

**Actividad 3. Programa Banco Mexicano (parte 2)**

**Lenguajes de Programación IV**

**Ingeniería en Desarrollo de Software**

**Tutor: Aarón Iván Salazar Macías**

**Alumno: José Luis Pacheco González**

**Fecha: 27 de marzo 2024**

## Índice

Introducción .....	3
Descripción .....	4
Justificación .....	5
Desarrollo .....	6
• Interfaz .....	6
• Codificación .....	8
Conclusión .....	16
Referencias .....	17

## Introducción

Esta es la segunda parte del programa del Banco Mexicano, donde el enfoque es implementar las funciones esenciales que permiten a los clientes realizar transacciones bancarias básicas: depósitos, retiros y consultas de saldos. Utilizando Java en conjunto con NetBeans para el desarrollo del código y MySQL Workbench para gestionar la base de datos, se puede crear un sistema robusto y eficiente.

La funcionalidad de depósitos permite a los clientes aumentar el saldo de sus cuentas bancarias mediante la introducción de una cantidad específica de dinero. Por otro lado, la opción de retiros les permite a los clientes disminuir su saldo mediante la extracción de fondos, siempre y cuando tengan suficientes recursos disponibles. Además, la función de consulta de saldos proporciona a los clientes una visión instantánea y precisa de la cantidad de dinero disponible en sus cuentas.

Esta segunda etapa del proyecto no solo implica la codificación de las funciones mencionadas, sino también asegurarse de que interactúen adecuadamente con la base de datos, garantizando la integridad y la precisión de la información financiera de los clientes.

## Descripción

El Banco Mexicano está desarrollando un programa para permitir a sus clientes realizar operaciones bancarias básicas, como depósitos, retiros y consultas de saldo. Han encargado a un ingeniero en desarrollo de software que diseñe una base de datos para satisfacer estas necesidades.

En la etapa anterior, se creó la interfaz del menú principal del programa junto con las funciones de "Depósito" y "Retiro". Ahora, el objetivo es completar la funcionalidad del menú del programa. Este menú debe incluir opciones para "Depósito", "Retiro", "Saldo" y "Salir", solicitando las respuestas del usuario mediante entrada de teclado.

Cuando el usuario elige la opción "Saldo", el programa mostrará en pantalla el monto disponible. Si elige "Salir", el programa informará que la sesión ha sido cerrada.

El desarrollo de estas funciones es crucial para proporcionar una experiencia bancaria completa y satisfactoria a los clientes del Banco Mexicano, ofreciendo facilidad de acceso y operatividad dentro del sistema bancario.

## **Justificación**

La implementación de las funciones de depósitos, retiros y consultas de saldos en el programa del Banco Mexicano desempeña un papel crucial en la experiencia del cliente y en la eficiencia operativa del sistema bancario en su conjunto. Estas funciones no solo facilitan la realización de transacciones financieras cotidianas, sino que también contribuyen a la confianza y satisfacción del cliente.

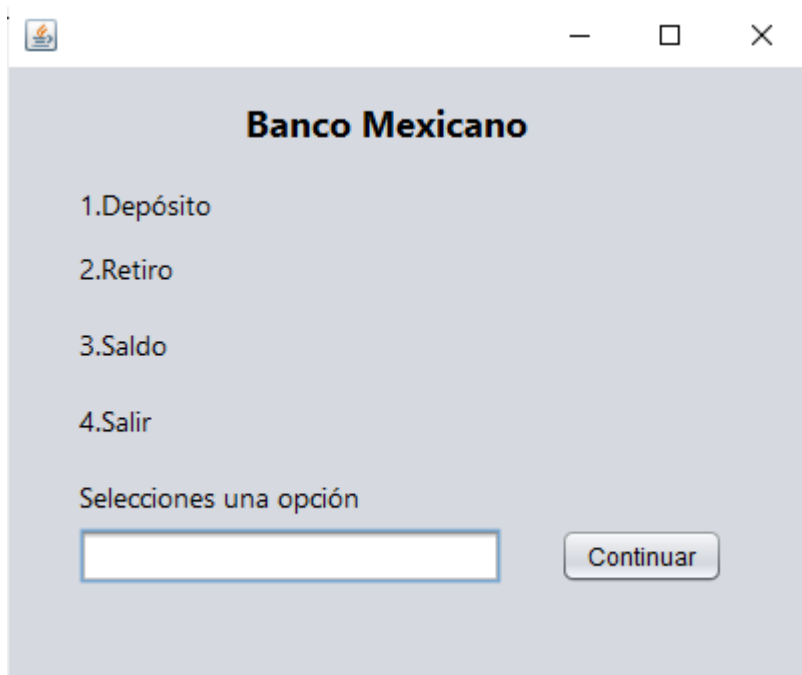
La capacidad de realizar depósitos brinda a los clientes la flexibilidad para agregar fondos a sus cuentas de manera conveniente y segura, lo cual es esencial para mantener y aumentar el capital en el banco. Por otro lado, la opción de retiros permite a los clientes acceder a sus fondos cuando lo necesiten, brindando una experiencia bancaria ágil y accesible. Además, la función de consulta de saldos proporciona transparencia y control al permitir que los clientes monitoreen sus finanzas en tiempo real.

