



UNINGA - CENTRO UNIVERSITÁRIO INGÁ

Orientador: Professor Rold Scheider
Título: Trabalho 2º bimestre
Orientado: Martins Kazakeviche Neto – ra:333652
<https://github.com/martinskso/programacaoWebII.git>

PROGRAMAÇÃO WEB II:

Análise e Desenvolvimento de Sistema

SUMÁRIO

INTRODUÇÃO	4
OBJETIVO.....	5
JUSTIFICATIVA.....	6
HTML E VUE.....	7
O QUE É DATA BIND.....	12
IMPLEMENTAÇÃO VUE	14
MÉTODO E INSTÂNCIA.....	17
VARIÁVEL VUE.....	19
SISTEMA WEB	21
DOM.....	24
REFERÊNCIA	25

INTRODUÇÃO

A partir da prova em anexo, produzir texto baseado em material de pesquisa (links, artigos e estudos).as questões de alternativa, justifique com exemplos a alternativa correta. As dissertativas, produza um texto também com links e material de referência.

2 OBJETIVOS

Objetivo Geral ou Primário

Os textos devem ser autorais, contendo apenas as referências para os materiais que comprovarão a sua explicação.

Objetivos Específicos ou Secundários

Cada explicação deve conter ao menos 1/2 página (baseando-se numa A4) e ter ao menos 3 fontes de referência.

3 JUSTIFICATIVA

O trabalho apresentado e seu desenvolvimento, vem a demonstrar o conhecimento e aprendizagem no que cabe ao aluno e futuro programador, as bases do desenvolvimento web, com a utilização de linguagem de marcação e a aplicação de frameworks.

4 HTML E VUE.JS

Sobre uma diferença prática de um HTML com e sem o uso do VUE.JS, que justifique sua utilização. Podemos demonstrar com um trecho do código e as explicações necessárias

4.1 HTML

Primeiramente é preciso entender que quando falamos de HTML (de HyperText Markup Language) estamos falando de uma linguagem de marcação de texto através das TAGs, que permite apresentar informações na internet tais como links, imagens e o próprio texto. Todas as páginas que você abre são a leitura e a interpretação de um arquivo que contém esse código HTML.

O HTML não é a única linguagem presente quando falamos de páginas web. Outras linguagens são usadas para enriquecer o conteúdo e a experiência do usuário como o CSS e o Javascript.

A screenshot of a code editor window titled 'index.html'. The editor shows a simple HTML document structure. The code is as follows:

```
1 <html>
2   <head>
3     <meta charset="UTF-8">
4     <title>html 5</title>
5   </head>
6   <body>
7     <h1>
8       Titulo 1
9     </h1>
10    <h1>
11      Titulo 2
12    </h1>
13    <h1>
14      Titulo 3
15    </h1>
16    <h1>Titulo do site</h1>
17  </body>
18 </html>
```

Exemplo de um código simples HTML

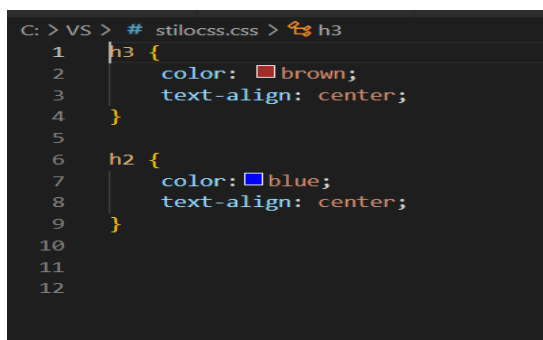
Pode observar aqui dentre as outras, as TAGs <title>, onde direciona o que ira ser escrito na pagina do site. Logo a baixo a tag <body> que identifica a msg a ser escrita na pagina, é onde fica todo o conteúdo de texto, imagem e vídeos, em que o usuário vê e interage, nele os conteúdos são estruturados pelas demais tags do HTML.

As tags são códigos que definem toda a estrutura da página, tais como o seu tamanho, a fonte da letra, as cores, as quebras de linha e etc. A maioria dos elementos do documento HTML são compostos por uma estrutura de abertura e uma de fechamento, como <tag> e </tag>. Há também tags de estrutura única, como a tag
 que realiza uma quebra de linha. (HOMEHOST,2006)

A aplicabilidade das outras linguagem no HTML tras a base da programação Front-End, que é o tripé: HTML + CSS + Javascript. Estas três linguagens de programação se completam e juntas elas trabalham de forma uniforme, como um corpo em sue todo.

4.2 O CSS

(Cascading Style Sheets) é um mecanismo para adicionar estilo com cores, fontes, espaçamento, entre outras a um documento web. Em vez de colocar a formatação dentro do documento, o CSS cria um link para uma página que contém os estilos.



```

C: > VS > # stilocss.css > h3
1  h3 {
2    color: brown;
3    text-align: center;
4  }
5
6  h2 {
7    color: blue;
8    text-align: center;
9  }
10
11
12

```

Observe a tag *style* do CSS: ela possui toda uma sintaxe especial. Antes de mais nada, devemos definir o tipo de estilo que vamos usar.

