

EVANDRO MANARA MILETTO
SILVIA DE CASTRO BERTAGNOLLI

DESENVOLVIMENTO DE SOFTWARE II

INTRODUÇÃO AO DESENVOLVIMENTO WEB
COM HTML, CSS, JAVASCRIPT E PHP

EIXO INFORMAÇÃO E COMUNICAÇÃO

» série tekne





D451 Desenvolvimento de software II [recurso eletrônico] : introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP / Organizadores, Evandro Manara Miletto, Sílvia de Castro Bertagnolli. – Dados eletrônicos. – Porto Alegre : Bookman, 2014.

Editado também como livro impresso em 2014.
ISBN 978-85-8260-196-9

1. Informática – Desenvolvimento de software. 2. HTML.
3. CSS. 4. JavaScript. 5. PHP. I. Miletto, Evandro Manara.
II. Bertagnolli, Sílvia de Castro.

CDU 004.41



>> capítulo 1

Definição do sistema

O desenvolvimento de um sistema computacional não é uma atividade trivial, pois envolve analisar e compreender determinado problema. Quando o software é desenvolvido para a plataforma Web, vários aspectos são incorporados de modo que ele possa ser acessado de forma remota e segura por meio de um navegador. Neste capítulo, você conhecerá alguns dos aspectos relacionados com a Internet e suas tecnologias, bem como um estudo de caso cuja aplicabilidade será demonstrada nos capítulos seguintes.

Objetivos de aprendizagem

- >> Identificar as categorias de sistemas.
- >> Reconhecer o funcionamento da Internet.
- >> Analisar o estudo de caso apresentado.

>> Introdução



>> ATENÇÃO

Software é diferente de programa de computador! Um **software** é um produto desenvolvido por meio de padrões de qualidade e que possui uma documentação agregada, podendo ser composto por um ou vários programas. Já um **programa** tem um conceito mais amplo, podendo ser definido como um conjunto de linhas de código (ou algoritmo executável) em determinada linguagem de programação, cujo objetivo é resolver um problema.



>> DEFINIÇÃO

A **disponibilidade** é o tempo durante o qual um sistema mantém seus serviços disponíveis para acesso pelos usuários. No caso de sistemas Web, a disponibilidade pretendida é 24×7 , ou seja, 24 horas do dia, 7 dias por semana.

Antes de apresentar os aspectos relacionados ao desenvolvimento Web, tema deste livro, é preciso analisar algumas classificações e aplicabilidades dos software, bem como o funcionamento dos **sistemas para Web**.

Nos dias atuais, o uso de software nas mais diversas áreas, e para diferentes fins, é uma atividade trivial. Um sistema que controla o metrô de uma grande cidade, um jogo de computador, um sistema da folha de pagamento de uma empresa, um sistema de vendas online ou um sistema embutido em um carro são exemplos de aplicabilidades de programas de computador.

Porém, o desenvolvimento dessas aplicações requer técnicas, métodos e mecanismos específicos para que elas funcionem conforme o esperado. Um jogo de computador, por exemplo, tem seu foco principal na interação do usuário, em um bom roteiro e na qualidade visual. Já os sistemas embutidos têm como foco a gerência e o controle de dispositivos físicos (hardware).

Um sistema de folha de pagamento pode ser considerado como uma aplicação que é executada em um **computador local** (não necessitando de conexão em rede). Para isso, todos os dados e recursos necessários para o seu funcionamento já estão disponíveis nesse computador. Assim, ele se concentra no processamento de dados, gerando informações condizentes com os dados de entrada.

Ao considerar um sistema de vendas online, é necessário observar que, em primeiro lugar, ele deve estar sempre disponível, ser executado a partir de um **computador remoto** (denominado servidor) e ser acessado pelos usuários através de seus computadores (clientes) por meio de um navegador (browser).

