

Programação Web Responsiva

Introdução

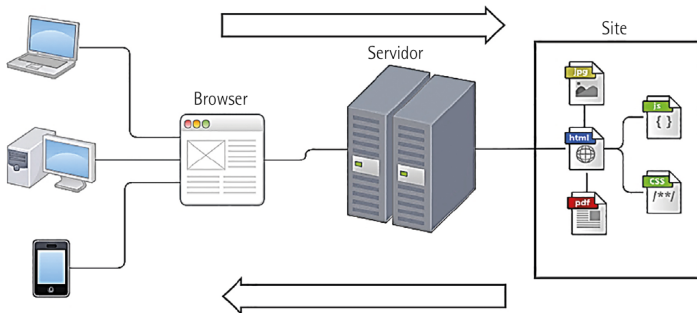
Prof. Bruno Azevedo

UNIP – Universidade Paulista



Comunicação Cliente-Servidor

- Cliente × Servidor.
- O servidor fornece os documentos solicitados pelo cliente.
- As solicitações e respostas são tratadas através do protocolo HTTP (Protocolo de Transferência de Hipertexto).
- Esse protocolo permite a transferência de texto, imagens, vídeos, páginas, scripts, entre outros recursos.



URL

- Cada página e recurso na Web possui seu próprio endereço chamado de URL (Localizador Uniforme de Recursos).
- Uma URL geralmente é composta por três componentes: o protocolo, o domínio e o caminho absoluto para o documento ou recurso.
- protocolo://domínio/caminho/recurso
- Por exemplo: `http://www.exemplo.com/exemplo/teste.html`
- Podem existir outros componentes (ex: porta, credenciais de login, etc).

URL

```
http://www.exemplo.com/exemplo/teste.html
```

- A primeira parte da URL especifica o protocolo utilizado na comunicação.
- O termo **HTTP** indica o uso do Protocolo de Transferência de Hipertexto.
- Também existem URLs iniciadas com **HTTPS**, que representam a versão segura do HTTP.
- Nesse caso, a comunicação entre o navegador e o servidor é criptografada, garantindo maior segurança na transferência de dados.

URL

```
http://www.exemplo.com/exemplo/teste.html
```

- **www.exemplo.com** é o nome do site ou o nome de domínio.
- Neste caso, **exemplo.com** é o nome de domínio e o prefixo **www** é um subdomínio comum, embora não seja necessário.
- É uma convenção que indica que o recurso está hospedado em um servidor da Web.

URL

```
http://www.exemplo.com/exemplo/teste.html
```

- `exemplo/teste.html` define o caminho e o recurso no servidor.
- Ou seja, o recurso está localizado dentro do diretório **exemplo** e o nome do arquivo é **teste.html**.
- Isso indica onde exatamente o servidor deve procurar o recurso solicitado.
- Sucintamente, a URL `http://www.exemplo.com/exemplo/teste.html` instrui o navegador a usar o protocolo HTTP para se conectar ao domínio **www.exemplo.com** e solicitar o documento **teste.html** que está localizado no diretório **exemplo**.

URL

- Entretanto, nem sempre as URLs contém esses componentes. Como mencionado, **www** é uma convenção e não é obrigatório.
- **HTTP** é o protocolo mais comum, mas existem outros como FTP, SSH e SMTP.
- Atualmente, os navegadores geralmente adicionam automaticamente HTTPS em vez de HTTP, por ser mais seguro, mas o servidor precisa suportar esse protocolo.

URL

- Em outros casos, o próprio recurso pode ser omitido.
- O diretório já pode possuir uma página padrão, que será carregada ao usar apenas o caminho.
- Por exemplo, `http://www.exemplo.com/exemplo/` poderia retornar o arquivo **teste.html**.
- E vocês já viram isso, afinal, quando visitando um site pela primeira vez vocês geralmente digitam `www.nomedosite.com.br` em vez de `www.nomedosite.com.br/recurso`.
- Existe uma página padrão sendo carregada quando fazem isso.

