

Alumno: Kevin Zamora Amela
Tarea: PSP05
Asignatura: Programación de Servicios y Procesos

Ejercicio 1.

Modificar el ejemplo del servidor HTTP (Proyecto java ServerHTTP, apartado 5.1 de los contenidos) para que incluya la cabecera Date.

- Para realizar el presente ejercicio, solamente debemos añadir la siguiente línea en el archivo “ServidorHTTP.java”, ubicándolo en el “condicional IF anidado dentro de otro ‘IF’” que se encuentra contenido en el interior del método “procesaPetición”:

printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));

En la siguiente captura de pantalla se puede apreciar el código modificado:

```
if (peticion.startsWith("GET")) {
    //extrae la subcadena entre 'GET' y 'HTTP/1.1'
    peticion = peticion.substring(3, peticion.lastIndexOf("HTTP"));

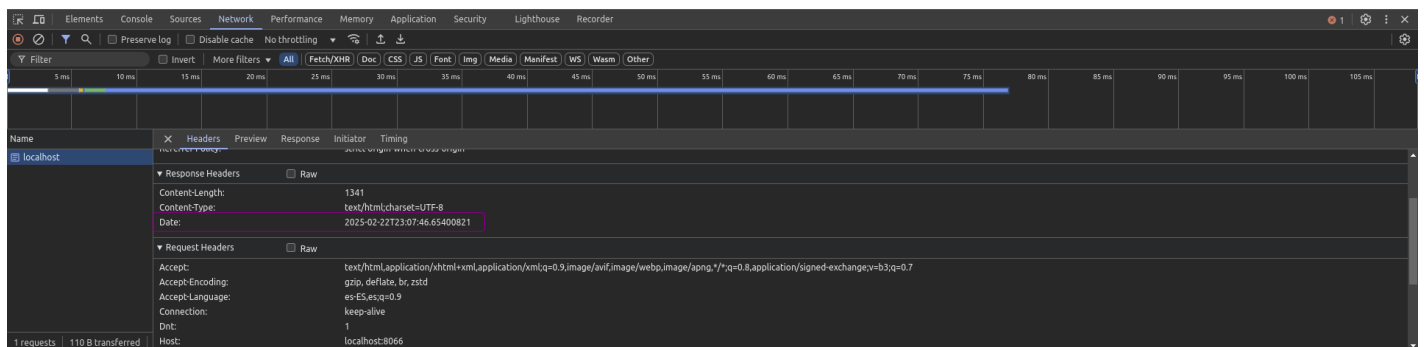
    //si corresponde a la página de inicio
    if (peticion.length() == 0 || peticion.equals("/")) {
        //sirve la página
        html = Paginas.html_index;
        printWriter.println(Mensajes.lineaInicial_OK);
        printWriter.println(Paginas.primerCabecera);
        printWriter.println("Content-Length: " + html.length() + 1);
        // Añadimos la cabecera de la hora actual en la página principal
        printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));
        printWriter.println("\n");
        printWriter.println(html);
    } //si corresponde a la página del Quijote
    else if (peticion.equals("/quijote")) {
        //sirve la página
        html = Paginas.html_quijote;
        printWriter.println(Mensajes.lineaInicial_OK);
        printWriter.println(Paginas.primerCabecera);
        printWriter.println("Content-Length: " + html.length() + 1);
        // Añadimos la cabecera de la hora actual en la página dedicada al quijote
        printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));
        printWriter.println("\n");
        printWriter.println(html);
    } //en cualquier otro caso
    else {
        //sirve la página
        html = Paginas.html_noEncontrado;
        printWriter.println(Mensajes.lineaInicial_NotFound);
        printWriter.println(Paginas.primerCabecera);
        printWriter.println("Content-Length: " + html.length() + 1);
        // Añadimos la cabecera de la hora actual en la página que se mostrará en caso de error
        printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));
        printWriter.println("\n");
        printWriter.println(html);
    }
}
```

Añadimos las importaciones pertinentes:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.logging.Level;
import java.util.logging.Logger;
```

Y comprobamos la existencia de la nueva cabecera:

Para acceder a la correspondiente sección/herramienta de análisis web desde nuestro navegador seleccionado, pulsamos la tecla F12 o hacemos clic derecho sobre la página que tengamos abierta (de las que componen las vistas HTML de nuestro servidor) y seleccionamos la opción “inspeccionar”. Una vez se abra esa nueva ‘ventana’/sección de “administración”, seleccionamos la pestaña “Network/Red” y dentro de esta “sección”, seleccionamos el nombre / etiqueta de la página que se acaba de cargar. Y una vez visualizamos el apartado correspondiente a las “cabeceras de respuesta”, se puede apreciar la nueva cabecera con la fecha actual.



Ejercicio 2.

Modificar el ejemplo del servidor HTTP (Proyecto java ServerHTTP, apartado 5.1 de los contenidos) para que implemente multihilo, y pueda gestionar la concurrencia de manera eficiente.

- Primeramente y para trabajar/modificar el proyecto con mayor comodidad y facilidad, extraemos el contenido del método “procesaPetición”, desde el archivo “ServidorHTTP.java”, y lo pegamos en el interior de una clase auxiliar a la que nosotros hemos llamado “HiloPetición”. En este archivo auxiliar no modificamos nada más y por ende, el código modificado se queda exactamente igual que en el del primer ejercicio.

