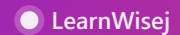


Esercizio di coding con l'IA

Costruisci il progetto VisualOperationsStudio 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

Aggiungi a VisualOperationsStudio un controllo TelemetryGauge riutilizzabile che disegna un anello, zone di soglia colorate, una lancetta, un valore numerico e una didascalia. Dagli le proprietà Value, Minimum, Maximum, WarningThreshold e CriticalThreshold, con setter che limitano l'input, escono subito quando nulla è cambiato e chiamano Invalidate() una volta sola. Metti i calcoli delle coordinate in un helper puro GaugeGeometry che riceve un Rectangle e restituisce i limiti dell'anello, gli angoli di apertura, i punti della lancetta e il rettangolo dell'etichetta, poi disegna a partire da quei valori con pen, brush e un GraphicsPath racchiusi in using, lasciando in pace il Graphics che ti viene passato. Gestisci senza eccezioni un controllo a dimensione zero, un intervallo invertito e un valore fuori intervallo, ed esponi la lettura tramite AccessibleDescription così da non affidarla ai soli pixel.

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

- ✓ Controllo TelemetryGauge con proprietà Value, Minimum, Maximum e di soglia che limitano il valore e chiamano Invalidate() una sola volta per ogni cambiamento reale
- ✓ Helper puro GaugeGeometry che restituisce i limiti dell'anello, gli angoli di apertura, i punti della lancetta e il rettangolo dell'etichetta
- ✓ Handler di Paint che fa il dispose di ogni pen, brush, font e path che crea e non fa mai il dispose di e.Graphics
- ✓ Protezioni per dimensione zero, intervallo invertito e valori fuori intervallo, con un AccessibleDescription che porta la lettura
- ✓ PaintCost.md che annota la dimensione del ridisegno, la sua frequenza e quando un controllo a tema costerebbe meno

Perché è fatto così

Il Modulo 1 ha messo un'unica lettura di telemetria su quattro superfici perché tu potessi vedere dove viene prodotto ogni pixel. Ora VisualOperationsStudio vuole quella lettura in sei punti, e copiare un handler Paint in sei pannelli è il modo in cui una schermata finisce con sei indicatori leggermente diversi, sei pen di cui nessuno fa mai il dispose e un ridisegno ogni volta che cambia un campo che non c'entra nulla. La soluzione è smettere di scrivere handler e scrivere un controllo: un solo TelemetryGauge le cui proprietà pubbliche sono l'unica cosa in grado di cambiarne l'immagine, e il cui costo sai esprimere in kilobyte.

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
Wisej.Web.PaintEventArgs	L'argomento dell'evento Paint: ti consegna un System.Drawing.Graphics che non hai creato tu e un rettangolo dentro cui disegnare, e il tuo codice gira sul server.
ClipRectangle	La regione che Wisej.NET ha messo a disposizione per questa operazione di disegno; il layout può consultare la client size del controllo, ma il disegno resta circoscritto al rettangolo che ti è stato dato.
helper GaugeGeometry	Una funzione pura che da un Rectangle più un valore normalizzato ricava i limiti dell'anello, gli angoli di apertura, i punti della lancetta e un rettangolo dell'etichetta, così i calcoli si possono verificare senza un controllo vivo.
Invalidate()	L'unica chiamata che chiede a Wisej.NET di renderizzare di nuovo la superficie del controllo; un setter di proprietà la fa esattamente una volta, e solo quando il cambiamento può alterare un pixel.
Proprietà delle risorse	L'handler fa il dispose di ogni Pen, Brush, Font, GraphicsPath e Bitmap che crea, e non fa mai il dispose del Graphics fornito dal framework.
Budget di costo del disegno	Tre numeri per ogni controllo disegnato a mano: la CPU del server spesa a disegnare, la memoria del server per l'immagine temporanea e i byte che il browser deve scaricare e decodificare.
AccessibleDescription	Porta la stessa lettura come testo, così un valore che esiste come arco disegnato resta disponibile anche a un utente che quell'arco non lo vede mai.

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



Usa Claude

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.

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 VisualOperationsStudio 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 - "Padroneggiare l'evento Paint dei controlli" - del corso LearnWisej "Disegno e grafica 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/drawing-and-painting/specs/module-02.md>

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

1. Progetto "VisualOperationsStudio", 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: Aggiungi a VisualOperationsStudio un controllo TelemetryGauge riutilizzabile che disegna un anello, zone di soglia colorate, una lancetta, un valore numerico e una didascalia. Dagli le proprietà Value, Minimum, Maximum, WarningThreshold e CriticalThreshold, con setter che limitano l'input, escono subito quando nulla è cambiato e chiamano `Invalidate()` una volta sola. Metti i calcoli delle coordinate in un helper puro GaugeGeometry che riceve un Rectangle e restituisce i limiti dell'anello, gli angoli di apertura, i punti della lancetta e il rettangolo dell'etichetta, poi disegna a partire da quei valori con pen, brush e un GraphicsPath racchiusi in using, lasciando in pace il Graphics che ti viene passato. Gestisci senza eccezioni un controllo a dimensione zero, un intervallo invertito e un valore fuori intervallo, ed esponi la lettura tramite AccessibleDescription così da non affidarla ai soli pixel.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Controllo TelemetryGauge con proprietà Value, Minimum, Maximum e di soglia che limitano il valore e chiamano `Invalidate()` una sola volta per ogni cambiamento reale
 - Helper puro GaugeGeometry che restituisce i limiti dell'anello, gli angoli di apertura, i punti della lancetta e il rettangolo dell'etichetta
 - Handler di Paint che fa il dispose di ogni pen, brush, font e path che crea e non fa mai il dispose di `e.Graphics`
 - Protezioni per dimensione zero, intervallo invertito e valori fuori intervallo, con un AccessibleDescription che porta la lettura
 - PaintCost.md che annota la dimensione del ridisegno, la sua frequenza e quando un controllo a tema costerebbe meno
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 sposta i calcoli dell'anello e della lancetta direttamente dentro l'handler Paint perché «sono solo poche righe». Quali due cose diventano poi impossibili da fare a basso costo, e da quale delle due dipende il Modulo 6?
- Il tuo indicatore è legato a un record di macchina che porta con sé anche il nome di un

operatore e una data di manutenzione. Se quei campi cambiano, il controllo deve invalidarsi? Spiega la risposta in termini di dipendenze visuali e non in termini di modello.

- Lo stesso indicatore finirà su una dashboard che ne mostra sessanta e si aggiorna ogni due secondi. Che cosa misureresti per primo, e che cosa cambieresti se le misure fossero cattive?

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 sposta i calcoli dell'anello e della lancetta direttamente dentro l'handler Paint perché «sono solo poche righe». Quali due cose diventano poi impossibili da fare a basso costo, e da quale delle due dipende il Modulo 6?
- ✓ Il tuo indicatore è legato a un record di macchina che porta con sé anche il nome di un operatore e una data di manutenzione. Se quei campi cambiano, il controllo deve invalidarsi? Spiega la risposta in termini di dipendenze visuali e non in termini di modello.
- ✓ Lo stesso indicatore finirà su una dashboard che ne mostra sessanta e si aggiorna ogni due secondi. Che cosa misureresti per primo, e che cosa cambieresti se le misure fossero cattive?

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/drawing-and-painting/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 VisualOperationsStudio 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 VisualOperationsStudio 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.