

# DESARROLLO DE INTERFACES DAM2

## Práctica 6 UT4

### Generación de Informes Dinámicos con JasperStudio

---

## 1. Creación de la Base de Datos

### 1.1 Acceso a MySQL Shell

- Abrir **MySQL**.
  - Conectarse al servidor local.
- 

### 1.2 Creación de la base de datos

```
CREATE DATABASE practica_informes;  
USE practica_informes;
```

- Se crea la base de datos `practica_informes`.
  - Se selecciona para trabajar sobre ella.
- 

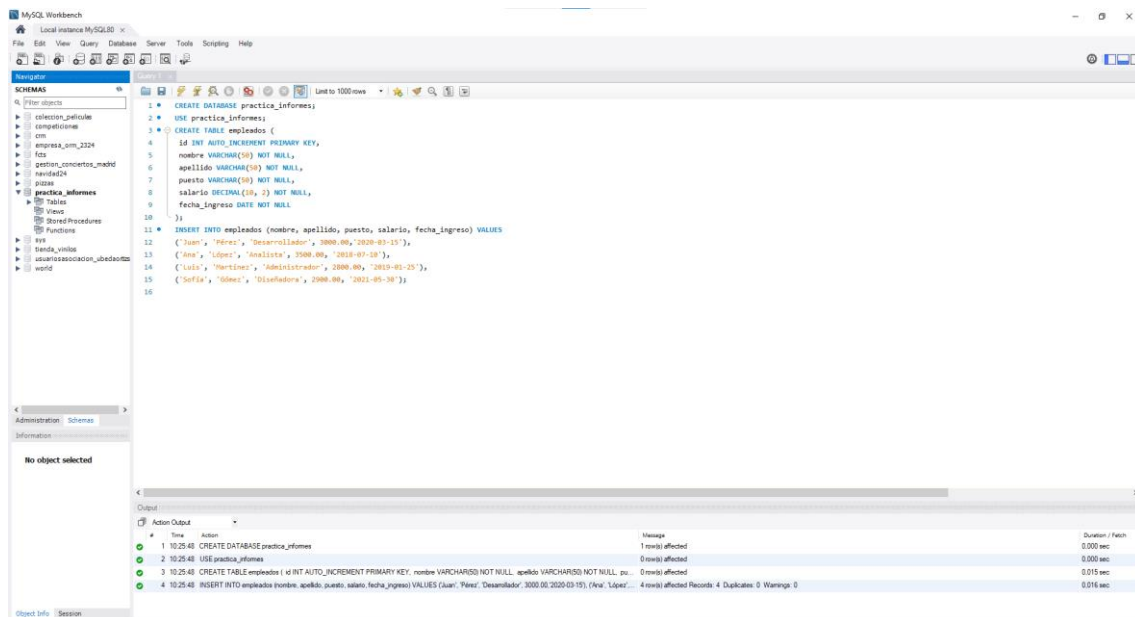
### 1.3 Creación de la tabla empleados

```
CREATE TABLE empleados (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  apellido VARCHAR(50) NOT NULL,  
  puesto VARCHAR(50) NOT NULL,  
  salario DECIMAL(10, 2) NOT NULL,  
  fecha_ingreso DATE NOT NULL  
);
```

- Se define la estructura de la tabla.
- Clave primaria autoincremental.

