**DOCUMENTACION EJERCICIOS I & II**

Edgar Núñez Arana

**Tabla de contenido**

[**EJERCICIO I** 2](#_Toc168004816)

[**Modelo (*M)*** 2](#_Toc168004817)

[**Controlador (C)** 3](#_Toc168004818)

[**Vista (V)** 6](#_Toc168004819)

[**Conclusión** 7](#_Toc168004820)

[**EJERCICIO II** 8](#_Toc168004821)

[**VISTA** 8](#_Toc168004822)

[**Ventana Principal** 8](#_Toc168004823)

[**Ventana Crear Cuenta** 9](#_Toc168004824)

[**Ventana Ver Cuenta** 10](#_Toc168004825)

[**Ventana Actualizar Saldo** 11](#_Toc168004826)

[**Ventana Borrar Cuenta** 12](#_Toc168004827)

[**Guardar Cuenta** 13](#_Toc168004828)

[**MODELO** 14](#_Toc168004829)

[**CONTROLADOR** 15](#_Toc168004830)

[**CONCLUSION** 15](#_Toc168004831)

# **EJERCICIO I**

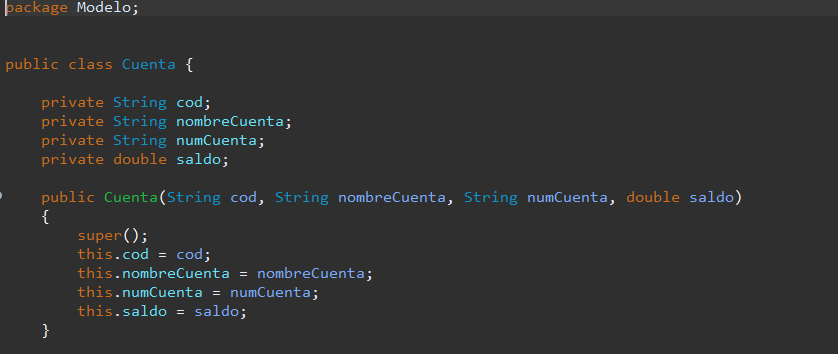
**En este ejercicio haremos una gestión de un banco desde Eclipse donde podremos crear cuentas y actuar sobre ellas, ver sus datos, borrarlas, o modificarlas para ello las reconoceremos con un ID que será el código identificativo de cada cuenta, todo ello estará en el formato *MVC* para el correcto funcionamiento y administración de cada clase del banco.**

## **Modelo (*M******)***

El apartado modelo (M) es el package, como se puede ver en la imagen 1, interno de nuestro banco el cual tiene la clase cuenta.java y es donde se guardara nuestras cuentas en formato .dat.

IMAGEN 1

Además la clase Cuenta tendrá unos atributos que serán los datos a ver de cada cuenta como puede ser Nombre del titular de la cuenta, numero de cuenta, saldo (tipo double) y por último el código identificativo que será el mas importante ya que con el podremos reconocer a que cuenta nos dirigimos. En esta clase Cuenta también tendremos su constructor y los “getter” y “setters”, véase en la imagen 2 (hay algunos getter y setter no todos).



