



**INSTITUTO FEDERAL**  
Sul de Minas Gerais

**2018**

MEDIOTEC - REDE - E TEC

# APOSTILA DE **Informática** Programação para Web III

Prof. : Elton Adriano Ribeiro da Silva



## Palavra do professor-autor

Prezado Aluno!

O objetivo desta apostila é auxiliá-lo da melhor maneira a compreender e a utilizar os principais recursos práticos e teóricos da disciplina Programação para WEB III. Por meio deste estudo, você poderá absorver e aumentar seu conhecimento através de pesquisas, leituras, práticas e discussões.

## Apresentação da disciplina

No andamento desta disciplina, teremos várias atividades relacionadas a cada aula estudada através de nosso Ambiente Virtual de Aprendizagem. Aliar a teoria com a prática é a melhor maneira de atingir o conhecimento. Busque participar ativamente e sempre envie suas dúvidas ao Professor Formador e ao Professor Mediador.

Discutiremos ao longo da disciplina:

- Funcionamento dos computadores pessoais.
- Principais dispositivos e componentes de um computador.
- Montagem de computadores.
- Possíveis problemas de drivers e dispositivos.
- Identificação e verificação de tensões de alimentação de um computador.
- Testes de funcionalidades de dispositivos.
- Configuração de sistemas operacionais.
- Backup.
- Segurança de dados.
- Periféricos.
- Conexão física entre dispositivos.

Entre outros assuntos pertinentes a nossa disciplina.

Minha função será orientá-lo, apoiá-lo e acompanhá-lo em seu processo de aprendizagem, ajudando-o em suas necessidades e dúvidas.

Sucesso!

# Sumário

	<b>1</b>
<b>Palavra do professor-autor</b>	<b>2</b>
<b>Apresentação da disciplina</b>	<b>2</b>
<b>Aula 1 – World Wide Web (WWW)</b>	<b>5</b>
1.1. O que é	5
1.2. O que é Hipertexto	5
<b>Aula 2 – O Funcionamento dos Sistemas Web – HTML</b>	<b>7</b>
2.1. – O HTML	7
2.2. – Estrutura básica de uma página HTML	7
<b>Aula 3 – Programação Cliente x Programação Servidor</b>	<b>11</b>
3.1. – Programação Cliente	11
3.2. – Programação Servidor	11
<b>Aula 4 – Arquitetura Web</b>	<b>13</b>
4.1. – Arquitetura da Informação	13
4.2. - Três Tipos de Arquitetura de Aplicação Web	13
4.2.1. - Servidor HTML	14
4.2.2. - “Widgets” de geração JS (AJAX)	14
4.2.3. - Página única orientada para os serviços de aplicações Web (Web 2.0 e aplicações HTML5)	14
4.3. - Programação para Web	14
<b>Aula 5 – A Linguagem de Programação PHP</b>	<b>17</b>
5.1. – PHP - definição	17
5.2. - Servidor de Teste	17
5.3. – IDE	20
5.4. - Testando o Ambiente	21
<b>Aula 6 – Sintaxe Básica e Estudo das Variáveis</b>	<b>23</b>
6.1 - Página PHP	23
6.2. - Sintaxe Básica	24
6.3. – Variáveis	25
6.3.1 - Nome das Variáveis	25
6.3.2 – Declaração de Variáveis	25
6.3.3 – Escopo de uma variável	25
6.4. – Strings	27
6.4.1 – Operação de Concatenação	27
6.4.2 – Função strlen ( )	27
6.4.3 - Função strpos( )	27
6.5. – Operadores	28

