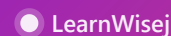


Esercizio di coding con l'IA

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.



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

Crea la pagina dashboard di GlobalDesk con un'intestazione di benvenuto, un pulsante di navigazione Customers e un piccolo pannello di anteprima della cultura. Aggiungi Resources/Strings.resx come risorsa neutra con almeno dodici chiavi semantiche stabili come Navigation.Customers e CustomerEditor.Save, scrivi un helper statico Texts sopra Wisej.Resources.ResourceManager che restituisce la chiave tra parentesi quadre quando manca, e sostituisci ogni etichetta scritta direttamente nel codice con una chiamata a Texts.Get. Poi riempi il pannello di anteprima con valori veri: formatta la data di oggi, una quantità e un importo con Application.CurrentCulture invece di comporre le stringhe a mano, e concludi con una breve nota di lab che separa il valore memorizzato dalla sua presentazione sensibile alla cultura.

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

- ✓ Pagina dashboard di GlobalDesk le cui etichette provengono tutte da chiavi di risorsa, senza testo visibile all'utente scritto direttamente nel codice
- ✓ File neutro Resources/Strings.resx con almeno dodici chiavi semantiche stabili
- ✓ Helper Texts sopra Wisej.Resources.ResourceManager che segnala una chiave mancante invece di restituire una stringa vuota
- ✓ Pannello di anteprima della cultura che formatta una data, un numero e un importo con Application.CurrentCulture
- ✓ Nota di lab che spiega quali valori sono dati e quali sono presentazione sensibile alla cultura

Perché è fatto così

GlobalDesk è un piccolo service desk scritto per un solo ufficio. Ogni intestazione, etichetta di pulsante e messaggio di conferma è un letterale inglese digitato direttamente nel C#, e la dashboard stampa importi e date come capita di farlo a una macchina americana. Quando apre una filiale tedesca, quel testo non ha nessun posto dove andare — e chi comincia a correggere le etichette a mano incontra la seconda metà del problema: un'etichetta tedesca sopra un importo che recita ancora 1,850.75 non è una schermata tedesca.

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
localizzazione	Adattare un'applicazione alla lingua e alla cultura dell'utente: testo tradotto, date, numeri e importi formattati correttamente, spazio per parole più lunghe, e un layout che sappia rispecchiarsi per le lingue che si scrivono da destra a sinistra.
nome di cultura	Un identificatore come en, de-DE o fr-CA che indica una lingua e, facoltativamente, una regione; la parte della lingua decide quali risorse vengono preferite, il nome completo guida le convenzioni di formattazione.
risorsa neutra	L'insieme completo di risorse su cui si ricade, contenuto in un file come Resources/Strings.resx senza cultura nel nome; ogni chiave che l'applicazione chiede deve esistere qui.
Wisej.Resources.ResourceManager	L'oggetto che prende un nome base di risorse e un assembly e restituisce il valore di una chiave nella cultura che corrisponde meglio, così che il tuo codice non apra mai un file di persona.
fallback di cultura	La regola di.NET per trovare un valore: prova la cultura completa, poi la cultura solo lingua, poi la risorsa neutra, e usa la prima che contiene la chiave.
chiave di risorsa semantica	Un nome stabile che descrive il significato invece della formulazione, come CustomerEditor.Save o Validation.Required, così che il testo inglese possa essere riscritto senza toccare nessun'altra lingua.
Application.CurrentCulture	La cultura della sessione Wisej.NET corrente; due utenti collegati nello stesso momento possono avere due valori diversi, ed è per questo che la cultura non deve mai vivere in un campo statico.

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 1. Puoi aprire tu stesso [module-01.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 1 - "Fondamenti di localizzazione: cultura, lingua e risorse" - 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-01.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: Crea la pagina dashboard di GlobalDesk con un'intestazione di benvenuto, un pulsante di navigazione Customers e un piccolo pannello di anteprima della cultura. Aggiungi `Resources/Strings.resx` come risorsa neutra con almeno dodici chiavi semantiche stabili come `Navigation.Customers` e `CustomerEditor.Save`, scrivi un helper statico `Texts` sopra `Wisej.Resources.ResourceManager` che restituisce la chiave tra parentesi quadre quando manca, e sostituisci ogni etichetta scritta direttamente nel codice con una chiamata a `Texts.Get`. Poi riempi il pannello di anteprima con valori veri: formatta la data di oggi, una quantità e un importo con `Application.CurrentCulture` invece di comporre le stringhe a mano, e concludi con una breve nota di lab che separa il valore memorizzato dalla sua presentazione sensibile alla cultura.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Pagina dashboard di GlobalDesk le cui etichette provengono tutte da chiavi di risorsa, senza testo visibile all'utente scritto direttamente nel codice
 - File neutro `Resources/Strings.resx` con almeno dodici chiavi semantiche stabili
 - Helper `Texts` sopra `Wisej.Resources.ResourceManager` che segnala una chiave mancante invece di restituire una stringa vuota
 - Pannello di anteprima della cultura che formatta una data, un numero e un importo con `Application.CurrentCulture`
 - Nota di lab che spiega quali valori sono dati e quali sono presentazione sensibile alla cultura
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:

- Cambi la formulazione inglese del pulsante Save da "Save" a "Save customer". Quali file devono cambiare se la tua chiave è `CustomerEditor.Save`, e quali avrebbero dovuto cambiare se la chiave fosse stata la frase inglese stessa?
- Il tuo helper `Texts` restituisce `[Preview.Amount]` per una label. Indica due errori diversi che producono esattamente quel marcatore, e di' come li distingueresti senza un debugger.
- Un collega propone di memorizzare il totale dell'ordine nel database come la stringa

"1.850,75 €" così che gli utenti tedeschi lo vedano sempre correttamente. Spiega che cosa si rompe, e quale parte di quella stringa è dato e quale è presentazione.

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:

- ✓ Cambi la formulazione inglese del pulsante Save da "Save" a "Save customer". Quali file devono cambiare se la tua chiave è CustomerEditor.Save, e quali avrebbero dovuto cambiare se la chiave fosse stata la frase inglese stessa?
- ✓ Il tuo helper Texts restituisce [Preview.Amount] per una label. Indica due errori diversi che producono esattamente quel marcatore, e di' come li distingueresti senza un debugger.
- ✓ Un collega propone di memorizzare il totale dell'ordine nel database come la stringa "1.850,75 €" così che gli utenti tedeschi lo vedano sempre correttamente. Spiega che cosa si rompe, e quale parte di quella stringa è dato e quale è presentazione.

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](#) 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.

- ✓ **Aggiungi il secondo caso.** *"Estendi questo progetto GlobalDesk 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/localization-in-wisej-net/specs/module-01.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."*



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.