

Programação Web Responsiva

Introdução

Prof. Bruno Azevedo

UNIP – Universidade Paulista



Objetivo

- Ser capaz de desenvolver Sites e Aplicações Web usando linguagens de *Mark Up* e de programação.

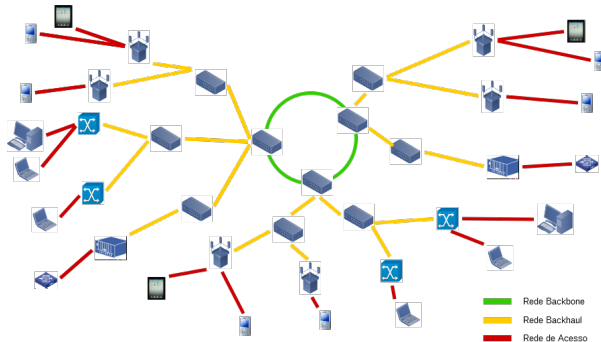
Web × Internet

Qual a diferença entre a Web (World Wide Web) e a Internet?

Web × Internet

Qual a diferença entre a Web (World Wide Web) e a Internet?

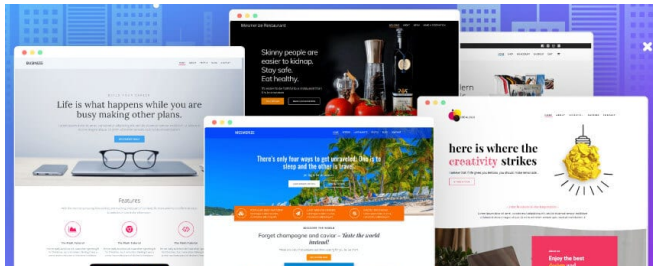
- A Internet é um sistema global de redes de computadores interconectadas que usa um conjunto de protocolos (TCP/IP) para se comunicar entre redes e dispositivos.
- É uma rede de redes que consiste em redes privadas, públicas, acadêmicas, empresariais e governamentais de âmbito local a global.



Web × Internet

Qual a diferença entre a Web (World Wide Web) e a Internet?

- A Web é um sistema baseado na Internet que permite acessar e compartilhar documentos. Utiliza o protocolo HTTP/HTTPS.
- Recursos são disponibilizados por meio de servidores Web e podem ser acessados por programas como os navegadores da Web (Chrome, Firefox, Brave, etc).



Web × Internet

Qual a diferença entre a Web (World Wide Web) e a Internet?

- Sucintamente, **a Internet é uma rede que conecta milhões de computadores pelo mundo, enquanto a Web é um dos serviços que funcionam na Internet.**

A Origem da Internet

- A Internet surgiu como um projeto militar nos anos 60, inicialmente chamada de ARPANET.
- A ARPANET foi criada em 1969, conectando quatro universidades nos EUA.
- Em 1971, foi enviada a primeira mensagem de e-mail.
- Nos anos 80, surgiram os primeiros protocolos de comunicação, como TCP/IP, permitindo a expansão da rede.
- Em 1990, a ARPANET foi desativada, dando lugar à Internet comercial.

A Origem da Web

- A Web foi criada por Tim Berners-Lee em 1989 no CERN, com o objetivo de facilitar o compartilhamento de informações entre cientistas.
- O primeiro site da história foi publicado em 1991, contendo informações sobre o próprio projeto da Web.
- A Web foi construída sobre três tecnologias principais:
 - HTML (HyperText Markup Language): linguagem usada para criar páginas web.
 - HTTP (HyperText Transfer Protocol): protocolo de comunicação entre servidores e navegadores.
 - URL (Uniform Resource Locator): endereçamento único para identificar recursos na Web.

Web 1.0 - A Web Estática

- A Web 1.0 dominou a década de 1990 e era caracterizada por páginas estáticas.
- Sites eram desenvolvidos em HTML puro, sem interatividade significativa.
- O conteúdo era majoritariamente unidirecional: os usuários apenas consumiam informações, sem possibilidade de interação.
- Empresas utilizavam a Web 1.0 como uma forma de presença digital, criando sites institucionais simples.
- Exemplos incluem portais de notícias, blogs iniciais e diretórios como o Yahoo!.

