

Trabalho de Desenvolvimento WEB com HTML, CSS, JavaScript e PHP

Antes de começar, existe um arquivo de CSS e de JavaScript, separados, (além das imagens na página, nas quais serão mostradas somente no print de cada uma das seções das páginas), nos quais estão listados abaixo:

Código do arquivo “estilo.css”

```
/* configurando o corpo e eliminando qualquer excessividade de
espaço, deixando apenas o necessário. */
body{
    margin: 0;
    padding: 5px;
}
/* auxiliar para criar uma página responsiva, deixando mais lado-a-
lado, parte 1 */
.resp-container {
    display: flex;
    flex-flow: row;
    justify-content: space-between;
    vertical-align: middle;
    horiz-align: center;
}
/* auxiliar para criar uma página responsiva, deixando mais lado-a-
lado, parte 2 */
.resp-item {
    white-space: nowrap;
    height: 100%;
}
/* Eliminando as margens e deixando o texto mais junto, eliminando
o excesso de espaço. */
#apresentacao .resp-item p{
    margin: 0;
    padding: 0;
    border: 0;
}
/* Aplicando estilos ao nome do aluno: Fundo verde claro e letras
em amarelo */
#apresentacao .resp-item p:nth-child(4){
    background-color: #50b466;
    color: #ffff00;
}
/* Aplicando estilos ao nome do professor: Fundo azul escuro e
```

Código do arquivo “scripts.js”

```
function changeOrdenateList(tp){
    let el = document.getElementsByClassName("ordenateList");
    for(let elementAt = 0; elementAt < el.length; ++elementAt){
        el[elementAt].attributes.getNamedItem("type").value = tp;
    }
}
function loadDataInTable() {
    let dadosFakeAPI =[
        {"qtde":6,"descricao":"Galão de Cloro 5
litros","vlr":3.52},
        {"qtde":10,"descricao":"Esponja abrasiva pacote c/ 1
und","vlr":2.25},
        {"qtde":15,"descricao":"Caixas de Detergente Líquido 200ml
c/ 20und","vlr":25.87},
        {"qtde":12,"descricao":"Desengordurante galão 5
litros","vlr":45.03},
        {"qtde":300,"descricao":"Esponjas amarela pct c/ 1
und","vlr":0.54},
        {"qtde":45,"descricao":"Refis de esfregão","vlr":8.84},
        {"qtde":7,"descricao":"Sabão em pó 5kg caixa","vlr":9.52},
        {"qtde":3,"descricao":"kit MOP (balde + torcedor de
esfregão)","vlr":280.06},
        {"qtde":8,"descricao":"Vassouras comum","vlr":6.33},
        {"qtde":2,"descricao":"Vassourão para calçada","vlr":9.68},
        {"qtde":5,"descricao":"Rodo","vlr":6.33},
        {"qtde":15,"descricao":"Cabos de esfregão","vlr":14.98},
        {"qtde":5,"descricao":"Galão de limpa-vidro 5
litros","vlr":6.51},
        {"qtde":20,"descricao":"Alcool gel caixa c/ 10
und","vlr":40.64},
        {"qtde":100,"descricao":"rolos papel higienico
100mts","vlr":7.63},
        {"qtde":250,"descricao":"papel toalha seca mão pct c/
500fls","vlr":7.12}
    ];
}
```

```

letras em amarelo */
#apresentacao .resp-item p:nth-child(5){
    background-color: #00088c;
    color: #ffff00;
}
/* Aplicando estilos aos detalhes: Paleta de cores (ou melhor,
detalhes das cores) da logo do Fedora e fonte branca. */
#apresentacao .resp-item p:nth-child(8){
    background-color: #51A2DA;
    color:#fff;
}
/* Aplicando estilo aos detalhes: Paleta de cores dos fundos da
logo do WebStorm (em degradê, via gradiente linear. Fonte preta. */
#apresentacao .resp-item p:nth-child(9){
    background-image: linear-
gradient(135deg,#07C3F2,#087CFA,#07C3F2,#FCF84A);
    color: #fff;
}
/* Aplicando estilo aos detalhes: Detalhes das cores da logo HTML5
e fonte branca */
#apresentacao .resp-item p:nth-child(10){
    background-color: #F16529;
    color:#fff;
}
/* Aplicando estilo aos detalhes: Cores da logo CSS3 e fonte branca
*/
#apresentacao .resp-item p:nth-child(11){
    background-color: #2965F1;
    color:#fff;
}
/* Aplicando estilo aos detalhes: cores da logo JavaScript: Fundo
amarelo e fonte preta */
#apresentacao .resp-item p:nth-child(12){
    background-color: #F7DF1E;
    color:#000;
}
#apresentacao .resp-item img {
    height: 15vmax;
    border-radius: 100%;
}
#apresentacao div:first-child{
    border: 5px solid #00088c;

```

```

let tabela = document
    .getElementById("tabelas")
    .getElementsByTagName('table')[0]
let todoSomatorio = 0;
dadosFakeAPI.map(function (data){
    let novaLinha = document.createElement('tr')
    let coluna = [
        novaLinha.appendChild(document.createElement('td')), //
        novaLinha.appendChild(document.createElement('td')), //
        novaLinha.appendChild(document.createElement('td')), //
        novaLinha.appendChild(document.createElement('td')), //
        novaLinha.appendChild(document.createElement('td')), //
    ];
    coluna[0].innerHTML = data.qtde;
    coluna[1].innerHTML = data.descricao;
    coluna[2].innerHTML = data.vlr.toLocaleString('pt-br',
{style:'currency',currency:'BRL'});
    coluna[3].innerHTML = (data.qtde *
data.vlr).toLocaleString('pt-br',
{style:'currency',currency:'BRL'});
    todoSomatorio += data.qtde * data.vlr;
    for(let posicaoColuna = 0; posicaoColuna < 4; +
posicaoColuna){
        novaLinha.appendChild(coluna[posicaoColuna]);
    }
    tabela
        .getElementsByTagName('tbody')[0]
        .appendChild(novaLinha);
    tabela
        .getElementsByTagName('tfoot')[0]
        .getElementsByTagName('td')[0]
        .innerHTML = todoSomatorio.toLocaleString('pt-br',
{style:'currency',currency:'BRL'});
    });
}

