



MATH$\langle \langle FUSION \rangle \rangle TECH$

Universidade do Estado do Rio de Janeiro - UERJ

Projetos de Extensão

MATERIAL COMPLEMENTAR GERADO PARA OS ENCONTROS DO PROJETO

Autora: Prof^a. Dr^a. Greice Keli Silva Lacerda

Email: greice.lacerda.uerj@gmail.com

Itaboraí

2025

MATERIAL COMPLEMENTAR II

1. Revisão Prática

Olá pessoal, para quem ainda não instalou as ferramentas essenciais (VS Code e GitHub Desktop), esta seção revisa o processo.

1.1. Instalação de Ferramentas

O primeiro passo é garantir que temos o nosso “laboratório” (o editor de código) e nossa “ferramenta de colaboração” (o Git) instalados.

- **Visual Studio Code (Editor):** code.visualstudio.com
- **GitHub Desktop (Cliente Git):** desktop.github.com

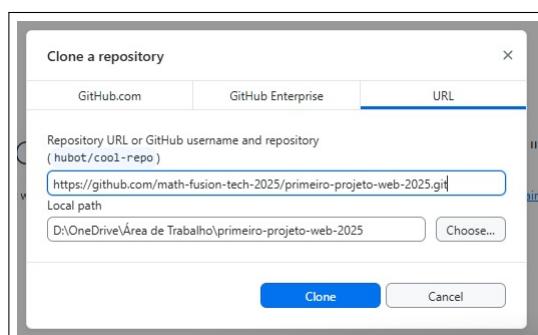
1.2. Clonando o Repositório do Projeto (GitHub Desktop)

Após instalar o GitHub Desktop, o primeiro passo é “clonar” (fazer uma cópia local) do nosso repositório da turma.

1. Abra o **GitHub Desktop**.
2. Vá em *File > Clone Repository...* (ou clique no botão “Clonar Repositório”).
3. Selecione a aba **URL**.
4. No campo URL, insira o caminho do nosso repositório:

`https://github.com/math-fusion-tech-2025/primeiro-projeto-web-2025.git`

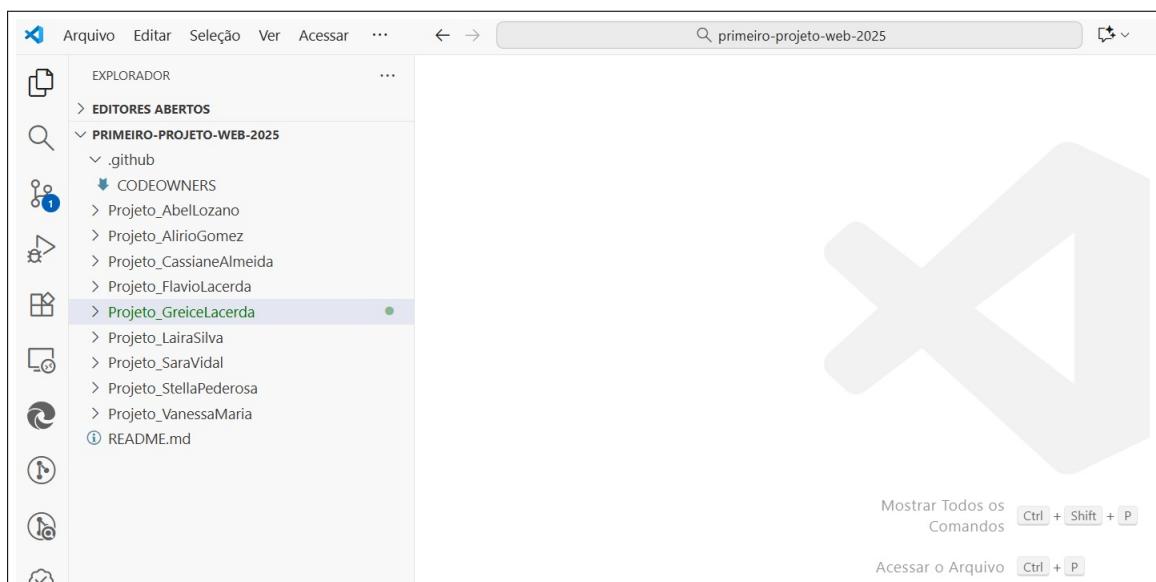
5. Em *Local path* (Caminho Local), clique em “Choose...” e escolha uma pasta fácil de achar no seu computador (ex: Documentos/Projetos).
6. Clique em **Clonar (Clone)**.