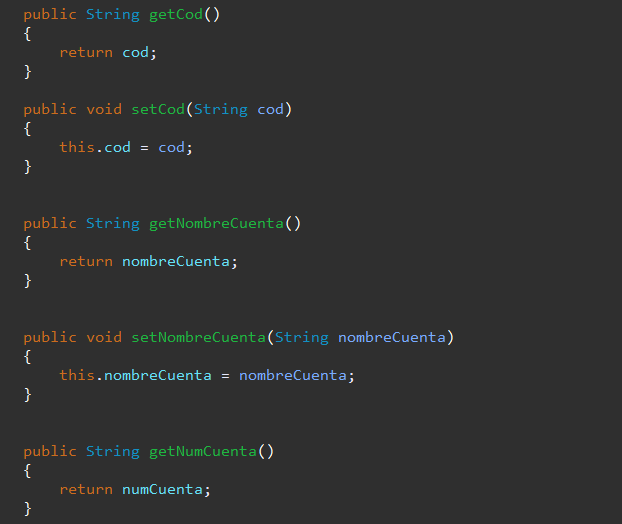


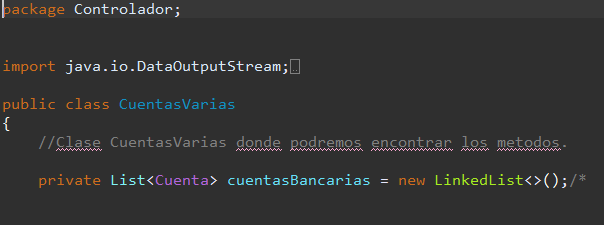
IMAGEN 2

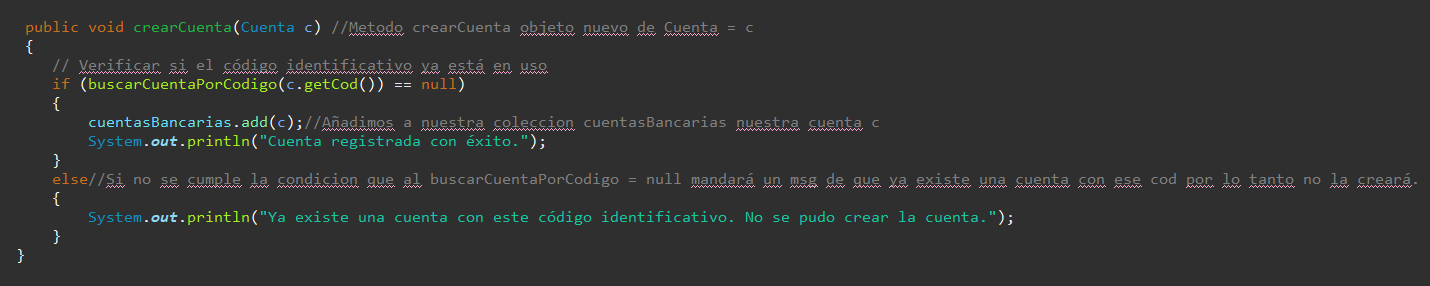
## **Controlador (C)**

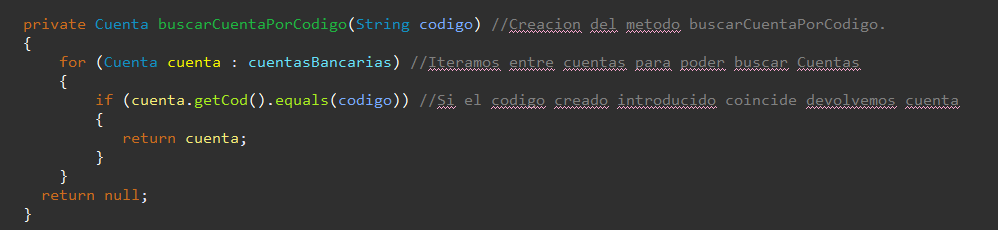
**En el apartado controlador (C) es el package, imagen 3, el cual es el encargado de dar las distintas funciones de nuestro banco**.

