



# Modelagem com UML

Fabio Perez Marzullo

[fabio@mz-empresarial.com.br](mailto:fabio@mz-empresarial.com.br)



IEEE Body of Knowledge on Services Computing  
*Sponsored by Technical Committee on Services Computing, IEEE Computer Society*

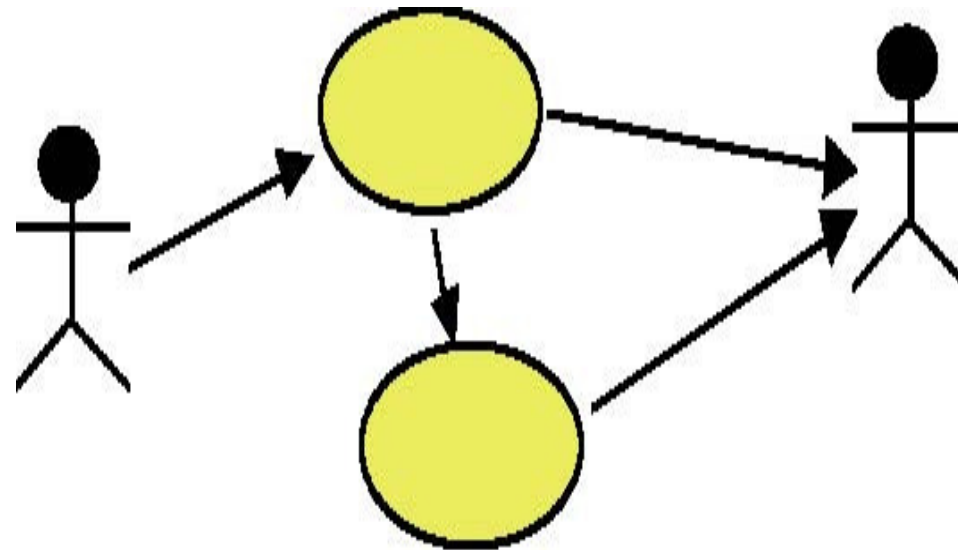


# Modelagem Estática

## (Visão Lógica)



- Modelagem do Comportamento do Sistema





# Casos de Uso

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- representa uma função completa do sistema:

inclusão + alteração + remoção + consulta

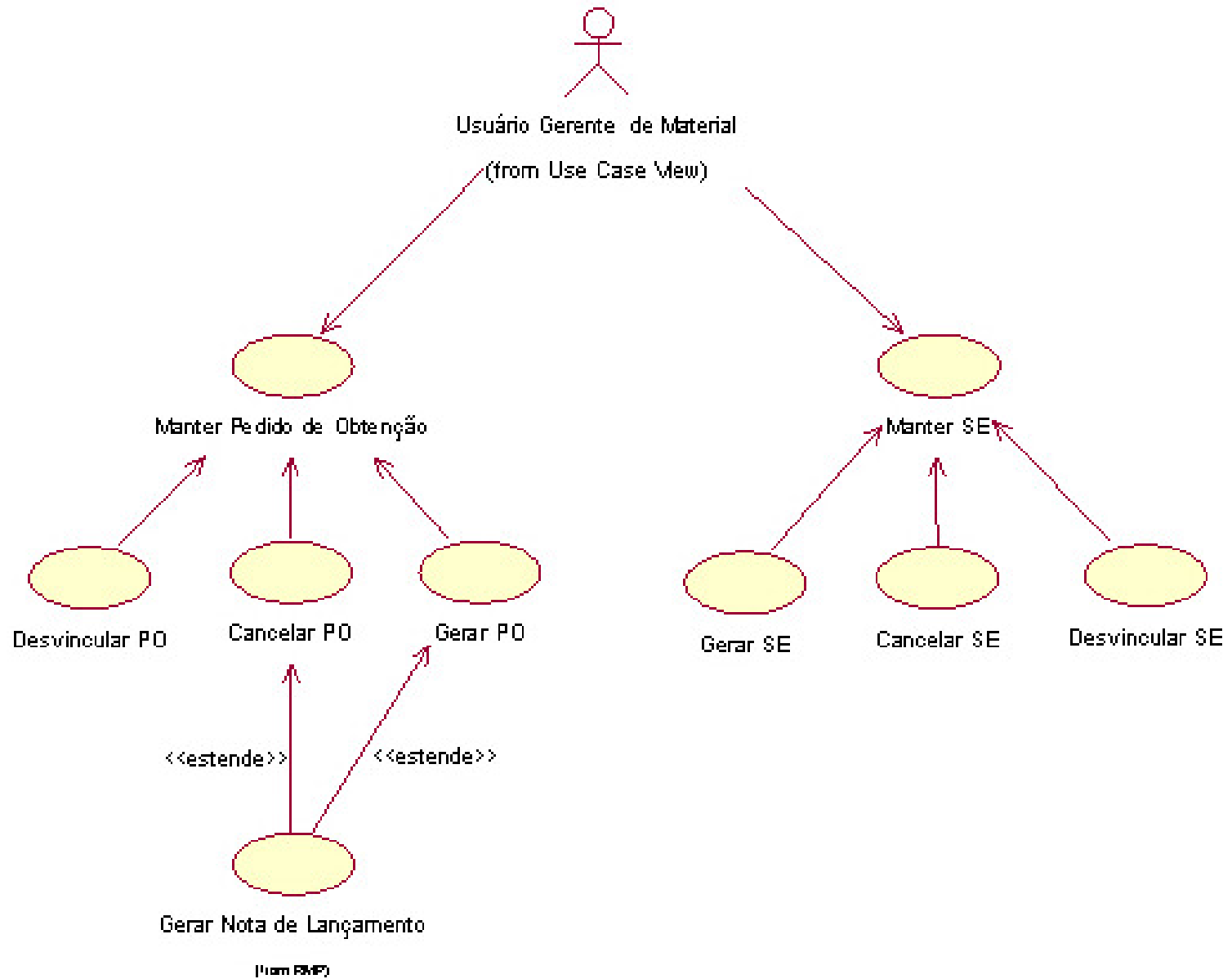
=

manutenção

*Manutenção = um único caso de uso.*



# Casos de Uso





# Descrevendo Casos de Uso

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Condição de início do caso de uso.
- O fluxo de ações do caso de uso.
- Subfluxos do caso de uso.
- Fluxos de tratamento de erro.
- Variantes do fluxo de ações.
- Documentos que descrevem todos os casos de uso normalmente são longos!



# Atores

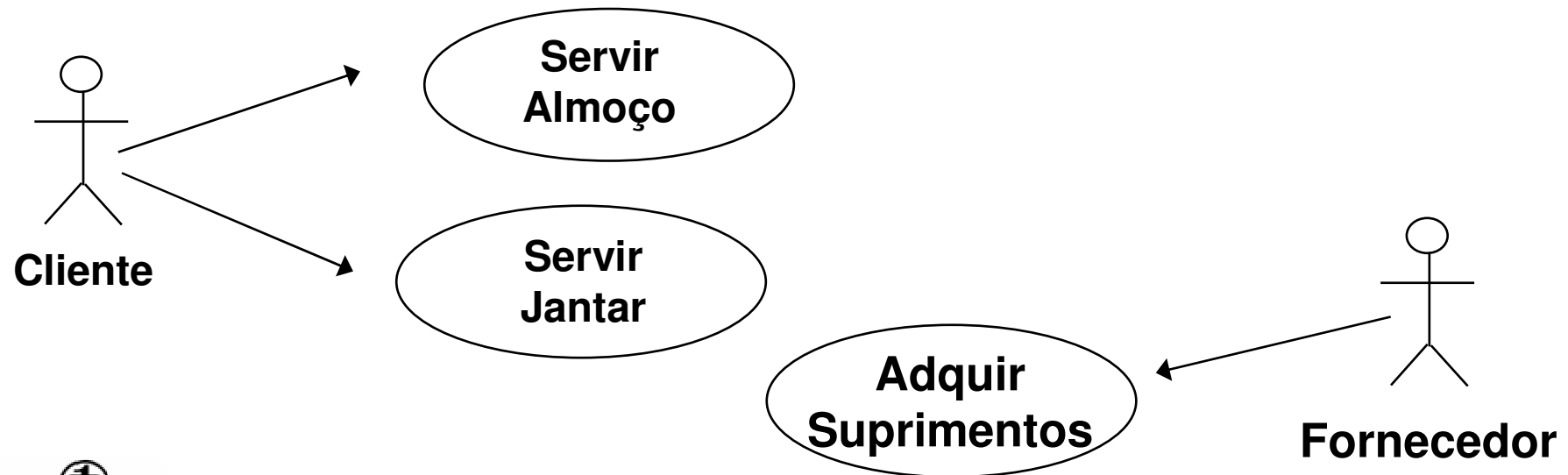
- Atores representam qualquer elemento que possa interagir com o sistema
- Atores não são parte do sistema, mas representam os elementos externos que interagem com o sistema
- Atores podem ser:
  - Pessoas.
  - Outros sistemas.
  - Equipamentos ligados ao sistema.





## Relação de uso:

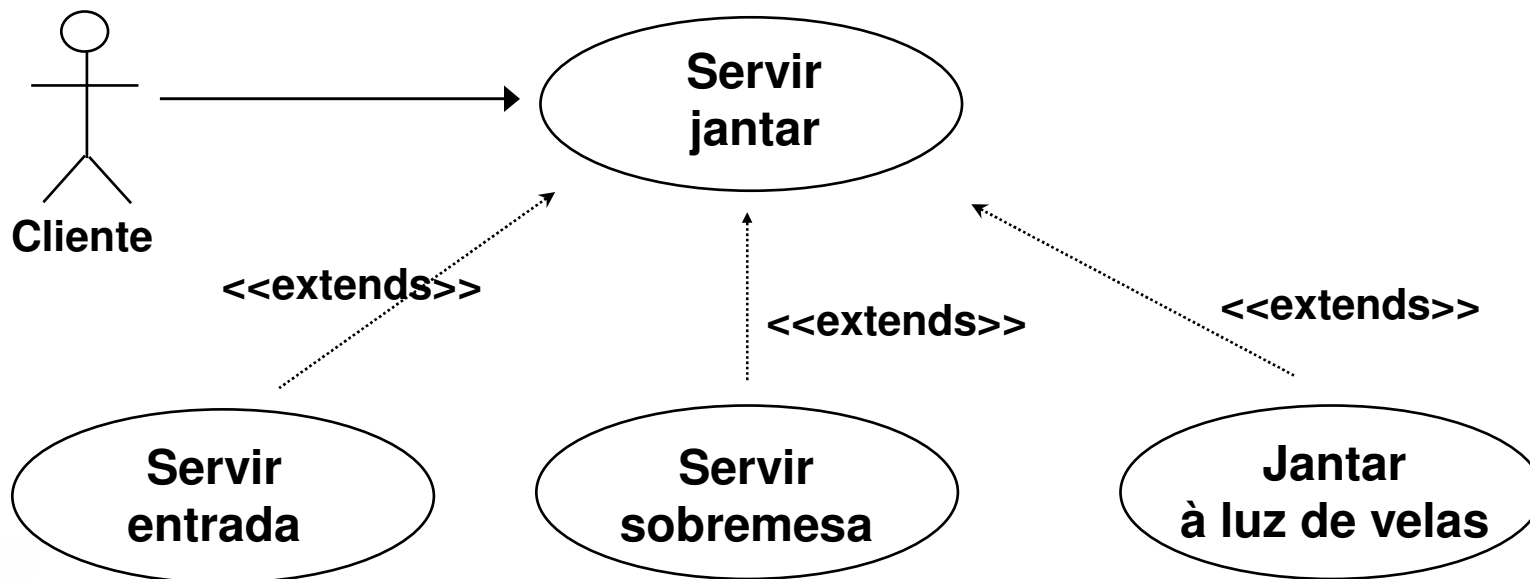
- ✓ Diversos casos de uso compartilham o mesmo comportamento.
- ✓ O comportamento é fatorado em um caso de uso independente.
- ✓ O novo caso de uso é reutilizado pelos demais.





## Relação de extensão:

- ✓ Indica comportamento opcional ou dependente de condição.
- ✓ Complementos de um caso de uso.





# Estereótipos

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Objetivo:
  - Estereótipos são nomes especiais, utilizados para descrever categorias de elementos em modelos UML.
  - Estereótipos são representados por nomes entre os símbolos << e >>.
- Nos diagramas de casos de uso, os estereótipos podem ser utilizados para nomear as relações de uso e extensão.



