

# Engenharia de Software I

Requisitos de Software:

- Engenharia de Requisitos

→ **Especificação de Requisitos**

→ **Diagrama de casos de uso**



**Elicitação/  
Especificação  
de requisitos**

**Manutenção  
Evolução**



**Projeto de software**



**Plano de testes**



**Codificação**



**Testes de software**

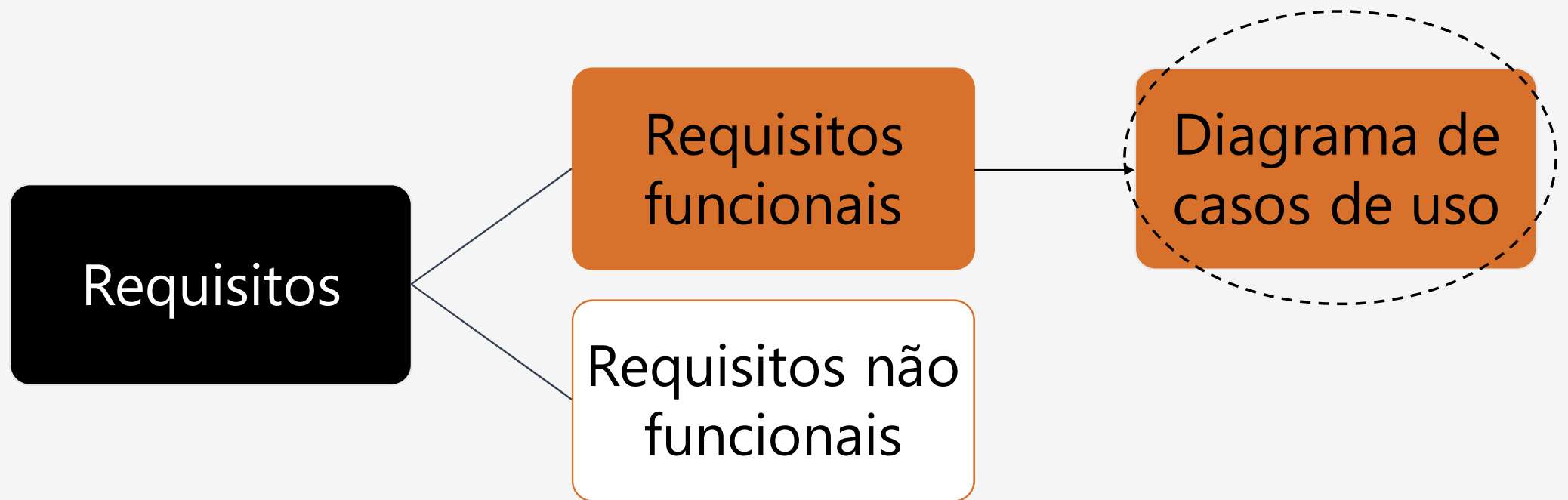


**Implantação**

## Exemplo: Sistema Loja Virtual – Módulo Administração da Loja

ID	REQUISITO FUNCIONAL	TIPO USUÁRIO DO SISTEMA	DESCRIÇÃO DO REQUISITO FUNCIONAL
RF01	Cadastrar categorias e subcategorias de produtos	Setor de vendas	O sistema deverá permitir cadastrar categorias de produtos. Cada categoria poderá ter subcategorias e sub-subcategorias de produtos que serão utilizadas para possibilitar filtros de busca de produtos pelo cliente final.
RF02	Cadastrar produtos	Setor de vendas	O sistema deverá permitir cadastrar produtos. Todo produto possui: nome, descrição, valor, categoria e subcategoria(s) de produtos. Os produtos poderão ter derivações: de cores e tamanhos diferentes. O estoque do produto poderá ser controlado por unidade ou por derivações: cores e tamanhos. Cada produto poderá ter várias de imagens (galeria), vídeos ou arquivos pdf associados. Impedir estoque negativo.
RF03	Cadastrar usuários	Administrador	O sistema deverá permitir cadastrar usuários que terão acesso ao módulo administração da loja. O cadastro de usuário contém informações de dados pessoais e do funcionário (conforme ficha de contratação). Cada usuário terá um código único e uma senha para acesso ao sistema. Deverá haver níveis de acesso diferentes ao sistema: administrador, comercial e financeiro. Ao sair da empresa um usuário deverá ser desativado impossibilitando o seu acesso ao sistema.
RF04	Setor de vendas	Administrador, Setor de vendas	Para acessar o sistema , o usuário deverá informar seu código e senha. Cada pessoa poderá acessar apenas as funcionalidades de acordo com o seu perfil de usuário. Deverão ser registrados logs de todas as ações dos usuários no sistema.

