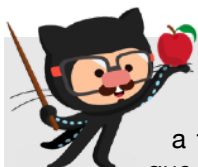


# DevWeb

## Capítulo 11

# Imagens Dinâmicas, Áudio e Vídeo

Fotos, áudios e vídeos são essenciais para a construção dos sites hoje em dia. Ilustrar bem o seu conteúdo é imprescindível para deixar clara a mensagem que queremos passar. Mas é preciso tomar cuidado com formatos e tamanhos, pois muita gente abrirá os sites em seus celulares e dispositivos móveis. Vamos falar sobre esse assunto no capítulo que iniciamos agora.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.



# Seu site deve se adaptar ao tamanho da tela

Como eu disse ali na introdução do capítulo, atualmente podemos acessar sites em todo tipo de dispositivo: smartphones, tablets, notebooks, computadores desktop, aparelhos de smart TV e muito mais. Pensando nisso, você que está começando a desenvolver sites vai precisar adaptar seu conteúdo a tamanhos de tela diferentes.



A mídia que mais sofre com esse problema de tamanho de exibição são as imagens. Uma TV pode exibir imagens gigantes, mas se usarmos essa mesma foto para ser apresentada em um celular, teremos que redimensionar forçadamente a imagem com CSS.

Porém, essa prática não vai fazer com que o tamanho (em bytes) diminua também. Isso acaba aumentando o consumo de dados em dispositivos móveis e deixando seu site muito pesado, e ninguém tem paciência para acessar site lento.



**CUIDADO!** Sites lentos aumentam a **taxa de rejeição** dos usuários e prejudicam a indexação da sua página em mecanismos de busca como o **Google**. Veja no vídeo a seguir, um especialista em SEO (otimização para mecanismos de busca) falando sobre lentidão de sites, principalmente via 3G e 4G do celular.

Portal SEO: <https://youtu.be/jWnMfvSdo1E>

## Imagens Flexíveis



Nosso primeiro passo no caminho de adaptar nosso conteúdo ao tamanho da tela vai ser aprender a gerar imagens de tamanho diferentes e a fazer o navegador carregar a imagem certa para cada situação. Para isso, devemos conhecer as tags `<picture>` e `<source>`.

Para esse exemplo, criamos as três imagens ao lado: a menor tem 300x300px, a média tem 700x700px e a maior tem 1000x1000px. Usamos o programa **GIMP**, que é um editor de imagens 100% gratuito. Essas imagens serão carregadas pelo navegador de acordo com o tamanho da janela atual. Para isso, criamos o seguinte código base:

```
<picture>
|   
</picture>
```

Note que colocamos a tag `<img>` exatamente como aprendemos no **capítulo 06** do nosso material. A novidade aqui é que inserimos essa imagem dentro da tag `<picture>`, que vai concentrar as outras fontes de imagem. Por padrão, a imagem **foto-g.png** (1000x1000px) será carregada.

O problema vai começar a surgir quando a janela do navegador chegar perto dos 1000 pixels de largura, pois a foto não vai mais caber lá. Vamos agora adicionar uma linha para resolver esse problema:

```
<picture>
  <source media="(max-width: 1050px)" srcset="foto-m.png" type="image/png">
  
</picture>
```

Note que a tag `<source>` possui três atributos:

- ▶ `type` vai indicar o *media type* da imagem que usamos (veja mais informações sobre *media types* no capítulo 10)
- ▶ `srcset` vai configurar o nome da imagem que será carregada quando o tamanho indicado for atingido
- ▶ `media` indica o tamanho máximo a ser considerado para carregar a imagem indicada no atributo `srcset`.



**ATENÇÃO!** Você pode até colocar o valor exato de 1000px na propriedade `max-width`, mas vai perceber que um valor ligeiramente acima vai gerar resultados mais interessantes.

Agora, recarregue seu código e mude o tamanho da janela do navegador. Você vai perceber que a imagem muda automaticamente conforme aumentamos ou diminuimos o tamanho da tela.

