

Programación 3

Conexión a base de datos (back-end en Java)

Dr. Andrés Melgar

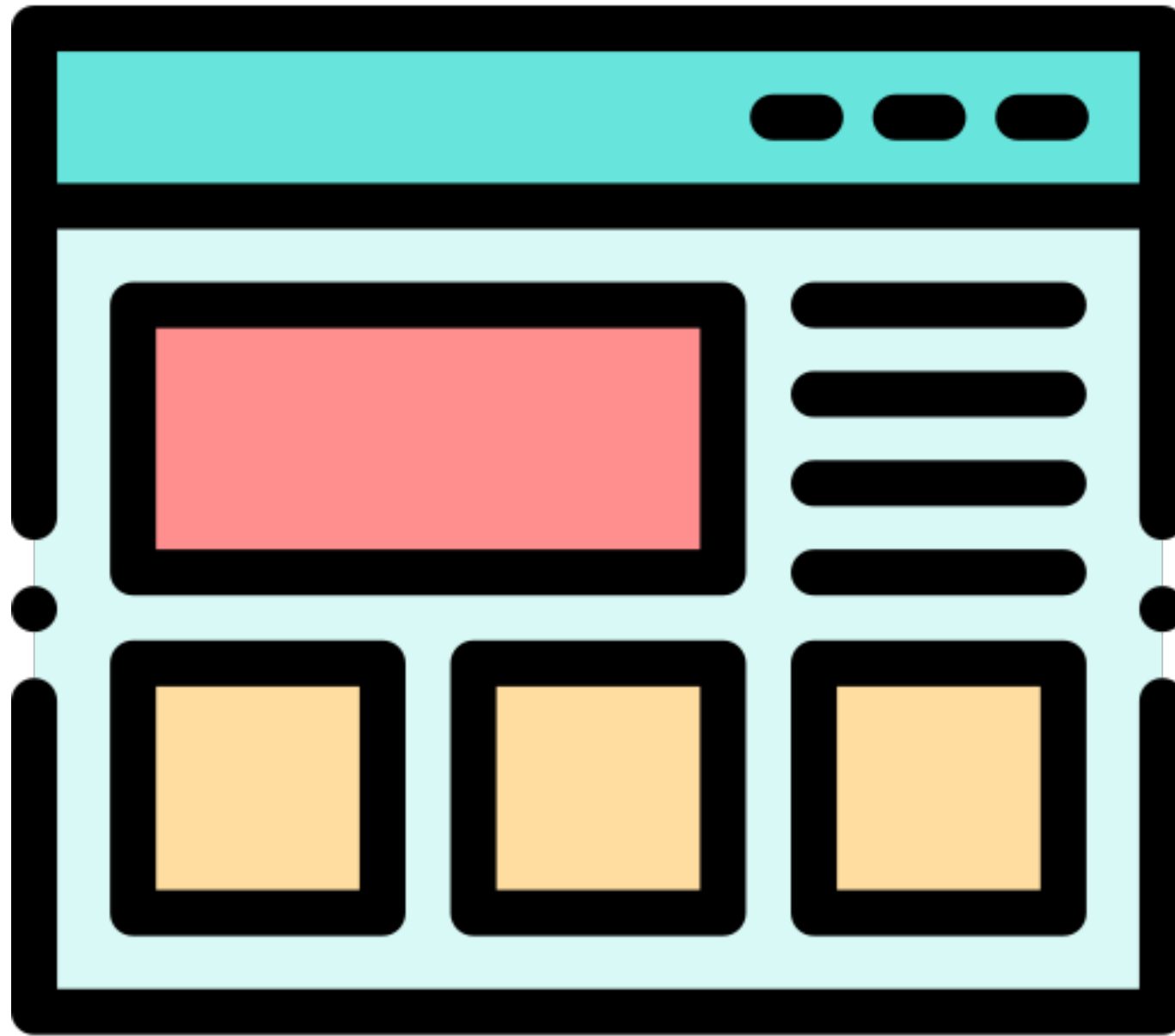
Cliente/Servidor con base de datos

Estilo Arquitectónico por Capas



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

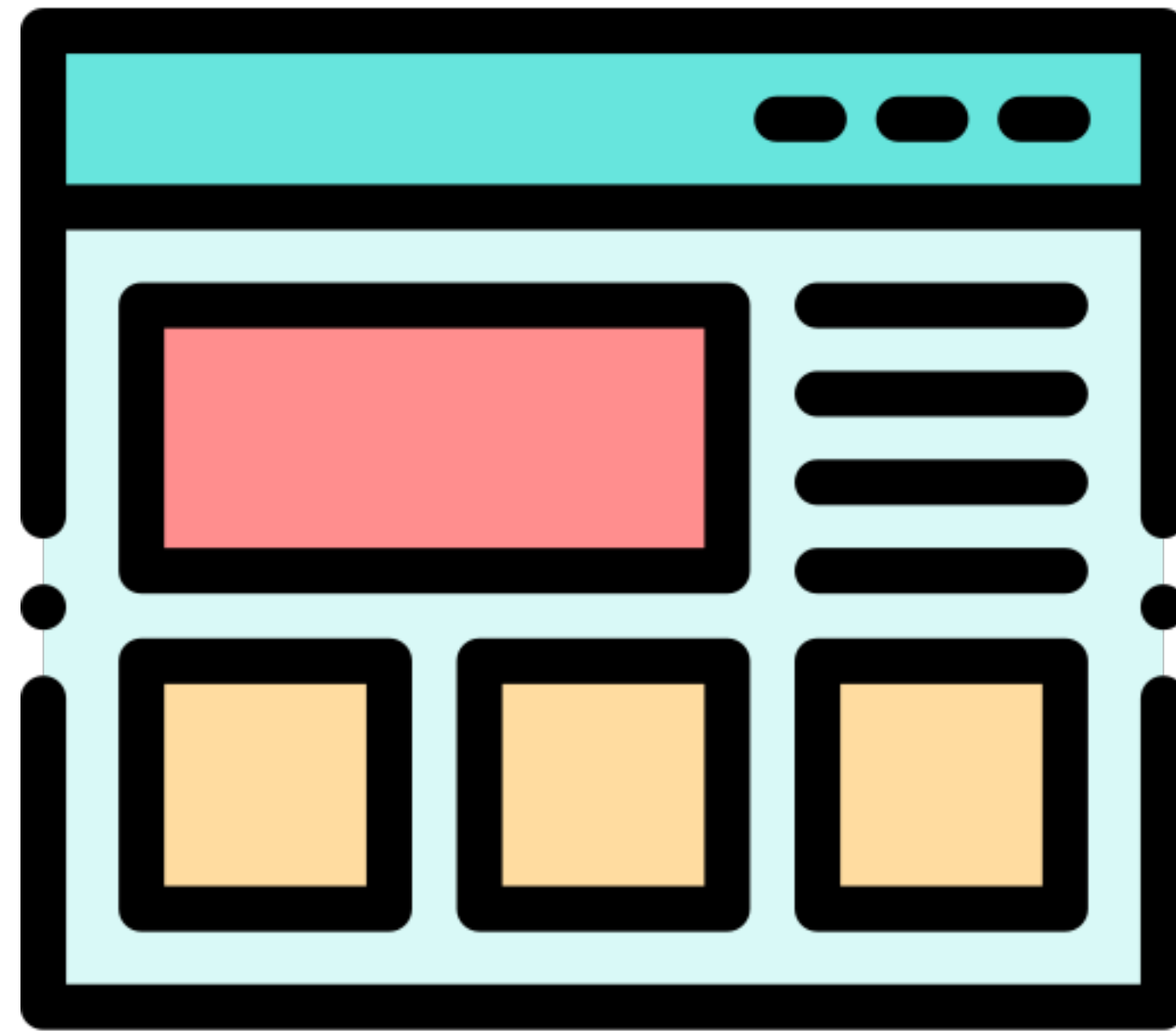


Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

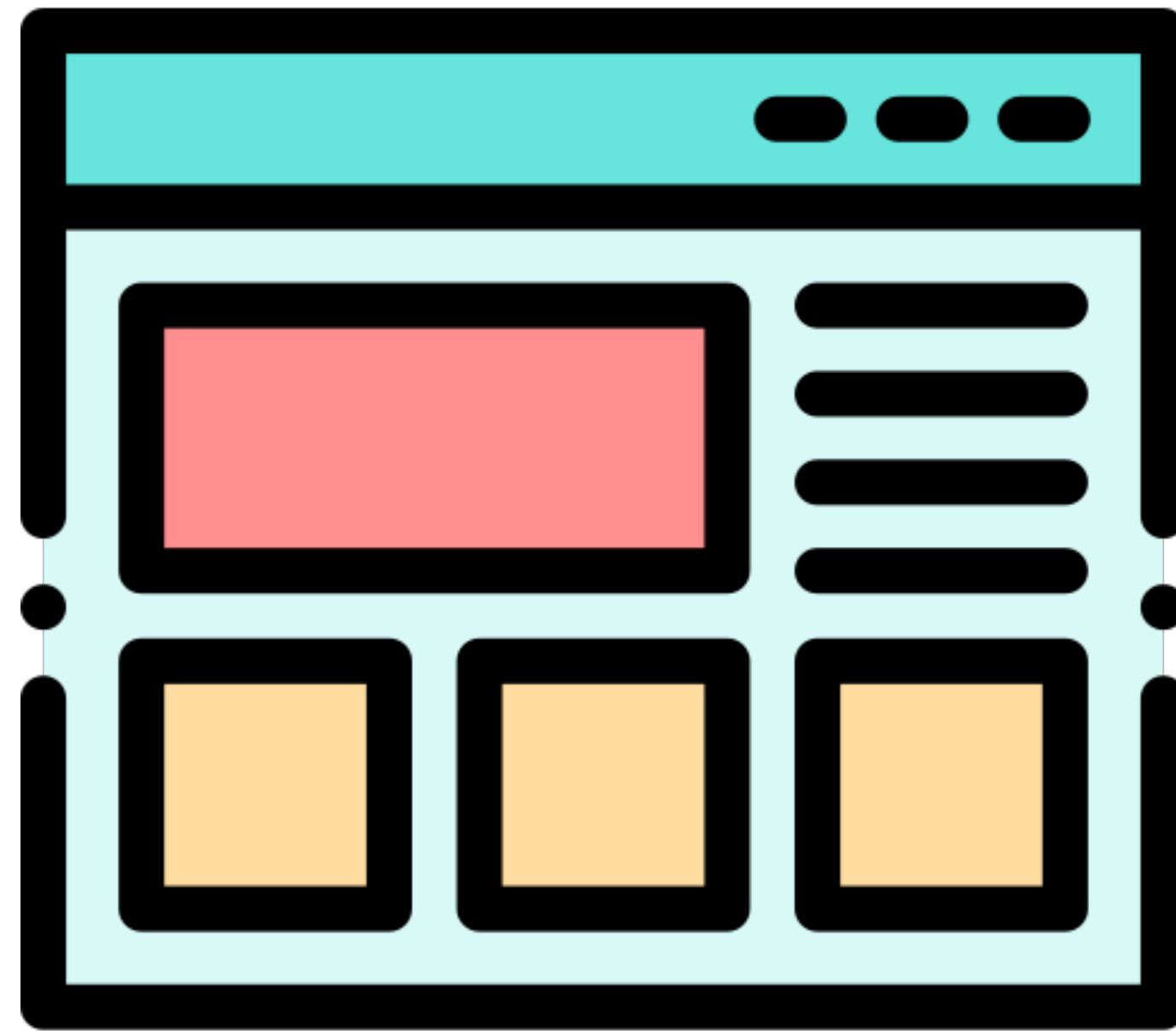


Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

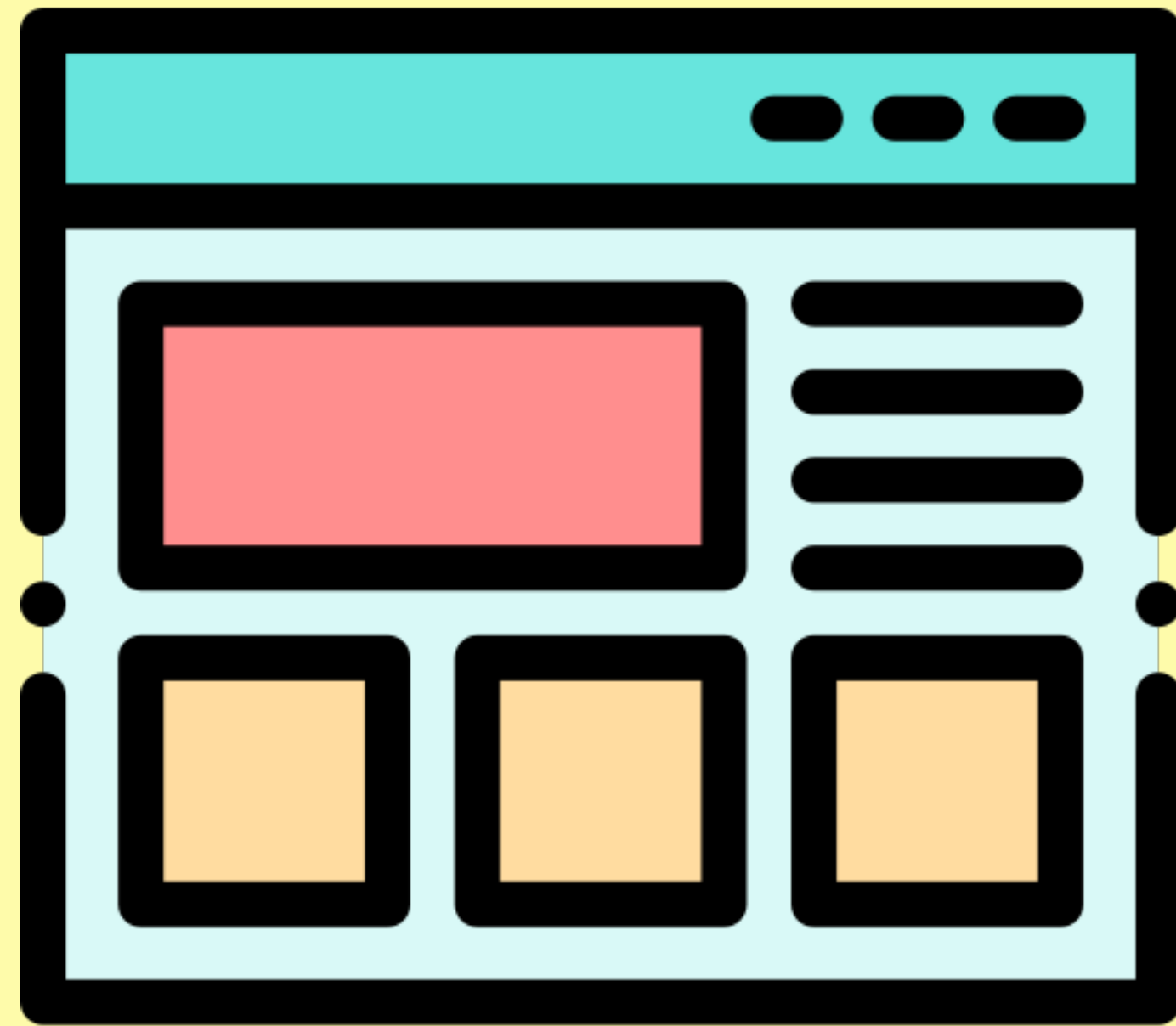


Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

Petición

Respuesta

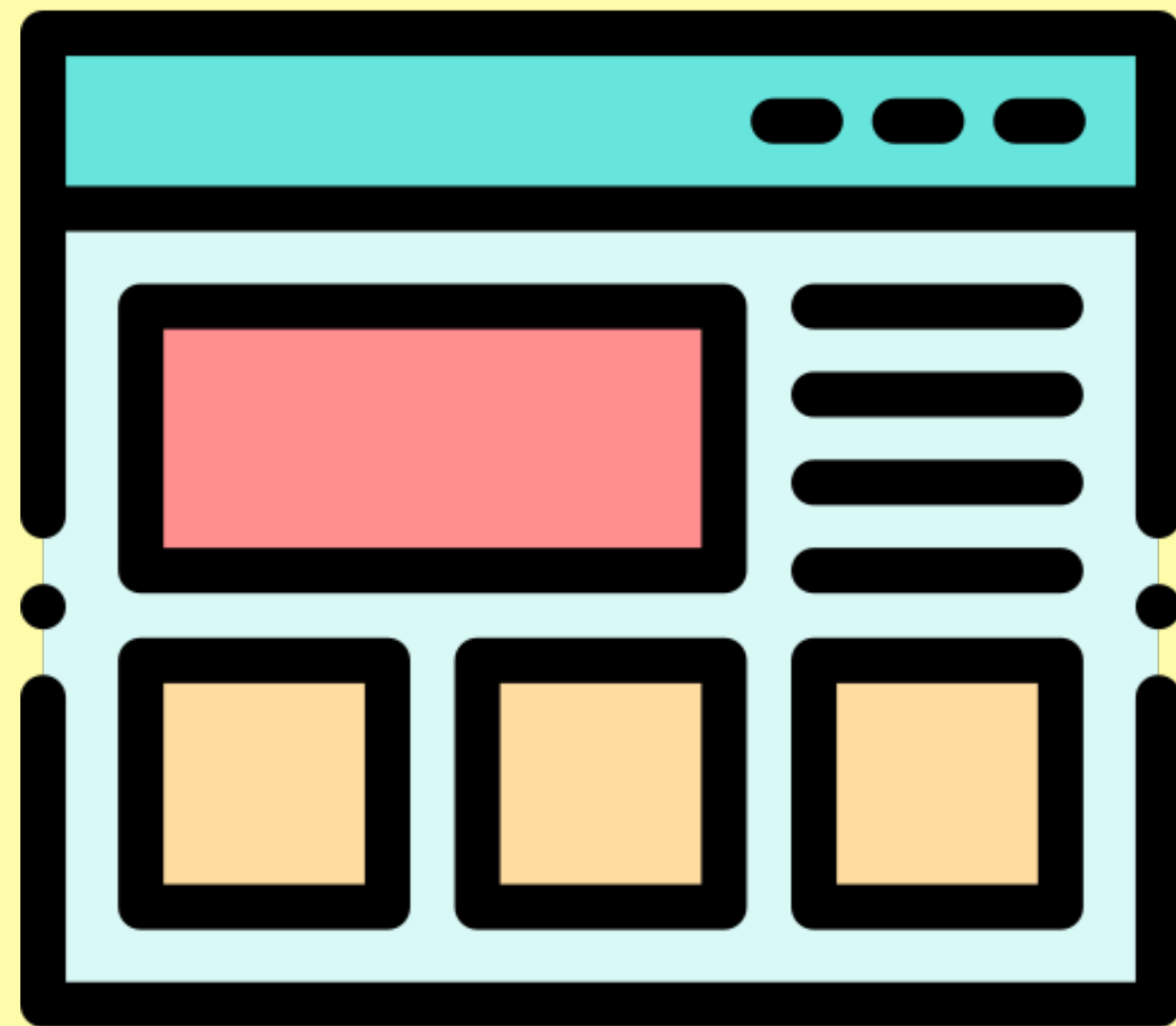


Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

Petición

Respuesta



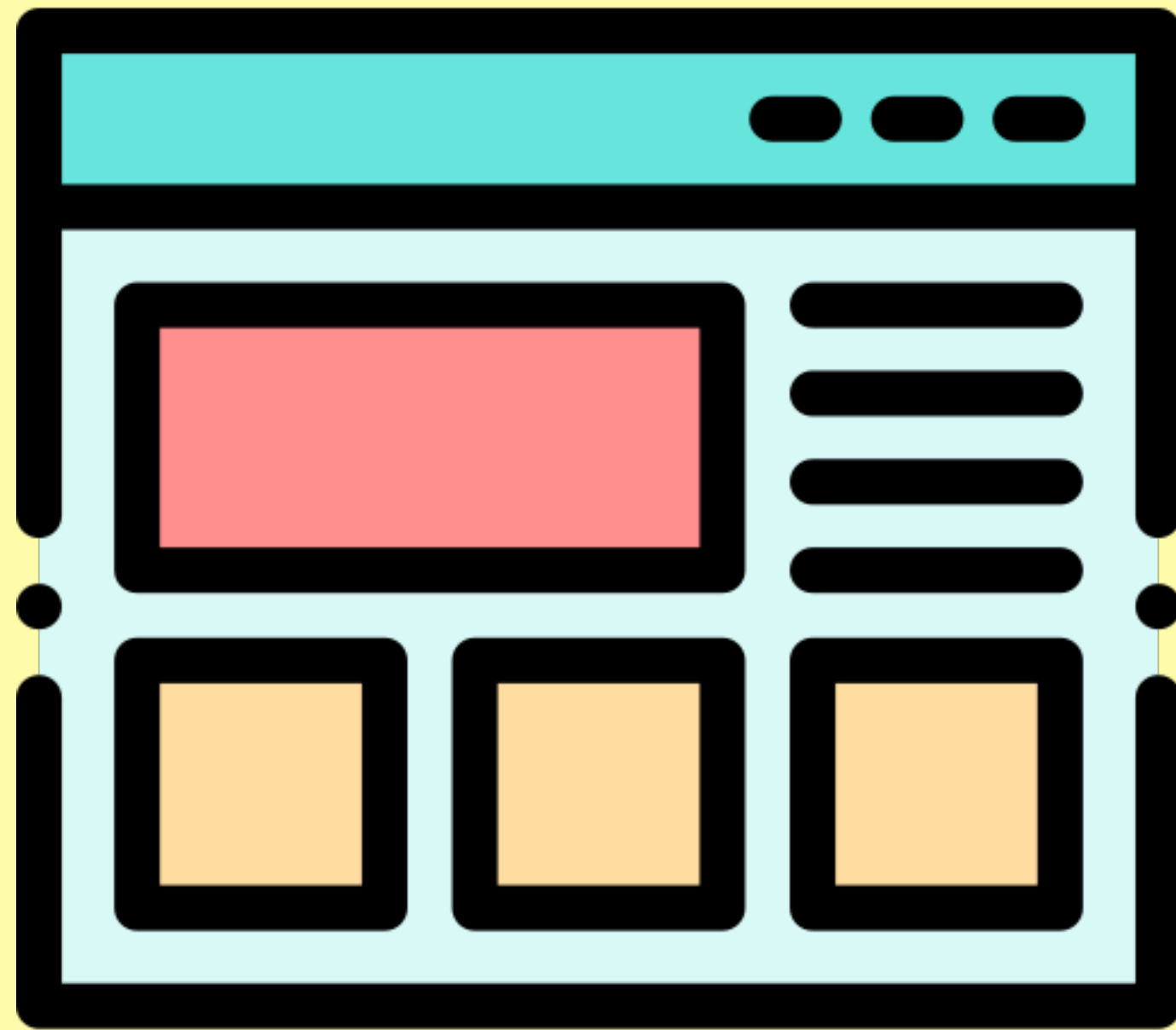
Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas

Cliente



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

Servidor



Servidor de base de datos

Petición

Respuesta



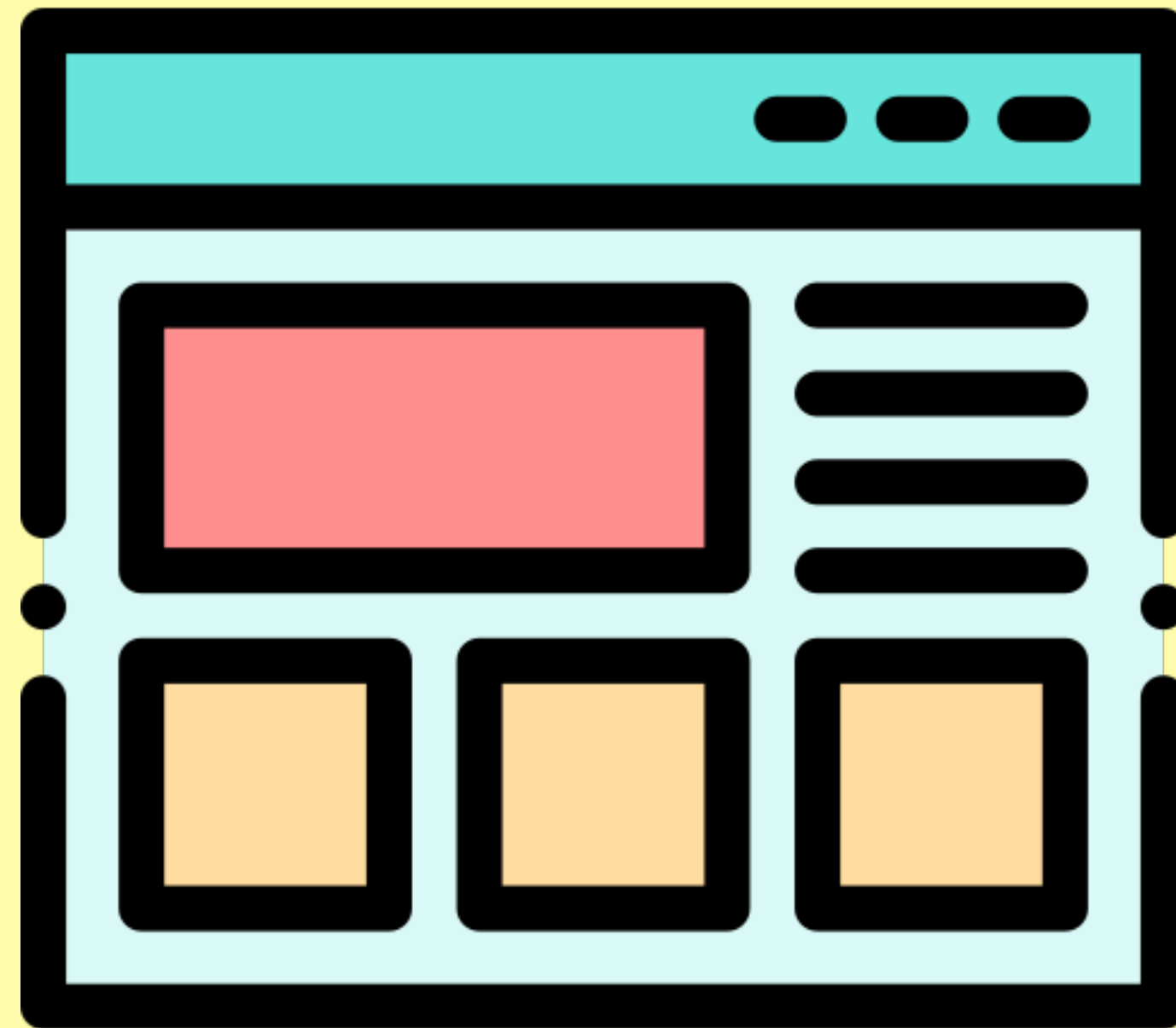
Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas

Cliente

Middleware

Servidor



Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

Petición

Respuesta



SQL

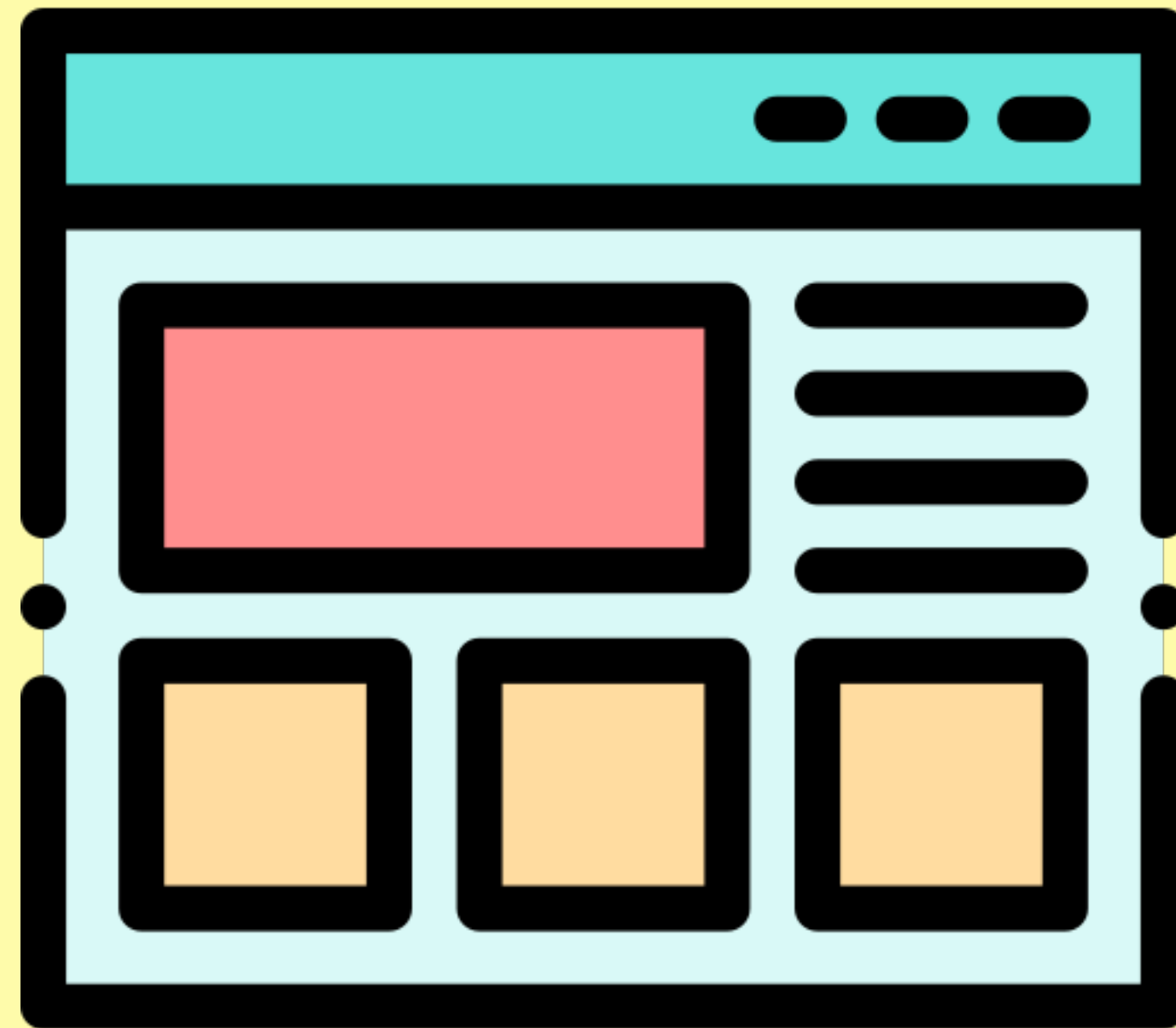
Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas

Cliente

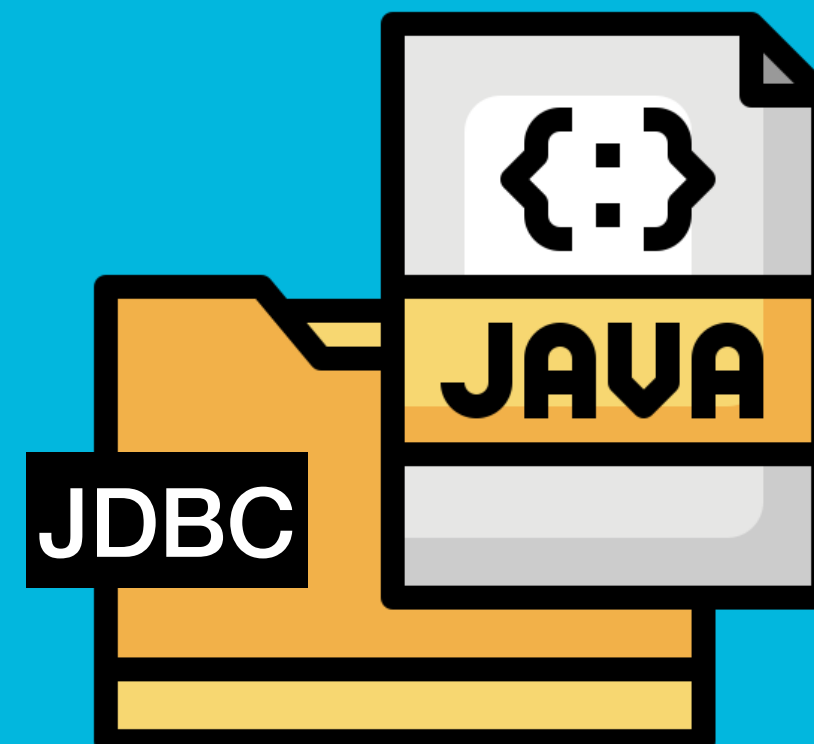


Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

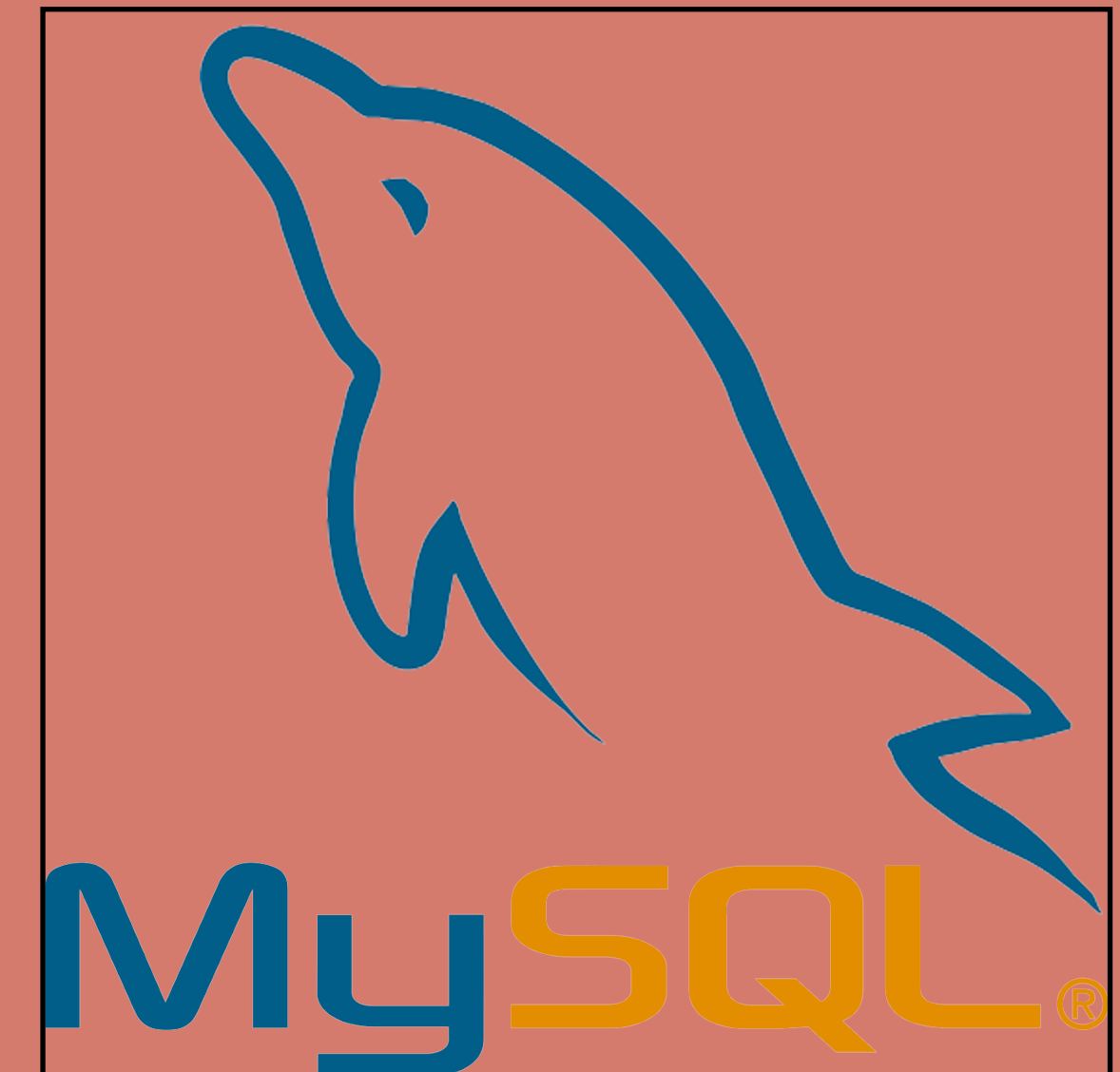
Middleware

Petición

Respuesta



Servidor



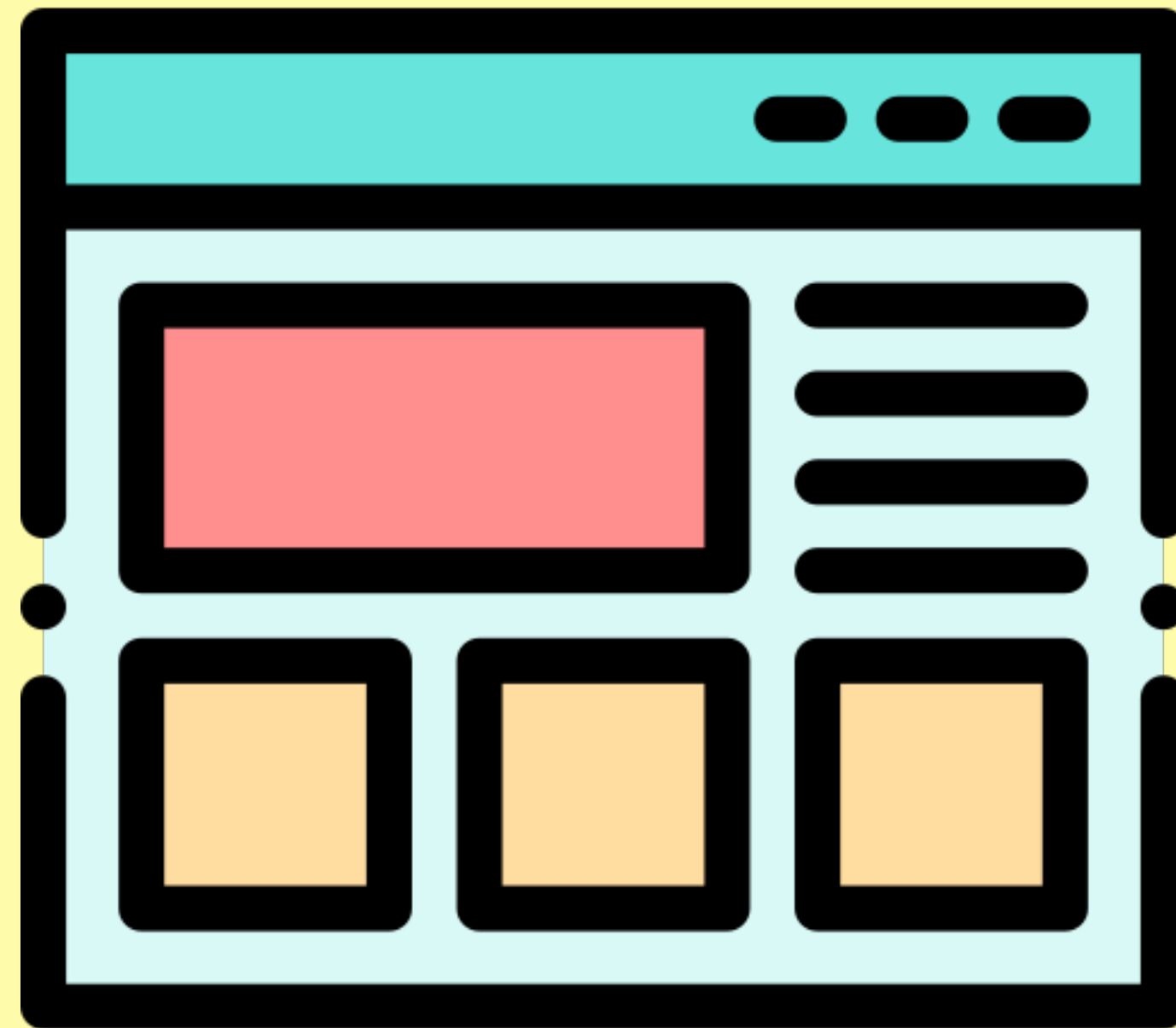
Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas

Cliente

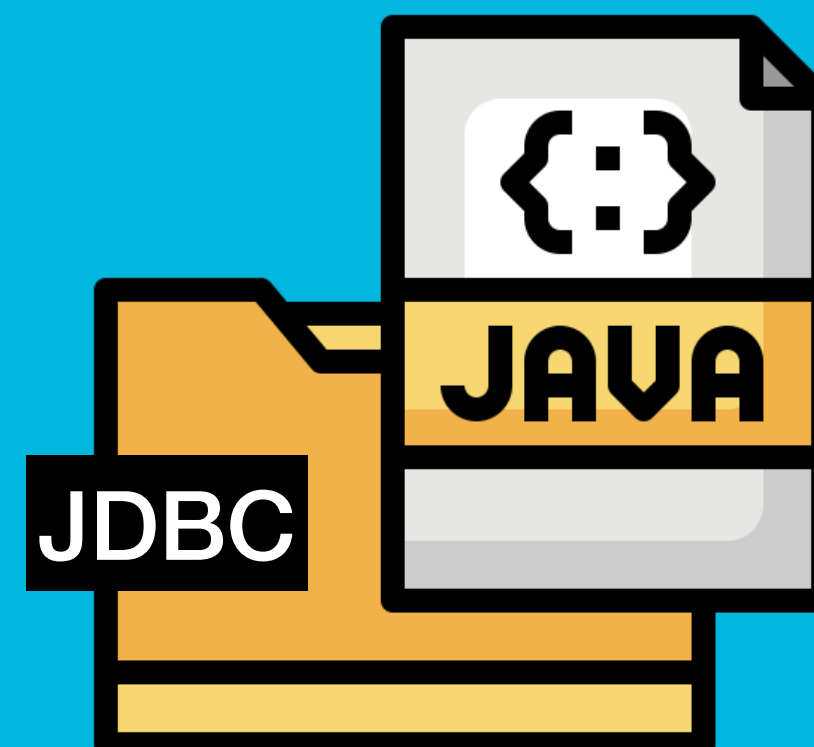


Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

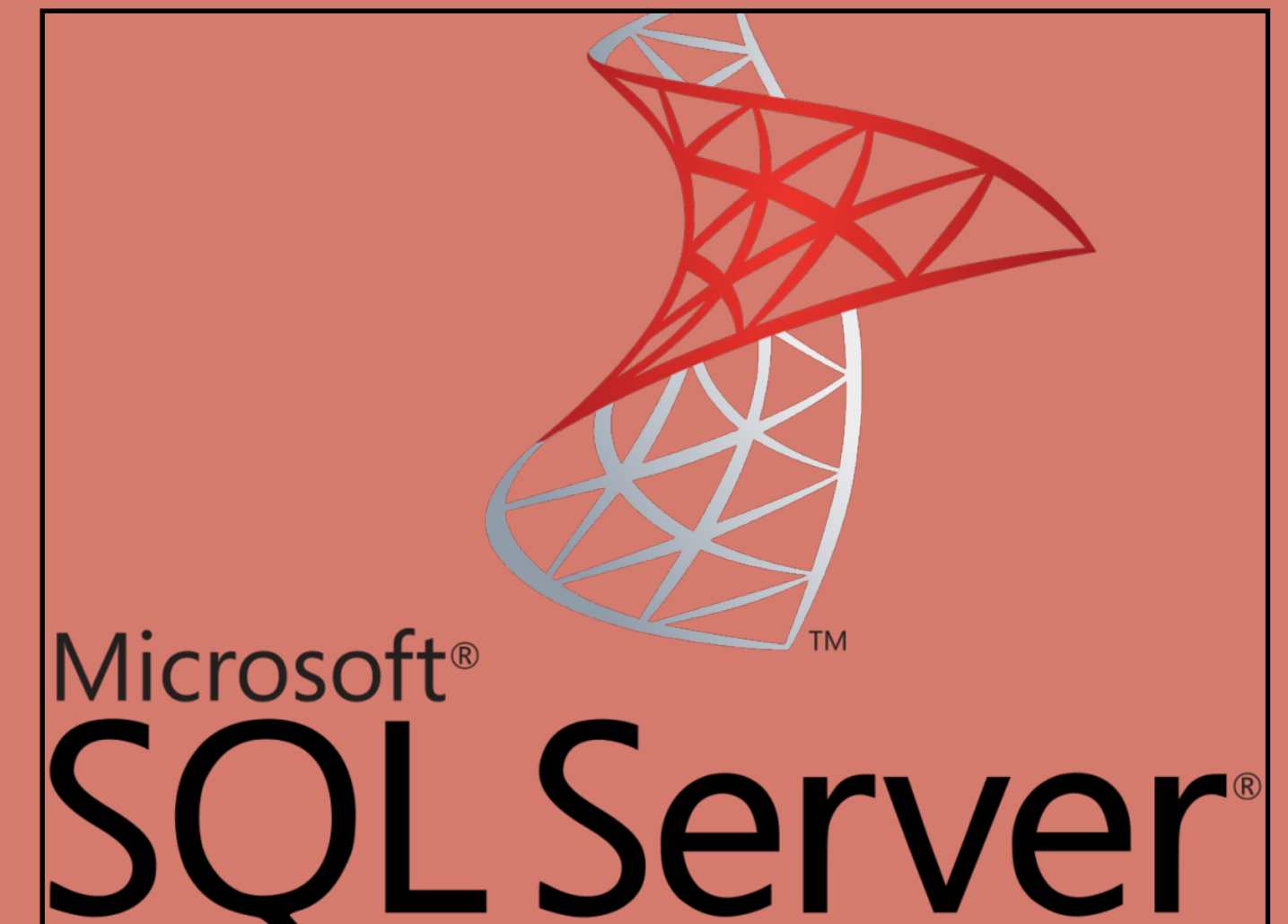
Middleware

Petición

Respuesta



Servidor



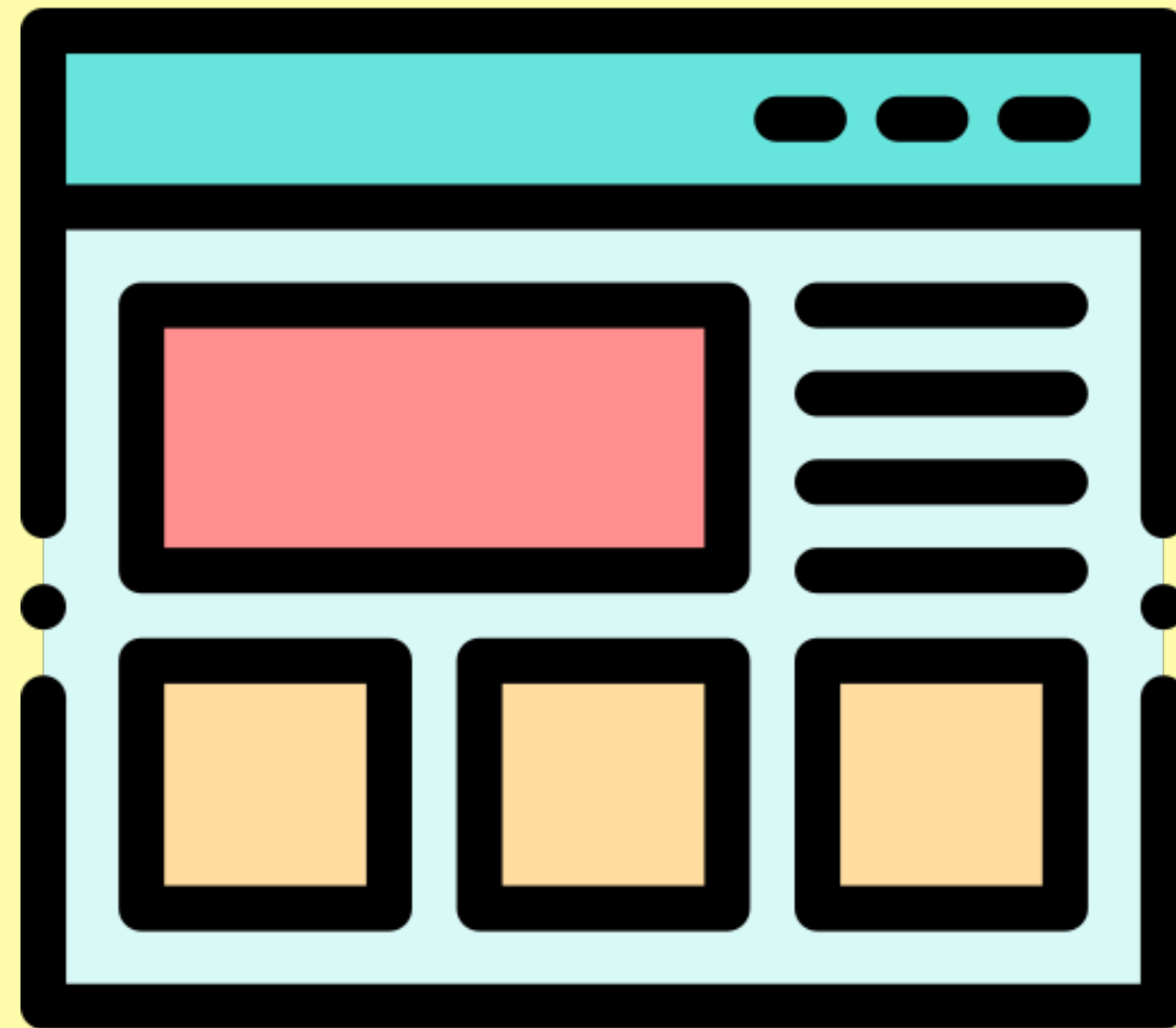
Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Estilo Arquitectónico por Capas

Cliente

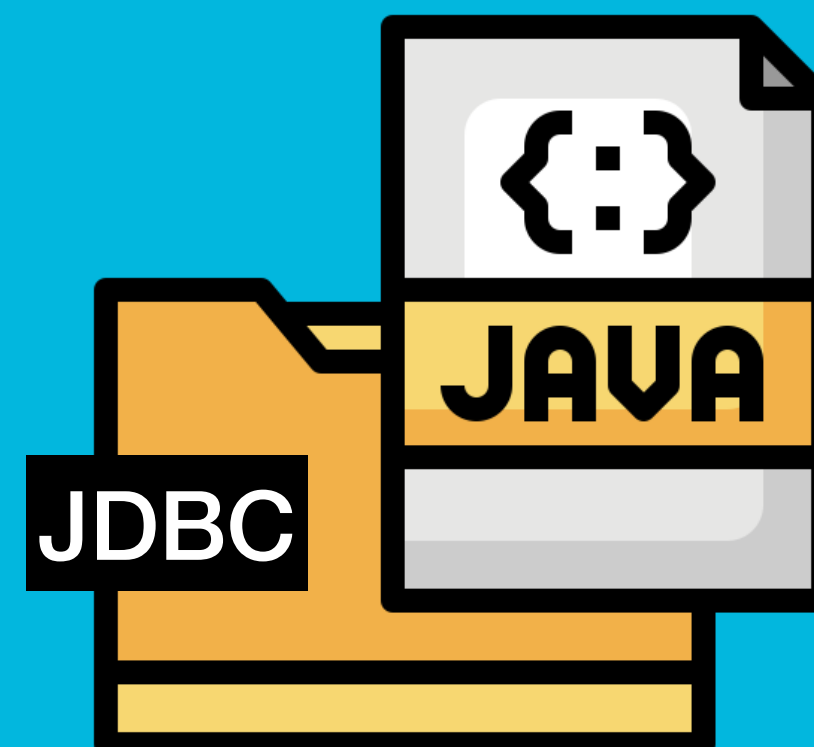


Aplicación cliente (capa de presentación, lógica del negocio, dominio y persistencia)

Middleware

Petición

Respuesta



Servidor



Servidor de base de datos



Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

MySQL

Estilo Arquitectónico por Capas

Cliente



ORACLE®

SQL Developer

Aplicación cliente

Middleware

SQL

Petición

Respuesta



Servidor

ORACLE
DATABASE

Servidor



En el curso 1INF33 - Base de datos, usaron Oracle SQL Developer como herramienta para ejecución de consultas SQL.

