

Crea Modelo Entidad Relación de Caso

Evidencia: GA6-220501096-AA1-EV02

Formación: Tecnología en Análisis y Desarrollo de Software

Aprendiz: Jefferson Harbey Mendez Castellanos

Instructor: John Alejandro Niño Tambo

Ficha: 2977395

Fecha: 28- 06 – 2025

Contenido

Crea Modelo Entidad Relación de Caso	1
Introducción	3
Objetivo.....	3
Análisis del modelo.....	3
Relaciones	8
Modelo ER.....	10
Conclusiones	11

Introducción

El objetivo de esta evidencia es desarrollar correctamente el modelo relacional de la base de datos de nuestro proyecto y asegurarnos de que se cumpla con los requerimientos. después de realizar el análisis respectivo a la evidencia anterior, en donde se aplico la cardinalidad y la normalización a nuestro modelo de base de datos. Se han definido las relaciones entre entidades de manera que se evite la redundancia de datos y dependencias innecesarias, esto nos garantiza la integridad de los datos de nuestro sistema y con ello una mayor escalabilidad de nuestro proyecto.

Objetivo

Crear el modelo relacional del software de gestión administrativa Matrona, de acuerdo a los requerimientos del proyecto y aplicando las reglas de cardinalidad y normalización hasta 3NF. Para el diseño se está utilizando MySQL Workbench, la cual es una potente herramienta visual que fue desarrollado por Oracle para modelar, desarrollar y administrar bases de datos MySQL.

Análisis del modelo

Se verifica el modelo relacional en busca de posibles errores e inconsistencias para garantizar la correcta cardinalidad entre las diferentes entidades de nuestro sistema.

Tabla que incluirá los roles de nuestro sistema, administrador/cliente/empleado.

rol
id_rol INT (PK)
nombre_rol VARCHAR

Tabla con todos los datos de los usuarios

usuarios
id_usuarios INT (PK)
id_rol INT (FK rol)
nombre VARCHAR
apellido VARCHAR
correo VARCHAR
contraseña VARCHAR
direccion VARCHAR

Tabla de clientes con las relaciones mediante FK de la tabla usuarios.

cliente
id_cliente INT (PK)
id_usuario INT (FK usuarios)
fecha_registro DATE

Tabla con el historial de compras del cliente y su relación con la tabla cliente

historial_compras
id_historial_compras INT (PK)
id_cliente INT (FK cliente)
fecha_compra DATE
tipo_cerveza VARCHAR
cantidad_comprada INT
total_pago DECIMAL(10,2)

Tabla con ID administrador y relación mediante FK de usuarios.

administrador
id_administrador INT (PK)
id_usuario INT (FK usuario)
Fecha_ingreso DATE

Tabla con los datos de los empleados y su relación con la tabla usuario mediante FK.

empleado
id_empleado INT (PK)
id_usuario INT (FK usuario)
fecha_contratacion DATE
salario DECIMAL(10,2)
fecha_pago DATE
area_laboral VARCHAR

Tabla que guardara los datos de los materiales que proveen los proveedores y su relación con la tabla usuario mediante FK para asegurar la vinculación de otros datos necesarios y así evitar duplicidad y redundancia.

proveedor
id_proveedor INT (PK)
id_usuario INT (FK usuario)
Material_que_provee VARCHAR
cantidadM INT

Tabla con los materiales para la producción y su vinculación mediante FK con el proveedor.

materiales
id_materiales INT (PK)
id_proveedor INT (FK proveedor)
tipo_material VARCHAR
cantidad_disponible VARCHAR

Tabla con los datos contables de la empresa y su vinculación con la entidad administrador.

contabilidad
id_contabilidad INT (PK)
id_administrador INT (FK administrador)
total_presupuesto DECIMAL(10,2)
ingresos DECIMAL(10,2)

Tabla con el inventario que incluye los diferentes datos de cada bebida de la empresa.

inventario
id_inventario INT (PK)
nombre_bebida VARCHAR
descripcion TEXT
ultimo_movimiento DATE
cantidad_disponible INT
unidades_agregadas INT

Tabla del catalogo de productos y su relación correcta con el inventario mediante FK.

catalogo
id_catalogo INT (PK)
id_inventario INT (FK inventario)
unidades_disponibles INT
sabor VARCHAR
alcohol VARCHAR
precio DECIMAL(10,2)

Entidad hija de catalogo que guarda los datos de cada cotización.