Vamos continuar e acrescentar mais um `<source>` à nossa imagem:

```
<picture>
  <source media="(max-width: 750px)" srcset="foto-p.png" type="image/png">
  <source media="(max-width: 1050px)" srcset="foto-m.png" type="image/png">
  
</picture>
```

É importante que existe uma ordem entre os `<source>`, e nessa nossa configuração, os itens mais acima sejam os menores tamanhos para `max-width` e que os seguintes sejam maiores, de forma crescente. O último item dentro de `<picture>` deve ser a imagem padrão.

Faça seus testes, modifique as ordens, entenda os resultados!

# Vamos falar sobre áudio

Quando comecei a produzir conteúdo para Internet, os vídeos ainda não eram uma realidade palpável, pois a Internet brasileira era muito lenta e cara. Iniciativas como Videolog e YouTube estavam começando a surgir, mas ainda mostravam vídeos muito curtos e com qualidade sofrível.

E foi em 2006 que comecei a produzir um podcast sobre tecnologia que chegou a ser bem conhecido e ter um número significativo de downloads por cada episódio.



Me lembro bem de que na época era bem complicado colocar conteúdo em áudio em um site. Precisava da adição de bastante JavaScript, manipulação de plugins específicos e um conhecimento bem consistente em RSS. A HTML não me ajudava em nada. Mais tarde, com a evolução dos vídeos e a melhoria da banda larga nacional, acabei optando por focar meus esforços no formato vídeo, mas tenho um carinho especial pelo áudio até hoje.



**QUER ME OUVIR?** Eu também participei de dois episódios de um dos maiores podcasts do Brasil: o **Nerdcast**. Dei meus pitacos sobre as profissões de Professor e de Programador. Você pode acessar o site abaixo e conferir.

[https://jovemnerd.com.br/?podcast\\_guest=gustavo-guanabara](https://jovemnerd.com.br/?podcast_guest=gustavo-guanabara)

Com a HTML5, veio também a facilidade em compartilhar áudio nos nossos sites e sem depender necessariamente de JavaScript ou plugins extras. A partir de agora, basta uma tag `<audio>` e alguns `<source>` para fazer seu site ser capaz de tocar qualquer áudio.

```
<audio preload="metadata" controls autoplay loop>
  <source src="midia/guanacast-33.mp3" type="audio/mpeg">
  <source src="midia/guanacast-33.ogg" type="audio/ogg">
  <source src="midia/guanacast-33.wav" type="audio/wav">
  <p>Seu navegador não suporta áudio <a href="midia/guanacast-33.mp3"
    download="guanacast-33.mp3" type="audio/mpeg"></a></p>
</audio>
```

Vamos analisar os principais atributos da tag `<audio>` antes de mais nada:

- ▶ O atributo `preload` indica se o áudio será pré-carregado ou não e aceita três valores:
  - `metadata` vai carregar apenas as informações sobre o arquivo (tamanho, tempo, informações de direitos, etc)
  - `none` não vai carregar absolutamente nada até que o usuário clique no botão play ou um script inicie a reprodução

- auto (padrão) vai carregar o arquivo de áudio inteiro assim que a página for carregada, mesmo que o usuário nunca aperte o play
- ▶ O atributo controls vai apresentar o player na tela. Caso não seja colocado na tag <audio>, o controle será transparente e o usuário não poderá interagir com ele.
- ▶ O atributo autoplay, quando inserido, vai iniciar a reprodução do áudio assim que a página for carregada.
- ▶ O atributo loop vai fazer com que o áudio seja repetido eternamente assim que terminar a sua reprodução.

Dentro da tag <audio>, adicionamos vários <source> com formatos diferentes do mesmo áudio. Coloque na parte de cima o seu formato favorito. Os demais só serão carregados caso o de cima falhe. Caso todos falhem, criamos um parágrafo que permite o download do arquivo MP3 para ouvir no player padrão do dispositivo.



