Diseño de la Base de Datos

1. Introducción

Este documento detalla el diseño de la base de datos para la aplicación de gestión de pedidos y facturación. Se ha optado por un sistema de gestión de bases de datos relacional (RDBMS), específicamente MySQL, debido a su robustez, escalabilidad y amplia adopción en el mercado. El diseño se enfoca en la normalización para asegurar la integridad de los datos, minimizar la redundancia y optimizar el rendimiento de las consultas.

2. Modelo Entidad-Relación (MER)

El siguiente diagrama de entidad-relación (MER) representa las entidades principales de la aplicación y las relaciones entre ellas. Este modelo sirve como base para la creación del esquema de la base de datos.

USUARIOS ||--o{ CLIENTES : "tiene"

CLIENTES ||--o{ PEDIDOS : "realiza"

PEDIDOS ||--o{ LINEA\_PEDIDO : "contiene"

PRODUCTOS ||--o{ LINEA\_PEDIDO : "incluye"

PEDIDOS ||--o{ FACTURAS : "genera"

FACTURAS ||--o{ LINEA\_FACTURA : "contiene"

USUARIOS {

VARCHAR id\_usuario PK

VARCHAR username

VARCHAR password\_hash

VARCHAR rol

}

CLIENTES {

VARCHAR id\_cliente PK

VARCHAR nombre

VARCHAR apellidos

VARCHAR direccion

VARCHAR telefono

VARCHAR email

VARCHAR id\_usuario FK

}

PRODUCTOS {

VARCHAR id\_producto PK

VARCHAR nombre

TEXT descripcion

DECIMAL precio

VARCHAR ruta\_imagen

}

PEDIDOS {

VARCHAR id\_pedido PK

DATETIME fecha\_pedido

VARCHAR estado

DECIMAL total

VARCHAR id\_cliente FK

}

LINEA\_PEDIDO {

VARCHAR id\_linea\_pedido PK

VARCHAR id\_pedido FK

VARCHAR id\_producto FK

INT cantidad

DECIMAL precio\_unitario

DECIMAL subtotal

}

FACTURAS {

VARCHAR id\_factura PK

DATETIME fecha\_factura

DECIMAL importe\_total

VARCHAR id\_pedido FK

}

LINEA\_FACTURA {

VARCHAR id\_linea\_factura PK

VARCHAR id\_factura FK

VARCHAR descripcion

INT cantidad

DECIMAL precio\_unitario

DECIMAL subtotal

}

3. Descripción de Tablas y Campos

A continuación, se describe cada una de las tablas, sus campos, tipos de datos y restricciones.

3.1. Tabla `USUARIOS`

Almacena la información de los usuarios que pueden acceder a la aplicación.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :-------------- | :-------------- | :------------------- | :---------------------------------------- |

| `id\_usuario` | `VARCHAR(36)` | `PRIMARY KEY` | Identificador único del usuario (UUID). |

| `username` | `VARCHAR(50)` | `NOT NULL`, `UNIQUE` | Nombre de usuario para el login. |

| `password\_hash` | `VARCHAR(255)` | `NOT NULL` | Hash seguro de la contraseña del usuario. |

| `rol` | `VARCHAR(20)` | `NOT NULL` | Rol del usuario (ej. 'cliente', 'admin'). |

3.2. Tabla `CLIENTES`

Contiene la información detallada de los clientes de la tienda.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :------------ | :-------------- | :------------------- | :---------------------------------------- |\n| `id\_cliente` | `VARCHAR(36)` | `PRIMARY KEY` | Identificador único del cliente (UUID). |

| `nombre` | `VARCHAR(100)` | `NOT NULL` | Nombre del cliente. |

| `apellidos` | `VARCHAR(100)` | `NOT NULL` | Apellidos del cliente. |

| `direccion` | `VARCHAR(255)` | `NOT NULL` | Dirección de envío/facturación. |

| `telefono` | `VARCHAR(20)` | `NULLABLE` | Número de teléfono del cliente. |

| `email` | `VARCHAR(100)` | `NOT NULL`, `UNIQUE` | Correo electrónico del cliente. |

| `id\_usuario` | `VARCHAR(36)` | `FOREIGN KEY` | Clave foránea a `USUARIOS`. |

3.3. Tabla `PRODUCTOS`

Almacena los detalles de los productos disponibles en la tienda.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :------------ | :-------------- | :------------------- | :---------------------------------------- |

| `id\_producto` | `VARCHAR(36)` | `PRIMARY KEY` | Identificador único del producto (UUID). |

| `nombre` | `VARCHAR(255)` | `NOT NULL`, `UNIQUE` | Nombre del producto. |

| `descripcion` | `TEXT` | `NULLABLE` | Descripción detallada del producto. |

| `precio` | `DECIMAL(10, 2)`| `NOT NULL`, `> 0` | Precio unitario del producto. |

| `ruta\_imagen` | `VARCHAR(255)` | `NULLABLE` | Ruta del archivo de imagen del producto. |

