

Casa do Código

Livros para o programador

© 2013, Casa do Código

Todos os direitos reservados e protegidos pela Lei nº9.610, de 10/02/1998.

Nenhuma parte deste livro poderá ser reproduzida, nem transmitida, sem autorização prévia por escrito da editora, sejam quais forem os meios: fotográficos, eletrônicos mecânicos, gravação ou quaisquer outros.

Casa do Código

Livros para o programador

Rua Vergueiro, 3185 - 8º andar

04101-300 – Vila Mariana – São Paulo – SP – Brasil



Casa do Código
Livros para o programador

**Uma editora de livros técnicos
feita por desenvolvedores
para desenvolvedores.**



Inscreva-se em nossa newsletter e
receba novidades e lançamentos

www.casadocodigo.com.br/newsletter



Curta nossa fanpage no Facebook

www.facebook.com/casadocodigo

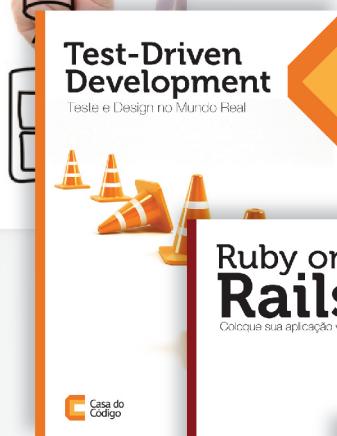


Caelum:
Cursos de TI presenciais e online

www.caelum.com.br



Dê seu feedback sobre o livro. Escreva para contato@casadocodigo.com.br



E muito mais em:
www.casadocodigo.com.br

Sumário

1	Progressive Enhancement: por um site mais acessível e mais fácil de man-	
	ter	1
1.1	Por onde começar?	3
1.2	Progressive Enhancement e JavaScript	3
1.3	Progressive Enhancement e CSS	3
1.4	Progressive Enhancement e HTML	3
1.5	Quando o Progressive Enhancement não é uma alternativa	3
	Índice Remissivo	3
	Bibliografia	5

Versão: 16.8.28

CAPÍTULO 1

Progressive Enhancement: por um site mais acessível e mais fácil de manter

Com navegadores cada vez mais modernos, cheios de recursos, a tendência é que nossos sites também fiquem cada vez mais sofisticados. Porém, não podemos esquecer: nem todo mundo que acessa nossos sites está usando um navegador com os últimos recursos.

O primeiro pensamento que aparece na cabeça de muitos quando ouvem algo parecido é: mas eu posso obrigar meus usuários a atualizarem seus navegadores. Ou então: usuário com navegador desatualizado não merece acessar meu site! Mas será que sempre podemos exigir navegadores atualizados dos nossos usuários? E será que navegadores desatualizados são o único problema?

Quando fazemos um site, normalmente o fazemos para um público imenso de pessoas. Não podemos esquecer que, nesse público imenso, temos pessoas que não

gostam de JavaScript sendo executado em suas máquinas, pessoas que **não podem** atualizar seus navegadores, pessoas acessando a Internet a partir de dispositivos limitados e pessoas com dificuldades motoras, visuais e auditivas que nem sempre conseguem utilizar o mouse para navegar ou dependem de leitores de tela para terem acesso ao nosso conteúdo. Mais ainda: temos um tipo de usuário muito importante para considerar, que não tem JavaScript nem CSS habilitados: as ferramentas de busca.

Isso significa, então, que não podemos fazer um site moderno e acessível ao mesmo tempo? Seria absurdo se a tecnologia evoluísse dessa forma. É por isso que os grupos que controlam o desenvolvimento das tecnologias que usamos na Internet, como o W3C, têm sempre em mente a preocupação com esses casos citados acima. Mas como podemos desenvolver sites levando em conta esses cenários?

Uma primeira forma de pensar é desenvolver seu site primeiro para o público geral, que tem acesso aos navegadores mais modernos e não tem restrições de acessibilidade para, num segundo momento, procurar atender aos usuários com mais limitações. Essa forma de pensar está presente há muitos anos no desenvolvimento de sistemas e a ideia de criar sistemas tolerantes a falhas; no mundo de desenvolvimento *front-end*, essa prática ficou mais conhecida como *graceful degradation*.

No entanto, pensar dessa forma pode nos levar a alguns problemas. Vamos pegar um exemplo para analisar: um menu com submenus que, por padrão, ficam escondidos. Podemos fazê-lo apenas com HTML e CSS:

```
<ul class="menu">
  <li class="menu-item"><a href="#">Item 1</a>
    <ul class="submenu">
      <li><a href="#">Item 1.1</a></li>
      <li><a href="#">Item 1.2</a></li>
      <li><a href="#">Item 1.3</a></li>
    </ul>
  </li>
  <li class="menu-item"><a href="#">Item 2</a>
    <ul class="submenu">
      <li><a href="#">Item 2.1</a></li>
      <li><a href="#">Item 2.2</a></li>
    </ul>
  </li>
</ul>
```

Para fazer os submenus ficarem escondidos por padrão, podemos usar a seguinte

regra:

```
.submenu {  
    display: none;  
}
```

e podemos mostrá-lo quando o usuário passar o mouse em cima:

```
.menu .menu-item:hover > .submenu {  
    display: block;  
}
```

Isso resolve nosso problema para a maior parte dos usuários. Agora paramos para pensar: quais casos não atendemos? Podemos começar lembrando dos usuários que não conseguem usar o mouse. Uma solução seria oferecer navegação por teclado também. Para conseguir isso, precisamos de um pouco de JavaScript.

1.1 POR ONDE COMEÇAR?

1.2 PROGRESSIVE ENHANCEMENT E JAVASCRIPT

1.3 PROGRESSIVE ENHANCEMENT E CSS

1.4 PROGRESSIVE ENHANCEMENT E HTML

1.5 QUANDO O PROGRESSIVE ENHANCEMENT NÃO É UMA ALTERNATIVA

Referências Bibliográficas