

CENTRO PAULA SOUZA



**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Desenvolvimento de Jogos
Digitais**

DESENVOLVIMENTO DE JOGOS DIGITAIS COM HTML 5

ALINE BOSSI

**Americana, SP
2011**

CENTRO PAULA SOUZA



**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Desenvolvimento de Jogos
Digitais**

DESENVOLVIMENTO DE JOGOS DIGITAIS COM HTML 5

ALINE BOSSI
alinebossi@hotmail.com

**Trabalho de Conclusão de Curso
desenvolvido em cumprimento à
exigência curricular do Curso Superior
de Tecnologia em Desenvolvimento de
Jogos Digitais, sob a orientação do
Prof. Me. Cleberson Eugenio Forte.**

Área: Jogos Digitais

**Americana, SP
2011**

BANCA EXAMINADORA

Prof. Me. Cleberson Eugenio Forte (Orientador)

Prof.^a. Dra. Me. Doralice de Souza Luro Balan

Prof. José William Pinto Gomes

AGRADECIMENTOS

Primeiramente, quero agradecer a meus pais, Odassir e Rafaela, meu irmão Fioravante e sua esposa Katiúscia, meu noivo Marcos e sua família, que com muito amor e carinho me auxiliaram nessa jornada, sempre acreditando em mim e em meu potencial. Por todas as horas de conversas e desabafos, com palavras suaves incentivando-me a superar os obstáculos e continuar em busca de meus objetivos.

Também agradeço a meu orientador Cleberson Forte por toda atenção e colaboração no desenvolvimento deste trabalho.

Aos meus amigos e colegas de classe, meus sinceros agradecimentos também, pois foram essenciais nesta jornada universitária.

E por fim, agradeço ao Ser supremo: Deus. Grata por me permitir ser o que sou hoje, por todas as dádivas e conquistas.

A todos, meus agradecimentos!

DEDICATÓRIA

*Dedico este trabalho à minha grande fonte de motivação,
minha família.*

“The Web does not just connect machines, it connects people.”

Tim Berners-Lee

RESUMO

O presente texto visa explorar as faces do desenvolvimento de documentos para a web amparados pela linguagem de marcação HTML. Com uma abordagem ampla, o estudo contido nesta pesquisa traz os principais aspectos no desenvolvimento de jogos *on-line*, expondo as primeiras tecnologias utilizadas nos jogos, comparando-as com o desenvolvimento dos mesmos utilizando HTML 5. Por fim, aplica-se o HTML 5 a um exemplo de projeto real (um jogo para a web).

Palavras Chave: HTML5, Aplicações web, Jogos Digitais.

ABSTRACT

This paper aims to explore the faces of the development of documents for the web supported by the HTML markup language. With a broad approach, the study contained in this research introduces the main aspects in the development of online gaming, exposing the first technologies used in games, comparing them with their development using HTML 5. Finally, apply the HTML 5 an example of a real project (a game for the web).

Keywords: HTML5, Web Applications, Digital Games.

SUMÁRIO

1 A LINGUAGEM HTML	14
1.1 W3C – <i>WORLD WIDE WEB CONSORTIUM</i>	16
1.2 WHATWG – <i>WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP</i>	17
2 O HTML 5	19
2.1 NOVAS CARACTERÍSTICAS DA LINGUAGEM	20
2.2 ESTRUTURA DO DOCUMENTO	21
2.3 ELEMENTOS ESTRUTURAIS.....	22
2.4 ELEMENTOS MULTIMÍDIA	24
2.5 GRÁFICOS NA WEB.....	25
2.5.1 CANVAS API	25
2.5.2 CANVAS E SVG	27
2.6 COMPATIBILIDADE DO HTML5	27
3 DESENVOLVIMENTO DE JOGOS COM HTML – JOGOS <i>ON-LINE</i>	30
3.1 JOGOS EM FLASH	36
3.2 JOGOS EM HTML 5	38
3.2.1 COMPLEMENTOS PARA O DESENVOLVIMENTO DE JOGOS COM HTML5	39
3.2.1.1 <i>ENGINES E FRAMEWORKS</i>	40
4 ESTUDO DE CASO: APLICANDO O HTML 5 EM UM PROJETO	42
4.1 CARACTERÍSTICAS DO JOGO E REQUISITOS BÁSICOS	42
4.2 DESENVOLVIMENTO	43
4.3 CODIFICAÇÃO	43

5 DISCUSSÃO.....	51
6 CONCLUSÃO.....	52
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	53

LISTA DE TABELAS E FIGURAS

Figura 1 - A evolução dos navegadores.....	18
Figura 2 - Estrutura básica de uma página em HTML4.....	20
Figura 3 - Estrutura básica de uma página em HTML5.....	20
Figura 4 - Meta tag utilizada em versões anteriores.....	21
Figura 5 - Estrutura básica de um documento feito com HTML5.....	21
Figura 6 - Elementos de marcação estruturais HTML5.....	24
Figura 7 - Exemplo de inclusão de áudio na página.....	24
Figura 8 - Exemplo de inclusão de vídeo na página.....	24
Figura 9 - Exemplo de inclusão do canvas no documento HTML.....	25
Figura 10 - Exemplo de utilização do contexto 2d para desenhar dois retângulos....	26
Figura 11 - Desenho utilizando o contexto 2d em canvas.....	26
Figura 12 - Código de verificação da API <i>Geolocation</i>	29
Figura 13 - <i>Star Trek</i> (COHEN, 2011).....	31
Figura 14 - <i>Mega War I</i> (KHOA, 2005).....	32
Figura 15 - <i>Air Warrior</i> no PC (HAWMAN; NOLAN, 2011).....	33
Figura 16 - <i>NeverWinter Nights</i> (KAO, 2009).....	33
Figura 17 - PacMan desenvolvido com HTML 5 (CAPUTO, 2010).....	39
Figura 18 - Estrutura básica do documento.....	43
Figura 19 - Posicionamento das tags canvas.....	44
Figura 20- Declaração de variáveis que serão utilizadas no jogo.....	45
Figura 21 - Função que representa a carta e suas características.....	46
Figura 22 - Função que prepara os pares de cartas e desenha-os na tela.....	46
Figura 23 - Função que manipula e embaralha as cartas.....	47
Figura 24 - Funções principais do jogo.....	49

Figura 25 - Função que carregará o jogo	50
Figura 26 - Tela principal do jogo	50
Figura 27 - Tela de resultados com o tempo utilizado para completar o jogo	50
Tabela 1 - Novos elementos estruturais do HTML5	23
Tabela 2 - Navegadores e seus respectivos motores	28
Tabela 3 - Compatibilidade entre navegadores e alguns módulos do HTML5	28

INTRODUÇÃO

Um documento web é composto por textos, imagens, sons, animações e vídeos, além destes elementos comuns possuem marcações feitas em linguagem HTML. Esta linguagem tem como finalidade formatar o texto do documento e fazer ligações entre páginas web, criando assim documentos com conceito de hipertexto. (RAMALHO, 1996).

O surgimento desta linguagem de marcação teve como principal objetivo suprir a necessidade global de distribuição de informações e fornecer entendimento universal dos documentos, não importando o dispositivo de acesso. Em sua mais recente versão, o HTML 5 traz diversas possibilidades e aperfeiçoamentos que permitiu a facilidade na manipulação dos atributos de marcação, realizando integração com elementos que definem o *layout* e outros que possibilitam a programação, além disso, a linguagem permitiu o acesso de conteúdo por computadores ou por dispositivos móveis sem a necessidade de instalações de *plugins*¹ adicionais.

O desenvolvimento de jogos para a web é principalmente lembrado por jogos desenvolvidos em Flash, tecnologia esta que atinge o público que busca diversão paralela a outras atividades realizadas na internet. Com o HTML 5, jogos para a web podem ser desenvolvidos para um amplo público, isso devido a sua compatibilidade com diversos dispositivos para acesso, o que se tornou uma vantagem perante as formas de desenvolvimento anteriores.

O objetivo central desta pesquisa é mostrar quão aplicáveis são as evoluções trazidas na versão 5 do HTML, em especial ao desenvolvimento de jogos digitais, direcionados para a web. Para alcançar este objetivo, o trabalho conta com seis capítulos. O primeiro conceitua a concepção da linguagem, o segundo mostra a evolução da linguagem, seus novos elementos e contribuição para o desenvolvimento da internet.

¹ *Plugins* são módulos de extensão que visam adicionar funções a outros programas maiores provendo algumas funcionalidades especiais e específicas. (HORTA, 2006).

O terceiro capítulo traz detalhes do conceito de desenvolvimento de jogos com HTML 5 realizando análise de alguns jogos que utilizam esta tecnologia. Assim, abre espaço para o quarto capítulo, que aplica os conceitos abordados em um exemplo prático, que será discutido em relação a suas vantagens e dificuldades no capítulo cinco. O sexto e último capítulo se reserva a expor as conclusões e destacar os pontos positivos da pesquisa.

1 A LINGUAGEM HTML

O HTML surgiu com o princípio de publicar documentos e conteúdos interligados na web e ser uma linguagem independente de plataformas, navegadores e outros meios de acessos. Sua sigla significa *HyperText Markup Language* ou linguagem de marcação de hipertexto em português. Esta linguagem também foi desenvolvida para quebrar barreiras como a acessibilidade de diversos dispositivos e outros meios com características diferentes, pois com ela cria-se a possibilidade de apenas um código poder ser lido por diversos meios de acessos, e desta forma contribuir com o desenvolvimento da Web sem proprietários, sem formatos limitados e incompatíveis.

Os documentos criados com o HTML são interpretados por diversos dispositivos, não importando o tamanho da tela, resolução ou variação de cor que estes possam apresentar até mesmo os dispositivos próprios para deficientes visuais e auditivos ou dispositivos móveis e portáteis são capazes de interpretar o documento formatado com esta linguagem. Devido a este entendimento universal, deu-se a possibilidade para a reutilização de informação de acordo com as limitações de cada meio de acesso.

A primeira versão da linguagem foi baseada em SGML que significa *Standard Generalized Markup Language* ou linguagem de marcação padrão e generalizado, em português. O SGML era utilizado para a estruturação de documentos e foi a partir destes conceitos que o HTML implementou diversas características na formatação de documentos, como a formatação de títulos utilizando a marcação <h1> até o <h6>, a formatação de cabeçalho com o <head> e a formatação de parágrafo <p>.

Segundo WILLIAM (2011), a grande diferença entre essas duas linguagens de marcação é que o HTML reformulou a tag <a> adicionando o atributo href, que possibilitou a esta nova linguagem a capacidade de realizar ligações de uma página a outra. Esse conceito de interligação entre documentos foi então considerado a base do funcionamento da Web e passou a ser conhecido como hipertexto, que tem por definição o conjunto de elementos que estão ligados por conexões de forma imprevista permitindo a comunicação de dados, estes elementos podem ser representados por palavras, imagens, vídeos, etc.

