



# Projeto de Sistemas WEB

*Maria Ivanilse Calderon Ribeiro*

*Juliana Braz da Costa*

*Jhordano Malacarne Bravim*



**Cuiabá - MT**

**2015**

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/RO para a Rede e-Tec Brasil, do Ministério da Educação em parceria com a Universidade Federal de 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

**Diagramação**  
Tatiane Hirata

**Revisão de Língua Portuguesa**  
Livia de Sousa Lima Pulchério

**Revisão Final**  
Oreste Preti

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

**Câmpus Porto Velho Zona Norte**

**Direção-Geral**  
Miguel Fabrício Zamberlan

**Direção de Administração e Planejamento**  
Gilberto Laske

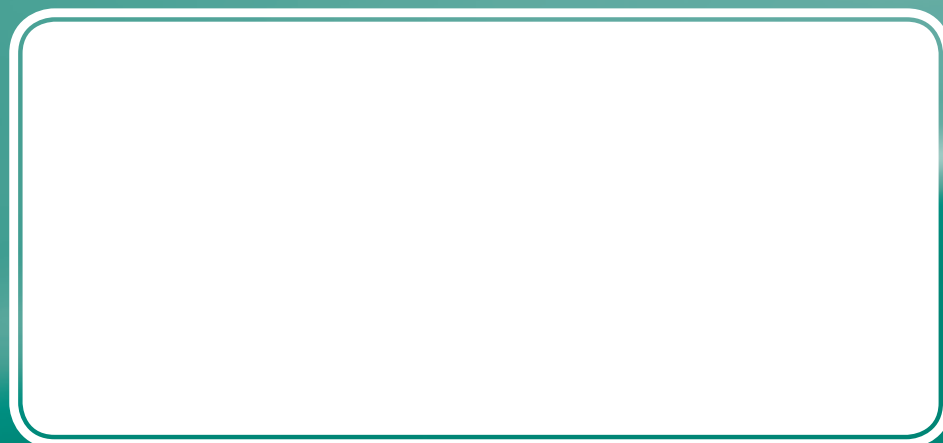
**Departamento de Produção de EaD**

**Coordenação de Produção de Material Didático Impresso**  
Ariádne Joseane Felix Quintela

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

**Coordenação da Rede E-Tec**  
Ruth Aparecida Viana de Souza

**Projeto Gráfico**  
Rede e-Tec Brasil/UFMT



# 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  
Julho de 2015

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.



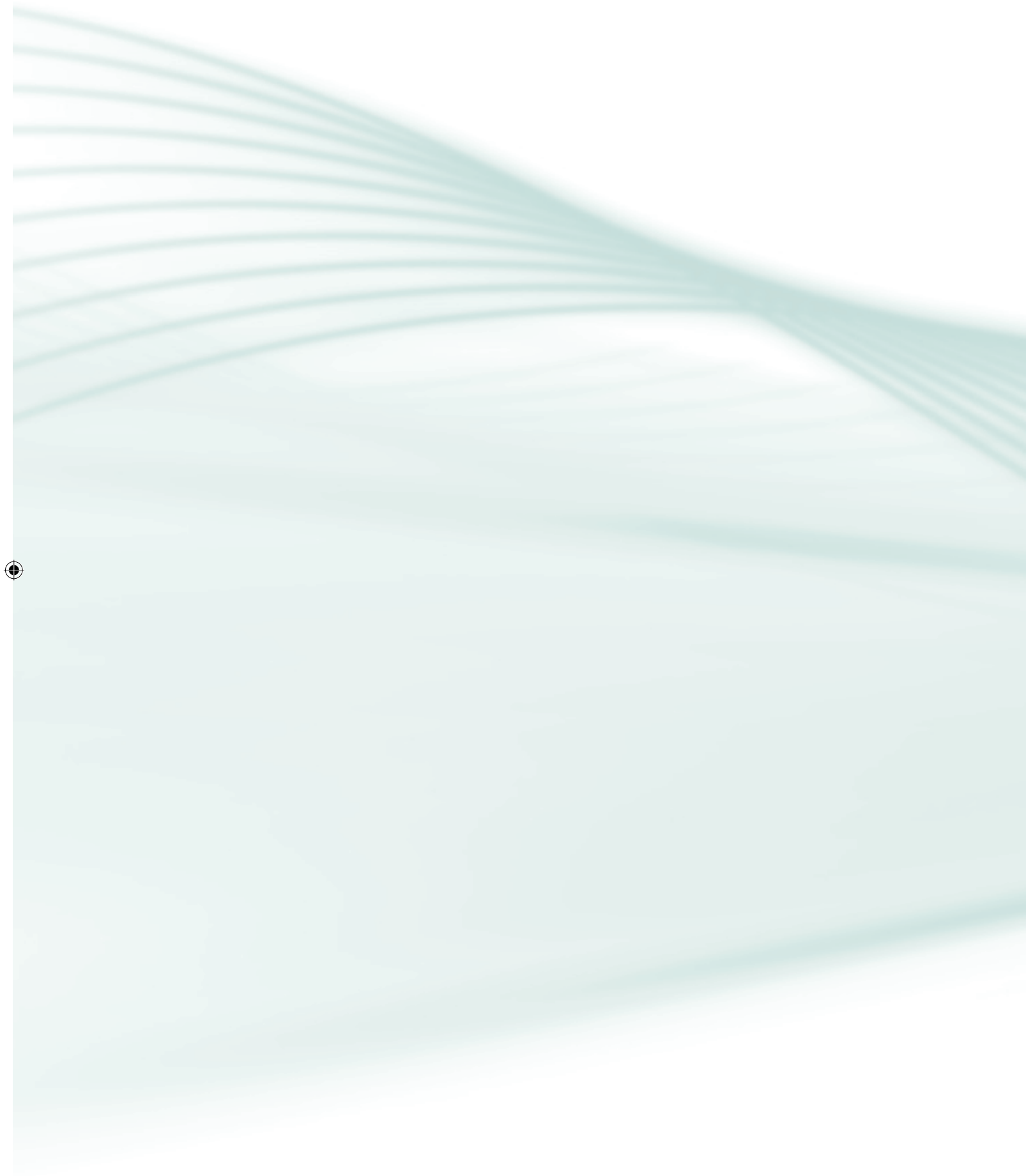
**Mídias integradas:** remete o tema para outras fontes: livros, filmes, músicas, *sites*, programas de TV.



**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.



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



# Palavra dos Professores-autores

Prezado(a) estudante:

Nós o cumprimentamos por estar agora fazendo parte de um grupo seleto de estudantes que querem o melhor em relação à formação e ao futuro profissional e por estar em busca de novos conhecimentos.

É ótimo contar com você aqui no curso Técnico em Informática para Internet e dar-lhe boas-vindas na disciplina Projeto de Sistemas Web.

Agora, como estudante desse curso, é importante que você esteja bastante motivado para o estudo do conteúdo dessa disciplina, pois o curso possibilitar-lhe-á novas oportunidades no mercado de trabalhos atual.

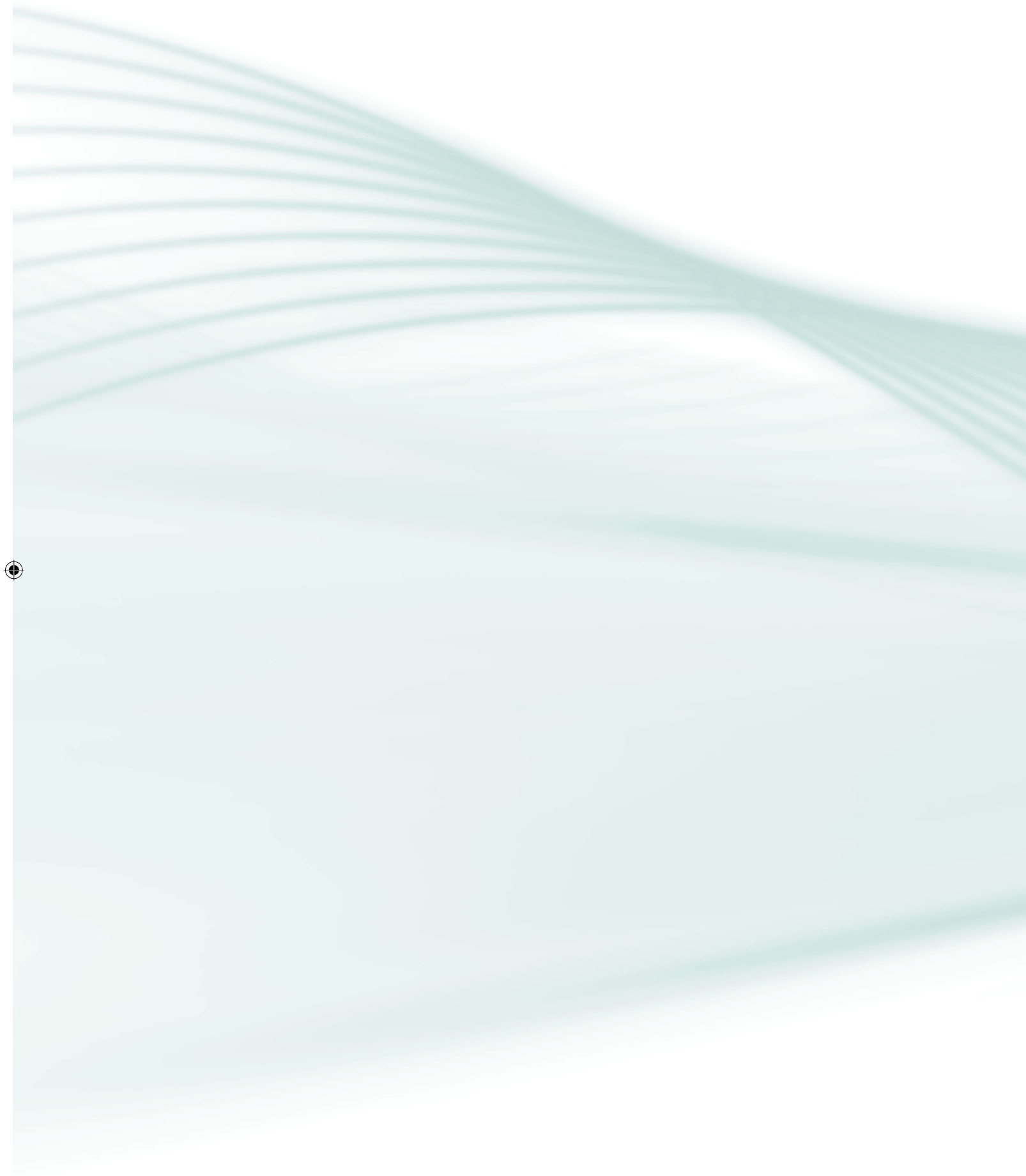
A disciplina Projeto de Sistemas Web proporcionar-lhe-á conhecimentos acerca da importância de se projetar um sistema Web para a prática do desenvolvimento evitando maiores falhas e perda de tempo. É a oportunidade que você tem para agregar novos conhecimentos acerca da prática e da construção de projetos de sistema Web na prática profissional.

Produzimos esse material com muito carinho e esperamos que seja de grande proveito para seu estudo nesse momento e em futuras pesquisas em sua prática profissional.

Durante o período das aulas da disciplina, você terá a oportunidade de integrar por meio do ambiente virtual de aprendizagem e participar de fóruns, *chats*, tarefas e outras atividades que o auxiliarão na construção de conhecimento, de aprendizagem e de outras habilidades.

Estamos como você e queremos que, a partir de agora, você avance mais uma fase na construção de seu conhecimento e formação profissional.

Bons estudos e não perca tempo. Conte conosco em sua caminhada rumo a novos conhecimentos!





# Apresentação da Disciplina

Esta disciplina tem por objetivo proporcionar-lhe conhecimentos sobre a construção de projetos de sistemas Web, no intuito de construir um profissional que compreenda o ciclo de vida de desenvolvimento de *softwares* e suas principais etapas.

Nesta disciplina, você também irá aprender sobre a utilização e importâncias dos padrões de projetos (*design patterns*) e sobre as principais metodologias de utilizadas para produção de projetos observando o futuro desenvolvimento de *softwares*, como utilizá-las buscando produzir da melhor forma o projeto de sistemas. O projeto de sistemas é parte inicial do processo de desenvolvimento do *software* e, portanto, é nessa disciplina que você aprenderá sobre a atividade de levantamento dos requisitos, análise e demais ações que envolvam esses requisitos do sistema.

Também irá conhecer e/ou aprimorar seus conhecimentos acerca da linguagem de modelagem unificada (UML), muito importante para a criação do projeto de sistema Web e para o desenvolvimento de sistemas.

Aprenderá, ainda, conceitos e processos da modelagem de sistemas, muito importantes para o futuro profissional técnico em Informática para internet, uma vez que será ele que produzirá ou mesmo gerenciará sistemas Web.

Além de estudar as tecnologias aplicadas ao projeto de sistemas Web, receberá sugestões para praticar e estudar essa disciplina de modo a ter bom aproveitamento nos estudos.

Assim, esperamos capacitá-lo a construir ou melhor a entender a construção e o planejamento do projeto de sistemas Web. Busque sempre praticar e refletir sobre as atividades propostas, pois ajudá-lo-ão a desenvolver suas habilidade para sua atuação e sucesso profissional.

Bons estudos e sucesso!



# Sumário

<b>Aula 1. Projeto de sistemas para Web</b> .....	<b>13</b>
1.1 Conceitos sobre projeto de sistemas Web.....	14
1.2 Importância do projeto de sistemas Web.....	16
1.3 Projeto de sistemas Web.....	18
1.4 Qualidade do projeto de sistemas Web.....	21
1.5 Projeto de sistemas e padrões de projeto ( <i>design patterns</i> ).....	24
1.6 Documentação do projeto .....	27
<b>Aula 2. Levantamento, especificação e análise de requisitos</b> .....	<b>31</b>
2.1 Levantamento de requisitos para projeto de sistemas Web.....	32
2.2 Importância dos requisitos para o projeto de sistemas Web.....	35
2.3 As especificações e análise de requisitos para projeto de sistemas Web.....	36
<b>Aula 3. Modelagem, engenharia e validação de requisitos</b> .....	<b>43</b>
3.1 Importância da modelagem para projeto de sistemas Web.....	44
3.2 Engenharia de requisitos para o projeto de sistemas Web.....	46
3.3 Validação dos requisito para o projeto de sistemas Web.....	47
<b>Aula 4. Metodologias e ferramentas para modelagem de sistemas Web</b> .....	<b>51</b>
4.1 Metodologias para projetos de sistemas Web.....	51
4.2 Modelo de análise de projeto de sistemas Web.....	53
4.3 Tipos de modelos para a modelagem de sistemas web.....	54
<b>Aula 5. Utilização da linguagem de modelagem unificada para projetos de sistema Web</b> .....	<b>61</b>
7.1 Aprendendo sobre a UML.....	62
7.2 Partes que compõem a UML.....	66
7.3 Diagramas da UML.....	67
<b>Palavras Finais</b> .....	<b>76</b>



<b>Guia de Soluções</b> .....	<b>77</b>
<b>Referências</b> .....	<b>79</b>
<b>Currículo dos Professores-autores</b> .....	<b>80</b>



# Aula 1. Projeto de sistemas para Web

## Objetivos:

- identificar as características, particularidades e qualidades do projeto de sistemas *Web*;
- compreender a importância do projeto de sistemas *Web*; e
- reconhecer a documentação e o padrão de projeto (*design pattern*).

Estimado(a) estudante:

Na primeira aula da disciplina, você terá uma ideia geral sobre projeto de sistemas *Web* e começará a construir base sólida de conhecimento para sua formação e atuação profissional.

Nós o convidamos para iniciar seus estudos nesta disciplina com motivação e interesse. Temos certeza de que o curso que está frequentando foi uma ótima escolha e o tornará um excelente profissional. Esperamos que esta disciplina contribuirá positivamente para a sua formação.

## Introdução

A construção de projeto de sistemas *Web* apresenta peculiaridades refletindo as necessidades a serem atendidas pelo sistemas, particularmente os aspectos da arquitetura e da usabilidade do projeto para sistema *Web* que envolve múltiplas especificidades e particularidades da plataforma..

A *Web* concretiza novos meios e importantes modos de comunicação comercial ou não, diversos do tradicional, os quais se refletem também no projeto de desenvolvimento de sistemas *Web*. As necessidades atuais com relação à agilidade e à usabilidade dos *sites* e sistemas *Web* propiciaram espaço promissor para o profissional técnico em informática para internet.

