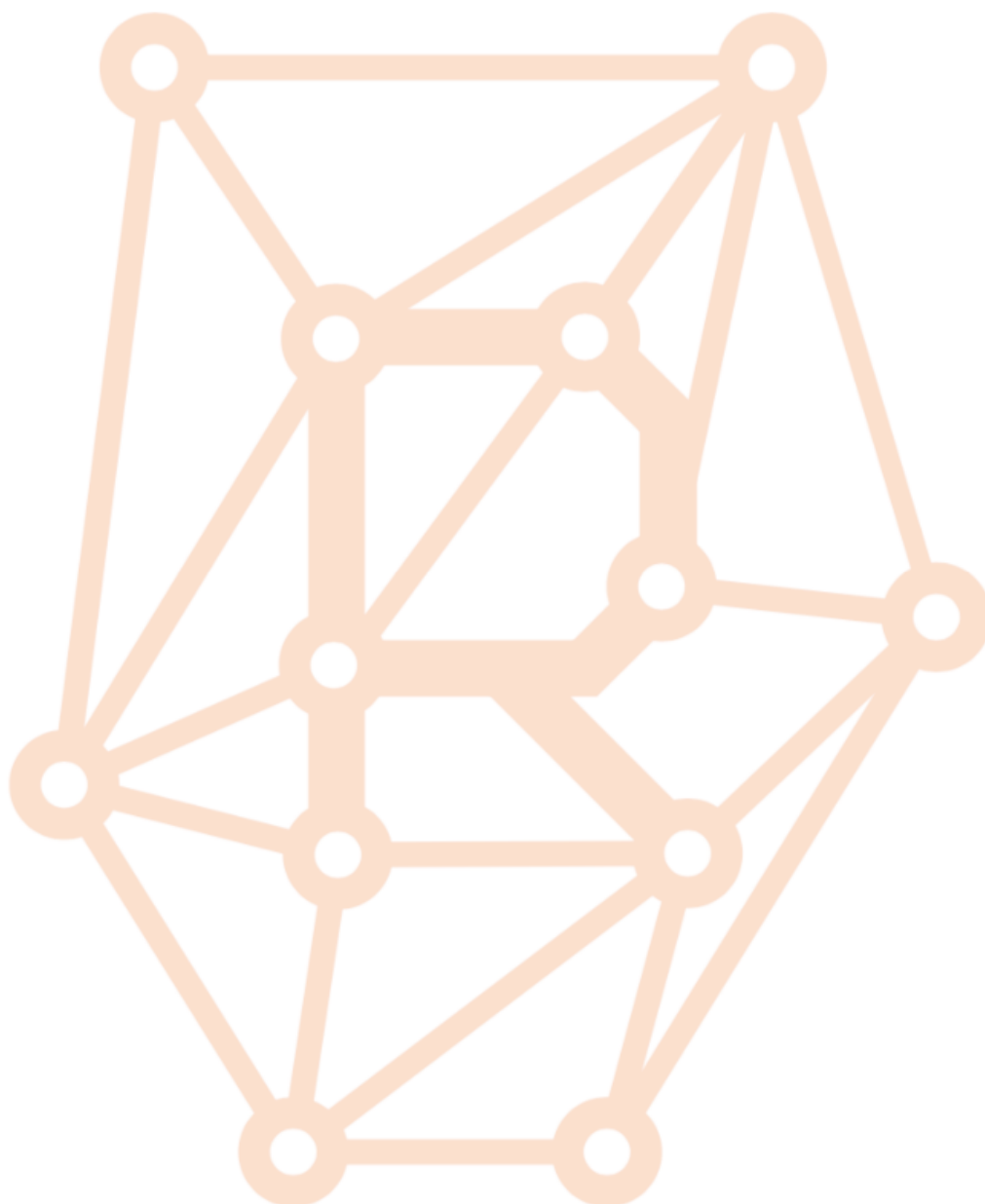


# Engenharia de Redes e Sistemas de Computadores Programação Web



Tomás Ferreira 30239

João Carvalhosa 30565

## Índice

Índice .....	2
Introdução .....	3
Modelo do sistema.....	4
Estrutura do Site .....	5
Esboço das páginas.....	6
Base de Dados .....	7
Páginas web.....	8
Código JavaScript.....	9
Evolução .....	10
Conclusão .....	11
Referências .....	12

## Introdução

No âmbito da disciplina de Programação Web, este projeto visa desenvolver uma aplicação que abrange diversas funcionalidades tendo uma zona pública e uma zona privada separadas por uma autenticação. Utilizando HTML, Bootstrap, CSS, JavaScript e um ficheiro JSON local, pretendemos criar uma plataforma que cumpra os requisitos estabelecidos.

