**Base de datos propuesta 3er parcial:**

**Jesse Isaac Guerrero Jiménez**

**Luis Mario Maurio Jiménez**

**Omar Reyes Morales**

**Universidad Politécnica de Tulancingo, Calle Ingenierías #100, Huapalcalco,**

**Tulancingo**

**Asignatura: Base de datos**

**Nombre del profesor: Víctor Hugo Fernández Cruz**

**Tulancingo Hidalgo 11/02/2024**

Contenido

[Enunciado 3](#_Toc164228191)

[Lista de entidades según enunciado de requerimientos 3](#_Toc164228192)

[Lista de atributos 3](#_Toc164228193)

[Relaciones 4](#_Toc164228194)

[Grafos binarios 4](#_Toc164228195)

[Grafo sin redundancias 4](#_Toc164228196)

[Matriz de relación 5](#_Toc164228197)

[Primera propuesta de diagrama MER **(Microsoft Visio)** 6](#_Toc164228198)

[Diagrama MERE o MER 7](#_Toc164228199)

[Diccionario de Datos 8](#_Toc164228200)

[Esquema transición 9](#_Toc164228201)

[Diagrama ER utilizando workbench 11](#_Toc164228202)

[Descripción de las 3 formas normales 12](#_Toc164228203)

[Primera forma normal: 12](#_Toc164228204)

[Segunda forma normal: 13](#_Toc164228205)

[Tercera forma normal: 14](#_Toc164228206)

[Codigo 15](#_Toc164228207)

[Datos de registro 17](#_Toc164228208)

[Joins 23](#_Toc164228209)

[REFERENCIAS 24](#_Toc164228210)

# Enunciado

El restaurante “La Brava” es un establecimiento de comida mexicana, ubicado en la Ciudad de México, por lo cual ha experimentado un incremento en sus ventas, por lo tanto necesita gestionar su registro en el cual se tiene un inventario de cada producto (gestionado por varios empleados); además se lleva a cabo un registro de los pedidos a los proveedores de nuevos productos, que se verán reflejados cuando se agreguen al inventario.

## Lista de entidades según enunciado de requerimientos

- REGISTRO

- PRODUCTO

- EMPLEADO

- PEDIDO

- PROVEEDOR

## Lista de atributos

- REGISTRO: ID\_Registro, Cantidad, Fecha del Registro.

- PRODUCTO: Id\_Producto, Nombre del Producto, Unidad de Medida, Precio.

- EMPLEADO: Id\_Empleado, Nombre(s), Apellido, Núm. Teléfono, Fecha de contratación, Usuario, contraseña.

- PEDIDO: ID\_Pedido, Fecha de entrega estimada, Fecha de Pedido, Estado del Pedido.

- PROVEEDOR: ID\_Proveedor, Nombre de la Empresa, RFC, Dirección, Correo electrónico, Teléfono.

