

SWEetCode

---

2023-10-27

# Valutazione capitolati

## Versione

---

v.0.0

## Componenti del gruppo

---

Bresolin G.  
Campese M.  
Ciriolo I.  
Dugo A.  
Feltrin E.  
Michelon R.  
Orlandi G.



# 1 C-1. Knowledge Management AI

- **Proponente:** AzzurroDigitale S.r.l.
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

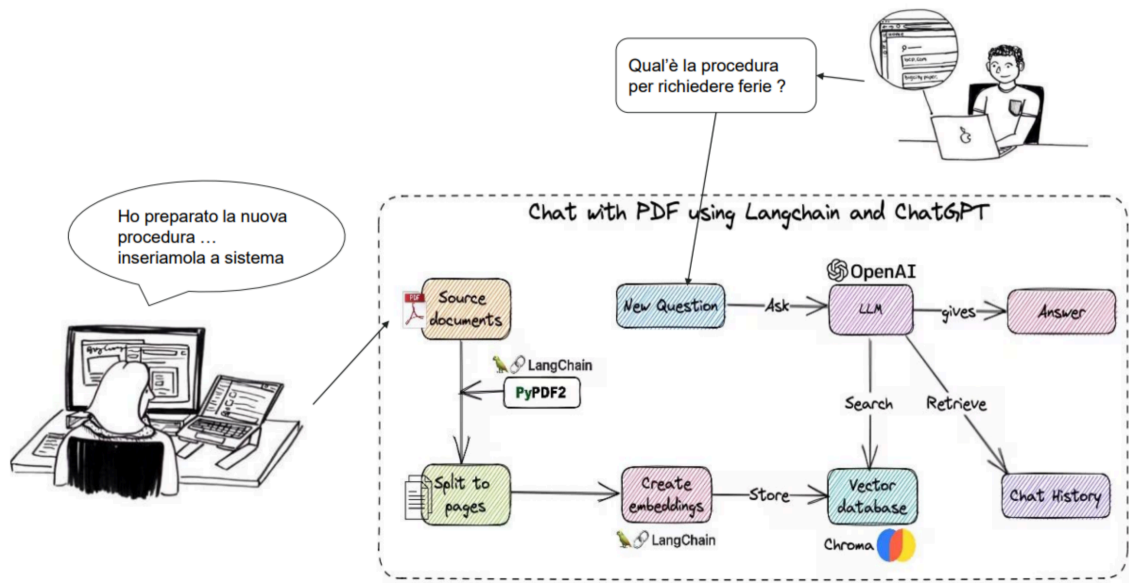
## 1.1 Riassunto

Con l'avanzamento dell'Intelligenza Artificiale, sono emerse nuove soluzioni che offrono un approccio innovativo per problemi di knowledge management.

Rispetto a tecniche manuali di archiviazione, le soluzioni basate sull'AI riescono a estrarre e strutturare i dati automaticamente, identificando anche relazioni nascoste tra di essi. Un sistema di questo tipo potrebbe essere uno strumento utile a operatori e dipendenti per reperire in maniera semplice ed efficiente informazioni specifiche riguardo manuali, regolamenti e indicazioni aziendali.

Volendo sfruttare questa opportunità, l'azienda AzzurroDigitale propone di sviluppare una piattaforma web per la gestione (caricamento, consultazione, eliminazione) dei documenti e un chatbot basato sull'intelligenza artificiale, il cui compito è quello di rispondere a richieste fatte dagli utenti basandosi sui dati contenuti nei documenti caricati nel sistema.

Di seguito è illustrato uno schema generale del funzionamento del sistema.



## 1.2 Obiettivo

Creare un sistema di Knowledge Management basato sull'AI per catturare, organizzare e distribuire le conoscenze all'interno di un'organizzazione. È richiesta la consegna del seguente materiale:

- Documentazione tecnica: diagrammi di flusso, schemi di funzionamento, documentazione delle API;
- Bug reporting;
- Codice sorgente, raccolto in una repository git messa a disposizione dall'azienda.



### 1.3 Dominio tecnologico

- *Node.js*, runtime system per l'esecuzione di codice *Javascript*;
- *Angular*, framework open-source *Typescript* per sviluppare la UI front-end;
- *OpenAI*, motore per le funzionalità di NLP (Natural Language Processing), tramite il quale viene elaborata la richiesta e generata una risposta opportuna;
- *LangChain*, framework open-source che semplifica la comunicazione con modelli di AI, permette di collegare altre fonti di dati e offre funzionalità come document analysis, summarization, code analysis e l'implementazione di chatbot;
- *Chroma*, un vector store open-source usato per la memorizzazione e il recupero di word embedding (o word vector) in modo efficiente.

