



Ciclo 2 Fundamentos de programación

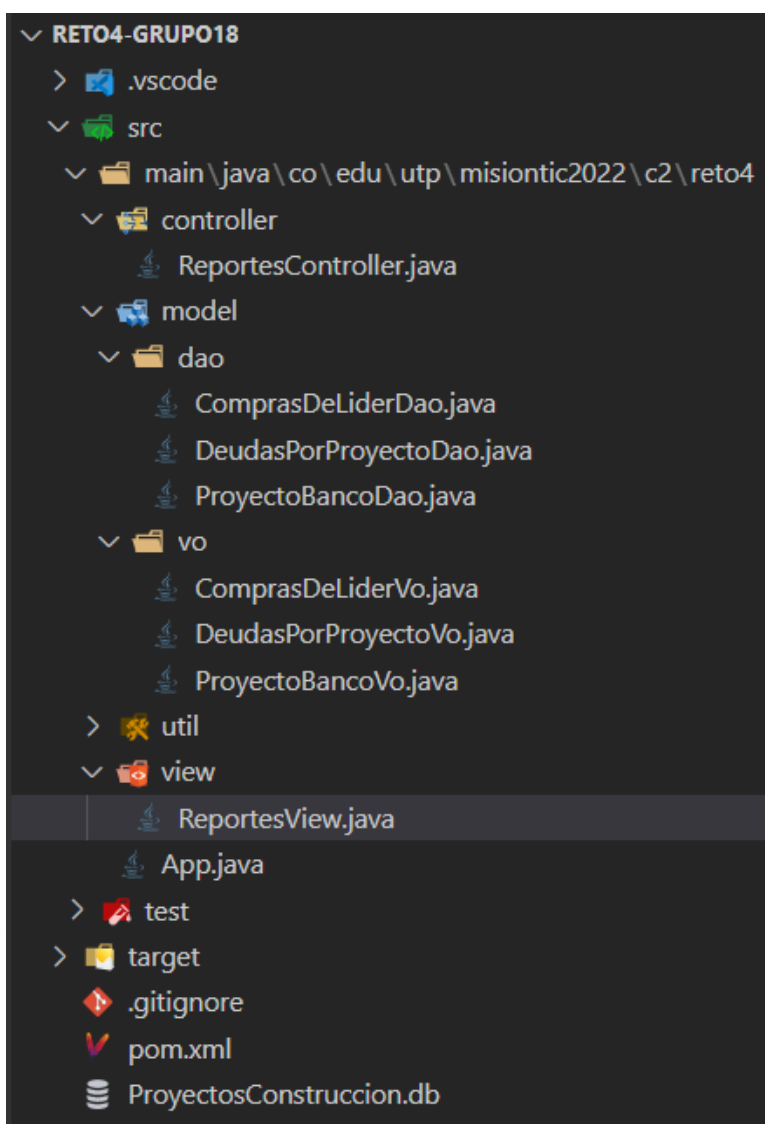
Reto 4 – Grupo 18

Descripción del problema:

El Ministerio de Vivienda ha quedado muy satisfecho por el trabajo realizado por usted en su última tarea (Reto 3) y quiere que usted realice una aplicación que extraiga la información de la base de datos y le muestre en pantalla los resultados de 3 de las consultas que anteriormente fueron realizadas.

