



# Manuale Utente

Versione: 0.0.1      12/04/2026

**Redattori**

Luca Parise

---

**Verifica**

---

**Approvazione**

---

**Uso**

Esterno

[nextsoftpadova@gmail.com](mailto: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	
0.0.3	11/04/2025	Luca Parise	Modifica alle immagini e continuazione stesura dei vari capitoli	
0.0.4	12/04/2025	Luca Parise	Aggiunto capitolo per la disinstallazione del prodotto	
0.0.5	14/04/2025	Luca Parise, Maria Fuensanta Trigueros Hernandez	Aggiunto capitolo della configurazione	
0.1.0	16/04/2025	Malik Giafar Mohamed, Luca Parise	Modificato il path per le immagini e migliorato il testo	
0.1.1	19/04/2025	Luca Parise, Malik Giafar Mohamed	Modifica capitolo per installazione immagine docker	

# Indice

1	Introduzione .....	5
1.1	Scopo del documento .....	5
1.2	Scopo del prodotto .....	5
1.3	Glossario .....	5
2	Installazione .....	6
2.1	Requisiti .....	6
2.1.1	Come creare il file .vsix .....	6
2.1.2	Come installare il plug-in .....	7
2.1.3	Attivare Ollama .....	7
2.1.4	Installazione dell'immagine docker .....	8
3	Configurazione .....	10
3.1	Modello LLM .....	10
3.2	Soglia di accettazione .....	11
4	Istruzioni all'uso .....	12
4.1	Prerequisiti .....	12
4.2	Importazione dei requisiti .....	12
4.3	Analisi dell'implementazione .....	14
4.4	Analisi dei requisiti .....	15
4.5	Esportazione requisiti .....	17
4.6	Ricerca requisiti .....	18
4.7	Analisi di un singolo requisito .....	18
4.8	Analisi dell'implementazione di un singolo requisito .....	19
4.9	Filtraggio dei requisiti .....	19
4.10	Approvazione di un requisito manuale .....	20
5	Disinstallazione del plug-in .....	21

## Elenco delle immagini

Figure 1	Configurazione del modello LLM .....	10
Figure 2	Configurazione dei modelli .....	11
Figure 3	Soglia di accettazione .....	11
Figure 4	Icona del plug-in .....	12
Figure 5	Icona per l'importazione dei requisiti .....	12
Figure 6	Requisiti importati .....	13
Figure 7	Bottone per l'analisi dell'implementazione .....	14
Figure 8	Tracciabilità di un requisito .....	14
Figure 9	Bottone per l'analisi dei requisiti .....	15
Figure 10	Stato di avanzamento analisi dei requisiti .....	15
Figure 11	Risultato dell'analisi .....	16
Figure 12	Bottone per l'esportazione dei requisiti .....	17
Figure 13	Campo di input per la ricerca dei requisiti .....	18
Figure 14	Bottone per l'analisi di un singolo requisito .....	18
Figure 15	Bottone per l'analisi dell'implementazione di un singolo requisito .....	19
Figure 16	Bottone per organizzare i requisiti in ordine di default .....	19
Figure 17	Bottone per organizzare i requisiti in ordine decrescente .....	20
Figure 18	Bottone per mostrare prima i requisiti analizzati .....	20
Figure 19	Bottone per mostrare prima i requisiti non analizzati .....	20
Figure 20	Icona delle estensioni .....	21
Figure 21	Icona del plug-in .....	22

# 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 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 <sup>G</sup> accanto alla parola.

## 2 Installazione

### 2.1 Requisiti

Sono necessarie le seguenti tecnologie installate:

- **Node.js**<sup>G</sup> (versione 20.x<sup>G</sup> o superiore)
- **npm**<sup>G</sup> (versione 10.9.0 o superiore)
- **TypeScript**<sup>G</sup> (versione 5.8.2 o superiore)
- **Visual Studio Code**<sup>G</sup> (versione 1.95.0 o superiore)
- **Ollama**<sup>G</sup> (versione 0.6.5 o superiore)

#### 2.1.1 Come creare il file .vsix

Per creare un *file .vsix*<sup>G</sup> (che è il pacchetto installabile di una estensione per Visual Studio Code), bisogna usare lo strumento **vsce**<sup>G</sup> (Visual Studio Code Extension Manager).

Di seguito i vari passaggi per creare tale pacchetto:

- **1)** Apri il terminale e installa **vsce** (Visual Studio Code Extension Manager), se ancora non lo hai installato, con il comando:

```
$ npm install -g vsce
```

- **2)** Naviga con il terminale fino alla cartella della tua estensione. Quindi vai nella cartella in cui si trova il file **package.json** della tua estensione. Puoi usare il seguente comando per navigare nella cartella:

```
$ cd /path/to/your/extension
```

- **3)** Costruisci il file .vsix con il seguente comando, il quale genera un file .vsix, ad esempio nome-estensione-0.0.1.vsix:

```
$ vsce package
```

- **4) (Facoltativo)** Ignora i file non necessari. Puoi usare un file .vscodeignore nella root della tua estensione per escludere i file o cartelle (come node\_modules, test/, ecc.) dal pacchetto finale.

