

Unidade Curricular		Turma
Recife	PROGRAMAÇÃO PARA INTERNET - N 279	N2

Docente Responsável	Titulação
Cloves Alves da Rocha	Mestre

Carga Horária / Créditos				
Carga Horária Teórica Presencial	40		Carga Horária Prática	40
Carga Horária Estágio	0		Carga Horária Extensão	0
Carga Horária EAD	0		Carga Horária Pesquisa	0
Carga Horária Total	80		Créditos	4

Carga Horária	Dia	Horário
EAD		
Presencial	Segunda-feira	19:00 às 22 h

Ementa

Estudo dos princípios e técnicas para o desenvolvimento de aplicações web, abordando as tecnologias essenciais para o frontend e o backend. A disciplina cobre HTML, CSS, JavaScript, frameworks de frontend populares (como React, Angular ou Vue.js), programação do lado do servidor (usando Node.js, Django, Flask ou Ruby on Rails), e integração com bancos de dados. Também serão abordados conceitos de responsividade, acessibilidade, SEO, e segurança na web.

Tópicos Geradores

1. Fundamentos da Web: História da Internet, funcionamento da web, protocolos HTTP/HTTPS.
2. Desenvolvimento Frontend: HTML5, CSS3, JavaScript, jQuery, frameworks (React/Angular/Vue.js).
3. Desenvolvimento Backend: Introdução ao Node.js, Python (Django, Flask) ou Ruby on Rails, APIs RESTful, autenticação e autorização.
4. Bancos de Dados para Web: Integração com SQL (PostgreSQL, MySQL) e NoSQL (MongoDB), ORM.
5. Práticas Modernas: Desenvolvimento responsivo, acessibilidade, fundamentos de SEO, segurança web (OWASP).
6. Ferramentas de Desenvolvimento e Deployment: Git, Docker, serviços de cloud (AWS, Azure, Google Cloud), CI/CD.
7. Projeto Prático: Desenvolvimento em grupo de uma aplicação web do início ao fim, aplicando as técnicas e tecnologias aprendidas.

Habilidades

1. Desenvolver páginas web utilizando HTML, CSS e JavaScript, aplicando as melhores práticas de codificação e design responsivo.
2. Utilizar frameworks de frontend para acelerar o desenvolvimento e melhorar a experiência do usuário.
3. Implementar lógica de servidor, manipulação de dados e integração com bancos de dados em aplicações web.
4. Aplicar conceitos de segurança, otimização de performance e SEO em projetos web.



Desenvolvimento de Projetos WEB em grupo, Simulação e Sala de aula invertida.

Cronograma de Atividades – Aulas Teóricas Presenciais +Práticas

Semana		Tema
Semana 1	04.08 segunda	AULA 0 - Apresentação da Disciplina e Docente. AVA + Ementa + Plano de Ensino e Plano de Aula VT DA DISCIPLINA DINÂMICAS E DESAFIOS EM GRUPOS. AULA 1- Introdução à Web e HTML Básico. Parte 1 de 2.
Semana 2	11.08 segunda	AULA 1- Introdução à Web e HTML Básico. Teoria: História da Internet, funcionamento da web (clientes, servidores), protocolos HTTP/HTTPS, conceitos de navegador. Estrutura básica do HTML5 (tags semânticas, atributos, elementos, head vs. body). Prática: Criar a primeira página web simples com estrutura HTML básica. Habilidades: Reconhecer a estrutura fundamental de uma página web. Atitudes: Curiosidade sobre como a web funciona.

Aula 1: Introdução à Web e HTML Básico

Descobrimos os pilares da internet e construindo sua primeira página web

Docente: Cloves Rocha

A História da Internet e da Web

1960s-1980s: ARPANET e o Início

A base da internet foi desenvolvida pela Agência de Projetos de Pesquisa Avançada (ARPA) dos EUA, com o objetivo de criar uma rede de comunicação robusta e descentralizada. Os primeiros passos incluíram a troca de mensagens e o desenvolvimento de protocolos básicos.

