

Universidad Tecnológica de Panamá
Facultad de Sistemas Computacionales
Asignatura: Desarrollo de Software III
Ejercicio Práctico3

Profesor: Napoleón Ibarra

Valor: 100 puntos

Estudiante: Jorge Javier Jiménez Ruiz

Cédula: 4-826-874

Procedimiento:

1. De manera individual, realizar la asignación.
2. Entregar el trabajo en formato digital en la plataforma utilizada. Sustente su desarrollo en clase.

Criterios de Evaluación:

Criterios	Puntos (Mínimo=1, Máximo=5)	Porcentaje
Sustentación	1 - 5	10 %
Puntualidad	1 - 5	10 %
Creatividad	1 - 5	10 %
Desarrollo	1 - 5	70 %

I PARTE. Desarrollo. Valor 70 Puntos.

¿Qué se debe entregar?

1. Código funcional del desarrollo (prototipo).
2. Interfaz gráfico (A su criterio) que inicia el desarrollo, desde ahí se llama la simulación. Este desarrollo de la interfaz es según criterio y desarrollo del grupo (diseño visual).
3. Simulación del desarrollo funcionando.
4. Contemple dentro de su código cálculos N veces para realizar la simulación, control dentro del código Try-Catch.

PROBLEMA.

Crea un programa en JAVA para realizar recargas de tiempo aire de las diferentes compañías telefónicas +Movile, TYGO, TLK. (Crea una clase independiente para cada compañía y un método para ser llamado por una clase Principal), el programa deberá de contar con los siguientes puntos:

1. Un menú para seleccionar a las compañías telefónicas.
2. Al seleccionar a la compañía del menú el programa pedirá:
 - 2.1. La cantidad al ser abonada en dinero. Controlar la forma de verificar el pago correspondiente.
 - 2.2. Solicitará el numero celular para abonarle el saldo indicado. Debe llegarle un mensaje al cliente que ya fue aprobado su saldo. Debe guardarse esa transacción en una base de datos para que pueda ser verificada en un momento determinado.
 - 2.3. Solicitar a que se confirme el numero celular. Puede usar un numeración para la confirmación u otro detalle.
 - 2.4. Si los números son validados aparecerá en la pantalla usted abono \$ X (Balboas) al número celular de lo contrario intente nuevamente ingresando el numero celular (Deberán existir precargados 5 números celulares en la base de datos, y la posibilidad de crear, modificar, eliminar)
- 2.5. Esta misma validación para todas las compañías telefónicas (Diferentes números).

