

Materia: Computación Tolerante a Fallas.

NRC: 179961

Maestro: Lopez Franco, Michael Emanuel

Aula: X-02

Sección: D06

Alumno: Zashuvath López Moreno, Ethan Israel

Código: 216493953



Índice

Introducción	3
Objetivo:	3
Contenido.....	3
Código HTML	3
Dockerfile	4
Codigo JS	4
Código CSS.....	5
Funcionamiento	5
Resultados:.....	6
Conclusión	6

Introducción

Después de la explicación dada en clases y una investigación del funcionamiento utilizaremos Docker para crear un servicio o aplicación.

Objetivo:

Realizar un servicio o aplicación con Docker.

Contenido

Código HTML

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Calculadora</title>
5 <style>
6   #calculator {
7     width: 240px;
8     margin: 0 auto;
9     padding: 20px;
10    border: 1px solid #ccc;
11    border-radius: 5px;
12    background-color: #f0f0f0;
13    text-align: center;
14  }
15
16  .button {
17    width: 60px;
18    height: 60px;
19    margin: 5px;
20    font-size: 18px;
21    background-color: #e0e0e0;
22    border: 1px solid #ccc;
23    border-radius: 5px;
24    cursor: pointer;
25  }
26
27  .button:hover {
28    background-color: #ddd;
29  }
30
31  .button.operator {
32    background-color: #ff8800;
33    color: white;
34  }
35
36  .button.operator-vertical {
37    width: 60px;
38    height: 60px;
39    margin: 5px;
```

```
40    font-size: 18px;
41    background-color: #ff8800;
42    border: 1px solid #ccc;
43    border-radius: 5px;
44    color: white;
45    cursor: pointer;
46  }
47
48  #display {
49    width: 100%;
50    height: 40px;
51    font-size: 20px;
52    margin-bottom: 10px;
53    text-align: right;
54  }
55
56  .row {
57    display: flex;
58    justify-content: center;
59  }
60 </style>
61 </head>
62 <body>
63   <div id="calculator">
64     <input type="text" id="display" readonly>
65     <div class="row">
66       <button class="button operator" onclick="clearDisplay()">C</button>
67       <button class="button operator-vertical" onclick="appendToDisplay('+')">+</button>
68       <button class="button operator-vertical" onclick="appendToDisplay('-')">-</button>
69       <button class="button operator" onclick="appendToDisplay('*')">*</button>
70       <button class="button operator" onclick="appendToDisplay('/')">/</button>
71     </div>
72     <div class="row">
73       <button class="button" onclick="appendToDisplay('7')">7</button>
74       <button class="button" onclick="appendToDisplay('8')">8</button>
75       <button class="button" onclick="appendToDisplay('9')">9</button>
76     </div>
77     <div class="row">
78       <button class="button" onclick="appendToDisplay('4')">4</button>
```

```

79     <button class="button" onclick="appendToDisplay('5')>5</button>
80     <button class="button" onclick="appendToDisplay('6')>6</button>
81 </div>
82 <div class="row">
83     <button class="button" onclick="appendToDisplay('1')>1</button>
84     <button class="button" onclick="appendToDisplay('2')>2</button>
85     <button class="button" onclick="appendToDisplay('3')>3</button>
86 </div>
87 <div class="row">
88     <button class="button operator-vertical" onclick="appendToDisplay('(')>(</button>
89     <button class="button operator-vertical" onclick="appendToDisplay(')>)</button>
90     <button class="button" onclick="appendToDisplay('0')>0</button>
91     <button class="button" onclick="appendToDisplay('.')>.</button>
92     <button class="button operator" onclick="calculateResult()">=</button>
93 </div>
94 </div>
95
96 <script>
97     function appendToDisplay(value) {
98         document.getElementById('display').value += value;
99     }
100
101     function clearDisplay() {
102         document.getElementById('display').value = '';
103     }
104
105     function calculateResult() {
106         const expression = document.getElementById('display').value;
107         try {
108             const result = eval(expression);
109             document.getElementById('display').value = result.toFixed(2); // Limitar a 2 decimales
110         } catch (error) {
111             document.getElementById('display').value = 'Error';
112         }
113     }
114 </script>
115 </body>
116 </html>
117

```

Dockerfile

