

# Proyecto TechDAM - JDBC Avanzado



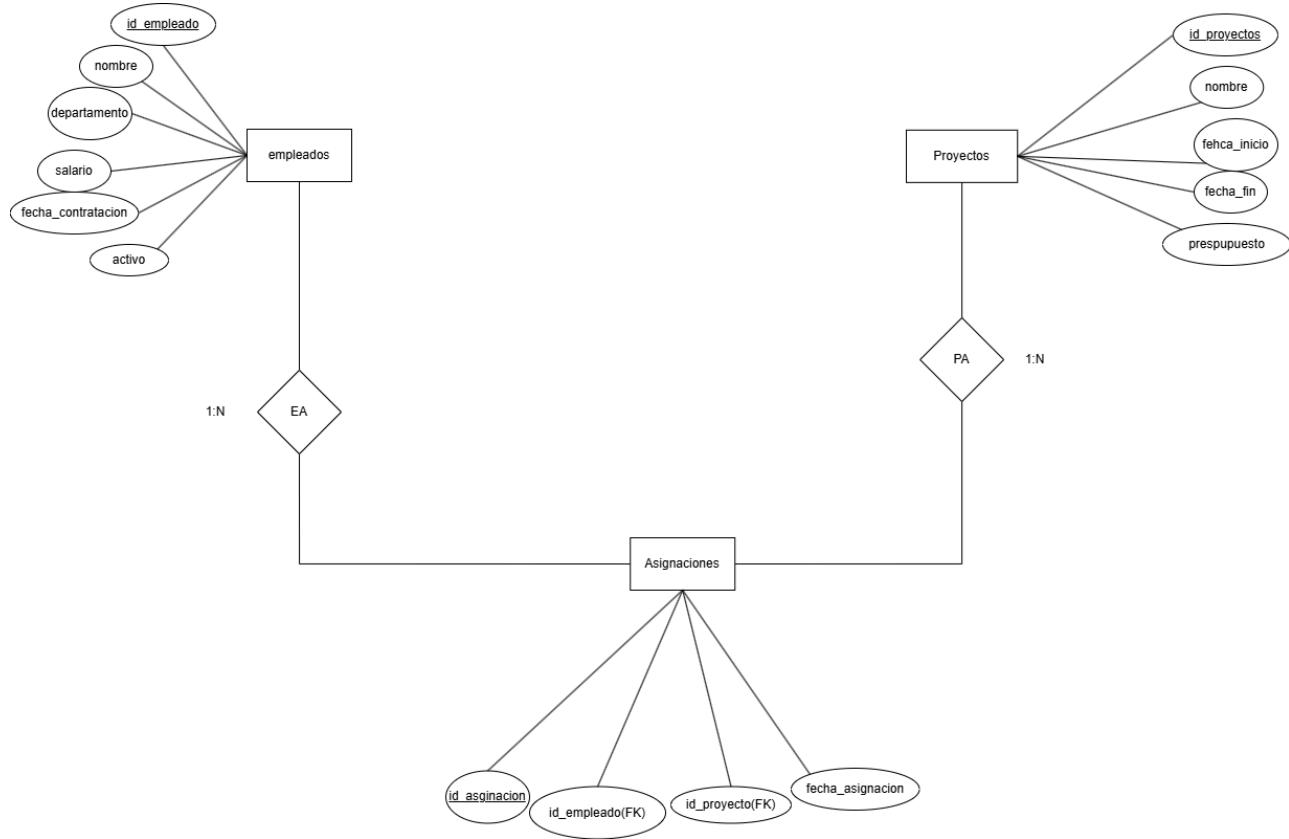
**Nombre:** Sergio Gómez Pila

**Fecha:** 15/11/2025

# Índice

|   |           |
|---|-----------|
| <b>1. Diagrama de Base de Datos.....</b>          | <b>3</b>  |
| <b>2. Capturas de Pantalla.....</b>               | <b>4</b>  |
| 2.1 Tablas creadas en MySQL Workbench.....        | 4         |
| 2.3 Ejecución exitosa de CRUD de empleados.....   | 5         |
| 2.4 Ejecución exitosa de CRUD de proyectos.....   | 7         |
| 2.5 Invocación de procedimiento almacenado.....   | 8         |
| 2.6 Transferencia de Presupuesto.....             | 9         |
| 2.7 Pool de conexiones configurado.....           | 9         |
| <b>3. Explicación de Decisiones Técnicas.....</b> | <b>11</b> |
| <b>4. Problemas Encontrados y Soluciones.....</b> | <b>12</b> |

