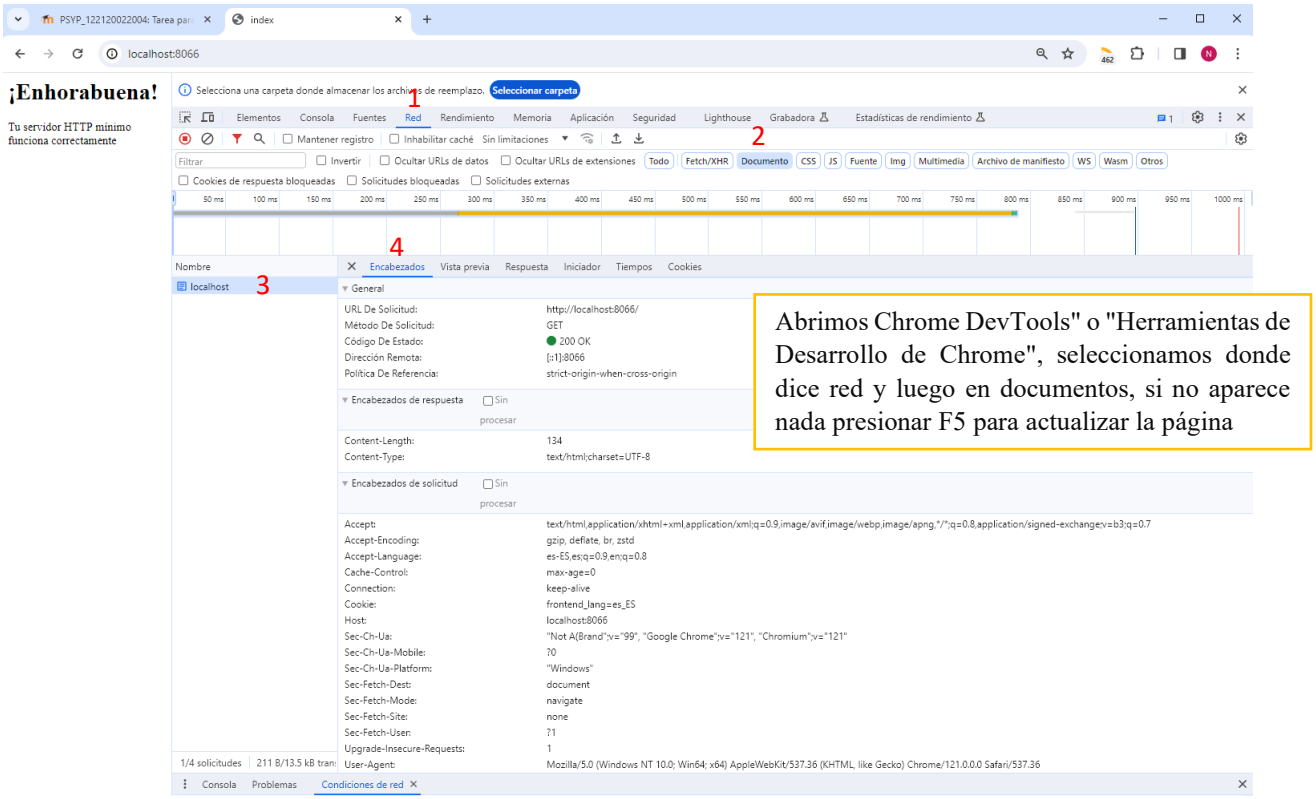
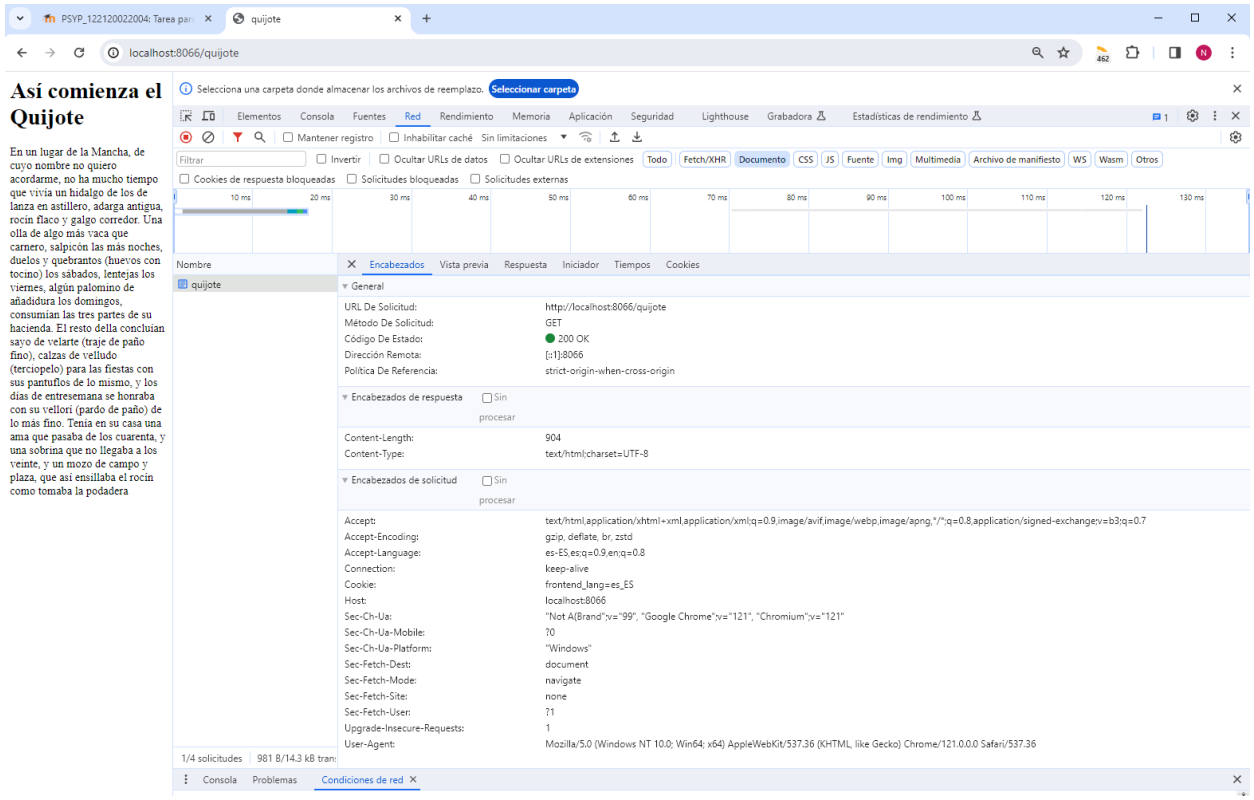
TAREA PARA PSP05

