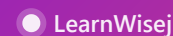


Ejercicio de programación con IA

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.



No vas a teclear este ejemplo. Vas a dirigirlo.

Ya casi nadie escribe código Wisej.NET partiendo de un archivo vacío — un asistente escribe el primer borrador y el desarrollador decide si es correcto. Eso solo funciona si dominas el tema, y eso es lo que te dio el resto de este módulo. Aquí le pasas ese conocimiento a ChatGPT o a Claude, recibes el ejemplo de este módulo como una solución de Visual Studio que funciona y lo revisas como revisarías el pull request de un compañero.

Qué estás pidiendo

Crea el proyecto IconDesk y dale un panel de comandos con los botones Open, Save, Print y Delete, cada uno con su icono a través de un mecanismo distinto. Carga btnOpen desde un System.Drawing.Image del lado del servidor, asigna una imagen de tema con nombre a btnSave.ImageSource y apunta btnPrint y btnDelete a una misma ImageList compartida mediante ImageKey. Añade una etiqueta de diagnóstico que inspeccione cada botón a través de Image e indique qué mecanismo está realmente en juego, después cambia la imagen almacenada bajo una clave de la ImageList y comprueba que todos los controles que usan esa clave siguen el cambio.

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

- ✓ Panel de comandos de IconDesk con los botones Open, Save, Print y Delete
- ✓ btnOpen con un System.Drawing.Image cargado en el servidor y btnSave con una imagen de tema con nombre en ImageSource
- ✓ btnPrint y btnDelete compartiendo una misma ImageList mediante ImageKey en lugar de ImageIndex
- ✓ Etiqueta de diagnóstico que lee cada botón a través de Image y nombra el mecanismo que hay detrás de su icono
- ✓ Nota de laboratorio que recoja qué ocurrió al sustituir una clave de la ImageList y al asignar dos propiedades de imagen en el mismo botón

Por qué tiene esta forma

Wisej.NET ofrece a propósito más de una forma de poner un gráfico sobre un control, y todas funcionan. El problema empieza cuando un equipo las usa todas por accidente: una barra de herramientas en la que un botón lee un archivo del disco del servidor, el siguiente apunta a un nombre de tema y un tercero perdió

su icono en silencio porque se asignaron dos propiedades al mismo control. Seis meses después, nadie sabe explicar de dónde sale realmente una imagen determinada ni por qué desapareció tras el despliegue. 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
Propiedad Image	Un objeto System.Drawing.Image que vive en la memoria del servidor; Wisej.NET lo envía al navegador como datos PNG, así que la imagen viaja como bytes que tu código ya decodificó.
Cadena ImageSource	Una cadena que identifica un recurso para que lo resuelva el cliente: una URL relativa, una URL absoluta, una imagen de tema con nombre, una ruta resource.wx o un elemento dentro del ensamblado de un pack de iconos.
IconSource	La misma idea de cadena en los controles que tienen un hueco de icono propio junto a su imagen principal, así que revisa la API del control que estés usando en lugar de suponer que todos los controles llaman igual a la propiedad.
Contrato Image	La familia común de imagen que exponen los controles capaces de mostrar imágenes — Image, ImageAlign, ImageIndex, ImageKey, ImageList e ImageSource —, que permite que un único método auxiliar decore de forma uniforme un Button, una Label o cualquier otra cosa.
ImageList	Una colección reutilizable de objetos de imagen a la que pueden apuntar varios controles, direccionada por posición o por nombre, y capaz de fijar un tamaño de imagen común para los controles que la usan.
ImageKey frente a ImageIndex	Una clave como "delete" transmite significado y sobrevive a los reordenamientos; el índice 3 pasa a ser en silencio la imagen equivocada el día en que alguien inserta una imagen por encima.
Asignación mutuamente excluyente	Todas estas propiedades alimentan un único hueco visual, así que asignar una puede restablecer las demás, y por eso el icono equivocado suele ser una segunda asignación y no un archivo roto.

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 1. Puedes abrir tú mismo [module-01.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



Usar Claude

Abre [claude.ai](#) con el mismo prompt ya cargado. Si usas Claude Code, puedes

Codex; cualquier plan que pueda navegar por la web descargará la especificación.

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 1 – «El modelo de imágenes de Wisej.NET: Image, ImageSource e ImageList» – 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-01.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: Crea el proyecto IconDesk y dale un panel de comandos con los botones Open, Save, Print y Delete, cada uno con su icono a través de un mecanismo distinto. Carga `btnOpen` desde un `System.Drawing.Image` del lado del servidor, asigna una imagen de tema con nombre a `btnSave.ImageSource` y apunta `btnPrint` y `btnDelete` a una misma `ImageList` compartida mediante `ImageKey`. Añade una etiqueta de diagnóstico que inspeccione cada botón a través de `IImage` e indique qué mecanismo está realmente en juego, después cambia la imagen almacenada bajo una clave de la `ImageList` y comprueba que todos los controles que usan esa clave siguen el cambio.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Panel de comandos de IconDesk con los botones Open, Save, Print y Delete
 - `btnOpen` con un `System.Drawing.Image` cargado en el servidor y `btnSave` con una imagen de tema con nombre en `ImageSource`
 - `btnPrint` y `btnDelete` compartiendo una misma `ImageList` mediante `ImageKey` en lugar de `ImageIndex`
 - Etiqueta de diagnóstico que lee cada botón a través de `IImage` y nombra el mecanismo que hay detrás de su icono
 - Nota de laboratorio que recoja qué ocurrió al sustituir una clave de la `ImageList` y al asignar dos propiedades de imagen en el mismo botón
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 etiqueta de diagnóstico dice que `btnDelete` usa un origen de imagen, pero tú solo estableciste su `ImageKey`. ¿Qué se ha tenido que ejecutar y dónde lo buscarías?
- Tanto imprimir como eliminar cambiaron al sustituir una entrada de la `ImageList`. ¿Seguiría siendo así si los botones usaran `ImageIndex`, y qué pasaría si se insertara una quinta imagen al principio de la lista?

– El botón Open lee su archivo en el servidor. ¿Qué parte de ese montaje se rompe cuando la aplicación se despliega en otra máquina, y qué parte no se rompería si esa misma imagen fuera una cadena de origen de imagen?

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 etiqueta de diagnóstico dice que btnDelete usa un origen de imagen, pero tú solo estableciste su ImageKey. ¿Qué se ha tenido que ejecutar y dónde lo buscarías?
- ✓ Tanto imprimir como eliminar cambiaron al sustituir una entrada de la ImageList. ¿Seguiría siendo así si los botones usaran ImageIndex, y qué pasaría si se insertara una quinta imagen al principio de la lista?
- ✓ El botón Open lee su archivo en el servidor. ¿Qué parte de ese montaje se rompe cuando la aplicación se despliega en otra máquina, y qué parte no se rompería si esa misma imagen fuera una cadena de origen de imagen?

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.

- ✓ **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-01.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."

- ✓ **Cambia una pieza sin mover las juntas.** *"Elige la parte de este proyecto IconDesk 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.