

# Manual Técnico



# **Funcionalidades**

---

El programa está diseñado para administrar las cuentas de los clientes del Banco De Gotham.

## **Organización:**



El programa se compone de un solo paquete, que alberga las clases y los jform.

## **Clases:**

Tiene un total de 16.



# Pantalla creación de cliente:

```
20 public void Creacion(long id, String NombreUsuario, String ApellidoUsuario){
21     boolean r1= true;
22     princ: //
23     for (int i = 0; i < ArregloUsu.length; i++){
24         if (ArregloUsu[i] != null){
25             if(ArregloUsu[i].getId() == id){
26                 JOptionPane.showMessageDialog(null,"Cuenta Existe","Error de login",JOptionPane.ERROR_MESSAGE);
27                 r1 = false;
28                 break;
29             }else{
30                 continue;
31             }
32         }else{
33             break;
34         }
35     }
36
37     if(r1 == true){
38         for (int i = 0; i < ArregloUsu.length; i++) {
39             //System.out.println("For");
40             if (ArregloUsu[i] == null) {
41                 int c = 100+i;
42                 String tempC = String.valueOf(c);
43                 ArregloUsu[i]= new Usuarios(id, NombreUsuario,ApellidoUsuario,tempC);
44                 //System.out.println(ArregloUsu);
45                 System.out.println(ArregloUsu[i].getContraseña());
46                 System.out.println(ArregloUsu[i].getId());
47                 System.out.println(ArregloUsu[i].getNombreUsuario());
48                 JOptionPane.showMessageDialog(null,"Usuario creado con exito, la contraseña es: "+tempC,"Usuario Creado", JOptionPane.INFORMATION_MESSAGE);
49             }
50         }
51     }
52 }
53 }
```

# Pantalla eliminación de cliente:

```
private void EliminarClienteActionPerformed(java.awt.event.ActionEvent evt) {
    long eli = Long.parseLong(IdCliente.getText());
    for (int i = 0; i < ArregloUsu.length; i++) {
        if(ArregloUsu[i]!=null){
            if(ArregloUsu[i].getId() == eli ){
                ArregloUsu[i]=null;
                //ArregloUsu[i].setNombreUsuario("null");
                //ArregloUsu[i].setId();
                //ArregloUsu[i].setContraseña("null");
                //System.out.println(ArregloUsu[i].getId());
            }
        }
    }
    // TODO add your handling code here:
}
```



