



## Acceso a datos

---

### Actividad UT 01 – 02 – 03

#### Programación multiproceso y programación multihilo

---

##### CONTENIDO

1.- Objetivos .....	2
2.- Equipos de trabajo .....	2
3.- Requisitos previos .....	2
4.- Visión general de la actividad .....	2
5.- Descripción del proceso .....	2
5.1.- Programa que genera el fichero de transferencias .....	2
5.2.- Programa que procesa el fichero de transferencias .....	3
5.3.- Programa principal.....	4
6.- Requisitos técnicos, entrega y evaluación .....	5
7.- Recomendaciones .....	5
Anexo – Ejemplos de ejecución.....	6
1.- Ejemplo de ejecución en la que el usuario ha indicado un fichero en lugar de un directorio .....	6
2.- Ejemplo de ejecución (parcial) en en la que el usuario se equivoca al indicar un número .....	6
3.- Ejemplo de ejecución en la que falla la creación del fichero .....	6
4.- Ejemplo de ejecución en la que se procesan todas las transferencias, y hay saldo para todas.....	7
5.- Ejemplo de ejecución en la que se procesan todas las transferencias, pero no hay saldo para todas .....	8
6.- Ejemplos de tres ficheros cuando se procesan todas las transferencias, pero no hay saldo para todas.....	9

## 1.- Objetivos

- Practicar la programación multiproceso, en la que varios procesos se coordinan para llevar a cabo un trabajo.
- Practicar la programación multihilo, en la que un programa divide su trabajo en hilos para realizar procesamiento de información de forma más eficiente.

## 2.- Equipos de trabajo

Esta actividad se realizará en equipos, de tres personas cada uno.

## 3.- Requisitos previos

- IntelliJ Idea (recomendado) u otro IDE a elección del alumno.

## 4.- Visión general de la actividad

La actividad está enmarcada dentro de las UT 01 (programación concurrente, paralela y distribuida), UT 02 (programación multiproceso) y UT 03 (programación multihilo).

Se pretende simular el proceso por el que un banco automatiza el pago de nóminas a empleados de una empresa usando un sistema automatizado. Se pretende simular cómo una empresa que quiera pagar la nómina a sus empleados enviará un fichero a su banco, que, cuando lo reciba, lo procesará.

El banco ingresará la nómina en las cuentas de los empleados, sacándolo de la cuenta de la empresa.

Esta práctica está inspirada en un proceso real que se produce habitualmente entre empresas y bancos, para el pago de nóminas, abono de becas, domiciliación de pagos, etc.

Las empresas pueden usar un fichero estandarizado, “cuaderno 34”, por el que le pueden ordenar a su banco en un mismo fichero múltiples (miles de hecho) operaciones.

Para más información sobre este procedimiento se puede consultar:

[https://www.caixabank.es/deployedfiles/empresas/Estaticos/pdf/Transferenciasyficheros/Guia\\_de\\_ayuda\\_migracion\\_cuaderno\\_34-xml\\_Febrero\\_2016.pdf](https://www.caixabank.es/deployedfiles/empresas/Estaticos/pdf/Transferenciasyficheros/Guia_de_ayuda_migracion_cuaderno_34-xml_Febrero_2016.pdf)

[https://docs.bankinter.com/stf/plataformas/empresas/gestion/ficheros/formatos\\_fichero/cuaderno\\_34\\_14\\_transferencias\\_s\\_eпа\\_version\\_en\\_ingles.pdf](https://docs.bankinter.com/stf/plataformas/empresas/gestion/ficheros/formatos_fichero/cuaderno_34_14_transferencias_s_eпа_version_en_ingles.pdf)

## 5.- Descripción del proceso

En la simplificación del proceso real que se ha diseñado para esta actividad, se realizarán tres procesos:

- Generar fichero de transferencias. Simula la generación por parte de la empresa y el envío al banco del fichero de transferencias. Se hará un primer programa para este proceso.
- Procesar el fichero de transferencias. Simula el procesamiento por parte del banco del fichero de transferencias. Se hará un segundo programa para este proceso.
- Los dos procesos anteriores se lanzarán desde un tercer proceso, el principal, de que alguna manera “orquestará” la ejecución, y permitirá parametrizarla y controlarla. Se hará otro programa para este proceso.

