



# L'altro modo di sviluppare applicazioni Web e Multidevice nel mondo .NET



Gabriele Del Giovine

<https://www.gabrieledelgiovine.it>

[gabriele@gabrieledelgiovine.it](mailto:gabriele@gabrieledelgiovine.it)

Ciao ChatGPT mi spieghi cosa è Wisej.NET in poche parole comprensibili da un Manager?

Wisej.NET è una piattaforma che permette di trasformare e sviluppare applicazioni web complesse in modo rapido, usando tecnologie .NET familiari e senza dover gestire la complessità del front-end. In pratica accelera lo sviluppo, riduce i costi e rende più semplice modernizzare applicazioni aziendali.



Ora me lo spieghi sempre con poche parole destinate ad uno Sviluppatore?

Wisej.NET è un framework .NET per costruire web app “desktop-like” con componenti server-side, evitando di scrivere JavaScript e gestire il front-end manualmente.





**Wisej.NET** è progettato per creare, eseguire il debug, distribuire e gestire applicazioni web di qualsiasi complessità tramite **Visual Studio** in una frazione del tempo rispetto ai metodi tradizionali ASP.NET/MVC, .NET Blazor o framework/librerie JS come Angular/React e simili.

È un framework altamente specializzato per applicazioni web in tempo reale; non è uno strumento generico per pagine web o siti web tradizionali.

Gli sviluppatori possono concentrarsi sui propri progetti senza preoccuparsi di HTML, gestione dello stato, servizi backend, callback, pannelli Ajax, sincronizzazione, DOM, CSS, JavaScript, sicurezza, autenticazione, concorrenza e altri dettagli tecnici.



# Curva di Apprendimento Minima

**Gli sviluppatori C# e VB.NET sono immediatamente produttivi, grazie a un ambiente e un paradigma di programmazione consolidati.**

**I sistemi di composizione della UI dichiarativi richiedono un cambiamento drastico nell'architettura e nel paradigma di programmazione orientato agli oggetti, basato sui componenti e guidato dagli eventi, ai quali la maggior parte degli sviluppatori di applicazioni aziendali (LOB) è abituata. Wisej.NET elimina questo cambiamento riconducendo il paradigma di sviluppo a quello delle .NET Windows Forms migliorandolo ed integrandolo con il meglio del mondo JavaScript.**

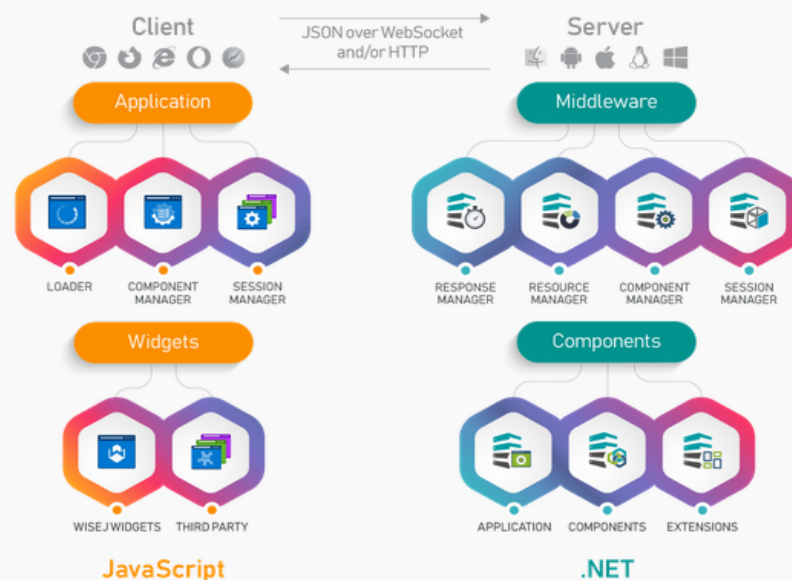


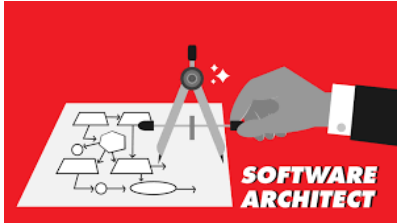
# Architettura Chiara

**Architettura Applicativa CHIARA, Integrata, efficiente, separata.**  
**JavaScript, HTML e CSS vengono usati per fare quello per cui sono stati progettati e realizzati.**