1.3. Abrindo o Projeto no VS Code

Agora que os arquivos estão no seu computador, vamos abri-los no editor.

1. Após clonar, você pode fechar o GitHub Desktop.
2. Abra o **Visual Studio Code**.
3. No menu superior, vá em *Arquivo (File) > Abrir Pasta... (Open Folder...)*.
4. Localize e selecione a pasta que você acabou de clonar (ex: `primeiro-projeto-web-2025`).
5. O VS Code abrirá o projeto.



Ao abrir, você verá no explorador lateral (à esquerda) todos os arquivos do projeto. Observe que, que cada membro da equipe tem a sua própria pasta de trabalho (ex: `/Projeto_GreiceLacerda/`, ..., `/Projeto_VanessaMaria/`).

2. Aperfeiçoando o Ambiente: Ferramentas Essenciais do VS Code

Agora que o projeto está aberto, vamos instalar duas extensões no VS Code que irão melhorar drasticamente nosso fluxo de trabalho. Extensões são “plugins” que adicionam superpoderes ao nosso editor.

2.1. A Importância da Indentação e o Prettier

O que é Indentação? Indentação é o recuo (usando “Tab” ou “espaços”) que aplicamos ao nosso código para mostrar a hierarquia. Uma tag “filha” (como `<h1>`) deve estar um nível de recuo “adentro” da

tag “pai” (como <header>).

Fazendo uma **analogia matemática**, um código sem indentação é como uma longa equação escrita em uma única linha, sem parênteses ou ordem de operações clara. É funcional, mas impossível de ler, depurar ou dar manutenção.

A Ferramenta: Prettier - Code formatter

Em vez de indentarmos manualmente (e errarmos), usamos uma extensão chamada **Prettier**.

- O Prettier é um formatador de código “opinativo”.
- Ele **formata automaticamente** o seu código, aplicando a indentação correta, quebrando linhas longas e padronizando aspas.
- Ele elimina debates de estilo (ex: “Tab vs. 2 espaços”) e garante que todo o código do nosso projeto colaborativo seja limpo e consistente.

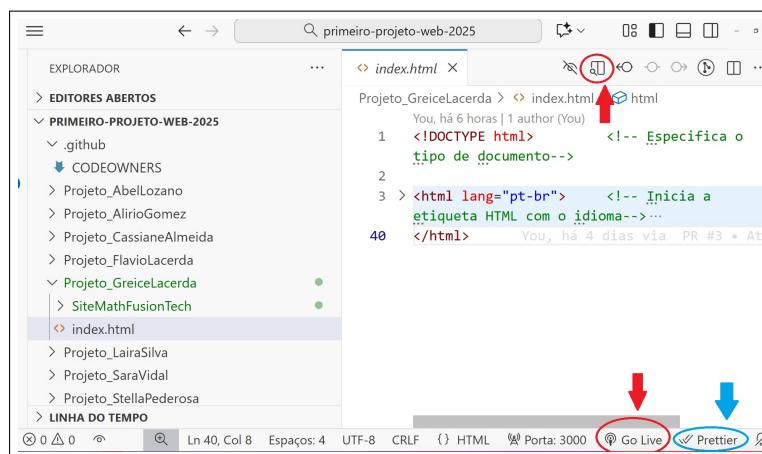
2.2. A Visualização Prévia e o Live Server

A Ferramenta **Live Server** cria um pequeno servidor de desenvolvimento no seu computador.

