

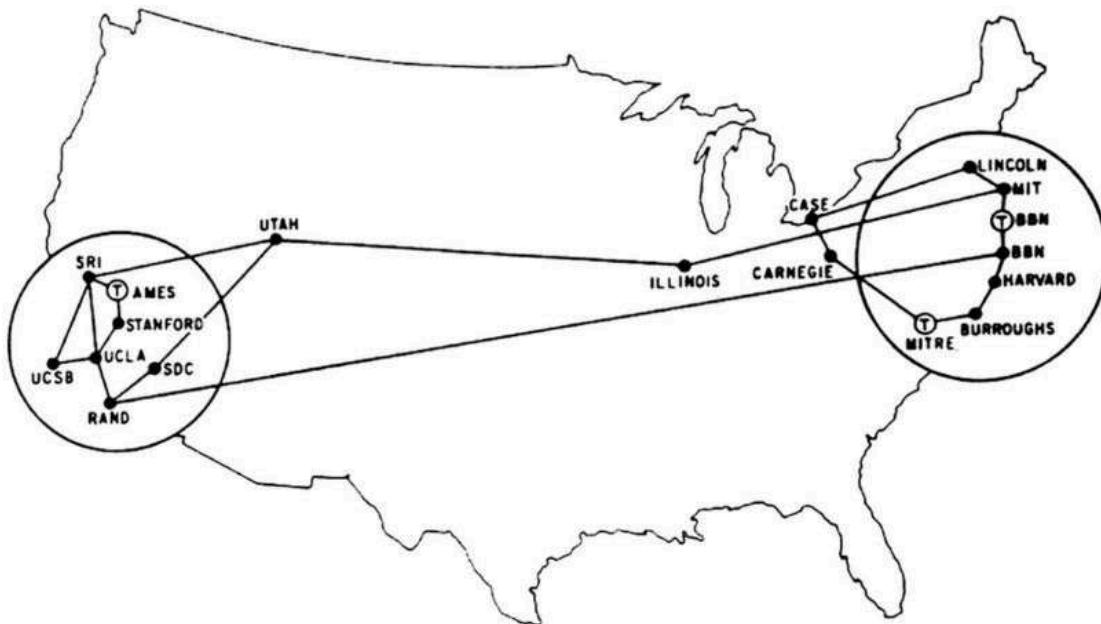
1. Introdução à Internet e Conceitos de Rede

A Internet é uma rede global de comunicação que revolucionou a forma como interagimos, compartilhamos informações e conduzimos negócios. Neste capítulo, exploraremos os fundamentos da Internet, examinando sua história, funcionamento e arquitetura. O objetivo é fornecer uma visão abrangente dos princípios que subjazem essa poderosa infraestrutura de comunicação.

1.1. História da Internet

A história da Internet remonta às origens da computação e da comunicação eletrônica. Seu desenvolvimento é frequentemente associado ao surgimento da ARPANET, projeto financiado pela ARPA (Advanced Research Projects Agency) do Departamento de Defesa dos Estados Unidos na década de 1960 (ABBATE, 1999). A Arpanet (Figura 1) foi a primeira rede a usar um protocolo de comutação de pacotes, uma inovação que permitia que os dados fossem transmitidos em pacotes discretos em vez de um fluxo contínuo.

Figura 1 - Mapa com pontos, destinos e conexões da Arpanet inicial



Fonte: Site BBC

Com o desenvolvimento da Arpanet, outras redes foram criadas, formando uma interconexão chamada de "Internet", e cada vez mais utilizando o Transmission Control Protocol/Internet Protocol (TCP/IP) (CERF; KAHN, 1974). A Internet se expandiu gradualmente, conectando instituições acadêmicas, governos e, posteriormente, corporações e o público.

A Internet é uma rede descentralizada construída em um modelo de arquitetura cliente-servidor. Dispositivos conectados à Internet, como computadores, servidores, roteadores e dispositivos móveis, atuam como clientes, solicitando e recebendo informações de servidores que armazenam e fornecem conteúdos e serviços (TANENBAUM; WETHERALL, 2011).

