

# Modelagem de Casos de Uso

**Prof. Vítor Souza**

Análise e Projeto Orientado a Objetos

Departamento de Informática  
Univ. Federal do Espírito Santo

11/04/2006

# Licença para uso e distribuição

Este material está disponível para uso não-comercial e pode ser derivado e/ou distribuído, desde que utilizando uma licença equivalente.



Atribuição-Uso Não-Comercial-Compatilhamento pela mesma licença, versão 2.5

<http://creativecommons.org/licenses/by-nc-sa/2.5/deed.pt>

Você pode copiar, distribuir, exibir e executar a obra, além de criar obras derivadas, sob as seguintes condições: (a) você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante; (b) você não pode utilizar esta obra com finalidades comerciais; (c) Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

# Sobre o curso

- Estes slides foram criados no Departamento de Informática da Universidade Federal do Espírito Santo (UFES) e estão disponível no seguinte endereço:

**<http://www.inf.ufes.br/~vsouza/>**

- O material usado como base foi cedido pelo professor Dr. Ricardo de Almeida Falbo, do DI/UFES:

**<http://www.inf.ufes.br/~falbo/>**

- Este curso tem como objetivo apresentar os conceitos da análise e projeto orientado a objetos a profissionais e estudantes de Engenharia de Software.

# Especificação

- Atividade da Engenharia de Requisitos;
- Elabora o produto final: Documento de Especificação de Requisitos;
- Este documento deve incluir:
  - Descrição de alto-nível dos requisitos para entendimento por parte dos usuários;
  - Modelos técnicos detalhados que especifiquem os requisitos para os desenvolvedores construírem o projeto do sistema.

# Objetivos

- Caracterizar os requisitos do sistema:
  - Identificar entidades relevantes, como se relacionam e como se comportam.
- Ser passível de compreensão tanto por desenvolvedores como por usuários;
- Descrever o sistema sob uma perspectiva externa (**o que** ele faz, não **como** faz) – abordagem caixa preta;
- Ser completo, consistente e não ambíguo.

# Como descrever requisitos

- Linguagem natural;
- Linguagem natural estruturada;
  - Formulários e templates.
- Linguagem de descrição de projeto;
  - Baseadas em linguagem de programação.
- Notações gráficas;
- Especificações matemáticas.

# Casos de uso

- Uma forma de estruturar requisitos:
  - Modelos gráficos e linguagem natural baseada em formulários;
  - Representam o que os usuários podem fazer no sistema;
  - São independentes do método de análise (OO, estruturado, etc.).

# Definições

*“Um caso de uso captura um contrato que descreve o comportamento do sistema sob várias condições a medida que ele responde a requisições de um de seus usuários.”  
(Alistair Cockburn)*

*“Um caso de uso conta uma história sobre como um usuário final (interpretando um de uma série de papéis) interage com o sistema dentro de um conjunto de circunstâncias.” (Roger Pressman)*

*“Um caso de uso especifica um comportamento de um sistema segundo uma perspectiva externa e é uma descrição de um conjunto de sequências de ações realizadas pelo sistema para produzir um resultado de valor observável por um ator.” (Grady Booch)*



# Em outras palavras...

- Um caso de uso:
  - É uma interação típica entre o sistema e um ator – humano, outro sistema ou dispositivo;
  - Captura uma função visível ao ator;
  - Busca atingir uma meta do usuário.

