
INTRODUÇÃO
DESIGN RESPONSIVO

Introdução

- Repare no site da CNN em 2012: Desktop

Introdução

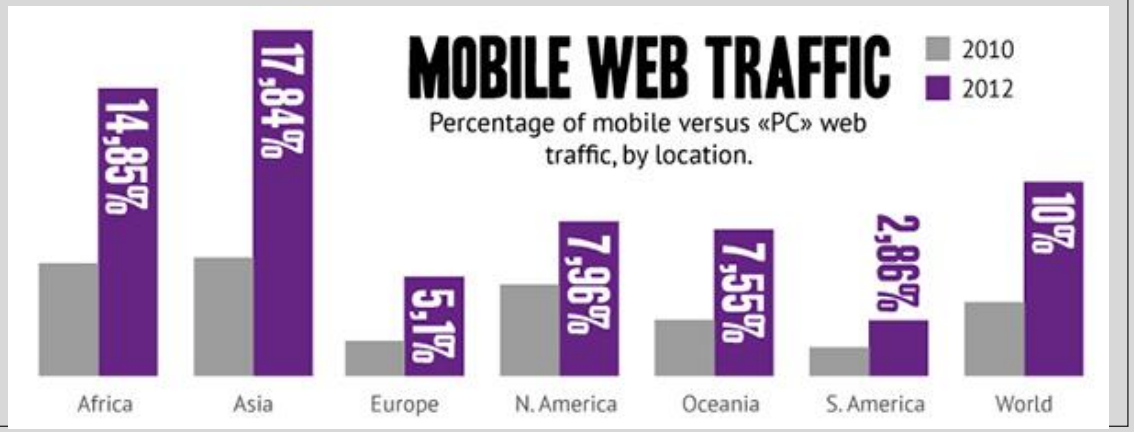
- Repare no site da CNN em 2012
- Visualização alterando as resoluções.
- Simulação (Outros dispositivos)
- Não faz scroll horizontal

Introdução

- E Agora ?
- Como está o site – www.cnn.com

Introdução

- Um site com Web Design Responsivo deve ser acessado de qualquer dispositivo
- Diga "Não" a versões especificas de seu site – "A WEB É ÚNICA"
- **Estatísticas do mundo mobile**
- Em 2010 as vendas de celulares ultrapassaram as de desktops
- O Brasil em 2010 bateu a casa de 200 milhões de celulares o mesmo número de habitantes na época.
- Em 2019 tínhamos 230 milhões de celulares ativo no país (Pop. 211 milhões).
- Em 2017 o acesso a web passou a ser prioritariamente pelo celular (49% da população utilizava apenas esse meio).



Princípios do Design Responsivo

- Como surgiu?
- O site "A list apart" → http://www.alistapart.com
  - Surgiu em 1998, sendo um site que publica livros e artigos
  - O escritor Ethan Marcotte em meados de 2010 publico um artigo intitulado "Responsive Web Design", onde o artigo começa assim:

"O controle que os designers têm no meio impresso e, muitas vezes, desejam ter no meio web, é simplesmente um reflexo da página impressa. Devemos aceitar o fato que a web não tem as mesmas restrições e projetar(o web design) para essa flexibilidade"

Trinca tecnológica do Web Design Responsivo

- Layout fluído;
- Imagens e recursos flexíveis; e
- Media Queries

Trinca tecnológica do Web Design Responsivo

- **Layout Fluido**
- Desenvolver sites com layouts fluídos (fluente layout) também chamados de grids flexíveis.
- O projeto deve primar desde sua concepção pela não especificação de medidas fixas.
- Isto torna natural e automática a adaptação do site a tela em que é exibido.