É fundamental entender que desenvolver projetos de sistemas para *Web* é



uma prática complexa quando se leva em consideração a plataforma em que se trabalha, por ser formada por componentes inter-relacionados. Além disso, é importante que você reconheça a necessidade de utilização de métodos, técnicas e ferramentas, observando prazos, orçamentos e clientes.

Deve compreender que projetos de sistemas Web devem ser desenvolvidos baseados em princípios de engenharia, observando o desenvolvimento de um produto complexo que envolve métodos, técnicas e ferramentas, modelos e princípios, planejamento e gerenciamento, qualidade do produto e do processo de desenvolvimento.



**Num projeto de *software* você sempre deve começar levando em consideração os dados - a base para todos os demais elementos do projeto. Após estabelecida a base, a arquitetura tem de ser extraída. Só então deve realizar outras tarefas de projeto (PRESSMAN, 2011).**

Portanto, é primordial estudar a evolução do desenvolvimento de sistemas Web, o aparecimento de um novo paradigma para a desenvolvimento de sistemas, diferente do desenvolvimento *desktop*.



Mas, o que é projeto de sistemas Web? Quais são as características e peculiaridades na construção de projeto Web? Como ocorreu a evolução do desenvolvimento de sistemas Web? Como são os processos de desenvolvimento de sistemas Web?

Venha conosco. Nessa aula, vamos rumo à construção de novos conhecimentos.

## 1.1 Conceitos sobre projeto de sistemas Web

O projeto de sistemas Web necessita ser observado a partir de princípios da engenharia de *softwares*. Trata-se de atividade complexa e formada por componentes inter-relacionados, pensados e desenvolvidos, geralmente, por equipes de especialistas, de profissionais que possuem conhecimentos sobre o assunto e familiarizados com técnicas e ferramentas que proporcionam melhor aplicação da metodologia escolhida para a construção do projeto.

Os sistemas para Web, diferentemente dos sistemas utilizados em outra plataforma, requerem desenvolvimento baseado em princípios de engenharia,





por ser um ambiente de maior complexidade.

Um projeto de sistemas Web deve apresentar características que funcionam de maneira eficaz com a construção dos sistemas. Propriedades como funcionalidade, eficiência, robustez, confiabilidade, portabilidade, facilidade para a utilização, entre outras características, você irá conhecer nessa aula.

Refleta sobre as seguintes questões:

- a) Por que projetar um sistema Web?
- b) Qual é a necessidade de produzir um projeto de sistemas Web?
- c) Qual a finalidade de um projeto de sistemas Web?

Anote suas reflexões:

---

---

---

---

---



Segundo o engenheiro de *software* Roger S. Pressman (2011), a atividade de projeto de *software* engloba um conjunto de princípios, conceitos e práticas que levam ao desenvolvimento de um sistema ou produto com alta qualidade. Os princípios de projeto estabelecem uma filosofia que prevalece sobre as atitudes e ações do desenvolvimento, orientando as atividades para realizar o projeto.

Para o autor norte-americano, o projeto de *software* é um processo iterativo através do qual os requisitos são traduzidos em uma “planta” para construir o *software*. Inicialmente, a planta representa uma visão **holística** do *software*. O projeto é representado em um alto nível de abstração - um nível que pode ser associado diretamente ao projeto específico do sistema e aos requisitos mais detalhados de dados, funcionalidade e comportamento.

Bem, esperamos que possa pensar um pouco sobre as possíveis respostas para tais questões. Estamos aqui para compartilhar com você conceitos e práticas que lhe darão suporte para responder tais questões de forma progressiva, à medida que construímos novos conhecimentos a partir do anda-



O termo **holismo** origina-se do grego *holos*, que significa todo, inteiro. Significa uma “visão global”, em que o “todo é maior que a simples soma das partes”.





mento desta disciplina.

## 1.2 Importância do projeto de sistemas Web

Sistemas e aplicações baseados na Web (WebApps) produzem uma complexa matriz de conteúdo e funcionalidade para ampla população de usuários finais. A engenharia da Web (WebE) é o processo usado para criar WebApps de alta qualidade. A WebE não é um clone perfeito da engenharia de *software*, mas toma emprestados muitos dos conceitos e princípios fundamentais da engenharia de *software*. Além disso, o processo WebE enfatiza atividades técnicas e de gestão similares. Há diferenças sutis no modo pelo qual essas atividades são conduzidas, mas a filosofia dominante determina uma abordagem disciplinada para o desenvolvimento de um sistema baseado em computador, conforme ensina Pressman no livro “Engenharia da Web”.

Nesse sentido, você já poderá imaginar ou mesmo inferir o porquê e a importância de estudarmos o projeto de sistemas Web. O referido autor menciona a complexidade matriz de conteúdo e funcionalidade ao referir-se a sistemas e aplicações para Web, contudo deixa claro também que devem ser seguidos alguns conceitos e princípios fundamentais da engenharia de *software*, com algumas diferenças sutis.

Para Pressman (2011), o objetivo da atividade de projetar é gerar um modelo ou representação que apresente solidez, comodidade e deleite. Para tanto, temos de praticar a diversificação e, depois, a convergência.



Pense, então, sobre a complexidade e a abrangência de um sistema que você tenha em mente para desenvolver ou mesmo que tenha necessidade para utilizar e logo perceberá a importância de se ter um projeto para aprimoramento gradual do produto final.

É importante que visualize sempre o escopo do projeto e a importância de se produzir um projeto para um sistema que será disponibilizado na Web. Deve, ainda, lembrar a estratégia de negócio a que o sistema irá atender e as peculiaridades e/ou necessidades das mais simples às mais complexas.

O projeto de *software* muda continuamente à medida que novos métodos, melhor análise e entendimento mais abrangente evoluem. Mesmo hoje em dia, a maioria das metodologias de projeto de *software* carece de profun-







didade, flexibilidade e natureza quantitativa que normalmente estão associadas às disciplinas mais clássicas de engenharia de projeto. Entretanto, existem efetivamente métodos para projeto de *software*, critérios para qualidade de projeto estão disponíveis e notação de projeto pode ser aplicada (PRESSMAN, 2011).

Pode-se dizer que a importância de se construir um sistema está relacionada a um sistema que atenderá às expectativas do cliente. Deve ser projetado para ser um sistema confiável, usual, adaptável, apresentando essas e outras características no sistema finalizado. É no projeto de sistema que os profissionais envolvidos, bem como o cliente podem perceber necessidades não contempladas durante a extração de requisito com o profissional responsável, podendo essas serem implantadas em tempo hábil, com menor desgaste dos profissionais e menor custo.

Além dos princípios gerais de projeto, Hooker (1996, apud PRESSMAN, 2006) enumera sete *princípios gerais* da engenharia de *software* que se aplicam também ao projeto de *software*. São eles:

- Um sistema de software existe para fornecer valor aos clientes e usuários.
- Todas as decisões, inclusive as de projeto, devem ser tomadas tendo isso em mente.
- Todo projeto de software deve ser tão simples quanto possível sem, no entanto, descartar características de qualidade importantes em nome da simplicidade.
- O comprometimento com a visão arquitetural do sistema é essencial para o sucesso do projeto de software.
- Os modelos elaborados na fase de projeto serão usados posteriormente por desenvolvedores responsáveis pela implementação, testes e manutenção do sistema. Assim, esses modelos devem ser claros, não ambíguos e fáceis de entender.
- Um sistema com um longo tempo de vida tem mais valor. Contudo, para ter vida longa, um sistema deve ser projetado para estar pronto para acomodar mudanças.



- A reutilização pode ajudar a poupar tempo e esforço, bem como aumentar a qualidade do sistema em desenvolvimento. Para conseguir um bom nível de reutilização, é necessário planejar o reuso com antecedência. Na fase de projeto, padrões arquitetônicos e padrões de projeto detalhado (*design patterns*) são bastante maduros e documentados.



**Podemos lembrar, ainda, que projetar proporcionará viabilidade econômica, melhoria da qualidade, redução de riscos, erros e retrabalhos, economia de recursos, aumento de produtividade, entre outros pontos importantes que você irá aprender durante as aulas desta disciplina.**

### 1.3 Projeto de sistemas Web

Com a popularização da Web, nada mais normal que a maioria das necessidades da sociedade esteja ligada à rede. Pense conosco: com maior necessidade e utilização da Web, maior é a necessidade de se projetarem e desenvolverem sistemas que acompanhem a veloz complexidade dessa plataforma.

Assim, pedimos-lhe que reflita sobre o que Pressman exemplifica sobre o desenvolvimento e a necessidade de se evoluir ou mesmo se adaptar à maneira de projetar sistemas voltados para a Web.

A *World Wide Web* e a internet, que lhe dá poder, são indiscutivelmente os desenvolvimentos mais importantes da história da computação. Essas tecnologias nos colocaram (com bilhões de pessoas a mais que irão eventualmente nos seguir) na era da informação.

Elas se tornaram integradas à vida diária na primeira década do século XXI. Para aqueles que ainda se lembram de um mundo sem a Web, o crescimento caótico da tecnologia nos remete ao passado, à outra época - os primeiros dias do *software*. Era uma época de pouca disciplina, mas de enorme entusiasmo e criatividade. Era uma época em que programadores frequentemente juntavam sistemas - alguns bons, outros ruins (PRESSMAN, 2009, p. 380).

Você deve ter em mente que, para se desenvolver um *software*, diversas atividades e profissionais são envolvidos, geralmente atividades técnicas, gerências, de garantia de qualidade dentre outras, em busca de produzir produto que satisfará as necessidades do cliente.



A fase de projeto buscar definir e especificar soluções visualizadas pelos profissionais envolvidos a partir do momento em que esses conhecem os requisitos necessários para a produção do projeto. Nessa fase, por haver muitas formas para solucionar certo problema, inicia-se uma fase na qual os envolvidos deverão tomar decisões.

O projeto também deve ser visto como uma fase ou processo que busca refinar, conhecer as possíveis possibilidades de soluções a serem apresentadas e ao ser definida a arquitetura do sistema, cada parte do projeto será tratada detalhadamente.

Para Pressman (2006), o projeto de *software* encontra-se no núcleo técnico do processo de desenvolvimento desse *software* e é aplicado independentemente do modelo de ciclo de vida e paradigma adotados. Assim que os requisitos do *software* tiverem sido modelados e especificados pelo menos parcialmente, entra-se na última atividade de modelagem. Por outro lado, corresponde à primeira atividade que leva em conta considerações de caráter tecnológico.

Você já deve ter percebido que o projeto de sistemas Web é o passo inicial e final para o desenvolvimento de um sistema nessa plataforma de tamanha complexidade que é a Web.

Lembre-se de que, independentemente do ciclo de vida do sistema ou mesmo do paradigma que você adotará para o implementar, deverá iniciar a primeira atividade que leva em conta questões de caráter tecnológico, ou seja, o projeto.



Por ser o projeto um processo de refinamento, busca as mais diversas e possíveis soluções para determinado problema ou necessidade do negócio em questão. Você deve entender que o projeto está em um nível alto de abstração, ou seja, nele está assinalada a estrutura geral do sistema.

Observe a figura 1 na próxima página e imagine um problema real do seu cotidiano que poderia ter um sistema Web capaz de gerenciar e solucionar eventuais necessidades, melhorando assim o desempenho e a solução do problema que você imaginou. Procure abstrair o problema real e elenque o quê precisa ser feito nesse sistema.





**Figura 1 - Fase de Projeto**

Adaptação: PRESSMAN, 2006



Após a apresentação do que você pensou que deve ser feito, ainda observando a figura 1, como seria implementado ou desenvolvido esse sistema?

Pois bem, é nessa fase em que estamos já falando do projeto desse sistema que você acaba de imaginar que o projeto permitirá que os profissionais envolvidos na busca da solução para o problema apresentado no mundo real possa ser resolvido no mundo computacional.

Mitch Kapur, citado por Pressman (2006), explica que o projeto é “onde você se instala com um pé em dois mundos – o mundo da tecnologia e o mundo das pessoas e objetivos humanos – e você tenta juntar os dois”.

O projeto é a fase do processo de *software* na qual os requisitos, as necessidades do negócio e as considerações técnicas se juntam na formulação de um produto ou sistema de *software* (PRESSMAN, 2006).

Independentemente do paradigma adotado, o processo de projeto envolve as seguintes atividades (PRESSMAN, 2006):

- Projeto da arquitetura do software - Visa definir os elementos estruturais do software e seus relacionamentos.
- Projeto dos elementos da arquitetura - Visa projetar em um maior nível de detalhes cada um dos elementos estruturais definidos na arquitetura, o que envolve a decomposição de módulos em outros módulos menores.





- Projeto detalhado - Tem por objetivo refinar e detalhar os elementos mais básicos da arquitetura do software: as interfaces, os procedimentos e as estruturas de dados. Deve-se descrever como se dará a comunicação entre os elementos da arquitetura (interfaces internas), a comunicação do sistema em desenvolvimento com outros sistemas (interfaces externas) e com as pessoas que vão utilizá-lo (interface com o usuário), bem como se devem projetar detalhes de algoritmos e estruturas de dados.

Perceba que as atividades elencadas pelo autor permitem obter mais detalhes e informações em relação às necessidades e à solução do problema. É um nível de detalhamento que permitirá entender o produto desde seus elementos estruturais até a modelagem final do sistema.

O projeto é o processo criativo de transformar a especificação de um problema na especificação de uma solução. No projeto de *software*, utilizam-se a especificação e os modelos de requisitos gerados na fase de análise e especificação de requisitos. A partir dos requisitos, muitas soluções são possíveis e, portanto, muitos projetos diferentes podem ser produzidos. Uma solução é considerada adequada ao problema, se ela satisfizer a todos os requisitos especificados (PFLEEGER, 2004).

## 1.4 Qualidade do projeto de sistemas Web

O responsável pela construção do projeto de sistema Web deve estar atento às normas de qualidade, ou seja, deve ter a preocupação em incorporar requisitos tecnológicos essenciais ao projeto capazes de agregar diversos atributos de qualidade geralmente definidos pela ISO/IEC.

O projetista deve atentar-se às características de qualidade que geralmente não estão restritas às funcionalidades do sistema, mesmo que estejam de alguma forma ligada a essas. Assim, serão evitados o retrabalho ou mesmo as reconstruções do sistema.

A ISO/IEC (2001) utiliza como referência para a avaliação de produtos de *software* seis *características de qualidade*, que você deve conhecer e estudar para entender melhor sobre a qualidade do projeto de sistemas Web, que são:

- Funcionalidade - Refere-se à existência de um conjunto de funções que satisfazem às necessidades explícitas e implícitas e suas propriedades es-





pecíficas. Tem como subcaracterísticas: adequação, acurácia, interoperabilidade, segurança de acesso e conformidade.

- Confiabilidade - Diz respeito à capacidade do software de manter seu nível de desempenho, sob condições estabelecidas, por um período de tempo. Tem como subcaracterísticas: maturidade, tolerância a falhas, recuperabilidade e conformidade.
- Usabilidade - Refere-se ao esforço necessário para se utilizar um produto de software, bem como o julgamento individual de tal uso por um conjunto de usuários. Tem como subcaracterísticas: inteligibilidade, apreensibilidade, operacionalidade, atratividade e conformidade.
- Eficiência - Diz respeito ao relacionamento entre o nível de desempenho do software e a quantidade de recursos utilizados sob condições estabelecidas. Tem como subcaracterísticas: comportamento em relação ao tempo, comportamento em relação aos recursos e conformidade.
- Manutenibilidade - Concerne ao esforço necessário para se fazer modificações no software. Tem como subcaracterísticas: analisabilidade, modificabilidade, estabilidade, testabilidade e conformidade.
- Portabilidade - Refere-se à capacidade do software de ser transferido de um ambiente para outro. Tem como subcaracterísticas: adaptabilidade, capacidade para ser instalado, coexistência, capacidade para substituir e conformidade.

Pressman (2011) relaciona diferentes diretrizes de qualidade, afirmando que, para avaliar a qualidade da representação de um projeto, você e os outros membros da equipe de *software* devem estabelecer critérios técnicos para um bom projeto, considerando as seguintes *diretrizes*:

- a) Um projeto deve exibir uma arquitetura que foi criada usando estilos ou padrões arquiteturais reconhecíveis, seja composta por componentes que apresentem boas características do projeto ou que possa ser implementada de forma evolucionária facilitando, portanto, a implementação e os testes.
- b) Um projeto deve ser modular e o software deve ser logicamente dividido em elementos ou subsistemas, de modo que seja facilmente testado e mantido.