1

1990s: Expansão e Navegadores

A década de 90 viu a explosão da web com o lançamento de navegadores gráficos como o Mosaic e o Netscape Navigator. A web deixou de ser um recurso exclusivo para pesquisadores e se tornou acessível ao público em geral, impulsionando o comércio eletrônico e a comunicação global.

3

2014: HTML5 e a Web Moderna

A padronização do HTML5 trouxe novas capacidades multimídia, semântica aprimorada e suporte a aplicações offline. Esta versão solidificou a web como uma plataforma global robusta, adaptável a diversos dispositivos e que conecta bilhões de pessoas diariamente.

5

2

1989: O Nascimento da World Wide Web

Tim Berners-Lee, um cientista do CERN (Organização Europeia para a Pesquisa Nuclear), propôs um sistema de gerenciamento de informações baseado em hipertexto. Este sistema, que se tornaria a World Wide Web, tinha como objetivo facilitar o compartilhamento de documentos entre pesquisadores.

4

2000s: Web 2.0 e Redes Sociais

A ascensão da "Web 2.0" marcou uma mudança para aplicações mais interativas e colaborativas. Plataformas como Facebook, YouTube e Wikipedia transformaram a web em um ambiente onde os usuários não apenas consumiam, mas também criavam e compartilhavam conteúdo.

A história da internet e da web é uma jornada fascinante de inovação contínua. Desde suas origens militares e acadêmicas até se tornar a espinha dorsal da comunicação global e da economia digital, a web tem moldado profundamente a sociedade, permitindo que bilhões de pessoas se conectem, aprendam e colaborem de formas nunca antes imaginadas. Essa evolução contínua é um testemunho da curiosidade e da engenhosidade humana.

Como Funciona a Web: Clientes, Servidores e Protocolos



Navegadores (Clientes) e Servidores

No coração da web está a relação cliente-servidor. Os navegadores web (seus "clientes"), como Chrome, Firefox ou Edge, atuam como seu intermediário para o mundo digital. Quando você digita um endereço ou clica em um link, seu navegador envia uma "solicitação" por meio da internet para um "servidor".

Um servidor é um computador poderoso que armazena os arquivos de um website (HTML, CSS, JavaScript, imagens, etc.). Ele está sempre ligado, aguardando solicitações de navegadores de todo o mundo. Ao receber uma solicitação, o servidor processa e envia os arquivos necessários de volta ao navegador.

Protocolos: HTTP e HTTPS

Para que clientes e servidores se entendam, eles usam "protocolos" – conjuntos de regras para a comunicação. O mais fundamental para a web é o **HTTP** (Hypertext Transfer Protocol). Ele define como as mensagens são formatadas e transmitidas, permitindo a troca de informações.

A evolução do HTTP é o **HTTPS** (Hypertext Transfer Protocol Secure). A diferença crucial é a segurança: o HTTPS criptografa a comunicação entre seu navegador e o servidor, protegendo dados sensíveis como senhas, informações de cartão de crédito e dados pessoais de interceptações maliciosas. É por isso que você sempre deve procurar o "cadeado" na barra de endereço ao navegar.

Em resumo, quando você acessa uma página web, seu navegador (cliente) solicita os arquivos ao servidor usando HTTP ou HTTPS. O servidor responde enviando esses arquivos, e o navegador então os interpreta para exibir a página visual que você vê.

O Papel do Navegador Web



Intérprete de Código

O navegador é o seu principal software para interagir com a web. Ele recebe os arquivos (HTML, CSS, JavaScript) enviados pelo servidor e tem a tarefa de "interpretar" esse código. Pense nele como um tradutor simultâneo que converte as instruções escritas em uma linguagem de programação para algo visualmente compreensível.



