



Descarga de Listas de Música Proyecto Examen Final

Temas: Sockets, Hilos, Base de Datos y Servlets/WebServices, Principios de Diseño de paquetes y de clases

- El aplicativo debe cumplir con los principios SOLID, además de estar organizado en tras paquetes (Modelo – Vista - Controlador).
- Se puede hacer uso de cualquier motor de BD
- Recuerde, a diferencia del segundo corte, donde el servidor desplegaba la interface en un Frame, para este caso, el servidor debe desplegar la interface de lectura en una página web, para ello puede usar Servlets, web Services, Applets, html.

Descripción:

El proyecto debe contar de

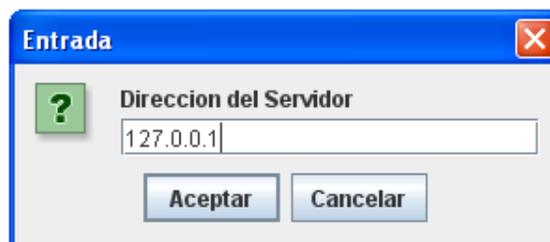
- Un servidor que escucha peticiones de clientes por un Socket;
- Varios clientes que pueden conectarse vía sockets ha dicho servidor.
- Servicios web de consulta y de pago de saldo de descargas de canciones

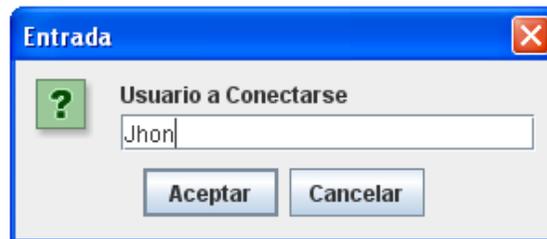
El proceso de conexión y transmisión de las canciones se realiza de la siguiente manera:

- Se inicializa el servidor de canciones. Al arrancar, queda escuchando peticiones de los clientes que quieren descargar canciones (estos clientes deben estar registrados previamente en una base de datos). Para ello haga uso de un ServerSocket.

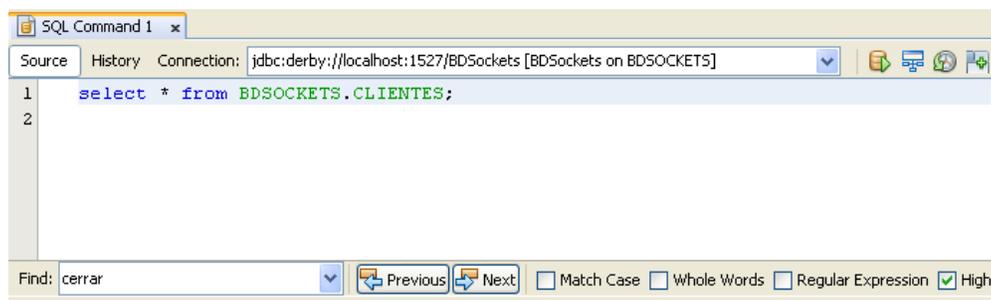
```
Output x
Apache Tomcat Log x Apache Tomcat x Java DB Database Process x Examen (run) x
run:
Servidor >> Direccion IP: JHON/192.168.0.3
IServidor >> El servidor esta recibiendo Conexiones.....
```

- Un cliente se conecta al servidor, vía socket. Tan pronto se crea su Socket para comunicarse con el servidor, le envía su usuario y contraseña al servidor.

