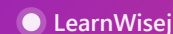


Esercizio di coding con l'IA

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.



L'assistente è la coppia di mani veloci. L'architetto resti tu.

Un assistente può produrre il progetto di questo modulo in pochi minuti. Quello che non può fare è decidere se il risultato è il progetto giusto per la tua applicazione: quella scelta resta tua. Il giudizio lo metti tu: istruiscilo per bene, poi verifica quello che ti torna indietro rispetto a ciò che il Modulo 7 ti ha appena insegnato.

Che cosa stai chiedendo

Completa il capstone di Adaptive Operations Console: un tema personalizzato con token con nome e almeno quattro appearance semantiche, un uso circoscritto di CssClass, uno shell desktop con regioni Flow, Table e Flex, ClientProfiles.json con comportamento per telefono, tablet, small desktop e desktop, proprietà responsive più un gestore ResponsiveProfileChanged idempotente, e un indicatore del profilo visibile in modalità di test. Esegui la revisione di accessibilità per focus frame, stati non validi, contrasto, aree di tocco e ordine di tabulazione, compila la matrice di QA responsive con screenshot per ogni profilo, e scrivi la nota di architettura e il grounding pack che spiegano ogni decisione su tema, layout e profili.

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

- ✓ Adaptive Operations Console completa con tema personalizzato, appearance semantiche, CSS circoscritto e regioni Flow, Table e Flex
- ✓ ClientProfiles.json, proprietà responsive e un gestore ResponsiveProfileChanged idempotente con un indicatore del profilo in modalità di test
- ✓ Revisione di accessibilità che copre focus frame, stati non validi, contrasto, aree di tocco e ordine di tabulazione
- ✓ Matrice di QA responsive con screenshot desktop, tablet e telefono in entrambi gli orientamenti
- ✓ Nota di architettura e grounding pack per l'IA che spiegano ogni decisione su tema, layout e profili

Perché è fatto così

Sei moduli di lavoro vivono ora in un'unica soluzione: un tema derivato da un tema di base, CSS circoscritto, uno shell di contenitori annidati, regioni Flow, Table e Flex, un ClientProfiles.json e un gestore del profilo. Ciò che trasforma tutto questo in un prodotto non è un'altra funzionalità. È la prova che la console è utilizzabile su ogni profilo, che un utente da tastiera o touch può usarla, che un

ridimensionamento non scatena ricostruzioni costose, e che il prossimo sviluppatore, o il prossimo aggiornamento di Wisej.NET, non può rompere in silenzio il sistema visivo. Questo modulo è quella prova, più il grounding pack che permette a un assistente IA di rispondere a domande sulle decisioni che hai preso.

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
set di schermate del capstone	Adaptive Operations Console completa: tema personalizzato, appearance semantiche, CSS circoscritto, uno shell desktop, uno shell tablet adattivo, uno shell per telefono, una griglia, un editor, feedback di validazione, navigazione e un indicatore del profilo visibile in modalità di test.
revisione dell'accessibilità visiva	Un passaggio su focus frame, stati disabilitati e non validi, contrasto, spaziatura per il tocco, tooltip sui pulsanti solo icona e ordine di tabulazione da tastiera, fatto usando davvero la schermata invece di leggere il tema.
gestore di profilo idempotente	Un gestore ResponsiveProfileChanged che assegna valori finali per il profilo corrente, così eseguirlo due volte, o dopo un ricaricamento, produce esattamente la stessa UI.
stabilità del layout	MinimumSize e MaximumSize impediscono alle regioni di collassare, e AutoSize resta fuori dalle grandi gerarchie di contenitori, così i ridimensionamenti sono prevedibili e leggeri.
governance del tema	Il JSON del tema, il foglio di stile circoscritto e ClientProfiles.json sono nel controllo del codice sorgente, revisionati come diff, documentati come token e mai modificati dentro un tema integrato.
personalizzazione sicura rispetto agli aggiornamenti	Appearance documentate, flussi del Theme Builder e proprietà lato server sopravvivono agli aggiornamenti; i selettori CSS che dipendono dal DOM interno di un widget potrebbero non farlo, quindi vanno circoscritti, documentati o evitati.
matrice di QA responsive e grounding pack	La matrice registra larghezze, orientamenti, profilo atteso e controlli per scenario; il grounding pack registra l'albero decisionale, le regole di risposta e la mappa delle fonti da cui risponde un assistente.