4.3 JAVASCRIPT

Já o JAVASCRIPT é uma linguagem de programação criada para ser parte dos navegadores web, para que scripts possam ser executados do lado do cliente e interajam com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

JavaScript é a linguagem de programação da Web. A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos – em computadores de mesa, consoles de jogos, tablets e smartphones – incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas. (Flanagan. Pag 19. 2013).

O JavaScript mudou muito com o passar dos anos. Hoje, trata-se de uma tecnologia que pode ser utilizada tanto no cliente-side de uma aplicação, quanto no server-side.

Trata-se de uma linguagem baseada em eventos. E, ainda, você conta com diversos frameworks(AngularJ, nodeJS, Ember.JS, ReactJS) e bibliotecas específicas para a linguagem, como o JQuery, que apoiam o seu desenvolvimento com a linguagem.

```
C: > VS > JS meuScript.js > mountedApp
1  const meuPrimeiroApp = Vue.createApp({
2    data() {
3      return {
4        msgInicial: 'Bem-vindo ao VUE.JS',
5        verMsg: false
6      }
7    }
8  })
9  const mountedApp = meuPrimeiroApp.mount('#divPrincipal')
```

Qual a finalidade da utilização da linguagem javascript em uma pagimna web? A resposta e simples, temos:

Validação de formulários: assim, o seu usuário não passa pela frustração de ter que recarregar a página inteira simplesmente porque digitou um campo errado;

Esconder ou apresentar informações da página, conforme o comportamento de seu usuário;

Janellas de aviso ou de confirmação;

Sistemas de autocomplete, algo parecido com quando você digita na busca do Google e ele completa com as palavras possíveis ou já salvas pelo seu navegador;

Sistemas ou aplicações web que mostram informações que mudam constantemente, conforme eventos internos ou externos ao site/aplicação

Interações com elementos HTML, para que estes mudem dinamicamente.

Interações com regras CSS, da mesma forma que com o HTML.

Dessa forma, podemos afirmar com certeza que o JavaScript é o responsável por dar vida a sua página ou aplicação web.

4.4 JAVASCRIPT E O VUE.JS

Para entendermos a aplicação do Vue é necessário saber um framework Javascript open source para a criação de aplicações web. O Vue.js possibilita criar aplicações de forma reativa. O Vue.js faz a utilização de um DOM virtual, o que faz com que seja extremamente performático na maioria das situações. Além disso, conta com uma arquitetura muito bem estruturada por meio da criação de componentes reusáveis.

Este framework é muito utilizado para criar aplicações SPA (Single Page Applications) que tem como sua principal característica é a Reatividade e progressividade, conceitos básicos utilizados hoje para uma programação de *frontend com aplicações modernas*, bem como sua importância para desenvolver vários outros tipos de interfaces, com foco na interação e experiência do usuário. Vale lembrar que estamos falando de um framework que teve um crescimento surpreendente, e superou grandes frameworks em comparação ao número de estrelas no GitHub.

4.5 ENTENDENDO A APLICAÇÃO

Para entender esse relacionamento na utilização de HTML sem o uso de um VUE.JS e preciso primeiro definir que quem utiliza o HTML, tecnicamente não é uma programação. Não se programa em HTML e CSS. Essas duas tecnologias não são efetivamente linguagem de programação. Javascript, PHP, PYTHON, C++ e C#; são linguagens de programação. A própria sigla (HTML, CSS) mostra que não são uma linguagem de programação, e sim de desenvolvimento. Melhor dizendo, cria-se sites em HTML.

A utilização do Hyper Text Markup Language na criação de um site, trabalha com a integração destes componentes citados:

HyperText: hiper texto (é um texto onde se pode clicar e “ir” para outro texto ou imagem);

Markup: marcas, é baseada em marcas. Não possui estruturas de condição, repetição, variáveis, objetos, BD, componentes que são utilizados em linguagens propriamente ditas de programação.

Language: THML é uma linguagem de marcação para hipertexto.

Sendo assim, tecnicamente a pratica de HTML sem o uso do VueJS, não interferiria diretamente no uso específico do HTML, pois estamos falando de duas situações amplamente diferentes.

Sendo o HTML uma linguagem focada em conteúdo, sendo texto, imagem, vídeos, tabelas e listas. Onde é possível definir a estrutura, posicionamento dos elementos e a divisão de uma página em blocos visuais e combinar texto, imagens e vídeos para aplicações web.

```
<!DOCTYPE html>
<html>
<body>

<h2>Absolute URLs</h2>
<p><a href="https://www.google.com/">Google</a></p>
<p><a href="https://www.facebook.com/">Facebook</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>

</body>
</html>
```

A aplicação do VueJS não sendo usada não afetaria um código somente em HTML como o do exemplo acima, pois as linhas de código direcionam para o acesso a outras páginas e endereços.

Mas a utilização do HTML sem a linguagem Javascript, sim, pode afetar diretamente o desenvolvimento de uma página web. Pois estamos falando de uma linguagem de programação dinâmica, onde permite a execução de instruções e respostas a seus usuários. Sem sua utilização junto com ao HTML, o site perde todo seu dinamismo.