Adicionando Links a Suas Páginas Web

- Links são compostos de três componentes: a Tag de âncora, o atributo href e o conteúdo que será exibido.
- Tag de âncora: `<a>`
- Atributo href para definir a URL.
- Texto apresentado no lugar do link.

```
<a href="http://www.unip.br"> Visite o site da Universidade Paulista </a>
```

Sobre o Atributo *href* em Links

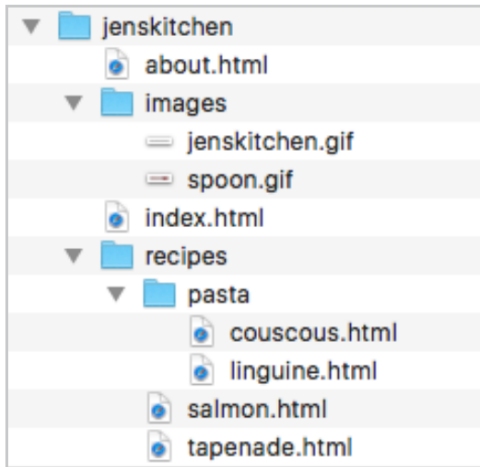
- Podem usar URLs absolutas: `http://www.unip.br/`.
- Podem usar URLs relativas: `caminho_em_meu_site/index.html`.
- URLs absolutas são simples de usar, basta utilizarmos a âncora com o atributo `href`.
- **URLs relativas exigem maior atenção.** Elas são usadas principalmente na criação de sites e aplicações web, pois dependem da estrutura de diretórios do projeto.

Arquivos e Pastas

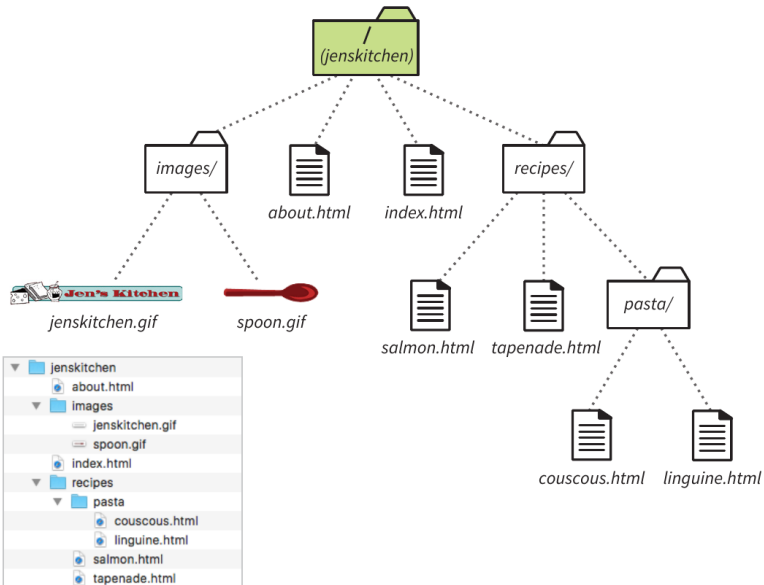
- Um arquivo é um conjunto de dados armazenados em um dispositivo de armazenamento de dados, como um SSD, um pendrive ou na nuvem.
- Arquivos podem conter diferentes tipos de dados, como texto, imagens, vídeos ou áudios.
- Por exemplo, você pode ter uma foto favorita em seu computador ou celular.
- Pastas são usadas para organizar os arquivos.
- Uma pasta pode conter arquivos ou outras pastas.

Arquivos e Pastas

- Isso resulta em uma **hierarquia de pastas**.

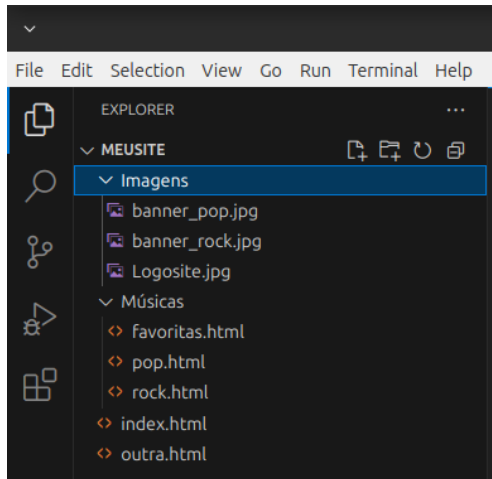


Arquivos e Pastas



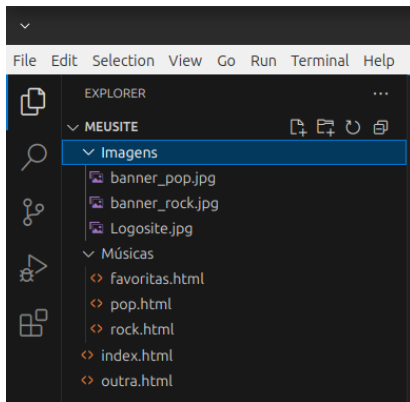
Usando Links Relativos

- Para usarmos links relativos, precisamos considerar a **estrutura de pastas** do nosso site.
- Abaixo temos uma estrutura de pastas e arquivos de um site visualizada no VSCode.



