



Interativa

Desenvolvimento de Software para Internet

Autor: Prof. Salatiel Luz Marinho

Colaboradores: Prof. Angel Antonio Gonzalez Martinez

Profa. Christiane Mazur Doi

Profa. Larissa Rodrigues Damiani

Professor conteudista: Salatiel Luz Marinho

Formado em Engenharia da Computação pela Universidade Paulista (UNIP) e pós-graduado em Gestão de Projetos pelo Instituto Brasileiro de Tecnologia Avançada (IBTA). Trabalha há mais de 15 anos no desenvolvimento de aplicações desktop e web, tendo prestado serviços a instituições financeiras e das áreas de varejo, de seguros e acadêmica. Leciona há mais de 10 anos disciplinas dos cursos de Análise e Desenvolvimento de Sistemas, Engenharia da Computação, Gestão de TI e Ciência da Computação.

Dados Internacionais de Catalogação na Publicação (CIP)

M338d Marinho, Salatiel Luz.

Desenvolvimento de Software para Internet / Salatiel Luz Marinho.
– São Paulo: Editora Sol, 2022.

240 p., il.

Nota: este volume está publicado nos Cadernos de Estudos e Pesquisas da UNIP, Série Didática, ISSN 1517-9230.

1. HTML. 2. CSS3. 3. Cores. I. Título.

CDU 681.3.062

U515.56 – 22

Prof. Dr. João Carlos Di Genio
Reitor

Profa. Sandra Miessa
Reitora em Exercício

Profa. Dra. Marília Ancona Lopez
Vice-Reitora de Graduação

Profa. Dra. Marina Ancona Lopez Soligo
Vice-Reitora de Pós-Graduação e Pesquisa

Profa. Dra. Claudia Meucci Andreatini
Vice-Reitora de Administração

Prof. Dr. Paschoal Laercio Armonia
Vice-Reitor de Extensão

Prof. Fábio Romeu de Carvalho
Vice-Reitor de Planejamento e Finanças

Profa. Melânia Dalla Torre
Vice-Reitora de Unidades do Interior

Unip Interativa

Profa. Elisabete Brihy
Prof. Marcelo Vannini
Prof. Dr. Luiz Felipe Scabar
Prof. Ivan Daliberto Frugoli

Material Didático

Comissão editorial:

Profa. Dra. Christiane Mazur Doi
Profa. Dra. Angélica L. Carlini
Profa. Dra. Ronilda Ribeiro

Apoio:

Profa. Cláudia Regina Baptista
Profa. Deise Alcantara Carreiro

Projeto gráfico:

Prof. Alexandre Ponzetto

Revisão:

Bruna Baldez
Jaci Albuquerque

Sumário

Desenvolvimento de Software para Internet

APRESENTAÇÃO	9
INTRODUÇÃO	9

Unidade I

1 MONTANDO O AMBIENTE DE ESTUDO	15
1.1 Editor Sublime Text.....	15
1.2 Visual Studio Code.....	21
1.2.1 Por que o Visual Studio Code?	22
1.2.2 Baixando e configurando o Visual Studio Code.....	23
2 ESTAÇÃO HTML	25
2.1 Introdução à linguagem de marcação.....	25
2.2 Hipertexto e marcação.....	26
2.3 História do HTML.....	27
2.4 HTML5	29
2.5 Arquitetura HTML.....	29
2.6 Motores de renderização.....	30
2.7 Codificação de caracteres e tabela ASCII	33
2.8 Tags HTML.....	37
2.8.1 Atributos de tag	37
2.8.2 Tags essenciais de um arquivo HTML.....	38
2.8.3 Cabeçalhos.....	41
2.8.4 Negrito: 	42
2.8.5 Itálico: <i> </i>	42
2.8.6 Sublinhado: <u> </u>	43
2.8.7 Pequeno: <small> </small>.....	43
2.8.8 Cor da fonte: 	44
2.8.9 Tamanho da fonte: 	48
2.8.10 Alteração do tamanho da fonte: 	48
2.8.11 Centro: <center> </center>	49
2.8.12 Ênfase: 	49
2.8.13 Ênfase forte: 	50
2.8.14 Parágrafos: <p> </p>	50
2.8.15 Quebra de linha: 	52
2.8.16 Divisão: <div> </div>	52
2.8.17 Tabela: <table> </table>	54
2.8.18 Table cell: <td> </td>	58

2.9 Ligações.....	59
2.9.1 Link básico: texto do link	59
2.9.2 Link para um fragmento: link	60
2.9.3 Janela de destino: link	61
2.9.4 Imagem como um link: 	61
2.9.5 Link de e-mail: 	62
2.10 Imagens.....	63
2.10.1 Imagem: 	63
2.10.2 Texto alternativo: 	64
2.10.3 Tamanho da imagem: 	65
2.10.4 Bordas 	66
2.10.5 Alinhamento de imagem: - 	67
2.10.6 Espaçamento: 	68
2.11 Estudo da estrutura de arquivos HTML	70
2.11.1 Doctype: definindo o documento	71
2.11.2 Head	71
2.11.3 Body	71

Unidade II

3 FORMULÁRIOS COM HTML.....	80
3.1 Elemento principal de um formulário: <form> ... </form>.....	80
3.1.1 Utilizando o método get.....	81
3.1.2 Utilizando o método post	81
3.2 Campo de entrada: <input>.....	82
3.3 Botão: <button>	83
3.4 Lista suspensa de seleção: <select> ... </select>	84
3.4.1 Item de seleção da lista suspensa: <option> </option>	84
3.5 Área de texto: <textarea> </textarea>	84
3.6 Elementos <fieldset> e <legend>.....	86
3.7 Validações de campos de um formulário em HTML.....	89
3.7.1 Atributo obrigatório	89
3.7.2 Atributos readonly e disabled	91
4 LISTAS COM HTML E INTRODUÇÃO AO XHTML.....	93
4.1 Listas HTML	93
4.1.1 Lista não ordenada.....	93
4.1.2 Lista ordenada.....	95
4.1.3 Lista de descrição.....	97
4.1.4 Atributo type do item de uma lista: <li type="">	98
4.2 Introdução ao XHTML.....	100
4.2.1 Recursos de documentos HTML e XHTML.....	102
4.2.2 Estrutura de um documento XHTML	104
4.2.3 Como converter HTML para XHTML.....	105

Unidade III

5 ESTAÇÃO CSS3.....	112
5.1 Introdução ao CSS3.....	112
5.2 Sintaxe básica do CSS.....	114
5.3 Formas de adição de CSS	114
5.4 Seletores CSS.....	121
5.4.1 Seletores simples.....	121
5.4.2 Seletores combinados.....	131
5.4.3 Seletores de atributo.....	135
5.4.4 Seletores de pseudoclasses.....	139
6 ESTUDO DE CORES E FONTES	150
6.1 Estudo de cores.....	150
6.1.1 Representação em sistema RGB.....	150
6.1.2 Representação em sistema hexadecimal	151
6.1.3 Representação por nomes das cores	152
6.2 Estudo de fontes.....	154
6.2.1 Gerando uma fonte adequada.....	156

Unidade IV

7 ESTAÇÃO JAVASCRIPT	163
7.1 JavaScript	163
7.1.1 Introdução ao JavaScript.....	163
7.1.2 Relembrando conceitos de linguagens de programação	163
7.1.3 Objetos JavaScript.....	165
7.1.4 Construtores JavaScript criam e retornam instâncias de objetos.....	170
7.1.5 Funções a partir do construtor Function()	172
7.1.6 Aplicações do método getElementById().....	174
7.1.7 Array.....	177
7.2 Uso de roteamento de soluções MVC.....	178
7.3 Razor.....	182
7.4 Rest	183
7.5 Exemplo de web API.....	186
7.6 Desenvolvimento de aplicações MVC	199
8 ACESSANDO DADOS COM MS ADO.NET	200
8.1 Instalar/configurar o banco de dados SQLite	221

APRESENTAÇÃO

Esta disciplina apresenta, de forma estruturada, os principais conceitos teóricos e práticos relativos ao desenvolvimento web, isto é, ao mundo da internet.

A disciplina busca expor pontos importantes quanto ao ciclo histórico da internet. A partir do desenho de uma linha do tempo, discorre sobre a concepção da internet, sua história, seus marcos e, principalmente, o estágio de desenvolvimento atual.

Os conceitos técnicos aqui estudados serão de grande importância para a sinergia do trabalho acadêmico e do projeto integrado multidisciplinar (PIM). Em diversos momentos, o aluno é inserido em um processo imersivo, simulando cenários variados de desenvolvimento de um projeto de software.

O principal objetivo desta disciplina é, portanto, colocar à disposição do aluno conceitos técnicos e teóricos que possam alavancar estudos futuros.

A intenção é propor um aprendizado rico e imersivo no que diz respeito ao desenvolvimento de software para internet, para que, ao final, o aluno esteja apto a desenvolver sites e módulos capazes de se integrar com aplicações desktop e mobile, bem como aplicar todo o conhecimento adquirido em seu dia a dia de trabalho.

INTRODUÇÃO

Caro(a) aluno(a), juntos iremos aprender sobre o desenvolvimento de softwares para internet, um tema que, em conjunto com os demais tratados na disciplina do curso, irá enriquecer ainda mais o seu conhecimento.

Neste livro-texto, passaremos por algumas estações de trabalho que apresentam itens importantes, como descrevemos a seguir.

- **Estação HTML:** linguagem de marcação HTML, do inglês, hypertext markup language (linguagem de marcação de hipertexto), que a cada dia vem ganhando mais força no ambiente web, nos tablets e nos smartphones.
- **Estação CSS:** técnica de formatação que utiliza o CSS, do inglês, cascading style sheets (folha de estilo em cascata). O CSS é usado para formatar de maneira eficaz a informação que vem do HTML nas formas de imagem, vídeo, texto, áudio etc.
- **Estação JavaScript:** criada basicamente para atuar com validações simples do lado cliente-servidor, a fim de garantir interações em tempo real sem a necessidade, por exemplo, de envio de dados para um servidor.
- **Configuração de estação de trabalho:** instalação/configuração de IDE, do inglês, integrated development environment (ambiente de desenvolvimento integrado), visual studio code e visual studio community.

- **Configuração de banco de dados:** instalação/configuração do banco de dados sugerido para estudo, o SQLite.

Ainda passaremos por outras estações que irão agregar mais conteúdo ao seu aprendizado.

- **MVC** (model-view-controller).
- **JSON** (JavaScript object notation, notação de objetos JavaScript).
- **XML** (extensible markup language).
- **Bootstrap**.
- **Editores**.

Vamos separar o desenvolvimento web nas duas etapas descritas a seguir.

- **Front-end (desenvolvimento da interface do site):** nesta etapa, abordaremos conceitos de desenvolvimentos de página utilizando recursos de HTML, CSS e JavaScript.
- **Back-end (desenvolvimento e implementação da regra de negócio do site):** nesta etapa, abordaremos conceitos de desenvolvimento e de aplicação de regras de negócio utilizando a linguagem de programação C#.

Vale ressaltar que todo fundamento de conhecimento técnico que envolve o desenvolvimento de software – para desktop, mobile ou web – tem como canal de entrada o computador, que, em geral, apresenta as vantagens a seguir.

- Elevada velocidade de processamento de dados.
- Grande capacidade de armazenamento de dados e informações.
- Execução de longas sequências de instruções.
- Realização de tarefas repetitivas com rapidez e precisão.

Na figura a seguir, podemos verificar um cenário sistêmico, que rege até hoje o princípio de todo e qualquer desenvolvimento. Observamos um ciclo de interação entre:

- usuários, que fornecem dados de entrada;
- computadores, que fornecem o processamento dos dados e o resultado esperado.

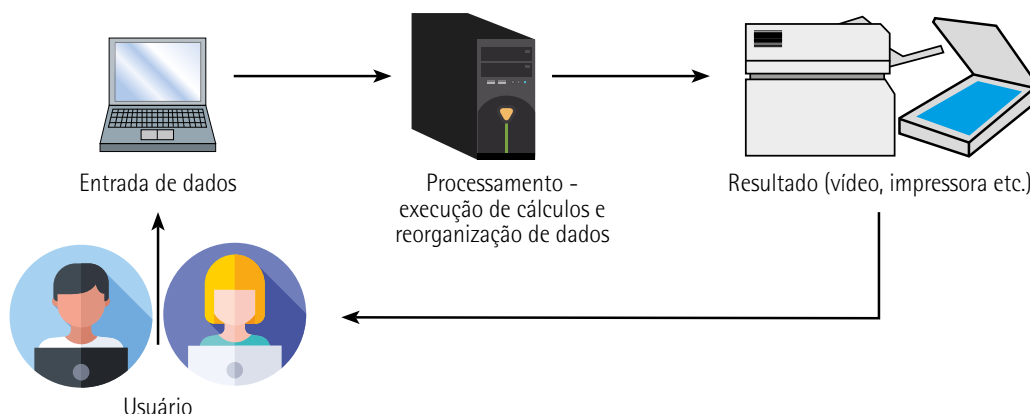


Figura 1 – Esquema simplificado (computador)

Nossa aprendizagem será voltada para três itens de extrema importância para o desenvolvimento web: HTML, CSS e JavaScript.

Faremos a integração desses itens para que possamos conhecer melhor a linguagem de marcação e enriquecer o layout com estilos aplicados, conforme apresentado na figura a seguir.



Figura 2 – Base para o desenvolvimento web

Vale destacarmos o que segue.

- **HTML 5:** é a linguagem de marcação que exibe, estrutura e organiza nossa informação. É a base da nossa programação web. Abordaremos a versão 5 dessa linguagem.
- **CSS 3:** é a linguagem de estilos responsável por formatar a aparência da informação, ou seja, deixar a página mais atraente. Com ela, podemos separar a parte estrutural da aplicação (que fica a cargo do HTML) da parte estética. Abordaremos a versão 3 dessa linguagem.
- **JS (JavaScript):** é responsável por manipular a nossa estrutura por meio de eventos de click ou de eventos de carregamento de página. Comanda as funcionalidades dinâmicas das nossas páginas web.

Falaremos sobre uma arquitetura que cresce a cada dia, chamada de MVC, *model-view-controller* (modelo-visão-controle), que, segundo Lotar (2011), pode ser compreendida como um padrão de projeto de software ou um padrão de arquitetura de software. Esse padrão, originado em meados da década de 1970, tem como principal objetivo o reuso de código e a separação de conceitos em três camadas interligadas, conforme apresentado na figura a seguir.

Lotar (2011) definiu essa arquitetura com os seguintes componentes:

- model, que contém o código da camada de dados;
- controller, que recebe as requisições do usuário;
- view, que implementa o design da aplicação.

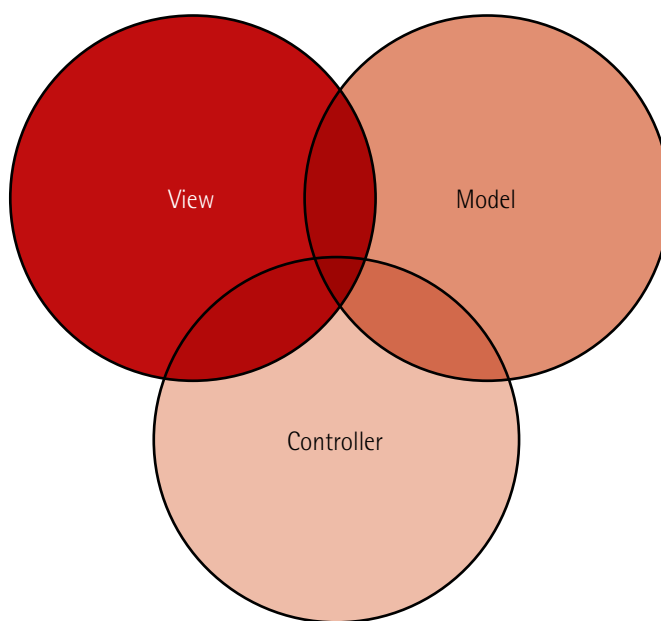


Figura 3 – Arquitetura MVC

Também discorreremos sobre a utilização do protocolo HTTP, *hypertext transfer protocol* (protocolo de transferência de hipertexto), que é a base da troca de dados na web. A API da web do ASP.NET, de acordo com Manzano e Toledo (2010) e Janones (2010), é uma estrutura para a criação de APIs da web sobre o .NET para servir dados e serviços. Os autores também abordam diferenças entre API web, serviço *rest* WCF e API web ASP.NET e discorrem sobre a criação estruturada de um passo a passo de uma API web ASP.NET e sobre as operações CRUD (figura a seguir), *create*, *read*, *update*, *delete* (criar, ler, atualizar e excluir).



Figura 4 – CRUD

Na figura a seguir, ilustramos a importância da Web API sobre aplicações web, mobile e desktop, que serão destacadas mais adiante.

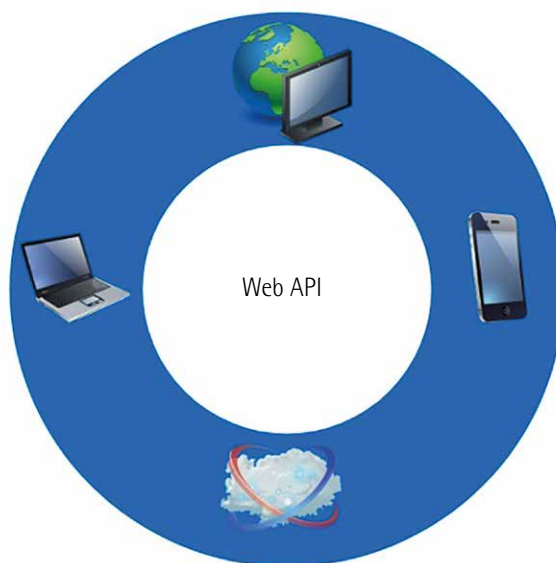


Figura 5 – Web API

Nossa intenção, neste livro-texto, é alinhar o direcionamento teórico a exemplos práticos e didáticos, a fim de que você assimile o desenvolvimento aplicado ao mundo web.

Bom estudo!

Unidade I

1 MONTANDO O AMBIENTE DE ESTUDO

1.1 Editor Sublime Text

Para que você desfrute integralmente do conteúdo apresentado neste livro-texto, recomendamos a escolha de um editor de texto. Ele será responsável por alocar todo o código apresentado e, principalmente, por criar uma familiaridade ainda maior com a linguagem de marcação HTML. Sugere-se, como ferramenta aos estudos aplicados no livro-texto, a instalação do editor Sublime Text – existe, porém, uma infinidade de editores no mercado. Ainda neste livro-texto, vamos apresentar mais um editor, o VSCode.

É importante que os exemplos apresentados em cada item sejam salvos em uma pasta em seu computador para que você consiga estudar e entender os conceitos aqui expostos. Também é importante que você crie comentários que possibilitem lembrar a aplicação dos conceitos quando você visitar o código.

Para criar nosso ambiente de estudos, precisamos seguir os passos indicados a seguir.

- Acesse o site do editor Sublime Text, no endereço sublimetext.com, para realizar o download (utilizado para transmitir dados de um dispositivo ao outro, usando um canal de comunicação previamente estabelecido no ambiente), conforme apresentado na figura.

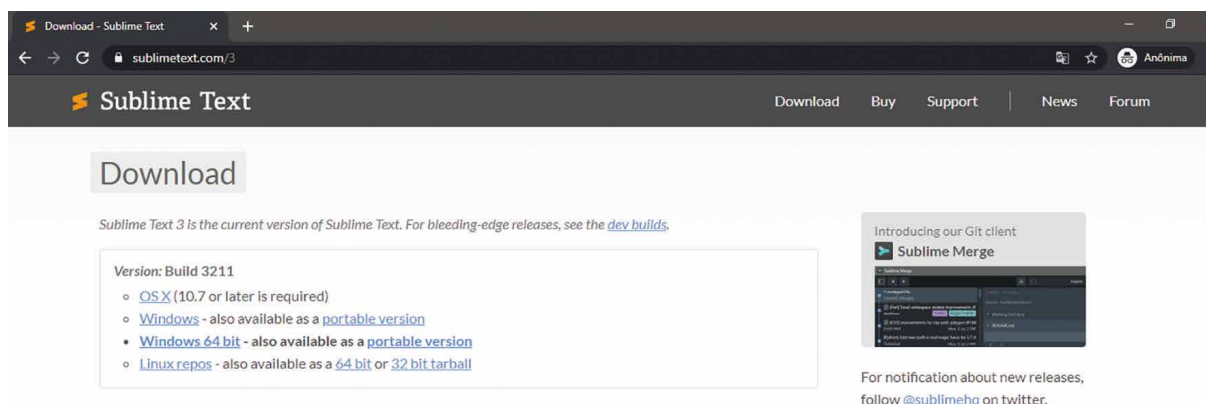


Figura 6 – Demonstrativo do site Sublime Text

- Escolha o link para efetuar o download conforme o seu sistema operacional. Há as seguintes versões (Build 3211):
 - OS X (10.7, ou posteriormente será necessário).
 - Windows (também disponível como uma versão portátil).
 - Windows 64 bits (também disponível como uma versão portátil).
 - Linux Repository (também disponível como *tarball* de 64 ou 32 bits).
- Após realizar o download, inicie a instalação do Sublime Text. Geralmente, o arquivo fica na pasta Downloads, para quem usa o sistema operacional Windows, ou em Transferências, para quem usa OS X.



