Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico-Matemáticas

Licenciatura en Multimedia y Animación Digital

Avance 2

Unidad de Aprendizaje: Programación y Aplicaciones para la Web

Maestro: José Orihuela Sánchez

Estudiante: Eliazar de Jesús Salazar Almaguer

Matrícula: 1663156

Grupo: 002

Fecha de entrega: miércoles, 10 de abril de 2019

**Contenido**

[Diagrama Entidad Relación Extendido 4](#_Toc5744361)

[Mapeo de datos 5](#_Toc5744362)

[Nombre de la base de datos: Eli DB 5](#_Toc5744363)

[Entidad 1: Usuario 5](#_Toc5744364)

[Entidad 2: Administrador 5](#_Toc5744365)

[Entidad 3: Cliente 5](#_Toc5744366)

[Entidad 4: Vendedor 5](#_Toc5744367)

[Entidad 5: Nota de crédito 6](#_Toc5744368)

[Entidad 6: Venta 6](#_Toc5744369)

[Entidad 7: Producto 6](#_Toc5744370)

[Entidad 8: Caja 6](#_Toc5744371)

[Relación 9.1: Administrador crea vendedor 7](#_Toc5744372)

[Relación 9.2: Administrador crea cliente 7](#_Toc5744373)

[Relación 9.3: Administrador crea caja 7](#_Toc5744374)

[Relación 10: Cliente pide Nota de crédito 7](#_Toc5744375)

[Relación 11: Cliente pide venta 8](#_Toc5744376)

[Relación 12: Nota de crédito devuelve venta 8](#_Toc5744377)

[Relación 13: Venta tiene producto 9](#_Toc5744378)

[Relación 14: Venta tiene caja 9](#_Toc5744379)

[Relación 15: Vendedor realiza venta 9](#_Toc5744380)

[Relación 16.1: Usuario hereda a administrador 10](#_Toc5744381)

[Relación 16.2: Usuario hereda a Vendedor 10](#_Toc5744382)

[Relación 16.3: Usuario hereda a Cliente 11](#_Toc5744383)

[Tablas 12](#_Toc5744384)

[Usuario 12](#_Toc5744385)

[Administrador 12](#_Toc5744386)

[Vendedor 12](#_Toc5744387)

[Cliente 12](#_Toc5744388)

[Nota\_crédito 12](#_Toc5744389)

[Venta 12](#_Toc5744390)

[Producto 12](#_Toc5744391)

[Caja 12](#_Toc5744392)

