

Universidad Técnica Particular de Loja

Carrera: Ingeniería en Ciencias de la computación

Asignatura: Programación Orientada a Objetos

Proyecto: Sistema de Gestión de Telefonía Móvil Estudiantil – Mov-UTPL

Integrantes del grupo:

☐ Raúl Andino

☐ Eimer Argimos

Docente: **BUSTAMANTE GRANDA WAYNER XAVIER**

2. ANÁLISIS

2.1. Propósito del sistema

El sistema **Mov-UTPL** busca gestionar servicios de telefonía móvil para estudiantes de la UTPL, permitiendo registrar clientes, asignarles uno o dos planes, y generar facturas automáticas con detalle de consumo y fechas.

2.2. Requerimientos funcionales

ID Requerimiento

RF1 Registro de clientes

RF2 Asignación de uno o dos planes por cliente

RF3 Gestión de planes (nombre, megas, minutos, costo)

RF4 Generación automática de facturas mensuales

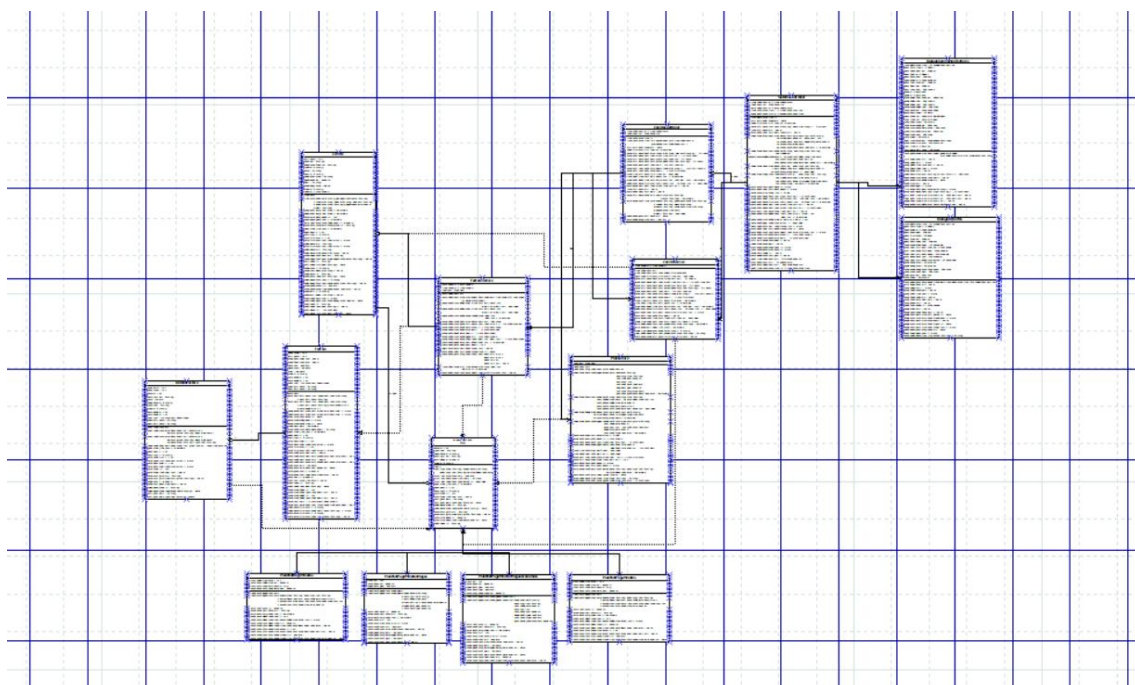
RF5 Consulta de facturación por cliente

2.3. Base de datos

SQLite es utilizada como motor embebido. Se implementan tres tablas principales:

- **Cliente (id, nombre, cédula, correo)**
- **Plan (id, tipo, minutos, megas, costo)**
- **Factura (id, idCliente, fecha, total, idPlan1, idPlan2)**

3. DISEÑO (DIAGRAMAS UML)



3.2. Diagrama de casos de uso

- Registrar cliente
- Registrar plan
- Asignar plan a cliente
- Generar factura
- Visualizar historial