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 7. Puoi aprire tu stesso [module-07.md](#) per vedere esattamente che cosa leggerà.



Usa ChatGPT



Usa Claude

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.

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 7 - "Capstone pronto per il mobile, accessibilità, prestazioni e governance di produzione" - 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-07.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: Completa il capstone di Adaptive Operations Console: un tema personalizzato con token con nome e almeno quattro appearance semantiche, un uso circoscritto di `CssClass`, uno shell desktop con regioni `Flow`, `Table` e `Flex`, `ClientProfiles.json` con comportamento per telefono, tablet, small desktop e desktop, proprietà responsive più un gestore `ResponsiveProfileChanged` idempotente, e un indicatore del profilo visibile in modalità di test. Esegui la revisione di accessibilità per focus frame, stati non validi, contrasto, aree di tocco e ordine di tabulazione, compila la matrice di QA responsive con screenshot per ogni profilo, e scrivi la nota di architettura e il grounding pack che spiegano ogni decisione su tema, layout e profili.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Adaptive Operations Console completa con tema personalizzato, appearance semantiche, CSS circoscritto e regioni `Flow`, `Table` e `Flex`
 - `ClientProfiles.json`, proprietà responsive e un gestore `ResponsiveProfileChanged` idempotente con un indicatore del profilo in modalità di test
 - Revisione di accessibilità che copre focus frame, stati non validi, contrasto, aree di tocco e ordine di tabulazione
 - Matrice di QA responsive con screenshot desktop, tablet e telefono in entrambi gli orientamenti
 - Nota di architettura e grounding pack per l'IA che spiegano ogni decisione su tema, layout e profili
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:

- Il tuo gestore `ResponsiveProfileChanged` viene eseguito due volte per il profilo Tablet dopo un ricaricamento del browser. Quale tua riga garantisce che la seconda esecuzione non cambi nulla, e che cosa vedrebbe l'utente se quella riga mancasse?
- Un revisore ti chiede di rimuovere il focus frame da `action-button` perché "sembra datato". Che cosa cambi invece, quale token lo governa, e quali altri due stati ricontrolli poi nel Theme Builder?

- L'intestazione della griglia dei ticket ha bisogno di un ritocco che solo un selettore sul DOM interno del widget può raggiungere. Che cosa provi per prima cosa, che cosa deve essere vero prima che il selettore sia ammesso, e che cosa metti per iscritto così che il prossimo aggiornamento non sorprenda il team?

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:

- ✓ Il tuo gestore ResponsiveProfileChanged viene eseguito due volte per il profilo Tablet dopo un ricaricamento del browser. Quale tua riga garantisce che la seconda esecuzione non cambi nulla, e che cosa vedrebbe l'utente se quella riga mancasse?
- ✓ Un revisore ti chiede di rimuovere il focus frame da action-button perché "sembra datato". Che cosa cambi invece, quale token lo governa, e quali altri due stati ricontrolli poi nel Theme Builder?
- ✓ L'intestazione della griglia dei ticket ha bisogno di un ritocco che solo un selettore sul DOM interno del widget può raggiungere. Che cosa provi per prima cosa, che cosa deve essere vero prima che il selettore sia ammesso, e che cosa metti per iscritto così che il prossimo aggiornamento non sorprenda il team?

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 7 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 7 su GitHub](#)

Oppure clona tutto con `git clone https://github.com/iceteagroup/LearnWisej-Samples` e apri `Theming and Responsive Layouts Course/Module 7`.



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.

- ✓ **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-07.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."*
- ✓ **Rivedilo come faresti con una pull request.** *"Accompagnami attraverso questo codice AdaptiveOps come se stessi rivedendo la pull request di un collega e non l'avessi scritto tu. Dove vive lo stato, che cosa succede dall'inizio alla fine su una singola azione dell'utente, che cosa rimanderesti indietro per modifiche, e quali parti non ti sentiresti tranquillo a mettere davanti a un cliente? Sii specifico e cita le righe di cui parli — non riassumermi il progetto."*
- ✓ **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."*



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.