## ✓ Generalização

→ Casos de Uso

→ Atores

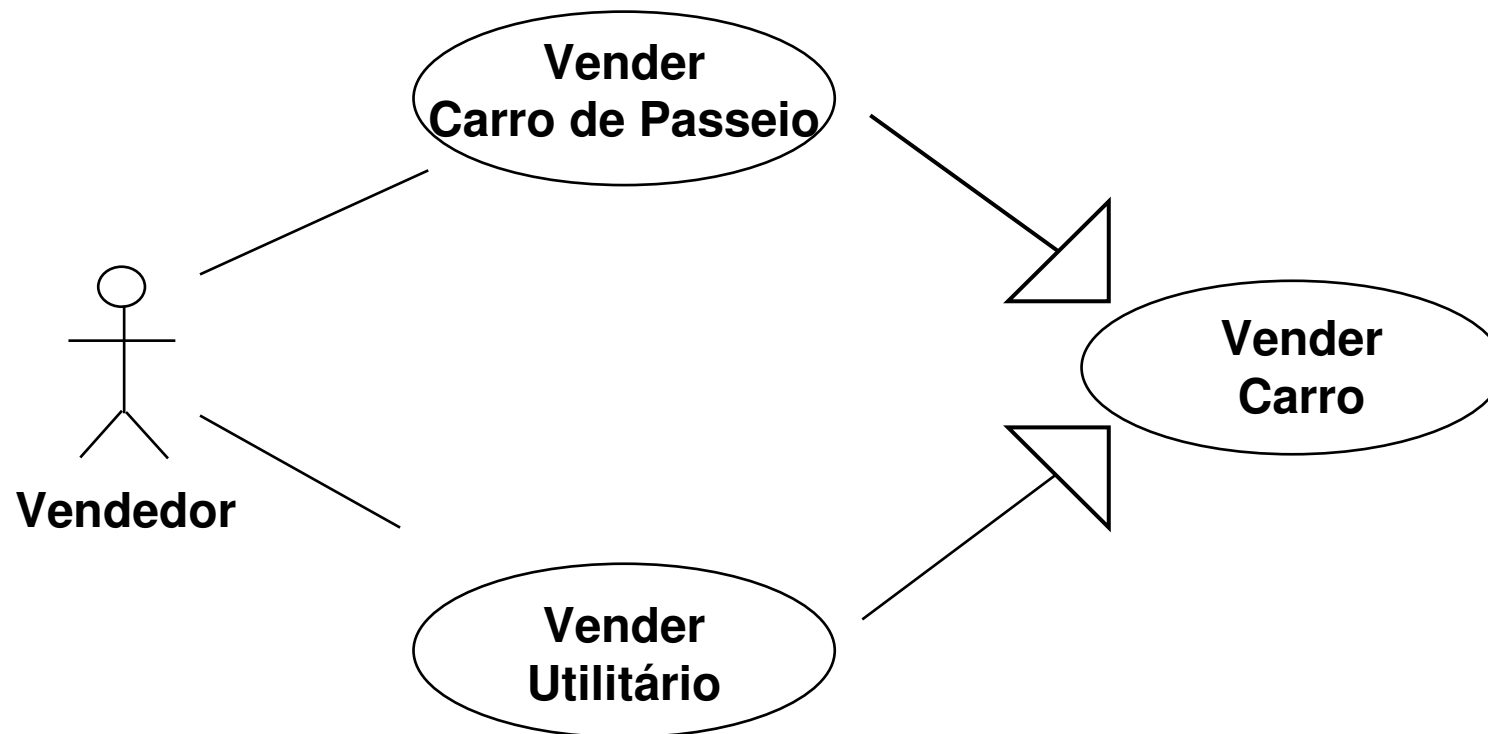
## ✓ Associações de Casos de Uso



# Conceitos Avançados

## Generalização de Casos de Uso

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

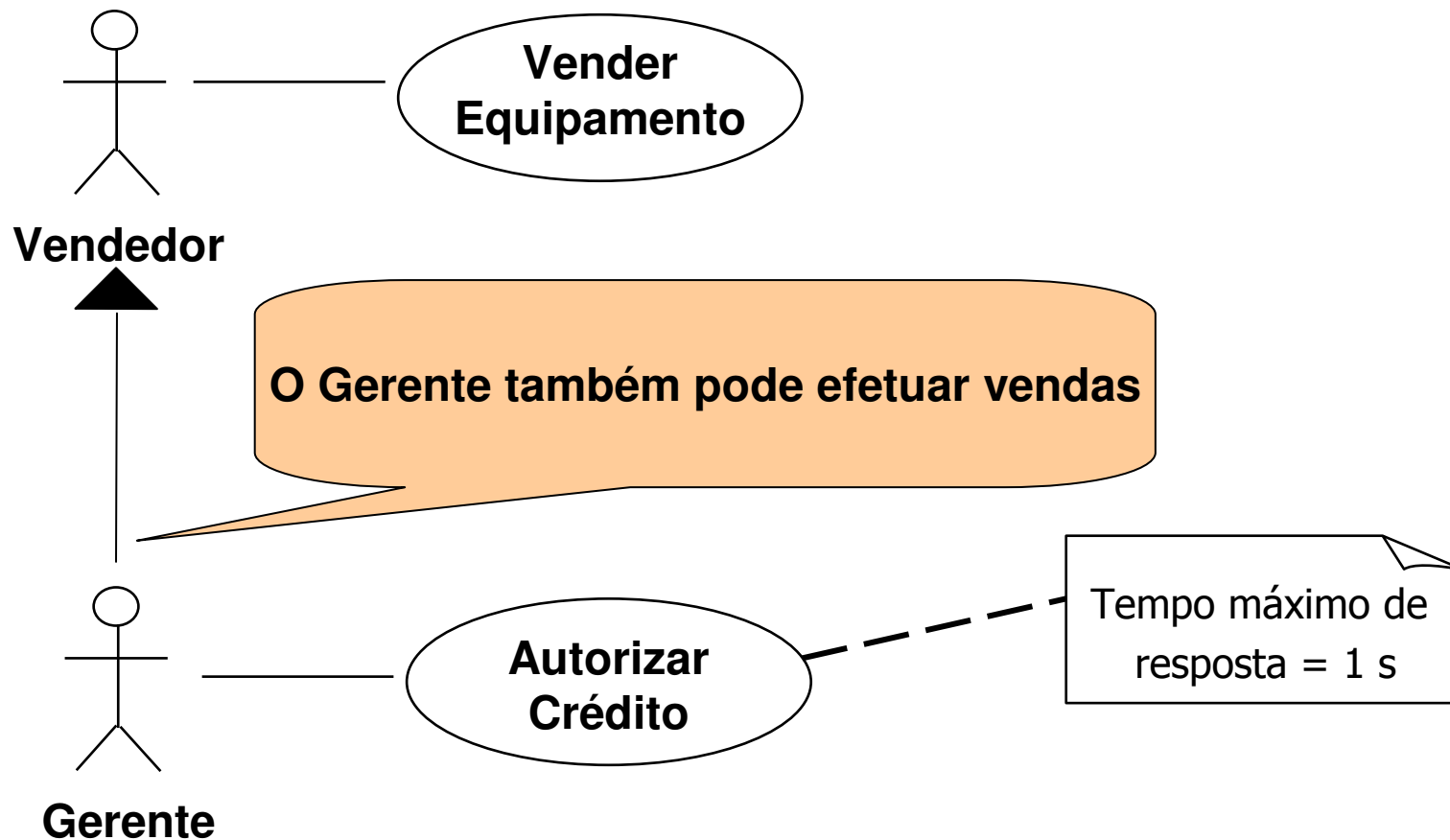




# Conceitos Avançados

## Generalização de Casos de Uso

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica





# Classes e Objetos

## Modelagem da estrutura dos sistemas



# Conceitos Básicos de OO

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

Classe



*Grupo de objetos*



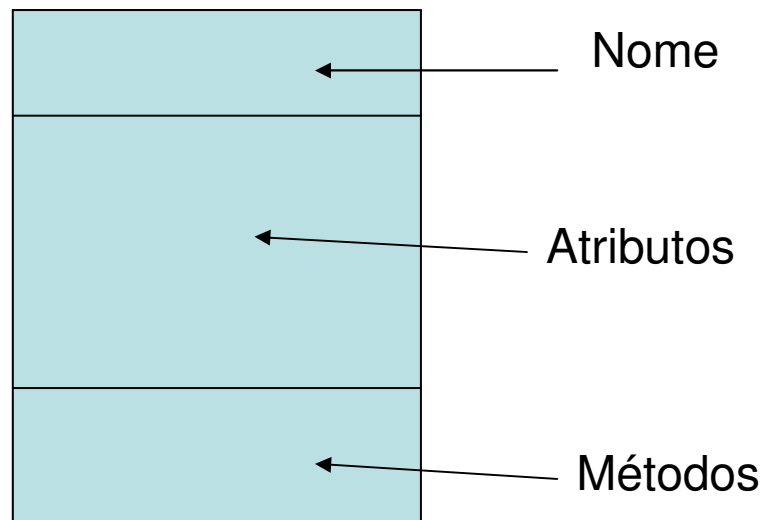
*Propriedades semelhantes.*



*Mesmo comportamento.*



- Em UML, uma classe é representada por um retângulo com três divisões:



- ✓ Nome da classe.
- ✓ Atributos.
- ✓ Métodos.



# Objetos

- Em UML, objetos são representados por retângulos, e sua identidade aparece sublinhada:

Monitor LG, 19",  
Preto

Monitor Samsung,  
22", Touch of Color



# Descobrimo Classes

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

Descobrir as classes de um sistema é a parte mais complexa do desenvolvimento orientado a objetos.



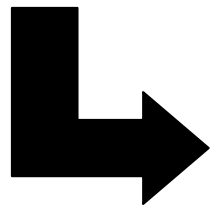
[fabio@mz-empresarial.com.br](mailto:fabio@mz-empresarial.com.br)



O domínio da aplicação:

- Uma boa classe captura um único conceito do domínio da aplicação.

## Domínio



- *Domínio da Aplicação*
- *Domínio do Negócio*
  - *Relacionamentos*
  - *Papéis*
  - *Atributos*



- *Domínio da Arquitetura*
  - *Interface*
  - *BD*
  - *Comunicação*
- *Domínio Básico*
  - *Semântico*
  - *Estrutural*
  - *Fundamental*



# Atributos

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

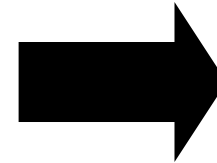
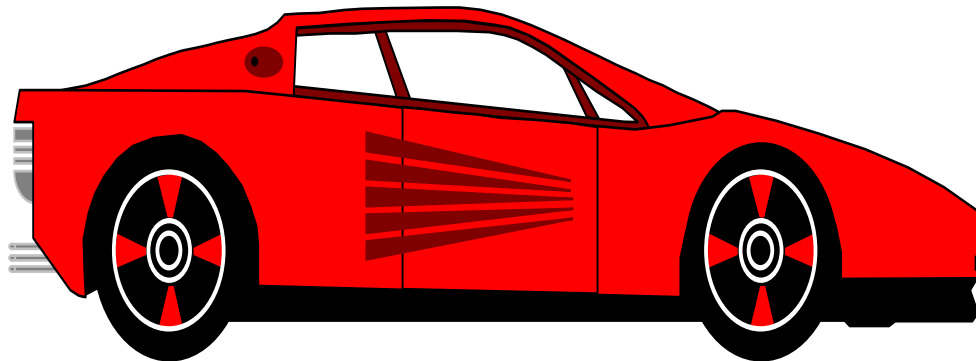
- Os atributos representam as características da classe.
- Cada atributo representa um conceito atômico.
- Atributos compostos devem ser decompostos.
  - Ex: Endereço = Rua, Número e Bairro.
    - Três atributos de uma classe.



# Atributos - Exemplo

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

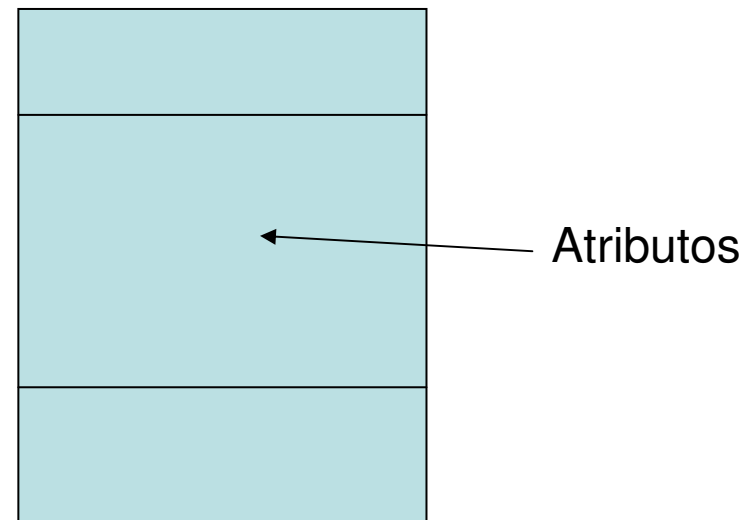
*Exemplo: Instância da Classe Veículo → Carro*



*Placa*  
*Chassi*  
*Potência*



- Em UML, os nomes dos atributos são apresentados no retângulo central do diagrama de classes:





# Visibilidade

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- A visibilidade de um atributo indica as classes (ou objetos) que podem consultar ou alterar seu valor:
  - Visibilidade pública (símbolo +)
    - Qualquer classe pode acessar o atributo
  - Visibilidade privada (símbolo -)
    - Apenas objetos da classe acessam o atributo
  - Visibilidade protegida (símbolo #)
    - Apenas objetos da classe e seus descendentes acessam o atributo



# Associações

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Quando um objeto possui outro objeto como atributo, dizemos que existe uma associação entre as classes dos objetos envolvidos.





