

CONTEÚDO AULA 1

Dia 17/09/2019:

APRESENTAÇÃO

1. O que a internet representa para você?

- I: Conectividade
- II: A internet possibilitou conectar diversas pessoas de diversas regiões do globo de uma forma rápida e simples, possibilitando assim acesso fácil e direto a informações em tempo real, melhorando a comunicação entre regiões afastadas, coisas que antigamente era muito difícil.
- III: Novas profissões
- IV: Com o avanço da tecnologia e da internet propriamente dita, vemos um cenário muito promissor, novas profissões foram surgindo gerando mais oportunidades de emprego e de desenvolvimento pessoal.
- V: Chave para o empreendedorismo
- VI: Vender seu produto (Primeiro período)
- VII: Claro não podemos deixar de citar, com a internet a sociedade passou a ter mais acesso a conteúdos de entretenimento de forma dinâmica e personalizada, contribuindo assim para o desenvolvimento no aspecto cultural da sociedade.

2. Como aprender a programa para web?

- I: Inicie...
- II: Principal problema de quem começa a programar: Não programar

Atividade:

✍ Criar um portfólio no github, com todos os seus sistemas que você criou/criará.

✍ Elabore um projeto de uma página que deseja desenvolver no libreoffice writer ou word.

Ex: Catalogo de produtos, currículo, lista de tarefas, lista telefônica, lista de casamento, receitas de bolo, convite, propagandas etc...

1. COMO TUDO COMEÇOU

- I: Antes de falarmos sobre web ou sobre desenvolvimento para web, precisamos falar de internet. E, embora muitos utilizem os dois termos (web e internet) de forma indiscriminada, eles não são sinônimos. Quando falamos de internet, estamos nos referindo a uma grande rede de dispositivos computadorizados de alcance mundial, podemos entendê-la como uma grande infraestrutura em rede. A web (uma derivação abreviada para a expressão World Wide Web, rede mundial de computadores) é apenas uma das funcionalidades da internet – no caso específico, navegar através do hipertexto.
- II: Antes da internet se tornar o que conhecemos hoje, houve um grande percurso na evolução dos computadores e das tecnologias de telecomunicações.
 - i. Durante o período pós-guerra (anos 60 do séculoXX), especialmente na Guerra Fria
- III: Meados de 1962, os Estados Unidos criaram a Cadeia de Comunicação Distribuída (CCD)
- IV: Em 1966, o Departamento de Defesa dos EUA, por meio da ARPANET (Advanced Research Projects Agency – Agência de Projetos e Pesquisas Avançadas), instalou, em 17 locais diferentes, computadores conectados às linhas telefônicas que, a partir de 1969, tornaram-se uma rede de computadores apenas para uso militar.
- V: Nos anos seguintes, algumas agências do governo e universidades subordinadas ao Departamento de Defesa dos EUA começaram a fazer uso restrito da ARPANET com fins de pesquisa.
- VI: A internet, como a conhecemos hoje, é fruto de constantes otimizações e de novas tecnologias que se incorporaram às ideias iniciais da ARPANET. Merece destaque, nesse cenário, o desenvolvimento do protocolo TCP/IP (Transmission Control Protocol/Internet Protocol) adotado pela ARPANET em 1982 e que, posteriormente, foi liberado para utilização civil.

2. INTRODUÇÃO À INTERNET, WORLD WIDE WEB E SOCIEDADE DA INFORMAÇÃO

- I: Como vimos anteriormente, a internet é uma rede de computadores de acesso público e ilimitado (sem um “dono”) que utiliza a infraestrutura de telecomunicações. Embora não exista um dono, existem consórcios internacionais, como o W3C (World Wide Web Consortium), com a tarefa de agregar empresas filiadas na tentativa de, em conjunto, desenvolver padrões para a internet.
- II: O que se pode fazer com a internet?
 - i. Em geral, as aplicações que funcionam sobre a internet obedecem a um mecanismo conhecido como cliente/servidor (client/server), por meio do qual duas aplicações conversam entre si através de um protocolo predefinido.
 - 1. Correio eletrônico
 - 2. Transferência de arquivos
 - 3. Conversa em tempo real
 - 4. Acesso remoto
 - 5. Navegação no hipertexto (Objeto de estudo)

3. PRINCÍPIOS DE FUNCIONAMENTO

O princípio de navegação pelo hipertexto foi pensado, inicialmente, por Tim Berners-Lee, um britânico que, em meados dos anos 90, trabalhava no núcleo de computação do CERN (Organização Europeia de Pesquisa Nuclear). Tim Berners-Lee buscava uma forma de organizar, eletronicamente, os textos e as pesquisas dos cientistas do CERN (e também de outras partes do mundo) de forma que os documentos produzidos pudessem ser interligados e compartilhados.

