

Programación de Servicios y Procesos

Programación multihilo

Ejercicio 02

Crea una aplicación de servicio que funcione de la siguiente forma:

- La configuración se leerá de un archivo que se llamará `config.properties` que estará almacenado en la carpeta de trabajo de la aplicación. El formato será Properties (accesible mediante la clase `java.util.Properties`). Este archivo puede tener las siguientes opciones:
 - `inputfolder`. Carpeta de entrada. En esta carpeta se irán dejando las imágenes sobre las que se va a trabajar. Valor por defecto: La carpeta de trabajo de la aplicación.
 - `outputfolder`. Carpeta de salida. En esta carpeta se irán generando las imágenes una vez procesadas. Valor por defecto: Carpeta llamada `output` situada dentro de la carpeta de trabajo.
 - `maxwidth`. Ancho máximo de imagen. La imagen de salida no puede tener un ancho mayor que este. Valor por defecto: 100
 - `maxheight`. Alto máximo de imagen. La imagen de salida no puede tener una altura mayor que ésta. Valor por defecto: 100
- Tendrá un hilo principal, que vamos a llamar hilo maestro o *dispatcher*. El trabajo del dispatcher es monitorizar la carpeta almacenada en la configuración mediante la clave `inputfolder`. La monitorización se realiza mediante un WatchService (busca información sobre el método para hacer esto en Internet) que debe detectar cuando se deposita un nuevo archivo en la carpeta monitorizada.
- Cuando el dispatcher detecta que se ha depositado un nuevo archivo en la carpeta, deberá crear un nuevo hilo hijo o *worker*. dispatcher proporciona a cada worker la ruta completa del archivo que debe procesar y la información adicional que crea necesaria para que el hilo realice la .
- El hilo worker debe:
 - Comprobar si el archivo es un archivo de imagen JPG. Si lo es, se debe abrir el archivo de imagen y generar una imagen nueva procesando la imagen original (más sobre esto más adelante). La imagen se debe guardar en la carpeta de salida con el mismo nombre que la carpeta original. Finalmente el archivo original, sea una imagen válida o no, se debe eliminar de la carpeta de entrada. Una vez terminada su tarea el hilo worker termina y desaparece.
 - La imagen se debe procesar de la siguiente manera. Si el ancho inicial de la imagen es mayor que la anchura máxima o la altura inicial de la imagen es mayor que la altura máxima, la imagen debe reducirse de forma que se cumplan las dos condiciones anteriores. La reducción de la imagen debe realizarse de forma proporcional, esto es, la imagen no se debe deformar.

Ejemplo de ejecución:

- La aplicación inicia y lee la configuración desde el archivo config.properties. Si el archivo no existe se deben usar los valores por defecto indicados anteriormente a fin de que la aplicación siga funcionando. Si el archivo existe lo debe leer y obtener la configuración desde las claves arriba indicadas. Si alguna de las claves no existe se debe usar para el parámetro de configuración el valor por defecto.
- Se inicia la monitorización de la carpeta de entrada y se espera a que se deposite o depositen archivos en la misma.
- Supongamos que se depositan dos archivos: "p1.jpg" y "p2.png". El servicio detecta la aparición de estos archivos y lo notifica a la aplicación. Ésta crea dos hilos de tipo worker. A uno le pasa el archivo "p1.jpg" y a otro el "p2.png". Cada worker, en paralelo, accede al archivo que le corresponde, lo carga, determina si debe cambiar el tamaño de la imagen y, si es necesario lo hace. Una vez procesada la imagen la almacena en la carpeta de salida con el mismo nombre que tenía en la de entrada. Por último elimina el archivo en la carpeta de entrada y termina.

Valoración:

Se valorará:

- Entrega en tiempo
- Adecuación de la aplicación a lo especificado
- Corrección del código