

CAjero automático(ATM)

Programación Orientada a Objetos



Entrega: 30 de noviembre de 2017

ITESM CEM

Nombres: {Erick Bautista Pérez A01379896 & Héctor David Hernández Rodríguez A01374009}

Profesor: Julio Guillermo Arriaga Blumenkron

**Índice**

**Introducción . . . . . . . . . 2**

**Desarrollo . . . . . . . . . . 2**

**Driagrama UML . . . . . . . . . 6**

**Manual de usuario . . . . . . . . 6**

**Conclusiones . . . . . . . . . 12**

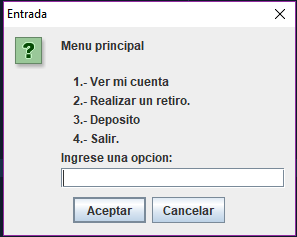
**Fuente de información . . . . . . . 13**

**Introducción**

Durante la realización de este proyecto se darán a conocer los distintos métodos usados en clase como son las de herencias, clases abstractas, polimorfismo, manejo de excepciones, manejo de eventos, arreglos, entre otros. Además, cabe mencionar que nuestro proyecto se fue enfocando a él solo uso de Diálogos/Mensajes que nos ofrecía Swing. Gracias a este proyecto podremos referenciarnos a hacer el uso del conocimiento adquirido en el curso de Programación Orientada a Objetos donde realizamos los pasos de un objeto orientado utilizando Diagramas UML con nuestro proyecto del cajero automático.

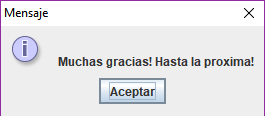
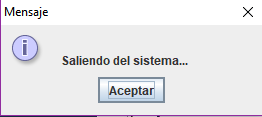
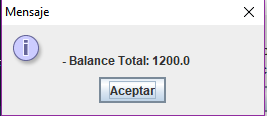
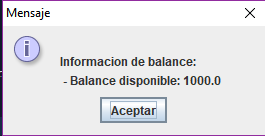
**Desarrollo**

Nuestro proyecto es un programa que realiza la simulación de un cajero automático y consiste en que al inicio tendrás que escribir un usuario y pin por medio de números, nosotros por default solo dejamos una cuenta agregada al programa que es Usuario: 12345 y PIN: 54321, si el programa no tiene un Usuario o PIN alojado en nuestro Array te lo invalidara y no se terminara el ciclo, después de haber tenido una entrada correcta refiriéndome al lado de validación de usuario se arrojara un menú con las diferentes opciones:

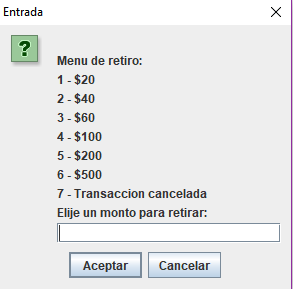


El programa posee 11 clases: ATM, BalanceAdquisicion, BankDataBase, Cuenta, Deposito, DepositeSlot, DispensadorEfectivo, Retiro, Screen, Transacción y la Clase ATMTest (que es donde se ejecuta todo nuestro proyecto) las cuales algunas de ellas heredan de la clase padre ATM. Tiene una clase abstracta la cual es Transacción que hereda de Deposito, Retiro y BalanceAdquisicion. Además, contiene una Interface que esta implementada en la clase principal que es ATM, como antes lo había mencionado empezamos a desarrollar este código haciéndonos referencia a una corrida en consola para después implementarle la parte de la interfaz.

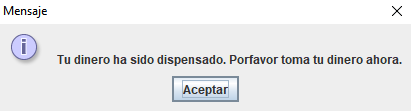
1.- Ver mi cuenta: Para la parte del menú que se desplego al momento de validar el usuario se mostraran 4 opciones, para la primera opción lo que arroja es que al momento de seleccionarla te arrojara un mensaje diciéndote el balance disponible con el que dispone el cajero, después un balance total que es el valor total de tu cuenta y consecuentemente el cajero terminara el proceso y mostrara un mensaje diciendo “Saliendo del sistema…”.