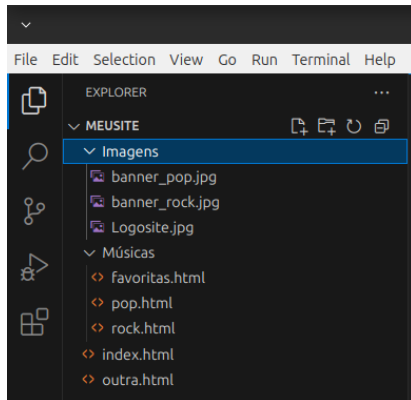
Usando Links Relativos

- A pasta MEUSITE contém todo o conteúdo do meu site.
- Outras pastas e vários arquivos.



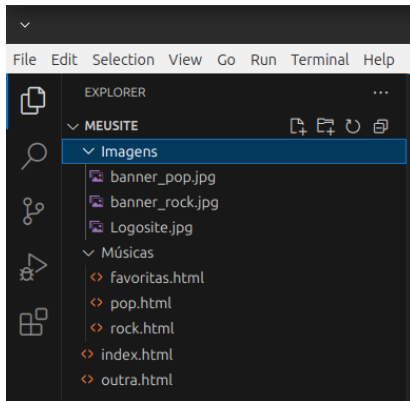
Usando Links Relativos

- A pasta MEUSITE contém dois arquivos HTML: `index.html` e `outra.html`.



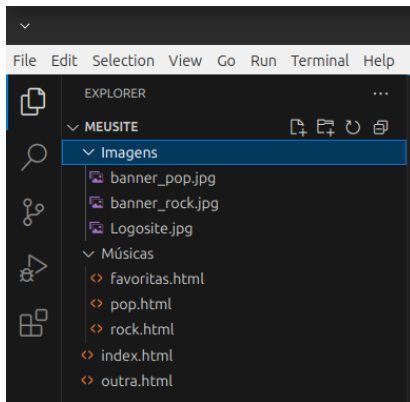
Usando Links Relativos

- A pasta Imagens contém as imagens que uso em meu site: banner_pop.jpg, banner_rock.jpg e Logosite.jpg



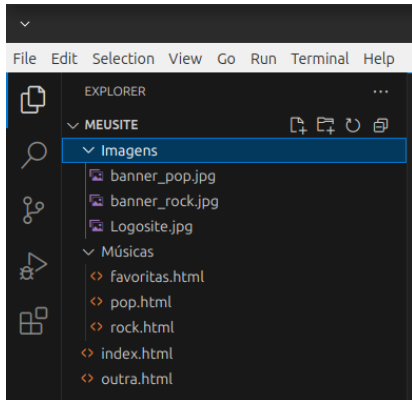
Usando Links Relativos

- A pasta Músicas contém mais arquivos HTML que uso em meu site: `favoritas.html`, `pop.html` e `rock.html`



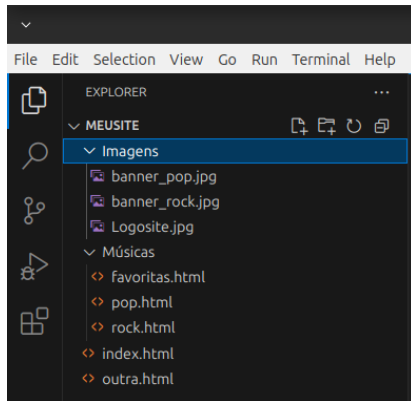
Usando Links Relativos

- Precisamos considerar a **estrutura do site** para criarmos URL relativas corretamente.
- Por exemplo, queremos criar um link para o arquivo `rock.html` em nossa página inicial `index.html`.
- Esse link precisa considerar a localização de `rock.html` **relativo** à `index.html`.