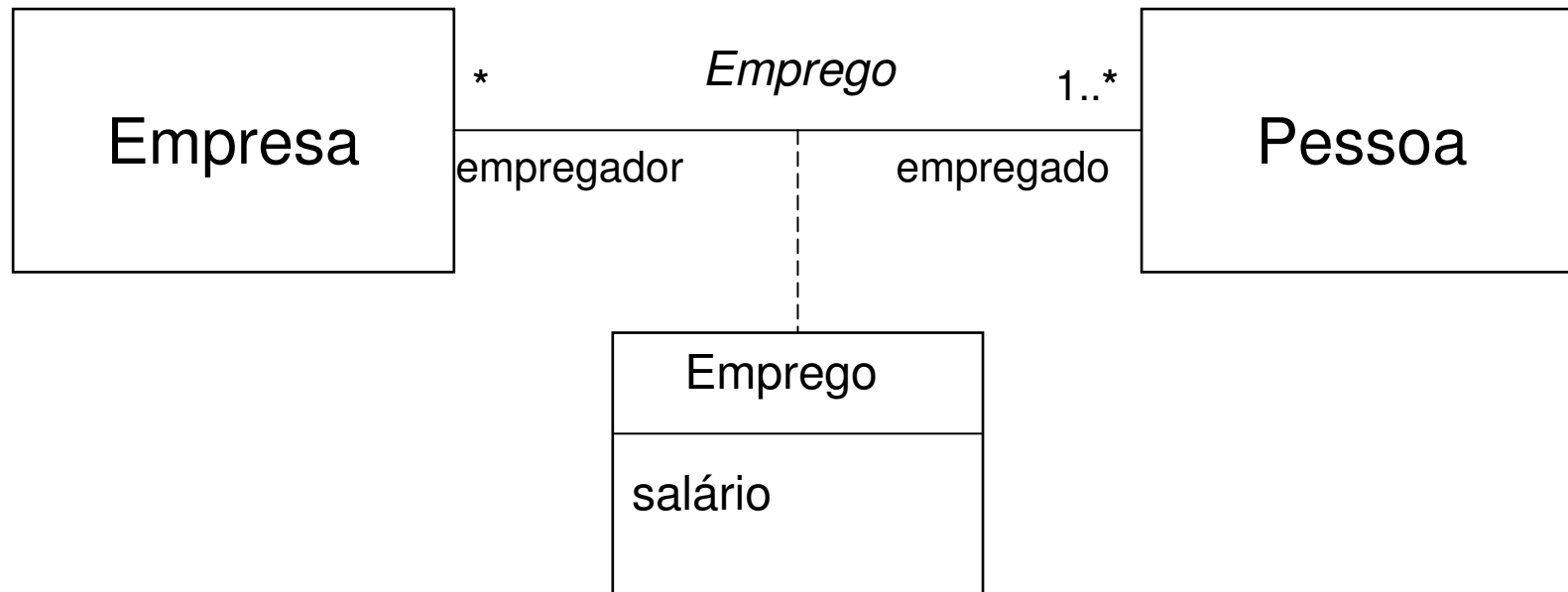
# Componentes e Associações

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Uma associação é representada por uma linha entre as duas classes.
- Uma associação é descrita pelas seguintes informações:
  - nome (opcional);
  - papel (opcional);
  - cardinalidade.



# Associações - Exemplo



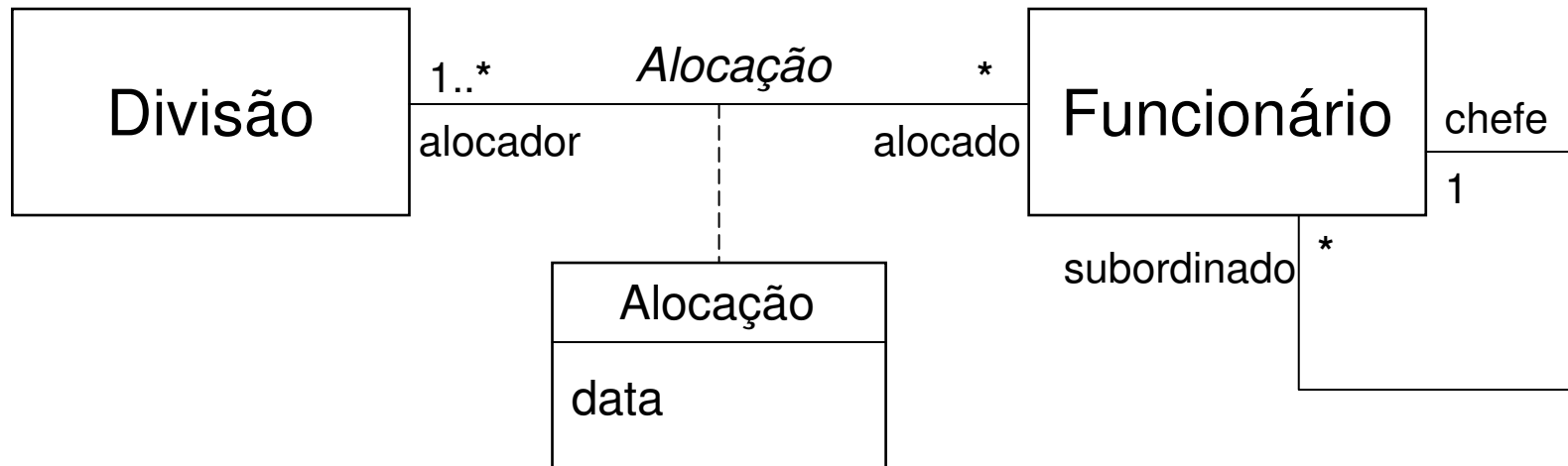


- A cardinalidade indica o número de objetos participando em cada lado da associação:
  - ✓ Indica o número mínimo e máximo de objetos.
  - ✓ Se o máximo for igual ao mínimo, apresenta um único número.
- Tipos comuns de cardinalidades:

Tipo	Significado
1, 0..*	Exatamente 1 objeto Zero ou mais objetos
1..*	Um ou mais objetos
0..1	Zero ou um objetos
5..8	5, 6, 7 ou 8 objetos
4..7, 9	4, 5, 6, 7 ou 9 objetos



# Cardinalidade





- Para cada classe em uma associação:
  - A associação é opcional?
    - Limite mínimo de zero.
  - A associação é obrigatória?
    - Limite mínimo de um.
  - Um único objeto deve ser associado?
    - Limite máximo de zero.
  - Diversos objetos podem ser associados?
    - Limite máximo infinito ( \* ).



# Agregação

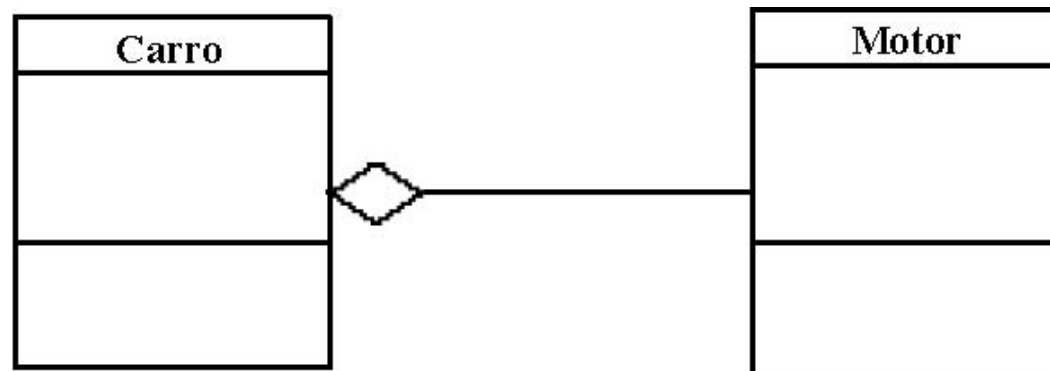
Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Uma agregação é um tipo especial de associação, utilizado quando existe uma relação de *contedor-conteúdo* entre as classes.
  - Exemplos:
    - Um carro contém um motor
    - Uma cidade contém bairros



# Representação

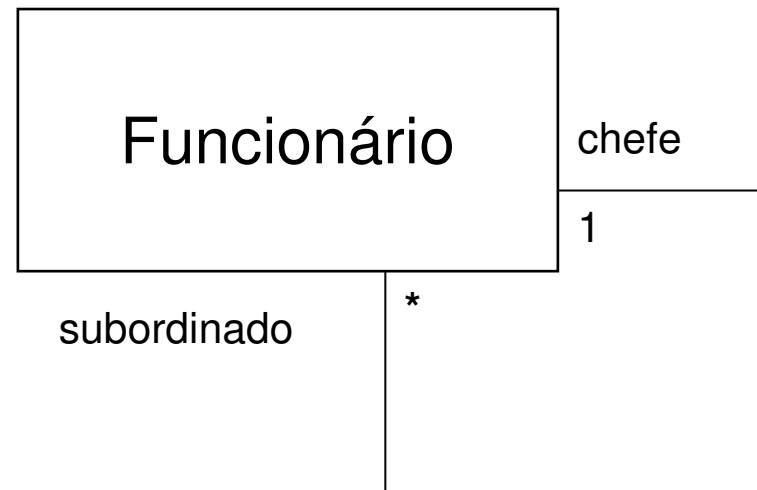
Uma agregação é representada como uma associação com um “diamante” indicando a classe que contém.





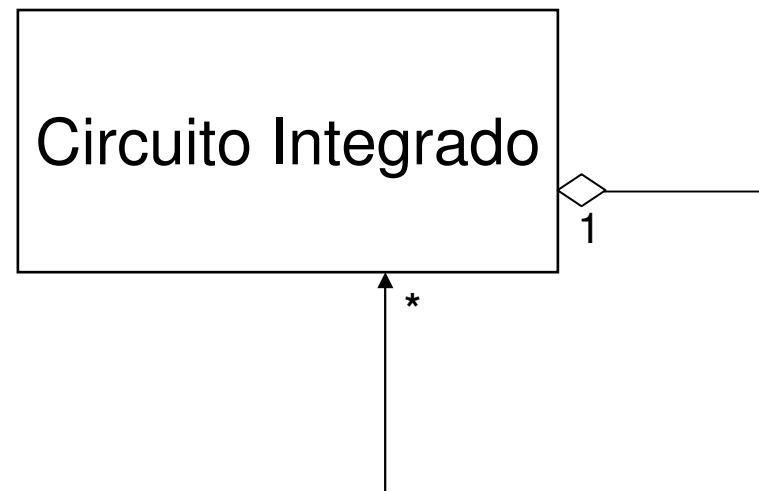
# Autorelacionamento

- Um auto-relacionamento ocorre quando uma classe possui um objeto da mesma classe como atributo, ou seja, quando uma classe possui uma associação ou agregação consigo mesma.





- Exemplo:
  - Um circuito eletrônico é composto de diversos circuitos eletrônicos mais simples.





→ Agregação (por referência)



→ Composição (por valor)





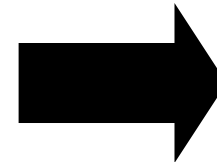
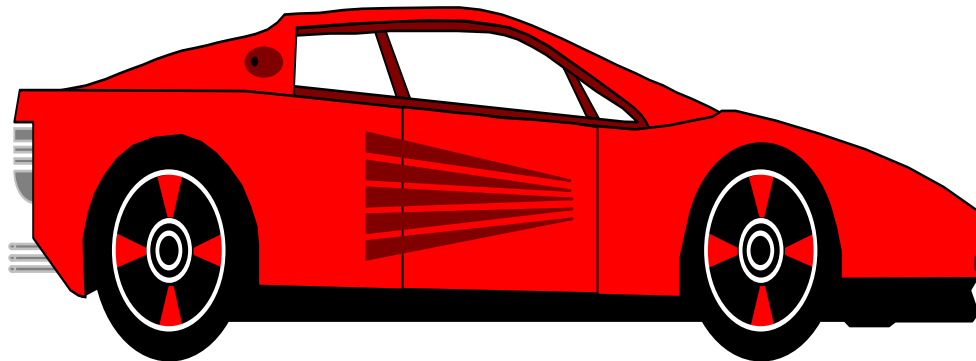
# Métodos

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Métodos indicam comportamento e destacam:
  - ✓ Operações que podem ser realizadas sobre o objeto.
  - ✓ Operações que o objeto realiza no contexto do sistema.
- Visibilidade:
  - ✓ Os métodos utilizam os mesmos modificadores de visibilidade utilizados nos atributos.
- Classificação de métodos:
  - ✓ Algoritmicamente simples.
  - ✓ Algoritmicamente complexos.



