

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
Código JS frontend.....	5
Código JS backend	6
Resultados:.....	7
Conclusión	7

Introducción

Después de la explicación dada en clases y una investigación del funcionamiento utilizaremos Docker para crear una aplicación utilizando microservicios.

Objetivo:

Genera una aplicación utilizando microservicios

Contenido

Código HTML

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Calculadora</title>
7      <style>
8          body {
9              display: flex;
10             align-items: center;
11             justify-content: center;
12             height: 100vh;
13             margin: 0;
14             font-family: 'Arial', sans-serif;
15         }
16
17         .calculator {
18             width: 300px;
19             background-color: #f4f4f4;
20             padding: 20px;
21             border-radius: 10px;
22             box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
23         }
24
25         #display {
26             width: 100%;
27             height: 40px;
28             margin-bottom: 10px;
29             font-size: 1.5em;
30             text-align: right;
31             border: none;
32             outline: none;
33             padding: 5px;
34             box-sizing: border-box;
35         }
36     </style>
37 </head>
38 <body>
39     <div class="calculator">
40         <input type="text" id="display" readonly>
41     </div>
42     <script src="script.js"></script>
43 </body>
44 </html>
```

Dockerfile

```
1  # Dockerfile
2  FROM node:14
3
4  WORKDIR /app/backend
5
6  COPY backend/package*.json .
7  RUN npm install
8
9  COPY backend .
10
11 EXPOSE 3000
12
13 CMD ["node", "app.js"]
14
```

docker-compose.yml

```
1  version: '3'
2  services:
3  backend:
4  build:
5    context: .
6    dockerfile: Dockerfile
7  ports:
8    - "3000:3000"
9  volumes:
10    - ./backend:/app/backend
11
12 frontend:
13 image: nginx
14 ports:
15   - "80:80"
16 volumes:
17   - ./frontend:/usr/share/nginx/html
18
```

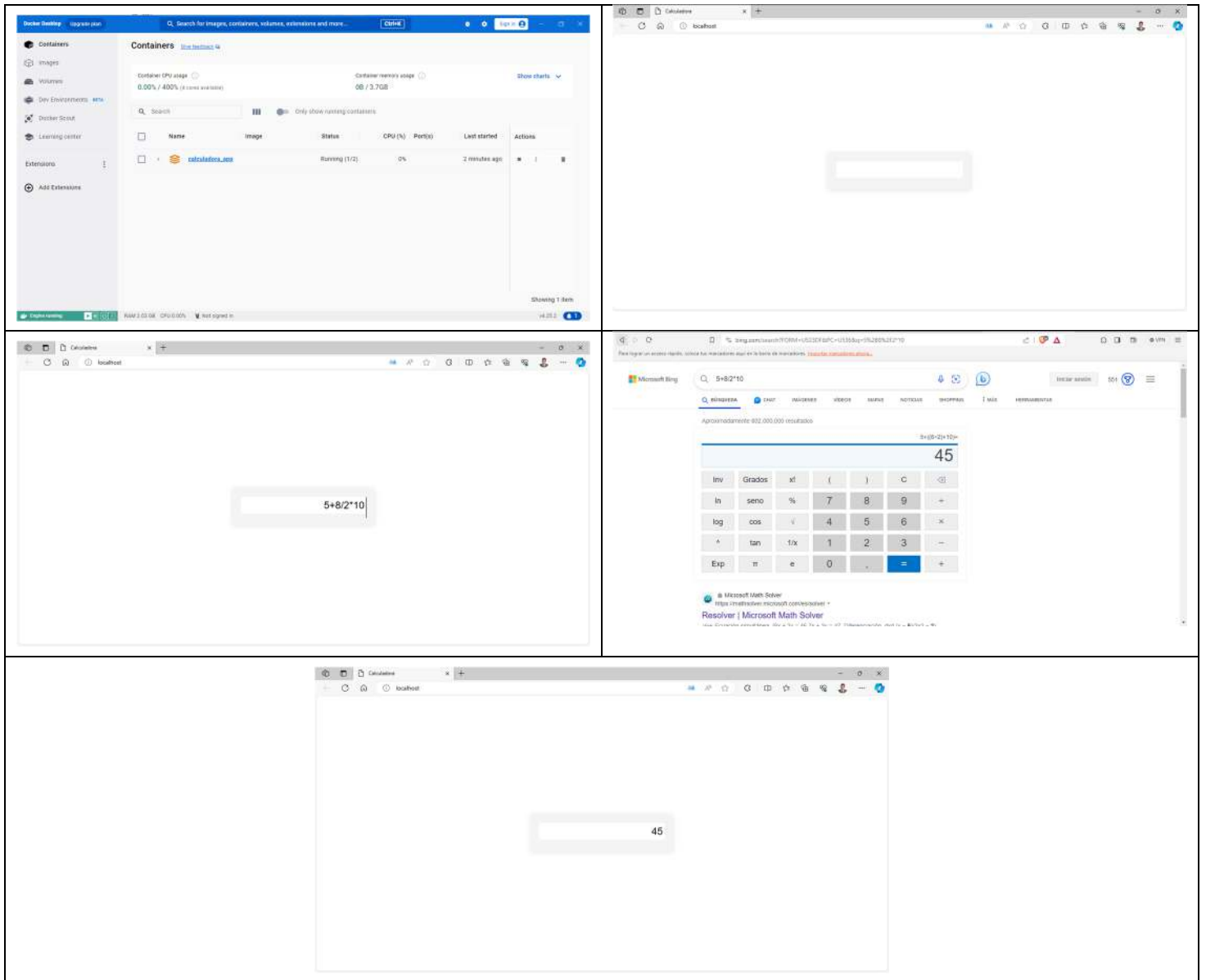
Código JS frontend