Estilo Arquitectónico por Capas

Cliente



Aplicación cliente

Middleware

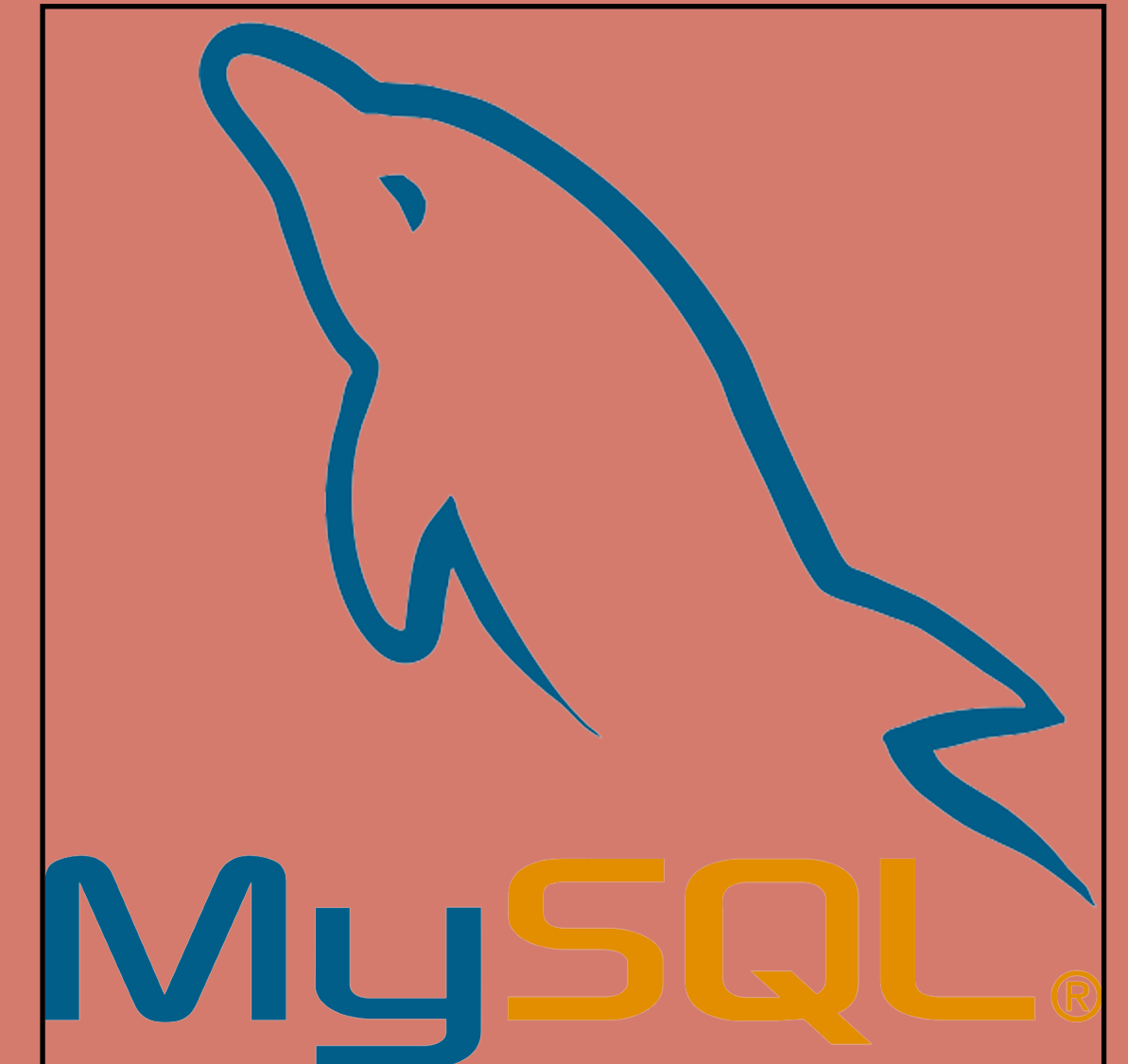


Petición

Respuesta



Servidor

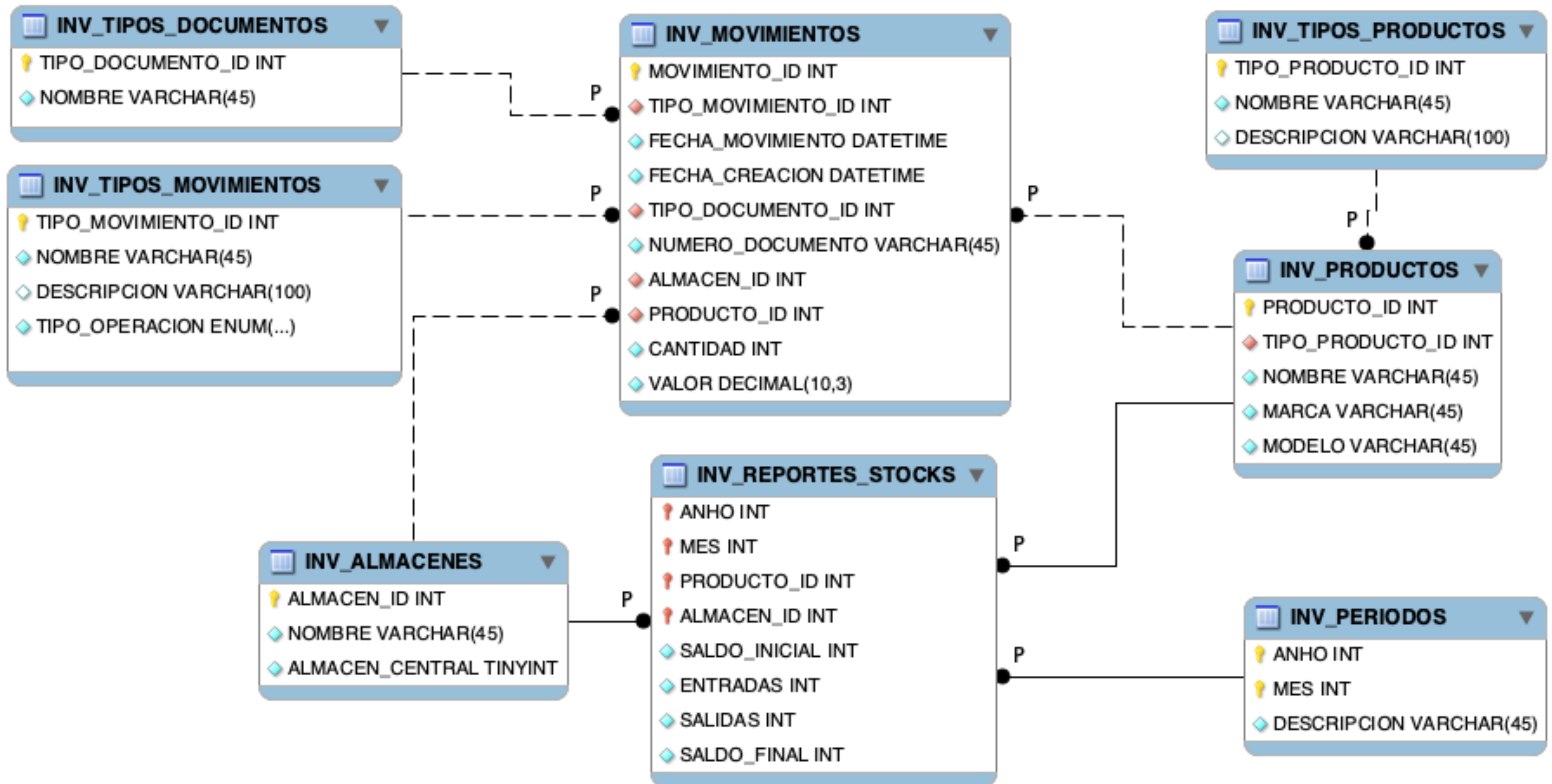


Servidor

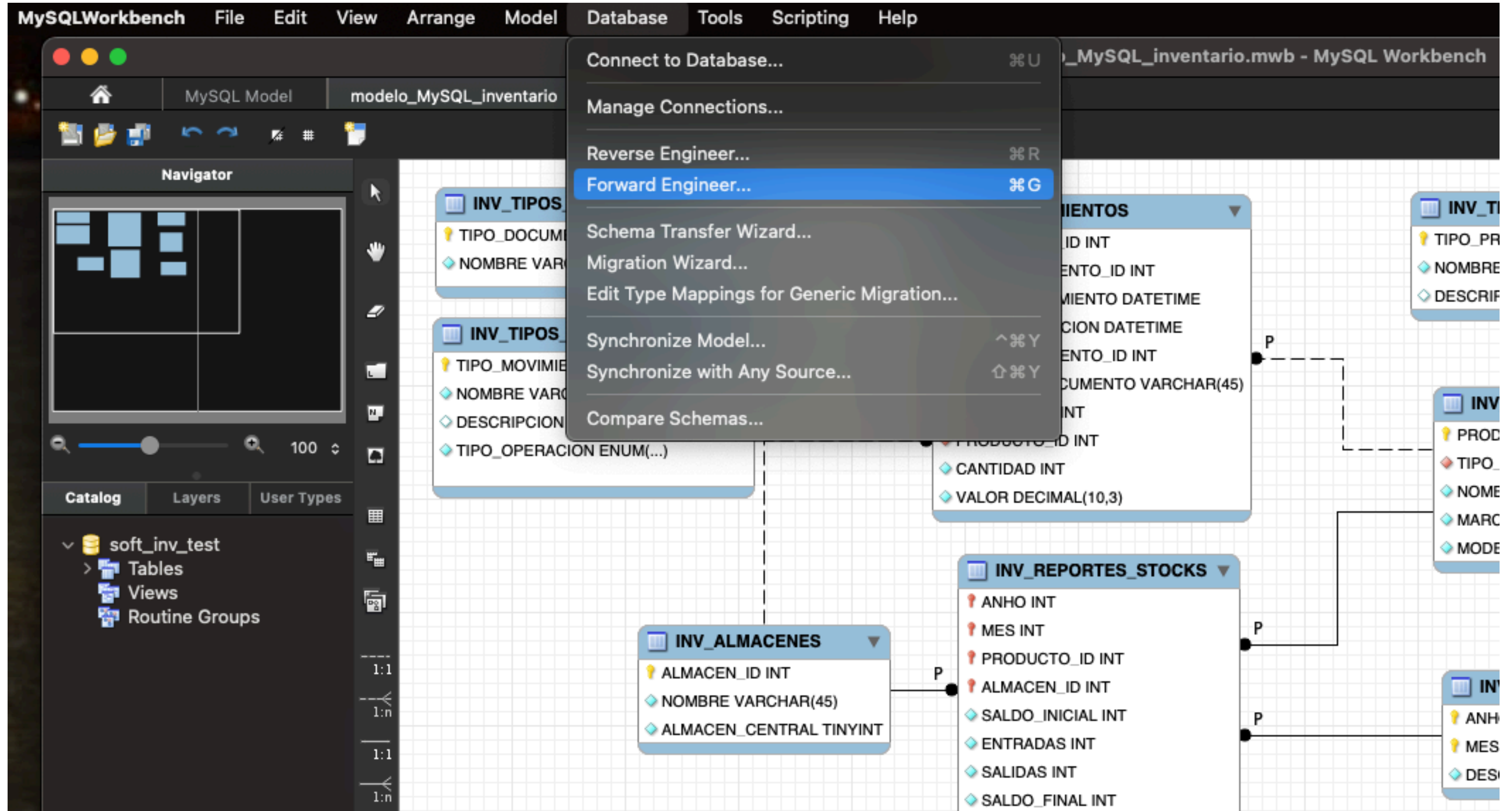


Mayor detalle sobre arquitectura de software, su representación y estilos, los verán en el curso 1INF31 - Arquitectura de Software

Modelo Relacional



Forward Engineering



AWS academy - RDS



Configuraciones importantes (1)

Acceso público [Información](#)

- ☒ **Sí**
RDS asigna una dirección IP pública a la base de datos. Las instancias de Amazon EC2 y otros recursos fuera de la VPC pueden conectarse a la base de datos. Los recursos de la VPC también pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.
- ☐ **No**
RDS no asigna una dirección IP pública a la base de datos. Solo las instancias de Amazon EC2 y otros recursos dentro de la VPC pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.



AWS academy - RDS



Configuraciones importantes (2)

inventario-mysql

Resumen

Identificador de base de datos
inventario-mysql

CPU
15.57%

Estado
Backing-up

Clase
db.t4g.micro

Rol
Instancia

Actividad actual
0 Conexiones

Motor
MySQL Community

Región y AZ
us-east-1d

Conectividad y seguridad

Supervisión

Registros y eventos

Configuración

Integraciones sin extracción, transformación y

Conectividad y seguridad

Punto de enlace y puerto

Punto de enlace
inventario-mysql.cw0x6jhjqez7.us-east-1.rds.amazonaws.com

Puerto
3306

Redes

Zona de disponibilidad
us-east-1d

VPC
vpc-0ccbea26e8bd60bc3

Grupo de subredes
default-vpc-0ccbea26e8bd60bc3

Subredes
subnet-0f8291872078df345
subnet-04f4d0dcc1e217e90
subnet-0f33b431ddae310c0
subnet-0746a99bfb1cba4bf
subnet-0427ce07d284f402d
subnet-0689045b4a3f44bce

Seguridad

Grupos de seguridad de la VPC
default (sg-0d3b31ae3fa56c1b9)
Activo

Accesible públicamente
Sí

Entidad de certificación
rds-ca-rsa2048-g1

Fecha de la entidad de certificación
May 25, 2061, 18:34 (UTC-05:00)

Fecha de expiración del certificado de instancia base de datos
March 06, 2026, 15:25 (UTC-05:00)

https://www.awsacademy.com/vforcesite/LMS_Login

20



AWS academy - RDS



Configuraciones importantes (3)

Editar reglas de entrada [Información](#)

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

Reglas de entrada [Información](#)

ID de la regla del grupo de seguridad	Tipo Información	Protocolo Información	Intervalo de puertos Información	Origen Información	Descripción: opcional Información	
-	MYSQL/Aurora ▼	TCP	3306	A... ▲		<div>Eliminar</div>

Agregar regla

Personalizada

Anywhere-IPv4 ✓

Anywhere-IPv6

Mi IP

⚠ Las reglas cuyo origen es 0.0.0.0/0 o ::/0 permiten a todas las direcciones IP acceder a la instancia. Recomendamos configurar reglas de grupo de seguridad para permitir el acceso únicamente desde direcciones IP conocidas.

