

## Práctica de Servidor Web en Java

2015

v1.0

### Descripción

La práctica consiste en programar un servidor web mínimo en Java, que sirva el contenido de un directorio y todo lo que este contenga. El servidor debe servir las páginas (el directorio) en la dirección `localhost:8181` y aceptar conexiones de un cliente de control (detallado más adelante) en la dirección `localhost:9090`.

El servidor puede hablar HTTP 1.1 en su versión más simple y limitarse a atender peticiones de tipo GET haciendo caso omiso de todas las opciones de HTTP.

Para servir las páginas (que corresponden a los ficheros en el disco) debe utilizar una cache, de tal forma que si la cache no tiene la página solicitada, o la versión de la página en la cache es más vieja que el fichero en el disco, debe cargarse desde el disco el fichero en la cache. En cualquier caso, el contenido de la página servida procede de la cache.

La cache no puede mantener en ningún caso más de 5 ficheros a la vez.

Hay que implementar tres políticas de reemplazamiento en la cache, esto es, tres formas de decidir cuándo hay que echar algo de la cache antes de meter otra cosa.

1. FIFO: Se expulsa la página que antes entró en la cache antes de incluir otra (una vez alcanzado el límite de páginas)
2. LRU: Se expulsa la página que hace más que no se utiliza en la cache.
3. RND: Se expulsa una página al azar.

El servidor se debe poder ejecutar como en el ejemplo que sigue:

```
$ jweb ~/mywebdir
started
```

Como puede verse, el servidor se queda ejecutando en primer plano.

El cliente de control del servidor es un programa que acepta en la línea de comandos (como argumento) una petición para el servidor (no HTTP, se trata de una petición de control). El cliente ha de conectarse al servidor y enviar la petición de control y recibir su respuesta.

Naturalmente, el servidor debe concurrentemente atender peticiones HTTP y una o mas peticiones de control.

Las peticiones de control son las siguientes. Para cada una se incluye un ejemplo que debe funcionar exactamente igual en la práctica realizada:

- flush: debe vaciar la cache.

```
$ jclient flush
ok
```

- delete: debe eliminar una página de la cache si está o informar de que no está. El nombre de la página es un camino absoluto considerando como raíz el directorio que se sirve al web.

```
$ jclient del /a/b.html
ok
$ jclient del /a/b.html
not found
```

- list: debe listar las páginas que hay en la cache, su tamaño en bytes y cuándo entraron en la caché.

```
$ jclient ls
/x.html    1345 bytes Thu May 12 17:30:02 GMT 2015
/a/b.html  231 bytes Thu May 12 17:30:03 GMT 2015
/a/c.html  8543 bytes Thu May 12 17:30:04 GMT 2015
```

En este caso no se permite enviar la fecha en el protocolo como un string.

- policy: cambia la política de la cache (fifo, lru o rnd).

```
$ jclient policy fifo
changed from rnd to fifo
```

- quit: debe hacer que el servidor termine su ejecución.

```
$ jclient quit
ok
```

Los mensajes deben tener:

- los enteros aplanados en **little** endian en 4 bytes.
- las fechas codificadas como número de nanosegundos desde el Epoch en 64 bits sin signo en little endian.
- los nombres y strings codificados en UTF-8

Todo el fuente debe estar contenido en los paquetes `lab15.XXX.server` (para el servidor), y `lab15.XXX.client` (para el cliente). En ambos casos, XXX debe ser tu nombre de usuario en Linux. Cualquier otro paquete auxiliar que necesites debe estar dentro de `lab15.XXX` utilizando el mismo convenio para XXX.

El código debe estar limpio y debe utilizar las técnicas de orientación a objetos donde corresponda.