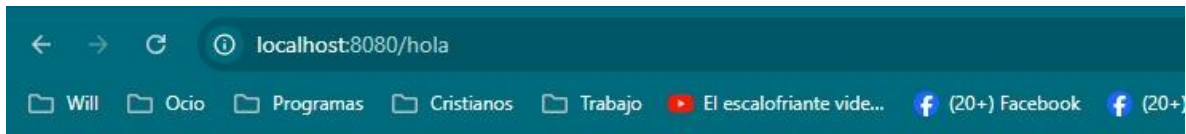


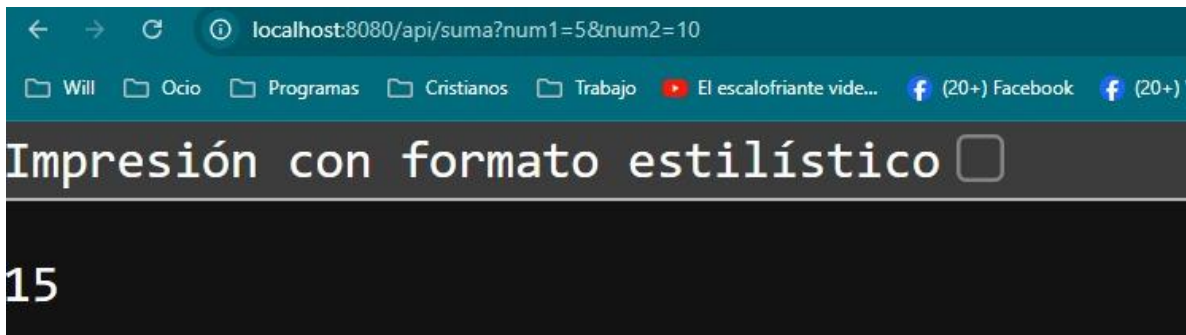
Proyecto:

pruebas para ver que funcione el localhost:8080/hola

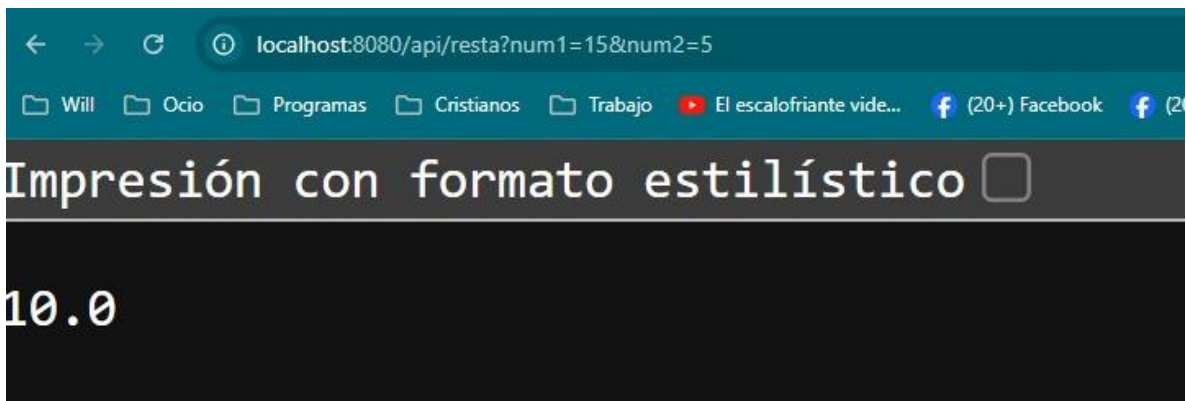


Holamundo!

