

FACULDADE SENAC GOIÁS
CURSO DE GESTÃO DE TECNOLOGIA DA INFORMAÇÃO
FUNDAMENTOS DE TECNOLOGIA DA INFORMAÇÃO

JOÃO NETTO PINHEIRO
JOÃO PAULO NASCIMENTO OLIVEIRA
PAULO ROBERTO VIEIRA

HTML 5

Professora: Kelly Alves Martins de Lima

GOIÂNIA
2018

Sumário

Introdução ao HTML 5.....	3
2 Vantagens e desvantagens do uso da linguagem HTML5	4
2.1 Vantagens	4
2.2 Desvantagens	4
3 Tipos de aplicações que possuem a linguagem HTML5 como base.....	5
4 Elementos Semânticos: Header, Nav, Section, Article e Footer.....	6
Referências	9

Introdução ao HTML5

Como muitas outras coisas na área de TI, o HTML surgiu da necessidade de padronizar o formato de comunicação web, pois, até determinada época, cada browser tinha seu padrão de código. Então surgiu o projeto Web Standards Project, este tinha o objetivo claro de fazer com que os browsers e desenvolvedores seguissem os padrões da W3C (World Wide Web Consortium). Conforme explica (EIS e FERREIRA, 2012).

Quando a web foi criada, era necessário distribuir a informação de uma maneira simples, mas organizada, era necessário então haver uma linguagem que fosse entendida universalmente por diversos meios de acesso. O HTML se propõe a ser esta linguagem. Desenvolvido originalmente por Tim Berners- Lee o HTML ganhou popularidade quando o Mosaic - browser desenvolvido por Marc Andreessen na década de 1990 - ganhou força. A partir daí, desenvolvedores e fabricantes de browsers utilizaram o HTML como base, compartilhando as mesmas convenções e seus conceitos.

Em português o significado da sigla é algo como, Linguagem de Marcação de Hipertexto (HyperText Markup Language), que é uma linguagem para publicar conteúdo para web.

HTML é baseado no conceito de hipertexto, organização de conteúdo não linear. Hipertexto, são conjuntos de elementos (palavras, imagens, vídeos, áudio, textos etc.) ligados por conexões essas conexões formam uma grande rede de informação. Para distribuir informação de uma maneira global, é necessário haver uma linguagem que seja entendida universalmente por diversos meios de acesso. O HTML se propõe a ser esta linguagem (EIS e FERREIRA, 2012).

Enquanto o W3C focava para criar a segunda versão do XHTML, um grupo chamado Web Hypertext Application Technology Working Group ou WHATWG trabalhava em uma versão do HTML que trazia mais flexibilidade para a produção de websites e sistemas baseados na web. O WHATWG foi fundado por desenvolvedores de empresas como Mozilla, Apple e Opera em 2004. Eles não estavam felizes com o caminho que a Web tomava e nem com o rumo dado ao XHTML. Por isso, estas organizações se juntaram para escrever o que hoje é chamado de HTML5 (W3C BRASIL, 2010).

2 Vantagens e desvantagens do uso da linguagem HTML5

2.1 Vantagens

Um dos principais objetivos do HTML5 é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final. O HTML5 também cria novas tags e modifica a função de outras. Nele é possível modificar a forma de como escrevemos código e organizamos a informação na página. Seria mais semântica com menos código, mais interatividade sem a necessidade de instalação de plugins e perda de performance. É a criação de código interoperável, pronto para futuros dispositivos e que facilita a reutilização da informação de diversas formas (EIS e FERREIRA, 2012, p.30).

Outro ponto que é considerando uma vantagem no HTML5 é o uso de API, um exemplo que podemos citar é o Canvas, este permite desenhar na tela do navegador utilizando javascript, ferramenta utilizada para criação de conteúdo gráfico (`<canvas width="300" height="225"></canvas>`).

Padronização, aumenta a penetração nos mais diversos dispositivos. As versões antigas do HTML não tinham um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, sidebar, menus. Não havia um padrão de nomenclatura de IDs, Classes ou tags. Também não havia um método de capturar de maneira automática as informações nos rodapés dos websites (EIS e FERREIRA, 2012, p.30).

