

## Práctica 1: Gestor de descargas de ficheros

El objetivo de esta práctica es desarrollar un gestor de descargas de ficheros concurrente. Para ello, se proporcionará una URL desde la que tendréis que descargar un fichero de texto plano con información de los ficheros a descargar.

Los ficheros estarán separados por líneas que contienen la cadena “Fichero: nombre.ext” donde *nombre* será el nombre del fichero y *ext* su extensión. Las siguientes líneas contienen la dirección de cada una de las partes en las que se ha dividido el fichero. Sabremos que tenemos todas las partes cuando aparezca una nueva línea que comience por “Fichero” o cuando lleguemos al fin de fichero. Cada línea representa la siguiente parte a descargar (se encuentran ordenadas). Es decir, la primera parte vendrá en la línea 1, la segunda en la línea 2, etc.

A partir de esta información, deberemos descargar cada una de las partes en una carpeta “downloads” (si no existe, el programa debe crearla) que estará situada en el mismo directorio desde el que se ejecuta la aplicación. Una vez descargadas todas las partes de un fichero, hay que unirlas para reconstruir el fichero original. Para unir los ficheros, debéis utilizar el método que tenéis disponible en el Código 1. Dicho método recibe como entrada el directorio donde se encuentran las partes descargadas y el nombre del fichero original (*nombre.ext* en el ejemplo anterior). La ejecución de este método generará el fichero original en el directorio indicado. Una vez creado el fichero original, el programa deberá eliminar las partes descargadas.

El gestor de descargas estará implementado en la clase *FileDownloader* y debe permitir realizar descargas simultáneas de varias partes de un mismo fichero. Para ello, el constructor de la clase recibirá como parámetro el número de descargas máximas que se pueden realizar de manera simultánea. Además, los ficheros deben ser descargados en orden, de manera que no comenzaremos la descarga de un nuevo fichero hasta que no haya finalizado la descarga de todas las partes del fichero anterior. La descarga de cada una de las partes se hará utilizando el código de apoyo que aparece en el Código 2.

Siempre deben existir vivos el número de *threads* indicados como número de descargas máximas, por lo que no se permitirá crear diferentes *threads* para cada parte, sino que habrá que reutilizar los originales. Se deja a criterio del estudiante la elección del mecanismo de sincronización que considere oportuna.

El número de ficheros a descargar será variable, así como el número de partes en las que se divide cada fichero, y el tamaño de cada una de las partes. Se deberán cumplir todos los requisitos vistos en clase para asegurar exclusión mutua en aquellos puntos en los que debe haberla.

Ejemplo de fichero de descarga:

<https://github.com/jesussanchezoro/PracticaPC/raw/master/descargas.txt>

```
public void mergeFile(String dir, String fileStart) {
    File ofile = new File(dir+"/"+fileStart);
    FileOutputStream fos;
    FileInputStream fis;
    byte[] fileBytes;
    int bytesRead = 0;
    String[] files = new File(dir).list((path, name) ->
Pattern.matches(fileStart+Pattern.quote(".")+"part.*", name));
    Arrays.sort(files);
    try {
        fos = new FileOutputStream(ofile,false);
        for (String fileName : files) {
            File f = new File(dir+"/"+fileName);
            System.out.println("Merging "+f.getAbsolutePath());
            fis = new FileInputStream(f);
            fileBytes = new byte[(int) f.length()];
            bytesRead = fis.read(fileBytes, 0,(int) f.length());
            assert(bytesRead == fileBytes.length);
            assert(bytesRead == (int) f.length());
            fos.write(fileBytes);
            fos.flush();
            fileBytes = null;
            fis.close();
            fis = null;
        }
        fos.close();
        fos = null;
    }catch (Exception exception){
        exception.printStackTrace();
    }
}
```

**Código 1: Reconstrucción de un fichero dividido en diferentes partes.**

```
URL website = new URL(url);
InputStream in = website.openStream();
Path pathOut = Paths.get(path);
Files.copy(in, pathOut, StandardCopyOption.REPLACE_EXISTING);
in.close();
```

**Código 2: Descarga de un fichero dada una url.**