Analisando o sistema de folha de pagamento e o de vendas online, é possível observar que há uma grande diferença entre eles. O primeiro é **centralizado**, o que significa dizer que tudo é executado e processado em um único computador. Já o segundo sistema é considerado **distribuído**, ou seja, parte do processamento ocorre em um computador e parte em outro, denominado servidor. Esse tipo de sistema é organizado como um **sistema cliente-servidor** (Figura 1.1).

A partir da Figura 1.1, é possível deduzir que o usuário, por meio da máquina cliente em que está executando o navegador, solicita informações a uma aplicação que está rodando em um servidor remoto, através da Internet. Esse servidor analisa a requisição enviada e responde com dados em forma de páginas Web.

Tratando-se de uma **aplicação Web**, é necessário levar em conta vários aspectos para que se possa, de fato, utilizá-la. Esses aspectos incluem o conhecimento dos requisitos exigidos pela aplicação, como:

- Quais tecnologias podem ser utilizadas.
- Como é realizado o processamento do sistema.
- Quais são o perfil do usuário e suas tarefas na interface gráfica da aplicação.
- Como armazenar os dados.

É importante saber, neste momento, que um sistema Web, para ser considerado robusto e consistente, normalmente separa esses aspectos em quatro partes diferentes. Essa separação está relacionada com as **boas práticas de desenvolvimento de software**, em que um dos objetivos é permitir a implementação de alterações e atualizações em cada uma das partes sem prejudicar as demais. A Figura 1.2 esquematiza um modelo com a separação dessas camadas. Cabe lembrar que, à medida que avançarmos nos capítulos deste livro, as relações entre esses aspectos e tecnologias e suas respectivas camadas ficarão mais evidentes.

A próxima seção apresenta o histórico do surgimento da Internet e algumas das principais tecnologias correlacionadas com a camada de apresentação (a qual será abordada com mais detalhes no Capítulo 3, “Projeto de interface com o usuário”).

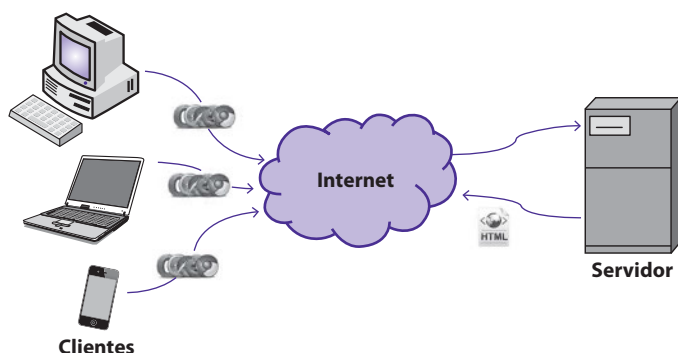


Figura 1.1 Estrutura cliente-servidor.

Fonte: dos autores.

»» PARA SABER MAIS

Para aprofundar seus conhecimentos sobre o funcionamento dos sistemas distribuídos e como eles são classificados consulte os livros *Sistemas distribuídos: conceitos e projeto*, 5. ed., e *Sistemas distribuídos*, 2. ed. (Veja as referências ao final do capítulo.)

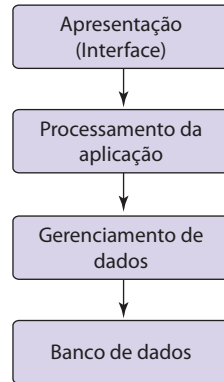


Figura 1.2 Organização da aplicação em camadas.

Fonte: dos autores.

>> A Internet e suas tecnologias

A **Internet** é conhecida como uma grande rede que interliga computadores de todo o mundo. Ela foi criada por Tim Berners-Lee quando ele pesquisava mecanismos para interligar e compartilhar documentos entre pesquisadores de diferentes instituições. O Quadro 1.1 apresenta um resumo histórico dos principais avanços relacionados com a Internet.