4. CODIFICACIÓN

4.1. Fragmentos relevantes

Ejemplo: Registro de cliente (Java)

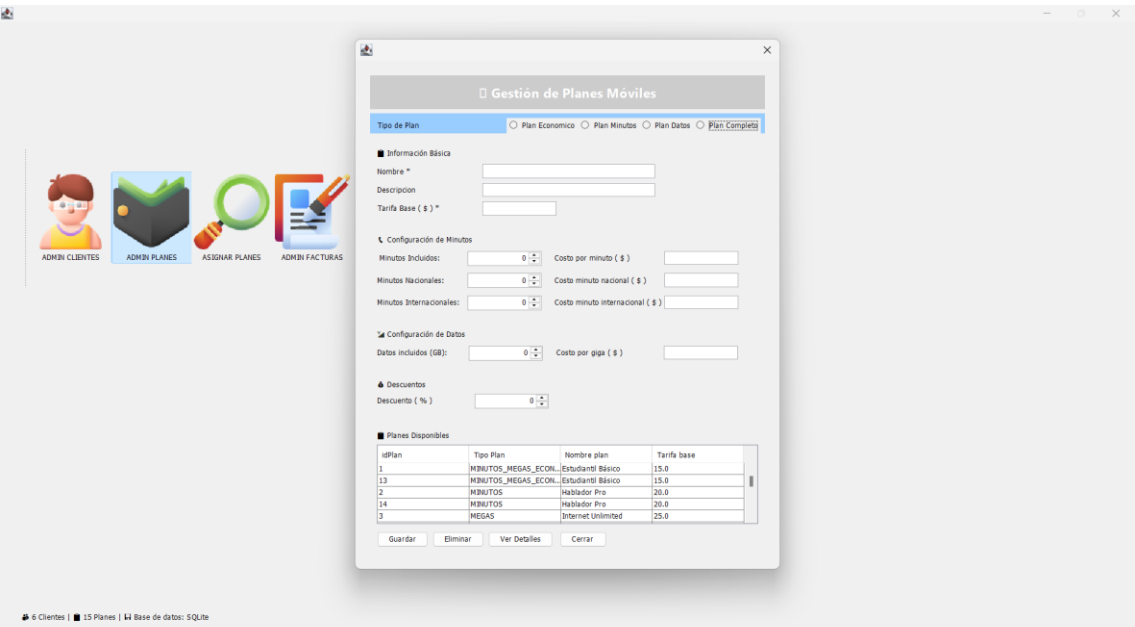
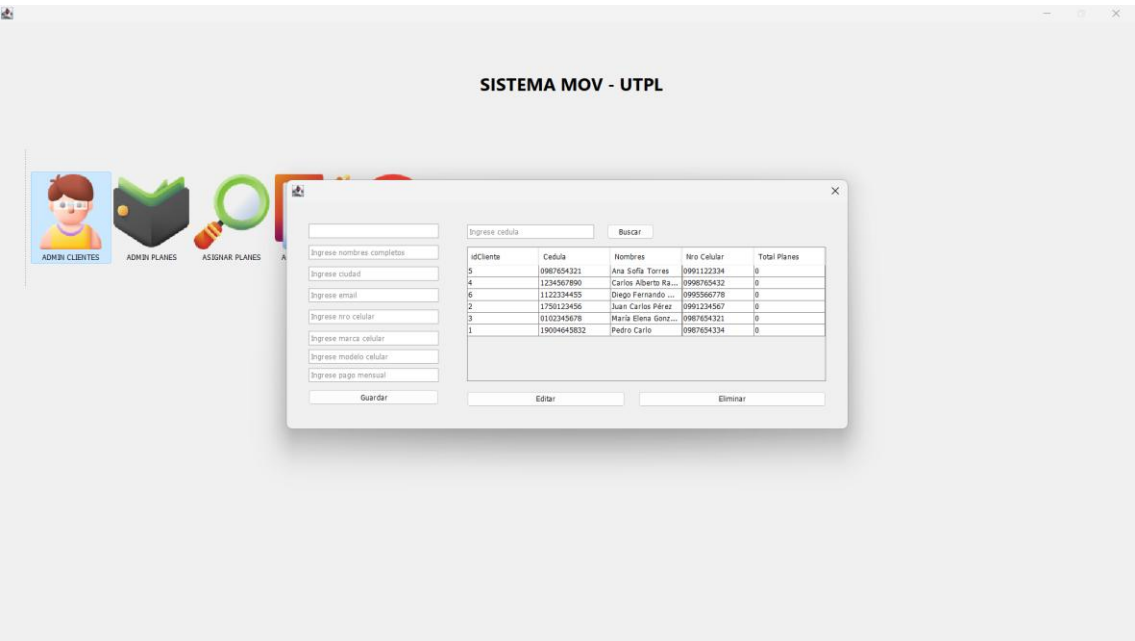
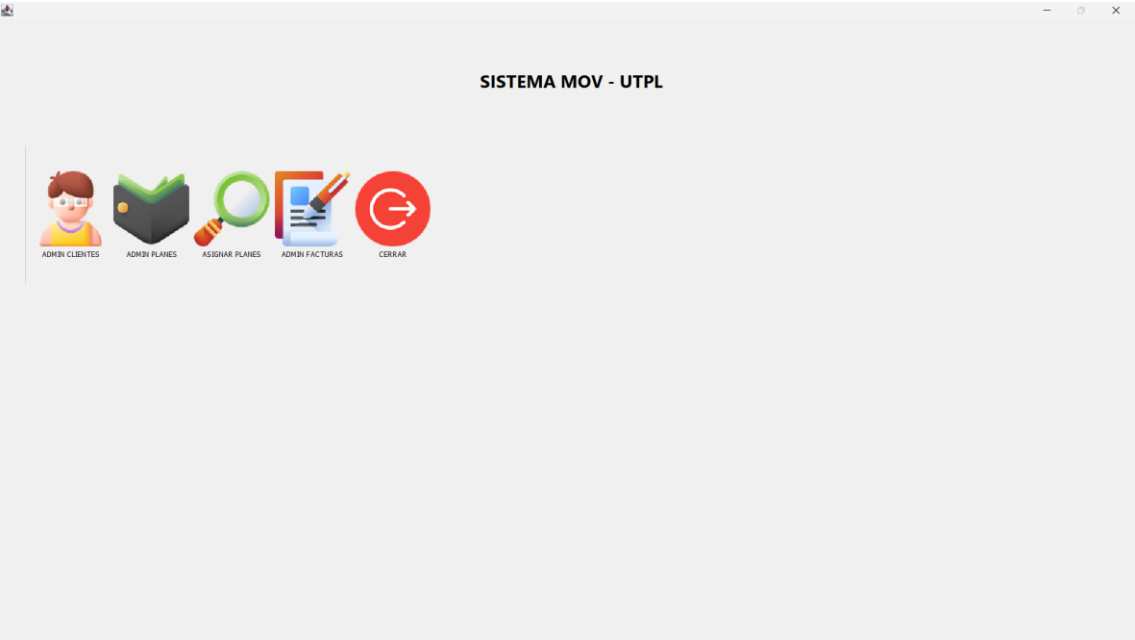
```
public String obtenerInformacionCompleta() {
    StringBuilder info = new StringBuilder();
    info.append("=== INFORMACIÓN DEL CLIENTE ===\n");
    info.append(String.format("ID: %d\n", getIdCliente()));
    info.append(String.format("Nombres: %s\n", getNombres()));
    info.append(String.format("Cédula/Pasaporte: %s\n", getPasaporteCedula()));
    info.append(String.format("Ciudad: %s\n", getCiudad()));