\* RF – Requisito funcional



# Diagrama de casos de uso

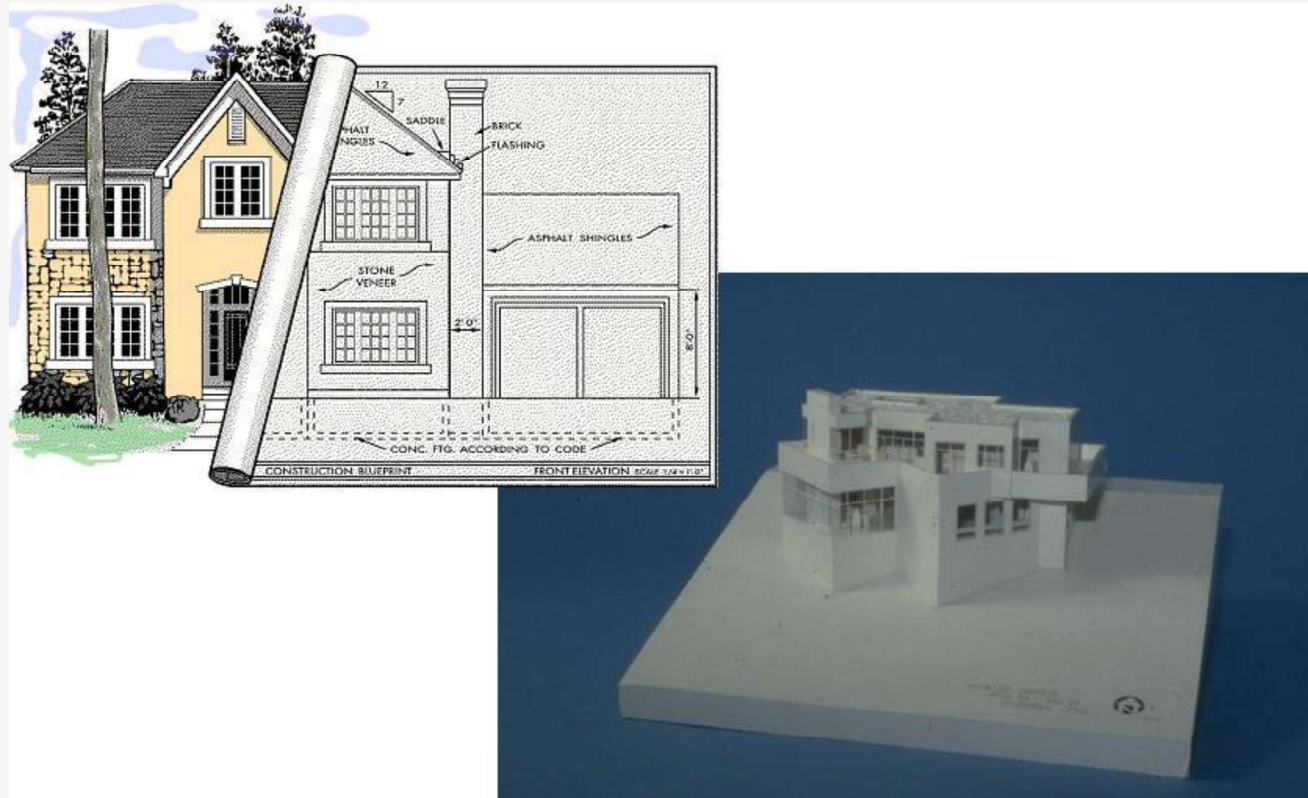
# UML

---



- UML – Linguagem de Modelagem Unificada
- A UML é uma linguagem visual utilizada para **modelar softwares** baseados no paradigma de orientação a objetos.
- Tornou-se, nos últimos anos, a linguagem **padrão de modelagem adotada internacionalmente** pela indústria de engenharia de software.