Observação

Windows é um sistema operacional desenvolvido pela Microsoft.

OS X (Mac OS) é um sistema operacional desenvolvido pela Apple.

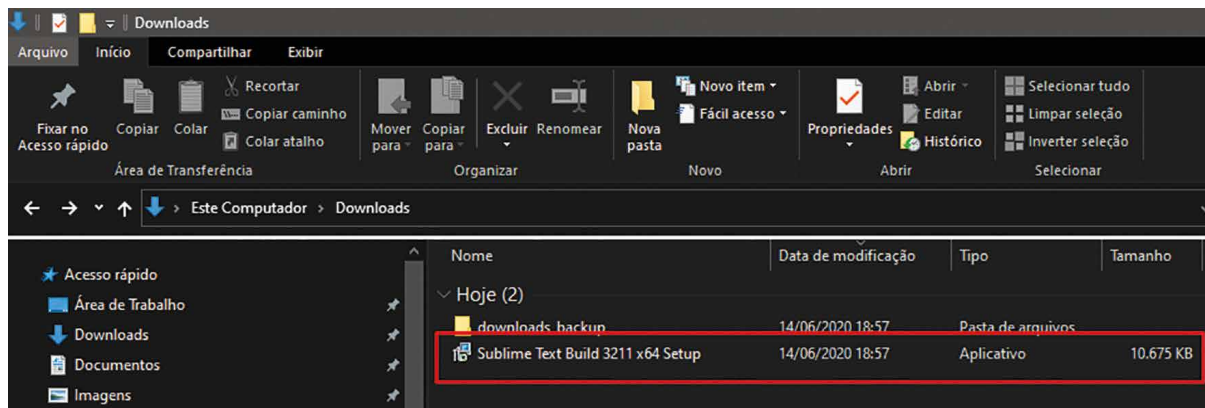


Figura 7 – Download efetuado na pasta Downloads

- Após realizar o download do Sublime Text, faça um duplo clique no ícone Sublime Text Build 3211 x64 Setup. Assim, será inicializada a instalação do editor, conforme apresentado na figura a seguir.

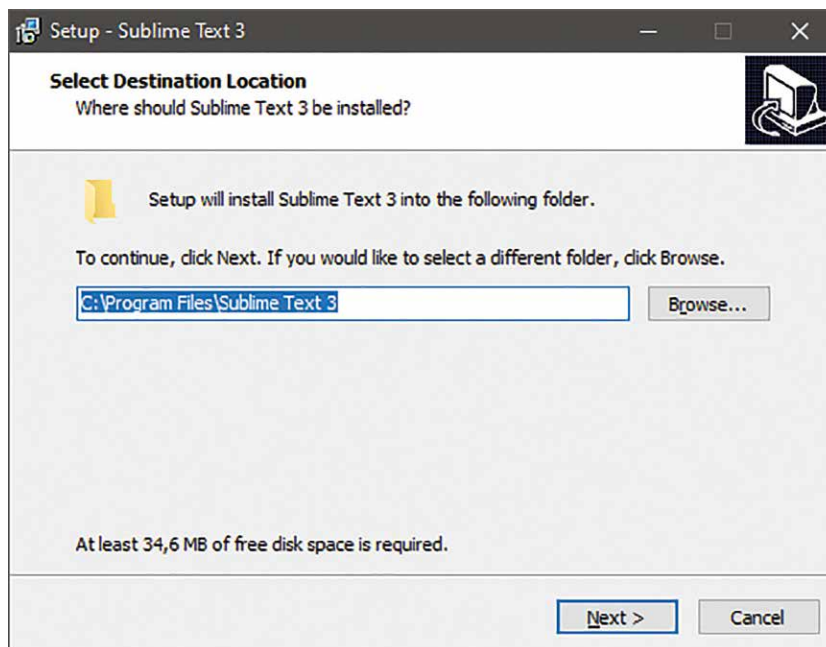


Figura 8 – Inicialização da instalação do Sublime Text

- Após acionar o botão Next > (próximo), estaremos na próxima tela de definição de diretório de instalação, conforme apresentado na figura a seguir.

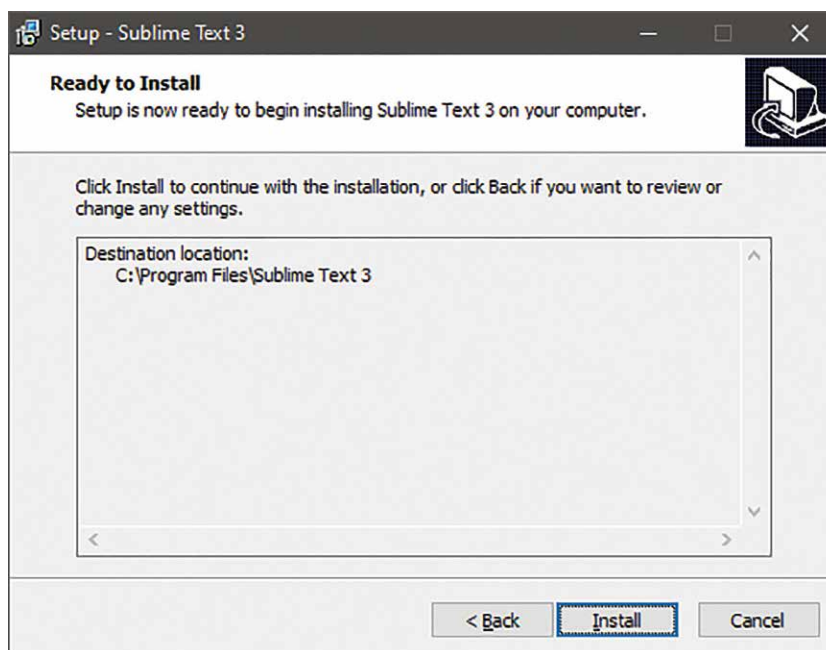


Figura 9 – Tela de instalação do Sublime Text

- Para iniciar a instalação do Sublime Text, acione o botão Install (instalar), como apresentado na figura anterior. Quando a instalação for finalizada, será mostrada a mensagem de sucesso na instalação, conforme figura a seguir.

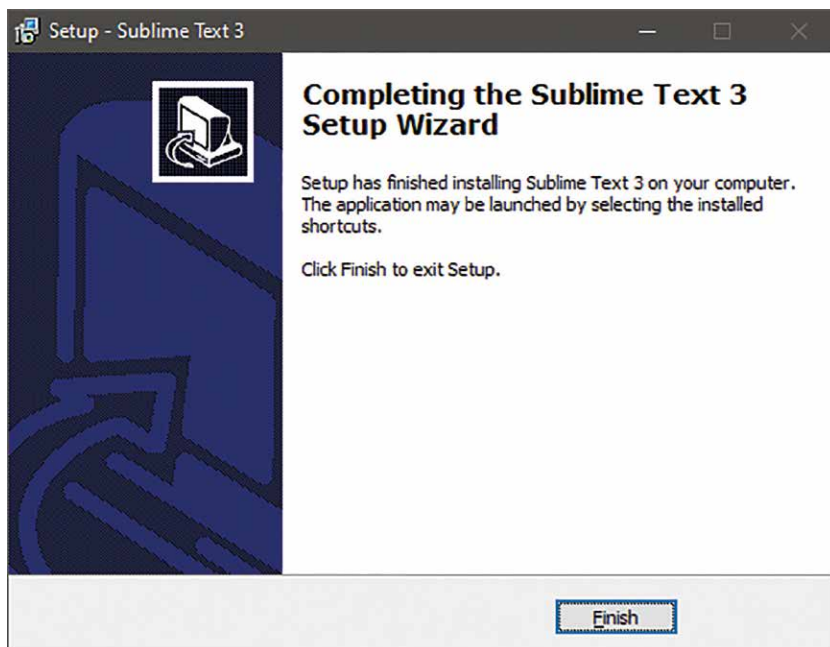


Figura 10 – Conclusão da instalação do Sublime Text

- Após a instalação ser concluída, podemos abrir o Sublime Text, como podemos ver na figura.

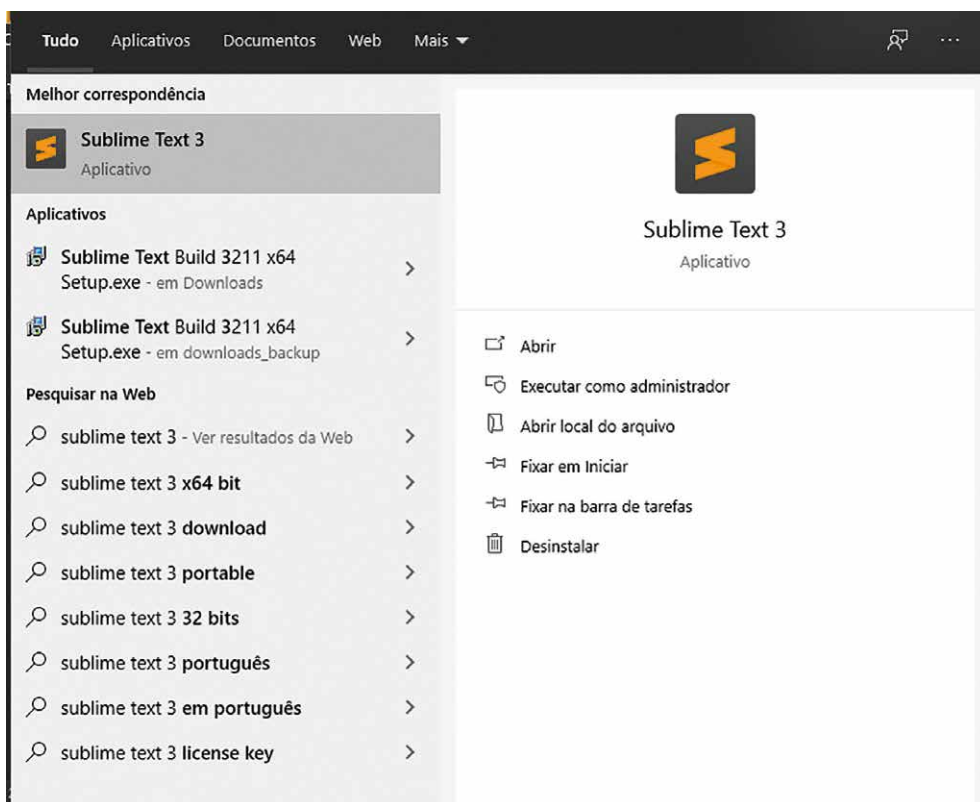


Figura 11 – Acessando o Sublime Text no computador com o sistema operacional Windows

É importante que os exemplos desenvolvidos sejam salvos em uma pasta no computador, de preferência acompanhados da extensão .html (exemplo: aula.html), conforme apresentado na figura a seguir.

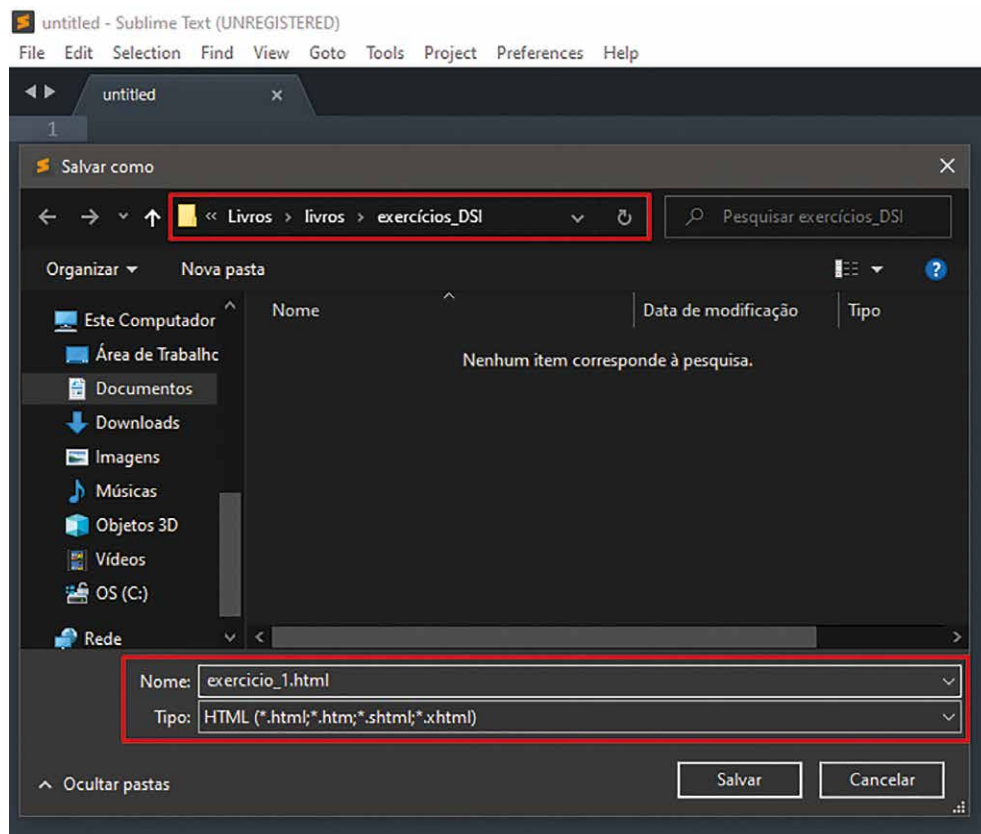


Figura 12 – Pasta criada para alocação dos exercícios

Vale indicar aqui um pequeno atalho para a construção da estrutura de página HTML, que acompanha as tags <HTML>, <head> e <body>, quando o arquivo HTML estiver aberto: digite <HTML>, aperte a tecla Tab do seu teclado e será disponibilizada a estrutura HTML de tags <HTML>, conforme pode ser visualizado nas figuras a seguir.

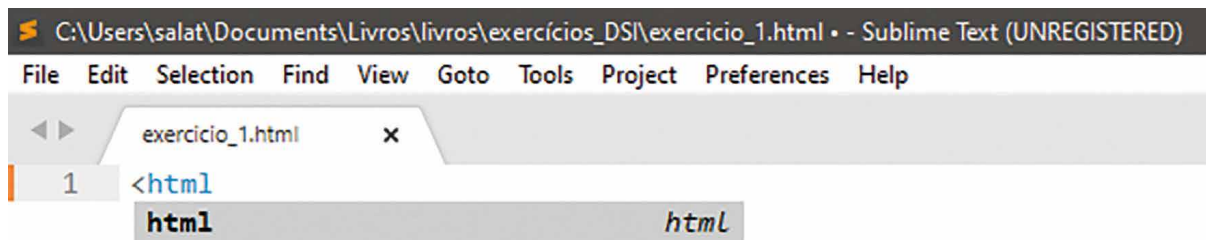
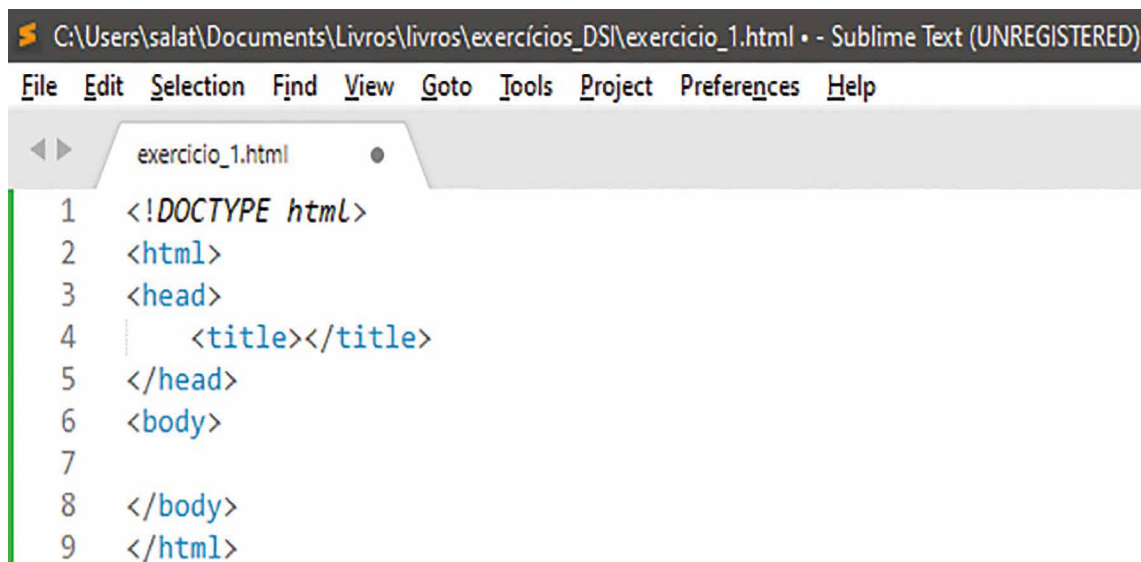


Figura 13 – Início da digitação da tag HTML



```
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7
8 </body>
9 </html>
```

Figura 14 – Estrutura inicial de uma página HTML



Observação

Tag significa etiqueta, rótulo, que pode ser definido como algo que marcamos para nos lembrar de informações importantes. A linguagem HTML utiliza tags em sua estrutura.

Após localizar o Sublime Text no computador, ele será visualizado conforme indicado na figura a seguir.

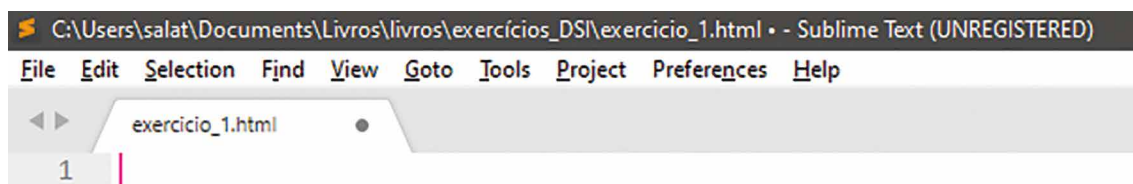


Figura 15 – Editor Sublime Text

Um dos facilitadores contidos no Sublime Text é a opção de preferências, que auxiliam na customização e visualização do software.

Para melhor visualização dos exemplos apresentados neste livro-texto, foi adotada a preferência Celeste (*Preferences Color Scheme*), conforme ilustrado na figura.

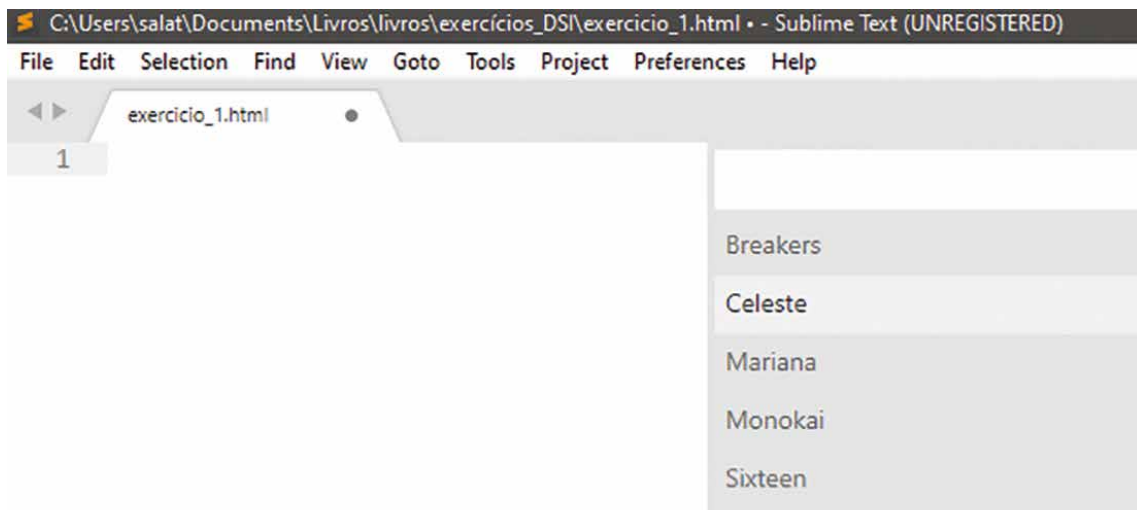


Figura 16 – Definição de cores do Sublime Text

No tópico 2.6.2 deste livro-texto, em que aprenderemos os comandos HTML, há um passo a passo que mostra a criação de uma página web simples utilizando como editor o Sublime Text. Por enquanto, vamos continuar com o conteúdo introdutório da nossa disciplina.



Lembrete

Um dos facilitadores contidos no Sublime Text é a opção de preferências, que auxiliam na customização e visualização do software.

1.2 Visual Studio Code

O Visual Studio Code não é apenas mais um bloco de notas desenvolvido com coloração de sintaxe e recuo automático. Em vez disso, é um ambiente de desenvolvimento focado em um código muito poderoso, projetado para facilitar a gravação de aplicativos da web, móveis e em nuvem, usando linguagens disponíveis para diferentes plataformas de desenvolvimento e dando suporte ao ciclo de vida de desenvolvimento de aplicativos com um recurso interno, depurador para Node.js, e com suporte integrado ao mecanismo de controle de versão Git.

