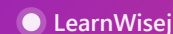


## Esercizio di coding con l'IA

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.



### Lo stesso modulo, costruito come costruirai davvero le cose.

Tutte le altre sezioni di questo modulo spiegano come funziona "FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel e proprietà di layout estese". Questa ti chiede di produrlo: dirigendo un assistente con la specifica del modulo stesso, poi verificando che il risultato giri e faccia quello che descrive la traccia del lab. È il flusso di lavoro che userai sul lavoro vero, provato su qualcosa di cui conosci già la risposta giusta.

## Che cosa stai chiedendo

Costruisci tre versioni dell'area contenuti di Adaptive Operations Console: un'area di filtri e card delle metriche con FlowLayoutPanel che va a capo e usa FillWeight e FlowBreak, un editor dei ticket con TableLayoutPanel con label ed editor allineati governati da RowStyles e ColumnStyles, e uno shell della dashboard con FlexLayoutPanel in cui le regioni della lista e dei dettagli si dividono lo spazio tramite FillWeight, con AlignX, AlignY, MinimumSize e MaximumSize impostati. Costruisci una delle tre da codice con le estensioni fluent di Wisej.Web.Markup e confronta come ciascuna regge un ridimensionamento del browser.

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

- ✓ Area di filtri e card delle metriche con FlowLayoutPanel che usa WrapContents, FillWeight e FlowBreak
- ✓ Editor dei ticket con TableLayoutPanel con RowStyles, ColumnStyles e span per campi allineati
- ✓ Shell della dashboard con FlexLayoutPanel con FillWeight, AlignX, AlignY, MinimumSize e MaximumSize
- ✓ Una regione costruita da codice con le estensioni fluent di Wisej.Web.Markup
- ✓ Screenshot di confronto al ridimensionamento e una nota su quale contenitore si adatta a ogni regione

## Perché è fatto così

Il Modulo 4 ha dato ad Adaptive Operations Console uno shell di regioni in dock, ma il docking decide solo dove sta una regione, non come si comporta il contenuto al suo interno quando il browser è largo 1400 pixel il lunedì e 700 su un tablet il martedì. Una riga di card delle metriche dovrebbe andare a capo, un editor dei ticket dovrebbe mantenere allineati label ed editor per quanto lunga sia la traduzione tedesca, e le regioni della lista e dei dettagli dovrebbero dividersi l'area di lavoro nella proporzione che

decidi tu. Ognuno è un comportamento del contenuto diverso, e Wisej.NET fornisce un contenitore per ciascuno: `FlowLayoutPanel`, `TableLayoutPanel` e `FlexLayoutPanel`. Scegliere quello sbagliato è il modo in cui i team finiscono per scrivere di nuovo codice di ridimensionamento.

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                                                                                                                                                                                                                                                 |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>FlowLayoutPanel</code>  | Dispone i figli in righe o colonne automatiche lungo <code>FlowDirection</code> , va a capo quando <code>WrapContents</code> è true e ignora <code>Location</code> , <code>Dock</code> e <code>Anchor</code> di ogni figlio, così chip e card si ridispongono da soli. |
| <code>TableLayoutPanel</code> | Colloca i figli in celle indicate da <code>Row</code> e <code>Column</code> , con <code>span</code> per le celle che si estendono, così label ed editor restano allineati mentre ogni figlio può ancora essere messo in dock o ancorato dentro la propria cella.       |
| <code>FlexLayoutPanel</code>  | Dispone i figli in un'unica riga o colonna, riempie sempre la sua area client e divide lo spazio rimanente per peso, il che si adatta alle regioni principali che suddividono l'area di lavoro in proporzione.                                                         |
| <code>FillWeight</code>       | Una proprietà estesa dei figli letta dai motori flow e flex: un peso maggiore di zero allarga il figlio nello spazio rimanente in proporzione agli altri figli con peso; zero mantiene la sua dimensione.                                                              |
| <code>FlowBreak</code>        | Una proprietà estesa dei figli di <code>FlowLayoutPanel</code> che forza il figlio successivo su una nuova riga o colonna, così una barra dei filtri termina una riga in modo deliberato invece di affidarsi alla larghezza.                                           |
| Stili di riga e di colonna    | <code>RowStyles</code> e <code>ColumnStyles</code> dichiarano se una riga o una colonna si dimensiona automaticamente, prende una dimensione fissa in pixel o divide il resto in percentuale; è qui che vivono le proporzioni di un form.                              |
| Margin e Padding              | Il <code>Padding</code> è lo spazio interno del contenitore attorno a tutti i figli; il <code>Margin</code> è la distanza del figlio dai vicini, e solo i motori che rispettano i margini lo useranno.                                                                 |

