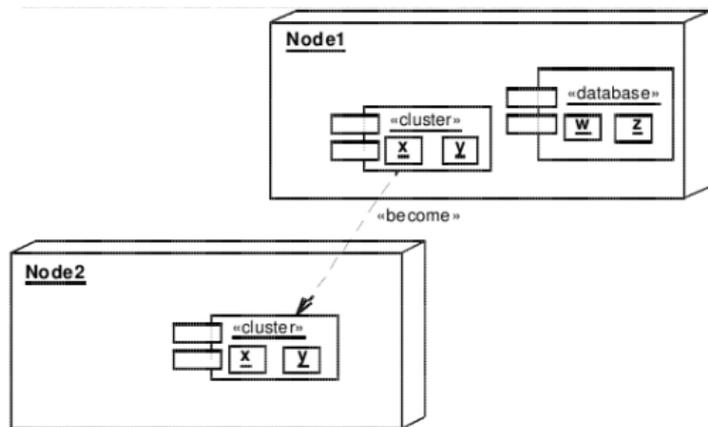
# Diagrama de Implantação

- Necessidades de hardware
- Características físicas como servidores, estações, topologias
- Apresentar características de alto nível relevantes
- Identificar os elementos arquiteturais relevantes



# Diagrama de Implantação

- Necessidades de hardware
- Características físicas como servidores, estações, topologias
- Apresentar características de alto nível relevantes
- Identificar os elementos arquiteturais relevantes



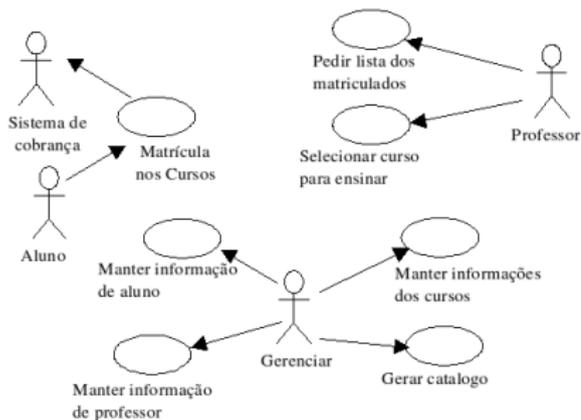
# Diagrama de Caso de Uso

- Diagrama mais geral e informal
- Levantamento e análise de requisitos
- Consultado durante todo o processo de modelagem



# Diagrama de Caso de Uso

- Diagrama mais geral e informal
- Levantamento e análise de requisitos
- Consultado durante todo o processo de modelagem



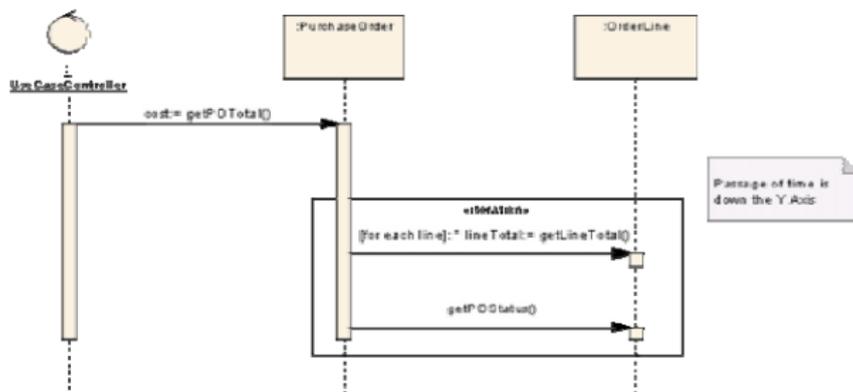
# Diagrama de Seqüencia

- Ordem temporal em que as mensagens são trocadas entre os objetos
- Identifica o evento gerador do processo, o ator responsável e como o processo irá se desenrolar



# Diagrama de Seqüência

- Ordem temporal em que as mensagens são trocadas entre os objetos
- Identifica o evento gerador do processo, o ator responsável e como o processo irá se desenrolar



# Ferramentas CASE - *Computer-Aided Software Engineering*

Ferramentas que de alguma forma colaboram para a execução de uma ou mais atividades realizadas durante o processo de Engenharia de Software.