## *Exemplo: Classe Carro*



*Trocar Marcha*

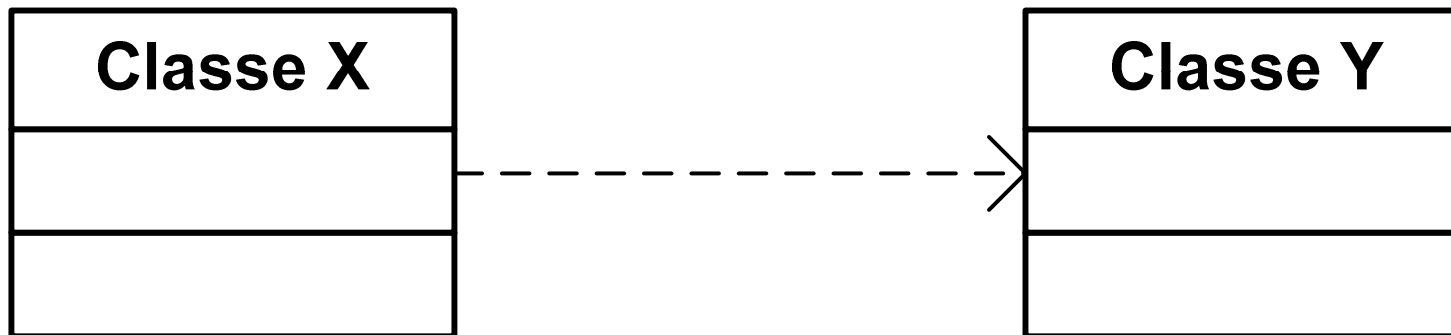
*Acelerar*

*Abastecer*



# Dependência entre Classes

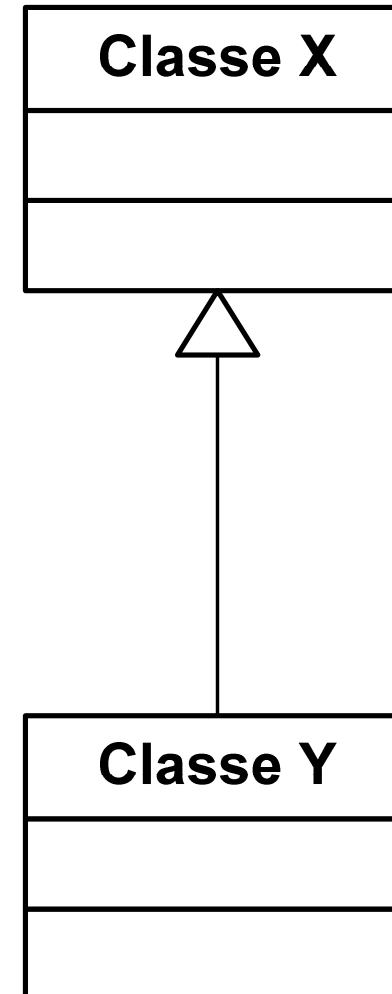
- Uma classe X depende de uma classe Y quando:
  - ✓ Possui um atributo da classe Y.
  - ✓ Executa algum método da classe Y.
  - ✓ Algum de seus métodos possui uma variável local, parâmetro ou tipo de retorno da classe Y.
  - ✓ A dependência é representada por uma seta pontilhada.





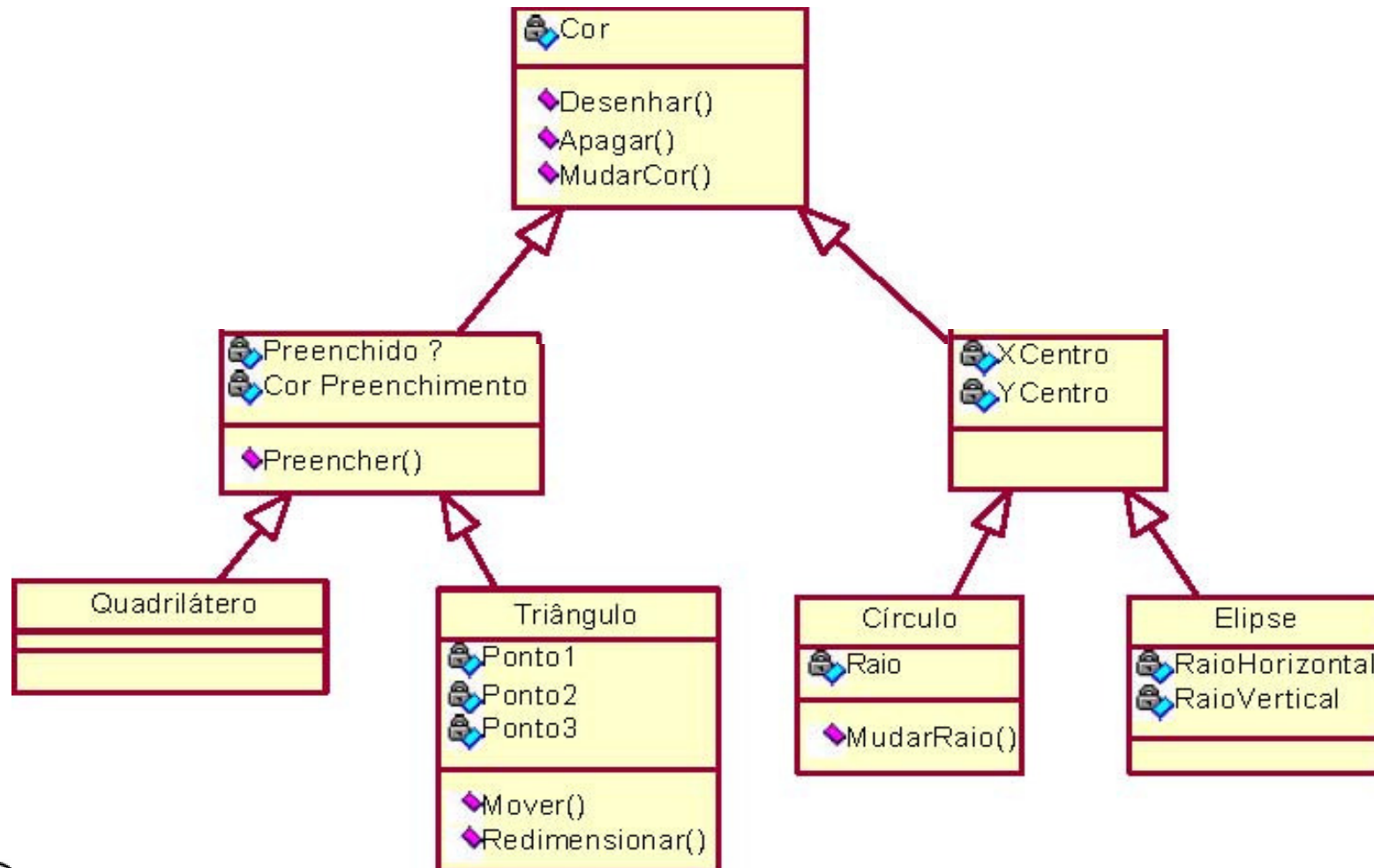
# Herança

- Em UML, a herança é representada como uma “seta vazada”:
  - ✓ A seta aponta a superclasse.
  - ✓ A herança **não** pode ter nome, papel de classe, navegação ou cardinalidade.





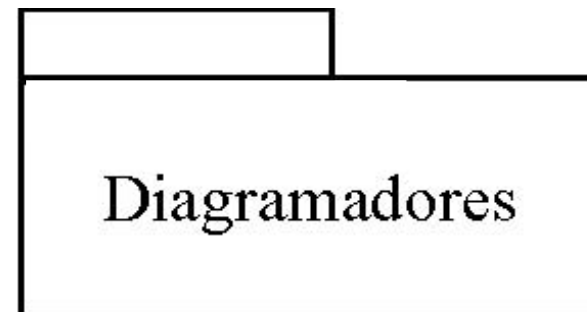
# Exemplo





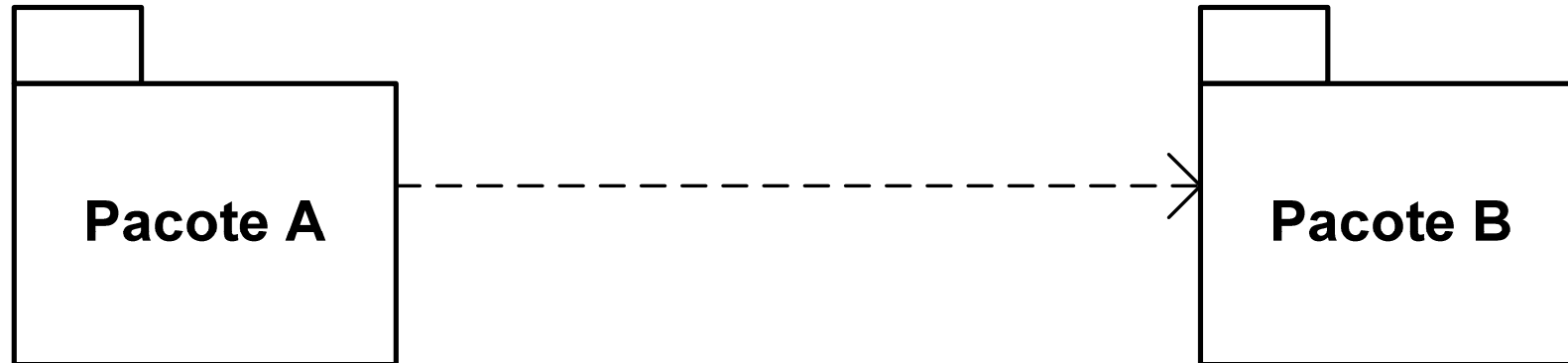
# Pacotes

- A maioria dos modelos contém diversas classes e como tal podem ser agrupadas em pacotes.
- ✓ Um pacote é uma coleção lógica de classes ou pacotes.
- ✓ O pacote é representado por uma pasta.





- Dependência entre pacotes:
  - ✓ Um pacote A depende de um pacote B quando ao menos uma classe do pacote A depende de uma classe do pacote B.





