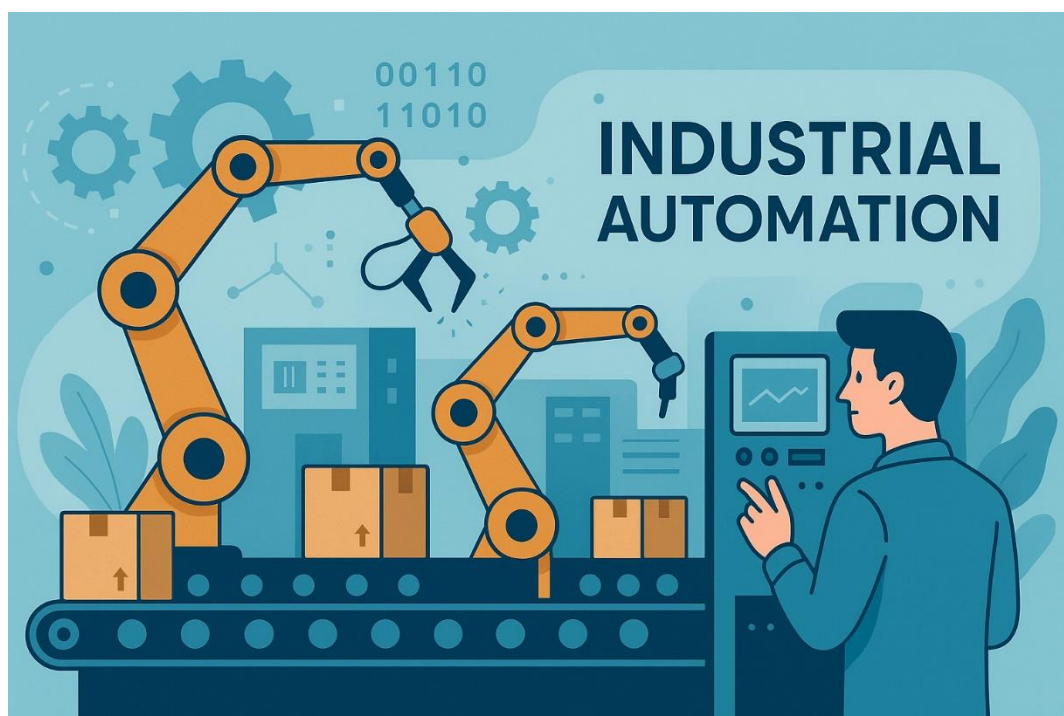


EMPRESA AUTOMATIZACIÓN INDUSTRIAL



https://github.com/Xurxo777/Proyecto_BBDD_XurxoBargeBlanco.git

Nome Alumno/a:

XURXO BARGE BLANCO

Xurxo barge blanco**Curso: 1º DAM****Materia: Bases de Datos – Proyecto Final 24/25****Contido**

1. Introducción	2
2. Descripción del Problema / Requisitos.....	2
3. Modelo Conceptual	3
4. Modelo Relacional.....	4
5. Proceso de Normalización.....	5
6. Script de Creación de la Base de Datos	6
7. Carga de Datos Inicial	17
8. Funciones y Procedimientos Almacenados.....	25
9. Triggers	33
10. Consultas SQL.....	37
11. Casos de Prueba y Simulación.....	45
12. Resultados y Verificación.....	51
13. Conclusiones y Mejoras Futuras.....	53
14. Enlace al Repositorio en GitHub.....	53

1. Introducción

O proxecto consiste nunha base de datos para xestionar os rexistros, datos de clientes y empregados y todo tipo de información de una empresa que se dedica a automatización industrial, facendo maquinas automatizadas, automatizando procesos en planta e ofrecendo un mantemento de estas ou de maquinas axenas a la empresa.

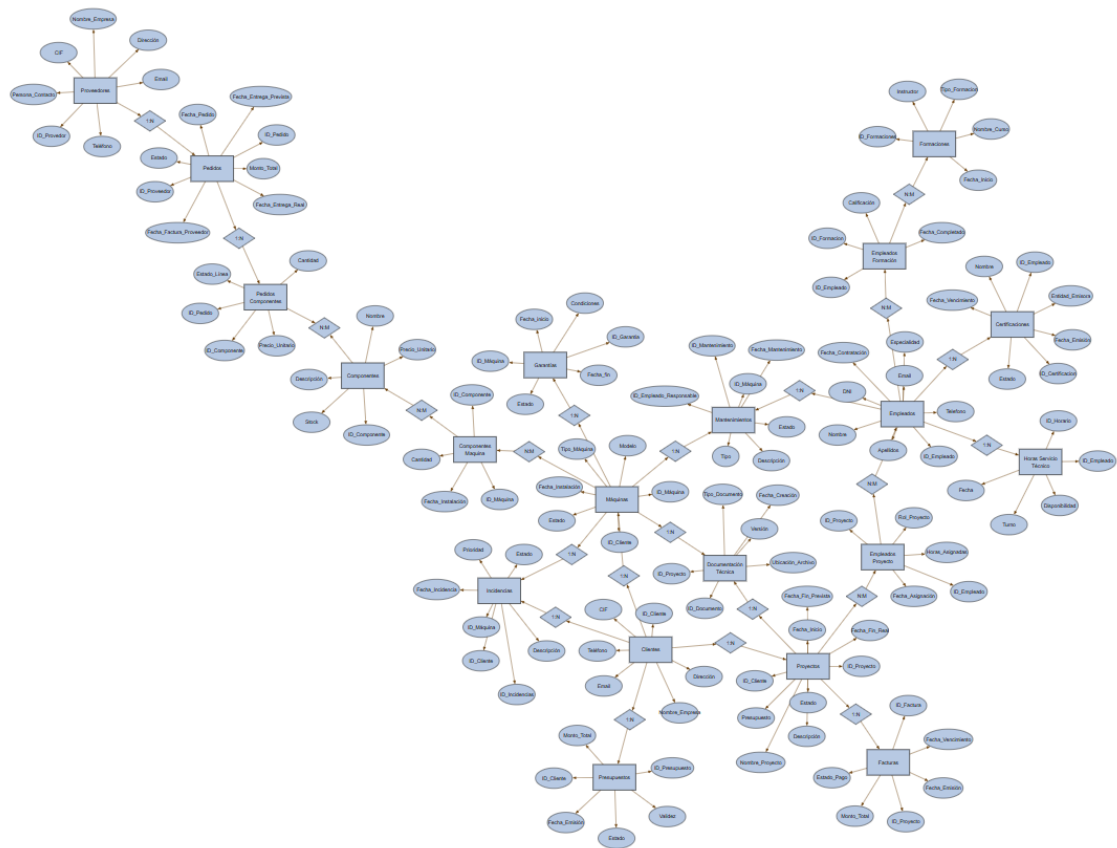
2. Descripción del Problema / Requisitos

O principal problema e unha mellor xestión da empresa xa que precisa xestionar gran cantidade de datos.

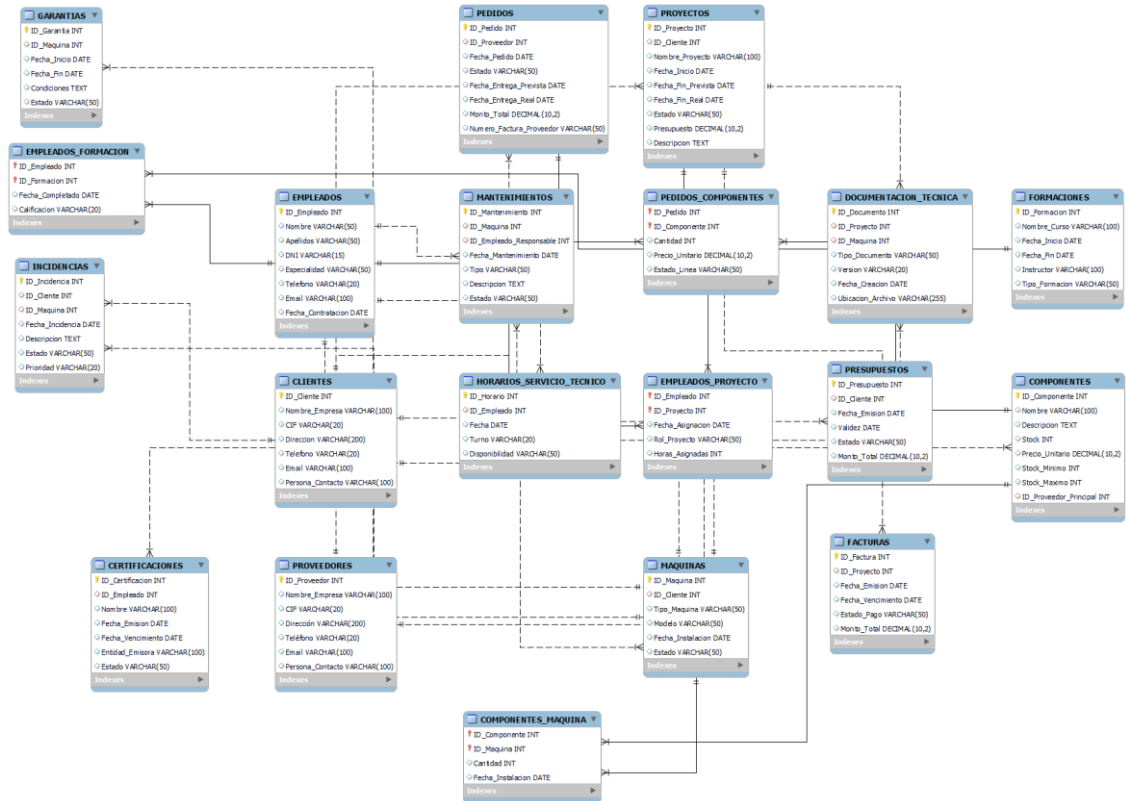
Características clave do negocio:

- Servicios de consultoría e diseño de solucións.
- Implementación de sistemas de control industrial.
- Programación de PLC's e robots.
- Mantemento preventivo e correctivo.
- Capacitación do persoal do cliente.
- Servicio de soporte técnico 24/7.

