

jQuery Mobile

**Desenvolva aplicações web para dispositivos
móveis com HTML5, CSS3, AJAX, jQuery e jQuery UI**

Maurício Samy Silva

Copyright © 2012 da Novatec Editora Ltda.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Revisão gramatical: Marta Almeida de Sá

Editoreção eletrônica: Carolina Kuwabata

Capa: Maurício Samy Silva/Carolina Kuwabata

ISBN: 978-85-7522-295-9

Histórico das impressões:

Fevereiro/2012 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Fax: +55 11 2950-8869

E-mail: novatec@novatec.com.br

Site: www.novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec

**Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)**

Silva, Maurício Samy
jQuery Mobile : desenvolva aplicações web para dispositivos móveis com HTML5, CSS3, Ajax, jQuery e jQuery UI / Maurício Samy Silva. -- São Paulo : Novatec Editora, 2012.

Bibliografia
ISBN 978-85-7522-295-9

1. Computação móvel 2. jQuery Mobile 3. Web sites - Desenvolvimento I. Título.

12-00220

CDD-005.2762

Índices para catálogo sistemático:

1. jQuery : Mobile : Desenvolvimento de aplicações
WEB : Computadores : Processamento de dados
005.2762
VC20120118



CAPÍTULO 1

Apresentação do jQuery Mobile

Neste capítulo, faremos a apresentação do framework jQuery Mobile. Veremos sua definição, um breve histórico de seu aparecimento e as configurações básicas para iniciar o desenvolvimento com uso do framework. jQuery Mobile é rotulado pelos seus criadores como um framework. Neste livro, ao fazermos referência ao framework (masculino), usaremos “o jQuery Mobile”.

Ao longo dos textos e das explicações contidas no livro, para não repetir exaustivamente o termo “framework jQuery Mobile” o abreviaremos para simplesmente: *framework*. Assim, neste livro, salvo indicação contrária, *framework* é o mesmo que *framework jQuery Mobile*.

1.1 Introdução

1.1.1 Definição

jQuery Mobile é um framework para desenvolvimento web, otimizado para interação por toque e que se destina à criação de aplicações para smartphones e tablets.

Seu desenvolvimento está voltado para fornecer mecanismos capazes de criar sistemas unificados de interface de usuário, baseados em HTML5 e CSS3 e capazes de ser funcionais em todas as plataformas móveis tendo como base de construção as bibliotecas jQuery e jQuery UI. Tais sistemas não apenas incluem funcionalidades para criação do layout (listas, painéis, layers etc.), mas também disponibilizam um rico conjunto de controles e *widgets* (sliders, toggles, abas etc.).

O framework aplica o princípio “fazer mais escrevendo menos”, que rege o desenvolvimento da biblioteca jQuery, com a finalidade de criar aplicações *únicas* para todos os dispositivos móveis e sistemas operacionais.

1.1.2 Suporte

Uma das diretrizes de desenvolvimento do framework estabelece que se trata de uma ferramenta capaz de desenvolver aplicações que sejam suportadas pela grande maioria dos modernos navegadores desktop e das plataformas para smartphones, tablets e e-readers.

Graças aos princípios da incrementação progressiva e da degradação graciosa, destinados a não bloquear acesso aos conteúdos, levados em conta pelo framework, navegadores antigos e dispositivos com limitadas possibilidades são também capazes de renderizar convenientemente as aplicações desenvolvidas com seu uso.

Para fins de suporte, o grupo de desenvolvimento do framework dividiu os diversos dispositivos móveis de usuário em três grupos, nomeando-os com as letras A, B e C. São classificados no grupo A os sistemas com suporte pleno às funcionalidades criadas pelo framework. No grupo B estão os sistemas com suporte pleno, exceto para AJAX; e, no grupo C, os sistemas com suporte para funcionalidades básicas.

Os dispositivos testados nos laboratórios do grupo de criação do framework e suas respectivas classificações nos grupos de suporte são mostrados nas tabelas a seguir.

