



Engenharia de Software

Aula 3

Maristela Weinfurter Teixeira

Organização da Aula

Modelagem de software

- Aspectos essenciais
- UML
- Modelo dinâmico
- Modelo estático

Modelagem de Software Aspectos Essenciais

Modelagem de Software

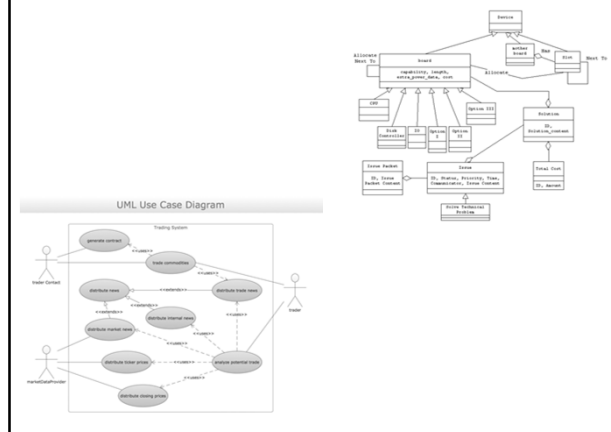
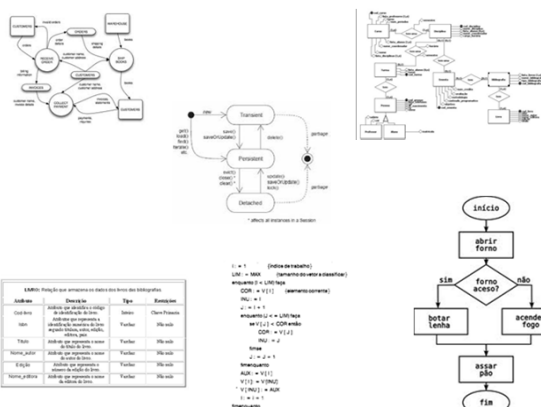
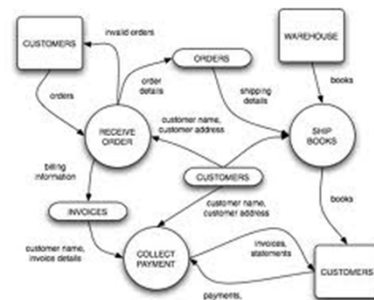
- Entende-se por **método**:
 - um caminho a ser percorrido por meio de etapas, aplicando-se um conjunto de técnicas, permitindo a construção de um software eficiente e seguro

- Métodos envolvem um conjunto de tarefas
 - Planejamento do projeto
 - Análise de requisitos
 - Projeto de estruturas de dados, arquitetura e algoritmos
 - Codificação, teste e manutenção
- Por que utilizar um MÉTODO?**

- Facilitar o treinamento de novos analistas
- Eliminar perdas na falta de controle de resultados



- Alguns modelos
 - Estruturado
 - Essencial
 - Orientado a objetos



- Estruturado
 - Visão macro — parte-se de um marco zero e se enxerga a totalidade
 - Todo-parte — fracionamento da totalidade em visões menores
 - Top-down
 - SGBDs X MER
 - Processos — DFD

- Essencial
 - Somente o que é essencial do modelo estruturado



- Orientado a objetos
 - Rumbaugh, Coad e Booch
 - Abstração
 - OMG (*Object Managment Group*)
 - UML (*Unified Modeling Language*)

Modelagem de Software

UML



Por que linguagem de modelagem?

- Linguagem **visual** para modelar sistemas orientados a objetos
- Representa conceitos por meio de linguagem visual
- Independente de linguagens de programação e de processo de desenvolvimento

- Contida na Especificação da Linguagem de Modelagem Unificada da OMG (www.omg.org)



▪ Trabalha com visões:

- casos de uso
- projeto
- implementação
- implantação
- processo

▪ Modelo dinâmico

- A UML possibilita a construção dos modelos comportamentais do sistema, os quais demonstram as funcionalidades do sistema

▪ Modelo estático

- A UML também possibilita a construção da estrutura interna do sistema pelo mapeamento para classes, objetos e relacionamentos

MODELO DINÂMICO

- Casos de uso
- Diagramas de interação
- Diagramas de sequência
- Diagramas de colaboração
- Diagramas de atividades

MODELO ESTÁTICO

- Diagramas de classe
- Diagramas de pacotes
- Diagramas de componentes
- Diagramas de implantação

Modelagem de Software

Casos de Uso (Dinâmico)



**Casos de uso**

- Representação das **funcionalidades** externamente observáveis do sistema e de elementos externos que interagem com o sistema