### 5.1.- Programa que genera el fichero de transferencias

Realizará las siguientes acciones, en este orden:

- Simulará un posible fallo en la generación del fichero. Para hacerlo:

- Generará un número aleatorio entre 0 y 100.
  - Si el número generado es menor que 30, saldrá sin hacer nada acabando el programa con código de error “100”.
  - Si el número es mayor o igual a 30, continúa su ejecución.
- Recogerá del stream de entrada:
  - El path del directorio de trabajo. Se asume que es correcto, no validar.
  - El nombre del fichero de transferencias. Se asume que es correcto, no validar.
  - El número de transferencias a generar. Se asume que es correcto, convertir sin validar.Estos datos los recibirá del proceso principal a través de un stream.
- Generará un fichero de transferencias que contendrá múltiples líneas, una por transferencia, y en cada línea, separados por un punto y coma:
  - Cuenta del empleado: número aleatorio de 9 cifras que SIEMPRE debe empezar por 1 o por 2. Dicho de otra forma, un número entre 100000000 y 299999999 ambos incluidos.
  - Importe de la nómina: número aleatorio entre 1.500 y 3.000, ambos incluidos, real, que se escribirá en el fichero con 2 decimales.
- Finalizará con un código de éxito si no ha habido problemas. Si se ha producido algún problema (distinto del error simulado), tendrá que terminar con un código de error 200.

Ejemplo de fichero generado:

```
186727813;2849.44
122306769;2494.76
152934832;2119.31
144992274;2139.57
110217146;2673.46
103472055;1511.27
113069851;2240.79
167240275;1937.59
189569038;2264.45
232838402;1604.18
152132016;2251.34
230673370;2338.61
```

### 5.2.- Programa que procesa el fichero de transferencias

Realizará las siguientes acciones, en este orden:

- Recogerá del stream de entrada:
  - El path del directorio de trabajo
  - El nombre del fichero con las transferencias
  - El número de transferencias que hay que procesar (NT)
  - El número de hilos que deben procesar el fichero (NH)Estos datos los recibirá del proceso principal a través de un stream.
- Creará un monitor que represente la cuenta del banco, con un saldo inicial que se calculará como  $NT * (\text{Aleatorio entre } 1500 \text{ y } 2700)$ , redondeado a 2 decimales. La idea es generar un saldo que a veces será suficiente para hacer las NT transferencias y a veces no.
- Leerá el fichero de transferencias y las guardará en otro monitor que contenga alguna clase de estructura de datos (cola, lista, pila).
- Abrirá tres streams (ficheros) para poder clasificar las transferencias procesadas:
  - Transferencias sin saldo. Transferencias que no se han realizado porque la cuenta de la empresa se ha quedado sin saldo. El fichero se llamará igual que el fichero de transferencias original, pero con la extensión “.sinsaldo” añadida.

- Transferencias internas. Son las de las cuentas que comienzan por 1. El fichero se llamará igual que el fichero de transferencias original, pero con la extensión “.internas” añadida.
  - Transferencias externas. Las de las cuentas que comienzan por 2. El fichero se llamará igual que el fichero de transferencias original, pero con la extensión “.externas” añadida.
- Creará NH hilos para procesar las transferencias. Estos hilos deben recibir:
  - Un nombre, para poder identificarlos.
  - El monitor que permite acceder a la cola de transferencias compartida.
  - El monitor que permite acceder a la cuenta de la empresa, para transferir el importe de esta a las cuentas de los empleados.
  - Los tres streams abiertos para guardar las operaciones realizadas (internas / externas) y las no realizadas por saldo insuficiente.
- Esperará a que los hilos terminen.
- Mostrará el importe procesado por cada hilo, y la suma de los importes procesados.
- Mostrará el saldo restante en la cuenta de la empresa.
- Terminará.