6.5.1 – Operadores Aritméticos	28
6.5.2 – Operadores de Atribuição	28
6.5.3 – Operadores de Incremento e Decremento	28
6.5.4 - Operadores de Comparação	28
6.5.5 - Operadores Lógicos	29
<b>6.6. – If...Else</b>	29
<b>6.7. – Switch</b>	30
<b>Aula 7 – Estruturas de Repetição</b>	<b>32</b>
7.1. – Estruturas de Repetição	32
7.1.1 – While	32
7.1.2 - Do...While	32
7.1.3 – For	33
<b>Aula 8 – Funções em PHP</b>	<b>34</b>
8.1 – Funções	34
8.2 – Adicionando Parâmetros	34
8.3 – Retornando Valores	35
8.4 – Formulários	35
8.4.1 – Método GET	35
8.4.2 – Método POST	36
<b>Aula 9 - Introdução à Programação em Linguagem Orientada a Objetos</b>	<b>37</b>
9.1. - Programação Orientada a Objetos (POO)	37
<b>Referências</b>	<b>39</b>

## Aula 1 – World Wide Web (WWW)

Objetivo: Conceituar World Wide Web

### 1.1. O que é

Segundo Bert & Sierra (2009), World Wide Web ou WWW é um sistema de documentos dispostos na Internet que permitem o acesso às informações apresentadas no formato de hipertexto. Para ter acesso a tais informações, pode-se usar um programa de computador chamado navegador. Os navegadores mais famosos são: [Internet Explorer](#), [Mozilla Firefox](#), [Google Chrome](#) e [Safari](#).

De acordo com o site Tech Tudo, a ideia de World Wide Web surgiu em 1980, na Suíça. O precursor foi o britânico Tim Berners-Lee. Um computador NeXTcube foi usado por Berners-Lee como primeiro servidor web e também para escrever o primeiro navegador, o WorldWideWeb, em 1990.

Em 6 de agosto de 1991, Tim Berners-Lee postou um resumo sobre todas as suas ideias e projetos no grupo de notícias de nome alt.hypertext. Essa data marca a estreia oficial da Web como um serviço publicado na Internet.

A partir de então, com o avanço das tecnologias de internet, a web também pôde se desenvolver grandiosamente, aumentando significativamente a quantidade de conteúdo disponível e também a interação entre usuários e páginas. A chamada Web 2.0, desenvolvida a partir do início dos anos 2000, é a fase na qual o conteúdo da web é oferecido também pelos seus usuários em blogs, vlogs, fotologs e redes sociais.

Atualmente, podemos dizer que vivemos na Web 3.0 (também conhecida como web semântica), que é a fase em que os sistemas de busca estão cada vez mais inteligentes e o conteúdo disponível na teia mundial torna-se cada vez mais personalizado.

### 1.2. O que é Hipertexto

Marcuschi diz que o hipertexto é um conceito associado às tecnologias da informação e que faz referência à escrita eletrônica. Desde sua origem, o hipertexto vem mudando a noção tradicional de autoria, uma vez que ele contempla diversos textos. Trata-se, portanto, de uma espécie de obra coletiva, ou seja, apresenta textos dentro de outros, formando, assim, uma grande rede de informações interativas. Nesse sentido, sua maior diferença é justamente a forma de escrita e leitura. Assim, num texto tradicional, a leitura segue uma linearidade, enquanto no hipertexto ela é não linear. Conforme demonstrado na figura abaixo:

