01

Adaptações maiores do Menu

Transcrição

Nesta aula iremos estender nosso conhecimento sobre o que seria o Design Responsivo que se adapta a diferentes resoluções, agora numa escala maior. Nós já vimos que as *media queries* são uma boa maneira de fazer adaptações no design por meio de regras CSS específicas para diferentes tamanhos de tela. Vimos vários casos como, por exemplo, mudar o tamanho da fonte e o número de colunas dependendo da resolução.

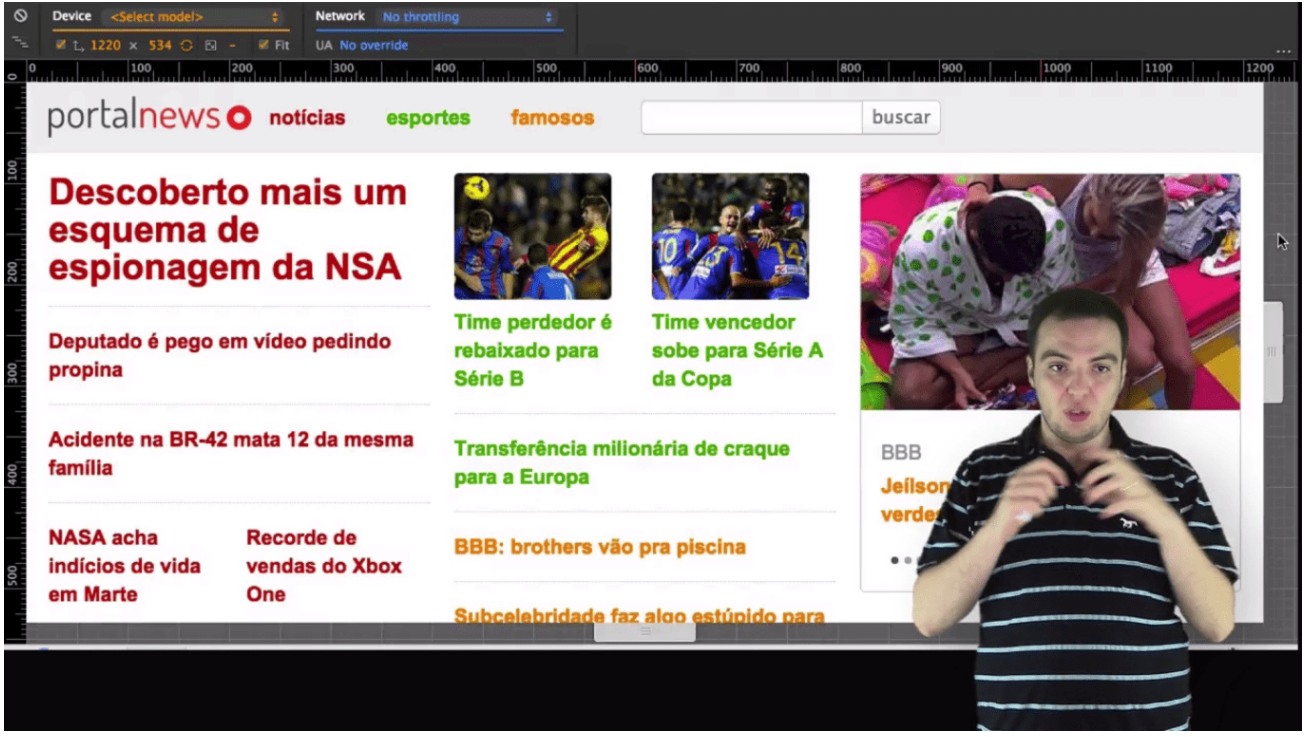
Porém, até agora, não vimos nenhum caso em que precisássemos de adaptações muito grandes como, por exemplo, quando o design mobile é muito diferente do design desktop. A maior parte delas estão em pequenos ajustes, pois não queremos reescrever partes do site o tempo inteiro. Mas em alguns cenários é importante haver mudanças maiores, tendo em vista a usabilidade.

Especificamente, vamos falar nesta aula sobre o ***Menu***. Queremos deixá-lo mais Responsivo, pensando principalmente em telas pequenas.

Observando o menu do topo do site, ele já possui alguns ajustes: o tamanho da fonte muda dependendo da resolução, na versão grande o menu fica em cima, ao lado do logo, sendo que na versão menor o menu vai para baixo do logo. Estas já são pequenas adaptações que já fizemos para as *media queries* anteriormente.

Porém, se imaginarmos um celular com tela ainda menor, precisaremos de ideias para resolver o problema do menu, que ficará muito apertado.