```

Dockerfile > ...
1 # Utiliza una imagen base con un servidor web para aplicaciones estáticas
2 FROM nginx:alpine
3
4 # Copia los archivos de la aplicación al directorio de trabajo en el contenedor
5 COPY ./ /usr/share/nginx/html/
6
7 # Expone el puerto 80 para el servidor web
8 EXPOSE 80

```

Código JS

```

1 function appendToDisplay(value) {
2     document.getElementById('display').value += value;
3 }
4
5 function clearDisplay() {
6     document.getElementById('display').value = '';
7 }
8
9 function calculateResult() {
10     const expression = document.getElementById('display').value;
11     try {
12         const result = eval(expression);
13         document.getElementById('display').value = result.toFixed(2); // Limitar a 2 decimales
14     } catch (error) {
15         document.getElementById('display').value = 'Error';
16     }
17 }
18
19 function calculateSquareRoot() {
20     const number = parseFloat(document.getElementById('display').value);
21     if (number >= 0) {
22         const result = Math.sqrt(number).toFixed(2);
23         document.getElementById('display').value = result;
24     } else {
25         document.getElementById('display').value = 'Error';
26     }
27 }
28
29 function calculateSine() {
30     const number = parseFloat(document.getElementById('display').value);
31     const result = Math.sin(number * (Math.PI / 180)).toFixed(2);
32     document.getElementById('display').value = result;
33 }
34

```

```

35 function calculateCosine() {
36     const number = parseFloat(document.getElementById('display').value);
37     const result = Math.cos(number * (Math.PI / 180)).toFixed(2);
38     document.getElementById('display').value = result;
39 }
40
41 function calculateNaturalLog() {
42     const number = parseFloat(document.getElementById('display').value);
43     if (number > 0) {
44         const result = Math.log(number).toFixed(2);
45         document.getElementById('display').value = result;
46     } else {
47         document.getElementById('display').value = 'Error';
48     }
49 }
50

```

Código CSS

```
style.css > .display
1  /* Estilos generales para la calculadora */
2  body {
3      font-family: Arial, sans-serif;
4      background-color: #f4f4f4;
5      display: flex;
6      justify-content: center;
7      align-items: center;
8      height: 100vh;
9      margin: 0;
10 }
11
12 #calculator {
13     width: 320px;
14     background-color: #f4f4f4;
15     border: 1px solid #ccc;
16     border-radius: 10px;
17     box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
18 }
19
20 .display {
21     width: 100%;
22     height: 60px;
23     text-align: right;
24     font-size: 24px;
25     border: none;
26     outline: none;
27     padding: 10px;
28 }
29
30 .row {
31     display: flex;
32     justify-content: space-between;
33 }
34
35 .button {
36     width: 25%;
37     height: 70px;
38     font-size: 24px;
39     border: none;
```

