**Informe Proyecto Bimestral 2**

**Bimestre 2 – Programación Orientado a Objetos**

**TEMA:**

**Gestión de telefonía móvil estudiantil: Mov-UTPL.**

**Integrantes:**

**Patricio Rafael Correa López**

**Darío Antonio Minga Peña**

**CARRERA:**

**Ciencias De La Computación**

**SEMESTRE:**

**Segundo Semestre**

**FECHA:**

**30/07/2025**

**Ingeniero:**

**Wayner Xavier Bustamante Granda**

1. **Índice**

* Análisis
* Problema
* Objetivo del sistema
* Estructura general del sistema
* Diseño UML
* Codificación
* Fragmentos
* Resultados

1. **Análisis**

El proyecto actual busca crear un sistema que administre clientes, planes de telefonía y facturación en el ámbito universitario, empleando fundamentos de programación orientada a objetos y la arquitectura MVC. Igualmente, se incorpora el almacenamiento de datos con SQLite y se implementan validaciones sólidas para garantizar la consistencia del sistema.

* **Problema:**  
    
  En la comunidad universitaria de la UTPL se requiere un sistema para gestionar planes móviles, permitiendo registrar clientes, asignar planes y generar facturas. El sistema debe permitir hasta 2 planes por cliente, validar datos, y calcular correctamente los costos de los planes contratados.
* **Objetivo del sistema:**

- Gestionar clientes y planes móviles.

- Calcular y almacenar automaticamente el costo total de los servicios contratados.

- Generar facturas detalladas y almacenarlas en base de datos.

- Mantener un historial coherente, validando entradas y salidas.

* **Estructura general del sistema:**  
  El sistema se divide en tres capas:
* Modelo (Model): contiene las clases Cliente, Factura, Plan, PlanPrepago, PlanPostpago.
* Controlador (Controller): gestiona la lógica de negocio.
* Vista (Viewer): contiene el menú principal y entradas por consola.

**Entidades principales:**

* **Cliente:** Atributos: nombres, cédula/pasaporte, ciudad, marca, modelo, número celular, pago mensual, email, fecha de nacimiento.
* **Planes:** Se implementan utilizando herencia y polimorfismo. Los tipos son:
  + PlanPostPagoMinutosMegasEconomico
  + PlanPostPagoMinutos
  + PlanPostPagoMegas
  + PlanPostPagoMinutosMegas
* **Factura:** Detalla la información del cliente, tipo(s) de plan asignado(s), fecha de emisión y costo total.