### Hilos que procesan las transferencias

Los hilos que procesan las transfrencias se comportan de la siguiente forma:

- Se ejecutan mientras haya transferencias que procesar, y mientras que no sean interrumpidos.
- En cada ciclo de su método run procesan una transferencia, lo que implica:
  - Obtener la transferencia de la cola.
  - Mostrar por la salida estándar qué operación se va a intentar.
  - Intentar obtener el saldo de la cuenta de la empresa.
  - Mostrar un mensaje indicando si se ha podido o no obtener el saldo.
  - Mostrar un mensaje indicando donde se guarda la operación (en qué fichero de los tres)
  - Acumulará el importe de la transferencia para llevar la cuenta de cuanto ha procesado. Ojo, aunque la transferencia vaya al fichero de transferencias sin fondos, se acumula su valor, porque la ha procesado.
- Los hilos se paran cuando se hayan procesado todas las transferencias, aunque no todas se hayan podido procesar, porque se agotó el saldo, o cuando son interrumpidos.

### 5.3.- Programa principal

Realizará las siguientes acciones, en este orden:

- Preguntará al usuario el nombre del directorio de trabajo, donde estarán los ficheros, y:
  - Si existe, pero no es un directorio, mostrará un mensaje de aviso y terminará.
  - Si no existe, lo creará, vacío. Si al crearlo falla, mostrará un mensaje de aviso y terminará.
- Preguntará al usuario el nombre del fichero con las transferencias, y:
  - Si ya existe, lo eliminará, para que no exista previamente.
  - Si al eliminarlo falla, mostrará un mensaje de aviso y terminará.
- Preguntará al usuario el número de transferencias a generar, y:
  - Si introduce un valor que no sea un número entero, o que no sea mayor que cero, mostrará un mensaje de aviso y volverá a preguntar.
  - Seguirá preguntando hasta que el usuario introduzca un número válido.
- Preguntará al usuario el número de hilos que se deben usar para procesar el fichero, y:
  - Si introduce un valor que no sea un número entero, o que no sea mayor que cero, mostrará un mensaje de aviso y volverá a preguntar.
  - Seguirá preguntando hasta que el usuario introduzca un número válido.

- Arrancará el programa que genera el fichero de transferencias. Pasará al programa la información necesaria (directorio de trabajo, nombre de fichero, número de transferencias) a través del stream del proceso iniciado.
- Esperará a que este proceso termine.
- Si el programa que genera el fichero falla, avisará al usuario, indicando claramente el código de salida del proceso, y finalizará.
- Arrancará el programa que lee y procesa el fichero de transferencias. Pasará al programa la información necesaria (directorio de trabajo, nombre de fichero, número de transferencias, número de hilos que procesarán el fichero) a través del stream del proceso iniciado.
- Esperará a que este proceso termine, mostrando por consola los mensajes del proceso lanzado, que capturará a través de un stream del proceso, para que el usuario pueda constatar el avance del proceso.

## 6.- Requisitos técnicos, entrega y evaluación

En este proyecto no se pueden usar bibliotecas de clases externas como, por ejemplo, Lombok. Para facilitar la corrección deben usarse solo clases que estén dentro de la API de Java.

Todas las clases se desarrollarán en el mismo paquete, que debe ser “es.dm2e.psp.actividad01.grupoX”, donde la X será un número que identifique al grupo. El profesor indicará qué número tiene cada grupo.

El programa principal estará obligatoriamente en una clase llamada “Main”, para facilitar al profesor localizar el punto de arranque. Esta clase Main debe estar preparada para que, al ejecutar el programa, invoque los otros dos procesos sin problemas, sin tener que hacer ajustes de rutas u otros cambios en el código fuente.

Se entregará sólo la carpeta “src” del proyecto comprimida en formato “zip”. Se penalizará la entrega de proyectos completos o que no cumplan con esta directriz.

Para poder registrar la calificación, cada uno de los integrantes de un equipo realizarán una entrega. La calificación será la misma para todos los integrantes del grupo, salvo casos evidentes en los que alguno de los integrantes no haya colaborado lo suficiente, en cuyo caso el profesor podrá poner una menor calificación a ese miembro del equipo.

## 7.- Recomendaciones

Algunas recomendaciones sobre el desarrollo de la actividad:

- En el programa principal (clase Main), al arrancar los procesos hijos usando ProcessBuilder, hacerlo con esta configuración:

```
ProcessBuilder pb = new ProcessBuilder("java", "clase-que-hay-que-arrancar");
pb.environment().put("JAVA_HOME", System.getProperty("java.home"));
pb.environment().put("CLASSPATH", System.getProperty("java.class.path"));
.
.
.
```

Esto hace que se arranquen los procesos con la misma versión de Java que Main, y que se puedan localizar las clases sin problemas.