A arquitetura da Internet é baseada em uma estrutura em camadas de redes interconectadas. No topo dessa hierarquia está o "backbone" global, a rede de alta capacidade responsável por rotear o tráfego entre as regiões geográficas. Abaixo da rede backbone estão as redes regionais, como universidades e redes corporativas, que se conectam à rede backbone e distribuem o tráfego para as redes locais.

Essa arquitetura distribuída garante a redundância e a resiliência da Internet, permitindo que o tráfego encontre rotas alternativas em caso de falha de um trecho da rede (TANENBAUM; WETHERALL, 2011).

1.2. Protocolos de Rede: TCP/IP, HTTP, HTTPS

Os protocolos de rede desempenham um papel fundamental na comunicação e interconexão de dispositivos em ambientes computacionais. Neste capítulo, abordaremos três protocolos essenciais: TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (Hypertext Transfer Protocol) e HTTPS (Hypertext Transfer Protocol Secure). Esses protocolos são amplamente utilizados na Internet e são responsáveis por possibilitar a transferência de dados e o acesso a recursos na rede mundial.

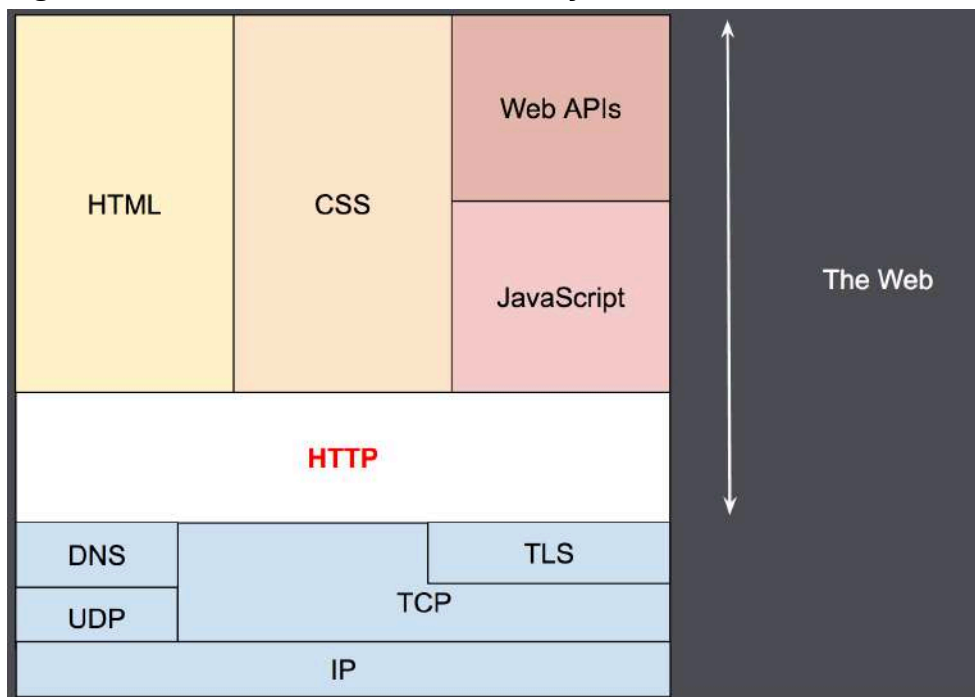
O TCP/IP é a base sobre a qual a Internet foi construída. Ele consiste em dois protocolos principais: TCP, que oferece um serviço de entrega confiável

de pacotes, e IP, que fornece endereçamento e roteamento de dados entre dispositivos em uma rede (TANENBAUM; WETHERALL, 2011).

O TCP/IP é conhecido por sua capacidade de interconectar redes heterogêneas, permitindo que diferentes dispositivos, sistemas operacionais e arquiteturas de hardware comuniquem-se de forma eficiente (COMER, 2006). Ele é um dos protocolos mais utilizados no mundo, sendo fundamental para a comunicação de dispositivos em redes locais e na Internet global.