## Integrated and separated

Business logic and visual logic run on the server interacting with each other. Client side JavaScript widgets are in live sync with the server through simple, optimized, JSON packets. The widget library does what it's supposed to: render the UI in the browser. **There is no mixing of concerns resulting in a clean, secure, and scalable development system.**





## Per i Software Architect..

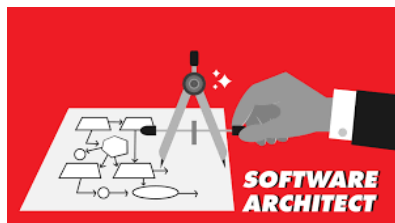
Wisej.NET non vi limita in alcun modo riguardo all'architettura software che volete usare per le vostre applicazioni, rimanendo ovviamente nei limiti dei macro-ambiti in cui si posiziona:

- 1) ASP.NET/ASP.NET Core;
- 2) Gestione della UI tramite Widget JavaScript concertati dagli handler server side;
- 3) Paradigma di Sviluppo di tipo Desktop (OO, Event Driven).

Siete quindi liberi di usare pattern tradizionali dello Sviluppo Windows Forms o pattern come MVVM (Mode, View, ViewModel), la gestione dei dati che volete (ADO, ORM vari, REST/API), le strategie di Dependency Injection che volete, le librerie .NET che volete. In sostanza qualsiasi cosa disponibile in .NET e JavaScript può essere usata.

**Wisej.NET non vi limita.**

Wisej.NET non è solo per programmatori C#, Wisej.NET fa le stesse cose anche se usate VB.NET



# Per i Software Architect che vogliono approfondire

<https://www.learnwisej.net/wisej.net-application-architectures/>



[Home](#) [Support](#) [Documentation](#)

Welcome

Discovering Wisej.NET >

Wisej.NET for Business Applications >

Wisej.NET Application Architectures ▾

Wisej.NET vs. MVC

Wisej.NET vs Blazor

Architecting 21st Century  
Web Apps using Wisej.NET >

Going Native with Wisej.NET Hybrid

Data Access with Wisej.NET >

Integrating with Wisej.NET >

Legacy Migration with Wisej.NET >

Authors >

Videos ↗

Concepts ↗

WISEJ.NET APPLICATION ARCHITECTURES

## Wisej.NET vs. MVC

by [Dino Esposito](#)

### Wisej.NET and ASP.NET: Two Sides of the Same Coin

In a previous article, we analyzed the general theme of migrating legacy Windows Forms applications to the web. A common practice entails wrapping up the backend of the application in an API layer and exposing it—altogether—as a service in a microservices architecture or a more traditional tiered (e.g., ASP.NET Core) application. In both cases, the sore point is the full redesign of the user interface with HTML5, CSS and JavaScript (or comprehensive frameworks such as Angular or React). Another nontrivial point to add up is that when you choose a modern frontend layer the mapping between UI connectors and legacy backend endpoints may not be as direct as it should be. The need of yet another intermediate query layer (e.g., GraphQL) emerges, adding costs and effort to the final bill.

In this context, Wisej.NET represents an alternate, and fairly unique, option.



