

# DESENVOLVIMENTO DE JOGOS PARA A WEB UTILIZANDO O PADRÃO HTML5 E A API CANVAS

SILVA, Rodrigo Cardoso da Cruz<sup>1</sup>

## RESUMO

O desenvolvimento de jogos para a *web* utilizando o padrão HTML5 e a API Canvas se expande rapidamente, pois, é crescente a busca por aplicativos *web* que dispensam a instalação de *softwares* e *plug-ins* na execução de aplicativos e que sejam interoperáveis, que permitam interpretar variados tipos de dispositivos que possuam apenas um navegador *web*. O objetivo desta pesquisa é apresentar a tecnologia HTML5, bem como suas vantagens quando se trata de desenvolvimento de jogos. Os resultados obtidos comprovam que a HTML5 surge como tecnologia de desenvolvimento de jogos, impondo-se ao Adobe Flash em se tratando de jogos *web*, pois, tem como característica principal que os diferencia, a dispensabilidade de *plug-ins*.

**Palavras-chave:** Canvas. Desenvolvimento de jogos. HTML5. *Web*.

## ABSTRACT

The development of games for the web using the HTML5 and Canvas API is rapidly expanding and it is increasing the search web applications that do not require the installation of software and plug-ins in the execution of applications and interoperable, to interpret various types of devices that have only a web browser. The objective of this research is to introduce HTML5 and its advantages when it comes to game development. The results show that the HTML5 technology emerges as game development, imposing upon the Adobe Flash when it refers to web games, because, it has the main feature that differentiates them: dispensability of plug-ins.

**Keywords:** Canvas. Game development. HTML5. *Web*.

---

<sup>1</sup> Rodrigo Cardoso da Cruz Silva é graduado em Sistemas de Informação das Faculdades Integradas de Fernandópolis (FIFE), Fundação Educacional de Fernandópolis (FEF), 2013, Fernandópolis-SP e graduando em Engenharia de Computação do Centro Universitário de Votuporanga (UNIFEV), 2016, Votuporanga-SP, Brasil. E-mail: ro.drigo Cardoso@hotmail.com.

## INTRODUÇÃO

A internet passa por uma revolução tanto na forma como os usuários a utilizam, quanto no desenvolvimento de *sites*. Atualmente, é praticamente impossível referir-se a *sites* sem associá-los a aplicativos, recursos de multimídia como texto, áudio e vídeo e até mesmo jogos.

A navegação na internet está ficando cada vez mais rápida, segura, prazerosa e visualmente atraente aos usuários, devido às inúmeras possibilidades de desenvolvimento. Isso se deve, em grande parte, ao surgimento da quinta geração do padrão HTML. A HTML é uma linguagem de marcação universal, ou seja, é compreendida por diversos meios de acesso, cuja principal função é facilitar a distribuição de informações na *web* de maneira global (FERREIRA; EIS, 2010).

A HTML5 surgiu com grandes mudanças e funcionalidades inovadoras, como novas marcações que facilitam a manipulação de multimídia, elementos e atributos, recursos para formulários, além de propiciar ao desenvolvedor *web* a criação de jogos (TERUEL, 2011).

Na possibilidade de trabalhar com elaboração de *games*, surge o elemento *canvas*, talvez a característica mais interessante da HTML5, que é uma região “desenhável” da página HTML e pode ser manipulada com JavaScript, portanto, possibilita desenhar, processar textos, construir gráficos, manipular e criar animações, jogos e composições de imagens, uma vez que gráfico e animação são aspectos fundamentais dos jogos; *canvas* torna-se, dessa forma, um importante elemento no desenvolvimento de *games* para *web*.

Devido à linguagem ser de fácil suporte aos navegadores e dispositivos móveis, a HTML5 acaba por substituir outras tecnologias de criação de animações interativas, como o Adobe Flash, que, além de consumir excessivamente memória e bateria dos dispositivos, necessita de programas instalados na máquina para dar funcionalidade (TERUEL, 2011).

