

Artigo

Invista em você! Saiba como a DevMedia pode ajudar sua carreira.



# Trabalhando com Design Responsivo em HTML5

Neste artigo será feita uma abordagem acerca dos conceitos que norteiam o universo de desenvolvimento para ambientes responsivos, que exigem o redimensionamento de suas telas dado os diferentes dispositivos onde elas podem ser executadas.

**Por que eu devo ler este artigo:** Esse artigo tem o objetivo de dar ao leitor uma compreensão geral e completa sobre as técnicas e conceitos mais

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

Artigos



HTML



## Trabalhando com Design Responsivo em HTML5

Em um passado recente, os sites eram desenvolvidos visando atender a um público oriundo, em sua grande maioria, de computadores. Os computadores eram a grande maioria.

A variação existente entre os modelos de monitores utilizados era muito pouca, o que tornava possível a definição de uma estratégia que visava definir uma largura máxima que não comprometesse a experiência dos usuários.

Essa estratégia foi bastante satisfatória até pouco tempo atrás, mas atualmente ela se tornou obsoleta devido ao fato de não conseguirmos mais definir uma largura máxima, em função da existência de uma grande variedade de dispositivos utilizados para navegar na internet.

São celulares, tablets, netbooks, notebooks, computadores, videogames, televisores, etc., o que torna a tarefa de proporcionar uma boa experiência para todos esses tipos de dispositivos um grande desafio.

Como podemos desenvolver um site que atenda a cada um desses milhares de dispositivos? Uma resposta pode ser a criação de versões específicas e independentes voltadas a atender cada tipo utilizado separadamente.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



um artigo com o nome de “Responsive Web Design”. Neste artigo, entre outras coisas, Ethan diz que devemos aceitar que diferentemente do design impresso (restrito a limitação física de uma página ou superfície), no design para internet essas restrições físicas não existem.

Ademais, também não devemos nos sustentar em soluções que restrinjam a internet, pelo contrário, devemos projetar soluções flexíveis e adaptativas. Esse artigo deu origem a uma nova forma de se enxergar o design e o desenvolvimento de sites, resultando no conceito hoje conhecido como: **design responsivo**.

O design responsivo visa solucionar esse problema apresentando um layout que seja adaptável aos mais diversos tipos de dispositivos existentes hoje no mercado. Essa adaptação acontece através da detecção de algumas características presentes nos mesmos que possibilitam a criação de regras específicas para cada um, possibilitando, assim, a manipulação do layout e oferecendo uma navegação mais simples, intuitiva e contável aos seus usuários de acordo com o dispositivo utilizado.

*Tornar um site que não é responsivo em um site responsivo não significa somente diminuir ou aumentar o tamanho dos elementos para que se encaixem nos mais variados tamanhos de tela, pois isso comprometeria toda a experiência do usuário.*

Dentre as características essenciais a um site responsivo, destacam-se:

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



- Simplificar elementos da tela para dispositivos móveis, onde o usuário normalmente tem menos tempo e menos atenção durante a navegação;
- Redimensionar as imagens e vídeos para que não sobrecarreguem a transferência de dados e também para que se adaptem ao dispositivo garantindo que os mesmos se apresentem de forma nítida, sem cortes e que não façam uso da barra de rolagem para serem visualizados;
- Ocultar ou remover elementos desnecessários nos dispositivos menores;
- Adaptar o tamanho de botões, links e menus para interfaces *touch* onde o ponteiro do mouse é substituído pelo dedo do usuário.

## Mobile First

O *Mobile First* é uma estratégia de desenvolvimento que diz que todo o planejamento de um projeto web deve iniciar pelos dispositivos móveis e somente depois gradualmente para os outros dispositivos, até chegar nos notebooks e desktops.

Em um passado recente, a prática de planejar e desenvolver projetos web voltados para dispositivos como computadores era bastante comum, até por questões históricas, para atender a demanda dos computadores, e onde se concentrava a web até então.