5.0 O QUE É DATA-BIND E COMO SE DÁ A SUA IMPLEMENTAÇÃO EM UM PROJETO VUE.

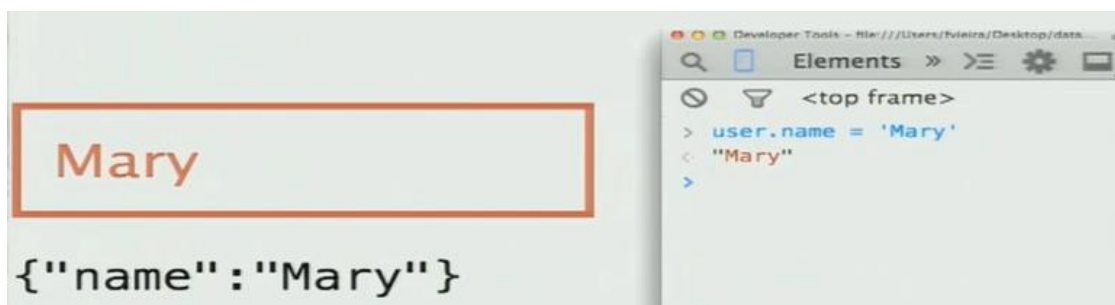
Data Binding é a maneira de puxar informações de um objeto e exibi-la na interface do usuário, sem escrita do código.(Cunha, José Antonio.2012)

Exemplificando melhor a compreensão do funcionamento, o DATABIND é o processo de sincronização de dados, onde se tem o objeto que representa as informações de alguma “coisa” e a seu modelo de interface. Qualquer alteração que se faça no objeto, essa alteração é refletida automaticamente na interface.

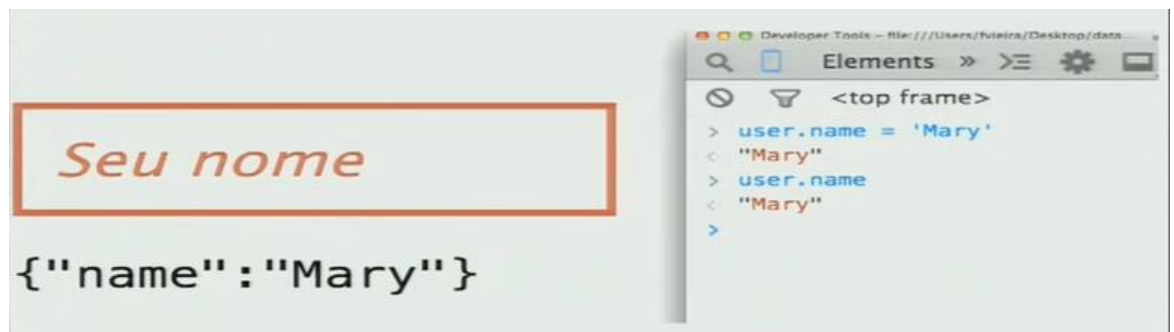
Existem dois tipos de DATA bind o ONE WAY BIND e o TWO WAY BIND.

5.1 ONE WAY BIND

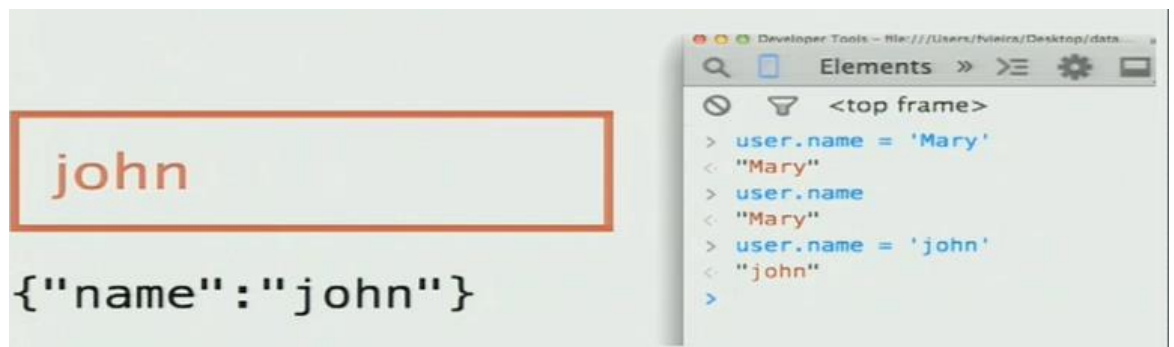
O One Way Bind parte do princípio que se tem o objeto “source”, e será pego os dados desse objeto e refletido nesta interface. Sendo assim qualquer alteração que for feita a partir da interface, esses dados não são alterados diretamente neste objeto. Sua funcionalidade trabalha como um “read only”.