## 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 5. Puoi aprire tu stesso [module-05.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



### Usa Claude

Apri [claude.ai](#) con lo stesso prompt già caricato. Chi usa Claude Code può invece

Codex; qualsiasi piano che possa navigare il web recupererà la specifica.

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 5 - "FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel e proprietà di layout estese" - 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-05.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: Costruisci tre versioni dell'area contenuti di Adaptive Operations Console: un'area di filtri e card delle metriche con FlowLayoutPanel che va a capo e usa FillWeight e FlowBreak, un editor dei ticket con TableLayoutPanel con label ed editor allineati governati da RowStyles e ColumnStyles, e uno shell della dashboard con FlexLayoutPanel in cui le regioni della lista e dei dettagli si dividono lo spazio tramite FillWeight, con AlignX, AlignY, MinimumSize e MaximumSize impostati. Costruisci una delle tre da codice con le estensioni fluent di Wisej.Web.Markup e confronta come ciascuna regge un ridimensionamento del browser.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
  - Area di filtri e card delle metriche con FlowLayoutPanel che usa WrapContents, FillWeight e FlowBreak
  - Editor dei ticket con TableLayoutPanel con RowStyles, ColumnStyles e span per campi allineati
  - Shell della dashboard con FlexLayoutPanel con FillWeight, AlignX, AlignY, MinimumSize e MaximumSize
  - Una regione costruita da codice con le estensioni fluent di Wisej.Web.Markup
  - Screenshot di confronto al ridimensionamento e una nota su quale contenitore si adatta a ogni regione
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:

- Hai dato alla regione dei dettagli FillWeight 1 ma nessun MinimumSize, e a 700 pixel è sparita. Spiega che cosa ha fatto il motore flex con lo spazio e perché la correzione è un vincolo di dimensione e non un peso diverso.
- Il pulsante Apply va a capo sulla seconda riga a 1024 pixel anche se ci sarebbe spazio. Quale proprietà estesa lo ha causato, e come si comporterebbe diversamente la FilterBar se la togliessi e impostassi WrapContents a false?
- Un Margin di 8 sulle card nella FlowLayoutPanel ha prodotto degli spazi, ma lo stesso Margin sulla barra di navigazione in dock non ha avuto effetto. Quale motore rispetta i

margin in each case, and which property creates the space for the navigation bar?

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:

- ✓ Hai dato alla regione dei dettagli FillWeight 1 ma nessun MinimumSize, e a 700 pixel è sparita. Spiega che cosa ha fatto il motore flex con lo spazio e perché la correzione è un vincolo di dimensione e non un peso diverso.
- ✓ Il pulsante Apply va a capo sulla seconda riga a 1024 pixel anche se ci sarebbe spazio. Quale proprietà estesa lo ha causato, e come si comporterebbe diversamente la FilterBar se la togliessi e impostassi WrapContents a false?
- ✓ Un Margin di 8 sulle card nella FlowLayoutPanel ha prodotto degli spazi, ma lo stesso Margin sulla barra di navigazione in dock non ha avuto effetto. Quale motore rispetta i margini in ciascun caso, e quale proprietà crea lo spazio per la barra di navigazione?

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 5 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 5 su GitHub](#)

Oppure clona tutto con `git clone https://github.com/iceteagroup/LearnWisej-Samples` e apri `Theming and Responsive Layouts Course/Module 5`.



### 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 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."*
- ✓ **Dimostra che funziona davvero.** *"Passa in rassegna questo progetto deliverable per deliverable rispetto alla specifica del modulo su <https://learnwisej.com/assets/courses/theming-and-responsive-layouts/specs/module-05.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.