Partindo-se desse anseio, Tim Berners-Lee desenvolveu um software próprio e um protocolo para recuperar hipertextos que foi denominado de HTTP (Hypertext Transfer Protocol). O formato do texto criado para ser transportado pelo protocolo foi chamado de HTML (Hypertext Markup Language) e consiste de uma linguagem de marcação pela qual é possível, por meio de comandos (tags), incluir ligações entre textos – inclusive entre materiais publicados em diferentes locais. Boa parte do que estudaremos neste material está diretamente relacionada às ideias de Tim Berners-Lee.

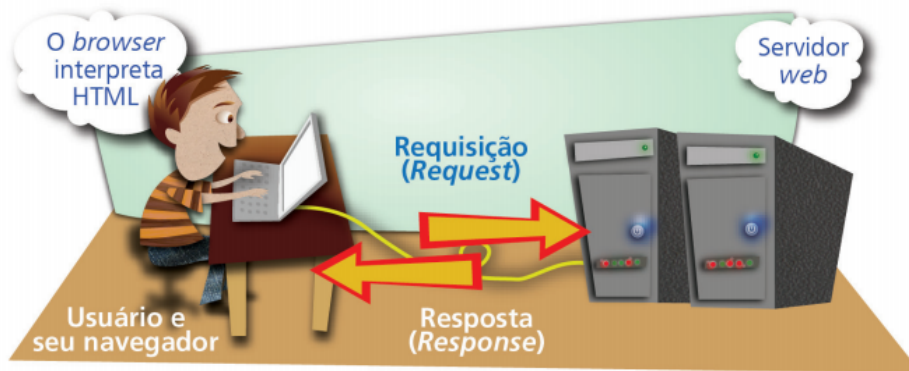
Além de um protocolo (conjunto de regras para que dois dispositivos “conversem”) e de uma linguagem de marcação (para permitir que os usuários se expressem), era necessário um software que, utilizando-se do protocolo desenvolvido, conseguisse obter os documentos escritos em HTML, interpretá-los e exibi-los. Por meio deste software, esperava-se que o usuário “navegasse” pelo hipertexto, ou seja, ao encontrar no texto uma ligação com outro material, com um simples clique, o usuário seria direcionado para uma nova página/conteúdo.

Diante de tal necessidade, Tim Berners-Lee criou um protótipo daquele que viria a ser um dos softwares mais indispensáveis para quem deseja utilizar a internet: o navegador (ou browser). O primeiro navegador foi, inicialmente, nomeado de World Wide Web, mas, posteriormente, para evitar confusão com a expressão “World Wide Web”, foi renomeado para Nexus.

Assim como os principais serviços da internet, a navegação pelo hipertexto obedece aos princípios dos sistemas cliente-servidor. Um documento HTML, ao qual podemos chamar de página, está hospedado em um servidor web (web server) que, nesse caso, desempenha o papel de aplicação servidora. Um servidor web recebe requisições de documentos por meio URLs (O Uniform Resource Locator, ou “localizador uniforme de recursos”) e as entrega por meio do protocolo HTTP. Pelo lado cliente, temos o navegador ou browser cuja função principal é submeter uma requisição no formato de uma URL e, após receber o resultado, na forma de um documento HTML, interpretá-lo e apresentá-lo ao utilizador.

Observe que, quando o navegador recebe a resposta do servidor web, o documento HTML retornado não é apresentado tal como foi recebido. O documento passa por um processo de interpretação e o resultado de tal etapa é o que é mostrado ao usuário. Atualmente, não é comum que uma mesma página seja exibida de forma diferente em navegadores concorrentes. Por isso, é muito importante que você, que está começando a desenvolver páginas e aplicações para a internet, preocupe-se sempre em seguir padrões estabelecidos, minimizando esse tipo de problema.

Além de disponibilizar conteúdo através de páginas interligadas, os conceitos difundidos pelas ideias de Tim Berners-Lee evoluíram. Hoje, em muitas situações, o conteúdo a ser apresentado é gerado dinamicamente. Isso é possível através do desenvolvimento de aplicações web, ou seja, programas de computador executados no servidor web e produtores de conteúdo que é enviado para interpretação e apresentação pelo navegador. Esse tipo de aplicação, normalmente, faz uso de servidores de banco de dados e de outros recursos comuns em sistemas de informação. A figura abaixo ilustra os principais elementos presentes em um cenário de navegação pela internet.



