



# MANUAL TECNICO

El Buen Samaritano

Saúl André Cerezo Taracena

201900285

El presente documento describe los aspectos técnicos informáticos del sistema de información. El documento familiariza al personal técnico especializado encargado de las actividades de mantenimiento, revisión, solución de problemas, instalación y configuración del sistema.

### **Información Destacada**

El manual técnico proporciona la información necesaria para guiar al personal en la concepción, diseño, análisis, programación e instalación del sistema. Es importante destacar que la redacción del manual técnico está dirigida a personas con conocimientos en sistemas y tecnologías de la información, así como experiencia en programación intermedia/avanzada en Python y desarrollo en el entorno del lenguaje de programación **MySQL**.

### **descripción de Modelos implementado:**

Se aclara que mediante el uso de **DataModeler** se basó la creación de las estructuras de la base de datos, a continuación, se mostrara a más detalle dichas estructuras.

1. Para saber en que base de datos estamos y saber que estamos aplicando código en dicha base, necesitamos hacer el uso de los siguientes comandos:

```
1 • CREATE DATABASE proyectouno;  
2 • USE proyectouno;
```

- **CREATE DATABASE** proyectouno, con este comando se creo la base de datos .
- **USE** proyectouno, con este comando hacemos uso de la base de datos en cuestión.

## 2. Explicación para cada una de las estructuras:

- Estructura para **categoría**:

```
• CREATE TABLE categoria(  
    id_categoria INTEGER NOT NULL,  
    nombre VARCHAR(25) NOT NULL  
);  
• ALTER TABLE categoria ADD CONSTRAINT categoria_pk PRIMARY KEY (id_categoria);
```

Se crea una tabla llamada "**categoría**" con dos columnas: "id\_categoria" y "nombre". La columna "id\_categoria" es de tipo INTEGER y no permite valores nulos (NOT NULL), mientras que la columna "nombre" es de tipo VARCHAR con una longitud máxima de 25 caracteres y también no permite valores nulos.

se utiliza ALTER TABLE para añadir una restricción de clave primaria (PRIMARY KEY) a la columna "id\_categoria" en la tabla "categoria" significa que la columna "id\_categoria" será única para cada fila en la tabla y será utilizada para identificar de manera única cada registro en la tabla.

- Estructura para **país**:

```
9 • CREATE TABLE pais(  
10     id_pais INTEGER NOT NULL,  
11     nombre VARCHAR(50) NOT NULL  
12 );  
13 • ALTER TABLE pais ADD CONSTRAINT pais_pk PRIMARY KEY (id_pais);
```

Se crea una tabla llamada "pais" con dos columnas: "id\_pais" y "nombre". La columna "id\_pais" es de tipo INTEGER y no permite valores nulos (NOT NULL), mientras que la columna "nombre" es de tipo VARCHAR con una longitud máxima de 50 caracteres y también no permite valores nulos.

Se añade una restricción de clave primaria (PRIMARY KEY) a la columna "id\_pais" en la tabla "pais". Esto significa que la columna "id\_pais" será única para cada fila en la tabla y será utilizada como clave primaria para identificar de manera única cada registro en la tabla.

- Estructura para **producto**:

```

15 • CREATE TABLE producto(
16     id_producto INTEGER NOT NULL,
17     nombre VARCHAR(50) NOT NULL,
18     precio NUMERIC(6,2) NOT NULL,
19     categoria_id_categoria INTEGER NOT NULL
20 );
21 • ALTER TABLE producto ADD CONSTRAINT producto_pk PRIMARY KEY (id_producto);
22 • ALTER TABLE producto ADD CONSTRAINT producto_categoria_fk FOREIGN KEY
23     (categoria_id_categoria) REFERENCES categoria (id_categoria);

```

**categoria\_id\_categoria:** Es de tipo INTEGER y no permite valores nulos (NOT NULL). Esta columna será utilizada como clave externa para establecer una relación con la tabla "categoria", lo que significa que contendrá valores que hacen referencia a la columna "id\_categoria" de la tabla "categoria".

Se le añade una restricción de clave externa (FOREIGN KEY) a la columna "categoria\_id\_categoria" en la tabla "producto", que hace referencia a la columna "id\_categoria" en la tabla "categoria". Esto establece una relación entre las dos tablas, asegurando que cada valor en "categoria\_id\_categoria" en la tabla "producto" coincida con un valor existente en la columna "id\_categoria" en la tabla "categoria".

