

# HTML5

Emanuele Baron<sup>1</sup>, Roger Bezerra<sup>1</sup>

<sup>1</sup>Universidade do Estado de Santa Catarina - UDESC  
Centro de Ciências Tecnológicas - CCT. Joinville - SC - Brasil CEP: 89.219-710

emanuele.bbae@gmail.com, makotocorp@hotmail.com

**Abstract.** *HTML is a hypertext markup language, which launched its fifth version in 2012. This new release brings important changes to the role of HTML in the world of web, through new features such as semantics, new elements for forms, new types and elements of audio and video. In this way the new functionalities facilitate the construction, maintenance and understanding of an HTML page.*

**Resumo.** *O HTML é uma linguagem de marcação de hipertexto, a qual lançou em 2012 a sua quinta versão. Esta nova versão traz importantes mudanças quanto ao papel do HTML no mundo da web, através de novas funcionalidades como semântica, novos elementos para formulários, novos tipos e elementos de áudio e vídeo. Dessa forma as novas funcionalidades facilitam a construção, manutenção e a compreensão de uma página HTML.*

## 1. Introdução

O HTML (*Hyper Text Markup Language*) é uma linguagem de marcação para publicação de conteúdo (texto, imagem, vídeo, áudio e etc) na *web*. É a primeira camada do desenvolvimento *client-side*, responsável por organizar e formatar páginas desde que foi introduzido à internet no início de 1990.

Desde então uma das grandes deficiências da *web* é a dependência de *plug-ins* para a reprodução de conteúdos multimídias. Pensando nisso e em outras necessidades, foi desenvolvido o HTML5, o qual apresenta novos elementos, atributos, entre outros. Tendo como um dos principais objetivos facilitar, para o desenvolvedor, a manipulação das características dos objetos de maneira que seja transparente para o usuário final[4].

O presente trabalho encontra-se organizado na seguinte estrutura. Na seção 2 é apresentado a história do HTML. Na seção 3 é apresentado a compatibilidade e na seção 4 é apresentado as características da linguagem HTML5, bem como sua estrutura básica, novos elementos e atributos, por fim os elementos de áudio e vídeo. Na seção 4 são descritas as conclusões obtidas com base no conteúdo apresentado nas seções anteriores.

## 2. A História do HTML

O HTML foi desenvolvido por Tim Berners-Lee e ganhou popularidade na década de 1990. Entre 1993 e 1995, o HTML ganhou as versões HTML+, HTML2.0 e HTML3.0 e em 1997 a W3C (*World Wide Web Consortium*), responsável por manter os padrões do código, trabalhou nas versões HTML2.0, HTML3.2, HTML4.0 e HTML4.01. Desde o início o HTML foi criado para ser uma linguagem independente de plataformas, *browsers* e outros meios de acesso[4].

No ano de 2007 a W3C, juntamente com a WHATWG (*Web Hypertext Application Technology Working Group*) desenvolveram uma nova versão, o HTML5. Após vários anos de trabalho a versão final foi apresentada em 2012.

### 3. Compatibilidade

Há uma grande diversidade de dispositivos que acessam a *internet*, entre eles, há uma série de *tablets*, *smartphones*, computadores e etc. Cada um destes meios de acesso utilizam um determinado *browser* para navegar na *web*.

Não há como os desenvolvedores manterem um bom nível de compatibilidade com todos estes *browsers* levando em consideração a particularidade de cada um. Uma maneira mais segura de manter o código compatível, é nivelar o desenvolvimento pelos motores de renderização. Cada *browser* utiliza um motor de renderização que é responsável pelo processamento do código da página[4].

Na figura 1 é mostrado os motores com seus respectivos navegadores, o motor *Webkit* por exemplo é o motor de renderização que tem maior compatibilidade com o HTML5 e CSS3. O mesmo é derivado do motor KHTML, do navegador Konqueror (facilmente encontrado nas distribuições Linux que utilizam a interface KDE).

