



PET.COMP

FUNDAMENTOS DE DESENVOLVIMENTO WEB

HTML



CSS



Tópicos

- Metodologia
- Formulário
- Conceitos Fundamentais do Desenvolvimento WEB
- Ferramentas Utilizadas
- HTML
- CSS
- Bootstrap
- Estudo de Caso

Metodologia

Esse minicurso será ministrado com base em uma abordagem integrada que combina diferentes metodologias de ensino. Inicialmente, será utilizada a **aula expositiva**, em que o conteúdo será apresentado de forma direta, com o apoio de slides e materiais impressos, para introduzir os principais conceitos. Em seguida, o **método socrático** será aplicado, incentivando a participação ativa dos alunos por meio de perguntas que estimulem o raciocínio crítico e a reflexão sobre o tema. Por fim, a **aprendizagem baseada em projetos (ABP)** será a base para a prática, onde os alunos irão desenvolver um projeto real, aplicando os conhecimentos adquiridos e solucionando problemas, promovendo a integração entre teoria e prática.

Formulário

Ao fim deste minicurso será disponibilizado um formulário para avaliar o:

- conhecimento adquirido;
- metodologia de ensino;
- qualidade do ensino.

O formulário possui a finalidade de aquisição de dados que serão de fundamental importância para a constante melhora dos minicursos ministrados pelos alunos do PET.COMP, bem como base de dados para produção de artigos acadêmicos. Sempre respeitando a Lei Geral de Proteção de Dados (LGPD).

Conceitos Fundamentais do Desenvolvimento WEB

Definição e Importância

- Desenvolvimento web refere-se ao processo de criação e manutenção de websites e aplicações web;
- Back-end (infraestrutura e lógica) e Front-end (interface de usuário);
- Essencial para a comunicação global, influenciando em diversos campos;
- Inovação constante por meio da utilização de tecnologias como IA e big data para criar serviços novos e mais eficientes;
- Auxilia no acesso à educação por meio de plataformas online, bem como na economia.

Estrutura de um WebSite

- Um website combina front-end, back-end, hospedagem e domínio para oferecer uma experiência funcional ao usuário;
- Front-end: Utiliza HTML para estrutura, CSS para estilização de página e JavaScript para interatividade;
- Back-end: Envolve linguagens de programação (Python, PHP, etc.), servidores (Apache, Nginx) e bancos de dados (MySQL, MongoDB);
- Hospedagem: Armazena arquivos acessíveis online;
- Domínio: É o endereço que facilita o acesso ao site, como www.exemplo.com.

Protocolos e Padrões da Web

- Os protocolos e padrões da web são acordos técnicos que permitem a interoperabilidade e a comunicação eficaz entre diferentes sistemas de computação na internet;
- O protocolo HTTP foi criado para garantir a interoperabilidade, e após sua criação surgiu o HTTPS, para adicionar uma camada de segurança ao mesmo.

