

Universidade Federal do Espírito Santo Centro de Ciências Exatas, Naturais e da Saúde Departamento de Computação



Fundamentos de Programação Web Responsividade

Fundamentos de Programação WEB Site: http://www.jeiks.net/fundpweb E-mail: jacson.silva@ufes.br

Design responsivo da Web

É quando você usa CSS e HTML para redimensionar, ocultar, reduzir, ampliar ou mover o conteúdo para que fique bem em qualquer tela.



Programação Web 1





Programação Web 1

O Curso

O desafio

anejas O curso é o início do desbravar da mente ao tentar atingir um ponto mais elevado na vida em direção do destino traçado pela vocação adquirida em berço. Cabe se conhecer e administrar de forma correta o tempo e seu futuro acontecerá.

O que é? Programação Web é desenvolver para o mercado atual. Onde? Qualquer dispositivo móvel e computadores. Como? Através da explosão da web semântica.





Design Responsivo da Web

- O design responsivo da Web:
 - Faz com que sua página da web fique "bem" em todos os dispositivos.
 - Usa apenas:
 - HTML e
 - CSS.
 - Não é um programa ou JavaScript.
- A página da Web:
 - Pode ser visualizada usando muitos dispositivos diferentes: desktops, tablets e telefones.
 - Deve ter uma boa aparência e ser fácil de usar, independentemente do dispositivo.
 - Não deve deixar de apresentar conteúdo para ajustar a telas menores, mas sim adaptar seu conteúdo para caber em qualquer dispositivo.



- A janela de visualização (*viewport*) é a área visível do usuário de uma página da web.
- No início, as páginas eram só pra telas e não eram adaptadas para celulares ou tablets.
 - Então, para corrigir isso, os navegadores desses dispositivos reduziam toda a página da Web para caber na tela.
- Não resolvia o caso, mas era uma adaptação até que a Web evoluísse e resolvesse isso.



 O HTML5 introduziu um método para permitir que os web designers assumam o controle da *viewport*, por meio da *tag* <meta>:

```
<meta name="viewport"
content="width=device-width, initial-scale=1.0">
```

- Isso fornece ao navegador instruções sobre como controlar as dimensões e o dimensionamento da página, onde:
 - width=device-width: define que a largura da página deve seguir a largura da tela do dispositivo (que varia de acordo com o dispositivo).
 - *initial-scale=1.0*: define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

Mais detalhes em:

https://developer.mozilla.org/en-US/docs/ Web/HTML/Viewport_meta_tag



• Exemplo:





Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend ontion conque nihil imperdiet domine

Com meta viewport



Sem meta viewport

- Dicas:
 - Os usuários estão acostumados a rolar sites verticalmente, mas não horizontalmente!
 - Portanto, se o usuário for forçado a rolar horizontalmente ou diminuir o zoom para ver toda a página da Web, isso resultará em uma experiência de usuário ruim.
 - não use grandes elementos de largura fixa.
 Exemplo: uma imagem exibida com uma largura maior que a janela de visualização.
 ajuste este conteúdo para caber na largura da janela de visualização.
 - não permita que o conteúdo dependa de uma largura de janela de visualização específica para renderizar bem,.
 ajuste o conteúdo para que seja bem exibido em qualquer largura de janela de visualização.
 - use @media query para aplicar estilos diferentes para telas pequenas e grandes.

Considere usar valores de largura relativos, como largura: 100%. Tenha cuidado ao usar valores de posicionamento absolutos grandes, pois pode deixar elementos fora da janela de visualização.



