

Università degli Studi di Trento
Dipartimento di Economia e Management

Quantrix–Claude MCP bridge (2026):

plugin per Quantrix Modeler, server Model Context Protocol
e skill di Claude per la modellistica finanziaria
assistita dall'intelligenza artificiale

Nota tecnica sul prodotto della ricerca di tipo *Software*

Luca Erzegovesi

Rilascio	v2026.1 (30 settembre 2026)
Componenti	plugin <code>it.unitn.lerzegov.actions</code> 1.0.1; server MCP 7.3.0; skill <code>quantrix-scripting</code>
Periodo di sviluppo	gennaio – settembre 2026
DOI	10.5281/zenodo.23078960
Prodotto precedente	10.5281/zenodo.23064178 (versione 2025)
Accesso	file riservati, concessi su richiesta per ricerca e didattica

Trento, 14 ottobre 2026

Scheda del prodotto

Che cos'è. Un ponte software che permette a un assistente di intelligenza artificiale (Claude, di Anthropic) di leggere, spiegare, costruire e modificare i modelli finanziari aperti in *Quantrix Modeler*, un foglio di calcolo multidimensionale. Il ponte lavora sulla struttura propria del modello (matrici, categorie, elementi, formule), non su una griglia di celle.

Componenti. (1) un plugin Java che gira dentro *Quantrix* ed espone il modello attraverso un server HTTP locale; (2) un server Model Context Protocol (MCP) in Node.js con 40 strumenti; (3) una *skill* di Claude che codifica il metodo di lavoro: percorsi di esecuzione, schemi dei parametri, regole di sicurezza prima delle scritture.

A che cosa serve. Didattica e ricerca in finanza aziendale: analisi di bilancio, pianificazione finanziaria, modelli di valutazione, revisione semiassistita dei modelli consegnati dagli studenti (settembre 2026: 81 studenti, 1.077 revisioni di modelli).

Perché l'accesso è riservato. Gli strumenti automatizzano compiti che gli studenti dei corsi dell'autore devono svolgere a mano; la diffusione pubblica ne renderebbe possibile un uso improprio nelle prove d'esame. Il codice è depositato su Zenodo con file ad accesso riservato, concesso su richiesta per scopi di ricerca o didattica.

Sommario

La nota descrive la versione 2026 del ponte fra Claude e *Quantrix Modeler*. Rispetto al prodotto 2025 (Erzegovesi 2026b), basato su uno script Groovy eseguito dentro il modello, il server HTTP è stato riscritto come plugin Java dell'applicazione, il server MCP è passato da 12 a 40 strumenti e il metodo di lavoro è stato codificato in una *skill*. La nota presenta il problema, lo stato dell'arte, il contributo originale e la sua cronologia, l'architettura, una serie di esempi d'uso su un modello dimostrativo costruito con il ponte stesso, i limiti e le condizioni di accesso. Poiché i file sono riservati e il software richiede una licenza di *Quantrix Modeler*, la nota sostituisce per i valutatori la prova diretta di installazione ed esecuzione.

Abstract

This note describes the 2026 version of a bridge between the Claude AI assistant and *Quantrix Modeler*, a multidimensional spreadsheet for financial modelling. Compared with the 2025 product, the HTTP server embedded in *Quantrix* was rewritten as a Java plugin, the Model Context Protocol server grew from 12 to 40 tools, and the working method was encoded as a Claude skill. The note covers the problem, related work, the original contribution and its timeline, the architecture, worked examples on a demonstration model built through the bridge itself, limitations, and access conditions. The files are restricted, because the tools automate tasks that students must carry out by hand; access is granted on request.

1 Il problema affrontato

1.1 Modelli multidimensionali e assistenti di intelligenza artificiale

Nei corsi di finanza aziendale e di pianificazione finanziaria dell'autore i modelli si costruiscono in Quantrix Modeler (Quantrix, s.d.), erede di Lotus Improv. A differenza del foglio di calcolo a celle, Quantrix separa i dati, organizzati in matrici con categorie ed elementi nominati, dalle formule, scritte una volta per intere sezioni del modello con riferimenti per nome (per esempio Utile netto = EBIT - Imposte). È un'impostazione vicina a quella dei sistemi OLAP (Codd et al. 1993) e riduce gli errori tipici dei fogli a celle (Panko 1998; Hermans et al. 2015).

Gli assistenti di intelligenza artificiale generalisti sanno ragionare su un modello finanziario, ma non vedono il modello aperto sul computer dell'utente. Senza un ponte, l'interazione si riduce a incollare testi e immagini nella conversazione: lento, soggetto a errori di trascrizione e inutilizzabile per modificare il modello.

1.2 Il bisogno didattico e di ricerca

Il ponte risponde a tre esigenze.

- **Spiegare i modelli.** Ricostruire come si calcola una cella (formula, dipendenze, valori di partenza) è il primo passo per correggere un modello e per insegnare a leggerlo.
- **Costruire e modificare in sicurezza.** Creare matrici a partire da dati strutturati, scrivere ipotesi, lanciare scenari, senza mai scrivere nel modello sbagliato o in celle diverse da quelle volute.
- **Esaminare molti modelli.** Le consegne degli studenti sono centinaia di revisioni di modelli: servono strumenti che leggano struttura e testi senza aprire ogni file a mano.