Hipertexto é um conjunto de documentos de qualquer tipo (imagens, texto, gráficos, tabelas, vídeos) conectados uns aos outros por links. Histórias escritas em hipertexto podem ser divididas em “páginas” que se desenrolam (como aparecem na *world wide web*). (MURRAY; DAHER, 2003)

Em 1980 o físico Timothy John Berners-Lee, conhecido também por Tim Berners-Lee, propôs um projeto baseado no conceito de hipertexto e este projeto foi denominado ENQUIRE, que inicialmente foi todo desenvolvido em linguagem Pascal e procurava associar e lembrar as conexões entre várias pessoas e projetos no laboratório onde trabalhava. Este projeto falhou por conta da dificuldade com ligações externas além de problemas com a manipulação de banco de dados. (WILLIAM, 2011).

Nove anos após as primeiras tentativas, em 1989, Tim Berners-Lee com a ajuda de um estudante chamado Robert Cailliau do CERN (Organização Europeia para Pesquisa Nuclear), conseguiu a primeira comunicação bem sucedida entre um cliente HTTP e um servidor através da internet. Surgia então a *World Wide Web*.

Pouco tempo depois, em 1990, surge o HTML que se popularizou quando o navegador *Mosaic* ganhou forças, este que foi desenvolvido por Marc Andreessen e segundo PILGRIM (2010) foi um dos primeiros navegadores web. A partir deste marco inicial os desenvolvedores e fabricantes de navegadores utilizaram a linguagem HTML como base para seus produtos.

Desde seu surgimento até o seu quinto ano de vida o HTML sofreu várias revisões e alterações na sua especificação. Neste período, os responsáveis pelo padrão era o CERN e a IETF (sigla em inglês de *Internet Engineering Task Force*), mas isso mudou logo em 1995 quando a W3C, órgão que regula os padrões Web, assumiu a regularização do HTML.

Segundo PILGRIM (2010) a sugestão de Dave Raggett, membro do time da W3C, em 1993 era evoluir a linguagem para o HTML+ o que nunca foi implementado, e as correções da versão anterior e formalização de algumas características só foi concretizada no HTML 2.0 o qual foi apresentado na primeira conferência mundial sobre Web, a *World Wide Web Conference* que o site oficial da W3C informa que aconteceu em Genebra, Suíça, de 25 a 27 de maio em 1994.

Logo após, Dave escreveu o HTML 3.0 que era baseado nos rascunhos feitos para o HTML+, entretanto, assim como o HTML+, esta versão também não foi

implementada, mas contribuiu para a versão 3.2, que trouxe mais correções e total compatibilidade com a versão anterior. Em dezembro de 1997, a W3C publicou o HTML 4.0 e também linguagens como XHTML, XForms, MathML além de outras especificações mais modernas, que também foram desenvolvidas com auxílio de Dave Raggett.

Dois anos depois, em 1999, o HTML 4.01 foi publicado trazendo como principal premissa a compatibilidade com as suas versões anteriores. Com o lançamento do HTML 4.0, o W3C alertou aos desenvolvedores sobre algumas boas práticas que deveriam ser seguidas ao produzir códigos *client-side*². Contudo, esta versão não trazia um diferencial real para a semântica do código, e não facilitava a manipulação dos elementos via JavaScript ou CSS.

1.1 W3C – WORLD WIDE WEB CONSORTIUM

Em agosto de 1994 inicia-se um consórcio formado por instituições comerciais e educacionais, com o objetivo de definir padrões para as respectivas áreas relacionadas à *Web*, este consórcio ficou conhecido como W3C. Segundo LAWSON e SHARP (2011), em 1998, o W3C decidiu que não continuaria a evoluir o HTML, pois acreditavam que o futuro era o XML³. Assim o HTML foi congelado na versão 4.01 e uma nova especificação foi lançada, o chamado XHTML, que foi um XML versão do HTML.

Em julho de 2004 a W3C organizou um *workshop* chamado “*Workshop on Web Applications and Compound Documents*”, que foi realizado em San Jose, Califórnia. Nesse encontro estavam presentes membros da W3C e companhias desenvolvedoras de navegadores como a fundação Mozilla e a Opera, onde foi apresentada uma visão do futuro da web com uma evolução do padrão HTML 4.0 incluindo novas características para aplicações modernas.

² Códigos *client-site* são interpretados no navegador de cada usuário

³ Formato para a criação de documentos com dados organizados de forma hierárquica que visa criar uma infra-estrutura única para diversas linguagens

Neste *workshop* a W3C evidenciou a todos que não seriam desenvolvidas novas tecnologias de extensões do HTML e CSS que não fossem criadas pelo atual grupo de trabalho da W3C, além de enfatizar a disposição para o desenvolvimento do XHTML. Diante desse impasse, um pequeno grupo de desenvolvedores do Opera que não estava convencido de que o XML era o futuro de todos os autores web começaram um trabalho extracurricular em uma especificação de prova de conceito, que estendia o conceito de formulários HTML sem quebrar a compatibilidade com versões anteriores. Estas especificações ficaram conhecidas como Web Forms 2.0, e foi posteriormente chamado de HTML 5. Em pouco tempo, os desenvolvedores da Mozilla e Apple apoiaram as novas especificações, e então surgiu um pequeno grupo que chamou-se o WHATWG (LAWSON; SHARP, 2011).

1.2 WHATWG – WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP

O *Web Hypertext Application Technology Working Group* ou grupo de trabalho tecnológico de aplicações de hipertexto para Web em português, também conhecido por suas siglas WHATWG é um grupo livre que conta com a colaboração dos desenvolvedores de navegadores e de seus interessados para desenvolver especificações baseadas no HTML e em tecnologias que buscam facilitar o desenvolvimento e a compatibilidade das aplicações Web, com a intenção de submeter os resultados para um padrão organizacional. O grupo foi criado para assegurar que o futuro desenvolvimento destas especificações fosse completamente livre, através de arquivos públicos e uma lista de discussão aberta.

O WHATWG foi fundado em 2004 como resultado de uma grande insatisfação de empresas como Mozilla, Apple e Opera com os rumos que a Web estava tomando. Estas organizações se juntaram para escrever o que seria chamado hoje de HTML 5, uma versão do HTML mais flexível para a produção de websites e sistemas baseados na Web. Entre outros assuntos, um dos temas principais focados pelo WHATWG era o *Web Forms 2.0* que foi incluído no HTML 5 e o *Web Controls 1.0* que atualmente está abandonado.

Por volta de 2006, o trabalho do WHATWG passou ser conhecido pelo mundo e principalmente pelo W3C - que até então trabalhavam separadamente - que reconheceu todo o trabalho do grupo. Em Outubro de 2006, Tim Berners-Lee

anunciou que trabalharia juntamente com o WHATWG na produção do HTML 5 em detrimento do XHTML 2. Contudo o XHTML continuaria sendo mantido paralelamente de acordo com as mudanças causadas no HTML. O grupo que estava cuidando especificamente do XHTML 2 foi descontinuado em 2009. O primeiro passo do trabalho em conjunto foi renomear o “*Web Application 1.0*” para “HTML5”.

Atualmente, a internet é um universo crescente de documentos e páginas interligadas além de aplicações constituídas por vídeos, fotos e conteúdo interativo. Ao longo do tempo as tecnologias web evoluíram a fim de oferecer aos desenvolvedores a capacidade de criar novas gerações de experiências úteis e imersivas para os usuários. A internet é hoje um resultado dos esforços em curso de uma comunidade aberta que ajuda a definir essas tecnologias web. A Figura 1 apresenta a evolução dos navegadores.

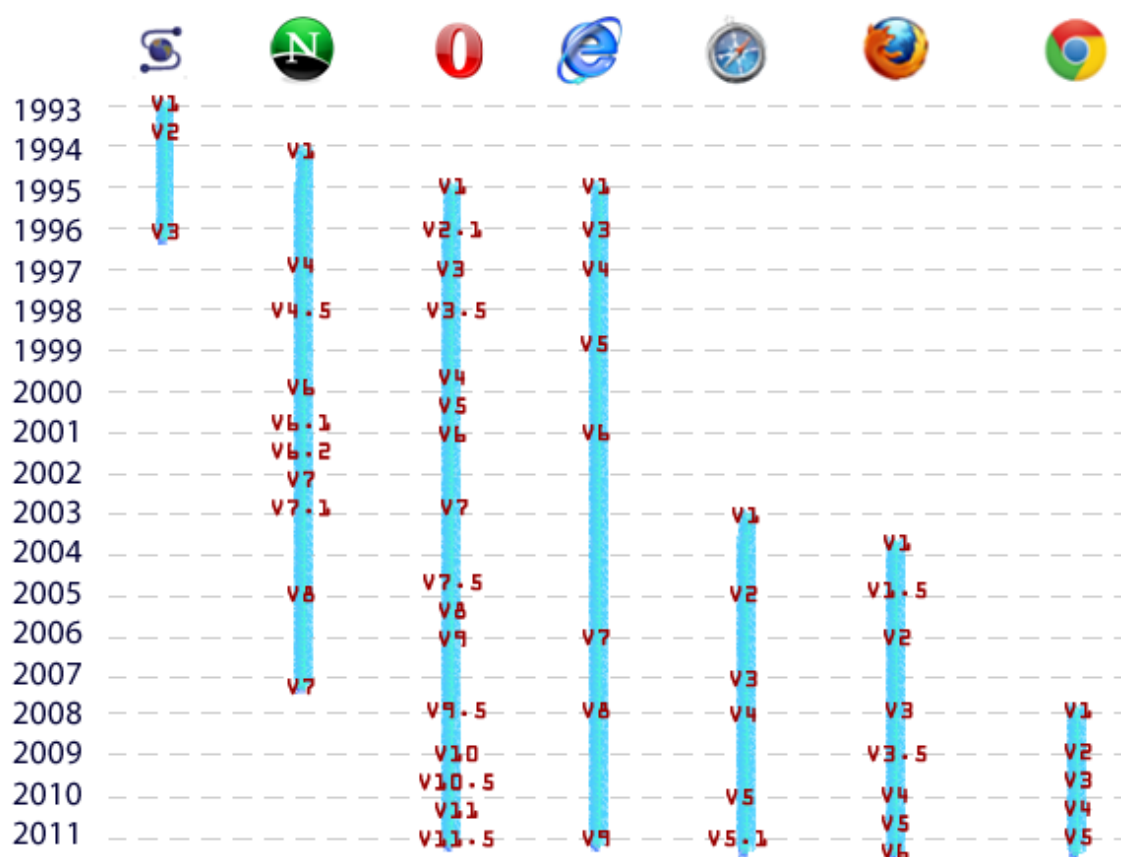


Figura 1 - A evolução dos navegadores

2 O HTML 5

Surgindo como a próxima geração do HTML e substituindo a versão 4.01, XHTML 1.0 e XHTML 1.1, o HTML 5 fornece novos recursos que são necessários para aplicações web modernas e padronização de muitas características da plataforma web que já são usadas por desenvolvedores há muitos anos, mas que nunca foram aprovados ou documentados por um comitê de padrões. (W3C, 2011).

