

SISTEMA DISTRIBUIDO DE PASARELA DE PAGOS

PRACTICA NO GUIADA SOCKETS (PRACTICA
1) – SISTEMAS DISTRIBUIDOS

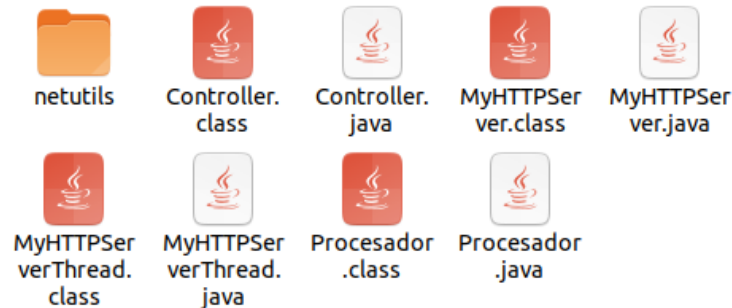
Pablo Guillén García

DNI: 48794309Q

Año académico 2019-2020

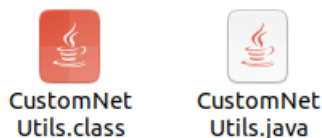
INFORME:

Esta práctica contiene unos ficheros base necesarios para su correcto funcionamiento. Son los siguientes:

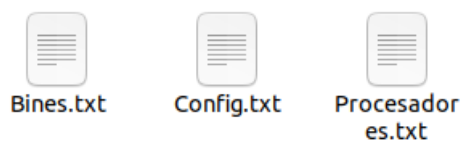


Empezando por el servidor, se divide en dos ficheros: MyHTTPServer.java(Servidor Concurrente) y MyHTTPServerThread.java(Hilo servidor). En el primero, pedimos los datos por pantalla necesarios para la conexión con el Gateway(Controlador) y después lanzamos el socket. En el segundo, que es lanzado después de aceptar el socket del primero, es donde se realizan todas las funciones del servidor, se procesa la petición, se negocian los recursos tanto estáticos como dinámicos y se tratan los errores 404 y 501, siendo este último equivalente al 405.

En cuanto al archivo Controller.java hace referencia al Controlador o Gateway del sistema. En este solicitamos el puerto de escucha y si no falla se acepta el socket y comenzamos a servir al cliente. En este momento comenzamos la conexión, comprobamos los parámetros con los del servidor, se solicita el recurso y después se finaliza la conexión. Para ayudar a comenzar y terminar las conexiones entrantes y salientes utilizamos el siguiente fichero situado dentro de la carpeta 'netutils' :



Este fichero contiene los métodos para realizar estas funciones. Además, son necesarios los ficheros de texto que se encuentran dentro de la carpeta txt que a su vez se encuentra en la carpeta res:



En el archivo bins.txt encontramos números del 1 al 9 haciendo referencia al primer número de la tarjeta de crédito de la persona que desea realizar el pago. Al lado de estos números, otro número que va del 1 al 3 haciendo referencia al procesador al que

apunta dicho número. En el archivo procesadores.txt tenemos a estos identificados por su número, con su ip respectiva y numero de puerto. En el archivo config.txt tenemos los valores de la configuración de estos procesadores.

Por último en el archivo Procesador.java pedimos el numero de puerto del procesador en cuestión y en casa de conectarse se acepta el socket y se tramita la petición del tipo que sea (AUTH, SET, GET).

GUIA DE DESPLIEGUE:

Ahora, procedamos con la explicación para desplegar la pasarela.

Requisitos:

- 3 maquinas virtuales con diferente ip.

- Ficheros fuente del programa introducidos en las maquinas previamente mencionadas.

- Conexión a internet al menos en la máquina que corresponda al servidor.

Comenzaremos por configurar las maquinas virtuales. Para cada maquina se deben instalar nettools y el compilador de java mediante los siguientes comandos:

```
sudo apt-get install default-jdk
```

```
sudo apt-get install net-tools
```

Después, debemos asegurarnos de que la maquina con el servidor tenga dos adaptadores, uno tipo NAT y otro solo host (host-only). Las otras dos maquinas solo es necesario que tengan el adaptador solo host (host-only). También es necesario, en caso de haber clonado las maquinas virtuales, generar una dirección MAC nueva en cada una de ellas para que al iniciarlas les asigne direcciones ip distintas.