La elección de utilizar Java en conjunto con NetBeans para el desarrollo del código y MySQL Workbench para la gestión de la base de datos es fundamental. Estas herramientas proporcionan un entorno de desarrollo robusto y eficiente, permitiendo una implementación ágil y escalable de las funciones bancarias. Además, la integración con una base de datos garantiza la seguridad y la integridad de la información financiera del cliente, lo cual es esencial para cumplir con los estándares regulatorios y mantener la confianza del público en el sistema bancario.

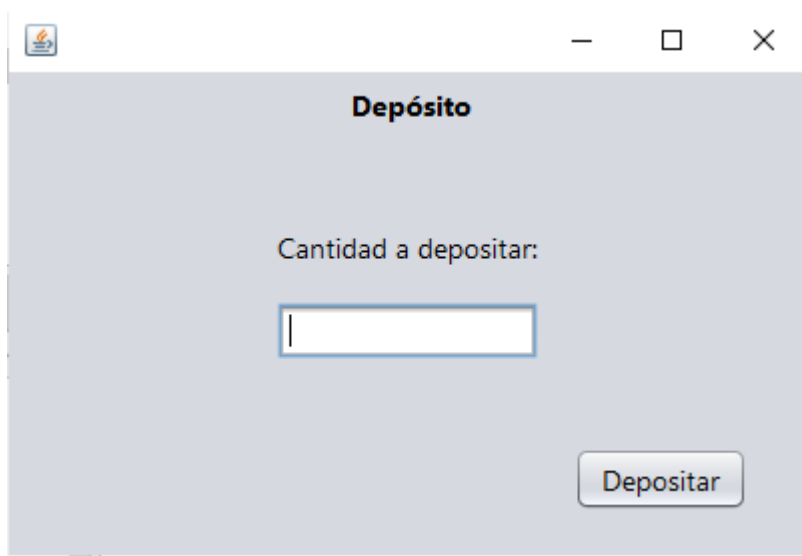
## Desarrollo

- **Interfaz**


La interfaz continúa siendo la misma que se diseñó en la actividad anterior, la cual se muestra a continuación.



A screenshot of a Java Swing window titled "Banco Mexicano". The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main content area has a light gray background. It displays a menu with four options: "1.Depósito", "2.Retiro", "3.Saldo", and "4.Salir". Below the menu, there is a prompt "Selecciones una opción" followed by a text input field and a "Continuar" button.




A screenshot of a Java Swing window titled "Depósito". It has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main content area has a light gray background. It displays a prompt "Cantidad a depositar:" followed by a text input field. At the bottom right, there is a "Depositar" button.

 — □ ×

**Retiro**

Monto:

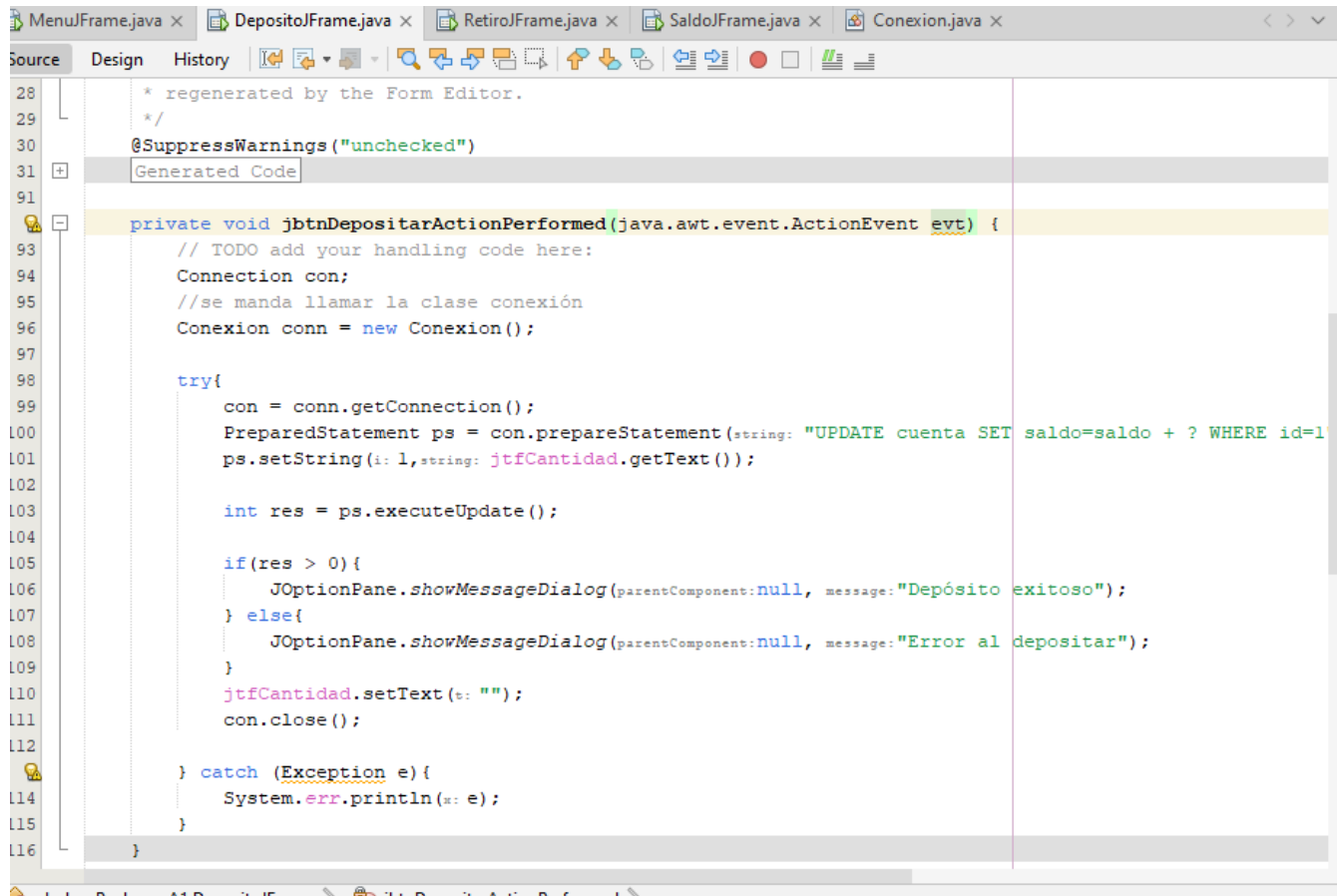
 — □ ×

**Saldo**

Monto:

- **Codificación**

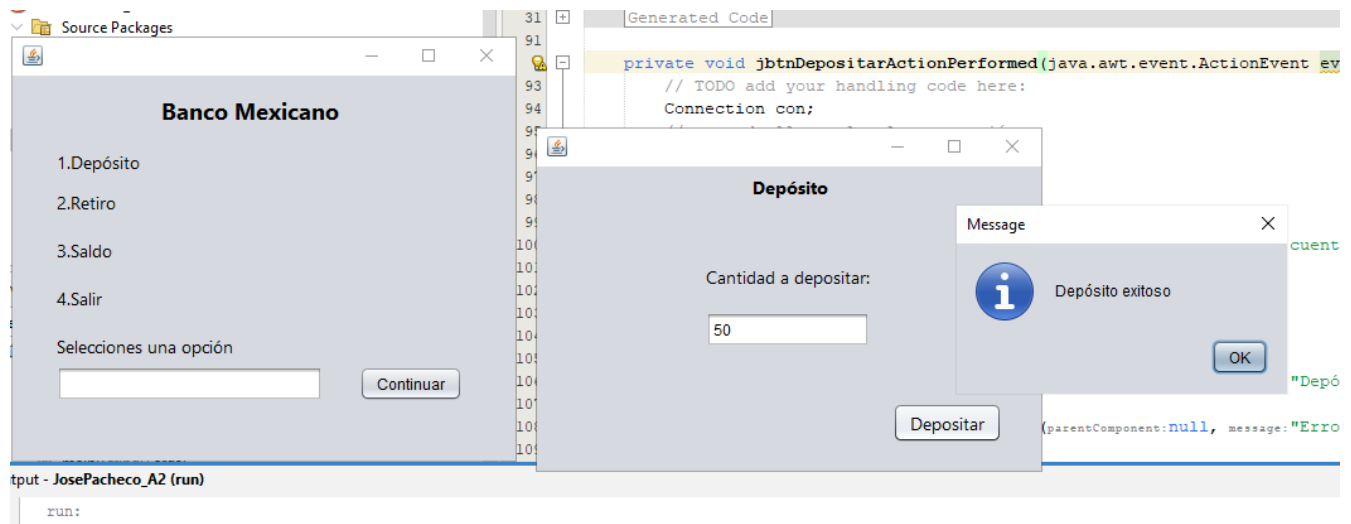
Se genera el código para realizar la conexión en del depósito, estableciendo la conexión, la actualización de saldo, así como los mensajes de “Depósito exitoso” y “Error al depositar”.