Começar pelos dispositivos móveis permite que a atividade de priorização do conteúdo seja feita de forma mais intuitiva, uma vez que em um dispositivo móvel somente as principais informações e funcionalidades de um projeto web serão disponibilizadas por conta do pouco espaço disponível e também pelas diferentes

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



dispositivos móveis, estas versões irão conter somente as informações principais, pois não há espaço disponível o suficiente para exibir tudo que teríamos na versão desktop.

Outrossim, se isso for um pré-requisito por parte do cliente, então uma estratégia de design, estruturação e usabilidade deverá ser iniciada visando reduzir as dificuldades por trás da mesma.

Posteriormente serão produzidas versões mais complexas voltadas para outros dispositivos, estas versões poderão conter mais informações de acordo com o espaço disponível proveniente de cada um.

Podemos dizer que para os dispositivos móveis é essencial que o layout seja objetivo e focado totalmente no conteúdo.

*Segundo a 9ª pesquisa TIC Domicílios divulgada pelo CETIC.br (Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação), somente em 2013 o número de brasileiros que acessaram a internet por meio de celular ou tablet atingiu 52,5 milhões, o que corresponde a 31% da população do país.*

*O percentual mais que dobrou nos últimos 2 anos, quando em 2011 esse número era apenas de 15%.*

## A Necessidade de Adaptatividade

Como a web em landscape (modo paisagem de visualização) torna-se cada vez

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

Desde grades fluídicas, imagens flexíveis até consultas de mídia para obter bons layouts, independentemente do tamanho de dimensões da tela do dispositivo.

No entanto, o contexto móvel é muito mais do que apenas o tamanho da tela. Os nossos dispositivos móveis estão conosco onde quer que vamos, desbloqueando novos casos de uso todos os dias. Porque temos constantemente nossos dispositivos móveis com a gente, a conectividade pode ser todo o tabuleiro, variando de sinais wi-fi fortes no sofá para 3G ou EDGE quando fora de casa.

Além disso, telas de toque abrem novas oportunidades de interagir diretamente com o conteúdo e a ergonomia móvel leva a diferentes considerações ao projetar layout e funcionalidade.

A fim de criar um site que é realmente concebido para o contexto móvel e não apenas para pequenas telas, queremos garantir que vamos enfrentar os muitos desafios do desenvolvimento móvel upfront. As restrições do contexto móvel nos forçam a focar em qual conteúdo é essencial e como apresentar esse conteúdo o mais rápido possível.

## Progressive Enhancement x Graceful Degradation

A internet é um ambiente extremamente dinâmico, devido às constantes evoluções das tecnologias que a envolvem. Um navegador que hoje é o mais moderno do mercado pode se tornar obsoleto amanhã com a chegada de uma nova tecnologia, caso essa não seja devidamente suportada.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



Como fazer para utilizar o que há de mais moderno em termos de desenvolvimento web sem consequentemente perder os usuários de navegadores mais antigos que não suportam certas tecnologias?

Para resolver esse dilema, podemos utilizar os conceitos do **Progressive Enhancement** (Melhoria Progressiva) e do **Graceful Degradation** (Degradação Graciosa). Eles constituem dois conceitos diferentes, mas que possuem em comum um mesmo objetivo: suportar o maior número possível de usuários oferecendo a mesma experiência ou funcionalidade independentemente do dispositivo, conexão ou navegador utilizado.

O Graceful Degradation segue a linha de raciocínio na qual devemos utilizar no desenvolvimento o que há de melhor e mais moderno disponível em termos de tecnologia, visando primeiramente os navegadores mais recentes que suportam as mesmas e para os navegadores mais antigos que não as suportam, realizando adaptações e viabilizando a navegação de modo que a experiência do usuário não seja comprometida.

Já o Progressive Enhancement segue uma linha de raciocínio oposta, na qual devemos utilizar para o desenvolvimento recursos que possam ser suportados por todos os tipos de navegadores, inclusive os mais antigos, realizando melhorias progressivamente para os navegadores mais novos que suportem tecnologias mais modernas.

Podemos dizer que entre os dois conceitos não existe um melhor a ser escolhido quanto a forma de oferecer suporte aos navegadores, pois ambos têm o mesmo

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



completa, que utiliza um maior número de recursos tecnológicos para um computador ou notebook, que possui maior poder de processamento e melhor conexão.