Grupo A – Suporte pleno, com animação de páginas baseada em AJAX	
Plataforma	Equipamentos testados
Apple iOS 3.2-5.0 beta	Dispositivos originais iPad (3.2/4.3), iPad 2 (4.3), iPhone (3.1), iPhone 3 (3.2), 3GS (4.3), e 4 (4.3/5.0 beta)
Android 2.1-2.3	HTC Incredible (2.2), Droid (2.2) original, Nook Color (2.2), HTC Aria (2.1), Google Nexus S (2.3). Funcionais em 1.5 & 1.6, mas com performance lenta. Testado no Google G1 (1.5)
Android Honeycomb	Samsung Galaxy Tab 10.1
Windows Phone 7	HTC 7 Surround
Blackberry 6.0	Torch 9800 e Style 9670
Blackberry 7	BlackBerry Torch 9810
Blackberry Playbook	PlayBook versões 1.0.1/1.0.5
Palm WebOS (1.4-2.0)	Palm Pixi (1.4), Pre (1.4), Pre 2 (2.0)

Palm WebOS 3.0	HP TouchPad
Firefox Mobile (Beta)	Android 2.2
Opera Mobile 11.0	iPhone 3GS e 4 (5.0/6.0), Android 2.2 (5.0/6.0), Windows Mobile 6.5 (5.0)
Meego 1.2	Nokia 950
Kindle 3	Built-in WebKit browser incluído no dispositivo Kindle 3
Chrome Desktop 11-13	OS X 10.6.7 e Windows 7
Firefox Desktop 3.6-4.0	OS X 10.6.7 e Windows 7
Internet Explorer 7-9	Windows XP, Vista e 7 (pequenas inconsistências CSS)
Opera Desktop 10-11	OS X 10.6.7 e Windows 7

Grupo B – Suporte pleno, exceto as funcionalidades AJAX para navegação	
Plataforma	Equipamentos testados
Blackberry 5.0	The Storm 2 9550, Bold 9770
Opera Mini (5.0-6.0)	Testado no iOS 3.2/4.3
Nokia Symbian^3	Nokia N8 (Symbian^3), C7 (Symbian^3), e também funciona no N97 (Symbian^1)

Grupo C – Suporte básico, sem incrementos HTML, contudo funcional	
Plataforma	Equipamentos testados
Blackberry 4.x	Curve 8330
Windows Mobile	Testado no HTC Leo (WinMo 5.2)
Todas as plataformas antigas para smartphone e featurephones	Todos os dispositivos sem suporte para media queries foram enquadrados neste grupo.

Ainda não classificado	
Plataforma	Equipamentos testados
Samsung Bada	O projeto desse dispositivo ainda não dispõe de dispositivos para testes ou emuladores, contudo o suporte parece ser razoável. Suporte para a versão 1.0 é indeterminado.

1.1.3 Histórico

Em 11 de agosto de 2010, John Resig, o criador da biblioteca jQuery, anunciou ao mundo que junto com uma equipe de colaboradores estava desenvolvendo um framework baseado nas bibliotecas jQuery e jQuery UI e destinado ao desenvolvimento para dispositivos móveis. Anunciou também que estavam apoiando o projeto a Palm Inc., a Mozilla, a Blackberry, a DeviceAtlas, a Nokia, a Adobe e o grupo Filament.

John Resig anunciou naquela ocasião que a estratégia geral para o desenvolvimento do framework visava a criar uma interface unificada para servir aos navegadores mais usados para smartphones e uma programação JavaScript topo de linha.

Em 16 de outubro de 2010, John Resig anunciou o lançamento da primeira versão alfa do framework, o jQuery Mobile 1.0 alfa 1, ou jQM 1.0a1, com funcionalidades para desenvolver o seguinte:

- tema e layout;
- topo e rodapé (fixos e persistentes);
- controles de formulário;
- caixas de diálogo;
- listagens;
- eventos;
- transições.

Em 12 de novembro de 2010, John Resig anunciou o lançamento da segunda versão alfa do framework, o jQuery Mobile 1.0 alfa 2, ou jQM 1.0a2. Essa versão corrigiu bugs e estendeu a versão anterior nas seguintes funcionalidades:

- eventos;
- API e métodos;
- manipulação de formulários com AJAX;
- manipulação de URL e transições;
- configuração global e layout.