1.3 Obiettivi della versione 2026

La versione 2025 (Erzegovesi 2026b) aveva dimostrato la fattibilità con dodici strumenti di lettura e scrittura, ma dipendeva da uno script avviato dentro un singolo modello. Gli obiettivi del 2026 erano: un componente stabile dentro l'applicazione, capace di servire tutti i modelli aperti; la costruzione di modelli, non solo la lettura; la resa di tabelle, grafici e poster come immagini; la sicurezza delle scritture; e un metodo di lavoro trasmissibile all'assistente.

2 Stato dell'arte

2.1 Il Model Context Protocol

Il Model Context Protocol (MCP), pubblicato da Anthropic nel novembre 2024 (Anthropic 2024) e oggi adottato da più fornitori, standardizza il modo in cui un assistente si collega a fonti di dati e strumenti esterni (Model Context Protocol 2025). Un *server MCP* dichiara un insieme di strumenti, ciascuno con nome, descrizione e schema dei parametri; l'assistente decide quando chiamarli e riceve testo o immagini in risposta. Il protocollo non dice nulla, però, su come rendere sicura la modifica di un'applicazione: questo resta a carico di chi scrive il server.

2.2 Assistenti nei fogli di calcolo

I principali fornitori hanno integrato assistenti nei fogli di calcolo a celle (per esempio Microsoft Copilot in Excel e Claude per Excel). Questi strumenti operano sulla griglia di celle e sulle sue conven-

zioni. Per i fogli multidimensionali come Quantrix, in cui la struttura per nomi è l'oggetto stesso della modellizzazione, non risultano all'autore integrazioni comparabili.

2.3 Skill e conoscenza procedurale

Le *Agent Skills* (Anthropic 2025) sono cartelle di istruzioni, esempi e riferimenti che l'assistente legge quando servono. Permettono di trasmettere conoscenza procedurale (come si fa, in che ordine, che cosa evitare) senza riaddestrare il modello. La skill di questo prodotto ne è un'applicazione a un dominio specialistico.

2.4 Lo spazio coperto

Il prodotto si colloca all'incrocio fra questi tre filoni: un server MCP specifico per un foglio multidimensionale, un componente nativo dentro l'applicazione e una skill che rende l'assistente un utente disciplinato del modello.

3 Contributo originale

3.1 Cosa è originale e cosa è ereditato

Sono **originali** dell'autore: il plugin Java (circa 11.400 righe in 50 file), il server MCP (circa 3.300 righe), la skill (circa 3.100 righe di documentazione più 17 script ed esempi) e gli script di libreria Groovy che costruiscono le matrici a partire da JSON. Sono **ereditati**, e non inclusi nel deposito: Quantrix Modeler e le sue librerie, il server HTTP JLHTTP (Freeutils, s.d.) e l'SDK MCP per Node.js. Il plugin usa l'interfaccia di scripting e i punti di estensione pubblici di Quantrix.

3.2 I contributi principali

1. **Un plugin nativo che espone il modello.** Il plugin aggiunge a Quantrix un menu *Strumenti* > *MCP Server* che avvia un server HTTP locale. I gestori coprono modelli, matrici, valori, formule, script, tabelle, grafici, poster e copia fra modelli; tutte le operazioni sono eseguite nel thread grafico di Quantrix, con limiti di tempo, e le modifiche come operazioni annullabili. Il plugin aggiunge inoltre azioni per comporre i poster (inserire una matrice, il suo titolo, impaginare, mostrare o nascondere categorie e formule) e un editor Markdown con espressioni vive.
2. **Un server MCP con guardie di sicurezza.** Il server traduce 40 strumenti in chiamate al plugin e aggiunge i controlli che il plugin non ha: verifica che il modello voluto sia davvero quello attivo, rifiuta le scritture su selezioni che non corrispondono esattamente ai valori forniti, riporta in ogni risposta il modello toccato e il valore precedente, convalida i parametri degli script con schemi dichiarati.
3. **Costruzione di modelli da dati strutturati.** Uno script di libreria riceve una descrizione JSON della matrice (categorie, elementi, dati, formule, formati) e la crea nel modello in un'unica operazione. È il percorso raccomandato per ogni modifica strutturale.
4. **Una skill che codifica il metodo.** La skill guida l'assistente: consultare i documenti pertinenti, dichiarare il percorso scelto, attendere conferma, eseguire, verificare e registrare quanto appreso. Ogni regola rimanda a un resoconto di sessione datato.

