



Aprendiz: Rafael Dario Escalante Sandoval

**Equipo de proyecto cafetería:**

Saray Acosta

Cristian Mosquera Rodríguez

Danny Alexander Minota Soto

Instructor: Andrés Rubiano Cucarian

modelos conceptual y lógico para el proyecto desarrollo de software GA4-220501095-AA1-EV02.

ADSO (2977466)

**Centro de servicios financieros**

Contents

<b>Introducción .....</b>	<b>3</b>
<b>Modelo Conceptual .....</b>	<b>3</b>
<b>Normalización del Modelo Lógico .....</b>	<b>3</b>
<b>Conclusión .....</b>	<b>8</b>

## Introducción

Un modelo conceptual es una descripción general de cómo un sistema se organiza y opera. Su tarea principal es especificar y describir los conceptos base, entre ellos. Con base en la definición previa, y con la lectura del componente formativo conceptos generales de bases de datos se debe diseñar un modelo conceptual que abarque también la parte lógica del proyecto de desarrollo de software a realizar

## Modelo Conceptual

Un modelo conceptual de base de datos es una representación abstracta y simplificada de una base de datos que se utiliza para describir la estructura y la organización de los datos de una manera independiente de cualquier sistema de gestión de bases de datos (DBMS) específico. Este modelo se centra en la comprensión de los datos y las relaciones entre ellos, sin entrar en detalles técnicos de implementación. El modelo planteado para la solución de software de la Institución Educativa el Remolino es el Modelo de Entidad-Relación (ER). Este modelo utiliza entidades para representar objetos del mundo real y relaciones para definir las asociaciones entre estas entidades. Es ampliamente utilizado para diseñar bases de datos relacionales. A continuación, se detallan las entidades del modelo

## Normalización del Modelo Lógico

La **normalización** busca organizar las tablas para reducir la redundancia y evitar inconsistencias. Aquí te muestro cómo se podría normalizar el modelo de tu cafetería en tres formas normales (3NF):

### 1. Primera Forma Normal (1NF):

- Garantiza que cada columna contenga datos atómicos (no repetidos ni listas de valores).
- En este modelo, cada entidad (Clientes, Pedidos, Productos, etc.) ya tiene datos atómicos.

### 2. Segunda Forma Normal (2NF):

- Cada atributo que no sea clave primaria debe depender completamente de la clave primaria de la tabla.

En este caso, todas las tablas ya cumplen con esta forma porque cada columna depende completamente de la clave primaria en cada entidad.

### 3. Tercera Forma Normal (3NF):

- No debe haber dependencia transitiva (una columna no clave depende de otra columna no clave).
- Cada tabla cumple con esta forma, ya que no hay dependencias indirectas entre las columnas.
- Por lo tanto, el modelo ya estaría en 3NF, con tablas bien definidas y sin redundancias innecesarias.

#### Clientes

- **id\_cliente:** INT, PK, NOT NULL, UNIQUE. Identificador único del cliente.
- **nombre:** VARCHAR(50), NOT NULL. Nombre del cliente.
- **apellido:** VARCHAR(50), NOT NULL. Apellido del cliente.
- **teléfono:** VARCHAR(15). Teléfono de contacto del cliente.
- **dirección:** VARCHAR(100). Dirección de entrega del cliente.
- **email:** VARCHAR(50), UNIQUE. Correo electrónico del cliente.

#### Productos

- **id\_producto:** INT, PK, NOT NULL, UNIQUE. Identificador único del producto.
- **nombre\_producto:** VARCHAR(50), NOT NULL. Nombre del producto.
- **descripción:** TEXT. Descripción del producto.
- **precio:** DECIMAL(10,2), NOT NULL. Precio de venta del producto.
- **disponibilidad:** BOOLEAN, DEFAULT TRUE. Indica si el producto está en stock.

#### Pedidos

- **id\_pedido:** INT, PK, NOT NULL, UNIQUE. Identificador único del pedido.
- **id\_cliente:** INT, FK, NOT NULL. Referencia al cliente que realizó el pedido.

- **fecha\_pedido:** DATE, NOT NULL. Fecha en la que se hizo el pedido.
- **hora\_pedido:** TIME. Hora en que se realizó el pedido.
- **tipo\_entrega:** ENUM('local', 'domicilio'), NOT NULL. Tipo de entrega del pedido.
- **estado:** ENUM('pendiente', 'preparacion', 'entregado', 'cancelado'). Estado actual del pedido.

### Detalles\_Pedido

- **id\_detalle:** INT, PK, NOT NULL, UNIQUE. Identificador único del detalle del pedido.
- **id\_pedido:** INT, FK, NOT NULL. Referencia al pedido asociado.
- **id\_producto:** INT, FK, NOT NULL. Referencia al producto en el pedido.
- **cantidad:** INT, NOT NULL, CHECK > 0. Cantidad de producto en el pedido.
- **subtotal:** DECIMAL(10,2), NOT NULL. Subtotal del producto en el pedido.

### Pagos

- **id\_pago:** INT, PK, NOT NULL, UNIQUE. Identificador único del pago.
- **id\_pedido:** INT, FK, NOT NULL. Referencia al pedido relacionado con el pago.
- **monto:** DECIMAL(10,2), NOT NULL. Monto pagado.
- **metodo\_pago:** ENUM('efectivo', 'tarjeta', 'Nequi'). Método de pago utilizado.
- **fecha\_pago:** DATE. Fecha en la que se registró el pago.
- **estado\_pago:** ENUM('completado', 'pendiente'). Estado actual del pago.