**GUANACAST:** Eu deixei no GitHub um episódio do meu antigo podcast. Basta acessar a área de exercícios de HTML, ir até a pasta ex011 e baixar o arquivo MP3 que está dentro da pasta mídia.



## E os vídeos?

Com a expansão da banda larga no Brasil (sim, eu sei que ainda estamos longe de algo ideal na maioria do território nacional) tornou-se possível até assistir um vídeo em 4K em nosso plano 4G do celular (é caro, mas possível).



Para inserir um vídeo em nosso site, podemos utilizar a nova tag <video> da HTML5, caso o arquivo esteja hospedado no nosso próprio servidor.

```
<video width="600" poster="thumb.jpg" controls autoplay>
  <source src="meu-video.webm" type="video/webm">
  <source src="meu-video.mp4" type="video/mp4">
  <source src="meu-video.ogv" type="video/ogg">
  <p>Infelizmente seu navegador não conseguiu carregar o vídeo.</p>
</video>
```

Antes de mais nada, vamos criar a tag <video> e configurar alguns atributos importantes:

- ▶ width vai indicar a largura que o vídeo vai ter na tela. Nesse exemplo, 600px.
- ▶ poster configura uma imagem que vai aparecer como uma capa, enquanto o visitante não aperta o play para reproduzir o vídeo
- ▶ controls vai configurar se os controles do vídeo vão aparecer na parte inferior da mídia. Por padrão, os controles não aparecerão, mas basta colocar a palavra controls na tag <video>.
- ▶ autoplay diz para o navegador se o vídeo vai começar a tocar automaticamente, assim que a página for carregada.

## Que formatos são esses?

Arquivos de vídeo não são tão simples quanto imagens e áudios, onde o formato indica o padrão para abrir e reproduzir a mídia. Vídeos possuem formatos e codecs e isso pode tornar o vídeo inviável de ser reproduzido pela maioria dos navegadores na maioria dos dispositivos. É preciso prestar muita atenção nesse pequeno detalhe.



Os formatos suportados são MPEG, WEBM e OGG, mas os dois primeiros são os que possuem maior compatibilidade com os navegadores atualmente.

Navegador	Arquivos compatíveis
Microsoft Edge	.mp4 .m4v
Apple Safari	.mp4 .m4v
Google Chrome	.mp4 .m4v .webm .ogv
Mozilla Firefox	.webm .ogv
Opera	.webm .ogv



**UM ÓTIMO CONVERSOR:** Para gerar arquivos em vários formatos e usando codecs padronizados, recomendo usar o programa open source chamado **Handbrake**, disponível para várias plataformas.

<https://handbrake.fr/downloads.php>

## Hospedar seus próprios vídeos pode ser caro

Quando colocamos vídeos no nosso próprio servidor, podemos passar por problemas com alto consumo de banda, site lento e incompatibilidades com alguns navegadores por conta dos codecs. E geralmente só percebemos esses problemas quando colocamos nosso projeto no ar e lançamos oficialmente.

Vamos fazer uma conta simples: um vídeo simples, com poucos minutos, em formato mp4 com codec padrão deve ocupar uns 150MB com facilidade. Agora imagine que você lance seu site e 200 visitantes (um número super possível) acessem seu site e reproduzam o vídeo. Pronto, você acabou de utilizar 29GB de tráfego! Imagine o quanto isso pode deixar seu site lento ou consumir sua franquia de hospedagem, caso não seja ilimitada.





E é claro que seu site possivelmente não vai ter apenas um vídeo, não vai ter apenas 200 visitantes em um dia e aí a conta só aumenta. É preciso tomar muito cuidado quando decidimos guardar nossos próprios vídeos.

