



MANUAL DE BASE DE DATOS

INTEGRANTES

Jhon Alexander Granados Ibarra
Brando Alexander Lopez Benavides
Andres Camilo Morales Chaves
Brigith Katherine Córdoba de la Cruz



Tabla de Contenido

Contenido

Tabla de Contenido	2
Introducción	3
Información Destacada.....	3
Objetivos	3
Requerimientos del Sistema	3
Introducción	3
Información Destacada.....	3
Objetivos	3
Requerimientos del Sistema	3
.....	4
Base de Datos	4
Tabla de Usuario	5
Tabla Categoría	5
Tabla Productos.....	6
Tabla Pedidos	7
.....	7
Tabla líneas_pedidos	7
Consultas	8
Consultas Usuarios.....	9
Consultas Productos	9
Consultas Categorías.....	10
Consultas Pedidos	11
Triggers.....	12



Introducción

Este documento describirá como esta creada la base de datos del proyecto de la tienda online, cuál será su finalidad, y como es su funcionamiento e instalación. Con esto cualquier persona tendrá la facilidad de manejar la base de datos.

Información Destacada

Este manual hace referencia a la información necesaria con el fin de orientar en cómo está desarrollado este proyecto. Está orientado a las personas con conocimientos en sistemas y tecnologías de información, y administración de bases de datos.

Objetivos

Instruir el uso adecuado de la base de datos, para el acceso oportuno y adecuado en la instalación del mismo, mostrando los pasos a seguir en el proceso de instalación, así como la descripción de las tablas y sus consultas.

Requerimientos del Sistema

Puede ser instalado en cualquier sistema operativo que cumpla con los siguientes requerimientos:

- Servidor Web Apache 2.4.35 o Superior
- MySQL 5.7.24 o Superior
- Gestor de base de datos

Base de Datos

Diagrama entidad relación:

A continuación, se muestra el diagrama relacional de las tablas que componen el sistema:

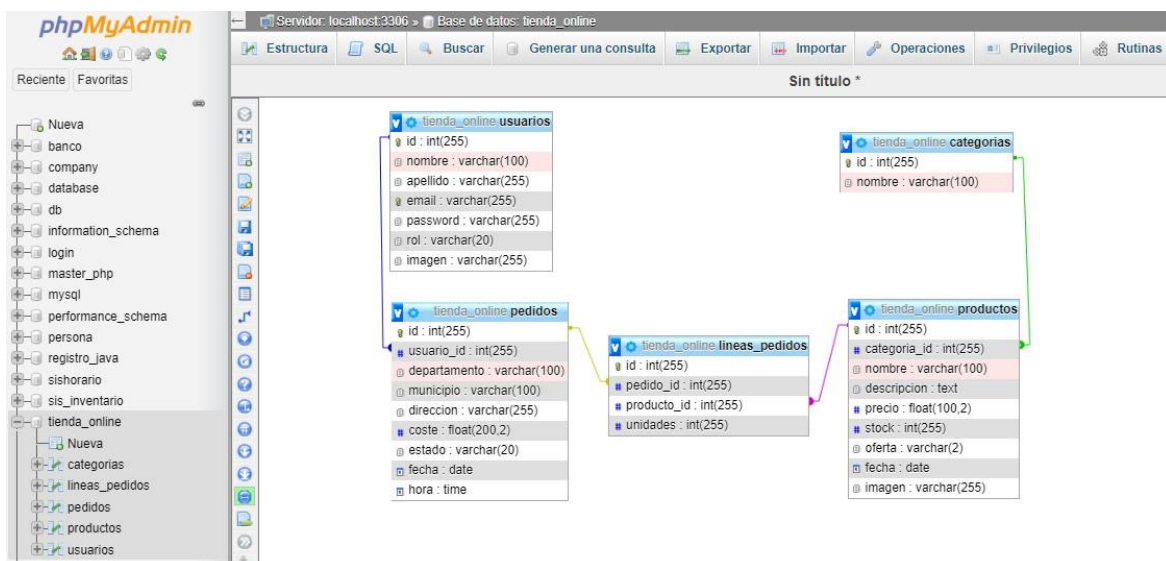
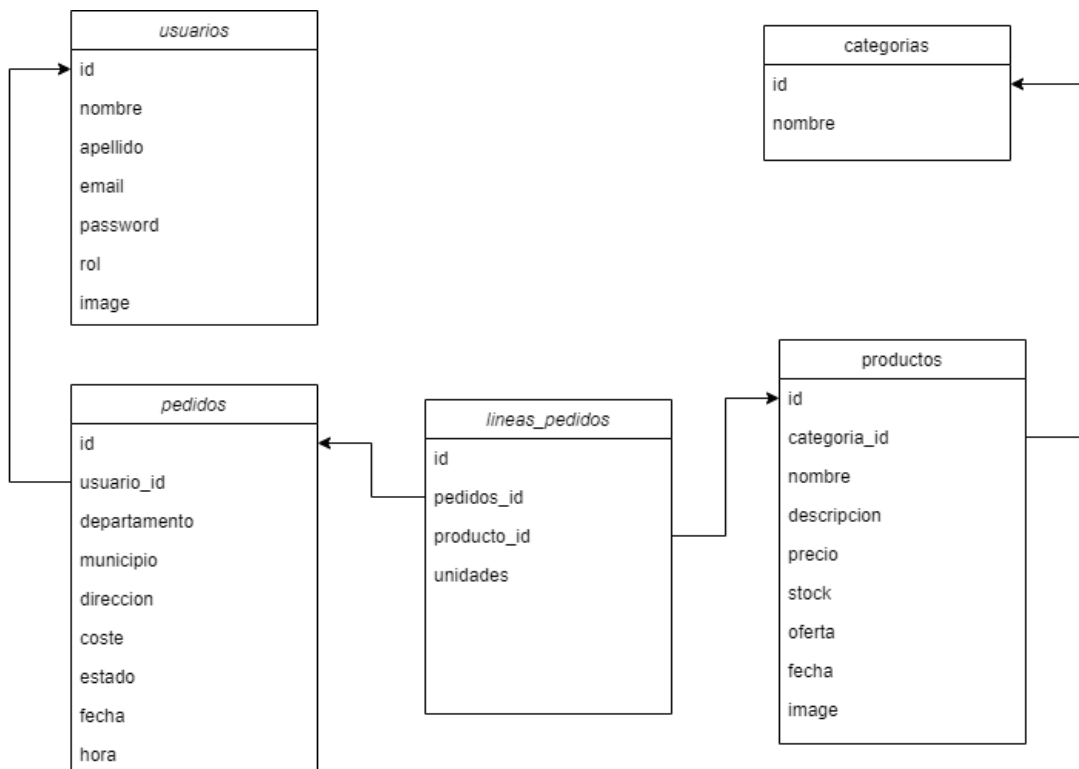









Tabla de Usuario

v  tienda_online usuarios	
	id : int(255)
	nombre : varchar(100)
	apellido : varchar(255)
	email : varchar(255)
	password : varchar(255)
	rol : varchar(20)
	imagen : varchar(255)

En esta tabla se van a guardar todos los registros de los usuarios y en donde habrá diferentes tipos de roles entre usuarios.

Tabla Categoría

v  tienda_online categorias	
	id : int(255)
	nombre : varchar(100)

En esta tabla se guardarán todos los registros de las categorías, lo que nos permitirá separar los productos, para saber a qué categoría pertenece cada uno de ellos.



Tabla Productos

v tienda_online productos	
🔑	id : int(255)
#	categoria_id : int(255)
📄	nombre : varchar(100)
📄	descripcion : text
#	precio : float(100,2)
#	stock : int(255)
📄	oferta : varchar(2)
📅	fecha : date
📄	imagen : varchar(255)

Se guardarán todos los productos que los usuarios registren y también encontraremos la primera relación de la base de datos que es “categoria_id”, en la cual le pondremos a cada producto una categoría.



Tabla Pedidos

v	tienda_online pedidos
🔑	id : int(255)
#	usuario_id : int(255)
📄	departamento : varchar(100)
📄	municipio : varchar(100)
📄	direccion : varchar(255)
#	coste : float(200,2)
📄	estado : varchar(20)
📅	fecha : date
🕒	hora : time

En esta tabla guardaremos todos los pedidos que realizaran los usuarios, aquí encontramos la segunda relación, en donde esta nos permitirá identificar que usuario hizo el pedido.

Tabla líneas_pedidos

v	tienda_online lineas_pedidos
🔑	id : int(255)
#	pedido_id : int(255)
#	producto_id : int(255)
#	unidades : int(255)



En esta tabla guardaremos todos los pedidos que realizaran los usuarios, aquí encontramos la segunda relación, en donde esta nos permitirá identificar que usuario hizo el pedido.