**Preprocesamiento del Problema:**

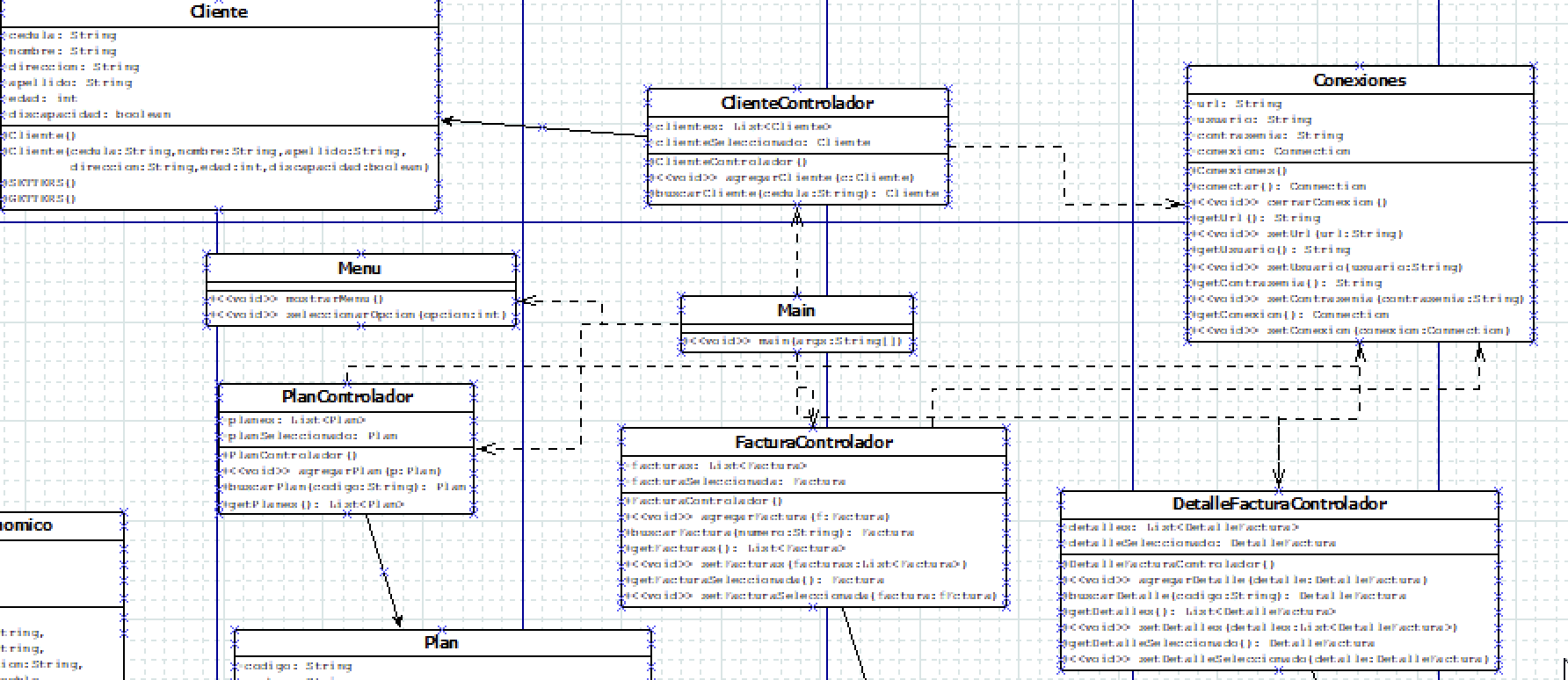
Se realizó un levantamiento de requerimientos funcionales, identificando:

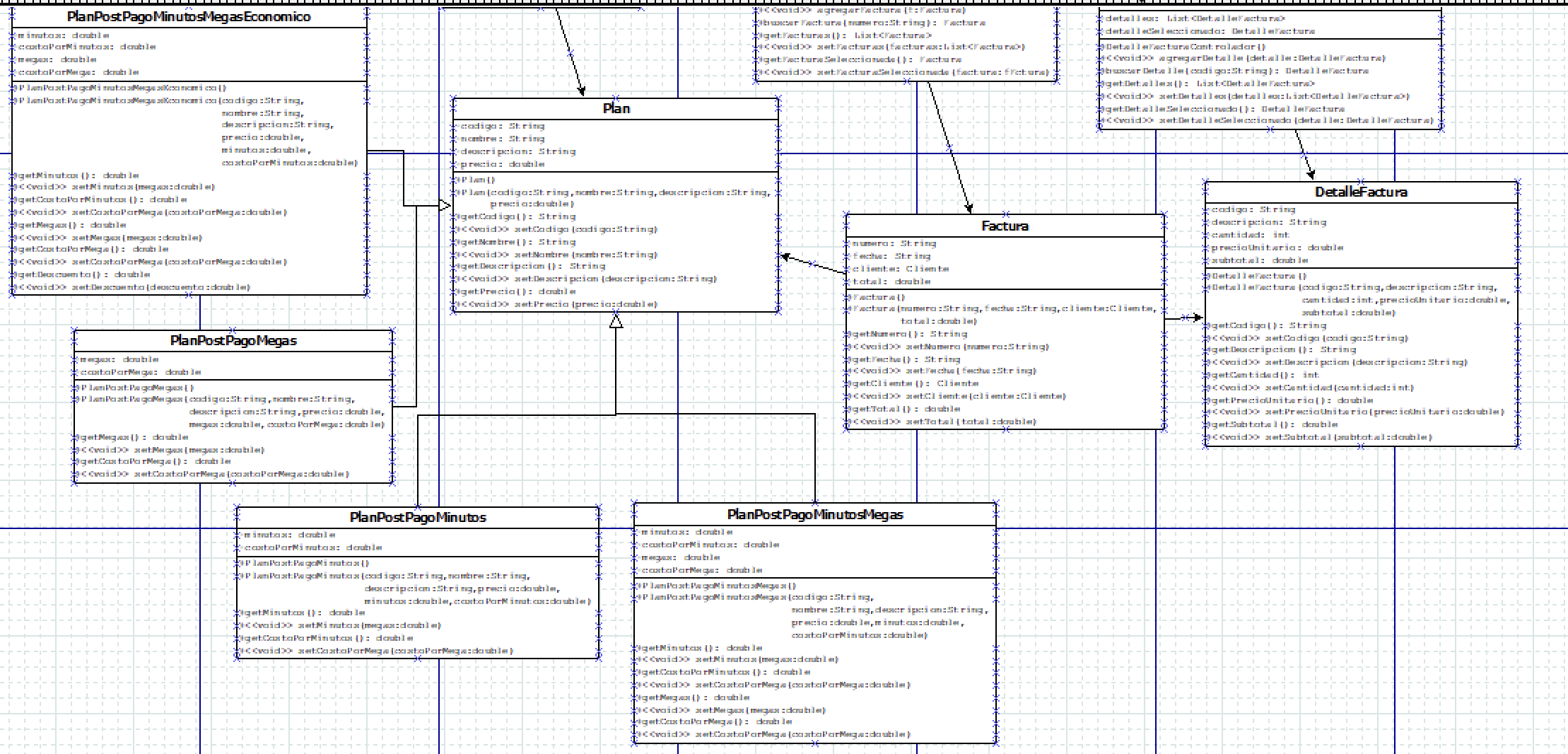
* CRUD completo para clientes y planes.
* Vínculo entre planes y clientes.
* Restricción de máximo dos planes por cliente.
* Uso de SQLite para la persistencia.
* Generación de facturas.

