



# Estácio

## DESENVOLVIMENTO WEB

HTML, CSS, JavaScript



Professor: Dr. **Rodrigo Fujioka**

<https://www.fujideia.com.br/fuji> | [linkedin/rodrigofujioka](https://www.linkedin.com/in/rodrigofujioka) | [@rodrigofujioka](https://www.instagram.com/rodrigofujioka)

# PROFESSOR / PROFISSIONAL

- **1º Semestre na Estácio.**
- Ministrando aulas desde 2010.
- **Ministrou aulas em diversas IES na graduação e pós graduação.**
- Avaliador do Mec/Basis desde 2018.
- **Atua no mercado de TI com carteira assinada/PJ desde 2008.**

Professor: Dr. Rodrigo Fujioka |  
<https://www.fujideia.com.br/fuji> | [linkedin/rodrigofujioka](https://www.linkedin.com/in/rodrigofujioka)  
| [@rodrigofujioka](https://www.instagram.com/rodrigofujioka)



Rodrigo Fujioka

Find the Wayback Machine useful? [DONATE](#)

DONATE

[www.rodrigofujioka.com](http://www.rodrigofujioka.com)

*rodrigo fujioka*

 8 0

0

0

24 web captures from 2008 to 2015

[rodrigofujioka.hpg.com.br](http://rodrigofujioka.hpg.com.br)

rodrigo c fujioka

 9 0

0

0

27 web captures from 2001 to 2003

# Academia



Arquiteturas de Referência 3D



Rodrigo Fujioka · Fernando F. Souza · Marcus S. Aquino

## Arquitetura de Referência para Gerenciamento de Ambientes Virtuais 3D

Uma abordagem baseada em Reuso, SOA  
e Computação em Nuvem para  
compartilhamento e distribuição de  
componentes 3D

Fujioka, Souza, Aquino

 Novas Edições  
Acadêmicas

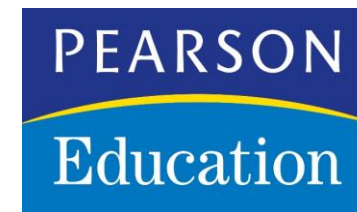




Estácio

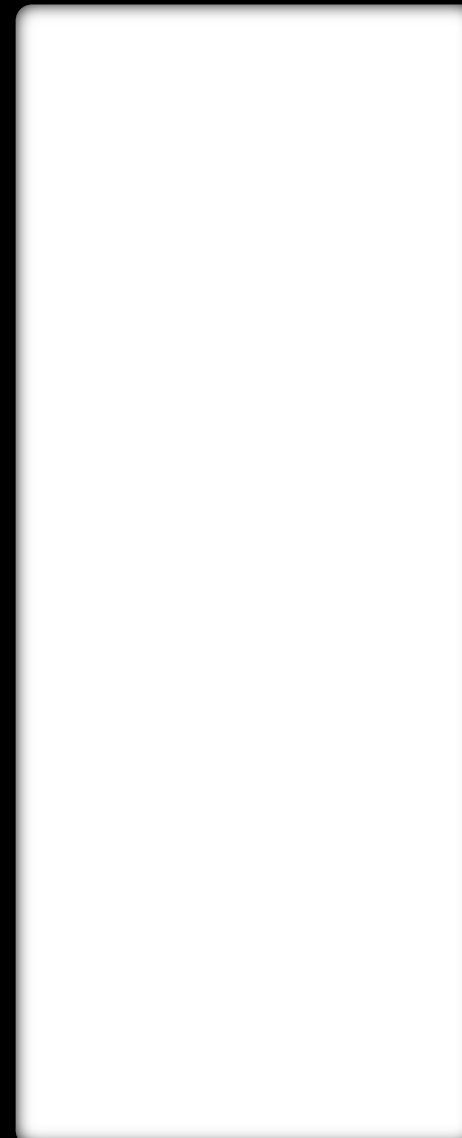


Rodrigo da Cruz Fujioka



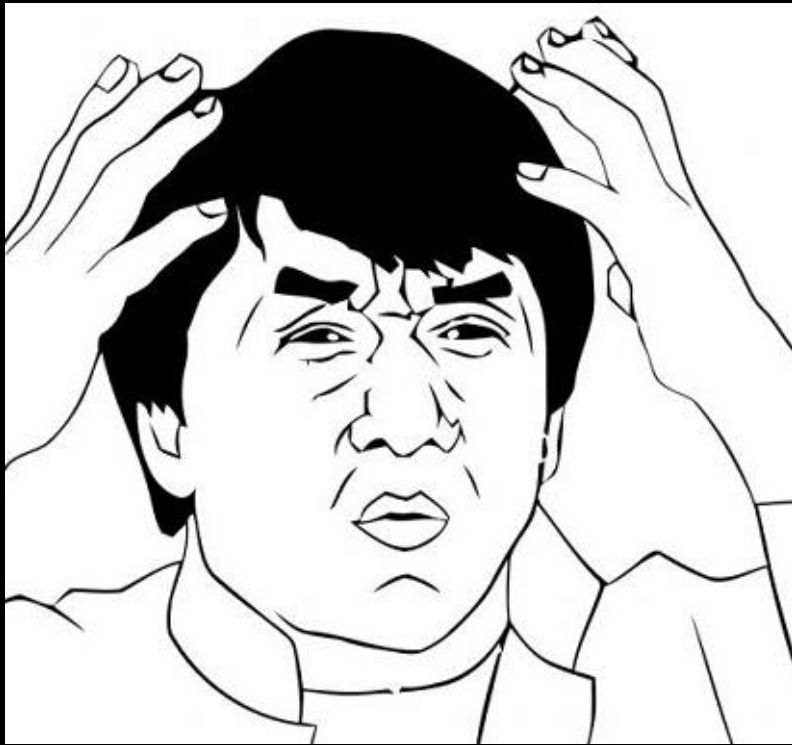
# O QUE VAMOS APRENDER DURANTE O CURSO.

- Estruturar páginas com HTML.
- Estilizar páginas com CSS.
- Programar com JavaScript e PHP.
- Criar uma APP com Angular (Extra Depende do Andamento).

