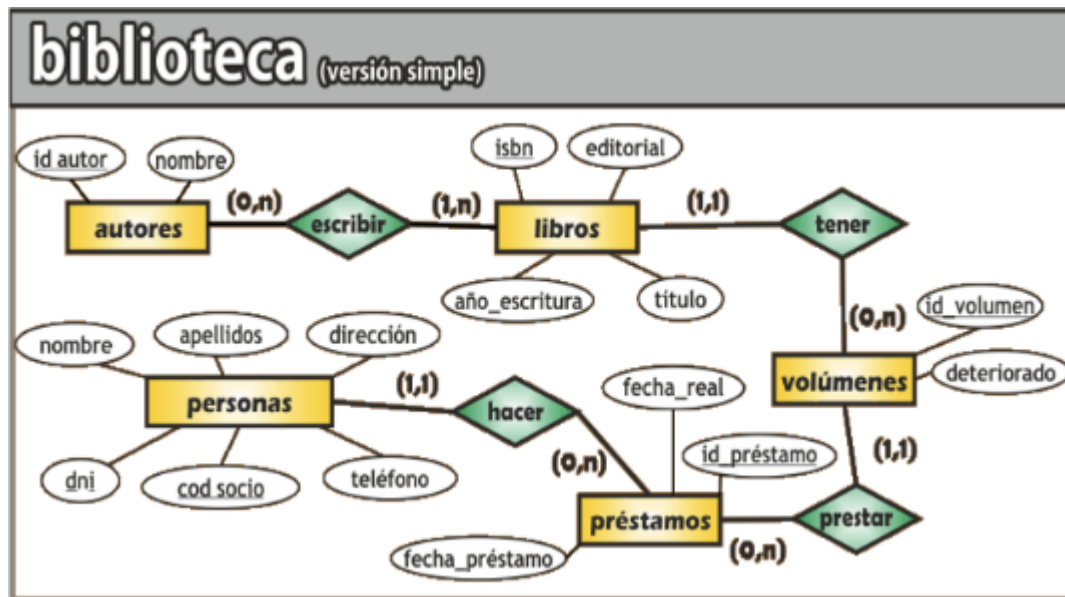


Modelo relacional

1. -



Autores(id autor, Nombre)

Libros(ISBN, Editorial, Año_escritura, Titulo)

Escribir(id autor, ISBN)

Volumenes(id volumen, deteriorado, isbn libro)

Personas(dni, cod socio, nombre, apellidos, direccion, telefono)

Préstamos(id préstamo, fecha préstamo, fecha_real, id volumen, id persona)

create database biblioteca;

use biblioteca;

create table autores(id varchar(20), nombre varchar(40), PRIMARY KEY(id));

create table libros(isbn varchar(20), editorial varchar(40), titulo varchar(35), anio_escritura int, varchar(20) PRIMARY KEY(isbn));

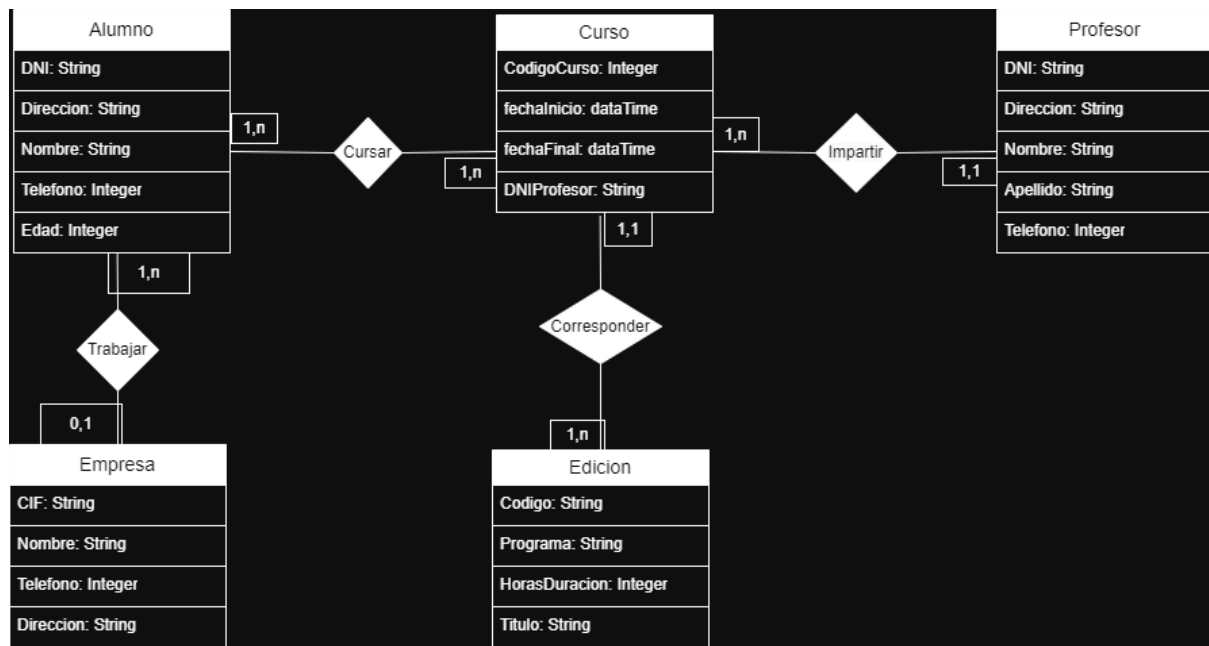
create table escribir(isbn varchar(20), id_autor varchar(20), PRIMARY KEY(isbn, id_autor));

create table volumenenes(id varchar(20), deteriorado boolean, isbn varchar(20), foreign key (isbn) references libros(isbn) on delete cascade on update cascade, PRIMARY KEY(isbn, id_autor));

create table personas(dni varchar(10), nombre varchar(30), apellidos varchar(40), direccion varchar(50), telefono varchar(12), cod_socio varchar(20), UNIQUE (cod_socio), PRIMARY KEY(isbn, id_autor));

create table prestamos(id varchar(20), fecha_real date, fecha_prestamo date, id_volumen varchar(20), id_persona varchar(10), foreign key (id_volumen) references volumen(id) on delete cascade on update cascade,foreign key (id_persona) references persona(dni) on delete cascade on update cascade, PRIMARY KEY(id));

2.-



Empresa(cif, Nombre, Telefono, Direccion)

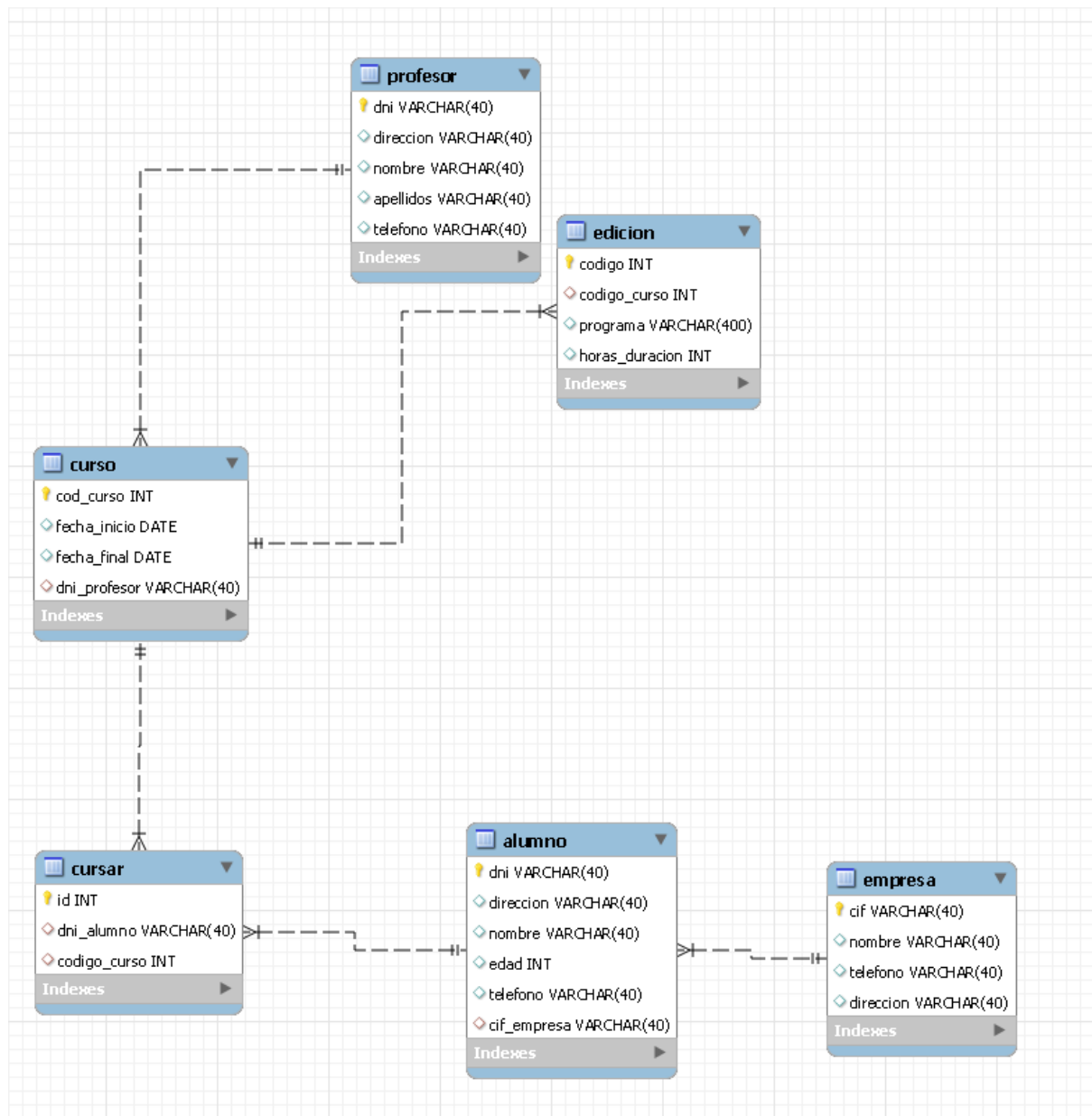
Profesor(DNI, Dirección, Nombre, Apellido, Telefono)

Alumno(dni_alumno, Direccion, Nombre, telefono, edad, cif_empresa)

Curso(CodigoCurso, fechaInicio, fechaFinal, DNIProfesor)

Cursar(dni, CodigoCurso)

Edicion(Codigo, CodigoCurso, programa, HorasDuración, Titulo)



```

create database academia;
use academia;

```

```

create table empresa(cif varchar(40),nombre varchar(40), telefono varchar(40),direccion
varchar(40), PRIMARY KEY(cif) );
create table profesor(dni varchar(40),direccion varchar(40), nombre varchar(40),apellidos
varchar(40),telefono varchar(40), PRIMARY KEY(dni) );
create table alumno(dni varchar(40),direccion varchar(40), nombre varchar(40),edad
int,telefono varchar(40),cif_empresa varchar(40),
foreign key (cif_empresa) references empresa(cif) on delete cascade on update cascade,
PRIMARY KEY(dni) );

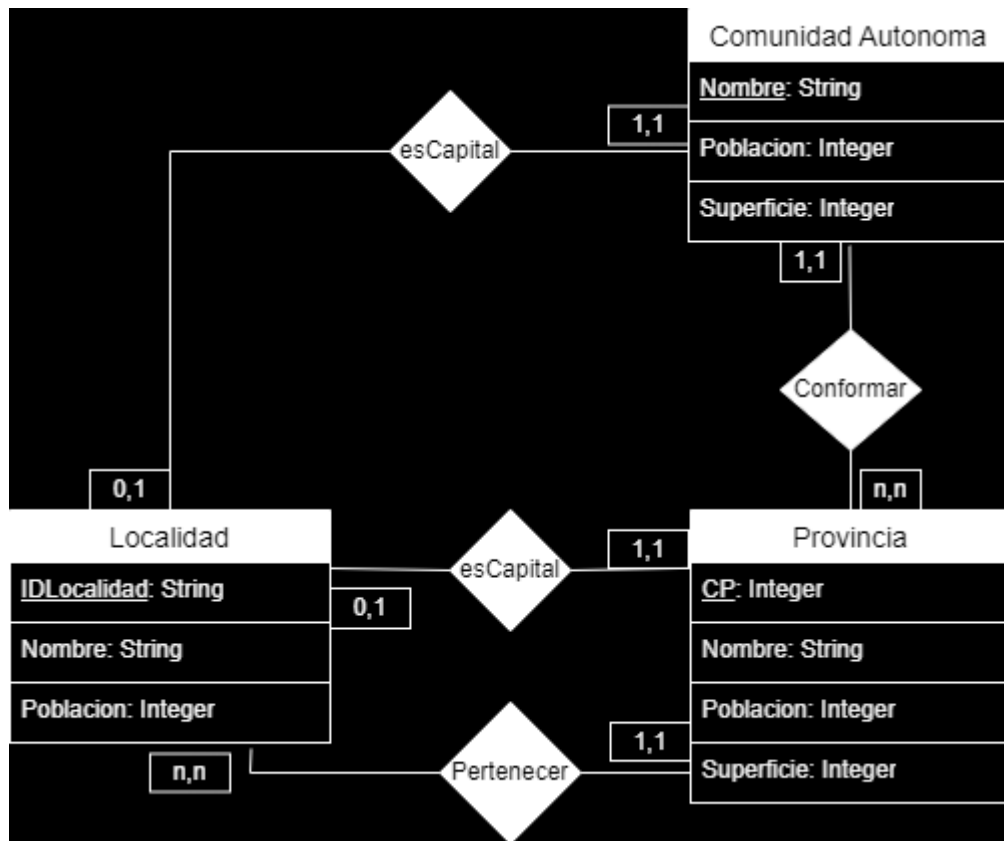
```

```
create table curso(cod_curso int,fecha_inicio date,fecha_final date,dni_profesor varchar(40),
PRIMARY KEY(cod_curso),
foreign key (dni_profesor) references profesor(dni) on delete cascade on update cascade );
```

```
create table cursar(id int auto_increment, dni_alumno varchar(40),codigo_curso int,
PRIMARY KEY(id),
foreign key (dni_alumno) references alumno(dni) on delete cascade on update cascade,
foreign key (codigo_curso) references curso(cod_curso) on delete cascade on update
cascade );
```

```
create table edicion(codigo int,codigo_curso int,programa varchar(400),horas_duracion int,
PRIMARY KEY(codigo),
foreign key (codigo_curso) references curso(cod_curso) on delete cascade on update
cascade );
```

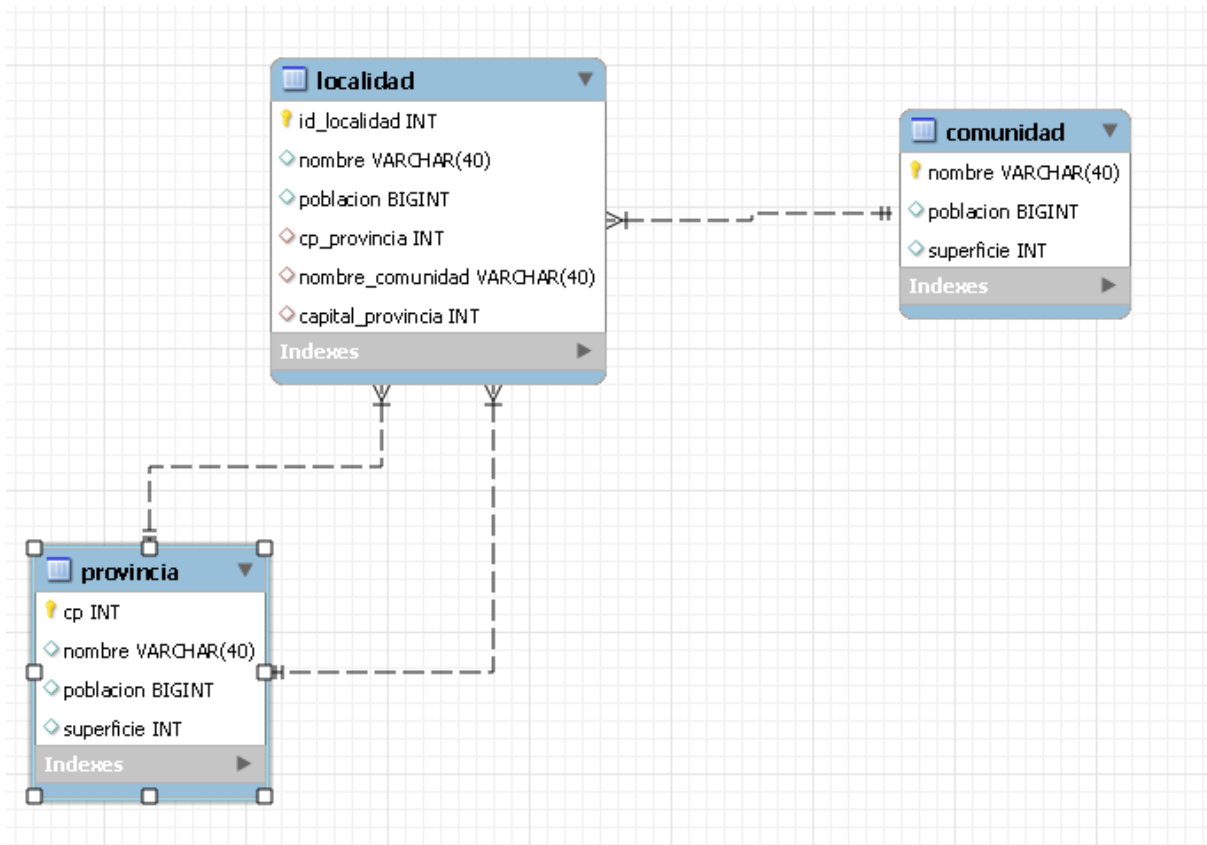
3.-



Comunidad_autonoma(nombre, poblacion, superficie)

Provincia(cp, nombre, poblacion, superficie, nomb_com_auton)