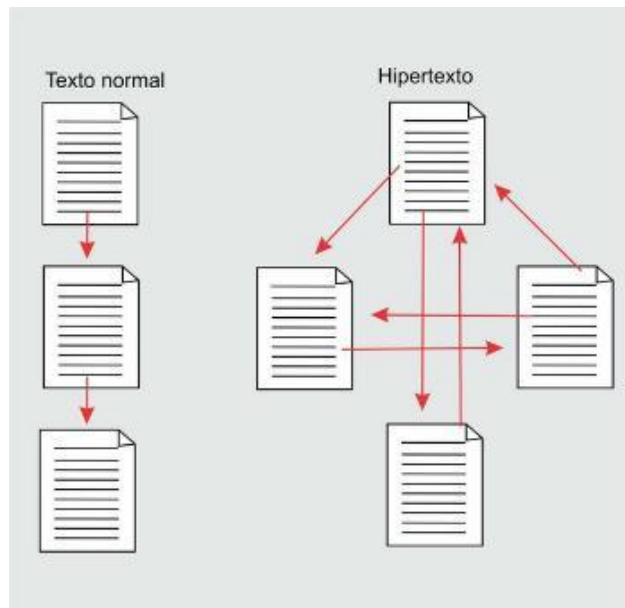


Figura 1 - A diferença entre um texto normal e o hipertexto. Fonte: TechTudo

Um forte exemplo de hipertexto são os artigos na internet. No corpo do texto eles apresentam diversos links ("ligação", em inglês) ou hiperlinks nas palavras ou nos assuntos que estejam relacionados. Isso permite que o próprio leitor tenha uma posição mais ativa, escolhendo as informações que prefere acessar. Além dos artigos da internet, um livro de contos, dicionários e enciclopédias são considerados exemplos de hipertextos. A informação contida neles proporciona um caráter não linear, em que o leitor também pode selecionar as informações e os caminhos de leitura que preferir. Sendo assim, a leitura de hipertexto é realizada por associações.



## **Aula 2 – O Funcionamento dos Sistemas Web – HTML**

Objetivo: Entender o funcionamento dos sistemas web

### **2.1. – O HTML**

De acordo com o site Info Escola, O HTML é uma linguagem de marcação utilizada para desenvolvimento de sites.

Essas linguagens são constituídas de códigos que delimitam conteúdos específicos, segundo uma sintaxe própria. O HTML tem códigos para criar páginas na web e são esses códigos que definem o tipo de letra, qual o tamanho, cor, espaçamento, e vários outros aspectos do site.

No início era muito complicado aprender HTML, pois eram muitos comandos para fazer algo simples. A cada nova versão, o HTML fica mais fácil de utilizar e adquire mais funções. Atualmente, qualquer pessoa pode acessar a internet e aprender a construir um site básico em questão de horas, seguindo os passos de tutoriais e aprendendo as funções de cada código.

O HTML foi à primeira linguagem de nível mundial, porém, não é a única. Existem muitas outras linguagens destinadas à criação de páginas da web, embora o HTML ainda prevaleça. Atualmente já é possível integrar várias linguagens na mesma página da web, possibilitando usar duas ou mais linguagens no mesmo site.

Para criar e editar códigos em HTML é necessário qualquer editor de texto comum, como bloco de notas. Para testar os códigos, basta salvar o arquivo em formato HTML e executar. Para o teste, é necessário ter um navegador configurado como padrão. Não é necessária internet, pois o arquivo com os códigos está na máquina onde está sendo executado.

### **2.2. – Estrutura básica de uma página HTML**

Ao acessar uma página web através de um navegador, ele é capaz de interpretar o código HTML e renderizá-lo de forma compreensível para o usuário final, exibindo textos, botões, etc., com as configurações definidas por meio das diversas tags que essa linguagem dispõe.

Atualmente a HTML encontra-se na versão 5 e é padronizada pelo W3C (World Wide Web Consortium), uma organização internacional responsável por estabelecer padrões para a internet.

A estrutura básica de uma página HTML pode ser vista na imagem a seguir, na qual podemos ver as principais tags que são necessárias para que o documento seja interpretado corretamente pelos browsers.

