



Actividad 3 - Programa Banco Mexicano Parte 2

Lenguajes de programación IV

Ingeniería en Desarrollo de Software

Tutor: Aarón Iván Salazar Macías

Alumno: Jusi Ismael Linares Gutiérrez

Fecha: 10/07/2023

Índice

Introducción	3
Descripción	3
Justificación	3
Desarrollo:	4
Interfaz	4
Codificación.....	6
Conclusión	7
GitHub	7

Introducción

En esta actividad, continuaremos trabajando en el desarrollo de un programa para Banco Mexicano que cumpla con las necesidades de sus clientes. En la actividad anterior, se estableció la estructura básica de la interfaz de usuario y se implementaron las funciones de depósito y retiro. Ahora, nos centraremos en completar la funcionalidad del menú principal del programa, abordando las opciones restantes.

El objetivo principal es permitir a los clientes realizar depósitos, retiros y consultas de saldo de manera eficiente y conveniente. La interfaz del programa proporcionará un menú claro y fácil de usar, donde los usuarios podrán seleccionar la acción que desean realizar ingresando la opción correspondiente.

Descripción

En la actividad anterior, ya se ha creado la interfaz del menú principal del programa, así como las funciones de depósito y retiro. Ahora, la tarea consiste en completar la funcionalidad de todas las opciones del menú.

En este sentido, al seleccionar la opción "Saldo", el programa debe mostrar en pantalla el monto actual del saldo del cliente. Esto implica acceder a la información almacenada y mostrarla de forma clara al usuario mediante un mensaje que le informe la cantidad de dinero con la que cuenta.

Por otro lado, al elegir la opción "Salir", el programa debe indicar en pantalla que la sesión se ha cerrado. Esto implica finalizar la ejecución del programa de manera adecuada y brindar al usuario un mensaje de despedida o confirmación de cierre de sesión.

Justificación

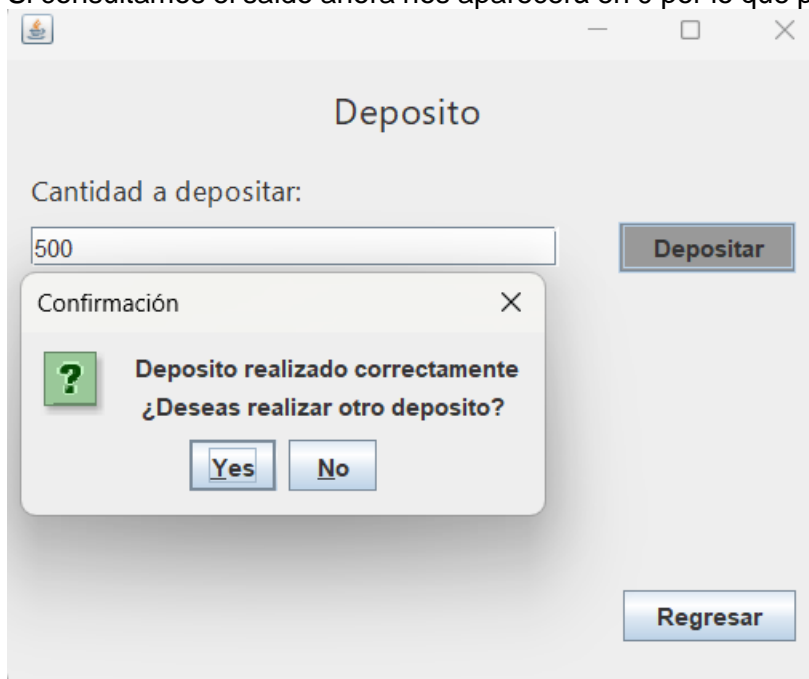
A continuación, se presentan algunas justificaciones para emplear este tipo de solución:

1. Automatización de procesos: Al desarrollar un programa que gestione las transacciones financieras, se automatizan los procesos de depósito, retiro y consulta de saldo. Esto evita la necesidad de realizar estas operaciones manualmente, lo que ahorra tiempo y reduce el riesgo de errores humanos.
2. Mejora de la eficiencia: Al tener una interfaz de usuario intuitiva y fácil de usar, los empleados pueden atender las necesidades de los clientes de manera más eficiente. El programa proporciona una forma rápida y precisa de registrar y procesar las transacciones financieras.
3. Facilidad de escalabilidad: Este tipo de solución es escalable, lo que significa que se puede adaptar y expandir para satisfacer las necesidades futuras del banco. A medida que el banco crece y se agregan más sucursales o servicios, el programa puede actualizarse y ampliarse sin dificultad.

