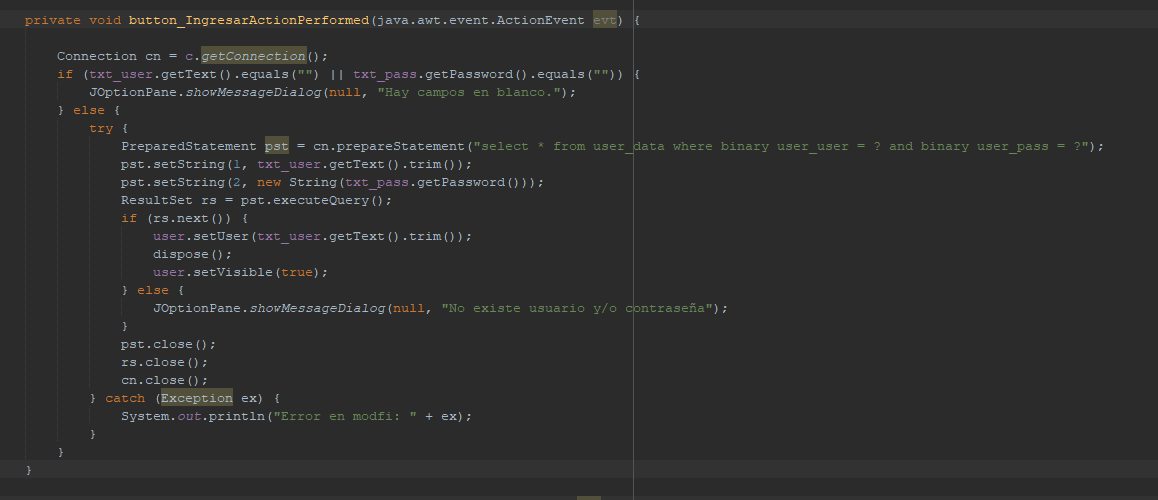
**Banco BVVA**

1. [Login](#Login)
   1. [Ingreso](#Ingreso)
   2. [Olvido de contraseña](#Olvidec)
   3. [Registro](#Registro)
2. [Panel](#Panel)
   1. [Ayuda](#ayuda)
   2. [Usuarios](#usuarios)
   3. [Ingreso](#IngresoMaster)
3. [Modificación Usuario](#ModifUsuario)
4. [Plata](#Plata)
5. [Tarjetas](#Tarjetas)
6. [Créditos](#Creditos)
7. [Seguros](#Seguros)
8. [Inversiones](#Inversiones)
9. [Mall](#mall)
10. [Base de Datos](#BaseDeDatos)