Para realizar esta tarea se ha utilizado **NetBeans 15, jdk17 y navegador web Google Chrome.**

Primero mostrare el resultado obtenido de la ejecución del ejemplo del servidor HTTP (Proyecto java ServerHTTP, apartado 5.1 de los contenidos)





EJERCICIO1: Modifica el ejemplo del servidorHTTP (Proyecto java ServerHTTP, apartado 5.1 de los contenidos) para que incluya la cabecera DATE.

- Se ha modificado el método **procesaPeticion,** donde se ha agregado las siguientes líneas de código:

```
// Código para obtener la fecha y hora actuales
Date fechaActual = new Date();
SimpleDateFormat formatoFecha = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss
z");
String fechaHTTP = formatoFecha.format(fechaActual);
```

- Se ha agregado a las respuestas del servidor la siguiente línea de código:

```
//Agregar la cabecera Date en la respuesta del servidor
printWriter.println("Date: " + fechaHTTP);
```

- Se han importado las siguientes librerías:

```
import java.text.SimpleDateFormat;
import java.util.Date;
```

- Para imprimir la cabecera en consola se ha agregado el siguiente bloque de código:

```
// Imprimir la cabecera en la consola
System.out.println("Cabecera enviada al cliente:");
System.out.println(Mensajes.lineaInicial_OK);
System.out.println("Date: " + fechaHTTP);
System.out.println(Paginas.primeraCabecera);
System.out.println("Content-Length: " + html.length());
System.out.println("\n");
```

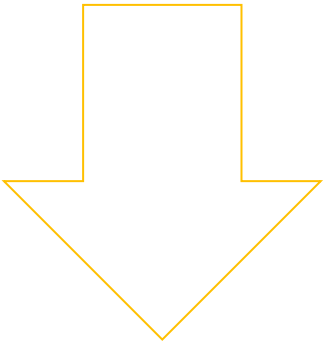
Código completo de la clase ServidorHTTP.java que incluye las modificaciones realizadas. El código encerrado en un rectángulo es el código que se ha agregado a la clase.

