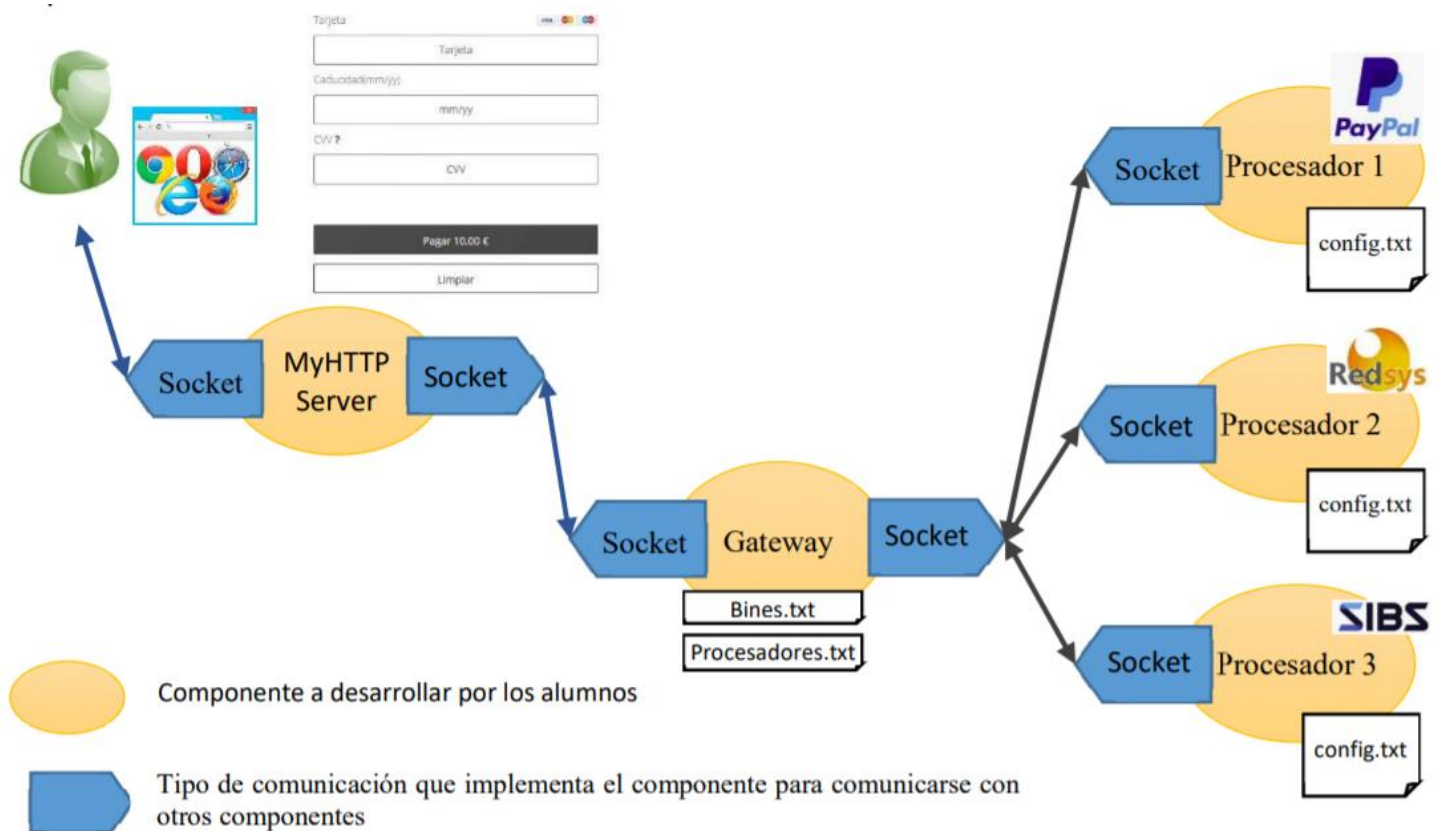


PRÁCTICA NO GUIADA SOCKETS

SISTEMAS DISTRIBUIDOS

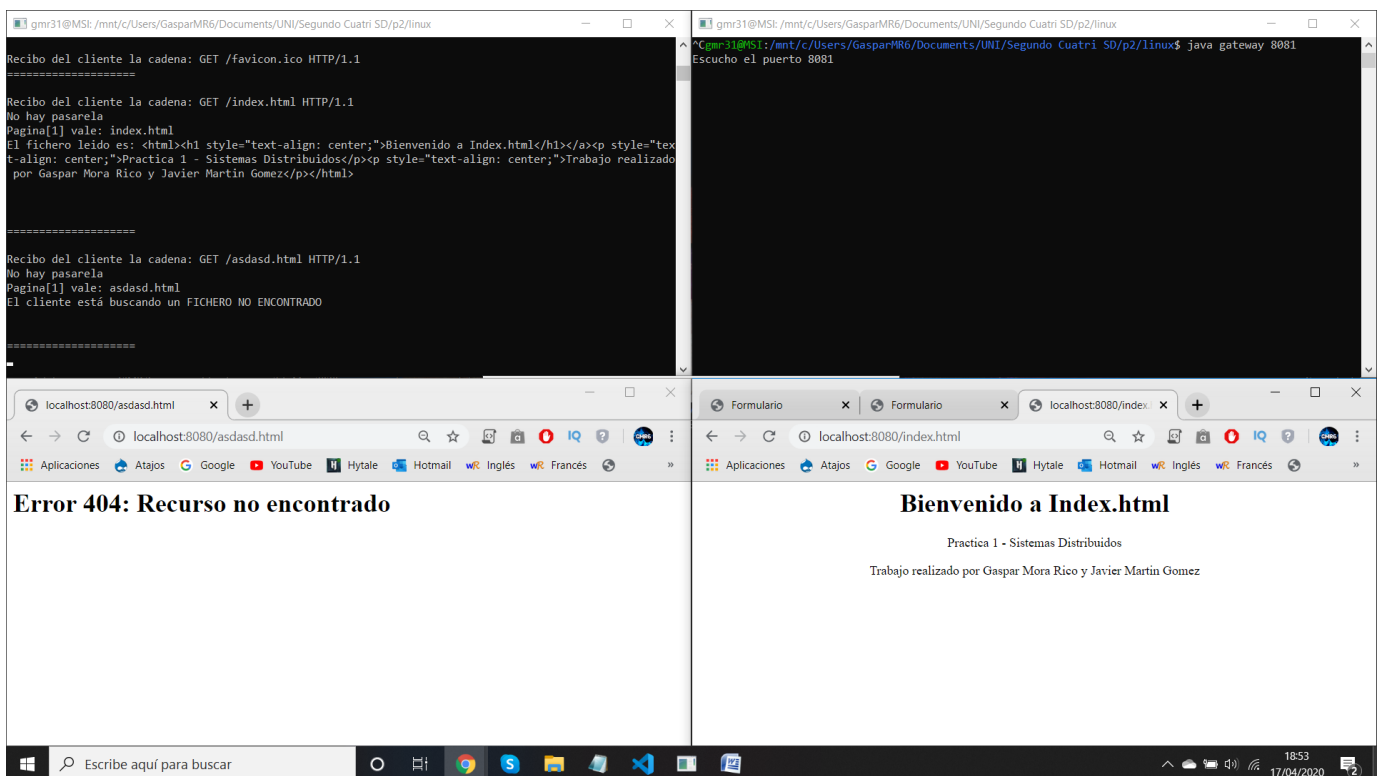


Gaspar Mora Rico - DNI: 20085930F

Javier Martín Gómez - DNI: 74016116TP

Hemos conseguido implementar la práctica utilizando varias terminales. En cada una de ellas hemos arrancado un componente (HTTPServer, Gateway, el primero procesador o el segundo). En todos los componentes podemos parametrizar sus puertos de escucha. Además, hemos creado un formulario para facilitar el funcionamiento. Cabe destacar que hemos conseguido conectar los componentes usando 2 máquinas virtuales. Sin embargo, después de cerrarlas y volver a abrirlas las IPs han cambiado de manera que ahora ambas tienen la misma e imposibilitando el funcionamiento de la misma. En el vídeo que adjuntamos lo explicamos mejor. De todos modos, en condiciones normales: con ayuda de 3 ordenadores del laboratorio sí que funcionaría con el código que te hemos adjuntado.

