

# Modelagem de Sistemas

Douglas O. Cardoso  
douglas.cardoso@cefet-rj.br  
docardoso.github.io



# Objetivos da aula

- Entender a utilidade de modelos gráficos para descrever software
- Diferenciar perspectivas de contexto, interação, estrutura e comportamento
- Conhecer alguns tipos de diagramas da UML

# Roteiro

- 1 Introdução
- 2 Modelos de Contexto
- 3 Modelos de Interação
- 4 Modelos Estruturais
- 5 Modelos Comportamentais

# O que é modelagem de sistemas?

É o processo de desenvolvimento de modelos abstratos de um sistema.

- Diversos modelos, pontos de vista e perspectivas
- Geralmente (mas não obrigatoriamente), modelos gráficos
- Mil e uma utilidades
  - Engenharia de requisitos
  - Arquitetura e design estrutural
  - Documentação do produto
  - Geração automatizada de software
  - **Diferentes quanto a corretude e completude!**

# Modelagem e Abstração

- Modelos não incluem detalhes, e isso pode ser bom
- Representação (e.g., tradução) vs Abstração (e.g., resumo)

# Algumas Perspectivas de Modelagem

- Externa: modelagem do contexto ou ambiente do sistema
- Interação: entre o sistema e o ambiente, ou entre diferentes componentes do sistema
- Estrutural: organização do sistema, ou estrutura dos dados que serão processados
- Comportamental: dinâmica do sistema, resposta a eventos

# Roteiro

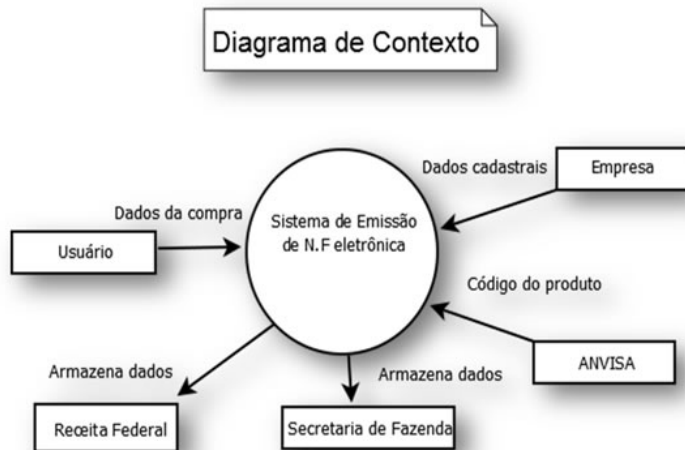
- 1 Introdução
- 2 Modelos de Contexto
- 3 Modelos de Interação
- 4 Modelos Estruturais
- 5 Modelos Comportamentais

# Noções Básicas

- São usados para ilustrar o contexto operacional de um sistema
- Ou seja, o que está faz fronteira com o sistema
- Estabelecer os limites do sistema pode ser subjetivo, ou envolver questões além de ES
  - Logística
  - Política
  - Finanças



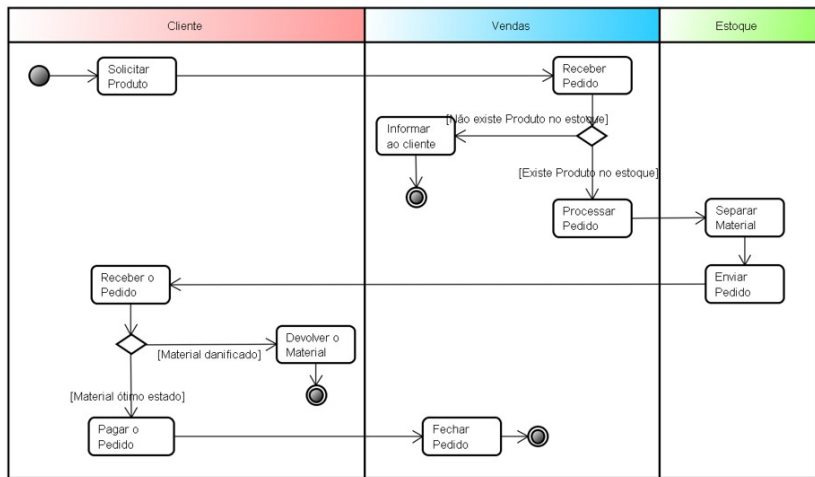
# Exemplo



# Perspectiva de Processos

- Diagramas de Contexto mostram relacionamentos entre o sistema e vizinhos
- Porém, não é indicado como esses relacionamentos ocorrem:
  - Compartilhamento de dados?
  - Prestação ou uso de serviços?
- Diagramas de processos cumprem o papel de descrever tais ações conjuntas