O objetivo deste artigo é apresentar a tecnologia, bem como suas vantagens quando se trata de desenvolvimento de jogos. No intuito de demonstrar a funcionalidade e algumas possibilidades da *tag Canvas* da HTML5, será desenvolvido um simples *game* como estudo de caso, trabalhando com

recursos como o JavaScript e imagens gerando efeitos ao jogo. Os conceitos, definições e informações contidas neste artigo são fundamentadas em livros e artigos, complementadas com um estudo de caso.

## 1 O PADRÃO HTML

É praticamente impossível referenciar-se a *web* sem associá-la a diversos recursos de interação e multimídia. Com tudo isso, a navegação na internet está atraindo cada vez mais usuários, pelo fato de a *web* fornecer progressivamente mais velocidade, segurança e interatividade, em virtude dos inúmeros recursos disponíveis, quando se trata de desenvolvimento de páginas *web*. Essa evolução deve-se, principalmente, ao surgimento da quinta geração do padrão HTML, que trouxe consigo recursos e funcionalidades inovadoras e, conseqüentemente, novas possibilidades (FERREIRA; EIS, 2010).

O padrão HTML, abreviação de *Hypertext Markup Language*, ou em português, Linguagem de Marcação de Hipertexto, basicamente é uma linguagem para publicação de conteúdo na *web*, que se baseia no conceito de hipertextos, que são conjuntos de elementos conectados. Tais elementos podem ser textos, imagens, vídeos, áudios, documentos, entre outros, que conectados através de referências específicas, conhecidas como *hiperlinks* ou *links*, formam uma grande rede de informação (FERREIRA; EIS, 2010).

Para que a informação seja interpretada universalmente por diversos meios de acesso, é necessária uma linguagem padrão e a HTML se encarrega dessa função. Desde o início, a HTML foi desenvolvida com o propósito de ser uma linguagem que independa de plataformas, *browsers* e outros meios, tornando-a menos custosa, ou seja, interoperável, afinal é necessário apenas um código HTML, ao invés de versões diferentes para cada tipo de dispositivo, ignorando características diferentes, como tamanho de tela, resolução e variação de cor. Assim, evitou-se que a *web* fosse desenvolvida em uma base proprietária, com formatos incompatíveis e limitada (TERUEL, 2011).

Desta forma, foi definido que a HTML deveria ser gerada no formato mais simples, portanto, pode ser gerada por qualquer editor de textos, desde que seja gravada com extensão *.htm* ou *.html* (FERREIRA; EIS, 2010).

### **1.1 O W3C e a evolução da web**

Originalmente, a HTML foi desenvolvida pelo cientista da computação e físico britânico Tim Berners-Lee, ficando conhecida junto ao *browser* Mosaic, desenvolvido por Marc Andreessen na década de 1990, desenvolvedores de *browsers*, então, passaram a se basear na HTML (FERREIRA; EIS, 2010).

Entre 1993 e 1995, a HTML passou pelas versões HTML+, HTML2.0 e HTML3.0. Todavia, a HTML ainda não era tratada como um padrão. Apenas em 1997, o World Wide Web Consortium (W3C), consórcio internacional responsável por manter e desenvolver o padrão da *web*, com o intuito de atingir o potencial máximo da World Wide Web (WWW), criou a versão 3.2 da linguagem, tornando-a prática comum (MEYER, 2011).

Em 1998 a W3C lançou a versão 4 da HTML, que fez a *web* partir da internet estática, a *Web 1.0*, para a *Web 2.0*, passando a proporcionar dinamismo e interatividade. Tempos depois, a HTML já não atendia totalmente às necessidades de recursos de desenvolvimento de aplicações. Diante disso, a W3C concebeu a Extensible Hypertext Markup Language (XHTML), que combina as *tags* da HTML com as rigorosas regras da XML, no entanto, também não agradou. Apesar da recusa da W3C em apoiar a evolução do padrão, algumas empresas como Apple Inc., Mozilla Foundation e Opera Software ASA, fundaram, em 2004, a *Web Hypertext Application Technology Working Group* (WHATWG), pois, desenvolvedores de tais organizações não estavam satisfeitos também com a HTML4, tomando como meta o desenvolvimento da quinta geração da HTML (TERUEL, 2011).