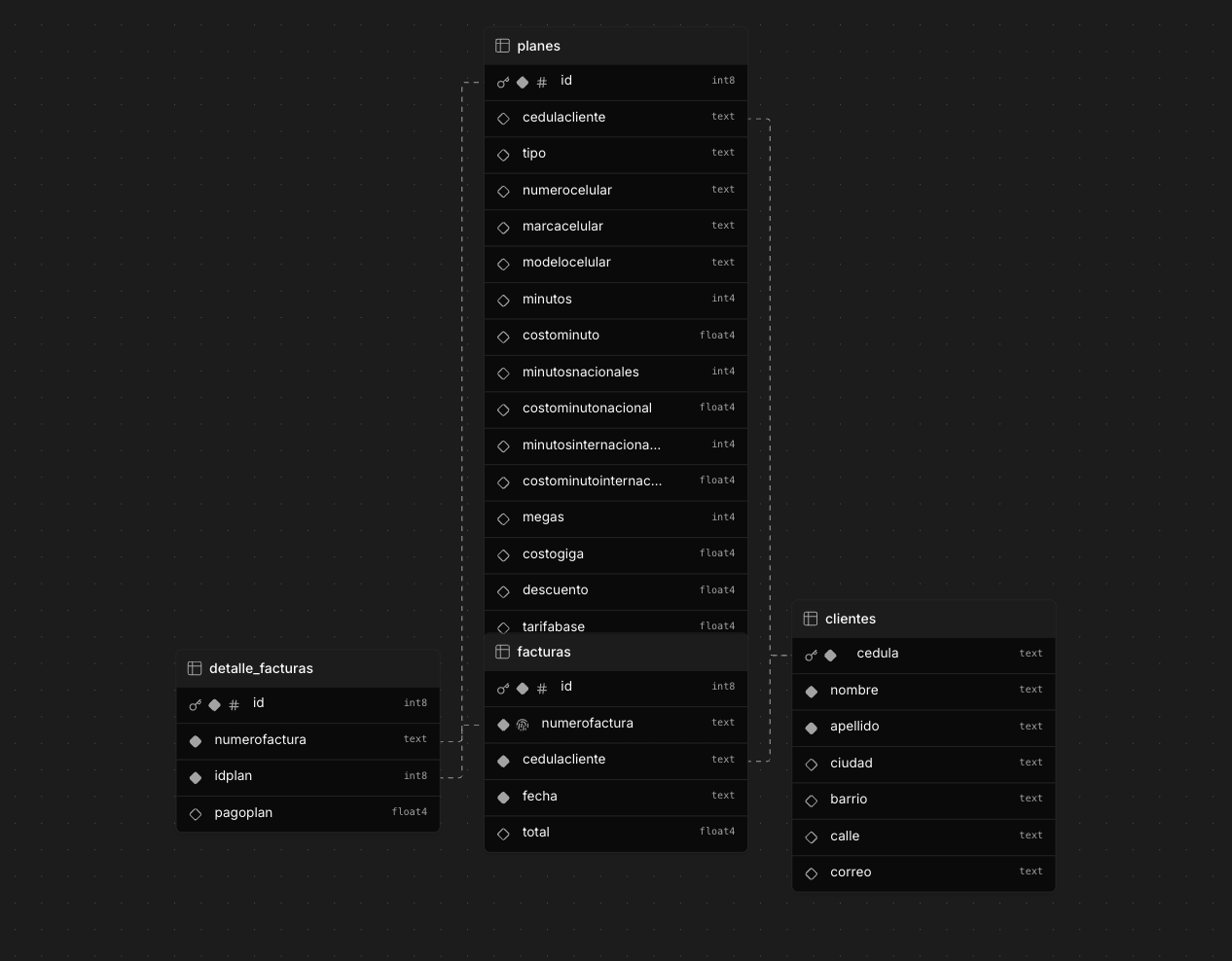
Tablas:

* **Clientes (id, nombres, cedula, ciudad, ...)**
* **Planes (id, tipo, atributos específicos según herencia)**
* **Facturas (id, idCliente, fecha, total, ...)**

1. **Diseño UML**

****

****

****

1. **Codificación**

Para nosotros todo nuestro es importante, pero cabe recalcar que como es un proyecto donde se tiene que aplicar base de datos, lo más importante vienen a ser todo lo relacionado a la misma. Aquí tenemos un ejemplo que es el controlador del cliente donde se aplica la base de datos dependiendo de la clase cliente y las conexiones.

public boolean agregarCliente(Cliente c) {

String sql = "INSERT INTO clientes (cedula, nombre, apellido, ciudad, barrio, calle, correo) VALUES (?, ?, ?, ?, ?, ?, ?)";

try (Connection conn = Conexiones.conectar();

PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setString(1, c.getCedula());

ps.setString(2, c.getNombre());

ps.setString(3, c.getApellido());

ps.setString(4, c.getCiudad());

ps.setString(5, c.getBarrio());

ps.setString(6, c.getCalle());

ps.setString(7, c.getCorreo());

ps.executeUpdate();

return true;

} catch (SQLException e) {

System.err.println("[ERROR] No se pudo insertar cliente: " + e.getMessage());

return false;

}

}

public List<Cliente> listarClientes() {

List<Cliente> lista = new ArrayList<>();

String sql = "SELECT \* FROM clientes";

try (Connection conn = Conexiones.conectar();

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery(sql)) {

while (rs.next()) {

Cliente c = new Cliente(

rs.getString("cedula"),

rs.getString("nombre"),

rs.getString("apellido"),

rs.getString("ciudad"),

rs.getString("barrio"),

rs.getString("calle"),

rs.getString("correo")