Podemos pedir un recurso estático de manera que si existe lo devuelve y en caso contrario muestra un error 404: de recurso no encontrado:



Hemos creado un formulario de manera que si el cliente introduce todos los datos correctos se le devolverá una respuesta de OK.

```
gm31@MSI: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux$
Recibo del cliente la cadena: GET /GatewaySD/auth?Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=23&CVV=111&Importe=124 HTTP/1.1
Hay pasarela

Escribo a localhost : 8081
He enviado a GATEWAY la cadena: GET /GatewaySD/auth?Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=23&CVV=111&Importe=124 HTTP/1.1
LRC de la cadena que paso: 116

Recibo la cadena: <STX>auth/ok/cvv=111/dia=17/mes=04/anyo=2020/importe=124<ETX>32
Recibo la cadena: auth/ok/cvv=111/dia=17/mes=04/anyo=2020/importe=124
LRC de la cadena: 32
Los LRC entre Gateway -> HTTPServer coinciden: cadena recibida con exito

Le envio al cliente: Aceptada<br>Codigo de Autorizacion: 111.<br>Fecha del pago: 17-04-2020. <br>Importe: 124 euros

=====
Recibo del cliente la cadena: GET /favicon.ico HTTP/1.1
=====

gm31@MSI: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux$
Devuelvo a Gateway: <STX>status/ok/set/1<ETX>108
=====
Sirviendo a Pasarela...

Recibo la cadena: <STX>1&Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=23&CVV=111&Importe=124&Procesador=1<ETX>88
LRC de la cadena: 88
Los LRC entre Gateway -> Procesador coinciden: cadena recibida con exito

He llega la cadena: <STX>1&Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=23&CVV=111&Importe=124&Procesador=1<ETX>88
Devuelvo a Gateway: <STX>auth/ok/cvv=111/dia=17/mes=04/anyo=2020/importe=124<ETX>32
=====

gm31@MSI: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux$
Escucho el puerto 8083
```

Aceptada
Codigo de Autorizacion: 111.
Fecha del pago: 17-04-2020.
Importe: 124 euros

Si hay algún error de conexión se devuelve un error 409. Esto podrá ocurrir si por ejemplo se nos olvida arrancar el Gateway o el Procesador o en caso de que estos estén escuchando por un puerto incorrecto.

En este caso no hemos arrancado el Gateway:

```
gm31@MSI: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux$
8081 localhost 300
Escucho el puerto 8080
Esperando a clientes...

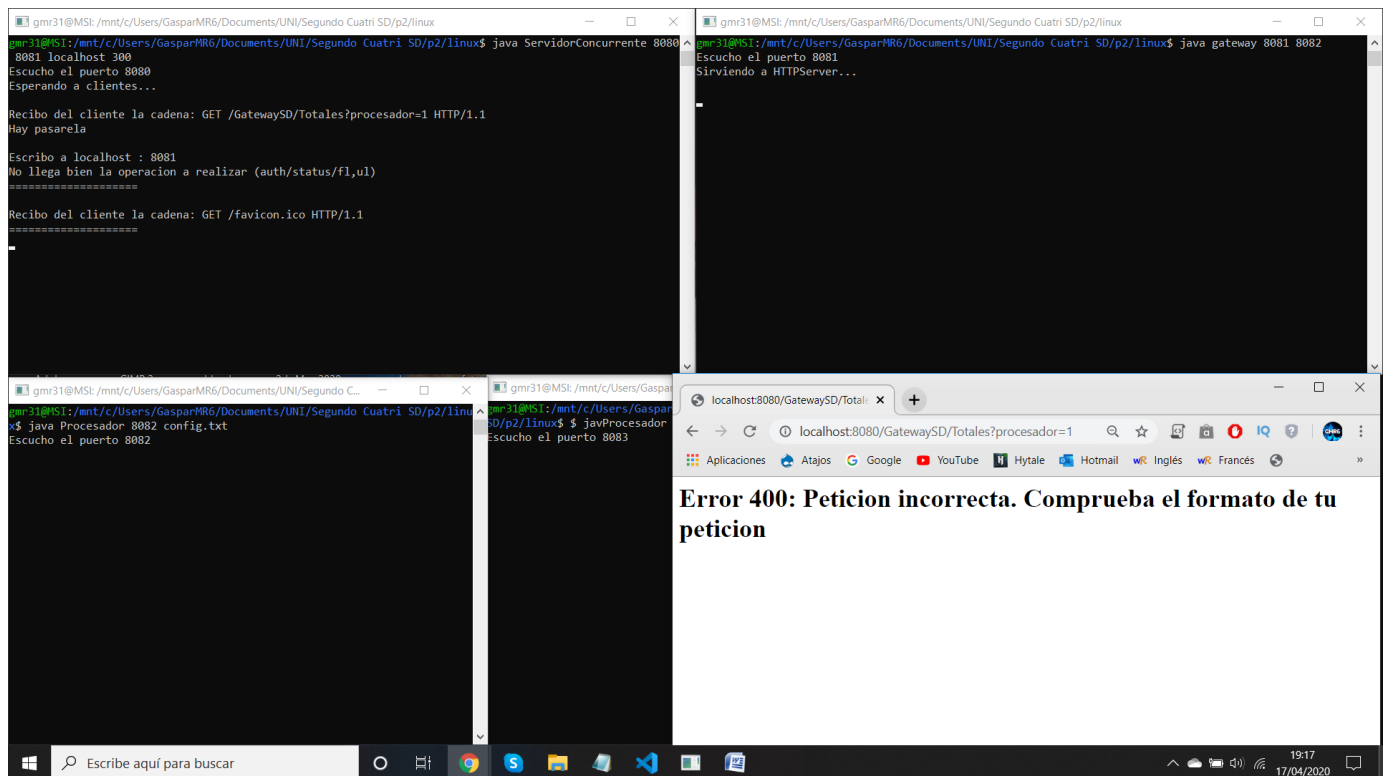
Recibo del cliente la cadena: GET /GatewaySD/status?procesador=1 HTTP/1.1
Hay pasarela

Escribo a localhost : 8081
=====
Recibo del cliente la cadena: GET /favicon.ico HTTP/1.1
=====

gm31@MSI: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux$
$ java Procesador 8082 config.txt
Escucho el puerto 8083
```

