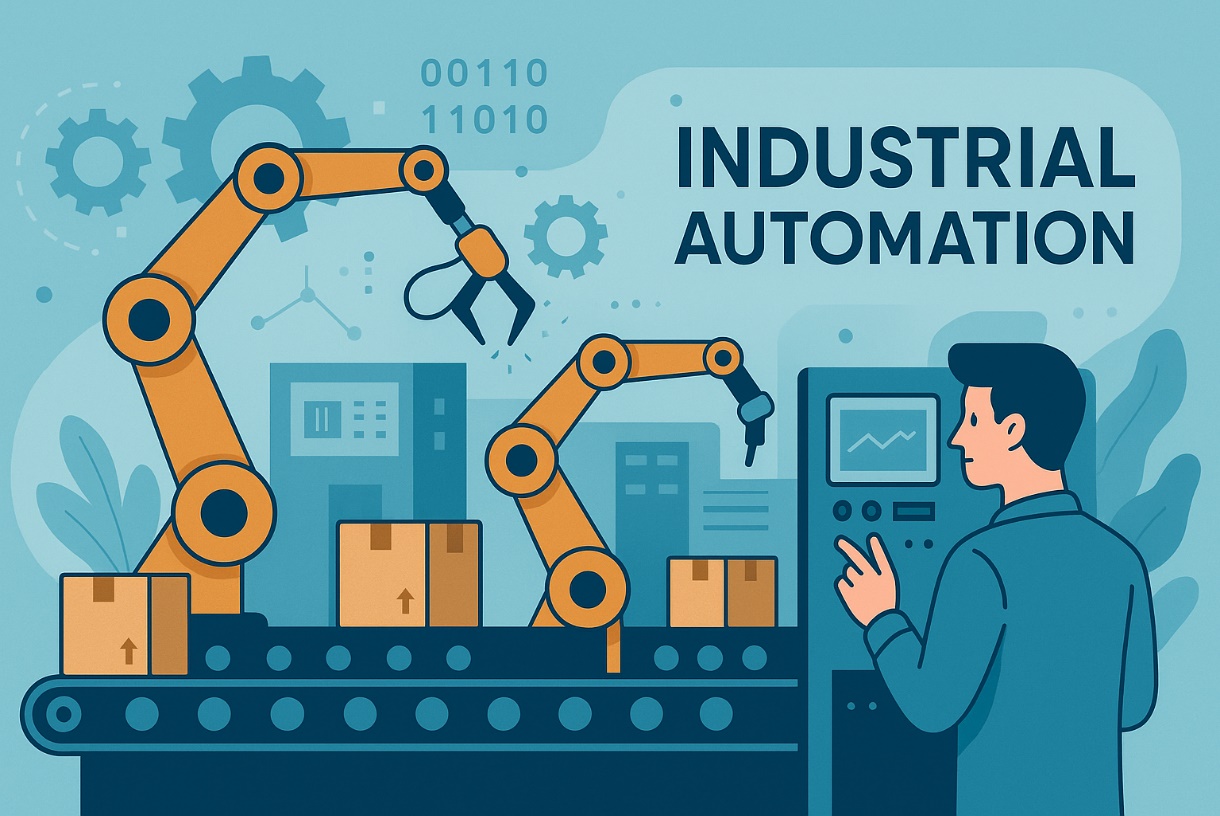
EMPRESA AUTOMATIZACIÓN INDUSTRIAL



***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](#_Toc197794426)

[2. Descripción del Problema / Requisitos 2](#_Toc197794427)

[3. Modelo Conceptual 2](#_Toc197794428)

[4. Modelo Relacional 2](#_Toc197794429)

[5. Proceso de Normalización 2](#_Toc197794430)

[6. Script de Creación de la Base de Datos 2](#_Toc197794431)

[7. Carga de Datos Inicial 2](#_Toc197794432)