Usando Links Relativos

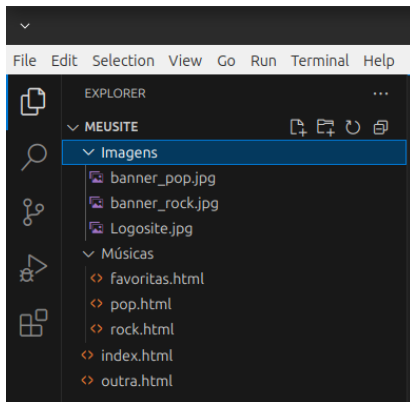
- No caso, o arquivo `rock.html` está em uma pasta chamada `Músicas`, que está no mesmo nível da estrutura.
- Portanto, **precisamos entrar** na pasta `Músicas` para acessar o arquivo `rock.html`.



Usando Links Relativos

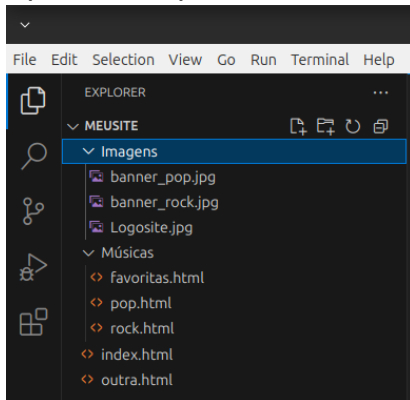
- Para este fim, basta utilizarmos o nome da pasta.
- Portanto, o código que adicionaremos em `index.html` para criar esse link deve ser:

```
<a href="Músicas/rock.html">Link para a página rock.html</a>
```



Usando Links Relativos

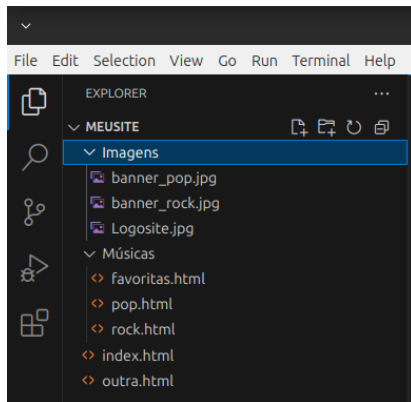
- Se colocássemos apenas:
`Link para a página rock.html`
- O link não funcionaria, já que o arquivo `rock.html` está dentro da pasta Músicas, **pela perspectiva do arquivo `index.html`**.



- Sempre considerem a perspectiva do arquivo em que queremos adicionar o link.

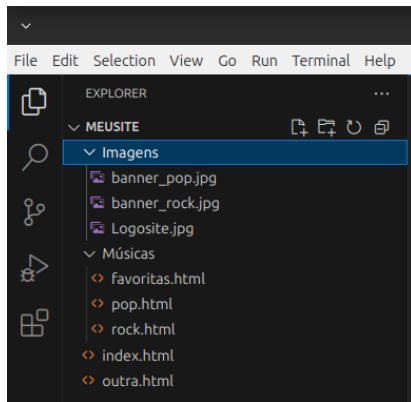
Usando Links Relativos

- E se fosse o contrário? Se queremos adicionar um link para `index.html` no arquivo `rock.html`?
- Da perspectiva do arquivo `rock.html` precisamos **sair da pasta Músicas** para então encontrar o arquivo `index.html`.



Usando Links Relativos

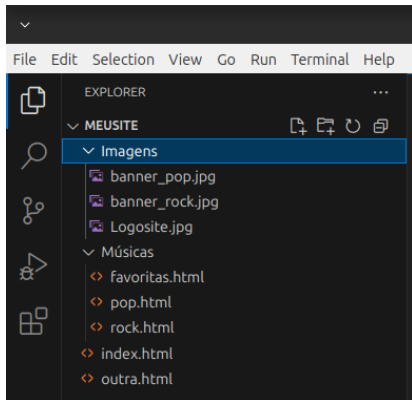
- Para sairmos da pasta Músicas usaremos o indentificador “..”.
- Este identificador permite “subirmos um nível” na hierarquia de pastas.