```

```
border-radius: 10px;
padding: 5px;
}
#apresentacao img.icon {
width: 48px;
height: 48px;
border-radius: 0;
}
```

Esses dois arquivos, estão vinculados ao cabeçalho (entre as tags <head> e </head>) da página HTML, seguido do código abaixo:

```
<head>
  <meta charset="UTF-8">
  <title>Trabalho de Desenvolvimento WEB</title>
  <script type="application/javascript" src="./scripts.js"></script>
  <link rel="stylesheet" href="./estilo.css" type="text/css">
  <meta name="viewport" content="width=device-width,initial-scale=1.0">
</head>
```

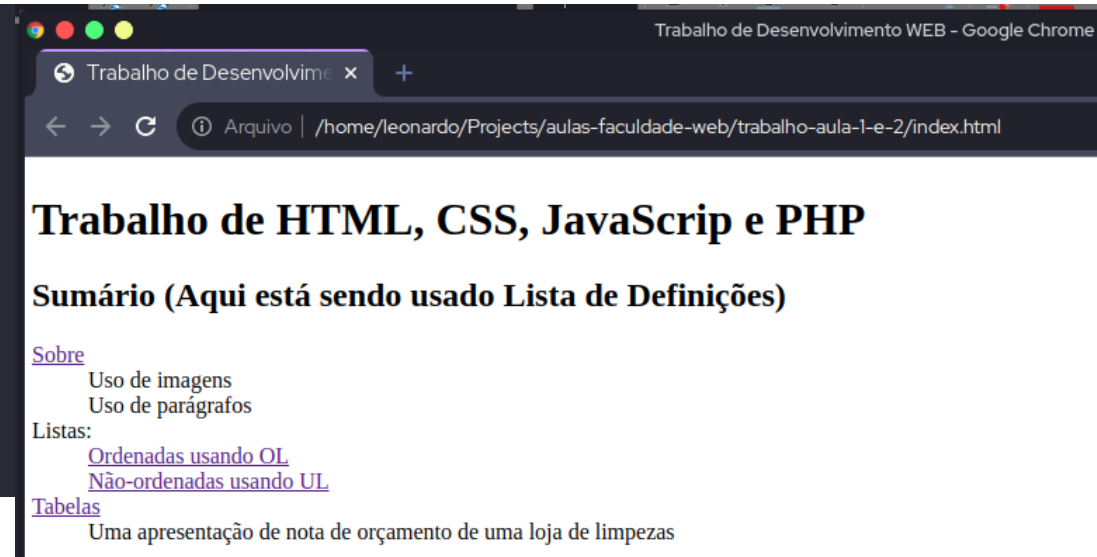
1. A primeira linha é a abertura do cabeçalho <head>
2. A segunda é uma “solicitação” ao navegador para renderizar corretamente os acentos, cedilha alguns caracteres especiais, etc.
3. A terceira linha é o título da minha página, no qual defini como **Trabalho de Desenvolvimento WEB**.
4. A quarta linha é a vinculação do arquivo javascript.
5. A quinta linha é a vinculação do arquivo de CSS.
6. A sexta linha é uma indicação para o navegador tentar comportar a página conforme redimensiona as janelas.
7. Fechamento da tag </head>.

Entre as tags <body> e </body>, vou citar por partes:

Pra começar, na abertura da tag <body>, eu coloquei o atributo “onload”, onde faz a chamada da função do javascript para o preenchimento dinâmico entre as tags <tbody> e </tbody> da tabela (no último elemento em que citarei):

Lista de Definição, H1 e H2

```
<h1>Trabalho de HTML, CSS, JavaScrip e PHP</h1>
<h2>Sumário (Aqui está sendo usado Lista de Definições)</h2>
<!-- Lista de Definições -->
<dl>
  <dt><a href="#apresentacao">Sobre</a></dt>
  <dd>Uso de imagens</dd>
  <dd>Uso de parágrafos</dd>
  <dt>Listas:</dt>
  <dd><a href="#OL">Ordenadas usando OL</a></dd>
  <dd><a href="#UL">Não-ordenadas usando UL</a></dd>
  <dt><a href="#tabelas">Tabelas</a></dt>
  <dd>Uma apresentação de nota de orçamento de uma loja de
limpezas</dd>
</dl>
```



Parágrafos, texto em si, estilização com CSS, H3 e imagens

Aqui faço minha apresentação. Aqui encontra-se bastante parágrafos. Trabalhei nela bastante no CSS.