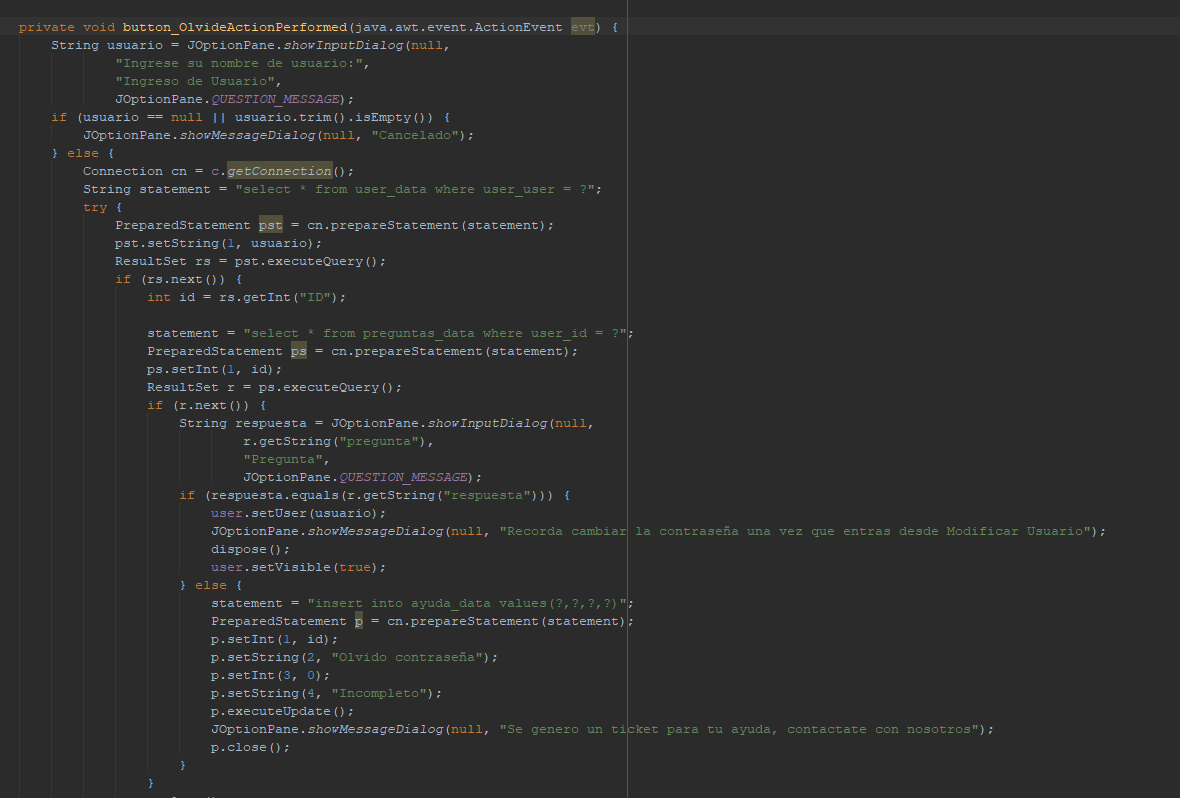
Login

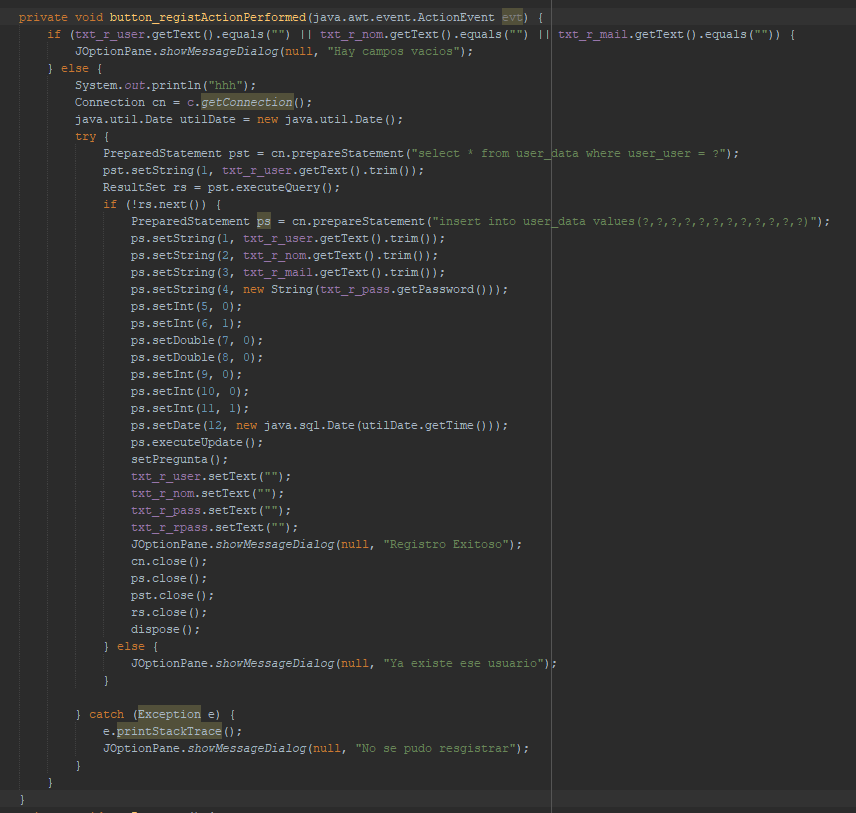
Login hace lo que dice, ingresar al banco.

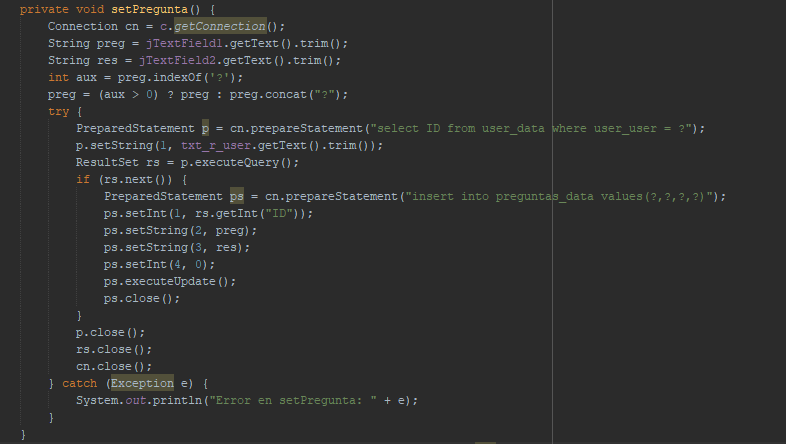
Ingresar: Este botón verifica de manera segura (binary) que sea exactamente el usuario y la contra, diferenciando de minúsculas y mayúsculas. En caso de error, se informa al usuario. Si accede se pasa a Panel.



Olvide Contraseña: Este botón te pide el usuario y en base a eso, accede a la Base de Datos, y le hace la pregunta guardada, en caso que no lo responda bien, se generará un ticket para que lo ayuden con el problema. Por como está la Base de Datos hecha, tiene que pedir el id así puede buscarlo en la tabla de preguntas\_data. Genera la pregunta y si no equivale la pregunta se crea ticket.



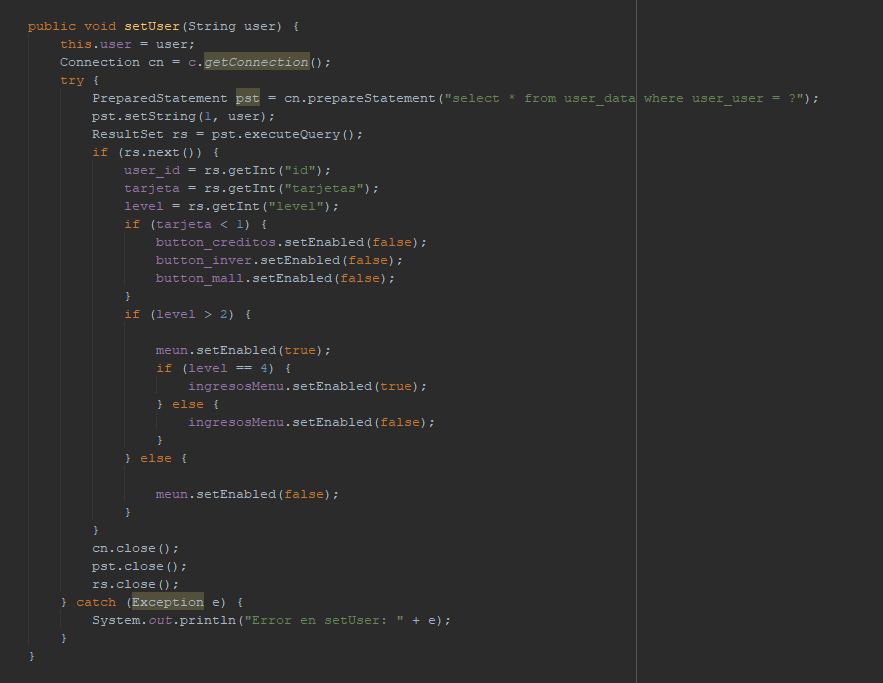
Registrarse: Ingreso de datos para que el cliente se registre. Verifica que no exista el usuario ingresado, ya que podría dar problemas y se genera una pregunta de seguridad.