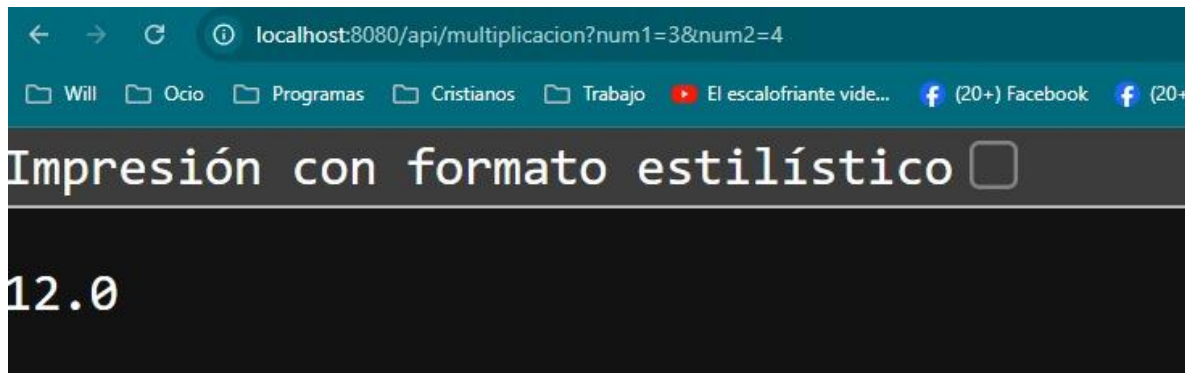
suma:



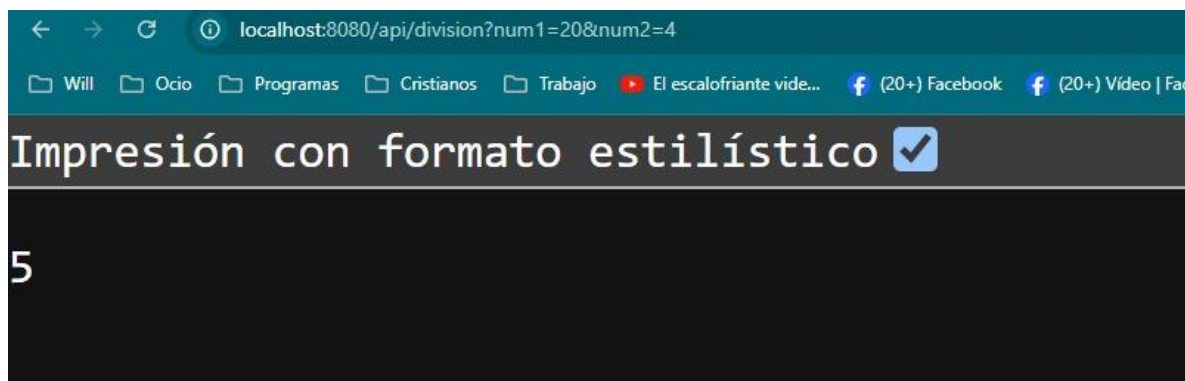
Resta:



División:



La división sin decimal:



División por 0:



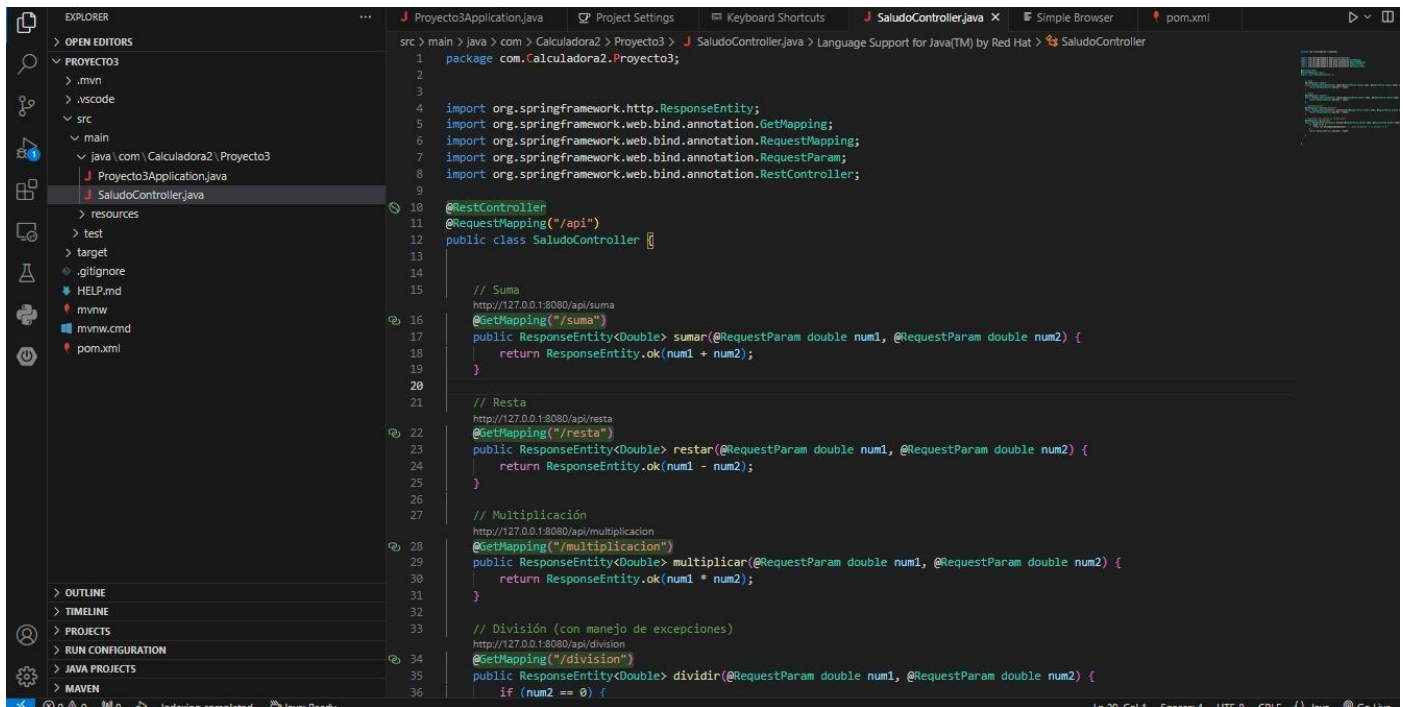
Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

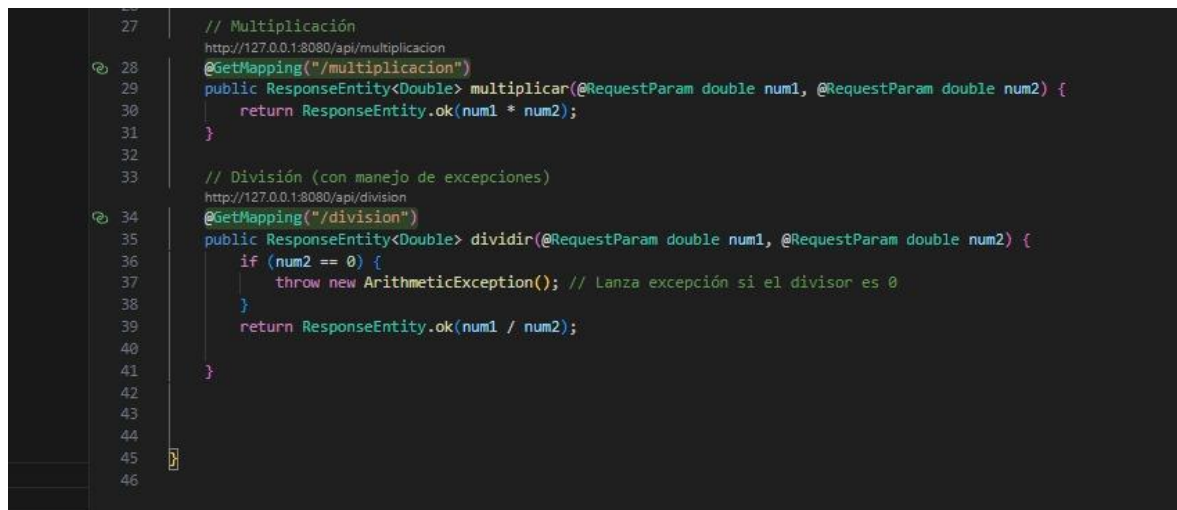
Thu Sep 12 23:58:34 CST 2024

There was an unexpected error (type=Internal Server Error, status=500).

