



# Análise de Sistemas

*Eliaana Rovay Detregiacchi Pires*



**Cuiabá-MT  
2014**

**Presidência da República Federativa do Brasil**  
**Ministério da Educação**  
**Secretaria de Educação Profissional e Tecnológica**  
**Diretoria de Integração das Redes de Educação Profissional e Tecnológica**

© Este caderno foi elaborado pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia-RO, para a Rede e-Tec Brasil, do Ministério da Educação em parceria com a Universidade Federal do Mato Grosso.

**Equipe de Revisão**

**Universidade Federal de Mato Grosso – UFMT**

**Coordenação Institucional**  
Carlos Rinaldi

**Coordenação de Produção de Material Didático Impresso**  
Pedro Roberto Piloni

**Ilustração**  
Tatiane Hirata

**Diagramação**  
Tatiane Hirata

**Revisão de Língua Portuguesa**  
Ewerton Viegas Romeo Miranda

**Revisão Final**  
Claudinet Antonio Coltri Junior

**Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO**

**Campus Porto Velho Zona Norte**

**Direção-Geral**  
Miguel Fabrício Zamberlan

**Direção de Administração e Planejamento**  
Gilberto Laske

**Departamento de Produção de EaD**  
Ariádne Joseane Felix Quintela

**Coordenação de Design Visual e Ambientes de Aprendizagem**  
Rafael Nink de Carvalho

**Coordenação da Rede e-Tec**  
Ruth Aparecida Viana da Silva

**Projeto Gráfico**

Rede e-Tec Brasil / UFMT

**Análise de Sistemas - Informática para Internet**

P667a Pires, Eliana Rovay Detregiacchi.

Análise de sistemas / Eliana Rovay Detregiacchi Pires; org. Instituto Federal de Educação, Ciência e Tecnologia; Universidade Federal do Mato Grosso - Cuiabá : UFMT; Porto Velho: IFRO, 2014.

87 p. ; -- cm.  
Curso Informática para internet.

ISBN 978-85-68172-01-8

1. Análise de sistemas. 2. Modelagem de sistemas. 3. Desenvolvimento de sistemas. 4. Projeto de software. I. Instituto Federal de Educação, Ciência e Tecnologia. II. Universidade Federal do Mato Grosso. III. Título.

CDD 001.642068  
CDU 004.45

# Apresentação Rede e-Tec Brasil

Prezado(a) estudante,

Bem-vindo(a) à Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino que, por sua vez, constitui uma das ações do Pronatec - Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (Setec) e as instâncias promotoras de ensino técnico, como os institutos federais, as secretarias de educação dos estados, as universidades, as escolas e colégios tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade e ao promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e a realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

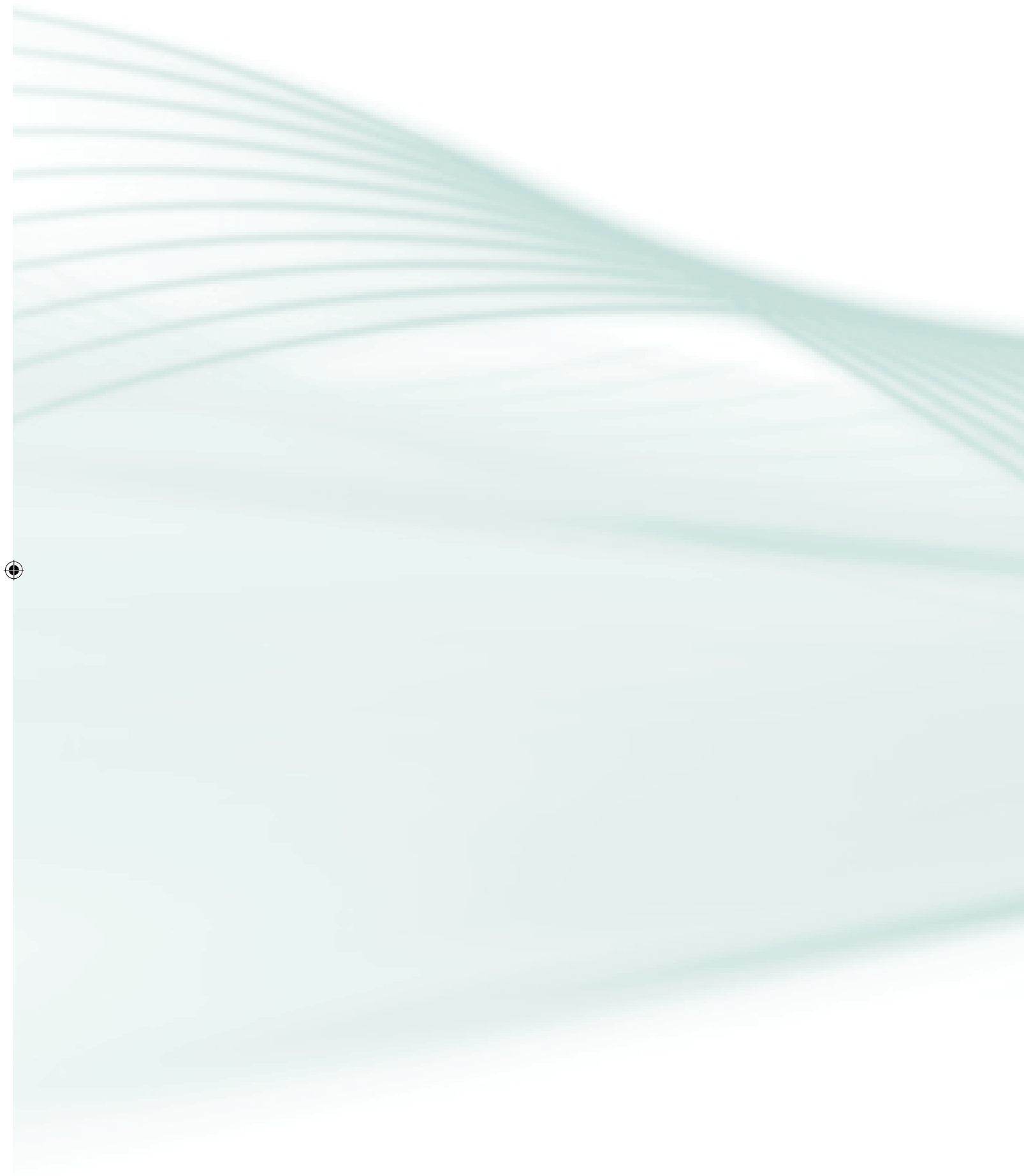
Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e da educação técnica – capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação  
Abril de 2014

Nosso contato  
[etecbrasil@mec.gov.br](mailto:etecbrasil@mec.gov.br)



# Indicação de Ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



**Atenção:** indica pontos de maior relevância no texto.



**Saiba mais:** oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



**Glossário:** indica a definição de um termo, palavra ou expressão utilizada no texto.



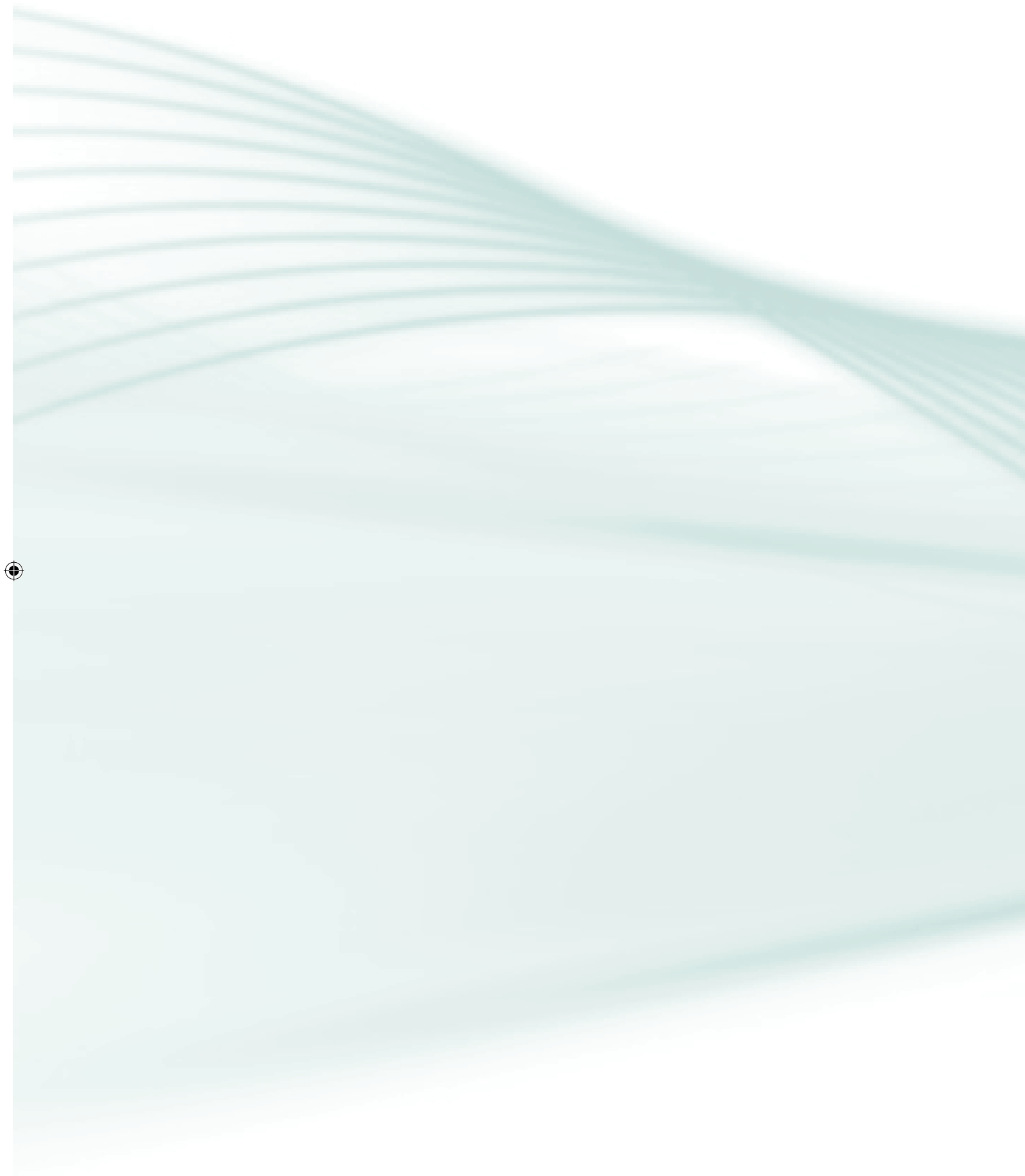
**Mídias integradas:** remete o tema para outras fontes: livros, filmes, músicas, *sites*, programas de TV.



**Atividades de aprendizagem:** apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.



**Reflita:** momento de uma pausa na leitura para refletir/escrever sobre pontos importantes e/ou questionamentos.



## Palavra da Professora-autora

Prezado(a) estudante,

Primeiramente quero lhe dar as boas-vindas à disciplina de Análise de Sistemas através do curso da Rede e-Tec Brasil.

Espero que você esteja bastante motivado(a) para abraçar esta oportunidade de aprendizado que pode lhe encaminhar para um mercado de trabalho promissor.

Desejo também que no decorrer das aulas você possa interagir com o ambiente virtual da Rede e-Tec Brasil para tirar suas dúvidas e reforçar seu aprendizado através dos fóruns, *chats* e outras atividades que o(a) ajudarão na construção de seu conhecimento e no desenvolvimento de suas habilidades. É com prazer e com muito entusiasmo que lhe convido, a partir de agora, para entender um pouco mais a criação dos projetos do mundo virtual da computação.

Fique comigo!

Forte abraço.







# Apresentação da Disciplina

Olá, caro(a) estudante,

É com satisfação que lhe apresento o conteúdo desta disciplina, com carga horária de 40 horas.

Neste caderno você irá estudar sobre análise de sistemas. Este material foi elaborado para capacitá-lo(a) a empregar metodologias de análise e projeto de sistemas para resolução de problemas de clientes, baseados na visão do mundo real.

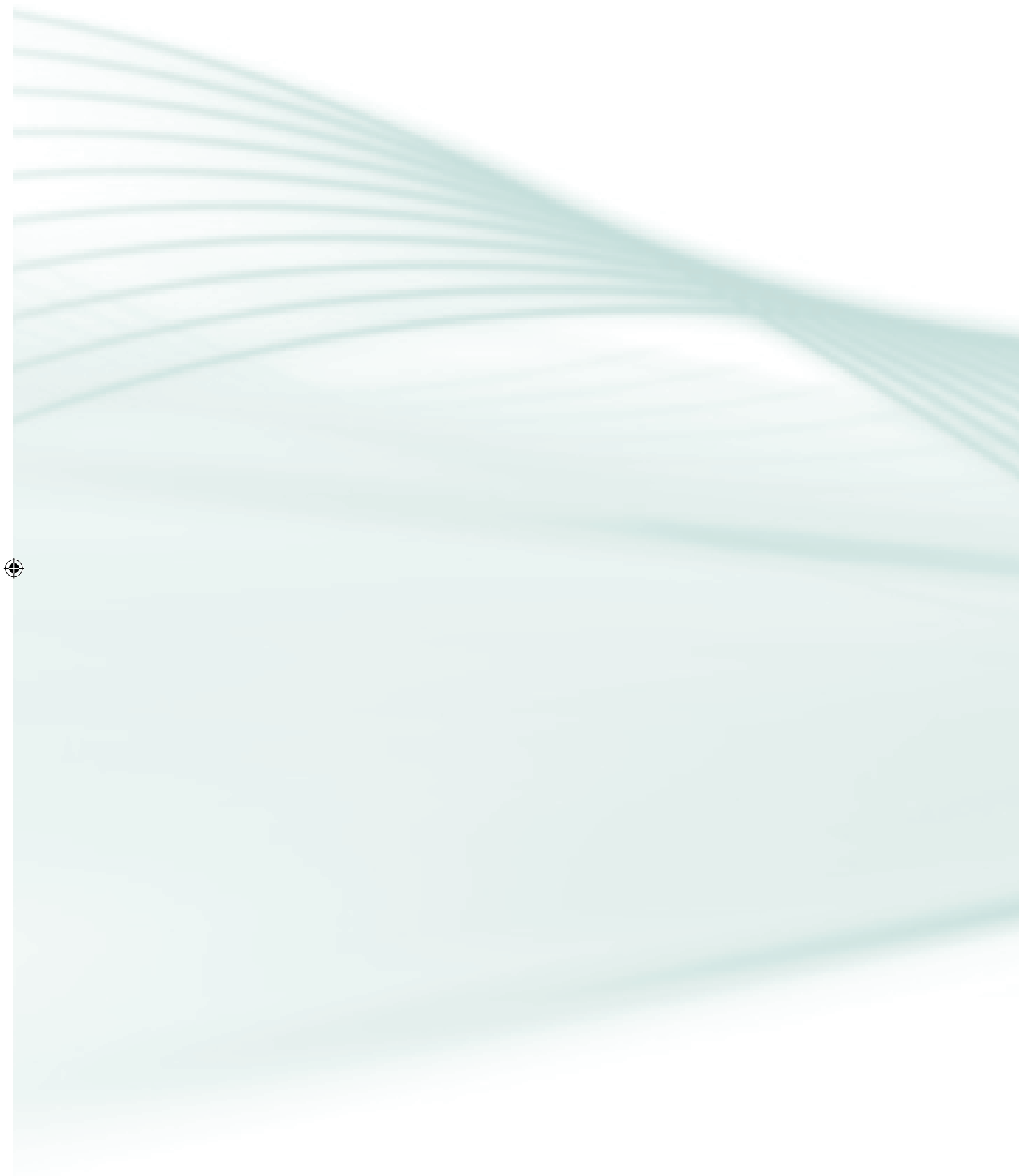
Através desta disciplina você obterá suporte teórico para desenvolver as habilidades empreendedoras buscando soluções por meio do estudo de casos de uso. Aprimorará ainda o raciocínio lógico e a capacidade de abstração para projetar e desenvolver sistemas de software.

As aulas serão bem dinâmicas e sua participação é importantíssima. Sempre que necessário, para suporte ao seu aprendizado, utilize os canais de comunicação que o curso oferece.

Conto com seu entusiasmo e dedicação.

Obrigada e bom curso!