A partir das ideias de Tim Berners-Lee surgiu o conceito de **hipertexto**, que introduziu uma nova forma de organização da informação. Ele possibilita percorrer partes do documento (e outros documentos) por meio de ligações ou palavras que aparecem em destaque no texto, chamadas de **hiperlinks** ou **links**.

Como já mencionamos, a Internet usa o modelo cliente-servidor (representado na Figura 1.1), o que permite interligar qualquer tipo de computador em qualquer lugar do mundo. Nesse modelo, o cliente solicita informações por meio de um “endereço virtual”, e o servidor responde à solicitação com a resposta ao pedido enviado.

Esse “endereço virtual” é informado em um software específico para navegar na Internet – o **navegador** (ou browser). Ele é um dos elementos principais para se utilizar uma aplicação Web. Os principais navegadores utilizados para acessar sites na Internet, distribuídos de forma gratuita em sua maioria, são: Internet Explorer, Firefox, Safari, Opera, Google Chrome.

Quadro 1.1 » Resumo histórico do surgimento da Internet

1991	Surgimento da lista de discussão eletrônica WWW-talk, cujo foco é compartilhar experiências relacionadas com a linguagem HTML.
1992	Desenvolvimento do navegador Mosaic.
1993	Criação do navegador de texto Lynx.
1994	Estabelecimento da especificação HTML 2 (unificação das diferentes formas de HTML). Criação da empresa Netscape Communications Corporation. Surgimento do W3C (<i>World Wide Web Consortium</i>).
1995	Definição de novas marcações para o HTML. Criação do conceito de frames. Lançamento do navegador Internet Explorer 2.0.
1996	Lançamento do HTML versão 4.
1997	O HTML passa a incorporar tabelas, <i>applets</i> , subscritos, sobrescritos e texto combinado com imagens.
1998	Reformulação do HTML em XML.
1999	Lançamento do XHTML 1.0.
2004	Criação do grupo WHATWG (<i>Web Hypertext Application Technology Working Group</i>).
2007	Início do desenvolvimento do HTML 5, que incorpora TAGs para animações.



» DICA

O **W3C** é um consórcio de empresas que desenvolve padrões para criação e interpretação dos conteúdos para a Web. Sistemas desenvolvidos segundo esses padrões podem ser mais facilmente acessados e visualizados por qualquer pessoa, com o uso de qualquer tecnologia, independentemente do hardware ou software utilizado.

O “endereço virtual”, também chamado de URL (*Uniform Resource Locator*), deve ser informado no navegador (na barra de endereços) para acessar o conteúdo disponibilizado. As partes que compõem esse endereço estão destacadas na Figura 1.3.

Já o protocolo identifica o tipo de transferência de arquivo que será utilizado. O HTTP (*HyperText Transfer Protocol*) é o protocolo utilizado para a transferência de documentos hipermídia (hipertexto) na Internet. Após o protocolo, coloca-se a identificação do servidor da Web para onde a requisição será enviada.

Em alguns casos, é necessário identificar o caminho do arquivo que se deseja acessar, especificando os diretórios nos quais é possível localizar o arquivo que está sendo solicitado, e, por fim, o nome do arquivo que será exibido no navegador. Esses arquivos podem ser escritos em várias linguagens (HTML e PHP, por exemplo), sendo que o browser é o responsável por interpretá-las e exibi-las para o usuário.

Outra questão relacionada ao desenvolvimento Web diz respeito às **ferramentas de desenvolvimento** utilizadas. Como as páginas Web são interpretadas pelo navegador, é possível, por exemplo, utilizar o bloco de notas para criar uma página.

www

» NO SITE

Conheça mais sobre o W3C, seus objetivos e suas padronizações no ambiente virtual de aprendizagem Tekne: www.bookman.com.br/tekne.



