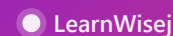


Esercizio di coding con l'IA

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

Esegui un giro di traduzione onesto su GlobalDesk. Apri la soluzione in ResX Resource Manager, aggiungi l'italiano come nuova colonna lingua e usa il filtro delle voci non tradotte per elencare ogni chiave ancora senza valore. Aggiungi commenti alle tre chiavi il cui significato è ambiguo fuori contesto, a partire dall'etichetta di una sola parola Open, poi traduci o fai tradurre in bozza dalla macchina i valori italiani e marca come invarianti il nome del prodotto e i due identificatori tecnici, così che nessuno li traduca. Esporta la colonna neutra e quella italiana, reimportale e rivedi il diff del .resx risultante prima di committarlo come un'unica modifica rivedibile. Concludi con una nota di una pagina sul flusso di lavoro che confronta l'editor di Visual Studio, ResX Resource Manager e una piattaforma cloud per un team i cui traduttori non sono sviluppatori.

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

- ✓ Strings.it.resx prodotto con ResX Resource Manager senza nessuna chiave non tradotta
- ✓ Commenti di risorsa su almeno tre chiavi ambigue perché un traduttore abbia il contesto
- ✓ Marcatori di invarianza sul nome del prodotto e sugli identificatori che non devono essere tradotti
- ✓ Esportazione, reimportazione e un diff.resx rivisto e committato come un'unica modifica
- ✓ Nota sul flusso di lavoro che confronta l'editor di Visual Studio, ResX Resource Manager e un TMS cloud

Perché è fatto così

GlobalDesk porta ormai un Resources/Strings.resx neutro, uno Strings.de.resx tedesco, le risorse del designer sull'editor cliente e un Resources-de.resx per le etichette di sistema di Wisej. Aggiungere l'italiano con l'editor di Visual Studio significa aprire quei file una scheda per volta e confrontare a occhio gli elenchi delle chiavi, e la persona che parla italiano di solito non è la persona che sa aprire la soluzione. Questo

modulo trasforma la traduzione in un flusso di lavoro con un elenco visibile delle lacune, una consegna e un passaggio di revisione.

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
Vista a matrice	Le chiavi sulle righe e le lingue sulle colonne, così un valore italiano mancante e un termine tedesco che è andato alla deriva si vedono entrambi in un colpo solo.
ResX Resource Manager	Uno strumento di localizzazione.NET open source, estensione di Visual Studio e applicazione standalone, che mostra le risorse RESX di una soluzione in un'unica griglia centrale.
Colonna lingua	Aggiungere una lingua di destinazione crea il file specifico per cultura accanto a quello neutro e ti dà una colonna vuota da riempire; le chiavi non cambiano mai.
Filtro delle voci non tradotte	Elenca solo le voci senza valore nella lingua selezionata, così "abbiamo finito?" diventa un numero invece che un'opinione.
Commento della risorsa	Una nota conservata accanto al valore che dice a un traduttore se Open è un verbo, un aggettivo, uno stato del ticket o un comando.
Stringa invariante	Un marcatore sulle voci che non devono mai essere tradotte, come il nome del prodotto e gli identificatori tecnici.
Giro di esportazione / importazione	Colonne inviate a un traduttore o a una piattaforma che gestisce il formato RESX e restituite come file.resx specifici per cultura che la build tratta come qualsiasi altra risorsa.

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 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 6 - "Flusso di lavoro della traduzione: ResX Resource Manager e altri strumenti" - 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-06.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: Esegui un giro di traduzione onesto su GlobalDesk. Apri la soluzione in ResX Resource Manager, aggiungi l'italiano come nuova colonna lingua e usa il filtro delle voci non tradotte per elencare ogni chiave ancora senza valore. Aggiungi commenti alle tre chiavi il cui significato è ambiguo fuori contesto, a partire dall'etichetta di una sola parola Open, poi traduci o fai tradurre in bozza dalla macchina i valori italiani e marca come invariati il nome del prodotto e i due identificatori tecnici, così che nessuno li traduca. Esporta la colonna neutra e quella italiana, reimportale e rivedi il diff del .resx risultante prima di committarlo come un'unica modifica rivedibile. Concludi con una nota di una pagina sul flusso di lavoro che confronta l'editor di Visual Studio, ResX Resource Manager e una piattaforma cloud per un team i cui traduttori non sono sviluppatori.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - `Strings.it.resx` prodotto con ResX Resource Manager senza nessuna chiave non tradotta
 - Commenti di risorsa su almeno tre chiavi ambigue perché un traduttore abbia il contesto
 - Marcatori di invarianza sul nome del prodotto e sugli identificatori che non devono essere tradotti
 - Esportazione, reimportazione e un diff.resx rivisto e committato come un'unica modifica
 - Nota sul flusso di lavoro che confronta l'editor di Visual Studio, ResX Resource Manager e un TMS cloud
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:

- La tua colonna italiana è piena e l'applicazione sembra corretta in it-IT, ma un collega sostiene che due chiavi non sono mai state tradotte. Come possono essere vere entrambe le affermazioni allo stesso tempo, e quale funzionalità dello strumento chiude la discussione?
- Una bozza automatica ha restituito Ultimo ordine: {1} il {0} per una chiave il cui valore neutro è Last order: {0} on {1}. Che cosa si rompe, quando si rompe, e che cosa l'avrebbe intercettato prima del commit?
- Il tuo team ha cinque lingue, due delle quali tradotte da un'agenzia che non ha mai aperto

Visual Studio. Quale dei tre flussi di lavoro adotteresti, e che cosa dovrebbe cambiare prima di passare a quello successivo?

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:

- ✓ La tua colonna italiana è piena e l'applicazione sembra corretta in it-IT, ma un collega sostiene che due chiavi non sono mai state tradotte. Come possono essere vere entrambe le affermazioni allo stesso tempo, e quale funzionalità dello strumento chiude la discussione?
- ✓ Una bozza automatica ha restituito Ultimo ordine: {1} il {0} per una chiave il cui valore neutro è Last order: {0} on {1}. Che cosa si rompe, quando si rompe, e che cosa l'avrebbe intercettato prima del commit?
- ✓ Il tuo team ha cinque lingue, due delle quali tradotte da un'agenzia che non ha mai aperto Visual Studio. Quale dei tre flussi di lavoro adotteresti, e che cosa dovrebbe cambiare prima di passare a quello successivo?

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*

net/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 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.