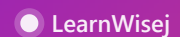


Ejercicio de programación con IA

Construye el ejemplo EnterpriseOps de este módulo con ChatGPT o Claude a partir de un prompt listo para usar que dirige al modelo a la especificación del módulo, y después revisa, ejecuta y amplía lo que te devuelva — incluso comparándolo con nuestra propia solución de referencia en GitHub.



El mismo módulo, construido como de verdad vas a construir.

Todas las demás secciones de este módulo explican cómo funciona «Security Architecture: Identity, SSO, Authorization, Audit, and Secure Deployment». Esta te pide que lo generes — dirigiendo a un asistente con la propia especificación del módulo y comprobando después que el resultado se ejecuta y hace lo que describe el enunciado del laboratorio. Es el flujo de trabajo que usarás en proyectos reales, practicado con algo cuya respuesta correcta ya conoces.

Qué estás pidiendo

Añade una simulación de autenticación empresarial, el mapeo de claims a permisos, la aplicación de roles por tenant, la autorización a nivel de servicio, la revisión de HTML seguro, la aprobación de exportaciones y el registro de auditoría de las acciones sensibles.

El módulo pide estos entregables, y el prompt de abajo le pide al asistente exactamente esos:

- ✓ Diseño del mapeo de identidades
- ✓ Matriz de permisos
- ✓ Implementación del servicio de permisos
- ✓ Pantalla del registro de auditoría
- ✓ Lista de comprobación de endurecimiento de la seguridad

Por qué tiene esta forma

Una arquitectura del lado del servidor reduce muchas superficies de ataque del navegador, pero los sistemas avanzados siguen necesitando integración de identidades, autorización a nivel de servicio, pistas de auditoría, renderizado seguro y una configuración operativa segura.

Estos son los términos que el asistente tiene que dominar. Si lo que te devuelve los usa a la ligera, no ha entendido el módulo:

Término	Qué significa aquí
límite OIDC/SSO	El punto en el que un proveedor de identidades externo entrega a la aplicación una identidad verificada. Todo lo que hay después de ese límite es responsabilidad de la aplicación.
mapeo de claims	Traducir los claims específicos del proveedor a los roles y permisos propios de la aplicación, para que el código de autorización nunca dependa del formato de un proveedor de identidades concreto.
servicio de permisos	Un único servicio que responde si el usuario actual puede realizar una acción, para que la regla viva en un solo lugar y no en cada pantalla.
autorización a nivel de servicio	Aplicar los permisos dentro de los servicios de aplicación, no solo en la UI, para que un botón oculto nunca sea lo único que protege una operación.
registro de auditoría	Un registro a prueba de manipulaciones de quién hizo qué y cuándo en las acciones importantes, para poder investigar los incidentes de seguridad y demostrar el cumplimiento normativo.
revisión de AllowHtml	Comprobar cada lugar en el que un control permite HTML sin procesar para asegurarse de que el contenido aportado por el usuario se sanea, porque el HTML sin revisar es un riesgo de cross-site scripting.
gestión de secretos	Guardar las cadenas de conexión, las claves y los tokens en una configuración segura o en un almacén de secretos, nunca en el control de código fuente ni en ajustes visibles desde el cliente.
cabeceras seguras	Cabeceras de respuesta HTTP como la política de seguridad de contenido y la seguridad de transporte estricta, que reducen la superficie de ataque de la aplicación en el navegador.

Constrúyelo — elige tu asistente

Los dos botones abren una pestaña nueva con el **mismo prompt listo para usar** ya escrito. El prompt dirige al modelo a la especificación de este módulo — un único archivo Markdown con la lección, el enunciado del laboratorio y los entregables — para que trabaje con el material real en lugar de adivinar de qué trata el Módulo 10. Puedes abrir tú mismo [module-10.md](#) para ver exactamente lo que leerá.



Usar ChatGPT

Abre [chatgpt.com](#) con el prompt de este módulo ya cargado. Funciona con GPT-5 o Codex; cualquier plan que pueda navegar por la web descargará la especificación.



Usar Claude

Abre [claude.ai](#) con el mismo prompt ya cargado. Si usas Claude Code, puedes pegarlo en una terminal abierta en una carpeta vacía.



Envíalo tal cual y obtendrás un proyecto, no un fragmento

El prompt termina pidiendo toda la solución como un único `.zip` con los pasos para compilarla y ejecutarla. Descárgalo, abre la solución en Visual Studio, pulsa F5 y EnterpriseOps se abrirá en `localhost`. Necesitas el SDK de .NET y el paquete NuGet `Wisej-4`; todo lo demás, el prompt le pide al asistente que lo escriba él mismo.

Estoy siguiendo el Módulo 10 – «Security Architecture: Identity, SSO, Authorization, Audit, and Secure Deployment» – del curso de LearnWisej «Enterprise Wisej.NET: Architecture to Cloud», y quiero construir contigo el ejemplo de este módulo. Respóndeme en español.