Perceba que o campo está atrelado ao objeto “user.name”, quando se altera a propriedade user.name de objeto ele define o valor do campo



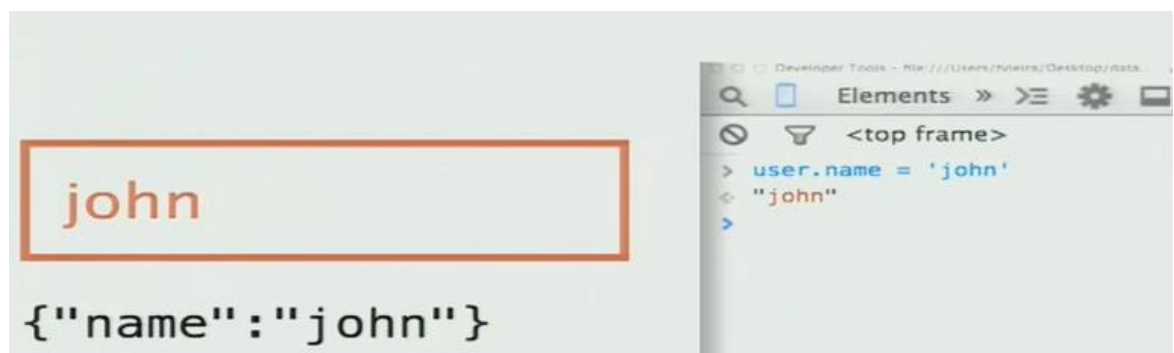
Mas se altera apaga a propriedade do campo, a propriedade permanece com seu valor inalterado.



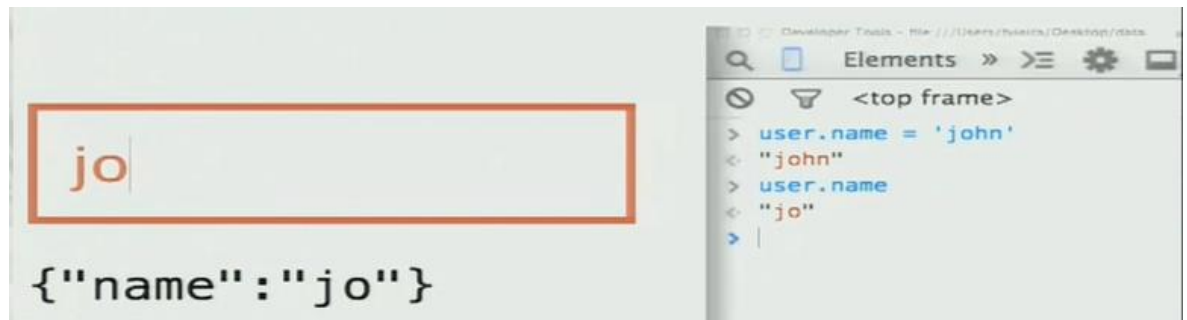
Mas se altera o valor da propriedade do objeto, reflete no campo novamente.

5.2 TWO WAY BIND

Esse sistema tem uma ação diferente do anterior, sendo que se altera o objeto ou alterar a partir da interface, isso reflete diretamente no objeto targete e vice-versa.



Foi definido a alteração na propriedade “name” sendo assim altera-se o objeto.



Se alterara o valor do campo diretamente na interface, o valor do objeto é alterado.

6.0 IMPLEMENTAÇÃO VUE

Cite os passos (conceituais) para a implementação de um projeto Vue desde o início, apresentando a sequência dos acontecimentos. Para tanto, use como exemplo os trechos abaixo (um de um arquivo index.html e o segundo de um arquivo meuScript.js)

- *index.html*

```
<script src="http://unpkg.com/vue@next"></script>

<DIV id="divPrincipal">
  <h1>{{ mensagemInicial }}</h1>
</DIV>

<script src="./meuScript.js"></script>
```

- *meuScript.js*

```
const meuPrimeiroApp = Vue.createApp({
  data() {
    return {
      mensagemInicial: 'Bem-vindo ao VUE.JS',
    }
  }
})

const mountedApp = meuPrimeiroApp.mount('#divPrincipal')
```

6.1 INDEX.HTML

O framework Vue.js consiste de um pequeno arquivo JavaScript e, portanto, sua “instalação” resume-se à adição da referência a esse script na página HTML, seja baixando o arquivo na página oficial do VUE, seja apontando para o script em seu src, da seguinte forma:

```
<script src="http://unpkg.com/vue@next"></script>
```

6.2 VUE.JS

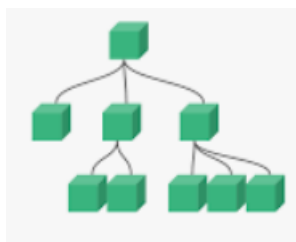
A partir daí, já é possível utilizar os recursos do Vue.js. O primeiro passo para isso é criar uma instância da classe Vue com sua propriedade e apontando para o id de um elemento da página, o qual representará o container principal da aplicação como o exemplo abaixo:

```
<DIV id="divPrincipal">  
  <h1>{{ mensagemInicial }}</h1>  
</DIV>
```

Logo abaixo temos a referência da utilização de uma tag `<h1>` e `</h1>`, onde está vinculada “mensagemInicial” dentro dos “bigodes duplos”.

O que isso está fazendo para o Vue.JS? Sempre que pensarmos na estrutura de funcionamento do VUE teremos a referência de acesso dos arquivos como uma árvore.