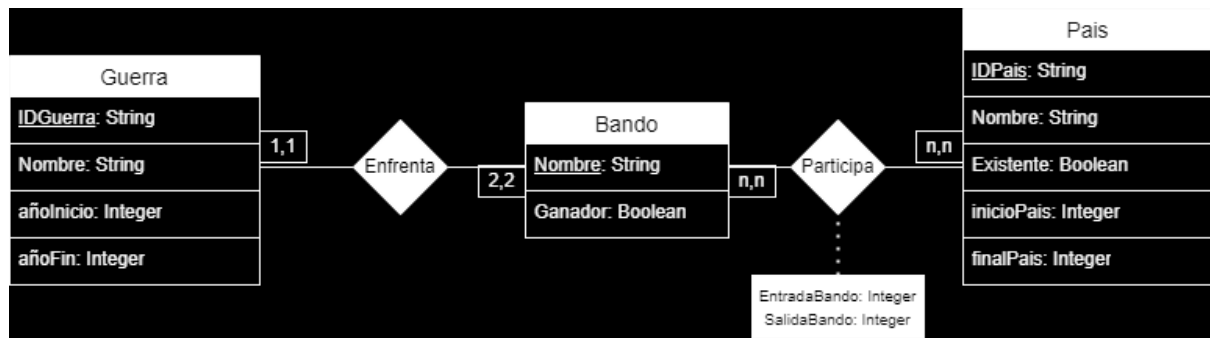
Localidad(id_localidad, nombre, poblacion, cp_provincia, id_capit_com, id_capit_prov)



```
create database provincias;
use provincias;
```

```
create table comunidad(nombre varchar(40),poblacion bigint, superficie int, primary
key(nombre));
create table provincia(cp int(2),nombre varchar(40),poblacion bigint, superficie int, primary
key(cp));
create table localidad(id_localidad int,nombre varchar(40),poblacion bigint,cp_provincia
int,nombre_comunidad varchar(40),capital_provincia int, primary key(id_localidad),
foreign key (cp_provincia) references provincia(cp),
foreign key (capital_provincia) references provincia(cp),
foreign key (nombre_comunidad) references comunidad(nombre)
);
```

4.-

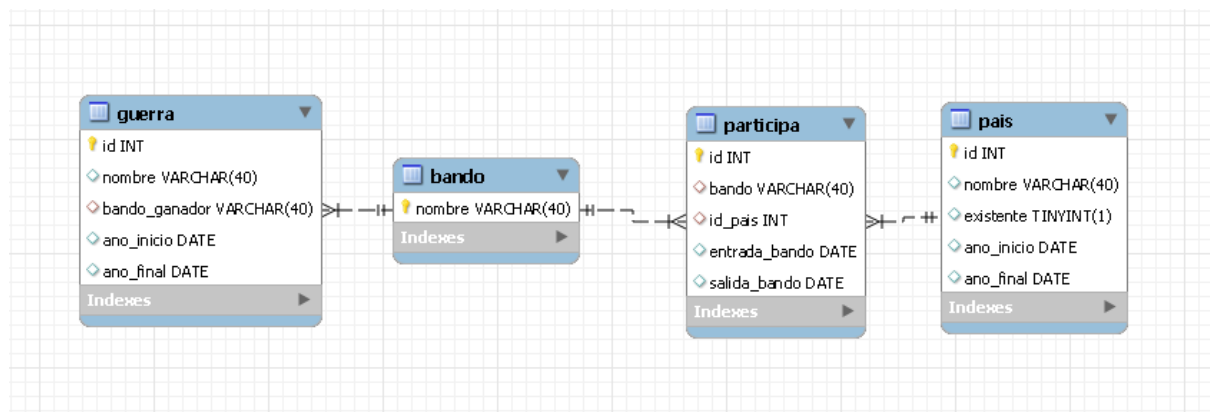


Bando(nombre)

Guerra(id_guerra, bando_ganador, nombre, ano_inicio, ano_fin, nombre_bando)

Pais(id_pais, nombre, existente, inicio_pais, final_pais)

Participa(id, id_bando, id_pais, entrada_bando, salida_bando)



```
create database guerras;
```

```
use guerras;
```

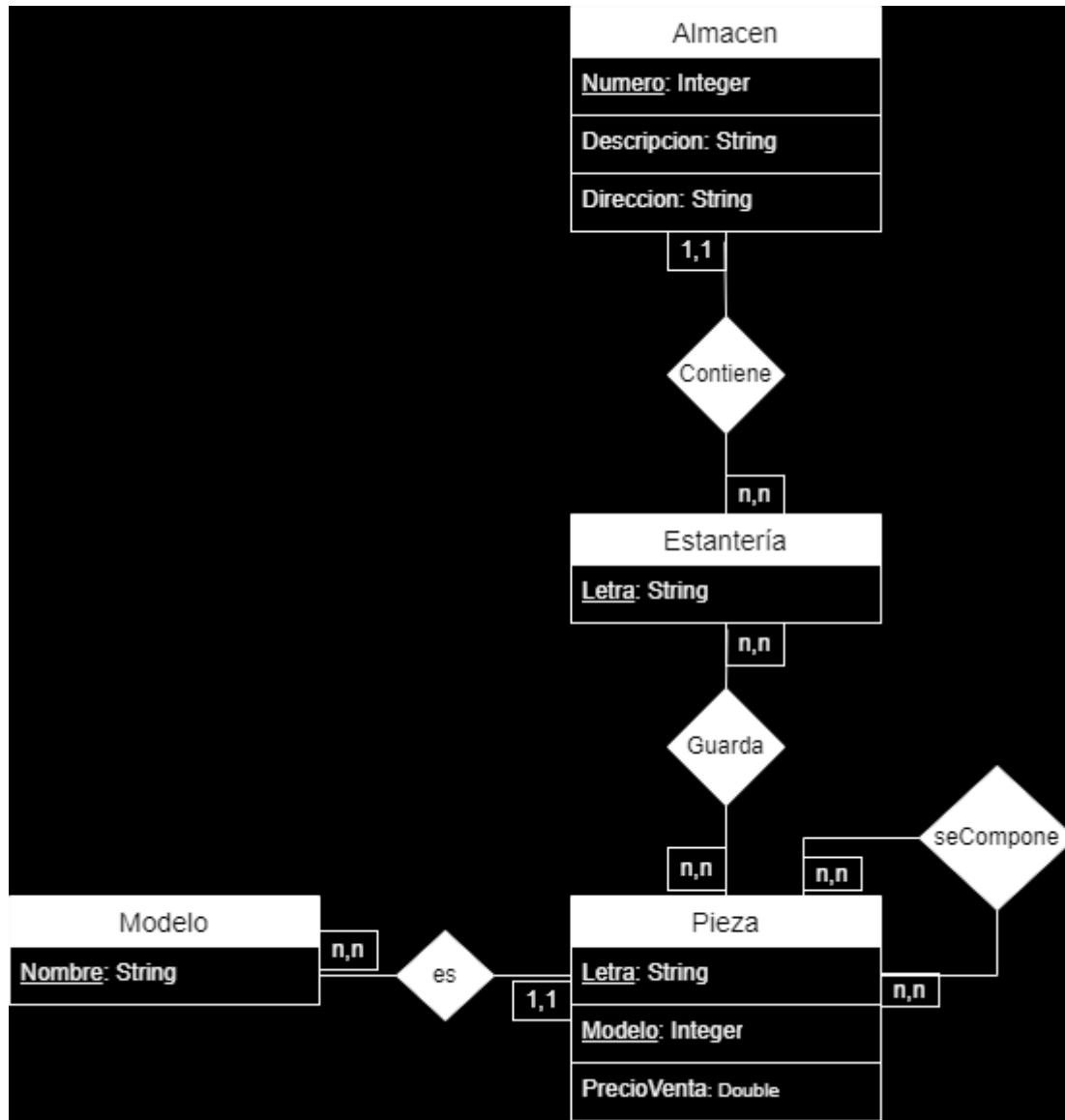
```
create table bando(nombre varchar(40), primary key(nombre));
```

```
create table guerra(id int auto_increment, nombre varchar(40),bando_ganador varchar(40),
ano_inicio date,ano_final date, primary key(id),
foreign key (bando_ganador) references bando(nombre)
);
```

```
create table pais(id int auto_increment,nombre varchar(40),existente boolean,ano_inicio
date,ano_final date, primary key(id));
```

```
create table participa(id int auto_increment, bando varchar(40),id_pais int, entrada_bando
date, salida_bando date, primary key(id),
foreign key (bando) references bando(nombre),
foreign key (id_pais) references pais(id)
);
```

5.-



Estanteria(Letra)
 Almacen(Numero, Descripcion, Direccion)
 Modelo(nombre)
 Pieza(Letra, nombre_modelo, precio_venta)
 seCompone(id, letra_pieza, letra_pieza)
 Guarda(id, letra_pieza, letra_estanteria)
 Contiene(id, almacen_id, estanteria_id)

```

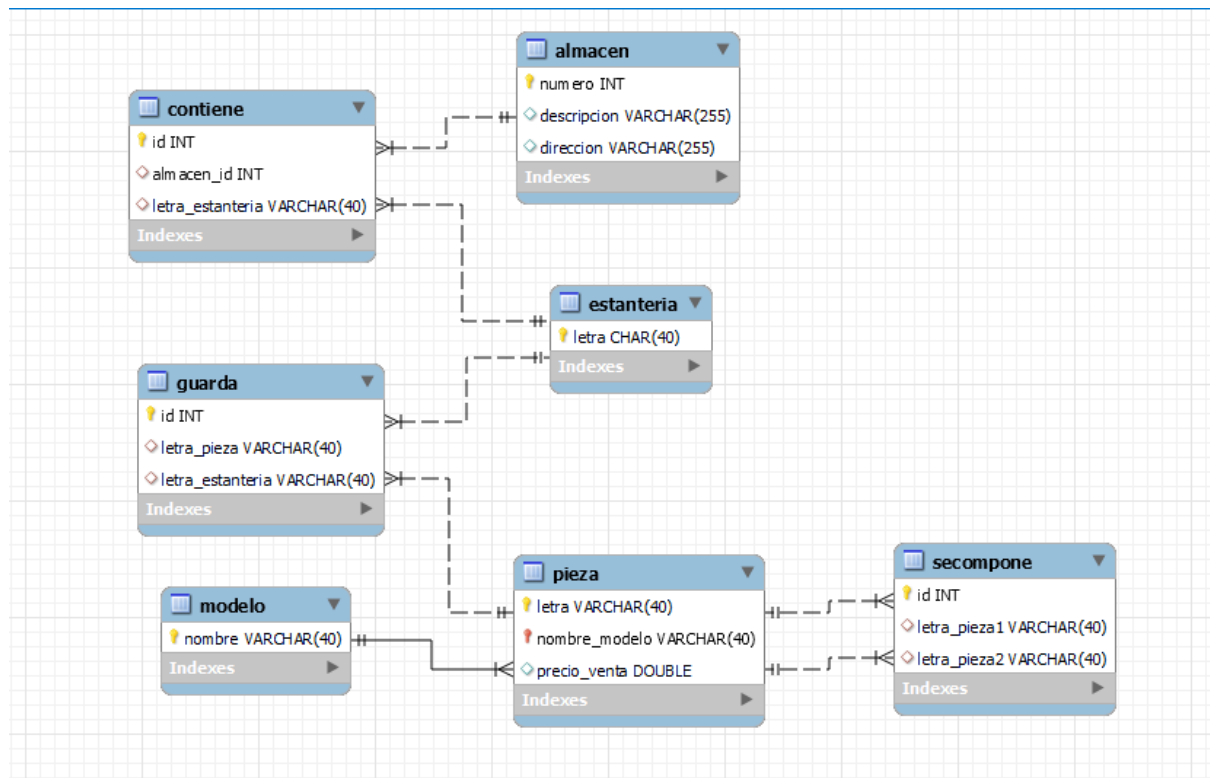
CREATE DATABASE almacenDeEstanteria;
USE almacenDeEstanteria;
CREATE TABLE estanteria (
    letra CHAR(40) PRIMARY KEY
);
CREATE TABLE almacen (
    numero INT PRIMARY KEY,
    descripcion VARCHAR(255),
  
```



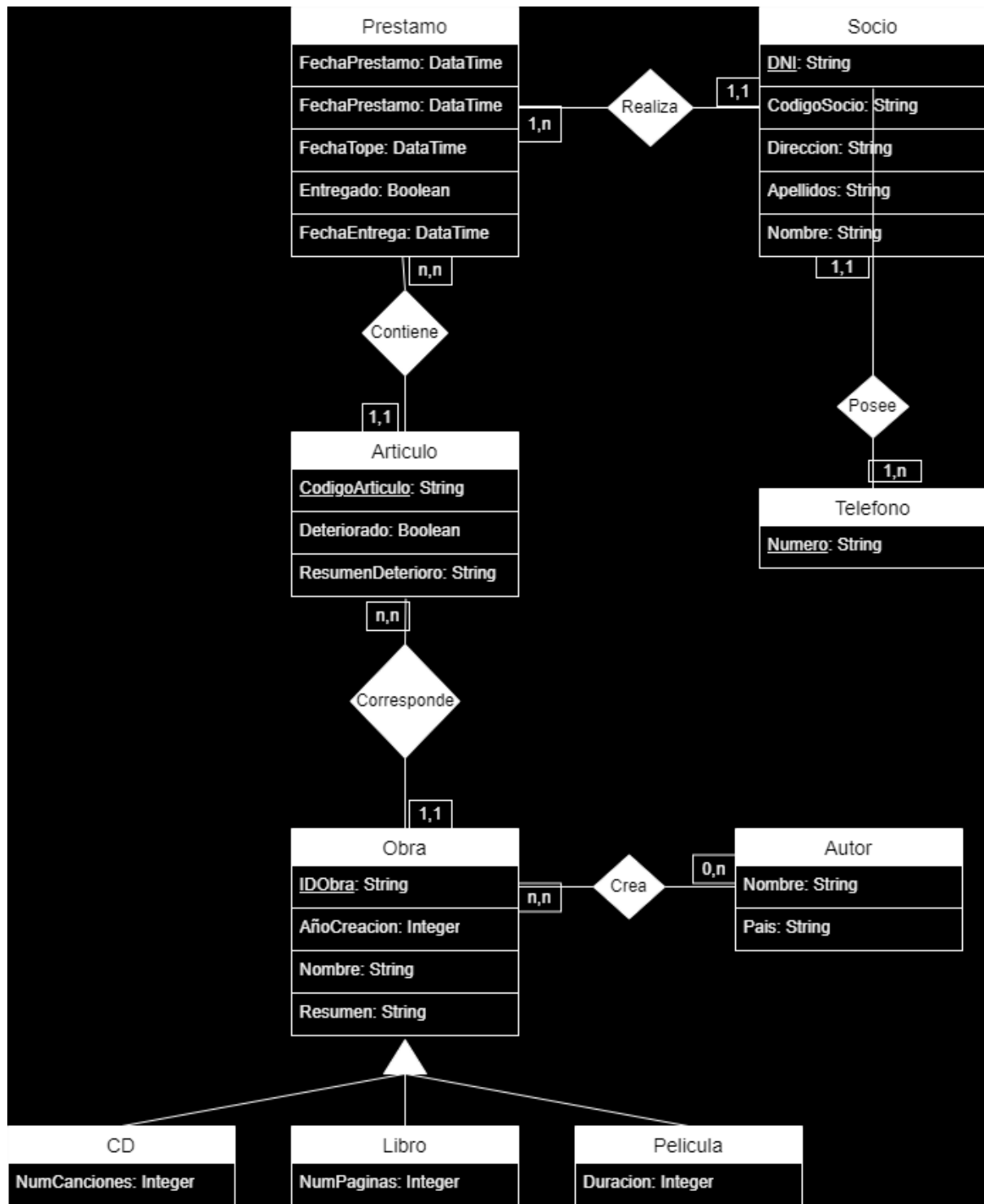
```

    direccion VARCHAR(255)
);
CREATE TABLE modelo (
    nombre VARCHAR(40) PRIMARY KEY
);
CREATE TABLE pieza (
    letra VARCHAR(40),
    nombre_modelo VARCHAR(40),
    precio_venta DOUBLE,
    PRIMARY KEY (letra, nombre_modelo),
    FOREIGN KEY (nombre_modelo) REFERENCES modelo(nombre)
);
CREATE TABLE secompone (
    id INT AUTO_INCREMENT PRIMARY KEY,
    letra_pieza1 VARCHAR(40),
    letra_pieza2 VARCHAR(40),
    FOREIGN KEY (letra_pieza1) REFERENCES pieza(letra),
    FOREIGN KEY (letra_pieza2) REFERENCES pieza(letra)
);
CREATE TABLE guarda (
    id INT AUTO_INCREMENT PRIMARY KEY,
    letra_pieza VARCHAR(40),
    letra_estanteria VARCHAR(40),
    FOREIGN KEY (letra_pieza) REFERENCES pieza(letra),
    FOREIGN KEY (letra_estanteria) REFERENCES estanteria(letra)
);
CREATE TABLE contiene (
    id INT AUTO_INCREMENT PRIMARY KEY,
    almacen_id INT,
    letra_estanteria VARCHAR(40),
    FOREIGN KEY (almacen_id) REFERENCES almacen(numero),
    FOREIGN KEY (letra_estanteria) REFERENCES estanteria(letra)
);

```



6.-



Socio(dni, codigo_socio, direccion, apellido, nombre)

Telefono(numero, socio)

Obra(id_obra, anio_creacion, nombre, resumen)

Articulo(codigo_articulo, deteriorado, resumen_deterioro, id_obra)

Prestamo(id_prestamo, fecha_prestamo, fecha_tope, entregado, fecha_entrega, dni_socio, cod_articulo)