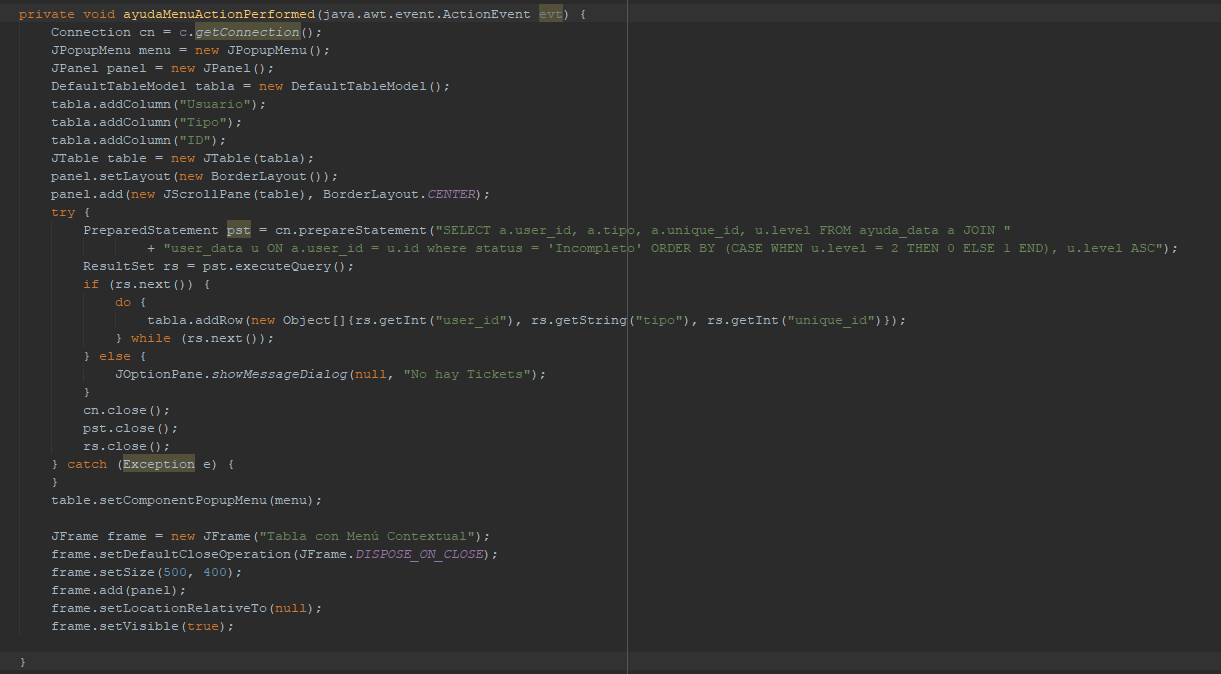
Panel

El panel principal es el que posee todas las acciones que pueden hacerse sobre la Base de Datos, discriminando en los niveles de los usuarios. Si es un Cliente del banco, solo tendrá acceso a modificar su cuenta y pedir paquetes al mismo banco. En caso distinto, podrán usar lo siguiente:

setUser: Obtiene el id, tarjeta y nivel, y en base a esto si este habilitado el menu.

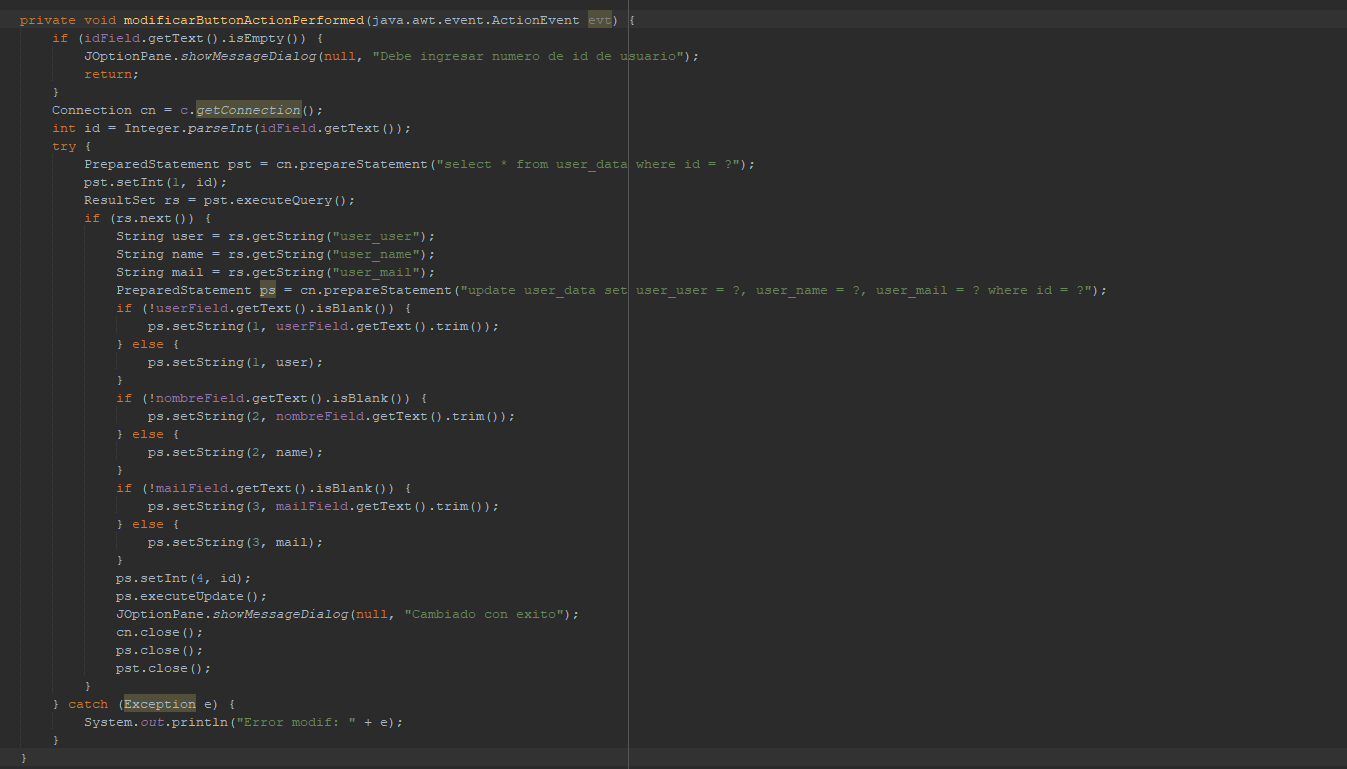


Ayuda: Te da los tickets de ayuda en una tabla obtenido de la Base de Datos, dándole prioridad a las cuentas PyME. Esto se logra con ORDER BY level, siendo level 2 el de las PyME y se obtienen las incompletas. Si no se encuentra nada, se informa.