O HTML é construído, manipulado, criado muito próximo como uma estrutura de árvore.



Quando se cria uma “DIV” estamos criando primeiro nó da estrutura desta árvore, e o Vue.JS precisa saber e ouvir cada elemento desse nó, onde a partir daí poderá interceptar ou alterar o comportamento de cada ponto dessa árvore como no exemplo acima.

Sendo assim, a String `{{mensagemInicial}}` é a forma de printar na saída, ou seja, no browser uma variável.

No núcleo do Vue.js está um sistema que nos permite declarativamente renderizar dados no DOM (Document Object Model) usando uma sintaxe de *template* simples:

```
<h1>{{ mensagemInicial }}</h1>
```

6.3 MEU SCRIPT.JS

A partir da aplicação de criação do Vue.JS

```
<script src="http://unpkg.com/vue@next"></script>
```

É possível criar o “vue.create.App” para criar um novo aplicativo pelo VUE, que receberá um objeto com todas as definições da aplicação criada.

```
const meuPrimeiroApp = Vue.createApp({
```

6.4 DATE

A propriedade “date” é um método que também retorna os dados da aplicação.

```
data() {
  return {
    mensagemInicial: 'Bem-vindo ao VUE.JS',
  }
}
```

Daí por diante é necessário montar o local desta aplicação em alguma parte do HTML, no caso a DIV com o id #app

```
const mountedApp = meuPrimeiroApp.mount('#divPrincipal')
```

Apartir daqui temos toda funcionalidade do Vue.JS utilizando sua aplicação de ser

reativo, ou seja. É onde amágica acontece. Se mudar a msg, o VUE automaticamente atualiza o HTML e seus dados, não apenas com Strings, mas com qualquer dados que se utilize, isso que faz o VUE ser considerado reativo.

7.0 MÉTODO E INSTANCIA

04 - O método que instancia uma aplicação VUE 3 no HTML é:

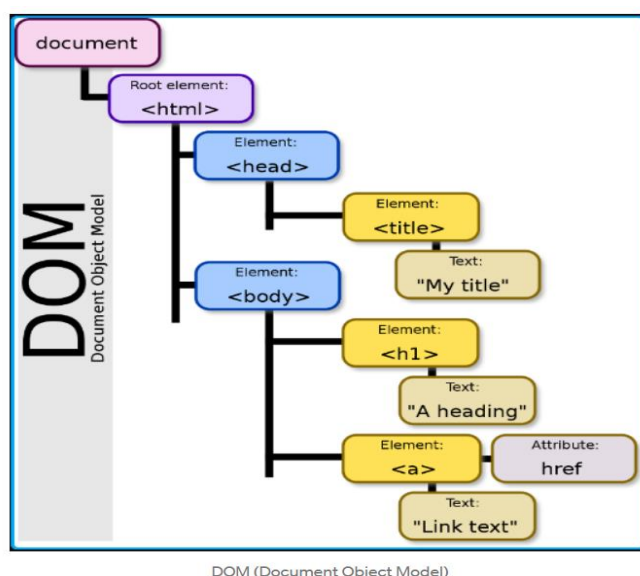
- A) mount
- B) script
- C) const
- D) body
- E) html

7.1 O QUE É A INSTÂNCIA VUE?

Primeiramente vamos entender a instância VUE que é a raiz de nossa aplicação. Ela é criada passando um objeto com suas opções. Nesse objeto tem uma variedade de propriedades opcionais que dão a instância a habilidade de armazenar dados e realizar ações.

```
const meuPrimeiroApp = Vue.createApp
```

A instância Vue ela precisa ser conectada com um elemento de sua escolha, para formar um relacionamento entre a instância e uma parte do DOM (Document Object Model).



7.2 O QUE É UM MÉTODO?

Métodos são uma parte fundamental da programação orientada a objetos. É por meio deles que alteramos e consultamos atributos dos nossos objetos. Eles são o meio que utilizamos para ler, modificar e definir os atributos de um objeto. Estão sempre associados a uma classe e, desta forma, podem ter seu escopo definido para poderem ser acessíveis a diferentes níveis: podemos definir métodos que somente podem ser acessados internamente naquela classe, métodos que não podem ser acessados diretamente e métodos que podem ser acessados de qualquer parte do nosso projeto.

Os métodos mais utilizados, que inclusive geraram em diversas IDEs soluções prontas para a criação deles são os métodos conhecidos como "Getters and Setters", basicamente são os métodos utilizados para se obter a informação sobre algum atributo do objeto (getters) e definir o valor de algum atributo do objeto (setters).

7.3 O QUE É UMA INSTÂNCIA?

Um aplicativo Vue.js começa com uma instância Vue. O objeto de **instâncias** é o objeto principal de nosso aplicativo Vue. Isso nos ajuda a usar os componentes do Vue em nosso aplicativo. Uma instância Vue usa o padrão MVVM (Model-View-View-Model). O construtor Vue aceita um único objeto JavaScript denominado objeto de opções. Quando você instancia uma instância do Vue, você precisa passar um objeto de opções que pode conter opções para dados, métodos, elementos, modelos, etc.

