



Análise de Sistemas

Aula 6

Prof. Emerson Klisiewicz

Contextualização

Aula 6

- Diagrama de Casos de Uso
- Diagrama de Classes e Diagrama de Objetos
- Diagrama de Sequência
- Diagrama de Máquina de Estados

Sucesso

- Clientes satisfeitos
- Eles estão satisfeitos quando você:
 - ✓ atende às expectativas
 - ✓ entrega no prazo
 - ✓ entrega tudo dentro do orçamento

- E para isso acontecer precisamos estar auxiliados por uma boa metodologia. A UML nos ajuda nisso!

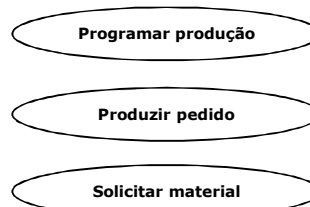
Instrumentalização



Diagrama de Casos de Uso

- Descreve o que o sistema faz do ponto de vista do observador externo
- Ajuda a esclarecer os requisitos do sistema
- Ajuda a dividir o desenvolvimento do sistema em tarefas

- **Caso de uso:** representa as funcionalidades que o sistema disponibiliza aos usuários



- **Atores:** são diferentes usuários que operam o sistema. Sistemas externos que interagem com outro

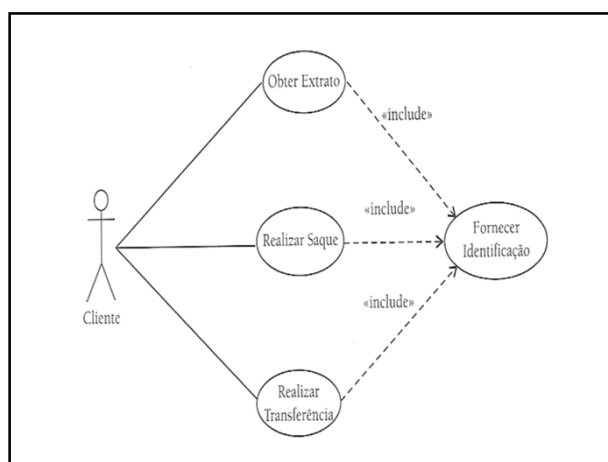


- **Associação:** representa a comunicação entre o ator e o caso de uso. Também existem associações entre casos de uso



▪ Inclusão

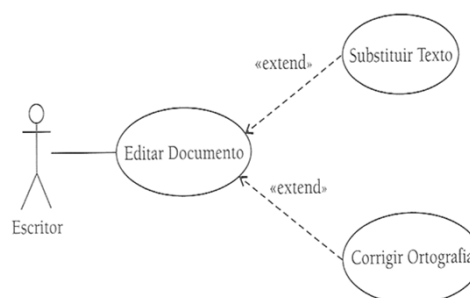
- Quando dois ou mais casos de uso incluem uma sequência comum de interações, tal sequência pode ser descrita em outro caso de uso
- Somente entre casos de uso. Vários casos de uso podem incluir o comportamento de outro em comum





▪ Extensão

- Somente entre casos de uso.
- Serve pra modelar situações em que diferentes sequências de interações podem ser inseridas em um mesmo caso de uso
- Essas sequências representam um comportamento eventual



▪ Generalização

- Pode existir entre dois casos de uso ou entre dois atores. Permite que um caso de uso (ou um ator) herde o comportamento de outro caso de uso (ou ator)
- É recomendado que o caso de uso pai sempre seja abstrato, para evitar problemas na especificação

- O caso de uso pai é utilizado apenas para representar a natureza dos casos de uso filhos

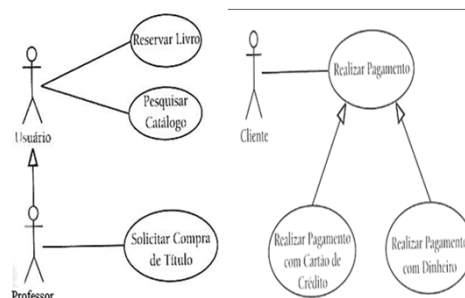


Diagrama de Classe

- Largamente utilizado
- Descreve os diferentes tipos de objeto e os relacionamentos entre eles