Web 2.0 - A Web Interativa

- Com o início dos anos 2000, surgiu a Web 2.0, marcada por maior interatividade e colaboração.
- O advento de novas tecnologias permitiu a criação de páginas dinâmicas e responsivas.
- Usuários passaram a interagir ativamente com conteúdo, podendo criar e compartilhar informações.
- Redes sociais, blogs, fóruns e wikis se tornaram populares, possibilitando a construção de comunidades online.
- Exemplos incluem Facebook, YouTube e Wikipedia.

Web 3.0 - A Web Semântica

- A Web 3.0 representa a evolução para uma internet mais inteligente e conectada.
- Caracteriza-se pelo uso de Inteligência Artificial e Blockchain.
- Busca estruturar os dados de forma semântica, permitindo que máquinas compreendam melhor o significado da informação.
- Exemplos incluem criptomoedas, assistentes virtuais como Alexa e Siri, além de plataformas descentralizadas como a Ethereum.

O Futuro?

- A Internet e a Web evoluíram rapidamente, transformando nossa forma de interagir com o mundo digital.
- A Web 1.0 era estática e informativa, a Web 2.0 trouxe interatividade e colaboração, e a Web 3.0 introduziu inteligência e descentralização.
- Novas inovações continuam a surgir, redefinindo a maneira como acessamos e compartilhamos conhecimento.

Navegamos na Web

- Para navegar na Web, usamos um programa capaz de exibir imagens, textos, áudios e vídeos.
- O ato de clicar em links, visitar sites e interagir com o conteúdo é chamado de “**navegar**”.
- A navegação é feita por um **Navegador Web**.

Os Navegadores da Web

O que são Navegadores Web?

Os Navegadores são programas que permitem aos usuários acessar páginas da Web, interpretando e renderizando código HTML, CSS e JavaScript.

- Os navegadores transformam código em interfaces visuais interativas.
- Exemplos populares: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari e Opera.
- Cada navegador pode ter pequenas diferenças na renderização das páginas.
- Mas onde estão essas páginas que “navegamos”? Onde elas se encontram?

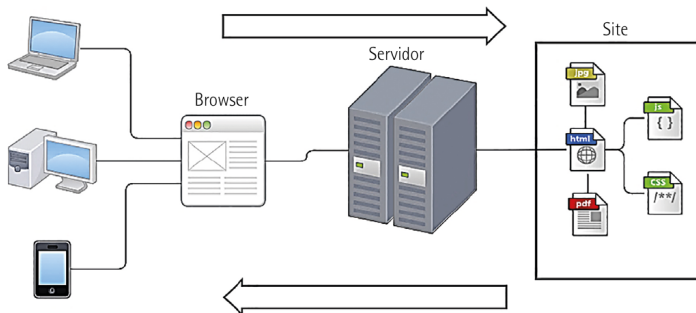
Os Servidores Web

O que são Servidores Web?

Servidores Web são softwares responsáveis por armazenar e entregar páginas da Web para os navegadores quando solicitadas pelos usuários.

- Recebem requisições HTTP e retornam respostas HTTP com conteúdo Web.
- Podem hospedar páginas estáticas (HTML, CSS) e dinâmicas (PHP, Node.js, Python).
- Principais servidores Web: Apache, Nginx, Microsoft IIS e LiteSpeed.

Os Servidores Web



Páginas Web

- O que constituem páginas Web? Vocês estão acostumado a visitar páginas, mas o que está por trás?
- Quais os principais elementos que compõem uma página Web?

Páginas Web

- O que constituem páginas Web? Vocês estão acostumado a visitar páginas, mas o que está por trás delas?
- Quais são os elementos principais compõem uma página Web?
- Documentos HTML, Folhas de Estilo CSS, Arquivos de Script JavaScript são alguns destes elementos.
 - HTML (Hypertext Markup Language). **Estrutura da página.**
 - CSS (Cascading Style Sheets). **Apresentação da página.**
 - JavaScript. **Interatividade da página.**

HTML

- HTML significa HyperText Markup Language, traduzindo, Linguagem de Marcação de Hipertexto.
- Ao contrário de quando criamos um documento no Word, temos que dizer o que cada elemento é.
- O Word permite a criação WYSIWYG (acrônimo para “o que você vê é o que você obtém”).
- Já o HTML é uma linguagem de **Mark Up** (Marcação) e a criação é feita de um modo diferente.