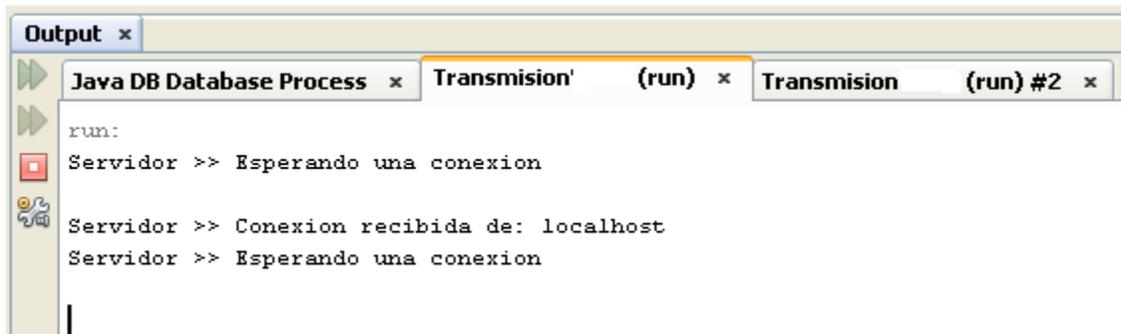




3. El servidor lee del canal el usuario y la contraseña enviados por el Cliente, solicita una instancia de la Conexión a la BD, y consulta dicho usuario en la Tabla de Clientes (usando usuario y contraseña).



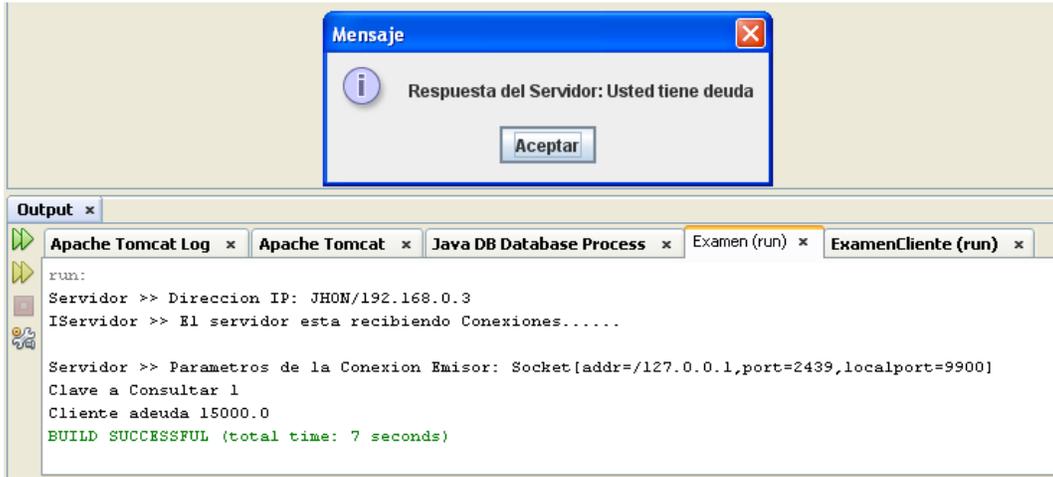
4. Si el Cliente está registrado en la Tabla de Clientes, el servidor crea un hilo de Servidor para atender el Cliente, y le envía una aprobación de servicio al cliente.
5. Al recibir la aprobación de servicio (en caso de que el cliente no tenga deuda), el cliente abre una interfaz (una página web), donde muestra el listado de canciones enviado desde el servidor, las cuales son posibles de descargar. Este listado debe estar almacenado en el servidor, en una base de datos de canciones disponibles. Este listado se debe desplegar en una página html.



