Curso | 2011 | 2012

# Práctica 5 Redes de Computadores

David Morales Sáez Alberto Manuel Mireles Suárez

### Introducción

En esta práctica, nuestro objetivo es la implementación de un proxy básico que sea capaz de redireccionar las peticiones http entrantes desde el cliente hacia el servidor que aloja la web. A su vez, deberá enviar las respuestas del mismo al cliente. Para ello, nos hemos basado en el código suministrado en el guión de la práctica, el cual estaba incompleto.

# Desarrollo de la práctica

Con el fin de realizar la práctica, hemos tenido que entender el funcionamiento de las clases usadas en el código suministrado y las clases necesarias para su manejo. Se trata de las clases:

- BufferedReader: Una clase que permite leer texto desde un buffer de entrada.
- InputStreamReader: Es el puente que permite la recepción de datos.
- Socket: Es uno de los extremos de la conexión entre dos máquinas.
- DataOutputStream: Permite escribir datos en un puente hacia un servidor.
- ServerSocket: Implementa un servidor de sockets, es decir, un servidor que espera solicitudes en la red.
- DataInputStream: Permite leer datos en un puente desde un servidor.

El código modificado de la práctica se encuentra en los ficheros adjuntos.

#### **Funcionamiento**

Al ejecutar la aplicación, podemos escoger entre dos modos, el modo depuración, el cual muestra todas las peticiones y los paquetes recibidos y el modo limpio, donde no se muestra ningún tipo de mensaje salvo que ocurra una excepción. A su vez, hemos de tener en cuenta que se ha de escoger un puerto que esté libre, para ello podemos consultarlo con el comando "netstat –a", el cual muestra todas las conexiones abiertas del equipo.

Una vez iniciado el proxy, este se inicializa abriendo una conexión en el puerto pasado por parámetros y establece el modo de ejecución. Un ejemplo de sentencia para ejecutarlo es el siguiente:

```
java ProxyCache 1026 1
```

El proxy estará continuamente esperando recibir una petición GET por el puerto. Una vez obtenida se obtendrá la cabecera, de donde obtendremos el servidor de destino. Se creará una conexión con este servidor y se enviará la petición recibida.

A continuación, esperaremos a obtener una respuesta del servidor; una vez obtenida, se fusionarán los distintos paquetes, hasta un tamaño ya definido, y se enviarán al cliente. Una vez completado este proceso se cerrará la conexión y se esperará a recibir una nueva petición.

## Conclusión

Tras haber realizado varias pruebas, hemos podido comprobar que cierto tipo de paquetes (pertenecientes a publicidad en las webs) bloquean el servidor estando en una espera continua. Esto se debe a que el servidor implementado es muy básico y solo es capaz de manejar peticiones HTML, mientras que algunas solicitudes no utilizan ese protocolo. Esto ocurre, por ejemplo, al acceder a la web de "www.elmundo.es".