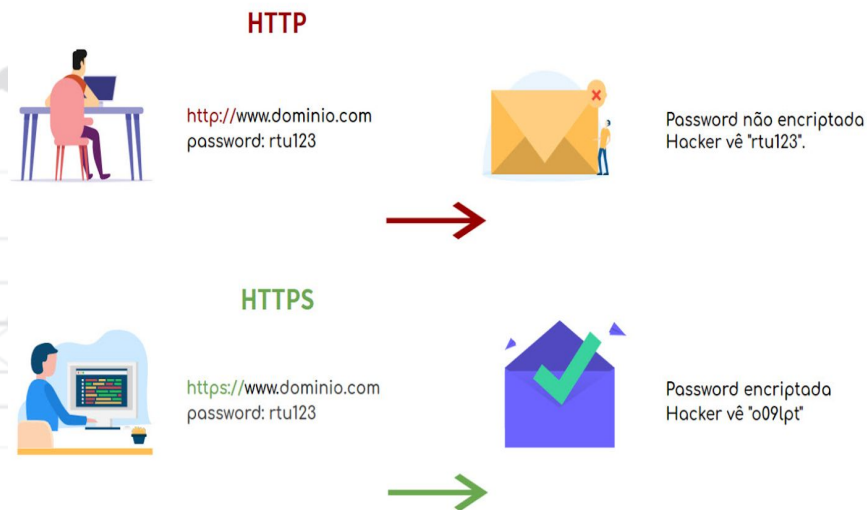
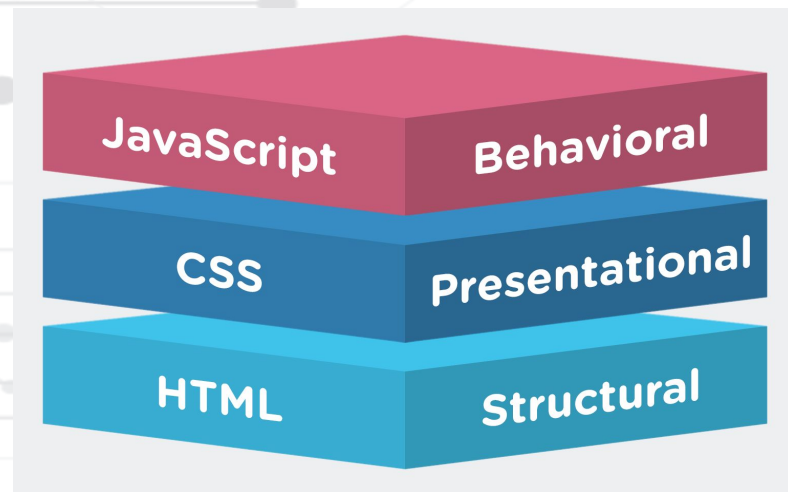


Figura 1: Funcionamento da Criptografia de senhas

Protocolos e Padrões da Web

- Organizações como o W3C publicam e mantêm padrões para tecnologias como HTML e CSS, enquanto o IETF foca em protocolos de internet, assegurando uma base sólida para o desenvolvimento web.
- As diretrizes WCAG garantem acessibilidade para pessoas com deficiências, promovendo inclusão digital.



Ferramentas Utilizadas

VS CODE

O Visual Studio Code (VSCODE) é um editor de código leve, porém poderoso, que suporta uma vasta gama de linguagens de programação e tecnologias web. Ele oferece funcionalidades como destaque de sintaxe, autocompletar de código, gestão de versões integrada e extensões personalizáveis. Isso tudo ocorre através dos *pluguins* (módulo de extensão).



Plugins

O live server é uma extensão que inicia um servidor local de desenvolvimento que atualiza automaticamente as modificações dos arquivos do projeto.

Para o estudo de caso no final desta aula vamos utilizá-lo, por isso, vamos instalar!

Live Server



HTML

Estrutura Básica do HTML

HTML (HyperText Markup Language) é a linguagem fundamental utilizada para criar e estruturar páginas na web. Ela permite que desenvolvedores definam a organização e o layout do conteúdo, incluindo texto, imagens, links e outros elementos multimídia. Utilizando uma série de tags, como `<html>`, `<head>`, `<body>`, e `<p>`, HTML forma a base de qualquer página web, estruturando o conteúdo de maneira semântica e hierárquica. Essa linguagem é essencial para a construção de sites, pois permite que os navegadores interpretem e exibam o conteúdo de forma adequada aos usuários, criando a experiência de navegação que conhecemos hoje.

