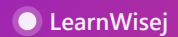


Esercizio di coding con l'IA

Costruisci il progetto GlobalDesk 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.



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 2 ti ha appena insegnato.

Che cosa stai chiedendo

Localizza l'editor cliente di GlobalDesk tramite il designer di Visual Studio. Completa prima il layout neutro in inglese dello user control CustomerEditor, poi imposta Localizable a true e usa il selettore Language del designer per aggiungere una variante tedesca, cambiando solo ciò che deve essere diverso: le etichette, la larghezza dei due pulsanti più lunghi e la Location di una label. Esegui l'applicazione sotto de-DE e confronta il risultato reale nel browser con la vista del designer, poi aggiungi un pulsante Recreate editor che distrugge e ricostruisce il controllo, così da vedere con i tuoi occhi perché le risorse del designer non seguono un cambio di cultura a runtime. Concludi con una nota di lab che elenca le proprietà che hai sovrascritto e quelle che hai deliberatamente lasciato al layout neutro.

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

- ✓ User control CustomerEditor con Localizable impostato a true sopra un layout neutro inglese completo
- ✓ Variante di lingua tedesca nel designer che sovrascrive le etichette più almeno una Size e una Location
- ✓ Ancoraggio, docking o AutoSize applicati in modo che le etichette tedesche non possano troncarsi
- ✓ Pulsante Recreate editor che distrugge e ricostruisce il contenitore dopo un cambio di cultura
- ✓ Nota di lab che elenca le proprietà sovrascritte e quelle lasciate al layout neutro

Perché è fatto così

Il Modulo 1 ha dato a GlobalDesk una dashboard le cui etichette arrivano tutte da Strings.resx, ed è quella la casa giusta per il testo che il codice genera. L'editor cliente è diverso: è una schermata disegnata, sistemata a occhio nel designer di Visual Studio. Far passare ognuna delle sue etichette da una chiamata a un helper significa modificare a mano il file del designer, e non risolve comunque il problema vero, cioè che un Kundennummer speichern tedesco non ci sta dentro un pulsante dimensionato per Save. Questo

modulo usa il flusso di localizzazione che Visual Studio offre già, così che le etichette e la geometria di una schermata disegnata possano cambiare da una lingua all'altra senza stringhe scritte nel codice e senza un secondo file di layout.

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
Localizable	Una proprietà a design time su un contenitore di primo livello; true porta il designer nel flusso di localizzazione, così i valori possono essere memorizzati per cultura.
Selettore Language del designer	Compare appena Localizable è true; la cultura selezionata è la versione che stai modificando, quindi ogni cambiamento viene registrato solo per quella cultura.
La lingua neutra	La versione predefinita su cui ricade ogni cultura, e l'unica in cui aggiungi, rimuovi o ristrutturati i controlli; le viste di cultura contengono sovrascritture, non una seconda schermata.
Proprietà localizzabili	Non solo Text: Size, Location, colori e immagini possono avere ciascuno un valore diverso per cultura, perché la traduzione cambia la geometria oltre alle parole.
File di risorse generati	Il designer scrive i valori localizzati in file di risorse invece che nel codice C# del designer, tenendo i valori specifici di una lingua fuori dal codice che leggi e fondi ogni giorno.
Applicati al momento della creazione	I valori localizzati nel designer vengono applicati mentre i controlli vengono creati, quindi cambiare più tardi la cultura della sessione non riscrive un contenitore che esiste già.
Layout tollerante alla traduzione	Docking, ancoraggio, FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel, AutoSize e padding lasciano crescere il testo da solo, così servono meno sovrascritture di geometria.

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 2. Puoi aprire tu stesso [module-02.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 GlobalDesk 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 2 - "Localizzare l'interfaccia Wisej.NET nel designer di Visual Studio" - del corso LearnWisej "Localizzazione in Wisej.NET", 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/localization-in-wisej-net/specs/module-02.md>

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

1. Progetto "GlobalDesk", 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: Localizza l'editor cliente di GlobalDesk tramite il designer di Visual Studio. Completa prima il layout neutro in inglese dello user control CustomerEditor, poi imposta Localizable a true e usa il selettore Language del designer per aggiungere una variante tedesca, cambiando solo ciò che deve essere diverso: le etichette, la larghezza dei due pulsanti più lunghi e la Location di una label. Esegui l'applicazione sotto de-DE e confronta il risultato reale nel browser con la vista del designer, poi aggiungi un pulsante Recreate editor che distrugge e ricostruisce il controllo, così da vedere con i tuoi occhi perché le risorse del designer non seguono un cambio di cultura a runtime. Concludi con una nota di lab che elenca le proprietà che hai sovrascritto e quelle che hai deliberatamente lasciato al layout neutro.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - User control CustomerEditor con Localizable impostato a true sopra un layout neutro inglese completo
 - Variante di lingua tedesca nel designer che sovrascrive le etichette più almeno una Size e una Location
 - Ancoraggio, docking o AutoSize applicati in modo che le etichette tedesche non possano troncarsi
 - Pulsante Recreate editor che distrugge e ricostruisce il contenitore dopo un cambio di cultura
 - Nota di lab che elenca le proprietà sovrascritte e quelle lasciate al layout neutro
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:

- Un collega aggiunge un nuovo campo telefono a CustomerEditor mentre nell'elenco Language è selezionato il tedesco. Che cosa vedrà probabilmente l'utente inglese, e qual è la regola che l'avrebbe evitato?
- La conferma di salvataggio che l'editor mostra dopo un aggiornamento riuscito continua ad arrivare da Strings.resx invece che dalle risorse del designer. Dai due ragioni per cui è quello il posto giusto, e indica un'etichetta di questa schermata che invece appartiene

davvero al designer.

– Il tuo pulsante Recreate editor funziona, ma un collega sostiene che la schermata dovrebbe semplicemente riapplicare le proprietà localizzate invece di ricostruirsi. Che cosa dovrebbe sapere quel codice che la ricostruzione non ha bisogno di sapere, e quando il suo approccio sarebbe il compromesso migliore?

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:

- ✓ Un collega aggiunge un nuovo campo telefono a CustomerEditor mentre nell'elenco Language è selezionato il tedesco. Che cosa vedrà probabilmente l'utente inglese, e qual è la regola che l'avrebbe evitato?
- ✓ La conferma di salvataggio che l'editor mostra dopo un aggiornamento riuscito continua ad arrivare da Strings.resx invece che dalle risorse del designer. Dai due ragioni per cui è quello il posto giusto, e indica un'etichetta di questa schermata che invece appartiene davvero al designer.
- ✓ Il tuo pulsante Recreate editor funziona, ma un collega sostiene che la schermata dovrebbe semplicemente riapplicare le proprietà localizzate invece di ricostruirsi. Che cosa dovrebbe sapere quel codice che la ricostruzione non ha bisogno di sapere, e quando il suo approccio sarebbe il compromesso migliore?

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.



Per questo corso non c'è ancora un esempio di riferimento — il tuo è il primo

Il [repository degli esempi LearnWisej](https://learnwisej.com/assets/courses/localization-in-wisej-net/specs/module-02.md) contiene build già fatte per quasi tutti i corsi, ma non ancora per questo. Significa che non c'è niente con cui confrontarsi e nessuna scorciatoia: la specifica e la tua revisione sono tutta la rete di sicurezza. Tieni quello che costruisci — e se viene bene, ci farebbe piacere vederlo.

Tre prompt per andare oltre

Punta il tuo assistente sul progetto che hai appena costruito 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/localization-in-wisej-net/specs/module-02.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 GlobalDesk 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 GlobalDesk 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.