Error 409: Error interno. Intentalo de nuevo mas tarde... No has arrancado el Gateway o esta escuchando por un puerto incorrecto

Ejemplo de petición: <http://localhost:8080/GatewaySD/Totales?procesador=1>



En la operación AUTH comprobamos las siguientes condiciones:

- Mes correcto (del 1 al 12):

(<http://localhost:8080/GatewaySD/auth?Titular=Erik&Tarjeta=1111111111111111&Mes=34&Anio=11&CVV=111&Importe=1111>)

- Tarjeta caducada:

<http://localhost:8080/GatewaySD/auth?Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=11&CVV=111&Importe=123>

- Fecha de caducidad: no es posterior a 5 años desde hoy (ya que las tarjetas caducan como mucho dentro de 5 años):

<http://localhost:8080/GatewaySD/auth?Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=30&CVV=111&Importe=123>

- Y evidentemente que el importe esté dentro de los límites inferior y superior y que el estado del procesador sea activo. En el siguiente caso el límite inferior del procesador era de 5

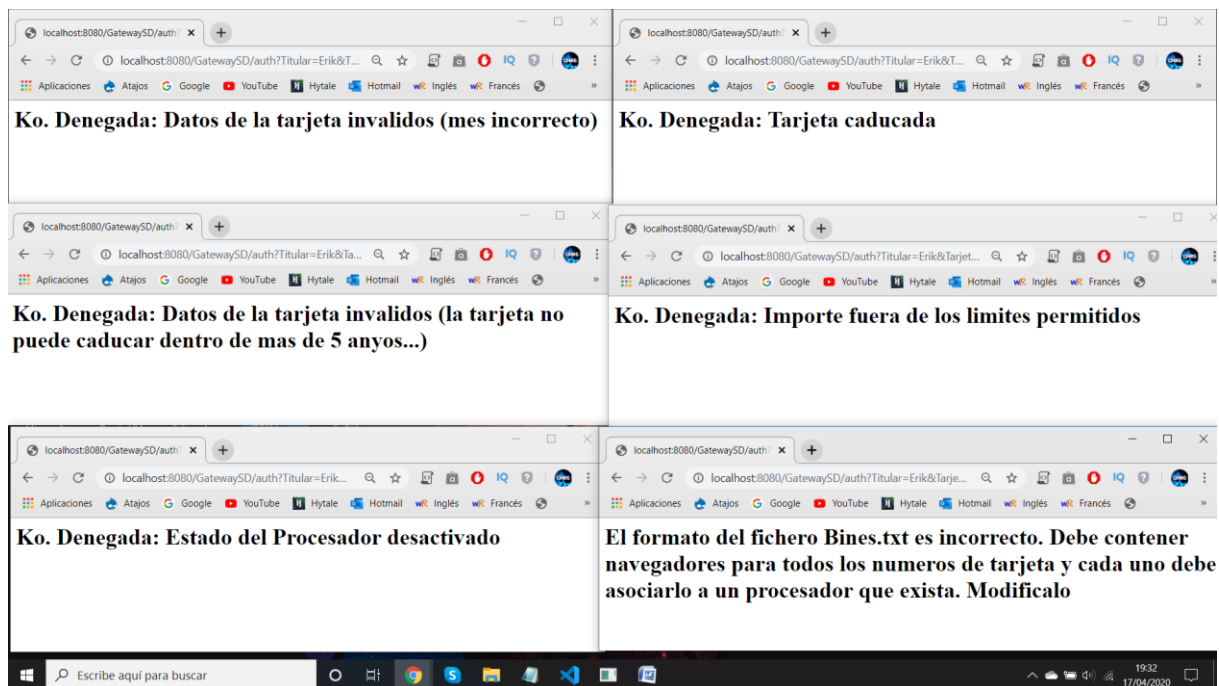
<http://localhost:8080/GatewaySD/auth?Titular=Erik&Tarjeta=1111111111111111&Mes=11&Anio=24&CVV=111&Importe=0>