### 1.4 Aspetti Positivi

- Argomento e tecnologie interessanti, sempre più utilizzate e richieste nel mondo del lavoro;
- Progetto che ha uno scopo ben preciso e delineato, con fini socialmente utili;
- Ottime tecnologie, supportate da una grande community;
- Flessibilità nella scelta delle tecnologie da utilizzare.

### 1.5 Aspetti Negativi

- Presentazione non molto dettagliata che omette alcuni dettagli implementativi.

### 1.6 Conclusioni

Sin da subito il capitolato proposto da AzzurroDigitale è stato apprezzato dall'intero gruppo e il successivo incontro effettuato con l'azienda ha confermato l'interesse. Per queste ragioni oltre che agli aspetti positivi precedentemente citati, dopo un'attenta votazione, il gruppo ha deciso di scegliere C-1 come prima preferenza.



## 2 C-2. Sistema di Raccomandazione

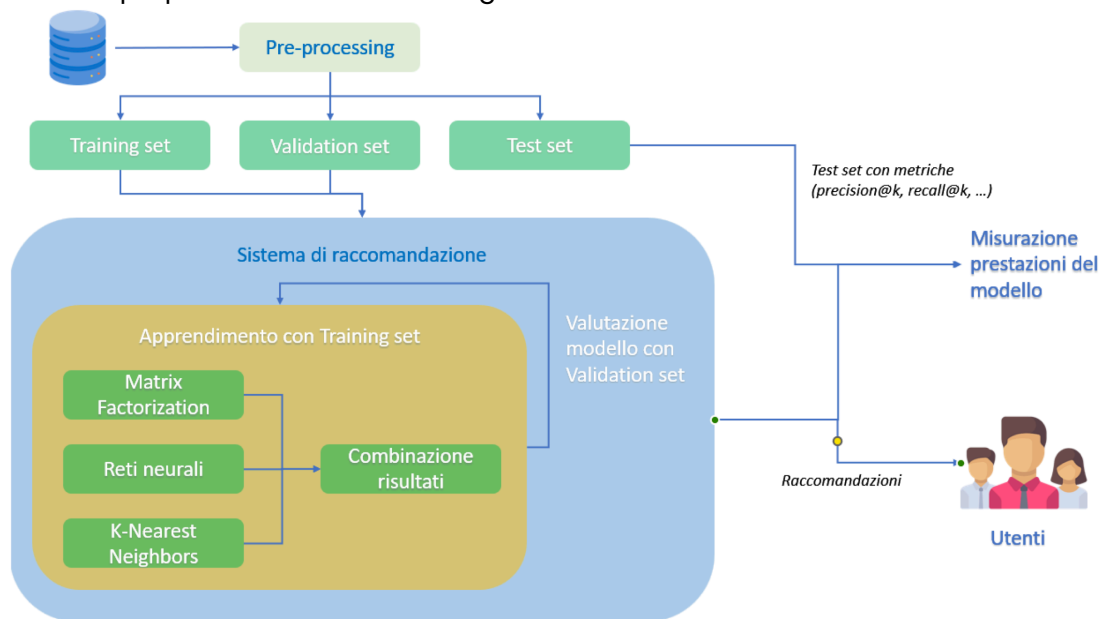
- **Proponente:** Ergon Informatica S.r.l.
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 2.1 Riassunto

Il progetto in questione è applicabile a imprese il cui principale focus aziendale è la commercializzazione di prodotti. Queste aziende eseguono campagne di marketing e piani commerciali per promuovere prodotti o incrementare le vendite adottando principalmente due strategie: una ad ampio spettro ed una mirata.

Grazie al possesso e l'analisi dei dati storici dei clienti, l'azienda sarà in grado di raggrupparli in base a similitudini e scoprire interessi non evidenti dai loro acquisti passati. Il progetto mira a creare un sistema che consigli all'azienda come indirizzare specifiche iniziative di marketing e commerciali, identificando gruppi di clienti target. Il sistema di raccomandazione utilizzerà feedback che potranno essere espliciti o impliciti. La sua implementazione invece potrà avvenire in uno dei seguenti modi: collaborative filtering oppure content-based filtering.

Il sistema proposto è riassunto nel seguente schema.