### 2.1.2 Come installare il plug-in

Una volta creato il pacchetto .vsix come delineato nel capitolo precedente è possibile installare il plug-in in Visual Studio Code. Per farlo, segui questi passaggi:

- 1) Apri Visual Studio Code
- 2) Apri un terminale e naviga nella cartella in cui hai salvato il file .vsix. Puoi usare il comando:

```
$ cd /path/to/your/extension
```

- 3) Usa il seguente comando per installare il pacchetto .vsix:

```
$ code --install-extension nome-estensione-0.0.1.vsix
```

- 4) Se non vedi la estensione installata prova a riavviare Visual Studio Code
- 5) Verifica che l'estensione sia installata correttamente. Puoi farlo andando nella sezione delle estensioni di Visual Studio Code e cercando il nome del plug-in oppure, via terminale usando il seguente comando:

```
$ code --list-extensions
```

Ora l'estensione dovrebbe essere stata installata con successo. Per poterla usare avrete anche bisogno di Ollama attivo e configurato con un *modello LLM*<sup>G</sup>. Per farlo leggete il prossimo capitolo.

### 2.1.3 Attivare Ollama

Se non avete ancora scaricato ed installato Ollama vi rimando alla documentazione ufficiale che potete trovare al seguente link "<https://ollama.com/>" (14/04/2025).

Per usare l'estensione è necessario che Ollama sia attivo e stia ascoltando sulla porta standard 11434. Per fare ciò, potete cercare e cliccare sull'icona di Ollama una volta che è stato installato oppure, aprite un terminale e digitate il comando

```
$ ollama run nome_modello
```

dove "nome\_modello" indica il modello usato. Di base l'estensione usa il modello **llama3.2:3b** ma potete cambiarlo in qualsiasi momento andando sulle impostazioni dell'estensione. Per fare questo vi rimando al capitolo apposito per la configurazione del modello LLM.

### 2.1.4 Installazione dell'immagine docker

Per poter utilizzare correttamente il plug-in, è necessario eseguire un *server*<sup>G</sup> esterno che gestisca le richieste in arrivo. Questo server è realizzato come una REST API che può essere facilmente avviata e gestita tramite *Docker*<sup>G</sup>, una tecnologia che consente di creare ambienti isolati e replicabili chiamati *container*<sup>G</sup>.

L'utilizzo di Docker permette di semplificare il processo di configurazione, evitando problemi legati a incompatibilità tra versioni di librerie o ambienti operativi. Seguendo i passaggi descritti in questa sezione, sarai in grado di:

- Installare Docker sul tuo sistema (se non già presente)
- Costruire localmente l'*immagine*<sup>G</sup> dell'applicazione partendo da un file *Dockerfile*<sup>G</sup>
- Verificare che l'immagine sia stata correttamente creata
- Avviare un container funzionante, pronto a ricevere ed elaborare richieste

L'intera procedura richiede pochi comandi da terminale ed è stata progettata per essere semplice e accessibile anche a chi non ha una lunga esperienza con Docker. Assicurati di avere una connessione a internet attiva durante l'installazione e, se lavori su un sistema Linux, di avere i permessi necessari per eseguire i comandi come amministratore.

- **1)** Prima di tutto assicurati di aver installato Docker sul tuo sistema. Puoi scaricare Docker Desktop (per windows e macOS) oppure installare Docker Engine (per Linux) seguendo la guida ufficiale disponibile al link "<https://docs.docker.com/get-docker/>" (14/04/2025).

Una volta installato, verifica che docker sia correttamente attivo eseguendo da terminale il comando:

```
$ docker --version
```

Se il comando restituisce la versione di docker, l'installazione è avvenuta con successo.

- **2)** Spostati nella cartella del codice sorgente del server in cui si trova il file "**Dockerfile**" e il file "**docker-compose.yaml**". Puoi farlo usando il comando `cd`. Ad esempio, se il codice sorgente si trova in una cartella chiamata "API" all'interno della tua home directory, puoi usare il comando:

```
$ cd ~path/del/server/API
```

- **3)** Una volta nella cartella del codice sorgente (all'interno della directory denominata "hexagonalTest" della cartella MVP) devi costruire l'immagine Docker eseguendo il comando:



```
$ docker build -t requirement-tracker-api
```

- 3) Verifica che l'immagine sia presente usando il comando

```
$ docker images
```

- 4) Per eseguire l'immagine e creare il container usa il comando

```
$ docker run -p porta:4000 -e VAR=VAL requirement-tracker-api
```

Nel caso il nome dell'immagine sia cambiato, allora sarà necessario cambiare anche il nome nei comandi sopra. Lo stesso vale per la porta.

