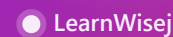


Architettura Wisej.NET enterprise, leadership tecnica e governance

Guida della lezione · Guida al lab / esame



Guida della lezione

Decisioni architetturali, standard di progetto, flusso di lavoro del team, gate di code review. Che cosa significa per un team Wisej.NET avanzato — e come affrontare questo modulo.



Focus di produzione

Decisioni architetturali, standard di progetto, flusso di lavoro del team, gate di code review.

1. Il problema in produzione

Il lavoro avanzato con Wisej.NET non riguarda più la possibilità di costruire una schermata. Riguarda la capacità di un team di far evolvere una grande applicazione per anni senza trasformare la produttività in debito tecnico nascosto.

2. Concetti chiave

Concetto	Perché è importante
struttura di una soluzione enterprise	Suddivide la soluzione in progetti di UI, servizi applicativi, dominio e infrastruttura, così le schermate restano sottili e le regole di business hanno un'unica sede.
architecture decision record	Documenti brevi che registrano contesto, opzioni, decisione e conseguenze di una scelta progettuale, così il team può rivederla e reconsiderarla in seguito.
suddivisioni compatibili con il designer	Mantiene form e pagine modificabili nel designer, mentre flusso di lavoro, validazione e accesso ai dati passano a servizi che il designer non tocca mai.
gate di revisione	Punti di controllo definiti in cui codice, architettura e sicurezza vengono rivisti prima che una funzionalità venga integrata, così la qualità è imposta e non solo sperata.
convenzioni del team	Regole condivise di denominazione, cartelle e registrazione dei servizi che rendono ogni schermata familiare a qualsiasi sviluppatore del team.
controlli sul debito tecnico	Pratiche come i registri del debito, i budget di refactoring e le regole di deprecazione, che mantengono le scorciatoie visibili e pianificate invece di dimenticarle.

3. Il filo della lezione

Gli sviluppatori di livello intermedio sanno costruire un'applicazione orientata alla produzione. Gli sviluppatori avanzati decidono come l'applicazione sopravvive alla crescita: più schermate, più utenti, più integrazioni, più team e più ambienti di deployment. L'obiettivo non è imitare un'architettura web generica. L'obiettivo è preservare il vantaggio di Wisej.NET - la ricca produttività in C# lato server - aggiungendo la disciplina sufficiente a sostenere il cambiamento enterprise.

Governance senza burocrazia

Un team Wisej.NET non dovrebbe aver bisogno di un comitato per aggiungere un pulsante. Ha però bisogno di standard su dove va la logica dei flussi di lavoro, su come le schermate usano i servizi, su come si impone la sicurezza, su come si astrae l'accesso ai dati e su come si aggiunge la diagnostica. Architecture decision record leggeri bastano, purché spieghino la decisione, il contesto, le conseguenze e la data di revisione.

Designer-first non significa senza architettura

Il designer di Visual Studio resta prezioso anche nei progetti avanzati. Il confine non è tra UI visuale e UI scritta in codice; il confine è tra lo stato della UI e le decisioni applicative. Form, Page e UserControl possono restare progettabili nel designer, mentre orchestrazione, autorizzazione, validazione, persistenza e integrazione restano nei servizi.



La regola di produzione avanzata

Non fermarti quando la schermata sembra corretta. Prima di considerare completo il modulo, spiega a chi appartiene lo stato, le implicazioni di sicurezza, i percorsi di errore e il comportamento in produzione.



Obiettivi di apprendimento

Alla fine di questo modulo sarai in grado di:

- ✓ Definire i confini architetturali di una grande applicazione Wisej.NET senza sovraingegnerizzarla.
- ✓ Creare standard di progetto per UI, servizi, accesso ai dati, integrazioni, sicurezza, diagnostica e deployment.
- ✓ Usare architecture decision record e gate di revisione per mantenere allineati i team.
- ✓ Spiegare come la produttività di Wisej.NET cambia le responsabilità di un technical lead.

Guida al lab / esame

Che cosa costruirai nel lab pratico, l'approccio suggerito e i deliverable richiesti.



Lab pratico

Crea la base della soluzione EnterpriseOps Command Center. Aggiungi progetti o cartelle per UI, controlli condivisi, dominio, servizi applicativi, accesso ai dati, integrazioni, sicurezza, diagnostica, risorse e deployment. Aggiungi una cartella per gli Architecture Decision Record e scrivi l'ADR-001 per la struttura di soluzione scelta.

Approccio suggerito

- ✓ Leggi il problema in produzione e individua il rischio architetturale prima di toccare la UI.
- ✓ Crea o aggiorna per primo l'incapsulamento in un servizio, comando, modello o componente coinvolto.
- ✓ Costruisci la schermata o il componente Wisej.NET seguendo gli standard di progetto stabiliti.
- ✓ Aggiungi una dimostrazione del percorso di errore, non solo del percorso felice.
- ✓ Annota che cosa è cambiato nel registro di apprendimento e aggiorna la checklist relativa.

Deliverable richiesti

- ✓ Screenshot o albero della struttura di soluzione e cartelle
- ✓ ADR-001 con la decisione sulla struttura della soluzione
- ✓ Pagina degli standard di codifica del team
- ✓ Esempio di denominazione di una schermata di riferimento e di incapsulamento nei servizi
- ✓ Checklist di code review applicata alla prima schermata

Domande di revisione per lo studente

- ✓ Un nuovo sviluppatore senior riesce a trovare il servizio di workflow di una schermata in meno di un minuto?
- ✓ Il team sa spiegare perché esiste un confine?
- ✓ Il designer è ancora utilizzabile dopo la suddivisione architetturale?



La regola di produzione avanzata

Non fermarti quando la schermata sembra corretta. Prima di considerare completo il modulo, spiega a chi appartiene lo stato, le implicazioni di sicurezza, i percorsi di errore e il comportamento in produzione.