



Specifica Tecnica

Versione: 0.1.0 18/03/2025

Redattori

Maria Fuensanta Trigueros Hernandez
Marco Perazzolo

Verifica

Approvazione

Uso

Esterno

nextsoftpadova@gmail.com

Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
0.1.0	18/03/2024	Maria Fuensanta Trigueros Hernandez, Marco Perazzolo	Creazione del documento e stesura dei capitoli Introduzione e Tecnologie	

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Riferimenti normativi	4
1.4.2	Riferimenti informativi	4
2	Tecnologie	5
3	Architettura	7

Elenco delle tabelle

Table 1 Tecnologie coinvolte nel progetto	5
---	---

1 Introduzione

1.1 Scopo del documento

L'obiettivo di questo documento è fornire una descrizione dettagliata e completa dell'architettura del prodotto sviluppato. Sono incluse le tecnologie utilizzate e i requisiti necessari per il suo funzionamento.

Il documento presenta la struttura del prodotto attraverso diagrammi delle classi ed esplica alcune funzionalità tramite diagrammi di sequenza. Inoltre, giustifica l'uso di alcuni pattern di progettazione impiegati nell'implementazione

1.2 Scopo del prodotto

Il prodotto, un *plug-in*^G per *Visual Studio Code*^G chiamato "Requirement Tracker Plug-in", è progettato per automatizzare il tracciamento dei *requisiti*^G nei progetti software complessi, con un focus particolare sull'ambito *embedded*^G. L'obiettivo principale è migliorare la qualità e la chiarezza dei requisiti, fornendo suggerimenti basati sull'analisi di un'*intelligenza artificiale*^G, riducendo al contempo i tempi e gli errori legati alla verifica manuale dell'implementazione nel codice sorgente. Il plug-in adotta un'architettura modulare che consente un'estensibilità semplice, rendendolo facilmente adattabile a nuove funzionalità o esigenze future. Inoltre, supporta gli sviluppatori avendo la capacità di utilizzare documenti tecnici come *knowledge*^G, ad esempio datasheet e manuali, in modo da garantire una corretta implementazione dei requisiti.

1.3 Glossario

I termini che potrebbero risultare ambigui sono contrassegnati alla loro prima apparizione con un apice ^G. La loro definizione completa è consultabile nel documento del glossario, in cui sono riportati in ordine alfabetico.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- Capitolato C8 : Requirement Tracker - Plug-in VS Code
 - <https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C8.pdf>
- Norme di Progetto
- Analisi dei Requisiti

1.4.2 Riferimenti informativi

- Progettazione e programmazione: Diagrammi delle classi: <https://www.math.unipd.it/~rcardin/swea/2023/Diagrammi%20delle%20Classi.pdf>

2 Tecnologie

Tecnologia	Versione	Descrizione
Linguaggio		
TypeScript	5.6.3	Linguaggio di programmazione ad alto livello, <i>open-source</i> ^G , che estende JavaScript aggiungendo <i>tipizzazione statica</i> ^G , <i>interfacce</i> ^G e controllo statico degli errori.
CSS	-	Linguaggio di stile utilizzato per descrivere l'aspetto e la formattazione di interfacce web o applicazioni. Nel nostro caso serve a definire lo stile dell'interfaccia del plug-in.
Strumenti		
Visual Studio Code	1.95.0	Editor di codice gratuito sviluppato da Microsoft, utilizzato per scrivere, testare e fare <i>debugging</i> ^G di codice. Permette l'installazioni di estensioni per estendere le funzionalità dell'editor.
npm	10.9.2	Gestore di pacchetti che consente di gestire librerie e <i>dipendenze</i> ^G in JavaScript.
Node.js	20.x	Ambiente di esecuzione JavaScript multiplatforma e open-source, utilizzato per sviluppare applicazioni lato server ed <i>API</i> ^G .
VS Code Extension API	1.95.0	API ufficiale di Visual Studio Code per creare estensioni che interagiscono e si integrano con l'editor.
Git	2.42.0	Git è un sistema di controllo di versione distribuito, utilizzato per la gestione del codice.
Librerie e Framework		
Ollama	0.6.0	<i>Framework</i> ^G open-source che permette di eseguire modelli <i>LLM</i> ^G in locale.
Axios	1.7.9	Libreria JavaScript per effettuare richieste HTTP da browser e Node.js, permette la gestione di richieste asincrone.

Express	4.21.2	Framework per Node.js che permette di creare API REST.
Mocha	10.0.9	Framework di test per Node.js, permette di eseguire test asincroni con TypeScript.
ESLint	9.13.0	Strumento open-source di analisi statica del codice per identificare <i>pattern</i> ^G problematici nel codice TypeScript.

Table 1: Tecnologie coinvolte nel progetto

3 Architettura