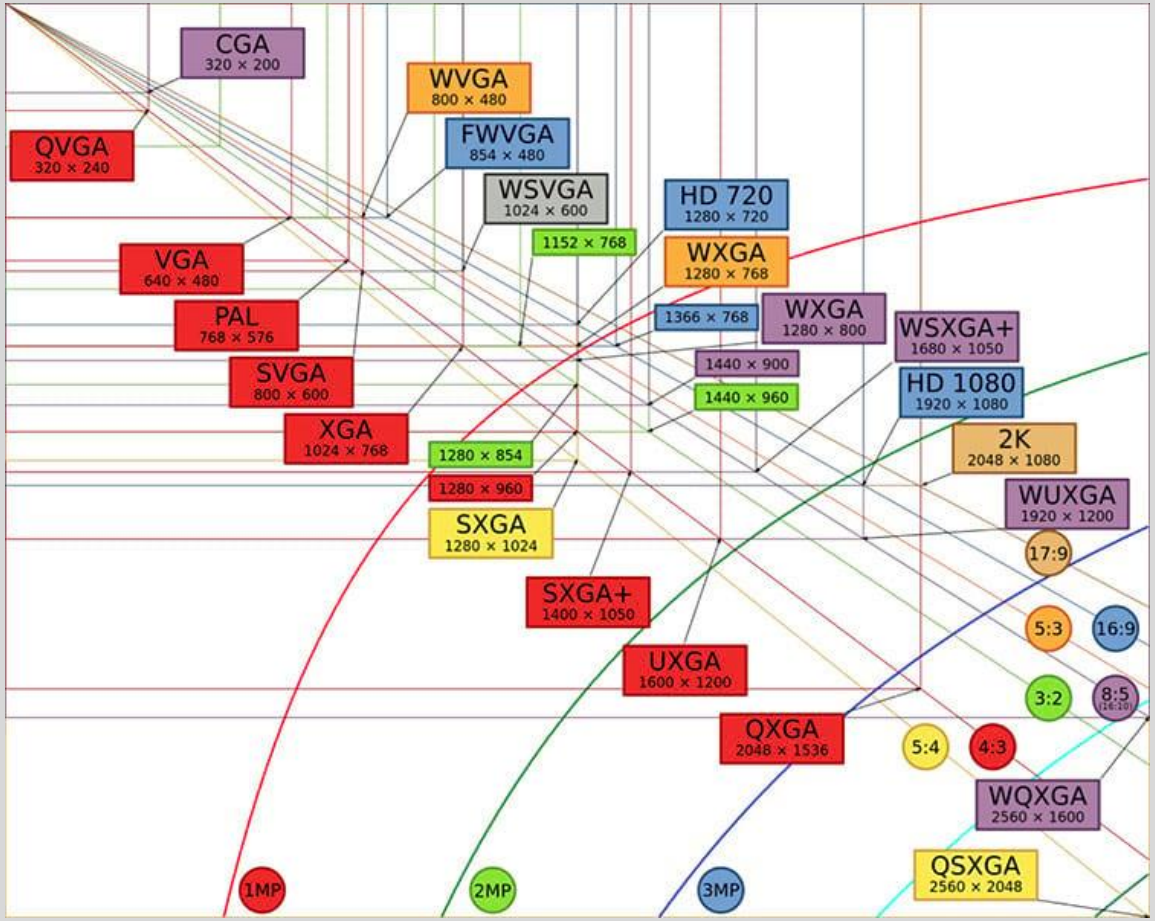
Trinca tecnológica do Web Design Responsivo

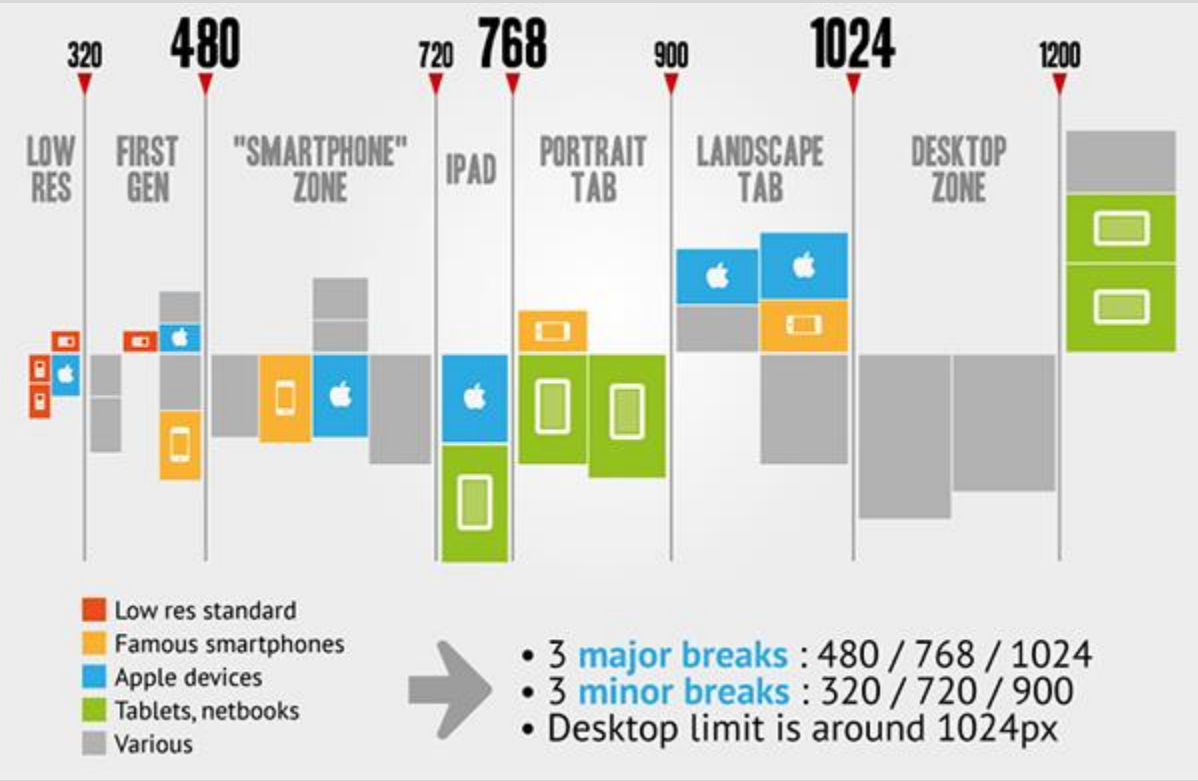
- **Imagens e recursos flexíveis**
- Não adiantaria nada tornar nosso site fluído se os recursos como imagens e vídeos também não fossem flexíveis para garantir que a experiência do visitante prime pela excelência independente do dispositivo utilizado.