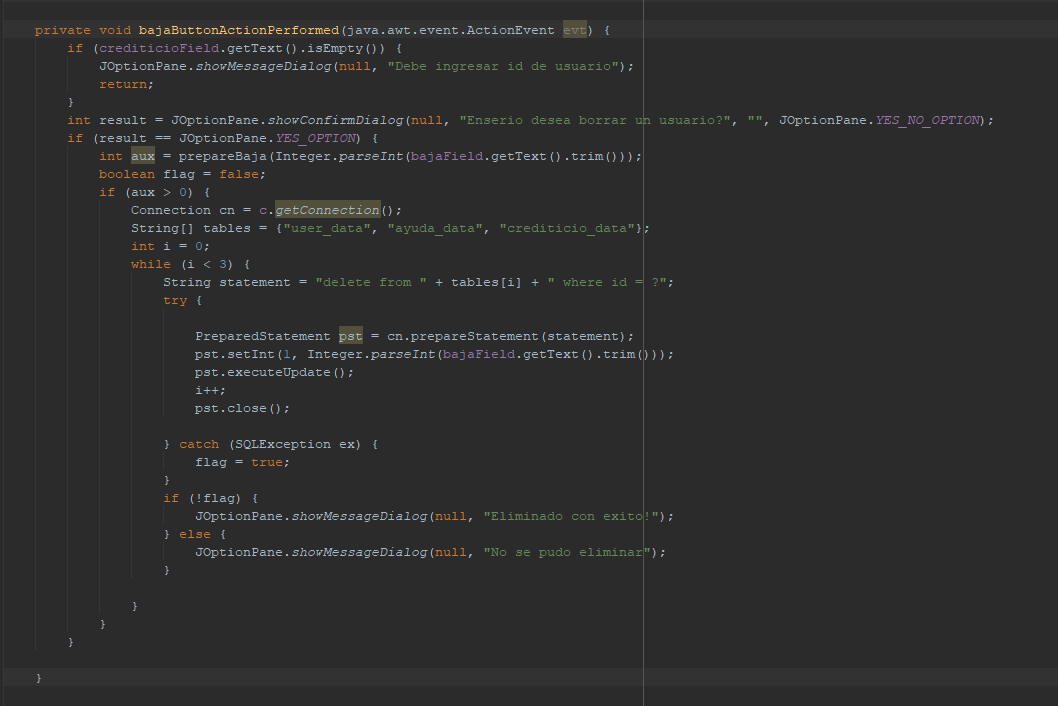


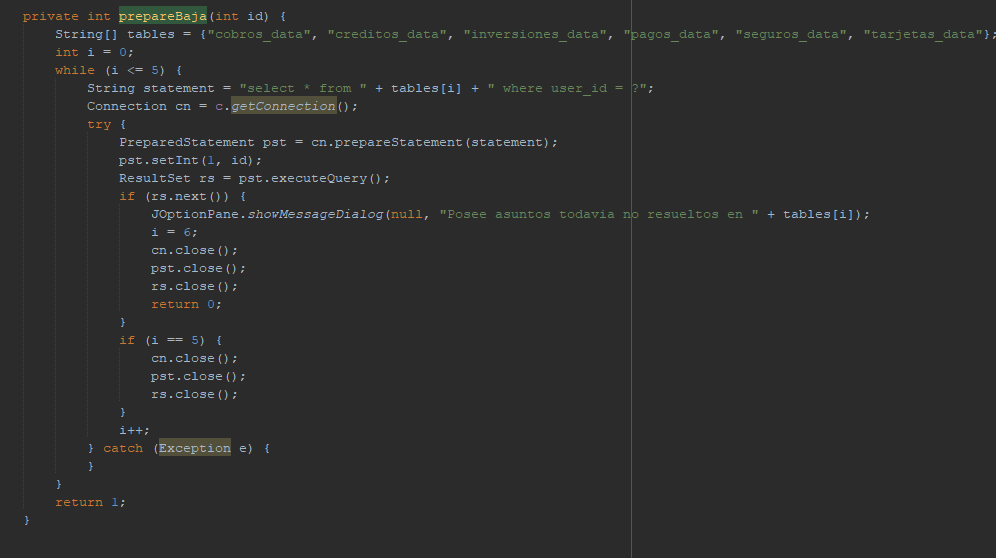
Usuarios: Podes modificar los datos del usuario y otorgar créditos, seguros y vista sobre los productos del mismo, y dar de baja al Usuario.

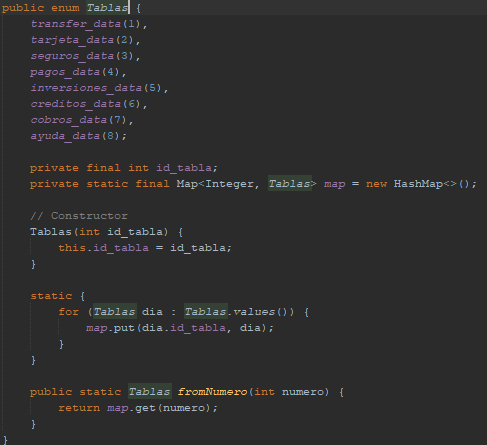
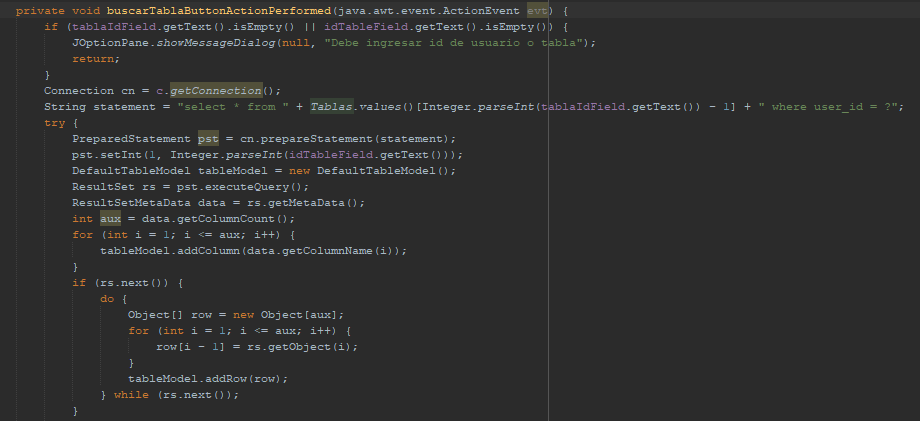
Modificar: Obtiene los valores anteriores y cambia solo lo introducido.

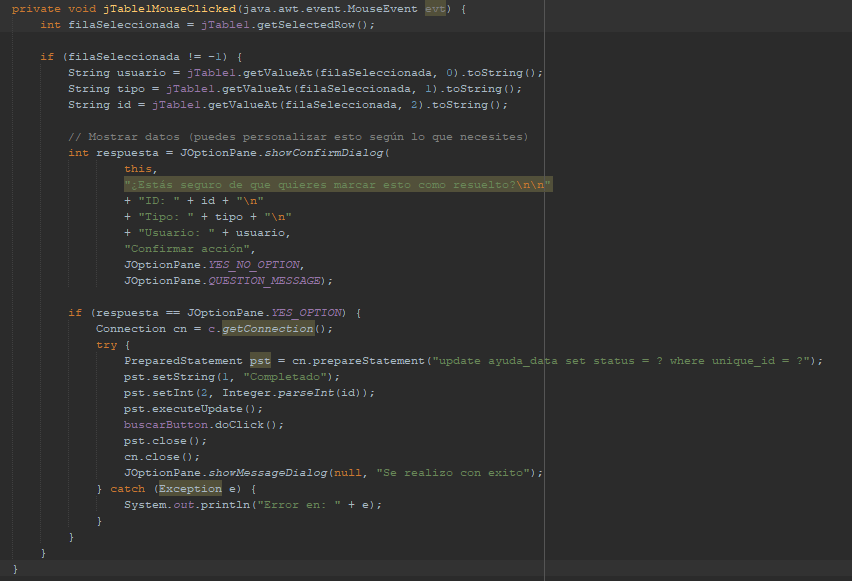
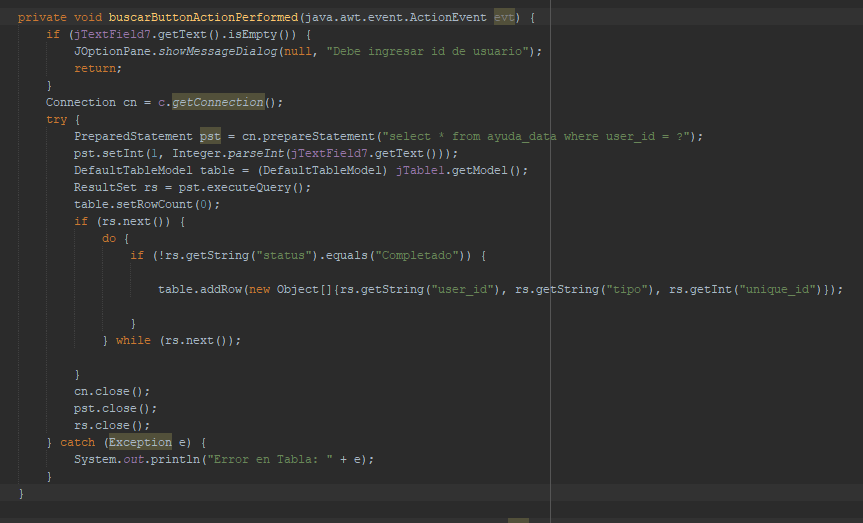


