

MANUAL TECNICO

Tabla de Contenido

Introducción.....	3
Información destacada	3
Objetivos	3
1.Requerimientos	4
2.Lógica del programa	5

Introducción

El presente documento describe los aspectos técnicos informáticos del sistema de información. El documento familiariza al personal técnico especializado encargado de las actividades de mantenimiento, revisión, solución de problemas, instalación y configuración del sistema.

Información destacada

El manual técnico hace referencia a la información necesaria con el fin de orientar al personal en la concepción, planteamiento análisis programación e instalación del sistema. Es de notar que la redacción propia del manual técnico está orientada a personal con conocimientos en sistemas y tecnologías de información, conocimientos de programación avanzada sobre entorno web, administración de bases de datos, responsables del mantenimiento e instalación del sistema en los servidores.

Objetivos

Instruir el uso adecuado del Sistema de Información, para el acceso oportuno y adecuado en la instalación del mismo, así como la descripción de las clases utilizados para el desarrollo del sistema los cuales nos orienten en la configuración y soporte del mismo.

1.

Requerimientos

El sistema puede ser instalado en cualquier sistema operativo que cumpla con los siguientes requerimientos:

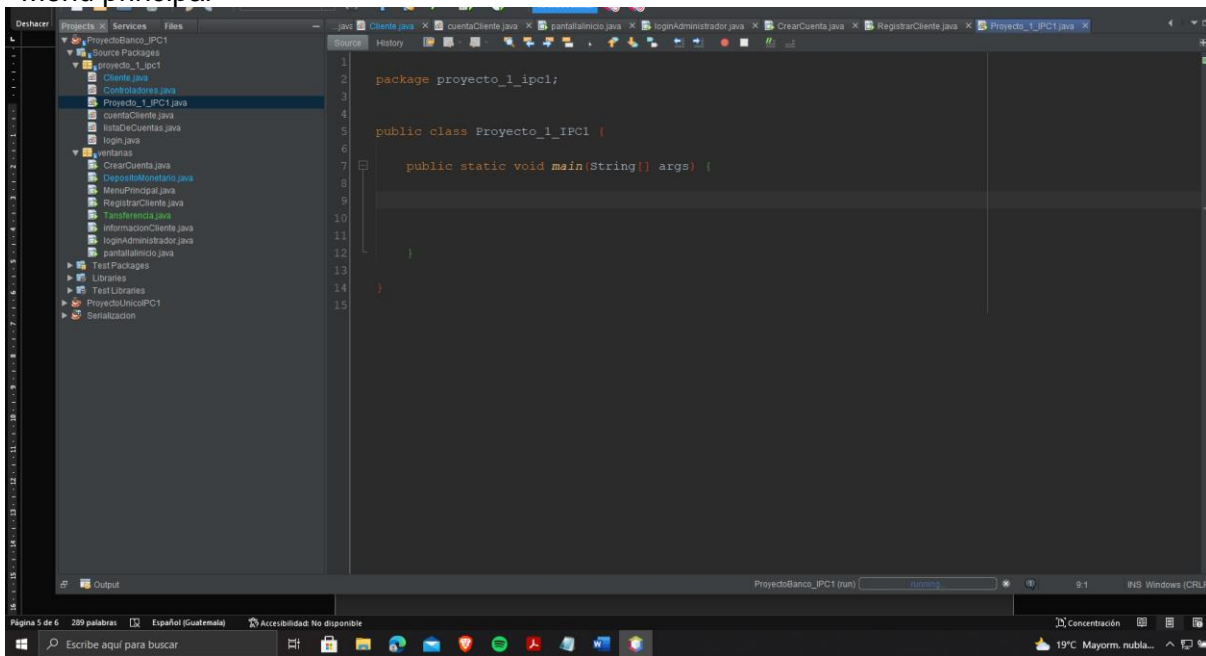
- **Windows xp en adelante**
- **Instalado el jdk**
- **Apache Netbeans**