## Relaciones

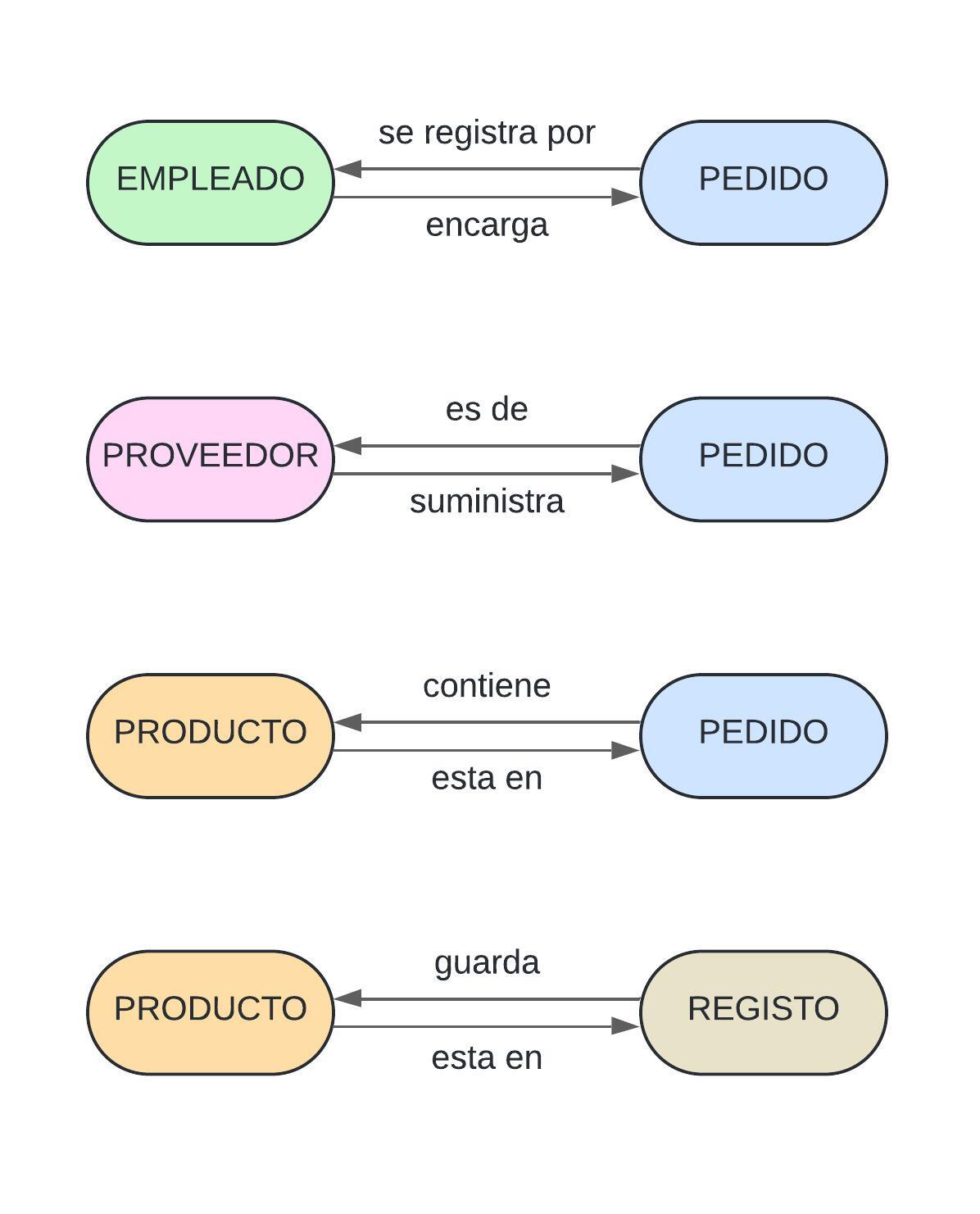
-EMPLEADO--------PEDIDO

-PROVEEDOR------PEDIDO

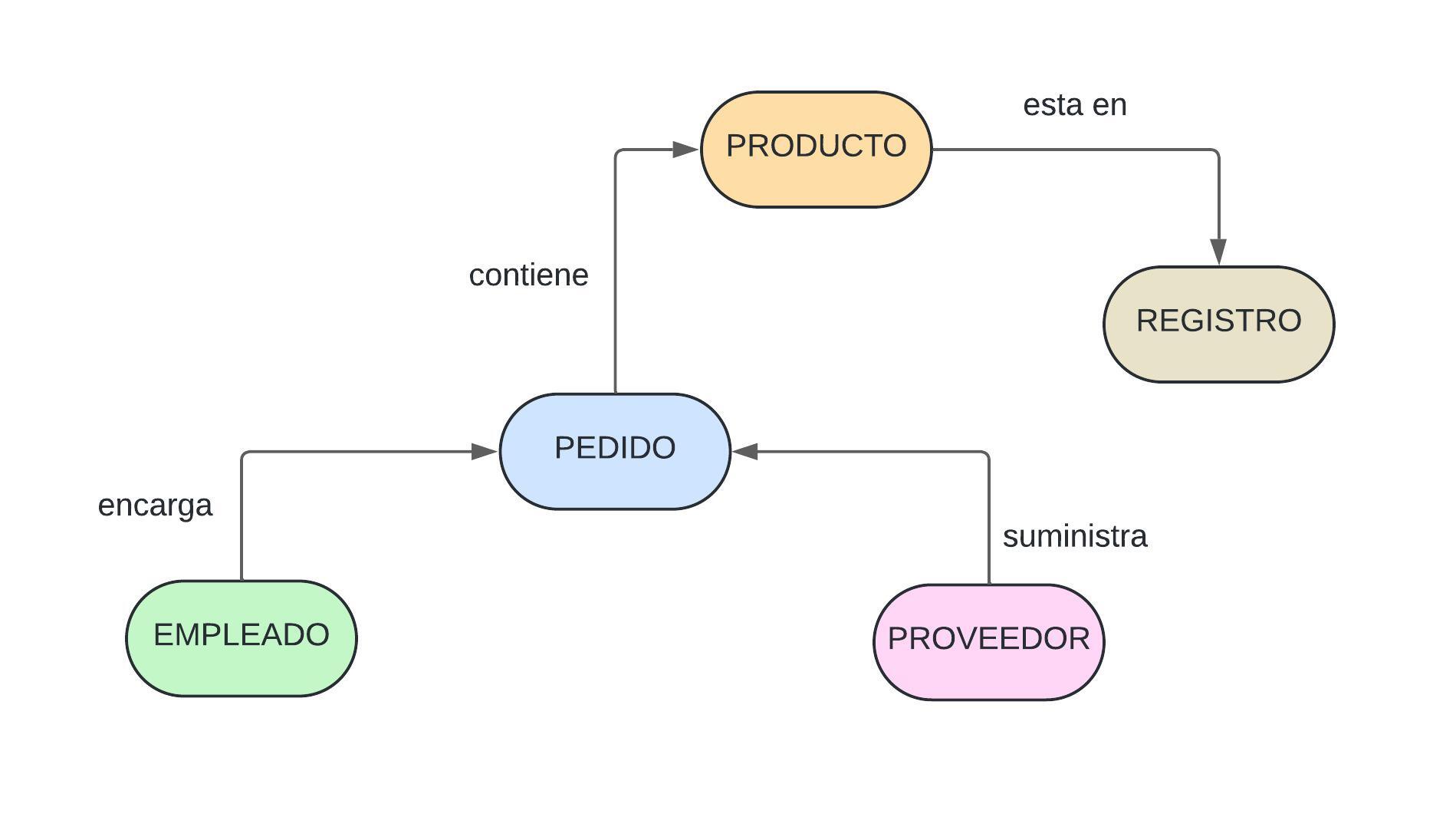
-PRODUCTO-------PEDIDO

-PRODUCTO-------REGISTRO

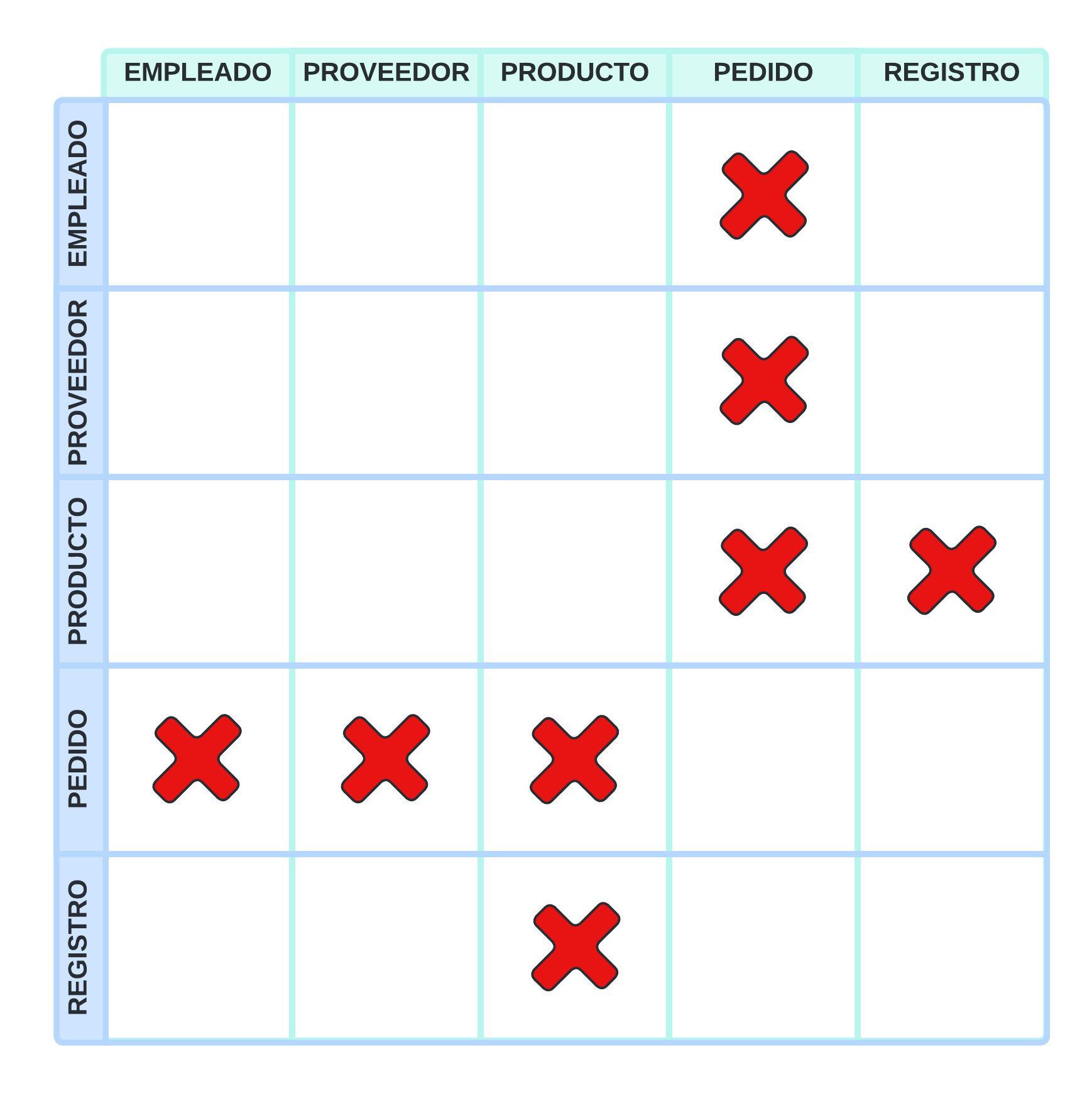
## Grafos binarios



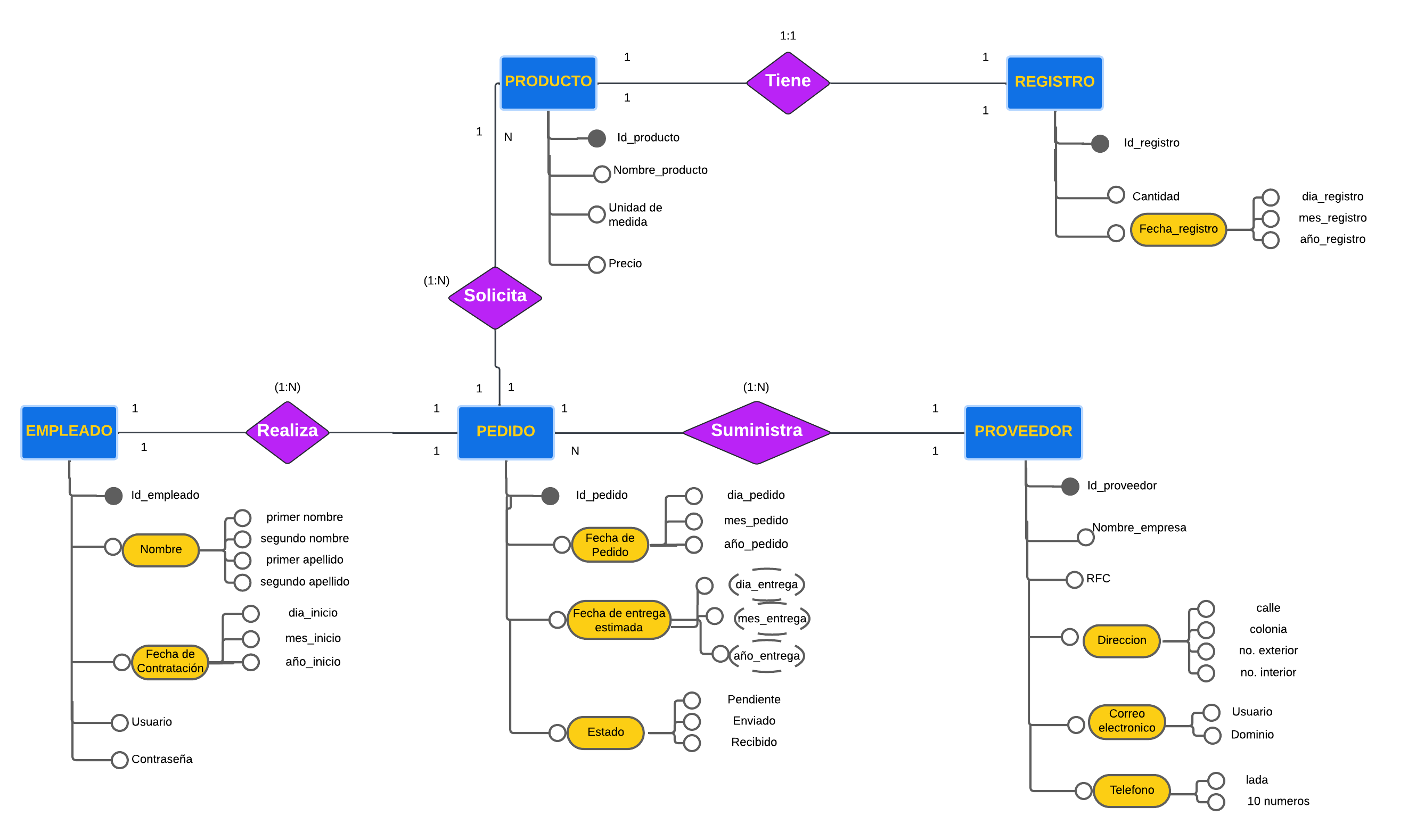
## Grafo sin redundancias



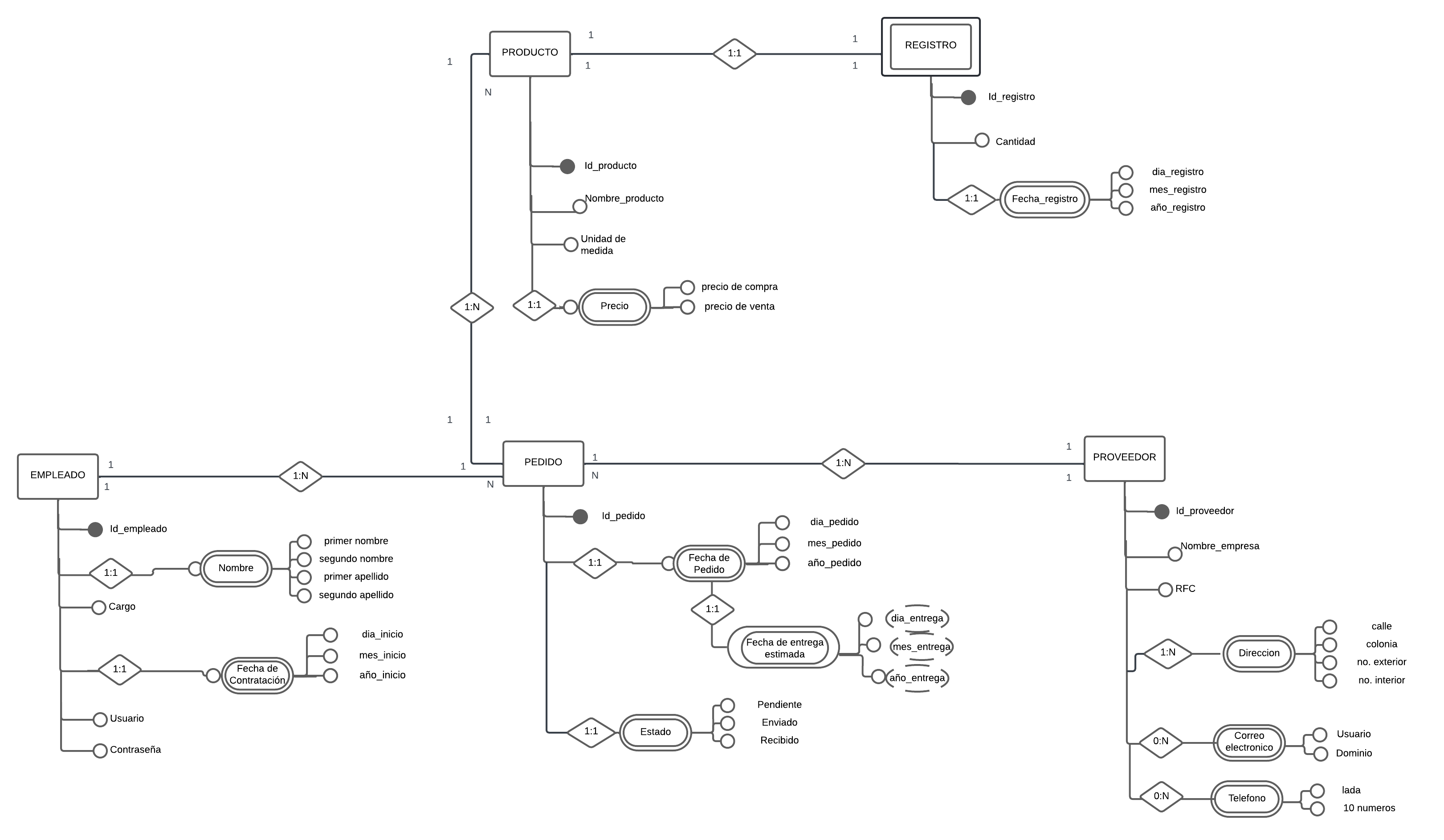
## Matriz de relación



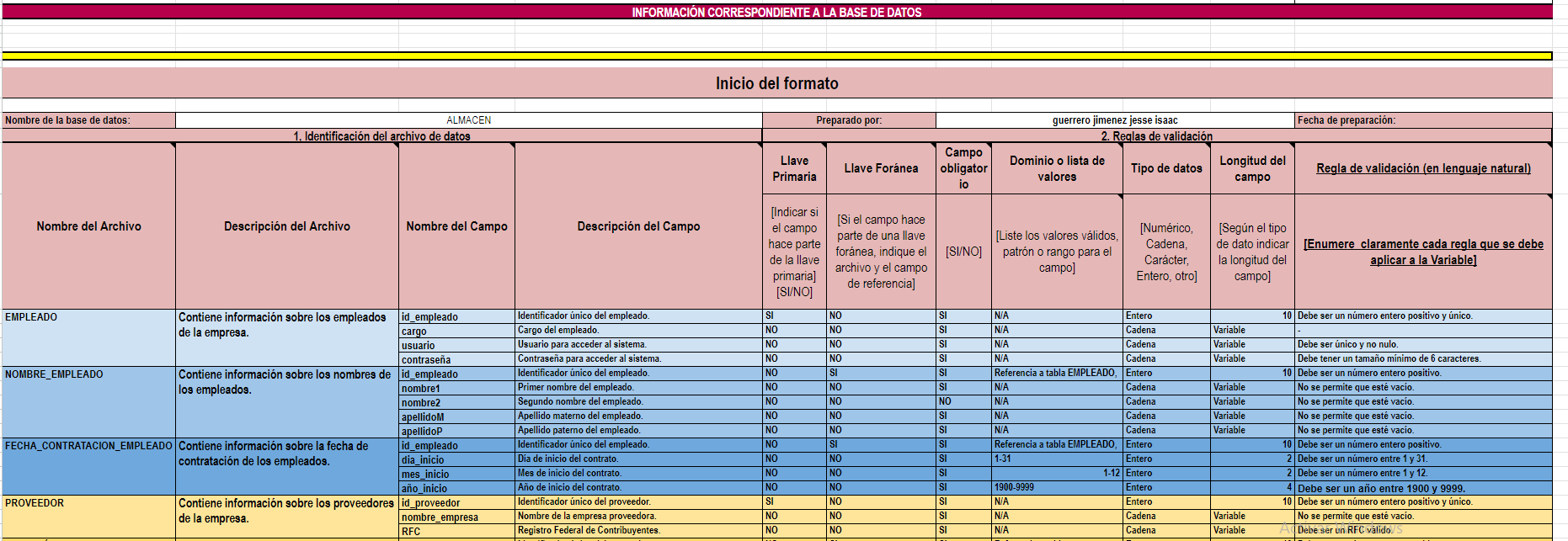
## Primera propuesta de diagrama MER **(Microsoft Visio)**



## Diagrama MERE o MER



## Diccionario de Datos



## Esquema transición

EMPLEADO (id\_empleado, cargo, usuario, contraseña) EMPLEADO.id\_empleado es una clave primaria.

NOMBRE (id\_empleado, nombre1, nombre2, apellidoM, apellidoP) NOMBRE.id\_empleado es una clave ajena que hace referencia a EMPLEADO.

FECHA\_CONTRATACION (id\_empleado, dia\_incio, mes\_inico, año\_inicio) FECHA\_CONTRATACION.id\_empleado es una clave ajena que hace referencia a EMPLEADO.

PROVEEDOR (id\_proveedor, nombre\_empresa, RFC) PROVEEDOR.id\_proveedor es una clave primaria.

DIRECCIÓN (id\_proveedor,calle, colonia, no.exterior, no.interior) DIRECCION.id\_proveedor es una clave ajena que hace referencia a PROVEEDOR.

CORREO (id\_proveedor,usuario, dominio) CORREO.id\_proveedor es una clave ajena que hace referencia a PROVEEDOR.

TELEFONO (id\_proveedor,lado,10\_numeros) TELEFONO.id\_proveedor es una clave ajena que hace referencia a PROVEEDOR.

PEDIDO (id\_pedido) PEDIDO.id\_pedido es una clave primaria.

FECHA\_PEDIDO (id\_pedido, dia\_pedido, mes\_pedido, año\_pedido) FECHA\_PEDIDO.id\_pedido es una clave ajena que hace referencia a PEDIDO.

FECHA\_ENTREGA (id\_pedido, dia\_entrega, mes\_entrega, año\_entrega) FECHA\_ENTREGA.id\_pedido es una clave ajena que hace referencia a PEDIDO.

ESTADO (id\_pedido, pendiente, enviado, recibido) ESTADO.id\_pedido es una clave ajena que hace referencia a PEDIDO.

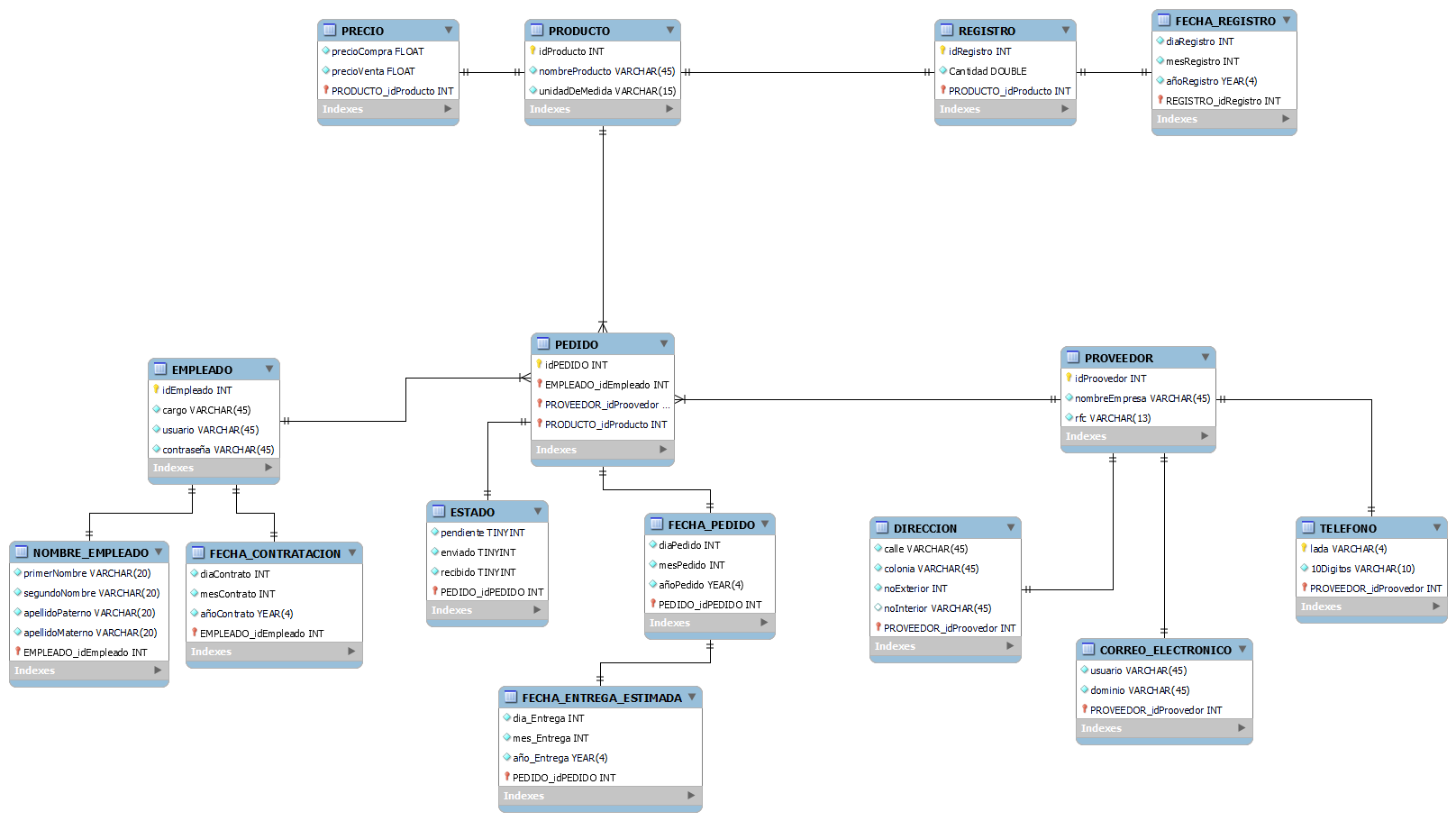
REGISTRO (id\_registro, cantidad) REGISTRO.id\_registro es una clave primaria.

