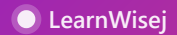


Ejercicio de programación con IA

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.



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

Crea la página del dashboard de GlobalDesk con un encabezado de bienvenida, un botón de navegación Customers y un pequeño panel de vista previa de la cultura. Añade Resources/Strings.resx como recurso neutro con al menos doce claves semánticas estables como Navigation.Customers y CustomerEditor.Save, escribe un helper estático Texts sobre Wisej.Resources.ResourceManager que devuelva un marcador de clave entre corchetes cuando falte una clave, y sustituye por una llamada a Texts.Get cada etiqueta escrita directamente en la página. Después rellena el panel de vista previa con valores reales: da formato a la fecha de hoy, a una cantidad y a un importe monetario con Application.CurrentCulture en lugar de componer las cadenas a mano, y termina con una nota breve de laboratorio que separe el valor almacenado de su presentación sensible a la cultura.

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

- ✓ Página del dashboard de GlobalDesk cuyas etiquetas proceden todas de claves de recursos, sin texto visible para el usuario escrito en el código
- ✓ Archivo neutro Resources/Strings.resx con al menos doce claves semánticas estables
- ✓ Helper Texts sobre Wisej.Resources.ResourceManager que marca una clave ausente en lugar de devolver una cadena vacía
- ✓ Panel de vista previa de la cultura que da formato a una fecha, un número y una moneda con Application.CurrentCulture
- ✓ Nota de laboratorio que explica qué valores son datos y cuáles son presentación sensible a la cultura

Por qué tiene esta forma

GlobalDesk es un pequeño service desk escrito para una sola oficina. Cada encabezado, etiqueta de botón y mensaje de confirmación es un literal en inglés tecleado directamente en el C#, y el dashboard imprime

importes y fechas tal y como los escribe una máquina estadounidense. Cuando abre una sucursal alemana, ninguno de esos textos tiene adónde ir, y el desarrollador que empieza editando etiquetas a mano se encuentra con la segunda mitad del problema: una etiqueta en alemán sobre un importe que sigue leyéndose 1,850.75 no es una pantalla alemana.

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
localización	Adaptar una aplicación al idioma y a la cultura del usuario: texto traducido, fechas, números e importes correctamente formateados, espacio para palabras más largas y un diseño que pueda reflejarse para los idiomas de derecha a izquierda.
nombre de cultura	Un identificador como en, de-DE o fr-CA que nombra un idioma y, opcionalmente, una región; la parte del idioma decide qué recursos se prefieren y el nombre completo gobierna las convenciones de formato.
recurso neutro	El conjunto completo de recursos de repliegue, guardado en un archivo como Resources/Strings.resx sin cultura en su nombre; toda clave que pida la aplicación debe existir aquí.
Wisej.Resources.ResourceManager	El objeto que toma un nombre base de recursos y un ensamblado y devuelve el valor de una clave en la cultura que mejor coincide, de modo que tu código nunca abre un archivo por su cuenta.
repliegue de cultura	La regla de.NET para encontrar un valor: probar la cultura completa, después la cultura solo de idioma, después el recurso neutro, y usar el primero que tenga la clave.
clave de recurso semántica	Un nombre estable que describe el significado y no la redacción, como CustomerEditor.Save o Validation.Required, para que el texto en inglés se pueda reescribir sin tocar ningún otro idioma.
Application.CurrentCulture	La cultura de la sesión actual de Wisej.NET; dos usuarios con la sesión iniciada a la vez pueden tener dos valores distintos, y por eso la cultura nunca debe vivir en un campo estático.

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 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 1 – «Fundamentos de localización: cultura, idioma y recursos» – 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-01.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: Crea la página del dashboard de GlobalDesk con un encabezado de bienvenida, un botón de navegación Customers y un pequeño panel de vista previa de la cultura. Añade `Resources/Strings.resx` como recurso neutro con al menos doce claves semánticas estables como `Navigation.Customers` y `CustomerEditor.Save`, escribe un helper estático `Texts` sobre `Wisej.Resources.ResourceManager` que devuelva un marcador de clave entre corchetes cuando falte una clave, y sustituye por una llamada a `Texts.Get` cada etiqueta escrita directamente en la página. Después rellena el panel de vista previa con valores reales: da formato a la fecha de hoy, a una cantidad y a un importe monetario con `Application.CurrentCulture` en lugar de componer las cadenas a mano, y termina con una nota breve de laboratorio que separe el valor almacenado de su presentación sensible a la cultura.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Página del dashboard de GlobalDesk cuyas etiquetas proceden todas de claves de recursos, sin texto visible para el usuario escrito en el código
 - Archivo neutro `Resources/Strings.resx` con al menos doce claves semánticas estables
 - Helper `Texts` sobre `Wisej.Resources.ResourceManager` que marca una clave ausente en lugar de devolver una cadena vacía
 - Panel de vista previa de la cultura que da formato a una fecha, un número y una moneda con `Application.CurrentCulture`
 - Nota de laboratorio que explica qué valores son datos y cuáles son presentación sensible a la cultura
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:

- Cambias la redacción en inglés del botón Save de "Save" a "Save customer". ¿Qué archivos tienen que cambiar si tu clave es `CustomerEditor.Save`, y qué archivos habrían tenido que

cambiar si la clave hubiera sido la propia frase en inglés?

- Tu helper Texts devuelve [Preview.Amount] en una etiqueta. Nombra dos errores distintos que produzcan exactamente ese marcador y explica cómo los distinguirías sin un depurador.
- Un compañero propone guardar el total del pedido en la base de datos como la cadena "1.850,75 €" para que los usuarios alemanes siempre lo vean bien. Explica qué se rompe y qué parte de esa cadena son datos y qué parte es presentación.

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:

- ✓ Cambias la redacción en inglés del botón Save de "Save" a "Save customer". ¿Qué archivos tienen que cambiar si tu clave es CustomerEditor.Save, y qué archivos habrían tenido que cambiar si la clave hubiera sido la propia frase en inglés?
- ✓ Tu helper Texts devuelve [Preview.Amount] en una etiqueta. Nombra dos errores distintos que produzcan exactamente ese marcador y explica cómo los distinguirías sin un depurador.
- ✓ Un compañero propone guardar el total del pedido en la base de datos como la cadena "1.850,75 €" para que los usuarios alemanes siempre lo vean bien. Explica qué se rompe y qué parte de esa cadena son datos y qué parte es presentación.

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 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-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 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."



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.