- 1 Rational Rose
  - Mais conhecida e utilizada
  - [www.rational.com](http://www.rational.com)
- 2 Umbrello, JUDE, Visual Paradigm for UML, Poseidon for UML
  - Ferramentas livres
- 3 Enterprise Architect
  - Bastante completa
  - [www.sparxsystems.com.au](http://www.sparxsystems.com.au)



# Agenda

- 1 Introdução
  - Conceitos básicos
- 2 UML
  - Características Gerais da UML
- 3 Diagramas da UML
  - Diagramas Estruturais
  - Diagramas Comportamentais
  - Ferramentas
- 4 Diagrama de Casos de Uso
  - Definições
- 5 Exemplo
  - Exercício 1



# Diagrama de Casos de Uso

## Definição

Os diagramas de caso de uso capturam o comportamento do sistema, de uma forma geral e informal, durante a fase de levantamento e análise de requisitos.

- Representam os requisitos funcionais do sistema



# Diagrama de Casos de Uso

## Definição

Os diagramas de caso de uso capturam o comportamento do sistema, de uma forma geral e informal, durante a fase de levantamento e análise de requisitos.

- Representam os requisitos funcionais do sistema
- Apresenta uma visão geral externa das funções e serviços
- Não se preocupa como serão implementadas
- Identifica os usuários, os seus papéis e as funções que serão requisitadas por cada usuário específico



# Utilidade

- Servir como base para toda a modelagem do sistema
- Ilustrar o comportamento do sistema
- Facilitar compreensão
- Auxiliar na identificação de mal-entendidos



# Atores

Papéis desempenhados por todos que de alguma forma utilizarão ou irão interagir com os serviços e funções do sistema

- Não são parte do sistema
- Usuários
- Hardware especial
- Software externo

É qualquer elemento externo que interaja com o sistema



# Atores

Papéis desempenhados por todos que de alguma forma utilizarão ou irão interagir com os serviços e funções do sistema

- Não são parte do sistema
- Usuários
- Hardware especial
- Software externo

É qualquer elemento externo que interaja com o sistema



Gerente



Funcionário



Sistema da Biblioteca



Caixa Eletrônico



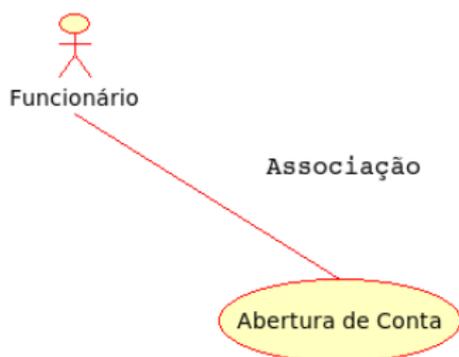
# Casos de Uso

- Serviços, tarefas ou funções que podem ser utilizados pelos atores
- Expressar e documentar os comportamentos pretendidos
- Representado por uma elipse e uma descrição sucinta e representativa



# Casos de Uso

- Serviços, tarefas ou funções que podem ser utilizados pelos atores
- Expressar e documentar os comportamentos pretendidos
- Representado por uma elipse e uma descrição sucinta e representativa



- Documentação fornece linhas gerais sobre seu funcionamento
- Evento gerador, atores ou restrições
- Formalismo fica à critério do engenheiro de software



# Relacionamentos entre Casos de Uso

## Especialização

- Dois ou mais casos de uso com características semelhantes
- Herança



# Relacionamentos entre Casos de Uso

## Especialização

- Dois ou mais casos de uso com características semelhantes
- Herança



# Relacionamentos entre Casos de Uso - Cont.

## Inclusão

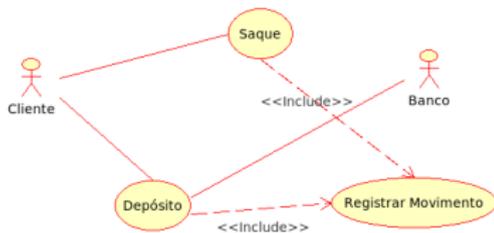
- Serviço, situação ou rotina comum a mais de um Caso de Uso
- Evitar reescrita de trechos semelhantes
- Relação de obrigatoriedade (comparado a uma sub-rotina)