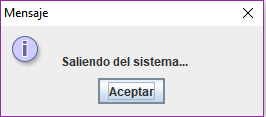


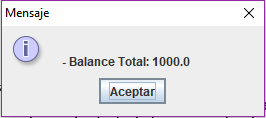
2.- Realizar un retiro: Para la segunda parte que nos referimos a como despliega un retiro se muestran las siguientes opciones:



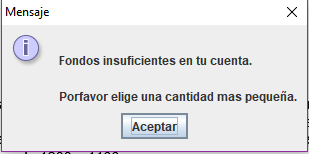
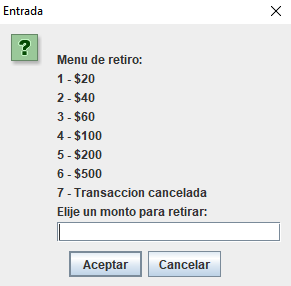
Donde el usuario sabrá cuanto dinero quiere sacar del cajero por así decirlo, y habrá tres opciones por default la primera es que si no quiere realizar un retiro escriba el numero 7 y consecuentemente el programa saldrá del menú de retiro y mostrara el mensaje de “Saliendo del sistema” regresándose otra vez a la validación del usuario, y para la segunda opción es que realice un retiro como se puede mostrar a continuación



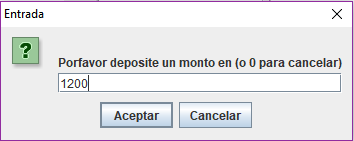
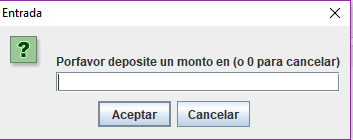


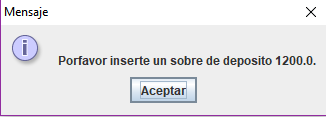
Y si se vuelve a realizar la opción de Ver tu cuenta se mostrará al usuario que en realidad si hizo un retiro de n cantidad de dinero en este caso se selecciono retirar 100 y como se puede mostrar en la siguiente imagen cual es la diferencia de su balance total que era de 1200 a 1100.

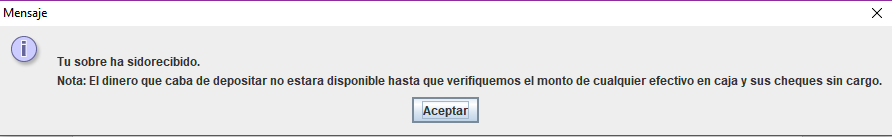
Y para la tercera opción me referiré a lo que puede pasar cuando el usuario no tenga fondos suficientes para hacer un retiro de una cantidad mayor a la que la cuenta tiene por default el programa arrojara un mensaje diciendo que los fondos son insuficientes y que elijar una opción de retiro menor a la que había realizado.



3.- Deposito: Para la parte de la clase Deposito el usuario debe de escribir una cantidad en números enteros donde el programa por definición lo convertirá a un valor de double como se puede ver a continuación.

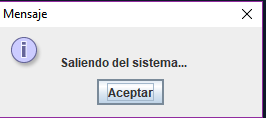


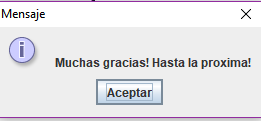


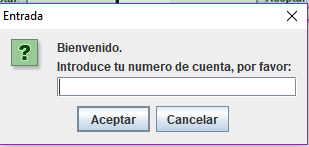


El programa te mostrara un mensaje de que debes de insertar un sobre de deposito de n cantidad y después de esto arrojara un mensaje de que el sobre ha sido recibido. Si se realiza no hacer un deposito por default el programa te dirá de nuevo que estará saliendo del sistema y te devolverá a la validación del usuario.