Velocidade, o HTML5 permite navegar na web de forma mais rápida e simples, isso melhora a performance e usabilidade nos sites.

Mídia, aqui podemos citar dois detalhes, a capacidade de gerenciar arquivos de mídias (sons, vídeos) é um diferencial do HTML5 e esse gerenciamento elimina a necessidade do usuário de fazer instalações de plugins.

Investimento, por ter várias empresas grandes e líderes de mercados em suas áreas, leva a crer que grande parte dos problemas serão resolvidos, se é que já não foram.

2.2 Desvantagens:

Uma das desvantagens é a retrocompatibilidade com versões antigas de browsers como o Internet Explorer. O que é um problema sério para aplicações web baseadas em tecnologias mais recentes.

A modulação na construção do HTML5 e CSS é um problema ou desvantagem, pois desenvolvedores e fabricantes de browser não trabalham de forma harmônica, por exemplo as propriedades do CSS mesmo são divididas em 3 grupos de trabalho, e cada um desses grupos podem lançar suas novidades, o bom é que o desenvolvimento ficou dinâmico, mas pode ocorrer problemas de compatibilidades com uma frequência maior, tudo isso explicado por Eis e Ferreira (2012, p.31).

O HTML5 é algo em constante desenvolvimento e aprimoramento, isso pode ser visto com um ponto negativo, considerando que não está “pronto”.

3 Tipos de aplicações que possuem a linguagem HTML5 como base

Conforme descrito no site da W3School as APIs são conhecidas como DOM Level 0. As APIs formam a base da arquitetura da web. Existe mais de uma dezena de APIs, as próprias ao HTML5 estão relacionadas as funcionalidades do escopo da página, mas existem outras que trabalham com funcionalidades tipo o armazenamento de dados e manipulação de arquivos, essas são mais complexas, essa variedade se deve ao fato de algumas serem desenvolvidas pelo W3C outras WHATWG e algumas desenvolvidas pelos dois de forma conjunta, tudo isso visando melhorias e correções em erros, citaremos a seguir algumas dessas APIs, vale ressaltar que se tratando de APIs, estamos falando de melhorias nas tecnologias web e não somente de HTML5.

Elemento device e Stream API, um elemento device é para inserir no html um elemento de acesso à webcam do usuário (<device type="media">). A função de update recebe um parâmetro stream. Trata-se de um objeto da classe Stream, que possui uma propriedade url, e um método record. O método record inicia a gravação do stream e retorna um objeto StreamRecorder. Esse último possui um método stop, que retorna o arquivo que foi gravado.

API Storage a interface de armazenamento da Web Storage API fornece acesso ao armazenamento de sessão ou armazenamento local para um domínio específico, permitindo que você, por exemplo, adicione, modifique ou exclua itens de dados armazenados

API de Drag and Drop As interfaces de Drag and Drop (arrastar e soltar) habilitam aplicações a usar funcionalidades de arrastar e soltar através do navegador. Por exemplo, com essas funcionalidades, o usuário pode selecionar

elementos arrastáveis (draggable) com o mouse, arrastar elementos até um elemento soltável (droppable), e soltar o elemento ao soltar o botão do mouse.

Server-Sent Events API é quando uma página da Web recebe automaticamente atualizações de um servidor. Isso também era possível antes, mas na Server-Sent Events enviados pelo servidor, as atualizações vêm automaticamente.

A Microdata API, segundo Pilgrim, 2010.

Microdados giram em torno de vocabulários personalizados. Pense no “conjunto de todos os elementos da HTML5” como um vocabulário. Este vocabulário inclui elementos para representar uma seção ou um artigo, mas isto não inclui elementos para representar uma pessoa ou um evento. Se você deseja representar uma pessoa na página web, você vai precisar definir seu próprio vocabulário. Microdata deixa você fazer isso. Qualquer um pode definir um vocabulário microdata e começar a incorporar propriedades personalizadas em suas próprias páginas web.