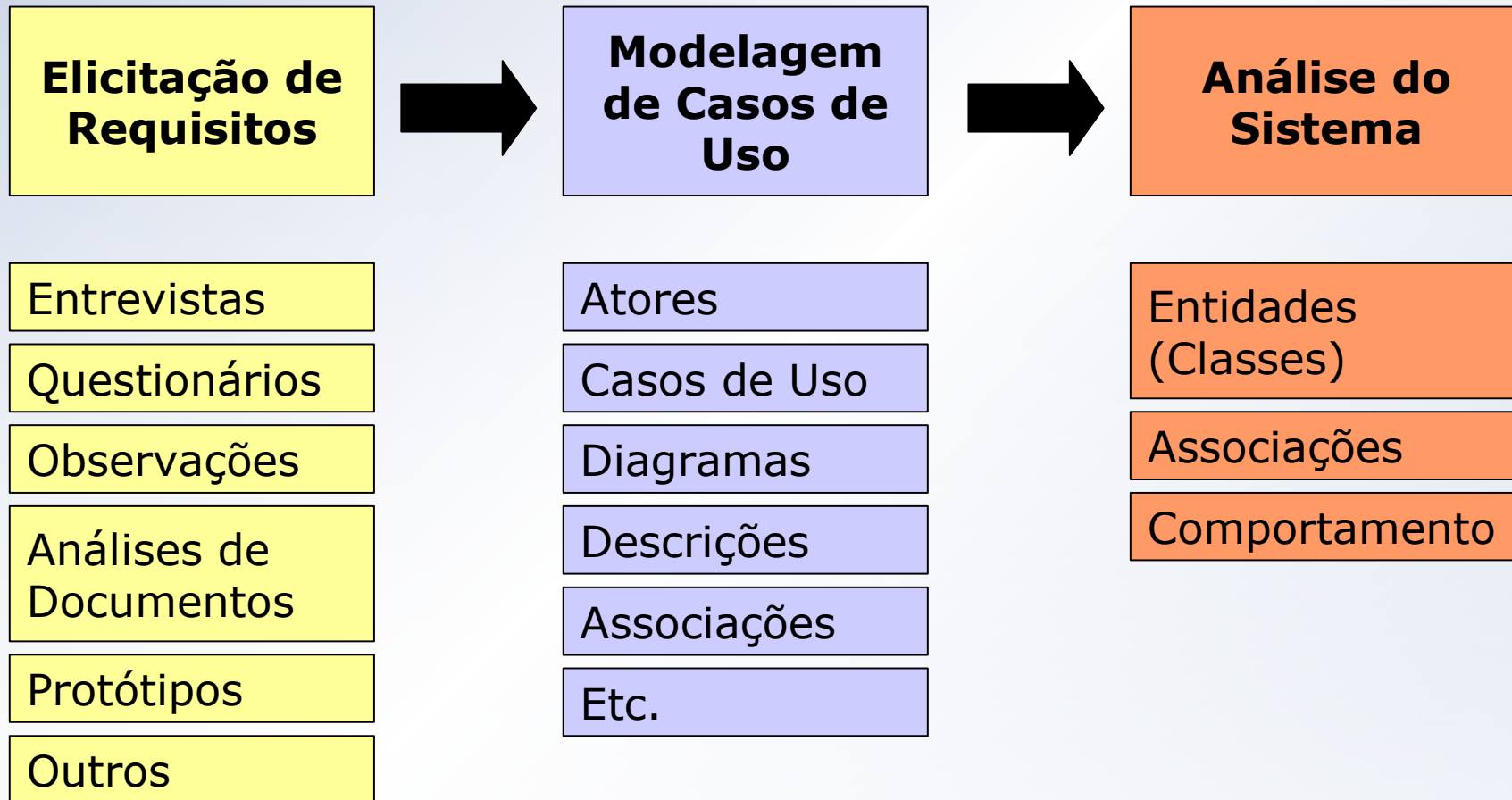
# Objetivos dos casos de uso

- Devem responder (Jacobson):
  - Quem são os atores?
  - Quais são seus objetivos?
  - Que pré-condições existem?
  - Quais as tarefas principais realizadas?
  - Que exceções devem ser consideradas?
  - Que variações são possíveis nas interações?
  - Que informações do sistema serão adquiridas, produzidas ou alteradas?

# Objetivos dos casos de uso

- Em resumo: representar o comportamento desejado do sistema (em termos de requisitos funcionais);
- Podem ser usados como base para:
  - Construção de casos de teste;
  - Estimativas de custo (cronograma) e tempo;
  - Identificação dos riscos;
  - Definição de prioridades;
  - Prototipação;
  - Manuais de usuário e documentação em geral.

# Ponte entre requisitos e análise



# Passos

1. Identificação dos atores;
2. Captura dos casos de uso;
3. Criação de diagramas de casos de uso;
4. Elaboração da descrição de cada caso de uso;
5. Análise de possíveis associações entre casos de uso;
6. Separação dos casos de uso em subsistemas.