## Meta Tag Viewport

O termo **viewport** corresponde à área disponível para exibição de conteúdo que cada dispositivo possui.

O valor do viewport varia de acordo com o dispositivo. Por exemplo, em um computador, o viewport corresponde a área (altura e largura) em pixels ocupada pelo navegador. Portanto, quando o navegador é redimensionado para uma tela menor, automaticamente o seu viewport também se torna menor.

A mesma lógica acontece quando o navegador é redimensionado para uma dimensão maior. Já em dispositivos móveis, não existe a possibilidade de redimensionar o navegador, pois as aplicações ocupam 100% da área disponível, ou seja, 100% do viewport.

Os sites não projetados e desenvolvidos com foco em dispositivos móveis, quando são acessados a partir dos mesmos, tem o zoom ajustado automaticamente fazendo com eles sejam encaixados de acordo com a largura do dispositivo. Isso ocorre porque nos dispositivos móveis a largura do viewport foi padronizada para 980 pixels na maioria das fabricantes.

Esse valor relacionado ao viewport foi adicionado pelo fato de que quando a navegação ocorre a partir de um dispositivo móvel (de viewport reduzido) seria

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar





Consequentemente, essa característica causa diversos problemas em sites já projetados para o mundo mobile. Isso porque o dispositivo irá encolher o site para exibi-lo de acordo com a largura de 980 pixels, o que impossibilita o funcionamento das regras responsivas, cujo problema a Meta Tag Viewport visa solucionar a partir de agora.

A meta tag viewport possui um formato semelhante ao de outras tags meta, conforme podemos observar a seguir:

```
1 | <meta name="viewport" content="">
```

Na declaração do content é possível especificar uma diversidade de parâmetros, tais como:

- **width:** define a largura do viewport.
- **height:** define a altura do viewport.
- **initial-scale:** define a escala inicial (zoom) do viewport.

Para que possamos assegurar o funcionamento das regras responsivas em todos os dispositivos, devemos declarar que a largura do viewport será igual à mesma largura do dispositivo e que a escala inicial (zoom) é 1, conforme demonstrado a seguir:



O atributo `Media Type` é o responsável por designar quais regras CSS serão interpretadas por um determinado tipo de dispositivo, oferecendo a melhor apresentação possível do conteúdo sem a necessidade de alteração do mesmo.

Através desse atributo é possível apresentar o site de maneira diferente aplicando o estilo mais apropriado de acordo com o tipo do dispositivo utilizado. Por exemplo, um determinado conteúdo é apresentado de uma forma para um computador e de uma outra forma completamente diferente para uma impressora, e assim sucessivamente.

Dentre os diversos tipos de `Media Type`, alguns se destacam e são importantes ao conhecimento do leitor:

- **All:** Se refere a todos os tipos de dispositivos.
- **Braille:** Se refere aos dispositivos táteis.
- **Embossed:** Se refere aos dispositivos que imprimem em braille.
- **Handheld:** Se refere aos dispositivos de mão, normalmente com telas pequenas de baixa resolução e largura de banda limitada.
- **Print:** Se refere aos dispositivos de impressão.
- **Projection:** Se refere a apresentações do tipo slides.
- **Screen:** Se refere a monitores ou dispositivos com telas coloridas e resolução adequada.
- **Speech:** Se refere a sintetizadores de voz ou leitores de tela.
- **TTY:** Se refere a terminais, teletypes e dispositivos portáteis com display limitado.
- **TV:** Se refere a televisores ou dispositivos com baixa resolução, quantidade

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

exemplos para cada uma.

