**Plataforma de Gestión de Productos**

**Descripción del Proyecto**

Este proyecto se trata de crear una plataforma en línea para gestionar productos y sus comentarios asociados. Los usuarios podrán tener uno o más productos asociados, y cada producto podrá tener uno o más comentarios. Se busca permitir operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en estas entidades, así como también registrar usuarios en el servidor. Se utilizará el sistema ORM Sequelize para interactuar con una base de datos MySQL.

**Justificación de la Temática**

La elección de esta temática es relevante debido a su aplicabilidad en numerosos contextos. Por ejemplo, en plataformas de comercio electrónico, redes sociales o sistemas de revisión de productos. La gestión de usuarios, productos y comentarios es esencial para crear una experiencia interactiva y enriquecedora para los usuarios.

**Objetivo:**

El objetivo del proyecto de "Plataforma de Gestión de Productos" es crear una aplicación en línea que permita a los usuarios administrar productos y participar en discusiones mediante comentarios. Los principales objetivos incluyen:

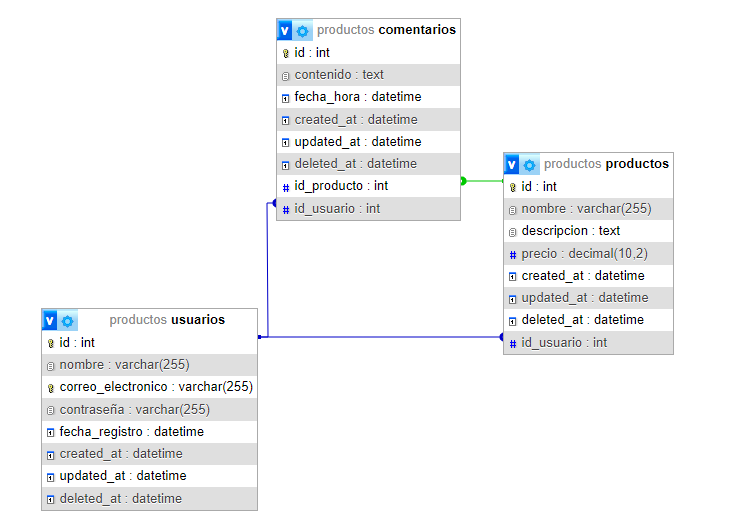
1. **Gestión de Productos:** Desarrollar una lógica que permitan a los usuarios realizar operaciones CRUD en los productos. Esto implica la capacidad de crear, leer, actualizar y eliminar productos de manera efectiva y eficiente.
2. **Interacción de Usuarios:** Facilitar la participación de los usuarios al permitirles dejar comentarios en los productos. Esto fomentará la interacción, permitirá opiniones y reseñas, y enriquecerá la experiencia del usuario.
3. **Validación de Datos:** Implementar mecanismos de validación para asegurar que los datos ingresados por los usuarios sean correctos y cumplan con ciertos criterios predefinidos. Esto garantizará la integridad de la información almacenada en la base de datos.
4. **Eficiencia en la Base de Datos:** Utilizar el sistema ORM Sequelize para optimizar el acceso y la manipulación de datos en la base de datos MySQL, asegurando consultas eficientes y un rendimiento óptimo.

**Red Conceptual**

Red conceptual para visualizar la lógica del proyecto:

| Usuario |+----------+ 1 n +----------+| Producto|+----------+ 1 n +----------+| Comentario|

|+-----------------------------------------+ 1 n +---------------------------------------------+|



Las relaciones serán las siguientes:

* La tabla "Productos" tendrá una clave foránea (FK) que hace referencia a la tabla "Usuarios" (uno a muchos).
* La tabla "Comentarios" tendrá dos claves foráneas (FK) que hacen referencia a las tablas "Usuarios" y "Productos" respectivamente (uno a muchos con ambas).

**1. Usuario**

La entidad "Usuario" representa a las personas que utilizan la plataforma. Cada usuario podrá agregar productos y dejar comentarios en los productos.

Atributos de la entidad "Usuario":

* **id:** Identificador único del usuario (clave primaria).
* **nombre:** Nombre del usuario.
* **correoElectrónico:** Correo electrónico del usuario.
* **contraseña:** Contraseña del usuario (se recomienda almacenar de manera segura y encriptada).
* **fechaRegistro:** Fecha y hora de registro del usuario.
* **createdAt:** Fecha y hora de creación del usuario (generado automáticamente).
* **updatedAt:** Fecha y hora de la última actualización del usuario (generado automáticamente).

**2. Producto**

La entidad "Producto" representa los elementos que los usuarios pueden agregar, editar, eliminar y revisar en la plataforma. Cada producto tendrá características específicas que describen su naturaleza y propósito.

Atributos de la entidad "Producto":

* **id:** Identificador único del producto (clave primaria).
* **nombre:** Nombre del producto.
* **descripción:** Descripción detallada del producto.
* **precio:** Precio del producto.
* **createdAt:** Fecha y hora de creación del producto (generado automáticamente).
* **updatedAt:** Fecha y hora de la última actualización del producto (generado automáticamente).

**3. Comentario**

La entidad "Comentario" permite a los usuarios expresar sus opiniones y comentarios sobre un producto específico. Los comentarios estarán vinculados a los productos y a los usuarios que los dejaron.

Atributos de la entidad "Comentario":

* **id:** Identificador único del comentario (clave primaria).
* **contenido:** Texto del comentario.
* **fechaHora:** Fecha y hora en que se dejó el comentario.
* **id\_producto:** Identificador del producto al que se refiere el comentario (clave foránea que referencia a la entidad Producto).
* **createdAt:** Fecha y hora de creación del comentario (generado automáticamente).
* **updatedAt:** Fecha y hora de la última actualización del comentario (generado automáticamente).

**Inicialización del Proyecto**

1. Crear una nueva carpeta para tu proyecto y navegar a ella en la terminal:

code .

1. Inicializar un nuevo proyecto Node.js:

npm init / npm install

**Instalación de Dependencias**

1. Instalar las dependencias necesarias: Sequelize, Express, MySQL, y express-validator:

npm install sequelize express mysql2 express-validator helmet cors dotenv nodemon morgan

**Configuración de Sequelize y Base de Datos**

1. Crear un archivo de configuración de la base de datos en **database/config.js**

**Definición de Modelos y Migraciones**

1. Crear los modelos y migraciones para "Usuario", "Producto" y "Comentario" de acuerdo a la red conceptual predefinida más arriba.
2. Ajustar las relaciones en los archivos de modelo generados en **models/**.

**Rutas y Controladores**

1. Crear las rutas y controladores en las respectivas carpetas (**routes/** y **controllers/**) para las operaciones CRUD en “Users”, “Products” y “Comments”. Implementa la validación de campos utilizando express-validator.

**Inicio del Servidor**

1. Crear el archivo principal del servidor en **app.js**

Importar las dependencias.

Configurar los middlewares.

Definir las rutas.

Iniciar el servidor.

**Ejecución**

1. Asegúrate de que todos los modelos, rutas y controladores estén correctamente definidos y configurados.
2. Ejecuta el servidor:

npm run dev (nodemon app.js)