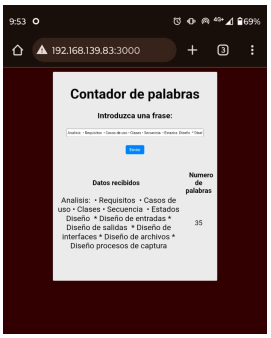
Equipo 1

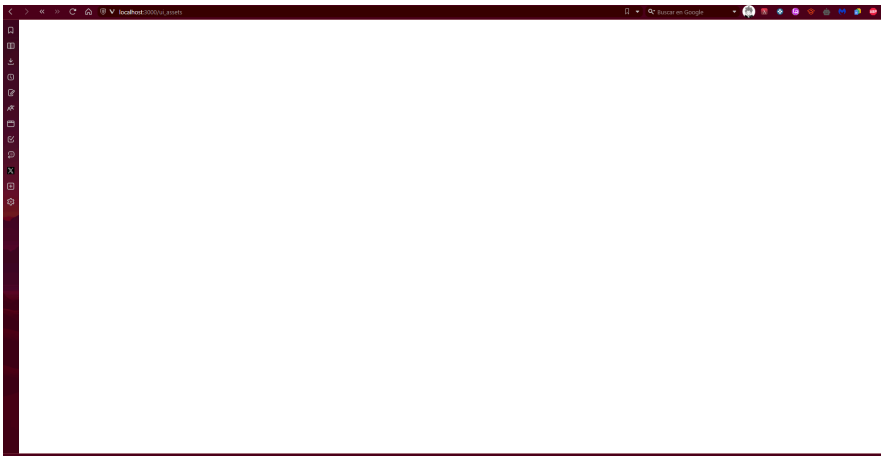
a). Intente acceder al servidor desde su teléfono celular y verifique que funcione correctamente. Es necesario que el firewall del equipo que ejecuta el servidor web permite las conexiones HTTP externas. También es posible que requiera ejecutar el archivo jar dentro de una consola DOS en lugar de usar WSL.



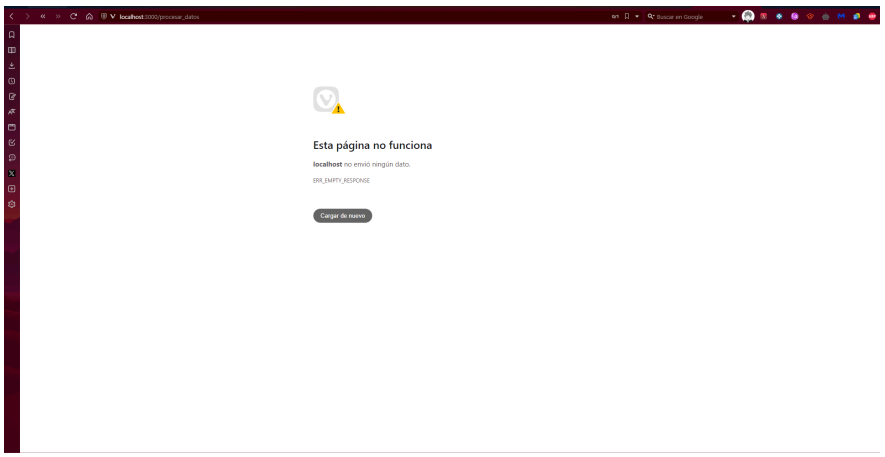
b).- Qué devuelve el servidor al intentar acceder a los endpoint declarados en WebServer.java: /status, /, /ui_assets y /procesar_datos

- Para /status aparece lo siguiente:

- Para /ui_assets



- Para /procesar_datos



c).- Agregue las impresiones que considere necesarias en el código WebServer.java para determinar cuál es el primer método HTTP enviado del navegador hacia el servidor web.

```
~/my-app is  v1.0-SNAPSHOT via  v17.0.10 took 2s
❯ java -jar target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar
Servidor escuchando en el puerto: 3000
Método HTTP recibido en handleRequestForAsset: GET
Método HTTP recibido en handleRequestForAsset: GET
Método HTTP recibido en handleRequestForAsset: GET
Método HTTP recibido en handleRequestForAsset: GET
```