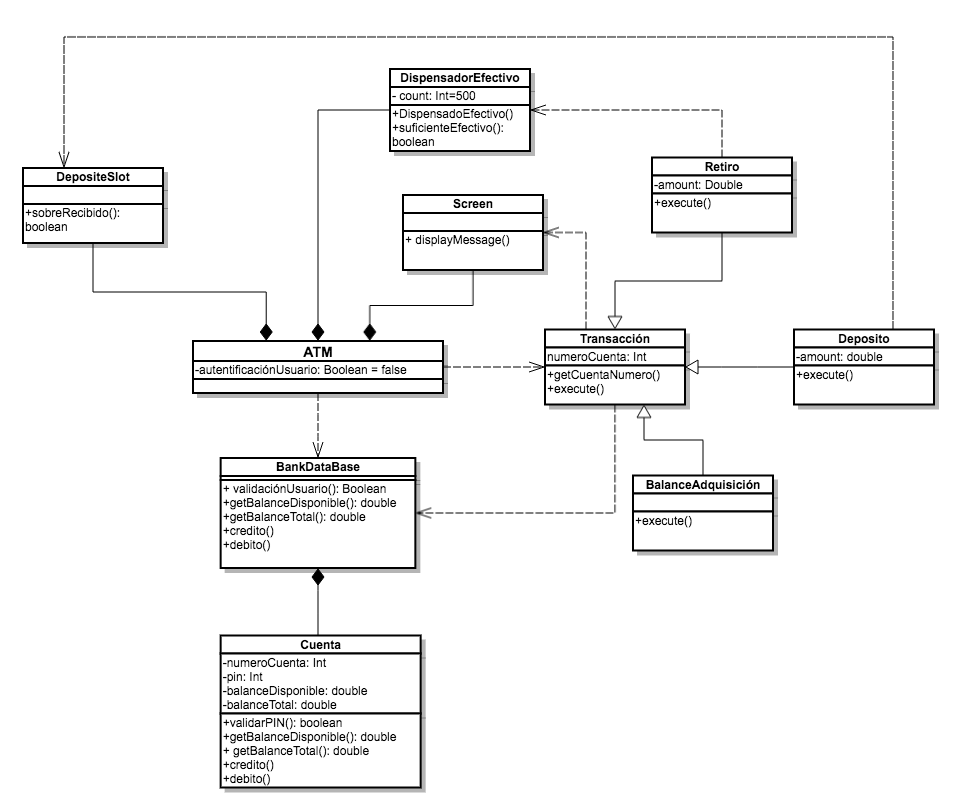
4.- Salir: Para este caso después de haberse validado como un usuario y seleccionar la opción de salir se estará refiriendo de salir de su cuenta como usuario, esto no termina el proceso del cajero automático, ya que se dejo como ocaso que sea un ciclo donde nunca termina ya que nos referimos a los cajeros automáticos verdaderos y prácticamente terminaría pulsando un clic en el cierre de la ventana.