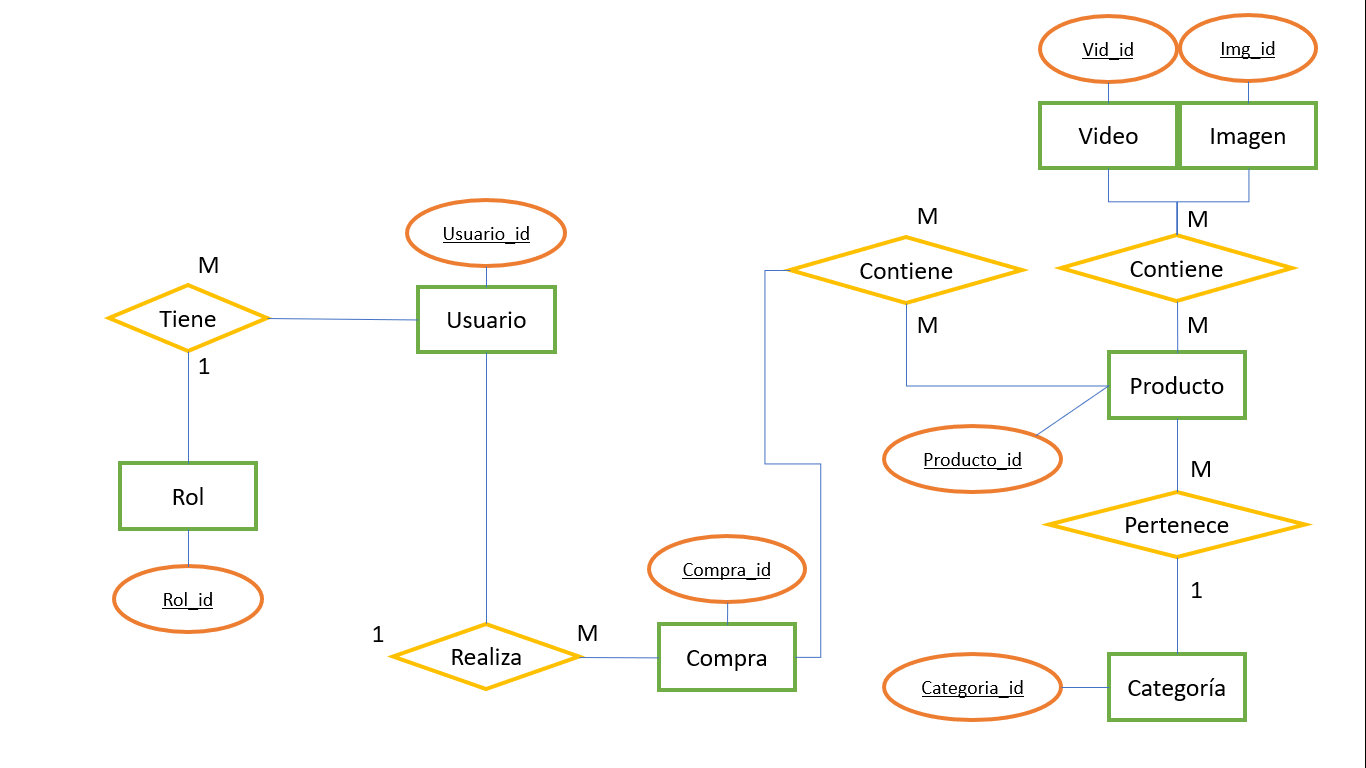
[Venta\_Producto 12](#_Toc5744393)

[Diagrama Relacional 13](#_Toc5744394)

[Normalización 14](#_Toc5744395)

# Modelo E-R y diagrama

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entidad | Atributos | Relación | Entidad con la que se relaciona | Cardinalidad |
| Usuario | Usuario\_id, nombre, contraseña, CURP, fecha\_nacimiento, correo, telefono | Tiene | Rol | M-1 |
| Realiza | Compra | 1-M |
| Maneja | Chat | M-M |
|  |  |  |  |  |
| Compra | Compra\_id, cantidad, subtotal, total, IVA, gran\_total, opción\_pago | Realiza | Usuario | M-1 |
| Tiene | Chat | 1-1 |
| Contiene | Producto | M-M |
|  |  |  |  |  |
| Rol | Rol\_id, nombre\_rol, descripción\_rol | Tiene | Usuario | 1-M |
|  |  |  |  |  |
| Producto | Producto\_id, nombre\_producto, descripción\_producto, precio, unidades\_existentes | Contiene | Compra | M-M |
| Pertenece | Categoría | M-1 |
| Contiene | Video | M-M |
| Contiene | Imagen | M-M |
|  |  |  |  |  |
| Categoría | Categoría\_id, nombre\_cat, descripción\_cat | Pertenece | Producto | 1-M |
|  |  |  |  |  |
| Imagen | Img\_id, dirección\_img | Contiene | Producto | M-M |
|  |  |  |  |  |
| Video | Vid\_id, dirección\_vid | Contiene | Producto | M-M |
|  |  |  |  |  |



# 

# Mapeo de datos

## Nombre de la base de datos: Eat\_DB

# Tablas Primas

## Usuario

(usuario\_id PK, nombre, contraseña, CURP, fecha\_nacimiento, correo, teléfono, rol\_id FK)

## Compra

(Compra\_id PK, cantidad, subtotal, total, IVA, gran\_total, opción\_pago, usuario\_id FK)