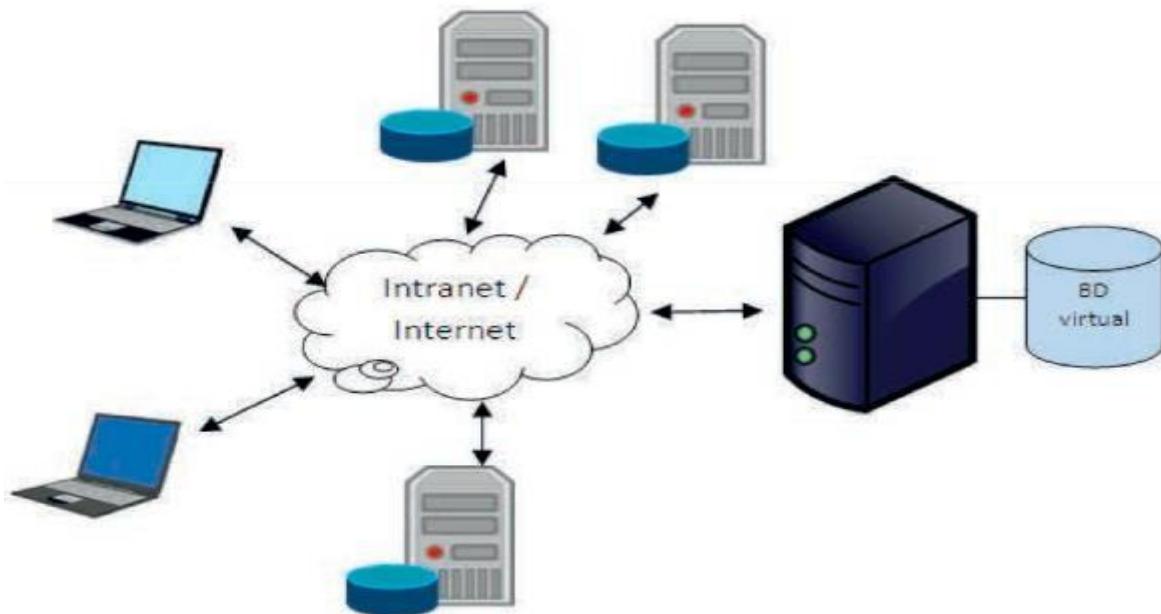


Figura 1. Esquema posible del Sistema.

Desarrollo - imágenes

NetBeans:

The screenshot shows the NetBeans IDE interface. On the left, the Projects tab displays several projects: arbol, arbolesBin, calculadoraPratica, mysql (with Source Packages and Libraries), Test Packages, Libraries, JDK 25 (Default), and Test Libraries. A red arrow points from the 'Librería añadida' label to the mysql-connector-j-9.3.0.jar file in the Libraries folder. The main window shows the DBConnection.java code:

```
1 package mysql;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5
6 public class DBConnection {
7     private static final String URL = "jdbc:mysql://localhost:3306/recargas";
8     private static final String USER = "root";
9     private static final String PASS = "25Junio2004";
10
11    public static Connection getConnection() throws SQLException {
12        return DriverManager.getConnection(URL, USER, PASS);
13    }
14 }
```

The screenshot shows the NetBeans IDE interface with the Movable.java code open. The code implements a method to recharge a phone number with a given amount. It uses the DBConnection class to check if the number exists and then inserts a new transaction into the transacciones table.

```
1 package mysql;
2 import java.sql.Connection;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import javax.swing.JOptionPane;
6
7 public class Movable {
8     public void recargar(String numero, double monto) {
9         try (Connection conn = DBConnection.getConnection()) {
10             String verificarSQL = "SELECT * FROM clientes WHERE numero = ?";
11             PreparedStatement stmt = conn.prepareStatement(verificarSQL);
12             stmt.setString(1, numero);
13             ResultSet rs = stmt.executeQuery();
14
15             if (rs.next()) {
16                 String insertSQL = "INSERT INTO transacciones (compania, numero, monto) VALUES (?, ?, ?)";
17                 PreparedStatement insertStmt = conn.prepareStatement(insertSQL);
18                 insertStmt.setString(1, "Movable");
19                 insertStmt.setString(2, numero);
20                 insertStmt.setDouble(3, monto);
21                 insertStmt.executeUpdate();
22
23                 JOptionPane.showMessageDialog(null, "Recarga exitosa de $" + monto + " al numero " + numero);
24             } else {
25                 JOptionPane.showMessageDialog(null, "Número no encontrado. Intente nuevamente.");
26             }
27         } catch (Exception e) {
28             JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
29         }
30     }
31 }
```