- En el programa que genera el fichero, no olvidar devolver un código de error distinto de cero (100 en un caso concreto) si el programa falla.
- Al procesar el fichero de transferencias:

- Usar un monitor para gestionar el saldo de la cuenta de la empresa, para que cuando un hilo está “sacando” dinero de esa cuenta para transferirlo, no pueda otro hilo hacerlo a la vez.
- Usar un monitor para procesar la cola de transferencias, para que no puedan dos hilos procesar la misma transferencia.
- Sincronizar las escrituras en los ficheros de salida (externas, internas, sin saldo), para evitar que se realicen escrituras simultáneas.
- Revisar los ejemplos de ejecución de la documentación para tener información de los mensajes que se esperan, y analizar el comportamiento esperado.
- En general, para las operaciones de ficheros, apoyarse en las clases Path y Files, que tienen métodos para manejar los ficheros y directorios cómodamente.

### Anexo – Ejemplos de ejecución

#### 1.- Ejemplo de ejecución en la que el usuario ha indicado un fichero en lugar de un directorio

Introduce el path del directorio de trabajo:  
prueba/transferencias.txt  
prueba/transferencias.txt no es un directorio.

#### 2.- Ejemplo de ejecución (parcial) en la que el usuario se equivoca al indicar un número

```
...
transferencias.txt
Introduce el número de transferencias que se quieren simular:
diez
Debe introducir un número entero mayor que cero.
-10
Debe introducir un número entero mayor que cero.
10
Introduce el número de hilos que se deben usar:
4
Main - Iniciando programa de generación de transferencias...
...
```

#### 3.- Ejemplo de ejecución en la que falla la creación del fichero

```
Introduce el path del directorio de trabajo:
prueba
Introduce el nombre del fichero de transferencias:
transferencias.txt
Introduce el número de transferencias que se quieren simular:
10
Introduce el número de hilos que se deben usar:
3
Main - Iniciando programa de generación de transferencias...
Main - Pasando parámetros al programa de generación de transferencias...
Main - Esperando a que el programa de generación de transferencias termine...
Main - Se ha producido un error al generar el fichero. Código de salida: 100.
Main - Terminando programa principal por error en generación de fichero...
```

El directorio “prueba”, si no existía antes, debería estar vacío.

#### 4.- Ejemplo de ejecución en la que se procesan todas las transferencias, y hay saldo para todas

```
Introduce el path del directorio de trabajo:
prueba
Introduce el nombre del fichero de transferencias:
transferencias.txt
Introduce el número de transferencias que se quieren simular:
10
Introduce el número de hilos que se deben usar:
3
Main - Iniciando programa de generación de transferencias...
Main - Pasando parámetros al programa de generación de transferencias...
Main - Esperando a que el programa de generación de transferencias termine...
Main - Iniciando programa de procesamiento de transferencias...
Main - Pasando parámetros al programa de procesamiento de transferencias...
Main - Mostrando mensajes generados por el programa que procesa el fichero...
Main - Procesador dice: Iniciando procesamiento de transferencias
Main - Procesador dice: Leyendo el fichero de transferencias...
Main - Procesador dice: Fichero de transferencias leído.
Main - Procesador dice: Suma de importes de TODAS las transferencias que hay que realizar: 23556.45€.
Main - Procesador dice: Saldo inicial de la cuenta de la empresa: 25112.53€.
Main - Procesador dice: Preparando transferencia de 1641.84€ a la cuenta 296617353
Main - Procesador dice: Retirados 1641.84€ de la cuenta de la empresa.
.
.
.
Main - Procesador dice: Preparando transferencia de 2359.78€ a la cuenta 147305222
Main - Procesador dice: Retirados 2359.78€ de la cuenta de la empresa.
Main - Procesador dice: Grabando transferencia interna a la cuenta 147305222.
Main - Procesador dice: El hilo Hilo 1 ha procesado 8655.61€.
Main - Procesador dice: El hilo Hilo 2 ha procesado 7463.02€.
Main - Procesador dice: El hilo Hilo 3 ha procesado 7437.82€.
Main - Procesador dice: Total procesado por los hilos: 23556.45€
Main - Procesador dice: Saldo final de la cuenta de la empresa: 1556.08€.
Main - Esperando a que el procesador de transferencias termine...
Main - Terminando programa principal correctamente.
```

