



Tecnologia em Análise e Desenvolvimento de Sistemas

Engenharia de Software II

UML

Prof. Claudemir Santos Pinto
profdemir@yahoo.com.br

UML – Unified Modeling Language

- Introdução

A UML é uma linguagem de *modelagem* e não uma *metodologia* de desenvolvimento, que diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar o desenho de seu projeto e a comunicação entre objetos.

UML – Unified Modeling Language

- Introdução

Com a modelagem podemos:

- especificar a estrutura e o comportamento do sistema
- visualizar o sistema como ele é ou como desejamos que ele seja
- documentar todas as decisões tomadas

UML – Unified Modeling Language

- Introdução

A importância da modelagem:

- construímos modelos para compreendermos melhor o sistema que estamos desenvolvendo

UML – Unified Modeling Language

■ Objetivos:

- Especificação, documentação e visualização lógica mais abrangente do desenvolvimento completo de um sistema de informação.
- É também uma maneira de padronizar as formas de modelagem.

UML – Unified Modeling Language

■ Visões

Construir sistemas de software é uma tarefa que requer a visualização de sua arquitetura sob várias perspectivas e em diferentes níveis de detalhamento:

- visão dos gerentes,
- visão dos analistas,
- visão dos programadores e
- visão dos usuários finais

UML – Unified Modeling Language

- Visões da Arquitetura de um Sistema



UML – Unified Modeling Language

■ Visões da Arquitetura de um Sistema

Visão	Descrição
Caso de Uso	<ul style="list-style-type: none">• focaliza os comportamentos de um sistema• deve ser transparente para todos os envolvidos na construção do sistema• diagramas utilizados<ul style="list-style-type: none">• diagramas de casos de uso• diagramas de interação• diagramas de atividades• diagramas de gráficos de estado

UML – Unified Modeling Language

■ Visões da Arquitetura de um Sistema

Visão	Descrição
Projeto	<ul style="list-style-type: none">• focaliza a estrutura do sistema• mostra as classes, colaborações e as interfaces do sistema• diagramas utilizados:<ul style="list-style-type: none">• diagrama de classes• diagramas de objetos• diagramas de interação• diagramas de atividades• diagramas de gráficos de estado

UML – Unified Modeling Language

■ Visões da Arquitetura de um Sistema

Visão	Descrição
Processo	<ul style="list-style-type: none">• focaliza as questões de:<ul style="list-style-type: none">• desempenho• escalabilidade• mecanismos de concorrência• mecanismos de sincronização• diagramas utilizados:<ul style="list-style-type: none">• diagrama de classes• diagramas de objetos• diagramas de interação• diagramas de atividades• diagramas de gráficos de estado

UML – Unified Modeling Language

■ Visões da Arquitetura de um Sistema

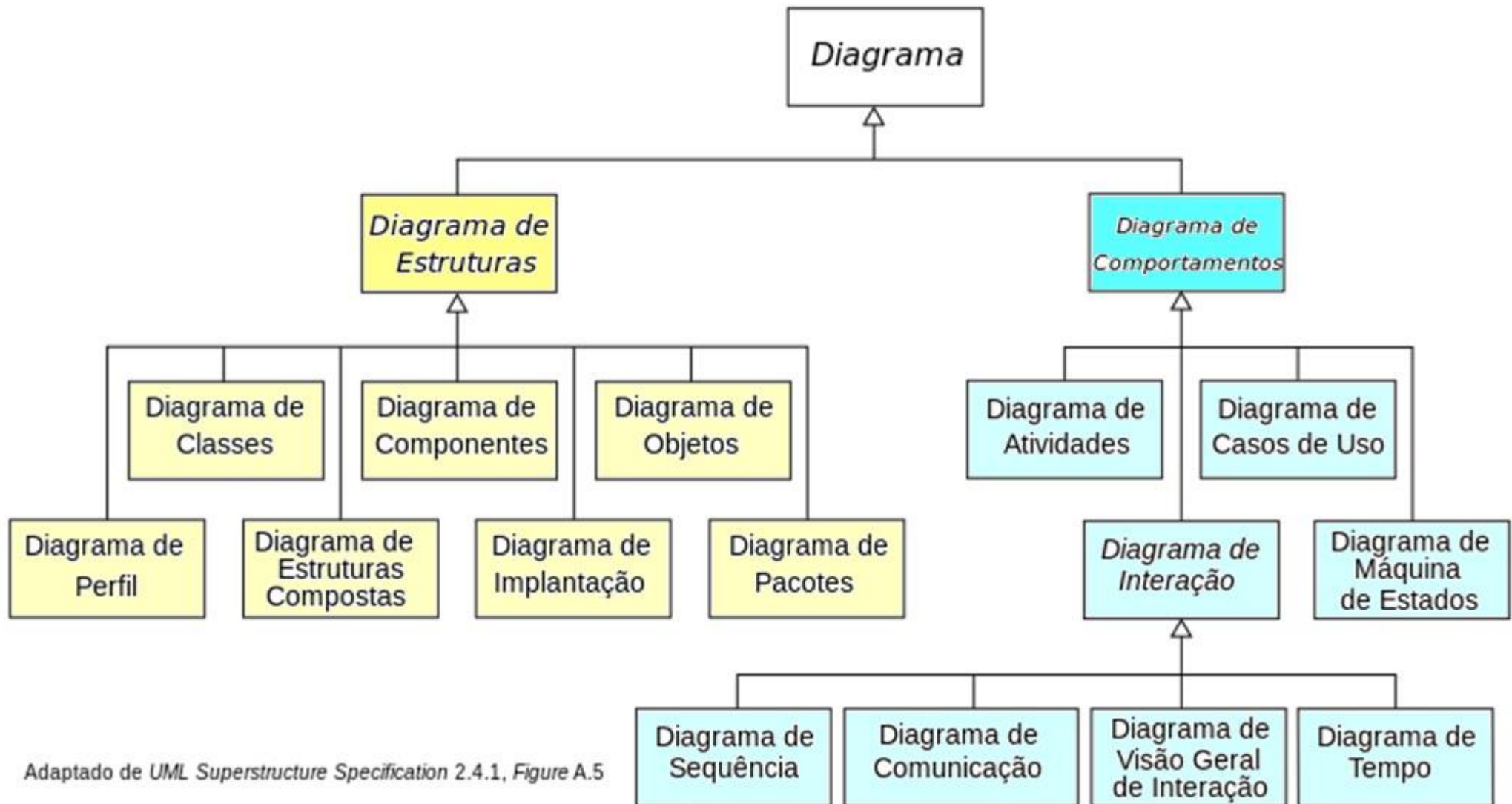
Visão	Descrição
Implementação	<ul style="list-style-type: none">• focaliza os artefatos físicos para efetiva montagem do sistema• são abordados os componentes e outros arquivos que servem para montagem do sistema• diagramas utilizados:<ul style="list-style-type: none">• diagramas de componentes• diagramas de interação• diagramas de atividades• diagramas de gráficos de estado