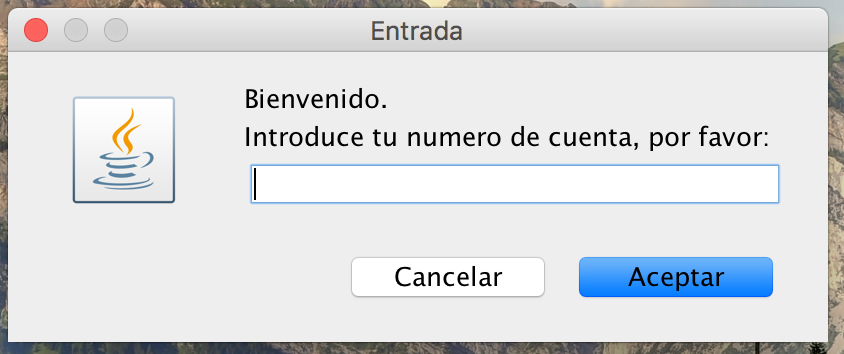


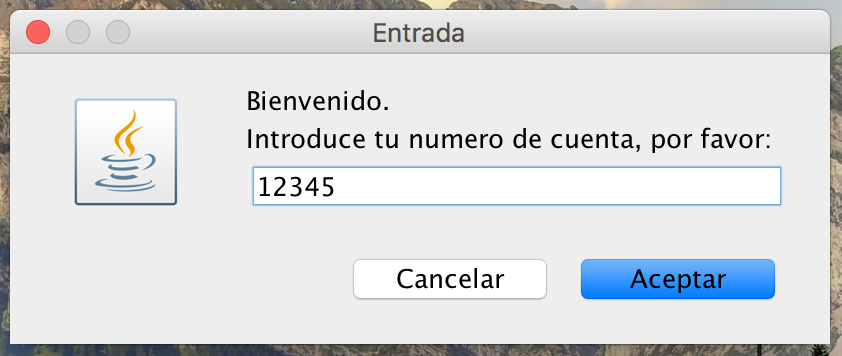


**Diagrama UML**

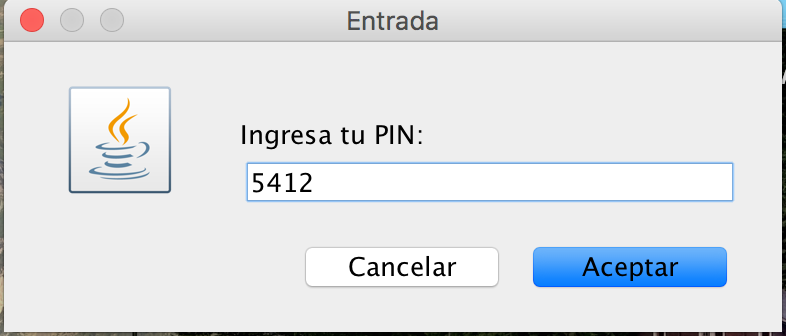
****

**Manual de usuario**

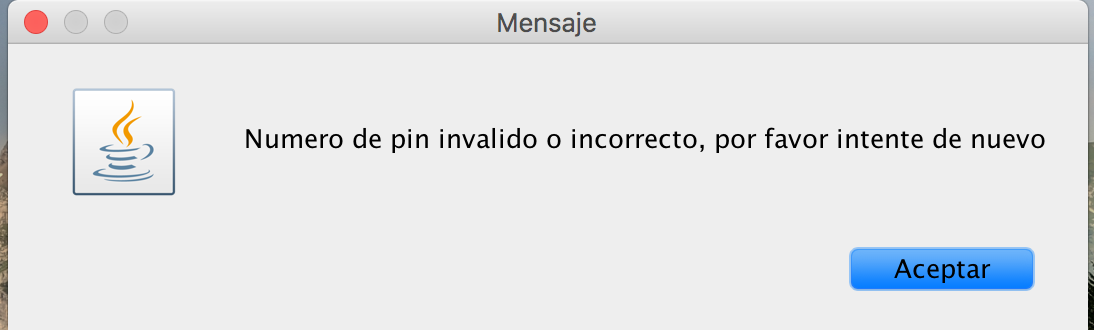
**1.-Introducir un numero de cuenta, para el testeo de esste solo dimos de alta dos numero de cuenta : 12345 y 98765**

****

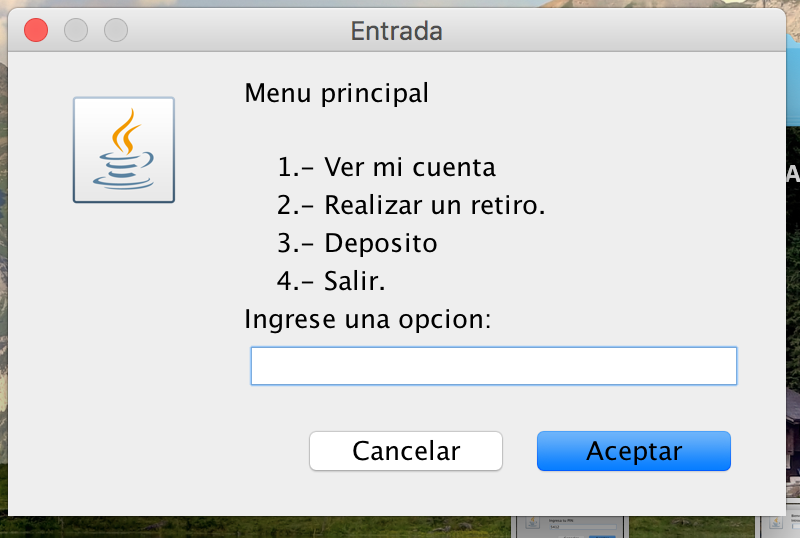
**2.- Tendras que validat tu entrada con un pin que para este testeo tenemos, 54321 y 56789**

****

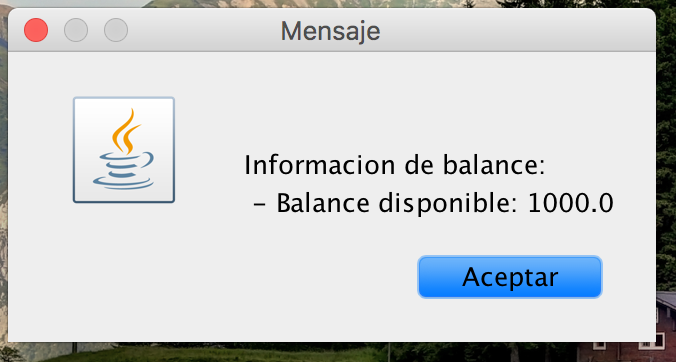
**2.1 De ser erroneo, deberas tendras un numeor incorrecto y tendras que regresar al inicio.**