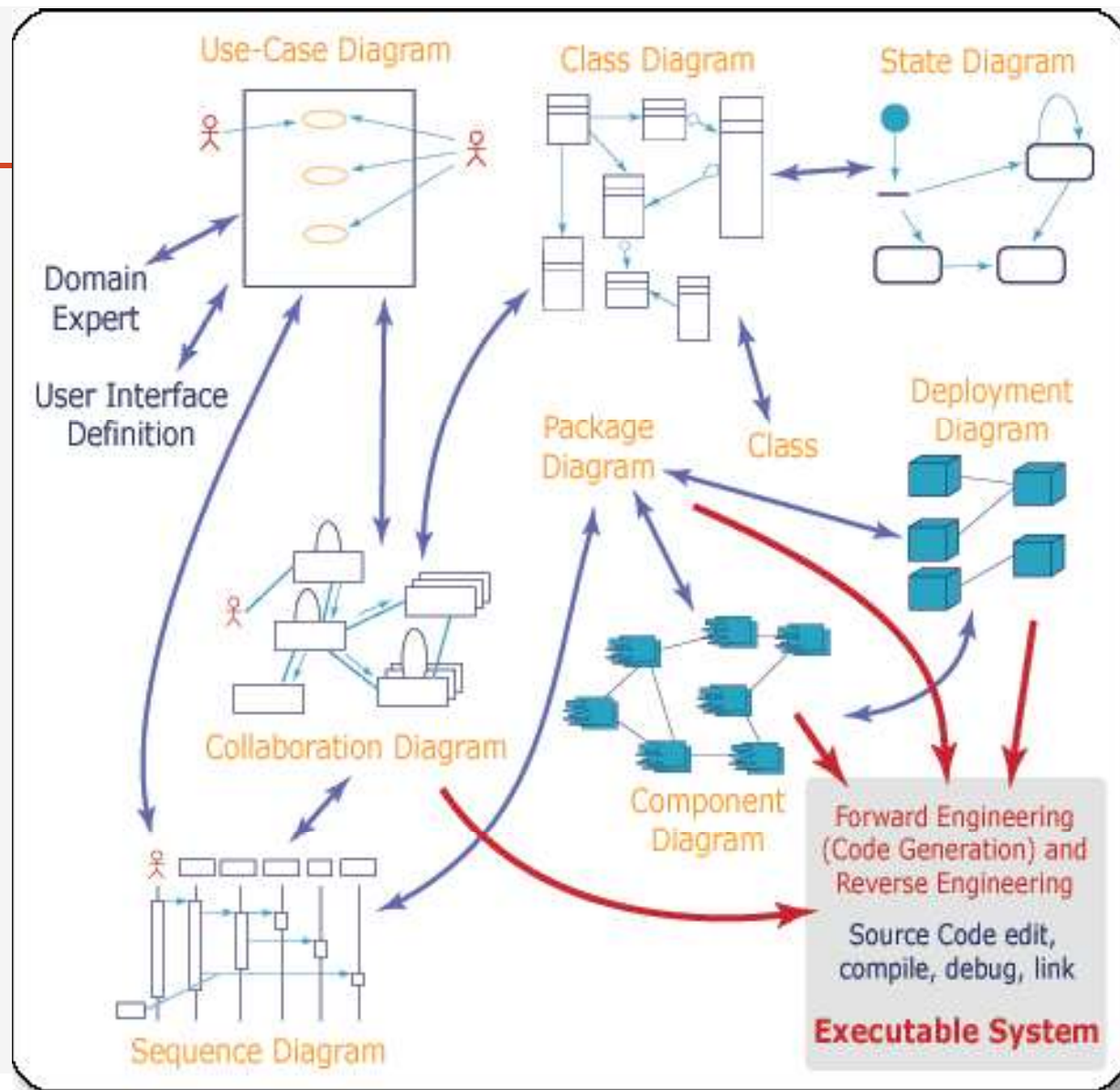
# Representação do projeto de software de forma visual e padronizada, como ocorre na construção civil



# Diagramas da UML

## Modelagem visual:

necessidade de modelar o sistema seguindo uma notação padrão





# UML

---

- A UML foi um passo importante em direção à padronização de desenvolvimento de software
- Surgiu em 1995 da união de três métodos de modelagem OO: Booch, OMT e OOSE
- Tornou-se padrão para modelagem e projetos de sistemas