## **2 A QUINTA GERAÇÃO DA HTML**

Em 2006, após o reconhecimento da W3C ao trabalho feito pela WHATWG, a empresa de Tim Berners Lee, que até então, trabalhava separadamente desenvolvendo a XHTML 2, deixou-a em segundo plano e se integrou a WHATWG no projeto de produção da HTML5. Essa veio para dar evolução aos *sites*, tanto no desenvolvimento, como na navegação, sendo suportada pelos principais navegadores do mercado, como o Opera, Mozilla

Firefox, Google Chrome, Safari e, inclusive, o Internet Explorer, que demorou um pouco mais para aderir à nova tecnologia (TERUEL, 2011).

## 2.1 Novos recursos e vantagens da HTML5

A HTML5 traz um conjunto de novos recursos para aprimorar o desenvolvimento de *sites*, as quais incluem *tags* (etiquetas), atributos e interfaces de programação de aplicativos (APIs). Tornando ainda mais simples a criação de conteúdos para a *web* e fornecendo mais alternativas. Diferentemente das suas versões anteriores, fornece ferramentas que facilitam no trabalho com a CSS<sup>2</sup>, DOM<sup>3</sup> e o JavaScript<sup>4</sup>. Essas antigas versões da HTML, de mesmo modo, não possuíam um padrão para a criação de seções de páginas, como artigo, rodapé, cabeçalho, *sidebar*, menus e padrão de nomenclatura de IDs, classes ou *tags* (FERREIRA; EIS, 2010).

Foram adicionados também novos atributos para campos de formulário e tipos de dados específicos, como *date*, *time*, *e-mail*, *url*, *number*, *image*, *search* entre outros, facilitando a validação de formulários através do atributo *pattern* na *tag* `<input>`. Outros novos recursos são os elementos `<audio>` e `<video>`, que são suportes para reprodução de áudio e vídeo respectivamente. Ainda foram acrescentadas APIs que manipulam acesso a banco de dados, validação de formulário, áudio e vídeo, gráficos, desenhos, entre outras (MEYER, 2011).

Apesar das mudanças, a HTML segue com sua grande vantagem de não necessitar de *softwares* e *plug-ins*, que são programas que auxiliam navegadores na execução de tipos especiais de conteúdos na *web*. A partir dessa facilidade, surge a API Canvas, que define uma área "desenhável" da tela, que pode ser manipulada com JavaScript (TERUEL, 2011).

---

<sup>2</sup> Cascading Style Sheets (CSS) ou, em português, Folha de Estilo em Cascata, formata a informação entregue pelo HTML5 (MEYER, 2011).

<sup>3</sup> Document Object Model (DOM), ou, em português, Modelo de Objetos do Documento, é a interface entre a linguagem e os objetos HTML (MEYER, 2011).

<sup>4</sup> JavaScript foi criada pela Netscap em parceria com a Sun Microsystems, com o intuito de fornecer um meio de adicionar interatividade a página *web* (MEYER, 2011).

## 2.2 API Canvas

A HTML5 possui agregada a suas novas funcionalidades a API Canvas, que teve sua origem na Apple Inc., no navegador Safari e, sequencialmente, outros fabricantes de navegadores aderiram à Canvas (GEARY, 2012).