7.4 SCRIPT:(RESPOSTA)

A tag <script> é utilizada para criar scripts (Javascript) dentro do seu arquivo HTML. Ela define um “script” em seu “Javascript” executado no lado do cliente (Front-end).

É um elemento que é executado no navegador da pessoa (cliente) que está acessando sua página, lendo o arquivo html e não no servidor onde está hospedado o seu arquivo html contratado pela hospedagem do site.

No caso aqui, permite o script ser escrito dentro dela e apontar uma outra fonte externa, no caso um método de aplicação do VUE.JS.

a) mount : O comando mount é um comando do sistema Linux, e é utilizado para montar um dispositivo na hierarquia do sistema de arquivos do Linux manualmente. Para desmontar um dispositivo, o comando umount pode ser usado.

c) const: A declaração const cria uma variável cujo o valor é fixo, ou seja, uma constante somente leitura. Isso não significa que o valor é imutável, apenas que a variável constante não pode ser alterada ou retribuída.

Assim como as variáveis declaradas com a palavra-chave let, constantes também tem escopo de bloco.(Alura.2021).

d) HTML e body : Define o corpo do documento HTML. Junto com a tag <html> e a <head> formam a tríade de tags mais importantes e essenciais de qualquer site feito em HTML.

Pensando numa forma dinamica como em um livro, tag <html> seria a capa do livro (tanto a parte da frente, como a de trás), elas definem o livro, onde começa e onde termina. Já a tag <head> seria o sumário, informações e dedicatória do autor, e onde se encontra as informações deste livro.

8 VARIÁVEL VUE

Qual o procedimento para disponibilizar o conteúdo de uma variável Vue a um arquivo HTML? (valor 1,0)

Escrever o nome dela dentro de uma tag h1
colocar o nome da variável cercada pelos símbolos {{ e }}
Escrever o nome dela dentro de uma tag p
Escrever o nome dela dentro da sessão style.
Apontar seu nome na tag head

8.1 PARAMENTRO

Para entendermos o passo a passo deste procedimento, comecemos pelo fim. No arquivo “meuscript Vue.JS primeiramente.

Criaremos uma constante com o nome de meuPrimeiroApp que recebe “Vue.createApp.”

Logo após, criaremos alguns parâmetros dentro da função chamada DATE :

```
const meuPrimeiroApp = Vue.createApp({
  data() {
```

8.2 O DATE II

Para colocarmos dados neste componente, utilizaremos uma função de retorno chamada “date”, ele tem uma notação de função de retorno, e seguindo é criado um objeto “return” que recebe um tipo de dado (‘Bem vindo ao VUE. JS’) e poderemos utilizar isso diretamente no componente. Ex:

```
return {
  mensagemInicial: 'Bem-vindo ao VUE.JS',
```

A partir daí temos nossa estrutura de função Vue.JS pronta para acessar o arquivo HTML.

```
<DIV id="divPrincipal">
  <h1>{{ mensagemInicial }}</h1>
</DIV>
```

A {{mensagemInicial}} funciona como uma variável que acessa o outro valor no arquivo Vue (‘Bem-vindo ao VUE.JS’) e imprime na tela.

No arquivo HTML a partir do momento que criamos uma DIV e entre os (bigodes duplos ou mustaches) referenciamos o código Vue dentro com a variável de acesso. Podemos assim mostrar seu conteúdo a um arquivo HTML.

Resposta “b”, colocar o nome da variável cercada pelos símbolos {{ e }}.

a) Escrever o nome dela dentro de uma tag h1: as tag H1 é a abreviação do inglês para header 1 ou cabeçalho 1, tanto h1, h2 e assim por diante estão relacionados aos elementos apresentados na página. O h1 tem sua função de título e o h2 de subtítulo.

c) tag <p> : representa um parágrafo no HTML, são usados para agrupar conteúdos relacionados de qualquer tipo, como imagens e campos de um formulário e não tem função de variável de acesso do Vue. A tag <p> pode ser utilizada como acessibilidade. O leitor anunciará a existência de um parágrafo, mas não lerá nenhum conteúdo, pois não há.

d) sessão style: O elemento style é um elemento do HTML, contém informações de estilo para um documento ou parte do documento. As informações de estilo específico estão contidas dentro dele.

e) Apontar seu nome na tag head: O head de um documento HTML não aparece no navegador quando a página é carregada. Ele contém informações como “title, links para CSS e outros meta dados.

9. SISTEMA WEB:

06 - Das alternativas abaixo, aponte a que mais representa uma limitação para um sistema WEB (valor 1,0)

- A) Um sistema WEB está limitado às capacidades do navegador WEB
- B) Um sistema WEB não pode gravar arquivos locais, a não ser por downloads
- C) Um sistema WEB não pode ter formulários vinculados ao banco de dados, mesmo com o auxílio do backend
- D) Um sistema WEB não pode ser executado em diferentes sistemas operacionais
- E) Desde que a Adobe descontinuou o plugin do Flash, os sistemas WEB não podem ter conteúdos dinâmicos

9.1 LIMITAÇÃO DO NAVEGADOR :

Entende-se que um Sistema Web de uma forma simplificada, seja como apenas um software hospedado na internet. Um dos maiores ou talvez um dos principais benefícios, seja a possibilidade de acesso em qualquer lugar através de qualquer dispositivo com navegador e conexão à internet.