# Relacionamentos entre Casos de Uso - Cont.

## Inclusão

- Serviço, situação ou rotina comum a mais de um Caso de Uso
- Evitar reescrita de trechos semelhantes
- Relação de obrigatoriedade (comparado a uma sub-rotina)



# Relacionamentos entre Casos de Uso - Cont.

## Extensão

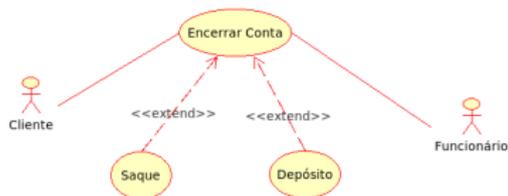
- Descrever cenários opcionais
- Situações específicas onde determinadas condições são satisfeitas



# Relacionamentos entre Casos de Uso - Cont.

## Extensão

- Descrever cenários opcionais
- Situações específicas onde determinadas condições são satisfeitas



# Documentação de Casos de Uso

Descrever por meio de uma linguagem simples, e em linhas gerais, a função do Caso de Uso

- Não existe um formato padrão
- Flexível
- Ideal é que seja simples, de tal forma que qualquer pessoa possa entendê-la



## Documentação de Casos de Uso - Cont.

É aconselhável que cada documentação contenha os seguintes campos:

- **Pré-Condições:** o sistema admite que tais condições são verdadeiras  
Ex. Para utilizar uma determinada função o usuário deve estar logado e autenticado
- **Pós-Condições:** descreve o que mudou com a execução do caso de uso  
Ex. Um usuário foi adicionado no sistema
- **Fluxo Principal:** descreve a seqüência normal de eventos  
Representa as ações dos atores e a resposta do sistema para tais ações
- **Fluxo Alternativo:** qualquer seqüência de eventos diferente do fluxo principal.



# Agenda

- 1 Introdução
  - Conceitos básicos
- 2 UML
  - Características Gerais da UML
- 3 Diagramas da UML
  - Diagramas Estruturais
  - Diagramas Comportamentais
  - Ferramentas
- 4 Diagrama de Casos de Uso
  - Definições
- 5 **Exemplo**
  - **Exercício 1**



