



Specifica Tecnica

2025-02-27

V0.0.1

sweetenteam@gmail.com

<https://sweetenteam.github.io>



Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin AzzurroDigitale
Redattori	Orlando Ferazzani
Verificatori	Mouad Mahdi

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
0.0.1	2025-02-27	Orlando Ferazzani	Mouad Mahdi	Prima stesura documento

Indice

1) Introduzione	5
1.1) Scopo del documento	5
1.2) Scopo del prodotto	5
1.3) Miglioramenti e maturità	5
1.4) Glossario	5
1.5) Riferimenti	6
1.5.1) Riferimenti normativi	6
1.5.2) Riferimenti informativi	6
1.5.3) Riferimenti Tecnici	6
2) Tecnologie	6
2.1) Typescript	7
2.2) Langchain	7
2.3) Node.js	7
2.4) Nest.js	7
2.5) GroqCloud	7
2.6) Qdrant	8
2.7) NomicAi	8
2.8) PostgreSQL	8
2.9) Octokit	8
2.10) JiraJs	8
2.11) ConfluenceJs	8
2.12) Docker	9
2.13) React.js	9
2.14) ReactQuery	9
2.15) TailwindCSS	9
2.16) Next.js	9
3) Analisi	10

Lista della immagini

Figura 1	Logo BuddyBot	5
Figura 2	Logo Typescript	7
Figura 3	Logo di Langchain	7
Figura 4	Logo di Node.js	7
Figura 5	Logo di Nest.js	7
Figura 6	Logo di GroqCloud	7
Figura 7	Logo di Qdrant	8
Figura 8	Logo di NomicAi	8
Figura 9	Logo di PostgreSQL	8
Figura 10	Logo di Octokit	8
Figura 11	Logo di JiraJs	8
Figura 12	Logo di ConfluenceJs	9
Figura 13	Logo di Docker	9
Figura 14	Logo di ReactJs	9
Figura 15	Logo di ReactQuery	9
Figura 16	Logo di TailwindCSS	9
Figura 17	Logo di Next.js	9

1) Introduzione

1.1) Scopo del documento

Il presente documento ha lo scopo di fungere da risorsa esaustiva per la spiegazione e conseguente comprensione degli aspetti tecnici del progetto [azzurro digitale](#):



Figura 1: Logo BuddyBot

La sua finalità primaria è quella di fornire una panoramica dettagliata e approfondita delle scelte progettuali, architetturali e tecnologiche del sistema sviluppato. In particolare, si intende fornire un'analisi profonda estesa al livello di progettazione più basso, includendo spiegazione, definizione e motivazione delle scelte effettuate, e dei *design pattern*_G adottati.

Il documento ha quindi scopi molteplici:

- Motivare le scelte progettuali e di sviluppo adottate;
- Fungere da guida per il processo di sviluppo e manutenzione del sistema;
- Fornire una vista panoramica e monitorare la *Code Coverage*_G dei requisiti del progetto identificati nel documento Analisi dei Requisiti (visionabile [qui](#));

L'adeguatezza e la completezza del documento (e del progetto) sono in costante evoluzione e miglioramento in base ai *feedback*_G ricevuti e sulla base dell'aggiornamento dei requisiti.

1.2) Scopo del prodotto

L'obiettivo del progetto è la realizzazione di un *chatbot*_G sotto forma di *Web App*_G atto a fornire un supporto al team di [azzurro digitale](#): nella gestione delle attività di un progetto in corso di sviluppo. Nella fattispecie, il chatbot utilizza delle *API*_G e un modello di *LLM*_G per, rispettivamente, reperire informazioni da sistemi esterni utilizzati dall'azienda (più specificatamente, Jira, GitHub e Confluence) e elaborare una risposta. Questa risposta può contenere del semplice testo, un link o un *code block*_G. Il chatbot ha una singola sessione per ogni utente, e può essere utilizzato da più utenti contemporaneamente.