Em 4 de fevereiro de 2011, John Resig anunciou o lançamento da terceira versão alfa do framework, o jQuery Mobile 1.0 alfa 3, ou jQM 1.0a3. Essa versão estendeu a versão anterior nos seguintes itens:

- o grupo de desenvolvimento estudou e solucionou cerca de 480 sugestões;
- foram corrigidos cerca de 150 bugs;
- foram criadas 240 unidades de testes e quase todas passaram na maioria dos navegadores;

- o core do framework e os plugins de navegação foram refatorados e aperfeiçoados em vários aspectos. Foram aperfeiçoados também os mecanismos que permitem dar consistência às APIs dos plugins;
- incremento do suporte a navegadores tipo Grupo A, de modo a incluir o Firefox Mobile (Fennec) e o Opera Mobile/Mini, e refinamento do suporte para iOS, Android, BlackBerry 6, Palm WebOS e para a maioria dos navegadores modernos em suas versões para desktop;
- foi incorporada a funcionalidade datepicker da jQuery UI otimizada para interação por toque (ela não constava, por padrão, do projeto inicial);
- incremento de funções de teclado e de suporte geral para acessibilidade no VoiceOver;
- conseguido suporte quase total no Grupo A para Windows Phone 7 e Nokia;
- incontáveis aperfeiçoamentos e otimizações em várias áreas.

Em 31 de março de 2011, Todd Parker anunciou o lançamento da quarta e última versão alfa do framework, o jQuery Mobile 1.0 alfa 4, ou jQM 1.0a4. Essa versão corrigiu bugs e estendeu a versão anterior, principalmente no que diz respeito ao suporte e aos mecanismos para evitar colisões com funcionalidades da HTML5 e de outros frameworks. Convém notar ainda a implementação de funcionalidades que tornaram nativos os menus de seleção.

Em 7 de abril de 2011, Todd Parker anunciou o lançamento de uma versão de manutenção da versão alfa anterior, o jQuery Mobile 1.0 alfa 4.1, ou jQM 1.0a4.1.

Em 20 de junho de 2011, Todd Parker anunciou o lançamento da primeira versão beta do framework, o jQuery Mobile 1.0 beta 1, ou jQM 1.0b1. A grande novidade dessa versão foi o suporte para Blackberry 5 e Opera Mini versão beta, além de uma série de melhorias no framework.

Em 3 de agosto de 2011, Todd Parker anunciou o lançamento da segunda versão beta do framework, o jQuery Mobile 1.0 beta 2, ou jQM 1.0b2. A grande novidade dessa versão foi o suporte para a série 60 dos smartphones da Nokia, além de uma série de melhorias no framework.

Em 29 de setembro de 2011, Todd Parker anunciou o lançamento da primeira versão RC (candidata a lançamento) do framework, o jQuery Mobile 1.0 RC1, ou jQM RC1. A grande novidade dessa versão foi o suporte para a plataforma Meego, além de uma série de correções de bugs e de refinamentos e otimizações no framework.

Em 19 de outubro de 2011, Todd Parker anunciou o lançamento da segunda versão RC (candidata a lançamento) do framework, o jQuery Mobile 1.0 RC2, ou jQM RC2. A grande novidade dessa versão foi a expansão do suporte para os controles do tipo `date`, `time` e `color` da HTML5, além de uma série de correções de bugs e de refinamentos e otimizações no framework.

Em 13 de novembro de 2011, Todd Parker anunciou o lançamento da terceira versão RC (candidata a lançamento) do framework, o jQuery Mobile 1.0 RC3, ou jQM RC3. A grande novidade dessa versão foi a melhoria da performance geral do framework.

Em 16 de novembro de 2011, Todd Parker anunciou o lançamento da versão 1.0 do framework, o jQuery Mobile 1.0, ou jQM 1.0. Essa é a versão 1.0 final para ser usada exclusivamente com a biblioteca jQuery 1.6.4. O suporte para a versão 1.7 da biblioteca jQuery virá com a versão 1.1 do framework.

1.2 Visão geral

Forneceremos uma visão geral dos princípios de desenvolvimento com uso do framework jQuery Mobile. Iniciaremos mostrando as configurações necessárias à criação de um ambiente de desenvolvimento compatível e capaz de ser manipulado pelas funcionalidades do framework e, a seguir, demonstraremos seus princípios e suas premissas de funcionamento desenvolvendo códigos simples para criação de páginas que podem ser visualizadas no site do livro ou em dispositivos móveis.