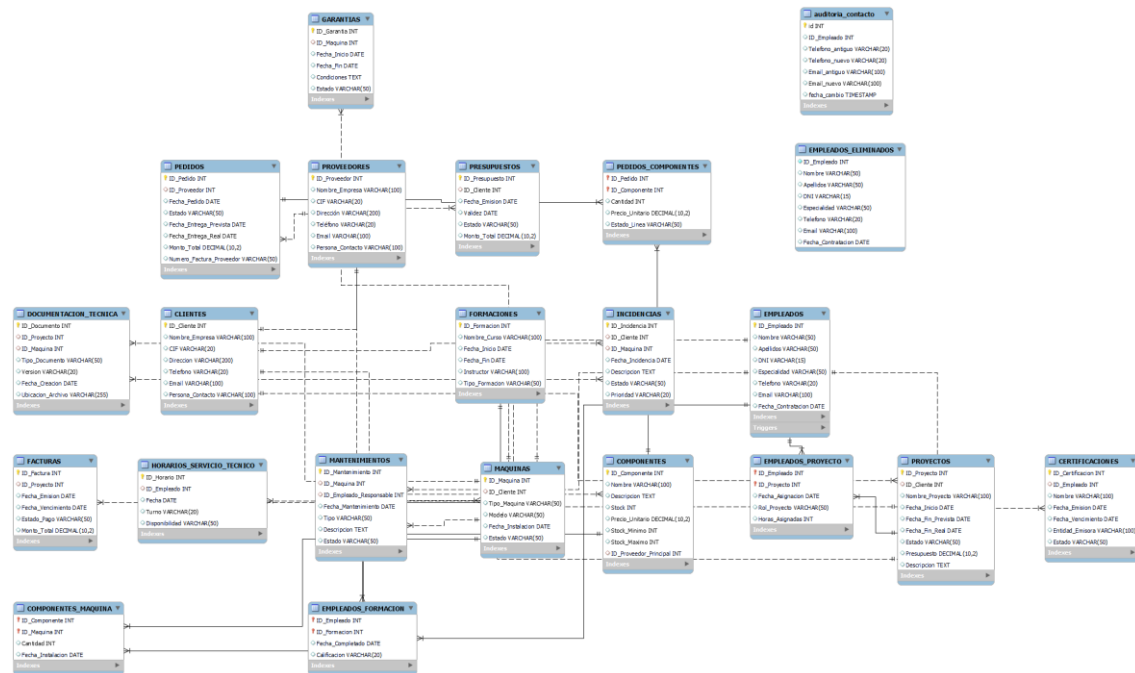
3. Modelo Conceptual



4. Modelo Relacional



En este modelo añadense 2 tablas implementadas na parte de triggers.



5. Proceso de Normalización

O proceso de normalización é unha técnica fundamental no deseño de bases de datos relacionais que buscan minimizar a redundancia de datos y asegurar a integridad da información. A través de diferentes formas normales (1FN, 2FN, 3FN...), garantízase que a estrutura da base de datos sea eficiente, coherente e escalable.

En este proxecto realizouse unha análise das seguintes relacións: CLIENTES, PROYECTOS, EMPLEADOS, EMPLEADOS_PROYECTO, MÁQUINAS, COMPONENTES, COMPONENTES_MÁQUINA, MANTENIMIENTOS, INCIDENCIAS, DOCUMENTACIÓN_TÉCNICA, PRESUPUESTOS, FACTURAS, PROVEEDORES, PEDIDOS, PEDIDOS_COMPONENTES, GARANTÍAS, FORMACIONES, EMPLEADOS_FORMACION, CERTIFICACIONES y HORARIOS_SERVICIO_TÉCNICO.

1ª Forma Normal (1FN)

Todas as tablas cumpren coa 1FN ao garantir que cada atributo contén un valor atómico e indivisible, sen repeticións de grupos de datos nin listas. Isto permite que cada celda da tabla conténha únicamente unha valor.

2ª Forma Normal (2FN)

Ao ter todas as tablas unha clave primaria claramente definida e non existir dependencias parciais (e dicir, que un atributo dependa solo dunha parte de unha clave composta), podemos afirmar que tamén se cumpre a 2FN. Aquelas tablas con claves compostas, como EMPLEADOS_PROYECTO, solo teñen atributos que dependen da totalidade da clave.

3ª Forma Normal (3FN)

As relacións tamén cumpren coa 3FN, xa que non existen dependencias transitivas (un atributo non clave que dependa doutro atributo non clave). A información está correctamente descomposta entre entidades separadas, como PROVEEDORES, COMPONENTES o PROYECTOS, evitando duplicidades e redundancia innecesaria.

Aplicación práctica

A normalización permitiu organizar a base de datos nunha estrutura clara onde:

- Os clientes e os proxectos están correctamente relacionados mediante claves externas.
- Os empregados poden participar en múltiples proxectos e formacións, mantendo a trazabilidade das súas actividades.
- As máquinas, componentes e mantementos están claramente separados pero interconectados, facilitando a súa xestión técnica.
- As incidencias, presupostos, facturas y pedidos permiten un seguimento completo da operativa e da relación comercial coa clientela e provedores.
- As garantías, documentación técnica e certificacións están asociadas aos seus elementos correspondentes, garantizando o control de calidade e trazabilidade normativa.

6. Script de Creación de la Base de Datos

```
CREATE DATABASE AUTOMATIZACION_INDUSTRIAL;
```

- *Comando de creación de la base de datos*

```
USE AUTOMATIZACION_INDUSTRIAL;
```

- *Comando de selección de la base de datos*

```
CREATE TABLE CLIENTES (  
    ID_Cliente INT PRIMARY KEY,  
    Nombre_Empresa VARCHAR(100),  
    CIF VARCHAR(20),  
    Direccion VARCHAR(200),  
    Telefono VARCHAR(20),  
    Email VARCHAR(100),  
    Persona_Contacto VARCHAR(100)  
);
```

- *Comando para la creación de la tabla “Clientes”, variables con tipo de variable acorde a la variable.*

CREATE TABLE PROYECTOS (**ID_Proyecto INT PRIMARY KEY,****ID_Cliente INT,****Nombre_Proyecto VARCHAR(100),****Fecha_Inicio DATE,****Fecha_Fin_Prevista DATE,****Fecha_Fin_Real DATE,****Estado VARCHAR(50),****Presupuesto DECIMAL(10,2),****Descripcion TEXT,****FOREIGN KEY (ID_Cliente) REFERENCES CLIENTES(ID_Cliente)****ON DELETE CASCADE ON UPDATE CASCADE****);**

- ***Comando para la creación de la tabla “Proyectos”, referenciada a la tabla clientes para saber los proyectos de cada cliente, uso de borrado y actualización en cascada.***

CREATE TABLE EMPLEADOS (**ID_Empleado INT PRIMARY KEY,****Nombre VARCHAR(50),****Apellidos VARCHAR(50),****DNI VARCHAR(15),****Especialidad VARCHAR(50),****Telefono VARCHAR(20),****Email VARCHAR(100),****Fecha_Contratacion DATE****);**

- ***Comando para la creación de la tabla “Empleados”.***


```
CREATE TABLE EMPLEADOS_PROYECTO (  
    ID_Empleado INT,  
    ID_Proyecto INT,  
    Fecha_Asignacion DATE,  
    Rol_Proyecto VARCHAR(50),  
    Horas_Asignadas INT,  
    PRIMARY KEY (ID_Empleado, ID_Proyecto),  
    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  
);
```

- ***Comando para la creación de la tabla “Empleados_Proyecto”, referenciada a la tabla Empleados y referencia a la tabla Proyecto para saber empleados en cada proyecto, uso de borrado y actualización en cascada.***

```
CREATE TABLE MAQUINAS (  
    ID_Maquina INT PRIMARY KEY,  
    ID_Cliente INT,  
    Tipo_Maquina VARCHAR(50),  
    Modelo VARCHAR(50),  
    Fecha_Instalacion DATE,  
    Estado VARCHAR(50),  
    FOREIGN KEY (ID_Cliente) REFERENCES CLIENTES(ID_Cliente)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Maquinas”, referenciada a la tabla clientes para saber las maquinas de los proyectos de cada cliente, uso de borrado y actualización en cascada.***

CREATE TABLE COMPONENTES (**ID_Componente INT PRIMARY KEY,****Nombre VARCHAR(100),****Descripcion TEXT,****Stock INT,****Precio_Unitario DECIMAL(10,2),****Stock_Minimo INT,****Stock_Maximo INT,****ID_Proveedor_Principal INT,****FOREIGN KEY (ID_Proveedor_Principal) REFERENCES PROVEEDORES(ID_Proveedor)****ON DELETE SET NULL ON UPDATE CASCADE****);**

- ***Comando para la creación de la tabla “Componentes ”, referenciada a la tabla proveedor para saber a que proveedor se adquirió cada pieza en caso de hacer una devolución, uso de actualización en cascada y eliminación dejando a null.***

CREATE TABLE COMPONENTES_MAQUINA (**ID_Componente INT,****ID_Maquina INT,****Cantidad INT,****Fecha_Instalacion DATE,****PRIMARY KEY (ID_Componente, ID_Maquina),****FOREIGN KEY (ID_Componente) REFERENCES COMPONENTES(ID_Componente)****ON DELETE CASCADE ON UPDATE CASCADE,****FOREIGN KEY (ID_Maquina) REFERENCES MAQUINAS(ID_Maquina)****ON DELETE CASCADE ON UPDATE CASCADE****);**

- ***Comando para la creación de la tabla “Componentes_Maquina ”, referenciada a la tabla Componentes para llevar un listado de los componentes de cada máquina, uso de borrado y actualización en cascada.***

CREATE TABLE MANTENIMIENTOS (

ID_Mantenimiento INT PRIMARY KEY,

ID_Maquina INT,

ID_Empleado_Responsable INT,

Fecha_Mantenimiento DATE,

Tipo VARCHAR(50),

Descripcion TEXT,

Estado VARCHAR(50),

FOREIGN KEY (ID_Maquina) REFERENCES MAQUINAS(ID_Maquina)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ID_Empleado_Responsable) REFERENCES EMPLEADOS(ID_Empleado)

ON DELETE SET NULL ON UPDATE CASCADE

);

- ***Comando para la creación de la tabla “Mantenimientos”, referenciada a la tabla Maquinas y Empleados para llevar un listado de los empleados que hacen un mantenimiento y a que máquina en concreto, uso de actualización en cascada y eliminación dejando a null.***

```
CREATE TABLE INCIDENCIAS (  
  ID_Incidencia INT PRIMARY KEY,  
  ID_Cliente INT,  
  ID_Maquina INT,  
  Fecha_Incidencia DATE,  
  Descripcion TEXT,  
  Estado VARCHAR(50),  
  Prioridad VARCHAR(20),  
  FOREIGN KEY (ID_Cliente) REFERENCES CLIENTES(ID_Cliente)  
    ON DELETE SET NULL ON UPDATE CASCADE,  
  FOREIGN KEY (ID_Maquina) REFERENCES MAQUINAS(ID_Maquina)  
    ON DELETE SET NULL ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Incidencias”, referenciada a la tabla Clientes y Máquinas para llevar listado a que cliente y máquinas corresponde cada incidencia, uso de actualización en cascada y eliminación dejando a null.***

```
CREATE TABLE DOCUMENTACION_TECNICA (  
  ID_Documento INT PRIMARY KEY,  
  ID_Proyecto INT,  
  ID_Maquina INT,  
  Tipo_Documento VARCHAR(50),  
  Version VARCHAR(20),  
  Fecha_Creacion DATE,  
  Ubicacion_Archivo VARCHAR(255),  
  FOREIGN KEY (ID_Proyecto) REFERENCES PROYECTOS(ID_Proyecto)  
    ON DELETE SET NULL ON UPDATE CASCADE,  
  FOREIGN KEY (ID_Maquina) REFERENCES MAQUINAS(ID_Maquina)  
    ON DELETE SET NULL ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Documentación_Técnica”, referenciada a la tabla Proyecto y Máquinas para llevar listado las documentaciones de los proyectos y sus máquinas, uso de actualización en cascada y eliminación dejando a null.***