Un'altro metodo per avviare il server tramite docker consiste nel creare un “**docker compose file**”. Si tratta di un file in formato “yaml” il quale permette di modificare più facilmente i parametri di esecuzione del container con l'immagine del server. Inoltre, consente di configurare modalità di deployment più complesse, come l'uso di più container o l'integrazione con altri servizi.

Tutto questo non è possibile con il comando “docker run” in quanto questo permette di eseguire solo un container alla volta.

#### 2.1.4.1 Attivare il server manualmente

Per fare questo dovete posizionarvi nella cartella API in cui troverete il codice in *typescript*<sup>G</sup> che gestisce il server. Poichè viene usato node sarà necessario convertire il codice in *javascript*<sup>G</sup>. Per fare ciò aprite un terminale, posizionatevi sulla cartella del codice 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.

## 3 Configurazione

### 3.1 Modello LLM

L'utente può impostare il modello da utilizzare nelle varie operazioni come mostrato in figura:

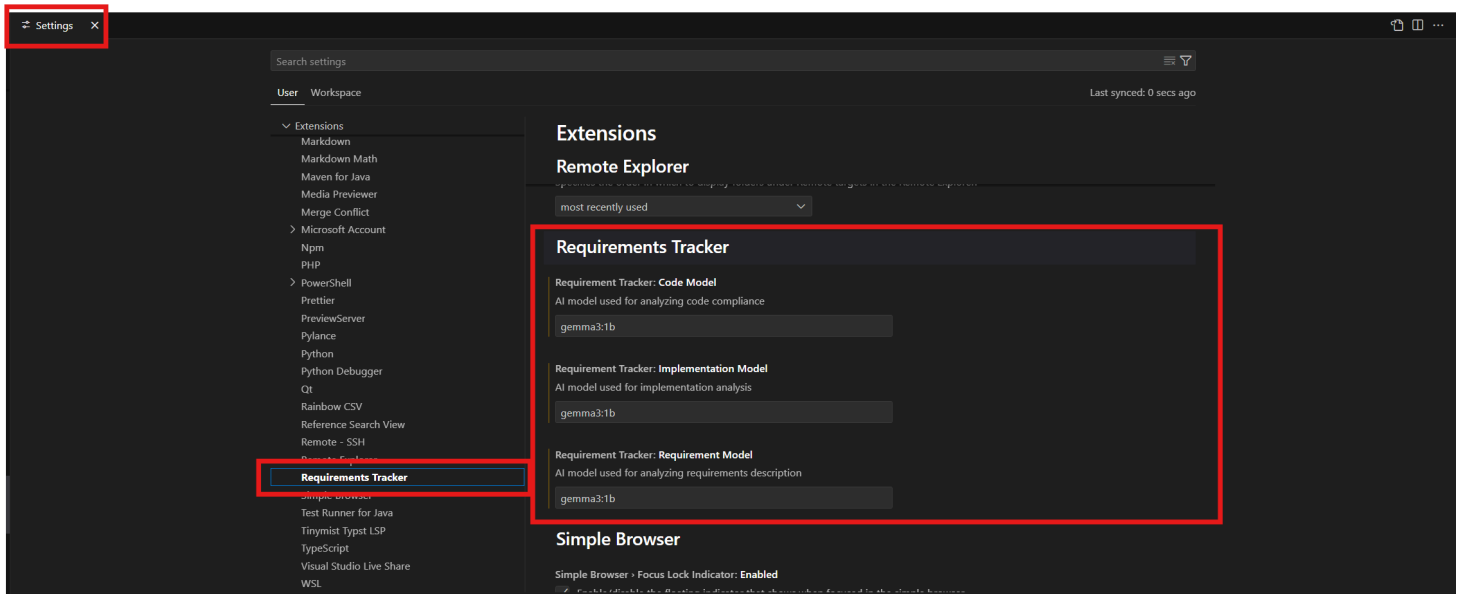


Figure 1: Configurazione del modello LLM

Il modello LLM di Requirement Tracker per Visual Studio Code viene eseguito tramite Ollama, permettendo un'analisi locale dei requisiti software. Visto il rapido progresso e la continua uscita di nuovi modelli, il plug-in permette all'utente di poter scegliere quale modello utilizzare, purchè questo sia installato all'interno della macchina.

