Francisco Caramalho nº30245 versão B

Parte I

Esta imagem representa um diagrama de arquitetura de software que ilustra a interação entre uma aplicação web e móvel com um serviço web e uma base de dados. As aplicações, desenvolvidas em HTML5/JavaScript para web e Java/Swift para dispositivos móveis, comunicam-se com o serviço web utilizando JSON. O serviço web, implementado com Node.js ou PHP, processa e interage com a base de dados MySQL para armazenar informações. Este diagrama faz uma troca de dados em formato JSON entre as aplicações e o serviço web, enfatizando a estrutura de uma arquitetura de três camadas composta por cliente, servidor e base de dados.

Parte II

1. A diferença entre o getElement e o getEements é o que cada um retorna. Enquanto o getElement só retorna 1 dado, o getElements retorna um conjunto de dados.

Usando o document.getElemenById(Id):

const element = document.getElementById('meuElemento'); // Seleciona o elemento com id "meuElemento" if (element) { console.log(element.textContent); // Exibe o conteúdo de texto do elemento } else { console.log('Elemento não encontrado');

Usando document.getElementsByTagName(tagName):

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<script>
  var itens = document.getElementsByTagName("li");
  for (var i = 0; i < itens.length; i++) {
    console.log(itens[i].textContent); // Saída: Item 1, Item 2
  }
</script>
```

Usando document.getElementsByClassName(className):

```
<div class="minhaClasse">Elemento 1</div>
<div class="minhaClasse">Elemento 2</div>
<script>
  var elementos = document.getElementsByClassName("minhaClasse");
  for (var i = 0; i < elementos.length; i++) {
    console.log(elementos[i].textContent); // Saída: Elemento 1, Elemento 2
  }
</script>
```

2. Vscode. parteII.js e parteII.xml

Parte III

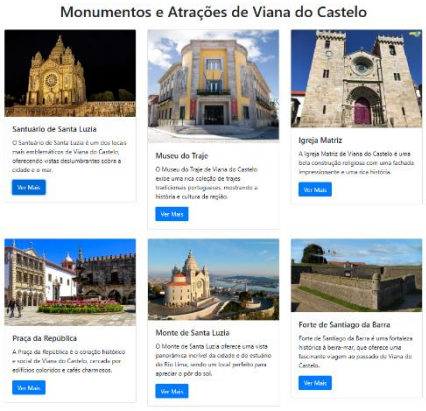
1. A diferença é que o <p> é para definir um parágrafo e o <pre> é para definir um texto pre-formatado.
2. O <meta charset="utf-8"> serve para definir um grupo de caracteres que o HTML vai utilizar.

Parte IV

1. Vscode. parteIV.css e parteIV.html

Parte V

1. Vscode. ParteV.css e parteV.html



Parte VI

1. Vscode. Controllers_products.js
2.

const express = require('express'); - Esta linha importa o módulo Express.js para o ficheiro.

const router = express.Router(); - Aqui cria uma nova instância do Router do Express. const productController = require('../controllers/productController'); - Esta linha importa o controlador de produtos para o seu ficheiro de

router.get('/produtos/:id', productController.getProduct); - Esta linha define uma rota GET para '/produtos/:id'. Quando um cliente faz uma solicitação GET para '/produtos/:id' (onde :id é o ID de um produto), o método getProduct do controlador de produtos é chamado.

router.post('/produtos', productController.createProduct); - Esta linha define uma rota POST para '/produtos'. Quando um cliente faz uma solicitação POST para '/produtos', o método createProduct do controlador de produtos é chamado.

router.put('/produtos/:id', productController.updateProduct); - Esta linha define uma rota PUT para '/produtos/:id'. Quando um cliente faz uma solicitação PUT para '/produtos/:id', o método updateProduct do controlador de produtos é chamado.

router.delete('/produtos/:id', productController.deleteProduct); - Esta linha define uma rota DELETE para '/produtos/:id'. Quando um cliente faz uma solicitação DELETE para '/produtos/:id', o método deleteProduct do controlador de produtos é chamado.

module.exports = router; - Finalmente, esta linha exporta o router para que possa ser usado em outros ficheiros.

```
const productRouter = require('express').Router();
const controller = require('../controllers/products');
const authMiddleware = require ('../middlewares/auth/auth');

// Importar o controlador de produtos
const productController = require('../controllers/productController')

// Rota para obter todos os produtos
router.get('/produtos', productController.getAllProducts);

// Rota para obter um produto específico
router.get('/produtos/:id', productController.getProduct);

// Rota para criar um novo produto
router.post('/produtos', productController.createProduct);

// Rota para atualizar um produto
router.put('/produtos/:id', productController.updateProduct);

// Rota para deletar um produto
router.delete('/produtos/:id', productController.deleteProduct);

module.exports = router;
```