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04     <meta charset="UTF-8"/>
05     <title>Document</title>
06 </head>
07 <body>
08     <!-- Conteúdo -->
09 </body>
10 </html>
```

Figura 2 – Estrutura básica de uma página HTML. Fonte: DevMedia.

**Linha 1:** a instrução DOCTYPE deve ser sempre a primeira a aparecer em uma página HTML e indica para o browser qual versão da linguagem está sendo usada. Nesse caso, estamos trabalhando com a HTML5, versão na qual a declaração do DOCTYPE é bastante simples, como podemos ver na listagem.

**Linhas 2 e 10:** abertura e fechamento da tag html, que delimita o documento. Sendo assim, as demais tags da página devem estar nesse espaço.

**Linhas 3 e 6:** abertura e fechamento da tag head, que define o cabeçalho do documento. O conteúdo nesse espaço não é visível no browser, mas contém instruções sobre seu conteúdo e comportamento. Dentro dessa tag, por exemplo, podem ser inseridas folhas de estilo e scripts.

**Linha 4:** a tag meta, nesse caso, especifica qual conjunto de caracteres (character set ou charset) será usado para renderizar o texto da página. O UTF-8 contém todos os caracteres dos padrões Unicode e ASCII, sendo, portanto, o mais utilizado em páginas web. A mesma tag meta, porém com outros atributos, pode ser utilizada para outros fins, como na SEO (Search Engine Optimization).

**Linha 5:** a tag title define o título da página, aquele que aparece na janela/aba do navegador.

**Linhas 7 e 9:** abertura e fechamento da tag body, marcando o espaço no qual deve estar contido o conteúdo visual da página. As demais tags que representam texto, botões, etc., devem ser adicionadas nesse intervalo.

**Linha 8:** nessa linha podemos observar a sintaxe para adição de comentários em HTML. Esse trecho não é processado pelo browser.

Para desenvolver páginas com HTML, precisamos basicamente de um editor de texto, como o Bloco de Notas, do Windows, Nano e Emacs, no Linux, entre outros. Há, ainda, editores com opções avançadas, como recursos de syntax highlight e autocomplete, como Sublime Text, Atom, Brackets e Visual Studio Code, que podem ser usados para editar documentos HTML. Independente do editor



utilizado, podemos simplesmente copiar o código da Figura 2 para um deles e salvar o arquivo com extensão html. Em seguida, podemos abrir esse arquivo em um browser. Outro aspecto importante da linguagem é que ela é case insensitive, ou seja, não leva em consideração a diferença entre letras maiúsculas e minúsculas. No entanto, o uso apenas de letras minúsculas tem sido utilizado como padrão pelos desenvolvedores.



## Aula 3 – Programação Cliente x Programação Servidor

Objetivo: Diferenciar Programação Cliente x Servidor.

### 3.1. – Programação Cliente

No lado do cliente, ou seja, do usuário, a aplicação roda diretamente em seu computador. O Lado do cliente dá a resposta em tempo real para qualquer interação que é feita no website.