Lógica del programa

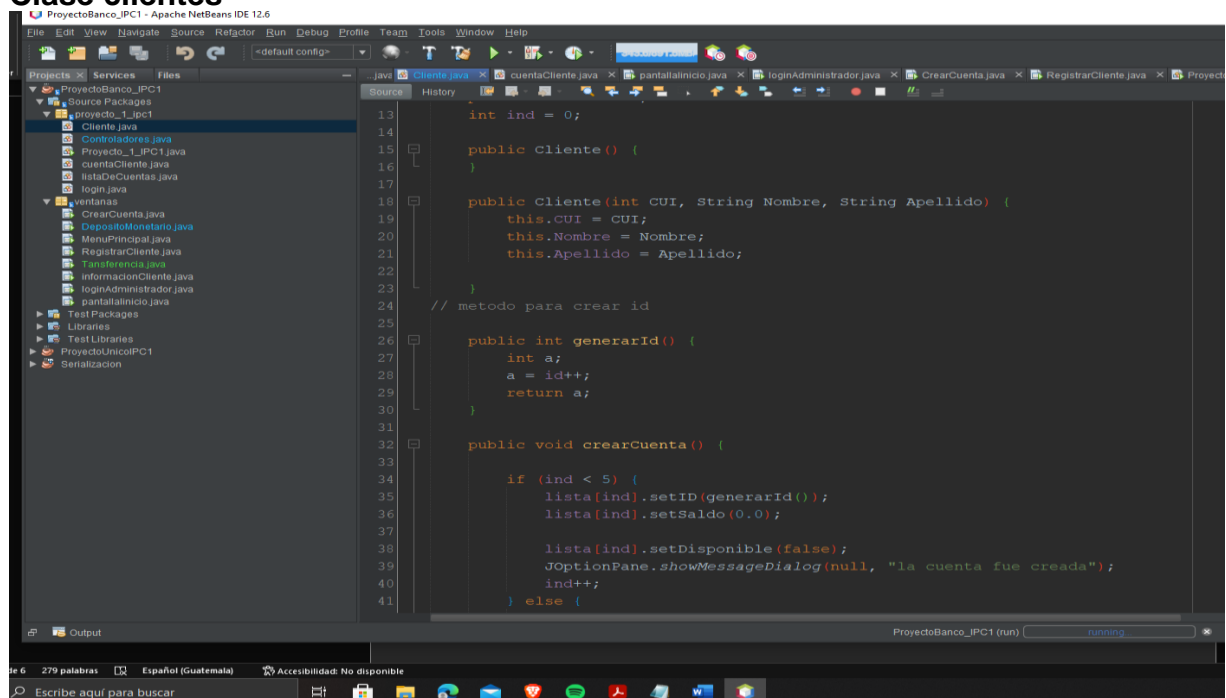
Este programa fue realizado en el lenguaje de Java, utilizando el IDE Apache NetBeans. Se utilizaron clases para efectuar las tareas que solicitaba el proyecto para que se viera ordenada la forma de programar.