**Figura 1. Motores de renderização.**

<i>Motor</i>	<i>Browser</i>
<i>Webkit</i>	Safari, Google Chrome
<i>Gecko</i>	Firefox, Mozilla, Camino
<i>Trident</i>	Internet Explorer 4 ao 9
<i>Presto</i>	Opera 7 ao 10

Fonte: ([10]).

É importante que desenvolvedores foquem em fazer códigos compatíveis com estes motores, visto que focando a compatibilidade nos motores de renderização é possível atingir uma amplitude maior de *browsers*. Por exemplo, se o código desenvolvido funcionar no *Webkit*, alcançará o Safari e o Chrome, dois dos principais *browsers* do mercado para *desktops*. Além disso, também alcança aparelhos como Blackberry, iPhone, iPod Touch, iPad e dispositivos que rodam Android[1].

Atualmente o *Chrome* é o *browser* com maior compatibilidade, seguido do Firefox e o Opera que já estão compatíveis com grande parte da especificação do HTML5. Apenas as versões antigas da *Internet Explorer*, como a versão 8 e 7 não tem quase nenhum suporte a linguagem. O que é um problema para aplicações web baseadas em tecnologias mais recentes, pois muitos usuários ainda utilizam as versões antigas do *Internet Explorer*. Na figura 2 é apresentado uma tabela simples de compatibilidade entre os *browsers* e alguns módulos do HTML5.

**Figura 2. Tabela de compatibilidade.**

	Safari	Chrome	Opera	Firefox	IE 8	IE 9
Local Storage	s	s	s	s	s	s
Histórico de Sessão	s	s	s	s	s	s
Aplicações Offline	s	s	n	s	n	n
Novos tipos de campos	s	s	s	n	n	n
Form: Autofocus	s	s	s	n	n	n
Form: Autocomplete	n	n	s	n	n	n
Form: Required	s	s	s	n	n	n
Video, Audio e Canvas Text	s	s	s	s	n	s

Fonte: ([11]).

Pelo fato de haver usuários que ainda utilizam *browsers* desatualizados ou *browsers* que não suportam todas as características do HTML5, foi desenvolvida bibliotecas que permitem a verificação do suporte da maioria das propriedades e funcionalidades da linguagem. Isso permite aos desenvolvedores testarem algumas das novas tecnologias e, em seguida, fornecer avisos para os navegadores que ainda não as suportam. Um exemplo de biblioteca Modernizr.

## 4. O HTML5

O padrão HTML5 é uma nova versão do HTML4, a qual introduz um conjunto de novos elementos que facilitam a estruturação e desenvolvimento *front-end*, capaz de fornecer uma informação qualitativa sobre os diferentes elementos da página[8].

Esta versão também trás um vasto suporte e ferramentas para a utilização de CSS e *JavaScript*, fazendo com que as aplicações ou *web sites* se tornem mais leves e práticos sem necessitarem da criação de grandes *scripts*. Inclusive houve a criação de novas *tags* e implementação de novas funções nativas da linguagem, como: visualizar vídeos, áudio e renderização de gráficos. Não havendo mais a necessidade da instalação de *plug-ins* externos para a execução de tarefas, viabilizando assim uma melhora significativa na performance.

### 4.1. Estrutura Básica

A estrutura básica do HTML5 é formada pelo *Doctype* ou DTD (*Document Type Definition*), o atributo *lang* e a *tag head*. O *Doctype* deve ser a primeira linha de código do documento antes da *tag HTML*. Ele não é uma tag do HTML, mas uma instrução para que o *browser* tenha informações sobre qual versão de código a marcação foi escrita[4]. Em versões anteriores, era necessário referenciar o DTD diretamente no código do Doctype, com o HTML5 a referência por qual DTD irá utilizar é responsabilidade do *browser*.