# Modelagem Dinâmica

## (Visão de Processo)



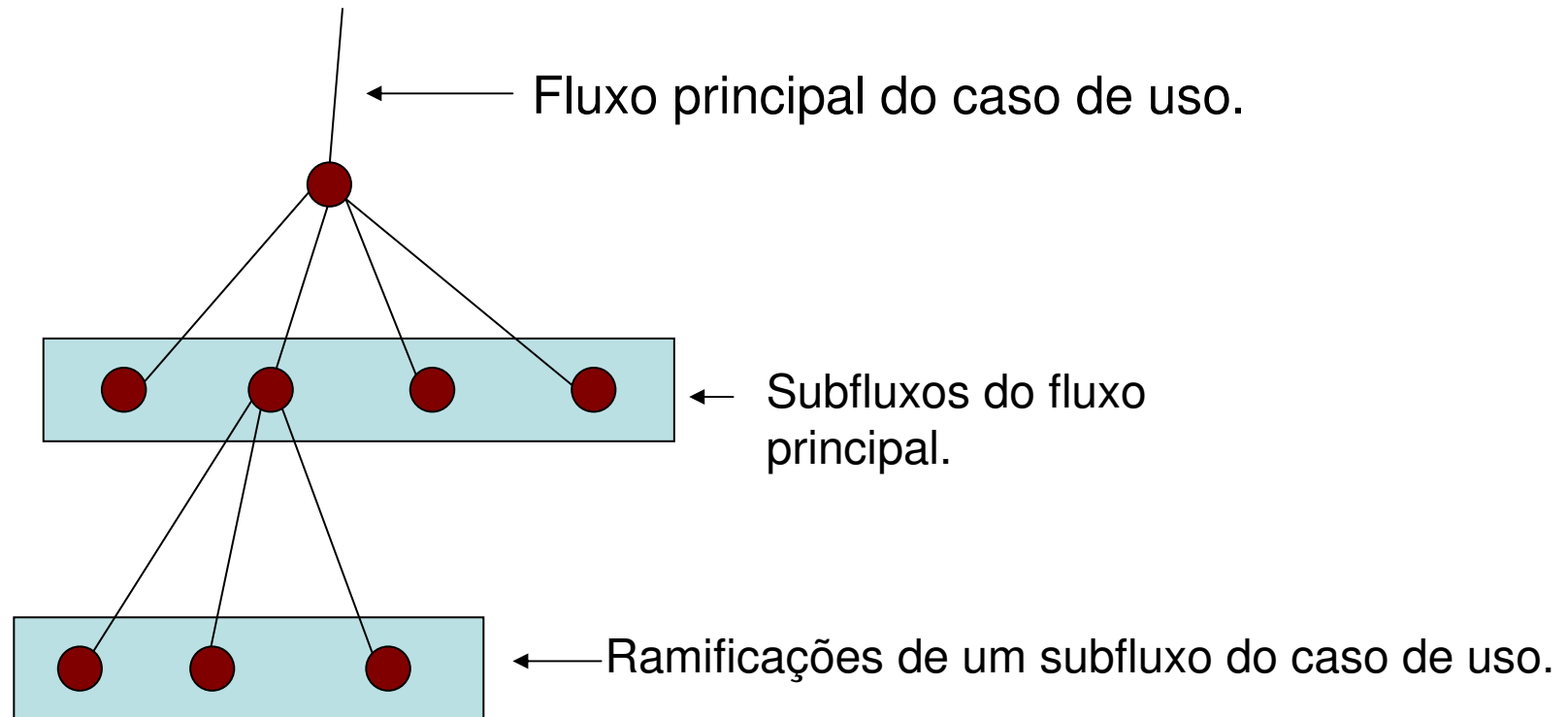


# Diagrama de Sequência

## Protocolos de Interação



# Fluxos e Subfluxos



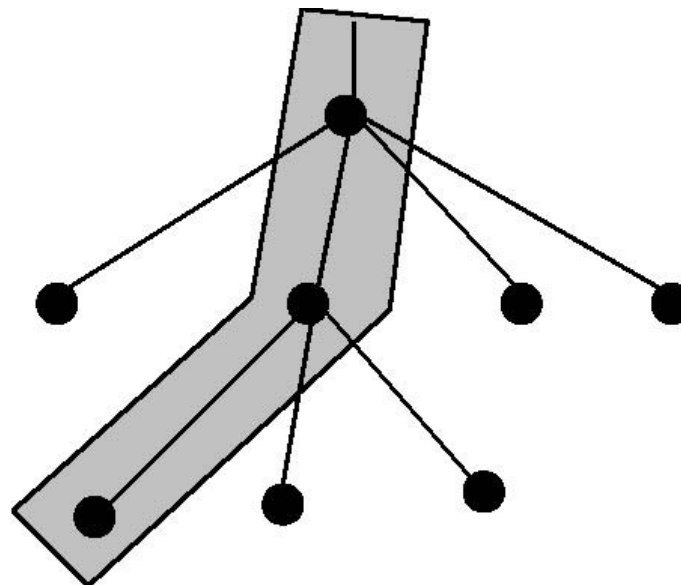
Subfluxos demonstram operações que podem ser selecionadas pelo usuário ou casos especiais de algumas operações.



# Cenário

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Um cenário é um caminho da raiz até uma folha da árvore de ramificações de um caso de uso.





# Diagramas de Sequência

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

Determinam os objetos responsáveis pela realização de um cenário e as mensagens que são trocadas entre eles.

O diagrama apresenta a ordem com que as mensagens são trocadas no tempo.

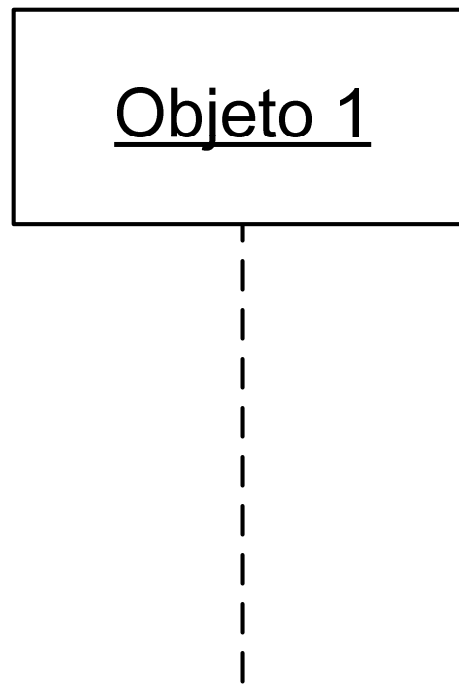




# Objetos

Um diagrama de sequências contém diversos objetos:

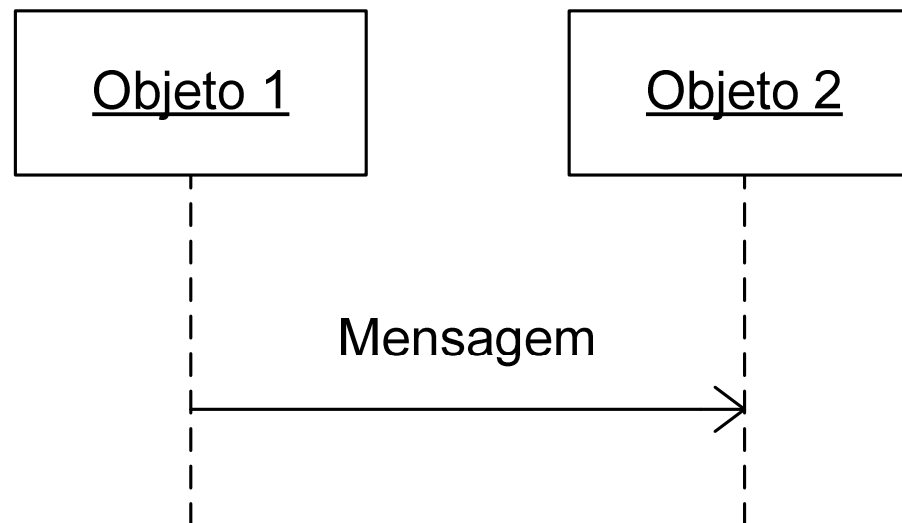
- ✓ Os objetos são alinhados no topo do diagrama.
- ✓ Suas linhas de vida descem em direção à base do diagrama.





## Representação:

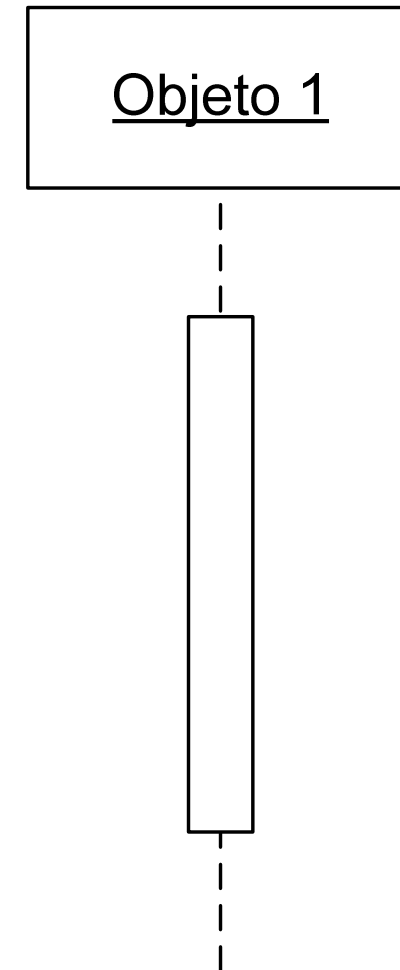
- ✓ Mensagens são representadas como setas entre dois objetos.
- ✓ As setas devem conter nomes de métodos do objeto destino.





Retângulos na linha de vida de um objeto:

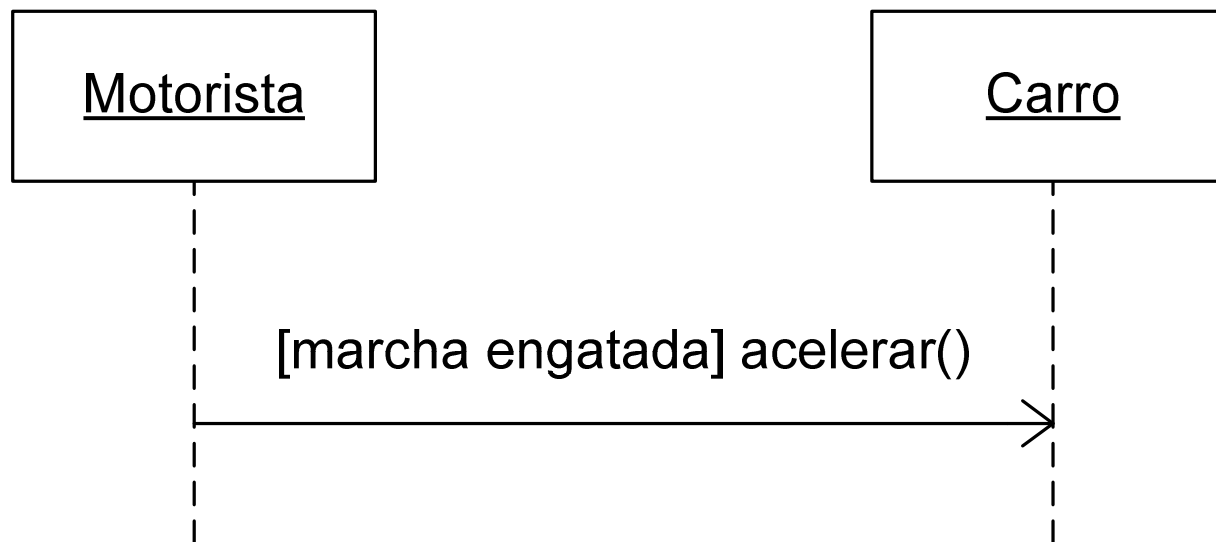
- ✓ Representam o tempo em que um objeto está ativo na troca de mensagens.
- ✓ Focos de controle são opcionais nos diagramas de sequência.





## Condição de mensagens:

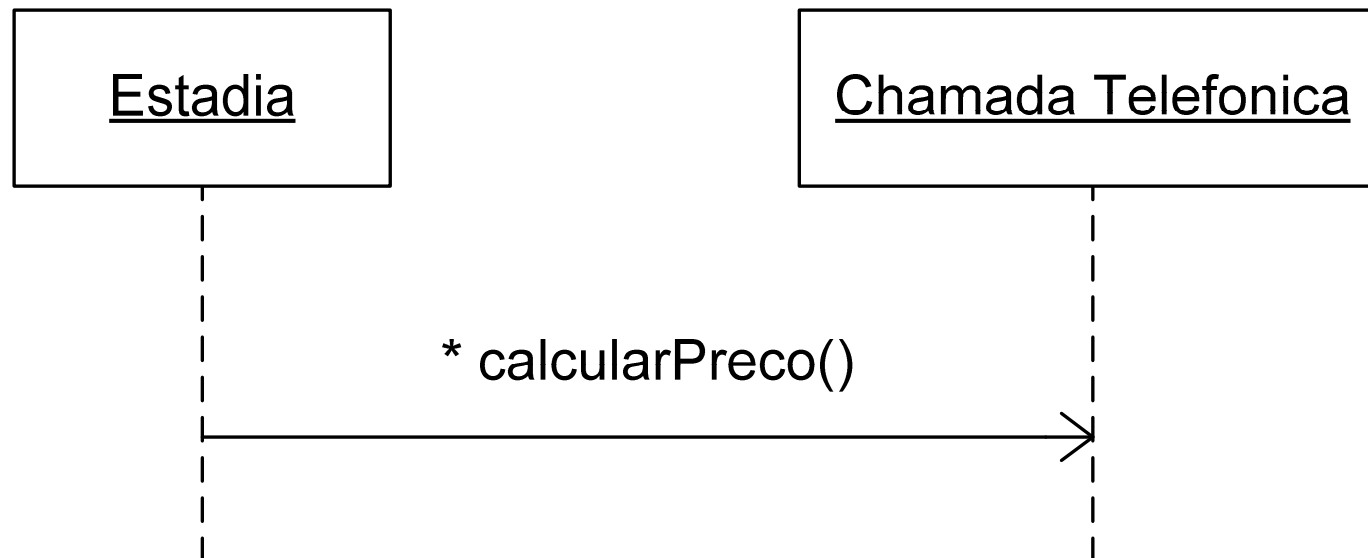
- ✓ A mensagem somente será emitida se a condição for verdadeira.
- ✓ A condição é expressa entre colchetes antes do nome do método.





## Repetição de Mensagens:

- ✓ A mensagem será emitida múltiplas vezes.
- ✓ Um asterisco é apresentado antes do nome do método.