Com o Code, é possível trabalhar com arquivos de código individuais ou sistemas de arquivos estruturados com base em pastas.

O diretório de trabalho é uma pasta no seu computador ou em um compartilhamento de rede. Cada desenvolvedor possui uma cópia completa de todo o código e dos arquivos associados. Podemos fazer todo o trabalho de desenvolvimento no diretório específico e deixar o Git gerenciar os aspectos de controle de versão para essa pasta.

O Git se refere a esse diretório de trabalho como um repositório ou um repositório para abreviar. Quando ele gerencia a pasta, nós obtemos todos os benefícios de controle de versão que oferece.



Saiba mais

Para mais informações sobre Git, acesse o site:

Disponível em: <https://github.com/>. Acesso em: 28 abr. 2022.

1.2.1 Por que o Visual Studio Code?

Antes de aprender a usar o Visual Studio Code, é útil entendermos seu objetivo, quais recursos ele oferece e como fornece uma experiência interessante de edição de código.

O Visual Studio Code não é um editor de código simples; é um ambiente poderoso que coloca a escrita de código no centro. O principal objetivo não é criar binários (como arquivos .exe e .dll), e sim facilitar a criação de códigos para plataformas da web, dispositivos móveis e nuvem, para qualquer desenvolvedor que trabalhe em diferentes sistemas operacionais, como Windows, Linux e OS X, mantendo-o independente dos ambientes de desenvolvimento proprietários. Para um melhor entendimento, veja o seguinte exemplo: pense no ASP.NET Core 1.0, a tecnologia de código aberto de plataforma cruzada capaz de executar no Windows, Linux e OS X que a Microsoft produziu para criar aplicativos web portáteis.

Forçar a criação de aplicativos web portáteis e de plataforma cruzada com o Microsoft Visual Studio 2019 causa dependência ao ambiente de desenvolvimento integrado (IDE).

Pode-se argumentar que o Visual Studio 2019 Community Edition é gratuito, mas é executado apenas no Windows. Por outro lado, embora certamente não pretenda substituir um ambiente mais poderoso e completo, como seu irmão principal, o Visual Studio Code, pode ser executado em uma variedade de sistemas operacionais e gerenciar diferentes tipos de projetos, além das linguagens de programação mais populares. Para fazer isso, ele fornece os recursos principais expostos a seguir.

- Suporte interno para codificação com vários idiomas, incluindo aqueles que normalmente utilizamos em cenários de desenvolvimento de plataforma cruzada, com recursos avançados de edição e suporte para idiomas adicionais via extensibilidade.
- Depurador interno para Node.js, com suporte para depuradores adicionais (como o Mono) via extensibilidade.
- Controle de versão baseado no mecanismo Git, que fornece uma experiência integrada de colaboração que suporta confirmações e ramificações de código. Trata-se da escolha adequada para uma ferramenta destinada a trabalhar com possivelmente qualquer idioma.

Para combinar adequadamente todos esses recursos em uma ferramenta, o Visual Studio Code fornece um ambiente de codificação diferente de outras ferramentas de desenvolvedor, como o Microsoft Visual Studio. De fato, o Visual Studio Code é um ambiente baseado em pasta, que facilita o trabalho com arquivos de código que não são organizados em projetos e oferece uma maneira unificada de trabalhar com diferentes idiomas. Partindo dessa premissa, o Code oferece uma experiência de edição avançada, com recursos comuns a qualquer idioma suportado, além de alguns recursos disponíveis para idiomas específicos.



Observação

Um software de plataforma cruzada é um programa (ou um aplicativo) que pode operar em várias plataformas ou sistemas operacionais.

1.2.2 Baixando e configurando o Visual Studio Code

Acesse o site do Visual Studio Code, no endereço code.visualstudio.com/download, e realize o download de acordo com seu sistema operacional, conforme apresentado a seguir.

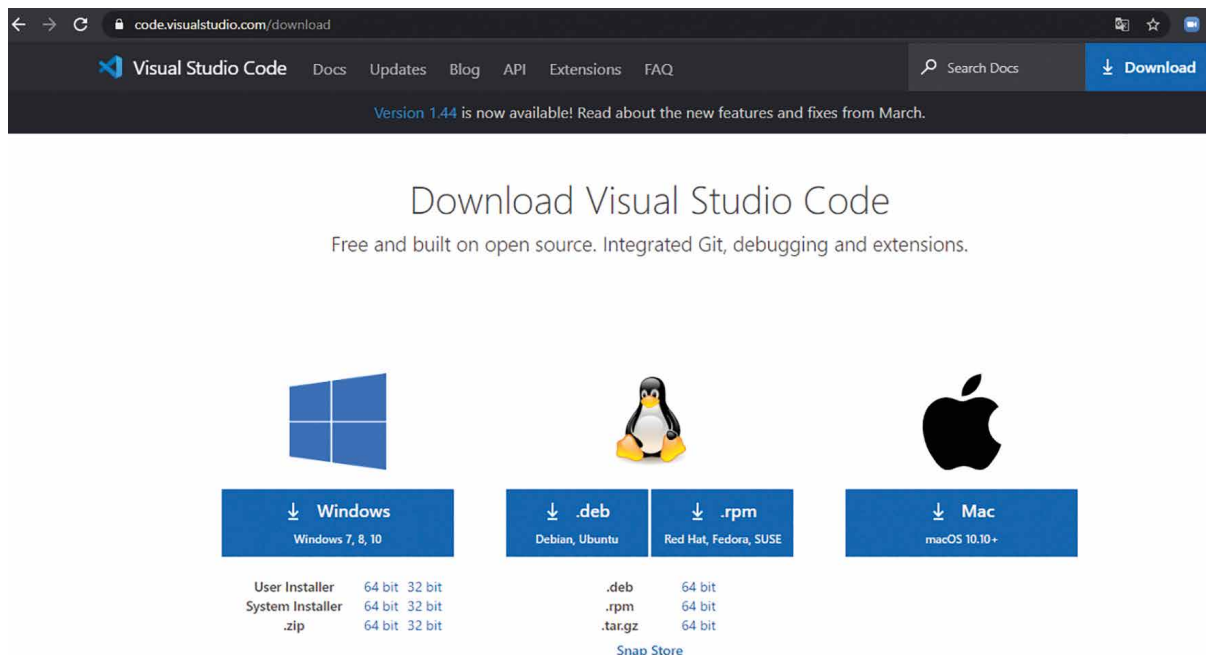


Figura 17 – Site do Visual Studio Code

Após a realização do download e instalação, será apresentado o IDE do Visual Studio, conforme ilustrado a seguir.

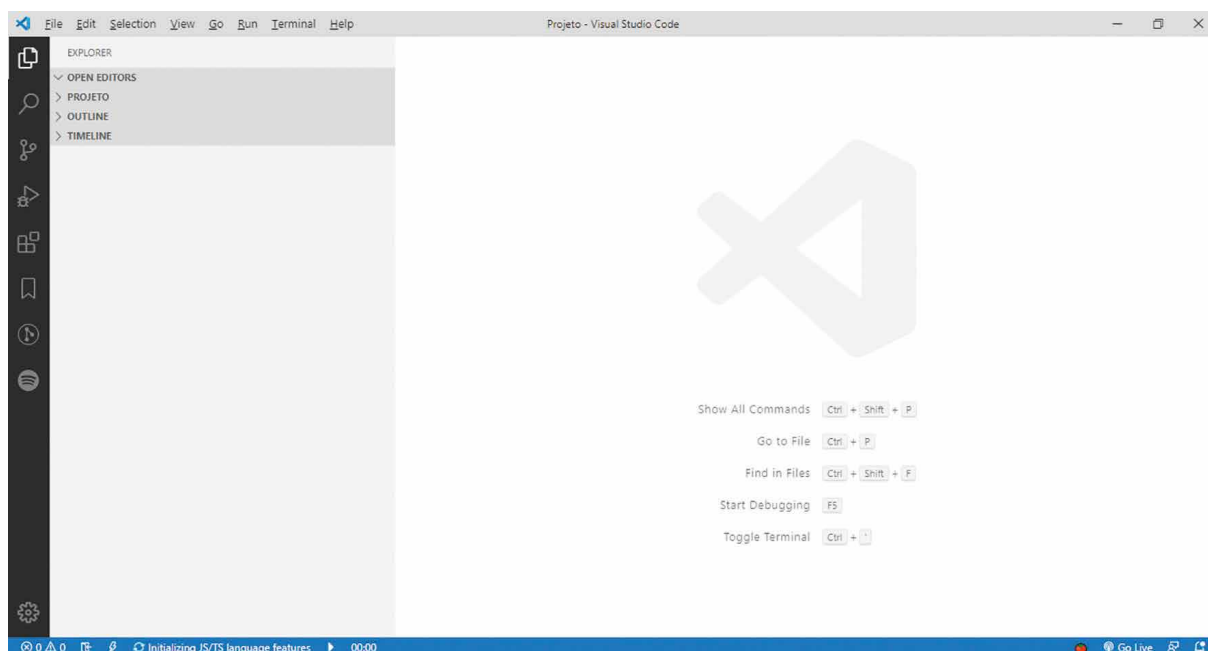


Figura 18 – Visual Studio Code

Na figura a seguir, é apresentada a estrutura de uma página HTML desenvolvida no Visual Studio Code.



Figura 19 – Estrutura HTML no Visual Studio Code



Observação

Todos os exemplos citados neste livro-texto poderão ser executados no Sublime Text, no Visual Studio Code ou no Visual Studio Community.

2 ESTAÇÃO HTML

2.1 Introdução à linguagem de marcação

A web é a plataforma de programação com maior relevância no mundo. Suas tecnologias principais, o HTML e o CSS, são fundamentais para o desenvolvimento de qualquer site. Quem aprender essas tecnologias, suas boas práticas e seus recursos mais avançados, sai na frente na busca de espaço no mercado de programação para *front-end* e pode participar de diversos projetos web.

Atualmente, todo produto digital necessita de um *front-end* web; por isso, estamos falando de um mercado com grandes oportunidades no mundo todo. No entanto, o bom profissional dessa área precisa lidar com inúmeros desafios, tais como saber as boas práticas de código, lidar com múltiplos navegadores e escrever sites precisos.

Você vai precisar estar atento às novidades do CSS, desenvolver sempre identificando oportunidades em aplicações em mobile e responsivo, assim como conhecer sobre acessibilidade e HTML semântico, que a cada dia vem ganhando mais força.

Segundo a W3C (W3C PRESS RELEASES, 2017), leva-se em consideração que a web é baseada nos três pilares indicados a seguir.

- Há a determinação de um esquema de nomes para a localização de fontes de informações na web. Esse esquema é denominado URI (do inglês, uniform resource identifier, identificador uniforme de recurso) e consiste em uma cadeia de caracteres compactada, utilizada para a identificação ou a denominação de um recurso na internet.
- Há a delimitação de um protocolo de acesso para visualizar as fontes, hoje denominado HTTP (do inglês, hypertext transfer protocol, protocolo de transferência), que possibilita às pessoas incluírem a URL (do inglês, uniform resource locator, localizador uniforme de recursos) de uma página (exemplo: www.unip.br) e verem o conteúdo que existe nela.
- Há uma linguagem de hipertexto, para agilizar na navegação entre fontes de informação, denominada HTML, que corresponde à apresentação das marcações atribuídas diante de páginas apresentadas no browser.



Saiba mais

Para saber mais a respeito do tema, sugere-se a leitura da história do W3C disponível no site a seguir.

W3C. *Sobre o W3C*. 2011b. Disponível em: <https://bit.ly/36WuzTT>. Acesso em: 28 abr. 2022.

2.2 Hipertexto e marcação

O HTML ainda pode ser definido como uma linguagem de marcação de hipertexto. Ele tem por finalidade ser uma linguagem de publicação de conteúdo, como imagens, áudios ou vídeos direcionados para a web.

Os elementos atribuídos ao hipertexto podem ser constituídos de palavras, imagens, áudios, vídeos e documentos, e podem ser conectados dentro de uma grande rede de informação.

Aqui, você pode se perguntar: por que uma linguagem universal? Considera-se que, desde o começo do HTML, seu intuito é o de ser uma linguagem independente de plataformas, browsers e qualquer outro meio de acesso, visto que a origem da linguagem era ser aplicada pensando em redução de custos, podendo até mesmo ser trabalhada com custo zero.

Sua construção foi pensada para que não houvesse nenhum tipo de barreira, ultrapassando os desafios que fossem impostos durante o seu ciclo de vida, fazendo com que, por exemplo, uma informação pudesse ser publicada por meio desse código e ficasse acessível por qualquer dispositivo e por meios com características distintas.

Há uma característica importante a ser ressaltada: podemos ter a implementação de HTML para dispositivos próprios destinados a pessoas com deficiência visual e auditiva ou para dispositivos móveis e portáteis, o que promove a inclusão social das pessoas.



Saiba mais

Para saber mais a respeito do tema, leia o texto indicado a seguir.

W3C. *GT acessibilidade*. 2011a. Disponível em: <https://bit.ly/3LvqVzs>. Acesso em: 28 abr. 2022.

Diante dos conceitos de HTML, ainda temos o que chamamos de **marcação**, que representa a notação de algum elemento contido no documento. Enfim, em HTML podemos definir a marcação por **tags**, mostrando, assim, a indicação da estrutura do documento.

Tags são definidas como marcas ou etiquetas que podem descrever de maneira isolada cada elemento dentro do texto. Logo, as tags em HTML possibilitam que o browser, de maneira interpretativa, faça a exibição da página conforme sua concepção, apresentando de modo fiel toda a sua construção e implementando todos os requisitos elaborados e desenvolvidos em HTML.

Podemos, então, definir o HTML como uma linguagem que possibilita realizar a marcação de hipertextos, basicamente com a finalidade de adicionar informações de maneira particular sobre as informações a serem apresentadas, ou seja, efetua a indicação da estrutura de um documento partindo do pré-requisito da utilização de tags.

Um item importante a ser levado em consideração é que um arquivo HTML precisa necessariamente conter a extensão htm ou HTML, podendo ser criado utilizando simplesmente um editor de textos (exemplo: Notepad, que acompanha a instalação do Windows). Os documentos ou os arquivos criados em HTML são denominados de páginas web.



Observação

Um editor de texto pode ser considerado um aplicativo de edição de arquivo de texto que tem por finalidade incluir, formatar e desenvolver. Editores de texto são frequentemente equipados com sistemas operacionais ou pacotes de desenvolvimento de software considerados *plug-ins* e também podem ser utilizados para alterar arquivos de configuração.

2.3 História do HTML

O conceito de internet nasceu de um projeto da agência norte-americana Advanced Research and Projects Agency (Arpa), que teve como objetivo conectar computadores dos seus departamentos de pesquisa. Assim, podemos dizer que a internet tem início com a Arpanet. Separar alguns instantes para estudar fundamentos históricos da internet é aprender com a evolução e se preparar para novas oportunidades e tecnologias que irão surgir.

A web teve como principal dependência o fato de diversos autores de páginas web e de inúmeras empresas compartilharem as mesmas convenções e atribuições da linguagem de marcação HTML. No entanto, podemos nos questionar a respeito do porquê de existir uma convenção para essa marcação. Isso seria simples de responder, pois, sem uma convenção, muito provavelmente essa linguagem estaria fadada ao fracasso, já que todo e qualquer segmento sem um padrão viraria uma bagunça.

Para que houvesse, então, uma convenção, foi necessário um trabalho árduo e em conjunto na criação de métricas e de especificações da linguagem de marcação HTML.

A partir da metade da década de 1990, criou-se o W3C, entidade que tem por finalidade ou objetivo específico o desenvolvimento e aprimoramento de padrões que possam ser consumidos na web, estabelecendo normas de base para o desenvolvimento da linguagem de marcação HTML, conforme apresentado na figura a seguir.

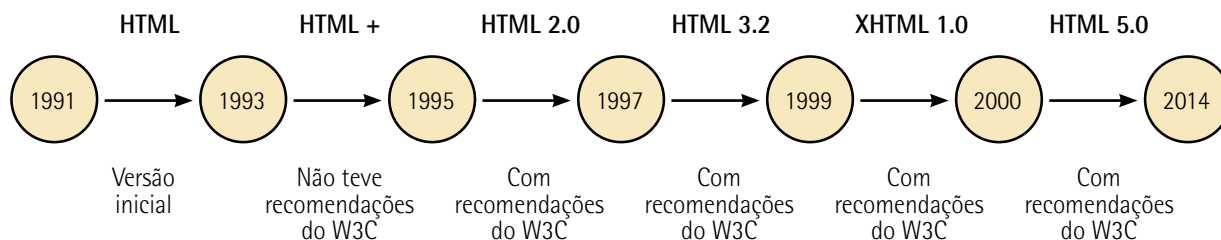


Figura 20 – Ciclo de vida do HTML

A cada atualização de versão do HTML, conforme apresentado no quadro a seguir, houve um consenso entre os industriais da época para que não houvesse nenhum desperdício do trabalho evolutivo e para que seus documentos não se tornassem ilegíveis ou mesmo inválidos mediante um período curto com a implementação de novas versões do HTML.

Quadro 1 – Ciclo de vida detalhado do HTML

Versão	Ano	Recomendações do W3C
HTML	1991	Versão inicial
HTML +	1993	Não houve recomendações
HTML 2.0	1995	Recomendações do W3C
HTML 3.2	1997	Recomendações do W3C
HTML 4.01	1999	Recomendações do W3C
XHTML 1.0	2000	Recomendações do W3C
HTML 5	2014	Recomendações do W3C

Assim, desde o começo, o HTML foi desenvolvido para ser uma linguagem de marcação independente de plataformas, browsers e outros meios de acesso, cujo trabalho consiste na exibição de páginas na web.

Porém, com esse trabalho, temos a possibilidade de realizar a criação apenas de um código de marcação HTML, que pode ser lido por diversos meios, em vez de versões diferentes para diversos dispositivos. Dessa forma, evitou-se que a web pudesse ser desenvolvida em uma base proprietária, limitando os formatos incompatíveis.

Com esse pensamento, vemos que o HTML nasceu para que todas as barreiras fossem ultrapassadas, fazendo com que as informações geradas diante de um código fossem apresentadas por dispositivos e outros meios com características distintas, como tamanho da tela, resolução ou até mesmo variação de cores. Podemos concluir que o HTML teve como principal objetivo ser entendido de maneira universal, possibilitando, assim, a reutilização da linguagem de marcação de acordo com as limitações de cada meio de acesso.

2.4 HTML5

Ao longo da evolução da tecnologia, de maneira gradativa, o W3C sinalizava aos desenvolvedores que o HTML4 já aplicaria os quesitos de boas práticas de desenvolvimento e, principalmente, com a estruturação e separação do código com sua formatação, colocaria como ponto de importância a acessibilidade, que foi inserida em diversas discussões e alertas aos fabricantes e programadores. Contudo, mesmo com os avanços tecnológicos e com as diversas implementações efetuadas, a evolução da linguagem de marcação do HTML4 não contemplava grandes diferenciais para a semântica. Um exemplo que pode ser citado, como meio que não facilitava o trabalho do programador, era a manipulação dos elementos JavaScript ou CSS, tornando seu trabalho oneroso.

Desse modo, o HTML5 nasceu com o objetivo de facilitar a manipulação do elemento, permitindo que o desenvolvedor modificasse as características dos objetos de forma clara e objetiva, de maneira que fosse transparente para o usuário final.

Uma das características do HTML5 que trazem melhorias em relação às versões anteriores é o fato de ele proporcionar ferramentas para o CSS e o JavaScript efetuarem o seu trabalho de forma eficaz.

2.5 Arquitetura HTML

Pode-se determinar a arquitetura do HTML com os elementos mostrados a seguir.

- **Interface do usuário:** camada responsável por fornecer determinadas funcionalidades, como o progresso visual para loading de páginas apresentadas no browser e a categorização de downloads inteligentes.
- **Motor do navegador:** camada responsável por disponibilizar uma ou mais interfaces para o motor de renderização. Também é responsável por efetuar o loading de documentos e, em determinados momentos, suportar determinadas ações do browser, como acionar o botão de voltar, avançar e até mesmo fazer a apresentação de páginas no browser.
- **Motor de renderização:** camada que tem por finalidade efetuar a produção e a reprodução visual de cada documento apresentado no browser, obtendo como definição que motores têm por finalidade renderizar informações alocadas em páginas HTML e documentos no formato XML.
- **Rede:** camada responsável por efetuar implementações dos protocolos que definem a transferência de dados, como HTTP e FTP (file transfer protocol), também responsável por interpretar diversos tipos de charsets (bibliotecas de caracteres) e mime types (tipos de documentos) que possam existir nesse contexto.
- **Interpretador JavaScript:** camada responsável por executar códigos escritos na linguagem JavaScript, sendo que alguns dos recursos utilizados em JavaScript, como abrir uma janela pop-juntar up, podem ser desabilitados pelo motor do browser ou até mesmo pelo motor de renderização, visto que aspectos de segurança podem ser relacionados à desativação.