FECHA\_REGISTRO (id\_registro, dia\_registro, mes\_registro, año\_registro) FECHA\_REGISTRO.id\_registro es una clave ajena de REGISTRO

PRODUCTO(id\_producto, nombre\_producto, unidad\_medida) PRODUCTO.id\_producto es una clave primaria.

PRECIO (id\_producto, precio\_compra, precio\_venta) PRECIO.id\_producto es una clave ajena de PRODUCTO

## Diagrama ER utilizando workbench



## Descripción de las 3 formas normales

La normalización de bases de datos en simples palabras es organizar nuestro conjunto de datos para evitar redundancias y duplicaciones.

Es un proceso que busca eliminar la redundancia, es decir, la duplicación innecesaria de información. Para lograr esto utilizaremos las formas normales; recordando que las formas normales son un conjunto de reglas que sirven como guía para estructurar y organizar las bases de datos de manera eficiente. Su objetivo principal es evitar la redundancia, mejorar la integridad de los datos y facilitar su acceso y manejo.

En nuestra propuesta sobre la gestión del almacén del restaurante “La Brava” hemos aplicado estas formas normales de la siguiente manera:

Primera forma normal: En esta etapa, nos enfocamos en eliminar la repetición innecesaria de datos dentro de la base de datos. Para ello comenzamos identificando y desglosando los atributos compuestos de cada entidad:

• PRODUCTO: En este caso se desgloso el apartado del “precio” para indicar que el producto debe tener un precio de compra y un precio de venta, a diferencia de un único precio fijo.

• REGISTRO: Teníamos como un atributo la “fecha de registro” y se desgloso por su “día, mes y año de registro”.

• EMPLEADO: El atributo “Nombre” se tenía que desglosar en sus distintos elementos que son “primer nombre”, “segundo nombre” (con la posibilidad de que sea nulo si el empleado no tiene dos nombres), “apellido paterno” y “apellido materno”. De la misma forma se descompuso el atributo “fecha de contratación” para tener una mejor gestión del “día, mes y año de contratación”.

• PEDIDO: Se ha desglosado tanto la “fecha de pedido” como la de “entrega” en sus respectivos “día, mes y año”, además del atributos “estado” que servirá para conocer en qué punto de la entrega se encuentra el pedido (“pendiente, enviado y/o entregado”

• PROVEEDOR: Se desgloso el atributo “dirección” considerando que al ser una empresa puede tener una o muchas sucursales y así solicitando la “calle,colonia, no. Exterior y no. Interior” (considerando la posibilidad de que puede o no tener no. Interior). También se aplica la primera forma normal al descomponer el atributo teléfono (“lada y 10\_dígitos”) y el “correo electrónico” (“usuario y dominio”) y a su vez aclarando que estos últimos dos atributos compuestos tienen una cardinalidad de cero a muchos (0:N), ya que pueden no tenerlos o tener varios.

Segunda forma normal: En la segunda forma normal cada atributo no-clave debe depender de la clave primaria completa, no solo de una parte de ella. Una dependencia funcional es una relación entre dos atributos, donde un atributo (el determinante) determina el valor del otro (el dependiente).

Para que una tabla cumpla con la segunda forma normal debe cumplir con dos requisitos:

1. Que cumpla con la primera forma normal.

2. No tener dependencias parciales.

Una vez que se había cumplido la primera condición en nuestro modelo procedimos a verificar que efectivamente no existieran dependencias parciales en nuestras tablas.

En un principio el atributo de la FECHA DE ENTREGA estaba siendo dependiente de PEDIDO, pero este a su vez dependía de la FECHA DE PEDIDO para determinar los valores que le corresponden a este atributo, por lo que no se cumplía la segunda forma normal, así que hicimos que FECHA DE ENTREGA fuera una tabla aparte que dependiera únicamente de FECHA DE PEDIDO

Tercera forma normal: La tercera forma normal se basa en la segunda forma normal y busca eliminar las dependencias transitivas, que pueden generar inconsistencias en los datos.

Para que una tabla cumpla con la tercera forma normal, debe cumplir con dos requisitos:

1. Estar en la segunda forma normal, en otras palabras, no tener dependencias parciales.

2. No tener dependencias transitivas, es decir que un atributo no-clave no puede depender de otro atributo no-clave, a menos que este último dependa de la clave primaria completa.

Llegados a este punto lo que teníamos que hacer era verificar que no existieran dependencias transitivas en nuestro modelo, y dado que esto se cumplía no fue necesario aplicar más modificaciones

## Codigo

create table Registro (

id\_registro INT PRIMARY KEY,

cantidad INT);

create table Fecha\_Registro (

id\_registro INT primary key,

dia\_registro int,

mes\_registro int,

año\_registro int,

foreign key (id\_registro) references Registro(id\_registro)

);

create table Producto (

id\_producto int primary key,

nombre\_producto varchar(255),

unidad\_medida varchar(50)

);

create table Precio (

id\_producto int primary key,

precio\_compra decimal(10, 2),

precio\_venta decimal (10, 2),

foreign key(id\_producto) references Producto(id\_producto)

);

create table Pedido (

id\_pedido int primary key,

id\_producto int,

cantidad int,

foreign key (id\_producto) references Producto(id\_producto)

);

create table Fecha\_pedido (

id\_pedido int primary key,

dia\_pedido int,

mes\_pedido int,

año\_pedido int,

foreign key (id\_pedido) references Pedido(id\_pedido)

);

create table Fecha\_entrega (

id\_pedido int primary key,

dia\_entrega int,

mes\_entrega int,

año\_entrega int,

foreign key (id\_pedido) references Pedido (id\_pedido)

);

create table Estado (

id\_estado int primary key,

estado\_nombre varchar(50)

);

create table Empleado (

id\_empleado int primary key,

cargo varchar (50),

usuario varchar (50),

contraseña varchar (50)

);

create table Nombre (

id\_empleado int primary key,

primer\_n varchar(50),

segundo\_n varchar(50),

primer\_ap varchar(50),

segundo\_ap varchar(50),

foreign key(id\_empleado) references Empleado(id\_empleado)

);

create table Fecha\_contratacion (

id\_empleado int primary key,

dia\_inicio int,

mes\_inicio int,

año\_inicio int,

foreign key (id\_empleado) references Empleado (id\_empleado)

);

create table Provedor (

id\_provedor int primary key,

nombre\_empresa varchar(100),

RFC varchar(50)

);

create table Direccion (

id\_provedor int,

calle varchar(100),

colonia varchar(100),

No\_exterior varchar(20),

No\_interior varchar(20),

foreign key (id\_provedor) references Provedor(id\_provedor)

);

create table Correo\_electronico (

id\_provedor int,

usuario varchar (50),

dominio varchar (50),

foreign key (id\_provedor) references Provedor(id\_provedor)

);

create table Telefono (

id\_provedor int,

lada varchar(50),

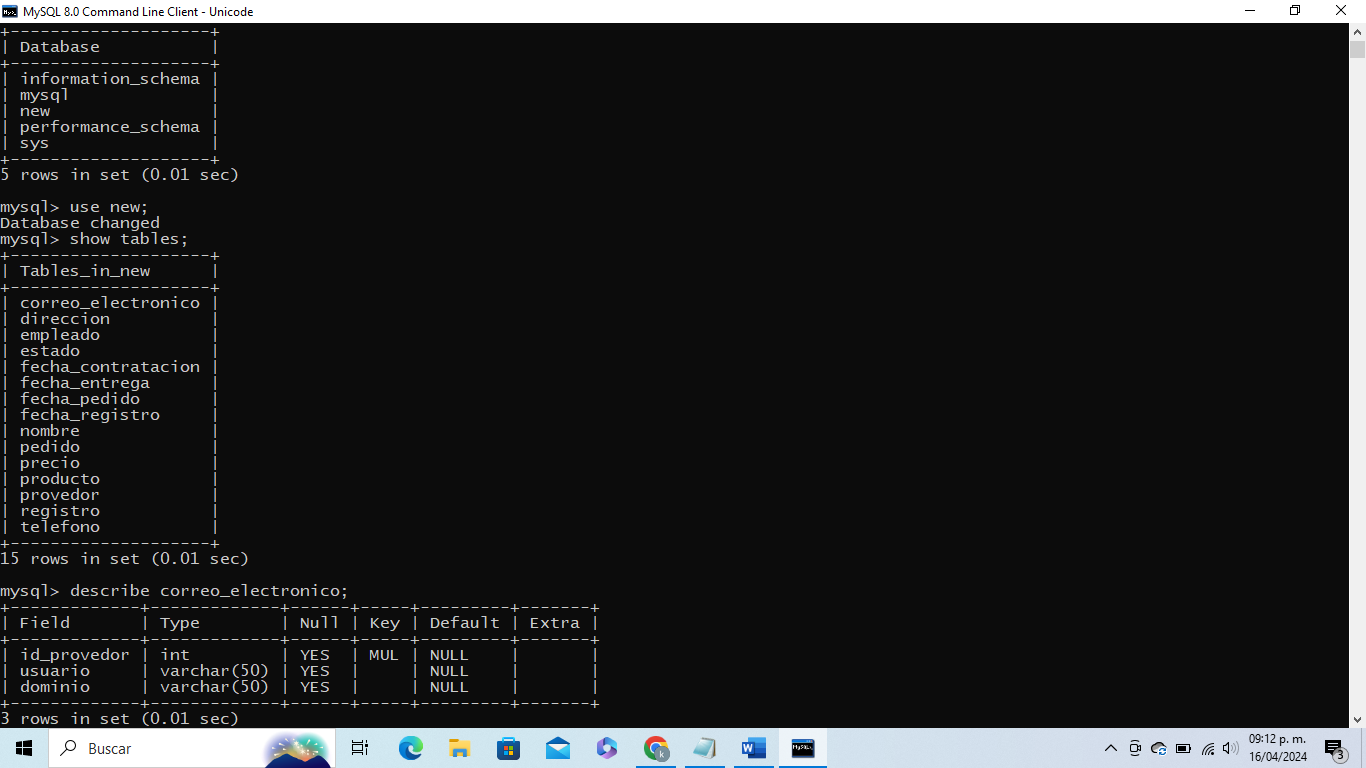
dominio varchar(50),

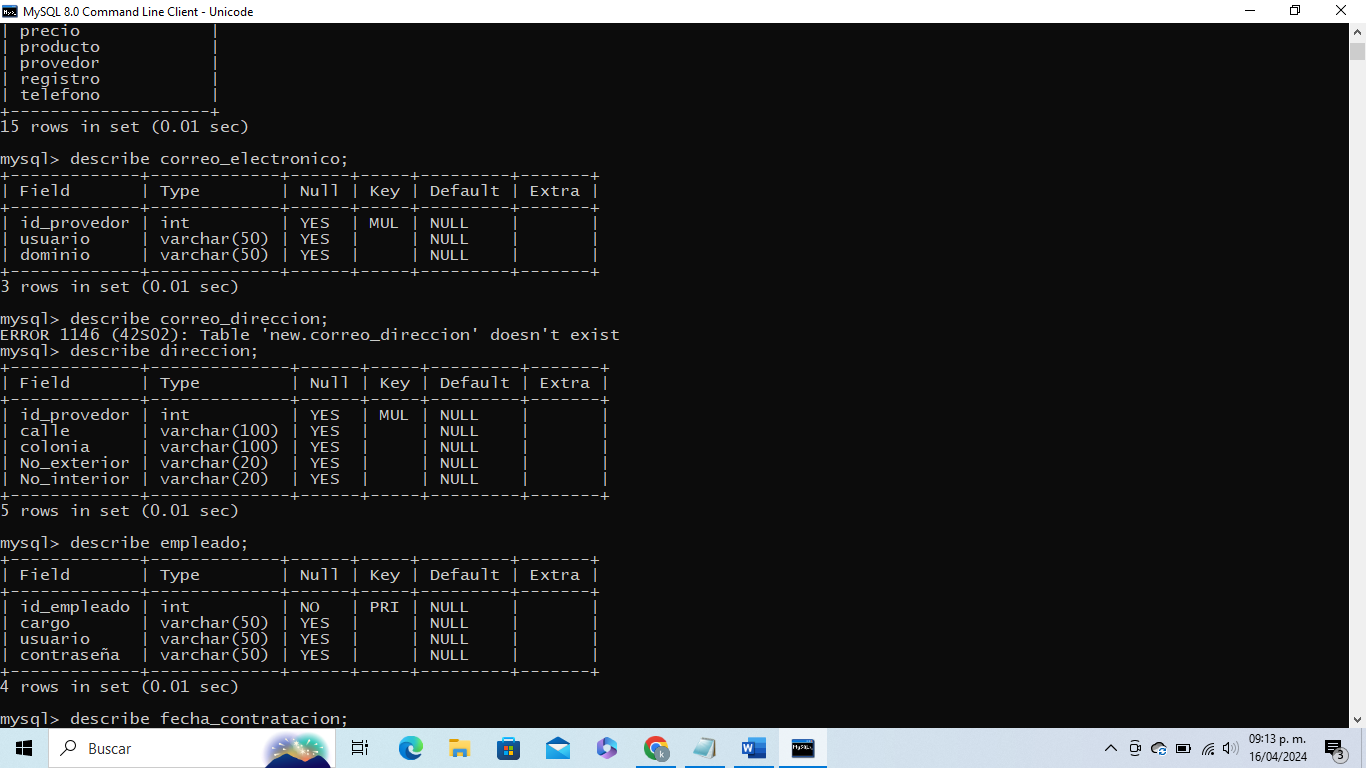
foreign key (id\_provedor) references Provedor(id\_provedor)