# 1) Identificação dos atores

- Um ator é um papel específico que um usuário pode desempenhar;
  - Um mesmo usuário pode desempenhar vários papéis, cada hora sendo um ator diferente.
- Modela qualquer coisa externa que possa interagir com o sistema:
  - Usuários, outros sistemas, dispositivos, etc.;
  - Delimitam o escopo do sistema;
  - Não é necessário ser descrito em detalhes.

# Perguntas para identificar atores

- Quem utiliza o sistema?
- Quem instala e mantém o sistema?
- Que outros sistemas/dispositivos utilizam o sistema ou são utilizados por ele?
- Quem obtém informação do sistema?
- Quem provê informação ao sistema?
- O que o sistema faz automaticamente?

## 2) Captura dos casos de uso

- Feita durante a concepção (conversas iniciais) e elicitaco (entrevistas, etc.);
  - Identifique as interaes discretas entre usurios e sistema;
  - Dê um nome a cada uma delas;
  - Escreva uma descrio textual pequena.
- Geralmente so identificados em paralelo com a identificao dos atores;
- Alguns casos e atores podem ser capturados em fases mais avanadas.



# Perguntas para identificar casos de uso

- Que funções o ator irá querer do sistema?
- O sistema armazena informações?
- Que atores irão criar, ler, atualizar ou apagar estas informações?
- O sistema precisa notificar algum ator sobre alguma mudança interna?
- Existem eventos externos que o sistema precisa estar ciente?
- Que atores informam o sistema sobre estes eventos?