```
CREATE TABLE PRESUPUESTOS (  
  ID_Presupuesto INT PRIMARY KEY,  
  ID_Cliente INT,  
  Fecha_Emision DATE,  
  Validez DATE,  
  Estado VARCHAR(50),  
  Monto_Total DECIMAL(10,2),  
  FOREIGN KEY (ID_Cliente) REFERENCES CLIENTES(ID_Cliente)  
  ON DELETE SET NULL ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Presupuestos”, referenciada a la tabla Clientes para saber a que cliente corresponde cada presupuesto de un proyecto, uso de actualización en cascada y eliminación dejando a null.***

```
CREATE TABLE FACTURAS (  
  ID_Factura INT PRIMARY KEY,  
  ID_Proyecto INT,  
  Fecha_Emision DATE,  
  Fecha_Vencimiento DATE,  
  Estado_Pago VARCHAR(50),  
  Monto_Total DECIMAL(10,2),  
  FOREIGN KEY (ID_Proyecto) REFERENCES PROYECTOS(ID_Proyecto)  
  ON DELETE SET NULL ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Facturas”, referenciada a la tabla Proyectos para guardar las facturas de los proyectos, uso de actualización en cascada y eliminación dejando a null.***

```
CREATE TABLE PROVEEDORES (  
  ID_Proveedor INT PRIMARY KEY,  
  Nombre_Empresa VARCHAR(100),  
  CIF VARCHAR(20),  
  Dirección VARCHAR(200),  
  Teléfono VARCHAR(20),  
  Email VARCHAR(100),  
  Persona_Contacto VARCHAR(100)  
);
```

- ***Comando para la creación de la tabla “Proveedores”,***

```
CREATE TABLE PEDIDOS (  
  ID_Pedido INT PRIMARY KEY,  
  ID_Proveedor INT,  
  Fecha_Pedido DATE,  
  Estado VARCHAR(50),  
  Fecha_Entrega_Prevista DATE,  
  Fecha_Entrega_Real DATE,  
  Monto_Total DECIMAL(10,2),  
  Numero_Factura_Proveedor VARCHAR(50),  
  FOREIGN KEY (ID_Proveedor) REFERENCES PROVEEDORES(ID_Proveedor)  
  ON DELETE SET NULL ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Proveedores”, referenciada a la tabla Proveedor para saber quien es el vendedor de cada pedido, uso de actualización en cascada y eliminación dejando a null.***

```
CREATE TABLE PEDIDOS_COMPONENTES (  
    ID_Pedido INT,  
    ID_Componente INT,  
    Cantidad INT,  
    Precio_Unitario DECIMAL(10,2),  
    Estado_Linea VARCHAR(50),  
    PRIMARY KEY (ID_Pedido, ID_Componente),  
    FOREIGN KEY (ID_Pedido) REFERENCES PEDIDOS(ID_Pedido)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (ID_Componente) REFERENCES COMPONENTES(ID_Componente)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Pedidos_Componentes”, referenciada a la tabla Pedidos y Componentes para saber que componentes pertenecen a cada pedido, uso de actualización y eliminación en cascada.***

```
CREATE TABLE GARANTIAS (  
    ID_Garantia INT PRIMARY KEY,  
    ID_Maquina INT,  
    Fecha_Inicio DATE,  
    Fecha_Fin DATE,  
    Condiciones TEXT,  
    Estado VARCHAR(50),  
    FOREIGN KEY (ID_Maquina) REFERENCES MAQUINAS(ID_Maquina)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Garantías”, referenciada a la tabla Máquina para saber a que máquina le pertenece cada garantía o su estado, uso de actualización y eliminación en cascada.***

```
CREATE TABLE FORMACIONES (  
    ID_Formacion INT PRIMARY KEY,  
    Nombre_Curso VARCHAR(100),  
    Fecha_Inicio DATE,  
    Fecha_Fin DATE,  
    Instructor VARCHAR(100),  
    Tipo_Formacion VARCHAR(50)  
);
```

- ***Comando para la creación de la tabla “Formaciones”,***

```
CREATE TABLE EMPLEADOS_FORMACION (  
    ID_Empleado INT,  
    ID_Formacion INT,  
    Fecha_Completado DATE,  
    Calificacion VARCHAR(20),  
    PRIMARY KEY (ID_Empleado, ID_Formacion),  
    FOREIGN KEY (ID_Empleado) REFERENCES EMPLEADOS(ID_Empleado)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (ID_Formacion) REFERENCES FORMACIONES(ID_Formacion)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Empleados_Formación”, referenciada a la tabla Empleados y Formaciones para saber que formación tiene cada empleado de la empresa, uso de actualización y eliminación en cascada.***


```
CREATE TABLE CERTIFICACIONES (  
  ID_Certificacion INT PRIMARY KEY,  
  ID_Empleado INT,  
  Nombre VARCHAR(100),  
  Fecha_Emision DATE,  
  Fecha_Vencimiento DATE,  
  Entidad_Emisora VARCHAR(100),  
  Estado VARCHAR(50),  
  FOREIGN KEY (ID_Empleado) REFERENCES EMPLEADOS(ID_Empleado)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Certificaciones”, referenciada a la tabla Empleados para saber si la formación de estos esta certificada correctamente, uso de actualización y eliminación en cascada.***

```
CREATE TABLE HORARIOS_SERVICIO_TECNICO (  
  ID_Horario INT PRIMARY KEY,  
  ID_Empleado INT,  
  Fecha DATE,  
  Turno VARCHAR(20),  
  Disponibilidad VARCHAR(50),  
  FOREIGN KEY (ID_Empleado) REFERENCES EMPLEADOS(ID_Empleado)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- ***Comando para la creación de la tabla “Horarios_Servicio_Técnico”, referenciada a la tabla Empleados para tener un seguimiento de las horas de trabajo de cada técnico, uso de actualización y eliminación en cascada.***

7. Carga de Datos Inicial

- ***Inserción de clientes: almacena datos básicos das empresas clientes da empresa de automatización, incluíndo contacto e dirección.***

INSERT INTO CLIENTES (ID_Cliente, Nombre_Empresa, CIF, Direccion, Telefono, Email, Persona_Contacto) VALUES

(1, 'Empresa 1', 'A00000001', 'Rúa Exemplo 1, Cidade', '+34 612345678', 'contacto1@empresa.com', 'Persoa 1'),

(2, 'Empresa 2', 'A00000002', 'Rúa Exemplo 2, Cidade', '+34 623456789', 'contacto2@empresa.com', 'Persoa 2'),

(3, 'Empresa 3', 'A00000003', 'Rúa Exemplo 3, Cidade', '+34 634567890', 'contacto3@empresa.com', 'Persoa 3'),

(4, 'Empresa 4', 'A00000004', 'Rúa Exemplo 4, Cidade', '+34 645678901', 'contacto4@empresa.com', 'Persoa 4'),

(5, 'Empresa 5', 'A00000005', 'Rúa Exemplo 5, Cidade', '+34 656789012', 'contacto5@empresa.com', 'Persoa 5'),

(6, 'Empresa 6', 'A00000006', 'Rúa Exemplo 6, Cidade', '+34 656789045', 'contacto6@empresa.com', 'Persoa 6'),

(7, 'Empresa 7', 'A00000007', 'Rúa Exemplo 7, Cidade', '+34 667890123', 'contacto7@empresa.com', 'Persoa 7'),

(8, 'Empresa 8', 'A00000008', 'Rúa Exemplo 8, Cidade', '+34 678901234', 'contacto8@empresa.com', 'Persoa 8');

- ***Inserción de empregados: información dos empregados da empresa, co seu DNI, especialidade, datos de contacto e data de contratación.***

INSERT INTO EMPLEADOS (ID_Empleado, Nombre, Apellidos, DNI, Especialidad, Telefono, Email, Fecha_Contratacion) VALUES

(1, 'Empleado1', 'Apellido1', '12345678X', 'Electrónica', '+34 600111222', 'empleado1@empresa.com', '2019-03-15'),

(2, 'Empleado2', 'Apellido2', '23456789Y', 'Mecánica', '+34 600222333', 'empleado2@empresa.com', '2018-07-10'),

(3, 'Empleado3', 'Apellido3', '34567890Z', 'Programación', '+34 600333444', 'empleado3@empresa.com', '2020-11-01'),

(4, 'Empleado4', 'Apellido4', '45678901W', 'Robótica', '+34 600444555', 'empleado4@empresa.com', '2021-02-20'),

(5, 'Empleado5', 'Apellido5', '56789012V', 'Electrónica', '+34 600555666', 'empleado5@empresa.com', '2022-09-05'),

(6, 'Empleado6', 'Apellido6', '56789012M', 'Gestión', '+34 600545666', 'empleado6@empresa.com', '2027-06-02'),

(7, 'Empleado7', 'Apellido7', '67890123N', 'Automatización', '+34 600666777', 'empleado7@empresa.com', '2023-05-10'),

(8, 'Empleado8', 'Apellido8', '78901234P', 'Electrónica', '+34 600777888', 'empleado8@empresa.com', '2020-10-01');

- ***Inserción de provedores: empresas que proveen compoñentes ou servizos, cos seus datos de contacto e persoas de referencia.***

INSERT INTO PROVEEDORES (ID_Proveedor, Nombre_Empresa, CIF, Dirección, Teléfono, Email, Persona_Contacto) VALUES

(1, 'Proveedor 1', 'P0000001', 'Rúa Proveedor 1', '+34 912345678', 'contacto@proveedor1.com', 'Contacto 1'),

(2, 'Proveedor 2', 'P0000002', 'Rúa Proveedor 2', '+34 923456789', 'contacto@proveedor2.com', 'Contacto 2'),

(3, 'Proveedor 3', 'P0000003', 'Rúa Proveedor 3', '+34 934567890', 'contacto@proveedor3.com', 'Contacto 3'),

(4, 'Proveedor 4', 'P0000004', 'Rúa Proveedor 4', '+34 945678901', 'contacto@proveedor4.com', 'Contacto 4'),

(5, 'Proveedor 5', 'P0000005', 'Rúa Proveedor 5', '+34 956789012', 'contacto@proveedor5.com', 'Contacto 5');

- ***Inserción de compoñentes: detalles dos compoñentes dispoñibles en stock, prezo, stock mínimo/máximo e provedores principais.***

INSERT INTO COMPONENTES (ID_Componente, Nombre, Descripcion, Stock, Precio_Unitario, Stock_Minimo, Stock_Maximo, ID_Proveedor_Principal) VALUES

(1, 'Componente A', 'Componente para control', 50, 25.50, 10, 100, 1),
(2, 'Componente B', 'Componente eléctrico', 30, 12.75, 5, 50, 2),
(3, 'Componente C', 'Sensor de temperatura', 40, 15.90, 10, 80, 3),
(4, 'Componente D', 'Relé de control industrial', 20, 9.80, 5, 40, 1),
(5, 'Componente E', 'Motor paso a paso', 15, 45.00, 3, 20, 4),
(6, 'Componente F', 'Pantalla HMI 7"', 10, 120.00, 2, 15, 5),
(7, 'Componente G', 'Variador de frecuencia', 8, 220.50, 2, 10, 3),
(8, 'Componente H', 'Interruptor de fin de carreira', 25, 6.50, 5, 30, 2),
(9, 'Componente I', 'Placa Arduino Mega', 35, 32.00, 10, 50, 4),
(10, 'Componente J', 'Módulo Wi-Fi ESP32', 60, 8.25, 15, 100, 5),
(11, 'Componente K', 'Fonte de alimentación 24V', 18, 55.30, 5, 25, 1),
(12, 'Componente L', 'Encoder rotativo óptico', 12, 27.80, 3, 20, 2);

- ***Inserción de máquinas: máquinas instaladas nos clientes, co tipo, modelo, data de instalación e estado operativo.***