Assim como seus predecessores, o HTML 5 é projetado para ser multi-plataforma, isso significa que um documento desenvolvido seguindo suas regras de marcação deve ser lido independente do sistema operacional que o usuário utiliza. A única necessidade desta nova linguagem é que o navegador esteja atualizado para as versões mais recentes que já possuem suporte e interpretação das novas características do HTML 5. Este requisito pode ser facilmente solucionado, devido ao fato de que existem navegadores modernos disponíveis gratuitamente para todos os principais sistemas operacionais. Os navegadores que vem pré-instalado em celulares e *tablets* também já possuem um excelente suporte as funcionalidades. (PILGRIM, 2010).

Uma das primeiras tarefas do HTML 5 foi documentar o não documentado, a fim de aumentar a interoperabilidade, deixando poucas adivinhações para autores da Web e implementadores de *browsers*. (LAWSON; SHARP, 2011)

A nova versão da linguagem tem como principal objetivo dar aos desenvolvedores a facilidade na manipulação dos elementos de marcação, possibilitando a modificação das características dos objetos de forma não intrusiva e transparente para o usuário final.

Diferente das versões anteriores, o HTML 5 fornece ferramentas para o CSS e o Javascript fazerem seu trabalho da melhor maneira possível, com isso, a linguagem permite a manipulação das características dos elementos através de suas APIs, também conhecidas como DOM, de forma que as páginas ou aplicações continuem leves e funcionais.

2.1 NOVAS CARACTERÍSTICAS DA LINGUAGEM

O HTML 5 oferece novas *tags* de marcações e aprimoramentos em algumas funções já existentes e um novo padrão universal para a criação de seções comuns e específicas nos documentos, como rodapé, menu, *sidebar* e etc. Além de um padrão de nomenclatura de IDs, Classes ou tags e métodos para auxiliar na captura automática das informações localizadas nos rodapés dos websites. Na Figura 2 e Figura 3 está respectivamente a estrutura desenvolvida com o HTML4 e a estrutura simplificada que o HTML5 oferece.

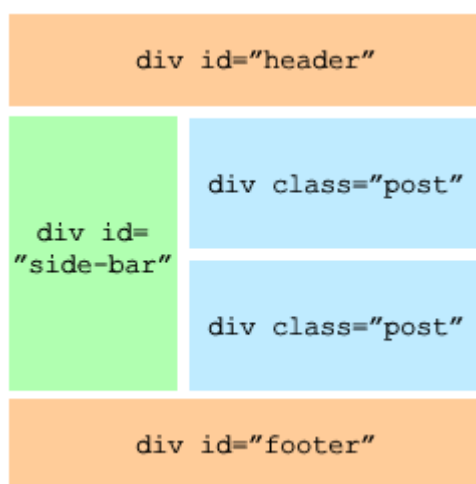


Figura 2 - Estrutura básica de uma página em HTML4

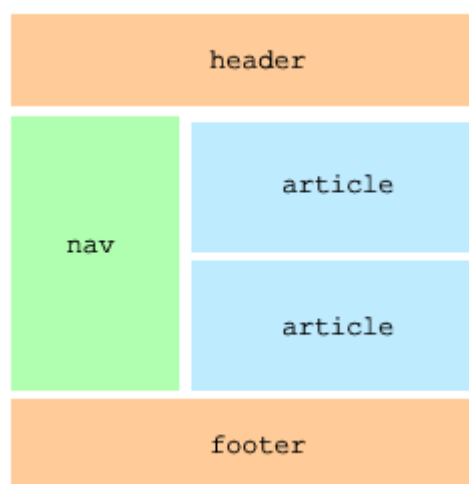


Figura 3 - Estrutura básica de uma página em HTML5

Como observado, esta nova versão da linguagem modifica a forma como os desenvolvedores deverão escrever seus códigos, dando a possibilidade de melhor organização das informações na página e fornecendo mais semântica com menor quantidade de código além de mais dinâmica através das atualizações freqüentes, isso devido a forma em que a comunidade de desenvolvedores da linguagem estão se dividindo, fazendo com que módulos pequenos da linguagem sejam escritos por pequenos grupos independente e cada grupo pode lançar atualizações a qualquer momento. (W3C, 2011).

Para os desenvolvedores de navegadores os freqüentes lançamentos de atualizações podem ser favoráveis, pois não é necessário esperar que todo padrão seja escrito e publicado para utilizá-lo. Porém é possível que ocorram problemas com a compatibilidade, e isso acaba tornando-se um meio de por em prática a

competitividade entre os fabricantes para manter-se atualizados perante seus concorrentes e garantir a leitura dos novos documentos.

Para os usuários, a linguagem traz mais interatividade sem a necessidade de instalação de *plugins* que podem afetar o desempenho da aplicação.

2.2 ESTRUTURA DO DOCUMENTO

A estrutura básica do HTML 5 ainda segue as estruturas das versões anteriores, apenas uma exceção na escrita do *DOCTYPE* que agora deve ser a primeira linha de código do documento antes da tag HTML, sua função é indicar ao navegador e para outros meios que acessam o documento qual a especificação de código utilizar. Vale lembrar que em versões anteriores era necessário especificar o DTD no código, mas agora esta referência torna-se de responsabilidade do navegador.

O propósito de uma DTD (*Document Type Definition*) é definir os blocos de construção legal de um documento XML. Um DTD define a estrutura do documento com uma lista de elementos jurídicos e atributos. (W3SCHOOLS, 2011)

Outra característica que mudou na estrutura do código básico do HTML é a *metatag*, esta que é responsável por chavear qual a tabela de caracteres será utilizada na página. Como nas figuras abaixo, a nova versão do HTML traz esta tag de forma simples e com seu significado claramente especificado, ao contrário da forma que era utilizado antes.

```
<html>
  <head>
    <meta http-equiv="Content-
Type" content="text/html;
charset=utf-8">
    <title>Exemplo</title>
  </head>
  <body>

  </body>
</html>
```

Figura 4 - Meta tag utilizada em versões anteriores

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Exemplo</title>
  </head>
  <body>

  </body>
</html>
```

Figura 5 - Estrutura básica de um documento feito com HTML5

2.3 ELEMENTOS ESTRUTURAIS

Nas versões anteriores do HTML, era possível marcar diversos elementos do *layout*, diferenciando um parágrafo de um título ou de subtítulo, mas não era possível diferenciar qual parte da página compunha o rodapé ou o cabeçalho. Esta diferenciação era apenas percebida visualmente pelo *layout* pronto ou pela posição dos elementos na estrutura.

Segundo LAWSON e SHARP (2011) em 2004, o editor da especificação HTML 5, Ian Hickson, examinou bilhões de páginas web através do índice do Google, olhando para ver do que a web "real" é feita. Uma dessas análises foi posteriormente publicada em uma lista com os nomes das classes mais populares dos documentos HTML. No topo da pesquisa estão os identificadores *footer*, *content*, *headers* e *container*, já no topo da lista de classes estão o *footer*, *menu*, *style1* entre outros.

Como resultado desta pesquisa, a nova versão da linguagem HTML traz aos desenvolvedores uma série de elementos que auxiliam na definição dos primeiros setores do documento. Com estes elementos, é possível diferenciar áreas importantes do site, como cabeçalho, menu, rodapé, entre outros que são muito utilizados. Estas mudanças simplificam o trabalho de sistemas como os dos buscadores que conseguem vasculhar o código de maneira mais eficaz e estocar rapidamente informações exatas sobre o documento. A Tabela 1, abaixo, apresenta resumidamente os novos elementos e suas respectivas descrições.

ELEMENTO	DESCRIÇÃO
<header>	Responsável por definir o cabeçalho da página ou de uma sessão, este elemento representa um grupo de introdução ou elementos de navegação. Pode ser utilizados para agrupar índices de conteúdos, campos de busca ou até mesmo logotipos. (W3C, 2011).
<hgroup>	Consiste em um grupo de título que é utilizado para agrupar um conjunto de elementos de título de <h1> até <h6>, quando o título tem múltiplo níveis, como subtítulos, títulos alternativos, e etc. (W3C, 2011).

<section>	Define uma nova seção genérica no documento. Segundo W3C (2011), a página inicial de um site poderia ser dividida em seções para uma introdução ou destaque, chamadas para conteúdo interno ou informações de contato.
<article>	Representa uma seção que consiste em uma composição ou itens pertencentes a uma parte independente de um documento, como uma postagem de fórum, uma revista ou artigo de jornal, ou qualquer outro item de conteúdo não dependente que se apresente completo ou faça sentido por si só. (W3C, 2011).
<footer>	Representa o rodapé de um documento e é o último elemento antes de fechar a <i>tag</i> HTML, geralmente contém metadados sobre o fechamento de sua seção, como autor, links para documentos relacionados, os dados de copyright, etc. (W3C, 2011).
<nav>	Responde pela seção de navegação em um documento HTML 5. É uma seção que contém links para outros documentos ou partes dentro do documento atual (W3C, 2011).
<aside>	Representa a um bloco de conteúdo que referencia os dados que estão em sua volta e que poderia de certa forma ser considerado separado do conteúdo do documento. (W3C, 2011).

Tabela 1 - Novos elementos estruturais do HTML5

A Figura 6 exemplifica a distribuição comum dos elementos estruturais descritos até o momento.

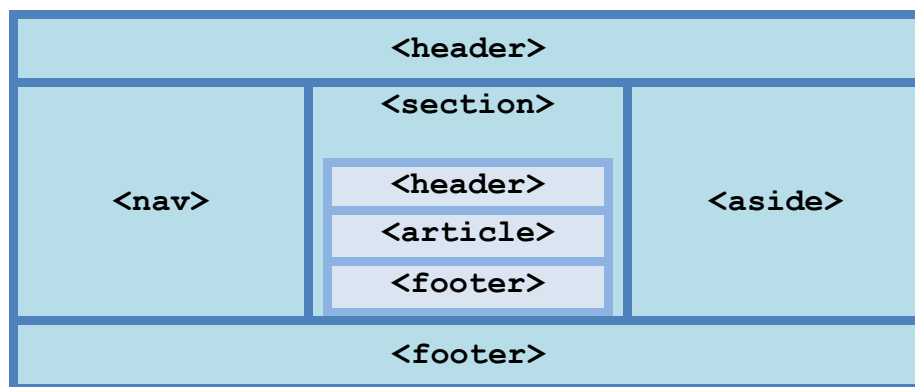


Figura 6 - Elementos de marcação estruturais HTML5.

Com estas alterações nos elementos da linguagem de marcação, houve alteração em alguns atributos, estas alterações foram feitas afim de que essas funções sejam controladas através de folhas de estilo.

2.4 ELEMENTOS MULTIMÍDIA

Em versões anteriores do HTML, não havia nenhuma maneira baseada em padrões para incluir vídeos ou áudios nas páginas web, a forma de se fazer isso era através de uma canalização de um *plugin* de terceiros - QuickTime, RealPlayer ou Flash. Mas estes *plugins* causavam muitos transtornos aos usuários que tentavam assistir a um vídeo em uma plataforma o qual não tinha suporte.

Com uma das soluções trazidas pela versão cinco da linguagem, o HTML introduziu uma forma padrão para incorporar vídeos em uma página web, oferecendo a habilidade de embutir mídia em documentos HTML. A Figura 7 e 8 apresentam exemplos de utilização das novas tags para inserção de áudio e vídeo na página, atribuindo valores para os atributos principais.