## E agora, quem poderá nos defender?

Euuuuuu! Ou melhor, o **YouTube** ou o **Vimeo**! Esses são serviços para a hospedagem de vídeos que vai evitar consumir nossos próprios recursos de host contratado. Cada um tem suas vantagens e desvantagens:



O **YouTube** é o serviço de hospedagem de vídeos mais popular do mundo e é gerenciado pelo Google. Sua principal vantagem é que seus servidores são ultra rápidos. Por outro lado, a ideia do Google é deixar todos os vídeos públicos e disponíveis, o que pode ser uma dor de cabeça caso você queira limitar quem vai ter acesso a determinado vídeo (uma escola online, por exemplo, onde queremos que apenas os alunos matriculados possam assistir).

O **Vimeo** resolve o problema que apontamos anteriormente. Ele permite limitar quem vai poder ver o vídeo, o que é especialmente vantajoso para quem quer criar produtos em forma de vídeo, entregues por demanda dentro de um site personalizado. Como desvantagens desse serviço, ele é pago por uma taxa anual e seus algoritmos não são tão eficientes quanto os do YouTube, logo os vídeos apresentam pequenos travamentos às vezes.



## E dá pra incorporar com esses aí em HTML?

Para incorporar vídeos que você subiu no **YouTube** ou **Vimeo**, existem recursos que te dão o código pronto em HTML5.

No **YouTube**, abra o vídeo que você quer incorporar e clique no link **COMPARTILHAR** que fica abaixo do título (veja imagem a seguir).



Ao clicar no link indicado anteriormente, uma janela vai aparecer, te dando as opções de compartilhamento. Escolha o item INCORPORAR.

Compartilhar



<https://youtu.be/SDAr3NIbIVM>

COPIAR

O código HTML personalizado vai aparecer em uma nova janela de contexto, incluindo um botão que permite COPIAR o código com a tag do `<iframe>` que vai aparecer diretamente na sua página. Volte ao seu editor de código e cole a tag no seu arquivo HTML que vai apresentar o vídeo.

Quando você está usando o Vimeo, o procedimento é bastante semelhante. Abra seu painel de controle do serviço e vá para a sua lista de vídeos hospedados. Na lista de mídias armazenadas, clique no botão com reticências e escolha a opção

**Incorporação** (veja imagem abaixo). Na janela, você vai poder clicar no botão Código de incorporação e a tag `<iframe>` personalizada também vai aparecer.

Vídeo incorporado



```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/SD
Ar3NIbIVM" frameborder="0"
allow="accelerometer; autoplay;
encrypted-media; gyroscope; picture-
in-picture" allowfullscreen></iframe>
```

☐ Começar em 0:23

COPIAR

		<code>&lt;/&gt;</code> Incorporação
<input type="checkbox"/>	M02A03 - W3TotalCache: configurações e teste de perfo...	13:34 Substituir
<input type="checkbox"/>	M02A03 - W3TotalCache	Baixar 0 16:10
<input type="checkbox"/>	M02A02 - Seu WordPress muito mais rápido	Excluir Ocultar Copiar link

<https://vimeo.com/386829741>

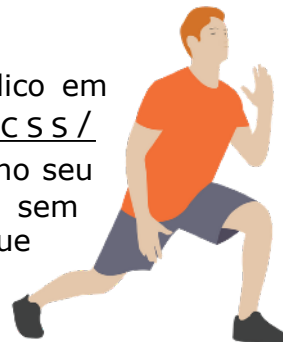
Código de incorporação





# Hora de exercitar

Chegou a hora de acessar o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/exercicios/> e executar o **exercício 011** e o **exercício 012** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



## Tenho desafios pra você!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d007**, o **desafio008** e o **desafio009**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!



Repositório em: <https://gustavoguanabara.github.io>

## Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo tem o conteúdo explicado como você leu aqui, só que de forma mais ilustrada. Reserve um tempo dos seus estudos para assistir esse vídeo todo.



Curso em Vídeo: [https://www.youtube.com/playlist?list=PLHz\\_AreHm4dlAnJ\\_jJtV29RFxnPHDuk9o](https://www.youtube.com/playlist?list=PLHz_AreHm4dlAnJ_jJtV29RFxnPHDuk9o)