Cancelar

Previsualizar los cambios

Guardar reglas

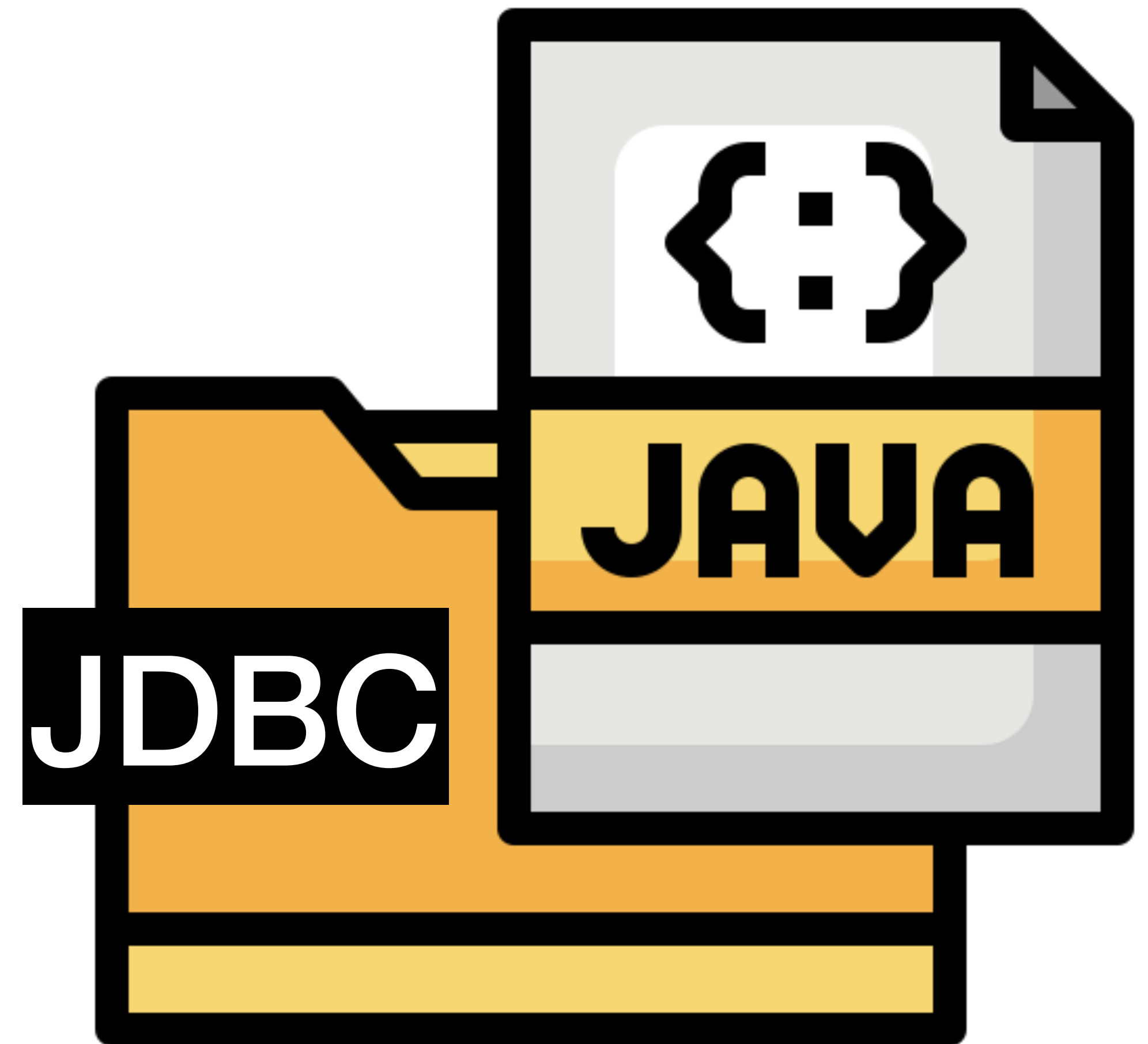
JDBC

Java Database Connectivity

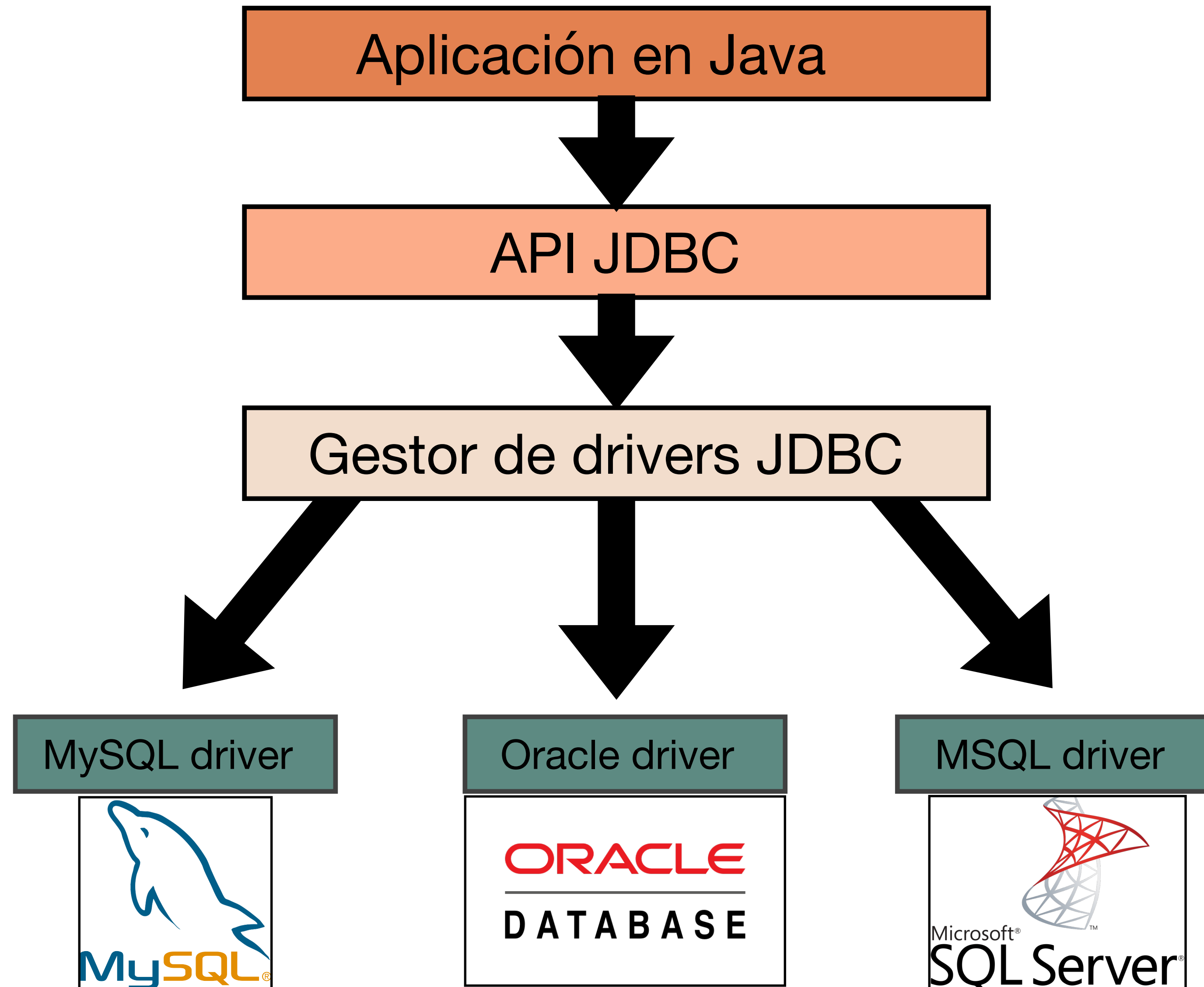
JDBC

Definición

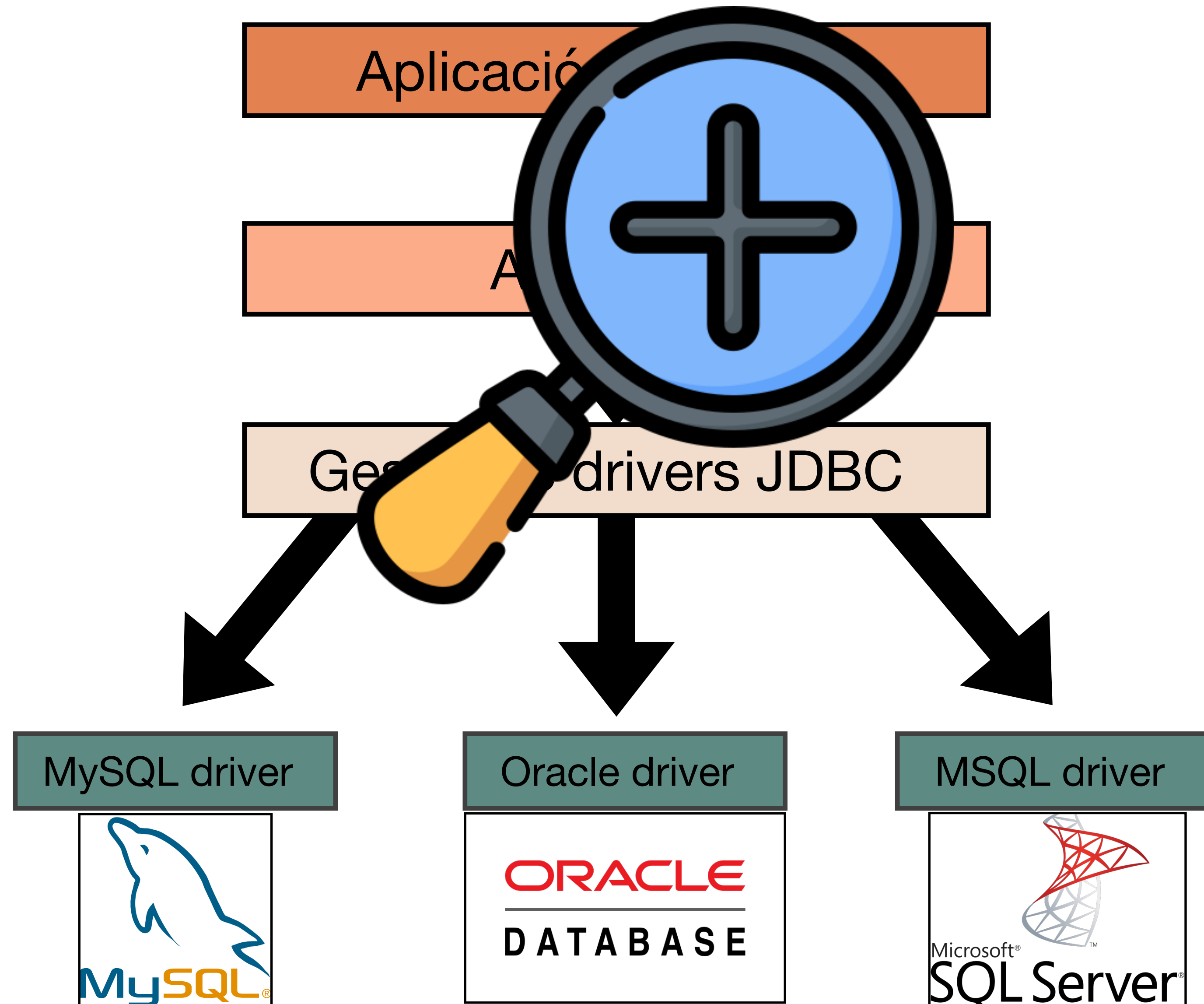
- JDBC (*Java Database Connectivity*) es una API de Java que permite a las aplicaciones conectarse e interactuar con **bases de datos** relacionales.
- JDBC proporciona **un conjunto de clases e interfaces** que permiten ejecutar consultas SQL, recuperar resultados y manipular datos desde aplicaciones Java.
- JDBC puede considerarse como si fuera un **middleware** de acceso a base de datos.



