CAPÍTULO 1

Introdução à SVG

Neste capítulo, faremos uma introdução à tecnologia SVG apresentando sua definição e suas finalidades. Mostraremos sua evolução relatada em um breve histórico – desde sua criação até os dias atuais. Mostraremos como os navegadores atuais se comportam com relação ao suporte para as diferentes funcionalidades da SVG, apresentaremos a marcação XML mínima para criação de gráficos com uso de SVG e, no final do capítulo, estudaremos o uso do sistema cartesiano para definir formas SVG.

1.1 Introdução

SVG é a sigla em inglês para *Scalable Vector Graphics*, que em tradução livre significa Gráficos Vetoriais Escaláveis.

Os três formatos tradicionais de gráficos destinados à inserção em aplicações web são GIF, JPG e PNG. Esses formatos são do tipo *bitmap* (mapa de bits), segundo o qual um gráfico é formado por um conjunto de pixels, cada um deles com sua cor. Assim, nesses tipos de gráficos, a informação necessária à renderização do gráfico é armazenada levando-se em conta cada pixel individualmente. É fácil imaginar a quantidade de pixels (informação) necessária para criar um gráfico simples.

As extensões GIF, JPG e PNG referem-se aos diferentes mecanismos de compressão daquelas informações, mecanismos estes que são proprietários e, por isso mesmo, de código fechado.

Gráficos adotando um dos formatos citados degradam (perdem qualidade de imagem) quando redimensionados. Um gráfico do tipo GIF, JPG ou PNG com dimensões iguais a 100x100px é capaz de ficar ilegível quando escalado para 500x500px.

A técnica de criação de gráficos vetoriais adota um mecanismo diferente, segundo o qual as informações sobre o gráfico estão contidas em vetores que nada mais são do que linhas unindo dois pontos. Esse mecanismo permite que se escale um vetor sem que as informações sejam perdidas ou "distorcidas", e em consequência, quando se escala um gráfico, sua qualidade permanece inalterada. Um gráfico com dimensões 10000x10000px apresenta a mesma qualidade (resolução) do gráfico original criado com dimensões de 50x50px, por exemplo.

Vários editores de imagem, tais como Photoshop, Fireworks, FreeHand, Illustrator e outros, são capazes de gerar gráficos SVG. Por outro lado, existem editores específicos para SVG, sendo os mais populares o SVG-Edit e o Inkscape, ambos gratuitos.

Gráficos SVG são descritos por vetores. Vetores são linhas (retas ou curvas) que unem dois pontos. Assim, para criar um gráfico vetorial, devemos descrever uma sequência de pontos formando um contorno aberto ou fechado e definir espessura e cor dos vetores, bem como cor e efeitos do preenchimento.

A maneira encontrada para descrever um gráfico segundo essas condições foi com uso de marcação XML. Os elementos da marcação definem o formato do vetor e seus atributos definem espessura, cor etc.

Podemos então afirmar que SVG é uma aplicação XML. Observe a marcação mostrada a seguir, que ilustra o que acabamos de estudar.

```
<line x1="0" y1="0" x2="90" y2="320" stroke-width="5" stroke="red"/>
```

Caso você não esteja familiarizado com a sintaxe XML, não estranhe a falta de espaço antes da barra de fechamento da tag. O espaço só é exigido em marcação HTML compatível com XML. Em XML, não há necessidade do espaço.

Mesmo que o leitor nunca tenha estudado SVG, com a simples leitura da marcação mostrada é possível imaginar o resultado no navegador.

Uma linha reta com espessura de 5px na cor vermelha unindo os pontos de coordenadas (0,0) e (90,320).

O elemento `line` define uma linha reta; os atributos `xn` e `yn` definem coordenadas cartesianas; o atributo `stroke-width` define a espessura da linha; e o atributo `stroke`, a cor da linha.

Nos capítulos seguintes, estudaremos os elementos e atributos da SVG.

1.2 Histórico

1.2.1 SVG no W3C

SVG é uma tecnologia em desenvolvimento pelo W3C, e seu status atual é de Recomendação do W3C.

O primeiro Rascunho de Trabalho para SVG 1.0 foi lançado em 11 de fevereiro de 1999. No mesmo ano foram lançados mais sete Rascunhos de Trabalho, e no ano seguinte, mais dois.