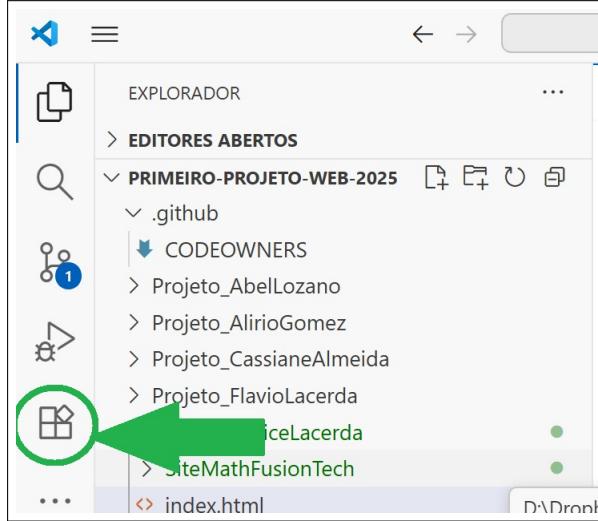
- Após instalado, um botão **”Go Live”** aparecerá no canto inferior direito do seu VS Code.
- Ao clicar nele, ele abre seu index.html no navegador.
- A partir daí, toda vez que você salvar (Ctrl+S) seu arquivo HTML ou CSS, o **navegador se atualiza automaticamente**.
- Isso é o que chamamos de **visualização prévia** (ou *live preview*), e acelera nosso ciclo de desenvolvimento de forma incrível.

2.3. Guia de Instalação das Extensões

Vamos instalar o Prettier (seta azul) e o Live Server (setas vermelhas) ilustrados na imagem a seguir.

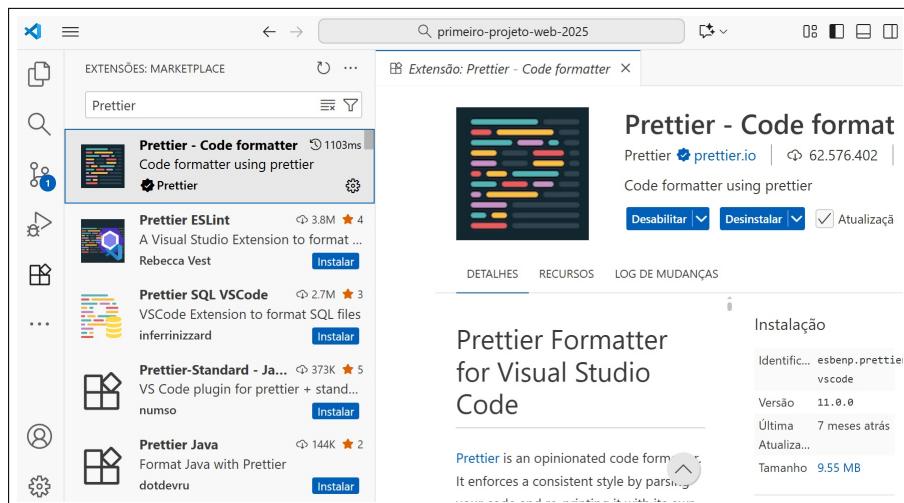


- 1. Abra a Aba de Extensões:** Na barra lateral esquerda do VS Code, clique no ícone que parece quatro blocos (o “Marketplace de Extensões”).