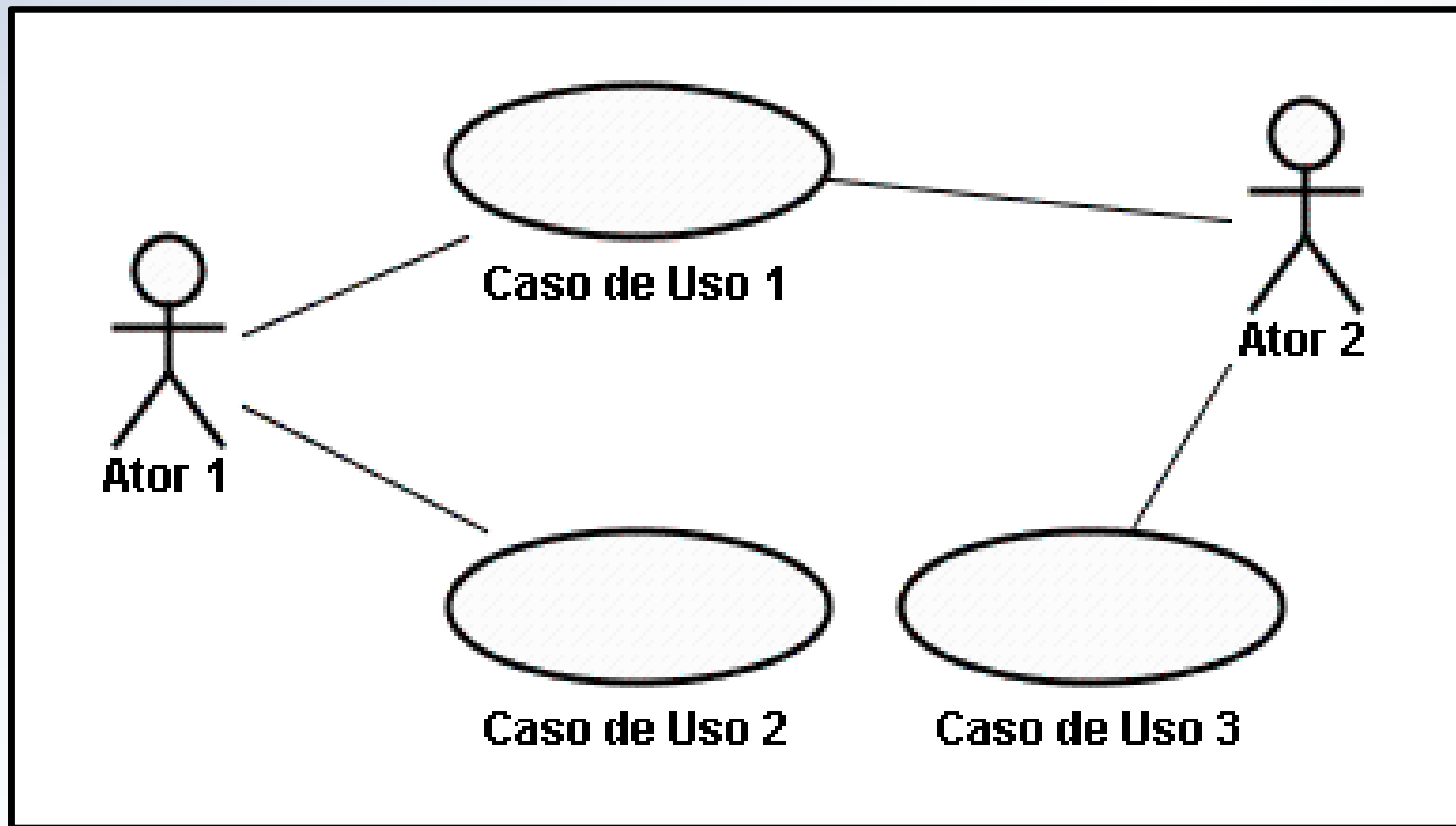
# Granularidade dos casos de uso

- Casos de uso não devem ser muito pequenos nem muito grandes;
- Um bom caso de uso compreende uma seqüência de transações realizadas pelo sistema que produzem *um resultado de valor observável para um ator específico*.

