



PROGRAMAÇÃO FRONT END



Professor Esp. Ronie Cesar Tokumoto

UNICESUMAR

Av. Guedner, 1610 - Jardim Aclimação
Cep 87050-900 - MARINGÁ - PARANÁ
unicesumar.edu.br
44 3027.6360

UNICESUMAR EDUCAÇÃO A DISTÂNCIA

NEAD - Núcleo de Educação a Distância
Bloco 4 - MARINGÁ - PARANÁ
unicesumar.edu.br
0800 600 6360

as imagens utilizadas neste
livro foram obtidas a partir
do site SHUTTERSTOCK.COM

FICHA CATALOGRÁFICA

C397 **CENTRO UNIVERSITÁRIO DE MARINGÁ.** Núcleo de Educação a Distância; **TOKUMOTO**, Ronie Cesar.

Programação Front End. Ronie Cesar Tokumoto.

Reimpressão, 2021.

Maringá-Pr.: UniCesumar, 2018.

168 p.

"Graduação - EaD".

1. Programação. 2. Front End. 3. EaD. I. Título.

ISBN 978-85-459-0252-2

CDD - 22 ed. 004.678
CIP - NBR 12899 - AACR/2

Ficha catalográfica elaborada pelo bibliotecário
João Vivaldo de Souza - CRB-8 - 6828

Impresso por:



Reitor

Wilson de Matos Silva

Vice-Reitor

Wilson de Matos Silva Filho

Pró-Reitor de Administração

Wilson de Matos Silva Filho

Pró-Reitor de EAD

William Victor Kendrick de Matos Silva

Presidente da Mantenedora

Cláudio Ferdinandi

NEAD - Núcleo de Educação a Distância

Direção Operacional de Ensino

Kátia Coelho

Direção de Planejamento de Ensino

Fabício Lazilha

Direção de Operações

Chrystiano Mincoff

Direção de Mercado

Hilton Pereira

Direção de Polos Próprios

James Prestes

Direção de Desenvolvimento

Dayane Almeida

Direção de Relacionamento

Alessandra Baron

Head de Produção de Conteúdos

Rodolfo Encinas de Encarnação Pinelli

Gerência de Produção de Conteúdos

Gabriel Araújo

Supervisão do Núcleo de Produção de Materiais

Nádila de Almeida Toledo

Coordenador de conteúdo

Danillo Xavier Saes

Qualidade Editorial e Textual

Daniel F. Hey, Hellyery Agda

Design Educacional

Yasmin Zaganel

Iconografia

Amanda Peçanha dos Santos

Ana Carolina Martins Prado

Projeto Gráfico

Jaime de Marchi Junior

José Jhonny Coelho

Arte Capa

Arthur Cantareli Silva

Editoração

Bruna Stefane Martins Marconato

Revisão Textual

Nayara Valenciano

Maraia da Silva

Viviane Notari

Gabriel Martins

Yara Dias



Professor
Wilson de Matos Silva
Reitor

Viver e trabalhar em uma sociedade global é um grande desafio para todos os cidadãos. A busca por tecnologia, informação, conhecimento de qualidade, novas habilidades para liderança e solução de problemas com eficiência tornou-se uma questão de sobrevivência no mundo do trabalho.

Cada um de nós tem uma grande responsabilidade: as escolhas que fizermos por nós e pelos nossos farão grande diferença no futuro.

Com essa visão, o Centro Universitário Cesumar assume o compromisso de democratizar o conhecimento por meio de alta tecnologia e contribuir para o futuro dos brasileiros.

No cumprimento de sua missão – “promover a educação de qualidade nas diferentes áreas do conhecimento, formando profissionais cidadãos que contribuam para o desenvolvimento de uma sociedade justa e solidária” –, o Centro Universitário Cesumar busca a integração do ensino-pesquisa-extensão com as demandas institucionais e sociais; a realização de uma prática acadêmica que contribua para o desenvolvimento da consciência social e política e, por fim, a democratização do conhecimento acadêmico com a articulação e a integração com a sociedade.

Diante disso, o Centro Universitário Cesumar almeja ser reconhecido como uma instituição universitária de referência regional e nacional pela qualidade e compromisso do corpo docente; aquisição de competências institucionais para o desenvolvimento de linhas de pesquisa; consolidação da extensão universitária; qualidade da oferta dos ensinos presencial e a distância; bem-estar e satisfação da comunidade interna; qualidade da gestão acadêmica e administrativa; compromisso social de inclusão; processos de cooperação e parceria com o mundo do trabalho, como também pelo compromisso e relacionamento permanente com os egressos, incentivando a educação continuada.



Janes Fidélis Tomelin

Pró-Reitor de Ensino de EaD

Kátia Solange Coelho

Diretoria de Graduação e Pós

Débora do Nascimento Leite

Diretoria de Design Educacional

Leonardo Spaine

Diretoria de Permanência

Seja bem-vindo(a), caro(a) acadêmico(a)! Você está iniciando um processo de transformação, pois quando investimos em nossa formação, seja ela pessoal ou profissional, nos transformamos e, consequentemente, transformamos também a sociedade na qual estamos inseridos. De que forma o fazemos? Criando oportunidades e/ou estabelecendo mudanças capazes de alcançar um nível de desenvolvimento compatível com os desafios que surgem no mundo contemporâneo.

O Centro Universitário Cesumar mediante o Núcleo de Educação a Distância, o(a) acompanhará durante todo este processo, pois conforme Freire (1996): “Os homens se educam juntos, na transformação do mundo”.

Os materiais produzidos oferecem linguagem dialógica e encontram-se integrados à proposta pedagógica, contribuindo no processo educacional, complementando sua formação profissional, desenvolvendo competências e habilidades, e aplicando conceitos teóricos em situação de realidade, de maneira a inseri-lo no mercado de trabalho. Ou seja, estes materiais têm como principal objetivo “provocar uma aproximação entre você e o conteúdo”, desta forma possibilita o desenvolvimento da autonomia em busca dos conhecimentos necessários para a sua formação pessoal e profissional.

Portanto, nossa distância nesse processo de crescimento e construção do conhecimento deve ser apenas geográfica. Utilize os diversos recursos pedagógicos que o Centro Universitário Cesumar lhe possibilita. Ou seja, acesse regularmente o AVA – Ambiente Virtual de Aprendizagem, interaja nos fóruns e enquetes, assista às aulas ao vivo e participe das discussões. Além disso, lembre-se que existe uma equipe de professores e tutores que se encontra disponível para sanar suas dúvidas e auxiliá-lo(a) em seu processo de aprendizagem, possibilitando-lhe trilhar com tranquilidade e segurança sua trajetória acadêmica.

Professor Esp. Ronie Cesar Tokumoto

É bacharel em Informática pela Universidade Federal do Paraná (2001). Pós-Graduações em Docência no Ensino Superior pela Unicesumar, Pós-Graduação em Tutoria em Educação a Distância pela Faculdade Eficaz e Pós-Graduação em Gestão Escolar Integrada e Práticas Pedagógicas. Experiência na área de educação desde 1993, em escolas profissionalizantes e colégios estaduais. Experiência em Coordenação de Curso Técnico no ano de 2013 e Participação em Palestras e Eventos Organizados para o Curso durante esse mesmo ano. Atualmente, atua como: Professor Mediador no EAD-Unicesumar no curso de Graduação EAD em Engenharia de Software e Gestão em Tecnologia da Informação; Professor do Curso Técnico em Informática ofertado pelo programa Pronatec, também pela Unicesumar; Professor na rede Estadual de Ensino do Paraná em Maringá-PR, nos Cursos Técnicos Profissionalizantes.

SEJA BEM-VINDO(A)!

Caro(a) aluno(a)! Seja bem-vindo(a) à disciplina de Programação Front-End. Sou o professor Ronie Cesar Tokumoto.

Apresento a você o livro que conduzirá seus estudos, auxiliando no aprendizado de Programação Front-End. O livro encontra-se estruturado em cinco unidades e cada uma conta com as seções Reflita, Saiba Mais, Leituras complementares e Atividades de Autoestudo. A seção Reflita apresenta frases que remetem a conteúdos relacionados à disciplina e permitem a livre interpretação e pensamento sobre o tema. O Saiba Mais conta com conteúdos relacionados aos temas da unidade, tendo, geralmente, um link para um conteúdo mais completo, para agregar conhecimento sobre o tema proposto. A seção Leitura Complementar oferece literaturas complementares para auxiliarem no aprendizado com sinopses e comentários. Por último, a seção Atividades de Autoestudo sintetizam os conteúdos abordados por meio de questões diversas para avaliação pessoal, contendo gabaritos para posterior conferência.

A unidade I apresenta conceitos introdutórios sobre Internet, *Front-end*, *Back-end* e produção de conteúdo, trazendo ao aluno uma noção de aspectos fundamentais para compreensão das demais unidades e para que o leitor compreenda de que maneira este livro pode auxiliar o profissional da área de *Webdesign* dentro da sua faixa específica do mercado de trabalho. Trata da diferenciação entre o trabalho com *Back-end* e *Front-End* e ferramentas indicadas para cada nicho, além de tratar de pontos importantes a respeito da produção de conteúdos próprios, excluindo o plágio de material produzido por outros profissionais e focando na criatividade.

A unidade II trabalha com os fundamentos da linguagem HTML tradicional e apresenta os comandos essenciais chamados de TAGs, com suas respectivas formas de uso e definições complementadas com exemplos práticos de fácil utilização, para permitir ao aluno compreender de que forma os *websites* são construídos utilizando essa linguagem fundamental para profissionais da área de *webdesign*.

A unidade III trata da implementação da linguagem HTML em sua versão 5, com ideias inovadoras e o acréscimo de diversos recursos que, juntamente com a linguagem CSS, complementa e simplifica enormemente o desenvolvimento do código HTML, por meio de estilos repletos de recursos que, em sua versão 3, permitem inúmeros efeitos e configurações para texto e imagens.

A unidade IV trabalha conceitos em Javascript e JQuery, mostrando um pouco do potencial da linguagem Javascript que a cada dia se aprimora e cresce em termos de inserção no mercado do desenvolvimento web, permitindo que páginas web sejam muito mais elaboradas, contendo programação mais avançada, que inclui uso de estruturas lógicas mais elaboradas, como laços e estruturas de dados. Já a biblioteca JQuery é um excelente complemento que reduz muito a criação de código para desenvolvimento web e permite que muitas rotinas sejam implementadas com menor quantidade de linhas de código.

APRESENTAÇÃO

Finalmente, a unidade V traz um estudo de caso em que é possível criar um produto completo, simulando a elaboração de um website simples, mas que contenha uma boa base dos conteúdos tratados no livro, servindo de oficina de aprendizado para os alunos e permitindo que eles, tomando por base o estudo de caso, possam evoluir em seus estudos e aprimorar o trabalho realizado, podendo, depois, criar novos *websites*, cada vez mais complexos, e evoluir profissionalmente.

Ao longo deste livro, você terá material suficiente para ingressar no universo do *webdesign* por meio de exemplos, atividades e leituras complementares. O caminho é longo, mas muito gratificante. Esforço sempre será necessário, mas recompensador. Bons estudos!

■ UNIDADE I

PRINCÍPIOS BÁSICOS PARA DESENVOLVIMENTO WEB

15	Introdução	
16	A Internet	
18	Front-End	
19	Back-End	
20	Produção de Conteúdos	
22	Considerações Finais	

■ UNIDADE II

HTML 5

31	Introdução	
32	Tags	
33	Estrutura Básica do Código HTML	
35	Comandos Essenciais	
41	Comandos Complementares	
47	Considerações Finais	



SUMÁRIO

UNIDADE III

HTML 5 E CSS 3

53	Introdução
54	HTML 5 e CSS 3
68	Estrutura Básica de Código CSS
70	Comandos Essenciais
76	Comandos Complementares
83	Considerações Finais

UNIDADE IV

JAVASCRIPT E JQUERY

93	Introdução
94	Javascript e sua Codificação
97	HTML, CSS e Javascript
101	Aquisição e Instalação da JQuery
103	Criação de Scripts
107	Considerações Finais



UNIDADE V

ESTUDO DE CASO

115 Introdução

116 Estruturação do Projeto

118 Produção de Conteúdo

119 Elaboração das Páginas

157 Publicação na Internet

158 Considerações Finais

163 CONCLUSÃO

165 REFERÊNCIAS

166 GABARITO



PRINCÍPIOS BÁSICOS PARA DESENVOLVIMENTO WEB

UNIDADE

I

Objetivos de Aprendizagem

- Mostrar como é a disciplina e seu foco em relação ao desenvolvimento de material para web.
- Organizar breve histórico da Internet.
- Conceituar Front-End e a sua importância no trabalho do desenvolvedor.
- Citar os conceitos relacionados ao Back-End e em quais disciplinas pode-se inserir dentro do curso.
- Mostrar exemplos e ferramentas capazes de auxiliar na produção de conteúdos para uso no desenvolvimento.

Plano de Estudo

A seguir, apresentam-se os tópicos que você estudará nesta unidade:

- A Internet
- Front-End
- Back-End
- Produção de conteúdos

INTRODUÇÃO

Olá, caro(a) aluno(a),

Temos em mãos um livro curioso e útil, pois ele traz conteúdos relacionados a uma das áreas mais interessantes da informática nos últimos tempos, que trata da criação de conteúdo para Internet.

Existem diversas áreas nas quais é possível trabalhar dentro da Internet, sendo algumas delas, por exemplo, o desenvolvimento de páginas web, que tenham foco informativo, interativo ou de comércio eletrônico.

Temos, também, profissionais que se especializam na produção de conteúdo para a internet, por exemplo, imagens, animações, vídeos, músicas e todo tipo de material multimídia, desenvolvido em softwares gratuitos ou não.

Existem objetos de aprendizagem: softwares, aplicativos ou jogos, todos online, que através da Internet despertam, também, objetivos de muitos interessados em trabalhar com as tecnologias voltadas para a rede mundial de computadores.

Como não é o objetivo da disciplina abranger tantas áreas da produção para Internet, será tratada, aqui, como objetivo principal, a produção de conteúdo para criação de websites informativos que possam agregar funções mais avançadas. No futuro, você, aluno(a) pode se especializar em áreas como o comércio eletrônico, por exemplo.

Este material, é focado em Front-End, responsável por aquilo que se vê em uma página web e a forma como podemos interagir com ela.

Assim, é muito importante observar o quanto a criatividade e a pesquisa pelos aspectos que sejam realmente atrativos e que estejam em vigência, no que diz respeito à aparência a layout de websites conhecidos, para seguir tendências e não criar materiais de pouco interesse para o público em geral.

Claro que cada um dará seu toque pessoal e, em alguns casos, pode até ser capaz de ditar novas tendências na rede, mas a grande maioria segue aquilo que está em uso no momento, podendo, sem aviso, seguir novas tendências ou evoluções tecnológicas ou de software.

Vamos lá? Bons estudos!



©shutterstock

Figura 1: Ilustração sobre a rede mundial de computadores.

A INTERNET

A famosa rede mundial de computadores existe há um bom tempo, mas sua concepção vai muito mais além do que muitos acreditam, pois a base de seu funcionamento não tinha o mesmo propósito no início e, com o tempo, ocorreram a sua expansão, as mudanças na forma de uso, o alcance e a sua complexidade.

Na década de 60, pesquisas encomendadas pelo governo norte-americano sugeriam a criação de uma forma de comunicação em longa distância, capaz de resistir até a ataques em grande escala que, na época, o país poderia sofrer.

Surgiu, então, a Arpanet entre as décadas 60 e 70, interligando algumas instituições de ensino, além de outras iniciativas que utilizaram protocolos de comunicação diversos e que, depois, precisaram ter um novo protocolo, capaz de interligar essas redes, tendo como condutora do processo, a própria Arpanet.

Assim, aos poucos, resultados muito expressivos foram sendo conseguidos e, inclusive, devido à maior popularização do acesso à rede Arpanet. Depois, a parte da rede ligada a dados militares foi separada por motivos óbvios de segurança.

Durante a década de 70, o termo internet começou a ser utilizado e os conteúdos foram adicionados à rede, mesmo que em pequenos volumes, pois a rede mundial de computadores não havia sido, ainda, popularizada, sendo restrita ao meio acadêmico e não comercial, muito menos difundida ao público em geral. No início dos anos 80, o protocolo TCP/IP (Transmission Control Protocol / Internet Protocol) foi desenvolvido e um processo efetivo de unificação foi

instalado, gerando interesse de outra grande Fundação chamada NSF (National Science Foundation), criando a NSFNET (rede da NSF).

Durante a década de 80, a rede foi crescendo e se popularizando ao ponto de surgirem os primeiros provedores de acesso, oferecendo baixas taxas de transmissão, mas que, rapidamente, foram evoluindo.

Durante os anos 90, a Arpanet e a NSFNET saíram de cena aos poucos, para que a Internet realmente começasse a avançar pelo mundo, tendo sua ampla difusão pelo território norte-americano.

A partir dos anos 2000, já não se via mais limites para o alcance da Internet, pois ela crescia de forma impressionante em tamanho, alcance e evoluções tecnológicas, até chegar aos níveis atuais de velocidade e tecnologia.

SAIBA MAIS



O Marco Civil da Internet é um projeto de lei que tem por objetivo consolidar direitos e deveres, além de estabelecer os princípios legais de uso e desenvolvimento da Internet no país.

Em agosto de 2011, o projeto de lei (2.126/2011) foi apresentado à Câmara dos Deputados para audiências e apreciação de seu conteúdo pela própria Internet a partir do site da Câmara, para que todos pudessem, abertamente, realizar consultas e fazer sugestões sobre o projeto.

A proposição nasceu de uma iniciativa da Secretaria de Assuntos Legislativos do Ministério da Justiça, que, em parceria com o Centro de Tecnologia e Sociedade da Escola de Direito da Fundação Getúlio Vargas no Rio de Janeiro, estabeleceu um processo aberto, colaborativo e inédito para a formulação de um marco civil brasileiro para uso da Internet. Seu principal elemento de inspiração foi a Resolução de 2009 do Comitê Gestor da Internet no Brasil (CGI.br) intitulada “Os princípios para a governança e uso da Internet” (RESOLUÇÃO CGI.br/RES/2009/003/P).

Fonte: O CGI.BR...(online).



REFLITA

A Internet é um território ainda complexo devido à sua liberdade e a falta de garantia de um eficaz controle de tudo que nela ocorre. Isso pode ser um ponto de partida para o questionamento sobre ela ser livre ou controlada, e se deveria ou não ser dividida em uma rede pública e não controlada e outra comercial e controlada.

Fonte: o autor.

FRONT-END

Na construção de *websites*, é possível que um banco de dados seja agregado para inserção, edição, consulta ou exclusão de dados, por exemplo.

No servidor, onde teríamos a parte programada, responsável por realizar os procedimentos citados, automatizados via programação, criada para tal, chamamos de parte do sistema de *back-end*. Esta é a parte que, normalmente, fica oculta dos usuários e não permite interação direta com estes.

Já, nos locais onde o usuário acessa o sistema de interação entre ele e o banco de dados do *website* ativo no servidor, temos apenas janelas para essa interação. Isso possibilita deixar a forma como o sistema processa as requisições invisíveis ao usuário e este apenas, tendo como experiência de uso, o que é oferecido pelo *front-end*.

Front-end representa, então, aquilo que o usuário vê e pode interagir, sendo, fundamental que isso seja planejado de acordo com o tipo de experiência de uso e impressão que se deseja causar.

Alguns *web designers* podem preferir um visual mais enxuto, leve e funcional para conteúdos acessados

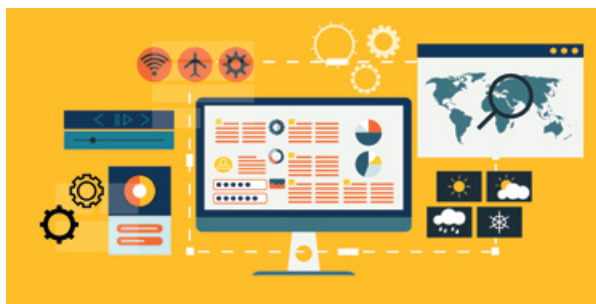


Figura 2: Ilustração sobre Front-end

por determinado público, ou páginas repletas de efeitos e cores, contendo muito conteúdo multimídia para impressões mais marcantes em determinadas situações e público.

É extremamente importante que reuniões sejam feitas entre produtores e clientes, no sentido de identificar o que estes querem em relação ao estilo que o site deverá mostrar. É preciso informar detalhes que auxiliem o direcionamento para a forma de interação mais adequada em relação ao seu nicho de público e mercado.

Para a programação *Front-end*, existem diversas linguagens desenvolvidas, justamente, para a estruturação das páginas web, como html, CSS e Javascript, tendo opções complementares com JQuery ou Foundation, que agregam facilidades no desenvolvimento dessas linguagens.

SAIBA MAIS



Com a simplificação na aparência dos websites, devido ao avanço dos dispositivos móveis, fica a impressão de que o conteúdo é o que importa agora, e não mais a aparência, até porque os dispositivos móveis têm, em geral, interfaces menores e, um visual muito carregado dificulta a visualização do que realmente se busca: conteúdo. Isso deve afetar o trabalho do web designer no futuro.

Fonte: o autor.

BACK-END

Back-end é a programação pesada, capaz de realizar os processos, normalmente, invisíveis ao usuário, é feita por outras linguagens complementares àquelas citadas como linguagens de programação *front-end*.

Essas linguagens são muito mais complexas e capazes de criar verdadeiros *softwares* que rodam por trás do *front-end*, permitindo que os dados sejam tratados, decisões sejam tomadas automaticamente, ou repetições executadas de forma controlada.

As tarefas mais complexas são realizadas por códigos mais elaborados, criados em linguagens como C#, ASP, Java, PHP ou Python, por exemplo. Essas linguagens são capazes de tratar eventos que as linguagens *front-end* não contemplam devido à sua finalidade.

Sistemas mais complexos como *e-commerce*, bancos de dados e sites gestores de conteúdo são desenvolvidos, unindo os dois tipos de linguagem e permitindo a criação de verdadeiros sistemas de gerenciamento, como aqueles desenvolvidos para computadores isolados ou redes locais em linguagens como Delphi, C e Visual Basic.

PRODUÇÃO DE CONTEÚDOS

Produzir material próprio ou para terceiros é fundamental na criação de *web-sites* e multimídia, mas, na Internet, atualmente, não é o que se vê em 100% dos locais disponíveis. A tendência é que nunca se chegue a esta porcentagem, pois, da mesma forma que existem inúmeras pessoas que produzem conteúdos com opiniões pessoais, experiências ou pesquisas fundamentadas, existem aqueles que apenas se apropriam daquilo que foi produzido por outros, sem

citar fontes de pesquisa, tomando para si os direitos dos conteúdos e reduzindo, de forma considerável, a confiabilidade daquilo que existe disponível na rede.

Felizmente, o público em geral começa a tomar consciência e, quando há a necessidade de se realizar buscas, procura-se conteúdos mais consistentes e que mostrem maior qualidade, pois, com a velocidade de acesso e buscas na rede, é possível observar os resultados até encontrar aquele que pareça mais confiável.



©Bruno Cesar Pardinho Figueiredo

Figura 3: Ilustração sobre a criatividade na produção.

Agora, quando se produz um material, este deve ser devidamente produzido, utilizando os equipamentos adequados com condições adequadas, desde que possível. É importante preservar os direitos daqueles que possam vir a ter sua imagem ou seus bens vinculados a estes materiais, pois a exposição sem permissão pode acarretar problemas jurídicos, inclusive.

Evitar rostos e identificação de locais e bens pode ser interessante para que o material seja o mais livre de empecilhos no futuro e, o uso de partes não identificáveis de pessoas e bens, pode ser uma boa solução.

Um rosto é identificável, mas apenas uma pequena parte dele pode não ser, assim como partes que sejam ocultadas na imagem por meio de efeitos ou tarjas que obstruam a visão, para que a identificação não seja feita. Seja qual for a técnica utilizada para isso, é preciso preservar os direitos alheios em primeiro lugar.

Pessoas contratadas para sessões de fotos são um caso diferente, pois estão cientes da exposição e podem ou não cobrar por isso, fato muito comum com profissionais ou celebridades que têm seus ganhos atrelados à exposição pessoal.