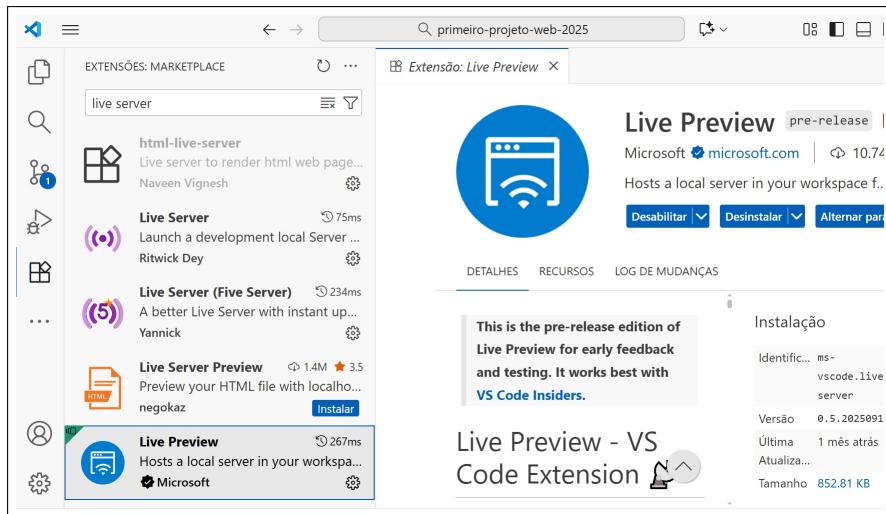
- 2. Instale o Prettier:**

- Na barra de busca, digite **Prettier - Code formatter**.
- Clique no primeiro resultado (de *Prettier.io*) e clique no botão azul **Instalar**.



- 3. Instale o Live Server:**

- Na mesma barra de busca, digite **Live Server**.
- Clique no resultado (*Microsoft*) e clique em **Instalar**.

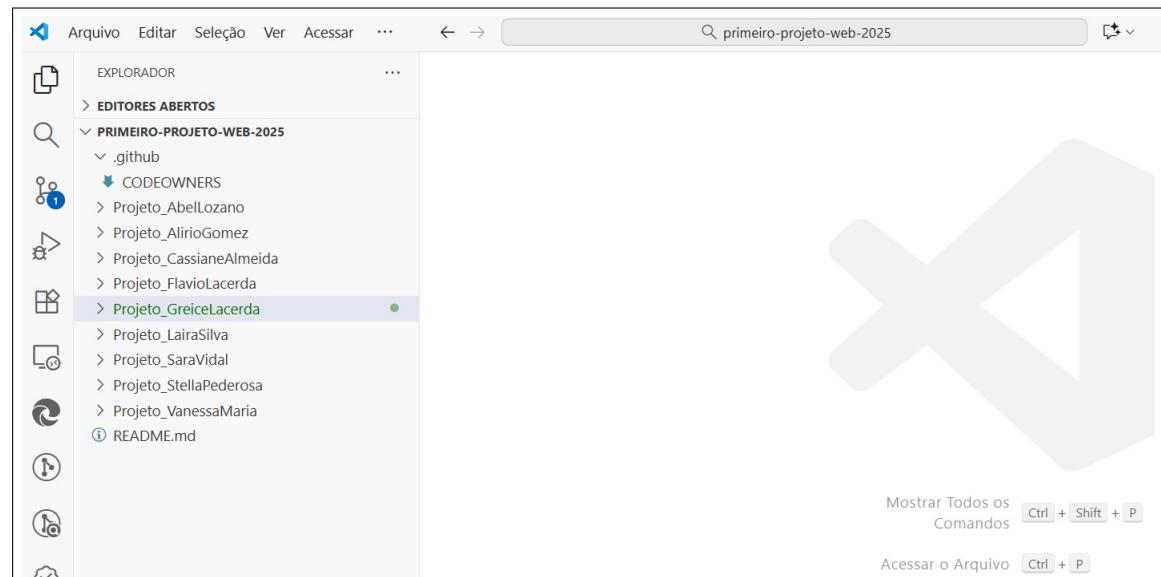


Com essas duas extensões, nosso ambiente está pronto para um desenvolvimento rápido, limpo e profissional.

3. Análise de Arquivos do Projeto

3.1. Regras de Colaboração: O Arquivo CODEOWNERS

No explorador do VS Code, você verá um arquivo chamado CODEOWNERS, dentro da pasta `.github/`). **O que é este arquivo?** Ele é um “mapa de responsabilidade” para o GitHub.