A API consiste em uma região de *bitmap* que possui um contexto 2D, que permite ao desenvolvedor renderizar textos e imagens, ou seja, processar gráficos diretamente para uma área de *bitmap* delimitada no navegador. Possibilita aplicar cores, rotações, manipulações de *pixel*, além de vários tipos de linhas, curvas, caixas, formas, sons, textos e imagens sobre a tela, tudo via JavaScript, possibilitando o desenvolvimento de fantásticas animações, aplicativos e inclusive jogos (GEARY, 2012).

É utilizada a *tag* de marcação `<canvas></canvas>`, que possui apenas os atributos *width* (largura) e *height* (altura), que definem as dimensões do Canvas no HTML, exibindo, assim, apenas o retângulo na tela, onde serão adicionados os elementos definidos no JavaScript. Para desenhar nessa tela, são necessárias coordenadas X e Y, cujo emprego define, respectivamente, linha e coluna. Cada unidade nessas coordenadas equivale a um *pixel* na tela. Portanto, a origem dessa área é situada no canto superior esquerdo, sendo assim, nas coordenadas (0, 0). No diagrama de coordenadas do Canvas, os valores do eixo X aumentam em direção à lateral direita e os valores do eixo Y aumentam em direção à extremidade inferior da área (GEARY, 2012).

Para se trabalhar com Canvas, é necessário definir, previamente, o contexto de renderização a ser aplicado; como, atualmente, o contexto 3D ainda não é aconselhável, por ainda estar em processo de pesquisa, é definido através da função `getContext('2d')` (FULTON; FULTON, 2011).

Como Canvas é propício à renderização de imagens, dentro de suas inúmeras possibilidades, o desenvolvimento de *games* para *web* surge como o ponto mais marcante da API, abrindo um novo mercado de jogos.

## 3 O MERCADO DE DESENVOLVIMENTO DE GAMES WEB

O Flash domina amplamente o mercado de *games* para *web*, porém, a HTML5 vem criando uma nova geração de jogos em navegadores. Isso se deve

principalmente pelo fato de a HTML ser um padrão de código aberto, o que a torna acessível, além de depender de processador e de navegador, pois, praticamente todos fornecem suporte a nova geração (TERUEL, 2011).

A HTML5 também não necessita de *softwares* e *plug-ins* instalados na máquina do usuário, como o que acontece no Flash, para que tais aplicações sejam executadas, o que permite que essas sejam interpretadas pelos variados tipos de dispositivos. Considere-se ainda, o fato de ser uma tecnologia que permite combinação com outras tecnologias como o próprio JavaScript. Nesse mercado a HTML5 conseguiu impor-se justamente pelo fato de os navegadores terem acompanhado essa evolução, disponibilizando mais recursos da tecnologia em cada nova versão do *software*. Por essas circunstâncias, os jogos em HTML5 prometem sobressair por definitivo ao Flash (TERUEL, 2011).

## 4 DESENVOLVIMENTO DO JOGO

Na intenção de demonstrar algumas das inúmeras possibilidades da HTML5, em se tratando de jogos, foi desenvolvido um simples *game* como estudo de caso, trabalhando com recursos, como a *tag Canvas* e o *JavaScript* criando efeito às imagens. O objetivo do jogo é apenas direcionar a barra para direita ou para esquerda de acordo com a posição que a bola cai na tela.

### 4.1 Jogo Pegue a Bola

No jogo foram criadas as imagens do jogador e da bola; a imagem do jogador deverá ser direcionada de acordo com a direção em que a bola cair.

A figura 1 representa as linhas de código do documento HTML. Como primeiro passo, é necessário criar um arquivo de extensão *.htm* ou *.html*, que conterà o código HTML do jogo. Neste arquivo é criado o elemento *Canvas* e é atribuído a ele os valores de largura, altura e um identificador que será chamado no JavaScript. Na *tag body*, o comando *onLoad* chama a função *play*, que será implementada no documento JavaScript.

