

ESCOLA E FACULDADE DE TECNOLOGIA SENAI ROBERTO MANGE

DESENVOLVIMENTO DE SISTEMAS

JULHO DE 2022 — SENAI-SP



SENAI
SÃO PAULO



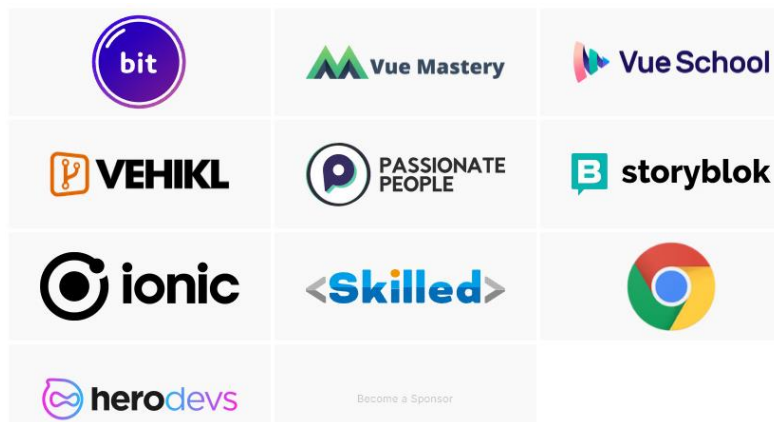
FRONTEND: FRAMEWORK VUEJS

Nesta etapa de estudo iremos utilizar um dos 3 grandes frameworks frontend que dominam o mercado: o **Vue JS**!



The Progressive JavaScript Framework

An approachable, performant and versatile framework for building web user interfaces.



→ Principais patrocinadores deste framework!





FRONTEND: FRAMEWORK NUXT 3

Contudo, da mesma forma que hoje temos o Next para React, utilizaremos o **NUXT 3** que se trata de um framework do framework Vue JS, dando-lhe ainda mais capacidade, produtividade, e performance com os conceitos de SSR+SPA.

The Intuitive Vue Framework


Nuxt is an open source framework that makes web development intuitive and powerful.
Create performant and production-grade full-stack web apps and websites with confidence.

 Get Started

 What is Nuxt?

Trusted by the best front-end teams

GitHub

 OpenAI

NASA

Google

fedora

 GitLab

upwork

Principais patrocinadores deste framework!