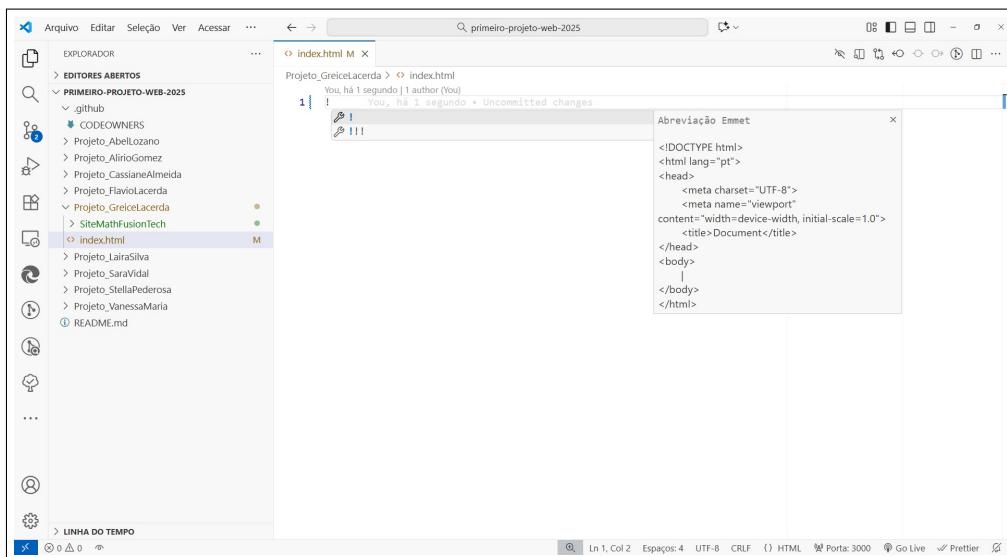
- **Função Principal:** Ele define *automaticamente* quem deve ser chamado para revisar alterações (Pull Requests) em pastas específicas.

- **Nossa Regra:** No nosso projeto, ele está configurado para que **cada aluno seja “proprietário” da sua própria pasta**.
- **Na Prática:** Se você fizer uma alteração na pasta por exemplo:/Projeto_GreiceLacerda/, o GitHub automaticamente solicitará a revisão da @GreiceLacerda para autorizar a atualização. Se você mexer na sua própria pasta, ele chamará você (ou a professora, dependendo das regras).

Ele garante que ninguém mexa na pasta do outro sem a devida supervisão, mantendo a organização do nosso projeto colaborativo.

3.2. Acesse a sua pasta: O Arquivo index.html

No explorador do VS Code, acesse a pasta com seu nome e clique no arquivo index.html. Observe que no campo central (Codespace) aparecerá um arquivo em branco com o cursor intermitente. Esse é o espaço para digitação do código. Digite “!” e clique em “Enter”.



O VS Code usará a ferramenta autocompletar para inserir uma estrutura básica de HTML5 no seu espaço de código. Vamos analisar cada linha de código nessa estrutura.

```

<!DOCTYPE html>           <!-- DOCUMENTAÇÃO DA PÁGINA ESPECIFICAÇÃO DE HTML5-->
<html lang="pt">          <!-- INÍCIO DA TAG HTML COM O LÍNGUA PORTUGUÊS BRASIL-->
<head>                   <!-- CABEÇA DA PÁGINA --> ...
</head>                  <!-- FECHAMENTO DA TAG HEAD-->
<body>                   <!-- CORPO DA PÁGINA --> ...
</body>                  <!-- FECHAMENTO DA TAG BODY -->
</html>                  <!-- FECHAMENTO DA TAG HTML-->

```

O Código `<!DOCTYPE html>` não é uma tag HTML ele é uma “declaração” (uma chamada de doctype). Ela é a primeira coisa que deve aparecer no arquivo e serve para informar ao navegador qual versão do HTML está sendo usada. Isso garante que o navegador renderize a página no “modo de padrões”, em vez de tentar adivinhar.