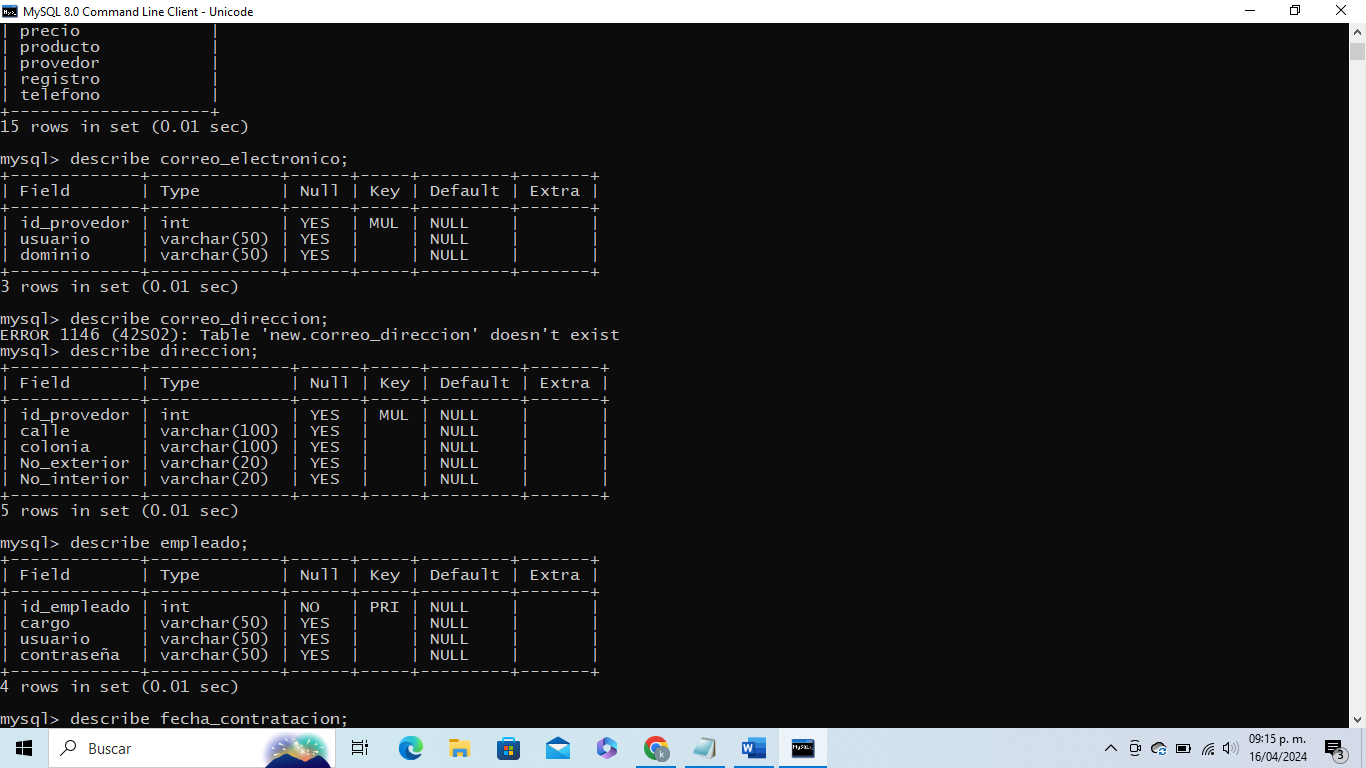
);

## Datos de registro

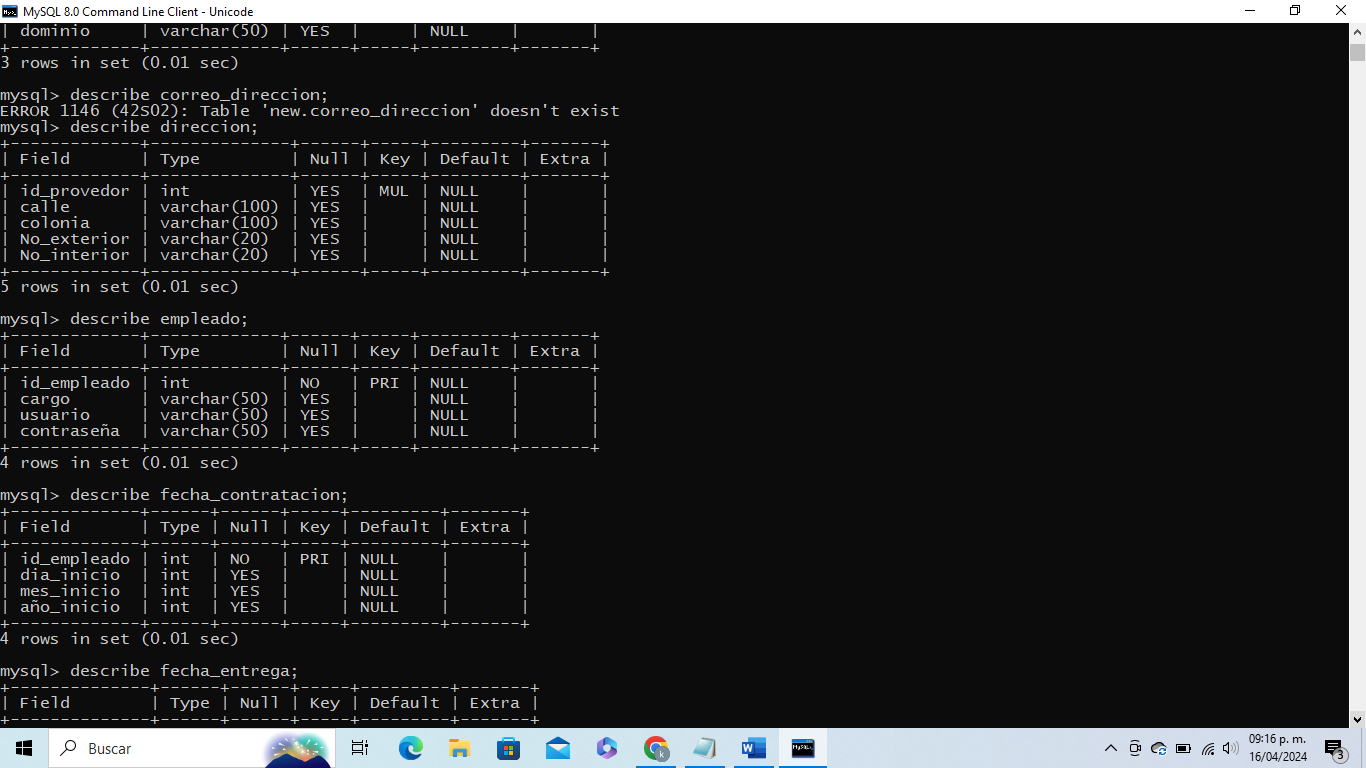
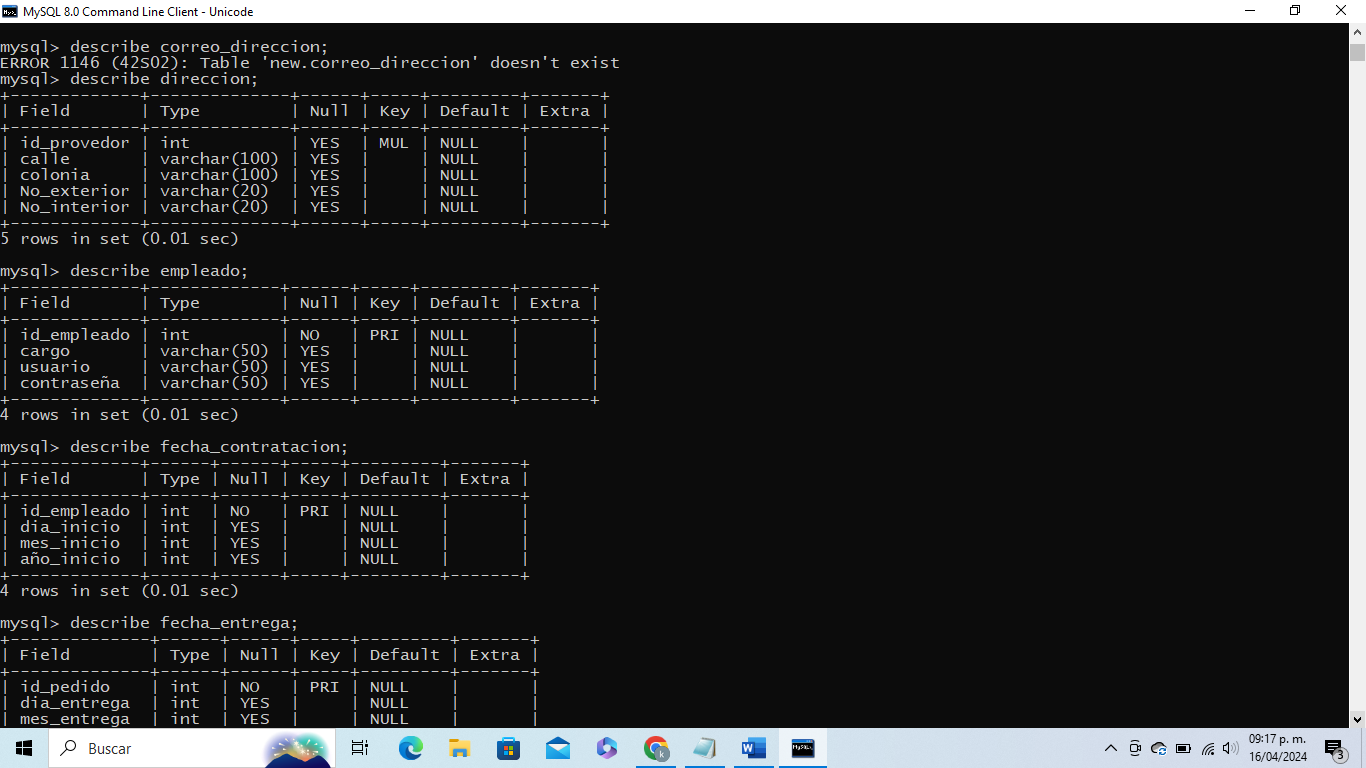
*Ilustración 1 mostrar las bases de datos, cambiando a la nueva base*