### 2.2 Obiettivo

Realizzare un Sistema di Raccomandazione basato sull'Intelligenza Artificiale per orientare le attività aziendali, suggerendo su quali clienti dovrebbero essere mirate le iniziative di marketing. In particolare la piattaforma da sviluppare dovrà essere formata dai seguenti componenti:

- Database relazionale per la gestione dei dati;
- Motore di raccomandazione;
- Interfaccia utente, UI desktop-based o web-based, per la consultazione dei risultati e ritorno di feedback degli utenti.



## 2.3 Dominio tecnologico

- *SQL Server Express, MySql o MariaDB*;
- *ML.NET* (basato su framework *.NET* utilizzando il linguaggio *C#*) o *Surprise* (libreria *Python*).

## 2.4 Aspetti Positivi

- libertà di scelta per le tecnologie da utilizzare;
- Il materiale informativo proposto è molto approfondito e l'incontro con l'azienda ha dimostrato una buona conoscenza nell'ambito tecnologico proposto;
- Proposta di formazione da parte dell'azienda.

## 2.5 Aspetti Negativi

- Tecnologie proposte hanno un basso learning rate.

## 2.6 Conclusioni

Sebbene le buone impressioni legate al capitolo siano state confermate dall'incontro, le tecnologie proposte ci hanno fatto virare verso altre opzioni. Dopo una fase di votazione, il gruppo ha deliberato di designare questo capitolo come terza scelta.



### 3 C-3. EasyMeal

- **Proponente:** Imola Informatica
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

#### 3.1 Riassunto

EasyMeal semplifica radicalmente il processo di prenotazione nel settore dei ristoranti, consentendo agli utenti di riservare un tavolo in modo intuitivo e rapido. Gli utenti inoltre potranno creare il proprio ordine da casa o dall'ufficio, personalizzandolo in base alle proprie esigenze, allergie e preferenze alimentari. EasyMeal offre anche un'esperienza di ordine coinvolgente e collaborativa. I clienti infatti potranno interagire direttamente con lo staff del ristorante per chiedere informazioni sulle pietanze e monitorare gli ordini degli altri membri del loro tavolo. Attraverso l'app si potrà anche dividere il conto tra i partecipanti, semplificando ulteriormente la gestione delle spese nei gruppi.

#### 3.2 Obiettivo

Sviluppare un'applicazione web responsive (EasyMeal) per semplificare il processo di prenotazione e ordinazione delle pietanze direttamente da remoto. Applicazione web responsive (PC, IOS e Android) con:

- Sistema di registrazione utenti (registrazione, login e logout);
- Consultare la lista di ristoranti con sistema di filtraggio per data, orario, città e tipologia di cucina;
- Collegamento di una prenotazione ad un ristorante e all'ordinazione creata dagli utenti di un tavolo;
- Chat privata tra utente e amministratore del ristorante;
- Notifiche agli utenti e agli amministratori dei ristoranti;
- Interazione con lo staff del ristorante;
- Pagamento di un conto;
- Consultazione delle prenotazioni da parte di un amministratore del ristorante;
- Inserimento di feedback e recensioni;
- Notifiche per amministrazione e utenti;
- Sistema di pagamento.

#### 3.3 Dominio tecnologico

Assenti.

#### 3.4 Aspetti Positivi

- L'azienda non propone nessuna tecnologia da utilizzare, lasciando totale libertà nella scelta implementativa;
- Funzionalità che prese singolarmente non risultano complesse.

#### 3.5 Aspetti Negativi

- Progetto con molte funzionalità e casi d'uso, la cui implementazione risulta complessa;



- L'argomento non presenta niente di innovativo.

### 3.6 Conclusioni

La proposta non ha suscitato interesse da parte dei componenti del gruppo, quindi non è stata inserita tra le possibili scelte del gruppo.



## 4 C-4. ChatGPT plugin con Nuvolaris serverless

- **Proponente:** Nuvolaris
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 4.1 Riassunto

Il progetto che l'azienda propone si basa sulla creazione di un plugin per una delle loro produzioni software, chiamato *Nuvolaris*, un servizio per lo sviluppo in cloud serverless e *FaaS* (Function-as-a-Service). *FaaS* è un tipo di servizio di cloud computing basato su eventi ed eseguito in container stateless che consente agli sviluppatori di creare, eseguire e gestire i pacchetti applicativi come funzioni, senza dover mantenere un'infrastruttura propria. Tramite *Nuvolaris* è possibile creare API e plugin scalabili, cioè capaci di aumentare o diminuire di scala in funzione delle necessità e disponibilità.