Vejamos uma solução, que é a de compactar o menu do topo para um menu lateral. Na versão desktop, temos algo assim:

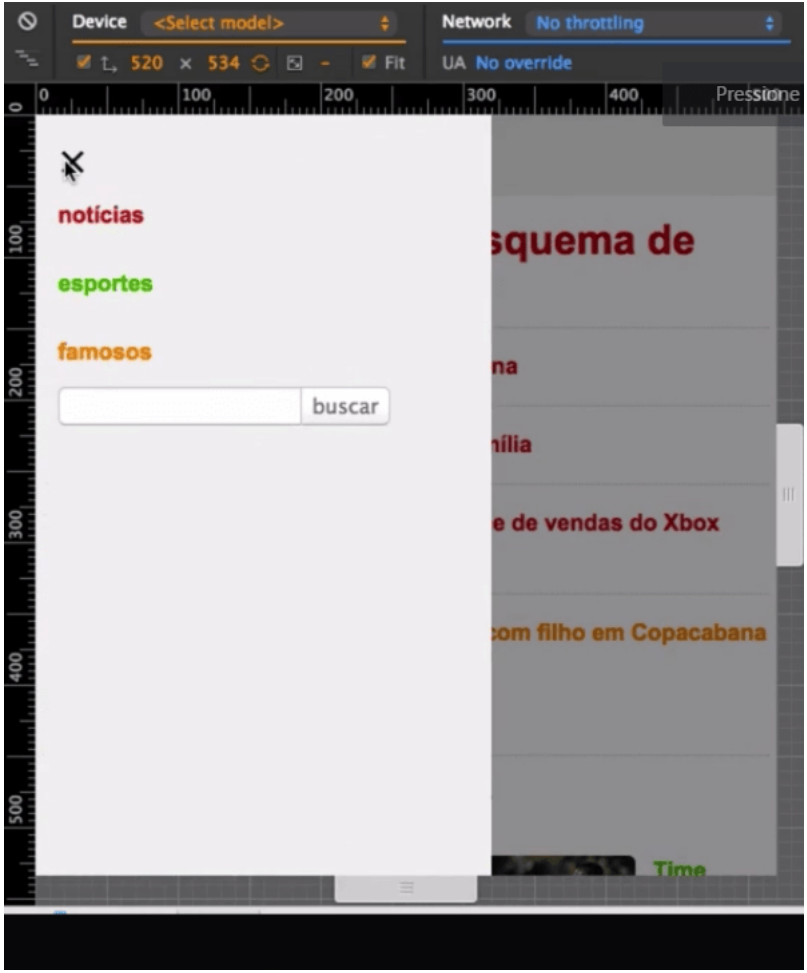


À medida que diminuimos a tela horizontalmente, o menu vai se adaptando sendo o primeiro passo seu deslocamento para baixo do logo:

Quando o *layout* for pequeno o suficiente, perceba que o menu desaparece e em seu lugar, à esquerda do logo, aparece um botão:


Device
<Select model>
Network
No throttling
594 × 534
Fit
UA No override
portalnews
Descoberto mais um esquema de espionagem da NSA
Deputado é pego em vídeo pedindo propina
Acidente na BR-42 mata 12 da mesma família
NASA acha indícios de vida em Marte
Recorde de vendas do Xbox One
Atriz da Globo passeia com filho em Copacabana
Estádios das Olimpíadas estão atrasados
Time perdedor
Time vencedor

Ao clicar nesse botão o menu aparece compactado à esquerda:


Device
<Select model>
Network
No throttling
520 × 534
Fit
UA
No override
Pressione
notícias
esportes
famosos
buscar
squema de
na
nília
e de vendas do Xbox
com filho em Copacabana
Time

Agora que vimos na prática que é possível adaptar o menu, vamos implementar essa adaptação atrelada à resolução de tela. Para isso, vamos quebrar o problema em algumas etapas:

1. A primeira etapa é o menu na versão mobile (esquecendo, por enquanto, o desenho do botão).
2. A segunda etapa é o menu aparecendo na versão mobile, logo após clicarmos no botão de abrir, mostrando o botão de fechar.

Para essas duas etapas é necessário criarmos os botões do menu:

```
<button class="menu-abrir">
    Abre Menu
</button>

<button class="menu-fechar">
    Fecha Menu
</button>
```

Nos exercícios veremos como posicioná-los. Essas duas classes irão nos ajudar a escrever o código HTML necessário para implementar a funcionalidade. Tal funcionalidade ainda não é possível de ser feita usando apenas o CSS, precisaremos do javascript para controlar o evento do clique desses botões.