# Exemplo: diagrama de processo (ou atividade, DA)



2

<sup>2</sup>Fonte: [alexpagernet.blogspot.com.br](http://alexpagernet.blogspot.com.br)

# Roteiro

- 1 Introdução
- 2 Modelos de Contexto
- 3 Modelos de Interação**
- 4 Modelos Estruturais
- 5 Modelos Comportamentais

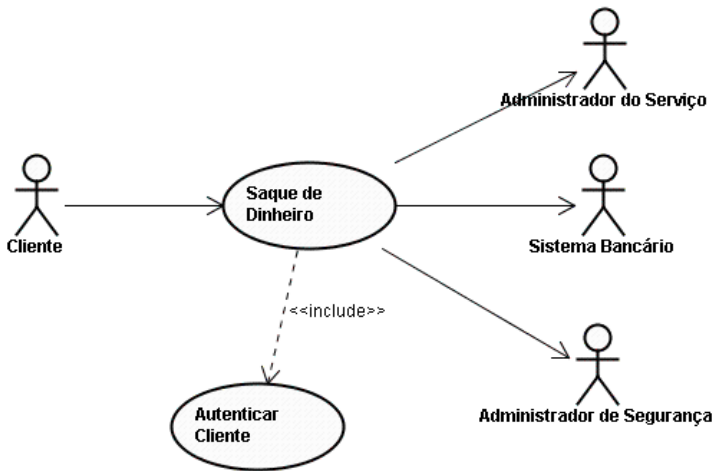
# Noções Básicas

- Úteis para a identificação de requisitos do usuário
- Modelar interações sistema-sistema podem explicitar futuros problemas de comunicação
- Interações entre componentes ajudam na análise de eficiência e confiabilidade do sistema

# Casos de Uso (Use Case, UC)

- Uma das principais ferramentas para elicitación de requisitos
- Um caso de uso, uma tarefa, um ou mais atores
- Representação gráfica, em mais alto nível
- Representação textual, com mais detalhes: atores, descrição, dados, estímulo, resposta, comentários etc

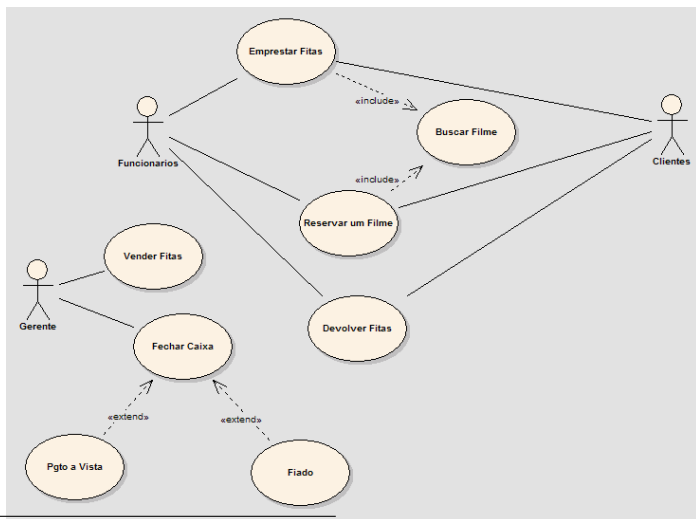
## Exemplo: diagrama de caso de uso



3

<sup>3</sup>Fonte: [helpdesenvolvedor.blogspot.com.br](http://helpdesenvolvedor.blogspot.com.br)