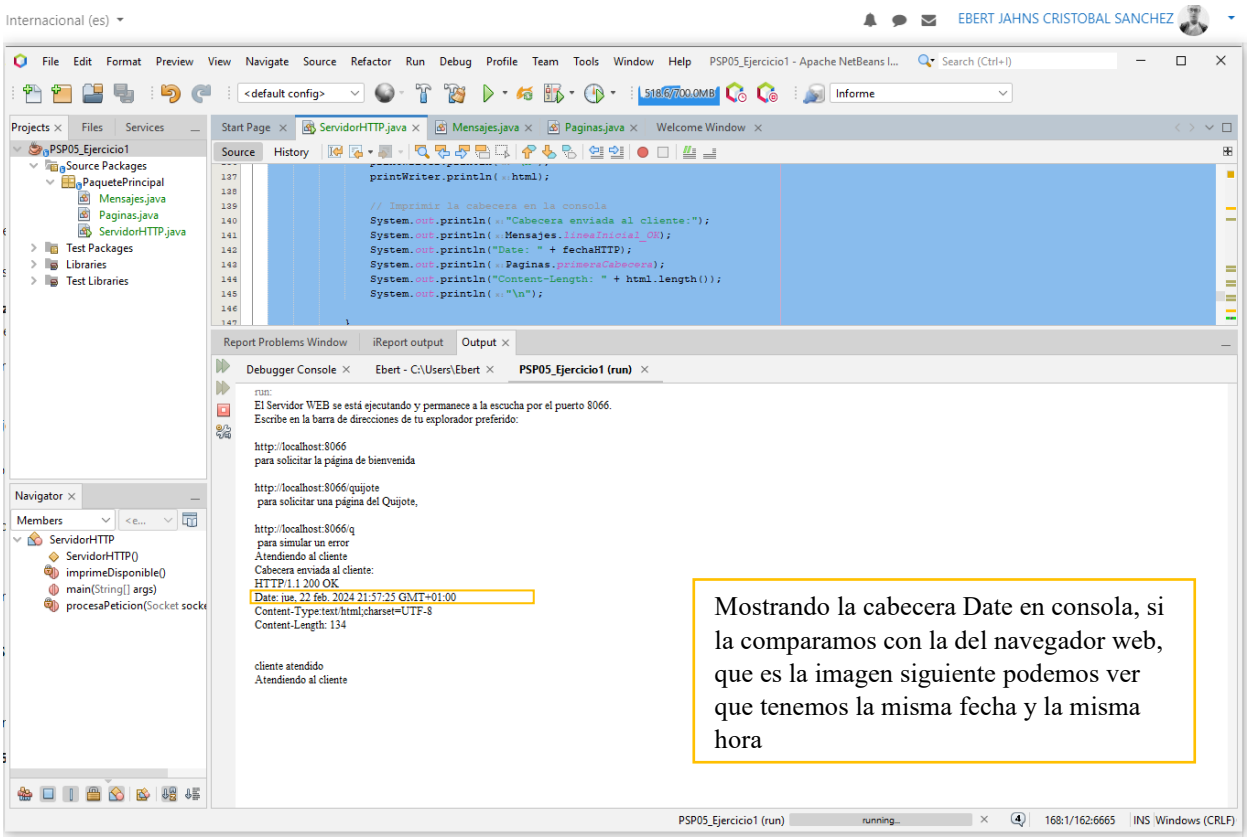
```
import java.io.BufferedReader;
import java.net.Socket;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 *
 * @author Ebert
 */
public class ServidorHTTP {

    /**
     * *********************************************************************
     * procedimiento principal que asigna a cada petición entrante un socket
     * cliente, por donde se enviará la respuesta una vez procesada
     *
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, Exception {

        //Asociamos al servidor el puerto 8066
        ServerSocket socServidor = new ServerSocket(8066);
```