Desarrollo:

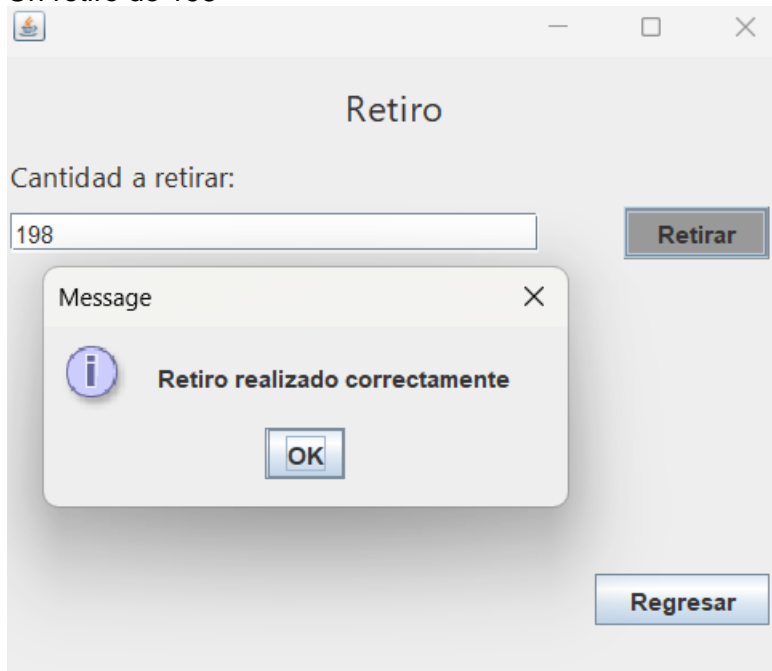
Interfaz

Si consultamos el saldo ahora nos aparecerá en 0 por lo que primero haremos un deposito de 500



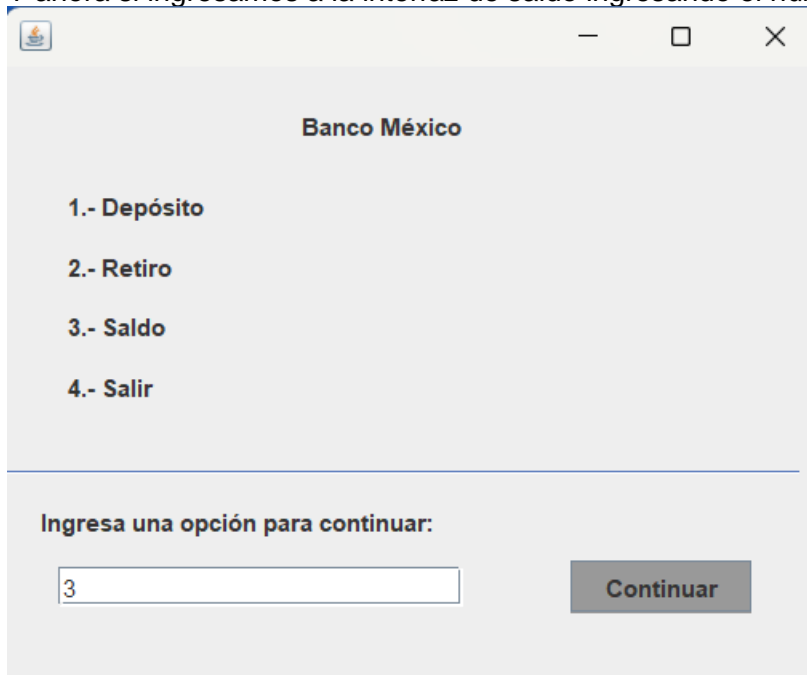
The screenshot shows a web application window titled "Deposito". It contains a text input field labeled "Cantidad a depositar:" with the value "500". To the right of the input field is a button labeled "Depositar". Below the input field, a confirmation dialog box is displayed. The dialog has a title bar "Confirmación" and a close button. It contains a green question mark icon, the text "Deposito realizado correctamente", and the question "¿Deseas realizar otro deposito?". At the bottom of the dialog are two buttons: "Yes" and "No". In the bottom right corner of the main window, there is a button labeled "Regresar".

