**Conceito de Sistema Computacional**

O computador é uma máquina ou dispositivo capaz de executar uma sequência de instruções definidas pelo homem para gerar um determinado resultado, o qual atenda a uma necessidade específica (ex.: realizar cálculos, gerar relatórios). Essa sequência de instruções é denominada algoritmo, o qual pode ser definido como um conjunto de regras expressas por uma sequência lógica finita de instruções, que ao serem executadas pelo computador, resolvem um problema específico. Assim, podemos dizer que um ou mais algoritmos compõem o que conhecemos como programa de computador, que no âmbito profissional da área de informática é conhecido como software.

As partes físicas de um computador, tais como: dispositivos de entrada e saída (ex.: monitor, teclado, impressora, webcam), dispositivos de armazenamento (ex. memória volátil e permanente), processador, assim como todo o conjunto de elementos que compõem um computador são chamados de hardware. A Figura 1.1 apresenta os elementos que compõem o hardware.

Desenho de um cachorro

Descrição gerada automaticamente

**Dispositivo:** mecanismo usado para obter um resultado.

**Instruções:** ordens ou comandos para que um computador execute uma solicitação do homem (usuário), a fim de atender a uma determinada necessidade.

Dessa forma, pode-se dizer que a combinação do hardware e do software forma o sistema computacional.

**Computadores Analógicos x Digitais**

Os computadores podem ser classificados em dois tipos principais: analógicos e digitais.

Os computadores analógicos não trabalham com números nem com símbolos que representam os números; eles procuram fazer analogia entre quantidades (ex. pesos, quantidade de elementos, níveis de tensão, pressões hidráulicas). Alguns exemplos desse tipo de computador são o Ábaco – que se utilizava de pequenos carretéis embutidos em um pequeno filete de metal para realizar cálculos – ou a régua de cálculo – que utiliza comprimentos de escalas especialmente calibradas para facilitar a multiplicação, a divisão e outras funções.

Imagem de jogo de vídeo game

Descrição gerada automaticamente com confiança baixaTela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Pode-se dizer que o computador analógico é uma categoria de computadores que se utiliza de eventos elétricos, mecânicos ou hidráulicos para resolver problemas do homem. Ou seja, tais computadores representam o comportamento de um sistema real utilizando-se para isso de grandezas físicas.

Diferentemente dos computadores analógicos, que representam números por meio da analogia direta entre quantidades, os computadores digitais resolvem problemas realizando operações diretamente com números, enquanto os analógicos medem. Os computadores digitais resolvem os problemas realizando cálculos e tratando cada número, dígito por dígito.

Tela de um computador

Descrição gerada automaticamente com confiança médiaInterface gráfica do usuário, Site

Descrição gerada automaticamente

Outra grande diferença dessa categoria de computadores é que eles podem resolver problemas por meio de uma sequência programada de instruções com o mínimo de intervenção humana.

Assim, podemos dizer que o computador digital surgiu como uma solução rápida e com um nível de automação bem mais elevado de realizar grandes computações numéricas.

**Dígitos:** são símbolos usados na representação numérica inteira ou fracionária.

Próximo

Monitor de computador

Descrição gerada automaticamente

Diversos dispositivos compõem a atual quinta geração (All-in-One, Desktops modernos, Notebooks, Tablets e Smartphones)

Atualmente dispomos de computadores extremamente velozes, que apresentam tamanhos cada vez menores; esses avanços são possíveis graças às pesquisas que não cessam, por parte de inúmeras universidades e instituições de pesquisa dispostas a descobrir o novo. Assim surgiu o que conhecemos por nanotecnologia, ou seja, a capacidade potencial de criar a partir da miniaturização, permitindo, assim, o desenvolvimento de dispositivos miniaturizados para compor um computador, por exemplo

**Componentes Básicos de um Sistema C**

Segundo a arquitetura de Von Neumann, os computadores possuem quatro componentes principais: Unidade Central de Processamento (UCP) – composta pela Unidade Lógica e Aritmética (ULA) e a Unidade de Controle (UC), a memória e os dispositivos de entrada e saída. Tais componentes são interconectadas por barramentos. E todos esses itens constituem o hardware de um computador (seu conjunto de componentes físicos), os quais são agrupados em módulos específicos, constituindo a estrutura básica de um computador. A Figura 1.8 mostra, de forma genérica, essa estrutura.

Cabe esclarecer que quando se fala em processador está se falando genericamente da UCP. Muitas pessoas usam a sigla UCP ou CPU para indicar o gabinete do computador, o que é errôneo.

Diagrama

Descrição gerada automaticamente

A seguir serão descritos cada um dos principais componentes de um computador:

**UCP (Unidade Central de Processamento)**

Sigla representativa de Unidade Central de Processamento. Podemos dizer que se trata do componente principal do computador. Algumas pessoas chamam de processador ou microprocessador. É responsável pela execução de dados e instruções armazenadas em memória (código de programas e dados);

**Memória**

Existem diversos tipos de memória em um computador (ex.: RAM (principal), ROM, cache, registradores), mas existe uma delas denominada memória principal, a qual é indispensável. A memória principal é tão importante quanto a UCP, pois sem ela não seria possível disponibilizar os programas e seus dados para o processamento pela CPU. Portanto, a memória é responsável por armazenar todos os programas que executam no computador e os dados que utilizam;

**Dispositivos de Entrada e Saída (E/S)**

São dispositivos responsáveis pelas entradas e saídas de dados, ou seja, pelas interações entre o computador e o mundo externo (usuários). São exemplos de dispositivos de E/S: monitor de vídeo, teclado, mouse, webcam, impressora, entre outros;

**Barramento**

É responsável por interligar todos os componentes listados acima. Trata-se de uma via de comunicação composta por diversos fios ou condutores elétricos por onde circulam os dados manipulados pelo computador.

Próximo

### Bases e sistemas de numeração

Desde o início de sua existência, o homem sentiu a necessidade de contar objetos, fazer divisões, diminuir, somar, entre outras operações aritméticas de que hoje se tem conhecimento. Diversas formas de contagem e representação de valores foram propostas. Podemos dizer que a forma mais utilizada para a representação numérica é a **notação posicional**.