Antes de empezar, debemos configurar nuestros archivos fuente para estas maquinas virtuales. Tenemos que cambiar la variable 'rootpath' de los archivos principales .java por la ruta donde se encuentre la carpeta del proyecto en dichas maquinas. Esta variable está declarada como miembro de clase al principio del archivo. Ej: rootpath = "/home/Usuario/Escritorio/CarpetaDelProyecto/res/"; .

Después debemos configurar los archivos de texto en la carpeta 'res' en la maquina virtual del Controlador, concretamente el archivo Procesadores.txt, cambiando las direcciones ip de acuerdo con las que tienen los procesadores. La dirección ip se puede consultar con el comando 'ifconfig', en este caso se debe consultar en la tercera maquina correspondiente al procesador.

Ahora debemos ejecutar y compilar cada uno de los ficheros .java en su maquina correspondiente, pero antes sería recomendable utilizar el comando 'ping' entre cada una de las maquinas para comprobar que se comunican entre ellas y no hay problemas de conexión. Ej:

Servidor 192.168.56.1

Controller 192.168.56.101

Procesador 192.168.56.102

Comprobar conexión del servidor al resto de componentes:

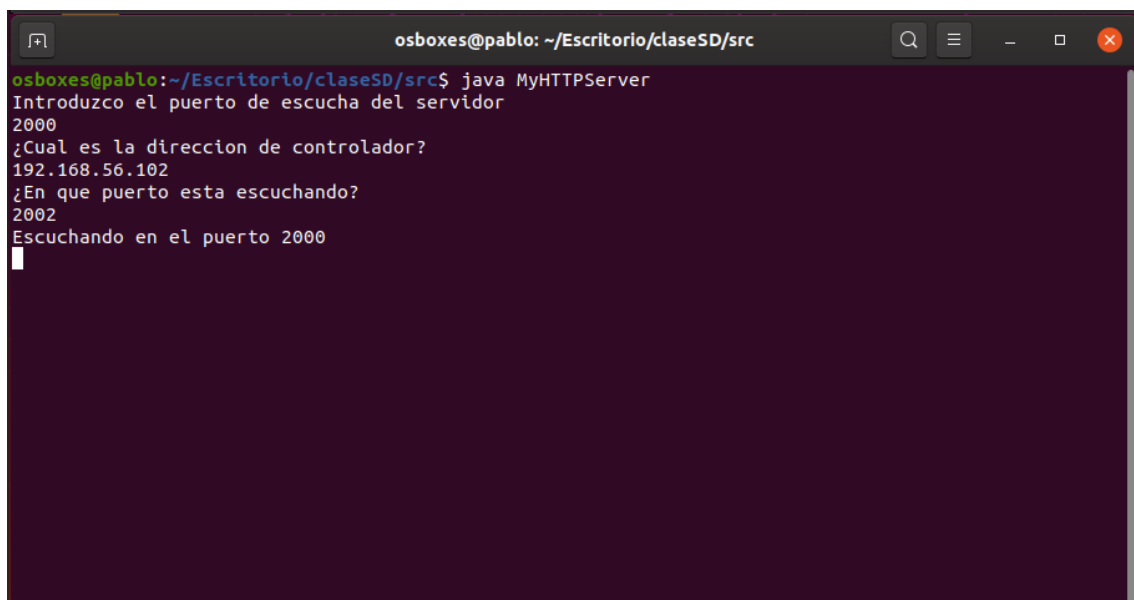
```
ping 192.168.56.101
```

```
ping 192.168.56.102
```

Una vez comprobado, se procede a ejecutar y compilar los ficheros en sus respectivas maquinas. Si funcionan debidamente y se conectan entre ellos, se podrá proceder a realizar peticiones al servidor desde el navegador.