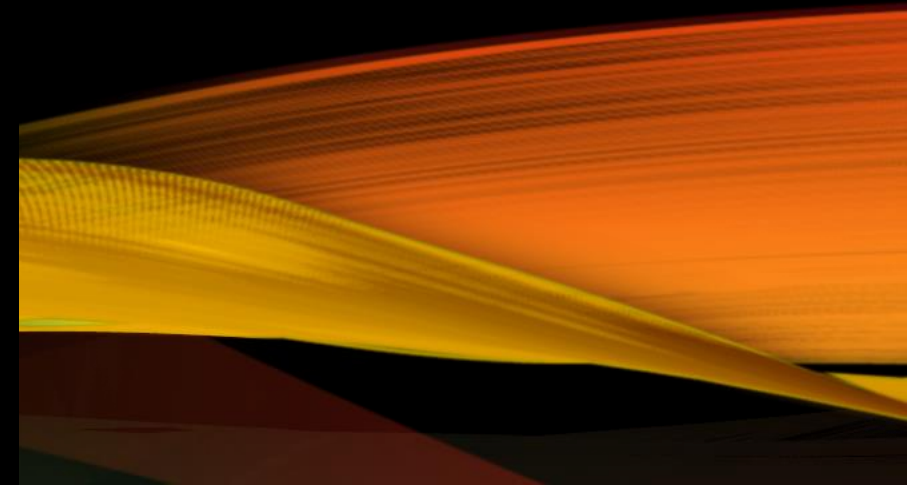


# REQUISITOS

- Visual Studio Code.
- PHPStorm (JetBrains).
- XAAMP
- GIT



**TODO  
SEMESTRE**







# AVALIAÇÃO

- Sistema de atendimento.
- Exercícios + Prova.



# VAMOS COMEÇAR ?

- HTML
- <https://repl.it/@RodrigoFujioka/Aula01-Estacio-de-Sa>
- <https://www.w3schools.com/html/>

## HTML



## CSS



## JS

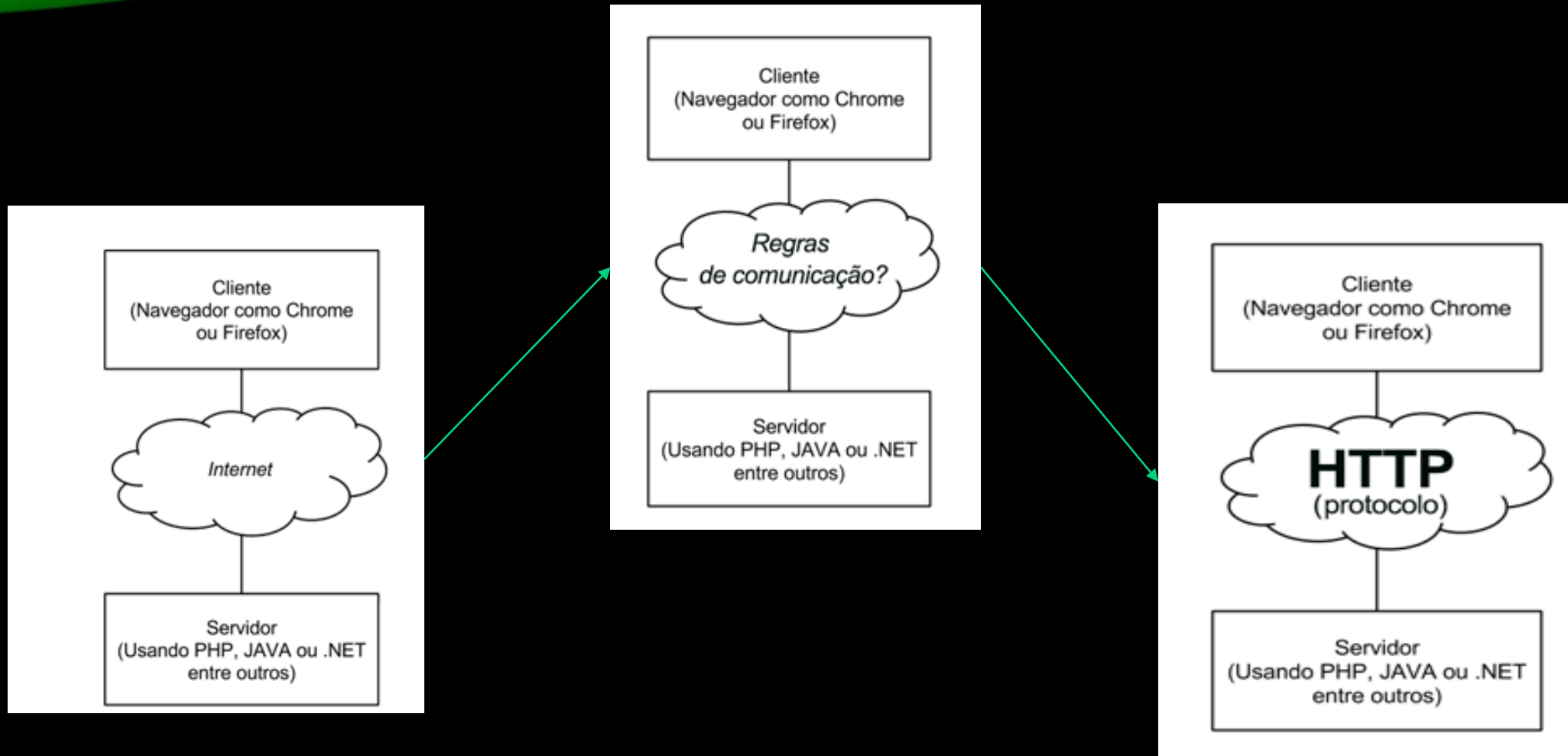


# ANTES DE INICIAR

- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/css/>
- <https://www.w3schools.com/js/>

