

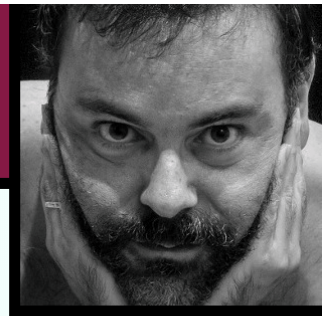
## Aprofundando no HTML

### Transcrição

Nas aulas anteriores, vimos um pouco de HTML e CSS e, com o que vimos, já conseguimos fazer uma página web simples, mas usável. Agora vamos além: faremos algo usável e bonito, seguindo boas práticas de código.

A página que fizemos ainda não está completa nem do ponto de vista visual (CSS) nem do ponto de vista de conteúdo (HTML). No HTML, faltam algumas imagens, o menu lateral e o rodapé da página, além de alguns links para auxiliar nossos visitantes na exploração do site.

# SOBRE MIM



## JOÃO DA SILVA

[home](#)  
[portfolio](#)  
[sobre mim](#)  
[blog](#)  
[contato](#)



Moro em São Paulo mas atendo clientes do mundo todo. Sou conhecido por fazer produtos de *qualidade, durabilidade* e que *agregam valor* para meus clientes.

Trabalho usando a web como plataforma, ou seja, respiro HTML5, CSS3 e JavaScript (ou melhor: ECMAScript). Crio sites para todos, seguindo as principais diretivas de **acessibilidade**, **responsividade** e **web semântica**, sem descuidar da qualidade de código.

## Como trabalho

Satisfazer meus clientes é prioridade. Para isso, garanto um processo de desenvolvimento altamente interativo, baseado em feedback contínuo. **Não trabalho com escopo fechado**: o cliente é que decide quando o produto está pronto.

Também não trabalho com prazos fechados: **qualidade é importante demais para ser sacrificada**.

João é o melhor desenvolvedor front-end com quem já trabalhei. Muito eficiente e muito capaz. Recomendo sem dúvidas!

*José Souza, Fiat*

## Experiência

João domina as tecnologias como ninguém. Eu apresentava um problema, ele tinha na ponta da língua a solução mais adequada com as tecnologias mais recentes.

*Manoel Santos, Petrobrás*

Já desenvolvi projetos para grandes empresas como BMW, UOL e IBM. Neles, o foco principal era entregar uma experiência imersiva e impactante para o usuário final sem descuidar do desempenho e da acessibilidade da página.

Também já fui contratado para transformar grandes portais, como Terra e G1, em páginas responsivas. Fui responsável por renovar o layout, reorganizar o conteúdo e re-escrever o código de forma mais reaproveitável.

## Comunidade

Procuro repassar meu conhecimento para a comunidade. Para isso, já dei [diversas palestras](#) e mantenho um [blog](#).

© João da Silva 2014

Vamos ver cada um deles com mais detalhes.

## Links

No nosso texto, citamos a realização de trabalhos para grandes empresas. Seria legal levar o nosso visitante a conhecer de perto esses trabalhos! Para isso, poderíamos transformar os nomes das empresas para as quais trabalhamos nesse trecho de texto:

Já desenvolvi projetos para grandes empresas como BMW, UOL e IBM...

em regiões clicáveis, em links.

grandes empresas como [BMW](#), [UOL](#) e [IBM](#). Neles, o foco é a acessibilidade da página.

Quando ele clicar no texto "BMW", queremos que o navegador abra a página da BMW (<http://www.bmw.com/> (<http://www.bmw.com/>)). Para tornar a região clicável, usamos a tag `<a>` e, para dizer para onde esse link levará nosso visitante quando clicado, usamos o atributo `href` dessa tag:

```
<p>... grandes empresas como <a href="http://www.bmw.com/">BMW</a>...</p>
```

Mais para baixo no texto, temos outro trecho onde pode ser interessante colocar um link

Para isso, já dei diversas palestras e mantenho um blog.

Podemos levar o visitante para assistir as palestras na página de portfolio ou ler alguns artigos no nosso blog. Então, no nosso HTML, fazemos

```
<p>... Para isso, já dei <a href="portfolio.html">diversas palestras</a> e mantenho um <a href="blog">
```

Repare que colocamos apenas os nomes dos arquivos nos atributos `href` mas, normalmente, o endereço que o navegador acessa é algo como `http://meusite.com/bio.html`. Então, se estamos num endereço como esse, para qual endereço o navegador vai quando clicamos nesse link do portfolio? Ele vai para `http://meusite.com/portfolio.html`, ou seja, **se você não passa o endereço completo, o navegador troca só o final dele**. Isso é o que chamamos de **link relativo**: relativo ao endereço que você está visitando no momento.