A figura 2 representa a primeira parte das linhas de código do documento JavaScript. É preciso criar um novo arquivo com extensão *.js* que conterà o código JavaScript. Neste arquivo são instanciadas as variáveis e atribuídos os

valores a elas, criada a função *play()*, que conterà a captura do elemento Canvas e a definição do contexto de renderização. Depois é iniciada a função *keyDown* responsável pelo direcionamento do jogador na tela.

```
<!doctype html>
<html>
  <head>
    <title>Game</title>
    <style type="text/css">
      canvas{
        border: 10px inset #000000;
      }
    </style>
    <script type="text/javascript" src="script.js"></script>
  </head>
  <body onload="play()">
    <canvas id="canvas" width="1000" height="600">
      Seu navegador não suporta Canvas!
      Atualize a versão do seu navegador!
    </canvas>
  </body>
</html>
```

**Figura 1** Jogo Pegue a Bola - Código HTML.  
Fonte: O autor, 2013.

```
//Variáveis
var jogadorAltura, jogadorLargura, jogadorPosicaoX, jogadorVelocidade;
var bolaDiametro, bolaPosicaoX, bolaPosicaoY, bolaVelocidade;
var pontosGanhos, colisao;

function play() //Função que é invocada pelo onLoad
{
  canvas = document.getElementById("canvas"); //Captura o elemento Canvas
  context = canvas.getContext("2d"); //Define o contexto de renderização como 2d

  jogadorAltura = 15; //Define a altura do jogador
  jogadorLargura = 90; //Define a largura do jogador
  pontosGanhos = 0; //Número de pontos atingidos
  jogadorPosicaoX = (canvas.width - jogadorLargura) / 2; //Centraliza o jogador
  jogadorVelocidade = 40; //Velocidade de movimentação do jogador em pixels

  bolaDiametro = 10; //Definindo diametro da bola
  bolaPosicaoX = canvas.width / 2; //centraliza a bola na largura do Canvas
  bolaPosicaoY = -10;
  bolaVelocidade = 11; //Velocidade de movimentação da bola em pixels
  colisao = false;

  document.addEventListener('keydown', keyDown);
  setInterval(gameLoop, 30); //Chama função gameLoop a cada 30 milissegundos
}

function keyDown(e)
{
  if(e.keyCode == 37) //Se código da tecla for 37, moverá o jogador para esquerda
  {
    if(jogadorPosicaoX > 0)
    { //Será diminuida a posição do jogador
      jogadorPosicaoX -= jogadorVelocidade;
    }
  }
}
```

**Figura 2** Jogo Pegue a Bola - Primeira parte do código JavaScript.  
Fonte: O autor, 2013.



A figura 3 representa a segunda e última parte das linhas de código do documento JavaScript, em que é finalizada a função `keyDown` e iniciada a função de *Loop* do jogo, a partir da qual será desenhada a bola, além disso, é realizada a verificação de colisão entre os elementos.

Posteriormente é definido o placar que mostrará a quantidade de pontos atingida. E por fim é desenhado o retângulo que representará o jogador.

```

if(e.keyCode == 39)//Se código da tecla for 39, moverá o jogador para direita
{
    if(jogadorPosicaoX < (canvas.width - jogadorLargura))
    { //Será aumentada a posição do jogador
        jogadorPosicaoX += jogadorVelocidade;
    }
}

function gameLoop()
{
    //Limpa Tela
    context.clearRect(0, 0, canvas.width, canvas.height);

    // Desenhando a Bola
    context.beginPath();
    //Atribuindo ao arc os valores de:
    //(coordenada x, coordenadaY, raio, início do ângulo,
    // fim do ângulo, no sentido horário/anti-horário)
    context.arc(bolaPosicaoX, bolaPosicaoY, bolaDiametro, 0, Math.PI * 2, true);
    context.fill();

    if(bolaPosicaoY <= canvas.height)
    {
        bolaPosicaoY += bolaVelocidade;
    }
    else
    {
        bolaPosicaoX = Math.random() * 1000;
        bolaPosicaoY = -10;
        colisao = false;
    }

    // Checar Colisão
    if((bolaPosicaoX > jogadorPosicaoX && bolaPosicaoX < jogadorPosicaoX + jogadorLargura)
        && bolaPosicaoY >= canvas.height - jogadorAltura && colisao == false)
    {
        pontosGanhos++;
        colisao = true;
    }

    // Escreve placar
    context.font = "50pt Verdana"; //Definindo fonte do placar
    //Definindo posição do placar para x e y do Canvas
    context.fillText(pontosGanhos, canvas.width - 1000, 70);

    // Desenhando o Jogador
    context.fillStyle = "rgb(255,0,0)"; //Definindo a cor vermelha
    //Definindo os valores do jogador que será um retângulo
    //Atribuindo ao fillRect os valores de:
    //(coordenada x, coordenadaY, largura do retângulo, altura do retângulo)
    context.fillRect(jogadorPosicaoX, canvas.height - jogadorAltura, jogadorLargura, jogadorAltura)
}

