

Ciclo 2. Programación básica

Reto 4 – Requerimiento extracción y visualización de la información de una base de datos

Descripción del problema

El Ministerio de vivienda ha quedado muy satisfecho por el trabajo realizado en su última tarea (reto 3). ¡Apreciado tripulante!, ahora el ministerio requiere que usted realice una aplicación que extraiga la información de la base de datos y además presente en pantalla los resultados de la información solicitada.

Aspectos a tener en cuenta

¡Tripulante! debe crear una aplicación (usando la consola), que muestre los resultados de las consultas de información solicitadas por el ministerio. Adicionalmente, debe usar una arquitectura MVC con conexión a la base de datos utilizando JDBC.

La estructura de la base de datos se presenta a continuación:

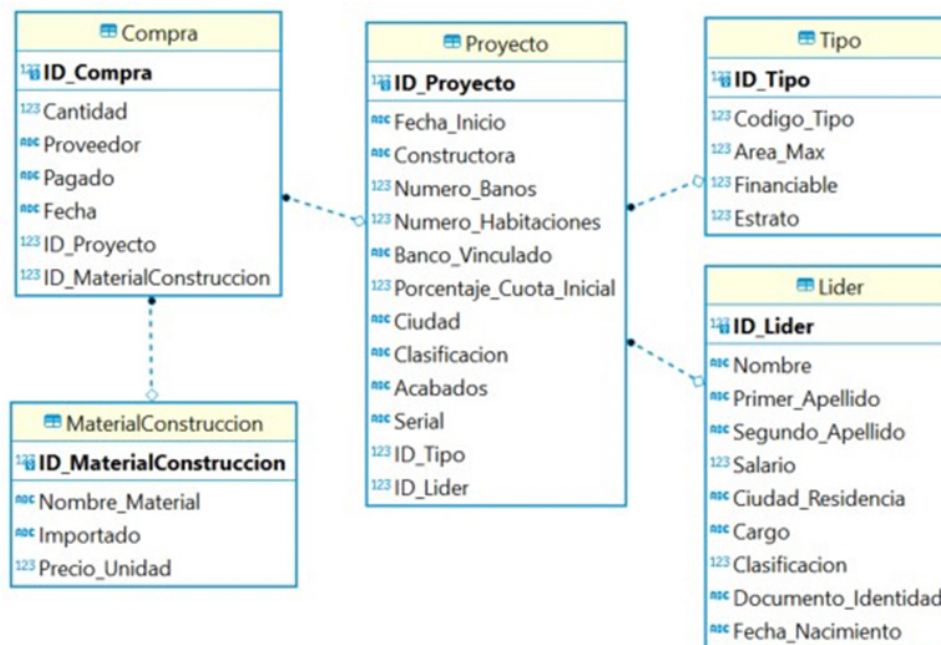


Figura 1. Estructura de la base de datos.

La estructura del proyecto y la ubicación de las clases a crear se presentan en la figura 2:

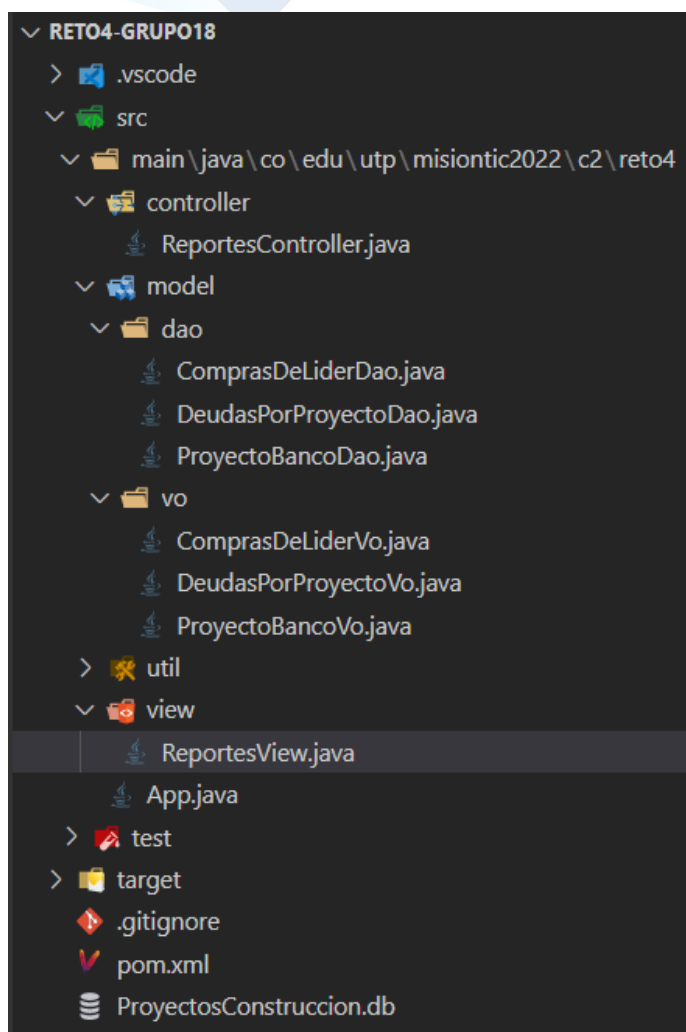


Figura 2. Estructura del proyecto y ubicación de las clases.

Información requerida

La información a consultar desde la aplicación es la siguiente:

1. Información de los proyectos financiados por un banco determinado (consulta 2 del reto 3).

Para cumplir con el requerimiento 1, se debe recibir como parámetro el **nombre del banco** que se desea consultar y en pantalla imprimir la información de salida. Esta información se debe presentar en el siguiente orden: id, constructora, ciudad, clasificación, estrato y líder.

Por ejemplo:

Entrada			
<pre>var reportesView = new ReportesView(); var banco = "Conavi"; reportesView.proyectosFinanciadosPorBanco(banco); ;</pre>			
Salida			
===== LISTADO DE PROYECTOS POR BANCO =====			
ID CONSTRUCTORA	CIUDAD	CLASIFICACION	ESTRATO LIDER
8 Hermanos Gallego	Armenia	Condominio	6 Carla Rodriguez Andrade
96 Pegaso	Armenia	Casa	3 Nadia Pelaez Galvis
46 Edificios y Edificios	Barranquilla	Casa	2 Carla Gomez Parra
91 Hermanos Gallego	Barranquilla	Apartaestudio	6 Homero Caceres Rojas
321 Arquitectura S.A.	Bucaramanga	Casa Campestre	2 Nadia Lopez Ortiz
202 Hermanos Gallego	Bucaramanga	Apartaestudio	6 Luis Barrera Parra
308 Ingenieros Ltda.	Bucaramanga	Apartaestudio	3 Bart Rivera Ortiz
70 Arquitectura S.A.	Cartagena	Casa	5 Homero Falcon Godoy
252 Edificios y Edificios	Cartagena	Apartaestudio	6 Beatriz Jimenez Ardila
253 Hermanos Gallego	Cartago	Casa	6 Homero Perez Parra
75 Pegaso	Ibague	Apartamento	5 Jorge Caceres Ortiz
368 Arquitectura S.A.	Neiva	Apartaestudio	6 Carla Gomez Parra
378 Arquitectura S.A.	Neiva	Condominio	5 Libia Pelaez Rojas
163 Arquitectura S.A.	Quibdo	Apartamento	5 Libia Pelaez Rojas
82 Hermanos Gallego	Quibdo	Casa	6 Homero Caceres Rojas
190 Hermanos Gallego	Quibdo	Casa	3 Homero Falcon Godoy
350 Ingenieros Ltda.	Quibdo	Casa Campestre	5 Juana Perez Galvis

Ayuda:

String.format("%3d %-25s %-20s %-15s %7d %-30s", id, constructora, ciudad, clasificacion, estrato, lider);

2. Listado del total adeudado en cada proyecto, filtrado por un límite inferior dado (consulta 4 del reto 3).

Para esta consulta, se necesita conocer el total adeudado (comprado y no pagado) por cada uno de los proyectos que superen un valor determinado. Este valor se tomará como parámetro de la aplicación. La información se debe presentar en el siguiente orden: id, valor.

Por ejemplo:

Entrada	
<pre>var reportesView = new ReportesView(); var limiteInferior = 50_000d; reportesView.totalAdeudadoPorProyectosSuperioresALimite(limiteInferior);</pre>	
Salida	
= TOTAL DEUDAS POR PROYECTO =	
ID	VALOR
293	151.973,0
386	117.302,0
283	112.077,0
387	100.400,0
179	96.158,0
85	94.629,0
280	93.170,0
351	87.441,0

Ayuda:

```
String.format("%3d %,15.1f", id, valor);
```

3. Listado del top 10 de los líderes que más dinero gastan en sus proyectos para la compra de materiales (consulta 5 del reto 3).