#### Palabras clave:

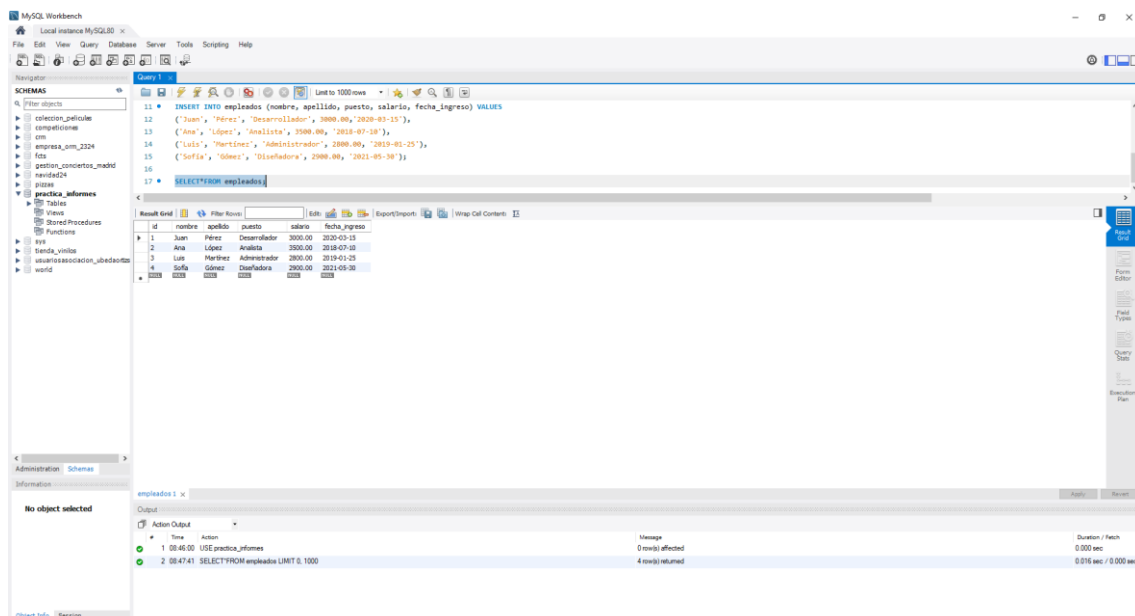
- `AUTO_INCREMENT`
- `PRIMARY KEY`
- `DECIMAL`
- `DATE`



## 1.4 Inserción de datos

INSERT INTO empleados (nombre, apellido, puesto, salario, fecha\_ingreso) VALUES  
('Juan', 'Pérez', 'Desarrollador', 3000.00, '2020-03-15'),  
('Ana', 'López', 'Analista', 3500.00, '2018-07-10'),  
('Luis', 'Martínez', 'Administrador', 2800.00, '2019-01-25'),  
('Sofía', 'Gómez', 'Diseñadora', 2900.00, '2021-05-30');

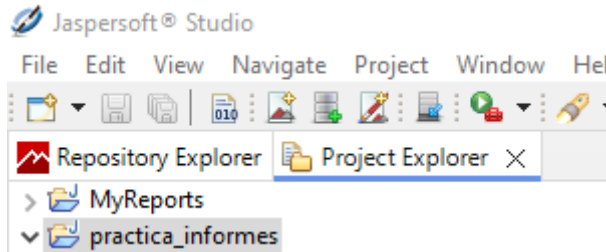
- Se insertan registros de prueba.



## 2. Creación del Informe en Jaspersoft Studio

## 2.1 Creación del proyecto

- Abrir **Jaspersoft Studio**.
- File → New → JasperReports Project.
- Nombre del proyecto: `practica_informes`.