## Rol

(Rol\_id PK, nombre\_rol, descripción\_rol)

## Producto

(Producto\_id PK, nombre, descripción, precio, unidades, categoría\_id FK)

## Categoría

(categoría\_id PK, nombre, descripción)

## Imagen

(img\_id PK, dirección\_img)

## Video

(vid\_id FK, dirección\_vid)

# Tablas Asociativas

## Compra\_Producto

(código\_asociativo PK, compra\_id FK, producto\_id FK)

## Imagen\_Producto

(código\_asociativo PK, img\_id FK, producto\_id FK)

## Video\_Producto

(código\_asociativo PK, vid\_id FK, producto\_id FK)

# Diagrama Relacional

drop database if exists `eat\_db`;

create database `eat\_db`;

use `eat\_db`;

drop table if exists `ROL`;

create table `ROL` (

rol\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

nombre\_rol VARCHAR(30) NOT NULL,

descripcion\_rol VARCHAR(100) NOT NULL

);

drop table if exists `IMAGEN`;

create table `IMAGEN` (

img\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

direccion\_img VARCHAR(1000) NOT NULL

);

drop table if exists `VIDEO`;

create table `VIDEO` (

vid\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

direccion\_vid VARCHAR(1000) NOT NULL

);

drop table if exists `CATEGORIA`;

create table `CATEGORIA` (

categoria\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(30) NOT NULL,

descripcion VARCHAR(100) NOT NULL

);

drop table if exists `USUARIO`;

create table `USUARIO` (

usuario\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

contraseña VARCHAR(30) NOT NULL,

CURP VARCHAR(18) UNIQUE,

fecha\_nacimiento DATE,

correo VARCHAR(50) NOT NULL,

telefono INTEGER NOT NULL,

rol\_id INTEGER UNIQUE,

foreign key(rol\_id) references `ROL` (rol\_id)

);

drop table if exists `PRODUCTO`;

create table `PRODUCTO` (

producto\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(30) NOT NULL,

descripcion VARCHAR(100) NOT NULL,

precio INTEGER NOT NULL,

unidades INTEGER NOT NULL,

cat\_id INTEGER UNIQUE,

foreign key(cat\_id) references `CATEGORIA` (categoria\_id)

);

drop table if exists `COMPRA`;

create table `COMPRA` (

compra\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

cantidad INTEGER NOT NULL,

subtotal INTEGER NOT NULL,

total INTEGER NOT NULL,

IVA INTEGER NOT NULL,

gran\_total INTEGER NOT NULL,

opcionDePago VARCHAR(30) NOT NULL,

usuario\_id INTEGER UNIQUE,

foreign key(usuario\_id) references `USUARIO` (usuario\_id)

);

drop table if exists `COMPRA\_PRODUCTO`;

create table `COMPRA\_PRODUCTO` (

codigo\_asociativo INTEGER PRIMARY KEY AUTO\_INCREMENT,

compra\_id INTEGER UNIQUE,

producto\_id INTEGER UNIQUE,

foreign key(compra\_id) references `COMPRA` (compra\_id),

foreign key(producto\_id) references `PRODUCTO` (producto\_id)

);

drop table if exists `IMAGEN\_PRODUCTO`;

create table `IMAGEN\_PRODUCTO` (

codigo\_asociativo INTEGER PRIMARY KEY AUTO\_INCREMENT,

img\_id INTEGER UNIQUE,

producto\_id INTEGER UNIQUE,

foreign key(img\_id) references `IMAGEN` (img\_id),

foreign key(producto\_id) references `PRODUCTO` (producto\_id)

);

drop table if exists `VIDEO\_PRODUCTO`;

create table `VIDEO\_PRODUCTO` (

codigo\_asociativo INTEGER PRIMARY KEY AUTO\_INCREMENT,

vid\_id INTEGER UNIQUE,

producto\_id INTEGER UNIQUE,

foreign key(vid\_id) references `VIDEO` (vid\_id),

foreign key(producto\_id) references `PRODUCTO` (producto\_id)

);