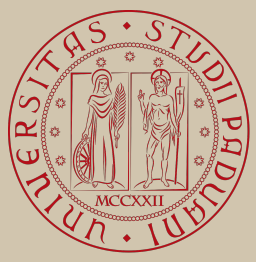
Manuale Utente

2025-04-16

V1.0.0

sweetenteam@gmail.com

https://sweetenteam.github.io



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

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
1.0.0	2025-04-16	Mouad Mahdi	Andrea Santi	Approvazione documento per PB
1.0.1	2025-03-10	Orlando Ferazzani	Mouad Mahdi	Prima stesura del documento

Indice

Lista della immagini

Lista delle tabelle

1) Introduzione

1.1) Scopo del documento

Il presente documento ha lo scopo di dare all'utente le istruzioni per l'installazione e il corretto utilizzo del prodotto. Nello specifico, il documento contiene le istruzioni per installare localmente l'applicazione e utilizzarla nel modo corretto, oltre che i requisiti minimi.

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 azzurrodigitale: 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 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 azzurrodigitale:, 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

- Glossario: Glossario

2) Requisiti

In questo breve capitolo sono elencati i requisiti minimi per l'installazione e l'utilizzo del prodotto. I requisiti sono divisi in due categorie: requisiti hardware e requisiti di browser.

2.1) Requisiti hardware

Dato che non sono stati specificati requisiti hardware da capitolato o da progetto, i seguenti requisiti sono stati decisi dal team di sviluppo e sono da considerarsi sufficienti per l'installazione e l'utilizzo del prodotto:

Componente	Requisito
CPU	2,5GHz Dual Core o superiore
RAM	8GB DDR4 o superiore
Connessione	Connessione ad internet stabile
Sistema Operativo	Windows 10 o superiore, Linux, MacOS

Tabella 1: Requisiti hardware

2.2) Requisiti di browser

Dato che il prodotto è una *Web App*$_G$, è necessario un browser per l'utilizzo. I requisiti di browser sono stati decisi dal team di sviluppo e sono da considerarsi sufficienti per l'installazione e l'utilizzo del prodotto:

Browser	Versione
Google Chrome	135.0.7049.42 o superiore
Mozilla Firefox	137.0.1 o superiore
Microsoft Edge	134.0.3124.83 o superiore
Safari	18.3 o superiore

Tabella 2: Requisiti di browser

3) Installazione

In questo paragrafo verrà spiegato come installare il prodotto sulla propria macchina in modo da poterne usufruire in locale. Si ricorda che il progetto è stato concepito per essere consegnato al proponente e che venga fatto operare sui loro server dedicati. L'installazione in locale è da considerarsi un'operazione non necessaria e non richiesta, ma è stata comunque implementata per facilitare lo sviluppo e il testing del prodotto.

3.1) Strumenti e tecnologie necessarie

- *Brew*$_G$ se la propria macchina è un sistema UNIX based: un gestore di pacchetti per macOS e Linux, che permette di installare facilmente software e librerie. È possibile installarlo con il comando:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Codice 1: Installazione di Brew

- *Choco*$_G$ se la propria macchina è un sistema Windows based: un gestore di pacchetti per Windows, che permette di installare facilmente software e librerie. È possibile installarlo aprendo il terminale in modalità amministratore e copiando il comando:

```
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -
InputFormat None -ExecutionPolicy Bypass -Command
"[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps
1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
```

Codice 2: Installazione di Choco

- *Docker*$_G$: un software che permette di eseguire applicazioni in container, isolando le dipendenze e le librerie necessarie per il loro funzionamento. È possibile installarlo con il comando:

```
brew install docker
```

Codice 3: Installazione di Docker su macOS e Linux

oppure

```
choco install docker-desktop
```

Codice 4: Installazione di Docker su Windows

- Git: un sistema di controllo versione distribuito, che permette di tenere traccia delle modifiche apportate al codice sorgente. È possibile installarlo con il comando:

```
brew install git
```

Codice 5: Installazione di Git su macOS e Linux

oppure

```
choco install git
```

Codice 6: Installazione di Git su Windows

3.2) Creazione delle API Key

Per utilizzare il prodotto, è necessario creare delle API Key per i servizi esterni utilizzati dal chatbot. Le API Key sono delle chiavi univoche che permettono di autenticarsi e accedere ai servizi esterni. Per creare l'Api key, è necessario creare un account per ogni servizio, e navigare nella pagina dedicata alle API Key nelle impostazioni dell'account. Di seguito sono riportati i link per creare le API Key per i servizi esterni utilizzati dal chatbot:

- https://nomic.ai
- https://console.groq.com/keys
- https://www.atlassian.com/software/jira
- https://www.atlassian.com/software/confluence
- https://github.com/settings/tokens

3.3) Installazione del prodotto

Per installare il prodotto, è necessario clonare il repository Git del progetto e installare le dipendenze necessarie. La cartella del prodotto è scaricabile anche in formato `.zip` da Github e può essere scompattata in una cartella a piacere. In tal caso, non è necessario Git. Per installare il prodotto, è necessario eseguire i seguenti comandi:

1. Aprire il terminale e navigare nella cartella in cui si desidera installare il prodotto (o navigare nella cartella in cui è stato scompattato il file `.zip`) con il comando:

```
cd /percorso/della/cartella
```

Codice 7: Navigazione nella cartella del prodotto

2. Clonare il repository Git del progetto con il comando (passaggio opzionale se si è scaricato il progetto da GitHub):

```
git clone git@github.com:SweeTenTeam/BuddyBot.git
```

Codice 8: Clonazione del repository Git

3. Navigare nella cartella del progetto con il comando:

```
cd BuddyBot
```

Codice 9: Navigazione nella cartella del progetto

4. A partire dai file `.env.example` presenti nella cartella del progetto, è necessario creare i file `.env`. I file `.env` contengono le variabili di ambiente necessarie per il corretto funzionamento del prodotto. Per farlo è possibile eseguire il comando:

```
cp .env.example .env
```

Codice 10: Creazione dei file .env

Una volta fatto, basta inserire le API Key create in precedenza nei file `.env` e il prodotto sarà pronto per essere utilizzato. Questo passaggio va ripetuto per ogni microservizio del prodotto.

1. Impostare la repositorye la branch da tracciare nel file `.env` del microservizio `apiGateway`;
2. Far partire il container di Docker con il seguente comando se è la prima volta:

```
docker-compose up --build
```

Codice 11: Build e Avvio del container di Docker

In caso contrario, è possibile eseguire il comando:

```
docker-compose up
```

Codice 12: Avvio del container di Docker

A questo punto, basterà aprire il browser preferito e recarsi all'indirizzo `http://localhost:3000` per visualizzare l'applicazione.

4) Istruzioni per l'uso

Il prodotto permette un'interazione con il chatbot tramite un'interfaccia grafica semplice e intuitiva. In questo capitolo verranno fornite le istruzioni per l'utilizzo del prodotto, con particolare attenzione alle funzionalità principali e alle modalità di interazione con il chatbot.

All'avvio dell'applicazione, l'utente si troverà di fronte a questa schermata:

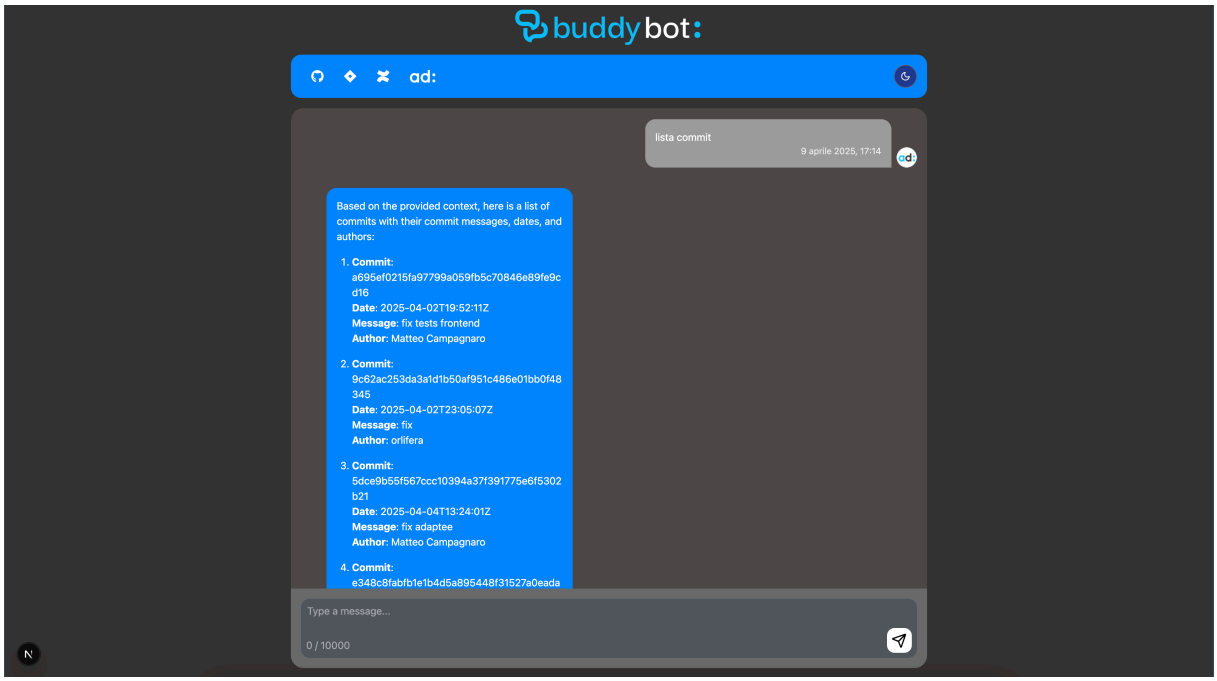


Figura 1: Schermata principale dell'applicazione

Nel caso in cui invece sia il primo avvio, e non ci siano messaggi nello storico, il messaggio visualizzato sarà questo:

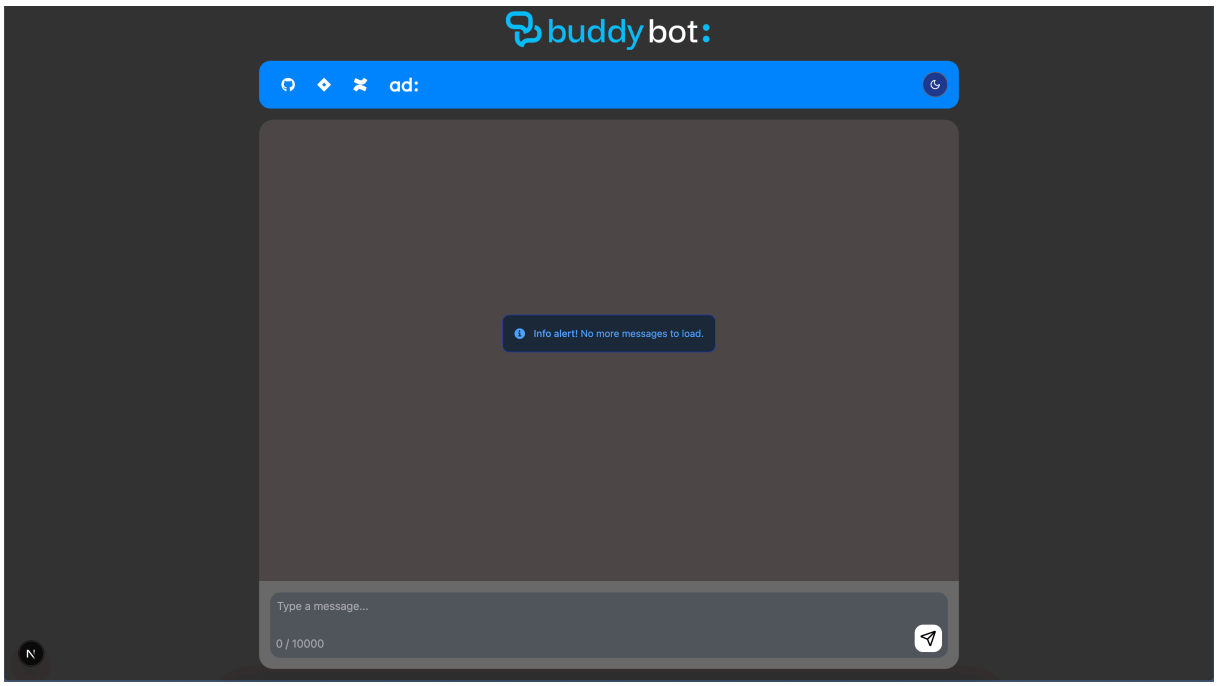


Figura 2: Messaggio di storico vuoto

L'utente può interagire con il chatbot tramite la barra di input presente nella parte inferiore della schermata. Per inviare un messaggio, è sufficiente digitare il testo desiderato e premere il tasto «Invio» sulla tastiera. In alternativa, è possibile fare clic sul pulsante a destra della barra di input.

