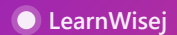


Ejercicio de programación con IA

Construye el ejemplo AdaptiveOps 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.



El mismo módulo, construido como de verdad vas a construir.

Todas las demás secciones de este módulo explican cómo funciona «FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel y propiedades de diseño extendidas». Esta te pide que lo generes — dirigiendo a un asistente con la propia especificación del módulo y comprobando después que el resultado se ejecuta y hace lo que describe el enunciado del laboratorio. Es el flujo de trabajo que usarás en proyectos reales, practicado con algo cuya respuesta correcta ya conoces.

Qué estás pidiendo

Construye tres versiones del área de contenido de Adaptive Operations Console: una zona de filtros y tarjetas de métricas con FlowLayoutPanel que se ajusta a la línea y usa FillWeight y FlowBreak, un editor de tickets con TableLayoutPanel con etiquetas y editores alineados mediante RowStyles y ColumnStyles, y un shell de panel de control con FlexLayoutPanel donde las regiones de lista y detalles reparten el espacio por FillWeight con AlignX, AlignY, MinimumSize y MaximumSize definidos. Construye una de las tres por código con las extensiones fluent de Wisej.Web.Markup y compara cómo resiste cada una el redimensionado del navegador.

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

- ✓ Zona de filtros y tarjetas de métricas con FlowLayoutPanel usando WrapContents, FillWeight y FlowBreak
- ✓ Editor de tickets con TableLayoutPanel con RowStyles, ColumnStyles y spans para campos alineados
- ✓ Shell de panel de control con FlexLayoutPanel con FillWeight, AlignX, AlignY, MinimumSize y MaximumSize
- ✓ Una región construida por código con las extensiones fluent de Wisej.Web.Markup
- ✓ Capturas comparativas del redimensionado y una nota sobre qué contenedor se adapta a cada región

Por qué tiene esta forma

El Módulo 4 dio a Adaptive Operations Console un shell de regiones acopladas, pero el acoplamiento solo decide dónde se sitúa una región, no cómo se comporta el contenido que hay dentro cuando el navegador mide 1400 píxeles de ancho el lunes y 700 en una tableta el martes. Una fila de tarjetas de

métricas debe pasar a la línea siguiente, un editor de tickets debe mantener alineados sus etiquetas y editores por larga que sea la traducción al alemán, y las regiones de lista y detalles deben repartirse el área de trabajo en la proporción que tú decidas. Cada uno es un comportamiento de contenido distinto, y Wisej.NET incluye un contenedor para cada uno: FlowLayoutPanel, TableLayoutPanel y FlexLayoutPanel. Elegir el equivocado es la forma en que los equipos acaban volviendo a escribir código de redimensionado.

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
FlowLayoutPanel	Coloca los hijos en filas o columnas automáticas a lo largo de FlowDirection, pasa a la línea siguiente cuando WrapContents es true e ignora Location, Dock y Anchor de cada hijo, así que los chips y las tarjetas se recolocan solos.
TableLayoutPanel	Coloca los hijos en celdas identificadas por Row y Column, con spans para las celdas que se extienden, de modo que etiquetas y editores permanecen alineados mientras cada hijo puede seguir acoplándose o anclándose dentro de su celda.
FlexLayoutPanel	Coloca los hijos en una sola fila o una sola columna, siempre ocupa su área cliente y reparte el espacio restante por peso, lo que encaja con las regiones principales que dividen el área de trabajo de forma proporcional.
FillWeight	Una propiedad extendida del hijo que leen los motores flow y flex: un peso mayor que cero estira el hijo en el espacio restante en proporción a los demás hijos con peso; cero mantiene su propio tamaño.
FlowBreak	Una propiedad extendida de los hijos de FlowLayoutPanel que fuerza al siguiente hijo a empezar una fila o columna nueva, de modo que una barra de filtros termina una línea de forma deliberada en lugar de depender del ancho.
Estilos de fila y de columna	RowStyles y ColumnStyles declaran si una fila o columna ajusta su tamaño automáticamente, toma un tamaño fijo en píxeles o reparte el resto por porcentaje; ahí residen las proporciones de un formulario.
Margin y Padding	Padding es el espacio interior del contenedor alrededor de todos los hijos; Margin es la distancia del hijo a sus vecinos, y solo la usan los motores que respetan los márgenes.