```
<audio src="mus.oga" controls="true" autoplay="true" />
```

Figura 7 - Exemplo de inclusão de áudio na página

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

Figura 8 - Exemplo de inclusão de vídeo na página

2.5 GRÁFICOS NA WEB

O HTML 5 apresenta o desenvolvimento de figuras e gráficos para a web através de SVG ou CANVAS.

O SVG (*Scalable Vector Graphics*) é uma linguagem de marcação para descrever aplicações gráficas bidimensionais e imagens, além de um conjunto de interfaces gráficas relacionadas a *script*. Este padrão aberto está se desenvolvendo desde 1999 por um grupo de trabalho do W3C com base nos formatos PGML da Adobe, e VML da Microsoft, submetidos ao W3C por essas empresas em 1998. A criação deste formato foi baseada em outros já existentes: CSS, DOM, JPEG, PNG, SMIL e XML.

Sua versão 1.1 é uma recomendação do W3C e é a versão mais recente da especificação completa, há também uma segunda edição do SVG 1.1 que inclui esclarecimentos e pequenas melhorias com base no *feedback* dos desenvolvedores. Atualmente o SVG 2 está em desenvolvimento, e irá adicionar novas facilidade de uso características para SVG, bem como uma maior integração com HTML e CSS.

Nos primeiros dias da Web, houve uma explosão de diferentes formatos e extensões para HTML. Desde o início, ficou claro que um formato gráfico vetorial para a Web seria útil. Chris Lilley escreveu um documento para requisitos gerais para gráficos vetoriais em 1996. (W3C, 2010).

2.5.1 CANVAS API

A Canvas API permite desenhar na tela do navegador via Javascript onde o único elemento HTML que deve ser adicionado no documento é o elemento canvas, pois todas as outras partes é de responsabilidade do Javascript. A Figura 9 mostra como inserir o elemento na página, desta forma, cria-se um retângulo vazio na página.

```
<canvas id="canvas" width="300" height="300"></canvas>
```

Figura 9 - Exemplo de inclusão do canvas no documento HTML

Para desenhar na área criada com o código acima, já na programação em Javascript é necessário obter um contexto e utilizar seus métodos. Na Figura 10 o

exemplo mostra como adquirir o contexto e posteriormente, como o utilizá-lo para desenhar, e a Figura 11, em seguida, mostra o resultado da interpretação do código no navegador.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo com função de verificação de contexto</title>
    <script type="text/javascript">
      // este exemplo cria uma função que irá retornar o contexto exato,
      fazendo verificações necessárias para que não seja mostrando erros aos
      usuários
      function CarregaContextoCanvas(idCanvas){
        var elem = document.getElementById(idCanvas);
        if(elem && elem.getContext){
          var contexto = elem.getContext('2d');
          if(contexto){
            return contexto;
          }
        }
        return FALSE;
      }

      window.onload = function(){
        var context = CarregaContextoCanvas('exemplo');
        context.fillStyle='#6666ff';
        context.fillRect(10,10,50,50);
        context.fillStyle='rgba(255,255,0,0.7)';
        context.fillRect(35,35,50,50);
      }
    </script>
  </head>
  <body>
    <canvas id="exemplo" width="500" height="500">
    </canvas>
  </body>
</html>
```

Figura 10 - Exemplo de utilização do contexto 2d para desenhar dois retângulos



Figura 11 - Desenho utilizando o contexto 2d em canvas

2.5.2 CANVAS E SVG

Para escolher entre usar Canvas ou SVG é necessário entender as diferenças entre um e outro. O primeiro, Canvas, é desenhado *pixel a pixel* via Javascript, já o segundo, SVG, é vetorial e baseado em XML, com isso é acessível via DOM⁴. Através destas informações é possível observar que a vantagem de utilizar o Canvas é em relação ao desempenho ser superior ao SVG e sua facilidade para desenhar, por ser via Javascript. Já as vantagens da utilização do SVG são por gerar um conteúdo totalmente acessível a leitores de tela, além de ser um gráfico escalável, que não perderá resolução quando redimensionado.

Para selecionar um elemento HTML e poder manipulá-lo via JavaScript/DOM é necessário ter algo que identifique esse elemento. Este gancho pode ser o identificador do elemento, o nome da classe, o nome da tag ou um atributo ou valor de atributo específico. Vale lembrar que o que manipulamos via JavaScript são referências a elementos HTML. Ou seja, uma variável que “aponta” para o elemento e que, uma vez modificada, reflete as modificações no elemento em si.

2.6 COMPATIBILIDADE DO HTML5

Atualmente, há uma grande diversidade de dispositivos que acessam a internet, podendo ser *tablets*, *smartphones*, computadores e etc. Cada um destes meios de acesso utilizam um determinado navegador para acessar a web e isto torna-se uma preocupação a mais para desenvolvedores que buscam manter um bom nível de compatibilidade com todos estes navegadores levando em consideração as particularidades de cada um. Uma forma segura para manter o código compatível, é nivelar o desenvolvimento de acordo com os motores de renderização de cada navegador, estes que são responsáveis pelo processamento do código da página. A Tabela 2 apresenta a lista com os principais navegadores e seus motores.

⁴ O Modelo de Objetos do Documento (DOM, na sigla em inglês) é a interface entre a linguagem Javascript e os objetos do HTML. DOM é o método padrão para construção de aplicações ricas com Javascript e é amplamente conhecido e utilizado. (W3C, 2011)

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

Tabela 2 - Navegadores e seus respectivos motores

É importante que o código esteja compatível com estes motores para atingir uma amplitude maior de entendimento dos navegadores. Atualmente o *Webkit* é o motor mais compatível com os padrões do HTML 5, isso envolve diversos fatores, mas o principal é que a Apple tem grande interesse no desenvolvimento e avanço da compatibilidade deste motor. Contudo, outros motores já estão trabalhando para atender as novas especificações do HTML 5 e CSS 3.

A grande preocupação para os desenvolvedores tem sido a retrocompatibilidade que é a compatibilidade com as versões antigas de navegadores. Abaixo uma Tabela 3 apresenta a compatibilidade entre os navegadores e alguns módulos do HTML 5.







						
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
Vídeo, áudio, e Canvas	S	S	S	S	N	S

Tabela 3 - Compatibilidade entre navegadores e alguns módulos do HTML5

De acordo com a avaliação do HTML5Test , site que classifica de 1 a 160 pontos o suporte dos browsers ao HTML5, temos:

- Google Chrome 4.1 – 118 pontos
- Opera 10.51 – 102 pontos
- Firefox 3.6.3 – 101 pontos
- Internet Explorer 7/8 – 19 pontos

Para solucionar problemas de incompatibilidade é necessária a utilização de algumas técnicas para verificar se o navegador suporta ou não os elementos do HTML 5, a partir disso o usuário é redirecionado para uma página adequada, ou para uma página que alerte o usuário sobre a importância da atualização do navegador.

Há diversas formas para se realizar estas verificações, a principal é a utilização da biblioteca *Modernizr* que permite verificar do suporte da maioria das características do HTML 5 e CSS 3. Para utilizá-lo, é necessário fazer sua chamada no *head* do documento e então realizar a chamada de seus métodos. A Figura 12, mostra como é desenvolvido o código para verificação de suporte da API de geolocalização utilizando o Modernizr.

```
if (Modernizr.geolocation) {  
    // Aceita a funcionalidade  
}else{  
    // Não aceita a funcionalidade testada.  
}
```

Figura 12 - Código de verificação da API *Geolocation*

3 DESENVOLVIMENTO DE JOGOS COM HTML – JOGOS *ON-LINE*

A internet é recebida de forma diferente pelo público quando comparada a outros meios de comunicação como a TV, o rádio ou a revista. Este fato ocorre devido à forma de consumo do material disponibilizado por este meio que exige uma participação ativa, curiosa e paciente de cada usuário. Segundo HUYSSSEN (2000) a modernidade alternativa se constrói por fluxos e embates horizontais da cultura, incrementados pelas novas mídias e pelo processo crescente de globalização e comunicação.

Como já visto no capítulo anterior, a internet na década de 1990 proporcionou o desenvolvimento dos sistemas em rede, da comunicação global e do conceito de interatividade como lugar de encontros fundados sobre os processos de comunicação. Já em âmbito dos jogos digitais, CELLEGAR (2011) afirma que as propostas dos que buscam explorar o desenvolvimento de jogos é de utilizar a linguagem multimídia e hipertextual da tecnologia e da internet, e desenvolvem-se próximo ao usuário, que através de suas escolhas, recebem respostas que a máquina lhe dá.

O computador ligado em rede atua como um telefone, ao oferecer a comunicação pessoa-a-pessoa em tempo real; como uma televisão ao transmitir filmes; um auditório, ao reunir grupos para palestras e discussões; uma biblioteca, ao oferecer grande número de textos de referências; um museu, em sua ordenada apresentação de informações visuais, como um quadro de avisos, um aparelho de rádio, um tabuleiro de jogos e, até mesmo, como um manuscrito, ao reinventar os rolos de textos dos pergaminhos. (MURRAY; DAHER, 2003).

Segundo SMED e HAKONEN (2003) os jogos sempre foram um passa-tempo popular, mas com o advento do computador eles passaram a se desenvolver de forma mais promissora. Para definir os *games*, HUIZINGA em seu clássico trabalho conhecido como *Homo Ludens* (1938) inicia a definição de jogar dizendo ser uma atividade que prossegue dentro de certos limites de tempo e espaço, em uma ordem visível, de acordo com regras livremente aceitas e fora a esfera da necessidade ou utilidade material, esta ação causa um sentimento de exaltação e tensão acompanhada de alegria e relaxamento.

Segundo os mesmos autores, SMED e HAKONEN (2003), o dicionário define o jogo com uma forma universal de recreação em geral, isso inclui qualquer atividade exercida por diversão ou o estabelecimento de uma situação que envolve

uma competição ou rivalidade. Também pode ser definido por uma estrutura que promove maior ou menor grau de interatividade, através de um contato ou uma participação que leva as mudanças contínuas na relação com o jogador. Em um jogo, há desafios a superar, regras a cumprir, caminhos a percorrer, erros que propõe ao jogador, voltar e descobrir novas soluções, desta forma, exige ações e interações.

