



**Università degli Studi di Padova**

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025



**Gruppo: SWEg Labs**

Email: [gruppo.sweg@gmail.com](mailto:gruppo.sweg@gmail.com)

# Verbale Riunione

06 Marzo 2025

## Indice

<b>1</b>	<b>Informazioni generali</b>	<b>1</b>
<b>2</b>	<b>Ordine del giorno</b>	<b>2</b>
<b>3</b>	<b>Diario della riunione</b>	<b>3</b>
<b>4</b>	<b>Decisioni</b>	<b>5</b>
<b>5</b>	<b>Todo</b>	<b>6</b>

## 1 Informazioni generali

- **Tipo di riunione:** esterna
- **Luogo:** piattaforma Google Meet
- **Data:** 06/03/2025
- **Ora inizio:** 17:00
- **Ora fine:** 17:50
- **Responsabile:** Riccardo Stefani
- **Scriba:** Riccardo Stefani
- **Partecipanti:**
  - Michael Fantinato
  - Giacomo Loat
  - Filippo Righetto
  - Riccardo Stefani
  - Federica Bolognini
  - Davide Verzotto
  
  - AzzurroDigitale:*
    - Nicola Boscaro
    - Martina Daniele

## 2 Ordine del giorno

- Presentare al *proponente*<sub>G</sub> *AzzurroDigitale*<sub>G</sub> i nostri obiettivi di funzionalità per l'*MVP*<sub>G</sub> aggiornati rispetto all'ultimo incontro;
- Illustrare le funzionalità che non verranno implementate nell'*MVP*;
- Chiedere consigli riguardo la gestione della modifica e dell'eliminazione dei documenti nel *database vettoriale*<sub>G</sub> *Chroma*<sub>G</sub>;
- Chiedere consigli riguardo la gestione della ricreazione della connessione verso i *database*<sub>G</sub>;
- Chiedere consigli riguardo la gestione dell'errore di salvataggio dei log nel *database relazionale*<sub>G</sub> *Postgres*<sub>G</sub>;
- Illustrare i risultati ottenuti per quanto riguarda la visualizzazione dei log mediante file di testo e richiedere l'approvazione della stessa;
- Chiedere come gestire il fatto che, mentre il database vettoriale sta venendo aggiornato, il chatbot non è in grado di rispondere;
- Illustrare i risultati ottenuti per quanto riguarda il *frontend*<sub>G</sub>, richiedere l'approvazione della grafica ottenuta e di alcune possibili aggiunte;
- Esporre i risultati ottenuti per quanto riguarda i *test*<sub>G</sub> e domandare quanta percentuale e quali tipologie di *coverage*<sub>G</sub> siano necessarie;
- Spiegare perchè siamo avanzati poco con la documentazione e descrivere quanto faremo a riguardo nella prossima *Sprint*<sub>G</sub>.

