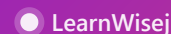


Esercizio di coding con l'IA

Costruisci il progetto OperationsConsole di questo modulo con ChatGPT o Claude partendo da un prompt già pronto che indirizza il modello alla specifica del modulo, poi rivedi, esegui ed estendi quello che ti restituisce — confrontandolo anche con la nostra build di riferimento su GitHub.



Questo progetto non lo scriverai a mano. Lo dirigerai.

Quasi nessuno scrive più codice Wisej.NET da un file vuoto: un assistente scrive la prima bozza e lo sviluppatore decide se è giusta. Funziona però solo se conosci la materia, ed è quello che ti ha dato il resto di questo modulo. Qui affidi quelle conoscenze a ChatGPT o a Claude, ti fai restituire il progetto del modulo come soluzione Visual Studio funzionante, e lo rivedi come rivedresti la pull request di un collega.

Che cosa stai chiedendo

Costruisci il tab dashboard di Operations Console: un grafico ChartJS a linee o a barre le cui etichette e DataSet arrivano da un view model della dashboard, una ProgressBar o un indicatore tipo gauge per un valore di stato, un'anteprima di contenuto con PdfViewer o HtmlPanel, un flusso di Upload, una mappa o un segnaposto di contenuto, e un unico metodo di refresh che aggiorna tutti i controlli della dashboard a partire da un solo modello.

Il modulo chiede questi deliverable, e il prompt qui sotto li chiede all'assistente esattamente così:

- ✓ Tab dashboard con un grafico ChartJS a linee o a barre
- ✓ Etichette e DataSet del grafico costruiti da un view model della dashboard
- ✓ ProgressBar o indicatore tipo gauge per un valore di stato
- ✓ Anteprima di contenuto con PdfViewer o HtmlPanel più un flusso di Upload
- ✓ Un unico metodo di refresh che aggiorna tutti i controlli della dashboard

Perché è fatto così

Le dashboard non sono solo grafici. Una dashboard è una composizione di controlli che trasforma i dati in decisioni, e il tab dashboard della Operations Console sarà la schermata che un responsabile del servizio apre per prima ogni mattina. Wisej.NET ti dà controlli nativi per contenuti e media più estensioni per grafici, mappe, visualizzazione di documenti e gauge; l'abilità intermedia sta nel capire che cosa appartiene al modello a controlli di Wisej.NET, che cosa conviene delegare a una libreria JavaScript specializzata incapsulata come estensione o Widget, e come far aggiornare l'intera schermata come un'unica unità invece che come un mucchio di elementi visuali che si ricaricano ciascuno per conto proprio.

Questi sono i termini che l'assistente deve azzeccare. Se quello che ti restituisce li usa con leggerezza, non ha capito il modulo:

Termine	Che cosa significa qui
dashboard	Una composizione di controlli il cui unico compito è rispondere a una domanda che porta a una decisione; se un elemento visivo non aiuta qualcuno a decidere qualcosa, è rumore.
estensione ChartJS	Un pacchetto NuGet che espone Chart.js come componente lato server: il server possiede lo stato del grafico, il browser ne rende l'aspetto, e il grafico entra nello stesso modello a eventi di tutti gli altri controlli.
etichette e DataSet	Le due metà dei dati di un grafico; le etichette danno un nome ai punti lungo l'asse e ogni DataSet porta una serie, quindi si impostano sempre insieme.
view model della dashboard	Un oggetto modellato per la visualizzazione, come DashboardModel, che contiene mesi, conteggi e un valore di completamento, separato sia da qualsiasi entità del database sia dal codice che lo disegna.
confine del browser	Il client è il browser e il server è dove gira il tuo codice Wisej.NET; file, PDF, HTML e tile delle mappe attraversano quella linea in una direzione o nell'altra.
Upload contro percorso locale	Upload sposta un file dal browser al server; non è la stessa cosa che leggere un percorso sul desktop, perché il server non può mai sfogliare il disco dell'utente.
refresh basato sul modello	Una sola chiamata al servizio restituisce un modello completo della dashboard e un solo metodo aggiorna tutti i controlli a partire da quello, in blocco, con un orario visibile.