- Estructura de cliente:

```

41 • CREATE TABLE vendedor(
42     id_vendedor INTEGER NOT NULL,
43     nombre VARCHAR(100) NOT NULL,
44     pais_id_pais INTEGER NOT NULL
45 );
46 • ALTER TABLE vendedor ADD CONSTRAINT vendedor_pk PRIMARY KEY (id_vendedor);
47 • ALTER TABLE vendedor ADD CONSTRAINT vendedor_pais_fk FOREIGN KEY
48     (pais_id_pais) REFERENCES pais (id_pais);

```

**pais\_id\_pais:** Es de tipo INTEGER y no permite valores nulos (NOT NULL). Esta columna será utilizada como clave externa para establecer una relación con la tabla "pais", lo que significa que

Se añade una restricción de clave externa (FOREIGN KEY) a la columna "pais\_id\_pais" en la tabla "vendedor", que hace referencia a la columna "id\_pais" en la tabla "pais". Esto establece una relación entre las dos

contendrá valores que hacen referencia a la columna "id\_pais" de la tabla "pais".

tablas, asegurando que cada valor en "pais\_id\_pais" en la tabla "vendedor" coincida con un valor existente en la columna "id\_pais" en la tabla "pais".

- Estructura de **orden**:

```
49 • CREATE TABLE orden(  
50     id_orden INTEGER AUTO_INCREMENT NOT NULL,  
51     linea_orden INTEGER NOT NULL,  
52     cantidad INTEGER NOT NULL,  
53     vendedor_id_vendedor INTEGER NOT NULL,  
54     producto_id_producto INTEGER NOT NULL  
55 );  
56 • ALTER TABLE orden ADD CONSTRAINT orden_pk PRIMARY KEY (id_orden);  
57 • ALTER TABLE orden ADD CONSTRAINT orden_compra_fk FOREIGN KEY  
58     (vendedor_id_vendedor) REFERENCES vendedor (id_vendedor);  
59 • ALTER TABLE orden ADD CONSTRAINT orden_producto_fk FOREIGN KEY  
60     (producto_id_producto) REFERENCES producto (id_producto);
```

cantidad: Es de tipo INTEGER y no permite valores nulos (NOT NULL). Esta columna almacenará la cantidad de productos en cada orden.

vendedor\_id\_vendedor: Es de tipo INTEGER y no permite valores nulos (NOT NULL). Esta columna será utilizada como clave externa para establecer una relación con la tabla "vendedor", lo que significa que contendrá valores que hacen referencia a la columna "id\_vendedor" de la tabla "vendedor".

producto\_id\_producto: Es de tipo INTEGER y no permite valores nulos (NOT NULL). Esta columna será utilizada como clave externa para establecer una relación con la tabla "producto", lo que significa que contendrá valores que hacen referencia a la columna "id\_producto" de la tabla "producto".

**ALTER TABLE orden ADD CONSTRAINT orden\_compra\_fk FOREIGN KEY (vendedor\_id\_vendedor) REFERENCES vendedor (id\_vendedor):** Añade una restricción de clave externa (FOREIGN KEY) a la columna "vendedor\_id\_vendedor" en la tabla "orden", que hace referencia a la columna "id\_vendedor" en la tabla "vendedor". Esto establece una relación entre las dos tablas, asegurando que cada valor en "vendedor\_id\_vendedor" en la tabla "orden" coincida con un valor existente en la columna "id\_vendedor" en la tabla "vendedor".

**ALTER TABLE orden ADD CONSTRAINT orden\_producto\_fk FOREIGN KEY (producto\_id\_producto) REFERENCES producto (id\_producto):** Añade una restricción de clave externa (FOREIGN KEY) a la columna "producto\_id\_producto" en la tabla "orden", que hace referencia a la columna "id\_producto" en la tabla "producto". Esto establece una relación

entre las dos tablas, asegurando que cada valor en "producto\_id\_producto" en la tabla "orden" coincida con un valor existente en la columna "id\_producto" en la tabla "producto".