En la clase principal donde se encuentra el main se inicializa la aplicación

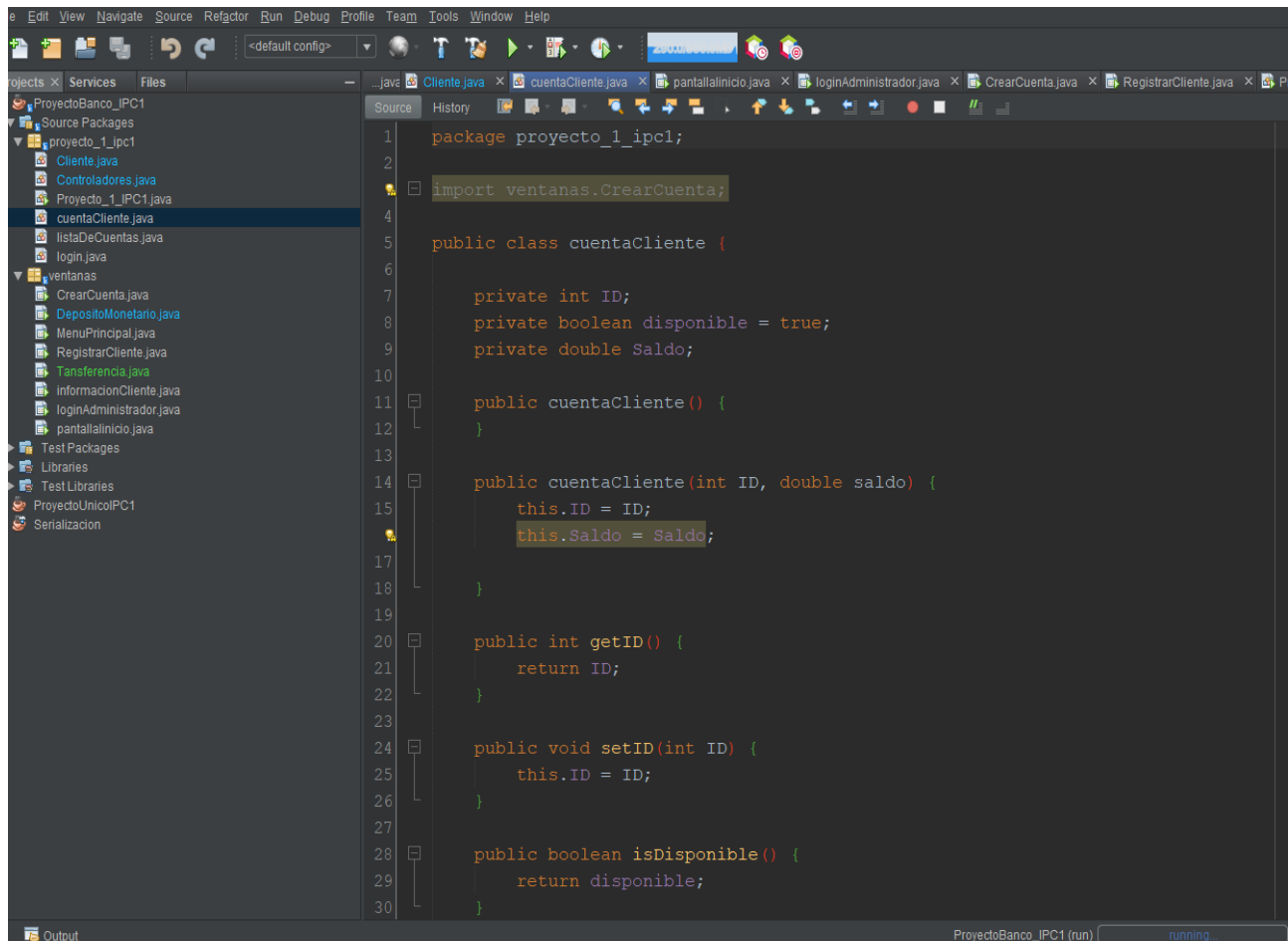
Menu principal



Clase clientes



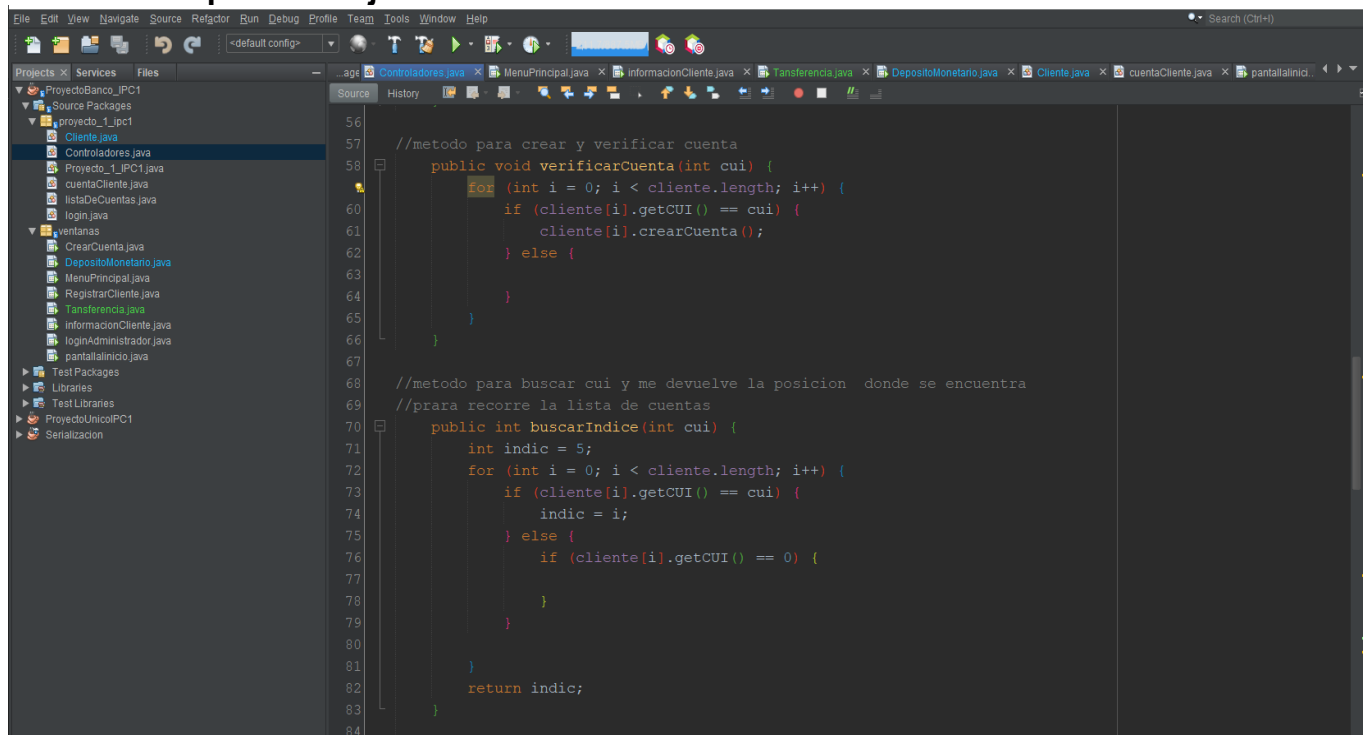
Clase cuenta



```
1 package proyecto_1_ipc1;
2
3 import ventanas.CrearCuenta;
4
5 public class cuentaCliente {
6
7     private int ID;
8     private boolean disponible = true;
9     private double Saldo;
10
11     public cuentaCliente() {
12     }
13
14     public cuentaCliente(int ID, double saldo) {
15         this.ID = ID;
16         this.Saldo = Saldo;
17     }
18
19
20     public int getID() {
21         return ID;
22     }
23
24     public void setID(int ID) {
25         this.ID = ID;
26     }
27
28     public boolean isDisponible() {
29         return disponible;
30     }
31 }
```