Profa. Eliana Rovay Detregiacchi Pires



# Sumário

<b>Aula 1. Visão geral da teoria de sistemas</b>	<b>15</b>
1.1 Introdução	15
1.2 Teoria geral dos sistemas	16
1.3 Conceito de dado, informação e sistema de informação	17
1.4 Conceitos de análise de sistemas e analista de sistemas	18
<b>Aula 2. Modelagem de sistema</b>	<b>21</b>
2.1 Introdução	21
2.2 Conceito de modelo de sistema	21
2.3 Conceito de sistema orientado a objeto	23
<b>Aula 3. Processo de desenvolvimento de sistema</b>	<b>31</b>
3.1 Introdução	31
3.2 Processo de desenvolvimento	31
<b>Aula 4. Metodologia para desenvolvimento de projeto de software</b>	<b>39</b>
4.1 Introdução	39
4.2 Ciclos iterativos	41
4.3 Caso de uso	41
4.4 Componentes de um modelo de caso de uso	44
4.5 Documentação do modelo de caso de uso	44
<b>Aula 5. Modelagem de classe de objeto</b>	<b>57</b>
5.1 Introdução	57
5.2 Identificação de classes	58
5.3 Identificação de atributos	58
5.4 Identificação de relacionamentos	59
5.5 Relacionamento entre classes	61
5.6 Diagrama de classe	62
<b>Aula 6. Diagrama de interação</b>	<b>67</b>
6.1 Introdução	67



6.2 Componentes do diagrama de sequência.....	68
6.3 Diagrama de sequência.....	71
<b>Aula 7. Desenvolvendo a análise de sistema.....</b>	<b>77</b>
7.1 Introdução.....	77
7.2 Primeiro passo: entrevista com cliente.....	78
7.3 Segundo passo: especificação de requisitos.....	78
7.4 Terceiro passo: diagrama de casos de uso.....	78
7.5 Quarto passo: diagrama de classes.....	79
7.6 Quinto passo: diagrama de sequência.....	79
<b>Palavras Finais.....</b>	<b>81</b>
<b>Guia de Soluções.....</b>	<b>82</b>
<b>Referências.....</b>	<b>86</b>
<b>Currículo da Professora-autora.....</b>	<b>87</b>



# Aula 1. Visão geral da teoria de sistemas

## Objetivos:

- identificar a origem da teoria geral dos sistemas, bem como os conceitos de dado, informação e sistema de informação; e
- reconhecer o que é análise de sistemas e o que faz o analista de sistemas.

Caro(a) estudante,

Bem-vindo(a) à nossa primeira aula! Espero que possa percorrer o caminho da nossa disciplina da melhor maneira possível. Passaremos juntos boa parte da caminhada de sua formação. Espero contribuir para a sua aprendizagem. Bons estudos.

## 1.1 Introdução

Nos dias de hoje, a informação, sob o impacto da utilização de tecnologia, tem uma influência cada vez maior no mundo dos negócios. A empresa que dispõe de mais informações sobre seu processo de negócio está em vantagem competitiva. Isso coloca a informação como o recurso-chave de competitividade efetiva e de diferencial de mercado.

A informação também é considerada e utilizada como um fator que instrumentaliza e estrutura a tomada de decisão dentro de uma organização. Dessa forma, é importante que sua leitura seja feita de forma mais precisa possível, para que os tomadores de decisão não possam vir a incorrer em erros.

Na criação de um sistema de informação, um dos seus componentes é o sistema de software, compreendido de seus módulos funcionais para automatizar e controlar as diversas tarefas.

Antes de tudo, vamos entender o que é um sistema?



## 1.2 Teoria geral dos sistemas

Um sistema é definido como um complexo de elementos em interação, sendo esta de natureza ordenada, gerando uma unidade.

Então, de acordo com a teoria geral dos sistemas:



Sistema é um conjunto de partes inter-relacionadas que se interagem para alcançar um objetivo comum.

Todo sistema é composto por subsistemas. Além de visualizá-lo e entendê-lo na sua totalidade, temos que observar e compreender as partes que se interagem para formá-lo.

Ex.: o corpo humano é um sistema formado de cabeça, tronco e membros (subsistemas).

Um sistema computacional de RH é composto pelos subsistemas de folha de pagamento, controle de frequência, gestão de pessoas etc.

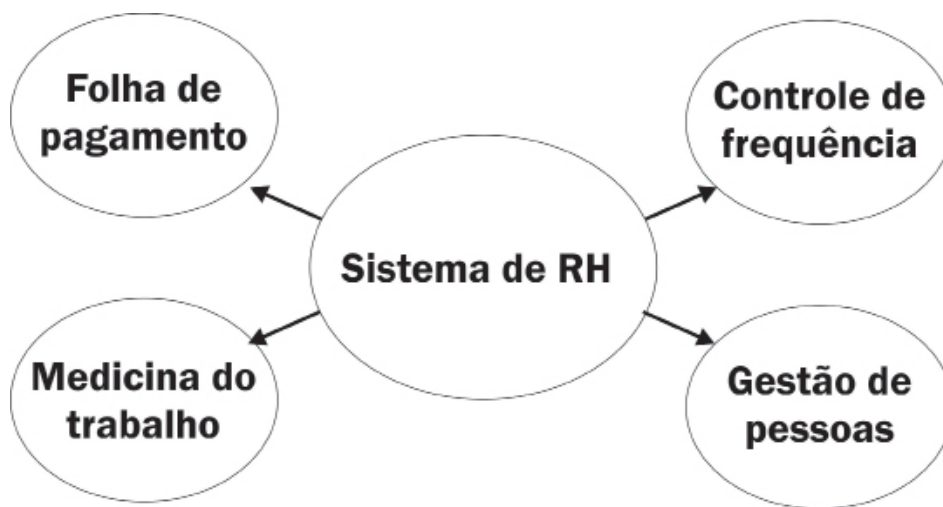


Figura 1. Exemplo de sistema

Fonte: autor



**Sistemas físicos ou concretos:** quando compostos de equipamentos, são denominados “hardware”.

**Sistemas abstratos ou conceituais:** quando compostos de conceitos, planos, hipóteses e ideias, são denominados “software”.



Vamos entender agora outros conceitos: o que é dado, informação e sistema de informação?

### 1.3 Conceito de dado, informação e sistema de informação

Dado é qualquer elemento identificado em sua forma bruta que por si só não conduz a uma compreensão de determinado fato ou situação (OLIVEIRA, 2005). Representa a matéria-prima a ser utilizada na produção de informações. Os dados partem inicialmente do olhar do observador sobre o objeto a ser identificado na forma visual ou simbólica. Ex.: nomes, número, modelo etc.

Informação é o dado processado. Ex.: relatório contábil, nome de cliente, data de nascimento etc.

E então, o que é sistema de informação?

Sistema de informação é um conjunto de regras e procedimentos organizados para o fornecimento de informações para o usuário. Ex.: sistema contábil.



Todo sistema de informação é composto por três fases:

- a entrada de dados (considerados elementos brutos);
- o processamento desses dados em função das necessidades de informação a serem geradas; e
- a saída em forma de informação.

O processo pode ser realimentado pelo mecanismo de *feedback*, pelo qual as saídas influenciam as novas entradas.

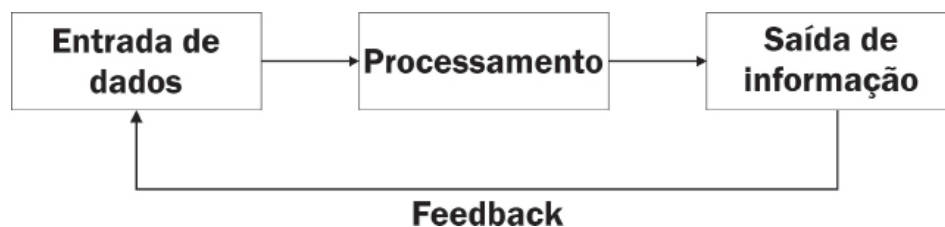


Figura 2. Modelo genérico de sistema

Fonte: autor



## 1.4 Conceitos de análise de sistemas e analista de sistemas

Vamos compreender o que é análise de sistemas:



**Análise de sistemas** é a atividade de realizar estudo dos processos, métodos e técnicas de investigação e especificação da solução do problema, a fim de, a partir dos requisitos levantados, encontrar o melhor caminho para a criação de software.

A análise é o processo de separação das partes de um sistema para facilitar o entendimento de sua natureza, funções e relações.

Podemos definir também análise de sistemas como o processo de analisar, projetar e implementar sistema de software.

E o analista de sistemas?



**Analista de sistemas** é o(a) profissional que define o que, para que e de que forma será desenvolvida uma solução executada pelo computador.

### Resumo

Nesta primeira aula vimos que, nos dias de hoje, a informação, sob o impacto da utilização de tecnologia, tem uma influência cada vez maior no mundo dos negócios. Apresentei também o conceito de sistema (um conjunto de partes inter-relacionadas que se interagem para alcançar um objetivo comum) através da teoria geral dos sistemas.

Conceituei ainda dados, informações e sistemas de informações. Vimos que dado é qualquer elemento identificado em sua forma bruta que por si só não conduz a uma compreensão de determinado fato ou situação, que informação é o dado processado e que sistema de informação, por sua vez, é um conjunto de regras e procedimentos organizados para o fornecimento de informações para o usuário.

Por fim, apresentei-lhe que análise de sistemas é a atividade que consiste em realizar estudo dos processos, métodos e técnicas de investigação e especificação da solução do problema, com a finalidade de, a partir dos requisitos levantados, encontrar o melhor caminho para a criação de software, e que analista de sistemas é o(a) profissional que define o que, para que e de que forma será desenvolvida uma solução executada pelo computador.





## Atividade de aprendizagem



Desenvolva uma pesquisa sobre o papel do(a) analista de sistemas e seu perfil profissional, bem como os tipos de sistemas de informação existentes. Utilize a internet, livros ou outras mídias para concluir sua pesquisa.

---

---

---

---

---

---

---

---

---

---

Apresentei no conteúdo desta aula inicial alguns conceitos importantes, como o que é sistema, dado, informação e análise de sistemas, para que você, caro(a) estudante, possa começar a entender a composição e a estrutura conceitual de sistema. Assim terminamos a nossa primeira etapa. Vamos em frente?





# Aula 2. Modelagem de sistema

## Objetivos:

- reconhecer o conceito de modelo do sistema; e
- identificar o que é um sistema orientado a objeto.

Prezado(a) estudante,

Bem-vindo(a) à nossa segunda aula! Nesta aula você terá a oportunidade de compreender alguns conceitos fundamentais para análise e projeto de sistemas e a definição de análise orientada a objeto. Um dos intuitos é que você possa conhecer e perceber a importância do uso dessa metodologia a ser aplicada na modelagem de uma solução de sistema de software.

## 2.1 Introdução

Para a criação de um sistema de informação, primeiramente, é essencial definir os objetivos a serem alcançados. Independentemente do método a ser adotado para a análise e desenvolvimento de um sistema, tudo se inicia com uma investigação das necessidades pelo cliente, seja ele interno (colaboradores dentro da mesma organização) e/ou externo. Assim, todo sistema de informação é concebido para resolver um ou mais problema(s).

É preciso esgotar toda discussão sobre as necessidades reais para solucionar o problema em questão, garantindo que o objetivo do sistema seja alcançado. E a complexidade do problema cresce à medida que cresce o tamanho do sistema. Para entender a complexidade do problema e definir um modelo ideal de solução, precisamos conhecer alguns conceitos da análise de sistema que vão ajudar a moldar a melhor solução para um problema em estudo.

## 2.2 Conceito de modelo de sistema

O modelo conceitual descreve a visão que o usuário tem das informações existentes que serão gerenciadas pelo sistema. Essas informações existentes



independem de um computador para transformá-las. O objetivo da análise é estudar o problema, e o sistema computacional é uma solução automatizada para resolvê-lo.

Imagine agora que você tenha que construir uma simples cadeira. A dificuldade que você encontraria poderia ser medida considerando o tempo e os recursos necessários para a construção dessa cadeira. Você precisaria de madeira, pregos, cola e ferramentas apropriadas. E o que fazer para construir um mobiliário completo, como mesa e cadeiras? Com certeza não seria uma tarefa tão simples. O tempo e os recursos seriam bem maiores. Certamente para construir um mobiliário completo é necessário um planejamento de design a ser seguido para a sua construção.

Assim como na construção do mobiliário completo, na construção de sistemas de software também é necessário um planejamento inicial, equivalente ao projeto do design do mobiliário. Isso nos leva ao conceito de modelo.



Modelo é uma representação do sistema a ser desenvolvido. Ex.: desenho da interface gráfica do sistema.

Por que construir modelo do sistema?

- Para ter um entendimento melhor da complexidade do sistema. Modelos descrevem as características relevantes de um sistema. Os detalhes não relevantes só aumentam a complexidade no entendimento do problema e podem ser ignorados aplicando o princípio da abstração.

Usamos o conceito de abstração quando nos concentramos nas coisas que consideramos essenciais e desprezamos as outras não relevantes dentro do contexto da análise. É a visão que obtemos da realidade do mundo real. Ex.: conceito de cadeira (assento, encosto) – abstração.

- Para difundir as informações relativas ao sistema entre os indivíduos envolvidos no projeto.
- Para gerenciar os custos no desenvolvimento do sistema. Erros identificados na fase de construção do modelo têm impacto menos desastroso.



Os modelos são representados por diagramas, que é a forma gráfica de expressar a ideia do sistema a ser desenvolvido.



## 2.3 Conceito de sistema orientado a objeto

Vamos voltar ao conceito de sistema, estudado na primeira aula, em que estabelecemos que um “sistema é um conjunto de partes inter-relacionadas que se interagem para alcançar um objetivo comum”.

Fazendo uma analogia, um sistema de software pode ser formado por objetos que se interagem através de mensagens para que uma tarefa computacional seja realizada.

Aplicada, a orientação a objetos na modelagem de sistemas de software faz com que os modelos construídos para solucionar um problema fiquem mais próximos do mundo real e estabelece que qualquer coisa é um objeto que pertence a uma classe, e esta é um repositório de informação associado ao objeto.

- Para que você, caro(a) estudante, compreenda melhor a construção dessa modelagem, vou exemplificar o conceito de classe, objeto e suas características.

### 2.3.1 Conceito de classe e objeto

Definimos como objeto todas as coisas do mundo real e que fazem parte de uma classe. Diz-se que um objeto é uma instância de uma classe.

Objeto é qualquer elemento do mundo real.

Ex.: bicicleta, professor, conta bancária, ícone etc.

Atributo é um conjunto de características de um objeto.

Ex.: nome de um livro, saldo bancário do cliente, modelo de um carro, cor da caneta, graduação do professor etc.

Classe ou classe de objeto é uma abstração das características relevantes de coisas do mundo real. É um conjunto de objetos que têm a mesma especificação.



Ex.1: Classe – aluno.

Objeto da classe - Pedro, Maria etc.



Atributos – matrícula, nome, endereço, notas, curso etc.

Ex.2: Classe – veículo.

Objeto da classe – ônibus, carro, moto etc.

Atributos - motor, marca, modelo etc.

### 2.3.2 Conceito de método e mensagem

Os métodos descrevem o comportamento do objeto, que são suas funcionalidades (funções, operações).

Os objetos interagem através do envio de mensagens. Isso significa que os objetos estão enviando mensagens uns aos outros com o objetivo de realizar alguma operação (requisitando serviço) no sistema.

Ex.: Classe: impressora.

Atributos: velocidade, resolução, processador.

Métodos: ligar / desligar, imprimirarquivo.

Ex.: o objeto **Relatório Cadastro de Fornecedor** envia mensagem **ImprimirArquivo** para o objeto impressora.

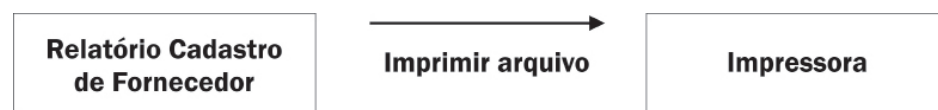


Figura 3. Representação gráfica de envio de mensagem entre objetos.

Fonte: autor





Ex.: Classe de objeto: aluno.

Atributos: nome, matrícula, nota, média do aluno.

Método: calcularmédia (operação).

### 2.3.3 Princípio da abstração

O princípio da abstração, dentro do conceito da análise orientada a objeto, define alguns mecanismos de análise para gerenciar a complexidade do problema e concentrar a atenção nas características essenciais de um objeto, como os mecanismos encapsulamento, polimorfismo e herança.

#### 2.3.3.1 Encapsulamento

O mecanismo de encapsulamento restringe o acesso às operações internas realizadas por um objeto. Um objeto envia mensagem a outro objeto para realizar determinada tarefa, sem se preocupar com o modo como a tarefa é realizada. A aplicação desse mecanismo esconde os detalhes de funcionamento interno de um objeto. Nesse mecanismo, são disponibilizadas para o usuário apenas as funções que o objeto executa através da interface do sistema e encapsula a sua implementação, protegendo assim o acesso indiscriminado aos dados.

O acesso é feito através de mensagens (operações) que são trocadas entre objetos. A implementação do objeto pode ser modificada sem perder a sua funcionalidade e a maneira como é operada e conhecida pelo usuário.



**Figura 4. Exemplo de encapsulamento.**

Fonte: autor

Na figura acima, temos o objeto computador e o objeto data show, em que cada um tem suas funções. A integração entre eles é feita através da ligação da saída de um objeto à entrada do outro objeto. O data show pode reproduzir as imagens do computador. A implementação está encapsulada no próprio objeto. O usuário precisa saber apenas onde conectar a entrada de



um objeto e a saída do outro objeto para produzir a imagem.

