**1 . APLICAÇÃO CLIENTE-SERVIDOR**

**1.1 O que é**

“O modelo de comunicação utilizado na internet é baseado em duas atribuições principais para os elementos envolvidos na comunicação: cliente e servidor. O servidor deve estar sempre disponível, à espera da iniciativa do cliente. O cliente aciona o servidor sempre que precisa fazer a comunicação. Esse princípio difere essencialmente dos modelos de comunicação nos quais qualquer participante da rede pode iniciar a comunicação. No modelo cliente-servidor, comunicação sempre se inicia no cliente.

O servidor é um software que mantém uma porta de comunicação aberta à espera do cliente. Sua localização, seja pelo seu endereço ou nome, deve ser conhecida por todos os clientes que querem acessá-lo. Um servidor pode receber um grande número de solicitações simultâneas de clientes, por isso, normalmente, executa em um computador de alto desempenho são geralmente denominados também servidores, apesar de o papel de servidor ser desempenhado por um software. Isso não impede que um dispositivo IoT execute a função de servidor, apesar do hardware limitado, recebendo solicitações de clientes.

O cliente é também um software, normalmente acionado por um usuário, razão pela qual é comum que tenha uma interface gráfica amigável. Um navegador web como Google Chrome, Mozilla Firefox ou Internet Explorer é um exemplo de cliente. Cabe ao cliente iniciar a comunicação com o servidor, seja acionada diretamente pelo usuário ou de forma automática, em resposta a um evento ou uma ação externa. Um dispositivo de IoT que coleta uma temperatura poderia se conectar a um servidor de bancos de dados periodicamente para enviar as informações de temperatura coletadas, ou quando a temperatura extrapolasse parâmetros previamente definidos.

Diversos servidores podem ser úteis em projetos de IoT, incluindo servidores de arquivos, servidores web, servidores de banco de dados, servidores de mensagens ou e-mails, entre outros. Vários deles serão citados ao longo deste livro.” (MINARDI, MARIANO, 2019).

**1.2 Linguagens do lado cliente**

“As páginas HTML podem ter seu comportamento alterado no front-end por scripts (aqui representados por arquivos JavaScript e suas bibliotecas, como jQuery). Apesar desses scripts serem carregados pelo servidor, eles podem interagir através de interações assíncronas (como AJAX).” (MINARDI, MARIANO, 2019).

“As linguagens client-side são linguagens onde apenas o seu NAVEGADOR vai entender. Quem vai processar essa linguagem não é o servidor, mas o seu browser (Chrome, IE, Firefox, etc...). Significa "lado do cliente", ou seja, aplicações que rodam no computador do usuário sem necessidade de processamento de seu servidor (ou host) para efetuar determinada tarefa.

Basicamente, ao se falar de aplicações client-side na web, estamos falando de JavaScript, e AJAX (Asynchronous Javascript And XML).

*Existem vantagens e desvantagens ao utilizar o JavaScript e AJAX.*

A principal vantagem está na possibilidade de você economizar *bandwidth* (largura de banda), e dar ao usuário uma resposta mais rápida de sua aplicação por não haver processamento externo.

Outra vantagem ao utilizar, agora o AJAX, seria o apelo visual de sua aplicação e rapidez de resposta. O que o AJAX faz é o processamento externo (server-side) parecendo ser interno (client-side). O usuário não percebe que houve um novo carregamento de página, pois ele busca informações no servidor e mostra rapidamente em um local específico da página através do JavaScript.

A principal desvantagem do JavaScript atualmente é que o usuário pode desativá-lo em seu navegador. Se a sua aplicação basear-se exclusivamente em JavaScript, nesse caso, ela simplesmente não vai funcionar.” (GIGASYSTEMS , 2022).

**1.3 Linguagens do lado servidor**

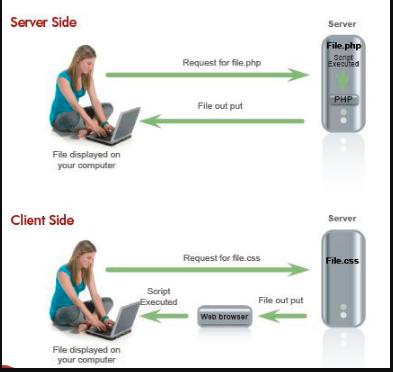
“As linguagens back-end (neste livro será utilizado a linguagem PHP) fica responsáveis por controlar as requisições do usuário, gerar acesso aos bancos de dados e retornar às páginas HTML devidamente formatadas com folhas em cascata.” (MINARDI, MARIANO, 2019).