Nota: TLK y TYGO tienen lo mismo, así que por eso solo estará esta imagen.

```
1 package mysql;
2 import javax.swing.JOptionPane;
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6
7 public class Main {
8     public static void main(String[] args) {
9         String[] menuOpciones = {"Recargas", "Verificar"};
10        String accion = (String) JOptionPane.showInputDialog(
11                null,
12                "¿Quequieres hacer?",
13                "Menu Principal",
14                JOptionPane.QUESTION_MESSAGE,
15                null,
16                menuOpciones,
17                menuOpciones[0]
18            );
19
20        if (accion == null) {
21            JOptionPane.showMessageDialog(null, "Operacion cancelada.");
22            return;
23        }
24
25        if (accion.equals("Verificar")) {
26            try (Connection conn = DBConnection.getConnection()) {
27                String sql = "SELECT numero FROM clientes";
```

```
28         PreparedStatement stmt = conn.prepareStatement(sql);
29         ResultSet rs = stmt.executeQuery();
30
31         StringBuilder numeros = new StringBuilder("Numeros registrados:\n");
32         while (rs.next()) {
33             numeros.append("- ").append(rs.getString("numero")).append("\n");
34         }
35
36         JOptionPane.showMessageDialog(null, numeros.toString(), "Clientes", JOptionPane.INFORMATION_MESSAGE);
37
38     } catch (Exception e) {
39         JOptionPane.showMessageDialog(null, "Error al consultar la base de datos: " + e.getMessage());
40     }
41     return; // termina aqui si elige "Verificar"
42 }
43
44 // Si selecciono "Recargas", continua con la logica existente
45
46 String[] opciones = {"+Movile", "TYGO", "TLK"};
47 String compania = (String) JOptionPane.showInputDialog(
48         null,
49         "Seleccione compania:",
50         "Menu de Recargas",
51         JOptionPane.QUESTION_MESSAGE,
52         null,
53         opciones,
54         opciones[0]);
```

```
55
56     if (compania == null) {
57         JOptionPane.showMessageDialog(null, "Operacion cancelada.");
58         return;
59     }
60
61     String numero = JOptionPane.showInputDialog("Ingrese numero celular:");
62     if (numero == null || numero.trim().isEmpty()) {
63         JOptionPane.showMessageDialog(null, "Operacion cancelada.");
64         return;
65     }
66     if (!numero.matches("\\d{8}")) {
67         JOptionPane.showMessageDialog(null, "Numero invalido. Debe tener exactamente 8 digitos numericos.");
68         return;
69     }
70
71     String confirmacion = JOptionPane.showInputDialog("Confirme numero celular:");
72     if (confirmacion == null || confirmacion.trim().isEmpty()) {
73         JOptionPane.showMessageDialog(null, "Operacion cancelada.");
74         return;
75     }
76     if (!confirmacion.equals(numero)) {
77         JOptionPane.showMessageDialog(null, "Los numeros no coinciden.");
78         return;
79     }
80
81     String montoStr = JOptionPane.showInputDialog("Ingrese monto a recargar:");


```

```
82     if (montoStr == null || montoStr.trim().isEmpty()) {
83         JOptionPane.showMessageDialog(null, "Operacion cancelada.");
84         return;
85     }
86
87     double monto;
88     try {
89         monto = Double.parseDouble(montoStr);
90         if (monto <= 0) {
91             JOptionPane.showMessageDialog(null, "El monto debe ser mayor que cero.");
92             return;
93         }
94     } catch (NumberFormatException e) {
95         JOptionPane.showMessageDialog(null, "Monto invalido. Debe ser un numero.");
96         return;
97     }
98
99     switch (compania) {
100         case "+Movile":
101             new Movile().recargar(numero, monto);
102             break;
103         case "TYGO":
104             new TYGO().recargar(numero, monto);
105             break;
106         case "TLK":
107             new TLK().recargar(numero, monto);
108             break;
109     }
110
111 }
```

```
112 }
113 }
```

MySQL Workbench:

The screenshot shows the MySQL Workbench interface with two tabs open: 'clientes' and 'transacciones'. Both tabs are connected to the 'recargas' schema.

clientes Tab:

```
1 • SELECT * FROM recargas.clientes;
```

	id	numero
▶	1	61234567
	2	62345678
	3	63456789
	4	64567890
	5	65678901
*	HULL	NULL

transacciones Tab:

```
1 • SELECT * FROM recargas.transacciones;
```

	id	compania	numero	monto	fecha
▶	1	Movible	61234567	5.00	2025-05-11 18:18:15
	2	TYGO	62345678	10.00	2025-05-11 19:00:35
	3	TLK	63456789	25.00	2025-05-11 19:00:57
▶	4	Movible	61234567	15.00	2025-05-12 11:45:45
*	NULL	NULL	NULL	NULL	NULL

Nota 2: Adjunte un video probando y mostrando los resultados del Código.