Figura 3: barra di input

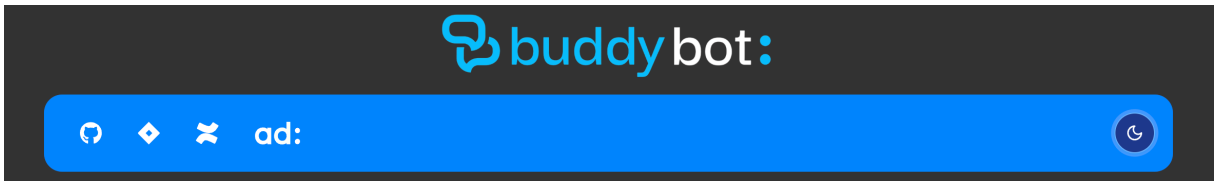


Figura 4: barra di input

Nella parte sinistra della Navbar, si possono trovare 4 icone, le prime tre per raffigurare i servizi terzi con cui il chatbot si interfaccia. In particolare, queste icone sono link diretti alle documentazioni di ciascun servizio. La quarta icona invece rappresenta il sito (o dashboard interna) del proponente.

Figura 5: link barra di input

L'applicazione prende come tema di default quello specificato dalle impostazioni di sistema dell'utente, tuttavia è possibile modificare tale impostazione con cliccando l'apposito bottone nella parte destra della navbar.

Figura 6: Bottone per il cambio tema

4.1) Errori

In caso di errore durante il fetch della cronologia dei messaggi, l'utente verrà avvisato con un messaggio di errore.

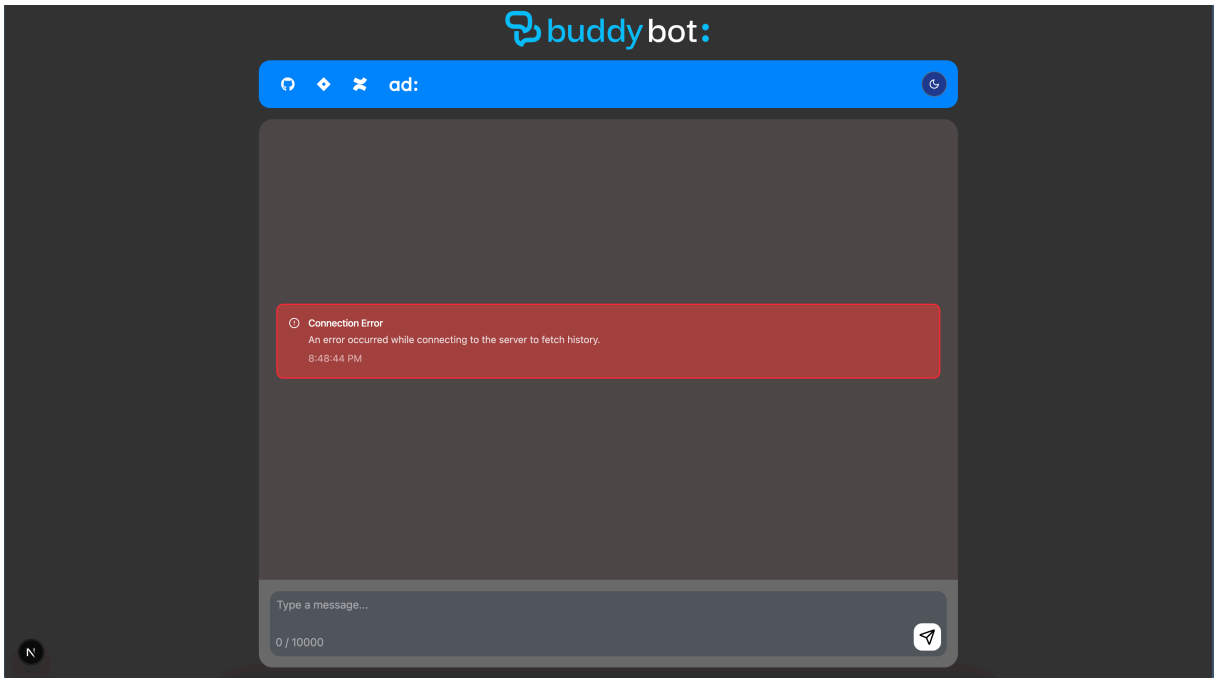


Figura 7: Messaggio di errore

Nel caso in cui il server non risponda entro un certo intervallo di tempo, l'utente verrà avvisato con un messaggio di errore, nello specifico, TimeOut error.