    info.append(String.format("Dispositivo: %s %s\n", getMarca(), getModelo()));
    info.append(String.format("Número: %s\n", getNumeroCelular()));
    info.append(String.format("Email: %s\n", getEmail()));
    info.append(String.format("Fecha de registro: %s\n", getFechaRegistro()));
    info.append(String.format("Estado: %s\n", getEstado()));
    info.append(String.format("Pago mensual base: $%.2f\n", getPagoMensual()));

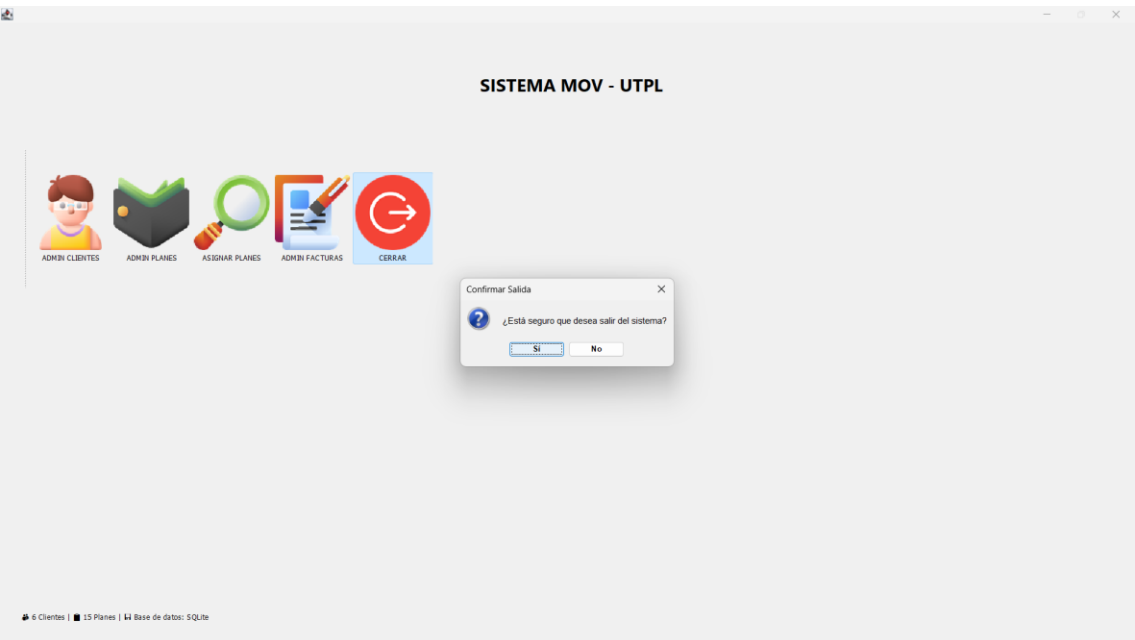
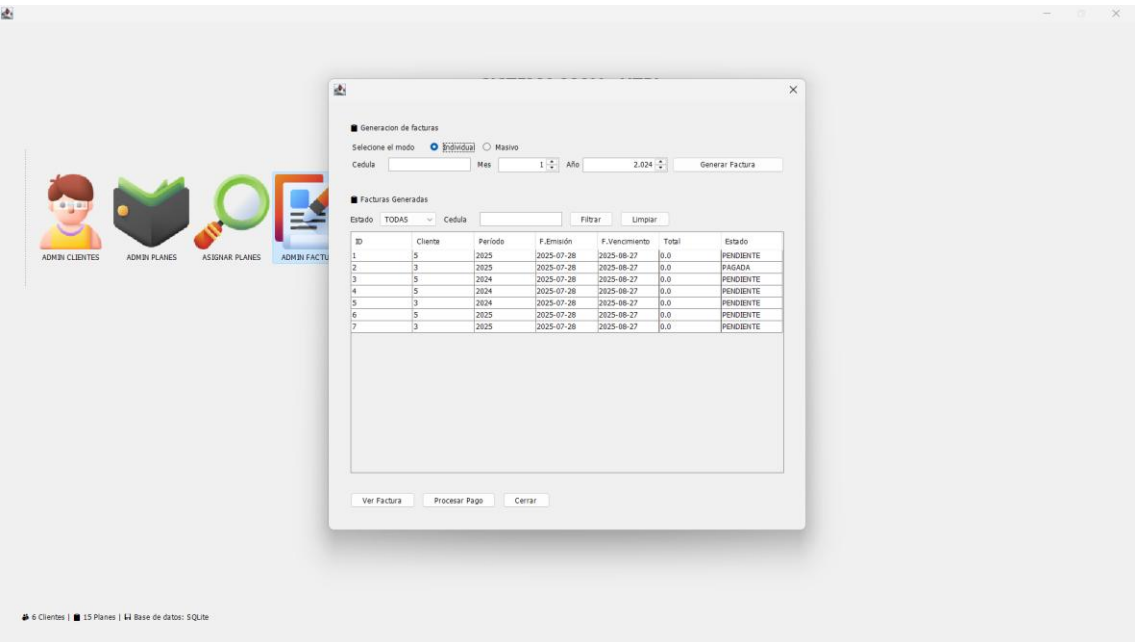
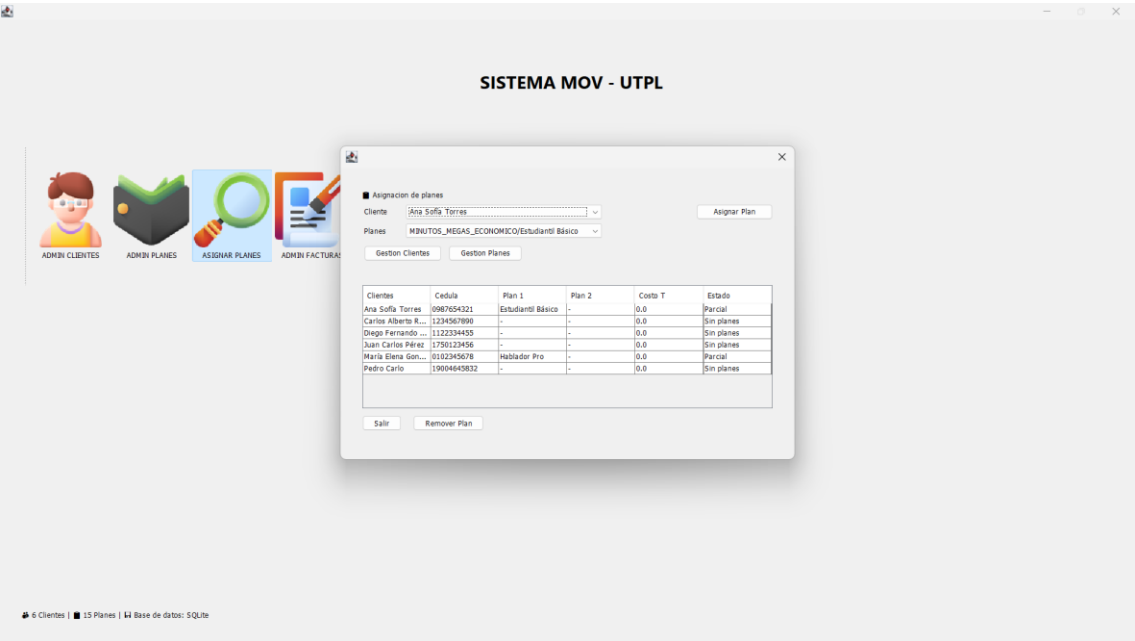
    info.append("\n=== PLANES CONTRATADOS ===\n");