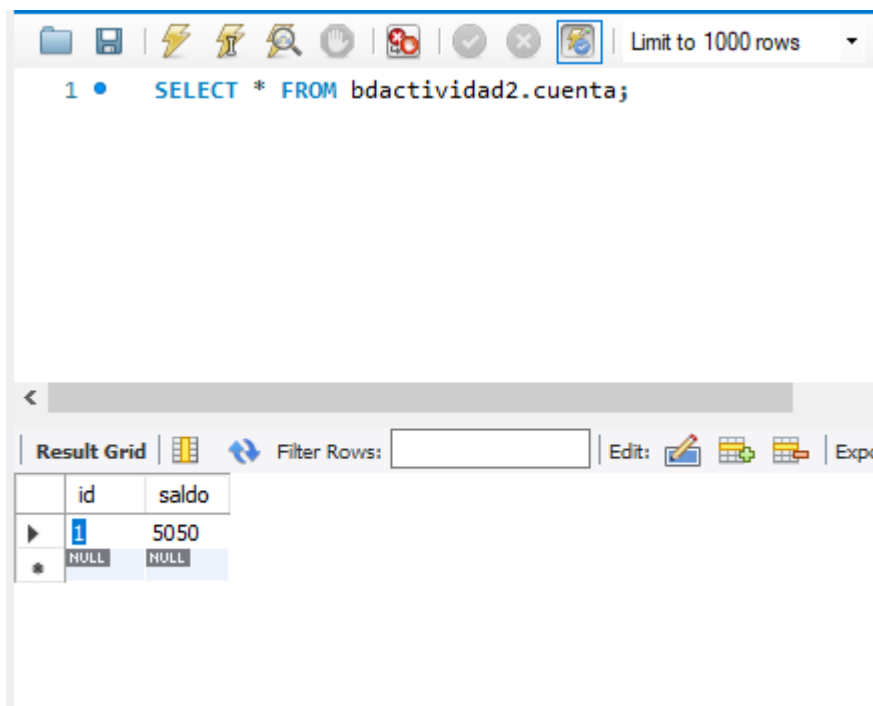
```
28      * regenerated by the Form Editor.
29      */
30      @SuppressWarnings("unchecked")
31      Generated Code
91
92      private void jbtnDepositarActionPerformed(java.awt.event.ActionEvent evt) {
93          // TODO add your handling code here:
94          Connection con;
95          //se manda llamar la clase conexión
96          Conexion conn = new Conexion();
97
98          try{
99              con = conn.getConnection();
100              PreparedStatement ps = con.prepareStatement(string: "UPDATE cuenta SET saldo=saldo + ? WHERE id=1");
101              ps.setString(i: 1,string: jtfCantidad.getText());
102
103              int res = ps.executeUpdate();
104
105              if(res > 0){
106                  JOptionPane.showMessageDialog(parentComponent:null, message:"Depósito exitoso");
107              } else{
108                  JOptionPane.showMessageDialog(parentComponent:null, message:"Error al depositar");
109              }
110              jtfCantidad.setText("");
111              con.close();
112
113          } catch (Exception e){
114              System.err.println(e);
115          }
116      }
```