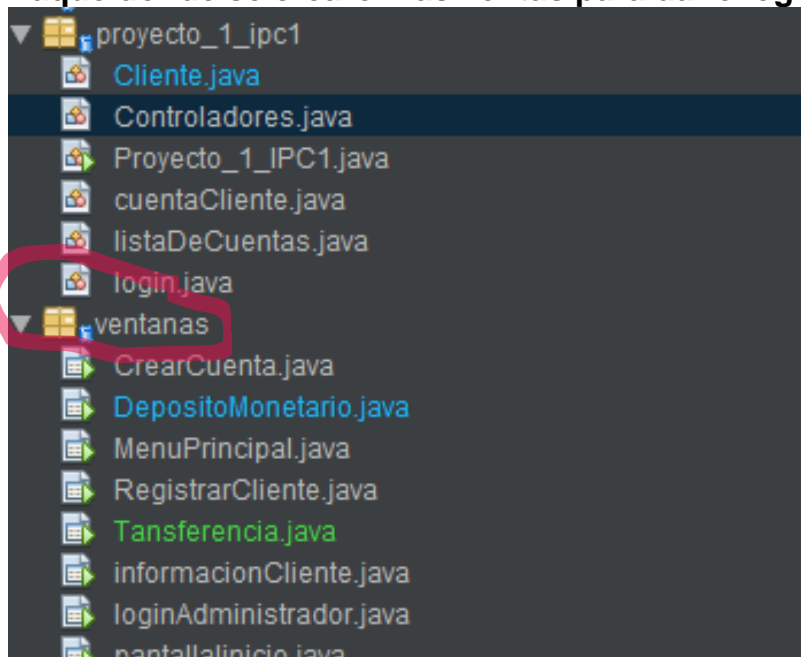
Clase controlador

Esto se creo para manejar los diferentes métodos creados en las diferentes clases