Segundo Monteiro (2007), na notação posicional, os algarismos componentes de um número *assumem valores diferentes, dependendo de sua posição* relativa nele. O valor total do número é a soma dos valores relativos de cada algarismo. Dessa forma, dependendo do sistema de numeração adotado, é dito que a quantidade de algarismos que o compõem é denominada **base**.

Assim, a partir do conceito de notação posicional, tornou-se possível a conversão entre diferentes bases.

A civilização ocidental adotou um sistema de numeração que possui dez algarismos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), denominado de sistema decimal. A quantidade de algarismos de um dado sistema é chamada de base; portanto, no sistema decimal a base é 10.

O sistema binário possui apenas dois algarismos (0 e 1), sendo que sua base é 2.

Próximo

### Conversão de Bases

O sistema de numeração que utilizamos é o decimal, e consiste em dez valores possíveis (0, 1, 2, 3, 4, 5, 6, 7,8 e 9) para cada casa numérica. Entretanto, no ramo da eletrônica e principalmente da computação são utilizadas outras bases numéricas, tais como a base binária, octal e hexadecimal. Deste modo, conforme a demanda e utilização de equipamentos eletrônicos aumenta, entender o funcionamento destas bases se torna ainda mais importante.

Uma das bases mais utilizadas pelos programadores é a base binária, onde apenas os números 0 e 1 podem assumir as casas numéricas. Este sistema de numeração se tornou o mais utilizado no setor, pois os mesmos números são utilizados para controlar o funcionamento (1) ou o desligamento (0) dos componentes. Sendo assim, vamos aprender agora a realizar a conversão do **sistema decimal para o sistema binário**.

O Quadro a seguir mostra a equivalência entre as bases decimal, binária, octal e hexadecimal:

Tabela

Descrição gerada automaticamente

**Conversão de Decimal para Binário**

Vamos realizar a conversão do número 500. O primeiro passo é realizar a divisão do número pelo número de sua base. Como estamos convertendo o número para a base binária, devemos dividi-lo pelo número 2. Na sequência, continuaremos dividindo os resultados até atingir o resultado 0. Veja abaixo:

Tabela

Descrição gerada automaticamente

Feito isso, devemos agora escrever o número binário, que será representado pelos restos da divisão, escritos debaixo (o último resto) para cima (o primeiro resto). Para o exemplo acima, o número binário que representa o número 500 é:

Tabela

Descrição gerada automaticamente

Outros exemplos de conversão:

Converter o número 29 do sistema decimal para binário.

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Converter o número 10 do sistema decimal para binário.

Diagrama, Esquemático

Descrição gerada automaticamente com confiança média

Converter o número 15 do sistema decimal para binário.

Diagrama, Esquemático

Descrição gerada automaticamente

O número binário é formado pelo quociente da ultima divisão e os restos das divisões sucessivas da direita para a esquerda , Todo número decimal par quando convertido para binário termina em zero, e todo número decimal impar termina em um quando convertido para binário.

Próximo

**Conversão de Binário para Decimal**

Podemos também realizar o processo inverso, ou seja, **transformar o número binário em um no sistema decimal**. Para isso vamos transformar o número 11010111 para o sistema de medida decimal.

O primeiro passo é escrever as potências do número 2, da direita para a esquerda, começando em 0 e aumentar o expoente até *n-1* números do binário. Como nosso exemplo tem 8 dígitos, o maior expoente será 7.

Em seguida, devemos reescrever o binário e realizar a multiplicação do valor do binário pela sua potência respectiva. Assim, os valores correspondentes ao número 0 se tornarão 0 e os correspondentes ao número 1 se manterão iguais. Deste modo, temos que:

Tabela

Descrição gerada automaticamente

Para converter um número binário para decimal, basta multiplicar cada bit pelo seu valor de posição e somar os resultados.

**Exemplos:**

Converter o número 1010 do sistema binário para decimal.

Tabela

Descrição gerada automaticamente

Converter o número 1101 do sistema binário para decimal.

Tabela

Descrição gerada automaticamente

Converter o número 1111 do sistema binário para decimal.

Tabela

Descrição gerada automaticamente

Próximo

**Funções e Portas Lógica**

De acordo com Monteiro (2007), um computador é constituído de elementos eletrônicos, como resistores, capacitores e principalmente transistores. Nesses computadores, os transistores são, em geral, componentes de determinados circuitos eletrônicos que precisam armazenar os sinais binários e realizar certos tipos de operações com eles. Esses circuitos, chamados de circuitos digitais, são formados por pequenos elementos capazes de manipular grandezas apenas binárias. Os pequenos elementos referidos são conhecidos como portas (gates) lógicas, por permitirem ou não a passagem desses sinais, e os circuitos que contêm as portas lógicas são conhecidos como circuitos lógicos.

Portas lógicas ou circuitos lógicos, são dispositivos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito. São geralmente usadas em circuitos eletrônicos, por causa das situações que os sinais deste tipo de circuito podem apresentar: presença de sinal, ou "1"; e ausência de sinal, ou "0". As situações "Verdadeira" e "Falsa" são estudadas na Lógica Matemática ou Lógica de Boole; origem do nome destas portas. O comportamento das portas lógicas é conhecido pela tabela verdade que apresenta os estados lógicos das entradas e das saídas.

Assim, circuitos lógicos são montados a partir da combinação de uma unidade básica construtiva denominada porta lógica, a qual é obtida mediante a combinação de transistores e dispositivos semicondutores auxiliares. Portanto, a porta lógica é a base para a construção de qualquer sistema digital (ex.: o microprocessador).

Em geral, os circuitos lógicos são agrupados e embutidos em um Circuito Integrado (CI). Esses dispositivos implementam uma determinada função com o objetivo de cumprir uma tarefa específica.

Portas lógicas são encontradas desde o nível de integração em Ultra Larga Escala (ULSI) ou Super Larga Escala (SLSI) até o nível de integração existente em circuitos digitais mais simples, desempenhando funções mais básicas (ex.: comparações, somas, multiplicações).