**Tabela 1.** Lista de situações e exemplos de inclusão dos atributos media type.

Descrição	Exemplo
Utilizando o atributo media na chamada de um arquivo CSS em um arquivo HTML.	<pre>&lt;link rel="stylesheet" href="arquivo.css" type="text/css"</pre>
Utilizando o atributo media na chamada de um arquivo CSS em um arquivo XML.	<pre>&lt;? xml-stylesheet rel="stylesheet" href="arquivo.css" me</pre>
Utilizando a tag @import	<pre>1 &lt;style type="text/css" media="screen"&gt; 2 @import "arquivo.css"; 3 /&gt;</pre>

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



Utilizando a tag @import em um arquivo CSS.	<pre>@import url("estilo.css") print;</pre>
Utilizando a tag @media em um arquivo CSS.	<pre>1  @media screen { 2 3      body { 4 5          background-color: #F00; 6 7      } 8 9  }</pre>

Fazendo um rápido comparativo de usabilidade em relação a estes termos, é possível observar que com o passar dos anos o uso de Media Types se tornou obsoleto devido à modernização e à grande quantidade de dispositivos com características totalmente diferentes pertencentes a um mesmo tipo. Com isso, somente os tipos **screen** e **print** ainda são utilizados em grande escala.

Por exemplo, os smartphones não pertencem ao tipo **handheld**. Embora sejam considerados dispositivos de mão, eles possuem monitores com telas coloridas e de alta resolução e por isso serão interpretados como sendo do tipo **screen**.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



funcionam como uma extensão do atributo `Media Type`, adicionando e combinando características com o uso de operadores lógicos, permitindo que sejam criadas regras mais específicas e eficientes e garantindo um controle maior para a apresentação do conteúdo em diferentes tipos de screens.

Essas expressões possuem, como resultado, um valor booleano, ou seja, são necessariamente verdadeiras ou falsas. As regras associadas a essas expressões somente serão aplicadas quando o resultado for verdadeiro.

As Media Queries adicionam características detectáveis, que são denominadas como Media Features. Cada uma delas possui funções e valores específicos, tais como os definidos na **Tabela 2**.

**Tabela 2.** Lista de atributos aceitos pelas media queries.

Atributo	Descrição
<b>aspect-ratio</b>	Se refere à proporção entre os valores de largura e altura. Os valores são compostos pela divisão da largura/valor da altura. Não aceita os prefixos min/max.
<b>device-aspect-ratio</b>	Se refere à proporção entre os valores de largura e altura <b>do dispositivo</b> . Os valores são compostos por valor da largura/valor da altura. Não aceita os prefixos min/max.
<b>Color</b>	Se refere ao número de bits por cor. Os valores são numéricos. Aceita os prefixos min/max.
<b>color-index</b>	Se refere ao número de entradas na tabela de pesquisa de cores do dispositivo de saída. Os valores são numéricos. Aceita os prefixos min/max.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



<b>Width</b>	Se refere a largura da janela do navegador. Os valores são medidas de comprimento. Aceita os prefixos min/max.
<b>device-width</b>	Se refere à largura da mídia. Os valores são medidas de comprimento. Aceita os prefixos min/max.
<b>Grid</b>	Se refere ao tipo de dispositivo, se é orientado a bitmaps ou grids. Os valores são <b>1</b> para dispositivos orientados a grid e <b>0</b> para dispositivos orientados a bitmap. Não aceita os prefixos min/max.
<b>Monochrome</b>	Se refere ao número de bits por pixel em um buffer de quadros monocromáticos. Os valores são numéricos. Aceita os prefixos min/max.
<b>Orientation</b>	Se refere à orientação da mídia. Os valores são <b>portrait</b> para vertical / retrato e <b>landscape</b> para horizontal / paisagem. Não aceita os prefixos min/max.
<b>Resolution</b>	Se refere à resolução, densidade por pixel, utilizada pelo dispositivo. Os valores são em <b>DPI</b> ou <b>DCM</b> . Aceita os prefixos min/max.
<b>Scan</b>	Se refere ao tipo de formação de imagens para televisores. Os valores são <b>progressive</b> ou <b>interlace</b> . Não aceita os prefixos min/max.