Conference 2025  
Milano 28/11/2025



L'altro modo di sviluppare applicazioni Web e  
Multidevice nel mondo .NET





# Aperto ed Integrato

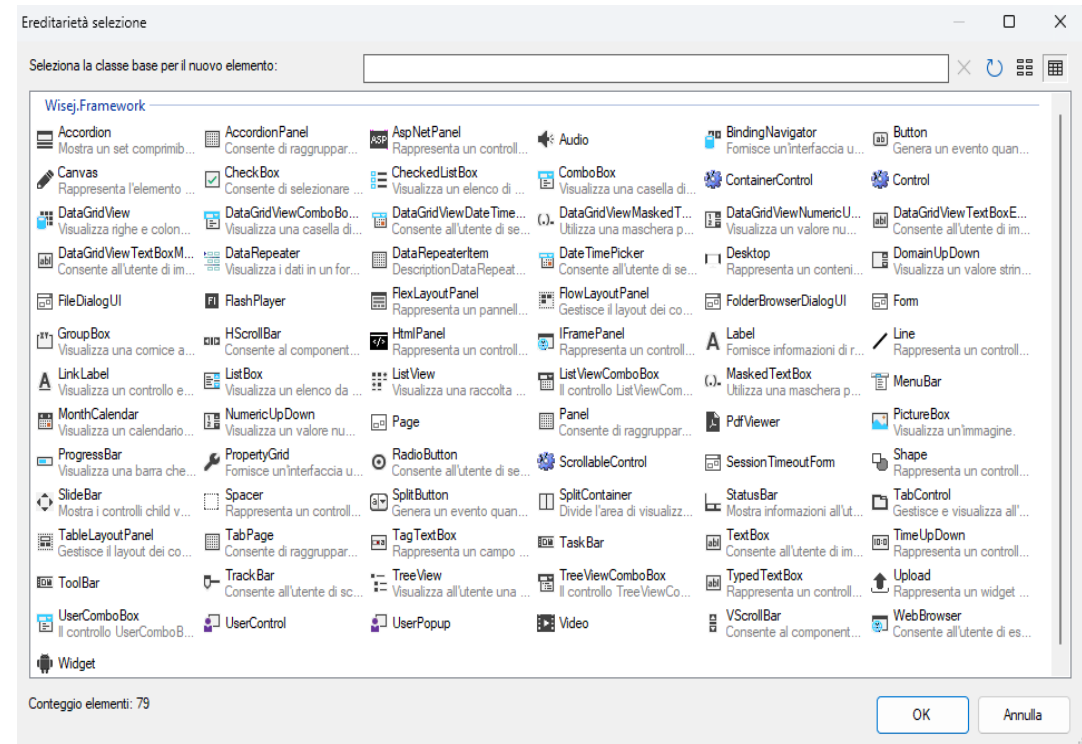
**Integra tutto il mondo .NET e la quasi totalità dei widget JavaScript in un framework standard e coerente, più semplice da mantenere e distribuire.**

**Wisej.NET si occupa del caricamento, dell'ottimizzazione, dell'inizializzazione, dell'instradamento, dello stile e della gestione dei widget, trattandoli sia come componenti lato server che come widget lato client.**



# Full Control Set

**Wisej.NET include già pronti all'uso tutti i controlli di cui avrai mai bisogno. E se alcuni dovessero mancare, si possono aggiungere tramite interfaccia Widget.**