Il plug-in permette di impostare un modello specifico per ognuna delle seguenti attività

- per analizzare la “code compliance”
- per analizzare l'implementazione
- per analizzare la descrizione dei requisiti

Per poter configurare il modello, l'utente deve accedere alle impostazioni della estensione installata ed inserire nei tre campi disponibili il nome del modello che vuole usare, come mostrato nelle figura di seguito :

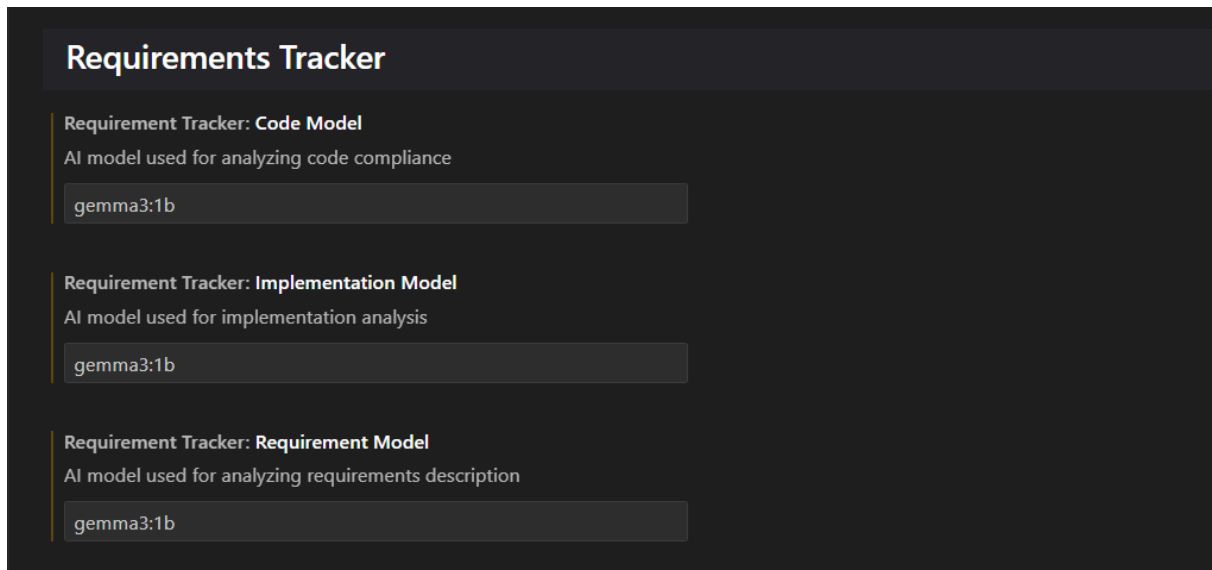


Figure 2: Configurazione dei modelli

### 3.2 Soglia di accettazione

Un'altra configurazione permessa dal plug-in è l'impostazione della *soglia di accettazione*<sup>G</sup> di un requisito. Ad ogni analisi, infatti, ogni requisito riceve un punteggio da 0-100 e viene considerato "passed" o "not passed" a seconda del superamento di tale soglia. Per impostarla, andate sulle impostazioni ("Settings") del plug-in e inserite nel campo riguardante la soglia il valore che preferite come mostrato di seguito in figura :

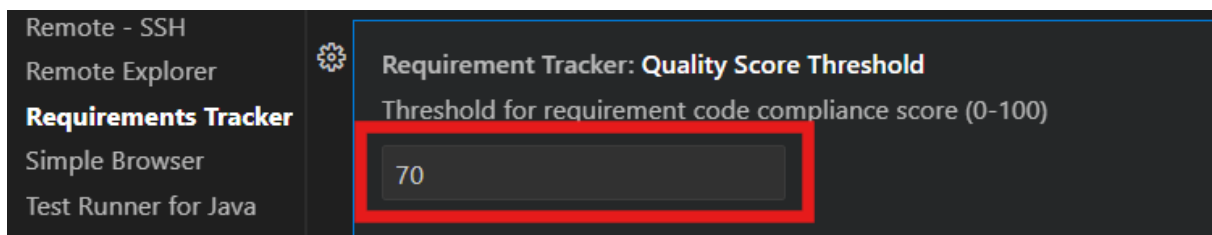


Figure 3: Soglia di accettazione

## 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.

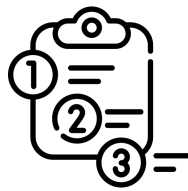


Figure 4: Icona del plug-in

### 4.2 Importazione dei 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.

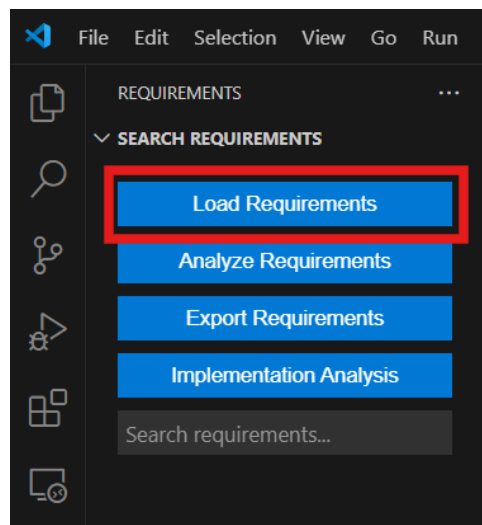


Figure 5: Icona per l'importazione dei requisiti

Una volta cliccato sull'icona verrà aperto il *file system*<sup>G</sup> 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.