Il team è confidente che questo genere di prodotto migliorerà il workflow del team di [azzurro digitale](#), riducendo i tempi di risposta e migliorando la qualità del lavoro svolto.

1.3) Miglioramenti e maturità

Questo documento è redatto con approccio incrementale e modificato nel tempo per riflettere l'andamento del progetto e le decisioni prese. In particolare, il documento è soggetto a modifiche in base ai feedback ricevuti e all'evoluzione dei requisiti del progetto. Per questo motivo, il documento non è considerabile definitivo, esaustivo e completo fino al raggiungimento di una versione stabile dello stesso (1.0.0 o superiore).

1.4) Glossario

Per evitare ambiguità e incomprensione riguardanti la terminologia tecnica utilizzata nel documento, viene redatto e adottato un Glossario contenente le definizioni dei termini tecnici utilizzati. Il Glossario è consultabile [qui](#) e i termini presenti nel documento sono evidenziati con *questo stile*_G.

1.5) Riferimenti

1.5.1) Riferimenti normativi

- Presentazione pdf del capitolato C9: [C9p.pdf](#) (*versione disponibile al 2025-03-20*)
- Norme di Progetto: [Norme di Progetto_v1.0.0.pdf](#)
- Piano di Qualifica: [Piano di Qualifica_v1.0.0.pdf](#)

1.5.2) Riferimenti informativi

- Analisi dei Requisiti: [Analisi dei Requisiti_v1.1.0.pdf](#)
- Glossario: [Glossario](#)
- I diagrammi dei casi d'uso: [Use case](#)
- Progettazione: I pattern architetturali [Software Architecture Patterns](#)
- Verifica e validazione: analisi statica (T10): [analisi statica](#)
- Verifica e validazione: analisi dinamica aka testing (T11): [analisi dinamica](#)
- Programmazione: [SOLID programming principles](#)

1.5.3) Riferimenti Tecnici

- Documentazione ufficiale Typescript: [Typescript](#)
- Documentazione ufficiale Langchain: [Langchain](#)
- Documentazione ufficiale NodeJs: [Node.js](#)
- Documentazione ufficiale NestJs: [Nest.js](#)
- Documentazione ufficiale Groq: [GroqCloud](#)
- Documentazione ufficiale Qdrant: [Qdrant](#)
- Documentazione ufficiale NomicAi: [NomicAi](#)
- Documentazione ufficiale PostgreSQL: [PostgresSQL](#)
- Documentazione ufficiale Oktokit: [Oktokit](#)
- Documentazione JiraJs: [JiraJs](#)
- Documentazione Confluence Js: [ConfluenceJs](#)
- Documentazione ufficiale Docker: [Docker](#)
- Documentazione ufficiale ReactJs: [React](#)
- Documentazione ufficiale ReactQuery (TanStack) [ReactQuery](#)
- Documentazione ufficiale TailwindCSS: [Tailwind CSS](#)
- Documentazione ufficiale NextJs [Next.js](#)

2) Tecnologie

In questo capitolo sono elencate tutte le tecnologie della *tech stack_G* che il team utilizza per lo sviluppo del progetto di *azzurro digitale*; come linguaggi di programmazione, *framework_G*, *librerie_G* e *ambienti di sviluppo_G*.

2.1) Typescript

Typescript è un linguaggio di programmazione open-source. È un super-set di JavaScript, che aggiunge forte tipizzazione statica. Il team ha scelto di utilizzare Typescript per la sua tipizzazione statica, che permette di ridurre gli errori di programmazione e di rendere il codice più leggibile e manutenibile.



Figura 2: Logo Typescript

2.2) Langchain

Langchain è un framework open-source per la creazione di applicazioni basate sull'utilizzo LLM. Il team ha scelto di utilizzare Langchain per la sua facilità d'uso e per la sua integrazione con altri servizi come Qdrant e Groq, oltre che ad avere una libreria in Typescript, rendendolo compatibile con il nostro linguaggio.