- El procesador está activo:

<http://localhost:8080/GatewaySD/auth?Titular=Erik&Tarjeta=7111111111111111&Mes=11&Anio=23&CVV=111&Importe=123>

- El fichero Bines tiene un procesador asignado para el primer dígito del número de la tarjeta:

<http://localhost:8080/GatewaySD/auth?Titular=Erik&Tarjeta=9111111111111111&Mes=11&Año=11&CVV=111&Importe=321>



Siendo parte del fichero Bines.txt: 1#1 7#2 9#20 (por lo que la tarjeta que comienza por 1, llama al procesador 1, la que comienza por 7 al 2 y la que comienza por 9 da el error del fichero Bines debido a que no existe el procesador 20)

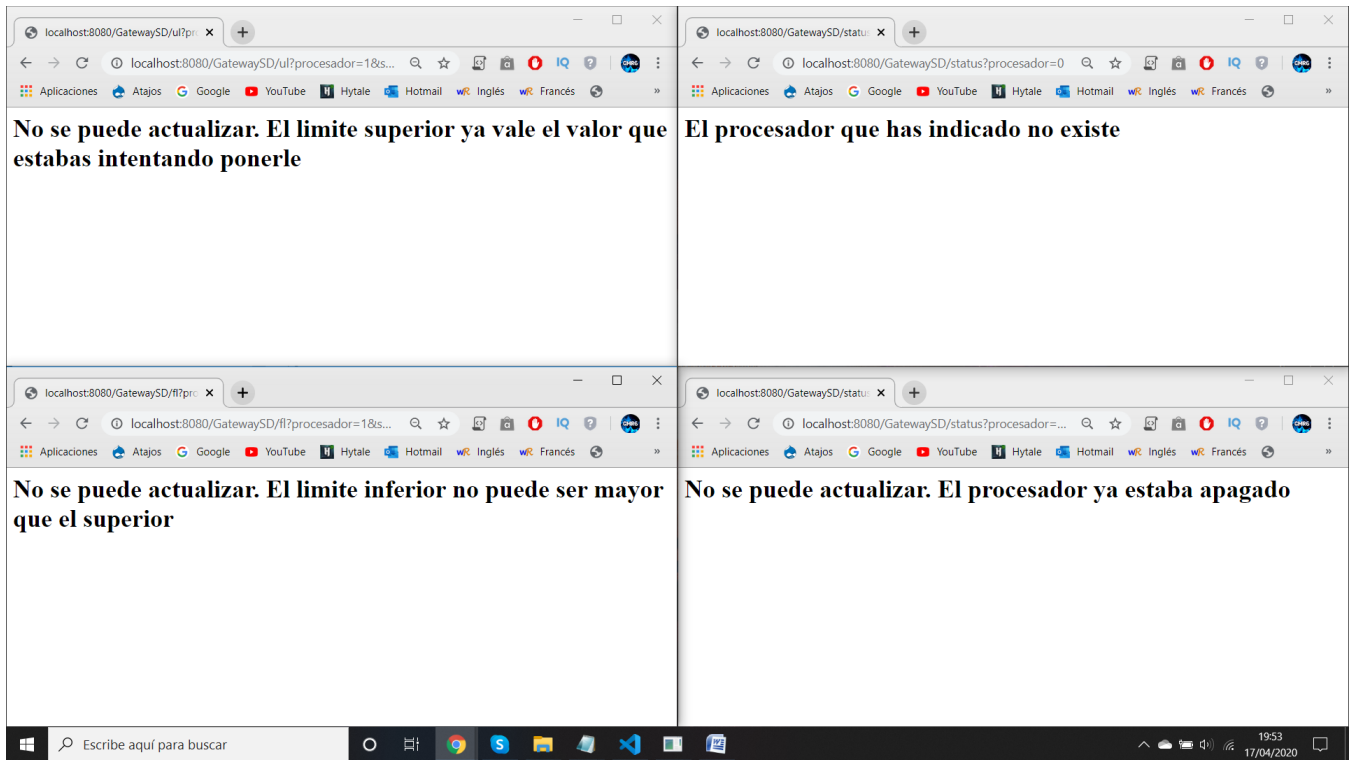
Por otro lado, en las peticiones status y de límites inferior y superior tenemos en cuenta:

- Que el procesador que se indique exista:
(<http://localhost:8080/GatewaySD/status?procesador=0>)
- Comprobamos que al modificar el límite superior el valor que introduzca el administrador sea distinto del que ya había puesto y, además, el nuevo valor del límite superior debe ser mayor que el límite inferior (en caso contrario se mostrará los dos tipos de errores).
- El sistema actuará de la misma forma si, en lugar de hablar del límite superior, nos fijamos en el límite inferior (el nuevo valor no puede ser mayor que el límite superior ni igual al que ya almacenaba el fichero Config.txt).

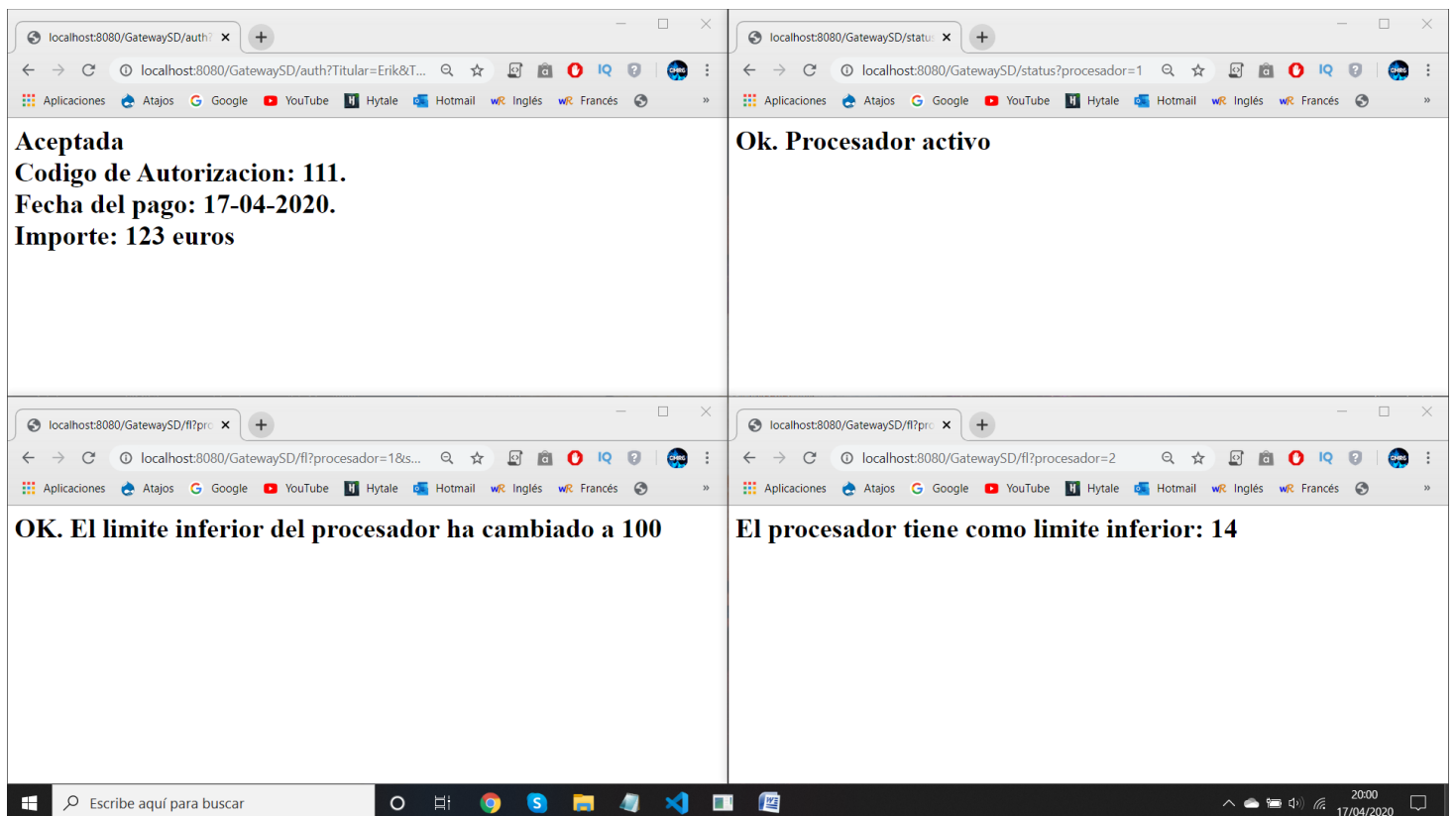
<http://localhost:8080/GatewaySD/status?procesador=1&set=0>