Existem exemplos, como o sistema de mapas da Google, que possui mecanismos para ocultamento de rostos e placas de carros devido a problemas ocorridos logo no início de suas atividades na Internet.

O uso de estúdios próprios, para a criação de cada tipo de material contribui muito para a qualidade do produto final, da mesma forma que os equipamentos e *softwares*.

Nem sempre se pode contar com estúdios próprios, e profissionais iniciantes, na maioria, são aqueles menos estruturados e que têm menos acesso a estes recursos. Evidentemente que esse fato não impede que esses profissionais deixem de realizar bons trabalhos, mas precisam de maior inspiração criativa e maior empenho para atingir resultados equivalentes, ou tão bons quanto aqueles obtidos por profissionais com maior tempo de mercado e com melhores condições de trabalho. Lembrando que, tais condições podem limitar, justamente, a principal qualidade do produtor que é a criatividade.



SAIBA MAIS

O que é Modern UI?

Criar um *website* depende muito da criatividade do *web designer*, mas depende também do profissional estar atualizado, buscar aprender e aplicar novas tecnologias nos trabalhos que realiza no dia a dia.

O estilo desenvolvido pela Microsoft serviu de base para a interface dos seus sistemas operacionais mais recentes, permitindo um visual limpo e eficiente, oferecendo interação e agilidade no uso.

O artigo mostra, além de características, um completo exemplo de criação, utilizando a IDE do Visual Studio da própria Microsoft.

Acesse o link para saber mais disponível em: <<https://msdn.microsoft.com/pt-br/library/dn205307.aspx>>. Disponível em: 21 dez. 2015.

Fonte: O autor.

CONSIDERAÇÕES FINAIS

A Internet, visivelmente, move diversas áreas da sociedade e as influencia diretamente, sendo responsável pela produção, divulgação, receitas e despesas e, até, pela completa dependência de si para existir.

Temos, nesta unidade, uma introdução a assuntos pertinentes e que serão importantes para a compreensão das unidades seguintes, pois serão trabalhadas, gradualmente, linguagens e ferramentas capazes de produzir páginas bem estruturadas para criar um ambiente agradável em termos de experiência de navegação.

Por meio de um Front-end bem trabalhado e criativo, contendo material bem produzido, é possível que uma grande quantidade de usuários possam ser satisfeitos, sendo influenciados em suas tomadas de decisões durante o tempo em que estejam navegando por estas páginas.

Assim, trabalhar com *Front-end*, mais voltado ao *web design*, ou a preferência em trabalhar na programação mais complexa e invisível do *Back-end*, é uma responsabilidade sua, caro(a) aluno(a).

Produzir material de qualidade para ser utilizado na Internet também é, por si só, um nicho de mercado que pode trazer bons benefícios pessoais e financeiros para aqueles que tenham a capacidade técnica e criativa para produzir excelentes materiais, diferenciados dos demais.

A Internet está em constante evolução e a cada momento uma nova ideia surge, podendo se tornar uma tendência ou não, de acordo com o nível de qualidade e de criatividade existente nos conteúdos que nela são inseridos e divulgados. As redes sociais, são exemplos de conteúdo produzido a baixo custo, pois são, em sua grande maioria, criado por pessoas que estejam conectadas, e que queiram fazer suas contribuições livremente.

Assim, partindo dos princípios descritos nesta unidade, iniciaremos na sequência, uma nova etapa, o processo de codificação em linguagem html tradicional, com uma série de exemplos práticos para auxiliar os estudos e compreensão do assunto.

ATIVIDADES



1. A Internet iniciou suas atividades entre o final dos anos 80 e início dos anos 90. Qual foi o primeiro nome oficial utilizado dentre as opções abaixo? Assinale a alternativa correta.
 - a. Arpanet.
 - b. Mundinet.
 - c. Globalcom.
 - d. Intranet.
 - e. Skynet.

2. *Front-end* é o termo mais importante da disciplina. Sobre o termo, observe as afirmações abaixo e assinale a alternativa correta.
 - I. Toda a programação em java faz parte apenas do *Front-End*.
 - II. A linguagem html é própria para desenvolvimento *Front-end*.
 - III. *Front-end* se refere mais à aparência do que do banco de dados em um *web-site* por exemplo.
 - IV. O termo *Front-end* se refere ao processo em que o usuário utiliza um *website* pronto, apenas, para cadastrar dados.

Assinale a alternativa correta:

 - a. Apenas I e II estão corretas.
 - b. Apenas II e III estão corretas.
 - c. Apenas I está correta.
 - d. Apenas II, III e IV estão corretas.
 - e. Nenhuma das alternativas está correta.

3. A respeito do termo Back-end, responsável pela codificação mais complexa do sistema, assinale Verdadeiro (V) ou Falso (F):
 - () Normalmente, usa-se a linguagem CSS para programar Back-end.
 - () Normalmente, acesso a bancos de dados é feito pelo Back-end.
 - () Na verdade, os termos Front-end e Back-end se referem a mesma parte de um website.

ATIVIDADES



4. O termo plágio é muito utilizado no meio acadêmico. Defina, com suas palavras, o que seria ou a que se refere esse termo.

5. Produzir material para o desenvolvimento de conteúdo para Internet é algo muito importante, pois permite uma maior personalização de um *website*, por exemplo. Que tipos de materiais podem ser produzidos para que se tenha um website criativo e de qualidade?



As linguagens de programação são ferramentas extraordinárias de criação de diversos tipos de produtos, podendo ir desde simples conteúdos estáticos para páginas web até softwares complexos como jogos e aplicativos comerciais.

O tipo de produto indica o tipo de linguagem mais indicada, pois uma linguagem de programação que permita acesso a recursos chamados de baixo nível capazes de manipular hardware como a linguagem C é indicada para programação de dispositivos com sistemas embarcados por exemplo.

Linguagens voltadas para desenvolvimento de sistemas comerciais como os chamados ERP (Planejamento de Recurso Corporativo) representam Sistemas Integrados de Gestão Comercial. Outros tipos de linguagens que sejam capazes de trabalhar com bases de dados, como Delphi para sistemas ERP convencionais e PHP para sistemas baseados web, trabalham em conjunto com outra linguagem, a chamada SQL, própria para manipulação de bases de dados.

Outro tipo de linguagem muito utilizada, para produção de conteúdo web, como HTML ou CSS, temas deste livro, são complementadas por opções, como javascript (mais robusta e complexa).

Existem, ainda, as linguagens mais recentes, criadas por gigantes da área de TI, como o Facebook, que criou a linguagem HACK, própria para desenvolvimento web que pode ser integrado com HTML ou PHP.

A linguagem criada pela Google chamada GO segue as características da linguagem C e se diferencia em detalhes técnicos na parte de orientação a objetos (técnica muito usada para programação atualmente).

A Apple desenvolveu uma linguagem chamada SWIFT para criação de aplicações para sua linha de produtos para substituir a linguagem Objective C padrão até a criação dessa nova opção.

Assim, é fácil perceber a quantidade de opções existentes, e é importante saber que não é interessante tentar conhecer todas, pois são diferentes e para finalidades diferentes. Focar em uma área de atividade e conhecer suas linguagens ou apenas conhecer uma delas é suficiente para o desenvolvimento de produtos.





LIVRO

Empreendedorismo na Internet

Dailton Felipini

Editora: Brasport Editora

Sinopse: este livro foi desenvolvido para atender às necessidades do mercado e de empresários empreendedores para auxiliar na escolha de uma forma de inserção no mercado digital, e qual produto pode ser mais interessante para a exploração dessa nova alternativa.



LIVRO

Mídia e produção audiovisual: uma introdução

Marcia Nogueira Alves

Editora: Intersaberes

Sinopse: o livro trata da correta produção de imagens, som de forma precisa, mostrando como é o mercado de massa e de que forma o material multimídia criado influencia na qualidade de uma publicidade.

Comentário: o livro é muito interessante, pois cita os processos de desenvolvimento de conteúdo como a criação de jingles publicitários e montagens para tentar aumentar a possibilidade de um usuário passar do estado de navegador a comprador na Internet.



HTML 5

UNIDADE



Objetivos de Aprendizagem

- Conceituar a linguagem e as características que a diferencia de outros tipos de linguagem.
- Descrever a ideia de tag e mostrar como são utilizadas genericamente.
- Citar todas as tags essenciais de um código html 5 e seus significados.
- Elencar as principais tags indicando sintaxe, conceitos e exemplos.
- Elencar outras tags úteis indicando sintaxe, conceitos e exemplos.

Plano de Estudo

A seguir, apresentam-se os tópicos que você estudará nesta unidade:

- Tags
- Estrutura Básica do Código Html
- Comandos Essenciais
- Comandos Complementares

INTRODUÇÃO

Olá, caro(a) aluno(a)! Seja bem vindo(a) a segunda unidade do nosso material. Vamos falar sobre o HTML 5, que é uma linguagem, que tem uma finalidade diferente e também é tratada de forma distinta das demais, pois não é utilizada para a criação de softwares em geral como as anteriores, e por isto não é compilada como se diz nas demais linguagens citadas.

O termo html é uma abreviação de *HyperText Markup Language* e indica que esta é uma linguagem de marcação, como seu próprio nome diz, usada para estruturação de páginas para a Internet através de comandos chamados tags, próprios para esta linguagem.

Um código html pode ser criado em qualquer editor de textos ou software próprio para desenvolvimento de programas que contenha um ambiente de desenvolvimento já elaborado, para facilitar o trabalho com o uso de cores e outras funções.

Um software como o Bloco de Notas do Windows ou qualquer similar, estilo notepad desse mesmo sistema operacional ou de qualquer outro, pode ser utilizado sem problemas, desde que os devidos cuidados sejam tomados no momento da gravação dos arquivos html.

Os arquivos gerados em html são estruturas para construção de páginas que são interpretadas por navegadores e, de acordo com a disposição das tags utilizadas no código, criam todo o layout de uma página, podendo esta, conter elementos de texto formatado, conteúdo multimídia e programação mais específica, com o auxílio de componentes adicionais aceitos pela html como CSS, Javascript, PHP etc.

Nesta unidade, trataremos das tags html e sua sintaxe, de exemplos ilustrativos, com o intuito de criar um conjunto interligado de páginas gravadas em arquivos html separados, que comporão um pequeno *website* modelo, para estudo que pode ser alterado e remontado para outros temas e contendo outros conteúdos para criar novos *websites* exemplo para prática e melhor aprendizado.

Vamos lá!

TAGS

Tag é o nome dado aos comandos aceitos na linguagem html, funcionam de forma um pouco diferente de outras linguagens de programação compiladas, pois tags mal utilizadas não impedem que o código em si funcione. Assim, o navegador tem o princípio de interpretar e estruturar todas as tags que ele possa compreender e ignorar tags incorretas ou mal estruturadas de acordo com sua sintaxe.

A lista de tags é longa e varia em função da versão da linguagem html e do navegador, que pode ou não ser totalmente compatível com cada tag. Nunca é cem por cento garantido que um código html seja totalmente funcional em qualquer equipamento, pois as diferentes opções de navegadores e suas versões e atualizações instaladas interferem no grau de compatibilidade entre navegador e código html.

Praticamente todo código html e suas tags são componentes que fazem parte do chamado *flow content*, que são relativos à construção das páginas web, exceto pela parte relativa aos metadados do código que são opcionais e não interferem na estruturação das páginas em geral.

Algumas destas tags são tema deste material, mas existe outras que podem ser estudadas para complementar seus estudos de acordo com sua necessidade ou curiosidade.

Existem tags que são essenciais para a disciplina e serão tratadas de forma individual. De forma que sejam incorporadas a medida que forem apresentadas para a construção de códigos cada vez mais completos e complexos.

Uma tag da linguagem html é representada através de comandos inseridos com seus respectivos parâmetros dentro de sinais de < e > para delimitar início e fim dos comandos e facilitar a compreensão do código em si, pois o uso de mudanças de linha não encerra uma tag, e esta pode se estender por várias linhas sem problemas até que o sinal > seja utilizado para finalizá-la.

Para se codificar em linguagem html, é possível utilizar, como dito anteriormente, o bloco de notas, mas existem, também, ferramentas com interfaces de produção no estilo WYSIWYG (*What You See Is What You Get*) significando que o *software* mostra enquanto se codifica, algo próximo do que será o resultado pronto, gerado pelo código.

Softwares como o *Adobe Dreamweaver*, por exemplo, são capazes de mostrar como fica a estrutura criada pelas tags enquanto se codifica, simplificando muito esta criação de programas e ainda facilitando a localização de erros no próprio ambiente. Isso proporciona integração e reduz a quantidade de trabalho de manutenção de um código html.

Menus com opções prontas permitem que partes de código sejam gerados com poucos cliques e menor quantidade de digitação, mas essa facilidade toda pode prejudicar o nível de aprendizado da linguagem e compreensão do código, gerado para futuras manutenções ou atualizações.

Para o aprendizado, os exemplos e exercícios partirão da ideia do uso de um editor simples sem recursos integrados de construção de layouts e demais itens de uma página.

ESTRUTURA BÁSICA DO CÓDIGO HTML

O html segue um padrão não muito rígido na construção de seus códigos, permitindo letras maiúsculas e minúsculas ou tags mal formuladas que são ignoradas, sem causar erros, muitas vezes. Tags sem o uso da tag `</>` para encerrá-la são normalmente aceitos pelos navegadores sem que ocorram problemas na estruturação das páginas web.

Na versão XHTML, todo código deve ser muito bem formatado, seguindo a correta sintaxe e aplicação das tags, todas com letras minúsculas, devidamente fechadas com a tag `</>` e devidamente aninhadas, e tags vazias devem ser fechadas ao final de todos os seus parâmetros como nos exemplos da figura 1.

TAG PADRÃO HTML - ABERTURA	TAG PADRÃO XHTML - FECHAMENTO
<code>
</code>	<code>
</code>
<code></code>	<code></code>
<code><h1></code>	<code><h1> </h1></code>

Quadro 1: Tabela ilustrativa de abertura e fechamento de tags

Fonte: o autor.

```
<!DOCTYPE html>
<html>
<body>
</body>
</html>
```

Figura 1: Exemplo de código essencial para html

Fonte: O autor

O Doctype indica para o navegador e para outros meios qual a especificação de código utilizar. O Doctype não é uma tag do HTML, mas uma instrução para que o browser tenha informações sobre qual versão de código a marcação foi escrita.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Mensagem</title>
</head>
<body>
</body>
</html>
```

Figura 2: Exemplo de código essencial para html contendo cabeçalho

Fonte: O autor.

O código HTML é uma série de elementos em aninhados, em que alguns elementos estão contidos em outros e assim por diante. O elemento principal dessa grande estrutura é sempre a tag HTML que deve ser fechada ao final do código.

A Tag HEAD é onde fica toda a parte da página que representa o cabeçalho responsável por diversos tipos de ajustes. No HEAD ficam os chamados metadados, que são informações sobre a página e o conteúdo ali publicado.

Neste exemplo há uma metatag¹ chamada charset, responsável por definir qual tabela de caracteres ASCII a página está utilizando caso precise de caracteres especiais, além daqueles mais convencionais do teclado como $\frac{1}{2}$ e $\frac{3}{4}$, por exemplo. As alternativas mais utilizadas, para esta metatag como definição do conjunto de caracteres da tabela ASCII utilizáveis por charset, são “UTF-8” e “ISO-8859-1”.

¹ Metatags são tags para informar ao navegador informações adicionais ao conteúdos a serem exibidos nas páginas web. São em parte importantes para uma correta interpretação do código htm

Existe, também, uma tag muito comum dentro de HEAD chamada TITLE que serve apenas para indicar qual mensagem deve ser exibida na barra de título do navegador ou da aba do navegador onde a página está sendo exibida. Esta deve ser fechada através de </TITLE> para evitar falhas no código html.

Após a indicação de todos os comandos para o cabeçalho da página, é preciso fechar a tag através de </HEAD> para permitir que a codificação html passe para sua próxima etapa do código chamada BODY. Nesta parte, todas as tags que estruturam uma página web são colocadas, e assim, formam o corpo de uma página web.

Fora das tags, qualquer texto digitado no código é considerado conteúdo a ser exibido no navegador, e por isto é essencial o correto uso dos sinais < e > para delimitar tags e impedir que o navegador se confunda entre uma e outra.

COMANDOS ESSENCIAIS

Como todo texto fora das tags é exibido no navegador, este segue o padrão de texto configurado no software, mas pode ser formatado para ter maior destaque e melhorar a impressão visual causada no usuário.

Podemos usar as tags <H1>, <H2>, <H3>, <H4>, <H5> e <H6> para definir de forma simples o tamanho da letra, seguindo o raciocínio de que H1 representa uma letra de tamanho maior e H6 de tamanho menor, sem esquecer que estas tags precisam ser fechadas com </H1>, </H2>, </H3>, </H4>, </H5> e </H6> respectivamente.

Também temos a tag <P> que permite a formatação de parágrafos e aceita vários tipos de ajustes de fonte, cor e alinhamento do texto através do uso de um parâmetro chamado STYLE. O quadro 2 a seguir contém alguns destes parâmetros e ajudará a compreender a forma de uso dos mesmos através de exemplos comentados.

EXEMPLO	COMENTÁRIO
<code><p style="color:red">TEXTO</p></code>	Seta a cor da fonte para vermelho.
<code><p style="font-family:courier">TEXTO</p></code>	Ajusta a fonte para o texto.
<code><p style="font-size:160%">TEXTO</p></code>	Ajusta o tamanho da fonte.

Quadro 2: Exemplos de sintaxe da tag <P>

Fonte: o autor.

Existem, também, outros ajustes que influenciam na formação do texto também precisam ser fechadas, e indicadas da mesma forma que na tabela do quadro 3.

EXEMPLO	COMENTÁRIO
<code><p>TEXTO.</p></code>	Efeito de negrito no texto.
<code><p>TEXTO.</p></code>	Efeito de texto reforçado (como negrito).
<code><p><i>TEXTO</i>.</p></code>	Efeito de itálico no texto.
<code><p><small>TEXTO</small>.</p></code>	Efeito de redução da fonte no texto.
<code><p><mark>TEXTO.</mark></code>	Efeito de marca texto.
<code><p>TEXTO.</p></code>	Efeito de texto excluído (traçado).
<code><p><sub>TEXTO</sub>.</p></code>	Efeito de texto subscrito.
<code><p><sup>TEXTO</sup>.</p></code>	Efeito de texto sobrescrito.

Quadro 3: Tags para formatação de texto em html

Fonte: o autor.

Comentários podem ser inseridos em qualquer ponto de um código sem afetar as demais tags. Isso mantém o código documentado, facilitando o suporte em futuras manutenções e atualizações. Comentários são permitidos em qualquer lugar do código e podem orientar a compreensão das funções que o mesmo exerce.

Este é um exemplo de sintaxe de comentários em html, observando que os sinais utilizados para delimitar a tag de comentário são `<!--` e `-->` sem espaços entre os símbolos, formando os delimitadores para os comentários.

```
<!-- Comentário -->
```

O uso de imagens é muito útil e permite que a qualidade visual de páginas melhore sensivelmente, desde que não sejam imagens plagiadas que possam culminar em

problemas futuros ou prejudicar o conteúdo como um todo em função de não ter relação com o tema ou não ser adequada devido à excessos ou mal colocação.

A tag IMG permite o uso de imagens de diversos tipos e não necessita de uma segunda tag de fechamento, como as demais citadas, pois executa uma função única e não pode agregar outras dentro dela. Sua sintaxe é simples e várias outras tags seguem modelo semelhante em que alguns parâmetros definem qual é a imagem a ser exibida, além de detalhes como tamanho em pixels ou porcentagem.

```

```

Neste exemplo, temos a tag IMG sendo utilizada para exibir a imagem definida pelo parâmetro SRC="", sendo que entre aspas deve ser introduzido o nome do arquivo.extensão gravado na mesma pasta do arquivo HTML ou o disco\caminho\arquivo.extensão, em que se encontra a imagem gravada, ou finalmente o endereço web no qual a foto pode ser adquirida no momento em que a página for exibida no navegador.

O parâmetro ALT="" indica um texto que é exibido em caso de falha na exibição da imagem, WIDTH="" indica a largura da imagem na tela, podendo ser definida em pixels ou porcentagem de acordo com o hábito ou necessidade do programador e HEIGHT="" que define a altura da imagem também em pixels ou porcentagem.

Outro recurso essencial na construção de web sites são os chamados links internos que servem para interligar páginas do próprio website ou pontos específicos da mesma página aos externos, que fazem ponte entre o próprio site e outros da Internet, por exemplo.

A tag responsável por trabalhar com links é <a> que precisa de alguns parâmetros para direcionar o funcionamento do link, sendo um deles href="" que indica o destino do link (interno ou externo), que deve ser digitado corretamente, seguindo o padrão URL para links externos e outro bastante interessante chamado target="", que indica como o link deverá trazer o destino (mesma página ou aba, ou outra diferente).

```
<a href="url" target="_blank">link text</a>
```

Na quadro 4 temos uma tabela indicativa de opções para uso no parâmetro target e seu efeito prático.²

VALOR PARA TARGET	EFEITO PRÁTICO
_blank	Abre nova página em nova aba ou janela.
_self	Abre nova página na mesma aba ou janela.
_top	Abre nova página em nova aba ou janela cheia em caso de antes a página estar dividida em frames.
_parent	Abre página no frame que contém a tag FRAMESET.
nome_da_janela	Abre nova página no frame indicado dentro da página.

Fonte: o autor.

Outro recurso simples que complementa o texto em páginas web é o uso de marcadores como pontos ou números para listar tópicos ou itens em listas.

Existem listagens não numeradas, que não indicam ordenação e utiliza símbolos para identificação dos itens. Estas listas são delimitadas por tags e .

No quadro 5, temos um índice de marcadores aceitos em listas não ordenadas, com suas respectivas descrições para melhor compreensão, lembrando que estes valores de tipos de marcadores devem ser indicados através de um parâmetro STYLE="" dentro da tag .

VALOR PARA TARGET	DESCRIÇÃO
list-style-type:disc	Marcador em forma de círculo cheio.
list-style-type:circle	Marcador em forma de círculo vazio.
list-style-type:square	Marcador em forma de quadrado.
list-style-type:none	Listagem sem símbolo marcador.

Quadro 5: Tabela indicativa de valores possíveis para parâmetro

Fonte: o autor.

² O uso de Frames não é mais uma técnica muito utilizada em novos trabalhos, mas ainda é utilizada, em diversos websites, para dividir uma página em divisões para nelas acrescentar conteúdos diversos como menus, banners e conteúdos.

Já os itens da lista devem ser limitados pelas tags e , sendo que cada um deles deve ser delimitado.

Listas ordenadas utilizam as tags e para delimitar listas com números ou letras ordenados através de quaisquer opções de ordenação definidas pelo parâmetro indicado no quadro 6.

TIPOS DE MARCADORES	DESCRIÇÃO
Type="1"	Lista numerada de 1 a n.
Type="A"	Lista ordenada de A a Z.
Type="a"	Lista ordenada de a a z.
Type="I"	Lista ordenada em números romanos maiúsculos.
Type="i"	Lista ordenada em números romanos minúsculos.

Quadro 6: Tabela de opções de marcação para listas ordenadas

Fonte: O autor.

Existe um terceiro tipo de lista com tópicos subdivididos em níveis que pode também ser útil na elaboração de conteúdos em uma página web, delimitada pelas tags <DL> e </DL>, e tendo as tags <DT> e <DD> para iniciar os níveis e </DT> e </DD> para encerrar as respectivas tags de início.

Um exemplo de uso dos marcadores é mostrado na figura 3, em que é exibida uma lista não ordenada, outra ordenada, e finalmente uma descritiva com tópicos em níveis indicados por símbolos que foram digitados junto ao texto do tópico apenas para efeito estético.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<title>Mensagem</title>
</head>
<body>
<ul style="list-style-type:none">
<li>Item</li>
<li>Item</li>
<li>Item</li>
</ul>
```

```
<ol type="1">
  <li>Item</li>
  <li>Item</li>
  <li>Item</li>
</ol>
<dl>
  <dt>* Item</dt>
  <dd>+ Subitem</dd>
  <dt>* Item</dt>
  <dd>+ Subitem</dd>
</dl>
</body>
</html>
```

Figura 3: Exemplo de código para geração de listas variadas

Fonte: o autor.