# HTTP *Hypertext Transfer Protocol*

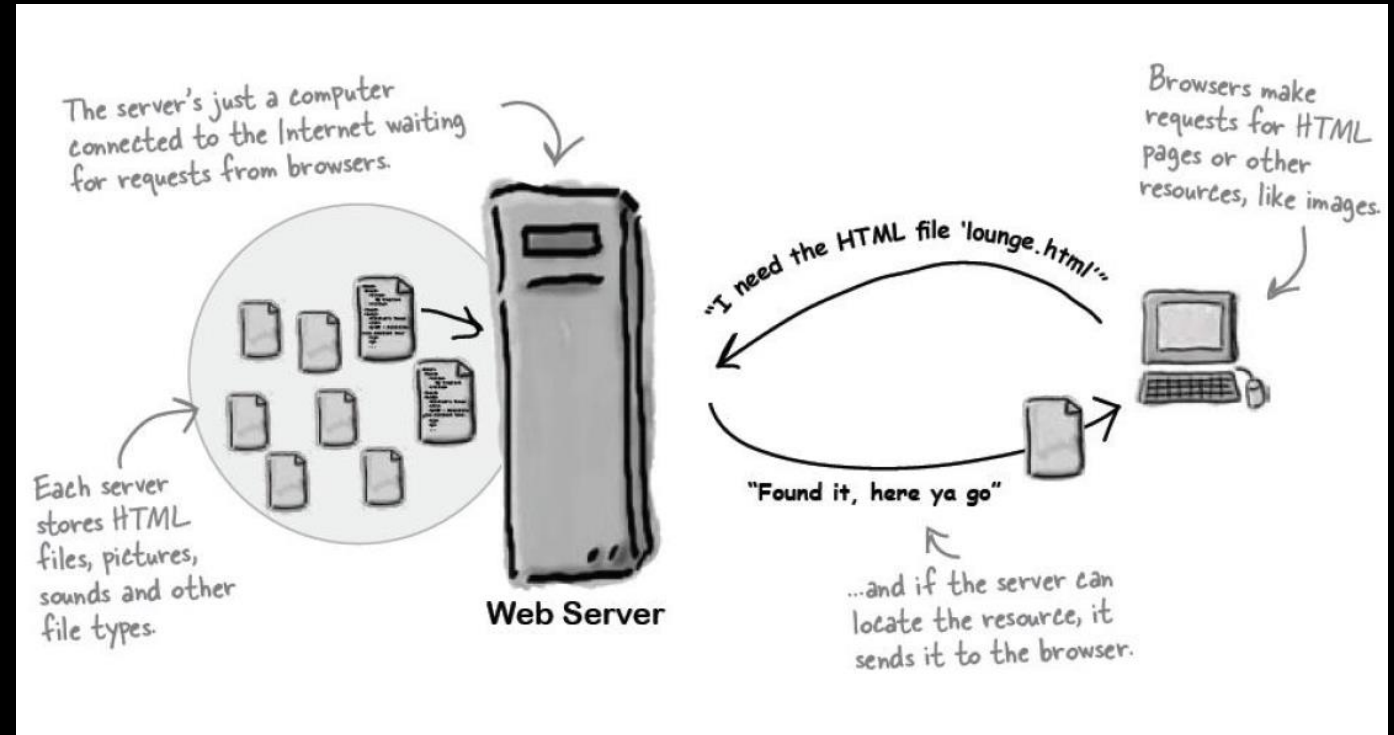
## HTTPS *Hyper Text Transfer Protocol Secure*



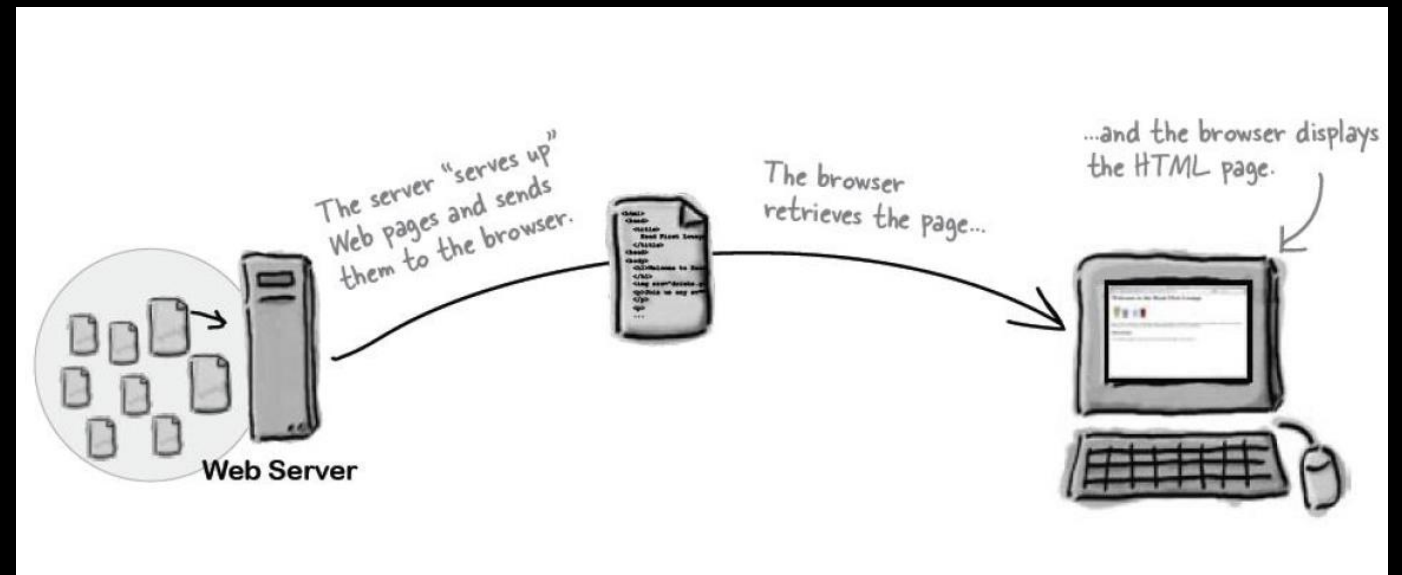


# SERVIDOR

Para comunicação na internet, sempre vai ser utilizado algum protocolo entre o cliente e o servidor. Sendo um dos mais importantes o HTTP, na comunicação as regras de como o processo vai ocorrer são definidas dentro do protocolo utilizado. Nas nossas aulas vamos utilizar o HTTP.



# CLIENTE



## ▼ General

**Request URL:** `https://www.fujideia.com.br/`

**Request Method:** GET

**Status Code:** 🟢 200

**Remote Address:** 185.201.10.132:443

**Referrer Policy:** no-referrer-when-downgrade

## ▼ Response Headers

**alt-svc:** quic=":443"; ma=2592000; v="43,46", h3-Q043=":443"; ma=2592000

**cache-control:** no-store, no-cache, must-revalidate

**content-encoding:** br

**content-type:** text/html; charset=utf-8

**date:** Fri, 28 Aug 2020 20:09:37 GMT

**expires:** Thu, 19 Nov 1981 08:52:00 GMT

**pragma:** no-cache

**server:** LiteSpeed

**status:** 200

**vary:** Accept-Encoding

**x-powered-by:** PHP/7.4.8



# VAMOS VER NO NAVEGADOR ?

# MÃO NA MASSA

- Vamos ver isso  
no navegador





# VERBOS HTTP



**GET**

Recupera informações  
Listar/exibir



**POST**

Gravar/Inserir  
Login/Cadastros



**PUT**

Atualizar informações



**DELETE**

Excluir informações

# VERBOS HTTP

## GET

O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

## HEAD

O método HEAD solicita uma resposta de forma idêntica ao método **GET**, porém sem conter o corpo da resposta.

