

HTML E CSS

TRILHA DE APRENDIZAGEM DO ALUNO

SUMÁRIO

MÓDULO 1	3
[INTRODUÇÃO] ELABORAR LEIAUTE PARA WEBSITES	3
[MÍDIA 1] INTRODUÇÃO À CRIAÇÃO DE LEIAUTES	3
[MÍDIA 2] PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO	19
[MÍDIA QUIZ] ETAPAS DO PROJETO	32
[SÍNTESE] MÓDULO 1	32
MÓDULO 2	34
[INTRODUÇÃO] ANÁLISE DA USABILIDADE E ACESSIBILIDADE DA INTERFACE DO WEBSITE	34
[MÍDIA QUIZ] AQUECIMENTO	34
[MÍDIA 1] CARACTERÍSTICAS FUNCIONAIS DO PROJETO	35
[MÍDIA 2] MELHORIAS E RESPONSIVIDADE	47
[POSTAGEM] PORTFÓLIO	55
[SÍNTESE] CURSO HTML E CSS	55

MÓDULO I

[INTRODUÇÃO] ELABORAR LEIAUTE PARA WEBSITES

Olá, aluno (a)! Bem-vindo (a) ao curso de HTML e CSS do Senac. Aqui iniciamos nosso primeiro módulo – Elaborar leiaute para websites.

A codificação de toda a interface visual de um site deve ser responsiva e dinâmica, o que aumenta a exigência do mercado por profissionais qualificados que busquem o uso de ferramentas computacionais, recursos que expandam as possibilidades criativas e sigam os padrões da web. O módulo está composto de três mídias:

[Mídia 1] *Introdução à criação de leiautes*

[Mídia 2] *Planejamento e desenvolvimento do projeto*

[Mídia Quiz] *Etapas do projeto*

Vamos lá, bons estudos!

[MÍDIA 1] INTRODUÇÃO À CRIAÇÃO DE LEIAUTES

O conhecimento de HTML é o primeiro requisito para criar uma página web e cabe à CSS (Cascading Style Sheets) a apresentação desse conteúdo.

Para saber mais, acesse a mídia, clicando em **[Mídia 1] Introdução à criação de leiautes**.

Olá, aluno (a)!

Seja bem-vindo (a) ao curso de HTML e CSS. Aqui vamos começar a desenvolver leiaute para websites, conforme propostas gráficas de comunicação definidas no projeto.

Palavra do autor

Olá, aluno (a), eu sou o Vitor Gabriel Martines Weinförtnner, autor do curso de HTML e CSS e gostaria de desejar as boas-vindas ao nosso curso!

No primeiro módulo vamos desenvolver a seguinte ação: Elaborar leiaute para websites, conforme propostas gráficas de comunicação definidas no projeto. Para isso, vamos entender:

- como surgiram e como funcionam essas linguagens tão utilizadas no desenvolvimento web;
- boas práticas de desenvolvimento, características e padrões de projeto, conceitos de prototipagem, o que são wireframes, imagens vetoriais e bitmaps.

E a partir dos conhecimentos construídos vamos desenvolver o nosso website. Durante o curso vamos desenvolver juntos um currículo online, que poderá ser utilizado como portfólio para você se destacar no mercado de trabalho. No currículo, você poderá colocar, além dos conhecimentos construídos anteriormente, o melhor curso de todos: o de HTML e CSS do Senac.

Brincadeiras à parte, espero que aproveite ao máximo o conteúdo e vamos entender melhor como tudo isso funciona.

Bons estudos!

Para elaborar leiautes para websites, conforme propostas gráficas de comunicação definidas no projeto, precisamos entender como surgiram e como funcionam essas linguagens tão utilizadas no desenvolvimento web, bem como boas práticas de desenvolvimento, características e padrões de projeto, conceitos de prototipagem, o que são wireframes, imagens vetoriais e bitmaps. A partir dos conhecimentos construídos vamos desenvolver o nosso website.

Protótipo

Durante o curso vamos desenvolver juntos um currículo online, que poderá ser utilizado como portfólio para você, aluno, se destacar no mercado de trabalho. Nele você poderá colocar, além dos conhecimentos construídos anteriormente, o curso de HTML e CSS do Senac, que fará muita diferença no seu currículo.

Brincadeiras à parte, espero que aproveite ao máximo o conteúdo e bora entender melhor como tudo isso funciona!

[MOD 1] Atividade: Criação de leiaute com editores de texto

Podemos criar uma página utilizando HTML e CSS até mesmo com os editores de texto mais simples, como o bloco de notas do Windows, por exemplo. Isso ocorre porque o responsável por interpretar as instruções que escrevemos é o navegador que o usuário utiliza.

Com base em seus conhecimentos, sobre o bloco de notas, observe as alternativas a seguir, atribuindo V para as alternativas que julgar verdadeiras e F para as falsas.

- () Podemos salvar nosso documento utilizando a tecla “ctrl” junto com a letra “s”.
- () Possui marcações para guiar o autor do texto durante a sua criação.
- () ctrl + S é a única maneira de salvarmos um arquivo no bloco de notas.
- () Serve para criar ou editar arquivos de texto que não exijam formatação.
- () Para salvar um documento com uma extensão diferente da .txt devemos utilizar a opção “Salvar como...”.

Agora, assinale a alternativa correta:

- a. F – V – V – F – F
- b. V – F – F – V – V
- c. V – V – V – F – F
- d. V – F – V – F – F

World Wide Web

Hoje em dia é muito difícil imaginarmos a nossa vida sem a internet, não é mesmo?

A web surgiu muito diferente de como conhecemos hoje. No início era utilizada no meio acadêmico e militar. Em 1980, o suíço Tim Berners-Lee criou o primeiro servidor web, que possibilitava o acesso às informações através da rede. Hoje em dia essa rede de computadores cresce a propor-

ções gigantescas, com conteúdo e informações quase infinitos disponíveis na palma da nossa mão (literalmente falando se pensarmos nos smartphones).

Com a rede de computadores atual, a internet está cada vez mais pessoal, permitindo que qualquer um publique informações, realize operações e até crie seu próprio site, como vamos aprender durante o curso, aproximando as pessoas de qualquer lugar do mundo.

Existem várias métricas que servem para auxiliar nas melhorias dos serviços para os usuários ou clientes. Depende do objetivo. Elas também permitem a validação se um site se encaixa nos padrões web pois as métricas são focadas em manter a melhor experiência possível para o usuário final.

Refleta

E para você, o que faz com que um site seja bom? Quais são os parâmetros que para você fazem com que um site seja legal e que o fazem voltar a visitar esse site?

Camadas de desenvolvimento: front-end e back-end

As páginas web são formadas por duas camadas de desenvolvimento, a camada front-end e a camada back-end. Essas camadas tratam da estrutura de qualquer software.

Back-end

A camada back-end, também conhecida como servidor, responsável pelas regras de negócio dos softwares, é a que irá receber uma solicitação do usuário, processar as informações, realizando cálculos, gravando e acessando informações do banco de dados, por exemplo, e devolver o resultado desse processo ao front-end.

Front-end

Já a camada front-end é responsável pela parte visual do software. É também conhecida como cliente. É essa camada que vemos quando acessamos algum site ou sistema. É por meio dela que fazemos solicitações para a camada back-end, como por exemplo o sistema de uma calculadora. Através do front-end escrevemos o que desejamos calcular e a operação que queremos fazer. O front-end envia essas informações para o back-end, onde são processadas, e exibe o resultado que retornou dessa solicitação. É nessa camada que são utilizadas as linguagens HTML e CSS, as quais vamos utilizar durante o curso para a elaboração de layouts e implementar técnicas para melhorar a usabilidade e acessibilidade da nossa página web.

HTML e CSS

Para facilitar o acesso a informações e serviços são criados os sites. A parte visual destes é disponibilizada através de uma estrutura de comandos interpretados pelos navegadores, de forma que apareçam da maneira que quisermos ao usuário que acessar essas informações. Por meio dos comandos em HTML montamos a estrutura de como as informações serão exibidas. Podemos passar para o navegador onde ficará o título da página e onde estarão os parágrafos do texto, por exemplo. Já o CSS fica responsável por estilizar essa estrutura, ou seja, deixar bonito e agradável para o usuário. Através dessas duas linguagens podemos elaborar layouts para websites da maneira que desejarmos, tornando possível oferecer algum serviço ou divulgar informações que poderão ser

acessadas pelo mundo todo. Vamos ver essas duas linguagens na prática durante o nosso curso.

O que acha de conhecermos mais sobre o HTML?

O começo de tudo

O HTML foi desenvolvido por Tim Berners-Lee, na Suíça, em 1991, e desde então fica cada vez mais popular. Por isso, passou durante os anos por diversas atualizações, a mais recente lançada em 2014, chamada de HTML 5, que adicionou diversos recursos à linguagem.

Expansão do HTML

Devido a essa expansão e popularidade, o HTML se tornou a linguagem padrão oficial na internet. É o mais básico das tecnologias de front-end e serve como base de estilização com CSS.

Criação de sites

O primeiro passo para a criação de sites é conhecer a linguagem de marcação HTML, que significa Hypertext Markup Language, ou linguagem de marcação de hipertexto, em português.

Linguagem de Marcação HTML

O HTML não é uma linguagem de programação. Isso porque essa linguagem não realiza operações, como uma soma matemática, por exemplo, não altera nem manipula os dados, apenas descreve o que fazer com eles; como e onde serão os parágrafos; onde teremos uma tabela; qual parte do texto é um link e para onde deve ser direcionado, por exemplo, de maneira que os navegadores de internet reconheçam e exibam esses dados. Por isso é conhecida como uma linguagem de marcação.

Hipertexto

Hipertexto pode também ser entendido como a escrita eletrônica. São basicamente textos, imagens, tabelas e links exibidos de maneira interativa, o que auxilia o leitor a ir de um texto ou imagem, por exemplo, a outro. Os links são grandes exemplos de hipertexto; eles aparecem destacados e sublinhados, e quando clicamos somos direcionados a outra página ou outra seção da mesma página.

Para exibir um conteúdo utilizando o HTML

Precisamos de um navegador que vai interpretar e exibir as informações que desejarmos. Para que o arquivo seja reconhecido corretamente deve ser salvo com a extensão .html ou .htm. Cada página HTML é formada por um conjunto de marcações que separam o conteúdo. Essas marcações são chamadas de tags.

Tags

As tags são sinais específicos que aparecem em pares na nossa página, indicando o início e o fim da nossa marcação. Um exemplo dessa marcação é a tag <p>; p é a letra reservada para indicar os parágrafos; para indicar que é uma tag devemos colocar o sinal de menor (<) no início e o sinal de maior (>) no fim. A seguir, podemos ver um exemplo de parágrafo escrito utilizando o HTML.

Tag de marcação para parágrafo

<p>Aqui está nosso parágrafo utilizando as tags de marcação HTML!</p>

Na instrução acima conseguimos verificar algumas informações importantes. Iniciamos nosso blo-

co com a tag de abertura do parágrafo <p>. Entre as tags fica o conteúdo do nosso parágrafo. Para finalizar, temos a tag de fechamento </p>, que se diferencia da tag de abertura pela barra após o sinal de menor. É dessa maneira que iniciamos e finalizamos uma tag no HTML. Essas tags podem ou não conter atributos, que são informações importantes para o bloco de informações.

Abertura	Conteúdo	Fechamento
<p atributo="valor ">	Aqui está nosso parágrafo utilizando as tags de marcação HTML!	</p>

Existe um conjunto de tags essenciais para que um arquivo HTML seja reconhecido corretamente pelo navegador. Algumas dessas tags são de: comentário; estrutura; título; texto; link; imagem; listas; formulários e alguns atributos. Para acessá-lo ou baixá-lo, abra o PDF clicando no ícone abaixo.

Tags e atributos essenciais

Segue o conjunto de tags essenciais para que um arquivo HTML seja reconhecido corretamente pelo navegador. Essas tags são:

<!doctype html> - Devemos iniciar nosso arquivo com essa tag. Ela é utilizada para que o navegador reconheça que esse arquivo é do tipo HTML.

<html> - Após a tag inicial, vai abrir no início e fechar ao final do documento, informando para o navegador que nessa região fica o conteúdo do site.

<head> - A tag head identifica o que é importante para o navegador, mas não é exibido na página, como por exemplo a tag de título **<title>**, que altera o título da página, e a tag **<meta>** de que vamos falar a seguir.

<meta> - Define a codificação da linguagem que vamos utilizar no documento. Temos como padrão a UTF-8. Essa tag torna possível que o navegador interprete caracteres especiais e letras com acento, por exemplo.

<body> - Fica logo após o fechamento da tag **<head>** e é nesse bloco que fica o corpo do nosso site, ou seja, todas as informações que serão exibidas para os usuários.

Podemos visualizar a estrutura básica de um documento HTML a seguir:

```
<!doctype html>
<html>
  <head>
    <title> Titulo da Página </title>
    <meta charset="utf-8" />
  </head>
  <body>
    <p>Aqui está nosso parágrafo utilizando as tags de marcação HTML!</p>
  </body>
</html>
```

TAG DE COMENTÁRIO

São reconhecidos como comentários em arquivos HTML os textos entre as seguintes chaves (lembrando que os comentários não são exibidos em tela, somente são informações necessárias durante o desenvolvimento):

`<!-- Comentário em uma página HTML -->`

main

TAG DE ESTRUTURA

<div> Elemento que é um verdadeiro coringa do HTML, pode ser utilizado para definir quase toda a estrutura das nossas páginas, desde o cabeçalho ao rodapé. Utilizada para separar conteúdos e organizá-los dentro da página, sendo aplicada para separar elementos em blocos, mas que não carrega nenhum significado semântico.

<head>, é utilizada dentro da tag **<body>** e seu conteúdo é exibido no site.

Com o advento do HTML5, novas tags vieram para melhorar a semântica dos elementos estruturais do código, são elas:

<header> - Tag utilizada para definir o cabeçalho de uma seção específica (ou da própria página), contendo títulos, introduções e outros elementos similares; diferente da tag

<nav> O elemento nav representa uma seção da página que contém links para outras partes do website. A tag nav indica que um determinado grupo é uma seção de navegação, ou seja, é um bloco de navegação, porém nem todo grupo de links deve ser tratado como um nav.

```
<nav>
  <h3>Esporte</h3>
  <ul>
    <li><a href="#">Olimpíadas 2024</a></li>
    <li><a href="#">Brasileirão</a></li>
    <li><a href="#">Fórmula 1</a></li>
    <li><a href="#">Basquete</a></li>
  </ul>
</nav>
```

Esporte

- [Olimpíadas 2024](#)
- [Brasileirão](#)
- [Fórmula 1](#)
- [Basquete](#)

<main> - Tag que define o conteúdo central da página, ainda que este não esteja localizado de fato no centro.

<footer> - Utilizada para definir rodapé de uma página.