Próximo

Tabela

Descrição gerada automaticamente

Conforme é possível observar, a regra é: “se o primeiro operando é 1 e o segundo operando é 1, o resultado é 1 (Verdadeiro), senão o resultado é 0 (Falso)”.

Um exemplo de aplicação de uma porta AND seria na composição de um circuito para a transferência de bits de dados de um local para outro (ex.: da memória para a CPU). Nesse caso, a finalidade seria a de garantir que um bit de origem seja o mesmo bit de destino (MONTEIRO, 2007).

Próximo

### Porta OR (OU)

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura abaixo. Os operandos são binários simples (0 e 1).

Diagrama, Diagrama de Venn

Descrição gerada automaticamente

Podemos dizer que a operação OR simula uma soma de binários, permitindo os possíveis resultados conforme mostra a Tabela abaixo:

Tabela

Descrição gerada automaticamente

Conforme podemos observar, a regra é: “se o primeiro operando é 1 ou o segundo operando é 1, ou se os dois operandos forem 1, o resultado é 1, senão o resultado é 0”.

As operações lógicas OR são muito utilizadas em lógica digital ou mesmo em comandos de decisão de algumas linguagens de programação (ex.: Se (X=1 OU Y=1) Então Executa uma ação) (MONTEIRO, 2007).

Próximo

**Porta NOT (NÃO)**

A porta NOT representa um inversor. Essa operação aceita apenas um operando ou uma entradas (A), conforme mostra a Figura abaixo. O operando pode ser um dígito binário (0 ou 1).

Forma

Descrição gerada automaticamente

Pode-se dizer que a operação NOT realiza a inversão de um dígito binário, permitindo os possíveis resultados conforme mostra a Tabela abaixo:

Aplicativo

Descrição gerada automaticamente com confiança baixa

Conforme é possível observar, a regra é: “se o operando for 1, o resultado é 0, senão o resultado é 1”.

Entre as principais aplicações dos circuitos inversores está a representação de números negativos no formato binário, quando se usa o que é chamado de complemento a 1 ou complemento a 2, fazendo-se necessária a inversão de um grupo de bits representativos de um número negativo (MONTEIRO, 2007).

Próximo

**Porta XOR (OU exclusivo)**

A denominação XOR é a abreviação do termo EXCLUSIVE OR. Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura abaixo. Os operandos são binários simples (0 e 1).

Diagrama, Diagrama de Venn

Descrição gerada automaticamente

Pode-se dizer que a operação XOR possui como principal função a verificação de igualdade, permitindo os possíveis resultados conforme mostra a Tabela a seguir:

Tabela

Descrição gerada automaticamente

Conforme é possível observar, a regra é: “se o primeiro operando ou o segundo operando for igual a 1, o resultado é 1; senão, o resultado é 0”. Ou seja, para entradas iguais a saída será 0 e para entradas diferentes a saída será 1.

A porta XOR possui inúmeras aplicações, sendo um elemento lógico bastante versátil, permitindo, por exemplo, a fabricação de um testador de igualdade entre valores, para testar, de modo rápido, se duas palavras de dados são iguais (MONTEIRO, 2007).

Próximo

**Outras Portas Lógicas e Circuitos C**

Existem outras portas lógicas derivadas das portas lógicas apresentadas acima, tais como as portas NAND (porta AND invertida) e a porta NOR (porta OR invertida). Ambas são apresentadas, respectivamente, nas Figuras abaixo:

Porta NAND:Forma, Quadrado

Descrição gerada automaticamente

Porta NOR:Diagrama, Diagrama de Venn

Descrição gerada automaticamente

As operações lógicas são realizadas em dois passos: primeiro a operação AND ou OR e, em seguida, o seu resultado é invertido. Esse tipo de portas lógicas também possui diversas aplicações, sendo utilizado para reduzir a complexidade e a quantidade de portas lógicas necessárias a um determinado circuito lógico.

Não esqueça de que um circuito lógico pode possuir diversas portas lógicas e, portanto, suas tabelas verdade poderão ter inúmeras entradas e inúmeras saídas (as quais poderão ser representadas por suas respectivas equações booleanas). A Figura a seguir mostra o resumo dos símbolos gráficos e matemáticos (equação booleana) de portas lógicas.

Diagrama

Descrição gerada automaticamente

A partir das portas lógicas básicas, é possível interligar diversas de suas unidades, de modo a construir redes lógicas, também chamadas de circuitos combinatórios. Monteiro (2007) explica que um circuito combinatório é definido como um conjunto de portas lógicas cuja saída em qualquer instante de tempo é função somente das entradas. O autor afirma ainda que existe outra categoria de circuitos que combina portas lógicas, denominada circuitos sequenciais, os quais, além de possuir portas, contêm elementos de armazenamento (uma espécie de memória).

A Figura abaixo exemplifica um circuito combinatório implementado para uma determinada função.

Diagrama

Descrição gerada automaticamente

Dentre os exemplos de circuitos combinatórios, destacamos o das funções essenciais da maioria dos computadores e sistemas digitais, ou seja, a realização de operações aritméticas tais como: adição, subtração, multiplicação e divisão. Essas operações são realizadas na Unidade Lógica Aritmética (ULA) destes sistemas digitais, onde uma série de portas lógicas são combinadas para adicionar, subtrair, multiplicar ou dividir números binários. No caso das operações de multiplicação e divisão, além das portas lógicas, há a necessidade de circuitos sequenciais.

Próximo

### Sites e Aplicações WEB

### ****Qual a diferença entre SITE e APLICAÇÃO WEB?****

Não há uma definição exata que diferencie claramente sites e aplicações web. Alguns defendem que os sites são read-only (somente leitura) enquanto as aplicações web são read-write (leitura e escrita). Nessa definição, os sites apenas fornecem conteúdo enquanto as aplicações web podem fornecer e/ou receber conteúdo.

Outras pessoas preferem utilizar o critério do propósito para classificar como site ou aplicação web. Por exemplo, se o propósito é divulgar as informações de uma empresa, os dados de um produto, as notícias de um determinado assunto, utiliza-se o termo site. Se o propósito é criar uma ferramenta para controlar as atividades administrativas de uma organização, utiliza-se a denominação aplicação web.