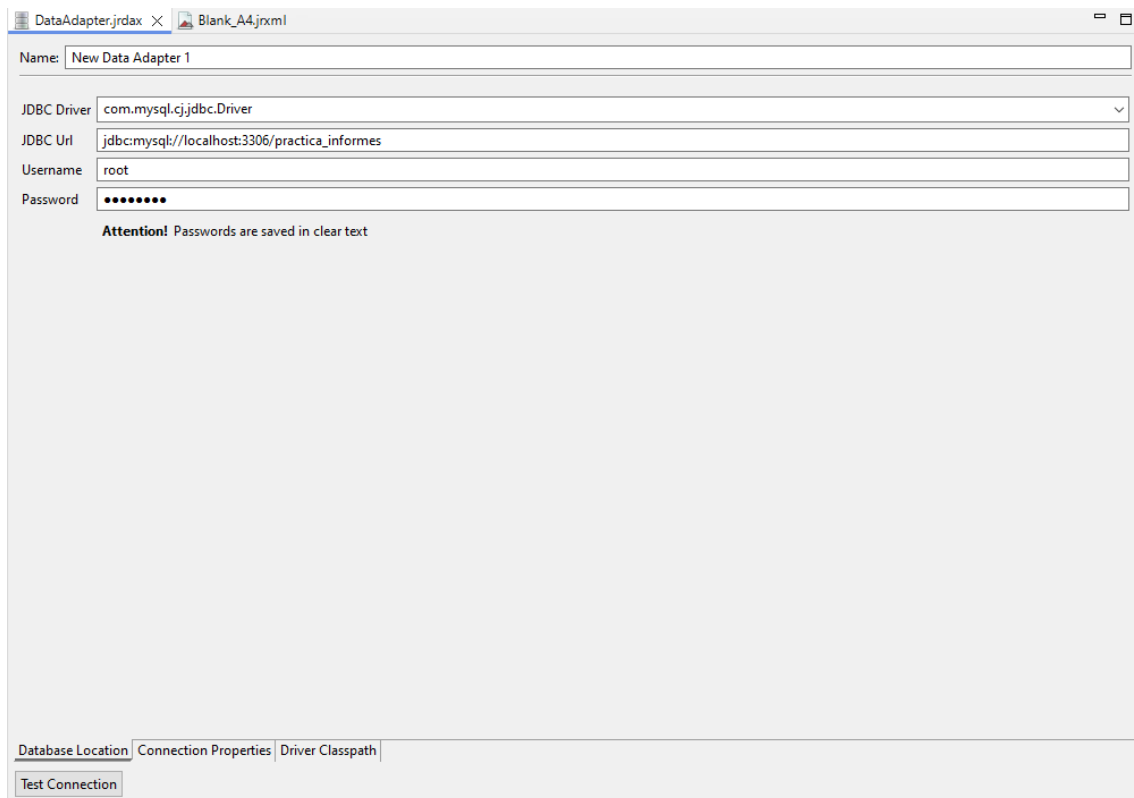


## 2.2 Configuración del Data Adapter

- Repository Explorer → Data Adapters → Create Data Adapter.
- Tipo: **Database JDBC Connection**.

### Configuración:

- JDBC URL: `jdbc:mysql://localhost:3306/practica_informes`
- Driver: `com.mysql.cj.jdbc.Driver`
- Usuario: `root`
- Contraseña: configurada en MySQL



