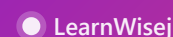


Ejercicio de programación con IA

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

Ejecuta una ronda de traducción honesta sobre GlobalDesk. Abre la solución en ResX Resource Manager, añade el italiano como nueva columna de idioma y usa el filtro de entradas sin traducir para listar todas las claves que aún no tienen valor. Añade comentarios a las tres claves cuyo significado es ambiguo fuera de contexto, empezando por la etiqueta de una sola palabra Open; después traduce o genera un borrador automático de los valores en italiano y marca como invariantes el nombre del producto y los dos identificadores técnicos para que nadie los traduzca. Exporta las columnas neutra e italiana, vuelve a importarlas y revisa el diff resultante del .resx antes de confirmarlo como un único cambio revisable. Termina con una nota de una página sobre el flujo de trabajo que compare el editor de Visual Studio, ResX Resource Manager y una plataforma en la nube para un equipo cuyos traductores no son desarrolladores.

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

- ✓ Strings.it.resx producido con ResX Resource Manager sin ninguna clave sin traducir
- ✓ Comentarios de recurso en al menos tres claves ambiguas para que el traductor tenga el contexto
- ✓ Marcas de invariante en el nombre del producto y en los identificadores que no deben traducirse
- ✓ Exportación, reimportación y un diff del.resx revisado y confirmado como un único cambio
- ✓ Nota de flujo de trabajo que compara el editor de Visual Studio, ResX Resource Manager y un TMS en la nube

Por qué tiene esta forma

GlobalDesk lleva ya un Resources/Strings.resx neutro, un Strings.de.resx alemán, recursos de diseñador en el editor de clientes y un Resources-de.resx para las etiquetas de sistema de Wisej. Añadir el italiano con el editor de Visual Studio significa abrir esos archivos de uno en uno y comparar listas de claves a ojo, y la

persona que habla italiano no suele ser la que puede abrir la solución. Este módulo convierte la traducción en un flujo de trabajo con una lista de huecos visible, una entrega y un paso de revisión.

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
Vista de matriz	Las claves en las filas y los idiomas en las columnas, así que un valor italiano que falta y un término alemán que se ha desviado se ven a la vez.
ResX Resource Manager	Una herramienta de localización.NET de código abierto, disponible como extensión de Visual Studio y como aplicación independiente, que muestra los recursos RESX de una solución en una única cuadrícula central.
Columna de idioma	Añadir un idioma de destino crea el archivo específico de esa cultura junto al neutro y te da una columna vacía que rellenar; las claves no cambian nunca.
Filtro de entradas sin traducir	Lista solo las entradas sin valor en el idioma seleccionado, de modo que "¿hemos terminado?" pasa a ser un número en lugar de una opinión.
Comentario de recurso	Una nota guardada junto al valor que le dice al traductor si Open es un verbo, un adjetivo, el estado de un ticket o una orden.
Cadena invariante	Una marca en las entradas que nunca deben traducirse, como el nombre del producto y los identificadores técnicos.
Ida y vuelta de exportación e importación	Columnas enviadas a un traductor o a una plataforma compatible con RESX y devueltas como archivos.resx específicos de cultura que la compilación trata como cualquier otro recurso.

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 6. Puedes abrir tú mismo [module-06.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 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 6 – «Flujo de trabajo de traducción: ResX Resource Manager y otras herramientas» – 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-06.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: Ejecuta una ronda de traducción honesta sobre GlobalDesk. Abre la solución en ResX Resource Manager, añade el italiano como nueva columna de idioma y usa el filtro de entradas sin traducir para listar todas las claves que aún no tienen valor. Añade comentarios a las tres claves cuyo significado es ambiguo fuera de contexto, empezando por la etiqueta de una sola palabra Open; después traduce o genera un borrador automático de los valores en italiano y marca como invariantes el nombre del producto y los dos identificadores técnicos para que nadie los traduzca. Exporta las columnas neutra e italiana, vuelve a importarlas y revisa el diff resultante del .resx antes de confirmarlo como un único cambio revisable. Termina con una nota de una página sobre el flujo de trabajo que compare el editor de Visual Studio, ResX Resource Manager y una plataforma en la nube para un equipo cuyos traductores no son desarrolladores.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Strings.it.resx producido con ResX Resource Manager sin ninguna clave sin traducir
 - Comentarios de recurso en al menos tres claves ambiguas para que el traductor tenga el contexto
 - Marcas de invariante en el nombre del producto y en los identificadores que no deben traducirse
 - Exportación, reimportación y un diff del.resx revisado y confirmado como un único cambio
 - Nota de flujo de trabajo que compara el editor de Visual Studio, ResX Resource Manager y un TMS en la nube
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:

- Tu columna italiana está completa y la aplicación se ve bien en it-IT, pero un compañero

afirma que dos claves nunca se tradujeron. ¿Cómo pueden ser ciertas las dos cosas a la vez y qué función de la herramienta zanja la discusión?

– Un borrador automático en italiano devolvió Último ordine: {1} il {0} para una clave cuyo valor neutro es Last order: {0} on {1}. ¿Qué se rompe, cuándo se rompe y qué lo habría detectado antes de la confirmación?

– Tu equipo tiene cinco idiomas, dos de ellos traducidos por una agencia que nunca ha abierto Visual Studio. ¿Cuál de los tres flujos de trabajo adoptarías y qué tendría que cambiar antes de pasar al siguiente nivel?

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:

- ✓ Tu columna italiana está completa y la aplicación se ve bien en it-IT, pero un compañero afirma que dos claves nunca se tradujeron. ¿Cómo pueden ser ciertas las dos cosas a la vez y qué función de la herramienta zanja la discusión?
- ✓ Un borrador automático en italiano devolvió Último ordine: {1} il {0} para una clave cuyo valor neutro es Last order: {0} on {1}. ¿Qué se rompe, cuándo se rompe y qué lo habría detectado antes de la confirmación?
- ✓ Tu equipo tiene cinco idiomas, dos de ellos traducidos por una agencia que nunca ha abierto Visual Studio. ¿Cuál de los tres flujos de trabajo adoptarías y qué tendría que cambiar antes de pasar al siguiente nivel?

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-06.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."



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.