<section> - Utilizado para representar uma seção genérica, geralmente como um cabeçalho próprio e o seu conteúdo ou a home de um website que foi dividida em diversas seções: introdução, destaque, novidades, informação de contato e chamadas para conteúdo interno. O elemento section substitui o div em muitos momentos. Como exemplo segue a imagem retirada do site Globo.com



<article> - É muito importante para a separação de informações relativas ao conteúdo principal, para destacar um artigo, por exemplo.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Teste de tags estruturais</title>
</head>
<body>
  <header>
    <div>
      Essa é nossa div dentro da tag header
    </div>
  </header>
  <main>
    <section>
      Aqui está nossa section!
    </section>
    <article>
      Aqui está nosso article!
    </article>
  </div>
  Essa é nossa div dentro da tag main
</main>
<footer>
  <div>
    Essa é nossa div dentro da tag footer
  </div>
</footer>
</body>
</html>

```

Essa é nossa div dentro da tag header

Aqui está nossa section!

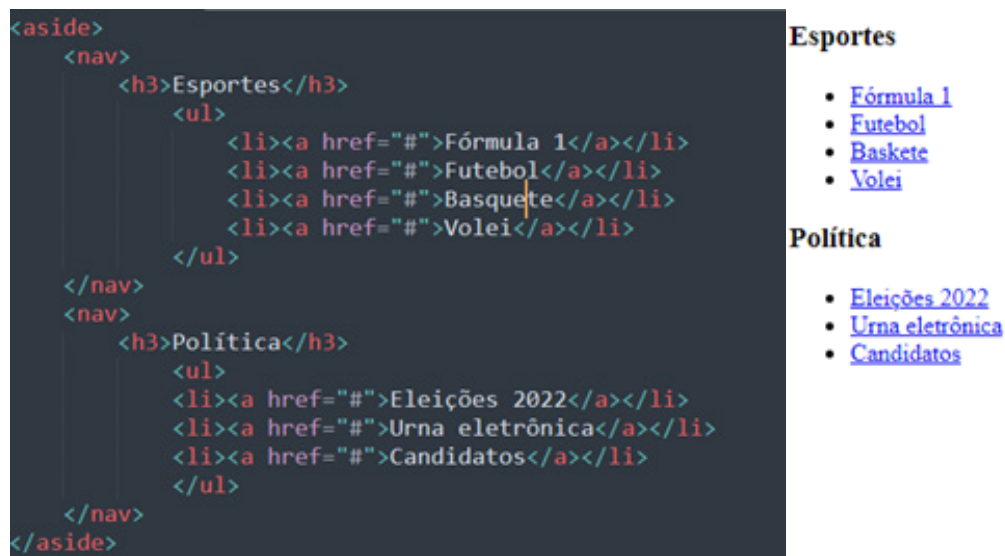
Aqui está nosso article!

Essa é nossa div dentro da tag main

Essa é nossa div dentro da tag footer

Exemplo de tags de estrutura, usadas para organizar a página. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

<aside> - Representa um bloco de informação relativas ao conteúdo principal. Pode ser utilizada em citações ou sidebars, agrupamento de publicidade, grupos e blocos de navegação ou para qualquer outro conteúdo que tenha importância secundária e relativa ao conteúdo principal da página.



TAG DE TÍTULO

Para as tags de título, temos as tags **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** e **<h6>**, utilizadas para destacar o texto como título dentro da página. O título mais importante fica dentro da tag **<h1>** e segue respectivamente até o título menos importante, adicionado à tag **<h6>**.



Exemplo de tags de título. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

TAG DE TEXTO

<p> - Como vimos anteriormente, essa tag é utilizada para identificar parágrafos de texto.

**** - Principalmente utilizada para adicionar elementos como imagens, legendas ou informações que vão adicionar conteúdo ao texto.

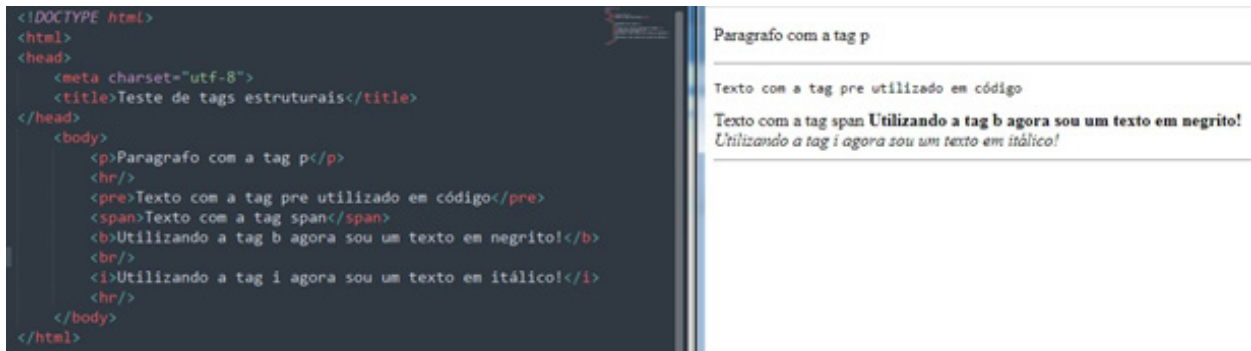
<pre> - Utilizada principalmente para a utilização de textos pré-formatados, como códigos, por exemplo.

**** - Transforma todo o texto do conteúdo em negrito.

<i> - Transforma todo o texto do conteúdo em itálico.

**
** - Utilizada para quebra de linha, essa tag não precisa de uma tag de fechamento.

<hr/> - Formação de linha horizontal no texto.

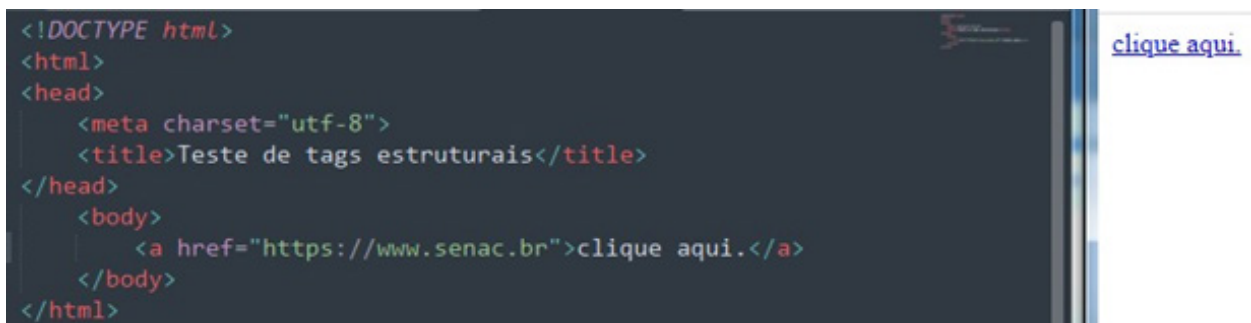


Exemplo de tags de texto. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

TAG DE LINK

Para adicionar links às nossas páginas utilizamos a tag **<a>**. Para indicarmos o caminho do link utilizamos o atributo **href**, como no exemplo a seguir. Um detalhe importante é que o link pode ser utilizado para navegar até mesmo dentro da própria página.

`clique aqui.`



Exemplo de link. Ao clicá-lo a página será redirecionada para o link atribuído como valor do atributo **href**, no caso, a página inicial do Senac. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

TAG DE IMAGEM

Para adicionar imagens ao nosso site utilizamos a tag ****, juntamente com o atributo **src**, que contém o caminho da imagem, como no exemplo a seguir:



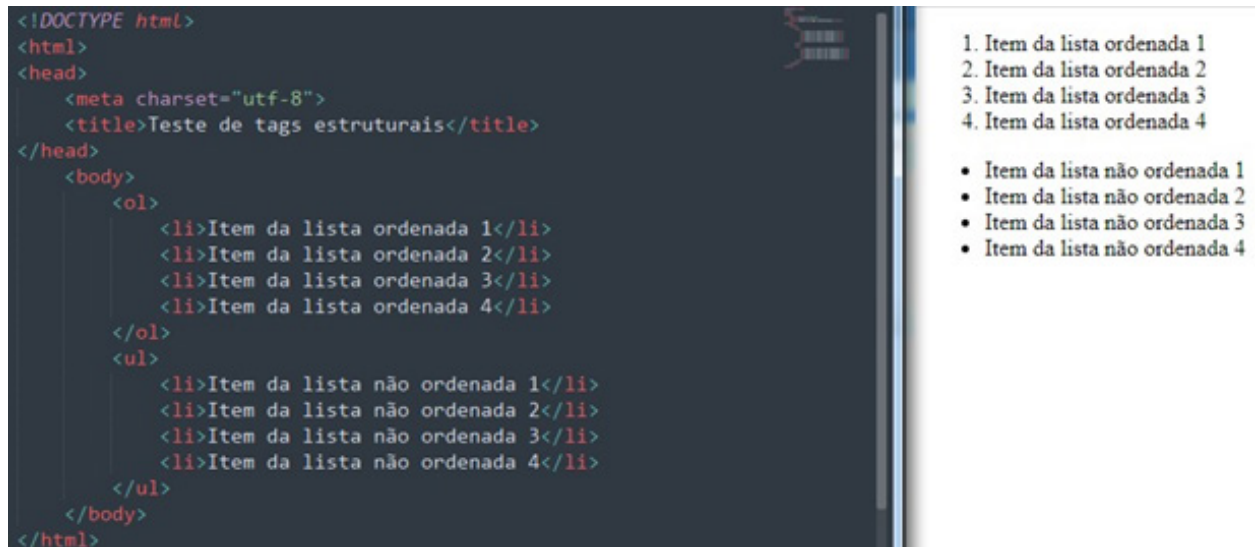
Exemplo de imagem. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

TAG DE LISTA

**** - Identifica o bloco como uma lista ordenada. Os itens serão identificados como 1,2,3 ou a,b,c por exemplo.

**** - Identifica o bloco como uma lista não ordenada.

**** - Item de uma lista.



The screenshot displays a code editor on the left and a browser preview on the right. The code defines an HTML document with a title 'Teste de tags estruturais'. It contains two lists: an ordered list (****) with four items labeled 'Item da lista ordenada 1' through '4', and an unordered list (****) with four items labeled 'Item da lista não ordenada 1' through '4'. The browser preview on the right shows the rendered output: a numbered list for the ordered list and a bulleted list for the unordered list.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Teste de tags estruturais</title>
</head>
<body>
  <ol>
    <li>Item da lista ordenada 1</li>
    <li>Item da lista ordenada 2</li>
    <li>Item da lista ordenada 3</li>
    <li>Item da lista ordenada 4</li>
  </ol>
  <ul>
    <li>Item da lista não ordenada 1</li>
    <li>Item da lista não ordenada 2</li>
    <li>Item da lista não ordenada 3</li>
    <li>Item da lista não ordenada 4</li>
  </ul>
</body>
</html>
```

- 1. Item da lista ordenada 1
- 2. Item da lista ordenada 2
- 3. Item da lista ordenada 3
- 4. Item da lista ordenada 4

- Item da lista não ordenada 1
- Item da lista não ordenada 2
- Item da lista não ordenada 3
- Item da lista não ordenada 4

Exemplo de listas. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

TAG DE FORMULÁRIO

<form> - Define um formulário.

<input> - Define um campo para a entrada de alguma informação, pode ser do tipo texto, número, data, etc.

<textarea> - Define uma área de texto.

<select> - Define uma caixa de seleção.

<option> - Fica dentro da tag **<select>** e define as opções da caixa de seleção.



The screenshot displays a code editor on the left and a browser preview on the right. The code defines an HTML document with a title 'Teste de tags estruturais'. It contains a form (**<form>**) with three elements: an input field (**<input>**), a text area (**<textarea>**), and a select box (**<select>**) with three options: 'Opção 1', 'Opção 2', and 'Opção 3'. The browser preview on the right shows the rendered output: a text input field, a text area, and a dropdown menu with three options, where 'Opção 3' is currently selected.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Teste de tags estruturais</title>
</head>
<body>
  <form>
    <input></input><br/><br/>
    <textarea></textarea><br/>
    <select>
      <option>Opção 1</option>
      <option>Opção 2</option>
      <option>Opção 3</option>
    </select>
  </form>
</body>
</html>
```

Aqui temos um input

Aqui temos um textarea

Opção 1 ▼
Opção 1
Opção 2
Opção 3

Exemplo de formulário. À esquerda podemos ver as instruções em HTML e à direita o resultado interpretado pelo navegador.

ALGUNS DOS PRINCIPAIS ATRIBUTOS

class - Indica classe do elemento, essa classe é utilizada para identificar um ou vários elementos.

Id - Indica o id do elemento, utilizado como um identificador único.

style - Inclusão de elementos CSS, que vão dar estilo ao HTML.

lang - Indica o idioma principal.

title - Título do elemento.

hidden - Utilizado para ocultar elementos.

align - Alinhamento do texto e de alguns elementos.

width - Largura do elemento.

height - Altura do elemento.

PROPRIEDADES MAIS POPULARES DO CSS

font-family - Define a família da fonte utilizada.

font-style - Define o estilo da fonte, que pode ser: normal, italic, oblique, initial e inherit.

font-weight - Controla a intensidade da fonte. Segue uma sequência de valores de '100' a '900'. Cada número indica uma fonte mais escura que o valor anterior.

font-size - Define o tamanho da fonte.

Letter-spacing - Controla o espaçamento entre as letras do texto.

word-spacing - Controla o espaçamento entre as palavras do texto.

Line-height - Controla a altura entre as linhas do texto.

text-align - Controla o alinhamento horizontal.

text-decoration - Define um efeito decorativo no texto que pode ser, por exemplo, underline (sublinhado) ou line-through (linha cortando o texto).

color - Define a cor do texto.

width - Define a largura de um elemento.

height - Define a altura de um elemento.

border - Define bordas para um elemento.

background - Define as propriedades relacionadas ao fundo de exibição.

margin - Controla as margens de um elemento.

padding - Controla os espaçamentos de um elemento.

float - Esta regra faz com que o elemento seja retirado de sua posição e flutuado para a direita ou para a esquerda.

text-align - Controla o alinhamento que um texto possuirá.

border-style - Controla o estilo da borda.

border-width - Controla a espessura.

border-color - Controla a cor da borda.

Folha de estilo CSS

Muito bem Aluno (a), a partir de agora iniciaremos os tutoriais sobre o funcionamento da Linguagem, mas antes ouça a explicação do autor acerca da organização das simulações em vídeo para que você tenha o entendimento necessário para obter o melhor aproveitamento do curso.

Aperte o PLAY e bom áudio!

Caso não consiga ouvir o áudio, leia a transcrição dele clicando abaixo.

Orientação sobre a criação de leiautes

Chegamos à parte mais legal, a de colocar a mão na massa!

Vamos iniciar a criação da nossa página web instalando o sublime, editor de texto muito útil para a criação de leiautes utilizando HTML e CSS, após a instalação vamos criar a nossa página de maneira mais didática, para facilitar o aprendizado. No final do curso, após finalizarmos as marcações e adquirirmos o conhecimento de HTML e CSS, vamos alterar nosso leiaute para que utilize a semântica correta do HTML5, que é o novo padrão de criação de leiautes com HTML.

Eu estou animado e você?

Decifrando o HTML

Na lâmina anterior, vimos muito sobre tags e atributos, e agora chegou a hora de você entender um pouco melhor como tudo aquilo funciona. Assista ao vídeo que o autor preparou para você sobre o assunto.

Caso não tenha conseguido acessar o vídeo, clique no PDF para conhecer sobre o assunto tratado.

Olá, aluno (a), agora vamos aprender sobre as tags básicas dos arquivos HTML!

Podemos criar arquivos HTML a partir de qualquer editor de texto, até mesmo dos mais simples. Para provar isso, vamos criar nossa primeira página a partir do bloco de notas, disponível nos sistemas Windows. Após abrir o bloco de notas, vamos adicionar o texto “Nosso primeiro parágrafo”. Vamos adicionar também o texto “Nosso segundo parágrafo”. Agora vamos salvar nosso arquivo através da opção Arquivo -> Salvar como...; é importante lembrarmos de adicionar a extensão

.html. Essa extensão é muito importante, pois é ela que diz para o navegador que este é um arquivo HTML. Já podemos acessá-lo pelo navegador. Estou utilizando o Google Chrome, mas você pode utilizar o de sua preferência. Aqui na nossa página já conseguimos visualizar o conteúdo, mas ainda não está organizado da maneira que queremos. Ainda não são dois parágrafos, é apenas uma linha com o conteúdo escrito no nosso documento. Vamos iniciar então as marcações da nossa página HTML para organizar esse conteúdo.

A primeira tag que um arquivo HTML deve conter é a DOCTYPE. Para isso, vamos abrir a tag com sinal de menor (<), seguido de exclamação (!), o nome da tag, que é o DOCTYPE, e o tipo do documento, ou seja, HTML, fechando a tag com sinal de maior (>). Após a DOCTYPE, temos outra tag muito importante para o nosso documento, que é a tag html. Vamos abrir e fechar a tag html. Essa tag deve ser aberta no início do arquivo e fechada no final. Dentro da tag html temos duas outras tags muito importantes, que são a tag head e a tag body. Vamos abrir e fechar as tags head e body. Na tag head vamos colocar informações importantes para o navegador, mas que não serão exibidas para o usuário. Já na tag body, vamos colocar o nosso conteúdo do site que será exibido para o usuário, no caso, os nossos parágrafos. Vamos colocar nossos parágrafos dentro da tag body. Agora que já temos as marcações principais do nosso arquivo, vamos abri-lo no navegador. O navegador já reconhece que ele é um arquivo em HTML, porém nosso texto continua sem estar formatado. Para resolver isso vamos utilizar a tag p, que é a tag para parágrafo. Vamos abrir a tag no início do parágrafo e após o parágrafo o nosso fechamento da tag. Essa tag, então, vai informar para o navegador que o bloco dentro dela é um parágrafo. Vamos adicionar também a tag no segundo parágrafo e fechá-la após o segundo parágrafo. Salvando o arquivo utilizando atalho ctrl + s, podemos abri-lo no navegador já da maneira que a gente quer, o primeiro parágrafo separado do segundo parágrafo.

Dentro da tag head temos informações muito importantes para o nosso navegador, como por exemplo a tag meta. Essa tag é um pouco diferente das tags que vimos até agora. Dentro dessa tag vamos ter o atributo charset e o valor UTF-8. Podemos verificar na tag meta que vamos abri-la com sinal de menor, seguido pelo nome da tag, ou seja, meta, com o atributo que desejamos alterar, que é o charset, ou seja, a codificação de caracteres. Após o carácter igual entre aspas, o valor do nosso tributo, ou seja, utf-8; aqui identificamos ao nosso navegador que esse documento será codificado com caracteres UTF-8. Essa tag vamos fechar de uma maneira um pouco diferente também das que vimos anteriormente, somente com a barra e o sinal de menor.

Alguns editores de texto disponibilizam que selecionemos o tipo de codificação do arquivo antes de salvá-lo. No caso do bloco de notas temos essa opção, e a padrão é a codificação ANSI. Vamos salvá-lo então com a codificação padrão. Ao abri-lo no navegador, podemos verificar esses caracteres estranhos no meio da nossa palavra. Isso acontece porque nós salvamos o nosso arquivo com a codificação ANSI, porém deixamos especificado que a codificação deve ser do tipo UTF-8. Para corrigir esse problema, vamos salvar novamente o arquivo com a codificação correta, a UTF-8. Agora podemos recarregar o navegador exibindo corretamente a frase que a gente deseja. É sempre bom ficarmos atentos à codificação, principalmente quando trabalhamos com arquivos vindos de servidores, onde a codificação do servidor pode ser diferente da codificação da nossa máquina local, onde se salvamos essa codificação diferente da que o arquivo solicita, pode gerar esses erros na tela.

Outra tag muito importante que fica no head da nossa página é a tag title. Lembrando de abrir e fechar a tag após o bloco que vamos utilizar. Dentro da tag title vai estar o título da nossa página. Esse título não vai aparecer no corpo da página, e sim na aba como título de informação e para

que o navegador reconheça esse documento. Vamos adicionar o título da página e acessá-la pelo navegador. Podemos verificar que o conteúdo da página não foi alterado, mas o seu título aqui em cima já ficou da maneira como a gente deseja.

Agora que sabemos como a linguagem HTML funciona e suas principais tags e atributos, vamos para o próximo passo: conhecer a linguagem de estilo CSS. Eu estou animado, e você?

Folha de estilo CSS

Assim como o HTML, o CSS não é uma linguagem de programação, pois não possui as características necessárias citadas anteriormente. É reconhecida como linguagem de estilo. É uma abreviação para Cascading Style Sheets, traduzido para o português como folhas de estilo em cascata. Pode ser entendido como um documento em que são definidas regras de formatação ou de estilos, a serem aplicadas aos elementos da marcação de um documento HTML, como cores, fontes, tamanhos, posições e muitas outras coisas.

Adicionando folha de estilo

Para adicionar a folha de estilo à nossa página, utilizamos a tag HTML **<style>**, identificando para o navegador que esse bloco vai conter informações em CSS, que serão interpretadas de maneira diferente das informações em HTML, como no exemplo a seguir:

```
<style type="text/css" title="título" media="all">
```

Propriedades CSS

```
</style>
```

Nossa tag contém atributos interessantes. Primeiramente o atributo **type**, que com o valor acima informa para o navegador que estamos utilizando um arquivo de texto para identificar nossos estilos. Temos também o atributo **title**, que identifica nossa folha de estilo, e o atributo **media**, que é utilizado para selecionar o tipo de mídia à qual o estilo será aplicado. Podemos utilizar o valor **screen**, por exemplo, para quando desejamos que essa folha de estilo seja aplicada em computadores. Temos também o tipo **TV**, que indica que essa folha de estilo só vai ser interpretada em TVs, o que possibilita utilizarmos diversas folhas de estilo para um mesmo documento. Legal, não é?

Adicionamos estilos às nossas tags HTML da seguinte maneira:

```
body {  
    background-color: red  
}
```

Tag Body

Podemos verificar no exemplo que devemos passar primeiramente a tag que desejamos estilizar. No exemplo anterior utilizamos a tag **body**. Isso significa que tudo que estiver dentro da tag **body** será estilizado, sendo seguida pela abertura do bloco com o carácter chave ({). Dentro desse bloco teremos as propriedades que vamos estilizar. No exemplo abaixo a propriedade **background-color** tem como função alterar a cor do plano de fundo da nossa página para vermelho, seguido pelo fechamento do nosso bloco com o carácter chave (}).

Adicionamos estilos às nossas tags HTML da seguinte maneira:

```
body{  
    background-color: red  
}
```

Adicionando estilo para classe específica

Podemos também adicionar o estilo a uma classe específica, alterando apenas os elementos com essa classe definida. Para isso, colocamos um ponto antes do nome da classe, como no exemplo a seguir:

```
<div class="classe-teste"> <!-- Exemplo de tag com classe -->  
<style type="text/css" title="titulo" media="all">  
.classe-teste {  
    margin: 10px;  
}  
</style>
```

No exemplo acima, alteramos através da propriedade **margin** a margem de todos os elementos dentro da **div** com a classe "classe-teste", para 10 pixels.

Adicionando estilo por meio do ID do elemento

Para fazer esse tipo de adição utilizamos o carácter cerquilha (#), como no exemplo a seguir, onde vamos alterar todo o texto que estiver dentro da nossa **div** para o tamanho pequeno, através da propriedade **font-size**, e a fonte para Arial, através da propriedade **font-family**.

```
<div id="id-teste"> <!-- Exemplo de tag com identificador -->  
<style type="text/css" title="titulo" media="all">  
#id-teste {  
    font-size: small;  
    font-family: Arial  
}  
</style>
```

Podemos verificar no exemplo que é possível adicionar diversas propriedades de estilo ao nosso elemento, sempre lembrando do ponto e vírgula (;) separando cada uma delas. Porém, a criação de páginas web não é simplesmente encher a tela de tags e instruções. Existe uma organização que mantém padrões de desenvolvimento de software, a W3C, abreviação para World Wide Web Consortium, a qual vamos conhecer na próxima etapa.

World Wide Web Consortium – W3C

O W3C é uma organização da qual fazem parte, além de uma equipe em tempo integral, centenas de empresas parceiras e também do público em geral, visando manter padrões de desenvolvimento web, com o objetivo de compartilhar com os desenvolvedores as melhores práticas na criação de

sites e sistemas, garantindo aos usuários acessibilidade e compatibilidade. Estes padrões são criados considerando aspectos de acessibilidade, privacidade, segurança e internacionalização. A ideia dessa organização é manter uma web para todos, com os padrões de acessibilidade e para todas as coisas, sendo acessível de qualquer dispositivo.

Um exemplo de padrão é a utilização semântica da linguagem. Em geral, semântica significa o estudo do significado. Dentro do nosso contexto desenvolver, uma página web com a semântica correta é utilizar e organizar as tags corretamente, como por exemplo tag de título para título, tag de parágrafo para parágrafo, etc. Afinal, interpretar uma página que utiliza uma tag de título para o corpo do texto ou uma tag de lista para o título fica mais complicado, não é mesmo?

Não pare por aqui! Saiba mais...

Para mais informações sobre a W3C e os padrões de desenvolvimento web, acesse: <https://www.w3c.br/>

Vamos relembrar algumas das principais tags HTML e para que são usadas na atividade a seguir.

[MOD 1] Atividade: Tags HTML

Como vimos anteriormente, as tags HTML são utilizadas durante a criação de websites para organizar o conteúdo da maneira que desejarmos. Temos disponíveis para utilização centenas de tags HTML para organizar nossa página web da maneira que desejarmos. De acordo com o que aprendemos anteriormente, relacione as colunas abaixo identificando a tag HTML, levando em consideração a semântica correta e seu uso recomendado.

- A. () Tag de título
- B. () Tag para lista não ordenada.
- C. () Tag para parágrafo.
- D. () Tag para divisão do conteúdo.
- E. () Tag para links.

1. <p>
2. <h1>
3.
4. <a>
5. <div>

[MÍDIA 2] PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

A criatividade é um dos pontos mais importantes na criação de um website. Afinal, dentre tanta informação que temos disponível para acesso, é importante nos destacarmos da concorrência, para que assim atinjamos o objetivo da nossa página de currículo online. Vamos lá!

Agora, mãos à obra! Acesse a mídia clicando em **[Mídia 2] Planejamento e desenvolvimento do projeto**.

Ao criar uma página, sempre tenha em mente que se o usuário gostar da sua página, se impressionar e se sentir confortável durante a navegação, poderá ser um diferencial e um bom motivo para que ele acesse seu site novamente, indicando a amigos e interessados no assunto, além de passar credibilidade da informação que estamos disponibilizando.

Então vamos lá, elaborar um projeto diferenciado!

Projetando o Website

Além da criatividade, outro passo muito importante é seguir no nosso projeto os padrões de usabilidade e acessibilidade descritos pela W3C. Esses padrões auxiliam a manter nosso site confortável para o usuário, evitam desconfortos visuais e ergonômicos para quem está utilizando o site e foram pensados visando manter a qualidade geral na web, por isso são tão importantes.

Além dos padrões da W3C, temos que nos atentar durante a criação de algum projeto com o sigilo no tratamento dos dados dos usuários e respeito ao direito de propriedade intelectual das informações que usamos e manipulamos. Essas são regras exigidas por lei e devem ser seguidas para evitar possíveis problemas no futuro. A lei de sigilo e proteção de dados descreve algumas responsabilidades para quem está lidando com informações dos usuários. Dentre elas está garantir que os dados acessados estejam sob o consentimento do usuário e proíbe que informações pessoais sensíveis sejam compartilhadas sem esse consentimento. Já o respeito na propriedade intelectual exige que ao utilizarmos imagens, marcas, desenhos, livros, citações, etc., adicionemos também a referência e o crédito devido ao autor do conteúdo, garantindo assim a qualidade e a credibilidade das informações do nosso leiaute.

Antes de projetar um website é muito importante planejar as ações que serão tomadas. Gastar um tempo com o planejamento evita que você gaste muito mais tempo no futuro pensando na hora a página que deseja criar, além de realizar correções por ter criado a página de maneira diferente do esperado, podendo não atingir todos os seus objetivos com o website. O planejamento do projeto tem como principal objetivo garantir que ele seja finalizado com sucesso, contendo o mínimo de problemas possíveis e com uma alta satisfação do cliente. As etapas para o desenvolvimento de aplicações variam de empresa para empresa, de acordo com a qual mais se adapta à equipe e ao projeto. Durante o nosso curso vamos utilizar as etapas a seguir para planejar e desenvolver nossa página. Vamos entender no decorrer do curso a importância de cada uma delas.



1. Planejamento

O foco nessa etapa é reunir o maior número de informações relevantes possível, decidir o que será feito e como será feito. Analisamos o mercado, o público-alvo, a concorrência e um diferencial para o nosso site. Tem como objetivo definir as telas e funcionalidades que serão desenvolvidas.



2. Design

O design é a etapa em que planejamos a parte visual do nosso site. Realizamos o estudo de posição dos itens na nossa página, verificamos os itens que devem ficar destacados, quais serão os botões, as cores, etc., e documentamos essas decisões em forma de protótipos.



3. Implementação

A implementação é a criação da página em si, levando em conta o planejamento e o design feito anteriormente, convertendo a imagem do protótipo em linguagem HTML e CSS.



4. Inclusão de Conteúdo

Após finalizarmos a criação do website, vamos adicionar o conteúdo necessário para nossa página, as informações, textos, imagens, etc.



5. Validação

Antes de entregar nosso projeto, devemos testar e validar se a página criada está de acordo com o planejamento e o design. É nessa etapa que vamos validar links, botões ou até mesmo algum possível bug, evitando que o problema chegue até o cliente.

Planejamento

Para o nosso aprendizado, durante o curso iremos desenvolver uma página de currículo, contendo informações pessoais importantes e relevantes para o mercado de trabalho, pensando sempre no cliente, ou seja, quem irá acessar o seu site. E o que um currículo deve conter?

Existem diversos modelos e ideias diferentes na hora de montar um currículo. O nosso vai exibir em destaque o nome, visando chamar a atenção de quem acessar o site. Após o nome, devem aparecer também em destaque as informações pessoais. São elas: nacionalidade, estado civil, idade, data de nascimento, dados para contato e endereço. A página deve exibir uma foto, links de acesso às redes sociais relevantes e uma breve descrição do perfil do candidato. Após os dados pessoais a página deve conter informações profissionais. São eles: experiências anteriores, local, cargo, data e principais atividades, seguidos dos dados sobre a formação, instituição e data de formação e os cursos já realizados. É desejável que o site tenha um menu de acesso rápido.

Verificamos nessa etapa todas as informações relevantes para o desenvolvimento. Entre elas temos os dados obrigatórios e os desejáveis. Isso significa que temos um caminho já definido para o nosso desenvolvimento, facilitando e muito a implementação da nossa ideia.

Design

Para a criação do nosso design, vamos entender um pouco melhor sobre a criação de protótipos, que podem ser entendidos como o modelo inicial, que será utilizado no decorrer do processo. Os protótipos não são exclusivos do desenvolvimento de sistemas. Eles são usados em todas as áreas, como na criação de carros, por exemplo, ou na criação de máquinas e equipamentos, em que são

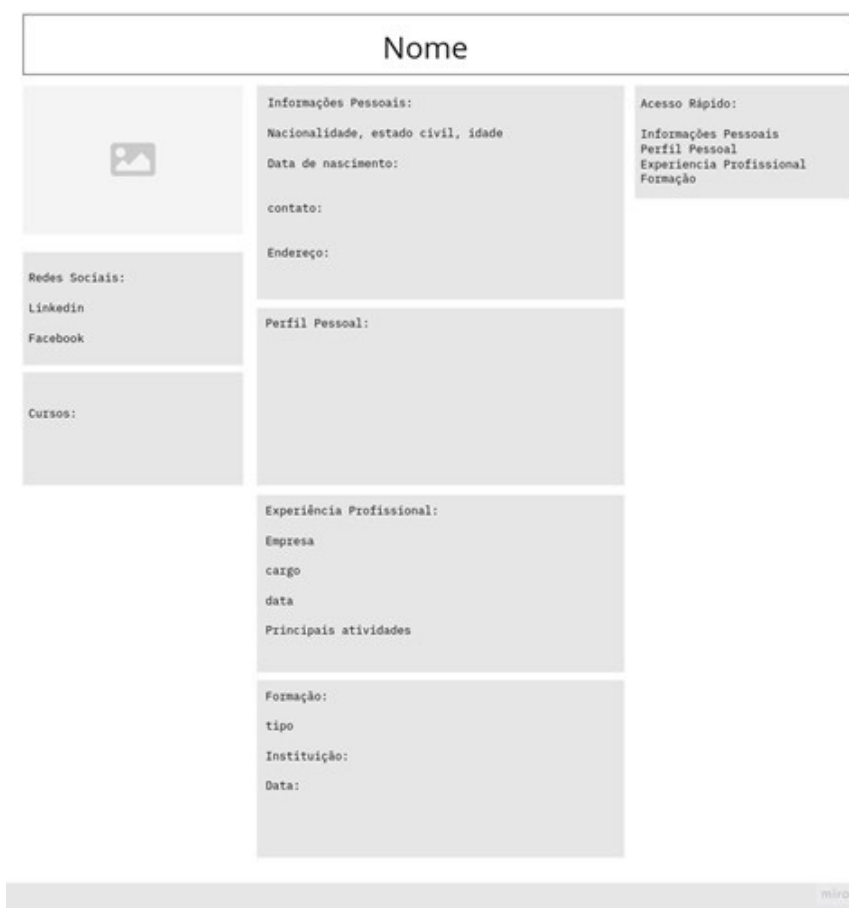
utilizadas impressoras 3D.

Para a prototipagem de páginas web, existem os wireframes, que são protótipos primitivos da página a ser desenvolvida. Podem ser entendidos também como um rascunho da página e são usados para guiar o desenvolvimento. Eles não contêm cores e estilos, pois o foco é apenas na organização. Para entendermos melhor, podemos dizer que ele é o esqueleto do site. Utilizamos os wireframes para visualizar como será nosso projeto final. É responsável por demonstrar para o desenvolvedor de que maneira a página deve ser dividida, onde ficarão as informações, botões, imagens, etc.

Introdução

A seguir está o wireframe criado para base da nossa página de currículo, através do site miro, um site gratuito para a criação de wireframes, mapas mentais, fluxos de trabalho e muitas outras funcionalidades, disponível em: <https://miro.com/>

Wireframe



Agora a ideia da nossa página ficou muito mais clara, não é mesmo?

Através do wireframe o desenvolvedor sabe o que tem que fazer e como espera que o resultado final seja. A partir desse modelo podemos seguir para o próximo passo, a implementação da nossa página web. Outro detalhe importante para a visualização da nossa página são as imagens que vamos adicionar. Antes de adicionar alguma imagem ao nosso site vamos conhecer dois tipos utilizados nas páginas web, os bitmaps e as imagens vetoriais.



Bitmaps

Bitmaps são imagens formadas por pixels. Um pixel nada mais é do que um ponto da tela em que a página está sendo exibida. Uma tela é formada por milhares de pixels, e através destes o bitmap cria uma espécie de mapa, formando a imagem e indicando a cor que cada pixel vai assumir. Esse é o tipo mais básico. Podemos reconhecer uma imagem bitmap ao darmos um zoom. Nela podemos verificar a marcação de vários quadrados, que são os pixels, que sofrem distorções conforme aproximamos.

Imagens vetoriais

As imagens vetoriais não usam um mapa de pixels para se localizar. Elas são criadas com base em cálculos matemáticos executados pelo computador, e tem como diferença que ao aumentarmos a imagem ela não se distorce como o bitmap. Se criarmos um retângulo vetorial, por exemplo, e aumentarmos seu tamanho em mil vezes, o retângulo continuará o mesmo, com a mesma definição e qualidade.

Implementação – Tutorial

Como vimos anteriormente, as instruções HTML e CSS podem ser escritas em qualquer arquivo de texto, até mesmo nos mais simples. Para a criação da nossa página vou utilizar o editor de texto Sublime, por ser um editor simples e ao mesmo tempo com várias funcionalidades que auxiliam o desenvolvedor, como cores diferentes para as tags, atributos, propriedades, etc., e que também disponibiliza a opção de completar automaticamente as instruções enquanto as escrevemos.

Agora, assista aos 2 vídeos a seguir: primeiro com o passo a passo da instalação do Sublime e o segundo um tutorial sobre as divisões e estrutura da sua página.

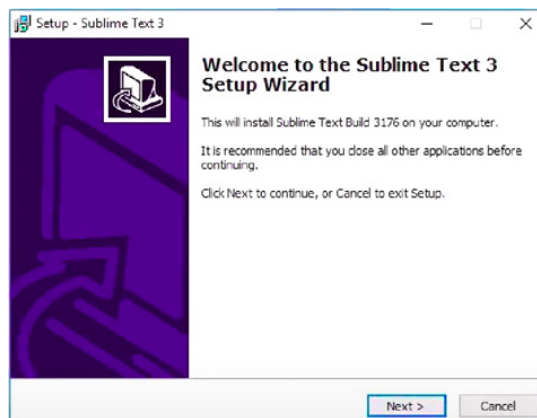
Caso não consiga abrir os vídeos, fique tranquilo. Acesse a transcrição deles em PDF, a seguir.

Instalação do Sublime mais divisões e estruturas

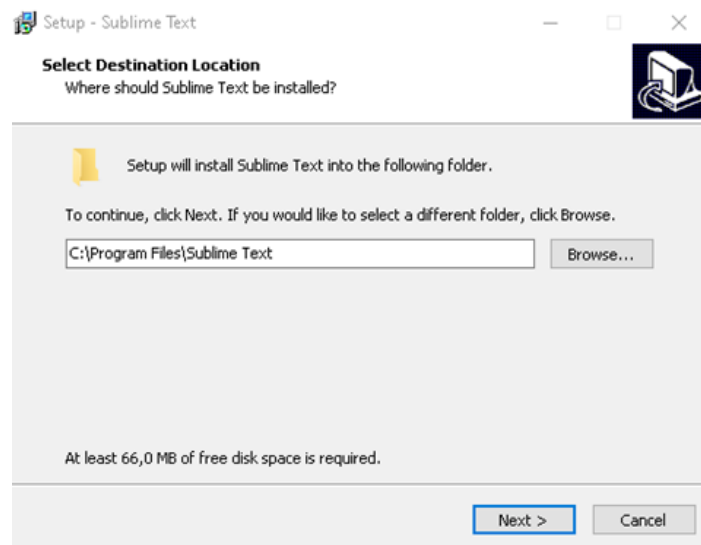
Instalando o Sublime

O primeiro passo é acessar o site oficial do Sublime para baixar a versão de acordo com o seu sistema operacional. O utilizado nesse curso é o sistema operacional Windows.

Para o próximo passo, execute o arquivo baixado e confirme a instalação através do botão Next.



Após essa etapa, será exibida a tela para escolha do local onde o Sublime será instalado. Escolha o local de sua preferência e selecione a opção Next para prosseguir com a instalação.



Agora clique em Install e depois em Finish para confirmar a instalação.

Introdução

Além do Sublime existem diversas ferramentas para edição de texto com funcionalidades muito úteis para a criação de leiautes, como o Visual Studio Code, Atom e o Notepad ++, por exemplo.

- Após abrir o editor de texto de sua preferência, crie o arquivo com o nome curriculo.html. Lembre-se que para que o navegador interprete que esse é um arquivo HTML devemos salvá-lo com a extensão .html.

Agora que criamos o arquivo vamos adicionar a estrutura básica de uma página HTML, a qual vimos anteriormente.

Uma dica muito útil é que, no Sublime, ao digitar HTML e utilizar o atalho ctrl + espaço, o editor completa automaticamente a estrutura básica do HTML, facilitando a criação da nossa página.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title></title>
</head>
<body>

</body>
</html>
```