No entanto, nada impede que coloquemos o endereço completo no link:

```
<p>... Para isso, já dei <a href="http://meusite.com/portfolio.html">diversas palestras</a> e manter
```

Esse tipo de link é o que chamamos de **link absoluto**. Quando queremos endereçar outro site, precisamos usar esse tipo de link. Agora, quando queremos endereçar nosso próprio site, podemos usar links absolutos ou relativos. Mas qual *devemos* usar?

Imagine que você hospedou seu site com o endereço `joaodasilva.com`. Depois, para facilitar a vida dos seus visitantes, você resolveu mudar para `joao.com`. Se você usou links absolutos no seu site, o que acontecerá com eles? Vão apontar para um endereço que não existe mais! Usando links relativos, não temos esse problema. Então, quando queremos referenciar nosso próprio site, o ideal é usar links relativos.

Na verdade, qualquer link que **não comece** com `http` é considerado um link relativo. Ou seja, os seguintes links são relativos:

- `portfolio.html`
- `/posts/html5-e-css3.html`
- `//facebook.com`

Para entender como o navegador interpreta cada um deles, considere que você está atualmente na URL `http://meusite.com/paginas/bio.html`. Os links levarão para os seguintes endereços, respectivamente:

- `http://meusite.com/paginas/portfolio.html`
- `http://meusite.com/posts/html5-e-css3.html`
- `http://facebook.com`

Ou seja: o primeiro link troca somente a última parte do endereço. O segundo, começando com uma barra, troca tudo que vem depois do `meusite.com`, que é o **domínio** do site. E o último tipo de link, começando com duas barras, troca **tudo que vem depois das duas barras**. Ou seja, estivéssemos no endereço `https://meusite.com/portfolio.html`, o último link nos levaria para `https://facebook.com`.

## Imagens

Queremos colocar nossa foto na nossa página para dar um ar mais pessoal, criar uma relação mais direta com nossos visitantes. Para colocar imagens na página, usamos a tag `<img>`. Nessa tag, precisamos indicar qual imagem queremos que o navegador carregue. Para isso, a tag `<img>` tem o atributo `src`. Então, se queremos carregar o arquivo `eu.jpg`, colocamos no HTML

```

```

### Comunidade

Procuo repassar meu conhecimento para a comunidade. Para isso, já dei [diversas palestras](#) e mantenho um [blog](#).



Assim como nos links, o endereço dentro do `src` pode ser *relativo* ou *absoluto*.

Quando usamos imagens e outros recursos visuais na nossa página, precisamos lembrar de um caso de uso bastante importante: nem todos os nossos visitantes conseguem **enxergar** a nossa página. No Brasil, [cerca de 3.5% da população tem problemas graves de visão ou cegueira](https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/estatisticas-da-deficiencia-visual/) (<https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/estatisticas-da-deficiencia-visual/>). Porém, mesmo com deficiência visual, essa parcela da população acessa à Internet. Para conseguirem fazer isso, utilizam programas chamados de **leitores de tela**, que são capazes de ler o conteúdo de uma página e oferecer uma navegação pela página usando o teclado.

Agora, quando usamos uma imagem ou outro recurso visual, como o usuário com deficiência visual terá acesso a ele? No caso de uma imagem, como esse usuário vai conseguir saber que ela existe e qual o conteúdo dela? **Precisamos oferecer uma descrição da imagem** para esses usuários; fazemos isso por meio do atributo `alt` da tag `<img>`:

```

```

**Tornar nossa página acessível deve ser uma preocupação constante**, afinal não podemos excluir tanta gente do acesso ao nosso site. E já é uma preocupação constante do W3C, o órgão que cuida dos padrões da Web. Tanto é que o W3C definiu que **o atributo alt é obrigatório nas tags <img>**.

Ainda não podemos esquecer que não só as pessoas com deficiência visual não enxergam nosso site. Ferramentas de busca também não enxergam e, ao fornecer a elas uma página acessível, facilitamos o trabalho de extração e catalogação de conteúdo que elas realizam para fornecerem seus resultados. Na imagem abaixo, por exemplo, podemos ver que, ao buscarmos por "alura" no Google, além da página inicial, ele nos mostra páginas internas do site. O Google consegue nos dar essas páginas internas pois nosso conteúdo está bem marcado.