Para esta consulta, se necesita conocer el valor total de compras realizadas por los líderes en todos los proyectos registrados. Tenga en cuenta que en este listado solo se deben presentar los 10 líderes que más dinero han gastado. La información se debe presentar en el siguiente orden: líder, valor. Por ejemplo:

Entrada	
<pre>var reportesView = new ReportesView(); reportesView.lideresQueMasGastan();</pre>	
Salida	
<pre>===== 10 LIDERES MAS COMPRADORES ===== LIDER VALOR ----- Maradona Jimenez Godoy 963.439,0 Beatriz Jimenez Ardila 963.291,0 Carla Barrera Cuadrado 935.764,0 Nadia Pelaez Galvis 828.636,0 Libia Pelaez Rojas 814.943,0 Libia Pelaez Hernandez 795.783,0 Juana Perez Galvis 769.053,0 ...</pre>	

Ayuda:

```
String.format("%-25s %,15.1f", lider, valor);
```

Esqueleto

Paquete **controller**:

```
public class ReportesController {
    // TODO Implementar la clase
}
```

Paquete **model.dao**:

```
public class ComprasDeLiderDao {
    // TODO Implementar la clase
}
```

```
public class DeudasPorProyectoDao {
    // TODO Implementar la clase
}
```

```
public class ProyectoBancoDao {
    // TODO Implementar la clase
}
```

Paquete **model.vo**:

```
public class ComprasDeLiderVo {  
    // TODO Implementar la clase  
}
```

```
public class DeudasPorProyectoVo {  
    // TODO Implementar la clase  
}
```

```
public class ProyectoBancoVo {  
    // TODO Implementar la clase  
}
```

Paquete **util**:

```
public class JDBCUtilities {  
    // Atributos de clase para gestión de conexión con la base de datos  
    private static final String UBICACION_BD = "ProyectosConstruccion.db";  
  
    public static Connection getConnection() throws SQLException {  
        String url = "jdbc:sqlite:" + UBICACION_BD;  
        return DriverManager.getConnection(url);  
    }  
}
```

Paquete **view**:

```
public class ReportesView {  
    private String repitaCaracter(Character caracter, Integer veces) {  
        String respuesta = "";  
        for (int i = 0; i < veces; i++) {  
            respuesta += caracter;  
        }  
        return respuesta;  
    }  
  
    public void proyectosFinanciadosPorBanco(String banco) {  
        System.out.println(repitaCaracter('=', 36) + " LISTADO DE PROYECTOS POR BANCO "  
            + repitaCaracter('=', 37));  
        if (banco != null && !banco.isBlank()) {  
            System.out.println(String.format("%3s %-25s %-20s %-15s %-7s %-30s",  
                "ID", "CONSTRUCTORA", "CIUDAD", "CLASIFICACION", "ESTRATO", "LIDER"));  
            System.out.println(repitaCaracter('-', 105));  
  
            // TODO Imprimir en pantalla la información del proyecto  
        }  
    }  
  
    public void totalAdeudadoPorProyectosSuperioresALimite(Double limiteInferior) {  
        System.out.println(repitaCaracter('=', 1) + " TOTAL DEUDAS POR PROYECTO "  
            + repitaCaracter('=', 1));  
        if (limiteInferior != null) {  
            System.out.println(String.format("%3s %15s", "ID", "VALOR "));  
            System.out.println(repitaCaracter('-', 29));  
  
            // TODO Imprimir en pantalla la información del total adeudado  
        }  
    }  
  
    public void lideresQueMasGastan() {
```

```
System.out.println(repitaCaracter('=', 6) + " 10 LIDERES MAS COMPRADORES "
    + repitaCaracter('=', 7));
System.out.println(String.format("%-25s %15s", "LIDER", "VALOR  "));
System.out.println(repitaCaracter('-', 41));

    // TODO Imprimir en pantalla la información de los líderes
}
}
```

Nota 1: para obtener la información requerida en los tres ítems (mencionados anteriormente), usted tripulante debe plantear las respectivas consultas en la base de datos “ProyectosConstruccion.db” . Esta base de datos la puede descargar desde la plataforma iMaster.

Nota 2: en este ejercicio tripulantes, tienen la libertad de llamar a los métodos de la manera que deseen, solo hay que respetar el nombre de la clase ReportesView, ya que esta se utiliza en las pruebas a desarrollar en la plataforma iMaster.

TRIPULANTE

¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 4!

Hechos

QUE

CONECTAN



#MisiónTIC2022