A categoria de jogos *on-line* permite que um jogador conecte-se a outros através da rede e isso possibilita jogos em equipes sem que ambos estejam no mesmo ambiente, além de desafiar seus adversários mesmo que eles estejam em outra cidade, estado ou país, essas ações faz com que a modalidade de jogos *on-line* seja considerada uma nova perspectiva de diversão.

Segundo a revista CDEXPERT (2002), este modo de jogar existe há muito tempo e historicamente as primeiras partidas *on-line* começaram por volta de 1969 quando Rick Blomme escreveu uma versão para 2 jogadores para o *game Spacewar*. Logo após esta criação, nos anos 70 surgem diversos jogos com a mesma ideia de diversão para várias pessoas simultaneamente, ideia que cativou os jogadores com sua praticidade, dentre os jogos *multiplayer* está a versão *online* para o *Star Trek*, (Figura 13).

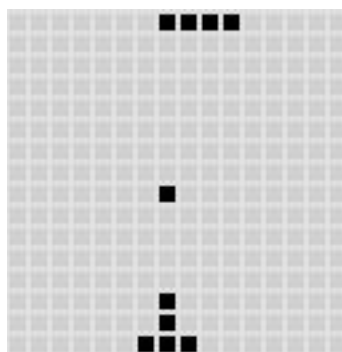


Figura 13 - Star Trek (COHEN, 2011)

Ao fim desta década, Roy Trubsham e Richard Bartle trabalharam juntos no desenvolvimento do primeiro *Multi-User Dungeon* (MUD), que seria a aplicação do conceito de jogo *on-line* no estilo *Role Playing Game* (RPG). O projeto foi todo desenvolvido em um Dec-10 da Universidade de Essex, em Colchester, Reino Unido.

Pensando em propósitos educacionais, Bartle distribuiu cópias do código desenvolvido para outras universidades e centros de estudos e em 1983 centenas de cópias já estavam espalhadas pelo mundo. Enquanto isso, Bill Louden criou o *Mega Wars I* (Figura 14).



Figura 14 - *Mega War I* (KHOA, 2005)

No ano seguinte, em 1984, foi lançada a primeira versão comercial de um jogo MUD e junto a isso, surgia o *Islands of Kesmai* que ficou ativo no mercado por 13 anos, o que gerou versões gráficas sofisticadas. No mesmo período Mark Jacob montou um sistema servidor doméstico que possibilitava a instalação de 8 linhas telefônicas para rodar um jogo de RPG chamado *Aradath*.

Após dois anos, em 1986, Kesmai reescreveu seu jogo de sucesso, *MegaWars I*, e desenvolveu seu primeiro jogo *multiplayer on-line* conhecido como *Stellar Warrior*. Neste mesmo ano, surge um jogo via email, que foi chamado de *The Rim Worlds War* e em seguida um jogo de simulação de combate aéreo chamado de *Air Warriror* (Figura 15).



Figura 15 - *Air Warrior* no PC (HAWMAN; NOLAN, 2011)

Em 1987 desenvolveu-se muitas evoluções no *MegaWars III* e foi lançado o *Jack's Cassino*. No ano seguinte surge o *Gamstone II* que superou a popularidade do *Air Warrior* e tornou-se o sucesso do momento. Nesse ano também, a *Quantum Computer Services* lançou a *AppleLink: Personal Edition* na *May AppleFest Convention*, em Boston, Estados Unidos. Oito anos depois surge o *Gemstone III* e *Dragon's Gate*, que era a versão comercial de *Aradath*. Junto a estes lançamentos, também nasceu outro clássico dos jogos online quando a *Quantum* adquiriu a licença e lançou *AD&D: NeverWinter Nights* (Figura 16).



Figura 16 - *NeverWinter Nights* (KAO, 2009)

Alguns anos seguintes foram marcados por lançamento de jogos espaciais de estratégia em tempo real e pelo surgimento do primeiro jogo 3D *on-line* da história, este jogo era conhecido como *A-Maze-ing*. Houve também o lançamento de um jogo baseado em texto no estilo RPG, este que posteriormente tornou-se o *Dragon's Gate*. Iniciou-se também um projeto chamado *Basic Services*, que oferecia além do convencional, um serviço de *chat*, com uma mensalidade de 8,95 dólares.

Dragon's Gate foi lançado em fevereiro, e logo estava ao topo dos 3 jogos mais jogados na GENie, alternando mês a mês o primeiro lugar com o campeão da época *Air Warrior*. Ken Williams, da *Sierra Online*, anuncia que está chegando "*The Sierra Network*".

Em 1992, com o fim da empresa GENie que era a responsável por tantos lançamentos, surge a MPG-NET, uma empresa fundada pelo criativo produtor de jogos Jim Hettinger, que lança um novo serviço de jogo *on-line*, serviço que se popularizou porque mais de 3000 jogadores se inscreviam para jogar 1 hora com preços de 3 a 5 dólares.

Ao lado da *Sierra Network*, a *Sierra Online* investiu no desenvolvimento de jogos desta categoria, oferecendo um serviço no qual jogavam de 2 a 4 jogadores, que se chamava *Nine Man's Morris*.

Em 1993 aproximadamente 4 milhões de lares já se aventuram na Internet, no ano seguinte, em 1994, foi lançado Doom que teve muitas requisições dos usuários para a inclusão da opção de ser jogado via internet, o que foi concretizado em sua segunda versão que ficou conhecida como Doom II.

Em 1995, outro jogo clássico tem a opção embutida para jogar via servidor de internet, o Quake foi a versão melhorada da série Doom. Uma estimativa afirma que nessa época mais de 300 jogos MUD baseados em texto ficavam disponíveis na Internet, e grande maioria, quase que totalmente grátis.

Um ano após, em 1996, finalmente foi lançado oficialmente o Quake, e em pouco tempo surgiram servidores deste jogo por quase todo o mundo, e em poucos dias mais de 80mil pessoas jogavam este clássico o que aguçou o gosto dos jogadores por este estilo de jogo. Como resultado disso, no mesmo ano foram lançados cerca de 20 jogos como estes que faziam o uso da conexão via Internet.

No final deste período, a maioria dos jogos RTS (*Real Time Strategy* ou Estratégia em Tempo real, em português) tinha suporte nativo a Internet, o que permitia aos jogadores de todo o mundo interagir uns com os outros, com isso surgiu alguns serviços para reunir as pessoas da sala de jogos em *chats* e compartilhar experiências de seus personagens.

Assim como os jogos para consoles e para computadores, os jogos *on-line*, interligados por redes estão em grande expansão, sempre acompanhando a expansão global dos meios que proporcionam esta comunicação. Estes jogos podem ser simples como textos, ou podem possuir gráficos complexos explorados por diversos jogadores simultâneos. Muitos destes jogos estão incorporados em redes sociais tornando uma atividade social, além do *single player*, pois a vantagem para esta tecnologia é a capacidade de conectar jogos *multiplayer*.

Jogos *on-line* é uma tecnologia em vez de um gênero, um mecanismo para ligar leitores juntos em vez de um determinado padrão de jogo. (ROLLINGS; ADAMS, 2006).

Com o surgimento de navegadores mais sofisticados, inicia-se também o desenvolvimento de jogos mais robustos, que utilizam o navegador como um *client*. Jogos simples foram feitos para serem jogados usando navegador web via HTML e *scripts*, já os jogos sofisticados entram em contato com servidores para permitir um ambiente *multiplayer*.

Esta evolução e sofisticação dos jogos vieram através da crescente popularidade de tecnologias *web-based* gráficas, tais como Flash e Java agregaram a complexidade para os jogos e levaram uma revolução da Internet, onde páginas web passaram a incorporar vídeos, áudios e novos conjuntos de interatividade ao usuário. Esta revolução abriu caminhos para sites que ofereciam jogos para seus internautas, popularizando ainda mais esta forma de desenvolvimento.

Com isso, muitos desenvolvedores ou empresas tentam inovar criando jogos criativos e simples para atrair os jogadores casuais, que são aqueles que jogam aproximadamente hora por dia e de forma esporádica. Esse perfil de jogador procura por jogos que proporcione diversão rápida e de forma simples. Eles acabam encontrando todos esses requisitos em jogos para a web, a maior parte desenvolvido em Flash, pois possuem a vantagem de serem acessados de forma simples pelo navegador, apenas requisitando a instalação de um *plugin* para que

seja carregado, e então iniciado. Muitos jogos lançados na década de 1980, como Pac-Man e Frogger, foram recriados como jogos utilizando o *plugin* do Flash em uma página web.

Este crescimento e popularização da Internet tornaram possíveis também a utilização de novas estratégias e ferramentas para apoiar o processo de ensino e aprendizado. Uma das formas de propagação desta idéia é através dos jogos via Web que possibilitam a aprendizagem assíncrona, agregação e processamento dos dados, interação em tempo real através de uma população geograficamente dispersa e um cenário dinâmico.

Financeiramente, os criadores dos jogos para este novo perfil de usuário buscam lucrar cobrando taxas mensais por seus serviços, ou oferecem opções básicas do jogo, mas se o jogador desejar ir além podem realizar o pagamento para desbloquear estes novos conteúdos, membros ou acessórios. Atualmente, produtores de games independentes encontram na internet um meio para divulgar seus projetos e os jogos que são lançados utilizam tecnologias web como Ajax para tornar as interações *multiplayers* mais complicadas, e o WebGL para gerar aceleração por hardware gráfico 3D sem a necessidade de *plugins*. Segundo estudo elaborado pela consultoria Gartner, a previsão do faturamento mundial do setor de jogos *on-line* é de aproximadamente US\$ 6 bilhões até o fim do ano de 2011.

3.1 JOGOS EM FLASH

No mesmo instante em que ocorria a explosão da Internet no Brasil, no período de 1998 a 2000, a empresa Macromedia conseguiu consagrar o Flash no mercado mundial. A evolução do Flash como ferramenta de autoria para Web acompanhou a evolução das transições de sites com conteúdo estático para sites com conteúdo dinâmico e integrados. (AWAMURA, 2000).

O Flash nem sempre foi uma plataforma rica e avançada como é hoje, no entanto, iniciou como um pacote de animação vetorial chamado *FutureSplash Animator* em 1996, criado por Jon Gay e Jackson Charlie, co-fundadores do software *FutureWave*. Esta primeira versão foi a extensão animada de *SmartSketch FutureWave*, um aplicativo de desenho vetorial. Seis meses após o anúncio da primeira versão a Macromedia compra a *FutureWave* e a nomeia para Flash.

No ano seguinte, em 1997, surge o Flash 2, versão que não incorporou muitas novidades, mas foi a versão considerada o primeiro contato para muitos usuários. Logo em seguida em 1998, o Flash 3 trouxe consigo um série de novas funcionalidade. De início, foram introduzidos os *MovieClips* e a transparência. Com isso, iniciou-se a criação de filmes em Flash.