» DEFINIÇÃO

Software proprietário é aquele que não pode ser copiado, distribuído ou modificado sem a permissão da empresa que o criou. Já o **software livre**, por outro lado, pode ter inclusive seu código-fonte analisado, pois o código geralmente costuma ser distribuído junto com o software.



» DEFINIÇÃO

Open source é um modelo de desenvolvimento colaborativo de software. Com o código-fonte publicado na Internet, desenvolvedores distribuídos pelo mundo contribuem com o processo de desenvolvimento do produto.

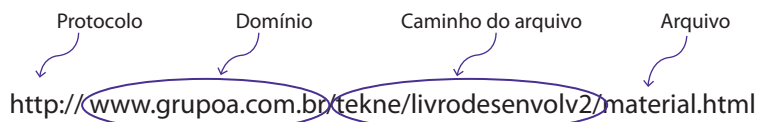


Figura 1.3 Compreendendo as partes de um endereço virtual.

Fonte: dos autores.

Porém, por ser um editor muito simples, o bloco de notas não oferece dicas nem aponta erros nas linhas de código de sua página, fazendo com que ele não seja o editor mais adequado para a criação de páginas Web.

Existem diversos editores de páginas Web disponíveis para download na Internet. Os mais recomendados são os classificados como IDE (*Integrated Development Environment*), que corresponde a um ambiente composto por um conjunto de ferramentas integradas que possibilitam criar o código, sinalizar erros, autocompletar partes do código ou visualizar o resultado do código executado.

A seguir, no Quadro 1.2, são apresentadas as principais ferramentas para desenvolvimento de páginas Web. Para mais informações ou download das ferramentas do Quadro 1.2, acesse o ambiente virtual de aprendizagem Tekne.

Quadro 1.2 » Principais ferramentas para desenvolvimento de páginas Web

Aptana Studio	Ambiente completo e gratuito para edição de páginas HTML, CSS e JavaScript. Possui extensões para desenvolvimento com Ruby on rails, Python e PHP.
Dreamweaver	Software proprietário desenvolvido pela Adobe.
Eclipse	Baseado na arquitetura de <i>plugins</i> , em que diferentes linguagens podem ser incorporadas ao ambiente com a adição de <i>plugins</i> .
HTML-Kit	Editor HTML gratuito, integrado a um visualizador de página Web. Realça as TAGs e fornece atalhos para o programador. Possibilita validação, publicação e gerenciamento de projetos Web.
NetBeans	IDE open source que possibilita a criação de programas em várias linguagens, além do desenvolvimento de páginas HTML 5 e PHP.
Notepad++	Editor para HTML e diversas outras linguagens, como PHP, Java e C. Não possui compiladores ou interpretadores integrados como NetBeans e Eclipse, mas é muito útil para a edição de programas.



>> capítulo 2

Projetando o site

Para projetar um site, é necessário reconhecer que as aplicações Web são um tipo de software que utiliza a Internet como um ambiente de execução. Essa aplicação amplia o conceito de Web site ao adicionar funcionalidades ao sistema. E, para compreender esse tipo de sistema, devemos entender os requisitos e mapear suas principais funcionalidades. Para projetar esse site, são necessários alguns passos, que devem ser mapeados em diagramas e modelos, de modo a facilitar a comunicação entre as pessoas envolvidas.

Objetivos de aprendizagem

- >> Definir o que é uma aplicação Web.
- >> Identificar as principais fases de desenvolvimento de uma aplicação Web.
- >> Modelar um sistema Web com diagramas e modelos.
- >> Reconhecer como um problema pode ter soluções mapeadas, com diagramas e modelos.

>> Introdução



>> DEFINIÇÃO

Uma aplicação projetada para uso na Internet e Intranet é um sistema de software baseado em tecnologias e padrões do W3C que provê recursos específicos, como conteúdo e serviços por meio de um cliente Web.

O projeto de uma aplicação para uso na Intranet e/ou Internet envolve o conhecimento de vários conceitos, os quais serão discutidos e apresentados neste capítulo.