```
INSERT INTO MAQUINAS (ID_Maquina, ID_Cliente, Tipo_Maquina, Modelo, Fecha_Instalacion, Estado) VALUES
```

```
(1, 1, 'Robot Industrial', 'Model X', '2023-01-15', 'Operativa'),  
(2, 2, 'CNC', 'Model Y', '2022-11-10', 'Mantenimiento'),  
(3, 3, 'Impresora 3D', 'Delta Pro', '2023-05-22', 'Operativa'),  
(4, 1, 'PLC Siemens', 'S7-1200', '2021-09-30', 'Operativa'),  
(5, 4, 'Brazo Robótico', 'UR5e', '2022-02-14', 'Mantenimiento'),  
(6, 5, 'Sistema SCADA', 'WinCC Advanced', '2023-08-01', 'Operativa'),  
(7, 2, 'Fresadora CNC', 'Model Z', '2020-12-20', 'Averiadada'),  
(8, 3, 'Robot Colaborativo', 'Cobot A1', '2024-01-10', 'Operativa'),  
(9, 4, 'Controlador PID', 'Omron E5CC', '2022-06-18', 'Mantenimiento'),  
(10, 5, 'Sistema de Visión', 'Cognex In-Sight', '2023-03-03', 'Operativa');
```

- ***Inserción de proxectos: proxectos que a empresa está a levar a cabo para os clientes, con datas, estado, orzamento e descrición.***

```
INSERT INTO PROYECTOS (ID_Proyecto, ID_Cliente, Nombre_Proyecto, Fecha_Inicio, Fecha_Fin_Prevista, Fecha_Fin_Real, Estado, Presupuesto, Descripcion) VALUES
```

```
(1, 1, 'Automatización Línea 1', '2024-01-10', '2024-06-10', NULL, 'En Progreso', 15000.00, 'Proyecto de instalación de robots en línea 1'),  
(2, 2, 'Actualización Máquina CNC', '2023-09-05', '2024-03-01', '2024-02-28', 'Finalizado', 8000.00, 'Actualización de software y hardware CNC'),  
(3, 3, 'Integración SCADA', '2024-02-01', '2024-07-01', NULL, 'En Progreso', 12000.00, 'Integración de SCADA en planta de producción'),  
(4, 4, 'Instalación Brazo Robótico', '2023-06-15', '2023-12-15', '2023-12-20', 'Finalizado', 9500.00, 'Instalación e posta en marcha de robot colaborativo');
```

- **Relación entre empregados e proxectos: asignación de empregados a proxectos con datas, rol e horas asignadas.**

INSERT INTO EMPLEADOS_PROYECTO (ID_Empleado, ID_Proyecto, Fecha_Asignacion, Rol_Proyecto, Horas_Asignadas) VALUES

(1, 1, '2024-01-15', 'Jefe de Proyecto', 120),
(2, 1, '2024-02-01', 'Técnico', 100),
(3, 2, '2023-09-10', 'Programador', 90),
(4, 3, '2024-02-05', 'Programador', 100),
(5, 4, '2023-06-20', 'Instalador', 80);

- **Mantementos realizados ou pendentes sobre máquinas, indicando quen é responsable, tipo, data e estado.**

INSERT INTO MANTENIMIENTOS (ID_Mantenimiento, ID_Maquina, ID_Empleado_Responsable, Fecha_Mantenimiento, Tipo, Descripcion, Estado) VALUES

(1, 1, 1, '2024-03-15', 'Preventivo', 'Revisión trimestral del robot', 'Realizado'),
(2, 2, 2, '2024-04-01', 'Correctivo', 'Reparación de fallo en eje', 'Pendiente'),
(3, 5, 5, '2024-05-10', 'Correctivo', 'Substitución de piezas desgastadas', 'Realizado'),
(4, 6, 6, '2024-05-18', 'Preventivo', 'Revisión mensual sistema SCADA', 'Pendiente');

- **Incidencias rexistradas para clientes e máquinas, con data, descrición, estado e prioridade.**

INSERT INTO INCIDENCIAS (ID_Incidencia, ID_Cliente, ID_Maquina, Fecha_Incidencia, Descripcion, Estado, Prioridad) VALUES

(1, 1, 1, '2024-04-10', 'Fallo no sistema de control', 'Abierta', 'Alta'),
(2, 2, 2, '2024-04-12', 'Problema coa alimentación eléctrica', 'Cerrada', 'Media'),
(3, 3, 3, '2024-05-15', 'Problemas de calibrado', 'Abierta', 'Media'),
(4, 5, 6, '2024-05-19', 'Erro de conexión coa rede', 'Abierta', 'Alta');

- **Documentación técnica asociada a proxectos e máquinas, incluíndo tipo de documento, versión, data e ubicación do arquivo.**

INSERT INTO DOCUMENTACION_TECNICA (ID_Documento, ID_Proyecto, ID_Maquina, Tipo_Documento, Version, Fecha_Creacion, Ubicacion_Archivo) VALUES

(1, 1, 1, 'Manual Técnico', 'v1.0', '2024-01-05', '/docs/manual_robot_x.pdf'),
(2, 2, 2, 'Plano Mecánico', 'v2.1', '2023-08-20', '/docs/plano_cnc_y.pdf'),
(3, 3, 6, 'Manual SCADA', 'v1.1', '2024-02-05', '/docs/manual_scada.pdf'),
(4, 4, 5, 'Esquema Eléctrico', 'v3.0', '2023-06-16', '/docs/esquema_robot.pdf');

- **Presupostos emitidos a clientes, con data, validez, estado (aprobado, rexeitado) e importe total.**

INSERT INTO PRESUPUESTOS (ID_Presupuesto, ID_Cliente, Fecha_Emision, Validez, Estado, Monto_Total) VALUES

(1, 1, '2023-12-01', '2024-03-01', 'Aprobado', 15000.00),
(2, 2, '2023-08-15', '2023-11-15', 'Rechazado', 8000.00);

- **Facturas xeradas por proxectos, con datas de emisión e vencemento, estado do pago e importe.**

INSERT INTO FACTURAS (ID_Factura, ID_Proyecto, Fecha_Emision, Fecha_Vencimiento, Estado_Pago, Monto_Total) VALUES

(1, 1, '2024-03-01', '2024-04-01', 'Pagada', 15000.00),
(2, 2, '2024-01-15', '2024-02-15', 'Pendiente', 8000.00),
(3, 3, '2024-05-10', '2024-06-10', 'Pendiente', 12000.00),
(4, 4, '2023-12-21', '2024-01-21', 'Pagada', 9500.00);

- ***Pedidos feitos a provedores, con datas, estado do pedido, datas previstas e reais de entrega, importe total e número de factura do provedor.***

```
INSERT INTO PEDIDOS (ID_Pedido, ID_Proveedor, Fecha_Pedido, Estado,  
Fecha_Entrega_Prevista, Fecha_Entrega_Real, Monto_Total, Numero_Factura_Proveedor)  
VALUES
```

```
(1, 1, '2024-03-20', 'Entregado', '2024-04-05', '2024-04-04', 1275.00, 'F2024-001'),  
(2, 2, '2024-04-01', 'En Proceso', '2024-04-20', NULL, 650.00, 'F2024-002'),  
(3, 3, '2024-04-15', 'Entregado', '2024-04-30', '2024-04-29', 330.00, 'F2024-003'),  
(4, 4, '2024-05-01', 'Pendiente', '2024-05-20', NULL, 890.00, 'F2024-004');
```

- ***Detalle dos compoñentes incluídos en cada pedido, con cantidade, prezo unitario e estado da liña.***

```
INSERT INTO PEDIDOS_COMPONENTES (ID_Pedido, ID_Componente, Cantidad,  
Precio_Unitario, Estado_Linea) VALUES
```

```
(1, 1, 50, 25.50, 'Recibido'),  
(2, 2, 30, 12.75, 'Pendiente'),  
(3, 3, 20, 15.90, 'Recibido'),  
(4, 4, 10, 9.80, 'Pendiente');
```

- ***Garantías asociadas a máquinas, con datas de inicio e fin, condicións e estado da garantía.***

```
INSERT INTO GARANTIAS (ID_Garantia, ID_Maquina, Fecha_Inicio, Fecha_Fin, Condiciones,  
Estado) VALUES
```

```
(1, 1, '2023-01-15', '2025-01-14', 'Garantía completa por dos anos', 'Activa'),  
(2, 2, '2022-11-10', '2024-11-09', 'Garantía limitada por piezas', 'Expirada'),  
(3, 3, '2024-01-10', '2026-01-10', 'Completa', 'Inclúe pezas e man de obra'),  
(4, 5, '2023-07-01', '2025-07-01', 'Limitada', 'Só pezas principais');
```


- **Formacións dispoñibles, co nome, datas, instructor e tipo (técnica, seguridade, etc.).**

INSERT INTO FORMACIONES (ID_Formacion, Nombre_Curso, Fecha_Inicio, Fecha_Fin, Instructor, Tipo_Formacion) VALUES

(1, 'Formación en Robótica Industrial', '2023-09-01', '2023-09-15', 'Ing. Pérez', 'Técnica'),
(2, 'Seguridad en el Trabajo', '2024-01-10', '2024-01-20', 'Dr. López', 'Seguridad');

- **Relación entre empregados e formacións, indicando data de finalización e calificación obtida.**

INSERT INTO EMPLEADOS_FORMACION (ID_Empleado, ID_Formacion, Fecha_Completado, Calificacion) VALUES

(1, 1, '2023-09-15', 'Aprobado'),
(2, 2, '2024-01-20', 'Sobresaliente');

- **Certificacións obtidas polos empregados, con nome, datas de emisión e vencemento, entidade emisora e estado.**

INSERT INTO CERTIFICACIONES (ID_Certificacion, ID_Empleado, Nombre, Fecha_Emission, Fecha_Vencimiento, Entidad_Emisora, Estado) VALUES

(1, 1, 'Certificación de Programación PLC', '2023-06-01', '2026-06-01', 'Entidad Certificadora A', 'Vigente'),
(2, 2, 'Certificación en Mantenimiento Industrial', '2022-11-15', '2025-11-15', 'Entidad Certificadora B', 'Vigente');

- **Horarios asignados aos empregados para o servizo técnico, con datas, turno e dispoñibilidade.**

INSERT INTO HORARIOS_SERVICIO_TECNICO (ID_Horario, ID_Empleado, Fecha, Turno, Disponibilidad) VALUES

(1, 1, '2024-05-01', 'Mañana', 'Disponible'),
(2, 2, '2024-05-01', 'Tarde', 'No Disponible');

INSERT INTO COMPONENTES_MAUQUINA (ID_Componente, ID_Maquina, Cantidad, Fecha_Instalacion) VALUES

(1, 1, 4, '2023-01-15'), -- *Componente A no Robot Industrial Model X*
(2, 2, 6, '2022-11-10'), -- *Componente B na CNC Model Y*
(3, 3, 2, '2023-05-22'), -- *Componente C na Impresora 3D Delta Pro*
(4, 4, 3, '2021-09-30'), -- *Componente D no PLC Siemens S7-1200*
(5, 5, 1, '2022-02-14'), -- *Componente E no Brazo Robótico UR5e*
(6, 6, 1, '2023-08-01'), -- *Componente F no Sistema SCADA WinCC Advanced*
(7, 7, 2, '2020-12-20'), -- *Componente G na Fresadora CNC Model Z*
(8, 8, 5, '2024-01-10'), -- *Componente H no Robot Colaborativo Cobot A1*
(9, 9, 2, '2022-06-18'), -- *Componente I no Controlador PID Omron E5CC*
(10, 10, 3, '2023-03-03'), -- *Componente J no Sistema de Visión Cognex In-Sight*
(11, 1, 1, '2023-01-15'), -- *Componente K tamén no Robot Industrial Model X*
(12, 2, 1, '2022-11-10'); -- *Componente L tamén na CNC Model Y*

8. Funciones y Procedimientos Almacenados

- FUNCIONES ALMACENADAS

- *Calculo de antigüedad en la empresa desde año de contratación introducido por teclado;*

DELIMITER //

CREATE FUNCTION calcular_antigüedad(p_fecha DATE)

RETURNS INT

DETERMINISTIC

BEGIN




RETURN TIMESTAMPDIFF(YEAR, p_fecha, CURDATE());

END

//

DELIMITER ;