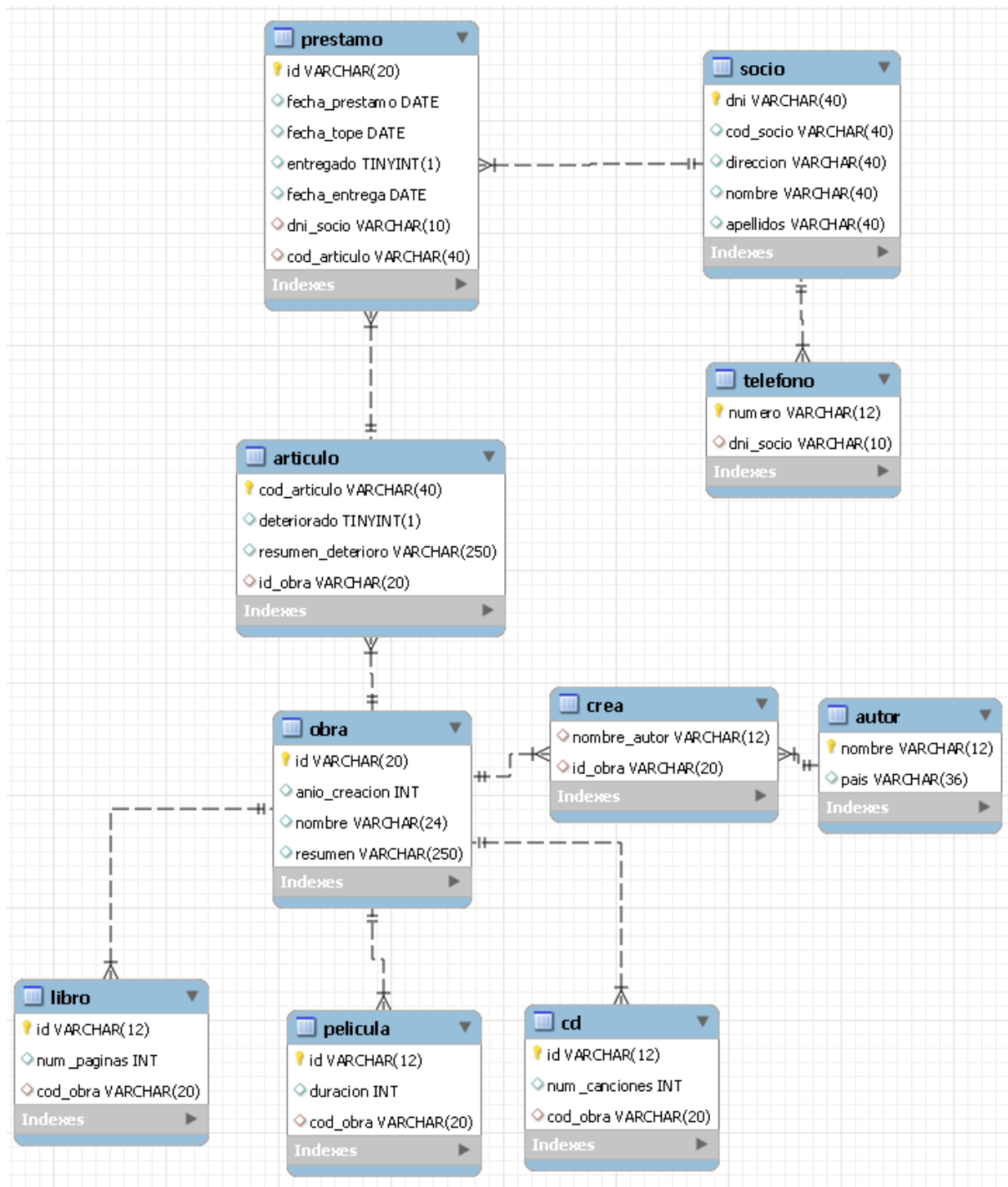
Autor(nombre, pais)

Crea(id_obra, nombre)

CD(num_canciones, id_obra)

Libro(num_paginas, id_obra)

Pelicula(duracion, id_obra)



create database bibliotecav2;

use bibliotecav2;

```
create table socio(dni varchar(40), cod_socio varchar(40), UNIQUE (cod_socio), direccion
varchar(40), nombre varchar(40),apellidos varchar(40), PRIMARY KEY(dni) );
```

```
create table telefono(numero varchar(12), dni_socio varchar(10), foreign key (dni_socio )
references socio (dni) on delete cascade on update cascade, PRIMARY KEY(numero) );
```

```
create table obra(id varchar(20), anio_creacion int, nombre varchar(24), resumen
varchar(250), PRIMARY KEY(id) );
create table articulo(cod_articulo varchar(40),deteriorado boolean, resumen_deterioro
varchar(250), id_obra varchar(20), foreign key (id_obra) references obra(id), PRIMARY
KEY(cod_articulo) );
```

```
create table prestamo(id varchar(20), fecha_prestamo date, fecha_tope date, entregado
boolean, fecha_entrega date, dni_socio varchar(10), cod_articulo varchar(40), foreign key
(dni_socio) references socio (dni) on delete cascade on update cascade, foreign key
(cod_articulo ) references articulo(cod_articulo ) on delete cascade on update cascade,
PRIMARY KEY(id) );
```

```
create table autor(nombre varchar(12), pais varchar(36), PRIMARY KEY(nombre) );
```

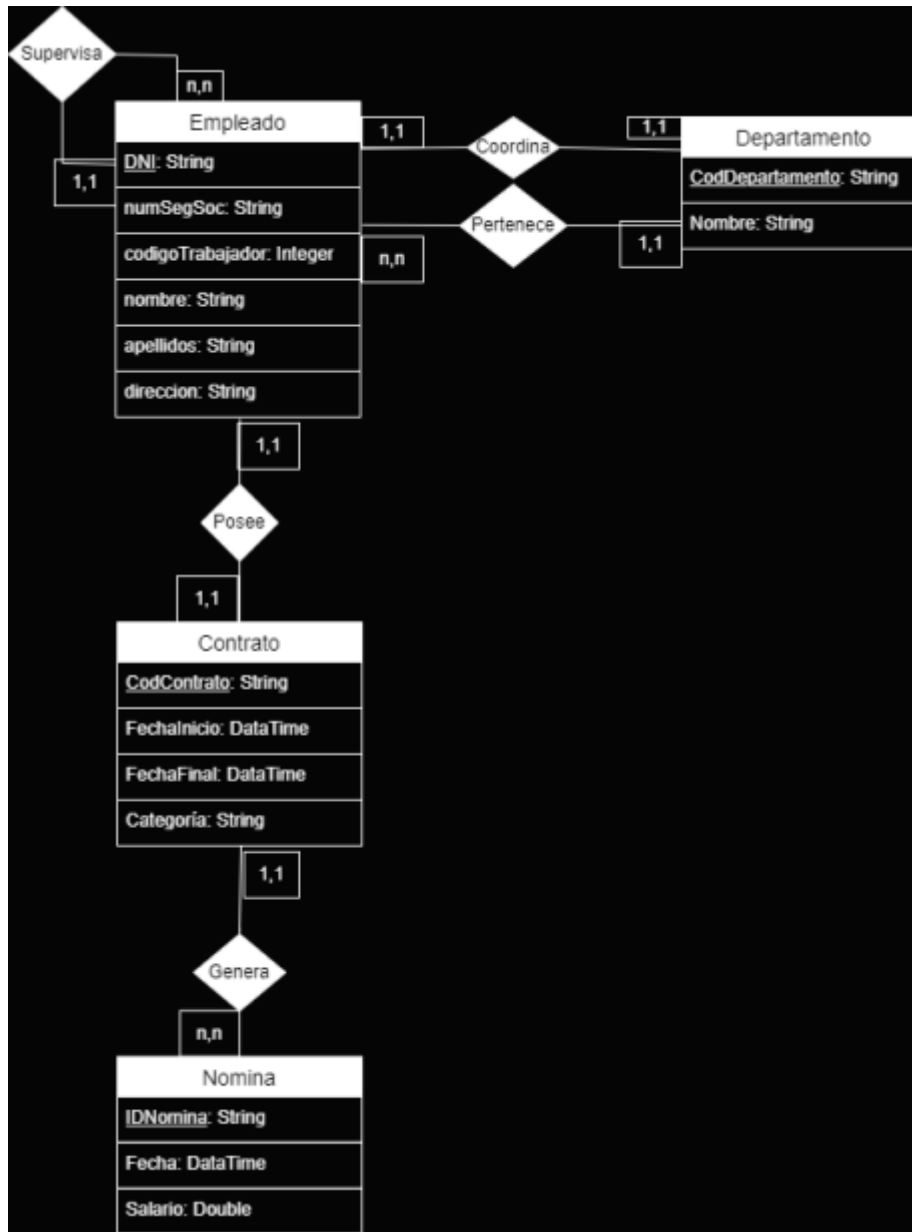
```
create table crea(nombre_autor varchar(12), id_obra varchar(20), foreign key
(nombre_autor) references autor(nombre) on delete cascade on update cascade,
foreign key (id_obra ) references obra(id) on delete cascade on update cascade );
```

```
create table cd(id varchar(12), num_canciones int, PRIMARY KEY(id),cod_obra
varchar(20), foreign key (cod_obra) references obra(id) );
```

```
create table libro(id varchar(12), num_paginas int, PRIMARY KEY(id), cod_obra
varchar(20), foreign key (cod_obra) references obra(id) );
```

```
create table pelicula(id varchar(12), duracion int, PRIMARY KEY(id), cod_obra varchar(20),
foreign key (cod_obra) references obra(id) );
```

7.-



Contrato(COD_CONTRATO, fecha_inicio, fecha_final, categoria)

Empleado(DNI, numSS, codigo_trabajador, nombre, apellidos, direccion, contrato_id)

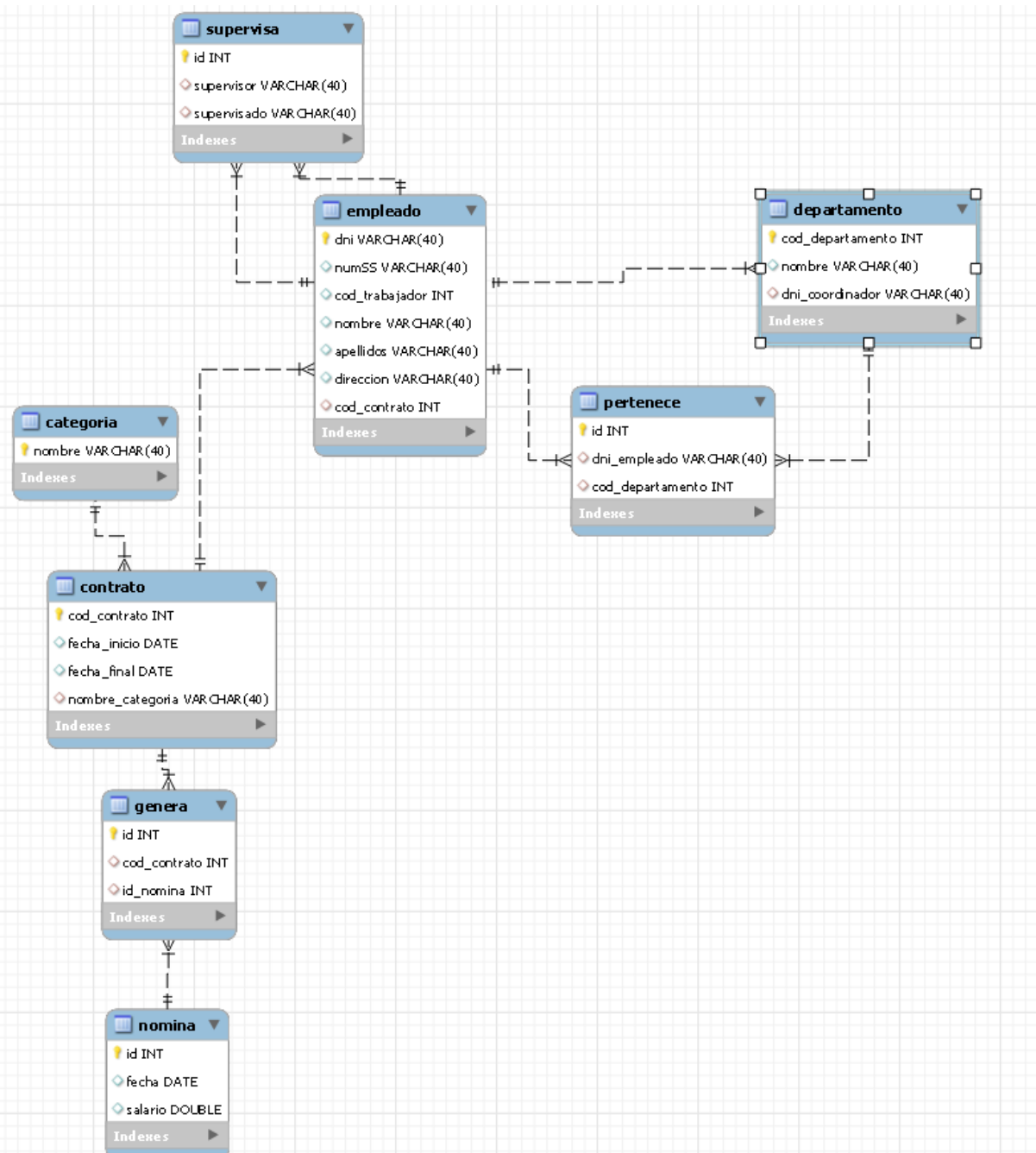
Departamento(COD_DEPARTAMENTO, nombre, dni_coordinador)

Supervisa(id, dni_empleado, dni_empleado)

Pertenece(id, dni_empleado, cod_departamento)

Nomina(ID_NOMINA, fecha, salario)

Genera(id, cod_contrato, id_nomina)



create database contratos;
use contratos;

```

create table categoria(nombre varchar(40),primary key(nombre));
create table contrato(cod_contrato int,fecha_inicio date,fecha_final date,nombre_categoria
varchar(40), primary key(cod_contrato),
foreign key (nombre_categoria) references categoria(nombre)
);
create table empleado(dni varchar(40),numSS varchar(40),cod_trabajador int, nombre
varchar(40),apellidos varchar(40),direccion varchar(40),cod_contrato int,primary key(dni),
foreign key (cod_contrato) references contrato(cod_contrato)
);
  
```

```
create table departamento(cod_departamento int,nombre varchar(40),dni_coordinador
varchar(40), primary key(cod_departamento),
foreign key (dni_coordinador) references empleado(dni)
);
```

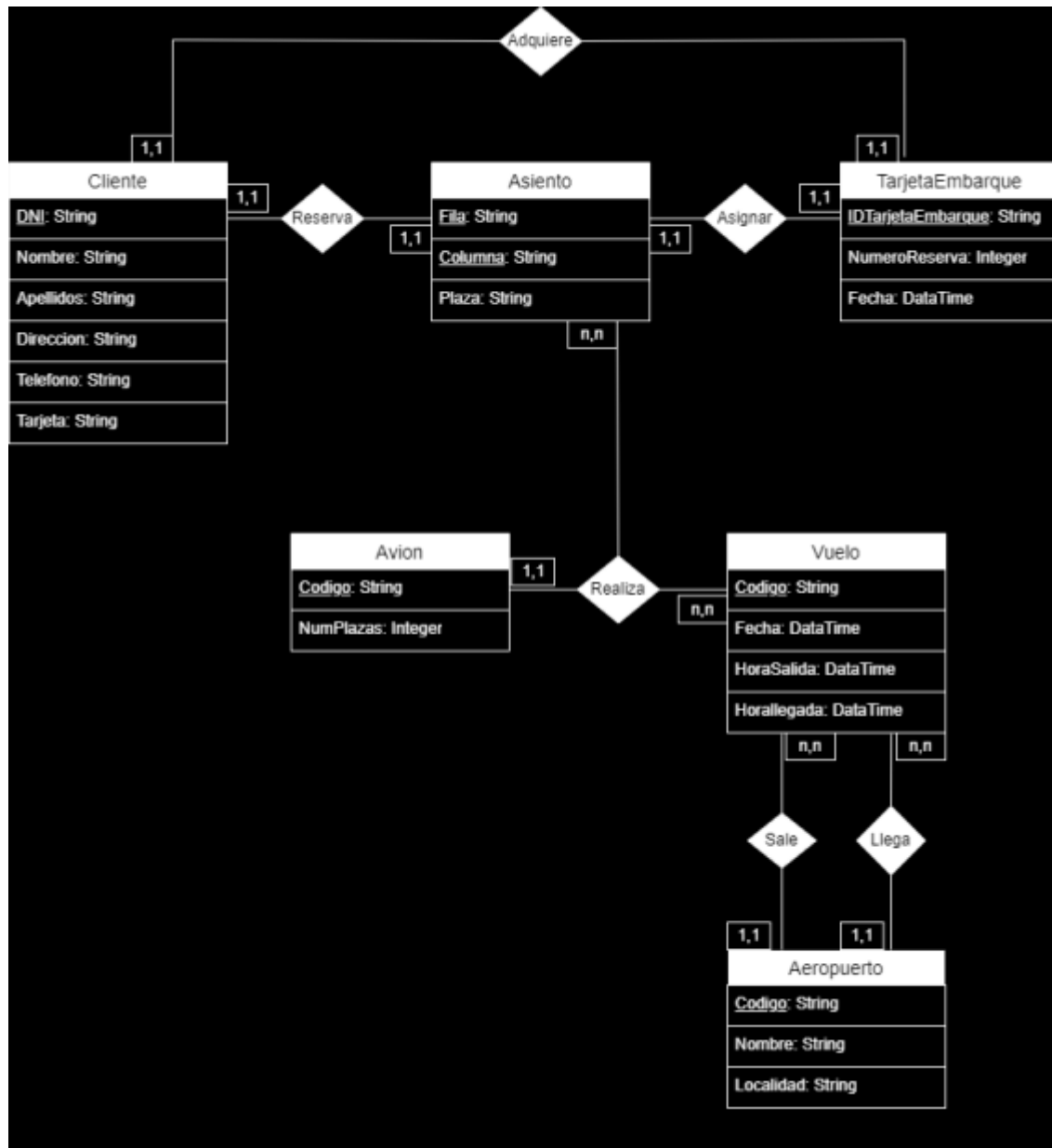
```
create table supervisa(id int auto_increment,supervisor varchar(40),supervisado varchar(40),
primary key(id),
foreign key (supervisor) references empleado(dni),
foreign key (supervisado) references empleado(dni)
);
```

```
create table pertenece(id int auto_increment,dni_empleado varchar(40),cod_departamento
int, primary key(id),
foreign key (dni_empleado) references empleado(dni),
foreign key (cod_departamento) references departamento(cod_departamento)
);
```

```
create table nomina(id int auto_increment,fecha date,salario double, primary key(id));
```

```
create table genera(id int auto_increment,cod_contrato int, id_nomina int, primary key(id),
foreign key (id_nomina) references nomina(id),
foreign key (cod_contrato) references contrato(cod_contrato)
);
```