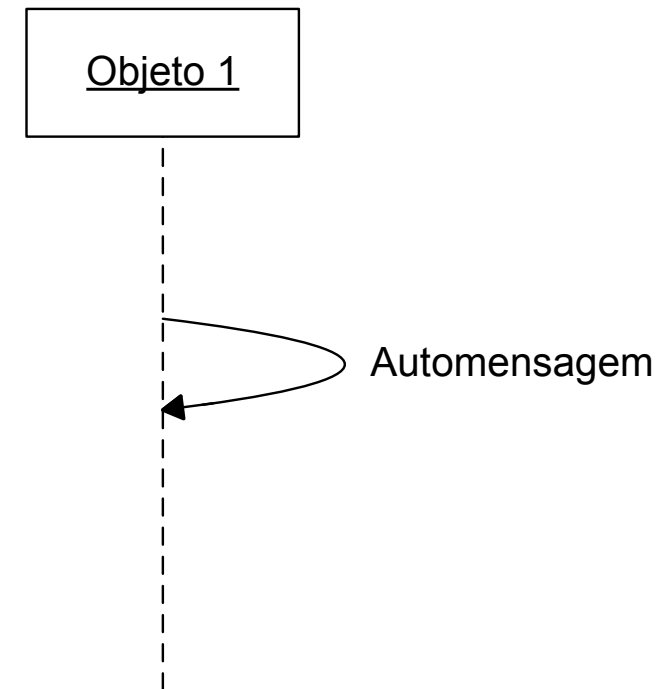


# Automensagem

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

Ocorre quando um objeto chama um método seu para realizar parte do cenário:

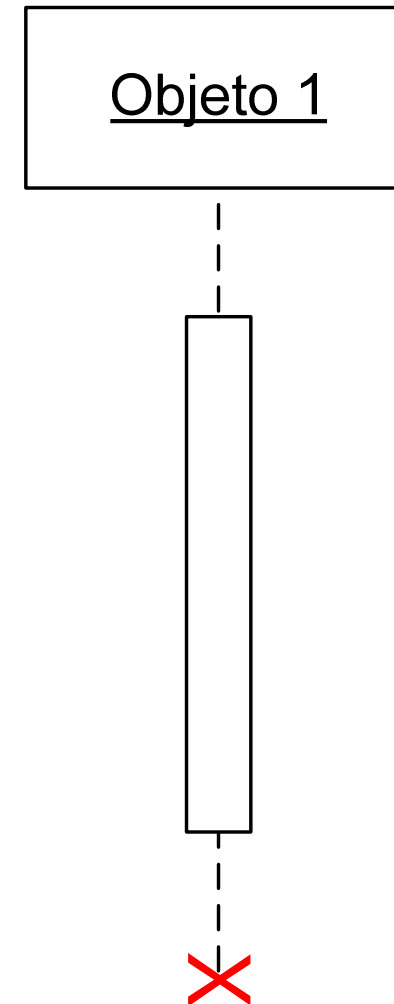
✓ Automensagens são representadas por setas saindo e retornando para o próprio objeto.





# Encerramento

Um X representa o fim da execução  
e da vida do objeto:





# Diagramas de Colaboração

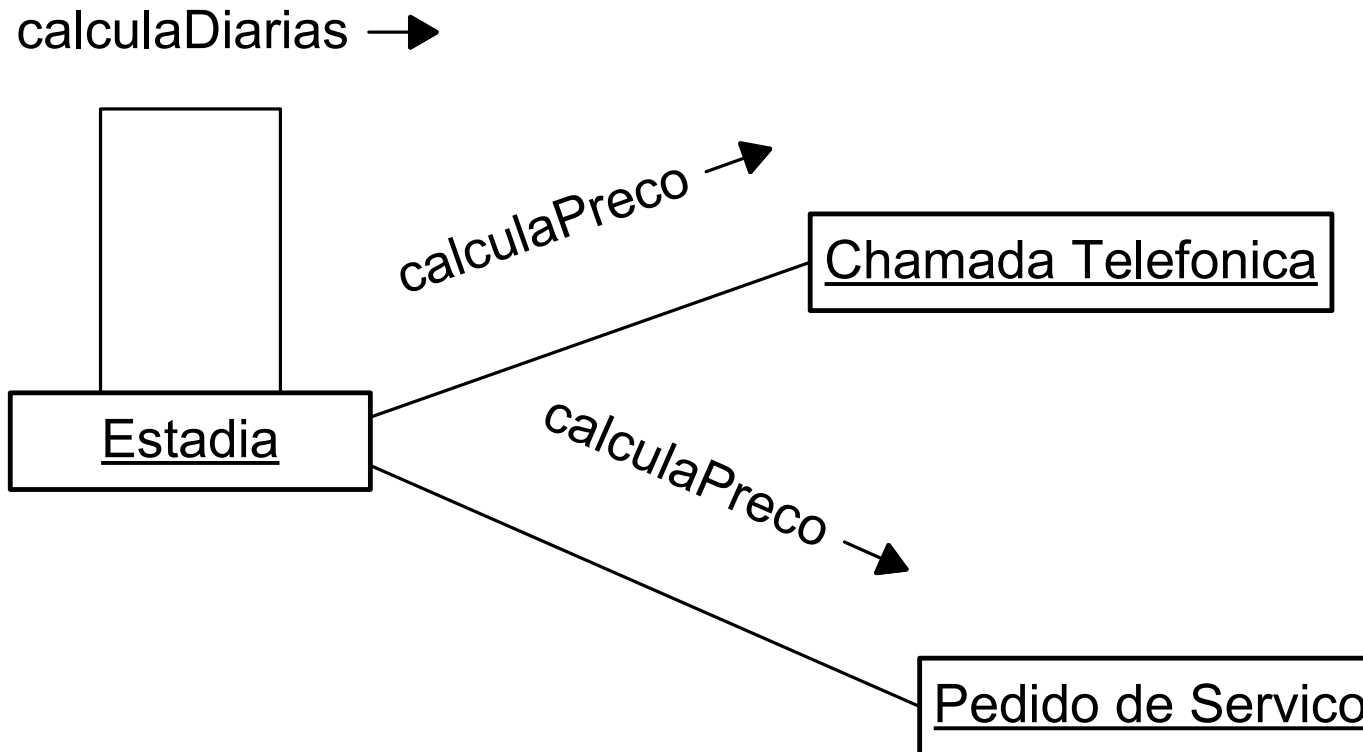
Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Alternativa ao diagrama de sequência.
- Não apresenta a linha de tempo dos objetos.
  - Objetos representados por retângulos.
  - Mensagens representadas como setas entre os retângulos.
  - Mensagens são numeradas (sequência de tempo).





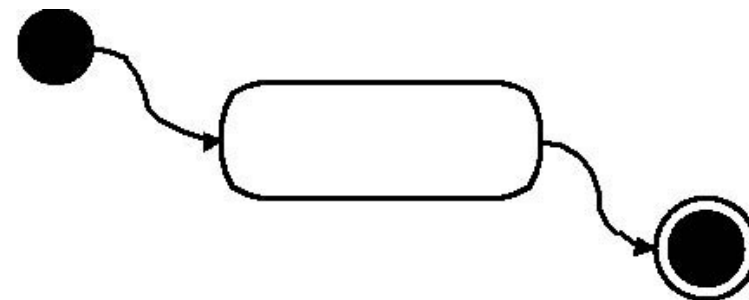
# Exemplo





# ESTADOS

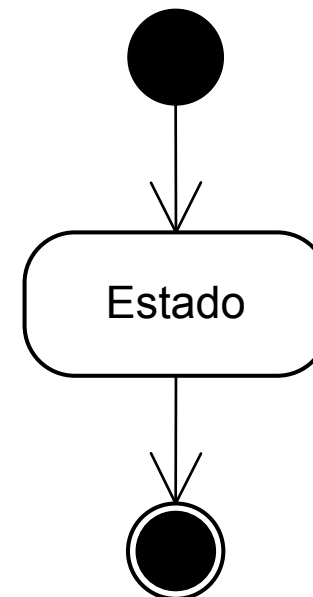
## Modelagem Dinâmica de Classes





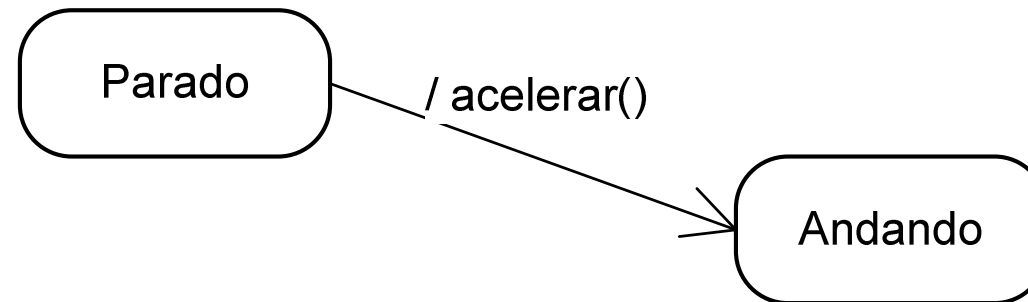
# Diagramas de Estado

- Estes diagramas são utilizados por diversos métodos de análise:
  - ✓ Análise estruturada moderna.
  - ✓ Análise essencial.
  - ✓ Outros métodos de análise orientada a objetos.
- Componentes:
  - ✓ Estado
  - ✓ Transição
  - ✓ Estado Inicial
  - ✓ Estado Final





- Representação UML
  - ✓ **Estado:** representado por um retângulo de bordas arredondadas.
  - ✓ **Transição:** representada por uma seta entre os estados.





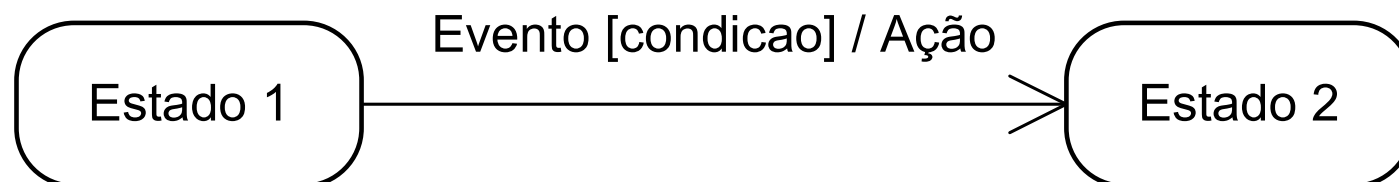
- Um diagrama de estados tem no mínimo dois estados:
  - ✓ **Estado Inicial:** um objeto recém criado no sistema se encontra neste estado.
  - ✓ **Estado Final:** estado final na cadeia de troca de estados do objeto. O objeto não poderá trocar de estado após atingir seu estado final.

*Em um diagrama de estados podem existir diversos estados finais.*



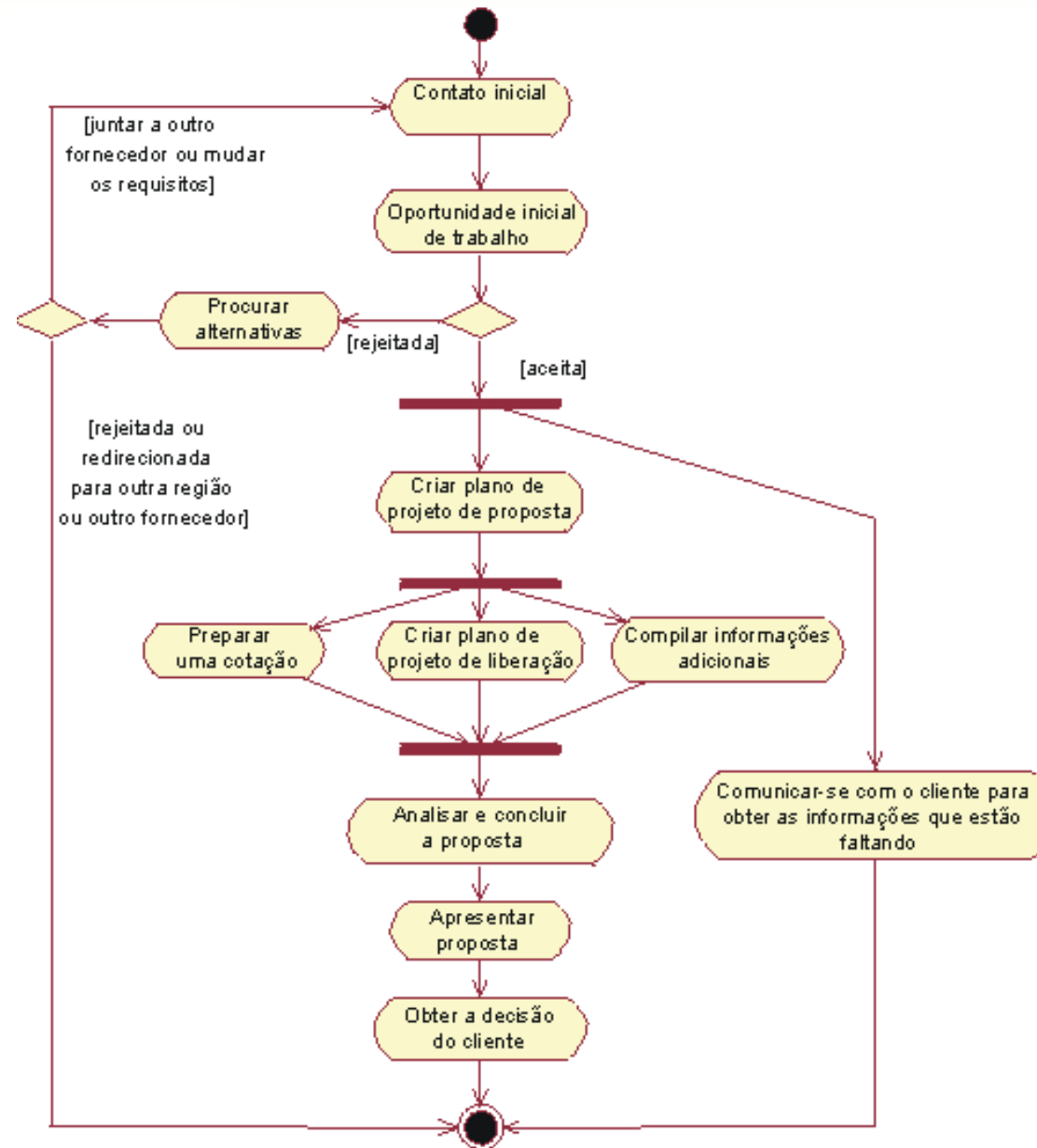
# Transições

- Uma transição pode estar associada a:
  - ✓ Uma ação, que indica um método do objeto que será executado quando a transição de estado se realizar.
  - ✓ Uma condição, também conhecida como guarda, que indica quando a transição de estado deve ocorrer.
  - ✓ Ambos são apresentados junto ao nome do evento na transição.
  - ✓ Todos são opcionais.