O Código: `<html lang="pt">` é a tag que inicia o documento HTML. Ela é a “raiz” de toda a página. Tudo o que vier depois deve estar dentro dela e da sua tag de fechamento (`</html>`). Observe a linha vertical que liga as duas tags, essa linha indica todos os elementos que estão dentro delas, ou seja, suas tags “filhas”. Essa verificação é possível, sempre que respeitamos a indentação do código. O atributo `lang="pt"` especifica o idioma principal do conteúdo da página (neste caso, “pt”para Português). Isso é muito importante para:

1. **SEO (Otimização de Busca):** Ajuda motores de busca (como o Google) a entender o idioma da sua página e mostrá-la para usuários que buscam naquele idioma.
2. **Navegadores:** Pode ser usado para sugerir traduções ou verificar a ortografia.

Na linha 3 do nosso código temos a tag `<head>`. Esta tag abre a seção “cabeça” do documento. O conteúdo dentro do “`<head>`” não é visível diretamente na área de conteúdo da página. Ele serve para carregar “metadados” sobre a página, ou seja, informações sobre o documento. Isso inclui o título da aba, a codificação de caracteres, links para arquivos CSS (estilo) e JavaScript (interatividade). O Código `</head>` é a tag de fechamento da seção “`<head>`”. Ela sinaliza ao navegador que a seção de metadados terminou.

Já na linha 11 do nosso código temos a tag `<body>`, esta tag abre a seção “corpo” do documento. É aqui que vai todo o conteúdo visível da sua página: textos, títulos (“`<h1>`”), parágrafos (“`<p>`”), imagens (“``”), links (“`<a>`”), tabelas, etc. Tudo o que o usuário vê na janela do navegador está dentro do “`<body>`” e “`</body>`” (Fechamento da tag `<body>`).

Resumo da Estrutura

- `<!DOCTYPE html>`: A declaração que define a versão do HTML.
- `<html>`: O elemento-raiz que contém tudo.
- `<head>`: A seção para metadados e informações sobre o documento.
- `<body>`: A seção para todo o conteúdo visível da página.

3.2.1. A tag `<head>` (Cabeça/Metadados)

A seção é o `<head>` “cérebro” invisível da página. Ela não contém conteúdo visual, apenas metadados e links para outros arquivos. Nela encontramos as tags vazias `<meta>` e a tag `<title>`

- `<meta charset="UTF-8">`: Esta é uma tag de “metadados”, cujo atributo `charset` (conjunto de caracteres) define a codificação de texto que o navegador deve usar para ler seu arquivo. Essa

tag é vazia (oficialmente chamada de *void element* em inglês) porque não possui uma tag de fechamento (`</tag>`). **UTF-8** é o padrão universal moderno. Ele é crucial porque permite que a página exiba corretamente todos os tipos de caracteres de quase todos os idiomas do mundo, incluindo acentos (como ç, ã, é) e até emojis. Sem esta linha, você poderia ter problemas com acentos aparecendo como símbolos estranhos (por exemplo:

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Esta é a tag de metadados mais importante para design responsivo (sites que se adaptam a celulares). `name="viewport"` diz ao navegador que estamos definindo regras para a “janela de visualização” (a tela do dispositivo). `content="width=device-width"` instrui o navegador a fazer a largura da página ser igual à largura da tela do dispositivo. (Sem isso, o celular tentaria mostrar o site como se fosse um desktop, “dando *zoom out*”). E `initial-scale=1.0` define o nível de zoom inicial para 100% (sem zoom) quando a página carrega.
- `<title>Document</title>`: Define o título da página. O texto “`Document`” (ou o que você colocar dentro da tag) é o que aparece na aba do navegador e é encontrado nos resultados de busca do Google.