Iniciando a responsividade

- Detalhes:
 - Defina o box-sizing para todos os elementos:

```
{
  box-sizing: border-box;
```

```
}
```

- Se quiser, pode criar uma classe para cada coluna:
 .col-1 {width: 25%;}
 .col-2 {width: 75%;}
- E definir as propriedades de todas essas classes:
 [class*="col-"] {
 float: left;
 padding: 15px;
 border: 1px solid red;
 }



Iniciando a responsividade

- Para organizar melhor,
 - pode-se colocar essas colunas em uma div, que será a linha:

```
<div class="linha">
     <div class="col-1">...</div>
     <div class="col-2">...</div>
</div>
```

 E evitar que outros elementos sobreponham essas colunas (pois estão flutuando devido ao float):

```
.linha::after {
    content: "";
    clear: both;
    display: table;
}
```

- Exemplo: view_port-float.html



Adicionando responsividade

- Agora, devemos tratar o site para que ele responda a diferentes tamanhos de janelas.
- Isso é feito com @media query.
- Uma dica: sempre desenvolva para dispositivos móveis primeiro. Exemplo:

```
/* Para dispositivos móveis: */
[class*="col-"] {
  width: 100%;
}
@media only screen and (min-width: 768px) {
  /* Para desktops: */
  .col-1 {width: 25%;}
  .col-2 {width: 75%;}
}
```

Faça isso no exemplo anterior: view_port-float.html



Atividade

- Baixe o exemplo: *media_query-ex.html*
- Torne-o responsivo da seguinte forma:
 - Para dispositivos móveis (menos de 600px): largura das colunas com 100%
 - Para tablets (mais que 600px):
 largura da coluna 1: 25%
 largura da coluna 2: 75%
 largura da coluna 3: 100%
 - Para desktops (mais que 768px): largura da coluna 1: 25% largura da coluna 2: 50% largura da coluna 3: 25%



Quais tamanhos considerar?

- Há milhares de dispositivos com diferentes larguras e alturas de telas.
- É praticamente impossível se adaptar a todos.
- Por isso, foque nestes em 5 grupos:

```
/* Bem pequenos (celulares, 600px e abaixo) */
@media only screen and (max-width: 600px) {...}
```

```
/* Pequenos (tablets e celulares grandes, 600px e acima) */
@media only screen and (min-width: 600px) {...}
```

```
/* Médios(tablets maiores, 768px e acima) */
@media only screen and (min-width: 768px) {...}
```

```
/* Grandes (laptops/desktops, 992px e acima) */
@media only screen and (min-width: 992px) {...}
```

/* Bem grandes (laptops/desktops grandes, 1200px e acima) */
@media only screen and (min-width: 1200px) {...}



Tratando imagens

• Defina só a largura da imagem e deixe que sua altura seja definida automaticamente:

```
img {
    width: 100%;
    height: auto;
}
```

- Assim, a imagem será responsiva e terá um tamanho ajustado ao seu container.
- Porém, para que ela não seja maior que o tamanho original, é melhor fazer:

```
img {
    max-width: 100%;
    height: auto;
}
```



Atividade

- Adicione uma imagem na coluna 2 do último exercício.
- Obtenha o a largura e a altura da imagem e defina essas propriedades na própria TAG
- Crie o CSS para que ela seja sempre 100% da largura de seu *container* (ajustando automaticamente sua altura).



Tratando imagens

- Às vezes, pode ser melhor mudar a imagem dependendo do tamanho do dispositivo.
- Para isso, utilize o elemento picture:

```
<picture>
<source srcset="img_pequena.jpg"
    media="(max-width: 400px)">
<source srcset="img.jpg">
<img src="img.jpg" style="width; auto;">
</picture>
```

• A primeira source que se encaixe nas preferências é a que é usada.



Frameworks

- Permitem ajustes automáticos em seu site.
- Você deve utilizar as classes e métodos já fornecidos por eles para estilizar seu site.
 Assim, seu site já se adaptará as mudanças de tela.
- Frameworks mais utilizados:
 - Bootstrap: https://getbootstrap.com (usa JavaScript)
 - W3.CSS:

https://www.w3schools.com/w3css/default.asp (é chamado de puro, isto é, sem JavaScript)