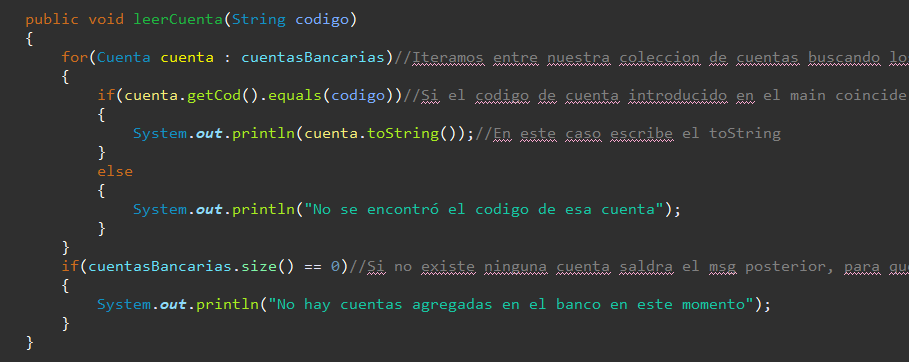
IMAGEN 3

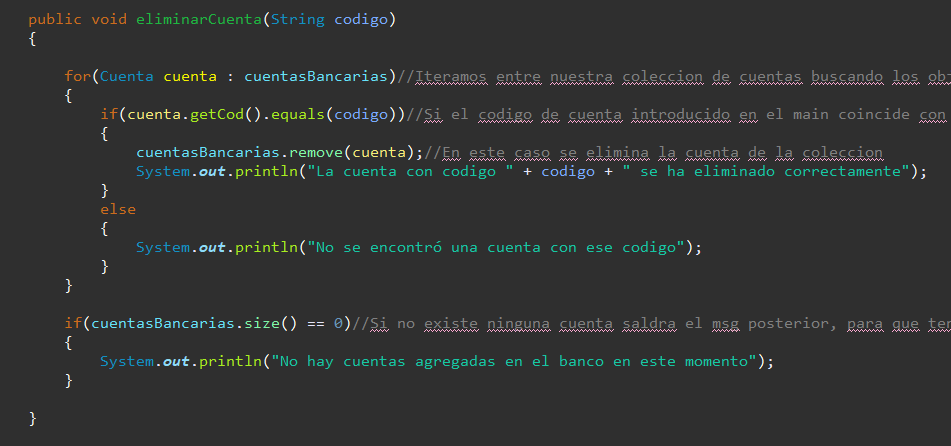
**Para ello crearemos una colección para poder actuar sobre ella.**