- **Analisador XML:** camada responsável por atuar na transformação de documentos no formato XML em, por exemplo, uma árvore relacionada ao tipo DOM.
- **Apresentar back-end:** camada responsável por interagir com o sistema operacional para que, assim, possam ser garantidas a integridade e a funcionalidade para ações do tipo geração de desenhos e de janelas sistêmicas.
- **Persistir dados:** camada responsável por definir a alocação de informações no disco; no caso, trata-se dos dados que forem associados a uma ou mais sessões do browser do usuário. Logo, esses dados têm como base de configuração, por exemplo, a atribuição de itens à barra do browser a cookies. Há certificados de segurança.

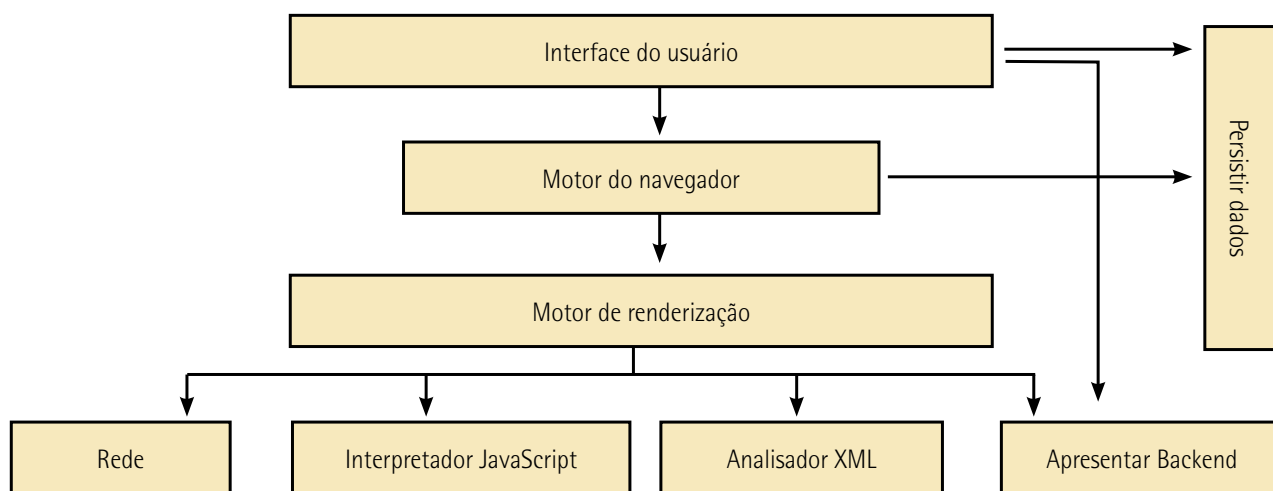


Figura 21 – Arquitetura de referência do browser

2.6 Motores de renderização

Conforme já abordado, um motor de renderização é o principal componente de software dos navegadores. Sua principal função é transformar arquivos HTML e outros recursos de uma página web em uma representação visual interativa no dispositivo do usuário. Por meio do motor, o navegador não exibe as tags HTML, mas as utiliza para interpretar o conteúdo da página.

Existe no mercado de web uma infinidade de motores de renderização que possibilitam que smartphones, tablets e outros tornem compatíveis os browsers de cada dispositivo, contribuindo, assim, para que os desenvolvedores mantenham o nível de compatibilidade de todos os browsers, levando em consideração a particularidade de cada um deles.

Para isso, encontrou-se um meio de assegurar de maneira compatível o código e encontrar um denominador que possa equiparar o desenvolvimento trabalhando com motores de renderização. Assim, cada browser tem por responsabilidade o processamento do código fonte de cada página que for requisitada a cada instante, conforme apresentado nos quadros a seguir.

Quadro 2 – Motores de renderização e browsers

Motor	Browser
Presto	Opera 7 ao 10
WebKit	Safari, Google Chrome
Trident	Internet Explorer 4 ao 9
Gecko	Firefox, Mozilla, Camino

Quadro 3 – Histórico dos motores de renderização

Motor de renderização	Data de lançamento	Versão inicial
Blink	3 de abril de 2013	Sem número
EdgeHTML	12 de novembro de 2001	12.0
Gecko	7 de dezembro de 1998	"Preview"
Presto	13 de novembro de 2002	1.0
Tasman	27 de março de 2000	0
Trident	Abril de 1997	Sem número
WebKit	7 de janeiro de 2003	48

A seguir, vamos conhecer as particularidades de cada motor.

Presto

Com implementação ainda em curso, o Presto foi o motor de renderização do navegador Opera por aproximadamente uma década, sendo desenvolvido pela Opera Software. Diante de muitos trabalhos realizados para a disseminação do motor Presto, foi liberada para a comunidade web em 28 de janeiro de 2003 a integração do Presto ao browser Opera 7.0 para Windows.

O Presto nasceu para entrar no lugar do motor Elektra, sendo muito utilizado nas versões de 4-6 do Opera. Logo, a página, ou parte dela, pode ser renderizada em diversos momentos em resposta a eventos como DOM e script, sendo que o Presto se encontra somente disponível como parte integrante do Opera ou de produtos relacionados.

Blink

Considerado um motor de browser web desenvolvido pela Google como parte integrante do projeto Chromium, o Blink foi colocado a público pela primeira vez em abril de 2013. Teve como principal elemento uma bifurcação do componente WebCore do WebKit, que basicamente iniciou sua utilização a partir do Chrome 28, do Opera 14 e do Yandex Browser.

A bifurcação não acarretou a adição de novos prefixos de CSS, porém essas alterações foram planejadas. Atualmente, o Blink continua convergindo de maneira relativamente similar ao WebCore.

EdgeHTML

O EdgeHTML é um mecanismo de renderização que teve seu lançamento como uma forma experimental no Internet Explorer 11, parte da compilação do Windows 10 Preview. Basicamente, era assim o motor principal utilizado pelo Microsoft Edge, até que o browser precisou ser refeito com base no Chromium, em 2019, e passou a utilizar o Blink.

Gecko

Considerado um motor de layout com código aberto, o Gecko é utilizado em aplicativos desenvolvidos pela Mozilla Foundation e pela Mozilla Corporation. É encapsulado no browser Firefox, que também tem como finalidade muitos outros projetos de software com código aberto, para agregar valor à comunidade web.

O Gecko teve a finalidade de suportar os padrões da internet da época de sua implementação e ser utilizado por diferentes aplicativos para apresentação de páginas da web. Trouxe também uma vasta e robusta web API, que possibilitou de maneira adequada uma variada gama de aplicações baseadas nos princípios da internet, como o browser.

Alguns padrões suportados pelo Gecko são indicados a seguir.

- HTML4, com suporte parcial ao HTML5.
- CSS estabelecido ao nível 2.1, porém com suporte parcial ao CSS3.
- XML 1.0.
- XHTML 1.0.



Observação

Uma web API pode ser considerada uma interface de programação de aplicações tanto para um servidor quanto para um browser, sendo que sua utilização tem por finalidade conseguir recuperar somente o valor requisitado em um banco de dados de um site.

Tasman

Considerado um motor de layout que teve como frente de desenvolvimento a Microsoft, para que fosse incluído na versão do Internet Explorer 5 para Macintosh, o Tasman é também uma tentativa de proporcionar melhoria aos padrões web, definidos assim pela World Wide Web Consortium.

O cientista da computação Tantek Çelik liderou a equipe que desenvolveu o Tasman, trabalhando como líder de desenvolvimento do software. Ele começou a ser utilizado como motor de layout para o MSN para Mac OS X e para o Office 2004 para Mac.

Trident

O Trident teve como concepção um componente de software para possibilitar que os desenvolvedores de software facilmente adicionassem a funcionalidade de navegação na web para seus próprios aplicativos, ou seja, apresentar através de uma interface COM para a edição de páginas web e o acesso a qualquer ambiente COM, como C++ e .NET. As funcionalidades do Trident tornaram-se disponíveis, proporcionando, assim, uma ligação entre o arquivo msHTML.dll e o projeto de software.

WebKit

O WebKit é um motor de renderização que teve sua utilização fundamentada em browsers para renderizar páginas, primeiramente em C++.

A princípio, foi desenvolvido pela Apple Inc. a partir da biblioteca KHTML para utilização no navegador Safari. Como é um projeto de código aberto, atualmente é mantido por desenvolvedores de diversas organizações, como Apple Inc., Google, Nokia e Samsung.

2.7 Codificação de caracteres e tabela ASCII

Vamos pensar em outro tema importante para os estudos de desenvolvimento de software para internet: os **padrões de codificação de caracteres** (também chamados de conjuntos de caracteres). Esses padrões permitem que o sistema computacional represente textos de sistemas de escrita existentes. Para exibir uma página HTML corretamente, um navegador da web deve conhecer o conjunto de caracteres utilizados na página. Vamos, a seguir, listar alguns padrões importantes.

- O ASCII foi o primeiro padrão de codificação de caracteres. Ele definiu 128 caracteres alfanuméricos diferentes, que poderiam ser usados na internet: números (0-9), letras em inglês (A-Z) e alguns caracteres especiais como !, \$, +, -, (,), @, <, >.
- O ISO-8859-1 foi o conjunto de caracteres padrão para HTML4. Esse conjunto de caracteres suportava 256 códigos de caracteres diferentes.
- O ANSI (Windows-1252) foi o conjunto de caracteres original do Windows. O ANSI é idêntico ao ISO-8859-1, exceto pelo fato de que o ANSI possui 32 caracteres extras. Se um navegador detectar ISO-8859-1 em uma página da web, o padrão será ANSI.
- O UTF-8 (8-bit Unicode Transformation Format) é um padrão de codificação que possibilita ampla cobertura de caracteres e símbolos. Por esse motivo, é considerado um tipo de codificação padrão para páginas web. Como o ANSI e o ISO-8859-1 eram muito limitados, o HTML4 também suportava UTF-8.

Figura 22 – Exemplo de aplicação UTF-8

O conjunto de caracteres ASCII utiliza valores, por exemplo, de 0 a 31 (também usa o 127), principalmente para caracteres de controle. Já o conjunto de caracteres ANSI utiliza a mesma sequência de caracteres do ASCII e agrega a sequência de 128 a 159, que se equipara ao UTF-8 para agregar valores de 160 a 255. O conjunto de caracteres ISO-8859-1 utiliza a mesma sequência do ASCII, não utiliza os valores 128 a 159 e agrega as sequências de valores 160 a 255, semelhantes ao UTF-8, apresentando algumas combinações e caracteres de exemplo, como ilustrado no quadro a seguir.

Quadro 4 – Lista de alguns caracteres ASCII

Alt + número	Descrição
33	Ponto de exclamação !
34	Aspa "
35	Sinal numérico #
36	Cifrão \$
37	Sinal de porcentagem %
38	Letra "e" comercial &
39	Apóstrofo '
40	Parêntese esquerdo (
41	Parêntese direito)
42	Asterisco *
43	Sinal de mais +
44	Vírgula ,
45	Hifen-menos –
46	Ponto-final .
47	Barra /



Saiba mais

A International Organization for Standardization (ISO), ou Organização Internacional de Padronização, é a entidade que promove a normalização de produtos e serviços e utiliza regras para que a qualidade seja melhorada. Conheça mais no site a seguir.

Disponível em: <https://www.iso.org/home.html>. Acesso em: 4 maio 2022.

Um bom exemplo de caso de estudo está nos problemas de apresentação de palavras, como em "pontua o" ou "pontuaÃ§Ã£o" em vez de "pontuação". Com isso, devemos armazenar novamente os dados com a codificação correta, ou até mesmo efetuar a troca da codificação de exibição da página. Para definir essa exibição, o cabeçalho HTML da página deve estar conforme o trecho apresentado nas figuras a seguir.


Para ISO-8859-1, temos o que segue.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Minha página - ADS</title>
5 </head>
6 <meta
7 http-equiv="Content-Type"
8 content="text/html"
9 charset="iso-8859-1">
10 </html>
```

Figura 23 – Exemplo de aplicação ISO-8859-1

Para UTF-8, temos o que segue.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Minha página - ADS</title>
5 </head>
6 <meta
7 http-equiv="Content-Type"
8 content="text/html"
9 charset="utf-8">
10 </html>
```

Figura 24 – Exemplo de aplicação UTF-8

Um fator de distinção entre o ISO-8859-1 e o UTF-8 é que o primeiro suporta até 256 caracteres (0 a 255, ou 0x00 a 0xFF) e o segundo suporta até 65.536 caracteres (0 a 65535, ou 0x0000 a 0xFFFF). Logo, definir o direcionamento de alguns programadores a utilizarem o ISO-8859-1 representa economia

de espaço, tráfego de rede e velocidade na exibição, no caso, de grandes quantidades de dados. Ao contrário da delimitação da condição da ISO-8859-1, é aconselhável, por padrão, o suporte aos idiomas de origem latina (português, espanhol, francês e italiano) e também inglês, alemão, dinamarquês e línguas dos demais países da Europa ocidental, conforme apresentado no quadro a seguir, com alguns caracteres que são utilizados nos idiomas citados anteriormente.

Quadro 5 – Acentuação

Descrição	Caractere	Representação do caractere
A maiúsculo com acento agudo	Á	Á
E maiúsculo com acento agudo	É	É
I maiúsculo com acento agudo	Í	Í
O maiúsculo com acento agudo	Ó	Ó
U maiúsculo com acento agudo	Ú	Ú
A minúsculo com acento agudo	á	á
E minúsculo com acento agudo	é	é
I minúsculo com acento agudo	í	í
O minúsculo com acento agudo	ó	ó
U minúsculo com acento agudo	ú	ú
A maiúsculo com acento circunflexo	Â	Â
E maiúsculo com acento circunflexo	Ê	Ê
O maiúsculo com acento circunflexo	Ô	Ô
A minúsculo com acento circunflexo	â	â
E minúsculo com acento circunflexo	ê	ê
O minúsculo com acento circunflexo	ô	ô
A maiúsculo com crase	À	À
A minúsculo com crase	à	à
U maiúsculo com trema	Ü	Ü
U minúsculo com trema	ü	ü
C com cedilha maiúsculo	Ç	Ç
C com cedilha minúsculo	ç	ç
A com til maiúsculo	Ã	Ã
O com til maiúsculo	Õ	Õ
A com til minúsculo	ã	ã
O com til minúsculo	õ	õ
N com til maiúsculo	Ñ	Ñ
N com til minúsculo	ñ	ñ
E comercial (<i>ampersand</i>)	&	&
Aspa dupla	"	"
Aspa simples	'	'
Menor que	<	<
Maior que	>	>

2.8 Tags HTML

O que são tags HTML? Uma tag é o conjunto dos caracteres `<`, `/`, `>` mais uma nomenclatura, que ainda pode conter atributos que possibilitam a sua customização. As tags também podem ser chamadas de marcas.

Tags HTML podem ser definidas como palavras-chave ocultas em uma página da web que definem como o navegador deve formatar e exibir o conteúdo. Elas delimitam um elemento HTML. A maioria dos elementos deve ter duas tags, uma de abertura e uma de fechamento. Por exemplo:

- `<HTML>` é a marca de abertura do elemento;
- `</HTML>` é a marca de fechamento do elemento.

Observe que a marca de fechamento possui o mesmo texto da marca de abertura, mas com um caractere de barra (`/`) adicional. Costuma-se interpretar isso como **final** ou **próximo**.

Um elemento, portanto, é uma entidade do código HTML delimitada por uma tag de abertura e uma tag de fechamento. Observe o elemento HTML a seguir.

```
<nomedatag>Conteúdo do elemento</nomedatag>
```

Ao longo deste livro-texto, vamos eventualmente nos referir aos elementos chamando-os apenas por sua tag de abertura. Por exemplo, o elemento fictício citado aqui pode ser chamado de elemento `<nomedatag>`.

Existem elementos tratados como exceção à regra, pois não necessitam do seu fechamento. O elemento ``, utilizado para exibir imagens em páginas web, é um exemplo disso. Cada arquivo HTML deve ter as tags essenciais para que seja válido, a fim de que os navegadores da web possam entendê-lo e exibi-lo corretamente. O restante do arquivo HTML pode conter o mínimo ou o máximo de tags que você deseja exibir em seu conteúdo.

2.8.1 Atributos de tag

Atributos são informações que passamos na tag para que ela se comporte da forma esperada. Em outras palavras, os atributos possibilitam personalizar um elemento. Eles são definidos na tag de abertura, conforme apresentado na figura a seguir.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Minha página - ADS</title>
5 </head>
6 <body>
7 
8 <p align="center"></p>
9 </body>
10 </html>

```

Figura 25 – Exemplo de aplicação tag

Os atributos são frequentemente atribuídos a um valor usando o sinal de igualdade, tal como em `border="0"` ou `width="50%"`. No entanto, há alguns que só precisam ser declarados na tag, como em `<hr noshade>`.

A maioria dos atributos é opcional para a maioria dos elementos e são usados apenas quando você deseja alterar algo sobre a maneira padrão como um elemento é exibido pelo navegador. Contudo, alguns elementos, como ``, precisam de atributos como `src` e `alt`, necessários para que o navegador exiba a página da web corretamente.

2.8.2 Tags essenciais de um arquivo HTML

Existem quatro pares de tags principais, que formam a estrutura básica de cada arquivo HTML, conforme apresentado na figura a seguir.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Minha página - ADS</title>
5 </head>
6 <body>
7 </body>
8 </html>

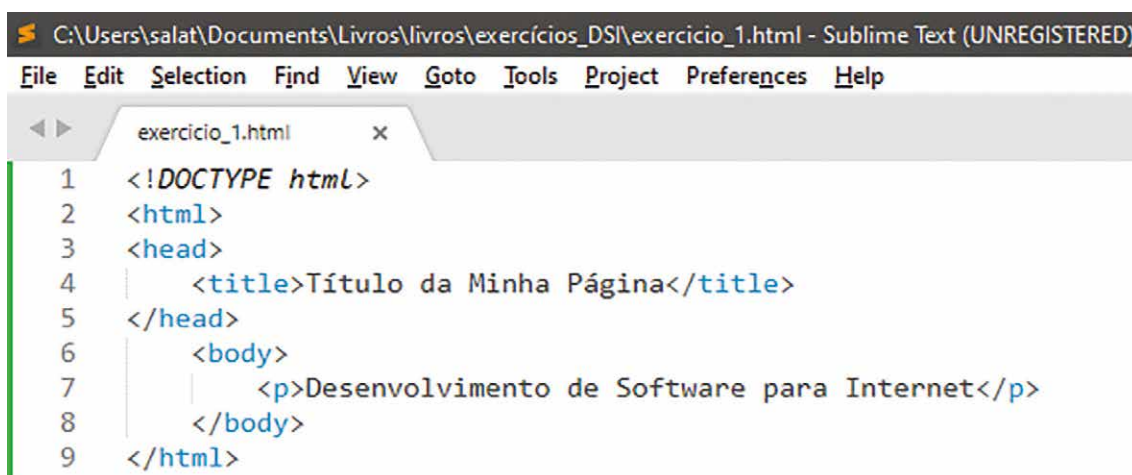
```

Figura 26 – Exemplo de estrutura de página HTML

- **Definição** – `<html> </html>`: esse par de tags define o documento como página da web. Também identifica o início e o fim do documento HTML. Todas as outras tags da sua página devem estar entre as tags HTML.
- **Cabeçalho** – `<head> </head>`: o elemento `<head>` contém informações sobre o documento que não apareceram na página real, como o título, o autor, a folha de estilo a ser usada e as metatags, que são tags utilizadas para descrever o conteúdo de uma página.
- **Título** – `<title> </title>`: esse elemento define o título que aparecerá na barra de título do seu navegador. O elemento `<title>` deve aparecer entre as tags `<head>` e `</head>`.
- **Corpo** – `<body> </body>`: o elemento `<body>` contém todas as informações e outros conteúdos visíveis na página. Todas as suas imagens, os links e os textos sem formatação devem estar entre as tags `<body>` e `</body>`.

Esses quatro pares de tags são especiais. Deve haver apenas um conjunto delas, e cada uma deve estar na ordem correta, como no exemplo ilustrado na figura a seguir. A parte divertida e criativa está no uso das tags básicas para adicionar conteúdo e títulos.

Em seguida, é apresentado um documento básico em HTML compreendendo todas as tags essenciais.