Figura 3: Logo di Langchain

2.3) Node.js

Node.js è un ambiente di runtime open-source per l'esecuzione di codice JavaScript lato server. Il team ha scelto di utilizzare Node.js per la sua scalabilità e per la sua facilità di utilizzo.



Figura 4: Logo di Node.js

2.4) Nest.js

Nest.js è un framework per la creazione di applicazioni server-side in Node.js. Il team ha scelto di utilizzare Nest.js per la sua struttura modulare e per la sua scalabilità e per la facilità con cui è possibile creare i design pattern più opportuni.



Figura 5: Logo di Nest.js

2.5) GroqCloud

È una piattaforma AI basata su hardware specializzato (LPU) per inferenza ad alte prestazioni, supporta modelli LLM e integrazione con strumenti AI per elaborazione in tempo reale.



Figura 6: Logo di GroqCloud

2.6) Qdrant

Qdrant è un motore di ricerca e analisi di dati non strutturati, supporta l'indicizzazione e la ricerca di dati in tempo reale, oltre che la ricerca di dati basata su vettori.



Figura 7: Logo di Qdrant

2.7) NomicAi

NomicAi è un servizio di elaborazione del linguaggio naturale (NLP) basato su modelli LLM che permette l'embedding di testo. Il team ha scelto di utilizzare NomicAi per la sua facilità d'uso e per la sua integrazione con altri servizi come Langchain e Groq.



Figura 8: Logo di NomicAi

2.8) PostgreSQL

PostgreSQL è un sistema di gestione di database relazionale open-source. Il team ha scelto di utilizzare PostgreSQL per la sua affidabilità e per la sua estensiva documentazione.



Figura 9: Logo di PostgreSQL

2.9) Octokit

Octokit è un toolkit per l'interazione con le API di GitHub. Il team ha scelto di utilizzare Octokit per la sua estesa documentazione e per utilizzare un prodotto ufficiale per interagire on GitHub stesso.



Figura 10: Logo di Octokit

2.10) JiraJs

JiraJs è un toolkit per l'interazione con le API di Jira. Il team ha scelto di utilizzare JiraJs per la sua documentazione affidabile e per la sua facilità d'uso.



Figura 11: Logo di JiraJs

2.11) ConfluenceJs

ConfluenceJs è un toolkit per l'interazione con le API di Confluence. Il team ha scelto di utilizzare ConfluenceJs per la sua documentazione affidabile e per la sua facilità d'uso.



Figura 12: Logo di ConfluenceJs

2.12) Docker

Docker è una piattaforma open-source per lo sviluppo, il deploy e l'esecuzione di applicazioni in container. Il team ha scelto di utilizzare Docker per la sua facilità di deploy e per la sua scalabilità.



Figura 13: Logo di Docker

2.13) React.js

ReactJs è una libreria open-source per la creazione di interfacce utente. Il team ha scelto di utilizzare ReactJs per la sua immediatezza nell'uso, per la sua scalabilità e per la sua estesa documentazione.



Figura 14: Logo di ReactJs

2.14) ReactQuery

ReactQuery è una libreria open-source per la gestione dello stato in React. Il team ha scelto di utilizzare ReactQuery per la sua integrazione con React.



Figura 15: Logo di ReactQuery

2.15) TailwindCSS

TailwindCSS è un framework CSS utilizzato per la creazione di interfacce utente. Il team ha scelto di utilizzare TailwindCSS per la sua facilità d'uso e per la sua documentazione dettagliata oltre che per utilizzare una tecnologia più compatibile con il resto.



Figura 16: Logo di TailwindCSS

2.16) Next.js

Next.js è un framework per la creazione di applicazioni web in React. Il team ha scelto di utilizzare Next.js per i metodi nativi a disposizione per le richieste alle API e per utilizzare una tecnologia più nuova rispetto al resto.



Figura 17: Logo di Next.js

3) Analisi