

Melhores práticas para um site mais rápido

Um pequeno resumo da documentação da extensão YSlow para o Firefox, encontrado em inglês em: <http://developer.yahoo.com/performance/rules.html>

A extensão, que mede o desempenho de sites no Firefox, pode ser encontrada em: <https://addons.mozilla.org/pt-BR/firefox/addon/5369>

Obs.: o original é bem mais detalhado e com muito mais informações, aqui trago apenas o que achei mais interessante ou importante.

80% do tempo gasto pelo visitante para baixar uma página é gasto no front-end (cliente) e somente 20 no back-end (servidor). Desses 80% a maioria é gasto baixando componentes: imagens, CSS, script JS, Flash, etc.

Reduzindo o número de componentes do site reduzirá as requisições HTTP para renderizar a página no cliente. Esta é a chave de páginas rápidas.

Simplificar o design das páginas reduzirá o número de componentes.

Abaixo algumas técnicas para reduzir o número de requisições e ao mesmo tempo manter um design rico.

Usar CSS sprites reduz o número de requisições de imagens, ao invés de requisitar várias requisita-se somente uma e o CSS mapeia usando o background-image e background-position. Exemplo em: <http://ribafs.org/portal/colaboracoes/1-ribamar-fs/123-sprites-em-css> Combinar, quando possível todo o CSS em um único arquivo externo.

Mova o CSS para o topo da página, entre as tags < head> e que a página será carregada mais rapidamente.

Já os scripts em JavaScript devem ser movidos para o final da página, para que a carga seja otimizada. É verdade que em algumas situações não dá para mover para o final da página, mas sempre que possível devemos fazê-lo.

O IE suporta expressões em CSS e para melhor desempenho evite estas expressões.

Sempre que possível tenha o CSS e o JavaScript em arquivos externos para melhor desempenho.

Otimize para reduzir o código do CSS e do JavaScript. Remova caracteres desnecessários, linhas e trechos duplicados e optimize ao máximo seu código. Isso vale também para toda a construção do site, evitar tudo que seja desnecessário e otimizar o necessário.

Evite redirecionamentos. Não tenha mais que 3 redirecionamentos na página. Exemplo: temos 3 banners cujas imagens vem de outras URLs. No caso o site só abrirá quando pegar as imagens dos outros sites. Estando tudo normal ele terá que fazer a requisição aos outros sites e imagine quando um deles está lento.

Use GET para as requisições de AJAX. São mais rápidas que usando POST.

Otimize ao máximo a página inicial, reduzindo a carga para que o visitante chegue. Nas demais páginas poderá adicionar componentes que colocaria na inicial.

Minimizar o número de iframes.

Evite erro 404 (páginas não encontradas). Use o serviço do servidor para redirecionar o usuário para uma página personalizada quando acontecer um erro 404.

Reduza o tamanho dos cookies.

Minimize acessos DOM.

Desenvolva manipuladores de eventos inteligentes. Acontece de um evento somente ser disparado quando um outro o for e isso pode complicar. Mais detalhes no documento original.

Ao fazer link dos arquivos externos prefira < link...> ao invés de @import.

Evite filtros no CSS.

Otimize as imagens. Sempre antes de enviar as imagens para o servidor, reduza seu tamanho, como também reduza a quantidade de cores sempre que razoável. Alguns serviços online que ajudam na otimização de imagens: <http://tools.dynamicdrive.com/imageoptimizer/>
<http://www.imageoptimizer.net/Pages/Home.aspx> <http://webresizer.com/resizer/>

Otimize os sprites CSS As imagens ficando na horizontal ao invés da vertical carregam mais rapidamente.

Não altere as dimensões de imagens somente no HTML. Isso altera somente a visualização, mas seu tamanho permanece o mesmo. Altere seu tamanho offline e envie para o servidor ao invés.

Reduza o tamanho dos favicon.ico.

Mantenha o tamanho dos componentes sempre inferior a 25KB. Lembre que esse valor é para o arquivo descompactado.

Mais informações no documento original em inglês:

<http://developer.yahoo.com/performance/rules.html>