

Diseño de la Base de Datos

1. Objetivo

El objetivo de este documento es definir los requerimientos para el diseño de la base de datos que soportará una aplicación de Ecommerce con control de inventario, autenticación de usuarios, filtros por categoría, carrito de compras y órdenes de compra. El diseño se enfoca en la normalización de los datos, el uso de índices y restricciones, y la implementación de procedimientos almacenados para optimizar las operaciones complejas.

2. Requerimientos Funcionales de la Base de Datos

2.1. Entidades y Relacionamientos

La base de datos deberá incluir las siguientes entidades clave:

1. Cuenta:

- Almacena la información de los clientes que interactúan con la plataforma.
- Debe incluir campos para el nombre, correo electrónico, contraseña cifrada, dirección de envío y roles (admin, cliente).

2. Productos:

- Información de los productos disponibles en el Ecommerce.
- Cada producto debe tener una categoría, un precio, una descripción, una cantidad en inventario y una imagen.

3. Categorías:

- Define las distintas categorías de productos (por ejemplo, "Electrónica", "Ropa", "Hogar").

4. Carrito de Compras:

- Relación entre los usuarios y los productos que han agregado al carrito.
- Incluye información sobre cantidades y estado del carrito (activo, procesado).

5. Órdenes de Compra:

- Cada orden debe estar relacionada con un usuario y contener los productos seleccionados.
- Incluye orden del producto con información como precio y cantidad

2.2. Relaciones y Cardinalidad

- **Un usuario puede tener múltiples órdenes de compra**, pero cada orden está asociada a un único usuario (relación uno a muchos).

- **Un usuario puede agregar múltiples productos a su carrito de compras y un producto puede estar en múltiples carritos** (relación muchos a muchos).
- **Cada producto pertenece a una categoría y una categoría puede tener múltiples productos** (relación uno a muchos).
- **Una orden puede contener varios productos y un producto puede pertenecer a varias órdenes** (relación muchos a muchos, con una tabla intermedia para los detalles de la orden).

3. Normalización y Estructura de las Tablas

Estructura básica de las tablas:

- **Tabla Usuarios:**
 - id_usuario (PK)
 - nombre
 - apellido
 - username
 - email
 - contraseña
 - numero_telefono
 - fecha_creacion
 - ultimo_ingreso
 - es_admin
 - es_staff
 - es_activo
 - es_superadmin
- **Tabla Productos:**
 - id_producto (PK)
 - nombre
 - descripción
 - slug
 - precio
 - cantidad_en_inventario

- imagen
- disponible
- fecha_creacion
- fecha_modificacion
- id_categoria (FK)

- **Tabla Categorías:**

- id_categoria (PK)
- nombre_categoria
- slug
- descripcion
- imagen

- **Tabla Carrito de Compras:**

- id_carrito (PK)
- fecha_creacion

- **Tabla Productos_Carrito** (tabla intermedia para relación muchos a muchos):

- id_producto_carrito (PK)
- id_carrito (FK)
- id_usuario (FK)
- id_producto (FK)
- cantidad
- activo

- **Tabla Órdenes de Compra:**

- id_orden (PK)
- id_usuario (FK)
- id_pago (FK)
- total
- estado_orden
- fecha_creacion
- nota

- ordenado
- **Tabla Orden de producto** (relación muchos a muchos entre órdenes y productos):
 - id_orden_producto (PK)
 - id_orden (FK)
 - id_producto (FK)
 - id_pago (FK)
 - precio
 - cantidad
 - ordenado
 - fecha_creacion

4. Índices y Constraints

Para optimizar el rendimiento de las consultas y garantizar la integridad de los datos, se aplicarán índices y restricciones:

4.1. Índices

- Crear **índices** en los campos más consultados para mejorar la eficiencia de las búsquedas:
 - Índice en email de la tabla **Usuarios** para agilizar la autenticación.
 - Índice en nombre de la tabla **Productos** para optimizar las búsquedas por nombre de producto.
 - Índice en id_categoria en **Productos** para mejorar las consultas por categoría.

4.2. Restricciones (Constraints)

- **Clave primaria** en todas las tablas (id_usuario, id_producto, etc.).
- **Clave foránea** (FK) en las tablas que relacionan otras entidades:
 - id_usuario en **Carrito de Compras** y **Órdenes de Compra**.
 - id_producto en **Productos_Carrito** y **Detalles_Orden**.
 - id_categoria en **Productos**.
- **Restricciones de unicidad**:
 - Unicidad en el campo email de la tabla **Usuarios** para evitar duplicidad de registros.
- **Restricciones de no nulidad** en campos esenciales como nombre, precio, cantidad_en_inventario en la tabla **Productos**.

- **Integridad referencial:** Si un producto o usuario es eliminado, sus entradas asociadas en las tablas relacionadas también deben ser eliminadas (cascada).

5. Procedimientos Almacenados

5.1. Operaciones Complejas

Se implementarán procedimientos almacenados para operaciones más complejas que involucran múltiples tablas o cálculos específicos.

5.1.1. Actualización del Stock Tras Compra

Un procedimiento almacenado actualizará la cantidad de inventario tras una compra. Este proceso disminuirá la cantidad disponible en la tabla de **Productos** y validará si el stock es suficiente antes de completar la orden.

5.1.2. Calcular el Total de una Orden

Otro procedimiento almacenado calculará el total de una orden sumando el precio y la cantidad de cada producto en el carrito, antes de confirmar el pedido en la tabla de **Órdenes de Compra**.

5.1.3. Historial de Compras de un Usuario

Un procedimiento almacenado recuperará el historial de compras de un usuario, uniendo datos de las tablas **Órdenes de Compra** y **Orden de producto**.

6. Consideraciones de Seguridad

- Las contraseñas deben ser **cifradas** antes de ser almacenadas en la base de datos.
- Implementar **roles y permisos** de usuario para garantizar que solo los administradores puedan acceder al panel de gestión de inventario.

7. Plan de Escalabilidad

- A medida que la base de datos crezca, se pueden considerar técnicas como el particionamiento de tablas grandes (ej. particionar la tabla de **Órdenes de Compra** por fecha).
- Uso de **replicación de bases de datos** para manejar grandes cantidades de tráfico y asegurar alta disponibilidad.