

## Aula 28

### Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

### Conteúdo:

**Casos de Uso**

Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

Conteúdo:

**Conceituação**

# Caso de Uso

- ➡ É uma descrição completa de um processo
- ➡ Algo que o usuário ou cliente quer que o sistema faça
- ➡ Podem ser simples como uma frase, ou altamente complexos

# Caso de Uso

- ➡ Conta uma história de como alcançar um objetivo
- ➡ Conta um conjunto de histórias de
  - ▬ como completar satisfatoriamente ou
  - ▬ como se pode falhar em uma tarefa

# Casos de Uso

- ➡ É a especificação, em forma de uma narrativa, de uma seqüência de interações entre um sistema e os agentes externos que utilizam esse sistema.
- ➡ Define o uso de uma parte da funcionalidade de um sistema.

# Objetivo

➡ Tornar um requisito funcional

- Legível pelo usuário e pelos outros interessados

- Verificável

➡ Tornar um requisito funcional

# Formas de Narrativa

- ➡ Descrição Contínua
- ➡ Descrição Numerada
- ➡ Narrativa Particionada

# Descrição Contínua

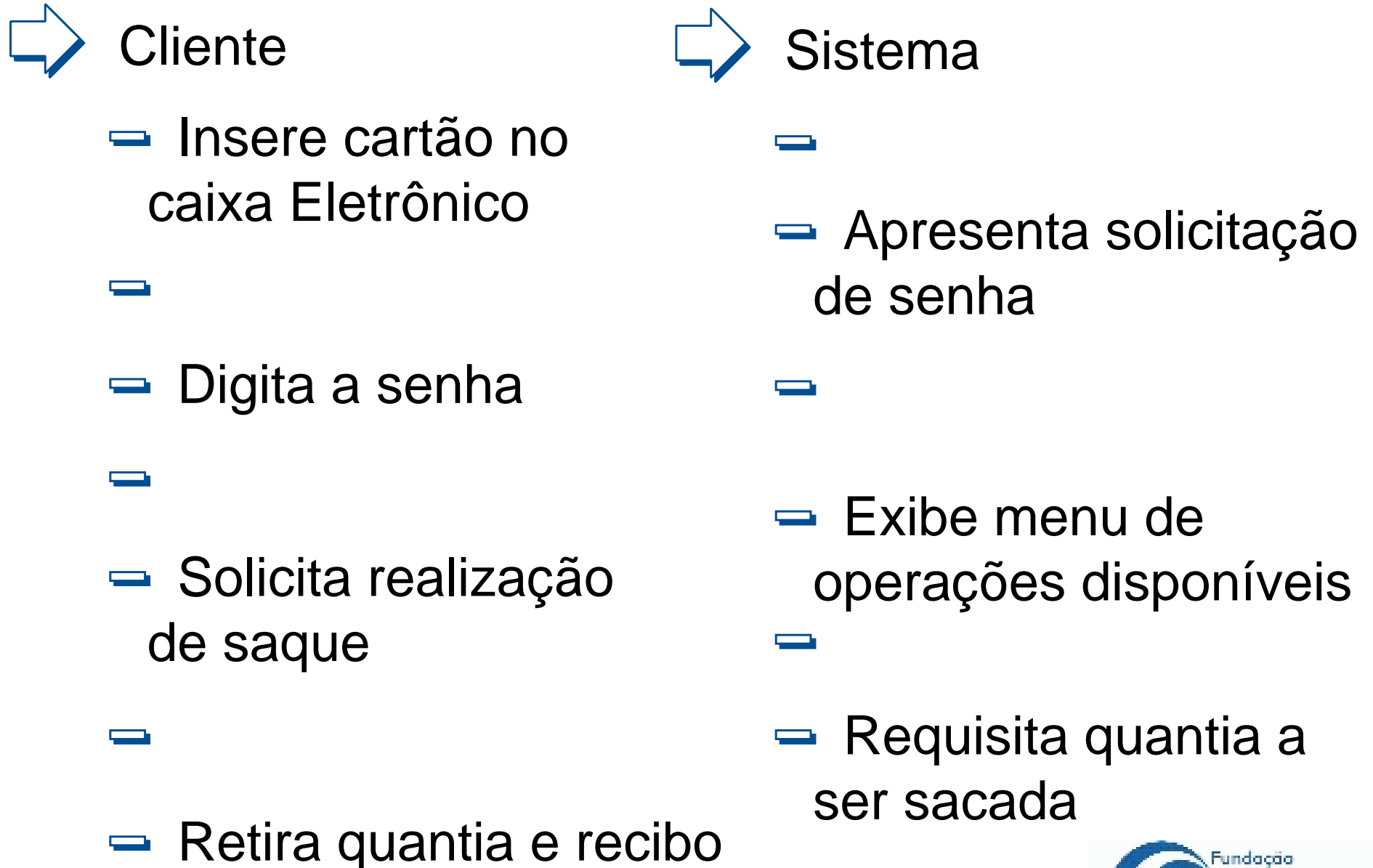
➡ O Cliente chega ao caixa eletrônico e insere seu cartão. O Sistema requisita a senha ao Cliente. Após o Cliente fornecer sua senha e esta ser validada, o Sistema exibe as opções de operações possíveis. O Cliente opta por realizar um saque. Então o Sistema requisita o total a ser sacado. O Sistema fornece a quantia desejada e imprime o recibo para o Cliente.



# Descrição Numerada

1. Cliente passa seu cartão no caixa eletrônico
2. Sistema apresenta solicitação de senha
3. Cliente digita senha
4. Sistema exibe menu de operações disponíveis
5. Cliente indica que deseja realizar um saque
6. Sistema requisita quantia a ser sacada
7. Cliente informa quantia a ser sacada
8. Sistema fornece dinheiro
9. Sistema imprime recibo

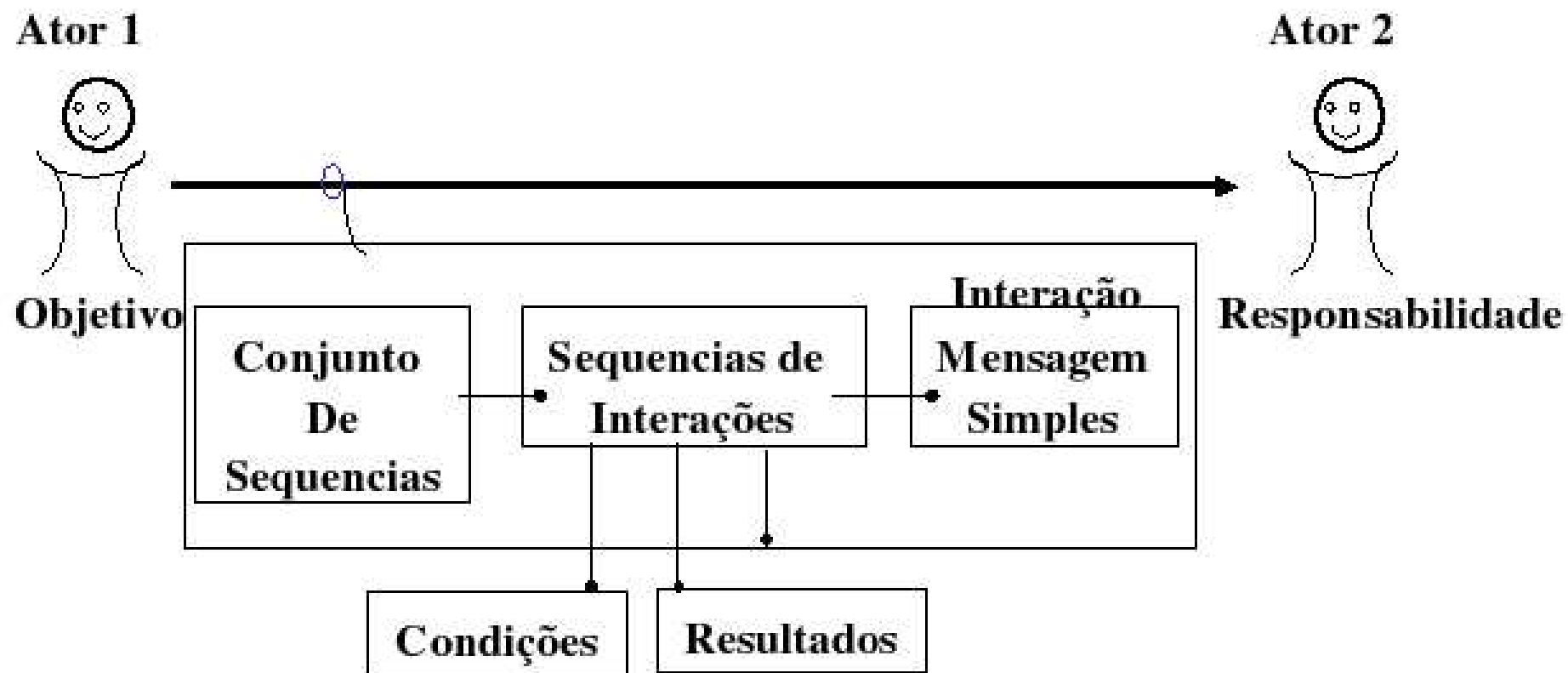
# Narrativa Particionada



# Principais Problemas

- ➡ Difíceis de estruturar
- ➡ Algumas dúvidas cruéis
- ➡ Muitas formas de entender
- ➡ Muitas formas de fazer
- ➡ Não são os gráficos de UML

# O Modelo



# Ator

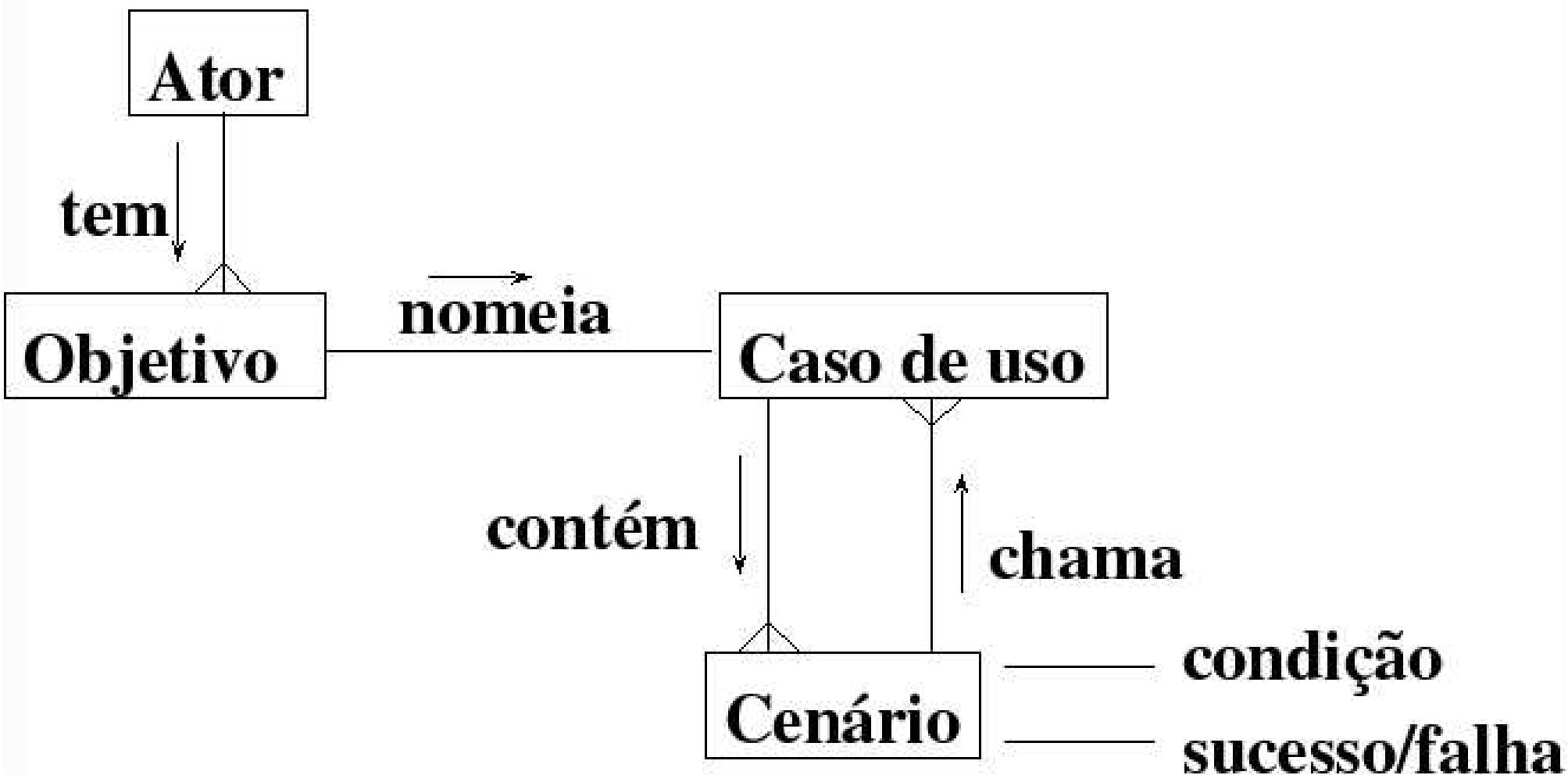
- ➡ Entidade externa ao sistema com comportamento próprio
- ➡ Um ator ativa uma interação, chamando outros atores a cumprir suas responsabilidades
- ➡ Atores interagem para alcançar seus objetivos
  - O sistema também é um "ator"