Iniciando as divisões e estruturas

Passo 1

O próximo passo é realizar as divisões e iniciar a estrutura da nossa página, seguindo o modelo proposto no wireframe. Podemos verificar que o site terá um header, ou cabeçalho, que irá conter o nome do candidato. Para isso, vamos:

- criar a divisão, utilizando a tag <div>, com a classe header, para identificarmos a nossa divisão.

Outro detalhe importante é que nas instruções da nossa página vamos seguir o padrão de desenvolvimento web e manter as instruções em inglês. Abaixo do header teremos o conteúdo da página, que deve ser dividido em três colunas. Para isso:

- criar uma divisão com a classe content, que significa conteúdo em inglês. Dentro do conteúdo do site vamos ter as colunas, também divididas e com a classe para identificá-las, ficando com a left-column, middle-column e right-column; traduzindo para o português, vamos ter a coluna esquerda, a coluna do meio e a coluna da direita, respectivamente.

Abaixo do conteúdo teremos o rodapé da página. Nossa classe fica com o nome de footer, para seguir o padrão da língua inglesa.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title></title>
</head>
<body>
  <div class="header">

  </div>

  <div class="content">
    <div class="left-column">

    </div>

    <div class="middle-column">

    </div>

    <div class="right-column">

    </div>
  </div>

  <div class="footer">

  </div>
</body>
</html>
```

Passo 2

O próximo passo é organizar o nosso cabeçalho. Para isso:

- colocar a tag de título <h2> (adicione o título ao seu arquivo e lembre-se sempre de fechar a tag para que ela seja interpretada corretamente).

```
<div class="header">
  <h2>Nome do Candidato</h2>
</div>
```

Passo 3

Dentro da divisão de conteúdo, vamos começar pela coluna esquerda da nossa página, a qual identificamos utilizando a classe left-column.

- Adicionar outra divisão dentro da coluna esquerda, com a classe “card”. Essa classe vamos estilizar mais para frente utilizando o CSS, para que ela pareça um card, ou cartão, no nosso site.

Primeiro card

O primeiro card vai conter a sua foto. Para adicioná-la à página:

- salve a foto que deseja utilizar na mesma pasta em que está seu arquivo, para facilitar o acesso. Como vimos anteriormente, para adicionar uma imagem utilizamos a tag e adicionamos o atributo src com o nome do arquivo como valor;
- adicione o formato da imagem que escolhemos e também o id profissional-picture para identificarmos essa imagem no CSS. No meu caso, o nome da imagem é foto-profissional, e o formato .jpg, conforme imagem a seguir. Nessa parte você irá colocar o nome da foto que você escolheu;
- atualizar sua página conforme os exemplos.

```
<div class="card">
  
</div>
```

Se ocorrer algum erro na hora do navegador interpretar a foto, verifique se o nome está correto, se o formato da imagem foi informado e se está correto ou se a tag foi aberta e fechada corretamente. Quando algum erro ocorrer o navegador vai exibir uma imagem padrão que informa que a imagem não foi encontrada.

Passo 4

Vamos colocar um nome incorreto na nossa imagem e visualizar o erro no navegador.

Nome do Candidato



Outro erro muito comum na criação de leiautes é utilizar a tag de maneira errada. Se isso acontecer o navegador vai exibir a tag como se fosse um texto em vez de interpretá-la como uma instrução. Para exemplo, vou fechar a tag div com a barra invertida, ou seja, um caractere diferente do esperado. Vendo essa informação no navegador sabemos que tem algo de errado com a tag criada, e deve ser corrigido para a exibição correta do nosso leiaute.

Nome do Candidato

<div>

Segundo card

O segundo card deve conter as suas redes sociais, iniciando com o título do card, seguido pelas redes sociais. Para exemplo, vamos adicionar o Facebook e o LinkedIn, mas fique à vontade para adicionar a rede que você acha mais relevante para o seu currículo. Para isso, vamos adicionar um link, separado como um parágrafo com a tag <p>, e através da tag <a> disponibilizamos acesso ao link para o usuário, que tem como destino o caminho da sua rede social. Nesse exemplo, vamos colocar a página inicial de cada rede e vamos adicionar mais um detalhe, o atributo target com o valor blank. Isso passa para o navegador que esse link deve ser aberto em uma nova aba, para evitar que o usuário saia da página que estamos criando. Dentro da tag de link vamos colocar a logo da rede social, da mesma maneira que adicionamos a sua foto anteriormente. Baixe em seu computador a logo da rede social que deseja utilizar. Elas ficam disponíveis para download em mecanismos de busca como o Google ou no site da rede que deseja adicionar. Nessa imagem vamos adicionar a classe social. Fique atento para adicionar o nome correto da imagem que baixou no atributo src, juntamente com o formato.

```
<div class="card">
  <h3>Redes Sociais</h3>
  <p>
    <a href="https://www.linkedin.com/" target="blank">
      
      LinkedIn
    </a>
  </p>
  <a href="https://www.facebook.com/" target="blank">
    
    Facebook
  </a>
</div>
```

Terceiro card

Para o terceiro e último card da nossa coluna da esquerda, adicione o título e outra divisão, com a classe card-content, ou seja, conteúdo do card, para facilitar a estilização com o CSS. Dentro desse conteúdo vamos ter a lista de cursos que você realizou. Vamos adicionar alguns exemplos.

```

<div class="card">
  <h2>Cursos:</h2>
  <div class="card-content">
    <ul>
      <li>HTML e CSS - SENAC</li>
      <li>Curso 2 - Instituição 2</li>
      <li>Curso 3 - Instituição 3</li>
    </ul>
  </div>
</div>

```

Pronto, finalizamos as marcações básicas da nossa coluna esquerda. Podemos verificar como está ficando abrindo esse arquivo através do navegador. Para isso, clique com o botão direito no arquivo e selecione a opção abrir com, e depois o navegador de sua preferência. O site deve estar parecido com o da imagem a seguir.



Pode ficar calmo, aluno, ainda não está muito bonito, né?

Vamos resolver isso mais para frente utilizando o CSS.

Passo 5

- Iniciar a marcação da nossa coluna do meio, que no seu primeiro card vai ter o título “Informações pessoais:”, com o id “personal-information”, seguidos dos dados pessoais, separados sempre por parágrafo. Atualize seu arquivo conforme imagem a seguir.

```

<div class="card">
  <h2 id="personal-information">Informações Pessoais:</h2>
  <p>Brasileiro, solteiro, 25 anos.</p>
  <h4>Data de Nascimento:</h4>
  <p>01/01/1996</p>
  <h4>Contato:</h4>
  <p>(45) 99999-9999 | (44) 99999-9999</p>
  <p>teste@email.com.br</p>
  <h4>Endereço:</h4>
  <p>Avenida, N° 1234</p>
  <p>Bairro</p>
  <p>Curitiba-PR</p>
</div>

```

- Conforme a análise, devemos criar também um card para a descrição do perfil pessoal.

```

<div class="card">
  <h2 id="personal-profile">Perfil Pessoal:</h2>
  <p>
    Aqui fica a descrição do seu perfil pessoal!
  </p>
</div>

```

Passo 6

Após o perfil pessoal, vamos ter as experiências profissionais.

- Adicione os dados conforme solicitado na análise. Com uma pequena diferença do que fizemos até agora, vamos adicionar a tag `` para deixar o texto em negrito, visando destacar essa parte, e uma linha horizontal com a tag `<hr />` para separar cada experiência.

```

<div class="card">
  <h2 id="professional-experience">Experiência Profissional:</h2>
  <div class="card-content">
    <h4>Empresa 123</h4>
    <p>
      <strong>Cargo:</strong>
      Teste
    </p>
    <p>
      <strong>Data:</strong>
      01/01/2020 a 01/01/2021
    </p>
    <p>
      <strong>Principais Atividades:</strong>
      Desenvolvedor de sistemas.
    </p>
  </div>
  <hr />
  <div class="card-content">
    <h4>Empresa ABC</h4>
    <p>
      <strong>Cargo:</strong>
      Teste
    </p>
    <p>
      <strong>Data:</strong>
      01/01/2020 a 01/01/2021
    </p>
    <p>
      <strong>Principais Atividades:</strong>
      Desenvolvedor de sistemas.
    </p>
  </div>
</div>

```

Passo 7

O último card da coluna do meio é o de formação, que fica muito semelhante ao de experiências profissionais.

- Atualize seu arquivo conforme a imagem a seguir.