O framework foi criado e otimizado para uso com marcação HTML5, CSS3 e as bibliotecas jQuery e jQuery UI.

1.2.1 Instalação

Nenhuma instalação é requerida para desenvolvimento com o framework. Basta criar, na marcação HTML do documento, um link para a biblioteca jQuery, outro para o arquivo do framework e mais um para a folha de estilos padrão do framework. Somente isso e estaremos em condições de tirar o máximo proveito das funcionalidades e principalmente da simplicidade de desenvolver para praticamente todos os dispositivos móveis existentes no mercado sem preocupações com modelos, marcas, plataformas ou sistemas operacionais.

Você pode fazer o download dos três arquivos citados anteriormente, hospedá-los em um servidor e linká-los para o endereço de hospedagem, contudo os criadores do framework advogam que é uma boa prática que os links apontem para o endereço de hospedagem dos arquivos em um CDN (Content Distribution Network – rede de distribuição de conteúdos). A escolha entre hospedagem própria e CDN é do desenvolvedor.

1.2.1.1 CDN

CDN é um sistema de computadores, contendo cópias de um arquivo de dados, colocados em diferentes nós de uma rede. Um CDN bem planejado e implementado tem inúmeras vantagens sobre o armazenamento em um servidor próprio; dentre elas, destacamos as seguintes: melhoria sensível do acesso aos dados em cache e conseqüentemente redução do consumo de banda; incremento do paralelismo, ou seja, da capacidade de oferecer acesso simultâneo; redução da latência servindo os dados de um servidor mais próximo do usuário. Os dados mais comuns em cache nos CDNs, para uso público, são os objetos web e objetos para download, tais como arquivos de mídia, softwares e documentos, aplicações, streaming de mídia e resultados de consultas a bancos de dados.

Os arquivos jQuery estão hospedados em três CDNs distintos, que são os seguintes:

- **Google Libraries API:** disponível em: <http://kwz.me/Yf> (<http://code.google.com/apis/libraries/devguide.html>);
- **Microsoft Ajax Content Delivery Network:** disponível em: <http://kwz.me/Yj> (<http://www.asp.net/ajaxlibrary/cdn.ashx>);
- **jQuery CDN:** disponível em: <http://kwz.me/Yq> (<http://code.jquery.com/>).

Cabe ao desenvolvedor escolher o CDN ou optar por hospedar os arquivos em servidor próprio, prática que não aconselhamos, a não ser que se tenha uma boa razão para essa escolha.

Neste livro adotaremos a jQuery CDN, seguindo a sugestão do grupo que desenvolve o framework, mas repito: o desenvolvedor é livre para escolher aquele que bem entender. Se você pretende analisar e comparar as vantagens e desvantagens deste ou daquele CDN, bem como conhecer os detalhes para criação dos links para os arquivos, recomendamos a leitura da documentação hospedada no Blog jQuery CDN, em <http://kwz.me/Yv> (<http://www.jquerycdn.net/>), e a documentação dos CDNs nos três endereços web listados anteriormente.

1.2.1.2 Template mínimo

O código a seguir mostra um template mínimo para desenvolver uma página para dispositivo móvel, e em destaque no código mostramos a maneira correta de se linkar ao documento os arquivos necessários para uso do framework.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Livro jQuery Mobile do Maujor</title>
  <link rel="stylesheet"
    href="http://code.jquery.com/mobile/latest/jquery.mobile.min.css" />
  <script src="http://code.jquery.com/jquery.min.js"></script>
  <script src="http://code.jquery.com/mobile/latest/jquery.mobile.min.js"></script>
</head>
<body>
<!-- marcação da página de acordo com as diretrizes do framework-->
</body>
</html>
```



Esse template mínimo, salvo indicação expressa em contrário, será usado em todos os exemplos mostrados neste livro, e fica desde já implícito e entendido que qualquer trecho de código mostrado como exemplo neste livro é supostamente inserido nesse template.

Metatag viewport

Para os que estão estranhando a presença da metatag `viewport` no template mínimo mostrado anteriormente ou ainda não conhecem sua finalidade, começamos dizendo que essa metatag é fundamental para o funcionamento do código em dispositivos móveis. Ela informa ao navegador do dispositivo móvel que o código que se segue deve ser renderizado levando em conta as dimensões da viewport do dispositivo, bem como define como será o zoom, característico em tais dispositivos.

Essa metatag foi inventada pela Apple para o navegador Safari destinado a dispositivos móveis e atualmente é suportada pela maioria dos navegadores e sistemas operacionais dos dispositivos móveis. A sintaxe geral para essa metatag é mostrada a seguir:

```
<meta name="viewport" content="width, height, user-scalabe, inicial-scale, maximum-scale,
  minimum-scale" />
```

O atributo `content` desse elemento `meta` admite uma lista de pares nome/valor definindo as características da viewport do dispositivo móvel, conforme descrito a seguir:

Característica	Descrição	Valores possíveis
<code>width</code>	Indica a largura da viewport	número de pixels
<code>height</code>	Indica a altura da viewport	número de pixels
<code>user-scalable</code>	Indica se o usuário poderá ou não dar zoom	no e yes
<code>inicial-scale</code>	Fator inicial para o zoom no documento	números inteiros
<code>maximum-scale</code>	Fator máximo para zoom no documento	números inteiros
<code>minimum-scale</code>	Fator mínimo para zoom no documento	números inteiros

Não é obrigatório constar todos os valores para o atributo. Observe, a seguir, alguns exemplos de sintaxe válida para essa metatag:

```
<meta name="viewport" content="width=320px, height=480px, user-scalable=yes,
    inicial-scale=1.0, maximum-scale=2, minimum-scale=0.5" />
<meta name="viewport" content="width=480px, height=640px, inicial-scale=1" />
<meta name="viewport" content="width=240px, inicial-scale=0.5" />
<meta name="viewport" content="width:680x, user-scalable=no" />
```

Segue um exemplo prático do uso dessa metatag para definir a largura do dispositivo móvel.

Suponha que não tenha sido definida a metatag `viewport` no código de uma página destinada a renderizar um texto marcado com o elemento `p`, conforme o código a seguir:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8" />
  <title>Livro jQuery Mobile do Maujor</title>
<style>
  p {
    background: red;
    color: white;
  }
</style>
</head>
<body>
  <p>Wiewport width<br />Não definida</p>
</body>
</html>
```

[<http://jqmovel.com.br/c1/1>]

A seguir, vamos definir explicitamente três larguras de viewport para a mesma página, conforme as alterações a seguir:

- **Largura igual a 200px**

```
...
  <meta name="viewport" content="width=200px" />
...
<body>
  <p>Viewport width<br />content="width=200px"</p>
</body>
</html>
```

[<http://jqmovel.com.br/c1/2>]

- **Largura igual a 320px**

```
...
  <meta name="viewport" content="width=320px" />
...
<body>
  <p>Viewport width<br />content="width=320px"</p>
</body>
</html>
```

[<http://jqmovel.com.br/c1/3>]

- **Largura igual a 500px**

```
...
  <meta name="viewport" content="width=500px" />
...
<body>
  <p>Viewport width<br />content="width=500px"</p>
</body>
</html>
```

[<http://jqmovel.com.br/c1/4>]

Os arquivos estão disponíveis no site do livro para visualização em um dispositivo móvel. Na figura 1.1, mostramos a renderização de cada um dos quatro arquivos.

Quando não se declara a metatag `viewport` na marcação HTML, o navegador do dispositivo móvel interpreta a marcação como sendo para desktop. Nesse caso, tudo se passa como se estivéssemos abrindo, no dispositivo móvel, um site desenvolvido para desktop. A renderização da largura total do site será

mostrada na largura da viewport. Observe esse caso (arquivo <http://jqmovel/c1/l>) mostrado no canto superior esquerdo da figura 1.1 e note que o texto do parágrafo é ilegível.



Figura 1.1 – Metatag viewport para textos.

Qual a razão da ilegibilidade? A resposta a essa pergunta envolve considerações que vão desde a resolução das telas de dispositivos móveis até o conceito de densidade de pixel. Não entraremos nos detalhes técnicos que respondem à pergunta, mas daremos uma resposta bem simples e básica. Se você estiver interessado nos detalhes técnicos, consulte os endereços <http://kwz.me/gR> e <http://kwz.me/lI>.

