



MINICURSO

Criando Websites Dinâmicos com Node.js

Vitor Bruno de Oliveira Barth INSTRUTOR
André Geraldo Guimarães Pinto MONITOR
Yuri Willians dos Santos MONITOR

Prof. Dr. Ruy de Oliveira ORIENTADOR

OBJETIVO DO MINICURSO

Ensinar a teoria e a prática do desenvolvimento de aplicações web dinâmicas utilizando Node.js

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

CRONOGRAMA

Dia 1 – Criação de APIs com Node.js

TEORIA

Estrutura de Aplicações Web

Conceitos de API REST

Node.js e Express.js

LABORATÓRIO

Trabalhar com APIs

Dia 2 – Criação de Páginas Web

TEORIA

Revisão do Dia 1

Estrutura Básica do Bootstrap

Requisições HTTP com Fetch

LABORATÓRIO

Criar Páginas Dinâmicas

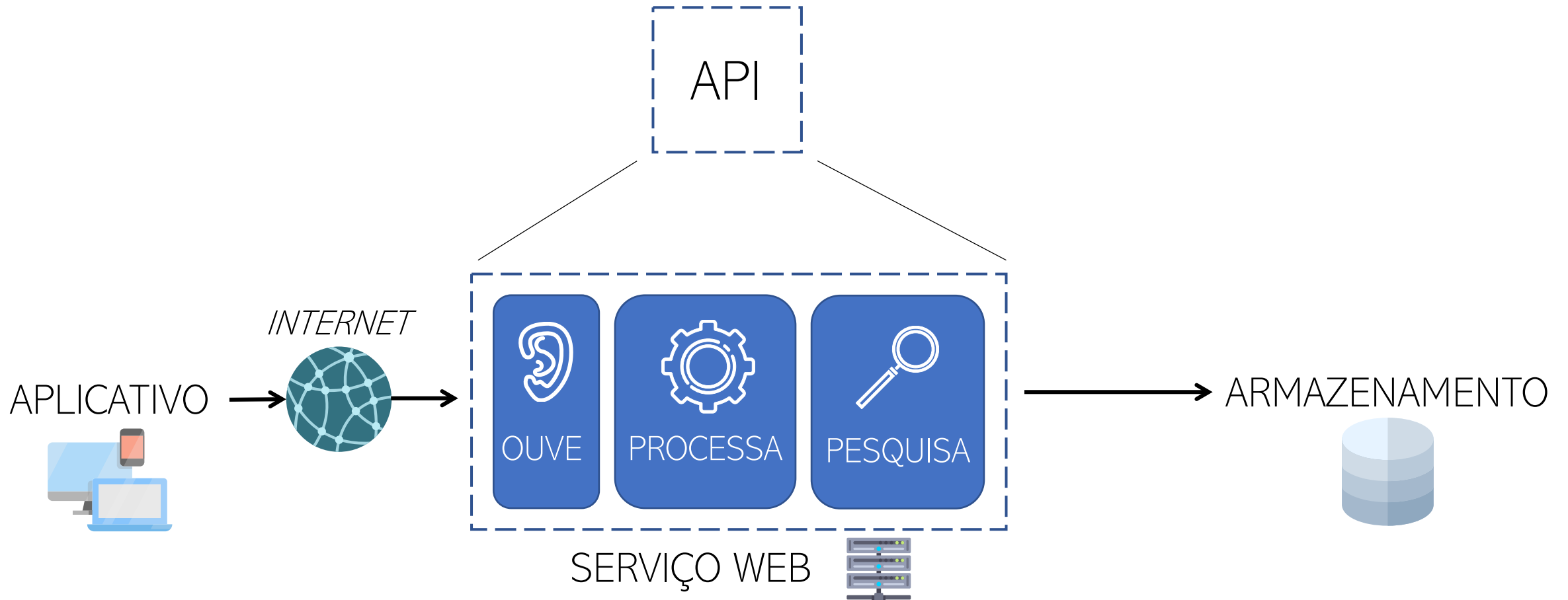
REVISÃO DO DIA 1

Estrutura de Aplicações Web, APIs e Node.js

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

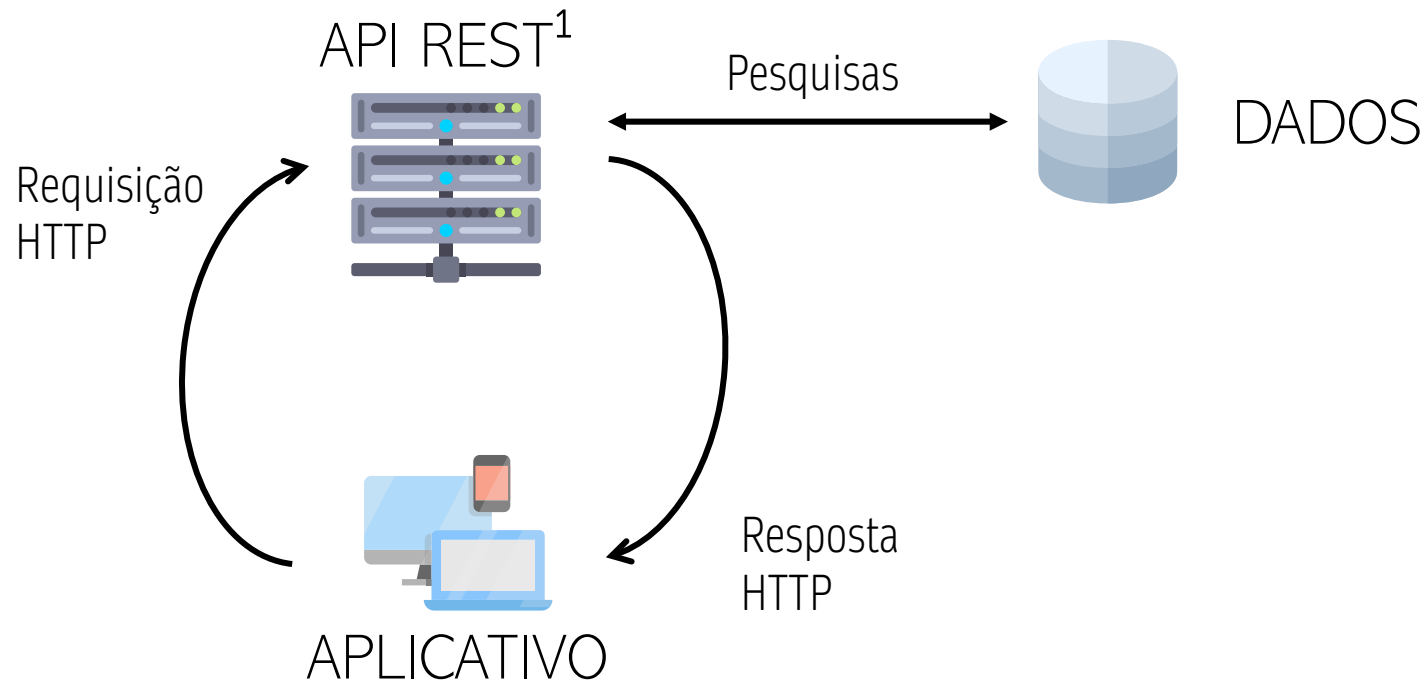
REVISÃO DO DIA 1

Estrutura de Aplicações Web



REVISÃO DO DIA 1

Conceitos de API REST



O padrão REST institui diversos métodos, destacando-se:

- > **GET** para buscar
- > **POST** para salvar
- > **PUT** para sobrescrever
- > **PATCH** para editar
- > **DELETE** para remover

¹ REST é acrônimo para *Representational State Transfer* (Transferência Representacional de Estado), o qual define uma linguagem padrão de comunicação entre serviços web.

REVISÃO DO DIA 1

Criando APIs com Node.js e Express

```
index.js

let express = require('express')
let app = express()

app.listen(8080, function (err) {
  if (err) throw err
  console.log('Servidor ouvindo em http://localhost:8080')
})

app.get('/', function (req, res) {
  res.send('Semana da Informática IFMT 2019')
})
```

TÓPICO 4

Estrutura básica de sites com Bootstrap

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

TÓPICO 4

Estrutura básica de sites com Bootstrap



Bootstrap

- É um framework para facilitar o desenvolvimento de front-ends
- Altamente customizável
- Possui diversos componentes (botões, pop-ups, menus...) pré-definidos
- Possui um sistema de grid para facilitar a organização dos elementos na tela