O HTTP é o protocolo utilizado para a comunicação entre clientes e servidores na World Wide Web. Ele possibilita a recuperação de recursos, como páginas web, imagens e outros arquivos, através de requisições feitas pelos navegadores (clientes) a servidores web (TANENBAUM; WETHERALL, 2011), conforme pode ser observado na Figura 2:

Figura 2 - Protocolo HTTP e comunicação com a Web



Fonte: Site MDN (Mozilla Developer Network)

O HTTP é baseado em uma arquitetura cliente-servidor, onde os clientes enviam solicitações HTTP e os servidores respondem com os recursos solicitados. Cada solicitação e resposta HTTP é composta por cabeçalhos e,

opcionalmente, um corpo, que contém os dados transmitidos (FIELDING et al., 1999).

O HTTPS é uma extensão do protocolo HTTP que adiciona uma camada de segurança à transferência de dados. Ele utiliza criptografia SSL/TLS (Secure Socket Layer/Transport Layer Security) para proteger as informações transmitidas entre o cliente e o servidor, garantindo a confidencialidade e integridade dos dados (RESNICK, 1996).

A adoção do HTTPS tornou-se cada vez mais importante para garantir a segurança e privacidade dos usuários na web, especialmente em transações que envolvem informações sensíveis, como senhas, dados de pagamento e informações pessoais.

1.3. Domínios, hospedagem e URLs

Os domínios são elementos-chave da estrutura da Internet e desempenham um papel crucial na identificação de recursos online. De acordo com Berners-Lee; Fielding; Masinter(2005), os domínios são sequências alfanuméricas utilizadas para identificar e localizar entidades na web. Cada domínio é único e pode representar sites, serviços ou outros recursos na Internet.

Os domínios são organizados em uma hierarquia, com a hierarquia mais alta à esquerda, sendo o "Top-Level Domain" (TLD), e a hierarquia mais específica à direita, representando o "Second-Level Domain" (SLD) e, opcionalmente, o "Third-Level Domain" (TLDD). Por exemplo, na URL "www.example.com", ".com" é o TLD, "example" é o SLD e "www" é o TLDD.

A hospedagem de sites refere-se ao serviço que permite a disponibilização de recursos, como páginas web, imagens e outros arquivos, na Internet. De acordo com Pollock (2021), a hospedagem de sites é realizada por provedores de hospedagem, que oferecem servidores e recursos de armazenamento para que os sites possam ser acessados globalmente.

Os provedores de hospedagem podem oferecer diferentes tipos de hospedagem, como hospedagem compartilhada, VPS (Virtual Private Server)

e hospedagem dedicada. Cada tipo de hospedagem oferece níveis diferentes de desempenho e controle, de acordo com as necessidades do cliente.

As URLs (Uniform Resource Locators) são utilizadas para localizar recursos específicos na web (BERNERS-LEE; FISCHETTI, 2000). Elas consistem em diferentes partes que descrevem o protocolo utilizado, o domínio e o caminho para o recurso específico. Por exemplo, na URL "https://www.example.com/pagina", "https://" é o protocolo utilizado, "www.example.com" é o domínio e "/pagina" é o caminho para o recurso específico.

As URLs permitem que os usuários acessem recursos na web de forma fácil e eficiente, fornecendo uma estrutura padronizada para a localização de recursos.

1.4. Introdução à Arquitetura Cliente-Servidor

A arquitetura cliente-servidor é um dos modelos fundamentais de comunicação e distribuição de recursos em sistemas computacionais. Neste capítulo, exploraremos os conceitos básicos dessa arquitetura, compreendendo o funcionamento da interação entre os componentes cliente e servidor, bem como suas vantagens e aplicações em diferentes cenários. A compreensão dessa arquitetura é essencial para a construção de sistemas escaláveis, eficientes e confiáveis.