Uma aplicação Web é diferente de um site estático. No site estático, o conteúdo é um arquivo ou documento pré-formatado, em que, por exemplo, todo o conteúdo está nas marcações em HTML conhecidas como TAGs, e nenhuma informação é carregada a partir de outros documentos ou bases de dados. Já uma **aplicação Web** é caracterizada por construir dinamicamente o seu conteúdo, com dados provenientes de um banco de dados, a partir da interação do usuário com as páginas, via navegadores.

A partir do esquema ilustrado pela Figura 2.1, podemos observar que um servidor recebe uma solicitação de um cliente por meio de um navegador. O servidor procura pelo documento em um sistema de arquivos e o envia de volta ao navegador para ser exibido ao cliente. Os recursos do sistema (texto, imagem, vídeo e áudio) são interligados entre si por links, que é a forma usual de navegação em aplicações Web. Para que uma aplicação seja executada, ela exige uma arquitetura Web, a qual está ilustrada na Figura 2.1.

As aplicações Web são diferentes das aplicações convencionais, pois apresentam um rápido crescimento no número de requisitos, conteúdo e funcionalidades, além de se caracterizarem pelas constantes alterações sofridas durante o ciclo de vida. Muitas vezes, essas aplicações são mal projetadas, devido a vários fatores, dentre os quais podemos citar:

- Necessidade de atualização constante das informações.
- Evolução extremamente rápida da tecnologia utilizada por esse tipo de aplicação.

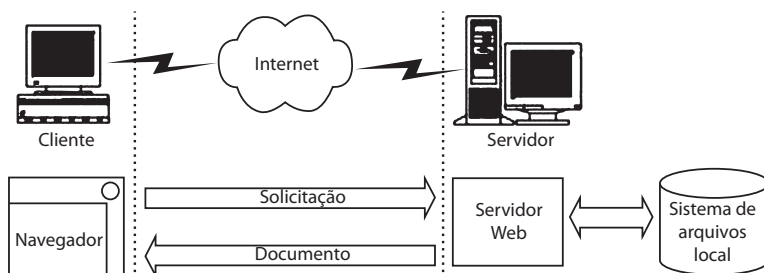


Figura 2.1 Arquitetura básica de site.

Fonte: Winckler e Pimenta (2002).

- Inexistência de modelos do processo.
- Planejamento incorreto e inadequação na definição de prazos e custos.
- Equipes mal organizadas e sem a competência adequada.
- Falta de documentação e dificuldades de implementação e manutenção.

Ao analisar esses fatores, é possível perceber que eles estão vinculados a algumas categorias: funcionamento, usabilidade, desempenho e portabilidade.

Os problemas de **funcionamento** geram no usuário as seguintes dúvidas: O sistema tem a informação ou o serviço de que eu preciso? Eu mandei os dados, mas será que o sistema os processou de maneira correta?

As dificuldades relativas à **usabilidade** estão vinculadas à localização da informação, e quando elas ocorrem o usuário se pergunta: Onde posso encontrar a informação ou o serviço? Como posso solicitar esse serviço? Quais informações eu devo fornecer? Qual foi o resultado? Era o que eu queria?

Já os problemas de **desempenho** têm relação com o tempo de resposta do sistema e podem ocasionar os seguintes questionamentos: Por que a página demora a carregar? A página será exibida, ou ocorreu algum erro?

As questões de **portabilidade** estão vinculadas diretamente com a visualização da informação e com as ações de formulários que não funcionam, gerando dúvidas como: O que aconteceu com a página? Ela não é exibida corretamente neste browser? A linguagem script não funciona neste cliente ou servidor?

Uma descrição explícita do site pode ajudar de várias formas no seu desenvolvimento. Eis os principais passos:

- Formalização de requisitos do usuário.
- Guia do projeto.
- Construção da aplicação.
- Documentação do conjunto de informações úteis no decorrer das atividades de avaliação e da aplicação.