3.3 Cronologia dello sviluppo

Tabella 1: Cronologia del prodotto 2026

Data	Passaggio
mag.–nov. 2025	Versione 2025: script Groovy dentro il modello e server MCP 3.0.0, 12 strumenti (Erzegovesi 2026b)
24 gen. 2026	Primo commit del repository 2026; server MCP 3.1.0
25 gen. 2026	Server 4.0.0: esecuzione di codice Groovy libero
feb. 2026	Server HTTP riscritto come plugin Java; creazione di matrici da JSON; esportazione di tabelle, grafici e poster
17–21 feb. 2026	Sessioni di prova documentate; flusso a più matrici; plugin 1.0.1 compilato e installato; server 7.0.0
mar.–giu. 2026	Uso in didattica e ricerca; affinamento della skill e dei modelli di prova
21 set. 2026	Server 7.1.0 dopo il primo ciclo di revisione delle consegne (81 studenti, 1.077 revisioni)
30 set. 2026	Server 7.2.0–7.3.0: attivazione verificata del modello, parametri uniformi; rilascio v2026.1
1 ott. 2026	Deposito su Zenodo, DOI 10.5281/zenodo.23078960

4 Architettura

4.1 Visione d'insieme

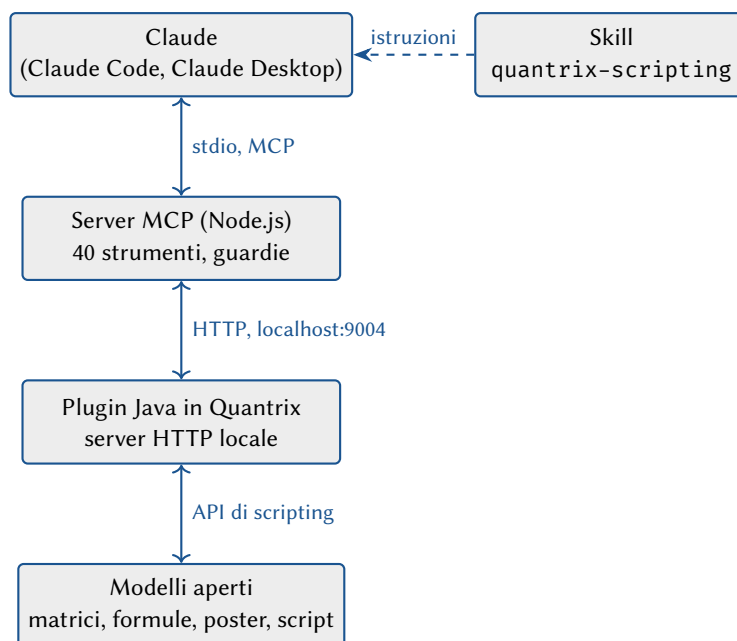


Figura 1: Architettura del ponte. L'assistente legge la skill quando serve e chiama gli strumenti del server MCP; il server inoltra le richieste al plugin, che opera sui modelli aperti in Quantrix.

L'assistente non tocca mai direttamente il modello: ogni operazione passa per il server MCP, che la convalida, e per il plugin, che la esegue dentro Quantrix (figura 1).

4.2 Il plugin Java

Il plugin `it.unitn.lerzegov.actions` è un bundle che Quantrix carica all'avvio dalla cartella dei plugin dell'applicazione. Il pacchetto `mcpserver` contiene il server HTTP e tredici gestori. Il pacchetto `canvasext` contiene le azioni per i poster e l'editor Markdown con espressioni del tipo `{Conto economico::Utile netto:2030}`, sostituite dal valore vivo della cella. Il pacchetto `nodes` gestisce taglio, incolla e cancellazione di nodi del navigatore fra modelli.

4.3 Il server MCP

Il server è scritto in JavaScript per Node.js e comunica con l'assistente sul canale standard (stdio). Ogni strumento ha una descrizione che contiene le regole da non dimenticare (sintassi delle selezioni, coordinate complete nelle scritture, verifica del modello attivo). Dalla versione 7.3.0 tutti gli strumenti usano gli stessi nomi di parametro (`matrix`, `selection`, `model`) e tutti quelli che agiscono sul modello accettano un parametro `model` facoltativo: il server attiva quel modello, verifica il cambio e, se non riesce, non esegue nulla.

4.4 I due percorsi di modifica

Le modifiche al modello seguono due percorsi.

- **Script di libreria con parametri (raccomandato).** L'assistente invia un JSON a uno script già presente nella libreria del modello; il plugin scrive i parametri in una matrice di servizio (Script params) e lancia lo script. Il risultato è riproducibile e il contratto dei parametri è dichiarato da uno schema.
- **Codice Groovy libero (ultima risorsa).** Per esplorazioni e correzioni puntuali.

4.5 Le guardie di sicurezza

Le guardie nascono da incidenti reali, documentati nei resoconti di sessione.

- **Modello attivo verificato.** Quantrix cambia modello attivo solo con un vero evento di attivazione della finestra; una richiesta inviata mentre un'altra applicazione è in primo piano può fallire senza errore. Il server porta Quantrix in primo piano, rilegge l'elenco dei modelli e segnala l'errore se il cambio non è avvenuto.
- **Scritture esatte.** Il server conta le celle a cui si riferisce la selezione e rifiuta la scrittura se non coincidono con i valori forniti: una selezione parziale aveva sovrascritto, in passato, la prima cella di una sezione.
- **Risposte che si possono controllare.** Ogni scrittura riporta il modello toccato e il valore precedente, così un errore si vede subito e si può annullare.

5 Esempi d'uso

Gli esempi usano un modello dimostrativo, *Piano finanziario demo*, costruito per questa nota con il ponte stesso: nessun dato di studenti o di terzi compare nelle figure.