Un retiro de 198



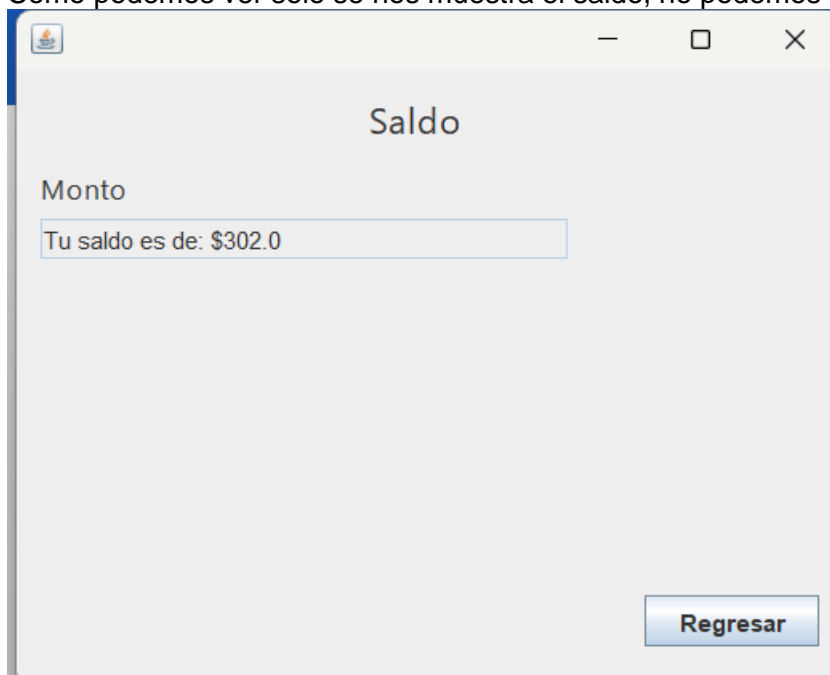
The screenshot shows a web application window titled "Retiro". It contains a text input field labeled "Cantidad a retirar:" with the value "198". To the right of the input field is a button labeled "Retirar". Below the input field, a message dialog box is displayed. The dialog has a title bar "Message" and a close button. It contains a blue information icon, the text "Retiro realizado correctamente", and an "OK" button. In the bottom right corner of the main window, there is a button labeled "Regresar".

Y ahora si ingresamos a la interfaz de saldo ingresando el número 3



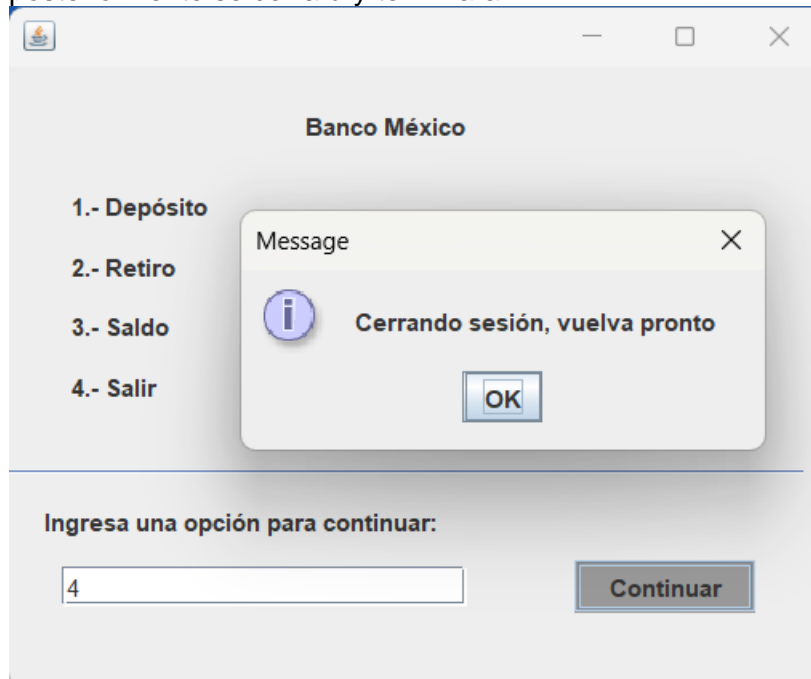
A screenshot of a web application window titled 'Banco México'. The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains a menu with four options: '1.- Depósito', '2.- Retiro', '3.- Saldo', and '4.- Salir'. Below the menu, there is a label 'Ingresa una opción para continuar:' followed by a text input field containing the number '3'. To the right of the input field is a gray button labeled 'Continuar'.