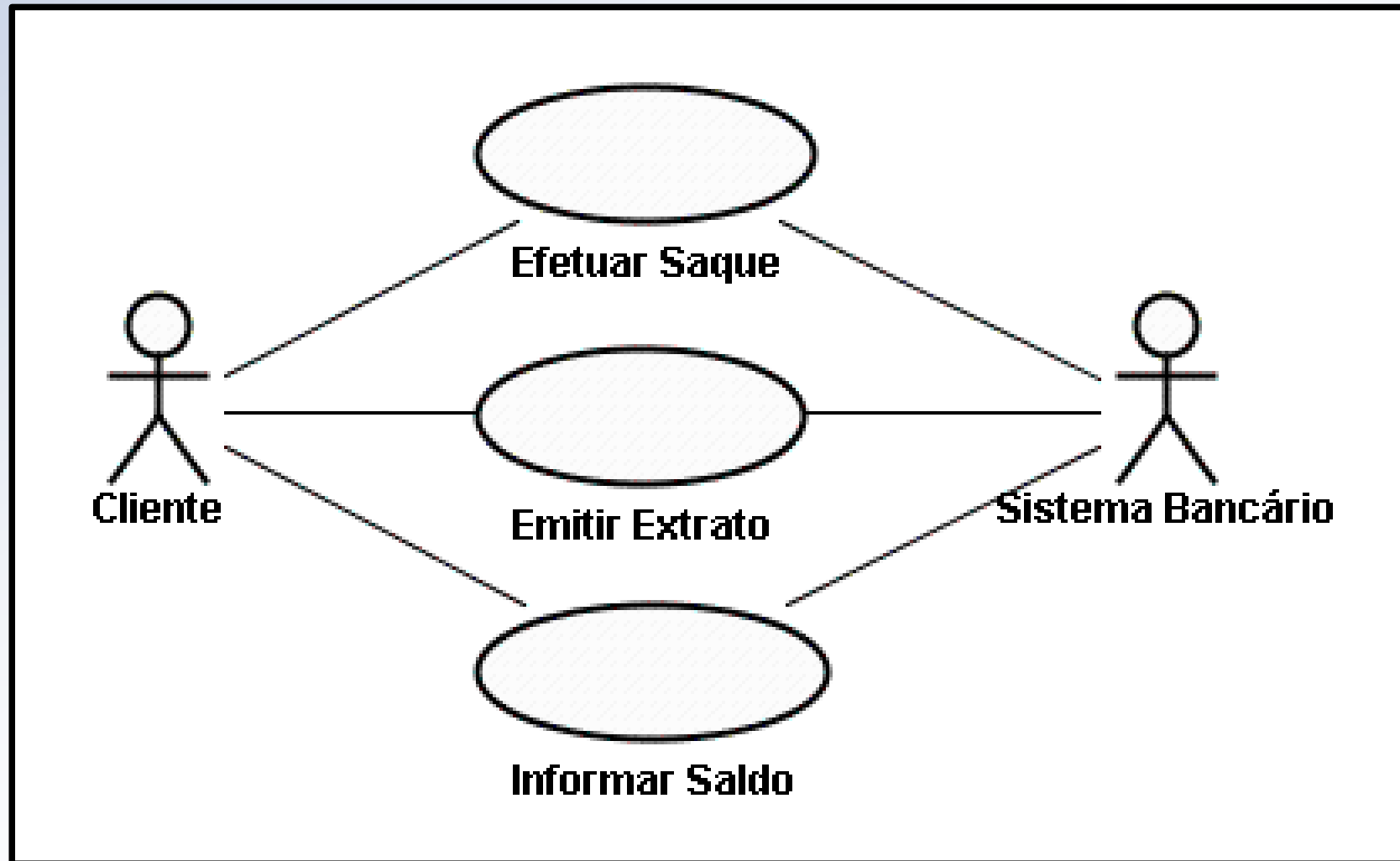
### 3) Diagramas de casos de uso

- Representam atores, casos de uso e suas associações;
- Uma associação entre um ator e um caso de uso significa que estímulos podem ser enviados entre atores e casos de uso, que se comunicam entre si;
- Provêem uma visão geral das funcionalidades do sistema.

# Elementos do diagrama de casos de uso

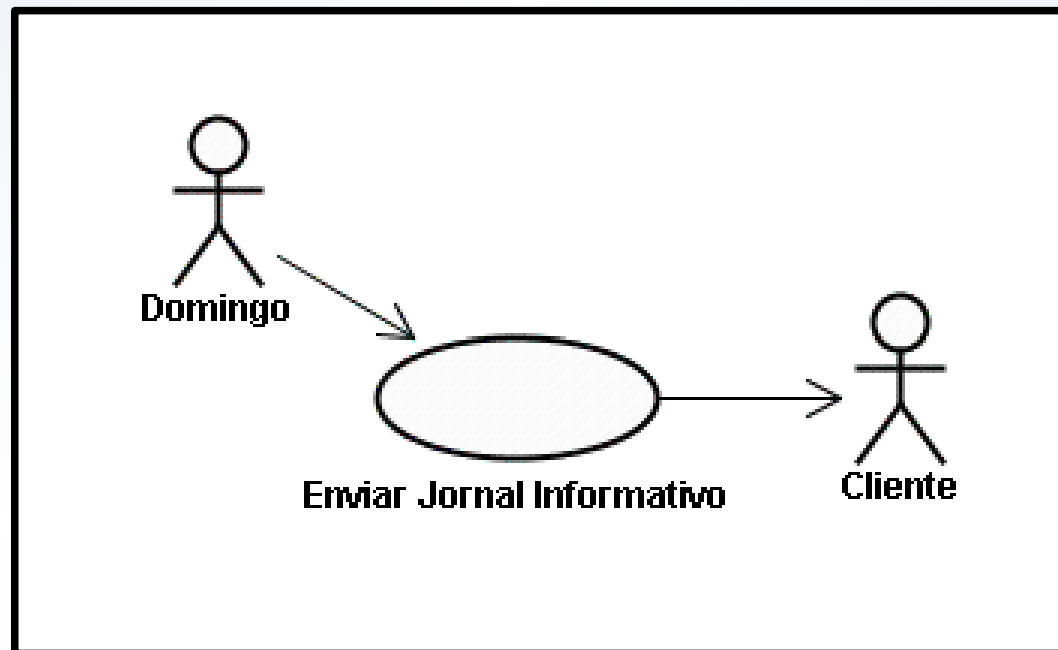


# Exemplo



# Eventos que ocorrem automaticamente

- Eventos podem ser disparados por determinadas condições, geralmente temporais:
  - Ex.: realizar backup a cada sexta-feira.
- Podemos mapear o evento como um ator ou tratá-lo como um elemento interno.



## 4) Descrição dos casos de uso

- O diagrama é insuficiente para dizer o que cada caso de uso faz;
- Deve-se descrever textualmente o fluxo de eventos de cada caso separadamente;
- Esta tarefa deve ser iniciada após alguma estabilidade dos casos de uso, para evitar perda de tempo.