Para realizar a combinação entre os Media Types e as Media Features que formam as regras condicionais e resultam nas Media Queries, são utilizados os **operadores**. Dentre eles, podemos destacar:

- **not:** É utilizado quando se deseja que o resultado de uma determinada expressão seja o oposto ao real. Veja na **Listagem 1** um exemplo do seu uso.
- **only:** É utilizado quando se deseja prevenir que navegadores antigos que não suportam Media Features tentem processar a expressão. Na **Listagem 2** podemos ver um exemplo fiel disso.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



- “,: É utilizado para juntar duas ou mais expressões diferentes que deverão executar um mesmo conjunto de regras. Funciona como um “ou” condicional na lógica de programação.

**Listagem 1.** Exemplo de utilização do operador lógico “not”.

```
1 | @media not print and (min-width: 768px) {  
2 |     body {  
3 |         background-color: #FF0000;  
4 |     }  
5 | }
```

Neste exemplo, todos os dispositivos que não forem do tipo `print` com largura mínima de `768 pixels` terão a cor de fundo do corpo do documento vermelha. Sem o uso do operador `not`, o resultado seria o oposto, e a cor de fundo do corpo do documento seria vermelha para todos os dispositivos do tipo `print` com a largura mínima de `768 pixels`.

**Listagem 2.** Exemplo de utilização do operador lógico “only”.

```
1 | @media only screen and (max-width: 320px) {  
2 |     h1 {  
3 |         text-decoration: underline;  
4 |     }  
5 | }
```

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



operador `only`, a expressão iria ser executada por todos os dispositivos, e no caso de dispositivos que não suportam Media Features, somente o `Media Type` seria reconhecido e as mesmas seriam ignoradas.

O que resultaria na execução da regra referente ao `h1` em dispositivos nos quais esta não deveria ser executada.

Na **Listagem 3** temos uma expressão múltipla relacionada aos dispositivos do tipo `screen`, onde duas Media Features são utilizadas dentro de uma mesma expressão. Uma Media Feature corresponde à largura mínima de `320 pixels` e outra à largura máxima de `480 pixels`. Portanto, caso positiva, o elemento `header` terá a largura de `100%` em dispositivos do tipo `screen` com a largura entre `320` e `480 pixels`.

**Listagem 3.** Exemplo de utilização do operador lógico “and”.

```
1 | @media only screen and (min-width: 320px) and (max-width: 480px) {  
2 |     header {  
3 |         width: 100%;  
4 |     }  
5 | }
```

Na **Listagem 4** há duas expressões diferentes, uma para dispositivos do tipo `handheld` com largura máxima de `500 pixels` e outra para dispositivos do tipo `screen` com largura máxima de `620 pixels`.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



#### Listagem 4. Exemplo de utilização do operador lógico “,”.

```
1 | @media only handheld and (max-width: 500px), only screen and (max-wi
2 |     p {
3 |         font-weight: bold;
4 |     }
5 | }
```

## Breakpoints

Os **breakpoints** são delimitadores das regras CSS para atenderem diferentes especificações, que são criados pelo uso de Media Queries. Tratam-se também de uma forma de organizar o uso das Media Queries, visando a criação de blocos previamente definidos de acordo com os dispositivos que serão suportados, e onde serão inseridas as respectivas regras CSS.

Isso evitará, desta forma, a utilização de diversas regras pontuais para a resolução de casos específicos.

Não existe um número ou um grupo exato de breakpoints a serem utilizados. A definição dos breakpoints está diretamente ligada a características específicas do site em questão, como o conteúdo e os dispositivos que serão suportados.

Portanto, um novo breakpoint deve ser criado sempre que a apresentação do conteúdo em determinado dispositivo não oferecer a melhor experiência possível