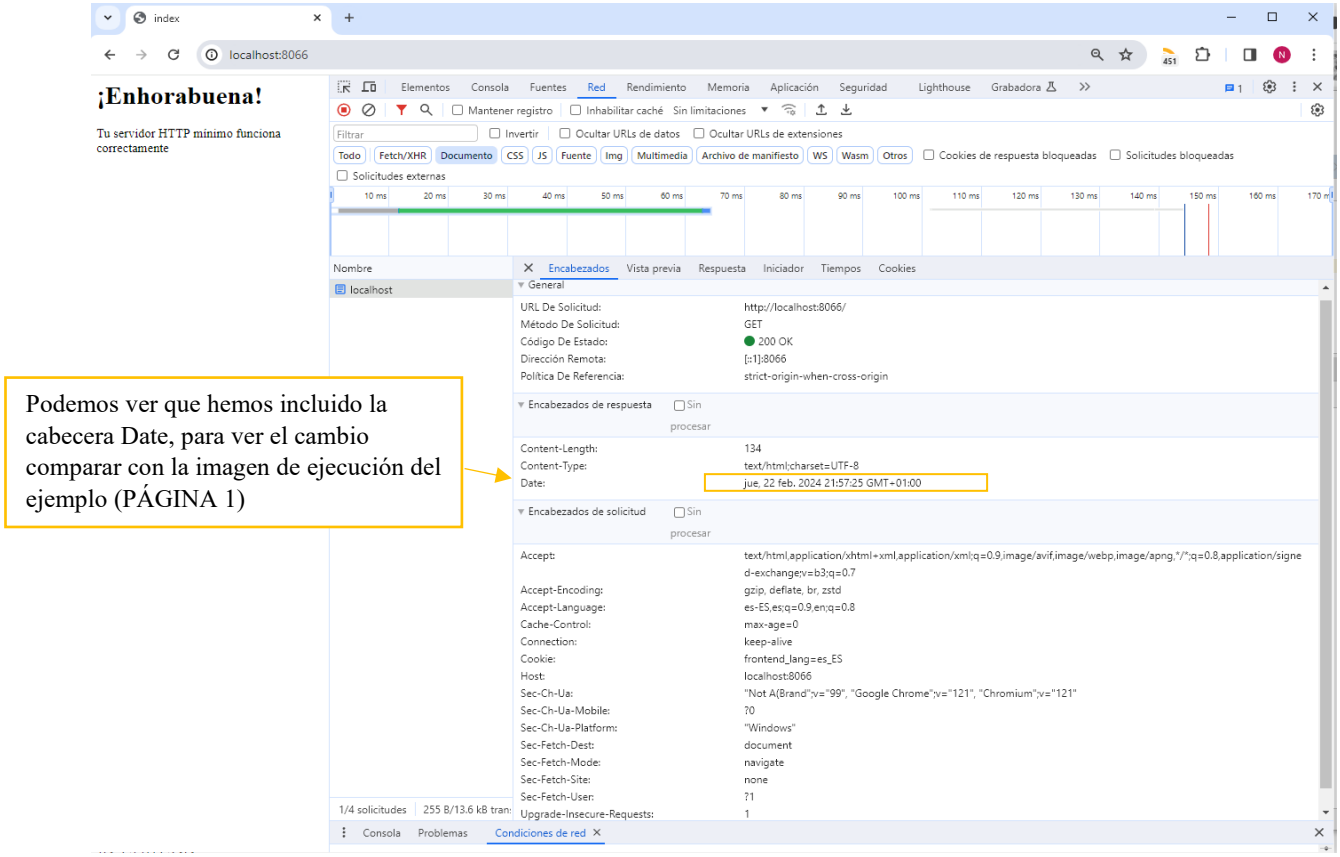
```
        imprimeDisponible();
        Socket socCliente;

        //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 ");
            procesaPeticion(socCliente);
            //cierra la conexión entrante
            socCliente.close();
            System.out.println("cliente atendido");
        }
    }

    /**
     ******************************************************************************
     * procesa la petición recibida
     *
     * @throws IOException
     */
    private static void procesaPeticion(Socket socketCliente) throws IOException {
        //variables locales
        String peticion;
        String html;

        //Flujo de entrada
        InputStreamReader inSR = new InputStreamReader(
                socketCliente.getInputStream());
        //espacio en memoria para la entrada de peticiones
        BufferedReader bufLeer = new BufferedReader(inSR);

        //objeto de java.io que entre otras características, permite escribir
        //'línea a línea' en un flujo de salida
        PrintWriter printWriter = new PrintWriter(
                socketCliente.getOutputStream(), true);

        //Código para obtener la fecha y hora actuales
        Date fechaActual = new Date();
        SimpleDateFormat formatoFecha = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z");
        String fechaHTTP = formatoFecha.format(fechaActual);

        //mensaje petición cliente
        peticion = bufLeer.readLine();

        //para compactar la petición y facilitar así su análisis, suprimimos todos
        //los espacios en blanco que contenga
        peticion = peticion.replaceAll(" ", "");

        //si realmente se trata de una petición 'GET' (que es la única que vamos a
        //implementar en nuestro Servidor)
        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("Date: " + fechaHTTP);
                printWriter.println(Paginas.primeraCabecera);
                printWriter.println("Content-Length: " + html.length());
                printWriter.println("\n");
                printWriter.println(html);

                // Imprimir la cabecera en la consola
                System.out.println("Cabecera enviada al cliente:");
                System.out.println(Mensajes.lineaInicial_OK);
                System.out.println("Date: " + fechaHTTP);
                System.out.println(Paginas.primeraCabecera);
                System.out.println("Content-Length: " + html.length());
                System.out.println("\n");

            } //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("Date: " + fechaHTTP);// Agregar la cabecera Date en la
```