5.1 Costruire un modello da dati strutturati

Il modello è nato vuoto. L'assistente (Claude Code) vi ha installato lo script di libreria testNewMatrix con save_script_from_file, ha copiato da un modello di prova la matrice di servizio Script params con copy_browser_nodes, poi ha creato due matrici con due chiamate a execute_script_with_params: le *Ipotesi* (ricavi dell'anno base, crescita, incidenza dei costi variabili, costi fissi, aliquota) e il *Conto economico* 2026–2030, con otto formule che collegano le voci fra loro e alle ipotesi. Il JSON che descrive il conto economico è riportato in forma abbreviata qui sotto.

```
{ "matName": "Conto economico",
  "columnCategoriesAndItems": [{"name": "anno",
    "items": ["2026", "2027", "2028", "2029", "2030"]}],
  "rowCategoriesAndNames": [{"name": "voce", "items": ["Ricavi", "Costi variabili",
    "Margine di contribuzione", "Costi fissi", "EBIT", "Imposte", "Utile netto"]}],
  "formulas": [
    "In anno[FIRST], Ricavi = Ipotesi::Ricavi anno base",
    "Ricavi = Ricavi:anno[PREV] * (1 + Ipotesi::Crescita ricavi)",
    "Costi variabili = Ricavi * Ipotesi::Incidenza costi variabili",
    "... ",
    "Utile netto = EBIT - Imposte" ] }
```

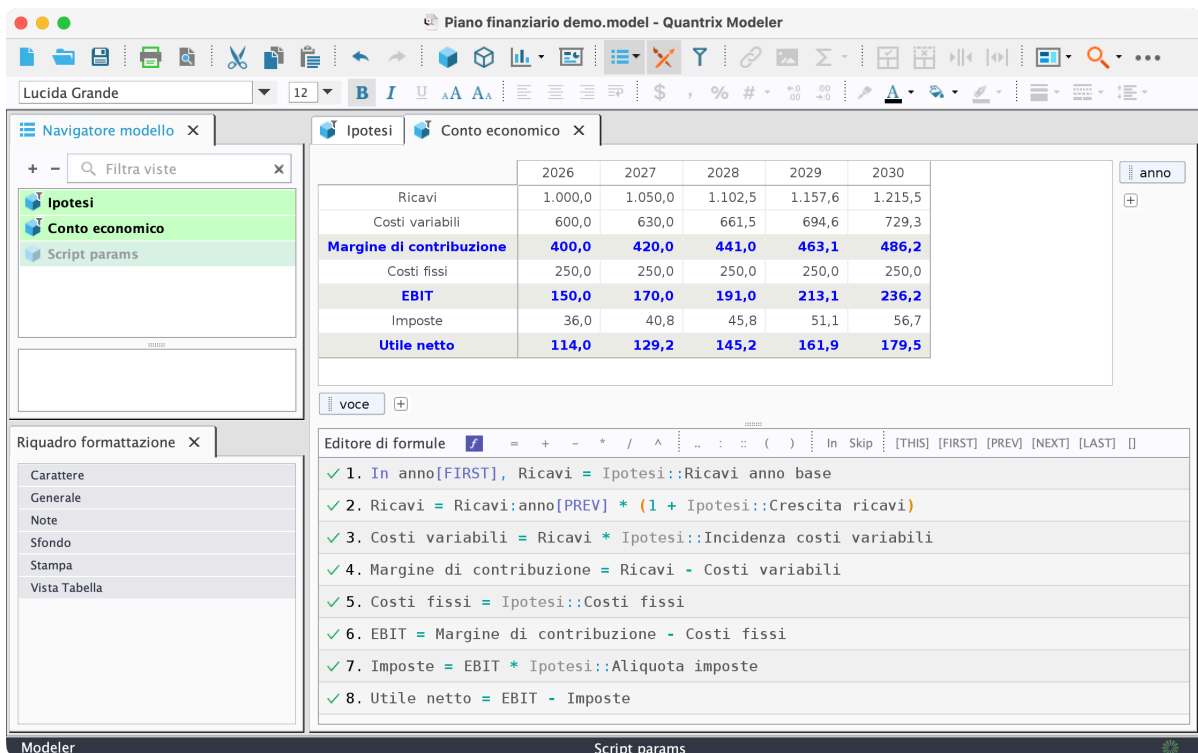


Figura 2: Il conto economico creato dal JSON, con l'editor delle formule di Quantrix: le otto formule sono valide e scritte per nome, non per indirizzo di cella. La formattazione è stata poi rifinita a mano.

