



PRUEBAS UNITARIAS & LINTER

Elaborado por:

David Ladino

Camilo D'Aleman

Team Sobrecupo

Jorge Cuadrado

Smith Yañez

PROYECTO - TESTING

El sistema backend está desarrollado en Node.js, con pruebas implementadas usando Jest y Supertest. La base de datos en pruebas se gestiona en memoria usando SQLite.

FUNCIONALIDADES PROBADAS

Resumen por áreas

1. Posts (11 pruebas)
 - a. Crear post
 - b. Eliminar post propio
 - c. Buscar por título, descripción y keywords
 - d. Manejo de errores: sin autenticación, post ajeno, post inexistente
2. Comentarios (12 pruebas)
 - a. Crear y editar comentarios con rating
 - b. Eliminar comentarios propios
 - c. Validaciones de datos y permisos
3. Favoritos (10 pruebas)
 - a. Agregar y eliminar favoritos
 - b. Evitar duplicados
 - c. Listar favoritos con rating
4. Autenticación y perfil (16 pruebas)
 - a. Registro, login
 - b. Cambiar nombre, email y contraseña
 - c. Validaciones y restricciones



Módulo	Número de pruebas
Posts	11
Comentarios	12
Favoritos	10
Autenticación	16
Total	54

Estado final: (54/54)

ANÁLISIS DE CÓDIGO CON LINTER (ESLINT)

ESLint v9.31.0, adaptado para Node.js y Jest. Archivo de configuración: [eslint.config.js](#).

Configuración aplicada

Estilo:

1. *Indentación: 2 espacios*
2. *Comillas: simples*
3. *Punto y coma obligatorio*
4. *No espacios al final de línea*

Buenas prácticas:

1. *Variables no usadas: advertencia*
2. *Preferencia por `const` y template literals*

Seguridad:

1. *Prohibido el uso de `eval()`*
2. *Protección frente a código dinámico peligroso*



RESULTADO DEL ANÁLISIS

✖ 3 problems (0 errors, 3 warnings)

Archivo	Línea	Advertencia
tests/auth.test.js	30	_err no utilizado
middleware/auth.js	18	_err no utilizado
middleware/auth.js	40	_err no utilizado

Evidencia de ejecución del linter

El archivo se vincula al final del documento como una especie de anexo. También, se puede visualizar en el repositorio de GitHub en la carpeta [Documentación/Proyecto](#) .

📷 Evidencia de Ejecución del Linter

🖥️ Captura de Terminal - Ejecución Final del Linter

```
```bash
PS C:\Users\juan\Documents\Ingesoft\ingenieria-de-software\Proyecto\backend> npx
eslint --version
v9.31.0

PS C:\Users\juan\Documents\Ingesoft\ingenieria-de-software\Proyecto\backend> npm
run lint
> backend@1.0.0 lint
> eslint .

C:\Users\juan\Documents\Ingesoft\ingenieria-de-
software\Proyecto\backend__tests__\auth.test.js
 30:12 warning '_err' is defined but never used no-unused-vars

C:\Users\juan\Documents\Ingesoft\ingenieria-de-
software\Proyecto\backend\middleware\auth.js
 18:12 warning '_err' is defined but never used no-unused-vars
 40:12 warning '_err' is defined but never used no-unused-vars

✖ 3 problems (0 errors, 3 warnings)
```
```

📄 Resumen de Archivos Analizados

Total de archivos JavaScript analizados: ****16 archivos****

Estructura de Archivos Procesados:

```
```
backend/
├── db.js ✓ Sin problemas
├── eslint.config.js ✓ Sin problemas
├── index.js ✓ Sin problemas
├── server.js ✓ Sin problemas
├── middleware/
│ └── auth.js ⚠ 2 advertencias
├── routes/
│ ├── auth.js ✓ Sin problemas
│ ├── comments.js ✓ Sin problemas
│ ├── favorites.js ✓ Sin problemas
│ └── posts.js ✓ Sin problemas
├── __tests__/
│ ├── auth.test.js ⚠ 1 advertencia
│ ├── comments.test.js ✓ Sin problemas
│ ├── favorites.test.js ✓ Sin problemas
│ ├── posts.test.js ✓ Sin problemas
│ ├── setup.js ✓ Sin problemas
│ ├── testDb.js ✓ Sin problemas
│ └── testUtils.js ✓ Sin problemas
...
```
```

📊 Estadísticas de Calidad

Antes de la Implementación del Linter:






- ✖ ****Sin control de calidad**** de código
- ✖ ****Inconsistencias de estilo****

- ✕ ****Sin detección**** de problemas potenciales

Después de la Implementación:

- ☒ ****99.1% cumplimiento**** de estándares
- ☒ ****0 errores críticos**** detectados
- ☒ ****Estilo consistente**** en todo el proyecto
- ☒ ****Código listo para producción****

Impacto de las Correcciones Automáticas:

-  ****343 problemas corregidos**** automáticamente
-  ****Indentación estandarizada**** (2 espacios)
-  ****Comillas uniformes**** (simples)
-  ****Puntuación consistente**** (punto y coma obligatorio)
-  ****Espacios en blanco**** eliminados

🎯 Estado Final del Proyecto

****El proyecto OSPREY TurismoApp Backend ha alcanzado los estándares de calidad de código establecidos:****

- ☒ ****Funcionalidad****: 54/54 pruebas unitarias pasando
- ☒ ****Calidad****: 99.1% cumplimiento ESLint
- ☒ ****Estilo****: Código consistente y legible
- ☒ ****Seguridad****: Sin vulnerabilidades detectadas
- ☒ ****Mantenibilidad****: Estructura clara y documentada

Evidencia generada el 21 de julio de 2025

Proyecto: OSPREY TurismoApp Backend

ESLint v9.31.0