Na zona pública, os utilizadores terão acesso a uma página de entrada que servirá como ponto de partida para a navegação na aplicação. Além disso, será possível realizar a criação de conta e iniciar sessão, garantindo assim a interação básica dos utilizadores com o sistema. Uma das funcionalidades importantes é a capacidade de ler dados de um arquivo JSON local para carregar uma página, dando uma abordagem mais realista ao problema.

Por outro lado, na zona privada, os utilizadores terão acesso a funcionalidades mais avançadas. Será possível gerir o perfil do utilizador, onde este poderá atualizar informações pessoais conforme necessário. Além disso, será implementado um CRUD (Create, Read, Update, Delete) que estará alinhado com o nosso tema, permitindo a manipulação de dados conforme as necessidades do utilizador. Nesta zona privada vamos ter dois tipos de utilizadores um “admin” com poderes para ler, criar, atualizar e apagar dados e um “guest” que só poderá ler os mesmos.

O tema escolhido para este trabalho foi “Hinoportuna Inventory” e pretendemos implementar um sistema de inventário dos vários ativos que a tuna académica do IPVC possui de forma a simplificar a contagem e gestão dos mesmos.

## Modelo do sistema

Neste modelo, apresentamos uma abordagem simplificada do que pretendemos implementar no nosso website. Um utilizador da Internet terá a capacidade de autenticar-se ou criar uma conta, caso ainda não possua uma. Posteriormente, terá acesso à gestão do seu perfil, à funcionalidade de logout e à possibilidade de criar, ler, atualizar e eliminar dados relacionados com um inventário, neste caso específico.

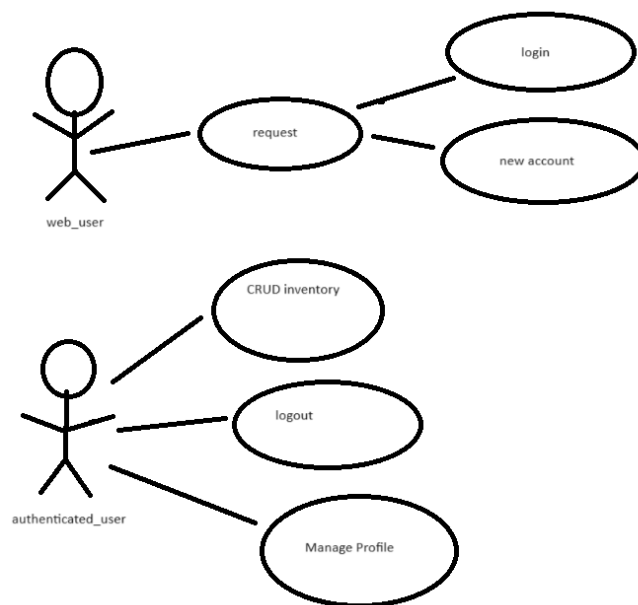
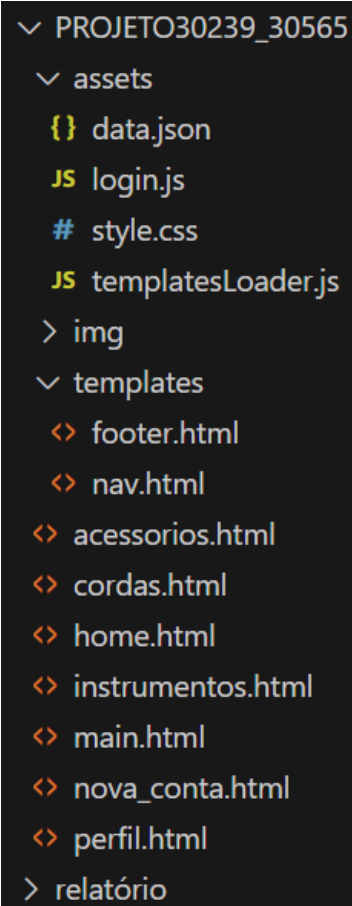