Vemos que el primer método que se envía desde el navegador hacia el servidor web es el método GET.

d).- ¿Cuál es el primer, segundo y tercer asset enviados del servidor hacia el navegador? En el método sendResponse agregue la impresión del número de bytes enviados para verificar que coinciden en tamaño con los archivos enviados.

```
~/my-app is  v1.0-SNAPSHOT via  v17.0.10 took 2s
❯ java -jar target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar
Servidor escuchando en el puerto: 3000
se recibió la solicitud de: /
se envió el index.html
Bytes sent: 707
se recibió la solicitud de: /ui_assets/style.css
se recibió la solicitud de: /ui_assets/javascript.js
Bytes sent: 562
Bytes sent: 1328
```

Vemos que el primer asset enviado es el Index con 707 bytes.
Luego se manda el style.css con 562 bytes
Y el tercero es el javascript.js con 1328 bytes.

e).- ¿En el código del servidor cuál es el propósito de ejecutar getClass().getResourceAsStream()?

El propósito de ejecutar getClass().getResourceAsStream() en el código es leer el contenido de un archivo desde el classpath. El classpath es un sistema de archivos virtual que incluye todos los recursos (como archivos de clase Java, archivos de configuración o archivos HTML) empaquetados con su aplicación.

f).- ¿Para qué se ejecuta el método addContentType() ?

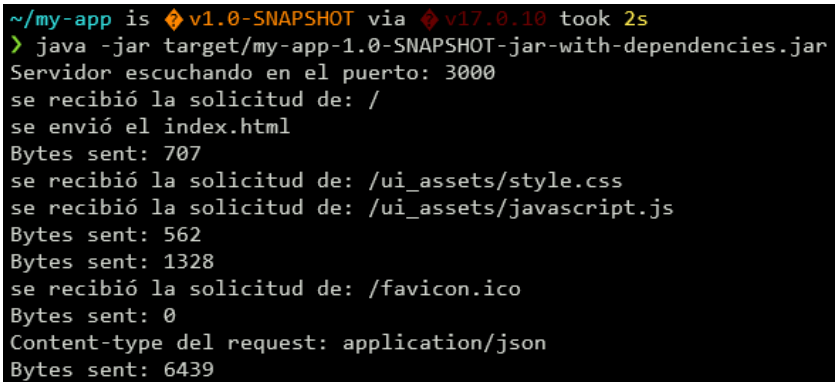
El método addContentType() establece el encabezado Content-Type apropiado en la respuesta HTTP. El encabezado Content-Type especifica el tipo de medio de los datos que se envían en la respuesta. Esto ayuda al cliente (navegador web) a procesar correctamente los datos.

g).- Después de recibir los assets, al introducir la frase en el navegador y dar click en enviar, ¿qué método HTTP se ejecuta, a qué endpoint accede y qué método del código Java se ejecuta?

Cuando el usuario ingresa una frase y hace clic en enviar, sucede lo siguiente:

Método HTTP: El método HTTP utilizado es POST. Esto se debe a que el cliente (navegador web) está enviando datos (la frase ingresada por el usuario) al servidor.
Endpoint al que se accede: El endpoint al que se accede es /procesar_datos (que se traduce como procesar datos en español). Este endpoint está mapeado al método handleTaskRequest en el código.
Método Java ejecutado: El método Java ejecutado es handleTaskRequest. Este método lee el cuerpo de la solicitud (que contiene la frase ingresada por el usuario), la analiza, cuenta la cantidad de palabras y envía una respuesta al cliente.

h).- En el inciso anterior ¿qué valor tiene el header “Content-type” cuando el mensaje llega al servidor? (imprimir su valor)



En este caso vemos que la cantidad de bytes del Content-type es de 6439.

i).- En el método handleTaskRequest: ¿Para qué se utiliza el método de Jackson readValue?¿qué se almacena en la variable String frase? ¿Qué se almacena en el objeto frontendSearchResponse? ¿Para qué se utiliza el método de Jackson writeValueAsBytes?

¿Para qué se utiliza el método de Jackson readValue?