# O que pode constar na descrição

- Nome do caso de uso;
- Descrição breve / objetivos;
- Pré-condições e pós-condições;
- Entradas e saídas de dados;
- Fluxos (normal, alternativos, cenários);
- Classes/entidades participantes;
- Restrições de domínio;
- Requisitos não-funcionais associados;
- Outras observações.



# Curso normal e cursos alternativos

- Curso Normal: mundo perfeito, tudo ocorre como planejado;
- Cursos Alternativos: exceções, erros, fluxos alternativos, etc.
- Para encontrá-los, analise o curso normal e pergunte, para cada item:
  - Tem alguma outra ação que pode ser feita?
  - Tem alguma coisa que pode dar errado?
  - Existe algum comportamento que pode ocorrer a qualquer momento?

# Exemplos de cursos alternativos

- O ator sai da aplicação;
- O ator cancela a operação corrente;
- O ator pede ajuda;
- O ator provê dados inválidos;
- O ator provê dados incompletos;
- O ator escolhe uma maneira alternativa de realizar o caso de uso;
- O sistema falha;
- O sistema está indisponível.

# Representação dos cursos

- Curso Normal:
  - Parágrafos;
  - Lista numerada (preferível).
- Cursos Alternativos:
  - Intercalados no curso normal como itens;
  - Intercalados no curso normal como parágrafos;
  - Em uma seção separada, de forma resumida;
  - Em uma seção separada, de forma detalhada (semelhante ao curso normal).

# Cenários / eventos

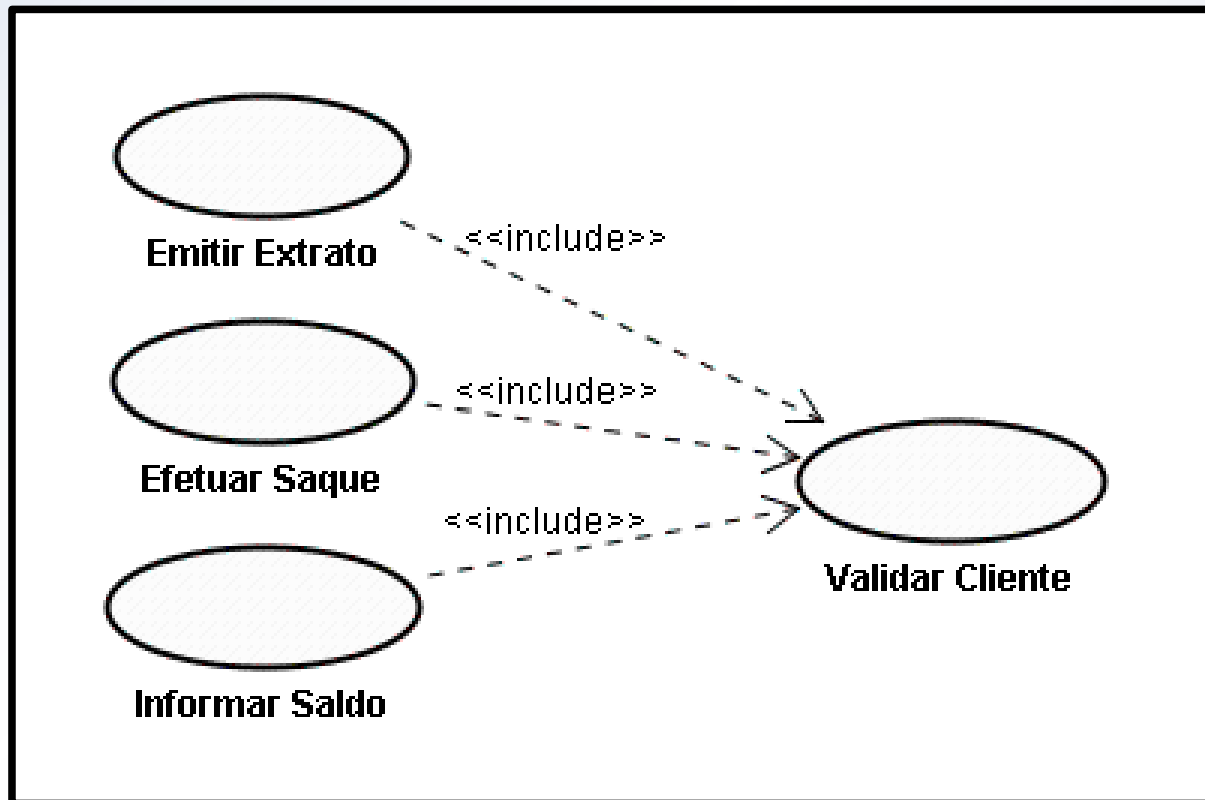
- Podemos agrupar casos de uso pequenos e similares num único caso de uso;
- Neste caso, temos um caso de uso e vários cenários;
  - Ex.: Cadastrar Cliente = Incluir + Consultar + Alterar + Excluir Cliente.
- Agrupar ou não é decisão do analista;
- Levar em consideração legibilidade e organização.