## POST

O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

# VERBOS HTTP

## PUT

O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

## DELETE

O método DELETE remove um recurso específico.

## CONNECT

O método CONNECT estabelece um túnel para o servidor identificado pelo recurso de destino.

# VERBOS HTTP

## OPTIONS

O método OPTIONS é usado para descrever as opções de comunicação com o recurso de destino.

## TRACE

O método TRACE executa um teste de chamada *loop-back* junto com o caminho para o recurso de destino.

## PATCH

O método PATCH é utilizado para aplicar modificações parciais em um recurso

# AS REQUISIÇÕES (RESPOSTAS)

- 2XX, 200, 203, 207 -> Respostas de Sucesso
- 3XX, 300, 301, 302 -> Mensagem de redirecionamento
- 4XX, 401, 404, 405 -> Respostas de erro do cliente
- 5XXX, 500, 502, 503 -> Respostas de erro do servidor



# STATUS HTTP

- <https://http.cat/>



**403**

Forbidden



**404**

Not Found



**405**

Method Not Allowed

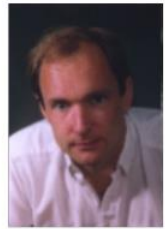
# PERGUNTAS ?

<https://github.com/rodrigofujioka/estaciodevweb>

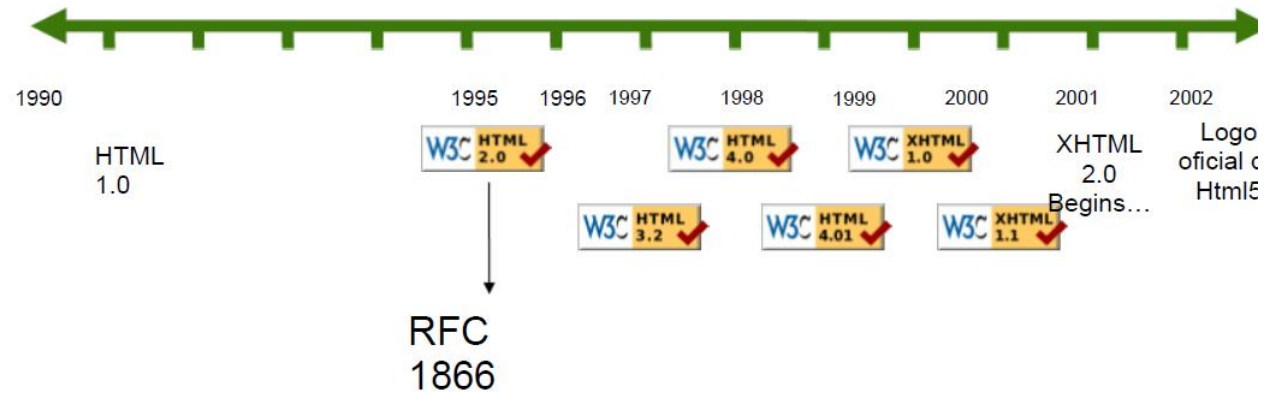


# HTML

## HTML Timeline



VS.



# HTML

- **HyperText Markup Language (HTML)** ou Linguagem de Marcação de Hipertexto é a linguagem usada pelos navegadores para mostrar as páginas web ao usuário.
- Ela permite combinar textos, imagens e áudios, além introduzir referências a outras páginas por meio dos links hipertextos.