A screenshot of a Sublime Text editor window. The title bar shows the file path "C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The editor window has a tab labeled "exercicio_1.html". The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6   <body>
7     <p>Desenvolvimento de Software para Internet</p>
8   </body>
9 </html>
```

Figura 27 – Exemplo de estrutura de página HTML

Existem outros elementos que podem ser utilizados para adicionar conteúdo nas páginas, como os mencionados a seguir.

- **Imagens:** ``
- **Títulos e subtítulos:** `<title>`
- **Parágrafos do texto:** `<p>`
- **Links para outras páginas:** `<a href>`

As aplicações dos elementos citados serão apresentadas nos tópicos posteriores, assim como a funcionalidade da linha `<!DOCTYPE HTML>`, que vemos no início do código.

Caso você ainda não o tenha feito, vamos, neste momento, criar a nossa primeira página web, cujo código HTML será idêntico ao da figura anterior. Para isso, utilizaremos o editor Sublime Text. Siga os passos a seguir.

- Crie uma pasta na sua máquina destinada aos seus arquivos HTML.
- Abra o aplicativo Sublime Text. Caso você ainda precise fazer a instalação do editor de texto, antes de prosseguir neste passo a passo, volte ao tópico 1.1 e siga os passos lá descritos.
- Com o editor aberto, clique em "File" > "Save As..."
- A janela "Salvar Como" deve estar aberta na sua tela. Selecione a pasta de destino, digite o nome `exercicio_1.html` para o arquivo e escolha o tipo HTML, conforme mostra o exemplo:

Nome: `exercicio_1.HTML`

Tipo: HTML (*.HTML;*.htm;*.sHTML;*.xHTML)

- Clique em "Salvar".
- Você deve voltar para a tela de edição de texto do Sublime Text, já enxergando o nome do arquivo na aba superior. Nessa tela, digite `<html` e, em seguida, pressione a tecla TAB.
- Neste momento, pelo menos a seguinte sequência de comandos deve aparecer na sua tela:

```
<!DOCTYPE HTML>
<html>
<head>
  <title></title>
</head>
<body>
</body>
</html>
```



Observação

Essa é a estrutura básica do HTML. Dependendo do seu ambiente de programação, tags adicionais podem aparecer ao pressionar TAB, como `<meta charset="utf-8">`. Nesse caso, o atributo **charset**, ao qual foi atribuído o valor `utf-8`, indica o padrão de codificação de caracteres utilizado no documento.

- No cabeçalho, vamos indicar o título da página como "Título da Minha Página". Para isso, escreva tal título entre as tags <title> e </title>. Note que elas já se encontram dentro do cabeçalho, ou seja, posicionadas entre as marcas <head> e </head>. Nosso cabeçalho deve ficar da seguinte forma:

```
<head>
  <title>Título da Minha Página</title>
</head>
```

- Caso haja a linha <meta charset="utf-8"> dentro do seu cabeçalho, deixe-a como está.
- Agora, vamos inserir um parágrafo relativo à nossa disciplina no corpo da página. Para isso, digite "Desenvolvimento de Software para Internet" entre as marcas <p> e </p>, que, por sua vez, devem ser inseridas entre as marcas <body> e </body>. Com isso, devemos ter o seguinte texto no nosso corpo:

```
<body>
  <p>Desenvolvimento de Software para Internet</p>
</body>
```

- Clique em "File" > "Save", ou digite CTRL+S, para salvar o arquivo.
- Uma das maneiras de executar seu arquivo HTML é abrir a pasta onde ele foi salvo e executá-lo com um duplo clique. Dessa forma, ele deverá abrir no seu navegador padrão. Note que o título da página contém a inscrição "Título da Minha Página" e que, no corpo da página, aparece apenas o texto "Desenvolvimento de Software para Internet", conforme a figura a seguir.

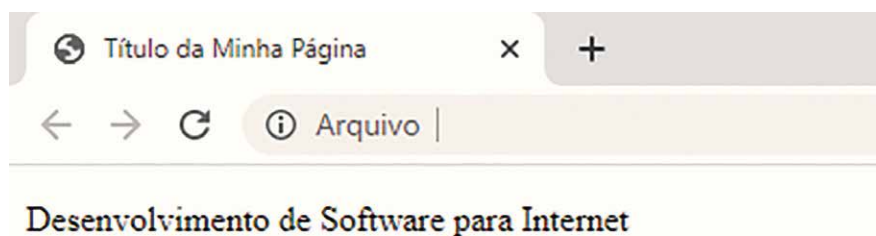


Figura 28 – Página HTML executada no navegador

A partir de agora, você será encorajado a criar arquivos HTML com os exemplos mostrados nos tópicos seguintes, sempre testando os resultados em seu navegador.

2.8.3 Cabeçalhos

O elemento <h> é utilizado para títulos, legendas e destaques de textos. Existem seis níveis de títulos disponíveis, desde h1 (maior e mais importante título) até h6 (menor). Algumas tags são responsáveis pelo auxílio da formatação do texto, ajudando a melhorar a aparência da página web. Na figura a seguir, é ilustrada a aplicação do elemento <h1> para o texto "Desenvolvimento de Software para Internet".

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <h1>Desenvolvimento de Software para Internet</h1>
8 </body>
9 </html>

```

Figura 29 – Exemplo de aplicação das tags <h1> e </h1>

2.8.4 Negrito:

O texto "Desenvolvimento de Software para Internet", que se encontra entre as tags e na figura a seguir, ficará em **negrito** e se destacará do texto ao redor.

```

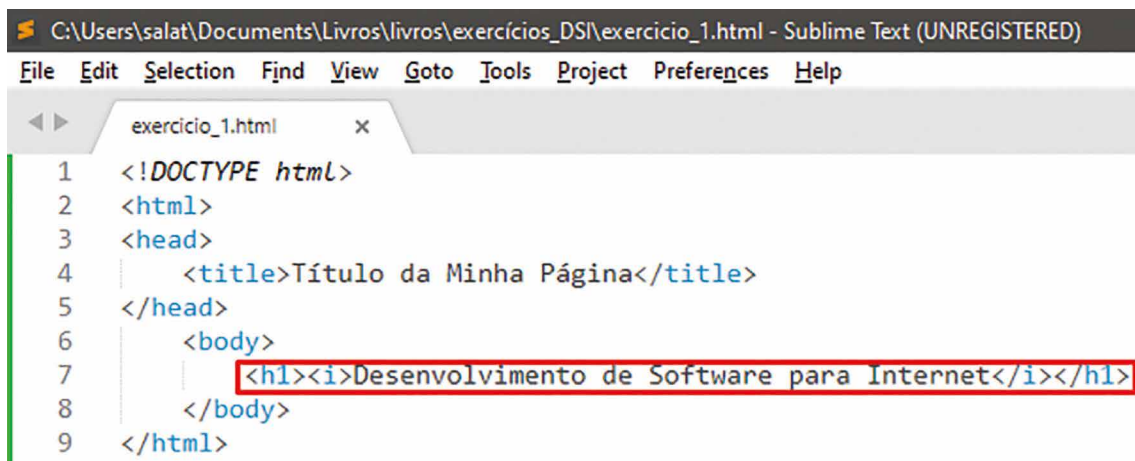
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <h1><b>Desenvolvimento de Software para Internet</b></h1>
8 </body>
9 </html>

```

Figura 30 – Exemplo de aplicação das tags e

2.8.5 Itálico: <i> </i>

O texto "Desenvolvimento de Software para Internet", que se encontra entre as tags <i> e </i> na figura a seguir, ficará em formato **itálico** e será exibido em um ângulo pequeno no navegador.

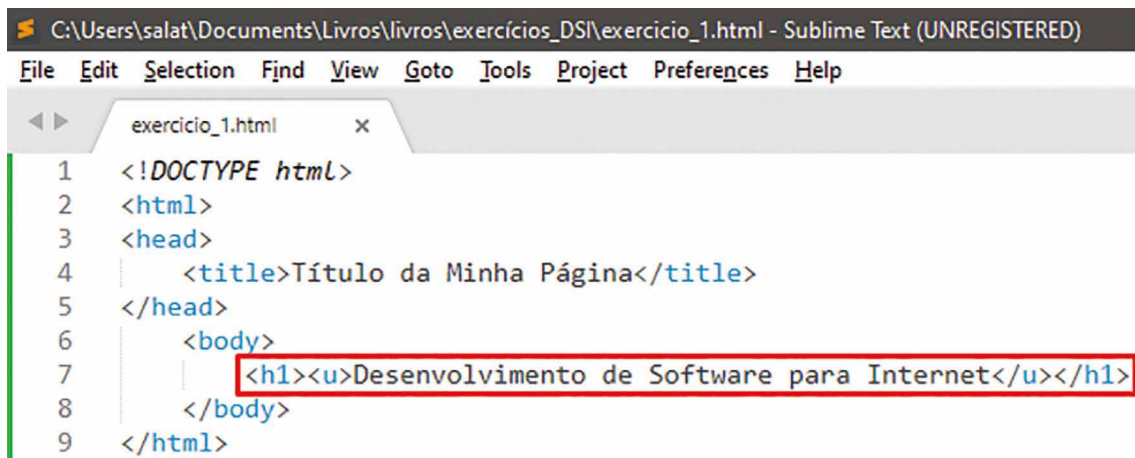


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <h1><i>Desenvolvimento de Software para Internet</i></h1>
8 </body>
9 </html>
```

Figura 31 – Exemplo de aplicação das tags <i> e </i>

2.8.6 Sublinhado: <u> </u>

O texto "Desenvolvimento de Software para Internet", que se encontra entre as tags <u> e </u> na figura a seguir, será apresentado de maneira **sublinhada** no navegador.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <h1><u>Desenvolvimento de Software para Internet</u></h1>
8 </body>
9 </html>
```

Figura 32 – Exemplo de aplicação das tags <u> e </u>

2.8.7 Pequeno: <small> </small>

Em vez de definir um tamanho de fonte, você pode utilizar o elemento <small> para renderizar um texto um pouco menor que o texto ao seu redor. Na figura a seguir, o texto "Desenvolvimento de Software para Internet" encontra-se entre as marcas <small> e </small>. Experimente escrever qualquer texto fora das tags <small> e </small>, para que você tenha um parâmetro de comparação, e teste em seu navegador.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <h1><small>Desenvolvimento de Software para Internet</small></h1>
8 </body>
9 </html>
  
```

Figura 33 – Exemplo de aplicação das tags <small> e </small>

2.8.8 Cor da fonte:

O elemento é utilizado para modificar a fonte do texto da página web. Por meio do atributo **color**, é possível alterar a cor de algumas palavras ou de uma seção de texto. Os itens na coluna "Código HEX HTML" representam o código de cores hexadecimal, conforme apresentado no quadro a seguir.

Quadro 6 – Códigos de cores HTML (valores hexadecimais)

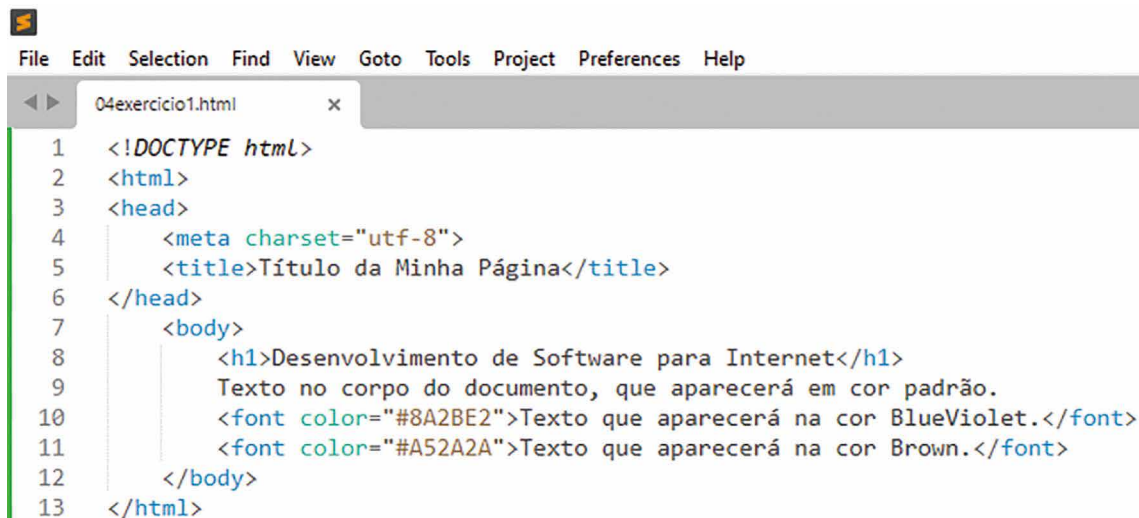
Nome da cor	Código HEX HTML
AliceBlue	#F0F8FF
AntiqueWhite	#FAEBD7
Aqua	#00FFFF
Aquamarine	#7FFFD4
Azure	#F0FFFF
Beige	#F5F5DC
Bisque	#FFE4C4
Black	#000000
BlanchedAlmond	#FFEBCD
Blue	#0000FF
BlueViolet	#8A2BE2
Brown	#A52A2A
BurlyWood	#DEB887
CadetBlue	#5F9EA0
Chartreuse	#7FFF00
Chocolate	#D2691E
Coral	#FF7F50
CornflowerBlue	#6495ED
Cornsilk	#FFF8DC
Crimson	#DC143C

Nome da cor	Código HEX HTML
Cyan	#00FFFF
DarkBlue	#00008B
DarkCyan	#008B8B
DarkGoldenRod	#B8860B
DarkGray	#A9A9A9
DarkGreen	#006400
DarkKhaki	#BDB76B
DarkMagenta	#8B008B
DarkOliveGreen	#556B2F
Darkorange	#FF8C00
DarkOrchid	#9932CC
DarkRed	#8B0000
DarkSalmon	#E9967A
DarkSeaGreen	#8FBC8F
DarkSlateBlue	#483D8B
DarkSlateGray	#2F4F4F
DarkTurquoise	#00CED1
DarkViolet	#9400D3
DeepPink	#FF1493
DeepSkyBlue	#00BFFF
DimGray	#696969
DodgerBlue	#1E90FF
FireBrick	#B22222
FloralWhite	#FFFAF0
ForestGreen	#228B22
Fuchsia	#FF00FF
Gainsboro	#DCDCDC
GhostWhite	#F8F8FF
Gold	#FFD700
GoldenRod	#DAA520
Gray	#808080
Green	#008000
GreenYellow	#ADFF2F
HoneyDew	#F0FF00
HotPink	#FF69B4
IndianRed	#CD5C5C
Indigo	#4B0082
Ivory	#FFFFF0
Khaki	#F0E68C
Lavender	#E6E6FA

Nome da cor	Código HEX HTML
LavenderBlush	#FFF0F5
LawnGreen	#7CFC00
LemonChiffon	#FFFACD
LightBlue	#ADD8E6
LightCoral	#F08080
LightCyan	#E0FFFF
LightGoldenRodYellow	#FAFAD2
LightGrey	#D3D3D3
LightGreen	#90EE90
LightPink	#FFB6C1
LightSalmon	#FFA07A
LightSeaGreen	#20B2AA
LightSkyBlue	#87CEFA
LightSlateGray	#778899
LightSteelBlue	#B0C4DE
LightYellow	#FFFFE0
Lime	#00FF00
LimeGreen	#32CD32
Linen	#FAF0E6
Magenta	#FF00FF
Maroon	#800000
MediumAquaMarine	#66CDAA
MediumBlue	#0000CD
MediumOrchid	#BA55D3
MediumPurple	#9370D8
MediumSeaGreen	#3CB371
MediumSlateBlue	#7B68EE
MediumSpringGreen	#00FA9A
MediumTurquoise	#48D1CC
MediumVioletRed	#C71585
MidnightBlue	#191970
MintCream	#F5FFFA
MistyRose	#FFE4E1
Moccasin	#FFE4B5
NavajoWhite	#FFDEAD
Navy	#000080
OldLace	#FDF5E6
Olive	#808000
OliveDrab	#6B8E23
Orange	#FFA500

Nome da cor	Código HEX HTML
OrangeRed	#FF4500
Orchid	#DA70D6
PaleGoldenRod	#EEE8AA
PaleGreen	#98FB98
PaleTurquoise	#AFEEEE
PaleVioletRed	#D87093
PapayaWhip	#FFEFD5
PeachPuff	#FFDAB9
Pink	#FFC0CB
Plum	#DDA0DD
PowderBlue	#B0E0E6
Purple	#800080
Red	#FF0000
RosyBrown	#BC8F8F
RoyalBlue	#4169E1
SaddleBrown	#8B4513
Salmon	#FA8072
SandyBrown	#F4A460
SeaGreen	#2E8B57
SeaShell	#FFF5EE
Sienna	#A0522D
Silver	#C0C0C0
SkyBlue	#87CEEB
SlateBlue	#6A5ACD
SlateGray	#708090
Snow	#FFFAFA
SpringGreen	#00FF7F
SteelBlue	#4682B4
Tan	#D2B48C
Teal	#008080
Thistle	#D8BFD8
Tomato	#FF6347
Turquoise	#40E0D0
Violet	#EE82EE
Wheat	#F5DEB3
White	#FFFFFF
WhiteSmoke	#F5F5F5
Yellow	#FFFF00
YellowGreen	#9ACD32

Para ver o efeito desse elemento HTML, teste o código apresentado na figura a seguir.



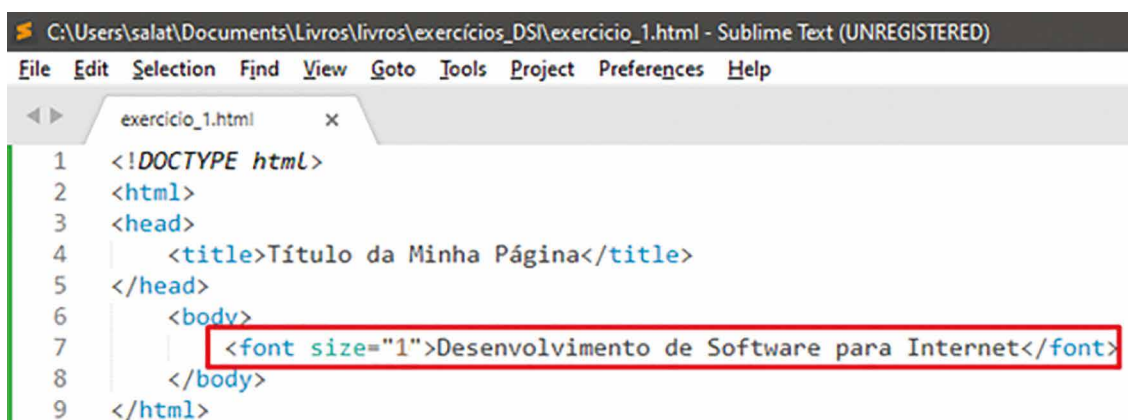
```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Título da Minha Página</title>
6  </head>
7      <body>
8          <h1>Desenvolvimento de Software para Internet</h1>
9          Texto no corpo do documento, que aparecerá em cor padrão.
10         <font color="#8A2BE2">Texto que aparecerá na cor BlueViolet.</font>
11         <font color="#A52A2A">Texto que aparecerá na cor Brown.</font>
12     </body>
13 </html>
    
```

Figura 34 – Exemplo de aplicação das tags e

2.8.9 Tamanho da fonte:

Ainda tratando do elemento , é possível incluir um número de **1** a **7** para alterar o tamanho da fonte no atributo **size** (tamanho), sendo 1 (um) o menor e 7 (sete) o maior. O valor padrão do atributo size é 3. Na figura a seguir, ilustramos o exemplo de aplicação dessa propriedade no texto "Desenvolvimento de Software para Internet", que se encontra entre as tags e .