I: Definição de Hipertexto, Multimídia e Hipermissão

A ideia de hipertexto surgiu no campo das artes e da literatura, e diz respeito a um tipo de texto ou a um tipo de mensagem, que não existe plenamente até que um leitor interaja com suas partes.

O hipertexto é um documento digital composto por diferentes blocos de informações interconectadas. Essas informações são amarradas por meio de elos associativos, os links. Os links permitem que o usuário avance em sua leitura na ordem que desejar. (LEÃO, 1999)

Ou seja, embora os elementos estejam todos ali, a mensagem só é criada e só passa a fazer sentido no momento em que um leitor escolhe a maneira como irá arranjá-la.

Com o avanço tecnológico, a Web começou a permitir também a incorporação de outros tipos de mensagens e códigos, não ficando restrita apenas a textos. A convergência digital permitiu o surgimento da chamada multimídia, em que vários tipos de linguagem podem ser reproduzidas no padrão digital.

(...) multimídia (...), ou seja, a incorporação de informações diversas como som, textos, imagens, vídeo, etc., em uma mesma tecnologia – o computador. (IDEM)

Foi então necessário reformular a terminologia utilizada na Web, final, chamar de hipertexto algo que envolve toda a variedade de linguagens parecia limitador.

Então, o tipo de arquitetura em que a Web se fundamente, passou a ser chamada de hipermissão, que nada mais é do que a junção dos termos hipertexto e multimídia.

A hipermissão, por sua vez, é uma tecnologia que engloba recursos do hipertexto e multimídia, permitindo ao usuário navegar por diversas partes do aplicativo, na ordem que desejar. (IDEM)

4. INTRODUÇÃO ÀS TECNOLOGIAS E ABORDAGENS DE PROGRAMAÇÃO WEB;

I: Principais linguagens

- i. **HTML:** Não é uma linguagem de programação, é uma linguagem de marcação. Bem resumidamente, é um conjunto de regras e códigos que define como os elementos da página são exibidos. Conhecimento em HTML é o básico para qualquer programador WEB.
- ii. **CSS (Cascading Style Sheet):** Em português, significa Folha de Estilo em Cascatas. O CSS é uma linguagem de programação usada para estilizar elementos de uma página desenvolvida com uma linguagem de marcação, como o HTML;

- iii. **JavaScript:** É uma linguagem de programação/scripts que são executados no navegador do usuário de um website (sítio eletrônico ou endereço eletrônico);
- iv. **PHP:** É uma linguagem de script em que o código é normalmente embutido no próprio HTML. No caso do PHP, o código é executado do lado do servidor e é enviado para o cliente apenas o resultando em HTML puro;

II: Outras linguagens: .NET, ASP, Java, Ruby, Python, Perl, C, C++.

III: Tecnologias e abordagens de programação Web

IV: Perfis de desenvolvedores WEB:

- i. Front-end
- ii. Back-end
- iii. DevOps

V: Conhecimentos necessários para todos os perfis de desenvolvedor web:

- i. GIT – Controle de versionamento;
- ii. Básico de uso de terminal;
- iii. Estrutura e algoritmos de dados;
- iv. GitHub;
- v. Licenças;
- vi. SSH;
- vii. HTTP/HTTPS e APIs;
- viii. Padrão de projetos;
- ix. Codificações.

RESUMO

Nesta aula, estudamos um pouco da história e do surgimento da internet. Aprendemos a definição de hipertexto, multimídia e hiperarquia. E de forma sucinta, tratamos das tecnologias e abordagens de Programação Web.

ATIVIDADES DE APRENDIZAGEM

1. Qual a principal motivação dos cientistas para desenvolver a internet?
2. O que você entende por URL?
3. Muito do que se desenvolve para a internet é baseado no modelo cliente/servidor. Explique com suas palavras como esse tipo de aplicação funciona.

Referências bibliográficas:

<https://infonauta.com.br/producao-multimidia/199/hipertexto-hiperarquia-multimidia/>

http://www.abepro.org.br/biblioteca/ENEGEP2000_E0111.PDF

<https://central3.to.gov.br/arquivo/453374/>

CONTEÚDO AULA 2

Dia 24/09/2019

3. ETAPAS DE CONSTRUÇÃO DE UM WEB SITE.

I: **Levantamento dos requisitos/Briefing:** O desenvolvimento de websites começa com a troca de informações entre o desenvolvedor de website e o cliente. É essencial que o responsável pelo website fique ciente da história da empresa, referências e objetivos da empresa.