3.2.2. A tag `<body>` (Corpo da Página)

A tag `<body>` é o contêiner que define o início e o fim de todo o conteúdo que o usuário vê na janela do navegador. Tudo o que você enxerga em um site — textos, imagens, vídeos, botões, links — está dentro da tag `<body>`. Isso inclui, mas não se limita a:

- **Títulos e Parágrafos:** `<h1>`, `<h2>`, ..., `<h6>`, `<p>`
- **Mídia:** `` (imagens), `<video>`, `<audio>`
- **Links:** `<a>` (links ou âncoras)
- **Listas:** `` (listas com bolinhas), `` (listas numeradas), `` (itens da lista)
- **Formulários:** `<form>`, `<input>`, `<button>`, `<textarea>`
- **Tabelas:** `<table>`, `<tr>`, `<td>`, `<th>`

Regras Fundamentais

- Só podem existir UMA tag `<body>` e `<head>` por documento HTML.
- Elas devem ser “irmãs”, ou seja, ambas estão diretamente dentro de `<html>`.

Veja que as duas tags tem o mesmo alinhamento dentro da tag `html`:

```

1 <html>
2 |
3 |   <head>
4 |   </head>
5 |
6 |   <body>
7 |   </body>
8 |
9 </html>

```

Listing 1: Código-Fonte 3.1: A tag <html> e suas tags filhas head e body

As divisores estruturais do corpo de uma página podem ser realizadas com as tags: <header>, <main> e <footer> como principais e com divisões genéricas, usando as tags <div>, <section>, <article> e <nav> como suas tags filhas. Falaremos mais a frente sobre cada uma delas. Por ora, vamos entender um pouco sobre a tag <header> e as tags para títulos de <h1> até <h6>.

4. Estruturas dentro da tag <body>

4.1. A tag <header>

Observe o código-fonte a seguir. Vamos analisar linha por linha do bloco <header> (o cabeçalho visível da página).

The screenshot shows a code editor with the following annotated code:

```

<!DOCTYPE html>           <!-- Especifica o tipo de documento--&gt;
&lt;html lang="pt-br"&gt;       &lt;!-- Inicia a etiqueta HTML com o idioma--&gt;
  &lt;head&gt;                 &lt;!-- Elemento que registra os meta dados da
    &lt;meta&gt;...             &lt;!-- página--&gt; ...
  &lt;/head&gt;

  &lt;body&gt;
    &lt;header&gt;
      &lt;h1 style="color: blue"&gt; Oficina de Construção de Projetos Web
      &lt;/h1&gt;
      &lt;h3 style="color: #rgb(35, 165, 6)"&gt; Do HTML básico à criação de um Mathlet interativo.
      &lt;/h3&gt;
      &lt;h4 style="color: #rgb(209, 192, 6)"&gt; Professora Dra. Greice Lacerda
      &lt;/h4&gt;
    &lt;/header&gt;
    &lt;hr&gt;
  &lt;/body&gt;
</pre>


Annotations:



- A blue arrow points from the text "Do HTML básico à criação de um Mathlet interativo." to the <h3> tag.
- A yellow arrow points from the text "Professora Dra. Greice Lacerda" to the <h4> tag.
- A large green arrow points from the text "Oficina de Construção de Projetos Web" to the <h1> tag.
- Text annotations above the code:
  - "Oficina de Construção de Projetos Web"
  - "Do HTML básico à criação de um Mathlet interativo."
  - "Professora Dra. Greice Lacerda"

```

A tag <header> é tags semântica do HTML5, que agrupa o conteúdo introdutório de uma página ou de uma seção. Pense nela como o “cabeçalho” visível do seu conteúdo. É muito comum colocar dentro desta tag: o título <h1> principal da página, logotipos, menus de navegação (<nav>) ou subtítulos, como no exemplo.

Importante!

Não confunda a tag `<header>` (que vai no `<body>` e é visível) com a tag `<head>` (que vem antes do `<body>` e contém metadados invisíveis).

Vajamos a estrutura do nosso código e a explicação de cada elemento nele escrito.

```

1 <body>
2
3     <header>
4
5         <h1 style="color:blue">
6             Oficina de Construção de Projetos Web
7         </h1>
8
9         <h3 style="color:green">
10            Do HTML básico à criação de um Mathlet interativo.
11        </h3>
12
13        <h4 style="color:orange">
14            Professora Dra. Greice Lacerda
15        </h4>
16
17     </header>
18
19     <hr>
20
21 </body>

```

Listing 2: Código-Fonte 3.2: A tag `<header>` e suas estruturas filhas no exemplo dado.