Figura 1 - Diagrama de Uso

## Estrutura do Site

A estrutura do nosso website é simples e concisa, com cada página HTML tendo nomes intuitivos. Além das páginas HTML, temos 2 scripts em JavaScript e um ficheiro JSON utilizado para autenticação no website. Dentro da pasta "img", encontram-se as imagens utilizadas.



```

  ▾ PROJETO30239_30565
    ▾ assets
      {} data.json
      JS login.js
      # style.css
      JS templatesLoader.js
    > img
    ▾ templates
      <> footer.html
      <> nav.html
      <> acessorios.html
      <> cordas.html
      <> home.html
      <> instrumentos.html
      <> main.html
      <> nova_conta.html
      <> perfil.html
    > relatório

```

Figura 2 - Estrutura do website

# Esboço das páginas

Neste esboço, delineamos a estrutura fundamental do nosso website, fornecendo um guia para a implementação subsequente do código, visando uma aparência semelhante.

O nosso objetivo é proporcionar uma experiência de utilizador fluída e eficiente, com destaque para a autenticação, gestão de perfil e operações CRUD de inventário. Este esboço serviu como um guia fundamental para o desenvolvimento futuro do nosso website.

email

password

login

new acc

Figura 3 - Esboço login

home

profile

instrumentos

acessórios

cordas

Figura 4 - Esboço "main page"