Usando Links Relativos

- Portanto, o código que adicionaremos em `rock.html` para criar um link para `index.html` será:

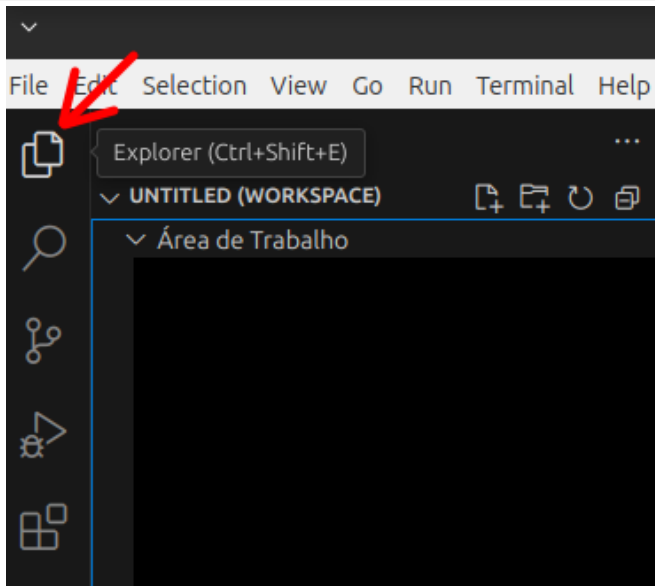
```
<a href="../index.html">Link para a página index.html</a>
```



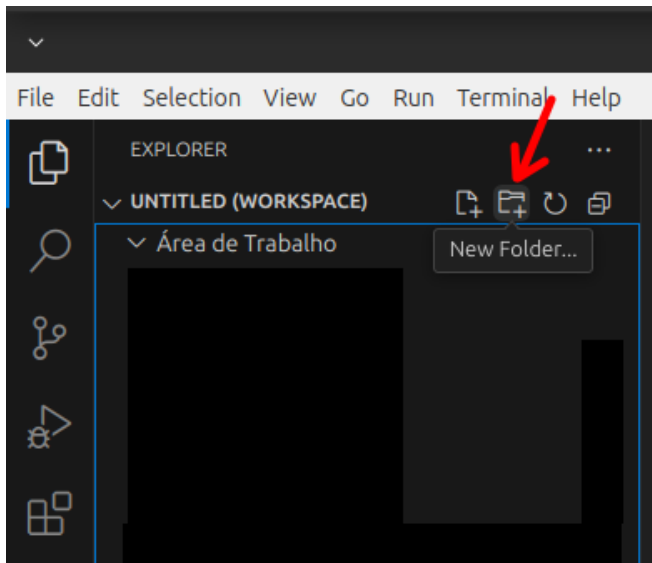
Usando o VSCode para Criar Arquivos e Pastas

- Vamos criar uma estrutura de pastas e arquivos para o nosso site.
- Usaremos o VSCode para criar tanto as pastas como os arquivos.

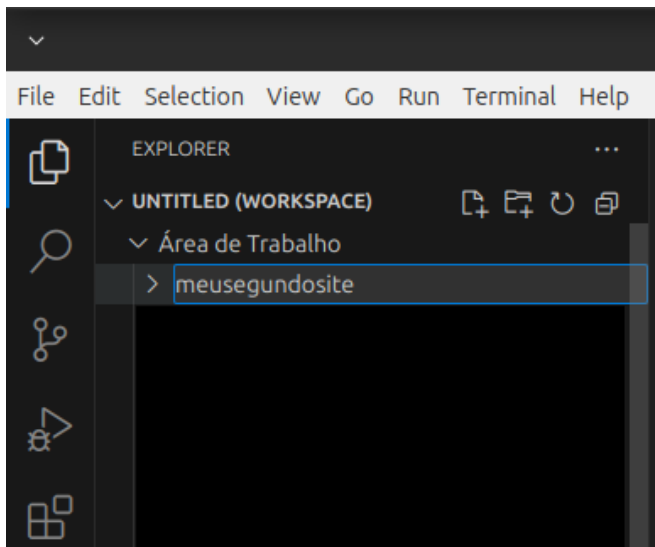
Usando o VSCode para Criar Arquivos e Pastas