Trinca tecnológica do Web Design Responsivo

- **Media Queries**
- Cada aparelho apresenta necessidades diferentes para o usuário utilizar o seu site.
- É com as Media Queries que é possível ocultar, fazer aparecer e reposicionar elementos e interações conforme a resolução atual que esteja sendo usada no momento da visitação.

Resoluções de tela





Layout Fluído

- **Não usar medidas absolutas no CSS!!!**
- Essa é a principal medida a ser tomada para um layout fluído
- Tipos de medidas em CSS:
  - **Pixel(px)** → é a unidade de medida mais usada, é um ponto indivisível na tela de exibição de um dispositivo. Não raramente web designers preferem essa medida para fazer sua estrutura html/css em *pixel perfect, não medindo esforços para estruturar seus documentos para que fiquem idênticos à imagem do web design.*
  - ***Ponto(point)*** → *pontos são tradicionalmente utilizados para CSSs de impressão. Um ponto é igual a 1/72 polegadas. Assim como os pixels, pontos são unidades de tamanho fixo.*

Layout Fluído

- **Ems(em)** → O "em" é uma unidade escalável. Quando se trata do tamanho da fonte, 1em é igual ao tamanho atual da fonte do documento-pai. Por exemplo, se o tamanho da fonte do elemento é 12pt, 1em é igual a 12pt. Ems são escaláveis por natureza, 2em seria igual a 24pt, 0.5 seria 6pt e etc.

- ***Porcentagem(%)*** → *A unidade porcento é muito parecida com a "em", mas possui algumas diferenças fundamentais. Em primeiro lugar, o atual tamanho da fonte é igual a 100% (ou seja 12pt=100%). Durante o uso da unidade porcento, o texto permanece totalmente escalável para dispositivos móveis.*

EMS x Porcento

- Devido as características que foram apresentadas, parece que é a mesma coisa usar “em” ou “%”, entretanto existem diferenças.
- O importante é saber que quando falamos em layout, marcar unidades com porcento fornece uma exibição mais consistente e acessível para o visitantes. Quando as configurações de exibição se alteram, as medidas marcadas por % se alteram de maneiram adequada.
- Deste modo é “consenso” entre desenvolvedores que usar “%” para lidar com tamanho de layout (larguras, margens, espaçamentos, etc) e usar “ems” para lidar com fontes.

Fórmula Mágica(Web Design Responsivo)

- Desde a concepção do projeto os desenvolvedores devem pensar de forma relativa (% e ems).
  - Por exemplo se na maneira antiga houvesse um título com tamanho 24px, então seria necessário converter em "em". **Mas como converter para "em" e continuar com a aparência de 24px?**

- Fórmula do Web Design Responsivo
  - **Resultado = alvo/contexto**
  - Alvo → elemento-alvo com medida atual
  - Contexto → onde o elemento-alvo está (baseado no elemento-pai)
  - Resultado → o valor relativo que se está procurando.

Fórmula Mágica(Web Design Responsivo)

- Com essa fórmula é possível realizar cálculos simples para saber as conversões de medidas absolutas de CSS para relativas.
- ***Existe uma certa convenção de que o tamanho padrão de fontes em browsers desktop é de 16px.***
- Então voltando ao exemplo anterior se querem um resultado parecido com **24px → 24/16 = 1,5 → 1,5em.**

- **Ok, mas se houver um link dentro do título, que no planejamento, teria 11px de tamanho. É preciso converter para a fórmula relativa aplicando a fórmula → 11 /16 = 0,6875em. Certo ???**