## Le applicazioni Wisej.NET in modalità Hybrid (Hybrid = Wisej.NET + .NET MAUI)

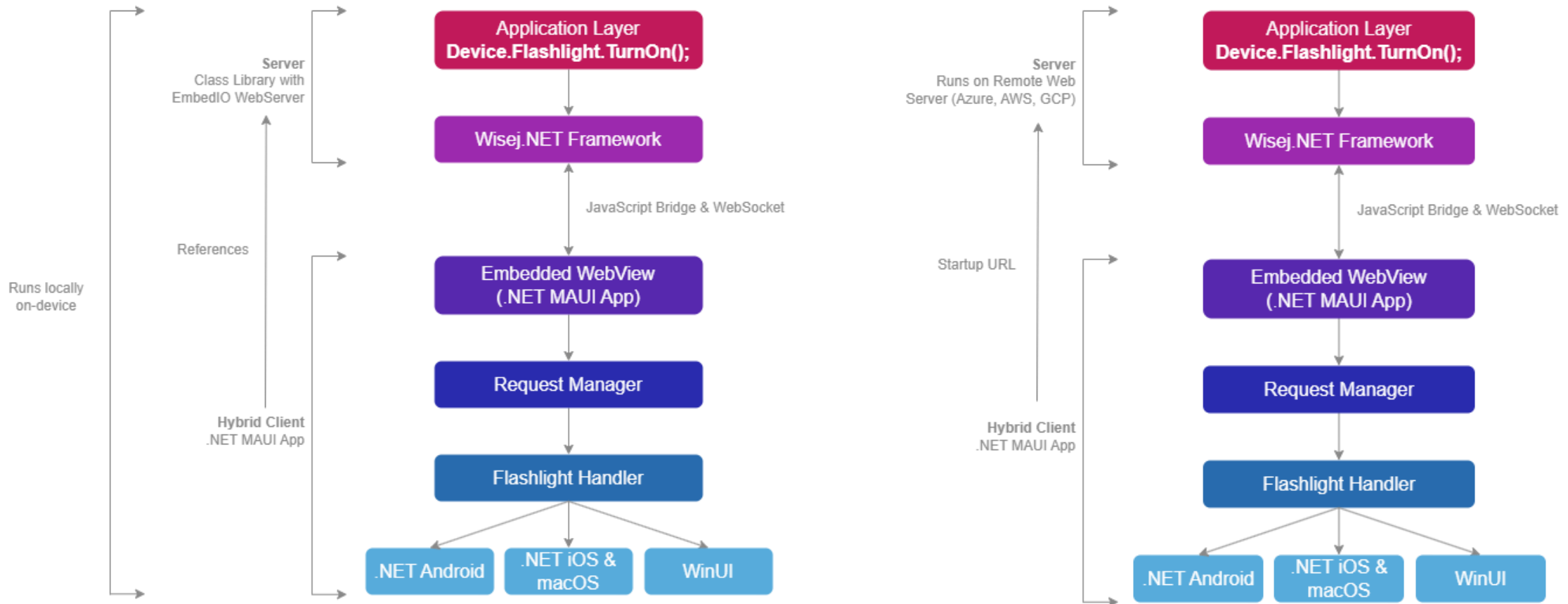
Le applicazioni Wisej.NET Hybrid sono applicazioni ASP.NET Core capaci di sfruttare le capacità NATIVE delle devices su cui girano. Le applicazioni possono funzionare localmente sulla device o in modalità remota.

<https://docs.wisej.com/hybrid/start/architecture>

<https://www.learnwisej.net/wisej.net-application-architectures/going-native-with-wisej.net-hybrid>



# Wisej.NET Hybrid Architettura

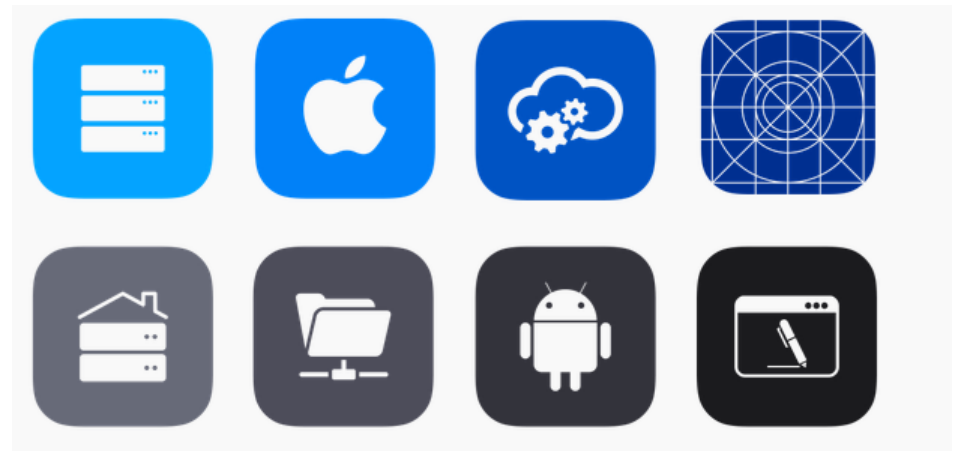




# Cloud Ready MultiPlatform Multidevice

**Le applicazioni Wisej.NET sono pronte per distribuzioni scalabili nel cloud, ambienti embedded e server locali.**

- Cloud Service
- Datacenter Server
- On-Premise Server
- Shared Site
- Self-Hosted Service
- Desktop Application
- PWA Application
- iOS Application
- Android Application