Código de calculadora:



```
1 package com.Calculadora2.Proyecto3;
2
3
4 import org.springframework.http.ResponseEntity;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.RestController;
9
10 @RestController
11 @RequestMapping("/api")
12 public class SaludoController {
13
14
15     // Suma
16     http://127.0.0.1:8080/api/suma
17     @GetMapping("/suma")
18     public ResponseEntity<Double> sumar(@RequestParam double num1, @RequestParam double num2) {
19         return ResponseEntity.ok(num1 + num2);
20     }
21
22     // Resta
23     http://127.0.0.1:8080/api/resta
24     @GetMapping("/resta")
25     public ResponseEntity<Double> restar(@RequestParam double num1, @RequestParam double num2) {
26         return ResponseEntity.ok(num1 - num2);
27     }
28
29     // Multiplicación
30     http://127.0.0.1:8080/api/multiplicacion
31     @GetMapping("/multiplicacion")
32     public ResponseEntity<Double> multiplicar(@RequestParam double num1, @RequestParam double num2) {
33         return ResponseEntity.ok(num1 * num2);
34     }
35
36     // División (con manejo de excepciones)
37     http://127.0.0.1:8080/api/division
38     @GetMapping("/division")
39     public ResponseEntity<Double> dividir(@RequestParam double num1, @RequestParam double num2) {
40         if (num2 == 0) {
41             throw new ArithmeticException(); // Lanza excepción si el divisor es 0
42         }
43         return ResponseEntity.ok(num1 / num2);
44     }
45 }
```



```
27 // Multiplicación
28 http://127.0.0.1:8080/api/multiplicacion
29 @GetMapping("/multiplicacion")
30 public ResponseEntity<Double> multiplicar(@RequestParam double num1, @RequestParam double num2) {
31     return ResponseEntity.ok(num1 * num2);
32 }
33
34 // División (con manejo de excepciones)
35 http://127.0.0.1:8080/api/division
36 @GetMapping("/division")
37 public ResponseEntity<Double> dividir(@RequestParam double num1, @RequestParam double num2) {
38     if (num2 == 0) {
39         throw new ArithmeticException(); // Lanza excepción si el divisor es 0
40     }
41     return ResponseEntity.ok(num1 / num2);
42 }
43
44
45
46 }
```

Suma: El método sumar responde a la solicitud /api/suma, suma los dos números y devuelve el resultado. <http://localhost:8080/api/division?num1=20&num2=4>

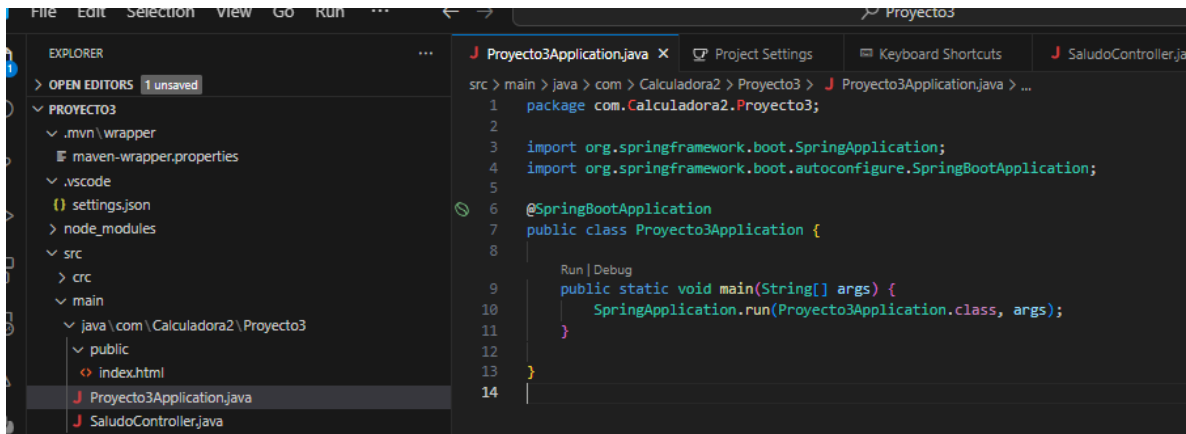
Resta: El método restar se llama desde la ruta /api/resta, realiza la resta de num1 y num2 y devuelve la respuesta. <http://localhost:8080/api/resta?num1=15&num2=5>