Renderização Visual

A principal função do navegador é transformar o código HTML (estrutura), CSS (estilo) e JavaScript (interatividade) em uma página web visual e interativa. Ele constrói a página elemento por elemento, aplicando cores, fontes, layouts e comportamentos para que você possa ver e clicar, e não apenas um amontoado de texto e tags.



Ocultando as Tags

Um ponto crucial é que o navegador não exibe as tags HTML na página final. Em vez disso, ele usa essas tags como instruções. Por exemplo, a tag `<p>` não aparece na tela; ela instrui o navegador a criar um novo parágrafo com o texto contido dentro dela. Isso permite uma experiência de usuário limpa e intuitiva, onde o foco está no conteúdo.



Motores de Renderização

Cada navegador possui seu próprio "motor de renderização" (por exemplo, Blink no Chrome, Gecko no Firefox, WebKit no Safari), que é o software responsável por essa interpretação e exibição. Embora todos sigam os mesmos padrões web, pequenas diferenças na implementação podem levar a variações sutis na forma como uma página é exibida em navegadores diferentes.

Estrutura Básica de um Documento HTML5

`<!DOCTYPE html>`

Esta é a primeira linha de qualquer documento HTML5. Ela não é uma tag HTML, mas uma "declaração de tipo de documento". Sua função é informar ao navegador que a página está escrita de acordo com a especificação HTML5. Isso garante que o navegador renderize a página no "modo padrão", evitando comportamentos inesperados.

`<head>...</head>`

O elemento `<head>` contém metadados sobre a página – informações que não são visíveis diretamente no navegador, mas são cruciais para seu funcionamento. Aqui incluímos o título da página `<title>` (que aparece na aba do navegador), a codificação de caracteres `<meta charset="UTF-8">` (garantindo que acentos e caracteres especiais sejam exibidos corretamente), links para folhas de estilo CSS e scripts JavaScript.

`<html lang="pt-br">...</html>`

Este é o "elemento raiz" de toda página HTML. Ele encapsula todo o conteúdo do documento, exceto a declaração DOCTYPE. O atributo `lang="pt-br"` é importante para indicar o idioma principal do conteúdo da página (no caso, Português do Brasil). Isso auxilia navegadores, leitores de tela e motores de busca.

`<body>...</body>`

O elemento `<body>` é onde todo o conteúdo visível da sua página web reside. Tudo o que o usuário vê na tela – texto, imagens, vídeos, links, botões, formulários – deve estar dentro desta seção. É o corpo principal onde você construirá a interface e o conteúdo interativo da sua página.

Compreender essa estrutura é o primeiro passo para criar qualquer página web. Ela fornece o arcabouço necessário para organizar o conteúdo de forma lógica e semântica, facilitando a interpretação tanto pelos navegadores quanto pelos desenvolvedores.

Tags Semânticas e Elementos Essenciais

Títulos (` `)

` a `

`)

Os elementos de título são fundamentais para a estrutura do seu conteúdo. Eles vão de `<h1>` (o título mais importante da página) a `<h6>` (o menos importante). O uso correto dos títulos não é apenas para estilo visual; ele ajuda os motores de busca a entender a hierarquia e o assunto da sua página, além de melhorar a acessibilidade para usuários de leitores de tela.

Parágrafos (``)

O elemento `<p>` é o bloco fundamental para qualquer texto corrido. Ele cria um parágrafo com margens padrão, separando blocos de texto para maior legibilidade. É a base para organizar o conteúdo textual em sua página.

O uso adequado dessas tags e atributos semânticos não apenas organiza o conteúdo visualmente, mas também o torna compreensível para máquinas (como motores de busca e leitores de tela), garantindo que sua página seja acessível e bem indexada na web.

Ênfase e Importância (` ` , ``)

Embora pareçam semelhantes à primeira vista (ambos podem deixar o texto em itálico ou negrito), `` e `` têm significados semânticos distintos:

- `` (*emphasis*): Indica ênfase textual, como se você estivesse enfatizando uma palavra ao falar.
- `` (*strong importance*): Indica que o conteúdo tem grande importância, gravidade ou urgência.

