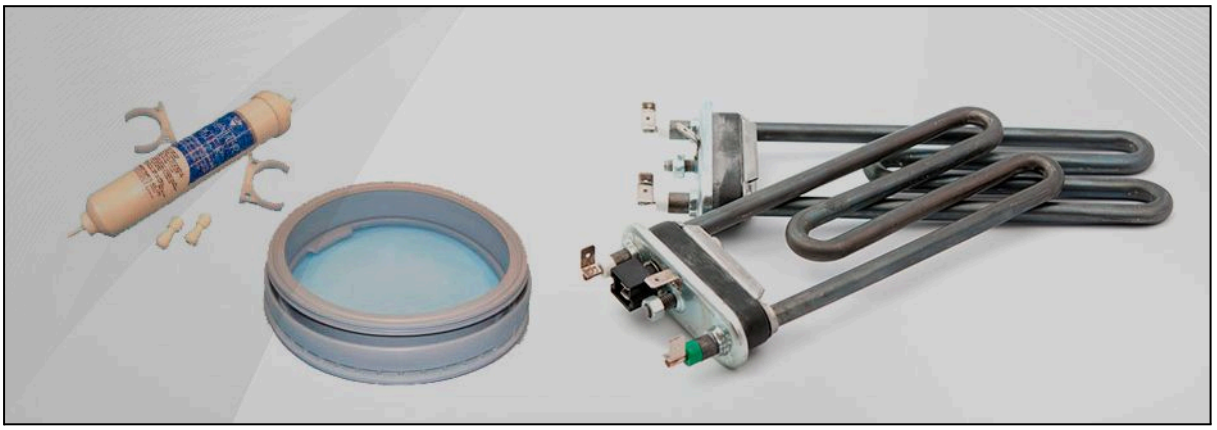
GENERALITAT VALENCIANA

Carlos Collado Calvo
DAM 1°



Microproyecto:

Inventario de piezas de electrodomésticos

Carlos Collado Calvo

03/05/2024

1. INTRODUCCIÓN

Necesitamos crear una aplicación para llevar el control del inventario de un pequeño almacén para piezas de electrodomésticos en un taller de reparación. La aplicación debe ser sencilla de usar, con capacidades limitadas, pues no es necesaria una gran complejidad debido a que por otras aplicaciones se llevan otros conceptos como facturación, fechas de llegada y salida de material, etc.

Teniendo ésto en cuenta más adelante expandiremos la información relativa a cada apartado.

2. HERRAMIENTAS Y MÉTODOS

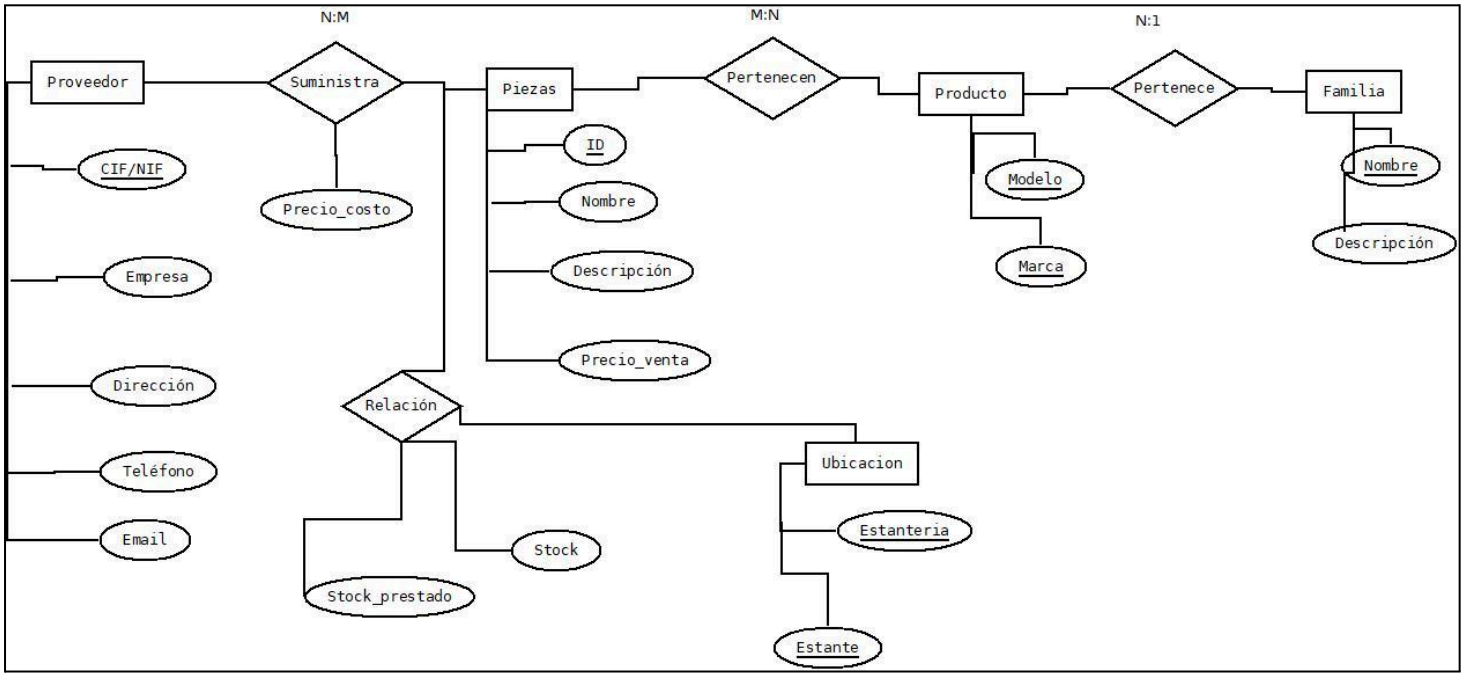
Breve descripción de las herramientas utilizadas en el proyecto:

MySQL	Gestión de Base de Datos
Draw.IO	Creación de Esquemas
DIA	Creación Esquemas E-R
ChatGPT	Consultas
Apuntes varios	Consultas
Ayuda profesional	Consultas y Apoyo Emocional

3. PERSPECTIVA ESTÁTICA

3.1. E/R (Entidad-Relación)

Diagrama que representa la estructura de las entidades y sus relaciones:



Mediante este primer esquema Entidad/Relación de la base de datos a utilizar para nuestra aplicación, obtenemos la información necesaria para obtener un inventario completo donde situemos tanto proveedores, piezas, productos e incluso el propio electrodoméstico, relacionando entre sí éstos elementos.

Sin embargo, a la hora de aplicar la base de datos a efectos de la aplicación a desarrollar, simplificaremos mucho la misma a fin de evitar redundancias y duplicidades innecesarias en otros departamentos de gestión de datos del taller dónde se pondrá en funcionamiento.

Por ejemplo, a efectos de control del inventario en un taller tan pequeño dónde el proveedor siempre es el mismo, no nos es necesario añadir en nuestra aplicación la opción de relacionar proveedores.

3.2. Pasos a tablas

Detalles sobre cómo se transformaron los diagramas E/R en tablas de base de datos:

Proveedor = (CIF_NIF, Empresa, Dirección, Teléfono, Email)
C.Alt: Empresa
VNN: Empresa

Piezas = (ID, Nombre, Descripción, Precio_venta)
VNN:Nombre

Producto = (Modelo, Marca)

Familia = (Nombre, Descripción)
VNN: Descripción

Ubicación = (Estantería, Estante)
R.I.: null(Estanteria) => null(Estante)

3.3. DDL (Data Definition Language)

Ejemplos de código que definen la estructura de la base de datos:

Para crear las tablas en mysql se han realizado las siguientes instrucciones:

```
CREATE DATABASE Inventario CHARACTER SET utf8 COLLATE utf8_spanish_ci;
use Inventario;

CREATE TABLE Proveedor (
  CIF varchar(10) NOT NULL,
  Empresa varchar(30) NOT NULL,
  Direccion varchar(50),
  Ciudad varchar (15),
  Telefono varchar (20),
  Email varchar(30),
  PRIMARY KEY (CIF)
) engine=innodb;

CREATE TABLE Pieza (
  ID varchar (30) NOT NULL,
  Nombre varchar(30) NOT NULL,
  Descripcion text NULL,
  Precio_venta double NOT NULL,
  PRIMARY KEY (ID)
) engine=innodb;

CREATE TABLE Pieza_Costo (
  CIF_Proveedor varchar(10) NOT NULL,
  ID_Pieza varchar(30) NOT NULL,
  Precio_costo double,
  PRIMARY KEY (CIF_Proveedor, ID_Pieza),
  CONSTRAINT PP_Pro_FK FOREIGN KEY (CIF_Proveedor) REFERENCES Proveedor
(CIF),
  CONSTRAINT PP_Pie_FK FOREIGN KEY (ID_Pieza) REFERENCES Pieza (ID)
) engine=innodb;

CREATE TABLE Ubicacion (
  Estanteria varchar(5),
  Estante varchar(5),
  PRIMARY KEY (Estanteria, Estante)
) engine=innodb;

CREATE TABLE Pieza_Ubicacion (
  ID_Pieza varchar(30) NOT NULL,
  EstanteriaUbicacion varchar(5)
  EstanteUbicacion varchar(5),
  Stock integer,
  Stock_prestado integer
  PRIMARY KEY (ID_Pieza, EstanteriaUbicacion, EstanteUbicacion),
  CONSTRAINT PU_Pie_FK FOREIGN KEY (ID_Pieza) REFERENCES Pieza (ID),
```

```
  CONSTRAINT PU_Ubi_FK FOREIGN KEY (EstanteriaUbicacion, EstanteUbica
REFERENCES Ubicacion (Estanteria, Estante)
) engine=innodb;

CREATE TABLE Familia (
  Nombre varchar (20),
  Descripcion text NULL,
  PRIMARY KEY (Nombre)
) engine=innodb;

CREATE TABLE Producto (
  Modelo varchar(30) NOT NULL,
  Marca varchar(15) NOT NULL,
  Medida int,
  Color varchar (15),
  NombreFamilia varchar (29),
  PRIMARY KEY (Modelo, Marca),
  CONSTRAINT Pro_Fam_FK FOREIGN KEY (NombreFamilia) REFERENCES Familia
(Nombre)
) engine=innodb;

CREATE TABLE Pieza_pertenece_Producto (
  ID_Pieza varchar (30) NOT NULL,
  ModeloProducto varchar(30) NOT NULL,
  MarcaProducto varchar(15) NOT NULL,
  PRIMARY KEY (ID_Pieza, ModeloProducto, MarcaProducto),
  CONSTRAINT Ppp_Pie_FK FOREIGN KEY (ID_Pieza) REFERENCES Pieza (ID),
  CONSTRAINT Ppp_Pro_FK FOREIGN KEY (ModeloProducto, MarcaProducto)
REFERENCES Producto (Modelo, Marca),
) engine=innodb;
```

3.4. DML (Data Manipulation Language)

Ejemplos de código que manipulan los datos en la base de datos:

Para poder tener información a consultar se han realizado una serie de instrucciones de "INSERT INTO" pudiendo insertar información en las tablas pertinentes.

```
INSERT INTO Proveedor VALUES ('9873490823','CNA Group','CALLE
PATATA','BARCELONA','+34 93 3561182','cacafuti@gmail.com');

INSERT INTO Pieza VALUES ('15101000', 'Motor', 'Grupo motor para campana
extractora de cocina', '95.59');
INSERT INTO Pieza VALUES ('485189911103', 'Condensador', 'Condensador 35 mm -
450V para lavadora', '6.85');
INSERT INTO Pieza VALUES ('83040745', 'Resistencia', 'Resistencia INFERIOR
900/300w para horno', '15.25');

INSERT INTO Pieza_Costo VALUES ('9873490823','15101000','47.40');
INSERT INTO Pieza_Costo VALUES ('7236483292','481946818049','14.97');
INSERT INTO Pieza_Costo VALUES ('9873490823','83040745','7.46');

INSERT INTO Ubicacion VALUES ('1','A');
INSERT INTO Ubicacion VALUES ('3','A');
INSERT INTO Ubicacion VALUES ('4','A');

INSERT INTO Pieza_Ubicacion VALUES ('15101000', '1', 'A', '9', '1');
INSERT INTO Pieza_Ubicacion VALUES ('481946818049', '3', 'A', '1', '0');
INSERT INTO Pieza_Ubicacion VALUES ('83040745', '4', 'A', '6', '0');

INSERT INTO Familia VALUES ('Campana', 'Extractor de humos activado por un
ventilador aspirador, con panel de control y temporizador.');
INSERT INTO Familia VALUES ('Lavadora', 'Lavadora carga frontal.');
INSERT INTO Familia VALUES ('Horno', 'Multifunción con HydrocleanECO, Sistema
de Limpieza automático, Eficiencia energética A+');

INSERT INTO Producto VALUES ('SYGMA', 'Cata', '90', 'Inox', 'Campana');
INSERT INTO Producto VALUES ('W19lMO1L', 'Whirlpool', 'NULL', 'Inox',
'Frigorifico');
INSERT INTO Producto VALUES ('HCB6535', 'Teka', '60', 'Inox', 'Horno');

INSERT INTO Pieza_pertenece_Producto VALUES ('15101000', 'SYGMA', 'Cata');
INSERT INTO Pieza_pertenece_Producto VALUES ('481946818049', 'W19lMO1L',
'Whirlpool');
INSERT INTO Pieza_pertenece_Producto VALUES ('83040745','HCB6535' 'Teka');
```

3.5. DQL (Data Query Language)

Ejemplos de código que consultan la base de datos:

1ª: Consulta de la tabla “Producto” dónde queremos que salgan los campos “Modelo” y “Marca” ordenados alfabéticamente de manera descendente:

```
SELECT
      Modelo,
      Marca
FROM Producto
ORDER BY desc
```

2ª: Consulta de la tabla “Pieza” dónde queremos que nos cuente el número de piezas existentes:

```
SELECT COUNT(ID) as Num_de_Piezas
FROM Pieza;
```

3ª: Consulta de la tabla “Pieza” dónde queremos saber el precio de la pieza con referencia “15101000”:

```
SELECT  Precio_venta
FROM  Pieza
WHERE ID="15101000";
```

3.6. DCL (Data Control Language)

Ejemplos de código que controlan los permisos y la seguridad de la base de datos:

En primer lugar le quiero dar permiso al usuario “user1” para poder meter información o realizar una consulta en la tabla de “Pieza”:

```
GRANT SELECT, INSERT ON Pieza TO user1;
```

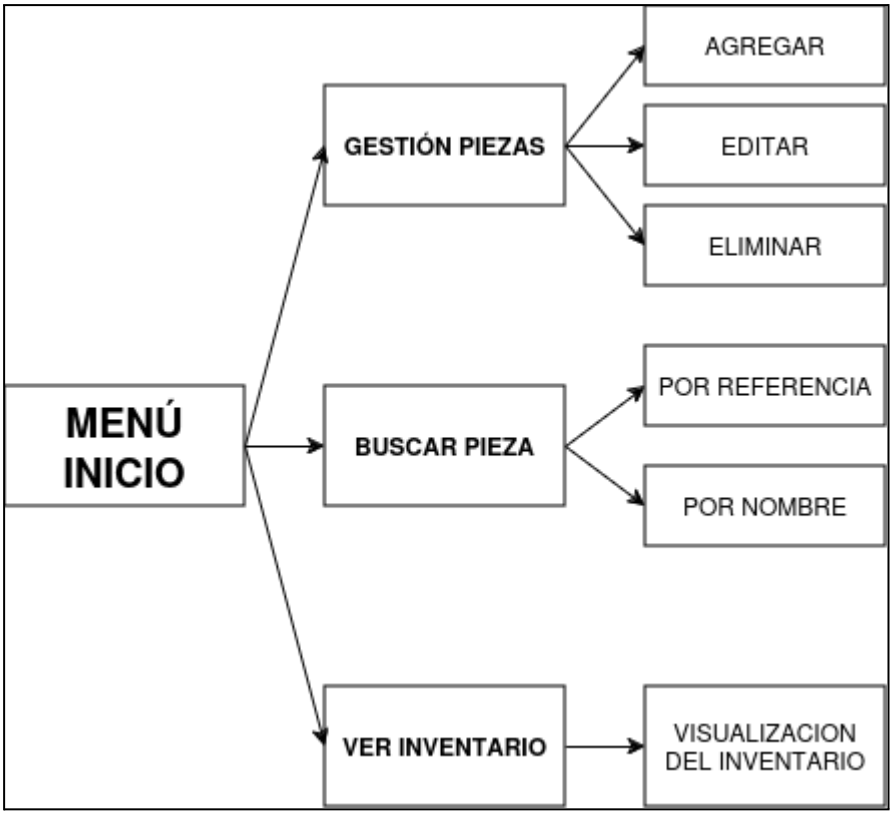
Sin embargo, cambio de opinión y sólo quiero darle permiso de consultar, por lo que le quito el de meter información:

```
REVOKE INSERT ON Pieza FROM user1;
```

Finalmente, quiero impedir que el usuario “user2” pueda borrar información de la tabla “Proveedor” por lo que utilizo la siguiente instrucción:

```
DENY DELETE ON Proveedor TO user2;
```

4. PERSPECTIVA DINÁMICA



5. CONCLUSIONES

Consideramos este proyecto un buen inicio de lo que querríamos que fuera una aplicación completa y con mucha más funcionalidad a la hora de ponerlo en práctica en el taller de reparación de electrodomésticos. La base de datos generada da para mucha más utilidad que la que en principio puede funcionar con el esquema de la interfaz, ya que no sólo se puede consultar, añadir, modificar o eliminar productos, sino que se podría insertar hasta la ubicación en físico de los productos y/o diferentes proveedores.

Además, se queda pendiente la opción de poder dar de alta a usuarios (futuros empleados) y darles permisos para poder manipular y operar con el inventario sin necesidad de que sea una sola persona la que lo utilice o directamente carecer de seguridad.

Aun así, consideramos que esta aplicación, aunque sencilla, proporciona una organización y ayuda extra, futuramente mejorable.