**Geração Tech Unimed BH Fullstack**

**DIO**

**Módulo 2 – Fundamentos Web com HTML e CSS**

**Primeiros passos para o desenvolvimento web**

**Conceitos**

* **Host**: servidor, de email por exemplo (gmail, outlook, yahoo, etc).
* **HTML**: linguagem de marcação para internet (não é linguagem de programação).
* **HTTP**: protocolo de comunicação.
* **URL**: endereço web.
* **Link**: ligação.
* **3G/4G/5G**: velocidades referentes à internet móvel.
* **Cookies**: pequenos registros de atividade que vários sites usam para exibir anúncios personalizados por exemplo.
* **IP**: endereço de um computador/site/e-mail na internet.
* **Redes**: conexão entre computadores.
* **Backbone**: estruturas complexas que gerenciam o tráfego da internet, isso é o que permite os computadores não precisarem ter muitas conexões.
* **Provedor de acesso**: regulam e permitem o acesso à internet.
* **DNS**: serviço que converte a url em um código que o computador entende.
* **127.0.0.1**: local host, ou seja, endereço IP do seu computador.
* **TCP/IP**: protocolos de comunicação entre computadores em rede.
* **TCP**: Transmission Control Protocol
* **IP**: Internet Protocol

**TCP/IP**

Para ocorrer uma transmissão de dados via internet é necessário passar por 4 camadas.

1. Física: por onde a conexão vai chegar ao computador, placa de rede
2. Rede: configuração que permite a conexão com a internet, IP
3. Transporte: forma de transporte de dados, TCP, UDP
4. Aplicação: tipo de dados que são enviados (quem envia os dados), FTP, SMTP, HTTP

**TCP x UDP**

Formas de conexão à rede.

UDP

* Rápido
* Não confiável (sem confirmação de envio ou recebimento)
* Não sabe quem está recebendo a informação
* Livestream

TCP

* Voltado à conexão.
* Lento.
* Possui confirmação de envio e recebimento.
* Integridade, ordem dos dados.
* Todos sabem pra quem estão enviando e quem está recebendo.
* Aplicativo de mensagem de texto (ordem importa).

**Portas**

Portas por onde os dados saem e chegam.

* 20: FTP – envio de arquivos
* 22: SSH – conexão segura entre computadores
* 25: SMTP – envio de emails
* 53: DNS – tradução de IP para nome e vice-versa
* 80: HTTP – requisição simples da internet
* 443: HTTPS – requisição segura da internet

**Dispositivos de conexão com a rede**

* **Modem**: hardware que converte dados em um formato que pode ser transmitido de um computador e lido por outro (modula e demodula sinais).
* Switch:
* **Roteador**: transforma conexão física em wireless, para distribuir internet.
* **Switch**: distribui internet para um ou mais dispositivos de uma rede, envia apenas para quem solicitou o dado.

**Celular, internet e outros dispositivos**

* **SMS**: envio de mensagens de texto entre celulares. Não tem custo para operadoras pois naturalmente os celulares trocam bits de informação com as torres e o SMS vai “de carona” nesse processo.
* **MMS**: mensagens multimídia (áudio e vídeo), precisam de uma forma mais robusta de conexão.
* **Conexões móveis** **(1G, 2G, 3G, 4G, 5G)**: relacionam a velocidade da rede.
* **Wi-FI**: padrões de comunicação sem fio definidos pelo IEEE.
* **Bluetooth**: conexão ponto a ponto que não depende da internet, existem várias classes e versões de Bluetooth, que indicam a taxa de transmissão e o alcance.

**Segurança na internet**

* **WEP**: chaves de 64 bits e de 128 bits.
* **WPA**: chaves trocadas periodicamente.
* **WPA2**: maior segurança, porém mais lento.

