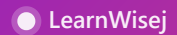


Ejercicio de programación con IA

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.



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

Todas las demás secciones de este módulo explican cómo funciona «Dominio del DataGridView». 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 la cuadrícula Orders de Operations Console: un DataGridView enlazado a un BindingSource con columnas explícitas y AutoGenerateColumns bajo control, una celda de estado renderizada con AllowHtml y CellFormatting, un editor personalizado o una columna de comando, una versión en modo virtual o respaldada por caché para la ruta de grandes volúmenes de datos, y una barra compuesta de filtro y estado acoplada dentro o alrededor de la cuadrícula.

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

- ✓ DataGridView Orders con columnas explícitas enlazado a un BindingSource
- ✓ Celda de estado renderizada con AllowHtml y CellFormatting
- ✓ Un editor personalizado o una columna de comando
- ✓ Versión de la cuadrícula en modo virtual o respaldada por caché para grandes volúmenes de datos
- ✓ Barra compuesta de filtro y estado acoplada dentro o alrededor de la cuadrícula

Por qué tiene esta forma

DataGridView es el control que con más probabilidad decide si una aplicación de negocio parece madura. Pedidos, facturas, plazos, logs, usuarios y documentos acaban todos en una cuadrícula, y los usuarios juzgan la aplicación por cómo esa cuadrícula carga, se desplaza, edita e informa del estado. Wisej.NET ofrece de todo, desde el rellenado manual hasta el modo virtual y el dibujo de celdas; dominarlo consiste en elegir la técnica adecuada para el volumen de datos y la tarea del usuario.

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
BindingSource	Un intermediario en el servidor entre las filas del view model y la cuadrícula, que le da un origen de datos estable, posición actual y ordenación sin acoplar la pantalla a la base de datos.
AutoGenerateColumns	Crea una columna por cada propiedad pública del tipo enlazado; útil para un prototipo, peligroso en producción porque un cambio en el modelo reorganiza la pantalla sin avisar.
columna explícita	Una columna que defines con enlace de campo, texto de cabecera, anchura, alineación, formato, estado de solo lectura, modo de ordenación y tipo de editor, para que la interfaz siga estable cuando cambie el modelo.
modo virtual	Con VirtualMode activado, la cuadrícula solo conoce su número de filas y pide cada valor visible mediante CellValueNeeded, así que no se crea nada para las filas que nadie ve.
caché en el servidor	Un búfer del tamaño de una página que alimenta CellValueNeeded para que el desplazamiento no consulte la base de datos por cada celda; el evento DataRead es el punto de enganche documentado para construirla y mantenerla.
editor de celda personalizado	Un control asignado a la propiedad Editor de una columna; debe devolver su valor a la celda, mediante CellBeginEdit y CellEndEdit si no tiene un evento de cambio adecuado.
HTML en celda frente a control en celda	El HTML producido en CellFormatting con AllowHtml es markup solo de visualización y casi gratuito; un control incrustado es un control de servidor y un widget del navegador reales, con un coste por celda.

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

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 5 - «Dominio del DataGridView» - 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-05.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: Construye la cuadrícula Orders de Operations Console: un DataGridView enlazado a un BindingSource con columnas explícitas y `AutoGenerateColumns` bajo control, una celda de estado renderizada con `AllowHtml` y `CellFormatting`, un editor personalizado o una columna de comando, una versión en modo virtual o respaldada por caché para la ruta de grandes volúmenes de datos, y una barra compuesta de filtro y estado acoplada dentro o alrededor de la cuadrícula.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - DataGridView Orders con columnas explícitas enlazado a un BindingSource
 - Celda de estado renderizada con `AllowHtml` y `CellFormatting`
 - Un editor personalizado o una columna de comando
 - Versión de la cuadrícula en modo virtual o respaldada por caché para grandes volúmenes de datos
 - Barra compuesta de filtro y estado acoplada dentro o alrededor de la cuadrícula
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 `OrderRow` gana dos propiedades nuevas en el próximo sprint, ¿qué cambia en la pantalla con columnas generadas y qué cambia con tus columnas explícitas?
- Tu manejador de `CellValueNeeded` funciona con 200 pedidos. ¿Qué le ocurre a la base de datos cuando un usuario arrastra la barra de desplazamiento a través de 200.000, y qué parte de tu diseño lo absorbe?
- ¿Por qué la insignia de estado corresponde a `CellFormatting` y no al modelo `OrderRow`, y qué saldría mal si se colocara el nombre de un cliente en ese HTML sin codificar?

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 OrderRow gana dos propiedades nuevas en el próximo sprint, ¿qué cambia en la pantalla con columnas generadas y qué cambia con tus columnas explícitas?
- ✓ Tu manejador de CellValueNeeded funciona con 200 pedidos. ¿Qué le ocurre a la base de datos cuando un usuario arrastra la barra de desplazamiento a través de 200.000, y qué parte de tu diseño lo absorbe?
- ✓ ¿Por qué la insignia de estado corresponde a CellFormatting y no al modelo OrderRow, y qué saldría mal si se colocara el nombre de un cliente en ese HTML sin codificar?

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 `Mastering the Control Library 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.

- ✓ **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."*
- ✓ **Haz que aguante un mal día.** *"Revisa este proyecto OperationsConsole 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 OperationsConsole 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.