Independentemente do critério de classificação utilizado, os tópicos discutidos nesse treinamento são importantes tanto para o desenvolvimento de sites quanto para o desenvolvimento de aplicações web.

**Navegadores**

As pessoas acessam sites e aplicações web através de navegadores (browsers) como Chrome, Firefox, Internet Explorer e Safari. Normalmente, esses navegadores possuem algumas diferenças na forma de exibir as páginas web aos usuários. Antigamente, essas diferenças eram maiores. Com o passar do tempo, os navegadores ficaram cada vez mais parecidos nesse aspecto.

Contudo, os desenvolvedores web ainda devem tomar cuidado com essas diferenças.

O objetivo de um navegador da web (Chrome, Edge, Firefox, Safari) é ler documentos HTML e exibi-los.

O navegador não exibe as tags HTML, mas as utiliza para determinar como exibir o documento, como no exemplo a seguir:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

A estrutura de uma página em HTML funciona como caixas(ou blocos) que podem ser colocados um dentro do outro, de maneira a permitir a organização dos conteúdos por caixa ou bloco, usamos as tags para delimitar o início e o fim das marcações.

Analise o exemplo a seguir:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente **RESPONSIVIDADE**de sites entre diversos aparelhos.

Outro problema de compatibilidade importante é causado pela grande variedade de dispositivos que podem ser utilizados para acessar os sites e as aplicações web. Atualmente, as pessoas acessam os sites e as aplicações web através de computadores tradicionais, tablets, celulares, televisores, entre outros.

Esses dispositivos possuem telas de tamanhos diferentes. Dessa forma, os desenvolvedores web devem considerar essas diferenças na criação das páginas web.

Próximo

**Como funciona a WEB?**

Antes de iniciarmos nossos estudos é necessário que tenhamos alguns conceitos bem fixados, já que muitos desses termos serão mencionados no decorrer do curso.

**• WWW:** Abreviação de World Wide Web, é um sistema de documentos em hipermídia (hypertextos, vídeos, sons, imagens, etc.) que são interligados. É constantemente utilizada como sinônimo de internet.

**• Site:**Conjunto de páginas web que representam uma pessoa, empresa ou instituição na web.

**• Browser:**Também denominado de navegador, é um software utilizado para ler e interpretar as páginas web, possibilitando que os usuários interajam com as mesmas. É através do browser que “navegamos” na internet.

**• URL:** Uniform Resource Locator – Localizador Uniforme de Recursos é uma sequência de caracteres que define o endereço de um site ou recurso da web.

**• HTTP** (Hypertext Transfer Protocol): Protocolo de Transferência de Hipertexto é um protocolo de comunicação (presente na camada de aplicação, segundo o Modelo de Referência OSI) muito utilizado na internet.

Próximo

### O que é HTML?

**HTML**é a linguagem de marcação padrão para criar páginas da Web.

**HTML**significa **Hyper Text Markup Language**

HTML descreve a estrutura de uma página da Web

HTML consiste em uma série de elementos

Os elementos HTML informam ao navegador como exibir o conteúdo

Os elementos HTML são representados por tags

As tags HTML rotulam partes do conteúdo, como "cabeçalho", "parágrafo", "tabela" e assim por diante

Os navegadores não exibem as tags HTML, mas as usam para renderizar o conteúdo da página

### Um documento HTML simples tem a seguinte estrutura:

HTML

<!DOCTYPE html>

 <html>

<head>

<title>Título da Página</title>

</head>

 <body>

<h1>Meu primeiro cabeçalho</h1>

<p>Meu primeiro parágrafo.</p>

</body>

</html>

### Exemplo explicado

A <!DOCTYPE html>declaração define este documento como HTML5

O <html>elemento é o elemento raiz de uma página HTML

O <head>elemento contém meta informações sobre o documento

O <title>elemento especifica um título para o documento

O <body>elemento contém o conteúdo da página visível

O <h1>elemento define um cabeçalho grande

O <p>elemento define um parágrafo

## **Tags em HTML**

Tags HTML são nomes que pertencem aos elementos entre colchetes e que delimitam a abertura e o encerramento de uma marcação, por exemplo:

### <tag de abertura> o conteúdo que você deseja exibir entra aqui ... </tag de encerramento>

Observe que as tags HTML normalmente vêm **em pares** como <p>e</p>

A primeira tag em um par é a **tag inicial,** a segunda tag é a **tag final**

A tag final é escrita como a tag inicial, mas com uma **barra** inserida antes do nome da tag

Próximo

### Vamos criar nossa primeira página!

## **Criando nossa primeira página!**

As páginas da Web podem ser criadas e modificadas usando editores profissionais de HTML.

No entanto, para aprender HTML, recomendamos um editor de texto simples, como o Bloco de Notas (PC).

Acreditamos que o uso de um editor de texto simples é uma boa maneira de aprender HTML.

Siga as etapas abaixo para criar sua primeira página da web com o Bloco de notas ou o Editor de texto.

## **Etapa 1: Abrir o bloco de notas (PC)**

**Windows 8 ou posterior:**

Abra a **tela inicial** (o símbolo da janela na parte inferior esquerda da tela). Digite **Notepad** .

**Windows 7 ou anterior:**

Abra **Iniciar** > **Programas> Acessórios> Bloco de Notas**

## **Etapa 2: Escreva algum HTML**

Escreva ou copie algum HTML no bloco de notas.

<!DOCTYPE html> <html lang="pt-br">  <head>    <meta charset="UTF-8" />    <title>Modelo de Site</title>  </head>  <body>    <h1>Meu primeiro cabeçalho</h1>    <p>Meu primeiro parágrafo.</p>    <p>Meu segundo parágrafo.</p>  </body> </html>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## **Etapa 3: salvar a página HTML**

Salve o arquivo no seu computador. Selecione **Arquivo> Salvar como** no menu Bloco de notas.

Nomeie o arquivo **"index.htm"** e defina a codificação para **UTF-8** (que é a codificação preferida para arquivos HTML).