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

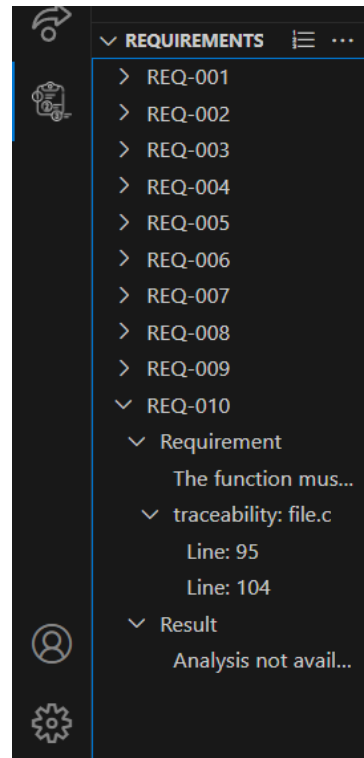


Figure 6: Requisiti importati

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.

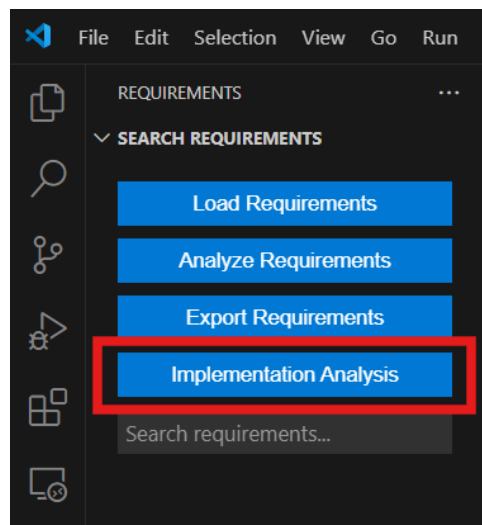


Figure 7: Bottone per l'analisi dell'implementazione

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":

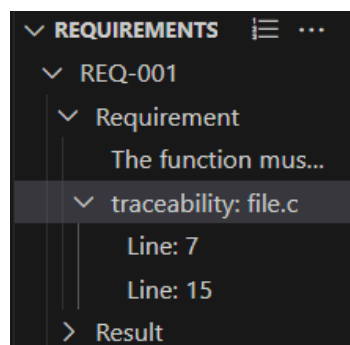


Figure 8: Tracciabilità di un requisito

## 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.

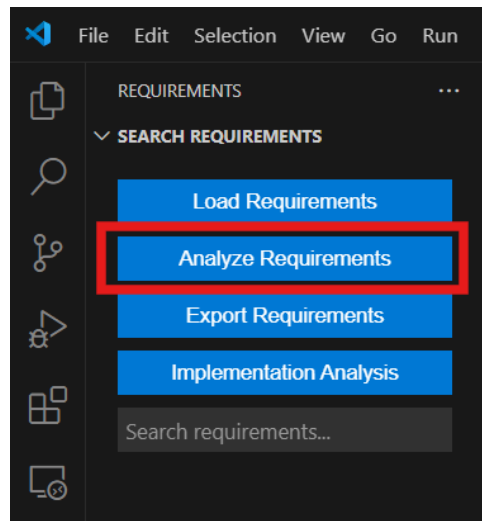


Figure 9: Bottone per l'analisi dei requisiti

In questo modo il plug-in eseguirà un'analisi approfondita di tutti i requisiti presenti nella lista. Potrete vedere in basso a destra della finestra un'icona di avanzamento che indica il progresso dell'analisi.

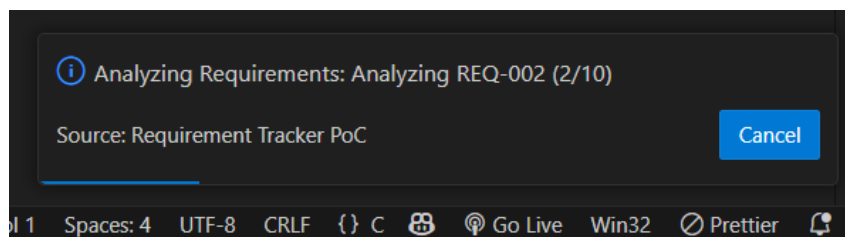


Figure 10: Stato di avanzamento analisi dei requisiti

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 ...

- Suggestions:
  - Suggestion 1
  - Suggestion ...