- c) Um projeto deve conter representações distintas de: dados, arquitetura, interfaces e componentes.
- d) Um projeto deve levar as estruturas de dados adequadas às classes a serem implementadas e baseadas em padrões de dados reconhecíveis.
- e) Um projeto deve levar a componentes que representem características funcionais independentes (baixo acoplamento).
- f) Um projeto deve levar a uma interface que reduza a complexidade das conexões entre os componentes e o ambiente externo (encapsulamento).
- g) Um projeto deve ser obtido usando-se um método repetível, isto é, dirigido por informações obtidas durante a análise de requisitos de software.
- i) Um projeto deve ser representado usando-se uma notação que efetivamente comunique seu significado.

Pressman (2011), ainda, enfatiza que essas diretrizes não são atingidas por acaso. Elas são alcançáveis por meio da aplicação de princípios de projeto fundamentais, de metodologia e sistemática e de revisão.

O PMBOK (2004) descreve que o projeto é um empreendimento temporário, de elaboração progressiva e com o objetivo de criar um produto ou serviço único, apresentando as seguintes fases:

- Temporário - O projeto possui início e fim bem definidos e pode ser de curta ou longa duração. O projeto chega ao fim quando os seus objetivos são atingidos.
- Elaboração progressiva - O desenvolvimento ocorre em etapas e continua por incrementos.
- Produto ou serviço único - Cada projeto é exclusivo.

A **qualidade** de *software* não pode ser entendida como perfeição.

Perceba que cada atributo de avaliação permite conhecer melhor o sistema que está sendo projetado, ou seja, relaciona a adoção de abordagem consistente em um contexto prático específico para o desenvolvimento de

## A-Z

**Qualidade** é um conceito multidimensional, realizado por um conjunto de atributos, representando vários aspectos relacionados ao produto: desenvolvimento, manutenção e uso. Qualidade é algo factível, relativo, dinâmico e evolutivo, adequando-se ao nível dos objetivos a serem atingidos (SIMÃO, 2002).



sistemas ou aplicações Web.



**É a partir de cada atributo de qualidade que o projetista tem a noção do que precisa ser feito, o que deve melhorar e o que já tem avançado progressivamente.**

## 1.5 Projeto de sistemas e padrões de projeto (*design patterns*)

Como você já conhece alguns conceitos acerca de projetos de sistemas, certamente, deve saber algo sobre *patterns*, ou seja, sobre padrões.

Qual a importância dos *patterns* no processo construção de projeto de sistemas e logicamente para o desenvolvimento do sistema projetado?



Bem, pense conosco sobre um projeto de desenvolvimento para um sistema Web que um cliente solicitou que você projete. Responda as seguintes questões antes de dar o primeiro passo rumo ao projeto:

- 1) Trata-se de um projeto novo, você ainda não tem nada parecido?
- 2) Existem projetos que já trataram do negócio apresentado?
- 3) É necessário partir do zero para iniciar tal projeto?

Com certeza, você deve ter percebido que as respostas para as três perguntas direcionam para a reutilização de um projeto já existente, vez que não é necessário partir do zero, caso algum projeto apresente aspectos desejáveis que possam ser reutilizados em outros projetos. Assim, estamos falando da reutilização que é um aspecto fundamental para o desenvolvimento de *software*.

Lembre-se de que, de alguma maneira, quando tratamos de um projeto de desenvolvimento de sistema, estamos tratando de um novo projeto que será novo na medida em que vamos progredindo com o novo sistema. Isso pode ocorrer por vários motivos e não é preciso que seja desenvolvido do zero, vez que podem existir peculiaridades em sistemas já projetados.

Assim, você deve entender que um padrão (*pattern*) é uma solução já existente, testada e aprovada para problemas que de certa forma podem surgir





em diversas outras necessidade de cliente diferentes.

*Design patterns* não são aplicados somente na informática, podem ser encontrados em diversas áreas de conhecimento como a engenharia e a arquitetura.

Foi o arquiteto austríaco Christopher Alexander (1936-) que introduziu a ideia de *patterns*, em 1970, para construir um vocabulário comum para discussões sobre *design*.

Para Pressman (2011),

Cada *Pattern* descreve um problema que ocorre várias vezes ao nosso redor e, com isso, descrevem a solução para o problema de uma maneira que você pode usar essa solução diversas vezes sem ter que fazer a mesma coisa duas ou mais vezes.



**Christopher Alexander**  
Fonte: [www.nbm.org](http://www.nbm.org)

O objetivo de um padrão de projeto é registrar uma experiência no projeto de *software*, que possa ser efetivamente utilizada por projetistas. Cada padrão sistematicamente nomeia, explica e avalia uma importante situação de projeto que ocorre repetidamente em sistemas (GAMMA *et al.*, 1995).

Um profissional que conheça padrões poderá utilizá-los evitando, assim, o trabalho de um novo projeto para problemas já conhecidos. Não será necessário pensar o mundo real para o mundo computacional, vez que já possui o conhecimento de abstrações já feitas e daí apenas prosseguir para a tomada de decisões do projeto.

Observando os estudos de Buschmann *et al.* (1996), um padrão tem os seguintes elementos:

- Nome: identificação de uma ou duas palavras, utilizadas para nomear o padrão.
- Contexto: uma situação que dá origem a um problema.
- Problema: explica o problema que surge repetidamente no dado contexto.
- Solução: descreve uma solução comprovada para o problema, incluindo





os elementos que compõem o projeto, seus relacionamentos, responsabilidades e colaborações.

- Consequências: são os resultados e os compromettimentos feitos ao se aplicar o padrão.

Um *pattern* não descreve um projeto específico, sendo, na verdade, uma oportunidade de reutilizar um problema já abstraído do mundo real. Por esse motivo, devem-se observar criteriosamente os resultados e o comprometimento existentes a partir dessa utilização.

De certa forma, um *pattern* de projeto descreve uma estrutura comumente recorrente de componentes que se comunicam, capaz de resolver um problema de projeto geral dentro de um particular contexto (GAMMA et al., 1995).

Observe a conversa entre os programadores demonstrada na “tirinha” a seguir:



Figura 2

Fonte: <http://vidadeprogramador.com.br/>



Pense conosco. Qual seria a motivação para se utilizarem os padrões de projetos, ou seja, por que motivo o projetista ou o desenvolvedor adotaria um padrão de modelo no momento da execução do projeto ou da implementação do sistema?

Assim, você pode solucionar tais indagações com a seguinte reflexão: Ora, se existem a frequência e/ou recorrência de determinados problemas, nada melhor que utilizarmos uma solução já pronta que resolva o problema que repetidamente ocorre”, não concorda?.

Os padrões de projeto devem ser observados como ferramentas para o projetista e o desenvolvedor de *software*, vez que auxiliará na solução de problemas que ocorrem com frequência, permitindo que os esforços de tais profissionais sejam utilizados em outra demanda.

## 1.6 Documentação do projeto

Por se tratar de projeto, você, estudante desse Curso Técnico em Informática para Internet, deve compreender a importância da documentação do projeto de sistemas Web. Você já aprendeu que o projeto de um sistema é o início técnico do processo de desenvolvimento e por isso é de grande importância, caso você queira obter sucesso com vida longa ao projeto implementado.

Assim, ao projetar um sistema, o projetista precisará da documentação do projeto do sistema para qualquer manutenção futura, lembrando que a equipe de profissionais envolvidos em determinado projeto pode ser transitória ou sofrer alguma modificação durante a construção do projeto. Outra necessidade que já se pode visualizar é em relação às peculiaridades de diferentes projetos.

No desenvolvimento de um sistema, uma gama de profissionais pode estar envolvida, às vezes com diferentes necessidades e, por isso, a documentação do projeto se faz importante. Em outras palavras, diferentes interessados vão requerer informações diferentes e a documentação de projeto é primordial para a comunicação entre esses profissionais.

Para Bass; Clements; Kazman (2003), a documentação do projeto de sistemas deve conter:

- informações gerenciais, tais como versão, responsáveis, histórico de alte-





rações;

- uma descrição geral do sistema; e
- uma lista das visões consideradas e informações acerca do mapeamento entre elas.

Lembre-se de que as visões podem ser apresentadas por meio de representação gráfica, tabular ou textual. Elas são essenciais e a documentação deve abranger visões consideradas relevantes e abrangentes do sistema.

Vamos resumir os pontos mais importantes que foram desenvolvidos até aqui?

## Resumo

Projeto de sistemas Web é uma fase muito importante para o desenvolvimento de *software*, pois os profissionais e as demais pessoas que utilizarão ou manipularão o sistema para qualquer finalidade necessitarão do projeto de sistema.

A dinamicidade que nos propõe a Web se reflete na mutação rápida das necessidades atuais e, por isso, a projeção do projeto de sistema dará suporte e respaldo aos profissionais envolvidos no desenvolvimento, teste e manutenção do sistema.

Sem dúvida, conhecer e entender as características, importância e peculiaridades acerca do projeto do sistema permitirá que você domine esse processo dinâmico que reflete as necessidades atuais dos usuários e clientes, vez que podemos contar com diversas tecnologias para buscar de maneira mais rápida e eficiente tal elaboração e construção.



## Atividade de Aprendizagem

Após a leitura do conteúdo desta primeira aula, você já ficou mais familiarizado com o projeto de sistemas Web. Com base em seus conhecimentos, escreva sobre a importância de se fazer um projeto de sistemas Web. Procure mencionar aspectos relacionados à qualidade e à utilização de padrões.

---

---

---

---





---

---

---

---

Você chegou ao final da primeira aula sobre características e particularidades do projeto de sistemas Web; já conhece a importância da construção do projeto de sistemas Web, a necessidade de se preocupar com a qualidade do projeto de sistemas Web; a documentação e padrões (*patterns*).

Esperamos que você tenha gostado do conteúdo desta aula. É importante, para o aprimoramento do conhecimento construído, que você faça outras leituras sempre que possível.





## Aula 2. Levantamento, especificação e análise de requisitos

### Objetivos:

- compreender a atividade de levantamento de requisitos;
- saber descrever a modelagem de requisitos para o projeto de sistemas Web; e
- ser capaz de especificar e analisar os requisitos do projeto de sistemas Web.

Prezado(a) estudante:

Iniciamos agora a segunda aula da disciplina “Projeto de Sistemas Web” do Curso Técnico em Informática para internet. Nessa aula, você compreenderá a atividade de levantamento de requisitos; aprenderá sobre os requisitos necessários para o projeto de sistemas Web e também terá noções gerais sobre a especificação e análise de requisitos do projeto de sistemas Web.

Nessa parte da disciplina, você começa a estudar os requisitos para o projeto de sistemas, ou seja, nesse momento passará a compreender a necessidade existente no mundo real para que seja projetada no mundo computacional.

Temos certeza de que, depois de estudar o conteúdo desta aula, você será capaz de realizar levantamento de requisitos e praticar a análise dos mesmo no intuito de iniciar o projeto de um sistema Web.

Vamos, então, a esse novo aprendizado com muito interesse? Certamente agora é uma ótima oportunidade para continuar a construir seu conhecimento para a prática profissional.

Ótimos estudos!



Como você já estudou na aula anterior, o desenvolvimento de sistemas Web requer um projeto que abranja todas as especificidades das necessidades apresentadas pelos usuários e/ou clientes.

Os profissionais envolvidos no projeto e no desenvolvimento de sistema Web compreendem que estão em um processo que envolve atividades técnicas e gerenciais, mas sempre observando a qualidade que se deve agregar em meio a essa complexa fase em busca do produto final.

Por onde começar?

Essa fase é iniciada a partir do levantamento dos requisitos, visando à coleta e à organização das necessidades percebidas no mundo real para que este seja projetado no mundo computacional. É essencial que, após o levantamento e coleta dos requisitos para o projeto e desenvolvimento do sistema, esses sejam modelados, avaliados, analisados e documentados.

Então, ao observar esses aspectos iniciais, você está convidado a compreender a atividade de levantamento de requisitos para o projeto e desenvolvimento de sistemas Web, além de aprender os requisitos necessários, a especificação e a análise de requisitos do projeto de sistemas Web.

## 2.1 Levantamento de requisitos para projeto de sistemas Web



Você deve estar-se perguntando: “O que são os requisitos para o projeto de sistemas Web?” Saberíamos responder?

---

---

---

---

---

É excelente que tenha essa questão em mente, pois já deve estar procurando uma maneira de responder e isso é parte do processo de aprendizado.

Pois bem, nessa etapa podemos dizer que a atividade na qual se busca conhecer os requisitos para a elaboração do projeto é a prática de identificar as necessidades que o cliente tem e buscar solução com a criação de determinado sistema.







Assim, quando o projetista realiza o levantamento de **requisitos** para criar o projeto de sistemas Web, o profissional está buscando entender a real necessidade do cliente para poder projetar e passar a desenvolver o sistema propriamente dito em busca de solução dos problemas apresentados pelo cliente.

Mas, o que é o “requisito” de que tanto falamos?

Sempre que você for estudar ou praticar alguma atividade em que tiver que realizar o levantamento de requisitos, deverá lembrar-se de que precisa conhecer bem o problema para o qual o cliente busca solução, ou seja, você deverá ter o domínio do negócio.

Quando falamos em domínio do negócio, você deve entender que os requisitos do sistema devem ser identificados a partir do conhecimento que o profissional encarregado por tal atividade tem sobre o problema no mundo real, pois isso refletir-se-á diretamente no desenvolvimento do software.

Durante suas pesquisas e estudos, você poderá deparar-se com os termos “domínio do problema” e “domínio da aplicação”. O que são?

São termos utilizados para referenciar o domínio do negócio também. Muitos autores, como Bezerra (2006), lembram que são termos utilizados no meio profissional de projetistas e desenvolvedores de sistemas.

A equipe de profissionais envolvidos no projeto e desenvolvimento de sistemas tem na atividade de levantamento de requisitos um meio para buscar entender ao máximo o problema exposto pelo cliente, pois todo esse levantamento será automatizado no mundo computacional.

Para compreender os requisitos para o projeto do sistema, o profissional realiza a atividade de levantamento de requisitos. É importante que você compreenda que essa tarefa é um estudo que busca explorar ao máximo e de forma direta as necessidades do usuário, com a finalidade de solucionar o problema, observando, ainda, a existência de outro sistema que pode estar em uso ou já ter sido utilizado.

O meio em que o usuário está inserido pode ser percebido pelo profissional mediante entrevistas com os usuários de setores e/ou departamentos distintos, com o responsável ou usuário que conheça a fundo o negócio e através



Segundo Maciaszek (2000), **requisito** é uma condição ou capacidade que deve ser alcançada ou possuída por um sistema ou componente deste para satisfazer um contrato, padrão, especificação ou outros documentos formalmente impostos.





de análise e avaliação de outros sistemas para o problema relatado.

O documento criado contendo o levantamento de requisitos é conhecido como documento de requisitos. Segundo Bezerra (2006),



**é o documento que declara os diversos tipos de requisitos do sistema. Normalmente é escrito em linguagem natural, ou seja, escrito de maneira que outro profissional poderá ler e entender.**

O autor, ainda, lembra que as principais seções de um documento de requisitos são os requisitos funcionais e os requisitos não funcionais, conforme quadro abaixo:

Requisitos Funcionais	Requisitos Não Funcionais
São os requisitos que definem as funcionalidades do sistema.	São os requisitos que declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidades
Exemplos	
O sistema deve permitir que cada professor realize o lançamento de notas das turmas na quais lecionou.	Confiabilidade: corresponde a medidas quantitativas da confiabilidade do sistema, tais como tempo médio entre falhas, recuperação de falhas ou quantidade de erros por milhares de linha de código-fonte.
O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas em um semestre.	Desempenho: requisitos que definem tempos de resposta esperados para as funcionalidades do sistema.
Os coordenadores de escola devem poder obter o número de aprovações, reprovações e trancamentos em todas as turmas em um determinado período.	Portabilidade: restrições sobre as plataformas de <i>hardware</i> e de <i>software</i> nas quais o sistema será implantado e sobre o grau de facilidade para transportar o sistema para outras plataformas.

Adaptado: Bezerra, 2006, p.26

Você deve entender que o documento de requisitos deve apresentar aos profissionais envolvidos clareza e direção que possibilitem o melhor entendimento relacionado ao negócio em questão. Note que, nesse documento, não devem ser tratadas questões de ordem técnica ou tecnológica.