new entry

delete

modify

instrumento	nome	alcunha

Figura 5 - Esboço tabela CRUD

## Base de Dados

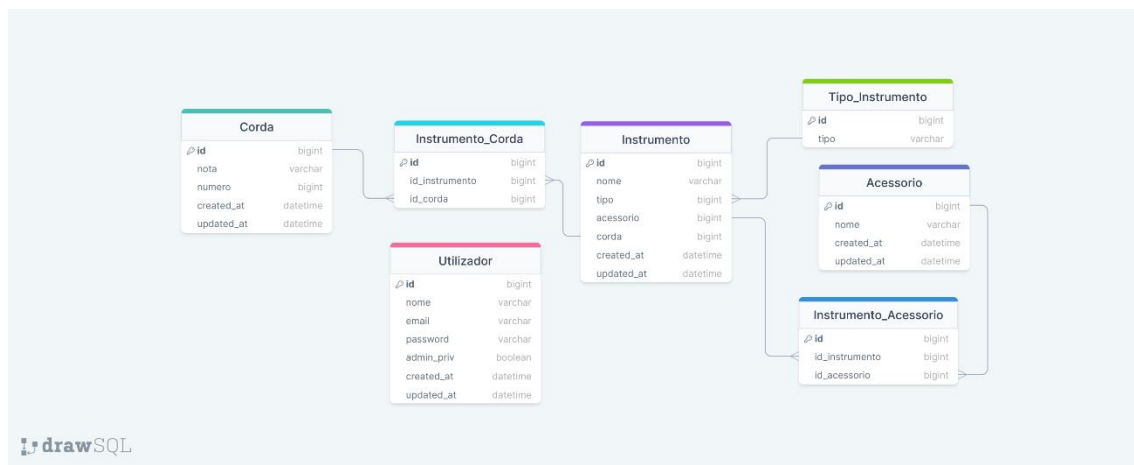


Figura 6 - Modelo Relacional

```

CREATE TABLE `Instrumento` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `nome` VARCHAR(255) NOT NULL,
  `created_at` DATETIME NOT NULL,
  `updated_at` DATETIME NOT NULL
);

CREATE TABLE `Instrumento_Corda` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `id_instrumento` BIGINT NOT NULL,
  `id_corda` BIGINT NOT NULL
);

CREATE TABLE `Tipo_Instrumento` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `tipo` VARCHAR(255) NOT NULL
);

CREATE TABLE `Instrumento_Acessorio` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `id_instrumento` BIGINT NOT NULL,
  `id_acessorio` BIGINT NOT NULL
);

CREATE TABLE `Utilizador` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `nome` VARCHAR(255) NOT NULL,
  `email` VARCHAR(255) NOT NULL,
  `password` VARCHAR(255) NOT NULL,
  `admin_priv` TINYINT(1) NOT NULL,
  `created_at` DATETIME NOT NULL,
  `updated_at` DATETIME NOT NULL
);

CREATE TABLE `Instrumento` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `nome` VARCHAR(255) NOT NULL,
  `tipo` BIGINT NOT NULL,
  `acessorio` BIGINT NOT NULL,
  `corda` BIGINT NOT NULL,
  `created_at` DATETIME NOT NULL,
  `updated_at` DATETIME NOT NULL
);

CREATE TABLE `Corda` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `nota` VARCHAR(255) NOT NULL,
  `numero` BIGINT NOT NULL,
  `created_at` DATETIME NOT NULL,
  `updated_at` DATETIME NOT NULL
);

ALTER TABLE
  `Instrumento` ADD CONSTRAINT `instrumento_corda_foreign`
FOREIGN KEY(`corda`) REFERENCES
  `Instrumento_Corda`(`id_instrumento`);
ALTER TABLE
  `Instrumento_Corda` ADD CONSTRAINT
  `instrumento_corda_id_corda_foreign` FOREIGN KEY(`id_corda`)
REFERENCES `Corda`(`id`);
ALTER TABLE
  `Instrumento` ADD CONSTRAINT `instrumento_tipo_foreign`
FOREIGN KEY(`tipo`) REFERENCES `Tipo_Instrumento`(`tipo`);
ALTER TABLE
  `Instrumento_Acessorio` ADD CONSTRAINT
  `instrumento_acessorio_id_acessorio_foreign` FOREIGN
  KEY(`id_acessorio`) REFERENCES `Acessorio`(`id`);
ALTER TABLE
  `Instrumento` ADD CONSTRAINT `instrumento_acessorio_foreign`