SAIBA MAIS

Cores são atributos essenciais na elaboração do design e na construção do código html, de forma a criar páginas que combinem texto e cores para melhorar a aparência, a facilidade de leitura e compreensão dos conteúdos inseridos nelas.

Existe uma grande quantidade de cores que podem ser utilizadas, mas em função desta mesma variedade é difícil lembrar tantos nomes em inglês ou códigos em sistema de numeração hexadecimal.

Por isto, o ideal é guardar apenas aquelas cores mais utilizadas ou consultar tabelas de cores disponíveis em inúmeros sites na Internet ou materiais impressos. Basta uma pesquisa simples para encontrar tais tabelas e poder consultar seus códigos ou nomes.

COMANDOS COMPLEMENTARES

Imagens podem ser mapeadas de forma a permitir a criação de links em partes da imagem, oferecendo um recurso muito interessante para personalizar e oferecer uma melhor interação entre usuário e página.

Na figura 4, temos um exemplo de código html completo que demonstra o uso deste recurso de mapeamento de uma imagem exemplo, chamada imagem.jpg, que, propositalmente, foi desenhada em quatro partes, em que cada parte será responsável por representar um link para um destino diferente.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Mensagem</title>
</head>
<body>
  

  <map name="imagem">
    <area shape="rect" coords="0, 0, 99, 99" alt="Microsoft"
href="http://www.microsoft.com.br">
    <area shape="rect" coords="101, 0, 200, 99" alt="Google"
href="http://www.google.com">
    <area shape="rect" coords="0, 101, 99, 200" alt="Wal-
mart" href="http://www.walmart.com">
    <area shape="rect" coords="101, 101, 200, 200" alt="Uni-
cesumar" href="http://www.cesumar.br">
  </map>

</body>
</html>
```

Figura 4: Exemplo de código com mapeamento de imagem em links

Fonte: o autor.

A tag , usada neste código, insere a imagem criada especificamente para este exemplo, com o nome imagem.png, tendo sua área dividida em quatro partes de cores diferentes, para indicar as áreas de cada link.

Dentro desta tag ainda temos o parâmetro ALT="", para indicar um texto que é exibido no navegador no lugar de uma imagem que não pode ser exibida por algum motivo.

Outro parâmetro é USEMAP="#", que criará uma âncora que permitirá ao resto do código html chamar esta tag quando necessário, como acontece na tag <MAP> que tem como parâmetro NAME="", que indicará o nome criado no momento da ancoragem do comando USEMAP.

Dentro da tag <MAP>, são inseridas quantas tags <AREA> forem necessárias para incluir todas as partes mapeadas em links dentro da imagem definida na tag ancorada com USEMAP="#".

Esta tag <AREA> tem como parâmetros a forma para a área indicada por SHAPE que pode receber valores como "RECT" ou "CIRCLE"; COORDS="", que define as coordenadas da forma, indicando sua posição e limitando seu tamanho; ALT="" para exibir, também, um texto alternativo como na tag em caso de falha no comando; e HREF="" para incluir o hyperlink interno ou externo conforme a necessidade.



SAIBA MAIS

O novo navegador Edge vem embutido com o Windows 10, tomando lugar do tradicional Internet Explorer que sofria com a falta de competitividade em termos de desempenho e recursos em relação a seus concorrentes diretos.

Firefox, Chrome, Safari e Opera estavam passos à frente do Internet Explorer tendo maior compatibilidade com alguns tipos de conteúdo web, além de terem praticamente abandonado qualquer suporte ao Silverlight (plugin para multimídia da Microsoft).

Este novo navegador tem suas características e qualidades atribuídas a evolução do antigo Internet Explorer para este novo Edge, e estas podem ser percebidas durante seu uso e em pesquisas na própria Internet.

Fonte: o autor.

Tabelas são recursos extremamente úteis para organizar conteúdos em páginas web, permitindo, como em softwares de planilhas eletrônicas e editores de texto, acrescentar, excluir, mesclar e dividir células livremente para obter as estruturas desejadas nestas tabelas.

A tag utilizada <TABLE> é aberta antes da indicação da primeira linha pela tag <TR> e da primeira coluna, indicada por <TD> que devem ser sempre fechadas para delimitar as células, e assim, pela sequência de tags para linhas e colunas, a tabela vai sendo estruturada.

Alguns parâmetros são aceitos pela tag <TABLE> como BORDER="" para indicar o tamanho das bordas das células, WIDTH e HEIGHT para definir comprimento total para a tabela que pode ser indicado em pixels ou porcentagem, como for desejado.

Para acertar linhas ou colunas, com quantidades diferentes de células, pode ser necessário usar as tags ROWSPAN="" para mesclar linhas e COLSPAN="" para mesclar colunas no interior de uma tabela.

O exemplo da figura 5 ajuda a compreender como se pode estruturar uma tabela com diversas linhas e colunas, tendo ou não bordas, células mescladas e medidas definidas.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Mensagem</title>
</head>
<body>

  <table border="1" style="width:100%">
    <tr>
      <td>Célula 1</td>
      <td>Célula 2</td>
      <td>Célula 3</td>
    </tr>
```

```

        <tr>
            <td>Célula 4</td>
            <td>Célula 5</td>
            <td>Célula 6</td>
        </tr>
    </table>
<br /> <br />

<table style="width:100%" border="2">
    <tr>
        <th>Produto</th>
        <th colspan="2">Modelo</th>
    </tr>
    <tr>
        <td>Item</td>
        <td>Modelo A</td>
        <td>Modelo B</td>
    </tr>
</table>
<br /> <br />

<table style="width:100%" border="3">
    <caption><strong>2015</strong></caption>
    <tr>
        <th>Semestre</th>
        <th>Vendas</th>
    </tr>
    <tr>
        <td>Primeiro</td>
        <td>R$ 150.000,00</td>
    </tr>

```

```
<tr>
    <td>Segundo</td>
    <td>R$110.000,00</td>
</tr>
</table>
</body>
</html>
```

Figura 5: Exemplo de código de criação de tabelas diversas em html
Fonte: o autor.

Existe a tag <DIV> para definir blocos que podem ser formatados, que contribuem para o chamado flat design, no qual blocos formatados são combinados de forma a criar um aspecto uniforme em páginas web.

Usando o parâmetro STYLE="", é possível definir diversos aspectos como cor de fundo, cor da fonte e tamanho do bloco. É preciso observar que toda vez que o parâmetro STYLE é utilizado em tags, sua formatação é diferenciada em termos de sintaxe e nomes de comandos.

```
<div style="background-color:black; color:white; padding:20px;">
    <h1><p align="center">Exemplo de Bloco</p></h1>
    <h2><p align="center">Fundo preto com letras brancas.</p></h2>
</div>

<div style="background-color:red; color:white; padding:20px;">
    <h1>Exemplo de Bloco</h1>
    <h2><p>Fundo vermelho com letras brancas</p></h2>
</div>
```

Figura 6: Exemplo de código html para uso de blocos.
Fonte: O autor.

O uso de frames foi muito comum até algum tempo, mas aos poucos a estruturação de páginas web vai sendo adaptada às necessidades de cada época e de cada onda evolutiva da própria Internet.

Com a tag <IFRAME> é possível criar sub-janelas capazes de exibir páginas externas ou internas ao website em construção, permitindo ajustes como largura, altura e bordas da janela dentro da página em si.

Na figura 7 temos um exemplo de código, demonstrando o uso de iframes para distribuir conteúdos em uma página web simples, apenas para ilustrar como podem ser incluídos diversos tipos de conteúdos, mas sabendo que excessos podem deixar o carregamento lento.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<title>Mensagem</title>
</head>
<body>
iframe name="iframe_1" width="100%" height="200" ></iframe>
iframe name="iframe_2" width="100%" height="100" ></iframe>
<iframe name="iframe_3" width="100%" height="100" ></iframe>
<h1>
  <p><a href="http://www.cesumar.br" target="iframe_1"
    style="border:none">Unicesumar</a>
    <a href="http://www.xbox.com/pt-BR/" target="iframe_2"
    style="border:none">Playstation</a>
    <a href="http://www.xbox.com/pt-BR/" target="iframe_3"
    style="border:none">Xbox</a></p>
</h1>
</body>
</html>
```

Figura 7: Exemplo de código usando iframes
Fonte: o autor.



REFLITA

A Microsoft disponibilizou junto com sua versão 10 do Windows, o navegador Edge que contém a base do plugin Flash Player em seu código. O abandono desta mesma tecnologia Flash por outros navegadores dividiu o mercado.

Fonte: O autor.

CONSIDERAÇÕES FINAIS

Caro(a) aluno(a), encerramos mais uma unidade. Compreendemos que a Internet é o maior depósito de conteúdos do mundo e, como a inserção destes é livre, temos material de toda qualidade incluído em seu gigantesco banco de dados. Desde pesquisas com fundamentação teórica e textos excelentes de autores conhecidos ou não, até material puramente plagiado, muitas vezes sem nenhum embasamento, proveniente apenas de conclusões pessoais.

Todo este material precisa ser alocado na rede mundial, mas esta tarefa não é feita simplesmente inserindo tais conteúdos, pois o que mantém a integridade da rede são inúmeros protocolos e ligações, físicas ou lógicas, que tem por base o uso de tecnologias diversas.

Para padronizar toda esta diversidade, uma das bases da Internet é a linguagem HTML e tudo aquilo que é possível agregar às páginas estruturadas a partir da linguagem, como plugins, *softwares* e linguagens complementares.

A linguagem HTML é antiga dentro da história da Internet e já teve algumas modificações para se adequar às necessidades de cada época, mantendo-se como base para programação *Front-End*.

Assim, foram tratados, nesta unidade, conceitos essenciais da linguagem HTML e como páginas completas de conteúdo podem ser implementadas, utilizando-se de exemplos simples e claros de como estruturar páginas web, contendo texto formatado, imagens e alguns outros tipos de conteúdo.

Também foram trabalhados conceitos mais avançados, utilizando tags mais complexas da linguagem, permitindo sua união com as tags mais simples, vistas inicialmente na unidade, para se chegar a um nível mais elevado de conhecimentos da linguagem.

Estes conceitos mais avançados permitem material de melhor qualidade na construção de páginas web e fornecem melhores formas de estruturar páginas atraentes e com melhor capacidade interativa.

Na sequência do livro, a unidade III tratará de avançar mais na linguagem HTML, introduzindo os conceitos da versão 5, tendo em mente, uma união entre esta linguagem moderna e a linguagem CSS, que contribui enormemente para a agilidade e qualidade do desenvolvimento de páginas web.

Bons estudos!

ATIVIDADES



1. Cite uma diferença entre o uso das tags `<HTML>` e `<BODY>`.
2. As tags para manipular texto podem possuir parâmetros que ajustam tamanho da fonte, cor e nome. Assinale a alternativa que contenha a tag relacionada a esta função.
 - a. P
 - b. A
 - c. TR
 - d. BR
 - e. Nenhuma das anteriores.
3. O uso de imagens permite que páginas web sejam mais atrativas e informativas. A sintaxe da tag `` permite diversas variações. Assinale a alternativa que contenha uma forma incorreta de estruturação desta tag.
 - a. ``
 - b. ``
 - c. ``
 - d. ``
 - e. ``
4. A tag `<USEMAP>` define que uma imagem pode ser subdividida em partes que podem ter links atribuídos a elas. Assinale a alternativa que contenha a tag responsável por indicar estes links.
 - a. MAP
 - b. AREA
 - c. HREF
 - d. IMG
 - e. TABLE
5. A codificação de páginas em html possui uma estrutura base comum em praticamente todas as páginas existentes. Elabore um exemplo de estrutura básica em html sem que esta precise exibir conteúdos na página, mas é preciso que a mesma tenha, pelo menos, definição de idioma, e título da aba no navegador.



DESENVOLVIMENTO PARA NUVEM NA PRÓPRIA NUVEM

Desenvolver conteúdo para web requer dedicação e pesquisa, de conhecimento técnico sobre a linguagem a ser utilizada para a tarefa, mas além disso, é preciso conhecer, também, a ferramenta que será utilizada para se desenvolver este material.

A partir daí, é interessante observar a evolução e as diferentes opções disponíveis na nuvem, para desenvolver código html ou de outras linguagens ligadas ao html como CSS e javascript.

Uma destas opções é o Eclipse Orion para java e outras linguagens web, contendo funções como auto complemento de código e realce de sintaxe. Seu uso não tem custos e sua interface é simples.

Outra opção é o eXo Cloud IDE que permite navegação em múltiplas janelas, além das características de auto complemento e realce do Orion. Possui também suporte, múltiplos ambientes, como ClouBee e Openshift, e não têm custos.

Cloud9 é uma alternativa com funções de gestão de projetos entre suas funcionalidades extras. Aceita implantação em ambiente Heroku, e não tem custos para projetos open source, mas cobra um valor mensal para projetos privados.

Integrar projetos criados em IDEs locais e na nuvem são um desafio em função de controle de versões e diferenças no funcionamento das IDEs, além da integração entre colaboradores que utilizam uma ou outra tecnologia para desenvolver seus trabalhos.

A evolução natural destas IDEs na nuvem é previsível e vai ocorrer naturalmente pela necessidade de se transferir atividades voltadas à Internet, para a própria nuvem, facilitando o desenvolvimento, testes e publicação.

Conhecer alternativas, para desenvolvimento, baseadas em softwares locais instalados na máquina ou remotos, que rodem pela nuvem, permitindo que a máquina rode mais “leve” com menos softwares instalados além do sistema operacional e outros essenciais.

Fonte: o autor.





LIVRO

Introdução à Programação Web com HTML, XHTML e CSS

Jon Duckett

Editora: Ciência Moderna

Sinopse: Este livro traz diversos conceitos e exemplos de como desenvolver páginas web utilizando HTML, XHTML e CSS com foco em um design atraente e prático, tendo também conceitos e previsões para futuras evoluções em função do que era utilizado na época.



HTML 5 e CSS 3



Objetivos de Aprendizagem

- Compreender as diferenças entre Html 5 e CSS 3.
- Complementar Html com CSS para produção de páginas web.

Plano de Estudo

A seguir, apresentam-se os tópicos que você estudará nesta unidade:

- Html 5 e CSS 3
- Estrutura Básica de Código HTML e CSS
- Comandos Essenciais
- Comandos Complementares

INTRODUÇÃO

Olá, caro(a) aluno(a), seja bem-vindo(a) a mais uma unidade de estudo de nosso material didático.

A linguagem CSS é um complemento muito útil para a linguagem html, pois permite a criação de regras de formatação, com o intuito de definir estilos para os elementos que compõem o html e que podem ser padronizados para uma página html ou todo um website.

Cada declaração em linguagem CSS indica uma parte da estrutura de um código html e toda vez que essa estrutura é utilizada durante a codificação de páginas, automaticamente, essas páginas seguem o padrão definido pelas regras CSS.

Isso pode parecer algo que reduz a efetividade do html, mas, na verdade, aumenta a capacidade da linguagem e, ainda, reduz o trabalho de codificação em quantidade de linhas e complexidade do código para eventuais manutenções e modificações.

Essa facilidade deve permitir que o programador html tenha mais tempo para elaborar melhor a estrutura das páginas que venha a criar, de forma a poder produzir melhores experiências de navegação para usuários e poder trabalhar melhor a forma como lidará com os conteúdos a serem dispostos em um *website*.

Páginas grandes, com uma quantidade maior de conteúdo, precisam ser bem elaboradas para gerarem uma melhor impressão no usuário e não fiquem confusas, ainda mais em *websites* que sejam baseados em um tema mais restrito e que possam facilmente ter sua formatação padronizada, como acontece em *websites* institucionais que representam alguma empresa que possua cores escolhidas como padrão.

Os chamados estilos CSS aceitam o uso de comandos semelhantes a algumas tags html, mas sua sintaxe é ligeiramente diferente, sendo preciso observar qual parte está desenvolvendo para evitar confundir as sintaxes.

Pode ser melhor codificar cada parte em etapas diferentes para evitar essa troca de sintaxe de comandos, permitindo ao programador se concentrar em uma determinada etapa e evitar uma maior quantidade de erros comuns durante a criação de programas em geral.

Vamos lá? Bons estudos.

HTML 5 E CSS 3



Figura 1: Ilustração do tema da unidade.

A linguagem HTML 5 trouxe inovações em relação às versões anteriores, incorporando tags com novas funções ou substituindo tags ultrapassadas por outras melhoradas. Além disso, a nova versão possui um conjunto muito eficiente de comandos para criação de conteúdos muito mais completos e elaborados para o desenvolvimento *Front-end*.

O quadro 1 indica um esboço de como uma página em formato HTML 5 pode ser dividida, um pouco diferente das versões anteriores da linguagem, proporcionando novos elementos semânticos para proporcionar formas diferentes de estruturar as páginas.

Além dos novos elementos estruturais, alguns elementos multimídia foram criados para que a linguagem fosse capaz de exibir conteúdo de áudio e vídeo sem a necessidade de plugins externos para exibir arquivos desse tipo.

Outros elementos adicionais permitiram evoluções em outros aspectos da produção de páginas web como atributos de controle para criação de formulários, que serão vistos nesta unidade, assim como a parte multimídia, para exemplificar o uso da notação em HTML 5.

CABEÇALHO DA PÁGINA (HEADER)	
LINKS DE NAVEGAÇÃO (NAV)	
SEÇÃO DA PÁGINA (SECTION)	CONTEÚDO ALÉM DO CONTEÚDO DA PÁGINA (ASIDE)
ARTIGO DA PÁGINA (ARTICLE)	
RODAPÉ DA PÁGINA (FOOTER)	

Quadro 1: Estrutura de uma página em HTML 5

Fonte: o autor.

Houve, nesta reformulação da linguagem, a eliminação de alguns elementos que passaram a ser contemplados por outra linguagem ou simplesmente substituídos por novas tags na nova versão, como <BASEFONT>, <BIG>, <CENTER>, e <STRIKE>, por exemplo, que migraram para a CSS, ou elementos como <FRAME>, <FRAMESET> e <NOFRAME>, que foram substituídas por <IFRAME>.

O quadro 2 contempla uma tabela em que estão listadas várias das novas tags da linguagem e uma breve descrição de suas funções para auxílio na codificação de páginas web, de acordo com a nova estruturação utilizada. Algumas dessas tags são pouco utilizadas, ao passo que outras trouxeram mudanças significativas na forma como se trata a codificação em html.

TAG HTML	DESCRIÇÃO
<article>	Espaço na página para algum conteúdo independente, como texto, formulário.
<aside>	Conteúdo relacionado com a página.
<command>	Botão, radiobutton ou checkbox para formulários.
<details>	Detalhamento de uma parte da página.
<summary>	Capitular ou sumário dentro da tag <details>.
<figure>	Legenda em uma imagem ou vídeo.
<figcaption>	Legenda da figura em uma seção.
<footer>	Rodapé de um documento ou seção, podendo incluir: data, autor, contatos e direitos autorais.
<header>	Usado para introduzir um documento ou seção e pode conter o elemento <nav> para inclusão de links de navegação.
<hgroup>	Cria títulos e subtítulos a partir de tags <h1> a <h6>, em que o maior é o título e os outros são subtítulos.
<mark>	Destacar textos.
<meter>	Medição. Usado quando existe valor máximo e mínimo conhecidos.
<nav>	Usado para uma seção de navegação por links (menu).
<progress>	Progresso de uma tarefa.
<section>	Seção de um documento, como capítulos, cabeçalhos, rodapés ou outras seções do documento.

<code><time></code>	Definição de tempo ou data.
<code><wbr></code>	Quebra de palavra. Para definir uma oportunidade de quebra de linha.

Quadro 2: Tabela de tags adicionais em HTML 5

Fonte: o autor.

A criação de formulários em HTML é uma prática bastante comum e influencia a dinamização de uma página web, permitindo a inserção de conteúdo por meio de campos para preenchimento.

O formulário é uma das possíveis formas de ligação entre o Front-end e o Back-end para tornar páginas web comuns em componentes de um sistema de banco de dados web.

Assim, é possível unir o HTML com linguagens como o PHP, capazes de acessar e manipular bancos de dados, ou linguagens como o javascript, que podem trabalhar com estruturas de dados mais elaboradas e realizar atividades mais complexas que o HTML.

Na figura 2, temos um exemplo básico da estrutura de um formulário em HTML que simplesmente cria um campo para preenchimento e um botão de ação para enviar o texto informado no campo.

```
<form>
  <p><label>Procurar <input type="search" name="query"><label></p>
  <p><button type="submit">Submeter</button></p>
</form>
```

Figura 2: Exemplo de código html para formulário

Fonte: o autor.

Existem diversas tags específicas para uso na criação de formulários, capazes de fornecer ferramentas suficientes para que formulários muito bons sejam produzidos, contendo diversos tipos diferentes de objetos para interação como listas, campos de preenchimento e opções de seleção.

Na quadro 3, temos uma listagem de algumas dessas tags para uso com formulários com suas respectivas descrições e sintaxes.

color	mostra uma paleta de cores.
date	mostra um calendário para escolha de meses e anos.
datetime	ajusta data e hora.
datetime-local	data e hora local.
email	navegador valida o valor de um e-mail válido estando ativada a validação.
month	mostra no campo opção para navegação entre meses.
number	valores "min" e "max" em intervalo de valores (step define valor para salto).
range	cria botão estilo deslizante.
search	usado para campos de pesquisa.
tel	utilizado entrada de números de telefones formatados.
time	usado para mostrar a hora do sistema.
url	cria campos para entrada de endereços de e-mail seguindo o padrão URL.
week	mostra campo formatado para escolha de dias de semana. A implementação desses elementos é feita usando a seguinte sintaxe: <code><input type="elemento" name="nome_elemento"/></code>

Quadro 3: Lista de tags incluídas na versão 5 para o campo <TYPE>

Fonte: o autor.

Na figura 3, é ilustrado um exemplo completo de código html que cria um formulário bastante completo, contendo diversos tipos de elementos para que seja possível compreender a forma de uso desses elementos.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="UTF-8859">
  <title>Figura 19</title>
</head>
<body>
<form action="pagina.php" method="post" target="_blank"
  accept-charset=" UTF-8859" enctype="application/x-www-form-
  -urlencoded" autocomplete="off" novalidate>
```

```

<p><button type="submit">Enviar</button></p>

<p><label>Procurar <input type="search" name="pesquisa">
</label></p>
<p><button type="submit"> Enviar </button></p>
<p><label>Horário do Evento <input type="" name="horario">
</label></p>
<p><button type="submit"> Enviar </button></p>
Lista de Itens
<select name="lista">
    <option value="Item 1" selected>Item 1</option>
    <option value="Item 2"> Item 2</option>
    <option value="Item 3"> Item 3</option>
    <option value="Item 4"> Item 4</option>
</select>
    <p><button type="submit"> Enviar </button></p>
    <p>Caixa de Texto<textarea name="mensagem" rows="10" col-
s="30">Digite seu texto aqui! </textarea></p>

<p>Escolha o navegador preferido<input list="navegador">
<datalist id="navegador">
    <option value="Microsoft Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
</datalist></p>

<p>Sexo<input type="radio" name="sexo" value="masculino"
checked>Masculino
<input type="radio" name="sexo" value="feminino">Feminino</p>
    <p><input type="checkbox" name="casado" value="Casado">
Casado <br />

```

```

<input type="checkbox" name="filhos" value="Filhos"> Tenho
filhos: <input type="number" name="filhos" min="0" max="99"
step="1" value="0"></p>

Data de Nascimento
<input type="date" name="nascimento" max="2015-01-01"><br>

Cor favorita<input type="color" name="cor"><br />

<fieldset>
    <legend>Informação</legend>
    <p><label>E-mail <input type="email" name="email" pla-
ceholder="usuário@domínio.com"></label></p>
    <p><label>Site <input type="url" name="site" placehol-
der="http://www.site.com/"></label></p>
    <p><label>Telefone <input type="tel" name="fone"
placeholder="99999-9999"></label></p>
</fieldset>

<fieldset>
    <legend>Login</legend>
    <p><label>Usuário <input type="text" name="usuario"
required></label></p>
    <p><label>Senha <input type="password" name="senha"
required> </label></p>
</fieldset>

<p><button type="submit" onclick="alert('Dados Enviados')">>
Enviar </button></p>

</form>
</body>
</html>

```

Figura 3: Exemplo de código HTML completo para uso de formulários

Fonte: o autor.