**Grady Booch**  
Método Booch



**Ivar Jacobson**  
Método OMT (Rational)



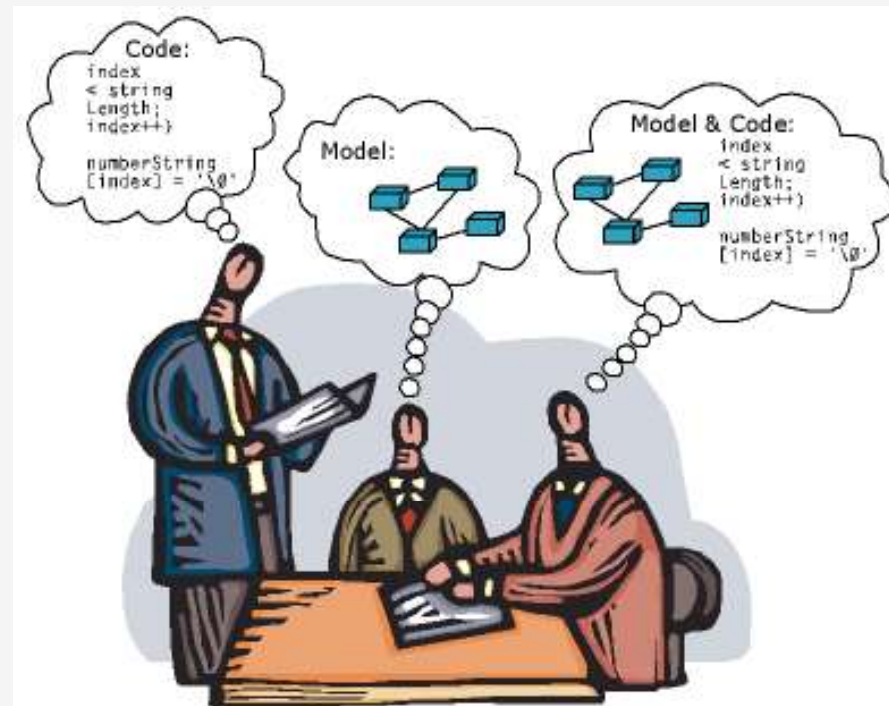
**James Rumbaugh**  
Método OOSE

# UML

---

## Linguagem consistente para:

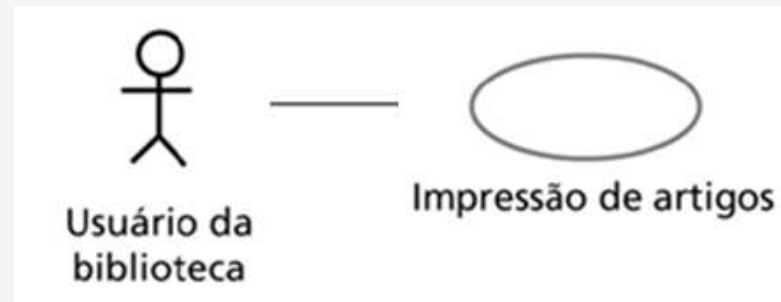
- Visualização
- Especificação
- Documentação
- Construção