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

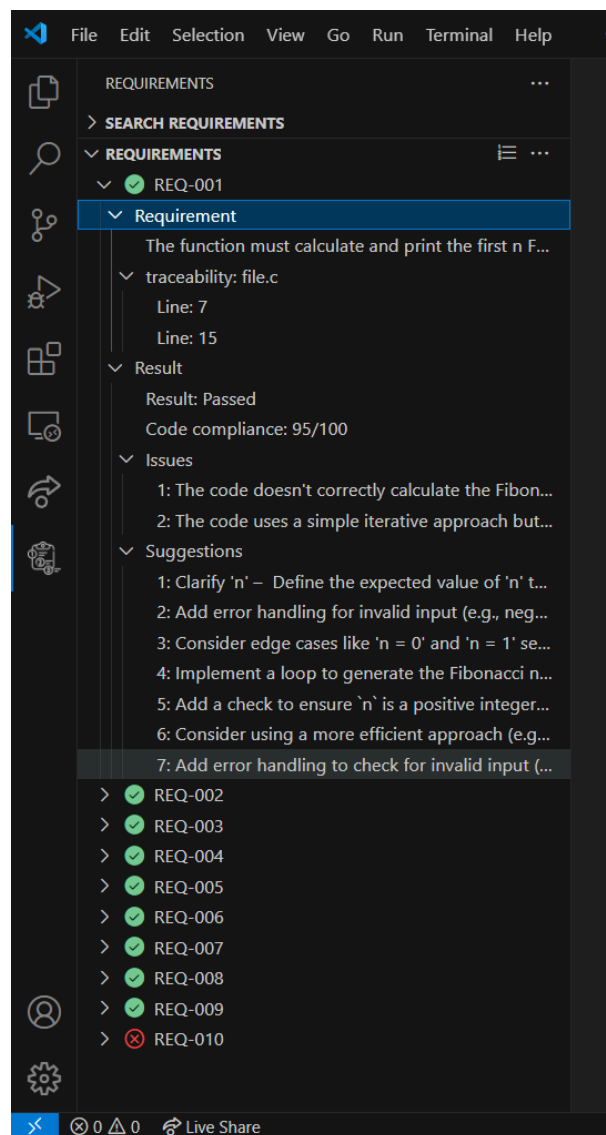


Figure 11: Risultato dell'analisi



## 4.5 Esportazione requisiti

Il plug-in presenta anche la funzione di esportazione, in formato CSV<sup>G</sup>, dei risultati dell'analisi, qualora volesse salvarli. Per fare ciò premete sul pulsante **“Export Requirements”** mostrato nella seguente figura.

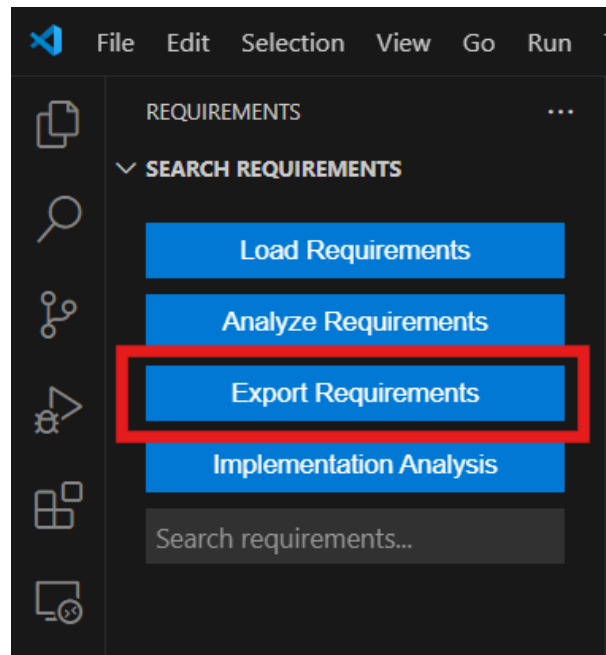


Figure 12: Bottone per l'esportazione dei requisiti

Quindi, una volta cliccato sul pulsante, verrà aperto il file system di sistema da cui selezionare la cartella in cui salvare il file CSV.

## 4.6 Ricerca requisiti

Il plug-in presenta anche la funzione di ricerca dei requisiti attraverso una barra di ricerca in cui inserire il codice del requisito ricercato. Per farlo, basta cliccare sull'icona “**Search Requirements**” mostrata in figura.

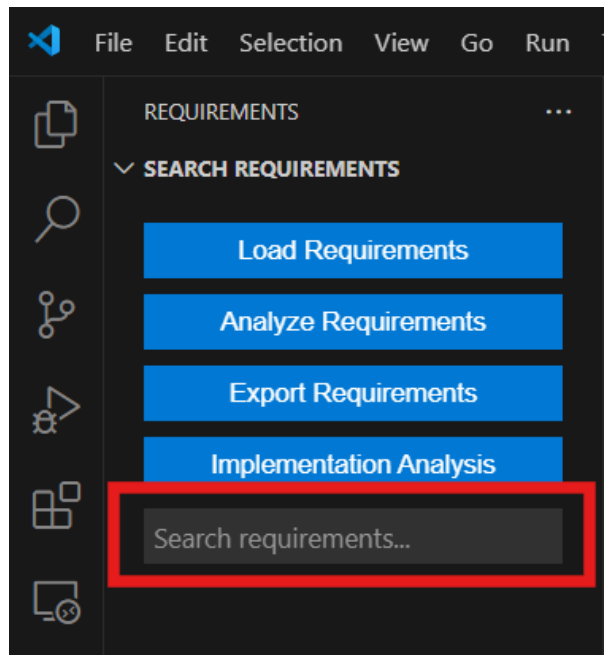


Figure 13: Campo di input per la ricerca dei requisiti

La lista **Requirements** mostrerà solo i requisiti che contengono il codice (o parte di esso) inserito nella barra di ricerca.

