

Video Club

Tarea 4

Objetivos:

- Experimentar con el acceso a la base de datos desde un programa Java.
- Experimentar en el uso de transacciones.

Parte 1 – Generación de mails de novedades

Se desea obtener un programa Java que genere un archivo de texto con los mails personalizados a enviar a los socios notificando de los ingresos de películas en su sucursal. Para esto se deben considerar los datos almacenados en la tabla **boletines** creada en la tarea 3.

El archivo a generar debe tener el siguiente formato:

```
#INICIO MAIL#
<mail del cliente>
Asunto: Novedades en tu sucursal del video club
Fecha: <fecha mail>

Hola <nombre cliente>!
Llegaron nuevas peliculas a nuestra sucursal.
Creemos que las siguientes peliculas te pueden interesar:
<Título de película 1>, las categorías de esta película son: <lista-categorías> y los actores que
participan en la misma son:<lista de actores>.
<Título de película 2>, las categorías de esta película son: <lista-categorías> y los actores que
participan en la misma son:<lista de actores>.
.....
.....
.....
<Título de película n>, las categorías de esta película son: <lista-categorías> y los actores que
participan en la misma son:<lista de actores>.

Te esperamos pronto!
Saludos <nombre del encargado de la sucursal>

(Nota: Los ingresos corresponden al periodo <fecha inicio> a <fecha fin>)
#FIN MAIL#

#INICIO MAIL#
<mail del cliente>
.....
.....
#FIN MAIL#
```

Los textos son fijos a excepción de los marcados entre <.....> cuyo contenido se describe a continuación:

<mail del cliente>	Contiene el valor del atributo mail de la tabla boletines
<fecha mail>	Contiene la fecha en que se genera el archivo. Formato: YYYY-MM-DD
<nombre cliente>	Contiene el valor del atributo nombrecliente de la tabla boletines
<Título de película> ,	Contiene el valor del atributo titulo de la tabla boletines
<lista-categorías>	Contiene el valor del atributo categorias de la tabla boletines
<lista de actores>.	Contiene el valor del atributo actores de la tabla boletines
<nombre del encargado de la sucursal>	Contiene el nombre del encargado de la sucursal que envía la notificación. El valor se debe presentar igual que para el nombre del cliente.
<fecha inicio>	Contiene el valor de la fecha mínima de ingreso de las películas a informar por cada cliente. El formato de la fecha es YYYY-MM-DD
<fecha fin>	Contiene el valor de la fecha máxima de ingreso de las películas a informar por cada cliente. El formato de la fecha es YYYY-MM-DD

Observaciones:

- Los mails deben quedar ordenados por el identificador de cliente.
- Las películas listadas en cada mail deben estar ordenadas por nombre de película y si fuera necesario por identificador de película.
- Asumir que existe la tabla boletines, que no es vacía y que es correcta.
- Una vez finalizada la generación del archivo de los mails se deben borrar todos los datos de la tabla de boletines.
- El archivo a generar se debe llamar bdatosNN-mails.txt siendo NN el número de grupo en dos dígitos.

Se debe entregar:

Un único archivo Tarea4Parte1.java que contendrá la clase con el programa principal que resuelve esta parte.

El programa debe poder ser invocado con los siguientes argumentos como entrada exactamente en el orden que se especifica a continuación:

```
java -classpath '.:postgresql-jdbc.jar' Tarea4Parte1 <dbhost> <dbport> <dbname> <username> <password>
```

Aquí se detallan los parámetros recibidos por el programa:

- *dbhost* es un string que indica el host en donde se encuentra el manejador.
- *dbport* es un número entre 1 y 65535 que indica el puerto en donde atiende el manejador.
- *dbname* es el nombre de la base de datos a utilizar.
- *username* y *password* corresponden al nombre de usuario y password para autenticarse contra el manejador

La opción -classpath es utilizada para incluir el jar que contiene las clases necesarias para comunicarse con el manejador.

Parte 2 – Actualización de movimientos de las sucursales

En esta parte del laboratorio, el concepto fundamental a evaluar es el de *transacción*. A los ojos de un usuario común, una transacción es una secuencia de operaciones sobre la base de datos que se ejecuta de manera atómica (es decir, que se ejecutan todas las operaciones o no se ejecuta ninguna) y aislada del resto del universo (el resultado no se ve influido por otras transacciones).