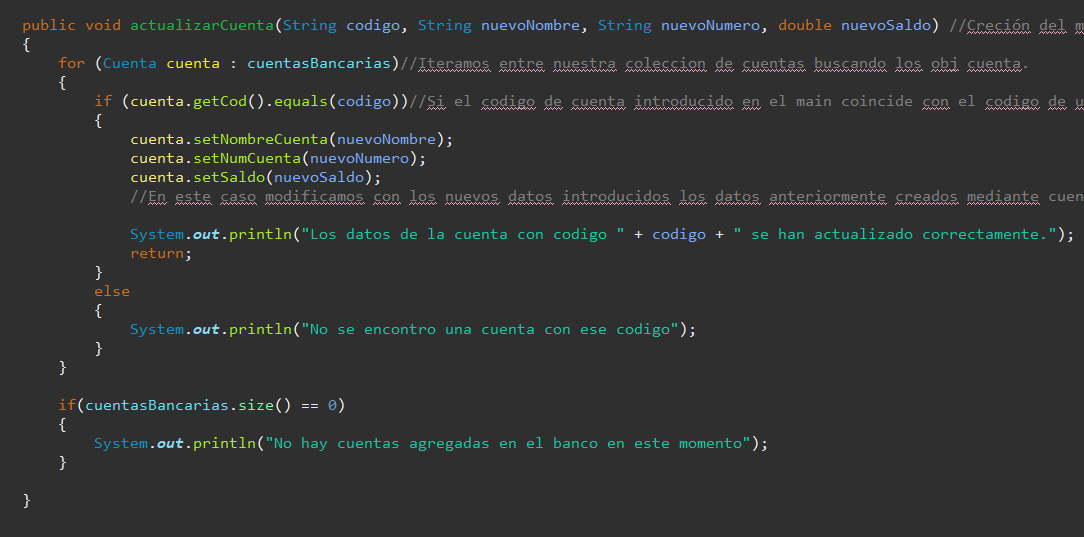












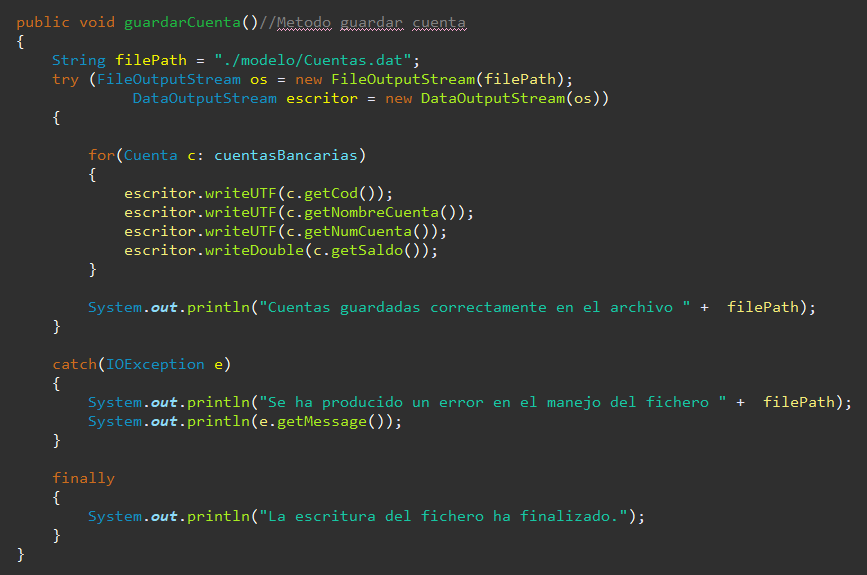
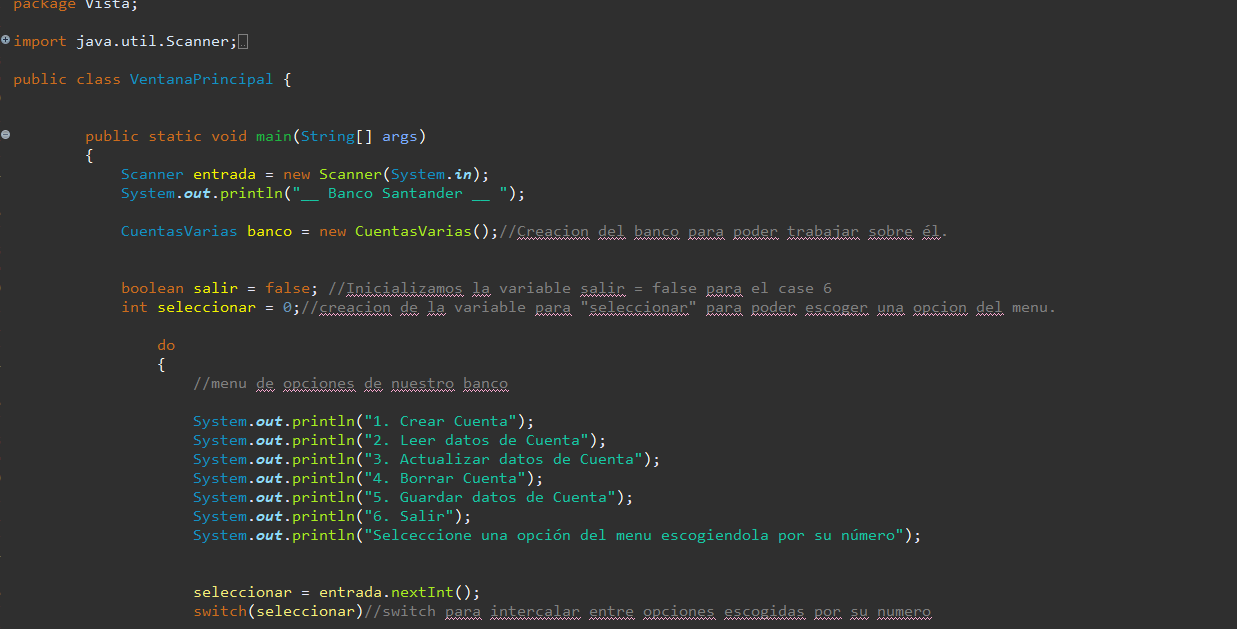