UML – Unified Modeling Language

■ Visões da Arquitetura de um Sistema

Visão	Descrição
Implantação	<ul style="list-style-type: none">• focaliza os nós que formam a topologia de hardware em que o sistema será executado• diagramas utilizados:<ul style="list-style-type: none">• diagramas de implantação• diagramas de interação• diagramas de atividades• diagramas de gráficos de estado

UML – Unified Modeling Language



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

UML – Unified Modeling Language

■ Diagrama de Caso de Uso

- Facilitam o entendimento de um sistema mostrando a sua “visão externa”
- São usados para modelar o contexto de um sistema, subsistema ou classe
- Uma das maneiras mais comuns de definir e documentar os requisitos do sistema (as funcionalidades)
- Representa os **requisitos funcionais** do sistema
- Delimitam o Sistema

UML – Unified Modeling Language

■ Utilidades do Diagrama de Caso de Uso

Equipe de clientes (validação)

- entendem o que o sistema deverá fazer
- aprovam o que o sistema deverá fazer

Equipe de desenvolvedores

- Ponto de partida para refinar requisitos de software.
- Designer (projetista): encontrar classes
- Testadores: usam como base para casos de teste

UML – Unified Modeling Language

■ Diagrama de Caso de Uso - ATORES

- Um sistema será descrito através de vários Casos de Uso que são executados por um número de **atores**
- Atores constituem as entidades do ambiente do sistema
- São pessoas ou outros subsistemas que interagem com o sistema em desenvolvimento
- Atores são externos ao sistema
- Primeiro é preciso identificar os usuários do sistema, que serão chamados de atores.
- Um ator é um tipo ou categoria de usuário.

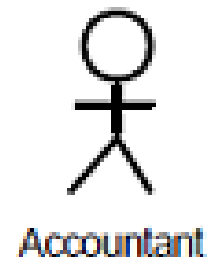
UML – Unified Modeling Language

■ EXEMPLO DE ATORES

- Cargos (Colaborador, Cliente, Gerente, Almoxarife, Vendedor, etc);
- Organizações (Empresa Fornecedora, Agência de Impostos, Administradora de Cartões, etc);
- Outros sistemas (Sistema de Cobrança, Sistema de Estoque de Produtos, etc).
- Equipamentos (Leitora de Código de Barras, Sensor, etc.)

