

Introdução à UML: Diagrama de Casos de Uso

Prof. Me. Lucas Bruzzone

Aula 11

O que é UML?

Unified Modeling Language

Linguagem gráfica padrão para modelagem de sistemas orientados a objetos

Características:

- Padrão da indústria (OMG)
- Independente de linguagem de programação
- Múltiplas visões do sistema
- Comunicação entre equipes
- Documentação visual

Tipos de Diagramas UML

Estruturais

- Diagrama de Classes
- Diagrama de Objetos
- Diagrama de Componentes
- Diagrama de Pacotes

Comportamentais

- Diagrama de Casos de Uso
- Diagrama de Sequência
- Diagrama de Atividades
- Diagrama de Estados

Focaremos nos 3 mais importantes para análise

Objetivo

Capturar requisitos funcionais mostrando interações entre usuários (atores) e sistema

Elementos básicos:

- **Atores:** Usuários ou sistemas externos
- **Casos de uso:** Funcionalidades do sistema
- **Relacionamentos:** Conexões entre elementos
- **Fronteira do sistema:** Delimita escopo

Perspectiva: Externa - como usuário vê o sistema

Definição

Entidades externas que interagem com o sistema

Tipos:

- **Primários:** Iniciam casos de uso
- **Secundários:** Reagem aos casos de uso
- **Pessoas:** Usuários humanos
- **Sistemas:** Outros sistemas

Exemplos:

- Cliente, Administrador, Vendedor
- Sistema de Pagamento, Base de Dados

Notação: Boneco ou retângulo com <<actor>>

Definição

Funcionalidade específica que o sistema oferece para atingir objetivo do ator

Características:

- Iniciados por ator
- Geram valor observável
- Sequência completa de ações
- Nível de abstração adequado

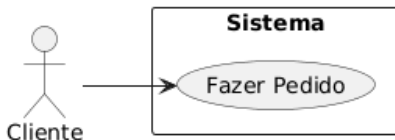
Nomenclatura:

- Verbos no infinitivo
- "Fazer Login", "Processar Pedido"
- Evitar detalhes técnicos

Notação: Elipse com nome do caso de uso

Associação (Comunicação):

- Liga ator a caso de uso
- Linha simples
- Indica que ator participa do caso de uso



Dicas:

- Ator deve ter pelo menos uma associação
- Caso de uso deve ter pelo menos um ator primário

Include (Inclusão):

- Caso de uso sempre executa outro
- Funcionalidade comum reutilizada
- Seta tracejada com <<include>>

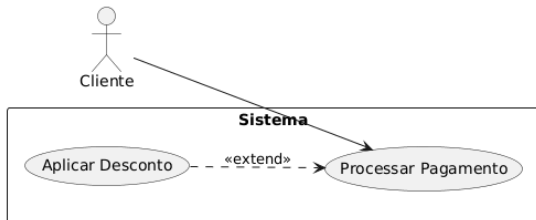


Quando usar:

- Evitar duplicação de funcionalidade
- Separar comportamento obrigatório

Extend (Extensão):

- Caso de uso opcionalmente executa outro
- Comportamento condicional
- Seta tracejada com <<extend>>



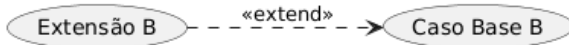
Quando usar:

- Funcionalidade opcional ou casos especiais
- Diferentes fluxos baseados em condições

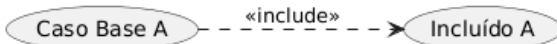
Atenção: Direção oposta ao include!

Include vs. Extend

Extend - Condicional



Include - Sempre Executa



Aspecto	Include	Extend
Execução	Sempre	Condicional
Direção	Base → Incluído	Extensão → Base
Propósito	Reutilização	Variação

Elementos da especificação:

- **Nome:** Identificação única
- **Ator primário:** Quem inicia
- **Objetivo:** Meta a ser alcançada
- **Pré-condições:** Estado inicial necessário
- **Fluxo principal:** Sequência normal de passos
- **Fluxos alternativos:** Variações e exceções
- **Pós-condições:** Estado final do sistema

Exemplo de Especificação

Caso de Uso: Fazer Login

Ator Primário: Usuário

Objetivo: Autenticar usuário no sistema

Pré-condições: Usuário possui conta válida

Fluxo Principal:

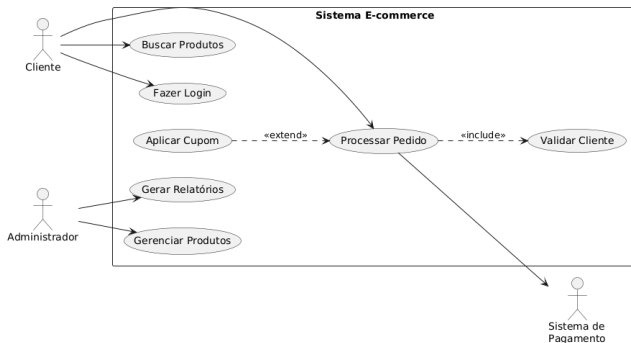
- 1 Usuário informa email e senha
- 2 Sistema valida credenciais
- 3 Sistema redireciona para página principal

Fluxo Alternativo - Credenciais Inválidas:

- 1 Sistema exibe mensagem de erro
- 2 Retorna ao passo 1

- **Representam papéis, não pessoas específicas**
 - Bom: "Cliente", "Administrador"
 - Ruim: "João", "Maria"
- **Nomes claros e significativos**
- **Evitar atores muito genéricos ("Usuário")**
- **Considerar herança entre atores**
 - "Funcionário" pode ser pai de "Vendedor"
- **Incluir sistemas externos quando relevantes**

Generalização de Atores

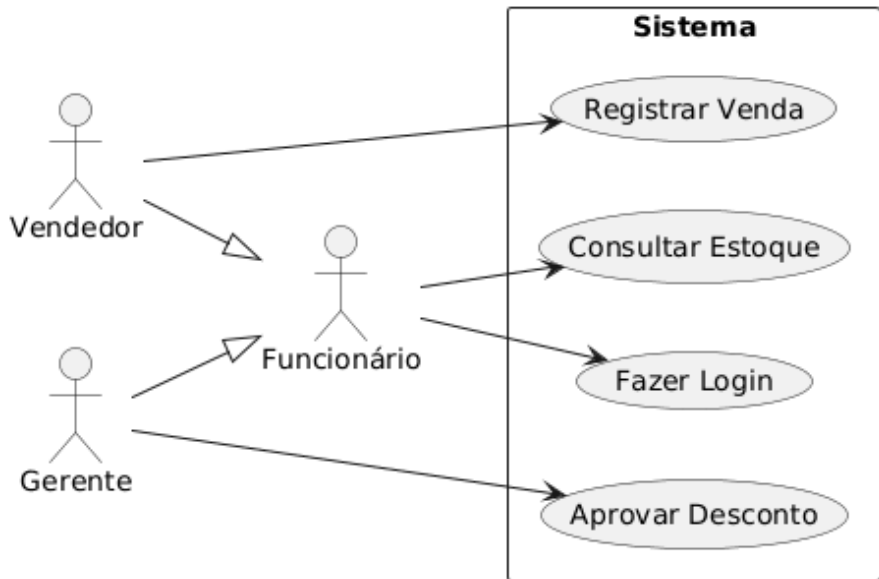


Herança: Atores especializados herdam associações do ator genérico

- **Nível de granularidade adequado**
 - Nem muito detalhado, nem muito abstrato
- **Casos de uso geram valor observável**
- **Verbos no infinitivo**
- **Foco no "o que", não no "como"**
- **Evitar casos de uso CRUD simples**
 - "Gerenciar Produtos" melhor que "Criar/Ler/Atualizar/Deletar Produto"
- **Entre 5-50 casos de uso por sistema**

- **Casos de uso muito técnicos**
 - "Conectar ao Banco" → deve ser transparente
- **Confundir com diagrama de fluxo**
 - Não mostrar sequência temporal
- **Muitos relacionamentos include/extend**
 - Complica desnecessariamente
- **Atores dentro do sistema**
 - Atores são sempre externos
- **Casos de uso sem valor para ator**

Exemplo Prático - E-commerce



Benefícios dos Diagramas de Casos de Uso

- **Comunicação:** Linguagem comum entre stakeholders
- **Escopo:** Define fronteiras do sistema
- **Planejamento:** Base para estimativas e cronograma
- **Testes:** Casos de teste derivados dos casos de uso
- **Rastreabilidade:** Liga requisitos à implementação
- **Validação:** Cliente pode verificar completude

Atividade

Baseado no trabalho do primeiro bimestre, desenvolva um diagrama de casos de uso da sua aplicação

Requisitos:

- Identificar todos os atores (humanos e sistemas)
- Listar os principais casos de uso (5-15 casos)
- Definir associações entre atores e casos de uso
- Aplicar relacionamentos include/extend quando apropriado
- Delimitar a fronteira do sistema

Entrega: Diagrama visual + Especificação de 2 casos de uso

Diagramas de Classe: Conceitos e Notação

Estudaremos a estrutura estática do sistema através de classes, atributos e relacionamentos