Em 2 de agosto de 2000, a especificação alcançou o status de Candidata à Recomendação. Em 19 de julho de 2001 foi lançada a versão Proposta de Recomendação, e em 30 de outubro do mesmo ano tornou-se uma Recomendação do W3C.

No dia 30 de outubro de 2001 foi anunciada a disposição do W3C em criar a versão SVG1.1 e paralelamente a versão SVG Tiny e SVG Basic destinada a dispositivos móveis.

No ano de 2002 foram lançados três Rascunhos de Trabalho da SVG 1.1. Em 30 de abril de 2002, a especificação atingiu o status de Candidata à Recomendação, e em 15 de novembro do mesmo ano, o de Proposta de Recomendação.

Finalmente, em 14 de janeiro de 2003, a especificação foi lançada como Recomendação SVG 1.1 de W3C.

Em 22 de junho de 2010 foi lançado o Rascunho de Trabalho para a SVG 1.1 – Segunda Edição. Em 12 de maio de 2011 foi lançada mais uma versão do Rascunho de Trabalho, e em 9 de junho do mesmo ano, essa versão atingiu o status de Proposta de Recomendação; finalmente, em 16 de agosto de 2011, tornou-se a Recomendação do W3C para SVG 1.1 – Segunda edição.

Neste livro, estudaremos as funcionalidades previstas na Recomendação do W3C para SVG 1.1 – Segunda Edição, que é a versão em vigor atualmente.

A especificação SVG Tiny e SVG Basic, também conhecida como SVG Mobile, é um subconjunto da Recomendação do W3C para SVG, descrevendo as funcionalidades aplicáveis a dispositivos móveis.

O primeiro Rascunho de Trabalho dessa especificação foi lançado em 30 de outubro de 2001.

No ano de 2002 foram lançados dois Rascunhos de Trabalho; e em 30 de abril de 2002, atingiu o status de Candidata à Recomendação. No dia 15

de novembro do mesmo ano foi elevada ao status de Proposta de Recomendação; e em 14 de janeiro de 2003 tornou-se uma Recomendação do W3C como subconjunto da SVG. A recomendação foi editada em 15 de junho de 2009 e é a versão em vigor atualmente.

1.2.2 Tecnologias SVG

Em 1985, a Adobe criou uma linguagem de programação denominada Post Script, independente de dispositivo e destinada a proporcionar qualidade de impressão superior para gráficos e textos.

Mídias destinadas a impressões gráficas e dotadas da programação de impressão da Adobe são capazes de produzir gráficos e textos de alta qualidade, independentemente do tamanho dos gráficos.

A linguagem Post Script não se destina à Web e nem é viável sua instalação em dispositivos para a web em consequência do grande tamanho dos arquivos. No entanto, logo foi percebida a necessidade de uma maneira de descrever gráficos e textos que os fizesse escaláveis para uso na web.

Em 1998, a Microsoft lançou uma linguagem de programação – baseada em XML – denominada VML (Vector Markup Language), destinada a web e capaz de criar gráficos vetoriais escaláveis. Tal linguagem não sensibilizou a comunidade, e a Microsoft abandonou seu desenvolvimento.

Em abril de 1998, um time constituído por representantes da Adobe, da IBM, da Netscape e da Sun produziu e enviou como proposta ao W3C um documento contendo as diretrizes de criação da linguagem de marcação baseada em XML, denominada PGML (Precision Graphics Markup Language), destinada a descrever gráficos vetoriais.

Em maio do mesmo ano, um time constituído por representantes da HP, da Macromedia, da Microsoft e da Visio produziu e enviou como proposta ao W3C um documento contendo as diretrizes de criação de uma linguagem de marcação baseada em XML, denominada VML (Vector Markup Language), destinada a descrever gráficos vetoriais.

Como consequência dos dois documentos recebidos, o W3C lançou em outubro de 1998 a abertura de uma lista de propostas para desenvolvimento da SVG. No início de 1999, o W3C lançou o primeiro Rascunho de Trabalho da SVG, vislumbrando a necessidade de uma maneira não proprietária de servir imagens na web sem perda de qualidade por escalonamento.

Na mesma época, a IBM e o Corel lançaram softwares capazes de exportar gráficos no formato SVG. Iniciativa essa que foi imediatamente seguida por vários fabricantes de softwares para diferentes dispositivos.