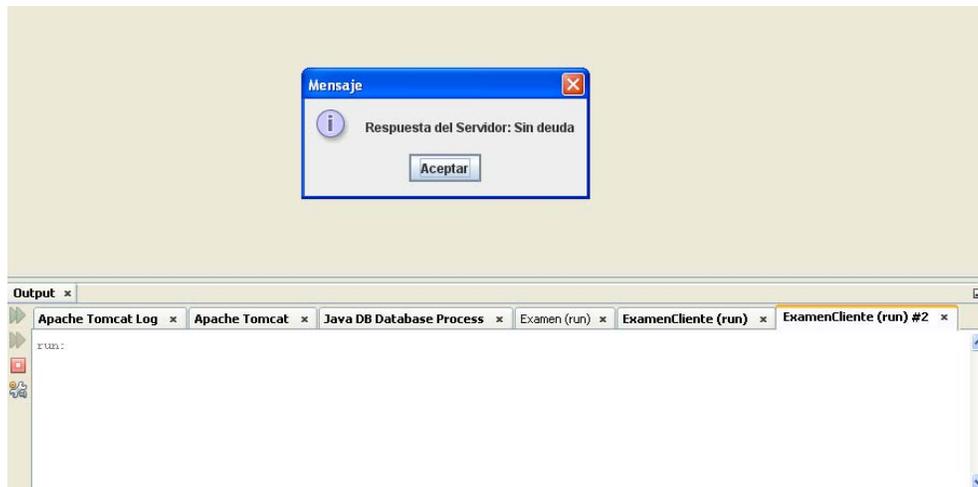
6. El cliente escoge una canción a descargar, solicitud que se envía al servidor, a través del socket. El nombre de la canción seleccionada se envía desde el cliente al servidor, el cual verifica que este cliente no tenga pagos pendientes por descargas anteriores. Si no hay pendientes, el servidor lleva a cabo la transferencia de la canción solicitada hacia el cliente, de lo contrario le manda un mensaje indicándole que tiene deuda y cierra la conexión del socket.



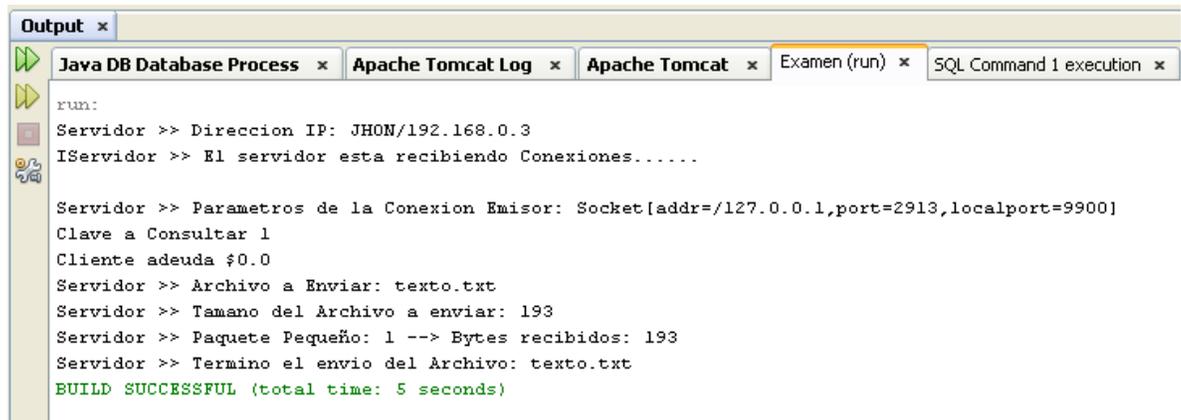
Si tiene deuda:



Si no tiene deuda:



Se hace la transferencia del archivo:





```
Output x
Java DB Database Process x Apache Tomcat Log x Apache Tomcat x Examen (run) x SQL Command 1 execution x ExamenCliente (run) x
run:
Cliente >> Tamaño del archivo a recibir: 193
ClienteRecibir >> Empieza la recepcion del archivo
ClienteRecibir >> Paquete pequeño: 1 --> Bytes recibidos: 193
ClienteRecibir >> Termino la Recepcion del Archivo
BUILD SUCCESSFUL (total time: 0 seconds)
```

7. Para el manejo de la deuda, el cliente se puede conectar a un servicio web, el cual tiene dos operaciones: consultar deuda y pagar deuda:



Al consultar deuda, se debe ingresar el código del cliente:

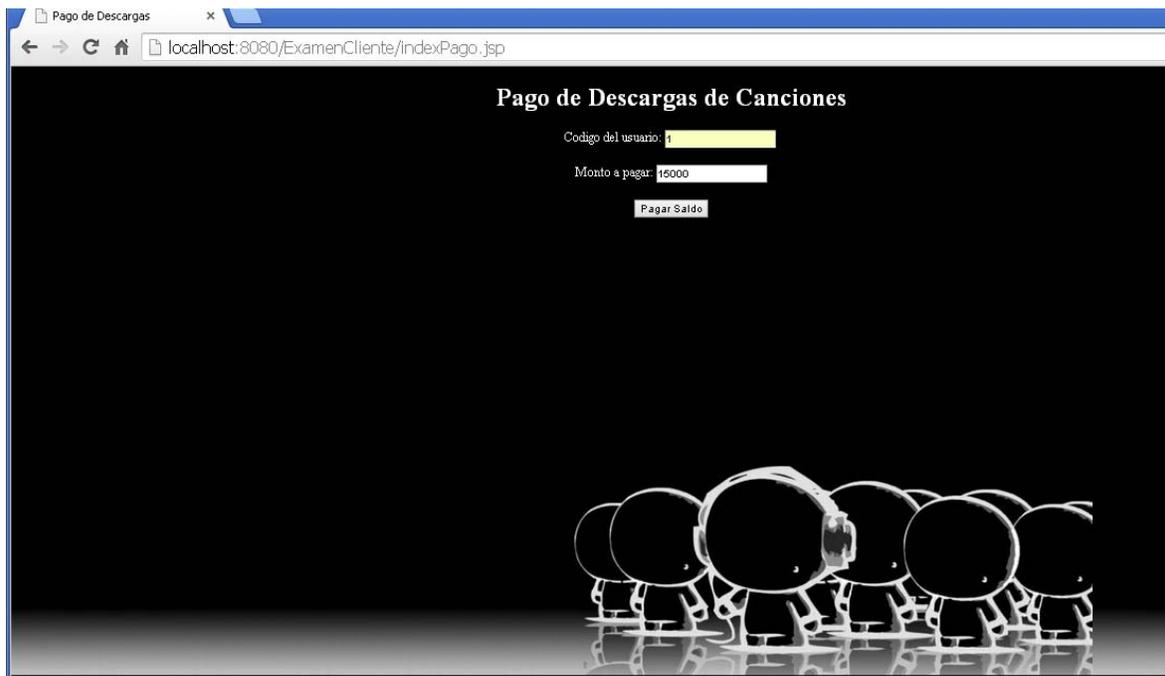
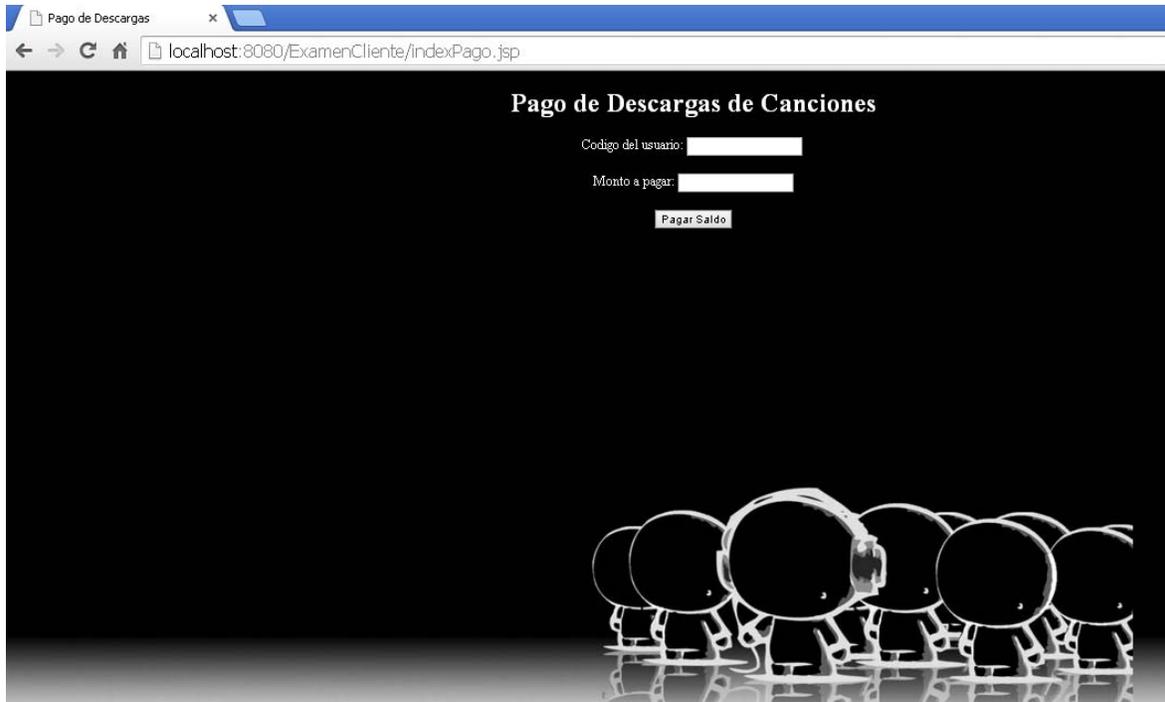