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



DEVIA

```
1  /* INICIO REGRAS PARA TODOS OS DISPOSITIVOS */
2  INSIRA REGRAS CSS AQUI!
3  /* FIM REGRAS PARA TODOS OS DISPOSITIVOS*/
4
5  /* INICIO REGRAS PARA DISPOSITIVOS DE IMPRESSÃO */
6  @media print {
7      INSIRA REGRAS CSS AQUI!
8  }
9  /* FIM REGRAS PARA DISPOSITIVOS DE IMPRESSÃO */
10
11 /* INICIO REGRAS PARA DISPOSITIVOS COM LARGURA MÍNIMA DE 320 PIXELS.
12 @media screen and (min-width: 320px) {
13     INSIRA REGRAS CSS AQUI!
14 }
15 /* FIM REGRAS PARA DISPOSITIVOS COM LARGURA MÍNIMA DE 320 PIXELS. PC
16
17 /* INICIO REGRAS PARA DISPOSITIVOS COM LARGURA MÍNIMA DE 768 PIXELS.
18 @media screen and (min-width: 768px) {
19     INSIRA REGRAS CSS AQUI!
20 }
21 /* FIM REGRAS PARA DISPOSITIVOS COM LARGURA MÍNIMA DE 768 PIXELS. PC
22
23 /* INICIO REGRAS PARA DISPOSITIVOS COM LARGURA MÍNIMA DE 1024 PIXELS
24 @media screen and (min-width: 1024px) {
25     INSIRA REGRAS CSS AQUI!
26 }
27 /* FIM REGRAS PARA DISPOSITIVOS COM LARGURA MÍNIMA DE 1024 PIXELS. F
```

Veja que a implementação desse tipo de estrutura é muito semelhante a estruturas já existentes em outras ferramentas e linguagens de programação, como o Visual Studio, por exemplo.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



Agora que já conhecemos em detalhes o que são os termos `Media Types`, `Media Features` e `Media Queries`, é hora de colocar em prática o conhecimento adquirido. Neste exemplo, vamos criar um site responsivo que deverá apresentar quatro versões de layout diferentes. A saber:

- **Primeira versão:** Se refere aos dispositivos considerados pequenos, com largura até 480 pixels;
- **Segunda versão:** Se refere aos dispositivos considerados médios, com largura maior que 480 pixels e até 1024 pixels;
- **Terceira versão:** Se refere aos dispositivos considerados grandes, com largura maior que 1024 pixels;
- **Quarta versão:** Se refere aos dispositivos de impressão.

Todas as quatro versões possuirão características específicas. Primeiramente, crie um documento HTML e nomeie-o como “`index.html`”, e adicione o conteúdo presente na **Listagem 6**, que se refere a estrutura inicial da página, contendo apenas algumas declarações das tags que vimos anteriormente, bem como um título e uma tag `h1` no corpo da página.

#### Listagem 6. Marcação HTML inicial.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4
5       <!-- INICIO META TAGS -->
6       <meta charset="utf-8">
```

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



```
15 </head>
16 <body>
17
18 <!-- INICIO TITULO DA PÁGINA -->
19 <h1 class="titulo">Teste Layout Responsivo</h1>
20 <!-- FIM TITULO DA PÁGINA -->
21
22 </body>
</html>
```

Repare que estamos usando as configurações referentes à nova especificação da HTML5 para melhor se adequar a realidade do artigo.

Após a criação do arquivo, já com a estrutura inicial, vamos incluir o conteúdo específico referente à cada versão do layout.

Na **Listagem 7** está presente o conteúdo específico referente à primeira versão, que se refere aos dispositivos considerados pequenos, com largura até 480 pixels. Faça a inclusão do conteúdo dentro da tag `<body>` logo após o título da página representando pela tag `h1`.

#### Listagem 7. Conteúdo específico da primeira versão.

```
1 <!-- INICIO CONTEUDO PRIMEIRA VERSAO -->
2
3 <section class="conteudo-dispositivo pequeno">
4   <h2 class="tipo">Dispositivo Pequeno</h2>
5   <p class="descricao">
```

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



No conteúdo presente na **Listagem 8** temos o código referente à segunda versão, que se refere aos dispositivos considerados médios, com largura maior que 480 pixels e até 1024 pixels. Insira este conteúdo no arquivo `index.html` logo após o conteúdo da primeira versão da listagem anterior.

#### Listagem 8. Conteúdo específico da segunda versão.