# Tipos de Atores

- ➡ Pessoas
- ➡ Organizações
- ➡ Outros sistemas
- ➡ Equipamentos

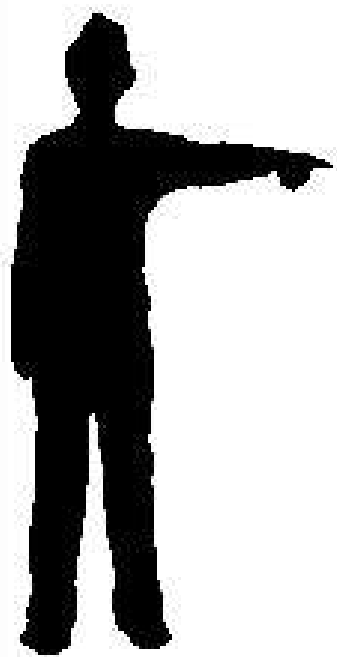
# Objetivos

- ➡ Atores possuem objetivos
- ➡ Objetivos dão nome aos casos de uso
- ➡ Casos de uso possuem cenários
  - Que dão nome a sub-casos de uso
- ➡ Examinar os objetivos que um sistema suporta  
fornece bons requisitos
  - Essenciais?
  - Objetivos resumem funções de forma usável e verificável





# Um usuário pode ter vários atores



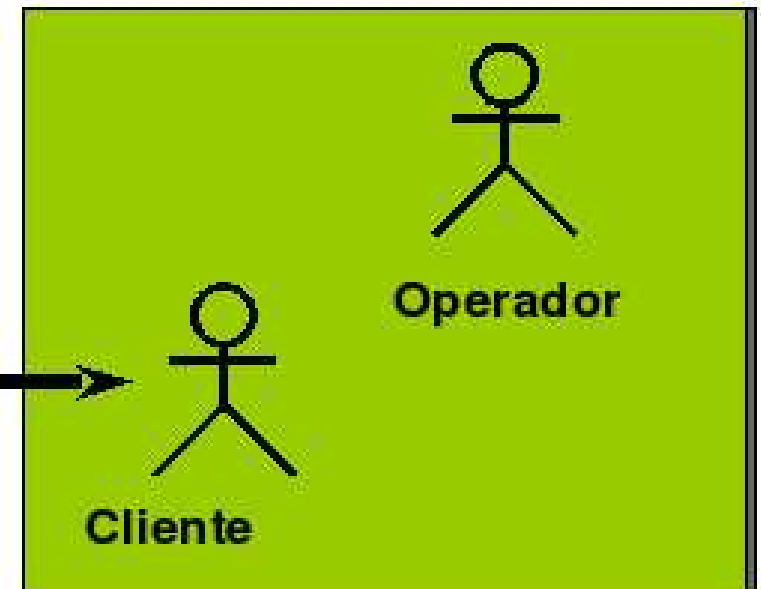
Carlos



Carlos  
como  
Cliente



Carlos como  
Operador

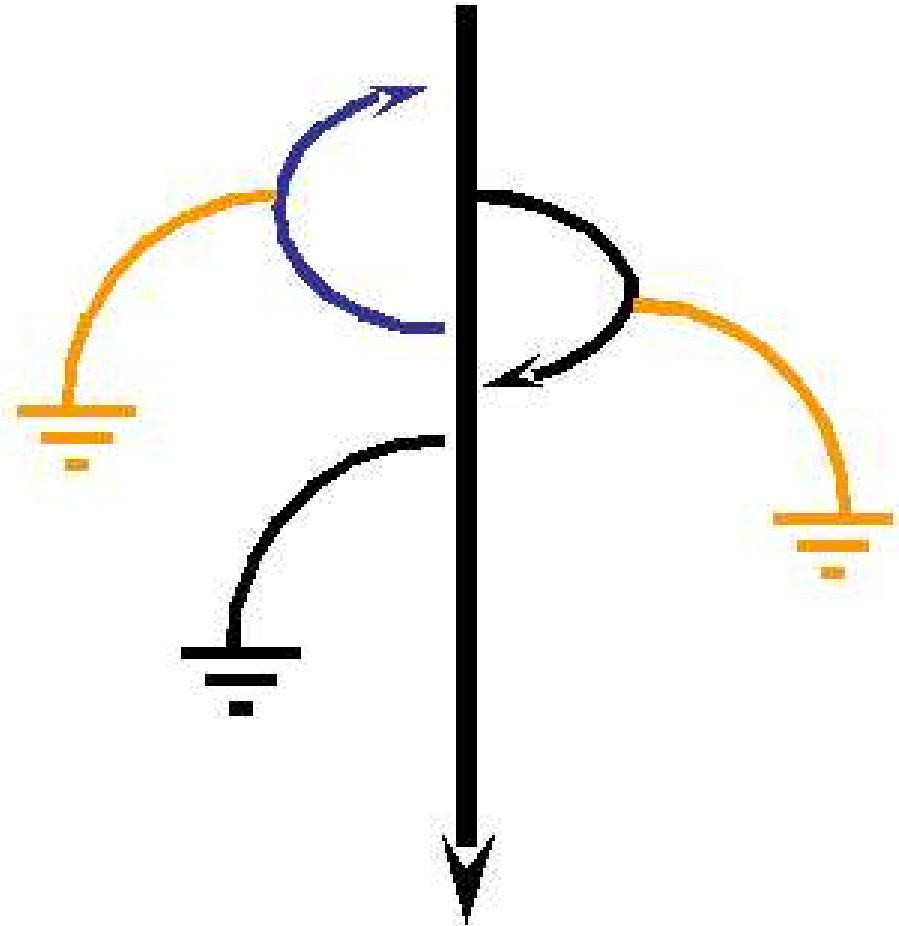


# Cenário

- ➡ Instância de Caso de Uso
- ➡ Uma seqüência específica de interações entre atores e o sistema
- ➡ Cada cenário mostra o resultado de uma condição
  - ▢ O usuário dá nome a condição
    - ▢ Tudo dá certo
    - ▢ Falta dinheiro no caixa

# Diferentes Fluxos de Eventos

- ➡ Caso Feliz!
- ➡ Casos Alternativos
  - ▢ Variantes Normais
    - ▢ Tirar dinheiro da conta
  - ▢ Casos Raros
    - ▢ Tirar US\$ 1 Milhão
  - ▢ Exceções e Erros
    - ▢ Não há mais dinheiro no caixa



# Calça do Caso de Uso

Objetivo: “Encomendar Livro”								
Subobjetivo:	sc1	sc2	sc3	sc4	sc5	sc6	sc7	...
Verificar ... crédito	S	S	F			S	F	
... estoque	S	F	S			F	F	
		S				F		
(cenários de sucesso)					(cenários de falha)			

# Caso de uso

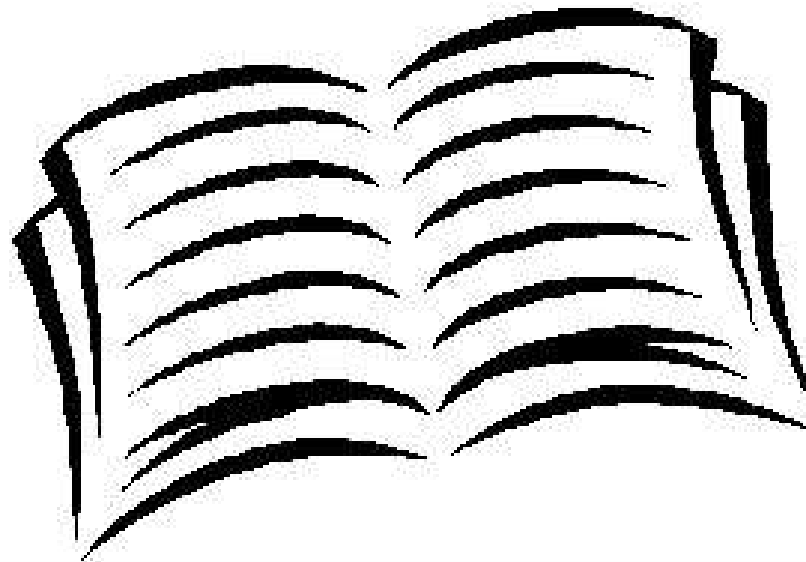
- ➡ Uma coleção de cenários de sucesso e fracasso descrevendo um determinado processo completo do sistema

# Estruturado em Narrativas

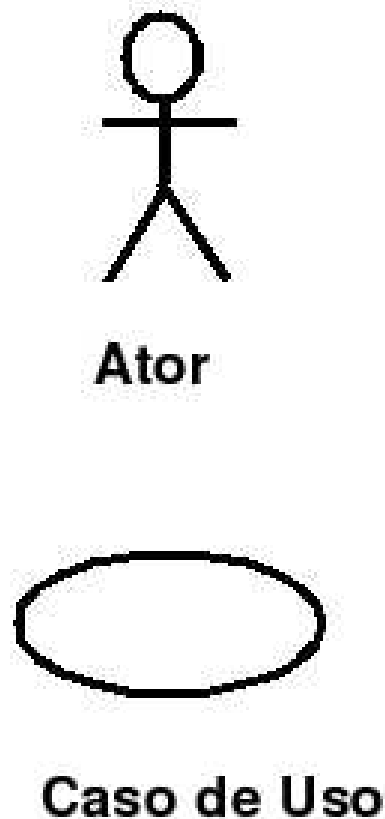
- ➡ Mantém o contexto visível
- ➡ Facilita a compreensão
  - ▬ Perde a abstração
- ➡ Deixa claro o valor para o usuário
  - ▬ Quem?
  - ▬ Quando?