# Pantalla ver clientes:

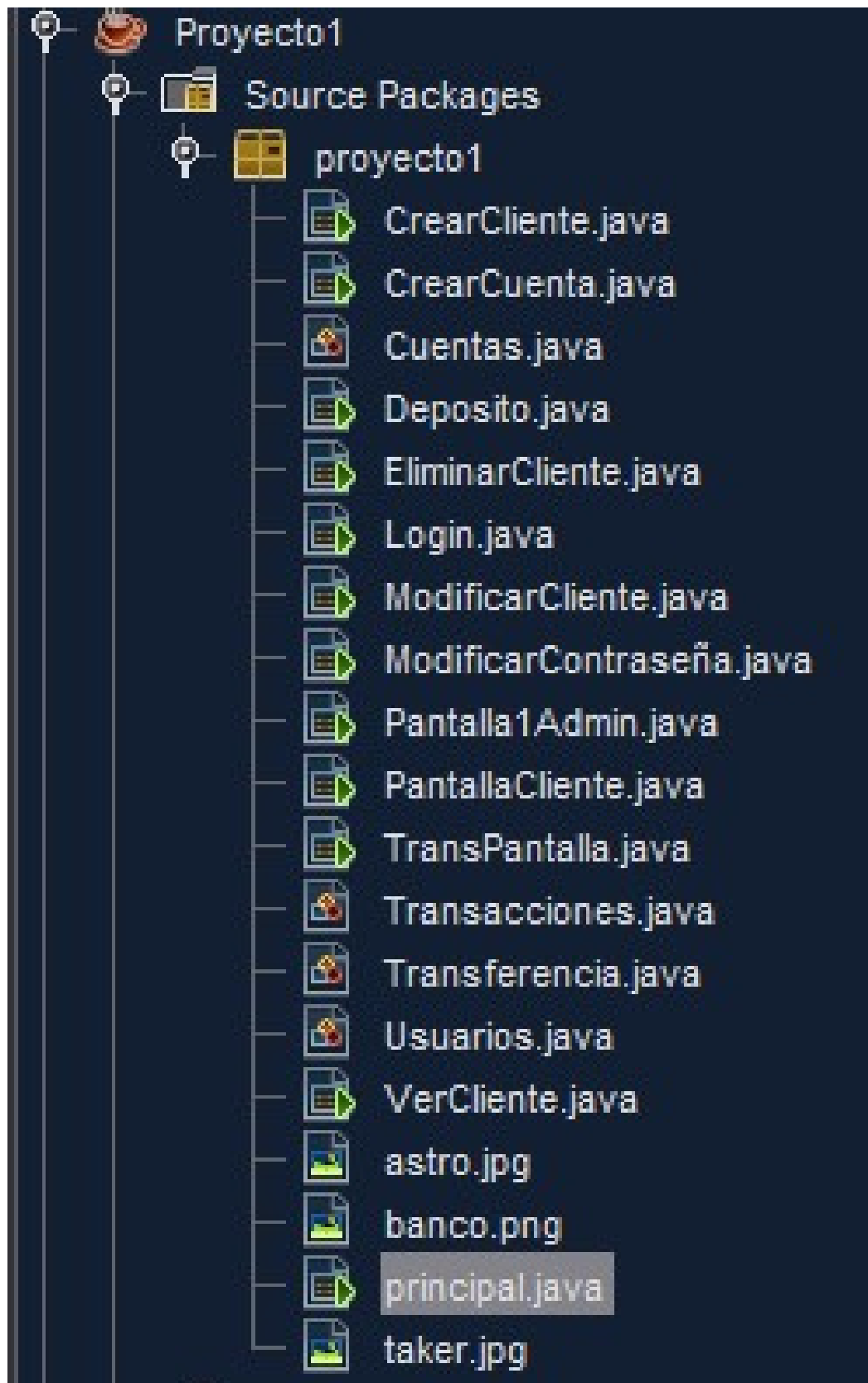
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    JTable.getModel();  
    DefaultTableModel modelo = (DefaultTableModel)JTable.getModel();  
    for (int i = 1; i < ArregloUsu.length; i++) {  
        if (ArregloUsu[i] != null) {  
            String[] fila = {String.valueOf(i), String.valueOf(ArregloUsu[i].getId()), ArregloUsu[i].getNombreUsuario(), ArregloUsu[i].getApellidoUsuario(), ArregloUsu[i].getContraseña()};  
            //NombreCliente.setText(ArregloUsu[i].getNombreUsuario());  
            //ApellidoCliente.setText(ArregloUsu[i].getApellidoUsuario());  
            //System.out.println(ArregloUsu[i].getId());  
            modelo.addRow(fila);  
        }  
        else {  
            JOptionPane.showMessageDialog(null, "Usuario no existe", "Error de login", JOptionPane.ERROR_MESSAGE);  
        }  
    }  
    //modelo.addRow(rowData);  
}  
  
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {  
    this.setState(Login.ICONIFIED);  
    // TODO add your handling code here:  
}  
  
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {  
    //System.exit(0);  
    this.setVisible(false);  
    new PantallaAdmin().setVisible(true);  
    // TODO add your handling code here:  
}  
  
private void jButton4MouseClicked(java.awt.event.MouseEvent evt) {  
    Close.setBackground(Color.red);  
    // TODO add your handling code here:  
}
```

# Pantalla deposito:

```
private void IdClienteFocusLost(java.awt.event.FocusEvent evt) {  
    long idi = Long.parseLong(IdCliente.getText());  
    int posicionl = -1;  
    for (int i = 0; i < ArregloUsu.length; i++) {  
        if (ArregloUsu[i] != null) {  
            if (ArregloUsu[i].getId() == idi) {  
                //Nombre.setText(ArregloUsu[i].getNombreUsuario() + " " + ArregloUsu[i].getApellidoUsuario());  
                posicionl = i;  
                //ApellidoCliente.setText(ArregloUsu[i].getApellidoUsuario());  
            }  
        }  
    }  
    try {  
        if (posicionl >= 0) {  
            for (int i = 0; i < ArregloUsu[posicionl].cuentas.length; i++) {  
                System.out.println(ArregloUsu[posicionl].cuentas.length);  
                if (ArregloUsu[posicionl].cuentas[i] != null) {  
                    System.out.println(ArregloUsu[posicionl].cuentas[i].getId());  
                    String Idcuenta = String.valueOf(Integer.toString(ArregloUsu[posicionl].cuentas[i].getId())); //Nuevo  
                    String Nombre = ArregloUsu[posicionl].getNombreUsuario() + " " + ArregloUsu[posicionl].getApellidoUsuario();  
                    //this.jComboBox1.addItem(Integer.toString(ArregloUsu[posicionl].cuentas[i].getId()));  
                    int aux = i+1;  
                    this.jComboBox1.addItem(aux + ". " + Idcuenta + " - " + Nombre);  
                }  
                else {  
                    continue;  
                }  
            }  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, "Ingreso Saldo", "Error", JOptionPane.ERROR_MESSAGE);  
    }  
    for (int i = 0; i < ArregloUsu[i].cuentas.length; i++) {  
        System.out.println(ArregloUsu[i].cuentas[i].getId());  
        this.jComboBox1.addItem(Integer.toString(ArregloUsu[i].cuentas[i].getId()));  
    }  
}
```

# Anexos

---



# MANUAL DE USUARIO

---

BANCO DE GOTHAM

---



# Funcionalidades

---

- El programa está diseñado para administrar las cuentas de los clientes del Banco De Gotham.

El administrador podrá hacer:

- Creación, modificación y eliminación de cliente.
- Creación de cuentas para cada cliente con un máximo de 3 cuentas.
- Depósitos a las cuentas de los clientes.
- Ver clientes creados.
- Crear y ver servicios.

El Cliente podrá hacer:

- Transferir a cuentas
- Pagar servicios
- Ver historial
- Modificar contraseña

# Administrador



\* \* \* \*

Usuario: 1024

Contraseña: password

Pantalla inicio:

- X



Salir

## Usuario

Crear

Ver

Nueva Cuenta

Modificar

Eliminar

Deposito Cuenta

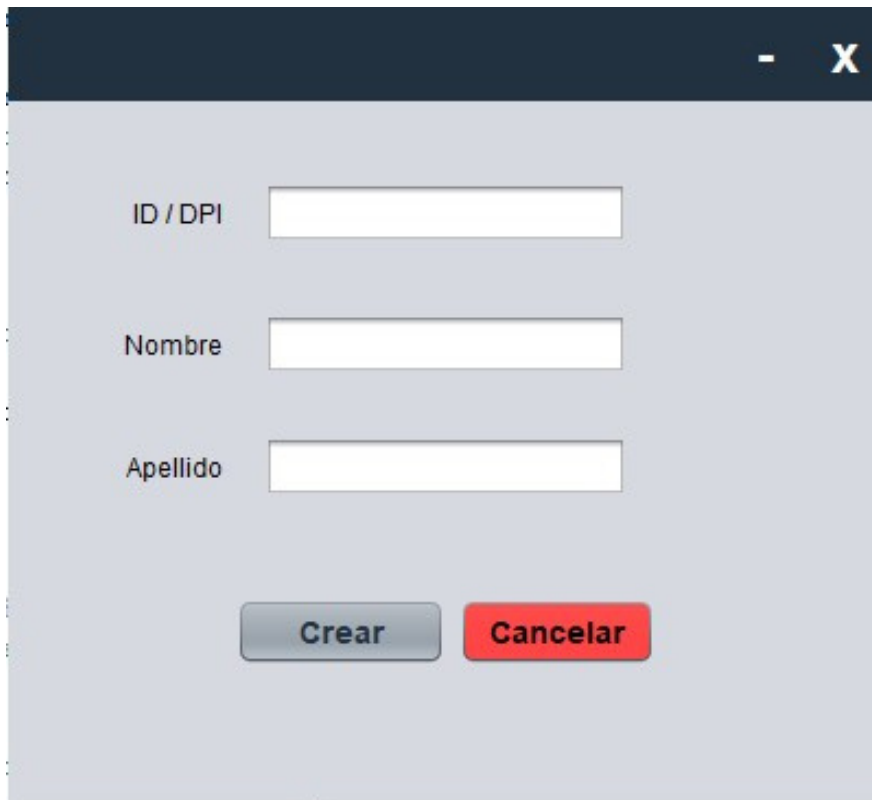
## Servicios

Crear Servicio

Ver Servicio



# Pantalla creación de cliente:



A screenshot of a web application window titled "Pantalla creación de cliente". The window has a dark blue header bar with a minus sign and a close button (X). The main content area is light gray and contains three input fields stacked vertically. The first field is labeled "ID / DPI", the second "Nombre", and the third "Apellido". Below the input fields are two buttons: a gray "Crear" button and a red "Cancelar" button.

# Pantalla eliminación de cliente:



A screenshot of a web application window titled "Pantalla eliminación de cliente". The window has a dark blue header bar with a minus sign and a close button (X). The main content area is light gray and contains a search bar with the text "1234567894321" and a "Buscar" button. Below the search bar are two input fields: the first contains the text "Jose" and the second contains the text "Gonzales". At the bottom are two buttons: a gray "Eliminar" button and a red "Cancelar" button.

# Pantalla ver clientes:

VER CLIENTES				
No	ID	Nombre	Apellido	Contraseña
1	1234567891234	Marvin	Matias	101
2	1234567894321	Jose	Gonzales	102

Mostrar Clientes

# Pantalla deposito:

1234567891234

Cuentas

Cuentas

11000 - Marvin Matias

21001 - Marvin Matias

31002 - Marvin Matias

Depositar

Cancelar