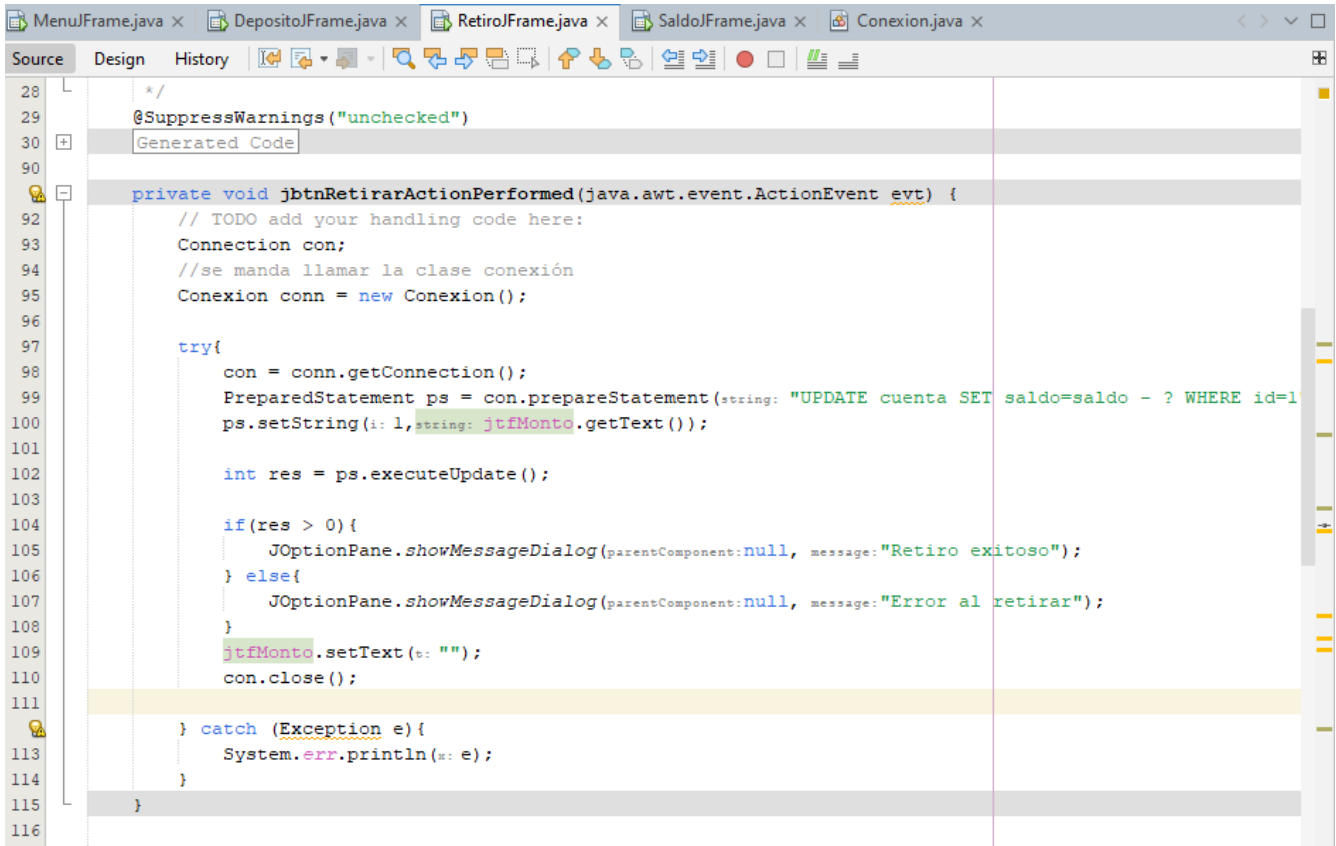
Después de generar el código necesario se realiza un depósito para validar que todo funcione de manera correcta, como se muestra en la siguiente captura.



Al actualizar se muestra el saldo depositado en la aplicación, ya que se contaba solo con \$5,000 en la cuenta.