# Formato

- ➡ Casos de uso são basicamente **texto**
  - ⇒ não diagramas
  - ⇒ embora exista um diagrama na UML para indicar quais atores participam de cada caso de uso



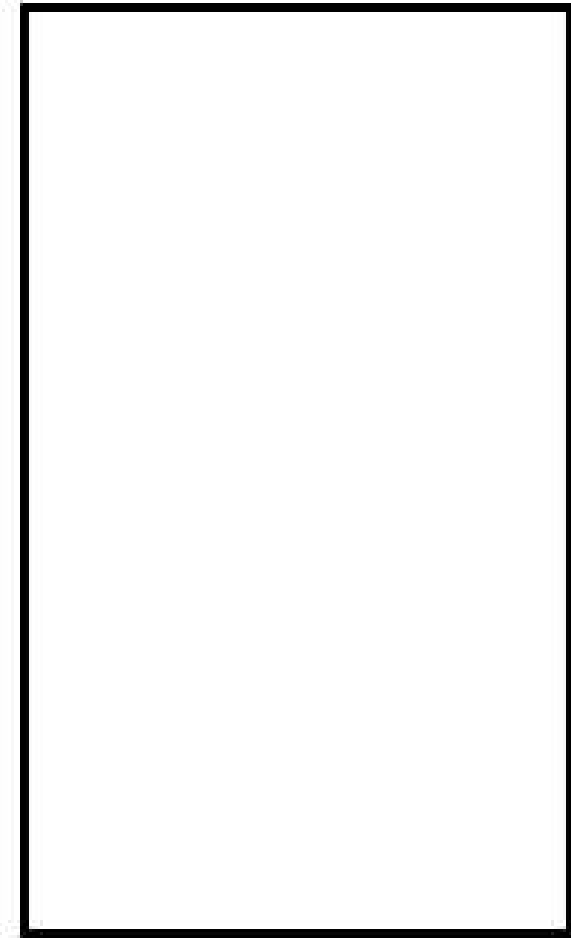
# Diagrama de Casos de Uso



**Comunicação  
(usa)**

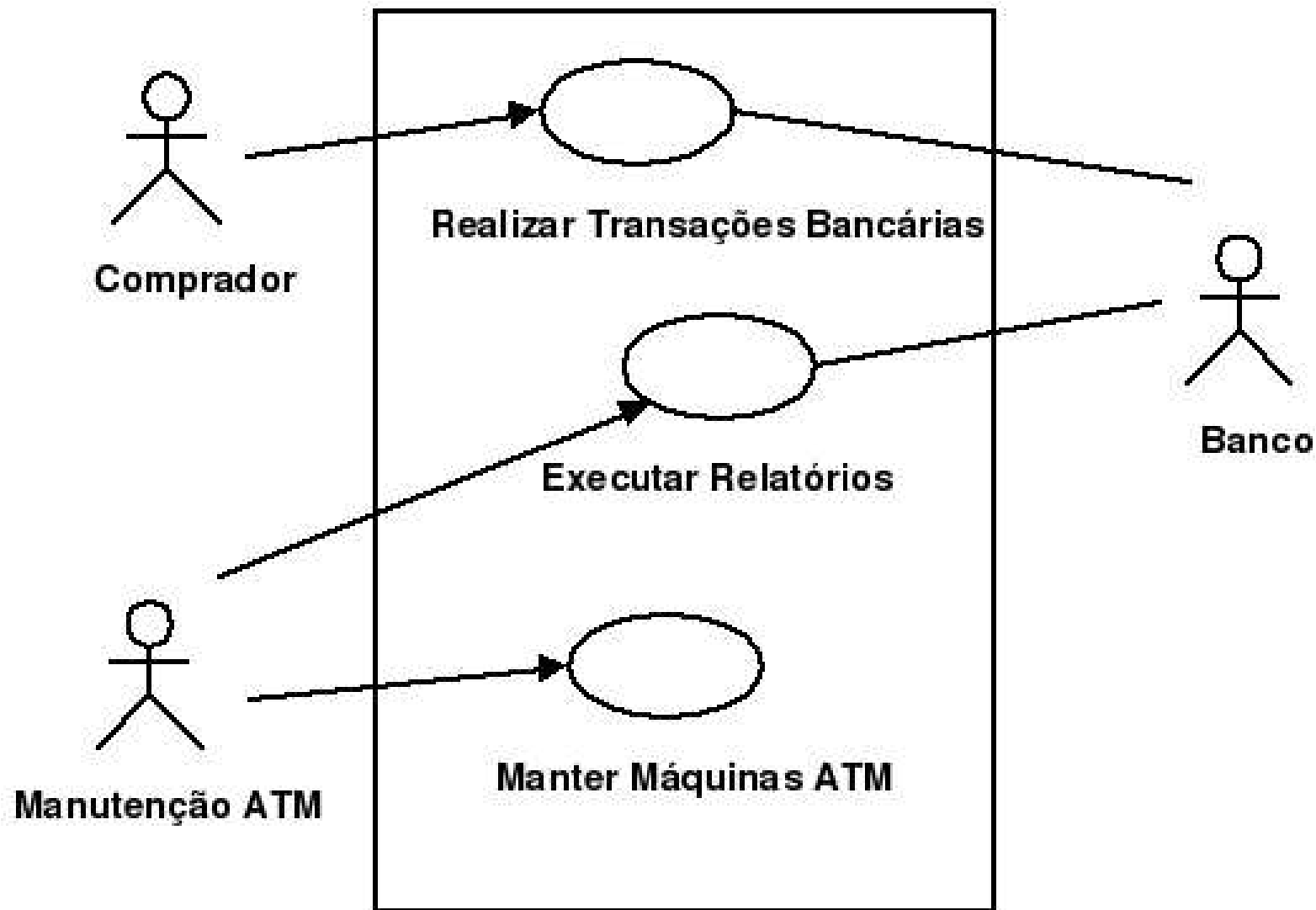


**Comunicação  
indicando  
iniciação**





# Modelo Exemplo



Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

Conteúdo:

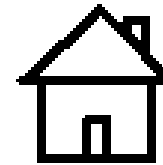
**Níveis de Abstração  
e  
Escopo**

# Escopo

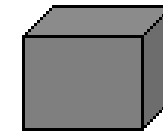
➡ Organização, caixa preta



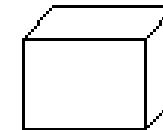
➡ Organização, caixa branca



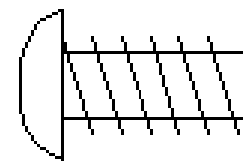
➡ Sistema, caixa preta



➡ Sistema, caixa branca



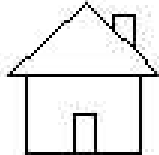
➡ Componente



# Escopo: Organização

➡ É um caso de uso de negócio, com a empresa como seu escopos

➡ Se a empresa é vista como uma caixa preta, use a casa escura 

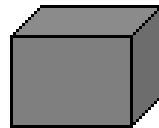
➡ Se o caso de uso trata de departamentos ou staff, use a casa branca 

# Escopo: Sistema

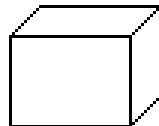
➡ O escopo é o sistema informatizado

➡ Use as caixas de acordo com o tipo de descrição

▢ Caixa-preta

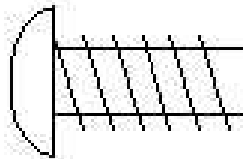


▢ Caixa-branca

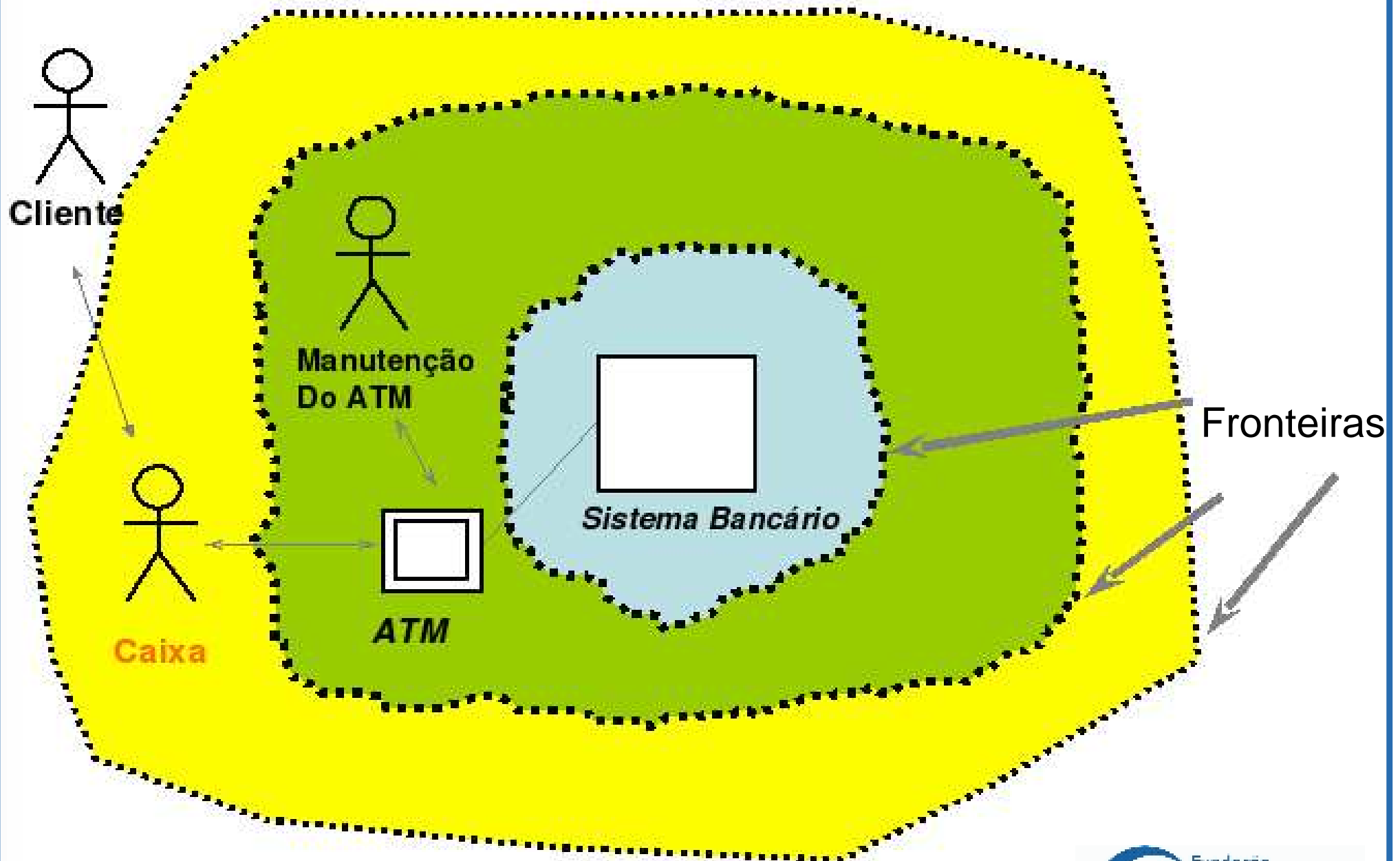


# Escopo: Componente

➡ Um subsistema ou componente do sistema sendo projetado



# Fronteiras e Escopo



# Níveis de Abstração I

## ➡ Casos de Uso Essenciais

- Não levam em consideração a tecnologia

## ➡ Casos de Uso Reais



# Aparência - Essencial



## Caso de Uso Essencial

- Cliente fornece a sua identificação
- Sistema identifica usuário
- Sistema oferece operações disponíveis
- Cliente solicita saque de uma determinada quantia
- Sistema fornece a quantia desejada da conta do cliente
- Cliente recebe dinheiro e recibo

# Aparência - Real



## Caso de Uso Real

- Cliente passa seu cartão no caixa eletrônico
- Sistema apresenta solicitação de senha
- Cliente digita senha
- Sistema exibe menu de operações disponíveis
- Cliente indica que deseja realizar um saque
- Sistema requisita quantia a ser sacada
- Cliente informa quantia a ser sacada
- Sistema solicita re-inserção do cartão
- Cliente insere o cartão
- Sistema fornece dinheiro
- Cliente retira dinheiro
- Sistema fornece recibo
- Cliente retira recibo
- Sistema libera cartão
- Cliente recupera cartão

# Como testar

## ➡ Teste do Tempo

- Essa narrativa seria válida 100 anos atrás?
- Essa narrativa será válida daqui a cem anos

## ➡ Testes das Tecnologias

- E se for feito com cartas?
- E se for feito via fax?
- E se for feito com reconhecimento e sintetização de voz?
- E se for feito com leitura de mente?