### 2.3.3.2 Polimorfismo

No mecanismo denominado polimorfismo, um objeto pode enviar a mesma mensagem para objetos semelhantes, mas as interfaces são implementadas de formas diferentes.

Ex.:



**Figura 5. Exemplo de polimorfismo. O objeto controle remoto envia mensagem (requisitando serviço) para o objeto DVD e o objeto televisão.**

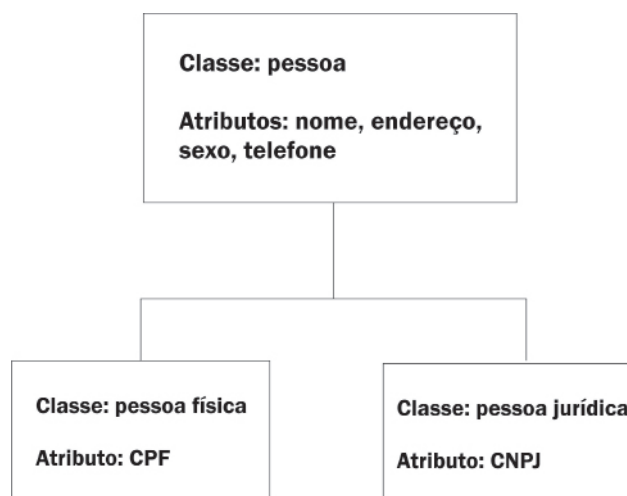
### 2.3.3.3 Generalização de abstração (herança)

O mecanismo conhecido por generalização de abstração (ou simplesmente herança) ocorre quando definimos um subconjunto de relacionamentos entre elementos de duas ou mais classes.

Ex.: classe veículo é a generalização da classe motocicleta, pois toda motocicleta é um veículo.

Por esse mecanismo podemos estabelecer relações entre as classes, compartilhando os atributos e operações semelhantes.

Ex.: Classe pessoa



**Figura 6. Exemplo de herança.**





## 2.3.4 Simbologia de classe

Padrões para definir os nomes das classes, objetos e atributos:

- usar nome no singular;
- sem hífens, tracinhos, espaços; e
- sem preposição (de, para, do etc).

Ex: Classe: cliente, produto, fornecedor, nomealuno etc.

NomeClasse
Atributos
Operações ( )

Figura 7. Representação gráfica de uma classe.

## Resumo

Nesta aula você teve a oportunidade de aprender que modelo é uma representação do sistema a ser desenvolvido (ex.: desenho da interface gráfica do sistema) e que modelagem de sistemas de software consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema. Viu ainda o que é classe de objeto, método e mensagem, os princípios da abstração e a simbologia de classe.

## Atividades de aprendizagem

1. Identifique as classes e seus atributos para os contextos:

**a)** A faculdade está ofertando o curso de técnicas em modelagem três vezes na semana e aos sábados das 8 às 12hs.

**b)** O exército brasileiro está recrutando soldados de todos os estados para compor uma equipe de elite, com formação de nível superior e que esteja atuando na base há mais de 4 anos, com idade mínima de 25 anos e que seja do sexo masculino.

---

---

---

---

---

---





2. Conhecendo os atributos e operações, determine as classes correspondentes:

a) Classe: \_\_\_\_\_

Atributos: nome, matrícula, datanascimento, nota, média.

Operações: consultarnota, calcularmédia.

b) Classe: \_\_\_\_\_

Atributos: nome, cargahorária, professorministrante.

Operação: atualizarcargahorária.

3. Identifique nas classes definidas abaixo quais atributos não fazem parte do escopo do problema:

a) Classe: aluno

Atributos: nomealuno, matrícula, endereço, sexo, corcabelo, nota, média.

Operação: consultarnota, calcularmédia.

b) Classe: funcionário

Atributos: nomefunc, funçãofunc, preferênciassexual, saláriofunc.

Operação: calcularreajustesalarial.

4. Quantas classes podem ser identificadas com as figuras abaixo? E quais são?

Professor    Violão    Computador    Gato    Carro

Piano    Tablet    Cachorro    Coordenador

---

---

---

---



5. Exemplifique o relacionamento de herança entre as classes pessoa e funcionário.

---

---

---

---

6. Analisando os substantivos abaixo, identifique os que são candidatos à classe.

a) Livro, autor, número de páginas, número da estante, assunto.

b) Carro, marca, motor, potência, número de chassis, piloto, autódromo, número de voltas.

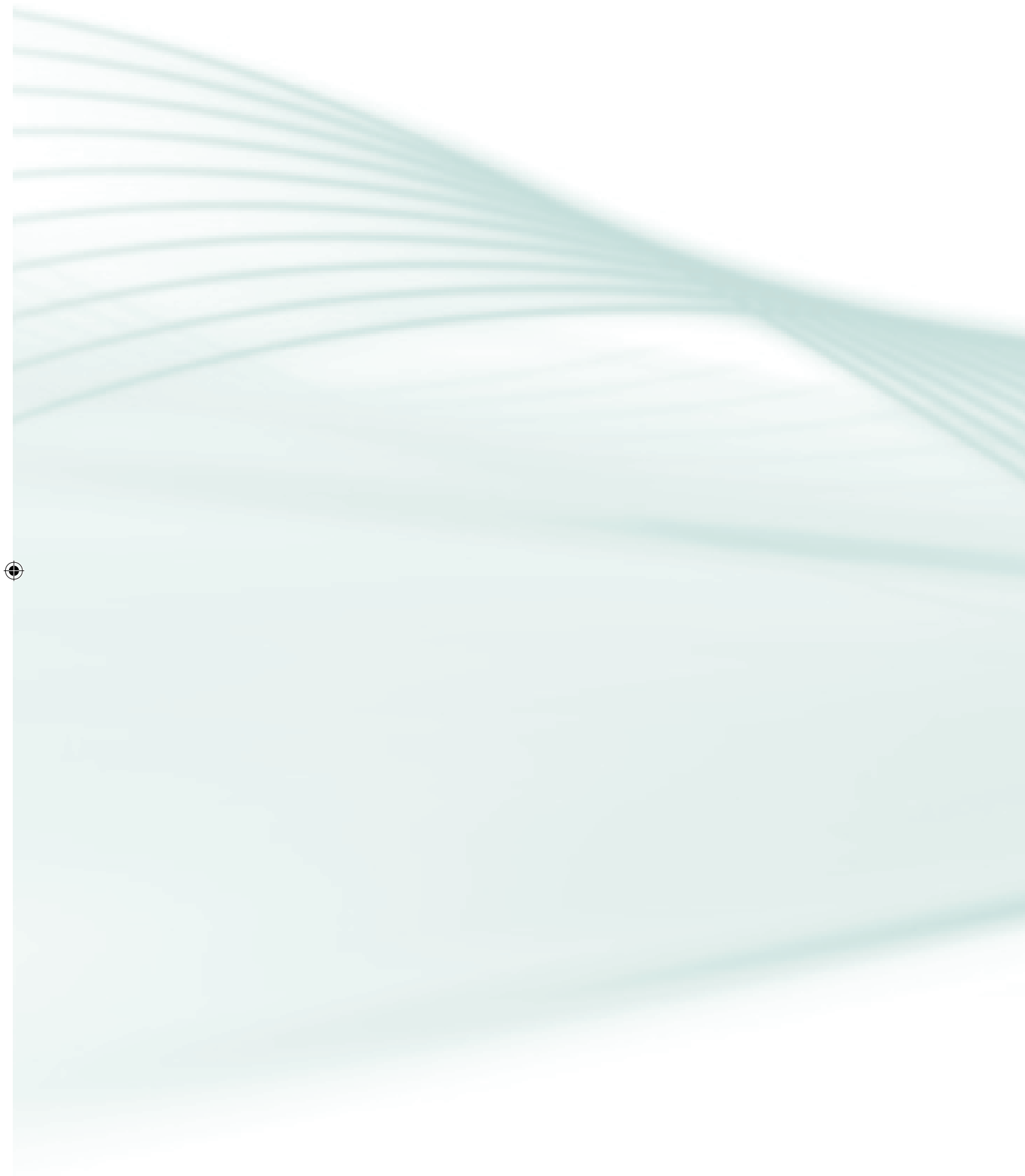
---

---

---

---

Chegamos ao fim do nosso segundo encontro. Você viu nesta segunda aula a utilização dos conceitos da análise de sistema orientada a objeto, que modela todas as coisas do mundo real em objetos do mundo virtual, com suas características e classificações para apoiar a criação da melhor solução através de uma representação gráfica de um problema em estudo. Espero você na próxima aula. Até lá!



# Aula 3. Processo de desenvolvimento de sistema

## Objetivos:

- identificar as fases de análise de sistema; e
- construir um modelo ideal para a solução de um problema real.

Caro(a) estudante,

Bem-vindo(a) à nossa terceira aula! Os motores já estão esquentando! Nesta aula você vai conhecer as fases da análise de sistema, e trataremos, principalmente, da fase que é o levantamento das informações para construir um modelo ideal para a solução de um problema real. Mãos à obra!

## 3.1 Introdução

O principal objetivo do levantamento de requisitos para o desenvolvimento de um software é que o usuário (cliente) e o analista (desenvolvedor) tenham a mesma visão do problema a ser resolvido, e que juntos consigam definir as necessidades (os requisitos) para a solução do problema.

O levantamento de requisitos que irá compor o sistema é a etapa mais importante, porque é nessa fase que será definida a qualidade do sistema, que é medida pela usabilidade do software. E a usabilidade será medida se os requisitos atenderem às necessidades do usuário.

Um sistema de informação é utilizado para automatizar processos, e esses processos e seus requisitos devem ser compreendidos antes do desenvolvimento do sistema de informação.

## 3.2 Processo de desenvolvimento

O processo de desenvolvimento de software compreende as atividades de levantamento de requisitos, análise, projeto, implementação e teste.



### 3.2.1 Levantamento de requisitos

Na fase de levantamento de requisitos, trabalha-se para levantar, analisar, documentar e validar as necessidades do cliente ou de algum produto novo dentro de um projeto de software. Essas necessidades são inicialmente definidas como funcionalidades, ideias ou desejos de como o sistema deve funcionar, e normalmente são transformadas em requisitos funcionais e não funcionais do projeto.

Mas o que são requisitos?



Requisito é o que o sistema tem que ter para atender os objetivos para o qual o sistema é criado.

E o que é requisito funcional?



Definimos a funcionalidade a ser desenvolvida para automatizar uma necessidade do sistema.

Ex.: funcionalidade: existe uma necessidade de definir uma funcionalidade no sistema para gerar um relatório de alunos por período.

Requisito funcional: o sistema deve permitir a geração de um relatório com todas as informações de alunos ocorridas entre duas datas informadas pelo usuário.

E requisito não funcional?



São definidos como requisitos não funcionais aqueles que descrevem apenas atributos do sistema ou atributos do ambiente do sistema. São os requisitos que geralmente estão mais ligados ao uso do software.

Ex.: o usuário, para ter acesso ao módulo de relatório através da interface do sistema, precisa estar cadastrado e possuir um login e senha.

Uma maneira de reconhecer os requisitos do sistema é criar um questionário intuitivo para identificar as funcionalidades do sistema e produzir um documento texto chamado de sumário executivo, que deve descrever o que o(a) analista conseguiu pesquisar de relevante sobre o sistema em estudo e o que o cliente espera que o sistema faça.



Ex.: Questionário:

- Qual o nome da funcionalidade? Cadastro de aluno.
- Quais as informações que irão compor esse cadastro? Nome, endereço, telefone, CPF, RG, e-mail etc.
- Onde essas informações serão armazenadas? Banco de dados, acesso pela rede.

Com essas perguntas básicas, definimos alguns requisitos funcionais e não funcionais para criar um cadastro de aluno.

- Certamente, para identificar todos os requisitos de um sistema, é necessário um questionário mais detalhado.

Fazendo uma analogia, vou exemplificar um estudo de caso conhecido de todos: como realizar o “Churrascão” de domingo.

- Nome da funcionalidade: Churrascão de domingo.
- Quais os requisitos funcionais do Churrascão: carne, tempero, carvão, bebida, farofa, arroz.
- Quais os requisitos não funcionais do Churrascão: onde o evento (churrascão) irá acontecer.

Vou aplicar agora outra metodologia para especificar os requisitos, exemplificando um sumário executivo para um sistema em que as funcionalidades são bem conhecidas de todos.

Ex.: sumário executivo: sistema de biblioteca.

Nesse documento é proposto o desenvolvimento de um sistema de controle de uma biblioteca para informatizar as funções de empréstimo e devolução de livros. O sistema deverá gerar relatório mensal de empréstimos por livro e usuários cadastrados e calcular automaticamente os pagamentos de multas em caso de devolução fora do prazo estipulado pelo sistema. O sistema deverá, ainda, inabilitar o cadastro de usuários da biblioteca que não devolverem os livros emprestados.



Esse documento descreve as principais necessidades e funcionalidades do sistema a ser desenvolvido, e que serão mais bem estruturadas nas outras fases do desenvolvimento.

Podemos criar uma tabela para identificar os requisitos para o sistema proposto no sumário executivo.

Requisito funcional: emprestar livro	
<b>Descrição:</b> o sistema deve registrar empréstimos de livros, informando o usuário e o livro emprestado, a data de empréstimo e a data prevista de devolução.	
Requisitos não funcionais: controle de acesso, identificação dos livros e identificação de usuários, armazenamento dos dados	
<b>Descrição:</b>	
<b>Controle de acesso</b>	Esta função só pode ser acessada por usuário do sistema com perfil de administrador do sistema.
<b>Identificação de livros</b>	Os livros serão identificados por um código de barras.
<b>Identificação do usuário</b>	O usuário será identificado pelo seu CPF.
<b>Armazenamento dos dados</b>	Os dados serão armazenados em um banco de dados.

Requisito funcional: devolver livro	
<b>Descrição:</b> o sistema deve registrar a devolução de livros, checar a data de devolução, se não estiver dentro do prazo, calcular multa sobre os dias de atraso.	
Requisitos não funcionais: controle de acesso, identificação dos livros e identificação de usuários	
<b>Descrição:</b>	
<b>Controle de acesso</b>	Esta função só pode ser acessada por usuário do sistema com perfil de administrador do sistema.
<b>Identificação de livros</b>	Os livros serão identificados por um código de barras.
<b>Identificação do usuário</b>	O usuário será identificado pelo seu CPF.

**Figura 8. Exemplo de tabelas que compõem os requisitos funcionais e não funcionais.**

- A fase de levantamento de requisitos deve ser uma fase de descoberta, em que o analista, cliente e usuários vão identificando as funcionalidades e as restrições. Os requisitos não identificados nessa fase deverão ser realocados ao longo do restante do processo de desenvolvimento. É possível também que os requisitos mudem durante o processo de desenvolvimento, e deve-se gerenciar esse tipo de mudança para eventualmente realocar novos requisitos.





### 3.2.2 Análise

Na fase da análise faz-se a investigação do problema. Essa fase é importante para entender com clareza um problema de sistema de informação, e para isso, o problema deve ser bem investigado.

A análise é para produzir uma ampla compreensão do problema, identificando e avaliando as informações levantadas para o projeto de sistema.



### 3.2.3 Projeto

A fase do projeto propõe uma solução que atenda os requisitos levantados na fase da análise, baseado no conhecimento adquirido.

A fase do projeto trata da construção das especificações detalhadas para a construção do sistema.



Essas especificações incluem o projeto das interfaces, banco de dados, estrutura física e lógica, modelagem, processo, escopo, dicionário de dados e cronograma.

São utilizados nessa fase os diagramas da UML, tais como diagrama de caso de uso, diagrama de classe, diagrama de sequência etc., para a construção do projeto.

### 3.2.4 Implementação

Nesta fase o sistema é codificado, ou seja, ocorre a tradução dos requisitos especificados na fase do projeto em código executável através do uso das linguagens de programação, como Java, frameworks e estruturas de dados.

### 3.2.5 Testes

Nesta fase são realizados testes para verificar se o sistema construído atende às especificações detalhadas na fase do projeto e a geração de relatórios de testes, que contém informações sobre os possíveis erros detectados.

- O envolvimento do cliente usuário é importantíssimo no desenvolvimento de um software e para a satisfação do cliente.



## Resumo

Nesta aula foram apresentadas algumas reflexões sobre a importância da fase de levantamento de requisitos e como são identificados os requisitos funcionais e não funcionais, através de uma tabela demonstrativa ou pelo sumário executivo, para modelar as informações que farão parte do projeto de desenvolvimento de um sistema de informação.