Temos na figura 3, um exemplo que serve como base para ilustrar a criação de um formulário completo contendo diversas tags muito comuns na codificação.

Analisando o código, temos, primeiramente, a tag <FORM> que inicia a construção do formulário e necessita ser fechada ao final, podendo ou não incluir diversos parâmetros para ajustar sua atuação.

```
<form action="pagina.php" method="post" target="_blank"
accept-charset=" UTF-8859" enctype="application/x-www-form-
-urlencoded" autocomplete="off" novalidate>
```

O parâmetro ACTION="PAGINA.PHP" indica que, quando submetido, o formulário enviará dados ao local indicado entre aspas, sendo, normalmente, um servidor PHP para tratamento dos dados ou um e-mail.

METHOD="POST" é outro parâmetro de <FORM> para definir o envio mais seguro dos dados sem exibição deles no endereço indicado na barra de endereços do navegador, ao passo que "GET" exhibe esses dados na barra e não se preocupa com o ocultamento de dados.

O parâmetro TARGET="_BLANK", também usado com a tag <A>, exibirá a próxima etapa do formulário quando submetido a uma nova aba ou janela do navegador, preservando a página contendo o formulário na janela ou aba atual. ACCEPT-CHARSET="UTF-8859" apenas define o conjunto de caracteres ASCII aceitos no preenchimento do formulário. ENCTYPE="APPLICATION/X-WWW-FORM-URLENCODED" define o método de encriptação de dados utilizado no envio de dados. AUTOCOMPLETE="OFF" desabilita o autopreenchimento do formulário, sendo "ON" valor padrão para esse parâmetro. Finalmente, NOVALIDATE indica que não haverá validação de campos, permitindo liberdade no preenchimento dos dados.

Outra tag importante na estruturação de um formulário é <FIELDSET>, que delimita um conjunto de campos, agrupando para posterior envio em conjunto por meio da tag <BUTTON>.

Já a tag <LEGEND> fará a inserção de um texto título para o conjunto de campos, permitindo melhor compreensão do agrupamento.

O parâmetro TYPE="SUBMIT" da tag <BUTTON> é responsável por enviar os dados dos campos agrupados ou não para o destino indicado na tag <FORM> pelo parâmetro ACTION.

ONCLICK é um parâmetro para indicar uma ação para um clique do mouse, podendo exibir informações em caixas de mensagem.

<p><button type="submit" onclick="alert('Dados Enviados')">> Enviar </button></p>

A tag <LABEL> é a tag essencial para criação dos campos do formulário e utiliza parâmetros como INPUT TYPE="SEARCH", que define o tipo do campo que será criado, e NAME="CAMPO", que indica um nome que é extremamente importante para o posterior tratamento dos dados por um código PHP, por exemplo.

Outra área bastante importante de aplicação da linguagem HTML é o desenvolvimento de formulários que possuem uma coleção particular de tags para uso na sua construção. No quadro 4, temos uma lista dessas tags e uma descrição de suas funções, tendo como exemplo prático de codificação da figura 3 anteriormente citada.

TAG	DESCRIÇÃO
BUTTON	Cria botões de ação em formulários
CHECKBOX	Cria opção do tipo SIM/NÃO, estando ou não em agrupamento
COLOR	Parâmetro de cor para campos
DATE	Cria campo do tipo data
DATETIME	Cria campo do tipo data e hora
DATETIME-LOCAL	Cria campo do tipo data, hora e local
EMAIL	Cria campo para entrada de e-mail
FILE	Cria campo para inclusão de arquivo
HIDDEN	Parâmetro para ocultamento de campo
IMAGE	Cria campo para inclusão de imagem
MONTH	Cria campo para entrada de mês
NUMBER	Cria campo para entrada de número

PASSWORD	Cria campo para entrada de senha usando caracter *
RADIO	Cria campo do tipo múltipla escolha em agrupamento
RANGE	Parâmetro que define limites para a digitação de valores
RESET	Parâmetro TYPE de botão para limpar o formulário
SEARCH	Cria campo de pesquisa
SUBMIT	Parâmetro TYPE de botão para enviar o formulário
TEL	Cria campo para inserção de número de telefone
TEXT	Cria campo de texto
TIME	Cria campo para entrada de hora
URL	Cria campo de entrada de endereço eletrônico (formato URL)
WEEK	Cria campo para entrada de dia da semana
DATALIST	Cria uma lista de opções para um campo
OPTION	Parâmetro para definição do tipo de campo de agrupamento
VALUE	Parâmetro para definição de valor padrão de um campo

Quadro 4: Listagem de tags para criação de formulários

Fonte: o autor.

Novas propriedades foram acrescentadas na versão HTML5 para estruturação de formulários. Na quadro 5, uma listagem de vários desses atributos com descrição permite formulários mais elaborados com mais funções.

Autocomplete	Habilita o auto complemento nos campos (on / off).
Novalidate	Ativa ou desativa validação de campos se ativado (on / off).
Autofocus	Foca o campo após página ser carregada pelo navegador.
Formaction	Substitui ACTION no formulário com URL válida para onde enviar os dados quando o formulário for submetido.
Formmethod	Substitui METHOD no formulário e define o método HTTP para o envio de dados para o URL usando GET, POST, PUT ou DELETE.
Formnovalidate	Substitui NOVALIDATE no formulário sendo TRUE para não validar.
Formtarget	Parâmetro para definir destino BLANK, SELF, _PARENT e TOP.

Width e Height	Largura e altura de campo do formulário (% ou px).
List	Referência a DATALIST onde um elemento com nome ID se associe a ID.
Multiple	Permite que o usuário selecione mais de um item em elemento.
Pattern	Cria restrições por expressões regulares: <input pattern="[0-9]{3}" name="digito" required title="Validação de Dados"/>.
Placeholder	Define ajuda para o usuário preencher campo com os dados corretos.
Step	Utilizado com tipos DATETIME, TIME, DATE, entre outros, para salto em intervalos numéricos.
Required	Dentro de um elemento INPUT torna o campo (validação ativada).

Quadro 5: Listagem de atributos implementados em html 5

Fonte: o autor.

Como esta disciplina tem foco na parte de Front-end, não será aprofundada a criação de formulários com acesso a bancos de dados, por exemplo, mas é possível unir formulários com codificação em javascript.

Na figura 4, um código completo em formato HTML 5 mostra como é estruturada uma página web com seus elementos semânticos que permitem a definição de áreas em uma página e melhor organizam seus conteúdos.

Esses elementos permitem a organização de sites e facilitam sua padronização em um website, podendo repetir alguns deles em todas as páginas, de forma facilitada em relação às versões anteriores de html.

```
<!DOCTYPE html>
<html lang="pt-br">
<title>Figura 22</title>
<meta charset="iso-8859-1">
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->
```

```
<style>
  body {
    font-family:Verdana,sans-serif;font-size:0.8em;
  }
  header,footer,section,article {
    border:2px solid red;
    margin:3px;margin-bottom:10px;padding:5px;
    background-color:yellow;
  }
  header,footer {
    color:black;background-color:gray;margin-bottom:3px;
  }
  section {
    background-color:blue;
  }
  nav ul {
    margin:0;padding:0;
  }
  nav ul li {
    display:inline; margin:3px;
  }
</style>
<body>

<header>
  <h1>Figura 21</h1>
</header>
```



```
<nav>
  <ul>
    <li>Home</li>
    <li>Unicesumar</li>
    <li>MEC</li>
  </ul>
</nav>

<section>
<h2>Exemplo de seção</h2>

<article>
  <h2>Primeiro Artigo</h2>
  <p>Conteúdo do Artigo</p>
</article>

<article>
  <h2>Segundo Artigo</h2>
  <p>Conteúdo do Artigo</p>
</article>

</section>

<footer>
  <p>Rodapé da Página</p>
</footer>

</body>
</html>
```

Figura 4: Exemplo de código HTML 5 com elementos semânticos.
Fonte: O autor.



REFLITA

Sendo a linguagem html usada para a criação de webpages, é realmente necessário esforço e tempo para o aprendizado de uma segunda linguagem?

Seria realmente uma vantagem a linguagem CSS criar seletores para formatar trechos de código HTML seguindo outra regra de codificação? Não seria melhor dedicar esse esforço realizado no aprendizado da nova linguagem para aperfeiçoar o HTML?

Fonte: o autor.

Nesse exemplo da figura 4, temos bem definidas as formatações para cada elemento semântico. Algumas dessas configurações podem ser aplicadas a grupos de elementos simultaneamente, como ocorre no ponto onde são definidos valores para atributos como header, footer, section e article ao mesmo tempo, fazendo com que tenham os mesmos valores para bordas, margens e cor de fundo. Como padrão em HTML 5, esses estilos definidos na tag `<STYLE>` têm uma sintaxe um pouco diferente, em que todos os atributos são inseridos entre `{ e }` e os valores desses são indicados após o símbolo `;`, ao invés do sinal de `=` padrão de atribuição na linguagem.

Outro ponto importante é a indicação de unidades de medida, como a abreviação de pixel indicada por “px”, “em” ou “rem” para o tamanho da fonte, sendo que cada caso tem suas particularidades e melhores situações para uso.

A inserção de itens de multimídia em páginas web é muito importante e permite que se possa agregar qualidade e conteúdo adicional, de forma a obter websites cada vez mais elaborados e com maior compatibilidade com as diversas alternativas de navegadores existentes.

Arquivos de vídeo dos tipos MPEG, AVI, WMV, MP4 e outros são aceitos, assim como arquivos de áudio como MP3 e WMA também podem ser exibidos pelo navegador apenas utilizando código HTML 5 sem adição de plug-ins.

```
<!DOCTYPE html>
<html>
<body>
  <video width="320" height="240" controls autoplay>
    <source src="video.mp4" type="video/mp4">
    <source src="video.ogv" type="video/ogg">
  </video>
</body>
</html>
```

Figura 5: Exemplo de código para exibição de vídeo
Fonte: o autor.

Nesse código da figura 5, temos a tag `<VIDEO>` que cria o elemento para exibição de vídeo com largura e altura definidos, além dos atributos `CONTROLS`, para exibir botões de controle da exibição, e `AUTOPLAY`, para que o vídeo inicie automaticamente.

A tag `<SOURCE>` indica o arquivo a ser exibido, sendo o atributo `TYPE` responsável por indicar o tipo de codec padrão da linguagem a ser utilizada na exibição, seguindo nomes padrões pré-estabelecidos para cada tipo de vídeo aceito.

No caso da reprodução de arquivos de áudio, a tag `<AUDIO>` funciona de forma semelhante, também utiliza os mesmos atributos para controlar a execução do som, tendo também predefinições para tipos de áudio aceitos. A figura 6 mostra um exemplo de uso desta tag `<AUDIO>`.

```
<!DOCTYPE html>
<html>
<body>
  <audio controls>
    <source src="som.ogg" type="audio/ogg">
    <source src="som.mp3" type="audio/mpeg">
  </audio>
</body>
</html>
```

Figura 6: Exemplo de código para exibição de áudio
Fonte: o autor.

ESTRUTURA BÁSICA DE CÓDIGO CSS

CSS é uma linguagem de estruturação de estilo para complementar a linguagem de criação de páginas em si, melhorando a velocidade de codificação e desenvolvimento de páginas, além de, cada vez mais, acrescentar recursos para evoluir visualmente a qualidade dos conteúdos desenvolvidos para web.

Na figura 7, temos um exemplo de como se unem html e CSS facilmente por meio da tag <STYLE>, que delimita todo o trecho do código onde são definidos os estilos CSS para formatação do conteúdo de uma página web.

```
<!DOCTYPE html>
<html>
<head>
<style>
  body {
    background-color: #d0e4fe;
  }

  h1 {
    color: orange;
    text-align: center;
  }

  p {
    font-family: "Times New Roman";
    font-size: 20px;
  }
</style>
</head>
<body>
  <p>Texto Exemplo</p>
</body>
</html>
```

Figura 7: Exemplo de codificação html com CSS

Fonte: o autor.

É possível observar, nesse exemplo da figura 7, que uma mudança na sintaxe dos comandos diferencia a linguagem html da CSS, de forma a facilitar a identificação de uma ou outra nos momentos em que sejam requeridas manutenções no código.

A tag <H1> ajusta automaticamente o tamanho de um texto a ser exibido de forma simples, permitindo valores de <H1> a <H6> para mudar o tamanho do texto, como já estudado. Podemos acrescentar opções extras dentro da tag usando CSS.

Diversos ajustes podem ser feitos utilizando comandos CSS para facilitar a codificação seguindo padrões, pois, ao configurar formatos com estilos CSS no início do código, depois, é só utilizar as tags ajustadas para que já estejam formatadas.

Em geral, os comandos CSS são chamados seletores e seguem uma sintaxe como a do exemplo a seguir:

```
SELETOR {PROPRIEDADE:VALOR; PROPRIEDADE:VALOR;}
```

Assim, podemos ter seletores que ajustam diversos elementos do código html, seguindo o formato como o da figura 8:

```
body {
    background-color: blue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: "Arial";
    font-size: 14px;
}
```

Figura 8: Exemplo de declaração de seletores CSS
Fonte: o autor.

Pode acontecer de mais de um selector precisar de configurações iguais para alguns atributos, para isso, podemos agrupar seletores para que todos tenham seus parâmetros configurados ao mesmo tempo. Observe o exemplo:

```
h1, h2, p {text-align: center; color: blue;}
```

É possível criar arquivos contendo as definições de todos os seletores desejados para que esse arquivo possa servir de base para um conjunto de páginas, sem que sejam necessárias repetições do estilo, economizando código e tempo, além de facilitar a padronização de um estilo em várias páginas.

Um arquivo contendo os seletores de estilo CSS deve ser gravado em um arquivo de texto com a extensão .CSS e deve ser adicionado a um código HTML por meio da tag <LINK>, dentro do cabeçalho <HEAD>, como no exemplo:

```
<head>  
  <link rel="stylesheet" type="text/css" href="estilo.css">  
</head>
```

Finalizando, é possível também utilizar um estilo CSS para configurar apenas uma tag html, tendo validade, apenas, enquanto a tag estiver ativa, como no exemplo a seguir:

```
<h1>style="color:blue;margin-left:30px;">Texto</h1>
```

COMANDOS ESSENCIAIS

Os seletores são muitos, assim como as tags html, mas, para facilitar, são semelhantes e, intuitivamente, podem ser lembrados, desde que pelo menos se conheça as tags html.

No quadro 6, temos os principais seletores CSS usados neste livro e, com eles, já será possível uma boa quantidade de configurações para códigos HTML.

background	Ajusta todas as propriedades de fundo de uma vez
background-color	Ajusta as propriedades da cor de fundo
background-image	Ajusta as propriedades da imagem de fundo
background-repeat	Ajusta as propriedades da repetição da imagem de fundo
background-attachment	Ajusta as propriedades da rolagem da imagem de fundo
background-position	Ajusta as propriedades da posição da imagem de fundo
border	Ajusta todas as propriedades de borda de uma vez
border-bottom	Ajusta as propriedades de borda de baixo de uma vez -color – ajusta a cor da borda -style – ajusta o tipo da linha de borda -width – ajusta a largura da linha de borda
border-top	Ajusta as propriedades de borda de cima de uma vez -color – ajusta a cor da borda -style – ajusta o tipo da linha de borda -width – ajusta a largura da linha de borda
border-left	Ajusta as propriedades de borda da esquerda de uma vez -color – ajusta a cor da borda -style – ajusta o tipo da linha de borda -width – ajusta a largura da linha de borda
border-right	Ajusta as propriedades de borda da direita de uma vez -color – ajusta a cor da borda -style – ajusta o tipo da linha de borda -width – ajusta a largura da linha de borda
margin	Ajusta todas as bordas do elemento de uma vez
margin-top	Ajusta as margens de cima do elemento
margin-right	Ajusta as margens da direita do elemento
margin-bottom	Ajusta as margens de baixo do elemento
margin-left	Ajusta as margens da esquerda do elemento
padding	Ajusta o espaçamento entre borda e conteúdo
padding-top	Ajusta o espaçamento entre borda de cima e conteúdo
padding-right	Ajusta o espaçamento entre borda da direita e conteúdo

padding-bottom	Ajusta o espaçamento entre borda de baixo e conteúdo
padding-left	Ajusta espaçamento entre borda da esquerda e conteúdo
color	Ajusta a cor do texto de um elemento (texto, tabela, etc.)
text-align	Ajusta o espaçamento horizontal do texto de um elemento
text-decoration	Permite destacar textos com linhas usando as opções: underline: linha sobre; line-through: linha riscando o texto; underline: linha abaixo do texto
text-transform	Modifica os caracteres do texto usando as opções: uppercase: maiúsculas; lowercase: minúsculas; capitalize: letra capital
line-height	Ajusta o espaçamento entre linhas de texto
text-shadow	Adiciona o efeito de sombra a um texto
font-family	Permite a escolha de fontes para texto
font-style	Ajusta efeitos entre texto normal ou itálico (itálico)
font-size	Ajusta o tamanho da fonte em px, em, rem, %
font-weight	Ajusta efeitos entre texto normal ou bold (negrito)
a:link	Comando para configurar estilos para links não clicados
a:visited	Comando para configurar estilos para links já clicados
a:hover	Comando para configurar estilos quando mouse em link
a:active	Comando para configurar estilos quando link é clicado
list-style-type	Seletor para criar tópicos em listas usando opções como: square (quadrado), circle (círculo), upper-roman (números romanos), lower-alpha (letras)
list-style-image	Utiliza imagens gravadas como marcadores em listas

Quadro 6: Listagem de seletores e parâmetros CSS essenciais

Fonte: o autor.

No exemplo da figura 9, temos um código HTML com CSS completo utilizando links para ajustar a cor do texto do link de acordo com este já ter sido clicado ou não, além da ação do mouse sobre ele. Observe que a página destino deve conter um endereço ou arquivo HTML válido.


```
<!DOCTYPE html>
<html>
<head>
<style>
  p {
    font-family: arial;
    font-size: 14px;
  }
  a:link {
    color: blue;
  }

  a:visited {
    color: red;
  }

  a:hover {
    color: black;
  }

  a:active {
    color: yellow;
  }
</style>
</head>
<body>
<p><b><a href="pagina.html" target="_blank">Observe como
  mudam as cores deste link</a></b></p>
</body>
</html>
```

Figura 9: Exemplo de código HTML e CSS com links
Fonte: o autor.



SAIBA MAIS

Ter diferentes medidas para ajustar tamanho de fonte é útil?

Imagine um site mobile ou um site para telas maiores ou projeções. Ao invés de você mudar as fontes para cada elemento, é possível ajustar diretamente o tamanho da fonte padrão para cada tag HTML de texto.

```
body {font: 100% verdana, arial, sans-serif;}  
h1{  
    font-size: 2.50em;  
}  
h2{  
    font-size: 1.50em;  
}  
p{  
    font-size: 0.8em;  
}
```

Figura 10: Exemplo de código para ajuste de tamanho da fonte
Fonte: o autor.

Mudando o valor de porcentagem da FONT do elemento BODY, proporcionalmente, a fonte e todo conteúdo padrão da página são ajustados.

SAIBA MAIS



Outra unidade chamada REM também pode ser utilizada para referenciar tamanho de fonte e funciona de maneira proporcional a medidas em pixels. A figura 11 mostra um exemplo de código utilizando essa unidade de medida.

```
body {font: 100% verdana, arial,sans-serif;}
h1 {
  font-size: 3.00rem;
}
h2 {
  font-size: 2.00rem;
}
p {
  font-size: 1.50rem;
}
```

Figura 11: Exemplo de código para ajuste de tamanho da fonte
Fonte: o autor.

Alguns navegadores podem ter problemas com essas unidades, mas isso ocorre mais com navegadores antigos, em que é interessante o uso de PX ou porcentagem mesmo como unidade padrão.

COMANDOS COMPLEMENTARES

Aos poucos, é possível perceber como existe uma quantidade significativa de tags, e essas vão sendo incrementadas até que tenhamos uma quantidade razoável de opções para criação de bons conteúdos para web.

Observe o quadro 7 abaixo para conhecer propriedades diversas de formatação de bordas e outras características relacionadas à construção de tabelas e caixas div.

border	Ajusta bordas para tags <table>, <th> e <td> em tabelas
border-collapse	Esse seletor permite deixar a borda simples ('collapse')
width	Seletor para largura de elementos (tabela, div - % ou px)
height	Seletor para altura de elementos (tabela, div - % ou px)
vertical-align	Assim como text-align, ajusta posição do texto (vertical)
padding	Seletor para ajuste da distância entre borda e texto
outline	Seletor para estilo, largura e cor de linha ao redor de div
outline-style	Define estilo de linha ao redor de um box (div), usando dotted (pontilhada), dashed (tracejada), solid (simples), double (dupla), groove, ridge, inset, outset (bordas 3d), none (nenhuma), hidden (oculta).
outline-width	Seletor para a largura da linha ao redor da div
outline-color	Seletor para a cor da linha ao redor da div
max-width	Seletor para definir limite máximo para largura de um div
max-height	Seletor para definir limite máximo para altura de um div
min-width	Seletor para definir limite mínimo para largura de um div
min-height	Seletor para definir limite mínimo para altura de um div
z-index	Seletor para posicionamento de objetos em camadas (-1)
float	Serve para posicionar elementos em relação a outros
clear	Seletor para anular o efeito do seletor float
clearfix	Classe que pode ser usada para ajustar elementos de forma a se organizarem com o parâmetro overflow: auto

floating-box	Caixas para alocar elementos soltos de tamanho variado
after-box	Limpa ajustes feitos por floating-box (clear: left;)
margin	Seletor para posicionamento de elementos (margin: auto;)
position	Seletor para posicionamento definido por parâmetros: position: absolute; para dimensões e posição exatas position: relative; para ajustes proporcionais a janela
opacity	Propriedade de transparência entre 0 e 1 (opacity: 0.4;)

Quadro 7: Listagem de seletores e parâmetros CSS complementares

Fonte: o autor.

É possível criar seletores combinados para casos específicos, em que eles estejam aninhados, como no exemplo da figura 13 utilizando propriedades do quadro 7, em que temos uma tag <TABLE> dentro de uma tag <DIV> para criação de um parágrafo dentro de uma caixa.

```
<!DOCTYPE html>
<html>
<head>
<style>
  div {
    width: 75%;
  }
  table {
    border-collapse: collapse;
    width: 50%;
  }

  table, td, th {
    border: 3px solid black;
  }
</style>
</head>
```

```

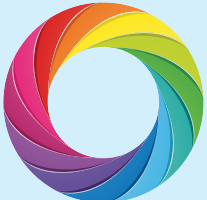
<body>
<div width="50%"><table>
  <tr>
    <th>TAG</th>
    <th>DESCRIÇÃO</th>
  </tr>
  <tr>
    <td>TR</td>
    <td>Inicia nova linha</td>
  </tr>
  <tr>
    <td>TD</td>
    <td>Inicia nova coluna</td>
  </tr>
</table></div>
</body>
</html>

```

Figura 13: Exemplo de código HTML com CSS para tabelas e divs
Fonte: o autor.