“As linguagens server-side são linguagens que o SERVIDOR (lado do servidor) entende, ou seja, aplicações que rodam no servidor. Isso quer dizer que você vai escrever um código onde o servidor vai processá-lo e então vai mandar para o seu navegador a resposta. Por exemplo: se criarmos um script em linguagem back-end (PHP, Asp, Java, Ruby, etc) que apenas calcula a soma de 1 + 1, será o SERVIDOR (ou back) que calculará este resultado. Se fizermos esse cálculo em alguma linguagem front-end, como o JavaScript, quem calculará essa conta é o BROWSER do usuário. Por isso o termo client (ou front).” (GIGASYSTEMS , 2022).

**1. 4 Como os lados se comunicam**

“No dogma central do desenvolvimento Web, as linguagens back end ficam responsáveis por controlar as requisições do usuário, gerar acesso aos bancos de dados e retornar às páginas HTML devidamente formatadas com folhas em cascatas. Essas páginas podem interagir com o ambiente back- end por meio das requisições de páginas e dos envios de formulários. As páginas HTML podem ter seu comportamento alterado no front-end por scripts. Apesar desses scripts serem carregados pelo servidor, eles podem interagir através de interações assíncronas.” (MINARDI, MARIANO, 2019).

**1.4.1 Exemplificar com imagem**



Fonte: Gigasystems, 2022.

**1.5 Software Navegador (*Browser*)**

**1.6 Definição**

“Mas, o que são navegadores? Também conhecido como web browser ou simplesmente browser, são uma espécie de ponte entre usuários e o conteúdo virtual da internet. Basicamente, os navegadores transformam as páginas codificadas em HTML, para visualização compreensível para o usuário.” (Techtudo, 2022).

“Também conhecido como browser (em inglÊs), é um programa destinado a visualizar documentos desenvolvidos com linguagem de marcação, ou, em sentido mais restrito, sites na internet.” (SILVA, 2008).

**1.7 Exemplos**

“Firefox, Internet Explorer, Google Chrome, Safari e Opera são alguns dos navegadores mais utilizados atualmente.” (SILVA, 2008).

**1.8 Linguagem de marcação HTML**

**1.9 Definição**

“HTML é a sigla em inglês para HypertText Markup Language, que, em português, significa linguagem para marcação de hipertexto. Podemos resumir hipertexto como todo o conteúdo inserido em um documento para web e que tem como principal característica a possibilidade de se interligar a outros documentos da web. O que torna possível a construção de hipertextos são os links, presentes nas páginas dos sites que estamos acostumados a visitar quando entramos na internet.” (SILVA, 2019)

**1.10 HTML 5**

“O consórcio da Word Wide Web estendeu a patente do HTML5 - que é a última versão do standard e a quinta revisão no total para um esquema em pleno funcionamento em 14-02-2011.

O HTML5 permite criar aplicações web que podem interagir com os dados locais e com servidores mais facilmente do que nunca; adicionalmente, fornece suporte para conteúdos multimédia ricos. Inicialmente, o consórcio da World Wide Web desenvolveu o HTML 5 para resolver questões de compatibilidade que vitimou a versão anterior, o HTML 4. A diferença maior entre todas as versões anteriores e o HTML 5 é que esta não requer APis e plug-ins de proprietário, que eram responsáveis pela criação das questões de compatibilidade. Pelo contrário, o HTML torna o carregamento das páginas mais fácil e assegura compatibilidade entre todos os browsers oferecendo interface comum para todos.

O html5 oferece muitas novas características que podem mudar por completo a forma como os utilizadores interagem com a web. Essas características incluem mas não estão limitadas a:

\* Novos atributos;

\* Novas regras de Parsing;

\*Falta de elementos e atributos redundantes;

\*Regras detalhadas para Parsing;

\* Edição offline

\* A habilidade concentra-se apenas em novos elementos, tags, e entidades encontradas no HTML5. (PRESCOTT, 2015).

**1.10.1 Semântica da linguagem**

“Existem cento e oito elementos previstos nas especificações do W3C para a HTML5:

* html

É o elemento raiz do documento que engloba as seções HEAD e BODY.