* Baja de Usuario: este primero entra en prepareBaja, el cual verifica si no existen datos pendientes de otros movimientos con el banco.

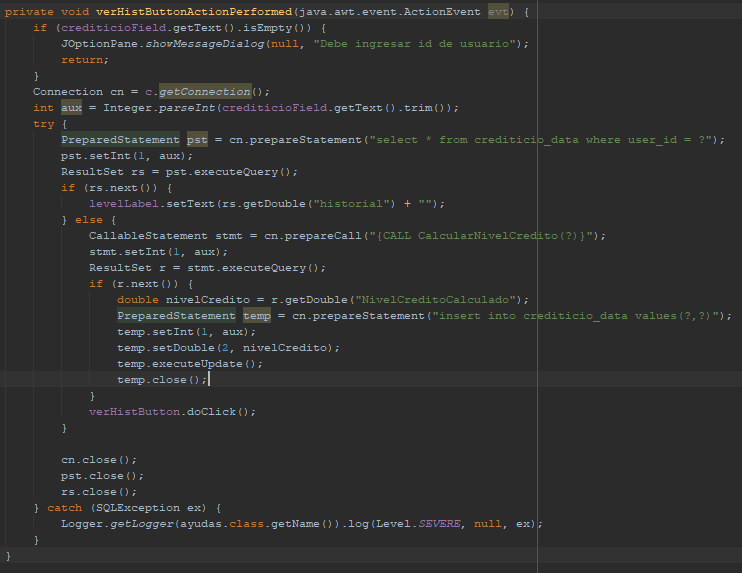




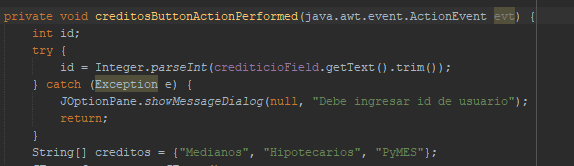
* Tabla: Busca los datos del cliente en la tabla mencionada por tablas
* Buscar Tabla: Te muestra sobre la tabla de la GUI los tickets que están incompletos, adicionalmente también tiene la función en la propia tabla de tocar y cerrar los tickets

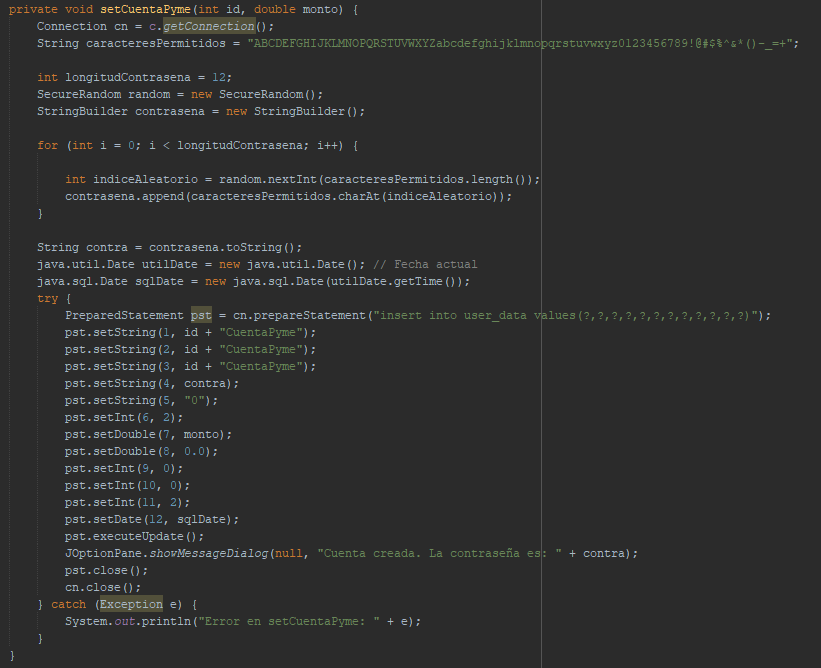


* Ver Historial: te muestra el historial crediticio, el cual, si no existe, llama a la función para que lo cree.



* Ver Creditos: Este sirve para otorgar los créditos correspondientes (La foto muestra como se llama el método porque es largo y tiene muchos elementos de Swing que son para crear un JFrame). En este método, mediante un getStateChange() de un JComboBox, crea un switch que toma la opción elegida como un String. En base al historial crediticio, se podrá (o no) tomar los créditos correspondientes, en caso que no le dé el nivel se mencionará con un JOptionPane. En el caso especial que elija un crédito PyME, se abrirá una cuenta especial para ello. Y para dejar asentado las gestiones, se guardan en setStatementC, que crea un historial en créditos\_data. setCuentaPyme cuenta con un creador de contraseña aleatorio.





* Ver Tarjetas: Este sirve para otorgar, dar de baja y actualizar las tarjetas (La foto muestra como se llama el método porque es largo y tiene muchos elementos de Swing que son para crear un JFrame). Tiene botones que se usan para este fin, y usan los métodos: getMetaData(se encarga de mostrar en tabla la tabla de la Base de Datos); setAlta(da alta de tarjeta); setBaja(da baja de tarjeta, siempre que no haya pagos pendientes).



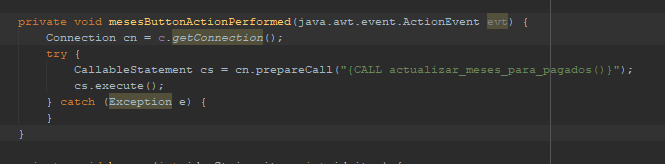
* Ver Seguros: Esta es la parte de cotización de los mismos(La foto muestra como se llama el método porque es largo y tiene muchos elementos de Swing que son para crear un JFrame), la cual se divide en joyas, domésticos y otros. Este con un JComboBox elije cual y, si tiene el historial crediticio necesario, lo entrega con un sistema de cotización. Para dejar asentado esto usa setStatementS, que deja un registro en seguros.

