

## 08-Mongoose

### Proyecto E-commerce: Consignas

A continuación, se detallan las consignas que deben completarse para avanzar en el desarrollo del proyecto de **e-commerce** utilizando **MongoDB** y **Mongoose**. Es importante que sigas cada paso de manera ordenada y verifiques el funcionamiento del proyecto al finalizar cada sección.

#### 1. Conectar una base de datos de MongoDB con Mongoose

Debes establecer una conexión entre tu aplicación de Node.js y MongoDB utilizando **Mongoose**. Asegúrate de configurar correctamente la URI de MongoDB, que puede ser una base de datos local o remota, dependiendo de tu entorno. La conexión debe manejar posibles errores de conexión para garantizar que la base de datos esté accesible antes de comenzar a operar.

- Configurar correctamente la conexión a la base de datos MongoDB.
- Asegurar que la base de datos se conecte antes de realizar cualquier operación.

#### 2. Definición de Esquemas de los Modelos de la Base de Datos

Crea los **esquemas** de los modelos que utilizarás en tu base de datos para manejar el e-commerce. Algunos de los modelos principales pueden incluir **Usuarios**, **Productos**, **Carrito de Compras**, y **Órdenes**. Define cada esquema especificando los campos necesarios para cada entidad.

- Definir esquemas de modelos que reflejen las entidades del e-commerce.
- Asegurarse de que cada modelo tenga los campos esenciales, como nombre de producto, precio, cantidad, etc.

#### 3. Uso de Tipos de Datos y Opciones de Esquema

Asegúrate de utilizar los **tipos de datos** correctos en cada esquema. Por ejemplo, los nombres suelen ser String, los precios son Number, y las fechas deben usar el tipo Date. También puedes aplicar **opciones adicionales**, como required, para garantizar que ciertos campos sean obligatorios, o default, para establecer valores predeterminados.

- Asignar correctamente los tipos de datos a cada campo del esquema.
- Utilizar opciones como required, default, unique, etc., según sea necesario.

#### 4. Validaciones

Configura **validaciones** dentro de los esquemas para asegurar que los datos que ingresen en la base de datos cumplan con ciertos requisitos. Por ejemplo, los correos electrónicos de los usuarios deben tener un formato válido, los precios no pueden ser negativos, y los nombres de productos deben tener una longitud mínima.

- Asegurar que los datos cumplan con las reglas necesarias antes de ser guardados en la base de datos.
- Implementar validaciones personalizadas para casos específicos si es necesario.

#### 5. Consultas y Operaciones CRUD

Debes implementar las **operaciones CRUD** (Crear, Leer, Actualizar, Eliminar) para cada uno de los modelos de la base de datos. Estas operaciones son fundamentales para la gestión de usuarios, productos, carritos y órdenes en la plataforma. Asegúrate de que las consultas sean eficientes y manejen correctamente los errores.

- Implementar rutas para crear, leer, actualizar y eliminar documentos en MongoDB utilizando Mongoose.

- Asegurar que las operaciones manejen los errores correctamente.

## 6. Relaciones entre Esquemas

Establece las **relaciones** necesarias entre los esquemas. Por ejemplo, un carrito debe estar asociado a un usuario, y una orden debe contener una lista de productos. Estas relaciones pueden implementarse utilizando **referencias** a otros documentos o **subdocumentos** según sea necesario. También, asegúrate de usar `populate()` cuando sea necesario para cargar documentos relacionados.

- Implementar correctamente las relaciones entre los modelos, como las relaciones entre usuarios y carritos, o entre órdenes y productos.
- Utilizar el método `populate()` para recuperar documentos referenciados completos cuando sea necesario.

---

Al completar cada una de estas consignas, realiza pruebas para verificar que la base de datos, las validaciones y las relaciones estén funcionando correctamente antes de avanzar al siguiente paso.