**Perceba que o levantamento de requisitos é a etapa mais importante no processo de criação do projeto de sistemas, vez que a realização bem sucedida dessa atividade refletirá em um projeto de sistema capaz de solucionar o problema apontado pelo cliente. É etapa crucial que permitirá seguir ou parar com o desenvolvimento do sistema.**





## 2.2 Importância dos requisitos para o projeto de sistemas Web

Vimos, então, que a atividade de levantamento de requisitos para o projeto de sistema Web é muito importante para que o projetista possa compreender o sistema que será produzido, buscando atender todas as necessidades percebidas durante a entrevista com o cliente.

Para Pressman (2011), o amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos é denominado de *engenharia de requisitos*. De que se trata?

Na perspectiva do processo de *software*, a engenharia de requisitos é uma ação de engenharia de *software* importante que se inicia durante a atividade de comunicação e continua na modelagem. Ela deve ser adaptada às necessidades do processo de projeto, do produto e das pessoas que estão realizando o trabalho.

Os requisitos compreendem a definição documentada de uma propriedade, particularidade e/ou comportamento que um produto ou serviço particular apresenta e que deve ser solucionado com a produção do sistema.

A formulação de sistemas e aplicações baseados na Web representa uma sequência de ações de engenharia da Web que:

- começa com a identificação das necessidades do negócio;
- avança para uma descrição dos objetivos da WebApp;
- define as principais características e funções; e
- realiza a coleta de requisitos que levam ao desenvolvimento de um modelo de análise.

A formulação permite aos interessados e à equipe de engenharia da Web estabelecerem um conjunto comum de metas e objetivos para a construção da WebApp. Também identifica o escopo do esforço de desenvolvimento e fornece meios para determinar um resultado bem-sucedido (PRESSMAN, 2011).



É imprescindível que você, estudante do Curso Técnico em Informática para Internet, entenda a importância dos requisitos para o projeto de sistemas Web a fim de compreender as necessidades do cliente, podendo assim dominar a regra de negócio do sistema que será projetado.

Nesse sentido, você deve, ao iniciar um projeto de sistema Web, refletir sobre qual a necessidade para o seu desenvolvimento, pois, nesse projeto, o projetista poderá abordar e especificar de forma mais abrangente a importância do levantamento dos requisitos para o sistema.

É necessário que sejam conhecidos os objetivos que se quer alcançar com o projeto e desenvolvimento do sistema. Outra relevante reflexão que deve ser feita é sobre o perfil do usuário do sistema, ou seja, deve-se pensar como o usuário do sistema, vez que poderá assim compreender a real necessidade e poder da melhor forma atendê-la com um produto robusto que abranja da melhor forma o que é conhecido a partir do levantamento dos requisitos.

Para Pressman (2011), os atributos de sistemas e aplicações baseados na Web têm profunda influência no processo WebE que é escolhido. Um engenheiro de *software* escolhe um modelo de processo com base nos atributos do *software* que deve ser desenvolvido. O mesmo continua valendo para um engenheiro da web.

Se imediatismo e evolução contínua são atributos principais de uma WebApp, uma equipe de engenharia da Web pode escolher um modelo de processo ágil que produz versões WebApp em uma sequência rápida. Por outro lado, se uma WebApp precisar ser desenvolvida durante um período de tempo maior (por exemplo, uma grande aplicação de *e-commerce*), um modelo de processo incremental pode ser escolhido.

## 2.3 As especificações e análise de requisitos para projeto de sistemas Web

Para que você compreenda melhor a análise de requisitos, é importante observar que a análise é o primeiro passo no modelo do processo, pois ela foca em “o que devo fazer” e não na forma como o modelo será implementado. Serve como contrato entre desenvolvedor e comprador. É o processo de aquisição, refinamento e verificação das necessidades do sistema.





Nessa fase do trabalho, o objetivo é sistematizar o processo de definição dos requisitos, obtendo uma especificação correta e completa do mesmo para elaboração do documento de requisitos.

Assim, você consegue obter os requisitos do sistema de forma mais abrangente, podendo estabelecer objetivos gerais que o sistema deverá cumprir em busca da satisfação do cliente. Deverá, então, verificar a funcionalidade ou não do sistema, o que será ou não implementado, observar o que será utilizado pelos usuários finais do sistema, pelos desenvolvedores e arquitetos do sistema.

Mas, como obter os requisitos para análise?

Entender os requisitos de um problema está entre as tarefas mais difíceis enfrentadas por um engenheiro de *software*. A análise de requisitos, como todas as outras atividades de engenharia de *software*, precisa ser adaptada às necessidades do processo, do projeto, do produto e do pessoal que está fazendo o trabalho (PRESSMAN, 2011).

Esse é o grande desafio. Porém, poderá ser vencido, pois a equipe responsável pelo levantamento de requisitos pode utilizar-se das técnicas existentes como (a) a entrevista com o cliente, (b) a aplicação de questionários, (c) a observação direta, a partir das sessões (d) *brainstorming*.

Vejamos, resumidamente, cada uma delas, embora acreditemos que já tenhamos algum tipo de experiência ou conhecimento sobre elas ou alguma delas.

### Entrevista

A entrevista, utilizada para o conhecimento mais abrangente da regra de negócio, é usada quando poucas pessoas conhecem as informações necessárias para o desenvolvimento do sistema. Precisa ser preparada antecipadamente, com um rol de perguntas objetivas. É importante que as perguntas realizadas durante a entrevista com o cliente não causem qualquer constrangimento ao mesmo.



**Figura 3**  
Fonte: morguefile.com





Nessa fase, é importante planejar antecipadamente a entrevista, buscando identificar a função e a responsabilidade do entrevistado, marcar horário, escolher local onde possa ser realizada de forma confortável e satisfatória. Nessa fase, devem ser levantados os pontos relevantes para o projeto do sistema.

### Questionário

Essa técnica é usada quando o número dos que conhecem a regra do negócio do sistema é maior, ou seja, mais pessoas estão envolvidas e conhecem as necessidades para o desenvolvimento do sistema.

É fundamental que seja preparado antecipadamente com questões pertinentes aos interesses do projetista, a fim de conhecer as necessidades do cliente e também a regra de negócio do sistema de forma objetiva.



**Figura 4**  
Fonte: sxc.hu

Com essa técnica é possível identificar o tipo de informação que se deseja obter, podendo-se enfatizar a importância do levantamento para as pessoas envolvidas, analisar e consolidar as informações recebidas.

### Observação direta

O projetista utiliza a observação direta como forma para a junção, compilação e confirmação de resultados obtidos a partir da entrevista e da aplicação do questionário

A observação direta proporcionará ao projetista a identificação de documentos que devem ser coletados para posterior análise. É a partir dessa observação que o profissional poderá observar diretamente quem desenvolve o trabalho e demais necessidades pertinentes a essa fase do projeto.





**Figura 5**  
Fonte: sxc.hu

### **Brainstorming**

Nesse contexto de projeto de sistema, as tarefas como a análise de requisitos, o seu levantamento e a sua elaboração determinam com detalhes suficientes os que são de interesse do cliente para dar continuidade às atividades a serem desenvolvidas.

O *brainstorming* é útil para obter rapidamente informações sobre a real condição da necessidade e também sobre as eventuais situações que não puderam ser percebidas anteriormente.

Nessa fase, é importante a participação dos diferentes profissionais da empresa que necessitam do sistema e que atuam nos diferentes níveis de função dentro da organização, para melhor conhecimento e entendimento do sistema desejado.



**Brainstorming** – do inglês, significa “tempestade cerebral”. É uma técnica de dinâmica de grupo. Se não a conhece, veja na internet, onde há muito coisa interessante sobre ela, como em: [www.youtube.com/watch?v=XFztzGi0Kkg](http://www.youtube.com/watch?v=XFztzGi0Kkg)



**Figura 6**  
Fonte: sxc.hu



Geralmente, nesse tipo de técnica, as reflexões e discussões são feitas em grupo e devem ser conduzidas por um mediador, pois o grupo poderá levantar maior número de necessidades e possíveis soluções para o que se deve projetar.

## Resumo

A atividade de levantamento de requisitos é parte fundamental para o início do projeto de sistemas, vez que o projetista entenderá melhor as regras do negócio para o desenvolvimento do sistema.

Assim, nesta aula apresentamos os requisitos funcionais e os não funcionais que você precisa conhecer e diferenciar por serem importantes para o projetista de sistema. Além disso, abordamos a especificação e a análise de requisitos do projeto de sistemas Web, pois assim você pôde compreender e abstrair o mundo real buscando projetá-lo para o mundo computacional.

Enfim, estudou importantes aspectos e práticas voltadas para a construção e produção do projeto de sistema Web que lhe permitirá o domínio do processo de construção do projeto de sistemas Web.



## Atividade de Aprendizagem

Com base no estudo do conteúdo desta aula, você aprendeu sobre a atividade de levantamento de requisitos e especificação e análise de requisitos para o projeto de sistemas Web. Agora vamos praticar o que você já sabe. Imagine um projeto de sistema em que você está trabalhando, um sistema que permitirá a organização e catálogo de livro, de sua biblioteca pessoal.

Liste no quadro abaixo três requisitos funcionais e três requisitos não funcionais:

Requisitos Funcionais	Requisitos Não Funcionais
São os requisitos que definem as funcionalidades do sistema.	São os requisitos que declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidades
Exemplos	








Chegamos, assim, a final do estudo da segunda aula. Agora você está mais informado sobre a atividade de levantamento de requisitos e sobre a especificação e a análise de requisitos do projeto de sistemas Web.

Essa parte do conteúdo da disciplina permite que você compreenda melhor a necessidade e a importância de conhecer os tipos de requisitos e da documentação a ser formulada com base nas entrevistas realizadas com o cliente ou com os usuários do sistema.

Agora, para que consiga compreender melhor o assunto que acabou de estudar, é importante assistir às videoaulas e praticar a atividade de aprendizagem proposta.

Na próxima aula, abordaremos temas relacionados à modelagem, à validação e à verificação de requisitos.

Continue seu estudo com interesse e motivação, pois são base para a aprendizagem.



# Aula 3. Modelagem, engenharia e validação de requisitos

## Objetivos:

- compreender a importância da modelagem para o projeto de sistemas Web;
- aprender a engenharia de requisitos para o projeto de sistemas Web; e
- descrever os processos de validação dos requisitos para o projeto de sistemas Web.

Prezado(a) estudante:

Iniciamos, agora, a terceira aula da disciplina Projeto de Sistemas Web do Curso Técnico em Informática para Internet. Nessa aula, você estudará e compreenderá a importância da modelagem para os sistemas Web, aprenderá a engenharia de requisitos para o projeto de sistemas Web e conjuntamente conhecerá características relacionadas à validação dos requisitos para o projeto de sistemas Web.

Nessa aula, você adquirirá conhecimentos novos da disciplina, podendo exercitar o que já aprendeu sobre os requisitos para o projeto de sistemas. Assim, poderá vivenciar a necessidade existente no mundo real para que seja projetada para o mundo computacional.

Ao término do estudo dessa parte do conteúdo da disciplina Projeto de Sistemas para Web, você será capaz de compreender a modelagem de sistemas Web, a engenharia e a validação de requisitos para o projeto de sistemas Web e praticá-las.

Certamente, competências novas você adquirirá com o estudo dessa aula que lhe possibilitarão ser um profissional gabaritado em projetos de sistemas Web.



Excelente estudo!

### 3.1 Importância da modelagem para projeto de sistemas Web

Uma visão de modelagem de requisitos considera os dados e os processos que os transformam como entidades distintas.

Os *objetos de dados* são modelados de maneira a definir seus atributos e relacionamentos. A análise de requisitos de WebApps engloba três tarefas principais: formulação, coleta de requisitos e modelagem de análise. Durante a formulação, a motivação básica (metas) e os objetivos para a WebApp são identificados e as categorias de usuários definidas (PRESSMAN, 2011).

A modelagem de sistemas, conforme Motamarri (1992), é um elemento importante do processo de engenharia de sistemas. Quer o foco esteja na visão de mundo ou em uma visão detalhada, o engenheiro cria modelos que:

- definem os processos, servindo as necessidades da visão que esta sendo considerada;
- representam o comportamento dos processos e os pressupostos nos quais o comportamento está baseado;
- definem explicitamente tanto entradas exógenas quanto endógenas para o modelo; e
- representam todas as ligações (inclusive saídas), que permitirão ao engenheiro entender melhor a visão.

A análise de requisitos fornece o mecanismo apropriado para entender o que o cliente deseja, analisando as necessidades, avaliando a exequibilidade, negociando uma condição razoável, especificando a solução de modo não ambíguo, validando a especificação e gerindo os requisitos à medida que eles são transformados em um sistema (THAYER, 1997).



**Para que entenda melhor, a modelagem de requisitos é a representação do que se pode abstrair do mundo real e que permite descrever ou mesmo de alguma forma prever comportamentos específicos de um sistema por meio da observação de certas características aparen-**



## tes e importantes.

Para Pressman (2011), a modelagem de análise leva à obtenção de cada um dos elementos. Entretanto, o conteúdo específico de cada elemento (os diagramas usados para construir o elemento e o modelo) pode diferir de projeto para projeto. Observe a figura abaixo.



**Figura 7**

Fonte: adaptado de Pressman (2011)

A figura acima apresenta o problema segundo um ponto de vista.

Os elementos baseados em cenários representam como o usuário interage com o sistema e a sequência específica de atividades que ocorre à medida que o *software* é utilizado.

Os elementos baseados em classes modelam os objetos que o sistema irá manipular, as operações que serão aplicadas aos objetos para efetuar a manipulação, os relacionamentos (alguns hierárquicos) entre os objetos e as colaborações que ocorrem entre as classes definidas.

Os elementos comportamentais representam como eventos externos mudam o estado ou as classes que neles residem.

Por fim, elementos orientados a fluxos representam o sistema como uma transformação de informações, indicando como os objetos de dados são transformados à medida que fluem pela várias funções do sistema (Press-



man, 2011).

O autor aborda, ainda, que existem visões diferenciadas para a modelagem de requisitos:

- Análise estruturada - Considera os dados e os processos que transformam os dados em entidades separadas. Os objetos de dados são modelados de maneira que definam seus atributos e relacionamentos.
- Análise orientada a objetos - Concentra-se na definição de classes e na maneira pela qual colaboram entre si para atender às necessidades dos clientes.

## 3.2 Engenharia de requisitos para o projeto de sistemas Web

É importante notar que algumas dessas funções de engenharia de requisitos ocorrem em paralelo e que todas são adaptadas às necessidades do projeto. Todas tentam definir o que o cliente deseja e servem para estabelecer uma fundação sólida para o projeto e a construção do que o cliente obtém (PRESSMAN, 2011).

Para que entenda melhor a engenharia de requisitos, imagine um grupo de diferentes profissionais trabalhando simultaneamente em um projeto, ou seja, cada qual com sua função a partir do conhecimento e tarefas que pode desempenhar.

A engenharia de requisitos, como todas as outras atividades de engenharia de *software*, precisa ser adaptada às necessidades do processo, do projeto e do pessoal que está fazendo o trabalho.

Na perspectiva do processo de *software*, a engenharia de requisitos (ER) é uma ação de engenharia de *software* que começa durante a atividade de comunicação e continua durante a atividade de modelagem (PRESSMAN, 2011).

O autor enfatiza que, embora seja uma boa ideia analisar o problema antes de começar o projeto, não é verdade que uma análise completa deve preceder todo o projeto. De fato, o projeto de uma parte específica da WebApp exige apenas uma análise daqueles requisitos que afetem apenas essa parte



da Webapp.

Observe que o grau com o qual a modelagem de requisitos para WebApps é enfatizado depende de fatores como:

- tamanho e complexidade do incremento de WebApp;
- número de interessados (a análise pode ajudar a identificar requisitos conflitantes provenientes de várias fontes);
- tamanho da equipe de desenvolvimento de WebApp;
- nível em que os membros da equipe de desenvolvimento de WebApps trabalharam juntos antes (a análise pode ajudar a desenvolver um entendimento comum do projeto); e
- nível de êxito da organização sendo diretamente dependente do êxito das WebApps.

Em outras palavras, para projetos de sistemas Web, também podemos aplicar tal grau mencionado por Pressman (2011).

Então, você deve refletir sobre como é importante continuar seus estudos sobre modelagem de requisitos para projeto de sistemas Web, vez que existem critérios diferentes e várias formas para análise e construção de projetos quando se trata de projetos de sistemas Web.

