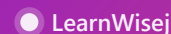


Esercizio di coding con l'IA

Costruisci il progetto AdaptiveOps 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.



Scrivere il codice è ormai la metà facile.

Ottenere codice funzionante per "Fondamenti di layout: docking, ancoraggio, AutoSize, AutoScroll e composizione dello shell" da un assistente moderno richiede un buon prompt. Giudicarlo, correggerlo ed estenderlo richiede tutto quello che questo modulo ha trattato. È su questa divisione che è costruito l'esercizio: tu metti la specifica e la revisione, l'assistente mette la digitazione.

Che cosa stai chiedendo

Riorganizza lo shell di Adaptive Operations Console in contenitori annidati: una toolbar in alto, una barra di navigazione a sinistra, un'area di lavoro su Fill, un pannello dei dettagli a destra e una regione di stato, ciascuno un Panel o uno UserControl con il proprio layout locale. Elimina ogni gestore Resize che imposta Bounds, correggi l'ordine di docking nel designer, usa Padding per gli spazi tra le regioni in dock, dai all'area di lavoro e alla griglia un MinimumSize, e fai scorrere l'editor dei dettagli con AutoScroll e barre di scorrimento nascoste alle larghezze strette.

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

- ✓ Shell ricostruito con contenitori annidati con le regioni toolbar, navigazione, area di lavoro, dettagli e stato
- ✓ Rimosso tutto il codice Resize manuale che impostava Bounds
- ✓ Ordine di docking, Padding e MinimumSize impostati in modo intenzionale su ogni regione
- ✓ Editor dei dettagli che scorre tramite AutoScroll con barre di scorrimento nascoste alle larghezze strette
- ✓ Regioni di navigazione e dettagli estratte in UserControl

Perché è fatto così

Adaptive Operations Console ora ha un tema e CSS circoscritto, ma il suo shell sposta ancora i controlli a mano: un gestore Resize ricalcola Bounds per la toolbar, la barra di navigazione, la griglia dei ticket e l'editor dei dettagli ogni volta che il browser cambia dimensione. Funziona alla larghezza che lo sviluppatore ha provato, si rompe a tutte le altre e non può reggere i profili tablet e telefono che introduce il Modulo 6. Il design adattivo in Wisej.NET si basa su contenitori e regole di layout, non su coordinate: regioni che occupano i bordi, un'area di lavoro che riempie ciò che resta, ed editor che scorrono invece di sovrapporsi.

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
motore di layout predefinito	Ogni controllo Wisej.NET ha un motore di layout; quello predefinito supporta posizione assoluta, docking e ancoraggio, così una schermata può combinare il posizionamento preciso nel designer con regioni che si ridimensionano da sole.
priorità di docking	I figli in dock occupano lo spazio nell'ordine in cui compaiono nella collezione dei controlli del genitore, ed è per questo che gli stessi cinque pannelli producono uno shell corretto o uno sovrapposto a seconda dell'ordine dei figli.
DisplayRectangle e Padding	I controlli in dock vengono disposti dentro il DisplayRectangle del genitore, che il Padding riduce; il motore predefinito ignora il Margin per i figli in dock, quindi gli spazi vengono dal padding o da controlli distanziatori.
ancoraggio	Un ancoraggio mantiene la distanza di un controllo dai lati selezionati del genitore, il che si adatta alle regioni simili a form: label in alto a sinistra, caselle di testo left-right, pulsanti di comando in basso a destra.
AutoSize e layout circolare	Un contenitore che si dimensiona in base a figli in dock o ancorati a esso non ha una risposta stabile; quel ciclo è il principale avvertimento del modulo su AutoSize.
AutoScroll con barre nascoste	AutoScroll permette a un contenitore di scorrere quando il contenuto non ci sta, e ScrollBars.Hidden mantiene lo scorrimento al tocco sui telefoni senza disegnare barre di scorrimento.
regione UserControl	Una regione dello shell con il proprio layout locale e una piccola superficie pubblica, così lo shell mostra, nasconde, mette in dock o sposta l'intera regione per profilo invece di conoscerne le coordinate interne.