<https://nuxt.com/>



INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT



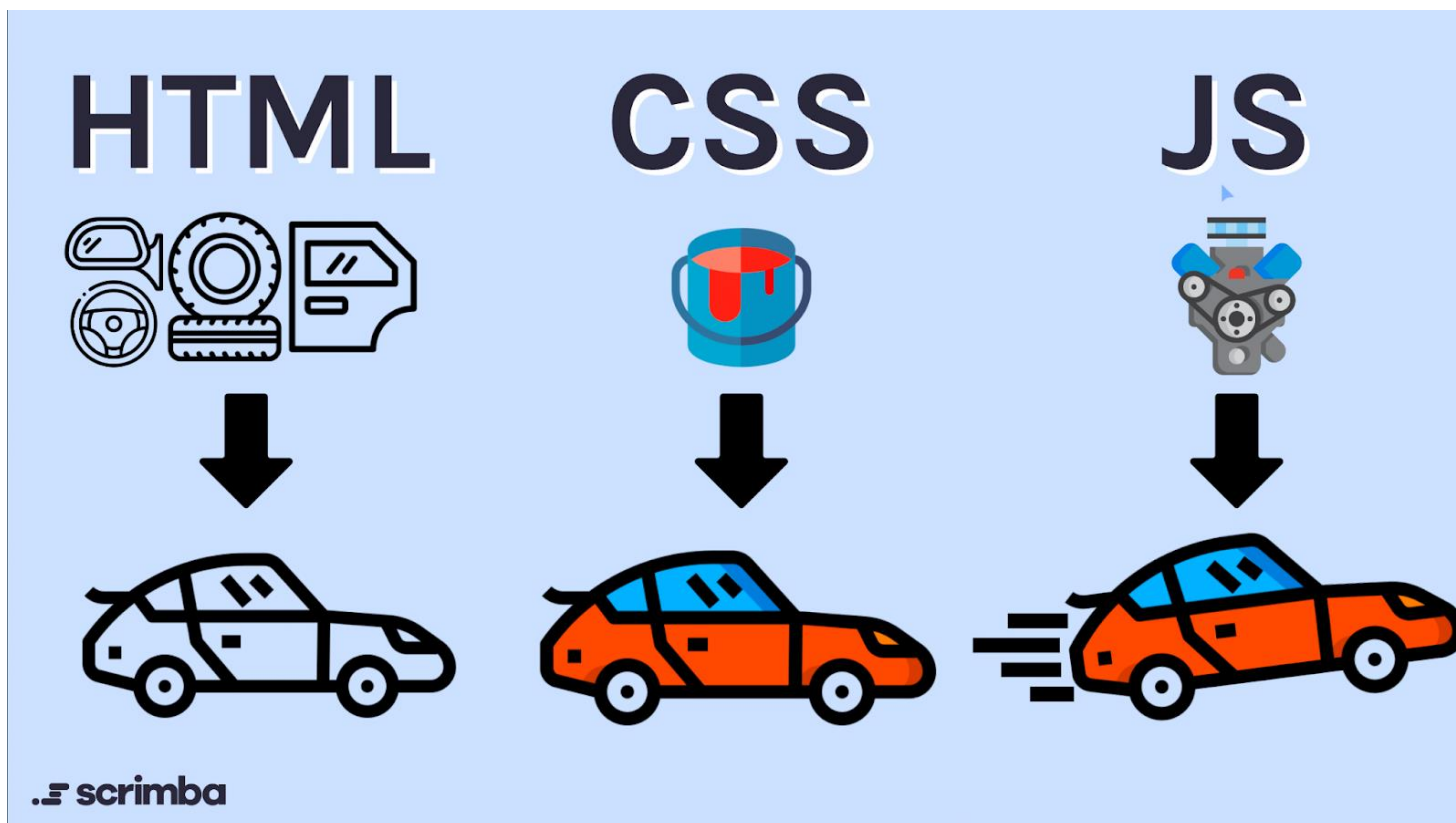
A navegação de qualquer página web que usamos atualmente é composta por no mínimo de 3 partes (independente se você está usando um framework ou não): **HTML, CSS, JAVASCRIPT**.

HTML → não é uma linguagem de programação, mas sim de marcação, ou seja, assim como tijolos em uma casa, é responsável por definir a estrutura de uma página.

CSS → usando apenas o html já possuímos uma estrutura que é o fundamental, contudo, geralmente essa estrutura não é tão visualmente agradável, fazendo parecer aqueles sites 'antigos', portanto o css é o responsável por definir como será o layout de nossa página web (não é uma linguagem de programação).

JAVASCRIPT → é o responsável por fazer com que as páginas ganhem 'vida', ou seja, deixem de ser estáticas e passem a ser dinâmicas e interativas, possibilitando a realização de lógica (é uma linguagem de programação)

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT



Esta analogia é interessante, o Html é as peças estruturais do carro, o Css é o layout do carro (cor, etc.), e o Javascript é a motorização que vai fazer o carro ter ação e ser dinâmico e obedecer os comandos do motorista.

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

```

<!DOCTYPE html>
<html Lang="pt_br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login University</title>
  <link rel="stylesheet" href="./style.css">
</head>
<body>
  <main class="login">
    <section class="login_panel">
      <div class="login_content">
        <div class="login_title">
          <h2>Login</h2>
          <a href="">Cadastrar?</a>
        </div>
        <form class="login_form" onsubmit="sendCredentialsToBackend(event)">
          <div class="form_content">
            <label for="e-mail">E-mail</label>
            <input type="email" class="login_input" id="e-mail"
              placeholder="Digite seu e-mail" required>
          </div>
          <div class="form_content">
            <label for="pass">Senha</label>
            <input type="password" class="login_input" id="pass"
              placeholder="Digite sua senha" required>
          </div>
          <div class="form_button">
            <button type="submit" id="login_button">Login</button>
          </div>
        </form>
      </div>
    </section>
  </main>
</body>
</html>

```

HTML

Observe como é estruturado o Html...