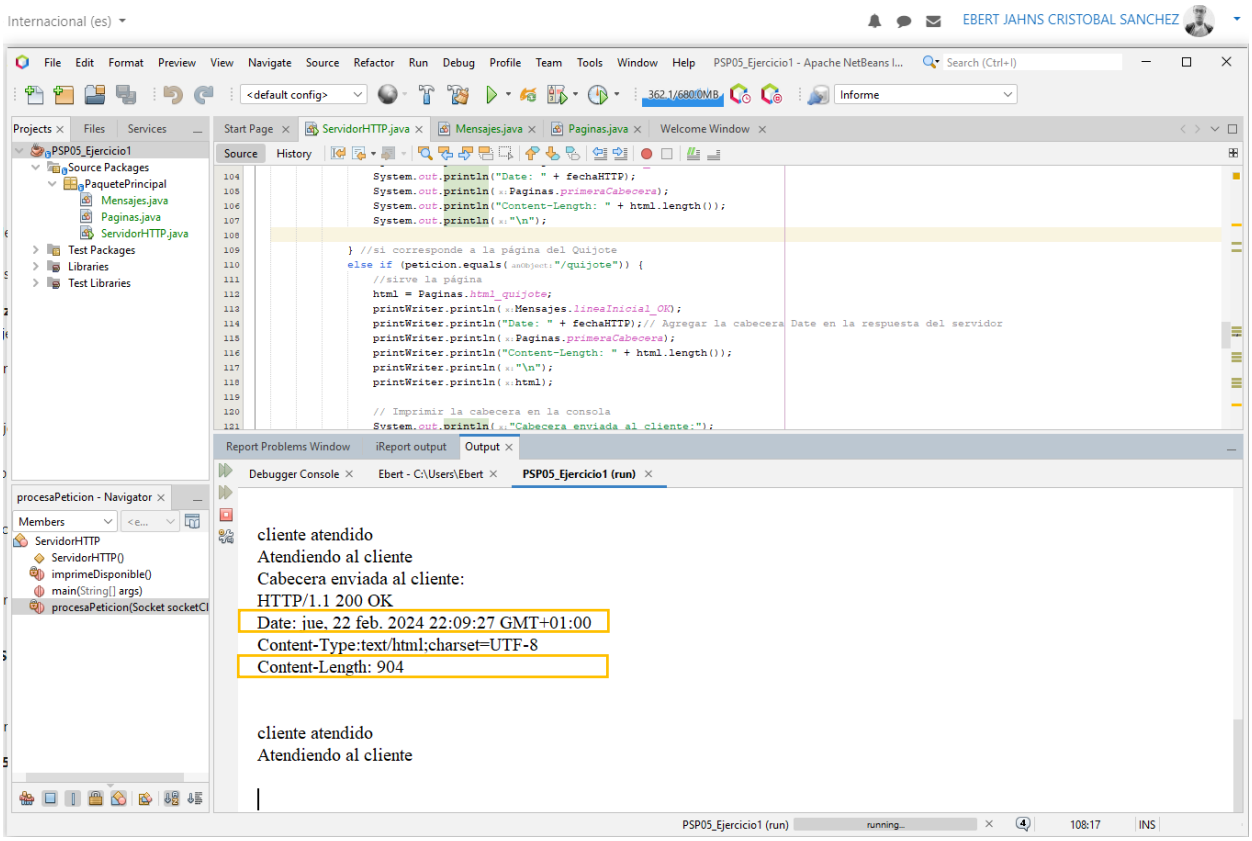
```
respuesta del servidor
                printWriter.println(Paginas.primeraCabecera);
                printWriter.println("Content-Length: " + html.length());
                printWriter.println("\n");
                printWriter.println(html);

                // Imprimir la cabecera en la consola
                System.out.println("Cabecera enviada al cliente:");
                System.out.println(Mensajes.lineaInicial_OK);
                System.out.println("Date: " + fechaHTTP);
                System.out.println(Paginas.primeraCabecera);
                System.out.println("Content-Length: " + html.length());
                System.out.println("\n");

            } //en cualquier otro caso
            else {
                //sirve la página
                html = Paginas.html_noEncontrado;
                printWriter.println(Mensajes.lineaInicial_NotFound);
                printWriter.println("Date: " + fechaHTTP); // Agregar la cabecera Date en la
respuesta del servidor
                printWriter.println(Paginas.primeraCabecera);
                printWriter.println("Content-Length: " + html.length());
                printWriter.println("\n");
                printWriter.println(html);

                // Imprimir la cabecera en la consola
                System.out.println("Cabecera enviada al cliente:");
                System.out.println(Mensajes.lineaInicial_OK);
                System.out.println("Date: " + fechaHTTP);
                System.out.println(Paginas.primeraCabecera);
                System.out.println("Content-Length: " + html.length());
                System.out.println("\n");

            }

        }
    }

    /**
     * ******************************************************************************
     * muestra un mensaje en la Salida que confirma el arranque, y da algunas
     * indicaciones posteriores
     */
    private static void imprimeDisponible() {

        System.out.println("El Servidor WEB se está ejecutando y permanece a la "
                + "escucha por el puerto 8066.\nEscribe en la barra de direcciones "
                + "de tu explorador preferido:\n\nhttp://localhost:8066\npara "
                + "solicitar la página de bienvenida\n\nhttp://localhost:8066/"
                + "quijote\n para solicitar una página del Quijote,\n\nhttp://"
                + "localhost:8066/q\n para simular un error");
    }

}
```