# Modernizzazione Applicazioni

[Introduction | Wisej.NET Migration](https://docs.wisej.com/migration)

<https://docs.wisej.com/migration>

[Dino Esposito Modernization White Paper – Wisej.NET](https://wisej.com/modernization-whitepaper-dino-esposito/)

<https://wisej.com/modernization-whitepaper-dino-esposito/>



# Cosa hanno fatto e fanno con **Wisej.NET**

<https://wisej.com/case-studies/>

<https://madewithwisej.com/>



# Documentazione & Learning

<https://wisej.com/documentation/>  
<https://www.learnwisej.net/>







**Connettere le business applications all'AI. SEMPLICEMENTE**  
**<https://wisej.ai/>**

Wisej.AI aggiunge funzioni di intelligenza artificiale alle applicazioni Wisej.NET senza dover scrivere codice complesso.

Funziona tramite un componente chiamato **SmartHub**, che collega i controlli dell'app a modelli AI, prompt e fonti dati.

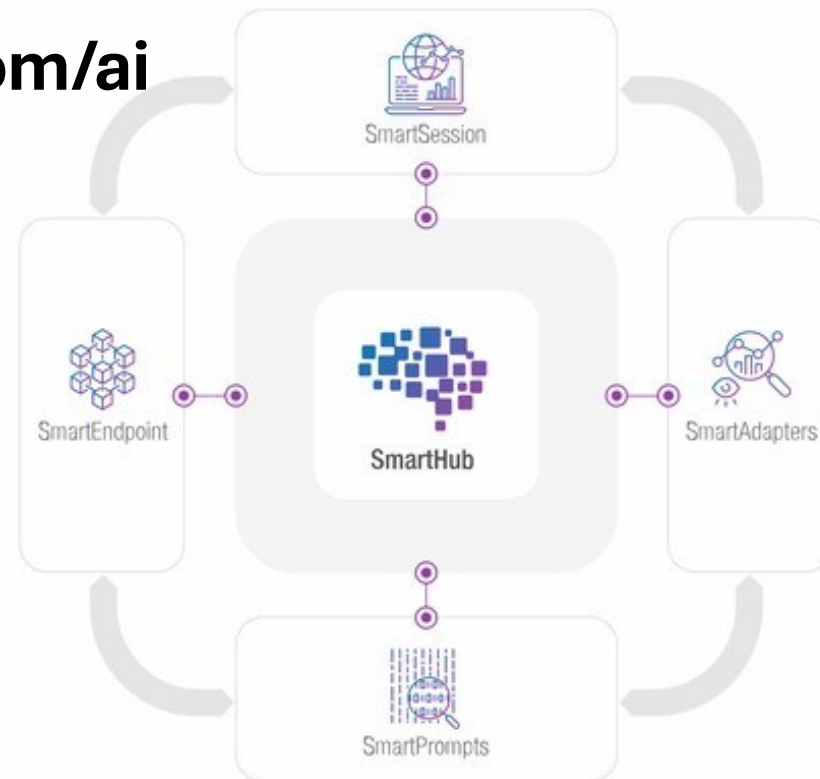
Una volta aggiunto il componente nel designer, ogni controllo può usare l'AI per generare testo, rispondere all'utente, classificare dati o automatizzare azioni, il tutto integrato nativamente nell'interfaccia Wisej.NET.



**Connettere le business applications all'AI. SEMPLICEMENTE**

**<https://wisej.ai/>**

**<https://docs.wisej.com/ai>**





## **Cos'è Wisej.AI**

È un insieme di **adattatori intelligenti (Smart Adapters)** che puoi collegare ai normali controlli Wisej (come ChatBox, ComboBox, Calendar, Chart, Audio, ecc.) per potenziarli con l'intelligenza artificiale.