## 4.7 Analisi di un singolo requisito

Il plug-in offre la possibilità di analizzare un singolo requisito. Per farlo, basta cliccare sull'icona presente vicino al requisito che si desidera analizzare come mostrato in figura.

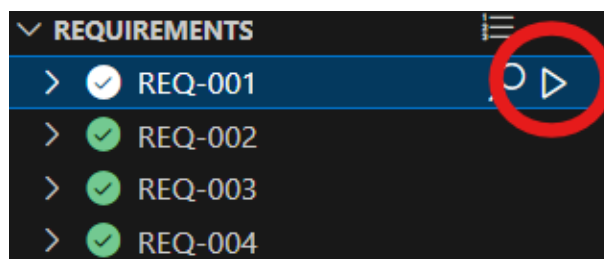


Figure 14: Bottone per l'analisi di un singolo requisito

## 4.8 Analisi dell'implementazione di un singolo requisito

Il plug-in offre la possibilità di analizzare l'implementazione (tracciamento) di un singolo requisito. Per farlo, basta cliccare sull'icona presente vicino al requisito che si desidera analizzare come mostrato in figura.

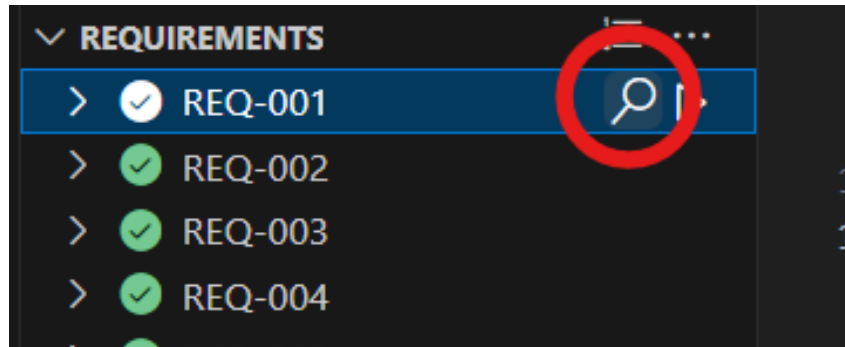


Figure 15: Bottone per l'analisi dell'implementazione di un singolo requisito

## 4.9 Filtraggio dei requisiti

Oltre a poter cercare i requisiti a seconda del loro codice, è possibile filtrare i requisiti in base al loro codice o al loro stato di analisi.

Ci sono quattro diversi modi in cui possono essere filtrati i requisiti:

- In ordine **crescente** (a seconda del loro codice), premendo il pulsante “**Default Order**” mostrato in figura

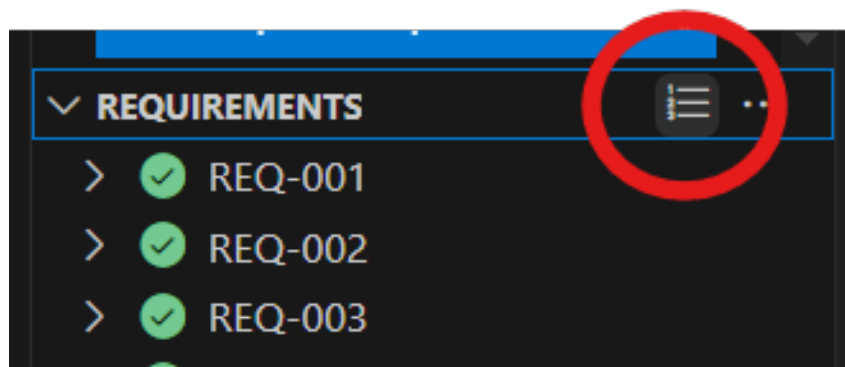


Figure 16: Bottone per organizzare i requisiti in ordine di default

- In ordine “**decrescente**”, premendo il pulsante “**Sort by ID (descending)**”. Mostrato in figura

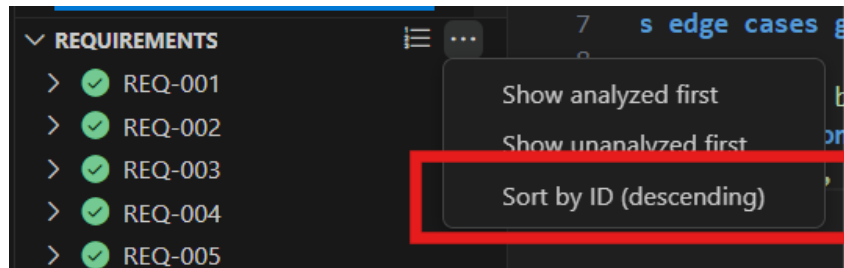


Figure 17: Bottone per organizzare i requisiti in ordine decrescente

- A seconda del loro **stato di analisi**:
  - **Analizzati**, usando il bottone “**Show analyzed first**” mostrato in figura