Complementando as tags listadas até o momento na figura 1, temos a seguir uma nova etapa na qual são introduzidas as mudanças implementadas para a versão CSS 3, que agregou uma grande quantidade de seletores pra melhorar o uso de imagens, animações e transformações em 2D e 3D, efeitos de texto, caixas e opções de layout no quadro 8.

border-radius	Propriedade para arredondamento de cantos pela ordem: sup. esquerdo, sup. direito, inf. esquerdo, inf. direito
border-image	Propriedade para uso de imagem para compor borda
background-image	Permite múltiplas imagens como fundo em camadas
background-size	Propriedade para ajustar o tamanho do plano de fundo (background-size: contain; ou background-size: cover;)

background-origin	Propriedade para especificar a posição do plano de fundo border-box: Plano de fundo inicia no canto sup. esquerdo padding-box: Plano de fundo inicia dentro da borda content-box: Plano de fundo inicia junto com o texto		
background-clip	Propriedade para especificar a posição da cor de fundo border-box: Cor de fundo inicia no canto sup. esquerdo padding-box: Cor de fundo inicia dentro da borda content-box: Cor de fundo inicia junto com o texto		
rgba	Define cores e transparência (cor, % saturação, % brilho)		
	315°	360° / 0°	45°
	270°		90°
	225°	180°	135°
hsla	Ajustes de cor, saturação, luminosidade e transparência em valores de 0 a 360, 0% a 100%, 0% a 100% e 0 a 1		
linear-gradient	Propriedade de ajuste de background em gradiente linear Direção: left-top, to bottom right, to right, to left, etc (0° a 360°), Cor inicial, Cor final		
radial-gradient	Propriedade de ajuste de background em gradiente radial Forma: circle, closest-side, farthest-side, closest-corner, farthest-corner, cores (nome) e porcentagens (%)		
text-shadow	Propriedade que cria sombra em texto com seus atributos Distância horizontal, Distância vertical, efeito borrar, cor		
box-shadow	Cria uma sombra em caixas div (tamanho, borrar, cor)		
transform	Efeitos 2D em elementos a partir de parâmetros diversos: translate() move o element horizontal e verticalmente (px) rotate() gira um elemento horário/anti-horário (deg ou -deg) scale() aumenta ou reduz o tamanho de um element (valor) skewX() inclina um element de acordo com o ângulo (deg) skewY() inclina um element de acordo com o ângulo (deg) matrix() agrupa todos os métodos acima em valores		

transform	Efeitos 2D em elementos a partir de parâmetros diversos: rotateX() gira sobre o eixo central X o elemento (deg) rotateY() gira sobre o eixo central Y o elemento (deg) rotateZ() gira sobre o eixo central Z o elemento (deg) translate3d() define um movimento em 3D (x,y,z) scale3d() aumenta ou reduz em 3D (x,y) rotate3D gira o elemento (x,y,z,ângulo) perspective() visão perspectiva de um elemento (n)
transition	Ajuste de parâmetros para controlar efeitos de animação transition: width 2s, height 4s; (modifica largura e altura)
transition-timing-function	Ajustes no tempo de execução dos efeitos ease: início lento, acelerando e reduzindo no final linear: do início ao fim com a mesma velocidade ease-in: início lento da transição ease-out: final lento da transição cubic-bezier(n,n,n,n): ajustes na curva de transição
transition-delay	Ajusta um tempo de espera para início da transição
column-count	Divide o conteúdo de um elemento em colunas (valor)
column-gap	Define a distância entre as colunas definidas (px)
column-rule-style	Define o estilo da linha divisória entre colunas (solid)
column-rule-width	Define a espessura da linha divisória entre colunas (px)
column-rule-color	Define a cor da linha divisória entre colunas (nome)
column-rule	Agrupa as definições para divisão de elementos em colunas
column-span	Mescla colunas divididas (column-span: all;)
column-width	Especifica a largura preferencial para colunas (px)
filter	Seletor para ajuste de filtros para tratamento de imagens. None: retira efeitos ativados blur(): Embaça a imagem (%) brightness(): Brilho da imagem (%) contrast(): Ajusta contraste (%) drop-shadow(): Sombra na imagem (Horiz, Vert, blur, cor) grayscale(): Converte a imagem em tons de cinza (%) hue-rotate(): Matiz da imagem (deg)

filter	invert(): Inverte as cores da imagem no estilo negativo (%) opacity(): Transparência da imagem (%) saturate(): Saturação da imagem (%) sepia(): Converte a imagem no estilo envelhecida (%)
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Quadro 8: Listagem de seletores e parâmetros adicionais para linguagem CSS 3

Fonte: o autor.

O exemplo da figura 14 demonstra algumas das propriedades descritas no quadro 8, no qual temos as propriedades específicas da linguagem CSS 3, em que efeitos mais elaborados de animação foram agregados, tornando essa versão muito poderosa.

```

<!DOCTYPE html>
<html>
<head>
<style>
  div {
    width: 200px;
    height: 200px;
    background-color: red;
    position: relative;
    /* Opções para maior compatibilidade com navegadores*/
    -webkit-animation-name: quadrado;
    -webkit-animation-duration: 3s;
    -webkit-animation-iteration-count: infinite;
    animation-name: quadrado;
    animation-duration: 3s;
    animation-iteration-count: infinite;
  }

  /* Navegadores Chrome, Safari, Opera */
  @-webkit-keyframes quadrado{
    0% {background-color:red; left:0px; top:0px;}
    25% {background-color:black; left:200px; top:0px;}
    50% {background-color:red; left:200px; top:200px;}
  }

```

```
75% {background-color:black; left:0px; top:200px;}
100% {background-color:red; left:0px; top:0px;}
}

@keyframes quadrado{
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:black; left:200px; top:0px;}
  50% {background-color:red; left:200px; top:200px;}
  75% {background-color:black; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
  div</div>
</body>
</html>
```

Figura 14: Código HTML 5 com CSS 3 para criação de efeitos
Fonte: o autor.

Observe que algumas linhas foram incluídas para aumentar a compatibilidade com os diversos navegadores mais populares disponíveis e suas particularidades, mas, em casos como de versões mais antigas, como aquelas do Internet Explorer anteriores a versão 9, podem não funcionar como esperado pela falta de suporte aos recursos.

Finalizando, lembre-se de que todos os exemplos podem ser customizados livremente para melhor aprendizado, além de novas tags e propriedades acrescentadas para criar páginas com mais recursos visuais, contribuindo com um melhor aprendizado.

SAIBA MAIS



HTML5 introduziu um método para permitir que web designers possam assumir o controle sobre a janela de exibição, por meio do <meta> tag.

Você deve incluir o elemento <meta> dentro da área de cabeçalho <HEAD> para indicar as propriedades de exibição em todas as suas páginas web:

```
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

Um <meta> elemento viewport dá as instruções do navegador sobre como controlar dimensões e escalonamento da página.

A largura = largura do dispositivo parte define a largura da página a seguir à tela de largura do dispositivo (que irá variar dependendo do dispositivo).

A escala inicial = 1,0 parte define o nível de zoom inicial, quando a página é carregada pela primeira vez pelo browser.

Fonte: o autor.

CONSIDERAÇÕES FINAIS

Caro(a) aluno(a), nesta unidade, foi possível perceber como existe uma quantidade significativa de comandos para a estruturação de páginas web oferecida pela linguagem HTML.

Assim, com as evoluções proporcionadas pela versão 5 da linguagem HTML, ela está cada vez mais independente de softwares adicionais, como plugins para exibição de conteúdo multimídia.

Podemos observar com exemplos práticos de códigos que podem ser testados, alterados e aprimorados o quão criativas podem ser as páginas web criadas e como podem resultar em uma ótima ferramenta de produção de conteúdo para Internet.

Com a adição da linguagem CSS aos códigos HTML, a qualidade visual e a facilidade de codificação aumentaram e permitiram aos desenvolvedores utilizar uma maior fatia de tempo e esforço de trabalho no processo criativo e desenvolvimento de conteúdo, reduzindo o tempo e o esforço em codificação.

Assim como a linguagem HTML que, em sua versão 5, trouxe acréscimos muito significativos, a linguagem CSS, em sua terceira versão, permite uma maior quantidade de efeitos, como planos de fundo contendo imagens e animações com elementos diversos.

É importante perceber que, mesmo tendo uma grande quantidade de comandos disponíveis nas duas linguagens, algumas regras são essenciais e devem ser respeitadas. Isso serve para melhorar a qualidade e a compatibilidade entre um código criado e uma das opções de navegadores disponível.

Regras como fechamento de todas as tags abertas, padronização do código em letras minúsculas e inclusão de todos os elementos essenciais ao código são características que melhoram a compatibilidade, além de serem alguns requisitos da versão HTML 5.

A união entre HTML e CSS tem ficado mais forte com o passar do tempo e à medida que são implementadas novas versões, é certo que essa união tende a ser duradoura, a menos que surja alguma revolução no desenvolvimento da Internet e que essas linguagens não sejam capazes de acompanhar.

Na próxima unidade, será tratada a linguagem javascript e uma das bibliotecas mais conhecidas para facilitar enormemente a codificação de páginas mais complexas, contendo, por exemplo, validações em formulários.

ATIVIDADES



1. Temos uma série de tags na linguagem html, mas algumas são obrigatórias. Assinale a alternativa que contenha apenas tags obrigatórias em códigos html 5.
 - a. `
`, `<html>`, `<body>`.
 - b. `<html>`, `<body>`, `<!DOCTYPE html>`.
 - c. `<body>`, `<p>`, `<a>`.
 - d. `<html>`, `<p>`, `<audio>`.
 - e. `<a>`, `<p>`, `
`.

2. A linguagem CSS contém seletores que facilitam a configuração de formatos e efeitos em códigos de páginas web. As afirmações a seguir citam propriedades válidas ou não em linguagem CSS. Assinale a alternativa que contenha apenas afirmações corretas:
 - I. A propriedade `width` ajusta largura de elementos.
 - II. A propriedade `background` define imagens de borda para elementos.
 - III. A propriedade `border` exibe bordas com espessura e cor escolhidas.
 - IV. A propriedade `text-align` define o alinhamento de imagens JPG.Assinale a alternativa correta:
 - a. Apenas I e II estão corretas.
 - b. Apenas II e III estão corretas.
 - c. Apenas I e III estão corretas.
 - d. Apenas II, III e IV estão corretas.
 - e. Nenhuma das alternativas está correta.

3. Elementos semânticos indicam áreas em uma página web e podem ser configuradas pelos estilos CSS. Em relação às afirmações indicando elementos semânticos, assinale Verdadeiro (V) ou Falso (F) para cada elemento:
 - () `body`.
 - () `nav`.
 - () `header`.
 - () `feet`.
 - () `social`.

ATIVIDADES



4. Supondo que numa página web tenha que ser acrescentada uma música gravada num arquivo chamado "MUSICA.MP3". Como poderia ser um código html 5 para executar automaticamente essa música nessa página?

5. Uma página precisa que seja ajustada sua fonte, tamanho e cor para ARIAL, 20px e AZUL, respectivamente, para o corpo da página como padrão. Elabore um exemplo de código capaz de ajustar esses parâmetros.



Existe a possibilidade de criar gráficos a partir de tags HTML 5 utilizando os elementos gráficos <CANVAS> ou <SVG> de forma bem simples. Nesta seção, será tratado o uso apenas do elemento SVG por ser mais compatível e simples de codificar.

A figura 15 mostra um exemplo completo de código que utiliza tags como <CIRCLE>, <RECT>, <POLIGON> ou <ELLIPSE> para criação de formas a partir de atributos que podem ser ajustados para construir figuras com cores e tamanhos variados.

```
<!DOCTYPE html>
<html>
<body>
<h1 align="center">Exemplos SVG - Scalable Vector Graphics</h1>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
    fill="yellow" />
</svg> <br/>
<svg width="100" height="100">
  <rect x="10" y="10" rx="20" ry="20" width="80" height="80"
    style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg> <br/>
<svg width="100" height="100">
  <polygon points="0,50 50,0 100,50 50,100" style="fill:lime" />
</svg> <br/>
<svg height="100" width="100">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="50" cy="50" rx="50" ry="30" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="22" font-family="Arial" x="20" y="60">HTML</text>
</svg>
</body>
</html>
```

Figura 15: Código completo para criação de gráficos com SVG

Fonte: o autor.





Em algumas tags, os atributos necessários para a criação de elementos gráficos são simples, mas, na tag <POLYGON>, por exemplo, são necessários vários parâmetros que indicam coordenadas dentro do espaço criado pela tag <SVG>.

Essas coordenadas devem ser adequadas ao tamanho da caixa definida em pixels pela tag <SVG>, sempre imaginando seu posicionamento dentro dessa área delimitada.

Outra tag complexa é <LINEAR GRADIENT> que utiliza parâmetros para indicar os pontos da figura onde ocorrem as passagens de cor, as cores de cada parte e de que forma o efeito ocorre na passagem entre as cores.



LIVRO

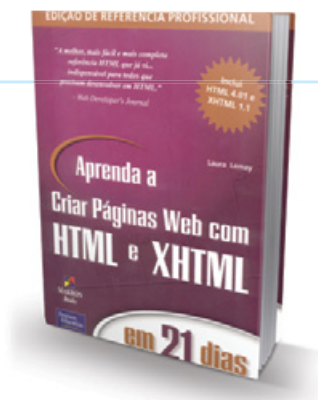
Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias

Laura Lemay

Editora: Pearson Education do Brasil

Sinopse: Elaborado para tratar de assuntos sobre o desenvolvimento para Web, esse material simples de fácil compreensão permite o aprendizado básico sobre HTML e XHTML para posteriores avançados na criação de elementos como tabelas, frames, JavaScript, e Folhas de Estilo CSS.

Comentário: Literatura com boa distribuição de conteúdos e capaz de auxiliar iniciantes e programadores mais experientes na compreensão das linguagens mais populares para desenvolvimento de páginas web.



JAVASCRIPT E JQUERY

UNIDADE

IV

Objetivos de Aprendizagem

- Conceituar o uso da linguagem Javascript e da biblioteca JQuery.
- Descrever a ideia da linguagem e suas funcionalidades básicas.
- Compreender a forma como o Javascript se liga a códigos Html e CSS.
- Conceitar a biblioteca JQuery e como prepará-la para uso.
- Elencar principais comandos JQuery com sintaxe e exemplos de uso.

Plano de Estudo

A seguir, apresentam-se os tópicos que você estudará nesta unidade:

- Javascript e sua codificação
- Html, CSS e Javascript
- Aquisição e Instalação da JQuery
- Criação de Scripts

INTRODUÇÃO

Olá, caro(a) aluno(a), tendo visto a linguagem HTML, e depois a CSS, percebemos que ocorreu uma evolução em termos de sintaxe dos comandos utilizados em cada uma, tendo como base que a HTML é mais antiga e segue um padrão mais formal que a CSS – que é mais aberta e permite maior liberdade.

A linguagem Javascript já parte para uma evolução maior ainda em termos de liberdade de codificação, além de ser mais complexa e capaz de realizar tarefas mais elaboradas, complementando as outras duas linguagens estudadas. Permite inúmeras outras estruturas de programação, como laços condicionais e de repetição.

Assim, uma página web que contenha código das três linguagens em conjunto é capaz de proporcionar uma experiência plena ao usuário em termos de navegação interativa e completa.

O Javascript permite que programação de alto nível seja embutida em códigos HTML de forma que esta união seja muito mais eficiente que a codificação HTML pura – mais limitada, não tendo tags suficientes para trabalhar com variáveis, por exemplo.

Assim, a evolução da linguagem Javascript fez com que ela, junto com a linguagem Java, ganhasse muito espaço no mercado e permitisse evoluções em conjunto com as sofridas por dispositivos móveis.

Uma linguagem como esta é capaz de criar verdadeiros softwares embutidos em páginas web sem que processos mais complexos tenham que ocorrer antes de sua efetiva utilização em navegadores, como compilações ou instalação de softwares adicionais – os chamados “plugins”.

Na verdade, o uso de plugins está sendo reduzido, devido às evoluções das linguagens para web e, por isso, uma tendência que vem se apresentando é a de que as linguagens se tornem cada vez mais poderosas e capazes de suprir as necessidades, num vislumbre possível do futuro. Pode ser também que os plugins se transformem em aliados capazes de oferecer recursos muito complexos para criação de websites avançados.



©shutterstock

Figura 1: Ilustração para o estudo de javascript.

Reprodução proibida. Art. 184 do Código Penal e Lei 9.610 de 19 de fevereiro de 1998.

JAVASCRIPT E SUA CODIFICAÇÃO

Basicamente, a linguagem Javascript permite que páginas HTML possam ser dinamicamente alteradas durante visualização em navegadores, podendo conter códigos mais elaborados, capazes de trabalhar com elementos não suportados pelas linguagens HTML ou CSS, devido a existência de limitações para área de atuação.

Temos que a web pode ser dividida em camadas: sendo a primeira mais superficial e visível ao usuário, a qual é chamada de Conteúdo, onde a linguagem HTML é fundamental em seu desenvolvimento; temos, também, uma camada de Apresentação, onde a linguagem CSS define toda a estilização da aparência que é utilizada pela camada superior na exibição de seu conteúdo; há também uma terceira camada, chamada de Comportamento, essa é mais interativa e dinâmica, capaz de modificar totalmente a usabilidade de uma página web com scripts que podem ser incorporados diretamente no código HTML ou gravados em arquivo avulso, e anexados pelo código HTML quando necessário.

Para a área de desenvolvimento *Front-end*, não é necessário muito aprofundamento na programação em si, pelo fato de ser uma área do desenvolvimento web que lida com aquilo que tenha relação com a experiência de navegação do usuário. Por outro lado, a programação *Back-end* é mais focada com processos que ocorrem internamente em websites, como gerenciamento e acesso a banco de dados, comércio online, criação de blogs e aplicativos.

Alguns conceitos básicos da programação Javascript envolvem uso de palavras-chave, como nas demais linguagens de programação, além de operadores, tipos de dados, laços, estruturas de dados e funções.

Este nível de programação pode ser estruturado ou orientado a objetos e é tratado em outras disciplinas que envolvam linguagens como JAVA ou PHP, mas, de qualquer forma, alguns conceitos básicos necessários serão tratados.

As principais palavras-chave utilizadas em Javascript podem ser vistas no quadro 1, onde é possível perceber que são poucas, mas suficientes para que cumpra sua função. Um detalhe é que todas devem ser digitadas em minúsculas, pois Javascript é *case-sensitive*, ou seja, diferencia maiúsculas de minúsculas.

PALAVRA-CHAVE	DESCRIÇÃO
break	Encerra a execução de um laço.
Continue	Reinicia um laço, retornando ao seu início.
debugger	Interrompe a execução de um código Javascript para verificação
do ... while	Laço de repetição que executa ao menos uma iteração para depois verificar a condição imposta em WHILE.
for	Laço de repetição com uma quantidade definida de iterações controlado por uma variável de controle.
function	Declaração de um função.
if ... else	Comando de decisão que pode ter mais de uma alternativa de execução dependendo das condições IF.
return	Comando de retorno do resultado de uma função.
switch	Comando de decisão onde são analisados possíveis valores de uma variável para escolha dos comandos a executar.
try ... catch	Comandos de tratamento de erros em blocos de comandos.
var	Define a declaração de uma variável.

Quadro 1: Principais palavras-chave da linguagem Javascript.

Fonte: O autor

Temos também operadores matemáticos e relacionais para realizar diversas funções dentro da codificação Javascript, com atenção para os operadores `===` e `!==` que, além de verificar se valores são iguais em uma comparação, verificam se seus tipos de dados também são.

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
++	Incremento
--	Decremento
==	Igualdade
===	Igualdade de valor e tipo de dado
!=	Diferente
!==	Diferente valor ou tipo de dado
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Quadro 2: Operadores da linguagem Javascript.
Fonte: O autor

Bem diferente da forma como é estruturado o código em HTML ou CSS, o Javascript tem codificação similar a outra linguagem de programação conhecida, chamada C, que inclusive serve de base para quase todas as demais linguagens.

Inicialmente temos o conceito de variável, que em Javascript é muito simplificado – e por isso fácil de compreender –, pois é uma linguagem com fraca tipagem, ou seja, não é preciso que cada variável seja rigidamente controlada e nem ter um tipo definido em sua declaração.

Assim, podemos observar na figura 2 um exemplo simples de código onde três variáveis são criadas para receberem, 10, 20 e 10+20, respectivamente, pois o valor de X e Y são somados para gerar o valor de Z.

Depois, para mostrar a fraca tipagem, o valor de X é trocado para texto sem problemas, mostrando como é flexível o uso de variáveis em Javascript, justificando, assim, a inclusão dos operadores que comparam valores e tipos === e !==.


```
<!DOCTYPE html>
<html>
<body>
<p>X = 10 e Y = 20 Z = X+Y</p>
<p id="identificador"></p>
<p>Mudando o valor de X</p>
<p id="identificador2"></p>

<script>
    var x = 10;
    var y = 20;
    var z = x + y;
    document.getElementById("identificador").innerHTML = z;
    x = "Texto"
    document.getElementById("identificador2").innerHTML = x;
</script>

</body>
</html>
```

Figura 2: Exemplo de código HTML com Javascript.

Fonte: O autor

HTML, CSS E JAVASCRIPT

Para entender como ocorre a fusão de comandos entre as 3 linguagens, o exemplo de código da figura 3 ajuda a compreender como os códigos referentes a cada linguagem são delimitados.

Observe que a linguagem HTML engloba as demais: tendo a tag <HTML> indicativa dos limites de atuação do código HTML na página; dentro dela, o código CSS embutido na tag <STYLE> e Javascript dentro da tag <SCRIPT>.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  font-family : Times New Roman;
  font-size : 16px;
  color: red;
  text-align: center;
}
</style>

<script>
  function funcao() {
    document.getElementById("texto").innerHTML = "Conteúdo depois
    do clique!";
  }
</script>

<title>TROCA DE TEXTO</title>
</head>

<body>
  <p id="texto">Conteúdo antes do clique!</p>
  <p><button type="button" onclick="funcao()"> Clique! < button></p>
</body>
</html>
```

Figura 3: Código completo contendo HTML, CSS e Javascript.

Fonte: O autor.

Neste exemplo, temos a configuração de exibição de textos em parágrafos definidos pela tag <P> na área de estilos CSS <STYLE>. Temos também um pequeno código em Javascript para criar uma função que troca o texto exibido logo no início do conteúdo da página, após a tag <BODY>.

Esta função chama um comando padrão de exibição de texto em Javascript `document.getElementById("texto").innerHTML` que, em função do texto inicial ter sido identificado por `id="texto"`, faz a ligação com o comando de captura de elemento `getElementById`.

Com isso, temos um exemplo simples, mas completo, de união entre HTML, CSS e Javascript, mostrando como a interação entre as linguagens ocorre de maneira tranquila, bastando apenas prática para compreender a melhor forma delas serem integradas.

Neste exemplo da figura 4, temos um código que faz a troca de imagens com nomes fictícios “inicial.jpg” e “final.jpg”. Essas devem ter seus nomes trocados por nomes reais de imagens que estejam gravadas na mesma pasta onde está o código.

```
<!DOCTYPE html>
<html>
<body>
  <h1>TROCA DE IMAGENS</h1>
  
  <p>CLIQUE PARA TROCAR A IMAGEM</p>

  <script>
    function troca() {
      var figura = document.getElementById('imagem');
      if (figura.src.match("inicial")) {
        figura.src = "final.jpg";
      } else {
        figura.src = "inicial.jpg";
      }
    }
  </script>
</body>
</html>
```

Figura 4: Exemplo de código HTML com Javascript.

Fonte: O autor.

A propriedade `id="imagem"` ajusta a forma de comunicação entre a tag `` e o comando `document.getElementById('imagem')` para que, a cada interação com a imagem transformada em objeto clicável pela propriedade `onclick="troca()"`, ocorra uma troca de imagens.

A função `troca()` criada através de um comando de decisão `if...else` decide qual das imagens deve ser exibida de acordo com um critério de busca pelo nome do arquivo em uso, através da tag ``.

Os códigos Javascript podem ser incluídos dentro do HTML a qualquer momento, bastando, para isso, utilizar a sintaxe `style.color` para poder definir uma cor pelo Javascript.

Assim, o código Javascript pode ficar mais integrado ao HTML e dar uma aparência de estar lidando com uma linguagem única, com uma grande quantidade de recursos.

Na figura 6, temos o uso das propriedades `onmouseover` e `onmouseout` que permitem ações de acordo

com eventos do mouse. É importante observar que o código Javascript é todo inserido entre aspas duplas, indicando quando inicia e termina o mesmo.



Figura 5: Ilustração sobre webdesign.

```
<!DOCTYPE html>
<html>
<head>
<title>TROCA DE COR</TITLE>
<style>
  h1 {
    font-size : 16px;
  }
</style>
```

```

</head>

<body>

<h1 onmouseover="style.color='blue'" onmouseout="style.
color='red'">PASSE O MOUSE</h1>

</body>

</html>

```

Figura 6: Exemplo de código Javascript com HTML e CSS.
Fonte: O autor.