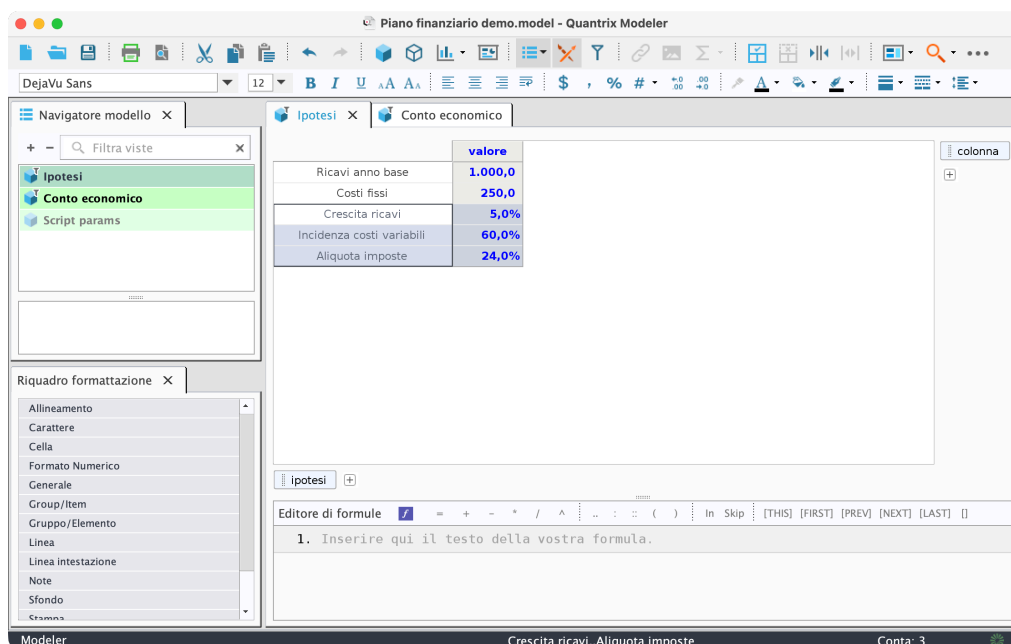


Figura 3: La matrice delle ipotesi, anch'essa creata da JSON: le celle di input sono evidenziate e le percentuali formattate come tali.

5.2 Spiegare un valore

Le figure che seguono sono tratte da una conversazione in Claude Desktop collegato al ponte. Il primo messaggio indicava il modello su cui lavorare e chiedeva di non modificare nulla senza autorizzazione.

Alla domanda «*come si calcola l'utile netto 2028?*» l'assistente chiama `explain_cell` indicando matrice, coordinate e modello; il server attiva il modello, verifica il cambio e restituisce valore, formula e albero delle dipendenze (Figura 4).



Figura 4: La domanda e la chiamata a `explain_cell`, con il modello attivato e verificato dal server.

Dall'albero restituito dallo strumento (17 foglie, molte ripetute perché alcuni percorsi sono condivisi) l'assistente ricava una versione compatta che arriva fino alle cinque ipotesi, e ne deduce la forma chiusa $\text{Utile netto}_{2028} = (1 - t) [R_0(1 + g)^2(1 - v) - F]$ (Figura 5).

Albero delle dipendenze

Lo strumento restituisce 17 nodi foglia, ma molti si ripetono perché alcuni percorsi sono condivisi (EBIT, ad esempio, entra sia direttamente sia nelle Imposte). Le ipotesi distinte sono solo cinque. Senza le ripetizioni, l'albero è questo:

Utile netto 2028 = 145,16	EBIT - Imposte
└ EBIT 2028 = 191	Margine di contribuzione - Costi fissi
└ Margine di contribuzione = 441	Ricavi - Costi variabili
└ Ricavi 2028 = 1102,5	Ricavi[anno prec.] × (1 + Crescita ricavi)
└ Ricavi 2027 = 1050	Ricavi[anno prec.] × (1 + Crescita ricavi)
└ Ricavi 2026 = 1000	= Ipotesi::Ricavi anno base
└ Costi variabili = 661,5	Ricavi × Incidenza costi variabili
└ Costi fissi 2028 = 250	= Ipotesi::Costi fissi
└ Imposte 2028 = 45,84	EBIT × Aliquota imposte

Ipotesi (foglie): Ricavi anno base 1000 · Crescita ricavi 5% · Incidenza costi variabili 60% · Costi fissi 250 · Aliquota imposte 24%

Se si sostituisce tutto fino alle ipotesi, la formula diventa:

$$\text{Utile netto 2028} = (1 - t) \cdot [R_0 \cdot (1 + g)^2 \cdot (1 - v) - F]$$

Qui R_0 sono i ricavi dell'anno base, g la crescita, v l'incidenza dei costi variabili, F i costi fissi e t l'aliquota. L'esponente 2 c'è perché il 2028 è due anni dopo l'anno base. La crescita è un'unica cella di Ipotesi, applicata uguale ogni anno.

Figura 5: L'albero delle dipendenze dell'utile netto 2028, ricostruito dall'assistente con i nomi del modello, e la formula chiusa che ne segue.

Alla seconda parte della domanda, quali ipotesi pesano di più, l'assistente risponde con un'analisi di sensitività calcolata dalla formula, senza toccare il modello, e riconosce nell'elasticità ai ricavi (2,31) la leva operativa del 2028, cioè il rapporto fra margine di contribuzione ed EBIT (441/191). Si offre poi di verificare i numeri nel modello, ma solo con l'autorizzazione dell'utente, perché servirebbe modificare temporaneamente le ipotesi (Figura 6).