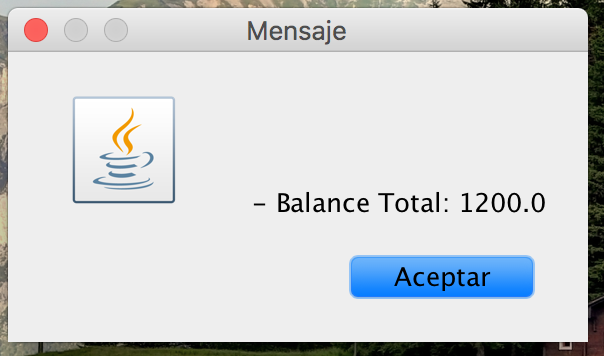
****

**3.- EL menu principal nos llevara a nuestra demas funciones.**

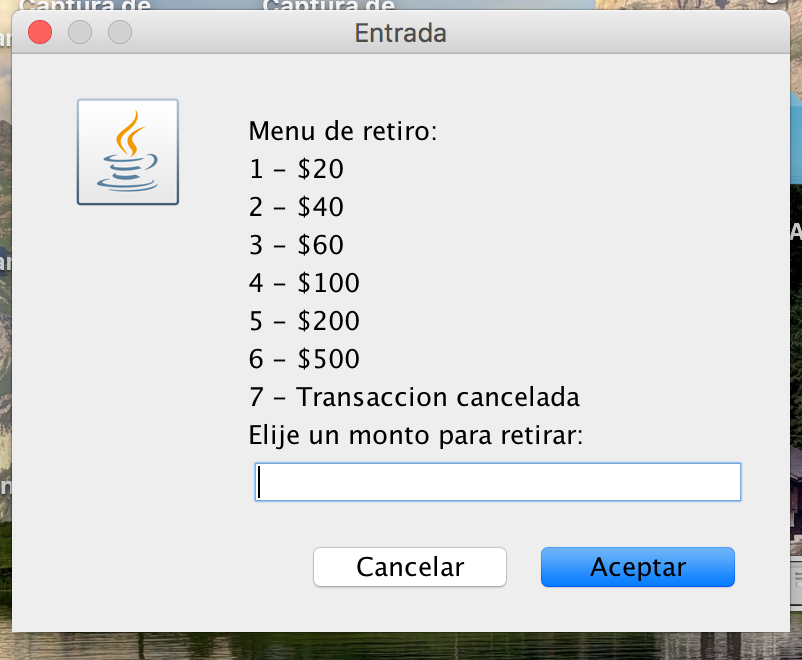
****

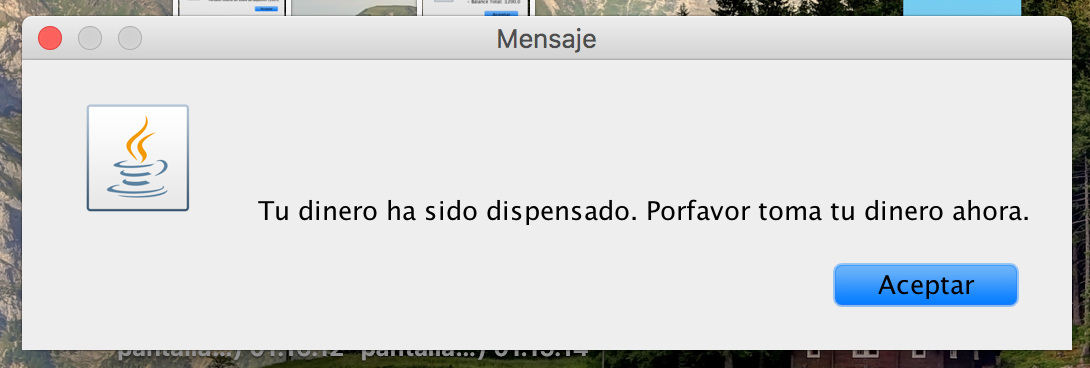
**4.- La opcion de ver mi cuenta me desplegara un balance total establecido al por el “BANCO/Administrador” y nos desplegara un balance total con el total de los depositos establecidos.**