);

lista.add(c);

}

} catch (SQLException e) {

System.err.println("[ERROR] Error al listar clientes: " + e.getMessage());

}

return lista;

}

public boolean actualizarCliente(Cliente c) {

String sql = "UPDATE clientes SET nombre=?, apellido=?, ciudad=?, barrio=?, calle=?, correo=? WHERE cedula=?";

try (Connection conn = Conexiones.conectar();

PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setString(1, c.getNombre());

ps.setString(2, c.getApellido());

ps.setString(3, c.getCiudad());

ps.setString(4, c.getBarrio());

ps.setString(5, c.getCalle());

ps.setString(6, c.getCorreo());

ps.setString(7, c.getCedula());

int filas = ps.executeUpdate();

return filas > 0;

} catch (SQLException e) {

System.err.println("[ERROR] No se pudo actualizar cliente: " + e.getMessage());

return false;

}

}

public void editarCliente() {

System.out.println("\n--- Editar Cliente ---");

System.out.print("Ingrese la cédula del cliente a editar: ");

String cedula = sc.nextLine().trim();

Cliente c = buscarCliente(cedula);

if (c == null) {

System.out.println("Cliente no encontrado.");

return;

}

System.out.println("Deje vacío para no modificar el campo.");

System.out.print("Nombre (" + c.getNombre() + "): ");

String nombre = sc.nextLine().trim();

if (!nombre.isEmpty()) c.setNombre(nombre);

System.out.print("Apellido (" + c.getApellido() + "): ");

String apellido = sc.nextLine().trim();

if (!apellido.isEmpty()) c.setApellido(apellido);

System.out.print("Ciudad (" + c.getCiudad() + "): ");

String ciudad = sc.nextLine().trim();

if (!ciudad.isEmpty()) c.setCiudad(ciudad);

System.out.print("Barrio (" + c.getBarrio() + "): ");

String barrio = sc.nextLine().trim();

if (!barrio.isEmpty()) c.setBarrio(barrio);

System.out.print("Calle (" + c.getCalle() + "): ");

String calle = sc.nextLine().trim();

if (!calle.isEmpty()) c.setCalle(calle);

System.out.print("Correo (" + c.getCorreo() + "): ");

String correo = sc.nextLine().trim();

if (!correo.isEmpty()) c.setCorreo(correo);

if (actualizarCliente(c)) {

System.out.println("Cliente actualizado correctamente.");

} else {

System.out.println("Error al actualizar cliente.");

}

}

public boolean eliminarCliente(String cedula) {

String sql = "DELETE FROM clientes WHERE cedula=?";

try (Connection conn = Conexiones.conectar();

PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setString(1, cedula);

int filas = ps.executeUpdate();

return filas > 0;

} catch (SQLException e) {

System.err.println("[ERROR] No se pudo eliminar cliente: " + e.getMessage());

return false;

}

}

public Cliente buscarCliente(String cedula) {

String sql = "SELECT \* FROM clientes WHERE cedula=?";

try (Connection conn = Conexiones.conectar();

PreparedStatement ps = conn.prepareStatement(sql)) {

ps.setString(1, cedula);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

return new Cliente(

rs.getString("cedula"),

rs.getString("nombre"),

rs.getString("apellido"),

rs.getString("ciudad"),

rs.getString("barrio"),

rs.getString("calle"),

rs.getString("correo")

);

}

} catch (SQLException e) {

System.err.println("[ERROR] Error al buscar cliente: " + e.getMessage());

}

return null;

}

GITHub: https://github.com/ProOrientadaObjetos-P-E-AA2025/aab2-25-proyecto-bimestral-2do-bim-grupal-PatitoCorrea

1. **Resultados**

Después de haber realizado el programa, procedimos a probarlo con datos reales de personas. A continuación pondremos la salida del código y la base de datos

run:

[INFO] Tablas creadas o verificadas con éxito.

[INFO] La columna 'total' ya existe en la tabla 'facturas'.