TÓPICO 4

Estrutura básica de sites com Bootstrap

Importar o Bootstrap

```
1 <html>
2
3 <head>
4   <title>MINICURSO | Criando Sites Dinâmicos com Node.js</title>
5   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
6     crossorigin="anonymous">
7 </head>
8
9 <body>
10 </body>
11
12 <footer>
13   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" crossorigin="anonymous"></script>
14   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" crossorigin="anonymous"></script>
15   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" crossorigin="anonymous"></script>
16 </footer>
```

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

TÓPICO 4

Estrutura básica de sites com Bootstrap

Organização das Páginas

```
9  <body>
10  <div class="container-fluid">
11  <div class="row"> <!-- Cada row (linha) possui tamanho 12 -->
12  <div class="col-4"> <!-- Cada col (coluna) possui largura variável -->
13  |   COLUNA 1
14  |
15  |
16  <div class="col-4">
17  |   COLUNA 2
18  |
19  |
20  <div class="col-4"> <!-- Para melhor resultado, a soma dos tamanhos das colunas deve ser 12 -->
21  |   COLUNA 3
22  |
23  |
24  </div>
    </div>
  </div>
</body>
```

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

TÓPICO 4

Estrutura básica de sites com Bootstrap

Container



```
<div class="container">  
  <!-- Conteúdo aqui -->  
</div>
```

Copy

Container-Fluid



```
<div class="container-fluid">  
  ...  
</div>
```

Copy

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

TÓPICO 4

Estrutura básica de sites com Bootstrap

Componentes

Oloco, meu! Olha esse alerta animado, como é chique! ×

```
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  <strong>Oloco, meu!</strong> Olha esse alerta animado, como é chique!
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
```

Copy

Mais informações na página de documentação do Bootstrap: <https://getbootstrap.com.br>

TÓPICO 4

Estrutura básica de sites com Bootstrap

Componentes



The image displays a collection of Bootstrap button styles and their corresponding HTML code. At the top, there is a row of buttons: Primary (blue), Secondary (gray), Success (green), Danger (red), Warning (yellow), Info (teal), Light (light gray), Dark (dark gray), and Link (blue text). Below this row, the HTML code for each button is listed in a light gray box. A 'Copy' button is located in the top right corner of the code box.

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Mais informações na página de documentação do Bootstrap: <https://getbootstrap.com.br>

TÓPICO 4

Estrutura básica de sites com Bootstrap

Componentes

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="Nome">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Sobrenome">
    </div>
  </div>
</form>
```

Copy

Mais informações na página de documentação do Bootstrap: <https://getbootstrap.com.br>

TÓPICO 4

Estrutura básica de sites com Bootstrap

Componentes

Navbar Home Destaques Preços Desativado

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar!
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-item nav-link active" href="#">Home <span class="sr-only">(Página atua
      <a class="nav-item nav-link" href="#">Destaques</a>
      <a class="nav-item nav-link" href="#">Preços</a>
      <a class="nav-item nav-link disabled" href="#">Desativado</a>
    </div>
  </div>
</nav>
```

Copy

Mais informações na página de documentação do Bootstrap: <https://getbootstrap.com.br>