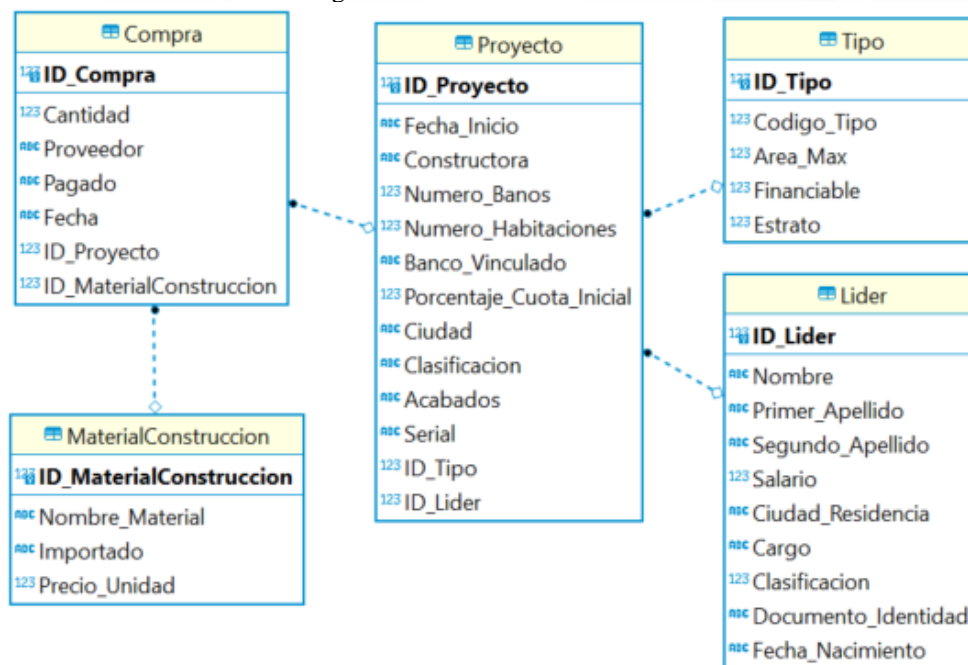
Usted debe crear usando una arquitectura MVC y con conexión a la base de datos usando JDBC, realizar una aplicación que, usando la consola, muestra los resultados de las consultas solicitadas.

La estructura del proyecto y la ubicación de las clases a crear debe ser la siguiente:





La estructura de la base de datos es la siguiente:



Las consultas que se quieren consultar desde la aplicación son las siguientes:

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

Se debe poder recibir como parámetro el **nombre del banco** que se quiera comprobar. A pesar que la consulta anterior estaba para un banco específico.

En pantalla debe imprimir la información de dicha consulta con el formato de cada línea de detalle es la siguiente:

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

Ejemplo 1:

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



116	Pegaso	Quibdo	Condominio	5	Maradona Jimenez Godoy
113	Ingenieros Ltda.	Santa Marta	Apartaestudio	2	Juana Falcon Rojas
130	Ingenieros Ltda.	Bogota	Casa Campestre	6	Paul Escobar Ortiz
30	Arquitectura S.A.	Cartagena	Casa	2	Jorge Gomez Velez
112	Ingenieros Ltda.	Cartago	Apartamento	6	Juana Rodriguez Velez
275	Arquitectura S.A.	Ibague	Apartamento	2	Juana Perez Galvis
1	Arquitectura S.A.	Manizales	Casa Campestre	5	Libia Pelaez Rojas
259	Arquitectura S.A.	Monteria	Apartamento	3	Homero Falcon Godoy
301	Arquitectura S.A.	Monteria	Apartaestudio	5	Diana Arteaga Rojas
261	Ingenieros Ltda.	Monteria	Casa Campestre	6	Monica Rivera Hernandez
327	Edificios y Edificios	Pereira	Condominio	6	Diana Rivera Arias
175	Ingenieros Ltda.	Pereira	Apartaestudio	3	Moe Falcon Cardenas
58	Pegaso	Pereira	Casa	2	Jorge Caceres Ortiz
333	Ingenieros Ltda.	Quibdo	Apartaestudio	5	Paul Escobar Ortiz
88	Edificios y Edificios	Santa Marta	Casa	6	Jorge Gomez Federer
337	Edificios y Edificios	Sta. Rosa de Cabal	Condominio	6	Juana Ocampo Velez
285	Ingenieros Ltda.	Sta. Rosa de Cabal	Apartamento	5	Homero Caceres Rojas
69	Edificios y Edificios	Armenia	Condominio	3	Moe Falcon Cardenas
150	Arquitectura S.A.	Barranquilla	Casa	6	Carlos Rodriguez Palacio
246	Hermanos Gallego	Bucaramanga	Condominio	3	Juana Rodriguez Velez
40	Pegaso	Bucaramanga	Casa	5	Nadia Pelaez Galvis
353	Ingenieros Ltda.	Cartagena	Apartaestudio	6	Maradona Jimenez Godoy
21	Edificios y Edificios	Ibague	Casa	2	Homero Perez Parra
356	Edificios y Edificios	Manizales	Condominio	6	Jack Escobar Cuadrado
213	Pegaso	Manizales	Apartaestudio	6	Nadia Lopez Ortiz
119	Pegaso	Neiva	Casa Campestre	6	Moe Falcon Cardenas
94	Edificios y Edificios	Pereira	Casa	3	Jorge Gomez Velez
267	Ingenieros Ltda.	Pereira	Casa	6	Carlos Camargo Arias
17	Arquitectura S.A.	Quibdo	Apartaestudio	3	Carlos Jimenez Federer
339	Pegaso	Quibdo	Apartamento	6	Carlos Jimenez Federer
298	Pegaso	Santa Marta	Casa Campestre	5	Maradona Arteaga Figueroa
284	Hermanos Gallego	Sta. Rosa de Cabal	Casa	5	Maradona Jimenez Godoy
31	Ingenieros Ltda.	Sta. Rosa de Cabal	Condominio	2	Jorge Gomez Federer
347	Pegaso	Barranquilla	Condominio	3	Carlos Linares Figueroa
398	Hermanos Gallego	Bucaramanga	Apartaestudio	2	Bart Linares Federer
330	Ingenieros Ltda.	Bucaramanga	Apartamento	3	Luis Barrera Parra
173	Arquitectura S.A.	Cartago	Condominio	6	Bart Linares Federer
41	Ingenieros Ltda.	Manizales	Casa Campestre	3	Libia Perez Hernandez
227	Arquitectura S.A.	Monteria	Apartaestudio	6	Nadia Ocampo Godoy
374	Edificios y Edificios	Monteria	Apartamento	3	Jorge Caceres Ortiz
108	Pegaso	Monteria	Apartaestudio	3	Beatriz Jimenez Ardila
105	Edificios y Edificios	Neiva	Casa	6	Homero Falcon Godoy
249	Edificios y Edificios	Pereira	Apartaestudio	6	Homero Caceres Rojas
218	Arquitectura S.A.	Santa Marta	Condominio	6	Carla Linares Velez
358	Arquitectura S.A.	Santa Marta	Casa Campestre	6	Carlos Rodriguez Palacio
241	Ingenieros Ltda.	Sta. Rosa de Cabal	Condominio	5	Carlos Linares Figueroa