- Concepção: Ivar Jacobson, 1970
- Trabalhava na Ericsson
- Incorporado à UML quando se uniu a Booch e Rumbaugh
- Este diagrama direciona o restante do ciclo de vida do sistema
- Modelo centrado no usuário

- Um caso de uso representa **quem** faz **o que** (interage) com o sistema, sem considerar o comportamento interno do sistema

O diagrama de casos de uso

- Elementos importantes
 - Cenário
 - Casos de uso
 - Atores
 - Relacionamentos

O caso de uso

- Parte da funcionalidade de um sistema
- Especificação de uma sequência de interações entre o sistema e agentes externos

O ator

- *Stakeholders* ou outros sistemas externos que interagem com o sistema
- Corresponde a um papel
- Um ator pode participar de diversos casos de uso



O cenário

- Descrição de uma das maneiras pelas quais um caso de uso pode ser realizado

O relacionamento

- Casos de uso e atores não existem sozinhos
- Tipos: comunicação, inclusão, extensão e generalização

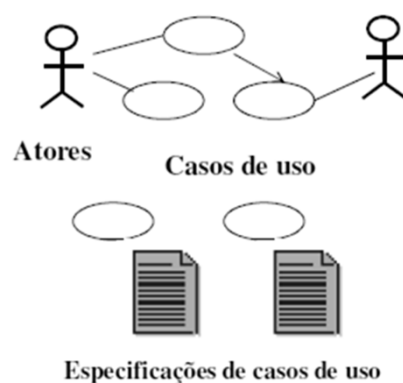
O diagrama de casos de uso

- Descreve o que acontece dentro do sistema
- Ajuda na comunicação entre *stakeholders* e desenvolvedores
- Demonstra as funcionalidades do sistema
- Captura o comportamento do sistema

Especificação de casos de uso

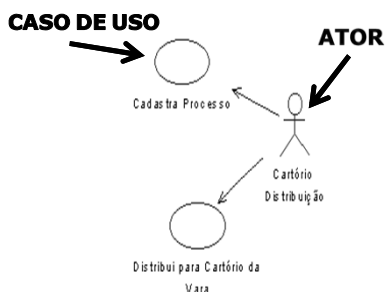
- Identificador do caso de uso
- Breve descrição
- Ator
- Prioridade
- Requisitos não funcionais associados

- Pré-condições
- Pós-condições
- Fluxo de eventos principal
- Fluxos secundários: alternativos e de exceção
- Interfaces associadas





Cenário 1: distribuição de processos



Espec. cenário 1: distribuição de processos

1. **Caso de uso:** cadastra processo

2. **Descrição:** início do processo pelo cadastramento da petição e geração automática de um número de processo jurídico. O advogado, por meio de sua digital, dá entrada no protocolo, entrega a guia de custas devidamente paga e autenticada. A guia é lida através do código de barras. E o sistema gera um número de processo e designa uma vara

3. **Ator:** cartório de distribuição

4. **Prioridade:** Nível 1 (de 5 níveis)

5. **Requisitos não-funcionais associados:** leitor de código de barras e leitor biométrico.

6. **Pré-condição:** guia de recolhimento de custas deve estar paga e autenticada.

7. **Pós-condição:** geração do número do processo e encaminhamento para vara sorteada

Modelagem de Software

Classes

(Estático)

Modelagem de Software



Diagrama de classes

- Permite a compreensão da estrutura interna para que as funcionalidades externas sejam produzidas
- Modelo estático porque demonstra a estrutura das classes de objetos e as relações entre elas



- Tipos
 - Domínio
 - Especificação
 - Implementação

- **Domínio**: representa classes do domínio do negócio. Construído na fase de **análise**
- **Especificação**: extensão do modelo de domínio. Adição de detalhes específicos à solução escolhida. Construído no **projeto**

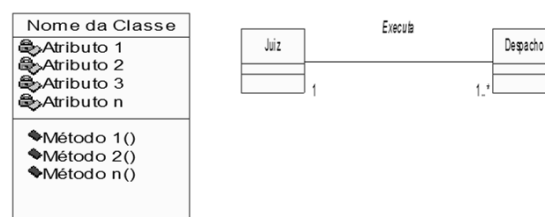
- **Implementação**: extensão do modelo de especificação. Classes já descritas na linguagem de programação. Construído na **implementação** do software

- Elementos importantes
 - Classe
 - Atributo
 - Método
 - Relacionamento

Identificando classes

- Dirigido por dados: ênfase na identificação da estrutura dos conceitos relevantes
- Dirigido por responsabilidades: identificado nas responsabilidades que cada classe deve ter dentro do sistema