A seguir, a IBM lançou um SVG Viewer, inaugurando a era de crescimento da SVG.

1.3 Vantagens da SVG

As principais vantagens da SVG sobre os formatos GIF, JPG, PNG e bitmaps em geral são relacionadas a seguir.

- Arquivos bem menores demandando menor tempo de carregamento e consumo de largura de banda.
- O escalonamento dos gráficos para acomodação em diferentes dispositivos e resoluções não causa a perda de qualidade (fenômeno conhecido como pixelização do gráfico).
- Os gráficos são construídos no dispositivo do usuário (normalmente o navegador), baseados em um arquivo de texto XML recebido do servidor. Isso resulta em necessidade de comunicação cliente-servidor muito mais ágil e simples do que a exigida para servir um gráfico do tipo mapa de bits.
- Por ser construído com marcação XML, os gráficos são acessíveis pelo DOM, facilitando a tarefa de manipulação com scripts a um nível que seria praticamente impossível com uso de um gráfico do tipo mapa de bits.
- Por serem acessíveis pelo DOM, gráficos em documentos diferentes podem interagir e responder à JavaScript.
- SVG está em conformidade com a Recomendação XML para Namespaces.
- SVG está em conformidade com XLink, XPointer, CSS, SMIL e RDF.

1.4 Suporte à SVG

Os principais navegadores modernos, com exceção do IE8 e dos anteriores, oferecem bom suporte para as funcionalidades básicas da SVG, e o suporte às funcionalidades avançadas como filtros e animações complexas vem crescendo a cada dia.

O site http://caniuse.com mantém uma página constantemente atualizada mostrando o suporte dos navegadores às funcionalidades da SVG. A página encontra-se em http://caniuse.com/#cats=SVG.

1.4.1 SVG Web

SVG Web é uma biblioteca JavaScript lançada em julho de 2009 e que à época em que este livro foi escrito estava em sua versão beta. Ela permite o uso efetivo de SVG em desenvolvimento web, pois, uma vez instalada e devidamente linkada aos documentos que utilizam SVG, provê suporte nos seguintes navegadores e dispositivos:

- Internet Explorer 6+
- Firefox 2+
- Safari 3+
- iPhone 2.1+
- Webkit
- Chrome

Para funcionar, a biblioteca requer que o navegador do usuário tenha instalado o Flash 10+, pois o suporte à SVG é suprido com uso dessa tecnologia para navegadores que não dão suporte nativo. Estatísticas dão conta de que 97% dos usuários web têm o Flash 10+ instalado, e assim o uso da biblioteca é seguro e recomendável.

Nos navegadores Firefox 3+, Safari 3+, Chrome, iPhone 2.1+, Webkit e IE9+, a renderização SVG se faz com suporte nativo, e a biblioteca é ignorada.

Você poderá verificar se a biblioteca está ou não sendo usada em um determinado navegador clicando com o botão direito do mouse sobre um gráfico SVG existente na página. Se o suporte for nativo, o menu de contexto que abrirá será o típico menu de quando se clica em um lugar qualquer da página. Se o suporte for pela biblioteca, abrirá um menu com links personalizado pela SVG Web e mais links típicos de objetos em Flash.

Para mais informações sobre o funcionamento e o uso da biblioteca, consulte http://code.google.com/p/svgweb/. Um manual do usuário da biblioteca está disponível em http://svgweb.googlecode.com/svn/trunk/docs/UserManual.html.

1.5 Estrutura XML

Veremos a seguir a estrutura mínima de um documento XML destinado a criar um gráfico SVG. Note que a sintaxe descrita segue as regras da sintaxe XML, prevista na Recomendação do W3C para XML 1.0.

1.5.1 Prólogo XML

A marcação de documentos XML deve obrigatoriamente começar com o chamado prólogo XML, cuja sintaxe mais simplificada declara apenas a versão da XML com a qual a marcação está em conformidade. Observe a seguir o prólogo XML mínimo.

```
<?xml version="1.0"?>
```

O prólogo XML pode conter outros atributos, além da versão do XML, que fornecem informações adicionais sobre o documento, como, por exemplo, a declaração `standalone` e a declaração `encoding` para a codificação de caracteres. A declaração `standalone` admite os valores `no` e `yes`, conforme a marcação dependa ou não, respectivamente, de uma marcação externa declarativa para ser convenientemente processada pelo parser XML. A ausência dessa declaração assume por padrão que o seu valor é `no`, ou seja, havendo marcação externa declarativa, ela será considerada no parseamento da marcação XML.