Quali ipotesi pesano di più

Ho calcolato la sensibilità in modo analitico dalla formula, senza toccare il modello. La tabella mostra di quanto cambia l'utile netto se ogni ipotesi aumenta dell'1% del proprio valore:

Ipotesi	Valore	+1% sull'ipotesi	Effetto sull'utile netto
Incidenza costi variabili	60%	→ 60,6%	-3,46% (-5,03)
Ricavi anno base	1000	→ 1010	+2,31% (+3,35)
Costi fissi	250	→ 252,5	-1,31% (-1,90)
Aliquota imposte	24%	→ 24,24%	-0,32% (-0,46)
Crescita ricavi	5%	→ 5,05%	+0,22% (+0,32)

L'ipotesi che pesa di più è l'incidenza dei costi variabili, seguita dai ricavi dell'anno base. Entrambe agiscono su grandezze grandi (1102,5 di ricavi) che finiscono in un EBIT relativamente piccolo (191). L'elasticità ai ricavi, 2,31, è proprio la leva operativa del 2028: margine di contribuzione / EBIT = 441 / 191. Con un margine EBIT del 17% circa, piccoli scostamenti sui ricavi o sui costi variabili si amplificano molto sull'utile.

Se invece si ragiona in punti percentuali, la classifica cambia per crescita e aliquota. +1 punto di crescita (dal 5% al 6%) vale +6,41 di utile (+4,4%). +1 punto di incidenza dei costi variabili vale -8,38 (-5,8%). +1 punto di aliquota vale -1,91 (-1,3%).

Posso verificare questi numeri direttamente in Quantrix, cambiando le ipotesi e rileggendo l'utile. Però vorrebbe dire modificare celle di Ipotesi, anche se solo temporaneamente, quindi lo faccio solo se mi dai il permesso. In alternativa posso estendere l'analisi a tutti gli anni 2026-2030.

Figura 6: Sensibilità dell'utile netto 2028 alle ipotesi e interpretazione in termini di leva operativa.

5.3 Analisi *what-if* con scrittura verificata

Alla richiesta di portare la crescita dei ricavi al 7 % l'assistente legge i valori di partenza, scrive una sola cella con coordinate complete (valore:Crescita ricavi) e rilegge il risultato (Figura 7). Nella risposta riporta il modello in cui ha scritto e il valore precedente restituito dal server (0,05), confronta l'utile netto 2030 prima e dopo (da 179,51 a 208,48) e controlla il risultato con la formula chiusa ricavata prima. Infine segnala che il modello non è stato salvato e chiede se tenere il nuovo valore o tornare al 5 % (Figura 8).



Figura 7: La chiamata a `post_value`: matrice, modello, selezione completa e valore.

Ho portato **Crescita ricavi** dal 5% al 7%. Nella risposta alla scrittura il server conferma il modello giusto (Piano finanziario demo.model) e restituisce come **valore precedente 0.05**, cioè il 5% atteso.

Effetto sull'utile netto 2030

	Crescita 5%	Crescita 7%	Variazione
Utile netto 2030	179,51	208,48	+28,97 (+16,1%)

Entrambi i valori li ho letti dal modello, prima e dopo la modifica. Coincidono con la formula chiusa vista prima, con quattro anni di crescita al posto di due:

$$(1 - 0,24) \cdot [1000 \cdot 1,07^4 \cdot 0,4 - 250] = 0,76 \cdot (524,32 - 250) = 208,48$$

Due punti di crescita in più fanno salire i ricavi 2030 da 1215,5 a 1310,8 (+7,8%). L'utile netto però sale del 16,1%, circa il doppio. È la stessa leva operativa vista per il 2028: i costi fissi restano a 250, quindi tutto il margine aggiuntivo arriva all'EBIT. L'effetto qui è più forte che sul 2028 perché la crescita si compone su quattro anni.

Il modello in Quantrix ora contiene il 7%, ma **non l'ho salvato su disco**. Posso riportare la crescita al 5% oppure salvare il modello con il nuovo valore: dimmi tu quale preferisci.

Figura 8: Il risultato dell'analisi *what-if*: valore precedente confermato dal server, effetto sull'utile netto 2030 e verifica con la formula chiusa.

Una scrittura ambigua, con un solo valore per una selezione che corrisponde a più celle, viene invece rifiutata dal server prima di toccare il modello. Questo è l'esito, sul modello dimostrativo, di un tentativo di scrivere un solo valore nei costi fissi di tutti e cinque gli anni:

Write refused — nothing was written
 Model: Piano finanziario demo.model · Matrix: Conto economico · Selection: Costi fissi
 Cells the selection resolves to: 5 · Values to write: 1

5.4 Tabelle, grafici e poster restituiti come immagini

Il plugin rende come immagini le viste del modello, così l'assistente vede quello che vede l'utente. La Figura 9 mostra il poster *Report*, composto a mano in Quantrix con le matrici e il grafico del modello; la Figura 10 è la stessa vista come la riceve l'assistente con `get_canvas`, ritagliata automaticamente.