Outra excelente forma de incrementar códigos em HTML, além das linguagens CSS e Javascript, é através de bibliotecas, como a JQuery – que traz muitas funções Javascripts prontas, permitindo que a codificação seja otimizada, além de oferecer formas cada vez mais completas de se elaborar páginas web.

AQUISIÇÃO E INSTALAÇÃO DA JQUERY

JQuery é uma biblioteca criada para oferecer diversos recursos prontos para facilitar e reduzir a quantidade de linhas de codificação na criação de páginas web mais elaboradas para programação *Front-end*.

Esta biblioteca pode ser baixada em diversos locais na internet, sem custo; contudo, existe o site oficial¹. Lá, pode-se baixar uma das versões da biblioteca que aos poucos vai sendo aperfeiçoada.

Existem diversas versões disponíveis², nas quais temos o código para salvar na pasta que desejar, utilizando, por exemplo, o nome “jquery.js” para futuras inclusões em código através de uma linha de comando, como:



Figura 7: Ilustração das linguagens utilizadas neste livro.

¹ Disponível em: <<http://jquery.com/>>. Acesso em 18 dez. 2015.

² Disponível em: <<http://code.jquery.com/jquery-2.1.4.js>>. Acesso em 18 dez. 2015

```
<script src='jquery.js'></script>
```

Na unidade V, no estudo de caso, será possível observar como todas as linguagens estudadas e a biblioteca JQuery podem funcionar juntas para melhores resultados.

Os seletores da biblioteca JQuery iniciam sempre com o símbolo \$ e funcionam como ações aos elementos HTML, permitindo o uso de atributos – antes muito custosos – para se realizar em termos de código.

A sintaxe básica de um seletor JQuery é indicada abaixo, com nomes genéricos apenas para ilustração de como estruturar o comando.

```
$ ("p") .ação ()
```

Observe que, no exemplo de sintaxe, o seletor <P>, de parágrafo, é incluído entre aspas duplas, indicando sobre qual elemento HTML a ação assinalada após o ponto deve agir.

O exemplo da figura 8 mostra um bom uso da JQuery para criar botões, sendo um para exibição de mensagem de texto no início da página e outro para ocultar esta mensagem, através das ações `show` e `hide`, definidos em uma ação `click` – que determinou uma função `function()`, contendo este seletor “p”.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  fonte-size : 20px;
  color : red;
}
</style>

<script src="jquery.js"></script>
<script>
  $(document).ready(function() {
    $("#ve").click(function() {
      $("p").hide();
    });
  });
};
```

```

        $("#naoVe").click(function() {
            $("p").show();
        });
    });
</script>
</head>
<body>
    <p>MENSAGEM EXEMPLO</p>
    <button id="ve">VER</button>
    <button id="naoVe">NÃO VER</button>
</body>
</html>

```

Figura 8: exemplo de código JQuery com HTML.

Fonte: O autor.

Importante observar que o funcionamento dos dois botões, baseados nas ações *show* e *hide*, depende de uma correta identificação dos elementos através do uso de `id=""` para poder estabelecer o que ocorre no código corretamente.

REFLITA



O uso de uma mescla de linguagens é a melhor solução para programação, como ocorre no estudo deste livro, ou uma linguagem única que seja responsável por toda codificação?

Fonte: O autor.

CRIAÇÃO DE SCRIPTS

A base da criação dos códigos Javascript precisa ser alocada dentro de tags `<SCRIPT>` ou `"`, em propriedades de elementos do código.

A criação de scripts parte do princípio que os principais elementos são aqueles básicos do HTML e, a partir deles, atribui-se ações adicionais para esses elementos.

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
    $(document).ready(function() {
        $("button").click(function() {
            $("#div1").fadeOut("slow");
            $("#div2").fadeOut(3000, 0.5);
        });
    });
</script>
</head>
<body>
    <button>CLIQUE</button><br><br>
    <div id="div1" style="width:100px; height:100px;
    background-color:red;"></div>
    <div id="div2" style="width:50px; height:50px;
    background-color:black;"></div>
</body>
</html>
```

Figura 9: exemplo de efeito utilizando JQuery.

Fonte: O autor.

A figura 9 mostra o uso de alguns efeitos visuais usando a biblioteca JQuery através de ações como `fadeOut` – efeito de aumento da propriedade de brilho das caixas DIV, tendo seu tempo de duração regulado de acordo com parâmetros indicados.

Os valores dos parâmetros indicam tempo de duração do efeito e porcentagem do efeito de brilho, sendo os valores “3000” e “slow” para ajuste da duração e “0.5” para ajuste do nível de brilho para a cor.



Geolocation é uma API (Interface de Programação de Aplicativos) para indicação da posição do dispositivo em páginas relacionadas à passagens ou pontos de interesse em mapas.

A figura 10 mostra o código que indica sua posição. Confira a opção de localização em seu navegador (configuração e privacidade em painel de controle).

```
<!DOCTYPE html>
<html>
<body>
<p>Clique para conhecer sua posição geográfica. </p>
<p>A opção precisa ser ativada nas configurações de alguns navegadores.</p>
<button onclick="localizar()">Localizar</button>
<p id="demo"></p>
<script>
    var texto = document.getElementById("demo");
    function localizar() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(coordenadas);
        } else {
            texto.innerHTML = "Geolocation não suportado. Verifique opções em seu navegador ou utilize outro";
        }
    }
    function coordenadas(position) {
        texto.innerHTML = "Lat: " + position.coords.latitude +
            "<br>Long: " + position.coords.longitude;
    }
</script>
</body>
</html>
```

Figura 10: exemplo de código HTML para localização geográfica.

Fonte: O autor.

A figura 11 mostra o mesmo efeito de clareamento da cor da caixa DIV, mas com a diferença de que a ação foi modificada de `fadeOut` para `fadeToggle`, que inverte o estado do efeito cada vez que for ativado. Com isto, uma vez clicado o botão, a caixa clareia; e com outro clique, volta a cor normal da caixa.

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
    $(document).ready(function() {
        $("button").click(function() {
            $("div").fadeToggle(1000);
        });
    });
</script>
</head>

<body>
    <button>CLIQUE</button><br><br>
    <div style="width:200px; height:200px;
    background-color:red;"></div>
</body>
</html>
```

Figura 11: exemplo de efeito utilizando JQuery.

Fonte: O autor.

Para aperfeiçoar seus conhecimentos tanto em Javascript quanto em JQuery, é preciso buscar alternativas de estudo para conhecer mais seletores, propriedades e ações para criar efeitos cada vez mais criativos – e deixar páginas web cada vez mais bonitas.

CONSIDERAÇÕES FINAIS

Caro(a) aluno(a), é possível observar, com a passagem das unidades, o quão extensa é a lista de palavras reservadas, elementos, seletores, ações e propriedades que podem ser utilizadas para criar conteúdo para web.

Vimos que a linguagem Javascript permite programação mais elaborada e capaz de transformar um simples código HTML em algo capaz de realmente simular o funcionamento de um software; contendo comandos muito mais complexos e que permitem estruturas mais elaboradas de código, laços de decisão ou repetição. Com isso, páginas mais interativas e mais atrativas.

Por estar em um nível superior da codificação *Back-end* de páginas web, o que concerne o *Front-end* em HTML não trata da programação *Back-end*, mais avançada em termos de lógica de programação e manipulação de dados.

A biblioteca JQuery é especialmente desenvolvida para facilitar a programação em Javascript e é uma das ferramentas mais populares na área de desenvolvimento web. Contendo diversas ações prontas capazes de gerar efeitos muito bons, é uma das ferramentas mais produtivas, mesmo sendo gratuita, capaz de satisfazer uma grande quantidade de profissionais da área.

O desenvolvimento *Front-end* é muito interessante por tratar mais de layout e aparência do que de funções que uma página execute em si, dentro da finalidade principal de um site.

Outro ponto fundamental para esta área é a criatividade, pois um *web designer* que não é criativo pode não conseguir criar trabalhos que sejam reconhecidos como boas referências pelo mercado eletrônico – e perder espaço no mercado para profissionais até menos capazes, mas que sejam mais criativos.

Existem outras opções para auxiliar programadores como *Foundation Framework*, mantida pela Zurb, uma companhia de design de Interação da Califórnia, sendo essa uma ótima opção de código aberto para desenvolvedores.

Assim, encerramos o conteúdo técnico deste livro, mas deixando a todos a ideia de que há muito ainda a pesquisar sobre o assunto e existem muitas boas fontes de pesquisa.

Para complementar seus estudos, a unidade final trará um estudo de caso, no qual um pequeno *website* completo será criado, mostrando alguns detalhes que são melhor compreendidos em um projeto completo.

ATIVIDADES



1. A criação de páginas web é feita a partir de diversas linguagens disponíveis. Assinale a alternativa abaixo que apresente uma dupla de linguagens para criação de páginas web.
 - a. Cobol com JAVA.
 - b. HTML com Javascript.
 - c. CSS com C.
 - d. JQuery com Basic.
 - e. Javascript com C++.

2. Quando criamos páginas web, o projeto envolve a criação de conteúdos que devem ser apresentados a partir de um visual definido. A parte da codificação responsável por esta etapa é definida por qual(is) da(s) afirmações abaixo?
 - I. *Front-end* é a etapa de definição do layout e aparência da página, e pode conter apenas código HTML.
 - II. *Back-end* é a etapa na qual se define apenas as cores dos elementos.
 - III. Javascript pode trabalhar nas camadas de *Back-end* e *Front-end*.
 - IV. JQuery possibilita a redução de linhas de código e efeitos elaborados para elementos HTML.

Assinale a alternativa correta:

 - a. Apenas I e II estão corretas.
 - b. Apenas II e III estão corretas.
 - c. Apenas I está correta.
 - d. Apenas I, III e IV estão corretas.
 - e. Nenhuma das alternativas está correta.

3. Abaixo, existem alternativas contendo ações JQuery. Assinale V (verdadeiro) para as alternativas que sejam relativas à biblioteca e F (falso) para as que não foram tratadas em aula:

() show	() fadeToggle
() hide	() style
() fadeOut	() src

ATIVIDADES



4. O código Javascript permite uma grande quantidade de variações de código utilizando seus comandos. Crie o código em Javascript para a seguinte necessidade em uma página web: mudar a cor para verde (green) de um texto ao passar com o mouse sobre ele.

5. A biblioteca JQuery possibilita uma boa quantidade adicional de efeitos prontos para HTML. Descreva como poderia ser uma função para criar um efeito de clareamento até 50% em um elemento HTML.



EM BUSCA DE UM INSIGHT

Aquele momento especial em que temos um clarão em nossas mentes e a solução de algum problema, alguma ideia inovadora, uma decisão importante ou a resposta de algum exercício é vislumbrada. Este momento que não ocorre a todo momento, e que podem mudar completamente os rumos da humanidade, é chamado insight.

A forma como ocorre é sempre perceptível, pois é um momento seguido de uma euforia proporcional à importância do insight e pode servir para contagiar todo um grupo em qualquer atividade.

Existem alguns insights históricos, como o momento no qual Newton pôde perceber a força da gravidade e sua relação com o movimento da Lua ao redor da Terra, pela suposta queda de uma maçã em sua cabeça quando estava embaixo da macieira. Ou quando Arquimedes, ao citar a célebre expressão “Eureca”, pôde compreender como poderia ser medido o volume de objetos e sua densidade ao entrar em uma banheira com água.

É claro que em geral, um insight não gera toda uma teoria matemática nova e complexa instantaneamente, mas ele é capaz de, no momento mais importante de uma pesquisa, elucidar um elo faltante em uma pesquisa ou a solução de um impasse.

A criatividade está intimamente associada a estes momentos, pois são neles que as inovações podem ser imaginadas, mesmo que apenas inicialmente, para depois serem planejada e implementadas.

John Kounios e Mark Beeman, cientistas da Universidade Drexel e Universidade Northwestern, nos Estados Unidos, respectivamente colocaram pessoas para resolver problemas dentro de máquinas de ressonância magnética e perceberam que uma área se torna mais ativa quando ocorre um insight. Provaram que a criatividade trabalha na parte direita do cérebro, alguns centímetros atrás da orelha.

É possível exercitar o lado criativo do cérebro, pois o lado direito é aquele onde ideias menos detalhadas são trabalhadas, de forma que essas podem ser distorcidas do que realmente seriam.

Assim, através destas distorções é possível criar novas interpretações e, com isso, ter os insights tão desejados para novas ideias – seja para algo já existente ou novo.

Observar objetos e simplificar seu uso para tentar descobrir novas finalidades é uma forma de treinar o lado direito do cérebro, assim como jogos que estimulem a criatividade, nos quais não é preciso sempre seguir as mesmas regras aplicadas nas mesmas situações.

Fonte: o autor.





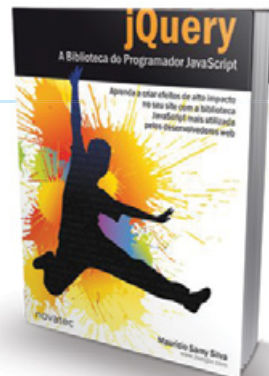
LIVRO

JQuery - A Biblioteca do Programador Javascript - 1ª Edição

Maurício Samy Silva

Editora: Novatec

Sinopse: Livro que apresenta a biblioteca, sua sintaxe e uso dos seletores JQuery. Contém diversos scripts de exemplo e estes podem ser baixados da página do livro no site da editora.



ESTUDO DE CASO

UNIDADE

V

Objetivos de Aprendizagem

- Explanar objetivos da unidade e sua finalidade.
- Definir o projeto e como ele será resolvido.
- Adquirir conteúdos necessários para o projeto.
- Demonstrar formas de se estruturar o layout e acrescentar conteúdos nas páginas do projeto.
- Explanar alternativas de hospedagem de websites.

Plano de Estudo

A seguir, apresentam-se os tópicos que você estudará nesta unidade:

- Estruturação do Projeto
- Produção de Conteúdo
- Elaboração das Páginas
- Publicação na Internet

INTRODUÇÃO

Olá, caro(a) aluno(a). Bem-vindo à última unidade do nosso material. Para completar a ideia deste trabalho, é interessante imaginar como podemos programar um pequeno projeto demonstrando todos os conceitos essenciais tratados neste livro, trazendo uma ideia de conjunto e não de conceitos isolados.

Criar algo maior que um mero exercício de aula traz outros tipos de benefícios, como noção de amplitude daquilo que pode ser realizado em um projeto real, mesmo em tamanho reduzido e faz com que seja possível compreender o quão trabalhoso é um projeto de maior porte e o quanto se aprende com maiores desafios, desde que sejam possíveis dentro da realidade proposta.

Não adianta, também, querer propor uma atividade imensa, uma vez que ela traria um excesso de trabalho sem que benefícios no aprendizado não sejam agregados na mesma proporção.

À medida que um projeto cresce, agrega novas ideias, porém, a partir de certo ponto, ele começa a ter mais repetições de conceitos já utilizados que de novos conceitos específicos para determinada etapa.

Assim, é interessante propor um projeto de tamanho adequado para as necessidades e a realidade geral, de forma que todos possam ser beneficiados, mesmo que em graus diferentes, uma vez que atender a todos é praticamente impossível.

Basicamente, a ideia desta unidade é propor a criação de um conjunto de páginas que contenha conteúdo produzido (texto e imagens), evitando todo tipo de plágio (a menos que devidamente citada a fonte de pesquisa), um certo número adequado de páginas, e a liberdade para futuras melhorias e acréscimo de mais páginas e conteúdo.

Nesta unidade, um estudo de caso contendo um pequeno site completo será analisado, dividindo as partes fundamentais de sua estrutura para facilitar a explicação e compreensão para, no final, ser disponibilizado o código completo para testes.

IMAGE BANNER

NAVIGATION

CONTENT

NEWS



©shutterstock

Figura 1: Ilustração de elaboração de layout de página web

ESTRUTURAÇÃO DO PROJETO

Unir conteúdos produzidos com estruturas criadas em HTML, CSS, javascript e JQuery é uma tarefa que exige cuidados e muita criatividade, pois a correta distribuição desses conteúdos definirá sua qualidade visual.

O item fundamental não é o quão complexa é a ferramenta utilizada para criar as páginas web, mas sim o quão prática e atrativa é a navegação por suas páginas. Isso é influenciado mais pela qualidade do Front-end que por detalhes técnicos da sua programação complexa.

Um site é desenvolvido a partir de ideias sobre objetivos e conteúdos que estarão contidos nas suas páginas, mas é preciso pensar que, normalmente, um site representa publicidade de longo alcance para um negócio.

Então, uma primeira meta é definir um tema para iniciar pesquisas e criação de conteúdos de acordo com a forma como será estruturado o site e qual o foco na definição do tipo de material a empregar.

Partimos, assim, do tema TURISMO para definir maiores detalhes do que poderia ser desenvolvido a partir de um tema tão vasto. Nesse caso, é interessante que seja estabelecido um conjunto de tópicos a serem abordados para reduzir a abrangência do tema.

Iremos, então, trabalhar com algumas capitais da América do Sul, mostrando imagens e curiosidades, tentando criar um pequeno site de turismo que possa servir de base para a estruturação de um site maior no futuro.

Abaixo, na figura 2, temos uma parte da página exibida pelo navegador, para ilustrar como é o resultado da codificação do site.



Figura 2: Parte da homepage exibida no navegador

Fonte: O autor.

Durante a unidade, será descrito o processo de codificação, exemplificando partes importantes de código, uma explicação do que faz e o seu resultado exibido pelo navegador.

SAIBA MAIS



Layout é um termo forte dentro da área de webdesign, haja vista que é nessa etapa que inovações podem ter efeito positivo ou negativo, podendo criar novos padrões ou adequar antigos a novas tecnologias.

Deve ser pesquisado e planejado de acordo com a especificação dada pelos interesses do cliente ou objetivos a atingir, e a escolha das ferramentas a utilizar no desenvolvimento pode ser influenciada por esta pesquisa, porém, em geral, depende mais da cultura da empresa desenvolvedora.

Dependendo do objetivo do site, um tipo de layout pode ser mais adequado que outro. Um hotsite direcionado à promoção de algo bem específico e com informações diretas terá um layout de página única, sem colunas.

Um portal maior já necessita de um conjunto de páginas para fazer a divisão do tema principal em subtemas para distribuir os conteúdos e não gerar páginas extensas extremamente carregadas.

Fonte: o autor.

PRODUÇÃO DE CONTEÚDO

Produzir conteúdo, como já citado, requer criatividade e noção de algumas ferramentas de hardware e software, incluindo câmeras profissionais ou não, e até fotos tiradas com dispositivos móveis. Softwares de tratamento de imagens simples ou mais complexos, dependendo do nível de experiência de cada um com essas ferramentas.

Basicamente, temos um conjunto de páginas HTML que serve de base para toda estruturação do projeto, sendo a homepage definida pelo arquivo `index.html`, e as demais com nomes que facilitem sua identificação.

Arquivos de imagens adquiridas todas do site Shutterstock¹ ajudarão a compor os conteúdos e o visual do site, complementando a parte textual do trabalho.

O texto contido em cada página retrata pontos de interesse e aspectos curiosos que compõem o conteúdo textual das páginas HTML e completam o material utilizado na codificação do projeto.

Finalmente, para completar a estruturação do site, fica incluso, dentro da mesma pasta dos demais arquivos do site, o arquivo “`jquery.js`” que contém a codificação da biblioteca JQuery utilizada na unidade IV e que pode ser baixada gratuitamente.

A figura 3 mostra uma parte do conteúdo da homepage em que imagens compõem a parte atrativa da página inicial do site que precisa de apelo visual para aumentar o interesse do usuário em navegar pelos seus conteúdos.



Figura 3: Conteúdo parcial da homepage
Fonte: o autor.

¹ Shutterstock é um site de fotos pago, que disponibiliza imagens de qualidade com resoluções variáveis, com opções de planos para acesso às fotos originais.

Essa parte da página tem função essencial para o site, uma vez que uma home-page atrativa prende temporariamente a atenção do usuário, mas os conteúdos das páginas adicionais precisam ser igualmente atraentes para que esse usuário se mantenha por mais tempo navegando dentro desse site.

Um detalhe muito importante é que a produção de conteúdos precisa ter o cuidado de utilizar material próprio ou com as devidas permissões de uso, para evitar problemas com direitos autorais.

Sites publicados precisam desse cuidado pela visibilidade mundial e existem diversos sites na Internet que oferecem figuras gratuitamente com liberação de direitos autorais para qualquer fim comercial ou não.

REFLITA



O plágio é extremamente prejudicial ao aprendizado, uma vez que, além de omitir fontes de pesquisa, impede o leitor de buscar maiores detalhes em outros materiais do autor, e ainda reduz a confiabilidade das informações.

Sempre utilize citações referenciando autores e fontes de pesquisa em sites ou livros para aumentar a qualidade de trabalhos que faça, inclusive na criação de websites com grande visibilidade e que possam criar problemas judiciais até.

Fonte: o autor.

ELABORAÇÃO DAS PÁGINAS

Para codificar todo o site proposto, temos, a seguir, uma explanação objetiva de partes do código do projeto, de forma a permitir que o você possa compreender cada parte do trabalho que será realizado.

Ao final desta unidade, será disponibilizado o código completo de cada página para que você possa testar e melhor compreender como é estruturado um *website* comum.

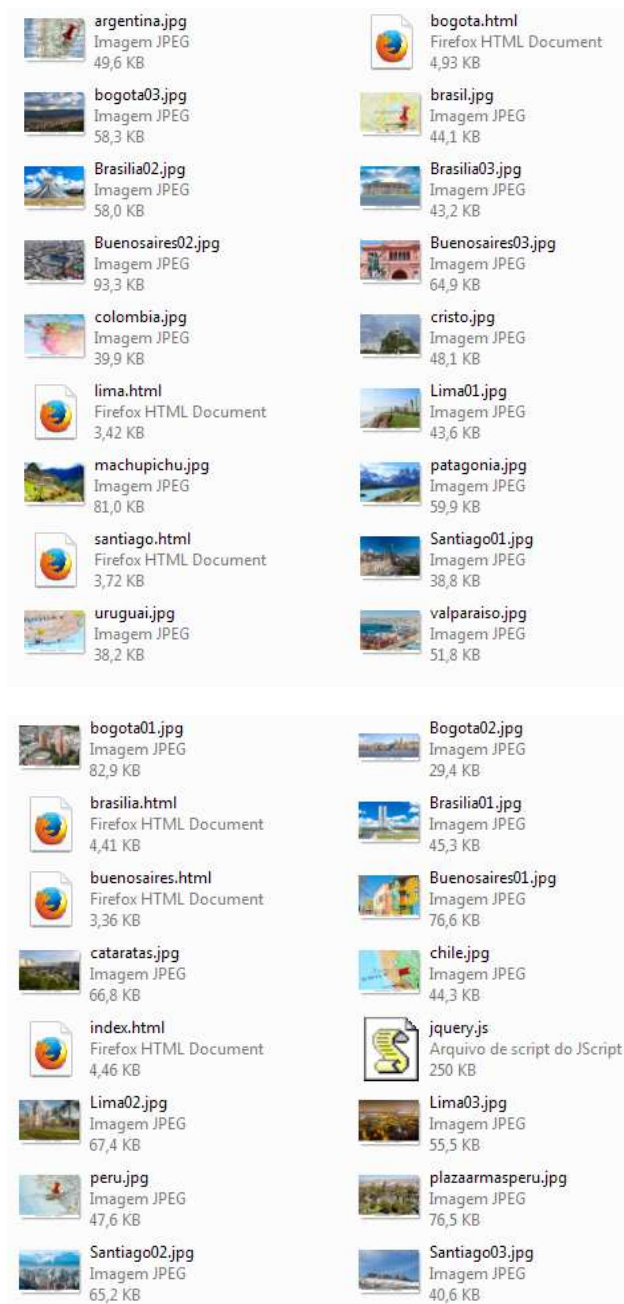


Figura 4: Listagem de todos os arquivos do Projeto
Fonte: o autor.

O arquivo “index.html” representa a homepage do site e contém código HTML, CSS e JQUERY unidos para criar o resultado final da página, sendo que a página poderia ser codificada de outras maneiras, de acordo com o nível de conhecimento, prática e design escolhido por cada programador.

A sua estrutura HTML básica segue o indicado na figura 4 que resume a parte essencial do código HTML dessa página em que temos as tags mais comuns e a tag <TITLE> para identificar a página na aba aberta no navegador.