Multiplicación: Con el endpoint /api/multiplicación, el método multiplicar calcula el producto de los dos números. <http://localhost:8080/api/multiplicacion?num1=3&num2=4>

División: El método dividir, accedido desde /api/division, divide los dos números. Si num2 es igual a 0, lanza una excepción aritmética para evitar la división por cero. Cada operación

devuelve el resultado dentro de un objeto ResponseEntity, que envía una respuesta HTTP con el valor calculado. "http://localhost:8080/api/division?num1=20&num2=4"

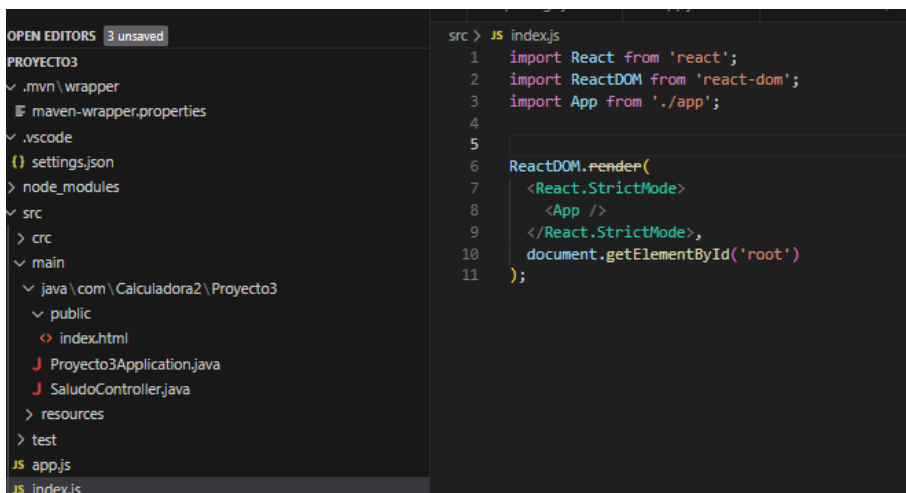
La base:



La clase Proyecto3Application es el punto de partida de una aplicación Spring Boot. La anotación @SpringBootApplication agrupa tres anotaciones esenciales de Spring (@Configuration, @EnableAutoConfiguration y @ComponentScan), lo que permite que Spring Boot configure automáticamente los componentes y escanee el proyecto en busca de otros componentes, controladores y servicios.

Es el "motor" que ejecuta el proyecto completo, iniciando todos los componentes y servicios necesarios para que la aplicación funcione.

index.js Frontend



Este código se encarga de renderizar la aplicación React en el navegador. Utiliza ReactDOM.render() para montar el componente principal App dentro del elemento HTML con id root. También incluye el uso opcional de un archivo CSS global para aplicar estilos. React.StrictMode ayuda a garantizar que el código siga las mejores prácticas de React, facilitando el desarrollo y la detección de errores.

