

# Prácticas Concurrencia y Distribución (16/17)

Arno Formella, Francisco Rodríguez Martínez,  
Javier Rodeiro Iglesias, David Olivieri

semana 24–28 abril

## 5. Una aplicación simple distribuida

Retomamos la actividad 2 donde implementamos concurrencia simple para trabajar con una imagen. En vez de lanzar hilos en la misma máquina queremos usar o bien otros procesos en el mismo ordenador o bien otros procesos en otro(s) ordenador(es). Los requisitos con más detalle son:

1. Se implementa un sistema cliente-servidor.
2. El servidor trabaja con una estructura de datos para las tareas. Las tareas se distribuyen a los clientes cuanto estos realizan peticiones.
3. Con los resultados recibidos como respuestas de los clientes el servidor compone la imagen resultante.
4. Para cada cliente que se conecta al servidor se usa en el servidor un hilo para realizar la gestión, es decir, para mandar la tarea y esperar la respuesta.
5. Se transmite un objeto desde el servidor al cliente para indicar los parámetros para calcular (básicamente la región de la imagen). Se devuelve un objeto desde el cliente al servidor con el resultado.

Con estos requisitos transforma el programa concurrente de la actividad 2 que realizaba la binarización en un programa distribuido con la siguiente ayuda:

1. El servidor se preocupa de la lectura y escritura de la imagen (igual como ya realizado en la actividad 2).
2. En vez de crear los hilos trabajadores en un bucle fijo, se espera para que se conecten los clientes y se crea para cada cliente el hilo que se dedica a la comunicación.
3. La comunicación entre servidor y clientes se realiza con `sockets` y flujos sobre estas conexiones tipo `ObjectOutputStream` y `ObjectInputStream`.
4. El objeto que se transmite al cliente contiene los parámetros de la región sobre la cual este cliente debe actuar, es decir, coordenadas, zona de la imagen, y umbral.

5. El objeto que se transmite al servidor contiene en la región los valores del resultado de la operación que el hilo de comunicación copia a la imagen de resultado.
6. Para realizar la sincronización al final, es decir, que el servidor determina que todos los hilos trabajadores respectivamente clientes, hayan terminado se usa un `CountDownLatch` de forma adecuada. (Recomendación: usa el `CountDownLatch` primero en una modificación de la actividad 2 para sustituir el uso de `join`.)