Costruiscilo — scegli il tuo assistente

Entrambi i pulsanti aprono una nuova scheda con lo **stesso prompt già pronto**. Il prompt indirizza il modello alla specifica di questo modulo — un unico file Markdown con la lezione, la traccia del lab e i deliverable — così lavora sul materiale vero invece di tirare a indovinare che cosa sia il Modulo 6. Puoi aprire tu stesso [module-06.md](#) per vedere esattamente che cosa leggerà.



Usa ChatGPT

Apri [chatgpt.com](#) con il prompt di questo modulo già caricato. Funziona con GPT-5 o Codex; qualsiasi piano che possa navigare il web recupererà la specifica.



Usa Claude

Apri [claude.ai](#) con lo stesso prompt già caricato. Chi usa Claude Code può invece incollarlo in un terminale dentro una cartella vuota.



Mandalo così com'è e ottieni un progetto, non un frammento

Il prompt si chiude chiedendo l'intera soluzione in un unico `.zip` con i passi per compilarla ed eseguirla. Scaricalo, apri la soluzione in Visual Studio, premi F5 e OperationsConsole si apre su `localhost`. Ti servono l'SDK .NET e il pacchetto NuGet `Wisej-4`; tutto il resto il prompt chiede all'assistente di scriverlo da sé.

Sto seguendo il Modulo 6 - "Grafici, dashboard, contenuti, media e documenti" - del corso LearnWisej "Padroneggiare la libreria di controlli", e voglio costruire con te il progetto del modulo. Rispondimi in italiano.

Comincia leggendo la specifica del modulo. È un unico file Markdown (in inglese) che contiene l'intera lezione, la traccia del lab e i deliverable:

<https://learnwisej.com/assets/courses/mastering-the-control-library/specs/module-06.md>

Poi costruisci un progetto Wisej.NET 4 completo che io possa aprire in Visual Studio:

1. Progetto "OperationsConsole", C#, net8.0 o successivo, pacchetto NuGet Wisej-4 4.1.x, host Kestrel che chiama app.UseWisej().
2. Costruisci quello che chiede il lab della specifica: Costruisci il tab dashboard di Operations Console: un grafico ChartJS a linee o a barre le cui etichette e DataSet arrivano da un view model della dashboard, una ProgressBar o un indicatore tipo gauge per un valore di stato, un'anteprima di contenuto con PdfViewer o HtmlPanel, un flusso di Upload, una mappa o un segnaposto di contenuto, e un unico metodo di refresh che aggiorna tutti i controlli della dashboard a partire da un solo modello.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Tab dashboard con un grafico ChartJS a linee o a barre
 - Etichette e DataSet del grafico costruiti da un view model della dashboard
 - ProgressBar o indicatore tipo gauge per un valore di stato
 - Anteprima di contenuto con PdfViewer o HtmlPanel più un flusso di Upload
 - Un unico metodo di refresh che aggiorna tutti i controlli della dashboard
4. Segui la specifica ovunque nomini una classe, una proprietà o un evento di Wisej.NET - usa esattamente quello. Dove lascia una scelta aperta, dimmi quale API hai scelto e perché.
5. Tieni l'autorità sul server: stato e validazione vivono in .NET, il browser disegna e riferisce. Non spostare mai una decisione in JavaScript solo perché lì è più comodo.
6. Rendi il risultato dimostrabile: devo poterlo eseguire e vedere il comportamento di cui parla questo modulo, compreso che cosa succede quando qualcosa va storto.

Prima di scrivere codice, riformula con parole tue che cosa costruirai e dove starà ogni pezzo, così posso correggere il progetto finché costa poco.

Quando il codice è pronto, impacchetta l'intera soluzione in un unico file .zip scaricabile e dammi i passi esatti per compilarla ed eseguirla.

Infine rispondi a queste domande della specifica, usando come prova la tua implementazione:

- Quale controllo del tuo tab dashboard fa il lavoro più importante, e quale domanda resterebbe senza risposta se lo togliessi?
- Che cosa si romperebbe per primo se l'insieme di dati dei ticket diventasse dieci volte più grande, e perché restituire aggregati dal servizio protegge il grafico?
- Dove si trova il file caricato in ciascun passo del flusso di Upload, e perché il server non può mai aprire un percorso digitato dall'utente?

Chiedimi qualsiasi cosa non ti sia chiara prima di cominciare.