```
1  let displayValue = '';
2
3  function appendToDisplay(value) {
4      displayValue += value;
5      updateDisplay();
6  }
7
8  function clearDisplay() {
9      displayValue = '';
10     updateDisplay();
11 }
12
13 function calculate() {
14     try {
15         const result = safeEval(displayValue);
16         displayValue = result.toString();
17         updateDisplay();
18     } catch (error) {
19         displayValue = 'Error';
20         updateDisplay();
21     }
22 }
23
24 function updateDisplay() {
25     document.getElementById('display').value = displayValue;
26 }
27
28 function safeEval(expr) {
29     return Function('"use strict";return (' + expr + ')')();
30 }
31
32 // Permitir entrada desde el teclado
33 document.addEventListener('keydown', function (event) {
34     const key = event.key;
35     if (/^[0-9]/.test(key)) {
36         event.preventDefault();
37         if (key === '=' || key === 'Enter') {
38             calculate();
39         } else {
40             appendToDisplay(key);
41         }
42     }
43 });
```

Código JS backend

```
1 | const express = require('express');
2 | const bodyParser = require('body-parser');
3 |
4 | const app = express();
5 | const port = 3000;
6 |
7 | app.use(bodyParser.json());
8 | app.use(express.static('frontend')); // Servir archivos estáticos desde el frontend
9 |
10 | // Rutas del API
11 | app.post('/api/calculate', (req, res) => {
12 |   const expression = req.body.expression;
13 |   try {
14 |     const result = safeEval(expression);
15 |     res.json({ result });
16 |   } catch (error) {
17 |     res.status(500).json({ error: 'Error evaluating expression' });
18 |   }
19 | });
20 |
21 | app.listen(port, () => {
22 |   console.log(`Backend server listening at http://localhost:${port}`);
23 | });
24 |
25 | function safeEval(expr) {
26 |   return Function("use strict";return (' + expr + ''))();
27 | }
28 |
```

Resultados:



Conclusión

Aunque fue realmente complicado lograr desarrollar la aplicación con microservicios al completarla y ver el funcionamiento de la misma comprendí que este es más rápido y con menos peso para mi computadora o bien esa impresión me dio al momento de lograr ejecutarla sin problemas.