# Níveis de Abstração II

➡ Objetivos Estratégicos

➡ Objetivos de Usuários

➡ Sub-funções

➡ Ligados entre si

# Objetivos Estratégicos

Objetivo do projeto

anunciar

pedir

cobrar

“branco”

## Obj. de Usuários

Preparar promoção

Referenciar promoção

Monitorar promoção

Fazer pedido

Criar fatura

Enviar fatura

“azul”

Identificar promoção

Registrar usuário

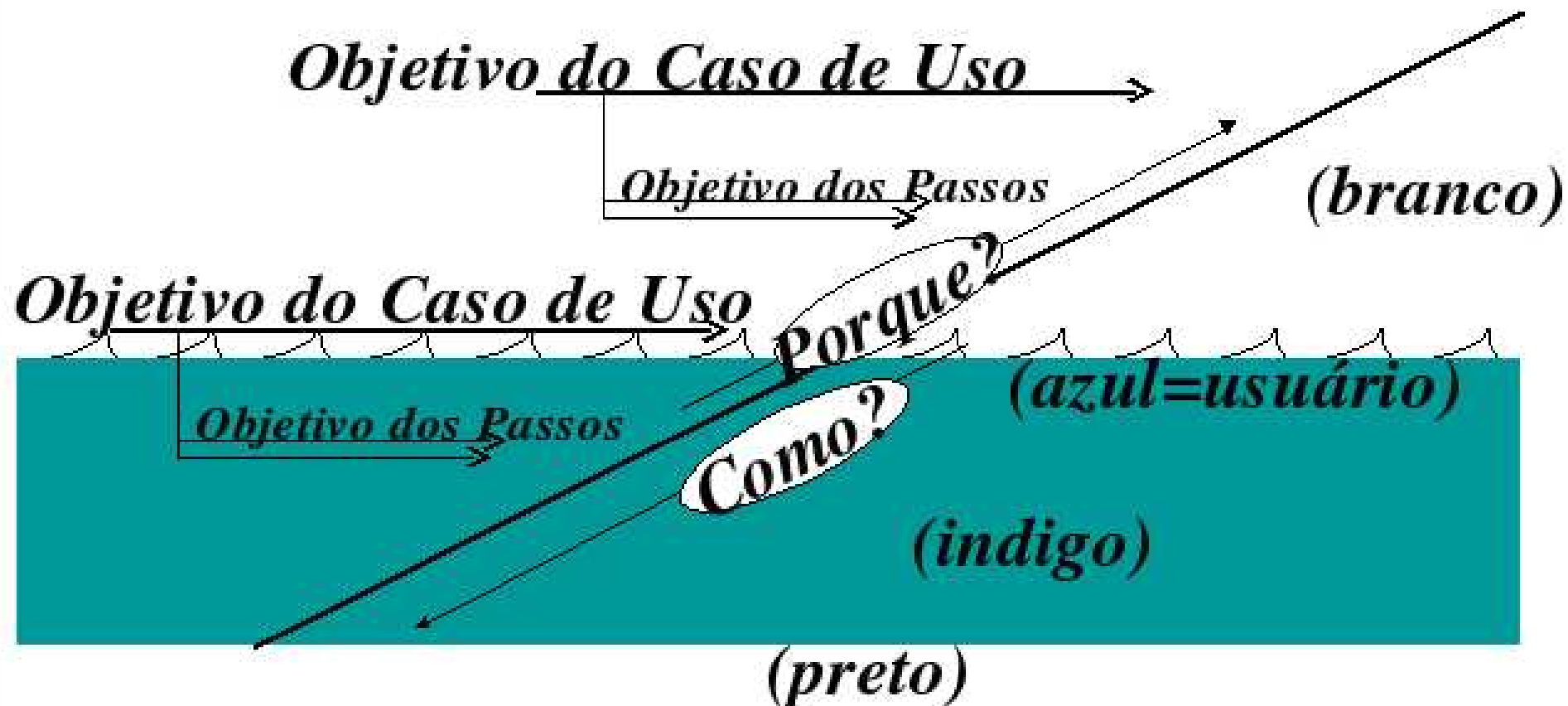
Identificar produto

Identificar Comprador

“indigo”

## Subfunções

# Objetivo Casos de Uso > Objetivos dos Passos

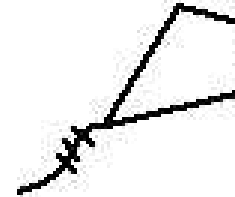


# Níveis

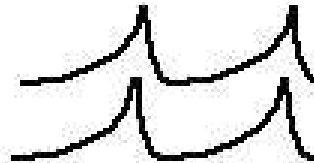
➡ Sumário de alto nível



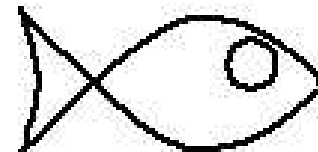
➡ Sumário



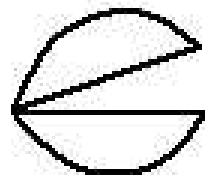
➡ Objetivo do Usuário



➡ Sub-Função



➡ Muito detalhado



# Níveis (nomes)

- ➡ Adicione "+" no final de casos de uso sumário
- ➡ Adicione "!" ou nada nos casos de uso de usuários
- ➡ Adicione "-" nos casos de uso detalhados



# Abstração



Nível

— Por que queremos esse objetivo?



Escopo


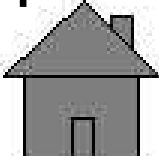
— Que limites do sistema queremos?



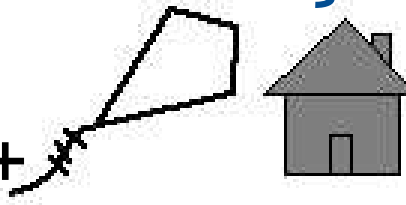
Detalhe

— Descrevemos intenção ou ações?

# Nível de Sumário

- ➡ Caso de Uso: Operar Companhia de Seguros+  
- Ator Primário: o comprador
  - Escopo: A companhia de seguros
  - Nível: Sumário
  - Passos
    1. O Comprador pede um cotação para o seguro
    2. O Comprador compra um seguro
    3. O Comprador solicita uma cobertura
    4. O comprador cancela o seguro

# Exemplo: Adicionar novo serviço

➡ UC: Adicionar novo serviço+ 

➡ Ator primário: Cliente

➡ Escopo: TeleFalaFácil

➡ Nível: Sumário

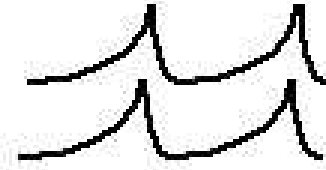
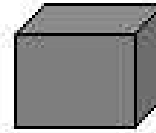
5. Cliente chama TeleFalaFácil e solicita um novo serviço

6. TeleFalaFácil...

# Exemplo: Adicionar novo serviço



UC: Adicionar novo serviço!



Ator primário: Atendente do Cliente Externo



Escopo: Sistema AtendeJá



Nível: Objetivo do Usuário

5. Cliente liga, Atendente discute pedido com o cliente

6. Atendente encontra o cliente no AtendeJá

7. AtendeJá apresenta os serviços já possuídos pelo cliente

8. ...

Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

Conteúdo:

**Detalhando o Caso  
de Uso**

# Tipos de detalhamento

- ➡ **Breve** - apenas uma frase ou parágrafo descrevendo o processo principal e típico
- ➡ **Casual** - descreve diferentes cenários, mas cada descrição é composta por apenas um parágrafo
- ➡ **Expandido** - Todos os passos e variações são detalhadamente descritos, incluindo pré-condições e pós-condições de sucesso.

# Exemplo de caso de uso Breve

➡ Caso de uso: Alugar Fitas

➡ Um cliente solicita a locação de algumas fitas.  
Após identificar-se e identificar as fitas ele pode levá-las para casa, ciente do prazo de devolução e do valor a ser pago.

# Exemplo de caso de uso casual

- ➡ Caso de uso: Alugar Fitas
- ➡ Um cliente solicita a locação de algumas fitas. Após identificar-se e identificar as fitas, se não houver problemas no seu cadastro e se as fitas não estiverem reservadas para outro cliente, ele pode levá-las para casa, ciente do prazo de devolução e do valor a ser pago.



# Exemplo de caso de uso expandido

➡ Caso de Uso: Alugar Fitas

➡ Fluxo Principal:

- ➡ 1. O cliente chega ao balcão com as fitas que deseja alugar .
- ➡ 2. O cliente informa seu nome e entrega as fitas ao funcionário.
- ➡ 3. O funcionário registra o nome do cliente e inicia a locação.
- ➡ 4. O funcionário registra cada uma das fitas.
- ➡ 5. O funcionário finaliza a locação, devolve as fitas ao cliente e lhe informa a data de devolução e o valor total da locação.
- ➡ 6. O cliente vai embora com as fitas.

# Fluxos Alternativos

## ≡ 3a. O cliente não possui cadastro.

≡ 3a.1 O cliente deve informar seus dados para cadastro.

≡ 3a.2 O funcionário registra o cadastro.

≡ 3a.3 Retorna ao fluxo principal no passo 3.

## ≡ 3b. O cliente possui pendências no cadastro (locação anterior não foi paga).

≡ 3b.1 O cliente paga seu débito.

≡ 3b.2 O funcionário registra a quitação do débito, eliminando assim a pendência.

≡ 3b.3 Retorna ao passo 3.

# Fluxos Alternativos

- 4a. Uma fita está reservada para outro cliente.
  - 4a.1 O funcionário informa que a fita não está disponível para locação.
  - 4a.2 Prossegue a locação do passo 4 sem incluir a fita reservada.
- 4b. Uma fita está danificada.
  - 4b.1 O funcionário informa que a fita está danificada.
  - 4b.2 O funcionário registra que a fita está danificada.
  - 4b.2 O funcionário verifica se existe outra fita disponível com o mesmo filme.
  - 4b.3 Se existir, o funcionário substitui a fita e segue no passo 4, senão segue do passo 4 sem incluir a fita danificada.

# Possíveis Seções para um Caso de Uso Expandido

- ➡ **Atores**
- ➡ Interessados
- ➡ Pré-condições
- ➡ Pós-condições de sucesso
- ➡ **Cenário principal de sucesso ou fluxo principal**
- ➡ **Extensões ou fluxos alternativos**
- ➡ Requisitos Especiais
- ➡ Variações tecnológicas e de dados
- ➡ Frequência
- ➡ Questões em aberto

# Atores

- ➡ São as classes de pessoas e sistemas externos que interagem com o sistema de alguma forma



# Interessados - Stakeholders

- ➡ São as classes de pessoas e sistemas externos que interagem com o sistema de alguma forma
- ➡ Quem tira proveito de seus resultados?
- ➡ Muitas vezes não são apenas os atores



# Pré-condições

- ➡ O que deveria ser sempre verdadeiro para que o caso de uso possa acontecer
- ➡ Pré-condições NÃO são testadas dentro do caso de uso.
- ➡ Elas são assumidas como verdadeiras antes do início dele.
- ➡ Devem comunicar APENAS questões dignas de nota, que constituam informação útil sobre o funcionamento do sistema

# Pós-Condições ou Garantias de Sucesso

- ➡ Estabelecem o que deve ser verdadeiro APÓS o caso de uso.
- ➡ Todos os interessados devem ser satisfeitos





# O "Caminho Feliz" ou Fluxo Principal

- ➡ Apresenta uma seqüência de passos
- ➡ Normalmente NÃO tem condições ou ramificações
- ➡ Exceções são tratadas como seqüências alternativas
- ➡ Os passos devem descrever trocas de informação (interação), validação ou mudança de estado.

# Tipos de Passos

- ➡ Obrigatórios - Fluxo de informação
- ➡ Complementares - Contextualização
- ➡ Não-recomendados - Controle e execução  
(passos internos ao sistema)

# Extensões ou Fluxos Alternativos

- ➡ Estão associadas aos passos do fluxo principal
- ➡ Identificam um erro, a forma de tratá-lo e como retornar ao fluxo principal, se for possível

# Um fluxo alternativo tem pelo menos quatro partes

- ➡ Identificação - o número da linha do fluxo principal onde a exceção ocorre e um identificador para a própria exceção na lista (por exemplo, 1a, 1b, ..., 2a, 2b, ...)
- ➡ Identificação da exceção - é necessário identificar qual a exceção que ocorreu, pois em uma mesma linha do fluxo principal podem ocorrer diferentes tipos de exceções. Por exemplo, "fita danificada", "fita reservada", etc.

# Um fluxo alternativo tem pelo menos quatro partes

- ➡ Ações corretivas - deve-se identificar a seqüência de ações que deveriam ser executadas para corrigir a exceção.
- ➡ Finalização - Se o caso de uso retorna ao fluxo principal depois das ações corretivas ou não.

# Finalização de uma exceção

- ➡ Voltar ao início do caso de uso, o que não é muito comum;
- ➡ Voltar ao início do passo que causou a exceção e executá-lo novamente, o que é mais comum.

# Finalização de uma exceção

- ➡ Depois das ações corretivas, ao invés de voltar para o mesmo passo, ir para o passo seguinte. Isso pode ser feito quando as ações corretivas realizam a operação que o passo deveria ter executado. Porém deve-se verificar se novas exceções não poderiam ainda ocorrer neste mesmo passo.
- ➡ Abortar o caso de uso. Neste caso, não se retorna ao fluxo principal.