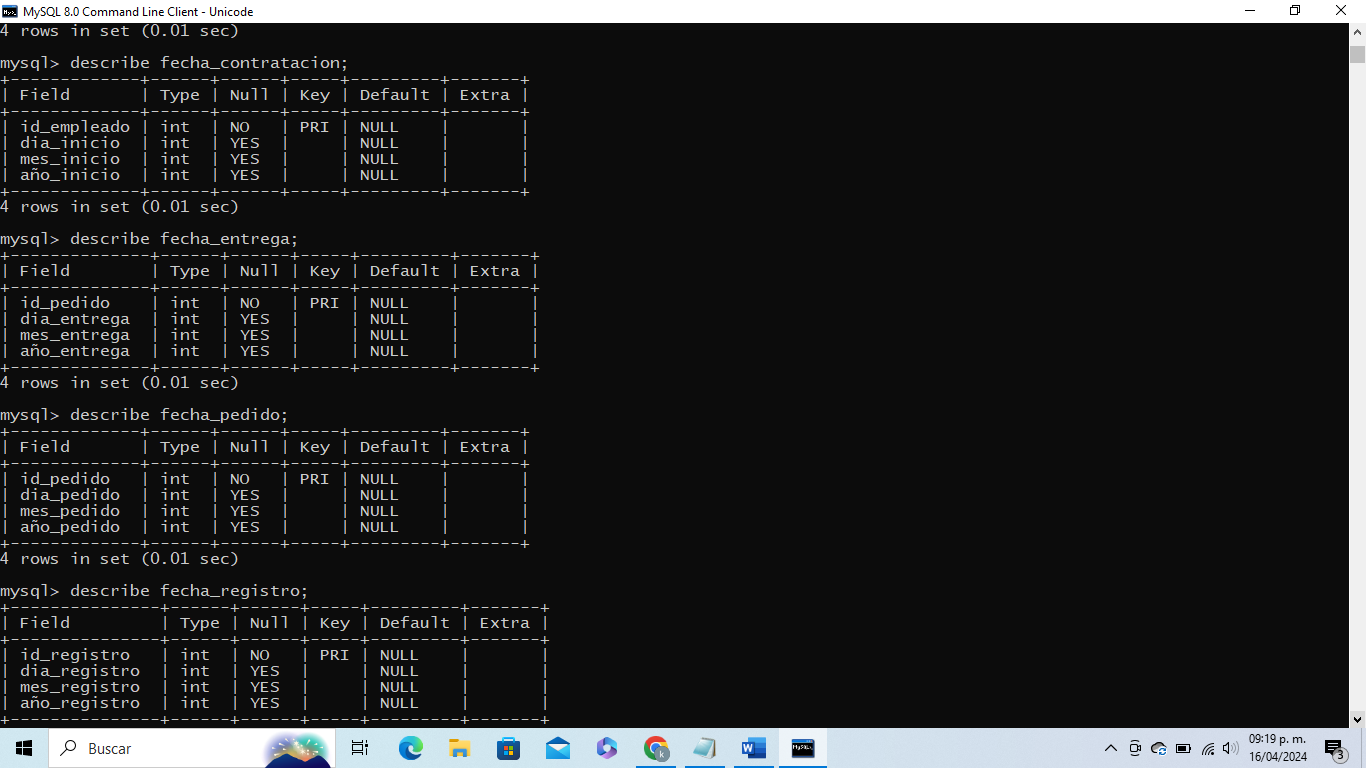
**

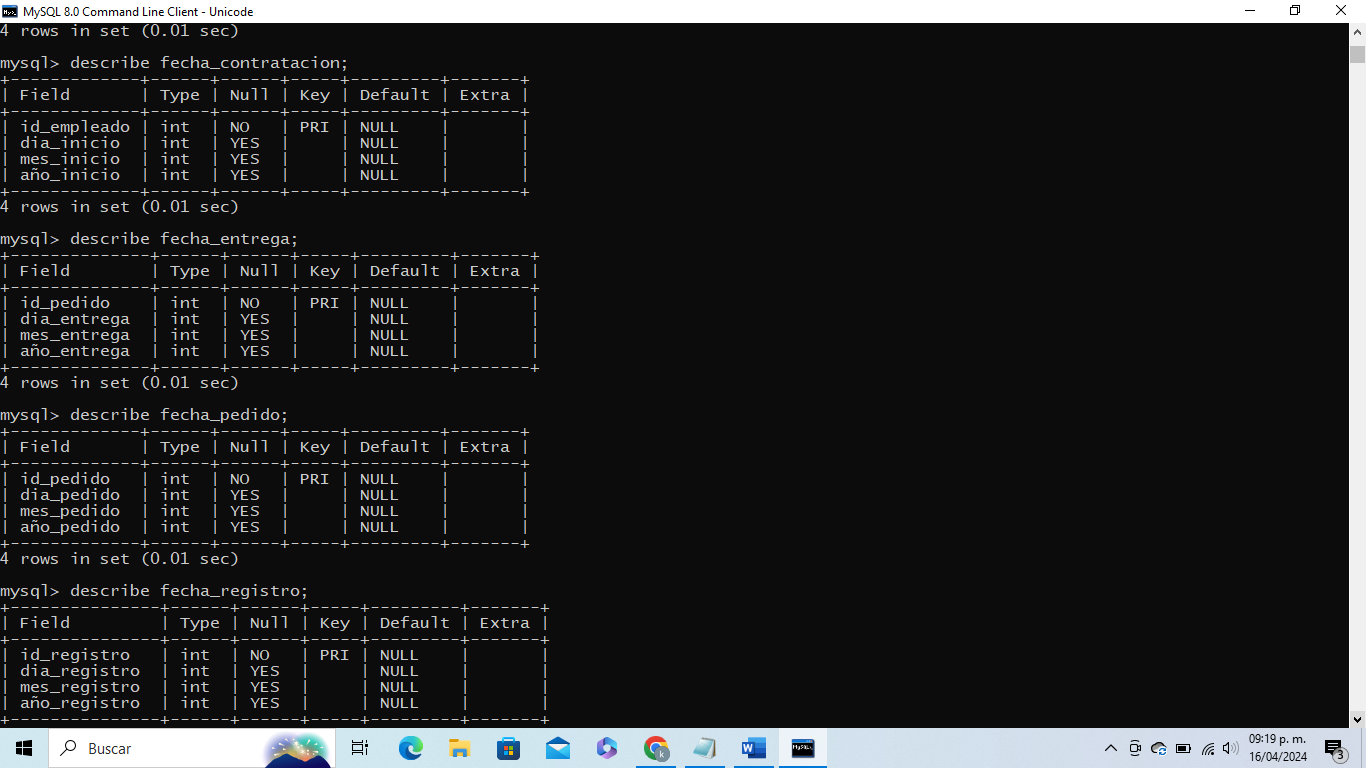
*Ilustración 2 mostrando todas las tablas creadas*

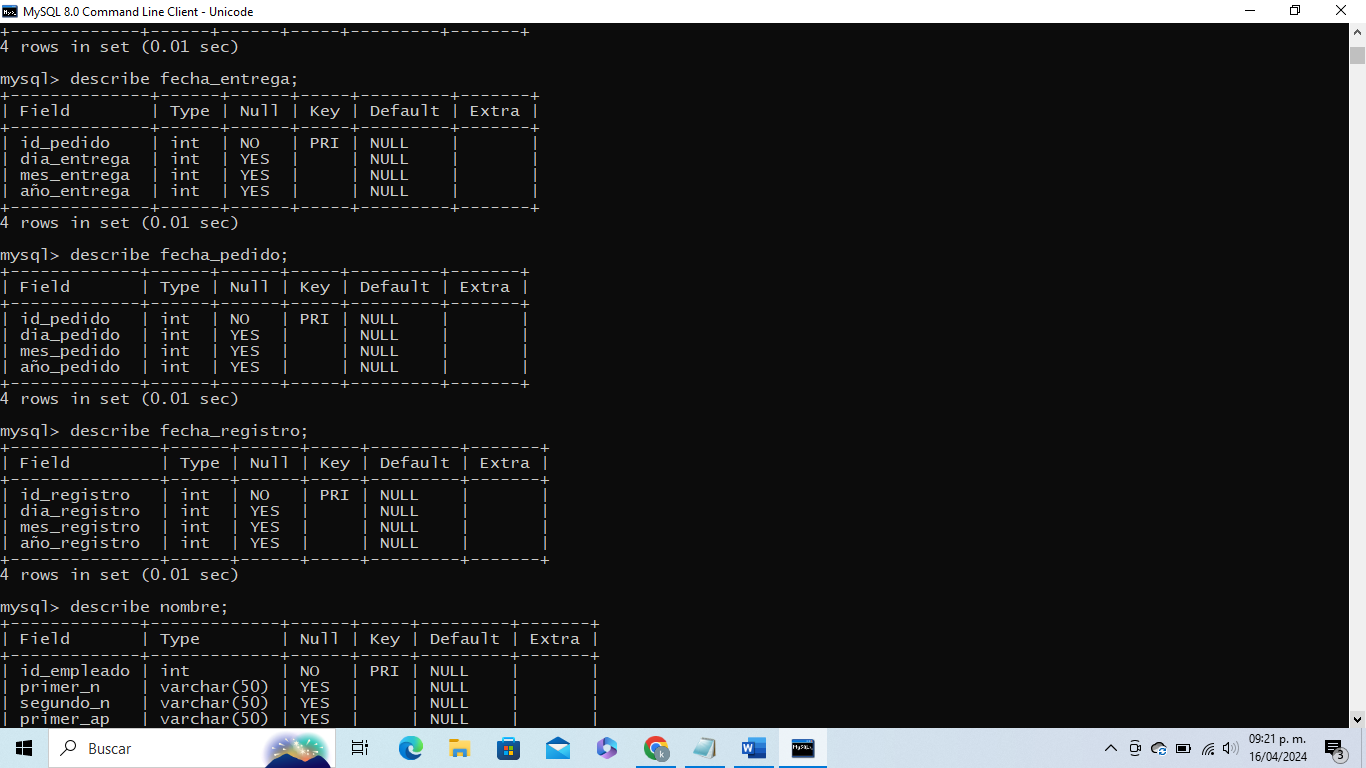
*Ilustración 3 describe la tabla de correo\_electronico*

*Ilustración 4 describe la tabla dirección*

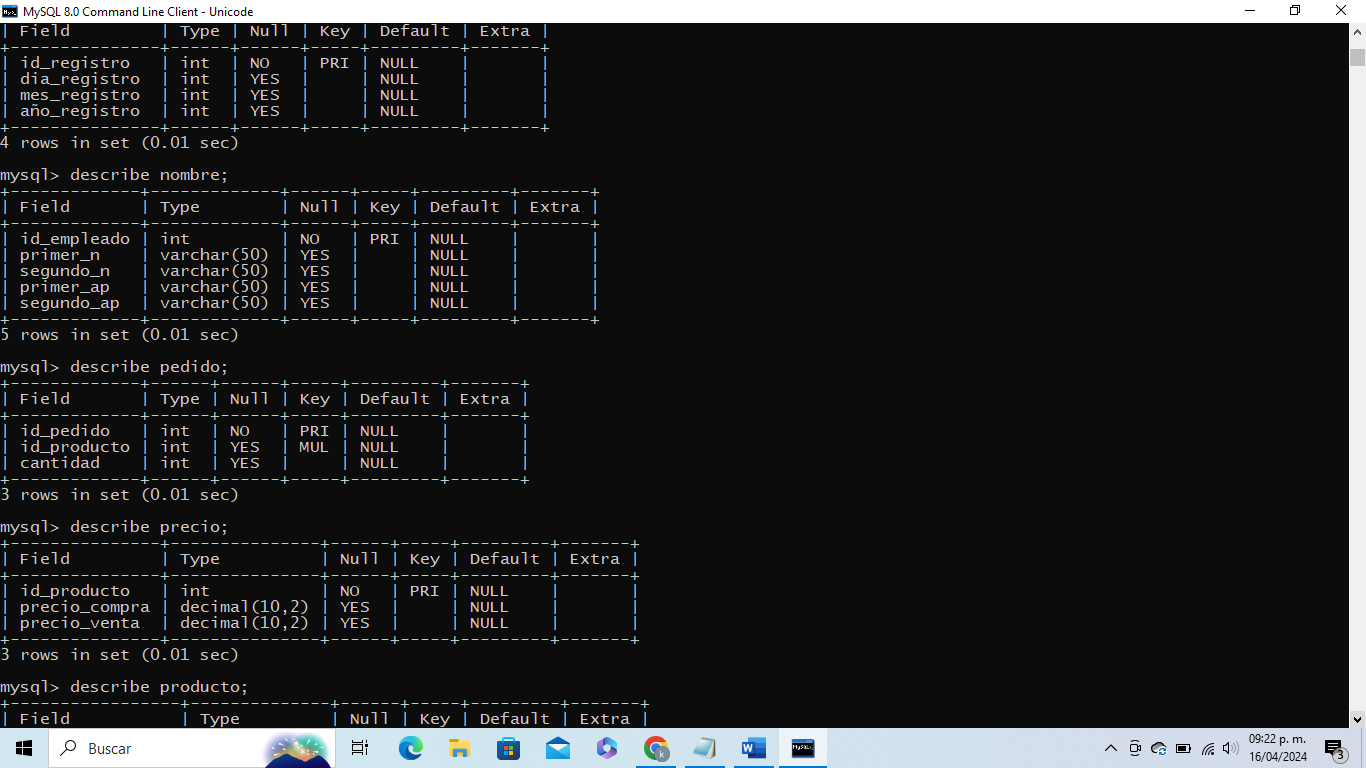
*Ilustración 5 describe la tabla empleada*

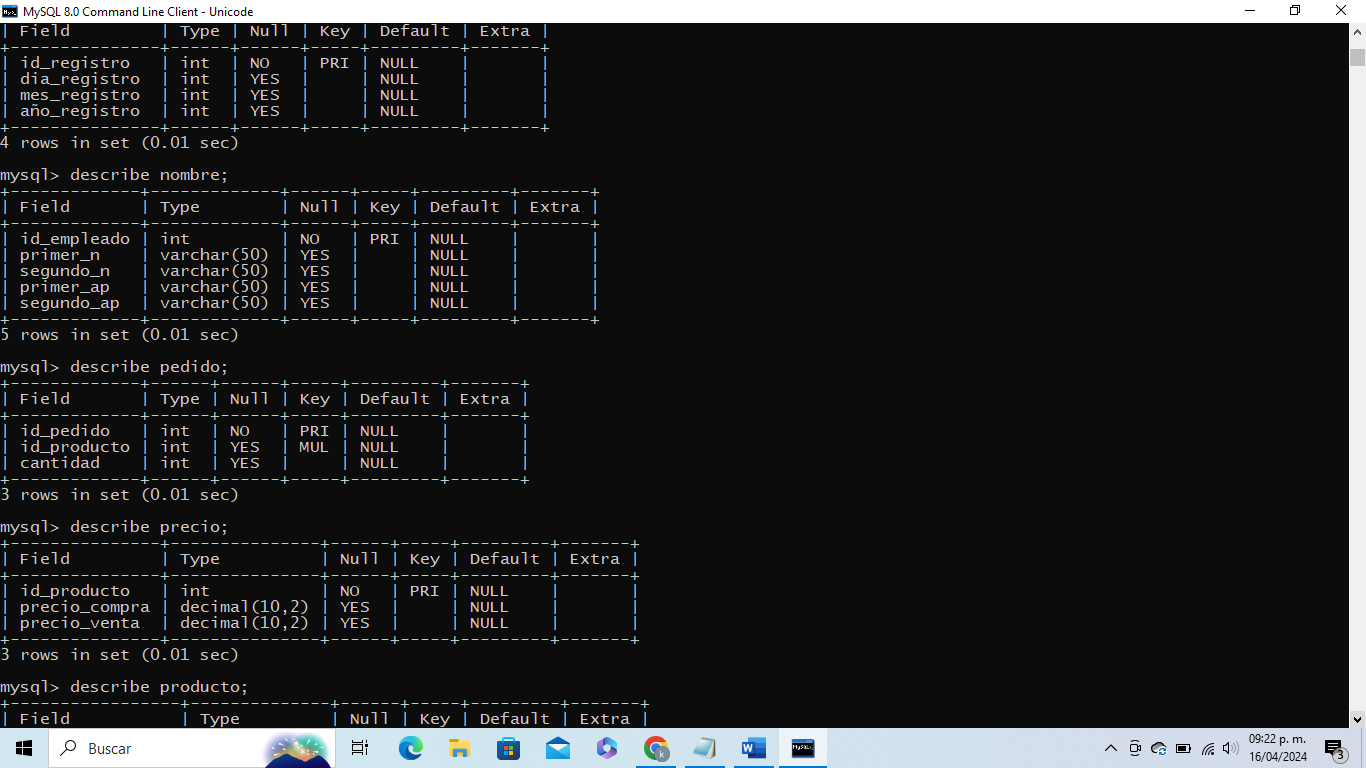
*Ilustración 6 describe la tabla fecha\_contratacion*

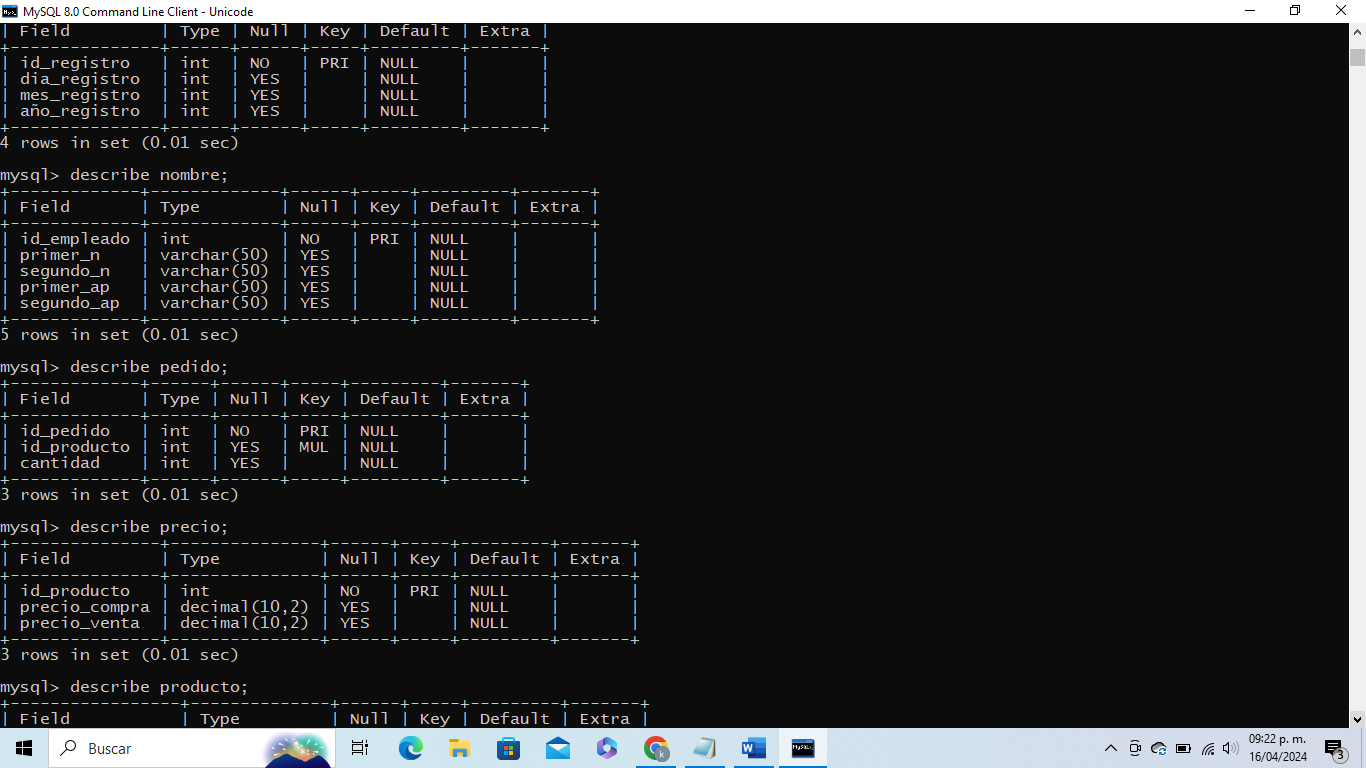
*Ilustración 7 describe la tabla fecha\_entrega*

