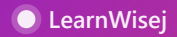


Ejercicio de programación con IA

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.



Escribir el código ya es la mitad fácil.

Conseguir que un asistente moderno genere código que funcione para «Packs de iconos de Wisej.NET y NuGet» solo requiere un buen prompt. Juzgarlo, corregirlo y ampliarlo requiere todo lo que ha cubierto este módulo. Ese es el reparto sobre el que se construye este ejercicio: tú aportas la especificación y la revisión; el asistente, el tecleo.

Qué estás pidiendo

Añade a IconDesk dos paquetes NuGet oficiales de packs de iconos Wisej-4 y construye una página IconCompare que muestre los mismos cinco conceptos, Save, Delete, Search, User y Settings, tomados de ambas familias para que la diferencia visual sea imposible de pasar por alto. Elige al menos cinco de los iconos con el explorador de imágenes de Visual Studio en lugar de escribir la ruta, asigna el resto desde el catálogo del pack en C#, recolorea dos de ellos con el sufijo ?color= y observa el panel de red para ver cómo los archivos de icono llegan petición a petición y después vienen de la caché del navegador. Cierra con una recomendación breve que nombre la familia de iconos principal del proyecto.

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

- ✓ Dos paquetes NuGet oficiales de packs de iconos Wisej-4 referenciados por IconDesk
- ✓ Página IconCompare que muestra Save, Delete, Search, User y Settings de ambos packs
- ✓ Al menos cinco iconos elegidos con el explorador de imágenes de Visual Studio en lugar de escritos a mano
- ✓ Dos iconos recoloreados con el sufijo de color y una nota sobre qué ilustración aceptó el recoloreado
- ✓ Recomendación de un párrafo que nombre la familia de iconos principal del proyecto, con las pruebas de caché del navegador que la respaldan

Por qué tiene esta forma

Tres módulos después, IconDesk ya sabe mostrar una imagen de cinco formas distintas, pero todos los iconos que tiene se dibujaron o se descargaron a mano. Una aplicación real necesita un icono de Save, uno de Delete, uno de Search, uno de User y uno de Settings el primer día, y otros cuarenta antes de que acabe el trimestre, todos en el mismo lenguaje visual. Copiar archivos SVG de uno en uno produce una carpeta de ilustraciones inconexas, grosores de trazo incoherentes y ningún rastro de licencias.

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
pack de iconos	Un ensamblado.NET corriente cuyos archivos de icono, en su mayoría SVG, están compilados dentro como recursos incrustados; referenciarlo añade un solo archivo a tu despliegue por muchos iconos que contenga.
recurso incrustado	Un archivo compilado dentro del ensamblado en lugar de copiado junto a él, de modo que los iconos no se pueden perder, renombrar ni olvidar cuando la aplicación se despliega en otro servidor.
resource.wx	El controlador de Wisej.NET que sirve un recurso incrustado por HTTP; el origen de imagen de un pack de iconos es una URL resource.wx que nombra el ensamblado del pack y el archivo que hay dentro.
catálogo de iconos	La clase estática que publica un pack para que quien la usa obtenga IntelliSense y un nombre comprobado en compilación en lugar de una ruta escrita a mano; el espacio de nombres y el nombre exacto de la clase varían de pack a pack.
petición por icono	Los iconos de un pack se obtienen de forma individual, en el primer uso, y después se sirven desde la caché del navegador, que es lo que hace asumible un catálogo de miles de iconos.
precarga de imágenes de tema	Las imágenes de tema viajan con el tema y ya están en la caché de temas del cliente cuando arranca la aplicación, así que se comportan de forma muy distinta a los iconos de un pack en el arranque.
familia de iconos principal	El único pack al que se compromete un producto para sus comandos del día a día, elegido de forma deliberada por cobertura, estilo y licencia en lugar de acumulado pack a pack.

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 4. Puedes abrir tú mismo [module-04.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 4 – «Packs de iconos de Wisej.NET y NuGet» – 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-04.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 IconDesk dos paquetes NuGet oficiales de packs de iconos Wisej-4 y construye una página IconCompare que muestre los mismos cinco conceptos, Save, Delete, Search, User y Settings, tomados de ambas familias para que la diferencia visual sea imposible de pasar por alto. Elige al menos cinco de los iconos con el explorador de imágenes de Visual Studio en lugar de escribir la ruta, asigna el resto desde el catálogo del pack en C#, recolorea dos de ellos con el sufijo `?color=` y observa el panel de red para ver cómo los archivos de icono llegan petición a petición y después vienen de la caché del navegador. Cierra con una recomendación breve que nombre la familia de iconos principal del proyecto.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Dos paquetes NuGet oficiales de packs de iconos Wisej-4 referenciados por IconDesk
 - Página IconCompare que muestra Save, Delete, Search, User y Settings de ambos packs
 - Al menos cinco iconos elegidos con el explorador de imágenes de Visual Studio en lugar de escritos a mano
 - Dos iconos recoloreados con el sufijo de color y una nota sobre qué ilustración aceptó el recoloreado
 - Recomendación de un párrafo que nombre la familia de iconos principal del proyecto, con las pruebas de caché del navegador que la respaldan
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:

- Referenciar un pack de tres mil iconos añade un ensamblado al despliegue. ¿Qué le cuesta eso al navegador, y cuándo?
- Uno de tus dos iconos recoloreados ignoró el sufijo de color. ¿Qué tiene esa ilustración y qué cambiarías en los criterios de selección del pack por ese motivo?
- Un compañero quiere que todos los iconos de comando sigan al tema cuando la aplicación

cambia de tema. ¿Cuáles de los cinco iconos de IconCompare deberían pasar a ser imágenes de tema y cuáles no?

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:

- ✓ Referenciar un pack de tres mil iconos añade un ensamblado al despliegue. ¿Qué le cuesta eso al navegador, y cuándo?
- ✓ Uno de tus dos iconos recoloreados ignoró el sufijo de color. ¿Qué tiene esa ilustración y qué cambiarías en los criterios de selección del pack por ese motivo?
- ✓ Un compañero quiere que todos los iconos de comando sigan al tema cuando la aplicación cambia de tema. ¿Cuáles de los cinco iconos de IconCompare deberían pasar a ser imágenes de tema y cuáles no?

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.

- ✓ **Haz que aguante un mal día.** *"Revisa este proyecto IconDesk en busca de los modos de fallo que todavía no gestiona — una entrada incorrecta o ausente, una dependencia lenta o no disponible, trabajo que llega después de que el usuario haya salido de la pantalla y dos acciones que compiten entre sí. Para cada uno, dime primero si es un riesgo real en Wisej.NET 4 y por qué, y solo después escribe la corrección. Si decides que algo no necesita gestión, dilo y explica el razonamiento en lugar de añadir código defensivo."*
- ✓ **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-04.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."*



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.