# Diagrama de casos de uso

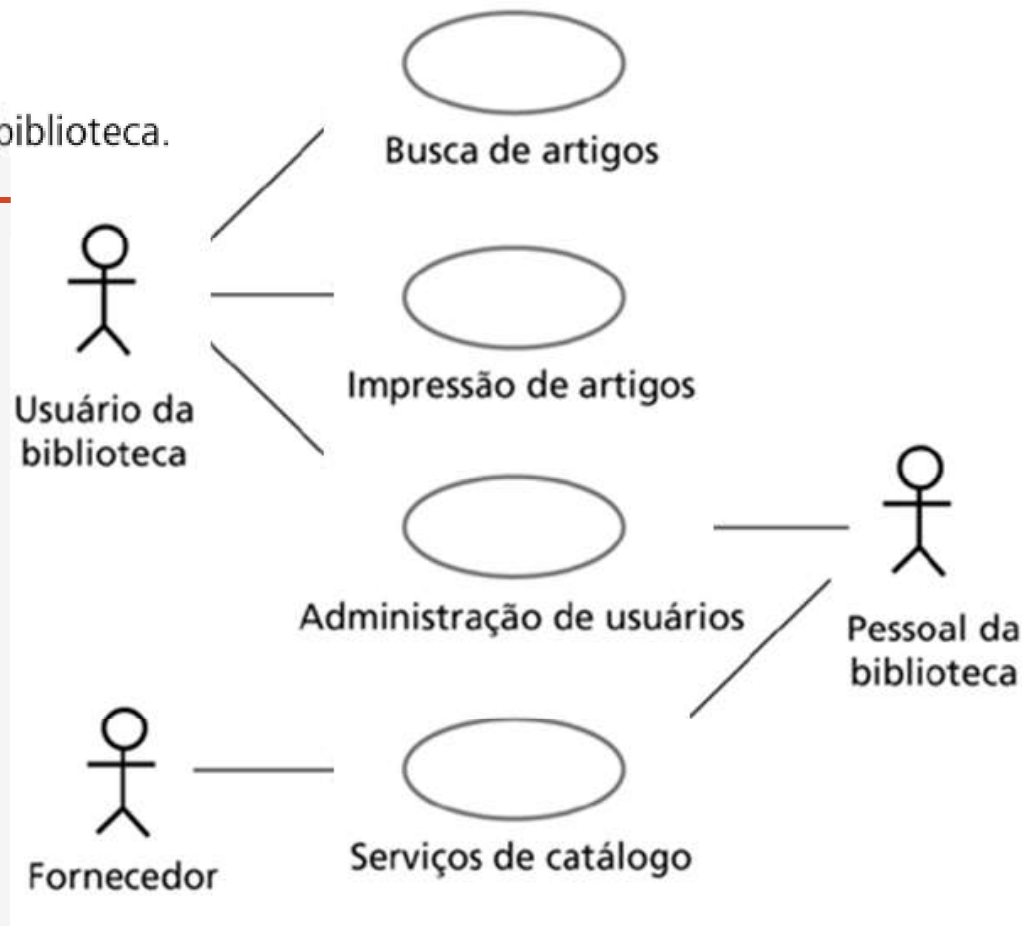
---

- Técnica **especificação de requisitos** baseada em cenários
- identificam os agentes em uma interação (atores) e descrevem a interação com o sistema (casos de uso).



**Figura 7.6**

Casos de uso para o sistema de biblioteca.



Mostra os potenciais usuários do sistema (atores) e as funcionalidades que serão úteis para eles (casos de uso)

# Casos de uso

---

- Um conjunto de casos de uso deve descrever todas as possíveis interações do ator com o sistema.
- Os diagramas de caso de uso são usados para especificar os requisitos funcionais do sistema.
- Permite que os requisitos fiquem acessíveis e compreensíveis a todos, por apresentarem uma visão externa sobre como os elementos que interagem com sistema.
- Modelagem dinâmica de sistema em alto nível de abstração