Empieza leyendo la especificación del módulo. Es un único archivo Markdown (en inglés) que contiene la lección completa, el enunciado del laboratorio y los entregables:

<https://learnwisej.com/assets/courses/azure-docker/specs/module-10.md>

Después construye un proyecto Wisej.NET 4 completo que pueda abrir en Visual Studio:

1. Proyecto «EnterpriseOps», C#, net8.0 o posterior, paquete NuGet Wisej-4 4.1.x, host Kestrel que llame a `app.UseWisej()`.
2. Construye lo que pide el laboratorio de la especificación: Añade una simulación de autenticación empresarial, el mapeo de claims a permisos, la aplicación de roles por tenant, la autorización a nivel de servicio, la revisión de HTML seguro, la aprobación de exportaciones y el registro de auditoría de las acciones sensibles.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Diseño del mapeo de identidades
 - Matriz de permisos
 - Implementación del servicio de permisos
 - Pantalla del registro de auditoría
 - Lista de comprobación de endurecimiento de la seguridad
4. Sigue la especificación siempre que nombre una clase, propiedad o evento de Wisej.NET – usa exactamente ese. Donde deje la elección abierta, dime qué API elegiste y por qué.
5. Mantén la autoridad en el servidor: el estado y la validación viven en .NET, y el navegador muestra e informa. Nunca traslades una decisión a JavaScript porque allí resulte más fácil.
6. Haz que el resultado se pueda demostrar: debo poder ejecutarlo y ver el comportamiento del que trata este módulo, incluido lo que ocurre cuando algo sale mal.

Antes de escribir una sola línea de código, explica con tus propias palabras qué vas a construir y dónde vivirá cada pieza, para que pueda corregir el diseño mientras todavía es barato.

Cuando el código esté terminado, empaqueta toda la solución en un único archivo .zip descargable y dame los pasos exactos para compilarla y ejecutarla.

Por último, responde a estas preguntas de la especificación usando tu propia implementación como prueba:

- ¿Dónde se aplica la autorización?
- ¿Qué elementos de la UI renderizan HTML?
- ¿Puede un botón habilitado eludir las reglas del servicio?

Pregúntame por cualquier cosa ambigua antes de empezar.

Después revisa lo que te devuelve

Que el código compile no significa que sea correcto. Las preguntas interesantes son de diseño: ¿es este el control de Wisej.NET que la tarea realmente necesita?, ¿sigue el ciclo de vida que te ofrece el framework?,

¿el estado de referencia está en el servidor o se quedó en el navegador porque era más cómodo? Lee el resultado como revisor, no como cliente.

Este módulo te da la lista de comprobación. Responde a cada punto basándote en el código que acabas de recibir:

- ✓ ¿Dónde se aplica la autorización?
- ✓ ¿Qué elementos de la UI renderizan HTML?
- ✓ ¿Puede un botón habilitado eludir las reglas del servicio?

Todo lo que hayas tenido que corregir merece quedar por escrito. Una nota breve con las convenciones y decisiones que acordaste, pegada en el primer mensaje de tu próxima sesión, vale más que cualquier prompt ingenioso.



Este ya lo construimos — compara tu versión con la nuestra

La implementación de referencia del Módulo 10 está en el repositorio público de ejemplos. Construye primero la tuya y después abre la nuestra: lo interesante son las diferencias.



[Abrir el ejemplo del Módulo 10 en GitHub](#)

O clona el conjunto completo con `git clone https://github.com/iceteagroup/LearnWisej-Samples` y abre `Enterprise Wisej.NET Course/Module 10`.



Nuestro ejemplo es un punto de partida, no un producto terminado

Se construyó igual que estás construyendo el tuyo — con un prompt y luego corregido — y se detiene a propósito donde se detiene el módulo. Espera soluciones provisionales donde iría código de producción, un manejo de errores mínimo y ninguna prueba. Tómallo como una invitación: usa lo que aprendiste en este módulo, junto con el asistente, para llevarlo más lejos. Mejorar código que no escribiste tú es precisamente la habilidad que enseña este curso.

Tres prompts para ir más allá

Dirige tu asistente al proyecto que acabas de construir — o al nuestro y prueba estos. Cada uno es un ejercicio de revisión y ampliación, no una reescritura.

- ✓ **Demuestra que de verdad funciona.** *"Recorre este proyecto entregable por entregable frente a la especificación del módulo en <https://learnwisej.com/assets/courses/azure-docker/specs/module-10.md> y muéstrame la prueba de que cada uno está realmente terminado — la ruta del código, dónde hacer clic y qué debería ver. Si un entregable solo se cumple en parte, dilo claramente en lugar de justificarlo. Después añade lo que falte."*
- ✓ **Revísalo como un pull request.** *"Explícame este código de EnterpriseOps como si estuvieras revisando el pull request de un compañero y no lo hubieras escrito tú. ¿Dónde vive el estado?, ¿qué ocurre de principio a fin con una acción del usuario?, ¿qué devolverías para que lo cambien? y ¿qué partes no te sentirías cómodo poniendo delante de un cliente? Sé concreto y cita las líneas a las que te refieres — no me resumas el diseño."*
- ✓ **Cambia una pieza sin mover las juntas.** *"Elige la parte de este proyecto EnterpriseOps que sea más claramente un sustituto de algo real — una fuente de datos falsa, un almacén en memoria, un servicio simplificado — y reemplázala por una implementación con forma de producción. El contrato a ambos lados no debe cambiar: los mismos tipos, las mismas llamadas, el mismo comportamiento ante fallos. Dime qué te obligó a gestionar la implementación real que el sustituto no gestionaba."*



Llévalo al laboratorio

El laboratorio práctico que cierra este módulo pide los mismos entregables que acabas de hacer que generara el asistente. Usa lo que tienes aquí como forma de tu respuesta y como prueba que la respalda — no como la respuesta en sí. El laboratorio se califica por tu razonamiento, y esa parte tiene que ser tuya.