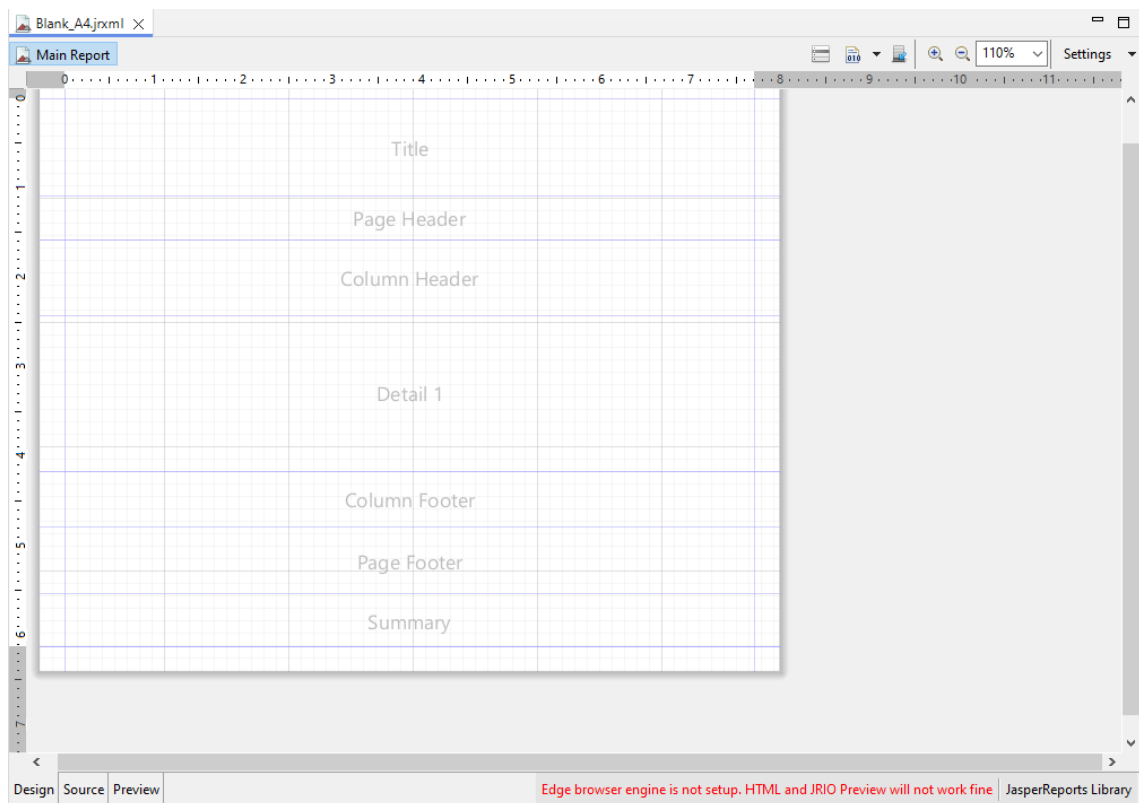
## 2.3 Añadir Driver JDBC

- Descargar **MySQL Connector/J**.
  - Añadir el `.jar` a Jaspersoft Studio.
- 

# 3. Creación del Informe

## 3.1 Nuevo informe

- File → New → Jasper Report.
- Plantilla: Blank A4.
- Nombre: `empleados_report.jrxml`.



## 3.2 Consulta SQL

```
SELECT * FROM empleados GROUP BY id;
```

- Se obtienen los datos de empleados.

### Campos añadidos:

- nombre
- apellido

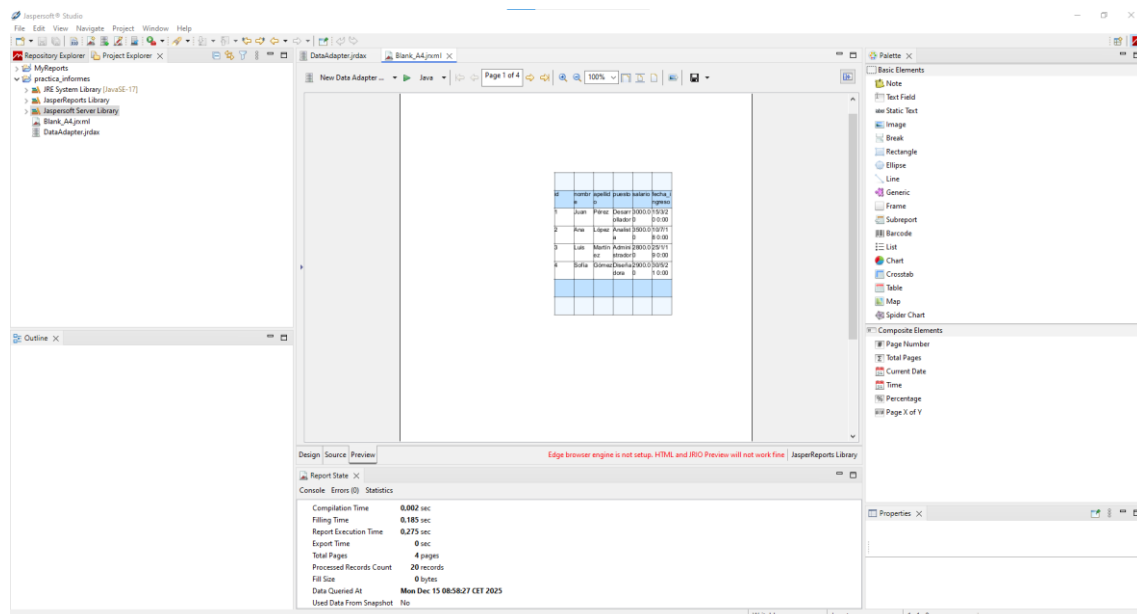
- puesto
- salario
- fecha\_ingreso

### 3.3 Diseño del informe

- Title: Título del informe.
- Column Header: nombres de columnas.
- Detail: campos dinámicos.