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Você pode usar .htm ou .html como extensão de arquivo. Não há diferença, depende de você.

## **Etapa 4: Exibir a página HTML no seu navegador**

Abra o arquivo HTML salvo no seu navegador favorito (clique duas vezes no arquivo ou clique com o botão direito do mouse - e escolha "Abrir com").

O resultado será mais ou menos assim:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Próximo

**O que mais podemos fazer com HTML?**

**Cabeçalhos do HTML**

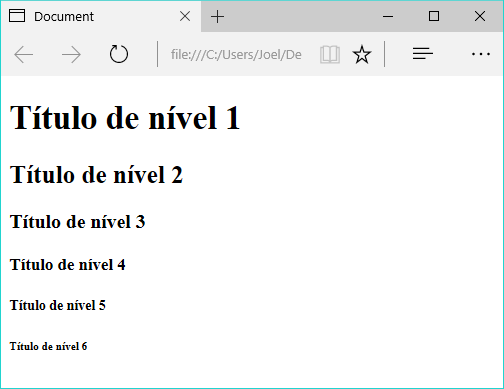
Cabeçalhos são normalmente utilizados para identificar páginas e seções e possuem aparência diferenciada do restante do texto. No HTML há seis níveis de cabeçalhos/títulos que podem ser utilizados por meio das tags h1, h2, h3, h4, h5 e h6, sendo h1 o maior/mais relevante e h6 o menor/menos relevante.

De acordo com as regras de SEO, é recomendado que uma página possua apenas uma tag <h1> que indique seu assunto, pois essa tag informa aos motores de busca qual sua principal palavra-chave.

No Exemplo de Código podemos ver como utilizar essas tags, e na **Figura 1** temos o resultado desse código. Para testar esse código, basta copiá-lo para o espaço indicado na **Linha 8** do Exemplo de Código e salvar o documento com a extensão .html.

<h1>Título de nível 1</h1> <h2>Título de nível 2</h2> <h3>Título de nível 3</h3> <h4>Título de nível 4</h4> <h5>Título de nível 5</h5> <h6>Título de nível 6</h6>

**Exemplo 1**. Diferentes níveis de cabeçalho



**Figura 1**. Diferentes níveis de cabeçalho

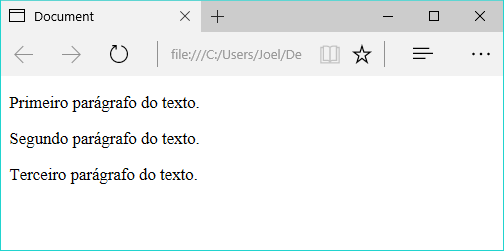
**Parágrafos no HTML**

Parágrafos de texto são gerados na HTML por meio das tags <p> </p>. Esse é um exemplo de tag cuja disposição na tela se dá em forma de bloco, ou seja, um parágrafo é posto sempre abaixo do outro.

O código do Exemplo a seguir adiciona três parágrafos, que podem ser visualizados na **Figura 2**.

<p>Primeiro parágrafo do texto.</p> <p>Segundo parágrafo do texto.</p> <p>Terceiro parágrafo do texto.</p>

**Exemplo 2**. Adicionando parágrafos



**Figura 2**. Parágrafos visualizados na página

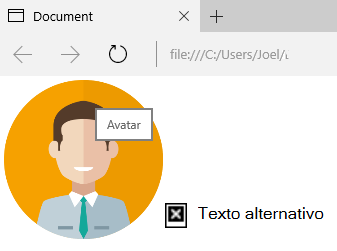
**Imagens no HTML**

A inserção de imagens em uma **página HTML** pode ser feita por meio da tag img, que recebe no atributo src o endereço do arquivo a ser carregado. Além desse, outros dois atributos importantes são o alt, que indica um texto alternativo que será exibido caso o arquivo não possa ser carregado, e title, que indica o texto que aparecerá como tooltip ao passar o mouse sobre a figura.

O código do Exemplo a seguir insere uma imagem a partir do arquivo avatar.png, localizado na mesma pasta do arquivo HTML, e uma tag img apontando para um arquivo inexistente. Observe na **Figura 3** que enquanto a primeira aparece corretamente, a segunda é exibida com um ícone de erro o texto alternativo que informamos. Note, ainda, que sobre a primeira figura está sendo exibido o tooltip definido no atributo title.

<code><img src="avatar.png" alt="Texto alternativo" title="Avatar" /> <img src="arquivo\_inexistente.jpg" alt="Texto alternativo" title="Outra figura"/></code>

**Exemplo 3**. Inserindo imagem



**Figura 3**. Exemplos de uso da tag img

No caso da tag img, ainda temos as tags figure e figcaption que serão abordadas no tópico de HTML Semântico, porém, a sua estrutura é simples e é representada pelo código abaixo:

<figure> <img src="http://meusite.com.br/assets/imagem.jpg" alt="Imagem"> <figcaption>Legenda da figura 1</figcaption> </figure>

**Exemplo 3.1**. Imagem com legenda

**Links no HTML**

Links são normalmente utilizados para direcionar o usuário para outras páginas, ou para outras partes da mesma página. Nos dois casos, utilizamos a tag a, que possui o atributo href no qual indicamos o destino daquele link.

O **Exemplo 4** mostra como adicionar um link para outra página, neste caso, indicada pelo arquivo pagina2.html.

<a href="pagina2.html">Página 2</a>

**Exemplo 4**. Inserindo link

Já o **Exemplo 5** mostra como adicionar um link para um elemento na mesma página. Nesse caso, ao clicar no link o browser mudará o foco para o elemento que possui o atributo id igual àquele indicado no href.

<a href="#paragrafo3">Ir para o parágrafo 3</a> <!--outros elementos--> <p id="paragrafo3">Parágrafo no rodapé.</p>

**Exemplo 5**. Inserindo link com id

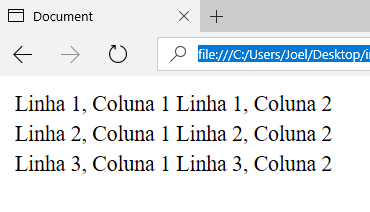
Note também que nesse caso o href requer ainda o sinal de cerquilha (#) antes do id do elemento que será o foco do link.