TÓPICO 5

Requisições HTTP com Fetch

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

TÓPICO 5

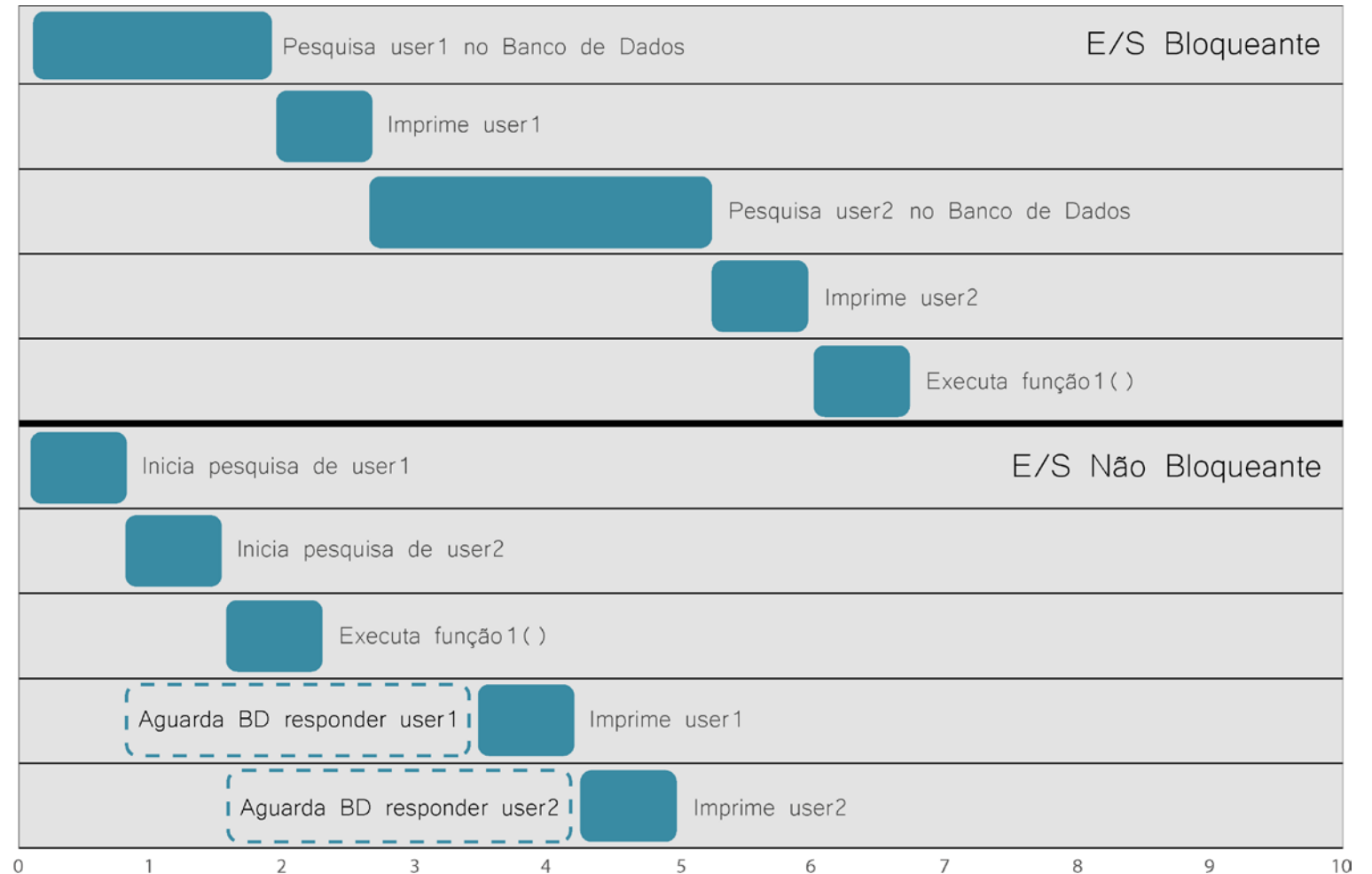
Requisições HTTP com Fetch

Pseudocódigo:

1. Busca Usuário 1 (U1)
2. Imprime U1
3. Busca Usuário 2 (U2)
4. Imprime U2
5. Executa Função1()
(que não depende de U1 e U2)

E/S Bloqueante: C, C++, Java, Python, ...

E/S Não-Bloqueante: JavaScript, Go



TÓPICO 5

Requisições HTTP com Fetch

Estrutura de uma Promise

Promises garantem a execução síncrona do código

```
1  fetch('/alunos')
2    .then(async listaAlunos => {
3      |    return(await listaAlunos.json())
4    })
5    .catch(error => {
6      |    console.error(error)
7    })
```

TÓPICO 5

Requisições HTTP com Fetch

Estrutura de uma Promise

Promises garantem a execução síncrona do código

```
1  fetch('/alunos') 1. BUSCA ALUNOS
2    .then(async listaAlunos => {
3      |    return(await listaAlunos.json())
4    })
5    .catch(error => {
6      |    console.error(error)
7    })
```

TÓPICO 5

Requisições HTTP com Fetch

Estrutura de uma Promise

Promises garantem a execução síncrona do código

```
1  fetch('/alunos') 2. RECEBEU RESPOSTA OK
2    .then(async listaAlunos => {
3      |   return(await listaAlunos.json())
4      | })
5    .catch(error => {
6      |   console.error(error)
7      | })
```

TÓPICO 5

Requisições HTTP com Fetch

Estrutura de uma Promise

Promises garantem a execução síncrona do código

```
1  fetch('/alunos')
2    .then(async listaAlunos => {
3      |    return(await listaAlunos.json())
4    })
5    .catch(error => {
6      |    console.error(error)
7    })
```