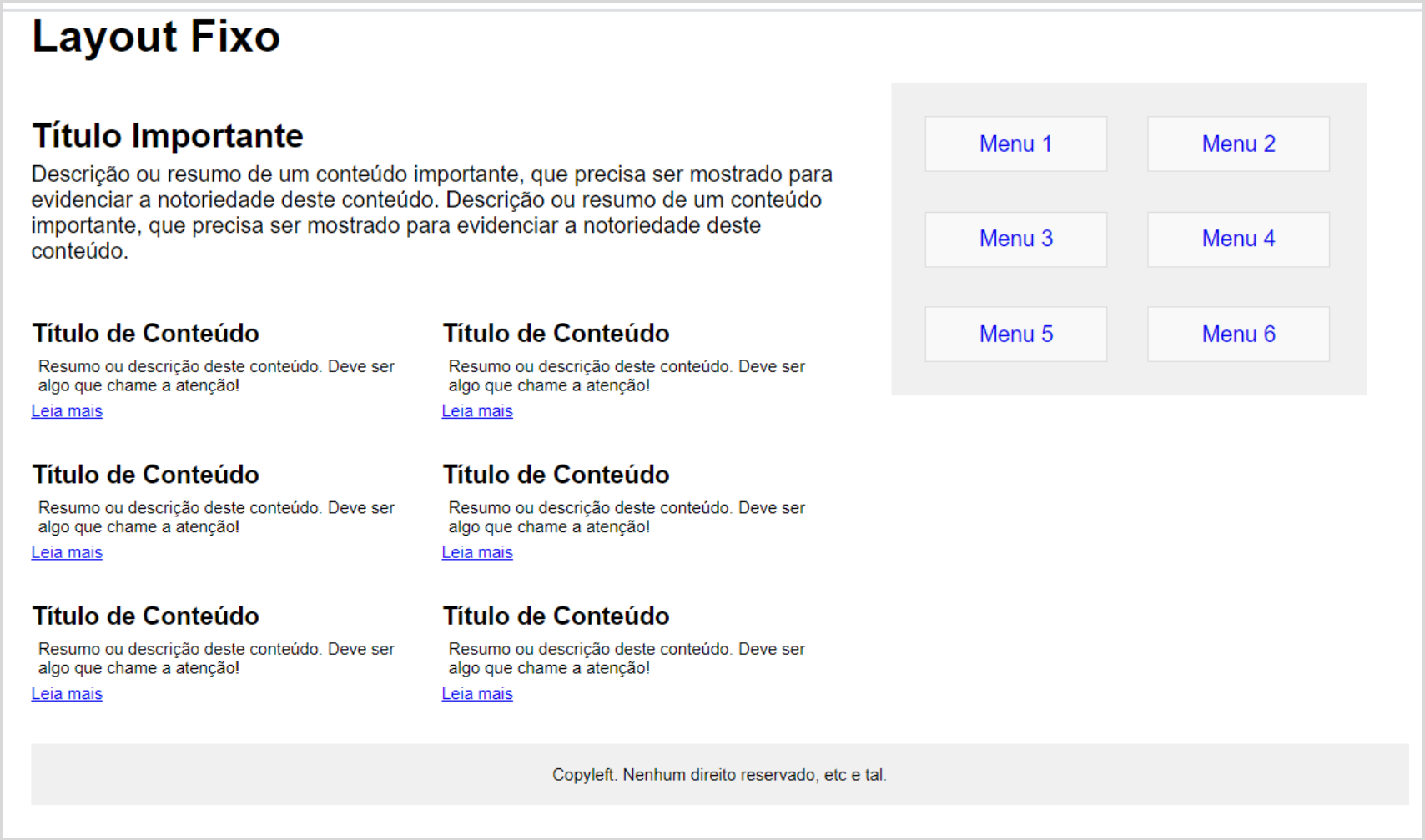
Fórmula Mágica(Web Design Responsivo)

- **Errado!! Pois nosso contexto mudou!!!**

- Agora o elemento-alvo não está mais no contexto geral da página (body). Então o resultado correto → **11/24 = 0,45833333em.**

- O elemento que, em medidas absolutas, deveria ter 11px, **está dentro do contexto do título** (que em medidas absolutas, teria 24px), que resulta na fórmula acima.

Exemplo de Layout Fixo



Metatag Viewport

- **Veremos como converter esse layout fixo em relativo**
- **Mas primeiro devemos nos ater a um dos mais importantes detalhes técnicos quando o assunto é web design responsivo.**
- **A metatag Viewport.**

Metatag Viewport

- **Metatags:**
- O termo meta tag é, na verdade, formado por 2 palavrinhas que se juntam para dar seu significado.

  - **Meta** → prefixo grego para "após", "que ultrapassa", "que engloba", "que está além". Este não é um prefixo utilizado somente no desenvolvimento web; você já deve ter ouvido falar em palavras como "metabolismo" ou "metamorfose". Este prefixo meta refere-se à **"coisa sobre a própria coisa"** ou, em outras palavras, dados sobre dados, informação sobre a informação.

  - **Tag** → é uma palavra do inglês que significa "etiqueta" ou "rótulo", em qualquer parte do mundo as etiquetas servem para identificar, nomear e marcar algo ou alguma coisa para que possa ser identificado e/ou corretamente catalogado.

Metatag Viewport

- **Metatags:**
- Juntando os termos “Meta” e “tag”, temos **meta tag**, que são as tags que descrevem o documento web a qual pertencem (metadados);
- as meta tags são para descrever informações sobre sites e páginas que as contém;
- Informam sobre qual conteúdo está ali e mostra informações extras a respeito deste conteúdo.
- As metatags são colocadas como elementos-filho do head no HTML. Sua sintaxe obedece a seguinte convenção

```
<meta name="nome-da-meta" content="conteúdo-da-meta">
```

Metatag Viewport

- **Metatag Viewport**

- Os browsers mobile tentam exibir páginas feitas para desktop ajustando, automaticamente, o zoom do display e isso pode ser problemático para os sites que já foram planejados/otimizados para telas pequenas.

- Felizmente, existe uma meta tag para contornar essa característica dos navegadores. É a meta tag viewport:

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

- Sendo possível que em "content" especificar uma diversidade de parâmetros e valores conforme o tipo de visualização que se deseja configurar às páginas.

Metatag Viewport

- **Metatag Viewport**

- Podemos pensar na viewport como "resoluções personalizadas" para os visitantes para determinados dispositivos.
- Os principais e mais usados parâmetros são:
  - Width → define a largura da viewport;
  - Height → define a altura da viewport;
  - Initial-scale → define a escala inicial (zoom) inicial da viewport.