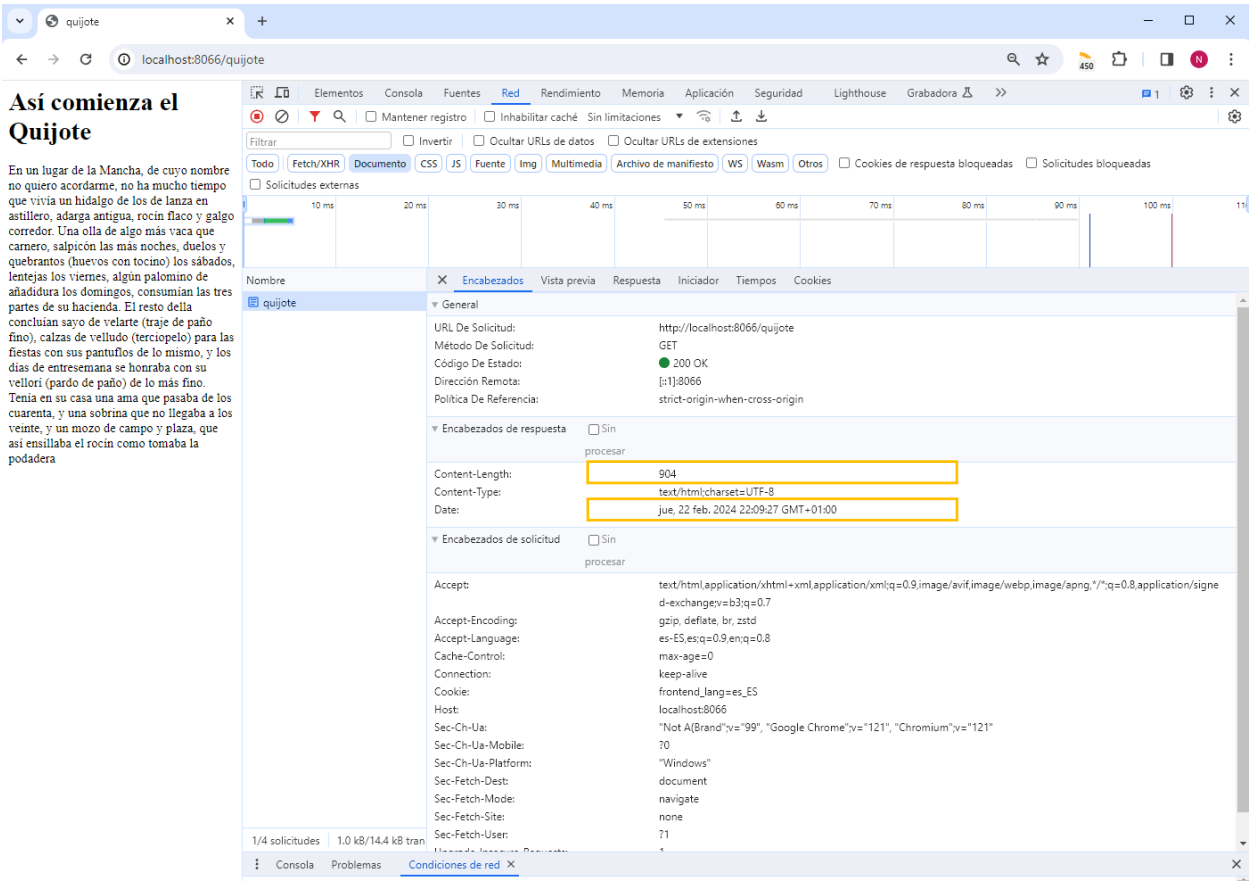
A continuación, vamos a probar el
EJERCICIO1