Como podemos ver solo se nos muestra el saldo, no podemos modificarlo



A screenshot of a web application window titled 'Saldo'. The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains the word 'Saldo' in a large font. Below it, the word 'Monto' is displayed. Under 'Monto', there is a text input field containing the text 'Tu saldo es de: \$302.0'. At the bottom right of the window is a blue button labeled 'Regresar'.

Regresamos al menú e ingresando el número 4 el programa nos manda un mensaje y posteriormente se cerrará y terminará



Con esto ya tenemos las 4 acciones del programa realizadas

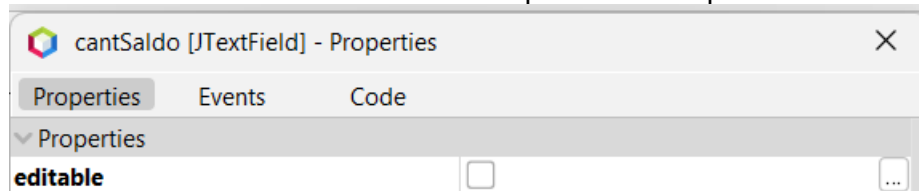
Codificación

Antes de iniciar con la codificación del retiro y la salida me gustaría agregar que realice una corrección a mi programa añadiendo una validación para la interfaz retiro que se asegura que el cliente cuente con la cantidad de dinero suficiente para retirar, de lo contrario se mostrara un mensaje indicando que el cliente no cuenta con el saldo suficiente

```
//Condición para verificar que no pueda retirar mas dinero del que tiene
if (retiro > cantidad) {
    javax.swing.JOptionPane.showMessageDialog(null, "No cuentas con el saldo suficiente para realizar esta acción");
} else {
    cantidad -= retiro;
    VariablesGlobales.setCantidad(cantidad);
    javax.swing.JOptionPane.showMessageDialog(null, "Retiro realizado correctamente");
    dispose();
}
```

Ya corregido este problema ahora si podemos pasar a la interfaz de consulta de saldo.

Antes de codificar debemos desmarcar la propiedad "editable" de nuestra textbox, esto con el fin de solo mostrar el saldo del cliente sin que el usuario pueda realizar ningún tipo de modificación



En una variable del tipo double llamada cantidad obtendremos el valor de nuestra variable global del mismo nombre, una vez obtenida la convertiremos a string y con esto imprimiremos el mensaje que muestra el saldo del cliente en la textbox de la interfaz.

```
public ConsultaSaldo() {  
    initComponents();  
    //Obtenemos el valor de la variable cantidad que es el dinero con el que cuenta el usuario  
    double cantidad = VariablesGlobales.getCantidad();  
    String txtSaldo = Double.toString(cantidad);  
    cantSaldo.setText("Tu saldo es de: $" + txtSaldo);  
}
```

Para la opción de salir escribiremos en el case del archivo del menú, aquí solamente lanzaremos el mensaje del cierre de sesión y posteriormente cerraremos y terminaremos el programa

```
case 4:  
    // Código a ejecutar para el caso 4  
    javax.swing.JOptionPane.showMessageDialog(null, "Cerrando sesión, vuelva pronto");  
    dispose();  
    break;
```

Conclusión

La implementación de este programa utilizando el lenguaje de programación Java y el entorno de desarrollo NetBeans permite aprovechar las características y capacidades de estas herramientas para crear una interfaz intuitiva y fácil de usar. Mediante la utilización de una interfaz gráfica de usuario, los clientes podrán acceder a las diferentes opciones del menú de manera clara y sencilla. La funcionalidad de depósito, retiro y consulta de saldo proporciona a los clientes la flexibilidad y el control sobre sus finanzas, permitiéndoles administrar sus cuentas de manera eficaz. Además, la inclusión de la opción "Salir" brinda la posibilidad de cerrar la sesión de manera segura, garantizando la privacidad y seguridad de la información del cliente.

En conclusión, la implementación de este programa utilizando Java y NetBeans es una solución efectiva para satisfacer las necesidades de los clientes de Banco Mexicano. Proporciona una experiencia de usuario agradable y funcionalidades que facilitan la gestión de las cuentas bancarias. Esta solución contribuye al campo laboral y a la vida cotidiana, ya que promueve la digitalización de los servicios financieros y brinda una mayor comodidad y accesibilidad para los clientes.

GitHub

https://github.com/JusiLinGu/Practicas_UMI