

Javier Romero Castro DNI: 44659463-H
Nara Menor Sierra DNI: 44479308-E
Esteban Puentes Silveira DNI: Y1883497H

5.1 Elabora un paquete de servidor de tal manera que el servidor esté a la espera de recibir peticiones de trabajo de clientes.

Creamos una clase servidor extendida de thread que controla todos los hilos de conexiones entrantes e implementa el método Main que establece el número de iteraciones necesarias para completar las fotos y se queda a espera de ellas.

5.2 Cuando un cliente pide trabajo al servidor, el servidor asigna a tal cliente un bloque de la imagen de tamaño adecuado para que el cliente calcule la parte correspondiente de la imagen, es decir, el cliente devuelve los píxeles calculados según la especificación del servidor (coordenadas de la ventana y parámetros necesarios).

En el código que creamos el cliente envía un paquete petición al servidor y este le entrega las coordenadas Y de inicio y fin los píxeles de la iteración que debe calcular, y una vez calculada son enviadas de vuelta al servidor.

5.3 El servidor colecciona todos los datos devueltos por los clientes hasta que la imagen esté completo, en cual momento escribe el fichero resultante.

Nuestro código en la parte servidor recibe la nueva imagen calculada y sustituye los píxeles antiguos con los nuevos en su lugar correspondiente. A medida que van llegando conexiones con resultados se van “concatenando” por decirlo de alguna forma, con la imagen que deberá quedar finalizada tras todas las iteraciones

5.4 Considera en el uso del patrón de diseño productor/consumidor para la implementación del control del flujo de datos

De la forma que implementamos el código el servidor se encuentra permanentemente esperando por nuevas conexiones (no intenta tomar resultados del buffer) y es capaz de gestionar todas las conexiones de forma que no se llene el buffer de conexiones. Utilizamos los ObjectOutputStream/InputStream para lograr la espera y recibimiento de los datos.

5.5 Al lanzar el servidor se debe especificar por lo menos la región del conjunto Mandelbrot por calcular, la resolución de la misma, el número máximo de iteraciones, y el nombre de fichero de salida

Explicado anteriormente, básicamente se crea la imagen vacía que se va rellenando a medida que se van resolviendo las conexiones clientes, las

conexiones necesarias se almacenan en una lista enlazada que se irá vaciando a medida que lleguen conexiones y en las cuales se almacena el número de coordenada Y donde comienza la iteración y el número de coordenada Y donde termina.

5.6 Al lanzar el cliente se debe especificar por lo menos los datos necesarios para contactar al servidor

Lo realizamos de la forma indicada mediante el uso de sockets y ObjectOutputStream/InputStream