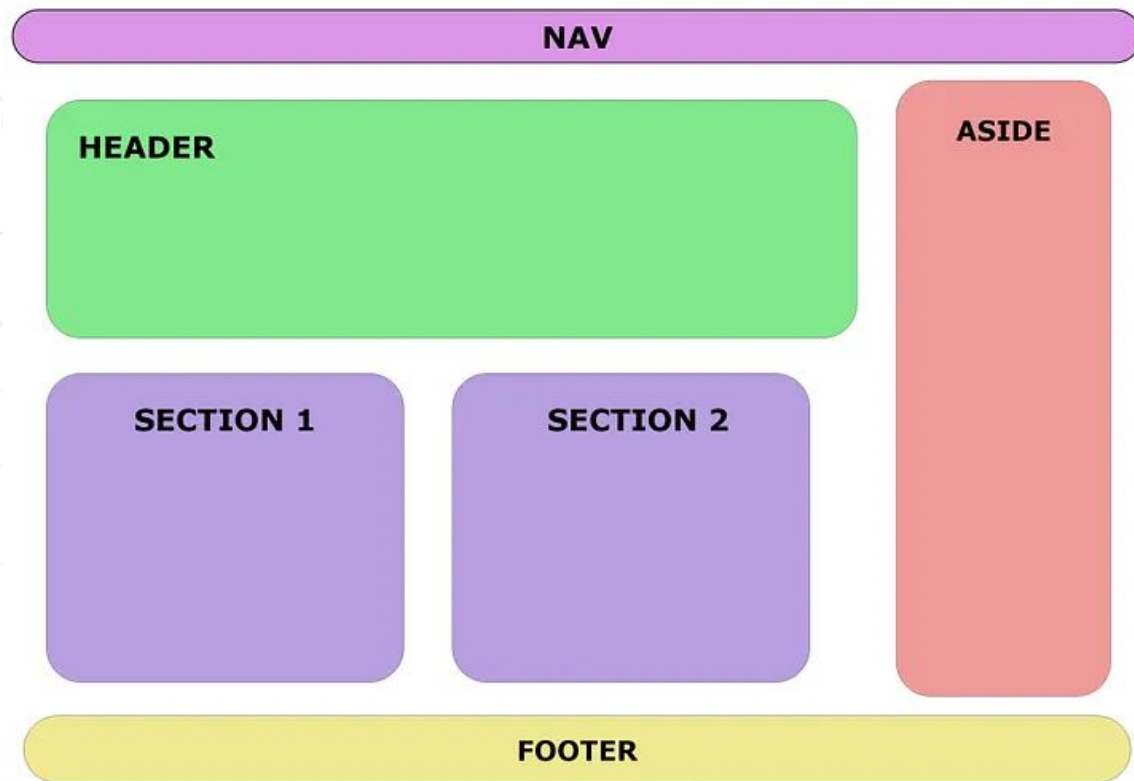
Estrutura Básica do HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!-- Seu código aqui -->
</body>
</html>
```

Tags Semânticas

Tags semânticas são tags que possuem um significado, elas dão sentido a informação de texto ao navegador e buscadores. Por exemplo: para definir um painel de navegação, cabeçalhos, conteúdos laterais, seções de conteúdos, entre outros.. É uma boa prática sempre utilizar essas tags semânticas com o intuito de melhorar o entendimento do código. Ademais, esse tipo de prática contribui muito no SEO do site (Otimização para motores de busca, responsável por ranquear melhor sua aplicação web nos motores de buscas como o Google).

Tags Semânticas



Tags sem semântica e estilos de formatação

- **Divisão:** a fim de separar os diversos conteúdos de uma página de forma genérica e com uma estilização específica usa-se a tag **<div>**.
- **Quebra de texto:** quando se tem o interesse em realizar uma quebra de linha obrigatória ou gerar um espaçamento entre parágrafos utiliza-se a tag **
**.
- **Marcações textuais:** existem diversas tags que são utilizadas para estilizar um parágrafo **<p>**, ou parte dele, de forma simples sem utilizar CSS.

<i>: Texto em itálico

: Texto em negrito

: Texto importante

: Texto enfatizado

<mark>: Texto marcado

<small>: Texto menor

: Texto deletado

<ins>: Texto inserido

<sub>: texto subscrito

<sup>: texto sobrescrito

Imagens

Para inserir imagens em uma página web, você usa a tag ****. Esta tag é autocontida e possui vários atributos que você pode definir. Os dois atributos mais importantes são **src** e **alt**.

```

```

Tags de Lista

Lista Não Ordenada

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Lista Ordenada

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Lista de Definições

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Hiperlink

A navegação da página se dá pelo uso da tag `<a>`, sendo ela responsável por gerar URLs clicáveis. Existem dois atributos importantes: **href** e **target**. As ações respectivas proporcionadas por esses atributos são: direcionar o navegador do usuário até a url desejada e especificar onde esse novo direcionamento será aberto, como por exemplo, possibilitando abrir essa url em nova aba (**_blank**) no navegador.