App.Js frontend

```
src > JS app.js > App
1 import React, { useState } from 'react';
2 import axios from 'axios'; // Necesitas instalar axios para hacer solicitudes HTTP
3
4 function App() {
5   // Estados para los dos números y el resultado
6   const [num1, setNum1] = useState('');
7   const [num2, setNum2] = useState('');
8   const [resultado, setResultado] = useState(null);
9
10  // Función para hacer las solicitudes al backend
11  const calcularResultado = (operacion) => {
12    const url = `http://localhost:8080/api/${operacion}?num1=${num1}&num2=${num2}`;
13
14    axios.get(url)
15      .then((response) => {
16        setResultado(response.data); // Obtén el resultado del backend
17      })
18      .catch((error) => {
19        console.error("Hubo un error con la operación:", error);
20      });
21  };
22
23  return (
24    <div>
25      <h1>Calculadora</h1>
26      <input
27        type="number"
28        value={num1}
29        onChange={(e) => setNum1(e.target.value)}
30        placeholder="Número 1"
31      />
```

```
32    <input
33      type="number"
34      value={num2}
35      onChange={(e) => setNum2(e.target.value)}
36      placeholder="Número 2"
37    />
38
39    <div>
40      <button onClick={() => calcularResultado('suma')}>Sumar</button>
41      <button onClick={() => calcularResultado('resta')}>Restar</button>
42      <button onClick={() => calcularResultado('multiplicacion')}>Multiplicar</button>
43      <button onClick={() => calcularResultado('division')}>Dividir</button>
44    </div>
45
46    <h2>Resultado: {resultado}</h2>
47  </div>
48  );
49 }
50
51 export default App;
52
```

Este código crea el frontend de mi aplicación calculadora en React, el cual se conecta con el backend desarrollado en Spring Boot. La interfaz permite al usuario ingresar dos números y realizar operaciones de suma, resta, multiplicación y división a través de botones.

Cuando se selecciona una operación, el frontend envía una solicitud HTTP a los endpoints del backend (/suma, /resta, etc.) usando la librería Axios. El backend realiza el cálculo y devuelve el resultado, que luego se muestra en la interfaz.

Pantallas de solucion mientras se esta ejecutando en el localhost:8080

```

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

:: Spring Boot ::      (v3.3.3)

2024-09-12T23:51:35.378-06:00 INFO 13176 --- [Projecto3] [main] c.C.Projecto3.Projecto3Application : Starting Proyecto3Application using Java 17.0.12 with PID 13176 (C:\Users\Willys GaBo\Desktop\Projecto3\target\classes started by Willys GaBo in C:\Users\Willys GaBo\Desktop\Projecto3)
2024-09-12T23:51:35.384-06:00 INFO 13176 --- [Projecto3] [main] c.C.Projecto3.Projecto3Application : No active profile set, falling back to 1 default profile: "default"
2024-09-12T23:51:36.668-06:00 INFO 13176 --- [Projecto3] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-09-12T23:51:36.685-06:00 INFO 13176 --- [Projecto3] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-12T23:51:36.685-06:00 INFO 13176 --- [Projecto3] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.28]
2024-09-12T23:51:36.748-06:00 INFO 13176 --- [Projecto3] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-09-12T23:51:36.749-06:00 INFO 13176 --- [Projecto3] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1303 ms
2024-09-12T23:51:37.398-06:00 INFO 13176 --- [Projecto3] [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1 endpoint beneath base path '/actuator'
2024-09-12T23:51:37.477-06:00 INFO 13176 --- [Projecto3] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2024-09-12T23:51:37.492-06:00 INFO 13176 --- [Projecto3] [main] c.C.Projecto3.Projecto3Application : Started Proyecto3Application in 2.546 seconds (process running for 3.132)
2024-09-12T23:51:38.062-06:00 INFO 13176 --- [Projecto3] [on(2)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-09-12T23:51:38.063-06:00 INFO 13176 --- [Projecto3] [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2024-09-12T23:51:38.064-06:00 INFO 13176 --- [Projecto3] [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2024-09-12T23:58:34.339-06:00 ERROR 13176 --- [Projecto3] [nio-8080-exec-8] o.a.c.c.C.[.][.][.][dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: java.lang.ArithmeticException] with root cause