```

Figura 7 - Código SQL da base de dados

Páginas web

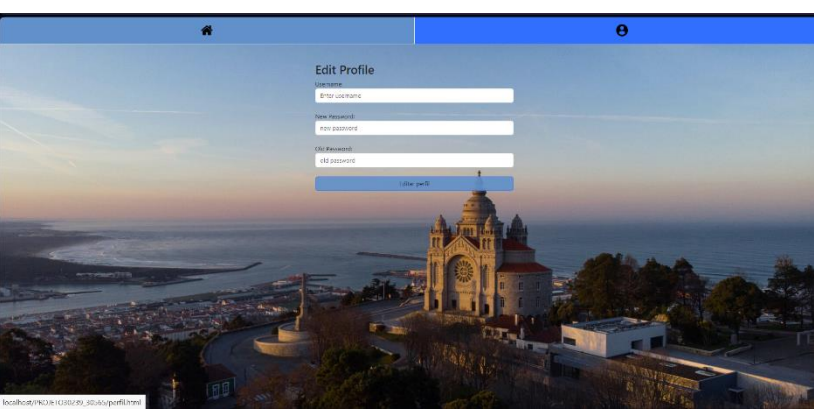
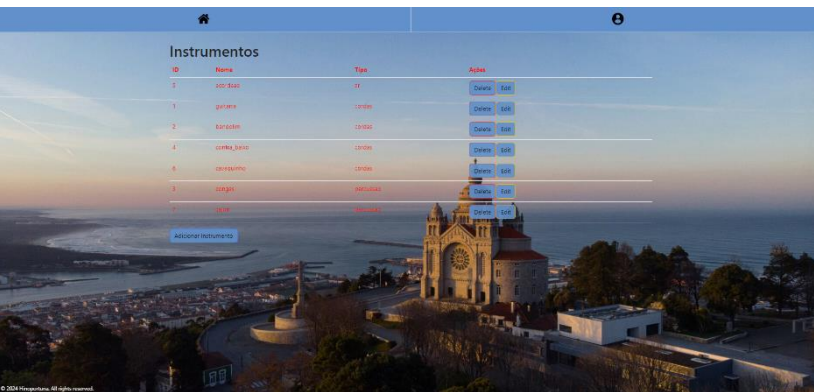
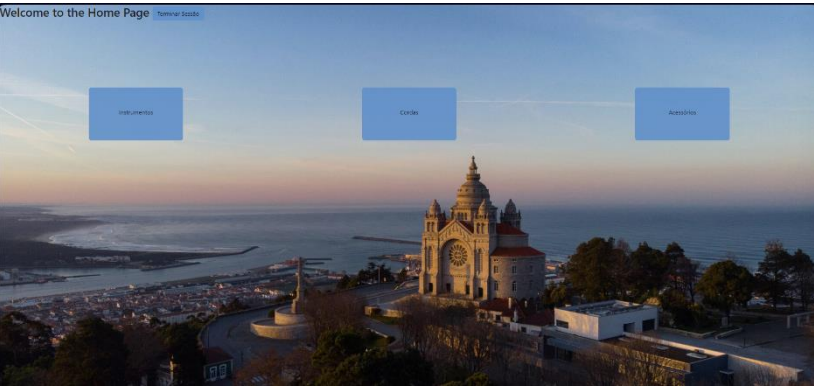
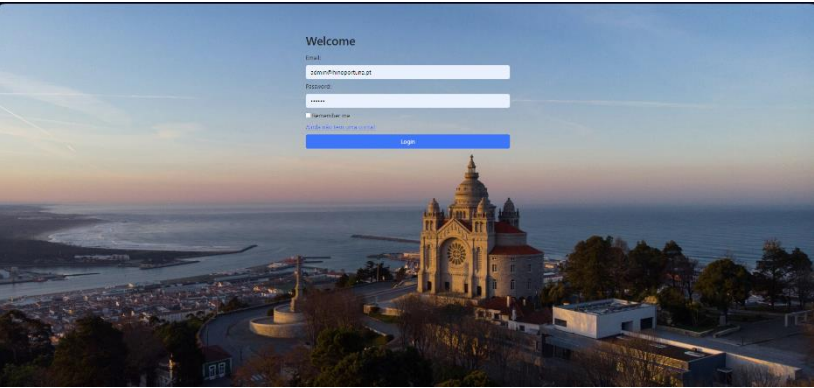


Figura 8 - principais páginas HTML



## Código JavaScript

```
PROJETO30239_30565 > assets > JS templatesLoader.js > ...
1   const loadNavBar = () => {
2     fetch("./templates/nav.html")
3       .then((response) => response.text())
4       .then((data) => {
5         document.getElementsByTagName("body")[0].insertAdjacentHTML("afterbegin", data);
6       });
7   };
8
9   const loadFooter = () => {
10    fetch("./templates/footer.html")
11      .then((response) => response.text())
12      .then((data) => {
13        document.getElementsByTagName("body")[0].insertAdjacentHTML("beforeend", data);
14      });
15  };
```

Figura 9- java script nº1

As funções “loadNavBar” e “loadFooter” têm como propósito carregar dinamicamente uma barra de navegação e um rodapé numa página HTML a partir de ficheiros HTML externos. Isto garante que, após a autenticação, a barra de navegação e o rodapé permaneçam sempre visíveis nas páginas.

```
PROJETO30239_30565 > assets > JS login.js > addEventListener("submit") callback
1   document.getElementById("loginForm").addEventListener("submit", function(event) {
2     event.preventDefault(); // prevent default form submission
3
4     // Get form data
5     const formData = new FormData(event.target);
6
7     // Convert form data to JSON
8     const jsonData = {};
9     formData.forEach((value, key) => {
10      jsonData[key] = value;
11    });
12
13    // Load users from JSON file (Replace with actual backend code in real application)
14    fetch("assets/data.json")
15      .then(response => response.json())
16      .then(data => {
17        const users = data.users;
18        // Check if user credentials match any user in the JSON file
19        const foundUser = users.find(user => user.email === jsonData.email && user.password === jsonData.password);
20        if (foundUser) {
21          localStorage.setItem('loggedInUser', JSON.stringify(foundUser));
22          window.location.href = "home.html";
23        } else {
24          displayMessage("Invalid email or password");
25        }
26      })
27      .catch(error => {
28        console.error('Error fetching users:', error);
29        displayMessage("Error fetching users. Please try again later.");
30      });
31
32    // Clear form fields
33    event.target.reset();
34  });
35
36  function displayMessage(message) {
37    const messageDiv = document.getElementById("message");
38    messageDiv.innerHTML = message;
39  }
```