Se desea generar un pequeño programa Java que ingrese los movimientos (alquileres y/o pagos) realizados en las diferentes sucursales en el sistema general. El programa recibirá un conjunto de parámetros de entrada (esto se especifica más adelante), donde uno de ellos es la ruta absoluta al archivo que contiene los datos asociados a cada movimiento de cada sucursal. El programa deberá actualizar las tablas correspondientes con los datos especificados. Desde el punto de vista transaccional, si ocurre cualquier tipo de error durante el procesamiento de una sucursal, la base de datos debe permanecer en el mismo estado en el que se encontraba antes de comenzar el proceso de esa sucursal.

Los movimientos que se manejan son:

- Ingreso en la tabla de alquileres
- Ingreso en la tabla pagos.
- Actualización de la fecha de devolución de un alquiler existente.
- Actualización del monto pago de un pago ya existente.

El único control que debe realizarse en el programa es para el caso de la actualización de la fecha de devolución en la tabla alquileres, en el que debe controlarse que dicha fecha sea posterior a la fecha del alquiler. Cualquier otro tipo de control queda liberado a los controles definidos en la base de datos.

Estructura del Archivo de Datos de entrada

El archivo de datos consiste en una secuencia de bloques de sucursales, en cada uno de los cuales se indican las movimientos de cada sucursal que se deben realizar en el sistema.

A continuación se especifica el formato del archivo que contendrá las operaciones a registrar en el sistema.

BloqueSucursal
BloqueSucursal

BloqueSucursal

Donde cada Bloque de sucursal tiene el siguiente formato:

```
#inicio sucursal: <identificador de sucursal>#  
<código de movimiento> <parámetros correspondientes al movimiento>  
.....  
.....  
<código de movimiento> <parámetros correspondientes al movimiento>  
#fin sucursal#
```

Observaciones:

- Entre el código de movimiento y los parámetros del mismo existe un espacio en blanco.
- Los parámetros son una lista de valores separados por comas.
La descripción de los códigos de movimientos y sus respectivos parámetros se detallan en el siguiente cuadro:

Códigos de movimiento	AA	Actualización de la fecha de devolución en la tabla alquileres
	IA	Ingreso de un nuevo alquiler en la tabla alquileres
	AP	Actualización del monto en la tabla pagos
	IP	Ingreso de un nuevo pago en la tabla pagos
Parámetros para cada tipo de movimiento	AA	idpelicula,idcliente,fecha,idpersonal,fechadevolucion
	IA	idpelicula,idcliente,fecha,idpersonal[,fechadevolucion]
	AP	idpeliculaalquilo,idclientealquilo,idSucursalalquilo,idpersonalalquilo,fechaalquilo,fecha,monto
	IP	idpeliculaalquilo,idclientealquilo,idSucursalalquilo,idpersonalalquilo,fechaalquilo,idpersonalalquilo,fecha,monto

Para el Ingreso de alquileres, la fecha de devolución es un parámetro opcional. En caso de no existir se toma por valor null.

- Para simplificar el *parsing* se puede asumir que:
 - no hay espacios en blanco de ningún tipo entre comas y atributos.
 - el carácter de separación de atributos (,) no aparece en los valores de ningún atributo.
- Asumir que el archivo de datos existe y no es vacío.

Registro de transacciones

Además de registrar los movimientos de las sucursales que correspondan en la base de datos, el programa deberá generar un archivo denominado `bdatosNN-resultados.txt` que contendrá un 1 o un 0 en la línea n indicando si el procesamiento correspondiente a la sucursal ubicada en la posición n del archivo de datos fue exitoso o fallido respectivamente.

Se debe entregar:

Un único archivo `Tarea4Parte2.java` que contendrá la clase con el programa principal que resuelve esta parte.

El programa debe poder ser invocado con los siguientes argumentos como entrada exactamente en el orden que se especifica a continuación:

```
java -classpath '.:postgresql-jdbc.jar' Tarea4Parte2 <inputDataPath> <dbhost> <dbport>
<dbname> <username> <password>
```

Aquí se detallan los parámetros recibidos por el programa:

- `inputDataPath` es la ruta absoluta donde se encuentra el archivo de datos
- `dbhost`, `dbport`, `dbname`, `username` y `password` como en la parte 1.

Consideraciones generales:

- Utilizar el driver JDBC para PostgreSQL que se encuentra en la página del laboratorio.
- Se debe entregar un único archivo .java para cada parte.
- No hacer uso de paquetes java.
- Respetar el orden dado de los parámetros para cada uno de los programas pedidos.
- Respetar el nombre de los archivos de salida a generar.
- Los programas deben funcionar tal cual como fueron descritos, en los equipos de FING.