Al enviar la consulta del web service, la operación se conecta al DAO y este a la Base de Datos, dando como respuesta la deuda del cliente:

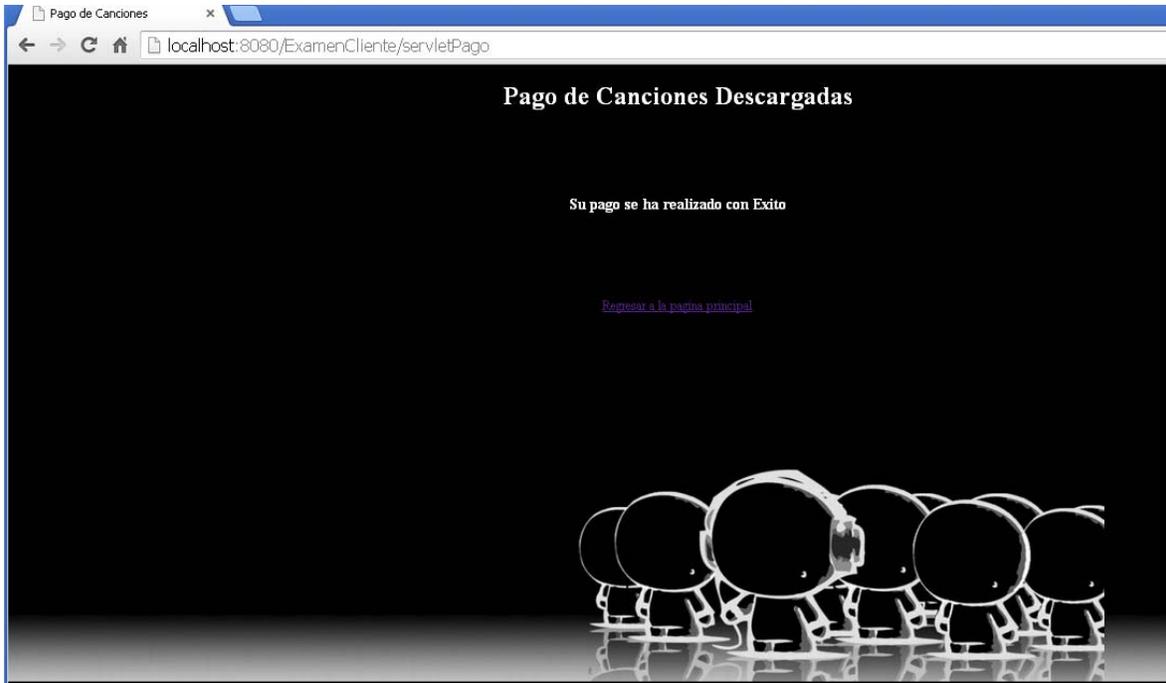


Esta página tiene un enlace para retornar a la página principal.