<http://localhost:8080/GatewaySD/ui?procesador=1&set=20000>

<http://localhost:8080/GatewaySD/fi?procesador=1&set=9999999>



Algunos casos de funcionamiento correcto son:



Hemos utilizado el estándar de empaquetado que se pedía calculando el LRC con esta función y comprobando que coincide siempre antes y después de enviar el mensaje.

```

64
65     public static int calculaLRC(String cad){
66
67         String LRC = cad;
68         int suma = 0;
69         for(int i=0; i<LRC.length();i++) {
70
71             if (i == 0) {
72                 suma = LRC.charAt(i);
73             }
74
75             else {
76                 char c = LRC.charAt(i);
77                 int ascii = (int)c;
78                 suma = suma ^ ascii;
79             }
80         }
81         return suma;
82     }
83

```

Aquí se muestra cómo se envían los mensajes siempre empaquetados con el estándar propuesto:

```

gmr31@MSK: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux
Recibo del cliente la cadena: GET /favicon.ico HTTP/1.1
=====
Recibo del cliente la cadena: GET /GatewaySD/fl?procesador=2 HTTP/1.1
Hay pasarela

Escribo a localhost : 8081
He enviado a GATEWAY la cadena: GET /GatewaySD/fl?procesador=2 HTTP/1.1
LRC de la cadena que paso: 29

Recibo la cadena: <STX>fl/ok/14<ETX>11
Recibo la cadena: fl/ok/14
LRC de la cadena: 11
Los LRC entre Gateway -> HTTPServer coinciden: cadena recibida con exito

Le envio al cliente: El procesador tiene como limite inferior: 14

=====
Recibo del cliente la cadena: GET /favicon.ico HTTP/1.1
=====

gmr31@MSK: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux
Devuelvo a Gateway: <STX>status/ok/ver/0N<ETX>95
=====
Sirviendo a Pasarela...

Recibo la cadena: <STX>4&procesador=1&set=100<ETX>86
LRC de la cadena: 86
Los LRC entre Gateway -> Procesador coinciden: cadena recibida con exito
Operacion 4
Procesador: 1 y ademas vector 1
Superior: 20000
Lo pongo en: 100
El fichero:

Devuelvo a Gateway: <STX>fl/ok/set/100<ETX>114
=====

gmr31@MSK: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux
LRC de la cadena: 114
Los LRC entre Procesador -> Gateway coinciden: cadena recibida con exito

He enviado a HTTPServer la cadena: <STX>fl/ok/set/100<ETX>114
=====
Sirviendo a HTTPServer...

Recibo la cadena: 4&procesador=2
LRC de la cadena: 29
Los LRC entre HTTPServer -> Gateway coinciden: cadena recibida con exito