# 1. Diagrama de Base de Datos



# 2. Capturas de Pantalla

## 2.1 Tablas creadas en MySQL Workbench

```
-- Base de datos techdam
CREATE DATABASE techdam_completo;
USE techdam_completo;
-- Tabla empleados
CREATE TABLE empleados (
    id_empleado INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    departamento VARCHAR(50),
    salario DECIMAL(10,2),
    fecha_contratacion DATE,
    activo BOOLEAN DEFAULT TRUE
);

-- Tabla proyectos
CREATE TABLE proyectos (
    id_proyecto INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    fecha_inicio DATE,
    fecha_fin DATE,
    presupuesto DECIMAL(12,2)
);

-- Tabla asignaciones
CREATE TABLE asignaciones (
    id_asignacion INT AUTO_INCREMENT PRIMARY KEY,
    id_empleado INT,
    id_proyecto INT,
    fecha_asignacion DATE,
    FOREIGN KEY (id_empleado) REFERENCES empleados(id_empleado)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (id_proyecto) REFERENCES proyectos(id_proyecto)
        ON DELETE CASCADE ON UPDATE CASCADE
);
;
```

## 2.3 Ejecución exitosa de CRUD de empleados

```
*C:\Program Files\Java\jdk-23\bin\java.exe* -javaagent:C:\Program Files\Java\jolokia-jar\tmp\jolokia.jar=port=8080,host=127.0.0.1,contextPath=/jolokia
== MENÚ PRINCIPAL ==
1. Gestión de Empleados
2. Gestión de Proyectos
3. Transferencia de Presupuesto
4. Procedimientos Almacenados
5. Salir
Seleccione una opción: 1

--- Gestión de Empleados ---
1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 1
[main] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1
[main] INFO com.zaxxer.hikari.pool.HikariPool - HikariPool-1
[main] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1
Ingrese el nombre del empleado:
Arturo
Ingrese el nombre del departamento:
Marketing
Ingrese el salario
600
Ingrese la fecha de contratacion (formato: yyyy-MM-dd):
2025-11-15
El empleado es activo? (true/ false)
true
Empleado insertado con éxito. Filas afectadas: 1
```

```

1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 2

== EMPLEADOS ==
ID: 1 | Nombre: Ana Pérez | Depto: Desarrollo | Salario: 3000,00 € | Fecha: 2020-03-01 | Activo: Si
ID: 2 | Nombre: Luis Gómez | Depto: Marketing | Salario: 2000,00 € | Fecha: 2019-07-15 | Activo: Si
ID: 3 | Nombre: Marta López | Depto: Desarrollo | Salario: 3240,00 € | Fecha: 2021-06-10 | Activo: Si
ID: 4 | Nombre: Carlos Díaz | Depto: Soporte | Salario: 1800,00 € | Fecha: 2018-02-20 | Activo: Si
ID: 5 | Nombre: Sofía Ruiz | Depto: Desarrollo | Salario: 3000,00 € | Fecha: 2017-11-05 | Activo: No
ID: 6 | Nombre: sergio | Depto: soporte | Salario: 5000,00 € | Fecha: 2025-11-14 | Activo: Si
ID: 7 | Nombre: iker | Depto: soporte | Salario: 6000,00 € | Fecha: 2025-11-14 | Activo: No
ID: 9 | Nombre: Arturo | Depto: Marketing | Salario: 600,00 € | Fecha: 2025-11-15 | Activo: Si

--- Gestión de Empleados ---
1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 3
Ingrese el ID del empleado: 9
ID: 9 | Nombre: Arturo | Depto: Marketing | Salario: 600,00 € | Fecha: 2025-11-15 | Activo: Si

```

```

--- Gestión de Empleados ---
1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 4
Introduce el id del empleado que quieres editar:
9
Introduce en nuevo nombre del empleado:
Manolito
Introduce el nuevo departamento:
Soporte
Introduce el salario nuevo:
8000
Introduce la fecha de contratacion ( formato: yyyy-MM-dd )
2025-10-21
Introduce si esta activo o no(True or False):
false
Empleado actualizado correctamente

--- Gestión de Empleados ---
1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 3
Ingrese el ID del empleado: 9
ID: 9 | Nombre: Manolito | Depto: Soporte | Salario: 8000,00 € | Fecha: 2025-10-21 | Activo: No