```
4
5  DELIMITER //
6  • CREATE FUNCTION calcular_antiguedad(p_fecha DATE)
7    RETURNS INT
8    DETERMINISTIC
9    BEGIN
10      RETURN TIMESTAMPDIFF(YEAR, p_fecha, CURDATE());
11    END
12  //
13  DELIMITER ;
14
15  • select calcular_antiguedad('2018-03-23');
16
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	calcular_antiguedad('2018-03-23')			
7				

- **Creación de el nombre completo de los empleados con su nombre y su apellido**

```
DELIMITER //
```

```
CREATE FUNCTION nombre_completo(p_nombre VARCHAR(50), p_apellidos VARCHAR(50))
```

```
RETURNS VARCHAR(101)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    RETURN CONCAT(p_nombre, ' ', p_apellidos);
```

```
END
```

```
//
```

```
DELIMITER ;
```

```

19  DELIMITER //
20  • CREATE FUNCTION nombre_completo(p_nombre VARCHAR(50), p_apellidos VARCHAR(50))
21  RETURNS VARCHAR(101)
22  DETERMINISTIC
23  BEGIN
24      RETURN CONCAT(p_nombre, ' ', p_apellidos);
25  END
26  //
27  DELIMITER ;
28
29  • SELECT nombre_completo('María', 'Pérez García') AS NombreCompleto;
30
31  -- PROCEDIMIENTOS ALMACENADOS
32
33  -- Procedimiento para insercción de nuevos empleados
34

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
NombreCompleto			
María Pérez García			

- ***Función que devuelve un resumen da información dun empregado.***

DELIMITER //

CREATE FUNCTION resumen_empleado(p_ID INT)

RETURNS TEXT

DETERMINISTIC

BEGIN

DECLARE v_nombre VARCHAR(50);

DECLARE v_apellidos VARCHAR(50);

DECLARE v_email VARCHAR(100);

DECLARE v_fecha DATE;

DECLARE v_antigüedad INT;

DECLARE v_resultado TEXT;

SELECT Nombre, Apellidos, Email, Fecha_Contratacion

INTO v_nombre, v_apellidos, v_email, v_fecha

FROM EMPLEADOS

WHERE ID_Empleado = p_ID;

SET v_antigüedad = TIMESTAMPDIF(YEAR, v_fecha, CURDATE());

SET v_resultado = CONCAT('Empleado: ', v_nombre, ' ', v_apellidos,

', Email: ', v_email,

', Antigüedad: ', v_antigüedad, ' anos');

RETURN v_resultado;

END //

DELIMITER ;

```

62    DELIMITER ;
63
64 •   SELECT resumen_empleado(1);
65
66    -- PROCEDIMIENTOS ALMACENADOS
67
68    -- Procedimiento para insercción de nuevos empleados

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
resumen_empleado(1)			
▶	Empleado: Empleado1 Apellido1, Email: ana.cambio2@example.com, Antigüidade: 6 anos		

- **PROCEDIMIENTOS ALMACENADOS**
 - *Procedimiento para insercción de nuevos empleados*

DELIMITER //

CREATE PROCEDURE insertar_empleado(

IN p_ID_Empleado INT,

IN p_Nombre VARCHAR(50),

IN p_Apellidos VARCHAR(50),

IN p_DNI VARCHAR(15),

IN p_Especialidad VARCHAR(50),

IN p_Telefono VARCHAR(20),

IN p_Email VARCHAR(100),

IN p_Fecha_Contratacion DATE

)

BEGIN

INSERT INTO EMPLEADOS (ID_Empleado, Nombre, Apellidos, DNI, Especialidad, Telefono, Email, Fecha_Contratacion)

VALUES (p_ID_Empleado, p_Nombre, p_Apellidos, p_DNI, p_Especialidad, p_Telefono, p_Email, p_Fecha_Contratacion);

END

//

DELIMITER ;

```
CALL insertar_empleado(  
  9, 'Ana', 'López García', '12345678A', 'Electricista',  
  '600111222', 'ana.lopez@example.com', '2020-01-15'  
);
```

✓	84	14:01:47	CALL insertar_empleado(9, 'Ana', 'López Garc...	1 row(s) affected	0.063 sec
---	----	----------	--	-------------------	-----------

- **Procedimiento para actualizar el contacto de un empleado**

```
DELIMITER //
```

```
CREATE PROCEDURE actualizar_contacto(  
  IN p_ID_Empleado INT,  
  IN p_Telefono VARCHAR(20),  
  IN p_Email VARCHAR(100)  
)  
BEGIN  
  UPDATE EMPLEADOS  
  SET Telefono = p_Telefono,  
      Email = p_Email  
  WHERE ID_Empleado = p_ID_Empleado;  
END  
//  
DELIMITER ;
```

```

DELIMITER //
• CREATE PROCEDURE actualizar_contacto(
    IN p_ID_Empleado INT,
    IN p_Telefono VARCHAR(20),
    IN p_Email VARCHAR(100)
)
• BEGIN
    UPDATE EMPLEADOS
    SET Telefono = p_Telefono,
        Email = p_Email
    WHERE ID_Empleado = p_ID_Empleado;
END
//
DELIMITER ;

• CALL actualizar_contacto(1, '666999000', 'ana.actualizado@example.com');

```

```

70     WHERE ID_Empleado = p_ID_Empleado;
71 END
72 //
73 DELIMITER ;
74
75 • CALL actualizar_contacto(1, '666999000', 'ana.actualizado@example.com');
76
77 • SELECT * FROM EMPLEADOS WHERE ID_Empleado = 1;
78
79
80 -- TRIGGERS

```

ID_Empleado	Nombre	Apellidos	DNI	Especialidad	Telefono	Email	Fecha_Contratacion
1	Empleado1	Apellido1	12345678X	Electrónica	666999000	ana.actualizado@example.com	2019-03-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Procedemento almacenado que elimina empregados con máis de X anos de antigüidade (con backup previo).**

```
CREATE TABLE IF NOT EXISTS EMPLEADOS_ELIMINADOS AS
```

```
SELECT * FROM EMPLEADOS WHERE 1=0;
```

```
DELIMITER //
```

```
CREATE PROCEDURE eliminar_empleados_antigos(IN p_anhos INT)
```

```
BEGIN
```

```
DECLARE v_data_limite DATE;
```

```
SET v_data_limite = DATE_SUB(CURDATE(), INTERVAL p_anhos YEAR);
```

```
INSERT INTO EMPLEADOS_ELIMINADOS
```

```
SELECT * FROM EMPLEADOS WHERE Fecha_Contratacion < v_data_limite;
```

```
DELETE FROM EMPLEADOS WHERE Fecha_Contratacion < v_data_limite;
```

```
END //
```

```
DELIMITER ;
```

136
137 • CALL eliminar_empleados_antigos(5);
138
139 -- TRIGGERS
140
141 -- Trigger BEFORE INSERT: Validar que el DNI no esté vacío ni nulo; si lo está, cancelar la
142
143 DELIMITER //

Output

#	Time	Action	Message	Duration / Fetch
10	13:03:08	CREATE PROCEDURE eliminar_empleados_a...	0 row(s) affected	0.031 sec
11	13:03:42	CALL eliminar_empleados_antigos(5)	3 row(s) affected	0.078 sec

	ID_Empleado	Nombre	Apellidos	DNI	Especialidad	Telefono	Email	Fecha_Contra
▶	3	Empleado3	Apellido3	34567890Z	Programación	+34 600333444	empleado3@empresa.com	2020-11-01
	4	Empleado4	Apellido4	45678901W	Robótica	+34 600444555	empleado4@empresa.com	2021-02-20
	5	Empleado5	Apellido5	56789012V	Electrónica	+34 600555666	empleado5@empresa.com	2022-09-05
	6	Empleado6	Apellido6	56789012M	Gestión	+34 600545666	empleado6@empresa.com	2027-06-02
	7	Empleado7	Apellido7	67890123N	Automatización	+34 600666777	empleado7@empresa.com	2023-05-10
	8	Empleado8	Apellido8	78901234P	Electrónica	+34 600777888	empleado8@empresa.com	2020-10-01
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

9. Triggers

-- TRIGGERS

-- Trigger BEFORE INSERT: Validar que el DNI no esté vacío ni nulo; si lo está, cancelar la inserción.

DELIMITER //

CREATE TRIGGER validar_DNI_before_insert

BEFORE INSERT ON EMPLEADOS

FOR EACH ROW

BEGIN

IF NEW.DNI IS NULL OR NEW.DNI = '' THEN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'DNI no puede estar vacío';

END IF;

END

//

DELIMITER ;

```

82  -- Trigger BEFORE INSERT: Validar que el DNI no esté vacío ni nulo; si lo está, cancelar la inserción.
83
84  DELIMITER //
85  • CREATE TRIGGER validar_DNI_before_insert
86  BEFORE INSERT ON EMPLEADOS
87  FOR EACH ROW
88  BEGIN
89      IF NEW.DNI IS NULL OR NEW.DNI = '' THEN
90          SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'DNI no puede estar vacío';
91      END IF;
92  END
93  //
94  DELIMITER ;
95
96  • INSERT INTO EMPLEADOS (ID_Empleado, Nombre, Apellidos, DNI, Especialidad, Telefono, Email, Fecha_Contratacion)
97  VALUES (2, 'Juan', 'Rodríguez', '', 'Programador', '611223344', 'juan@example.com', '2021-06-01');
98
99
100

```

Output

#	Time	Action	Message	Duration / Fets
✓ 1	12:45:04	USE AUTOMATIZACION INDUSTRIAL	0 row(s) affected	0.000 sec
✓ 2	12:45:12	CALL actualizar_contacto(1, '666999000', 'ana.actualizado@example....	1 row(s) affected	0.062 sec
✓ 3	12:45:44	SELECT * FROM EMPLEADOS WHERE ID_Empleado = 1 LIMIT 0, ...	1 row(s) returned	0.000 sec / 0.
✗ 4	12:47:03	INSERT INTO EMPLEADOS (ID_Empleado, Nombre, Apellidos, DNI, ...	Error Code: 1644. DNI no puede estar vacío	0.016 sec