- Composto por:
 - classes
 - associações — relacionamento entre as classes

**▪ Atributos e métodos**

- Atributos — armazenam os “dados” dos objetos
- Métodos — funções que uma instância da classe pode executar

▪ Visibilidade:

- “+” = visibilidade pública — pode ser utilizado por qualquer classe
- “#” = visibilidade protegida — somente a própria classe ou suas subclasses podem ter acesso
- “-” = visibilidade privada — somente a classe possuidora do atributo poderá utilizá-lo

▪ Relacionamentos

- Associações: vínculo que ocorre normalmente entre duas classes, entre uma classe com ela mesma e entre várias classes

- Agregação: tipo especial de associação em que tenta demonstrar que as informações de um objeto precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe

- Composição: variação da associação de agregação. Vínculo mais forte entre objetos-todo e objetos-parte
 - ✓ Objetos-parte têm de pertencer exclusivamente a um objeto-todo

- Especialização: tipo especial de relacionamento. Identificar classes-mãe (gerais) e classes-filha (especializadas)
- Dependência: demonstra certo grau de dependência de uma classe a outra. A mudança numa classe deverá refletir na outra



- Classe associativa: quando da ocorrência de associações que possuem multiplicidade em todas as suas extremidades

- Complemento do diagrama de classes
- Exibe os valores armazenados pelos objetos de um diagrama de classes

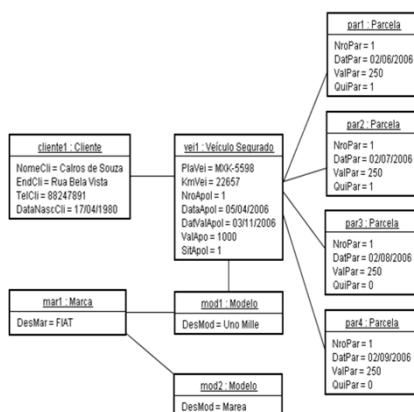


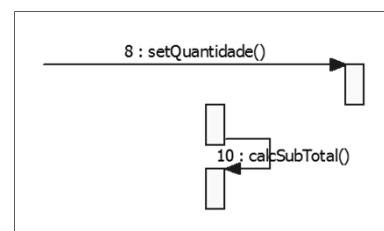
Diagrama de Sequência

- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Baseia-se em um caso de uso
- Costuma identificar o evento gerador do processo modelado, bem como o ator responsável por este evento
- Dá ênfase à sequência de mensagens

- Objetos: representa uma instância de uma determinada classe



- Mensagens: representa troca de mensagens entre os objetos





- Fragmentos: servem para separar blocos de mensagens condicionadas a teste ou laço

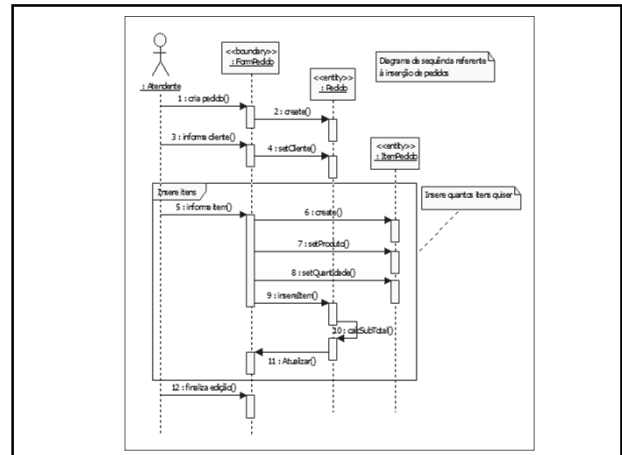
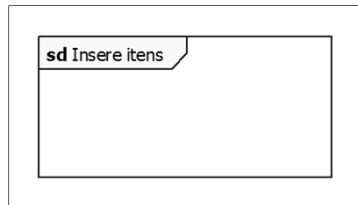
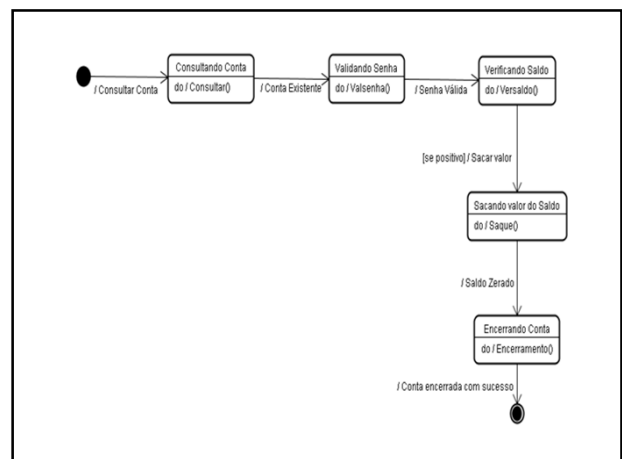