## Atividades de aprendizagem

1. Tomando como base um cenário do mundo real, tente visualizar uma necessidade de controle das informações de uma escola perto de sua casa ou que você conheça. Procure conversar com alguém que tenha o conhecimento do trâmite das informações que devem ser controladas e, fundamentando-se nos exemplos estudados, faça um levantamento de requisitos para o desenvolvimento de um sistema escolar.

Responda o questionário proposto ou crie o seu questionário, baseado na conversa com o seu cliente, nesse caso a escola.

Faça uma explanação inicial ao dirigente da escola sobre o propósito das perguntas para que você, caro(a) estudante, possa criar uma base de conhecimento sobre as necessidades de controlar determinadas informações de uma escola para o desenvolvimento de um sistema de informação.

Questionário proposto:

**a)** Quais os objetivos do sistema?

---

---

---

---

**b)** Quais as atividades relacionadas ao sistema?

---

---

---

---

**c)** Quais os problemas referentes a cada atividade?

---

---

---

---



**d)** Quais as prováveis soluções para os problemas de cada atividade?

---

---

---

---

**e)** Quais as informações envolvidas nas soluções?

---

---

---

---

**f)** Quais as decisões associadas à atividade?

---

---

---

---

**g)** Quais os produtos, bens ou serviços produzidos por cada atividade?

---

---

---

---

**2.** Escreva o sumário executivo para o sistema proposto.

---

---

---

---

---

---

---

---

**3.** Descreva os requisitos funcionais e não funcionais para o sistema proposto.

---

---

---

---



---

---

---

---

Dessa maneira terminamos a nossa terceira aula. Já estamos quase na metade da nossa caminhada. Vamos prosseguir? Encontramo-nos na quarta aula. Até!



# Aula 4. Metodologia para desenvolvimento de projeto de software

## Objetivo:

- identificar a metodologia da linguagem UML.

Prezado(a) estudante,

Bem-vindo(a) à nossa quarta aula! Nesta aula você vai conhecer a metodologia da linguagem UML, aplicada para a análise e desenvolvimento de projeto de software. A UML é uma linguagem de modelagem padronizada que permite visualizar e entender o relacionamento entre os objetos definidos. Vamos à aula.

## 4.1 Introdução

UML quer dizer Unified Modeling Language (Linguagem de Modelagem Unificada) e é uma linguagem que pode ser usada para descrever coisas do mundo real. É uma linguagem padrão para a elaboração da estrutura de projeto de software. A UML é utilizada para a modelagem de sistema, abrangendo todas as visões necessárias ao desenvolvimento desse sistema, independentemente da linguagem de programação a ser utilizada na implementação da solução do problema.

A UML é uma linguagem constituída de elementos gráficos (visuais), e a partir desses elementos podem-se construir diagramas que vão representar as diversas perspectivas para a construção de um sistema orientado a objeto.

- Sintaxe: cada elemento gráfico possui uma forma predeterminada de ser desenhado.
- Semântica: cada elemento gráfico possui um significado e é utilizado para algum fim.

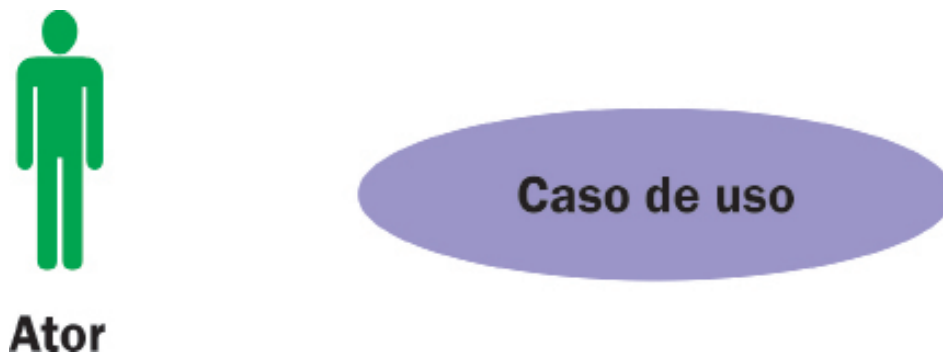


Figura 9. Elementos gráficos da UML.

A utilização da UML como linguagem de modelagem, em um processo de desenvolvimento, envolve a criação de diversos documentos.

- Estes documentos podem ser textuais ou gráficos.
- Estes documentos são denominados artefatos de software.
- São os artefatos que compõem as visões do sistema.
- Os artefatos gráficos produzidos durante o desenvolvimento de um sistema de software são definidos através da utilização dos diagramas da UML.

#### 4.1.1 As 5 visões da UML

Segundo Bezerra (2011), a modelagem da arquitetura de um sistema abrange como o sistema será construído.

- **Visão de caso de uso:** descreve o sistema de um ponto de vista externo como um conjunto de interações entre o sistema e os agentes externos ao sistema.
- **Visão de projeto:** enfatiza as características do sistema que dão suporte às funcionalidades visíveis do sistema.
- **Visão de implementação:** abrange o gerenciamento de versões do sistema, construídas através do agrupamento de módulos e subsistemas.



- **Visão de implantação:** corresponde à distribuição física do sistema em seus subsistemas e à conexão entre essas partes.
- **Visão de processo:** esta visão enfatiza as características de concorrência (paralelismo), sincronização e desempenho do sistema.



Figura 9. Elementos gráficos da UML.

## 4.2 Ciclos iterativos

Uma vez feito o levantamento de requisitos, o próximo passo é organizá-los de forma a abordá-los nos ciclos iterativos, que se baseiam em casos de uso. Cada ciclo iterativo cumpre as fases de análise, projeto e programação.

O modelo iterativo incremental envolve a participação do usuário nas atividades de desenvolvimento do sistema, o que diminui a possibilidade de interpretações erradas em relação aos requisitos levantados.

- **Iterativo:** o sistema de software é desenvolvido em vários passos similares.
- **Incremental:** em cada passo, o sistema é estendido com mais funcionalidades.

## 4.3 Caso de uso

Segundo Jacobson, caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo.



Caso de uso é uma técnica de modelagem usada para descrever o que um novo sistema deve fazer.

- É construído através de um processo interativo em que as discussões entre o cliente e os analistas conduzem a uma especificação do sistema no qual todos estão de acordo com o entendimento do projeto.
- Descreve as operações que o sistema deve executar para cada usuário.
- Apresenta uma lista completa das interações entre um usuário e o sistema para cumprir uma tarefa do início até o fim.

Casos de uso estão associados a um conjunto de requisitos funcionais do sistema e aos atores envolvidos. O objetivo é levantar informações sobre como o sistema interage com os usuários.

- Descrever os requisitos funcionais do sistema.
- Fornecer uma descrição clara do que o sistema deve fazer.
- Permitir a descoberta dos requisitos funcionais das classes e operações do sistema.

O(a) analista deve descrever “o que” acontece entre o usuário e o sistema, sem se preocupar em informar “como” as interações vão ocorrer, nem com qual interface comunicará usuário e sistema.

A narrativa do caso de uso deve enfatizar “o que” um processo faz e não “como” o processo é efetuado. Deve descrever os principais fluxos de ações do sistema e as sequências das possíveis exceções.

Um caso de uso tem passos obrigatórios que envolvem a troca de informações dos atores para o sistema e do sistema para os atores, e a falta deles faz com que o caso de uso esteja incorretamente descrito.

No exemplo da aula anterior, na especificação dos requisitos de um sistema de biblioteca, o fluxo principal do caso de uso de “emprestar livro” deve conter obrigatoriamente os passos que indicam o momento nos quais o atendente (usuário) identifica o cliente e o livro no sistema.





Por que esses passos são obrigatórios?

Porque sem essas informações o sistema de biblioteca seria incapaz de registrar corretamente um empréstimo de livro. Não teria razão nenhuma para um sistema, registrar um empréstimo de livro sem indicar quem é o cliente que emprestou tal material.

A ausência de fluxo de interação que identifica essa troca de informação entre o ator e sistema e sistema e ator deixa os casos de uso sem sentido.



Depois de descrever o fluxo principal do caso de uso, podemos analisar o que poderia dar errado em cada um dos passos identificando os fluxos alternativos (exceções).

Uma exceção é um evento capaz de impedir o prosseguimento do caso de uso que não é tratado corretamente. A exceção não impede que o processo seja iniciado, mas impede sua conclusão.

No exemplo do sistema de biblioteca, o fato de o cliente não possuir cadastro válido não o impede de solicitar empréstimo de livro. Porém, a conclusão do caso de uso vai depender se ele tem uma identificação. Então, o processo foi iniciado, mas não pôde ser concluído. Cada exceção corresponde a uma ramificação do fluxo principal, e deve ser tratada para que o processo possa começar e terminar corretamente.

Ex.: Caso de uso: emprestar livro

Caso de uso: emprestar livro	
Fluxo principal	Exceções
1. O cliente seleciona os livros na biblioteca que deseja emprestar. 2. O cliente informa ao funcionário seu nome e os livros que serão emprestados. 3. O funcionário informa o nome do cliente no sistema e inicia o processo de "emprestar livro". 4. O funcionário registra cada livro no sistema. 5. O funcionário finaliza o processo de "emprestar livro", informando a data de devolução. 6. O Cliente deixa a biblioteca com os livros.	3.1. O cliente não possui cadastro. 3.1.1 O cliente deve informar seus dados para cadastro. 3.1.2 O funcionário registra o cadastro. 3.1.3 Retorna ao fluxo principal 3.

Figura 11. Exemplo de fluxo principal e exceções para um caso de uso.



## 4.4 Componentes de um modelo de caso de uso

- Ator – papel que estimula/solicita ações/ eventos do sistema e recebe reações. Pode participar de vários casos de uso.
- Caso de uso – documento narrativo que descreve a sequência de eventos feitos por um ator no uso do sistema.
- Sistema – sistema a ser modelado.

### Relacionamento

O relacionamento representa uma interação existente entre o sistema e seus componentes com o ambiente externo. Ocorre de quatro maneiras diferentes:

- **Comunicação:** representa troca de mensagens.
- **Inclusão:** representa um caso que, ao ser executado, deve executar um outro caso de uso também.
- **Extensão:** representa um caso que, ao ser executado, pode ou não executar um ou mais casos de uso também.
- **Generalização:** representa um caso de uso ou um ator que herda características de outro caso de uso ou ator.

## 4.5 Documentação do modelo de caso de uso

O modelo de casos de uso é composto por:

- **diagrama** de casos de uso;
- documentação dos **atores**; e
- documentação dos **casos de uso**.

### 4.5.1 Diagrama de casos de uso

O diagrama de caso de uso serve para dar suporte à parte escrita do mode-



lo. Basicamente, um diagrama de caso de uso apresenta dois elementos: os atores e os casos de uso.

Dependendo do grau de complexidade do sistema, o diagrama pode ser apresentado:

- Exibindo um caso de uso de seus relacionamentos.
- Exibindo todos os casos de uso para um ator.
- Exibindo todos os casos de uso a serem implementados em um ciclo de desenvolvimento.

Ex.: diagrama que representa a relação entre ator e o caso de uso.

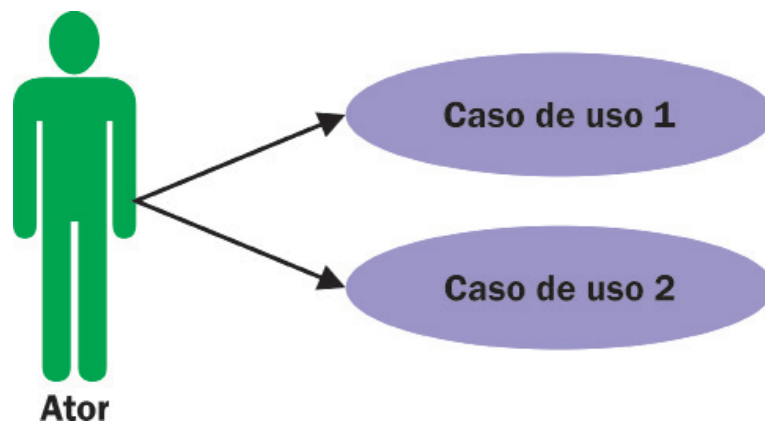


Figura 12. Representação básica da UML.

A utilização de setas nas relações entre atores e casos de uso indica qual ator ativa o caso de uso.



Ex.: relacionamentos em casos de uso.

Existem três tipos de relações entre casos de uso (inclusão, extensão e generalização):

- **Relacionamento de inclusão**

Um relacionamento de inclusão é uma relação através da qual um caso de uso insere em seu interior um outro caso de uso, que isoladamente não teria sentido. Ele é um integrante de um caso de uso maior e completo.

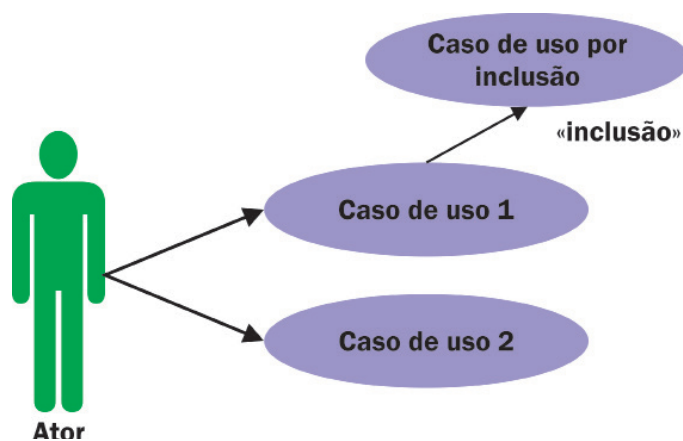


Figura 13. Representação da UML de relacionamento de inclusão em um caso de uso.

- **Relacionamento de extensão**

Um relacionamento de extensão é uma relação entre dois casos de uso na qual um caso de uso maior é estendido por um caso de uso menor. A extensão inclui algum serviço ou alguma funcionalidade que ocorrem em dada situação.

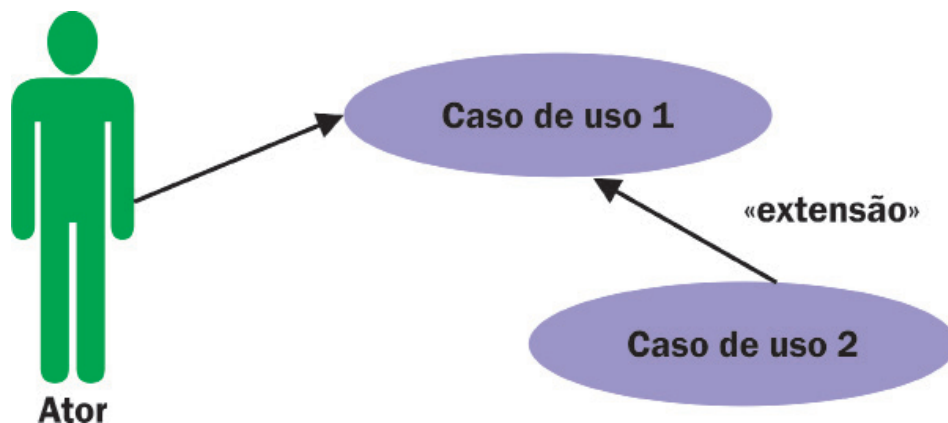


Figura 14. Representação da UML de relacionamento de extensão em um caso de uso.

- **Relacionamento de generalização**

Um relacionamento de generalização é uma relação entre um caso de uso mais geral e um caso de uso mais específico. Podemos dizer que o caso de uso mais geral é uma generalização (abstração) do caso de uso mais específico. A representação gráfica da UML para relacionamento de generalização é uma reta que une os dois casos de uso. Essa reta possui um triângulo na extremidade apontando para o caso de uso mais geral.

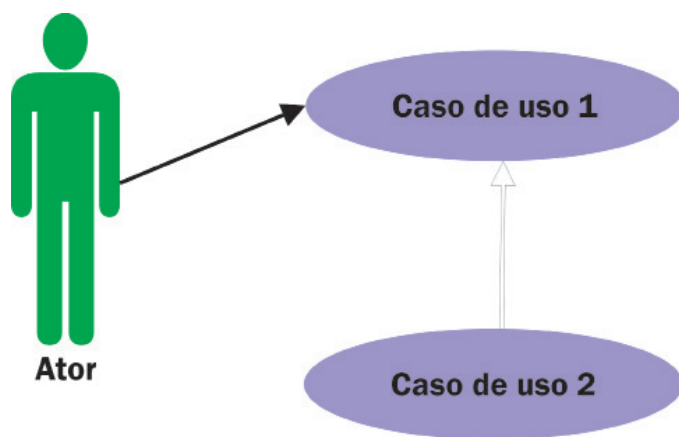


Figura 15. Representação da UML de relacionamento de generalização entre casos de uso.

### 4.5.2 Documentação dos atores

Deve conter uma breve descrição para cada ator.

### 4.5.3 Documentação dos casos de uso

Um caso de uso representa como os atores interagem com o sistema e como o sistema responde. É iniciado por um ator ou por outro caso de uso.