****

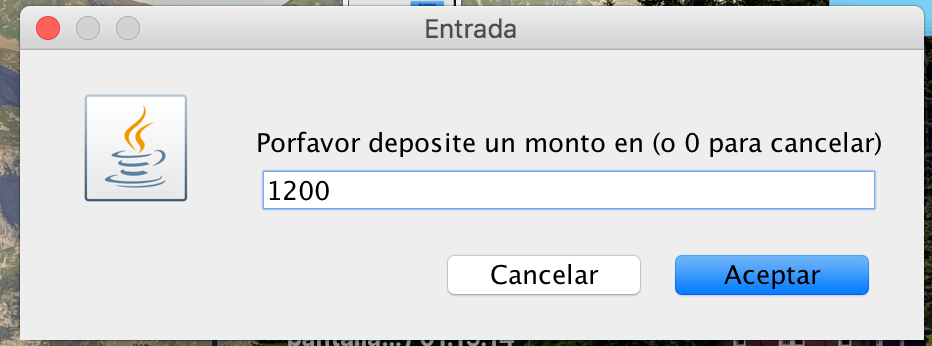
****

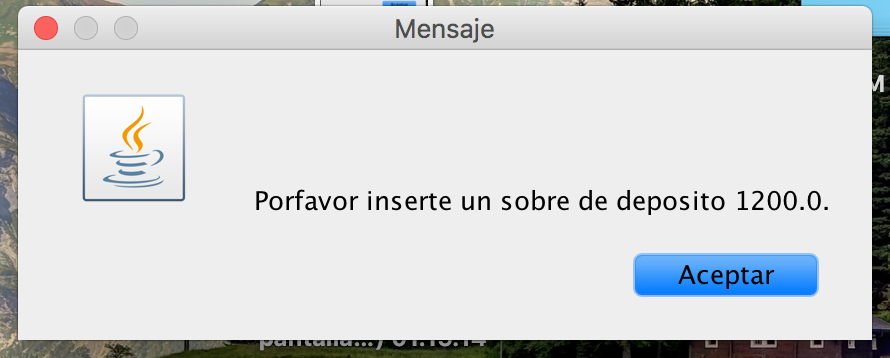
**5.- La opcion de realizar un retiro nos muestra un meno que resta a nuestro balance total los gastos aquí realizados.**

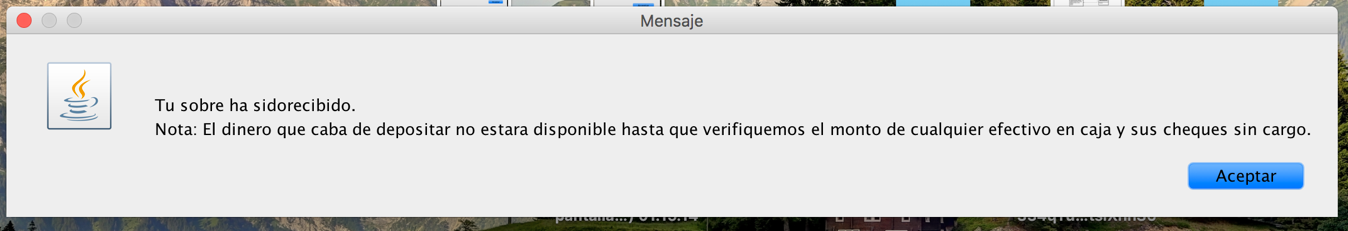
****

****

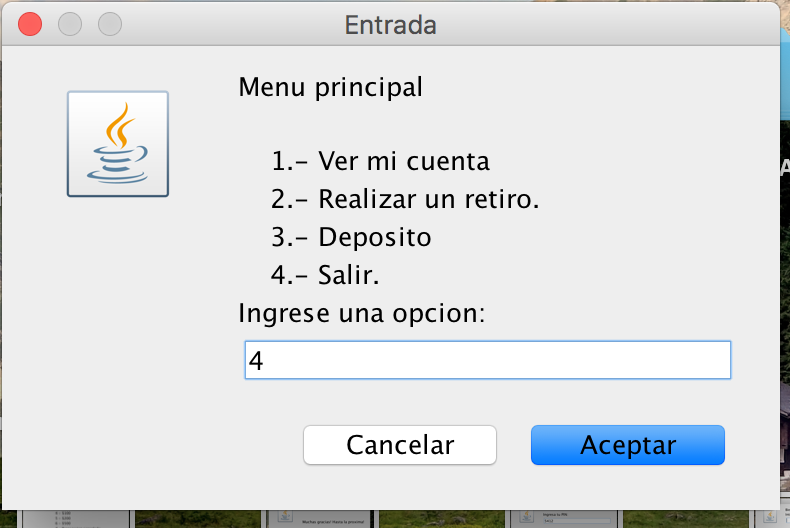
**6.- La opcion de Deposito nos pedira el monto a depositar en la cuenta, y nos lo sumara a nuestra cuenta en el balanceTotal.**