Internacional (es)
EBERT JAHNS CRISTOBAL SANCHEZ
Mostrando la cabecera Date en consola, si la comparamos con la del navegador web, que es la imagen siguiente podemos ver que tenemos la misma fecha y la misma hora


¡Enhorabuena!
Tu servidor HTTP mínimo funciona correctamente
Podemos ver que hemos incluido la cabecera Date, para ver el cambio comparar con la imagen de ejecución del ejemplo (PÁGINA 1)


cliente atendido
Atendiendo al cliente
Cabecera enviada al cliente:
HTTP/1.1 200 OK
Date: jue, 22 feb. 2024 22:09:27 GMT+01:00
Content-Type:text/html;charset=UTF-8
Content-Length: 904
cliente atendido
Atendiendo al cliente


localhost:8066/quijote
Así comienza el Quijote
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos (huevos con tocino) los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda. El resto della concluían sayo de velarte (traje de paño fino), calzas de velludo (terciopelo) para las fiestas con sus pantuflos de lo mismo, y los días de entresemana se honraba con su vellorí (pardo de paño) de lo más fino. Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que no llegaba a los veinte, y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la podadera
Content-Length: 904
Content-Type: text/html;charset=UTF-8
Date: jue, 22 feb. 2024 22:09:27 GMT+01:00

EJERCICIO2: Modifica el ejemplo del servidorHTTP (Proyecto java ServerHTTP, apartado 5.1 de los contenidos) para que implemente multihilo, y pueda gestionar la concurrencia de manera eficiente.

Se han realizado las modificaciones necesarias para poder realizar este ejercicio.

He creado la clase **HiloDespachador.java** y he agregado el siguiente código, donde el método **procesaPeticion(Socket socketCliente)** se ha eliminado de la clase **ServidotHTTP** y lo hemos pasado a esta clase.