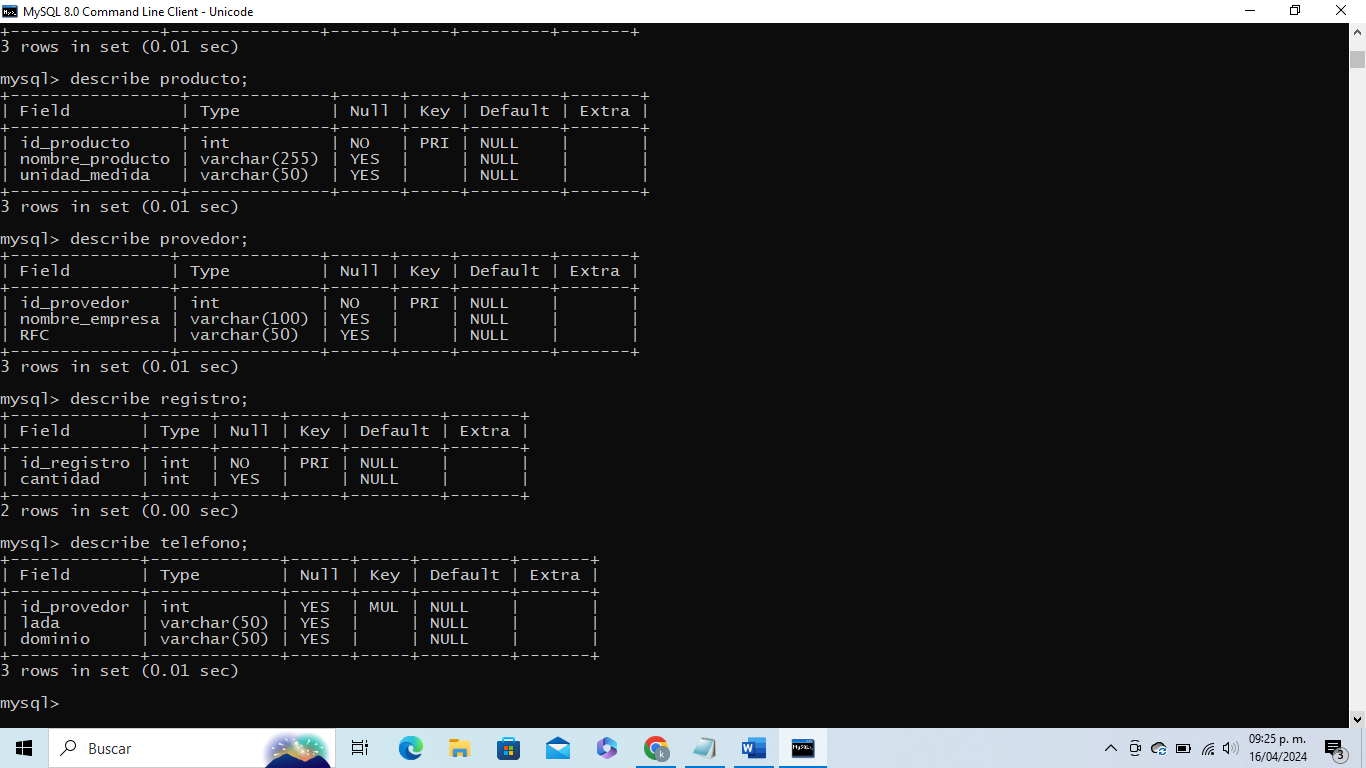
*Ilustración 8 describe la fecha\_pedido*

*Ilustración 9 describe la tabla fecha\_registro*

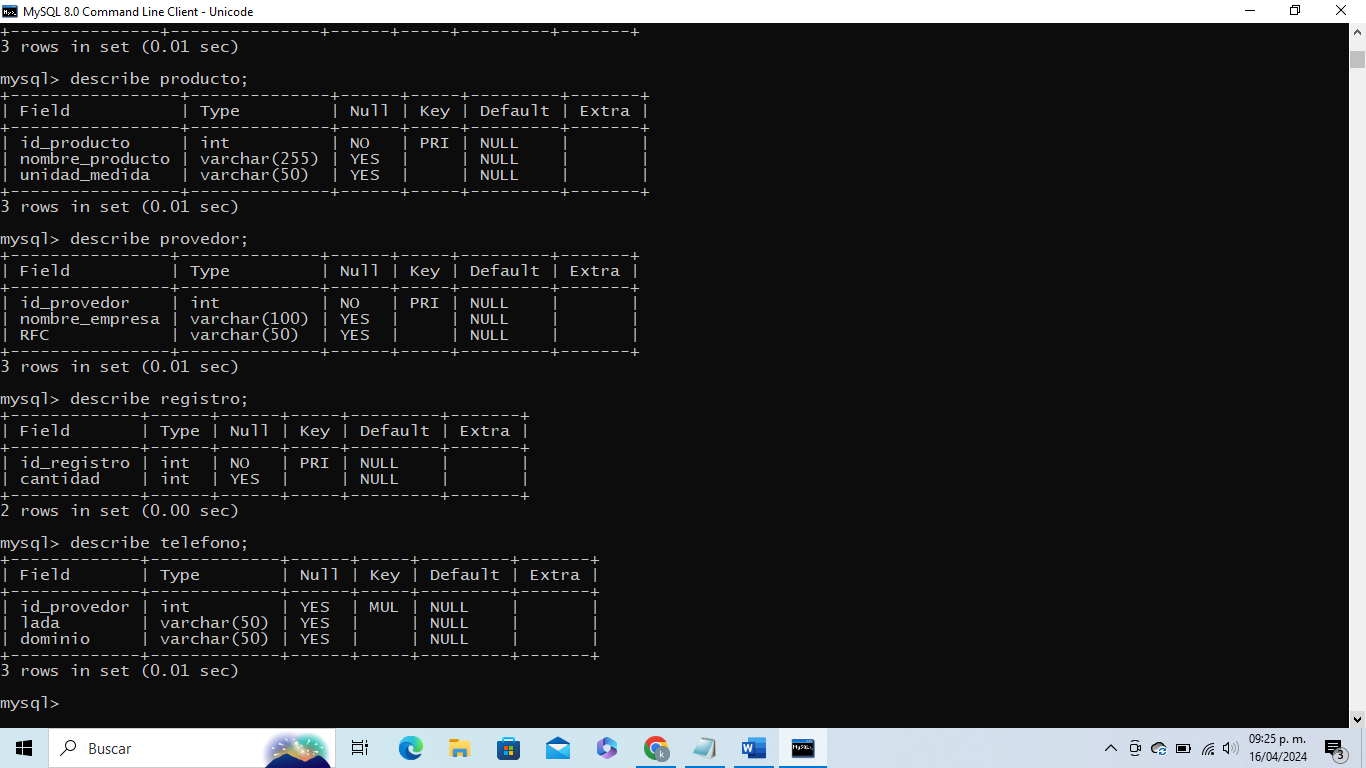
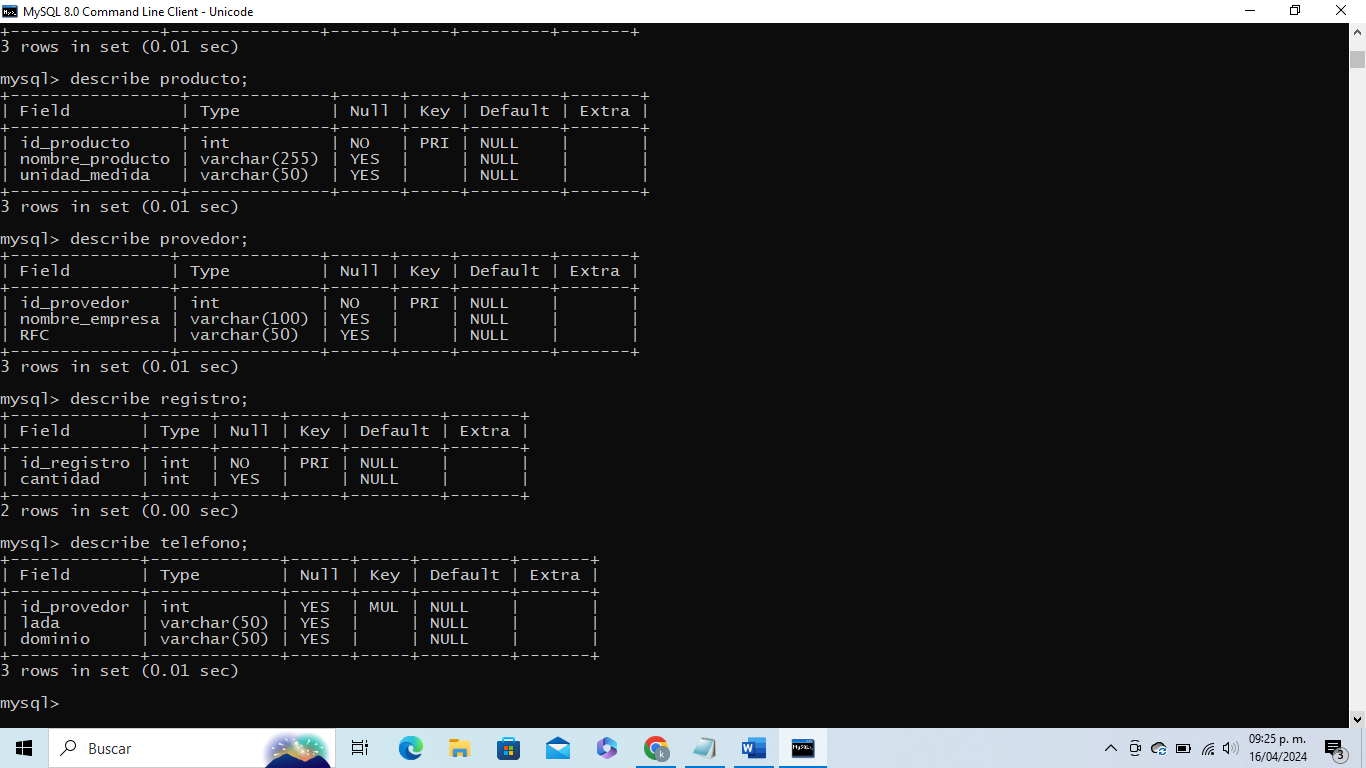
* Ilustración 10 describe la tabla nombre*

**

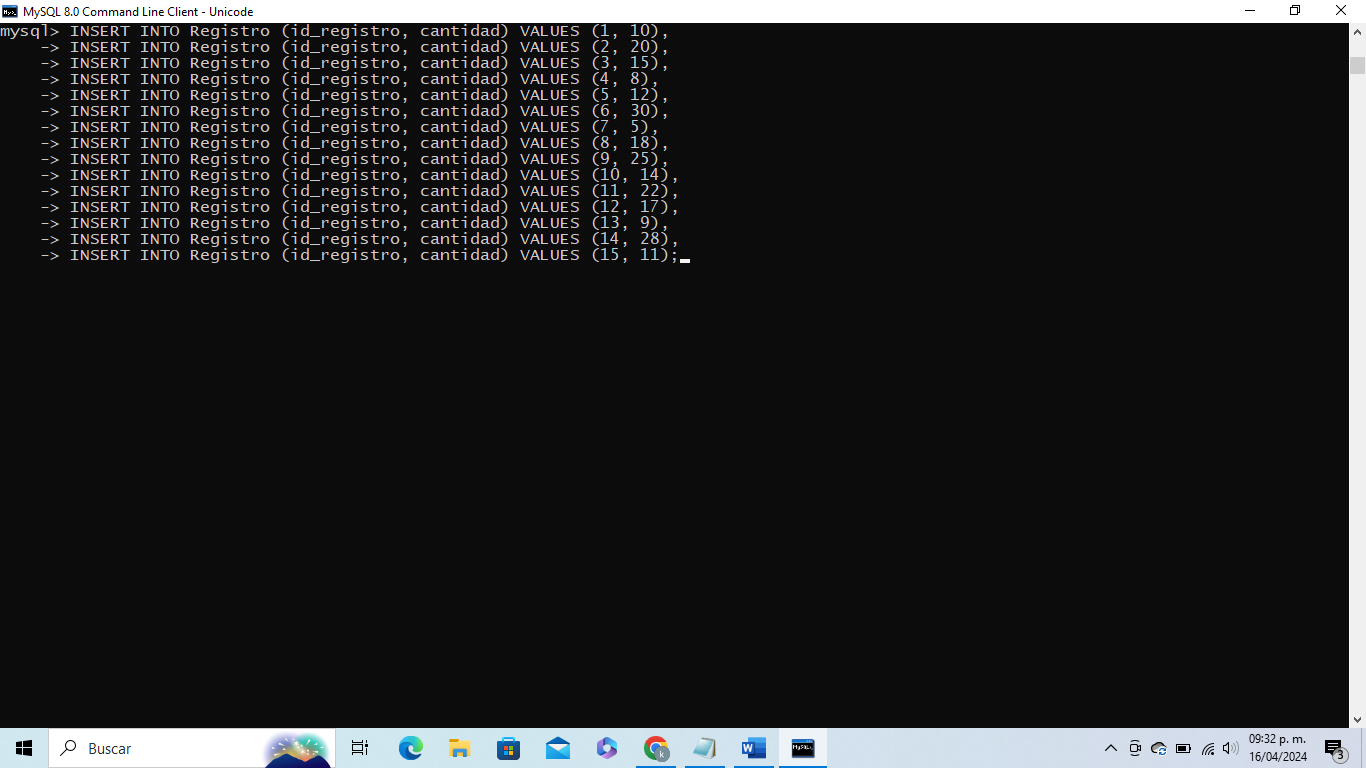
*Ilustración 11 describe la tabla pedido*