Ahora, para comprobar el funcionamiento, vamos a realizar diferentes peticiones para ver los resultados en las siguientes imágenes. Primero, veamos un ejemplo de como configurar las terminas de cada maquina antes de realizar una petición al servidor:


-Terminal del Servidor:

A screenshot of a terminal window with a dark background. The title bar at the top reads 'osboxes@pablo: ~/Escritorio/claseSD/src'. The terminal content shows a user running 'java MyHTTPServer'. The program prompts for a listening port, with '2000' entered. It then asks for the controller's IP address, with '192.168.56.102' entered. Finally, it asks for the port the controller is listening on, with '2002' entered. The program concludes with the message 'Escuchando en el puerto 2000' and a cursor on a new line.

```
osboxes@pablo:~/Escritorio/claseSD/src$ java MyHTTPServer
Introduzco el puerto de escucha del servidor
2000
¿Cual es la direccion de controlador?
192.168.56.102
¿En que puerto esta escuchando?
2002
Escuchando en el puerto 2000
```

Con puerto 2000, comprobamos la dirección ip de la maquina virtual del controlador, en este caso la 192.168.56.102 y ponemos el puerto en el que esta escuchando el controlador.

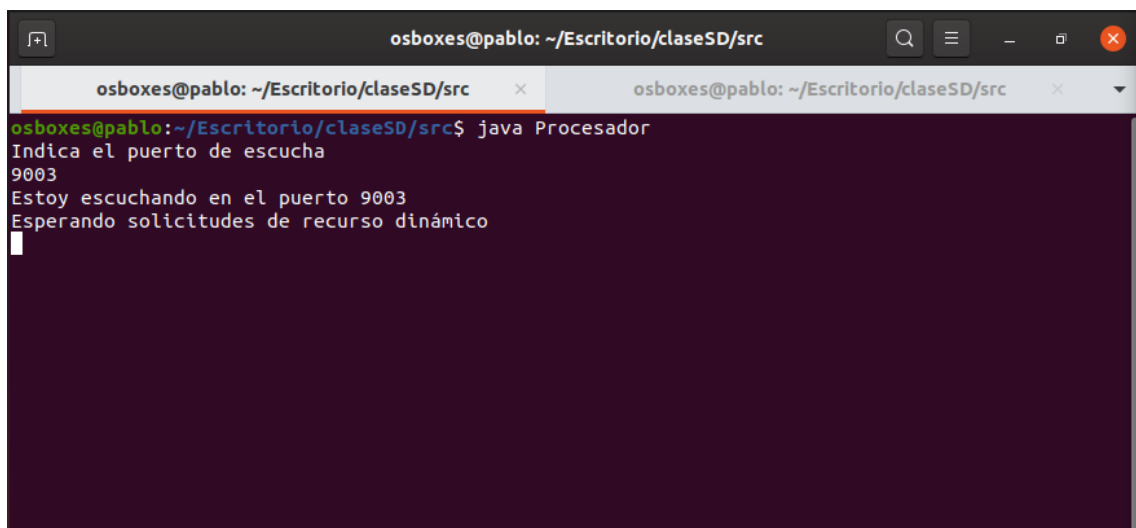
-Terminal del Controlador:



```
osboxes@pablo: ~/Escritorio/claseSD/src
osboxes@pablo:~/Escritorio/claseSD/src$ java Controller
1#192.168.56.103#9004
2#192.168.56.102#9002
3#192.168.56.103#9003
Indica el puerto de escucha
2002
Estoy escuchando en el puerto 2002
Esperando solicitudes de recurso dinámico
```

Se muestra el contenido del archivo procesadores.txt y después indicamos que en este caso el puerto de escucha del controlador va a ser el 2002, el mismo que hemos puesto antes en el servidor para establecer la conexión.

-Terminal del procesador:



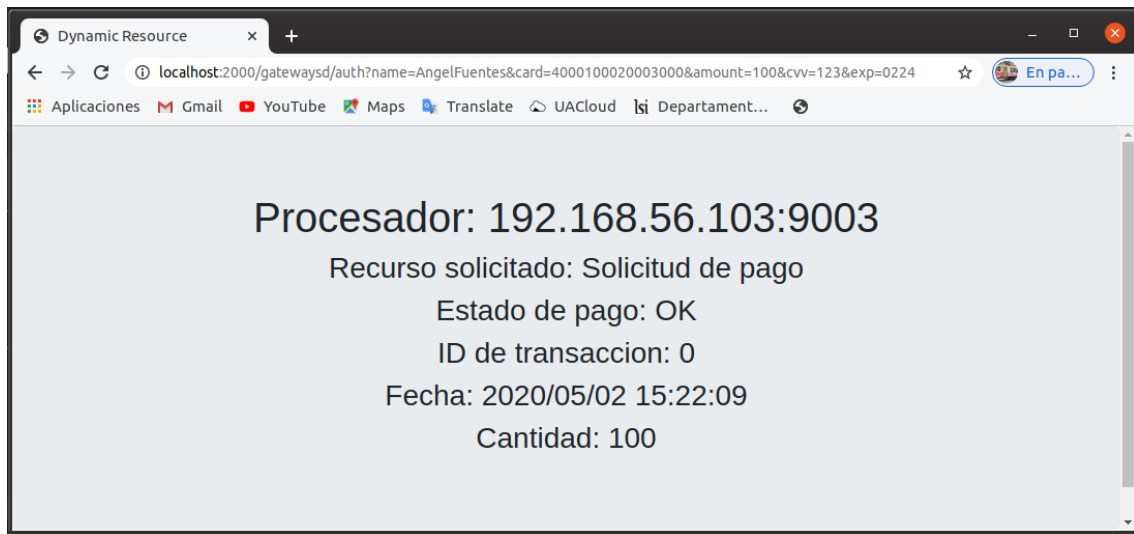
```
osboxes@pablo: ~/Escritorio/claseSD/src
osboxes@pablo:~/Escritorio/claseSD/src$ java Procesador
Indica el puerto de escucha
9003
Estoy escuchando en el puerto 9003
Esperando solicitudes de recurso dinámico
```

En este caso, vamos a hacer una petición con el primer dígito de tarjeta correspondiente al 4. Si consultamos el archivo Bines.txt, el número 4 debería apuntar al procesador 3, el cual vemos por el archivo procesadores.txt que corresponde al puerto 9003, el cual ponemos en nuestra terminal del procesador.

En este momento, procedemos a realizar peticiones al servidor.

Realizamos la siguiente petición desde el navegador del servidor:

Desde localhost con el puerto numero 2000, solicito un pago a nombre de Ángel Fuentes con número de tarjeta 4000100020003000, una cantidad de 100 euros, cvv 123 y código 0224. Resultado:



Parece que todo ha salido bien, pero para asegurarnos miramos las terminales para ver la respuesta de cada uno de los componentes:

-Servidor

```
osboxes@pablo: ~/Escritorio/claseSD/src
osboxes@pablo:~/Escritorio/claseSD/src$ java MyHTTPServer
Introduzco el puerto de escucha del servidor
2000
¿Cual es la direccion de controlador?
192.168.56.102
¿En que puerto esta escuchando?
2002
Escuchando en el puerto 2000
Sirviendo Cliente...
Escuchando en el puerto 2000
Sirviendo Cliente...
Escuchando en el puerto 2000
GET /gatewaysd/auth?name=AngelFuentes&card=4000100020003000&amount=100&cvv=123&exp=0224 HTTP/1.1
Estoy solicitando un recurso
sOC/ENQ
sOC/STX
/gatewaysd/auth?name=angelfuentes&card=4000100020003000&amount=100&cvv=123&exp=0224
ACK
2#OK#0#2020/05/02 15:22:09#100
eOCACK
eOCEOT
2#OK#0#2020/05/02 15:22:09#100
--Cliente Servido--
```

-Controlador

```
osboxes@pablo: ~/Escritorio/claseSD/src
osboxes@pablo:~/Escritorio/claseSD/src$ java Controller
1#192.168.56.103#9004
2#192.168.56.102#9002
3#192.168.56.103#9003
Indica el puerto de escucha
2002
Estoy escuchando en el puerto 2002
Esperando solicitudes de recurso dinámico
Sirviendo Cliente...
NEGOCIANDO
sIC/ACK
CONECTADO AL SERVIDOR HTTP
/gatewaysd/auth?name=angelfuentes&card=4000100020003000&amount=100&cvv=123&exp=0224
auth?name=angelfuentes&card=4000100020003000&amount=100&cvv=123&exp=0224
sOC/ENQ
sOC/STX
ACK
2#OK#0#2020/05/02 15:22:09#100
eOCACK
eOCEOT
eIC2#OK#0#2020/05/02 15:22:09#100
eICEOT
Escribiendo respuesta HTML...
Respuesta: 2#OK#0#2020/05/02 15:22:09#100
--HTTPServer Servido--
```