### 3.3 Validação dos requisitos para o projeto de sistemas Web

Os produtos de trabalho resultantes da engenharia de requisitos são válidos quanto à qualidade durante o passo de validação. A validação de requisitos examina a especificação para garantir que todos os requisitos do *software* tenham sido declarados de modo não ambíguo; que as inconsistências, omissões e erros tenham sido detectados e corrigidos e que os produtos de trabalho estejam de acordo com as normas estabelecidas para o processo, o projeto e o produto (PRESSMAN, 2011).

Você pode estar pensando que a atividade para o levantamento de requisitos parece muito simples, contudo deve observar algumas peculiaridades que





aparecem durante essa prática, pois não é uma tarefa fácil não.

O interesse deve ser sobre quais são os objetivos do sistema ou produto em que se está trabalhando. Tais questionamentos devem ser feitos ao cliente, aos usuários que estarão direta ou indiretamente envolvidos ou utilizando o sistema.



**É essencial que, nessa atividade de levantamento de requisitos, sejam compreendidos aspectos tais como o questionamento quanto ao fato de o sistema poder atender as necessidades observadas ou mesmo quanto à forma como o sistema será utilizado no cotidiano após sua construção.**

Christel e Kang (1992) identificam vários problemas que nos ajudam a compreender porque o levantamento de requisitos é difícil.

- **Problema de escopo** - O limite do sistema é mal definido ou o cliente/usuário especifica detalhes técnicos desnecessários que podem confundir, em vez de esclarecer os objetivos globais do sistema.
- **Problema de entendimento** - O cliente/usuário não está completamente certo do que é necessário, tem pouca compreensão das capacidades e limitações de seu ambiente computacional, não tem pleno entendimento do domínio do problema, tem dificuldade de informar as necessidades ao engenheiro de sistemas, omite informações que acredita ser “óbvias”, especifica requisitos conflitantes com as necessidades de outros clientes/usuários ou especifica requisitos que são ambíguos ou impossíveis de testar.
- **Problema de volatilidade** - Os requisitos mudam ao longo do tempo.

Por isso, o autor enfatiza que os profissionais envolvidos nessa fase do projeto devem abordar a atividade de coleta dos requisitos de um modo organizado.

## Resumo

A modelagem para o projeto de sistemas Web é parte muito complexa, vez que podemos observar diferentes visões a partir da observação dos diferentes profissionais envolvidos no projeto.







O trabalho torna-se complexo pela possibilidade de diferentes abstrações dos requisitos levantados durante a fase de levantamento dos requisitos, sejam estes funcionais ou não funcionais.

Quanto à validação dos requisitos, tal tarefa irá refletir-se diretamente no desenvolvimento do projeto, pois isso propõe a mutação das necessidades conhecidas e necessárias para o referido desenvolvimento.

Os estudos sobre essas diferentes atividades tornarão ainda mais amplo seu conhecimento acerca da importância e das peculiaridades que envolvem o projeto do sistema Web e permitirá que você aprofunde ainda mais seu conhecimento sobre as diversas tecnologias para a elaboração de projetos.

## Atividade de Aprendizagem



Nesta aula, você estudou sobre a modelagem para o projeto de sistemas Web, que é uma fase muito complexa por apresentar diferentes visões a partir da observação dos diferentes profissionais envolvidos no projeto. Com base nesse conhecimento, cite as visões diferenciadas para modelagem no entendimento de Pressman (2011) e que foi apresentada nessa aula.

---

---

---

---

---

Você conclui agora a terceira aula da disciplina Projeto de Sistemas WEB. Pôde estudar a importância da modelagem de sistemas Web; aprendeu sobre engenharia de requisitos para o projeto de sistemas Web e conjuntamente conheceu características relacionadas à validação dos requisitos para o projeto de sistemas Web.

Continue seus estudos, busque sempre ler outras referências para aprimorar seu conhecimento acerca da modelagem, engenharia e validação de requisitos para melhor desenvolver suas atividades relacionadas ao projeto de sistemas Web.

Vamos, então, para a próxima aula, a quarta da disciplina. Nessa aula, você estudará as metodologias e ferramentas para modelagem de sistemas Web.





# Aula 4. Metodologias e ferramentas para modelagem de sistemas Web

## Objetivos:

- reconhecer as metodologias utilizadas em projeto de sistema Web;
- identificar diferentes modelos de análise de projeto de sistemas Web; e
- analisar os tipos de modelos para a modelagem de sistemas Web.

Nessa aula, você começa a estudar a respeito de modelagem e tipos de modelos para modelagem. É importante que tenha estudado, apreendido os temas das aulas anteriores e tenha praticado as atividades propostas para que melhor possa aproveitar o conteúdo dessa aula e da última.

É certo que, ao final dessa aula, com a compreensão e prática do conteúdo, você será capaz de praticar e discutir sobre os modelos de análise para modelagem de sistemas Web. É uma oportunidade para a progressão de seu aprimoramento profissional.

Ótimo estudo!

## 4.1 Metodologias para projetos de sistemas Web

Nessa aula, você irá aprender conceitos referentes às **metodologias**, aos **métodos** para a construção de projetos de sistema Web. Mas, o que é um método?

Nesse sentido, você deve entender quais são os passos que você dará de forma progressiva para a construção do projeto de sistemas, ou seja, como você, observando os requisitos já conhecidos durante a atividade de levan-

### A-Z

Para Lakatos e Marconi (2010), **método** é um conjunto das atividades sistemáticas e racionais que, com maior segurança e economia, permite alcançar o objetivo.

A **metodologia** é a lógica dos procedimentos científicos em sua gênese e em seu desenvolvimento, não se reduzindo, portanto, a uma metrologia ou tecnologia da medida dos fatos científicos.



tamentos dos mesmos, irá traçar uma rotina para a construção de uma possível condição para as necessidades elencadas.

Uma metodologia deve abordar as ciências sob o ângulo do produto delas, como resultado em forma de conhecimento científico, mas também como processo, isto é, como gênese desse próprio conhecimento. (BRUYNE et al., 1977)

Assim, você deve entender a metodologia como o caminho a ser trilhado, de modo que, com o pensamento construído, você possa agregar o conjunto de métodos capazes de serem empregados com a finalidade de produzir conhecimento a partir de estudos científicos.



**O panorama de métodos da Web engloba um conjunto de tarefas técnicas que habilitam um engenheiro Web a entender, caracterizar e então construir uma WebApp de alta qualidade.**

Os métodos Web podem ser categorizados da seguinte maneira, segundo Pressman (2011):

- **Métodos de comunicação** - Definem a abordagem usada para facilitar a comunicação entre os engenheiros Web e todos os outros interessados na WebApp (por exemplo, usuários finais, clientes de negócio, especialistas no domínio do problema, projetistas de conteúdo, líderes de equipe, gerentes de projeto). As técnicas de comunicação são particularmente importantes durante a coleta de requisitos e sempre que um incremento da WebApp precise ser avaliado.
- **Métodos de análise de requisitos** - Fornecem uma base para o entendimento do conteúdo a ser entregue por uma WebApp, a função a ser fornecida para o usuário final e os modos de interação que cada classe de usuários irá requerer à medida que a navegação pela WebApp ocorre.
- **Métodos de projeto** - Abrangem uma série de técnicas de projeto que cuidam do conteúdo, arquitetura da aplicação e da informação, projeto da interface e estrutura de navegação da WebApp.
- **Métodos de teste** - Incorporam revisões técnicas formais do conteúdo e modelo de projeto e uma ampla variedade de técnicas de teste, que tratam de tópicos no nível de componente e arquitetural, testes de navega-



ção, testes de usabilidade, testes de segurança e testes de configuração.

Embora métodos de Web adotem muitos dos mesmos conceitos e princípios subjacentes aos métodos de engenharia de *software*, os mecanismos de análise, projeto e teste devem ser adaptados para acomodar as características especiais das WebApps.

É importante que você compreenda que, na atividade de construção de um projeto de sistemas Web, você precisa seguir uma série de passos possíveis, ou seja, aqueles passos definidos a partir do conhecimento do problema.

Assim, a criação do projeto será produtiva e o resultado final um sucesso. Será entregue no prazo estabelecido, com as características referentes à regra de negócio e com qualidade.

## 4.2 Modelo de análise de projeto de sistemas Web

Um modelo de análise de WebApp é guiado por informação contida nos casos de uso desenvolvidos para a aplicação. Descrições de caso de uso são analisadas gramaticalmente para identificar classes potenciais de análise e operações e atributos associados com cada classe. O conteúdo a ser apresentado pela WebApp é identificado e funções a serem realizadas são extraídas das descrições de casos de uso (PRESSMAN, 2011).

- Análise funcional define as operações que serão aplicadas ao conteúdo da WebApp e descreve outras funções de processamento independentes do conteúdo, mas necessárias para o usuário final.
- Análise de configuração descreve o ambiente e a infraestrutura nos quais a WebApp reside.
- A informação coletada durante essas quatro tarefas de análise deve ser revisada, modificada quando for necessário e então organizada em um modelo que pode ser passado aos projetistas da WebApp.

Para o referido autor, o modelo em si contém elementos estruturais e dinâmicos. O que são?

### A-Z

**Elementos estruturais** identificam as classes de análise e objetos de conteúdo necessários para criar uma WebApp que atenda às necessidades dos interessados.

Os **elementos dinâmicos** do modelo de análise descrevem como os elementos estruturais interagem entre si e com os usuários finais.





Assim, após o desenvolvimento do sistema será percebida ou não a qualidade da construção do projeto. Para Pressman (2011), WebApps bem-sucedidas permitem aos clientes satisfazerem as suas necessidades por si mesmos, melhor e de forma mais rápida.

A partir do estudo da complexidade e dinamicidade da plataforma Web, reflita um pouco sobre a vasta gama de ferramentas e sobre a tecnologia existente e sua constante evolução.

Os sistemas Web têm-se tornado mais sofisticados e difundidos, acompanhando as necessidades da sociedade atual. Tais tecnologias englobam grande variedade de descrição de conteúdo e linguagens de modelagem (por exemplo: HTML, UML, XML): linguagens de programação (como Java, PHP, C, C++), recursos de desenvolvimento baseado em componentes, navegadores/*browsers*, ferramentas multimídia, ferramentas de segurança, de auditorias, de gestão e análise de *sites* dentre outros.

### 4.3 Tipos de modelos para a modelagem de sistemas Web

Certamente, você já compreendeu bem os modelos de análise para projeto de sistemas Web. Assim, nesse tópico da quarta aula, apresentaremos conceitos sobre os tipos de modelos para modelagem de sistemas Web descritos por Pressman, (2011).

Aprenderá conceitos introdutórios sobre: o modelo de conteúdo, modelo de interação, modelo funcional e por fim o modelo de configuração. Fique atento para as diferenças e aplicabilidade para cada tipo de modelos apresentados a seguir.

#### 4.3.1 O modelo de conteúdo

Contém elementos estruturais que fornecem visão importante dos requisitos de conteúdo para uma WebApp. Esses elementos estruturais englobam objetos de conteúdo (como texto, imagens gráficas, fotografias, imagens de vídeo, áudio) apresentados como partes da WebApp.

Além disso, o modelo de conteúdo inclui todas as classes de análise - entidades visíveis ao usuário criadas ou manipuladas à medida que um usuário interage com a WebApp.



Uma classe de análise engloba atributos que a descrevem, operações que realizam o comportamento necessário para a classe e colaborações que permitem que a classe se comunique com outras classes.

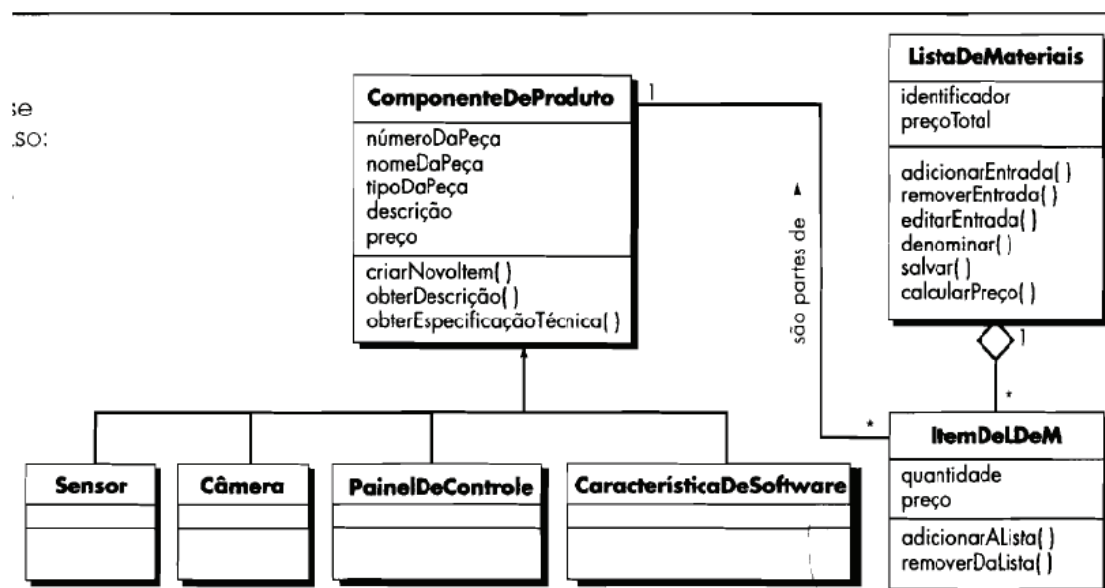
**Como outros elementos do modelo de análise, o modelo de conteúdo é derivado de um exame cuidadoso dos casos de uso desenvolvidos para a WebApp. Casos de uso são analisados sintaticamente para extrair os objetos de conteúdo e as classes de análise**



### 4.3.2 Modelo de interação

É composto de quatro elementos: (1) casos de uso, (2) diagramas de sequência, (3) diagramas de estados e (4) um protótipo de interface com o usuário. Em adição a essas representações, a interação também é representada no contexto do modelo navegacional.

*Casos de uso* - São o elemento dominante do modelo de interação para WebApps. Não é incomum descrever 100 ou mais casos de uso quando WebApps são grandes e complexas (Figura abaixo).



**Figura 8**

Fonte: Pressman, 2011.

*Diagramas de sequência* - Diagramas de sequência UML fornecem representação abreviada da maneira pela qual ações de usuário (os elementos dinâmicos de um sistema definido por casos de USO) colaboram com classes



de análise (os elementos estruturais de um sistema definido por diagramas de classe).

Como as classes de análise são extraídas das descrições de casos de uso, há necessidade de garantir que exista rastreabilidade entre as classes que foram definidas e os casos de uso que descrevem as interações do sistema.

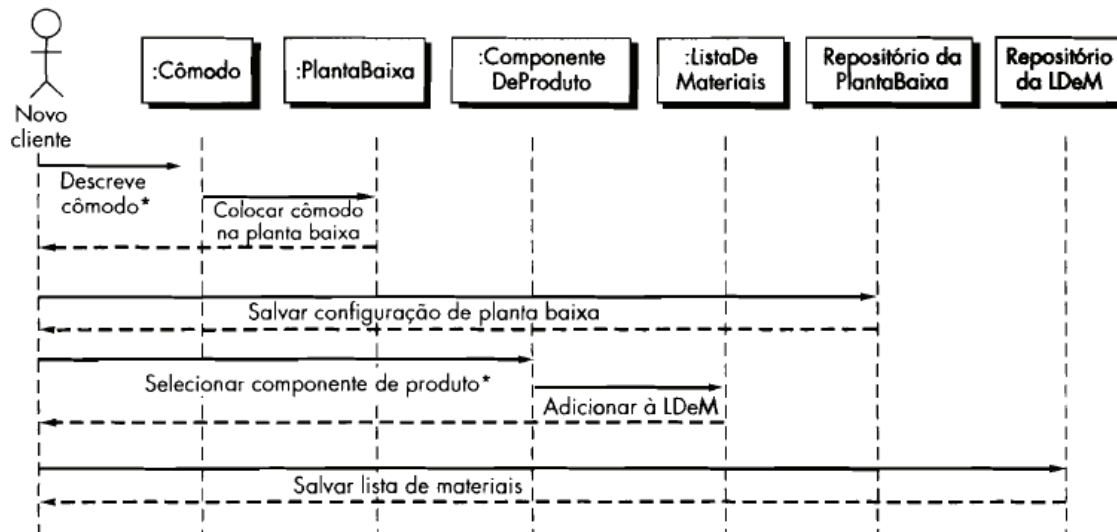


Figura 9

Fonte: Pressman, 2011.

