Métodos de Desenvolvimento de Software

Modelos de Design de Software

Material Baseado nos livros:

- Swebok v3
- Introdução a Arquitetura e Design de Software.
- Software Engineering / Ian Sommerville. 9th ed.

UML (Unified Modeling Language)

- UML é uma linguagem visual utilizada para especificar, construir e documentar os artefatos dos sistemas
- Ou seja, é uma notação diagramática para desenhar ou apresentar figuras relacionadas a software, principalmente, orientadas a objetivos
- ► A UML se tornou uma linguagem de modelagem padrão para modelos orientada a objetos
- ▶ UML Possui vários tipos de diagramas e, dessa forma, apoia a criação de tipos de diferentes modelos de sistema.

UML: Perspectivas

Conceitual:

 Os diagramas são interpretados como descrevendo coisas em uma situação do mundo real ou domínio de interesse

Especificação (software):

▶ Descrevem abstrações de software ou componentes, com especificações e interfaces, mas nenhum comprometimento com uma implementação particular (por exemplo, C# ou Java)

Implementação (software):

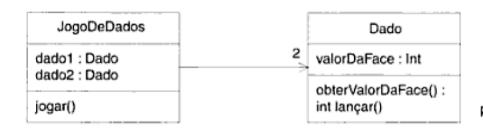
▶ Descrevem implementações em uma tecnologia em particular (como, Java)

UML: Perspectivas (cont.)



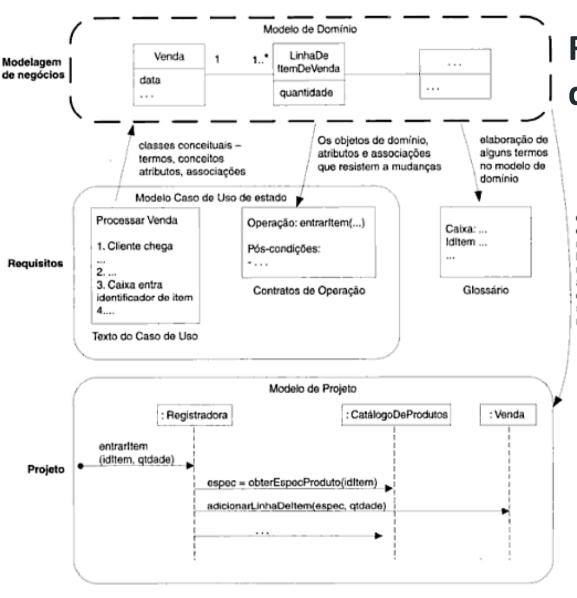
Perspectiva conceitual (modelo do domínio)

Notação do esboço do Diagrama de Classe UML usada para visualizar os conceitos do mundo real



Perspectiva de especificação ou implementação (diagrama de classe de projeto) Notação do Diagrama de

Notação do Diagrama de Classe UML bruto usado para visualizar os elementos de software



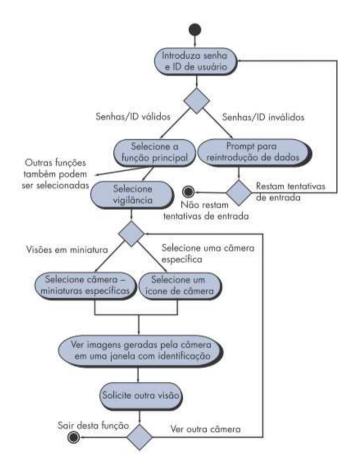
Relação entre modelos de níveis diferentes

classes conceituais no domínio inspiram os nomes de algumas classes de software no projeto

Exemplos de Diagramas UML

- Diagramas de atividades:
 - Mostram as atividades envolvidas em um processo ou no processamento de dados.
- Diagramas de casos de uso:
 - Mostram as interações entre um sistema e seu ambiente.
- Diagramas de sequência:
 - Mostram as interações entre os atores e o sistema, e entre os componentes do sistema.
- Diagramas de classe:
 - Mostram as classes de objeto no sistema e as associações entre elas.
- Diagramas de estado:
 - ▶ Mostram como o sistema reage aos eventos internos e externos.

Diagrama de Atividades (exemplos)



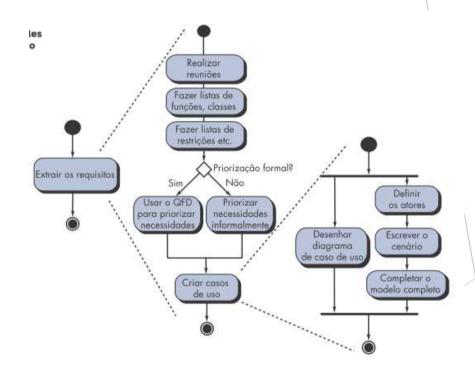
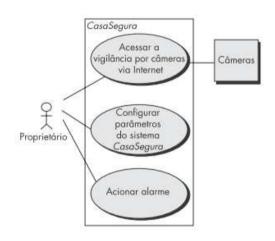


Diagrama de Casos de Uso (exemplos)