Código HiloDespachador.java

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

/**
 *
 * @author Ebert
 */
public class HiloDespachador extends Thread {

    private Socket socketCliente;

    //constructor
    public HiloDespachador(Socket socketCliente) {
        this.socketCliente = socketCliente;

    }

    @Override
    public void run() {

        // Mensaje indicando que se está atendiendo al cliente
        System.out.println("Atendiendo al cliente : " + socketCliente.toString());

        try {

            // Procesa la petición del cliente
            procesaPeticion(socketCliente);
            //cierra la conexión entrante
            socketCliente.close();
            System.out.println("cliente atendido");
        } catch (IOException ex) {

            Logger.getLogger(HiloDespachador.class.getName()).log(Level.SEVERE,
null, ex);
        }

    }

    /**
     ********************************************************************************
     * procesa la petición recibida, este código lo hemos borrado de ServidorHTTP
     *
     * @throws IOException
     */
    private static void procesaPeticion(Socket socketCliente) throws IOException {
        //variables locales
        String peticion;
        String html;
```

```
        //Flujo de entrada
        InputStreamReader inSR = new InputStreamReader(
                socketCliente.getInputStream());
        //espacio en memoria para la entrada de peticiones
        BufferedReader bufLeer = new BufferedReader(inSR);

        //objeto de java.io que entre otras características, permite escribir
        //'línea a línea' en un flujo de salida
        PrintWriter printWriter = new PrintWriter(
                socketCliente.getOutputStream(), true);

        //mensaje petición cliente
        peticion = bufLeer.readLine();

        //para compactar la petición y facilitar así su análisis, suprimimos todos
        //los espacios en blanco que contenga
        peticion = peticion.replaceAll(" ", "");

        //si realmente se trata de una petición 'GET' (que es la única que vamos a
        //implementar en nuestro Servidor)
        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
                System.out.println("Cliente está conectado en
http://localhost:8066/");
                html = Paginas.html_index;
                printWriter.println(Mensajes.lineaInicial_OK);
                printWriter.println(Paginas.primeraCabecera);
                printWriter.println("Content-Length: " + html.length());
                printWriter.println("\n");
                printWriter.println(html);
            } //si corresponde a la página del Quijote
            else if (peticion.equals("/quijote")) {
                //sirve la página
                System.out.println("Cliente está conectado a
http://localhost:8066/quijote");
                html = Paginas.html_quijote;
                printWriter.println(Mensajes.lineaInicial_OK);
                printWriter.println(Paginas.primeraCabecera);
                printWriter.println("Content-Length: " + html.length());
                printWriter.println("\n");
                printWriter.println(html);
            } //en cualquier otro caso
            else {
                //sirve la página
                System.out.println("Cliente  conectado en  http://localhost:8066" +
petición + (" (No existe)"  );
                html = Paginas.html_noEncontrado;
                printWriter.println(Mensajes.lineaInicial_NotFound);
                printWriter.println(Paginas.primeraCabecera);
                printWriter.println("Content-Length: " + html.length());
                printWriter.println("\n");
                printWriter.println(html);
            }

        }
    }

}
```

En la **clase ServidorHTTP.java** se ha eliminado el código correspondiente con el metodo **procesaPeticion(Socket socketCliente)** y se ha agregado la siguiente línea de código:

```
HiloDespachador hilo;
hilo = new HiloDespachador(socCliente);
hilo.start();
```

Código ServidorHTTP.java

```
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

/**
 *
 * @author Ebert
 */
public class ServidorHTTP {

    /**
     * ********************************************************************************
     * procedimiento principal que asigna a cada petición entrante un socket
     * cliente, por donde se enviará la respuesta una vez procesada
     *
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, Exception {

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

        // Ciclo infinito para manejar conexiones entrantes de los clientes
        while (true) {
            //acepta una petición, y le asigna un socket cliente para la respuesta
            socCliente = socServidor.accept();

            //crea un nuevo hilo para despacharla por el socketCliente que le
asignó
            hilo = new HiloDespachador(socCliente);
            hilo.start();

        }
        /**
         *
*********************************************************************************
         * muestra un mensaje en la Salida que confirma el arranque, y da
         * algunas indicaciones posteriores
         */
    }


    private static void imprimeDisponible() {

        System.out.println("El Servidor WEB se está ejecutando y permanece a la "
                + "escucha por el puerto 8066.\nEscribe en la barra de direcciones
"
                + "de tu explorador preferido:\n\nhttp://localhost:8066\npara "
                + "solicitar la página de bienvenida\n\nhttp://localhost:8066/"
                + "quijote\n para solicitar una página del Quijote,\n\nhttp://"
                + "localhost:8066/q\n para simular un error");
    }
```

A continuación, vamos a probar el **EJERCICIO2:**