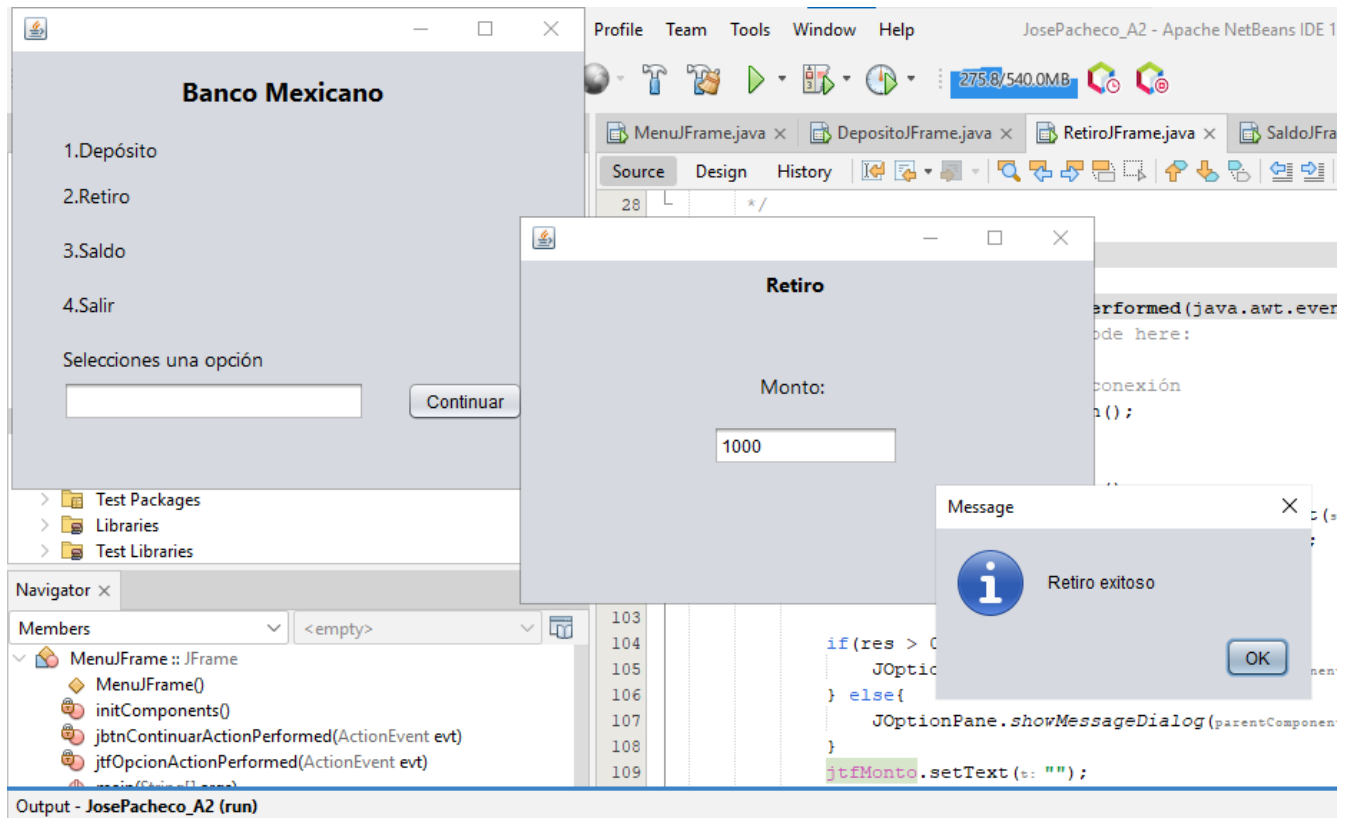


Como paso siguiente se realiza el código para poder realizar el retiro. Es similar al código del depósito, solo se cambia los parámetros que corresponden al retiro, se agregan también los mensajes de “Retiro exitoso” y “Error al retirar”.

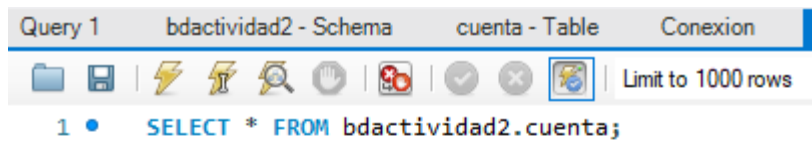


```
28  */
29  @SuppressWarnings("unchecked")
30  Generated Code
90
91  private void jbtnRetirarActionPerformed(java.awt.event.ActionEvent evt) {
92      // TODO add your handling code here:
93      Connection con;
94      //se manda llamar la clase conexión
95      Conexion conn = new Conexion();
96
97      try{
98          con = conn.getConnection();
99          PreparedStatement ps = con.prepareStatement(string: "UPDATE cuenta SET saldo=saldo - ? WHERE id=1");
100         ps.setString(1, jTextFieldMonto.getText());
101
102         int res = ps.executeUpdate();
103
104         if(res > 0){
105             JOptionPane.showMessageDialog(parentComponent:null, message:"Retiro exitoso");
106         } else{
107             JOptionPane.showMessageDialog(parentComponent:null, message:"Error al retirar");
108         }
109         jTextFieldMonto.setText("");
110         con.close();
111
112     } catch (Exception e){
113         System.err.println(e);
114     }
115 }
116
```

También se realizó la prueba exitosa del retiro, como se muestra en la captura.