O atributo *lang* identifica a linguagem principal na qual o conteúdo do documento será escrito. E a *tag head* contém, em geral, informações sobre a página (como título e me-

tadados), além de chamadas para outros arquivos, como documentos CSS e JavaScript[9]. Metadado que é definido dentro da *tag head* refere-se as informações sobre a página, como a codificação. Recomenda-se utilizar a codificação UTF-8 pelo fato de ser um dos mais abrangentes, permitindo a visualização correta da página em qualquer plataforma, programa e idioma. Na figura 3 é apresentada a estrutura básica do HTML5, formado por instrução e *tags* explicadas acima.

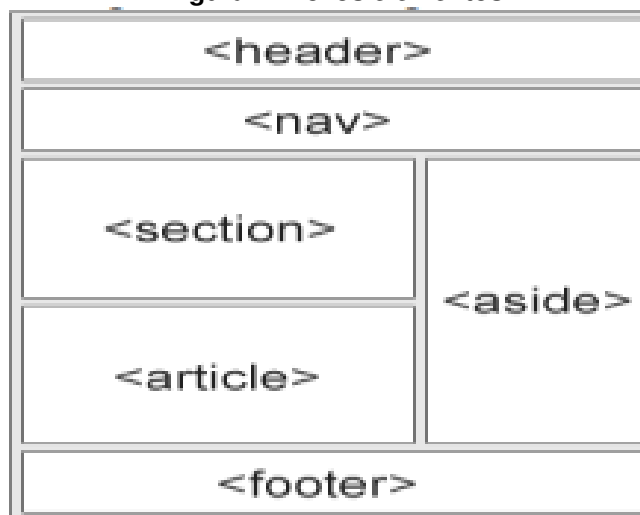
**Figura 3. Estrutura básica HTML5**

```
<!doctype html>
<html lang= "pt-BR">
<head>
  <meta charset= "UTF-8">
  <title>Pagina HTML5 Basica</title>
</head>
<body>
  <!-- Conteúdo da página -->
</body>
</html>
```

## 4.2. Novos Elementos

O HTML5 trouxe uma série de elementos que ajudam a definir um padrão universal para a criação de seções comuns no documento HTML. Com a ajuda destes elementos pode-se por exemplo diferenciar diretamente pelo código áreas importantes do site como *sidebar*, rodapé, cabeçalho, menu entre outros[4]. A figura 4 representa o posicionamento destes elementos em uma página web.

**Figura 4. Novos elementos**



Fonte: ([2]).

O elemento *header* possibilita a criação de um cabeçalho no documento, já o elemento *nav* representa uma seção da página que contém link para outras páginas ou partes do mesmo *website*. O *nav* pode estar presente em todos os elementos do HTML, como no *header* para definir menus, nos portais, *article* e entre outros. Na figura 5 é apresentado um exemplo de código da utilização do *nav*.

Figura 5. Exemplo da utilização do elemento nav.

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

<nav>
  <a href="/html/">HTML</a>
  <a href="/html/">HTML5</a>
  <a href="/css/">CSS</a>
  <a href="/js/">JavaScript</a>
</nav>

</body>
</html>
```

O elemento *section* define um bloco ou um grupo de um assunto específico, agrupa diversos elementos que tenham relação entre si. Por exemplo, a página da Globo.com, os assuntos são divididos em várias seções como, notícias gerais, esporte e entretenimento. Como mostra a figura 6, cada uma das colunas aborda um tema específico.

Figura 6. Exemplo da utilização do elemento section



O elemento *aside* define um bloco de informações referentes ao conteúdo principal, podemos utilizá-lo como citações ou sidebars, agrupamento de publicidade ou grupos de navegação. Este elemento nos permite criar, por exemplo, um menu lateral com os grupos de informações do site[7]. O elemento *article* é um bloco de texto onde é definido um conteúdo independente, pode ser um artigo, um bloco de comentários de usuários ou apenas um bloco de texto comum. E o *footer* representa literalmente o rodapé da página, na figura 7 é apresentado um exemplo de implementação abordando todos os novos elementos citados nesta seção.

Os *designers* e desenvolvedores podem agora utilizar esses novos elementos do HTML5 para identificar o cabeçalho, o rodapé, a barra de navegação e outras seções que antes eram marcadas com o elemento *div*[5]. O que gerava elementos genéricos, sem valor semântico específico. Como mostra a figura 8, o qual todos as seções eram chamadas de

Figura 7. Exemplo da utilização da nova semântica do HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "UTF-8">
    <title> Título da Página</title>
  </head>
  <body>

    <header> Cabeçalho </header>
    <nav> Menu </nav>
    <section> Título do Conteúdo
    <article> Conteúdo </article>
    </section>
    <aside> Barra lateral </aside>
    <footer> Roda Pé </footer>

  </body>
</html>
```