```
if (peticion.startsWith("GET")) {
    //extrae la subcadena entre 'GET' y 'HTTP/1.1'
    peticion = peticion.substring(3, peticion.lastIndexOf("HTTP"));

    //si corresponde a la página de inicio
    if (peticion.length() == 0 || peticion.equals("/")) {
        //sirve la página
        html = Paginas.html_index;
        printWriter.println(Mensajes.lineaInicial_OK);
        printWriter.println(Paginas.primerCabecera);
        printWriter.println("Content-Length: " + html.length() + 1);
        // Añadimos la cabecera de la hora actual en la página principal
        printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));
        printWriter.println("\n");
        printWriter.println(html);
    } //si corresponde a la página del Quijote
    else if (peticion.equals("/quijote")) {
        //sirve la página
        html = Paginas.html_quijote;
        printWriter.println(Mensajes.lineaInicial_OK);
        printWriter.println(Paginas.primerCabecera);
        printWriter.println("Content-Length: " + html.length() + 1);
        // Añadimos la cabecera de la hora actual en la página dedicada al quijote
        printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));
        printWriter.println("\n");
        printWriter.println(html);
    } //en cualquier otro caso
    else {
        //sirve la página
        html = Paginas.html_noEncontrado;
        printWriter.println(Mensajes.lineaInicial_NotFound);
        printWriter.println(Paginas.primerCabecera);
        printWriter.println("Content-Length: " + html.length() + 1);
        // Añadimos la cabecera de la hora actual en la página que se mostrará en caso de error
        printWriter.println("Date: " + LocalDateTime.now().format(DateTimeFormatter.ISO_DATE_TIME));
        printWriter.println("\n");
        printWriter.println(html);
    }
}
```

Para hacer que nuestro programa pueda implementar multihilo y gestionar la concurrencia adecuadamente, añadimos el siguiente fragmento de código en el interior del método “main” de nuestra clase “ServidorHTTP.java”. PD: Para entender mejor las modificaciones realizadas, vamos a comparar la versión original con la versión modificada.

- Versión original (Aportada por el equipo docente):

```
public static void main(String[] args) throws IOException, Exception {

    //Asociamos al servidor el puerto 8066
    ServerSocket socServidor = new ServerSocket(8066);
    imprimeDisponible();
    Socket socCliente;

    //ante una petición entrante, procesa la petición por el socket cliente
    //por donde la recibe
    while (true) {
        //a la espera de peticiones
        socCliente = socServidor.accept();
        //atiendo un cliente
        System.out.println("Atendiendo al cliente ");
        procesaPetición(socCliente);
        //cierra la conexión entrante
        socCliente.close();
        System.out.println("cliente atendido");
    }
}
```

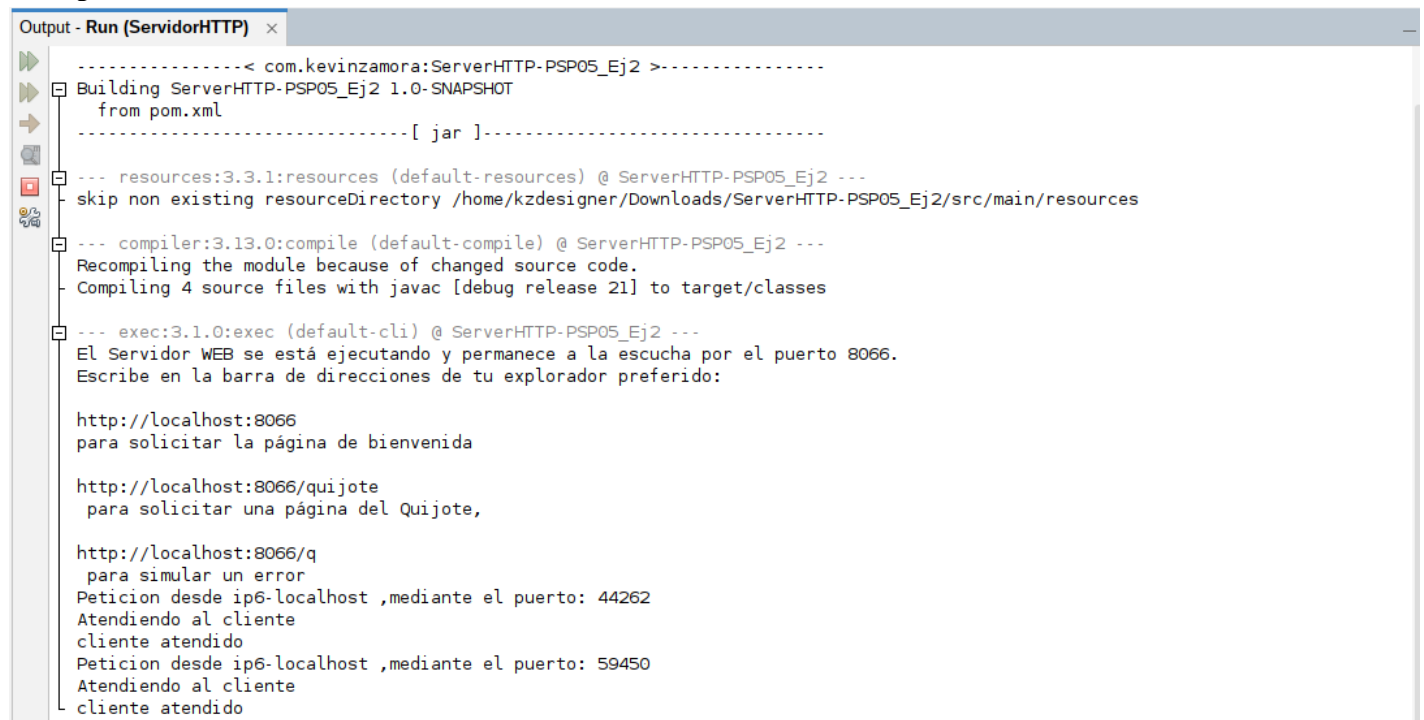
- Versión modificada (Alumno):