3. RECEBEU ERRO

TÓPICO 5

Requisições HTTP com Fetch

Requisições POST usando Fetch

```
fetch('/aluno',  
  {  
    headers: {  
      'Accept': 'application/json',  
      'Content-Type': 'application/json'  
    },  
    method: 'POST',  
    body: JSON.stringify(aluno)  
  })  
  .then(async res => {  
    console.log(await res.json())  
  })
```

INTERVALO

15 MINUTOS

TÓPICO 6

Reconhecendo eventos e Alterando o HTML com JavaScript

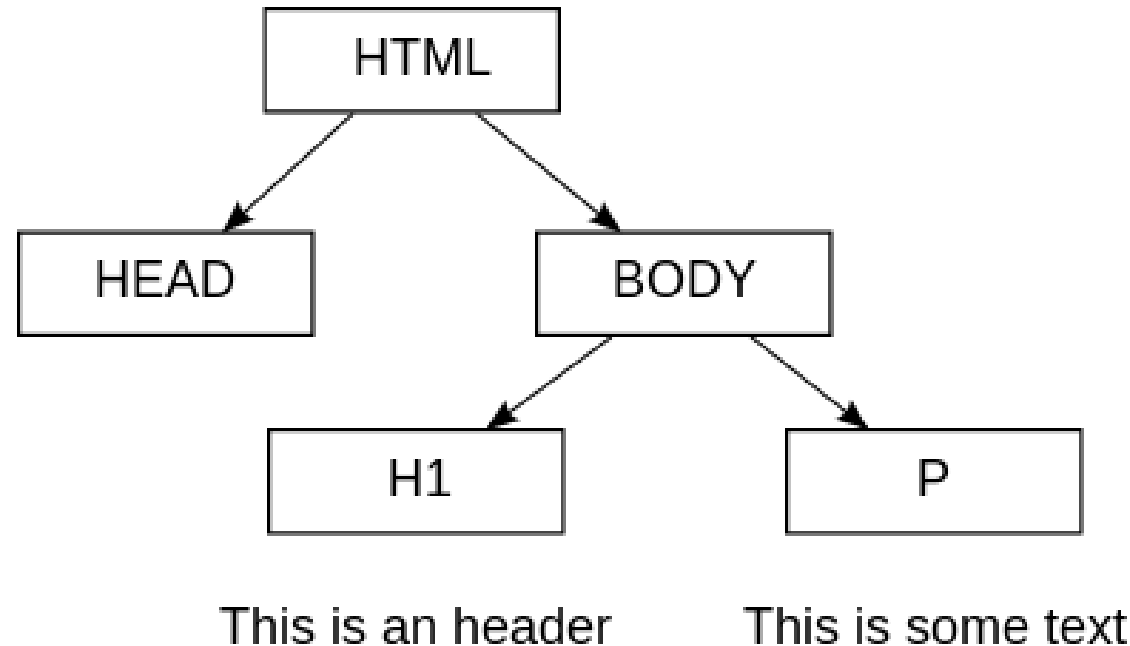
P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

TÓPICO 6

Reconhecendo eventos e Alterando o HTML com JavaScript

DOM

Páginas HTML seguem uma hierarquia, chamada DOM (Document Object Model)



TÓPICO 6

Reconhecendo eventos e Alterando o HTML com JavaScript

DOM

Cada elemento do DOM, quando exibido pelo navegador, possui propriedades. Ex.:

- **innerHTML**: o HTML interno a ele
- **value**: valor (para itens alteráveis, como caixas de texto)
- **classList**: lista de classes
- **id**: identificador

Estas propriedades podem ser alteradas via JavaScript

TÓPICO 6

Reconhecendo eventos e Alterando o HTML com JavaScript

index.html

```
<body>
  <div class="container-fluid">
    <div class="row" id="linha1">
      <div class="col-4">
        COLUNA 1
      </div>
    </div>

    <script src="./alterar.js"></script>
  </body>
```

