1. **Tabla de Usuarios:**
   * Esta tabla almacena la información de los usuarios, que puede ser útil si planeas permitir que los usuarios guarden sus pedidos o realicen pedidos mientras están autenticados.

SQL

CREATE TABLE `users` (

`idUsuario` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(25) NOT NULL,

`password` VARCHAR(255) NOT NULL,

`email` VARCHAR(100) NOT NULL,

PRIMARY KEY (`idUsuario`),

UNIQUE KEY `username` (`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

1. **Tabla ubicaciones:**

SQL

CREATE TABLE `ubicaciones` ( `idUbicacion` INT UNSIGNED NOT NULL AUTO\_INCREMENT, `idUsuario` INT UNSIGNED NOT NULL, `nombre` VARCHAR(80), `C.P.` VARCHAR(5), `calle` VARCHAR(50), `colonia` VARCHAR(80), `numTelefono` VARCHAR(10), `instruccionesEntrega` VARCHAR(250), PRIMARY KEY (`idUbicacion`), FOREIGN KEY (`idUsuario`) REFERENCES `users`(`idUsuario`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

Esta tabla tiene los siguientes campos:

* **idUbicacion**: Un identificador único para cada ubicación, configurado como **INT UNSIGNED** y con **AUTO\_INCREMENT**.
* **idUsuario**: El ID del usuario al que pertenece esta ubicación, configurado como **INT UNSIGNED** y establecido como una clave externa que hace referencia al campo **idUsuario** en la tabla **users**.
* **nombre**: El nombre de la ubicación, configurado como **VARCHAR(80)**.
* **C.P.**: Código Postal, configurado como **VARCHAR(5)**.
* **calle**: El nombre de la calle, configurado como **VARCHAR(50)**.
* **colonia**: El nombre de la colonia, configurado como **VARCHAR(80)**.
* **numTelefono**: El número de teléfono, configurado como **VARCHAR(10)**.
* **instruccionesEntrega**: Instrucciones de entrega, configurado como **VARCHAR(250)**.

Ambas tablas utilizan el motor de almacenamiento InnoDB para garantizar la integridad referencial y están configuradas con el conjunto de caracteres UTF-8 y el conjunto de collation utf8\_unicode\_ci para admitir caracteres multilingües de manera adecuada.

Con esta estructura de base de datos, puedes almacenar los productos en la tabla **Productos**, gestionar los pedidos en la tabla **Pedidos** y registrar los detalles de cada pedido en la tabla **DetallesPedido**. Asegúrate de ajustar esta estructura según los requisitos específicos de tu aplicación. Además, puedes agregar más campos según sea necesario, como información de dirección para los pedidos o información adicional del usuario.

1. **Tabla de Productos:**
   1. Esta tabla almacena la información de cada producto, como su nombre, descripción, precio e imagen.

SQL

CREATE TABLE Productos (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255) NOT NULL,

descripcion TEXT,

precio DECIMAL(10, 2) NOT NULL,

imagen VARCHAR(255)

);

1. **Tabla de Pedidos:**
   1. Esta tabla almacena la información de los pedidos realizados por los usuarios.

SQL

CREATE TABLE Pedidos (

id INT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id INT,

fecha\_pedido TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

estado\_pedido ENUM('pendiente', 'en\_proceso', 'completado') DEFAULT 'pendiente',

FOREIGN KEY (usuario\_id) REFERENCES Usuarios(id)

);

1. **Tabla de Detalles del Pedido:**
   1. Esta tabla almacena los detalles de cada pedido, como los productos que contiene y la cantidad.

SQL

CREATE TABLE DetallesPedido (

id INT AUTO\_INCREMENT PRIMARY KEY,

pedido\_id INT,

producto\_id INT,

cantidad INT,

FOREIGN KEY (pedido\_id) REFERENCES Pedidos(id),

FOREIGN KEY (producto\_id) REFERENCES Productos(id)

);

1. **Verificar si el usuario ha iniciado sesión (Código en PHP).**

<?php

session\_start();

// Verificar si el usuario ha iniciado sesión

if (isset($\_SESSION['loggedin']) && $\_SESSION['loggedin'] === true) {

$username = $\_SESSION['username'];

} else {

// Si el usuario no ha iniciado sesión, redirigirlo a la página de inicio de sesión

header('Location: login.php');

exit;

}

?>