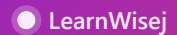


Esercizio di coding con l'IA

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.



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

Aggiungi ad Adaptive Operations Console un ClientProfiles.json con i profili Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop e Desktop ordinati dal più stretto al più largo. Nel designer, usa le proprietà responsive per nascondere la barra di navigazione e il pannello dei dettagli sul telefono, passare i pulsanti della toolbar a un Display solo icona e rimettere in dock la regione dei dettagli sotto la griglia sul tablet. Gestisci ResponsiveProfileChanged per mostrare il nome del profilo attivo nella label di stato, registrare ogni cambio nel log e aprire l'editor dei dettagli come form modale sul telefono invece che come pannello laterale.

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

- ✓ ClientProfiles.json con i profili Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop e Desktop in ordine di prima corrispondenza
- ✓ Valori delle proprietà responsive assegnati nel designer per nascondere la barra di navigazione e il pannello dei dettagli sul telefono
- ✓ Display solo icona della toolbar e regione dei dettagli rimessa in dock per il tablet, impostati tramite proprietà responsive
- ✓ Gestore ResponsiveProfileChanged che mostra e registra nel log il nome del profilo attivo
- ✓ Editor dei dettagli aperto come form modale sul telefono tramite il gestore del profilo

Perché è fatto così

Adaptive Operations Console ha ora uno shell con tema costruito con contenitori Flow, Table e Flex che regge un ridimensionamento senza un gestore Resize, ma mostra a tutti la stessa schermata: su un telefono da 390 pixel la barra di navigazione, la griglia dei ticket e l'editor dei dettagli si contendono la stessa larghezza. Le media query CSS non possono risolverlo, perché i controlli vivono sul server: nascondere un pannello, passare un pulsante a solo icona o rimettere in dock una regione è una modifica di proprietà lato server, e un utente su telefono che perde il pannello dei dettagli ha bisogno di un altro

modo per modificare un ticket, non di una copia rimpicciolita di quello desktop. Questo modulo aggiunge il meccanismo che Wisej.NET possiede per questo: profili client, proprietà responsive e l'evento ResponsiveProfileChanged.

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
ClientProfile	Un insieme con nome di regole di corrispondenza legate al dispositivo (larghezza del browser, larghezza dello schermo, tipo di dispositivo, user agent, orientamento orizzontale) che descrive come viene usata una classe di client, non il nome commerciale di un dispositivo.
ClientProfiles.json	Il file che Wisej.NET legge; i profili vengono valutati dall'alto verso il basso e vince il primo che corrisponde, quindi l'ordine delle voci fa parte della configurazione.
Application.ActiveProfile	Il profilo attualmente selezionato per questa sessione; il suo Name è ciò che mostra la label di stato e ciò su cui si ramifica il gestore del profilo.
proprietà responsive	Una normale proprietà.NET marcata con ResponsivePropertyAttribute che contiene un valore per ogni profilo client e applica il valore corrispondente ogni volta che cambia il profilo attivo.
adattamento guidato dal designer	I valori responsive si assegnano solo nel designer di Visual Studio, il che li mantiene serializzati, visibili e deterministici; il codice serve per gli adattamenti che dipendono dai dati o dallo stato a runtime.
ResponsiveProfileChanged	L'evento sollevato su Application, su contenitori come Form, Page, Desktop e UserControl e su DataGridView e ListView quando cambia il profilo attivo; ResponsiveProfileChangedEventArgs.CurrentProfile porta il nuovo profilo.
matrice di QA responsive	Il piano di test che elenca larghezza, altezza, orientamento, tipo di dispositivo, tocco, barre di scorrimento, scala dei font e il nome del profilo che ti aspetti a ogni dimensione.

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



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 6 - "ClientProfile e proprietà responsive" - 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-06.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: Aggiungi ad Adaptive Operations Console un `ClientProfiles.json` con i profili Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop e Desktop ordinati dal più stretto al più largo. Nel designer, usa le proprietà responsive per nascondere la barra di navigazione e il pannello dei dettagli sul telefono, passare i pulsanti della toolbar a un Display solo icona e rimettere in dock la regione dei dettagli sotto la griglia sul tablet. Gestisci `ResponsiveProfileChanged` per mostrare il nome del profilo attivo nella label di stato, registrare ogni cambio nel log e aprire l'editor dei dettagli come form modale sul telefono invece che come pannello laterale.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - `ClientProfiles.json` con i profili Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop e Desktop in ordine di prima corrispondenza
 - Valori delle proprietà responsive assegnati nel designer per nascondere la barra di navigazione e il pannello dei dettagli sul telefono
 - Display solo icona della toolbar e regione dei dettagli rimessa in dock per il tablet, impostati tramite proprietà responsive
 - Gestore `ResponsiveProfileChanged` che mostra e registra nel log il nome del profilo attivo
 - Editor dei dettagli aperto come form modale sul telefono tramite il gestore del profilo
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 Small Desktop venisse spostato in cima a `ClientProfiles.json`, quali degli altri cinque profili potrebbero ancora essere individuati, e che cosa vedrebbe un tablet in orizzontale?
- Il nuovo docking del tablet vive nel designer e l'editor modale del telefono vive nel codice. Che cosa decide quale meccanismo possiede ciascuna modifica, e che cosa andrebbe storto se li scambiassi?
- Un tester ha ruotato un telefono quattro volte e il tuo gestore è stato eseguito quattro volte. Quale stato deve essere identico dopo la quarta esecuzione e dopo la prima, e come lo

garantisce il tuo codice?

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 Small Desktop venisse spostato in cima a ClientProfiles.json, quali degli altri cinque profili potrebbero ancora essere individuati, e che cosa vedrebbe un tablet in orizzontale?
- ✓ Il nuovo docking del tablet vive nel designer e l'editor modale del telefono vive nel codice. Che cosa decide quale meccanismo possiede ciascuna modifica, e che cosa andrebbe storto se li scambiassi?
- ✓ Un tester ha ruotato un telefono quattro volte e il tuo gestore è stato eseguito quattro volte. Quale stato deve essere identico dopo la quarta esecuzione e dopo la prima, e come lo garantisce il tuo codice?

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 `Theming and Responsive Layouts 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.

- ✓ **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-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."*
- ✓ **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."*



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.