# Requisitos Especiais

➡ Requisitos não funcionais associados ao caso de uso, como

- ▬ eficiência desejada,
- ▬ tecnologia de implementação,
- ▬ etc.



# Variações Tecnológicas e de Dados

- ➡ Se for o caso, indique as diferentes formas de realizar tecnologicamente os diferentes passos do caso de uso

# Questões em aberto

➡ Tudo o que deve ser esclarecido posteriormente



Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

Conteúdo:

**Analisando o Caso  
de Uso**

# Um BOM caso de uso...








- ➡ Corresponde a um processo elementar da empresa
- ➡ **NÃO** é um passo único como "deletar um item" ou "imprimir um relatório".
- ➡ **NÃO** leva dias ou múltiplas sessões, como "negociar um contrato".
- ➡ É uma tarefa concluída em uma sessão e que produz um resultado mensurável deixando as informações em um estado consistente.

# Como descobrir casos de uso?

- ➡ Estabeleça o limite do sistema: o que está fora e o que está dentro?
- ➡ Descubra os atores (externos ao limite do sistema) que realizam os processo básicos
- ➡ Entreviste-os para descobrir mais informações sobre seus objetivos
- ➡ Possivelmente a cada objetivo corresponderá um caso de uso

# Decomposição Funcional

## Sintomas

-  Casos de Uso pequenos
-  Muitos Casos de Uso
-  Dificuldade de entender o modelo
-  Nomes com operações de baixo nível
  -  Operação+objecto
  -  Função+dados
  -  Exemplo: Inserir Cartão

# Decomposição Funcional



## Ações Corretivas

- Busque um contexto mais amplo
  - Por que o sistema está sendo feito?
- Se coloque no papel do usuário
  - O que o usuário quer alcançar?
  - Que valor esse caso de uso adiciona?

# Caso de Uso: Comprar produtos

1. O cliente passa no caixa com os produtos.
2. O funcionário identifica cada um dos produtos.
3. O funcionário indica que todos os produtos foram identificados.
4. O sistema informa o valor total da venda.
5. O cliente realiza o pagamento:
  1. Dinheiro: Ver variante 5.1
  2. Cheque: Ver variante 5.2
  3. Cartão de crédito: Ver variante 5.3
6. O funcionário conclui a venda.



# Variantes

## 5.1: Dinheiro

5.1.1. O cliente entrega a quantia em dinheiro.

5.1.2. O funcionário registra a quantia.

5.1.3. O sistema informa o troco.

5.1.4. O funcionário entrega o troco ao cliente.

## 5.2: Cheque

5.2.1. O cliente entrega o cheque.

5.2.2. O funcionário solicita a presença do gerente.

5.2.3. O gerente dá o visto no cheque.

## 5.3: Cartão de Crédito

...

# Variantes em Casos de Uso Diferentes

- ➡ Caso de Uso Pagar Aluguel
- ➡ 1. O cliente se identifica.
- ➡ 2. O funcionário identifica o aluguel a ser pago, informado o valor ao cliente.
- ➡ 3. O cliente realiza o pagamento:
  - ➡ 1. Dinheiro: Ver Caso de Uso "Comprar Produtos" variante 5.1
  - ➡ 2. Cheque: Ver Caso de Uso "Comprar Produtos" variante 5.2
  - ➡ 3. Cartão de crédito: Ver Caso de Uso "Comprar Produtos" variante 5.3
- ➡ 4. O cliente vai embora.

Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

Conteúdo:

**Como Levantar  
Casos de Uso**

# Como fazer casos de uso

1. Identifique os atores e seus objetivos
2. Para cada caso: escreva um caso simples
3. Para cada caso: escreva as condições de falha e extensões
4. Para cada condição de falha: descreva o que acontece até que volte ao norma ou acabe (em falha)
  - Resolva as falhas
5. Detalhe as variações de dados

# 1) Identifique os atores e seus objetivos

- ➡ Quais computadores, subsistemas e pessoas vão dirigir o sistema
  - ➡ Um ator é qualquer coisa com comportamento
- ➡ O que cada ator precisa que o sistema faça
  - ➡ Cada necessidade mostra um gatilho do sistema
- ➡ Resultado
  - ➡ Lista de casos de uso
  - ➡ Visão geral do sistema
  - ➡ Lista pequena e usável das funções do sistema

## 2) Para cada caso: escreva um caso simples

- ➡ O objetivo é alcançado
- ➡ O cenário principal de sucesso
  - "caso do dia feliz"
  - Mais fácil de ler e entender
  - Qualquer coisa a mais é uma complicação
- ➡ Capture a intenção e responsabilidade de cada ator, da ativação até alcançar o objetivo
  - Diga que informação é passada entre atores
  - Numere cada linha
- ➡ Resultado
  - Descrição legível das funções do sistema

### 3) Escreva as condições de falha e extensões

- ➡ Normalmente, cada passo pode falhar
- ➡ Anote cada condição de falha separadamente após o cenário principal de sucesso
- ➡ Resultado:
  - ▬ Lista de cenários alternativos

## 4) Resolva as falhas.

➡ Extensões recuperáveis voltam ao caso principal

➡ Extensões não-recuperáveis falham

➡ Cada cenário vai do gatilho ao fim

- ▬ Extensões são apenas uma forma resumida de escrever

- ▬ Pode escrever "se"

- ▬ Pode escrever cenário do início ao fim

➡ Resultado:

- ▬ Casos de uso completos



## 5) Detalhe as variações de dados

➡ Algumas extensões são muito "baixo nível" para fazer agora

- Ex: Reembolse comprador

- Como? Cheque, dinheiro, etc.?

➡ Adie variações que podem ser tratadas por casos de uso de menor abstração

# Boas Perguntas

- ➡ Quais são as tarefas de um ator?
- ➡ O ator precisa ser informador de certas ocorrências dentro do sistema?
- ➡ O ator precisa informar o sistema de mudanças externas?
- ➡ O sistema fornece ao negócio o comportamento adequado?

# Boas Perguntas

- ➡ Todos os requisitos funcionais foram atendidos pelos casos de uso?
- ➡ Que casos de uso vão suportar e manter o sistema?
- ➡ Que informação precisa ser modificada ou criada?

# Casos de Uso Especiais

- ➡ Início e Parada do sistema
- ➡ Manutenção do sistema
- ➡ Manutenção da informação
  - ▬ Normalmente aparece mais tarde
- ➡ Adicionar nova funcionalidade a sistema funcionando
  - ▬ Sistemas que não podem parar
- ➡ Portar o sistema rodando para um novo ambiente
  - ▬ Quando o ator é a organização desenvolvedora

# Comentários

- ➡ O valor dos casos de falha é detectar situações anormais e manter a completude
- ➡ Todo cenário vai do início ao fim (sem "ses"), mas a descrição pode ser abreviada
- ➡ Os requisitos cobrem as falhas recuperáveis ou não
  - ▬ Mas não são falhas do sistema interno, mas do ambiente
- ➡ O "cenário ideal" ajuda a descrever as falhas

# Comentários

- ➡ Um cenário pode se referir a objetivos de nível inferior
  - ▬ Caso de uso subordinados
  - ▬ Funções comuns
- ➡ Um caso de uso superior só se interessa se o caso de uso inferior alcança o sucesso ou falha
  - ▬ Não analisa os detalhes
- ➡ Cada passo de um cenário é um sub-objetivo
  - ▬ Esconde um sub caso de uso
  - ▬ Pode ser tão profundo que não é descrito
- ➡ Cada sentença em cada nível é um objetivo

# Casos de uso NÃO

- ➡ Mostram requisitos de interface
  - ▢ Colete por caso de uso
- ➡ Mostram requisitos de desempenho
  - ▢ Conecte-os ao caso de uso
- ➡ Coletam fórmulas, estados, cardinalidades
  - ▢ Capture separadamente

Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

Conteúdo:

**Relacionamentos  
dos Casos de Uso**

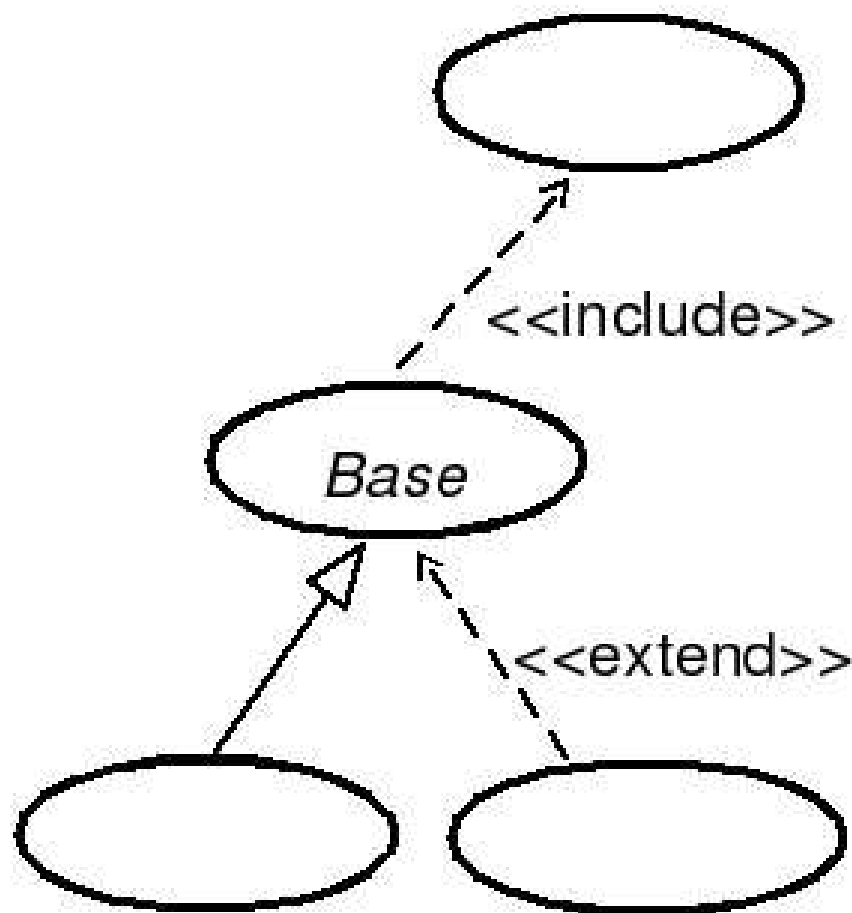


# Tipos de Relacionamentos

➡ Estende

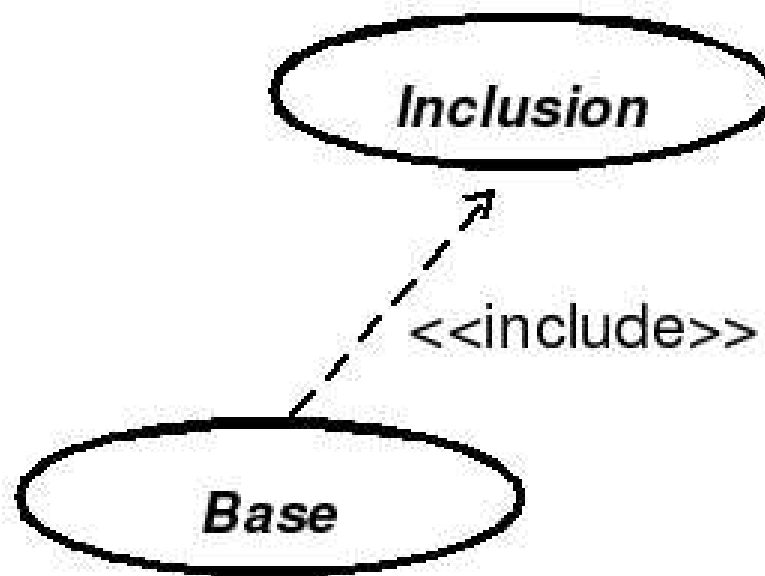
➡ Inclui

➡ Generaliza

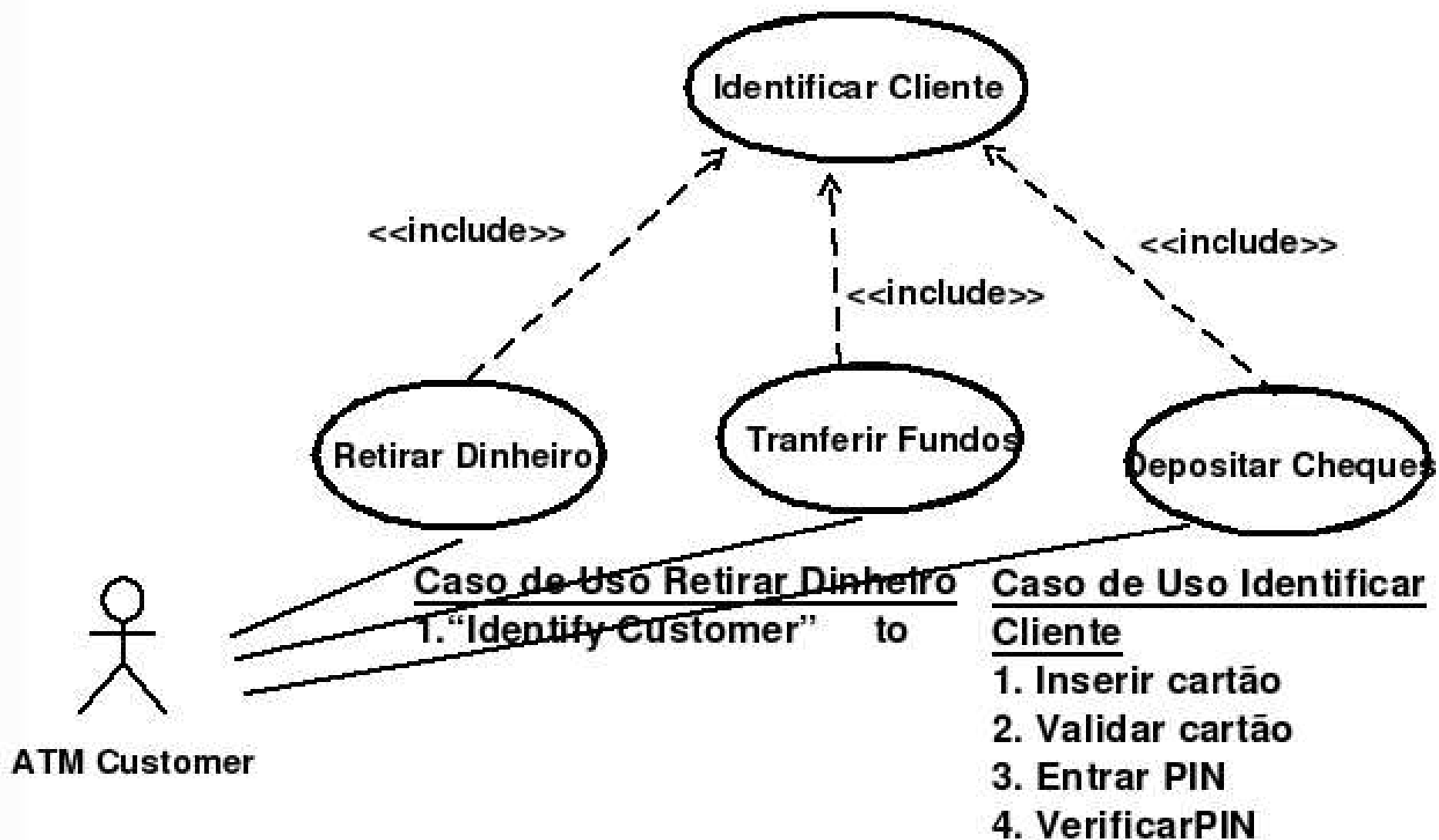


# Inclui

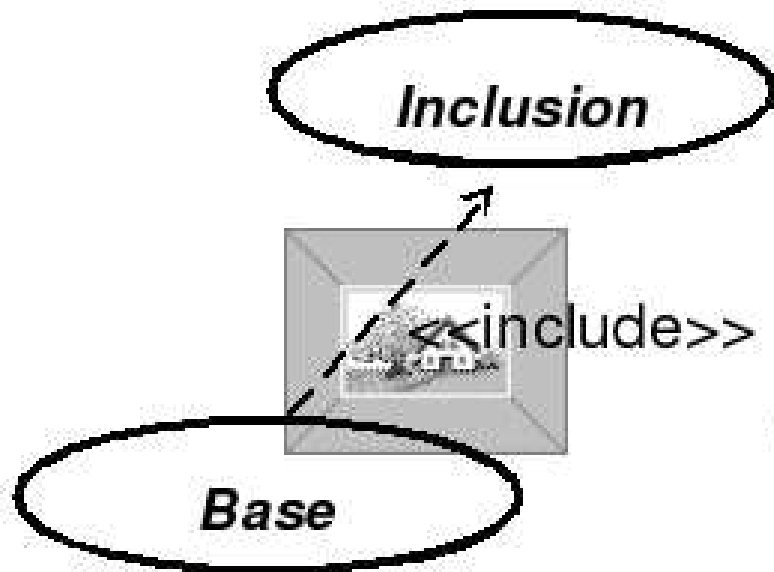
- ➡ É uma relação entre um caso de uso básico e um caso de uso incluído
- ➡ Significa que o caso de uso é explicitamente inserido no caso de uso base



# Include : Exemplo



# Por que usar Inclui?



- Fatorar comportamento comum entre dois ou mais casos de uso
  - Evita descrever o mesmo comportamento duas vezes
  - Garante consistência
- Fatorar e encapsular comportamento de um caso de uso base
  - Simplificar fluxo complexo de eventos
  - Fatorar comportamento que não é parte do objetivo primário

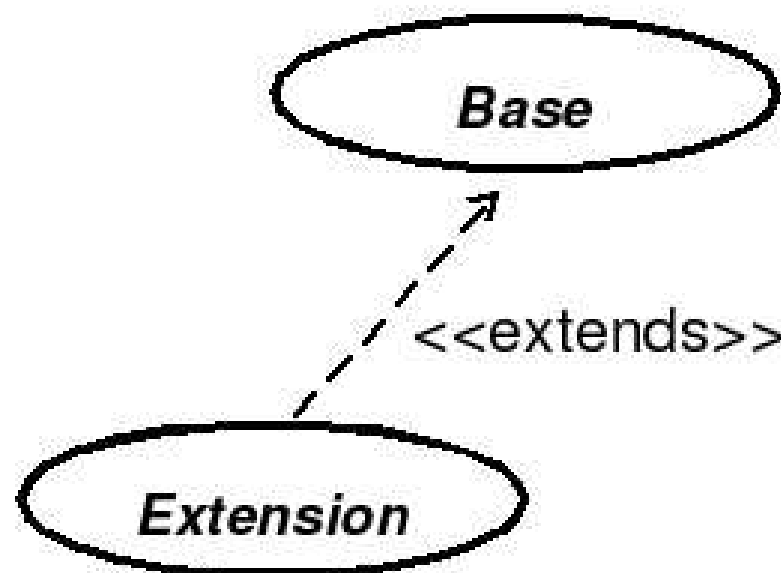
# Executando o Include

- ➡ Executado totalmente quando é chamado
- ➡ Se não deve ser executado, a decisão é do caso de uso chamador

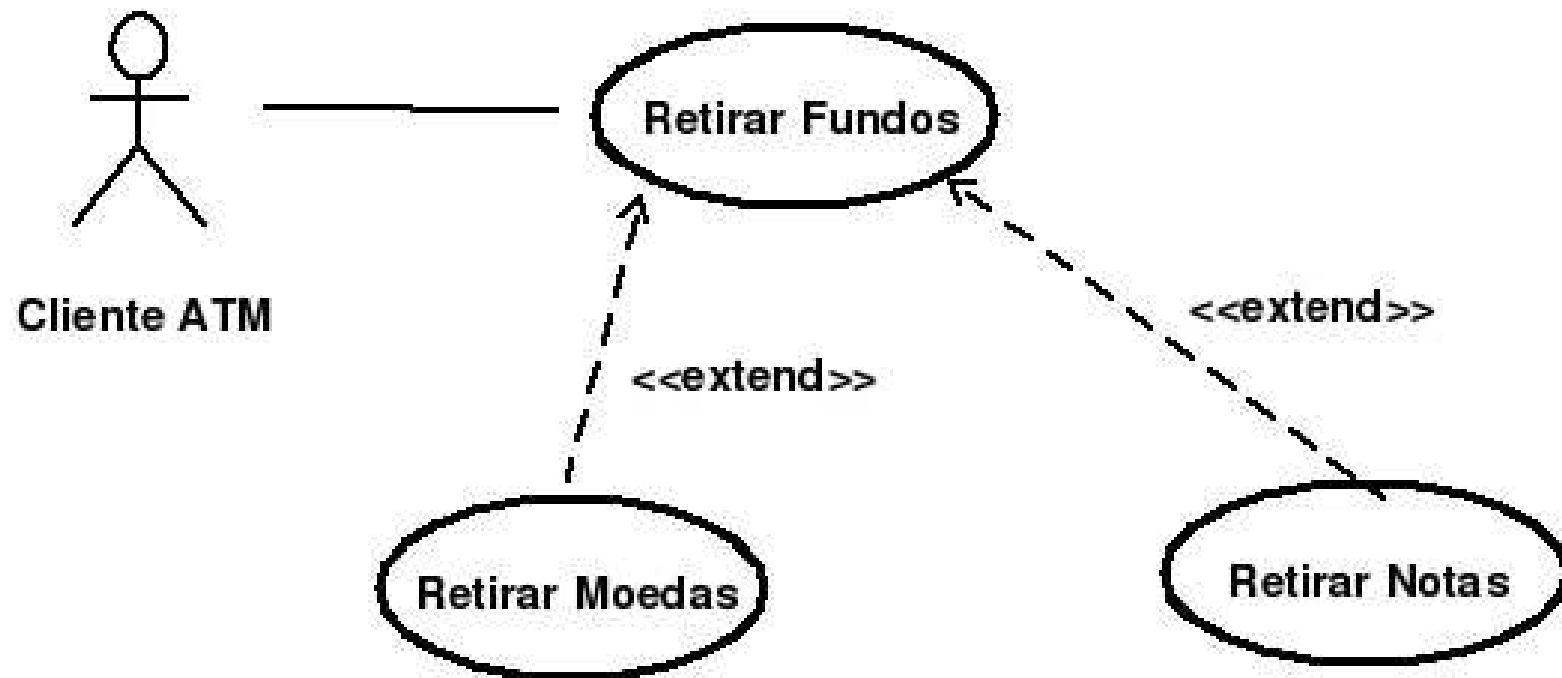
# O que é Estende?

➡ É uma relação entre o caso de uso que estende e seu caso base

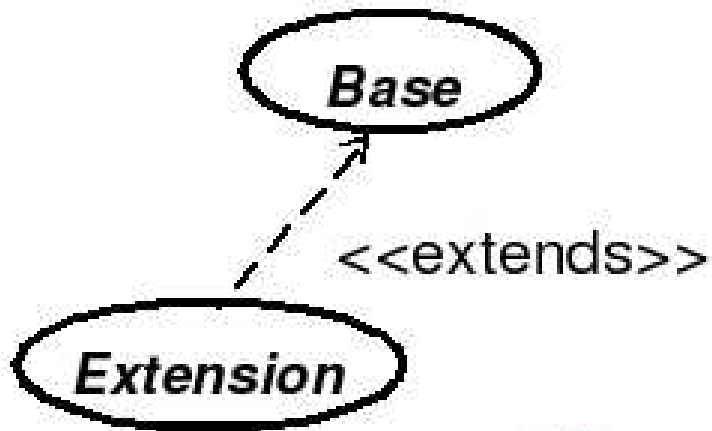
- ➡ Insere uma extensão no comportamento do caso base
- ➡ Inserida apenas se uma condição de extensão for verdadeira
- ➡ Inserida em um ponto de extensão



# Exemplo



# Por que usar Extende?



Fatorar comportamento

excepcional ou opcional

- Executado em apenas algumas condições
- Simplificando o fluxo de eventos base
  - Exemplo: alarme