  - Sendo possível usar mais de um parâmetro, ou mesmo todos, como valor de "content" da meta tag viewport.

Metatag Viewport

- **Metatag Viewport – exemplos (Apple)**

default
Width = 980px

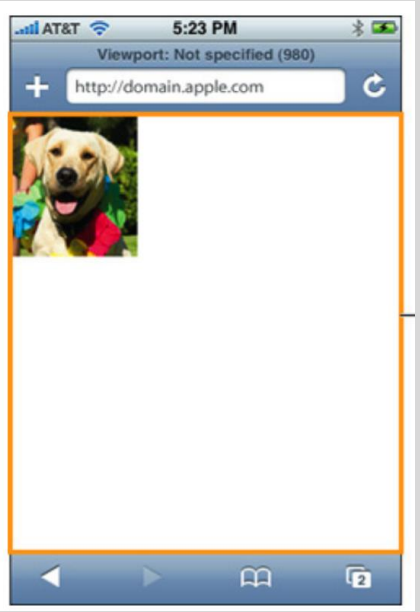
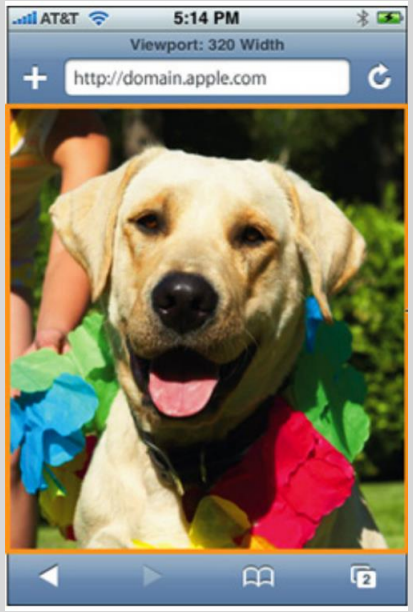
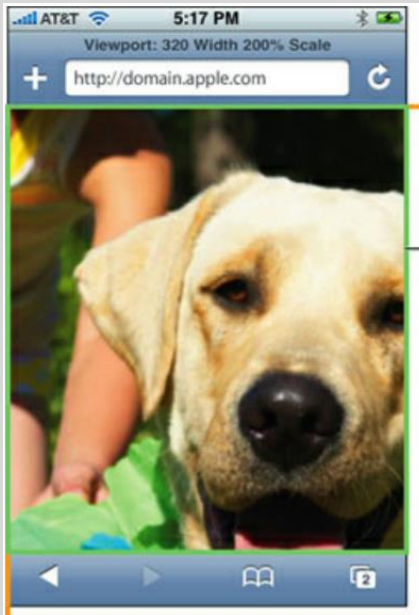


Imagem de 320x356px apresentada em um iphone.

"width=320,initial-scale=1"

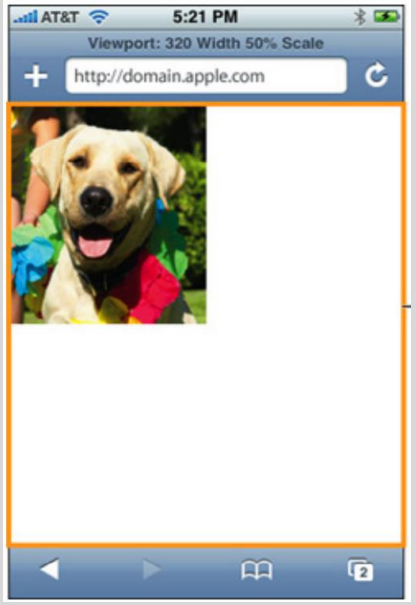


"width=320,initial-scale=1.5"



Zoom maior que a área visível

"width=320,initial-scale=0.5"



Nos dispositivos móveis, a pessoa também pode fazer zoom in e zoom out usando gestos.

Quando isso acontece, ela não altera o tamanho da viewport; altera a sua escala!. Consequentemente, o pan e o zoom não alteram o layout da página

Metatag Viewport

- **Configuração ideal da Metatag Viewport**
- Observando os exemplos do slide anterior, há uma questão que se levanta aos olhos. Quando pensamos em possibilidades de uso de "width": para contemplar todos os dispositivos móveis, seria necessário conhecer a largura de cada um deles!
- Então devemos utilizar uma forma de especificar automaticamente o tamanho do dispositivo, como se tivéssemos programado a largura padrão do dispositivo utilizado:

```
<meta name="viewport" content="width=device-width">
```

- Em condições normais, também é interessante que assim que renderize a página não haja alterações na escala inicial. Para isso devemos também utilizar a initial-scale=1.