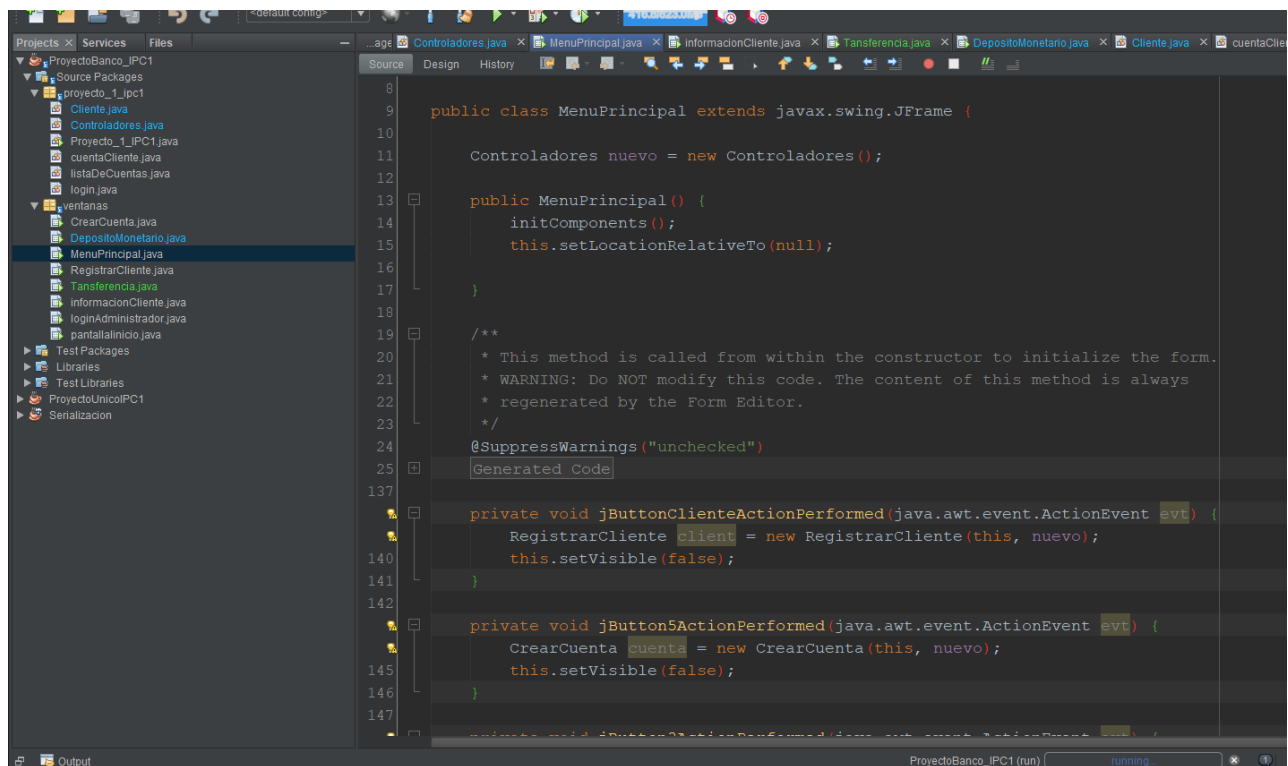


```
56
57 //metodo para crear y verificar cuenta
58 public void verificarCuenta(int cui) {
59     for (int i = 0; i < cliente.length; i++) {
60         if (cliente[i].getCUI() == cui) {
61             cliente[i].crearCuenta();
62         } else {
63         }
64     }
65 }
66
67
68 //metodo para buscar cui y me devuelve la posicion donde se encuentra
69 //para recorrer la lista de cuentas
70 public int buscarIndice(int cui) {
71     int indic = 5;
72     for (int i = 0; i < cliente.length; i++) {
73         if (cliente[i].getCUI() == cui) {
74             indic = i;
75         } else {
76             if (cliente[i].getCUI() == 0) {
77             }
78         }
79     }
80     return indic;
81 }
82
83 }
```

