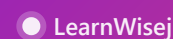


Ejercicio de programación con IA

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 asistente pone las manos rápidas. El arquitecto sigues siendo tú.

Un asistente puede generar el ejemplo de este módulo en minutos. Lo que no puede hacer es decidir si el resultado es el diseño adecuado para tu aplicación — esa decisión sigue siendo tuya. Tú pones el criterio: dale unas buenas instrucciones y luego contrasta lo que te devuelva con lo que te acaba de enseñar el Módulo 2.

Qué estás pidiendo

Construye una página ImageLab en IconDesk que haga visibles las dos canalizaciones una al lado de la otra. Muestra un mismo recurso ráster cuatro veces con SizeMode en Normal, Zoom, Cover y StretchImage para que las diferencias de recorte y distorsión resulten evidentes, añade un PictureBox que traiga un JPEG remoto con LoadAsync y mantenga la página receptiva mientras llega, y coloca un SVG asignado mediante ImageSource a 16, 24, 32 y 48 píxeles para demostrar que se mantiene nítido. Termina con un manejador que informe, para cada PictureBox de la página, si su imagen vino de Image o de ImageSource.

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

- ✓ Página ImageLab que muestra un mismo recurso ráster en los modos de tamaño Normal, Zoom, Cover y StretchImage
- ✓ PictureBox que obtiene un JPEG remoto con LoadAsync sin bloquear la página
- ✓ Un SVG asignado mediante ImageSource y renderizado a 16, 24, 32 y 48 píxeles
- ✓ Manejador de informe que indica, para cada PictureBox, si la imagen la suministró Image o ImageSource
- ✓ Nota de formatos que enumere qué recursos del laboratorio viajan por System.Drawing y cuáles siguen siendo cadenas de recurso

Por qué tiene esta forma

«¿Qué formatos de imagen admite Wisej.NET?» es la pregunta que hace todo equipo en la primera semana, y la respuesta honesta depende de qué vía de la API tome la imagen. Si la contestas con una lista vaga, acabas con un logotipo estirado en un portátil de alta densidad, con una página que se congela mientras se descarga la foto de un producto desde un servidor lento y con un informe de error porque un archivo.ico no se resuelve a través de una cadena de recurso. Este módulo sustituye la lista vaga por dos listas precisas, cada una asociada a la propiedad de control que la consume.

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
Canalización de System.Drawing	La propiedad Image contiene un System.Drawing.Image en la memoria del servidor; Wisej.NET lo envía al navegador como datos PNG. Sus orígenes documentados son archivos bitmap, metafile, icon, JPEG, GIF y PNG.
Canalización de orígenes de imagen	Una cadena que el cliente resuelve por su cuenta: una URL relativa o absoluta, una imagen de tema con nombre o una ruta resource.wx. Las extensiones de imagen documentadas para resource.wx son.jpg,.jpeg,.png,.gif,.svg y.bmp.
PictureBox.Image	El punto de entrada para los píxeles que tu código C# ya tiene o acaba de generar, como un gráfico dibujado con Graphics.
PictureBox.ImageSource	El punto de entrada para una imagen que el navegador puede obtener por sí mismo. Viaja una cadena corta en lugar de los bytes de la imagen, y por eso SVG pertenece aquí.
PictureBox.SizeMode	Los seis valores Normal, AutoSize, CenterImage, StretchImage, Zoom y Cover, que deciden si una imagen se recorta, se centra, se deforma, se ajusta o se recorta por los bordes.
Load y LoadAsync	Load(url) actualiza la imagen desde una URL de forma síncrona; LoadAsync(url) hace la misma petición de forma asíncrona, así que un servidor lento no bloquea la pantalla.
Vectorial frente a ráster	Un SVG es marcado que se dibuja al tamaño que pida el diseño, y una ilustración monocroma adecuada se puede recolorar; un archivo ráster es una rejilla fija de píxeles que solo puede remuestrearse.

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 2. Puedes abrir tú mismo [module-02.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 2 – «Bitmaps, SVG y PictureBox: formatos, tamaño y carga» – 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-02.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: Construye una página ImageLab en IconDesk que haga visibles las dos canalizaciones una al lado de la otra. Muestra un mismo recurso ráster cuatro veces con `SizeMode` en Normal, Zoom, Cover y `StretchImage` para que las diferencias de recorte y distorsión resulten evidentes, añade un `PictureBox` que traiga un JPEG remoto con `LoadAsync` y mantenga la página receptiva mientras llega, y coloca un SVG asignado mediante `ImageSource` a 16, 24, 32 y 48 píxeles para demostrar que se mantiene nítido. Termina con un manejador que informe, para cada `PictureBox` de la página, si su imagen vino de `Image` o de `ImageSource`.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Página ImageLab que muestra un mismo recurso ráster en los modos de tamaño Normal, Zoom, Cover y `StretchImage`
 - `PictureBox` que obtiene un JPEG remoto con `LoadAsync` sin bloquear la página
 - Un SVG asignado mediante `ImageSource` y renderizado a 16, 24, 32 y 48 píxeles
 - Manejador de informe que indica, para cada `PictureBox`, si la imagen la suministró `Image` o `ImageSource`
 - Nota de formatos que enumere qué recursos del laboratorio viajan por `System.Drawing` y cuáles siguen siendo cadenas de recurso
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 recurso ráster es más ancho que alto y el cuadro es cuadrado. ¿Cuál de Zoom y Cover pierde parte de la imagen, cuál deja espacio vacío y cuál defenderías para el logotipo de un cliente?
- El servidor del JPEG remoto está inaccesible durante diez segundos. ¿Qué experimenta el usuario de tu página con `LoadAsync` y qué habría experimentado con el Load síncrono?
- Tu manejador de informe dice que la imagen de un control vino de `ImageSource`, pero el

archivo del diseñador también asigna Image en ese mismo control. ¿Cuál se dibuja realmente y cómo lo demostrarías en lugar de adivinarlo?

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 recurso ráster es más ancho que alto y el cuadro es cuadrado. ¿Cuál de Zoom y Cover pierde parte de la imagen, cuál deja espacio vacío y cuál defenderías para el logotipo de un cliente?
- ✓ El servidor del JPEG remoto está inaccesible durante diez segundos. ¿Qué experimenta el usuario de tu página con LoadAsync y qué habría experimentado con el Load síncrono?
- ✓ Tu manejador de informe dice que la imagen de un control vino de ImageSource, pero el archivo del diseñador también asigna Image en ese mismo control. ¿Cuál se dibuja realmente y cómo lo demostrarías en lugar de adivinarlo?

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.

- ✓ **Revisalo como un pull request.** *"Explícame 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."

- ✓ **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."*



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.