1. Abrir o Navegador de sua preferência

2. Entrar em um site (digitar URL e teclar enter)

3. Clicar com o botão direito

4. Clicar na opção Exibir Código-fonte

5. Irá abrir uma janela com o código HTML


Faça esses passos em mais dois sites



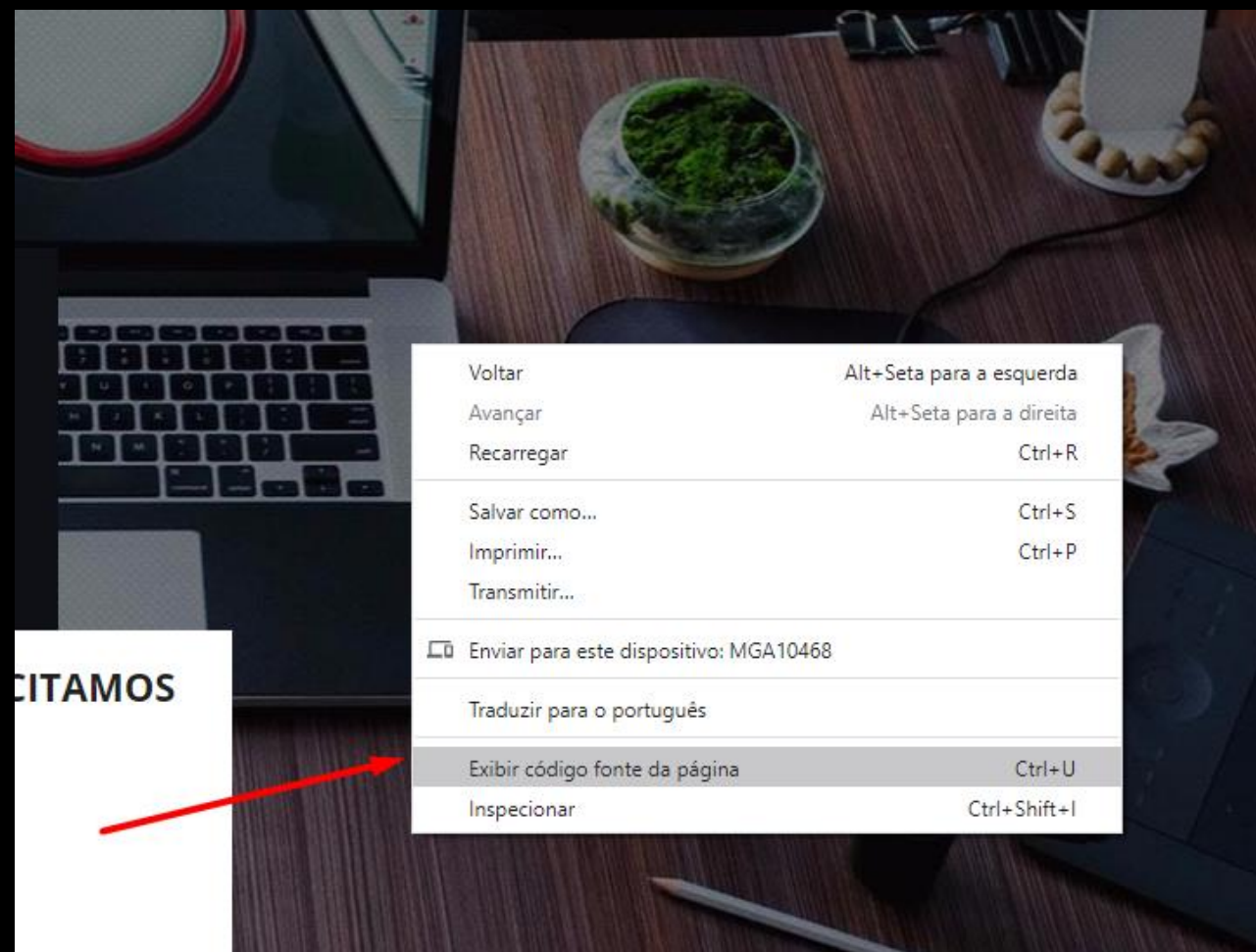
# HTML

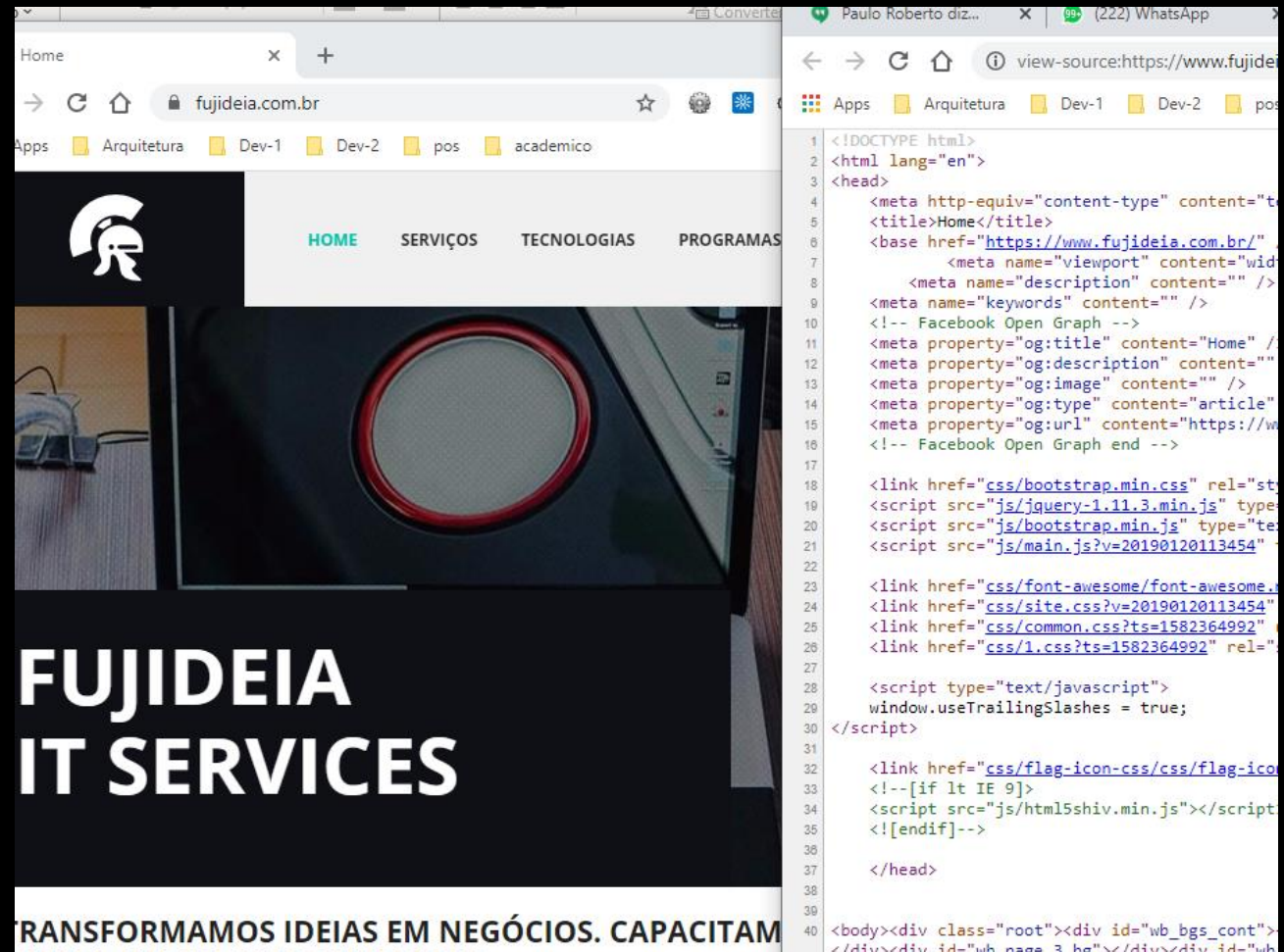
- Introdução
- Estrutura HTML5



- 
- Como escrevo uma página para a internet?
  - Estrutura básica de um documento HTML
  - Conceitos
  - Algumas formatações em HTML

- Como escrevo uma página para a internet?
- Estrutura básica de um documento HTML
- Conceitos
- Algumas formatações em HTML





# HELLO WORLD

Abrir o Visual Studio Code e digitar o seguinte texto:

```
<html>
<head>
</head>
  <body>
    Olá!
    MUNDO!!!
  <body>
</html>
```

Salvar na pasta htdocs com o nome exemplo1.html

- No Firefox, digitar:

`http://localhost/~seulogin/exemplo1.html`

- O que aparece?

 **XAMPP** Apache + MariaDB + PHP + Perl

### O que é o XAMPP?

XAMPP é o ambiente de desenvolvimento PHP mais popular

O XAMPP é completamente gratuito, de fácil de instalar a distribuição Apache, contendo MySQL, PHP e Perl. O pacote de código aberto do XAMPP foi criado para ser extremamente fácil de instalar e de usar.



Baixar  
Clique aqui para outras versões

 XAMPP para Windows  
7.4.9 (PHP 7.4.9)

 XAMPP para Linux  
7.4.9 (PHP 7.4.9)

 XAMPP para OS X  
7.4.9 (PHP 7.4.9)



# MÃO NA MASSA

- Configurar Ambiente.





# VAMOS ENTENDER

<html>, </html>, <head>, </head>, <body>, </body> são palavras chamadas tags

- Uma tag é um comando (instrução) de formatação
  - Não são exibidas pelo navegador
- Sempre são escritas entre < >
- Possuem seu respectivo elemento de fechamento

</ >

## TAGS, ELEMENTOS

- A tag tem a seguinte forma geral:

**<tag> . . . . . </tag>**

- Tudo que estiver contido entre uma tag de abertura <> e uma tag de fechamento </> será processado segundo o comando contido na tag.
- Algumas tags, chamadas tags de comandos isolados, não necessitam de um conteúdo para serem processados.