```
<!DOCTYPE html>
<html>
<head>
  <title>América do Sul</title>
</head>
<body>
</body>
</html>
```

Figura 5: Código essencial HTML da homepage
Fonte: o autor.

Dentro do cabeçalho, são definidas as inserções de recursos da linguagem CSS e da biblioteca JQuery e, nas figuras 6 e 7, estão indicadas essas duas inserções dentro do código HTML.

```
<script src="jquery.js"></script>
<script>
  $(document).ready(function() {
    $("#barra").click(function() {
      $("#painel").slideToggle("slow");
    });
  });
</script>
```

Figura 6: Script JQuery para animação de barra com conteúdo
Fonte: o autor.

Nesse script da figura 6, temos indicados os seletores criados em CSS “barra” e “painel” que, por meio da ação “slideToggle”, trocam de lugar a cada clique na própria barra num movimento lento indicado pelo parâmetro “slow”.

Turismo na América do Sul

O turismo na América do Sul é uma das áreas que mais cresce na economia sul-americana. Com a maior floresta tropical do mundo (Amazônia), o maior rio do mundo (Amazonas), a segunda maior cadeia de montanhas (Andes), ilhas oceânicas isoladas (Galápagos, Ilha de Páscoa e Fernando de Noronha), praias paradisíacas (litoral do Nordeste Brasileiro), desertos (Atacama), paisagens glaciais (Patagônia e Terra do Fogo), a mais alta cachoeira do mundo (Salto Angel, com 979 metros de queda, na Venezuela) e as quedas com o maior volume de água (Cataratas do Iguaçu), dentre muitos outros monumentos naturais e criados pelo homem que atraem turistas de todo o mundo.

Figura 7: Resultado do clique para troca de “barra” por “painel”

Fonte: o autor.

Também fica indicada na figura 7 a mudança ocorrida ao clicarmos na barra, ativando o efeito no qual ocorre a troca da caixa menor “barra” pela maior “painel”.

Já a figura 8 indica todos os seletores configurados na homepage delimitados pelas tags <STYLE> e </STYLE> para ajustar cores, tamanhos e demais detalhes de fonte e caixas DIV.

```
<style>
#painel, #barra {
    padding: 5px;
    text-align: center;
    background-color: #EEC900;
    border: solid 0px;
    font-size: 20px
}
#painel {
    padding: 20px;
    display: none;
    font-size: 14px
}
body {
    background-color : white;
}
```

```
p{
    font-family: "Times New Roman", Times, serif;
    color: black;
    font-size: 14px;
}
```

Figura 8: Parte do código CSS para marcações em HTML
Fonte: o autor.

Os elementos ajustados na figura 8 são referentes à barra animada (#BARRA e #PAINEL), conteúdo geral (BODY) e parágrafos de texto (P). Pode-se observar que as configurações variam de acordo com os efeitos desejados e podem conter mais ou menos linhas de código.

Além desses elementos, também são ajustados os itens para criar os tópicos do menu (UL, LI e .MENU), a aparência dos links da página e o menu (A:LINK, A:VISITED, A:HOVER, ACTIVE) e, finalmente, os ajustes para as caixas DIV (.CAIXA e .CAIXA2) encerrando a área de estilos no cabeçalho.

Depois, a figura 9 mostra parte do código HTML utilizado para estruturar a homepage e indica a parte em que o menu, contendo link interno e externos, é codificado, utilizando as configurações definidas pelo código CSS.

```
<div class="menu"><nav> |
    <a href="index.html">HOME</a> |
    <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
    <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>
```

Figura 9: Exemplo de código HTML para criação de menu
Fonte: o autor.

Já na figura 10, você poderá observar que temos a parte em que um gráfico padrão SVG é criado para servir como um banner que indica em qual página o usuário se encontra no site.

```
<div class="menu">
<svg width="100%" height="100">
  <rect width="100%" height="100" style="fill:rgb(0,0,0);" />
  <text fill="#EEC900" font-size="30" font-family="Times New Roman"
x="50" y="40">AMÉRICA DO SUL</text>
  <text fill="#EEC900" font-size="20" font-family="Times New Roman"
x="50" y="80">Capitais Turísticas a Preços Acessíveis</text>
</svg>
```

Figura 10: Código HTML para criação de figura SVG

Fonte: o autor.

Na sequência da página, temos um pequeno formulário exemplo que pede ao usuário que preencha com seu nome e e-mail para um posterior envio por meio do botão “Enviar”.

Observe que o código da figura 11 envia os dados preenchidos para a mesma página “index.html” como exemplo, indicando na barra de endereços do navegador, os dados enviados em função da propriedade “method=“get”. Poderiam ser enviados a um servidor PHP, por exemplo, para armazenamento, trocando o nome indicado na propriedade “action”.

```
<form action="index.html" method="get">
  | Nome: <input type="text" name="nome" required>
  E-mail: <input type="email" name="email" required>
  <input type="submit" value="Enviar">
</form>
</div>
```

Figura 11: Código HTML para estruturação de formulário

Fonte: o autor.

Na figura 12, temos o código responsável por exibir no navegador uma das imagens que compõem a sequência de imagens representada na figura 3, e precisa ser replicado para cada uma das demais figuras.

```
<div class="caixa2">
  Cristo</div>
```

Figura 12: Código HTML para exibição de imagem em caixa DIV

Fonte: o autor.

O trecho de código exibido na figura 13 mostra uma das imagens maiores que serve de link interno para a respectiva página do site, contendo o conteúdo específico da cidade indicada.

```
<div class="caixa">
  <a href="buenosaires.html"><h2>Argentina</h2>
  </a>
</div>
```

Figura 13: Código HTML para imagem com link em caixa DIV

Fonte: o autor.

Finalmente, uma caixa DIV adicional é utilizada para indicar a fonte de pesquisa que também poderia indicar informações, como endereço físico, direitos autorais ou outras informações que não necessitam de destaque na figura 14.

```
<div class="caixa"><p>
  Fonte: Wikipédia, a enciclopédia livre.
</p></div>
```

Figura 14: Código HTML para para fonte de pesquisa em caixa DIV

Fonte: o autor.

Obviamente, existem sites muito mais complexos, contendo ligações com bancos de dados para criação de sistemas web, para gerenciamento de dados em empresas ou e-commerce para criação de lojas virtuais, por exemplo.

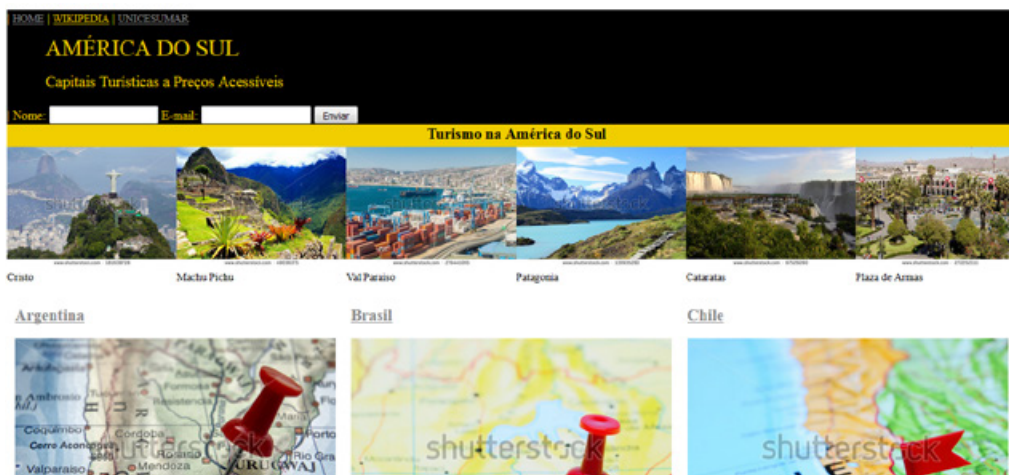


Figura 15: Ilustração da homepage completa

Fonte: o autor.

Continuando a explanação do projeto, uma das páginas que mostram conteúdos chamados pela homepage são descritos a seguir limitando a explicação a uma das páginas devido às demais seguirem exatamente a mesma estruturação de conteúdos e padrões CSS.

Nas demais páginas, temos uma estrutura semelhante, porém menor para estruturação das páginas de conteúdo, sendo que a figura 16 mostra como é exibida uma das páginas de conteúdo no navegador.



Figura 16: Página de conteúdo exibida pelo navegador

Fonte: o autor.

Nessas páginas, as codificações JQuery e CSS se mantêm iguais, padronizando a formatação de todo o site, tendo pequenas alterações em configurações de cores para os elementos da página.

Obviamente, os conteúdos textuais e imagens mudam a cada página de acordo com a cidade referente a cada página do website. O conjunto dessas páginas de cada cidade e a homepage formam o website.

A base do código HTML permanece também semelhante seguindo o mesmo visual “flat design” (estilo simples e leve) utilizado em dispositivos móveis e a partir do Windows 8.

Na figura 17, temos um exemplo de parte do código das demais páginas em que é possível perceber a grande semelhança com a codificação da homepage (“index.html”).

```
<div class="caixa">
    <h2>La Santa Maria</h2>
    
</div>
```

Figura 17: Exemplo de código HTML para exibição de imagem

Fonte: o autor.

Uma das pequenas diferenças é o uso da tag <H2> para definir uma fonte maior, também definida pelos seletores CSS para maior destaque do texto. Em geral, as mudanças são muito sutis e podem ser suprimidas, pois seguem a mesma linha de raciocínio da codificação da página inicial.

Finalizando, segue a lista dos arquivos do site conforme a ordem indicada na listagem abaixo:

- Figura 18, o código HTML do arquivo “index.html”.
- Figura 19, o arquivo “bogota.html”.
- Figura 20 para “brasil.html”.
- Figura 21 para “buenosaires.html”.
- Figura 22 para “lima.html”.
- Figura 23 para “santiago.html”.

```
<!DOCTYPE html>
<html>
<head>
<title>América do Sul</title>
<script src="jquery.js"></script>
<script>
    $(document).ready(function() {
        $("#barra").click(function() {
            $("#painel").slideToggle("slow");
        });
    });
</script>
```

```
<style>

#painel, #barra {
    padding: 5px;
    text-align: center;
    background-color: #EEC900;
    border: solid 0px;
    font-size: 20px
}

#painel {
    padding: 20px;
    display: none;
    font-size: 14px
}

body {
    background-color : white;
}

p{
    font-family: "Times New Roman", Times, serif;
    color: black;
    font-size: 14px;
}

ul {
    list-style-type: none;
    margin: 0px;
    padding: 10px;
    overflow: hidden;
}
```



```
li {
    float: left;
}

a:link {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : #EEC900;
}

a:visited {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : gray;
}

a:hover {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : green;
}

a:active {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : #EEC900;
}
```

```
.menu {
    float: left;
    width: 100%;
    margin: 0px;
    padding: 0px;
    background-color: black;
    color: #EEC900;
}

.caixa {
    float: left;
    margin: 5px;
    padding: 5px;
    width: 31.5%;
    border: 0px;
}

.caixa2 {
    float: left;
    width: 16.66%;
    border: 0px;
    font-size: 14px;
}

</style>
</head>

<body>

<div class="menu"><nav> |
    <a href="index.html">HOME</a> |
    <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
    <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>
```

```

<div class="menu">
<svg width="100%" height="100">
  <rect width="100%" height="100" style="fill:rgb(0,0,0);" />
  <text fill="#EEC900" font-size="30" font-family="Times New Roman"
x="50" y="40">AMÉRICA DO SUL</text>
  <text fill="#EEC900" font-size="20" font-family="Times New Roman"
x="50" y="80">Capitais Turísticas a Preços Acessíveis</text>
</svg>
<form action="index.html" method="get">
  | Nome: <input type="text" name="nome" required>
  E-mail: <input type="email" name="email" required>
  <input type="submit" value="Enviar">
</form>
</div>
<div id="barra"><strong>Turismo na América do Sul</strong></div>
<div id="painel">O turismo na América do Sul é uma das áreas que mais
cresce na economia sul-americana. Com a maior floresta tropical do
mundo (Amazônia), o maior rio do mundo (Amazonas), a segunda maior
cadeia de montanhas (Andes), ilhas oceânicas isoladas (Galápagos,
Ilha de Páscoa e Fernando de Noronha), praias paradisíacas (lito-
ral do Nordeste Brasileiro), desertos (Atacama), paisagens glaciais
(Patagônia e Terra do Fogo), a mais alta cachoeira do mundo (Salto
Angel, com 979 metros de queda, na Venezuela) e as quedas com o maior
volume de água (Cataratas do Iguaçu), dentre muitos outros monumentos
naturais e criados pelo homem que atraem turistas de todo o mundo.
</div>

<div class="caixa2">
  Cristo</div>
<div class="caixa2">
  Machu Pichu</div>

```

```
<div class="caixa2">
  Val Paraíso</div>
<div class="caixa2">
  Patagonia</div>
<div class="caixa2">
  Cataratas</div>
<div class="caixa2">
  Plaza de Armas</div>

<div class="caixa">
  <a href="buenosaires.html"><h2>Argentina</h2>
  </a>
</div>
<div class="caixa">
  <a href="brasil.html"><h2>Brasil</h2>
  </a>
</div>
<div class="caixa">
  <a href="santiago.html"><h2>Chile</h2>
  </a>
</div>
<div class="caixa">
  <a href="bogota.html"><h2>Colômbia</h2>
  </a>
</div>
<div class="caixa">
  <a href="lima.html"><h2>Peru</h2>
  </a>
</div>
<div class="caixa">
  <a href="montevideo.html"><h2>Uruguai</h2>
  </a>
</div>
```

```
<div class="caixa"><p>
    Fonte: Wikipédia, a enciclopédia livre.
</p></div>

</body>
</html>
```

Figura 18: Código fonte para o arquivo “index.html”
Fonte: o autor.

```
<!DOCTYPE html>
<html>

<head>
<title>Bogotá</title>
<script src="jquery.js"></script>
<script>
    $(document).ready(function() {
        $("#barra").click(function() {
            $("#painel").slideToggle("slow");
        });
    });
</script>

<style>
    #painel, #barra {
        padding: 5px;
        text-align: center;
        background-color: #FFA500;
        border: solid 0px;
        font-size: 20px
    }
```

```
#painel {  
    padding: 20px;  
    display: none;  
    font-size: 14px  
}  
  
body {  
    background-color : white;  
}  
  
p {  
    font-family: "Times New Roman", Times, serif;  
    font-size : 14px;  
    color : #FFA500;  
}  
  
h2 {  
    color: #FFA500;  
    font-size: 20px;  
}  
  
a:link {  
    font-family: "Times New Roman", Times, serif;  
    font-size : 14px;  
    color : red;  
}  
  
a:visited {  
    font-family: "Times New Roman", Times, serif;  
    font-size : 14px;  
    color : gray;  
}
```

```

a:hover {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : orange;
}

a:active {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : red;
}

.caixa {
    float: left;
    margin : 5px;
    padding: 15px;
    width: 29.5%;
    border: 0px;
}
</style>
</head>

<body>
<div class="menu"><nav> |
    <a href="index.html">HOME</a> |
    <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
    <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>

```

```
<svg width="100%" height="100">
  <rect width="100%" height="100" style="fill:rgb(255,255,0);" />
  <text fill="#FF0000" font-size="30" font-family="Times New Roman"
    x="50" y="40">BOGOTÁ - COLÔMBIA</text>
  <text fill="#FF0000" font-size="20" font-family="Times New Roman"
    x="50" y="80">Quase de 7 Milhões de Habitantes em Seu Entorno</
  text>
</svg>
```

```
<div id="barra"><strong>História da Capital Colombiana</strong></div>
<div id="painel">Através da história, Bogotá foi conhecida por dife-
rentes nomes. O nome original era "Thybzacá" ou "Teusacá" para depois
se chamar Teusaquillo.<br />
```

Em 1538, o conquistador Gonzalo Jiménez de Quesada fundou a cidade, como Nuestra Señora de la Esperanza (Nossa Senhora da Esperança). Em 1539, o nome foi alterado para Santafé

A constituição de 1991 muda o nome da capital para Santafé de Bogotá. Em 2000 uma reforma constitucional alterou o nome para Bogotá apenas.

```
</div>
```

```
<div style="background-color:#B22222; color:#ffffff; padding:20px;">
```

```
<h2>Bogotá - Turismo</h2>
```

<p>Os pontos turísticos importantes em Bogotá são o "Jardim Botânico José Celestino Mutis", a "Quinta de Bolívar", o "Observatório Astronômico Nacional", o "Planetário de Bogotá", o "Maloka" (parque temático sobre ciencia e tecnologia) , o "Mirador Metropolitano da Torre Colpatria", o "Mirador de La Calera", o "Monumento de Banderas a las Américas (ao lado da estação banderas do TransMilenio) e La Candelaria, localidade que é patrimônio e Bem de Interesse Cultural de Carater Nacional. Além disso, a cidade conta com numerosos parques de atrações mecânicas como: "Salitre Mágico", "Mundo Aventura" e "Camelot".

Em Bogotá existe uma variada oferta de alojamentos que oscila de pousadas para mochileiros a hotéis cinco estrelas. A oferta de hotéis no centro histórico de La Candelaria e seus arredores, é dirigida ao público amante da cultura e das artes. Os hotéis localizados próximo a Ciudad Salitre são destinados aos visitantes que fazem escalas curtas em Bogotá ou necessitam das proximidades do Aeroporto Internacional El Dorado. Por outro lado, os hotéis localizados para o norte da cidade, estão focados para o turismo de negócios, compras ou lazer. Os arredores de Bogotá como a "Laguna de Fúquene" e os "Embalses del Neusa", "Tominé" e "Sisga", estão habilitados para a prática do camping e esportes aquáticos. Também é possível acampar no "Parque Nacional Natural Sumapaz" (entre o Distrito Capital e o departamento de Meta), ou no "Parque Nacional Natural Chingaza" a leste de Bogotá.

A cidade conta com o "Trem Turístico da Savana de Bogotá" que circula aos finais de semana e feriados do centro de Bogotá até o "Parque Jaime Duque", Zipaquirá e em ocasiões, até Nemocón. Em Zipaquirá está a Catedral de Sal (Patrimônio Histórico e Monumento Nacional). Na mesma região está Guatavita, local provável da lenda de Eldorado. Por outro lado, próximo ao Salto de Tequendama, uma queda de água localizada a uma hora de viagem do sul de Bogotá, pode-se encontrar o "Zoológico Santa Cruz". Também ao sul, em Soacha, se encontra o "Parque Natural Chicaque", e saindo pela Autopista Sul, a quase três horas de Bogotá, se encontram os balneários de Melgar (Tolima) e Girardot.

Em Facatativá, a oeste de Bogotá, se encontra o centro arqueológico conhecido "Parque Arqueológico Piedras del Tunjo" e em Nemocón se encontram algumas famosas minas de sal.</p>
</div>
<div class="caixa">
<h2>La Santa Maria</h2>

</div>

```

<div class="caixa">
    <h2>Catedral Principal de Bogotá</h2>
    
</div>

<div class="caixa">
    <h2>Vista Aérea de Bogota</h2>
    </p>
</div>

<div id="footer"><p>
Fonte: http://www.melhoresdestinos.com.br/lima-peru.html.
</div>

</body>
</html>

```

Figura 19: Código fonte para o arquivo “bogota.html”

Fonte: o autor.

```

<!DOCTYPE html>
<html>

<head>
<title>Brasília</title>
<script src="jquery.js"></script>
<script>

    $(document).ready(function() {
        $("#barra").click(function() {
            $("#painel").slideToggle("slow");
        });
    });
</script>

```

```
<style>

#painel, #barra {
    padding: 5px;
    text-align: center;
    background-color: #4169E1;
    border: solid 0px;
    font-size: 20px
}

#painel {
    padding: 20px;
    display: none;
    font-size: 14px
}

body {
    background-color : white;
}

p {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : #0000CD;
}

h2 {
    color: #0000CD;
    font-size: 20px;
}
```

```
a:link {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : green;
}

a:visited {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : gray;
}

a:hover {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : yellow;
}

a:active {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : green;
}

.caixa {
    float: left;
    margin : 5px;
    padding: 15px;
    width: 29.5%;
    border: 0px;
}

</style>
</head>
```

```

<body>
<div class="menu"><nav> |
  <a href="index.html">HOME</a> |
  <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
  <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>

<svg width="100%" height="100">
  <rect width="100%" height="100" style="fill:rgb(0,100,0);" />
  <text fill="#FFFF00" font-size="30" font-family="Times New Roman"
x="50" y="40">BRASÍLIA - BRASIL</text>
  <text fill="#ffff00" font-size="20" font-family="Times New Roman"
x="50" y="80">Cerca de 4 Milhões de Habitantes em Seu Entorno</text>
</svg>

<div id="barra"><strong>História da Atual Capital Brasileira</
strong></div>
<div id="painel">Desde de 1.891, a primeira constituição republicana
já indicava que a capital federal deveria se transferir do Rio de
Janeiro para o interior do país. <br />
Em 1.922, no ano do Centenário da Independência do Brasil, após pro-
jeto enviado a Câmara, foi assentada a pedra fundamental para o início
da construção da cidade.<br />
De 1.955 a 1.960, ocorreu toda a construção da cidade, sendo então
inaugurada na data de 21 de abril de 1.960.
</div>

<div style="background-color: #FFD700; color:black; padding:20px;">
  <h2>Brasília</h2>
  <p>Brasília é classificada como Patrimônio Cultural da Humanidade pela
Unesco, uma agência da ONU e recebe cerca de um milhão de visitantes
anualmente. Entre as suas atrações mais visitadas estão os diver-
sos projetos arquitetônicos de Oscar Niemeyer.

```

O turismo cívico é valorizado por estarem localizados na capital os órgãos governamentais da administração direta e os representantes dos três poderes republicanos. Os principais monumentos da cidade encontram-se no Eixo Monumental: Catedral Militar Rainha da Paz, Praça do Cruzeiro (Memorial da Primeira Missa), Memorial JK, Memorial dos Povos Indígenas, Complexo Poliesportivo Ayrton Senna: Ginásio de Esportes Nilson Nelson e Estádio Nacional de Brasília Mané Garrincha; Centro de Convenções Ulysses Guimarães (CCUG), Torre de TV, Teatro Nacional Cláudio Santoro, Complexo Cultural da República João Herculino: Biblioteca Nacional de Brasília Leonel de Moura Brizola (BNB) e Museu Nacional Honestino Guimarães; Catedral Metropolitana de Brasília Nossa Senhora Aparecida, Esplanada dos Ministérios, Palácio da Justiça, Palácio Itamaraty, Praça dos Três Poderes: Congresso Nacional, sede do Poder Legislativo brasileiro, Palácio do Planalto, sede do Poder Executivo brasileiro, Supremo Tribunal Federal (STF), sede do Poder Judiciário brasileiro e Panteão da Pátria e da Liberdade Tancredo Neves; além de outros. Entre outros monumentos estão o Palácio da Alvorada, residência oficial da Presidência da República (no Setor Palácio Presidencial - SPP), o Catetinho (ao longo da EPIA Sul), o Santuário Dom Bosco (na Via W3 Sul), o Museu Vivo da Memória Candanga (no Núcleo Bandeirante) e a Ponte Juscelino Kubitschek, mais conhecida como Ponte JK ou "terceira ponte", premiada internacionalmente (no Lago Paranoá, entre o Setor de Clubes Esportivos Sul (SCES), na Asa Sul, e o Setor de Habitações Individuais Sul (SHIS), no Lago Sul).

</div>

<div class="caixa">

<h2>Congresso Nacional</h2>

</div>

```
<div class="caixa">
  <h2>Catedral de Brasília</h2>
  
</div>

<div class="caixa">
  <h2>Estádio Mané Garrincha</h2>
  
</div>

<div id="footer"><p>
  Fonte: Wikipedia, a enciclopedia livre.</p>
</div>
</body>
</html>
```

Figura 20: Código fonte para o arquivo “brasil.html”

Fonte: o autor.