Durante a fase de especificação do site, são produzidos os modelos que descrevem os requisitos do(s) usuário(s), bem como as tarefas e a estrutura que serão utilizadas para implementar a aplicação Web. Observe, na Figura 2.2, que, no ciclo de desenvolvimento de uma aplicação Web, há um atalho que permite a implementação logo após a análise de requisitos, sem passar pela etapa de especificação, o que é frequentemente observado na prática.



>> DICA

Os desenvolvedores de aplicações Web devem ter em mente que não basta saber HTML e um pouco de programação. São necessárias várias competências técnicas, bem como adotar um ciclo de desenvolvimento de software para Web, que deve ser baseado em um processo sistemático e cíclico.

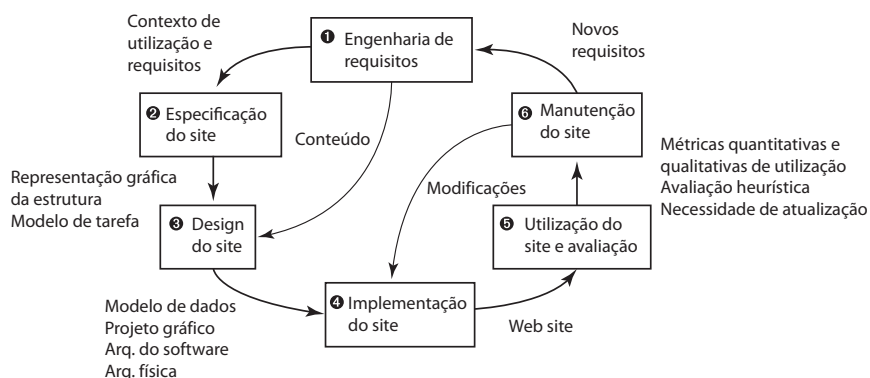


Figura 2.2 Ciclo de desenvolvimento de uma aplicação Web.

Fonte: Winckler e Pimenta (2002).



>> Agora é a sua vez!

1. Em sua opinião, qual é o principal fator para que uma aplicação Web seja mal definida e mal projetada?
2. Analise os problemas relacionados ao desenvolvimento de aplicações Web e dê sugestões de como resolvê-los.

>> O processo de desenvolvimento de software

A fim de evitar os erros e problemas relatados na seção anterior, a partir de agora vamos mostrar os principais passos de desenvolvimento e modelagem de uma aplicação Web.



>> PARA REFLETIR

Imagine que você foi convidado para desenvolver um sistema. Qual é o seu ponto de partida? Que passos você deve seguir? Como você vai registrar as descobertas realizadas em cada passo? Quais serão as pessoas envolvidas? Esses são alguns dos questionamentos que surgem quando se desenvolve um software.

Ao iniciar o desenvolvimento de um software, podemos optar por usar, ou não, um processo de desenvolvimento. Sua adoção permite definir passos e artefatos de documentação, bem como atribuir responsabilidades.

Um processo abrange as atividades necessárias para compreender, modelar, desenvolver, testar, implantar e manter um produto de software. Portanto, ele deve:

- Definir as atividades que serão executadas ao longo do projeto.
- Determinar quando, como e por quem serão executadas as atividades.
- Delimitar os artefatos que serão utilizados e qual é o melhor momento para aplicá-los.
- Determinar as ferramentas que auxiliarão na construção do produto de software.
- Fornecer um “guia” para as atividades realizadas a todos os envolvidos com o desenvolvimento do produto.

Existem diversos processos, conforme já comentado no Capítulo 1, que podem ser encontrados na literatura (PRESSMAN, 2011; SOMMERVILLE, 2007):

- Modelo cascata ou sequencial linear ou ciclo de vida clássico.
- Baseados em prototipagem.
- Modelos RAD (*Rapid Application Development*).
- Modelo espiral.
- Modelo incremental
- Modelo iterativo e incremental.
- Modelos ágeis.
- Processo unificado.