```

**Figura 3** Jogo Pegue a Bola - Segunda parte do código JavaScript.  
**Fonte:** O autor, 2013.

## 4.2 Análise dos resultados

Os resultados obtidos comprovam o objetivo principal da pesquisa: a HTML5 surge como plataforma de desenvolvimento de jogos *web*, impondo-se ao Adobe Flash e outras tecnologias; a quinta geração do padrão HTML apresenta a característica principal que os tornam diferentes; a dispensabilidade de *plug-ins*. Não há necessidade, pois, de programas instalados no dispositivo ao computador para acelerar e operar o *game*, nem tampouco qualquer *plug-in* que venha dificultar ou “retardar” as respostas no jogo. Dessa forma, exige apenas um navegador para executar os jogos.

## CONSIDERAÇÕES FINAIS

A mais recente versão da HTML disponibiliza novas possibilidades ao desenvolvedor *web* e dentre elas, tem talvez como maior destaque a *tag* Canvas. O elemento Canvas é dimensionável e define uma região “desenhável” na página *web*, permitindo a manipulação de imagens através do JavaScript.

A pesquisa teve como foco principal demonstrar a funcionalidade e algumas possibilidades da *tag* canvas da HTML5. Como estudo de caso, foi desenvolvido um simples *game*, trabalhando com recursos, como o próprio JavaScript, e com imagens gerando efeitos ao jogo. O jogo e sua codificação são muito simples, todavia, demonstram que com a HTML5 é possível desenvolver jogos para a internet sem a necessidade de instalação de *plug-ins*, garantindo interoperabilidade. O desenvolvimento de jogos com HTML5 e Canvas tem como ponto negativo a dificuldade de se encontrar materiais que auxiliem o desenvolvedor iniciante em seus trabalhos com essa tecnologia, contudo, a codificação é bem simples de compreender.

## REFERÊNCIAS

FERREIRA, Elcio; EIS, Diego. **HTML5**: Curso W3C Escritório Brasil. W3C, 2010. Disponível em: <<http://www.w3c.br>>. Acesso em: 05 abr. 2013.

FULTON, Steve; FULTON, Jeff. **HTML 5 canvas**: native interactivity and animation for the *web*. Sebastopol (USA) : O'Reilly, 2011. Disponível em: <<http://pt.scribd.com>>. Acesso em: 18 abr. 2013.

GEARY, David. **Core HTML 5 canvas**: graphics, animation and game development. Crawfordsville (USA) : Prentice Hall, 2012. Disponível em: <<http://baixaaquilivros.blogspot.com.br>>. Acesso em: 06 jun. 2013.

MEYER, Jeanine. **O guia essencial do HTML 5**: usando jogos para aprender HTML5 e JavaScript. Rio de Janeiro : Ciência Moderna, 2011.

TERUEL, Evandro Carlos. **HTML5**: guia prático. São Paulo : Érica, 2011.