## 5) Associações entre casos de uso

- Existem três tipos de associações que podem ser detectadas à medida que os diagramas de casos de uso são refinados:
  - Relação de inclusão;
  - Relação de especialização (herança);
  - Relação de extensão.

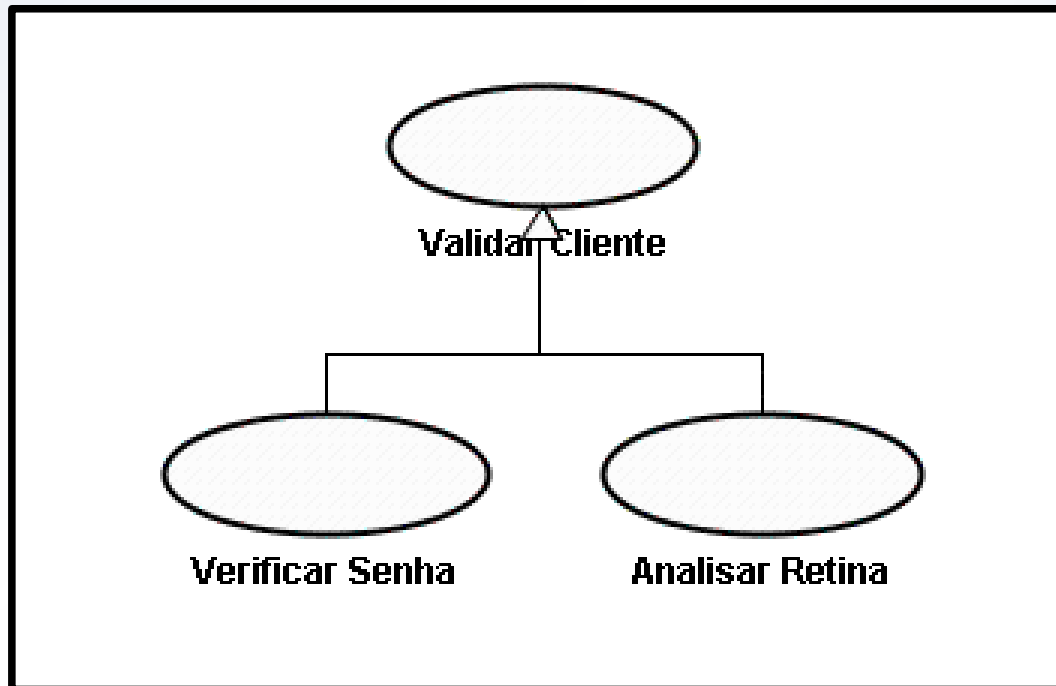
# Relação de inclusão

- Um caso de uso incorpora explicitamente o comportamento de outro;
- Funcionalidade comum é separada em um caso que é reutilizado por outros.