div, com isso ocasionava falta de padrão no HTML. O elemento *div* servirá no HTML5 para agrupar elementos dentro das *sections* ou fazer outros detalhes que não precisam de significado semântico, apenas visual.

Figura 8. Exemplo da utilização do div nas versões anteriores do HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "UTF-8">
    <title> Título da Página</title>
  </head>
  <body>
    <div id = "header">
      Cabeçalho
    <div id = "navbar"> Menu </div>
    </div>
    <div id = "content"> Conteúdo </div>
    <div id = "sidebar"> Barra lateral </div>
    <div id = "footer "> Roda Pé </div>
  </body>
</html>
```

Outro elemento novo no HTML5 é o canvas, poderosa ferramenta que permite desenhos de textos, edição, sombras, inclusão de imagens, manipulação de *pixels*, rotação e transformação de objetos[6]. Um importante elemento que viabiliza desenhar na tela do navegador via JavaScript, sendo o único que possibilita este tipo de função.

#### 4.3. Novos Tipos e Atributos

Com o HTML5 novos tipos e atributos à elementos que já existentes foram criados, como mostra figura 9. Um dos atributos mais relevantes é o *required*, o qual permite criação de formulários sem a necessidade de scripts externos para validação das informações. Atributos como autofocus, *placeholder*, *pattern*, *novalidate* e *formnovalidate* também facilitam a criação de aplicações web baseadas em formulários. A seguir será explicado cada um deles.

**Figura 9. Novos tipos e atributos**

New Input Types	New Input Attributes
<ul style="list-style-type: none"><li>• color</li><li>• date</li><li>• datetime</li><li>• datetime-local</li><li>• email</li><li>• month</li><li>• number</li><li>• range</li><li>• search</li><li>• tel</li><li>• time</li><li>• url</li><li>• week</li></ul>	<ul style="list-style-type: none"><li>• autocomplete</li><li>• autofocus</li><li>• form</li><li>• formaction</li><li>• formenctype</li><li>• formmethod</li><li>• formnovalidate</li><li>• formtarget</li><li>• height and width</li><li>• list</li><li>• min and max</li><li>• multiple</li><li>• pattern (regexp)</li><li>• placeholder</li><li>• required</li><li>• step</li></ul>

Fonte: [2].

O atributo *pattern* permite definir expressões regulares de validação, sem *JavaScript*. Podem haver situações em que não haja necessidade de um formulário ser validado, nestes casos basta incluir no elemento *form* o atributo *novalidate*. Outra situação comum é querer que o formulário não seja validado dependendo da ação de *submit*, nestes casos pode-se usar no botão de *submit* o atributo *formnovalidate*[3].

O atributo *autofocus* pode ser especificado no *textarea*, *select* e *button*. O qual tem como finalidade manter o foco no campo automaticamente ao carregar a página. Já o *placeholder* é um atributo de uso exclusivo com os elementos *input* e *textarea* e destina-se a definir uma dica de preenchimento do campo. O atributo *required* é para uso exclusivo com os elementos *input*, *select* e *textarea* e destina-se a marcar um campo de formulário como obrigatório. Na figura 10 é mostrado o código de um formulário utilizando os atributos *placeholder* e *required*, utilizando CSS. Na figura 11 é apresentado o resultado da figura 10, o formulário possui os campos Nomes, Sobrenome e Cidade. Como o campo Sobrenome não foi preenchido então é realizada a validação através do atributo *required* solicitando o preenchimento.

Na figura 12, no lado esquerdo é possível visualizar os novos tipos do HTML5 que facilitam a manipulação dos dados. Um dos tipos é o *date*, outro exemplo é o tipo *range*, o qual possibilita definir o intervalo desejado com as variáveis *min*.e *max*., como mostra a figura 13.

Figura 10. Código de um formulário utilizando os novos atributos.