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 5. Puedes abrir tú mismo [module-05.md](#) para ver exactamente lo que leerá.



Usar ChatGPT



Usar Claude

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.

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 AdaptiveOps 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 5 – «FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel y propiedades de diseño extendidas» – del curso de LearnWisej «Temas y diseños adaptables», 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/theming-and-responsive-layouts/specs/module-05.md>

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

1. Proyecto «AdaptiveOps», 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 tres versiones del área de contenido de Adaptive Operations Console: una zona de filtros y tarjetas de métricas con FlowLayoutPanel que se ajusta a la línea y usa FillWeight y FlowBreak, un editor de tickets con TableLayoutPanel con etiquetas y editores alineados mediante RowStyles y ColumnStyles, y un shell de panel de control con FlexLayoutPanel donde las regiones de lista y detalles reparten el espacio por FillWeight con AlignX, AlignY, MinimumSize y MaximumSize definidos. Construye una de las tres por código con las extensiones fluent de Wisej.Web.Markup y compara cómo resiste cada una el redimensionado del navegador.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Zona de filtros y tarjetas de métricas con FlowLayoutPanel usando WrapContents, FillWeight y FlowBreak
 - Editor de tickets con TableLayoutPanel con RowStyles, ColumnStyles y spans para campos alineados
 - Shell de panel de control con FlexLayoutPanel con FillWeight, AlignX, AlignY, MinimumSize y MaximumSize
 - Una región construida por código con las extensiones fluent de Wisej.Web.Markup
 - Capturas comparativas del redimensionado y una nota sobre qué contenedor se adapta a cada región
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:

- Diste a la región de detalles FillWeight 1 pero ningún MinimumSize, y a 700 píxeles desapareció. Explica qué hizo el motor flex con el espacio y por qué la solución es una restricción de tamaño y no un peso distinto.
- El botón Apply pasa a la segunda fila a 1024 píxeles aunque hay sitio para él. ¿Qué propiedad extendida lo provocó, y cómo se comportaría FilterBar de otra forma si la quitaras

y pusieras `WrapContents` a `false`?

– Un `Margin` de 8 en las tarjetas del `FlowLayoutPanel` produjo espacios, pero el mismo `Margin` en el carril de navegación acoplado no hizo nada. ¿Qué motor respeta los márgenes en cada caso, y qué propiedad crea el espacio para el carril?

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:

- ✓ Dististe a la región de detalles `FillWeight` 1 pero ningún `MinimumSize`, y a 700 píxeles desapareció. Explica qué hizo el motor flex con el espacio y por qué la solución es una restricción de tamaño y no un peso distinto.
- ✓ El botón `Apply` pasa a la segunda fila a 1024 píxeles aunque hay sitio para él. ¿Qué propiedad extendida lo provocó, y cómo se comportaría `FilterBar` de otra forma si la quitaras y pusieras `WrapContents` a `false`?
- ✓ Un `Margin` de 8 en las tarjetas del `FlowLayoutPanel` produjo espacios, pero el mismo `Margin` en el carril de navegación acoplado no hizo nada. ¿Qué motor respeta los márgenes en cada caso, y qué propiedad crea el espacio para el carril?

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 5 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 5 en GitHub](#)

O clona el conjunto completo con `git clone https://github.com/iceteagroup/LearnWisej-Samples` y abre `Theming and Responsive Layouts Course/Module 5`.



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.

- ✓ **Haz que aguante un mal día.** *"Revisa este proyecto AdaptiveOps 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 AdaptiveOps 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."*
- ✓ **Demuestra que de verdad funciona.** *"Recorre este proyecto entregable por entregable frente a la especificación del módulo en <https://learnwisej.com/assets/courses/theming-and-responsive-layouts/specs/module-05.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."*



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.