### 4.2 Obiettivo

Sviluppare un plugin di *ChatGPT* per costruire app di una specifica categoria, ad esempio CRUD app con SQL database, eCommerce app, forms come *Google Forms*, e molto altro. In particolare, vengono presentate le seguenti richieste:

- Creare un template per un tipo di applicazione utilizzando file di configurazione;
- API per modificare e aggiornare il file di configurazione;
- Sistema per ricostruire l'applicazione basandosi sul file di configurazione. L'app deve essere live dopo ogni richiesta e deve essere generata usando funzioni serverless.

### 4.3 Dominio tecnologico

- *Nuvolaris*
- *ChatGPT API*

### 4.4 Aspetti Positivi

- L'azienda fornisce materiali utili come *Nuvolaris* environment in Cloud, un account *ChatGPT Pro*, documentazione relativa alle tecnologie proposte, supporto tramite *Slack*.

### 4.5 Aspetti Negativi

- Complesso da progettare idealmente;
- Proposta non chiara, soprattutto la descrizione del loro prodotto;
- Obbligo di utilizzo di una tecnologia proprietaria, sconosciuta al team;
- Non è ben specificato cosa intendono con template di app e che tipo di informazioni dovrebbe contenere un file di configurazione.

### 4.6 Conclusioni

Proposta non interessante per il gruppo, non considerata nelle preferenze.





## 5 C-5. WMS3: warehouse management 3D

- **Proponente:** Sanmarco Informatica SPA
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 5.1 Riassunto

Un Warehouse Management System (WMS) ha come obiettivo principale la tracciabilità e il controllo dello stoccaggio nei magazzini, supervisionando il loro flusso, gestendo tutte le operazioni, dal ricevimento dei beni alla loro spedizione o all'utilizzo nei relativi reparti di produzione. Questo tipo di sistema è progettato per assicurare che le richieste di prelievo vengano evase puntualmente e per massimizzare l'efficiente utilizzo degli spazi fisici nel magazzino. I principali obiettivi di un sistema WMS sono il controllo delle performance, la razionalizzazione dei costi e l'ottimizzazione dei processi di logistica.

Per fare ciò è preferibile implementare un'esperienza 3D che rappresenta il magazzino e tutti i suoi reparti, con lo scopo di massimizzare la visibilità e migliorare l'esperienza utente.

Di seguito alcune raccomandazioni esplicite suggerite da Sanmarco Informatica SPA:

- Nessun login;
- Sessioni volatili (no persistenza delle modifiche, quali l'ultima posizione visionata);
- Solo gestione "amministratore";
- Non è richiesta l'integrazione con sistemi di VR;
- Contatti frequenti gruppo di lavoro – proponente.

### 5.2 Obiettivo

Gestire correttamente il magazzino come chiave per raggiungere l'efficienza dei flussi e dei processi aziendali. In particolare, sono state presentate le seguenti richieste:

- Sviluppare un'applicazione che offra un'area 3D in cui sia possibile aggiungere manualmente scaffalature e modificarle a piacimento;
- Fornire la possibilità di codificare le coordinate della scaffalatura in righe e colonne per identificare le aree disponibili in cui posizionare gli oggetti;
- Consentire agli utenti di selezionare un prodotto (oggetto all'interno della scaffalatura) e richiederne lo spostamento in un'area diversa;
- Consentire il caricamento di un'immagine SVG da utilizzare come base della struttura;
- Implementare il precaricamento del layout del magazzino da un database;
- Offrire la possibilità di posizionare gli oggetti in coordinate specifiche o di evidenziare un prodotto cercato.

### 5.3 Dominio tecnologico

- *Three.js (Javascript)* (consigliate);
- *Unity (C#), Unreal Engine (C++)* (alternative)



#### 5.4 Aspetti Positivi

- Documentazione proposta è visivamente molto curata, con esempi visuali che illustrano una possibile implementazione della proposta.

#### 5.5 Aspetti Negativi

- Tecnologie complesse che non sono risultate interessanti a tutti i componenti del gruppo;
- Non molto chiare le modalità di inserimento e spostamento del materiale di stoccaggio.

#### 5.6 Conclusioni

L'argomento e l'ambito di applicazione del progetto non sono stati fonte di motivazione ed interesse per i membri del gruppo.