[8. Funciones y Procedimientos Almacenados 2](#_Toc197794433)

[9. Triggers 2](#_Toc197794434)

[10. Consultas SQL 2](#_Toc197794435)

[11. Casos de Prueba y Simulación 2](#_Toc197794436)

[12. Resultados y Verificación 2](#_Toc197794437)

[13. Capturas de Pantalla (opcional) 2](#_Toc197794438)

[14. Conclusiones y Mejoras Futuras 2](#_Toc197794439)

[15. Enlace al Repositorio en GitHub 3](#_Toc197794440)

# Introducción

El proyecto consiste en una base de datos para gestionar los registros, datos de clientes y empleados y todo tipo de información de una empresa que se dedica a la automatización industrial , haciendo maquinas automatizadas, automatizando procesos en planta y ofreciendo un mantenimiento de estas o de maquinas ajenas a la empresa.

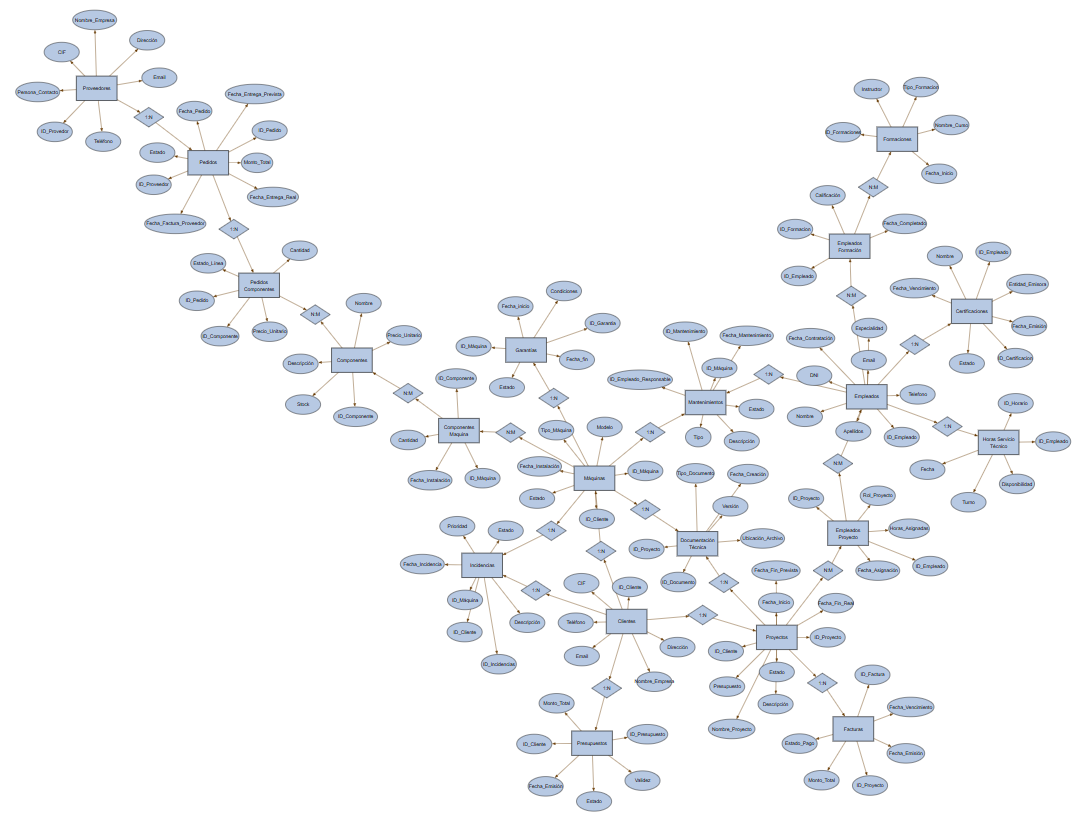
# Descripción del Problema / Requisitos

El principal problema es una mejor gestión de la empresa ya que precisa gestionar gran cantidad de datos.