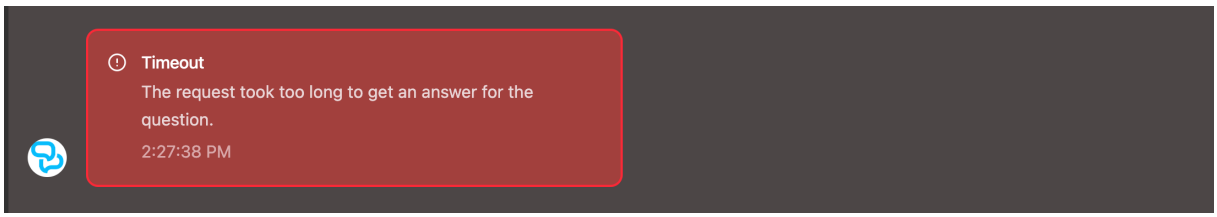


Figura 8: Messaggio di errore

Nella parte inferiore sinistra dell'area di input è presente contatore di caratteri, che mostra il numero di caratteri inseriti. Se il numero di caratteri supera il limite massimo, il contatore diventa rosso e il messaggio non viene inviato. Il limite massimo è di 10000 caratteri.

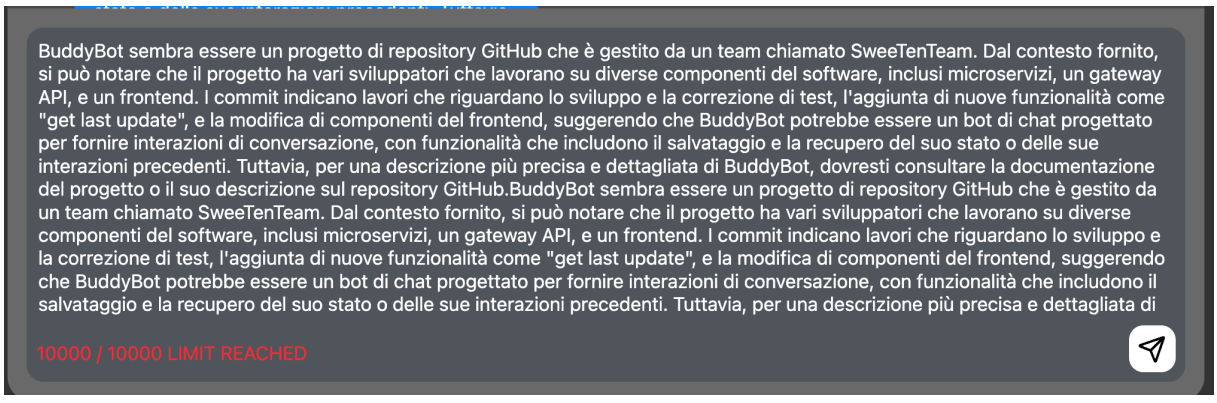


Figura 9: Messaggio di errore

Nel caso in cui il sistema generi una risposta troppo lunga, viene generato un messaggio di errore che informa l'utente dell'accaduto. In tal caso, l'utente può provare a riformulare la domanda in modo da ottenere una risposta più breve. TOFIX

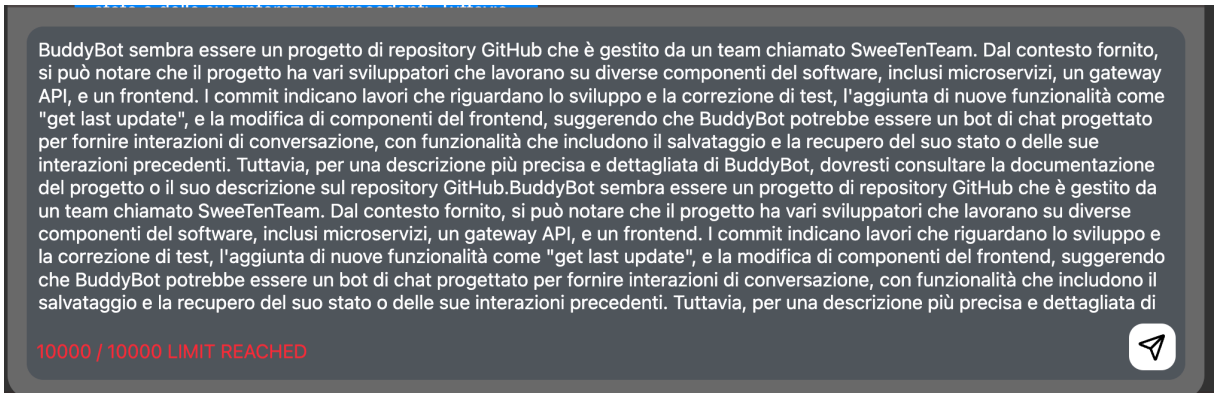


Figura 10: Messaggio di errore