-- Trigger AFTER UPDATE: Registrar cambios en teléfono o email en una tabla de auditoría

```

CREATE TABLE auditoria_contacto (
    id INT AUTO_INCREMENT PRIMARY KEY,
    ID_Empleado INT,
    Telefono_antiguo VARCHAR(20),
    Telefono_nuevo VARCHAR(20),
    Email_antiguo VARCHAR(100),
    Email_nuevo VARCHAR(100),
    fecha_cambio TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

DELIMITER //

CREATE TRIGGER auditar_cambios_contacto_after_update

AFTER UPDATE ON EMPLEADOS

FOR EACH ROW

BEGIN

IF OLD.Telefono <> NEW.Telefono OR OLD.Email <> NEW.Email THEN

INSERT INTO auditoria_contacto (ID_Empleado, Telefono_antiguo, Telefono_nuevo, Email_antiguo, Email_nuevo)

VALUES (NEW.ID_Empleado, OLD.Telefono, NEW.Telefono, OLD.Email, NEW.Email);

END IF;

END

//

DELIMITER ;

```
11
12 DELIMITER //
13 • CREATE TRIGGER auditar_cambios_contacto_after_update
14 AFTER UPDATE ON EMPLEADOS
15 FOR EACH ROW
16 BEGIN
17     IF OLD.Telefono <> NEW.Telefono OR OLD.Email <> NEW.Email THEN
18         INSERT INTO auditoria_contacto (ID_Empleado, Telefono_antiguo, Telefono_nuevo, Email
19         VALUES (NEW.ID_Empleado, OLD.Telefono, NEW.Telefono, OLD.Email, NEW.Email);
20     END IF;
21 END
22 //
23 DELIMITER ;
24
25 • CALL actualizar_contacto(1, '699888777', 'ana.cambio2@example.com');
```

✓ 5 12:48:38 CALL actualizar_contacto(1, '699888777', 'ana.ca... 1 row(s) affected 0.016 sec

```
126
127 • SELECT * FROM auditoria_contacto;
128
129
```

Result Grid							
Filter Rows:							
	id	ID_Empleado	Telefono_antiguo	Telefono_nuevo	Email_antiguo	Email_nuevo	fecha_cambio
▶	1	1	+34 600111222	666999000	empleado1@empresa.com	ana.actualizado@example.com	2025-05-28 10:45:11
	2	1	666999000	699888777	ana.actualizado@example.com	ana.cambio2@example.com	2025-05-28 10:48:36
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Un trigger que evita eliminar empregados da especialidade “Director”.**

DELIMITER //

CREATE TRIGGER evitar_eliminar_director

BEFORE DELETE ON EMPLEADOS

FOR EACH ROW

BEGIN

IF OLD.Especialidad = 'Director' THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'Non se pode eliminar un Director.';

END IF;

END //

DELIMITER ;

```
204
205 -- INSERT INTO EMPLEADOS VALUES (10, 'Laura', 'Santos', '99887766A', 'Director',
206 --                                '600000001', 'laura@example.com', '2010-03-01');
207
208 -- Isto lanzará erro:
209 • DELETE FROM EMPLEADOS WHERE ID_Empleado = 10;
210
211
```

Output					
Action Output					
#	Time	Action	Message	Duration / F	
✓ 15	13:18:37	INSERT INTO EMPLEADOS VALUES (10, 'Laura', 'S...	1 row(s) affected	0.016 sec	
✗ 16	13:19:55	DELETE FROM EMPLEADOS WHERE ID_Empleado...	Error Code: 1644. Non se pode eliminar un Director.	0.000 sec	

- Como debería ser ao intentar eliminar o empleado coa categoria “Director” lanzase unha mensaxe que nos di que non é posible a acción e non se executa

10. Consultas SQL

Listar os clientes o nome da sua empresa e o numero total de proxectos de cada cliente.

```

1  -- 1. Clientes con número de proxectos
2
3  • SELECT c.ID_Cliente, c.Nombre_Empresa, COUNT(p.ID_Proyecto) AS Total_Proyectos
4  FROM CLIENTES c
5  JOIN PROYECTOS p ON c.ID_Cliente = p.ID_Cliente
6  GROUP BY c.ID_Cliente, c.Nombre_Empresa;
7

```

ID_Cliente	Nombre_Empresa	Total_Proyectos
1	Empresa 1	1
2	Empresa 2	1
3	Empresa 3	1
4	Empresa 4	1

Listar os empregados con mais de 3 proxectos asignados (en este caso nadie ten mais de 3 proxectos asignando).

```

7
8  -- 2. Empleados con máis de 3 proxectos asignados
9
10 • SELECT e.ID_Empleado, e.Nombre, e.Apellidos, COUNT(ep.ID_Proyecto) AS Proyectos_Asignados
11 FROM EMPLEADOS e
12 JOIN EMPLEADOS_PROYECTO ep ON e.ID_Empleado = ep.ID_Empleado
13 GROUP BY e.ID_Empleado, e.Nombre, e.Apellidos
14 HAVING COUNT(ep.ID_Proyecto) > 3;
15

```

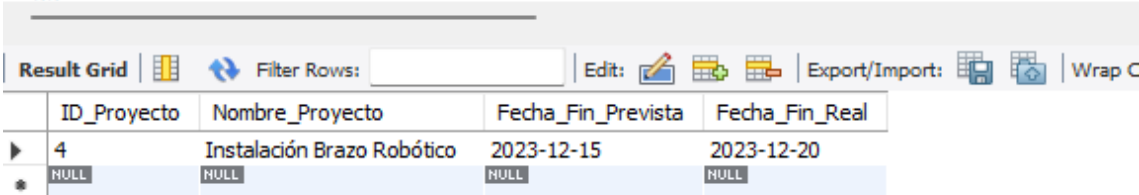
ID_Empleado	Nombre	Apellidos	Proyectos_Asignados
-------------	--------	-----------	---------------------

Listar os proxectos que por retrasos ou outros casos acabaron despois da fecha prevista.

```

16  -- 3. Proxectos que acabaron despois da data prevista
17
18  • SELECT ID_Proyecto, Nombre_Proyecto, Fecha_Fin_Prevista, Fecha_Fin_Real
19  FROM PROYECTOS
20  WHERE Fecha_Fin_Real > Fecha_Fin_Prevista;
21

```



The screenshot shows a database query result grid. The toolbar includes 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The grid has five columns: ID_Proyecto, Nombre_Proyecto, Fecha_Fin_Prevista, and Fecha_Fin_Real. The first row shows project 4, 'Instalación Brazo Robótico', with a planned finish date of 2023-12-15 and an actual finish date of 2023-12-20. A second row with all NULL values is also visible.

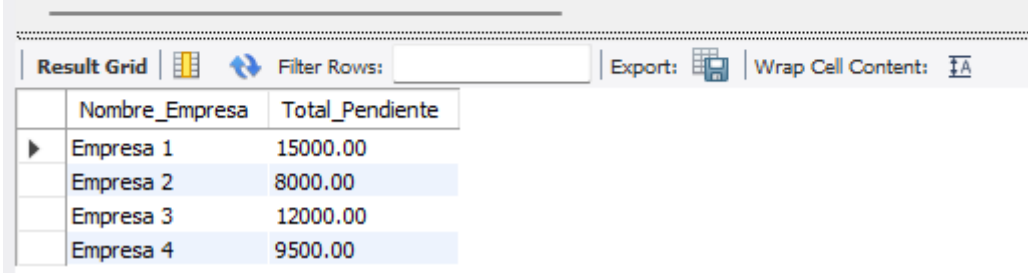
ID_Proyecto	Nombre_Proyecto	Fecha_Fin_Prevista	Fecha_Fin_Real
4	Instalación Brazo Robótico	2023-12-15	2023-12-20
NULL	NULL	NULL	NULL

Listar as empresas que teñen pendente o pago.

```

22  -- 4. Facturas pendientes de pago por cliente
23
24  • SELECT c.Nombre_Empresa, SUM(f.Monto_Total) AS Total_Pendiente
25  FROM FACTURAS f
26  JOIN PROYECTOS p ON f.ID_Proyecto = p.ID_Proyecto
27  JOIN CLIENTES c ON p.ID_Cliente = c.ID_Cliente
28  WHERE f.Estado_Pago <> 'Pagado'
29  GROUP BY c.Nombre_Empresa;

```



The screenshot shows a database query result grid. The toolbar includes 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The grid has two columns: Nombre_Empresa and Total_Pendiente. It lists four companies with their respective pending payment totals.

Nombre_Empresa	Total_Pendiente
Empresa 1	15000.00
Empresa 2	8000.00
Empresa 3	12000.00
Empresa 4	9500.00

Polo que parece a empresa non ten presa polos cobros.

Listar as empresas que teñen maquinas con mais de unha incidencia:

```
31 -- 5. Subconsulta correlacionada: Clientes que teñen máquinas con máis dunha incidencia rexistrada
32
33 • SELECT c.Nombre_Empresa
34 FROM CLIENTES c
35 WHERE EXISTS (
36     SELECT 1
37     FROM MAQUINAS m
38     JOIN INCIDENCIAS i ON i.ID_Maquina = m.ID_Maquina
39     WHERE m.ID_Cliente = c.ID_Cliente
40     GROUP BY m.ID_Maquina
41     HAVING COUNT(i.ID_Incidencia) > 1
42 );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Nombre_Empresa

Listar os provedores en funcion de pedidos.

```
44 -- 6. Mellhores provedores segundo total de pedidos
45
46 • SELECT pr.Nombre_Empresa, COUNT(p.ID_Pedido) AS Total_Pedidos, SUM(p.Monto_Total) AS Total_Comprado
47 FROM PROVEEDORES pr
48 JOIN PEDIDOS p ON pr.ID_Proveedor = p.ID_Proveedor
49 GROUP BY pr.Nombre_Empresa
50 ORDER BY Total_Comprado DESC;
51
52
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Nombre_Empresa	Total_Pedidos	Total_Comprado
►	Proveedor 1	1	1275.00
	Proveedor 4	1	890.00
	Proveedor 2	1	650.00
	Proveedor 3	1	330.00

Horas totales asignadas por proyecto

```
53
54 -- 7. Horas totais asignadas por proxecto
55
56 • SELECT p.Nombre_Proyecto, SUM(ep.Horas_Asignadas) AS Total_Horas
57 FROM PROYECTOS p
58 JOIN EMPLEADOS_PROYECTO ep ON p.ID_Proyecto = ep.ID_Proyecto
59 GROUP BY p.Nombre_Proyecto;
60
61 -- 8. Componentes con stock por debaixo do mínimo
62
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	Nombre_Proyecto	Total_Horas
▶	Actualización Máquina CNC	90
	Integración SCADA	100
	Instalación Brazo Robótico	80

Componentes con stock por debaixo do mínimo

```
60
61 -- 8. Componentes con stock por debaixo do mínimo
62
63 • SELECT ID_Componente, Nombre, Stock, Stock_Minimo
64 FROM COMPONENTES
65 WHERE Stock < Stock_Minimo;
66
67
68
```