Em 1999 surge a versão 4 do Flash, neste mesmo período o Flash foi considerado o padrão para as animações na web. Esta nova versão trouxe ferramentas alteradas e ícones novos, melhorando a interface e possibilitando aos usuários moverem-se entre as cenas e objetos. Nesta versão, o Flash passou a apoiar o formato de som MP3, com isso foram produzidos diversos players de músicas e vídeos, além de som, houve também a incorporação de variáveis que foi uma funcionalidade de destaque.

Com as grandes facilidades propostas pela versão 4 do Flash, muitos usuários realizavam experimentos com os novos recursos e foram responsáveis pela criação de surpreendentes novos aplicativos e exemplos.

Em 2000 foi anunciado o Flash 5, que ressurgiu com uma série de novos recursos. Esta versão trouxe como uma de suas principais características a introdução do *ActionScript*, que é uma linguagem de programação orientada a objetos. Neste período experimentos em Flash tornou-se ainda mais numerosos e os jogos produzidos com esta ferramenta passaram a multiplicar-se.

Após dois anos, em 2002, o Flash foi integrado ao conjunto de MX, percursos do *Creative Suite*, e passou a se chamar Flash MX. Nesta versão, foram introduzidos diversos componentes de interface e a característica marcante deste lançamento foi a introdução do suporte a vídeos, com isso *Streaming* de vídeos tornaram-se comuns e base para o *YouTube*.

Em 2004 e 2003, o Flash MX 2004, introduziu mais componentes, *ActionScript 2*, e suporte a *Unicode*, padrão que permite o computador manipular textos. Este ano marcou o surgimento de "empresa" Flash, movendo o software mais longe de suas origens humildes e para ser a plataforma RIA (*Rich Internet Applications* ou Ricas Aplicações de Internet em português) dominante.

Em 2005 surge o Flash 8 que introduziu filtros e modos de mistura para simplificar efeitos especiais de animação e alguns objetos para desenhos foram adicionados.

Em 2007 o Flash se integra ao *Adobe Creative Suite* e então surge o Flash CS3 que renovou sua interface, acrescentando ferramentas para melhor manuseio de vídeos. Este lançamento foi marcado pela integração completa com o resto da *Creative Suite*, particularmente *Photoshop* e *Flex*, e da introdução de *ActionScript* 3.0.

Um ano após a junção entre o Flash e a Adobe, em 2008, foi feita e uma revisão da interface e do editor de movimentos e suporte ao 3D básico. Com isso, foi lançado o Flash CS4.

Em 2010 surge o Flash CS5 que melhorou o editor de código, a cinemática inversa, acrescentou trechos de código, entre outras funcionalidades que visava simplificar o gerenciamento do código fonte e reduzir os erros associados com o formato de arquivo.

Atualmente a Adobe já anunciou o apoio ao HTML 5, informando que descontinuará o desenvolvimento do Flash, inicialmente para dispositivos móveis e posteriormente para outro meios de acesso.

3.2 JOGOS EM HTML 5

As primeiras versões do HTML buscavam realizar a interação básica entre o usuário e o documento através de *hiperlinks*. O aprimoramento desta versão foi marcado por diversas inclusões de funcionalidades para aumentar a produtividade e atrair ainda mais os usuários.

O HTML 5 apresenta algumas vantagens sob o Flash, uma das principais e mais destacável é o fato de que esta linguagem não requer nenhum *plugin* para ser executado. Com esta linguagem de marcação, a produção de gráficos de vetor e animações poder ser desenvolvidas sem o Flash, apenas com algumas linhas de códigos e bibliotecas específicas. (*Seattle Web Development*, 2010).

Para os produtores de jogos, a requisição de um *plugin* para a execução de seus jogos pode representar um perda de vendas, mesmo sabendo que uma grande porcentagem de computadores já possui estes pré-requisitos.

Por conta desta melhor integração com os usuários que a versão 5 do HTML fornece, diversos jogos, dentre eles alguns clássicos como PacMan foram reescritos para este novo formato. A Figura 17 apresenta as imagens destes jogos.

O jogo funciona exatamente como toda versão original do Pacman onde o usuário controla a direção através das setas do teclado e controla a fuga do personagem principal dos fantasmas.



Figura 17 - PacMan desenvolvido com HTML 5 (CAPUTO, 2010)

Outro jogo a ser citado é *Quake II GWT Port* que foi desenvolvido por alguns funcionários do Google usando WebGL, o Canvas API, elementos de HTML5, dentre outras.

3.2.1 COMPLEMENTOS PARA O DESENVOLVIMENTO DE JOGOS COM HTML5

Algumas ferramentas e bibliotecas são consideradas essenciais para o desenvolvimento de jogos *on-line*. A versão 5 do HTML modifica a forma com que os desenvolvedores devem trabalhar, trazendo praticidade e bons resultados, além de integração de grupos de desenvolvimento, isso devido a padronização que a nova versão impõe. O desenvolvimento da lógica e efeitos sofisticados do jogo são itens de responsabilidade da linguagem javascript, esta que possibilita a manipulação dos elementos que estão no documento. Já o desenvolvimento do *layout* do jogo requer

a utilização da linguagem de estilo conhecida como CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata, em português).

A especificação do HTML5 introduz novos recursos para o desenvolvimento de jogos, como as novas tags **<canvas>**, **<vídeo>** e elementos de **<audio>**. Estes novos elementos da linguagem quando combinado com o de outras tecnologias web que não requerem a instalação de *plugins* faz com que o desenvolvimento de jogos seja elevado para um nível totalmente novo.

Na sua forma mais básica os jogos codificados com o HTML 5 consistem em JavaScript manipulando o elemento **<canvas>** através da sua propriedade de identificação. Já em jogos mais complexos é necessário programação através de WebGL 3D, *WebSockets* para que seja possível a utilização da capacidade *multiplayer*, e armazenamento *off-line* para a possibilidade de jogar mesmo quando não conectado à internet.

No nível mais abstrato, o HTML 5 está buscando reescrever a web para que jogos e aplicativos não necessitem dos *plugins* do Flash, esta é a justificativa dos rumores entre desenvolvedores de que o HTML 5 substituiria o Flash.

O HTML 5 está prestes a ser considerado como a base da era de jogos para dispositivos moveis, mesmo ainda em fase de desenvolvimento, a linguagem tem progredido rapidamente, com isso muitas empresas já estão se adequando e fazendo pequenas mudanças em seus códigos.

Os benefícios do HTML 5 estão evidentes, e com o apoio de grandes empresas será um incentivo para as mudanças acontecerem nos jogos *on-line*, tornando-os ainda melhores.

3.2.1.1 ENGINES E FRAMEWORKS

Além do JavaScript e CSS, há algumas ferramentas que também auxiliam com o desenvolvimento de jogos, algumas permitem a construção de jogos sem conhecimentos profundos de programação. Em uma breve descrição a seguir as *Engines* e *Frameworks* que surgiram no último ano:

Isogenic Engine: Esta *engine* tem como objetivo permitir que os desenvolvedores criem diversos tipos de jogos, desde plataformas, estratégias 2D e

multiplayers em rede e jogos sociais. O site oficial com informações desta *engine* é <http://www.isogenicengine.com/>.

LimeJS: Este *framework* para *games* em HTML 5 já foi utilizado para o desenvolvimento de jogos como *RoundBall* e *Zlizer*. As informações e links para downloads podem ser encontradas no site: <http://www.limejs.com/>.

Game Closure: ainda na versão beta privado, este HTML *game framework* permitirá aos desenvolvedores a criação de jogos multi-plataforma e jogos *multiplayer*. Todas as informações deste *framework* podem ser encontradas no site oficial <http://gameclosure.com/>.

Impact: Uma *engine* usada para criar jogos com alta qualidade e desempenho como *Biolab disaster* e *Z-type*. Esta *engine* já possui versões comercializadas e estão disponíveis por US \$99, o site com informações oficiais é <http://impactjs.com/>.

Akihabara: Esta é uma HTML 5 *game engine* foi desenvolvida por Francesco Cottone, e permite ao desenvolvedor criar jogos estilo 8/16 bits sem a necessidade de nenhum *plugin* e que também trabalham em dispositivos móveis e sensíveis ao toque. As informações oficiais e demonstrações estão disponíveis no site <http://www.kesiev.com/akihabara/>.

Construct 2: Esta ferramenta propões a construção de jogos com visual leve para desenvolvedores iniciantes e avançados. Com esta ferramenta, será possível criar jogos de plataforma, *puzzle*, jogos de espaço. As informações oficiais estão disponíveis no site <http://www.scirra.com/>.

Pixie Engine: Esta *engine* permite o desenvolvimento de jogos mesmo sem muito conhecimento, ela oferece uma coleção de ferramentas simples e elegantes que ajudam na criação de jogos sem aborrecimentos. As informações oficiais podem ser encontradas no site <http://www.pixieengine.com/>.

4 ESTUDO DE CASO: APLICANDO O HTML 5 EM UM PROJETO

O jogo escolhido para demonstrar as técnicas pesquisadas e apresentadas é um jogo com cartas conhecido como jogo de memória ou concentração. A codificação foi inspirada em exemplos encontrados no livro *The Essential Guide to HTML 5*, (MEYER, 2010). O objetivo do projeto é utilizar algumas das características que foram apresentadas na nova versão do HTML.

4.1 CARACTERÍSTICAS DO JOGO E REQUISITOS BÁSICOS

- **Nome do jogo:** Jogo da memória;
- **Tipo de jogo:** *puzzle*;
- **Gráficos:** 2D;
- **Jogadores:** *single player*.

O jogo consiste em um conjunto de cartas com suas faces viradas para baixo, o jogador deve clicar sobre elas a fim de encontrar os pares combinados. Após encontrar a combinação de cartas corretas, ambas desaparecem se estiverem incoerentes, tornam a ficar com a face para baixo deixando apenas alguns segundos para que o jogador memorize as cartas e suas localizações, para a próxima escolha. Quando o jogador encontrar todos os pares de cartas, uma mensagem mostrará quanto tempo foi necessário para a conclusão do jogo.

As tarefas que o jogo realizará são as seguintes:

- Compor a tela com as cartas;
- Embaralhar as cartas para que as combinações fiquem em lugares diferentes a cada partida;
- Identificar o primeiro e o segundo clique do jogador;
- Mostrar a face com a imagem ao clicar na carta;
- Retirar pares que formam uma combinação;
- Estar preparado para ações inesperadas dos usuários.

4.2 DESENVOLVIMENTO

Para o desenvolvimento do jogo foram utilizados alguns recursos do HTML 5 e outros específicos do JavaScript. O principal elemento do HTML utilizado foi o canvas que contém todas as animações e figuras do jogo.

Já na linguagem Javascript, foi definido uma representação para as cartas, onde estão as informações como localização além do estado de sua visualização (frente ou verso).

Durante a codificação do jogo foram desenvolvidas algumas funções principais como virar as cartas, preparar e embaralhar as cartas, além da contabilização do tempo que tem como finalidade solucionar o desafio do jogo. Todos os movimentos do jogador serão captados através do evento *click* dos objetos, este que retorna as coordenadas do clique.