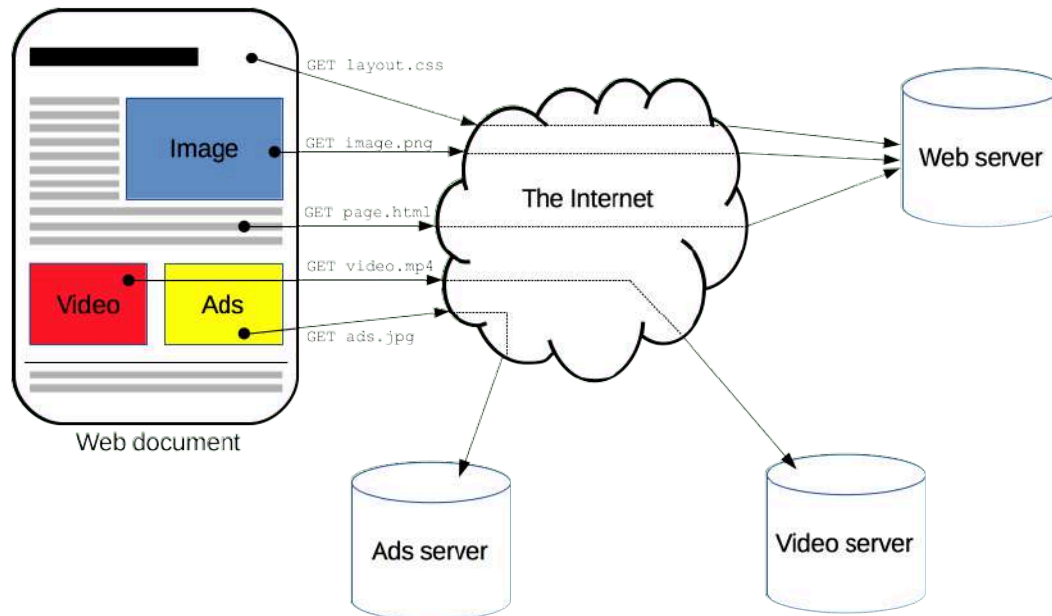
A arquitetura cliente-servidor é um modelo de comunicação que envolve dois tipos principais de entidades: o cliente e o servidor. Segundo Tanenbaum e Wetherall (2011), o cliente é um componente que solicita recursos ou serviços a um servidor, enquanto o servidor é responsável por fornecer os recursos ou serviços solicitados pelo cliente.

A comunicação na arquitetura cliente-servidor é baseada em trocas de mensagens, onde o cliente envia solicitações ao servidor e este responde com as informações ou ações requisitadas. Essa interação acontece de forma assíncrona, permitindo a comunicação entre clientes e servidores em diferentes localizações geográficas.

A interação entre cliente e servidor é baseada em protocolos de comunicação, como o protocolo HTTP para a web (TANENBAUM; WETHERALL, 2011). Quando um cliente deseja acessar um recurso ou

serviço, ele envia uma solicitação ao servidor, contendo informações sobre a ação desejada (Figura 3). O servidor recebe a solicitação, processa-a e envia uma resposta ao cliente, contendo os dados ou informações solicitadas.

Figura 3 - Solicitações realizadas pelo lado do cliente



Fonte: Site MDN (Mozilla Developer Network)

A arquitetura cliente-servidor é escalável, permitindo que múltiplos clientes se comuniquem com o mesmo servidor simultaneamente, e vice-versa. Além disso, o modelo cliente-servidor possibilita a centralização de recursos e serviços em servidores dedicados, simplificando a administração e o gerenciamento de sistemas complexos.

A arquitetura cliente-servidor oferece diversas vantagens em relação a outras abordagens de comunicação em sistemas distribuídos. Uma das principais vantagens é a modularidade, permitindo que os componentes cliente e servidor sejam desenvolvidos e mantidos de forma independente, o que Coulouris et al. (2012).

Essa arquitetura é amplamente utilizada em sistemas web, onde os navegadores atuam como clientes, solicitando e renderizando páginas web servidas por servidores web. Além disso, aplicativos móveis, sistemas de gerenciamento de bancos de dados e muitos outros serviços se beneficiam da estrutura cliente-servidor para oferecer interações eficientes e seguras com os usuários.