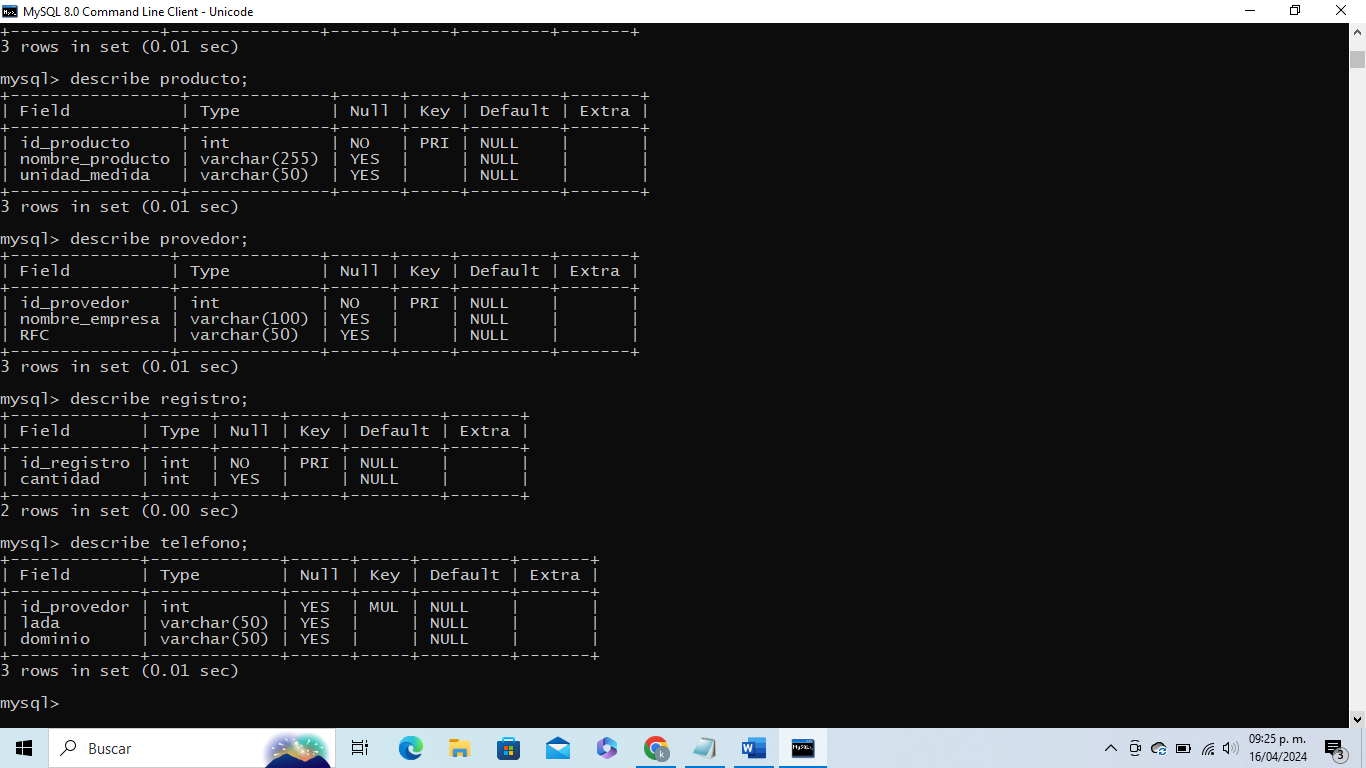
*Ilustración 12 describe la tabla precio*

*Ilustración 13 describe la tabla producto*

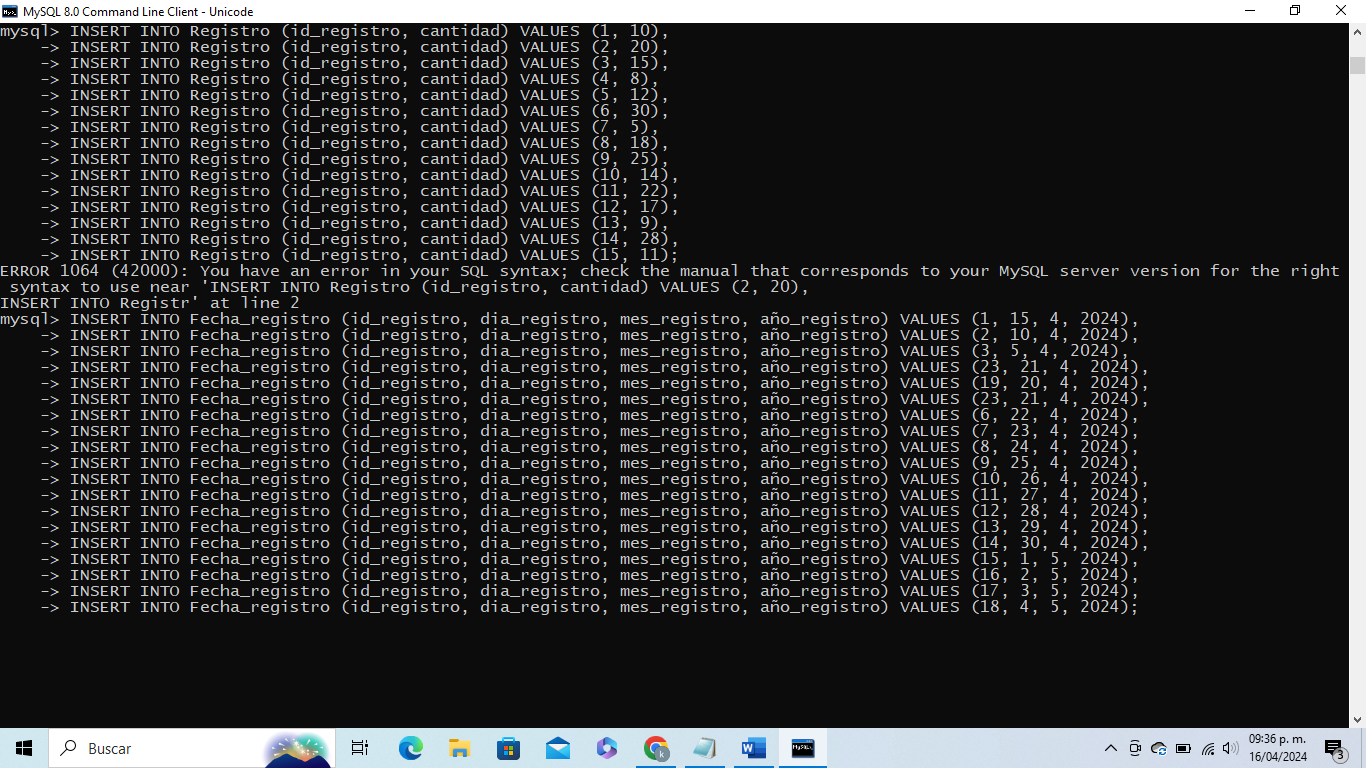
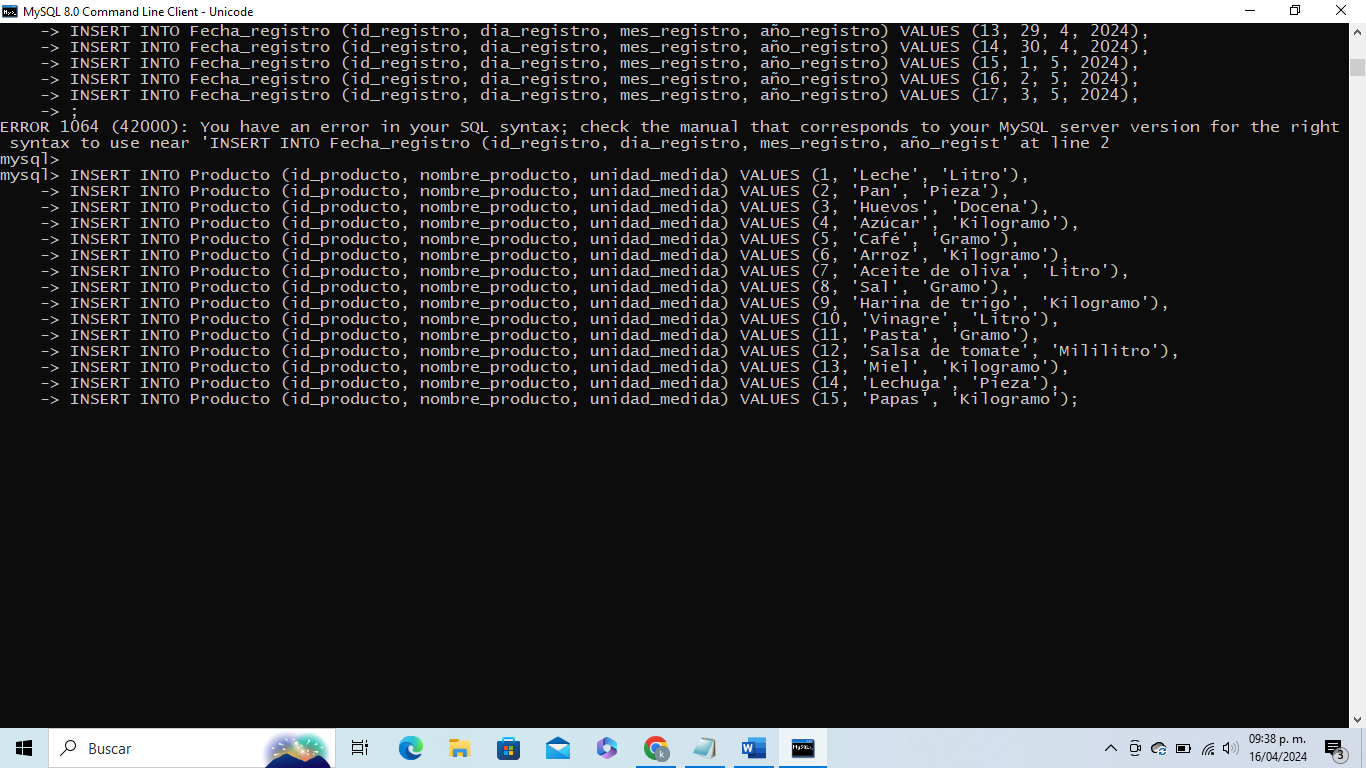
*Ilustración 14 describe la tabla provedor*

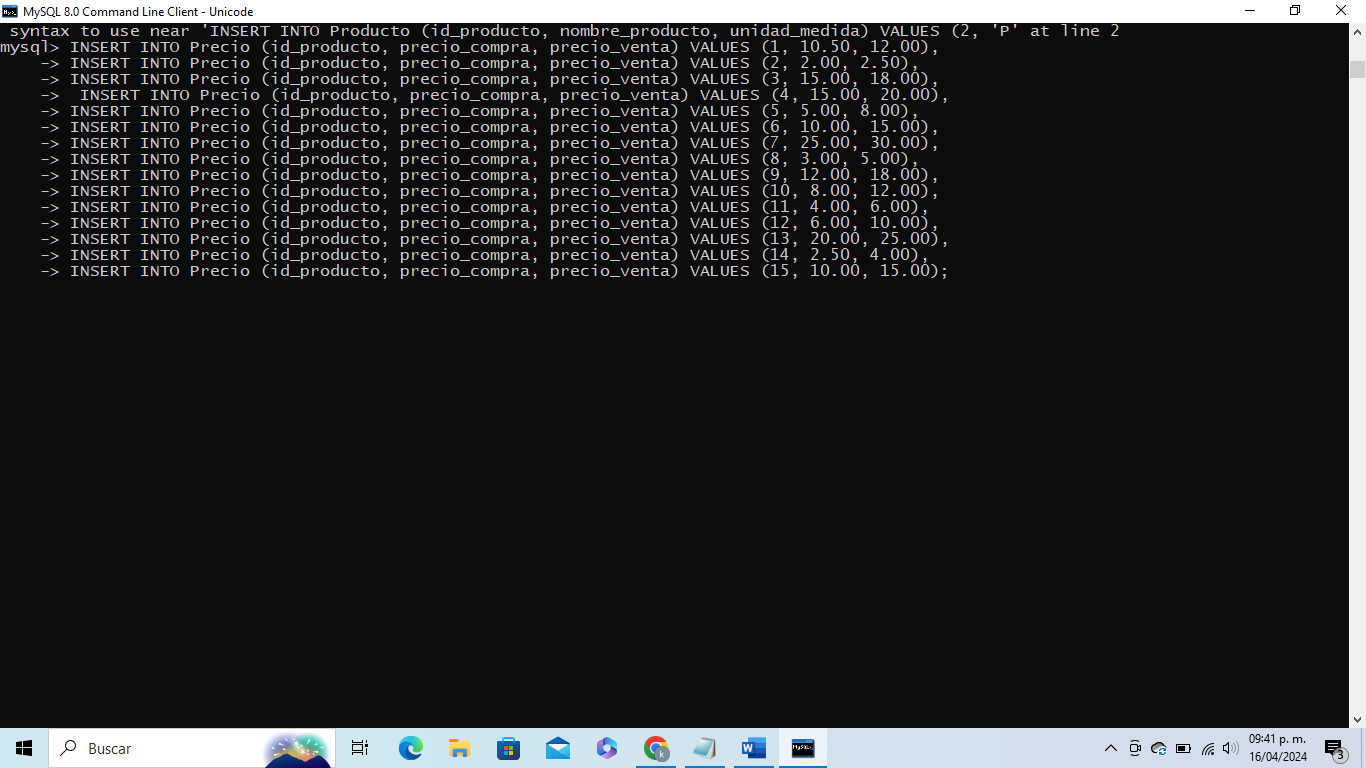
*Ilustración 15 describe la tabla registro*