☉ **Briefing:** É um documento muito importante e que servirá como orientação para os profissionais que vão criar o web site.

☛ Nele, estarão alocadas as seguintes informações:

- ☛ Detalhes do produto ou serviço;
- ☛ O panorama atual em que o mercado está inserido;
- ☛ O perfil do consumidor final;
- ☛ Objetivos da empresa e do cliente.

Exemplo: Venda de empadas Tempero & Arte Cidade de Goiás

O que o cliente quer: E-commerce/Vendas

Qual o conteúdo: Catálogo de pratos

Qual o contexto: Aumentar o faturamento através de um novo canal de vendas através da internet

Usuário: Cidadão Vilaboense/Turista

Identidade visual para seguir: Logo e cores padrão da empresa

Seções do site: Vendas, Contrato/Quem somos

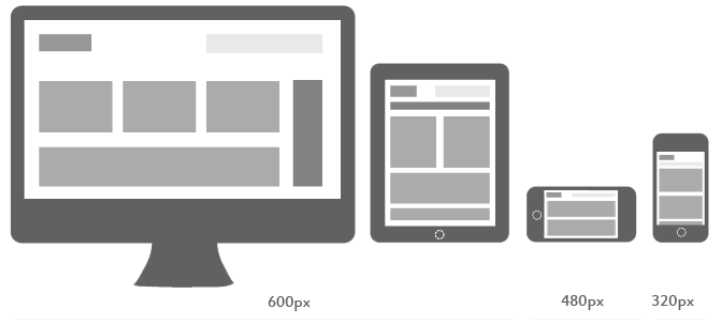
Responsável pelo conteúdo: Cliente

☉ **Contrato:** Elabore o contrato de prestação de serviço.

II: **Design e Layout:** Essa etapa normalmente parece simples para muitos... mas não é! O design por trás da criação de um site é complexo e trabalhoso. É preciso escolher um template inicial – que o designer usará de base para transformar em um site personalizado – o esquema de cores, layout das informações (como por como, por exemplo, quais serão os tópicos na barra ‘menu’ ou onde e como ficará as informações para contato) e ainda o tratamento de todas as imagens que serão usadas para compor o website. É um trabalho que exige bom gosto, talento, agilidade e muita dedicação.

III: **Codificação/Programação:** Programador codifica as funcionalidades já estabelecidas anteriormente. Através do uso de linguagens HTML / CSS é feita a codificação do layout criado.

IV: **Testes e Aprovação:** Depois de montar uma primeira versão do site, ele precisa ser testado para ter certeza que o funcionamento está correto, o responsivo está configurado (funcionalidade que permite adaptação de telas) e, principalmente, que o cliente está satisfeito. Nesse momento que serão apontados os detalhes e pequenas mudanças que o cliente gostaria de fazer para que tudo fique do seu jeito.



V: **Publicação:** Conteúdo, layout, design e programação criado e aprovado.

VI: **Manutenção:** A fim de garantir que todos os recursos estejam funcionando corretamente, que a cara do site esteja sempre atual e que novos visitantes se interessem pelo seu negócio, é necessário realizar a manutenção do site.

4. REQUISITOS DE QUALIDADE DE APLICAÇÕES WEB;

- I: Usabilidade;
- II: Funcionalidade;
- III: Confiabilidade;
- IV: Eficiência;
- V: Portabilidade;
- VI: Manutenibilidade.

5. INTRODUÇÃO A PADRÕES DE PROJETO PARA APLICAÇÕES WEB;

- I: Estado da arte
- II: Principais padrões de projeto e cenários de aplicação

RESUMO

Nesta aula, estudamos um pouco, quais as etapas de construção de aplicações web, requisitos de qualidade e uma breve introdução sobre padrões de interfaces.

