

PROYECTO INDIVIDUAL 1 (Calculadora)

ENLACE GIT HUB → https://github.com/WuiliGusU/Programacion-2_Mini-Proyecto-1-Calculadora-git

Tecnologías utilizadas:

- Java
- Spring Framework
- React

FUNCIONAMIENTO

Backend:

Fue desarrollado por medio de un código implementado en java el funcionamiento de las operaciones a realizarse (Suma, Resta, Multiplicación y División), lo cual le desplegara la opción de ingresar los números deseados a operar y seleccionar la operación a realizar:

```
CalculatorController.java X
calculadora-backend > src > main > java > com > calculadora > controller > CalculatorController.java > CalculatorController > sumar(double, double)
1 package com.calculadora.controller;
2
3 import org.springframework.web.bind.annotation.*;
4
5 @RestController
6 @RequestMapping("/api")
7 @CrossOrigin(origins = "http://localhost:5173")
8 public class CalculatorController {
9
10     @GetMapping("/sumar")
11     public double sumar(@RequestParam double a, @RequestParam double b) {
12         return a + b;
13     }
14
15     @GetMapping("/restar")
16     public double restar(@RequestParam double a, @RequestParam double b) {
17         return a - b;
18     }
19
20     @GetMapping("/multiplicar")
21     public double multiplicar(@RequestParam double a, @RequestParam double b) {
22         return a * b;
23     }
24
25     @GetMapping("/dividir")
26     public double dividir(@RequestParam double a, @RequestParam double b) {
27         if (b == 0) {
28             throw new ArithmeticException(s:"División por cero no permitida.");
29         }
30         return a / b;
31     }
32 }
```

Frontend:

Se realizo un apartado en React para la manera interactiva y visual de manipular la calculadora, lo que nos mostrara de manera visual como es nuestra calculadora, conectándose a al Backend en este caso realizado por medio de java.

```
App.jsx
calculadora-frontend > src > App.jsx > App

1  import React, { useState } from 'react';
2
3  function App() {
4    const [operacion, setOperacion] = useState('sumar');
5    const [a, setA] = useState('');
6    const [b, setB] = useState('');
7    const [resultado, setResultado] = useState(null);
8
9    const calcular = async () => {
10     try {
11       const response = await fetch(`http://localhost:8080/api/${operacion}?a=${a}&b=${b}`);
12       const data = await response.json();
13       setResultado(data);
14     } catch (error) {
15       console.error('Error al realizar la operación:', error);
16     }
17   };
18
19   return (
20     <div>
21       <h1>Calculadora</h1>
22       <input type="number" value={a} onChange={(e) => setA(e.target.value)} placeholder="Número A" />
23       <input type="number" value={b} onChange={(e) => setB(e.target.value)} placeholder="Número B" />
24       <select value={operacion} onChange={(e) => setOperacion(e.target.value)}>
25         <option value="sumar">Sumar</option>
26         <option value="restar">Restar</option>
27         <option value="multiplicar">Multiplicar</option>
28         <option value="dividir">Dividir</option>
29       </select>
30       <button onClick={calcular}>Calcular</button>
31       {resultado !== null && <h2>Resultado: {resultado}</h2>}
32     </div>
33   );
34 }
35
36 export default App;
```

```
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color-scheme: light dark;
  color: #242424;
  background-color: #f4f4f4;

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

a {
  font-weight: 500;
  color: #646cff;
  text-decoration: inherit;
}
a:hover {
  color: #535bf2;
}

body {
  margin: 0;
  display: flex;
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}

h1 {
  font-size: 3.2em;
  line-height: 1.1;
}
```

DESPLIEGUE Y UTILIZACION

Desde la ubicación de nuestro proyecto en el área de Backend ejecutamos por medio del siguiente comando: `mvn clean`, lo que nos permitirá limpiar algún proceso para poder desplegar nuestra calculadora sin ningún fallo que nos podamos encontrar.

```
C:\Users\soporte\Downloads\calculadora-backend\calculadora-backend>mvn clean
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model
for com.calculadora:calculadora-backend:jar:0.0.1-SNAPSHOT
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must
be unique: org.springframework.boot:spring-boot-starter-web:jar -> duplica
te declaration of version (?) @ line 40, column 21
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must
be unique: org.springframework.boot:spring-boot-starter-test:jar -> duplic
ate declaration of version (?) @ line 45, column 21
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threat
en the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support bui
lding such malformed projects.
[WARNING]
[INFO]
[INFO] -----< com.calculadora:calculadora-backend >-----
[INFO]
[INFO] Building CalculadoraBackend 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ calculadora-backend ---
[INFO] Deleting C:\Users\soporte\Downloads\calculadora-backend\calculadora-b
ackend\target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  0.759 s
[INFO] Finished at: 2024-09-14T15:12:01-06:00
[INFO] -----
```