Para cada caso de uso deverá existir:

- Um nome composto por um verbo seguido por um substantivo.
- Um identificador que permita identificá-lo de forma única.
- Uma pequena descrição do caso de uso.
- O nome do ator primário. Isto é, aquele que inicia o caso de uso.
- Os nomes dos atores secundários. Isto é, os demais atores participantes do caso de uso, se existirem.
- Fluxo que representa a sequência de passos principais do caso de uso, o qual é chamado de fluxo principal.
- Fluxo principal com alternativas independentes, o qual descreve a sequência de eventos alternativo(s) para um caso de uso.
- Fluxo principal com alternativas exclusivas entre si: descreve uma sequência de eventos alternativo(s) para um caso de uso, em que apenas um



deles pode ocorrer por vez.

- Fluxo de exceção: descreve a sequência de eventos quando algo inesperado acontece na interação entre ator e caso de uso.

Vou exemplificar a modelagem de um caso de uso: sistema de controle de biblioteca.

## Descrição do problema

O objetivo de um sistema para uma biblioteca é automatizar o controle de empréstimo e devolução de livros.

O sistema deve manter o controle dos cadastros de usuários da biblioteca e o acervo de livros, os empréstimos e devolução de livros pelos usuários da biblioteca.

O usuário/cliente da biblioteca poderá acessar o sistema pela internet e fazer a reserva e o cancelamento da reserva de livros.

Se o empréstimo do livro não for devolvido dentro do prazo estipulado, será cobrada uma multa.

Para identificarmos os atores que irão interagir com o sistema, podemos responder algumas perguntas, como:

- Qual o ator que irá operar o sistema? R: operador do sistema.
- Qual o ator que fará a manutenção do sistema? R: operador do sistema.
- Qual o ator que receberá as informações do sistema? R: operador do sistema, usuário da biblioteca.
- Qual o ator que fornecerá as informações para o sistema? R: operador do sistema, usuário da biblioteca.

Portanto, os atores identificados para interagir com o sistema são:

- Operador do sistema (bibliotecário) – responsável pelas operações de controle de empréstimo e devolução de livros, controle do cadastro de

usuários da biblioteca e controle do acervo dos livros.

- Usuário da biblioteca (cliente) – responsável por fornecer as informações cadastrais, solicitar reserva de livro e cancelar reserva de livro.

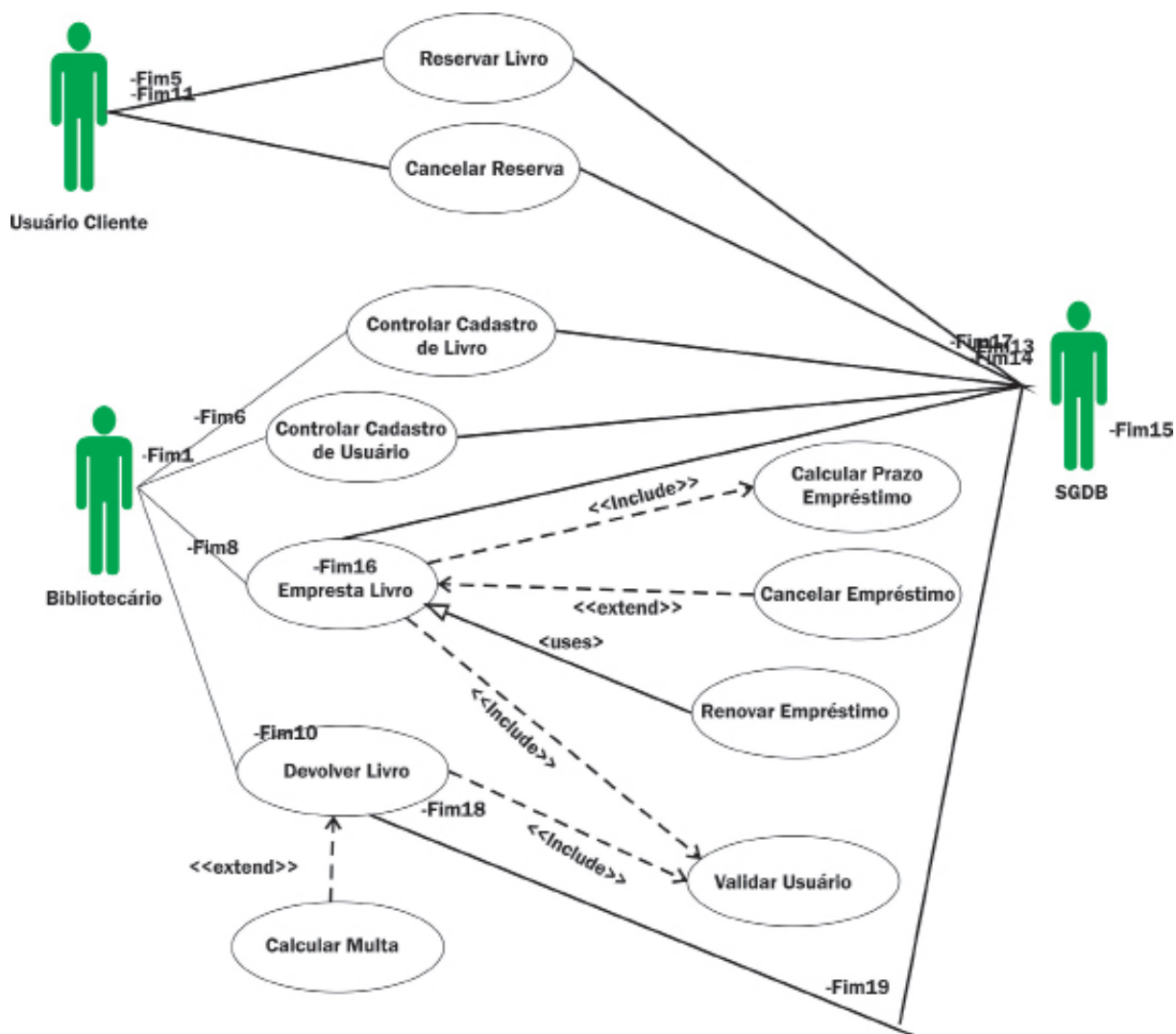


Figura 16. Representação gráfica do caso de uso biblioteca.



Vou agora descrever cada caso de uso.

- O caso de uso deve receber um nome único.
- Os atores devem ser declarados.
- A declaração do caso de uso deve descrever o que ele realmente faz.

<b>Caso de uso: controlar cadastro de livro</b>
Objetivo: inserir, consultar, alterar, excluir dados de livro no sistema ou cancelar.
Ator principal: usuário (bibliotecário).
Ator secundário: SGDB (Sistema Gerenciador de Banco de Dados, onde ficarão armazenadas as informações).
Precondição: usuário (bibliotecário) ser identificado no sistema.
<b>Fluxo Principal:</b>
1. O sistema exibe a tela inicial para incluir as informações de cadastro do novo livro.
2. O sistema valida as informações inseridas pelo usuário (bibliotecário). Emite mensagem de "Dados Válidos".
3. Caso de uso encerrado.
<b>Fluxo Alternativo:</b>
<b>FA1. Consultar cadastro de livros</b>
O sistema exibe a tela de consultar para informar o nome do livro.
O sistema exibe a lista de consulta de acordo com o filtro solicitado pelo usuário e detalha a informação.
O sistema solicita finalização da consulta (OK?).
O sistema retorna ao passo 3 do fluxo principal.
<b>FA2. Alterar cadastro de livros</b>
O sistema exibe a tela de alterar para informar o nome do livro.
O sistema mostra as informações sobre o livro especificado no filtro.
O usuário altera as informações desejadas.
O sistema valida as informações alteradas pelo usuário.
O sistema solicita finalização da alteração (OK?).
O sistema retorna ao passo 3 do fluxo principal.
<b>FA3. Excluir livro do cadastro</b>
O sistema exibe a tela de exclusão para informar o nome do livro.
O sistema mostra as informações sobre o livro especificado no filtro.
O sistema solicita confirmação de exclusão dos dados no cadastro de livros.
O usuário confirma a solicitação de exclusão.
O sistema valida opção pelo usuário.
O sistema retorna ao passo 3 do fluxo principal.
<b>FA4. Cancelar operação</b>
O sistema exibe a opção de cancelar. O usuário seleciona operação cancelar.
O sistema retorna ao passo 3 do fluxo principal.





<b>Caso de uso: controlar cadastro de usuário/cliente</b>
Objetivo: inserir, consultar, alterar, excluir dados de cliente no sistema ou cancelar.
Ator principal: usuário (bibliotecário).
Ator secundário: SGDB (Sistema Gerenciador de Banco de Dados, onde ficarão armazenadas as informações).
Precondição: usuário (bibliotecário) ser identificado no sistema.
<b>Fluxo Principal:</b>
1. O sistema exibe a tela inicial para incluir as informações cadastrais do novo cliente.
2. O sistema valida as informações inseridas pelo usuário (bibliotecário). Emite mensagem de "Dados Válidos".
3. Caso de uso encerrado.
<b>Fluxo Alternativo:</b>
<b>FA1. Consultar cadastro de cliente</b>
O sistema exibe a tela de consulta para informar a identificação do cliente.
O sistema exibe a lista de consulta de acordo com o filtro solicitado pelo usuário e detalha a informação.
O sistema solicita finalização da consulta (OK?).
O sistema retorna ao passo 3 do fluxo principal.
<b>FA2. Alterar cadastro de cliente</b>
O sistema exibe a tela de alterar para informar a identificação do cliente.
O sistema mostra as informações sobre o cliente especificado no filtro.
O usuário (bibliotecário) altera as informações desejadas.
O sistema valida as informações alteradas pelo usuário (bibliotecário).
O sistema retorna ao passo 3 do fluxo principal.
<b>FA3. Excluir cliente do cadastro</b>
Opção excluir: o sistema exibe a tela de exclusão para informar a identificação do cliente.
O sistema mostra as informações sobre o cliente especificado no filtro.
O sistema solicita confirmação de exclusão dos dados no cadastro de cliente.
O usuário (bibliotecário) confirma a solicitação de exclusão.
O sistema valida opção pelo usuário (bibliotecário).
O sistema retorna ao passo 3 do fluxo principal.
<b>FA4. Cancelar operação</b>
O sistema exibe opção de cancelar. O usuário (bibliotecário) seleciona operação cancelar.
O sistema retorna ao passo 3 do fluxo principal.



<b>Caso de uso: emprestar livro</b>
Objetivo: possibilitar que o cliente possa emprestar um livro.
Ator principal: usuário (bibliotecário).
Ator secundário: SGDB (Sistema Gerenciador de Banco de Dados, onde ficarão armazenadas as informações).
Precondição: usuário (bibliotecário) ser identificado no sistema.
<b>Fluxo Principal:</b>
1. O sistema exibe a tela inicial e usuário (bibliotecário) informa a identificação do cliente e do livro a ser emprestado.
2. O Sistema valida o cliente mostrando informações de cadastro.
3. O sistema verifica se existe uma reserva do livro para o cliente identificado.
4. O sistema verifica a disponibilidade de exemplar para o livro desejado.
5. O sistema registra o empréstimo e calcula prazo de devolução do livro.
6. O sistema emite recibo contendo as informações do empréstimo.
7. Caso de uso encerrado.
<b>Fluxo Alternativo:</b>
<b>FA1. Reserva de livro para outro cliente</b>
O sistema emite mensagem de "Livro Reservado" para o usuário (bibliotecário).
O sistema retorna ao passo 7 do fluxo principal.
<b>FA2. Reserva do livro pelo cliente</b>
O sistema dá baixa na reserva do livro pelo cliente.
O sistema retorna ao passo 4 do fluxo principal.
<b>FA3. Renovar empréstimo</b>
O sistema verifica o prazo de entrega e emite mensagem de "Renovar Empréstimo" para o usuário (bibliotecário).
O sistema retorna ao passo 5 do fluxo principal.
<b>FA4. Cancelar empréstimo</b>
O sistema exibe a opção cancelar. O usuário (bibliotecário) seleciona operação cancelar empréstimo.
O sistema retorna ao passo 7 do fluxo principal.

<b>Caso de uso: devolver livro</b>
Objetivo: possibilitar que o cliente devolva um livro emprestado.
Ator principal: usuário (bibliotecário).
Ator secundário: SGDB (Sistema Gerenciador de Banco de Dados, onde ficarão armazenadas as informações).
Precondição: usuário (bibliotecário) ser identificado no sistema.
<b>Fluxo Principal:</b>
1. O sistema exibe a tela inicial e usuário (bibliotecário) informa a identificação do cliente e do livro a ser devolvido.
2. O Sistema valida o cliente mostrando informações de cadastro.
3. O sistema identifica registro de empréstimo.
4. O usuário (bibliotecário) confirma devolução do livro.
5. O sistema registra a devolução e verifica se houve atraso no prazo de devolução do livro.
6. O sistema emite mensagem de "Devolução no Prazo".
7. Caso de uso encerrado.
<b>Fluxo Alternativo:</b>
<b>FA1. Devolução em atraso.</b>
O sistema calcula o número de dias em atraso e o valor da multa.
O sistema emite recibo de pagamento no valor da multa.
O usuário (bibliotecário) registra o pagamento da multa no sistema.
O sistema retorna ao passo 7 do fluxo principal.

Vamos determinar algumas regras básicas que definirão a política de controle do sistema de biblioteca.

Regras de negócio
RN01. No sistema de controle de biblioteca, um cliente não pode ter mais de uma reserva de livro em seu nome.
RN02. Depois de feita a reserva de um livro, o cliente tem prazo de 48 horas para a retirada do livro na biblioteca.
RN03. Um cliente só pode ter dois livros em seu poder ao mesmo tempo.
RN04. O valor da multa a ser aplicada é de R\$ 2,00 por dia de atraso.

## Resumo

A UML é uma linguagem de modelagem padronizada que permite visualizar e entender o relacionamento entre os objetos definidos.

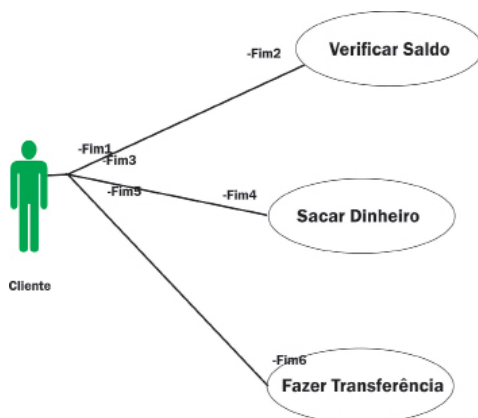
A identificação dos casos de uso de um sistema é, em muitas vezes, um processo intuitivo, em que a utilização da UML permite especificar as funcionalidades do sistema e a relação existente entre os atores através do diagrama. E através do documento de caso de uso, no qual é declarado o fluxo principal e os fluxos alternativos, que será descrito como cada caso de uso deve funcionar.

A declaração das regras de negócio vai definir a política adotada para o tratamento das particularidades das funcionalidades do sistema.

## Atividades de aprendizagem

1. Descrição do problema: cliente dirige-se ao caixa eletrônico do banco para retirar dinheiro de sua conta corrente e fazer uma transferência bancária.

- Especificado o diagrama de caso de uso abaixo, gerar o documento do caso de uso para o diagrama proposto, descrevendo o fluxo principal e alternativo e as regras de negócio.





---

---

---

---

---

---

---

---

---

---

**2.** Escrever a declaração dos casos de uso “Reservar Livro” e “Cancelar Reserva”, conforme diagrama (Figura 16) do exemplo proposto nesta quarta aula: sistema de controle de biblioteca.

---

---

---

---

---

---

---

---

---

---

**3.** Descrição do problema: a escola de sua cidade disponibilizou via internet, no site oficial, o preenchimento de cadastro para reserva de matrícula para o ano letivo. A escola quer ter uma estatística do quantitativo de novas matrículas para a criação de infraestrutura para o próximo período. Cada candidato irá informar seus dados cadastrais e o período desejado. O candidato poderá consultar o número de vagas disponíveis para cada período letivo. Através de sua inscrição, o candidato irá acompanhar a data prevista para realizar a sua matrícula na escola.

Identificar os casos de uso do problema descrito.

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

Rede e-Tec Brasil



# Aula 5. Modelagem de classe de objeto

## Objetivo:

- construir um diagrama de classes através da linguagem UML.

Caro(a) estudante,

Bem-vindo(a) à nossa quinta aula! Estamos entrando na segunda metade da nossa disciplina. Vamos, nesta aula, partindo do modelo de casos de uso, construir através da linguagem UML o diagrama de classes identificando as classes de objeto.

E então, o que é o diagrama de classe da UML?

## 5.1 Introdução

O diagrama de classe da UML é uma ferramenta utilizada para a construção do modelo de classes de objetos para a documentação de software.



Identificamos o levantamento das classes do sistema pelos casos de uso definidos no projeto do sistema. Uma classe descreve um grupo de objetos com propriedades (atributos), comportamento (operações) e os relacionamentos, e pode ser empregada em mais de um caso de uso.