8.-



Aeropuerto(codigo_aerop, nombre, localidad)

Cliente(dni, nombre, apellidos, direccion, telefono, tarjeta)

Tarjeta_embarque(id_tarj_emb, num_reserva, fecha)

Asiento(fila, columna, plaza, dni_cliente, id_tarj_emb)

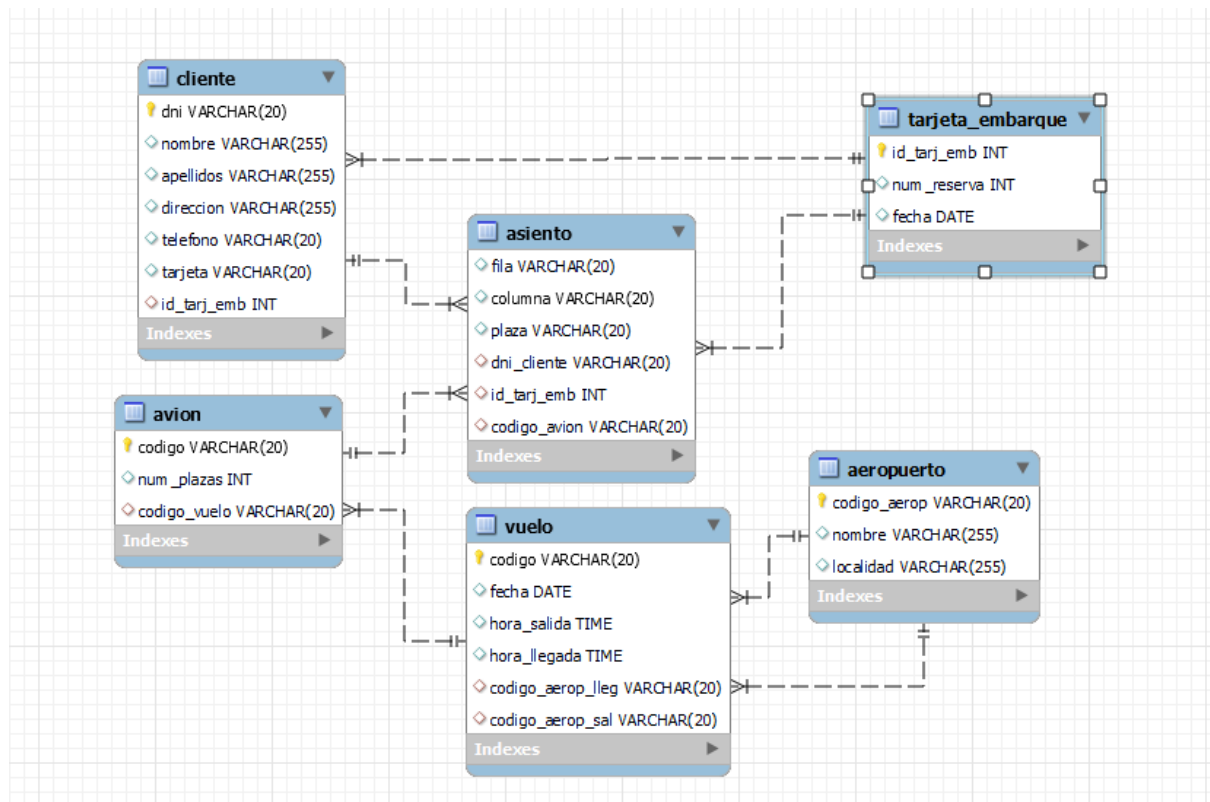
Avion(codigo, num_plazas)

Vuelo(codigo, fecha, hora_salida, hora_llegada, codigo_aerop_lleg, codigo_aerop_sal)

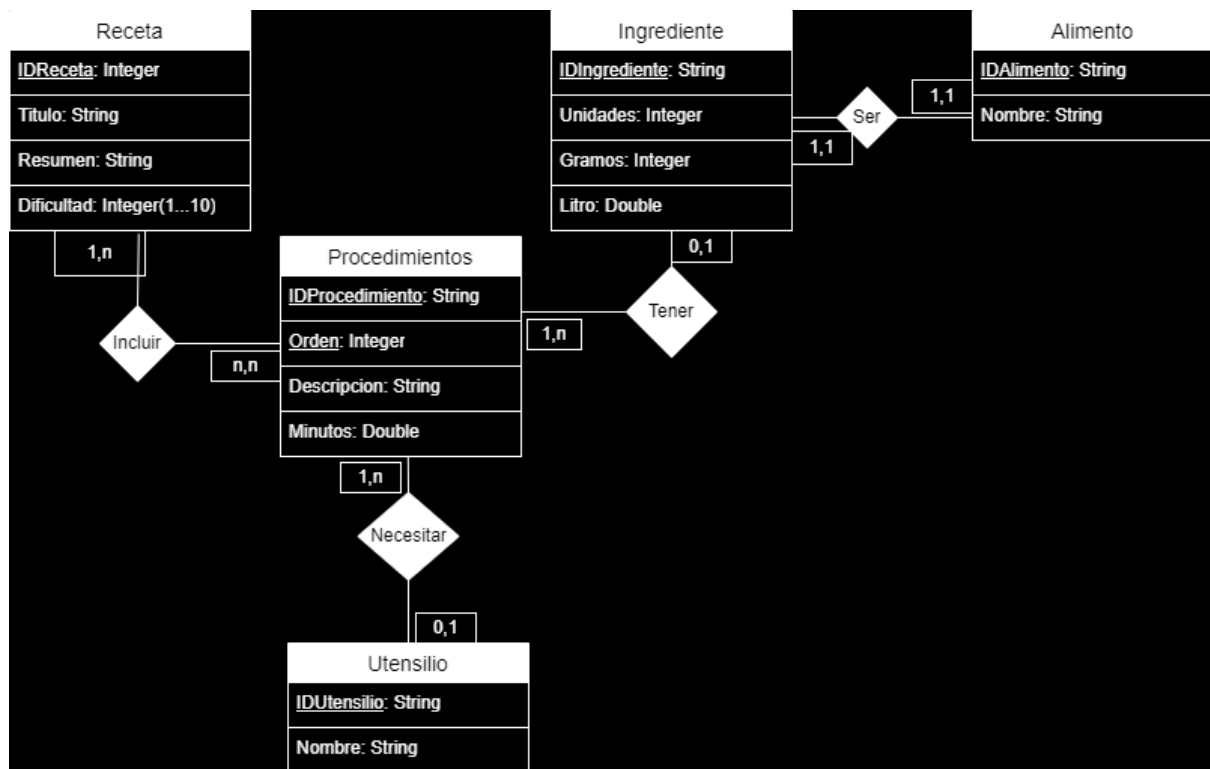
Realiza(cod_avion, **cod_vuelo**, **fila**, **columna**)

```
create database losVuelos;  
use losVuelos;
```

```
create table tarjeta_embarque (  
    id_tarj_emb int auto_increment primary key,num_reserva int,fecha date  
);  
create table cliente (  
    dni varchar(20) primary key,nombre varchar(255),apellidos varchar(255),direccion  
    varchar(255),telefono varchar(20),tarjeta varchar(20),id_tarj_emb int,  
    foreign key (id_tarj_emb) references tarjeta_embarque(id_tarj_emb)  
);  
create table aeropuerto (  
    codigo_aerop varchar(20) primary key,nombre varchar(255),localidad varchar(255)  
);  
create table vuelo (  
    codigo varchar(20) primary key,fecha date,hora_salida time,hora_llegada  
    time,codigo_aerop_lleg varchar(20),codigo_aerop_sal varchar(20),  
    foreign key (codigo_aerop_lleg) references aeropuerto(codigo_aerop),foreign key  
    (codigo_aerop_sal) references aeropuerto(codigo_aerop)  
);  
create table avion (  
    codigo varchar(20) primary key,num_plazas int,codigo_vuelo varchar(20),  
    foreign key (codigo_vuelo) references vuelo(codigo)  
);  
create table asiento (  
    fila varchar(20),columna varchar(20),plaza varchar(20),dni_cliente  
    varchar(20),id_tarj_emb int, codigo_avion varchar(20),  
    foreign key (dni_cliente) references cliente(dni),  
    foreign key (id_tarj_emb) references tarjeta_embarque(id_tarj_emb),  
    foreign key (codigo_avion) references avion(codigo)  
  
);
```



9.-



Receta(ID,titulo,resumen,dificultad)

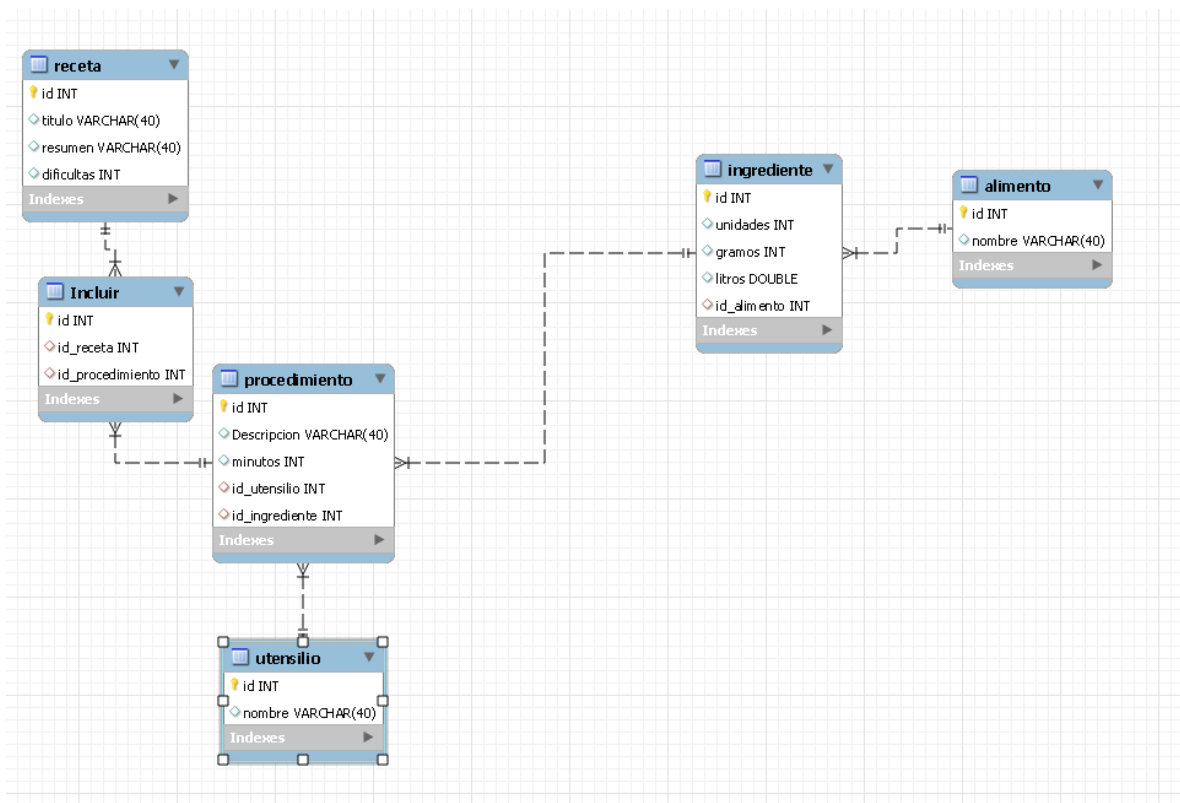
Alimento(ID, Nombre)

Ingrediente(ID, Unidades, Gramos, Litro, IDAlimento)

Utensilio(ID, Nombre)

Procedimientos(ID,orden,Descripción,minutos, IDUtensilio, IDIngrediente)

Incluir(ID,id_receta,id_procedimientos)



```
create database receta;
```

```
use receta;
```

```
create table receta(id int auto_increment,titulo varchar(40),resumen varchar(40), dificultad  
int, PRIMARY KEY(id) );
```

```
create table alimento(id int auto_increment,nombre varchar(40), PRIMARY KEY(id) );
```

```
create table ingrediente(id int auto_increment,unidades int,gramos int,litros  
double,id_alimento int, PRIMARY KEY(id) ,  
foreign key (id_alimento) references alimento(id));
```

```
create table utensilio(id int auto_increment,nombre varchar(40), primary key(id));
```

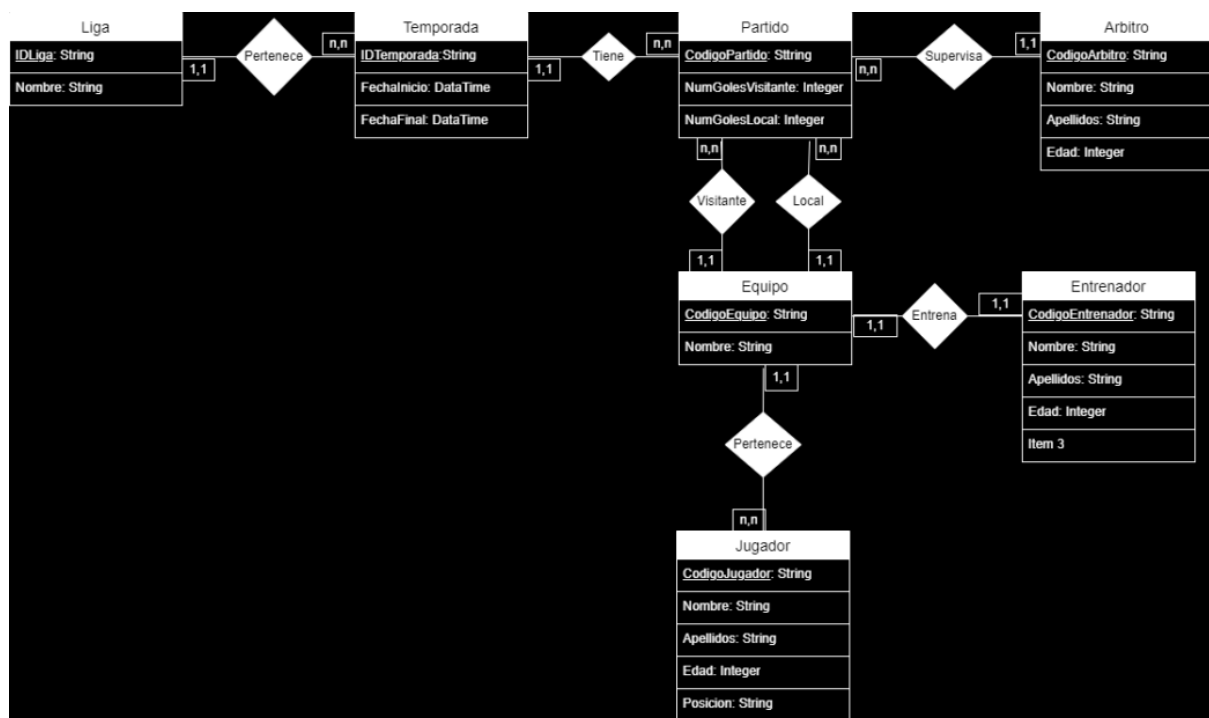
```
create table procedimiento(id int auto_increment,Descripcion varchar(40),minutos  
int,id_utensilio int, id_ingrediente int, primary key(id),  
foreign key (id_utensilio) references utensilio(id),  
foreign key (id_ingrediente) references ingrediente(id)  
);
```

```

create table Incluir(id int auto_increment,id_receta int, id_procedimiento int, PRIMARY
KEY(id) ,
foreign key (id_receta) references receta(id),
foreign key (id_procedimiento) references procedimiento(id)
);

```

10.-



Liga(id_liga, nombre)

Arbitro(codigo_arbitro, nombre, apellido, edad)

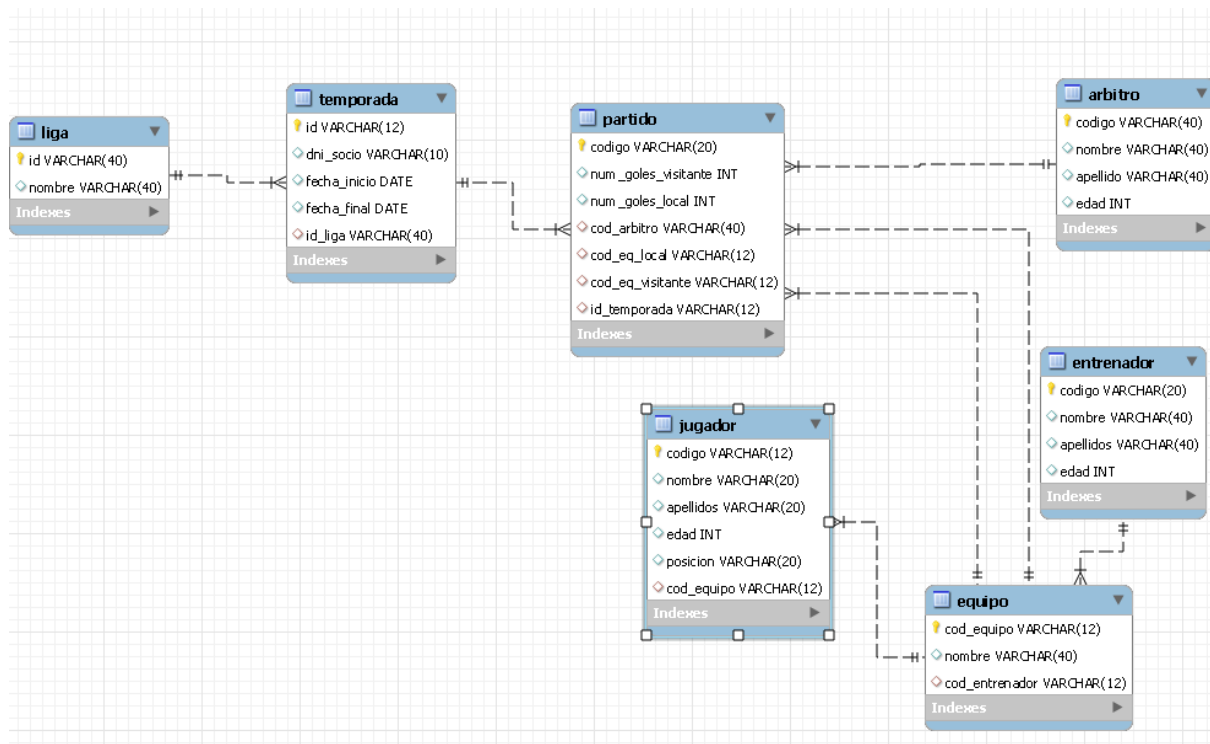
Temporada(id_temporada, fecha_inicio, fecha_final, id_liga)