Os itens na imagem em verde são as **TAGS**, isto é, estruturas previamente criadas que possuem funcionalidade específica.

Em laranja estão os **ATRIBUTOS** das tags, ou seja, propriedades e configurações das tags.

Em branco entre uma tag e outra temos o **CONTEÚDO**, que podem ser textos ou outras tags.

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login University</title>
  <link rel="stylesheet" href="/style.css">
</head>
<body>
  <main class="login">
    <section class="login_panel">
      <div class="login_content">
        <div class="login_title">
          <h2>Login</h2>
          <a href="">Cadastrar?</a>
        </div>
        <form class="login_form" onsubmit="sendCredentialsToBackend(event)">
          <div class="form_content">
            <label for="e-mail">E-mail</label>
            <input type="email" class="login_input" id="e-mail"
              placeholder="Digite seu e-mail" required>
          </div>
          <div class="form_content">
            <label for="pass">Senha</label>
            <input type="password" class="login_input" id="pass"
              placeholder="Digite sua senha" required>
          </div>
          <div class="form_button">
            <button type="submit" id="login_button">Login</button>
          </div>
        </form>
      </div>
    </section>
  </main>
</body>
</html>
```

Principais tags e uso:

DOCTYPE: define o tipo do documento para informar o navegador;

html: abriga toda a estrutura em html da página;

head: contém as principais configurações da página;

meta: realizar configurações importantes, como codificação dos caracteres, descrição, meta dados, etc... (é opcional, mas importante);

body: contém todo o corpo da página, ou seja, a maior parte e a principal;

div: tags genéricas que servem como 'caixas' que literalmente podem conter qualquer coisa, inclusive várias tags dentro de si.

main: tag semântica par indicar que contém a parte principal da sua página (tags semânticas são tags não obrigatórias e que podem ser substituídas por outras, em sua maioria pelas tags div, porém tem o sentido especial de dar 'dicas' para o navegador/browser entender melhor a estrutura de sua página);

form: tag que indica que dentro dela existe um formulário a ser preenchido



INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

ESTRUTURA BÁSICA:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```



```
</body>
```

```
</html>
```

HTML

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

```
<section>
  <h5>Cadastre-se em nossa newsletter! 📧</h5>
  <form onsubmit="saveNewsletterUser(event)">
    <div>
      <label for="name">Nome: </label>
      <input
        id="name"
        type="text"
        maxlength="30"
        placeholder="Digite seu nome completo"
        required
      />
    </div>
    <div>
      <label for="email">E-mail:</label>
      <input
        id="email"
        type="email"
        maxlength="30"
        placeholder="Digite seu melhor e-mail"
        required
      />
    </div>
    <div>
      <button type="submit">Cadastrar!</button>
    </div>
  </form>
  <br><br>
</section>
```

Principais tags e uso:

input: tag que recebe dados do usuário que está acessando a página, podendo ser de vários tipos definidos no atributo *type*, ou seja, podem ser desde campos de texto, caixas de seleção, seletores de calendário, etc...

label: geralmente usada em conjunto com input, serve para nomear o input, como se fosse uma etiqueta;

button: como o nome diz, é um botão que tem o objetivo de fazer uma ação;

h1 ~ h6: as tags h1 até o h6 servem para imprimir textos importantes, como títulos por exemplo;

p: tag de parágrafo, imprime um texto e após a finalização do mesmo quebra uma linha (inicia o próximo texto em outro parágrafo);

span: semelhante a tag p, porém não quebra uma linha, ou seja, dois spans normalmente irão ficar um em frente ao outro (caso não seja alterado via css);

a: tag que serve para receber um link e direcionar o Usuário para o link desejado quando clicado;

img: tag para mostrar uma imagem local ou da internet.

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

```
.Login{
  height: 100vh;
  width: 100vw;
  display: flex;
  flex-direction: row;
}

.Login_panel{
  height: 100vh;
  width: 50vw;
  background-color: white;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}

