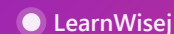


## Esercizio di coding con l'IA

Costruisci il progetto EnterpriseOps 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

Crea un Escalation Wizard che raccoglie motivo, allegati, approvatore, data di scadenza e opzioni di notifica. Implementa l'orchestrazione in un servizio, non nelle pagine del wizard, e aggiungi la compensazione per un errore di notifica simulato.

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

- ✓ Flusso delle schermate del wizard
- ✓ Servizio di workflow
- ✓ Oggetti comando e risultato tipizzati
- ✓ Matrice dei percorsi di errore
- ✓ Esempio di compensazione

### Perché è fatto così

I flussi di lavoro enterprise raramente stanno in un solo pulsante Save. Le schermate avanzate coordinano approvazioni, assegnazioni, validazioni, upload e verifiche esterne preservando la fiducia dell'utente.

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                                                                                                                                                                    |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| orchestrazione dei flussi di lavoro | Un servizio che possiede la sequenza, le regole e lo stato di un processo di business a più passi, così il flusso non resta sepolto in diversi form.                                      |
| stato del wizard                    | Le risposte accumulate e lo stato di validazione di un wizard a più passi, tenuti in un unico oggetto tipizzato invece che sparsi tra i controlli.                                        |
| risultato tipizzato                 | Un valore di ritorno che modella esplicitamente gli esiti di successo, errore di validazione ed errore, così i chiamanti devono gestire ogni percorso invece di affidarsi alle eccezioni. |
| comando di compensazione            | Un'operazione che annulla gli effetti di un passo precedente quando un passo successivo fallisce, mantenendo il sistema coerente senza un'unica gigantesca transazione.                   |
| flusso di lavoro riprendibile       | Un processo che rende persistente il proprio avanzamento, così un utente può chiudere il browser e continuare più tardi, o un errore può essere recuperato dall'ultimo passo riuscito.    |
| matrice dei percorsi di errore      | Una tabella che elenca ogni passo, che cosa può andare storto lì e come risponde il flusso di lavoro, così la gestione degli errori è progettata e non improvvisata.                      |

## 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

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 EnterpriseOps 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 - "UX avanzata dei flussi: procedure guidate, orchestrazione modale e compensazione" - del corso LearnWisej "Wisej.NET enterprise: dall'architettura al cloud", 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/azure-docker/specs/module-07.md>

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

1. Progetto "EnterpriseOps", 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: Crea un Escalation Wizard che raccoglie motivo, allegati, approvatore, data di scadenza e opzioni di notifica. Implementa l'orchestrazione in un servizio, non nelle pagine del wizard, e aggiungi la compensazione per un errore di notifica simulato.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
  - Flusso delle schermate del wizard
  - Servizio di workflow
  - Oggetti comando e risultato tipizzati
  - Matrice dei percorsi di errore
  - Esempio di compensazione
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 flusso di lavoro si può testare senza il wizard?
- Quale stato sopravvive a un aggiornamento del browser?
- Che cosa succede se la notifica riesce ma la scrittura del log di audit fallisce?

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 flusso di lavoro si può testare senza il wizard?
- ✓ Quale stato sopravvive a un aggiornamento del browser?
- ✓ Che cosa succede se la notifica riesce ma la scrittura del log di audit fallisce?

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 `Enterprise Wisej.NET 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.

- ✓ **Sostituisci un pezzo senza spostare le giunzioni.** *"Scegli la parte di questo progetto EnterpriseOps 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 EnterpriseOps 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 EnterpriseOps 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.