En el directorio “prueba” debería haber cuatro ficheros:

```
transferencias.txt
transferencias.txt.external
transferencias.txt.internal
transferencias.txt.notEnoughBalance
```

El fichero de transferencias sin saldo debería estar vacío, y las transferencias del fichero generado se deben repartir entre los dos ficheros (internas y externas)

El importe de todas las transferencias debe ser igual que el total procesado.

La suma de las cantidades debe cuadrar: Saldo final = Saldo inicial – Total procesado

En el ejemplo:  $1556,08 = 25112,53 - 23556,45$

*5.- Ejemplo de ejecución en la que se procesan todas las transferencias, pero no hay saldo para todas*

```
Introduce el path del directorio de trabajo:
prueba
Introduce el nombre del fichero de transferencias:
transferencias.txt
Introduce el número de transferencias que se quieren simular:
10
Introduce el número de hilos que se deben usar:
3
Main - Iniciando programa de generación de transferencias...
Main - Pasando parámetros al programa de generación de transferencias...
Main - Esperando a que el programa de generación de transferencias termine...
Main - Iniciando programa de procesamiento de transferencias...
Main - Pasando parámetros al programa de procesamiento de transferencias...
Main - Mostrando mensajes generados por el programa que procesa el fichero...
Main - Procesador dice: Iniciando procesamiento de transferencias
Main - Procesador dice: Leyendo el fichero de transferencias...
Main - Procesador dice: Fichero de transferencias leído.
Main - Procesador dice: Suma de importes de TODAS las transferencias que hay que realizar: 21623.96€.
Main - Procesador dice: Saldo inicial de la cuenta de la empresa: 15517.89€.
Main - Procesador dice: Preparando transferencia de 1532.18€ a la cuenta 157145382
Main - Procesador dice: Retirados 1532.18€ de la cuenta de la empresa.
.
.
.

Main - Procesador dice: No se han podido retirar 2654.01€ de la cuenta de la empresa.
Main - Procesador dice: Grabando transferencia sin fondos a la cuenta 136531345.
Main - Procesador dice: El hilo Hilo 1 ha procesado 6858.75€.
Main - Procesador dice: El hilo Hilo 2 ha procesado 8484.44€.
Main - Procesador dice: El hilo Hilo 3 ha procesado 6280.77€.
Main - Procesador dice: Total procesado por los hilos: 21623.96€
Main - Procesador dice: Saldo final de la cuenta de la empresa: 588.37€.
Main - Esperando a que el procesador de transferencias termine...
Main - Terminando programa principal correctamente.
```

En el directorio “prueba” debería haber los cuatro mismos ficheros, pero las transferencias del fichero generado se deben repartir entre los tres ficheros (internas, externas, sin saldo).

El importe de todas las transferencias debe ser igual que el total procesado.

La suma de las cantidades debe cuadrar, pero en este caso hay que tener en cuenta las que no se han completado porque no había saldo: Saldo final = Saldo inicial – (total procesado – suma de sin saldo)

En ejemplo estas son las procesadas sin saldo suman 6694,44:

```
269476823;1915.27
201254368;2125.16
136531345;2654.01
```

Y comprobando que cuadren:  $588,37 = 15517,89 - (21623,96 - 6694,44)$ .



*6.- Ejemplos de tres ficheros cuando se procesan todas las transferencias, pero no hay saldo para todas*

Fichero original	Internas	Externas	Sin saldo
145308125;2879.78	145308125;2879.78	218496640;2459.60	178543420;2062.45
218496640;2459.60	138039414;2057.40	214508651;2903.23	
138039414;2057.40	159980332;2954.53	234736525;2203.94	
159980332;2954.53	154522039;1727.64	228850464;1762.25	
154522039;1727.64		232885723;2804.71	
214508651;2903.23			
234736525;2203.94			
228850464;1762.25			
232885723;2804.71			
178543420;2062.45			