# Principais elementos gráficos

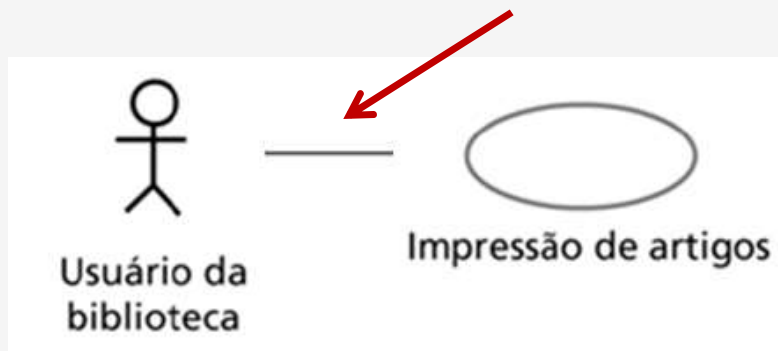
---



➔ Atores



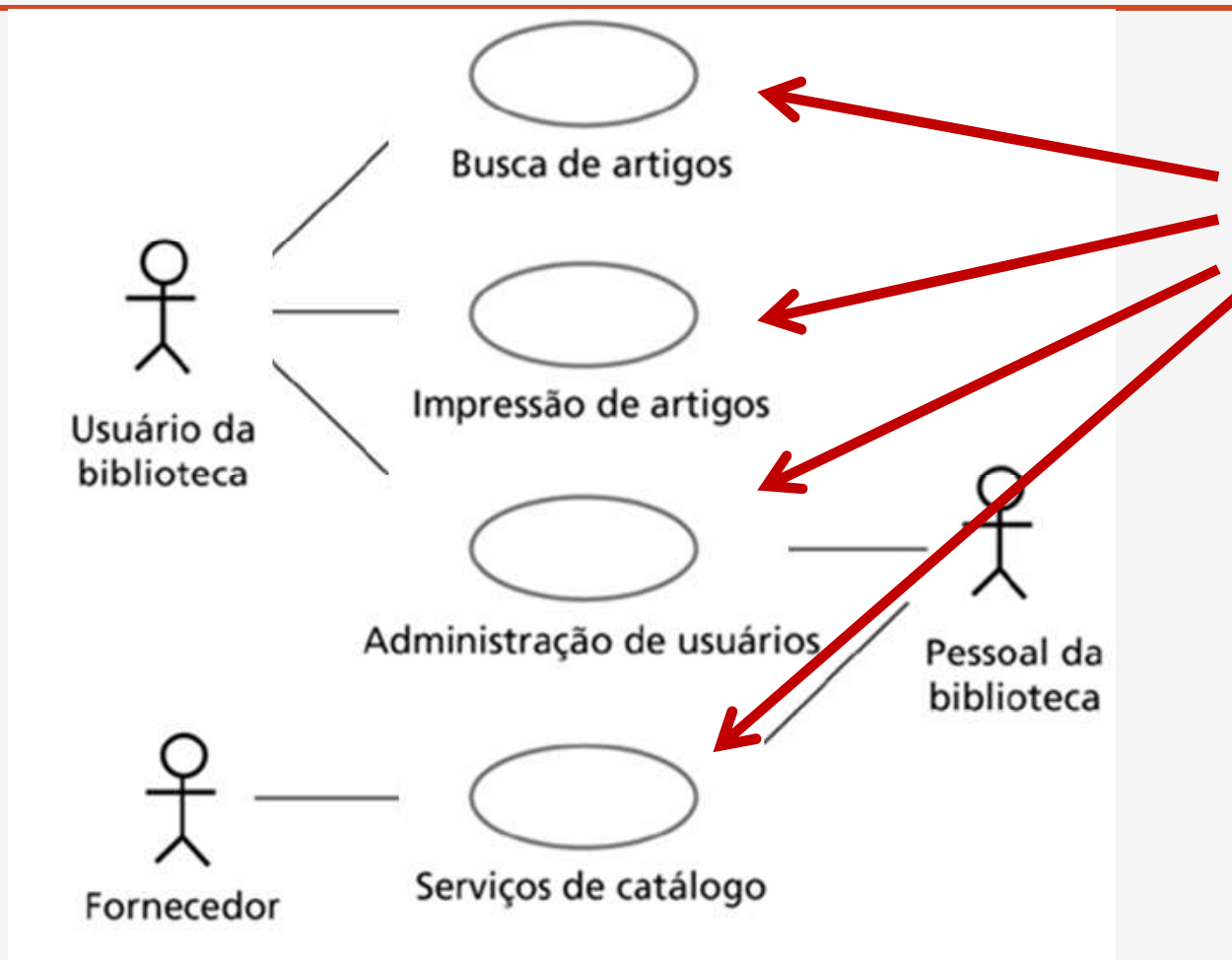
➔ Casos de uso



➔ Relacionamentos entre os elementos atores e casos de uso

**Figura 7.6**

Casos de uso para o sistema de biblioteca.



**CASOS DE USO**

# Casos de uso

---

- Uma funcionalidade do software
- Atômica, completa (não uma fração)
- Externamente perceptível
- EX: cada uma das opções do menu de um caixa eletrônico de banco
  - *emissão de extrato de conta corrente*
  - *Realizar saque*
  - *Transferir dinheiro*



# Mono sessão

---

- Um bom caso de uso deve ser mono sessão. Isso significa que ele deve iniciar e terminar sem ser interrompido.
- Por exemplo, o registro de uma encomenda de livros é feito em uma única sessão de uso do sistema.

## Dica

---

Pode-se pensar assim:

- Somente será um caso de uso um **processo completo**
- Um usuário iria ao computador, ligaria o sistema, executaria o processo e em seguida poderia desligar o computador porque o processo estaria completo.