=== Sistema Mov-UTPL ===

1. Gestionar Clientes

2. Gestionar Planes

3. Asignar Plan a Cliente

4. Generar Factura

5. Listar Facturas

0. Salir

Seleccione una opción: 1

--- Gestión de Clientes ---

1. Crear cliente

2. Listar clientes

3. Editar cliente

4. Eliminar cliente

0. Volver al menú principal

Seleccione una opción: 0

\*Regresando al menú principal\*

=== Sistema Mov-UTPL ===

1. Gestionar Clientes

2. Gestionar Planes

3. Asignar Plan a Cliente

4. Generar Factura

5. Listar Facturas

0. Salir

Seleccione una opción: 1

--- Gestión de Clientes ---

1. Crear cliente

2. Listar clientes

3. Editar cliente

4. Eliminar cliente

0. Volver al menú principal

Seleccione una opción: 2

Listado de clientes:

┌─────────────── Cliente ───────────────┐

│ Cédula : 1105839417

│ Nombre : Patricio Correa

│ Ciudad : Loja

│ Barrio : Los Geranios

│ Calle : Lirios y Crisantemos

│ Correo : rafaelcorrea@gmail.com

└───────────────────────────────────────┘

--- Gestión de Clientes ---

1. Crear cliente

2. Listar clientes

3. Editar cliente

4. Eliminar cliente

0. Volver al menú principal

Seleccione una opción: 0

\*Regresando al menú principal\*

=== Sistema Mov-UTPL ===

1. Gestionar Clientes

2. Gestionar Planes

3. Asignar Plan a Cliente

4. Generar Factura

5. Listar Facturas

0. Salir

Seleccione una opción: 2

--- Gestión de Planes ---

1. Crear plan

2. Listar planes

3. Editar plan

4. Eliminar plan

0. Volver al menú principal

Seleccione una opción: 2

Listado de planes:

┌───────────────── Plan PostPago Minutos ─────────────────┐

│ ID : 1

│ Cliente : 1105839417

│ Número : 0998722869

│ Marca : apple

│ Modelo : iphone 13 pro max

│ Minutos Nacionales : 1000

│ Costo Minuto Nacional : $0,10

│ Minutos Internacionales : 500

│ Costo Minuto Internacional : $0,02

└─────────────────────────────────────────────────────────┘

--- Gestión de Planes ---

1. Crear plan

2. Listar planes

3. Editar plan

4. Eliminar plan

0. Volver al menú principal

Seleccione una opción: 0

\*Regresando al menu\*

=== Sistema Mov-UTPL ===

1. Gestionar Clientes

2. Gestionar Planes

3. Asignar Plan a Cliente

4. Generar Factura

5. Listar Facturas

0. Salir

Seleccione una opción: 4

Generar factura

Ingrese cédula del cliente: 1105839417

Factura guardada en base de datos.

