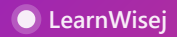


Ejercicio de programación con IA

Construye el ejemplo FieldOps 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

Convierte FieldOps en una PWA: añade el manifiesto con iconos, color del tema y visualización standalone, registra un service worker que almacene en caché el shell y los recursos estáticos, muestra una invitación a instalar después de la primera sesión correcta, muestra una página sin conexión con el último resumen sincronizado de las órdenes de trabajo cuando el servidor no esté disponible, e implementa el flujo de actualización que recarga la aplicación cuando hay una versión nueva.

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

- ✓ Manifiesto de la aplicación web con iconos y colores
- ✓ Service worker que almacena el shell en caché
- ✓ Invitación a instalar tras la primera sesión
- ✓ Página sin conexión con el último resumen sincronizado
- ✓ Flujo de actualización y comprobación de versión

Por qué tiene esta forma

Los técnicos de FieldOps no abren un navegador y escriben una URL. Tocaban un icono en un teléfono que ha pasado toda la mañana en una furgoneta, en una sala de máquinas donde la señal desaparece a mitad de un trabajo. Una progressive web app les da ese icono, una ventana sin el marco del navegador y una página que sigue diciendo algo útil cuando desaparece la red; pero FieldOps es una aplicación Wisej.NET, así que su interfaz vive en el servidor, y fingir lo contrario es la forma en que una PWA se convierte en una incidencia de soporte.

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
manifiesto de la aplicación web	Un pequeño archivo JSON, enlazado desde la página de inicio, que indica al navegador el nombre de la aplicación, sus iconos, sus colores, su modo de visualización y su URL de inicio; sin él, el navegador no tiene nada que instalar.
modo de visualización	El miembro display del manifiesto. standalone abre FieldOps en su propia ventana sin barra de direcciones, que es lo que hace que una aplicación instalada deje de parecer un marcador.
iconos maskable	Iconos declarados con purpose: "maskable" y un margen de seguridad, para que Android pueda recortarlos con su propia forma sin cortar el dibujo.
service worker	Un script que el navegador ejecuta junto a la página, servido desde el propio origen de la aplicación por HTTPS, que puede responder a las solicitudes de la página desde una caché cuando la conexión es lenta o no existe.
shell de la aplicación	Los archivos estáticos que necesita una sesión antes de que el servidor pueda decir nada: la página de inicio, los scripts de cliente, el tema y los iconos. Es lo que el worker puede almacenar en caché.
primero caché frente a solo red	La regla de enrutamiento dentro del controlador fetch del worker: los archivos estáticos del shell se sirven primero desde la caché, mientras que todo lo que la sesión de Wisej.NET intercambia con el servidor debe ir directamente a la red, nunca a una copia en caché.
ciclo de vida de la actualización	Un worker nuevo se instala mientras el antiguo sigue controlando la página, y después espera. skipWaiting, clients.claim y una recarga controlada son lo que convierte ese worker en espera en la versión que el técnico ejecuta realmente.

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 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 FieldOps 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 – «Configuración de una progressive web app» – del curso de LearnWisej «Empaquetado híbrido y nativo», 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/hybrid-native-packaging/specs/module-02.md>

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

1. Proyecto «FieldOps», 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: Convierte FieldOps en una PWA: añade el manifiesto con iconos, color del tema y visualización standalone, registra un service worker que almacene en caché el shell y los recursos estáticos, muestra una invitación a instalar después de la primera sesión correcta, muestra una página sin conexión con el último resumen sincronizado de las órdenes de trabajo cuando el servidor no esté disponible, e implementa el flujo de actualización que recarga la aplicación cuando hay una versión nueva.
3. Genera estos entregables e indícame dónde está cada uno dentro de la solución:
 - Manifiesto de la aplicación web con iconos y colores
 - Service worker que almacena el shell en caché
 - Invitación a instalar tras la primera sesión
 - Página sin conexión con el último resumen sincronizado
 - Flujo de actualización y comprobación de versión
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 revisor pregunta por qué el service worker no almacena simplemente en caché todas las respuestas para que FieldOps siga funcionando bajo tierra. Explica qué haría realmente una respuesta reproducida de la sesión de Wisej.NET en la pantalla del técnico y dónde está el límite honesto.
- Tu oferta de instalación funciona en tu escritorio pero nunca aparece en el teléfono del técnico. Nombra tres condiciones distintas que podrían ser la causa y di cómo las distinguirías desde el dispositivo.
- Un técnico tiene la aplicación instalada y abierta cuando despliegas una nueva compilación. Describe todos los pasos entre tu despliegue y ese teléfono ejecutando el nuevo shell, e identifica qué paso fallaría en silencio si el nombre de la caché no cambiara

nunca.

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 revisor pregunta por qué el service worker no almacena simplemente en caché todas las respuestas para que FieldOps siga funcionando bajo tierra. Explica qué haría realmente una respuesta reproducida de la sesión de Wisej.NET en la pantalla del técnico y dónde está el límite honesto.
- ✓ Tu oferta de instalación funciona en tu escritorio pero nunca aparece en el teléfono del técnico. Nombra tres condiciones distintas que podrían ser la causa y di cómo las distinguirías desde el dispositivo.
- ✓ Un técnico tiene la aplicación instalada y abierta cuando despliegas una nueva compilación. Describe todos los pasos entre tu despliegue y ese teléfono ejecutando el nuevo shell, e identifica qué paso fallaría en silencio si el nombre de la caché no cambiara nunca.

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 FieldOps 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 FieldOps 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/hybrid-native-packaging/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."*



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.