

Práctica WEB Backend Node.js

Web19 2025

Imaginemos que un cliente nos pasa el siguiente briefing para que le hagamos este trabajo:

Desarrollar un website con SSR (ejs) para un servicio de venta de artículos de segunda mano llamado Nodepop.

El servicio mantiene productos a la venta y permite buscar como poner filtros por varios criterios.

- Cada producto publicado tiene los siguientes datos:
- Nombre del producto, un anuncio siempre tendrá un solo artículo
- Propietario. Usuario que ha publicado el producto
- Precio
- Tags del producto. Podrá contener uno o varios de estos cuatro: work, lifestyle, motor y mobile

Operaciones que debe permitir realizar:

- Lista de productos del usuario, con posibilidad de paginación. Con filtros por tag, rango de precio (precio min. y precio max.) y nombre de producto (que empiece por el dato buscado)
- Por ejemplo si hacemos una petición a `http://localhost:3000/?skip=1&limit=3&sort=name&tag=lifestyle` mostraría los productos correspondientes.
- Creación de producto
- Borrado de producto

Importante, cada usuario solo puede ver o eliminar sus propios productos.

Los sistemas donde se desplegará el website utilizan base de datos MongoDB.

Se solicita que el entregable venga acompañado de una mínima documentación y el código esté bien formateado para facilitar su mantenimiento. En esta fase, ya que se desea probar si el modelo de negocio va a funcionar, no serán necesarios ni tests unitarios ni de integración.

Notas para el desarrollador

Cómo empezar

El orden de las primeras tareas podría ser:

1. Crear app Express y probarla (`npx express-generator nodepop --ejs`)
2. Instalar Mongoose, modelo de productos y probarlo (con algún `product.save` por ejemplo)
3. Hacer un script de inicialización de la base de datos, que cargue productos y usuarios iniciales. Se puede llamar p.e. `initDB.js`, debería borrar las tablas y cargar posteriormente. Lo podemos poner en el `package.json` para poder usar `npm run initDB`. Referencias:
4. Cargar vuestro módulo `connectMongoose.js` y vuestros modelos
5. Usar `deleteMany` e `insertMany` para cargar las entidades.
6. Estas operaciones deberán hacerse una detrás de la otra, o dicho de otro modo, cuando termine la primera se lanzará la segunda. Para esto podéis usar `callbacks`, `promesas` (si miráis la doc veréis que ambos devuelven una promesa) o `promesas con async/await` (recomendado).
7. Hacer un fichero `README.md` con las instrucciones de uso puede ser una muy buena idea, lo ponemos en la raíz del proyecto y si apuntamos ahí como arrancarlo, como inicializar la BD, etc nos vendrá bien para cuando lo olvidemos o lo coja otra persona
8. Hacer una primera versión básica, por ejemplo `GET /` que devuelva la lista de productos.
9. Crear la página de inicio del site y sacar la lista de productos
10. Hacer `login` y `logout`
11. Mejorar la lista de productos del usuario poniendo filtros, paginación, etc
12. A partir de aquí ya podemos meternos con la creación y borrado!



KEEPCODING
Tech School