1.5. Segurança na Web: HTTPS, Certificados SSL/TLS

A segurança na web é um tema de extrema importância na atualidade, considerando a crescente quantidade de informações sensíveis compartilhadas online. Neste capítulo, abordaremos os conceitos de segurança relacionados ao protocolo HTTPS e aos certificados SSL/TLS, os quais desempenham um papel essencial na proteção das comunicações e na garantia da privacidade dos usuários na internet. A compreensão desses mecanismos é fundamental para promover uma experiência de navegação segura e confiável.

O HTTPS (Hypertext Transfer Protocol Secure) é uma extensão do protocolo HTTP que adiciona uma camada de segurança à transferência de dados na web. Segundo Resnick(1996), ele utiliza criptografia para proteger as informações transmitidas entre o cliente e o servidor, garantindo a confidencialidade e integridade dos dados durante a comunicação.

Essa camada de segurança é implementada através dos protocolos SSL (Secure Socket Layer) ou TLS (Transport Layer Security), os quais estabelecem uma conexão segura entre o cliente e o servidor antes que os dados sejam trocados. O uso do HTTPS é especialmente relevante em transações que envolvem informações sensíveis, como senhas, dados de pagamento e informações pessoais.

Os certificados SSL/TLS são elementos fundamentais para a segurança do HTTPS. Segundo Fielding et al. (1999), esses certificados são emitidos por Autoridades de Certificação (AC) e contêm informações sobre a identidade do titular do certificado e a chave pública utilizada para a criptografia.

Quando um cliente acessa um site através do HTTPS, o servidor apresenta seu certificado SSL/TLS para autenticação. O cliente verifica a validade do certificado, garantindo que ele foi emitido por uma autoridade confiável e que está associado ao domínio correto. Após a verificação, o cliente e o servidor estabelecem uma conexão segura, utilizando a chave pública do certificado para criptografar os dados durante a comunicação.

A adoção do HTTPS e o uso adequado de certificados SSL/TLS são essenciais para garantir a segurança das comunicações na web. Além de proteger as informações dos usuários contra ataques de interceptação e manipulação de dados, o uso do HTTPS também contribui para a confiança e credibilidade de um site.

Muitos navegadores modernos exibem um ícone de cadeado ou uma indicação visual de "Site Seguro" ao lado da URL do site que utiliza HTTPS, sinalizando aos usuários que suas informações estão sendo protegidas. Essa percepção de segurança é crucial para encorajar os usuários a compartilhar informações pessoais e realizar transações online com tranquilidade.

2. Introdução ao HTML

Para o desenvolvimento web, o HTML (Hypertext Markup Language) é uma ferramenta essencial, pois permite a criação e estruturação de páginas web usando uma linguagem de marcação básica. Este capítulo tem como objetivo fornecer uma visão geral do HTML, incluindo seus conceitos básicos e importância. A familiaridade com esta linguagem é necessária para aqueles interessados em construir páginas web interativas e acessíveis.

2.1. Visão geral do HTML e sua importância no desenvolvimento web

Conforme Berners-Lee, Connolly e Dardailler (1995), o HTML faz uso de elementos ou tags para ordenar e caracterizar o conteúdo da página da web. Essa linguagem de marcação estabelece um sistema para apresentar informações aos usuários de maneira estruturada, por meio do uso de tags baseadas em sintaxe que representam diferentes elementos.

Nas páginas da web, existem marcadores específicos conhecidos como **tags HTML** que são usados para designar o início ("`<tag>`") e o fim ("`</tag>`") de um elemento. Por exemplo, a tag "``" é utilizada para retratar imagens em uma página da Web, assim como a tag "`<p>`" é utilizada para criar parágrafos.

Segundo Duckett (2011), um documento HTML básico compreende elementos cruciais que ditam sua formação. Normalmente, os seguintes componentes abrangem um documento HTML:

Doctype: A declaração "**<!DOCTYPE html>**" especifica a versão do HTML utilizada no documento.