**<BR>**

- Tag para pular linha

# ELEMENTOS ESTRUTURAIS

Element	Description
<code>&lt;html&gt;&lt;/html&gt;</code>	Surrounds the entire page
<code>&lt;head&gt;&lt;/head&gt;</code>	Contains header information (metadata, CSS styles, JavaScript code)
<code>&lt;title&gt;&lt;/title&gt;</code>	Holds the page title normally displayed in the title bar and used in search results
<code>&lt;body&gt;&lt;/body&gt;</code>	Contains the main body text. All parts of the page normally visible are in the body

## A Tag **<b>**

- **<b>** indica ao navegador para escrever em negrito
- **</b>** indica onde deve-se terminar de aplicar a formatação de negrito

<html>

Indica que este documento possui uma estrutura HTML que se inicia aqui

</html>

E termina aqui

(Silva e Oliveira, 2012)

```
<html>
```

```
<head>
```

```
</head>
```

O cabeçalho (head ) é elemento fundamental de um documento HTML. O propósito é apresentar informações gerais sobre a página.

```
</html>
```

(Silva e Oliveira, 2012)



<html>

<head>

</head>

<body>

O corpo (body) é outro elemento fundamental. O conteúdo dentro de <body> será exibido pelo navegador.

<body>

</html>

(Silva e Oliveira, 2012)

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
  Olá!
```



Conteúdo

```
    Este é um <b>exemplo de código HTML.</b>
```

```
</body>
```

```
</html>
```

(Silva e Oliveira, 2012)

# AS TAGS <I>, <U> E <S>

- <i> exibe o texto em itálico
- <u> exibe o texto sublinhado
- <s> exibe o texto riscado

CRIE O SEU CURRÍCULO ONLINE COM  
AS INFORMAÇÕES QUE TEMOS ATÉ  
AGORA.

TIME 20 min.

# CSS

CSS



```
body {  
  font: x  
  backgrou  
  color: bl  
  margin: 0  
  padding: 0
```

# JAVASCRIPT

- Apenas para introduzir, vamos ver muito mais detalhes e uma contextualização muito mais aprofundada para entender melhor.



# JAVASCRIPT

- Variáveis, dados e operações
- Funções
- Constantes
- Operadores Lógicos e Condicionais
- Laços de repetição
- TypeScript e Node.JS

# VAMOS VER ?



## JavaScript Demo: Expressions - Spread syntax

```
1 function sum(x, y, z) {  
2   return x + y + z;  
3 }  
4  
5 const numbers = [1, 2, 3];  
6  
7 console.log(sum(...numbers));  
8 // expected output: 6  
9  
10 console.log(sum.apply(null, numbers));  
11 // expected output: 6  
12
```

(Monzila, 2020)

# VARIÁVEIS

- Não tem tipagem estática.
  - Você não precisa definir o tipo como em Java, C#, etc.
    - `Integer valor = 8; // Essa variável armazena apenas valores inteiros.`
- Tipagem dinâmica.

# VARIÁVEIS

```
file Edit Selection View Go Run Terminal Help variaveis.html - frameworkswb - Visual Studio Code

EXPLORER
> OPEN EDITORS
✓ FRAMEWORKSWEB
  > atendimentoapi
  > expressjs
  ✓ javascript-intro
    <> variaveis.html U
  > pdf
  > praticas
  .gitignore
  LICENSE
  README.md

variaveis.html
javascript-intro > <> variaveis.html > html > body > script
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <p id="aulafuji"></p>
5   <script>
6     var codigo = 202009101;
7     var nome = "Frameworks Web";
8     var opcional = false;
9     var semestre = 2020.2;
10    var cargaHoraria = 80;
11
12    var disciplinas = ['Frameworks Web', 'Express', 'Mobile'];
13
14    var disciplina = {
15      codigo : 202009101,
16      nome : "Frameworks Web",
17      opcional : false,
18      semestre : 2020.2,
19      cargaHoraria: 80,
20
21    };
22    console.log(disciplina);
23    console.log(disciplinas);
24    document.getElementById("aulafuji").innerHTML = nome;
25  </script>
26
27 </body>
28 </html>
29
```

variaveis.html

Arquivo | D:/ambiente/siste... ☆ (≡) C: R O Outros favoritos

Apps Arquitetura Dev-1 Dev-2 pos

Frameworks Web

Elements Console Sources >> ⚙ ⋮ ✕

top 🔍 Filter Default levels ⚙

variaveis.html:22

```
{codigo: 202009101, nome: "Frameworks Web", opcional: false, semestre: 2020.2, cargaHoraria: 80}
```

cargaHoraria: 80  
codigo: 202009101  
nome: "Frameworks Web"  
opcional: false  
semestre: 2020.2  
▶ \_\_proto\_\_: Object

variaveis.html:23

```
(3) ["Frameworks Web", "Express", "Mobile"]
```

0: "Frameworks Web"  
1: "Express"  
2: "Mobile"  
length: 3  
▶ \_\_proto\_\_: Array(0)

>

javascript-intro > <> operacoes.html > html > body > script

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p id="aulafuji"></p>
5          <script>
6              var anoNascimento = 1982;
7              var anoAtual = 2020;
8              var idade = anoAtual - anoNascimento;
9              var anoPassado = anoAtual-1;
10             var anoFuturo = anoAtual + 1;
11
12             //anoatual += 1;  2021
13             //anoatual -= 1;  2019;
14
15             console.log(anoFuturo);
16         </script>
17
18 </body>
19 </html>
```

# OPERADORES

# FUNÇÕES

- Escopo das variáveis;
- Chamada.