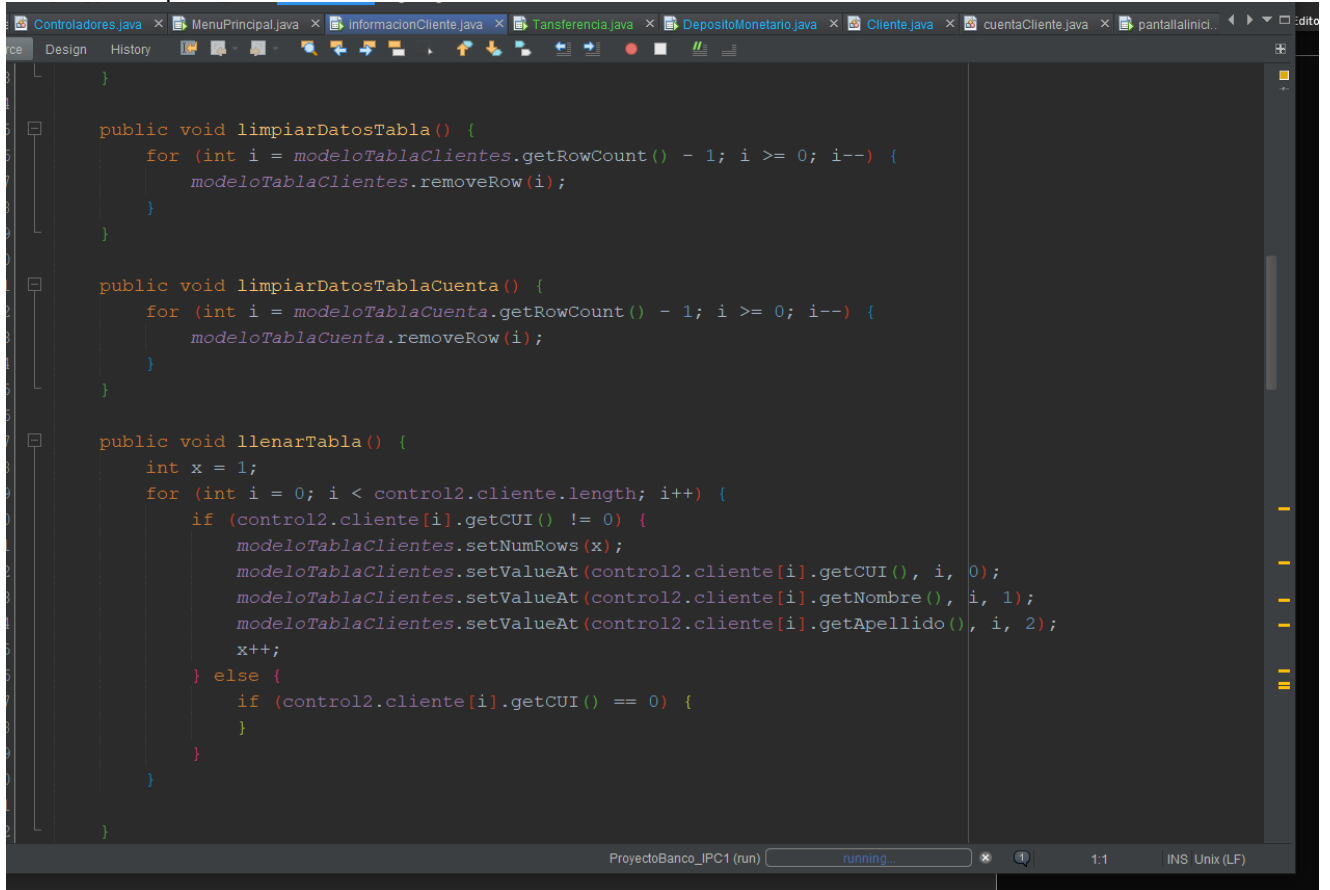
Paquete donde se crearon las ventanas para darle lógica a la aplicación



En este paquete se encuentran los diferentes jframes form las cuales se utilizo para crear las tablas los combobox y en estas clases se crearon los diferentes metos para poder maneja los datos del programa



La forma en que se llenaron las tablas



```
Controladores.java x MenuPrincipal.java x InformacionCliente.java x Transferencia.java x DepositoMonetario.java x Cliente.java x cuentaCliente.java x pantallaInicio... :dito
ce Design History

}

public void limpiarDatosTabla() {
    for (int i = modeloTablaClientes.getRowCount() - 1; i >= 0; i--) {
        modeloTablaClientes.removeRow(i);
    }
}

public void limpiarDatosTablaCuenta() {
    for (int i = modeloTablaCuenta.getRowCount() - 1; i >= 0; i--) {
        modeloTablaCuenta.removeRow(i);
    }
}

public void llenarTabla() {
    int x = 1;
    for (int i = 0; i < control2.cliente.length; i++) {
        if (control2.cliente[i].getCUI() != 0) {
            modeloTablaClientes.setNumRows(x);
            modeloTablaClientes.setValueAt(control2.cliente[i].getCUI(), i, 0);
            modeloTablaClientes.setValueAt(control2.cliente[i].getNombre(), i, 1);
            modeloTablaClientes.setValueAt(control2.cliente[i].getApellido(), i, 2);
            x++;
        } else {
            if (control2.cliente[i].getCUI() == 0) {
            }
        }
    }
}

ProyectoBanco_IPC1 (run) running... 1:1 INS Unix (LF)
```