Tag <html>: A tag <html> é o elemento raiz do documento e engloba todo o conteúdo HTML.

Tag <head>: A tag <head> contém metadados do documento, como título, codificação de caracteres e referências a arquivos externos.

Tag <body>: A tag <body> contém o conteúdo visível da página, como texto, imagens, links e outros elementos interativos.

O HTML é a base para a criação de websites, pois ele oferece a possibilidade de criar páginas que podem ser vistas e interpretadas por *browsers* em diferentes dispositivos (Duckett, 2011). É através do HTML que os desenvolvedores definem a estrutura e o conteúdo de uma página da web, o que permite a apresentação de informações de forma organizada e descomplicada.

Podemos desenvolver com HTML páginas web acessíveis utilizando tags semânticas para descrever o conteúdo de forma significativa para leitores de tela e outros dispositivos assistivos (W3C, 2017). Isso assegura que todas as pessoas tenham a capacidade e necessidade de interagir com o digital de forma eficaz.

2.2. Estrutura básica de um documento HTML

A estrutura básica de um documento HTML é o ponto de partida essencial para a criação de páginas web. Neste capítulo, abordaremos os elementos fundamentais que compõem a estrutura de um documento HTML, fornecendo exemplos de código para ilustrar cada parte. A compreensão dessa estrutura é crucial para qualquer desenvolvedor web, pois é a base para a construção de páginas acessíveis, interativas e bem estruturadas

Tag HTML: O Elemento Raiz

Todo documento HTML inicia com a tag <html>, que é o elemento raiz do documento. Essa tag envolve todo o conteúdo HTML da página e define o escopo do documento. É importante lembrar de sempre fechar a tag <html> no final do documento com a tag </html>, conforme demonstrado no exemplo de código abaixo:

```
1  <!DOCTYPE html>
2  <html>
3  |   <!-- Conteúdo da página aqui -->
4  </html>
```

Tag Head: Metadados do Documento

A tag <head> é utilizada para conter metadados e informações importantes sobre o documento. Esses metadados não são visíveis na página, mas desempenham funções cruciais, como definir o título da página, a codificação de caracteres e referências a arquivos externos, como folhas de estilo CSS e scripts JavaScript. A seguir, temos um exemplo de código de uma estrutura básica de documento HTML com a tag <head>:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Título da Página</title>
6  |   <!-- Referências a arquivos externos (opcional) -->
7  </head>
8  <!-- Conteúdo da página aqui -->
9  </html>
```

Tag Body: O Conteúdo Visível da Página

A tag <body> é utilizada para conter todo o conteúdo visível da página, como textos, imagens, links e outros elementos interativos. É dentro da tag <body> que os desenvolvedores estruturam o layout da página e apresentam as informações aos usuários. Veja o exemplo de código a seguir, que inclui a tag <body>:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Título da Página</title>
6      <!-- Referências a arquivos externos (opcional) -->
7  </head>
8  <body>
9      <h1>Exemplo de Título</h1>
10     <p>Este é um parágrafo de exemplo.</p>
11     
12     <a href="pagina.html">Link para outra página</a>
13 </body>
14 </html>
```

2.3. Principais elementos HTML: cabeçalhos, parágrafos, listas, links

Os elementos HTML são os blocos fundamentais de construção para a criação de páginas web. Neste capítulo, abordaremos os principais elementos HTML utilizados para estruturar e formatar o conteúdo das páginas. Exploraremos os elementos de cabeçalhos, parágrafos, listas e links, fornecendo exemplos de código para ilustrar a utilização de cada um deles. A compreensão desses elementos é essencial para a criação de páginas bem organizadas e de fácil leitura para os usuários.