Cotizaciones
id_cotizaciones INT (PK)
Id_catalogo INT (FK catalogo)
cantidad_cotizado DECIMAL(10,2)
total_cotizado DECIMAL(10,2)
fecha_hora DATETIME

Entidad con los datos de los pedidos y su relación con el cliente mediante FK.

Pedidos
id_pedidos INT (PK)
id_cliente INT (FK cliente)
total_pedido DECIMAL(10,2)
fecha_pedido DATE

Tabla con los diferentes detalles de cada pedido de cerveza y su relación con el catalogo y la tabla pedidos mediante las FK.

Detalles pedido
id_detalle_pedido INT (PK)
id_pedidos INT (FK ventas)
id_catalogo INT (FK catalogo)
precio_unitario DECIMAL(10,2)
subtotal DECIMAL(10,2)
cantidad_pedido_uds INT

Relaciones

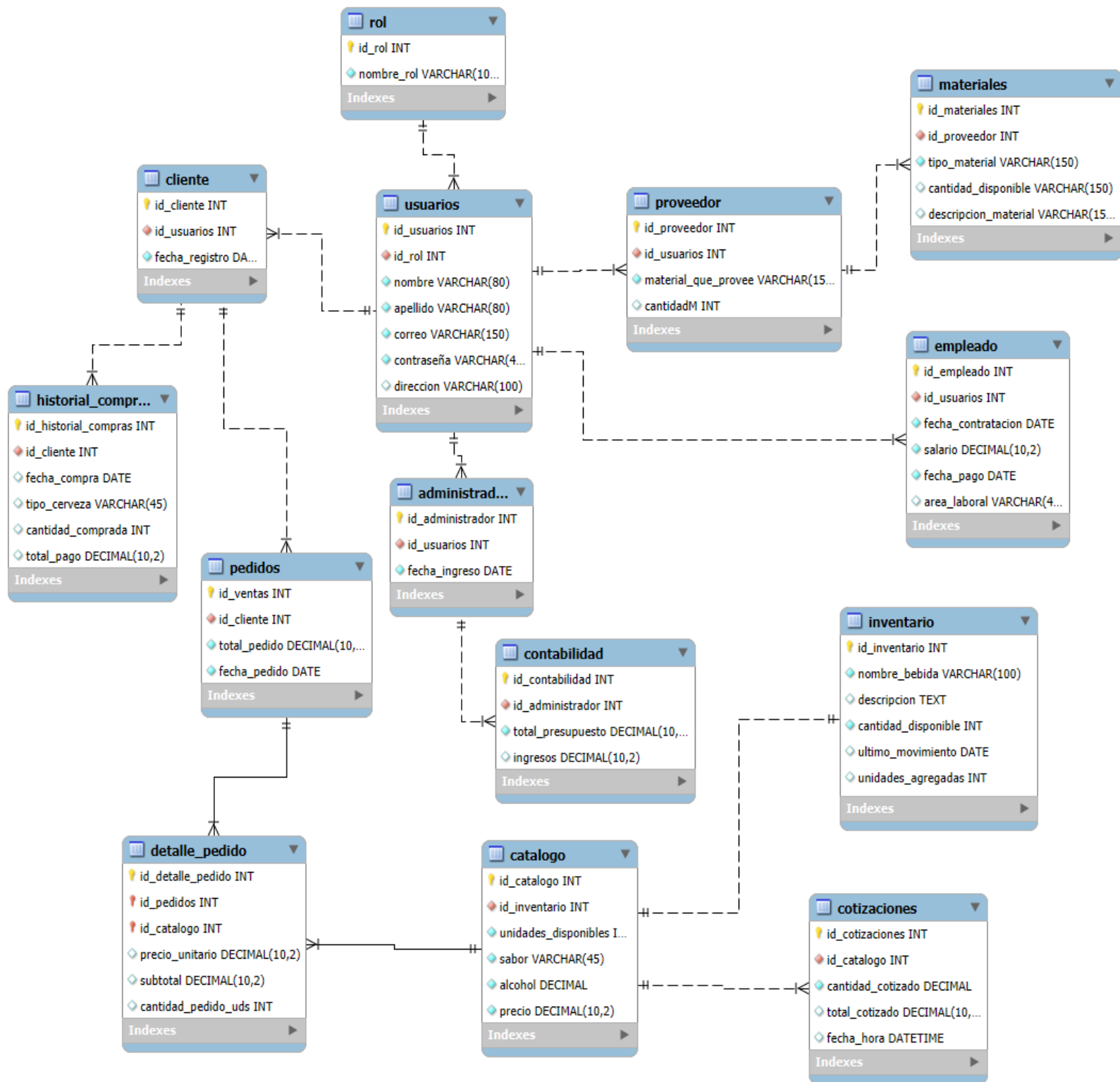
A continuación, se presentan las relaciones entre cada una de nuestras entidades.