### Alura: Cursos online de Tecnologia

[www.alura.com.br/](http://www.alura.com.br/) ▼

Aprenda, crie e programe no **Alura**. Cursos online de tecnologia para você reinventar sua carreira. Acesso completo a todos os cursos. R\$197,00/mês

#### Como funciona

Como funciona o Alura. A plataforma que te torna um ...

#### Java

Curso online de Java, .NET, Web, Android, Banco de dados, JSF ...

#### Primeiro Passos

Dando os primeiros passos na web com HTML e CSS. 20 ...

[Mais resultados de alura.com.br »](#)

#### Mobile

Trilha Mobile. Os celulares estão por toda a parte. Aprenda ...

#### PHP e Rails

Trilha PHP e Rails. Pronto para criar aplicações web de ...

#### Java Avançado

Trilha Java Avançado. Está na hora de se tornar um mestre na ...

## Recomendações

Para dar mais credibilidade ao nosso trabalho, é legal colocarmos recomendações de outros profissionais que trabalharam conosco e aprovam nosso serviço. Então vamos citar algumas recomendações que fizeram do nosso trabalho:

João domina as tecnologias como ninguém -- Manoel Santos, Petrobrás

Podemos colocar esse texto em nossa página dentro de uma tag <p> como fizemos com o resto do texto:

```
<h2>Experiência</h2>
```

```
<p>João domina as tecnologias como ninguém -- Manoel Santos, Petrobrás</p>
```

```
<p>Já <a href="portfolio.html">desenvolvi projetos...</a>...</p>
```

Mas vamos retomar um pouco que acabamos de ver com as imagens: isso é acessível? É um texto como qualquer outro, então o computador consegue ler isso. Mas queremos que seja realmente um texto como qualquer outro? Não, esse não é um texto como qualquer outro, afinal estamos *citando* alguém. Da mesma forma que colocamos o título da página em uma tag <h1> para indicar que ele era, de fato, o título, precisamos colocar a citação dentro de uma tag que indique que ela é realmente uma citação! É aí que entra a tag <blockquote> :

```
<blockquote>João domina as tecnologias como ninguém -- Manoel Santos, Petrobrás</blockquote>
```

Mas essa frase não deixa de ser um parágrafo de texto, então podemos manter a tag <p> **dentro** da tag <blockquote> :

```
<blockquote><p>João domina as tecnologias como ninguém -- Manoel Santos, Petrobrás</p></blockquote>
```

Se você escrever isso no seu HTML, vai notar que o navegador já coloca um estilo diferente nessa tag, justamente porque ele identificou que é uma citação.

## Como trabalho

João é o melhor desenvolvedor front-end com quem já trabalhei. Muito eficiente e muito capaz. Recomendo sem dúvidas!

*José Souza, Fiat*

Satisfazer meus clientes é prioridade. Para isso, garanto um processo de desenvolvimento altamente interativo, baseado em feedback contínuo. produto está pronto.

Também não trabalho com prazos fechados: **qualidade é importante demais para ser sacrificada.**

Ainda podemos melhorar um pouco. O nome do autor da frase, por enquanto, está misturado com a própria frase. Podemos separar o nome da frase colocando-os em tags diferentes:

```
<blockquote>
  <p>João domina as tecnologias como ninguém</p>
  <p>Manoel Santos, Petrobrás</p>
</blockquote>
```

Mas, de novo, o texto "Manoel Santos, Petrobrás" não é um texto qualquer; é o nome do autor da citação. Então, novamente, podemos usar uma tag mais específica, que transmita essa informação. A tag que indica autores de citações é a tag `<cite>` :

```
<blockquote>
  <p>João domina as tecnologias como ninguém</p>
  <cite>Manoel Santos, Petrobrás</cite>
</blockquote>
```

No caso da tag `<cite>` , não precisamos colocar a tag `<p>` dentro. Na verdade, não faz sentido colocá-la: o nome do autor da frase não chega a formar um parágrafo de texto.

## HTML e semântica

Essa discussão que acabamos de fazer na verdade vale para todo tipo de conteúdo que queremos representar na página. Devemos descrever bem o conteúdo para que leitores de tela e buscadores consigam processar melhor esse conteúdo e, assim, tornar a informação mais acessível aos nossos visitantes.