Cabeçalhos: <h1> a <h6>

Os elementos de cabeçalho são utilizados para indicar a hierarquia de títulos em uma página. O HTML possui seis níveis de cabeçalhos, do <h1> ao <h6>, onde o <h1> representa o título mais importante e o <h6> o título de menor importância. Segundo o W3C (2017), os cabeçalhos são úteis para melhorar a acessibilidade e a estrutura do conteúdo da página. Abaixo, temos um exemplo de código com o uso dos elementos de cabeçalhos:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Título da Página</title>
6  </head>
7  <body>
8      <h1>Este é um Título de Nível 1</h1>
9      <h2>Este é um Título de Nível 2</h2>
10     <h3>Este é um Título de Nível 3</h3>
11     <h4>Este é um Título de Nível 4</h4>
12     <h5>Este é um Título de Nível 5</h5>
13     <h6>Este é um Título de Nível 6</h6>
14 </body>
15 </html>
```

Parágrafos: <p>

O elemento <p> é utilizado para criar parágrafos de texto na página. Segundo Duckett (2011), o elemento <p> é uma forma simples e semântica de organizar blocos de texto. O navegador exibe cada parágrafo com espaçamento entre si, proporcionando uma leitura mais clara e organizada para os usuários. Veja o exemplo de código abaixo, onde utilizamos o elemento <p>:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Título da Página</title>
6 </head>
7 <body>
8   <p>Este é um parágrafo de exemplo.</p>
9   <p>Outro parágrafo para ilustrar.</p>
10 </body>
11 </html>
```

**Listas: , e **

As listas são elementos HTML utilizados para organizar itens em formatos de lista. Existem dois tipos principais de listas: as listas não ordenadas e as listas ordenadas . Segundo Berners-Lee et al. (1993), os elementos são utilizados para criar cada item da lista. A seguir, temos um exemplo de código com a utilização de listas:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Título da Página</title>
6  </head>
7  <body>
8      <h2>Lista Não Ordenada:</h2>
9      <ul>
10         <li>Item 1</li>
11         <li>Item 2</li>
12         <li>Item 3</li>
13     </ul>
14
15     <h2>Lista Ordenada:</h2>
16     <ol>
17         <li>Primeiro Item</li>
18         <li>Segundo Item</li>
19         <li>Terceiro Item</li>
20     </ol>
21 </body>
22 </html>

```

Links: <a>

O elemento <a> é utilizado para criar links para outras páginas ou recursos na web. De acordo com Felt et al. (2011), o atributo href dentro da tag <a> é utilizado para definir o endereço do link. Os usuários podem clicar no link para serem direcionados à página ou recurso associado ao link. Veja o exemplo de código abaixo, com a utilização do elemento <a>:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Título da Página</title>
6  </head>
7  <body>
8  |   <p>Visite o site oficial do
9  |       <a href="https://www.exemplo.com">Exemplo.com</a>
10 </p>
11 </body>
12 </html>

```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Título da Página</title>
6  </head>
7  <body>
8  |   <p>Visite o site oficial do
9  |       <a href="https://www.exemplo.com">Exemplo.com</a>
10 </p>
11 </body>
12 </html>

```

2.4. Organização do conteúdo com tags semânticas

A organização adequada do conteúdo em uma página web é essencial para torná-la acessível, bem estruturada e compreensível tanto para os usuários quanto para os mecanismos de busca. Neste capítulo, discutiremos a importância da utilização de tags semânticas em HTML para organizar o conteúdo de forma significativa. Abordaremos as principais tags semânticas disponíveis em HTML5, fornecendo exemplos de código para ilustrar como utilizá-las adequadamente. O entendimento dessas tags é crucial para desenvolver páginas web acessíveis e otimizadas para os motores de busca. As tags semânticas em HTML fornecem um significado e uma estrutura claros ao conteúdo da página, permitindo que os navegadores e outros agentes compreendam a relação entre os elementos e o propósito de cada parte do documento. Conforme W3C (2017), as tags semânticas melhoram a

acessibilidade para usuários com necessidades especiais e facilitam a indexação do conteúdo pelos motores de busca, resultando em uma melhor classificação nos resultados de pesquisa.

Principais Tags Semânticas em HTML5

<header>: A tag <header> é utilizada para definir o cabeçalho da página ou de uma seção específica. Geralmente, é usado para incluir logotipos, títulos e elementos de navegação. Vejamos um exemplo de código:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Título da Página</title>
6  </head>
7  <body>
8      <header>
9          <h1>Nome do Site</h1>
10         <nav>
11             <!-- Links de navegação aqui -->
12         </nav>
13     </header>
14
15     <!-- Conteúdo da página aqui -->
16 </body>
17 </html>
```