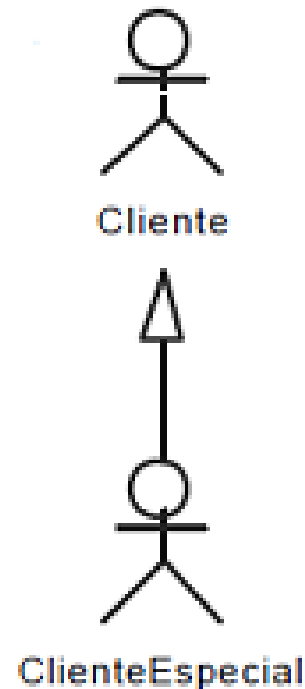
UML – Unified Modeling Language

- Diagrama de Caso de Uso – ATORES
 - Um ator é alguém ou algum outro sistema que deve interagir com o sistema em desenvolvimento



UML – Unified Modeling Language

- Diagrama de Caso de Uso - ATORES
 - É possível definir tipos gerais de atores e especializá-los usando o relacionamento de especialização:



UML – Unified Modeling Language

■ Encontrando ATORES

Podem ser identificados pelas seguintes questões:

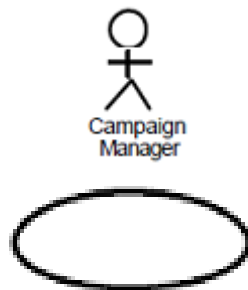
- Quem usará a funcionalidade principal do sistema (atores primários)?
- Quem precisará do auxílio do sistema para fazer suas tarefas diárias?
- Quem precisará manter, administrar, conservar o sistema funcionando (atores secundários)?
- Com que outros sistemas o sistema precisa interagir?
- Quem ou o que tem interesse nos resultados que o sistema produz?
- Não considere apenas os usuários que usam o sistema diretamente, mas todos os outros que precisam dos serviços do sistema

UML – Unified Modeling Language

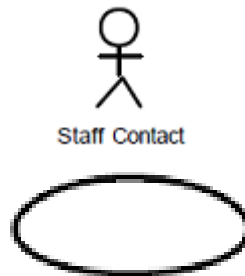
■ ATORES

Atores são examinados para determinar as suas necessidades:

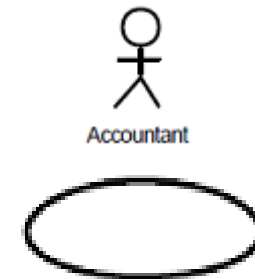
- Gerente de Campanha - adiciona um novo cliente
- Funcionário de Contato - Altera um contato do cliente
- Contador -Registra o pagamento do cliente



Add new client



Change a client contact



Record client payment

UML – Unified Modeling Language

■ CASOS DE USO

- Pode ser numa forma resumida ou numa forma mais detalhada na qual a interação entre o ator e o caso de uso é descrita passo a passo.
- Descreve interações assim como o usuário vê.
- Um documento com o fluxo de eventos é criado para cada caso de uso
- Escrito sob o ponto de vista de um ator

UML – Unified Modeling Language

■ CASOS DE USO

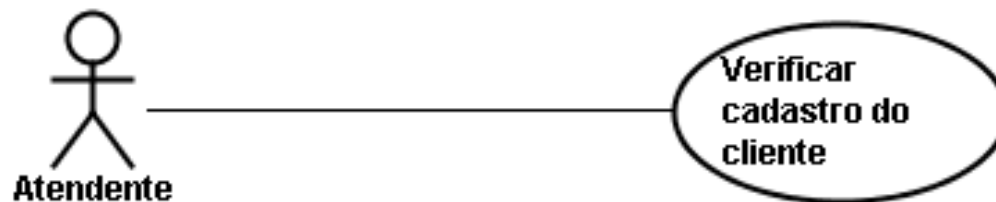
- Detalha o que o sistema deve oferecer para o ator quando o Caso de Uso é executado, entretanto, a UML não define nada acerca de como essa descrição textual deve ser construída.
- Por conta disso, há várias dimensões independentes sob as quais a descrição textual de um caso de uso pode variar:
- Grau de abstração (essencial ou real)
- Formato (contínua, tabular, numerado)
- Grau de detalhamento (sucinta ou expandida)

UML – Unified Modeling Language

- Boas práticas na documentação de casos de uso
 - Comece o nome do caso de uso com um verbo no infinitivo - para indicar um processo ou ação .
 - Não descreva como o sistema realiza internamente um passo de um caso de uso.
 - Tente dar nomes a casos de uso seguindo perspectiva do ator primário.
 - Foque no objetivo desse ator. Exemplos: Registrar Pedido, Abrir Ordem de Produção, Manter Referência, Alugar Filme, etc.
 - Tente manter a descrição de cada caso de uso no nível mais simples possível...

UML – Unified Modeling Language

- Exemplo:
- Caso de uso - Verificar cadastro do cliente



UML – Unified Modeling Language

Tabela de descrição do caso de uso:

Caso de Uso	Verificar Cadastro do Cliente
Visão Geral	O atendente consulta os dados relativos ao cliente
Objetivo	Verificar se o cliente está cadastrado e se há algum débito em seu nome
Ator Primário	Atendente
Pré-condição	O atendente precisa ter acesso a Verificar Cadastro do Cliente
Pós-condição	Dados do cliente são apresentados ao atendente ou informação de que não há o cadastro
Sequência típica de eventos	Atendente informa o nome do cliente e o sistema retorna informações sobre o mesmo. Os dados do cliente são: código, nome, endereço, telefone, pessoas autorizadas, histórico de locação
Sequência alternativa de eventos	O cliente pode não estar cadastrado, então o sistema informa ao atendente