Diagrama de Estados

- Exibe os possíveis estados de um objeto
- Exibe também as transições que causam a mudança de estado



Aplicação

Diagrama de Casos de Uso

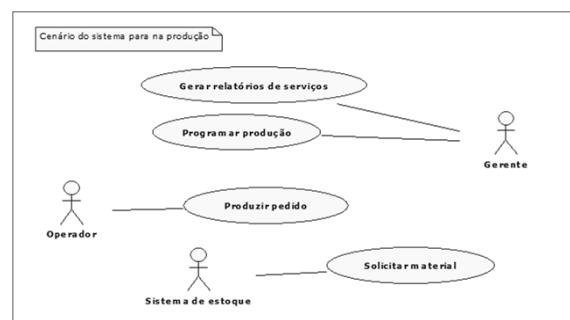




Diagrama de Classe

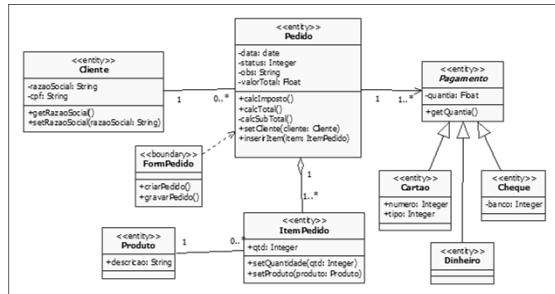


Diagrama de Sequência

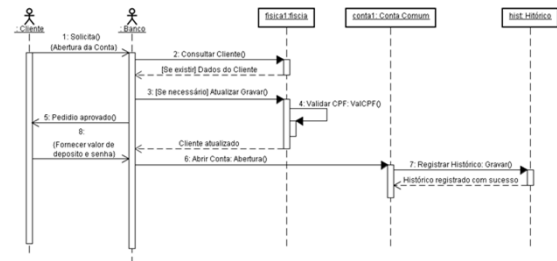
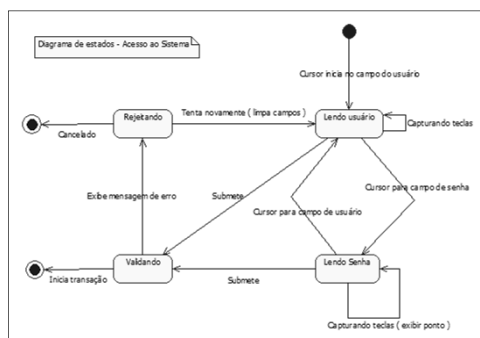


Diagrama de Estados



Síntese

Pontos-chave

- Um modelo pode ser visto como uma representação idealizada de um sistema a ser construído, uma simplificação da realidade que nos ajuda a entender um problema complexo

- A modelagem de sistemas de software consiste na utilização de notações gráficas e textuais para construir modelos que representam o essencial de um sistema
- Ela ajuda no gerenciamento da complexidade inerente ao desenvolvimento de software



- Ajuda também na comunicação entre as pessoas envolvidas e na previsão do futuro comportamento do sistema