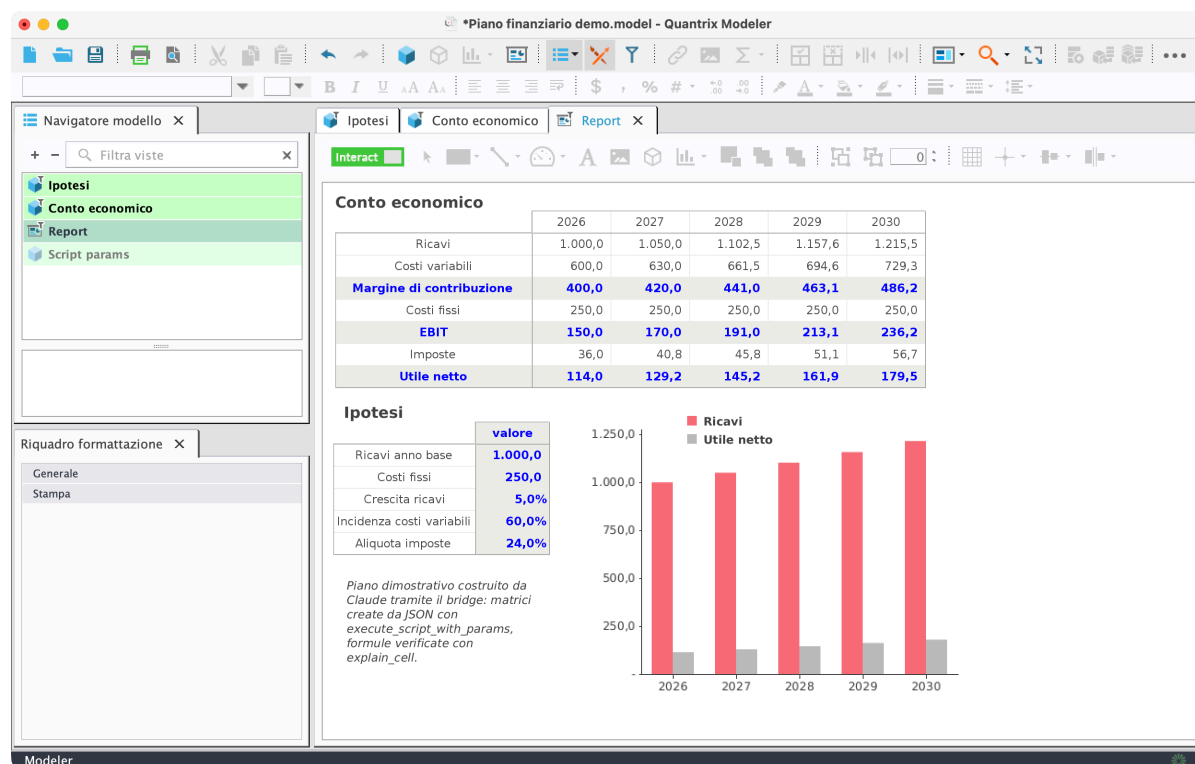


Figura 9: Il poster *Report* in Quantrix: conto economico, ipotesi, grafico e nota di testo. (Nell'interfaccia italiana di Quantrix il *canvas* si chiama *poster*.)

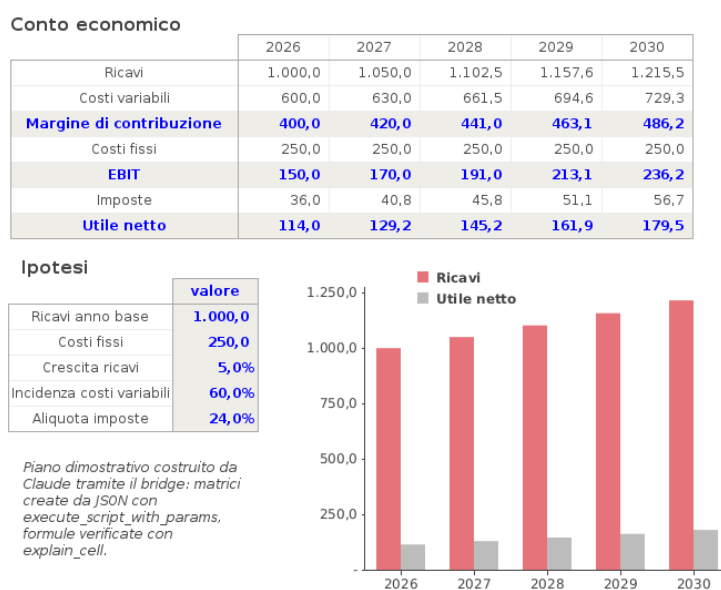


Figura 10: Lo stesso poster restituito all'assistente dallo strumento `get_canvas`.

5.5 Revisione delle consegne

Nel settembre 2026 il ponte ha sostenuto la revisione semiassistita delle consegne di un laboratorio di pianificazione finanziaria: 81 studenti, 1.077 revisioni di modelli. Per riservatezza la nota non mostra modelli né dati degli studenti. La revisione combina la lettura dei modelli attraverso il ponte con controlli automatici e con il giudizio del docente, che resta responsabile di ogni valutazione. Le difficoltà emerse in quel ciclo (cambio di modello non riuscito, errori non informativi, scritture parziali) hanno prodotto le guardie descritte nella sezione precedente.

6 Limiti e sviluppi

6.1 Limiti noti

- **Un modello attivo alla volta.** Il plugin opera sul modello attivo; il parametro `model` è realizzato attivando e verificando il modello a ogni chiamata, non instradando le richieste dentro il plugin. Su sistemi diversi da macOS l'attivazione può non riuscire, e lo strumento allora rifiuta di procedere.
- **Esito degli script.** Uno script di libreria che intercetta i propri errori risulta eseguito con successo anche quando non ha prodotto nulla; il risultato va verificato, come prescrive la skill.
- **Esportazioni legate alla vista.** Tabelle e grafici si rendono al meglio con la vista aperta in Quantrix; il grafico reso da solo non riporta la legenda, che invece compare nella resa del poster.
- **Licenza.** Il ponte richiede una licenza di Quantrix Modeler e una sottoscrizione a Claude.