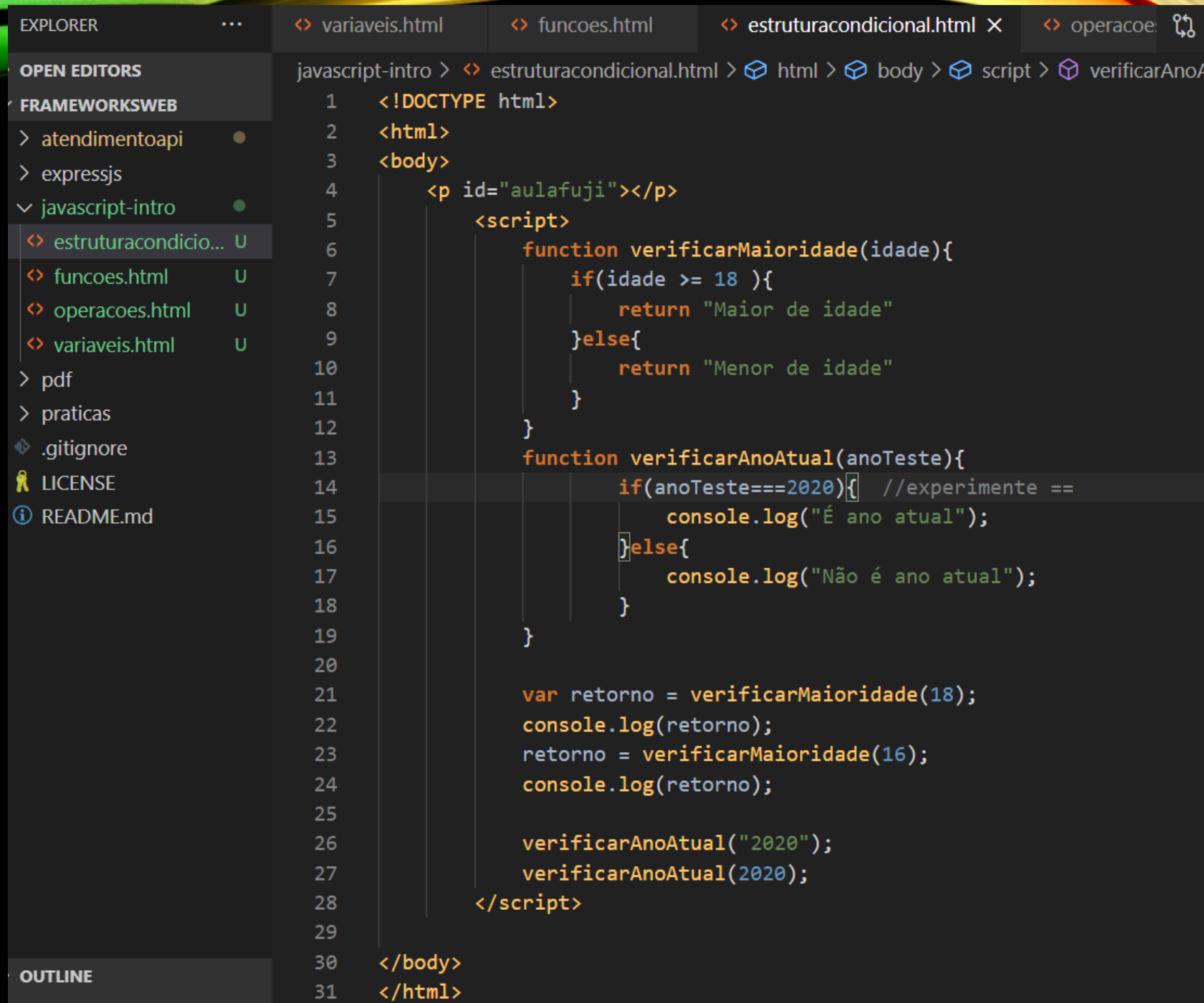
javascript-intro > <> funcoes.html > html > body > script >

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p id="aulafuji"></p>
5      <script>
6          function somar(valorA, valorB){
7              var soma = valorA + valorB;
8              return soma;
9          }
10
11         function exibirNome(nome){
12             window.alert(nome);
13         }
14
15         var soma = somar(4,5);
16         console.log(soma);
17         exibirNome("Rodrigo Fujioka");
18
19     </script>
20
21 </body>
22 </html>
```



# CONDICIONAIS

- IF , ELSE



```
EXPLORER  ...  variaveis.html  funcoes.html  estruturacondicional.html x  operacoes.html  verificarAnoA

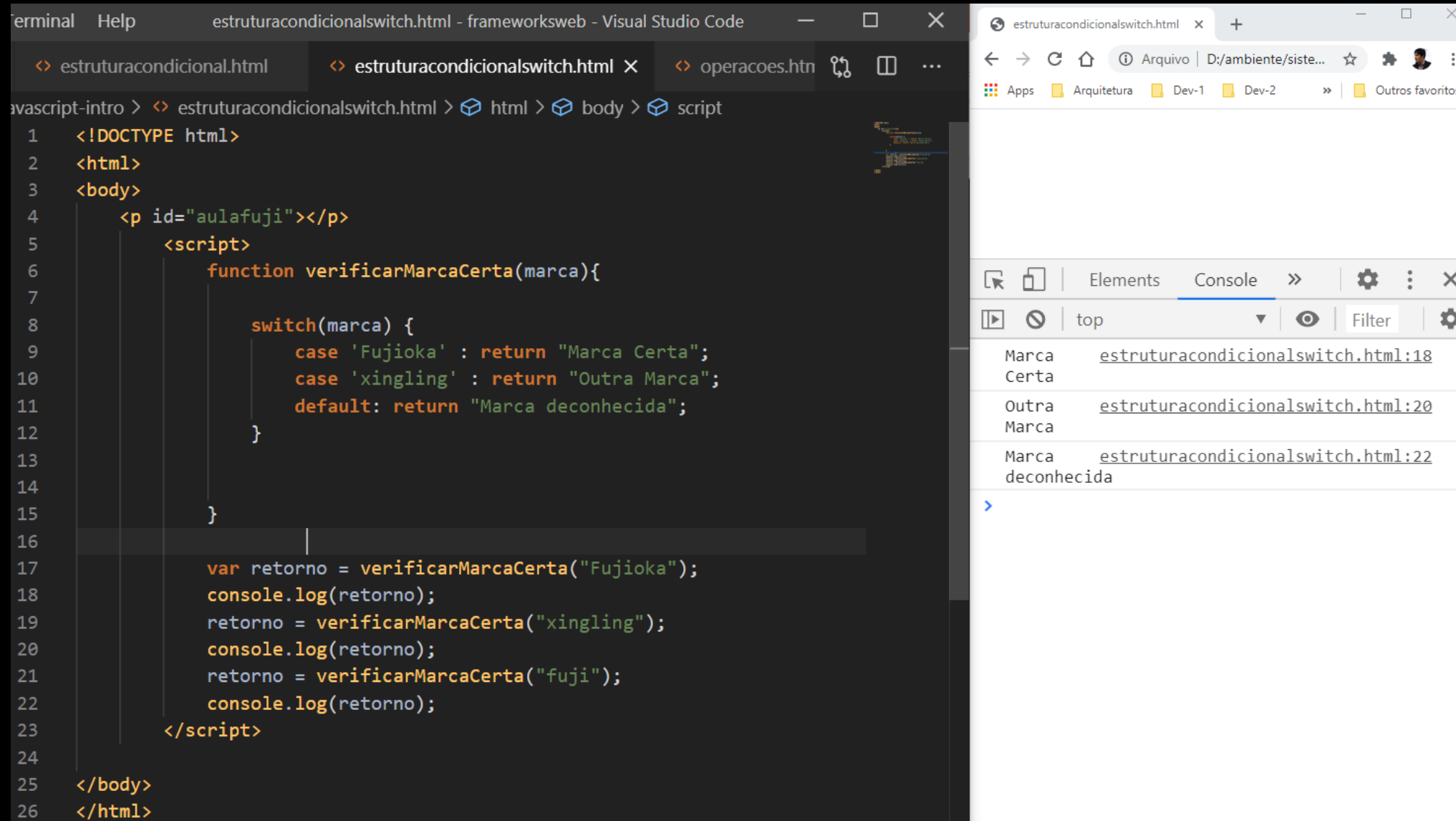
OPEN EDITORS
FRAMEWORKSWEB
  > atendimentoapi
  > expressjs
  v javascript-intro
    <> estruturacondicio... U
    <> funcoes.html U
    <> operacoes.html U
    <> variaveis.html U
  > pdf
  > praticas
  .gitignore
  LICENSE
  README.md

