APLICACIÓN DE GESTIÓN DE COMPRAS.

Usue, Ana, Alex, Kamila
2º DAM, ACCESO A DATOS
Gestión de compras de una empresa

1. Descripción general del proyecto

Muchas empresas encuentran problemas a la hora de vender sus productos, ya que el proceso de compra se realiza de forma manual o en hojas de cálculo, lo que puede generar errores y pérdida o duplicación de información que puede ser crucial para la empresa.

El objetivo general de este proyecto es crear una aplicación que permita gestionar las compras que se realizan en la empresa, los proveedores y los productos, llegando así a optimizar el proceso de compraventa de la empresa.

Como objetivos específicos, se registrarán los proveedores con sus datos de contacto y dirección y se administrarán los productos con su precio, el stock y el proveedor que se encarga de proveerlo. Se generarán también unos reportes con los detalles de cada compra, la cantidad de producto vendido y el precio.

En conclusión, gracias a esta aplicación, las empresas podrán centralizar toda la información de sus compras, reducir los errores y controlar con mayor eficacia todas las acciones administrativas de la empresa.

2. Modelos Entidad-Relación

Se realizará el modelo Entidad-Relación, que servirá para tener una idea clara de cómo se tendrá que crear la base de datos y sus tablas.

Como entidades principales, se podrá identificar la figura del proveedor (Proveedores), el producto (Productos), la compra realizada (compras) y los detalles de la compra (Detalle-compra).

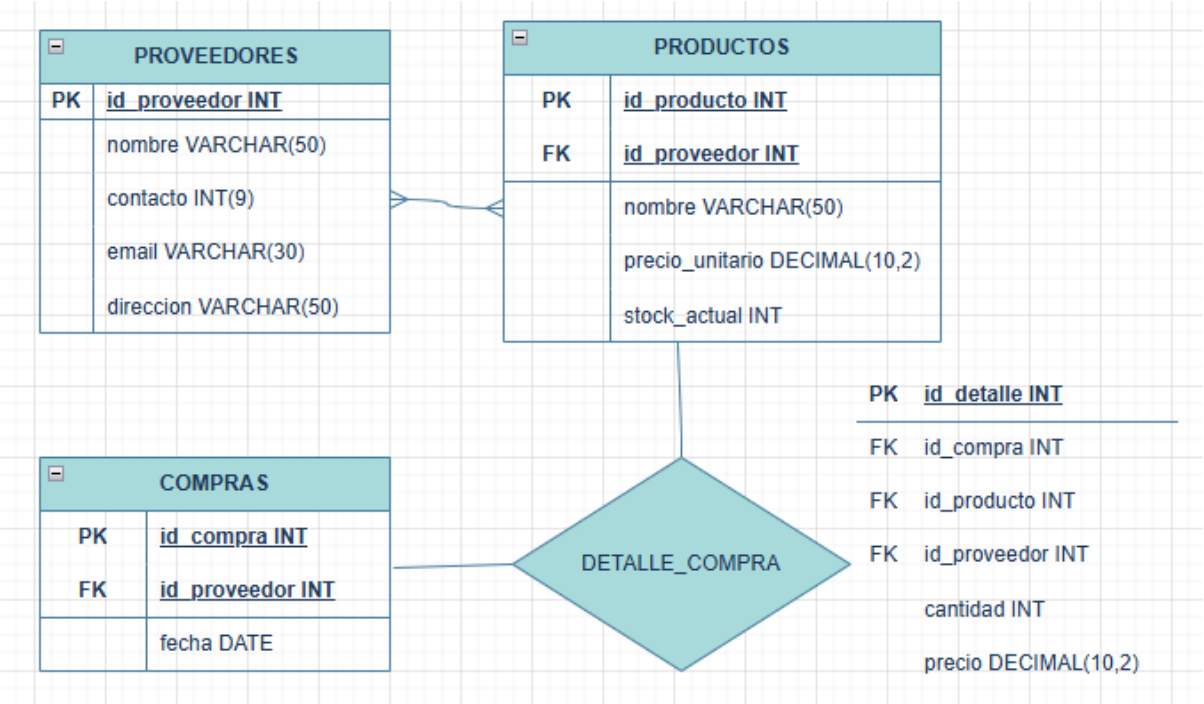
Atributos:

Entidad	Atributos principales
Proveedores	**Id_proveedor**, nombre, contacto, email, direccion
Productos	**Id_producto**, id_proveedor, nombre, precio_unitario, stock_actual
Compras	**Id_compra**, id_proveedor, fecha
Detalle_Compra	**Id_detalle**, id_compra, id_producto, id_proveedor, cantidad, precio

Relaciones entre entidades:

Entidad 1	Entidad 2	Tipo de relación	Cardinalidad
Proveedores	Productos	1 a N	Un proveedor puede ofrecer varios productos, pero cada producto pertenece a un solo proveedor.
Productos	Detalle_Compra	1 a N	Un producto puede estar en varios detalles de compra, pero cada detalle se asocia a un solo producto.
Compras	Detalle_Compra	1 a N	Una compra puede tener varios detalles, pero cada detalle pertenece a una sola compra.