Por exemplo: A maioria dos websites tem um formulário para ser preenchido e dentro desse formulário é preciso ter validações para verificar se o campo não ficou em branco ou se foram digitados caracteres inválidos.

A maioria dessas validações é feita em Java script (client-side), ou seja, assim que o usuário interage com o formulário, ele já responde de imediato, seja por um botão ou por um comportamento, como mudança de foco numa caixa de texto ou até mesmo enquanto o usuário está digitando algo. O navegador não precisa buscar o código no servidor, pois este já foi baixado para computador do usuário.

Linguagens Client-Side:

- HTML
- CSS
- Java script

A imagem abaixo retrata o fluxo da programação cliente:

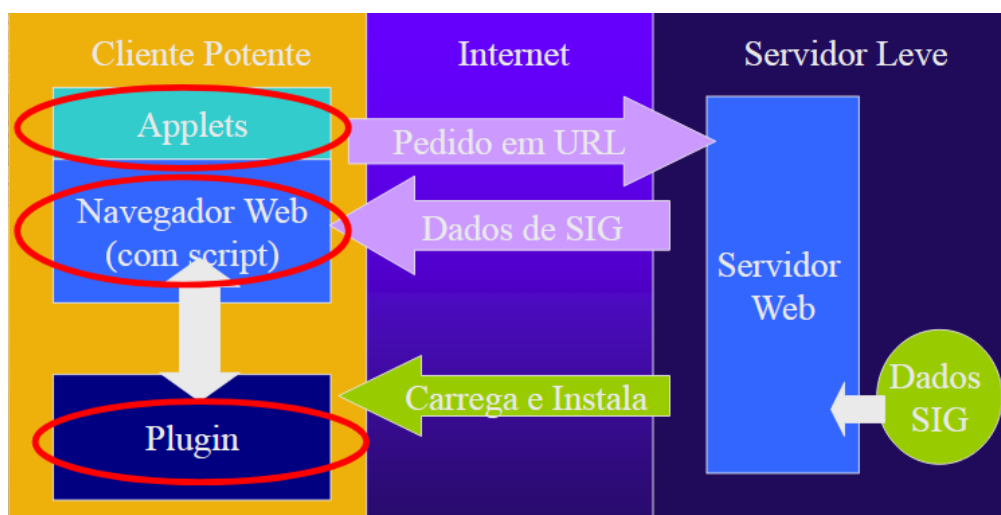


Figura 3 – Programação Cliente. Fonte: SlidPlayer

### 3.2. – Programação Servidor

No lado do servidor, rodam as aplicações necessárias para um website funcionar, como banco de dados e a linguagem que o programador esteja usando para a criação do software, como, por

exemplo, o PHP. Toda vez que o usuário abrir uma página que tenha um código PHP, o navegador vai executar o comando direto do servidor.

Como exemplo pode-se citar o acesso a um banco de dados qualquer, não é possível usar uma linguagem client-side, como o Java script. Teremos de usar, automaticamente, uma linguagem server-side, como PHP.

Linguagens server-side:

- Banco de dados como SQL, MySQL, Oracle...
- PHP
- ASP
- ASP.NET

A imagem abaixo retrata o fluxo da programação servidor:

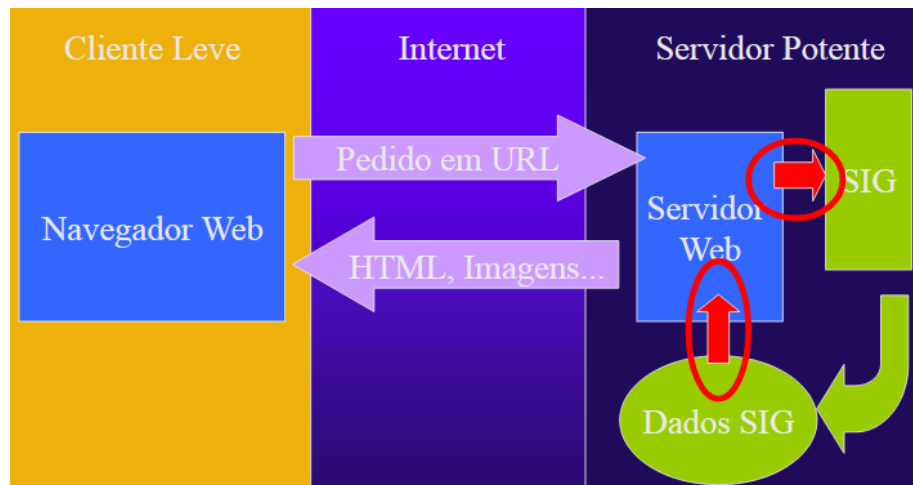


Figura 4 – Programação Servidor. Fonte: SlidPlayer

## Aula 4 – Arquitetura Web

Objetivo: Conceituar Arquitetura web

### 4.1. – Arquitetura da Informação

A arquitetura web tem por objetivo criar as estruturas de organização da informação de um website para que o usuário consiga compreendê-lo com facilidade. Na web, a arquitetura de informação cuida de projetar a estrutura, o esqueleto de um website sobre o qual todas as demais partes irão se apoiar.

West (2001) cita que “arquitetura de informação é a prática de projetar a infraestrutura de um website, especialmente a sua navegação”, e Shiple (2001) afirma que “arquitetura de informação é a fundação para um ótimo web design. Ela é o esquema do website sobre o qual todos os outros aspectos são construídos – forma, função, metáfora, navegação e interface, interação e design visual”. Esse esquema, essa infraestrutura, tem um propósito: atender as necessidades de informação dos usuários do website.

Dijck (2003) cita que “o principal trabalho de um arquiteto de informação é organizar a informação de um website para que seus usuários possam encontrar coisas e alcançar seus objetivos”. Toub (2000) reforça isso citando a minha definição favorita: “Arquitetura de informação é a arte e a ciência de estruturar e organizar ambientes de informação para ajudar as pessoas a satisfazerem suas necessidades de informação de forma efetiva”.