```

```

--- Gestión de Empleados ---
1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 5
Ingrese el ID a eliminar: 9
Empleado eliminado correctamente

--- Gestión de Empleados ---
1. Crear empleado
2. Listar empleados
3. Buscar empleado por ID
4. Actualizar empleado
5. Eliminar empleado
0. Volver
Seleccione una opción: 2

== EMPLEADOS ==
ID: 1 | Nombre: Ana Pérez | Depto: Desarrollo | Salario: 3000,00 € | Fecha: 2020-03-01 | Activo: Sí
ID: 2 | Nombre: Luis Gómez | Depto: Marketing | Salario: 2000,00 € | Fecha: 2019-07-15 | Activo: Sí
ID: 3 | Nombre: Marta López | Depto: Desarrollo | Salario: 3240,00 € | Fecha: 2021-06-10 | Activo: Sí
ID: 4 | Nombre: Carlos Díaz | Depto: Soporte | Salario: 1800,00 € | Fecha: 2018-02-20 | Activo: Sí
ID: 5 | Nombre: Sofía Ruiz | Depto: Desarrollo | Salario: 3000,00 € | Fecha: 2017-11-05 | Activo: No
ID: 6 | Nombre: Sergio | Depto: soporte | Salario: 5000,00 € | Fecha: 2025-11-14 | Activo: Sí
ID: 7 | Nombre: Iker | Depto: soporte | Salario: 6000,00 € | Fecha: 2025-11-14 | Activo: No

```

## 2.4 Ejecución exitosa de CRUD de proyectos

```

--- Gestión de Proyectos ---
1. Crear proyecto
2. Listar proyectos
3. Buscar proyecto por ID
4. Actualizar proyecto
5. Eliminar proyecto
0. Volver
Seleccione una opción: 1
Ingrese el nombre del proyecto:
Tata Árboles
Ingrese la fecha de inicio (yyyy-MM-dd):
2023-03-11
Ingrese la fecha de fin (yyyy-MM-dd):
2023-04-22
Ingrese el presupuesto:
15000
Proyecto insertado con éxito. Filas afectadas: 1

--- Gestión de Proyectos ---
1. Crear proyecto
2. Listar proyectos
3. Buscar proyecto por ID
4. Actualizar proyecto
5. Eliminar proyecto
0. Volver
Seleccione una opción: 2

== PROYECTOS ==
ID: 1 | Nombre: Sistema ERP | Fecha_Inicio: 2023-01-10 | Fecha_Fin: 2023-12-31 | Presupuesto: 50000,00 €
ID: 2 | Nombre: App Móvil | Fecha_Inicio: 2023-04-01 | Fecha_Fin: 2023-10-01 | Presupuesto: 30000,00 €
ID: 3 | Nombre: Campaña Publicitaria | Fecha_Inicio: 2023-06-01 | Fecha_Fin: 2023-09-30 | Presupuesto: 15000,00 €
ID: 4 | Nombre: Portal Web | Fecha_Inicio: 2022-02-15 | Fecha_Fin: 2022-12-30 | Presupuesto: 20000,00 €
ID: 5 | Nombre: CRM Interno | Fecha_Inicio: 2024-01-01 | Fecha_Fin: 2024-12-31 | Presupuesto: 45000,00 €
ID: 7 | Nombre: Tata Árboles | Fecha_Inicio: 2023-03-11 | Fecha_Fin: 2023-04-21 | Presupuesto: 15000,00 €

