



Manuale Utente

Versione: 0.0.1 10/04/2026

Redattori

Luca Parise

Verifica

Approvazione

Uso

Esterno

nextsoftpadova@gmail.com

Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
0.0.1	19/03/2025	Luca Parise	Creazione documento, stesura introduzione e altri capitoli	
0.0.2	10/04/2025	Luca Parise	Stesura dei capitoli riguardanti le funzionalità del plug-in	

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
2	Installazione	5
2.1	Requisiti	5
2.1.1	Requisiti minimi	5
2.1.2	Requisiti consigliati	5
2.1.3	Come installare il plug-in	5
3	Configurazione	6
3.1	Modello LLM	6
3.1.1	Prerequisiti	6
3.1.2	Come configurare il modello LLM ?	6
3.2	Soglia di accettazione	6
3.2.1	Prerequisiti	6
3.2.2	Come configurare la soglia dei requisiti ?	6
4	Istruzioni all'uso	7
4.1	Prerequisiti	7
4.2	Importazione requisiti	7
4.3	Analisi dell'implementazione	9
4.4	Analisi dei requisiti	10
4.5	Esportazione requisiti	12
4.6	Ricerca requisiti	12
4.7	Analisi di un singolo requisito	12
4.8	Analisi dell'implementazione di un singolo requisito	12
4.9	Filtraggio dei requisiti	12
5	Disinstallazione del plug-in	12

6 Elenco delle immagini	12
-------------------------------	----

1 Introduzione

“**Requirement Tracker per Visual Studio Code**” è un plug-in progettato per l’omonimo editor, con l’obiettivo di supportare l’analisi dei requisiti software. Dato un insieme di requisiti, sia tracciati che non tracciati, il plug-in esegue un’analisi approfondita e fornisce una valutazione del loro grado di implementazione all’interno del codice sorgente.

1.1 Scopo del documento

Il seguente manuale fornisce una guida dettagliata all’installazione, configurazione e utilizzo del plug-in. L’obiettivo è consentire agli utenti di comprendere il funzionamento dello strumento e di sfruttarne appieno le funzionalità per il monitoraggio e la valutazione dei requisiti software all’interno del codice sorgente.

In particolare, il documento si propone di:

- Fornire le istruzioni passo-passo per l’installazione e la configurazione
- Descrivere le funzionalità principali del plug-in
- Spiegare come eseguire l’analisi dei requisiti e interpretare i risultati

1.2 Scopo del prodotto

Il plug-in è progettato per supportare gli sviluppatori e i team di progetto nel tracciamento e nella verifica dei requisiti software. Il plug-in analizza il codice sorgente per identificare il livello di implementazione dei requisiti, fornendo una valutazione chiara e strutturata.

Il prodotto sfrutta l’intelligenza artificiale basata su modelli di linguaggio di grandi dimensioni (LLM) per migliorare l’analisi del codice e il riconoscimento dei requisiti. Grazie a questa tecnologia, il plug-in è in grado di comprendere il contesto del codice sorgente, rilevare correlazioni con i requisiti specificati e suggerire miglioramenti o integrazioni.

L’obiettivo principale è migliorare la gestione dei requisiti all’interno del ciclo di sviluppo, garantendo maggiore coerenza tra specifiche e codice, riducendo il rischio di requisiti mancanti o non conformi.

1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzate è stato creato un documento denominato **Glossario**. Questo documento comprende tutti i termini tecnici scelti dai membri del gruppo e utilizzati nei vari documenti con le relative definizioni. Tutti i termini inclusi in questo glossario vengono segnalati all’interno del documento con l’apice ^G accanto alla parola.

2 Installazione

2.1 Requisiti

Sono necessarie le seguenti tecnologie installate:

- Node.js versione 20.0.0 o superiore
- npm
- TypeScript (installabile con `npm install -g typescript`)
- Visual Studio Code
- Ollama attivo sulla porta standard localhost:11434

2.1.1 Requisiti minimi

2.1.2 Requisiti consigliati

2.1.3 Come installare il plug-in

Dovete innanzitutto scaricare il progetto sul vostro pc dalla cartella MVC su github.

2.1.3.1 Attivare Ollama

Per usare l'estensione è necessario che Ollama sia attivo e stia ascoltando sulla porta standard 11434. Per fare ciò, aprite un terminale e digitate il comando

```
$ ollama run nomemodello
```

dove nome modello indica il modello usato. Di base l'estensione usa il modello llama3.2:3b ma potete cambiarlo in qualsiasi momento andando su settings del plug-in. Per fare questo vi rimando al capitolo apposito per la configurazione del modello da usare.

