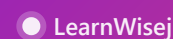


Ejercicio de programación con IA

Construye el ejemplo IconDesk 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 «Recursos incrustados y resource.wx». 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 a la aplicación IconDesk una carpeta Resources con logo.svg, status-ok.svg, status-warning.svg, photo.png y badge.gif, y marca todos ellos con Build Action Embedded Resource. Construye una página ResourceLab que muestre los cinco mediante orígenes de imagen resource.wx, escribe uno de esos orígenes con el nombre del ensamblado de la aplicación delante del archivo para que la búsqueda no pueda ser ambigua, y deja el resto sin calificar. Después coloca un archivo de reemplazo junto a la aplicación desplegada para sobrescribir un recurso incrustado y captura el antes y el después. Termina escribiendo mal a propósito una URL de recurso y anotando los pasos que identificaron el fallo.

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

- ✓ Carpeta Resources con logo.svg, status-ok.svg, status-warning.svg, photo.png y badge.gif, todos marcados como Embedded Resource
- ✓ Página ResourceLab que muestra cada recurso incrustado mediante un origen de imagen resource.wx
- ✓ Un origen de imagen calificado con el nombre del ensamblado de la aplicación y el resto sin calificar
- ✓ Archivo desplegado que sobrescribe un recurso incrustado, capturado antes y después
- ✓ Nota de resolución de problemas con los pasos que identificaron una URL de recurso mal escrita

Por qué tiene esta forma

Hasta ahora, todos los recursos de IconDesk han sido algo que la aplicación encontró en disco o le pidió al tema. Eso funciona hasta que alguien despliega el sitio y se olvida de una carpeta: el navegador pide un icono, el servidor responde 404 y el usuario ve un rectángulo vacío donde debería estar el distintivo de estado. Nadie lo detecta en la revisión, porque la máquina del desarrollador sigue teniendo el archivo.

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
Build Action = Embedded Resource	La propiedad de archivo de Visual Studio que compila un archivo dentro del manifiesto del ensamblado en lugar de copiarlo junto al ejecutable, de modo que el recurso se distribuye dentro de la DLL.
controlador resource.wx	El controlador HTTP de Wisej.NET que sirve recursos incrustados al navegador; lo que asignas es una cadena de origen de imagen, nunca un objeto de imagen del lado del servidor.
URL de recurso sin calificar	resource.wx/company-logo.svg le pide a Wisej.NET que busque en los ensamblados cargados un recurso con ese nombre, lo cual es cómodo y ligeramente ambiguo.
URL de recurso calificada con el ensamblado	resource.wx/MyCompany.Branding/company-logo.svg nombra el ensamblado (sin.dll) para que la búsqueda no pueda caer en el archivo de otro con el mismo nombre.
Tipos de recurso permitidos	El controlador documenta exactamente nueve extensiones:.css,.js,.html,.jpg,.jpeg,.png,.gif,.svg y.bmp; cualquier otra cosa vuelve como un 404.
Sobrescritura por archivo desplegado	Un archivo desplegado con la aplicación se puede servir en lugar del recurso incrustado en la misma ruta, lo que permite a una instalación cambiar la marca o parchear una ilustración sin recompilar el ensamblado.
Nombre de recurso del manifiesto	El nombre que.NET almacena realmente, derivado del nombre del ensamblado, el espacio de nombres predeterminado y la carpeta en la que está el archivo; adivinarlo es la razón habitual de que una URL de recurso no devuelva nada.

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 IconDesk 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 - «Recursos incrustados y resource.wx» - del curso de LearnWisej «Iconos e imágenes en Wisej.NET», 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/icons-and-images/specs/module-05.md>

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

1. Proyecto «IconDesk», 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 a la aplicación IconDesk una carpeta Resources con `logo.svg`, `status-ok.svg`, `status-warning.svg`, `photo.png` y `badge.gif`, y marca todos ellos con Build Action Embedded Resource. Construye una página ResourceLab que muestre los cinco mediante orígenes de imagen `resource.wx`, escribe uno de esos orígenes con el nombre del ensamblado de la aplicación delante del archivo para que la búsqueda no pueda ser ambigua, y deja el resto sin calificar. Después coloca un archivo de reemplazo junto a la aplicación desplegada para sobrescribir un recurso incrustado y captura el antes y el después. Termina escribiendo mal a propósito una URL de recurso y anotando los pasos que identificaron el fallo.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Carpeta Resources con `logo.svg`, `status-ok.svg`, `status-warning.svg`, `photo.png` y `badge.gif`, todos marcados como Embedded Resource
 - Página ResourceLab que muestra cada recurso incrustado mediante un origen de imagen `resource.wx`
 - Un origen de imagen calificado con el nombre del ensamblado de la aplicación y el resto sin calificar
 - Archivo desplegado que sobrescribe un recurso incrustado, capturado antes y después
 - Nota de resolución de problemas con los pasos que identificaron una URL de recurso mal escrita
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:

- Dos ensamblados referenciados incrustan ambos un archivo llamado `logo.svg`. ¿A qué se resuelve un origen sin calificar y qué cambiarías para que la respuesta deje de depender del orden de carga?
- Tu sobrescritura funciona en el servidor de pruebas pero no después de la siguiente

versión. ¿Qué partes del proceso de despliegue podrían haberla eliminado y dónde debería haberse registrado?

– Un compañero incrusta `icon.ico` e informa de que `resource.wx` está roto. ¿Qué está pasando en realidad y cuál es el cambio más pequeño que lo arregla?

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:

- ✓ Dos ensamblados referenciados incrustan ambos un archivo llamado `logo.svg`. ¿A qué se resuelve un origen sin calificar y qué cambiarías para que la respuesta deje de depender del orden de carga?
- ✓ Tu sobrescritura funciona en el servidor de pruebas pero no después de la siguiente versión. ¿Qué partes del proceso de despliegue podrían haberla eliminado y dónde debería haberse registrado?
- ✓ Un compañero incrusta `icon.ico` e informa de que `resource.wx` está roto. ¿Qué está pasando en realidad y cuál es el cambio más pequeño que lo arregla?

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 IconDesk 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/icons-and-images/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."*
- ✓ **Revisalo como un pull request.** *"Explicame este código de IconDesk 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.