Modelo Entidad-Relación:



3. Conexión desde la aplicación

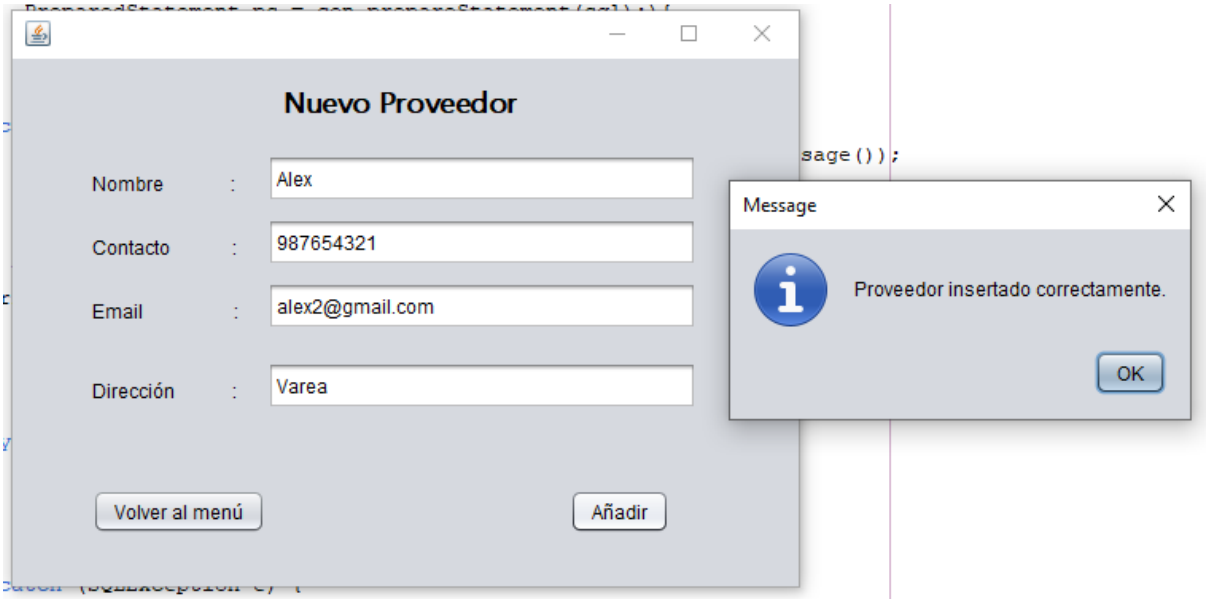
La conexión de esta aplicación se ha realizado mediante un controlador MySQL. Esta conexión permite realizar las operaciones CRUD(Insertar, Leer, Actualizar y Eliminar).

```
public class Conexion {

    private static String URL = "jdbc:mysql://localhost:3306/gestion_compras";
    private static String USER = "root";
    private static String PASSWORD = "";

    public static Connection getConnection() {
        try {
            return DriverManager.getConnection(url:URL, user:USER, password:PASSWORD);
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
```

A continuación, se muestra la correcta conexión con la base de datos y la opción de inserción de datos a la tabla con su correspondiente mensaje de “Proveedor insertado correctamente”.