```
public static void main(String[] args) throws IOException, Exception {

    //Asociamos al servidor el puerto 8066
    ServerSocket socServidor = new ServerSocket(8066);
    imprimeDisponible();
    Socket socCliente;

    //ante una petición entrante, procesa la petición por el socket cliente
    //por donde la recibe
    while (true) {
        //a la espera de peticiones
        socCliente = socServidor.accept();
        System.out.println("Petición desde " + socCliente.getInetAddress().getHostName()
            + " ,mediante el puerto: " + socCliente.getPort());
        //atiendo un cliente
        HiloPetición hiloPetición = new HiloPetición(socCliente);
        Thread hilo = new Thread(hiloPetición);
        hilo.start();
    }
}
```

- Respuesta obtenida:



```
-----< com.kevinzamora:ServerHTTP-PSP05_Ej2 >-----
Building ServerHTTP-PSP05_Ej2 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ ServerHTTP-PSP05_Ej2 ---
skip non existing resourceDirectory /home/kzdesigner/Downloads/ServerHTTP-PSP05_Ej2/src/main/resources

--- compiler:3.13.0:compile (default-compile) @ ServerHTTP-PSP05_Ej2 ---
Recompiling the module because of changed source code.
Compiling 4 source files with javac [debug release 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ ServerHTTP-PSP05_Ej2 ---
El Servidor WEB se está ejecutando y permanece a la escucha por el puerto 8066.
Escribe en la barra de direcciones de tu explorador preferido:

http://localhost:8066
para solicitar la página de bienvenida

http://localhost:8066/quijote
para solicitar una página del Quijote,

http://localhost:8066/q
para simular un error
Petición desde ip6-localhost ,mediante el puerto: 44262
Atendiendo al cliente
cliente atendido
Petición desde ip6-localhost ,mediante el puerto: 59450
Atendiendo al cliente
cliente atendido
```

- Comentarios relacionados con el proceso de generación de la documentación con JavaDoc de ambos proyectos/ejercicios:

- Primeramente y como ya deberíamos saber, para ejecutar la herramienta JavaDoc y proceder a generar la documentación de nuestro proyecto, debemos hacer clic derecho sobre el directorio raíz de nuestro proyecto con java y seleccionar la opción “Generar JavaDoc/Generate JavaDoc”, según tengamos en Español o en Inglés, respectivamente, la interfaz de nuestro IDE Netbeans.
- Al ejecutar la citada opción “Generar JavaDoc”, nos aparecerá/apareció un mensaje de error comentándonos que nuestra versión es demasiado antigua (siendo la 5 y en consecuencia, siendo anterior a la 8) y por ende, la herramienta de JavaDoc no se encuentra disponible ni resulta compatible.
- Para solucionarlo, simplemente hemos creado un nuevo proyecto de Java 21 (como versión instalada en nuestro sistema) con Maven y hemos “migrado”(o copiado y pegado) las clases de nuestros programas, a dos proyectos de Java nuevos e independientes. Al ejecutar la opción de “Generar JavaDoc”, ahora ya se nos generará/generó correctamente la documentación de nuestros proyectos/ejercicios.