2.1.3.2 Attivare il server API

Per fare questo dovete posizionarvi nella cartella API in cui troverete il codice in typescript che gestisce il server. Siccome viene usato node sarà necessario convertire il codice in javascript. Per fare ciò aprite un terminale, posizionatevi sulla cartella ed eseguite il comando

```
$ npm install
```

```
$ npx tsc
```

Questi comandi installeranno le dipendenze e convertiranno il codice typescript in codice javascript in una cartella chiamata dist. Quindi per avviare il server spostatevi dal terminale sulla cartella dist e cercate il file main.js. Quindi eseguite il comando

```
$ node main.js
```

Questo comando avvierà il server che rimarrà in ascolto delle richieste e le invierà ad Ollama. Se tutto è andato liscio, potete proseguire.

2.1.3.3 Attivare l'estensione

Ora posizionatevi sulla cartella dedicata all'estensione (vsExtensionDev) e aprite un terminale. Dal terminale usate il comando `$npm install`

3 Configurazione

3.1 Modello LLM

Il modello LLM di Requirement Tracker per Visual Studio Code viene eseguito tramite Ollama, permettendo un'analisi locale dei requisiti software. Esamina la qualità testuale dei requisiti e verifica la loro implementazione nel codice sorgente, restituendo un punteggio da 0 a 100 per indicare il grado di conformità. Inoltre, aiuta a identificare le sezioni di codice correlate ai requisiti e consente la configurazione di modelli personalizzati per ottimizzare le prestazioni dell'analisi.

3.1.1 Prerequisiti

I modelli di Ollama utilizzati sono configurabili tramite un file dedicato: `config.json`. Modello di default: `llama3.2:3b`

- Assicuratevi che Ollama sia attivo sulla porta 11434 prima di utilizzare l'estensione e che i relativi modelli siano disponibili.
- La porta diversa da quella di default si può specificare nel file: `.env` insieme alla porta dell'API.
- Nel file `config.json` del plugin si può modificare l'url del server node.

3.1.2 Come configurare il modello LLM ?

[Clicca qui](#) > [Clicca la](#) > [E Uala](#)

3.2 Soglia di accettazione

Quando un requisito viene sottoposto ad analisi, il modello LLM ne valuta l'implementazione nel codice sorgente e assegna un punteggio numerico compreso tra 0 e 100. Questo valore indica il grado di conformità del codice rispetto al requisito, fornendo una misura quantitativa della correttezza e completezza dell'implementazione.

3.2.1 Prerequisiti

3.2.2 Come configurare la soglia dei requisiti ?

4 Istruzioni all'uso

4.1 Prerequisiti

Se siete arrivati qui, allora avete installato il plug-in e siete pronti ad usarlo.

Quindi, cliccate sull'icona del plug-in presente nella colonna a sinistra della finestra di visual studio code per aprire la finestra di analisi dei requisiti. L'icona è mostrata nella figura sottostante.

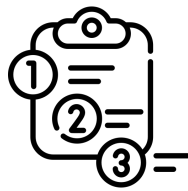


Figure 1: Icona del plug-in

4.2 Importazione requisiti

Per poter usare il plug-in è necessario importare i requisiti da analizzare. Per fare ciò, cliccate sull'icona denominata “**Load Requirements**” mostrata in figura.

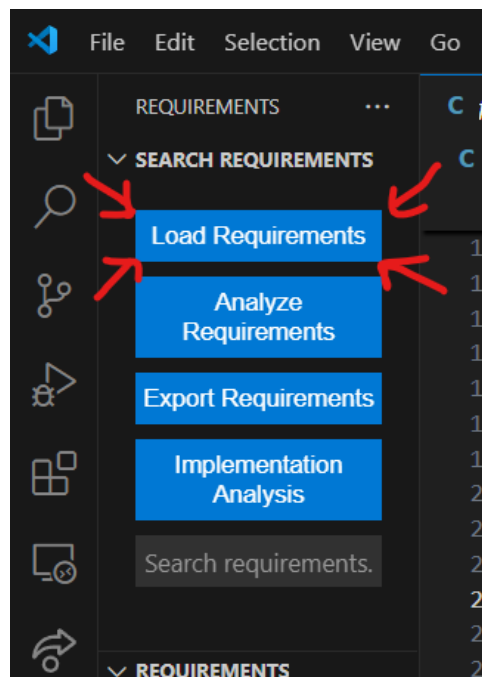


Figure 2: Icona per l'importazione dei requisiti

Una volta cliccato sull'icona verrà aperto il file system di sistema da cui selezionare il **file CSV** contenente i requisiti. In caso venga selezionato un file di formato diverso o non

valido, il plug-in restituirà un messaggio di errore.

E' consigliato, anche se non necessario, aver aperto in Visual Studio Code il progetto contenente il codice sorgente.