Operacion UL
He enviado a Procesadores la cadena: <STX>4&procesador=2<ETX>29

Recibo la cadena: fl/ok/14
LRC de la cadena: 11
Los LRC entre Procesador -> Gateway coinciden: cadena recibida con exito

He enviado a HTTPServer la cadena: <STX>fl/ok/14<ETX>11
=====

gmr31@MSK: /mnt/c/Users/GasparMR6/Documents/UNI/Segundo Cuatri SD/p2/linux
ador=2&<ETX>90
LRC de la cadena: 90
Los LRC entre Gateway -> Procesador coinciden: cadena recibida con exito

Me llega la cadena: <STX>1&Titular=Erik&Tarjeta=7111111111111111&Mes=11&Anio=23&CVW=111&Importe=123&Proc
esador=2&<ETX>90

Devuelvo a Gateway: <STX>auth/ko/proc_apagado<ETX>32
=====
Sirviendo a Pasarela...

Recibo la cadena: <STX>4&procesador=2<ETX>29
LRC de la cadena: 29
Los LRC entre Gateway -> Procesador coinciden: cadena recibida con exito

Operacion 4
RET vale: fl/ok/14

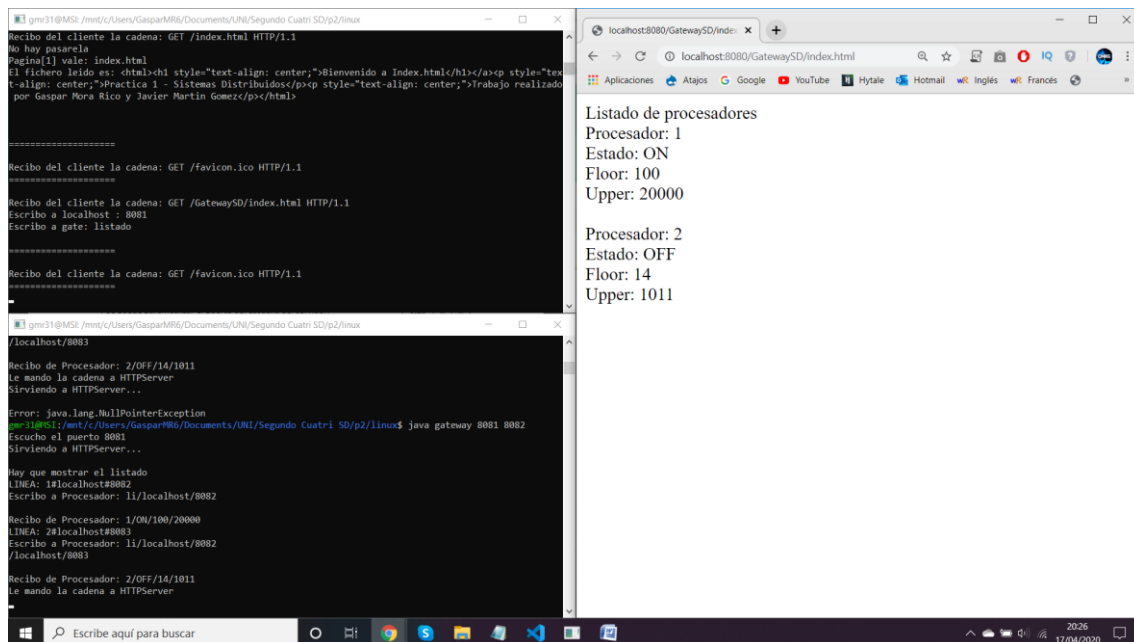
Devuelvo a Gateway: <STX>fl/ok/14<ETX>11
=====

```

Siempre devolvemos la respuesta HTTP al cliente con sus correspondientes cabeceras:

- Content-type: text/html; charset=utf-8
- Content-Length: tamaño en bytes de la página html
- Connection: close
- Server: Servidor HTTP SD

Hemos añadido un index.html en el que se muestra el listado de procesadores disponibles, su estado y parámetros: <http://localhost:8080/GatewaySD/index.html>



Como mostramos en el vídeo, es posible añadir cuando se quiera otro procesador.

En caso de que se solicite un recurso que no sea GET se devolverá el error 405 (método no soportado). Además, hemos parametrizado el número máximo de accesos simultáneos de manera que si indicamos que por ejemplo solo pueda haber 2 accesos simultáneos, el servidor solo atenderá a los 2 primeros clientes y enviará un mensaje de error a los demás. Una vez los dos primeros hayan recibido la respuesta podremos responder a otros dos más. Esto lo podemos comprobar con la función sleep (te adjuntamos un vídeo en el que se comprueba el funcionamiento de estas dos situaciones).

También hemos manejado otro error "413: Request Entity Too large". Este lo añadimos antes de que se publicara la guía de corrección para probar con distintos códigos de error. Pusimos el límite de 1000 caracteres, de manera que si se hace una petición con más larga se devolvería este error.

(El código de cada uno de los componentes se adjunta a parte)