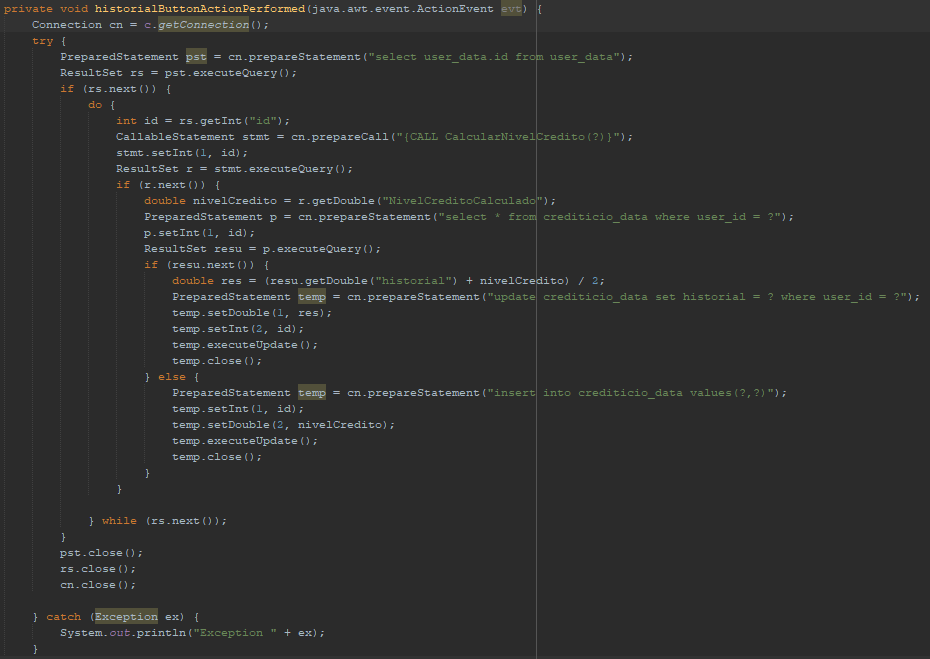


Ingreso: Solo el admin puede ingresar a modificar tanto Ids de ítems relacionados a ventas, como las llamadas correspondientes para actualizar los valores crediticios y los pagos mensuales. Cada botón utiliza un método compartido para generar una tabla con los datos cuando tocas el botón Ver [objeto] con el método setTable, pueden modificar o agregar con el botón Modificar, creando un nuevo JFrame que te da para agregar en base a la tabla elegida (es un método compartido de modifEvent). En el caso especial de ítems tiene un JFileChooser para el icono del item. Borrar también es un método compartido, siendo este borrar. Al ser métodos compartidos, todos usan un array de Strings con los nombres de las tablas que se agregan a un Statement.



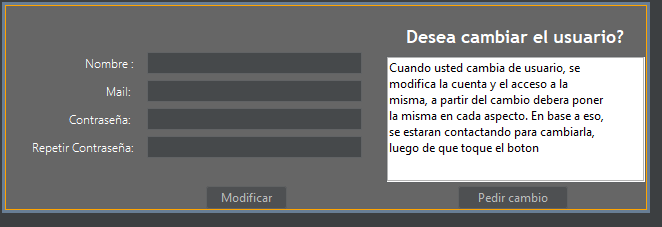
También posee 3 botones más que sirven para realizar movimientos mensuales de manera manual sobre la base de datos. A este fin tenemos:

* Plazos Fijos: Realiza los pagos sobre estos plazos, en caso que haya llegado al límite de meses. Si se llega a pagar, se descontará de la cuenta de Admin
* Actualizar Meses: Llama a la función actualizar\_meses\_para\_pagados, que suma un mes siempre que no se pase de los meses correspondientes. 
* Historial crediticio: Llama a la función CalcularNivelCrediticio, la cual devuelve un nivel de crédito que, en base a si existe ya uno anterior, suma y divide y, en caso que no, se crea con este valor.



Modificar Usuario

Modificar usuario posee lo que dice, el cliente puede modificar su nombre, mail y contraseña(la contraseña posee un passwordListener), si no se modifica una columna se tomara el valor anterior desde una consulta previa. En caso que quiera modificar su usuario, esto generara un ticket para cambiarlo desde el empleado, asi no genere errores.

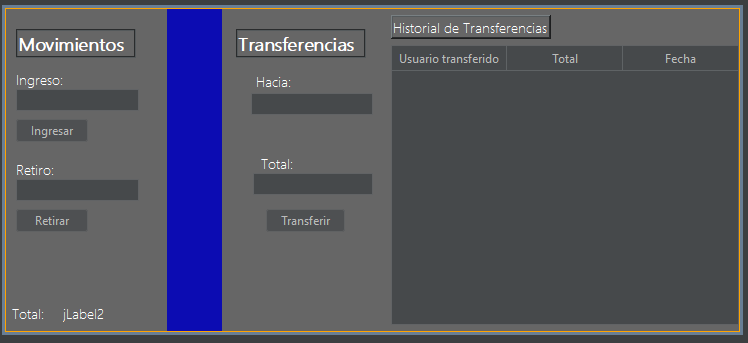


Plata

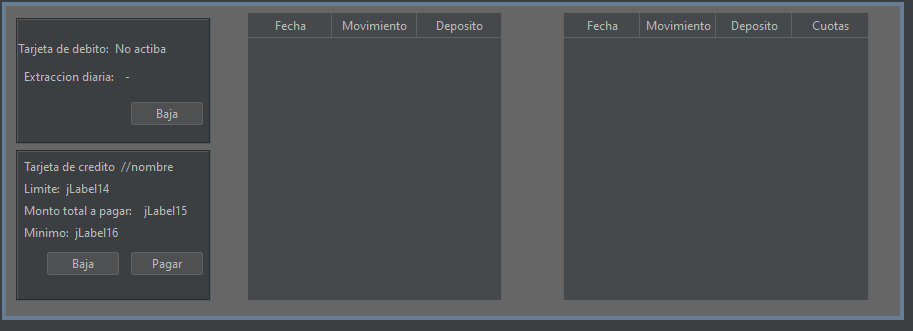
Se encarga de movimientos de plata, en este caso, al ser simplemente una demo y no un programa real, se puede ingresar y sacar la cantidad deseada como si fuera personal, adicionalmente se puede transferir.

Movimientos: Se puede insertar plata, siendo variables double pudiendo ingresar o retirar con decimos. No se podrá retirar mas de lo que se tiene en la cuenta, que figura abajo a la izquierda como Total.

Transferencias: Se realizan transferencias de la cuenta del cliente a una distinta, la cual no puede superar el valor total mencionado anteriormente. Cuenta con una jTable que muestra las transferencias hechas con los resultados de la Base de Datos.

