



Architettura del Software - Quantum

Indice

1. 1. Introduzione
2. 2. Stack Tecnologico
3. 3. Architettura Generale
4. 4. Dettaglio Componenti
5. 5. Flussi Principali
6. 6. Sistema di Composizione Interfaccia (Docking)
7. 7. Sezione dedicata a Friuliantincendi srl
8. 8. Stato attuale e sviluppi futuri

1. Introduzione

Quantum è un software sviluppato da Aptasoftware srl, attivo e in continua evoluzione dal 2018. È una soluzione solida e matura, progettata per ambienti professionali in cui l'affidabilità, la flessibilità e la possibilità di personalizzazione sono fondamentali.

2. Stack Tecnologico

Livello	Tecnologie Principali
Database	PostgreSQL (compatibile con altri DBMS)
Backend	.NET Core 8 (C#) con supporto API REST, SignalR e logging integrato
Hosting	Windows Server (infrastruttura locale o cloud)
Frontend	Razor Pages, JavaScript, jQuery, librerie custom, DevExtreme (dxDataGrid)

3. Architettura Generale

Lo schema architetturale di Quantum si basa su una separazione chiara tra frontend, backend e database, con logica business interamente lato DB.

- Frontend (UI)
 - Razor Pages
 - JavaScript / jQuery
 - DevExtreme (dxDataGrid, popup, ecc.)

↓↑ Comunicazione: API REST + SignalR

- Backend (.NET Core 8, C#)

- Controller MVC / API Controller
- Chiamate dirette alle funzioni SQL
- ORM custom dinamico (no ADO, no EF)
- Logging centralizzato

↓↑ Accesso diretto a funzioni SQL

- Database (Business Logic)
 - PostgreSQL
 - Funzioni STABLE, viste, relazioni
 - Logica business centralizzata nel DB

4. Dettaglio Componenti

4.1 Database

- Cuore della logica business, sviluppata interamente in funzioni SQL.
- Utilizza PostgreSQL, ma con struttura compatibile anche con altri DBMS.
- Uso esteso di funzioni STABLE, viste, indici e logiche relazionali ottimizzate.

4.2 Backend

- Sviluppato in C# su .NET Core 8.
- Funziona da bridge tra frontend e DB, senza contenere logica di business.
- Utilizza un ORM dinamico custom (no ADO, no EF Core), che adatta gli oggetti alla struttura di viste e tabelle.
- Include API REST, SignalR e logging centralizzato.

4.3 Frontend

- Interfaccia moderna basata su Razor Pages, JavaScript, jQuery e DevExtreme.
- Componenti avanzati come dxDataGrid, popup dinamici, filtri, pulsanti SVG personalizzati.
- Integrazione con librerie custom per condizioni logiche e gestione intelligente del layout.

4.4 Hosting

- Quantum è ospitato su Windows Server, localmente o in cloud.
- Pubblicazione tramite IIS con supporto HTTPS.

5. Flussi Principali

Esempio – Inserimento Dato:

- L'utente inserisce dati tramite form DevExtreme.
- Il frontend invia la richiesta al controller API.

- Il backend chiama direttamente la funzione SQL nel DB.
- Il DB esegue la logica completa e restituisce l'esito.
- Il frontend aggiorna l'interfaccia dinamicamente.

6. Sistema di Composizione Interfaccia (Docking)

Quantum utilizza un sistema dichiarativo configurato tramite tabella nel database, che definisce dinamicamente la composizione dell'interfaccia utente in base alla voce di menu selezionata.

- Specifica quale form (Razor Page) utilizzare
- Dove posizionarlo nell'area di lavoro (layout docking)
- Relazioni master-detail tra form collegati

Questo approccio consente un'interfaccia flessibile e personalizzabile senza modificare il codice sorgente.

7. Stato attuale e sviluppi futuri

- Software maturo, in sviluppo dal 2018
- SignalR e logging già integrati
- Ulteriori sviluppi previsti su: esportazioni, moduli IA, ottimizzazioni UX
- Non si utilizzano framework UI complessi (es. React/Angular), per garantire massima trasparenza e controllo in produzione.