Usar as tags corretas para o propósito certo melhora a semântica do seu código, beneficiando a acessibilidade e o SEO.

Atributos

Atributos são informações adicionais fornecidas dentro da tag de abertura de um elemento HTML. Eles modificam o comportamento ou as características do elemento. Por exemplo, o atributo `href` no elemento `<a>` (âncora) especifica o URL para o qual o link aponta: `Google`. Outros atributos comuns incluem `src` para imagens, `alt` para texto alternativo e `class` ou `id` para estilização.

Diferença entre Tags, Elementos e Atributos

Tag

Uma tag HTML é uma palavra-chave entre colchetes angulares (< e >). As tags são a base da sintaxe HTML e geralmente vêm em pares: uma tag de abertura (e.g., <p>) e uma tag de fechamento (e.g., </p>). A tag de fechamento sempre inclui uma barra (/) antes do nome da tag. Existem também "tags vazias" que não possuem conteúdo nem tag de fechamento, como e
.

Exemplo: <p> (tag de abertura), </p> (tag de fechamento).

Elemento

Um elemento HTML é composto por uma tag de abertura, seu conteúdo e uma tag de fechamento. Ele representa uma parte completa da estrutura do documento. O conteúdo é o que será exibido ou processado pelo navegador.

Exemplo: <p>Este é um parágrafo.</p>
Aqui, "Este é um parágrafo." é o conteúdo.

Mesmo tags vazias como são consideradas elementos, mesmo que não tenham conteúdo entre as tags de abertura e fechamento explícitas.

Atributo

Atributos fornecem informações adicionais sobre um elemento HTML. Eles são sempre especificados na tag de abertura e geralmente vêm em pares de nome/valor, como nome="valor". Os atributos podem modificar o comportamento, a aparência ou o significado semântico de um elemento.

Exemplo: No elemento Link, href é o nome do atributo e "pagina.html" é o valor do atributo, indicando para onde o link aponta. Outros exemplos incluem src para imagens, alt para texto alternativo, id e class para CSS/JavaScript.

Compreender a distinção entre esses três conceitos é fundamental para escrever HTML correto e eficaz. As tags definem o tipo de elemento, os elementos encapsulam o conteúdo e os atributos fornecem os detalhes e as configurações para esses elementos.