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 4. Puoi aprire tu stesso [module-04.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 AdaptiveOps 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 4 - "Fondamenti di layout: docking, ancoraggio, AutoSize, AutoScroll e composizione dello shell" - del corso LearnWisej "Temi e layout responsive", 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/theming-and-responsive-layouts/specs/module-04.md>

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

1. Progetto "AdaptiveOps", 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: Riorganizza lo shell di Adaptive Operations Console in contenitori annidati: una toolbar in alto, una barra di navigazione a sinistra, un'area di lavoro su Fill, un pannello dei dettagli a destra e una regione di stato, ciascuno un Panel o uno UserControl con il proprio layout locale. Elimina ogni gestore Resize che imposta Bounds, correggi l'ordine di docking nel designer, usa Padding per gli spazi tra le regioni in dock, dai all'area di lavoro e alla griglia un MinimumSize, e fai scorrere l'editor dei dettagli con AutoScroll e barre di scorrimento nascoste alle larghezze strette.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Shell ricostruito con contenitori annidati con le regioni toolbar, navigazione, area di lavoro, dettagli e stato
 - Rimosso tutto il codice Resize manuale che impostava Bounds
 - Ordine di docking, Padding e MinimumSize impostati in modo intenzionale su ogni regione
 - Editor dei dettagli che scorre tramite AutoScroll con barre di scorrimento nascoste alle larghezze strette
 - Regioni di navigazione e dettagli estratte in UserControl
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:

- Se nell'ordine dei figli il pannello dell'area di lavoro fosse messo in dock prima del pannello di stato, che cosa vedrebbe l'utente in fondo alla griglia, e quale comando del designer lo corregge senza cambiare una sola dimensione?
- Hai impostato un Margin di 12 pixel sulla barra di navigazione e lo spazio non è comparso. Perché, e quali due meccanismi producono invece uno spazio tra le regioni in dock?
- L'editor dei dettagli è uno UserControl con AutoScroll. Quali dei suoi membri deve conoscere lo shell per spostarlo sotto la griglia su un tablet, e di quali non deve mai avere bisogno?

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:

- ✓ Se nell'ordine dei figli il pannello dell'area di lavoro fosse messo in dock prima del pannello di stato, che cosa vedrebbe l'utente in fondo alla griglia, e quale comando del designer lo corregge senza cambiare una sola dimensione?
- ✓ Hai impostato un Margin di 12 pixel sulla barra di navigazione e lo spazio non è comparso. Perché, e quali due meccanismi producono invece uno spazio tra le regioni in dock?
- ✓ L'editor dei dettagli è uno UserControl con AutoScroll. Quali dei suoi membri deve conoscere lo shell per spostarlo sotto la griglia su un tablet, e di quali non deve mai avere bisogno?

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 4 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 4 su GitHub](#)

Oppure clona tutto con `git clone https://github.com/iceteagroup/LearnWisej-Samples` e apri `Theming and Responsive Layouts Course/Module 4`.



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.

- ✓ **Sostituisci un pezzo senza spostare le giunzioni.** *"Scegli la parte di questo progetto AdaptiveOps che più evidentemente sta al posto di qualcosa di reale — una sorgente dati finta, un archivio in memoria, un servizio semplificato — e sostituiscila con un'implementazione di forma produttiva. Il contratto ai due lati non deve cambiare: stessi tipi, stesse chiamate, stesso comportamento in caso di errore. Dimmi che cosa l'implementazione reale ti ha costretto a gestire e che la versione provvisoria non gestiva."*
- ✓ **Fallo sopravvivere a una brutta giornata.** *"Esamina questo progetto AdaptiveOps 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 AdaptiveOps 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."*



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.