* head

Se destina a marcar a seção HEAD do documento.

* title

Se destina a marcar o título do documento.

* meta

Se destina a fornecer informações adicionais sobre o documento, tais como a codificação de caracteres, o nome do autor do documento e outras.

* body

Se destina a marcar a seção body do documento. Tudo que está dentro dessa seção será renderizado pelo agente de usuário (por exemplo: o navegador).

* h1-h6

Eles são do tipo nível de bloco e destinam-se a marcar os textos de títulos e subtítulos dos conteúdos do documento.

* p

Ele é do tipo nível de bloco e se destina a marcar parágrafos.

* a

Esse elemento é do inline e se destina a marcar links.” (SILVA, 2015)

“O HTML5 foi criado em prol de uma web mais semântica. Isso significa que teremos um código padronizado e organizado. Para isso ser possível foram criadas diversas novas tags enquanto outras foram excluídas.

Cada uma delas é definida por uma nova tag. São elas:

<header>: cabeçalho de uma seção. Como vemos, cabeçalhos podem ser usados no documento ou em seções específicas com a seção <article>: cria uma seção de conteúdo relacionado. É qualquer conteúdo que tenha alguma relação com o conteúdo principal. Pode ser um menu secundário, banners, links ou imagens de parceiros, link tipo "indique" ou "cadastre-se aqui", anúncios em geral, entre vários outros exemplos;

<section>: cria uma seção de conteúdo. É uma seção "genérica" e pode agrupar conteúdos diversos. Pode ser utilizada inclusive para agrupar outras seções que pertençam a um mesmo conteúdo. Exemplo: artigo, cabeçalho do artigo e rodapé do artigo;

<article>: criar uma seção de artigo. É o conteúdo principal do documento. Pode possuir textos, imagens, vídeos etc;

<nav>: cria uma seção de menu. É o menu principal do site, que conterá links para acessar as outras páginas e/ ou arquivos. Não deve ser usada para agrupar qualquer link, mas somente um menu. Um menu é composto por links (em modo texto ou imagem) que darão acesso às várias partes do site. Pode ser utilizada também para criar seções de menus secundários;

<footer>: cria uma seção de rodapé. O rodapé pode ser do documento ou de determinadas seções (tal como o rodapé de <article>). O rodapé da página possui informações geralmente visíveis a todas as páginas do site, tais como telefones, e-mails e endereços, avisos de proteção e de direitos autorais etc.” (FERREIRA, 2013).

**1.10.2 Acessibilidade**

“Essa funcionalidade visa a facilitar o acesso ao conteúdo independentemente do dispositivo ou das necessidades especiais do usuário. Isso não significa que uma funcionalidade deva ser descartada se um grupo de usuários não tiver acesso a ela, mas se devem prever meios alternativos de acesso. É o caso do elemento img que não deve ser banido porque usuários cegos não conseguem ver imagens. Usa-se o atributo alt para fornecer um meio alternativo de acesso a imagem para aqueles usuários.” (SILVA, 2019).

**1.11 Definições de tag, atributo e valor**

“Todo o documento terá esse conjunto de instruções que foram o corpo básico da maioria das páginas.

Tags são palavras-chave cercadas por colchete angulares, por exemplo, <html>.

Tags geralmente se apresentam em pares, como <body> e </body>, sendo que:

- A primeira tag em um par é a tag inicial ou de abertura, e a segunda é a tag final de fechamento.

- Existem tags especiais, chamadas abertas ou vazias. Elas não possuem fechamento. Por exemplo: <br>, <a>.

- A tag final é escrita como a marca inicial, com a adição de uma barra antes do nome da tag, como em </body>.” (SILVA, 2019).

“Os elementos HTML podem ter atributos que fornecem informações adicionais sobre um elemento. Eles são sempre especificados em uma tag inicial e devem ser informados em pares com nome e valor.

Exemplo: <a href="http://www.sp.senac.br">Acesse o lin</a>

Em que href é o nome e "http://www.sp.senac.br" é o valor. Note que os valores dos atributos devem sempre estar entre aspas. Veja que, após o atributo, você deve colocar um sinal de igual "=".” (GOMES, MARTELLI, 2016).