2. Listado del total adeudado de cada proyecto filtrado por un límite inferior dado (consulta 4 del Reto 3)

Para esta otra consulta, se necesita conocer el total adeudado (comprado y no pagado) por cada uno de los proyectos que superen un valor dado. Este valor se tomará como parámetro de la aplicación.

El formato de salida de la lista de datos es: `String.format("%3d %15.1f", id, valor);`

Ejemplo 2:

Entrada	
<pre>var reportesView = new ReportesView(); var limiteInferior = 50_000d; reportesView.totalAdeudadoPorProyectosSuperioresALimite(limiteInferior);</pre>	
Salida	
<pre>= 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</pre>	



284	79.774,0
117	76.880,0
295	73.657,0
331	72.921,0
224	66.640,0
202	66.304,0
336	65.563,0
350	65.103,0
87	64.015,0
296	63.649,0
16	63.432,0
393	61.388,0
93	60.382,0
134	60.058,0
48	58.830,0
341	58.258,0
187	57.880,0
5	57.856,0
133	57.426,0
154	56.576,0
52	56.446,0
315	56.294,0
118	55.630,0
146	55.568,0
116	55.556,0
46	55.274,0
61	55.234,0
22	54.615,0
158	54.565,0
343	54.562,0
358	54.259,0
12	54.206,0
288	53.997,0
171	52.739,0
301	51.461,0
390	51.454,0
338	51.392,0
167	51.068,0
53	50.716,0
148	50.575,0
356	50.402,0

3. El top 10 de los líderes que más gastan en sus proyectos, ósea, los que compran más materiales para cumplir con sus proyectos. (consulta 5 del Reto 3).

Este devuelve el total de compras realizadas por los líderes en todos los proyectos registrados. Solo retornar los 10 líderes que más gastan.

El formato de salida de la lista de datos es: `String.format("%-25s %15.1f", lider, valor);`

Ejemplo 3:

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>	



Homero Caceres Rojas	730.407,0
Maradona Arteaga Figueroa	699.268,0
Paul Jimenez Parra	690.199,0

Para obtener esta información usted debe plantear las respectivas consultas sobre la base de datos **ProyectosConstruccion.db** que debe descargar desde iMaster (la misma usada en el Reto 3).

Esqueleto:

En este ejercicio tienen la libertad de llamar los métodos de la manera que deseen, solo hay que respetar los nombres de la clase ReportesView que son las que se usarán en las pruebas en iMaster.

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 character, Integer veces) {  
        var respuesta = "";  
        for (int i = 0; i < veces; i++) {  
            respuesta += character;  
        }  
        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  
}  
}
```