6.2 Sviluppi

Instradamento delle richieste per modello dentro il plugin; restituzione degli errori degli script di libreria; estensione della convalida degli schemi; uso del ponte in altri corsi e, con le cautele dovute, come strumento di autoverifica per gli studenti.

7 Disponibilità, accesso e riproducibilità

7.1 Dove si trova

Il prodotto è depositato su Zenodo (Erzegovesi 2026a) come record pubblico con file riservati. Il deposito contiene il rilascio v2026.1 del repository dell'autore (sorgenti del plugin, server MCP con il file di blocco delle dipendenze, skill, modelli di prova propri), il plugin compilato e un file `SOURCES.md` con il commit di origine e l'impronta SHA-256 di ogni file. Il repository GitHub dell'autore resta privato. Il prodotto 2025 è depositato separatamente (Erzegovesi 2026b) e il record 2026 vi è collegato con la relazione *Continues*.

7.2 Accesso

L'accesso ai file è concesso su richiesta, tramite Zenodo, per scopi di ricerca o didattica. Non sono inclusi Quantrix Modeler, le sue librerie e la sua documentazione, né dati di studenti o modelli di terzi.

7.3 Prova del funzionamento

Non essendo possibile una prova di tipo *clona ed esegui*, il rilascio è stato verificato dall'autore il 30 settembre 2026 con Quantrix Modeler 26.1 su macOS, su copie di modelli di prova e su modelli creati

per l'occasione: 35 dei 40 strumenti sono stati chiamati, comprese le guardie di scrittura (selezione parziale rifiutata), la costruzione di matrici da JSON, il cambio verificato del modello attivo (8 cambi su 8), la copia fra modelli e la resa di grafici e poster. Il plugin installato è stato confrontato con i sorgenti del rilascio ricompilandoli: le firme di tutte le classi coincidono.

Una precisazione sul file `SOURCES.md` del deposito: indica come ultima modifica dei sorgenti del plugin il commit `c594885`; l'ultima modifica è in realtà il commit `b0411ce` dello stesso giorno (21 febbraio 2026), registrato pochi minuti dopo la compilazione del plugin installato. Il confronto descritto sopra conferma che il plugin depositato corrisponde ai sorgenti.

Riferimenti bibliografici

- Anthropic. 2024. «Introducing the Model Context Protocol». novembre. <https://www.anthropic.com/news/model-context-protocol>.
- Anthropic. 2025. «Equipping agents for the real world with Agent Skills». <https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills>.
- Codd, E. F., S. B. Codd, e C. T. Salley. 1993. *Providing OLAP to User-Analysts: An IT Mandate*. E. F. Codd & Associates.
- Erzegovesi, Luca. 2026a. *Quantrix–Claude MCP bridge (2026): Quantrix Modeler plugin, Model Context Protocol server and Claude skill for AI-assisted financial modelling*. Zenodo, released. <https://doi.org/10.5281/zenodo.23078960>.
- Erzegovesi, Luca. 2026b. *Quantrix–Claude MCP bridge: Model Context Protocol server and embedded HTTP server for AI-assisted analysis of Quantrix Modeler financial models (2025)*. Zenodo, released. <https://doi.org/10.5281/zenodo.23064178>.
- Freeutils. s.d. *JLHTTP: Java Lightweight HTTP Server*. <https://www.freeutils.net/source/jlhttp/>.
- Hermans, Felienne, Martin Pinzger, e Arie van Deursen. 2015. «Detecting and Refactoring Code Smells in Spreadsheet Formulas». *Empirical Software Engineering* 20 (2): 549–75.
- Model Context Protocol. 2025. «Model Context Protocol: Specification and documentation». <https://modelcontextprotocol.io>.
- Panko, Raymond R. 1998. «What We Know About Spreadsheet Errors». *Journal of End User Computing* 10 (2): 15–21.
- Quantrix. s.d. *Quantrix Modeler*. <https://www.quantrix.com>.

8 Gli strumenti del server MCP

Tabella 2: I 40 strumenti del server MCP 7.3.0

Gruppo	Strumenti
Lettura	matrices, matrix, selection, get_values, explain_cell, analyze_formula, get_matrix_formulas, model_summary, clear_cache
Scrittura	post_value, post_values, set_value, set_values, create_scenario
Script	execute_script_with_params, trigger_action, save_script_to_library, save_script_from_file, list_script_libraries, get_parameter_schema, list_script_schemas, execute_script_code
Esportazione	get_table_html, get_table_data, list_charts, get_chart, list_canvases, get_canvas, list_views
Modelli	list_models, open_model, create_model, save_model, save_model_as, activate_model, close_model, copy_browser_nodes
Diagnostica	server_info, canvas_text, read_model_file

canvas_text e read_model_file richiedono un lettore del formato dei file .model che non fa parte del deposito; senza di esso segnalano l'assenza del lettore.