**1.12 Estrutura básica de uma página**

“<!DOCTYPE html>

<html>

<head></head>

<body></body>

</html>”. (GOMES, MARTELLI, 2016).

**1.12.1 Tag meta**

“<meta>: Essa tag contém elementos que são usados para especificar o comportamento e a descrição da página, as palavras-chave, o autor do documento, a última modificação, entre outros metadados. Estes podem ser utilizados pelos navegadores (como exibir conteúdo ou página recuperada), pelos motores de busca (palavras-chave), ou por outros serviços da web.”(GOMES, MARTELLI, 2016).

**2 EDITORES DE TEXTO PARA FRONT-END**

**2.1 Exemplos de alguns editores e seus recursos (visão geral apenas)**

“Para programar com HTML5 podemos usar até um simples bloco de notas. Mas existem softwares que facilitam a programação, oferecendo recursos tais como a numeração de linhas, uso de cores diferentes na sintaxe da linguagem etc. Recomendamos o uso de dois programas bem famosos.

Notepad++;

Aptana Studio Communit Edition;”. (GOMES, MARTELLI, 2016).

**3. CODIFICAÇÃO DE CARACTERES**

**3.1 O que é e sua importância**

“Conteúdo é composto de uma sequência de caracteres. Caracteres representam letras do alfabeto, pontuação, etc. Conteúdos são armazenados em um computador como uma sequência de bytes, que são valores numéricos. Em alguns casos um simples caractere é representado por mais de um byte. Tal como os códigos usados em espionagem, a maneira como uma sequência de bytes é convertida em caracteres depende do formato como o conteúdo foi codificado. Nesse contexto tal *formato* é denominado codificação de caracteres.” (W3C, 2016).

**3.2 UTF-8**

“A codificação baseada em Unicode, tal como UTF-8, oferece suporte para vários idiomas e assim sendo admite páginas e formulários em qualquer combinação de idiomas. Isso dispensa a necessidade de se criar uma lógica no servidor capaz de determinar a codificação para cada página servida ou para cada conjunto de dados recebidos de um formulário. Fica significativamente reduzida a complexidade própria do gerenciamento de um site ou aplicação multi-idiomas.

A codificação Unicode, mais do que qualquer outra codificação, permite que muito mais idiomas sejam usados em uma única página.

Oferecer suporte para uma determinada codificação, inclusive Unicode, não significa necessariamente que o agente de usuário apresentará o texto de forma correta. Muitos scripts, tais como árabe e sânscrito necessitam de regras adicionais para transformar a sequência de caracteres gravadas na memória em uma sequência apropriada de glifos para apresentação correta.

Nos dias atuais praticamente não existem barreiras para o uso do Unicode. Em janeiro de 2012 o Google informou que mais de 60% da Web está usando UTF-8. Se forem consideradas as páginas codificadas em ASCII (lembrando que ASCII é um subconjunto de UTF-8) aquele percentual sobe para 80%.” (W3C, 2016).

**Referências:**

MINARDI, Raquel; MARIANO, Diego. **Introdução à programação Web para Bioinformática: HTML, CSS PHP & JavaScript**. Belo Horizonte, 2017.

GIGASYSTEMS. **Client-side e Server-side**. 2022, <https://www.gigasystems.com.br/artigo/60/client-side-e-server-side>

TECHTUDO. **Saiba o que é um navegador e um sistema operacional.** 2022, <https://www.techtudo.com.br/noticias/2011/08/saiba-o-que-e-um-navegador-e-um-sistema-operacional.ghtml>.

SILVA, Maurício. **Criando sites com HTML. Sites de alta qualidade com HTML E CSS**. São Paulo: Novatec Editora, 2008.

SILVA, Maurício. **A Linguagem de marcação que revolucionou a Web**. São Paulo: Novatec, 2019.

PRESCOTT, Preston. **HTML5**. São Paulo: Babelcube, 2015.

SILVA, Maurício. **Fundamentos de HTML5 e CSS3**. Novatec: São Paulo, 2015.

FERREIRA, Silvio. **Guia prático de HTML**. Universo dos livros Editora: São Paulo, 2013.

MARTELLI, Richard; GOMES, Ana Laura. **HTML5 e CSS3**. Senac Editoria: São Paulo, 2016

W3C. **Escolha e uso de codificação de caracteres**. 2016, <https://www.w3.org/International/questions/qa-choosing-encodings.pt-br>