**Tabelas no HTML**

Tabelas são elementos utilizados com frequência para exibir dados de forma organizada em linhas e colunas. No HTML, elas são formadas por três tags básicas: table, para delimitar a tabela; tr, para indicar as linhas; e td para formar as colunas. O **Exemplo 6** mostra um exemplo simples de tabela com três linhas e duas colunas, e seu resultado pode ser visto na **Figura 4**.

<table> <tr> <td>Linha 1, Coluna 1</td> <td>Linha 1, Coluna 2</td> </tr> <tr> <td>Linha 2, Coluna 1</td> <td>Linha 2, Coluna 2</td> </tr> <tr> <td>Linha 3, Coluna 1</td> <td>Linha 3, Coluna 2</td> </tr> </table>

**Exemplo 6**. Exemplo de uso de tabelas

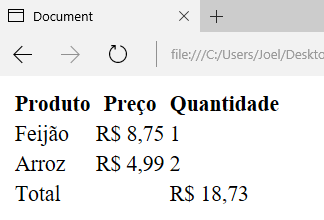


**Figura 4**. Exemplo de tabela com três linhas e duas colunas

Existem ainda outras três tags utilizadas para delimitar, de forma mais organizada, as partes da tabela: thead para o cabeçalho; tbody para o corpo; e tfoot para o rodapé. O **Exemplo 7** traz um exemplo de tabela mais complexa, utilizando todas as tags. Seu resultado é apresentado na **Figura 5**.

<table> <thead> <tr> <th>Produto</th> <th>Preço</th> <th>Quantidade</th> </tr> </thead> <tbody> <tr> <td>Feijão</td> <td>R$ 8,75</td> <td>1</td> </tr> <tr> <td>Arroz</td> <td>R$ 4,99</td> <td>2</td> </tr> </tbody> <tfoot> <tr> <td>Total</td> <td></td> <td>R$ 18,73</td> </tr> </tfoot> </table>

**Exemplo 7**. Tabela com cabeçalho e rodapé



**Figura 5**. Tabela com cabeçalho e rodapé

Por padrão, as tabelas não possuem bordas. Isso deve ser adicionado por meio das CSS.

**Listas no HTML**

Listas são elementos úteis para organizar e ordenar itens que estão relacionados de alguma forma. No HTML é possível criar três tipos de listas: ordenadas (com a tag ol), não ordenadas (com a tag ul), e de definição (por meio da tag dl).

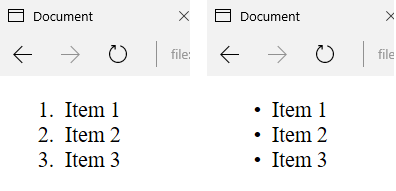
No **Exemplo 8** temos um exemplo de lista ordenada, enquanto o **Exemplo 9** traz um exemplo de lista não ordenada. Em seguida, a **Figura 6** mostra o resultado dos dois códigos. Observe que cada item das primeiras listas é definido pela tag li.

<ol> <li>Item 1</li> <li>Item 2</li> <li>Item 3</li> </ol>

**Exemplo 8**. Lista ordenada

<ul> <li>Item 1</li> <li>Item 2</li> <li>Item 3</li> </ul>

**Exemplo 9**. Lista não ordenada



**Figura 6**. Listas ordenada e não ordenada

No caso das listas ordenadas temos o atributo type que permite especificar o tipo de marcador a ser usado: números (padrão), números romanos ou letras.

<ol type="I"> <li>Item 1</li> <li>Item 2</li> <li>Item 3</li> </ol>

**Exemplo 8.1**. Lista ordenada por número romano maiúsculo

Os valores permitidos para esse atributo são:

1 - números (padrão)

i - números romanos minúsculos

I - números romanos maiúsculos

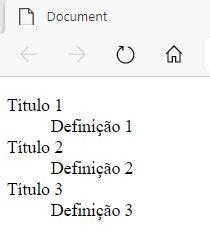
a - letras minúsculas

A - letras maiúsculas

As listas de definição têm um comportamento um pouco diferente, uma vez que cada item é composto por um título (dt) e uma definição (dd), semelhante ao que ocorre em dicionários, nos quais temos os verbetes e suas definições. No**Exemplo 10** temos um exemplo desse tipo de lista, com seu resultado ilustrado na **Figura 7**.

<dl> <dt>Título 1</dt> <dd>Definição 1</dd> <dt>Título 2</dt> <dd>Definição 2</dd> <dt>Título 3</dt> <dd>Definição 3</dd> </dl>

**Exemplo 10**. Exemplo de lista de definição



**Figura 7**. Listas de definição

**Formulários no HTML**

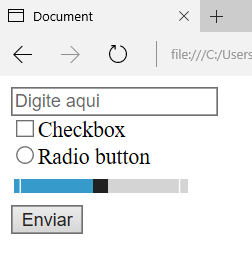
Formulários são normalmente utilizados para integrar a página HTML a algum processamento no lado servidor. Nesses casos, a página envia dados para uma aplicação (Java, PHP, .NET etc.), que os recebe, trata e retorna algum resultado.

No HTML, geralmente usamos a tag form para delimitar a área na qual se encontram os campos a serem preenchidos pelo usuário, a fim de serem enviados para processamento no back-end (enquanto a página HTML é chamada de front-end da aplicação).

O **Exemplo 11** mostra um exemplo de formulário com vários tipos de campos para entrada de dados e um botão para submetê-los ao servidor. Na **Figura 8** vemos seu resultado.