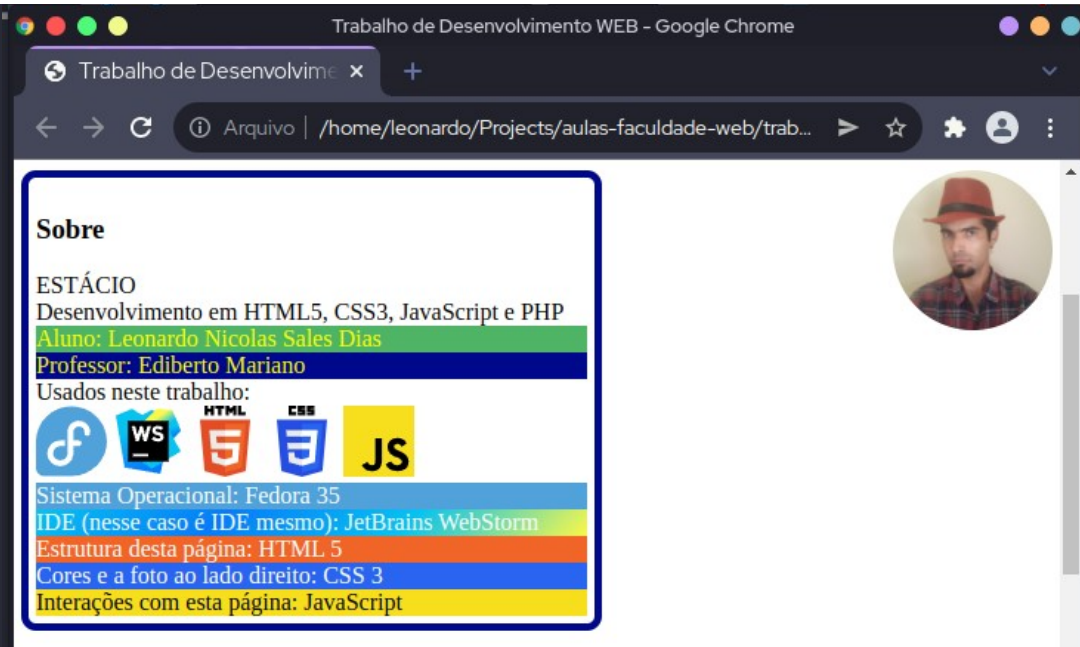
```
<!-- Imagens, parágrafos, H3, etc. -->
<div id="apresentacao" class="resp-container">
  <div class="resp-item">
    <h3>Sobre</h3>
    <p>ESTÁCIO</p>
    <p>Desenvolvimento em HTML5, CSS3, JavaScript e PHP
    &nbsp; &nbsp;</p>
    <p>Aluno: Leonardo Nicolas Sales Dias</p>
    <p>Professor: Ediberto Mariano</p>
```

A O print está na próxima página, nesta mesma coluna. Pois o tamanho dela não coube aqui.

```

<p>Usados neste trabalho:</p>
<p>
    
    
    
    
    
</p>
<p>Sistema Operacional: Fedora 35</p>
<p>IDE (nesse caso é IDE mesmo): JetBrains WebStorm</p>
<p>Estrutura desta página: HTML 5</p>
<p>Cores e a foto ao lado direito: CSS 3</p>
<p>Interações com esta página: JavaScript</p>
</div>
<!-- No browser, a foto aparece oval, por causa do CSS. -->
<div class="resp-item">
    
</div>
</div>

```



Listas Ordenadas:

Aqui as listas ordenadas, que inclusive foi introduzido uma caixa de seleção para indicar se é exibido no padrão (numérico, em ordem alfabética ou números romanos) e javascript para fazer a tal alteração nas listas ordenadas, utilizando o atributo na abertura da tag `<select>` como `onchange="changeOrdenateList(this.value)"`.

Os print's da página:

Em ordem numérica (normal):

Trabalho de Desenvolvimento WEB - Google Chrome

Trabalho de Desenvolvimento x +

Arquivo | /home/leonardo/Projects/aulas-faculdade-web/trabalho-aula-1-e-2/index.html

Lista Ordenada de Frutas

Selecione como deseja ordenar a lista:

Ordem numérica (1, 2, 3, 4, 5) ▼

1. Abacaxi
2. Melão
3. Melancia
4. Banana
 1. D'água
 2. Da Terra
 3. Prata
5. Acerola
6. Kiwi
7. Jaca
8. Manga
 1. Espada
 2. Carlotinha
9. Maracujá
10. Goiaba
11. Abacaxi
12. Limão
13. Laranja
14. Pêra
15. Maçã

Observação:

Na caixa de listagem ao lado foi utilizado JavaScript, por isso existe uma iteração com as mudanças.

Usando Algoritmos romanos com letras em minúsculo (alterado dinamicamente com javascript, pela caixa de seleção):

Trabalho de Desenvolvimento WEB - Google Chrome

Trabalho de Desenvolvimento x +

Arquivo | /home/leonardo/Projects/aulas-faculdade-web/trabalho-aula-1-e-2/index.html

Lista Ordenada de Frutas

Selecione como deseja ordenar a lista:

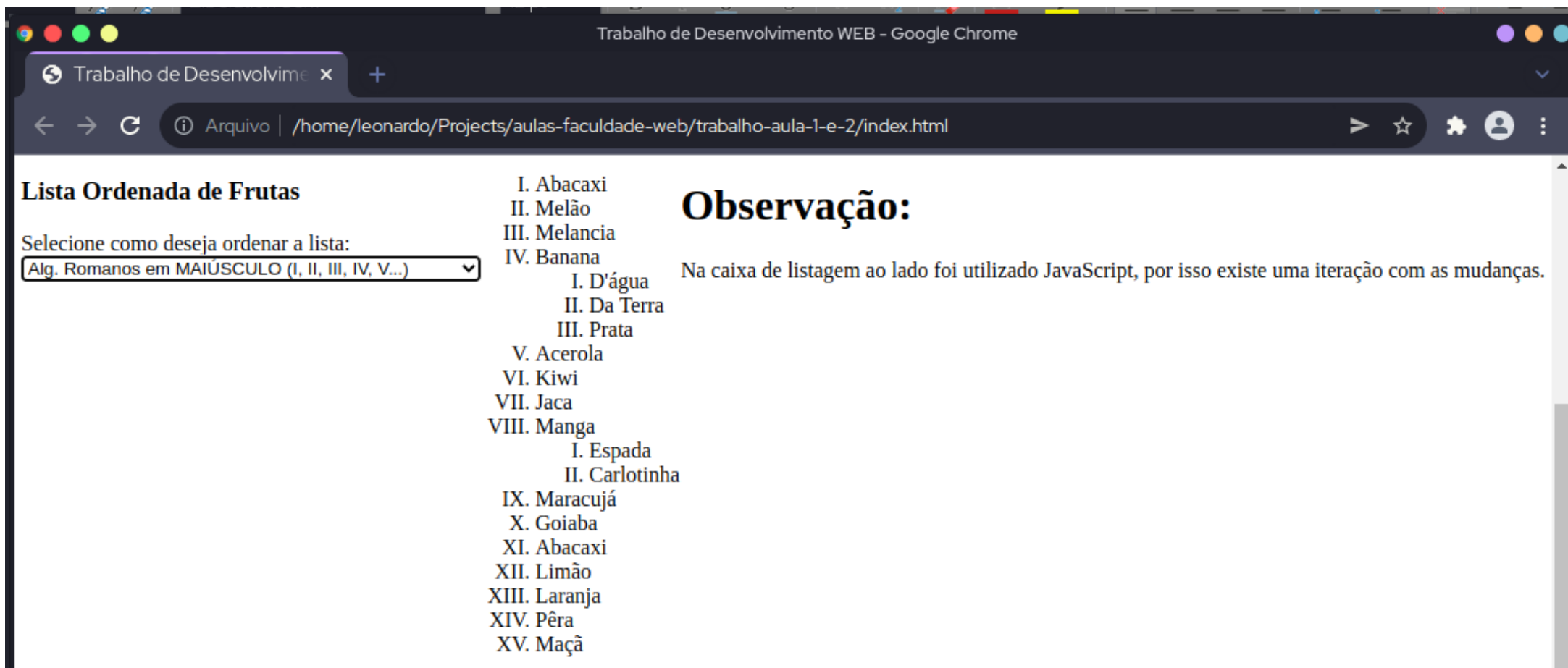
Alg. Romanos em minúsculo (i, ii, iii, iv, v...) ▼

- i. Abacaxi
- ii. Melão
- iii. Melancia
- iv. Banana
 - i. D'água
 - ii. Da Terra
 - iii. Prata
- v. Acerola
- vi. Kiwi
- vii. Jaca
- viii. Manga
 - i. Espada
 - ii. Carlotinha
- ix. Maracujá
- x. Goiaba
- xi. Abacaxi
- xii. Limão
- xiii. Laranja
- xiv. Pêra
- xv. Maçã

Observação:

Na caixa de listagem ao lado foi utilizado JavaScript, por isso existe uma iteração com as mudanças.

Usando Algoritmos romanos com letras em MAIÚSCULO (alterado dinamicamente com javascript, pela caixa de seleção):



Usando ordem alfabética em minúsculo (alterado dinamicamente com javascript, pela caixa de seleção):

Trabalho de Desenvolvimento WEB - Google Chrome

Trabalho de Desenvolvimento x +

Arquivo | /home/leonardo/Projects/aulas-faculdade-web/trabalho-aula-1-e-2/index.html

Lista Ordenada de Frutas

Selecione como deseja ordenar a lista:

Ordem alfabética em minúsculo (a, b, c, d, e, f...) ▼

- a. Abacaxi
- b. Melão
- c. Melancia
- d. Banana
 - a. D'água
 - b. Da Terra
 - c. Prata
- e. Acerola
- f. Kiwi
- g. Jaca
- h. Manga
 - a. Espada
 - b. Carlotinha
- i. Maracujá
- j. Goiaba
- k. Abacaxi
- l. Limão
- m. Laranja
- n. Pêra
- o. Maçã

Observação:

Na caixa de listagem ao lado foi utilizado JavaScript, por isso existe uma iteração com as mudanças.

Usando ordem alfabética em minúsculo (alterado dinamicamente com javascript, pela caixa de seleção):

Trabalho de Desenvolvimento WEB - Google Chrome

Trabalho de Desenvolvimento x +

Arquivo | /home/leonardo/Projects/aulas-faculdade-web/trabalho-aula-1-e-2/index.html

☆

Lista Ordenada de Frutas

Selecione como deseja ordenar a lista:

Ordem alfabética em MAIÚSCULO (A, B, C, D, E, F...) ▼

A. Abacaxi

B. Melão

C. Melancia

D. Banana

A. D'água

B. Da Terra

C. Prata

E. Acerola

F. Kiwi

G. Jaca

H. Manga

A. Espada

B. Carlotinha

I. Maracujá

J. Goiaba

K. Abacaxi

L. Limão

M. Laranja

N. Pêra

O. Maçã

Observação:

Na caixa de listagem ao lado foi utilizado JavaScript, por isso existe uma iteração com as mudanças.