## Exemplo: diagrama de caso de uso (2)



4

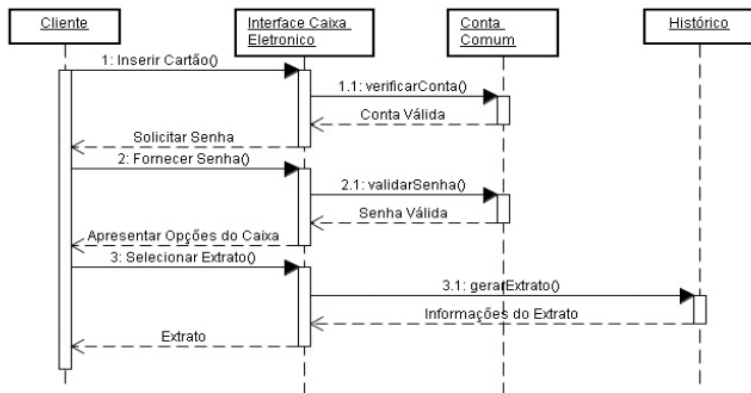
<sup>4</sup>Fonte: <http://blogdoabu.blogspot.com.br>



# Diagrama de Sequência

- Modelam interações entre atores e objetos internos ao sistema
- Detalha o que é apresentado num UC
- Função similar ao DA com relação ao diagrama de contexto

# Exemplo: diagrama de sequência



5

<sup>5</sup>Fonte: <http://pt.slideshare.net/ItaloCosta4/>

# Roteiro

- 1 Introdução
- 2 Modelos de Contexto
- 3 Modelos de Interação
- 4 Modelos Estruturais**
- 5 Modelos Comportamentais

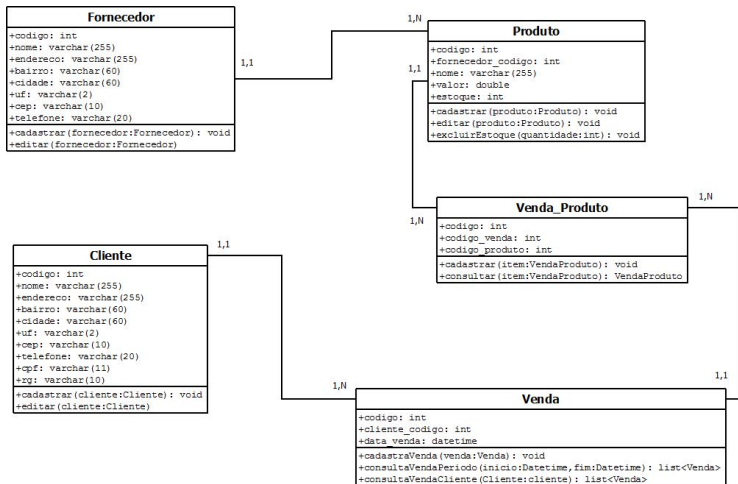
# Noções Básicas

- Apresentam a organização dos componentes do sistema, seus relacionamentos
- Modelos estáticos: estrutura arquitetural do sistema
- Modelos dinâmicos: organização dos componentes durante o funcionamento do sistema

# Diagrama de Classes

- Voltado para sistemas OO
- Apresenta as classes do sistema e suas associações
- Pode também incluir atributos e operações (i.e., métodos) das classes