```
<!DOCTYPE html>
<html>
<style>
input[type=text], select {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

input[type=submit] {
    width: 100%;
    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
}

section {
    border-radius: 5px;
    background-color: #f2f2f2;
    padding: 20px;
}

h3{
    text-align: center
}
</style>
<body>
<h3>Validação utilizando Required</h3>

<section>
  <form>
    <label for="nome">Nome</label>
    <input type="text" id="nome" name="nome" placeholder="Seu nome.." required>

    <label for="sobrenome">Sobrenome</label>
    <input type="text" id="sobrenome" name="sobrenome" placeholder="Seu sobrenome.." required>

    <label for="cidade">Cidade</label>
    <input type="text" id="cidade" name="cidade" placeholder="Sua cidade.." required>

    <input type="submit" value="Enviar">
  </form>
</section>
</body>
</html>
```

Figura 11. Resultado do código apresentado na figura 10.

Nome

Emanuele

Sobrenome

Seu sobrenome..

Cidade

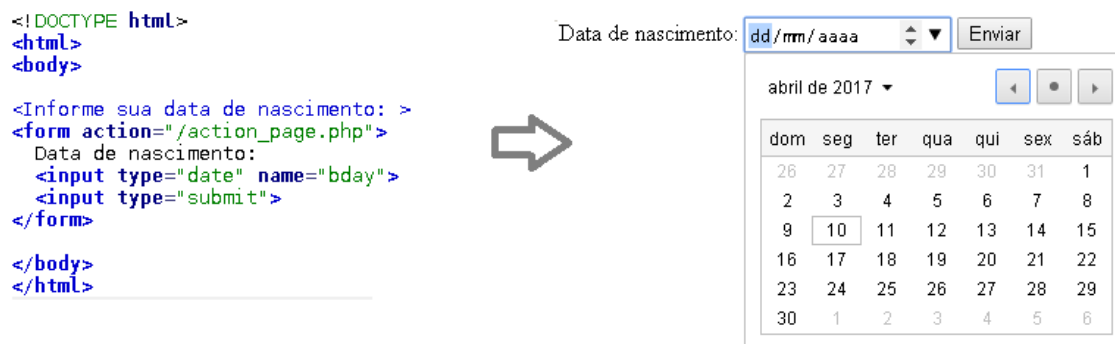
Joinville

Preencha este campo.

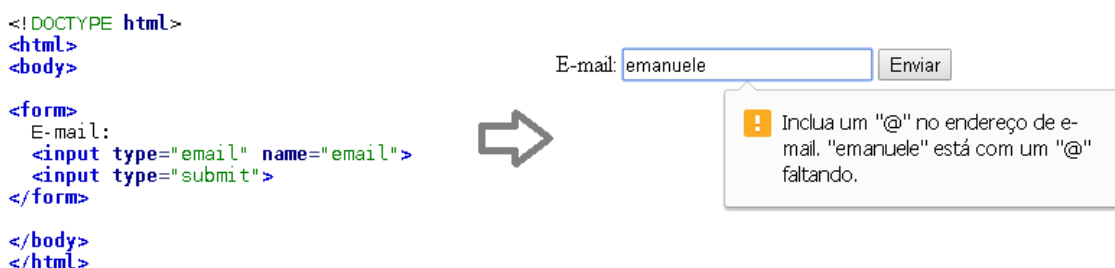
Enviar



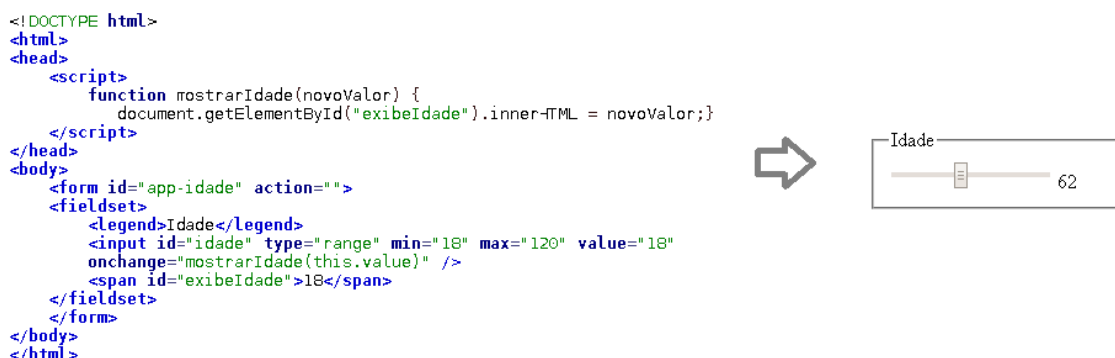
**Figura 12.** No lado esquerdo da imagem é apresentado o código e ao lado o resultado da implementação do tipo *date*.



**Figura 13.** Implementação do tipo email, o qual faz a validação e informa ao usuário caso o email inserido esteja errado.



**Figura 14.** Implementação do tipo *range* utilizando Javascript para mostrar na tela o valor escolhido dentro do intervalo definido.



#### 4.4. Tags Inválidas

As tags que são apresentadas na figura 5 não são mais válidas no HTML5. Algumas foram removidas ou substituídas e outras foram retiradas pois são melhores controladas pelo CSS, como o atributo *align*, *background*, *bgcolor*, *basefont*, *s*, e uma das mais comuns, a tag *center*.