<nav>: A tag <nav> é utilizada para definir um bloco de navegação na página, contendo links que permitem ao usuário navegar para outras páginas ou seções do site. Exemplo:

```
1 <nav>
2   <ul>
3     <li><a href="pagina1.html">Página 1</a></li>
4     <li><a href="pagina2.html">Página 2</a></li>
5     <li><a href="pagina3.html">Página 3</a></li>
6   </ul>
7 </nav>
```

<main>: A tag <main> é usada para indicar o conteúdo principal da página. Deve ser utilizada apenas uma vez em todo o documento HTML. Exemplo:

```
1 <main>
2   <h2>Seção Principal</h2>
3   <p>Conteúdo principal da página aqui.</p>
4 </main>
```

<section>: A tag <section> é utilizada para agrupar conteúdo relacionado, geralmente com um título descritivo. Pode ser usada dentro de outras tags semânticas, como <article> ou <main>. Exemplo:

```
1 <section>
2   <h3>Seção 1</h3>
3   <p>Conteúdo da primeira seção.</p>
4 </section>
```

<article>: A tag <article> é usada para marcar um conteúdo independente e auto contido, como uma postagem de blog ou uma notícia. Exemplo:

```
1 <article>
2   <h2>Título do Artigo</h2>
3   <p>Conteúdo do artigo aqui.</p>
4 </article>
```

2.5. Inserção de Imagens e Multimídia

A inserção de imagens e multimídias é uma prática fundamental para enriquecer a experiência do usuário em páginas web. Neste capítulo, discutiremos a importância da utilização de elementos de mídia em HTML, explorando as tags relevantes para inserir imagens, vídeos e áudios em uma página. Forneceremos exemplos de código para ilustrar a correta utilização dessas tags e suas opções de atributos. A compreensão dessa temática é essencial para criar conteúdo envolvente e atrativo em uma variedade de formatos visuais e auditivos.

Inserção de Imagens:

A tag é utilizada para inserir imagens em uma página web. Segundo Duckett (2011), essa tag possui um atributo obrigatório, src, que especifica o caminho ou URL da imagem a ser exibida. Além disso, é possível adicionar atributos opcionais, como alt, para fornecer uma descrição da imagem, e width e height, para definir as dimensões da imagem em pixels. Vejamos um exemplo de código para inserir uma imagem:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Título da Página</title>
6 </head>
7 <body>
8   
9 </body>
10 </html>
```

Inserção de Vídeos: <video>

A tag <video> é utilizada para incorporar vídeos em uma página web. De acordo com W3C (2017), essa tag suporta diferentes formatos de vídeo, permitindo que o navegador escolha o formato adequado com base em sua

compatibilidade. Além do atributo obrigatório `src`, é possível adicionar atributos opcionais, como `controls`, para exibir controles de reprodução, e `width` e `height`, para definir as dimensões do vídeo em pixels. Exemplo:

```
1 <video controls width="640" height="360">
2   <source src="video.mp4" type="video/mp4">
3   <source src="video.webm" type="video/webm">
4   Seu navegador não suporta a reprodução de vídeos.
5 </video>
```

Inserção de Áudios: `<audio>`

A tag `<audio>` é utilizada para inserir arquivos de áudio em uma página web. Assim como a tag `<video>`, a tag `<audio>` suporta diferentes formatos de áudio para maior compatibilidade com os navegadores. É possível adicionar atributos como `src`, `controls`, `autoplay` e `loop`, para controlar o comportamento do áudio. Exemplo:

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  <source src="audio.ogg" type="audio/ogg">
  Seu navegador não suporta a reprodução de áudio.
</audio>
```