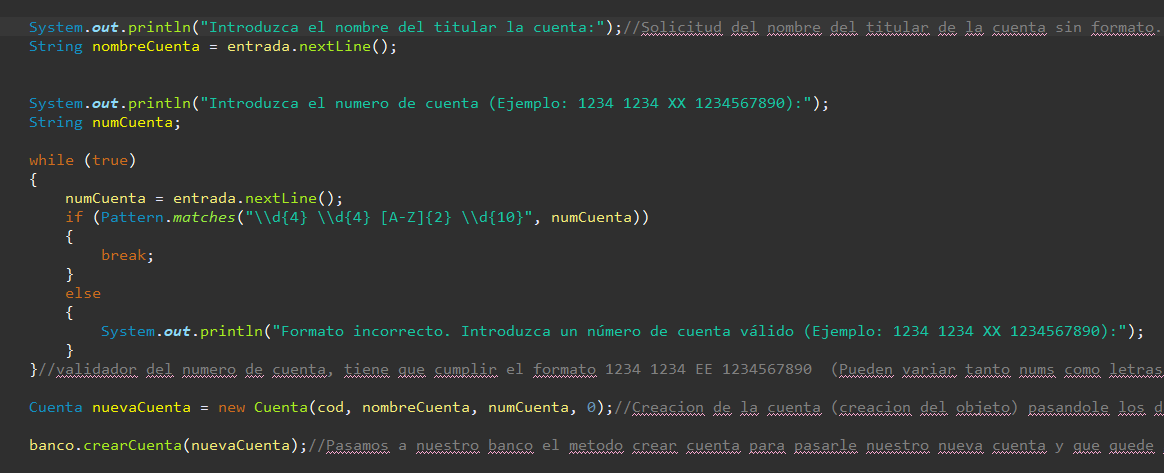
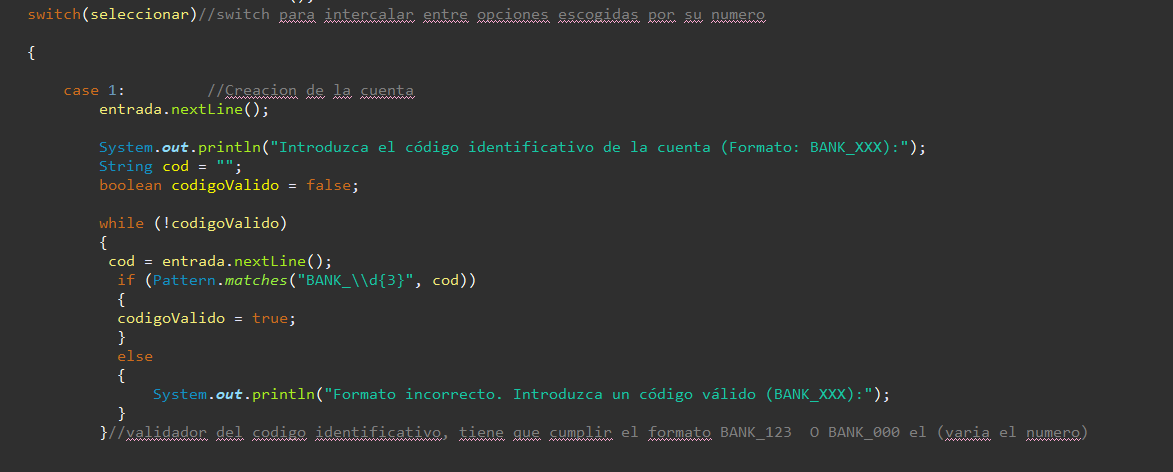
IMAGEN 4

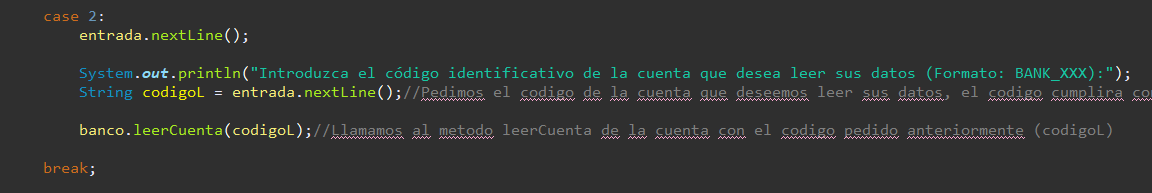
## **Vista (V)**

Por ultimo en el apartado vista, imagen 5,será el encargado de darle el feedback con el cliente donde este último podrá tener acciones con el programa escoger las opciones en el menú y crear borrar modificar cuentas…

Para ello crearemos un banco donde podremos realizar las acciones.