O Código:

```

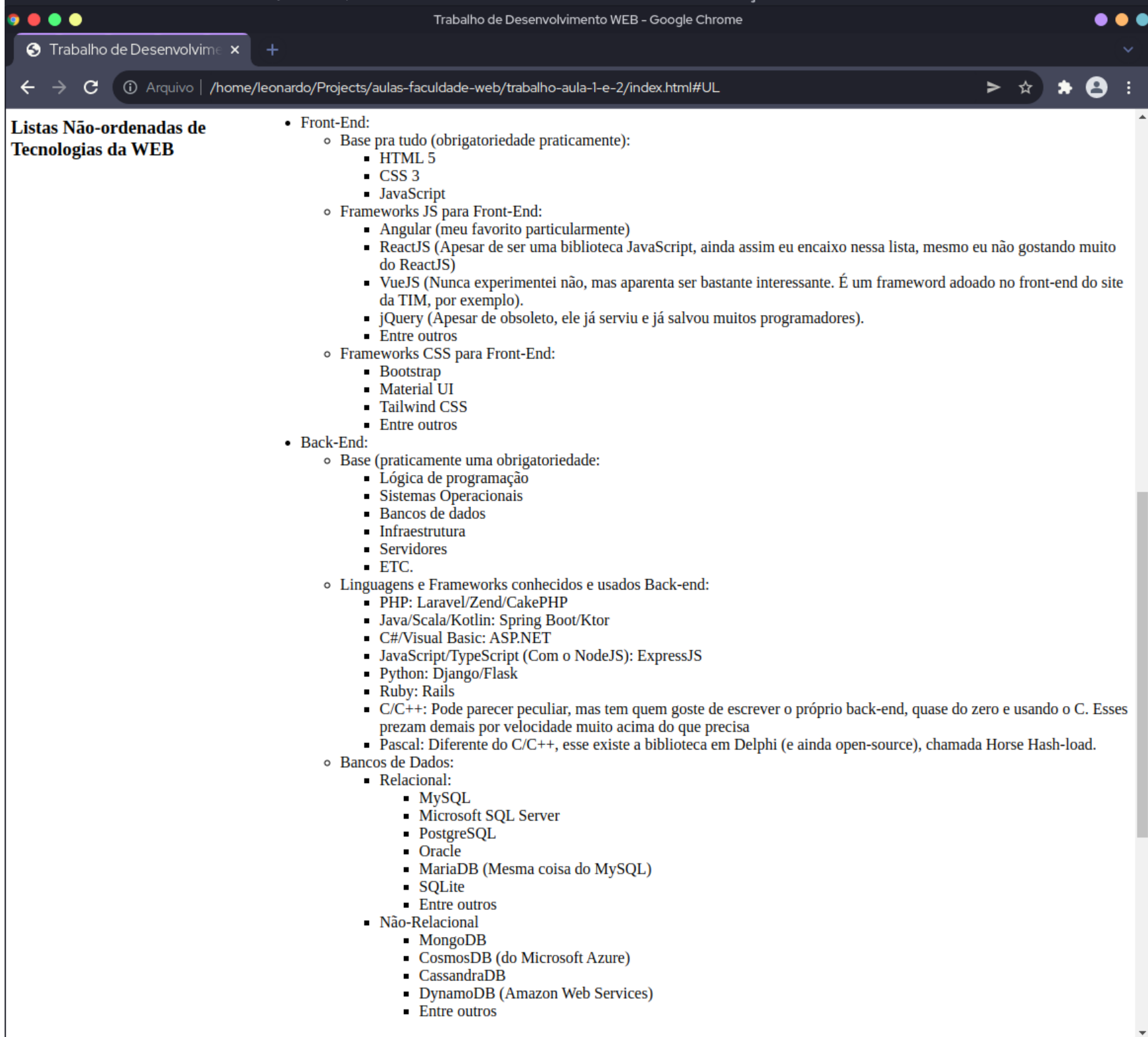
<!-- Lista ordenada -->
<div id="OL" class="resp-container">
  <div class="resp-item">
    <h3>Lista Ordenada de Frutas</h3>
    <label for="selForOrderList">Selecione como deseja ordenar a lista:</label><br>
    <select name="selForOrderList" id="selForOrderList" onchange="changeOrdenateList(this.value)">
      <option value="1">Ordem numérica (1, 2, 3, 4, 5)</option>
      <option value="i">Alg. Romanos em minúsculo (i, ii, iii, iv, v...)</option>
      <option value="I">Alg. Romanos em MAIÚSCULO (I, II, III, IV, V...)</option>
      <option value="a">Ordem alfabética em minúsculo (a, b, c, d, e, f...)</option>
      <option value="A">Ordem alfabética em MAIÚSCULO (A, B, C, D, E, F...)</option>
    </select>
  </div>
  <ol class="ordenateList resp-item" type="1">
    <li>Abacaxi</li>
    <li>Melão</li>
    <li>Melancia</li>
    <li>Banana
      <ol class="ordenateList" type="1">
        <li>D'água</li>
        <li>Da Terra</li>
        <li>Prata</li>
      </ol>
    </li>
    <li>Acerola</li>
    <li>Kiwi</li>
    <li>Jaca</li>
    <li>Manga
      <ol class="ordenateList" type="1">
        <li>Espada</li>
        <li>Carlotinha</li>
      </ol>
    </li>
    <li>Maracujá</li>
    <li>Goiaba</li>
    <li>Abacaxi</li>
    <li>Limão</li>
    <li>Laranja</li>
    <li>Pêra</li>
    <li>Maçã</li>
  </ol>

```

```
<div class="resp-item">
  <h1>Observação:</h1>
  <p>Na caixa de listagem ao lado foi utilizado JavaScript, por isso existe uma iteração com as mudanças.</p>
</div>
</div>
```

Lista não Ordenada:

O print ao lado:



The screenshot shows a web browser window with the title "Trabalho de Desenvolvimento WEB - Google Chrome". The address bar shows the URL: `/home/leonardo/Projects/aulas-faculdade-web/trabalho-aula-1-e-2/index.html#UL`. The page content is titled "Listas Não-ordenadas de Tecnologias da WEB" and contains a list of technologies categorized into Front-End and Back-End.