Entrenador(codigo_entrenador, nombre, apellidos, edad)

Equipo(codigo_equipo, nombre, codigo_entrenador)

Partido(codigo_partido, num_goles_vistate, num_goles_local, codigo_arbitro, codigo_equipo_local, codigo_equipo_visitante, id_temporada)

Jugador(codigo_jugador, nombre, apellidos, edad, posicion, cod_equipo)



```
create database futbol;
use futbol;
```

```
create table liga(id varchar(40), nombre varchar(40), PRIMARY KEY(id) );
```

```
create table arbitro(codigo varchar(40), nombre varchar(40), apellido varchar(40), edad int,
PRIMARY KEY(codigo) );
```

```
create table temporada(id varchar(12), dni_socio varchar(10), fecha_inicio date, fecha_final
date, id_liga varchar(40), foreign key (id_liga ) references liga(id) on delete cascade on
update cascade, PRIMARY KEY(id) );
```

```
create table entrenador(codigo varchar(20), nombre varchar(40), apellidos varchar(40),
edad int, PRIMARY KEY(codigo) );
```

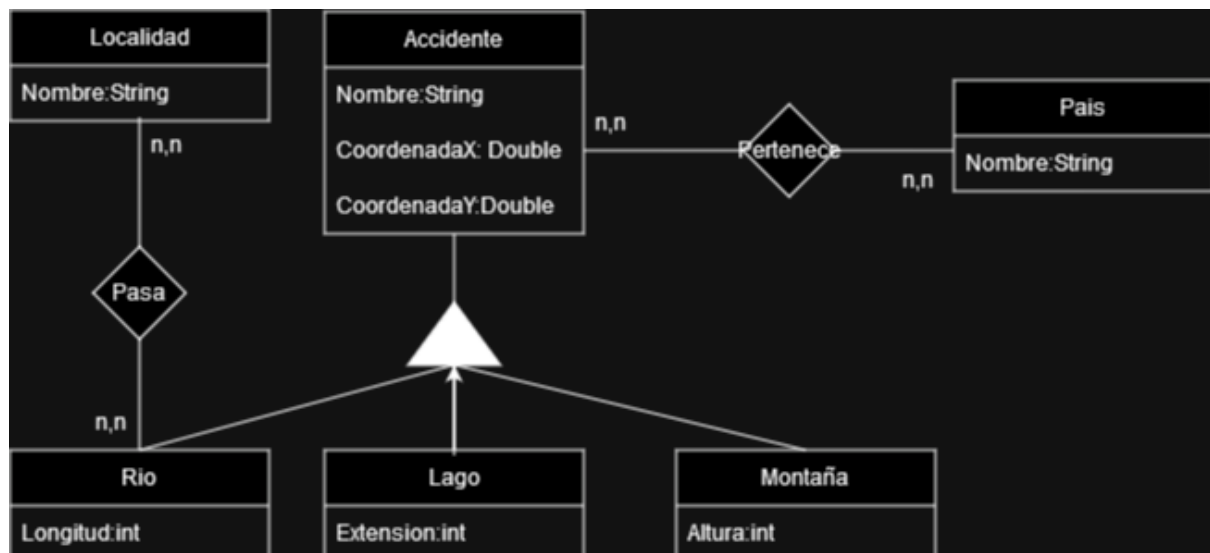
```
create table equipo(cod_equipo varchar(12), nombre varchar(40), cod_entrenador
varchar(12), foreign key (cod_entrenador ) references entrenador(codigo) , PRIMARY
KEY(cod_equipo) );
```

```
create table partido(codigo varchar(20), num_goles_visitante int, num_goles_local int,
cod_arbitro varchar(40), cod_eq_local varchar(12), cod_eq_visitante varchar(12),
id_temporada varchar(12),
foreign key (cod_arbitro) references arbitro(codigo) on delete cascade on update cascade,
foreign key (cod_eq_local) references equipo(cod_equipo) ,
foreign key (cod_eq_visitante) references equipo(cod_equipo) ,
```

foreign key (id_temporada) references temporada(id) on delete cascade on update cascade, PRIMARY KEY(codigo));

create table jugador(codigo varchar(12), nombre varchar(20), apellidos varchar(20), edad int, posicion varchar(20), cod_equipo varchar(12),
foreign key (cod_equipo) references equipo(cod_equipo) on delete cascade on update cascade, PRIMARY KEY(codigo));

11.-



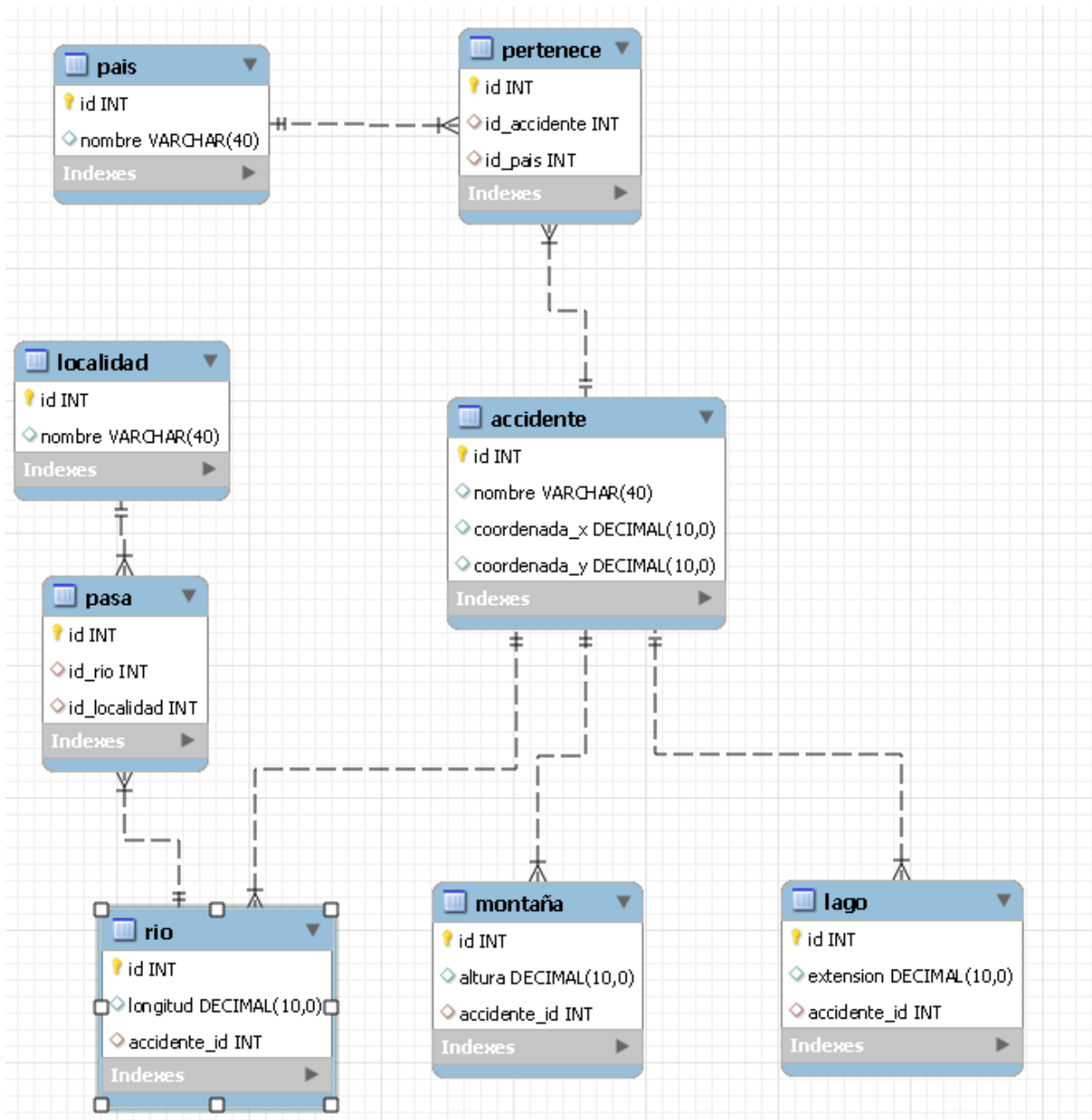
Localidad(id,Nombre)

Pais(id,Nombre)

Accidente(id,Nombre,coordenada_x,coordenada_y)

Rio(id,Longitud,accidente_id)

Lago(id,Extension,accidente_id)
 Montaña(id,Altura,accidente_id)
 Pasa(id,id_rio,id_localidad)
 Pertenece(id,id_accidente,id_pais)



```
create database accidentes;
use accidentes;
```

```
create table localidad (id int auto_increment , nombre varchar(40),primary key(id));
```

```
create table pais (id int auto_increment, nombre varchar(40),primary key(id));
```

```
create table accidente (id int auto_increment , nombre varchar(40), coordenada_x double,
coordenada_y double,primary key(id));
```



```
create table rio (id int auto_increment, longitud decimal, accidente_id int, primary key(id),  
foreign key (accidente_id) references accidente(id));
```

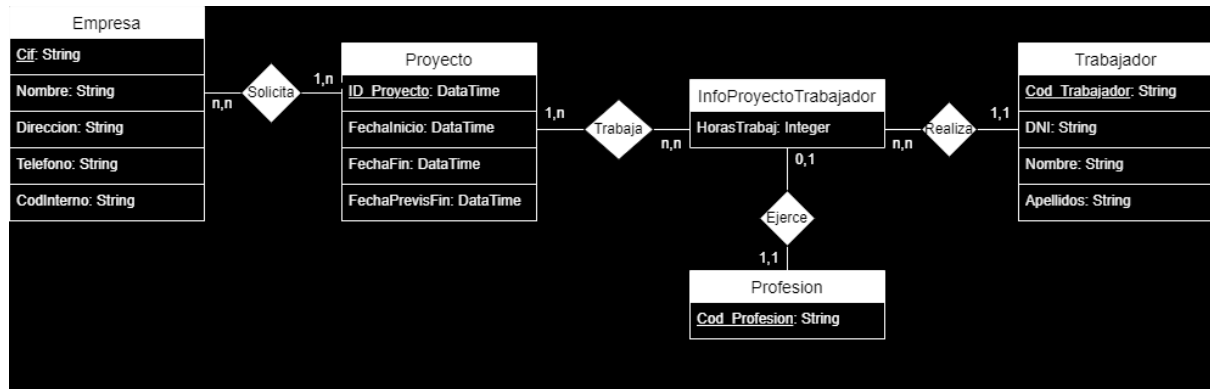
```
create table lago (id int auto_increment, extension decimal, accidente_id int, primary key(id),  
foreign key (accidente_id) references accidente(id));
```

```
create table montaña (id int auto_increment , altura decimal, accidente_id int, primary key(id),  
foreign key (accidente_id) references accidente(id));
```

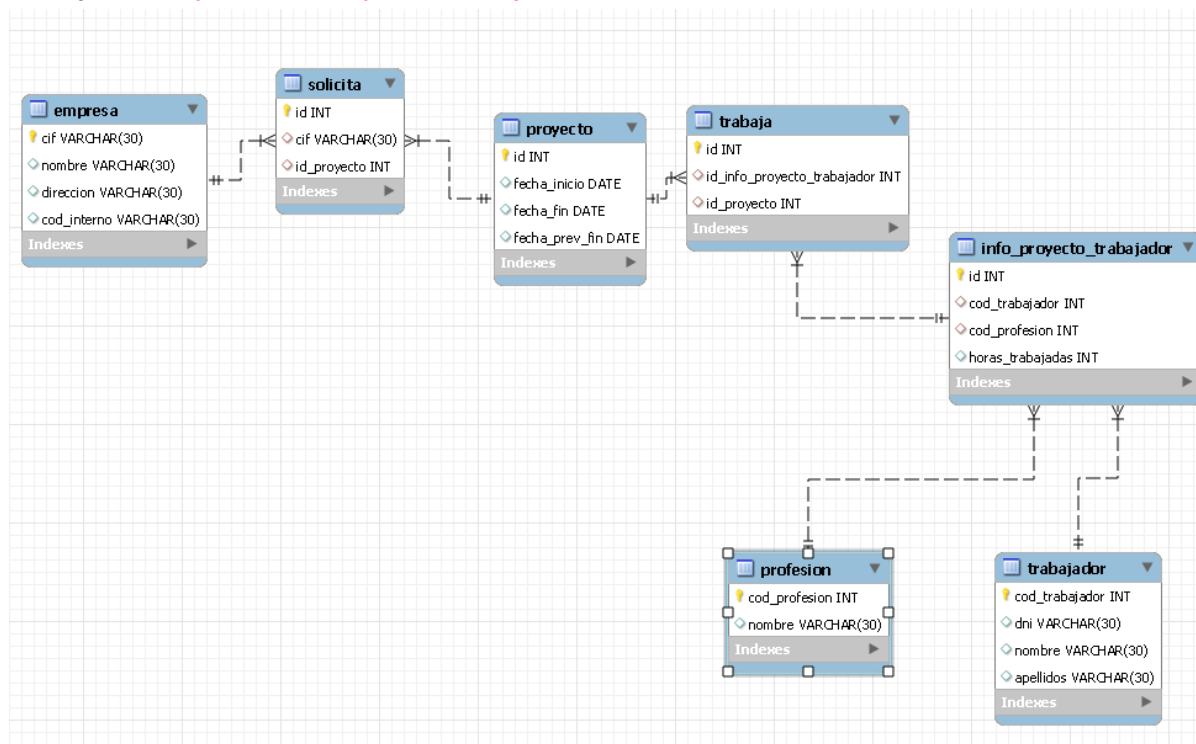
```
create table pasa (id int auto_increment, id_rio int, id_localidad int, primary key(id),  
foreign key (id_rio) references rio(id),  
foreign key (id_localidad) references localidad(id));
```

```
create table pertenece (id int auto_increment , id_accidente int, id_pais int, primary key(id),  
foreign key (id_accidente) references accidente(id),  
foreign key (id_pais) references pais(id));
```

12.-



Empresa(CIF, nombre, dirección, telefono, codigo_interno)
 Proyecto(id_proyecto, fecha_inicio, fecha_fin, fecha_prev_fin)
 Solicita(cif_empresa, id_proyecto)
 Profesion(Cod_profesion, nombre)
 Trabajador(cod_trabajador, dni, nombre, apellidos)
 InfoProyectoTrabajador(id_proyecto_trabajador, cod_trabajador, cod_profesion, horas_trabajadas)
 Trabaja(id_proyecto, id_proyecto_trabajador)



```

create database empresa_software;
use empresa_software;
    
```

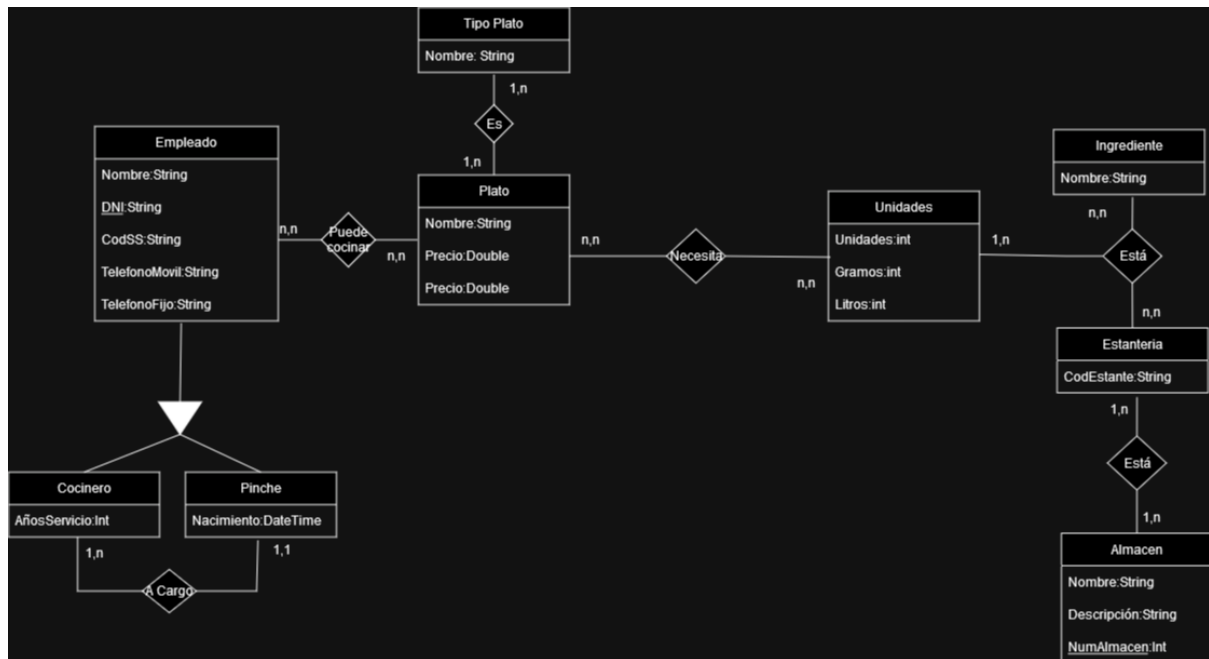
```

create table empresa(cif varchar(30),nombre varchar(30),direccion varchar(30),cod_interno
varchar(30), PRIMARY KEY(cif) );
create table proyecto(id int auto_increment,fecha_inicio date,fecha_fin date, fecha_prev_fin
date, PRIMARY KEY(id));
create table solicita(id int auto_increment,cif varchar(30), id_proyecto int, PRIMARY
KEY(id),
foreign key (id_proyecto) references proyecto(id) on delete cascade on update cascade,
foreign key (cif) references empresa(cif) on delete cascade on update cascade
);
create table profesion(cod_profesion int auto_increment,nombre varchar(30), primary
key(cod_profesion) );
create table trabajador(cod_trabajador int,dni varchar(30),nombre varchar(30),apellidos
varchar(30), primary key(cod_trabajador));

create table info_proyecto_trabajador(id int auto_increment, cod_trabajador int,
cod_profesion int, horas_trabajadas int, primary key(id),
foreign key (cod_trabajador) references trabajador(cod_trabajador) on delete cascade on
update cascade,
foreign key (cod_profesion) references profesion(cod_profesion) on delete cascade on
update cascade
);
create table trabaja(id int auto_increment, id_info_proyecto_trabajador int,id_proyecto int,
primary key(id),
foreign key (id_info_proyecto_trabajador) references info_proyecto_trabajador(id) on delete
cascade on update cascade,
foreign key (id_proyecto) references proyecto(id) on delete cascade on update cascade
);