#### Buenas prácticas:

- Alineación correcta.
- Formato de moneda.
- Formato de fecha.



## 4. Exportación del Informe

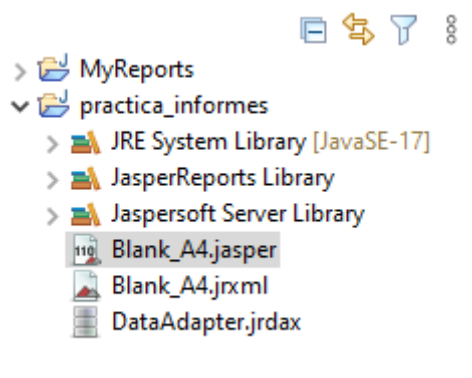
- Guardar el archivo .jrxml.
- Compilar → generar .jasper.

**Captura 11:** Informe compilado.

### 4.1 Ubicación del archivo

- El archivo .jasper se guarda en:
  - o target/classes

- 0 src/main/resources



---

## 5. Integración del Informe en Java

### 5.1 Proyecto Java con Maven

- Crear proyecto Java.
- Usar Maven.

#### Dependencias:

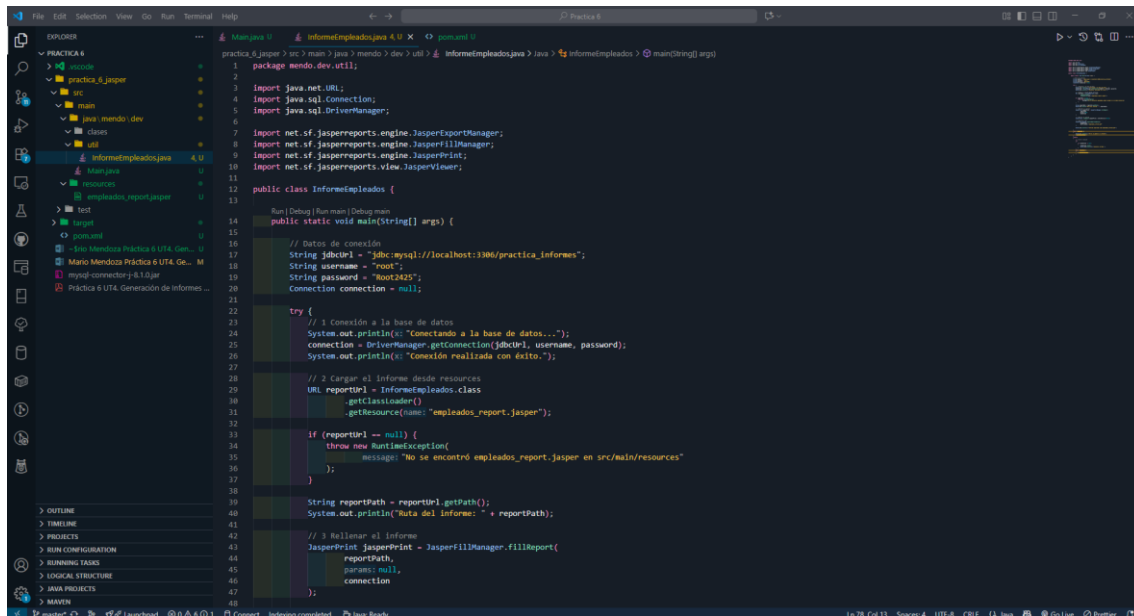
- MySQL Connector/J
- JasperReports

**Captura 13:** pom.xml configurado.

---

### 5.2 Código Java

- Clase: InformeEmpleados.
- Funcionalidades:
  - Conexión JDBC.
  - Carga del informe.
  - Visualización.
  - Exportación a PDF.



