01

Dicas de UX para mobile

Transcrição

Nesta última aula do nosso curso de *Webdesign Responsivo* veremos algumas boas práticas de usabilidade para *mobile*.

Já vimos todos os códigos de *media queries* e *view ports*, o que nos falta é melhorá-los para o usuário.

Content parity

É considerada uma boa prática que o usuário, na versão *mobile* de uma página, consiga fazer tudo aquilo que ele faz no desktop. Ou seja, não é bom cortar ações, funcionalidades e conteúdo só porque a tela é menor. A necessidade não é que tudo tenha que ser igual, mas parecido e acessível para todas as plataformas.

No *mobile*, por exemplo, poderíamos quebrar um texto, que aparece em uma página no desktop, em duas páginas. Podemos formatar de maneiras diferentes um conteúdo.

Não existe "contexto mobile"!

Não é porque um usuário de *mobile* tem o perfil de acessar vários conteúdos com mais rapidez que devemos mostrar menos a ele. Não confundemos *tipo de aparelho* com *contexto de uso*. Não é possível inferirmos o modo com que o usuário utiliza uma plataforma pelo seu aparelho.

Ofereça todo o conteúdo e todos os cenários para que o usuário possa optar pelo aparelho que ele quiser.

Priorização de conteúdo

O *mobile* pode despertar a ideia de que o conteúdo deve ser priorizado. Mas a questão é que se o site consegue oferecer tudo com poucas opções, ele deve conseguir fazê-lo em todas as plataformas. No fim, o conteúdo deve ser:

- Focado
- Priorizado
- O mesmo

Mobile-first

Como já vimos, uma técnica de melhorar a experiência do usuário *mobile* é a de *Mobile-first*, com a qual priorizamos a criação de sites próprios para smartphones e tablets e só depois pensamos em sua versão para desktop.

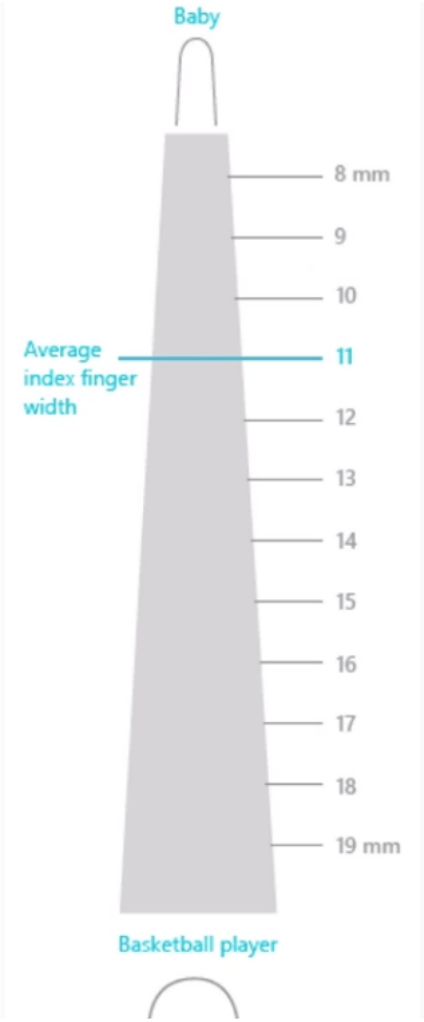
Touch-first

Paralelo ao *mobile-first*, um conceito que vem ganhando espaço é o de *Touch-first*, com o qual priorizamos o código para a interatividade de *touchscreen*. Hoje em dia não apenas os smartphones e os tablets possuem tal recurso. O *touchscreen* vem ganhando espaço entre os desktops.

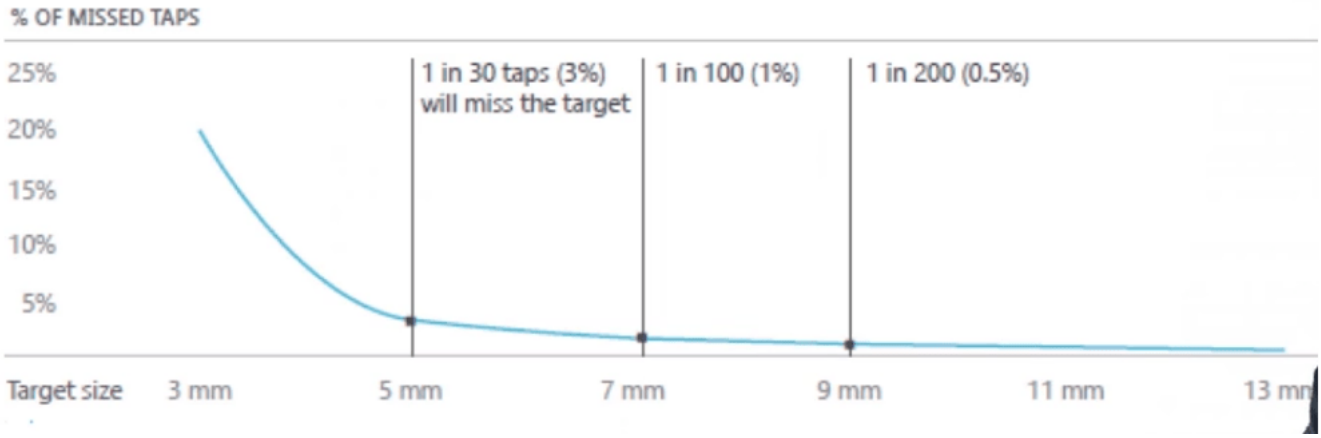
Como o *touchscreen* traz algumas dificuldades. Pensemos, desde o início da escrita do código, na possibilidade de implementá-lo com as ferramentas necessárias para o funcionamento do recurso.