Gli adattatori comunicano con modelli AI come **OpenAI GPT-4/4o, Anthropic Claude, Google Gemini, Llama 3**, e anche con servizi locali o custom.



### ◆ SmartDataEntryAdapter

È l'adattatore AI che si occupa di **arricchire i controlli di input** con metadati semantici, cioè informazioni che descrivono *il significato del campo*, non solo il suo contenuto.

Quando associ lo `SmartDataEntryAdapter` a un form o container, esso:

- estende i controlli figli con proprietà aggiuntive (es. `FieldName`, `FieldPrompt`, `FieldType`, ecc.),
- analizza i nomi, le etichette e i prompt per generare una mappa semantica del form,
- permette all'AI (tramite `SmartCopilot` o `SmartSession`) di **capire** cosa rappresentano i campi (es. "Email", "Numero ordine", "Data di nascita").



### ◆ SmartHub

Lo **SmartHub**, invece, **non aggiunge proprietà ai controlli**.

Il suo ruolo è **coordinare e orchestrare** le comunicazioni tra:

- sessioni AI (`SmartSession`)
- adattatori attivi (`SmartChatBoxAdapter`, `SmartCopilotAdapter`, ecc.)
- strumenti (`SmartTool`)
- modelli o provider AI configurati.

In altre parole:

🔧 **SmartDataEntryAdapter** arricchisce i controlli.

🧠 **SmartHub** gestisce le connessioni e le sessioni AI.



## Licensing

<https://wisej.com/developer-licenses/>

### Developer Licenses

License Model	Trial free	Community free	Professional \$1,190	Enterprise \$1,690
Developer Licenses	✓	✓	✓	✓
Server Licenses	✓	✓	✓	✓
Hybrid Licenses	✓	✓	✓	✓
Technology Partner	✓	✓	✓	✓
Services	✓	✓	✓	✓
Designer	✓	✓	✓	✓
Controls & Extensions	✓	✓	✓	✓
Integration Options	✓	✓	✓	✓
Theme Builder	✓	✓	✓	✓
Commercial Use	✓	✓	✓	✓
Premium Extensions	✓	✗	✓	✓
Direct Named Support	✓	✗	✗	✓
Priority Bugfixes Warranty	✓	✗	✗	✓
	<a href="#">Get It Now</a>	<a href="#">Get It Now</a>	<a href="#">Buy Now</a>	<a href="#">Buy Now</a>



## Licensing

<https://wisej.com/developer-licenses/>

# Hybrid Licenses

### License Model

Developer Licenses

Server Licenses

Hybrid Licenses

Technology Partner

Services

### Pricing (in USD)

free

[Get It Now](#)

from \$990

ask

Wisej.NET Hybrid Starter

Limited features, up to 100 instances

Wisej.NET Hybrid for Technology Partners

Full feature set, unlimited deployment – per year

Wisej.NET Hybrid Managed Deployment per App & Platform

White-label build & deployment services for major App Stores – per year



# Requisiti d'uso

## Ambiente di Sviluppo

**.NETFX 4.8 o superiore, .NET 8 o superiore;**

**Visual Studio 2019\*, 2022, 2026;**

**Windows 10/11 – Windows Server 2019/2022/2025**

**\* solo versione Wisej.NET 3.5**

## Ambiente Runtime

**Windows .NETFX 4.8 o superiore per target .NETFX, OS supportato da ASP.NET 8 o superiore per target .NET**





# Installazione ambiente di sviluppo

## Scaricare ed installare Designer Wisej.NET 2019 o 2022\*

\* Il designer 2022 è compatibile con Visual Studio 2026


**1) Da Visual Studio Market Place**

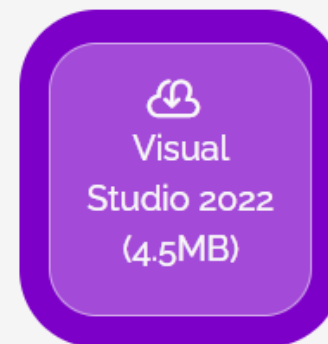
**2) Dal sito Wisej.Com - <https://wisej.com/builds/>**

### Designer

To use the Wisej.NET pixel-perfect designer with Visual Studio, download and install the latest version of the Visual Studio VSIX installer. It contains the designer, all the templates, the designer wizards and the F1 help extension.