Geolocation API há três maneiras famosas de um agente descobrir sua posição (Geolocalização IP, Triangulação GPRS, GPS). Para obter a posição do usuário, basta executar o script: [navigator.geolocation.getCurrentPosition\(showpos\)](#).

Aplicações web offline podem ser acessados através do Caching. O HTML5 indica ao navegador quais elementos são necessários e devem ser colocados em cache para que uma aplicação funcione offline. Através do javascript pode-se acessar o objeto ApplicationCache que controla o status e operações de caching na página.

4 Elementos Semânticos: Header, Nav, Section, Article e Footer

Basicamente são elementos que juntam grupos de textos no documento. Este é ponto que houve maior mudança na introdução do HTML5, pois foi onde inseriu mais tgs novas, estas vieram para melhorar a semântica dos elementos estruturais do código.

O elemento **HEADER** define os elementos de cabeçalho. O grande diferencial deste elemento, é que não é apenas utilizado para determinar o cabeçalho da página e sim para qualquer característica de cabeçalho, deve-se cuidar e não confundir com a tag <head> que serve para informar características da página. O Header é quem vai encabeçar uma região do site.

O elemento **NAV** é responsável pelas listas de navegação, representa uma seção da página que contém links para outras partes do website. Mas nem todos os grupos de links são elementos nav. Estes são aqueles considerados links mais importantes dentro do site, por exemplo o menu principal, em caso de portais aqueles são aqueles menus laterais (EIS e FERREIRA, 2012). De forma bem resumida Eis e Ferreira, p. 43, explicam o seguinte: Com a tag nav, há uma indicação de que aquele grupo é uma seção (nav é um tipo de section. Enquanto a tag section serve para indicar seções no site, a tag nav indica que um determinado grupo é uma seção de navegação) é um bloco de navegação. Convém ressaltar que nav não carrega qualquer tipo de link, ele irá carregar grupos de links ligados ao site, mais conhecidos como links internos.

A tag **SECTION** ou elemento Section define uma nova seção genérica no documento. Por exemplo, a home de um website pode ser dividida em diversas seções: introdução, destaque, novidades, informação de contato e chamadas para conteúdo interno (EIS e FERREIRA, 2012). Este substitui o DIV em muitos momentos, o uso de Section permite que os sistemas de buscas ou outras aplicações saibam que cada um dos “blocos” trata de um assunto diferente. É bom ressaltar que o section apenas divide os assuntos, não é como Header, Footer, Aside, Nav e Article que dividem informações específicas, cada um destes tem sua função específica na página.