```

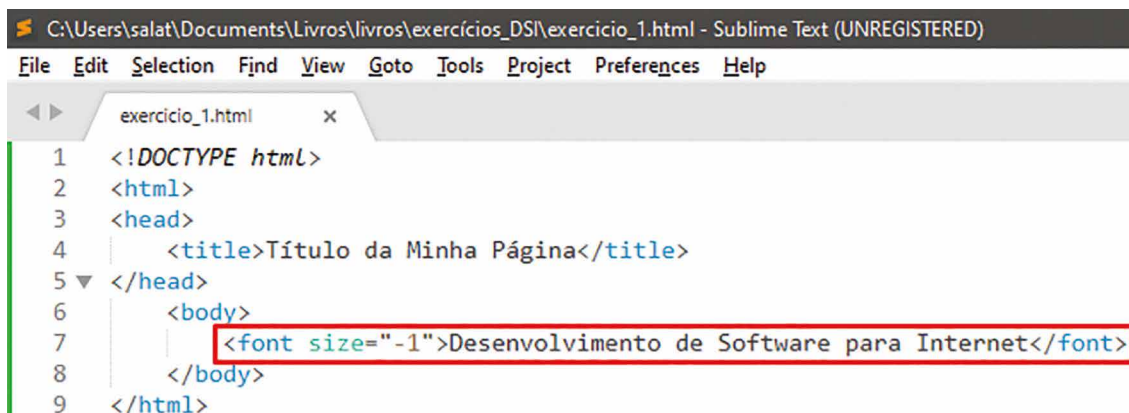
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Título da Minha Página</title>
5  </head>
6      <body>
7          <font size="1">Desenvolvimento de Software para Internet</font>
8      </body>
9  </html>
    
```

Figura 35 – Exemplo de aplicação das tags e

2.8.10 Alteração do tamanho da fonte:

Para uma alteração imediata do tamanho da fonte em relação ao tamanho da fonte padrão do documento, temos um tipo de valor de atributo que aumenta ou diminui o tamanho da fonte pelo número especificado na marcação, como ilustrado na figura a seguir, com o texto "Desenvolvimento

de Software para Internet". Nesse caso, o texto será exibido em tamanho uma unidade menor do que a fonte padrão. Se o padrão é 3, por exemplo, ele será exibido com tamanho 2.

The screenshot shows the Sublime Text editor with a file named 'exercicio_1.html'. The code is as follows:

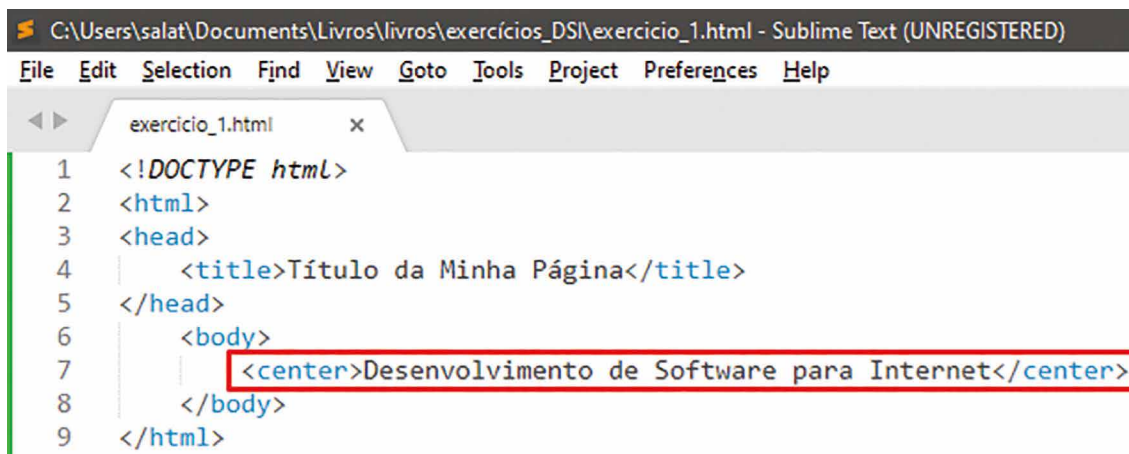
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <font size="-1">Desenvolvimento de Software para Internet</font>
8 </body>
9 </html>
```

The line containing the font tag is highlighted with a red rectangle.

Figura 36 – Exemplo de aplicação das tags e

2.8.11 Centro: <center> </center>

É um elemento muito útil, que possibilita que tudo que esteja entre as marcas fique centralizado. Na figura a seguir, o texto "Desenvolvimento de Software para Internet" será apresentado no centro da página.

The screenshot shows the Sublime Text editor with the same file. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <center>Desenvolvimento de Software para Internet</center>
8 </body>
9 </html>
```

The line containing the center tag is highlighted with a red rectangle.

Figura 37 – Exemplo de aplicação das tags <center> e </center>

2.8.12 Ênfase:

Este elemento é usado para enfatizar o texto, que geralmente aparece em itálico, mas isso pode variar de acordo com o browser. Na figura a seguir, o texto "Desenvolvimento de Software para Internet" será apresentado em formato **itálico** no browser.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <em>Desenvolvimento de Software para Internet</em>
8 </body>
9 </html>

```

Figura 38 – Exemplo de aplicação das tags e

2.8.13 Ênfase forte:

Esse elemento é utilizado para enfatizar o texto, que geralmente aparece em negrito, mas pode variar de acordo com o browser. Na figura a seguir, o texto "Desenvolvimento de Software para Internet" será apresentado em formato **negrito** no browser.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Título da Minha Página</title>
5 </head>
6 <body>
7   <strong>Desenvolvimento de Software para Internet</strong>
8 </body>
9 </html>

```

Figura 39 – Exemplo de aplicação das tags e



Lembrete

Você pode criar uma pasta chamada de "Desenvolvimento de Software Para Internet" e nela adicionar todos os exercícios. Para facilitar a localização, crie a pasta na área de trabalho (desktop).

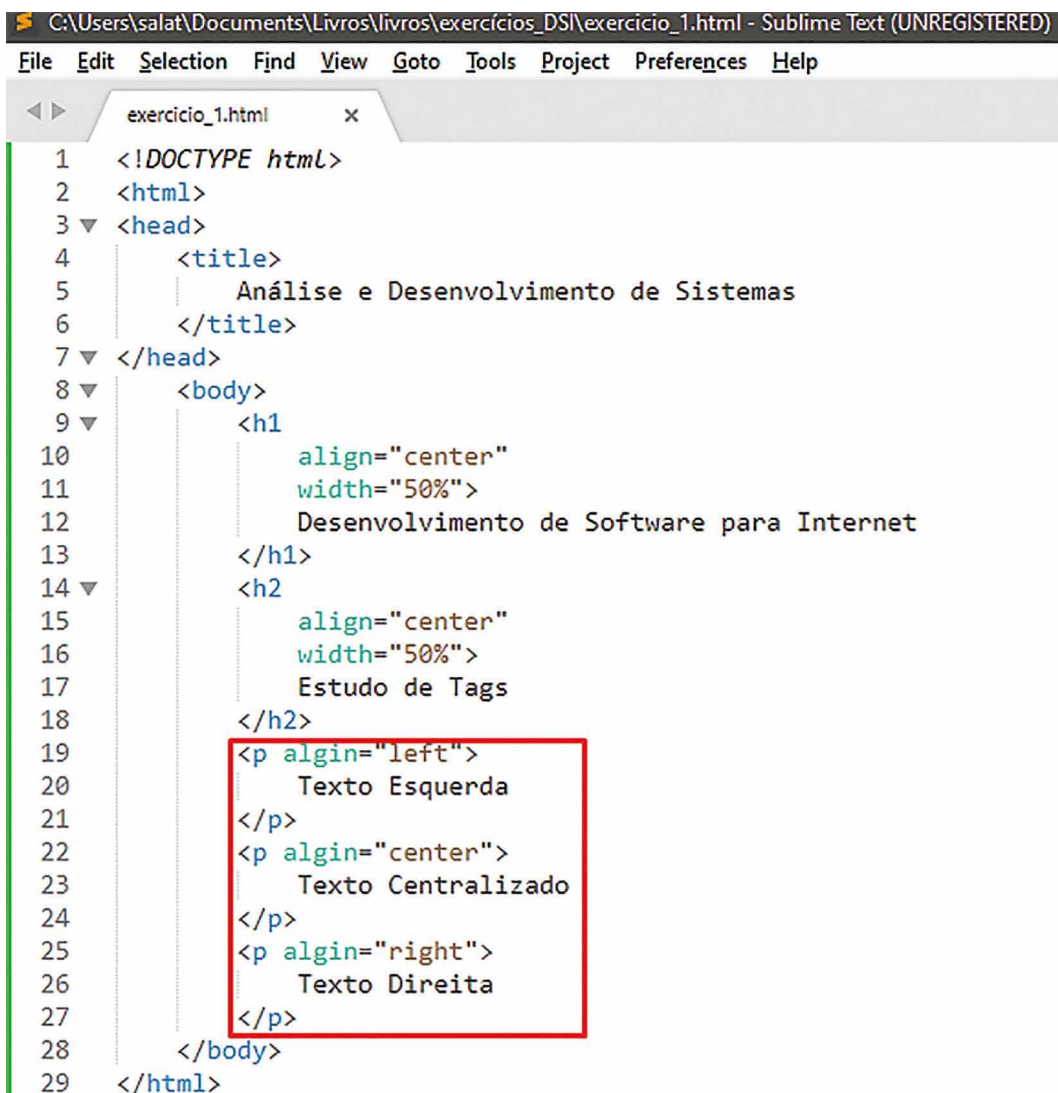
2.8.14 Parágrafos: <p> </p>

É possível inserir textos diretamente no corpo do documento HTML (ou seja, diretamente entre as marcas <body> e </body>), mas é convencional que a maior parte do conteúdo de uma página da web simples apareça em parágrafos ou seções.

Para definirmos parágrafos, utilizaremos as tags `<p>` e `</p>`, e os navegadores colocarão o texto automaticamente na próxima linha, assim que atingir a borda da tela. Quando uma tag `<p>` for utilizada, indicará o próximo parágrafo, e o navegador adicionará um espaço em branco entre os parágrafos. O elemento `<p>` trabalha com atributos, entre os quais destacamos os que seguem.

- **`align=""`**: alinhamento de texto no parágrafo. Assume os valores `left`, `center` ou `right` (esquerda, centralizado ou direita).
- **`style="width:x%"`**: o parágrafo ocupará uma porcentagem fixa da página, que deve substituir o valor do `x`. Por exemplo, `<p style="width:10%">` fará com que a largura do parágrafo ocupe 10% da largura total da janela do navegador. O valor padrão é 100%. Nesse caso, `width` é uma propriedade CSS e `x%` é um valor CSS. Essa linguagem de estilos será abordada com mais profundidade adiante.

Na figura a seguir, vemos elementos `<p>` sendo aplicados com o atributo **`align`**.

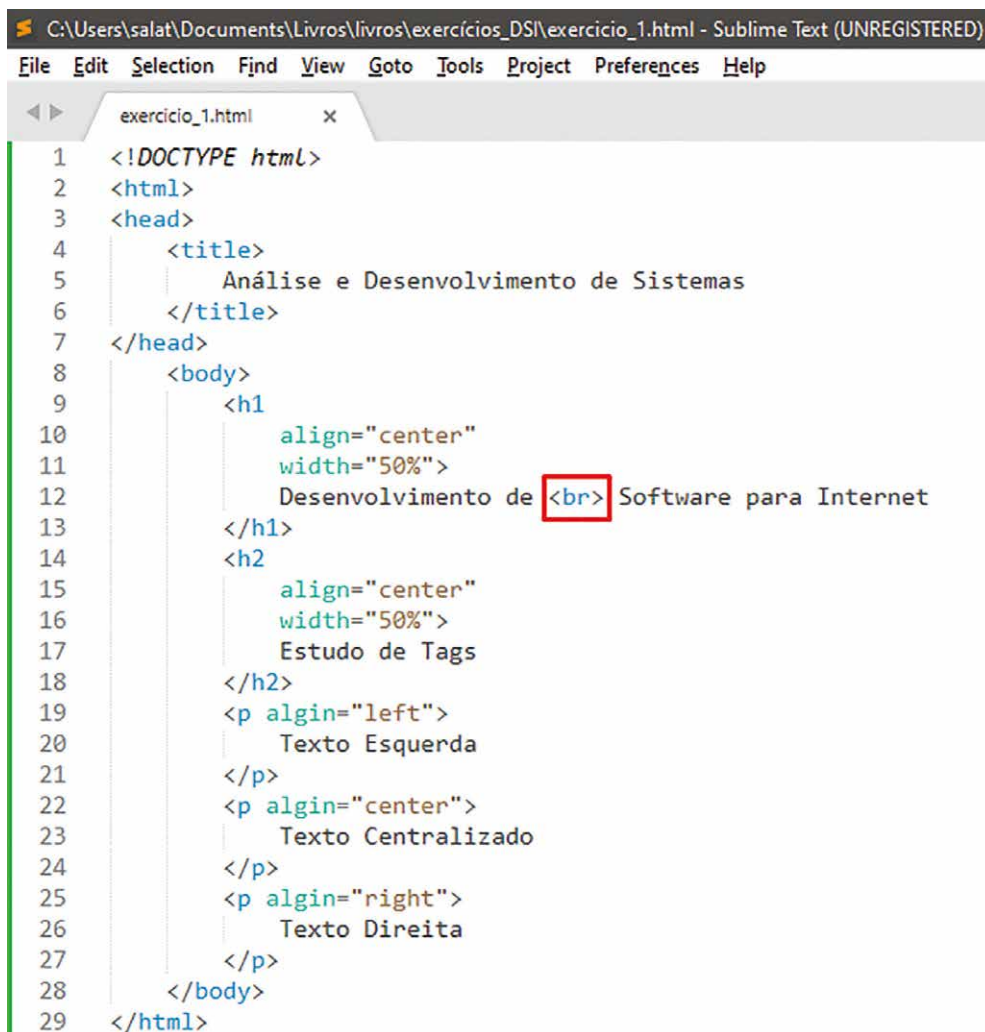


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>
5     Análise e Desenvolvimento de Sistemas
6   </title>
7 </head>
8 <body>
9   <h1>
10     align="center"
11     width="50%">
12     Desenvolvimento de Software para Internet
13   </h1>
14   <h2>
15     align="center"
16     width="50%">
17     Estudo de Tags
18   </h2>
19   <p align="left">
20     Texto Esquerda
21   </p>
22   <p align="center">
23     Texto Centralizado
24   </p>
25   <p align="right">
26     Texto Direita
27   </p>
28 </body>
29 </html>
```

Figura 40 – Exemplo de aplicação das tags `<p>` e `</p>`

2.8.15 Quebra de linha:

Esta tag é utilizada para iniciar o texto em uma nova linha. Várias tags
 em uma linha criarão um grande espaço vertical em uma página da web. Um exemplo de utilização dessa tag é ilustrado na figura a seguir.

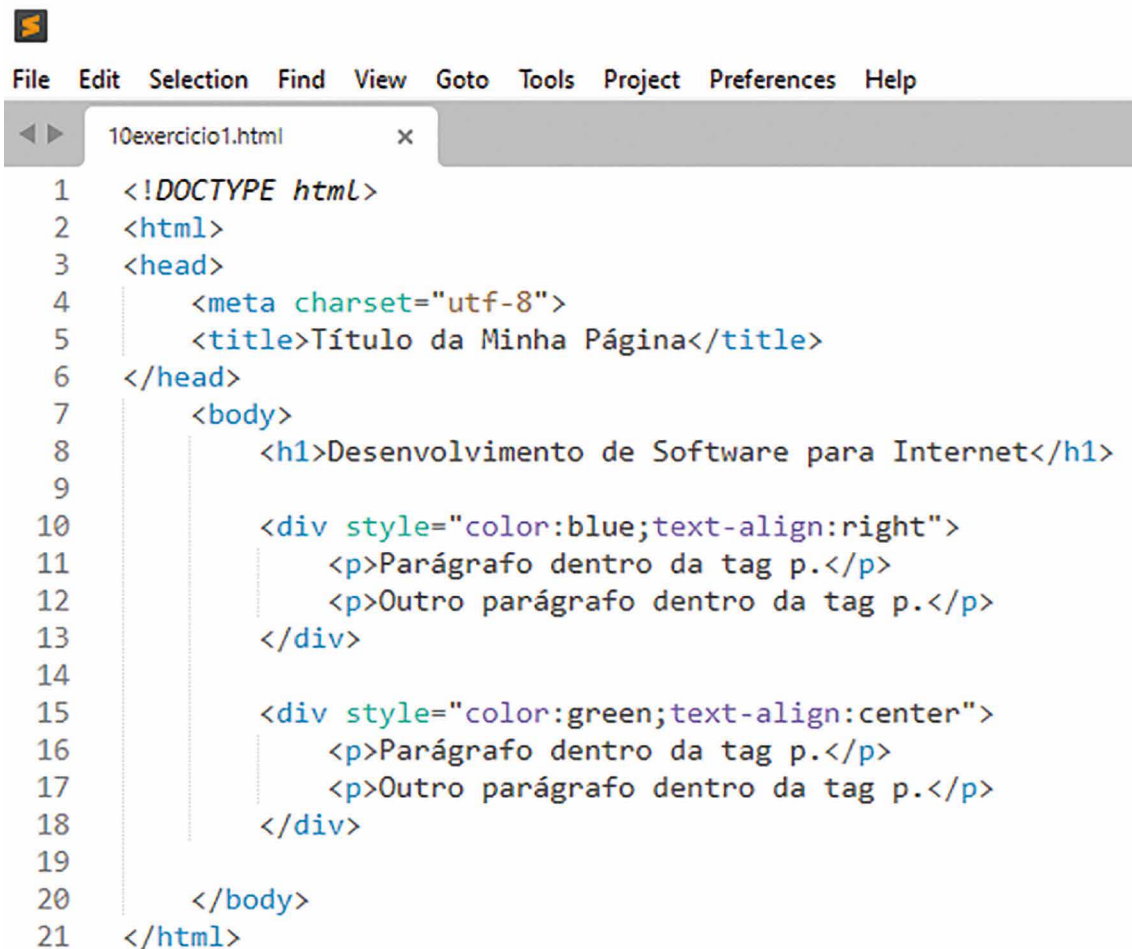


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>
5     Análise e Desenvolvimento de Sistemas
6   </title>
7 </head>
8 <body>
9   <h1
10     align="center"
11     width="50%">
12     Desenvolvimento de <br> Software para Internet
13   </h1>
14   <h2
15     align="center"
16     width="50%">
17     Estudo de Tags
18   </h2>
19   <p align="left">
20     Texto Esquerda
21   </p>
22   <p align="center">
23     Texto Centralizado
24   </p>
25   <p align="right">
26     Texto Direita
27   </p>
28 </body>
29 </html>
```

Figura 41 – Exemplo de aplicação da tag

2.8.16 Divisão: <div> </div>

O elemento <div> define uma seção ou divisão dentro de um arquivo HTML. Geralmente, contém títulos, parágrafos, tabelas ou outros elementos que precisam ser agrupados. Também podemos agregar CSS, definindo características na tag de abertura <div style="">. O atributo **style** pode ser utilizado para definir a aparência de uma seção de uma página web, como cor e alinhamento do texto. Dessa forma, toda a seção, que deve estar entre as marcas <div> e </div>, assume a aparência ajustada pelo atributo, conforme demonstrado nas duas figuras a seguir. Você pode testar outros valores para os atributos demonstrados, sempre verificando o resultado em seu navegador.



```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Título da Minha Página</title>
6  </head>
7      <body>
8          <h1>Desenvolvimento de Software para Internet</h1>
9
10         <div style="color:blue;text-align:right">
11             <p>Parágrafo dentro da tag p.</p>
12             <p>Outro parágrafo dentro da tag p.</p>
13         </div>
14
15         <div style="color:green;text-align:center">
16             <p>Parágrafo dentro da tag p.</p>
17             <p>Outro parágrafo dentro da tag p.</p>
18         </div>
19
20     </body>
21 </html>
    
```

Figura 42 – Exemplo de aplicação da tags <div> e </div>

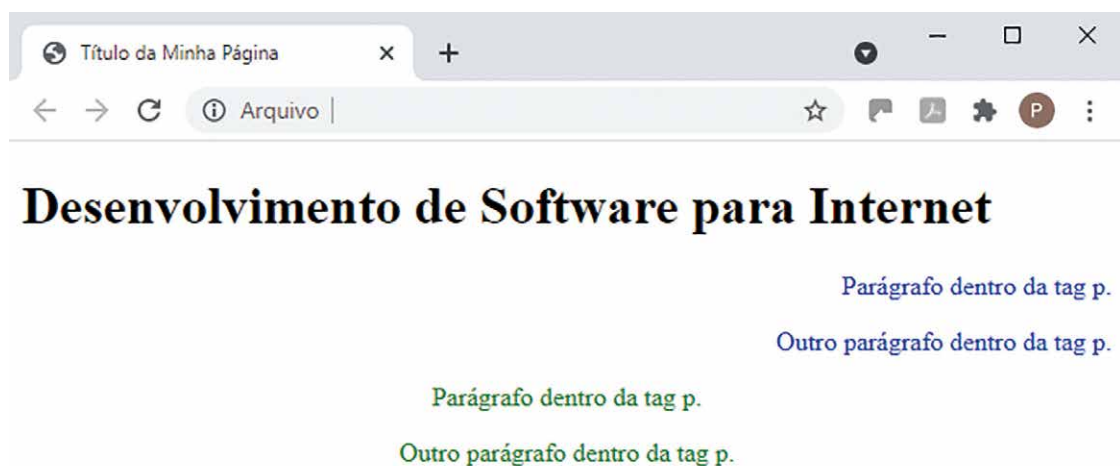


Figura 43 – Resultado, no navegador, da aplicação do elemento <div> no código

Agregando CSS ao nosso código, podemos definir características na tag `<div class="">`. O atributo **class** também pode ser utilizado para definir a aparência de uma seção, mas, nesse caso, precisamos criar uma classe que define esses atributos dentro do elemento `<style>`.

O elemento `<style>` é especificamente utilizado para definir informações de estilo CSS em um documento. Quando criamos uma classe, definimos um estilo que pode ser aplicado a diferentes partes do documento HTML, se necessário. Esse tema será abordado na unidade III do nosso livro-texto, quando trataremos mais especificamente da linguagem CSS.

2.8.17 Tabela: `<table>` `</table>`

O elemento `<table>` é utilizado para definir uma tabela na sua página HTML. As linhas, os títulos das colunas e o conteúdo de cada célula são definidos por meio de tags específicas, listadas a seguir.

- **Table row** `<tr>` `</tr>`: utilizadas para iniciar uma linha da tabela, as marcas `<tr>` e `</tr>` devem aparecer dentro das marcas `<table>` e `</table>`. Deve haver pelo menos uma linha em uma tabela.
- **Table cell** `<td>` `</td>`: uma célula da tabela é o elemento onde o conteúdo será apresentado. As células devem existir dentro de linhas, e o número de células em uma linha determina o número de colunas na tabela.
- **Header cell** `<th>` `</th>`: semelhante a uma célula da tabela, uma célula do cabeçalho deve aparecer dentro de uma linha da tabela. Normalmente encontradas na primeira linha, as células do cabeçalho geralmente são mostradas em negrito e centralizadas pelo navegador.

No exemplo da figura a seguir, criamos uma tabela com os integrantes de uma família. Foram criadas quatro linhas (note que há quatro elementos `<tr>`) e três títulos de colunas (note que há três elementos `<th>` dentro da primeira linha criada). Com os elementos `<td>`, posicionamos os conteúdos das células da nossa tabela, linha a linha. Por meio do atributo `style`, indicamos que a tabela deve ocupar 30% da janela do navegador.



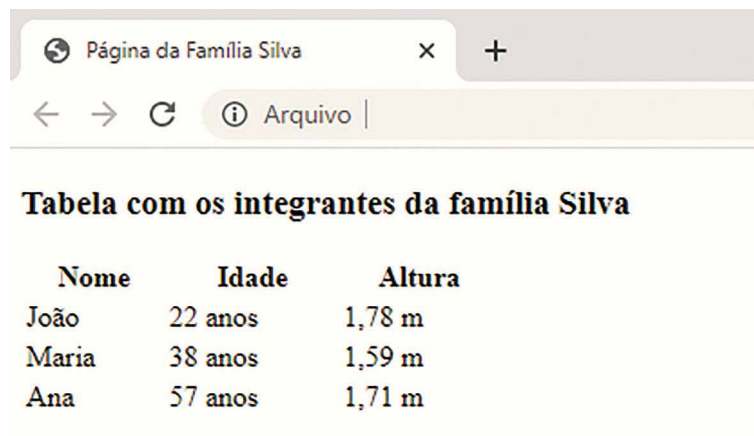
The screenshot shows a web browser window with a single tab titled '11exercicio1.html'. The browser's address bar is empty. The page content is an HTML document with the following structure:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Página da Família Silva</title>
6 </head>
7 <body>
8   <h3>Tabela com os integrantes da família Silva</h3>
9   <table style="width:30%">
10     <tr>
11       <th>Nome</th>
12       <th>Idade</th>
13       <th>Altura</th>
14     </tr>
15     <tr>
16       <td>João</td>
17       <td>22 anos</td>
18       <td>1,78 m</td>
19     </tr>
20     <tr>
21       <td>Maria</td>
22       <td>38 anos</td>
23       <td>1,59 m</td>
24     </tr>
25     <tr>
26       <td>Ana</td>
27       <td>57 anos</td>
28       <td>1,71 m</td>
29     </tr>
30   </table>
31 </body>
```