Consultas

Consulta de pedidos:

```
SELECT
    lineas_pedidos.unidades,
    pedidos.coste,
    pedidos.fecha,
    pedidos.estado
FROM
    pedidos
INNER JOIN
    lineas_pedidos
ON
    pedidos.id = lineas_pedidos.pedido_id
```

Carrito de compra:

```
SELECT
    usuarios.nombre,
    usuarios.email,
    productos.nombre,
    productos.precio,
    lineas_pedidos.unidades
FROM
    usuarios,
    productos
INNER JOIN
    lineas_pedidos
ON
    productos.id = lineas_pedidos.producto_id
```




Direccion del pedido:

```
SELECT
    pedidos.departamento,
    pedidos.municipio,
    pedidos.direccion
FROM
    Pedidos
```

Consultas Usuarios

Guardar usuarios:

```
INSERT INTO usuarios VALUES (NULL,'brando','lopez', 'brando@brando.com',
    '231313113','user', null);
```

Buscar el email del usuario para el login:

```
SELECT * FROM usuarios WHERE email = 'brando@gmail.com'
```

Consultas Productos

Retornar todos los productos:

```
SELECT * FROM productos ORDER BY id DESC
```

Producto específico de una categoria:

```
SELECT p.*, c.nombre AS 'catnombre' FROM productos p
    INNER JOIN categorias c ON c.id = p.categoria_id
    WHERE p.categoria_id
    ORDER BY id DESC
```

Aparecer 6 productos aleatorios:

```
SELECT * FROM productos ORDER BY RAND() LIMIT 6
```

Buscar un producto específico:

```
SELECT * FROM productos WHERE id = 26
```





Guardar un producto:

```
INSERT INTO productos VALUES (NULL, 2, 'camisa', 'azul', 250000, 23, null, CURDATE (), null);
```

Actualizar Producto:

```
UPDATE productos SET nombre="camiseta" WHERE nombre='camisa';
```

Eliminar producto:

```
DELETE FROM productos WHERE id=26
```

Consultas Categorías

Mostrar las categorías:

```
SELECT * FROM categorias ORDER BY id DESC
```

Buscar una categoría en específico:

```
SELECT * FROM categorias WHERE id=6
```

Guardar una categoría:

```
INSERT INTO categorias VALUES (NULL,"zapatos")
```

Editar una categoría:

```
UPDATE categorias SET nombre="zapateria" WHERE nombre='zapatos';
```

Eliminar una categoría:

```
DELETE FROM categorias WHERE id=6
```



Consultas Pedidos

Buscar pedidos en específico:

```
SELECT * FROM pedidos WHERE id=4
```

Buscar un pedido por usuario:

```
SELECT p.id, p.coste FROM pedidos p WHERE p.usuario_id ORDER BY id DESC LIMIT 1
```

Listar todos los productos que existan en la tabla de lineas_pedidos:

```
SELECT pr.*, lp.unidades FROM productos pr INNER JOIN lineas_pedidos lp ON pr.id =  
lp.producto_id WHERE lp.pedido_id
```

Guardar un pedido:

```
INSERT INTO pedidos VALUES(NULL, 2, 'putumayo', 'mocoa', 'villa sofia', 15000, 'confirmado',  
CURDATE(), CURTIME());
```

Guardar en la tabla de lineas_pedidos:

```
INSERT INTO lineas_pedidos VALUES (null, 2, 4, 12)
```

Editar el estado de los productos:

```
UPDATE pedidos SET estado='confirmado' WHERE id=4;
```

Buscar el usuario que hizo un pedido:

```
SELECT * FROM pedidos P INNER JOIN usuarios U ON p.usuario_id = U.id
```

Triggers

Stock de los productos:

```
UPDATE productos SET stock = stock - NEW.unidades WHERE  
productos.id = NEW.producto_id
```

<title>localhost / localhost / tienda_online | phpMyAdmin 5.0.1</title>

Detalles

Nombre del disparador	before_lineas_pedidos
Tabla	lineas_pedidos
Tiempo	BEFORE
Evento	INSERT
Definición	<pre>1 UPDATE productos SET stock = stock - NEW.unidades WHERE productos.id = NEW.producto_id</pre>
Definidor	root@localhost

Continuar Cerrar