Entenda-se por marcação externa declarativa um documento público contendo um conjunto de instruções para parseamento XML (uma DTD) criadas por terceiros e necessárias para parseamento de uma marcação XML personalizada.

A declaração `encoding` define a codificação de caracteres adotada no documento. Em XML, o padrão geral – e a recomendação – é a adoção de *utf-8*.

Observe a seguir um prólogo XML com as duas declarações que acabamos de estudar.

```
<?xml version="1.0" standalone="yes" encoding="utf-8"?>
```

1.5.2 DOCTYPE XML

As Recomendações do W3C para SVG 1.1 estão divididas em três módulos, a saber:

- SVG 1.1 Full (módulo completo)
- SVG 1.1 Basic (módulo simplificado para dispositivos móveis)

- SVG 1.1 Tiny (módulo simplificado do módulo Basic)

Para cada um desses módulos existe um DOCTYPE com a sintaxe mostrada a seguir.

- **SVG 1.1 Full**

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/
DTD/svg11.dtd">
```

- **SVG 1.1 Basic**

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Basic//EN" "http://www.w3.org/Graphics/
SVG/1.1/DTD/svg11-basic.dtd">
```

- **SVG 1.1 Tiny**

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Tiny//EN" "http://www.w3.org/Graphics/
SVG/1.1/DTD/svg11-tiny.dtd">
```

Observe a seguir um exemplo de marcação do prólogo e DOCTYPE para um documento SVG destinado a desktop.

```
<?xml version="1.0" standalone="yes" encoding="utf-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

Ao contrário da HTML e da XHTML servida como `text/html`, a presença do DOCTYPE em documento XML não tem qualquer influência sobre o modo de renderização do documento. Dessa forma, o uso de DTD em documento XML é facultativo.

A DTD para SVG 1.1 Full é constituída de duas partes:

- **Identificador público para SVG 1.1**

```
PUBLIC "-//W3C//DTD SVG 1.1//EN"
```

- **Identificador de sistema para a Recomendação SVG 1.1**

```
http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd
```

O identificador de sistema aponta para um documento hospedado no site do W3C descrevendo a sintaxe da linguagem XML e é usado, também, para fins de validação. Esse documento é modularizado, ou seja, está dividido em vários outros documentos, significando que o validador terá que processar os vários documentos. E aqui é bom lembrar que documento XML inválido causa erro fatal e, como consequência, o processamento é interrompido e o resultado é uma mensagem de erro no agente de usuário.

As Recomendações do W3C para SVG 1.1 afirmam que a validação de documentos XML que adotam uma DTD é problemática em razão da forma como DTDs manipulam namespaces, e em consequência, embora elas mencionem as DTDs para SVG, recomendam NÃO usar DTD em documentos XML para SVG.

1.5.3 Elemento raiz

Todo documento XML deve conter um elemento raiz, e o elemento raiz de documentos SVG é o elemento `svg` com suas tags de abertura e fechamento `<svg></svg>`.

O elemento raiz deve conter a declaração de *namespace* `xmlns="http://www.w3.org/2000/svg"`, e se o gráfico usar links em sua marcação, deve-se incluir também o *namespace* `xmlns:xlink="http://www.w3.org/1999/xlink"`.

Adicionalmente, é de boa prática declarar as dimensões da área ocupada pelo gráfico com uso das declarações `width` e `height` e, logo após, o elemento raiz usar os elementos `<desc></desc>` e `<title></title>` destinados a fornecer uma descrição sumária do gráfico e declarar um título para o documento, respectivamente.

Assim, a estrutura mínima recomendada para escrever um documento *svg* é conforme a mostrada a seguir.