El método readValue de Jackson se utiliza para deserializar el cuerpo de la solicitud HTTP POST. La deserialización es el proceso de convertir los datos binarios (en este caso, el cuerpo de la solicitud) en un objeto Java.

¿Qué se almacena en la variable string frase?

La variable frase se almacena en el objeto FrontendSearchRequest después de que se deserializa el cuerpo de la solicitud HTTP POST usando el método readValue de Jackson. La variable frase probablemente contiene la consulta de búsqueda real ingresada por el usuario en el formulario web.

¿Qué se almacena en el objeto frontendSearchResponse?

El objeto FrontendSearchResponse se crea para almacenar la respuesta que se enviará al cliente. En este caso, almacena la consulta de búsqueda original (frase) y el número de palabras en la consulta de búsqueda. El número de palabras se calcula usando un objeto StringTokenizer para dividir la frase en tokens (palabras) y luego contando el número de tokens.

¿Para qué se utiliza el método de Jackson writeValueAsBytes?

El método writeValueAsBytes de Jackson se utiliza para serializar el objeto FrontendSearchResponse en una representación de bytes. La serialización es el proceso de convertir un objeto Java en una secuencia de bytes que se pueden almacenar o transmitir.

j).- En el servidor HTTP analizado con anterioridad, la línea final del método sendResponse era exchange.close() , ¿Por qué se omitió esta vez?

Omitir exchange.close() en este caso es una decisión basada en la gestión adecuada de los recursos y en la intención de permitir un manejo más flexible del intercambio HTTP. El enfoque actual se asegura de que el OutputStream se cierre correctamente, lo que normalmente es suficiente para indicar al servidor que el intercambio ha finalizado. Si el servidor HTTP maneja el cierre de la conexión automáticamente después de cerrar el OutputStream, no es necesario llamar explícitamente a exchange.close().

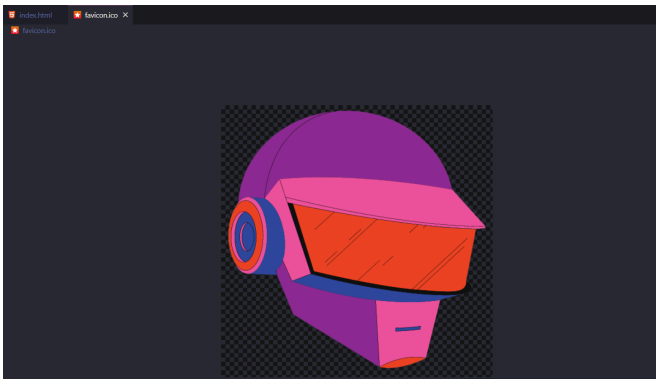
k).- Además de los assets HTML, CSS y JS, el navegador solicita un cuarto asset. ¿Cuál es y que significa? Realice las modificaciones necesarias para que también se envíe el cuarto

asset y se muestre en el navegador. Envíe la captura de pantalla correspondiente para demostrar que funciona.

```
~/my-app is v1.0-SNAPSHOT via v17.0.10 took 2s
❯ java -jar target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar
Servidor escuchando en el puerto: 3000
se recibió la solicitud de: /
se envió el index.html
Bytes sent: 769
se recibió la solicitud de: /ui_assets/style.css
se recibió la solicitud de: /ui_assets/javascript.js
Bytes sent: 562
Bytes sent: 1328
se recibió la solicitud de: /favicon.ico
Bytes sent: 0
```

Primero vemos que el navegador que estoy usando, en este caso vivaldi si nos pide el cuarto asset, pero como no tenemos nada no lo usa.

Y ahí está el nuevo icono que pusimos, que es la siguiente imagen:



```
~/my-app is v1.0-SNAPSHOT via v17.0.10 took 2s
❯ java -jar target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar
Servidor escuchando en el puerto: 3000
se recibió la solicitud de: /
se envió el index.html
Bytes sent: 778
se recibió la solicitud de: /ui_assets/style.css
se recibió la solicitud de: /ui_assets/javascript.js
Bytes sent: 562
Bytes sent: 1328
se recibió la solicitud de: /ui_assets/favicon.ico
Bytes sent: 44895
```

Y al correr el servidor notamos que la imagen tiene un peso de 44895 bytes.