O modelo de caso de uso representa a visão do sistema do ponto de vista externo, no qual os atores visualizam as solicitações de informação, requisitos de cálculo etc. Porém, os objetos precisam trocar informações, ter uma interação uns com os outros para produzir os resultados que o sistema requer.

Como vimos na segunda aula deste caderno, a representação gráfica de uma classe é composta de 3 (três) partes: o nome da classe, declaração dos atributos e a declaração das operações.



NomeClasse
Atributos
Operações ( )

Figura 17. Representação gráfica de uma classe.

## 5.2 Identificação de classes

O primeiro passo para identificar as classes que vão fazer parte da modelagem do sistema é fazer uma leitura do levantamento de requisitos para determinar os substantivos que possam ser classificados como os possíveis candidatos as classes. Nessa leitura serão verificados quais os eventos que fazem parte do requisito, quem (clientes, alunos etc.) interage direta ou indiretamente com o sistema, locais que fazem parte do problema, como departamentos, setores, projetos que são informações relevantes para a modelagem do sistema. E em uma classe precisam ser identificados pelo menos dois ou mais atributos.

## 5.3 Identificação de atributos

Os atributos representam as propriedades de uma entidade ou de um relacionamento que está associado a um conjunto de valores válidos para o atributo.

Graficamente é representado por uma elipse.

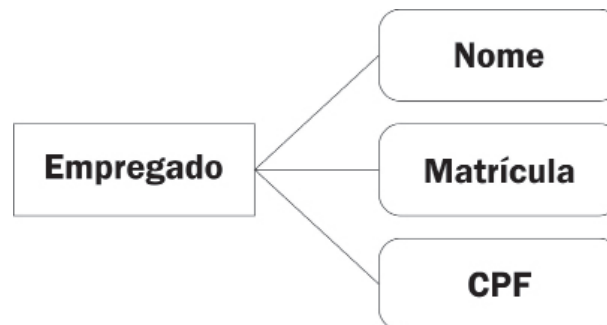


Figura 18. Representação gráfica de uma entidade e seus atributos.

Os relacionamentos também podem ter atributos específicos que identificam relações.



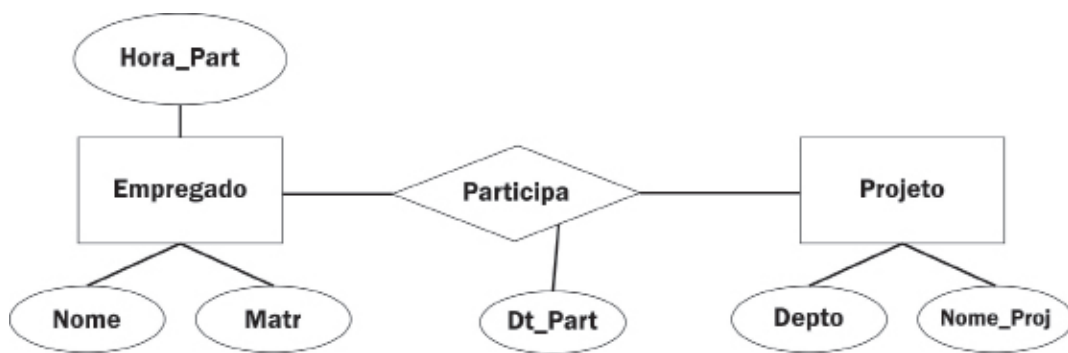


Figura 19. Representação gráfica de atributos de relacionamento.

## 5.4 Identificação de relacionamentos

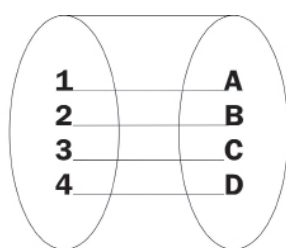
O que é relacionamento?

Relacionamento é a associação entre os objetos durante a execução do sistema. Cada associação tem uma cardinalidade que indica quantos objetos podem participar de um relacionamento.

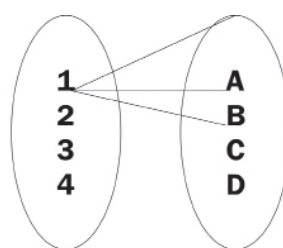


Definimos o relacionamento entre as entidades representado em um diagrama entidade/relacionamento através de uma linha unindo as entidades com um losango ao meio, com o nome do relacionamento (um verbo).

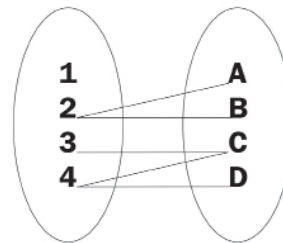
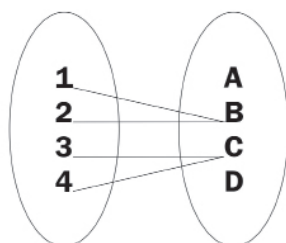
- Vamos ver uma das características dos relacionamentos, que é a cardinalidade. A cardinalidade representa o número de vezes que cada elemento de uma entidade pode participar da relação.



Relacionamento um-para-um (1:1)



Relacionamento um-para-muitos (1:N)





Vamos exemplificar para entender melhor.

Ex1.: Cardinalidade (N:N)



Podemos dizer que cada entidade de pacientes se relaciona por meio de consultar com várias entidades de médicos. Da mesma forma que uma entidade de médico se relaciona por meio de consultar com várias entidades de pacientes.

Consideramos nesse exemplo que muitos médicos podem consultar muitos pacientes e que muitos pacientes podem se consultar com muitos médicos.

Ex2.: Cardinalidade (1:1)



A entidade empregado se relaciona por meio do relacionamento coordenar com a entidade departamento.



Podemos dizer que um único empregado exerce a função de coordenar um único departamento.

Ex3.: Cardinalidade (1:N)





No exemplo acima, o 1 anotado na aresta que liga alocar e departamento, indica que cada entidade de departamento pode estar relacionada por meio de alocar com no máximo uma entidade de departamento. E o N anotado na aresta que liga alocar e empregado, indica que cada entidade de departamento pode estar relacionada por meio de alocar com um número qualquer de entidades de empregado.

Podemos dizer que um departamento pode alocar vários empregados, ou vários empregados são alocados por um departamento.



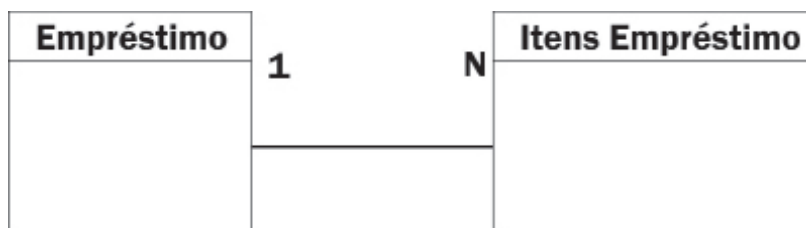
## 5.5 Relacionamento entre classes

Tipos de relacionamento entre classes e objetos:

- Relacionamento por associação
- Relacionamento por agregação
- Relacionamento por composição
- Relacionamento por generalização (herança)

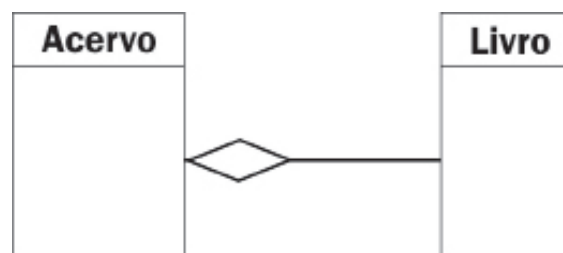
### 5.5.1 Relacionamento por associação

O relacionamento por associação representa as ligações entre os objetos de duas classes.



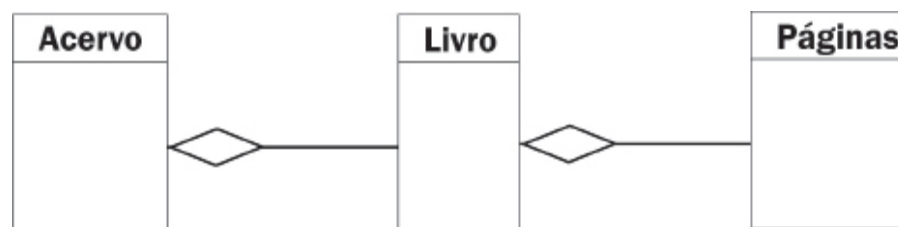
### 5.5.2 Relacionamento por agregação

O relacionamento por agregação é uma associação na qual o elemento associado é parte do elemento principal. É representado por uma haste com o símbolo de um diamante junto à classe agregadora.

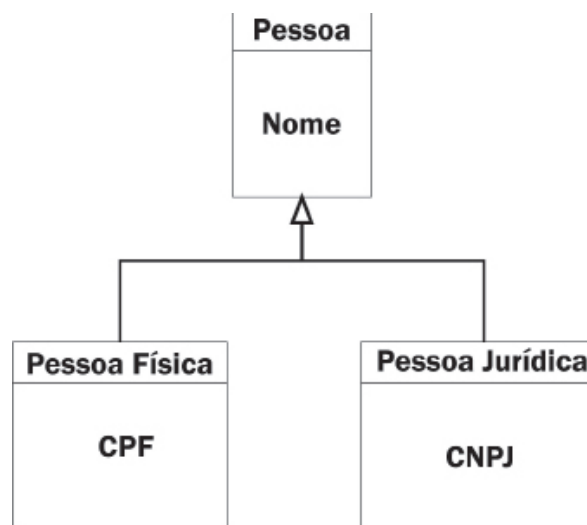


### 5.5.3 Relacionamento por composição

O relacionamento por composição é um tipo de agregação em que a parte indicada deve necessariamente existir. Não podemos pensar no objeto da classe principal sem os objetos que o compõem.



### 5.5.4 Relacionamento por generalização (herança)



Definimos que as classes `pessoa_física` e `pessoa_jurídica` herdam as características da classe `pessoa`.

## 5.6 Diagrama de classe

Vamos considerar o sistema de controle de biblioteca, definido na quarta aula para identificarmos as classes do projeto. Nesta fase de análise, e co-



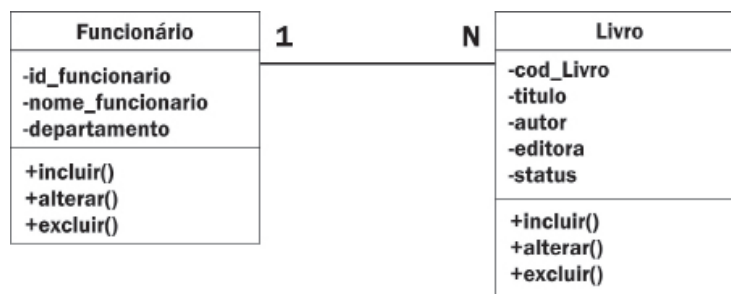
nhecendo os casos de uso do sistema, podemos desenhar o diagrama de classe que irá representar o agrupamento dos objetos e seus relacionamentos.

Consideremos:

- **Casos de uso principal:** reservar livro, cancelar livro, controlar cadastro de livro, controlar cadastrar usuários, emprestar livro, devolver livro.
- **Objetos:** livro, funcionário, cliente, empréstimo, devolução, reserva, pagamento, prazos.
  - o que o sistema vai controlar?
  - como o sistema vai controlar?
  - quem vai utilizar o sistema?
- **Classes:** livro, funcionário, cliente, empréstimo, devolução, reserva.
- **Operações:** incluir, alterar, excluir, consultar.

Vou exemplificar um caso de uso para melhor entender o diagrama de classes

Caso de uso 1: controlar cadastro de livro

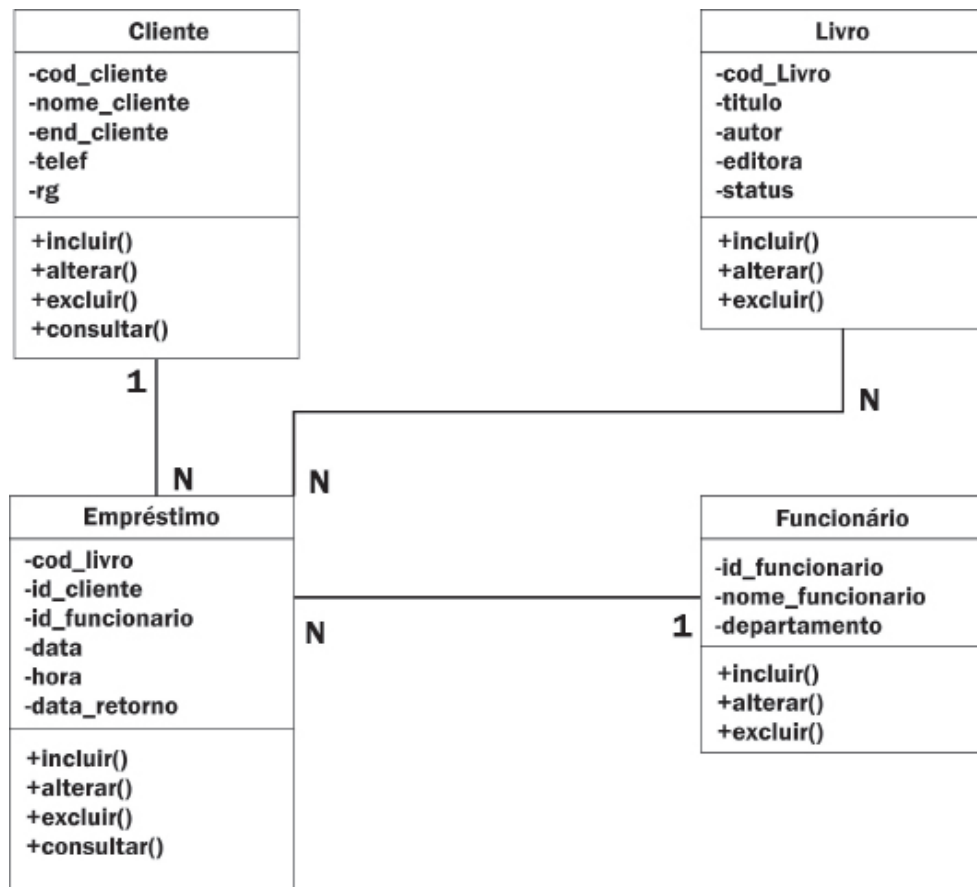


Podemos dizer que um funcionário opera as funções de controle de vários livros.





## Caso de uso 2: controlar empréstimo de livro



Podemos dizer que um funcionário controla um ou mais de um empréstimo de livro solicitado por um cliente.

## Resumo

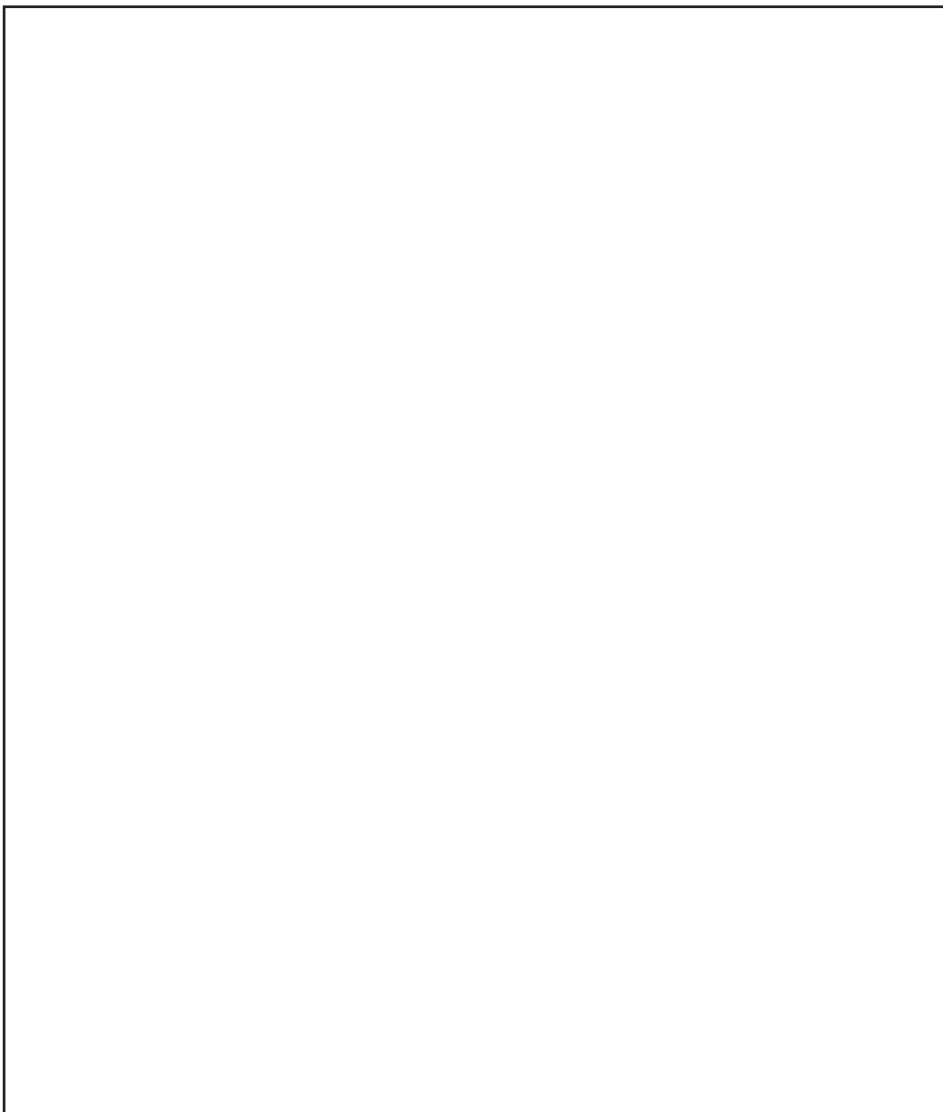
Nesta aula você observou que o diagrama de classe da UML é uma ferramenta utilizada para a construção do modelo de classes de objetos para a documentação de software. Você teve a oportunidade de aprender que o primeiro passo para identificar as classes que vão fazer parte da modelagem do sistema é fazer uma leitura do levantamento de requisitos para determinar os substantivos que possam ser classificados como os possíveis candidatos as classes. Por fim, você viu que relacionamento é uma associação entre os objetos durante a execução do sistema e que cada associação tem uma cardinalidade que indica quantos objetos podem participar de um relacionamento.