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

	ID_Componente	Nombre	Stock	Stock_Minimo
*	NULL	NULL	NULL	NULL

Non hai compoñentes con stock por debaixo do minimo

Proxectos cun custo en facturas superior ao seu presuposto (Non hai)

```
67 -- 9. Proxectos cun custo en facturas superior ao seu presuposto
68
69 • SELECT p.ID_Proyecto, p.Nombre_Proyecto, p.Presupuesto, SUM(f.Monto_Total) AS Total_Facturado
70 FROM PROYECTOS p
71 JOIN FACTURAS f ON p.ID_Proyecto = f.ID_Proyecto
72 GROUP BY p.ID_Proyecto, p.Nombre_Proyecto, p.Presupuesto
73 HAVING SUM(f.Monto_Total) > p.Presupuesto;
74
75
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ID_Proyecto	Nombre_Proyecto	Presupuesto	Total_Facturado

Número de mantementos por tipo de máquina

```
74
75 -- 10. Número de mantementos por tipo de máquina
76
77 • SELECT m.Tipo_Maquina, COUNT(mt.ID_Mantenimiento) AS Total_Mantenimientos
78 FROM MAQUINAS m
79 JOIN MANTENIMIENTOS mt ON m.ID_Maquina = mt.ID_Maquina
80 GROUP BY m.Tipo_Maquina;
81
82
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Tipo_Maquina	Total_Mantenimientos		
Robot Industrial	1		
CNC	1		
Brazo Robótico	1		
Sistema SCADA	1		

Clientes sen proxectos rexistrados

```
80 GROUP BY m.Tipo_Maquina,  
81  
82 -- 11. Clientes sen proxectos rexistrados  
83  
84 • SELECT c.ID_Cliente, c.Nombre_Empresa  
85 FROM CLIENTES c  
86 LEFT JOIN PROYECTOS p ON c.ID_Cliente = p.ID_Cliente  
87 WHERE p.ID_Proyecto IS NULL;  
88
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ID_Cliente	Nombre_Empresa		
5	Empresa 5		
6	Empresa 6		
7	Empresa 7		
8	Empresa 8		

Empleados sen formación completada

```
89 -- 12. Empleados sen formación completada  
90  
91 • SELECT e.ID_Empleado, e.Nombre, e.Apellidos  
92 FROM EMPLEADOS e  
93 LEFT JOIN EMPLEADOS_FORMACION ef ON e.ID_Empleado = ef.ID_Empleado  
94 WHERE ef.ID_Formacion IS NULL;  
95  
96
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ID_Empleado	Nombre	Apellidos	
1	Empleado1	Apellido1	
2	Empleado2	Apellido2	
3	Empleado3	Apellido3	
4	Empleado4	Apellido4	
5	Empleado5	Apellido5	
6	Empleado6	Apellido6	
7	Empleado7	Apellido7	
8	Empleado8	Apellido8	
10	Laura	Santos	

Facturas vencidas sen pagar

```
95
96  -- 13. Facturas vencidas sen pagar
97
98  • SELECT f.ID_Factura, f.Fecha_Vencimiento, f.Monto_Total, p.Nombre_Proyecto
99  FROM FACTURAS f
100  JOIN PROYECTOS p ON f.ID_Proyecto = p.ID_Proyecto
101  WHERE f.Estado_Pago <> 'Pagado'
102  AND f.Fecha_Vencimiento < CURDATE();
103
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ID_Factura	Fecha_Vencimiento	Monto_Total	Nombre_Proyecto
1	2024-04-01	15000.00	Automatización Línea 1
2	2024-02-15	8000.00	Actualización Máquina CNC
3	2024-06-10	12000.00	Integración SCADA
4	2024-01-21	9500.00	Instalación Brazo Robótico



Proxectos sen documentación técnica asignada

```
103
104  -- 14. Proxectos sen documentación técnica asignada
105
106  • SELECT p.ID_Proyecto, p.Nombre_Proyecto
107  FROM PROYECTOS p
108  LEFT JOIN DOCUMENTACION_TECNICA d ON p.ID_Proyecto = d.ID_Proyecto
109  WHERE d.ID_Documento IS NULL;
110
111
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ID_Proyecto	Nombre_Proyecto		

Máquinas con garantía activa hoxe

```
110
111 -- 15. Máquinas con garantía activa hoxe
112
113 • SELECT m.ID_Maquina, m.Modelo, g.Fecha_Inicio, g.Fecha_Fin
114 FROM MAQUINAS m
115 JOIN GARANTIAS g ON m.ID_Maquina = g.ID_Maquina
116 WHERE CURDATE() BETWEEN g.Fecha_Inicio AND g.Fecha_Fin;
117
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: 				
	ID_Maquina	Modelo	Fecha_Inicio	Fecha_Fin
▶	3	Delta Pro	2024-01-10	2026-01-10
	5	UR5e	2023-07-01	2025-07-01

11. Casos de Prueba y Simulación

Nº	PROBA	ACCIÓN	ESPERADO
1	Alta válida	Inserir un novo compoñente cun provedor existente	Inserción exitosa
2	Alta válida	Crear un pedido e engadir liñas de compoñentes	Pedido e liñas creadas correctamente
3	Alta válida	Rexistrar unha máquina e asociarlle compoñentes	Asociación correcta
4	Alta válida	Crear un mantemento asignado a un técnico	Mantemento creado con FK válida
5	Alta válida	Rexistrar unha formación e asociar a un empregado	Relación formación-empregado rexistrada correctamente
6	Alta válida	Emitir unha certificación para un empregado	Certificación rexistrada correctamente
7	Alta válida	Rexistrar unha incidencia cun cliente e máquina existentes	Incidencia rexistrada correctamente
8	Erro por FK	Asociar un compoñente a unha máquina inexistente	Erro por FK (restrición de integridade referencial)
9	Erro por FK	Crear unha incidencia cun cliente inexistente	Erro por FK (cliente non existente)
10	Erro por PK duplicada	Inserir dúas veces o mesmo rexistro en EMPLEADOS_FORMACION	Erro por clave primaria composta duplicada
11	Borrado con SET NULL	Eliminar un empregado con mantementos asignados	Campo ID_Empleado_Responsable en MANTENIMIENTOS pasa a NULL
12	Borrado con CASCADE	Eliminar unha máquina asociada a compoñentes e mantementos	Eliminación en cascada de asociacións en COMPONENTES_MAQUINA e outros

-- Proveedor

```
INSERT INTO PROVEEDORES VALUES (7, 'Provedora Técnica S.A.', 'A11111111', 'Rúa dos  
Provedores, 45', '981123456', 'contacto@provetec.com', 'Manuel Castro');
```

```
3 -- Proveedor
4 • INSERT INTO PROVEEDORES VALUES (7, 'Provedora Técnica S.A.', 'A11111111', 'Rúa dos Provedores, 45', '981123456', 'contacto@provetec.com', 'Manuel Castro');
```

Output			
Action Output			
#	Time	Action	Message
1	11:13:37	INSERT INTO PROVEEDORES VALUES (7, 'Provedora Técnica S.A.', 'A11111111', 'Rúa dos Prov...	1 row(s) affected

Duration / Fetch: 0.016 sec

Como se mostra na tabla a inserción e permitida pola BD.

-- Compoñente

```
INSERT INTO COMPONENTES VALUES (13, 'Sensor de temperatura', 'Sensor analóxico 4-  
20mA', 100, 15.00, 20, 200, 1);
```

```
6 -- Compoñente
7 • INSERT INTO COMPONENTES VALUES (13, 'Sensor de temperatura', 'Sensor analóxico 4-20mA', 100, 15.00, 20, 200, 1);
8
```

Output			
Action Output			
#	Time	Action	Message
1	11:17:36	INSERT INTO COMPONENTES VALUES (13, 'Sensor de temperatura', 'Sensor analóxico 4-20mA', 1...	1 row(s) affected

Como se mostra na tabla a inserción e permitida pola BD.

-- Pedido

```
INSERT INTO PEDIDOS VALUES (5, 1, '2025-05-10', 'Solicitado', '2025-05-20', NULL, 1500.00,  
'F123-987');
```

```
9 -- Pedido
10 • INSERT INTO PEDIDOS VALUES (5, 1, '2025-05-10', 'Solicitado', '2025-05-20', NULL, 1500.00, 'F123-987');
11
```

Output			
Action Output			
#	Time	Action	Message
1	11:19:38	INSERT INTO PEDIDOS VALUES (5, 1, '2025-05-10', 'Solicitado', '2025-05-20', NULL, 1500.00, 'F12...	1 row(s) affected

Como se mostra na tabla a inserción e permitida pola BD.

-- Pedido de compoñente

```
INSERT INTO PEDIDOS_COMPONENTES VALUES (5, 10, 100, 15.00, 'Pendiente');
```

```
11
12 -- Pedido de compoñente
13 • INSERT INTO PEDIDOS_COMPONENTES VALUES (5, 10, 100, 15.00, 'Pendiente');
14
```

Output

#	Time	Action	Message
✓ 1	11:20:40	INSERT INTO PEDIDOS_COMPONENTES VALUES (5, 10, 100, 15.00, 'Pendiente')	1 row(s) affected

Como se mostra na tabla a inserción e permitida pola BD.

-- Máquina

```
INSERT INTO MAQUINAS VALUES (11, 6, 'PLC', 'Siemens S7-1500', '2025-04-01', 'Operativa');
```

```
15 -- Máquina
16 • INSERT INTO MAQUINAS VALUES (11, 6, 'PLC', 'Siemens S7-1500', '2025-04-01', 'Operativa');
17
18 -- Compoñente instalado na máquina
19
```

Output

#	Time	Action	Message
✓ 1	11:22:04	INSERT INTO MAQUINAS VALUES (11, 6, 'PLC', 'Siemens S7-1500', '2025-04-01', 'Operativa')	1 row(s) affected

Como se mostra na tabla a inserción e permitida pola BD.

-- Compoñente instalado na máquina

```
INSERT INTO COMPONENTES_MAQUINA VALUES (13, 11, 2, '2025-04-02');
```

```

18      -- Compoñente instalado na máquina
19 •    INSERT INTO COMPONENTES_MAQUINA VALUES (13, 11, 2, '2025-04-02');
20

```

Output				
Action Output				
#	Time	Action	Message	
✓	1 11:25:51	SELECT * FROM AUTOMATIZACION_INDUSTRIAL.COMPONENTES_MAQUINA LIMIT 0, 1000	12 row(s) returned	
✓	2 11:27:49	INSERT INTO COMPONENTES_MAQUINA VALUES (13, 11, 2, '2025-04-02')	1 row(s) affected	

Como se mostra na tabla a inserción e permitida pola BD.

-- Máquina 999 non existe

```
INSERT INTO COMPONENTES_MAQUINA VALUES (14, 999, 2, '2025-04-02');
```

-- Resultado: ERRO por FK

```

21      -- Máquina 999 non existe
22 •    INSERT INTO COMPONENTES_MAQUINA VALUES (14, 999, 2, '2025-04-02');
23      -- Resultado: ERRO por FK

```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✗	1 11:28:56	INSERT INTO COMPONENTES_MAQUINA VALUES (14, 999, 2, '2025-04-02')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (AUTOMATIZACION...	0.000 sec

Como o id da maquina non corresponde a ningunha maquina non e posible a inserción.

--Mantenimiento a la maquina añadida

```
INSERT INTO MANTENIMIENTOS VALUES (5, 11, 300, '2025-05-15', 'Preventivo', 'Revisión anual', 'Completado');
```