Poi rivedi quello che ti è tornato indietro

Codice che compila non è ancora codice giusto. Le domande interessanti sono domande di progetto: è questo il controllo Wisej.NET che il compito richiede davvero, segue il ciclo di vita che il framework ti offre,

lo stato autorevole sta sul server o è stato parcheggiato nel browser perché era comodo? Leggi il risultato da revisore, non da cliente.

Questo modulo ti dà la checklist. Rispondi a ciascuna domanda guardando il codice che hai appena ricevuto:

- ✓ Quale controllo del tuo tab dashboard fa il lavoro più importante, e quale domanda resterebbe senza risposta se lo togliessi?
- ✓ Che cosa si romperebbe per primo se l'insieme di dati dei ticket diventasse dieci volte più grande, e perché restituire aggregati dal servizio protegge il grafico?
- ✓ Dove si trova il file caricato in ciascun passo del flusso di Upload, e perché il server non può mai aprire un percorso digitato dall'utente?

Tutto ciò che hai dovuto correggere vale la pena di scriverlo. Una breve nota con le convenzioni e le decisioni a cui sei arrivato, incollata nel primo messaggio della sessione successiva, vale più di qualsiasi prompt ingegnoso.



Questo l'abbiamo già costruito — confronta il tuo con il nostro

L'implementazione di riferimento del Modulo 6 si trova nel repository pubblico degli esempi. Costruisci prima il tuo, poi apri il nostro: la parte interessante sono le differenze.



[Apri l'esempio del Modulo 6 su GitHub](#)

Oppure clona tutto con `git clone https://github.com/iceteagroup/LearnWisej-Samples` e apri `Mastering the Control Library Course/Module 6`.



Il nostro esempio è un punto di partenza, non un prodotto finito

È stato costruito nello stesso modo in cui stai costruendo il tuo — con un prompt, poi corretto — e si ferma deliberatamente dove si ferma il modulo. Aspettati soluzioni provvisorie dove andrebbe codice di produzione, gestione degli errori essenziale e nessun test. Prendilo come un invito: usa quello che hai imparato in questo modulo, più l'assistente, per portarlo più avanti. Migliorare codice che non hai scritto tu è l'abilità che questo corso insegna davvero.

Tre prompt per andare oltre

Punta il tuo assistente sul progetto che hai appena costruito — o sul nostro e prova questi. Ognuno è un esercizio di revisione ed estensione, non una riscrittura.

- ✓ **Fallo sopravvivere a una brutta giornata.** *"Esamina questo progetto OperationsConsole cercando le modalità di guasto che non gestisce ancora — input sbagliato o mancante, una dipendenza lenta o non disponibile, lavoro che arriva dopo che l'utente ha cambiato pagina, e due azioni che si accavallano. Per ciascuna dimmi prima se è un rischio reale in Wisej.NET 4 e perché, e solo dopo scrivi la correzione. Dove decidi che non serve gestire nulla, dillo e spiega il ragionamento invece di aggiungere codice difensivo."*
- ✓ **Aggiungi il secondo caso.** *"Estendi questo progetto OperationsConsole con un secondo caso che segue lo stesso schema del primo — un'altra entità, schermata o controllo della stessa forma, collegato allo stesso modo. Estrai in una classe base o in un servizio ciò che è davvero condiviso, ma solo dove se lo merita; due cose simili non chiedono sempre un'astrazione. Mostrami il diff e giustifica ogni cosa che hai estratto."*
- ✓ **Dimostra che funziona davvero.** *"Passa in rassegna questo progetto deliverable per deliverable rispetto alla specifica del modulo su <https://learnwisej.com/assets/courses/mastering-the-control-library/specs/module-06.md> e mostrami la prova che ciascuno è davvero fatto — il percorso nel codice, su che cosa cliccare e che cosa dovrei vedere. Dove un deliverable è soddisfatto solo in parte, dillo chiaramente invece di arrampicarti sugli specchi. Poi aggiungi quello che manca."*



Portalo dentro il lab

Il lab pratico che chiude questo modulo chiede gli stessi deliverable che hai appena fatto produrre all'assistente. Usa quello che hai qui come forma della tua risposta e come prova a sostegno — non come la risposta in sé. Il lab viene valutato sul tuo ragionamento, e quella parte deve essere tua.