## Atividade de aprendizagem



1. Escrever o diagrama de classe para todos os casos de uso do sistema de controle de uma biblioteca, conforme o modelo estudado nesta aula.



Como vimos nesta aula, as classes e objetos são elementos construtivos de um sistema.

A notação UML facilita o entendimento e a definição dos relacionamentos existentes entre as classes e objetos para a realização dos casos de uso. E então, vamos em frente?





# Aula 6. Diagrama de interação

## Objetivos:

- construir um diagrama de sequência através da linguagem UML; e
- reconhecer os componentes do diagrama de sequências.

Prezado(a) estudante,

Bem-vindo(a) à nossa sexta aula! Agora você terá a oportunidade de aprender como se constrói, através da linguagem UML, o diagrama de sequência, que é de interação e descreve a sequência de mensagens enviadas e recebidas pelos objetos de um caso de uso. Podemos seguir?

## 6.1 Introdução

O diagrama de sequência da UML descreve a lógica de um cenário específico de como um caso de uso é implementado e como as classes trocam mensagens na realização do caso de uso na sequência em que acontecem os eventos.

- Um diagrama de sequência é um diagrama de objetos que contém um conjunto de objetos de diferentes classes.
- Os diagramas de sequência têm este nome porque descrevem ao longo de uma linha de tempo a sequência de comunicações entre objetos.
- Como podem existir muitos processos em um sistema, é favorável a construção dos diagramas de sequência por caso de uso.

As classes e os atores são representados na parte superior do diagrama de sequência. As setas que interagem uma classe com a outra representam os envios de mensagens, e as linhas pontilhadas que saem das classes posicio-



nam as mensagens no diagrama de sequência.

Vamos ver quais são os componentes do diagrama de sequência.

## 6.2 Componentes do diagrama de sequência

Componentes:

O diagrama de sequência possui dois eixos:

- o eixo horizontal mostra um conjunto de objetos; e
- o eixo vertical mostra o tempo.

Relacionados a estes eixos estão:

- classes e objetos;
- linhas da vida;
- mensagens; e
- barras de ativação.

### Classes e objetos

- São representados por um retângulo possuindo o nome do objeto e o nome da classe separado por dois pontos (":").

### Representação gráfica

**Objeto1:classe1**

**Objeto2:classe2**

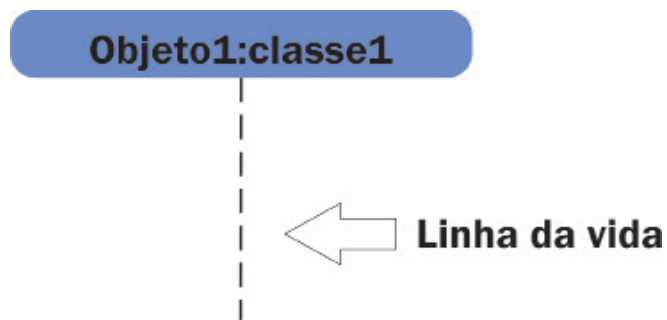
### Linhas da vida

- Um objeto possui uma linha pontilhada vertical, chamada de **linha de**



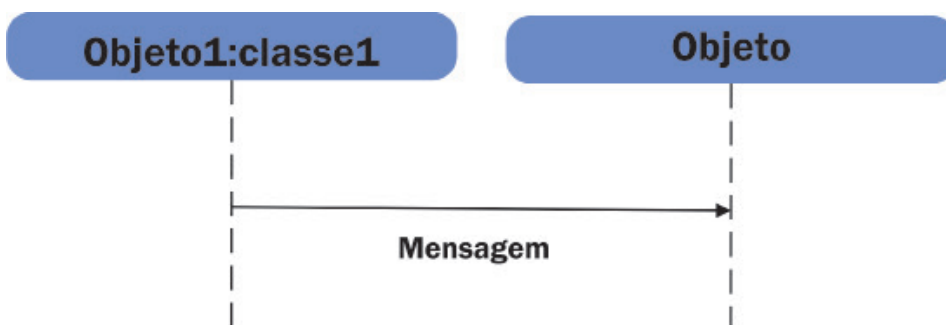
**vida** do objeto, que representa sua **existência**.

- Quando um objeto é excluído, a sua linha de vida é interrompida e um "X" é usado para indicar a destruição.



### Mensagens

- Uma mensagem é uma troca de comunicação entre objetos (emissor e receptor) e espera uma resposta (ação ou atividade).

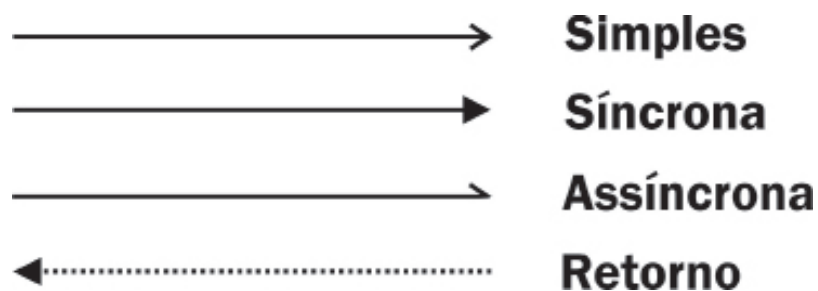


Existem vários tipos de mensagens. Vamos ver quais são eles?

#### 6.2.1 Tipos de mensagens

As mensagens podem ser:

- simples;
- síncrona;
- assíncrona;
- retorno; e
- reflexiva.



Fonte: Bezerra (2011)

- Mensagem simples: mostra como a comunicação é passada de um objeto para outro.
- Mensagem síncrona: indica que o objeto remetente espera que o objeto receptor processe a mensagem antes de recomeçar o seu processamento.
- Mensagem assíncrona: quando o objeto envia a mensagem e continua seu processamento, não espera a resposta para prosseguir. Dois métodos são executados simultaneamente.
- Mensagem reflexiva: um objeto pode enviar mensagem para si próprio. O objeto pode requisitar a execução de uma operação definida na sua própria classe.

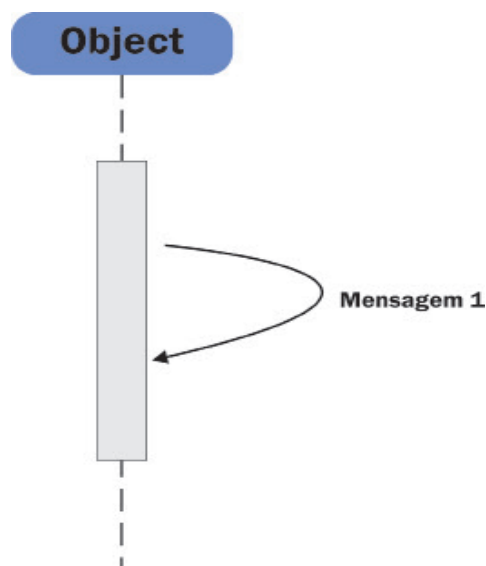


Figura 20. Representação gráfica de mensagem reflexiva no diagrama de sequência.



Segundo Vergilio, uma mensagem implica na existência de uma operação no objeto receptor. A resposta do objeto receptor ao recebimento de uma mensagem é a execução da operação correspondente. Disponível em: <<http://www.inf.ufpr.br/silvia/ESNovo/UML/ppt/DiagIntera.ppt>> Acesso em 27 nov.2013



## 6.3 Diagrama de sequência

Exemplificarei o diagrama de sequência para o caso de uso: emprestar livro.

A figura 21 demonstra o sistema interagindo com a interface gráfica e o banco de dados do sistema, que efetuará as atualizações e busca de informação.

O sistema disponibiliza a interface do caso de uso efetuar empréstimo, no qual através da solicitação de empréstimo o usuário começa a interagir com o sistema informando o nome do livro desejado. O sistema recupera essa informação no banco de dados, e exibe na tela as informações do livro. O sistema solicita, então, confirmação do empréstimo do livro. O usuário confirma solicitação de empréstimo e o sistema atualiza e grava no banco de dados as informações referentes ao empréstimo. O sistema retorna para o usuário a informação da data de devolução do livro e encerra o processo.

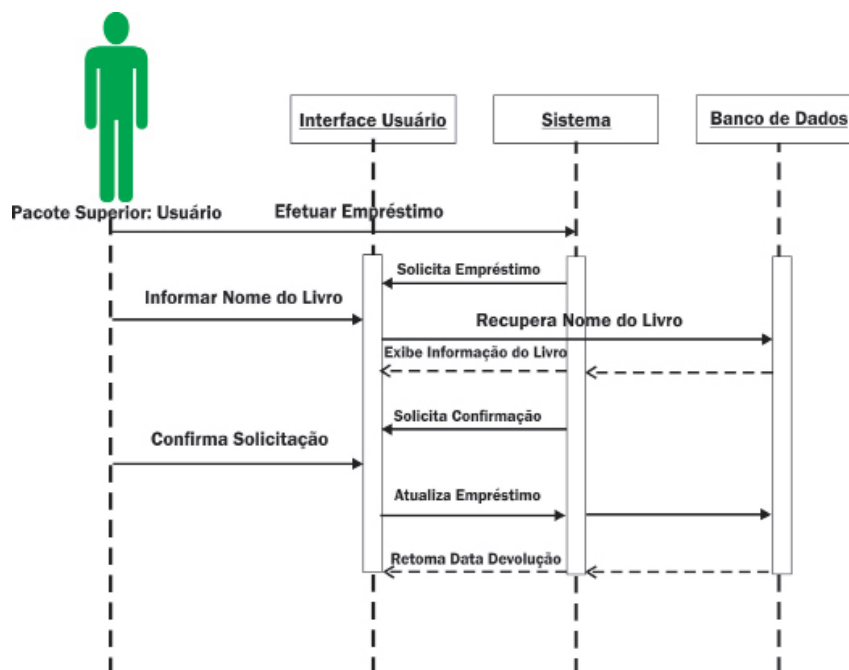


Figura 21. Diagrama de sequência do caso de uso: efetuar empréstimo.



Ficou difícil de entender? Vamos lá!

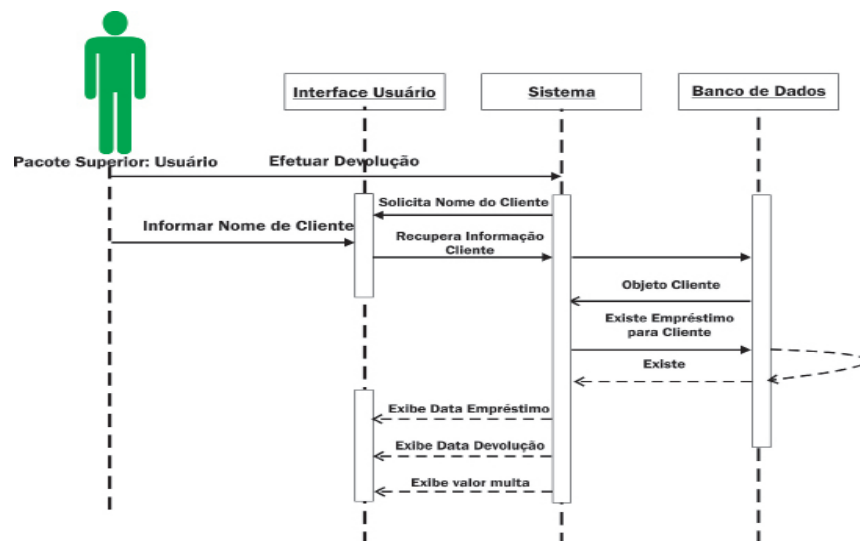


O diagrama de sequência é a representação passo a passo da troca de informação (mensagem) entre os objetos de um sistema. Ele retrata um conjunto de cenários que faz a interação entre o usuário (ator) e a interface gráfica, que relaciona o ator e o sistema (algoritmo), sendo este responsável por processar e atualizar a informação que está num repositório de dados (banco de dados).

Mostrarei outro caso de uso: efetuar devolução.

A figura exposta anteriormente mostra a interação do usuário (bibliotecário) com a interface gráfica para saber se existe empréstimo, após validar o nome do cliente no banco de dados. Se houver devolução de livro ligado ao objeto cliente, o sistema envia mensagem, interagindo com a interface, mostrando as informações do empréstimo, a data de devolução e a multa, se a devolução estiver fora do prazo.

Exemplificarei agora o diagrama de sequência para o caso de uso: validar usuário.



**Figura 22. Exemplo do diagrama de sequência para o caso de uso: efetuar devolução.**

A figura da próxima página mostra a funcionalidade do sistema solicitando ao usuário a sua identificação (matrícula/senha) através da interface gráfica e recuperando e validando o usuário na base de dados.

Ilustrarei o diagrama de sequência para o caso de uso: renovar empréstimo.

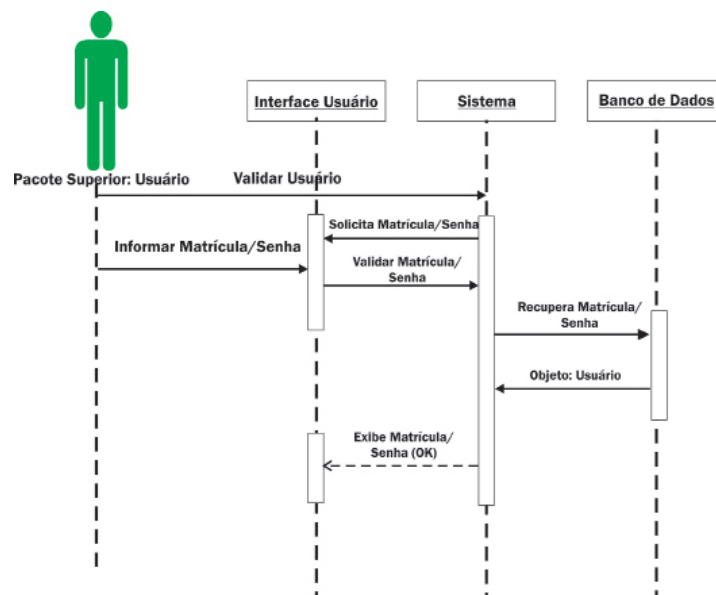


Figura 23. Exemplo do diagrama de sequência para o caso de uso: validar usuário.

A figura a seguir apresenta o diagrama de sequência no qual o sistema solicita o nome do livro a ser renovado no empréstimo. O sistema resgata essa informação na base de dados, recupera o objeto livro e verifica se está disponível para renovação, e por fim encaminha mensagem de OK através da interface.

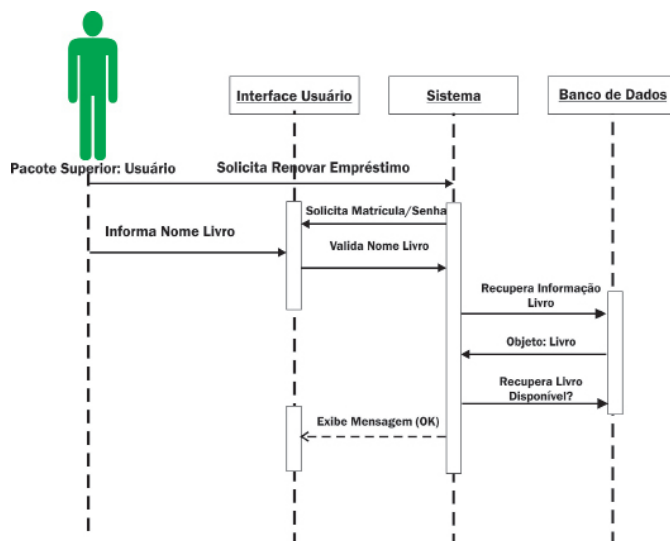


Figura 24. Exemplo do diagrama de sequência para o caso de uso: renovar empréstimo.



