

A Linguagem UML

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

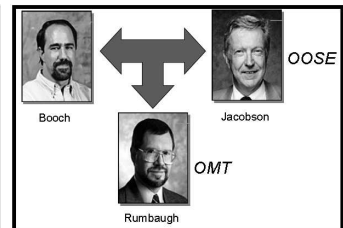
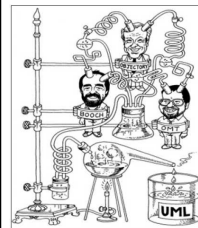
A Linguagem UML

- UML (Linguagem de Modelagem Unificada)
- É uma notação gráfica (visual) para projetar sistemas OO
 - Não é uma linguagem de programação
- Define diagramas padronizados
- É extensível
- É complexa (muitos diagramas)
 - Usaremos apenas um sub-conjunto da UML

De onde surgiu?

- Da união de três metodologias de modelagem
 - Método de Booch - Grady Booch
 - Método OMT - Ivar Jacobson
 - Método OOSE - James Rumbaugh
- Os três amigos

Fundadores da UML



História da UML

- 1994: Booch, Jacobson e Rumbaugh começaram a unificar suas notações
- 1996: Primeira versão (beta) da UML foi liberada
- 1996/97: Grandes empresas formaram a "UML Partners"
 - HP, IBM, Microsoft, Oracle, etc.
- 1997: UML foi adotada pela a OMG (*Object Management Group*) como linguagem padrão de modelagem

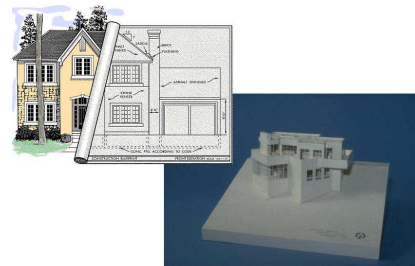
O que é modelagem?

- Atividade de construir modelos que expliquem as características ou comportamentos de um sistema
- A UML pode ser usada com todos os processos de software
 - Modelo Cascata, Iterativo, Espiral ...
- Utilizado principalmente em duas atividades dos processos
 - Especificação de requisitos
 - Projeto

Por que modelar?

- Tão essencial quanto ter uma planta antes da construção de uma casa
 - Melhora a comunicação entre os membros da equipe e o cliente
 - A equipe entende melhor o sistema
 - Permite analisar o sistema sobre vários aspectos
 - Facilita a programação e a manutenção
 - Diminui a possibilidade de erros

Projetar é Fundamental



Por que usar UML?

- Bons modelos são essenciais para a comunicação entre os *stakeholders*
- Padronização
 - Todo o time entende a modelagem, facilitando a manutenção
- Facilita a programação
 - Ferramentas para modelagem e geração de código

Modelagem Orientada a Objetos

- Maneira natural de visualizar o software
- Modela o software semelhante ao mundo real - usando objetos
 - Pessoas, animais, plantas, carros, etc.
- Humanos pensam em termos de objetos
 - Mais alto nível

UML Define Diagramas

- Tipos Principais de Diagramas
 - Estrutural
 - Comportamental
- Objetivos
 - Visualizar o sistema
 - Especificar estrutura e/ou comportamento
 - Guiar e documentar as decisões

Alguns Diagramas UML

- Diagramas Estruturais (Estáticos)
 - Diagrama de Casos de Uso
 - Diagrama de Classes
 - Diagramas de Objetos
 - Diagrama de Componentes, etc.
- Diagramas Dinâmicos
 - Diagrama de Sequência
 - Diagrama de Estados
 - Diagrama de Atividades
 - Diagrama de Colaboração, etc.

Resumo de Alguns Diagramas UML

Diagrama de Caso de Uso

- Diagrama mais geral da UML
- Usado geralmente na fase de Especificação de Requisitos
- Mostra
 - Quais usuários realizam que funcionalidades do sistema
 - Alguns relacionamentos entre estas funcionalidades

Diagrama de Caso de Uso

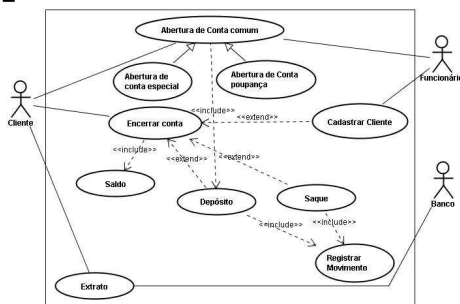


Diagrama de Sequência

- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Pode se basear em um Caso de Uso
- Identifica
 - Os eventos associados a funcionalidade modelada
 - O ator responsável por este evento

Diagrama de Sequência

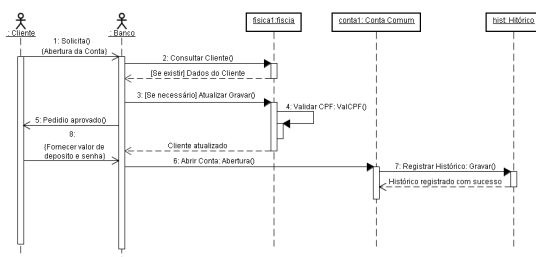


Diagrama de Classes

- Diagrama mais utilizado da UML
- Serve de apoio para a maioria dos outros diagramas
- Define a estrutura das classes do sistema
- Estabelece como as classes se relacionam