- Estructura para **detalle**: esta es la tabla que se utiliza para hacer la primera forma normal, se utiliza para filtrar información de la tabla **orden**

```

61 • CREATE TABLE detalle (
62     id_detalle SERIAL PRIMARY KEY,
63     fecha_orden VARCHAR(10) NOT NULL,
64     cliente_id_cliente INTEGER NOT NULL,
65     orden_id_orden INTEGER NOT NULL,
66     CONSTRAINT unique_orden_cliente_fecha UNIQUE (orden_id_orden,
67     fecha_orden, cliente_id_cliente)
68 );
69 • ALTER TABLE detalle ADD CONSTRAINT detalle_pk PRIMARY KEY (id_detalle);
70 • ALTER TABLE detalle ADD CONSTRAINT detalle_cliente_fk FOREIGN KEY
71     (cliente_id_cliente) REFERENCES cliente (id_cliente);
72 • ALTER TABLE detalle ADD CONSTRAINT detalle_orden_fk FOREIGN KEY
73     (orden_id_orden) REFERENCES orden (id_orden);

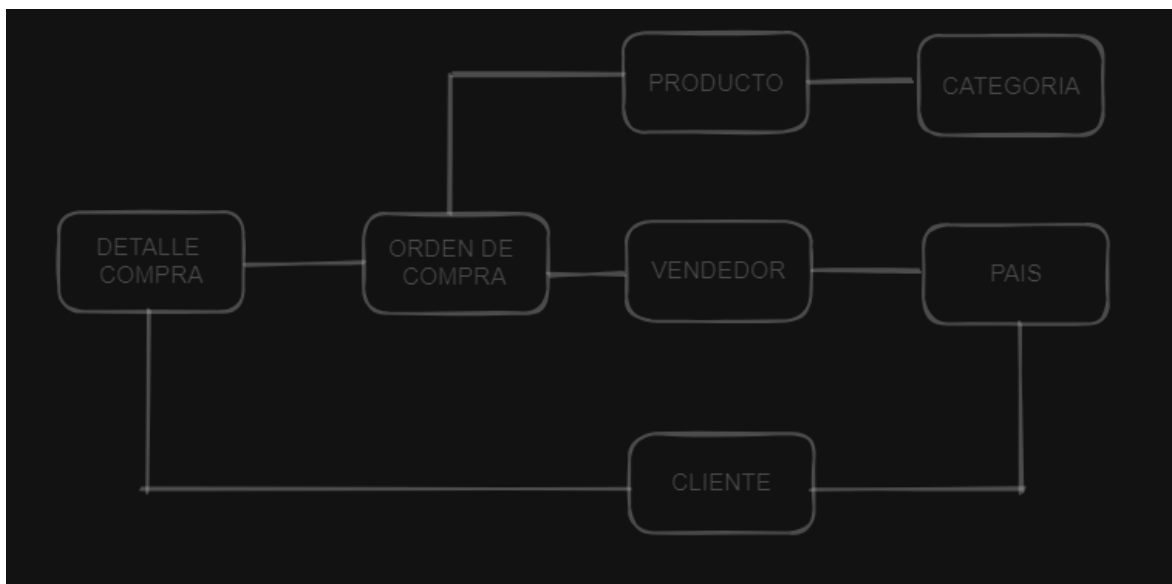
```

fecha\_orden: Es de tipo VARCHAR con una longitud máxima de 10 caracteres y no permite valores nulos (NOT NULL). Esta columna almacena la fecha de la orden.

cliente\_id\_cliente: Es de tipo INTEGER y no permite valores nulos (NOT NULL). Esta columna se utiliza como clave externa para establecer una relación con la tabla "cliente", lo que significa que contendrá valores que hacen referencia a la columna "id\_cliente" de la tabla "cliente".

**ALTER TABLE detalle ADD CONSTRAINT detalle\_orden\_fk FOREIGN KEY (orden\_id\_orden) REFERENCES orden (id\_orden):** Añade una restricción de clave externa (FOREIGN KEY) a la columna "orden\_id\_orden" en la tabla "detalle", que hace referencia a la columna "id\_orden" en la tabla "orden". Esto establece una relación entre las dos tablas, asegurando que cada valor en "orden\_id\_orden" en la tabla "detalle" coincida con un valor existente en la columna "id\_orden" en la tabla "orden".

Para la mejor comprensión de la estructura y las relaciones de las entidades se proporciona la siguiente imagen **Modelo Conceptual**



Dentro de esta misma carpeta se encuentra el **Modelo lógico y Modelo físico**, para que usted como programador pueda entender mas a profundidad el funcionamiento correcto de esta base de datos.

Esperamos que este manual haya sido de gran ayuda para el entendimiento del uso de la base de datos. Nuestro objetivo ha sido proporcionarte las herramientas y el conocimiento necesarios para optimizar tus procesos, resolver desafíos técnicos y llevar tus proyectos al siguiente nivel.

¡Hasta pronto, y que tus futuros proyectos estén llenos de logros y satisfacciones!"