```
40     outline: none;
41     background-color: #f0f0f0;
42     cursor: pointer;
43     transition: background-color 0.2s;
44 }
45
46 .button:hover {
47     background-color: #d0d0d0;
48 }
49
```

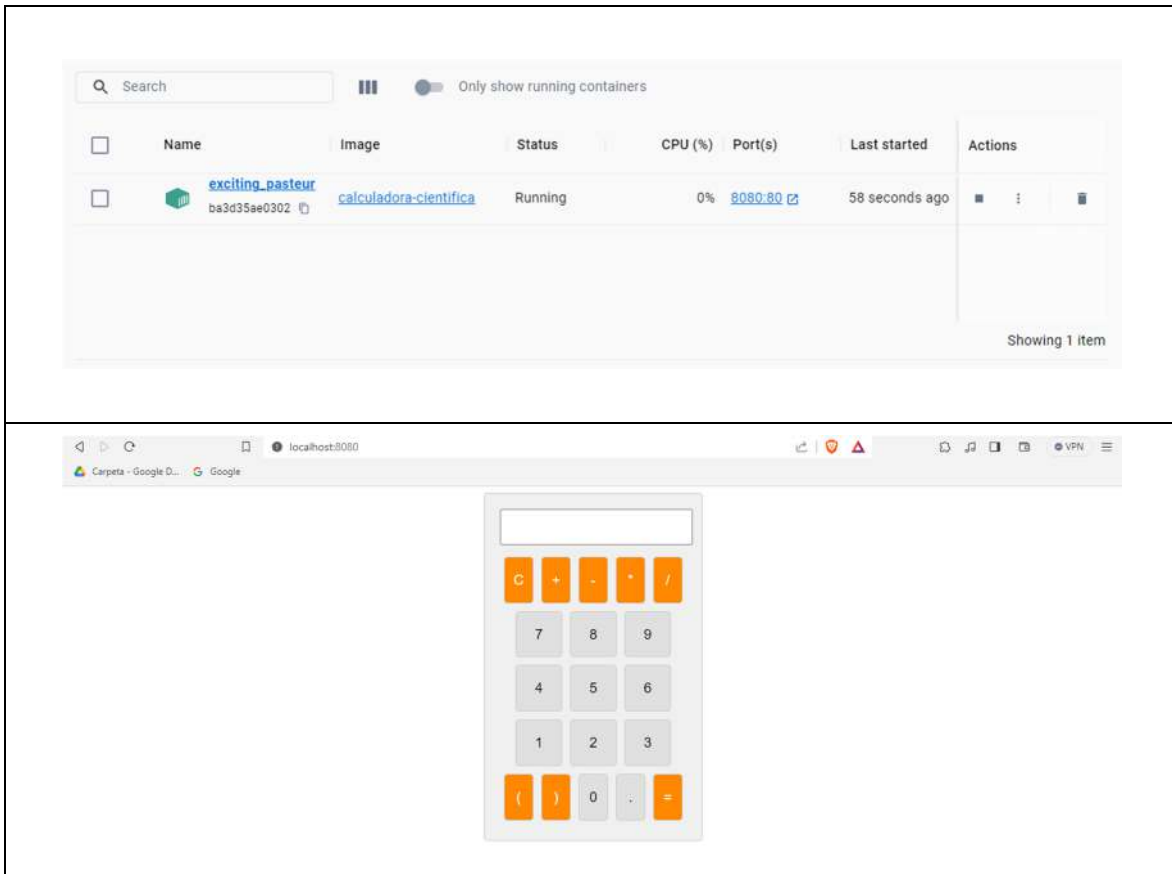
Funcionamiento

Para que funcione el programa debemos hacer lo siguiente:

1. Debemos crear en la misma carpeta los archivos HTML, JS, CSS y el archivo Docker, este último sin extensión.
2. Una vez elaborados los archivos abriremos el CMD y buscaremos la dirección en la que tenemos guardada la carpeta.
3. Debemos abrir el Docker Desktop para seguir con el paso 4
4. Cuando el CMD tiene como dirección la ubicación de la carpeta debemos insertar el siguiente comando **"docker build -t calculator-app ."**
5. El comando tiene el siguiente significado;
 - a. **"docker build"** es el comando para construir una imagen de Docker.
 - b. **"-t calculator-app"** le da un nombre a la imagen, en este caso, la llamamos calculator-app.
 - c. **."** indica que el archivo Dockerfile se encuentra en el directorio actual.

6. Para comprobar que funcione el comando insertamos **“docker images”**

Resultados:



Conclusión

La implementación de Docker fue más sencilla de lo que supuse, afortunadamente logre terminar la aplicación en tiempo y forma, lo más complicado realmente fue la interfaz para que los usuarios pudieran llevar a cabo las operaciones básica.