Conhecer os usuários, suas necessidades, hábitos, comportamentos e experiências é fundamental para elaborar a arquitetura de informação de um website, mas não é suficiente. É necessário também entender as características do conteúdo que será apresentado (volume, formato, estrutura, governança, dinamismo, etc.) e as especificidades do contexto de uso (objetivo do website, cultura e política da empresa, ambiente de uso, restrições tecnológicas, etc.). Essa trinca, usuário–conteúdo–contexto e suas interdependências são únicos para cada website e o papel do arquiteto é justamente conseguir equilibrá-las para que a informação certa seja acessada pela pessoa certa no momento certo (ROSENFELD; MORVILLE, 2002).

### 4.2. - Três Tipos de Arquitetura de Aplicação Web

Expressões como “aplicação web”, “arquitetura front-end”, “Web 2.0” e “aplicações para HTML5” tornaram-se bastante correntes. Acontece, porém, serem frequentemente utilizadas de uma forma errônea, alienada da especificidade da implementação e do uso da arquitetura de aplicação web. Apresento aqui três tipos de aplicações:

#### **4.2.1. - Servidor HTML**

A arquitetura mais conhecida. O servidor gera conteúdo HTML que envia para o cliente como uma página completa de HTML. Por vezes, esse tipo de arquitetura é chamado “Web 1.0”, por ter sido o primeiro a aparecer e a dominar a rede.

#### **4.2.2. - “Widgets” de geração JS (AJAX)**

Arquitetura mais desenvolvida que o primeiro tipo, com a diferença que a página exibida pelo navegador consiste em “widgets” (unidades funcionalmente independentes). Os conteúdos são enviados através da consulta do AJAX ao servidor, ora como página de HTML já construída, ora como JSON, que se converte no conteúdo da página através da ligação Java Script.

A sua principal vantagem é que as atualizações do servidor chegam à parte da página solicitada pelo cliente. Também é positivo que os “widgets” estejam separados de um modo funcional, pois, se um “widget” específico trabalhar com uma parte da página, as mudanças introduzidas não afetam as restantes.

#### **4.2.3. - Página única orientada para os serviços de aplicações Web (Web 2.0 e aplicações HTML5)**

A expressão “Web 2.0” não é a mais adequada, por implicar que os utilizadores introduzam conteúdo. No fundo, refere-se a projetos e serviços ativamente desenvolvidos e sujeitos a melhorias pelos próprios utilizadores, como acontece nos blogs, wikis e nas redes sociais.

Neste caso, uma página HTML contendo código Java Script é descarregada pelo servidor. O código remete para um serviço web específico e recolhe apenas conteúdos corporativos. Estes são utilizados pela aplicação Java Script, que gera a página com conteúdo HTML.

Este tipo de arquitetura representa uma evolução face à anterior, que não permite exibir uma grande quantidade de funções estruturadas e inter-relacionadas.

Note-se ainda que atualmente são raras as aplicações que funcionem autonomamente “off-line”. Esta abordagem permite uma fácil conversão inversa, que consiste na publicação de uma aplicação já existente na web.