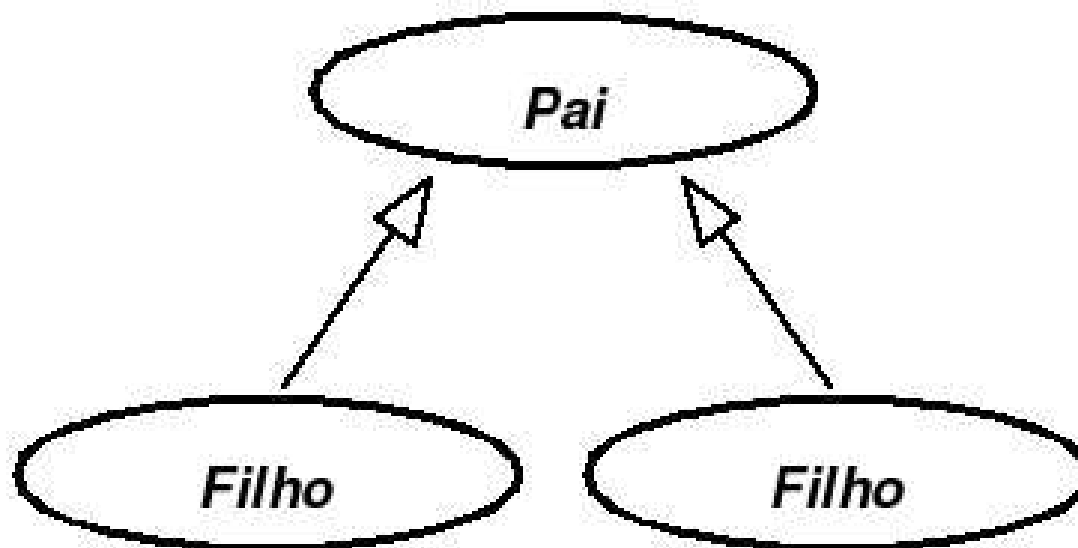
Extender o comportamento

- Comportamento adicional criado separadamente
  - Nova versão

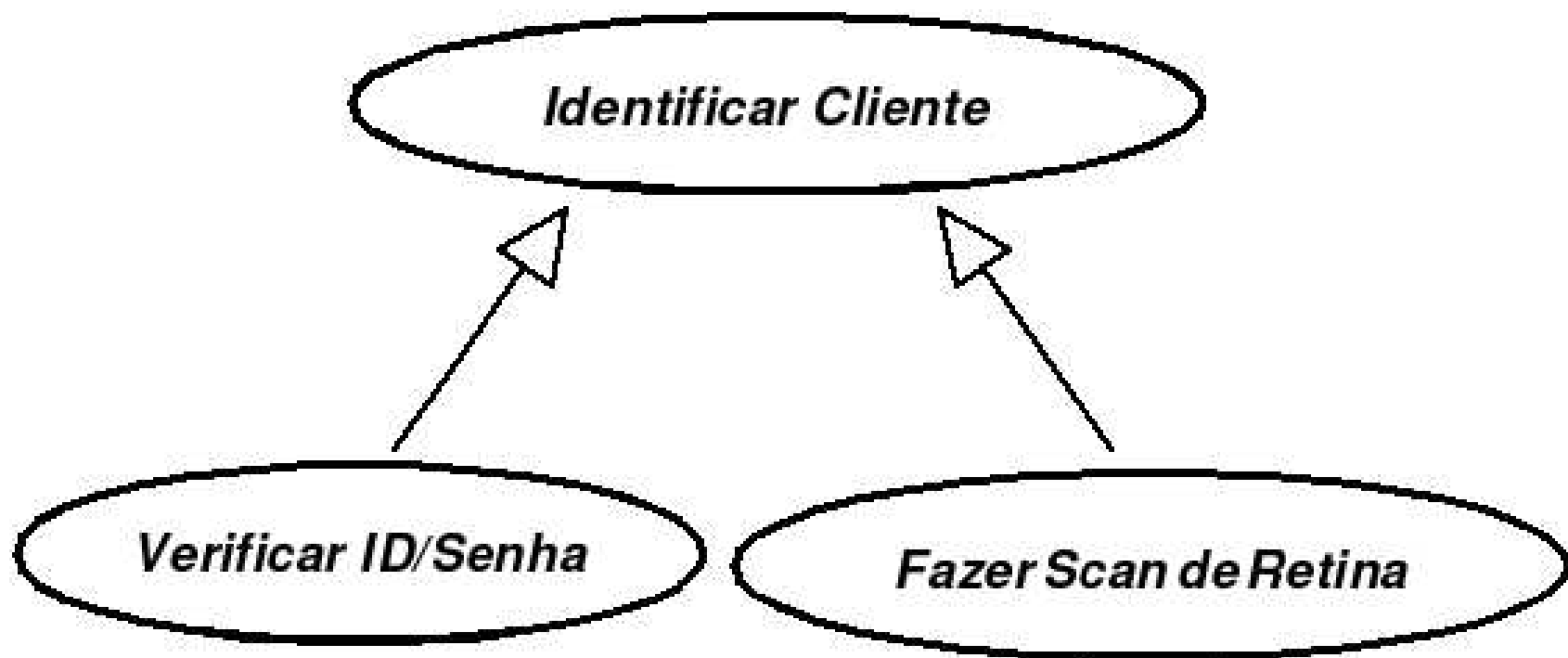


# Generalização

- ➡ É um relacionamento de um "filho" para um "pai"
- ➡ Descreve um comportamento geral compartilhado com o parente
  - ➡ Descreve que o "filho" tem um comportamento especializado



# Exemplo



# Por que usar Generalização?

- ➡ Mostrar
  - Comportamento comum
  - Estrutura comum
  - Objetivo Comum
- ➡ Mostrar que os "filhos" são parte de uma família de casos de uso
- ➡ Evitar descrever o mesmo comportamento em muitos casos de uso
- ➡ Assegurar que comportamento comum se mantém consistente

# Executando Generalização

- ➡ Uma instância de caso de uso executando um caso especializado vai seguir o fluxo de eventos descritos pelo caso parente, inserindo comportamento adicional e modificando seu comportamento como definido no fluxo de eventos do caso especializado

# Tipos de Casos de Uso



## Concreto

- Tem que ser completos e úteis
- São "instanciados" diretamente



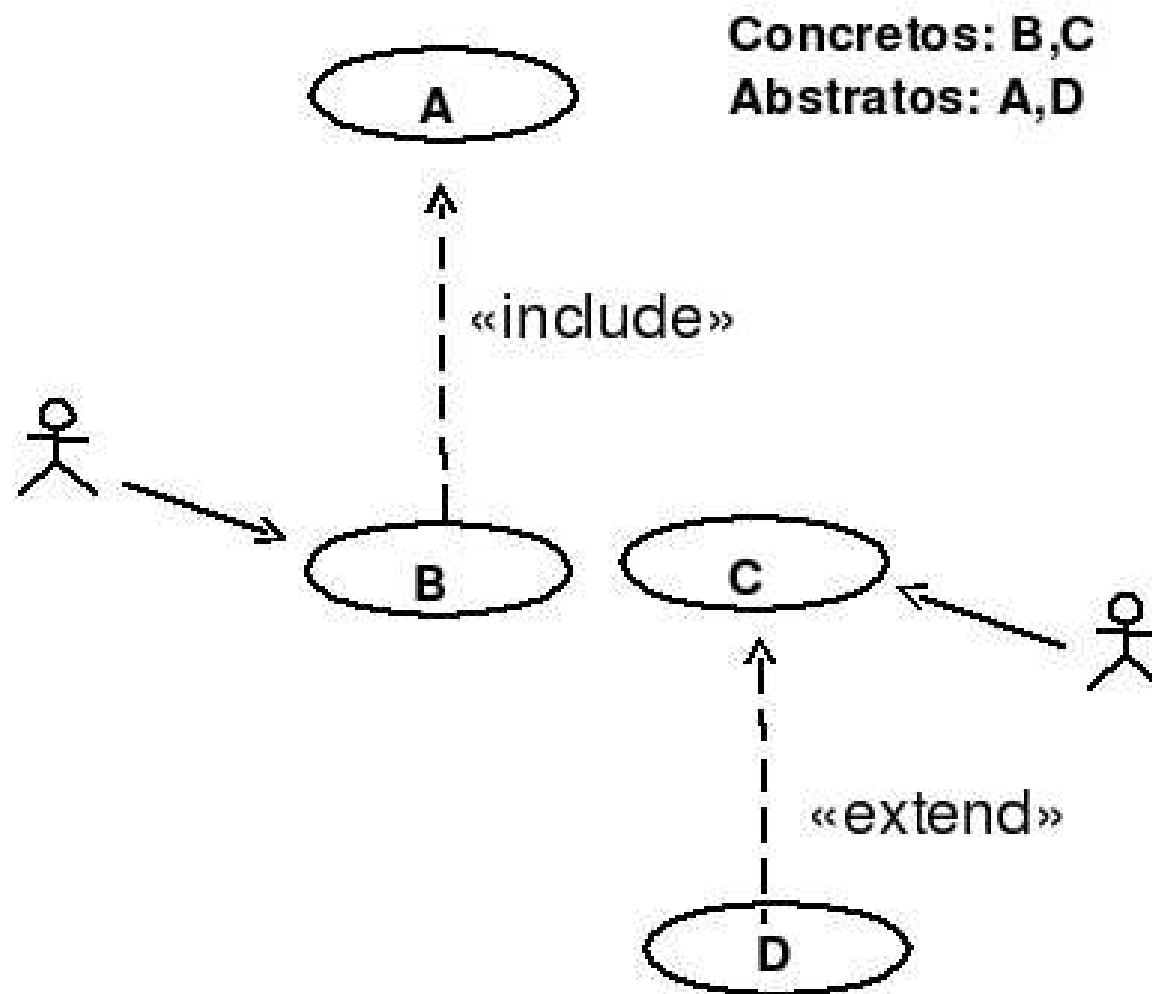
## Abstrato

- Não precisam ser completos
- Só existem para ajudar outros casos de uso
- Nunca são "instanciados"



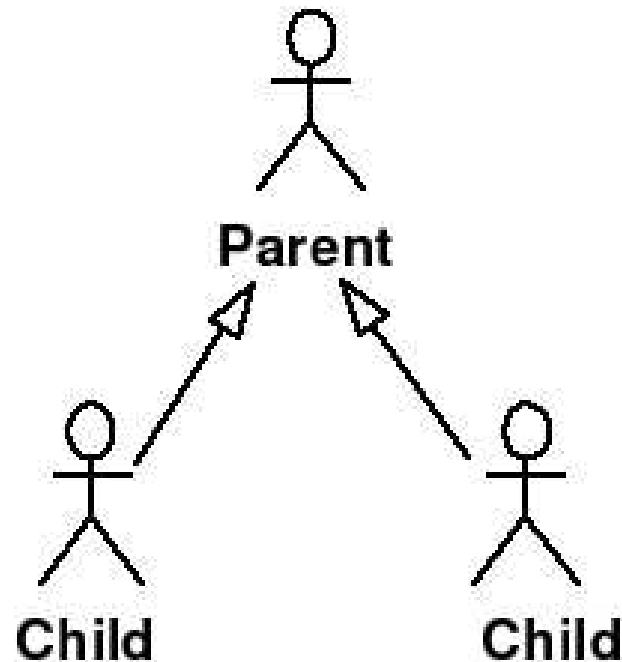
Dica: eliminando os casos de uso abstratos,  
ainda compreendemos o sistema