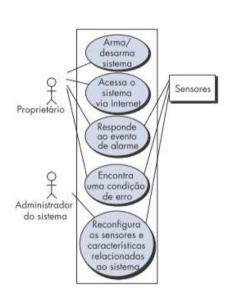


Diagrama de Sequencia (exemplos)

Cenário simples de Processar Venda com pagamento em dinheiro

- O Cliente chega a um ponto de pagamento equipado com um PDV, trazendo vários bens ou serviços que deseja comprar.
- 2. O Caixa inicia uma nova venda.
- 3. O Caixa digita o identificador do item.
- O Sistema registra a linha de item da venda e exibe a descrição, o preço do item e o total parcial corrente.
- O caixa repete os passos 3 e 4 até que indique ter terminado.
- O Sistema apresenta o total, com os impostos já calculados.
- O Caixa informa o total ao Cliente e solicita o pagamento.
- O Cliente paga e o Sistema trata o pagamento.

Ι.

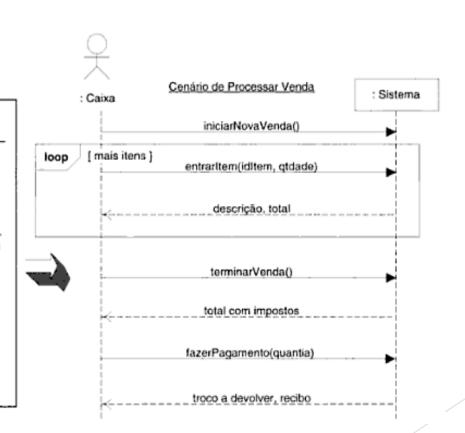
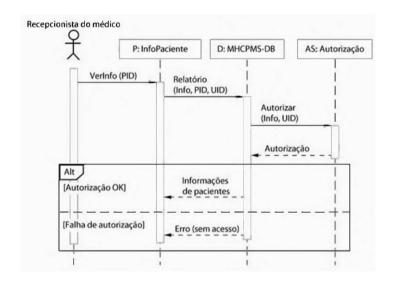


Diagrama de Sequencia (exemplos)



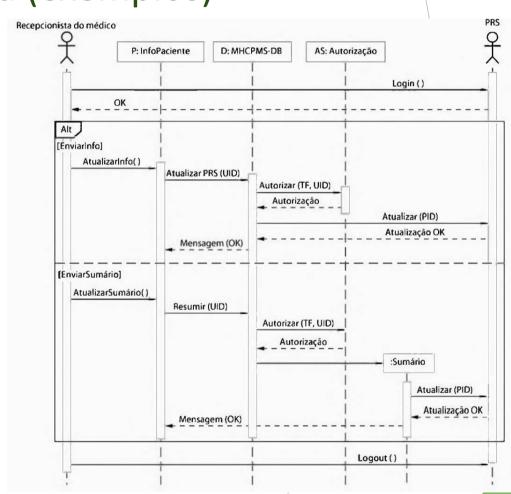
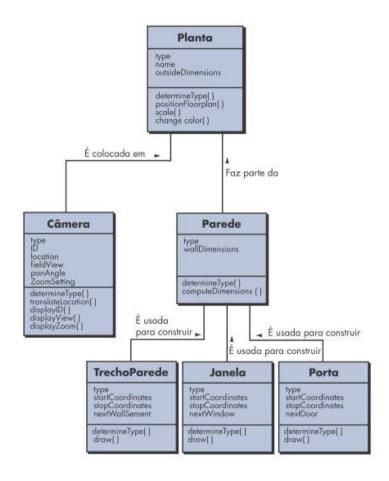
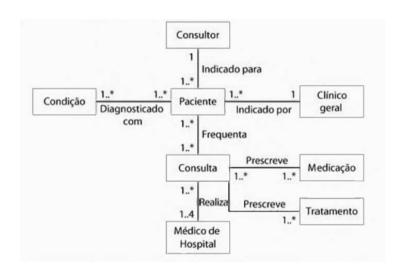
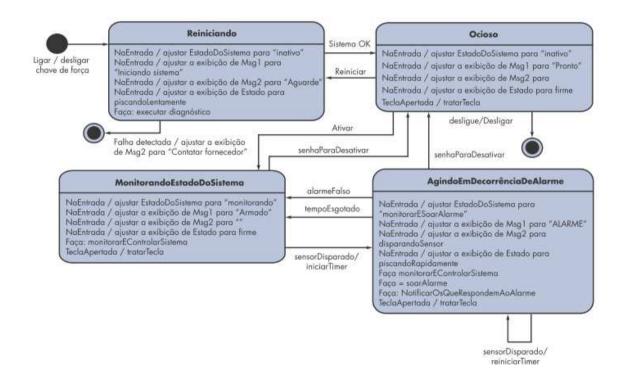


Diagrama de Classes (Exemplos)





Diagramas de Estados (Exemplos)



Métodos de Desenvolvimento de Software

Modelos de Design de Software



Material Baseado nos livros:

- Swebok v3
- Introdução a Arquitetura e Design de Software.
- Software Engineering / Ian Sommerville. 9th ed.