ATIVIDADES DE APRENDIZAGEM

Referencias bibliográficas:

<https://setup.technology/etapas-de-desenvolvimento-de-websites/>

<https://sinnapse.com/5-importantes-etapas-para-criacao-de-um-site/>

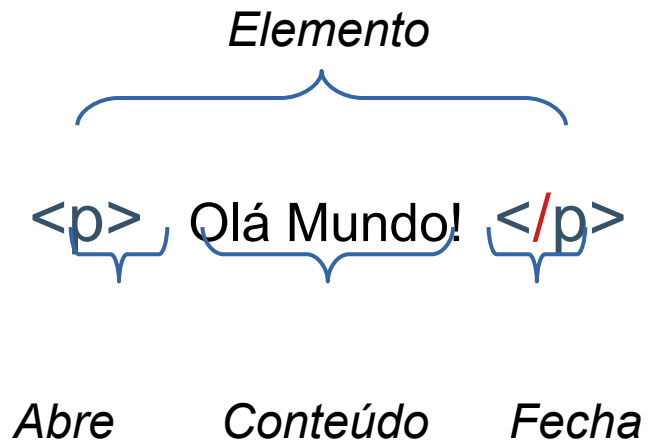
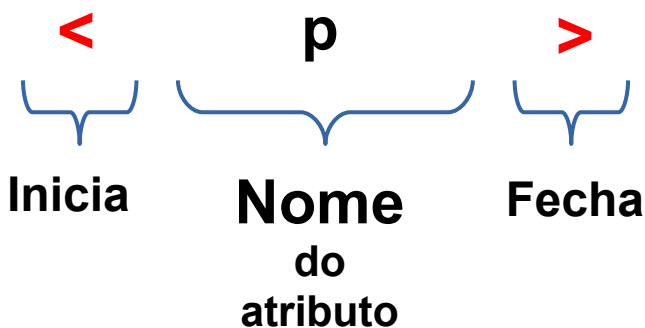
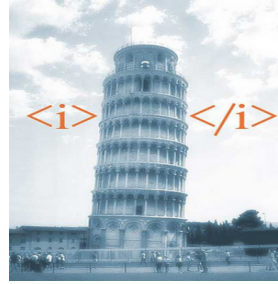
<https://rgarte.com.br/blog/etapas-do-desenvolvimento-de-um-site/>

<https://www.youtube.com/watch?v=8HrUVANj0wE>

Tag HTML

Importante

Objetivo principal hoje é que você saiba reconhecer um código HTML quando ele estiver na sua frente!



<html> - marca do início do documento HTML

<head> - início do cabeçalho

<title> - título da página

<body> - corpo do documento

**
** Pula linha

<p> - Parágrafo

**** **Texto em negrito;**

<i> *Texto em itálico;*

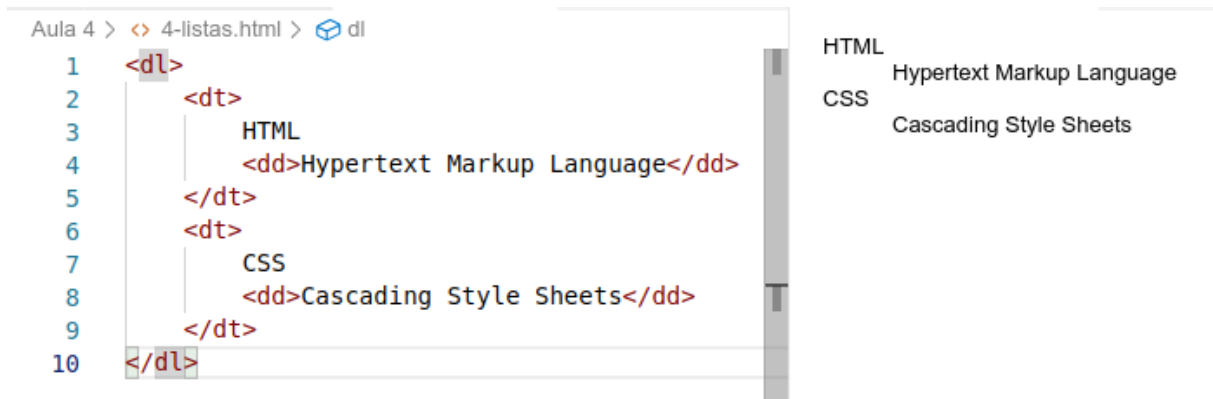
<u> Texto sublinhado;

**** Tamanho do texto;

**** - **Cor do texto;**

A tag **** é uma tag normal, pois necessita que tenha início e fim envolvendo um texto, mas podemos notar que ela recebe alguns atributos como “size” e “color”, entre outros.