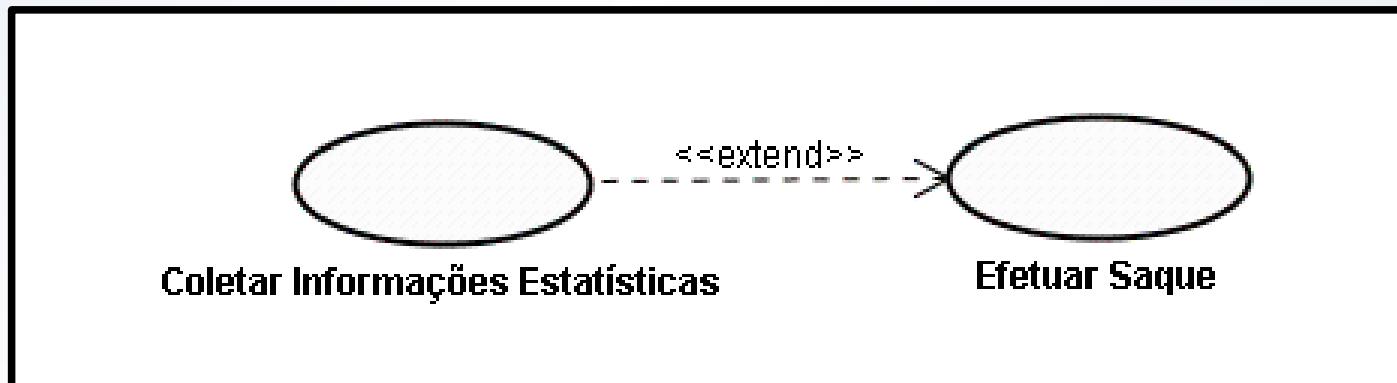
# Relação de generalização

- Um caso de uso filho herda o comportamento e o significado do caso de uso pai;
- Acrescenta ou sobrescreve comportamento do pai e pode substituir o pai em qualquer lugar que este apareça.



# Relação de extensão

- Um caso de uso base incorpora implicitamente o comportamento de um outro caso de uso em um local especificado;
- Permite capturar os requisitos funcionais de um sistema de forma incremental.





# Relação de extensão

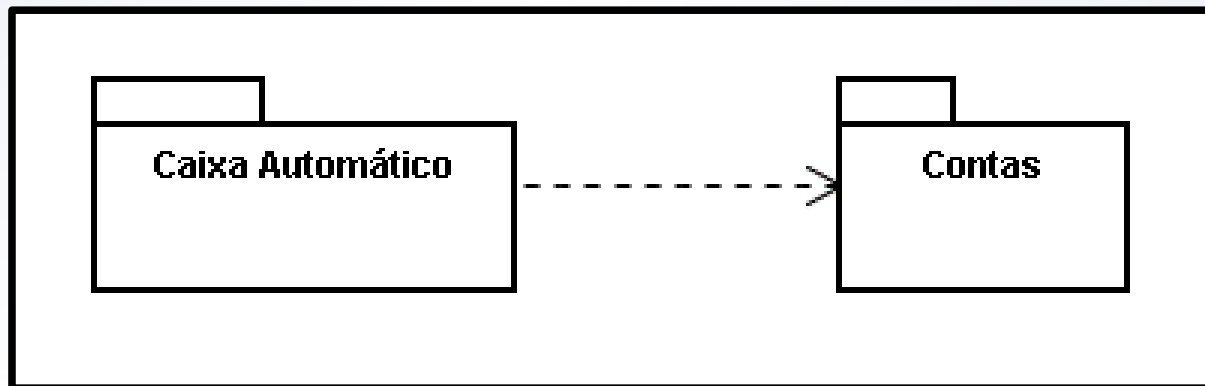
- Pode ser usado para modelar:
  - Partes opcionais de casos de uso;
  - Cursos complexos e alternativos;
  - Subseqüências que são executadas apenas em certos casos;
  - A inserção de diversos casos de uso diferentes dentro de um outro.

# Inclusão x extensão

- Extensão:
  - Quando estiver descrevendo uma variação de um curso normal;
  - O caso estendido conhece o caso base.
- Inclusão:
  - Quando houver repetição de um mesmo fluxo em dois ou mais casos de uso e quer se evitar isso;
  - O caso base conhece o caso incluído.

## 6) Separação em subsistemas

- Facilita o entendimento e a leitura;
- Utiliza-se o ícone de pacote da UML;
- Setas pontilhadas indicam dependência – um pacote solicita serviços de outro.



# Apoio de ferramentas

- Uma ferramenta CASE (*Computer-Aided Software Engineering*) auxilia no desenho de diagramas de caso de uso;
- Há várias ferramentas disponíveis;
- Recomendamos o Jude UML:
  - Download em <http://jude.change-vision.com>;
  - Versão *Community* é gratuita.

**JUDE**  
UML MODELING TOOL

# Padrão de nomenclatura

- Verifique o padrão de nomenclatura antes de começar:
  - Atores;
  - Casos de uso;
  - Pacotes;
  - Descrição do caso de uso.
- Para a descrição, use o modelo.