Arquitectura JDBC



Arquitectura JDBC



Arquitectura JDBC



Aplicación en Java

API JDBC

Gestor de drivers JDBC

MySQL driver

ResultSet

Statement

ResultSet

PreparedStatement

ResultSet

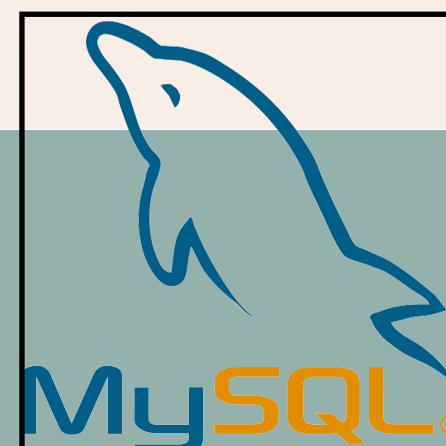
CallableStatement

Connection

DriverManager

`Class.forName(com.mysql.cj.jdbc.Driver);`

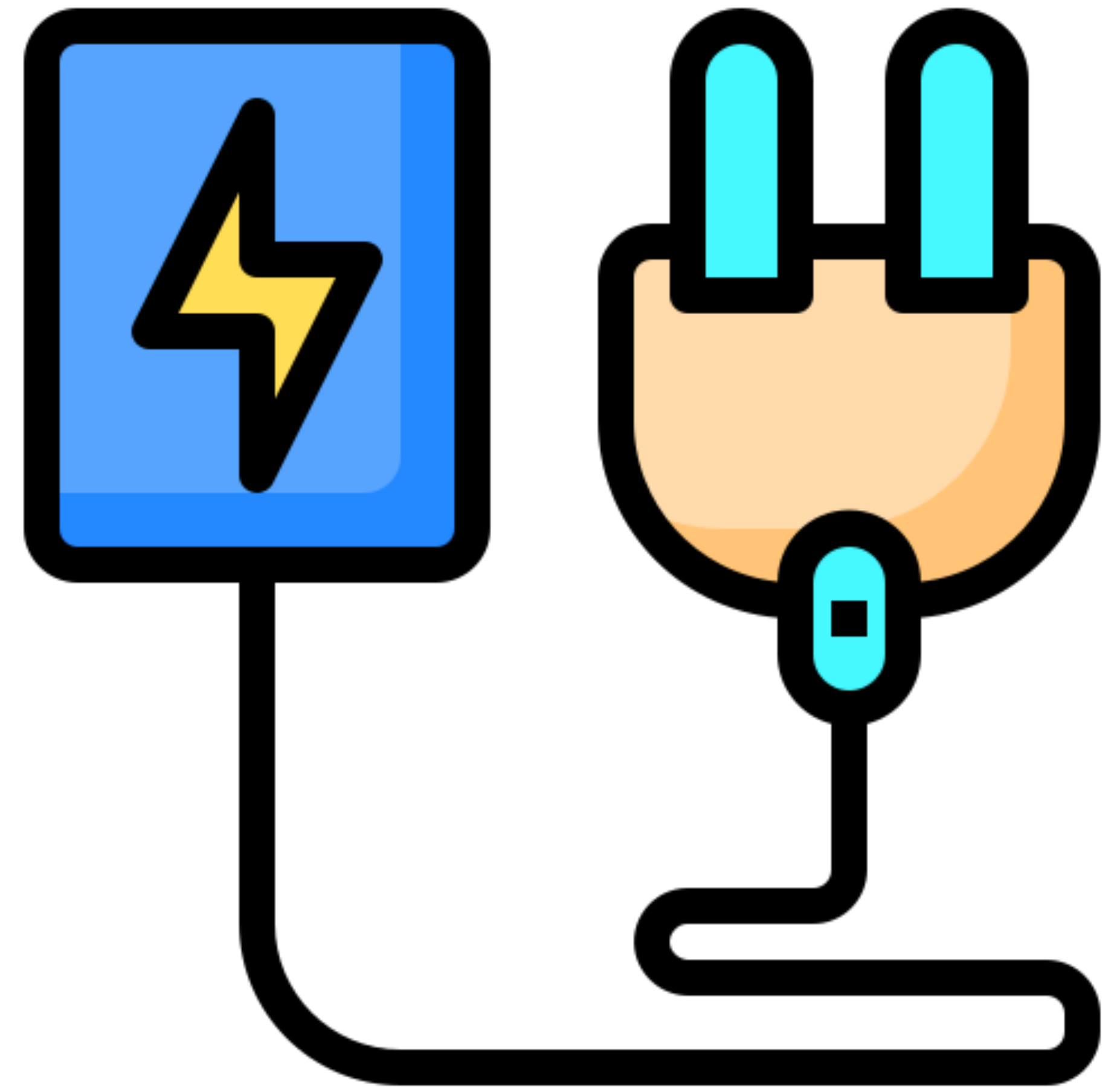
Connector/J 9.2.0



JDBC

Algoritmo getConnection

1. Obtener **datos de la conexión** (normalmente desde un archivo de texto con la contraseña encriptada).
2. Cargar dinámicamente el **driver**
(`Class.forName("com.mysql.cj.jdbc.Driver")`).
3. Generar la **URL de conexión**.
4. **Obtener la conexión** a partir del driver de conexión
(`DriverManager.getConnection()`).

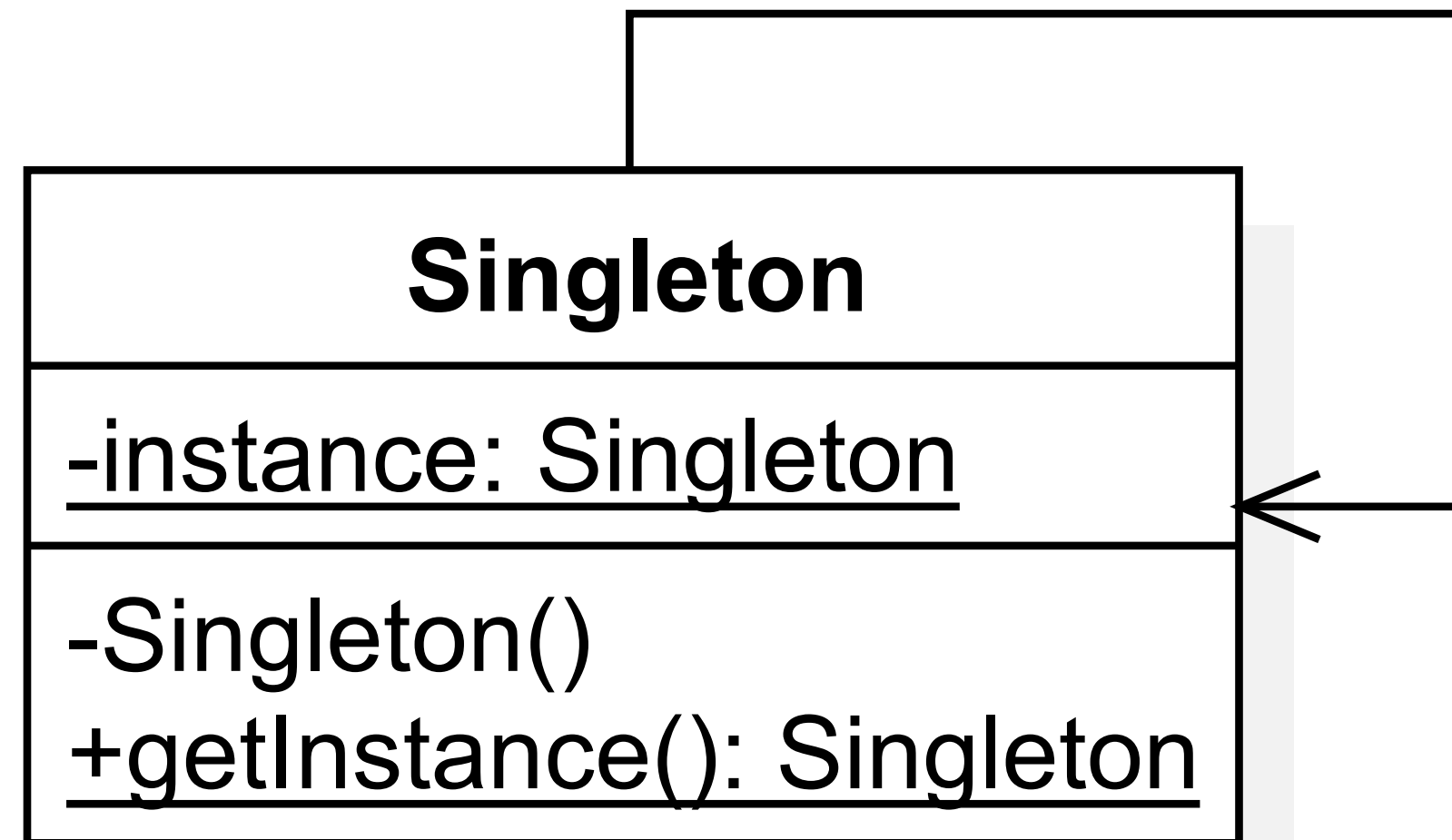


Patrón Singleton

Patrón Singleton

Definición

- El patrón **Singleton** es un **patrón de diseño creacional** que garantiza que una clase tenga una única instancia en todo el programa y proporciona un punto de acceso global a ella.
- Características
 - Única instancia: Solo se crea una instancia de la clase.
 - Acceso global: Se accede a la instancia a través de un método estático.
 - Control sobre la instanciación: Se impide la creación de nuevas instancias desde fuera de la clase pues el constructor es privado.



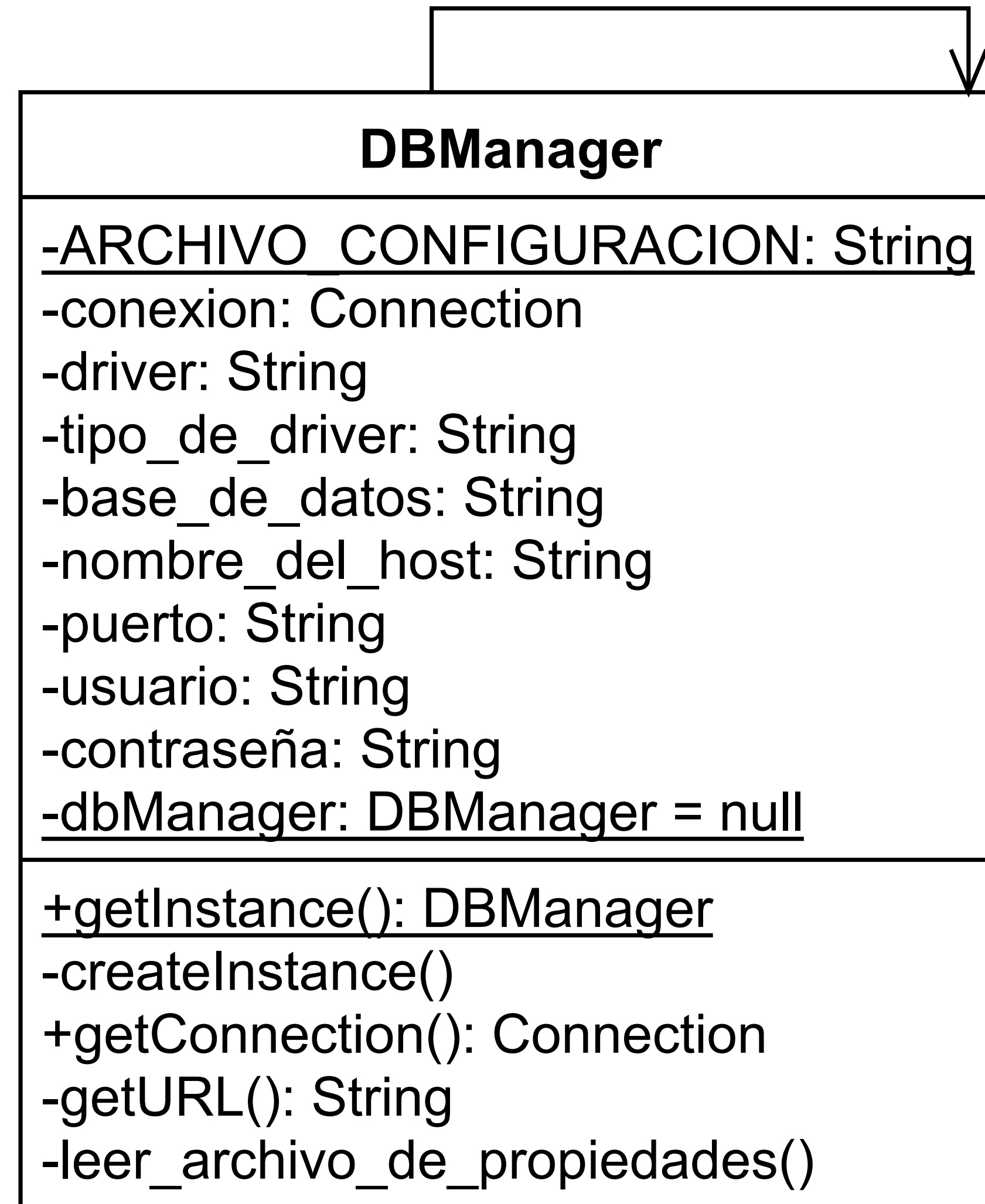
```
if (instance==null){
instance = new Singleton();
}
return instance;
```



Mayor detalle sobre diagrama de clases de diseño, notación UML y patrones, los verán en el curso 1INF50 - Diseño de Software

