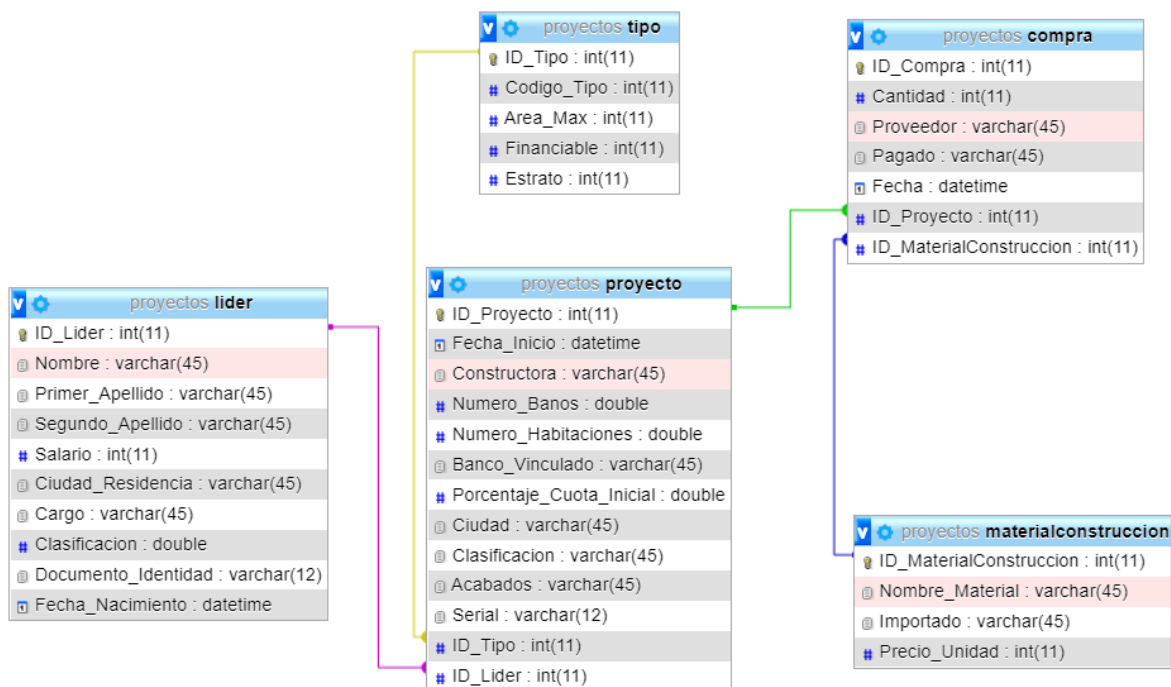




Ciclo 2 Programación Básica

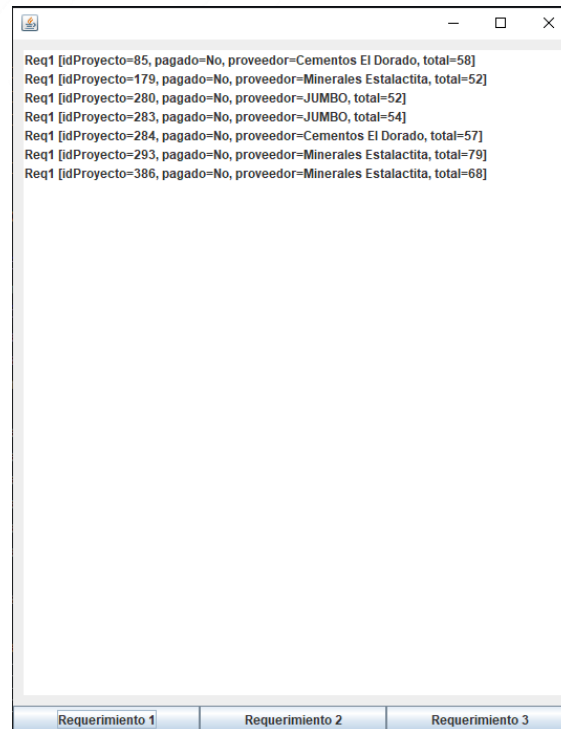
Reto 5

La alcaldía del municipio Rosita Rocera está evaluando los proyectos y la cartera a fin de determinar la viabilidad de muchos de estos, por lo cual necesita determinar, lo siguiente:

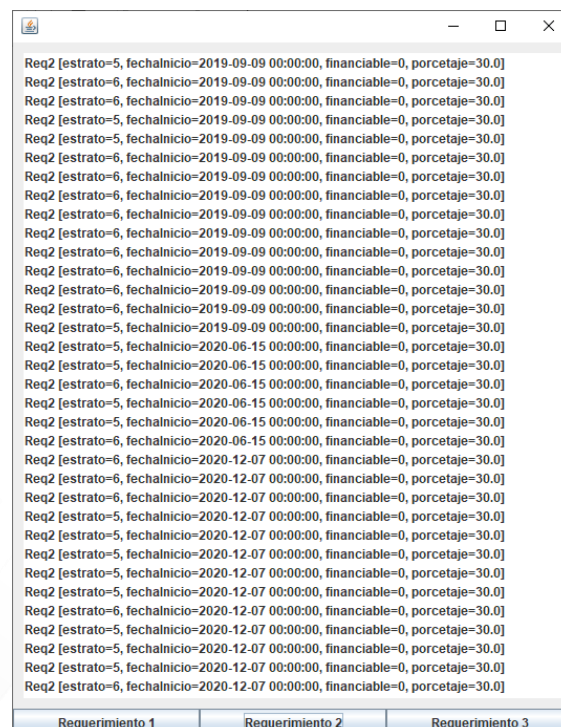


- 1) Con el fin de determinar las acciones a tomar de acuerdo a la cartera del año en curso, se están evaluando los proyectos de acuerdo a su información.
- 2) Para la evaluación es necesario contar con la información necesaria por esto realice las consultas pertinentes para obtener la siguiente información de la base de datos:

- Proyectos que no hayan sido pagados, cuyo total (suma) en la cantidad de compra supere los 50, agrupe esta consulta por el id del proyecto.

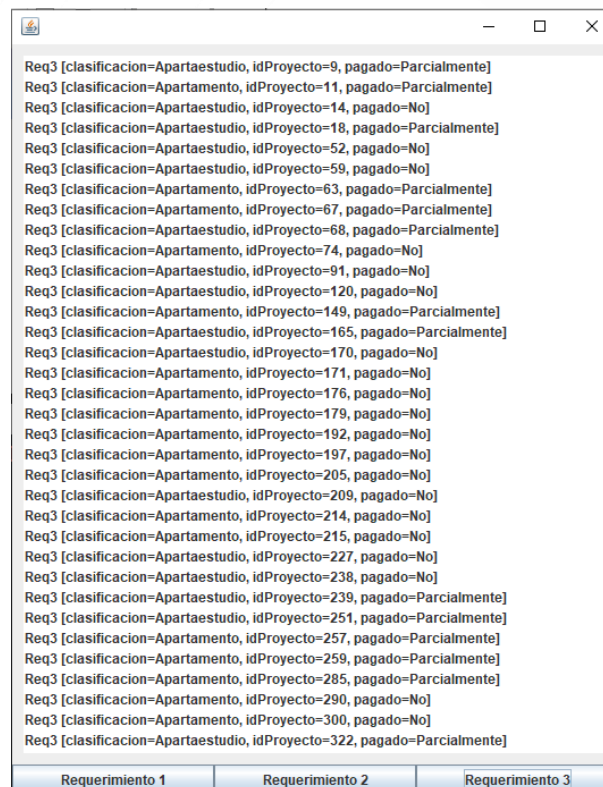


- Proyectos iniciados en los últimos años ordenados desde el más antiguo al más reciente cuyos estratos sean mayores o iguales a 5, los cuales no sean financiados, pero el porcentaje de cuota inicial sea mayor al 30%





- Proyectos de apartamentos o aparta-estudios no pagados, o pagados parcialmente que dentro de sus compras tengan material de construcción importado y que no se entreguen acabados



Presentación del reto

Tienes la libertad creativa para realizar un diseño GUI para dicho requisito, y para evaluar este reto es necesario que realices lo siguiente:

1. Compartir el repositorio de GitHub con el código de la aplicación
2. Debes realizar un video corto mostrando la aplicación en funcionamiento. Puedes realizar este video en herramientas como:
 - a. Loom: (<https://www.loom.com/>) **Recomendado.**
 - b. Screencast-O-Matic: (<https://screencast-o-matic.com/>)
 - c. Screencastify: (<https://www.screencastify.com/>)
 - d. Online Screen Recorder (<https://screenapp.io/>)
 - e. Screen Capture (<https://www.screencapture.com/>)

Este video debe ser publicado (por ejemplo, en youtube) y compartido para poder ser visualizado en línea para el proceso de evaluación.