Em javascript controlamos o evento de clique fazendo

```
document.querySelector('.menu-abrir').onclick = function() {
}
```

Esse evento sinalizará para o CSS abrir o menu lateral. Fazemos isso criando uma classe, a qual chamaremos de `menu-ativo` . Vamos optar por inserí-la na raiz do documento, ou seja, na *tag* HTML. Em javascript funciona dessa maneira:

```
document.querySelector('.menu-abrir').onclick = function() {
    document.documentElement.classList.add('menu-ativo');
};
```

Esse evento apenas faz clicar e adicionar essa classe. Para o evento contrário, de fechar o menu, que remove a classe:

```
document.querySelector('.menu-fechar').onclick = function() {
    document.documentElement.classList.remove('menu-ativo');
};
```

Por enquanto temos dois botões no HTML:

- Um para abrir o menu, que adiciona a classe "menu-ativo";
- Um para fechar o menu, que remove a classe "menu-ativo".

Precisamos fazer com que o CSS comporte-se diferente dependendo dessa classe no HTML.

Vamos começar pelas propriedades visuais:

```
.barra-nav {
    background: #f0f0f0;
    padding: 1em;
    margin: 0;

    height: 100%;
    width: 90%
    max-width: 320px;
}
```

Agora a posição desse menu, que deve estar por cima do conteúdo e no topo à esquerda da página:

```
.barra-nav {
    background: #f0f0f0;
    padding: 1em;
    margin: 0;

    height: 100%;
    width: 90%
    max-width: 320px;
```

```
    position: fixed;
    z-index: 1;
    top: 0;
}
```

Quando estamos selecionando a classe `.barra-nav` ainda não sabemos se o menu está ativo ou não. Vamos pensar que ele esteja inativo, ou seja, não esteja visível. Fazemos isso posicionando-o para fora da tela, à esquerda, na mesma medida que sua largura:

```
.barra-nav {
    background: #f0f0f0;
    padding: 1em;
    margin: 0;

    height: 100%;
    width: 90%
    max-width: 320px;

    position: fixed;
    z-index: 1;
    top: 0;
    left: -90%;
}
```

Por último, vamos acrescentar uma animação para quando a barra de menu se mover:

```
.barra-nav {
    background: #f0f0f0;
    padding: 1em;
    margin: 0;

    height: 100%;
    width: 90%
    max-width: 320px;

    position: fixed;
    z-index: 1;
    top: 0;
    left: -90%;

    transition: left 0.3s ease-out;
}
```

Agora vamos trabalhar a classe `.menu-ativo`, onde a classe `.barra-nav`, efetivamente ficará por cima do conteúdo:

```
.menu-ativo .barra-nav {
    left: 0;
}
```

Um detalhe importante: queremos fazer tudo isso dentro de uma *media query*:

```
@media (max-width: 600px) {
    .barra-nav {
        background: #f0f0f0;
        padding: 1em;
        margin: 0;

        height: 100%;
        width: 90%
        max-width: 320px;

        position: fixed;
        z-index: 1;
        top: 0;
        left: -90%;

        transition: left 0.3s ease-out;
    }
    .menu-ativo .barra-nav {
        left: 0;
    }
}
```

Usamos uma *media* de *max-width* porque essa é uma funcionalidade apenas do *mobile*, ou seja, ela só será executada para resoluções de tela com 600 px ou menos.

Voltando aos botões, nós os inserimos no HTML. Porém esses só fazem sentido quando estivermos na versão mobile do site. Na versão desktop é melhor escondermos esses menus. Fazemos isso com o "inverso", com a "negação" da *media query* de largura máxima de 600 px:

```
@media not all and (max-width: 600px) {
    .menu-abrir,
    .menu-fechar {
        display: none;
    }
}
```

Nosso menu interativo para a versão mobile está pronto. Mas perceba nas imagens anteriores que, quando o menu é aberto, o conteúdo por trás escurece, é aplicado a ele um *overlay*. Além disso é possível clicar fora do menu e fechá-lo, em vez de clicarmos no botão. Implementaremos essa interatividade usando pseudoelementos do CSS direto na *tag* HTML.

```
.menu-ativo:after {
    content: "";
    display: block;
```

```
        position: fixed;
        top: 0;
        left: 0;
        bottom: 0;
        right: 0;

        background: rgba(0,0,0,0.4);
    }
```

Tal implementação permite que o fundo fique opaco ao abrirmos o menu. Para que este se feche ao clicarmos no fundo criamos um evento em javascript:

```
document.documentElement.onclick = function(event) {
    if (event.target === document.documentElement) {
        document.documentElement.classList.remove('menu-ativo');
    }
};
```

Nesta aula vimos adaptações muito maiores utilizando *media queries*. Vimos como criar um menu totalmente repensado para telas de resoluções pequenas, e não apenas mudanças de tamanho de fonte ou deslocamento de elementos. Aprendemos também:

- Botões HTML;

- Classe via Javascript;
- CSS com media query;
- Fundo com pseudoelemento.