```
<div class="card">
  <h2 id="formation">Formação:</h2>
  <div class="card-content">
    <h4>Ensino médio Completo</h4>
    <p>
      <strong>Instituição:</strong>
      Instituição de teste
    </p>
    <p>
      <strong>Data:</strong>
      01/01/2020 a 01/01/2021
    </p>
  </div>
</div>
<hr />
<div class="card-content">
  <h4>Sistemas de informação</h4>
  <p>
    <strong>Instituição:</strong>
    Instituição de teste
  </p>
  <p>
    <strong>Data:</strong>
    01/01/2017 a 01/01/2021
  </p>
</div>
</div>
```

A coluna da direita terá apenas um card, para acesso rápido. Esse card vai possibilitar que o usuário clique sobre o tópico que deseja e o site vá para esse tópico. Ou seja, nessa parte vamos adicionar um link, e como sua referência são os próprios itens da página, fazemos isso utilizando o caractere cerquilha (#), seguido pelo id do nosso tópico.

```
<div class="right-column">
  <div class="card">
    <h3>Acesso Rápido:</h3>
    <ul>
      <a href="#personal-information"><li>Informações Pessoais</li></a>
      <a href="#personal-profile"><li>Perfil Pessoal</li></a>
      <a href="#professional-experience"><li>Experiência Profissional</li></a>
      <a href="#formation"><li>Formação</li></a>
    </ul>
  </div>
</div>
```

[MOD 1] Atividade: Estrutura HTML do leiaute

É muito importante seguirmos a estrutura correta do nosso leiaute criado com o HTML. Utilizando a estrutura correta garantimos que o nosso site será interpretado corretamente, não só pelo navegador, mas também por leitores de tela, por exemplo, que possibilitam acesso ao site a usuários com deficiência visual, melhorando a acessibilidade da página. Outro benefício é ajudar os sites de busca a identificar nosso conteúdo e classificá-lo melhor, aumentando os acessos a nossa página. De acordo com o conhecimento obtido até aqui no nosso curso, assinale a alternativa correta, relacionando com a estrutura correta para a criação de páginas HTML.

A. A tag <html> deve ser aberta no início do arquivo, logo abaixo da tag doctype, e tem como objetivo informar ao navegador que o que está dentro dessa tag deve ser interpretado como instruções HTML.

B. A tag <head> fica localizada após a tag <body>, e deve conter informações que devem aparecer no leiaute da página.

C. A tag <body> contém o conteúdo do nosso site; dentro dela adicionamos as informações que irão aparecer para os usuários.

D. A tag <body> é a primeira tag que devemos adicionar à página; ela fica localizada logo abaixo da tag <html>.

E. Informamos ao navegador o título da página através da tag <title>; ela deve estar presente dentro da tag <head>.

[MOD 1] Atividade: Identificar e corrigir erros de codificação 1

Identificar e corrigir erros durante a codificação faz parte do dia a dia de todo desenvolvedor de sistemas e de criadores de leiautes para páginas web. Essa é uma das habilidades mais importantes para o sucesso nessas áreas de atuação, pois sem ela, após qualquer erro, nosso projeto não iria para frente. Na imagem a seguir podemos visualizar ao lado esquerdo as instruções em HTML contendo nosso header, e ao lado direito o resultado interpretado pelo navegador, que está exibindo alguns caracteres indesejados. Assinale a alternativa que julga descrever o problema na criação das marcações e que resultaram na exibição incorreta das informações.

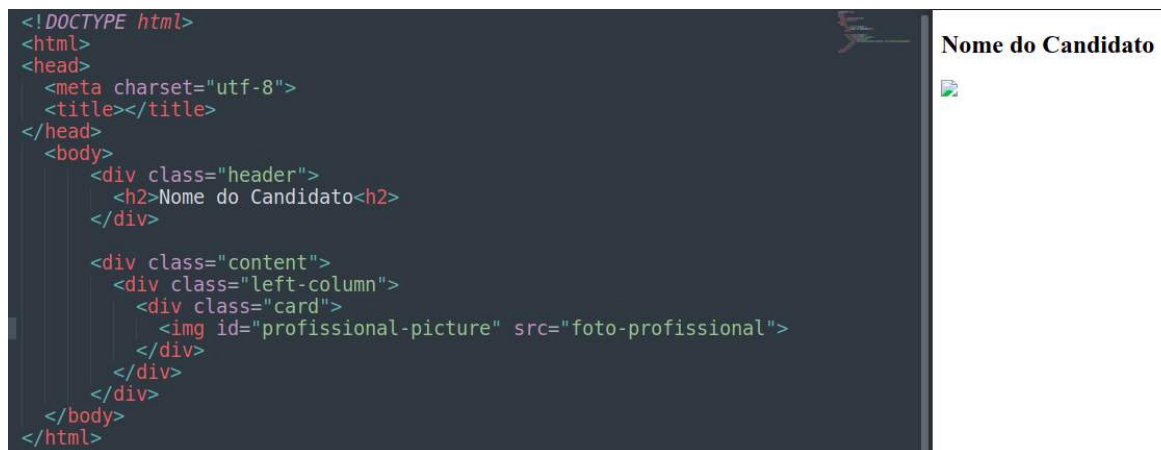
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title></title>
</head>
<body>
  <div class="header">
    <h2>Nome do Candidato h2>
  </div>
</body>
</html>
```

Nome do Candidato h2>

- A. O conteúdo deveria ficar dentro da tag <head> e por isso não foi exibido corretamente.
- B. Ao fechar a tag <h2>, faltou o caractere menor (<) para iniciar o fechamento da tag, fazendo com que o navegador interprete como um texto normal.
- C. A tag <h2> está dentro de uma tag <div>, e por isso não é exibida corretamente.
- D. Falta informarmos o atributo id para identificar o nosso texto, e sem essa identificação ele não é exibido corretamente.

[MOD 1] Atividade: Identificar e corrigir erros de codificação 2

Seguindo com a análise e resolução de problemas, temos à esquerda da imagem a seguir as instruções HTML para exibir uma imagem no nosso site. De acordo com a imagem a seguir, assinale a alternativa que acredita descrever melhor o motivo da exibição incorreta da imagem no nosso leiaute.



- A. O conteúdo deveria ficar dentro da tag <head> e por isso não foi exibido corretamente.
- B. Provavelmente o nome da imagem está incorreto, sendo que quando esse erro é exibido significa que o navegador não encontrou a imagem.
- C. Falta fechar a tag , o que faz com que o navegador não interprete corretamente essa tag.
- D. Falta informarmos o formato da imagem, que pode ser .jpn, .jpeg, .png, entre outros.

Passo final

O passo final na criação das marcações do nosso leiaute é testar e validar a nossa página. Nesse passo é importante que você acesse sua página e valide as informações que já inserimos, se estão todas de acordo com a análise, se as fotos estão aparecendo corretamente e se os links estão apontando para os locais corretos, da mesma maneira que o nosso menu de acesso rápido, que ao ser selecionado deve ir até o tópico escolhido. É muito importante testarmos e validarmos nossa página web, pois através desses processos garantimos a qualidade do nosso leiaute e evitamos que erros apareçam para o usuário final. Corrigindo possíveis erros durante o desenvolvimento evita-

mos que o usuário perca a credibilidade no nosso site quando erros aparecem na tela e/ou algum processo que queremos realizar não é executado corretamente. Eu mesmo já tive uma experiência em um site de venda, em que erros durante a navegação me fizeram desistir da compra. Você já passou por alguma situação parecida? Se sim, entende o que eu estou falando e tenho certeza que vai testar várias vezes seu leiaute. E mesmo se não passou conseguimos imaginar o quanto perder a venda e um cliente por erros de leiaute é um problema muito sério, não é mesmo?

[MÍDIA QUIZ] ETAPAS DO PROJETO

Agora que finalizamos as marcações solicitadas na análise e na prototipação, realize a atividade a seguir antes de partirmos para o próximo módulo.

Clique no link [\[Mídia Quiz\] Etapas do projeto](#).

[MOD 1] Atividade: Etapas do Projeto

Verificamos até aqui que planejar e executar cada etapa do projeto é muito importante para que o resultado final seja satisfatório. De acordo com o modelo que aprendemos anteriormente, ordene os itens a seguir de maneira que representem o fluxo correto do desenvolvimento do projeto.

1. PLANEJAMENTO
2. DESIGN
3. IMPLEMENTAÇÃO
4. INCLUSÃO DE CONTEÚDO
5. VALIDAÇÃO

() Planejar e definir o objetivo do site e o que a nossa página de currículo deve conter.

() Definir a estrutura da página e onde ficarão organizadas as informações do currículo.

() Desenvolver o leiaute da página web através das linguagens HTML e CSS.

() Adicionar o conteúdo do site já desenvolvido, as informações pessoais e profissionais e imagens.

() Validar as informações apresentadas no site já finalizado, verificar se as imagens estão corretas, se os links estão levando ao lugar certo e se as imagens estão sendo exibidas corretamente.

[SÍNTESE] MÓDULO I

Parabéns, aluno, finalizamos a marcação da nossa página. Fizemos um grande progresso até aqui! Fique à vontade para verificar nosso site e testar o acesso rápido. No módulo seguinte iremos estilizar nossa página utilizando o CSS, alterando as cores, fazendo com que cada coluna fique no seu lugar correto, estilizando os cards e corrigindo o tamanho das imagens.

Para fecharmos ouça o áudio que gravei para você, só dar um PLAY!

Sintetizando os aprendizados

Parabéns, aluno (a)!

Finalizamos a marcação da nossa página e fizemos um grande progresso até aqui. Fique à vontade para verificar nosso site e testar o acesso rápido. No módulo seguinte, iremos estilizar nossa página utilizando o CSS, alterando as cores, fazendo com que cada coluna fique no seu lugar correto, estilizando os cards e corrigindo o tamanho das imagens.

Eu estou ansioso, e você?

Vamos lá! Até o próximo módulo!

MÓDULO 2

[INTRODUÇÃO] ANÁLISE DA USABILIDADE E ACESSIBILIDADE DA INTERFACE DO WEBSITE

Olá, aluno (a), seja bem-vindo ao segundo módulo do curso de HTML e CSS. Até agora vimos como planejar e criar o leiaute de um website, por meio de protótipos utilizando um wireframe e levando em consideração os padrões da W3C. Aprendemos também o que são e como funcionam as linguagens de marcação e estilo HTML e CSS, bem como suas principais características. Agora, vamos tratar sobre “Análise da usabilidade e acessibilidade da interface do website”. Para isso, o módulo está organizado da seguinte forma:

[Mídia Quiz] *Aquecimento*

[Mídia 1] *Características funcionais do projeto*

[Mídia 2] *Melhorias e responsividade*

Vamos lá, bons estudos!

[MÍDIA QUIZ] AQUECIMENTO

Essa é uma mídia Quiz de aquecimento, para você fazer uma breve retomada no que vimos no módulo anterior e estar pronto para agregar novos conhecimentos nesse novo módulo, antes de iniciar a estilização do site.

Realize a atividade a seguir para “aquecer” seus conhecimentos prévios construídos anteriormente e estar a todo vapor para as próximas abordagens.

Boa atividade!

[MOD 2] Atividade: Conceitos do projeto de elaboração de leiaute

Dentro do processo de desenvolvimento de softwares e criação de leiautes, temos diversos conceitos importantes que auxiliam o desenvolvimento do nosso projeto e garantem a qualidade do sistema e das informações transmitidas. De acordo com o que estudamos no módulo anterior, assinale cada alternativa em Certo e Errado.

1. () Wireframes são prints de tela do projeto já pronto. São utilizados como base para o leiaute e devem conter alto nível de detalhes.
2. () As imagens vetoriais são formadas por pixels, onde cada pixel recebe uma cor específica, formando assim a imagem completa.
3. () As etapas do desenvolvimento de um projeto são planejamento, design, implementação, inclusão de conteúdo e validação.
4. () O foco da etapa de planejamento de um software é reunir o maior número de informações relevantes possível, decidir o que será feito e como será feito, analisando o mercado, o público-alvo, a concorrência e um diferencial para o nosso site. Tem como objetivo também definir as telas e funcionalidades que serão desenvolvidas.
5. () O desenvolvimento de leiautes utilizando a semântica correta da linguagem é um exemplo de padrão de desenvolvimento da W3C e tem como objetivo principal facilitar a interpretação da página pelo navegador e demais ferramentas que precisam reconhecer o conteúdo.

[MÍDIA 1] CARACTERÍSTICAS FUNCIONAIS DO PROJETO

Bem-vindo (a) a nossa primeira mídia de conteúdo do módulo. Aqui começamos a tratar sobre “*Análise da usabilidade e acessibilidade da interface do website*”, entendendo como isso é importante e como podemos melhorar esses conceitos no nosso site.

Vamos lá!

Palavra do autor

Olá, aluno (a)!

Aqui quem fala é o Vitor Gabriel Martines Weinfortner, criador do curso. Seja bem-vindo ao segundo módulo do curso de HTML e CSS. Até agora vimos como planejar e criar o leiaute de um website, por meio de protótipos, utilizando um wireframe e levando em consideração os padrões da W3C. Aprendemos também o que são e como funcionam as linguagens de marcação e estilo HTML e CSS, bem como suas principais características. Neste módulo iremos analisar a usabilidade e a acessibilidade da interface do website, entender como isso é importante e como podemos melhorar esses conceitos no nosso site.

Espero que esteja gostando do curso até aqui. Bons estudos!

Usabilidade

A usabilidade de um website tem foco total no usuário e pode ser entendida como um termo que define o grau de facilidade com o qual ele consegue utilizar o site e realizar alguma tarefa dentro dele. Tem como objetivo desenvolver interfaces capazes de permitir uma navegação agradável, interativa, eficiente e eficaz. Para isso precisamos entender as necessidades do usuário, bem como tornar o site claro e objetivo, oferecendo uma experiência agradável e satisfatória, pensando em alguns detalhes como a acessibilidade, por exemplo. Se um usuário com daltonismo acessar o nosso site e este não estiver preparado para essa peculiaridade, ele não será útil para esse usuário, que consequentemente vai deixar de visitá-lo.

Segundo Sharp, Preece e Roger (2005), as metas da usabilidade do sistema são:

Eficácia – Se o sistema faz corretamente o que se espera dele.

Eficiência – O sistema deve facilitar que o usuário realize sua tarefa, afinal, se o usuário entrar no site para comprar um produto, mas não conseguir achar como finalizar a compra, o site não está sendo eficiente no seu objetivo.

Segurança – O site deve proteger os usuários de situações indesejadas. Garantir a segurança das informações deve ser a principal preocupação.

Utilidade – Se o site é útil para os usuários realizarem o que queriam ao acessá-lo.

Facilidade ao aprender – Refere-se à facilidade com que o usuário consegue aprender a usar o sistema.

Facilidade para memorizar – Refere-se à facilidade que o usuário tem em lembrar como se usa o sistema. Depois de aprender a usá-lo, o sistema deve ser simples o suficiente para que ele lembre como realizar determinados passos novamente.

Normal

Item	Height (approximate)	Pattern
Item 1	80	Brown checkered
Item 2	60	Orange diagonal stripes
Item 3	30	Green grid
Item 4	40	Red solid

Acessibilidade

A proposta da acessibilidade é dar autonomia para pessoas com necessidades especiais, garantindo que os websites funcionem bem com tecnologias que auxiliam esses usuários, como leitores de tela, ampliação de conteúdo e software de reconhecimento de fala para pessoas com deficiência visual, por exemplo. Outro exemplo pode ser um site desenvolvido possibilitando a utilização sem o mouse, tornando o sistema acessível não só para pessoas com deficiência visual, mas também com deficiência física.

