



## **Proyecto Bimestral**

**Universidad Técnica Particular de Loja**



## **Programación Orientada a Objetos**

### **Integrantes:**

- **Samuel Reyes**
- **William Medina**
- **Lenin Gonzales**
- **Juan Pablo Salinas**

### **Docente:**

- **Ing. Wayner Bustamante**

**2023 – 2024**

## - **Análisis**

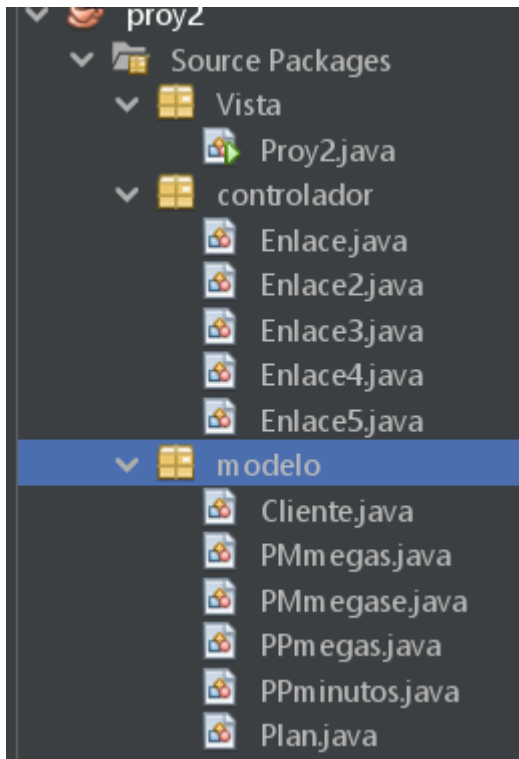
Primeramente, para poder llegar a la solución del problema planteado, comenzamos a analizar que para los planes se nos pedía atributos diferentes para cada plan, la solución que llegamos fue crear una base de datos para cada plan presentado, esto lo hicimos con el fin de no tener que alterar las variables.

Todo esto se lo hizo con el fin de crear un programa más sencillo y fácil de comprender, también tuvimos en cuenta que, para la creación del código, se creo la clase padre “Plan”, con esto nos aseguramos de que esta clase herede los demás planes junto con sus atributos y los atributos que no se necesitan se convierten en variables nulas. Por lo que esto facilita la manera de crear el código.

Cabe recalcar que utilizamos valores quemados dentro de la ejecución del código con el fin de utilizar menos recursos, con esto optimizamos el código e manera que sea mas sencillo a la hora de la codificación.

## - Codificación

Dentro de la codificación, nosotros lo dividimos en 3 paquetes. “Vista”, “Controlador” y “Modelo”. La clase Vista se centra únicamente en la clase principal, por lo que esta clase solo se encarga de la ejecución el programa. La clase Controlador solo contiene los enlaces los cuales sirven para conectar los datos quemados del programa dentro de la tabla de la base de datos que se creó en el programa SQLite. Por último, la clase “Modelo” contiene las clases que trabajan con los planes que indica el problema, junto con su clase padre llamada “Plan” la cual se mencionó anteriormente.



En este apartado de la clase principal se puede apreciar el uso que le damos a los valores quemados que se mencionaron anteriormente.

```
*/  
public static void main(String[] args) {  
    Enlace c = new Enlace();  
    Cliente m = new Cliente();  
    m.setNombre( nombre: "Samuel Reyes");  
    m.setId( id: "1150547550");  
    m.setCiudad( ciudad: "Loja");  
    m.setMarca( marca: "Samsung");  
    m.setModelo( modelo: "A54");  
    m.setNumero( numero: "09682975563");  
    m.setPago( pago: 10);  
    m.setPlan( plan: "PostPagoMinutos");  
    c.insertarCliente(m);  
    for (int i = 0; i < c.obtenerDataCliente().size(); i++) {  
        System.out.printf( format: "%s\n", args: c.obtenerDataCliente().get( index: i));  
    }  
    Enlace2 c2 = new Enlace2();  
    PMmegase m2 = new PMmegase();  
    m2.setNombre( nombre: "Samuel Reyes");  
    m2.setId( id: "1150547550");  
    m2.setMinutos( minutos: 20);  
    m2.setCostoMinutos( costoMinutos: 0);  
    m2.setMegasEnGigas( megasEnGigas: 3);  
    m2.setCostoPorGiga( costoPorGiga: 6);  
    m2.setPorcentajeDescuento( porcentajeDescuento: 10);  
}
```