OUTLINE

javascript-intro > <> estruturacondicional.html > html > body > script > verificarAnoA
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p id="aulafuji"></p>
5      <script>
6          function verificarMaioridade(idade){
7              if(idade >= 18 ){
8                  return "Maior de idade"
9              }else{
10                 return "Menor de idade"
11             }
12         }
13         function verificarAnoAtual(anoTeste){
14             if(anoTeste===2020){ //experimente ==
15                 console.log("É ano atual");
16             }else{
17                 console.log("Não é ano atual");
18             }
19         }
20
21         var retorno = verificarMaioridade(18);
22         console.log(retorno);
23         retorno = verificarMaioridade(16);
24         console.log(retorno);
25
26         verificarAnoAtual("2020");
27         verificarAnoAtual(2020);
28     </script>
29
30 </body>
31 </html>
```

# CONDICIONAIS

- Switch



The image shows a development environment with Visual Studio Code on the left and a web browser on the right. The VS Code editor displays a file named `estruturacondicionalswitch.html` with the following code:

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p id="aulafuji"></p>
5      <script>
6          function verificarMarcaCerta(marca){
7
8              switch(marca) {
9                  case 'Fujioka' : return "Marca Certa";
10                 case 'xingling' : return "Outra Marca";
11                 default: return "Marca desconhecida";
12             }
13
14         }
15
16
17         var retorno = verificarMarcaCerta("Fujioka");
18         console.log(retorno);
19         retorno = verificarMarcaCerta("xingling");
20         console.log(retorno);
21         retorno = verificarMarcaCerta("fuji");
22         console.log(retorno);
23     </script>
24
25 </body>
26 </html>
```

The web browser on the right shows the rendered page. The console is open, displaying the output of the JavaScript code:

Log Entry	Source
Marca Certa	<a href="#">estruturacondicionalswitch.html:18</a>
Outra Marca	<a href="#">estruturacondicionalswitch.html:20</a>
Marca desconhecida	<a href="#">estruturacondicionalswitch.html:22</a>

# OPERADORES

- && (E)
- || (OU)
- ! (NOT)

```
avascript-intro > <> estruturacondicionalswitch.html > ...
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p id="aulafuji"></p>
5          <script>
6              function verificarMarcaCerta(marca){
7
8                  if(marca === 'Fujioka' && marca === 'Fuji'){
9                      return "Marca certa"
10                 }
11
12                 if(marca !== 'xingling' ){
13                     return "Outra Marca"
14                 }else {
15                     return "Marca XingLing"
16                 }
17
18             }
19             var retorno = verificarMarcaCerta("Fujioka");
20             console.log(retorno);
21             retorno = verificarMarcaCerta("xingling");
22             console.log(retorno);
23             retorno = verificarMarcaCerta("Fuji");
24             console.log(retorno);
25             retorno = verificarMarcaCerta("TainheilenGambielMeloOka");
26             console.log(retorno);
27         </script>
28
29     </body>
30 </html>
```

# LAÇOS DE REPETIÇÃO

- setInterval(função, milisegundos);
- setTimeout(função, milisegundos);

```
javascript-intro > <> intervalor_timeout.html > html > body > script
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p id="aulafuji"></p>
5      <script>
6
7          function exibeData(){
8              console.log(new Date())
9          }
10
11          //setInterval(exibeData, 1000);
12          setInterval(() => {
13              exibeData();
14          }, 1000);
15          //setTimeout(exibeData, 1000);
16          setTimeout(() => {
17              exibeData();
18          }, 5000);
19
20      </script>
21
22  </body>
23  </html>
```

# EXERCÍCIO

- 1 – Crie em javascript uma calculadora.
- 2 – Crie em uma função para exibir todos os números pares em um intervalo de números ex entre 1 e 100.
- 3 – Crie uma função que verifique o sexo e a idade do usuário.
  - A -> Se Sexo Masculino e Idade  $\geq 18$  pode entrar no Bar com taxa.
  - A2 -> Se Sexo Masculino e Idade  $< 18$  não pode entrar no Bar.
  - B -> Se sexo Feminino e Idade  $\geq 18$  pode entrar no bar sem taxa.
  - B2 -> Se sexo Feminino e Idade  $< 18$  Não pode entrar no bar.

# VAMOS PREPARAR O AMBIENTE ?

## XAMPP Apache + MariaDB + PHP + Perl

### O que é o XAMPP?


XAMPP é o ambiente de desenvolvimento PHP mais popular


O XAMPP é completamente gratuito, de fácil de instalar a distribuição Apache, contendo MySQL, PHP e Perl. O pacote de código aberto do XAMPP foi criado para ser extremamente fácil de instalar e de usar.




**Baixar**

Clique aqui para outras versões

 XAMPP para  
**Windows**  
7.4.9 (PHP 7.4.9)

 XAMPP para **Linux**  
7.4.9 (PHP 7.4.9)

 XAMPP para **OS X**  
7.4.9 (PHP 7.4.9)

[https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)



# PERGUNTAS ?

<https://github.com/rodrigofujioka/estaciodevweb>



## REFERÊNCIAS

- GITHUB. <https://github.com/rodrigofujioka/demo-quarkus>
- DevMedia. Guia Angular. 2020. Disponível em <https://www.devmedia.com.br/angular/>.
- Express.js <https://expressjs.com/pt-br/>