```
<!DOCTYPE html>
<html>
<head>
<title>Buenos Aires</title>
<script src="jquery.js"></script>
<script>
  $(document).ready(function() {
    $("#barra").click(function() {
      $("#painel").slideToggle("slow");
    });
  });
</script>
```

```
<style>

#painel, #barra {
    padding: 5px;
    text-align: center;
    background-color: #1E90FF;
    border: solid 0px;
    font-size: 20px
}

#painel {
    padding: 20px;
    display: none;
    font-size: 14px
}

body {
    background-color : white;
}

p {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : #000080;
}

h2 {
    font-family: "Times New Roman", Times, serif;
    font-size : 20px;
    color : #000080;
}
```



```

a:link {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : blue;
}

a:visited {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : gray;
}

a:hover {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : lightblue;
}

a:active {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : blue;
}

.caixa {
    float: left;
    margin: 5px;
    padding: 15px;
    width: 29.5%;
    border: 0px;
}
</style>
</head>

```

```

<body>
<div class="menu"><nav> |
  <a href="index.html">HOME</a> |
  <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
  <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>

<svg width="100%" height="100">
  <rect width="100%" height="100" style="fill:rgb(0,0,100);" />
  <text fill="#ffffff" font-size="30" font-family="Times New Roman"
x="50" y="40">BUENOS AIRES - ARGENTINA</text>
  <text fill="#ffffff" font-size="20" font-family="Times New Roman"
x="50" y="80">Cerca de 3 Milhões de Habitantes em Seu Entorno</text>
</svg>

<div id="barra"><strong>História da Atual Capital Argentina</strong></div>
<div id="painel">Pedro de Mendoza fundou a cidade em 1.536 chama-
do-a de Nuestra Señora del Buen Ayre.<br />
  Foi arrasada por índios, e em 1.580 refundada por Juan de Garay
com o nome de Ciudad de la Santísima Trinidad y Puerto de Nuestra
Señora del Buen Ayre.<br />
  Começou seu crescimento somente após 1.776 quando se tornou capi-
tal do vice-reino do Rio-da-Prata.
</div>

<div style="background-color:#87CEEB; color:black; padding:20px;">
  <h2>Buenos Aires</h2>
  <p>A cidade conta com o Aeroporto Internacional de Ezeiza (situado a
35 quilômetros da cidade) e com um aeroporto doméstico, o Aeroparque
Jorge Newbery. É servida pelo terminal rodoviário de Retiro, de
onde partem e chegam linhas de ônibus (autocarros) para todas as
regiões do país e para cidades do Chile, Peru, Bolívia, Paraguai,

```

Uruguai e Brasil. Conta ainda com ferry-boats que a conectam com as cidades de Colônia do Sacramento e Montevidéu , no vizinho Uruguai. Fica a 1630 quilômetros ao norte de outro importante destino turístico argentino, a cidade de Bariloche. Os shows de tango são um dos principais atrativos da cidade, até porque constituem uma das características mais afamadas da cultura argentina. São muitas as casas de espetáculos que apresentam shows variados de dança e música, muito procurados por quem faz turismo na cidade.</p>

</div>

```
<div class="caixa">
  <h2>La Boca</h2>
  
</div>
```

```
<div class="caixa">
  <h2>La Bombonera</h2>
  
</div>
```

```
<div class="caixa">
  <h2>Casa Rosada</h2>
  
</div>
```

```
<div class="footer"><p>
  Fonte: Wikipedia, a enciclopedia livre.</p>
</div>
```

</body>

</html>

Figura 21: Código fonte para o arquivo “buenosaires.html”

Fonte: o autor.

```
<!DOCTYPE html>
<html>

<head>
<title>Lima</title>
<script src="jquery.js"></script>
<script>
    $(document).ready(function(){
        $("#barra").click(function(){
            $("#painel").slideToggle("slow");
        });
    });
</script>

<style>
    #painel, #barra {
        padding: 5px;
        text-align: center;
        background-color: #EE3B3B;
        border: solid 0px;
        font-size: 20px
    }

    #painel {
        padding: 20px;
        display: none;
        font-size: 14px
    }

    body {
        background-color : white;
    }
```

```
p {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : #8B1A1A;
}

h2 {
    font-family: "Times New Roman", Times, serif;
    font-size : 20px;
    color : #8B1A1A ;
}

a:link {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : red;
}

a:visited {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : gray;
}

a:hover {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : green;
}

a:active {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : red;
}
```

```

        .caixa {
            float: left;
            margin: 5px;
            padding: 15px;
            width: 29.5%;
            border: 0px;
        }
    </style>
</head>

<body>
<div class="menu"><nav> |
    <a href="index.html">HOME</a> |
    <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
    <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>

<svg width="100%" height="100">
    <rect width="100%" height="100" style="fill:rgb(100,0,0);" />
    <text fill="#ffffff" font-size="30" font-family="Times New Roman"
x="50" y="40">LIMA - PERU</text>
    <text fill="#ffffff" font-size="20" font-family="Times New Roman"
x="50" y="80">Quase 9 Milhões de Habitantes em Seu Entorno</text>
</svg>

<div id="barra"><strong>História da Atual Capital Peruana</strong></div>
<div id="painel">Fundada em 1.535 e chamada de a Cidade dos Reis, em
território formado por vales de rios.<br />
Por ter sido fundada em território indígena, foi sitiada em 1536, mas
foram derrotados pelos espanhóis unidos a indígenas aliados.<br />
Alvo de piratas, muralhas foram construídas nos anos 1680, e abalos
sísmicos também já causaram grandes estragos a cidade.
</div>

```

```
<div style="background-color: #CD5C5C ; color:black; padding:20px;">
  <h2>Lima - Turismo</h2>
  <p>Lima é uma metrópole com uma história e geografia peculiar. Além
    de suas atrações como sítios arqueológicos, praças, parques e uma
    culinária fantástica, Lima serve como ponto de partida para uma via-
    gem por diversos destinos no Peru.

    Fazer um city tour é sempre uma boa opção para ter uma visão geral
    da cidade e conhecer seus principais pontos turísticos. Locais como
    a Plaza de Armas, a Catedral e o Parque del Amor são imperdíveis.
    Lima tem quase 8 milhões de habitantes e como toda grande cidade tem
    muitas opções de restaurantes, bares, boates, shoppings. Um local
    muito agradável é que junta tudo isso é o Larcomar, um Shopping a
    céu aberto com vista para o Oceano Pacífico.

    Esse é um dos atrativos do Peru. Lima é chamada por alguns de capi-
    tal gastronômica da América. Lá há muitos restaurantes com os mais
    variados pratos. Os pratos típicos de lá são muito saborosos, colo-
    ridos e custam pouco.
  </p>
</div>

<div class="caixa">
  <h2>Paisagem Costeira</h2>
  
</div>

<div class="caixa">
  <h2>Catedral Principal</h2>
  
</div>
```

```

<div class="caixa">
  <h2>Visão Noturna</h2>
  
</div>

<div id="footer"><p>
  Fonte: http://www.melhoresdestinos.com.br/lima-peru.html.</p>
</div>
</body>
</html>

```

Figura 22: Código fonte para o arquivo "lima.html"

Fonte: o autor.

```

<!DOCTYPE html>
<html>
<head>
<title>Santiago</title>
<script src="jquery.js"></script>
<script>
  $(document).ready(function(){
    $("#barra").click(function(){
      $("#painel").slideToggle("slow");
    });
  });
</script>

<style>
  #painel, #barra {
    padding: 5px;
    text-align: center;
    background-color: #4682B4;
    border: solid 0px;
    font-size: 20px
  }

```



```
#painel {
    padding: 20px;
    display: none;
    font-size: 14px
}

body {
    background-color : white;
}

h2 {
    font-family: "Times New Roman", Times, serif;
    font-size : 20px;
    color : #0000CD;
}

p {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : #0000CD;
}

a:link {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : blue;
}

a:visited {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : gray;
}
```

```

a:hover {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : red;
}

a:active {
    font-family: "Times New Roman", Times, serif;
    font-size : 14px;
    color : blue;
}

.caixa {
    float: left;
    margin: 5px;
    padding: 15px;
    width: 29.5%;
    border: 0px;
}
</style>
</head>

<body>
<div class="menu"><nav> |
    <a href="index.html">HOME</a> |
    <a href="https://pt.wikipedia.org" target="_blank">WIKIPEDIA</a> |
    <a href="http://www.cesumar.br" target="_blank">UNICESUMAR</a>
</nav></div>

<svg width="100%" height="100">
    <rect width="100%" height="100" style="fill:rgb(0,0,150);" />
    <text fill="#ffffff" font-size="30" font-family="Times New Roman"
    x="50" y="40">SANTIAGO - CHILE</text>

```

```
<text fill="#ffffff" font-size="20" font-family="Times New Roman"
x="50" y="80">Mais de 6 Milhões de Habitantes em Seu Entorno</text>
</svg>
```

```
<div id="barra"><strong>História da Capital Chilena</strong></div>
```

```
<div id="painel">Fundada pelo conquistador espanhol Pedro de Valdivia
em fevereiro de 1.541 sob o nome de Santiago de Nueva Extremadura,
santo patrono da Espanha.<br />
```

```
Na Guerra do Arauco, a cidade foi destruída por nativos liderados por
Michimalonco em setembro no mesmo ano.<br />
```

```
Dificuldades com relação a água e estando cercada pela cordilheira,
começou e se desenvolver na década de 1.880 após a criação de uma
estrada que facilitou op acesso e expansão da cidade.
```

```
</div>
```

```
<div style="background-color:#B22222; color:white; padding:20px;">
```

```
<h2>Santiago - Turismo</h2>
```

```
<p>A cidade é um importante destino turístico a nível nacional, por ser
a principal porta de entrada ao país através do Aeroporto Internacional
Comodoro Arturo Merino Benítez e do Paso Transandino Internacional Los
Libertadores. Ambos concentram 55,2% do total de pessoas que ingres-
sam no Chile por ano, o equivalente a 1 119 840 de turistas em 2005.
Segundo um estudo do Serviço Nacional de Turismo, 52,3% dos turistas
(tanto nacionais como internacionais) têm como destino a categoria
"Santiago e seus arredores", sendo que quase toda esta parcela tam-
bém visita as estações de esqui localizadas na Cordilheira dos Andes.
Com vários atrativos históricos e culturais, museus, palácios, miran-
tes, shows, vida noturna intensa, compras, entre outros, se tornou,
principalmente após o "boom" econômico da década de 1990, uma grande
opção turística. Além disso, se encontra a menos de 100 km do lito-
ral e de estâncias de esqui (entre elas, a mais moderna da América
do Sul). Provenientes de todas as partes do mundo, os turistas que
```

```
conhecem a cidade são principalmente europeus, estadunidenses, cana-
denses e brasileiros.</p>
```

```
</div>
```

```
<div class="caixa">
```

```
  <h2>Praça de Armas</h2>
```

```
  
```

```
</div>
```

```
<div class="caixa">
```

```
  <h2>Santiago e a Cordilheira</h2>
```

```
  
```

```
</div>
```

```
<div class="caixa">
```

```
  <h2>Vale Nevado</h2>
```

```
  
```

```
</div>
```

```
<div id="footer"><p>
```

```
  Fonte: Wikipedia, a enciclopedia livre.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Figura 23: Código fonte para o arquivo “santiago.html”

Fonte: o autor.

Alguns podem não ter percebido ao implementar esses código, mas a página que gera a última parte do site, que mostra o conteúdo para Montevideo, capital do Uruguai, não foi colocada para que você possa, como exercício, elaborá-la seguindo o padrão da codificação das demais.

PUBLICAÇÃO NA INTERNET

A publicação de um site tem uma sequência de passos básicos necessários que devem ser observados para que tudo esteja de acordo e dentro dos padrões estabelecidos para a Internet.

Um primeiro passo é o registro de um domínio para Internet, que pode ser feito por um provedor, o qual também disponibilizará o espaço para a postagem dos arquivos do site, ou pelo site de registro, disponível em: <<https://registro.br/>>. Nesse site, temos a opção de escolher, entre domínios disponíveis, aquele que mais seja adequado à ideia proposta pelo site, gerando uma taxa anual de valor simbólico a ser paga em banco.

O domínio é único e o registro o torna propriedade daquele que o cadastrou e pagou a taxa de registro anual, podendo ser de tipos como .COM, .ORG, .INF, .COM.BR etc.

Após essa etapa, é preciso um provedor de serviços de hospedagem de sites, como Locaweb, UOL etc., que são responsáveis por manter o site no ar após inseridos os arquivos no espaço, por meio de um gerenciador de arquivos na própria nuvem.

Esse provedor de hospedagem pode ser gratuito ou pago, tendo diferenças entre espaço disponível, quantidade de e-mails disponibilizados, velocidade de banda e quantidade de acessos por mês. Uma empresa precisa optar por um provedor pago confiável para garantir que o conteúdo esteja disponível 24 horas por dia durante os 365 dias do ano se possível de modo aceitável.

Finalmente, após a publicação, é possível acionar cadastros em motores de busca, como Google ou Bing, para facilitar a visualização e divulgação, visto que, hoje em dia, as pessoas não conhecem mais endereços de sites, mas sim os encontram por esses sites de busca.

Adicionalmente, pode-se utilizar agregadores de links como os dos sites Colméia (disponível em: <<http://www.colmeia.blog.br/>>.) e Uêba (disponível em: <<http://ueba.com.br/>>.) para adicionar páginas em diretórios públicos.

A utilização de páginas em redes sociais como o Facebook e perfis em locais, como o Twitter, são ótimas formas de divulgação, podendo incluir neles links para seu site publicado.

Já o Google AdWords², exige um pequeno investimento inicial por meio de uma conta criada no Google (gmail) para que, a partir disso, seja possível criar anúncios personalizados que terão destaque em resultados de buscas com palavras-chaves definidas.

CONSIDERAÇÕES FINAIS

Muito bem, caro(a) aluno(a), finalizando esta unidade, foi possível compreender como unir os conceitos vistos por meio das unidades do livro com um exemplo completo de código para elaboração de um site, exemplo com várias páginas. Esse código contemplou um pouco de cada conteúdo das demais unidades do livro com o intuito de unir todos esses conceitos, mostrando como existe relação entre eles.

Além de compreender a relação entre as linguagens, é possível compreender o ponto em que cada linguagem é inserida dentro do código HTML, para permitir que as páginas que compõem o site tenham conteúdos atrativos e formatados de maneira a tornarem a navegação mais agradável.

O conjunto de páginas mostrou, também, que nomes de fácil identificação contribuem para melhor compreensão do conteúdo da pasta de trabalho e facilitam na identificação de cada elemento na codificação HTML.

Nomes simplificados e sem acentuação melhoram a portabilidade do site em diferentes sistemas operacionais e, possivelmente, até em navegadores, tornando o site universalizado.

² Sistema de anuncio do Google em que os primeiros resultados de uma pesquisa representam os sites cadastrados no AdWords para maior visibilidade, indicados pela palavra ANÚNCIO.

Um detalhe interessante que pode ser destacado é o exercício da criatividade, que contribui muito com a qualidade do design das páginas, além de permitir uma diferenciação em relação a sites que seguem padrões prontos oferecidos por provedores de hospedagem.

O layout das páginas deve ser cuidadosamente elaborado a fim de proporcionar, além de beleza, eficiência e facilidade na compreensão de como flui a navegação por estas páginas.

Um *website* pode trazer enormes benefícios a um negócio por gerar uma boa propaganda com enorme visibilidade, no entanto, em contrapartida, um *website* mal feito pode gerar desconfiança na qualidade dos serviços prestados pela empresa dona do site.

O mais importante é praticar para que o tempo traga seus frutos e permita que, com pesquisas e experiência, o programador *Front-end* possa, a cada novo projeto, criar conteúdos mais bem estruturados que façam com que seu trabalho seja reconhecido pelo mercado.

ATIVIDADES



1. Um website é composto por partes que juntas oferecem um conjunto de _____ que podem conter texto e _____. Assim, temos a ideia de que um site pode ser constituído por diversos _____.
 - a. páginas, conteúdo multimídia, computadores.
 - b. páginas, conteúdo multimídia, arquivos.
 - c. tipos de dados, conteúdo multimídia, arquivos.
 - d. páginas, sons, computadores.
 - e. sons, conteúdo multimídia, computadores.

2. Todo site possui um arquivo que representa sua homepage para o navegador. Como se chama este arquivo por padrão?
 - a. "indice.html".
 - b. "homepage.html".
 - c. "home.html".
 - d. "index.html".
 - e. Nenhuma das alternativas está correta.

3. Algumas elementos foram utilizados na composição do site desta unidade. Assinale verdadeiro (V) ou falso (F) para os elementos indicados abaixo e que estejam no código da homepage.
 - () <html>
 - () <php>
 - () <css>
 - () <style>
 - () <script>

4. Crie um exemplo de seletor CSS qualquer que contenha pelo menos duas configurações quaisquer para fonte ou outro elemento.

5. A ligação entre as páginas é feita utilizando uma tag para criação de links. Dê um exemplo de link que pudesse chamar uma página gravada em um arquivo chamado "pagina01.html" por exemplo.



Nada melhor que exercitar novos conhecimentos com atividades e exemplos práticos na área de programação, seja ela de uma linguagem mais simples como o HTML, ou uma mais elaborada, como javascript.

O importante é estar atento no momento de realizar essas atividades para compreender os códigos e não apenas copiando conteúdos, pois uma boa base assimilada de programação permite que níveis mais avançados de programação sejam alcançáveis muito mais facilmente.

Como é possível observar, os códigos estão bem divididos e cada parte gera um arquivo separado que representa uma página do website. Esses arquivos separados são necessários para gerar um site de maior porte, porém, em hotsites, é interessante agrupar todo código e conteúdos em uma página única.

Outro ponto importante foi a definição de um layout simples no estilo “flat design” para simplificar compreensão e leveza das páginas, permitindo que qualquer dispositivo seja capaz de reproduzir a maior parte dos conteúdos sem dificuldade.

Esse layout simplificado também favorece a compatibilidade do código em navegadores diversos e sistemas operacionais variados em que certos detalhes de páginas mais elaboradas podem não ser visíveis.

Foi mantido um padrão nos códigos das páginas para mostrar que códigos padronizados facilitam sua manutenção e compreensão de como foi estruturado uma página. Isso melhora mudanças que precisem ser feitas após o código estar pronto, em uma etapa da programação em que ocorrem as evoluções no código para acréscimo de novos conteúdos.

No caso de websites que precisam de atualização regular, essa manutenção no código é mais constante e pode afetar mais de um arquivo, necessitando que um mesmo trecho de código seja alterado em vários arquivos, mostrando a vantagem da padronização na codificação.

Fonte: o autor.





LIVRO

Web Design Responsivo

Maurício Samy Silva

Editora: Novatec Editora

Sinopse: Livro que trata do design responsivo, detalhando métodos e tecnologias. Linguagem corrente e clara, evitando termos técnicos avançados, quando possível, deixam a leitura mais tranquila.

Todas as explicações são acompanhadas por conteúdo HTML para melhor compreensão e testes visando a criação de sites acessíveis e agradáveis em qualquer dispositivo móvel ou não.

Comentário: Websites responsivos têm a vantagem de poder ajustar seu conteúdo de acordo com o tamanho do display em que é visualizado e uma tendência do mercado é a ampliação de conteúdo para dispositivos móveis e, com isso, um website único pode atender a essa demanda e de computadores tradicionais.



CONCLUSÃO

Caro(a) aluno(a), encerrando uma disciplina que permite o uso sem fim da criatividade, é possível perceber, ao longo das unidades, que é possível criar muito com o material aqui colocado, mas é possível criar muito ainda complementando seus conhecimentos com pesquisas e outras disciplinas.

A programação *Front-end* permite que todo o design e a parte visual de páginas web sejam pensados e planejados visando a melhor interatividade possível. O mais importante é saber adequar conteúdo e código para produzir sites cada vez mais elaborados e capazes de sempre agregar algo novo a cada projeto.

Existem sites com conteúdo altamente estilizado, com grande quantidade de efeitos que obtêm um resultado muito bonito. Em contrapartida, é preciso um computador não tão lento para auxiliar a renderização e execução dos scripts junto dos conteúdos multimídia.

Sites leves, baseados no modelo “*flat design*”, tendem a não ser tão exuberantes, mas acabam sendo as opções ideais para dispositivos móveis dotados de menos recursos de hardware e, muitas vezes, utilizando Internet de baixa qualidade para exibir conteúdos web.

Também conta que esses dispositivos, muitas vezes, utilizam planos com limitação de acesso em quantidade de bytes acessados, além da velocidade em bits por segundo. Isso é consumido muito rapidamente por sites pesados e conteúdos multimídia.

Há pouco tempo, *softwares* muito populares, como o *Macromedia Flash* (hoje, pertencente a Adobe), eram utilizados para a construção de sites inteiros, mas, hoje, estão restritos a atividades complementares, como desenvolvimento de animações.

Linguagens como ASP e PHP também perderam espaço com alternativas mais modernas, ou com recursos mais fáceis para produção mais rápida de sistemas web.

O mais importante é saber que a área de TI evolui e muda rapidamente, mas tudo relacionado à Internet muda mais rapidamente; o que se aprende hoje não será, de forma alguma, perene.



REFERÊNCIAS

ALVES, M. N. **Mídia e produção audiovisual:** uma introdução. Curitiba-PR: Intersaberes, 2012.

CHAK, A. **Como Criar sites Persuasivos:** clique aqui. São Paulo: Pearson Education do Brasil, 2004.

FREEMAN, E.; FREEMAN, E. **Use a Cabeça (Head First):** HTML com CSS e XHTML. 2. Ed. Rio de Janeiro: Alta Books, 2008.

JARVIS, J. **O que a Google faria?** – Como atender às novas exigências do mercado. Barueri-SP: Manole, 2010.

LEMAY, L. **Aprenda a Criar Páginas Web com HTML e XHTML em 21 dias.** São Paulo: Pearson Education do Brasil, 2002.

MANZANO, J. A. N. G. **Guia de orientação e desenvolvimento de sites HTML, XHTML, CSS e JavaScript.** São Paulo: Érica, 2008.

SHARMA, R.; SHARMA, V. **Desenvolvendo Sites de e-Commerce.** São Paulo: Makron Books Ltda, 2001.

SILVA, M. S. **Construindo sites com CSS e (X)HTML:** sites controlados por folhas de estilo em cascata. São Paulo: Novatec Editora, 2008.

SILVA, O. J. **JavaScript avançado:** animação, interatividade e desenvolvimento de aplicativos. São Paulo: Érica, 2003.

YANK, K.; ADAMS, C.; TORTELLO, J. E. N. **Só Javascript:** tudo o que você precisa saber sobre Javascript a partir do zero. Porto Alegre: Bookman, 2009.



GABARITO

UNIDADE I

1. a
2. b
3. F, V, F
4. Plagiar conteúdos é apropriar-se deles sem citação do autor e nem fonte de pesquisa, para indicar a procedência dos conteúdos. Cópia não autorizada de trechos ou textos completos escritos por alguém ou publicados em algum lugar.
5. Textos inéditos não são frutos de plágio, imagens diversas produzidas ou fotos tiradas sem violação de direitos de imagem ou propriedade, ou outros conteúdos multimídia produzidos da forma mais personalizada possível.

UNIDADE II

1. HTML inicia toda a codificação da página e BODY se refere ao conteúdo a ser exibido na página, ou seja, BODY está contido em HTML.
2. A
3. E
4. B
5. `<!DOCTYPE HTML>`
`<html lang="pt-br">`
`<head>`
`<meta charset="UTF-8">`
`<title>Mensagem</title>`
`</head>`
`<body>`
`</body>`
`</html>`



GABARITO

UNIDADE III

1. B.
2. C.
3. V, V, V, F, F.
4. <audio autoplay>
 <source src="musica.mp3" type="audio/mpeg">
 </audio>
5. body {
 color: blue;
 font-family: "Arial";
 font-size: 20px;
 }

UNIDADE IV

1. B
2. D
3. (V) (V)
 (V) (F)
 (V) (F)
4. <p onmouseover="style.color='green'">TEXTO</p>
5. \$(document).ready(function(){
 \$("#button").click(function(){
 \$("#div1").fadeOut("slow");
 \$("#div2").fadeOut(3000, 0.5);
 });
 });



GABARITO

UNIDADE V

1. B.

2. D.

3. V, F, F, V, V.

4. p {
 font-size: 12px;
 color: red;
}

5. LINK