Una volta importati, i requisiti verranno presentati in una lista nella sezione Requirements, come mostrato nella seguente figura:

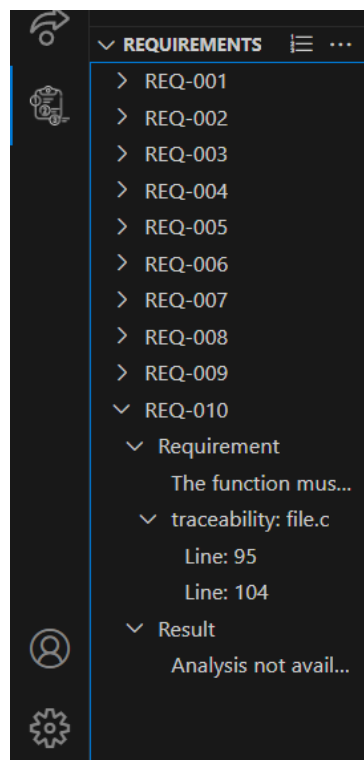


Figure 3: Requisiti importati

La lista, una volta selezionato un file CSV valido, elencherà per ogni requisito all'interno del documento:

- ID
- Descrizione
- Tracciabilità nel codice sorgente (solo se presente nel file CSV precedentemente caricato)

In caso di requisiti molto lunghi o di più righe sarà sufficiente posizionare il cursore sopra il requisito per visualizzare il testo completo.

4.3 Analisi dell'implementazione

Una volta importati i requisiti è possibile procedere con l'analisi dell'implementazione. Questa funzionalità serve a tracciare i requisiti nel codice sorgente qualora questi non fossero già tracciati (funziona anche nel caso lo fossero).

Per farlo, cliccate sull'icona “**Implementation Analysis**” mostrata in figura.

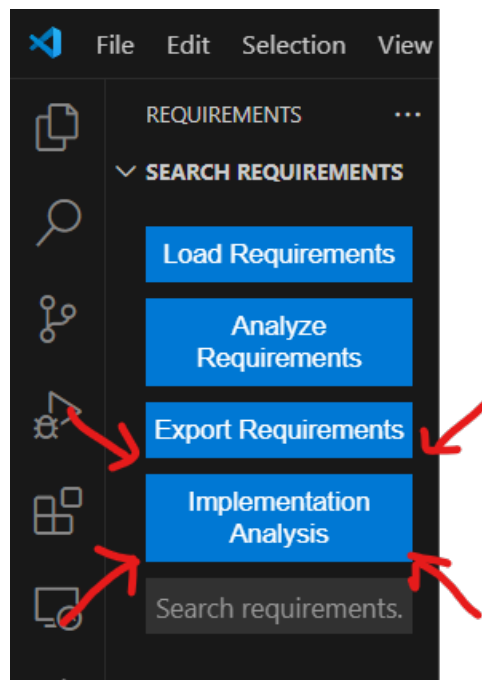


Figure 4: Bottone per l'analisi dell'implementazione

Una volta eseguita e completata l'analisi, ad ogni requisito verranno associate delle linee di codice che indicano il tracciamento del requisito all'interno del codice sorgente.

In caso di requisiti non tracciati, il plug-in restituirà un messaggio di errore. La figura seguente mostra il tracciamento nella sezione “Requirements”:

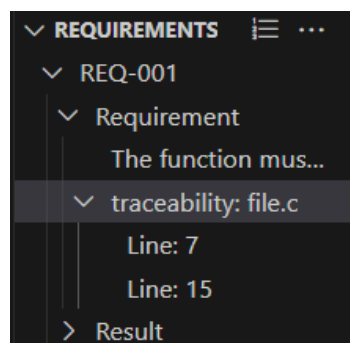


Figure 5: Tracciabilità di un requisito

4.4 Analisi dei requisiti

Questa funzionalità serve a valutare la qualità dei requisiti e il loro grado di implementazione nel codice sorgente. Una volta importati i requisiti, cliccate sull'icona **"Analyze Requirements"** mostrata in figura. In questo modo il plug-in eseguirà un'analisi approfondita dei requisiti. Potrete vedere in basso a destra della finestra un'icona di avanzamento che indica il progresso dell'analisi.

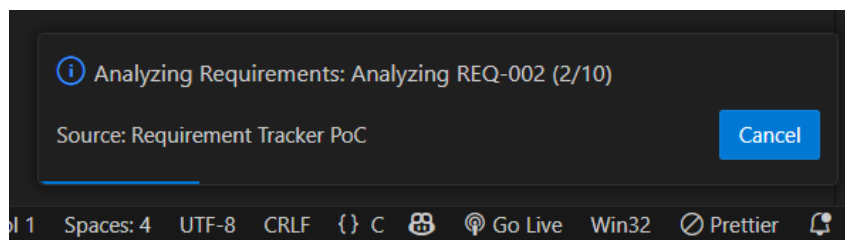


Figure 6: Bottone per l'analisi dei requisiti

Alla fine dell'analisi ogni requisito nella lista verrà aggiornato con i seguenti campi:

- Result
 - Result: passed/not passed
 - Code Compliance: 0-100
 - Issues:
 - Issue 1
 - Issue 2
 - Suggestions:
 - Suggestion 1
 - Suggestion 2

Inoltre, ad ogni requisito sarà associato un'icona che indica se è passato o meno all'analisi. Di seguito una figura che mostra una generica lista a seguito dell'analisi.