.Logo_panel{
  height: 100vh;
  width: 50vw;
  background-color: #231f20;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}

.Logo_content{
```

```
#login_button {
  width: 100%;
  height: 35px;
  font-weight: bold;
  border: none;
  border-radius: 5px;
  color: white;
  background: linear-gradient(
    135deg,
    rgba(16,221,211,0.849),
    rgba(2,165,230,0.87)
  );
}
```

```
html, body{
  height: 100%;
}
```

Observe como é estruturado o CSS...

Basicamente o principal objetivo dos comandos em css é **selecionar / localizar** as **Tags, Classes ou Ids** existentes no Html e modificar o seu respectivo layout.

Nestes exemplos temos alguns desses tipos de seletores:

- (ponto) → é usado para selecionar as classes do html.

→ usado para selecionar os Id do html

Nome da tag → usado para selecionar as tags pelo nome no html

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

```
function showMessage(){
    console.log("Exibindo minha mensagem no console!!!!");
    //alert("Clicou no botão!!!"); //parar a aplicação até dar ok no aviso
}

function saveNewsletterUser(event){
    //evitar o comportamento de reload da página ao enviar formulario
    event.preventDefault();
    console.log('evento: ', event);
    //seleciona os campos existentes no html usando id (#)
    const elementName = document.querySelector("#name").value;
    const elementEmail = document.querySelector("#email").value;

    console.log("elementName: ", elementName);
    console.log("elementEmail: ", elementEmail);

    alert("Formulário enviado!");
}
```

Observe como é estruturado o Javascript...

Basicamente, como se trata de uma linguagem de programação, vai conter variáveis, funções, lógicas como if, else, for, while.....

Tipos básicos de variáveis em Javascript (JS):

const → variável que após definida não pode ser mudada;

var → variável que pode mudar e é válida em todo Javascript (variável global);

let → variável que pode mudar mas é válida apenas no contexto onde ela é declarada (variável local), ou seja, se nascer em uma função só poder ser acessada dentro dessa função.

INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

```
function showMessage(){
  console.log("Exibindo minha mensagem no console!!!!");
  //alert("Clicou no botão!!!"); //parar a aplicação até dar ok no aviso
}

function saveNewsletterUser(event){
  //evitar o comportamento de reload da página ao enviar formulario
  event.preventDefault();
  console.log('evento: ', event);
  //seleciona os campos existentes no html usando id (#)
  const elementName = document.querySelector("#name").value;
  const elementEmail = document.querySelector("#email").value;

  console.log("elementName: ", elementName);
  console.log("elementEmail: ", elementEmail);

  alert("Formulário enviado!");
}
```

Informações sobre o JS:

console.log → função que imprime no console do navegador um texto ou conteúdo de uma variável, ou seja, usada apenas para teste;

alert → função que imprime na tela um texto ou o conteúdo de uma variável, porém ao usarmos frameworks essa funcionalidade não é tão usada pois existem maneiras melhores e mais personalizáveis de se mostrar mensagens.

document.querySelector → uma forma nativa de fazer o javascript selecionar um elemento existente no html (neste caso a tag com id name e email) para poder consultar valor digitado, etc.



INTRODUÇÃO AOS CONCEITOS HTML, CSS, JAVASCRIPT

CONSIDERAÇÕES:

Esta forma de criarmos páginas usando simplesmente Html, Css e Javascript não é a que adotaremos, pois necessitamos criar aplicações mais complexas, confiáveis e com boa performance, portanto, utilizaremos à partir de agora o framework Nuxt usando o Vue JS como base.

Em suma, o Html , Css e Javascript consistirão em sua essência quase que a mesma coisa que vimos nesta apresentação, porém agora teremos tags e funções especiais herdadas pelo framework que estaremos usando.

Obs.: quando estiver desenvolvendo sem framework, como sugestão de uso você pode fazer o download da extensão do vsCode chamada LiveServer, e, após instalada, clique com o botão direito de sua página html e selecione o LiveServer, desse modo, você terá um servidor com carregamento automático (auto reload) de sua página.