LISTA

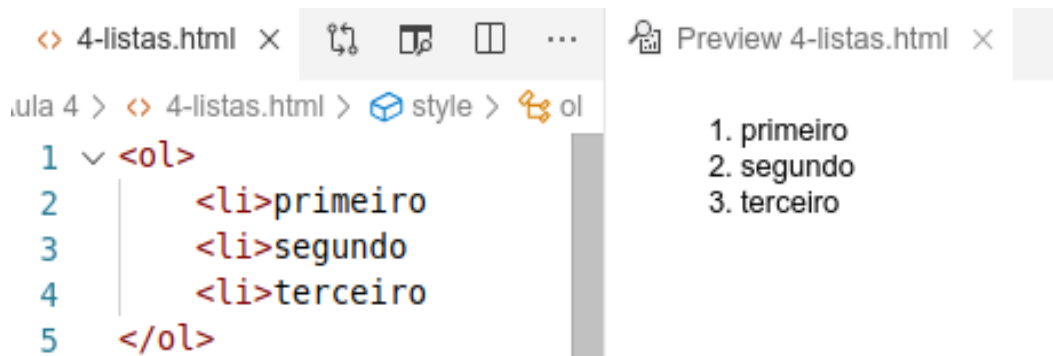


```
Aula 4 > <> 4-listas.html > dl
1 <dl>
2   <dt>
3     HTML
4     <dd>Hypertext Markup Language</dd>
5   </dt>
6   <dt>
7     CSS
8     <dd>Cascading Style Sheets</dd>
9   </dt>
10 </dl>
```

HTML
Hypertext Markup Language

CSS
Cascading Style Sheets

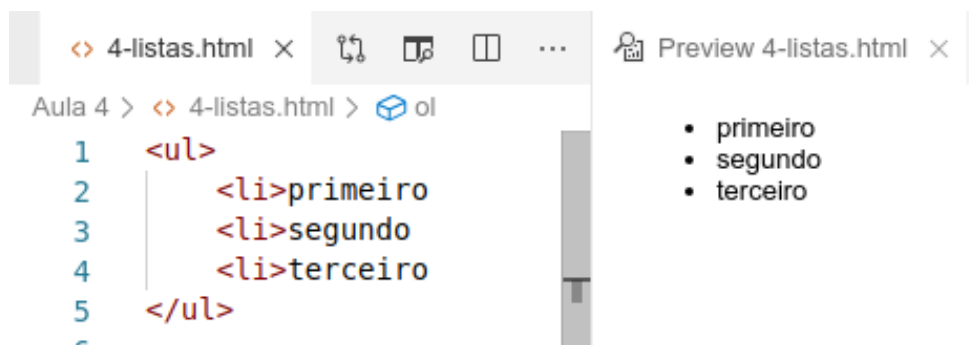
Lista de **definição**.



```
<> 4-listas.html x [Icons] ... Preview 4-listas.html x
Aula 4 > <> 4-listas.html > style > ol
1 <ol>
2   <li>primeiro
3   <li>segundo
4   <li>terceiro
5 </ol>
```

1. primeiro
2. segundo
3. terceiro

Lista **ordenada**.



```
<> 4-listas.html x [Icons] ... Preview 4-listas.html x
Aula 4 > <> 4-listas.html > ol
1 <ul>
2   <li>primeiro
3   <li>segundo
4   <li>terceiro
5 </ul>
```

• primeiro
• segundo
• terceiro

Lista não **ordenada**.

 inclui uma imagem em uma página HTML.

Sintaxe para :

```
<img href="http://google.com/search?
      q=imagem+monalisa.png">
```

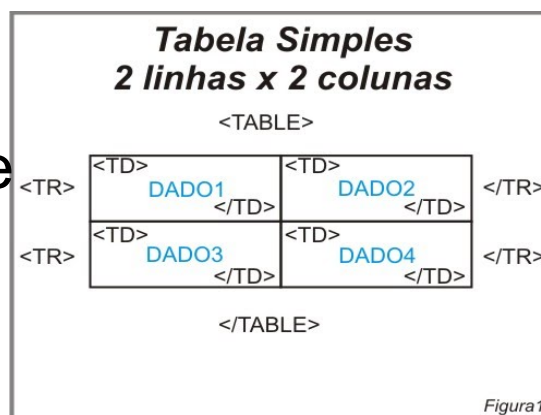


TABELAS

<table> - Cria uma tabela;

<tr> - Cria uma linha de uma tabela;

<td> - Cria uma célula numa linha.



```
<> 3-tabelas.html x  <> 4  Preview 3-tabelas.html x
Aula 4 > <> 3-tabelas.html > table
1  <table border='1px'>
2    <tr>
3      <td>Dado 1</td>
4      <td>Dado 2</td>
5    </tr>
6    <tr>
7      <td>Dado 3</td>
8      <td>Dado 4</td>
9    </tr>
10  </table>
```

Dado 1	Dado 2
Dado 3	Dado 4

Elementos de formulários

Elementos de formulários são elementos que permitem ao usuário entrar com informações em um formulário como:

text fields;

password;

drop-down menus;

radio buttons etc.