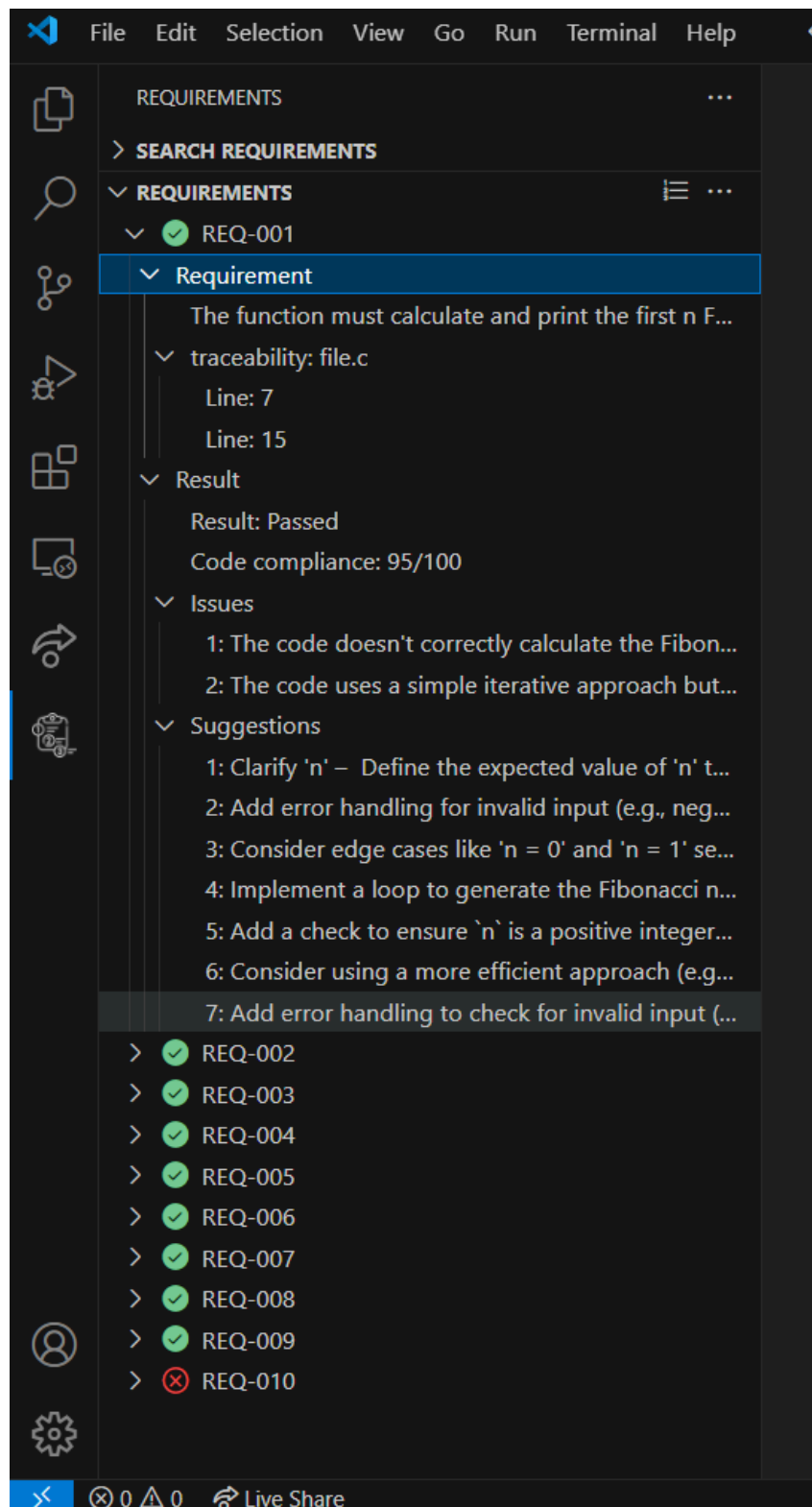


Figure 7: Risultato dell'analisi

4.5 Esportazione requisiti

4.6 Ricerca requisiti

4.7 Analisi di un singolo requisito

4.8 Analisi dell'implementazione di un singolo requisito

4.9 Filtraggio dei requisiti

5 Disinstallazione del plug-in

6 Elenco delle immagini