- Cada rol puede tener muchos usuarios **1 : N** rol a usuarios.
- Un usuario puede ser varios administradores **1 : N** usuarios a administradores.
- Un usuario pueden ser muchos clientes **1 : N** usuarios a clientes.
- Un usuario pueden ser varios empleados **1 : N** usuarios a empleados.

- Un usuario pueden ser varios proveedores **1 : N** usuarios a proveedores.
- Un cliente puede tener varios historiales de compras **1 : N** cliente a historial de compras.
- Cada bebida en el inventario tiene una entrada correspondiente en el catálogo **1 : 1** inventario a catálogo.
- El catálogo puede generar muchas cotizaciones **1 : N** catálogo a cotizaciones.
- El catálogo procesa cada pedido y muchos pedidos son realizados en el catálogo **N : M** catálogo a pedidos
- Cada pedido tiene varios detalles **1 :N** pedidos a detalles de pedidos
- Un cliente puede realizar muchos pedidos **1 : N** cliente a pedidos

Modelo ER

A continuación nuestro modelo terminado de nuestra base de datos.



Conclusiones

Se ha creado el modelo de la base de datos de nuestro proyecto mediante la aplicación de las reglas de cardinalidad y normalización las cuales garantizan la coherencia, eficiencia e integridad de los datos. El modelo garantiza el correcto almacenamiento de los diferentes datos de la empresa como los usuarios del sistema, la información de los materiales y su proveedor, la información de los empleados, los datos del inventario de bebidas, el catálogo de bebidas con sus cotizaciones y diferente información de las ventas, el historial de compras de cada cliente etc. Se han relacionado cada una de las entidades mediante claves foráneas con un análisis profundo buscando garantizar la cardinalidad entre cada una de las relaciones, además, se definieron los diferentes tipos de datos que se utilizaran en cada uno de los atributos. Cada una de las acciones realizadas en conjunto con la aplicación de los conceptos aprendidos, nos permite el correcto diseño de la base de datos de nuestro software de gestión administrativa Matrona, el cual queda listo para ser implementado en el sistema relacional de gestión de bases de datos MySQL. Finalmente concluimos que es fundamental aplicar cada una de las reglas para el diseño relacional de una base de datos, tanto la cardinalidad como las reglas de normalización son vitales para la eficiencia de las consultas, la recuperación de datos, evitar relaciones incorrectas, reducir la redundancia de datos, optimizar el almacenamiento de los datos, evitar duplicidad entre datos etc. Cada uno de estos conceptos técnicos son muy importantes y siempre los debemos tener en cuenta en nuestro futuro como desarrolladores.