*Ilustración 16 describe la tabla teléfono*

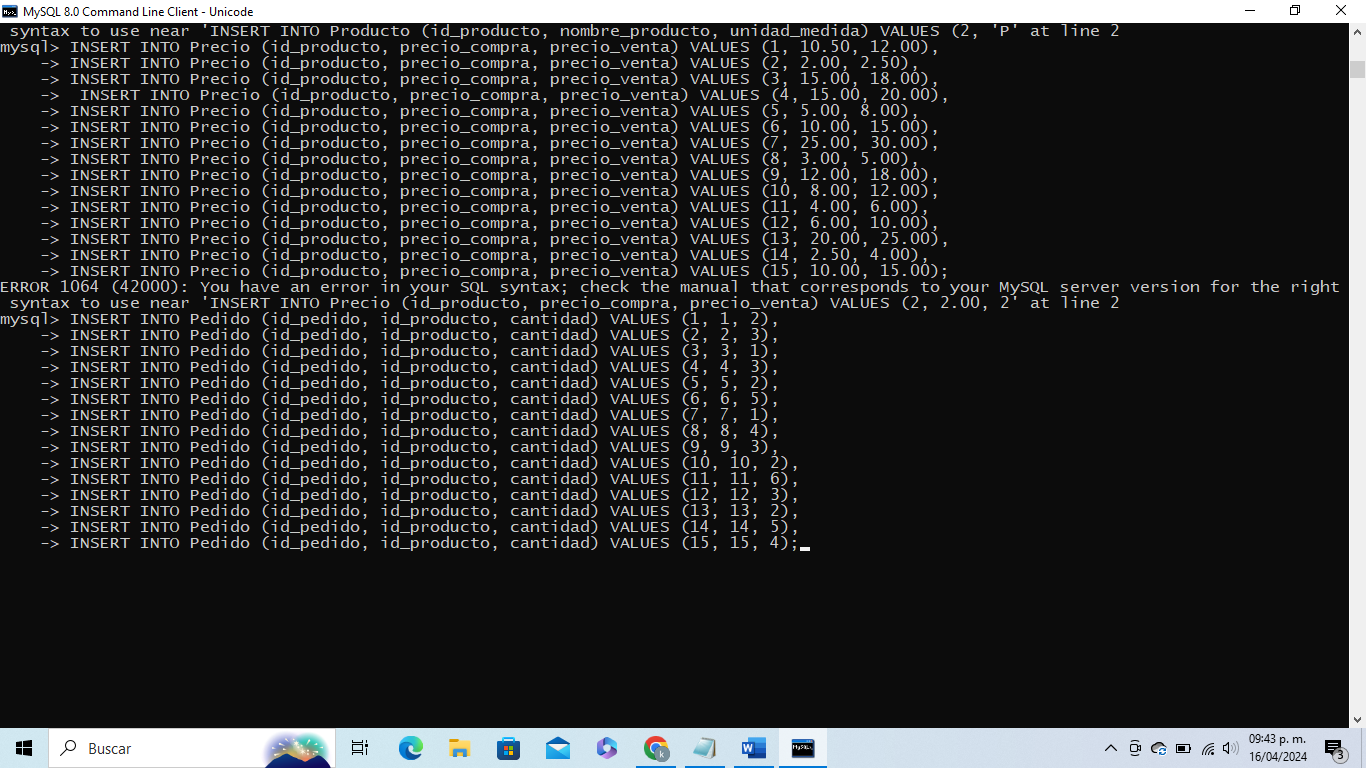
**

*Ilustración 17 código con datos ingresados a la tabla Registro*

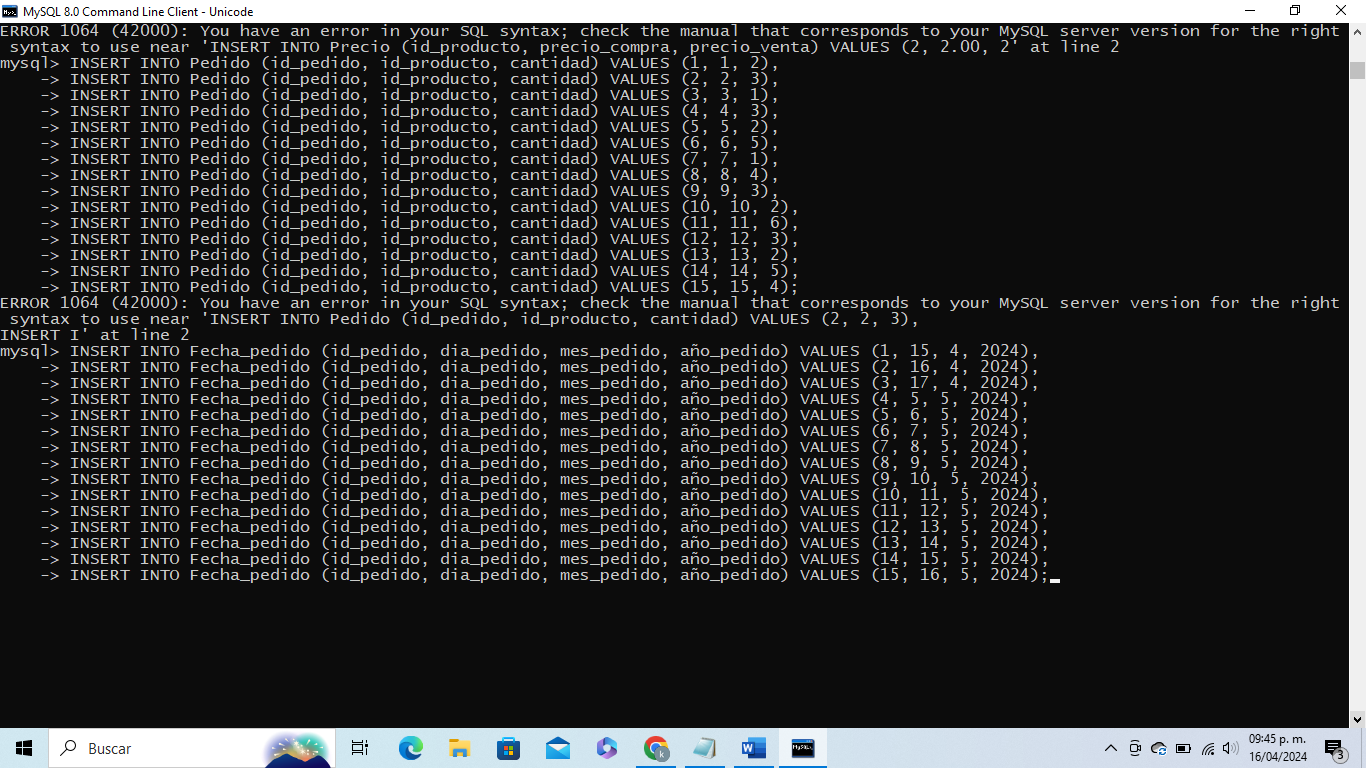
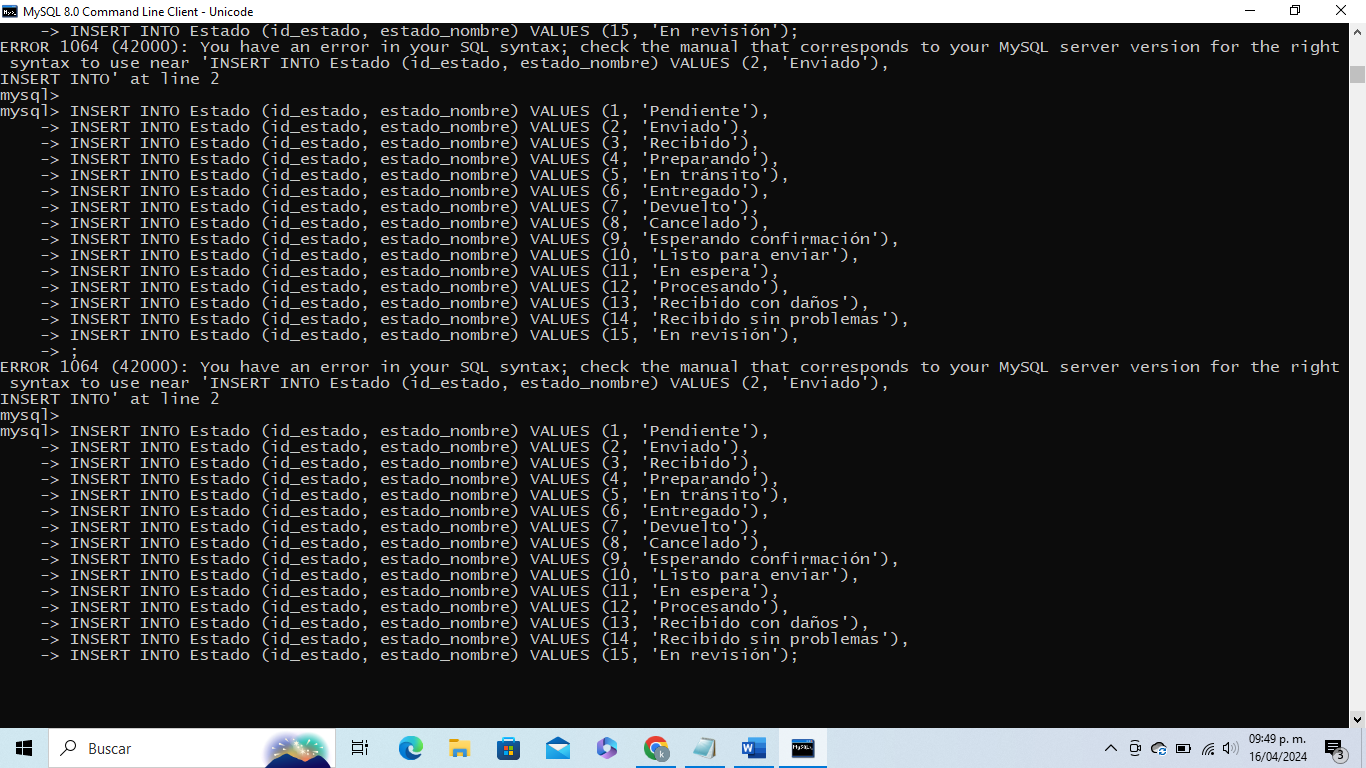
*Ilustración 18 código con datos ingresados a la tabla Fecha\_registro*

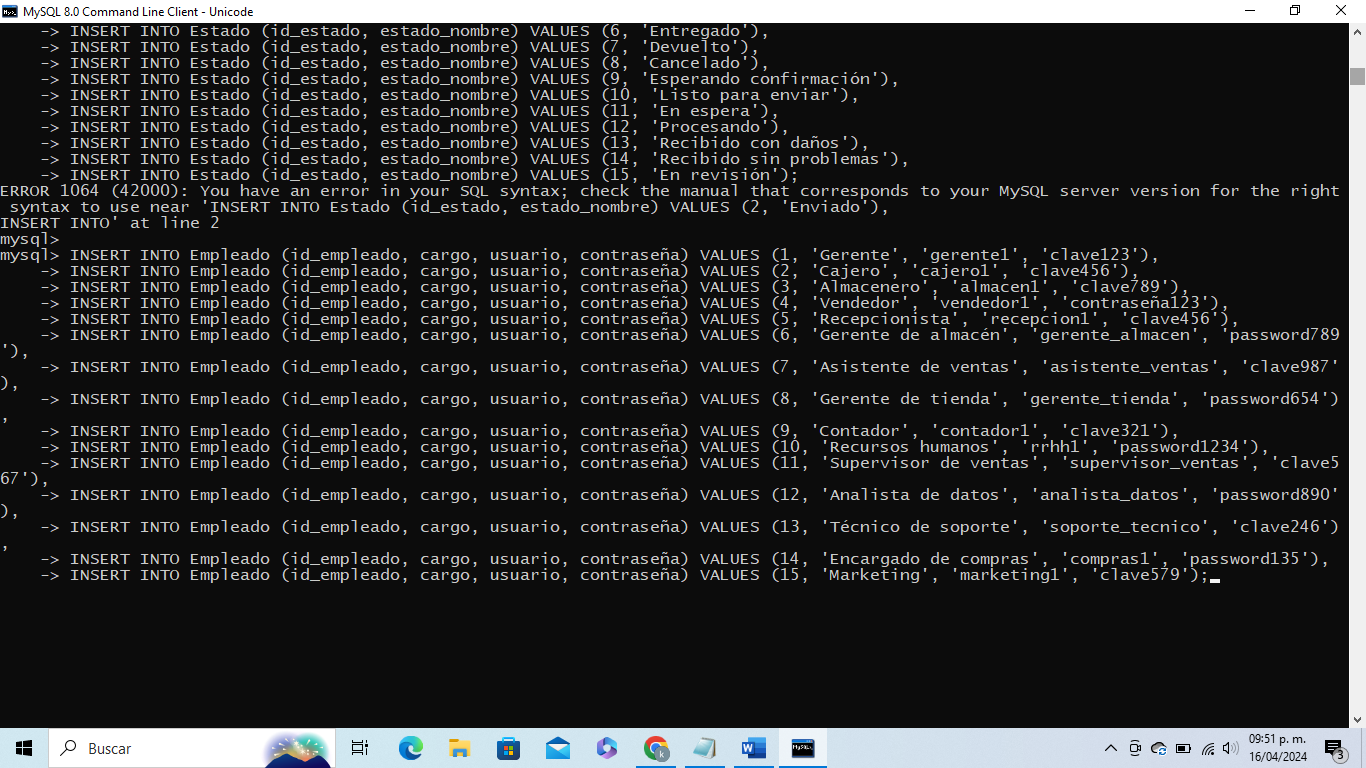
*Ilustración 19 código con datos ingresados a la tabla Producto*

*Ilustración 20 código con datos ingresados a la tabla Precio*

**

*Ilustración 21 código con datos ingresados a la tabla Pedido*

*Ilustración 22 codigo con datos ingresados a la tabla Fecha\_pedido*

*Ilustración 23 código con datos ingresados a la tabla Estado*

*Ilustración 24 código con datos ingresados a la tabla Empleado*

## Joins

*Ilustración 25 código con datos ocupando “join “*

## REFERENCIAS

Helenclu. (n.d.). *Descripción de la normalización de la base de datos - Microsoft 365 Apps*. Microsoft Learn. https://learn.microsoft.com/es-es/office/troubleshoot/access/database-normalization-description

Pérez, S. D. (2023, March 23). Gestor de Base de Datos: Qué es, funcionalidades y

ejemplos. Intelequia. https://intelequia.com/es/blog/post/gestor-de-base-de-datos-

qu%C3%A9-es-funcionalidades-y-ejemplos

Inc., L. (2010, January 18). Qué es un modelo de base de datos.

https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-

datos#:~:text=Un%20modelo%20de%20base%20de%20datos%20muestra%20la%20

estructura%20l%C3%B3gica,c%C3%B3mo%20se%20accede%20a%20ellos.

Torrejón, H. C. (2022, November 24). Cómo realizar la Normalización de bases de Datos y

Por Qué. OpenWebinars.net. https://openwebinars.net/blog/como-realizar-la-

normalizacion-de-bases-de-datos-y-por-que/