# Diagrama de Estados





# Modelagem Física

## (Visão Física)



# Diagrama de Componentes



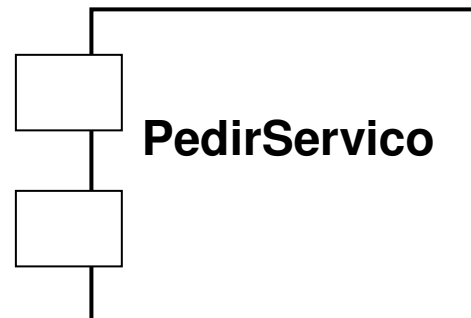
# Diagrama de Componentes

Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

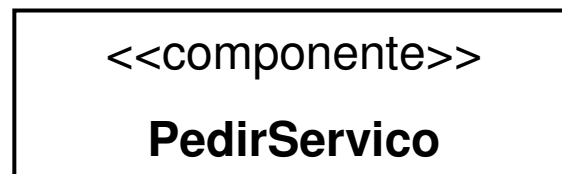
- Apresenta os módulos físicos do software e suas relações;
- O software se torna um conjunto de unidades modulares e reutilizáveis;
- Podem representar uma classe, aplicações, subsistemas ou sistemas.



- Notação
  - UML 1.4



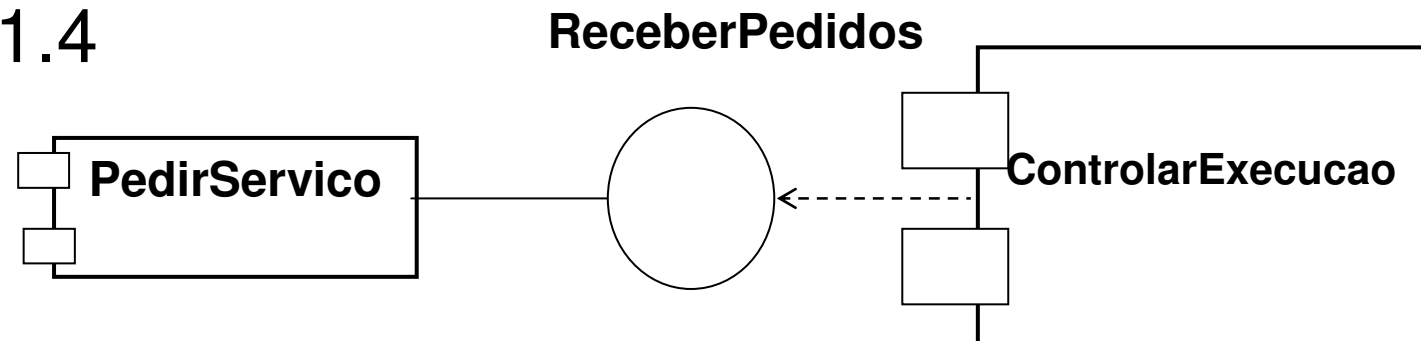
- UML 2.0



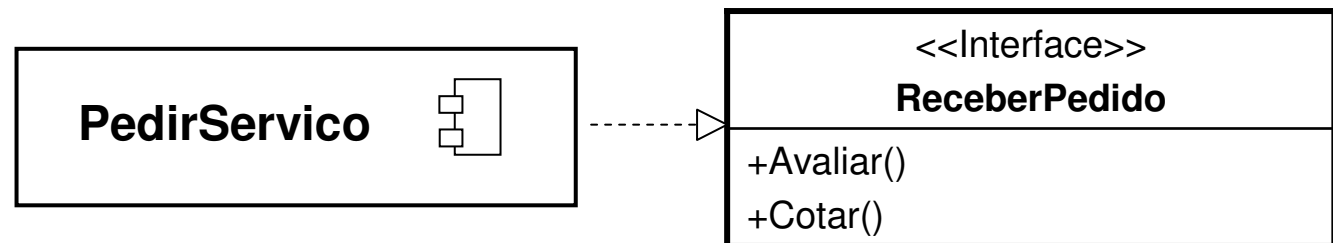
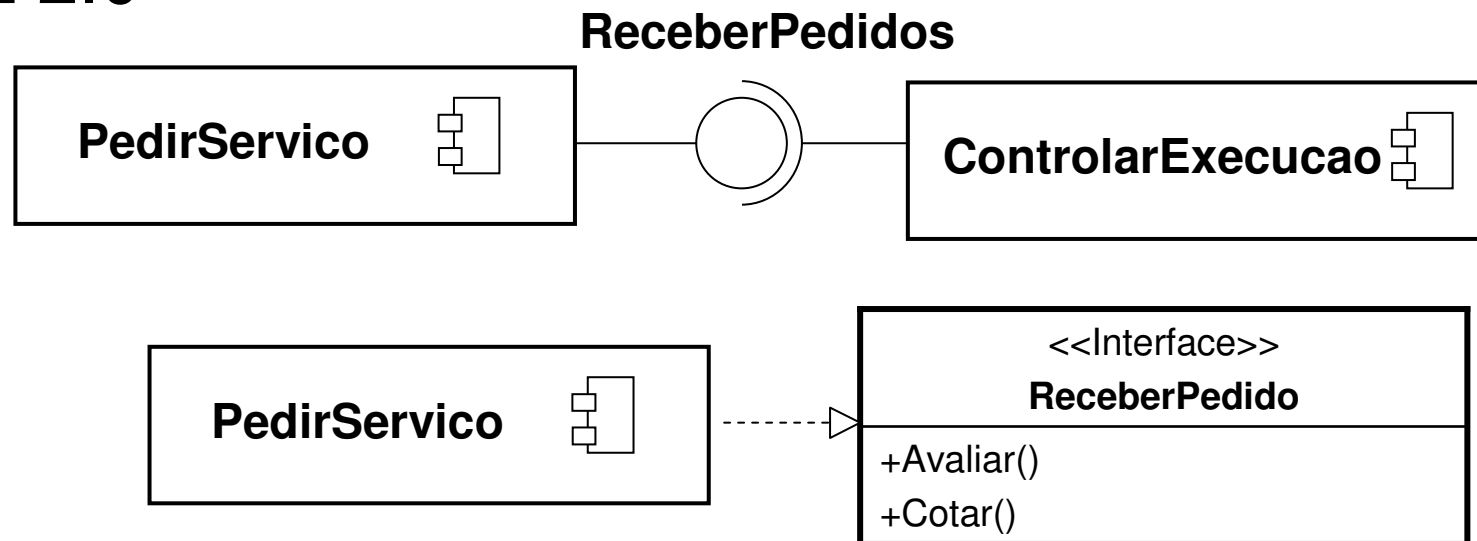