<form action="salvar\_dados.php" method="post"> <input type="text" placeholder="Digite aqui"> <br/> <input type="checkbox">Checkbox <br/> <input type="radio">Radio button <br/> <input type="range"> <br/> <input type="submit" value="Enviar"> </form>

**Exemplo 11**. Formulário com campos



**Figura 8**. Formulário para envio de dados

**Formatação de texto**

As tags de formatação de texto ajudam a destacar trechos da parte escrita da página, seja para fins de SEO ou por requisitos do conteúdo. Formatações como negrito e itálico podem ser aplicadas com facilidade utilizando as várias tags disponíveis para esse fim:

b e strong para negrito/texto forte;

i e em para itálico/ênfase;

sup e sub para sobrescrito e subscrito, respectivamente;

ins e del para indicar trechos que foram incluídos ou removidos, respectivamente;

small para textos menores que o padrão;

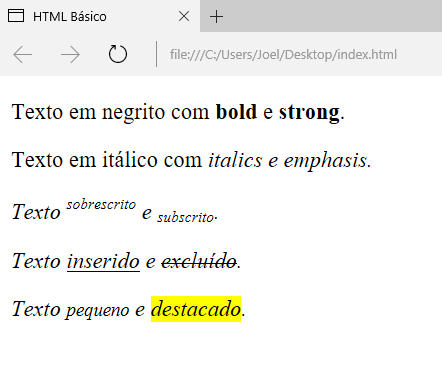
mark para texto destacado.

Apesar da tag <b> e <strong> terem resultados visualmente idênticos, eles têm usos diferentes. A tag <b> deve ser utilizada quando quer se destacar algo que é apenas visual, como por exemplo, deixar um link com mais visível. A tag <strong> por sua vez, além de destacar o visual, dá um destaque semântico ao texto, então se o conteúdo que você quer dar destaque é uma parte relevante do texto, utilize a tag <strong>. O mesmo vale para as tags <i> e <em>, onde a tag <i> é utilizada apenas para formatação visual e a tag <em>dá ênfase semântica ao texto, não devendo ser utilizada para uso puramente estético.

No **Exemplo 12** vemos exemplos de uso dessas tags, e seu resultado pode ser observado em seguida, na **Figura 9**.

<p>Texto em negrito com <b>bold</b> e <strong>strong</strong>.</p> <p>Texto em itálico com <i>italics<i> e <em>emphasis</em>.</p> <p>Texto <sup>sobrescrito</sup> e <sub>subscrito</sub>.</p> <p>Texto <ins>inserido</ins> e <del>excluído</del>.</p> <p>Texto <small>pequeno</small> e <mark>destacado</mark>.</p>

**Exemplo 12**. Tags de formatação de texto



**Figura 9**. Textos com formatação especial

**Áudio no HTML**

Com HTML também é possível inserir áudio e vídeo nas páginas com facilidade. Para áudio, podemos utilizar a tag audio, da mesma forma que apresentada No **Exemplo** **13**.

<audio controls src="musica.mp3" > Seu browser não suporta áudio. </audio>

**Exemplo 13**. Inserindo áudio

Na tag audio, o atributo src aponta para o arquivo de áudio que será executado (MP3, OGG ou WAV). Já o atributo controls indica que devem ser exibidos os controles de gerenciamento do áudio (botões play, pause etc.). Além dele, outros também merecem destaque: autoplay, para fazer com que o áudio seja executado assim que for carregado; loop, para que o áudio seja executado repetidas vezes.

Caso o browser não ofereça suporte a essa tag, o texto contido eu seu interior será exibido.

Também podemos informar mais de um arquivo de áudio (opções alternativas de formato, por exemplo). Nesse caso, precisamos utilizar a tag source, como no exemplo do **Exemplo 14**.

<audio controls src="musica.mp3" >

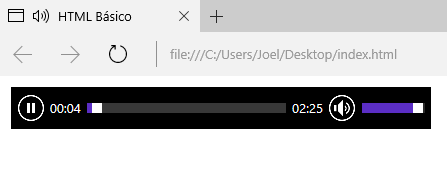
<source src="musica.ogg" type="audio/ogg">

<source src="musica.mp3" type="audio/mpeg"> </audio>

**Exemplo 14**. Inserindo áudio com a tag source

Dessa vez, os arquivos serão buscados na ordem em que aparecem na lista. Caso um não seja localizado ou não possa ser carregado, o browser buscará imediatamente o próximo.

A **Figura 10** ilustra o resultado desses códigos, com o controle de áudio exibido no browser. Note também que no topo da janela (na aba do navegador) é exibido um ícone informando ao usuário que essa página está executando um áudio.



**Figura 10**. Áudio em execução na página

**Vídeo no HTML**

Semelhante ao áudio, também podemos inserir vídeos nas páginas [HTML utilizando](https://htmleditor.tools/) a tag video, adicionada na HTML5, como No **Exemplo 15**.

<video src="video.mp4" width="640" height="480" controls> Seu browser não suporta vídeo HTML5. </video>

**Exemplo 15**. Inserindo vídeo

Dessa vez, além dos atributos já vistos no áudio, também precisamos informar a largura (width) e a altura (height) do vídeo, a fim de mantê-lo adequado ao layout.

O código do **Exemplo 16** pode ser utilizado quando desejarmos informar mais de um arquivo de vídeo como opções de carregamento, da mesma forma que vimos na tag audio. Nesse caso, os formatos suportados são MP4, WebM e Ogg.

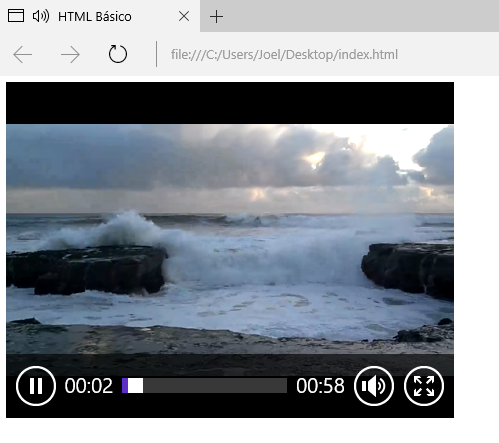
<video width="640" height="480" controls>

<source src="video.webm" type="video/webm">

<source src="video.mp4" type="video/mp4"> </video>

**Exemplo 16**. Inserindo vídeo

O resultado agora pode ser visto na **Figura 11**.



**Figura 11**. Vídeo em execução na página

**Div e Span no HTML**

As tags div e span são duas das mais utilizadas no HTML, com objetivos distintos, porém com grande importância para a composição do layout das páginas e formatação do texto.