### 4.3.3 O modelo funcional

Atende a dois elementos de processamento da WebApp, cada um representando um diferente nível de abstração procedimental: (1) funcionalidade observável pelo usuário que é entregue pela WebApp aos usuários finais, e (2) as operações contidas nas classes de análise que implementam comportamentos associados com a classe.

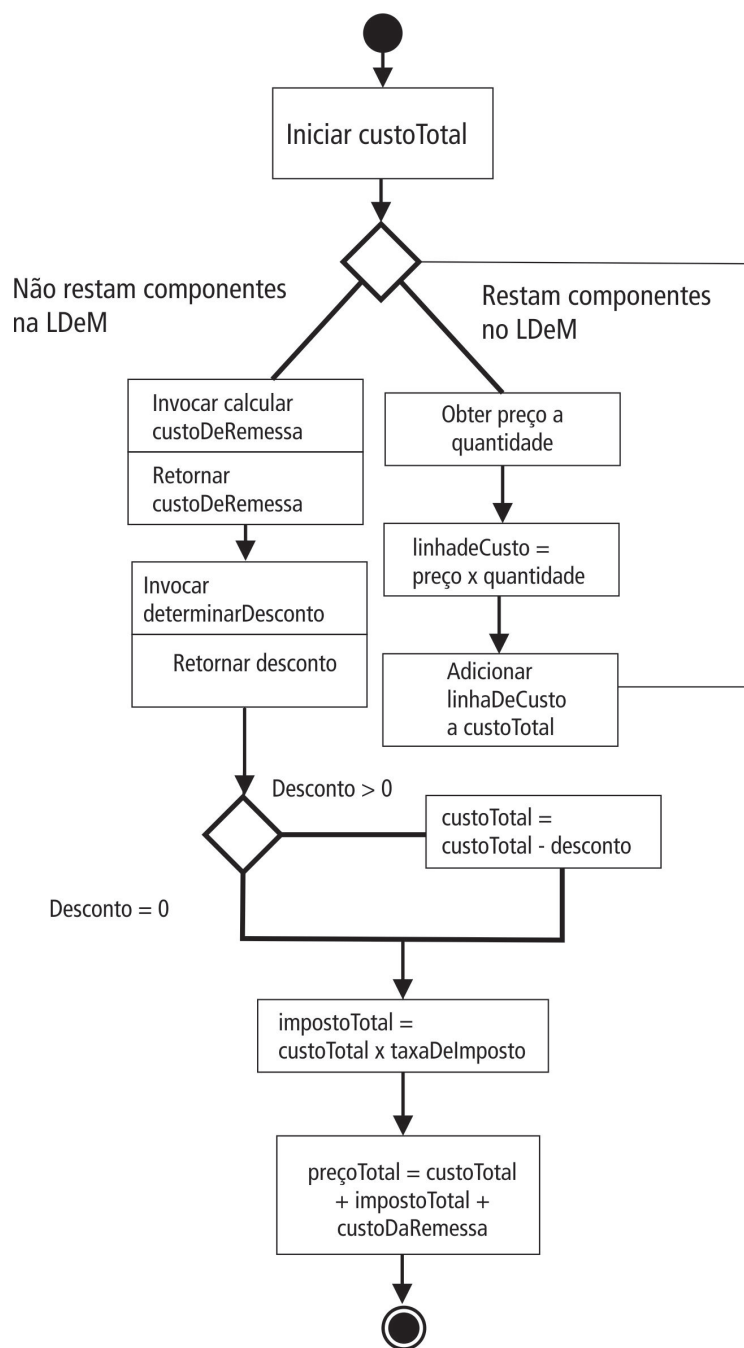
A *funcionalidade observável* pelo usuário engloba quaisquer funções de processamento que são iniciadas diretamente pelo usuário. Por exemplo, um *site* financeiro da Web poderia implementar uma variedade de funções financeiras (por exemplo, um calculador de economia nos pagamentos de universidade ou um calculador de economias de aposentadoria).

Essas funções podem ser realmente implementadas, usando-se operações dentro de classes de análise, mas, do ponto de vista do usuário final, a função (mais corretamente, os dados fornecidos pela função) é a saída visível.





Em um nível mais baixo de abstração procedimental, o modelo de análise descreve o processamento a ser realizado pelas operações da classe de análise. Essas operações manipulam atributos de classe e são envolvidas à medida que as classes colaboram umas com as outras para conseguir algum comportamento necessário.



**Figura 10**  
Fonte: Pressman, 2011.



### 4.3.4 Modelo de configuração

WebApps devem ser projetadas e implementadas de modo que acomodem uma variedade de ambientes tanto do lado do servidor quanto do lado do cliente. A WebApp pode residir em um servidor que fornece acesso via internet, intranet ou extranet.

O *hardware* do servidor e o ambiente do sistema operacional devem ser especificados. Além disso, considerações de interoperabilidade do lado do servidor devem ser levadas em conta. Se a WebApp precisar ter acesso a um grande banco de dados ou interoperar com aplicações da empresa que existem no lado do servidor, interfaces apropriadas, protocolos de comunicação e informação colaborativa relacionada devem ser especificadas.

O *software* do lado do cliente fornece a infraestrutura que possibilita acesso à WebApp a partir da localização do usuário. Em geral, o *software* de navegação é usado para entregar o conteúdo e a funcionalidade da WebApp que é baixada do servidor.



**Embora existam normas, cada navegador tem suas próprias peculiaridades. Por essa razão, a WebApp deve ser rigorosamente testada com cada configuração de navegador que é especificada como parte do modelo de configuração.**

Os conceitos e os exemplos trazidos acima foram estudados e testados pelo autor Pressman, durante seus vários anos em atuação e estudos na área de engenharia de *software*. Esperamos que você os tenha compreendidos.

### Resumo

Parabéns! Aqui você finaliza os estudos da quarta aula da disciplina Projeto de Sistemas Web que versou sobre as metodologias para construção de projetos Web, os modelos de análise de sistemas Web e os tipos de modelos para a modelagem de sistemas Web.

É importante que estude outras referências e pratique as atividades propostas ao final dessa aula, para que possa apreender os temas aqui abordados.

Estamos certos de que, ao final dessa aula, você será capaz de praticar e discutir o conteúdo tratado até essa fase de seus estudos.





## Atividade de Aprendizagem



Com base no seu conhecimento anterior e o novo apreendido ao longo da disciplina, disserte sobre a diferença entre os tipos de modelos para modelagem de sistema propostos por Pressman (2011) e relacionados nessa aula.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Finalizamos a quarta aula que tratou das metodologias e ferramentas para modelagem de sistemas Web e da importância da modelagem e utilização de ferramentas adequadas para construção de projetos de Sistemas Web.

Com o conhecimento adquirido sobre tais metodologias e ferramentas, conseguirá agregar qualidade aos trabalhos relacionados ao projeto de sistemas Web com base nas competências desenvolvidas até essa parte de seus estudos.

Note que está na reta final para a conclusão da disciplina. Portanto, é importante que, além da leitura do material e da prática das atividades propostas, você busque compartilhar e interagir com seus colegas de curso.

Na aula seguinte, você estudará a Utilização da Linguagem de Modelagem Unificada para Projetos de Sistema Web. Assim, poderá finalizar o estudo da disciplina com ótimo conhecimento acerca de projetos de sistemas Web.

Bom estudo!





# Aula 5. Utilização da linguagem de modelagem unificada para projetos de sistema Web

## Objetivos:

- saber explicar a notação da linguagem de modelagem unificada;
- identificar diferentes diagramas da UML;
- reconhecer as partes que compõem a UML; e
- praticar a utilização da linguagem para o projeto de sistemas Web.

Prezado(a) estudante:

Iniciamos agora a última aula da disciplina Projeto de Sistemas Web do Curso Técnico em Informática para Internet. Trataremos da notação da linguagem de modelagem unificada. Você aprenderá sobre os diagramas da UML, conhecerá as partes que compõem a UML e os itens de interação, ou seja, construirá um sólido conhecimento acerca da linguagem.

Finalizando o estudo do conteúdo e a prática das atividades propostas desse caderno, você estará apto a realizar atividade técnicas que envolvam o projeto de sistemas Web.

Mas, o que é a UML?

A UML pode ser utilizada nas diferentes fases no desenvolvimento de sistemas de *software*, como na análise de requisitos, análise, *design* (projeto), programação e testes. É uma linguagem padrão para a estruturação de projetos de *software*. Sua abrangência vai desde a modelagem de sistemas de informação corporativos a serem distribuídos a aplicações baseadas em Web, até sistemas complexos embutidos de tempo real.

## A-Z

A linguagem de modelagem unificada foi desenvolvida por Grady Boock, James Rumbaugh e Ivar Jacobson, conhecidos como os três amigos. “É um padrão gráfico para visualizar, especificar, construir e documentar os artefatos de um sistema de *software*”. (BOOCH et al., 1998), permitindo representar um sistema de maneira padronizada.



Antes de continuar a leitura desse material, é indicado que você acesse o *link* da página “Começando com UML” na URL <http://www.uml.org/>. Aproveite para interagir com seus colegas de curso e professores durante as teleaulas a partir do que conhecerá com o acesso a página indicada.

A primeira versão da linguagem de modelagem unificada foi lançada em 1996. Em 1997, foi adotada pela a OMG (*Object Management Group* – Grupo de Gerenciamento de Objetos) como linguagem padrão de modelagem.

É importante que você compreenda que a linguagem permite a modelagem do sistema sob diferentes perspectivas. Cada uma destas perspectivas é uma abstração apresentada por diagramas criados a partir dos recursos oferecidos pela linguagem de modelagem.

## 7.1 Aprendendo sobre a UML

Para Larman (2011), a UML é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de orientação a objeto, linguagem padrão de modelagem de *software* adotada internacionalmente pela indústria de engenharia de *software*.

Um diagrama em UML é um conjunto de itens e relacionamentos.



A UML é uma linguagem visual universalmente utilizada para o desenvolvimento de *software*, tanto em nível acadêmico quanto industrial. Um mesmo diagrama pode ser utilizado nas fases de análise e projeto, mudando-se apenas sua visão (GUEDES, 2011, pg. 43).

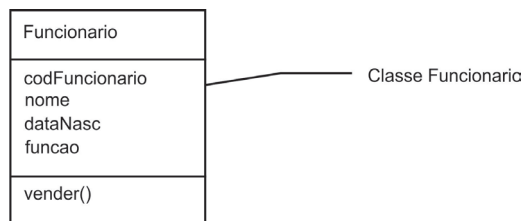
Em UML, a criação de diagramas envolve a identificação de itens que formam o vocabulário do sistema e a especificação de como estes itens relacionam-se entre si.

A linguagem apresenta itens estruturais, as partes mais estáticas do modelo e representam elementos conceituais ou físicos.

UML não é uma metodologia. Não diz quem deve fazer o quê, quando e como fazer. UML não é uma linguagem de programação (LARMAN, 2011, pg. 37).

### São itens estruturais:

*Classes* - É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamento e semântica. É uma abstração do mundo real para ser construído e manipulado no mundo computacional.



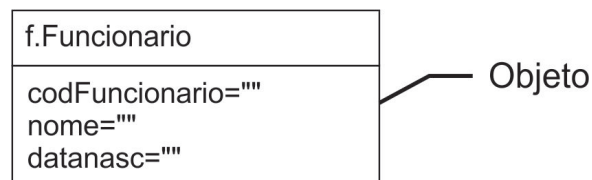
**Figura 11**  
Fonte: autores





A notação é a forma para representar no mundo computacional que no sistema deve haver organização dos dados do funcionário, para que seja possível pesquisa, exclusão, inserção dentre outras ações com tais dados no sistema. Observe a figura acima “classe funcionário”.

*Objetos* - Um objeto ou instância é uma ocorrência de uma classe, ou seja, um objeto possui estado de comportamento específico e uma identidade única dentro do contexto de uma classe.



**Figura 12**  
Fonte: autores

Pense assim: um objeto na UML pode ser entendido como elemento que se pode criar, manipular e cujo comportamento pode ser percebido, ou seja, existe uma dada interação.

O objeto na linguagem satisfaz a intenção de representar uma coisa que terá certo significado para o projeto e desenvolvimento do sistema, ou seja, é algo/coisa que faz sentido para o projeto a partir da observação do projetista.

Um **caso de uso** é utilizado para estruturar o comportamento de um sistema sem precisar especificar sua implementação, além de envolver a interação entre os atores, humanos ou sistemas automatizados

Na linguagem, são utilizados itens que são capazes de demonstrar o comportamento no tempo e espaço no projeto de sistema. É uma possibilidade de o projetista observar como será o comportamento e/ou interação entre as classes e os objetos.

Os *Itens Comportamentais* da UML são as partes dinâmicas de um modelo em UML e representam comportamentos no tempo e espaço, permitindo assim aos profissionais simulações a partir de observações modeladas com a linguagem UML.

Perceba que são dois os itens comportamentais apresentados pela linguagem de modelagem unificada, que você aprenderá agora. Leia com atenção



É importante que você saiba que uma classe obrigatoriamente terá que conter um nome, mas não necessariamente deverá possuir atributos ou operações



## A-Z

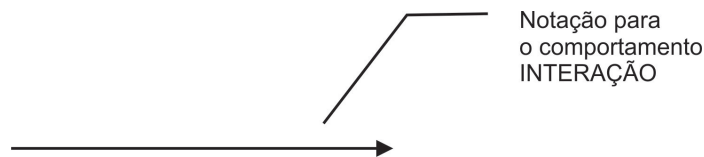
**Caso de uso** - Representa um conjunto de ações realizadas pelo sistema que geram resultados observáveis por um ator.





para perceber a importância desses itens na representação do comportamento dos elementos no projeto de sistema.

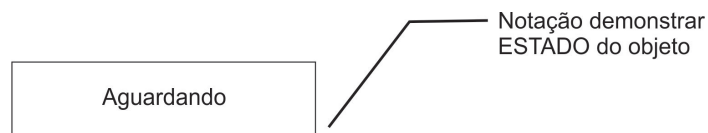
**Interações** - Uma interação representa um conjunto de mensagens que são trocadas por um conjunto de objetos de um sistema, apresentando, desta forma, o comportamento destes objetos sob um contexto específico. Uma mensagem é representada graficamente por uma seta cheia, que geralmente inclui o nome de suas operações:



**Figura 13**  
Fonte: autores

**Máquina de estados** - representa a sequência de estados que um objeto assume durante o seu tempo de vida, em resposta a eventos. Desta forma, uma máquina de estados modela o comportamento de um objeto individual.

Você deve entender que um estado é uma condição ou situação na vida de um objeto durante a qual ele satisfaz alguma condição, executa alguma atividade ou aguarda um evento (Figura abaixo).



**Figura 14**  
Fonte: autores

Com a utilização da UML, o projetista pode representar os relacionamentos existentes entre os itens que ele pode relacionar a partir do levantamento de requisitos, ou seja, pode demonstrar os relacionamentos que têm a função e mostrar como os itens abstraídos de um sistema se combinam.

A linguagem proporciona a notação dos três principais tipos de relacionamento em UML, vejamos:

**Dependência** - É a relação existente entre dois itens quando um dos itens depende do outro. Na relação de dependência, existe um item dependente e outro independente. Desta forma, uma alteração feita no item independente pode afetar a semântica do item dependente.





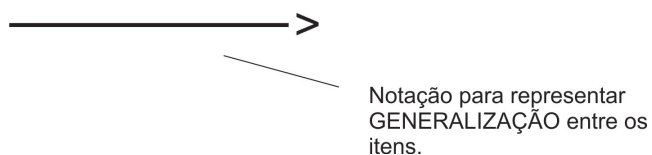
Uma dependência é representada graficamente por uma seta aberta, com linha tracejada:



**Figura 15**  
Fonte: autores

**Generalização** - É um relacionamento hierárquico entre classes. A maior classe de maior nível hierárquico é conhecida como superclasse, enquanto a outra classe da relação é conhecida como subclasse.

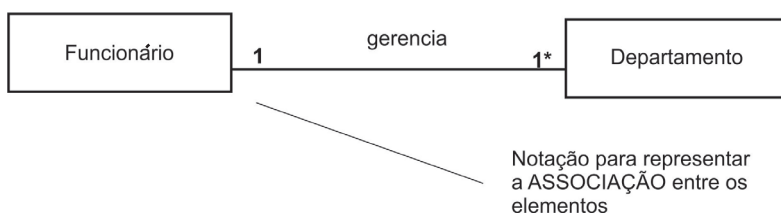
Graficamente, uma generalização é representada por uma linha contínua com uma seta em branco apontado sempre a superclasse.