### 3 Diario della riunione

- Sono stati presentati al *proponente* i nostri obiettivi di funzionalità per l'*MVP* aggiornati rispetto all'ultimo incontro, ovvero:
  - Salvataggio e recupero dello storico di sessione;
  - Proporre domande per proseguire la conversazione;
  - Visualizzazione del badge dell'esito dell'aggiornamento automatico;
  - Visualizzazione di un txt con i log dell'aggiornamento automatico;
  - Migliorare l'aggiornamento automatico per gestire l'eliminazione di file;
  - Migliorare l'aggiornamento automatico dei file che subiscono modifiche;
  - Visualizzazione dei link dei file da cui il bot ha preso la risposta;
  - Pulsante di copia messaggio e pulsante di copia snippet di codice.
- Sono state presentate al *proponente* le funzionalità che non verranno implementate nell'*MVP*, ovvero:
  - Proporre domande per iniziare la conversazione;
  - Integrazione delle *API<sub>G</sub>* verso *Telegram<sub>G</sub>* e *Slack<sub>G</sub>*.
- Abbiamo chiesto consiglio al *proponente* riguardo la gestione della modifica dei documenti nel *database vettoriale Chroma*, e ci è stato consigliato di provare a prelevare la data di ultima modifica del documento dalla sua piattaforma di provenienza (es.: *GitHub<sub>G</sub>*) e quindi di confrontarla con la data di inserimento del documento nel *database vettoriale*: se la data di modifica è più recente, allora il documento deve essere aggiornato, altrimenti no;
- Abbiamo chiesto consiglio al *proponente* riguardo la gestione dell'eliminazione dei documenti nel *database vettoriale Chroma*, e ci è stato consigliato, per maggiore efficienza, di prelevare solamente gli id dei documenti presenti in *Chroma* per confrontarli con gli id dei documenti prelevati dalle piattaforme, senza prelevare i documenti interi, come avevamo implementato inizialmente;
- Abbiamo chiesto consiglio al *proponente* riguardo la gestione della ricreazione della connessione verso i *database*, e ci è stato riferito che nel mondo aziendale si utilizzano dei *framework<sub>G</sub>* che gestiscono la connessione in modo automatico, e che quindi non è necessario gestire manualmente la ricreazione della connessione. Per il nostro progetto, la questione è stata valutata come non necessaria;
- Abbiamo chiesto consiglio al *proponente* riguardo la gestione dell'errore di salvataggio dei log nel *database relazionale Postgres* da parte del *cron<sub>G</sub>*, in particolare a causa del fatto che, in caso di tale fallimento, *Angular<sub>G</sub>* potrebbe recuperare dal database i log di due volte prima, e quindi non i log più recenti. Il *proponente* ci ha esposto tre possibilità di gestione del consenso dei dati tra quanto salva il *cron* e quanto recupera *Angular*:
  - *WebSocket<sub>G</sub>*: il *cron* invia i log ad *Angular* tramite *WebSocket* ed *Angular* così risulta aggiornato in tempo reale senza passare per un database. Tuttavia, questa soluzione è stata valutata come troppo complessa per il nostro progetto;
  - *Polling REST<sub>G</sub>*: come già avviene attualmente, il *cron* ed *Angular* agiscono in modo autonomo, cioè *Angular* recupera i log dal database ogni tot tempo; a questo punto, utilizzando particolari framework, è possibile garantire il consenso dei dati tra i due;
  - Verificare lo stato dell'aggiornamento automatico solamente al momento di apertura o refresh della scheda del *browser<sub>G</sub>*: è stato infatti constatato assieme al proponente che è molto improbabile che l'utente mantenga aperta la scheda del *browser* per un tempo sufficientemente lungo da causare problemi di sincronizzazione tra il *cron* ed *Angular*, e quindi è possibile anche questa terza opzione perchè *Angular* ottenga l'esito dell'ultimo aggiornamento.

Mentre la gestione degli errori è stata valutata come requisito non necessario, le ultime due possibilità di sincronizzazione del badge del *frontend<sub>G</sub>* sono state valutate come equivalenti per il proponente, e quindi la scelta è stata lasciata a noi;

- Abbiamo mostrato al proponente il file di testo dove sono stati stampati i log nella nostra implementazione attuale, e ci è stato detto che la visualizzazione è accettabile. Inoltre, ci è stata lasciata carta bianca per il salvataggio dei log come pila o come coda;
- Abbiamo mostrato al proponente il nostro secondo file per il salvataggio dei log, `cron.log`, spiegando qual è il rapporto di quest'ultimo con il file di testo: il file `txt` contiene il log consuntivo di ogni tentativo di aggiornamento automatico, mentre il file `cron.log` consente di monitorare l'operato del cron, poichè trascrive tutto ciò che viene stampato nello standard output della console durante la sua esecuzione. Il proponente ha approvato questa soluzione di spartizione delle responsabilità tra i due file;
- Abbiamo chiesto al proponente come gestire il fatto che, mentre il database vettoriale sta venendo aggiornato, il chatbot non è in grado di rispondere, e ci è stato consigliato di riportare il fenomeno nel documento *Manuale Utente<sub>G</sub>*;
- Abbiamo mostrato al proponente la prima bozza del *frontend*, e ci è stato detto che la grafica è accettabile, e che possiamo procedere seguendo tale direzione. Per i link ai documenti da cui è stata ricavata la risposta, è stata approvata la proposta di fare un riquadro in fondo al messaggio, con un link cliccabile per ogni documento. Ci è stato inoltre approvato l'inserimento della data e ora di invio nei messaggi;
- Abbiamo riportato al *proponente* i risultati attuali per quanto riguarda i test: abbiamo fino ad ora svolto i *test di integrazione<sub>G</sub>*, che ci hanno permesso di ottenere una *coverage* delle righe del 60% per il *backend<sub>G</sub>*, e del 70% per il *frontend*. Abbiamo poi chiesto al proponente quanta coverage sia richiesta, e quali tipologie di coverage siano necessarie. Il proponente ci ha risposto che ci farà sapere in seguito. Abbiamo inoltre comunicato che nella prossima *sprint* ci occuperemo di sviluppare i *test di unità<sub>G</sub>*;
- Abbiamo chiesto al *proponente* a quanto ammonta in termini di denaro il nostro utilizzo della *API Key<sub>G</sub>* di *OpenAI<sub>G</sub>* allo stato attuale, e ci è stato risposto che il consumo finora è stato abbastanza basso, e che quindi possiamo continuare ad utilizzarla in modo normale per le nostre necessità;
- Abbiamo infine riportato al *proponente* che siamo avanzati poco con la documentazione a causa del fatto che nella scorsa *sprint* abbiamo puntato quasi esclusivamente sulla progettazione e sulla programmazione, e abbiamo promesso che nella prossima *sprint* riprenderemo i lavori con la parte documentale: in particolare, riporteremo i *Diagrammi delle classi<sub>G</sub>* nel documento di *Specifica Tecnica<sub>G</sub>*.