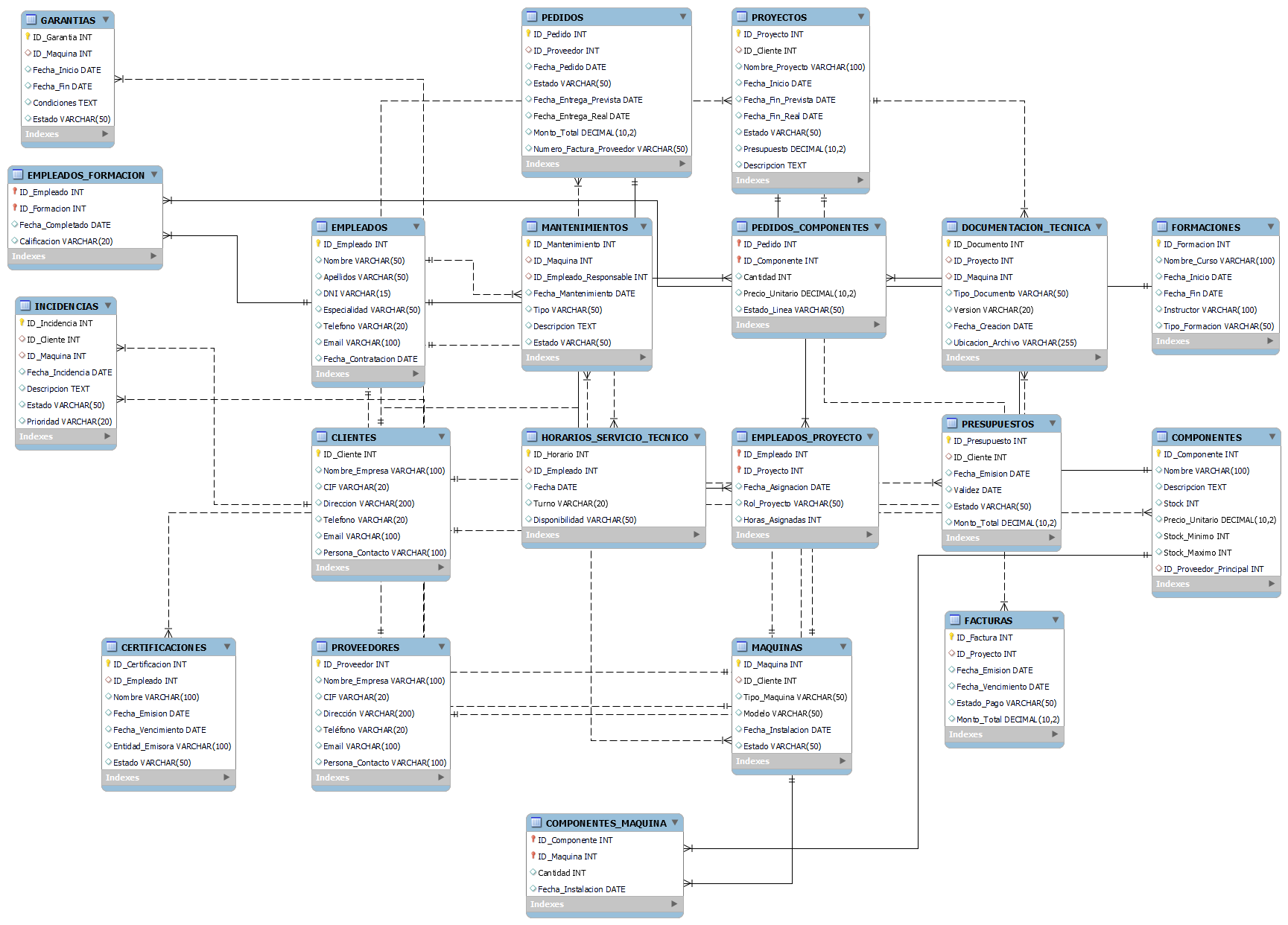
Características clave del negocio:

* Servicios de consultoría y diseño de soluciones.
* Implementación de sistemas de control industrial.
* Programación de PLC’s y robots.
* Mantenimiento preventivo y correctivo.
* Capacitación del personal del cliente.
* Servicio de soporte técnico 24/7.