Diagrama de classes

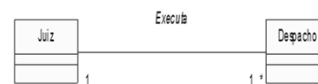




Relacionamentos

- Associação
- Agregação
- Generalização

▪ Associação



Apenas um	1
Zero ou Muitos	0..*
Um ou Muitos	1..*
Zero ou Um	0..1
Intervalo Específico	1..1 _s

▪ Agregação



▪ Generalização

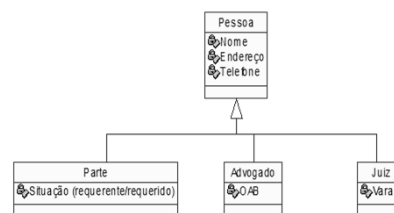
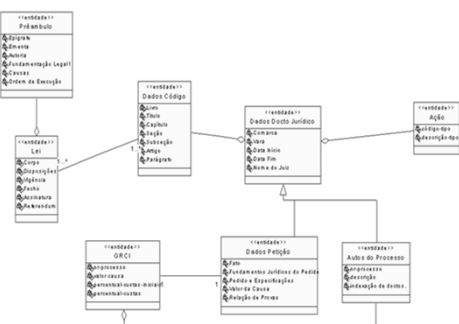


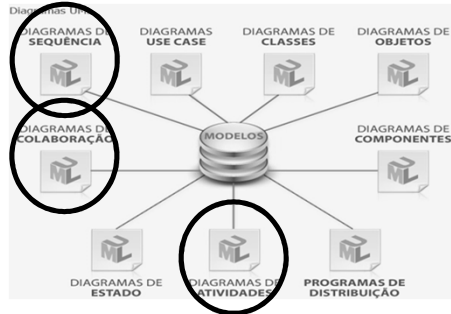
Diagrama de classes do domínio



Modelagem de Software Sequência, Colaboração e Atividades (Dinâmico)



Modelagem de Software



Diagramas de interação

- Consolidação do entendimento dos aspectos dinâmicos do sistema iniciado nos diagramas de casos de uso
- Interação por meio da troca de mensagens

- Demonstra a **comunicação** entre ator e objetos dentro do sistema
- São dois tipos:
 - sequência
 - colaboração

- Elementos importantes
 - Conexões
 - Mensagens
 - Fluxo de controle

Diagrama de sequência

- Troca de mensagens dentro de uma **linha do tempo**
- Elementos
 - Atores
 - Objetos
 - Classes
 - Linhas da vida
 - Loop
 - Condição
 - Recursividade

Diagrama de sequência

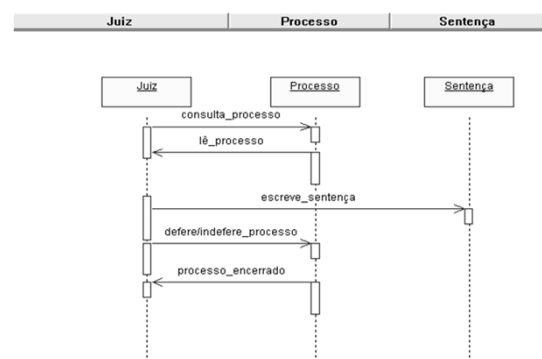




Diagrama de colaboração

- Demonstra a troca de mensagens entre os objetos com foco nas mensagens trocadas. Similar ao diagrama de sequência, porém são adicionados setas e rótulos de mensagens nas ligações entre objetos

Elementos

- Atores
- Objetos
- Classes
- Linhas da vida
- *Loop*
- Condição
- Recursividade
- Rótulos de mensagens

Diagrama de colaboração

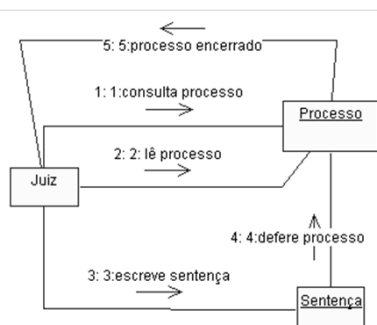


Diagrama de atividades

- Especificam o comportamento de uma entidade (objeto ou **classe**)
- Tipo especial de diagrama de estados, no qual representam-se os estados de uma atividade. Orientado a fluxos de controle

- Estado da ação
- Estado da atividade
- Estados inicial, final e condição
- Sincronização
- Raias
- Fluxo de objeto

Diagrama de atividades





Referências de Apoio

- Page-Jones, M. **Fundamentos do desenho orientado a objeto com UML**. São Paulo: Pearson, 2001.
- Pfleeger, S. L. **Engenharia de software: teoria e prática**. 2. ed. São Paulo: Prentice Hall, 2004.

- Presman, R. **Engenharia de Software**. 7. ed. Porto Alegre: Bookman, 2011.
- Sommerville, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.