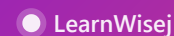


Ejercicio de programación con IA

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.



Instrúyelo, léelo y luego rómpelo.

Este ejercicio tiene tres partes, y solo la primera trata del prompt. Le das al asistente la especificación del módulo y recibes una solución IconDesk; lees lo que escribió y decides si de verdad hace lo que pedía el módulo; y después lo llevas más allá de lo que pedía el módulo. En la segunda y la tercera parte es donde se aprende.

Qué estás pidiendo

Rellena una página IconGallery en IconDesk usando el Image Selector del diseñador en lugar de escribir las cadenas de origen a mano. Asigna imágenes a cinco controles: una imagen de tema con nombre, un SVG del proyecto referenciado por URL relativa, una URL absoluta y al menos dos recursos de un pack de iconos si hay un pack instalado. Recolorea un SVG monocromo con el sufijo ?color= y un nombre de color del tema, abre el archivo Designer.cs generado y lee las cadenas de origen de imagen que produjo el selector, después cambia la aplicación entre dos temas de Wisej y anota qué recursos cambiaron y cuáles se quedaron exactamente igual.

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

- ✓ Página IconGallery con cinco controles cuyas imágenes se asignaron con el Image Selector del diseñador
- ✓ Entre ellas, una imagen de tema con nombre, un SVG relativo del proyecto, una URL absoluta y al menos dos recursos de un pack de iconos
- ✓ SVG monocromo recolorado con el sufijo?color= y un nombre de color del tema
- ✓ Extracto de Designer.cs que muestre las cadenas de origen de imagen generadas por el selector
- ✓ Nota del cambio de tema que recoja qué recursos siguieron al tema y cuáles se mantuvieron fijos

Por qué tiene esta forma

IconDesk ya tiene un panel de comandos y un laboratorio de imágenes, y todas sus imágenes son un archivo desplegado más una ruta que alguien escribió. Eso funciona hasta que la aplicación cambia de tema, se despliega bajo un directorio virtual distinto o tiene que mostrar el mismo símbolo en color de advertencia en una pantalla y en un color neutro en otra. Este módulo saca el origen de una imagen de la carpeta del proyecto y lo lleva al tema, a una URL o al diseñador, para que la cadena que lleva un control siga significando algo después de que cambien la ilustración, el tema o el servidor.

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
imagen de tema con nombre	Un nombre semántico como icon-print que el tema activo asocia a una ilustración, de modo que otro tema puede aportar otra imagen sin ningún cambio en C#.
caché de temas del cliente	Las imágenes de tema viajan con el tema y ya están en el navegador cuando un control pide una, lo que convierte al tema en el hogar adecuado para un conjunto pequeño de símbolos de comando y en el hogar equivocado para miles de ellos.
origen con URL relativa	Una cadena de origen de imagen resuelta con respecto a la aplicación: la forma más sencilla para un logotipo desplegado con la aplicación.
origen con URL absoluta	Una dirección externa completa que convierte la imagen en una dependencia de otro sistema, con consecuencias de disponibilidad, privacidad, CORS y Content Security Policy.
imagen base64 incrustada	Los bytes de la imagen escritos dentro de la propia definición del tema, de modo que la ilustración no necesita una segunda petición, a costa de un tema más pesado.
Image Selector	El selector de tiempo de diseño de Visual Studio que explora las imágenes del proyecto, las del tema y los recursos de los packs de iconos referenciados, y después serializa la cadena de origen elegida en Designer.cs.
sufijo?color=	Un sufijo en un origen de imagen que le pide a Wisej.NET que recoloree un SVG monocromo compatible con un color CSS o un nombre de color del tema.

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 3. Puedes abrir tú mismo [module-03.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 3 – «Imágenes de tema, URL, color de SVG y el selector de imágenes del diseñador» – 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-03.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: Rellena una página `IconGallery` en `IconDesk` usando el `Image Selector` del diseñador en lugar de escribir las cadenas de origen a mano. Asigna imágenes a cinco controles: una imagen de tema con nombre, un SVG del proyecto referenciado por URL relativa, una URL absoluta y al menos dos recursos de un pack de iconos si hay un pack instalado. Recolorea un SVG monocromo con el sufijo `?color=` y un nombre de color del tema, abre el archivo `Designer.cs` generado y lee las cadenas de origen de imagen que produjo el selector, después cambia la aplicación entre dos temas de Wisej y anota qué recursos cambiaron y cuáles se quedaron exactamente igual.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Página `IconGallery` con cinco controles cuyas imágenes se asignaron con el `Image Selector` del diseñador
 - Entre ellas, una imagen de tema con nombre, un SVG relativo del proyecto, una URL absoluta y al menos dos recursos de un pack de iconos
 - SVG monocromo recolorado con el sufijo `?color=` y un nombre de color del tema
 - Extracto de `Designer.cs` que muestre las cadenas de origen de imagen generadas por el selector
 - Nota del cambio de tema que recoja qué recursos siguieron al tema y cuáles se mantuvieron fijos
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:

- La galería se renderiza correctamente en tu máquina y no muestra nada para la URL absoluta en la de un compañero. ¿Cuál de los cinco mecanismos podría producir esa diferencia y qué comprobarías primero?
- Dos de tus iconos se niegan a cambiar de color cuando añades el sufijo. ¿Qué hay dentro de esos archivos y cuándo editarlos es la respuesta equivocada?

– Un diseñador pide el mismo símbolo en la galería, en una barra de herramientas y en una celda de cuadrícula. ¿Qué cuesta menos cambiar después, la clave de tema, la URL relativa o el recurso del pack, y por qué?

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:

- ✓ La galería se renderiza correctamente en tu máquina y no muestra nada para la URL absoluta en la de un compañero. ¿Cuál de los cinco mecanismos podría producir esa diferencia y qué comprobarías primero?
- ✓ Dos de tus iconos se niegan a cambiar de color cuando añades el sufijo. ¿Qué hay dentro de esos archivos y cuándo editarlos es la respuesta equivocada?
- ✓ Un diseñador pide el mismo símbolo en la galería, en una barra de herramientas y en una celda de cuadrícula. ¿Qué cuesta menos cambiar después, la clave de tema, la URL relativa o el recurso del pack, y por qué?

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.

- ✓ **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."*
- ✓ **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."*



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.