Em outras palavras, devemos tentar deixar o nosso código HTML **semântico**, usando, para isso, as tags mais adequadas a cada tipo de conteúdo.

Vejamos outros exemplos. Na nossa página, teremos uma região com links na lateral direita da tela. Essa região terá um estilo próprio, diferente do resto do texto. Então é interessante isolá-la do resto do texto usando, para isso, uma tag. É bem comum fazermos isso usando a tag `<div>` :

```
<h2>Comunidade</h2>
<p>Procuro repassar meu conhecimento...</p>

<div>
  <a href="home.html">Home</a>
  <a href="portfolio.html">Portfolio</a>
```

```
<a href="bio.html">Sobre mim</a>
</div>
```

A tag `<div>` serve justamente para isso: criar regiões, *divisões* na página. Mas podemos ser mais específicos quanto ao tipo de região que queremos criar! Nesse caso, estamos especificando um conjunto de links para que o usuário possa **navegar** pelo site. Na versão mais recente do HTML surgiu a tag `<nav>`, que tem justamente esse propósito: indicar regiões da página que contêm links de navegação pelo site.

```
<h2>Comunidade</h2>
<p>Procuro repassar meu conhecimento...</p>

<nav>
  <a href="home.html">Home</a>
  <a href="portfolio.html">Portfolio</a>
  <a href="bio.html">Sobre mim</a>
</nav>
```

Visualmente, não teremos nenhuma diferença, mas agora os leitores de tela podem oferecer um atalho que leve o usuário mais rapidamente ao menu de navegação do site e os buscadores podem oferecer esses links no resultado da busca. Um HTML semântico melhora a acessibilidade do site para todos.

Um HTML semântico é tão importante que, em sua última atualização, a linguagem HTML incluiu diversas novas tags cujo único propósito é permitir uma marcação mais semântica para nossas páginas! Além da tag `<nav>` que acabamos de usar, temos diversas outras:

- `<main>` : conteúdo principal da página
- `<header>` : cabeçalho da página ou de uma região dela
- `<footer>` : mesma ideia da tag `<header>` para o rodapé
- `<aside>` : conteúdo auxiliar ao conteúdo principal, como links relacionados ao conteúdo
- `<article>` : conteúdo que, por si só, já tem um sentido completo, como um post de um blog ou uma notícia
- `<section>` : parte/seção de uma página ou texto

Há ainda muitas outras, que estudaremos aos poucos. Por hora, podemos usar algumas das tags para deixar nosso código mais semântico. O texto da biografia, por exemplo, pode ser embrulhado numa tag `<main>`. O `<nav>` da lateral pode ser embrulhado numa tag `<aside>` junto com o título e os outros links que ainda precisamos colocar para essa parte. Podemos colocar um rodapé na página com uma mensagem de copyright com a tag `<footer>`. Nosso código ficaria assim:

```
<!DOCTYPE html>
<html>
<head>
  <!-- aqui continua igual -->
</head>
<body>
  <main>
    <h1>Sobre mim</h1>
    <!-- texto que já escrevemos -->
  </main>
  
  <aside>
    <h1>João da Silva</h1>
    <nav>
```

```
<a href="index.html">Home</a>
<a href="portfolio.html">Portfolio</a>
<a href="bio.html">Sobre mim</a>
<a href="#">Blog</a>
<a href="contato.html">Contato</a>
</nav>
<!-- links das redes sociais -->
</aside>
<footer>
  © João da Silva 2014
</footer>
</body>
</html>
```

Mais para a frente colocaremos os links das redes sociais mas, por hora, podemos fazer mais uma melhoria do ponto de vista semântico nos links do `<nav>` : podemos indicar que eles formam uma lista de links. Para isso, usamos as tags `<ul>` e `<li>` :

```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="portfolio.html">Portfolio</a></li>
    <li><a href="bio.html">Sobre mim</a></li>
    <li><a href="blog.html">Blog</a></li>
    <li><a href="contato.html">Contato</a></li>
  </ul>
</nav>
```

A tag `<ul>` não é a única tag que existe para fazer listas. Existem também as tags `<ol>` e `<dl>` , que veremos mais para a frente.

## Entidades HTML

No footer, queremos colocar o símbolo e a mensagem de copyright. Podemos colocar sem problema algum o símbolo © direto no arquivo HTML mas, dependendo do teclado que você usa, pode ser difícil digitá-lo. Além disso, em algumas codificações de arquivo possíveis, o símbolo pode simplesmente não existir (no nosso exemplo, estamos usando o UTF-8, então não corremos esse risco).

