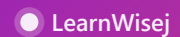


Ejercicio de programación con IA

Construye el ejemplo VisualOperationsStudio 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.



No vas a teclear este ejemplo. Vas a dirigirlo.

Ya casi nadie escribe código Wisej.NET partiendo de un archivo vacío — un asistente escribe el primer borrador y el desarrollador decide si es correcto. Eso solo funciona si dominas el tema, y eso es lo que te dio el resto de este módulo. Aquí le pasas ese conocimiento a ChatGPT o a Claude, recibes el ejemplo de este módulo como una solución de Visual Studio que funciona y lo revisas como revisarías el pull request de un compañero.

Qué estás pidiendo

Construye el shell de VisualOperationsStudio y representa una única lectura de telemetría en cuatro superficies a la vez. Crea una VisualOperationsPage que contenga una sola instancia del modelo TelemetrySample y después presenta su valor como (1) un Label simple más un ProgressBar, (2) un Panel con un controlador de Paint que dibuja una barra a partir de PaintEventArgs.Graphics y e.ClipRectangle, (3) un Wisej.Web.Canvas cuyo controlador de Redraw reconstruye la misma barra a partir del modelo y (4) un Bitmap fuera de pantalla renderizado con Graphics.FromImage y mostrado en un PictureBox. Conecta un botón que cambie el valor del modelo y actualice las cuatro superficies mediante Invalidate() y un redibujado del Canvas; después redimensiona el navegador y anota qué superficies sobreviven y cuáles hay que reconstruir.

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

- ✓ VisualOperationsPage con un único modelo TelemetrySample del que leen todas las superficies
- ✓ Controlador de Paint que dibuja desde PaintEventArgs.Graphics dentro de e.ClipRectangle
- ✓ Controlador de Redraw del Canvas que reconstruye toda la escena a partir del modelo después de un redimensionado
- ✓ Bitmap fuera de pantalla renderizado con Graphics.FromImage y mostrado en la página
- ✓ SurfaceNotes.md que registre qué cruza la red en cada superficie y qué elementos visuales sobreviven a un redimensionado

Por qué tiene esta forma

VisualOperationsStudio tiene que mostrar la misma lectura de máquina en cuatro sitios: un dial en la página de operaciones, una barra de salud dentro de una fila de cuadrícula, un nodo en un mapa de topología y un PNG adjunto al informe nocturno de turno. Todos son «dibujar un número» y, sin embargo,

echar mano de la API de dibujo que recuerdes del escritorio produce cuatro fragmentos de código con comportamientos radicalmente distintos en CPU, ancho de banda y redimensionado. Este módulo pone la elección de la superficie por delante del código de dibujo, porque la superficie decide dónde se producen los píxeles y qué viaja hasta el navegador.

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
Paint / PaintEventArgs	El evento de dibujo a nivel de control: tu C# se ejecuta en el servidor sobre el Graphics que proporciona, dentro del ClipRectangle que indica, y Wisej.NET convierte el resultado en una imagen para el navegador.
DataGridView.CellPaint	El mismo mecanismo de dibujo acotado a una celda; su DataGridViewCellPaintEventArgs deriva de PaintEventArgs y añade RowIndex y ColumnIndex.
UserPaint	El interruptor que necesita una celda o una columna entera para que la cuadrícula dispare CellPaint, lo que mantiene el dibujo como algo opcional por columna en lugar de extenderlo a toda la cuadrícula.
Wisej.Web.Canvas	Una superficie 2D de modo inmediato en el navegador que refleja el modelo CanvasRenderingContext2D de HTML5; las llamadas de dibujo las ejecuta el cliente, no se renderizan como imagen en el servidor.
Redraw	El evento del Canvas en el que se reconstruye toda la escena a partir de tu modelo, porque el canvas guarda píxeles y los pierde cuando se recrea o se redimensiona.
System.Drawing.Managed	Renderizado fuera de pantalla sin ningún control visible: crea un Bitmap, obtén su Graphics con Graphics.FromImage() y guarda los bytes de la imagen para un informe, una exportación o una descarga.
Invalidate()	La llamada que indica a un control dibujado que su representación ha quedado obsoleta; actualiza el widget del cliente y la superficie del diseñador de Visual Studio, así que corresponde al punto donde cambió el estado, no al interior de los ayudantes de dibujo.