**Browser, sites aplicativos e webserver**

* **Browser**: programa que interpreta linguagem de programação utilizada para criar sites. Permite navegar pela internet.
* **Cache**: informações que sites deixam no computador para aumentar a velocidade de navegação.
* **Cookie**: “rastros” deixados por sites no computador que podem ser utilizados por outros sites para marketing, por exemplo.
* **Página estática**: pouca interação por parte do usuário.
* **Página dinâmica**: muita interação com o usuário.
* **Web-server**: servidor onde ficam armazenadas as informações dos arquivos, softwares, bancos de dados de serviços conectados à rede.
* **Web-service:** interface disponível para fazer requisições e consultas em banco de dados inacessíveis (permite fazer solicitações sem interagir diretamente com o banco de dados). Essa funcionalidade aumenta a segurança pois o usuário não tem acesso a todo o banco de dados, apenas as respostas para os tipos de solicitações que são permitidas. Exemplo, consulta de CEP.

Um site, ou aplicativo, precisa estar hospedado em um servidor para poder ser acessado, assim como o banco de dados de um site também precisam estar armazenados em um web-server.

**O que são Stacks?**

Pilhas de tecnologia, ou seja, conjuntos de softwares necessários e suficientes para executar um aplicativo / programa. Podem incluir: linguagem de programação, ambientes e ferramentas de interação, bancos de dados etc.

Uma stack corresponde ao ambiente tecnológico que a empresa utiliza, e é importante entender quais os pontos fortes e fracos desse ambiente. Se relaciona com a capacidade e limitação de performance da empresa.

**Front-end**

* Trabalha com a parte da frente do site ou app.
* Interface com o usuário (UI).
* Experiência de uso do usuário (UX).
* Lógica de programação: HTML, CSS, jQuery, JS, AJAX, PHP
* Bootstrap /outros frameworks.

**Back-end**:

* Parte de trás dos sites/app.
* Servidores, banco de dados.
* Meio de campo entre a interface e o banco de dados, regras de negócios, validações.
* MySQL, Oracle, protocolos de comunicação, PHP, Java, node.js

**Fullstack**

* Profissional que sabe trabalhar em todas as camadas das tecnologias de desenvolvimento/execução de um app ou software.

Interface gráfica do usuário

Descrição gerada automaticamente

Pesquisar sobre JQuery pois HTML está caindo em desuso.

Pesquisar sobre Reach pois está crescendo muito.

**Linguagens de programação, marcação e termos comuns**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

* **HTML e CSS**: são linguagens de marcação.
* **404**: erro de endereço inexistente.
* **ALT**: texto alternativo (quando passa o mouse em cima e aparece).
* **Biblioteca ou dll**: conjunto de ferramentas disponíveis para funções específicas.
* **Framework**: caixa de ferramentas disponível para desenvolvimento de software.
* **SSL**: camada de segurança.
* **XAMPP**: servidor web utilizado para criar aplicações (https://www.apachefriends.org/pt\_br/index.html).

**Introdução a criação de websites com HTML5 e CSS3**

**Conceitos**

Diagrama

Descrição gerada automaticamente com confiança baixa

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**HTML5**

**Definição e estrutura básica**

Com o HTML definimos o significado e a estrutura do conteúdo da web e, além de texto, nossas páginas precisam de imagens, vídeos e vários outros formatos e para isso temos os elementos HTML.

Um elemento HTML é formado pela tag de abertura e seus atributos, o conteúdo e uma tag de fechamento. Existem elementos que não tem tag de fechamento.

Com esses elementos podemos agrupar tipos de conteúdo, alterar tamanho e forma de fontes e adicionar diferentes mídias à nossa página na web.

E agora podemos ver como é a estrutura básica de um arquivo HTML.

A primeira linha do documento deve ser o **<!DOCTYPE html>**, apesar de parecer um elemento HTML ela apenas diz ao navegador que ele está lidando com um arquivo do tipo HTML5.