- Seguidamente Ejecutamos el comando: `mvn install` para poder compilar e instalar todos los recursos que se puedan necesitar todo igualmente desde nuestra carpeta de Backend.

```
C:\Users\soporte\Downloads\calculadora-backend\calculadora-backend>mvn insta
ll
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model
for com.calculadora:calculadora-backend:jar:0.0.1-SNAPSHOT
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must
be unique: org.springframework.boot:spring-boot-starter-web:jar -> duplica
te declaration of version (?) @ line 40, column 21
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must
be unique: org.springframework.boot:spring-boot-starter-test:jar -> duplic
ate declaration of version (?) @ line 45, column 21
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threat
en the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support bui
lding such malformed projects.
[WARNING]
[INFO]
[INFO] -----< com.calculadora:calculadora-backend >-----
[INFO]
[INFO] Building CalculadoraBackend 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ calculadora-backe
nd ---
[INFO] skip non existing resourceDirectory C:\Users\soporte\Downloads\calcul
adora-backend\calculadora-backend\src\main\resources
[INFO] skip non existing resourceDirectory C:\Users\soporte\Downloads\calcul
adora-backend\calculadora-backend\src\main\resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ calculadora-backend -
---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 4 source files with javac [debug release 17] to target\clas
```

- Ejecutamos por último el siguiente comando: `mvn spring-boot:run` para compilar y que se pueda mantener ejecutando nuestro Backend para darle funcionamiento a nuestra calculadora esto siendo posible por medio de la conexión con Spring Framework.

```

calculadora-backend ---
[INFO] No sources to compile
[INFO]
[INFO] <<< spring-boot:3.1.3:run (default-cli) < test-compile @ calculadora-backend >>>
[INFO]
[INFO]
[INFO] --- spring-boot:3.1.3:run (default-cli) @ calculadora-backend ---
[INFO] Attaching agents: []

  ____ _
 / ___ \| | | |
/ /   \| |_| |
\ \   /| | | |
 \ \_/ \| |_| |
  \___|_||_|_|_|

:: Spring Boot ::
              (v3.1.3)

2024-09-14T15:12:39.769-06:00 INFO 20200 --- [main] c.c.CalculadoraBackendApplication : Starting CalculadoraBackendApplication using Java 17.0.12 with PID 20200 (C:\Users\soporte\Downloads\calculadora-backend\calculadora-backend\target\classes started by soporte in C:\Users\soporte\Downloads\calculadora-backend\calculadora-backend)
2024-09-14T15:12:39.772-06:00 INFO 20200 --- [main] c.c.CalculadoraBackendApplication : No active profile set, falling back to 1 default profile: "default"
2024-09-14T15:12:40.407-06:00 INFO 20200 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2024-09-14T15:12:40.419-06:00 INFO 20200 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-14T15:12:40.419-06:00 INFO 20200 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.12]
2024-09-14T15:12:40.540-06:00 INFO 20200 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-09-14T15:12:40.544-06:00 INFO 20200 --- [main] w.s.c.ServletContextWebServerApplicationContext : Root WebApplicationContext: initialization completed in 722 ms
2024-09-14T15:12:40.846-06:00 INFO 20200 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with c

```

- Del lado donde tenemos nuestra estructura del Frontend realizamos el mismo procedimiento ejecutando el código: `mvn spring-boot:run`, para ejecutar nuestro frontend.

```

PS C:\Users\soporte\Downloads\calculadora-backend\calculadora-frontend> npm run dev

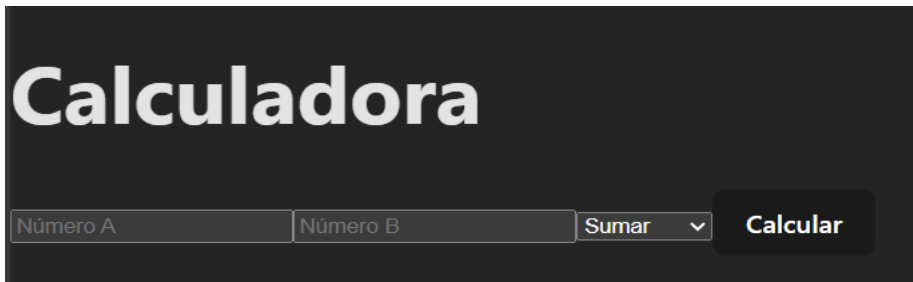
> calculadora-frontend@0.0.0 dev
> vite

VITE v5.4.5 ready in 289 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

```

- Al momento de Ejecutar nuestro comando nos proporcionara un URL LOCAL, el cual podemos abrir en nuestro navegador para poder desplegar nuestra calculadora para su uso:




Nuestra calculadora nos solicitara ingresar 2 números para ser operados y nos solicitara también el seleccionar la operación deseada a realizarse.

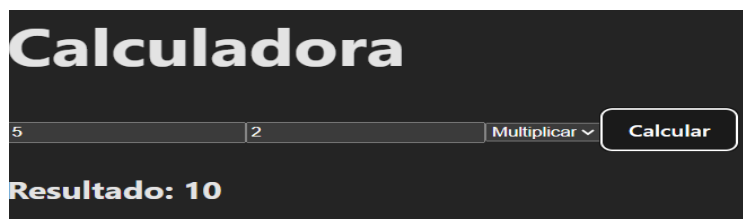
- SUMA



- RESTA



- MULTIPLICACION



- DIVISION