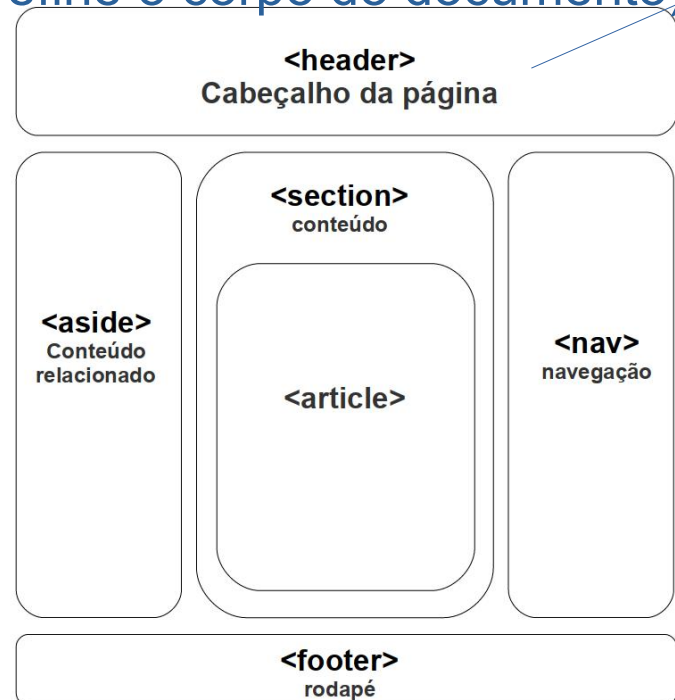
The screenshot shows a web page for 'MEDITEC' with a dark blue header containing the logo and links for 'Ajuda' and 'Acessar'. Below the header is a blue section with the title 'Login' and a link 'Ainda não tem uma conta? clique aqui e cadastre-se'. The main content area is white and contains two text input fields. The first field is labeled 'Email ou CPF' in red text and has a red underline. The second field is labeled 'Senha' in gray text. Below the password field is a 'Entrar' button.

```
<> 1-tags.html <> 5-form.html X <> 4-listas.html ... Preview 5-form.html X
Aula 4 > <> 5-form.html > form
1 <form action="" method='post'>
2
3 <label>E-mail ou CPF</label><br>
4 <input type="text"></p>
5
6 <label>Senha</label><br>
7 <input type="password"></p>
8
9 <button type="submit">Entrar</button>
10 </form>
```