# Concretos x Abstratos

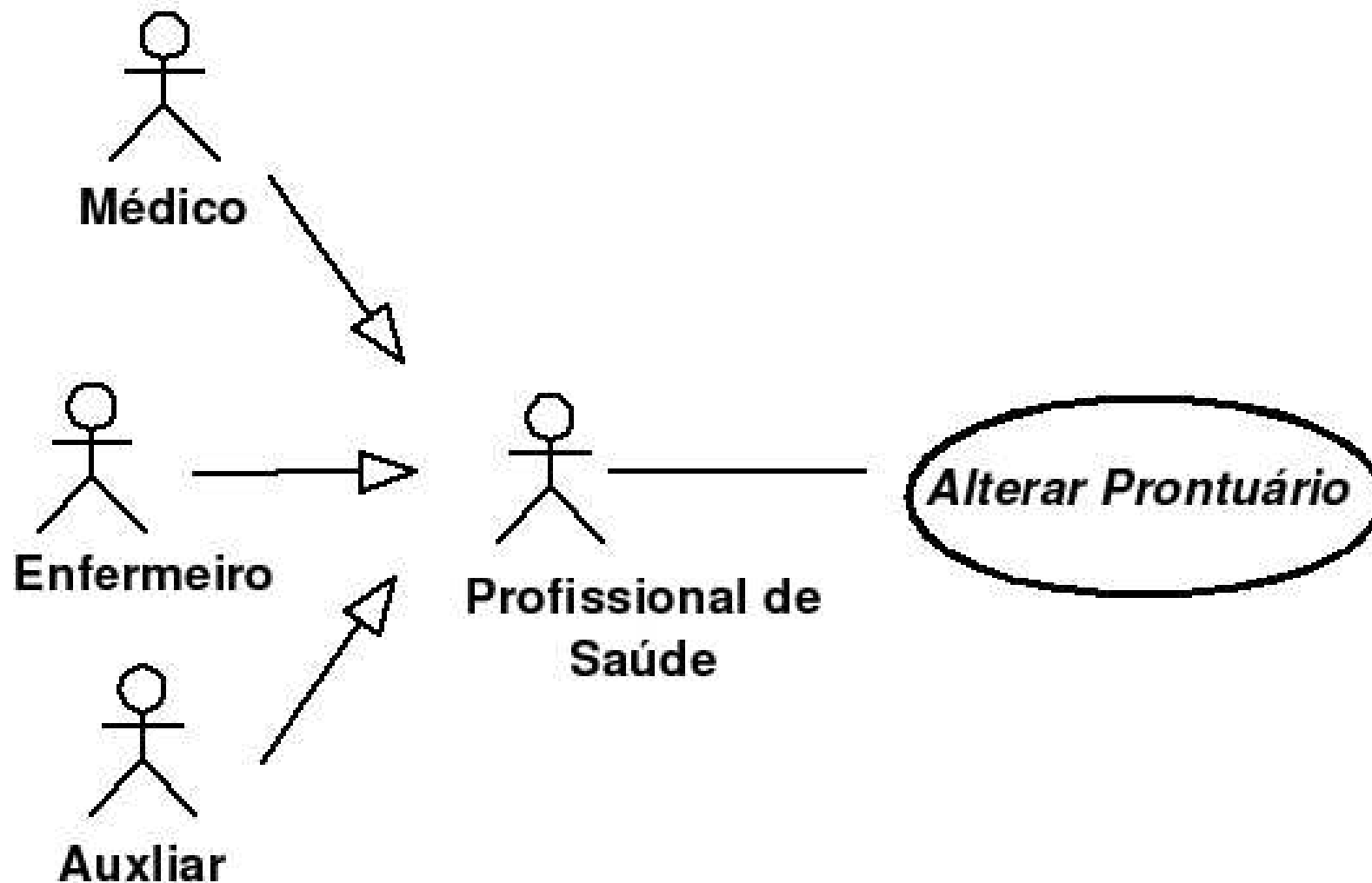


# Generalização de Atores

- ➡ Atores podem ter características comuns
- ➡ Múltiplos atores podem ter papéis comuns ao interagir com um caso de de uso



# Exemplo





# Por que usar generalização

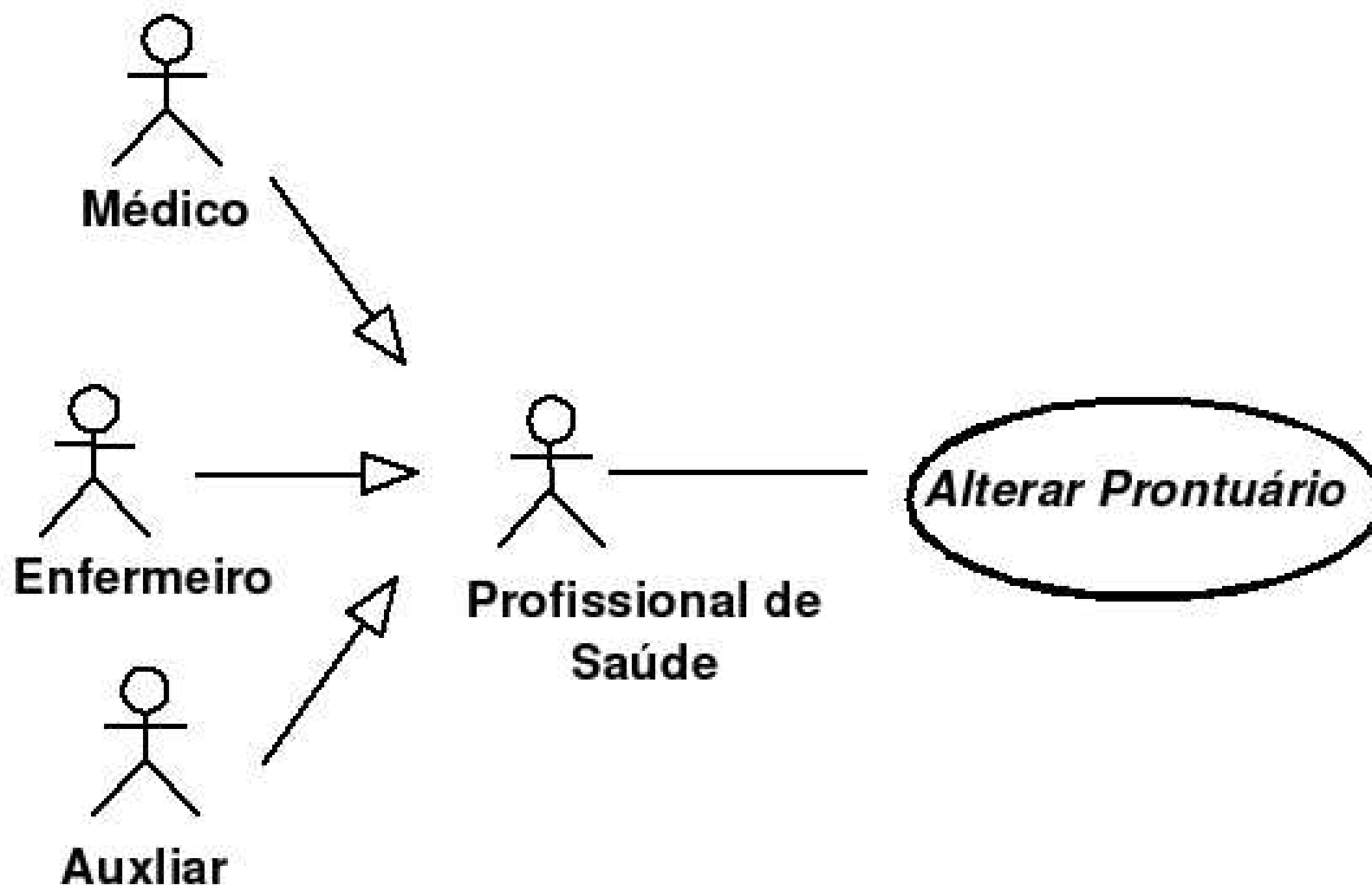
- ➡ Simplificar relações entre muitos atores e um caso de uso
- ➡ Mostrar que uma instância de um ator especializado por fazer tudo que outro tipo de ator faz



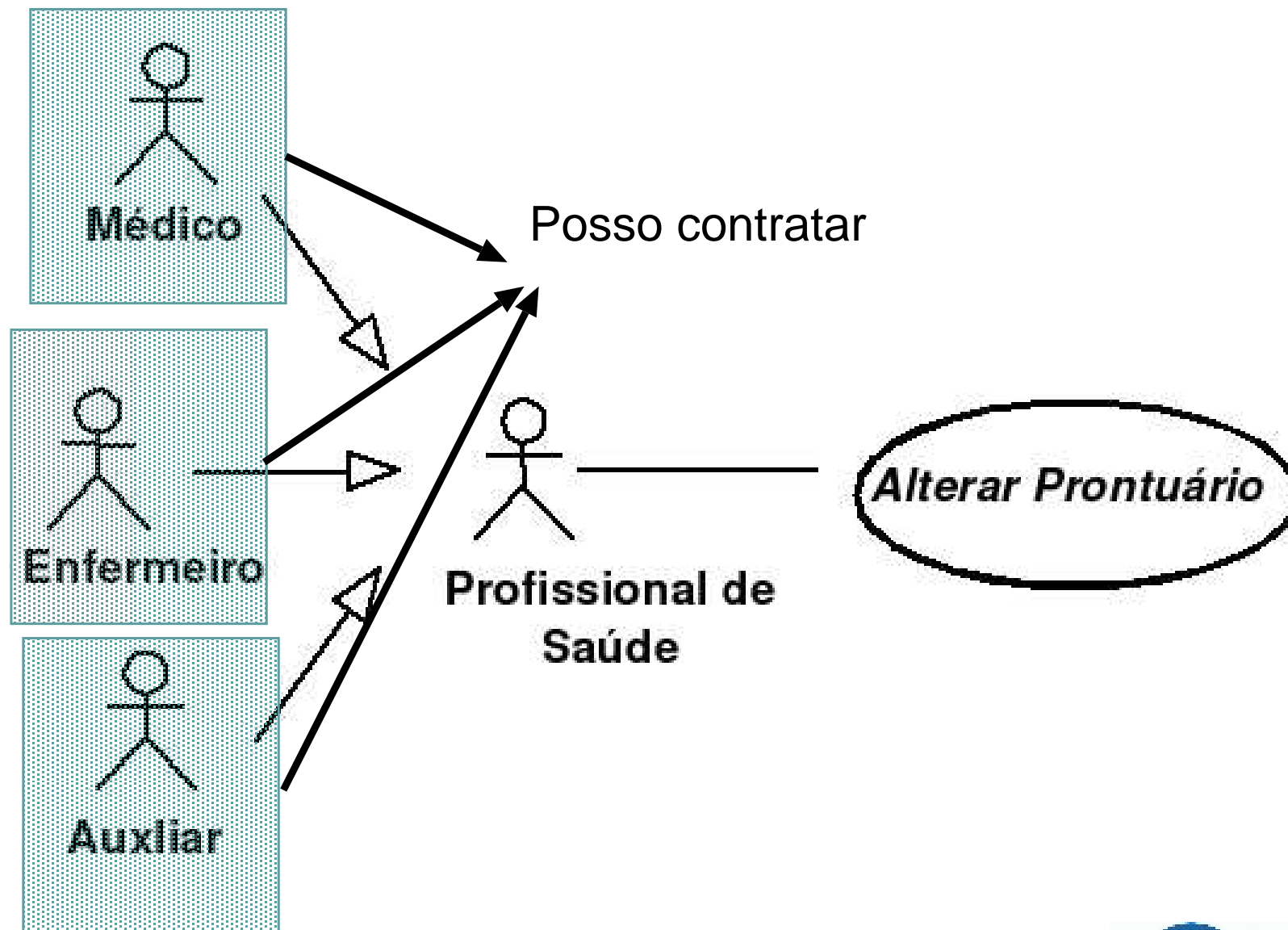
# Atores Abstratos e Concretos

- ➡ Um ator abstrato contém a parte comum dos papéis
  - ⇒ Não pode ser instanciado
- ➡ Um ator concreto pode ser instanciado

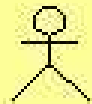

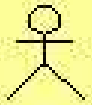

# Concreto x Abstrato (cont.)







# Concreto x Abstrato (cont.)



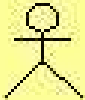

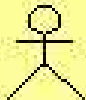

# Relacionamentos

<div>de para</div>		
		
		





# Relacionamentos

<div>de para</div>		
	generalização	
		

# Relacionamentos



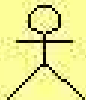

<div>de para</div>		
	generalização	Se comunica com
		

# Relacionamentos

<div>de para</div>		
	generalização	Se comunica com
	Se comunica com	



# Relacionamentos

<div>de para</div>		
	generalização	Se comunica com
	Se comunica com	<<include>> <<extend>> generalization

# Priorizando Casos de Uso

- ➡ Que casos de uso devem ser implementados?
  - ▬ Associar os casos de uso aos requisitos originais
- ➡ Em que seqüência devem ser implementados?
  - ▬ Selecionar os casos de uso para iterações de arquitetura
    - ▬ Que representem funcionalidade central significativa
    - ▬ Que cubram grande parte da arquitetura
    - ▬ Que forcem ou ilustrem um ponto específico e delicado da arquitetura
  - ▬ Priorize os casos de uso/cenários para iterações futuras

## Aula 28

### Professor:

Geraldo Xexéo  
DCC/IM/UFRJ  
PESC/COPPE/UFRJ

### Conteúdo:

**FIM: Casos de Uso**