Ergonomia

36

Ao desenvolver um sistema, devemos levar em consideração princípios ergonômicos, visando de forma integrada a saúde, a segurança e o bem-estar de todos. Esses princípios trazem confiança ao usuário e tornam a utilização do sistema muito mais agradável, fazendo com que ele continue utilizando seu site e até indique para os amigos.

Existem diversos critérios que qualificam a ergonomia de um site. Dentre os principais, podemos citar os seguintes:

Sobriedade

Trata da organização do site, tendo como principal objetivo reforçar a credibilidade do usuário. Podemos utilizar como exemplo os gifs. Como eles chamam muito a atenção, devem ser usados somente para mensagens importantes. Ao entrarmos em um site com muita informação e vários gifs temos uma imagem negativa, perdendo a segurança para continuar utilizando.

Estrutura

Podemos levar como exemplo os textos, que devem ser organizados em parágrafos e títulos de diferentes níveis, para facilitar a leitura. Anteriormente no nosso curso vimos que podemos utilizar as tags <h1> para títulos maiores e mais importantes e a <h6> para títulos menos importantes.

Facilidade

A principal regra de facilidade é a 'regra dos três cliques', bem conhecida na parte de planejamento do projeto. Essa regra diz que qualquer informação do nosso site deve poder ser acessada em no máximo três cliques.

Tempo de carregamento

Deve ser o mais curto possível, para evitar que o usuário perca a paciência e deixe de usar o nosso site.

Cores

Esse critério é muito importante. As cores devem formar um contraste no site para que a informação seja legível para pessoas que não distinguem as cores corretamente. Utilizar um fundo branco com a letra preta, por exemplo, facilita a leitura e evita que o usuário canse os olhos por ter que forçá-los.

[MOD 2] Atividade: Usabilidade e acessibilidade

É muito importante para o sucesso do nosso site que ele mantenha os padrões de usabilidade e acessibilidade. Isso garante que o usuário se mantenha na página por mais tempo, explore melhor as informações e até mesmo a indique para outras pessoas interessadas. Analise as alternativas a seguir e assinale a que julgar **INCORRETA**.

A. A acessibilidade de uma página está diretamente ligada ao desenvolvedor, garantindo a qualidade do código e facilidade ao acessar a página.

B. As cores são um dos itens mais importantes para auxiliar na usabilidade e acessibilidade de um site. Através delas podemos destacar informações importantes e dar sobriedade às menos importantes, para evitar o cansaço do usuário. São importantes também para garantir a acessibilidade para pessoas com diversidades como o daltonismo, por exemplo.

- C. As metas da usabilidade do sistema são a eficácia, eficiência, segurança, utilidade, facilidade para aprender e facilidade para memorizar.
- D. Ferramentas de leitura de tela são exemplos de ferramentas que auxiliam a acessibilidade do leiaute.
- E. A organização e sobriedade do site são muito importantes para manter a credibilidade da informação.

Estilização do Website

E aí aluno, o que está achando do curso até aqui? Não é tão simples assim desenvolver um leiaute, não é mesmo? Existem vários conceitos que devemos levar em consideração. Agora que aprendemos alguns dos principais, podemos seguir com a criação do nosso website de currículo.

Assista ao vídeo “Iniciando a estilização do leiaute” para aprender mais e arrasar na apresentação do seu website de currículo.

Caso não tenha conseguido abrir o vídeo ou queira ler o conteúdo do vídeo, veja o PDF “Iniciando a estilização do leiaute” abaixo.

Iniciando a estilização do leiaute

Vamos iniciar a estilização do nosso website adicionando a tag de estilo dentro da tag head do nosso HTML. Como dito anteriormente, a tag responsável por marcar que o trecho de código é uma folha de estilo é a `<style>`. Dentro da tag style podemos adicionar nossa folha de estilo e personalizar o nosso site. A primeira alteração que vamos fazer é no corpo da nossa página, referenciando a tag body. Vamos alterar todo o conteúdo que está dentro dessa tag, nesse caso, todo o corpo do site. A primeira propriedade que iremos utilizar é a `font-family`, com o valor `Arial`. Essa propriedade controla o tipo da letra do nosso site. Vamos alterar para uma fonte mais agradável para leitura, a `Arial`. Vamos alterar também o `padding` da nossa página. Como vimos no módulo anterior, o `padding` é uma das principais propriedades CSS. Ela controla o espaçamento entre o conteúdo e a sua borda. Podemos verificar a diferença ao atualizar nosso site. Por último, vamos alterar a cor de fundo do nosso site utilizando a propriedade `background-color`. Essa propriedade normalmente é utilizada com valores hexadecimais que representam as cores, mas também aceita as cores em inglês (como `red`, `blue`, `grey`, etc.).

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style type="text/css" title="style" media="all">
    body {
      font-family: Arial;
      padding: 20px;
      background-color: #f1f1f1;
    }
  </style>
  <title></title>
</head>
```

Com padding:

Cursos:

- HTML e CSS - SENAC
- Curso 2 - Instituição 2
- Curso 3 - Instituição 3

Informações Pessoais:

Brasileiro, solteiro, 25 anos.

Data de Nascimento:

01/01/1996

Contato:

(45) 99999-9999 | (44) 99999-9999
teste@email.com.br

Endereço:

Avenida, N° 1234
Bairro
Curitiba-PR

Sem padding:

Cursos:

- HTML e CSS - SENAC
- Curso 2 - Instituição 2
- Curso 3 - Instituição 3

Informações Pessoais:

Brasileiro, solteiro, 25 anos.

Data de Nascimento:

01/01/1996

Contato:

(45) 99999-9999 | (44) 99999-9999
teste@email.com.br

Endereço:

Avenida, N° 1234
Bairro
Curitiba-PR

A fim de organizar o nosso código, vamos separar as propriedades CSS em um novo arquivo, e importá-lo na nossa página. Para isso, crie um arquivo chamado `style.css` e salve na mesma pasta da nossa página, a fim de facilitar a importação. Após a criação do arquivo, vamos importá-lo através da tag `<link>`. Dentro da tag link adicionamos as propriedades `rel` com o valor `stylesheet`, que indica que é um arquivo de estilo. Informamos também o tipo do arquivo, que no caso é um texto contendo informações CSS. Por último, o caminho do arquivo de estilo. No nosso caso, o arquivo está na mesma pasta. Então, é só referenciar o nome e a extensão. Lembre-se de passar nossa instrução, que estiliza o body do arquivo `.html` para o arquivo `.css`.

```
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="style.css" />
  <title></title>
</head>
```

No próximo passo, vamos estilizar o header da nossa página. Nessa parte está o nome do candidato. Vamos adicionar ao header as propriedades **font-size**, para alterar o tamanho da fonte, e **text-align**, para o alinhamento do texto, e vamos utilizar uma maneira diferente para alterar a cor de fundo, **background** com o valor **white**, ou seja, o fundo do nosso header ficará branco e claro, além de um pouco de **padding** para organizar nosso conteúdo. Podemos verificar a diferença ao atualizar o layout.

```
body {
  font-family: Arial;
  padding: 20px;
  background-color: #f1f1f1;
}

/* Header com o titulo principal */
.header {
  padding: 30px;
  font-size: 40px;
  text-align: center;
  background: white;
}
```

Nome do Candidato

Nome do Candidato

Seguindo com nossa folha de estilo, vamos organizar nossa coluna da esquerda, começando pelas imagens das redes sociais. Anteriormente adicionamos à classe social as imagens das redes sociais. Através dessa classe vamos alterar o tamanho dessas imagens, utilizando o ponto seguido do nome da classe, onde alteramos a largura através da propriedade **width** e a altura através da propriedade **height** para 16 pixels. Não se esqueça de atualizar seu arquivo de estilo.

```
.social {
  width: 16px;
  height: 16px;
}
```

A coluna esquerda vai conter três propriedades. A primeira é o **float**, que indica que o elemento deve ser retirado do seu fluxo normal e “flutuado” para a direção que indicamos. Isso vai manter nossa coluna fixa à esquerda. Vamos usar também a propriedade **width**, só que dessa vez o tamanho não é fixo. Ele deve ocupar 25% da página. Por último, vamos colocar um padding que separa as colunas. Para isso, vamos utilizar a propriedade **padding-right**, que adiciona o padding desejado somente do lado direito do elemento.

```
/*Coluna esquerda */
.left-column {
  float: left;
  width: 25%;
  padding-right: 20px;
}
```

Seguindo com nosso menu lateral, vamos organizar a foto profissional. Anteriormente adicionamos o identificador único **id** com o valor **profissional-picture** à nossa foto. Para estilizá-la utilizamos o carácter cerquilha (#) seguido com o valor do indicador. Vamos alterar a cor do fundo para destacar nossa foto e fazer com que ela ocupe 100% da divisão em que ela se encontra, ou seja, da coluna esquerda.

```
#profissional-picture {  
  background-color: #aaa;  
  width: 100%;  
  padding: 20px;  
}
```

É isso aí! Finalizamos a coluna esquerda. O nosso site já está com uma cara bem diferente, não é? Vamos seguir alterando a coluna do meio. Assista ao próximo vídeo.

É a sua vez: treine, teste, arrisque sua estilização com base nas funções apresentadas no tutorial. Aproveite seu tempo de estudo explorando ao máximo as possibilidades. Após ter feito isso, vamos ao vídeo “Organização das colunas”.

Caso não tenha conseguido abrir o vídeo ou queira ler o conteúdo do vídeo, veja o PDF “Organização das colunas” abaixo.

Organização das Colunas

Agora que finalizamos nossa primeira coluna, vamos estilizar a coluna do meio, a qual identificamos com a classe **middle-column**. Para isso, vamos adicionar a propriedade **float** também com o valor **left**, o que vai fazer a coluna flutuar para a esquerda. Porém, como antes dela já existe uma coluna flutuando à esquerda, ela vai ser fixada após a primeira coluna. Assim conseguimos deixá-la no meio do nosso leiaute. Vamos alterar também o tamanho da nossa coluna para 50% do tamanho da página.

```
/* Coluna do meio */  
.middle-column {  
  float: left;  
  width: 50%;  
}
```

Mas as colunas ainda não estão bem divididas. Uma está se sobrepondo à outra. Para corrigir isso, vamos adicionar a todos os elementos da nossa página, através do identificador *****, a propriedade **box-sizing**, com o valor **border-box**. Isso indica para o navegador que o tamanho de cada coluna deve levar em conta também a borda e o padding, separando melhor nossas colunas. Não se esqueça de atualizar seu arquivo de estilo.

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




Nossas colunas já estão organizadas da maneira como queremos. Entretanto, vamos estilizar também a coluna direita, para fixá-la no seu lugar. Vamos adicionar ao arquivo de estilo as propriedades **float**, **width** e **padding-left** para a sua coluna direita.

```
/* Coluna da Direita */
.right-column {
  float: left;
  width: 25%;
  padding-left: 20px;
}
```

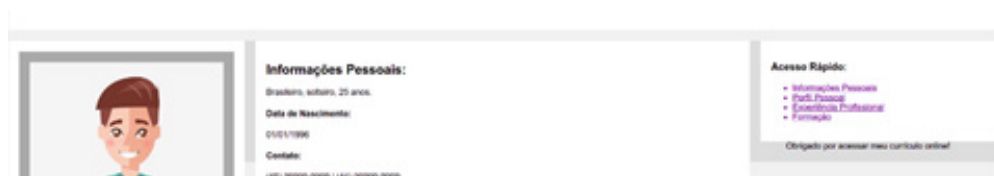


Que legal, nosso site já está muito mais bonito, não é mesmo? Porém, podemos melhorar. Vamos estilizar nossos cards a fim de organizar nosso conteúdo e deixar mais fácil de identificar cada informação. Vamos fazer isso através das propriedades **background-color**, **padding** e **margin-top**. Como vimos anteriormente, a propriedade **margin** é bem parecida com a **padding**, porém a **margin** controla o espaço entre um elemento e outro, já a **padding** controla o espaço entre o conteúdo do elemento e sua borda.

```
.card {
  background-color: white;
  padding: 20px;
  margin-top: 20px;
}
```

Por fim, vamos estilizar o nosso **footer**, adicionando as seguintes propriedades:

```
/* Rodapé */
.footer {
  padding: 20px;
  text-align: center;
  background: #ddd;
  margin-top: 20px;
}
```



Porém, nosso footer não está localizado no rodapé da página. Para corrigir isso, vamos aprender sobre o elemento **after**, “depois” em inglês. Esse elemento pode ser usado juntamente com os elementos que criamos. No nosso caso, vai ser usado juntamente com a classe **content**, que é o conteúdo do nosso site. Isso fará com que o navegador interprete que após a classe **content** ele deve adicionar um espaço vazio no nosso caso e organizá-lo como uma tabela através da propriedade **display**. Através da propriedade **clear**, indicamos que o footer não deve ter elementos flutuantes ao seu lado. Isso vai deixá-lo abaixo das outras colunas.

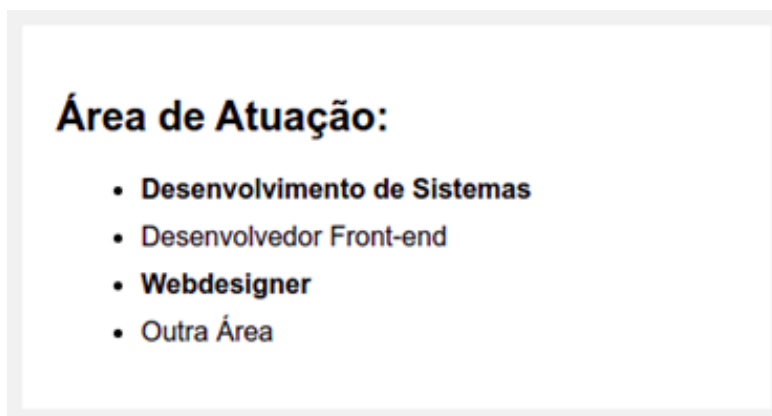
```
.content:after {
  content: "";
  display: table;
  clear: both;
}
```

Agora sim. Se você aplicou tudo, tenho certeza que seu site ficou bem organizado da maneira que foi proposta pela análise e pelo wireframe. E já que terminamos o nosso layout principal, realize a atividade a seguir para praticarmos o que aprendemos até agora.

[MOD 2] Portfólio: Praticando HTML e CSS

A prática é a melhor forma para melhorarmos nossa habilidade no desenvolvimento web. Utilizando o conhecimento obtido durante o curso, crie um novo card no seu layout, que deve conter a área de atuação de interesse e deve se localizar na coluna esquerda logo abaixo do card que contém os cursos. Porém, com um diferencial em comparação com os demais cards, esse deve conter a margem de 10 pixels entre um item e outro da lista, bem como alternar entre a fonte em negrito e normal em cada item. Você pode personalizar cada item adicionando uma classe ou identificador no HTML e referenciá-lo na folha de estilo CSS informando como ele será exibido.

Siga o exemplo abaixo, mas fique à vontade para adicionar as informações conforme sua área de atuação.



Baixe o portfólio clicando no link abaixo. Tire um print de tela contendo o layout completo do seu site, juntamente com o novo card que desenvolveu e cole no documento. Salve e guarde o portfólio com as informações, pois posteriormente haverá mais uma inclusão de atividade nele.

Você postará o portfólio para avaliação somente ao final do curso na seção de postagem!

Design responsivo

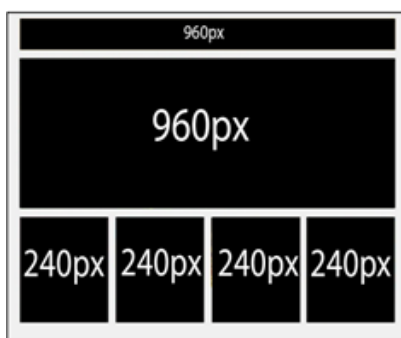
Você provavelmente já acessou algum site que não se encaixou muito bem no seu celular, não é mesmo? Aquele que fica com ícones tão pequenos que não conseguimos acessar, ou até mesmo

aquele site que tem as imagens muito grandes com o tamanho do texto pequeno que só te fazem ter vontade de sair logo dele. Pensando na variedade de dispositivos que existem hoje, se não prepararmos nosso leiaute com certa responsividade, nosso site não vai atingir o objetivo desejado, que é transmitir a informação que precisamos com confiança e clareza. Hoje em dia um usuário pode acessar nossa página do celular, computador, televisão, tablet e muitos outros. Esses dispositivos têm tamanho de tela completamente diferente e mesmo assim, seguindo as regras de acessibilidade, devemos deixar o nosso site o mais agradável possível em cada uma dessas telas. É nesse contexto que surge o design responsivo, que tem o objetivo de fazer com que sua página web seja acessada por um maior número de pessoas. Afinal, quanto mais dispositivos e resoluções suportadas, mais chances de atrair o usuário final ao nosso site. Além disso, é inviável atualmente criar um leiaute para cada dispositivo. Por meio da responsividade conseguimos fazer com que nosso leiaute “encaixe” em todos eles.

Leiautes fluidos

A técnica mais utilizada para a criação de leiautes responsivos é a criação de leiautes fluidos, em que as medidas do site tenham relativos, e não fixos, fazendo com que se adaptem ao tamanho da tela. Para o exemplo, podemos pensar nas duas principais medidas utilizadas no CSS, o pixel e a porcentagem, em que o pixel faz com que o elemento tenha um tamanho fixo, independentemente da tela. Se declararmos uma imagem com 500 pixels ela será exibida tanto no celular quanto em uma televisão de tela plano ocupando o espaço de 500 pixels. Já para a porcentagem, se declararmos uma imagem com o tamanho de 50%, ela vai assumir metade da divisão onde ela se encontra, independentemente do tamanho do dispositivo.

1. Podemos visualizar um leiaute fixo, que tem como tamanho da tela 960 pixels, e quando dividido em quatro temos divisões de 240 pixels.



2. Já aqui é possível verificar o mesmo leiaute utilizando porcentagens, em que ele manteria as mesmas proporções mesmo se alterarmos o tipo da tela.



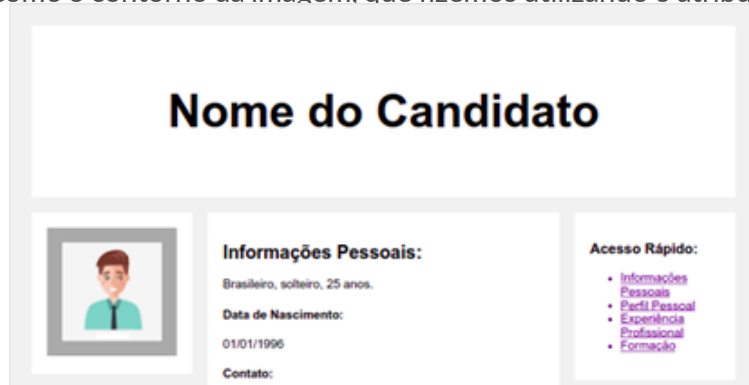
3. Outro exemplo de medidas flexíveis são as colunas do nosso site, que mantêm a mesma proporção independentemente do tamanho da tela. Definimos a coluna esquerda (classe left-column) e a coluna direita (classe right-column) com a propriedade width de 25%, ou seja, a largura delas vai ser 25% do tamanho total da página. Já a coluna do meio (classe middle-column) tem a largura de 50%, o que faz com que fique sempre com a largura da metade da tela. Faça o teste na sua página, aumentando e diminuindo o tamanho da janela do navegador.

4. Nosso leiaute na tela do computador



5. Nosso leiaute em um dispositivo móvel

Podemos verificar na imagem que as colunas manterão as proporções corretamente. Entretanto, alguns detalhes, como o contorno da imagem, que fizemos utilizando o atributo padding, por exem-



plo, não ficaram com a mesma proporção do leiaute acessado pelo computador. O contorno ficou bem mais grosso, pois como a tela é menor e seu tamanho é fixo, agora ele ocupa uma parte maior da tela do dispositivo.

Podemos resolver isso de uma maneira muito simples, utilizando o conceito de leiaute fluido.

Parte 1

Anteriormente criamos nosso CSS com essas estilizações para o elemento com a foto profissional.

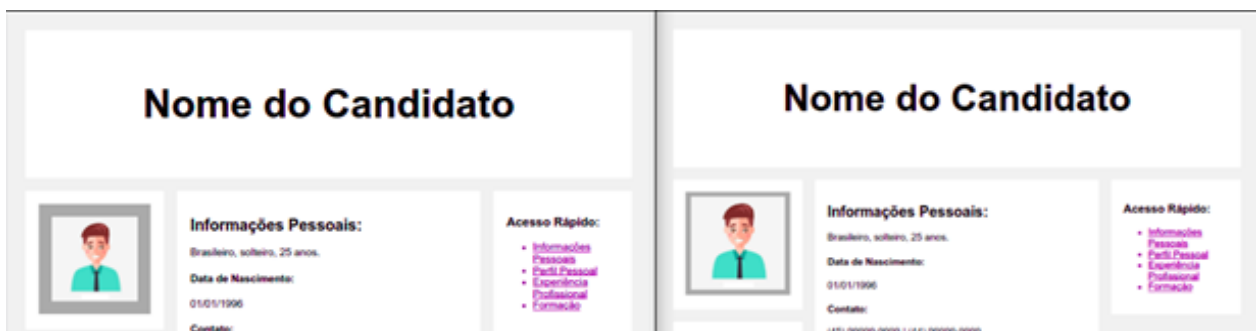
```
#profissional-picture {  
  background-color: #aaa;  
  width: 100%;  
  padding: 20px;  
}
```

Parte 2

Como vimos antes, acessamos o nosso identificador id do HTML utilizando a cerquilha (#) antes do nome do identificador. Podemos verificar que o atributo padding está fixo em 20 pixels e por isso não é responsivo. Vamos alterá-lo para 5%, como no exemplo acima.

```
#professional-picture {  
  background-color: #aaa;  
  width: 100%;  
  padding: 5%;  
}
```

Parte 3



Antes x Depois

Como resultado, teremos a mesma proporção em qualquer tipo de tela. Como no exemplo acima, onde vemos que em um dispositivo mobile do mesmo tamanho do exemplo anterior, a borda continua com um tamanho agradável e proporcional ao tamanho da foto.

Chegamos ao fim dessa mídia. Agora retorne ao AVA para continuar seus estudos!

[MÍDIA 2] MELHORIAS E RESPONSIVIDADE

Bem-vindo (a) a nossa segunda mídia de conteúdo do módulo. Aqui começamos a tratar sobre as “Melhorias e responsividades” do nosso site.

Vamos lá!

Media Queries

Nosso site está ficando legal, não é mesmo?

Porém, ainda podemos melhorar. Conforme verificamos acima, nosso site está com certa responsividade, mas em dispositivos móveis ele ainda não fica agradável para a navegação. Para isso existe outra técnica muito utilizada, que é a utilização de media queries, que são usadas para aplicar a folha de estilo de acordo com o dispositivo acessado. Através das media queries podemos dizer para o navegador que somente aquela parte do arquivo CSS será interpretada em uma televisão, por exemplo, e outra parte em um dispositivo mobile, tornando o layout diferente para cada um deles.

Declaramos uma **media query** conforme a imagem de exemplo a seguir, que informa ao navegador

que será utilizada a instrução dentro do bloco apenas em dispositivos do tipo **screen**, ou seja, em telas coloridas, através da propriedade **only**, que tenham tamanho mínimo de 768 pixels através da propriedade **min-device-width**, e tamanho máximo de 1024 pixels através da propriedade **max-devicewidth**.

```
@media only screen and (min-device-width: 768px) and (max-devicewidth: 1024px){  
    /*Folha de Estilo CSS*/  
}
```

Os tipos disponíveis de mídia são:

All: Aplica o estilo CSS para todos os dispositivos.

Braille: Aplica o CSS para leitores de Braille.

Embossed: Aplica o estilo CSS para impressoras de Braille.

Handheld: Aplica o CSS para dispositivos de mão, por exemplo, os antigos palm-tops.

Print: Aplica o CSS para impressoras convencionais.

Projection: Aplica o CSS para apresentações de slides.

Screen: Aplica o CSS para telas coloridas.

Speech: Aplica o CSS para sintetizadores de voz.

Tty: Aplica o CSS para teleimpressores e terminais.

Tv: Aplica o CSS para televisores.

Implementação da media query

Bora para a prática de implementar a media query também no nosso leiaute?

Passo 1

O primeiro passo é adicionar uma metatag no HTML, para auxiliar o navegador a entender o tamanho da janela de visualização. Fazemos isso através da tag meta, conforme exemplo a seguir, onde adicionamos a tag <meta> e identificamos, através do atributo **name** com o valor **viewport**, que vamos alterar a janela de visualização, dizendo para o navegador que essa janela terá o tamanho do dispositivo através do atributo **content** e o valor **width=device-width**. Verificamos também que a tag deve ficar dentro da tag <head>, que é a tag responsável por passar informações para o navegador que não serão exibidas em tela. Não esqueça de atualizar também o seu arquivo HTML.

```
<head>  
  <meta charset="utf-8">  
  <title></title>  
  <link rel="stylesheet" type="text/css" href="style.css" />  
  <meta name="viewport" content="width=device-width">  
</head>
```

Passo 2

O próximo passo é adicionar a media query no arquivo CSS. Para isso, utilizamos **@media**, seguido do tipo de media que desejamos, nesse caso, **all**, indicando todos os tipos de mídia como visto anteriormente. Separamos a próxima instrução com a palavra **and**, que significa “e” em inglês, e o tamanho da tela que desejamos que interprete esse grupo de instruções; nesse caso, utilizamos o **max-width** com o valor de 900 pixels para dizer ao navegador que essas instruções vão ser interpretadas quando a tela for menor do que 900 pixels.

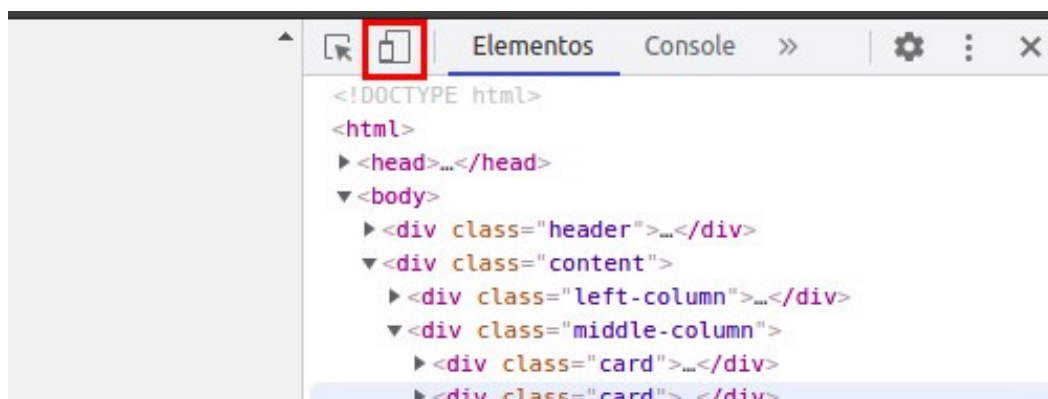
Passo 3

Para melhor organizar nosso conteúdo visual e textual e manter o comprometimento com padrões de usabilidade e acessibilidade, vamos alterar a nossa classe **header**, fazendo com que a fonte ocupe 100% do espaço disponível, e que seja responsiva de acordo com o tamanho da tela. Alteramos também as colunas principais do nosso leiaute, fazendo com que elas tomem o espaço total disponível através do atributo **width** 100%, de maneira que encaixem melhor nos dispositivos mobile, com um detalhe nas colunas esquerda (classe **left-column**) e direita (**right-column**). Retiramos o **padding** que separava elas da coluna do meio quando as três estavam lado a lado, o que não é mais necessário neste tamanho de tela.

```
@media all and (max-width: 900px) {  
  
  .header {  
    font-size: 100%;  
  }  
  
  /*Coluna esquerda */  
  .left-column {  
    width: 100%;  
    padding-right: 0px;  
  }  
  
  /* Coluna do meio */  
  .middle-column {  
    width: 100%;  
  }  
  
  /* Coluna da Direita */  
  .right-column {  
    width: 100%;  
    padding-left: 0px;  
  }  
}
```

Passo 4

Olha que legal! Nosso leiaute já está totalmente responsivo. Para testar, altere o tamanho da janela do seu navegador, ou clique com o botão direito do mouse no navegador, acesse a opção inspecionar e depois o ícone responsável por simular um dispositivo móvel, destacado na imagem abaixo. Dessa maneira podemos visualizar nosso leiaute preparado para telas menores. Legal, né?



Agora vamos à demonstração de um leiaute responsivo. Assista ao vídeo de demonstração de leiaute em aparelho móvel, clicando abaixo.

Nosso leiaute já está de acordo com as especificações da análise e seguindo os padrões do wireframe que criamos no módulo 1, mas não pare por aqui. Nos tempos atuais, onde milhares de sites são criados por dia e a quantidade de informações é extremamente grande, a inovação e criatividade são importantíssimas para que seu site chame a atenção da maneira que esperamos. Por meio da

inovação e da criatividade garantimos um diferencial ao leiaute desenvolvido, o que faz com que cada vez mais visitantes acessem nossa página, aumentando o tempo de acesso e a visibilidade das informações que estamos disponibilizando. Portanto, precisamos de um diferencial em nosso site, a cereja do bolo. O que acha de adicionarmos um pouco de interatividade a nossa página?

[MOD 1] Atividade: Criação de leiaute com editores de texto

De acordo com o que aprendemos até agora no processo de design do nosso website, analise as alternativas a seguir e assinale a alternativa **INCORRETA**.

- A. A responsividade do leiaute é importante para garantir a qualidade de navegação do usuário, bem como os padrões de acessibilidade e usabilidade em qualquer tipo de tela.
- B. A instrução `@media screen and (max-width: 900px)` indica para o navegador que o bloco de instruções será interpretado somente em telas coloridas com tamanho máximo de 900 pixels.
- C. O `padding` é o atributo responsável por controlar o espaçamento entre o conteúdo e a sua borda.
- D. Através da propriedade `width` alteramos a largura do nosso elemento, e através da propriedade `height` a altura.
- E. Os valores das propriedades utilizando a porcentagem (%) são utilizados quando não sabemos o tamanho exato que precisamos em pixel e não há outra funcionalidade considerável.

Páginas interativas

Podemos entender como páginas interativas as que possuem materiais dinâmicos e que estimulam a participação do usuário. Com a internet esse tipo de conteúdo ganha uma infinidade de formatos. Atualmente temos vídeos interativos, páginas com calculadoras, gifs animados e muitos outros formatos para exibir ou coletar informações da maneira mais instigante e divertida possível para o usuário final. O maior exemplo disso são as redes sociais, que têm como principal objetivo a interação entre os usuários, onde eles postam novos conteúdos e interagem com conteúdos de outras pessoas através de curtidas e comentários.

Propriedades CSS

O CSS possui algumas propriedades muito interessantes para tornar o site interativo, dando conforto e qualidade à navegação do usuário final. Vamos conhecer algumas dessas propriedades adicionando a interatividade à nossa página de currículo.

Atributo `position`

Para o primeiro exemplo, vamos fixar o menu lateral direito, que contém nossos links de acesso rápido, no topo da página, gerando um efeito que acompanha o usuário enquanto ele rola a página. Faremos isso adicionando o atributo **`position`** com o valor **`sticky`**. O posicionamento **`sticky`** é baseado na posição da barra de rolagem da página. Ele altera entre o posicionamento fixo e relativo, sempre de acordo com a barra de rolagem.

Atributo `top`

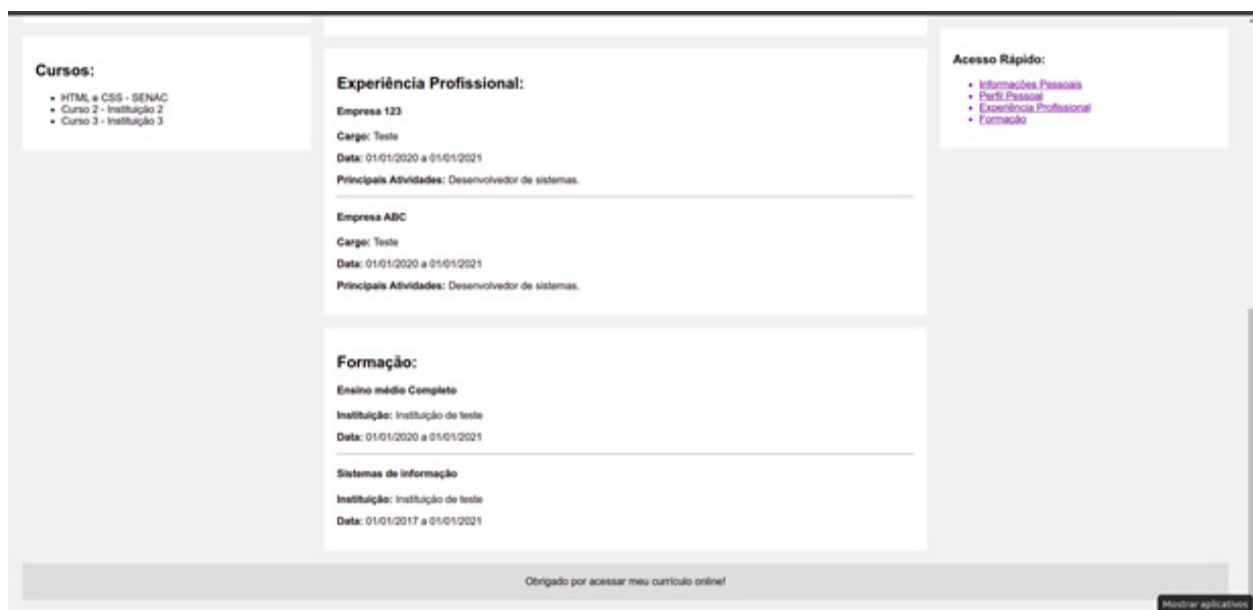
Vamos então alterar nosso arquivo CSS adicionando o atributo **`position`** com o valor **`sticky`** e o

atributo **top** com o valor 0. Isso indica que nosso menu ficará fixo ao topo da tela quando o usuário mexer na barra de rolagem. Nossa coluna da direita (classe `right-column`) fica então da seguinte maneira. Não esqueça de atualizar também o seu arquivo CSS.