**Figura 16**  
Fonte: autores

A generalização é conhecida com um relacionamento “é um tipo de”, ou seja, permite a comparação de duas classes sob a perspectiva de que a sub-classe envolvida no relacionamento é um tipo da superclasse relacionada.

**Associação** - É um relacionamento estrutural que conecta classes, ou seja, por meio de uma associação entre classes, pode-se chegar em um objeto de uma classe a partir do objeto da outra classe relacionada.



**Figura 17**  
Fonte: autores



## 7.2 Partes que compõem a UML

Como vimos, a UML é usada para expressar e comunicar ideias, ou seja, é uma forma utilizada a partir de símbolos para que a ideia do negócio do mundo real seja projetada para o mundo computacional.



É importante que você lembre que a UML não é uma metodologia, é apenas uma linguagem que permite a descrição de um sistema em um alto nível de abstração, ou seja, de forma a permitir identificar os reais componentes e ações.

Precisa, então, compreender que a UML é composta por partes, isto é, por termos utilizados para demonstrar a percepção da realidade pelo projetista a partir da entrevista e levantamento dos requisitos do sistema a ser projetado.

Quais são as partes que compõem a UML?

- **Atores** - O ator é o elemento que interage com o sistema a ser desenvolvido, classe usada para modelar o papel do usuário no sistema. Pode ser primário ou secundário tendo como símbolo, o *stick man*.



**Figura 18**  
Fonte: autores

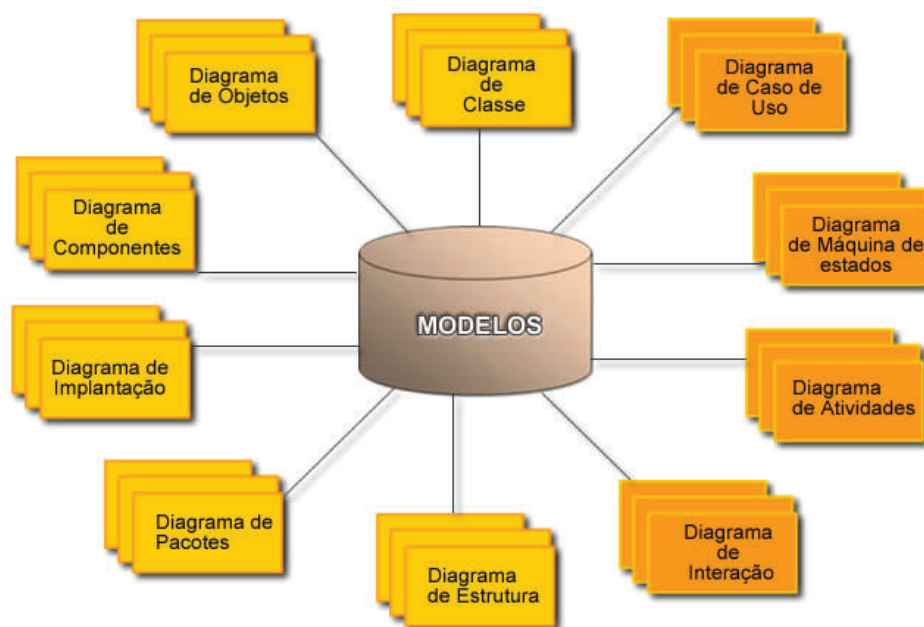
- **Primário** - É o usuário direto do sistema.
- **Secundário** - São o administrador e os beneficiários do sistema, outros sistemas que se comunicam.
- **Visões** - Mostram diferentes aspectos do sistema que está sendo modelado. É uma abstração consistindo em uma série de diagramas.
- **Modelos de elementos** - Os conceitos usados nos diagramas são modelos de elementos que representam definições comuns da orientação a objetos com as classes incluindo associações, dependências e heranças.
- **Mecanismos gerais** - Promovem comentários suplementares, informações ou semântica sobre os elementos que compõem os modelos.

- **Diagramas** - São os gráficos que descrevem o conteúdo em uma visão. A UML possui nove tipos de diagramas que são usados em combinação para prover todos as visões do sistema.

## 7.3 Diagramas da UML

Na UML, os diagramas são recursos gráficos utilizados para se visualizar um sistema sob diferentes perspectivas, ou seja, é uma forma de observar o comportamento e a composição das partes que compõem o projeto do sistema a ser desenvolvido. Você deve entender que os diagramas são geralmente compostos por itens e relacionamento.

Por padrão, a UML oferece nove tipos diferentes de diagramas que podem ser classificados como os diagramas estruturais e/ou diagramas comportamentais. Observe a figura a seguir.



**Figura 19**

Fonte: <http://www.infoescola.com/wp-content/uploads/2010/03/DiagramasdaUML.jpg>

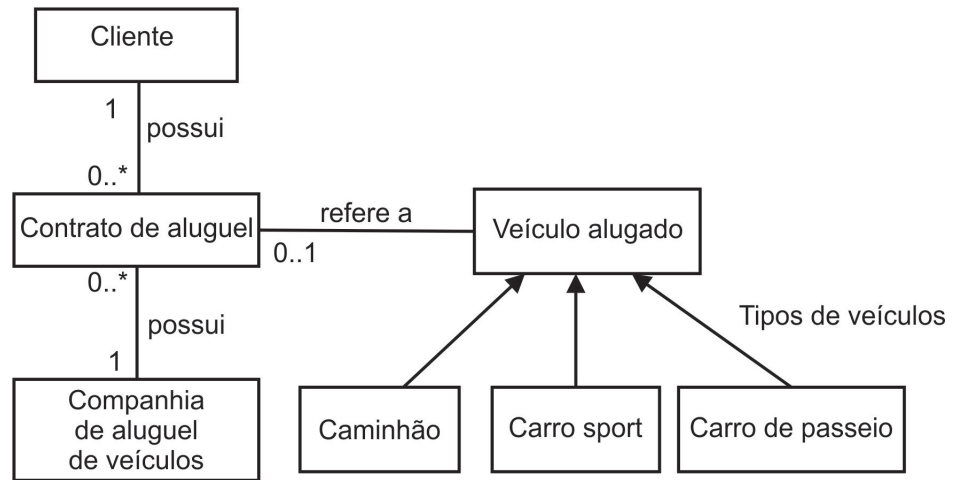
Entenda que cada diagrama é utilizado para especificar e conhecer os diferentes tipos de comportamento que as classes podem possuir e também seus relacionamentos e ações, conforme regra do negócio.

*Diagramas Estruturais* - Em UML, os aspectos estáticos de um sistema, ou seja, a representação de seu esqueleto e estrutura, relativamente estáveis,



são visualizados através dos diagramas estruturais.

*Diagrama de classes* - São os diagramas mais utilizados em sistemas de modelagem orientados a objetos. O diagrama de classe é composto basicamente por um conjunto de classes relacionais entre si. Este diagrama modela os aspectos estáticos de um sistema pelo fato de sua estrutura ser sempre válida em qualquer ponto do ciclo de vida do sistema.

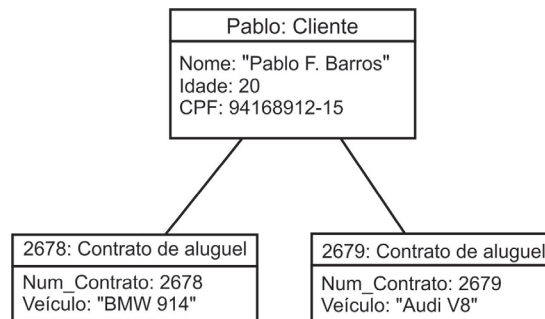


**Figura 20**

Fonte: adaptado de Larman, 2011.

Observe a figura acima e perceba que o diagrama de classes demonstra a estrutura estática das classes de um sistema no qual estas representam as "coisas" que são gerenciadas pela aplicação modelada.

*Diagrama de Objetos* - É um diagrama de classes instanciado, ou seja, o diagrama de objetos modela instâncias dos itens de um diagrama de classes e seus respectivos relacionamentos. Desta forma, os itens do diagrama de objetos assumem valores, correspondentes a seus atributos, que no seu conjunto representam a situação estática do sistema em um determinado ponto de tempo.



**Figura 21**

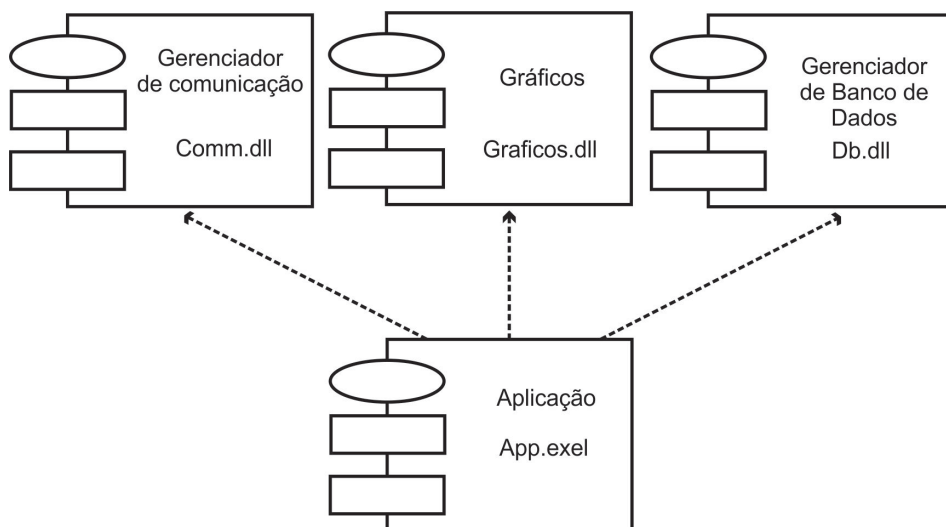
Fonte: adaptado de Larman, 2011.



Analise a ilustração acima e perceba que o diagrama de objetos é uma variação do diagrama de classes e utiliza quase a mesma notação. A diferença é que o diagrama de objetos mostra os objetos que foram instanciados das classes. O diagrama de objetos é como se fosse o perfil do sistema em um certo momento de sua execução.

*Diagrama de Componentes* - Mostra o sistema por um lado funcional, expondo as relações entre seus componentes e a organização de seus módulos durante sua execução.

Observe na imagem abaixo que a visão das relações entre os componentes é melhor percebida a partir desse tipo de representação e modelagem.



**Figura 22**

Fonte: adaptado de Larman, 2011.

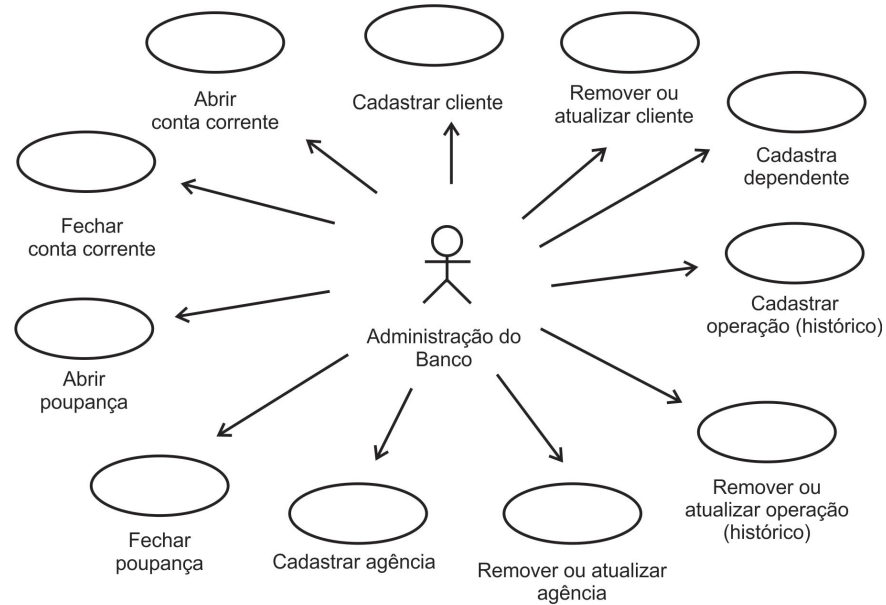
*Diagramas comportamentais* - São responsáveis pela modelagem dos aspectos dinâmicos de um sistema, ou seja, pela representação das partes do sistema que sofrem alterações, como, por exemplo, o fluxo de mensagens ao longo do tempo e a transição de estados que o mesmo pode assumir.

*Diagrama de casos de uso* - Assumem papel importante na modelagem comportamental de um sistema, um subsistema ou uma classe, por apresentar uma visão externa de como esses elementos podem ser utilizados no contexto.

Um diagrama de caso de uso mostra o relacionamento entre os atores e os casos de uso dentro de um sistema. É um padrão de comportamento que o sistema exibe



Contém um conjunto de casos de uso e modela interações entre os atores e o sistema e do próprio sistema. Observe o exemplo da figura abaixo e perceba a visão global, ou seja, a visão de alto nível do sistema.



**Figura 23**

Fonte: adaptado de Larman, 2011.

Descreve um conjunto de cenários. Captura os requisitos de usuários. Delimita o escopo do sistema.

Generalização entre os casos de uso: o filho herda o comportamento do pai, podendo adicionar e redefinir passos em pontos arbitrários do comportamento do pai.

*Diagrama de sequência* - É um diagrama de interação responsável pela modelagem dos aspectos dinâmicos de um sistema e que dá ênfase à ordenação temporal das mensagens trocadas entre objetos deste sistema.

Para proporcionar melhor visualização do fluxo de controle ao longo do tempo, o diagrama de sequências é organizado colocando-se os objetos correspondentes na parte superior do diagrama, ao longo do eixo horizontal, e suas respectivas mensagens são colocadas ao longo do eixo vertical, em uma ordem cronológica, de cima para baixo.

As mensagens são representadas por setas horizontais, nas quais a ponta da seta identifica o objeto alvo. Desta forma, um diagrama de sequências ao ser criado deve ser associado a um diagrama de casos de uso ou a um diagrama



de classes ou objetos.

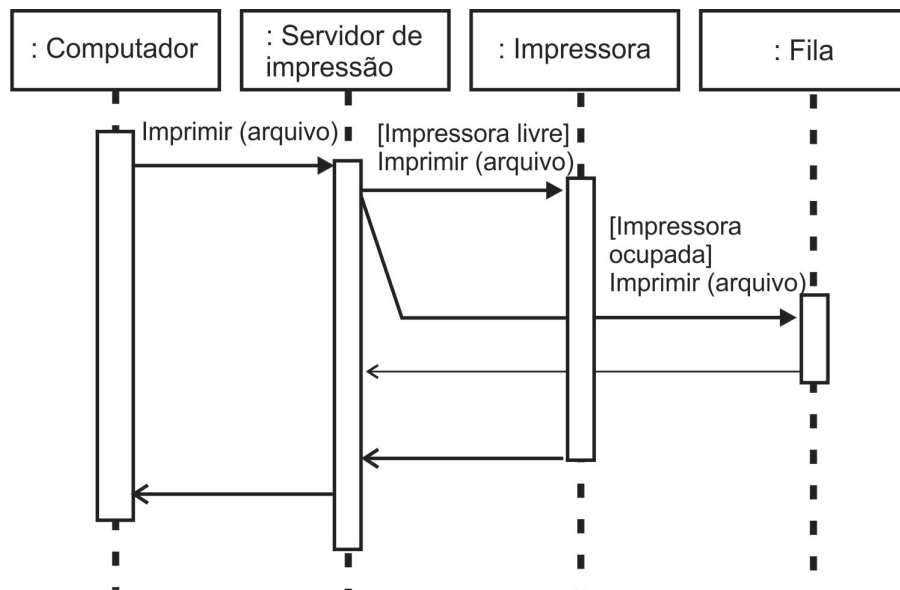


Figura 24

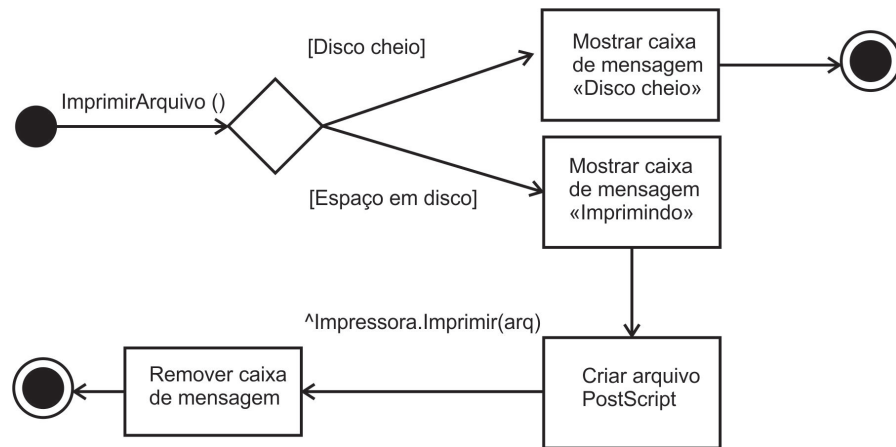
Fonte: adaptado de Larman, 2011.