alterar.js

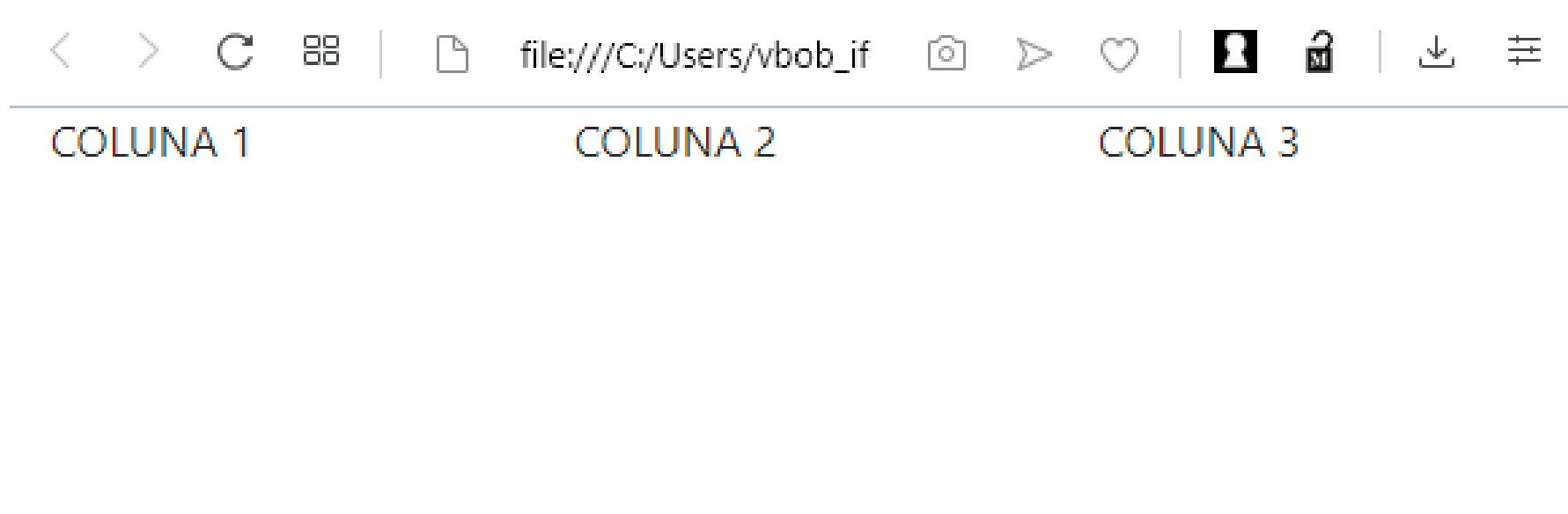
```
let container = document.getElementById('linha1')

container.innerHTML += `
  <div class="col-4">
    COLUNA 2
  </div>
`

container.innerHTML += `
  <div class="col-4">
    COLUNA 3
  </div>
`
```

TÓPICO 6

Reconhecendo eventos e Alterando o HTML com JavaScript



ATIVIDADE 2

Gerar Páginas Dinâmicas com Dados da API

P.S.: Todo o material utilizado está disponível em <http://pastebin.com/PFawQgtc>

ATIVIDADE 2

Gerar Páginas Dinâmicas com Dados da API

IFMT [Início](#) [Cadastrar Aluno](#)

id	Nome	Idade	Endereço	E-Mail	Opções
1	Vitor Barth	21	Rua Zulmira Canavarros, 95	vitor.barth@gmail.com	Editar Apagar
2	André da Silva	33	Rua Zulmira Canavarros, 95	andre.silva@gmail.com	Editar Apagar

ATIVIDADE 2

Gerar Páginas Dinâmicas com Dados da API

IFMT [Início](#) [Cadastrar Aluno](#)

Cadastro de Novo Aluno

Nome

Idade

Endereço

E-mail

Cadastrar

ATIVIDADE 2

Gerar Páginas Dinâmicas com Dados da API

IFMT [Início](#) [Cadastrar Aluno](#)

Remover Aluno

ID	Nome	Idade
1	Vitor Barth	21

Endereço

Rua Zulmira Canavarros, 95

E-mail

vitor.barth@gmail.com

Apagar

ATIVIDADE 2

Gerar Páginas Dinâmicas com Dados da API

IFMT [Início](#) [Cadastrar Aluno](#)

Editar Aluno

ID	Nome	Idade
1	Vitor Barth	21

Endereço

Rua Zulmira Canavarros, 95

E-mail

vitor.barth@gmail.com

Salvar

Todo o material utilizado está disponível em
<http://pastebin.com/PFawQgtc>

FIM DO MINICURSO

Obrigado pela presença!

Realização



Agradecimentos



Contato

André Guimarães – andregp019@gmail.com

Yuri Santos – yuriduli12@gmail.com

Vitor Barth – vitor.barth@gmail.com

Prof. Dr. Ruy de Oliveira – ruy@cba.ifmt.edu.br