En caso de querer cancelar la deuda pendiente para poder seguir descargando canciones, se debe desplegar una operación que cargue el monto a pagar. El usuario puede pagar menos del saldo o el saldo completo, el cual será descontado de la base de datos. Si el usuario paga menos del saldo, no podrá descargar canciones.



Al realizarse el pago parcial del saldo, se mostrara una página informando cuanto es el excedente pendiente. En caso de pagar completo el saldo, mostrara la siguiente página:



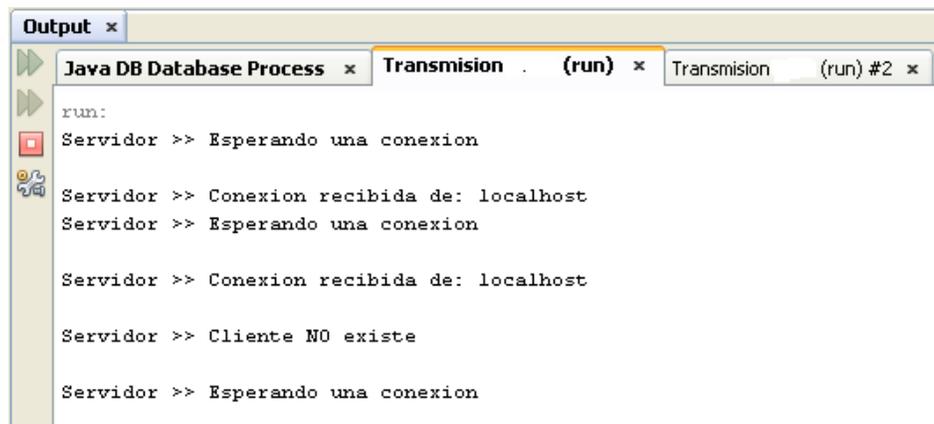
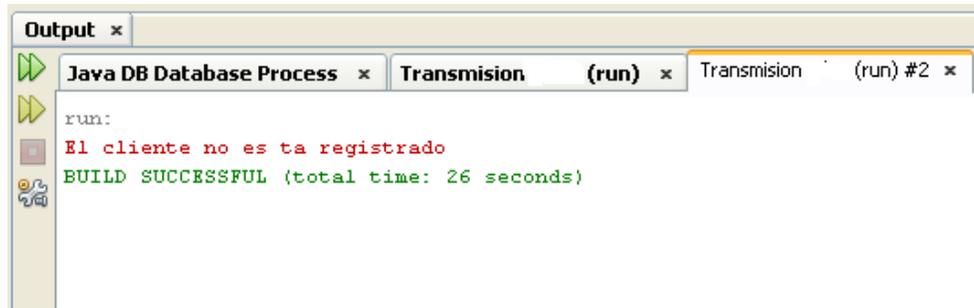
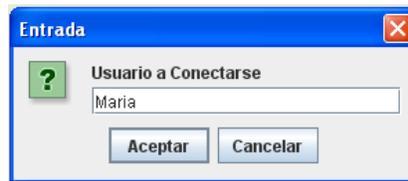
8. El cliente puede descargar varias canciones, una tras otra, las cuales cada una tiene un costo de \$15.000, costo que el servidor debe ir registrando. Esta transmisión se termina cuando el cliente se salga de su sesión (envíe un logout). Inmediatamente también se cierra la interfaz abierta en el Servidor, y se desactiva el cliente. El servidor debe tener registrado el valor a pagar por las descargas realizadas en la sesión actual.
9. El servidor continúa escuchando peticiones de transmisión de canciones de otros clientes.

```
Output x
Java DB Database Process x Transmision' (run) x Transmision (run) #2 x
run:
Servidor >> Esperando una conexion
Servidor >> Conexion recibida de: localhost
Servidor >> Esperando una conexion
```