**<html>**

A tag html é a raiz do seu documento, todos os elementos HTML devem estar dentro dela. E nela nós informamos ao navegador qual é o idioma desse nosso documento, através do atributo lang, para o português brasileiro usamos pt-BR.

**<head>**

A tag head contém elementos que serão lidos pelo navegador, como os metadados - um exemplo é o charset, que é a codificação de caracteres e a mais comum é a UTF-8, o JavaScript com a tag script, o CSS através das tags style e link.

**<body>**

E dentro da tag body colocamos todo o conteúdo visível ao usuário: textos, imagens, vídeos.

**Semântica**

A semântica nos permite descrever mais precisamente o nosso conteúdo, agora um bloco de texto não é apenas uma div, agora é um article e tem mais significado assim. E temos vários elementos para ressignificar as divs:

**<section>**

Representa uma seção genérica de conteúdo quando não houver um elemento mais específico para isso.

**<header>**

É o cabeçalho da página ou de uma seção da página e normalmente contém logotipos, menus, campos de busca.

**<article>**

Representa um conteúdo independente e de maior relevância dentro de uma página, como um post de blog, uma notícia em uma barra lateral ou um bloco de comentários. Um article pode conter outros elementos, como header, cabeçalhos, parágrafos e imagens.

**<aside>**

É uma seção que engloba conteúdos relacionados ao conteúdo principal, como artigos relacionados, biografia do autor e publicidade. Normalmente são representadas como barras laterais.

**<footer>**

Esse elemento representa o rodapé do conteúdo ou de parte dele, pois ele é aceito dentro de vários elementos, como article e section e até do body. Exemplos de conteúdo de um <footer> são informações de autor e links relacionados.

**<h1>-<h6>**

Eles não foram criados na versão 5 do HTML e nem são específicos para semântica, mas servem para esse propósito. São utilizados para marcar a importância dos títulos, sendo <h1> o mais importante e <h6> o menos. Uma dica: use apenas um <h1> por página, pois ele representa o objetivo da sua página.

**<p>**

Para textos maiores e mais densos usamos o elemento p que representa um parágrafo, mas ele não suporta apenas texto, podemos adicionar imagens, código, vídeos e vários outros tipos de conteúdo dentro dele.

**<a>**

Significa anchor/âncora, ele representa um hyperlink, é ele que interliga vários conteúdos e páginas na web.

O elemento a tem vários atributos, mas vamos focar em dois, o href e o target.

O **href** representa o hyperlink para onde sua âncora aponta, pode ser uma página do seu ou de outro site, um e-mail e até mesmo um telefone, os dois últimos precisam dos prefixos mailto: e tel:, respectivamente.

O target neste momento vai servir para nos ajudar a abrir nossos links em outra aba do navegador usando o valor \_blank.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**<img>**

A web também é feita de imagens e para representá-las temos o elemento <img>, ele é um daqueles elementos sem tag de fechamento. Possui apenas 2 atributos próprios, o src e o alt.

O src é obrigatório e guarda o caminho para a imagem que você quer mostrar na página.

O alt não é obrigatório, mas é altamente recomendado por melhorar a acessibilidade, ele mostra a descrição da imagem caso ela não carregue e leitores de tela usam esse atributo para descrever a imagem para o usuário saber o que ela significa.

**Listas**

Os últimos elementos que veremos neste módulo são os relacionados a listas: <ul>, <ol> e <li>.

Listas servem para agrupar uma coleção de itens, como uma lista de ingredientes ou uma lista com contatos.

**<ul>**

O elemento ul cria uma lista não ordenada, onde a ordem dos elementos não é importante, e é representada com pontos, círculos ou quadrados.

**<ol>**

O <ol> serve para criar lista ordenadas, nessas a ordem importa, portanto elas são representadas com números, algarismos romanos ou letras.

**<li>**

E o elemento li é um item dentro de uma dessas listas. Um <li> pode conter vários tipos de conteúdo, como parágrafos, imagens e até outras listas.