Porém podemos dizer que entre as alternativas propostas, a mais próxima que possa limitar tal sistema, seja às capacidades do navegador WEB ou web browser ou simplesmente browser. Conforme alguns exemplos, a relação Sistema Web e Browsers ainda é pauta para desafios de muitos desenvolvedores web:

Embora cada navegador tenha suas características e recursos, há pontos problemáticos comuns entre eles. Isso faz com que a evolução da tecnologia de sites seja interrompida até que as brechas ou ausências de funcionalidades sejam corrigidas.

A iniciativa começou em 2019, quando Google, Mozilla e outros menores criaram o encontro para descobrir as dificuldades dos desenvolvedores da web. Como o saldo foi positivo, cada fabricante passou a incentivar a reunião anual no intuito de reconhecer as falhas e apontar soluções para corrigi-las. Os interessados podem enviar comentários e sugestões na página do Interop 2022, já que há um fórum inteiro de discussões para tratar dos detalhes. (CANALTECH, GOOGLE, APPLE, MICROSOFT, 2022).

Apesar do avanço do desenvolvimento do funcionamento das paginas web hoje, algumas aplicações e funcionamentos podem ser afetadas ou limitadas com diferentes navegadores.

Mas como isso pode acontecer? Um navegador é um programa que possibilita que acessemos as páginas existentes na web. Isso acontece porque é ele quem transforma as diversas linguagens da internet as quais os sites são escritos, para a maneira que vemos as páginas. Assim, uma página nada mais é que uma série de códigos – HTML, ASP, PHP, CSS – que o navegador transforma na imagem que vemos.

A web está em constante transformação, com um navegador antigo que não esteja atualizado algumas funcionalidades, sites ou aplicativos podem não funcionar adequadamente. Sendo assim, quando for acessado o site do banco, ou ver um vídeo, pode ser que eles simplesmente não abram, e isso não esta ligado ao site, mas sim do navegador desatualizado. Talvez dentre eles este seja um dos maiores limitadores de tal sistema.

Pode-se entender como uma limitação também de um Sistema Web, o fato de não ter acesso a arquivos locais ou somente por download. Ao utilizar um Sistema Web é fato de o usuario ter restrições de acesso a certos recursos de hardware do dispositivo em que está sendo executado.

Navegadores modernos, recusam-se a passar do protocolo http para protocolo de arquivo, sendo possivel acessar estes arquivos somente localmente. Um protocolo é algumas letras, dois pontos e depois duas barras. HTTP e FTP são protocolos validos C:/ não é um edereço valido na web.

Seu navegador pode não assumir que está se referindo a um arquivo local. Tem poucas razões para fazer essa suposição, porque os sites públicos geralmente não tentam vincular aos arquivos locais das pessoas.(STACKOVERFLOW. 2018).

Sites do navegador não podem interagir com o sistema operacional muito bem. Se um site por nome de palestra.mp4 para abrir em VLC.exe por exemplo, um site mal-intencionado poderia dizer-lhe para abrir virus.bat em CMD.exe. Ou pode apenas dizer ao PC para executar alguns arquivos Uninstall.exe, ou abrir o File Explorer um milhão de vezes. Isso pode não ser conveniente para o usuário, mas a segurança de HTML e navegador não foi realmente projetada para o evitar tal possibilidade. Dentre este tipo de necessidade para um Sistema Web torna-se obsoleto dentre tantas outras aplicações e desenvolvimento no mercado.

Os Sistemas Web modernos utilizando das suas estruturas HTML juntamente com acesso as linguagens de programação (Javascript, PHP) podem ser acessados a base de dados. Esses dados disponíveis em host, faz com que um formulário em HTML se torne algo mais dinâmico dentro dos processos de programação em Fron-end. Essas linguagens e seus “scripts” fazem uma solicitação para uma base de dados. Esse base dado respondendo essa “query” a linguagem transformando em uma resposta HTML, uma vez gerado esse script, gerado esse HTML até a resposta ao cliente. Dentre elas, hoje existem varias opções para utilização de banco de dados para PHP ou Javascript em banco de Dados, fazendo que com isso, não seja uma limitação de um Sistema Web.

Planilhas são comumente usadas como banco de dados de clientes, produtos, funcionários e afins por empresas dos mais diversos portes. Contudo, as planilhas são difíceis de operar, exigem grandes quantidades de tempo para sua manutenção e organização e são complicadas de analisar a longo prazo. Como dito anteriormente, os sistemas web são aplicações projetadas para serem utilizadas por meio de um navegador de internet. Sendo assim, eles são uma excelente ferramenta para o gerenciamento de banco de dados. (EJCM.2022) A Usabilidade e personalização de um sistema web e uma das mais vantajosas aplicabilidades as necessidades do usuário. Pois seus Web services funcionam com qualquer sistema operativo, plataforma de hardware ou linguagem de programação de suporte Web. Estes transmitem apenas informação, ou seja, não são aplicações Web que suportam páginas que podem ser acedidas por utilizadores através de navegadores Web.