Cada modelo de processo é recomendado para determinadas situações. Assim, sistemas orientados a eventos podem utilizar o modelo RAD, enquanto sistemas móveis podem ser desenvolvidos utilizando modelos ágeis.



>> DEFINIÇÃO

Um processo de desenvolvimento de software define um conjunto de passos sistematizados, que devem ser executados em uma dada sequência, de modo a produzir software com qualidade e com características de produto.



>> DICA

Os processos de desenvolvimento de software são importantes porque transformam modelos conceituais em modelos de solução detalhados.



>> ATENÇÃO

Requisitos de segurança estão relacionados com autenticação, desempenho, disponibilidade, confidencialidade, entre outros. **Requisitos de qualidade** têm relação direta com usabilidade, funcionalidade, confiabilidade e eficiência.



>> NO SITE

Obtenha mais informações sobre técnicas de planejamento e gerenciamento de projetos de software e sobre o *Project Management Body of Knowledge (PMBOK)* no ambiente virtual de aprendizagem Tekne: www.bookman.com.br/tekne.

Da mesma forma, o desenvolvimento de uma aplicação Web difere do desenvolvimento de aplicações tradicionais porque se fundamenta em vários tipos de elementos: intensivo uso de redes de computadores, requisitos de segurança, tempo de resposta reduzido, disponibilidade, qualidade visual e prazo de desenvolvimento curto, processamento correto dos links, cuidados especiais com a navegação.

O **modelo iterativo e incremental** é um dos processos recomendados para o desenvolvimento de aplicações Web, pois permite incorporar requisitos de segurança e de qualidade à medida que o software é desenvolvido. Também possibilita que as funcionalidades sejam divididas em partes menores ou iterações.

O ponto fundamental do modelo iterativo e incremental é que a equipe envolvida com o desenvolvimento pode definir um núcleo base de funcionalidades, refinando e ampliando esse núcleo de modo a obter o produto de software desejado.

Além disso, agrega elementos do modelo cascata com a proposta iterativa da prototipagem. Cada sequência produz um incremento nas funcionalidades do software. O número de iterações é estabelecido conforme o escopo e as funcionalidades previstas para o sistema. Para cada sistema é definido o número de iterações necessárias para concluir todas as funcionalidades previstas para o produto de software. Em cada iteração são identificadas as seguintes fases:

Planejamento: estabelece aspectos relacionados com o projeto, como delimitação do escopo, recursos (humanos, hardware e software) e estimativas de tempo para compor um cronograma. Existem diversas técnicas que podem ser utilizadas para realizar o planejamento de um sistema, e elas não serão abordadas neste livro, pois fogem de seu escopo. Optamos por criar uma seção que delimita o sistema, determinando alguns itens que são essenciais para o desenvolvimento do sistema, mas que não exigirão do leitor um conhecimento muito avançado de planejamento e gerenciamento de projetos.

Análise: a criação de um modelo de análise descrevendo os requisitos pode ser realizada de forma textual, visual (com modelos e diagramas) ou, ainda, textual/visual, que alia modelos, diagramas e representações visuais dos principais elementos de conteúdo das páginas.

Projeto: define os modelos e o projeto visual e gráfico da aplicação (tópico também abordado no Capítulo 3), que também são essenciais para criar uma aplicação Web.

Implementação: consiste na elaboração das páginas, dos scripts de páginas e de banco de dados (tópicos apresentados nos Capítulos 4 a 10).

Testes: etapa que contempla os testes (unidade, sistema e integração) das funcionalidades previstas para as páginas e das consultas ao banco de dados.

Algumas dessas etapas serão descritas nas próximas seções. Além disso, seus aspectos teórico-práticos serão apresentados e correlacionados com o estudo de caso apresentado no Capítulo 1.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.