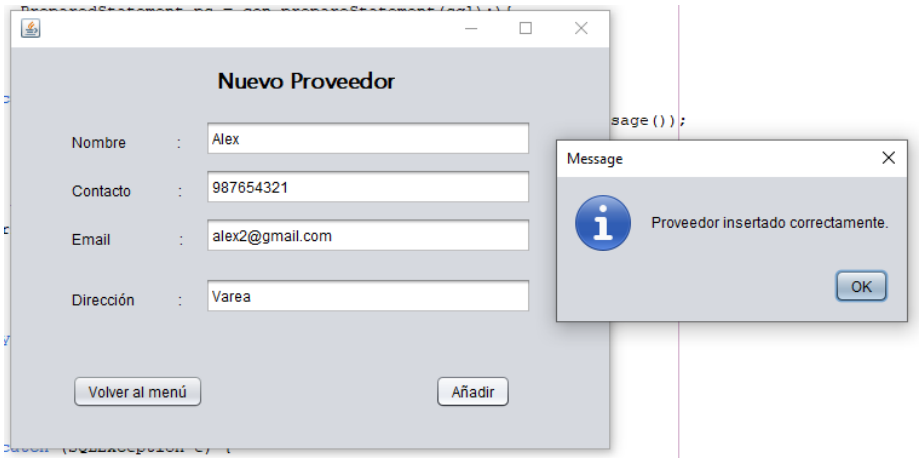


4. CRUD Implementado

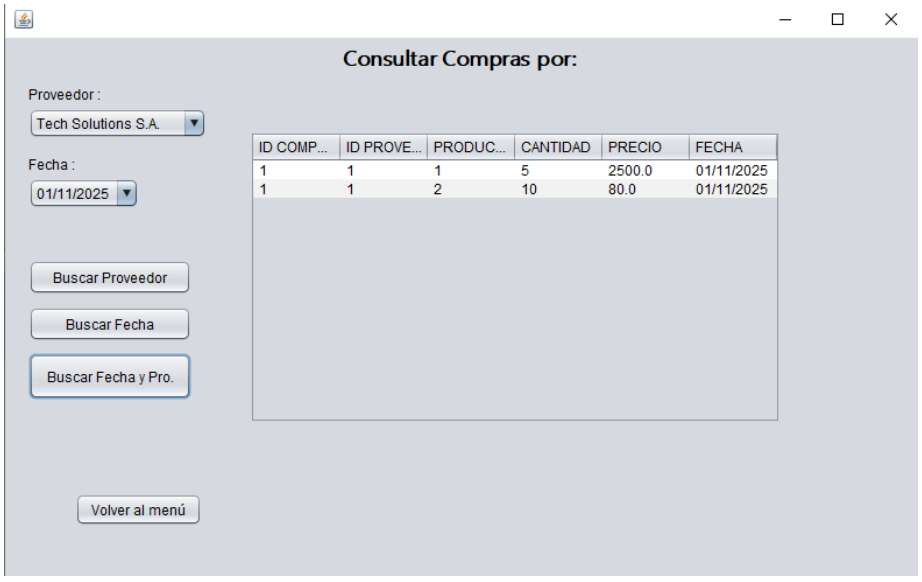
Se implementaron en este proyecto las cuatro operaciones básicas para la gestión correcta de los datos:

Operación	Descripción	Ejemplo
Insertar (Insert)	Se registran nuevos proveedores.	Alta de un nuevo proveedor.
Leer (Select)	Se consulta la información filtrada por proveedor, fecha o fecha y proveedor.	Búsqueda de compras por fecha y proveedor.
Actualizar (Update)	Permite actualizar el precio de un producto y modificar los datos de las compras realizadas.	Modificación del precio de un producto.
Eliminar (Delete)	Permite eliminar proveedores y registrarlos en una tabla de historicos.	Baja controlada de los proveedores.

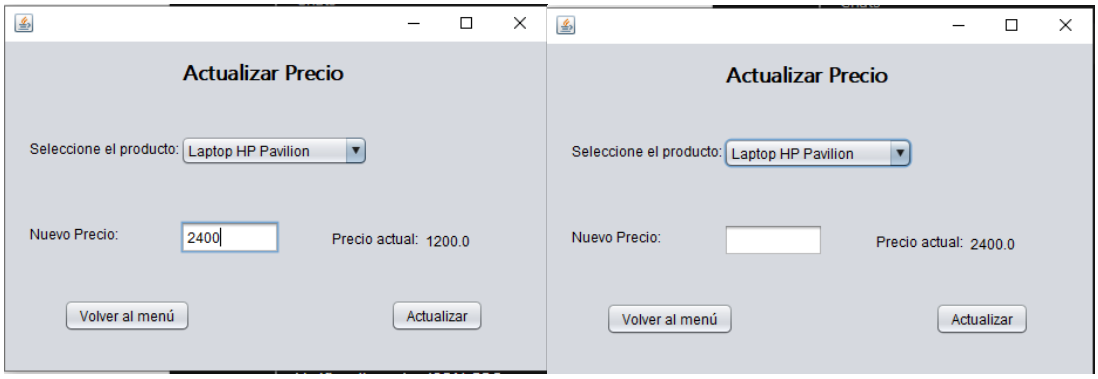
Insertar:



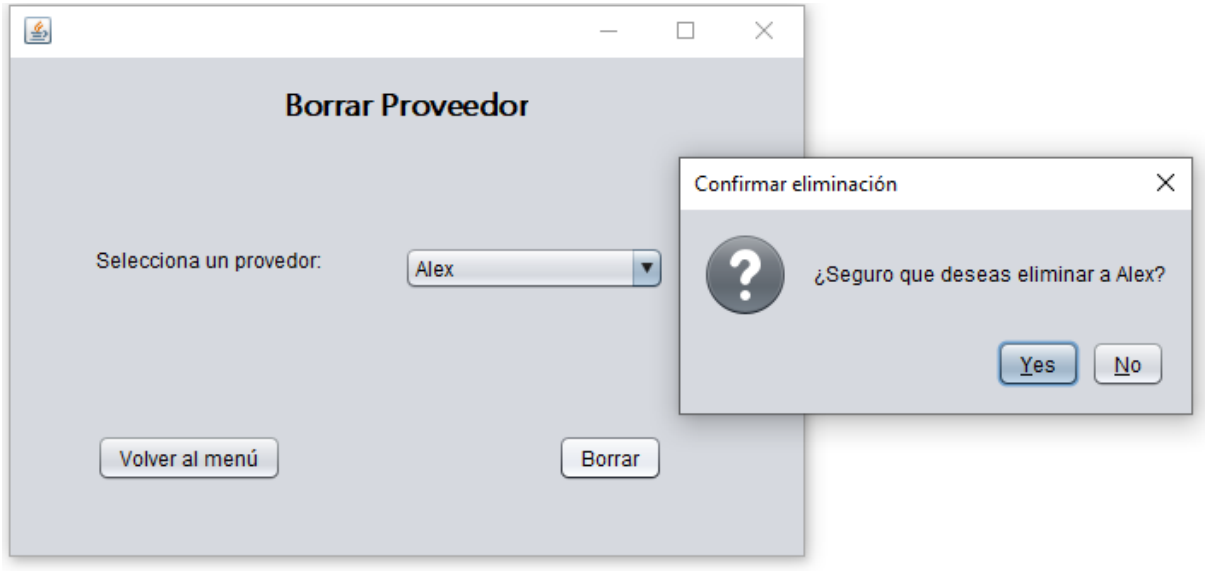
Leer:



Actualizar:



Borrar:



#	id_historico	id_proveedor	nombre	email	direccion	contacto	fecha_eliminacion
1	1	8	ana	dfgddgdgdfgdfg	dfgdfgdfgdfgfdgf	123.456.778	2025-11-07 12:47:23
2	2	9	Kamila	kamila@gmail.com	logrono	123.456.789	2025-11-10 09:28:55
3	3	10	Alex	alex@gmail.com	logrono	135.799.753	2025-11-23 09:29:14
4	4	5	Papelería Total	ventas@papeleriastotal.com	Calle Comercio 10	976.543.210	2025-11-10 09:29:25
5	5	1	Tech Solutions S.A.	contacto@techsolutions.com	Av. Siempre Viva 123	912.345.678	2025-11-03 12:29:44
6	6	2	Distribuidora Global	ventas@distribuidoraglobal.com	Calle Central 45	987.654.321	2025-11-10 09:32:36
7	7	11	Alex	alex2@gmail.com	Varea	987.654.321	2025-11-10 10:02:33

5. Procedimiento Almacenado

Para la eliminación controlada de los proveedores, se creó un procedimiento llamado eliminar_proveedor, que permite almacenar los proveedores eliminados en una tabla de históricos.

```sql
DELIMITER $$

CREATE OR REPLACE PROCEDURE eliminar_proveedor(IN p_id INT)
BEGIN
    DECLARE v_nombre VARCHAR(100);
    DECLARE v_email VARCHAR(100);
    DECLARE v_direccion VARCHAR(150);
    DECLARE v_contacto INT;

    -- Obtener datos del proveedor antes de eliminarlo
    SELECT nombre, email, direccion, contacto
    INTO v_nombre, v_email, v_direccion, v_contacto
    FROM proveedores
    WHERE id_proveedor = p_id;

    -- Insertar en tabla histórica
    INSERT INTO proveedores_historico (id_proveedor, nombre, email, direccion, contacto)
    VALUES (p_id, v_nombre, v_email, v_direccion, v_contacto);

    -- Eliminar del original
    DELETE FROM proveedores WHERE id_proveedor = p_id;
END$$

DELIMITER ;
```