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

Metatag Viewport

- **Tabela de parâmetros de Viewport**

Parâmetro	Exemplos	Valores possíveis	Padrão	Descrição
width	width=device-width width=320	1 a 10000 pixels ou device-width	device-width	Largura da viewport (em pixels)
height	height=device-height height=640	1 a 10000 pixels ou device-height	device-height	Altura da viewport (em pixels)
initial-scale	initial-scale=0.5 initial-scale=2.0	0.1 to 10	1.0 (sem zoom)	Zoom para o primeiro carregamento da página (valores altos causam "zoom in"; valores baixos, "zoom out")
user-scalable	user-scalable=no	yes ou no	yes	Se a página permite zoom pelo usuário
minimum-scale	minimum-scale=0.5 minimum-scale=1	0.1 a 10	0.25 (ajuste pelo valor de width)	Limite do "zoom out" do usuário
maximum-scale	maximum-scale=1.5 maximum-scale=2.5	0.1 a 10	5 (ajuste pelo valor de width)	Limite do "zoom in" do usuário
target-densityDpi	target-densityDpi=high-dpi	device-dpi (sem ampliação), medium-dpi (zoom de 1.3) ou high-dpi (zoom de 1.5)	device-dpi	Modifica a densidade da tela, ampliando a página para uma maior resolução (DPI)

Metatag Viewport

- **Cuidado!!!**
- Lembrando para tomar cuidade e prestar atenção no uso combinado dos parâmetros possíveis para não causar efeitos desagradáveis e/ou não planejados. Como em:

```
<meta name="viewport" content="width=device-width,initial-scale=1,maximu-scale=1">
```

- Isso impossibilitaria a pessoa que está acessando o site de dar zoom nas páginas.
- Se for que se pretende, tudo certo, senão tome cuidado.

Convertendo Layout fixo em fluído

- **Vamos alterar nosso CSS do modelo de layout fixo utilizando apenas medidas relativas.**
- A largura de .container é de 960px, isso não ajuda em nada a montagem da grid flexível. No caso do elemento container, não temos um contexto que sirva como base, então de forma abritraria especificamos a largura para o elemento, então vamos tentar uma largura relativa que, quando vista no browser se pareça com 960px

```
.container {
margin: 0 auto;
width: 67.5%; /* +/- 960 */
}
```

Convertendo Layout fixo em fluído

- Em relação ao valor 67.5% para o container, foi totalmente opcional e servindo somente para ilustrar o exemplo.

- Didaticamente acredita-se que é interessante preservar a característica de sites centralizados para a adaptação e mais fácil assimilação dos conceitos de web design responsivo.

Convertendo Layout fixo em fluído

- Entretanto poderíamos termos utilizado outras abordagens:
  - Deixar o width em 960px fixos e transformar o resto em "%" e "em". Não tão eficiente, já que vai ficar com esta largura independente da tela ser maior ou menor que 960px. Não seria uma solução responsiva.
  - "Responsivar" para telas mobile menores que 960px, mas manter os 960px para telas maiores. Usaria-se max-width:960px e continuaria igual para resoluções "desktop-like", mas, ao ser exibido em resoluções menores, vai se ajustar por não ter largura fixa.
  - A solução anterior é responsiva para telas pequenas, mas limita em 960px para telas maiores. Também é possível responsivar para todas , colocando width=100% no container. Para projetos que precisam/queiram contar com a largura total em quaisquer resoluções, está é a abordagem ideal.

Convertendo Layout fixo em fluído

- Vamos alterar a fonte h1 para medidas relativas.
- Lembre-se que o padrão de fonte para navegadores é 16px, então quer dizer que 32px (alvo) divido por 16px(contexto) é igual a 2(resultado). Então utilizando medidas "em" temos:

```
/* Antes */
h1 {
font-size: 32px;
}

/* Depois */
h1 {
font-size: 2em; /* 32 / 16 */
}
```

Convertendo Layout fixo em fluído

- O próximo elemento do layout → .content. Usando a fórmula temos que 15px (alvo) dividido por 960px (contexto) que é nossa antiga largura fixa é igual a 0.15625, mas como estamos lidando com % temos que multiplicar o seu valor por 100 → **1.5625%**

```
/* Antes */
.content {
margin: 15px 0;
}

/* Depois */
.content {
margin: 1.5625% 0; /* 15 / 960 */
}
```

Convertendo Layout fixo em fluído

- O objeto .content-main é um elemente de .content que possuía largura de 960px. Então temos a relação 593px (alvo) / 960(container pai).

```
/* Antes */
.content-main {
float: left;
width: 593px; /* Medida maior da Proporção Áurea aplicada ;-) */
}

/* Depois */
.content-main {
float: left;
width: 61.7708%; /* 593 (.content-main) / 960 (.container) */
}
```

Convertendo Layout fixo em fluído

- O elemento .hero (e seu descendente .brief), dentro de .content-main, também precisa ser convertido. Deve-se levar em conta o calculo da relatividade de .hero, **.brief é 5/593=0.008431.** Contudo apesar da matemática ser exata, desenvolver grids flexíveis demanda do fator humano então podemos arredondar alguns valores para melhor ajustes.

```
/* Antes */
.hero {
    margin: 25px 0;
}

.brief {
    margin: 5px 0;
}
```

```
/* Depois */
.hero {
    margin: 4.2158% 0; /* 25 / 593 */
}

.brief {
    margin: 1% 0; /* 5 / 593 */
    /* = 0.8431%, que, opcionalmente, pode ser arredondado para 1% */
}
```