# Modelo Conceptual



# Modelo Relacional



# Proceso de Normalización

El proceso de normalización es una técnica fundamental en el diseño de bases de datos relacionales que busca minimizar la redundancia de datos y asegurar la integridad de la información. A través de diferentes formas normales (1FN, 2FN, 3FN...), se garantiza que la estructura de la base de datos sea eficiente, coherente y escalable.

En este proyecto se realizó un análisis de las siguientes relaciones: CLIENTES, PROYECTOS, EMPLEADOS, EMPLEADOS\_PROYECTO, MÁQUINAS, COMPONENTES, COMPONENTES\_MÁQUINA, MANTENIMIENTOS, INCIDENCIAS, DOCUMENTACIÓN\_TÉCNICA, PRESUPUESTOS, FACTURAS, PROVEEDORES, PEDIDOS, PEDIDOS\_COMPONENTES, GARANTIAS, FORMACIONES, EMPLEADOS\_FORMACION, CERTIFICACIONES y HORARIOS\_SERVICIO\_TÉCNICO.

#### **1ª Forma Normal (1FN)**

Todas las tablas cumplen con la 1FN al garantizar que cada atributo contiene un valor atómico e indivisible, sin repeticiones de grupos de datos ni listas. Esto permite que cada celda de la tabla contenga únicamente un valor.

#### **2ª Forma Normal (2FN)**

Al tener todas las tablas una clave primaria claramente definida y no existir dependencias parciales (es decir, que un atributo dependa solo de una parte de una clave compuesta), podemos afirmar que también se cumple la 2FN. Aquellas tablas con claves compuestas, como EMPLEADOS\_PROYECTO, solo contienen atributos que dependen de la totalidad de la clave.

#### **3ª Forma Normal (3FN)**

Las relaciones también cumplen con la 3FN, ya que no existen dependencias transitivas (un atributo no clave que dependa de otro atributo no clave). La información está correctamente descompuesta entre entidades separadas, como PROVEEDORES, COMPONENTES o PROYECTOS, evitando duplicidades y redundancia innecesaria.

#### **Aplicación práctica**

La normalización permitió organizar la base de datos en una estructura clara donde:

* Los **clientes** y los **proyectos** están correctamente relacionados mediante claves externas.
* Los **empleados** pueden participar en múltiples proyectos y formaciones, manteniendo la trazabilidad de sus actividades.
* Las **máquinas**, **componentes** y sus **mantenimientos** están claramente separados pero interconectados, facilitando su gestión técnica.
* Las **incidencias**, **presupuestos**, **facturas** y **pedidos** permiten un seguimiento completo de la operativa y de la relación comercial con la clientela y proveedores.
* Las **garantías**, **documentación técnica** y **certificaciones** están asociadas a sus elementos correspondientes, garantizando el control de calidad y trazabilidad normativa.

# 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 provedor para saber a que provedor 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 estes 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 seguimento de las horas de trabajo de cada técnico, uso de actualización y eliminación en cascada.

# Carga de Datos Inicial

Describe aquí...

# Funciones y Procedimientos Almacenados

Describe aquí...

# Triggers

Describe aquí...

# Consultas SQL

Describe aquí...

# Casos de Prueba y Simulación

Describe aquí...

# Resultados y Verificación

Describe aquí...

# Capturas de Pantalla (opcional)

Describe aquí...

# Conclusiones y Mejoras Futuras

Describe aquí...

# Enlace al Repositorio en GitHub

Describe aquí...