```

```

src / main / java / com / Calculadora / Projecto3 / SaludoController.java / Language Support for Java(TM) by Red Hat / SaludoController

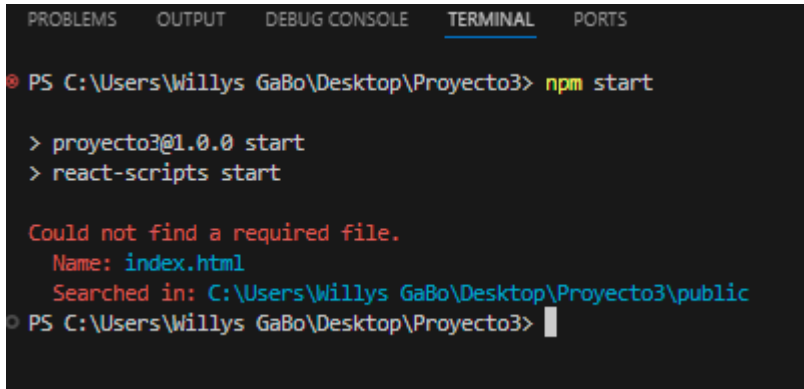
17 public class SaludoController {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:116) ~[spring-web-6.1.12.jar:6.1.12]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:164) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:140) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:201) ~[spring-web-6.1.12.jar:6.1.12]
at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:116) ~[spring-web-6.1.12.jar:6.1.12]
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:164) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:140) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:167) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:90) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:483) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:115) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:93) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:74) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:344) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:384) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:904) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1741) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1190) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1190) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:63) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:63) ~[tomcat-embed-core-10.1.28.jar:10.1.28]
at java.base/java.lang.Thread.run(Unknown Source) ~[na:na]

```

El único error para abrir react



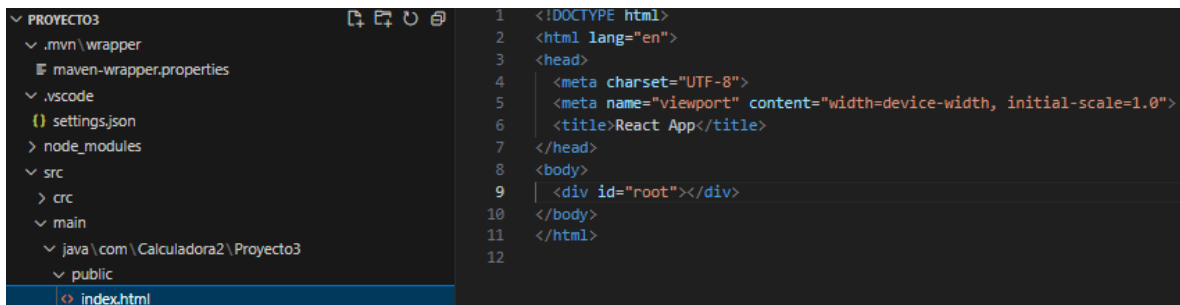
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Willys GaBo\Desktop\Proyecto3> npm start

> proyecto3@1.0.0 start
> react-scripts start

Could not find a required file.
  Name: index.html
  Searched in: C:\Users\Willys GaBo\Desktop\Proyecto3\public
PS C:\Users\Willys GaBo\Desktop\Proyecto3>
```

Tira que no hay nada en index.html, pero si tengo todo y no jala y estuve va de probar, pero no jala el programa, ya reinicié también la computadora y nada estuve probando varias soluciones como descargar los paquetes de npm install axios y npm install y si instalo todo, pero no jala.



```
PROJECT03
├── .mvn\wrapper
│   └── maven-wrapper.properties
├── .vscode
│   └── settings.json
├── node_modules
├── src
│   ├── crc
│   └── main
│       ├── java\com\Calculadora2\Proyecto3
│       └── public
│           └── index.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>React App</title>
7 </head>
8 <body>
9   <div id="root"></div>
10 </body>
11 </html>
12
```