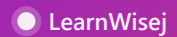


Ejercicio de programación con IA

Construye el ejemplo OperationsConsole 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 — incluso comparándolo con nuestra propia solución de referencia en GitHub.



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 OperationsConsole; 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

Sustituye los marcadores provisionales del Módulo 1 por el shell real de Operations Console: un SplitContainer con la navegación a la izquierda, una superficie de detalle TabControl a la derecha, una ToolBar para los comandos frecuentes y una StatusBar que muestre el estado real. Mueve la interfaz repetida a UserControls que expongan solo propiedades y eventos públicos significativos, implementa un diseño para perfil estrecho con propiedades responsive o lógica según el perfil, y documenta tus decisiones de docking y anclaje en LayoutNotes.md.

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

- ✓ Shell de navegación construido con SplitContainer, TabControl, ToolBar y StatusBar
- ✓ Interfaz repetida movida a UserControls con una superficie pública pequeña
- ✓ Diseño para perfil estrecho con propiedades responsive o lógica según el perfil
- ✓ LayoutNotes.md que documenta las decisiones de docking y anclaje
- ✓ Capturas de pantalla en anchos de escritorio y estrechos

Por qué tiene esta forma

Los contenedores son la arquitectura de la interfaz de usuario: deciden cómo se agrupan, redimensionan, navegan y reutilizan los controles. La Operations Console del Módulo 1 todavía tiene paneles provisionales, y una pantalla en la que cada control se posiciona a mano se vuelve frágil en cuanto se redimensiona el navegador. Los problemas de diseño suelen ser problemas de concepción, no de píxeles, y este módulo sustituye el ajuste píxel a píxel por un plan: regiones acopladas, un SplitContainer, un TabControl, un modelo de comandos real y UserControls que ocultan sus detalles internos.

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
orden de docking	El docking fija un hijo a un lado de su contenedor o le hace rellenar lo que queda, y se aplica en el orden de los hijos, así que los mismos cuatro controles producen shells distintos según el orden en que se añadieron.
anclaje	El anclaje mantiene los bordes de un control ligados a los bordes de su contenedor; anclado a izquierda y derecha se estira con la anchura, y sin anclaje en ningún eje flota manteniendo su posición relativa.
SplitContainer	Dos regiones adyacentes que el usuario puede redimensionar; la elección adecuada siempre que sea la persona, y no el desarrollador, quien decida cuánto espacio recibe el árbol de navegación o el panel de detalle.
paneles de diseño	FlowLayoutPanel coloca los hijos uno junto a otro y los ajusta a la línea siguiente, TableLayoutPanel alinea filas y columnas, y FlexLayoutPanel reparte el espacio y ajusta líneas como un diseño responsive moderno.
propiedades responsive	Una propiedad que guarda un valor distinto por perfil de cliente, fijado en tiempo de diseño, para que un diseño de escritorio pueda cambiar unas pocas propiedades en pantallas estrechas sin una segunda pantalla.
modelo de comandos	Una regla coherente sobre dónde vive cada comando: MenuBar, ToolBar, RibbonBar, ContextMenu o StatusBar, cada uno con su función, y todos invocando comandos de la aplicación en lugar de contener lógica de negocio.
superficie de un UserControl	La unidad de reutilización: un control compuesto que expone unas pocas propiedades, eventos y métodos significativos en lugar de cada control hijo, para que quienes lo usan nunca se acoplen a su implementación.

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 OperationsConsole 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 – «Contenedores, diseños, navegación y reutilización» – del curso de LearnWisej «Dominar la biblioteca de controles», 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/mastering-the-control-library/specs/module-03.md>

Después construye un proyecto Wisej.NET 4 completo que pueda abrir en Visual Studio:

1. Proyecto «OperationsConsole», 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: Sustituye los marcadores provisionales del Módulo 1 por el shell real de Operations Console: un SplitContainer con la navegación a la izquierda, una superficie de detalle TabControl a la derecha, una ToolBar para los comandos frecuentes y una StatusBar que muestre el estado real. Mueve la interfaz repetida a UserControls que expongan solo propiedades y eventos públicos significativos, implementa un diseño para perfil estrecho con propiedades responsive o lógica según el perfil, y documenta tus decisiones de docking y anclaje en LayoutNotes.md.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Shell de navegación construido con SplitContainer, TabControl, ToolBar y StatusBar
 - Interfaz repetida movida a UserControls con una superficie pública pequeña
 - Diseño para perfil estrecho con propiedades responsive o lógica según el perfil
 - LayoutNotes.md que documenta las decisiones de docking y anclaje
 - Capturas de pantalla en anchos de escritorio y estrechos
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:

- Si el SplitContainer se hubiera añadido a la página antes que la ToolBar y la StatusBar, ¿qué vería el usuario y por qué importa el orden?
- ¿Qué miembros de tu UserControl podrías renombrar sin tocar ninguna página que lo use, y cuáles no? ¿Qué te dice eso sobre su superficie pública?
- ¿Qué propiedades cambiaron en el perfil estrecho y por qué no construiste una pantalla estrecha aparte?

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:

- ✓ Si el SplitContainer se hubiera añadido a la página antes que la ToolBar y la StatusBar, ¿qué vería el usuario y por qué importa el orden?
- ✓ ¿Qué miembros de tu UserControl podrías renombrar sin tocar ninguna página que lo use, y cuáles no? ¿Qué te dice eso sobre su superficie pública?
- ✓ ¿Qué propiedades cambiaron en el perfil estrecho y por qué no construiste una pantalla estrecha aparte?

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.



Este ya lo construimos — compara tu versión con la nuestra

La implementación de referencia del Módulo 3 está en el repositorio público de ejemplos. Construye primero la tuya y después abre la nuestra: lo interesante son las diferencias.



[Abrir el ejemplo del Módulo 3 en GitHub](#)

O clona el conjunto completo con `git clone https://github.com/iceteagroup/LearnWisej-Samples` y abre `Mastering the Control Library Course/Module 3`.



Nuestro ejemplo es un punto de partida, no un producto terminado

Se construyó igual que estás construyendo el tuyo — con un prompt y luego corregido — y se detiene a propósito donde se detiene el módulo. Espera soluciones provisionales donde iría código de producción, un manejo de errores mínimo y ninguna prueba. Tómallo como una invitación: usa lo que aprendiste en este módulo, junto con el asistente, para llevarlo más lejos. Mejorar código que no escribiste tú es precisamente la habilidad que enseña este curso.

Tres prompts para ir más allá

Dirige tu asistente al proyecto que acabas de construir — o al nuestro 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/mastering-the-control-library/specs/module-03.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."*
- ✓ **Revísalo como un pull request.** *"Explícame este código de OperationsConsole 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 OperationsConsole 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.