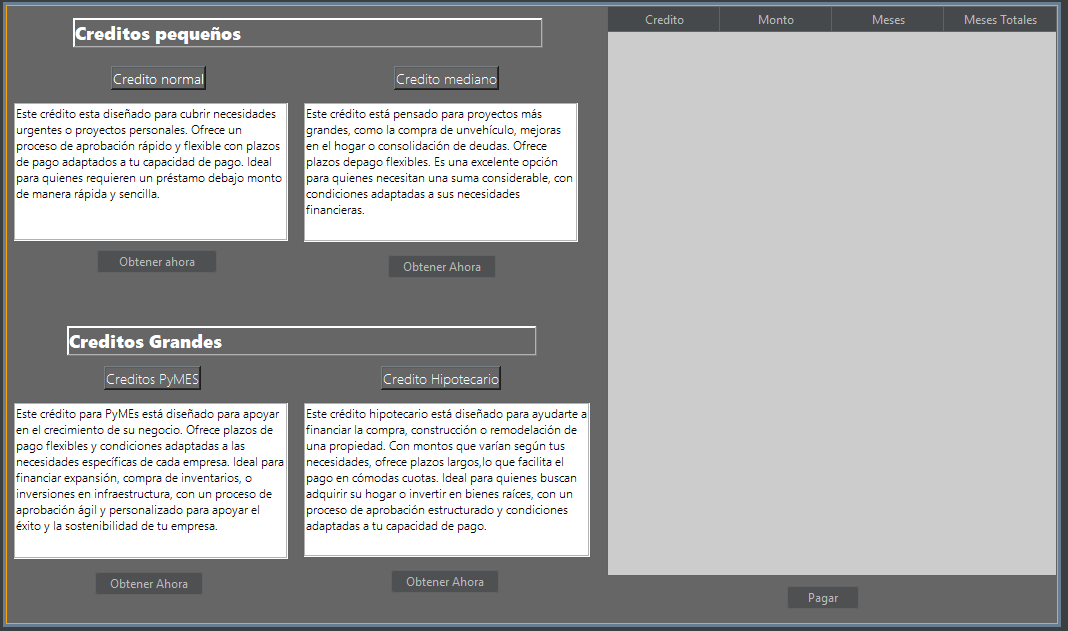


Tarjetas

El cliente podrá dar de baja las tarjetas de crédito y debito, consultar los movimientos hechos en las mismas mediante unas jTable , pagar y podrá pedir la obtención de las tarjetas correspondientes. Este JFrame posee varios JPanel con CardLayout que sirven para mostrar si el cliente tiene o no tarjetas, estos dependerán que el número que tenga en tarjetas en user\_data. Cuenta con un botón para pagar el total de la tarjeta y que generara distintos pagos\_data independientes que se agregaran. 

Créditos

El cliente podrá seleccionar créditos para su cuenta. Los pequeños, el primero es hasta 10000 pesos, el segundo es mayor pero solo se obtiene si cumple con los requisitos de historial crediticio, lo mismo aplica para los dos mayores, que generaran un ticket de ayuda. Posee una jTable que muestra los créditos y un botón para pagar, ambos obtienen los datos de la Base. Podrá pagar la totalidad de lo que debe y generará distintos pagos\_data para agregarle. Al acceder a un crédito se descuenta de la cuenta Admin.

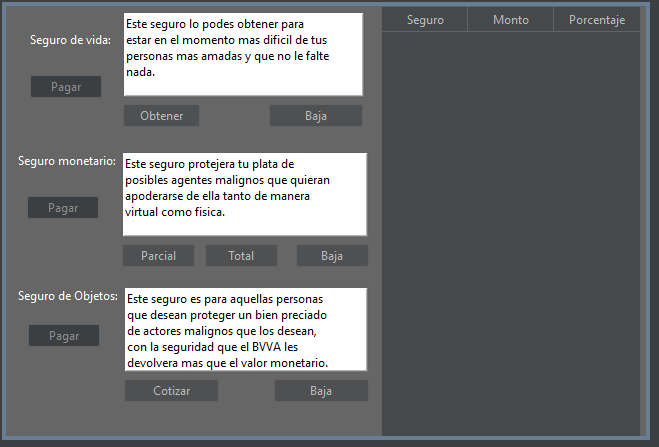


Seguros

El cliente podrá pedir Seguros de vida, de la plata actual y de objetos personales. Posee botones para pagar cada seguro de manera particular, y la baja de los mismos. Los seguros de vida se otorgarán en caso que el income de la Base de Datos sea mayor a 0, y podrá pedir hasta 4 de los mismos. El seguro de la plata actual tiene dos maneras, el parcial y el total, los cuales el total solo se puede si es que income es mayor a 0. Los objetos personales se cotizarán con un empleado, por lo que generara un ticket de ayuda. Además, una JTable les mostrar los seguros activos.

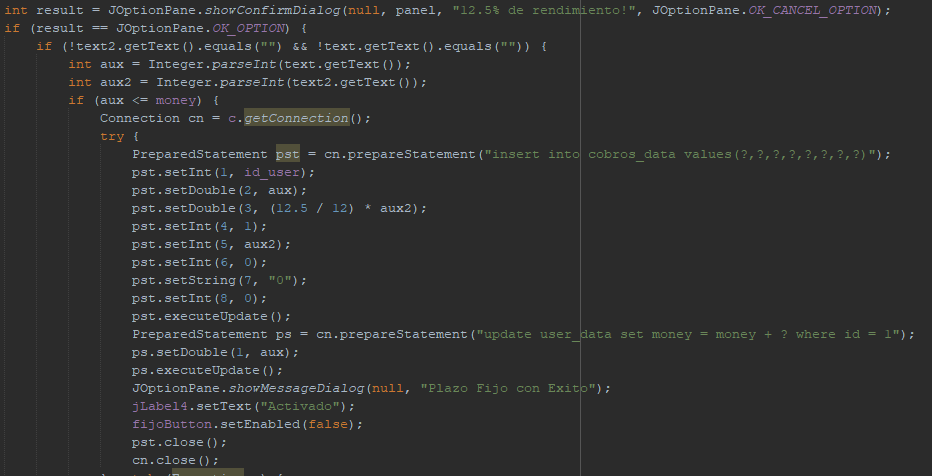
Tiene distintos métodos, pero el más usado es isUsed, que revisa que el nombre del seguro que está tratando de agregar ya no este usado, caso contrario se informará y se tendrá que usar otro. Además, también usa el método recurrente de setPago, que toma el pago de cada uno de los tipos de seguro, y los agrega a pagos\_data. Luego de pagar, otro método llamado updateCrediticio pasa todos los pagados a 1.

Las bajas se pueden dar varias a la vez, haciendo uso de un JCheckBox en cada uno y solo se darán de baja si pasaron menos de 7 días de la adquisición o ya está pagado.