## 6 C-6. SyncCity (Smart City Monitoring Platform)

- **Proponente:** Sync Lab S.r.l.
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 6.1 Riassunto

Sync Lab propone di sviluppare un sistema che riesca a simulare nel modo più realistico possibile i dati dei sensori di misurazione di una città, rappresentandoli in una serie di dashboard. I dati dovranno essere simulati attraverso tool di generazione di informazioni random, ma qualora ci fosse l'opportunità di raccogliarli da fonti esterne reali, l'azienda potrebbe discuterne l'utilizzo.

La città monitorata potrebbe essere suddivisa in celle e per ognuna di esse potrebbero essere prese varie sorgenti di dati, come ad esempio: informazioni atmosferiche, inquinamento, traffico, lavori di manutenzione, guasti, criticità strutturali e molto altro.

La piattaforma dovrà raccogliere questi dati e salvarli su una piattaforma *OLAP* (On-Line Analytical Processing), permettendo agli utenti di visualizzarli in dashboard dedicate.

### 6.2 Obiettivo

Realizzare una piattaforma e relativa dashboard che consenta a chi la usa di avere sotto controllo una serie di informazioni sullo stato di salute della città, in modo da prendere decisioni veloci, efficaci ed analizzare poi gli effetti conseguenti.

### 6.3 Dominio tecnologico

- *Faker* di *Python*, framework per la simulazione dei dati;
- *Apache Kafka*, broker per disaccoppiare lo stream di informazioni provenienti dai diversi simulatori;
- *ClickHouse*, database *OLAP* che mantiene la persistenza di dati con numerosità elevata;
- *Grafana*, piattaforma di data visualization per la creazione di diagrammi e grafici, raccolti in dashboard.

### 6.4 Aspetti Positivi

- Dominio tecnologico proposto considerato molto interessante e attuale.

### 6.5 Aspetti Negativi

- Integrazione complessa tra tutte le tecnologie proposte;
- La proposta di simulare la generazione dei dati ricevuti dal sistema, seppur presentata come semplificazione, ha rimosso l'entusiasmo iniziale del gruppo.

### 6.6 Conclusioni

Nonostante la proposta originale e il dominio applicativo sempre più richiesto, il gruppo ha optato per altri progetti ritenuti complessivamente più interessanti.



## 7 C-7. ChatGPT vs BedRock developer Analysis

- **Proponente:** Sync Lab S.r.l.
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 7.1 Riassunto

Zero12 propone di creare un middleware che riceva in input dei requisiti di business e produca epic e user stories associate ai requisiti di business, comparando la capacità di *ChatGPT* e quella di *AWS BedRock* nell'interpretare del codice sorgente ed associare le user stories generate. Il team dovrà svolgere un'analisi sulle API di *ChatGPT* e *BedRock*, studiandone un approccio modulare che permette un uso alternativo dei due sistemi di Generative AI.

### 7.2 Obiettivo

- Creare un plugin per *Visual Studio Code*;
- Creare un plugin per *Apple Xcode*;
- Confrontare i risultati forniti dal sistema utilizzando *ChatGPT* e *BedRock*.

### 7.3 Dominio tecnologico

- *AWS Fargate*: un servizio serverless per la gestione di container;
- *MongoDB*: un database documentale ideale per la gestione di progetti basati su eventi;
- *Node.js*: utilizzato per lo sviluppo di API REST in formato JSON;
- *Python*: utilizzato come plugin per *Xcode*;
- *TypeScript*: utilizzato come plugin per *Visual Studio Code (VSCode)*;
- *API ChatGPT*: una interfaccia di programmazione applicativa per l'accesso al modello di linguaggio GPT-3;
- *AWS Bedrock*: un framework o servizio di AWS utilizzato nel progetto.

### 7.4 Aspetti Positivi

- Il progetto risulta interessante e le tecnologie proposte sono state gradite dal gruppo;
- L'azienda è sembrata molto disponibile, offrendo varie proposte di supporto, tra cui una sessione di design thinking, seguita da guide sulla creazione dei requisiti di business, formazione sull'uso di tecnologie cloud *Amazon Web Services*, sessioni tecniche di review del lavoro svolto;
- Offerta di fornitura di repository *GIT*, *Jira*, crediti i e account *ChatGPT*.

### 7.5 Aspetti Negativi

- Rispetto alla creazione di un intero sistema, la creazione di plugin non ha suscitato interesse ad alcuni componenti del gruppo;
- Non sono chiari i criteri di confronto dei due sistemi di AI.



## 7.6 Conclusioni

L'interesse per questo progetto non era condiviso da tutti i componenti del gruppo. Dopo una fase di votazione, il gruppo ha deciso di abbandonare la proposta.