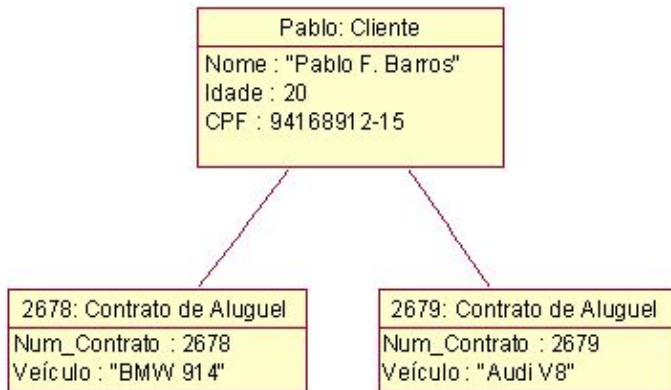
# Exemplo: diagrama de classes



6

<sup>6</sup>Fonte: <http://eufacoprogramas.com/>

## Exemplo: diagrama de objetos



7

---

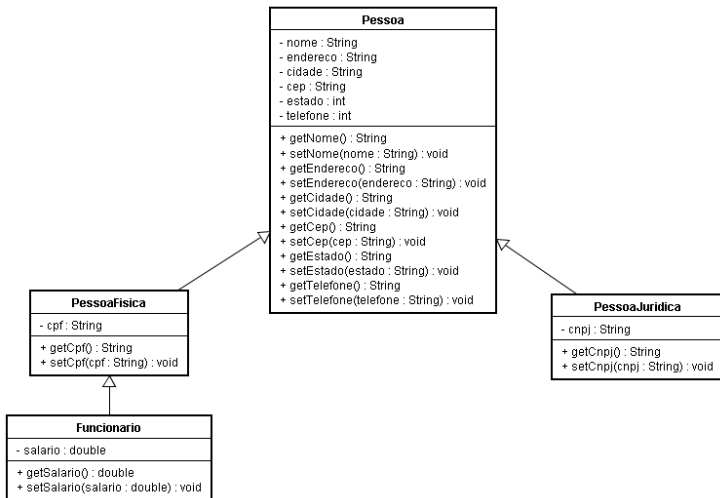
<sup>7</sup>Fonte: <http://eufacoprogramas.com/>

# Generalização

- Uma técnica básica para gerenciar complexidade
- Descrever classes específicas baseando-se em classes mais gerais em comum
- No contexto de programação, é conhecido como Herança
- Facilita lidar com alterações no sistema (requisitos, arquitetura)



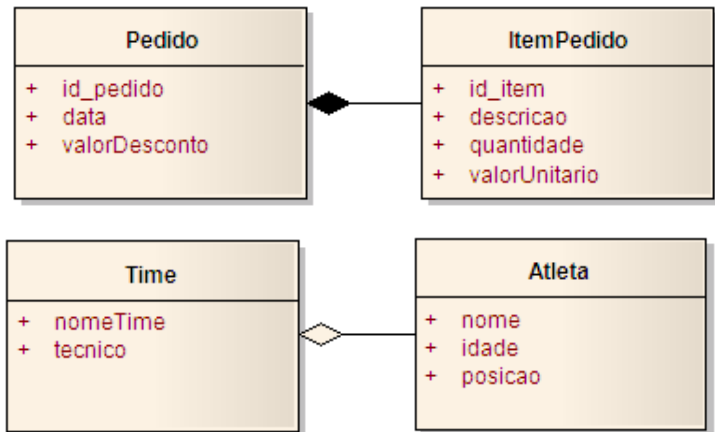
# Exemplo: generalização



# Agregação e Composição

- São tipos de relacionamento entre classes
- Envolvem uma classe (objeto-todo) que é uma coleção da outra (objeto-parte)
- Agregação: objeto-parte existe, tem sentido sem o objeto-todo
- Composição: não há razão de ser para o objeto-parte sem o objeto-todo

## Exemplo: agregação e composição



9

<sup>9</sup>Fonte: <http://imasters.com.br/>

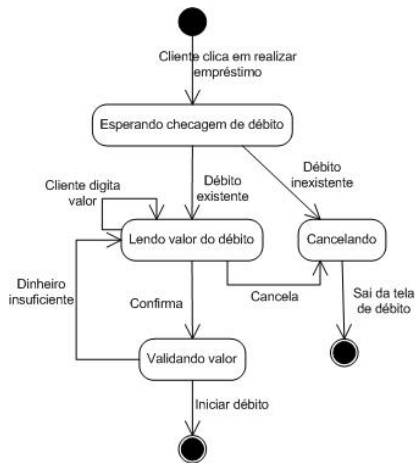
# Roteiro

- 1 Introdução
- 2 Modelos de Contexto
- 3 Modelos de Interação
- 4 Modelos Estruturais
- 5 Modelos Comportamentais**

# Noções Básicas

- Referem-se ao comportamento dinâmico do sistema durante seu funcionamento
- Mostram o que acontece (ou deveria) quando o sistema recebe algum estímulo:
  - Dados os quais o sistema espera para realizar uma ação
  - Eventos diversos, os quais podem ou não ter dados associados

## Exemplo: diagrama de estados



10

<sup>10</sup>Fonte: <http://dropsti.blogspot.com.br/>

## Trabalho 2: modelagem na prática

- Trabalho individual
- Propor um software que atenda a certa demanda real
- Modelar tal sistema, desenhando diagramas que o descrevam
- Ao menos 1 diagrama de cada uma das 4 perspectivas apresentadas
- Obviamente, mais diagramas e detalhes são bem-vindos
- Data de Entrega: 05/10/2016