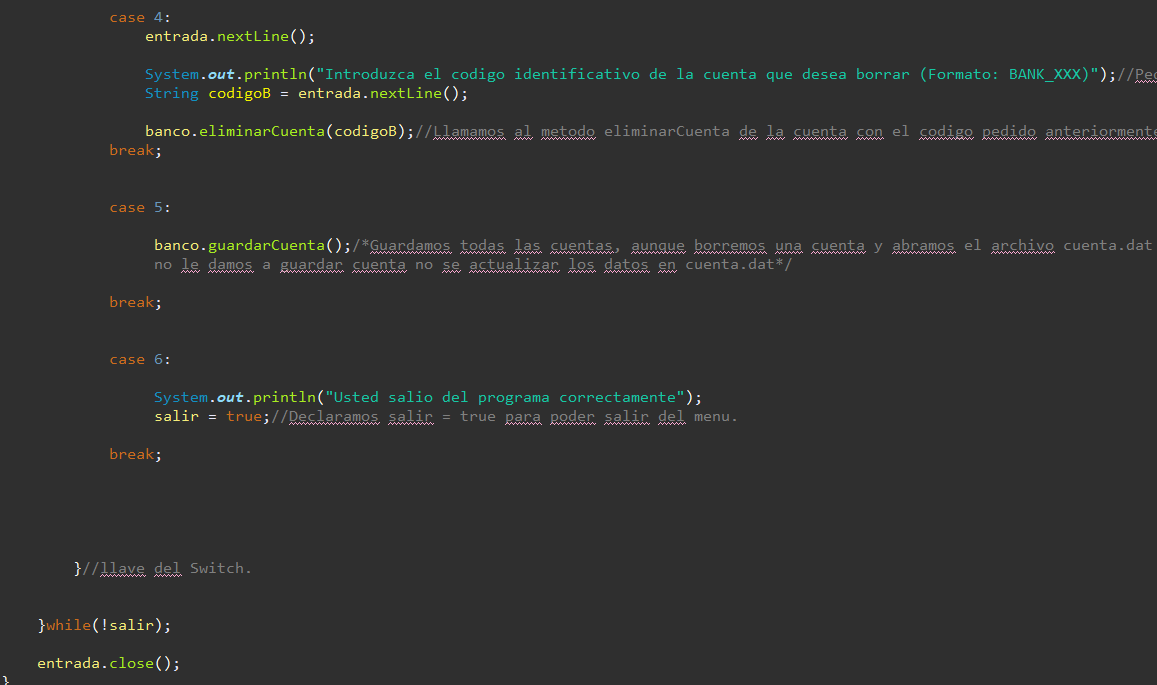
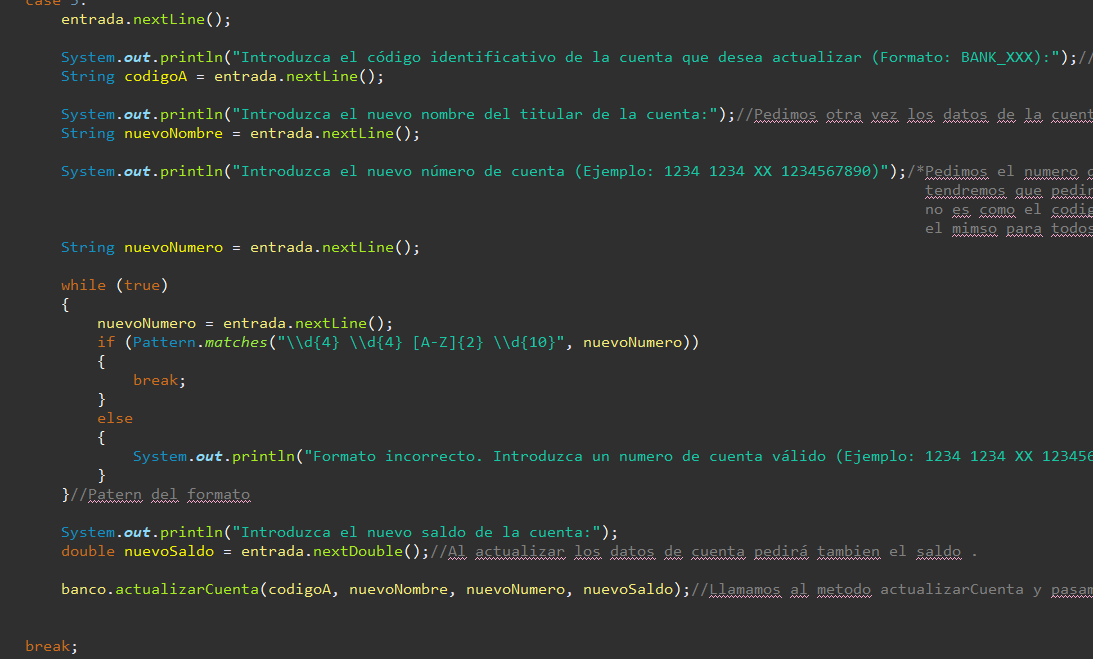