### Repartidores

- **id\_repartidor:** INT, PK, NOT NULL, UNIQUE. Identificador único del repartidor.
- **nombre\_repartidor:** VARCHAR(50), NOT NULL. Nombre del repartidor.
- **telefono:** VARCHAR(15). Teléfono de contacto del repartidor.

- **disponibilidad:** BOOLEAN, DEFAULT TRUE. Indica si el repartidor está disponible.

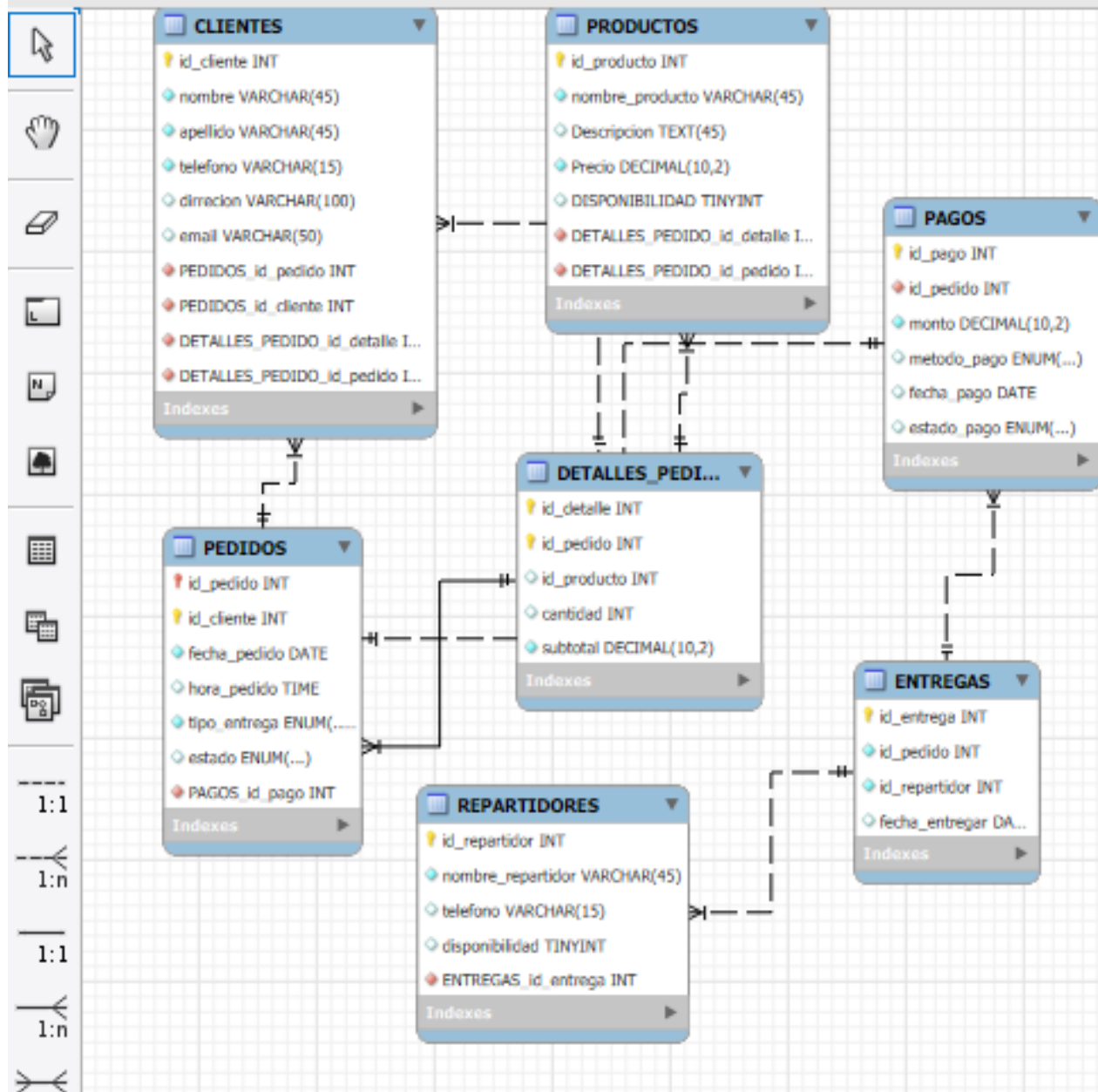
## Entregas

- **id\_entrega:** INT, PK, NOT NULL, UNIQUE. Identificador único de la entrega.
- **id\_pedido:** INT, FK, NOT NULL. Referencia al pedido que se entregará.
- **id\_repartidor:** INT, FK, NOT NULL. Repartidor asignado a la entrega.
- **fecha\_entrega:** DATE. Fecha de la entrega.
- **hora\_entrega:** TIME. Hora de la entrega.
- **estado\_entrega:** ENUM('en camino', 'entregado'). Estado actual de la entrega.

---

Cada atributo tiene especificaciones de tipo de dato, restricciones y una descripción clara para guiar el uso adecuado de la información. Este formato te permite mantener un registro organizado sin necesidad de tablas formales en tu documento.

## Diagram



## **Conclusión**

En el escenario cambiante y dinámico de la programación y la tecnología, MySQL Workbench destaca como un director de orquesta hábil para la gestión y optimización de bases de datos MySQL. Su interfaz intuitiva, funciones potentes y capacidad para simplificar tareas complejas lo convierten en una herramienta esencial en el kit de herramientas de cualquier profesional de bases de datos. Al aprovechar las funcionalidades de MySQL Workbench, los desarrolladores y administradores pueden navegar por el complejo mundo de las bases de datos con confianza y eficacia.

## **Fuentes**

<https://keepcoding.io/blog/que-es-mysql-workbench/>

<https://www.ibm.com/docs/es/imdm/11.6?topic=concepts-key-entity-attribute-entity-type>