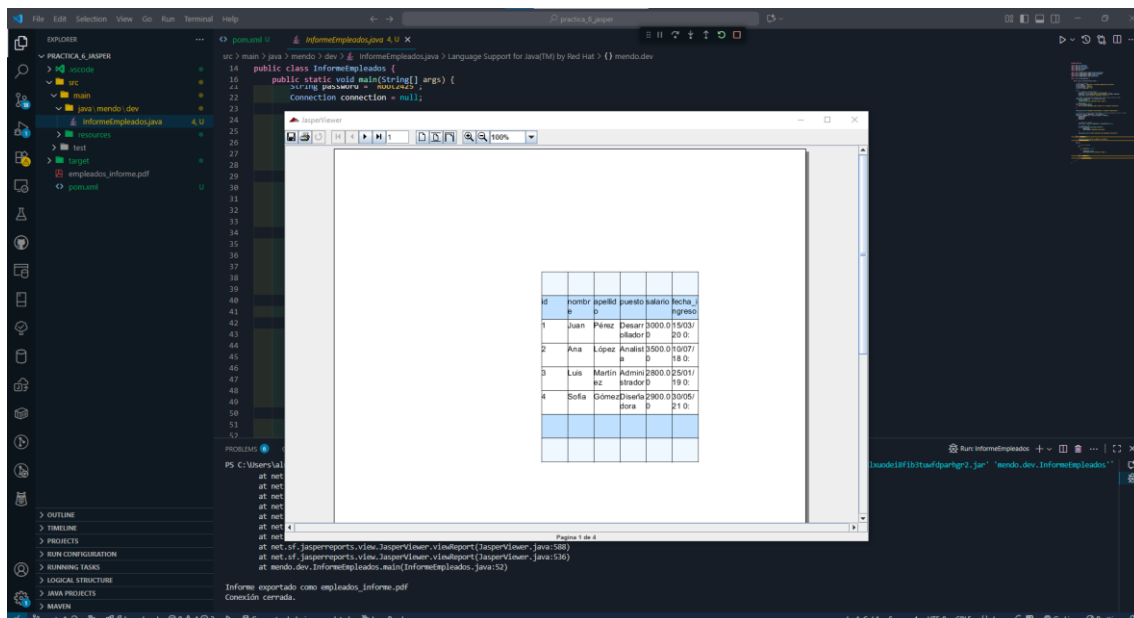
```
practica_6_jasper > src > main > java > mendo > dev > util > InformesEmpleados.java > java > InformesEmpleados > main(String[] args)
1 package mendo.dev.util;
2
3 import java.net.URL;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6
7 import net.sf.jasperreports.engine.JasperExportManager;
8 import net.sf.jasperreports.engine.JasperFillManager;
9 import net.sf.jasperreports.engine.JasperPrint;
10 import net.sf.jasperreports.view.JasperViewer;
11
12 public class InformesEmpleados {
13
14     public static void main(String[] args) {
15
16         // Datos de conexión
17         String jdbcUrl = "jdbc:mysql://localhost:3306/practica_informes";
18         String username = "root";
19         String password = "root3425";
20         Connection connection = null;
21
22         try {
23             // 1 Conexión a la base de datos
24             System.out.println("Conectando a la base de datos...");
25             connection = DriverManager.getConnection(jdbcUrl, username, password);
26             System.out.println("Conexión realizada con éxito.");
27
28             // 2 Cargar el informe desde recursos
29             URL reportUrl = InformesEmpleados.class
30                 .getClassLoader()
31                 .getResource(name: "empleados_report.jasper");
32
33             if (reportUrl == null) {
34                 throw new RuntimeException(
35                     "No se encontró empleados_report.jasper en src/main/resources"
36                 );
37             }
38
39             String reportPath = reportUrl.getPath();
40             System.out.println("Ruta del informe: " + reportPath);
41
42             // 3 Rellenar el informe
43             JasperPrint jasperPrint = JasperFillManager.fillReport(
44                 reportPath,
45                 params: null,
46                 connection
47             );
48
49             JasperExportManager.exportReportToPdfStream(
50                 jasperPrint.printStream(),
51                 "empleados_informe.pdf"
52             );
53
54             JasperViewer.viewReport(jasperPrint, false);
55
56         } catch (Exception e) {
57             e.printStackTrace();
58         }
59     }
60 }
```