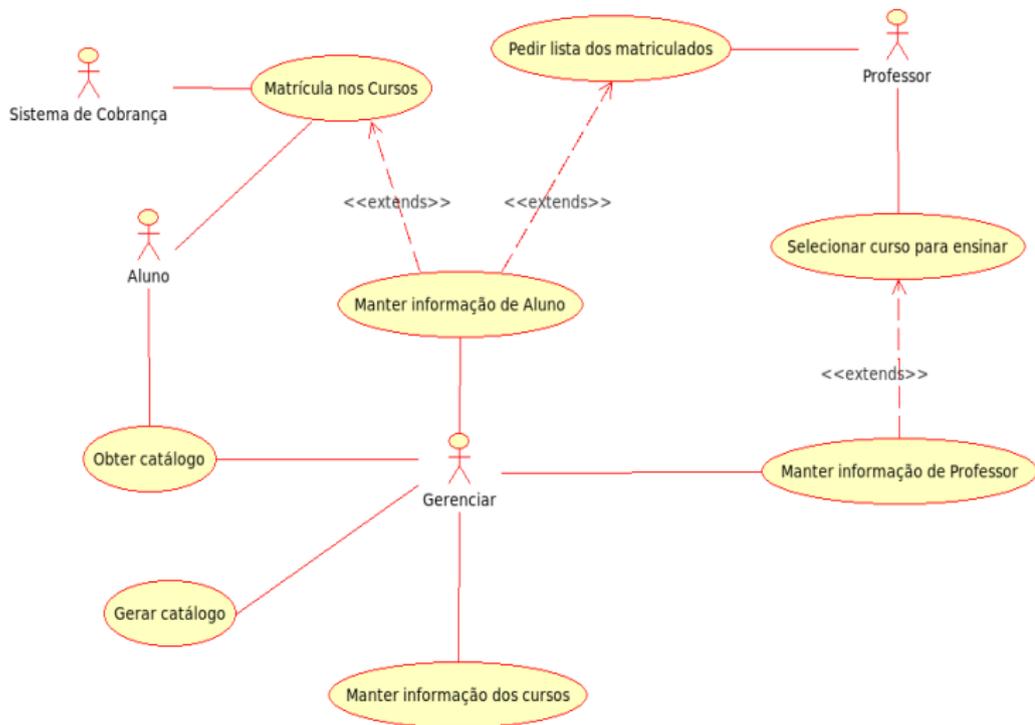
# Especificação do problema

No início de cada semestre os alunos devem requisitar um catálogo de cursos contendo os cursos oferecidos no semestre. Este catálogo deve conter informações a respeito de cada curso tais como: professor, departamento e pré-requisitos. Desse modo, os alunos podem tomar suas decisões mais apropriadamente. O novo sistema permitirá que os alunos selecionem quatro cursos oferecidos para o próximo semestre. Além disso, o aluno indicará dois cursos alternativos, caso o aluno não possa ser matriculado na primeira opção. Cada curso terá o máximo de 10 e o mínimo de 3 alunos. Um curso com o número de alunos inferior a 3 será cancelado. Para cada matrícula feita por um aluno, o sistema envia informação ao sistema de cobrança para que cada aluno possa ser cobrado durante o semestre. Os Professores devem acessar o sistema on-line, indicando quais cursos irão lecionar. Eles também podem acessar o sistema para saber quais alunos estão matriculados em cada curso. Em cada semestre, há um prazo para alteração de matrícula. Os alunos devem poder acessar o sistema durante esse período para adicionar ou cancelar cursos.



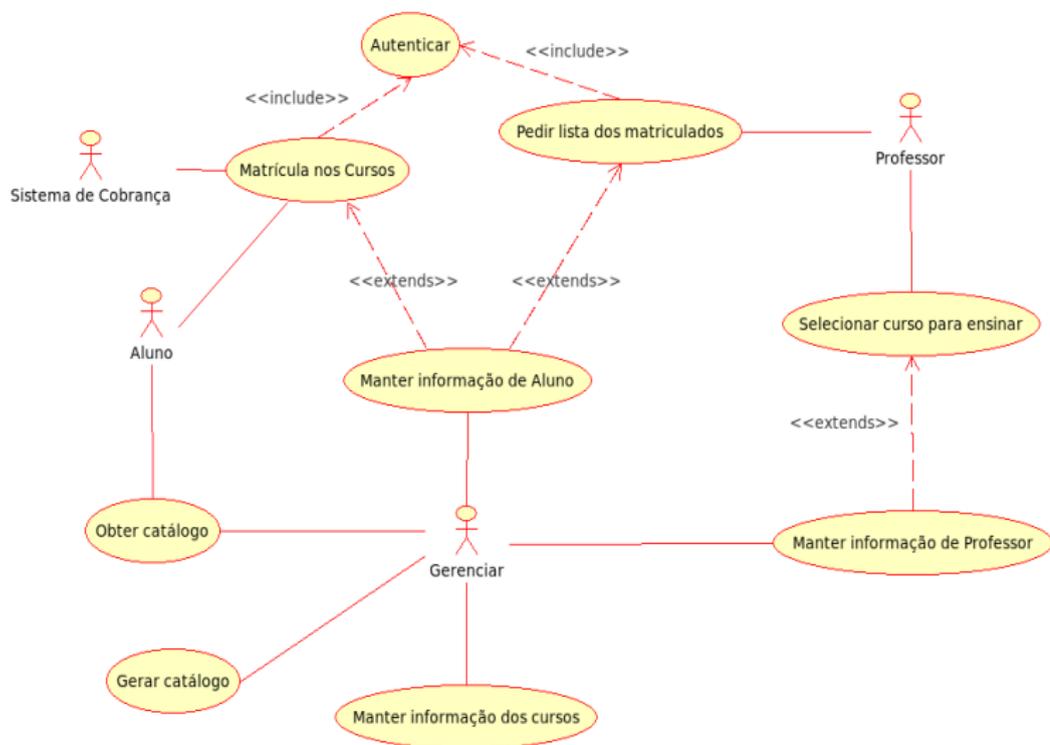
## Exercício 1

# Diagrama de Casos de Uso



## Exercício 1

# Diagrama de Casos de Uso



Obrigado!  
marcio@great.ufc.br