Convertendo Layout fixo em fluído

- Prosseguindo, temos os elementos descendentes de .content-main.

```
/* Antes */
.last-contents {
font-size: 12px;
}

    .last-content-call {
    float: left;
    margin: 15px 15px 15px 0;
    width: 280px;
    }

        .last-content-call .brief {
        margin: 5px;
        }
```

```
/* Depois */
.last-contents {
font-size: .75em; /* 12 / 16 */
}

    .last-content-call {
    float: left;
    margin: 2.5295% 2.5295% 2.5295% 0; /* 15 / 593 (.content-main) */
    width: 47.2175%; /* 280 / 593 */
    }

        .last-content-call .brief {
        margin: 1.7857% 0; /* 5 / 280 */
        }
```

Convertendo Layout fixo em fluído

- A conversão da barra lateral, com todos os seus descendentes:

```
/* Antes */
.content-sidebar {
background-color: #F0F0F0;
float: right;
padding: 10px;
width: 322px;
}

    .main-nav ul {
    list-style-type: none;
    }
```

```
/* Depois */
.content-sidebar {
background-color: #F0F0F0;
float: right;
padding: 1.0416%; /* 10 / 960 */
width: 33.5416%; /* 322 / 960 */
}

    .main-nav ul {
    list-style-type: none;
    }
```

Convertendo Layout fixo em fluído

- A conversão da barra lateral, com todos os seus descendentes:

```
.main-nav li {
background-color: #F9F9F9;
float: left;
margin: 15px;
outline: 1px solid #DEDEDE;
text-align: center;
width: 130px;
}

    .main-nav a {
    display: block;
    padding: 10px;
    text-decoration: none;
    }
```

```
.main-nav li {
background-color: #F9F9F9;
float: left;
margin: 4.6583%; /* 15 / 322 (.content-sidebar) */
outline: 1px solid #DEDEDE;
text-align: center;
width: 40.3726%; /* 130 / 322 */
}

    .main-nav a {
    display: block;
    padding: 7.6923%; /* 10 / 130 */
    text-decoration: none;
    }
```