```
/* Coluna da Direita */
.right-column {
  float: left;
  width: 25%;
  padding-left: 20px;
  position: sticky;
  top: 0;
}
```

Experiência do usuário

Acessando nosso site podemos verificar que o menu segue o usuário durante toda a página, trazendo um efeito interativo ao nosso site. Isso melhorou não só a experiência do usuário, mas também a usabilidade do site, sendo que agora o usuário pode clicar em algum dos itens do menu de acesso rápido quando quiser.



Atributo animation

Agora vem a parte mais legal: vamos adicionar animações ao nosso site. Faremos isso através do atributo **animation**. O primeiro passo é criar nossa **keyframe** no arquivo CSS, conforme imagem a seguir, onde podemos verificar que foi criado um **keyframe** com o nome **rotate-image**. Esse nome é o identificador do **keyframe** e será usado no item que queremos girar, ou seja, na imagem. Dentro do nosso **keyframe** vamos criar outro bloco com o atributo 50%. Isso indica para o navegador que quando o item afetado estiver com 50% do movimento completo ele deve assumir as características desse bloco.

Atributo transform

O próximo passo é adicionar o nosso movimento através do atributo **transform**. Nesse caso, ele

vai iniciar o movimento indo 10 pixels para o fundo da página, através do atributo **translateZ(-10px)**. Após esse movimento o nosso item irá girar 60 graus para o lado, através do atributo **rotateX(60deg)**, girar 29 graus para o fundo da tela através do atributo **rotateZ(29deg)** e por fim girar para baixo através do atributo **rotateY(-180deg)**. Adicione a alteração também no seu arquivo CSS.

```
@keyframes rotate-image {  
  50% {  
    transform: translateZ(-10px) rotateX(60deg) rotateZ(29deg) rotateY(-180deg);  
  }  
}
```

Agora que você acompanhou todo passo a passo, vamos criar outro bloco no nosso arquivo CSS referenciando a imagem através do id e passando para o navegador que esse item terá uma animação chamada **rotate-image**, por meio do atributo **animation-name**.

Alteramos a duração da animação através do atributo **animation-duration** para 4 segundos, e a velocidade da animação com o atributo **animation-timing-function**. Nesse caso, vamos utilizar o **ease-in-out**, que é um movimento mais lento.

Teremos também um delay para o início da animação de 2 segundos, ou seja, a página vai aguardar 2 segundos antes de iniciar a animação. Controlamos isso através do atributo **animation-delay**.

E, por fim, vamos alterar também quantas vezes a animação vai realizar o movimento através do atributo **animation-iteration-count**.

Altere também seu arquivo CSS e recarregue a página do navegador para verificar os resultados.

```
#profissional-picture {  
  animation-name: rotate-image;  
  animation-duration: 4s;  
  animation-timing-function: ease-in-out;  
  animation-delay: 2s;  
  animation-iteration-count: 2;  
}
```

Vamos praticar tudo o que aprendemos até agora adicionando mais algum diferencial ao seu leiaute. Para isso, realize a atividade a seguir.

Por que você deve aprender HTML5 e CSS?

O objetivo principal da internet é compartilhar informação com todas as pessoas de forma rápida, dinâmica e acessível por qualquer dispositivo eletrônico.

Se você pretende seguir trabalhando com tecnologia, seja desenvolvendo um sistema interno para uma loja, um portal de notícias, ou sites de diversos temas, é imprescindível ter uma ótima base de conhecimento em HTML e CSS, pois tudo que escrevemos utilizando-os ganha vida dentro dos navegadores escolhidos por quem acessa as páginas.

O HTML é a primeira camada em que o usuário se serve da informação disponibilizada. É ele que dá significado para ser reconhecido por aplicações, sistemas, robôs, ou qualquer outro meio onde a informação foi publicada. Um sistema de busca, leitores de tela, smartphones, browsers, aplicações, sistemas, não tem discernimento para “enxergar” e entender visualmente o que é um título, um parágrafo, uma imagem etc. A importância do HTML é definir a marcação para cada informação.

Não havia um padrão universal para criação de seções comuns e específicas nas versões antigas do HTML, havia um problema semântico que definia a estrutura de um site sem hierarquia própria, por exemplo, a tag “div”, por ser um elemento genérico era utilizada excessivamente em quase toda estrutura de um site, mesmo isso nunca tendo impedido um web designer de colocar um site no ar, acabava gerando um certo ruído e uma necessidade de se utilizar classes CSS para identificar o papel de cada “div” dentro de um site.

A seguir, assista ao vídeo de atualização das marcações feitas anteriormente de forma híbrida (HTML4 e 5) agora toda em HTML5.

Caso queira ler a transcrição do vídeo clique [AQUI!](#)

HTML 5

E aí aluno, o que está achando do curso até agora?

Nosso leiaute está pronto, de acordo com as especificações da análise e seguindo a estrutura do nosso wireframe. Vamos agora realizar algumas alterações deixando nosso leiaute no padrão HTML5, tornando nosso leiaute mais semântico, ou seja, utilizando as tags corretas para cada marcação que criamos, facilitando o entendimento da estrutura e a interpretação de ferramentas externas, como o google por exemplo.

Uma das novidades do HTML é a tag **<header>**, através dela nós evitamos utilizar uma **<div>** com a classe header como fizemos até aqui, alterando podemos verificar que nossa marcação fica mais bonita e organizada não é mesmo? Além de que quando um sistema de busca como o google for analisar nossa página, ele vai entender que essa parte é o header do nosso site. Vamos recarregar o site e podemos verificar que o estilo que criamos se perdeu, a final, não estamos mais utilizando a classe header para marcar essa sessão, para corrigir isso, vamos atualizar também nosso arquivo style.css e alterar a marcação **.header**, onde o ponto indica que é uma classe, para **header**, sem o ponto, identificando ao navegador que o header agora é uma tag HTML.

O HTML5 disponibiliza também a tag **<footer>**, da mesma maneira que o header, podemos alterar a nossa **<div>** com a classe footer somente para a tag **<footer>**, identificando para o navegador que esse é o rodapé da nossa página, lembre-se que devemos alterar também o arquivo style.css, retirando o ponto que indica que o footer é uma classe, pois agora ele passa a ser uma tag de marcação, para que assim, não perdermos a estilização dessa parte do site.

Para identificar o conteúdo principal da nossa página, vamos alterar a **<div>** com a classe content, que era responsável por identificar o conteúdo, para a tag **<main>**, que marca o conteúdo principal e deve ser usado somente uma vez em cada página. Devemos alterar também no arquivo de estilo.

Outra tag que é do HTML5 interessante para a semântica da nossa página é a **<section>**, essa tag define as sessões que teremos no site, que no nosso caso são três, a coluna da esquerda, a do meio e a da direita. Vamos alterar as divs que separavam essas seções para section e utilizá-las da mesma

maneira, mantendo a classe que identifica cada uma.

Vamos alterar também os nossos cards, que anteriormente criamos utilizando uma `<div>` com a classe card e agora vamos utilizar a tag `<article>`, que define algum conteúdo que vamos adicionar ao nosso site. As ferramentas de busca vão identificar esse bloco como um conteúdo importante para a página, pode ser um texto, um documento e outros.

Por fim, outra marcação interessante para adicionarmos ao nosso layout é a tag `<nav>`, essa tag é responsável por identificar uma área de navegação, contendo links que redirecionam para outra parte da página, essa alteração não terá mudanças visuais, somente de marcação, organizando as linhas de código e facilitando a interpretação de outras ferramentas.

Com o surgimento do HTML5, a forma como escrevemos código e organizamos a informação na página foi modificada, trazendo elementos que supriram essa falta de tags mais descritivas e semânticas ao substituir elementos clássicos como cabeçalhos, menus de navegação, rodapés e áreas de conteúdo secundário, por tags como “header”, “nav”, “section”, “article”, “aside” e “footer”.

O HTML5 é mais semântico pois utiliza menos código, é mais interativo, e não há necessidade de instalarmos plugins e consequentemente perdermos performance na nossa aplicação.

Para que o conteúdo dessa informação seja agradável, o CSS é responsável em formatar a informação exibida pelo HTML, para que ela seja consumida da melhor maneira possível, de modo que esteja acessível em qualquer dispositivo e isso vai muito além do que pintar divs, formatar letras e cores.

Este curso foi início da sua caminhada na área de desenvolvimento, para aprender ainda mais, existem sites, livros onde você pode aprender a respeito de HTML e CSS.

[MOD 2] Portfólio: Inovação e criatividade

Inovação e criatividade são importantíssimas para o ambiente web. Através delas criamos um diferencial no nosso site em relação aos demais, chamando a atenção dos usuários interessados. Afinal, em um currículo online o interessante é que o máximo de pessoas possível acesse suas informações profissionais e consiga absorver o conteúdo da página, aumentando assim as chances de atingir o objetivo final, que é entrar para o mercado de trabalho.

- A partir do conhecimento obtido durante o curso, juntamente com a sua capacidade de inovação e criatividade, altere seu layout a fim de criar um diferencial em relação aos demais e ao layout padrão que criamos durante o nosso curso.
- Altere as cores, adicione animações, adicione informações que julgue importante e altere a forma de exibição, se julgar necessário, sempre lembrando de manter os padrões de usabilidade e acessibilidade para que seu layout fique incrível! Personalize seu layout para torná-lo único e poste para que inspire os demais alunos.

Orientação

As oportunidades disponíveis para estilizarmos um layout são extremamente grandes. Em outras palavras, a sua criatividade é o limite!

- Você pode decidir entre inúmeras combinações de cores.
- Organizar a tela da maneira que desejar e até adicionar animações que alteram a exibição inicial da página, conforme o usuário a usa e utiliza a barra de rolagem, por exemplo.

Personalize seu leiaute para torná-lo único e alimente no seu portfólio, usado na mídia anterior. Salve e envie ao seu tutor. Para isso, retorne ao AVA.

Chegamos ao fim dessa mídia. Agora retorne ao AVA para postar seu portfólio completo na seção [Postagem]!

[POSTAGEM] PORTFÓLIO

Parabéns, estamos chegando à conclusão da sua última atividade do módulo. Esperamos que tenha realizado com muita atenção e segurança nos processos aprendidos sobre combinações, organização e estilização em geral desenvolvidos no curso.

Realizadas as duas ações e inseridas no portfólio, poste-o ao seu tutor clicando em [Postagem] Portfólio.

[MOD 2] Portfólio: Praticando HTML e CSS

Colar print aqui.

Observações:

<Anotar aqui seus insights e/ou informações para explicar ao tutor, caso ache necessário.>

[MOD 2] Portfólio: Inovação e criatividade

Colar print aqui.

Observações:

<Anotar aqui seus insights e/ou informações para explicar ao tutor, caso ache necessário.>

[SÍNTESE] CURSO HTML E CSS

Parabéns, aluno, já estamos com o leiaute pronto para a próxima etapa da criação do projeto, adicionar as informações ao site e validar todas as funcionalidades, imagens, textos e links e garantir que nenhum problema chegue ao usuário final. Espero que tenha atingido a sua expectativa e o seu objetivo neste curso. Nunca se esqueça que a prática leva à perfeição.

Palavra do autor, clique no PLAY!

Caso não consiga ouvir, leia o PDF de transcrição.

Palavra do autor

Parabéns, aluno (a)!

Já estamos com o leiaute pronto para a última etapa da criação do projeto, que é adicionar as informações ao site, validar todas as suas funcionalidades, imagens, textos e links e garantir que nenhum problema chegue ao usuário final.

Espero que tenha atingido a sua expectativa e o seu objetivo neste curso. Nunca se esqueça que a prática leva à perfeição. Praticar muito e criar um portfólio legal é muito importante para o mercado de trabalho no ramo de desenvolvimento.

Bom trabalho, bons estudos e até mais!

Pratique muito e crie um portfólio legal, pois isso é muito importante para o mercado de trabalho no ramo de desenvolvimento.

Não perca tempo, se desenvolva e bom trabalho!

Vitor Gabriel Martines Weinfortner



```
function getProductById() {  
  return this.getProductById('product_id');  
}  
  
function getCategoryById() {  
  return this.getProductById('category');  
}  
  
function getProduct() {  
  const url = new URL(window.location.href);  
  const productId = url.searchParams.get('product_id');  
  const categoryId = url.searchParams.get('category_id');  
  const products = this.getProductById(productId);  
  const categories = this.getProductById(categoryId);  
  const productCollection = products.map(product => {  
    id: product.id,  
    name: product.name,  
    price: product.price,  
    category: product.category,  
    status: product.status,  
    createdAt: product.createdAt,  
    updatedAt: product.updatedAt,  
  });  
  const categoryCollection = categories.map(category => {  
    id: category.id,  
    name: category.name,  
    status: category.status,  
    createdAt: category.createdAt,  
    updatedAt: category.updatedAt,  
  });  
  return {  
    productCollection,  
    categoryCollection,  
  };  
}
```


Senax