```

```
--- Gestión de Proyectos ---
1. Crear proyecto
2. Listar proyectos
3. Buscar proyecto por ID
4. Actualizar proyecto
5. Eliminar proyecto
0. Volver
Seleccione una opción: 4
Introduce el ID del proyecto que deseas editar:
7
Introduce el nuevo nombre del proyecto:
intercambio de cromos
Introduce la nueva fecha de inicio (yyyy-MM-dd):
2024-03-21
Introduce la nueva fecha de fin (yyyy-MM-dd):
2027-11-03
Introduce el nuevo presupuesto:
60000
Proyecto actualizado correctamente.

--- Gestión de Proyectos ---
1. Crear proyecto
2. Listar proyectos
3. Buscar proyecto por ID
4. Actualizar proyecto
5. Eliminar proyecto
0. Volver
Seleccione una opción: 5
Ingrese el ID del proyecto: 7
ID: 7 | Nombre: intercambio de cromos | Inicio: 2024-03-21 | Fin: 2027-11-03 | Presupuesto: 60000,00 €
```

```
--- Gestión de Proyectos ---
1. Crear proyecto
2. Listar proyectos
3. Buscar proyecto por ID
4. Actualizar proyecto
5. Eliminar proyecto
0. Volver
Seleccione una opción: 5
Ingrese el ID del proyecto a eliminar:
7
Proyecto eliminado correctamente.

--- Gestión de Proyectos ---
1. Crear proyecto
2. Listar proyectos
3. Buscar proyecto por ID
4. Actualizar proyecto
5. Eliminar proyecto
0. Volver
Seleccione una opción: 2

*** PROYECTOS ***
ID: 1 | Nombre: Sistema ERP | Fecha_Inicio: 2023-01-10 | Fecha_Fin: 2023-12-31 | Presupuesto: 50000,00 €
ID: 2 | Nombre: App Móvil | Fecha_Inicio: 2023-04-01 | Fecha_Fin: 2023-10-01 | Presupuesto: 30000,00 €
ID: 3 | Nombre: Campaña Publicitaria | Fecha_Inicio: 2023-06-01 | Fecha_Fin: 2023-09-30 | Presupuesto: 15000,00 €
ID: 4 | Nombre: Portal Web | Fecha_Inicio: 2022-02-15 | Fecha_Fin: 2022-12-30 | Presupuesto: 20000,00 €
ID: 5 | Nombre: CRM Interno | Fecha_Inicio: 2024-01-01 | Fecha_Fin: 2024-12-31 | Presupuesto: 45000,00 €
```

## 2.5 Invocación de procedimiento almacenado

```

== MENÚ PRINCIPAL ==
1. Gestión de Empleados
2. Gestión de Proyectos
3. Transferencia de Presupuesto
4. Procedimientos Almacenados
5. Salir
Seleccione una opción: 4

--- Procedimientos Almacenados ---
1. Actualizar salarios por departamento
2. Obtener empleados por proyecto
0. Volver
Seleccione una opción: 1
Departamento: Marketing
Porcentaje aumento: 30
Filas afectadas: 1

--- Procedimientos Almacenados ---
1. Actualizar salarios por departamento
2. Obtener empleados por proyecto
0. Volver
Seleccione una opción: 2
ID proyecto: 5
Número de empleados en el proyecto: 1

```

Result Grid | Filter Rows:  Edit: Export/Import:

|   | id_empleado | nombre      | departamento | salario | fecha_contratacion | activo |
|---|-------------|-------------|--------------|---------|--------------------|--------|
| ▶ | 1           | Ana Pérez   | Desarrollo   | 3000.00 | 2020-03-01         | 1      |
|   | 2           | Luis Gómez  | Marketing    | 2600.00 | 2019-07-15         | 1      |
|   | 3           | Marta López | Desarrollo   | 3240.00 | 2021-06-10         | 1      |
|   | 4           | Carlos Díaz | Soporte      | 1800.00 | 2018-02-20         | 1      |
|   | 5           | Sofía Ruiz  | Desarrollo   | 3000.00 | 2017-11-05         | 0      |
| * | 6           | sergio      | soporte      | 5000.00 | 2025-11-14         | 1      |
| * | 7           | iker        | soporte      | 6000.00 | 2025-11-14         | 0      |
| * | NULL        | NULL        | NULL         | NULL    | NULL               | NULL   |

## 2.6 Transferencia de Presupuesto

```
==> MENÚ PRINCIPAL ==>
1. Gestión de Empleados
2. Gestión de Proyectos
3. Transferencia de Presupuesto
4. Procedimientos Almacenados
5. Salir
Seleccione una opción: 3

--- Transferencia de Presupuesto ---
ID proyecto origen: 3
ID proyecto destino: 4
Dinero a transferir: 500
Transferencia de presupuesto completada correctamente.
```