****

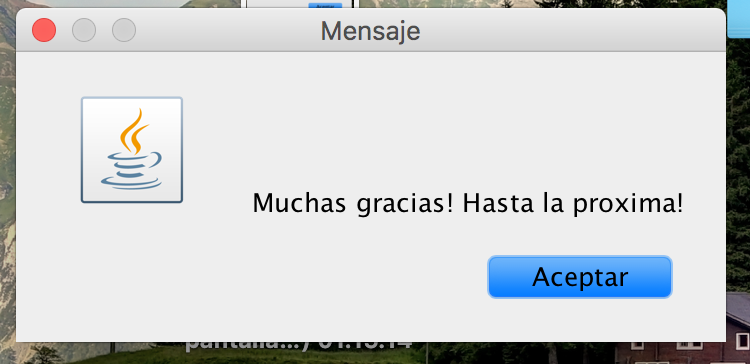
****

****

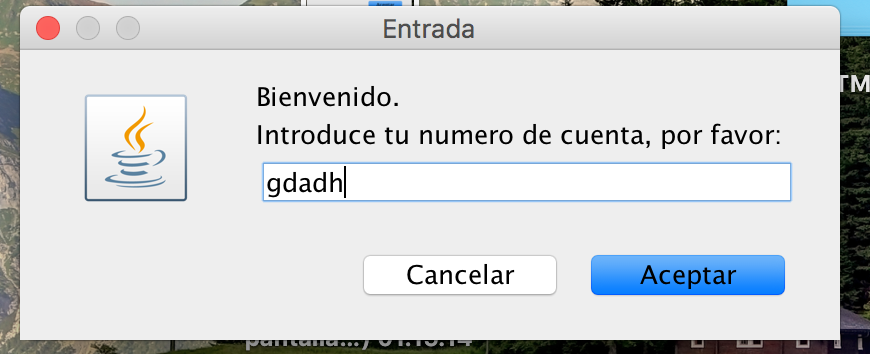
**7.- La opcion salir, nos cerrara nuestra sesion en el cajero.**

****

****

****

**#.- Todos los campos estan protegidos ante errores de dedo.**

****

**Conclusiones individuales**

**Héctor:**  La elobracion de este trabajo fue muy dinamica y siento que las experencias que tuvimos al investigar, nos ayudo q eu pudieramos buscar como equipo las mejores soluciones que cumplieran con los requisistos de esta ultima entrega, la implementacion de clases y su relaicon entre ellas fue algo muy dinamico de este proyecto, auqnue siento que pudimos implementar mas cosas en la intrefaz de Swing, siento que por mas que investigamos, no encontramos una mejor manera de hacerlo, este proyecto fue un gran repaso del curso, y batallamos en ciertos momentos de este, pero logramos sobrepasarlos con ayuda de las actividades y de ciertas documentacions que encontramos. Un protecto que siento que aunque cumplio con los requisitos vistos en clase, me deja con ganas de mejorar algo de esto en el futuro.

**Erick**: En este semestre haciéndome referencia al curso de programación orientada a objetos pude observar que he repasado algunos temas que ya había visto anteriormente en vacaciones y los cuales me faltaba practicar más, con este proyecto acerca de como implementar clases y como relacionarlas cada una de ellas haciéndolas abstractas o relacionarlas con herencias se me hicieron más fácil de crearlas, pero el momento de crear se me hizo muy difícil ya que desafortunadamente nosotros creamos al principio un Diagrama de UML muy escaso pero con la parte de la practica de estos mismos al final se pudo finalizar con un buen proyecto, solo que me hubiese gustado ver mejor las interfaces graficas ya que se me complicaron implementarlas en mi proyecto final.

**Fuentes de información**

[**https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html**](https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html)