```
<a href="Endereço">Titulo</a>
```

```
<a href="https://www.petcomp.cefetmg.br/" target="_blank">PET.COMP</a>
```

CSS

Estrutura Básica do CSS

CSS (en. Cascading Style Sheets; pt.br. Folhas de Estilo em Cascata) é uma linguagem de estilo utilizada para descrever a apresentação de documentos **HTML**. Ele permite que desenvolvedores web separem o conteúdo da estrutura visual, oferecendo maior flexibilidade e controle sobre o layout e o design das páginas. Desta forma é possível definir estilos para textos, ajustar espaçamentos, posicionar elementos e criar animações, tornando a experiência do usuário mais agradável e interativa.

A estrutura do **CSS** consiste em regras que são aplicadas aos elementos **HTML**. Cada regra é composta por um seletor e um bloco de declarações. As declarações, por sua vez, são compostas por propriedades **CSS** e seus respectivos valores. Além disso, é possível agrupar múltiplos seletores em uma única regra, separando-os por vírgulas, para aplicar o mesmo conjunto de estilos a diferentes elementos.

Sintaxe Básica e Estrutura

A sintaxe básica do **CSS** é composta por seletores e declarações. Um seletor indica qual elemento **HTML** será estilizado, enquanto a declaração é composta por uma propriedade e um valor;



```
p {  
    color: blue;  
    font-size: 16px;  
}
```

Seletores

Os seletores são uma parte fundamental do **CSS**, permitindo que os desenvolvedores escolham quais elementos **HTML** estilizar. Existem vários tipos de seletores, incluindo seletores de **tipo**, **classe**, **ID**, entre outros. Cada tipo de seletor tem uma sintaxe específica e serve para diferentes propósitos, proporcionando uma forma eficiente de aplicar estilos a elementos específicos ou a grupos de elementos.

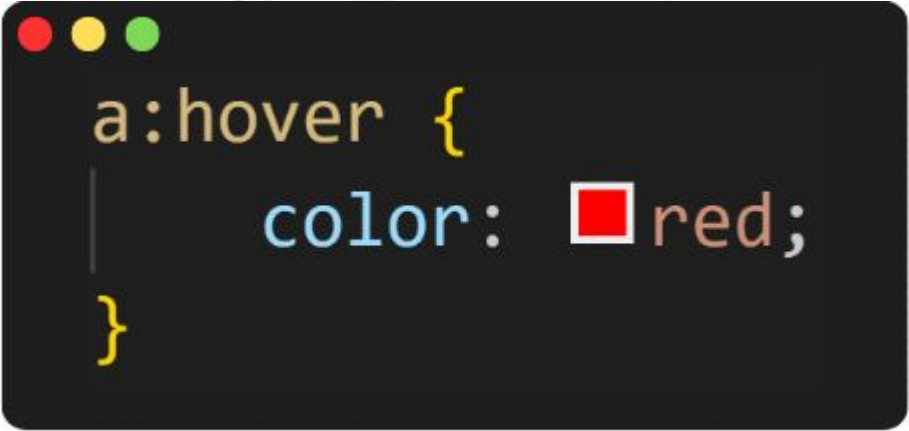
```
p {  
  color: blue;  
  font-size: 16px;  
}
```

```
#id {  
  background-color: black;  
  border-radius: 5px;  
}
```

```
.class{  
  text-align: center;  
  justify-content: center;  
}
```

Pseudo-Classes

As pseudo-classes são usadas para definir um estado especial de um elemento. Elas permitem estilizar elementos com base em informações dinâmicas, como o estado do link (:hover, :visited), a posição do elemento (:first-child, :last-child), entre outros.



```
a:hover {  
  color:  red;  
}
```

Prioridades

A prioridade determina qual regra será aplicada quando várias regras conflitantes se aplicam ao mesmo elemento. A especificidade é calculada com base na **origem do estilo** (en. inline; pt.br. em linha, interno ou externo), **tipo de seletor** (ID, classe, tipo), e a **ordem das regras no CSS**. Regras inline têm a maior prioridade, seguidas por seletores de ID, classes, e finalmente, seletores de tipo.