## 5.3 Ejecución

- Verificar que `empleados_report.jasper` esté en `src/main/resources`.
- Ejecutar la aplicación.

### Resultado:

- Informe mostrado.
- PDF generado.

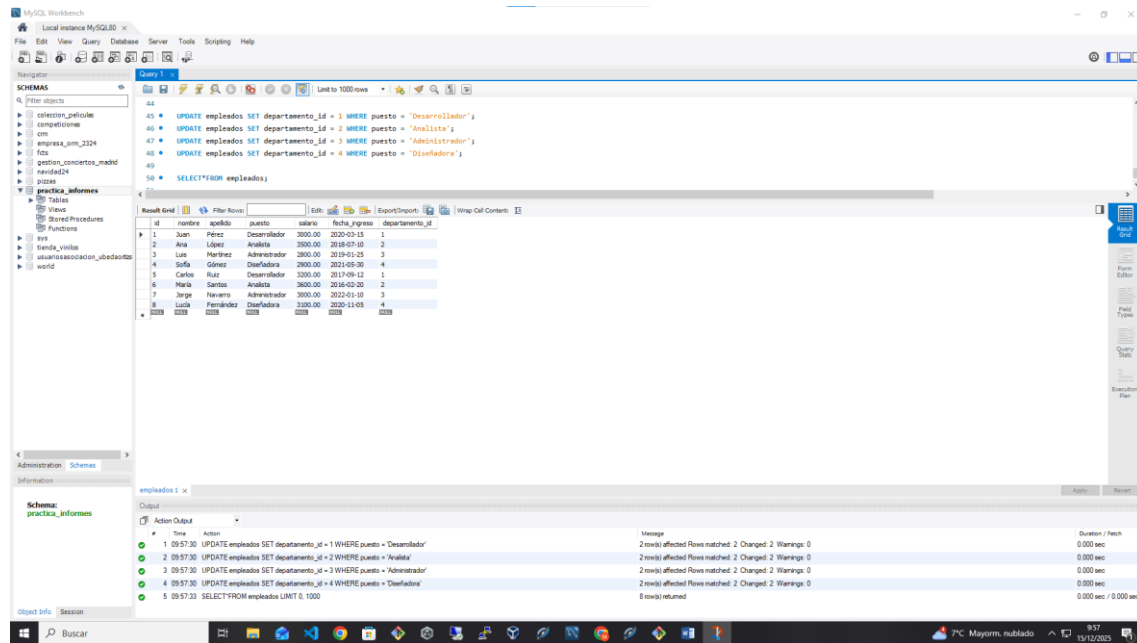


id	nombre	apellido	puesto	salario	fecha_nacimiento	fecha_registro
1	Juan	Pérez	Desar	3000	01/03/1990	2024-03-20 0:18:0
2	Ana	López	Analist	3500	01/07/1985	2024-03-20 0:18:0
3	Luis	Martín	Admin	2800	02/01/1991	2024-03-20 0:19:0
4	Sofía	Gómez	Diseña	2900	03/05/1995	2024-03-20 0:21:0

## 6. Ejercicios Propuestos

## 6.1 Extensión de la Base de Datos

- Añadir más empleados.
- Crear tabla departamentos.
- Relación empleado-departamento.



## 6.2 Personalización del Informe

- Ordenación por puesto.
- Gráfico de barras:
  - Salario promedio por puesto.
  -

