



## Actividad 4

- Fecha de entrega 11 de feb a las 23:59
- Puntos 100
- Entregando una carga de archivo
- Disponible 2 de feb en 7:00 - 11 de feb en 23:59

Esta tarea fue bloqueada 11 de feb en 23:59.

### Descripción:

En esta actividad, crearás una aplicación web para la gestión de productos, en la cual los usuarios autenticados podrán agregar, modificar, ver y eliminar productos. La aplicación incluirá un sistema CRUD para los productos y gestionará el registro, inicio y cierre de sesión de usuarios, usando JWT para proteger rutas específicas. Para el desarrollo, se emplearán Node.js, Express.js, MongoDB, y Jest para pruebas unitarias. Además, el proyecto se desplegará mediante herramientas como GitHub Actions y una plataforma SaaS (por ejemplo, Vercel) con el objetivo de brindar una experiencia práctica en el desarrollo y despliegue seguro de aplicaciones web.

### Objetivo:

Desarrollar una aplicación *web* de Node JS utilizando las herramientas mencionadas en la descripción para gestionar productos en una base de datos a través de las operaciones CRUD e implementar un sistema de autenticación y autorización.

### Instrucciones:

#### a. Documentación del proyecto.

1. Crea un documento donde se describan los requerimientos funcionales y no funcionales de la aplicación.

#### b. Configuración del servidor.

1. Desarrolla una aplicación en Node.js, usando el *framework* de Express.js e implementando los *middlewares* necesarios para la autenticación mediante JWT y para la conexión a MongoDB como base de datos.

#### c. Generación de rutas.

1. Implementa middlewares para gestionar las rutas de los recursos de la aplicación, asegurando el acceso adecuado a las operaciones de cada recurso.

#### d. Implementación de autenticación y autorización.

1. Implementa el flujo de autenticación mediante JWT y protege las rutas de productos, incluyendo la creación de una vista estática con HTML para el *login* de la aplicación.

#### e. Generación de controladores.

1. Desarrolla controladores que gestionarán la interacción con la base de datos de MongoDB para cada recurso, utilizando la librería *Mongoose* para las consultas.

#### f. Generación de pruebas unitarias.

1. Configura la librería *Jest* para crear pruebas unitarias automatizadas que verifiquen el funcionamiento de los controladores y rutas.
2. Utiliza una herramienta de IA, como Copilot, para generar automáticamente pruebas para los controladores y rutas basadas en el código existente.
3. Ajusta manualmente las pruebas generadas por la IA para asegurar que cubran los casos específicos y cumplan con los requeridos del proyecto.

Nota. Revisa el siguiente enlace que puede ayudar a conocer más el funcionamiento de Copilot.

fuente: Jordán, L. (2024). *Cómo usar Copilot: configuración y aprendizaje de todos los trucos útiles de la interacción con IA*.

recuperado de [https://www.hostinger.es/tutoriales/como-usar-copilot?utm\\_source=chatgpt.com](https://www.hostinger.es/tutoriales/como-usar-copilot?utm_source=chatgpt.com)  
[https://www.hostinger.es/tutoriales/como-usar-copilot?utm\\_source=chatgpt.com](https://www.hostinger.es/tutoriales/como-usar-copilot?utm_source=chatgpt.com)

El siguiente enlace es externo a la Universidad Tecmilenio, al acceder a éste considera que debes estar de acuerdo a sus términos y condiciones.

#### g. Despliegue.

1. Despliega la aplicación en una plataforma SaaS como Vercel (u otra similar) que permita alojar el servidor y operaciones de Node.js.
2. Configura un *pipeline* de CI/CD mediante GitHub Actions que permita automatizar el despliegue de la aplicación.

#### h. Documentación del proyecto que incluya.

1. Adjunta en un documento Word, los requerimientos funcionales y no funcionales. Además, incluye el diagrama de entidad-relación y una justificación sobre la elección de la plataforma SaaS para el despliegue de la aplicación.
2. Adjunta el enlace del repositorio que contenga todos los archivos del proyecto, incluyendo el código fuente, el archivo de configuración del pipeline de CI/CD y las instrucciones para ejecutar la aplicación.

## Criterios de evaluación:

Considera que tu actividad se evaluará con esta [rúbrica](https://utm-cdn-labcontenidos-hftaarehf2gcfycs.a01.azurefd.net/contenido/profesional/Isti2313/anexos/actividades/v1/Rub_act4.pdf)  ([https://utm-cdn-labcontenidos-hftaarehf2gcfycs.a01.azurefd.net/contenido/profesional/Isti2313/anexos/actividades/v1/Rub\\_act4.pdf](https://utm-cdn-labcontenidos-hftaarehf2gcfycs.a01.azurefd.net/contenido/profesional/Isti2313/anexos/actividades/v1/Rub_act4.pdf)).

## Entregables:

Carpeta comprimida en formato .zip o .rar, junto con la URL de GitHub que contenga los elementos mencionados del punto h. La carpeta se entregará al impartidor por el medio que te indique (plataforma, *email*, etcétera).