```
1 <!-- INICIO CONTEUDO SEGUNDA VERSAO -->
2 <section class="conteudo-dispositivo medio">
3   <h2 class="tipo">Dispositivo Médio</h2>
4   <p class="descricao">
5     Esse texto será exibido somente em dispositivos com largura ma
6   </p>
7 </section>
8 <!-- FIM CONTEUDO SEGUNDA VERSAO -->
```

Em seguida, temos a **Listagem 9** que apresenta o conteúdo específico referente à terceira versão, que se dirige aos dispositivos considerados grandes, com largura maior que 1024 pixels.

Insira este conteúdo no arquivo `index.html` logo após o conteúdo referente à segunda versão. Note que os contextos se assemelham em relação ao texto e tags criados nas mesmas seções ( `section` ). Logo, finalizaremos com a explicação em

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



```
1 <!-- INICIO CONTEUDO TERCEIRA VERSAO -->
2 <section class="conteudo-dispositivo grande">
3     <h2 class="tipo">Dispositivo Grande</h2>
4     <p class="descricao">
5         Esse texto será exibido somente em dispositivos com largu
6     </p>
7 </section>
8 <!-- FIM CONTEUDO TERCEIRA VERSAO -->
```

Finalmente, no conteúdo presente na **Listagem 10** temos o código referente à quarta versão citada, que se refere aos dispositivos de impressão. Insira este conteúdo no arquivo index.html logo após o conteúdo referente à terceira versão.

**Listagem 10.** Conteúdo específico da versão para dispositivos de impressão.

```
1 <!-- INICIO CONTEUDO QUARTA VERSAO -->
2 <section class="conteudo-dispositivo impressao">
3     <h2 class="tipo">Dispositivo de Impressão</h2>
4     <p class="descricao">
5         Esse texto será exibido somente em dispositivos de impres
6     </p>
7 </section>
8 <!-- FIM CONTEUDO QUARTA VERSAO -->
```

A estrutura dos conteúdos específicos apresentados respectivamente nas listagens anteriores é a mesma, o que muda é apenas o conteúdo de cada uma

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



Agora com o arquivo `index.html` devidamente preenchido, vamos criar o arquivo CSS a ser utilizado. Denomine o arquivo como “`stylesheet.css`” e, em seguida, adicione o trecho de código CSS presente na **Listagem 11**.

Este trecho possui regras comuns que serão aplicadas a todos os dispositivos independentemente da versão apresentada, tais como definições gerais de espaçamento, margens, alinhamento de texto e exibição dos componentes na página.

Observe também que por se tratar de regras CSS, as mesmas serão aplicadas em cascata, obedecendo ao modelo de herança do CSS3.

**Listagem 11.** Regras CSS comuns a todos os dispositivos.

```
1  /* INICIO REGRAS CSS COMUNS À TODOS OS DISPOSITIVOS */
2  .titulo {
3      margin: 20px 0 80px;
4      text-align: center;
5  }
6
7  .conteudo-dispositivo {
8      display: none;
9      text-align: center;
10 }
11 /* FIM REGRAS CSS COMUNS À TODOS OS DISPOSITIVOS */
```

Na **Listagem 12** temos as Media Queries responsáveis pela detecção e aplicação

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



componentes, para facilitar o entendimento de uma forma geral.

**Listagem 12.** Regras CSS referentes à primeira, segunda, terceira e quarta versões.

```
1  /* INICIO REGRAS CSS REFERENTES À PRIMEIRA VERSÃO */
2  @media only screen and (max-width: 480px) {
3
4      body {
5          background-color: #D46A6A;
6          color: #FFF;
7      }
8
9      .conteudo-dispositivo.pequeno {
10         display: block;
11     }
12
13 }
14 /* INICIO REGRAS CSS REFERENTES À SEGUNDA VERSÃO */
15 @media only screen and (min-width: 481px) and (max-width: 1024px) {
16
17     body {
18         background-color: #758AA8;
19         color: #FFF;
20     }
21
22     .conteudo-dispositivo.medio {
23         display: block;
24     }
25
26 }
27 /* INICIO REGRAS CSS REFERENTES À TERCEIRA VERSÃO */
28 @media only screen and (min-width: 1025px) {
29
30
```

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar





```
38     }
39
40 }
41 /* INICIO REGRAS CSS REFERENTES À QUARTA VERSÃO */
42 @media print {
43
44     .titulo {
45         margin: 0 0 50px;
46         text-align: left;
47     }
48
49     .conteudo-dispositivo {
50         text-align: left;
51     }
52
53     body {
54         background-color: #FFF;
55         color: #000;
56     }
57
58     .conteudo-dispositivo.impressao {
59         display: block;
60     }
61 }
```

Podemos notar que as regras referentes à cada versão são bastante semelhantes. Todas são compostas por uma Media Query e suas respectivas regras que correspondem à forma como o conteúdo será apresentado em uma das versões definidas.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

pela detecção e aplicação das regras CSS aos dispositivos de impressão, essa versão possui também algumas regras com a função de anular as regras CSS comuns a todos os dispositivos, apresentadas anteriormente.

Isso se deve ao fato de que por se tratar de um dispositivo de impressão, as regras referentes à distribuição e cor dos elementos na página deixaram de fazer sentido.

Agora o arquivo `stylesheet.css` está completo e precisamos incluir no arquivo `index.html` a chamada para o mesmo, possibilitando assim que possamos visualizar na prática o funcionamento das Media Queries.

Verifique na **Listagem 13** o conteúdo para efetuar a chamada ao arquivo



```
<title> .
```

### Listagem 13. Chamada para o arquivo `stylesheet.css`.

```
1 | <!-- INICIO ARQUIVOS CSS -->
2 | <link type="text/css" rel="stylesheet" href="stylesheet.css">
3 | <!-- FIM ARQUIVOS CSS -->
```

Agora podemos visualizar o resultado da execução do arquivo HTML. Na **Figura 1** apresentamos o resultado que será exibido em dispositivos com largura até 480 pixels, referenciados pela primeira versão.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

**Figura 1.** Resultado apresentado pela primeira versão.

Na **Figura 2** apresentamos o resultado que será exibido em dispositivos com largura maior que 480 pixels e até 1024 pixels , referenciados pela segunda versão.

**Figura 2.** Resultado apresentado pela segunda versão.

Na **Figura 3** apresentamos o resultado que será exibido em dispositivos com largura maior que 1024 pixels , referenciados pela terceira versão.

**Figura 3.** Resultado apresentado pela terceira versão.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

**Figura 4.** Resultado apresentado pela quarta versão.

Através desse exemplo podemos verificar que ao fazer uso das Media Queries é possível apresentar um único site com conteúdo totalmente diferente e de diversas formas, de acordo com o dispositivo utilizado, explorando devidamente suas características, a fim de melhorar a experiência do usuário sem a necessidade de alterar o conteúdo, e de forma extremamente simples.

Após a leitura deste artigo, o leitor terá adquirido conhecimento sobre técnicas e conceitos importantes e extremamente necessários para o planejamento correto do desenvolvimento de sites responsivos. Além dos exemplos práticos que conseguem maximizar o impacto no aprendizado como um todo.

#### Links

- [Site com estatísticas sobre o uso de navegadores](#)
- [Site com diversos exemplos de sites responsivos](#)
- [Dicas para projetos de sites responsivos](#)

#### Tecnologias:

[CSS](#)[HTML](#)

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar



# HTML5

Como usar Atributos Globais do HTML5



# ARTIGOS

HTML Meta Tags: Entendendo o uso de...

## PARA QUEM QUER SER PROGRAMADOR DE VERDADE. **VAGAS LIMITADAS**

12x **R\$ 58,80**

Em caso de dúvidas chame no whatsapp 

### PLANO PRO

Formação FullStack completa  
Projetos reais  
Professores online  
Exercícios gamificados  
Certificado de autoridade

**MATRICULE-SE**

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

Aceitar

**DEVMEDIA**

EM 2015

[Tecnologias](#)[Exercicios](#)[Artigos](#)[Quem Somos](#)[Fale conosco](#)[Plano para Instituição de ensino](#)[Assinatura para empresas](#)[Assine agora](#)

Hospedagem web por Porta 80 Web Hosting.

Utilizamos cookies para fornecer uma melhor experiência para nossos usuários, consulte nossa política de privacidade.

[Aceitar](#)