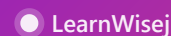


Ejercicio de programación con IA

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 mismo módulo, construido como de verdad vas a construir.

Todas las demás secciones de este módulo explican cómo funciona «Idiomas de derecha a izquierda y diseños a prueba de localización». 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

Haz que GlobalDesk sobreviva a una cultura de derecha a izquierda. Añade el árabe al selector de idioma y pon `rightToLeft` en auto en `Default.json`, deja los controles hijos en `RightToLeft.Inherit` y activa `RightToLeftLayout` en el dashboard y en el editor de clientes para que sus hijos se reflejen. Mantén exactamente un control de izquierda a derecha, el campo del código de cliente, y explica en un comentario por qué ese identificador no debe invertirse. Reconstruye con docking, anchoring y un contenedor de diseño la parte del diseño con posiciones fijas para que una traducción larga no pueda recortarse, añade un archivo de recursos seudolocalizado cuyos valores estén alargados y entre corchetes, y anota lo que hace realmente la cuadrícula `DataGridView` de contactos bajo RTL en lugar de dar por supuesto que se refleja. Termina con una lista de cinco casos de prueba de diseño que puedas repetir en cualquier pantalla.

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

- ✓ Entrada de árabe en el selector de idioma con `rightToLeft` en auto en `Default.json`
- ✓ Dashboard y editor de clientes usando `RightToLeftLayout` con sus hijos en `Inherit`
- ✓ Un control deliberadamente de izquierda a derecha con un comentario que explique la excepción
- ✓ Diseño reconstruido con docking, anchoring y un contenedor de diseño para que las traducciones largas no puedan recortarse
- ✓ Lista de pruebas RTL que cubra texto corto, texto largo, una ventana estrecha, una cultura RTL y la limitación del `DataGridView`

Por qué tiene esta forma

GlobalDesk ya puede cambiar de idioma mientras se ejecuta: el selector establece `Application.CurrentCulture`, el manejador de `Application.CultureChanged` vuelve a aplicar todas las etiquetas respaldadas por recursos y el panel de vista previa reformatea sus ejemplos. Entonces un tester

abre el dashboard en árabe. Todas las palabras están traducidas y todas están en el sitio equivocado: las etiquetas se pegan al borde izquierdo de sus campos, la columna de navegación queda donde la mirada de un lector árabe llega en último lugar, y las etiquetas alemanas largas del módulo anterior ya empujan un botón fuera de un panel de ancho fijo.

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
RightToLeft	Una propiedad de control con los valores Inherit, No y Yes que indica a un control que adapte su propio texto y la dirección de sus elementos internos.
RightToLeft.Inherit	La opción normal para un control hijo: sigue a su padre, así que toda la pantalla cambia de dirección a partir de una única decisión en lo alto.
RightToLeftLayout	Una propiedad distinta que refleja la posición horizontal de los hijos directos de un contenedor cuando RTL está activo, de modo que la disposición visual se invierte mientras tu diseño lógico se mantiene tal y como lo hiciste.
El ajuste de aplicación rightToLeft	Una entrada de Default.json; con el valor "auto" hace que Wisej.NET derive el comportamiento RTL de la cultura de la sesión en lugar de comprobaciones de nombres de cultura repartidas por tus pantallas.
Application.RightToLeft	La misma decisión expresada desde el código, para las sesiones que hay que cambiar deliberadamente y no por detección de cultura.
Expansión del texto	Una etiqueta traducida rara vez mide lo mismo que la inglesa, y por eso las posiciones fijas en píxeles y los anchos de panel medidos a mano son lo primero que se rompe en una compilación localizada.
Límites RTL documentados	Wisej.NET documenta que DataGridView, ListView y PropertyGrid no admiten del todo RTL, así que un plan de pruebas tiene que nombrarlos en lugar de dar por hecho que se reflejan.

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



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 5 – «Idiomas de derecha a izquierda y diseños a prueba de localización» – 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-05.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: Haz que GlobalDesk sobreviva a una cultura de derecha a izquierda. Añade el árabe al selector de idioma y pon `rightToLeft` en auto en `Default.json`, deja los controles hijos en `RightToLeft.Inherit` y activa `RightToLeftLayout` en el dashboard y en el editor de clientes para que sus hijos se reflejen. Mantén exactamente un control de izquierda a derecha, el campo del código de cliente, y explica en un comentario por qué ese identificador no debe invertirse. Reconstruye con docking, anchoring y un contenedor de diseño la parte del diseño con posiciones fijas para que una traducción larga no pueda recortarse, añade un archivo de recursos seudolocalizado cuyos valores estén alargados y entre corchetes, y anota lo que hace realmente la cuadrícula `DataGridView` de contactos bajo RTL en lugar de dar por supuesto que se refleja. Termina con una lista de cinco casos de prueba de diseño que puedas repetir en cualquier pantalla.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Entrada de árabe en el selector de idioma con `rightToLeft` en auto en `Default.json`
 - Dashboard y editor de clientes usando `RightToLeftLayout` con sus hijos en `Inherit`
 - Un control deliberadamente de izquierda a derecha con un comentario que explique la excepción
 - Diseño reconstruido con docking, anchoring y un contenedor de diseño para que las traducciones largas no puedan recortarse
 - Lista de pruebas RTL que cubra texto corto, texto largo, una ventana estrecha, una cultura RTL y la limitación del `DataGridView`
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 pone `RightToLeft.Yes` en veinte controles y la pantalla en árabe se ve correcta, pero la pantalla en alemán no ha cambiado. ¿Qué ha configurado en realidad y qué única propiedad habría hecho el trabajo para todo el contenedor?

- Tu pasada seudolocalizada muestra tres etiquetas todavía en inglés a secas. Nada se recortó y nada lanzó una excepción. ¿Qué te dice eso sobre esas tres etiquetas y en qué módulo anterior quedó un hueco de trabajo?
- La cuadrícula de contactos no se refleja igual que el resto del dashboard. De tus opciones (aceptarlo y documentarlo, sustituir el control o escribir el reflejo a mano), ¿cuál puedes defender ante un revisor y qué tendrías que verificar antes de prometer cualquiera de ellas?

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 pone RightToLeft.Yes en veinte controles y la pantalla en árabe se ve correcta, pero la pantalla en alemán no ha cambiado. ¿Qué ha configurado en realidad y qué única propiedad habría hecho el trabajo para todo el contenedor?
- ✓ Tu pasada seudolocalizada muestra tres etiquetas todavía en inglés a secas. Nada se recortó y nada lanzó una excepción. ¿Qué te dice eso sobre esas tres etiquetas y en qué módulo anterior quedó un hueco de trabajo?
- ✓ La cuadrícula de contactos no se refleja igual que el resto del dashboard. De tus opciones (aceptarlo y documentarlo, sustituir el control o escribir el reflejo a mano), ¿cuál puedes defender ante un revisor y qué tendrías que verificar antes de prometer cualquiera de ellas?

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.

- ✓ **Haz que aguante un mal día.** *"Revisa este proyecto GlobalDesk 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 GlobalDesk 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/localization-in-wisej-net/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.