-Procesador

```
osboxes@pablo: ~/Escritorio/claseSD/src
osboxes@pablo:~/Escritorio/claseSD/src$ java Procesador
Indica el puerto de escucha
9003
Estoy escuchando en el puerto 9003
Esperando solicitudes de recurso dinámico
Sirviendo al Controlador...
sIC/ACK
Tramitando
2#angelfuentes#4000100020003000#100#123#0224
Request ACK
Estado=ON
Montando respuesta de pago...
eIC2#OK#0#2020/05/02 15:22:09#100
eICEOT
--Controlador Servido--
```

Los mensajes de cliente servido, HTTPServer servido y Controlador servido, nos indican que han llegado a realizar cada uno su función.

Ahora, comprobaremos el estado del procesador con el comando status apuntando al procesador 3 que es el que estamos usando en este momento:



Como vemos el procesador existe y está encendido.

Veamos que sucede cuando llamamos a un procesador que existe pero no esta encendido:

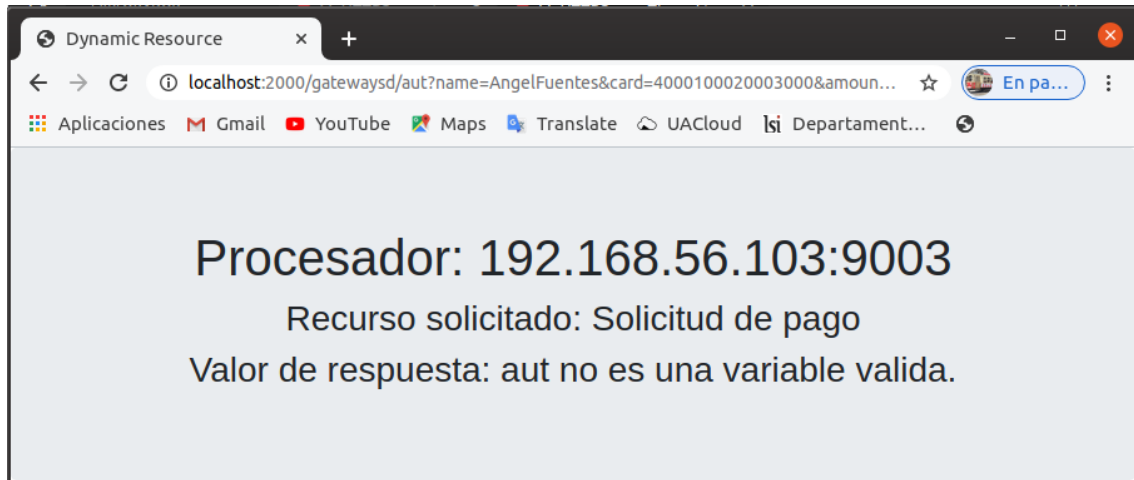


Y al llamar a uno que no existe:

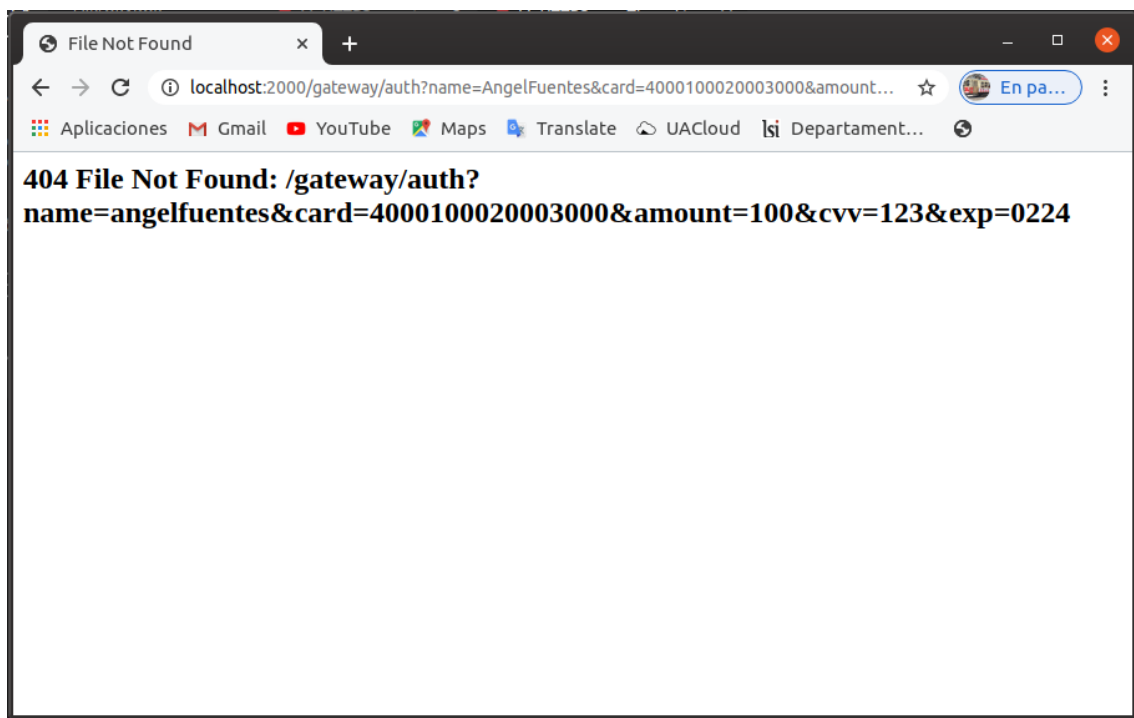


Ahora veamos algunos otros errores posibles.

Si introducimos una variable no valida como auth o status:

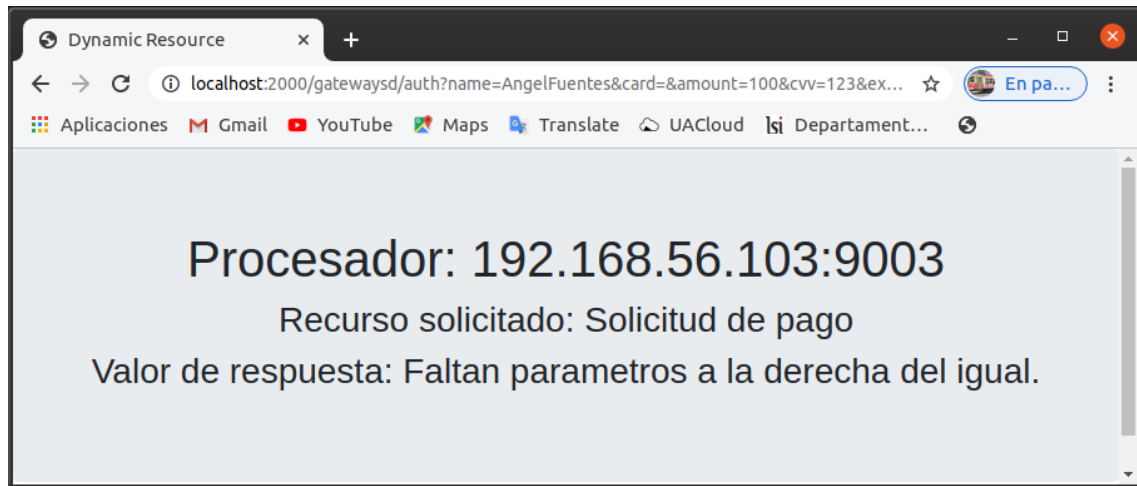


Si no ponemos bien el archivo en este caso 'gatewaysd' provocamos el error de navegador 404:



Si utilizamos un método distinto a GET provocaremos el error 501(similar al 405) de método no implementado, el cual no veremos demostrado por pantalla ya que habría que crear otro método que no fuera GET para poder llamarla y no se puede hacer desde el navegador.

Si faltan parámetros a la derecha de uno de los iguales en los datos de la petición web:



También tengo implementados por todo el código mensajes de error en caso de que falle la conexión en cualquier punto para indicarlo debidamente.