You can also install, check for updates, or uninstall the Wisej.NET Designer in [Visual Studio's Extension Manager](#). Refer to our [troubleshooting](#) guide in case there are problems with Visual Studio.

 [Release Notes \(3.5.16\)](#)



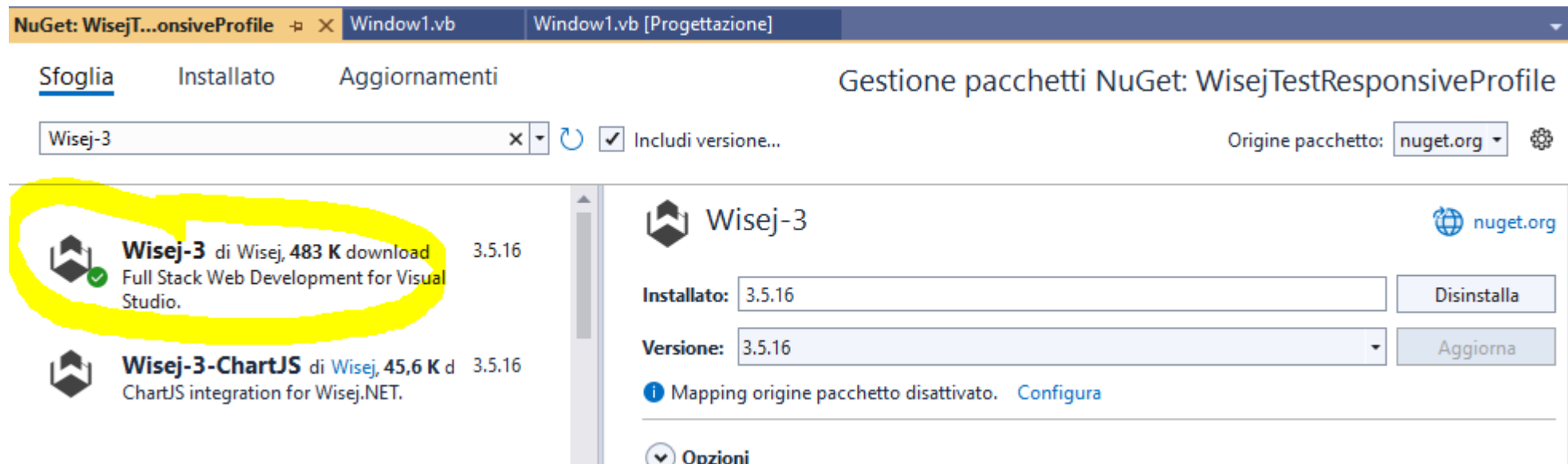


# Installazione ambiente di sviluppo

Via Web: scaricare ed installare il Package NuGET desiderato

<https://www.nuget.org/packages/Wisej-3> o

<https://www.nuget.org/packages/Wisej-4>



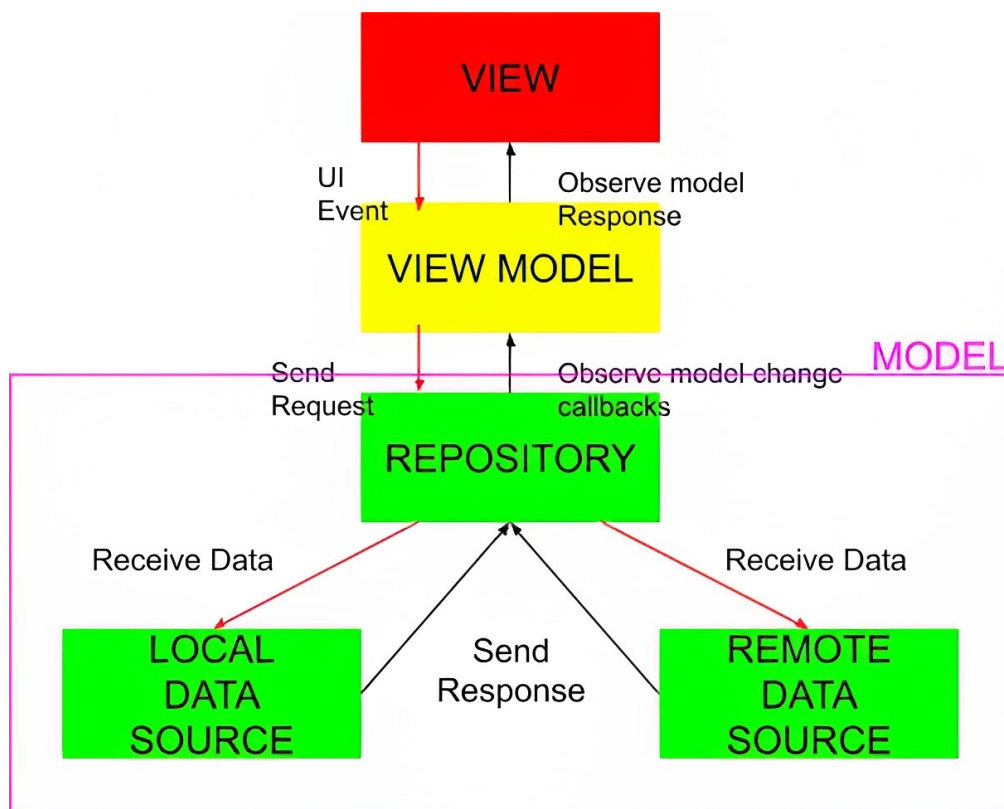


# Passero.Framework

Un Framework in tecnologia .NET per lo sviluppo di applicazioni gestionali WEB realtime (SPA) che usa Wisej.NET, il Pattern MVVM (Model, View, ViewModel e Repository) ed il MicroORM Dapper.

# I pilastri di Passero.Framework

- Microsoft .NET e Microsoft Visual Studio (quello vero);
- Usa Wisej.NET per lo sviluppo delle applicazioni WEB real-time capaci di funzionare su browser (desktop, tablet e smartphone) usando un modello semplice, potente e collaudato per lo sviluppo di applicazioni LOB (Line-Of-Business);
- Usa il Pattern Architetture MVVM per uno sviluppo ottimale delle varie parti dell'applicazione;