3.4. Tabla `PEDIDOS`

Registra los pedidos realizados por los clientes.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :------------ | :-------------- | :------------------- | :---------------------------------------- |

| `id\_pedido` | `VARCHAR(36)` | `PRIMARY KEY` | Identificador único del pedido (UUID). |

| `fecha\_pedido`| `DATETIME` | `NOT NULL` | Fecha y hora en que se realizó el pedido. |

| `estado` | `VARCHAR(50)` | `NOT NULL` | Estado actual del pedido (ej. 'creado', 'en transporte', 'entregado'). |

| `total` | `DECIMAL(10, 2)`| `NOT NULL`, `>= 0` | Importe total del pedido. |

| `id\_cliente` | `VARCHAR(36)` | `FOREIGN KEY` | Clave foránea a `CLIENTES`. |

3.5. Tabla `LINEA\_PEDIDO`

Detalla los productos incluidos en cada pedido.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :---------------- | :-------------- | :------------------- | :---------------------------------------- |

| `id\_linea\_pedido` | `VARCHAR(36)` | `PRIMARY KEY` | Identificador único de la línea de pedido (UUID). |

| `id\_pedido` | `VARCHAR(36)` | `FOREIGN KEY` | Clave foránea a `PEDIDOS`. |

| `id\_producto` | `VARCHAR(36)` | `FOREIGN KEY` | Clave foránea a `PRODUCTOS`. |

| `cantidad` | `INT` | `NOT NULL`, `> 0` | Cantidad del producto en esta línea. |

| `precio\_unitario` | `DECIMAL(10, 2)`| `NOT NULL`, `> 0` | Precio unitario del producto en el momento del pedido. |

| `subtotal` | `DECIMAL(10, 2)`| `NOT NULL`, `>= 0` | Subtotal de la línea (`cantidad \* precio\_unitario`). |

3.6. Tabla `FACTURAS`

Almacena la información de las facturas generadas.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :-------------- | :-------------- | :------------------- | :---------------------------------------- |

| `id\_factura` | `VARCHAR(36)` | `PRIMARY KEY` | Identificador único de la factura (UUID). |

| `fecha\_factura` | `DATETIME` | `NOT NULL` | Fecha y hora de emisión de la factura. |

| `importe\_total` | `DECIMAL(10, 2)`| `NOT NULL`, `>= 0` | Importe total de la factura. |

| `id\_pedido` | `VARCHAR(36)` | `FOREIGN KEY`, `UNIQUE` | Clave foránea a `PEDIDOS`. Una factura por pedido. |

3.7. Tabla `LINEA\_FACTURA`

Detalla los ítems incluidos en cada factura.

| Campo | Tipo de Dato | Restricciones | Descripción |

| :---------------- | :-------------- | :------------------- | :---------------------------------------- |

| `id\_linea\_factura`| `VARCHAR(36)` | `PRIMARY KEY` | Identificador único de la línea de factura (UUID). |

| `id\_factura` | `VARCHAR(36)` | `FOREIGN KEY` | Clave foránea a `FACTURAS`. |

| `descripcion` | `VARCHAR(255)` | `NOT NULL` | Descripción del ítem facturado. |

| `cantidad` | `INT` | `NOT NULL`, `> 0` | Cantidad del ítem. |

| `precio\_unitario` | `DECIMAL(10, 2)`| `NOT NULL`, `> 0` | Precio unitario del ítem. |

| `subtotal` | `DECIMAL(10, 2)`| `NOT NULL`, `>= 0` | Subtotal de la línea (`cantidad \* precio\_unitario`). |

4. Relaciones entre Tablas

Las relaciones entre las tablas se establecen mediante claves foráneas, asegurando la integridad referencial.

`USUARIOS` a `CLIENTES` (Uno a Uno): Un usuario puede estar asociado a un único cliente. Esto permite que un usuario autenticado tenga un perfil de cliente. (`CLIENTES.id\_usuario` referencia `USUARIOS.id\_usuario`).

`CLIENTES` a `PEDIDOS` (Uno a Muchos): Un cliente puede realizar múltiples pedidos. (`PEDIDOS.id\_cliente` referencia `CLIENTES.id\_cliente`).

`PEDIDOS` a `LINEA\_PEDIDO` (Uno a Muchos): Un pedido puede contener múltiples líneas de pedido. (`LINEA\_PEDIDO.id\_pedido` referencia `PEDIDOS.id\_pedido`).