Outras novas funcionalidades do HTML 5 que foram utilizadas é a escrita de textos na tag canvas, através de código para que possa ser concatenado com resultados dinâmicos e os desenhos das cartas.

4.3 CODIFICAÇÃO

Para iniciar a codificação do jogo é necessário criar a estrutura básica do documento HTML. Na Figura 18 é possível observar que a estrutura foi criada com a versão 5 do HTML devido a declaração do DOCTYPE.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Jogo da memória</title>
  </head>
  <body>
  </body>
</html>
```

Figura 18 - Estrutura básica do documento

Como uma das tags mais importantes deste documento, a tag canvas é adicionada entre as tags de abertura e fechamento body. É importante que sua propriedade Id seja informada, pois será através desta propriedade que o código Javascript conseguirá atribuir algumas características e ações. Na Figura 19 é possível observar o destaque para a tag canvas criada. A mensagem que está definida entre as tags só será apresentada ao usuário em caso de não reconhecimento do elemento pelo navegador.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title> Jogo da memória - HTML5</title>
  </head>
  <body>
    <canvas id="jogo" width="1000" height="400">
      Seu browser não tem suporte para o elemento canvas do HTML5
    </canvas>
  </body>
</html>
```

Figura 19 - Posicionamento das tags canvas

Seguindo para o desenvolvimento da lógica do jogo e iniciando a programação com a linguagem Javascript, adicionam-se as tags <script> </script> e entre elas toda a codificação necessária.

Para preparar o desenvolvimento criam-se algumas variáveis que estão representadas na Figura 20 onde é possível verificar qual a funcionalidade de cada variável através do comentário que estão logo acima de cada uma.

```

//variáveis para definir o tamanho da tag canvas
var jogo_width = 1000;
var jogo_height = 400;
//variável que receberá o contexto 2d
var contexto;
var primeira_foto = true;
//variável que armazenará informações sobre a primeira escolha
var primeira_carta = -1;
//variável que armazenará informações sobre a segunda escolha
var segunda_carta;
// variável que configura a cor da carta
var cor_carta = "rgb(255,102,51)";
// variável que configura a cor do fundo do jogo
var cor_tabela = "rgb(51,51,51)";
var deck = [];
//variável que contém a posição x da primeira carta
var firstsx = 30;
//variável que contém a posição y da primeira carta
var firstsy = 50;
// Define a distancia entre as cartas
var margin = 30;
// Define o tamanho da carta
var carta_width = 100;
var carta_height = 130;
var tid;
var matched;
var iniciar_time;
var count = 0;
// Definimos os pares de imagens
var pares = [
["imagem1_1.jpg","imagem1_2.jpg"],
["imagem2_1.jpg","imagem2_2.jpg"],
["imagem3_1.jpg","imagem3_2.jpg"],
["imagem4_1.jpg","imagem4_2.jpg"],
["imagem5_1.jpg","imagem5_2.jpg"]
]

```

Figura 20- Declaração de variáveis que serão utilizadas no jogo

Como próxima etapa do desenvolvimento, será necessário a codificação de uma função que receba as características da carta e atribua às variáveis, esta função servirá como um objeto que será a representação das propriedades da carta, pois conterà sua localização e suas dimensões e informações que será informada ao *array* posteriormente. Na Figura 21 está codificado uma função chamada Carta que assumirá estas características além da função que desenha o verso da carta.

```

function carta(sx,sy,width,height, img, info) {
    this.sx = sx;
    this.sy = sy;
    this.width = width;
    this.height = height;
    this.info = info; //info será usada no array na formação das cartas
    this.img = img;
    this.draw = desenha_verso;
}
function desenha_verso() {
    contexto.fillStyle = cor_carta;
    contexto.fillRect(this.sx,this.sy,this.width,this.height);
}

```

Figura 21 - Função que representa a carta e suas características

Seguindo com a codificação das principais funções do jogo, a função responsável por preparar as cartas com seus respectivos pares está programada para percorrer o *array* com os dados dos pares, atribuir os dados de localização e desenhar a carta no canvas. A Figura 22 exibe a codificação elaborada com estas funcionalidades.

```

function formar_cartas() {
    var i; // variável para o for
    var carta_a; // variável para armazenar a primeira carta
    var carta_b; // variável para armazenar a segunda carta
    var figura_a; // variável para armazenar a primeira imagem
    var figura_b; // variável para armazenar a segunda imagem do par
    var cx = firstsx;
    var cy = firstsy;
    for(i=0;i<pares.length;i++) {
        figura_a = new Image();
        figura_a.src = pares[i][0];
        carta_a = new Carta(cx,cy, carta_width, carta_height, figura_a, i);
        deck.push(carta_a);
        figura_b = new Image();
        figura_b.src = pares[i][1];
        carta_b= new carta(cx, cy+carta_height+margin, carta_width,
carta_height, figura_b, i);
        deck.push(carta_b);
        cx = cx+carta_width+ margin;
        carta_a.draw();
        carta_b.draw();
    }
}

```

Figura 22 - Função que prepara os pares de cartas e desenha-os na tela

A Figura 23 apresenta o código com a função responsável por embaralhar as cartas, esta função faz com que o jogo não seja sempre igual e desta forma o jogador poderá tentar encontrar os pares diversas vezes, e a cada vez, as cartas estarão dispostas de forma diferente.

```
function embaralhar() {
  //variáveis para armazenar referências da carta
  var i;
  var k;
  //variáveis para armazenar informações e imagem da carta (variáveis
  temporárias para a troca)
  var carta_info;
  var carta_img;
  //variável que armazenará a quantidade de cartas
  var dl = deck.length
  var nt;
  for (nt=0;nt<3*dl;nt++) {
    i = Math.floor(Math.random()*dl);
    k = Math.floor(Math.random()*dl);
    carta_info = deck[i].info;
    carta_img = deck[i].img;
    deck[i].info = deck[k].info;
    deck[i].img = deck[k].img;
    deck[k].info = carta_info;
    deck[k].img = carta_img;
  }
}
```

Figura 23 - Função que manipula e embaralha as cartas

Nesta próxima etapa, será desenvolvida a codificação que pode ser considerada parte principal do jogo, pois ela identifica quando os pares são formados e chama a função para virar a carta, se a combinação estiver correta, vira-se a carta com a cor do fundo do jogo e configura seus valores para -1, identificando que a carta já está fora do jogo, caso a combinação estiver incorreta a carta volta para o estado inicial, com sua face virada para baixo. A Figura 24 mostra a programação necessária para essas tarefas.


```

function escolha(ev) {
    var out;
    //variáveis que armazenam a localização (x,y) do mouse
    var mx;
    var my;
    var pick1;
    var pick2;
    // verificação para que as funções do mouse funcione
    // bem em vários navegadores
    if ( ev.layerX || ev.layerX == 0) { // Firefox
        mx= ev.layerX;
        my = ev.layerY;
    } else if (ev.offsetX || ev.offsetX == 0) { // Opera
        mx = ev.offsetX;
        my = ev.offsetY;
    }
    var i;
    for (i=0;i<deck.length;i++){
        var carta = deck[i];
        //verifica se o usuário clica em espaços vazios,
        // onde as cartas já foram removidas
        if (carta.sx >=0)
            if ((mx>carta.sx)&&(mx<carta.sx+carta.swidth)
                &&(my>carta.sy)&&(my<carta.sy+carta.sheight)) {
                //verifica se não queicou duas vezes na mesma carta
                if ((primeira_foto)|| (i!=primeira_carta)) {
                    break;
                }
            }
        }
        if (i<deck.length) {
            if (primeira_foto) {
                primeira_carta = i;
                primeira_foto = false;
                contexto.drawImage(carta.img, carta.sx, carta.sy, carta.swidth,
carta.sheight);
            } else {
                segunda_carta = i;
                contexto.drawImage(carta.img, carta.sx, carta.sy, carta.swidth,
carta.sheight);
                // verifica se elas são pares combinados
                if (carta.info==deck[primeira_carta].info) {
                    combinacao = true;
                    pontos++;
                    contexto.fillStyle= cor_tabela;
                    contexto.fillRect(10,340,900,100);
                    contexto.fillStyle=cor_carta;
                    contexto.fillText("Pares encontrados: " + String(pontos), 10,
380);
                }
                if (pontos>= .5*deck.length) {
                    var now = new Date();
                    var nt = Number(now.getTime());
                    var seconds = Math.floor(.5+(nt-iniciar_time)/1000);
                    contexto.fillStyle= cor_tabela;
                    contexto.fillRect(0,0,900,400);
                    contexto.fillStyle=cor_carta;
                    out = "Parabéns, você encontrou as combinações em " +
String(seconds) + " segundos.";
                }
            }
        }
    }
}

```

```

        contexto.fillText(out,10,100);
        contexto.fillText("Recarregue a página e tente novamente para
obter melhor tempo!!!",10,300);
        return;
    }
    } else {
        combinacao = false;
    }
    primeira_foto = true;
    tid = setTimeout(vira_carta,1000);
}
}
}

function vira_carta() {
    var card;
    //se a formar uma combinação desenha um
    //retângulo com a cor do fundo, se não formar,
    //vira a carta com a fazer para baixo
    if (!combinacao) {
        deck[primeira_carta].draw();
        deck[segunda_carta].draw();
    } else {
        contexto.fillStyle = cor_tabela;
        contexto.fillRect(deck[segunda_carta].sx,
            deck[segunda_carta].sy,
            deck[segunda_carta].swidth,
            deck[segunda_carta].sheight);
        contexto.fillRect(deck[primeira_carta].sx,
            deck[primeira_carta].sy,
            deck[primeira_carta].swidth,
            deck[primeira_carta].sheight);
        deck[segunda_carta].sx = -1;
        deck[primeira_carta].sx = -1;
    }
}
}

```

Figura 24 - Funções principais do jogo

Para finalizar a codificação do jogo, será desenvolvida a função que ficará responsável por adquirir o contexto 2d através da propriedade `Id` informada na tag `canvas`, além de configurar a tela principal e iniciar a contabilização do tempo de jogo. A seguir a Figura 25 apresenta a função carregar e logo abaixo, como será sua chamada diretamente no elemento `onLoad` da tag `body`.

```

function carregar(){
  contexto = document.getElementById('jogo').getContext('2d');
  contexto.fillStyle=cor_tabela;
  contexto.fillRect(0, 0, 1000, 400);
  canvas1 = document.getElementById('jogo');
  canvas1.addEventListener('click',escolha,false);
  formar_cartas();
  embaralhar();
  contexto.font="bold 20pt sans-serif";
  contexto.fillText("Escolha duas cartas e
  faça combinações",10,30);
  contexto.fillText("Pares encontrados: 0",10,380);
  iniciar_time = new Date();
  iniciar_time = Number(iniciar_time.getTime());
}
</script>
</head>
<body onload="carregar();">
  <canvas id="jogo" width="1000" height="400">
    Seu browser não tem suporte para o elemento canvas do HTML5
  </caas>
</body>
</html>

```

Figura 25 - Função que carregará o jogo

A visualização do jogo pode ser vista na Figura 26, e as mensagens com os resultados do jogo podem ser vistas na Figura 27.