### **4.3. - Programação para Web**

O computador ainda não entende exatamente o que o ser humano fala. O que o computador entende são zeros e uns. Graças à velocidade que possui, ele pode ler uma gigantesca quantidade de zeros e uns e fazer coisas incríveis, mas, para os seres humanos é impossível instruir o computador dessa forma. Por isso, foram criados códigos que dão ordens básicas. A organização dessas ordens é

uma gramática e seu conjunto de palavras-chave é sua ortografia, e assim temos uma linguagem, uma forma de comunicação com o computador que diz a ele o que é para ele fazer e quando.

Dessa forma podemos escrever mais facilmente e pedir para uma aplicação traduzir esta linguagem para a que o computador entenda, ou seja, dizemos que uma linguagem mais próxima dos seres humanos é de alto nível, já a linguagem de zeros e uns é de baixo nível. Essa tradução pode possuir duas abordagens: “compilativa” e “interpretativa”.

Na compilação a tradução é feita de uma vez. Todo o programa é traduzido para a linguagem de baixo nível e pode ser executado indefinidamente, sem precisar ser compilado novamente. É como se um tradutor de inglês pegasse um livro e traduzisse ele para o português. O livro traduzido pode ser impresso várias vezes sem precisar ser traduzido novamente. Como exemplos de linguagem “compilativa” temos Pascal e C.

Na interpretação a tradução é realizada enquanto as linhas são lidas. Toda vez que a linha é lida, é traduzida e executada. É como se um tradutor estivesse em uma palestra e, enquanto o palestrante fala, ele vai traduzindo para o público. Se o palestrante repetir tudo de novo, em um momento que ele precise explicar novamente para alguém, o tradutor terá que traduzir novamente. Então, se uma linha for lida mil vezes em um código, ela será interpretada e executada mil vezes. Como exemplo de linguagens interpretativas, temos o PHP, Python e Java Script.

A forma como a internet é construída privilegia as linguagens interpretativas. Observe a Figura 5, a seguir. No passo 1, o usuário solicita uma página para um servidor web. O servidor percebe que a página solicitada é uma aplicação em PHP e solicita sua interpretação no passo 2. Como resultado, o interpretador pode criar uma página web com o resultado da computação e envia como resposta para o usuário, no passo 3.

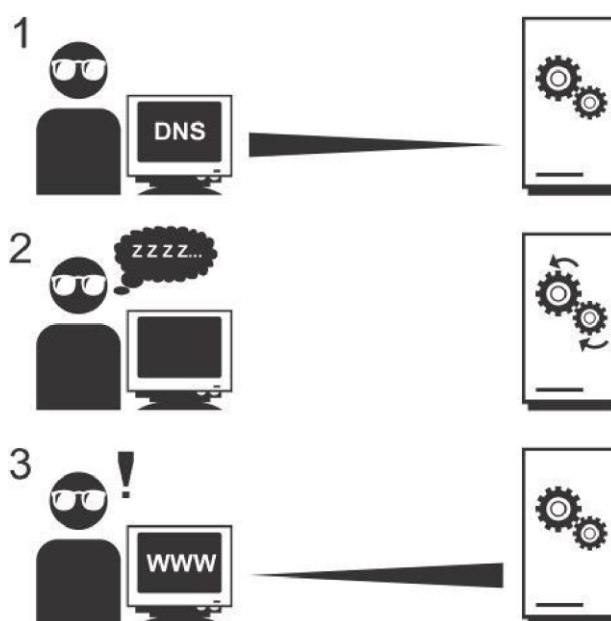


Figura 5 - Ilustração que mostra os passos simplificados da solicitação de uma página web. Fonte: Próprio autor.



Não importa quantas vezes o usuário faça isso ou quantos usuários diferentes solicitem a mesma página, toda vez o aplicativo (programa) vai ter que interpretar e executar. A velocidade de tradução depende de vários fatores, mas, geralmente a interpretação é mais lenta que a compilação.

É dessa forma que vamos pensar de agora em diante. O usuário solicita algo ao programa, ele faz a computação e devolve uma página como resposta.