`PRODUCTOS` a `LINEA\_PEDIDO` (Uno a Muchos): Un producto puede aparecer en múltiples líneas de pedido. (`LINEA\_PEDIDO.id\_producto` referencia `PRODUCTOS.id\_producto`).

`PEDIDOS` a `FACTURAS` (Uno a Uno): Un pedido genera una única factura. (`FACTURAS.id\_pedido` referencia `PEDIDOS.id\_pedido`).

`FACTURAS` a `LINEA\_FACTURA` (Uno a Muchos): Una factura puede contener múltiples líneas de factura. (`LINEA\_FACTURA.id\_factura` referencia `FACTURAS.id\_factura`).

5. Script SQL para Creación de la Base de Datos

Se generará un script SQL (`schema.sql`) que contendrá las sentencias `CREATE TABLE` para todas las tablas, incluyendo la definición de claves primarias, claves foráneas e índices. Este script permitirá recrear la estructura de la base de datos fácilmente.

sql

Ejemplo de estructura (el script completo se generará en la fase de desarrollo)

CREATE TABLE USUARIOS (

id\_usuario VARCHAR(36) PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password\_hash VARCHAR(255) NOT NULL,

rol VARCHAR(20) NOT NULL

);

CREATE TABLE CLIENTES (

id\_cliente VARCHAR(36) PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

apellidos VARCHAR(100) NOT NULL,

direccion VARCHAR(255) NOT NULL,

telefono VARCHAR(20),

email VARCHAR(100) NOT NULL UNIQUE,

id\_usuario VARCHAR(36),

FOREIGN KEY (id\_usuario) REFERENCES USUARIOS(id\_usuario)

);

CREATE TABLE PRODUCTOS (

id\_producto VARCHAR(36) PRIMARY KEY,

nombre VARCHAR(255) NOT NULL UNIQUE,

descripcion TEXT,

precio DECIMAL(10, 2) NOT NULL CHECK (precio > 0),

ruta\_imagen VARCHAR(255)

);

CREATE TABLE PEDIDOS (

id\_pedido VARCHAR(36) PRIMARY KEY,

fecha\_pedido DATETIME NOT NULL,

estado VARCHAR(50) NOT NULL,

total DECIMAL(10, 2) NOT NULL CHECK (total >= 0),

id\_cliente VARCHAR(36),

FOREIGN KEY (id\_cliente) REFERENCES CLIENTES(id\_cliente)

);

CREATE TABLE LINEA\_PEDIDO (

id\_linea\_pedido VARCHAR(36) PRIMARY KEY,

id\_pedido VARCHAR(36),

id\_producto VARCHAR(36),

cantidad INT NOT NULL CHECK (cantidad > 0),

precio\_unitario DECIMAL(10, 2) NOT NULL CHECK (precio\_unitario > 0),

subtotal DECIMAL(10, 2) NOT NULL CHECK (subtotal >= 0),

FOREIGN KEY (id\_pedido) REFERENCES PEDIDOS(id\_pedido),

FOREIGN KEY (id\_producto) REFERENCES PRODUCTOS(id\_producto)

);

CREATE TABLE FACTURAS (

id\_factura VARCHAR(36) PRIMARY KEY,

fecha\_factura DATETIME NOT NULL,

importe\_total DECIMAL(10, 2) NOT NULL CHECK (importe\_total >= 0),

id\_pedido VARCHAR(36) UNIQUE,

FOREIGN KEY (id\_pedido) REFERENCES PEDIDOS(id\_pedido)

);

CREATE TABLE LINEA\_FACTURA (

id\_linea\_factura VARCHAR(36) PRIMARY KEY,

id\_factura VARCHAR(36),

descripcion VARCHAR(255) NOT NULL,

cantidad INT NOT NULL CHECK (cantidad > 0),

precio\_unitario DECIMAL(10, 2) NOT NULL CHECK (precio\_unitario > 0),

subtotal DECIMAL(10, 2) NOT NULL CHECK (subtotal >= 0),

FOREIGN KEY (id\_factura) REFERENCES FACTURAS(id\_factura)

);

6. Consideraciones Adicionales

Índices: Se crearán índices en las columnas utilizadas frecuentemente en cláusulas `WHERE` y `JOIN` para optimizar el rendimiento de las consultas.

Tipos de Datos: Los tipos de datos se han seleccionado para optimizar el almacenamiento y asegurar la precisión. Se utilizarán `VARCHAR(36)` para los IDs para permitir el uso de UUIDs, lo que facilita la generación de IDs únicos distribuidos.

Restricciones: Se han definido restricciones `NOT NULL` y `CHECK` para mantener la integridad de los datos.

Normalización: El diseño sigue los principios de la tercera forma normal (3NF) para minimizar la redundancia y evitar anomalías de actualización.