```

13.-



Empleado(dni, nombre, codss, tef_movil, telf_fijo)

Pinche(dni, fecha_nacimiento)

Cocinero(dni, años_servicio, dni_pinche)

TipoPlato(id_tipo, nombre)

Plato(nombre, precio, descripcion)

Es(nomb_plato, id_tipo)

Puede_cocinar(dni, nomb_plato)

Unidades(id_unidad, unidades, gramos, litros)

Necesita(id_unidad, nomb_plato)

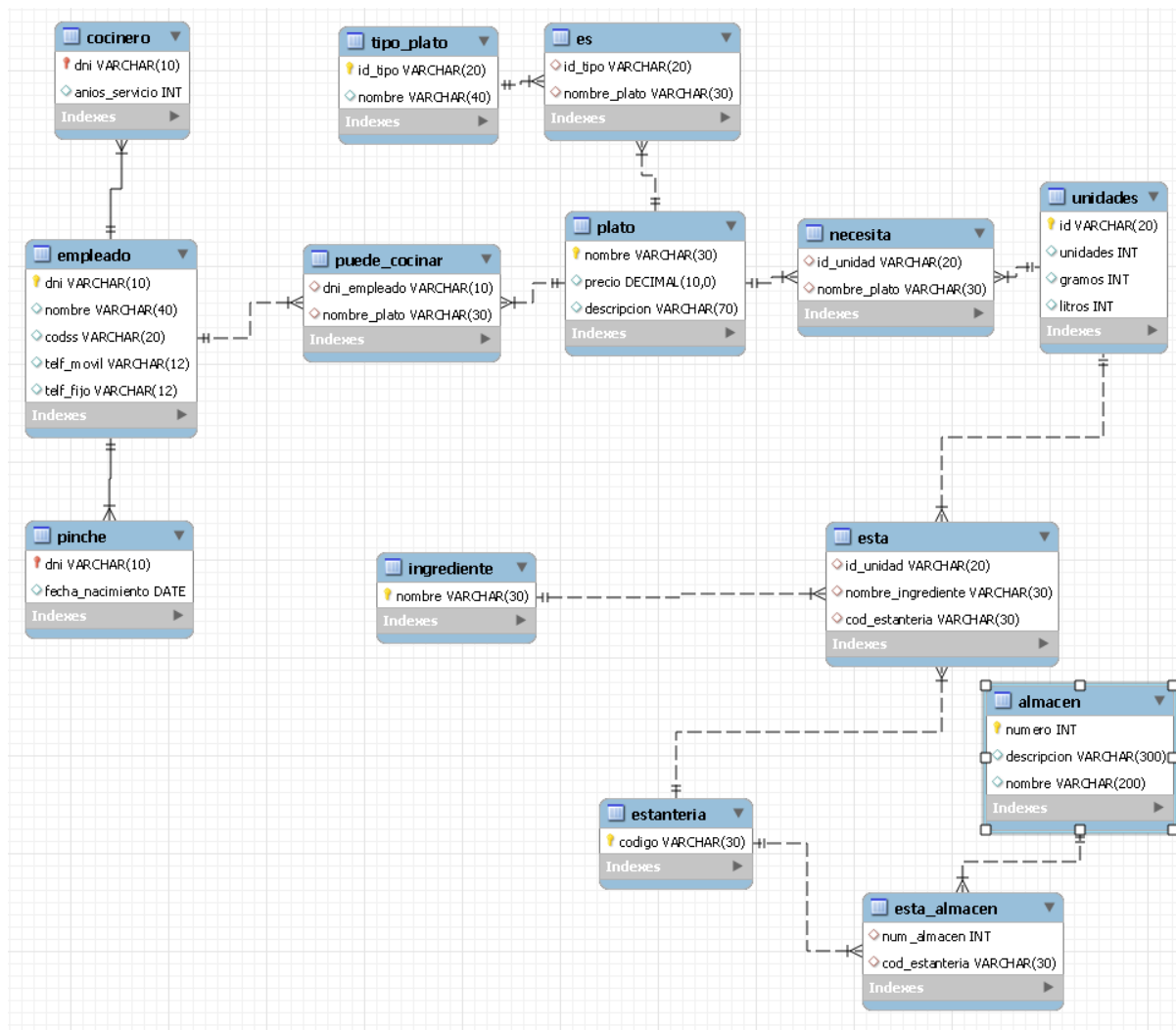
Ingrediente(nombre_ing)

Estanteria(cod_estant)

Almacen(num_almacen, nombre, descripcion)

Esta(id_unidad, nombre_ing, cod_estant)

Esta_Almacen(cod_estant, num_almacen)



create database empresa_comida;

use empresa_comida;

create table empleado(dni varchar(10), nombre varchar(40), codss varchar(20), UNIQUE (codss), telf_movil varchar(12), UNIQUE (telf_movil),telf_fijo varchar(12), UNIQUE (telf_fijo), PRIMARY KEY(dni));

create table pinche(dni varchar(10), fecha_nacimiento date, foreign key (dni) references empleado(dni) on delete cascade on update cascade, PRIMARY KEY(dni));

create table cocinero(dni varchar(10), anios_servicio int, foreign key (dni) references empleado(dni) on delete cascade on update cascade, PRIMARY KEY(dni));

create table tipo_plato(id_tipo varchar(20), nombre varchar(40), PRIMARY KEY(id_tipo));

create table plato(nombre varchar(30), precio decimal, descripcion varchar(70), PRIMARY KEY(nombre));

```
create table es(id_tipo varchar(20), nombre_plato varchar(30), foreign key (id_tipo)
references tipo_plato(id_tipo) on delete cascade on update cascade,
foreign key (nombre_plato ) references plato(nombre) on delete cascade on update
cascade);
```

```
create table puede_cocinar(dni_empleado varchar(10), nombre_plato varchar(30), foreign
key (dni_empleado) references empleado(dni) on delete cascade on update cascade,
foreign key (nombre_plato) references plato(nombre) on delete cascade on update
cascade);
```

```
create table unidades(id varchar(20), unidades int, gramos int, litros int, PRIMARY KEY(id) );
```

```
create table necesita(id_unidad varchar(20), nombre_plato varchar(30), foreign key
(id_unidad) references unidades(id) on delete cascade on update cascade, foreign key
(nombre_plato) references plato(nombre) on delete cascade on update cascade);
```

```
create table ingrediente(nombre varchar(30), PRIMARY KEY(nombre) );
```

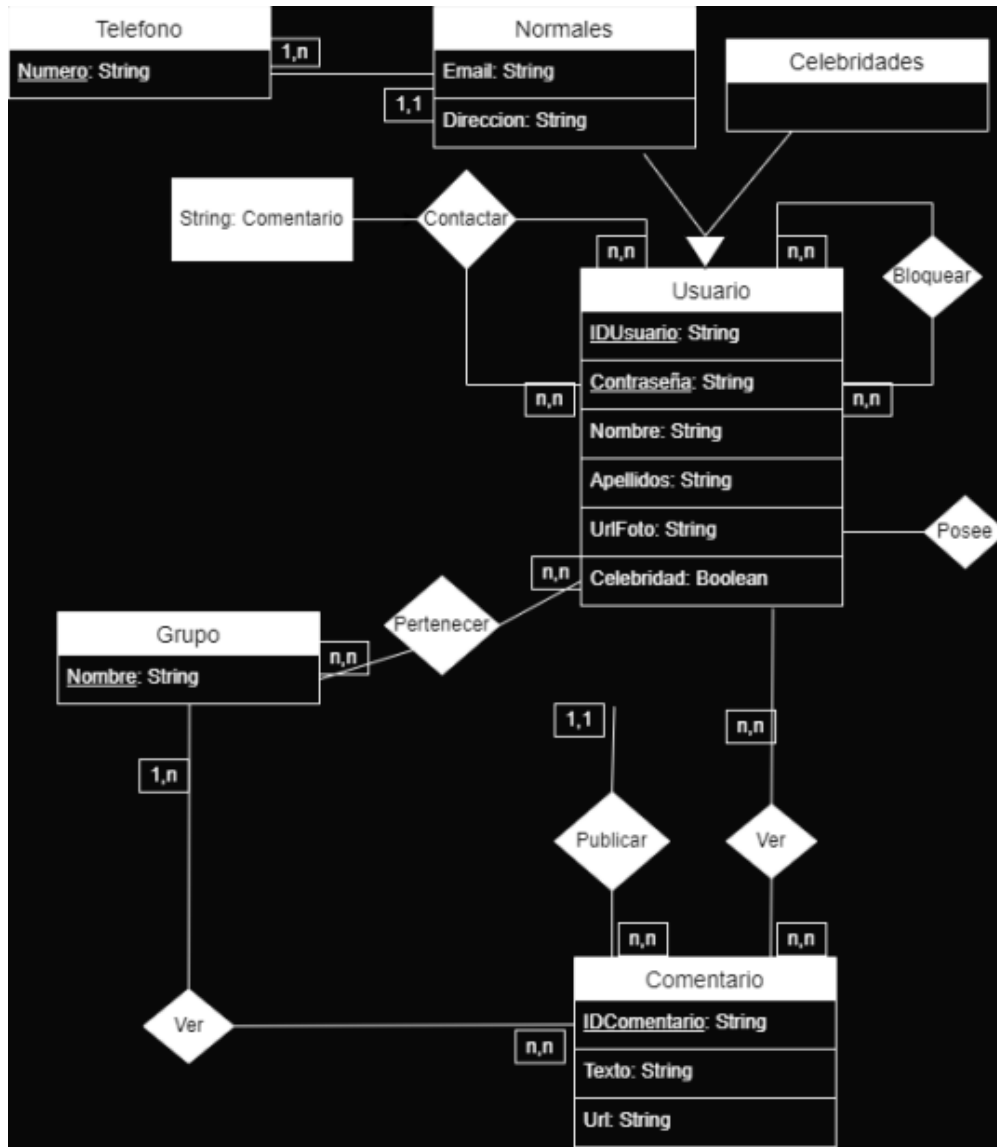
```
create table estanteria(codigo varchar(30), PRIMARY KEY(codigo) );
```

```
create table almacen(numero int auto_increment, descripcion varchar(300), nombre
varchar(200), PRIMARY KEY(numero) );
```

```
create table esta(id_unidad varchar(20), nombre_ingrediente varchar(30), cod_estanteria
varchar(30),
foreign key (id_unidad) references unidades(id) on delete cascade on update cascade,
foreign key (nombre_ingrediente ) references ingrediente(nombre) on delete cascade on
update cascade,
foreign key (cod_estanteria ) references estanteria(codigo) on delete cascade on update
cascade);
```

```
create table esta_almacen(num_almacen int, cod_estanteria varchar(30), foreign key
(num_almacen) references almacen(numero) on delete cascade on update cascade,
foreign key (cod_estanteria) references estanteria(codigo) on delete cascade on update
cascade);
```

14.-



Usuario(ID, Contraseña, Nombre, Apellido, UrlFoto, Celebridad)

Contactar(ID, id_usuario, id_usuario, comentario)

```
Bloquear(ID,id_usuario,id_usuario)
```

Grupo(ID,Nombre)

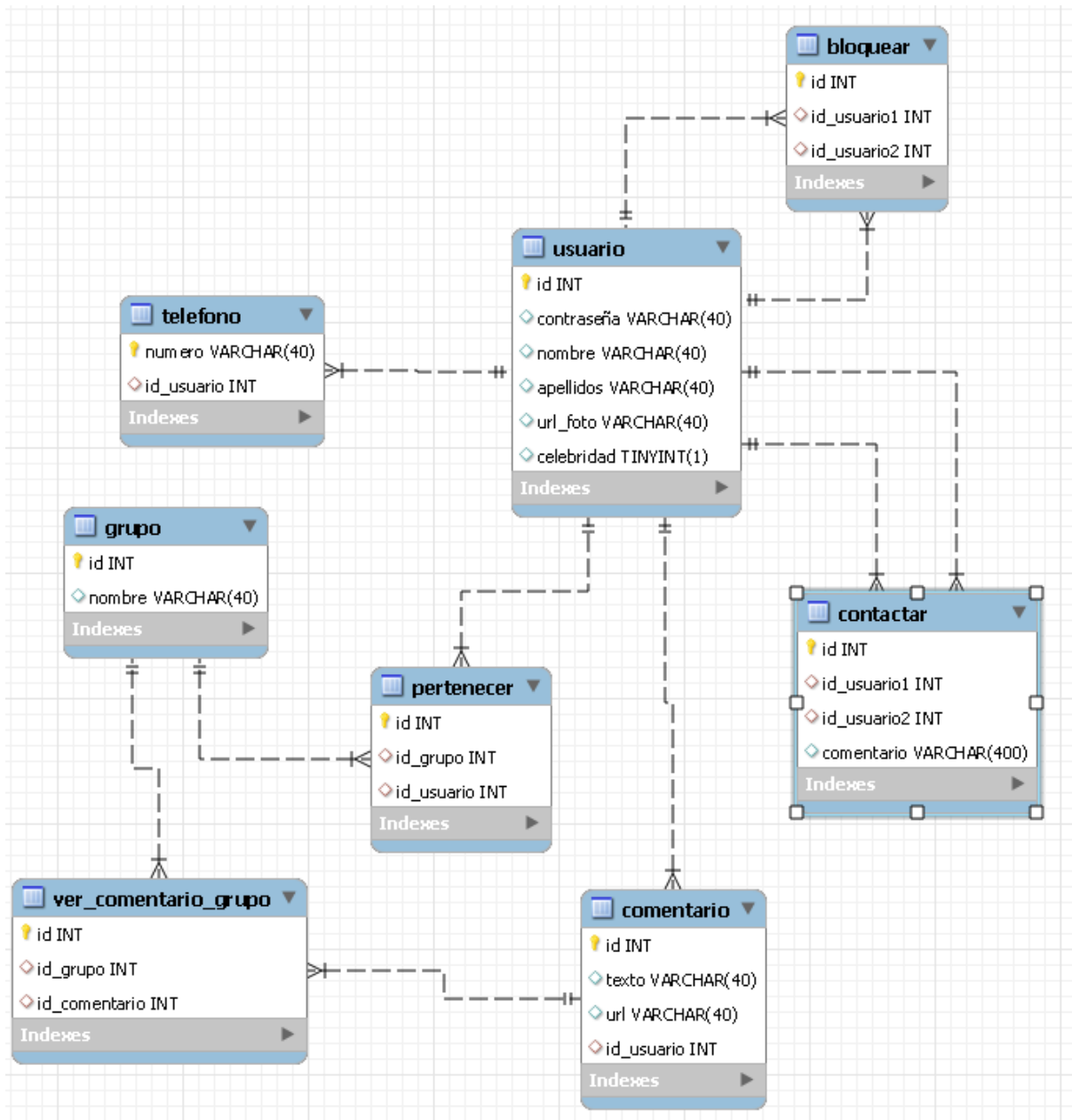
Pertenecer(ID, id_usuario, id_grupo)

Comentario(ID,texto,url,id_usuario)

VerComentario(ID, id_usuario, id_comentario)

```
VerComentarioGrupo(id,id_grupo,id_comentario)
```

Teléfono(Numero, id_usuario)



```

create database red_social;
use red_social;

```

```

create table usuario(id int auto_increment,contraseña varchar(40),nombre
varchar(40),apellidos varchar(40),url_foto varchar(40),celebridad boolean, PRIMARY
KEY(id) );

```



```
create table contactar(id int auto_increment,id_usuario1 int, id_usuario2 int,comentario
varchar(400), PRIMARY KEY(id) ,
foreign key (id_usuario1) references usuario(id),
foreign key (id_usuario2) references usuario(id)
);
```

```
create table bloquear(id int auto_increment,id_usuario1 int, id_usuario2 int, PRIMARY
KEY(id) ,
foreign key (id_usuario1) references usuario(id),
foreign key (id_usuario2) references usuario(id)
);
```

```
create table grupo(id int auto_increment,nombre varchar(40), PRIMARY KEY(id));
```

```
create table pertenecer(id int auto_increment,id_grupo int, id_usuario int, PRIMARY KEY(id)
,
foreign key (id_usuario) references usuario(id),
foreign key (id_grupo) references grupo(id)
);
```