10. De forma paralela, cuando un cliente está conectado y descargando canciones del servidor, otro cliente puede conectarse, validar su usuario y descargar canciones del servidor. En esta situación, el servidor abrirá un hilo de atención por cada cliente que se conecte. Tenga en cuenta que no se van a desarrollar interfaces para registrar canciones ni para cancelar canciones. Estos valores se deben cargar y eliminar manualmente de la base de datos. Pero de igual forma, las canciones borradas o adicionadas deben ser refrescadas en las vistas de los clientes.



Si un Cliente se intenta conectar con un Usuario que no esté registrado en la BD, este será rechazado, y el servidor continuara recibiendo peticiones.



Por último, tenga en cuenta que dependiendo del formato de audio de la canción y de la forma de transmisión, debe tener en cuenta que algunas veces requeriría configurar los códec de audio y las salidas de audio correspondientes. Hay sonidos que se reproducen por el altavoz del sistema, pero hay otros que requieren de una configuración especial, ya sea en el IDE o en la máquina para ser reproducidos. Revise y estudie la librería Java Media Framework (JMF).

La interfaz en el cliente y en el servidor debe manejar una apariencia adecuada, con un look and feel accesible y usable. En el ejemplo presentado el servidor está mostrando su actividad en consola, pero para el desarrollo del parcial, debe mostrarla en entorno gráfico. Por otro lado, los clientes deben tener entorno gráfico, donde muestren las canciones, la solicitud de descarga, el proceso de descarga y la finalización de la misma.

Por último, implemente un servicio que se consuma en línea, referente al tema de la música y/o las canciones.



Algunas referencias:

http://es.wikipedia.org/wiki/Java_Media_Framework

<http://www.igutierrezgil.es/proyectofinal/>

<http://www.dlsi.ua.es/asignaturas/sid/JSocket.pdf>

Para el desarrollo y sustentación de este aplicativo, tenga en cuenta que:

- a. El aplicativo debe estar distribuido en tres capas (Modelo – Control - Vista)
- b. Debe cumplir con todos los principios de diseño orientado a objetos
- c. Debe haberse subido al aula, en la fecha prevista. Entrega que no se haya subido al aula no se recibirá por ningún medio, y su nota será 0.0
- d. El aplicativo se debe desarrollar en grupos de a tres persona
- e. Se debe adjuntar los diagramas de clase y los diagramas de secuencia del aplicativo, exportados como imágenes JPG
- f. Debe implementarse el manejo de Bases de Datos (cualquier motor), Threads, Sockets y servicios web.
- g. No debe haber ninguna validación, ni regla de negocio, etc., en las clases del mundo ni en las clases de la vista.
- h. La interfaz se puede desarrollar por asistente. Tenga cuidado que se deben modificar algunos componentes internos, situación que complica un poco el asistente.
- i. El entorno grafico debe tener una adecuada separación de paneles, marcos, eventos, etc.
- j. Si el proyecto funciona pero las personas en la sustentación no lo saben manejar, no reconocen su estructura, no saben donde están los elementos, etc., su entrega será anulada.
- k. Evítese cualquier inconveniente con cualquier intento de plagio entre compañeros.
- l. Puede cambiar el look and feel del aplicativo. En la red pueden encontrar documentación acerca de las clases a utilizar, puede hacer uso de ella, pero debe tener claro cómo funciona.
- m. Para cualquier información de requerimientos de entorno grafico, puede consultar el API de Java.