



**INSTITUTO POLITECNICO NACIONAL**

ESCUELA SUPERIOR DE CÓMPUTO

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Cajero automatico**

Práctica No. 1

**Alumno:**

Nava Hernandez Jeovanny Martin

**Boleta:** 2010630345

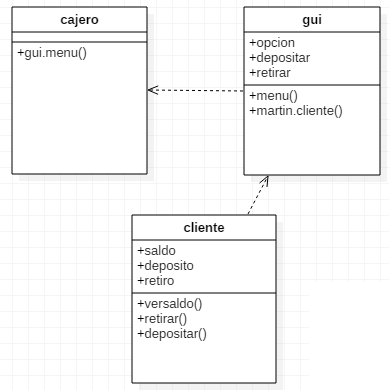
**Grupo:** 2CM7

**Hora límite:** Clase de lab.

**Descripción del problema general:**

Crear un cajero automático que pueda retirar, depositar y ver saldo.

**Diseño:**



**Codificación:**

cliente.java

1. public class cliente{
2. String nombre=null;
3. int saldo;
4. int deposito;
5. int retiro;
6. public cliente(){
7. nombre="Martin nava";
8. saldo=1500;
9. deposito=0;
10. retiro=0;
11. }
12. public int retirar(int r){
13. retiro=r;
14. saldo=saldo-retiro;
15. return saldo;
16. }
17. public int depositar(int d){
18. deposito=d;
19. saldo=saldo+deposito;
20. return saldo;
21. }
22. public int versaldo(){
23. if(saldo<=999){
24. saldo=saldo - 35;
25. }
26. return saldo;
27. }
28. }

gui.java

1. import javax.swing.\*;
2. public class gui{
3. public static void menu(int o){
4. cliente martin=new cliente();
5. int a=2;
6. JOptionPane.showMessageDialog(null, "Bienvenido al cajero automatico");
7. do{
8. String op=JOptionPane.showInputDialog(null, "1.-Ver saldo\n2.-Depositar\n3.-Retirar\n4.-Salir");
9. int intop=Integer.parseInt(op);
10. int regresar=1;
11. switch(intop){
12. case 1:
13. int saldo=martin.versaldo();
14. JOptionPane.showMessageDialog(null,"Tu saldo es" + saldo);
15. break;
16. case 2:
17. String dep=JOptionPane.showInputDialog(null,"¿Cuanto quieres depositar?");
18. int intdep=Integer.parseInt(dep);
19. JOptionPane.showMessageDialog(null,"Depositaste:\n" + "$"+intdep);
20. martin.depositar(intdep);
21. martin.versaldo();
22. JOptionPane.showMessageDialog(null,"Tu nuevo saldo es de: \n" + "$"+martin.versaldo(), "hola", a );
23. break;
24. case 3:
25. String ret=JOptionPane.showInputDialog(null,"¿Cuanto quieres retirar?");
26. int intret=Integer.parseInt(ret);
27. JOptionPane.showMessageDialog(null, "Has retirado la cantidad de:\n\t" +"$"+ intret);
28. martin.retirar(intret);
29. JOptionPane.showMessageDialog(null,"Tu nuevo saldo es de: \n" + "$ "+martin.versaldo() );
30. break;
31. case 4:
32. default:
33. JOptionPane.showMessageDialog(null,"Hasta la proxima");
34. System.exit(0);
35. break;
36. }
37. }while(a==2);
38. }
39. }

cajero.java

1. public class cajero{
2. public static void main(String args[])
3. {
4. cliente martin=new cliente();
5. int o=0;
6. gui.menu(o);
7. }
8. }

**Preguntas:**

1. ¿En que paquete se encuentra la clase JOptionPane?

javax.swing

1. Mediante el uso de JOptionPane, ¿cómo se crea un diálogo de mensaje?

JOptionPane.showMessageDialog(null, “Mensaje”);

1. ¿Cómo se puede modificar el icono del tipo del diálogo de mensaje?, ¿cuáles son?

JOptionPane.showMessageDialog(frame,"Mensaje","Titulo", JOptionPane.WARNING\_MESSAGE, new ImageIcon("./data/ima1.png"));

1. Mediante el uso de JOptionPane, ¿cómo se crea un diálogo de entrada en el que el usuario tiene que teclear un valor?, ¿de qué tipo es la entrada que el diálogo regresa?, ¿cómo se realiza la conversión del tipo de la entrada de ese diálogo a flotante?

String valor=JOptionPane.showInputDialog(null,”Introduce valor”);

Regresa un String

float val=Float.parsefloat(valor);

1. Mediante el uso de JOptionPane, ¿cómo se crea un diálogo de entrada en el que el usuario tiene que seleccionar un valor de una lista?, ¿de qué tipo es la entrada que el diálogo regresa?

JOptionPane.InputDialog(null,”Valor1\nValor2\n…\nValorn”);

Regresa un String y si se quiere otro tipo de dato hay que hacer la conversión mediante los métodos de las clases envolventes.

**Conclusión**

Mediante el uso de cajas de dialogo podemos hacer la interfaz mucho más entendible y agradable, el único problema es que es necesario hacer una transformación de datos.