Inversiones

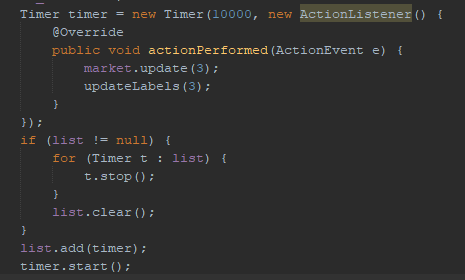
El cliente podrá optar por adquirir un plazo fijo, el cual es único y se carga con la cantidad de meses que lo quiere invertir; Contara con inversiones riesgosas las cuales podrán dar más rendimiento en base a su porcentaje, pero podrían perder plata por lo mismo; y con un stock market, el cual se actualiza en tiempo real sobre la Base de Datos y puedes comprar o vender los mismo con el valor correspondiente.

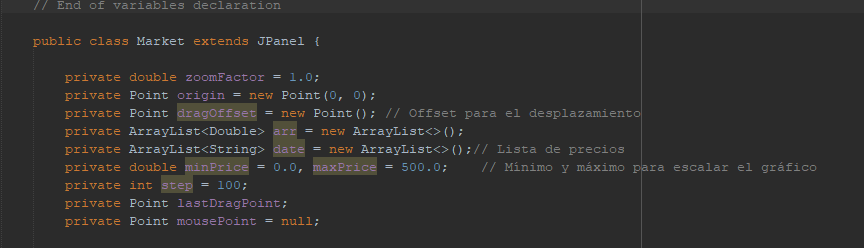
El plazo fijo añadirá plata a la cuenta admin y creará un nuevo registro en cobros\_data, pero solo se podrá tener uno por vez. 

Las inversiones riesgosas se adquieren con un JChechBox, el cual estará añadido a un field para indicar la cantidad. Estas se añadirán a la JTable y podrás ver todas las inversiones allí. En el momento de vender estas acciones, se calcula el porcentaje anual dividido los 365 días para sacar un rendimiento, y se multiplica por un numero random que va de -1 a 1, indicando si ganaste o perdiste plata.



El stock market funciona obteniendo los valores del precio de los stocks de la Base de Datos desde el historial, ósea stocks\_prices, y dibuja las líneas correspondientes como picos de lo que paso, además también muestra hora y fecha de estos cambios. Al tener un timer, se actualiza solo cada vez que este actualizado en la Base.

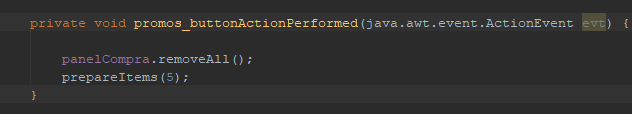


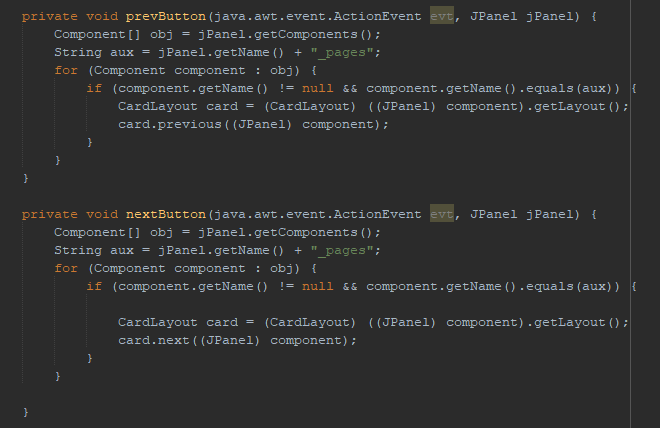
Se compone de un JPanel con CardLayout, el cual cambia de cada stock en base al JPanel seleccionado arriba(“Tabs”). Los stocks se pueden vender todos juntos o por separado, con unos JSpinners que van para seleccionar el total y se venden al valor actual del stock. Se trata como una clase nueva sobre la misma clase.

Mall

El cliente puede adquirir ítems que venda el banco, los cuales tendrán 5 secciones distintas, en las cuales una será por los puntos acumulados al comprar con las tarjetas. También se podrá ver los ítems que posee el cliente.

Mall se compone principalmente del metodo prepareItems, que crea labels, buttons y JPanels de 4 en 4, además de dibujar las líneas que separan los ítems. Cada button hace que el JPanel principal, teniendo CardLayout, cambie de JPanel e inserte los ítems allí. Posee dos botones adicionales que pasan de página, tanto la anterior como la siguiente y el button de compra discrimina si se hace con tarjeta de crédito o débito.





Base de Datos

El corazón del banco, posee:

* User\_data: Se guardan todos los datos de los usuarios del mismo, todo hace referencia a la tabla.
* Transfer\_data: Se guarda todas las transferencias hechas de los clientes.
* Tarjeta\_data: Se guardan todos los movimientos hechos
* Tarjeta\_id: Se guarda la info de las tarjetas y su limite
* Stock\_prices: Guarda el historial de los precios
* Stock\_inver: Guarda el historial de los stocks comprados por clientes
* Stocks\_data: Guarda el nombre y precio de los stocks actuales
* Seguros\_id: Guarda los nombres de los seguros
* Seguros\_data: Guarda la informacion de los seguros de los clientes
* Preguntas\_data: Guarda la informacion de las preguntas de los clientes
* Pagos\_data: Guarda la información de los pagos de los clientes
* Items\_store: Guarda los ítems de los clientes
* Items\_data: Todos los ítems del Mall se encuentran aca
* Inversiones\_id: Guarda los datos de las inversiones
* Inversiones\_data: Guarda los datos de las inversiones de los clientes
* Créditos\_id: Guarda los datos de los créditos
* Creditos\_data: Guarda los datos de los créditos de los clientes
* Crediticio\_data: Guarda el historial crediticio de los clientes.
* Cobros\_data: Guarda los plazos fijos de los clientes
* Ayuda\_data: Guarda los tickets generados
* Eventos:
  + **UpdateStockPricesEvent:** Actualiza los precios del stock, guardándolos en stock\_prices.
* Funciones:
  + **CalcularAntiguedadPrecisa**: Calcula con la fecha actual y la de creación de la cuenta y saca el promedio si de cuánto tiempo es.
  + **CalcularPagoSinMes**: Calcula la cantidad de días pasados del pago. Si no está pagado, se guarda hasta el día actual.
  + **CalcularPorcentaje:** Calcula cuanto del income es gastado en general.
  + **CalcularPorcentajeLimite**: Calcula la cantidad del límite de la tarjeta de crédito que este ocupando el cliente.
  + **CalcularPromedio**: Suma la totalidad de créditos y seguros, y también sobre la variedad de estos. Te da un puntaje en base a estos parámetros
  + **CalcularPromedio**: Usa todos los anteriores, verifica que haya algún historial crediticio anterior y se lo suma y divide por esto. Si no existe, crea uno.
* Procedimientos:
  + **actualizar\_meses\_para\_pagados**: Actualiza los meses de pago de plazos fijos y créditos.