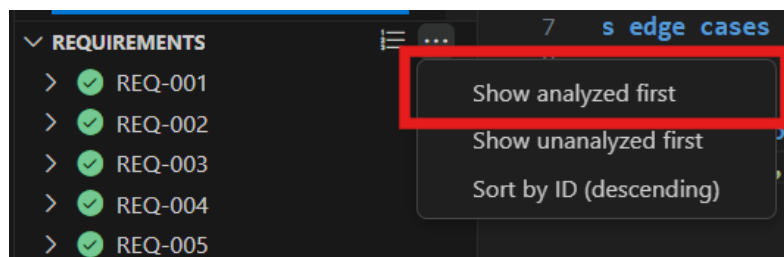


Figure 18: Bottone per mostrare prima i requisiti analizzati

- **Non Analizzati**, usando il bottone “**Show unanalyzed first**” mostrato in figura

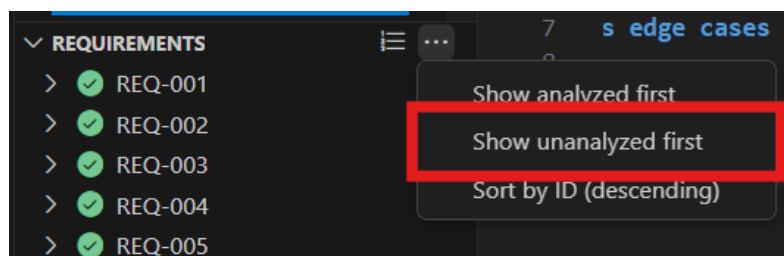


Figure 19: Bottone per mostrare prima i requisiti non analizzati

#### 4.10 Approvazione di un requisito manuale

Il plug-in offre la possibilità di approvare manualmente un requisito. Per farlo, basta cliccare sull'icona presente vicino al requisito che si desidera approvare come mostrato in figura :

## 5 Disinstallazione del plug-in

La disinstallazione del plug-in è molto semplice. Per farlo basta usare l'interfaccia di Visual Studio Code. Andate in basso a sinistra e cliccate sull'icona delle impostazioni a forma di ingranaggio e selezionate **"Extensions"**. Fatto ciò, vi comparirà la lista delle estensioni installate e raccomandate come in figura :

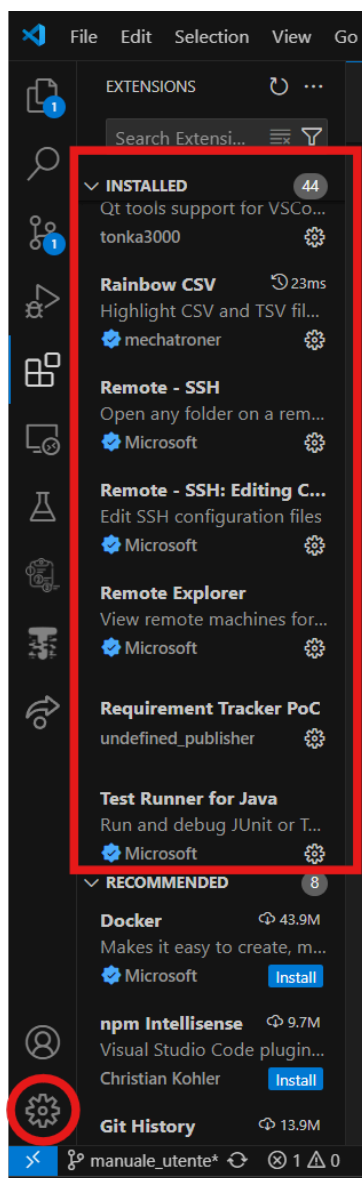


Figure 20: Icona delle estensioni

Quindi, cercate l'estensione **Requirement Tracker** e cliccate sull'icona a forma di ingranaggio vicino ad essa. Vi comparirà un menù a tendina in cui dovrete selezionare **“Uninstall”** come mostrato in figura:

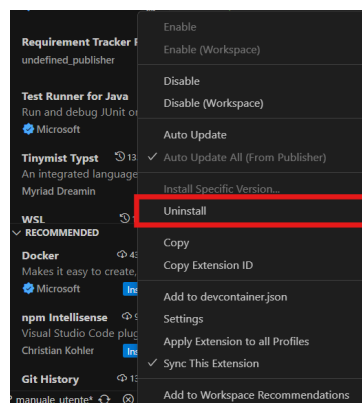


Figure 21: Icona del plug-in

Una volta cliccato Uninstall l'estensione verrà disinstallata e non sarà più visibile nella lista delle estensioni installate.