Uma questão importante é a otimização da área de toque: devemos estar atentos para o tamanho e a disposição dos botões, eles devem ser grandes e espaçados o suficiente para que o dedo consiga tocar. Temos o mesmo tipo de problema com os links: um tamanho recomendado, pela Apple, é 6,8mm. Pela Mozilla, 5,9 a 9mm, e assim por diante. Perceba que é muito mais plausível apresentar esses números em milímetros, uma vez que os links têm interação com o mundo externo, não com um ponteiro de *mouse*.

Desses dados sobre o tamanho dos links, se destaca a documentação da Microsoft - que recomenda 9mm -, a qual é baseada em um estudo prático sobre a largura média do dedo dos usuários:



Os desenvolvedores da Microsoft perceberam que o ideal de tamanho é de 9mm para mais, pois com essa medida a taxa de erro é de 0,5%:



Já em relação ao espaçamento entre os botões, a Microsoft indica que o mínimo seja de 2mm.

Porém, na prática, quantos pixeis equivalem aos 9mm? Fazendo a conversão e deixando de lado algum erro, os botões devem ter, em média, 50px.

Sem *hover*

No *mobile* não existe o *hover*, efeito este que acontece quando passamos o ponteiro do *mouse*, sem clicar, por cima de um elemento. Então vamos evitá-lo no código que prioriza o *Touch*.

Exemplos do que não fazer

Veremos agora algumas más praticas de *UX* para *mobile*.

1. Veja esta tela de *mobile*. A página sugere que a adicionemos como *Home Screen*.

O problema é que ela mostra uma seta para um botão que pode não existir, neste caso um botão próprio do Safari. O design não é responsivo para outros navegadores.

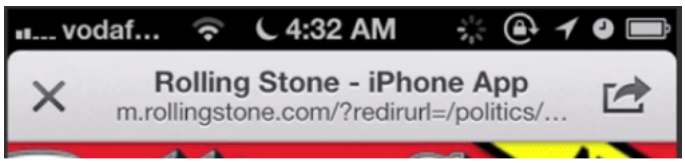
Lição: não tente detectar em qual navegador o usuário está.

1. Este é um caso de compatibilidade de modos de uso.

O site avisa que foi desenhado apenas para o modo retrato e não funciona em paisagem.

Lição: pense em todas as possibilidades de uso. O usuário pode e deve ter a vontade de navegar como ele quiser.

1. Mais um caso problemático, agora em relação ao conteúdo da home page.



Logo na primeira página do site da revista Rolling Stones aparece um *Ad* sugerindo que instalássemos o aplicativo para Iphone da revista. Isto é muito incômodo.

Lição: saiba o que o usuário quer visualizar primeiro em sua home page. Se ele quiser instalar algum aplicativo, deixe-o navegar, mas não impeça o acesso ao site.

1. Um outro caso relacionado a aplicativos:

No Quora, um site de perguntas e respostas, só é possível ler a primeira resposta. Para continuar navegando ele nos *obriga* a baixar seu aplicativo. No desktop não há tal limitação.

Lição: Novamente, não obrigue seu usuário a fazer aquilo que muito provavelmente não quer. Ele deve ser livre para navegar.

1. Agora um caso de mal design de menus. Aqui o usuário entrou numa página para ler um artigo:

Neste caso o menu é tão mal desenhado e grande que o artigo só aparecerá após rolarmos o menu muito para baixo.

Lição: preste atenção naquilo que é mais importante ser mostrado na página. Uma página que à primeira vista só mostra o menu de opções confunde o usuário.

1. Vimos anteriormente o *viewport*. Mas acrescentamos aqui o parâmetro `user-scalable=no` :

```
<meta name="viewport"
 content="width=device-width, user-scalable=no">
```

Tal parâmetro desabilita o *zoom* para o usuário. Não faz sentido impedir o usuário de mover e ampliar a tela. Além do zoom ser um gesto padrão da *Web* nos dispositivos móveis.

Impedir esse gesto também não é confiável. O próprio usuário pode, nas configurações do navegador, forçar o zoom.

Lição: Mantenha o máximo de *features* em sua página. Não há razão para diminuir a quantidade de ferramentas para o usuário se elas não influenciam no resto do site.

1. Quando trabalhamos com *mobile*, 3G, telas pequenas ou dispositivos limitados, devemos ficar atentos com a **performance**. Não podemos deixar o usuário visualizar por muito tempo a tela em branco, ou seja, muito tempo carregando a página.

O Futuro é *mobile*

Devemos pensar em *mobile* desde o início. E o design responsivo é a única forma de estarmos atualizados em relação a diferentes e novos dispositivos. Ele permite nos adaptarmos aos novos tempos.