6. Roles de equipo

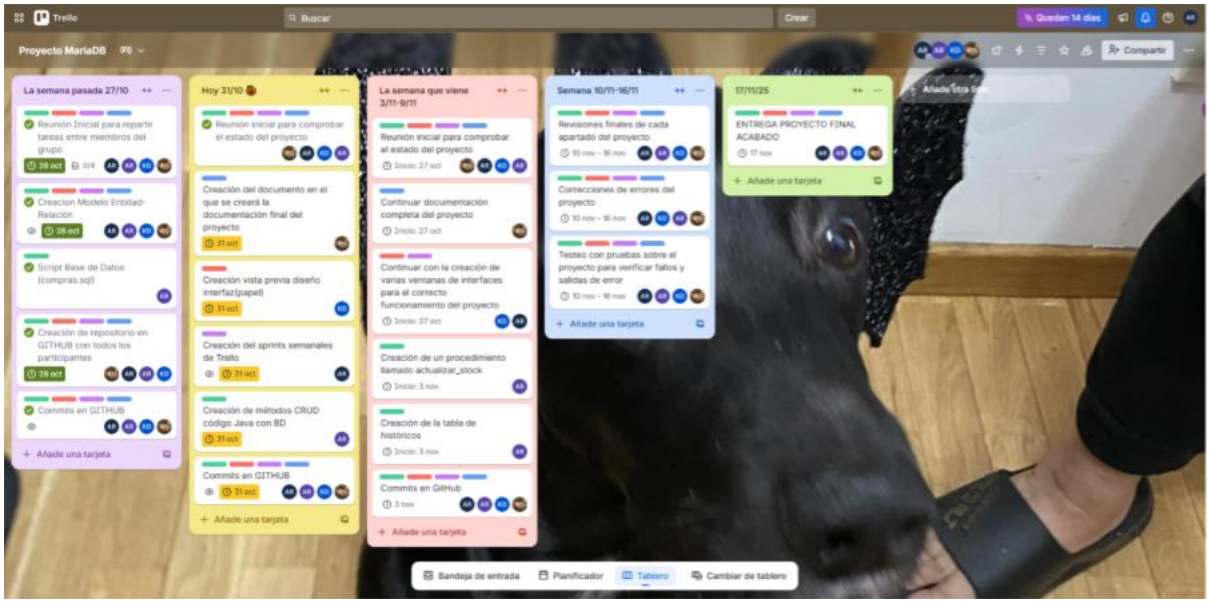
El trabajo se organizó teniendo en cuenta la metodología SCRUM y sus roles predefinidos:

Rol	Integrante	Responsabilidades
Product Owner	Usue	Descripción del proyecto, modelo E–R, documentación técnica, presentación final.
Scrum Masters	Ana y Kamila	Gestión del Trello, definición de sprints.
Developer SQL	Alex	Diseño de la base de datos, creación del script SQL, CRUD Y procedimiento almacenado.
Developer App	Ana y Kamila	Desarrollo de la aplicación y conexión con la base de datos, implementación del CRUD.

7. Sprint Backlog

El trabajo se organizó en la página de Trello, donde se incluyen las tareas específicas de cada uno y la realización de estas, todo organizado por días.

Durante el desarrollo, se realizó un seguimiento continuo de las tareas y avances en Trello.
Cada tarjeta representaba una actividad asignada a un miembro del equipo, marcando su progreso.


Trello
Buscar
Crear
Quedan 14 días
Proyecto MariaDB
Compartir
La semana pasada 27/10
Reunión Inicial para repartir tareas entre miembros del grupo
Creacion Modelo Entidad-Relación
Script Base de Datos (compras.sql)
Creación de repositorio en GITHUB con todos los participantes
Commits en GITHUB
Añade una tarjeta
Hoy 31/10
Reunión inicial para comprobar el estado del proyecto
Creación del documento en el que se creará la documentación final del proyecto
Creación vista previa diseño interfaz(papel)
Creación del sprints semanales de Trello
Creación de métodos CRUD código Java con BD
Commits en GITHUB
Añade una tarjeta
La semana que viene 3/11-9/11
Reunión inicial para comprobar el estado del proyecto
Continuar documentación completa del proyecto
Continuar con la creación de varias ventanas de interfaces para el correcto funcionamiento del proyecto
Creación de un procedimiento llamado actualizar_stock
Creación de la tabla de históricos
Commits en GitHub
Añade una tarjeta
Semana 10/11-16/11
Revisiones finales de cada apartado del proyecto
Correcciones de errores del proyecto
Testeo con pruebas sobre el proyecto para verificar fallos y salidas de error
Añade una tarjeta
17/11/25
ENTREGA PROYECTO FINAL ACABADO
Añade una tarjeta
Bandeja de entrada
Planificador
Tablero
Cambiar de tablero