    if (getPlanes().isEmpty()) {
        info.append("No tiene planes contratados\n");
    } else {
        for (int i = 0; i < getPlanes().size(); i++) {
            info.append(String.format("Plan %d:\n", i + 1));
            info.append(getPlanes().get(i).obtenerDetallePlan().append("\n\n"));
        }
        info.append(String.format("TOTAL MENSUAL: $%.2f\n", calcularCostoTotalMensual()));
    }

    return info.toString();
}
```

```
public Cliente(String nombres, String pasaporteCedula, String ciudad,
    String marca, String modelo, String numeroCelular,
    double pagoMensual, String email) {
    this();
    this.nombres = nombres;
    this.pasaporteCedula = pasaporteCedula;
    this.ciudad = ciudad;
    this.marca = marca;
    this.modelo = modelo;
    this.numeroCelular = numeroCelular;
    this.pagoMensual = pagoMensual;
    this.email = email;
}
```

5. RESULTADOS





CONCLUSIÓN

El sistema **Mov-UTPL** cumple los requisitos planteados para una plataforma de gestión de servicios móviles para estudiantes. Se evidencia el uso de programación orientada a objetos, herencia, polimorfismo, y persistencia en base de datos SQLite. El diseño modular y la estructura MVC permiten una fácil escalabilidad.