Listas Não-ordenadas de Tecnologias da WEB

- Front-End:
 - Base pra tudo (obrigatoriedade praticamente):
 - HTML 5
 - CSS 3
 - JavaScript
 - Frameworks JS para Front-End:
 - Angular (meu favorito particularmente)
 - ReactJS (Apesar de ser uma biblioteca JavaScript, ainda assim eu encaixo nessa lista, mesmo eu não gostando muito do ReactJS)
 - VueJS (Nunca experimentei não, mas aparenta ser bastante interessante. É um framework adorado no front-end do site da TIM, por exemplo).
 - jQuery (Apesar de obsoleto, ele já serviu e já salvou muitos programadores).
 - Entre outros
 - Frameworks CSS para Front-End:
 - Bootstrap
 - Material UI
 - Tailwind CSS
 - Entre outros
- Back-End:
 - Base (praticamente uma obrigatoriedade):
 - Lógica de programação
 - Sistemas Operacionais
 - Bancos de dados
 - Infraestrutura
 - Servidores
 - ETC.
 - Linguagens e Frameworks conhecidos e usados Back-end:
 - PHP: Laravel/Zend/CakePHP
 - Java/Scala/Kotlin: Spring Boot/Ktor
 - C#/Visual Basic: ASP.NET
 - JavaScript/TypeScript (Com o NodeJS): ExpressJS
 - Python: Django/Flask
 - Ruby: Rails
 - C/C++: Pode parecer peculiar, mas tem quem goste de escrever o próprio back-end, quase do zero e usando o C. Esses prezam demais por velocidade muito acima do que precisa
 - Pascal: Diferente do C/C++, esse existe a biblioteca em Delphi (e ainda open-source), chamada Horse Hash-load.
 - Bancos de Dados:
 - Relacional:
 - MySQL
 - Microsoft SQL Server
 - PostgreSQL
 - Oracle
 - MariaDB (Mesma coisa do MySQL)
 - SQLite
 - Entre outros
 - Não-Relacional
 - MongoDB
 - CosmosDB (do Microsoft Azure)
 - CassandraDB
 - DynamoDB (Amazon Web Services)
 - Entre outros

O Código:

```
<!-- Lista não ordenada -->
<div class="resp-container" id="UL">
  <h3>Listas Não-ordenadas de Tecnologias da WEB</h3>
  <ul>
    <li>Front-End:
      <ul>
        <li>Base pra tudo (obrigatoriedade praticamente):
          <ul>
            <li>HTML 5</li>
            <li>CSS 3</li>
            <li>JavaScript</li>
          </ul>
        </li>
        <li>Frameworks JS para Front-End:
          <ul>
            <li>Angular (meu favorito particularmente)</li>
            <li>ReactJS (Apesar de ser uma biblioteca JavaScript, ainda assim eu encaixo nessa lista, mesmo eu não gostando
muito do ReactJS)</li>
            <li>VueJS (Nunca experimentei não, mas aparenta ser bastante interessante. É um frameword adoado no front-end
do site da TIM, por exemplo).</li>
            <li>jQuery (Apesar de obsoleto, ele já serviu e já salvou muitos programadores).</li>
            <li>Entre outros</li>
          </ul>
        </li>
        <li>Frameworks CSS para Front-End:
          <ul>
            <li>Bootstrap</li>
            <li>Material UI</li>
            <li>Tailwind CSS</li>
            <li>Entre outros</li>
          </ul>
        </li>
      </ul>
    </li>
    <li>Back-End:
      <ul>
        <li>Base (praticamente uma obrigatoriedade:
          <ul>
            <li>Lógica de programação</li>
            <li>Sistemas Operacionais</li>
            <li>Bancos de dados</li>
          </ul>
        </li>
      </ul>
    </li>
  </ul>
</div>
```

- Infraestrutura
 - Servidores
 - ETC.

Linguagens e Frameworks conhecidos e usados Back-end:

- - PHP: Laravel/Zend/CakePHP
- Java/Scala/Kotlin: Spring Boot/Ktor
- C#/Visual Basic: ASP.NET
- JavaScript/TypeScript (Com o NodeJS): ExpressJS
- Python: Django/Flask
- Ruby: Rails
- C/C++: Pode parecer peculiar, mas tem quem goste de escrever o próprio back-end, quase do zero e usando o C. Esses prezam demais por velocidade muito acima do que precisa
- Pascal: Diferente do C/C++, esse existe a biblioteca em Delphi (e ainda open-source), chamada Horse Hash-Load.

-
 -
 - Bancos de Dados:
 -
 - Relacional:
 -
 - MySQL
 - Microsoft SQL Server
 - PostgreSQL
 - Oracle
 - MariaDB (Mesma coisa do MySQL)
 - SQLite
 - Entre outros
 -
 -
 - Não-Relacional
 -
 - MongoDB
 - CosmosDB (do Microsoft Azure)
 - CassandraDB
 - DynamoDB (Amazon Web Services)
 - Entre outros
-
-
-

```
</li>
</ul>
</li>
</ul>
</div>
```


Tabela:

O print ao lado:

Trabalho de Desenvolvimento WEB - Google Chrome

Trabalho de Desenvolvimento x +

Arquivo | /home/leonardo/Projects/aulas-faculdade-web/trabalho-aula-1-e-2/index.html#UL

Tabela (Nesse exemplo foi utilizado javascript para preencher linha por linha)

Sobre o JavaScript, foi criado um objeto (uma variável) genérica para simular uma chamada de API, de interceptação de dados.

Limpezeria casa de limpeza pesadas LTDA.			Nº nota:
Rua das Margaridas 1350 - Centro Cidade Maravilhosa - Estado Perigoso - Brasil			
Qtde	Produto ou Descrição	R\$/UN ou R\$/KG	Vlr Total
6	Galão de Cloro 5 litros	R\$ 3,52	R\$ 21,12
10	Esponja abrasiva pacote c/ 1 und	R\$ 2,25	R\$ 22,50
15	Caixas de Detergente Líquido 200ml c/ 20und	R\$ 25,87	R\$ 388,05
12	Desengordurante galão 5 litros	R\$ 45,03	R\$ 540,36
300	Esponjas amarela pct c/ 1 und	R\$ 0,54	R\$ 162,00
45	Refis de esfregão	R\$ 8,84	R\$ 397,80
7	Sabão em pó 5kg caixa	R\$ 9,52	R\$ 66,64
3	kit MOP (balde + torcedor de esfregão)	R\$ 280,06	R\$ 840,18
8	Vassouras comum	R\$ 6,33	R\$ 50,64
2	Vassourão para calçada	R\$ 9,68	R\$ 19,36
5	Rodo	R\$ 6,33	R\$ 31,65
15	Cabos de esfregão	R\$ 14,98	R\$ 224,70
5	Galão de limpa-vidro 5 litros	R\$ 6,51	R\$ 32,55
20	Alcool gel caixa c/ 10 und	R\$ 40,64	R\$ 812,80
100	rolos papel higienico 100mts	R\$ 7,63	R\$ 763,00
250	papel toalha seca mão pct c/ 500fls	R\$ 7,12	R\$ 1.780,00
Subtotal:			R\$ 6.153,35