O Adobe Flash Player é um dos reprodutores de mídia mais populares da Internet. O software, criado em 1996 e adquirido pela Adobe em 2005, é multiplataforma, ou seja, permite que diversos navegadores reproduzam vídeos, jogos e animações em formato Flash. Entretanto, o Flash Player começou a ganhar concorrentes mais modernos, o que resultou na decisão da Adobe de suspender o seu funcionamento no final do ano de 2020. A empresa anunciou em 2017 que descontinuará o software este ano, e a data está próxima: no dia 31 de dezembro, o Adobe Flash Player irá, de fato, acabar. No comunicado, a própria companhia reconheceu que outras ferramentas, como HTML5, WebGL e WebAssembly, conseguem desempenhar o papel que antes era realizado pelo plugin.(TECHTUDO.2020).

O Flash Player se integrava aos principais navegadores da época por meio de plugins, como é o caso do Mozilla Firefox, do Internet Explorer e do Safari. O aplicativo permitia que o browser reproduza todo conteúdo criado por meio do Flash presente em qualquer site da web. Os navegadores Google Chrome e Opera trazem um plugin embutido em um sandbox,

dispensando a instalação manual do dispositivo por parte do usuário. Esse tipo de tecnologia durou por muito tempo, até começar a perder espaço para o HTML5, levando em conta que é um sistema mais leve e que permite a fácil manipulação dos elementos de áudio e vídeo sem a necessidade de plugins, como o próprio software da Adobe.

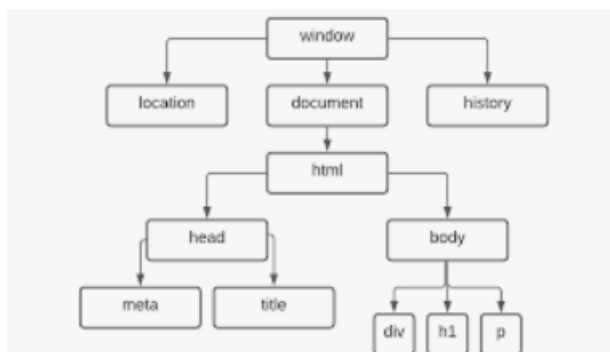
10. DOM

Na programação WEB, o termo DOM se refere a que?

O Modelo de Objeto de documentos (DOM) é uma interface de programação para documentos HTML, XML e SVG. Ele fornece uma representação estruturada do documento como uma árvore. O DOM define métodos que permitem acesso à árvore, para que eles possam alterar a estrutura, estilo e conteúdo do documento. O DOM fornece uma representação do documento como um grupo estruturado de nós e objetos, possuindo várias propriedades e métodos. Os nós também podem ter manipuladores de eventos que lhes são inerentes, e uma vez que um evento é acionado, os manipuladores de evento são executados essencialmente, ele conecta páginas web a scripts ou linguagem de programação.

De uma forma simples o Modelo de Objeto para Documentos, é basicamente um conjunto de objetos dentro do navegador que irá dar acesso aos componentes internos do web site utilizado.

O DOM não funciona dentro de um NODE.JS por exemplo. Mas sim quando eu estou rodando um Javascript dentro do navegador, por isso a estrutura do site criada com HTML. Assim entenderemos melhor a árvore DOM de um site.



O DOM tem início na raiz e basicamente essa raiz dentro do navegador, nós a chamamos de Window. Tudo dentro do javascript está dentro de um objeto chamado window.

- Dentro do window temos vários outros objetos, dentre eles temos conforme o desenho o:

- Location : que diz qual é a localização do site, qual é a URL a pagina atual, qual a pagina anterior.
- Document: que é o documento atual
- History: “de onde você veio e para onde você vai ” que facilita a navegação dentro de seu site.

Dentro do objeto “document” temos um documento muito importante, que é o HTML, que é a propria estrutura do site. E dentro dele mais dois documentos filhos sendo o cabeçalho e o corpo. Nesta árvore hierarquica teremos mais e mais documentos representativos de um site que foi criado.

Com o DOM, o programador ganha infinitas possibilidades. Ele permite, por exemplo, que sejam criadas aplicações que alteram os dados da página, sem a necessidade de efetuar atualização.

É possível também criar aplicações possíveis de serem customizáveis pelo usuário e mudar o layout da página também sem precisar da atualização. Ele literalmente permite arrastar, mover e excluir elementos facilmente. Exigindo do profissional, apenas, criatividade para inovar.

REFERÊNCIAS

FLANAGAN, David. **JavaScript: o guia definitivo**. 6. Ed. Porto Alegre: Bookman, 2013.

DEITEL, Paul. **Java: como programar**. 10. Ed. São Paulo: Pearson Education do Brasil, 2017

SILVA, Maurício Samy. **Fundamentos de Java e CSS**. 1. Ed. São Paulo: Novatec Editora Ltda, 2016