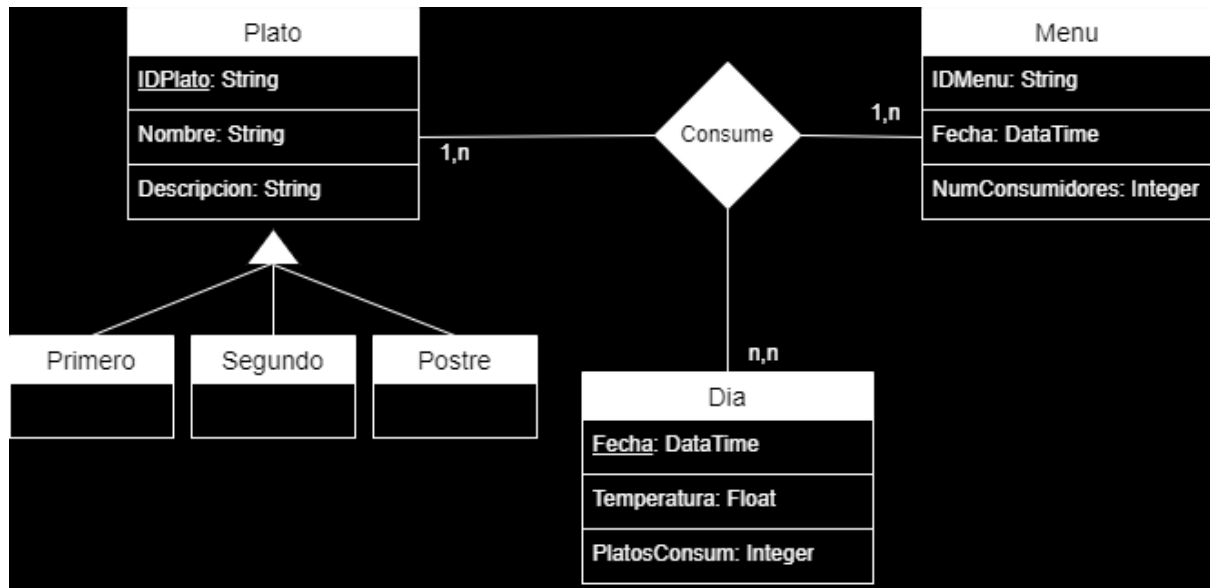
```
create table comentario(id int auto_increment,texto varchar(40),url varchar(40) ,id_usuario
int, PRIMARY KEY(id) ,
foreign key (id_usuario) references usuario(id)
);
```

```
create table ver_comentario(id int auto_increment,id_usuario int, id_comentario int,
PRIMARY KEY(id) ,
foreign key (id_usuario) references usuario(id),
foreign key (id_comentario) references comentario(id)
);
```

```
create table ver_comentario_grupo(id int auto_increment,id_grupo int, id_comentario int,
PRIMARY KEY(id) ,
foreign key (id_grupo) references grupo(id),
foreign key (id_comentario) references comentario(id)
);
```

```
create table telefono(numero varchar(40) ,id_usuario int, PRIMARY KEY(numero) ,
foreign key (id_usuario) references usuario(id)
);
```

15.-



Plato(id, nombre, descripcion)

Primero(id, id_plato)

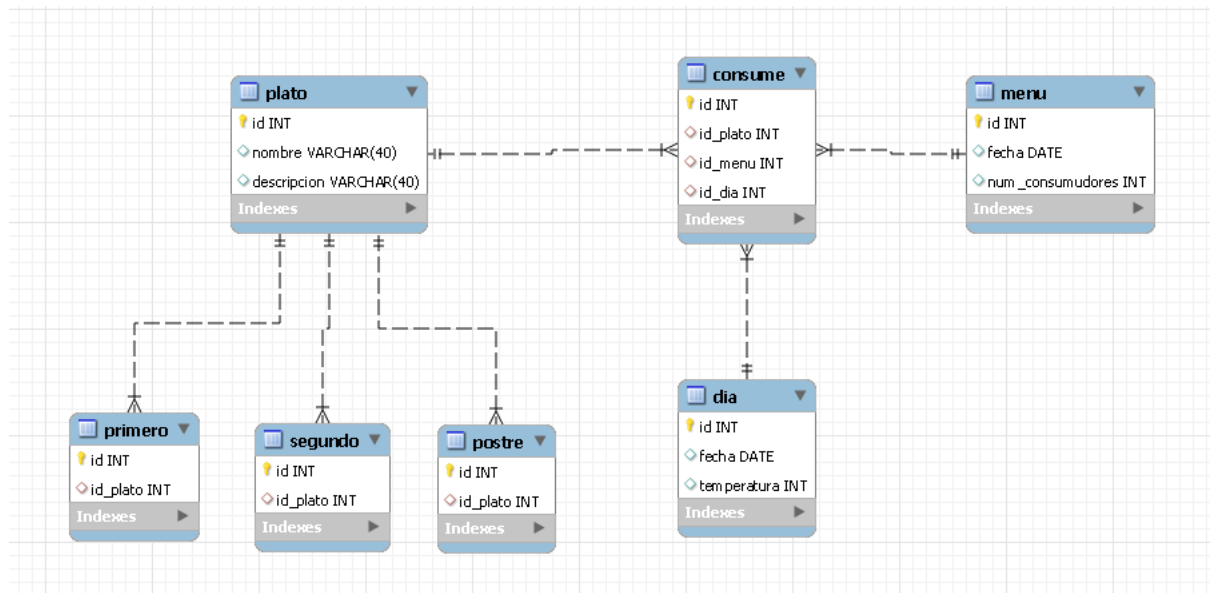
Segundo(id, id_plato)

Postre(id, id_plato)

Menu(id, fecha, num-consumidores)

Dia(id, fecha, temperatura, plato_consum)

Consume(id, id_plato, id_menu, id_dia)



create database platos;

use platos;

create table plato (id int auto_increment , nombre varchar(40),descripcion
varchar(40),primary key(id));

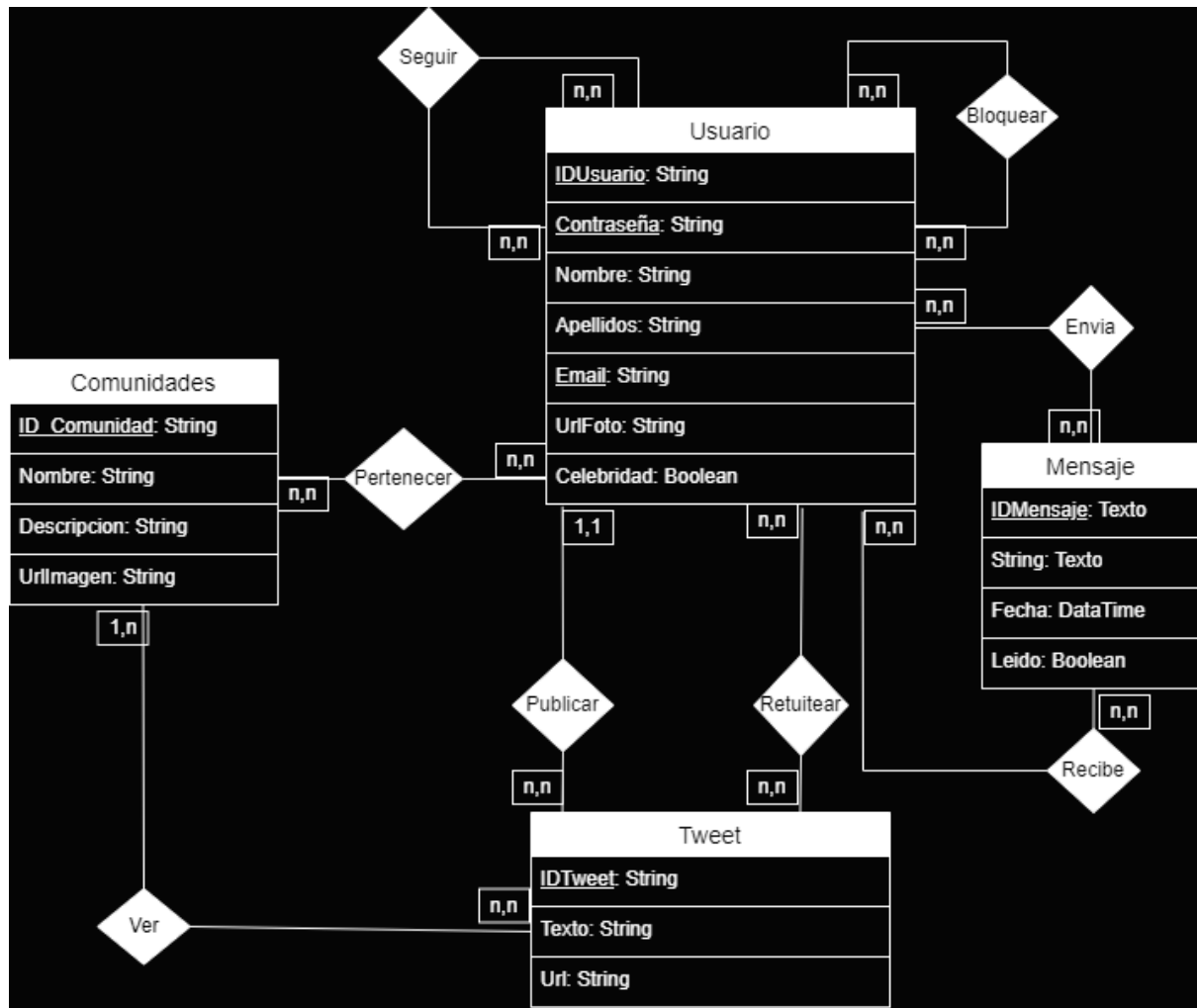
create table primero(id int auto_increment, id_plato int, primary key(id),

```
foreign key (id_plato) references plato(id)
);
create table segundo(id int auto_increment, id_plato int, primary key(id),
foreign key (id_plato) references plato(id)
);
create table postre(id int auto_increment, id_plato int, primary key(id),
foreign key (id_plato) references plato(id)
);
create table menu (id int auto_increment , fecha date,num_consumidores int,primary
key(id));

create table dia (id int auto_increment , fecha date,temperatura int, primary key(id));

create table consume(id int auto_increment, id_plato int, id_menu int, id_dia int, primary
key(id),
foreign key (id_plato) references plato(id),
foreign key (id_menu) references menu(id),
foreign key (id_dia) references dia(id)
);
```

16.-



Usuario(id,contraseña,nombre,apellidos,email,url_foto,celebridad)

Seguir(id, id_usuario,id_usuario)

Bloquear(id, id_usuario, id_usuario)

Tweet(id,texto,url, id_usuario)

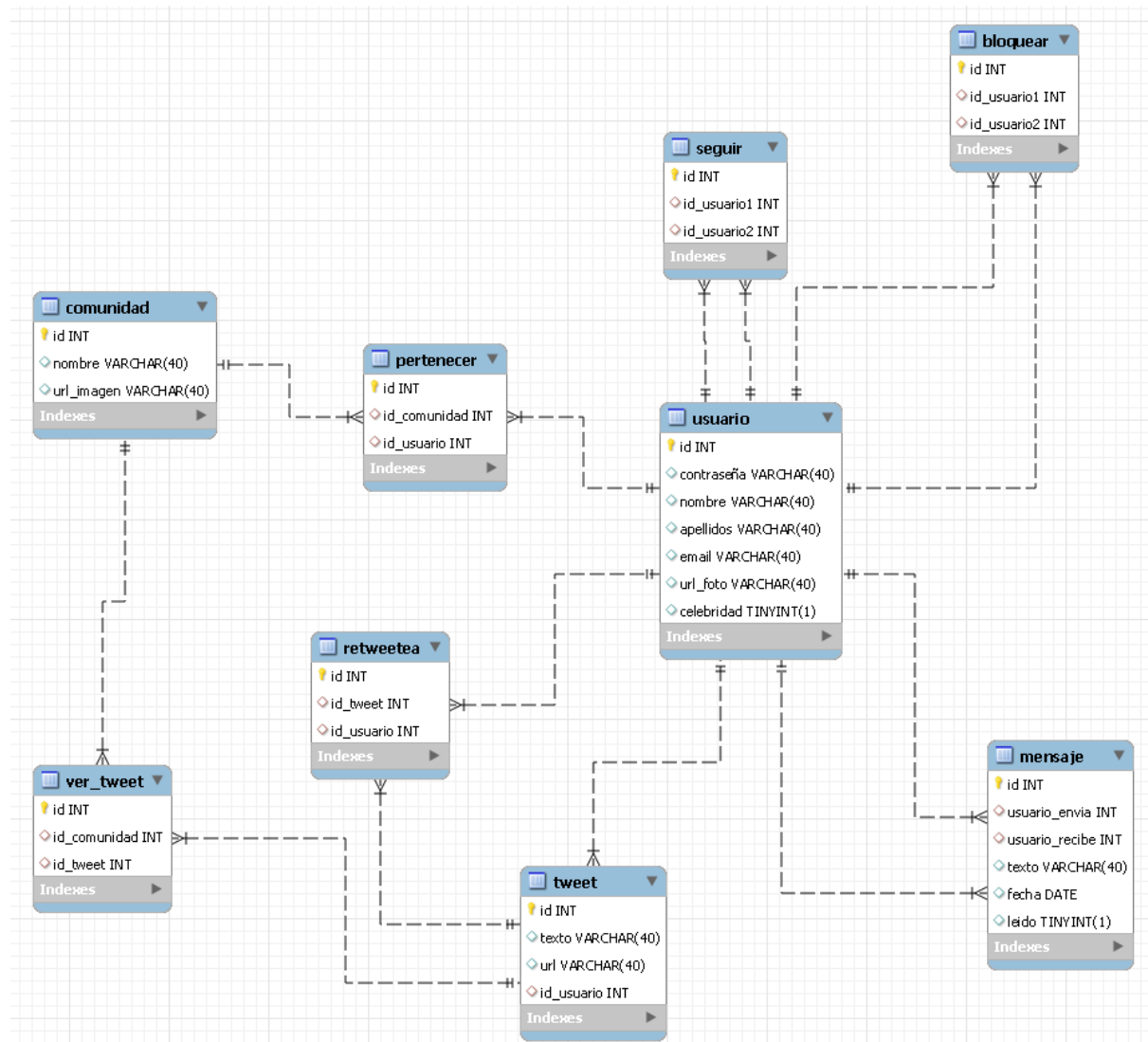
Retuitea(id,id_tweet,id_usuario)

Comunidad(id,nombre,descripcion,urlimagen)

Pertenece(id_usuario,id_Comunidad)

VerTweet(id, id_comunidad, id_tweet)

Mensaje(id,id_usuario_envia,id_usuario_recibe, Texto,fecha,leido)



create database twitter;
use twitter;

create table usuario(id int auto_increment,contraseña varchar(40),nombre varchar(40),apellidos varchar(40),email varchar(40),url_foto varchar(40),celebridad boolean, PRIMARY KEY(id));

create table bloquear(id int auto_increment,id_usuario1 int, id_usuario2 int, PRIMARY KEY(id) ,
foreign key (id_usuario1) references usuario(id),
foreign key (id_usuario2) references usuario(id)
);

create table seguir(id int auto_increment,id_usuario1 int, id_usuario2 int, PRIMARY KEY(id)
,
foreign key (id_usuario1) references usuario(id),
foreign key (id_usuario2) references usuario(id)
);

```
create table tweet(id int auto_increment,texto varchar(40),url varchar(40) ,id_usuario int,
PRIMARY KEY(id) ,
foreign key (id_usuario) references usuario(id)
);
```

```
create table retweetea(id int auto_increment,id_tweet int,id_usuario int, PRIMARY KEY(id) ,
foreign key (id_usuario) references usuario(id),
foreign key (id_tweet) references tweet(id)
);
```

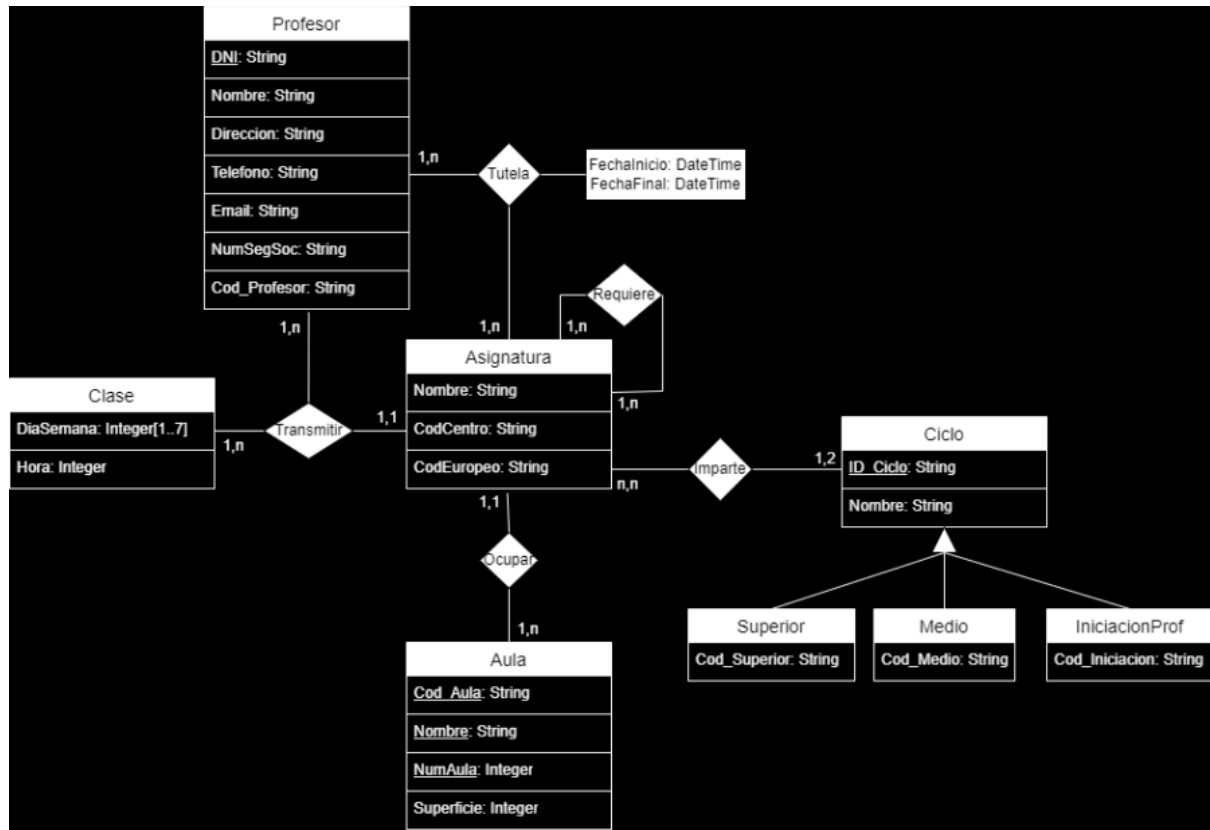
```
create table comunidad(id int auto_increment,nombre varchar(40),url_imagen varchar(40),
PRIMARY KEY(id) );
```

```
create table pertenecer(id int auto_increment,id_comunidad int,id_usuario int, PRIMARY
KEY(id) ,
foreign key (id_usuario) references usuario(id),
foreign key (id_comunidad) references comunidad(id)
);
```

```
create table ver_tweet(id int auto_increment,id_comunidad int,id_tweet int, PRIMARY
KEY(id) ,
foreign key (id_tweet) references tweet(id),
foreign key (id_comunidad) references comunidad(id)
);
```

```
create table mensaje(id int auto_increment,usuario_envia int, usuario_recibe int,texto
varchar(40),fecha date,leido boolean, PRIMARY KEY(id) ,
foreign key (usuario_envia) references usuario(id),
foreign key (usuario_recibe) references usuario(id)
);
```