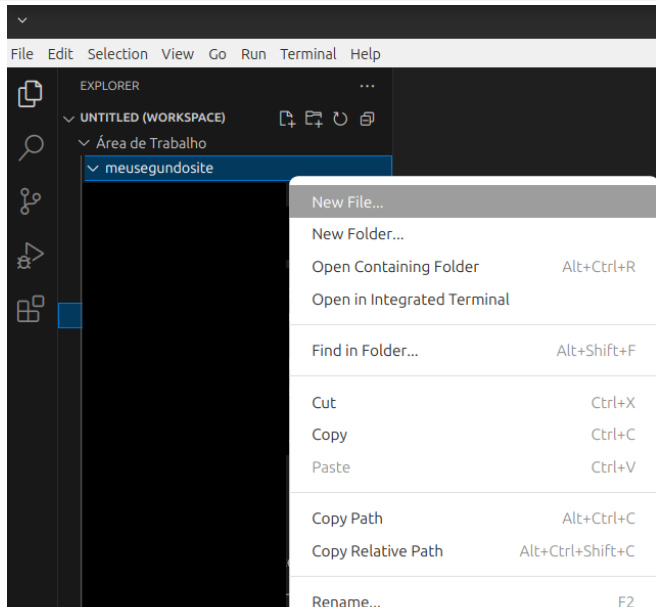
Usando o VSCode para Criar Arquivos e Pastas



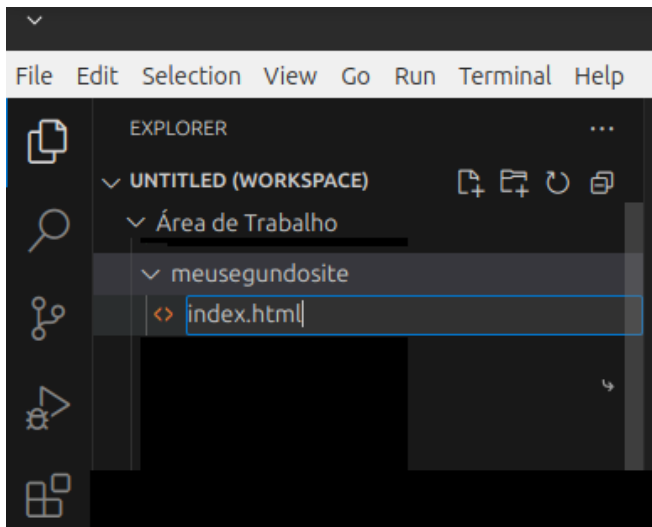
Usando o VSCode para Criar Arquivos e Pastas



Usando o VSCode para Criar Arquivos e Pastas

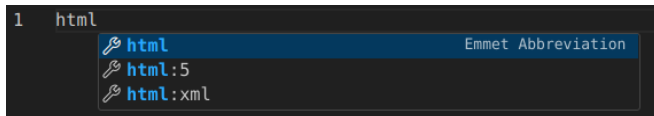


Usando o VSCode para Criar Arquivos e Pastas



Dica

- No VSCode, para gerar rapidamente o esqueleto básico de uma página HTML, digitem `html` e o editor mostrará várias opções.



- Seleccionem a segunda opção: `html:5`. Isso irá gerar a estrutura padrão de um documento HTML5.
- Com isso, seu arquivo HTML será preenchido automaticamente com o esqueleto básico, incluindo tags `html`, `head`, `body`, entre outras, facilitando o início do desenvolvimento.

Atividade Prática 2

- ❶ Desenvolver um site simples para um Pequeno Negócio Local, como um café, restaurante ou mercado, com três páginas:
 - index.html (página inicial com título e breve introdução).
 - sobre.html (página com uma explicação sobre a Web).
 - contato.html (página com um link de e-mail de contato).
 - Ampliação: incluir mais páginas com textos e imagens sobre a empresa escolhida.

Vocês devem estruturar a navegação entre as páginas utilizando a Tag `<a>`.

A página `index.html` deve possuir links para as outras duas páginas.

As páginas `sobre.html` e `contato.html` devem possuir links para a página `index.html`.

- ❷ Cada aluno deve produzir um relatório de 1 a 3 páginas contendo:
 - Resumo teórico: explicação sobre a estrutura de um site e a função de cada página.
 - Código-fonte comentado: explicação dos links e estrutura do projeto.

Critérios de Avaliação

Critério	Peso	Descrição
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula.
Estrutura e Organização do Código	3,0	Organização e clareza do código HTML.
Funcionamento da Solução	3,0	Uso correto das tags básicas e comentários explicativos dentro do código.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo.

- A nota final será a soma dos valores obtidos em cada critério de avaliação.