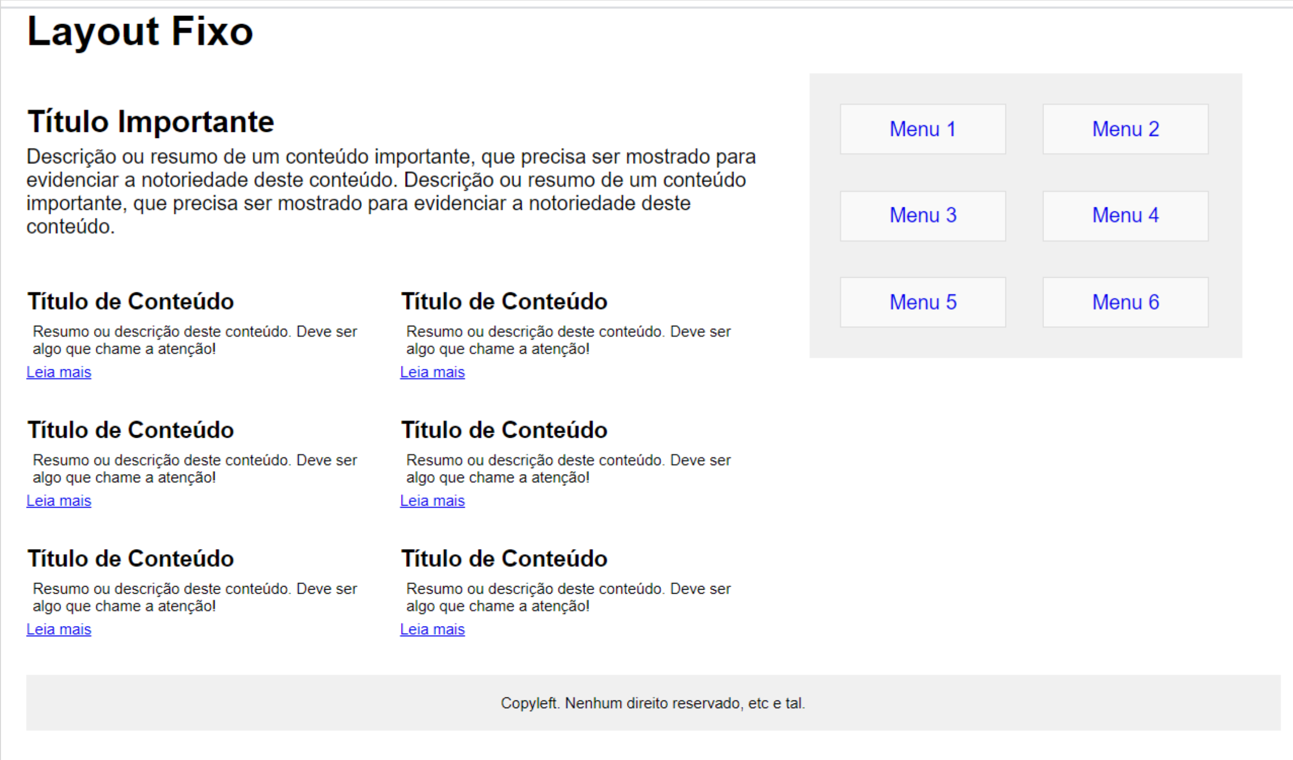
Convertendo Layout fixo em fluído

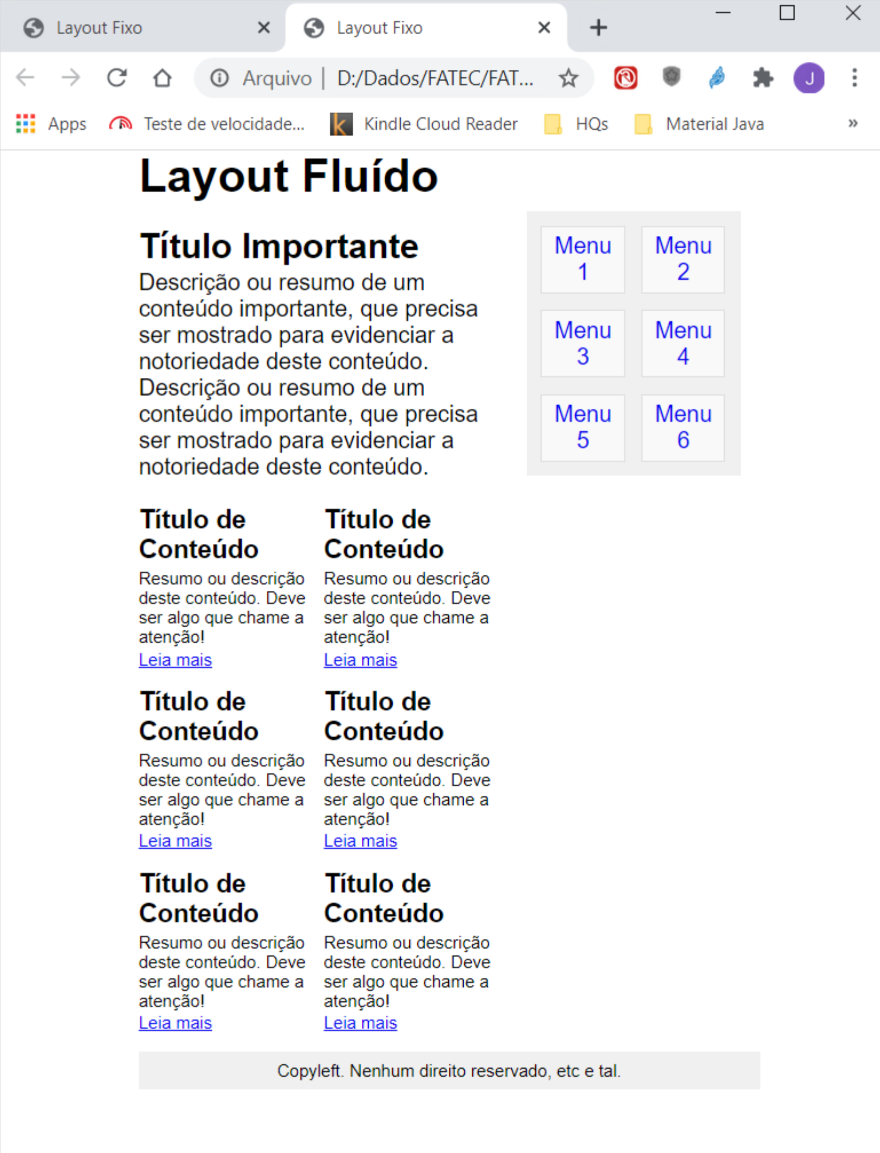
- Finalmente nosso humilde rodapé.

```
/* Antes */
.main-footer {
background-color: #F0F0F0;
clear: both;
float:left;
font-size: 12px;
margin: 15px 0;
padding: 15px;
text-align: center;
width: 100%;
}
```

```
/* Antes */
.main-footer {
background-color: #F0F0F0;
float:left;
font-size: .75em; /* 12 / 16 */
margin: 1.5625% 0; /* 15 / 960 */
padding: 1.5625%; /* 15 / 960 */
text-align: center;
width: 100%;
}
```

Resultado


Layout Fixo
Título Importante
Descrição ou resumo de um conteúdo importante, que precisa ser mostrado para evidenciar a notoriedade deste conteúdo. Descrição ou resumo de um conteúdo importante, que precisa ser mostrado para evidenciar a notoriedade deste conteúdo.
Menu 1
Menu 2
Menu 3
Menu 4
Menu 5
Menu 6
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Copyleft. Nenhum direito reservado, etc e tal.


Layout Fixo
Layout Fixo
Arquivo | D:/Dados/FATEC/FAT...
Apps
Teste de velocidade...
Kindle Cloud Reader
HQs
Material Java
Layout Fluído
Título Importante
Descrição ou resumo de um conteúdo importante, que precisa ser mostrado para evidenciar a notoriedade deste conteúdo. Descrição ou resumo de um conteúdo importante, que precisa ser mostrado para evidenciar a notoriedade deste conteúdo.
Menu 1
Menu 2
Menu 3
Menu 4
Menu 5
Menu 6
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Título de Conteúdo
Resumo ou descrição deste conteúdo. Deve ser algo que chame a atenção!
Leia mais
Copyleft. Nenhum direito reservado, etc e tal.