IMAGEN 5

Se puede apreciar que cada case será una opción que se podrá escoger mediante el menú.

## **Conclusión**

En conclusión, con esta organización MVC podremos gestionar de una manera óptima y organizada las clases para su correcto funcionamiento de banco.

# **EJERCICIO II**

**En este ejercicio haremos una gestión de un banco desde NetBeans el funcionamiento es parecido al del ejercicio I pero aquí podremos ver la interfaz gráfica de cómo se vería la aplicación de nuestro banco.**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**

# **VISTA**

**Para empezar con el modelo vista tendremos las distintas ventanas que interactuara el usuario siendo la main la Ventana Principal:**

## **Ventana Principal**

Interfaz de usuario gráfica

Descripción generada automáticamenteEsta venta será un archivo JFrame y será la encargada de proporcionar un menú al usuario donde este podrá escoger la opción que desee.

## **Ventana Crear Cuenta**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**En la ventana crear cuenta (JDialog) nos dará la funcionalidad de crear cuenta donde escribiremos su ID nombre número de cuenta y saldo los cuales se guardarán en un nuevo objeto cuenta.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteSi la cuenta se ha creado correctamente mandara un mensaje de se ha creado la cuenta con éxito.

## **Ventana Ver Cuenta**

En este caso tendremos la ventana ver cuenta (JDialog) la cual tendremos que introducir un ID de cuenta previamente creado para que nos muestre sus datos.

**Interfaz de usuario gráfica

Descripción generada automáticamente**

Como se ve a continuación si introducimos el ID de la cuenta creada en el apartado anterior nos mostrara los datos de dicha cuenta.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

Si se introduce un ID de cuenta que aun no hayamos añadido antes saltará el mensaje de que no se ha podido encontrar dicha cuenta.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

## **Ventana Actualizar Saldo**

En la ventana actualizar saldo (jDialog) la función será cambiar los datos de cada cuenta, primero introduciremos el id de la cuenta para poder escoger que cuenta modificar.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

Si la cuenta existe nos mostrará una nueva ventana donde podremos cambiar los datos en este caso modificar el nombre.

**Interfaz de usuario gráfica

Descripción generada automáticamente**

Si la cuenta con ese ID no existe saldra el mensaje de que no existe dicha cuenta.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

# **Ventana Borrar Cuenta**

Ahora en la ventana borrar cuenta podremos borrar la cuenta de nuestra lista cuentas mediante su id.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

Si la cuenta con el id existe se borrará de la lista cuentas y nos mostrará el siguiente mensaje.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

Si la cuenta con ese ID no existiera mostrara el mensaje de que no existe cuenta bancaria con ese código identificativo.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Guardar Cuenta**

Aunque la opción guardar cuenta no se encuentre como otro JDialog en una ventana especifica, está implementada en Ventana Principal la cual si está en Vista (MVC). Esta función será la encargada de guardar la cuenta en un directorio especifico de nuestro equipo.

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

# **MODELO**

**En el package modelo nos encontraremos con la clase cuenta.java que es nuestro objeto principal el cual es usado en todo el programa.**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

# **CONTROLADOR**

**En este caso nuestro package controlador estará vacio ya que decidí darle toda la funcionalidad dentro de las propias ventanas de VISTA.**

# **CONCLUSION**

**Habiendo terminado el proyecto banco MVC desde “NetBeans” he concluido que podemos hacer la aplicación del banco con sus requisitos añadiéndole una interfaz grafica con la que el usuario le será más fácil e intuitivo adaptarse y poder ejecutar nuestro programa con sus distintas funciones.**