## 2.7 Pool de conexiones configurado

```
package config;

import com.zaxxer.hikari.HikariConfig;
import com.zaxxer.hikari.HikariDataSource;

import java.io.InputStream;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.Properties;

public class DatabaseConfigPool {

    // Objeto del pool de conexiones (HikariDataSource)
    private static HikariDataSource dataSource;

    // Bloque estático: se ejecuta solo una vez cuando la clase se
    // carga
    static {
        try {
            Properties props = new Properties();
            props.load();
```

```
        try (InputStream input =
DatabaseConfigPool.class.getClassLoader().getResourceAsStream("config/
db.properties")) {
            // Carga de las propiedades desde el fichero de
configuración.
            // Se espera que 'db.url', 'db.user' y 'db.password'
estén presentes.
            props.load(input);
        } catch (Exception ex) {
            throw new RuntimeException("No se pudo encontrar el
archivo db.properties en config/");
        }

        // Configuración mínima de HikariCP. Se explican los
parámetros relevantes.
        HikariConfig config = new HikariConfig();
        config.setJdbcUrl(props.getProperty("db.url"));
// URL JDBC, p.ej. jdbc:mysql://host:3306/schema
        config.setUsername(props.getProperty("db.user"));
// Usuario de la BD
        config.setPassword(props.getProperty("db.password"));
// Contraseña de la BD
        config.setMaximumPoolSize(5);      // Máximo de 5 conexiones
activas
        config.setMinimumIdle(2);          // Al menos 2 conexiones
en espera
        config.setIdleTimeout(30000);       // Tiempo de inactividad
(30s)
        config.setMaxLifetime(1800000);    // Vida máxima de una
conexión (30 minutos)
        // Creamos el pool
        dataSource = new HikariDataSource(config);
    } catch (Exception e) {
        throw new RuntimeException("Error al inicializar
DatabaseConfigPool", e);
    }
}

public static Connection getConexion() {
    try{
        return dataSource.getConnection();
    }

} catch (SQLException e) {
```

```
        throw new RuntimeException("No se pudo obtener la
conexión", e);
    }
}
/**
 * Método para cerrar el pool al finalizar la aplicación
 */
public static void cerrarPool() {
    if (dataSource != null && !dataSource.isClosed()) {
        dataSource.close();
        System.out.println("Pool de conexiones cerrado
correctamente.");
    }
}
```

### 3. Explicación de Decisiones Técnicas

#### 1. ¿Por qué usar PreparedStatement en lugar de Statement?

Usar PreparedStatement en lugar de Statement es preferible porque separa los datos del código y así evita la inyección SQL, además, el motor puede precompilar y reutilizar el plan de ejecución, lo que mejora el rendimiento en consultas repetidas. También ayuda a mapear correctamente los tipos (fechas, binarios, números) y simplifica el mantenimiento al eliminar concatenaciones frágiles y facilitar ejecuciones en lote.

#### 2. ¿Qué ventajas tiene el pool de conexiones?

Un pool de conexiones aporta menor latencia al reutilizar conexiones en vez de crearlas y cerrarlas en cada operación, limita el número máximo de conexiones para proteger la base de datos, mejora la estabilidad mediante validaciones, keep-alive y timeouts, y aumenta la escalabilidad bajo carga concurrente.

#### 3. ¿En qué casos usarías procedimientos almacenados?

Usaría procedimientos almacenados cuando la lógica sea intensiva y cercana a los datos, cuando quiera reducir viajes de red ejecutando varios pasos de forma atómica en el servidor, cuando necesite seguridad extra exponiendo solo procedimientos, o cuando quiera encapsular una unidad de trabajo transaccional.

#### 4. ¿Por qué es importante el control de transacciones?

El control de transacciones es crucial para garantizar las propiedades ACID: atomicidad , consistencia, aislamiento y durabilidad. Permite recuperar ante errores con rollback, coordinar correctamente la concurrencia y ajustar el nivel de aislamiento al caso de uso.

## 4. Problemas Encontrados y Soluciones

Uno de los problemas que he tenido ha sido a la hora de hacer esto `obtenerPorId(int id) → devuelve Optional<T>`, no sabia como hacer para devolver eso porque nunca lo he usado, la forma en la que lo he resuelto ha sido con la ayuda de chatgpt y usando `return Optional.of(empleado);`

Otro de los problemas que tenia era que no me conectaba a la base de datos, porque no me estaba cogiendo la ruta adecuada del db.properties, try (`InputStream input = DatabaseConfigPool.class.getClassLoader().getResourceAsStream("config/db.properties")`), estaba poniendo mal la ruta porque tenia el db.properties metido en el mismo paquete de config, para resolverlo he cambiado el db.properties a la carpeta resources.