# Ator

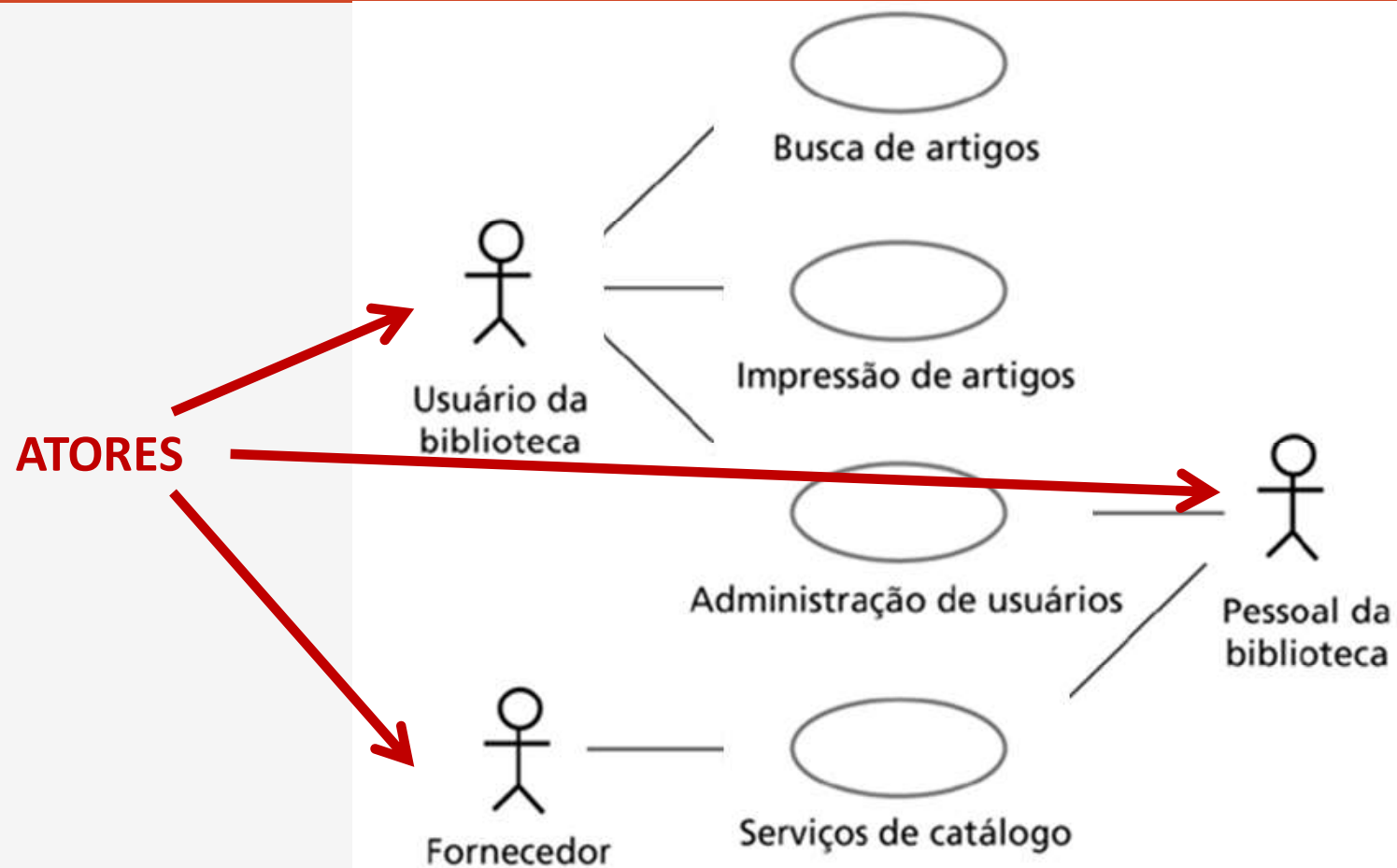
---

- É alguém ou algo que interage com o sistema, ou seja, é quem usa o sistema.
- Um ator pode ser um humano, outro sistema ou um software
  - *Vendedor (pessoa)*
  - *Impressora (hardware)*
  - *Sistema bancário (outro software)*



**Figura 7.6**

Casos de uso para o sistema de biblioteca.



# Associação entre ator e caso de uso

---

- Participação do ator na utilização de um caso de uso
  - Um ator sempre está associado a pelo menos um caso de uso
  - Um caso de uso sempre estará associado a pelo menos um ator