**Una rúbrica (3)**

Criterios	Calificaciones			Pts
1. Configuración del servidor.	<b>20 para &gt;17.0 pts</b> <b>Altamente competente</b> El servidor está configurado de manera óptima en Node.js con Express.js; la conexión con MongoDB funciona correctamente y se han implementado todos los middleware necesarios para la autenticación JWT y el manejo de solicitudes.	<b>17 para &gt;14.0 pts</b> <b>Competente</b> El servidor está configurado adecuadamente, aunque presenta fallas menores en la conexión con MongoDB o en la implementación de algunos middleware de autenticación o manejo de solicitudes.	<b>14 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b> El servidor no está configurado correctamente, la conexión con MongoDB no funciona, o los middleware necesarios no se encuentran implementados.	20 pts
2. Generación de rutas.	<b>10 para &gt;8.0 pts</b> <b>Altamente competente</b> Las rutas CRUD funcionan correctamente para todas las operaciones y gestionan adecuadamente los errores, incluyendo respuestas claras y manejo de excepciones.	<b>8 para &gt;6.0 pts</b> <b>Competente</b> Las rutas CRUD funcionan para la mayoría de las operaciones, pero presentan errores en el manejo de excepciones o en la funcionalidad de ciertas rutas específicas.	<b>6 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b> Las rutas CRUD no funcionan, o no se implementó un manejo adecuado de errores y excepciones.	10 pts
3. Procesos de autenticación y autorización.	<b>20 para &gt;17.0 pts</b> <b>Altamente competente</b> Implementa correctamente el flujo de autenticación mediante JWT, las rutas de productos están protegidas y se incluye una vista estática en HTML para el login, que es funcional y presenta un diseño estéticamente adecuado.	<b>17 para &gt;14.0 pts</b> <b>Competente</b> El flujo de autenticación mediante JWT está parcialmente implementado, las rutas están protegidas de manera básica, y la vista estática para el login es funcional, aunque carece de algunos elementos estéticos o de usabilidad.	<b>14 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b> No se implementó el flujo de autenticación mediante JWT, las rutas de productos no están protegidas, o no se incluyó la vista estática para el login.	20 pts
4. Generación de controladores.	<b>15 para &gt;12.0 pts</b> <b>Altamente competente</b> Los controladores gestionan correctamente	<b>12 para &gt;9.0 pts</b> <b>Competente</b> Los controladores funcionan en su mayoría,	<b>9 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b>	15 pts

Criterios	Calificaciones			Pts
	las operaciones CRUD con MongoDB; el código es limpio y organizado, con un uso adecuado de la librería Mongoose para consultas y operaciones.	pero presentan errores o inconsistencias en el uso de Mongoose o en la organización del código.	Los controladores no gestionan adecuadamente las operaciones CRUD, o no se utiliza la librería Mongoose de forma correcta.	
5. Generación de pruebas unitarias.	<b>10 para &gt;8.0 pts</b> <b>Altamente competente</b> Configura Jest para generar pruebas unitarias automatizadas que cubren exhaustivamente los controladores y rutas. Además, utiliza una IA como Copilot para generar pruebas iniciales y se ajustan manualmente para cubrir todos los casos requeridos del proyecto.	<b>8 para &gt;6.0 pts</b> <b>Competente</b> Configura Jest y utiliza una IA como Copilot para generar pruebas unitarias, pero los ajustes manuales realizados solo cubren parcialmente los casos específicos del proyecto.	<b>6 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b> No se configuran las pruebas unitarias con Jest, tampoco utiliza una de IA, además no ajusta las pruebas generadas automáticamente para cubrir los casos requeridos del proyecto.	10 pts
6. Despliegue.	<b>15 para &gt;12.0 pts</b> <b>Altamente competente</b> La aplicación está correctamente desplegada en una plataforma SaaS que aloja el servidor de Node.js; el pipeline CI/CD está configurado y funciona automáticamente, asegurando un despliegue eficiente.	<b>12 para &gt;9.0 pts</b> <b>Competente</b> La aplicación está desplegada en una plataforma SaaS, pero el pipeline CI/CD presenta errores o no está completamente automatizado.	<b>9 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b> La aplicación no está desplegada en una plataforma SaaS, o no se configuró el pipeline CI/CD.	15 pts
7. Documentación del proyecto.	<b>10 para &gt;8.0 pts</b> <b>Altamente competente</b> La documentación incluye los requerimientos funcionales y no funcionales detallados, un diagrama de entidad-	<b>8 para &gt;6.0 pts</b> <b>Competente</b> La documentación incluye la mayoría de los requerimientos funcionales y no funcionales, un diagrama de entidad-relación	<b>6 para &gt;0 pts</b> <b>Aún sin desarrollar la competencia</b> La documentación está incompleta, carece de detalles relevantes en los requerimientos, no incluye diagrama de	10 pts

Criterios	Calificaciones			Pts
	relación claro y una justificación bien argumentada sobre la elección de la plataforma SaaS.	comprensible y una justificación aceptable sobre la elección de la plataforma SaaS.	entidad-relación y la justificación de la plataforma SaaS es incoherente o inexistente.	
Puntos totales: 100				