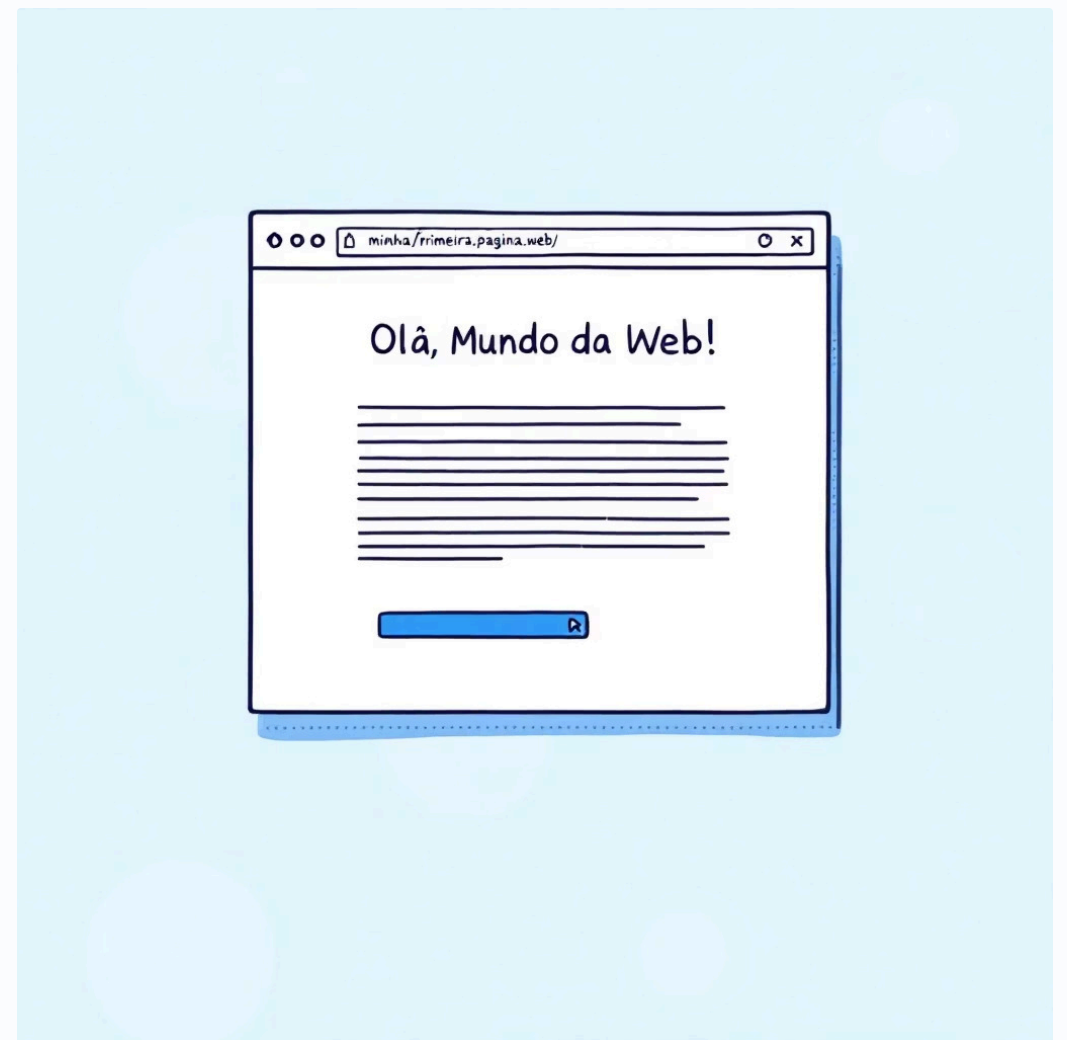
Exemplo Prático: Código da Primeira Página HTML

Chegou a hora de colocar a mão na massa! Este é o código da sua primeira página web. Copie e cole este código em um editor de texto simples (como Bloco de Notas no Windows, TextEdit no Mac - certifique-se de salvar como texto puro, ou VS Code, Sublime Text, etc.).

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Minha Primeira Página Web</title>
</head>
<body>
  <h1>Olá, Mundo da Web!</h1>
  <p>Bem-vindo à minha primeira página HTML.</p>
  <p>Estou aprendendo sobre a estrutura básica da web.</p>
  <a href="https://www.w3schools.com/html/"
target="_blank">Aprenda mais sobre HTML</a>
</body>
</html>
```

Instruções:

- Cole o código em um editor de texto.
- Salve o arquivo como `index.html` (ou outro nome com a extensão `.html`).
- Abra o arquivo salvo diretamente no seu navegador web (clique duplo no arquivo).



O que você deve observar:

- O título da aba do navegador é "Minha Primeira Página Web", conforme definido na tag `<title>`.
- O texto "Olá, Mundo da Web!" aparece grande e como um título principal (definido por `<h1>`).
- Os parágrafos de texto aparecem em blocos separados, graças às tags `<p>`.
- Você verá um link clicável que o levará para a página do W3Schools sobre HTML quando clicado. O `target="_blank"` faz com que ele abra em uma nova aba.

Habilidade Adquirida: Você agora é capaz de reconhecer a estrutura fundamental de uma página web e criar um documento HTML básico que um navegador pode interpretar e exibir.

Atitude Estimulada: A curiosidade sobre como a web funciona e a vontade de explorar mais a fundo as possibilidades do HTML!