Você pode entender a ilustração do diagrama acima, de modo que esse mostra a interação dinâmica entre os objetos ao longo do tempo e apresenta os objetos que participam da interação. O diagrama demonstra a sequência de mensagens trocadas entre os objetos.

*Diagrama de atividades* - É responsável pela modelagem do fluxo de controle entre as atividades de um sistema, ou seja, ele modela o fluxo sequencial das atividades geradas por uma operação do sistema ou por um agente externo. O diagrama de atividades é composto por estados de atividades, estados de ações e transições.

O diagrama de atividades é uma variação do diagrama de gráfico de estados, sendo que o diagrama de atividade não necessita especificar qual evento causou a mudança de um estado para outro no sistema.

Outra característica peculiar do diagrama de atividades é o fato de este ser modelado com um fluxograma, detalhando uma determinada operação de um sistema e também podendo considerar agentes externos que interagem com o mesmo.



**Figura 25**

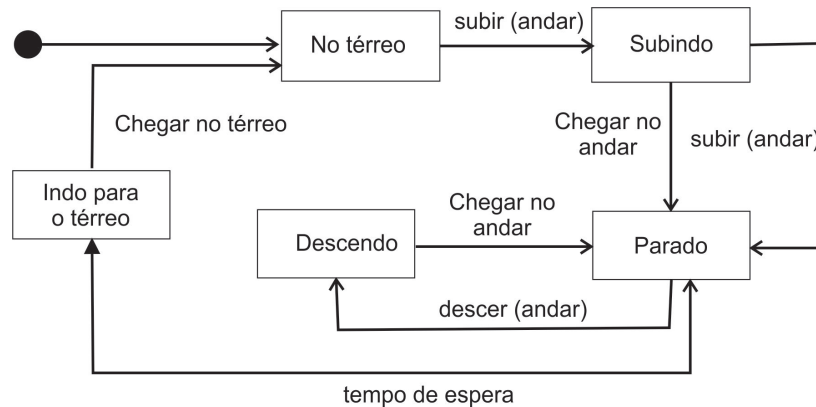
Fonte: adaptado de Larman, 2011.

Observe que o diagrama de atividades captura ações e seus resultados. Esse tipo de diagrama, conforme demonstra a ilustração acima, foca o trabalho executado na implementação de uma operação (método) e suas atividades numa instância de um objeto.

**A-Z**

Uma **máquina de estados** é o conjunto de estados pelo qual um objeto passa quando responde a um determinado evento.

*Diagrama de estados* - Envolve a modelagem comportamental de objetos reativos, modela o fluxo de controle de um estado para outro mediante **máquinas de estados**. O que vem a ser?



**Figura 26**

Fonte: adaptado de Larman, 2011.

Você pode perceber que, na figura acima, o diagrama apresenta os estados possíveis em que objetos de certa classe se podem encontrar e quais os eventos do sistemas que provocam tais mudanças.

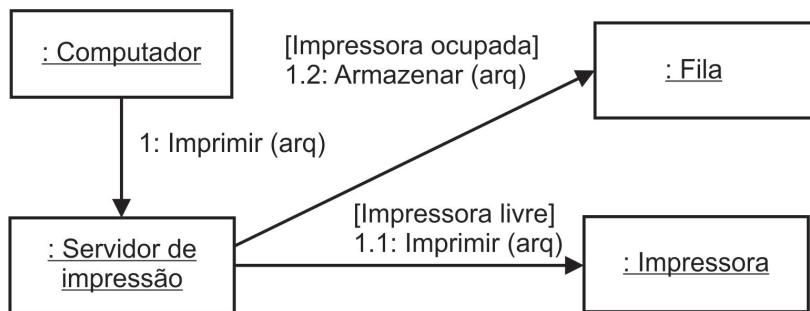
*Diagrama de Colaboração* - de maneira semelhante ao diagrama de sequência, apresenta a colaboração dinâmica entre os objetos. Você pode escolher







entre utilizar o diagrama de colaboração ou o diagrama de sequência.



**Figura 27**

Fonte: adaptado de Larman, 2011.



É importante que você busque outras referências e fontes para conhecer mais sobre os diagramas da UML e por isso indicamos que você acesse o link [https://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/uml-basics.html](https://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-basics.html). Nessa página você encontrará um maior detalhamento e uma visão rápida dos fundamentos da UML

## Resumo

A linguagem de modelagem unificada permite a profissionais de áreas diferentes trabalharem no mesmo projeto e desenvolvimento do sistema, proporcionando um modo de comunicação por meio de simbologia que todos conhecem e entendem.

Os diagramas proporcionam a visualização do sistema em diversas perspectivas e permite a especificação estrutural ou comportamental do sistema. De certa forma, é uma guia para a construção do sistema e facilita a produção da documentação das decisões tomadas durante a execução do desenvolvimento do *software*.

No mundo computacional, as necessidades são dinâmicas e mutantes, daí a facilidade que uma linguagem conhecida e interpretada de maneira uniforme é essencial para dar maior rapidez e suporte à construção e desenvolvimento para o projeto de sistema.

Com certeza você irá utilizar e praticar bastante, em sua atividade profissional, os conceitos e os diferentes diagramas apresentados nessa última aula da disciplina. É importante estudar este assunto, reler esta aula para não deixar nenhuma dúvida guardada para depois.

Sempre que sentir necessidade, volte e retome a leitura de qualquer das aulas desse caderno e procure outras referências sobre o assunto, pois um ótimo profissional sempre pesquisa e atualiza os conhecimentos adquiridos.





## Atividade de Aprendizagem

A partir do que estudou nessa quinta aula, disserte sobre a importância da utilização dos diagramas apresentadas pela UML. Em seguida, eleja e descreva um tipo de diagrama que você julga ser necessário para iniciar a prática da construção de um projeto de sistemas Web.

---

---

---

---

---

---

---

---

---

---

Chegamos ao final do estudo do conteúdo dessa disciplina e, com certeza, você deve ter-se dado conta da importância de temas como: a construção projeto de sistemas Web, a qualidade do projeto de sistemas Web, a documentação e os padrões (*patterns*), a atividade de levantamento de requisitos necessários para o projeto de sistemas Web, a especificação e a análise de requisitos do projeto de sistemas Web.

Estudou sobre modelagem de sistemas Web, aprendeu a respeito da validação de requisitos para o projeto de sistemas Web e, conjuntamente, acerca das características relacionadas à verificação dos requisitos para o projeto de sistemas Web.

Pôde conhecer também as metodologias utilizadas no projeto de sistemas Web, as ferramentas utilizadas para a modelagem de sistemas Web e as vantagens e os desafios da modelagem de sistemas.

Estudou acerca da notação da linguagem de modelagem unificada; aprendeu sobre os diagramas da UML; conheceu as partes que compõem a UML, os itens de interação, ou seja, construiu sólido conhecimento acerca da linguagem.

Então, perceba que você detém vasto conhecimento sobre projetos de sistemas Web e já pode por em prática o que aprendeu ao longo da dessa disciplina no Curso Técnico em Informática para Internet.





Parabéns pela conclusão do estudo dessa disciplina. Temos certeza de que você está no caminho para a conquista da realização profissional.





## Palavras Finais

Estimado(a) estudante:

Gostaríamos de parabenizá-lo por ter chegado até aqui, por ter concluído o estudo da Disciplina Projeto de Sistemas Web.

Temos certeza de que você já é um vencedor por ter chegado à última aula desse caderno e que os conhecimentos adquiridos até aqui farão a diferença em sua vida profissional, proporcionando-lhe, conseqüentemente, melhor desempenho profissional.

Você não pode esquecer que a disciplina Projeto de Sistemas WEB é muito importante para o curso Técnico em Informática para Internet. Essa disciplina contribui para seu maior aprimoramento com relação à abstração do mundo real e à projeção do mesmo para o mundo computacional.

Esperamos que nossos objetivos aqui, como autores deste Caderno, foram alcançado, pois procuramos explicar o conteúdo da disciplina da melhor maneira possível, com carinho e clareza.

Esperamos que você tenha gostado do material apresentado e que também tenha aprendido sobre os conceitos acerca da atividade de construção do projeto de sistema Web.

Parabéns!

# Guia de Soluções

## Aula 1

Espera-se que o estudante possa identificar a importância de elaborar um projeto de sistemas Web e que mencione aspectos relacionados à qualidade e à utilização de padrões.

Poderá escrever nesse sentido: Os Padrões de Projeto devem ser observados como ferramentas para o projetista e o desenvolvedor de *software*, vez que auxiliará na solução de problemas que ocorrem com frequência, permitindo que os esforços de tais profissionais sejam utilizados em outra demanda.

## Aula 2

Espera-se que o estudante possa imaginar um projeto de sistema em que esteja trabalhando, um sistema que permitirá a organização e a catalogação de livros de sua biblioteca pessoal e que liste os requisitos funcionais e os requisitos não funcionais, pelo menos três, conforme exemplo:

Requisitos Funcionais	Requisitos Não Funcionais
São os requisitos que definem as funcionalidades do sistema.	São os requisitos que declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidades
Exemplos	
O sistema deve permitir que cada professor realize o lançamento de notas das turmas na quais lecionou.	Confiabilidade: corresponde a medidas quantitativas da confiabilidade do sistema, tais como tempo médio entre falhas, recuperação de falhas ou quantidade de erros por milhares de linha de código-fonte.
O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas em um semestre.	Desempenho: requisitos que definem tempos de resposta esperados para as funcionalidades do sistema.
Os coordenadores de escola devem poder obter o número de aprovações, reprovações e trancamentos em todas as turmas em um determinado período.	Portabilidade: restrições sobre as plataformas de <i>hardware</i> e de <i>software</i> nas quais o sistema será implantado e sobre o grau de facilidade para transportar o sistema para outras plataformas.

## Aula 3

Espera-se que o estudante possa imaginar diferentes visões a partir da observação dos diferentes profissionais envolvidos no projeto, com base nas descrições das visões diferenciadas para modelagem mencionadas por Pressman (2011).

**Poderá escrever nesse sentido:** O autor aborda que existem visões diferenciadas para a modelagem de requisitos, sendo:



- Análise Estruturada - Considera os dados e os processos que transformam os dados em entidades separadas. Os objetos de dados são modelados de maneira que definam seus atributos e relacionamentos.
- Análise orientada a objetos - Concentra-se na definição de classes e na maneira pela qual colaboram entre si para atender às necessidades dos clientes.

#### Aula 4

Espera-se que o estudante disserte sobre a importância dos tipos de modelos para modelagem de sistema propostos por Pressman (2011) e estabeleça a diferença entre eles.

**Poderá escrever nesse sentido:** Os modelos de análise para projeto de sistemas Web são: o modelo de conteúdo, o modelo de interação, o modelo funcional e, por fim, o modelo de configuração. Os tipos de modelagem permitem visões de diferentes aspectos e peculiaridades do sistema.

#### Aula 5

Espera-se que o estudante escreva sobre a importância da utilização dos diagramas apresentados pela UML e eleja um tipo de diagrama que julga ser necessário para iniciar a prática da construção de um projeto de sistemas Web.

**Poderá escrever nesse sentido:** O diagrama de casos de uso é o mais comum e utilizado para diversas visões do sistema.

- Os diagramas de casos de uso assumem papel importante na modelagem comportamental de um sistema, um subsistema ou uma classe, por apresentarem uma visão externa de como esses elementos podem ser utilizados no contexto.
- Um diagrama de caso de uso mostra o relacionamento entre os atores e os casos de uso dentro de um sistema. É um padrão de comportamento que o sistema exhibe.



## Referências

BOOCH, Grady; RAUMBUGH, James; JACOBSON, Ivar. **UML - Guia do Usuário**. 2. ed. São Paulo: Elsevier, 2012.

\_\_\_\_\_, RUMBAUGH, J., JACOBSON, I. **UML - Guia do Usuário**. 2 ed. S. Paulo: Elsevier, 2006.

BRAUDE, E. **Projeto de Software: da Programação à Arquitetura**. Porto Alegre: Bookman, 2005.

BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAD, P., STAL, M. **Pattern-Oriented Software Architecture: A System of Patterns**. Vol. 1. S.l.: Wiley, 1996.

CARDOSO, A.C., MARQUES, A.S., **Sistemas Baseados em Regras**. 2007. Disponível em: <<http://mestradosiad.blogspot.com/2007/11/sistemas-baseados-em-regras.html>>

FOWLER, M. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 3. ed. Porto Alegre: Bookman, 2005.

GAMMA, E., HELM, R., JOHNSON, R., VLISSIDES, J.M., **Design Patterns - Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.

LARMAN, Craig. **Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos**. 3. ed. Porto Alegre: Bookman, 2007.

LAKATOS, Eva Maria. MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 7. ed. São Paulo: Atlas 2010.

Motamarri, S. Systems Modeling and Description. In: **Software Engineering Notes**. vol. 17, n. 2, abr. 1992, p. 57-63.

PFLIEGER, S.L. **Engenharia de Software: Teoria e Prática**. 2 ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, R.S.. **Engenharia de Software**. McGraw-Hill. 6 ed. 2006.

PMBOK. **Um guia do Conjunto de Conhecimentos em Gerenciamento de Projetos**. Uma Norma Nacional Americana ANSI/PMI 99-001-2004. 3. ed.

SIMÃO, R. P. S. **Características de Qualidade para Componentes de Software**. 2002. Dissertação (Mestrado em Informática Aplicada) - Programa de Pós-Graduação em Informática, Fundação Educacional Edson de Queiroz, Universidade de Fortaleza, Fortaleza.

THAYER, R. H., e Dorfman, M. **Software Requirements Engineering**. 2 ed. IEEE Computer Society Press, 1997.

WAZLAWICK, R.S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. São Paulo: Elsevier, 2004.





## Currículo dos Professores-autores



**Maria Ivanilse Calderon Ribeiro** é bacharel em Sistemas de Informação pela Universidade Luterana do Brasil - Ulbra e pós-graduada lato sensu em MBA - Desenvolvimento de *Software* para Web, pelo Centro Universitário Luterano de Ji-Paraná-RO/Ceulji/Ulbra e especialista lato sensu em Metodologia e Didática do Ensino Superior pela Faculdade de Ciências Humanas, Exatas e Letras de Rondônia - Faro. É mestra em Geografia pela Universidade Federal de Rondônia – Unir, pesquisadora do grupo Gepgênero da Unir e professora do Instituto Federal de Rondônia - Ifro. Atua como analista criminal da Secretária de Segurança e Defesa da Cidadania de Rondônia. Tem experiência na área da ciência da computação, com ênfase em geotecnologias, banco de Dados e desenvolvimento para Web.



**Juliana Braz da Costa** é bacharel em Sistemas de Informação pela Universidade Luterana do Brasil - Ulbra e pós-graduada lato sensu em MBA - Desenvolvimento de *Software* para Web, pelo Centro Universitário Luterano de Ji-Paraná-RO/Ceulji/Ulbra e especialista lato sensu em Metodologia e Didática do Ensino Superior pela Faculdade de Ciências Humanas, Exatas e Letras de Rondônia - Faro. É mestra em Ciência da Computação pela Universidade Federal de Pernambuco - UFPE. Atualmente, atua como professora do ensino básico, técnico e tecnológico do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - Ifro. Tem experiência na área de ciência da computação, com ênfase em engenharia de *software*, gestão de projetos, banco de dados e informática aplicada.



**Jhordano Malacarne Bravim** é tecnólogo em Redes de Computadores pelo Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, pós-graduado lato sensu em Tecnologia da Informação pela Faculdade São Lucas e mestrando em Administração pela Universidade Federal de Rondônia. Atualmente é analista de tecnologia da informação do Instituto Federal de Educação Ciência e Tecnologia de Rondônia, tendo sido Professor na Faculdade Porto. Tem experiência na área de ciência da computação, com ênfase em governança de tecnologia da informação e redes, atuando princi-







palmente nos seguintes temas: governança de TI, tecnologia da informação, planejamento estratégico e processos.