17.-



Profesor(dni, nombre, direccion, telefono, email, num_seg_social, cod_profesor)

Clase(id, dia_semana, hora)

Aula(cod_aula, nombre, num_aula, superficie)

Ciclo(id_ciclo, nombre, cod_centro)

Asignatura(nombre, cod_centro, cod_europeo, **cod_aula**, **id_clase**, **id_ciclo**)

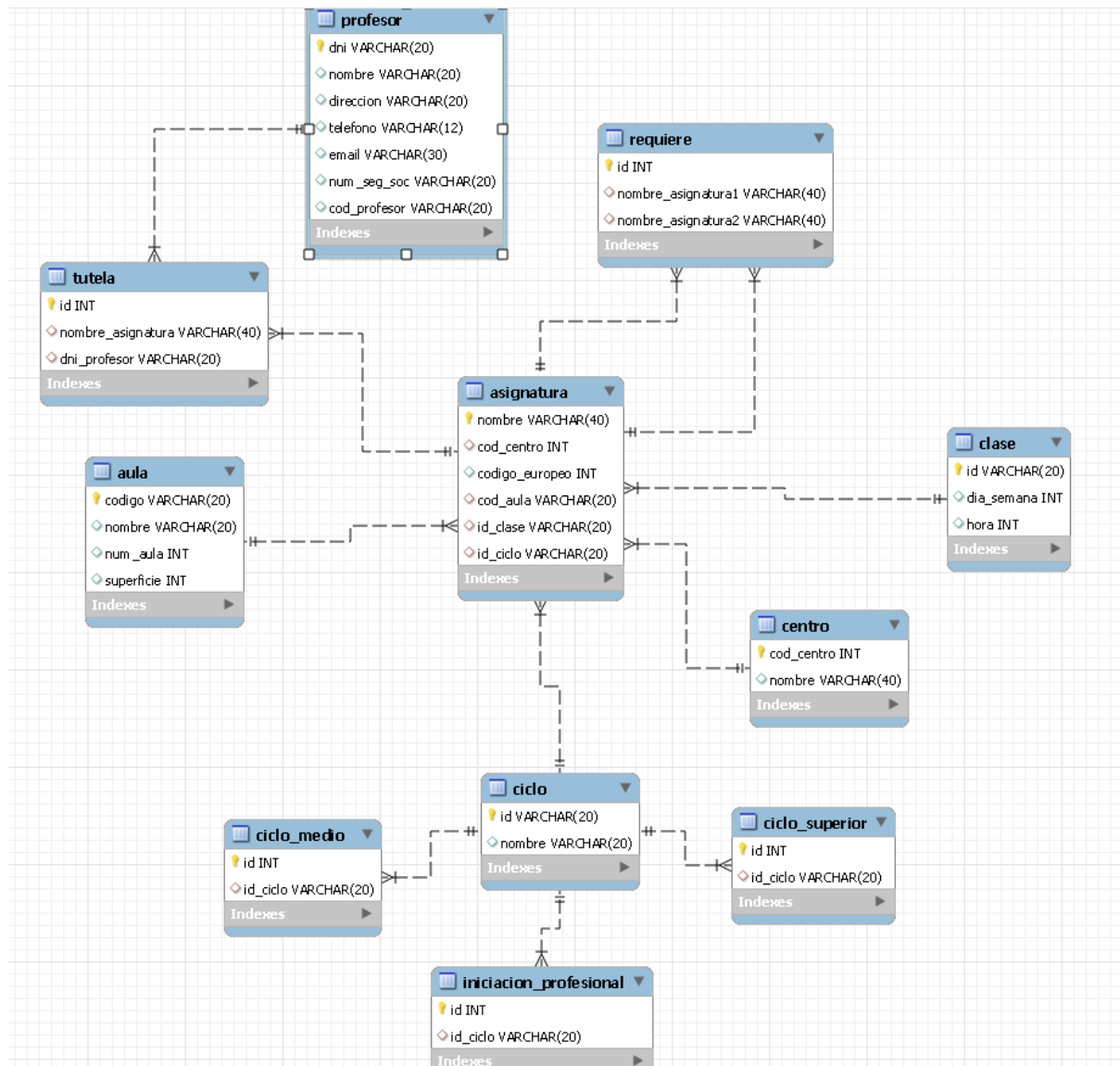
Requiere(id, **id_asignatura**, **id_asignatura**)

Tutela(id, **id_profesor**, **id_asignatura**)

Superior(id, **id_ciclo**)

Medio(id, **id_ciclo**)

IniciacionProf(id, **id_ciclo**,)



create database horario_escolar;

use horario_escolar;

```
create table profesor(dni varchar(20), nombre varchar(20), direccion varchar(20), telefono
varchar(12), email varchar(30), num_seg_soc varchar(20), cod_profesor varchar(20),
UNIQUE (cod_profesor), PRIMARY KEY(dni) );
```

```
create table clase(id varchar(20), dia_semana int, hora int, PRIMARY KEY(id) );
```

```
create table aula(codigo varchar(20), nombre varchar(20), num_aula int, superficie int,
PRIMARY KEY(codigo) );
```

```
create table ciclo(id varchar(20), nombre varchar(20), PRIMARY KEY(id) );
```

```
create table centro(cod_centro int auto_increment,nombre varchar(40),primary
key(cod_centro));
```

```
create table asignatura(nombre varchar(40),cod_centro int, codigo_europeo int, cod_aula
varchar(20), id_clase varchar(20), id_ciclo varchar(20), primary key(nombre),
```



```
foreign key (cod_centro) references centro(cod_centro) on delete cascade on update cascade,  
foreign key (cod_aula) references aula(codigo) on delete cascade on update cascade,  
foreign key (id_clase) references clase(id) on delete cascade on update cascade,  
foreign key (id_ciclo) references ciclo(id) on delete cascade on update cascade  
);
```

```
create table requiere(id int auto_increment,nombre_asignatura1  
varchar(40),nombre_asignatura2 varchar(40),primary key(id),  
foreign key (nombre_asignatura1) references asignatura(nombre) on delete cascade on  
update cascade,  
foreign key (nombre_asignatura2) references asignatura(nombre) on delete cascade on  
update cascade  
);
```

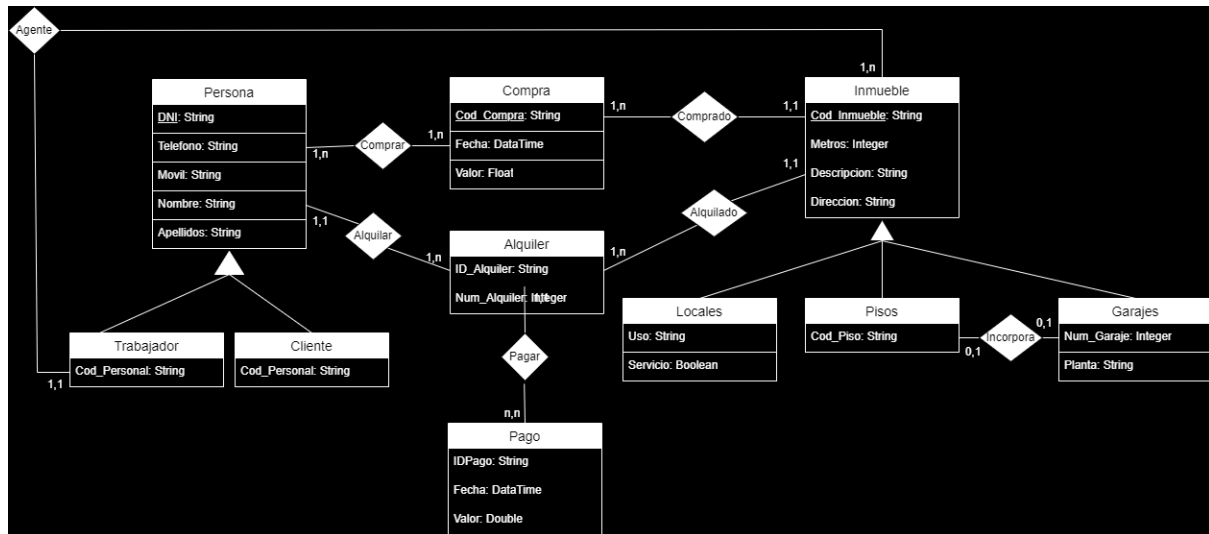
```
create table tutela(id int auto_increment,nombre_asignatura varchar(40),dni_profesor  
varchar(20),primary key(id),  
foreign key (nombre_asignatura) references asignatura(nombre) on delete cascade on  
update cascade,  
foreign key (dni_profesor) references profesor(dni) on delete cascade on update cascade  
);
```

```
create table ciclo_superior(id int auto_increment,id_ciclo varchar(20),primary key(id),  
foreign key (id_ciclo) references ciclo(id) on delete cascade on update cascade  
);
```

```
create table ciclo_medio(id int auto_increment,id_ciclo varchar(20),primary key(id),  
foreign key (id_ciclo) references ciclo(id) on delete cascade on update cascade  
);
```

```
create table iniciacion_profesional(id int auto_increment,id_ciclo varchar(20),primary key(id),  
foreign key (id_ciclo) references ciclo(id) on delete cascade on update cascade  
);
```

18.-



Persona(dni, Telefono, Movil, Nombre, apellidos)

Trabajador(cod_Personal, dni,)

Cliente(cod_Personal, dni,)

Inmueble(cod_inmueble, metros, descripcion, direccion, dni_agente)

Locales(cod_inmueble, uso, servicio)

Garajes(cod_inmueble, num_garaje, planta)

Pisos(cod_inmueble, cod_piso, num_garaje, cod_inmueble_garaj)

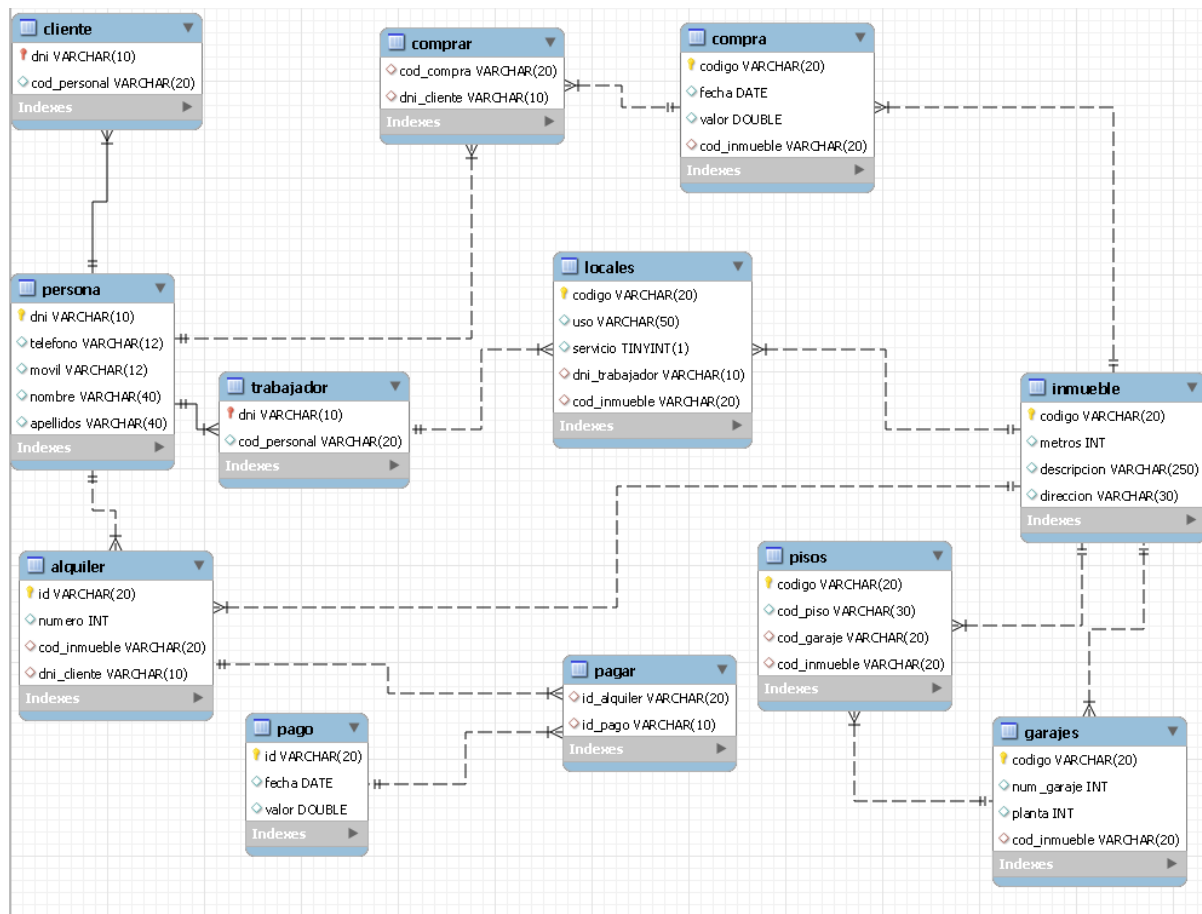
Compra(cod_compra, fecha, valor, cod_inmueble)

Alquiler(id_alquiler, num_alquiler, cod_inmueble, dni)

Comprar(cod_compra, dni)

Pago(id_pago, fecha, valor)

Pagar(id_alquiler, id_pago)



create database inmuebles;
use inmuebles;

create table persona(dni varchar(10), telefono varchar(12), movil varchar(12), nombre varchar(40), apellidos varchar(40), PRIMARY KEY(dni));

create table trabajador(dni varchar(10), cod_personal varchar(20), UNIQUE (cod_personal), foreign key (dni) references persona(dni) on delete cascade on update cascade, PRIMARY KEY(dni));

create table cliente(dni varchar(10), cod_personal varchar(20), UNIQUE (cod_personal), foreign key (dni) references persona(dni) on delete cascade on update cascade, PRIMARY KEY(dni));

create table inmueble(codigo varchar(20), metros int, descripcion varchar(250), direccion varchar(30), PRIMARY KEY(codigo));

create table locales(codigo varchar(20), uso varchar(50), servicio boolean, dni_trabajador varchar(10), cod_inmueble varchar(20), foreign key (cod_inmueble) references inmueble(codigo) on delete cascade on update cascade,

```
foreign key (dni_trabajador ) references trabajador(dni) on delete cascade on update cascade,  
PRIMARY KEY(codigo));
```

```
create table garajes(codigo varchar(20), num_garaje int, planta int, cod_inmueble  
varchar(20), foreign key (cod_inmueble ) references inmueble(codigo) on delete cascade on  
update cascade, PRIMARY KEY(codigo));
```

```
create table pisos(codigo varchar(20), cod_piso varchar(30), cod_garaje varchar(20),  
cod_inmueble varchar(20), foreign key (cod_garaje ) references garajes(codigo) on delete  
cascade on update cascade, foreign key (cod_inmueble ) references inmueble(codigo) on  
delete cascade on update cascade, PRIMARY KEY(codigo));
```

```
create table compra(codigo varchar(20), fecha date, valor double, cod_inmueble  
varchar(20), foreign key (cod_inmueble ) references inmueble(codigo) on delete cascade on  
update cascade, PRIMARY KEY(codigo) );
```

```
create table alquiler(id varchar(20), numero int, cod_inmueble varchar(20), dni_cliente  
varchar(10),  
foreign key (cod_inmueble) references inmueble(codigo) on delete cascade on update  
cascade,  
foreign key (dni_cliente ) references persona(dni) on delete cascade on update cascade,  
PRIMARY KEY(id) );
```

```
create table comprar(cod_compra varchar(20), dni_cliente varchar(10), foreign key  
(cod_compra ) references compra(codigo) on delete cascade on update cascade, foreign  
key (dni_cliente) references persona(dni) on delete cascade on update cascade);
```

```
create table pago(id varchar(20), fecha date, valor double, PRIMARY KEY(id) );
```

```
create table pagar(id_alquiler varchar(20), id_pago varchar(10), foreign key (id_alquiler)  
references alquiler(id) on delete cascade on update cascade, foreign key (id_pago)  
references pago(id) on delete cascade on update cascade);
```