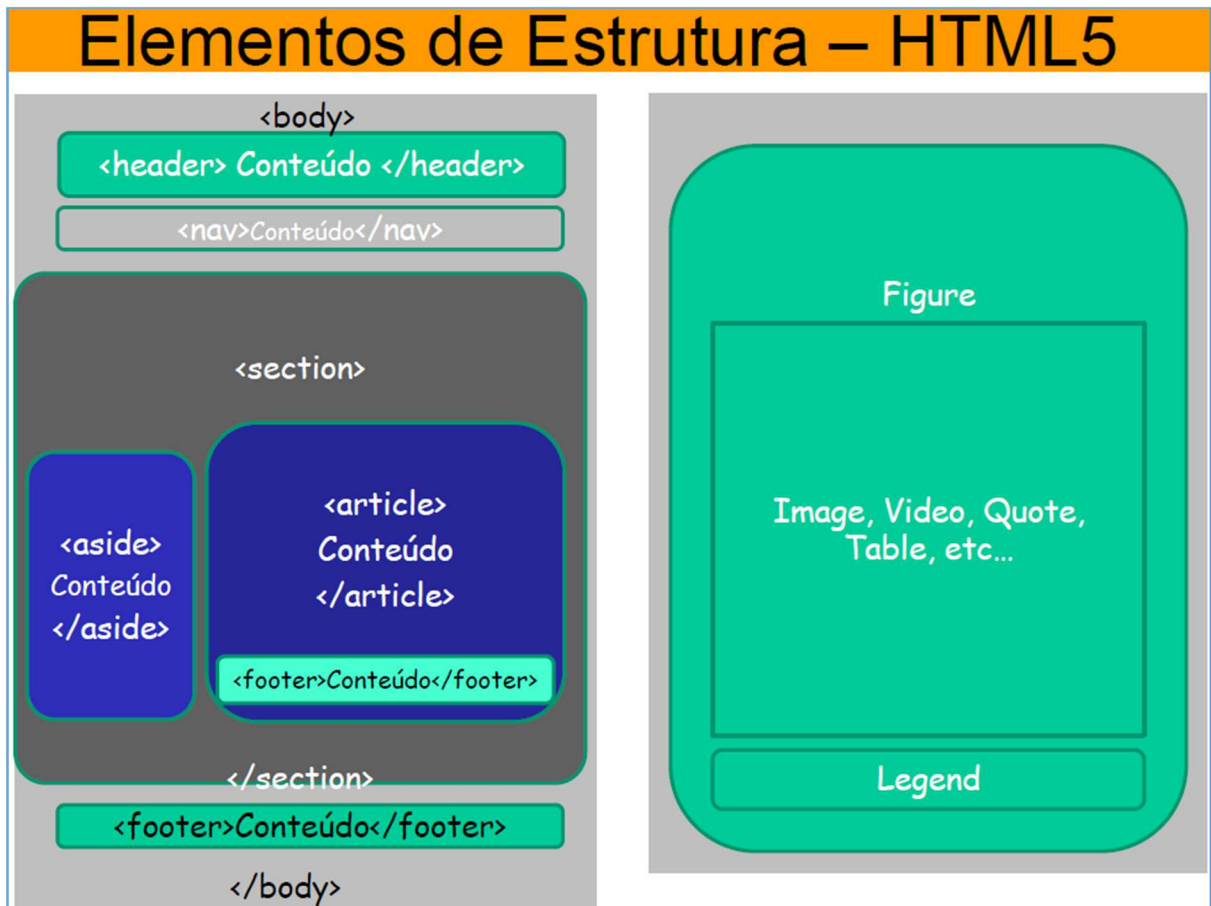
O elemento **ARTICLE**, neste é onde se inseri o texto ou a informação principal, neste elemento é onde será colocado informações relacionados ao artigo, tais como data de publicação, título, autor, o próprio texto, informações de artigos relacionados. Você já recebeu um feed de notícias? Se sim tenha em mente que o texto recebido é o conteúdo do elemento article. Os autores alertam para o seguinte ponto:

o article é um elemento mais específico que o section e o div. O article marca o texto principal do site. É o filé da página. O section é apenas um bloco de separação de assuntos diferentes. O div, o mais genérico de todos, é apenas aplicado para separar elementos em blocos, por isso ele não carrega nenhum significado semântico (EIS e FERREIRA, 2012, p.46).

O elemento **FOOTER** representa literalmente o rodapé da página. Seria o último elemento do último elemento antes de fechar a tag HTML. O elemento footer não precisa aparecer necessariamente no final de uma seção. O funcionamento do

elemento footer é bem semelhante ao header, a diferença é que o footer é utilizado para determinar os elementos de rodapé, tanto da página como de outros elementos dentro da página (COSTA JÚNIOR e MAGALHÃES, 2013, p.4). Em seguida exemplificamos uma visualização das “posições” e aplicação de cada elemento em uma página HTML.

Imagem 1 - Slide: aula expositiva



Fonte: Alves (2018)

Referências

ALVES, Kelly. **Introdução ao HTML**. Goiânia: Senac, 2018. 38 slides, color.

COSTA JUNIOR, Djalma Gonçalves; MAGALHÃES, Willian Barbosa. **HTML5 E WEB SEMÂNTICA, A WEB COM SIGNIFICADO**. Paranavaí: Unipar, 2013.

EIS, Diego; FERREIRA, Elcio. **HTML5 e CSS3: com farinha e pimenta**. São Paulo: Tableless, 2012. 2019 p.

PILGRIM, Mark. **HTML5: Up and Running: Dive into the Future of Web Development**. California: O'reilly, 2010. 222 p.

W3C BRASIL (Brasil). **HTML5: Curso W3C Escritório Brasil**. 2010. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 18 maio 2018.