**Figura 15. Tags inválidas**

Tags excluídas	Removida ou Substituída
<acronym>	Substituída por <abbr>
<applet>	Substituída por <object>
<basefont>	CSS lida melhor com essa função
<bgsound>	Substituída por <audio>
<big>	CSS lida melhor com essa função
<blink>	Removida de HTML5
<center>	CSS lida melhor com essa função
<dir>	Substituída por <ul>
<font>	Removida de HTML5
<frame>	Removida de HTML5
<frameset>	Removida de HTML5
<isindex>	Substituída pelo uso explícito de <form>
<marquee>	Removida de HTML5
<multicol>	Removida de HTML5
<nobr>	Removida de HTML5
<noframes>	Removida de HTML5
<noscript>	Somente em conformidade com a sintaxe HTML
<s>	CSS lida melhor com essa função
<spacer>	Removida de HTML5
<strike>	CSS lida melhor com essa função
<tt>	CSS lida melhor com essa função
<u>	CSS lida melhor com essa função

Fonte: [?].

#### 4.5. Elementos de Áudio e Vídeo

Nesta seção serão apresentados os elementos de áudio e vídeo, os quais não necessitam da utilização de *plug-ins* externos, como *Flash*, *QuickTime* e Windows Media Player para a execução de conteúdo multimídia como era nas versões anteriores.

O áudio destina-se a incorporar um som, para inserir áudio em uma página web basta usar o elemento áudio e declarar o caminho para o arquivo de som no atributo src do elemento. O som é reproduzido com as funcionalidades nativas do navegador[11], como mostra a figura 16.

**Figura 16. Representação de um código utilizando o novo elemento de áudio**