Para solucionar esses casos, a linguagem HTML nos permite especificar caracteres especiais por meio das **entidades**. Então, em vez de digitar ©, podemos digitar `&copy;` e o navegador entenderá que queremos colocar esse símbolo na página.

Há diversas entidades que podemos usar no nosso código. Abaixo uma lista com alguns exemplos:

- `&euro;` : €
- `&yen;` : ¥
- `&pound;` : £
- `&reg;` : ®
- `&hearts;` : ♥

Também temos algumas entidades úteis para poder mostrar código HTML sem que o navegador o entenda como tags. Para podermos colocar o texto "`<h1>` serve para títulos" na página, por exemplo, precisamos escrever no HTML:



<p>&lt;h1&gt; serve para títulos</p>

As entidades &lt; e &gt; representam, respectivamente o símbolo de menor que (<) e maior que (>).

## Arrumando o CSS

Com as alterações que fizemos na página, o visual vai ficar meio estranho. Em particular, o conteúdo da nossa barra lateral ganhou um fundo roxo, igual ao título da página.

### Sobre mim

Moro em São Paulo mas atendo clientes do mundo todo. Sou conhecido por fazer produtos de *qualidade, durabilidade* e que *agregam valor* para meus clientes.

Trabalho usando a web como plataforma, ou seja, respiro HTML5, CSS3 e JavaScript (ou melhor: ECMAScript). Crio sites para todos, seguindo as principais diretrizes de **acessibilidade, responsividade e web semântica**, sem descuidar da qualidade de código.

### Como trabalho

João é o melhor desenvolvedor front-end com quem já trabalhei. Muito eficiente e muito capaz. Recomendo sem dúvidas!

*José Souza, Fiat*

Satisfazer meus clientes é prioridade. Para isso, garanto um processo de desenvolvimento altamente interativo, baseado em feedback contínuo. **Não trabalho com escopo fechado:** o cliente é que decide quando o produto está pronto.

Também não trabalho com prazos fechados: **qualidade é importante demais para ser sacrificada.**

### Experiência

João domina as tecnologias como ninguém. Eu apresentava um problema, ele tinha na ponta da língua a solução mais adequada com as tecnologias mais recentes.


*Manoel Santos, Petrobrás*

Já desenvolvi projetos para grandes empresas como BMW, UOL e IBM. Neles, o foco principal era entregar uma experiência imersiva e impactante para o usuário final sem descuidar do desempenho e da acessibilidade da página.

Também já fui contratado para transformar grandes portais, como Terra e G1, em páginas responsivas. Fui responsável por renovar o layout, reorganizar o conteúdo e re-escrever o código de forma mais reaproveitável.

### Comunidade

Procuo repassar meu conhecimento para a comunidade. Para isso, já dei diversas palestras e mantenho um blog.



### João da Silva

- [Home](#)
- [Portfolio](#)
- [Sobre mim](#)
- [Blog](#)
- [Contato](#)

© João da Silva 2014

Faz um certo sentido isso ter acontecido, afinal, no nosso CSS, temos a seguinte declaração:

```
h1 {  
  text-align: center;  
  background-color: #851944;  
  color: #FFF;  
}
```

Estamos selecionando **todos** os <h1> da página, inclusive o da barra lateral! Podemos arrumar isso de duas formas: ou não usamos a tag <h1> na barra lateral ou selecionamos o <h1> que queremos de algum outro jeito.

Faz sentido o `<h1>` da barra lateral ser, de fato, um `<h1>`, afinal ele é o título principal da barra lateral. Então vamos ver um outro jeito de selecionar o título do texto principal.

Se olharmos com atenção, veremos que o `<h1>` com fundo roxo é o único `<h1>` dentro da tag `<main>`. Podemos usar essa informação para selecioná-lo. Ou seja, podemos dizer que queremos só os `<h1>` dentro de `<main>`. Podemos dizer isso no CSS com o seletor `main h1`:

```
main h1 {  
  text-align: center;  
  background-color: #851944;  
  color: #FFF;  
}
```

Quando colocamos nomes de tags separados por espaços no seletor CSS, queremos dizer que um elemento deve estar **dentro** de outro.

Usando esse truque, já conseguimos arrumar o estilo que ficou quebrado. Ainda há bastante coisa para arrumar no visual, mas veremos como nos próximos capítulos.