As divs são normalmente utilizados para representarem containers para outros elementos, agrupando-os visualmente dentro de um bloco que pode conter dimensões e posição definidas. Por padrão, uma div não possui aparência características visuais definidas, isso precisa ser feito via CSS ao atribuir bordas, cores etc. Sua principal característica, no entanto, é que essa tag representa um elemento do tipo bloco, ou seja, que quando adicionado na página, automaticamente gera uma nova linha no layout (semelhante a um parágrafo), ao invés de ser alocado lateralmente nos demais componentes.

O código do **Exemplo 17** demonstra um uso básico das divs.

<input type="text" value="input 1">

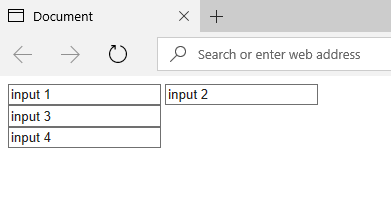
<input type="text" value="input 2">

<div><input type="text" value="input 3"></div>

<div><input type="text" value="input 4"></div>

**Exemplo 17**. Inserindo divs

Perceba no resultado da **Figura 12** que, enquanto os dois primeiros inputs são dispostos lateralmente, os dois últimos aparecem um abaixo do outro, uma vez que estão dentro de divs diferentes.



**Figura 12**. Exemplos de uso das divs

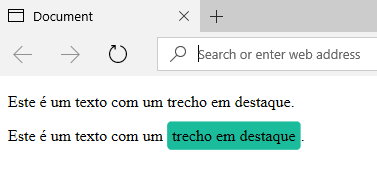
Já a tag span é um elemento do tipo inline, ou seja, quando adicionado na página, ele é inserido lateralmente após os demais componentes, diferente das divs que são elementos do tipo bloco.

Elementos span, por padrão, também não possuem nenhuma característica visual definida, isso precisa ser feito [via CSS](https://htmlcheatsheet.com/css/) para destacar ou aplicar uma formatação especial para um certo trecho do texto. Por exemplo, o **Exemplo 18** demonstra o uso do span em dois casos. No primeiro, a tag não conta com nenhum atributo adicional; no segundo, adicionamos a ela uma aparência diferenciada via CSS.

<p>Este é um texto com um <span> trecho em destaque</span>.</p> <p> Este é um texto com um <span style="background: #1abc9c; border-radius:5px; padding:5px"> trecho em destaque</span>. </p>

**Exemplo 18**. Trabalhando com span

O resultado pode ser visto na **Figura 13** e, como esperado, no primeiro caso não conseguimos perceber nenhuma diferença visual devido ao uso do span.



**Figura 13**. Exemplo de uso do span

Próximo

### Introduzindo o CSS

## CSS básico

CSS (Folha de Estilo em Cascata) é o código que você usa para dar estilo à sua página Web. *CSS básico* apresenta tudo que você precisa para começar. Responderemos a perguntas do tipo: Como mudo meu texto para preto ou vermelho? Como faço para que meu conteúdo apareça em determinados lugares na tela? Como decoro minha página com imagens e cores de fundo?

## **Então, o que realmente é CSS?**

Assim como o HTML, o CSS não é realmente uma linguagem de programação. Também não é uma linguagem de marcação — é uma *linguagem de folhas de estilos*. Isso significa que o CSS permite aplicar estilos seletivamente a elementos em documentos HTML. Por exemplo, para selecionar **todos** os elementos parágrafo de uma página HTML e tornar o texto dentro deles vermelho, você escreveria este CSS:

p { color: red; }

Vamos tentar: cole as três linhas de CSS acima em um novo arquivo, no seu editor de texto, e salve o arquivo como estilo.css na sua pasta estilos.

Ainda assim, precisamos aplicar o CSS ao seu documento HTML. Do contrário, o estilo CSS não irá afetar a maneira como o seu navegador mostra seu documento HTML (se você não estiver acompanhando o nosso projeto, leia Lidando com arquivos e HTML básico para descobrir o que você precisa fazer primeiro).

Abra seu arquivo index.html e cole o seguinte código em algum lugar no cabeçalho, ou seja, entre as tags <head> e </head>:<link href="estilos/estilo.css" rel="stylesheet">

Salve o arquivo index.html e abra ele no seu navegador. Você deve ver uma página como essa:

Se o texto do seu parágrafo estiver vermelho, parabéns! Você acabou de escrever seu primeiro CSS de sucesso.

### Anatomia de um conjunto de regras CSS

Vamos dar uma olhada no CSS acima com mais detalhes:

Toda essa estrutura é chamada de **conjunto de regras**(mas geralmente usamos o termo "regra", por ser mais curto). Note os nomes das partes individuais:

**Seletor (Selector)**

O nome do elemento HTML no começo do conjunto de regras. Ele seleciona o(s) elemento(s) a serem estilizados (nesse caso, elementos <p>). Para dar estilo a um outro elemento, é só mudar o seletor.

**Declaração (Declaration)**

Uma regra simples como color: red; especificando quais das **propriedades** do elemento você quer estilizar.

**Propriedades (Property)**

Forma pela qual você estiliza um elemento HTML. (Nesse caso, color é uma propriedade dos elementos <p>.) Em CSS, você escolhe quais propriedades você deseja afetar com sua regra.

**Valor da propriedade (Property value)**

À direita da propriedade, depois dos dois pontos, nós temos o **valor de propriedade**, que escolhe uma dentre muitas aparências possíveis para uma determinada propriedade (há muitos valores color(cor) além do red(vermelho)).

Note outras partes importantes da sintaxe:

Cada linha de comando deve ser envolvida em chaves ({}).

Dentro de cada declaração, você deve usar dois pontos (:) para separar a propriedade de seus valores.

Dentro de cada conjunto de regras, você deve usar um ponto e vírgula (;) para separar cada declaração da próxima.

Então para modificar múltiplos valores de propriedades de uma vez, você deve escrevê-los separados por ponto e vírgula, desse modo:

p {

 color: red;

 width: 500px;

 border: 1px solid black;

}