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink" width="nnn" height="nnn">
    <desc>Breve descrição do gráfico</desc>
    <title>Título do documento</title>
    <!-- código XML do gráfico -->
</svg>
```

Um documento XML que cria um gráfico SVG poderá ser salvo com a extensão xml ou svg.

1.6 Inserção SVG em página HTML

Gráficos SVG são criados com uso de marcação XML, e o documento contendo a marcação XML que constrói o gráfico deve ser salvo com a extensão *.svg*. Assim, inserir um gráfico SVG em um documento HTML implica inserir marcação XML dentro de marcação HTML.

Os elementos da linguagem HTML que normalmente são usados para inserir SVG (marcação XML) são: `embed`, `object`, `iframe`, `img` e `script`.

O elemento `embed` funciona, mas é inválido em linguagem XHTML; e se você faz questão de que seu documento valide, não use esse elemento em documentos XHTML. Contudo, em HTML5, o elemento `embed` é válido, portanto, poderá ser usado para inserir SVG sem invalidar a marcação.

O elemento `object` não é suportado pelo IE8 e pelos anteriores.

O elemento `iframe` e o elemento `img` são suportados por todas as versões de navegadores e validam em todas as marcações (X)HTML.

A seguir, mostraremos exemplos de códigos destinados a inserir um gráfico SVG em um documento HTML com uso de cada um dos elementos citados.

1.6.1 Elemento `embed`

Para inserir um arquivo SVG com uso desse elemento, use a marcação mostrada a seguir:

```
...
<body>
    <!-- marcação HTML da página -->
    <embed type="image/svg+xml" src="../grafico.svg"/>
    ...
</body>
</html>
```

O atributo `type` define o tipo de conteúdo inserido pelo elemento `embed` e o valor `image/svg+xml` informa ao navegador que se trata de uma imagem SVG. O atributo `src`, nosso velho conhecido, mostra o caminho em que se encontra o arquivo.

1.6.2 Elemento `object`

Para inserir um arquivo SVG com uso desse elemento, use a marcação mostrada a seguir:

```
...
<body>
    <!-- marcação HTML da página -->
    <object type="image/svg+xml" data="../grafico.svg">
        <p>Conteúdo alternativo ao gráfico SVG</p>
    </object>
```

```
    ...
</body>
</html>
```

O atributo `type` define o tipo de conteúdo inserido pelo elemento `object` e o valor `image/svg+xml` informa ao navegador que se trata de uma imagem SVG. O atributo `data`, nosso velho conhecido, mostra o caminho em que se encontra o arquivo.

O elemento `object` permite definir um conteúdo alternativo para navegadores que não o suportam, como consta no código mostrado anteriormente.

1.6.3 Elemento `iframe`

Para inserir um arquivo SVG com uso desse elemento, use a marcação a seguir:

```
...
<body>
    <!-- marcação HTML da página -->
    <iframe width="ll" height="hh" src="../grafico.svg">
        <p>Conteúdo alternativo ao gráfico SVG</p>
    </iframe>
    ...
</body>
</html>
```

Os atributos `width` e `height` definem as dimensões do elemento `iframe` e devem ser compatíveis com as dimensões do gráfico a inserir. O atributo `src`, nosso velho conhecido, mostra o caminho em que se encontra o arquivo.

O elemento `iframe` permite definir um conteúdo alternativo para navegadores que não o suportam, como consta no código mostrado anteriormente.

1.6.4 Elemento `img`

Para inserir um arquivo SVG com uso desse elemento, use a marcação a seguir:

```
...
<body>
    <!-- marcação HTML da página -->
    <img width="ll" height="hh" src="../grafico.svg" alt=" "/>
    ...
</body>
</html>
```

Os atributos `width` e `height` definem as dimensões do gráfico a inserir. O atributo `src`, nosso velho conhecido, mostra o caminho em que se encontra o arquivo.

1.7 Sistema cartesiano

A criação de gráficos SVG, basicamente, consiste na definição de uma série de pontos em um espaço bidimensional. Pontos esses que, quando interligados, definem uma forma. Por exemplo: dois pontos definem uma reta; três pontos não colineares definem um triângulo; um ponto e um comprimento definem uma circunferência, e assim por diante.

Em nosso velho conhecido sistema cartesiano, as formas citadas são representadas como mostrado na figura 1.1:

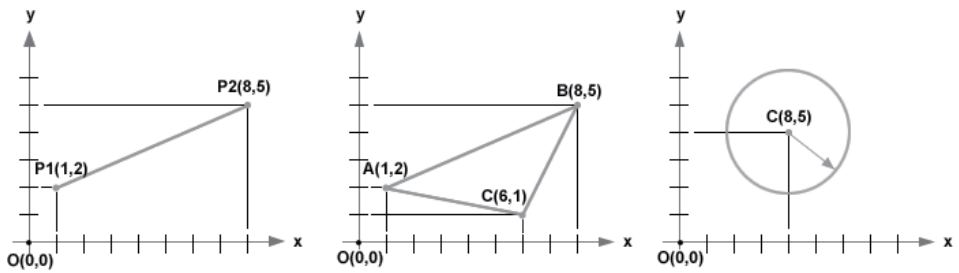

Figura 1.1 – Gráficos no sistema cartesiano.

Tanto as formas simples como as complexas são definidas como uma sucessão de pontos.

Uma das características da linguagem de marcação XML é que podemos "inventar" ou criar os nomes dos elementos da marcação à nossa escolha. Poderíamos, por exemplo, inventar um elemento denominado `<livro></livro>` para armazenar informações sobre um livro.

Da mesma forma, poderíamos inventar um elemento denominado linha `<line></line>` para armazenar informações sobre uma linha, ou um elemento denominado círculo `<circle></circle>` para armazenar informações sobre um círculo.

Assim, para criar as três formas mostradas nos gráficos cartesianos da figura 1.1, poderíamos escrever as marcações XML mostradas a seguir:

```
<!-- linha -->
    <line x1="1" y1="2" x2="8" y2="5"></line>
<!-- triângulo -->
    <line x1="1" y1="2" x2="8" y2="5"></line>
    <line x1="8" y1="5" x2="6" y2="1"></line>
    <line x1="6" y1="1" x2="1" y2="2"></line>
<!-- círculo -->
    <circle cx="8" cy="5" r="2"></circle>
```

O sistema cartesiano para a web é semelhante ao sistema cartesiano matemático. A única diferença é o ponto origem O(0,0). Enquanto, na matemática, o quadrante das coordenadas positivas tem sua origem no canto inferior esquerdo, como mostrado na figura 1.1, na web, a origem é o canto superior esquerdo da tela (ou da área de renderização) com a coordenada horizontal x positiva no sentido da esquerda para a direita e a coordenada vertical y positivo no sentido de cima para baixo, como mostrado na figura 1.2.

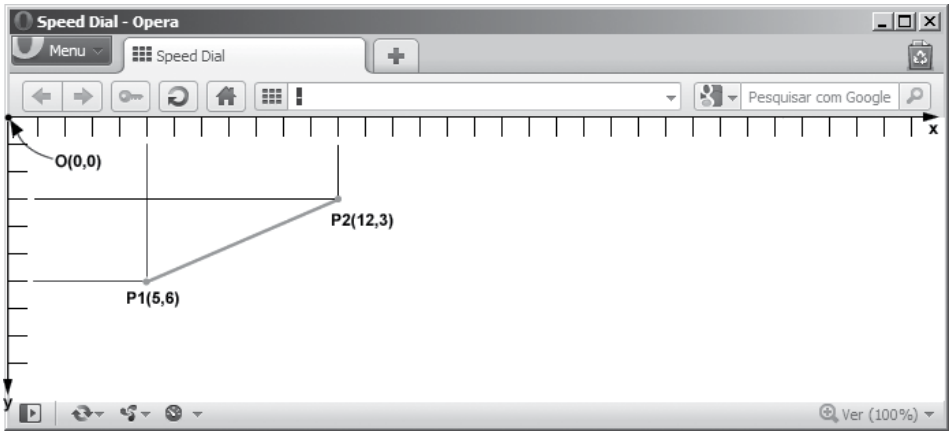

Figura 1.2 – Sistema cartesiano na web.

Tal como no sistema cartesiano da matemática, na web, também são admitidos valores negativos para as coordenadas. Declarar um valor de *x* negativo implica posicionar o ponto fora da tela e à esquerda, e uma coordenada *y* negativa posiciona fora da tela e acima. Note ainda que, para posicionar fora da tela à direita ou abaixo, basta declarar coordenadas com valor maior do que as dimensões horizontal e vertical da tela (a resolução).

Mas por que alguém iria querer posicionar um gráfico fora da tela? Isso poderá ser bastante útil em animações quando pretendermos obter um efeito de entrada de um objeto gráfico na tela.