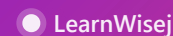


Esercizio di coding con l'IA

Costruisci il progetto IconDesk 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 il progetto IconDesk e dagli un pannello comandi con i pulsanti Open, Save, Print e Delete, ciascuno con la propria icona portata da un meccanismo diverso. Carica btnOpen da un System.Drawing.Image lato server, assegna a btnSave.ImageSource un'immagine del tema indicata per nome e fai puntare btnPrint e btnDelete a un'unica ImageList condivisa tramite ImageKey. Aggiungi una label diagnostica che ispeziona ogni pulsante attraverso Image e segnala quale meccanismo è davvero in gioco, poi cambia l'immagine memorizzata sotto una chiave della ImageList e verifica che tutti i controlli che usano quella chiave seguano il cambiamento.

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

- ✓ Pannello comandi di IconDesk con i pulsanti Open, Save, Print e Delete
- ✓ btnOpen che porta un System.Drawing.Image caricato sul server e btnSave un'immagine del tema indicata per nome su ImageSource
- ✓ btnPrint e btnDelete che condividono un unico ImageList tramite ImageKey invece che ImageIndex
- ✓ Label diagnostica che legge ogni pulsante attraverso Image e dà un nome al meccanismo dietro la sua icona
- ✓ Nota di lab che registra che cosa è successo quando è stata sostituita una chiave dell'ImageList e quando sullo stesso pulsante sono state impostate due proprietà immagine

Perché è fatto così

Wisej.NET offre di proposito più di un modo per mettere una grafica su un controllo, e funzionano tutti. I guai iniziano quando un team li usa tutti per caso: una toolbar in cui un pulsante legge un file dal disco del server, quello accanto punta al nome di un tema e un terzo ha perso silenziosamente la sua icona perché sullo stesso controllo sono state impostate due proprietà. Sei mesi dopo nessuno sa più spiegare da dove arrivi davvero una certa immagine né perché sia sparita dopo il deploy.

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
proprietà Image	Un oggetto System.Drawing.Image che vive nella memoria del server; Wisej.NET lo invia al browser come dati PNG, quindi l'immagine viaggia come byte che il tuo codice ha già decodificato.
stringa ImageSource	Una stringa che identifica una risorsa che il client dovrà risolvere: un URL relativo, un URL assoluto, un'immagine del tema indicata per nome, un percorso resource.wx o un elemento dentro l'assembly di un pacchetto di icone.
IconSource	La stessa idea di stringa sui controlli che possiedono uno slot icona dedicato accanto all'immagine principale: controlla l'API del controllo che stai usando invece di dare per scontato che tutti chiamino la proprietà allo stesso modo.
contratto IImage	La famiglia comune di proprietà immagine che i controlli con supporto alle immagini espongono — Image, ImageAlign, ImageIndex, ImageKey, ImageList e ImageSource — che permette a un unico metodo di supporto di decorare in modo uniforme un Button, una Label o qualunque altro controllo.
ImageList	Una collezione riutilizzabile di oggetti immagine a cui più controlli possono puntare, indirizzata per posizione o per nome, e in grado di fissare una dimensione comune delle immagini per i controlli che la usano.
ImageKey contro ImageIndex	Una chiave come "delete" porta un significato e sopravvive ai riordini; l'indice 3 diventa silenziosamente l'immagine sbagliata il giorno in cui qualcuno ne inserisce una più in alto.
Assegnazione mutuamente esclusiva	Tutte queste proprietà alimentano un unico slot visivo, quindi assegnarne una può azzerare le altre — ed è per questo che l'icona sbagliata di solito è una seconda assegnazione, non un file rotto.

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 IconDesk 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 - "Il modello delle immagini di Wisej.NET: Image, ImageSource e ImageList" - del corso LearnWisej "Icone e immagini 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/icons-and-images/specs/module-01.md>

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

1. Progetto "IconDesk", 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 il progetto IconDesk e dagli un pannello comandi con i pulsanti Open, Save, Print e Delete, ciascuno con la propria icona portata da un meccanismo diverso. Carica `btnOpen` da un `System.Drawing.Image` lato server, assegna a `btnSave.ImageSource` un'immagine del tema indicata per nome e fai puntare `btnPrint` e `btnDelete` a un'unica `ImageList` condivisa tramite `ImageKey`. Aggiungi una label diagnostica che ispeziona ogni pulsante attraverso `IImage` e segnala quale meccanismo è davvero in gioco, poi cambia l'immagine memorizzata sotto una chiave della `ImageList` e verifica che tutti i controlli che usano quella chiave seguano il cambiamento.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Pannello comandi di IconDesk con i pulsanti Open, Save, Print e Delete
 - `btnOpen` che porta un `System.Drawing.Image` caricato sul server e `btnSave` un'immagine del tema indicata per nome su `ImageSource`
 - `btnPrint` e `btnDelete` che condividono un unico `ImageList` tramite `ImageKey` invece che `ImageIndex`
 - Label diagnostica che legge ogni pulsante attraverso `IImage` e dà un nome al meccanismo dietro la sua icona
 - Nota di lab che registra che cosa è successo quando è stata sostituita una chiave dell'`ImageList` e quando sullo stesso pulsante sono state impostate due proprietà immagine
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 label diagnostica dice che `btnDelete` usa una sorgente immagine, ma tu hai impostato solo la sua `ImageKey`. Che cosa deve essere stato eseguito, e dove lo cercheresti?
- Sia `print` sia `delete` sono cambiate quando hai sostituito una voce dell' `ImageList`. Sarebbe ancora vero se i pulsanti usassero `ImageIndex`, e che cosa succederebbe se una quinta immagine venisse inserita in cima alla lista?
- Il pulsante `Open` legge il proprio file sul server. Quale parte di questa soluzione si rompe quando l'applicazione viene distribuita su un'altra macchina, e quale non si romperebbe se la stessa immagine fosse una stringa di sorgente immagine?

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 label diagnostica dice che btnDelete usa una sorgente immagine, ma tu hai impostato solo la sua ImageKey. Che cosa deve essere stato eseguito, e dove lo cercheresti?
- ✓ Sia print sia delete sono cambiate quando hai sostituito una voce dell' ImageList. Sarebbe ancora vero se i pulsanti usassero ImageIndex, e che cosa succederebbe se una quinta immagine venisse inserita in cima alla lista?
- ✓ Il pulsante Open legge il proprio file sul server. Quale parte di questa soluzione si rompe quando l'applicazione viene distribuita su un'altra macchina, e quale non si romperebbe se la stessa immagine fosse una stringa di sorgente immagine?

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.

- ✓ **Dimostra che funziona davvero.** *"Passa in rassegna questo progetto deliverable per deliverable rispetto alla specifica del modulo su <https://learnwisej.com/assets/courses/icons-and-images/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 IconDesk 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 IconDesk 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.