**Conceitos básicos de CSS 3**

Definição e seletores

Após a criação do HTML a necessidade de formatar as páginas ficou evidente, assim, em 1996, foi criada a linguagem de estilo que conhecemos por CSS.

A sintaxe é bem simples e pode ser explicada com a frase "você cria regras de estilo para elementos ou grupos de elementos".

Uma regra CSS é representada por um seletor ou um grupo de seletores (que são elementos HTML) e declarações, que alteram as características dos seletores informados.

Diagrama

Descrição gerada automaticamente

Percebam que podemos colocar vários seletores em uma regra separando-os por vírgula.

**ID x Classe**

Se criamos uma regra que altera um elemento HTML diretamente, todos os elementos ficarão com aquela aparência, e normalmente temos sites mais complexos que precisam de várias regras diferentes para elementos iguais, é aí que entram os **IDs e Classes**.

* **ID**: é representado pelo símbolo # (hash) seguido de um nome para esse ID.
* **Classe**: é representado pelo ponto seguido de um nome para essa classe.

E a diferença mais importante entre eles é a forma como devem ser usados: o ID só pode ser usado uma vez em uma página HTML enquanto a classe não tem restrições.

É possível adicionar CSS de duas formas, com o elemento style, e assim suas regras ficarão no arquivo HTML, ou podemos criar um arquivo CSS e adicioná-lo na página através do elemento link, e é essa forma que usaremos.

**Box-model**

Diagrama

Descrição gerada automaticamente

Cada elemento HTML é representado como uma caixa retangular, isso é o box-model. E com CSS nós alteramos a aparência dessa caixa (largura, altura, cor de fundo, etc.). Essa caixa é composta por 4 áreas: o conteúdo, o padding, a borda e a margem.

* **As margens (margin):** são espaçamentos entre elementos;
* **As bordas (border):** circundam o padding e content;
* **O padding**: é um espaçamento entre as bordas e o conteúdo, a diferença para as margens é que declarações de imagem de fundo funcionam nele;
* **O conteúdo (content):** é o que o seu bloco representa, um texto, uma imagem, um vídeo;

**Padding e Margin**

Anteriormente usamos o padding e o margin da forma mais básica, com apenas um valor, é possível atribuir tamanhos diferentes para cada lado do box.

**Border**

A propriedade border pode ter 3 valores: a largura, a cor e o estilo, mas existem algumas particularidades nisso.

A largura pode ser usada com várias unidades, como px, em e mm. A cor pode ser atribuída pelo nome ou por um código hexadecimal, assim como fizemos com o background, e o estilo é representada por palavras-chave, vamos ver algumas delas:

* **solid**: mostra uma borda simples e reta;
* **dotted**: são bolinhas com um pequeno espaçamento entre elas;
* **dashed**: forma uma linha tracejada.

**Border-radius**

Permite arredondar os cantos de um elemento. Podemos usar várias unidades, mas as mais comuns são os pixels e a porcentagem.

Colocando apenas um valor mudamos todos os cantos do elemento, mas seguindo aquela mesma ordem que vimos no padding e margin - topo, direita, inferior e esquerda - conseguimos alterar cada canto separadamente.

**font-family**

Com o font-family podemos alterar a fonte dos nossos textos, como uma fonte da internet ou uma que esteja instalada no nosso computador.

**font-size**

Permite mudar o tamanho do texto utilizando unidades de medida como o pixel.

**font-style**

Usamos o font-style para tornar um texto itálico (italic), na maioria das vezes você usará apenas o valor italic para ele, mas se precisar tirar o itálico de um texto você pode usar o valor normal.

**font-weight**

Usamos o font-weight para tornar um texto negrito (bold).

**text-transform**

uppercase: todo texto em caixa alta (maiúscula).

lowercase: todo texto em caixa baixa (minúscula).

capitalize: primeira letra de cada palavra em maiúscula.