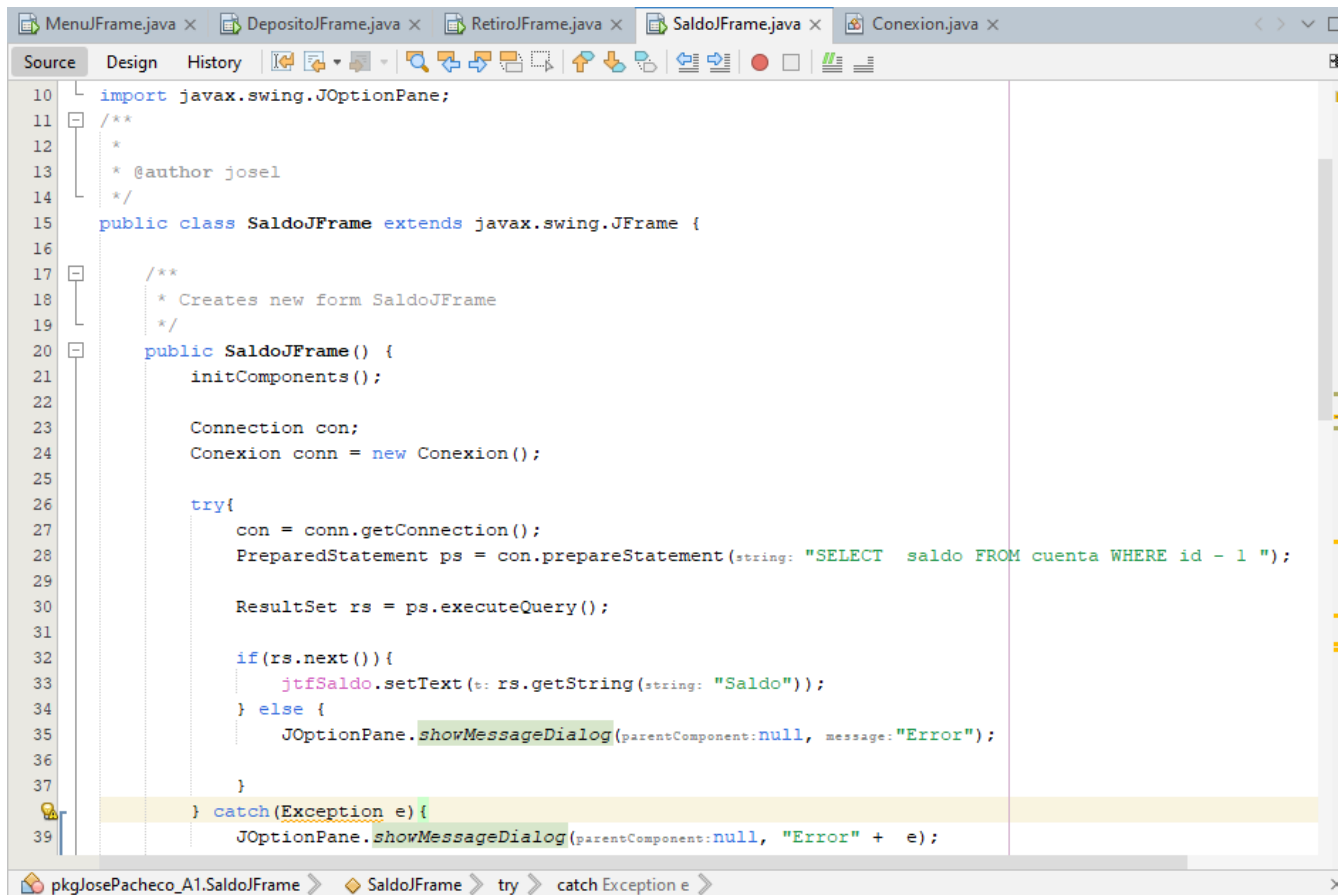


Aquí se muestra como se restaron los \$1,000 retirados, ya que se contaba con un saldo inicial de \$5,050.