Por padrão, navegadores de dispositivos móveis são projetados para resoluções que variam em torno de 900px (de 800px a 980px na maioria dos casos). Imagine um site projetado para desktop com largura de 900px sendo renderizado em um dispositivo móvel. Acomodar os 900px em uma tela de 320px somente será possível aplicando zoom para diminuir até caber, causando ilegibilidade, como mostrado.

Ao contrário, se o site foi projetado com largura de 200px, o zoom terá que ser para aumentar até caber nos 320px. É o que demonstram as quatro telas que constam da figura 1.1.

Na figura 1.2, mostramos a renderização de uma imagem com dimensões iguais a 320px x 212px inserida em uma página web em quatro situações diferentes quanto à declaração da metatag `viewport`. A declaração da metatag como consta da marcação de cada página é mostrada na própria figura. No primeiro caso, a imagem diminuída; no segundo, aumentada com barra de rolagem horizontal; no terceiro, tem suas dimensões reais; e, no quarto, a maior diminuição da imagem.

Os códigos das páginas que demonstram as quatro situações descritas estão disponíveis no site do livro, nos arquivos:

[<http://jqmovel.com.br/c1/5>]

[<http://jqmovel.com.br/c1/6>]

[<http://jqmovel.com.br/c1/7>]

[<http://jqmovel.com.br/c1/8>]

É óbvio que não basta declarar a metatag `viewport` para servir uma imagem qualquer para dispositivos móveis. Muitas outras variáveis estão em questão. O exemplo simplesmente demonstra na prática o efeito da metatag `viewport`.

Dispositivos móveis não são padronizados com uma resolução de tela igual a 320px, como mostrado na figura 1.2. Smartphones e tablets possuem resoluções que vão de 240px a mais de 1000px. Além disso, muitos dispositivos móveis possibilitam leitura nas posições *portrait* (retrato, ou na vertical) e *landscape* (paisagem, ou na horizontal) com larguras diferentes.

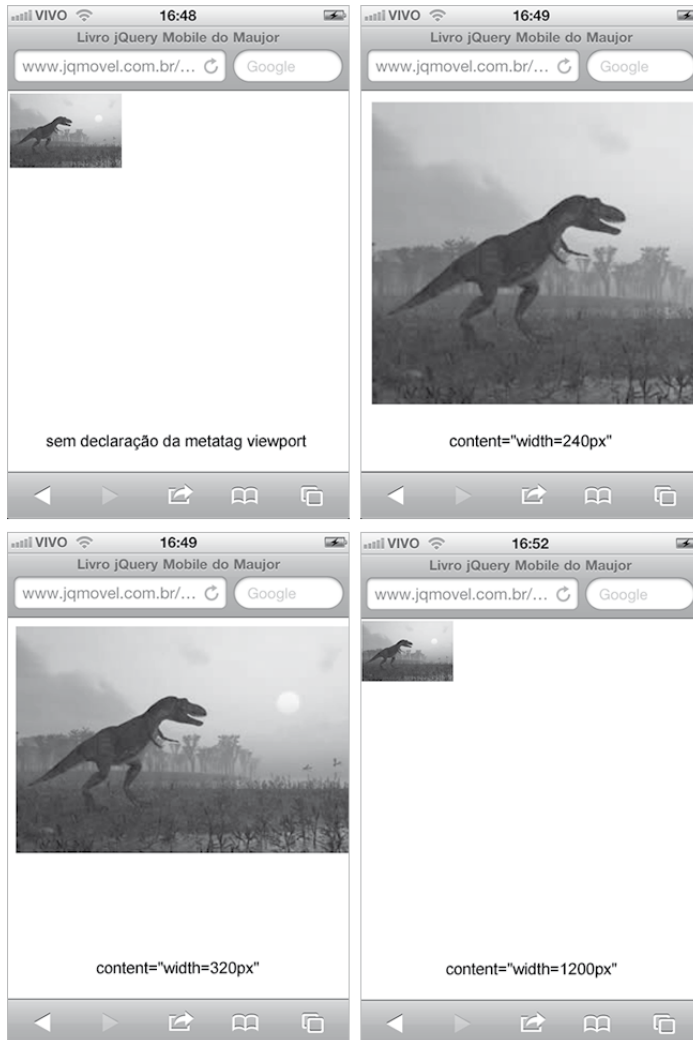


Figura 1.2 – Metatag viewport para imagens.