## 4 Decisioni

Durante la riunione sono state prese le seguenti decisioni:

Codice	Descrizione
VE 10.1	E' stato deciso che, per la modifica dei file in <i>Chroma</i> , si preleverà la data di ultima modifica del documento dalla sua piattaforma di provenienza, e la si confronterà con la data di inserimento del documento nel database vettoriale per capire se procedere con una sostituzione o meno.
VE 10.2	E' stato deciso che, per l'eliminazione dei documenti in <i>Chroma</i> , si preleveranno solamente gli id dei documenti presenti nel <i>database vettoriale</i> per confrontarli con gli id dei documenti prelevati dalla piattaforma, senza prelevare i documenti interi, per ragioni di efficienza.
VE 10.3	E' stato deciso che non è necessario gestire manualmente la ricreazione della connessione verso i database.
VE 10.4	E' stato deciso che la gestione dell'errore di salvataggio dei log nel database relazionale <i>Postgres</i> da parte del <i>cron</i> non è richiesto.
VE 10.5	E' stato deciso che, per trasmettere i log dal <i>cron</i> ad <i>Angular</i> , viene lasciata libera scelta a noi tra <i>Polling REST</i> e verifica dello stato dell'aggiornamento automatico solamente all'apertura o refresh della finestra del <i>browser</i> .
VE 10.6	E' stato deciso che la visualizzazione dei log è accettabile, che siamo liberi di implementare il salvataggio degli stessi come pila o come coda a nostra discrezione, e che le responsabilità del file txt e del file cron.log sono state scelte in modo oculato.
VE 10.7	E' stato deciso che bisogna riportare nel documento <i>Manuale Utente</i> il fatto che, mentre il database vettoriale sta venendo aggiornato, il chatbot non è in grado di rispondere.
VE 10.8	E' stato deciso che la grafica del <i>frontend</i> è accettabile, che la soluzione gradita per la visualizzazione dei link verso le fonti è inserire un riquadro dedicato in fondo al messaggio di risposta, e che è una buona aggiunta la visualizzazione della data e ora di invio nei messaggi.
VE 10.9	E' stato deciso che possiamo continuare ad utilizzare la <i>API Key</i> di <i>OpenAI</i> in modo normale per le nostre necessità.
VE 10.10	È stato deciso che Riccardo Stefani ha il compito di redigere il verbale della riunione esterna del 06/03/25 e di inviarlo per mail assieme alle slide della riunione convertite in PDF.

## 5 Todo

Durante la riunione sono emersi i seguenti task da svolgere:

Codice	Dalla decisione	Assegnatario	Task Todo
BUD-385	VE 10.10	Riccardo Stefani	Convertire le slide della riunione odierna in PDF ed inviarle per mail ad <i>Azzurro-Digitale</i> assieme al verbale esterno.
BUD-384	VE 10.10	Riccardo Stefani	Redigere il verbale della riunione esterna del 06/03/25.