The table displayed on the page has the following data:

Nome	Idade	Altura
João	22 anos	1,78 m
Maria	38 anos	1,59 m
Ana	57 anos	1,71 m

Figura 44 – Exemplo de aplicação das tags <table> e </table>



The screenshot shows a web browser window with the title 'Página da Família Silva'. The address bar contains 'Arquivo |'. Below the browser window, a table titled 'Tabela com os integrantes da família Silva' is displayed. The table has three columns: 'Nome', 'Idade', and 'Altura'. The rows list three family members: João (22 anos, 1,78 m), Maria (38 anos, 1,59 m), and Ana (57 anos, 1,71 m).

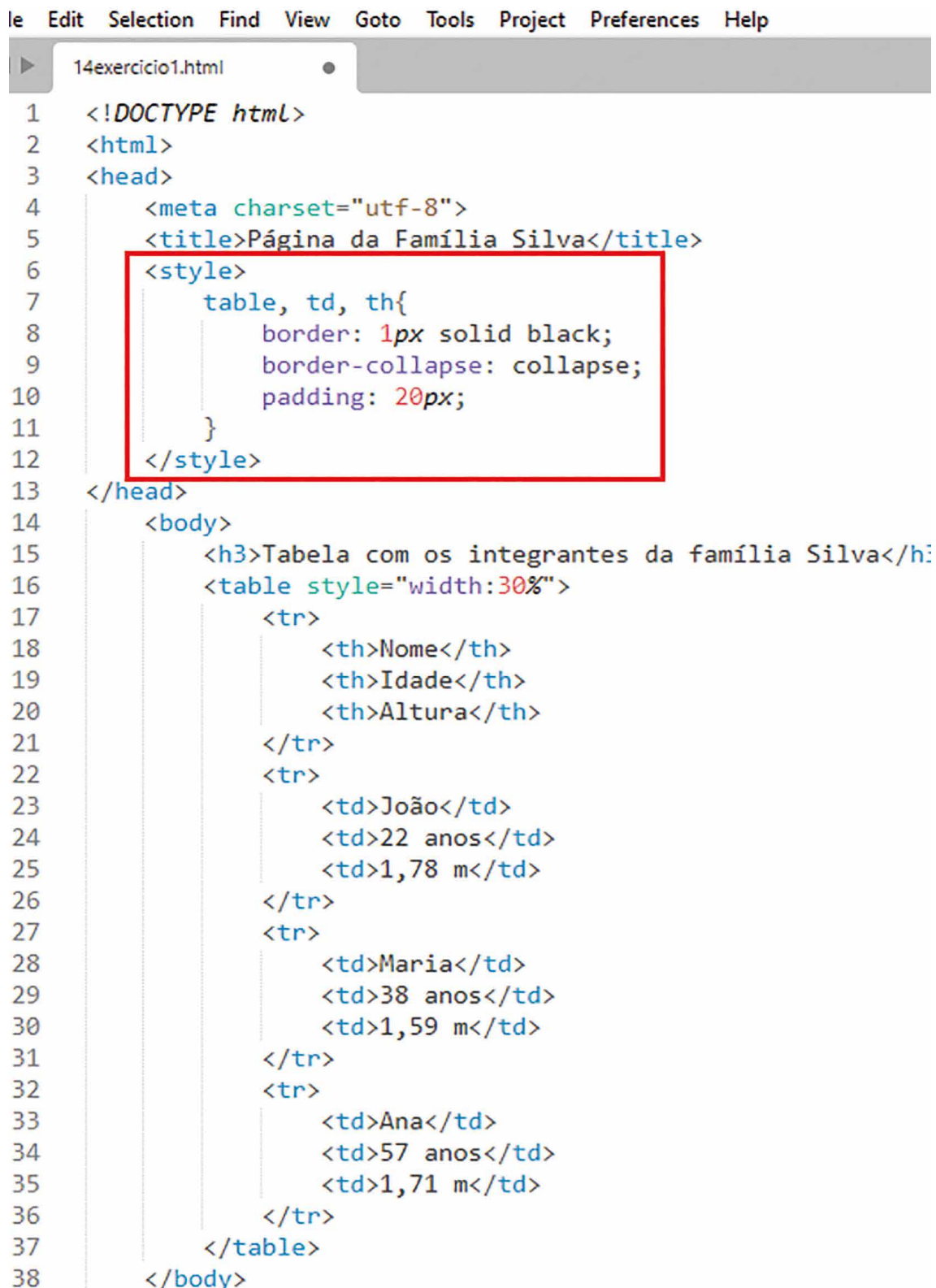
Nome	Idade	Altura
João	22 anos	1,78 m
Maria	38 anos	1,59 m
Ana	57 anos	1,71 m

Figura 45 – Resultado, no navegador, da aplicação das tags `<table>` e `</table>` no código

É possível também aplicar bordas à tabela. Podemos fazer isso por meio da propriedade `border` do CSS. Vamos ajustar as bordas da nossa tabela do exemplo anterior, de acordo com a figura adiante. Para isso, devemos incluir algumas informações de formatação dentro do elemento `<style>` do nosso cabeçalho, conforme código mostrado na figura seguinte. Definimos uma borda com espessura de 1 px na cor preta, aplicada tanto à parte externa da tabela quanto às suas células.

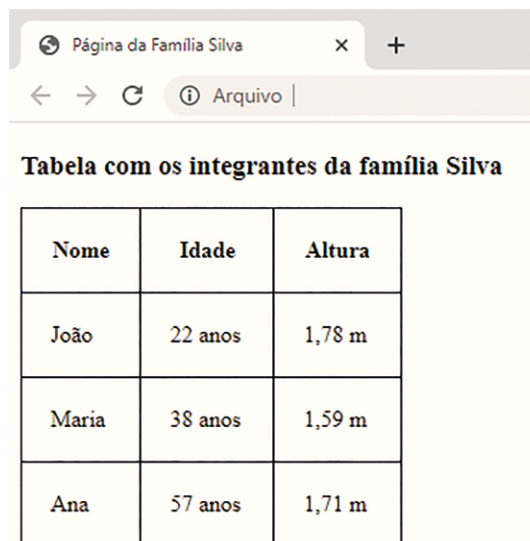
A linha `border-collapse: collapse;` faz com que não haja separação entre a borda externa da tabela e as bordas das células. Dessa forma, elas entram em colapso, tornando-se apenas uma linha. Experimente tirar essa linha do código e veja o que acontece.

Já a linha `padding 20 px` faz com que haja uma distância de pelo menos 20 px entre as bordas e o conteúdo de nossas células. De forma geral, a propriedade `padding` define uma distância entre o conteúdo de um elemento e suas bordas, podendo ser aplicada a outros elementos, além das tabelas.



```
le Edit Selection Find View Goto Tools Project Preferences Help
14exercicio1.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Página da Família Silva</title>
6      <style>
7          table, td, th{
8              border: 1px solid black;
9              border-collapse: collapse;
10             padding: 20px;
11         }
12     </style>
13 </head>
14 <body>
15     <h3>Tabela com os integrantes da família Silva</h3>
16     <table style="width:30%">
17         <tr>
18             <th>Nome</th>
19             <th>Idade</th>
20             <th>Altura</th>
21         </tr>
22         <tr>
23             <td>João</td>
24             <td>22 anos</td>
25             <td>1,78 m</td>
26         </tr>
27         <tr>
28             <td>Maria</td>
29             <td>38 anos</td>
30             <td>1,59 m</td>
31         </tr>
32         <tr>
33             <td>Ana</td>
34             <td>57 anos</td>
35             <td>1,71 m</td>
36         </tr>
37     </table>
38 </body>
```

Figura 46 – Exemplo de aplicação do elemento <table> com estilo definido no elemento <style>



A screenshot of a web browser window. The address bar shows 'Página da Família Silva' and 'Arquivo'. Below the browser window, a table is displayed with the title 'Tabela com os integrantes da família Silva'. The table has three columns: 'Nome', 'Idade', and 'Altura'. It contains three rows of data: João (22 anos, 1,78 m), Maria (38 anos, 1,59 m), and Ana (57 anos, 1,71 m).

Nome	Idade	Altura
João	22 anos	1,78 m
Maria	38 anos	1,59 m
Ana	57 anos	1,71 m

Figura 47 – Resultado, no navegador, da tabela estilizada

O elemento `<table>` trabalha com atributos que podem ser utilizados para definir o seu layout, tais como:

- **Border=""**: define o tamanho da borda (em pixels) ao redor da tabela (utilizado em nosso exemplo).
- **Cellspacing=""**: define o espaço (em pixels) entre cada célula (por exemplo, entre linhas ou colunas).
- **Cellpadding=""**: define o espaço ou a margem entre o conteúdo de uma célula e sua borda.



Observação

O pixel (px), em CSS, é uma unidade de comprimento que corresponde, aproximadamente, à dimensão de um ponto que pode ser visto pelo olho humano, mas é o menor possível.

2.8.18 Table cell: `<td>` `</td>`

Conforme mencionado no tópico anterior, é nas células da tabela que posicionamos os dados por ela apresentados. As células devem existir dentro de linhas, sendo que o número de células em uma linha determina o número de colunas na tabela.

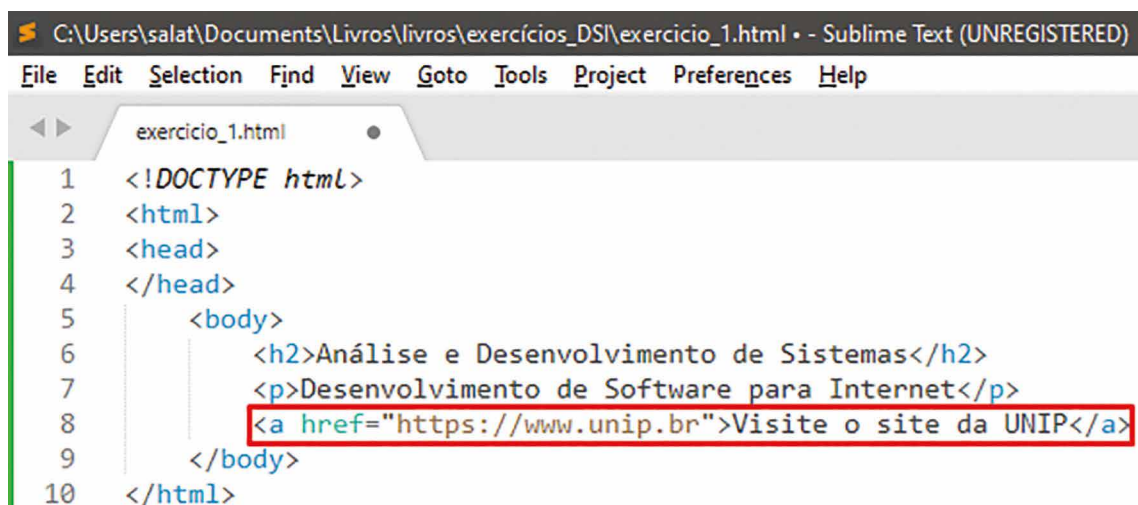
As propriedades de uma célula podem ser configuradas usando os atributos a seguir.

- **Align=""**: refere-se ao alinhamento do texto na célula – left, center ou right (esquerda, centralizado ou direita).

- **Valign=""**: refere-se ao alinhamento vertical da célula – top, middle ou bottom.
- **Width=""**: especifica uma largura para uma célula; por padrão, eles ocuparão apenas o espaço necessário.
- **Colspan=""**: refere-se à extensão da coluna que permite que uma célula ocupe mais de uma coluna, para corresponder aos layouts de outras linhas.
- **Rowspan=""**: refere-se à extensão da linha, semelhante à extensão da coluna; força uma célula a ocupar mais de uma linha.
- **Nowrap**: nenhum texto na célula será quebrado na próxima linha, semelhante à <nobr>, tag para parágrafos.

2.9 Ligações

O elemento <a>, conhecido como âncora, serve para definir hyperlinks no documento HTML. Os links são necessários para que os usuários "pulem" de uma página para outra. Na estrutura da página, as marcas desse elemento são sinalizadas como , conforme ilustrado a seguir, pois o atributo **href** é o mais importante desse elemento. Ao abrir o arquivo HTML no navegador, você deve observar um link que, ao ser clicado, irá direcioná-lo para a página da UNIP. Faça o teste.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <h2>Análise e Desenvolvimento de Sistemas</h2>
7 <p>Desenvolvimento de Software para Internet</p>
8 <a href="https://www.unip.br">Visite o site da UNIP</a>
9 </body>
10 </html>
```

Figura 48 – Exemplo de aplicação das tags <a href> e

2.9.1 Link básico: texto do link

Conforme pudemos observar, existem duas partes principais em um elemento âncora: o texto em que um usuário pode clicar e o endereço da web para o qual ele é direcionado, se nele clicar. Os caracteres posicionados entre as tags <a> e representam o texto do link, geralmente exibido em azul e sublinhado. O atributo href="url" é o endereço da web, em que o URL pode ser definido de várias maneiras, conforme ilustrado na figura.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <h2>Análise e Desenvolvimento de Sistemas</h2>
7 <p>Desenvolvimento de Software para Internet</p>
8 <a href="unidade1.html">acessar outra página no diretório atual</a>
9 <a href="ads/unidade1.html">definir localização relativa</a>
10 <a href="https://www.ads.br/uniade1.html">definir um endereço completo</a>
11 </body>
12 </html>

```

Figura 49 – Exemplo de variação da tag <a href>

2.9.2 Link para um fragmento: link

Muitas vezes, é útil criar um link para outros lugares na mesma página da web, como outras seções ou capítulos mais abaixo na página. O termo técnico para isso é vinculado a um fragmento, em que os browsers tentam automaticamente rolar para a parte da página.

2.9.3 Janela de destino: link

Nem sempre desejamos vincular e apresentar sobre a página que será visualizada no momento. O atributo **target** é útil para auxiliar na disponibilização de outra página quando requisitada e é apresentado em outra aba do browser. Para realizar essa implementação, utilizamos a `target="_blank"`, conforme figura a seguir.

```

C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <p>
7   Imagem como link:
8   <a href="https://www.unip.br">
9     
15   </a>
16 </p>
17 </body>
18 </html>

```

Figura 52 – Exemplo de imagem como link



Figura 53 – Exemplo de imagem como link

A imagem indicada na figura (`src="unip.jpeg"`) precisa estar na mesma pasta onde estão sendo aplicados os exemplos apresentados neste livro-texto; `src` é uma abreviação de *source* (fonte), que indica em qual diretório se encontra a imagem.

2.9.5 Link de e-mail: ``

Um tipo especial de link, o `mailto`, instrui o browser a compor e enviar um e-mail com o endereço especificado, utilizando o programa de e-mail padrão (Outlook, por exemplo), conforme apresentado a seguir.


```
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <p>Link para envio de e-mail, ao clicar será aberto seu software de envio de e-mail</p>
7 <p><a href="mailto:aluno@unip.br">Enviar E-mail</a></p>
8 </body>
9 </html>
```

Figura 54 – Exemplo de link e-mail

Você também pode definir um assunto para o e-mail utilizando a tag `Unidade 1`.

2.10 Imagens

Adicionar fotos ou outras imagens à página da web é muito simples: com a tag ``, conforme apresentado na figura a seguir, foram utilizados os atributos para definição de borda (*border*), rótulo da imagem (*alt*), altura e largura da figura (*height* e *width*) e a localização da imagem através do *source* (*src*).

```
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 
13 </body>
14 </html>
```

Figura 55 – Exemplo de aplicação da tag ``

2.10.1 Imagem: ``

Para exibir uma imagem, precisamos especificar seu endereço utilizando o atributo `src`, substituindo `url` pelo nome do arquivo da imagem. Existem várias maneiras de fazer isso (ver figura).

- **Src = "unidade1.jpeg"**: buscará a imagem na raiz do projeto.
- **Src= "images/Picture.jpeg"**: buscará a imagem dentro da pasta Imagens.
- **Src="http://www.ads.com/images/unidade1.jpeg"**: buscará a imagem na URL <http://www.ads.com/images/unidade1.jpeg>.

```
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercico_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercico_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <!-- o nome do arquivo se a imagem estiver no mesmo diretório que o arquivo HTML -->
7 
14 <!-- um caminho relativo quando a imagem está em outro diretório -->
15 
22 <!-- uma URL completa também pode ser utilizada -->
23 
30 </body>
31 </html>
```

Figura 56 – Exemplo de aplicação da tag

2.10.2 Texto alternativo:

O alt é o atributo que define o texto apresentado no lugar de uma imagem quando ela não pode ser carregada. Esse é um atributo necessário para o HTML válido e deve descrever brevemente o que a imagem normalmente faria, conforme apresentado na figura.


```
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6     <!-- o nome do arquivo se a imagem estiver no mesmo diretório que o arquivo HTML -->
7     
14    <!-- um caminho relativo quando a imagem está em outro diretório -->
15    
22    <!-- uma URL completa também pode ser utilizada -->
23    
30 </body>
31 </html>
```

Figura 57 – Exemplo de aplicação da tag

2.10.3 Tamanho da imagem:

Uma imagem normalmente será mostrada no tamanho real, mas, utilizando os atributos **width** (largura) e **height** (altura), podemos alterar o tamanho da imagem a ser exibida. Podemos especificar o tamanho em pixels ou como uma porcentagem.