Figura 10 - java script nº2

Este script cria um formulário de início de sessão numa página HTML. Quando o utilizador preenche e envia o formulário, os dados são capturados e convertidos em formato JSON. Em seguida, o script faz um pedido para carregar um ficheiro JSON que contém informações dos utilizadores (simulando uma autenticação de utilizador). Se as credenciais fornecidas pelo utilizador coincidirem com as de algum utilizador no ficheiro JSON, o utilizador é autenticado e

redirecionado para a página inicial. Caso contrário, é exibida uma mensagem de erro. Após o envio do formulário, os campos são limpos para uma nova entrada.

## Evolução

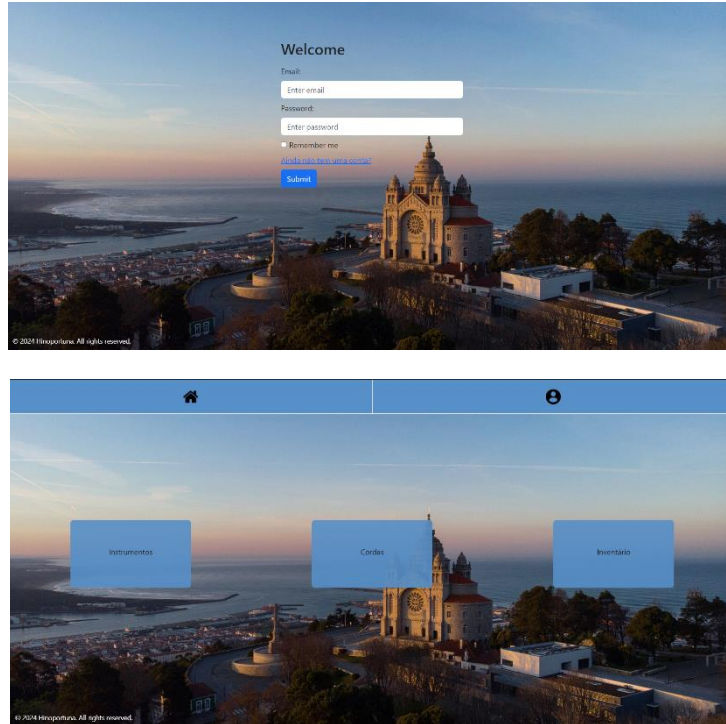


Figura 5 - primeira iteração do website

## Conclusão

Em resumo, este projeto da disciplina de Programação Web teve como objetivo criar uma aplicação robusta que proporcionasse uma experiência completa aos utilizadores, com zonas pública e privada distintas, separadas por autenticação. Utilizando tecnologias como HTML, bootstrap, CSS, JavaScript e um ficheiro JSON local, estabelecemos uma base sólida para o desenvolvimento da plataforma, permitindo futuras iterações e melhorias.

Este projeto não apenas demonstrou as nossas habilidades técnicas na implementação de um website, mas também evidenciou a nossa capacidade de abordar e resolver desafios reais encontrados no desenvolvimento web. A etapa que mais nos desafiou foi, sem dúvida, a de carregar dinamicamente o conteúdo do website por meio de um arquivo JSON local, assim como aprimorar o design e a estética do website.

Estamos satisfeitos com o resultado do nosso trabalho, pois conseguimos alcançar os objetivos propostos e criar uma plataforma funcional e visualmente atraente. Este projeto serviu como uma valiosa oportunidade de aplicar os conhecimentos adquiridos em aula, na prática, preparando-nos para desafios futuros no mundo do desenvolvimento web.

## Referências

W3Schools. (n.d.). Bootstrap 5 Tutorial. Retirado 12 Abril, 2024, de <https://www.w3schools.com/bootstrap5/index.php>

Bootstrap. (n.d.). Introduction. In Bootstrap. Retirado 12 Abril, 2024, de <https://getbootstrap.com/docs/5.0/getting-started/introduction/>