Esqueleto

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;
```



```
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

public class Req1 {
    private int total;
    private int idProyecto;
    private String pagado;
    private String proveedor;

    public Req1(int total, int idProyecto, String pagado, String proveedor)
    {
        //TODO inicializar los atributos usando los parametros
    }

    //TODO crear el toString usando el autogenerador, (click derecho-
    >source action->generate toString )
}

public class Req2 {
    private String fechaInicio;
    private int financiable;
    private int estrato;
    private double porcetaje;

    public Req2(String fechaInicio, int financiable, int estrato, double por
    cetaje) {
        //TODO inicializar los atributos usando los parametros
    }

    //TODO crear el toString usando el autogenerador, (click derecho-
    >source action->generate toString )
}

public class Req3 {
    private int idProyecto;
    private String pagado;
    private String clasificacion;

    public Req3(int idProyecto, String pagado, String clasificacion) {
        //TODO inicializar los atributos usando los parametros
    }

    //TODO crear el toString usando el autogenerador, (click derecho-
    >source action->generate toString )
}
```



```
}

public class Conector {

    private Connection connect;

    public Conector(String url) {
        super();
        connectDB(url);
    }

    public ResultSet getReq1DB() {
        //TODO
        String sql = "Escribe aquí la Primera query";
        return executeQuery(sql);
    }

    public ResultSet getReq2DB() {
        //TODO
        String sql = "Escribe aquí la Segunda query";
        return executeQuery(sql);
    }

    public ResultSet getReq3DB() {
        //TODO
        String sql = "Escribe aquí la Tercera query";
        return executeQuery(sql);
    }

    public Connection connectDB(String url) {
        try {
            connect = DriverManager.getConnection(url, "root", "");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return connect;
    }

    public ResultSet executeQuery(String sql) {
        ResultSet rs = null;
        try {
            Statement stmt = connect.createStatement();
            rs = stmt.executeQuery(sql);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```



```
    }  
    return rs;  
  }  
}  
  
public class Controller {  
  
    private ArrayList<String> req1s;  
    private ArrayList<String> req2s;  
    private ArrayList<String> req3s;  
  
    private Conector conectorDB;  
  
    public Controller(String url) {  
        super();  
        //TODO inicializar las listas de reqs  
        conectorDB = new Conector(url);  
    }  
  
    public ArrayList<String> getReq1s() {  
        return req1s;  
    }  
  
    public ArrayList<String> getReq2s() {  
        return req2s;  
    }  
  
    public ArrayList<String> getReq3s() {  
        return req3s;  
    }  
  
    public void req1() {  
        //TODO logica de llenado de la lista req1s  
    }  
  
    public void req2() {  
        //TODO logica de llenado de la lista req2s  
    }  
  
    public void req3() {  
        //TODO logica de llenado de la lista req3s  
    }  
}
```



```
public class View extends JFrame implements ActionListener {

    //TODO crear 3 botones

    //TODO crear una lista JList<String>

    //TODO crear un atributo de Controller

    public View(String url) {
        super();
        setSize(550, 700);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        //TODO inicializar el controlador, pasando la url por parametro

        //TODO inicializar los botones

        //TODO inicializar lista

        JPanel panelBotones = new JPanel();
        panelBotones.setLayout(new GridLayout());
        //TODO agregar al panelBotones los 3 botones

        JPanel panelCentro = new JPanel();
        panelCentro.setLayout(new GridLayout());
        panelCentro.setBorder(new EmptyBorder(10, 10, 10, 10));
        //TODO agregar al panelCentro la lista

        add(panelCentro, BorderLayout.CENTER);
        add(panelBotones, BorderLayout.SOUTH);
    }

    public void req1(ArrayList<String> listadoReq1) {
        //TODO crear y llenar la lista de datos a partir del ArrayList parametro
    }

    public void req2(ArrayList<String> listadoReq2) {
        //TODO crear y llenar la lista de datos a partir del ArrayList parametro
    }

    public void req3(ArrayList<String> listadoReq3) {
```



```
//TODO crear y llenar la lista de datos a partir del ArrayList param  
etro  
}  
  
@Override  
public void actionPerformed(ActionEvent e) {  
    if (e.getActionCommand().equals(btnReq1.getActionCommand())) {  
        //TODO ejecutar el requerimiento 1 desde el controlador  
        //TODO ejecutar el req1 de la clase  
    }  
    if (e.getActionCommand().equals(btnReq1.getActionCommand())) {  
        //TODO ejecutar el requerimiento 1 desde el controlador  
        //TODO ejecutar el req1 de la clase  
    }  
    if (e.getActionCommand().equals(btnReq1.getActionCommand())) {  
        //TODO ejecutar el requerimiento 1 desde el controlador  
        //TODO ejecutar el req1 de la clase  
    }  
}  
}  
  
//Pruebas unicamente  
public class App {  
    public static void main(String[] args) throws Exception {  
        //Pruebas locales  
        View principal = new View("jdbc:mysql://localhost:3306/proyectos");  
        // View principal = new View("jdbc:sqlite:ProyectosConstruccion.db")  
;  
        principal.setVisible(true);  
    }  
}
```