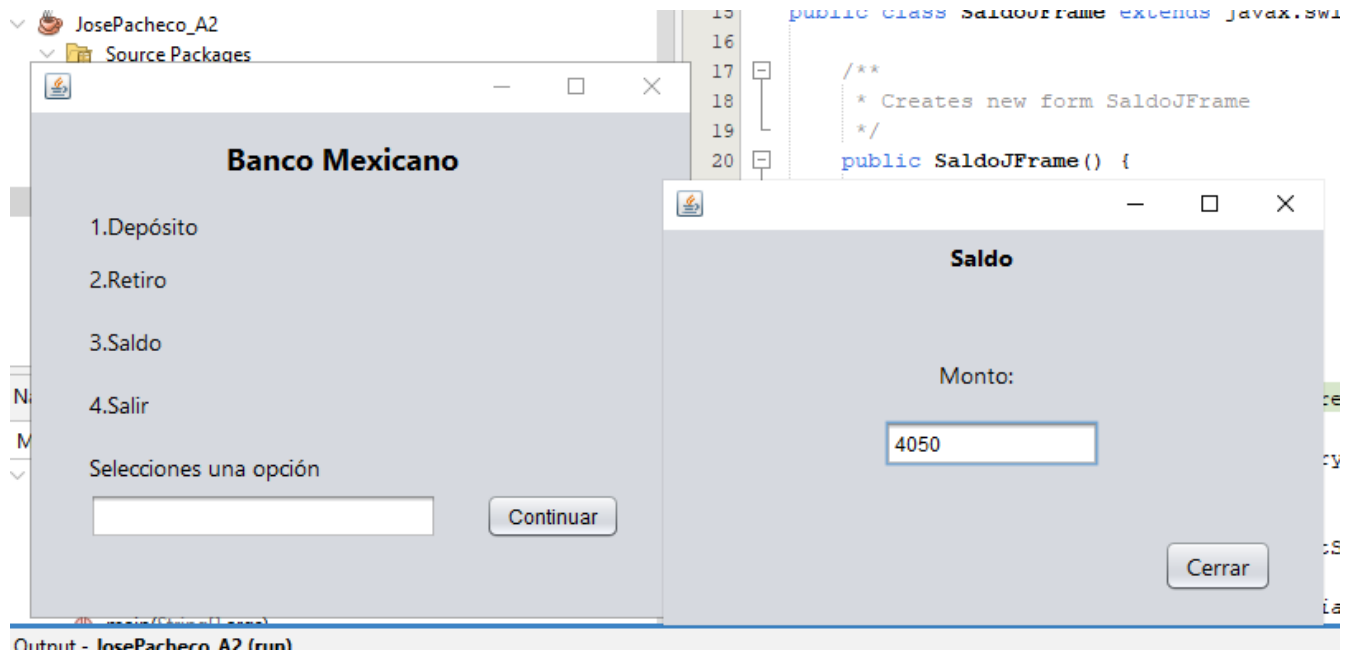
	id	saldo
▶ 1	4050	
*	NULL	NULL

Esta captura corresponde al código del saldo, donde también se establece la conexión, se indica de donde tomará la información en la base de datos y los mensajes correspondientes.




```
10 import javax.swing.JOptionPane;
11 /**
12  *
13  * @author josel
14  */
15 public class SaldoJFrame extends javax.swing.JFrame {
16
17     /**
18      * Creates new form SaldoJFrame
19      */
20     public SaldoJFrame() {
21         initComponents();
22
23         Connection con;
24         Conexion conn = new Conexion();
25
26         try{
27             con = conn.getConnection();
28             PreparedStatement ps = con.prepareStatement(string: "SELECT  saldo FROM cuenta WHERE id = 1 ");
29
30             ResultSet rs = ps.executeQuery();
31
32             if(rs.next()){
33                 jtfSaldo.setText(rs.getString(string: "Saldo"));
34             } else {
35                 JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
36             }
37         } catch(Exception e){
38             JOptionPane.showMessageDialog(parentComponent:null, "Error" + e);
39         }
```

A continuación, se muestra la prueba del funcionamiento correcto de la función para consultar saldo.



Por último, se genera el código par poder ejecutar la función de salir.



The screenshot shows a Java IDE with a package explorer on the left and a code editor on the right. The package explorer shows a package named 'Saldo' with a class named 'SaldoJFrame'. The code editor displays the following Java code:

```
1 private void jButtonCerrarActionPerformed(java.awt.event.ActionEvent evt) {  
2     // TODO add your handling code here:  
3     this.dispose();  
4 }  
5  
6 private void jButtonSaldoActionPerformed(java.awt.event.ActionEvent evt) {  
7     // TODO add your handling code here:  
8 }  
9  
10 /**  
11  * @param args the command line arguments  
12  */  
13 public static void main(String args[]) {  
14     /* Set the Nimbus look and feel */  
15     Look and feel setting code (optional)  
16  
17     /* Create and display the form */  
18     java.awt.EventQueue.invokeLater(new Runnable() {  
19         public void run() {  
20             new SaldoJFrame().setVisible(true);  
21         }  
22     });  
23 }
```

## Conclusión

En conclusión, la implementación de las funciones de depósitos, retiros y consultas de saldos en el programa del Banco Mexicano, desarrollado con Java utilizando NetBeans y MySQL Workbench, ofrece una solución integral para las necesidades financieras tanto en ámbitos laborales como personales.

Las ventajas de esta solución son significativas. En el ámbito laboral, la utilización de herramientas como Java y MySQL Workbench permite a las instituciones financieras crear sistemas robustos y eficientes que agilizan las operaciones diarias, mejoran la experiencia del cliente y garantizan la seguridad de los datos financieros. Además, la integración de una base de datos proporciona una plataforma estable para gestionar grandes volúmenes de información de manera segura y organizada.

En el ámbito personal, esta solución brinda a los usuarios la comodidad y la flexibilidad de realizar transacciones bancarias desde la comodidad de sus hogares o dispositivos móviles. La accesibilidad a funciones como depósitos, retiros y consultas de saldos facilita la administración de las finanzas personales y promueve una mayor transparencia en la gestión de los recursos financieros.



## Referencias

Java con bases de datos. Crear conexión. (2021, January 29). Clasesdeinformaticaweb.com; admin. <https://www.clasesdeinformaticaweb.com/java-desde-cero/java-con-bases-de-datos/>

(N.d.). Researchgate.net. Retrieved March 28, 2024, from [https://www.researchgate.net/publication/291030523\\_Construccion\\_dinamica\\_de\\_menus\\_en\\_Java\\_La\\_clase\\_MiMenu](https://www.researchgate.net/publication/291030523_Construccion_dinamica_de_menus_en_Java_La_clase_MiMenu)



**Enlace Github**