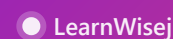


Ejercicio de programación con IA

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 7.

Qué estás pidiendo

Termina el proyecto final de IconDesk. Referencia una hoja de estilos de fuente de iconos desde Default.html, activa AllowHtml en una pequeña barra de herramientas o tarjeta y renderiza sus iconos como glifos de fuente, después construye una columna de DataGridView cuyas celdas contengan acciones con fuente de iconos y lee qué elemento pulsó el usuario para que Edit y Delete hagan cosas distintas. Construye una página de resumen que demuestre a la vez todos los mecanismos vistos en el curso: una imagen de tema, un icono de un pack oficial, un icono de un pack propio, un logotipo incrustado y un SVG recoloreado. Después escribe la política de iconos de producción del proyecto y haz una pasada de QA en un tema claro y en uno oscuro.

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

- ✓ Hoja de estilos de fuente de iconos referenciada desde Default.html y una barra de herramientas o tarjeta con AllowHtml que renderiza glifos de fuente
- ✓ Columna de DataGridView cuyas celdas renderizan acciones con fuente de iconos y responden al elemento pulsado
- ✓ Página de resumen de IconDesk que demuestra una imagen de tema, un icono de un pack oficial, un icono de un pack propio, un recurso incrustado y un SVG recoloreado
- ✓ Política de iconos de producción escrita, que cubra accesibilidad, licencias, caché y dependencias externas
- ✓ Pasada de QA realizada en un tema claro y en uno oscuro con los hallazgos registrados

Por qué tiene esta forma

Después de seis módulos, IconDesk sabe dibujar un icono de cinco formas distintas, y todas funcionan. Lo que al proyecto le sigue faltando es una regla de decisión: cuando una pantalla nueva necesita un símbolo de Save, nadie debería tener que adivinar si viene del tema, de un pack oficial de Wisej-4 o de

IconDesk.Icons. Este módulo añade la última opción, las fuentes de iconos renderizadas como HTML, y escribe después la política que dice a cuál recurrir y qué comprobar antes de publicar una pantalla.

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
fuentes de iconos	Una fuente web cuyos glifos son imágenes en lugar de letras, que se eligen con nombres de clase CSS y dibuja el navegador dentro del contenido HTML, no a través de ninguna propiedad de imagen.
AllowHtml	El interruptor por control que le dice a Wisej.NET que interprete el texto de un control como marcado; sin él las etiquetas aparecen como caracteres literales, porque el texto es texto plano de forma predeterminada.
cabecera de Default.html	La página anfitriona de la aplicación, donde un único enlace a una hoja de estilos pone el CSS y los archivos de fuente del pack de iconos a disposición de todas las pantallas que abra la sesión.
clic de elemento según su rol	Un clic dentro de contenido HTML puede informar de qué elemento en línea se pulsó, así que una misma celda de cuadrícula puede ofrecer un destino de Edit y otro de Delete sin dos controles separados.
CDN frente a alojamiento propio	Apuntar el enlace a una CDN pública es una decisión de producción sobre disponibilidad, privacidad, Content Security Policy y licencias; las aplicaciones internas suelen alojar ellas mismas los archivos.
política de iconos por capas	Una regla escrita que le da a cada tipo de símbolo un hogar: imágenes de tema para las semánticas, un pack principal para los comandos, el pack propio para la ilustración del producto y archivos ráster para las fotos.
pasada de QA de iconos	Un recorrido por las pantallas terminadas en un tema claro y en uno oscuro con más de una escala de pantalla, comprobando que cada icono sigue viéndose y sigue teniendo una alternativa textual.

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 7. Puedes abrir tú mismo [module-07.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 7 – «Fuentes de iconos, contenido HTML y estrategia de iconos en producción» – 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-07.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: Termina el proyecto final de IconDesk. Referencia una hoja de estilos de fuente de iconos desde `Default.html`, activa `AllowHtml` en una pequeña barra de herramientas o tarjeta y renderiza sus iconos como glifos de fuente, después construye una columna de `DataGridView` cuyas celdas contengan acciones con fuente de iconos y lee qué elemento pulsó el usuario para que `Edit` y `Delete` hagan cosas distintas. Construye una página de resumen que demuestre a la vez todos los mecanismos vistos en el curso: una imagen de tema, un icono de un pack oficial, un icono de un pack propio, un logotipo incrustado y un SVG recoloreado. Después escribe la política de iconos de producción del proyecto y haz una pasada de QA en un tema claro y en uno oscuro.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Hoja de estilos de fuente de iconos referenciada desde `Default.html` y una barra de herramientas o tarjeta con `AllowHtml` que renderiza glifos de fuente
 - Columna de `DataGridView` cuyas celdas renderizan acciones con fuente de iconos y responden al elemento pulsado
 - Página de resumen de IconDesk que demuestra una imagen de tema, un icono de un pack oficial, un icono de un pack propio, un recurso incrustado y un SVG recoloreado
 - Política de iconos de producción escrita, que cubra accesibilidad, licencias, caché y dependencias externas
 - Pasada de QA realizada en un tema claro y en uno oscuro con los hallazgos registrados
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:

- El enlace a la hoja de estilos está en `Default.html` y los nombres de clase son correctos, pero una `Label` muestra las etiquetas en bruto mientras que el `Button` de al lado muestra el glifo. ¿Qué diferencia hay entre los dos controles y por qué la configuración es por control y no por aplicación?

- Tu columna de acciones funciona y un compañero pide meter el texto de la nota del cliente en esa misma celda para poder darle estilo. ¿Qué le dices y qué construirías en su lugar?
- En la página de resumen, ¿cuáles de las cinco imágenes cambian cuando la aplicación cambia de tema y cuáles se quedan exactamente como estaban dibujadas? ¿Qué te dice eso sobre dónde se almacena cada una?

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:

- ✓ El enlace a la hoja de estilos está en Default.html y los nombres de clase son correctos, pero una Label muestra las etiquetas en bruto mientras que el Button de al lado muestra el glifo. ¿Qué diferencia hay entre los dos controles y por qué la configuración es por control y no por aplicación?
- ✓ Tu columna de acciones funciona y un compañero pide meter el texto de la nota del cliente en esa misma celda para poder darle estilo. ¿Qué le dices y qué construirías en su lugar?
- ✓ En la página de resumen, ¿cuáles de las cinco imágenes cambian cuando la aplicación cambia de tema y cuáles se quedan exactamente como estaban dibujadas? ¿Qué te dice eso sobre dónde se almacena cada una?

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.

- ✓ **Revísalo 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.