Uma dica é especificar o tamanho usando o tamanho real da imagem em pixels para forçar os browsers a alocar espaço para a imagem antes que ela seja carregada, garantindo que o layout da página permaneça o mesmo, com ou sem imagens exibidas, conforme apresentado na figura.

```
C:\Users\salat\Documents\Livros\livros\exercicios_DS\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <!-- o nome do arquivo se a imagem estiver no mesmo diretório que o arquivo HTML -->
7 
14 <!-- um caminho relativo quando a imagem está em outro diretório -->
15 
22 <!-- uma URL completa também pode ser utilizada -->
23 
30 </body>
31 </html>
```

Figura 58 – Exemplo de aplicação da tag

2.10.4 Bordas

Trata-se da tag utilizada para especificar a espessura em pixels de uma borda. Podemos definir o atributo border="0" como a remoção da borda adicionada quando as imagens são utilizadas como links, conforme apresentado a seguir.

```
C:\Users\sala\Documents\Livros\livros\exercicios_DS\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6     <!-- o nome do arquivo se a imagem estiver no mesmo diretório que o arquivo HTML -->
7     
14    <!-- um caminho relativo quando a imagem está em outro diretório -->
15    
22    <!-- uma URL completa também pode ser utilizada -->
23    
30 </body>
31 </html>
```

Figura 59 – Exemplo de aplicação da tag

2.10.5 Alinhamento de imagem: –

Por padrão, uma imagem aparece no local especificado no código HTML (como em qualquer outra tag). No entanto, podemos alinhar uma imagem com o texto ou parágrafo ao redor, definindo qualquer um deles `align="left | right | top | bottom | middle"`, conforme apresentado a seguir.

```

C:\Users\salat\Documents\Livros\livros\exercícios_DSI\exercício_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercício_1.html x
1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5  <body>
6      <!-- o nome do arquivo se a imagem estiver no mesmo diretório que o arquivo HTML -->
7      
15     <!-- um caminho relativo quando a imagem está em outro diretório -->
16     
24     <!-- uma URL completa também pode ser utilizada -->
25     
33 </body>
34 </html>

```

Figura 60 – Exemplo de aplicação da tag

2.10.6 Espaçamento:

Atributos **vspace** e **hspace** da tag proporcionam ajuste de espaço em branco (ou espaço de contorno) ao redor de uma imagem, em pixels. Utiliza-se **vspace** para ajustar o espaçamento vertical acima e abaixo, ou **hspace**, para os lados esquerdo e direito, conforme apresentado na figura.

```
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 
16 
26 </body>
27 </html>
```

Figura 61 – Exemplo de aplicação das tags e

O quadro a seguir apresenta um resumo dos principais elementos utilizados para o desenvolvimento de uma página web.

Quadro 7 – Elementos HTML essenciais para construção de uma página web

Elemento	Exemplo	Resultado
Parágrafo	<p> algum texto </p>	Algum texto
Quebra de linha	<p> algum texto quebrado </p>	Algum texto quebrado
Cabeçalho principal	<h1> cabeçalho 1 </h1>	Cabeçalho 1
Subtítulo	<h5> rubrica 5 </h5>	Rubrica 5
Negrito	 texto em negrito 	Texto em negrito
Itálico	<i> texto em Itálico </i>	Texto em itálico
Texto colorido	<p style="color:red;">	Texto colorido vermelho
	Texto colorido vermelho </p>	
Alterar fonte	<p style="font-family:serif;">	Texto em fonte Serif
	Texto em fonte Serif</p>	
Texto citado	<q> texto citado </q>	Texto citado

2.11 Estudo da estrutura de arquivos HTML

Pela figura a seguir, podemos compreender que um documento em HTML se torna simples e objetivo, considerando uma divisão simples de blocos compostos por tags e seguindo a estrutura:

- definição do documento (doctype);
- cabeçalho (head);
- corpo (body).

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <title>Título da página</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8     Aqui aplicaremos o código HTML que possibilitará que o seu site aparecer.
9   </body>
10 </html>

```

Figura 62 – Exemplo de estrutura de página HTML

2.11.1 Doctype: definindo o documento

O doctype, que estamos utilizando desde o início de nossos exemplos, é um item de extrema importância e que sempre deve ser levado em consideração. Em arquivos HTML, ele sempre deve existir, no formato `<!DOCTYPE HTML>`.

O doctype não é uma tag HTML, mas sim uma instrução para que o navegador e outros programas possam então ler seu site, sendo que o código encontrado ali será um código HTML. Essa instrução indica para o navegador qual versão do HTML deve ser utilizada para renderizar a página. Assim, os navegadores conseguem entender o que fazer para apresentar seu site da melhor forma. Lembre-se de que o doctype é **obrigatório**: deve ser sempre a **primeira linha** do seu documento de marcação.

2.11.2 Head

Esse elemento contém informações que, por definição, não são transpostas visivelmente para o usuário/leitor do documento de marcação. Logo, trata-se de dados implícitos, que podem ser considerados de:

- uso e controle do documento;
- vinculação com ênfase em outros arquivos;
- aplicação de lógica de programação de scripts.

Ou seja, na prática, todo e qualquer conteúdo do cabeçalho fica delimitado entre a abertura e o fechamento do elemento *head*.

2.11.3 Body

Trata-se do elemento que envolve a marcação HTML, ou seja, a informação legível para o usuário/leitor do documento de marcação. Todo e qualquer texto que se deseja apresentar, assim como toda e qualquer forma de mídia de saída, tais como imagens, sons, miniaPLICATIVOS embutidos, conteúdo multimídia etc., será incluído entre as tags `<body>` e `</body>`. Além disso, toda e qualquer apresentação de entrada de dados (formulários) também tem como forma de aplicação nessa seção do documento de marcação, ou seja, na prática, o corpo do documento de marcação é delimitado por esse par de tags.

Consideramos, então, que esse é o preceito básico que deve ser considerado para que fique muito claro em que as marcações se aplicam e quais são os resultados desse modelo – por exemplo, se você deseja informar conteúdo textual para saída legível ao usuário da sua aplicação web, essa marcação deverá, obrigatoriamente, estar no bloco do corpo da página, e, por fim, para definir qual o tipo de codificação da página, esta deve estar marcada no cabeçalho do mesmo documento, ou seja, dentro do elemento *body*. Toda a sua estrutura de página terá os elementos semânticos da construção da sua página, que, por finalidade, será alocada onde serão declarados e identificados cabeçalhos, rodapé, conteúdo principal etc. As figuras a seguir ilustram a aplicação da estrutura HTML.


```

C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Minha página de teste</title>
6   </head>
7   <body>
8     
12  </body>
13 </html>

```

Figura 63 – Exemplo de estrutura de página HTML

Na figura, visualizamos a marcação de uma página HTML que atribui uma imagem que será apresentada no browser.

```

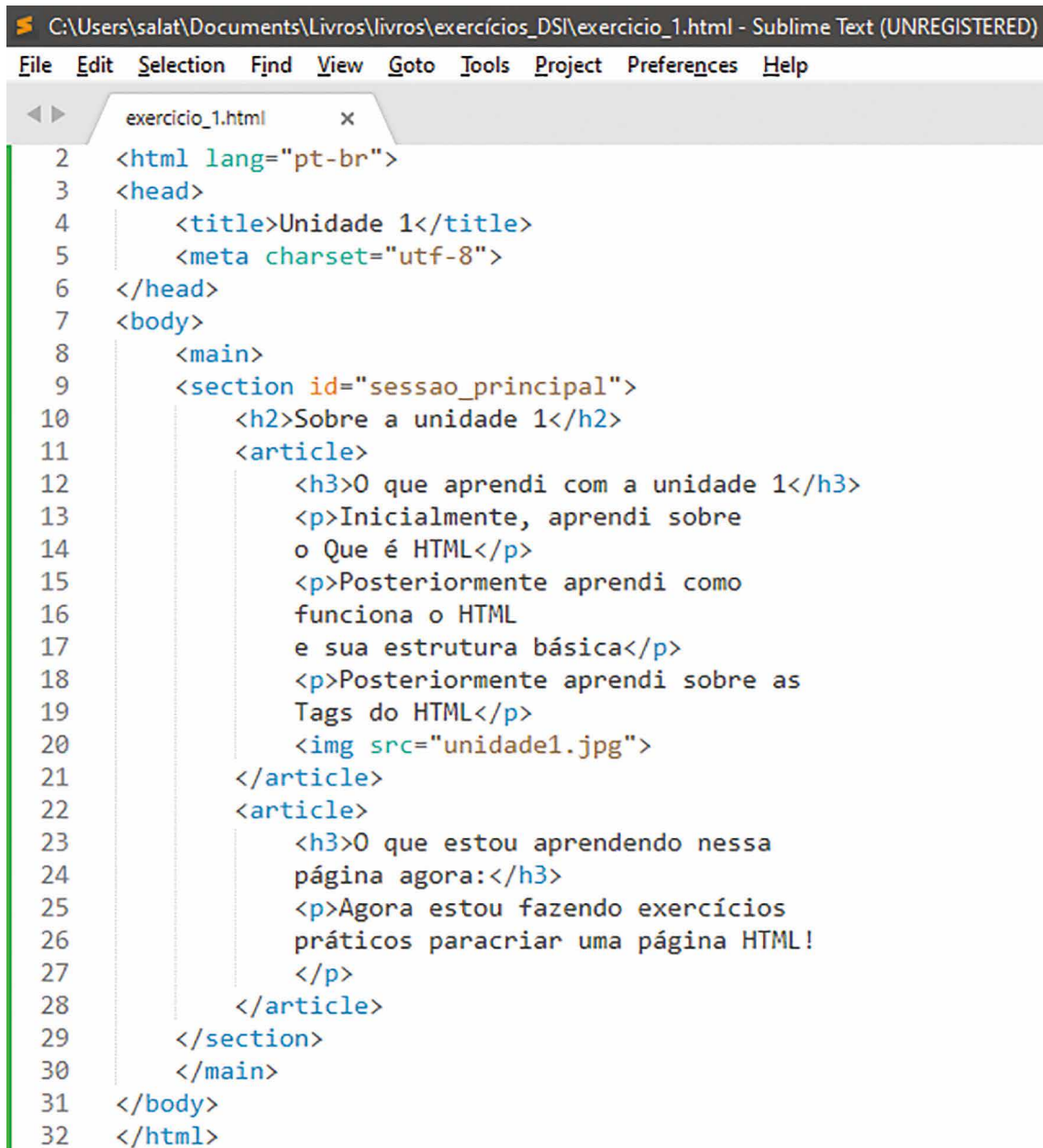
C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercicio_1.html x
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <title>Unidade 1</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8     <header>
9       <h1>Minha primeira página HTML</h1>
10      <p>Desenvolvimento de Software para Internet</p>
11      <br/>
12      <nav>
13        <p>Menu:</p>
14        <ul>
15          <li>item 1</li>
16          <li>item 2</li>
17          <li>item 3</li>
18        </ul>
19      </nav>
20    </header>
21  </body>
22 </html>

```

Figura 64 – Exemplo de estrutura de página HTML

Já na figura anterior, visualizamos a marcação de uma página HTML que disponibiliza, por meio do elemento <nav>, um menu com os itens 1, 2 e 3.



```
1  C:\Users\salat\Documents\Livros\livros\exercicios_DSI\exercicio_1.html - Sublime Text (UNREGISTERED)
2  File Edit Selection Find View Goto Tools Project Preferences Help
3  exercicio_1.html x
4  2 <html lang="pt-br">
5  3 <head>
6  4     <title>Unidade 1</title>
7  5     <meta charset="utf-8">
8  6 </head>
9  7 <body>
10 8     <main>
11 9     <section id="sessao_principal">
1210         <h2>Sobre a unidade 1</h2>
1311         <article>
1412             <h3>O que aprendi com a unidade 1</h3>
1513             <p>Inicialmente, aprendi sobre
1614             o Que é HTML</p>
1715             <p>Posteriormente aprendi como
1816             funciona o HTML
1917             e sua estrutura básica</p>
2018             <p>Posteriormente aprendi sobre as
2119             Tags do HTML</p>
2220             
2321         </article>
2422         <article>
2523             <h3>O que estou aprendendo nessa
2624             página agora:</h3>
2725             <p>Agora estou fazendo exercícios
2826             práticos paracriar uma página HTML!
2927             </p>
3028         </article>
3129     </section>
3230 </main>
3331 </body>
3432 </html>
```

Figura 65 – Exemplo de estrutura de página HTML

Na figura anterior, visualizamos a marcação de uma página HTML que atribui uma imagem que será apresentada no browser, bem como um texto em meio aos elementos <article>, <h3> e <p>.

A seguir, podemos ver com detalhes a estrutura de uma página HTML, com as distribuições das tags.

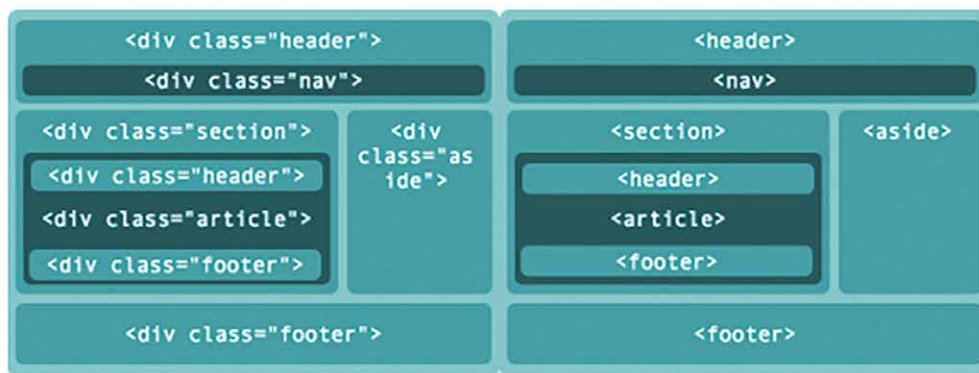


Figura 66 – Estrutura de uma página HTML

Podemos observar toda a parte de marcação que foi desenvolvida e sua apresentação no browser, conforme apresentado nas figuras a seguir.

```

C:\Users\salat\Documents\Livros\livros\exercícios_DS\exercício_1.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

exercício_1.html x
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <title>Unidade 1</title>
5   <meta charset="utf-8">
6 </head>
7 <body>
8   <header>
9     
10    <h1>Estudando a estrutura e semântica do HTML5</h1>
11    <form action="www.unip.br/pesquisa" id="searchform" method="get">
12      <label for="s">Pesquisa de Notas</label>
13      <input type="text" placeholder="Pesquisa" id="s">
14      <input type="submit" value="Pesquisa" class="submit">
15    </form>
16  </header>
17 </body>
18 </html>
  
```

Figura 67 – Estrutura de uma página HTML



Figura 68 – Página exibida no browser



Observação

Para que as páginas apresentadas nesta unidade sejam executadas no browser, deve-se salvar cada uma com a extensão .html e clicar no arquivo. Então, será apresentada a página no seu browser padrão.



Resumo

Nesta unidade, conhecemos, de maneira ampla e objetiva, um pouco da história e alguns conceitos relacionados ao desenvolvimento de softwares para internet.

Mostramos que o desenvolvimento de software para internet não é novo; porém, sua arquitetura descritiva poderia levá-lo a pensar que se trata de algo recente.

Um grande passo para a evolução da construção de sites foi a disseminação da linguagem de marcação HTML e a sua utilização em editores de texto, o que gerou uma série de facilidades, como a amplitude da escolha de editores, a possibilidade de customização de editores e o desenvolvimento de novos editores.

Em relação aos aspectos evolutivos dos sites, pudemos compreender os benefícios dos avanços teóricos e a contribuição na linguagem de marcação HTML com base nos principais pontos que a transformaram em um layout amigável e, principalmente, enriquecedor aos olhos humanos.

Também abordamos alguns conceitos que possibilitam que as tags funcionem de maneira mais eficaz, como cores, parágrafos, imagens, formatação de texto etc.

Por fim, deixamos como nota de estudo para esta unidade conceitos básicos para o desenvolvimento web, como o conhecimento da história da internet, que tem como concepção toda a década de 1970, quando nasceu o TCP/IP (*transmission control protocol/internet protocol*). Na realidade, o TCP/IP refere-se a um conjunto de protocolos que são a base da internet até os dias atuais, o que proporcionou o início e o crescimento da web.

Vale ressaltar que o conceito de internet nasceu de um projeto da agência norte-americana Advanced Research and Projects Agency (ARPA), que teve como objetivo conectar computadores dos seus departamentos de pesquisa. Logo, podemos dizer que a internet é iniciada a partir da Arpanet. Separar alguns instantes para estudar fundamentos históricos da internet é

aprender com a evolução e se preparar para novas oportunidades e tecnologias que irão surgir hoje e sempre.

A evolução da internet fez com que a cada dia houvesse aprimoramentos na linguagem de marcação HTML, para que, assim, fosse gerada uma experiência agradável na página web.



Exercícios

Questão 1. (Copese-UFT 2018, adaptada) Leia o texto a seguir a respeito da linguagem de marcação de hipertexto (HTML).

Não é possível imaginarmos a internet sem pensarmos nos inúmeros sites com os mais diversos recursos multimídia. Hoje, podemos editar um documento, escutar músicas, assistir a vídeos ou editá-los, trocar informações nas mais diversas redes sociais diretamente na internet, apenas utilizando um navegador. Por trás de tudo, desde a homepage mais simples até o portal mais sofisticado, existe algo em comum: a linguagem de marcação HTML. Todos esses sites dependem dessa linguagem para serem construídos e interpretados pelos navegadores. Existem diversos editores que permitem construir sites. No entanto, conhecer a linguagem HTML é extremamente importante, caso precise elaborar um sistema mais sofisticado ou efetuar uma manutenção sem depender necessariamente dessas ferramentas. A versão HTML5 é o resultado da quinta maior revisão da linguagem HTML, considerada o novo padrão para web. Em relação às versões anteriores, a linguagem HTML5 traz inúmeras alterações que possibilitam uma maior interação nos navegadores, principalmente nos dispositivos móveis, como smartphones ou tablets, sem necessitar de recursos externos para, por exemplo, executar um vídeo ou uma música diretamente no navegador. Portanto, estar preparado e atualizado é essencial para quem trabalha na área.

Quanto à estrutura básica dos documentos HTML, a declaração doctype deve ser a primeira instrução do código, aparecendo antes da tag `<html>`. A linguagem HTML5 requer um elemento `<DOCTYPE>` para garantir que a página seja renderizada pelo navegador de maneira correta. A declaração doctype no HTML5 é `<!DOCTYPE html>` e não é case-sensitive. Além disso, em um documento HTML, existem quatro elementos, delimitados por tags de abertura e de fechamento, que são obrigatórios.

Adaptado de: TERUEL, E. C. *HTML5: guia prático*. 2. ed. São Paulo: Érica, 2014.

Acerca da estrutura de um documento HTML, avalie as afirmativas.

I – O HTML é composto de diversas tags, sendo que cada uma apresenta uma funcionalidade específica.

II – O doctype não é uma tag HTML, mas uma instrução. Ela indica para o navegador qual versão do HTML deve ser utilizada para renderizar a página.

III – Um documento HTML válido precisa conter, obrigatoriamente, os elementos `<html>`, `<head>`, `<title>` e `<body>`, além da instrução `doctype`.

IV – O elemento `<head>` não contém informações sobre o documento que são de interesse do navegador, registrando apenas informações dos visitantes do site. São informações exibidas de forma simplificada na área do documento no navegador.

É correto o que se afirma em:

A) I, apenas.

B) II e III apenas.

C) III e IV, apenas.

D) I, II e III, apenas.

E) I, II, III e IV.

Resposta correta: alternativa D.

Análise das afirmativas

I – Afirmativa correta.

Justificativa: as tags são estruturas da linguagem de marcação que abrigam instruções utilizadas pelo navegador para renderizar uma página web. Dessa forma, as tags informam ao navegador a estrutura da página. Elas delimitam os elementos HTML, que costumam abrir com uma tag do tipo `<>` e fechar com uma tag `</>`. As tags têm funcionalidades específicas. Por exemplo, as tags `<title>` e `</title>` servem para abrigar o título do documento HTML.

II – Afirmativa correta.

Justificativa: o `<DOCTYPE>` não é uma tag HTML, mas sim uma instrução para que o navegador saiba a versão da linguagem de marcação que está sendo utilizada. Dessa forma, os navegadores conseguem entender o que fazer para apresentar a página web da melhor forma.

III – Afirmativa correta.

Justificativa: os elementos `<html>`, `<title>`, `<head>` e `<body>` são mandatórios em um documento HTML válido. Além disso, a declaração `doctype` deve ser a primeira instrução do código, aparecendo antes da tag `<html>`.

IV – Afirmativa incorreta.

Justificativa: as tags `<head>` e `</head>` envolvem o cabeçalho de um documento HTML. Ali, são posicionadas informações que não são transpostas visivelmente para o usuário, denominadas metadados (dados a respeito dos próprios dados), como vinculação com outros arquivos, aplicação de lógica de programação de scripts, padrões de codificação de caracteres, entre outras. Esses metadados são informações sobre o documento, que são de interesse do navegador.

Questão 2. A linguagem básica do desenvolvimento web é o HTML. A respeito dos elementos dessa linguagem de marcação, avalie as afirmativas.

I – O elemento `` amplia o tamanho da fonte de um texto.

II – O elemento `<i>` é utilizado para apresentar um texto em itálico.

III – O elemento `<u>` serve para sublinhar um texto.

IV – O elemento `<a>` serve para que imagens sejam inseridas no documento.

É correto o que se afirma em:

A) I, apenas.

B) II e III apenas.

C) I, II e III, apenas.

D) III e IV, apenas.

E) I, II, III e IV.

Resposta correta: alternativa B.

Análise das afirmativas

I – Afirmativa incorreta.

Justificativa: o elemento `` apresenta um texto em negrito no navegador. Para isso, devemos envolver o texto entre as tags `` e ``. Por exemplo, o trecho `Olá!`, no código HTML, deve resultar na exibição "Olá!" no navegador. Obviamente, o resultado real da exibição depende de outros estilos definidos no documento, como tamanho de fonte, cores de texto etc.

III – Afirmativa correta.

IV – Afirmativa incorreta.

```
<a href="url"><img ... ></a>
```

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