Uma observação MUITO IMPORTANTE: As tags <tbody> e </tbody> estão vazias, sem um elemento sequer, por causa que é o JavaScript quem faz a inserção dinâmica de cada uma das linhas. Na abertura da tag <body>, ela acabou ficando <body onload="loadDataInTable();"> justamente para carregar o preenchimento dinâmico das linhas da tabela, dentro da tag <tbody></tbody>. Por isso que não existe conteúdo entre as tags <tbody> </tbody>. É feito de forma automatizada, dinâmica. Na unica tag <td></td> dentro de <tfoot></tfoot>, também é preenchido automaticamente usando JavaScript, como no exemplo do print acima, foi preenchido o somatório correto do valor total de item a item.

O código:

```
←!— Tabela →
<div id="tabelas" class="resp-container">
  <div class="resp-item">
    <h3>Tabela (Nesse exemplo foi utilizado javascript para preencher linha por linha)</h3>
    <p>Sobre o JavaScript, foi criado um objeto (uma variável) genérica para simular uma chamada de API, de interceptação de
dados.</p>
    <table cellpadding="0" border="1" width="100%" cellspacing="0">
      <thead>
        <tr>
          <th colspan="3" align="center">Limpezeria casa de limpezas pesadas LTDA.</th>
          <td rowspan="2">Nº nota:</td>
        </tr>
        <tr>
          <td colspan="3" align="center">Rua das Margaridas 1350 - Centro <br>
Cidade Maravilhosa - Estado Perigoso - Brasil</td>
        </tr>
        <tr>
          <th>Qtde</th>
          <th>Produto ou Descrição</th>
          <th>R$/UN ou R$/KG</th>
          <th>Vlr Total</th>
        </tr>
      </thead>
      <!-- O tbody está vazio, sem dados nenhum, porque o javascript já faz todos os trabalho de preenchimento. -->
      <tbody></tbody>
      <tfoot>
        <tr>
          <th colspan="3" align="right">Subtotal:</th>
          <td></td>
        </tr>
      </tfoot>
    </table>
  </div>
</div>
```

Concluído a apresentação!

Segue todo código por completo abaixo:

O lang="pt-br" é para indicar ao navegador que a página está em português do Brasil e não ter necessidade de tradutor (mais em específico navegadores com motor chromium, como o Google Chrome, Microsoft Edge, Brave, novas versões do Opera, entre outros.)

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Trabalho de Desenvolvimento WEB</title>
  <script type="application/javascript" src="./scripts.js"></script>
  <link rel="stylesheet" href="./estilo.css" type="text/css">
  <meta name="viewport" content="width=device-width,initial-scale=1.0">
</head>
<body onload="loadDataInTable()">
  <h1>Trabalho de HTML, CSS, JavaScrip e PHP</h1>
  <h2>Sumário (Aqui está sendo usado Lista de Definições)</h2>
  <!-- Lista de Definições -->
  <dl>
    <dt><a href="#apresentacao">Sobre</a></dt>
    <dd>Uso de imagens</dd>
    <dd>Uso de parágrafos</dd>
    <dt>Listas:</dt>
    <dd><a href="#0L">Ordenadas usando OL</a></dd>
    <dd><a href="#UL">Não-ordenadas usando UL</a></dd>
    <dt><a href="#tabelas">Tabelas</a></dt>
    <dd>Uma apresentação de nota de orçamento de uma loja de limpezas</dd>
  </dl>

  <!-- Imagens, parágrafos, H3, etc. -->
  <div id="apresentacao" class="resp-container">
    <div class="resp-item">
      <h3>Sobre</h3>
      <p>ESTÁCIO</p>
      <p>Desenvolvimento em HTML5, CSS3, JavaScript e PHP &nbsp; &nbsp;</p>
      <p>Aluno: Leonardo Nicolas Sales Dias</p>
      <p>Professor: Ediberto Mariano</p>
      <p>Usados neste trabalho:</p>
      <p>
        
      </p>
    </div>
  </div>
</body>
</html>
```

```

        
        
        
        
    </p>
    <p>Sistema Operacional: Fedora 35</p>
    <p>IDE (nesse caso é IDE mesmo): JetBrains WebStorm</p>
    <p>Estrutura desta página: HTML 5</p>
    <p>Cores e a foto ao lado direito: CSS 3</p>
    <p>Interações com esta página: JavaScript</p>
</div>
<!-- No browser, a foto aparece oval, por causa do CSS. -->
<div class="resp-item">
    
</div>
</div>

<!-- Lista ordenada -->
<div id="OL" class="resp-container">
    <div class="resp-item">
        <h3>Lista Ordenada de Frutas</h3>
        <label for="selForOrderList">Selecione como deseja ordenar a lista:</label><br>
        <select name="selForOrderList" id="selForOrderList" onchange="changeOrdenateList(this.value)">
            <option value="1">Ordem numérica (1, 2, 3, 4, 5)</option>
            <option value="i">Alg. Romanos em minúsculo (i, ii, iii, iv, v...)</option>
            <option value="I">Alg. Romanos em MAIÚSCULO (I, II, III, IV, V...)</option>
            <option value="a">Ordem alfabética em minúsculo (a, b, c, d, e, f...)</option>
            <option value="A">Ordem alfabética em MAIÚSCULO (A, B, C, D, E, F...)</option>
        </select>
    </div>
    <ol class="ordenateList resp-item" type="1">
        <li>Abacaxi</li>
        <li>Melão</li>
        <li>Melancia</li>
        <li>Banana
            <ol class="ordenateList" type="1">
                <li>D'água</li>
                <li>Da Terra</li>
                <li>Prata</li>
            </ol>
        </li>
    </ol>

```