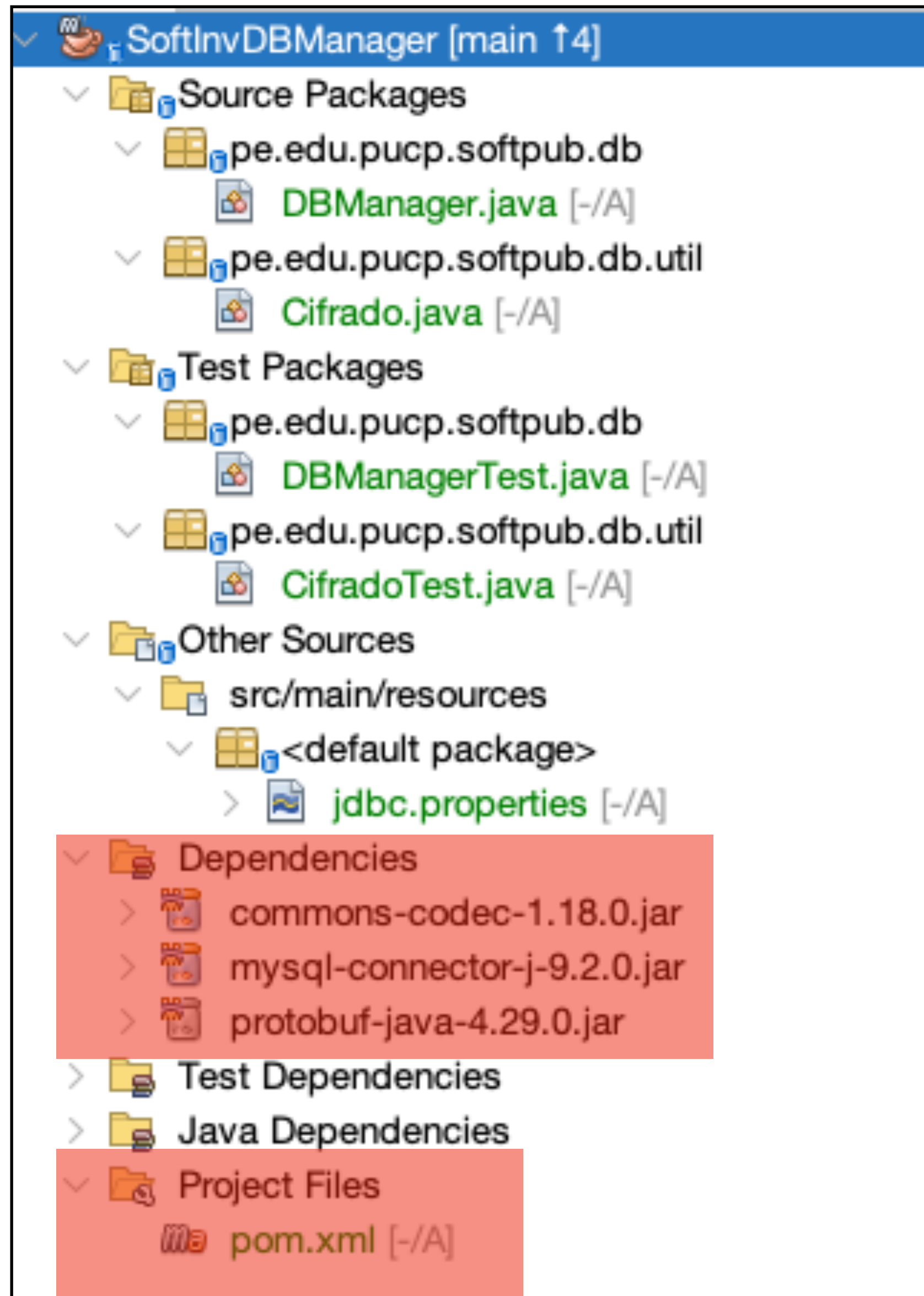
Implementación del DBManager

Implementación del DBManager



Mayor detalle sobre diagrama de clases de diseño, notación UML y patrones, los verán en el curso 1INF50 - Diseño de Software

Implementación del DBManager



Dependencias (pom.xml - *Project Object Model*)

```
<dependency>
```

```
  <groupId>commons-codec</groupId>
```

```
  <artifactId>commons-codec</artifactId>
```

```
  <version>1.18.0</version>
```

```
</dependency>
```

```
<dependency>
```

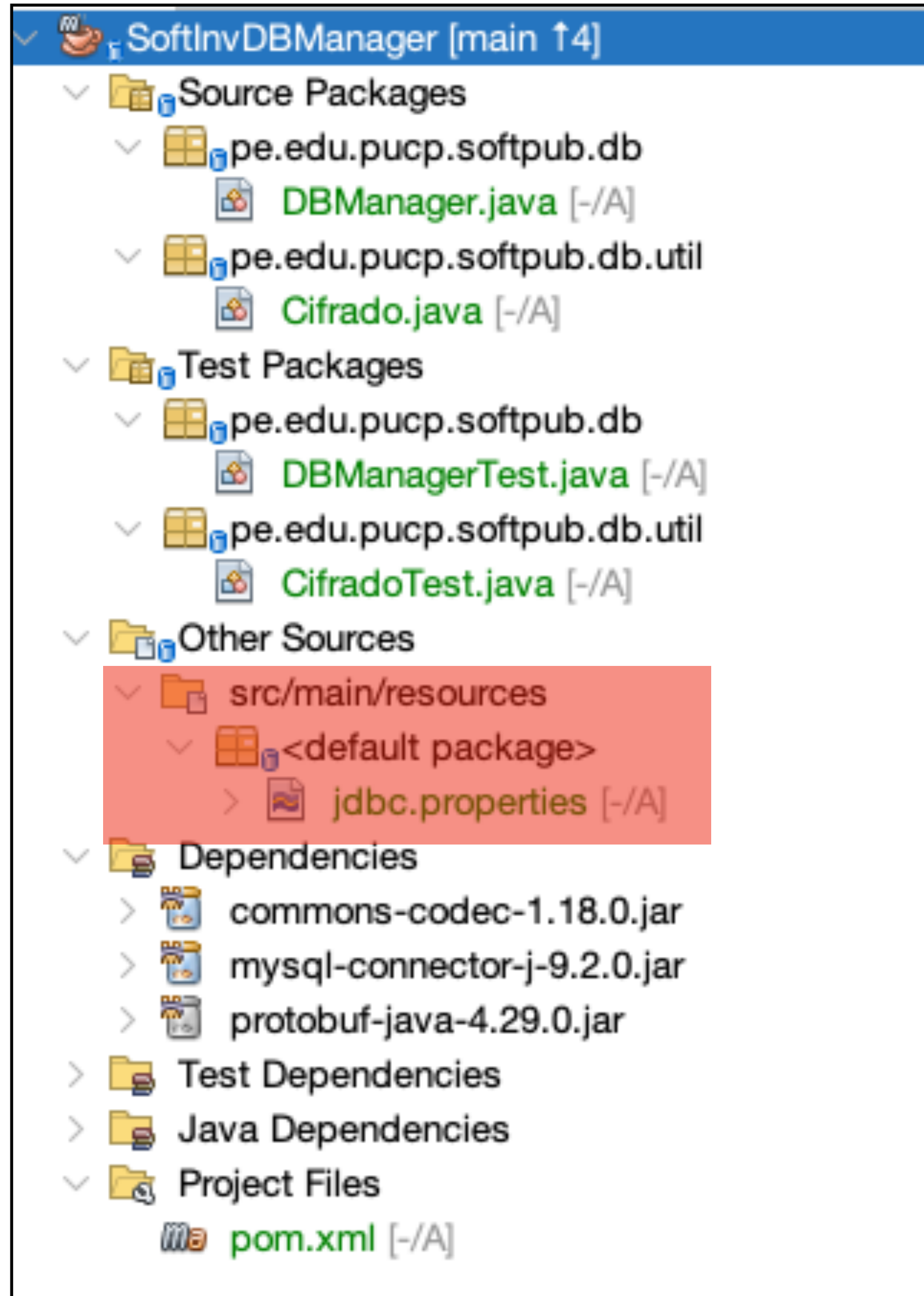
```
  <groupId>com.mysql</groupId>
```

```
  <artifactId>mysql-connector-j</artifactId>
```

```
  <version>9.2.0</version>
```

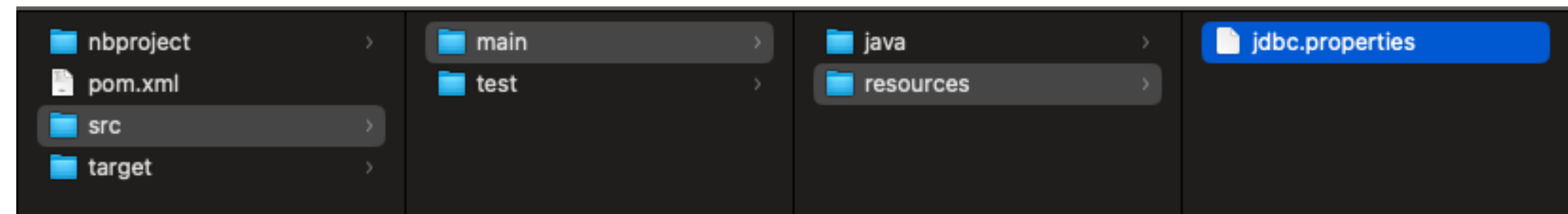
```
</dependency>
```


Implementación del DBManager

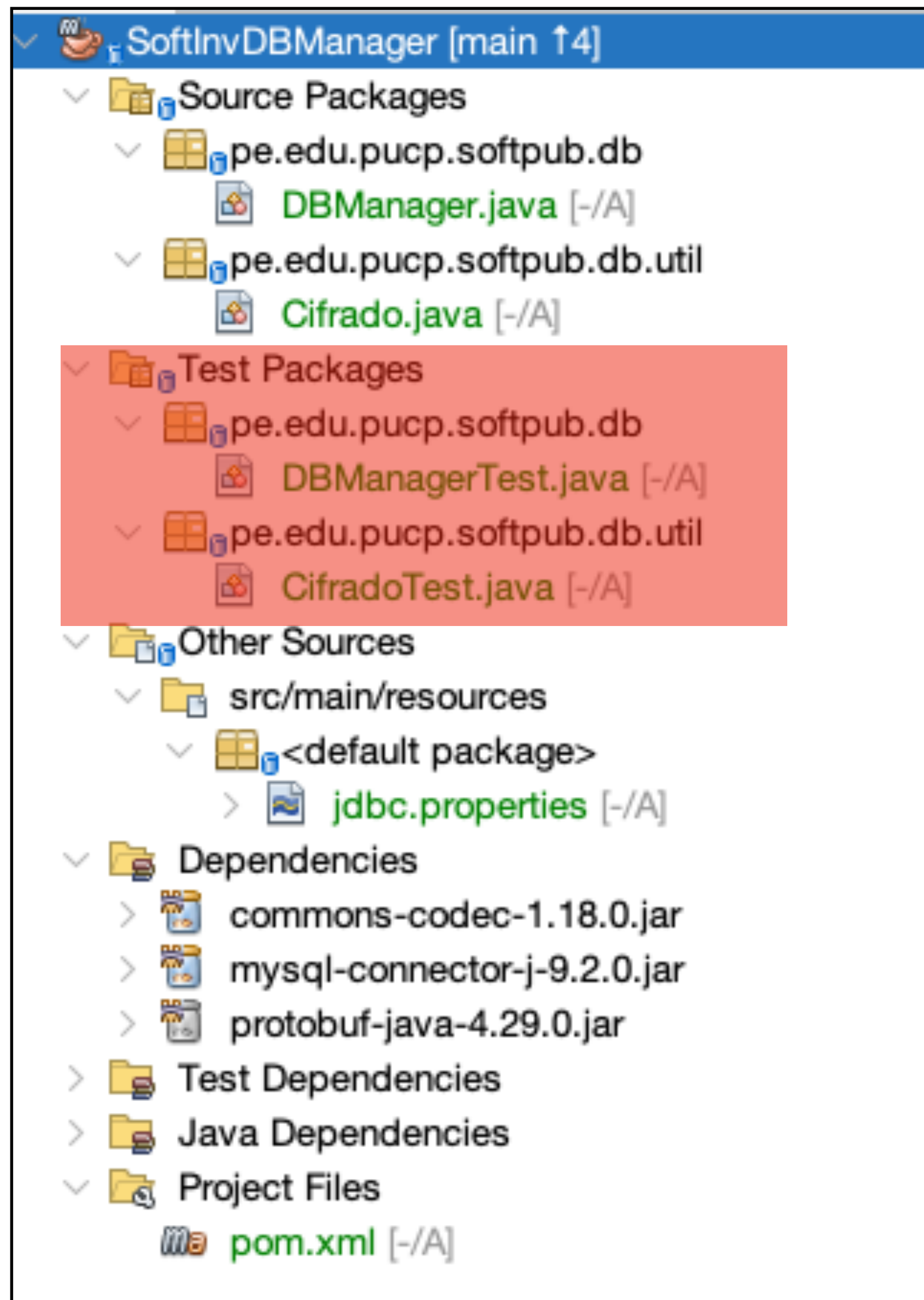


jdbc.properties

```
jdbc.properties
1 driver=com.mysql.cj.jdbc.Driver
2 tipo_de_driver=jdbc:mysql
3 base_de_datos=soft_inv_test
4 nombre_de_host=localhost
5 puerto=3306
6 usuario=admin
7 contrasenha=GFvzT/oALwhgBEP\FFB2EA==
```



Implementación del DBManager



JUnit