```
<!DOCTYPE HTML>
<html>
<body>

<audio src="prasersincero.mp4" controls = "true" autoplay="true"> </audio>

</body>
</html>
```

O valor de *controls* define se um controle de áudio, com botões de *play*, *pause*, volume, barra de progresso, contador de tempo, etc., será exibido na tela. Se for colocado como falso, será preciso controlar o *player* via *javascript*. O valor de *autoplay* define se o áudio vai começar a tocar assim que a página carregar.[4]. O áudio suporta os seguintes formatos: oga, mp3 e wma.

O elemento video trabalha com três formatos principais, sendo: ogv, mp4 e wma. O uso do vídeo é muito semelhante ao de áudio, como mostra a figura 17. A utilização destas *tags* é importante a inclusão do atributo *source*, o qual destina-se para disponibilizar vídeo ou áudio codificados em formatos diferentes para navegadores diferentes. Como pode ser observado na figura 18, o qual representa o elemento de vídeo.

**Figura 17. Representação de um código utilizando o novo elemento de vídeo**

```
<!DOCTYPE HTML>
<html>
<body>

<video src="video.ogv" width="400" height="300"> </video>

</body>
</html>
```

**Figura 18. Utilização do atributo source.**

```
<!DOCTYPE HTML>
<html>
<body>

<video controls="true" autoplay="true" width="400" height="300" >

<source src="u.ogv">

<source src="u.mp4">

<p> Faça o <a href="u.mp4" >download do vídeo </a></p>
</video>

</body>
</html>
```

A figura 18 é reproduzido o vídeo ogv em navegadores que suportam o formato ogv, se o navegador não suportar ogv, o navegador reproduz o arquivo mp4.

## 5. Conclusão

O HTML5 sem dúvidas é a evolução do HTML4, a qual corrige problemas de sua versão anterior e principalmente traz muitas melhorias à linguagem, pois oferece facilidade na manipulação de elementos, mais funcionalidades, independência de plataforma, tratamento de exceção e suporte nativo aos conteúdos multimídias.

O suporte nativo aos conteúdos multimídias é um dos recursos que recebem mais atenção no HTML5. Sendo possível inserir áudios e vídeos em uma página sem usar componentes externos (como *plugins*). Isso faz com que a navegação se torne mais rápida e simples, melhorando a performance da página e consequentemente melhorando a experiência do usuário.

O HTML5 ainda é uma tecnologia em desenvolvimento, porém vem sendo muito utilizada, programas como *Adobe Flash Player*, *Silverlight* vem sendo substituídos pelo HTML5, o que mostra que cada vez mais a linguagem está revolucionando o mundo da *web*.

## Referências

- [1] Html5 - análise do suporte atual pelo navegadores e estratégia de uso., 2015. Disponível em: <https://channel9.msdn.com/posts/Anlise-do-suporte.html> Acesso em: 20 de maio 2017.
- [2] W3schools, html5 tutorial, 2015. Disponível em: <http://www.w3schools.com> Acesso em: 03 de maio 2017.
- [3] Diego Eis and Elcio Ferreira. *HTML5 e CSS3 com farinha e pimenta*. Lulu.com, 2012.
- [4] Elcio Ferreira and Diego EIS. *Html5: Curso w3c escritório brasil*. São Paulo, 2011.
- [5] Fábio Flatschart. *HTML 5-Embarque Imediato*. Brasport, 2011.
- [6] Clodoaldo Hiroiti Iamashita and Willian Barbosa Magalhães. *Html 5, um estudo sobre seus novos recursos*.
- [7] Djalma Gonçalves Costa Junior and Willian Barbosa Magalhães. *Html5 e web semântica, a web com significado*.
- [8] Késsia Rita Martins, Guilherme Marchi. *Análise comparativa dos recursos e diferenças das tecnologias de programação html5 e html4*.
- [9] Guilherme Muller. Uma página de html básica, 2011. Disponível em: <http://guilhermemuller.com.br/pt/elearning/htmlcssbasico/licao/1/pagina.html> Acesso em: 30 de março 2017.
- [10] Kelvin Rodrigues. *Compatibilidade dos navegadores com o html5*, 2015. Disponível em: <http://fabrica.ms.senac.br/2014/10/compatibilidade-entre-os-navegadores-com-o-html5/> Acesso em: 24 de abril 2017.
- [11] Mauricio Samy Silva. *HTML5–2ª Edição: A linguagem de marcação que revolucionou a web*. Novatec Editora, 2014.