A tag `<h1>` é a tag de título de nível 1 (Heading 1). Ela marca o título mais importante da sua página, e é dita **Semântica**, porque é responsável por informar aos navegadores, leitores de tela e ao Google que “**Oficina de Construção de Projetos Web**” é o tópico principal de todo este documento. A boa prática em HTML diz que o ideal é existir apenas uma tag `<h1>` por página, e ela deve ser o primeiro título da página.

As tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>` são todas tags de título (ou cabeçalho, do inglês headings) do HTML. A função principal delas é criar uma hierarquia de importância e uma estrutura de tópicos para o seu conteúdo. Pense nelas não como “etiquetas de texto grande”, mas como a forma de criar um “índice” para a sua página. A melhor maneira de pensar nelas é como os títulos e capítulos de um livro ou de um documento oficial:

- `<h1>`: O Título do Livro. É o título principal e mais importante de toda a página. Define o assunto central.
- `<h2>`: Os Títulos dos Capítulos. Marca o início das seções principais da sua página. Você pode ter vários `<h2>` (vários ”capítulos”).

- <h3>: Os Títulos das Sub-seções. Marca um subtópico dentro de um <h2>.
- <h4>, <h5> e <h6>: Sub-sub-tópicos.

A tag `<hr>` (Linha 19) A tag “Horizontal Rule”. Cria uma linha horizontal para separar visualmente o cabeçalho do resto do conteúdo, que viria a seguir, como o `<main>` e o `<footer>`, que veremos em outro material.

4.2. Breve comentário sobre as tags `<main>` e `<footer>`

Construímos um modelo de “cabeçalho” e no próximo encontro, vamos montar as duas partes que completam o esqueleto de uma página. A tag `<main>` que é o “coração” da página, onde todo o conteúdo principal e mais importante da sua página vai morar. E o `<footer>` que são os “pés”, fechando a página de forma organizada. Discutir como dividir essas áreas é o que realmente transforma uma página simples em um site profissional e bem estruturado. Estou animado para montar esse quebra-cabeça com você!

5. Resumo das Estrutura Estudadas

1. O Esqueleto Básico do HTML5:

`<!DOCTYPE html>`: A declaração que “liga” o modo moderno do HTML5.

`<html>`: A tag raiz que envolve toda a página.

`<head>`: O “cérebro” invisível da página, onde colocamos:

`<meta charset=“UTF-8”>` (Para acentuação correta).

`<meta name=“viewport”>` (Essencial para celulares).

`<title>` (O título que aparece na aba do navegador).

`<body>`: O “corpo” visível da página, onde todo o conteúdo é colocado.

2. Tags Vazias (Void Elements):

`
`: quebra de linha

`<hr>`: linha horizontal

3. O Ponto Principal: Semântica e Hierarquia

`<head>`: é invisível (metadados)

`<header>`: é visível (o cabeçalho do conteúdo da página, como um banner ou título).

`<h1>` ao `<h6>`: definem a estrutura (o “índice”) da sua página.

6. Tarefas para o Próximo Encontro

1. Sugestões de atividades de revisão e estudo do Material Atual:

- Assistir à gravação do vídeo da nossa sessão.

- Ler e revisar o material desta nossa conversa (sobre as tags head, body, header, h1-h6, etc.).
- Examinar o sumário que está localizado na pasta SiteMathFusionTech (dentro da pasta da GreiceLacerda).

2. Sugestões de atividades de práticas para a próxima Aula:

- Pensar e Rascunhar a estrutura de uma nova página inicial de Perguntas e Respostas (FAQ).
- Como vimos no exemplo anterior, definir e digitar no seu arquivo index, o <title> e <header> pensado para a página.