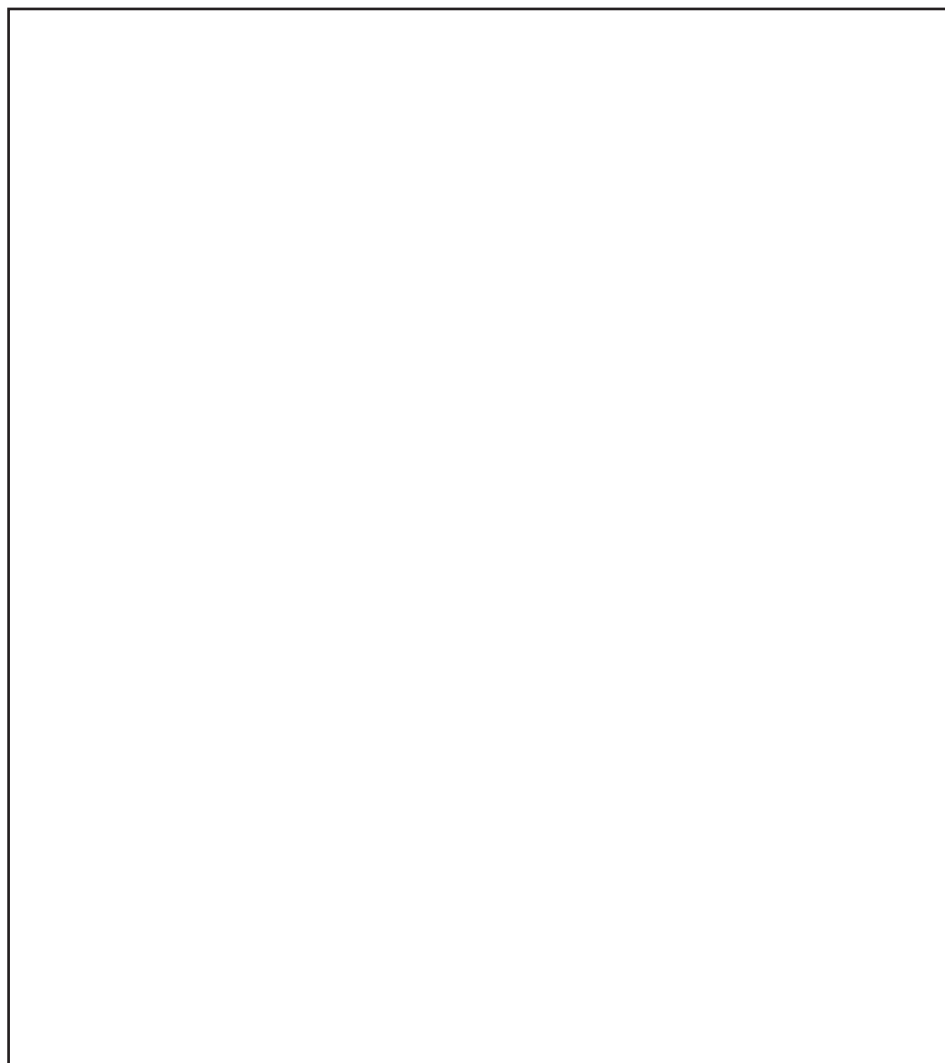
## Resumo

Você viu nesta aula que o diagrama de sequência da UML indica a cronologia de ocorrência das mensagens entre objetos para realizar um procedimento em um sistema. Observou ainda que ele tem este nome porque descreve através da linha de tempo a sequência de comunicação (troca de mensagem) entre objetos. Ao final do capítulo você também teve a oportunidade de aprender que, para cada caso de uso, é construído um diagrama de sequência que representa um conjunto de cenários que faz a interação entre os atores, a interface e os objetos.

## Atividades de aprendizagem

1. De acordo com a descrição da atividade 1 da quarta aula deste caderno, escreva os diagramas de sequência para o problema proposto.

Sugiro utilizar uma ferramenta de design, como o Visio da Microsoft.







2. De acordo com a atividade 3 desenvolvida, também, na quarta aula deste caderno, escreva os diagramas de sequência para os casos de uso do problema proposto.



E assim terminamos o nosso sexto encontro. Só falta uma aula para terminarmos a nossa disciplina. Vamos prosseguir? Um grande abraço e até a nossa sétima aula.



# Aula 7. Desenvolvendo a análise de sistema

## Objetivos:

- sintetizar os principais passos a serem identificados no processo da análise de sistemas; e
- identificar na análise como o sistema vai utilizar os recursos de dados.

Caro(a) estudante,

Bem-vindo(a) à nossa sétima aula! Agora vamos construir o processo da análise de sistemas passo a passo, como um roteiro a seguir para a construção de um software.

## 7.1 Introdução

Antes de você projetar um novo sistema, é importante estudar o sistema que será desenvolvido ou substituído.

A análise de sistemas não é um estudo preliminar. Ela é um estudo em profundidade sobre necessidades de informação do usuário final e produz requisitos funcionais que são utilizados como base para o projeto de um novo sistema de informação.

Todo projeto de software é motivado por alguma necessidade de negócio: a necessidade de corrigir um defeito em uma aplicação existente; a necessidade de controlar o negócio; a necessidade de criar um novo produto, serviço ou sistema.

O sucesso de um sistema baseado em computadores pode ser medido de muitas formas, mas a satisfação do cliente encontra-se em primeiro lugar. Se você entender como os usuários finais querem interagir com um sistema, então você estará capacitado(a) a caracterizar os requisitos e a construir modelos de análise e projetos proveitosos.



Portanto, a modelagem de requisitos com UML começa com a criação de cenários na forma de casos de uso, diagrama de atividades e diagrama de classes.

Posso agora sintetizar os principais passos para a análise de sistemas porque você já conhece todos os conceitos de sistema, conceitos de classes e objetos, modelagem e o uso da linguagem UML.

## 7.2 Primeiro passo: entrevista com cliente

Como primeiro passo para o desenvolvimento de um software, após identificar e conhecer a oportunidade de negócio, o(a) analista deverá visitar seu cliente para captar os requisitos, conhecer as informações sobre o negócio a ser controlado, descobrir a real necessidade do cliente. Esses requisitos específicos de cada processo do sistema serão obtidos em uma entrevista com seu cliente.

O(a) analista deve seguir as normas gerais de entrevista. Não se esquecer de fazer todas as anotações necessárias, conseguir cópias de fichas, relatórios ou planilhas existentes utilizados nos processos manuais que serão informatizados.

## 7.3 Segundo passo: especificação de requisitos

Após a captação dos requisitos, eles devem ser transcritos em um modelo de formulário, no qual o(a) analista possa entender o que o cliente deseja, ou seja, o problema a ser resolvido.



Assunto verificado na terceira aula.

Essa organização facilita a especificação dos requisitos e a identificação dos atributos do sistema a ser desenvolvido.

## 7.4 Terceiro passo: diagrama de casos de uso

Após a especificação dos requisitos, você pode utilizar o diagrama de casos de uso para identificar os atores e seus relacionamentos do cenário estudado, conforme a quarta aula.



## 7.5 Quarto passo: diagrama de classes

A partir da identificação das classes de entidades, extraídas do problema estudado, que representam coisas que devem ser armazenadas em um banco de dados, podemos elaborar o diagrama de classes, que é fundamental para a implementação do software, conforme a nossa quinta aula.

## 7.6 Quinto passo: diagrama de sequência

A partir do diagrama de casos de uso, podemos fazer o diagrama de sequência/atividade, que destaca a interação do “ator” com as diferentes interfaces oferecidas pelo sistema, conforme a nossa sexta aula.

### Resumo

As tarefas de especificação de requisitos são conduzidas para estabelecer uma base para a construção do sistema. A especificação de requisitos ocorre durante as atividades de comunicação com o cliente e de modelagem do sistema a ser desenvolvido. Os interessados estabelecem os requisitos básicos do problema, definem restrições e tratam as principais características e funções que têm de estar presentes para que o sistema atenda seus objetivos. Cada requisito e o modelo de análise como um todo é validado em relação às necessidades do cliente para garantir que será construído o sistema correto.

### Atividade de aprendizagem

Como sugestão, exercite cada passo do processo da análise de sistema para o cenário sugerido.



Cenário: um hospital deseja informatizar os processos de agendamento de consulta e internação dos pacientes. O hospital tem várias atendentes responsáveis pelo preenchimento da ficha dos pacientes, que contém dados como o número de identificação, nome, endereço, cidade, telefone, e-mail, data de nascimento, nome da mãe. As atendentes também agendam os pacientes, anotando o dia, mês, ano, horário e o nome do médico. Atualmente o processo é feito manualmente com uma agenda para cada médico. Os médicos possuem especialidades diferentes, além de seus dados pessoais, como CRM, nome, endereço, telefone, e-mail.

---

---

---

---



E assim terminamos a nossa sétima e última aula. Espero que tenha aproveitado bem a nossa disciplina. Na sequência ainda tenho algumas palavras para você. Um grande abraço.





## Palavras Finais

Prezado(a) estudante, quero compartilhar com você esta vitória de ter chegado ao final desta disciplina. Sabemos que a disciplina de Análise de Sistemas é o trampolim para o ingresso na carreira de desenvolvimento de software. Então continue firme!

Espero que você tenha gostado do conteúdo e mergulhado no entendimento deste mundo virtual.

Faça contato conosco para expor as suas expectativas de aprendizagem e se elas foram atendidas no conteúdo deste caderno. O seu entendimento e aprendizado são muito importantes para mim e para todos que fazem parte da Rede e-Tec Brasil.

Mas quero que você saiba, caro(a) estudante, que isto é só o começo. Continue estudando, pesquisando, elaborando projetos. Conte com a ajuda de ferramentas como a internet, livros e este canal de aprendizagem da Rede e-Tec Brasil.

Parabéns e sucesso!



# Guia de Soluções

## Aula 1

Espera-se que o(a) estudante seja capaz de relatar os princípios fundamentais e os tipos de sistema de informação, bem como o papel fundamental do(a) analista no processo de desenvolvimento de software.

## Aula 2

2.

a) Classe aluno

b) Classe disciplina

3.

a) Atributo = CorCabelo

b) Atributo = PreferenciaSexual

4. Classes: Professor/Coordenador

Animais

Instrumento Musical

Hardware

5. Espera-se que o(a) estudante aplique as regras de herança estudada nesta aula.

6.

a) Livro, autor

b) Carro, piloto, autódromo

## Aula 3





1. Espera-se que o(a) estudante seja capaz de aplicar a lógica estudada nesta terceira aula. Ele(a) deve fazer o levantamento das informações baseadas nas necessidades de controle de uma escola, respondendo os questionamento a seguir. Como sugestão:

a) O sistema deverá fazer o controle de notas e frequências dos alunos da escola.

b) Espera-se que o(a) estudante descreva as operações necessárias para manter o controle de notas e frequências no sistema da escola.

c) Espera-se que o(a) estudante narre as dificuldades encontradas pelo cliente no controle de tais informações.

d) Descrever as possíveis soluções que você visualiza, tendo como base os conceitos estudados nesta aula.

e) Descrever os requisitos de informações que a escola precisa para controlar as notas e frequências.

f) Espera-se que o(a) estudante defina a lógica de requisitos de acordo com a real necessidade do cliente e defina também junto com o cliente as regras de negócio do sistema, como: "aluno com média abaixo de 5,0, emitir mensagem de REPROVADO".

g) A partir da análise das necessidades do cliente, descrever o que você, como analista, pode sugerir de melhorias para os requisitos do sistema.

2. Espera-se que o(a) estudante aplique os conceitos estudados nesta terceira aula.

3. Espera-se que o(a) estudante defina os requisitos funcionais e não funcionais do sistema estudado, conforme exemplos observados nesta terceira aula.

## Aula 4

1. Espera-se que o(a) estudante descreva os casos de uso para o exemplo já mapeado.



2. Espera-se que o(a) estudante desenvolva a atividade conforme exemplos estudados nesta quarta aula.

3. Espera-se que o(a) estudante identifique os casos de uso, como sugestão:

- Preencher cadastro
- Consultar o número de vagas
- Consultar data prevista para matrícula
- Gerar informações estatísticas.

Atores identificados:

- Candidato (aluno); e
- Dirigente da escola (ator que vai gerenciar as informações do sistema).

Espera-se que o(a) estudante faça o diagrama de casos de uso, o documento de casos de uso e defina as regras dos casos de uso, conforme exemplos estudados nesta quarta aula.

## Aula 5

As atividades 1 e 2 são sugestivas para que o(a) estudante descreva os diagramas destas atividades de acordo com os exemplos estudados nesta quinta aula.

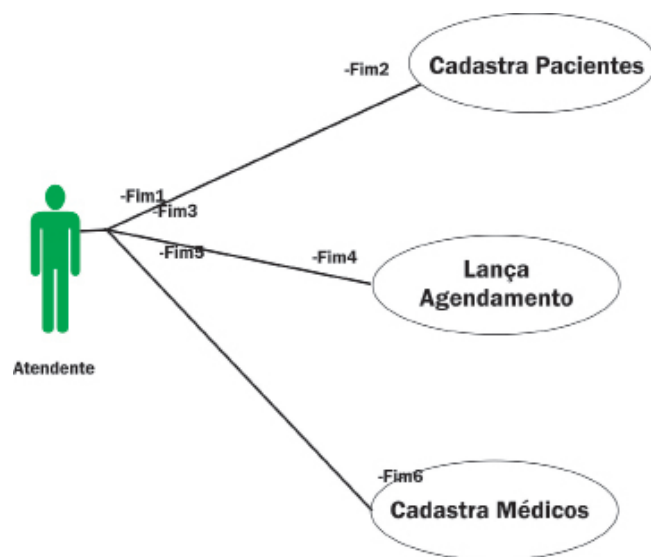
## Aula 6

As atividades 1 e 2 são sugestivas para que o(a) estudante descreva os diagramas destas atividades, que são uma sequência dos exemplos estudados nesta sexta aula.

## Aula 7

Espera-se que o(a) estudante desenvolva o processo de análise conforme os principais passos descritos nesta sétima aula e descreva os diagramas para o modelo sugerido.

Diagrama de casos de uso do cenário estudado.





## Referências

BEZERRA, Eduardo. **Princípios de análise e projeto de sistema com UML**. Rio de Janeiro: Campus Elsevier, 2011.

MELO, Ana Cristina. **Desenvolvendo aplicações com UML 2.2 do Conceito a Implementação**. 3. ed. Rio de Janeiro: Brasport, 2010.

\_\_\_\_\_. **Desenvolvendo aplicações com UML 2.2**. 3. ed. Rio de Janeiro: Brasport, 2010.

PRESSMAN, Roger S. **Engenharia de Software** – Uma abordagem profissional. 7. ed. New York: McGraw Hill, 2011.

SBROCCO, José Henrique de Carvalho; MACEDO, Paulo Cesar de. **Metodologias Ágeis – Engenharia de Software**. 1. ed. São Paulo: Érica, 2012.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson, 2004.

VERGILIO, Silvia R. **Diagrama de Interação**. Disponível em: <<http://www.inf.ufpr.br/silvia/ESNovo/UML/ppt/DiagIntera.ppt>> Acesso em 27 nov. 2013.

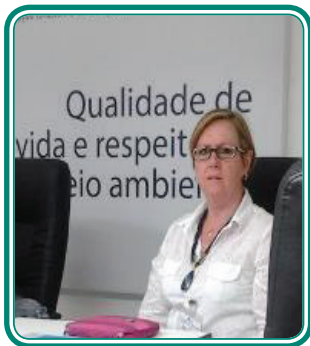
WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 2. ed. Rio de Janeiro: Campus Elsevier, 2011.

## Bibliografia Básica

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. 2. ed. Rio de Janeiro: Campus Elsevier, 2011.



## Currículo da Professora-autora



**Eliana Rovay Detregiacchi Pires** possui graduação em Análise de Sistemas pela Pontifícia Universidade Católica de Campinas - PUCC (1981), especialização em Ciências da Computação pela Universidade Federal de Santa Catarina - UFSC (2001), MBA em Gestão de Negócios e MBA de TI.

É analista de sistemas e coordenadora de TI na Companhia de Águas e Esgotos de Rondônia-CAERD e atua como professora especialista na Faculdade de Ciências Administrativas e de Tecnologia - FATEC-RO e na Faculdade São Lucas / São Mateus RO.

Ministrou as disciplinas de Análise de Sistemas, Banco de Dados, Gerenciamento de Projeto de Sistemas, Sistema de Informação e Gestão Estratégica de TI.