En esta parte se puede apreciar la clase padre Plan la cual hereda las clases que contienen los planes que se piden en el problema

```
package modelo;

/**
 *
 * @author Samuel
 */
public class Plan {
    protected String nombre;
    protected String id;
    protected int minutos;
    protected int costoMinutos;
    protected int megasEnGigas;
    protected int costoPorGiga;

    public Plan(String nombre, String id, int minutos, int costoMinutos, int megasEnGigas, int costoPorGiga) {
        this.nombre = nombre;
        this.id = id;
        this.minutos = minutos;
        this.costoMinutos = costoMinutos;
        this.megasEnGigas = megasEnGigas;
        this.costoPorGiga = costoPorGiga;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setId(String id) {
        this.id = id;
    }

    public void setMinutos(int minutos) {
        this.minutos = minutos;
    }

    public void setCostoMinutos(int costoMinutos) {
        this.costoMinutos = costoMinutos;
    }
}
```

```
package modelo;

/**
 *
 * @author Samuel
 */
public class FMmegase extends Plan {
    private int porcentajeDescuento;

    public FMmegase(String nombre, String id, int minutos, int costoMinutos, int megasEnGigas, int costoPorGiga, int porcentajeDescuento) {
        super(nombre, id, minutos, costoMinutos, megasEnGigas, costoPorGiga);
        this.porcentajeDescuento = porcentajeDescuento;
    }

    // Constructor sin argumentos
    public FMmegase() {
        super(nombre: "", id: "", minutos: 0, costoMinutos: 0, megasEnGigas: 0, costoPorGiga: 0);
        this.porcentajeDescuento = 0;
    }

    public int getPorcentajeDescuento() {
        return porcentajeDescuento;
    }

    public void setPorcentajeDescuento(int porcentajeDescuento) {
        this.porcentajeDescuento = porcentajeDescuento;
    }
}
```

## - Resultados

```
Fun:
Ingrese nombre:
Juan
Ingrese cedula:
1105573206
Ingrese ciudad:
Loja
Ingrese marca:
samsung
Ingrese modelo:
A20
Ingrese numero:
0993568597
Ingrese pago:
1
Ingrese plan:
Connection to SQLite has been established.
Connection to SQLite has been established.
INSERT INTO Cliente (nombre, id, ciudad, marca, modelo, numero, pago, plan) VALUES ('Juan', 1105573206, 'Loja', 'samsung', 'A20', 0993568597, 1, 1);
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
modelo.Cliente@233c0b17
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
modelo.Cliente@63d4e2ba
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
modelo.Cliente@7bb11784
Connection to SQLite has been established.
Connection to SQLite has been established.
=====MEN=====
Elija que tipo de plan movil que desea:
```

```
Elija que tipo de plan movil que desea:
Plan Post Pago Minutos Megas Economico      [1]
Plan Post Pago Minutos                      [2]
Plan Post Pago Megas                        [3]
Plan Post Pago Minutos Megas                [4]
Salir                                         [0]
2
Connection to SQLite has been established.
Connection to SQLite has been established.
INSERT INTO PPminutos (nombre, id, minutosNacionales, minutosInternacionales, minutosExtranjeros, minutosNacionales, minutosInternacionales, minutosExtranjeros) VALUES ('Juan', 1105573206, 150, 0, 0, 150, 0, 0)
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
modelo.PPminutos@47fd17e3
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
modelo.PPminutos@7cdbc5d3
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
Connection to SQLite has been established.
modelo.PPminutos@3aa9e816
Connection to SQLite has been established.
Connection to SQLite has been established.

===== Factura =====
Nombre: Juan
ID: 1105573206
Tipo de Plan:
Costo Total del plan: $150
=====
```