- Interfaces

- UML 1.4

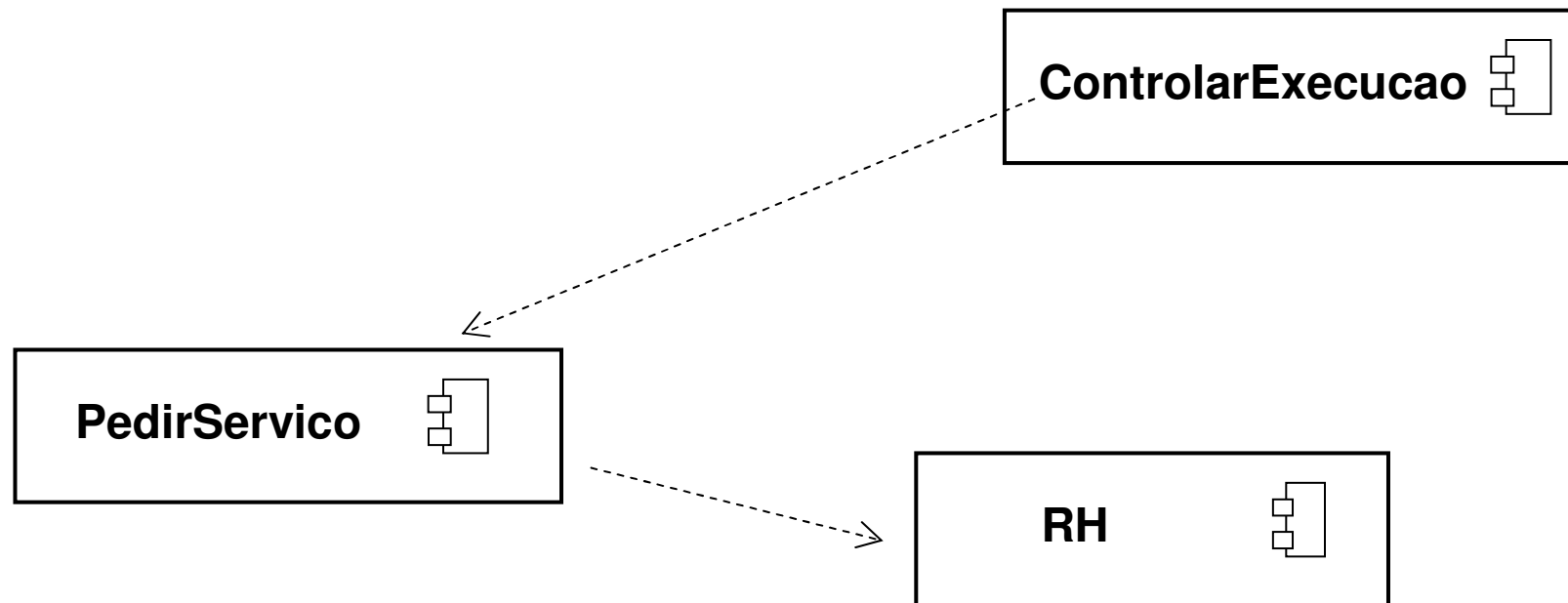


- UML 2.0



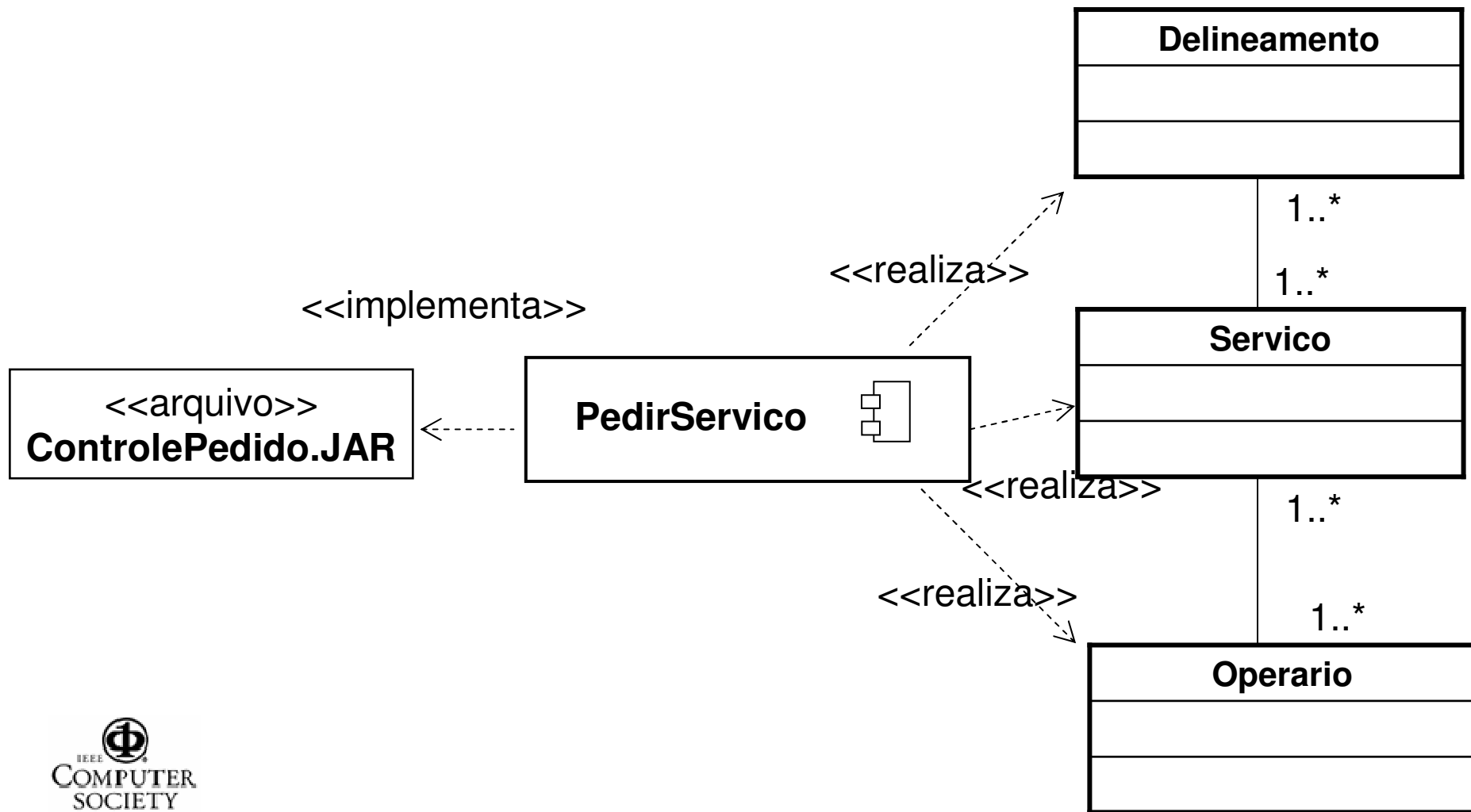


- Dependências



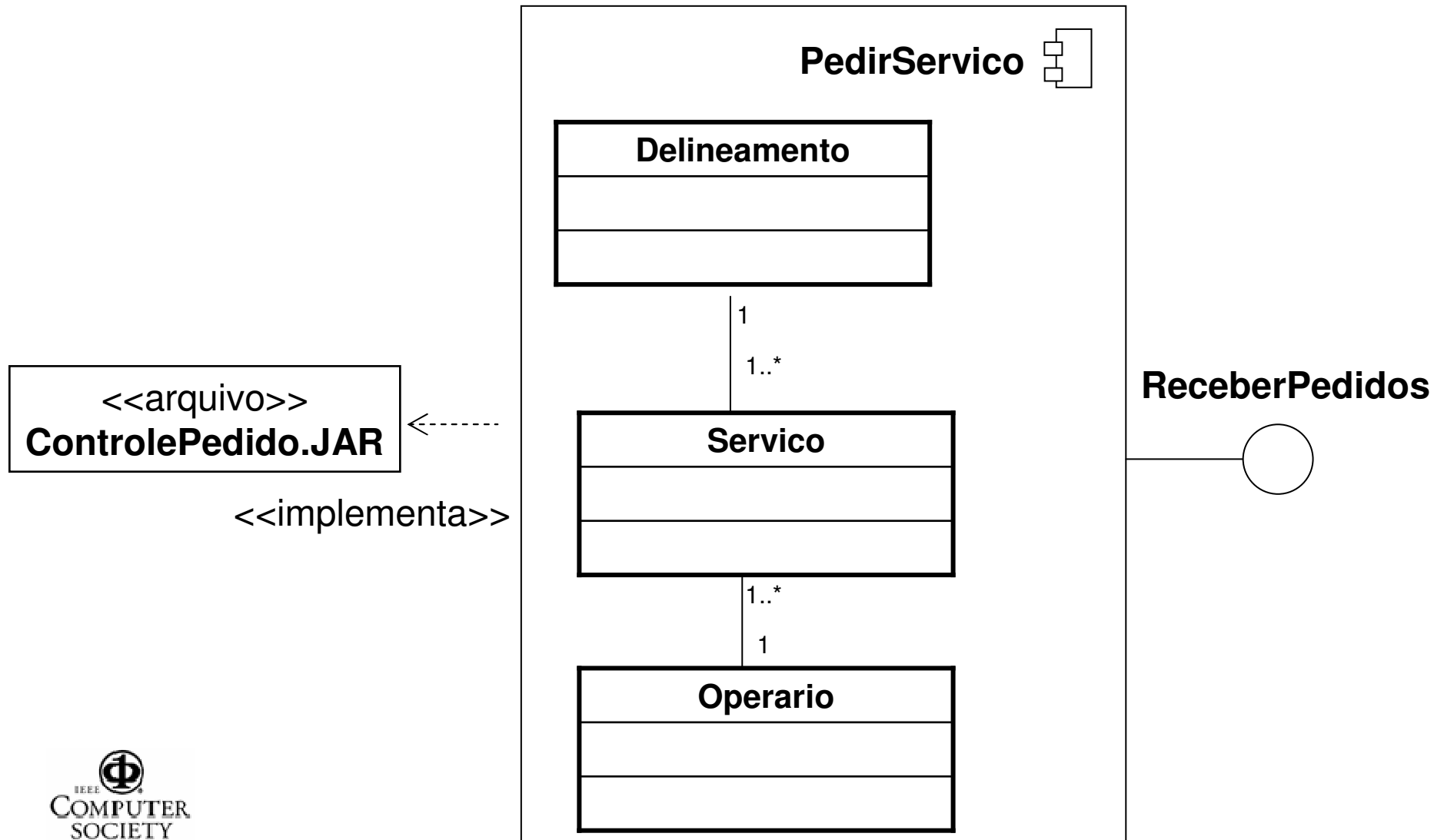


# Diagrama de Componentes





# Diagrama de Componentes





# Diagrama de Distribuição



# Diagrama de Componentes

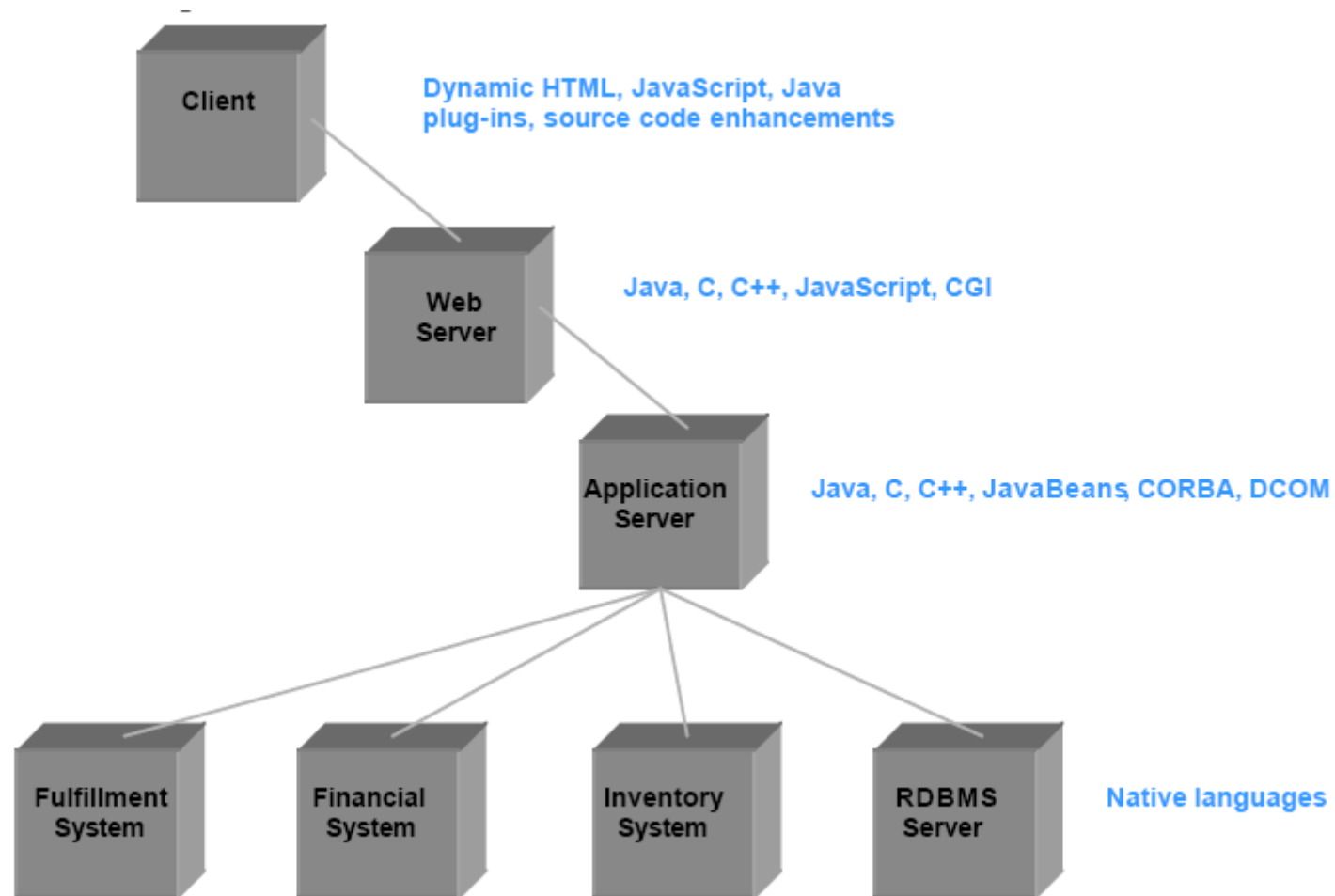
Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

- Captura a topologia (ambiente) de hardware de um sistema sobre a qual são executados os componentes de software:
  - **Construído como parte da especificação da arquitetura física.**
- **Objetivo:**
  - **Especificar a distribuição de componentes.**
  - **Identificar problemas de desempenho.**



# Diagrama de Componentes

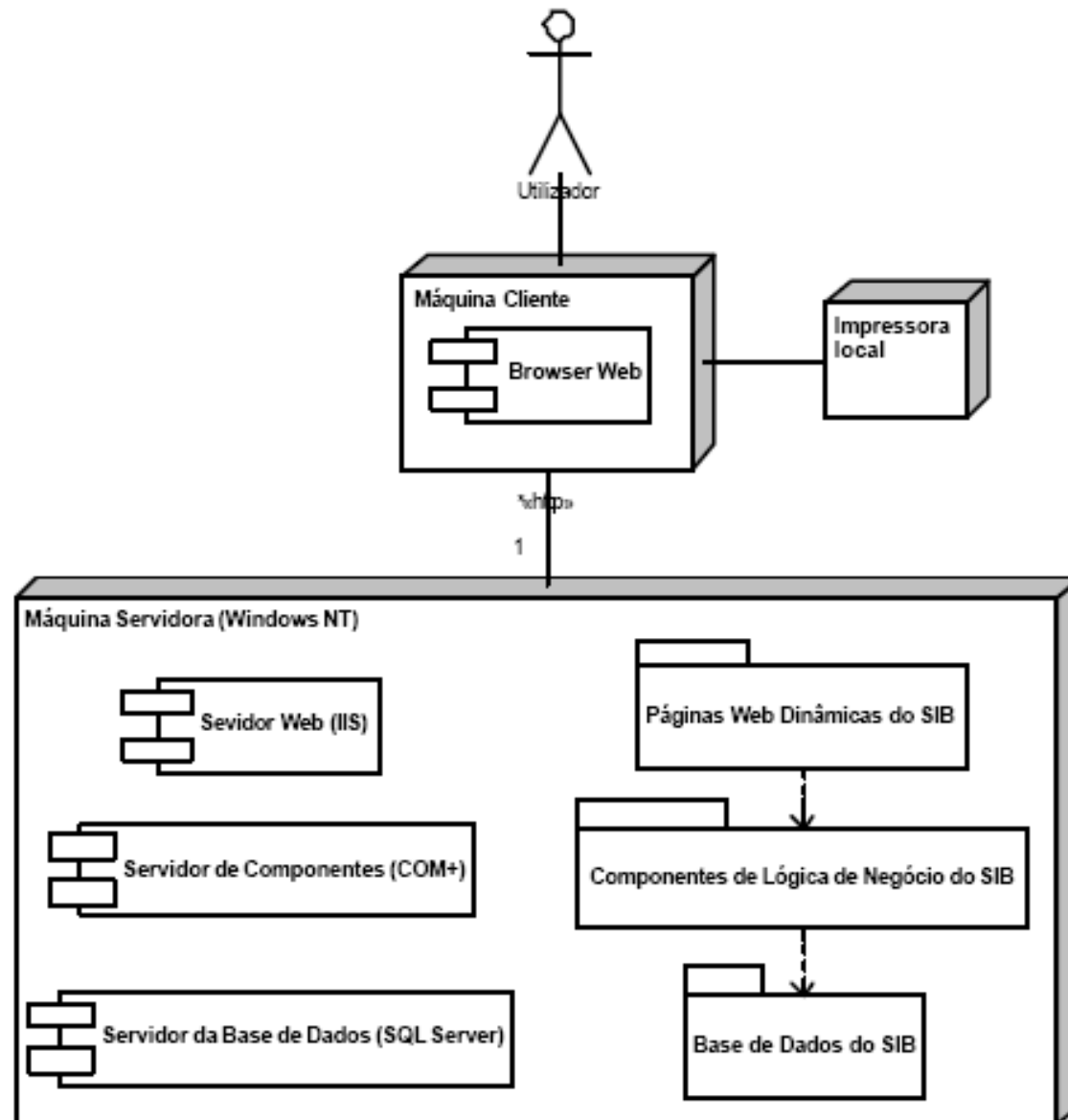
Universidade Federal  
do Rio de Janeiro  
Escola Politécnica



[fabio@mz-empresarial.com.br](mailto:fabio@mz-empresarial.com.br)



# Diagrama de Componentes





# Diagrama de Componentes