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 1. Puedes abrir tú mismo [module-01.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 VisualOperationsStudio 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 1 – «La arquitectura de dibujo de Wisej.NET» – del curso de LearnWisej «Dibujo y pintura 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/drawing-and-painting/specs/module-01.md>

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

1. Proyecto «VisualOperationsStudio», 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 el shell de VisualOperationsStudio y representa una única lectura de telemetría en cuatro superficies a la vez. Crea una VisualOperationsPage que contenga una sola instancia del modelo TelemetrySample y después presenta su valor como (1) un Label simple más un ProgressBar, (2) un Panel con un controlador de Paint que dibuja una barra a partir de PaintEventArgs.Graphics y e.ClipRectangle, (3) un Wisej.Web.Canvas cuyo controlador de Redraw reconstruye la misma barra a partir del modelo y (4) un Bitmap fuera de pantalla renderizado con Graphics.FromImage y mostrado en un PictureBox. Conecta un botón que cambie el valor del modelo y actualice las cuatro superficies mediante Invalidate() y un redibujado del Canvas; después redimensiona el navegador y anota qué superficies sobreviven y cuáles hay que reconstruir.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - VisualOperationsPage con un único modelo TelemetrySample del que leen todas las superficies
 - Controlador de Paint que dibuja desde PaintEventArgs.Graphics dentro de e.ClipRectangle
 - Controlador de Redraw del Canvas que reconstruye toda la escena a partir del modelo después de un redimensionado
 - Bitmap fuera de pantalla renderizado con Graphics.FromImage y mostrado en la página
 - SurfaceNotes.md que registre qué cruza la red en cada superficie y qué elementos visuales sobreviven a un redimensionado
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:

- Estableces la lectura del modelo desde el botón, pero olvidas la llamada a Invalidate() en el panel dibujado. La etiqueta y el canvas se actualizan. Explica exactamente por qué el panel no lo hace y qué está mirando el usuario hasta que algo más fuerce el redibujado.

- Tu primera versión del Canvas dibuja la barra una sola vez al cargar la página, en lugar de hacerlo desde Redraw. Describe qué ve el usuario después de redimensionar el navegador y por qué el Bitmap fuera de pantalla de la cuarta área se comporta de otra forma.
- Quien revisa tu trabajo pregunta por qué está en el laboratorio la primera área, el Label y el ProgressBar simples. Responde con los costes que pagan las otras tres superficies y ella no, y nombra un requisito que justificaría pagarlos.

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:

- ✓ Estableces la lectura del modelo desde el botón, pero olvidas la llamada a Invalidate() en el panel dibujado. La etiqueta y el canvas se actualizan. Explica exactamente por qué el panel no lo hace y qué está mirando el usuario hasta que algo más fuerce el redibujado.
- ✓ Tu primera versión del Canvas dibuja la barra una sola vez al cargar la página, en lugar de hacerlo desde Redraw. Describe qué ve el usuario después de redimensionar el navegador y por qué el Bitmap fuera de pantalla de la cuarta área se comporta de otra forma.
- ✓ Quien revisa tu trabajo pregunta por qué está en el laboratorio la primera área, el Label y el ProgressBar simples. Responde con los costes que pagan las otras tres superficies y ella no, y nombra un requisito que justificaría pagarlos.

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.

- ✓ **Añade el segundo caso.** *"Amplía este proyecto VisualOperationsStudio 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/drawing-and-painting/specs/module-01.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."
- ✓ **Revisalo como un pull request.** "Explícame este código de VisualOperationsStudio 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."



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.