HTML

- As linguagens de marcação são usadas para criar documentos legíveis por máquina e por humanos.
- Consistem em elementos de marcação, que são Tags especiais inseridas no texto para indicar a estrutura e o significado dos elementos (semântica).
- No HTML, as Tags são usadas para definir elementos como parágrafos, listas, links e imagens, entre outros.

HTML

- Para escrevermos em HTML usamos **Tags**.
- Uma Tag HTML consiste em seu nome entre colchetes angulares (< >).
- Geralmente, o nome da Tag aparece em sua abertura e novamente em seu fechamento precedido por uma barra (/), mas existem exceções.
- Por exemplo:

```
<p> Na disciplina de Programação Web Responsiva aprendemos  
o desenvolvimento Web responsivo. </p>
```
- Aqui temos um **conteúdo** entre Tags de parágrafo (<p>).

Atributos HTML

- Todas as Tags de HTML podem possuir atributos.
- Atributos fornecem informações adicionais sobre elementos.
- Geralmente são compostos pelo par `nome="valor"`.
- Por exemplo, no código abaixo, `src` é um **atributo** da Tag `img`, que define o nome da imagem a ser exibida na página.

```

```

Tags HTML

- Atenção! No HTML, a capitalização dos nomes dos elementos não é importante. Portanto, ``, `` e `` são todos iguais para o navegador.
- Mas a maioria dos desenvolvedores prefere a consistência de escrever os nomes dos elementos em minúsculas.

Estrutura Básica de um Documento HTML

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Metadados da página -->
  </head>
  <body>
    <!-- Conteúdo da página -->
  </body>
</html>
```


Estrutura Básica de um Documento HTML

`<!DOCTYPE html>`

- Não é uma Tag HTML.
- É uma declaração de tipo de documento que permite aos navegadores modernos saber qual especificação HTML usar para interpretar o documento.
- Esta declaração DOCTYPE identifica o documento como escrito em HTML.

Estrutura Básica de um Documento HTML

```
<html> </html>
```

- Todo o documento está contido dentro de um elemento html.
- O elemento html é chamado de elemento raiz porque contém todos os elementos no documento, e não pode ser contido dentro de nenhum outro elemento.
- Dentro do elemento html, o documento é dividido em uma cabeça (head) e um corpo (body).

Estrutura Básica de um Documento HTML

`<head> </head>`

- O elemento head contém elementos que se referem ao documento e não são renderizados como parte do conteúdo.
- Por exemplo, o título do documento (não confundir com títulos no conteúdo), folhas de estilo, scripts e metadados em geral.

Estrutura Básica de um Documento HTML

`<body> </body>`

- O elemento body contém tudo o que queremos que seja mostrado na janela do navegador.
- É o conteúdo **visível** da página.
- Formatado com as Tags HTML.

Exemplo de uma Página em HTML

- Vamos ver um exemplo:

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Minha Primeira Página!</title>
  </head>
  <body>
    <h1>Criei um título!</h1>
    
  </body>
</html>
```

Exemplo de uma Página em HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Minha Primeira Página!</title>
  </head>
  <body>
    <h1>Criei um título!</h1>
    
  </body>
</html>
```

- O atributo lang na Tag <html> define o **idioma principal do documento**.
- <html lang="pt-BR"> define o idioma do documento como português do Brasil.
- Importante para acessibilidade, pois leitores de tela ajustam a pronúncia corretamente.

Exemplo de uma Página em HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Minha Primeira Página!</title>
  </head>
  <body>
    <h1>Criei um título!</h1>
    
  </body>
</html>
```

- O atributo charset na Tag <meta> define a **codificação de caracteres** da página.
- UTF-8 é uma codificação universal que suporta acentos, caracteres especiais e múltiplos idiomas.
- Evita problemas como caracteres estranhos aparecendo no texto.

Exemplo de uma Página em HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Minha Primeira Página!</title>
  </head>
  <body>
    <h1>Criei um título!</h1>
    
  </body>
</html>
```

- A Tag <title></title> define o **título da página** exibido na aba do navegador.

Exemplo de uma Página em HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Minha Primeira Página!</title>
  </head>
  <body>
    <h1>Criei um título!</h1>
    
  </body>
</html>
```

- A Tag `h1` e `img` definem um título e uma imagem que serão exibidos na página.