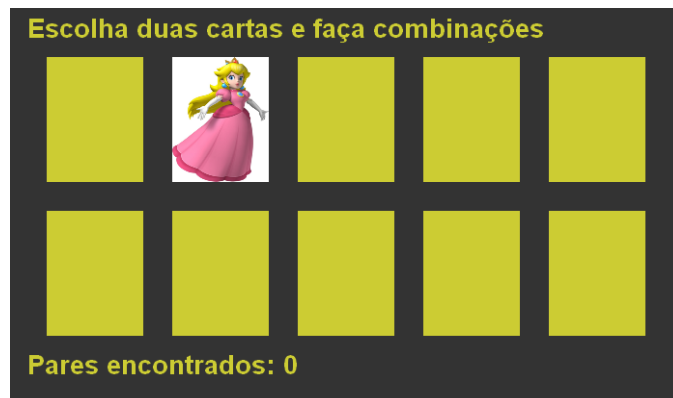


Figura 26 - Tela principal do jogo

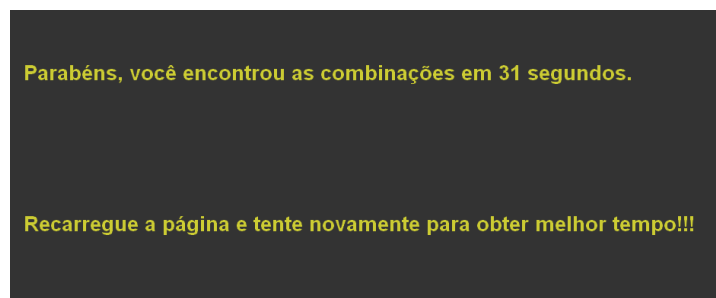


Figura 27 - Tela de resultados com o tempo utilizado para completar o jogo

5 DISCUSSÃO

O jogo desenvolvido no capítulo anterior utilizou os conceitos elementares da linguagem, apenas incorporando o elemento *canvas* na estrutura principal do documento, programação com a linguagem JavaScript e configurando alguns itens de *layout* através de CSS. Este jogo representa um padrão, pois poderá ser executado em todos os computadores e dispositivos móveis que contenham seus navegadores atualizados.

Com o HTML 5 não há necessidade de executar *plugins* de proprietários como o Flash, que é constantemente considerado uma ameaça de segurança devido a seus erros e vulnerabilidades.

Para os desenvolvedores, não será necessário aprender uma linguagem de programação específica para um dispositivo, pois deverá criar aplicativos e jogos para diversos interpretadores. O resultado será jogos de alta qualidade que podem ser jogados em qualquer lugar.

A compatibilidade com os navegadores pode ser considerada um dos pontos mais críticos para a nova linguagem, isso devido ao fato de que seu grande fluxo de atualizações faz com que os navegadores precisem de adaptações para a interpretação correta do aplicativo ou jogo. É por conta deste ponto que alguns jogos podem funcionar perfeitamente em um navegador, e nem se quer carregar a tela principal em outros.

Para desenvolver os jogos que sejam específicos para celulares ou dispositivos móveis são necessárias algumas otimizações referentes à forma de exibição específica para todos os dispositivos. No entanto, mesmo os jogos que não estão projetados com estes requisitos podem ser reproduzidos no navegador de um telefone.

6 CONCLUSÃO

A partir da pesquisa feita, é possível verificar as mudanças no desenvolvimento para a web, seja na construção de simples documentos com interligações ou jogos digitais complexos. Além das mudanças propostas para os desenvolvedores que visam a integração de códigos.

No primeiro capítulo deste documento, é explicitada a história da linguagem de marcação, bem como os estudos que foram adquiridos como forma de apoio da tecnologia. Seguindo para o desenvolvimento de jogos, esse trabalho mostra que é possível desenvolver jogos simples sem a obrigatoriedade da utilização de *plugins* de terceiros, o que garante a web livre sem códigos que exigem instalações.

O ponto crítico levantado como resultado desta pesquisa é a compatibilidade com os navegadores, pois devido a isso há a necessidades de adaptações para a interpretação correta do aplicativo ou jogo. Esta etapa faz parte da evolução da linguagem e será inevitável para se atingir o objetivo final de uma experiência web verdadeiramente livre.

Com base nas pesquisas realizadas para este trabalho, podemos acreditar que a tecnologia não só é adequada quando aplicada ao desenvolvimento de jogos para web, mas que se apresenta, também, como uma boa alternativa de diferencial e especialização do conhecimento aos desenvolvedores que pretendem se lançar no desenvolvimento deste tipo de jogos.

Como trabalhos futuros, evidenciamos a necessidade de que se realizem pesquisas sobre cada biblioteca que possibilita o desenvolvimento de jogos, além do acompanhamento das especificações atualizadas pela W3C.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AWAMURA, Ken. Integrando o Flash com Tecnologias Externas, 2000. Disponível em: <http://pontoflash.com.br/tutoriais_con.php?autor=1&tema=4&post=1>. Acesso em: 30 de outubro 2011.

BISNET, Achilles. Gráficos Vetoriais em Jogos. Disponível em: <http://www.programadoresdejogos.com/trab_academicos/achilles_froes.pdf> Acesso em: 13 Setembro 2011.

BRADLEY, Milton. Star Trek: Phaser Strike - Microvision. Disponível em: <<http://classicgames.about.com/od/toppicks/ig/Best-Classic-Video-Games-1979/Star-Trek--Phaser-Strike.htm>> Acesso em: 29 Setembro 2011.

CAPUTO, Vitor. 6 jogos para conhecer o HTML5, 2010. Disponível em <<http://info.abril.com.br/noticias/blogs/download-da-hora/windows/experimente-seis-jogos-em-html5/>>. Acesso em: 01 de novembro 2011.

CELLEGAR, Tania. Jogos na web e o ensino da história da arte, 2011. Disponível em : <<http://www.revistasusp.sibi.usp.br/pdf/ced/v10n1/v10n1a03.pdf>>. Acesso em: 12 Setembro 2011.

HAWMAN, Jeff and NOLAN, Brian. The History of Air Warrior, 2011. Disponível em: <<http://airwarrior.afkamm.co.uk/AboutAW/history.shtml>>. Acesso em 01 novembro 2011

HTML 5 games. HTML5 Game Engines: Update, 2011. Disponível em <<http://html5games.com/2011/02/html5-game-engines-update//>>. Acesso em: 01 de novembro 2011.

HTML 5 games. HTML5 Game Engines, 2011. Disponível em < <http://html5games.com/resources/>>. Acesso em: 01 de novembro 2011.

HUYSSSEN, Andreas. Reconsidering the High/Low Debate in Global Context: Aesthetics and Politics in an Age of Globalization (Reconsiderando os debates no

contexto global: estética e política na Era da Globalização). Rio de Janeiro, p. 23-24, ago. 2000.

JACKSON, Dean. The W3C Workshop on Web Application and Compound Documents. W3C Interaction domain, 2004. Disponível em: <<http://www.w3.org/2004/04/webapps-cdf-ws/>>. Acesso em: 31 Agosto 2011.

LAWSON, Bruce e SHARP, Remy. Introducing HTML 5. Editora: Pearson Education, 2011

LEE. Tim. Intent Pioneers. Disponível em: <<http://www.ibiblio.org/pioneers/lee.html>>. Acesso em: 02 Setembro 2011.

MENDES, André. Distribuição pela web movimenta mercado independente de games. Disponível em: <<http://www.metodista.br/rronline/noticias/tecnologia/2011/distribuicao-pela-web-movimenta-mercado-independente-de-games>> Acesso em: 13 Setembro 2011.

MEYER, Jeanine. The Essential Guide to HTML5: Using Games to Learn HTML5 and JavaScript. Editora: Apress - ex-Peer Information, 2010.

MURRAY, Janet Horowitz e DAHER, Elissa Khoury. Hamlet no Holodeck: O futuro da narrativa no ciberespaço. São Paulo: UNESP, 2003.

PILGRIM, Mark. HTML5-Up-and-Running. O'Reilly Media, Inc., 2010.

Portal terra Brasil, História dos jogos online, 2009. Disponível em: <<http://www.portalterrabrasil.com/2009/04/historia-dos-jogos-online.html>> Acesso em: 29 Setembro 2011.

PXLEYES. A evolução do Adobe Flash: de 1996 a 2010. Disponível em: <<http://www.pxleyes.com/blog/2010/07/evolution-of-flash-from-1996-to-2010/>>. Acesso em: 04 Outubro 2011.

RAGGETT, Dave. About Me. Disponível em: <<http://www.w3.org/People/Raggett/>>. Acesso em: 01 Setembro 2011.

RAMALHO, José Antônio Alves. Iniciando em HTML. São Paulo: Makron Books do Brasil Editora Ltda., 1996.

REMICK, Jarel. 10 HTML5 Games Paving The Way. Disponível em: <<http://web.appstorm.net/roundups/browsers/10-html5-games-paving-the-way/>> . Acesso em: 06 Outubro 2011.

ROLLINGS, Adams. Fundamentals of Game Design. Disponível em: <http://wps.prenhall.com/bp_gamedev_1/54/14050/3596994.cw/index.html> Acesso em: 12 Setembro 2011.

KAO, Sir. Origens do MMORPG – Capítulo II, 2009. Disponível em: <<http://www.retrofantasy.com.br/origens-do-mmorpg-capitulo-ii/>> . Acesso em :

KHOA, Tommy. Rockman - The Mega Wars, 2010. Disponível em<<http://www.caiman.us/scripts/fw/f1904.html>>. Acesso em: 01 novembro 2011.

WILLIAN, David. A história do HTML. FRONT END BRASIL, 2011. Disponível em: <<http://www.frontendbrasil.com.br/artigos/a-historia-do-html/>>. Acesso em: 31 Agosto 2011..

W3C. Curso Oficial HTML5. Disponível em: <<http://www.w3c.br/cursos/html5/conteudo/>>. Acesso em: 08 Agosto 2011.

W3C. The Secret Origin of SVG. Disponível em: <http://www.w3.org/Graphics/SVG/WG/wiki/Secret_Origin_of_SVG>. Acesso em: 10 Agosto 2011.

W3C. The W3C Workshop on Web Applications and Compound Documents. Disponível em: <<http://www.w3.org/2004/04/webapps-cdf-ws/>>. Acesso em: 01 Setembro 2011.

W3C. HTML5 – Curso W3C Escritório Brasil. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 07 Setembro 2011.

W3Schools. DTD Newspaper Example. Disponível em: <<http://www.w3schools.com/dtd/default.asp>> Acesso em: 19 Setembro 2011

WHATWG Community. Disponível em: <<http://www.whatwg.org/>> Acesso em: 19 Setembro 2011.