```

    </li>
    <li>Acerola</li>
    <li>Kiwi</li>
    <li>Jaca</li>
    <li>Manga
      <ol class="ordenateList" type="1">
        <li>Espada</li>
        <li>Carlotinha</li>
      </ol>
    </li>
    <li>Maracujá</li>
    <li>Goiaba</li>
    <li>Abacaxi</li>
    <li>Limão</li>
    <li>Laranja</li>
    <li>Pêra</li>
    <li>Maçã</li>
  </ol>

  <div class="resp-item">
    <h1>Observação:</h1>
    <p>Na caixa de listagem ao lado foi utilizado JavaScript, por isso existe uma iteração com as mudanças.</p>
  </div>
</div>

<!-- Lista não ordenada -->
<div class="resp-container" id="UL">
  <h3>Listas Não-ordenadas de Tecnologias da WEB</h3>
  <ul>
    <li>Front-End:
      <ul>
        <li>Base pra tudo (obrigatoriedade praticamente):
          <ul>
            <li>HTML 5</li>
            <li>CSS 3</li>
            <li>JavaScript</li>
          </ul>
        </li>
        <li>Frameworks JS para Front-End:
          <ul>
            <li>Angular (meu favorito particularmente)</li>
            <li>ReactJS (Apesar de ser uma biblioteca JavaScript, ainda assim eu encaixo nessa lista, mesmo eu não

```

```
gostando muito do ReactJS)</li>
    <li>VueJS (Nunca experimentei não, mas aparenta ser bastante interessante. É um frameword adoado no front-
end do site da TIM, por exemplo).</li>
    <li>jQuery (Apesar de obsoleto, ele já serviu e já salvou muitos programadores).</li>
    <li>Entre outros</li>
</ul>
</li>
<li>Frameworks CSS para Front-End:
    <ul>
        <li>Bootstrap</li>
        <li>Material UI</li>
        <li>Tailwind CSS</li>
        <li>Entre outros</li>
    </ul>
</li>
</ul>
</li>
<li>Back-End:
    <ul>
        <li>Base (praticamente uma obrigatoriedade:
            <ul>
                <li>Lógica de programação</li>
                <li>Sistemas Operacionais</li>
                <li>Bancos de dados</li>
                <li>Infraestrutura</li>
                <li>Servidores</li>
                <li>ETC.</li>
            </ul>
        </li>
        <li>Linguagens e Frameworks conhecidos e usados Back-end:
            <ul>
                <li>PHP: Laravel/Zend/CakePHP</li>
                <li>Java/Scala/Kotlin: Spring Boot/Ktor</li>
                <li>C#/Visual Basic: ASP.NET</li>
                <li>JavaScript/TypeScript (Com o NodeJS): ExpressJS</li>
                <li>Python: Django/Flask</li>
                <li>Ruby: Rails</li>
                <li>C/C++: Pode parecer peculiar, mas tem quem goste de escrever o próprio back-end, quase do zero e usando
o C. Esses prezam demais por velocidade muito acima do que precisa </li>
                <li>Pascal: Diferente do C/C++, esse existe a biblioteca em Delphi (e ainda open-source), chamada Horse
Hash-load.</li>
            </ul>
        </li>
    </ul>
</li>
```

```

    </li>
    <li>Bancos de Dados:
      <ul>
        <li>Relacional:
          <ul>
            <li>MySQL</li>
            <li>Microsoft SQL Server</li>
            <li>PostgreSQL</li>
            <li>Oracle</li>
            <li>MariaDB (Mesma coisa do MySQL)</li>
            <li>SQLite</li>
            <li>Entre outros</li>
          </ul>
        </li>
        <li>Não-Relacional
          <ul>
            <li>MongoDB</li>
            <li>CosmosDB (do Microsoft Azure)</li>
            <li>CassandraDB</li>
            <li>DynamoDB (Amazon Web Services)</li>
            <li>Entre outros</li>
          </ul>
        </li>
      </ul>
    </li>
  </ul>
</div>

```

←!— Tabela →

```
<div id="tabelas" class="resp-container">
```

```
  <div class="resp-item">
```

```
    <h3>Tabela (Nesse exemplo foi utilizado javascript para preencher linha por linha)</h3>
```

```
    <p>Sobre o JavaScript, foi criado um objeto (uma variável) genérica para simular uma chamada de API, de interceptação de dados.</p>
```

```
    <table cellpadding="0" border="1" width="100%" cellspacing="0">
```

```
      <thead>
```

```
        <tr>
```

```
          <th colspan="3" align="center">Limpezeria casa de limpezas pesadas LTDA.</th>
```

```
          <td rowspan="2">Nº nota:</td>
```

```
        </tr>
```

```
<tr>
  <td colspan="3" align="center">Rua das Margaridas 1350 - Centro <br>
    Cidade Maravilhosa - Estado Perigoso - Brasil</td>
```

```
</tr>
```

```
<tr>
```

```
  <th>Qtde</th>
```

```
  <th>Produto ou Descrição</th>
```

```
  <th>R$/UN ou R$/KG</th>
```

```
  <th>Vlr Total</th>
```

```
</tr>
```

```
</thead>
```

```
<!-- O tbody está vazio, sem dados nenhum, porque o javascript já faz todos os trabalho de preenchimento. -->
```

```
<tbody></tbody>
```

```
<tfoot>
```

```
  <tr>
```

```
    <th colspan="3" align="right">Subtotal:</th>
```

```
    <td></td>
```

```
  </tr>
```

```
</tfoot>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```