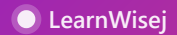


Ejercicio de programación con IA

Construye el ejemplo GlobalDesk 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 2.

Qué estás pidiendo

Localiza el editor de clientes de GlobalDesk desde el diseñador de Visual Studio. Termina primero el diseño neutro en inglés del control de usuario CustomerEditor, después pon Localizable en true y usa el selector Language del diseñador para añadir una variante en alemán, cambiando solo lo que debe diferir: las etiquetas, el ancho de los dos botones más largos y la Location de una etiqueta. Ejecuta la aplicación con de-DE y compara la salida real del navegador con la vista del diseñador; después añade un botón Recreate editor que haga Dispose del control y lo reconstruya, para que compruebes por ti mismo por qué los recursos del diseñador no siguen un cambio de cultura en tiempo de ejecución. Termina con una nota de laboratorio que enumere las propiedades que has sobrescrito y las que has dejado deliberadamente al diseño neutro.

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

- ✓ Control de usuario CustomerEditor con Localizable en true sobre un diseño neutro en inglés completo
- ✓ Variante de idioma alemán en el diseñador que sobrescribe las etiquetas y al menos un Size y una Location
- ✓ Anchoring, docking o AutoSize aplicados para que las etiquetas en alemán no puedan recortarse
- ✓ Botón Recreate editor que hace Dispose del contenedor y lo reconstruye tras un cambio de cultura
- ✓ Nota de laboratorio con las propiedades sobrescritas y las que se han dejado al diseño neutro

Por qué tiene esta forma

El Módulo 1 le dio a GlobalDesk un dashboard cuyas etiquetas proceden todas de Strings.resx, y ese es el sitio adecuado para el texto que genera el código. El editor de clientes es distinto: es una pantalla diseñada, colocada a ojo en el diseñador de Visual Studio. Hacer pasar cada una de sus etiquetas por una llamada al helper obliga a editar a mano el archivo del diseñador, y sigue sin resolver el problema real: un

Kundennummer speichern alemán no cabe en un botón dimensionado para Save. Este módulo usa el flujo de localización que Visual Studio ya ofrece, de modo que las etiquetas y la geometría de una pantalla diseñada puedan diferir por idioma sin cadenas escritas en el código y sin un segundo archivo de diseño. 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
Localizable	Una propiedad de tiempo de diseño en un contenedor de nivel superior; true hace que el diseñador entre en el flujo de localización para que los valores se puedan guardar por cultura.
Selector Language del diseñador	Aparece en cuanto Localizable está en true; la cultura seleccionada es la versión que estás editando, así que cada cambio se registra solo contra esa cultura.
El idioma neutro	La versión predeterminada a la que se repliegan todas las culturas, y la única en la que añades, quitas o reestructuras controles; las vistas de cultura guardan sobrescrituras, no una segunda pantalla.
Propiedades localizables	No solo Text: Size, Location, los colores y las imágenes pueden llevar un valor distinto por cultura, porque la traducción cambia la geometría además de las palabras.
Archivos de recursos generados	El diseñador escribe los valores localizados en archivos de recursos en lugar de en el código C# del diseñador, lo que mantiene los valores específicos de cada idioma fuera del código que lees y fusionas a diario.
Aplicados al crearse	Los valores localizados en el diseñador se aplican mientras se crean los controles, así que cambiar la cultura de la sesión más tarde no reescribe un contenedor que ya existe.
Diseño tolerante a las traducciones	El docking, el anchoring, FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel, AutoSize y el padding dejan que el texto crezca por sí solo, así que necesitas menos sobrescrituras de geometría.

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 2. Puedes abrir tú mismo [module-02.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 GlobalDesk 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 2 – «Localizar la interfaz de Wisej.NET en el diseñador de Visual Studio» – del curso de LearnWisej «Localización 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/localization-in-wisej-net/specs/module-02.md>

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

1. Proyecto «GlobalDesk», 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: Localiza el editor de clientes de GlobalDesk desde el diseñador de Visual Studio. Termina primero el diseño neutro en inglés del control de usuario `CustomerEditor`, después pon `Localizable` en `true` y usa el selector `Language` del diseñador para añadir una variante en alemán, cambiando solo lo que debe diferir: las etiquetas, el ancho de los dos botones más largos y la `Location` de una etiqueta. Ejecuta la aplicación con `de-DE` y compara la salida real del navegador con la vista del diseñador; después añade un botón `Recreate editor` que haga `Dispose` del control y lo reconstruya, para que compruebes por ti mismo por qué los recursos del diseñador no siguen un cambio de cultura en tiempo de ejecución. Termina con una nota de laboratorio que enumere las propiedades que has sobrescrito y las que has dejado deliberadamente al diseño neutro.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Control de usuario `CustomerEditor` con `Localizable` en `true` sobre un diseño neutro en inglés completo
 - Variante de idioma alemán en el diseñador que sobrescribe las etiquetas y al menos un `Size` y una `Location`
 - `Anchoring`, `docking` o `AutoSize` aplicados para que las etiquetas en alemán no puedan recortarse
 - Botón `Recreate editor` que hace `Dispose` del contenedor y lo reconstruye tras un cambio de cultura
 - Nota de laboratorio con las propiedades sobrescritas y las que se han dejado al diseño neutro
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:

- Un compañero añade un campo de teléfono nuevo a `CustomerEditor` con el alemán seleccionado

en la lista Language. ¿Qué es probable que vea el usuario en inglés y cuál es la regla que lo habría evitado?

- La confirmación de guardado que muestra el editor tras una actualización correcta sigue viniendo de Strings.resx y no de los recursos del diseñador. Da dos razones por las que ese es el sitio adecuado y nombra una etiqueta de esta pantalla que sí pertenezca al diseñador.
- Tu botón Recreate editor funciona, pero un compañero sostiene que la pantalla debería simplemente volver a aplicar las propiedades localizadas en lugar de reconstruirse. ¿Qué tendría que saber ese código que la reconstrucción no necesita saber, y cuándo sería su enfoque la mejor opción?

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:

- ✓ Un compañero añade un campo de teléfono nuevo a CustomerEditor con el alemán seleccionado en la lista Language. ¿Qué es probable que vea el usuario en inglés y cuál es la regla que lo habría evitado?
- ✓ La confirmación de guardado que muestra el editor tras una actualización correcta sigue viniendo de Strings.resx y no de los recursos del diseñador. Da dos razones por las que ese es el sitio adecuado y nombra una etiqueta de esta pantalla que sí pertenezca al diseñador.
- ✓ Tu botón Recreate editor funciona, pero un compañero sostiene que la pantalla debería simplemente volver a aplicar las propiedades localizadas en lugar de reconstruirse. ¿Qué tendría que saber ese código que la reconstrucción no necesita saber, y cuándo sería su enfoque la mejor opción?

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.

- ✓ **Demuestra que de verdad funciona.** "Recorre este proyecto entregable por entregable frente a la especificación del módulo en <https://learnwisej.com/assets/courses/localization-in-wisej-net/specs/module-02.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.** "Explicame este código de GlobalDesk 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 GlobalDesk 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.