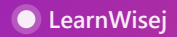


## Ejercicio de programación con IA

Construye el ejemplo AccessOps 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.



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

Todas las demás secciones de este módulo explican cómo funciona «Protección de datos, subidas, descargas y handlers». 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

Protege la función de documentos de AccessOps: subidas limitadas por tamaño y tipo con análisis del contenido, almacenadas fuera de la raíz web con un nombre aleatorio y vinculadas al departamento propietario; descargas que pasan por la autorización de DocumentService y por Application.Download; un handler personalizado para un enlace compartido firmado que caduca en una hora; y una revisión de que todas las consultas están parametrizadas y de que toda cadena proporcionada por el usuario se codifica antes de mostrarse.

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

- ✓ Canal de subida validado con carpeta de cuarentena
- ✓ Descarga autorizada a través del servicio
- ✓ Handler de enlaces compartidos firmados y con caducidad
- ✓ Revisión de consultas parametrizadas y codificación de la salida
- ✓ Prueba con una solicitud de descarga falsificada

### Por qué tiene esta forma

El área de documentos confidenciales es la parte de AccessOps que de verdad quiere un atacante: cartas salariales, expedientes disciplinarios, contratos firmados, documentación del consejo. Todas las demás pantallas mueven filas entre cuadrículas; esta mueve bytes a través del límite hacia un navegador que no controlas, y de vuelta. La identidad y los permisos quedaron resueltos en los módulos anteriores, y este trata de los lugares por donde un archivo, una URL o una cadena los esquivan.

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                                                                                                                                                                                                              |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| descarga transmitida por el servidor   | Llamar a <code>Application.Download</code> desde código del servidor transmite un archivo al navegador durante un evento que ya has autorizado, así que no tiene una dirección que guardar en marcadores, compartir o adivinar. |
| exposición de archivos estáticos       | Todo lo que está bajo la raíz web se sirve antes de que se ejecute una sola línea de tu código, así que los permisos del archivo son exactamente «quien conozca la URL».                                                        |
| evento Uploaded                        | El control Upload lo lanza en el servidor con los flujos de los archivos entrantes: el único lugar donde se puede inspeccionar un archivo antes de que entre en tu sistema.                                                     |
| comprobación de la firma del contenido | Comparar los primeros bytes de un archivo con la cabecera conocida del formato, porque la extensión y el tipo de contenido declarado los proporciona el cliente.                                                                |
| almacenamiento en cuarentena           | Guardar la subida aceptada fuera de la raíz web con un nombre generado hasta que el análisis salga limpio, y solo entonces vincularla al departamento propietario.                                                              |
| handler HTTP personalizado             | Una clase del servidor registrada en la aplicación que recibe solicitudes en bruto fuera del flujo de eventos, así que no tiene identidad de sesión y debe autenticar y validar todo por sí misma.                              |
| enlace firmado con caducidad           | Una URL que lleva un id de documento, una caducidad y un HMAC calculado sobre ambos con una clave guardada en el servidor, para que los valores no puedan editarse y el enlace deje de funcionar por sí solo.                   |

## 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 5. Puedes abrir tú mismo [module-05.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 AccessOps 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 5 – «Protección de datos, subidas, descargas y handlers» – del curso de LearnWisej «Patrones de seguridad y autenticación», 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/security-and-auth-patterns/specs/module-05.md>

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

1. Proyecto «AccessOps», 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: Protege la función de documentos de AccessOps: subidas limitadas por tamaño y tipo con análisis del contenido, almacenadas fuera de la raíz web con un nombre aleatorio y vinculadas al departamento propietario; descargas que pasan por la autorización de `DocumentService` y por `Application.Download`; un handler personalizado para un enlace compartido firmado que caduca en una hora; y una revisión de que todas las consultas están parametrizadas y de que toda cadena proporcionada por el usuario se codifica antes de mostrarse.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
  - Canal de subida validado con carpeta de cuarentena
  - Descarga autorizada a través del servicio
  - Handler de enlaces compartidos firmados y con caducidad
  - Revisión de consultas parametrizadas y codificación de la salida
  - Prueba con una solicitud de descarga falsificada
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:

- Tu controlador de subidas comprueba la extensión con una lista de permitidas y el archivo la supera. Explica qué puede enviar todavía un atacante y qué aporta la comprobación de la firma que la comprobación de la extensión no puede aportar.
- Un responsable reenvía un enlace compartido a alguien de fuera de la empresa antes de que caduque. ¿Qué partes de tu diseño limitan el daño y qué tendrías que añadir para revocar ese único enlace sin cambiar la clave para todos?
- Tu botón de descarga está oculto para los usuarios sin permiso de lectura, y la prueba de la solicitud falsificada sigue pasando. Explica con precisión qué línea de código hace que pase y por qué el botón oculto no tiene nada que ver.

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:

- ✓ Tu controlador de subidas comprueba la extensión con una lista de permitidas y el archivo la supera. Explica qué puede enviar todavía un atacante y qué aporta la comprobación de la firma que la comprobación de la extensión no puede aportar.
- ✓ Un responsable reenvía un enlace compartido a alguien de fuera de la empresa antes de que caduque. ¿Qué partes de tu diseño limitan el daño y qué tendrías que añadir para revocar ese único enlace sin cambiar la clave para todos?
- ✓ Tu botón de descarga está oculto para los usuarios sin permiso de lectura, y la prueba de la solicitud falsificada sigue pasando. Explica con precisión qué línea de código hace que pase y por qué el botón oculto no tiene nada que ver.

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.



### Todavía no hay un ejemplo de referencia para este curso — el tuyo es el primero

El [repositorio de ejemplos de LearnWisej](#) tiene soluciones completas para la mayoría de los cursos, pero todavía no para este. Eso significa que no hay nada con qué compararte ni ningún atajo: la especificación y tu propia revisión son toda tu red de seguridad. Guarda lo que construyas — y si sale bien, nos encantaría verlo.

## Tres prompts para ir más allá

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

- ✓ **Añade el segundo caso.** *"Amplía este proyecto AccessOps con un segundo caso que siga el mismo patrón que el primero — otra entidad, pantalla o control con la misma forma, conectado de la misma manera. Extrae a una clase base o a un servicio lo que de verdad sea común, pero solo donde se justifique; dos cosas parecidas no siempre piden una abstracción. Muéstrame el diff y justifica cada cosa que hayas extraído."*
- ✓ **Demuestra que de verdad funciona.** *"Recorre este proyecto entregable por entregable frente a la especificación del módulo en <https://learnwisej.com/assets/courses/security-and-auth-patterns/specs/module-05.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 AccessOps 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."



### 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.