Felizmente, a sintaxe para declaração de largura do dispositivo móvel na metatag viewport admite um valor que não fixa uma largura em pixels, mas vale para a largura da viewport do dispositivo móvel, seja ela qual for. Essa sintaxe é mostrada a seguir:

```
<meta name="viewport" content="width=device-width" />
```

Declarar `width=device-width` equivale a dizer para o dispositivo móvel o seguinte: os conteúdos deste documento devem ser mostrados em uma largura igual à largura disponível neste dispositivo.

Para nosso template mínimo mostrado em [1.2.1.2], a declaração dessa metatag se fará com uso dos valores para a largura da viewport e para a escala inicial de renderização, como mostrado a seguir:

```
<meta name="viewport" content="width=device-width, inicial-scale=1" />
```

Arquivos externos

Para usar o framework, precisamos linkar, obrigatoriamente, três arquivos externos à página a ser servida para dispositivos móveis, conforme mostrado a seguir:

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/latest/jquery.mobile.min.css " />  
<script src="http://code.jquery.com/jquery.min.js"></script>  
<script src="http://code.jquery.com/mobile/latest/jquery.mobile.min.js "></script>
```

A ordem dos links é como mostrada e não deve ser alterada. Primeiro, declare o link para folha de estilos, depois, para a biblioteca jQuery, e, por último, para o framework jQuery Mobile.

1.2.2 Características

As principais características do framework jQuery Mobile são listadas a seguir:

- Foi desenvolvido tendo como base a biblioteca jQuery e incorporando os mecanismos e as funcionalidades da jQuery UI. Sua sintaxe, baseada na da biblioteca jQuery, proporciona aprendizado rápido, sem exigir programação sofisticada, como é característico da programação com uso da jQuery.
- É compatível com a maioria dos dispositivos móveis, tablets, e-readers e das plataformas desktop – iOS, Android, Blackberry, Palm WebOS, Nokia/Symbian, Windows Phone 7, MeeGo, Opera Mobile/Mini, Firefox Mobile, Kindle, Nook e todos os navegadores modernos com suporte aos três grupos de classificação dos navegadores, como mostrado em [1.1.2].
- Tamanho reduzido e mínima dependência de imagens propiciam carregamento rápido.
- Arquitetura modular permite seleção personalizada dos componentes de acordo com as necessidades de cada aplicação.

- Marcação HTML5 para estrutura e inserção de comportamento nas páginas, resultando em desenvolvimento ágil e mínima dependência de scripts.
- Uso de técnicas que permitem desenvolver aplicações ricas com máximo proveito das funcionalidades avançadas para dispositivos modernos sem bloquear acesso aos dispositivos mais antigos.
- Técnicas de desenvolvimento que permitem escalar automaticamente a interface das aplicações para servir tanto a smartphones e tablets quanto a dispositivos para desktop.
- Mecanismos de navegação baseados em AJAX permitem animar transição de páginas sem comprometer a funcionalidade de botões de retrocesso, o armazenamento de favoritos e o controle de URLs.
- Funcionalidades para acessibilidade com inclusão dos mecanismos da WAI-ARIA, assegurando correta leitura das páginas por leitores de tela (por exemplo: VoiceOver no iOS), e outras tecnologias assistivas.
- Suporte para interação por toque, para eventos de mouse e de foco com uso de uma API simples.
- Conjunto padronizado de widgets para interface de usuário otimizados para interação por toque, estilizáveis, fáceis de usar, multiplataforma e capazes de substituir controles nativos.
- Framework para aplicação de temas, ThemeRoller, personalizável e fácil de usar.
- Uso de uma API própria, robusta e completa, capaz de fornecer funcionalidades para desenvolvimento de scripts de personalização para todos os mecanismos do framework.

1.2.3 Módulos

O framework foi desenvolvido adotando uma arquitetura modular. Os módulos foram divididos em dois grandes grupos: componentes e API, conforme listados a seguir.

- **Módulos componentes:**
 - Módulo para páginas e caixas de diálogo.
 - Módulo para barras de ferramentas.

- Módulo para botões.
- Módulo para formatação de conteúdos.
- Módulo para elementos de formulários.
- Módulo para listas.
- **Módulos API:**
 - Módulo para configurações padrão.
 - Módulo para eventos.
 - Módulo para métodos e utilidades.
 - Módulo para temas.