```

24
25 •    INSERT INTO MANTENIMIENTOS VALUES (5, 11, 5, '2025-05-15', 'Preventivo', 'Revisión anual', 'Completado');
26

```

Output				
Action Output				
#	Time	Action	Message	
✓	1 11:34:56	INSERT INTO MANTENIMIENTOS VALUES (5, 11, 5, '2025-05-15', 'Preventivo', 'Revisión anual', 'C...	1 row(s) affected	

Como se mostra na tabla a inserción e permitida pola BD a maquina introducida anteriormente.

-- Curso

```
INSERT INTO FORMACIONES VALUES (3, 'Curso PLC avanzado', '2025-01-15', '2025-02-15', 'Xosé López', 'Presencial');
```

4 11:36:36 INSERT INTO FORMACIONES VALUES (3, 'Curso PLC avanzado', '2025-01-15', '2025-02-15', 'Xosé López', 'Presencial'); 1 row(s) affected 0.015 sec

Insercción de novo curso permitido

-- Asistencia do empregado

```
INSERT INTO EMPLEADOS_FORMACION VALUES (1, 1, '2025-02-16', 'Aprobado');
```

```
INSERT INTO CERTIFICACIONES VALUES (1, 1, 'Certificación Siemens TIA', '2025-03-01', '2028-03-01', 'Siemens', 'Vixente');
```

```
30 -- Asistencia do empregado
31 • INSERT INTO EMPLEADOS_FORMACION VALUES (1, 1, '2025-02-16', 'Aprobado');
32
33 • INSERT INTO CERTIFICACIONES VALUES (1, 1, 'Certificación Siemens TIA', '2025-03-01', '2028-03-01', 'Siemens', 'Vixente');
34
```

Output

#	Time	Action	Message
1	11:42:24	INSERT INTO EMPLEADOS_FORMACION VALUES (1, 1, '2025-02-16', 'Aprobado')	1 row(s) affected
2	11:42:33	INSERT INTO CERTIFICACIONES VALUES (1, 1, 'Certificación Siemens TIA', '2025-03-01', '2028-03-01', 'Siemens', 'Vixente')	1 row(s) affected

Inserccións permitidas pola base de datos.

-- Repetimos a mesma entrada en EMPLEADOS_FORMACION

```
INSERT INTO EMPLEADOS_FORMACION VALUES (1, 1, '2025-02-16', 'Aprobado');
```

```
35 -- Repetimos a mesma entrada en EMPLEADOS_FORMACION
36 • INSERT INTO EMPLEADOS_FORMACION VALUES (1, 1, '2025-02-16', 'Aprobado');
37 -- ERRO por PK duplicada (ID_Empleado + ID_Formacion)
38
```

Output

#	Time	Action	Message
1	11:44:08	INSERT INTO EMPLEADOS_FORMACION VALUES (1, 1, '2025-02-16', 'Aprobado')	Error Code: 1062. Duplicate entry '1-1' for key 'EMPLEADOS_FORMACION.PRIMARY'

ERRO por PK duplicada (ID_Empleado + ID_Formacion)

-- Carlos ten mantementos asignados

DELETE FROM EMPLEADOS WHERE ID_Empleado = 1;

-- Esperado: En MANTENIMIENTOS, o campo ID_Empleado_Responsable pasa a NULL

```
--
39 -- Carlos ten mantementos asignados
40 • DELETE FROM EMPLEADOS WHERE ID_Empleado = 2;
41 -- Esperado: En MANTENIMIENTOS, o campo ID_Empleado_Responsable pasa a NULL
42
```

Output :

#	Time	Action	Message
1	11:53:14	DELETE FROM EMPLEADOS WHERE ID_Empleado = 2	1 row(s) affected

ID_Mantenimiento	ID_Maquina	ID_Empleado_Responsable	Fecha_Mantenimiento	Tipo	Descripcion	Estado
1	1	NULL	2024-03-15	Preventivo	Revisión trimestral del robot	Realizado
2	2	NULL	2024-04-01	Correctivo	Reparación de fallo en eje	Pendiente
3	5	5	2024-05-10	Correctivo	Substitución de piezas desgastadas	Realizado
4	6	6	2024-05-18	Preventivo	Revisión mensual sistema SCADA	Pendiente
5	11	5	2025-05-15	Preventivo	Revisión anual	Completado
NULL	NULL	NULL	NULL	NULL	NULL	NULL

O empleado con id 2 tñía asociado 2 mantementos;

-- Inserción de incidencia:

INSERT INTO INCIDENCIAS VALUES (5, 4, 5, '2025-05-20', 'Avería na comunicación Modbus', 'Abierta', 'Alta');

```
--
43 -- Inserción de incidencia:
44 • INSERT INTO INCIDENCIAS VALUES (5, 4, 5, '2025-05-20', 'Avería na comunicación Modbus', 'Abierta', 'Alta');
45
46 -- Cliente 999 non existe
47 • INSERT INTO INCIDENCIAS VALUES (2, 999, 50, '2025-05-21', 'Proba con erro', 'Abierta', 'Media');
48 -- ERRO por FK
```

Output :

#	Time	Action	Message
1	11:55:44	INSERT INTO INCIDENCIAS VALUES (1, 4, 5, '2025-05-20', 'Avería na comunicación Modbus', 'Aber...	Error Code: 1062. Duplicate entry '1' for key 'INCIDENCIAS.PRIMARY'
2	11:55:57	INSERT INTO INCIDENCIAS VALUES (5, 4, 5, '2025-05-20', 'Avería na comunicación Modbus', 'Aber...	1 row(s) affected

Inserccións permitidas pola base de datos.

-- Cliente 999 non existe

```
INSERT INTO INCIDENCIAS VALUES (2, 999, 50, '2025-05-21', 'Proba con erro', 'Aberta', 'Media');
```

```
46 -- Cliente 999 non existe
47 INSERT INTO INCIDENCIAS VALUES (6, 999, 50, '2025-05-21', 'Proba con erro', 'Aberta', 'Media');
48 -- ERRO por FK
```

#	Time	Action	Message	Duration / Fetch
1	11:58:43	INSERT INTO INCIDENCIAS VALUES (6, 999, 50, '2025-05-21', 'Proba con erro', 'Aberta', 'Media')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (AUTOMATIZACIO...	0.000 sec

ERRO por FK

```
DELETE FROM EMPLEADOS WHERE ID_Empleado = 9;
```

-- Resultado: os mantementos que tiñan como responsable o empregado 9 agora teñen ese campo a NULL

```
50
51 DELETE FROM EMPLEADOS WHERE ID_Empleado = 9;
52 -- Resultado: os mantementos que tiñan como responsable o empregado 9 agora teñen ese campo a NL
```

#	Time	Action	Message	Duration / Fetch
1	12:01:12	DELETE FROM EMPLEADOS WHERE ID_Empleado...	0 row(s) affected	0.000 sec

Neste caso parece que casualmente o empleado 9 non tiña mantementos asociados

12. Resultados y Verificación

Unha vez feita toda a base de datos, cos seus datos cargados e relacións definidas, fixéronse varias probas para comprobar que todo funciona como debería. A idea era ver se o sistema responde ben tanto nas operacións normais como cando se lle meten datos erróneos ou se fan accións complexas.

Comprobación da integridade dos datos

- As relacións entre táboas funcionan ben: cando un dato depende doutro (por exemplo, un proxecto dun cliente ou un mantemento dunha máquina), os valores están ben ligados e cumpren as regras de integridade.
- Probouse a eliminación de datos con distintas configuracións:

- Cando se borra un empregado con mantementos asignados, o campo de responsable pasa a NULL, tal e como se esperaba (ON DELETE SET NULL).
- Cando se elimina unha máquina, tamén se borran automaticamente os compoñentes instalados nela e os mantementos rexistrados (ON DELETE CASCADE).

Altas, baixas e modificacións

- Insertáronse clientes, proxectos, máquinas, compoñentes, empregados e relacións entre eles sen erros.
- Tamén se fixeron modificacións, como cambiar teléfonos ou emails de empregados, e comprobouse que se gardaban ben.
- Probouse a eliminación de empregados e máquinas, comprobando que os datos relacionados se actualizaban ou borraban correctamente.

Probas con erros (intencionados)

- Intentouse inserir datos con claves foráneas erróneas (por exemplo, asociar un compoñente a unha máquina que non existe), e o sistema rexeitouno, como era de esperar.
- Tamén se fixo unha proba de duplicar rexistros en táboas con claves primarias compostas (como empregados en formacións), e saltou o erro correspondente.
- Os triggers tamén responderon ben:
 - O que impide inserir empregados sen DNI funcionou correctamente.
 - O que evita borrar directores tamén.
 - O que garda cambios de contacto nunha táboa de auditoría tamén se executou cando se fixo un UPDATE.

Resultados das consultas

As consultas preparadas permitiron extraer datos útiles como:

- Clientes con máis proxectos.
- Empregados que traballan en varios proxectos.
- Incidencias abertas ou prioritarias.
- Proxectos que superan o orzamento previsto.
- Compoñentes con stock baixo.
- Empregados que aínda non fixeron cursos de formación.

Resumo final

A base de datos responde ben, mantén a coherencia dos datos, reacciona ante erros e permite obter información importante para a empresa. O deseño parece axeitado e está preparado para un uso real con datos dinámicos. Pódense facer operacións complexas sen que haxa inconsistencias, e os procedementos, funcións e triggers aportan valor engadido.

13. Conclusiones y Mejoras Futuras

A base de datos deseñada para a empresa de automatización industrial cumpre cos obxectivos iniciais do proxecto. A estrutura creada permite:

- Xestionar de maneira organizada os clientes, proxectos, empregados, máquinas, compoñentes e incidencias.
- Levar un control detallado de mantementos, formacións, certificacións e servizos técnicos, algo fundamental neste tipo de negocio.
- Garantir a integridade dos datos mediante claves foráneas, restricións e accións automáticas (como CASCADE e SET NULL) que aseguran que non haxa información illada ou errónea.
- Mellorar a funcionalidade e seguridade coa implementación de funcións, procedementos almacenados e triggers, que automatizan certas tarefas e prevenen erros comúns.

Durante o desenvolvemento realizáronse probas completas (altas, baixas, modificacións e situacións con erros) e o comportamento do sistema foi o esperado en todos os casos. Ademais, as consultas SQL creadas permiten extraer información útil para a toma de decisións.

Melloras futuras

Aínda que a base de datos está funcional e lista para uso real, hai algunhas melloras que se poderían aplicar nun futuro:

- Interfaz de usuario: Desenvolver unha aplicación web ou de escritorio que se conecte coa base de datos para facilitar o uso a usuarios non técnicos.
- Xestión de permisos: Engadir roles de usuario na base de datos (administrador, técnico, comercial...) con diferentes niveis de acceso.
- Auditoría avanzada: Crear un sistema máis completo para rexistrar accións importantes (insercións, modificacións e eliminacións), con datas e usuarios.
- Validacións máis completas: Engadir máis CHECK constraints ou procedementos para controlar valores lóxicos (ex. stock negativo, datas erróneas, etc.).
- Historial de mantementos e incidencias: Gardar rexistros históricos cando se faga un cambio de estado para manter trazabilidade completa.
- Documentación técnica: Incorporar unha mellor organización e almacenamento dos documentos técnicos asociados a máquinas e proxectos.
- Exportación de datos: Preparar informes en PDF ou Excel para facilitar a comunicación cos clientes ou provedores.

14. Enlace al Repositorio en GitHub

[REPOSITORIO GIT](#)