COMPONENTES

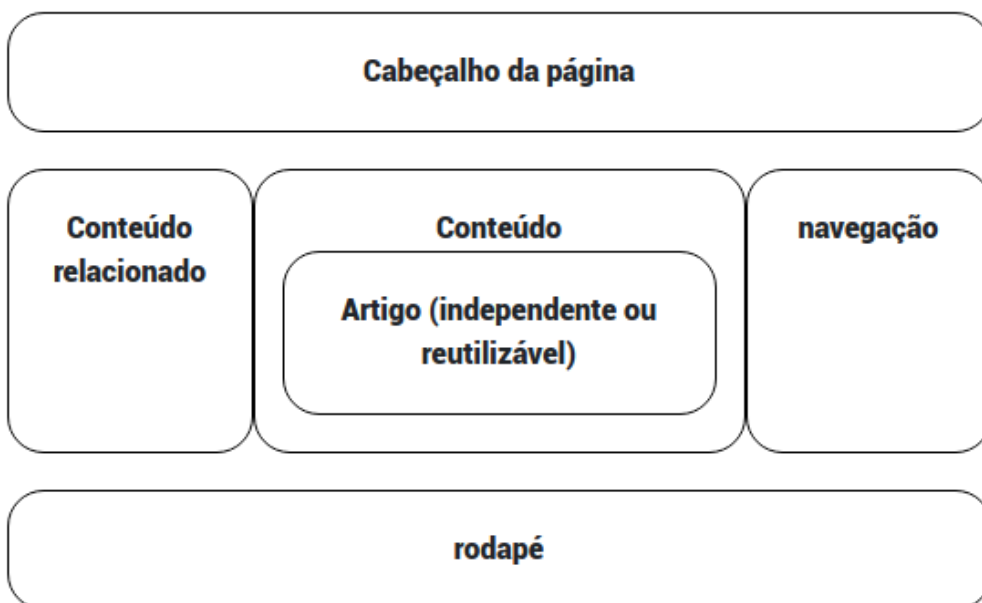
<body>

(Define o corpo do documento)



```
<body>  
  <header>  
    Cabeçalho da página  
  </header>  
  
  <aside>  
    Conteúdo relacionado  
  </aside>  
  
  <section>  
    Conteúdo  
  
    <article>  
      Artigo  
      (independente ou reutilizável)  
    </article>  
  
  </section>  
  
  <nav>  
    navegação  
  </nav>  
  
  <footer>  
    rodapé  
  </footer>  
</body>
```

Resultado



ETIQUETAS DOCUMENTOS HTML

<!DOCTYPE> Documento HTML5

<html> documento html

<head> informação sobre o documento

<body> corpo da página

ETIQUETAS <HEAD>

<link> Link para uma folha de estilos externa
rel="stylesheet"
type="text/css"
href="theme.css"
media="print" especifica para qual mídia

<meta> Define informações metadado e auxiliam os buscadores
charset = "UTF-8"

<script> define um script
type="text/javascript"

<style> Define um estilo
media="print" especifica para qual mídia
title="Meu titulo"
type="text/css" especificar o tipo de mídia

<title> título do documento

ELEMENTOS DEFINIÇÃO

<a> hyperlink

**** imagem

<map> uma imagem de mapa

<menu> lista de "menus"

<p> parágrafo

<header> cabeçalho

ATRIBUTOS GLOBAIS

Id="" identificador (único)
class="" classe (múltiplas)
lang="" idioma
translate= [yes],[no]
title="" título
data-*="" metadados personalizados
style="" estilo CSS
dir="" ltr, rtl, auto direção do texto

ALTERAÇÃO DE TEXTO

<h1> H1 até H6: Título

**** texto em ênfase

<i> texto em itálico

<pre> texto pré-formatado

<q> citação curta

<small> pequeno texto

**** texto forte (similar ao negrito)

<sub> texto subscrito

<sup> texto sobrescrito

**** texto em negrito

DEFINIÇÕES

<address> endereço.

**
** quebra de linha simples

<cite> citação

<code> código texto do computador

<dd> descrição de definição

<div> seção no documento

<dl> lista de definição

<dt> termo de definição

**** itens da lista

**** lista ordenada

**** seção no documento

**** lista desordenada

<hr> regra horizontal.

TEBELAS

<col> atributos da coluna da tabela

<table> Define uma tabela

<tbody> Define o corpo da tabela

<thead> Define o cabeçalho da tabela

<tfoot> Define o rodapé da tabela

<td> Define uma célula da tabela

<th> Define o cabeçalho da tabela

<tr> Define uma linha da tabela

FORMULÁRIO

<form> Define um formulário

name: nome do formulário

method: get ou post

action: URL

autocomplete: [on][off]

novalidate: não validação

target: _self, _blank ou _parente

enctype: multipart/form-data, txt/plain

<button> botão de comando

name: nome do campo

type: submit, reset ou button

value: valor inicial

<fieldset> conjunto de campos (fieldset)

<input> campo de inserção

name: nome do campo

type: text, hidden, search, tel, url, e-mail, number, range, color, password, file, datetime, date, time, week, month, checkbox, radio, submit, image, reset, button

value: valor inicial

size: tamanho do campo

list: associa id de <datalist>

placeholder: sugestão

accept: formtatos suportados

alt: texto alternativo

src:URL

width height: largura/altura

autocomplete: [on][off]

checked: campo selecionado

multiple: permite vários valores

<label> "label" define um rótulo para um elemento
for: Especifica elemento que está vinculado.

<optgroup> grupo de opção

<option> opção em uma lista suspensa

value: nome do campo

label: texto visível

selected: opção selecionada

<select> lista selecionável

name: nome do campo

size: tamanho do campo

multiple: permite vários valores

<textarea> área de texto

name: nome do campo

ATRIBUTOS FORM

spellcheck revisão ortográfica

autofocus coloca focus no campo

hidden oculta o campo

contenteditable permite editar o campo

VALIDAÇÕES FORM

Campos numéricos: **min, max, step**

Campos textuais: **minlength, maxlength, pattern** (REGEX)

Geral: **required, disabled, readonly**

<!-- COMENTÁRIOS -->

Obrigado!