```
<div class="div content">
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</div>
```

```
.div.content ul li {
  color: blue !important;
}
```

Box Model

O modelo de caixa (Box Model) é um conceito fundamental no **CSS** que descreve como os elementos são representados na página. Cada elemento é considerado uma caixa retangular composta por quatro partes: **conteúdo**, **padding** (preenchimento), **borda** e **margem**. Compreender o Box Model é essencial para controlar o layout e o espaçamento dos elementos na página.



Tipos de Unidades

Existem várias unidades de medida para definir tamanhos e espaçamentos, incluindo unidades **absolutas** (pixels, centímetros) e **relativas** (em, rem, porcentagem). Unidades absolutas são fixas e não mudam com o contexto, enquanto unidades relativas são baseadas em outras medidas, permitindo designs mais flexíveis e responsivos.

```
div {  
  width: 25%;  
  height: 100px;  
}
```

Posições estáticas, absolutas e relativas

Permite controlar o posicionamento dos elementos na página através das propriedades de posição. A posição **estática** é o valor padrão, onde os elementos são posicionados conforme aparecem no fluxo do documento. A posição **relativa** permite deslocar um elemento em relação à sua posição original. A posição **absoluta** remove o elemento do fluxo do documento e o posiciona em relação ao seu elemento pai mais próximo com posição relativa ou absoluta.

Posições estáticas, absolutas e relativas

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

```
div.relative {  
  position: relative;  
  left: 30px;  
  border: 3px solid #73AD21;  
}
```

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

Posições estáticas, absolutas e relativas

```
div.sticky {  
  position: sticky;  
  top: 0;  
  background-color:  green;  
  border: 2px solid  #4CAF50;  
}
```

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid  #73AD21;  
}
```

Importações

A regra **@import** permite importar folhas de estilo externas dentro de uma folha de estilo **CSS**. Isso é útil para organizar e modularizar os diversos arquivos **CSS**, especialmente em projetos grandes. A sintaxe básica é:

```
@import url('styles.css');
```

Essa regra deve ser colocada no início da folha de estilo, antes de quaisquer outras regras **CSS**.

Bootstrap

Introdução

É um framework de front-end amplamente utilizado para o desenvolvimento de sites e aplicações web responsivas e móveis. Criado pelo Twitter, ele fornece uma coleção de ferramentas e componentes prontos para usar, como grids, botões, formulários, e modais, facilitando o processo de design e garantindo consistência visual. Utilizando **HTML**, **CSS** e **JavaScript**, o **Bootstrap** permite que desenvolvedores criem layouts modernos e funcionais de maneira eficiente e com menos esforço.

Como utilizar

Para utilizar em um projeto, é necessário incluir seus arquivos **CSS** e **JavaScript**, que podem ser baixados ou referenciados via Rede de Distribuição de Conteúdo (CDN, Content Delivery Network). Com o **Bootstrap**, os desenvolvedores podem aplicar classes pré-definidas aos elementos **HTML** para estilizar componentes e layout da página. A biblioteca oferece um sistema de *grid* flexível, que facilita a criação de layouts responsivos, adaptando-se automaticamente a diferentes tamanhos de tela. Além disso, o **Bootstrap** inclui diversos *plugins* **JavaScript** que adicionam funcionalidades interativas aos componentes.

Modelos prontos

Fornece uma variedade de modelos prontos (templates) que servem como ponto de partida para o desenvolvimento de sites e aplicações. Esses templates cobrem diversas necessidades, desde páginas iniciais simples até *dashboards* (painéis) complexos e sites de comércio. Utilizando esses modelos, os desenvolvedores podem economizar tempo, garantindo ao mesmo tempo que os designs sigam as melhores práticas de usabilidade e responsividade. Esses modelos podem ser personalizados facilmente, permitindo a adaptação às especificidades de cada projeto.

Estudo de Caso

COSTOU DESSE CONTEÚDO?

ACESSE NOSSAS REDES E FIQUE POR
DENTRO DO TRABALHO DO PET.COMP!



petcomp.cefetmg.br



pet.comp.cefetmg@gmail.com



[@pet.comp.cefetmg](https://www.instagram.com/pet.comp.cefetmg)

PET.COMP