Diagrama de Classes

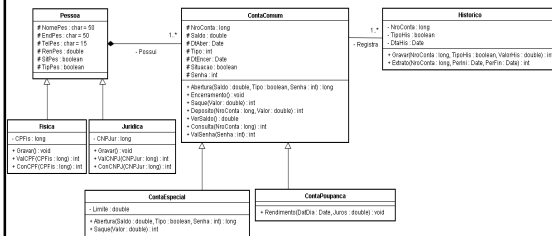


Diagrama de Objetos

- Complemento do Diagrama de Classes
- Exibe os valores armazenados pelos objetos de um Diagrama de Classes

Diagrama de Objetos

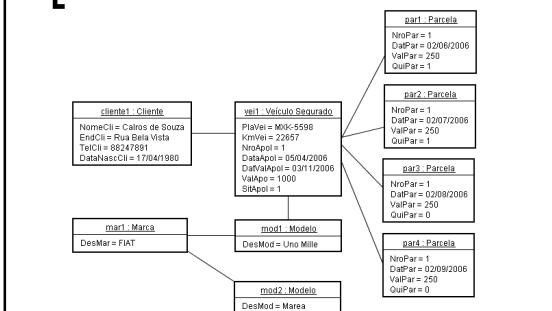


Diagrama de Colaboração

- Amplamente associado ao Diagrama de Sequência
- São complementares
- Não se preocupa com a temporalidade
- Define
 - Como os objetos estão vinculados
 - Quais mensagens são trocadas entre objetos

Diagrama de Colaboração

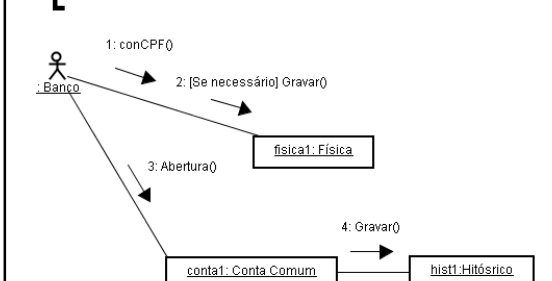


Diagrama de Estados

- Modela as mudanças sofridas por um objeto dentro de um determinado processo
- É utilizado para acompanhar os estados por que passa uma instância de uma classe

Diagrama de Estados

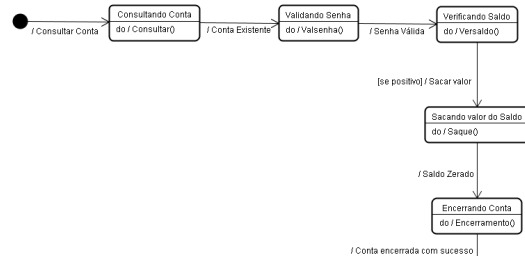


Diagrama de Atividades

- Descreve as atividades a serem executadas para a conclusão de um processo
- Concentra-se na representação do fluxo de controle de um processo

Diagrama de Atividades

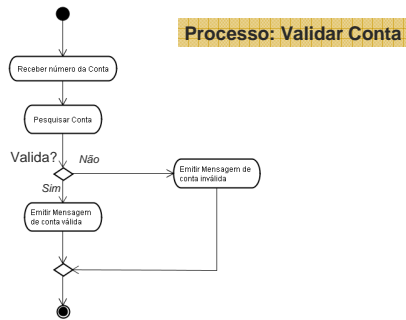


Diagrama de Componentes

- Associado a linguagem de programação a ser utilizada
- Este diagrama representa os componentes do sistema
- Os componentes serão implementados como
 - Classes de código-fonte
 - Bibliotecas
 - Arquivos de ajuda, etc.

Diagrama de Componentes

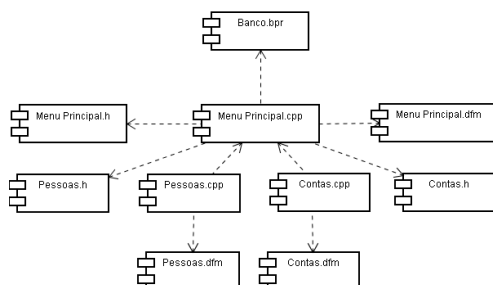
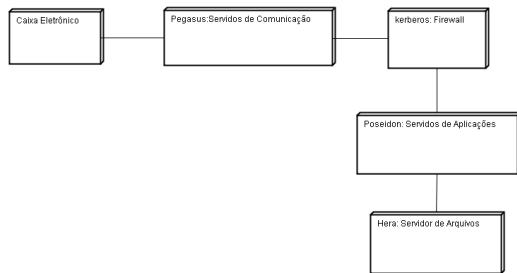


Diagrama de Implantação

- Determina as necessidades de hardware
- Características físicas do sistema
 - Servidores
 - Estações
 - Topologias de rede
 - Protocolos de comunicação, etc.

[Diagrama de Implantação]



[Outros diagramas]

- Diagrama de Pacotes
 - Representar os sub-sistemas englobados por um sistema
- Diagrama de Interação Geral
 - Fornece uma visão geral dentro de um sistema ou processo de negócios
- Diagrama de Temporização
 - Descreve a mudança no estado ou na condição de uma instância de uma classe ou seu papel durante um período de tempo

[Bibliografia]

- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
 - Capítulos 1