Primeiras Tags – Conteúdo e Estruturação

- A Tag `<p>` é usada para criar parágrafos de texto.
- As Tags `<h1>` até `<h6>` são usadas para definir títulos e subtítulos com diferentes níveis de importância.
- A Tag `` deixa o texto em negrito para indicar importância.

Primeiras Tags – Conteúdo e Estruturação

- A Tag `` aplica ênfase ao texto, geralmente em itálico.
- A Tag `
` é usada para criar quebras de linhas.
- A Tag `<hr>` é usada para criar uma linha horizontal. Podemos usar para separar conteúdo.

Tag de Imagem – Conteúdo e Estruturação

- A Tag é usada para inserir imagens em uma página web.
- O atributo src especifica o caminho da imagem.
- O atributo alt fornece uma descrição alternativa da imagem, importante para acessibilidade.

Exemplo:

```

```

Comentários em HTML

- Os comentários em HTML são inseridos entre `<!--` e `-->`.
- Eles não são exibidos no navegador.
- São utilizados para explicar o código ou adicionar observações sobre a estrutura da página.
- Exemplo de um comentário em HTML:

```
<!-- Este é um comentário em HTML -->
```

Importância dos Comentários em HTML

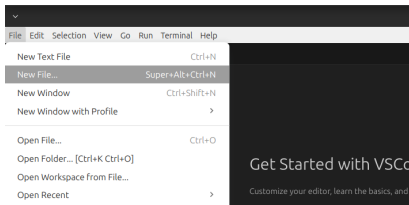
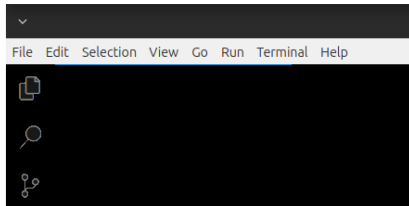
- Comentários tornam o código mais fácil de entender.
- Ao retornar ao código, os comentários ajudam a lembrar o propósito de diferentes partes.
- Quando vários desenvolvedores trabalham no mesmo projeto, os comentários ajudam a explicar as decisões de código.
- Exemplo de uso prático de comentários:

```
<!-- Criei um parágrafo aqui embaixo. -->  
<p> Isso é um parágrafo! </p>  
<!-- O texto abaixo está em itálico! -->  
<em> Olá, tudo bem? </em>
```

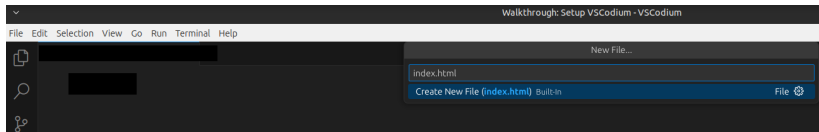
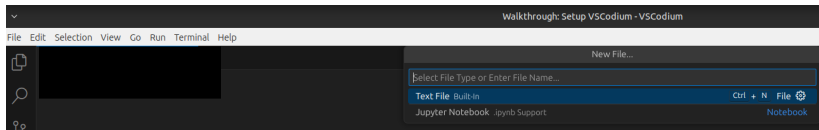
Primeira Página Web

- Nesta aula, daremos início ao nosso primeiro projeto.
- Um dos editores mais utilizados é o Visual Studio Code (VSCode), disponível gratuitamente e que também possui versão Web (`vscode.dev`).
- Vocês podem usar a versão instalada ou a versão Web.
- Vamos explorar as primeiras Tags e elementos essenciais para estruturar uma página HTML.

Criando Nossa Primeira Página Web com o VSCode



Criando Nossa Primeira Página Web com o VSCode



Criando Nossa Primeira Página Web com o VSCode

- Salvem após editar usando as teclas CTRL+S ou clique em “Save” ou “Salvar” no menu).
- Para visualizar a página: abram o arquivo `index.html` criado efetuando um duplo clique em seu ícone.

Atividade Prática 1

- ❶ Criar uma página HTML simples contendo:
 - Um título principal `<h1>`.
 - Um parágrafo `<p>` descrevendo o que entenderam sobre a Web.
 - Uma imagem representando a evolução da internet.
 - Ampliação: incluir mais conteúdo na página incluindo textos e imagens.
- ❷ Cada aluno deve produzir um relatório de 1 a 3 páginas contendo:
 - Resumo teórico: explicação sobre a estrutura de um documento HTML.
 - Código-fonte comentado: cada linha do código deve conter um comentário explicando sua função.

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.