- Usa il MicroORM Dapper per l'accesso ai dati e lo estende con funzionalità avanzate;
- Usa Microsoft Reporting Services o FastReport per la parte di reportistica con le medesime funzionalità sia in .NET Framework che .NET.



- **MODEL** contiene la definizione dei dati;
- **VIEW** contiene l'interfaccia utente;
- **VIEWMODEL** contiene la logica di gestione dei dati del MODEL e risponde alle richieste della VIEW
- **REPOSITORY** esegue fisicamente le operazioni di scrittura/lettura dei dati nel MODEL da e verso le **SORGENTI DATI** (locali e remote)

# Ed ora passiamo al pratico...

# Demo codice:

- Una applicazione Wisej.NET di base e le basi dell'ambiente di sviluppo;
- L'interazione fra mondo server e mondo client (si, potete usare JavaScript, HTML e CSS);
- Passero.Framework, il mondo MVVM in Wisej.NET
  - PubsDemo, HotelDemo + HotelDemo Hybrid
- Uso dell'AI e di GitHub Copilot per la migrazione di applicazioni VB6.
- Wisej.AI;
- SharePoint SPFX;

# Materiale Demo

- Le demo (escluse quelle relative a Wisej.AI) le trovate su

<https://github.com/gdelgiovine/NETCONF2005>



# Q & A

# GRAZIE PER LA PAZIENZA!

Gabriele Del Giovine

email: [gabriele@gabrieledelgiovine.it](mailto:gabriele@gabrieledelgiovine.it)