┌───────────────────────────── FACTURA Mov-UTPL ─────────────────────────────┐

│ Número de Factura : FAC-1753895419512 │

│ Fecha : 30/07/2025 │

├────────────────────────────── Cliente ────────────────────────────────┤

│ ┌─────────────── Cliente ───────────────┐ │

│ │ Cédula : 1105839417 │

│ │ Nombre : Patricio Correa │

│ │ Ciudad : Loja │

│ │ Barrio : Los Geranios │

│ │ Calle : Lirios y Crisantemos │

│ │ Correo : rafaelcorrea@gmail.com │

│ └───────────────────────────────────────┘ │

├──────────────────────────── Detalle de Planes ──────────────────────────┤

│ ┌───────────────── Plan PostPago Minutos ─────────────────┐ │

│ │ ID : 1 │

│ │ Cliente : 1105839417 │

│ │ Número : 0998722869 │

│ │ Marca : apple │

│ │ Modelo : iphone 13 pro max │

│ │ Minutos Nacionales : 1000 │

│ │ Costo Minuto Nacional : $0,10 │

│ │ Minutos Internacionales : 500 │

│ │ Costo Minuto Internacional : $0,02 │

│ └─────────────────────────────────────────────────────────┘ │

│ Pago mensual : $ 110,00 │

├───────────────────────────────────────────────────────────────────────────┤

│ TOTAL A PAGAR : $ 110,00 │

└───────────────────────────────────────────────────────────────────────────┘

Factura guardada en archivo: FAC-1753895419512.txt

=== Sistema Mov-UTPL ===

1. Gestionar Clientes

2. Gestionar Planes

3. Asignar Plan a Cliente

4. Generar Factura

5. Listar Facturas

0. Salir

Seleccione una opción: 5

Listado de facturas:

┌───────────────────────────── FACTURA Mov-UTPL ─────────────────────────────┐

│ Número de Factura : FAC-1753818061484 │

│ Fecha : 29/07/2025 │

├────────────────────────────── Cliente ────────────────────────────────┤

│ ┌─────────────── Cliente ───────────────┐ │

│ │ Cédula : 1105839417 │

│ │ Nombre : Patricio Correa │

│ │ Ciudad : Loja │

│ │ Barrio : Los Geranios │

│ │ Calle : Lirios y Crisantemos │

│ │ Correo : rafaelcorrea@gmail.com │

│ └───────────────────────────────────────┘ │

├──────────────────────────── Detalle de Planes ──────────────────────────┤

│ ┌───────────────── Plan PostPago Minutos ─────────────────┐ │

│ │ ID : 1 │

│ │ Cliente : 1105839417 │

│ │ Número : 0998722869 │

│ │ Marca : apple │

│ │ Modelo : iphone 13 pro max │

│ │ Minutos Nacionales : 1000 │

│ │ Costo Minuto Nacional : $0,10 │

│ │ Minutos Internacionales : 500 │

│ │ Costo Minuto Internacional : $0,02 │

│ └─────────────────────────────────────────────────────────┘ │

│ Pago mensual : $ 110,00 │

├───────────────────────────────────────────────────────────────────────────┤

│ ┌───────────────── Plan PostPago Minutos ─────────────────┐ │

│ │ ID : 1 │

│ │ Cliente : 1105839417 │

│ │ Número : 0998722869 │

│ │ Marca : apple │

│ │ Modelo : iphone 13 pro max │

│ │ Minutos Nacionales : 1000 │

│ │ Costo Minuto Nacional : $0,10 │

│ │ Minutos Internacionales : 500 │

│ │ Costo Minuto Internacional : $0,02 │

│ └─────────────────────────────────────────────────────────┘ │

│ Pago mensual : $ 110,00 │

├───────────────────────────────────────────────────────────────────────────┤

│ TOTAL A PAGAR : $ 220,00 │

└───────────────────────────────────────────────────────────────────────────┘

Detalles de la factura:

Plan ID: 1, Pago: $110,00, Tipo: PlanPostPagoMinutos, Número Celular: 0998722869

Plan ID: 1, Pago: $110,00, Tipo: PlanPostPagoMinutos, Número Celular: 0998722869

──────────────────────────────────────────────

┌───────────────────────────── FACTURA Mov-UTPL ─────────────────────────────┐

│ Número de Factura : FAC-1753895419512 │

│ Fecha : 30/07/2025 │

├────────────────────────────── Cliente ────────────────────────────────┤

│ ┌─────────────── Cliente ───────────────┐ │

│ │ Cédula : 1105839417 │

│ │ Nombre : Patricio Correa │

│ │ Ciudad : Loja │

│ │ Barrio : Los Geranios │

│ │ Calle : Lirios y Crisantemos │

│ │ Correo : rafaelcorrea@gmail.com │

│ └───────────────────────────────────────┘ │

├──────────────────────────── Detalle de Planes ──────────────────────────┤

│ ┌───────────────── Plan PostPago Minutos ─────────────────┐ │

│ │ ID : 1 │

│ │ Cliente : 1105839417 │

│ │ Número : 0998722869 │

│ │ Marca : apple │

│ │ Modelo : iphone 13 pro max │

│ │ Minutos Nacionales : 1000 │

│ │ Costo Minuto Nacional : $0,10 │

│ │ Minutos Internacionales : 500 │

│ │ Costo Minuto Internacional : $0,02 │

│ └─────────────────────────────────────────────────────────┘ │

│ Pago mensual : $ 110,00 │

├───────────────────────────────────────────────────────────────────────────┤

│ ┌───────────────── Plan PostPago Minutos ─────────────────┐ │

│ │ ID : 1 │

│ │ Cliente : 1105839417 │

│ │ Número : 0998722869 │

│ │ Marca : apple │

│ │ Modelo : iphone 13 pro max │

│ │ Minutos Nacionales : 1000 │

│ │ Costo Minuto Nacional : $0,10 │

│ │ Minutos Internacionales : 500 │

│ │ Costo Minuto Internacional : $0,02 │

│ └─────────────────────────────────────────────────────────┘ │

│ Pago mensual : $ 110,00 │

├───────────────────────────────────────────────────────────────────────────┤

│ TOTAL A PAGAR : $ 110,00 │

└───────────────────────────────────────────────────────────────────────────┘

Detalles de la factura:

Plan ID: 1, Pago: $110,00, Tipo: PlanPostPagoMinutos, Número Celular: 0998722869

Plan ID: 1, Pago: $110,00, Tipo: PlanPostPagoMinutos, Número Celular: 0998722869

──────────────────────────────────────────────

=== Sistema Mov-UTPL ===

1. Gestionar Clientes

2. Gestionar Planes

3. Asignar Plan a Cliente

4. Generar Factura

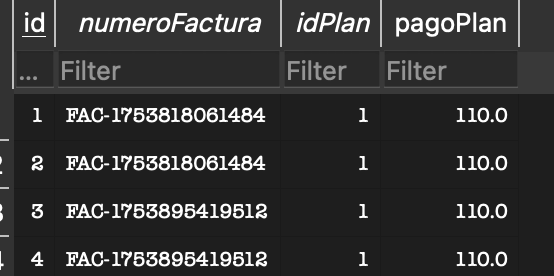
5. Listar Facturas

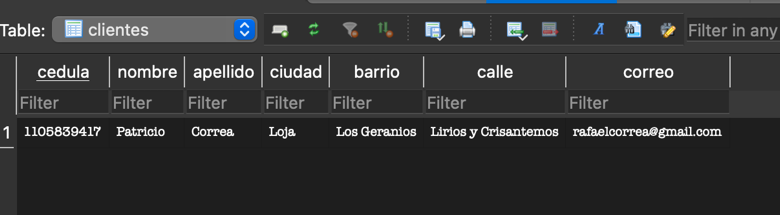
0. Salir

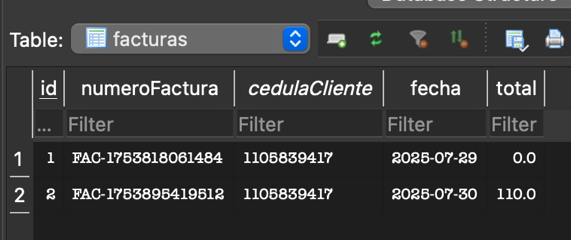
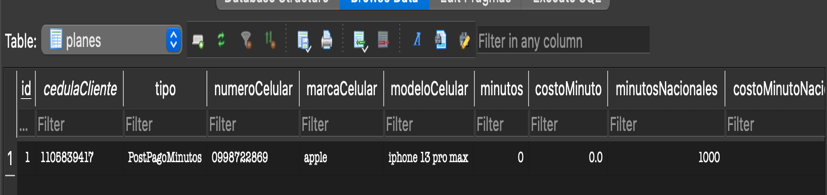
Seleccione una opción: 0

Gracias por usar nuestro sistema :D!! \*Saliendo del sistema\*

BASE DE DATOS:

****

****

****