## 8 C-8. JMAP

- **Proponente:** Zextras
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 8.1 Riassunto

Il prodotto principale dell'azienda proponente è *Carbonio*, un software per la collaborazione online on-premise che viene rilasciato come pacchetto installabile nell'infrastruttura server del cliente. Attualmente, Carbonio basa il proprio servizio email su vari protocolli standard, tra cui *IMAP*, *POP* e *Exchange ActiveSync*. Tuttavia, questi protocolli presentano diverse limitazioni, tra cui:

- Non sono estendibili, il che vincola le funzionalità di Carbonio a ciò che offrono questi standard;
- Le funzionalità del calendario e dei contatti non sono erogate tramite un protocollo specifico, ma attraverso chiamate non standard basate su protocollo SOAP;
- L'uso di licenze *ActiveSync* richiede il pagamento di onerosi diritti a *Microsoft*.

Il protocollo di posta elettronica open-source *JMAP*, a differenza dei protocolli citati precedentemente, è stato concepito con l'obiettivo di ottimizzare lo scambio di dati tra client e server, migliorando le prestazioni delle applicazioni attraverso richieste e risposte ottimizzate. *JMAP* utilizza il formato JSON e definisce un set di API che permette ai client di accedere ai dati e gestirli in modo efficiente. Sfruttando i meccanismi di sincronizzazione efficienti su cui si basa, *JMAP* consente a più client di comunicare con il server in modo che le modifiche vengano visualizzate simultaneamente. Inoltre, *JMAP* è progettato per essere estendibile, consentendo l'aggiunta di nuove funzionalità senza la necessità di modificare il protocollo di base.

### 8.2 Obiettivo

Capire se ha senso estendere *Carbonio* aggiungendo la possibilità di utilizzare il protocollo *JMAP*, mantenendo i vecchi standard per compatibilità con i client attualmente supportati, ma permettendo all'azienda di espandere le funzionalità offerte ai client di nuova generazione, offrendo anche uno standard implementativo alla versione *Open Source* di *Carbonio*. L'azienda propone quindi di sviluppare un servizio di demo testabile con uno dei client pubblicizzati nel sito web ufficiale di [JMAP](#). Questo servizio verrà poi utilizzato internamente a Zextras (assieme agli stress test che opzionalmente verranno forniti dal gruppo) per valutare le prestazioni, la manutenibilità e la completezza del protocollo *JMAP*, paragonandolo agli attuali protocolli attualmente sviluppati in *Carbonio*. Seguono i requisiti funzionali esposti dall'azienda:

- Invio e la ricezione di una mail;
- Gestione delle cartelle;
- Gestione dei contenuti di una cartella;
- Eliminazione di un oggetto;
- Eliminazione di una cartella;
- Condivisione di una cartella;
- Eliminazione di una condivisione di cartella.



### 8.3 Dominio tecnologico

Nell'implementazione del protocollo *JMAP* per il servizio *Carbonio*, sono state stabilite le seguenti specifiche tecniche aggiuntive:

- Linguaggio a scelta del gruppo di lavoro, con preferenza per *Java* (essendo il linguaggio principale dello stack tecnologico di Carbonio);
- Libreria *INPUTmice/JMAP* per l'implementazione del protocollo *JMAP*. Alternativamente nel caso si scelga di optare per un linguaggio diverso, utilizzare l'implementazione presente in uno dei progetti elencati qui *JMAP Software Implementations* per sviluppare la propria soluzione;
- Eseguitibile in un sistema container, come *Docker*. Questo ci permette di poter lanciare in maniera molto semplice una o più istanze del servizio, permettendoci di eseguire in batteria test di funzionalità e di performance;
- Deve essere scalabile mediante l'inizializzazione di più nodi stateless. Per stateless si intende che alla richiesta di uno specifico client fatta ad un'architettura contenente più di un'istanza del servizio dato, può rispondere una qualsiasi istanza del servizio, perché nessuna istanza contiene dati specifici di stato rispetto alle richieste dei client.

### 8.4 Aspetti Positivi

- Proposta originale che permette di esplorare campi tecnologici non profondamente conosciuti dal gruppo;
- Prodotto di utilizzo reale e utile in ambito applicativo per l'azienda.

### 8.5 Aspetti Negativi

- Non chiaro il metodo di integrazione del protocollo *JMAP* con il loro prodotto interno Carbonio

### 8.6 Conclusioni

Il capitolato in questione è stato scartato poiché tutti i membri del gruppo hanno manifestato opinioni negative a riguardo.



## 9 C-9. ChatSQL

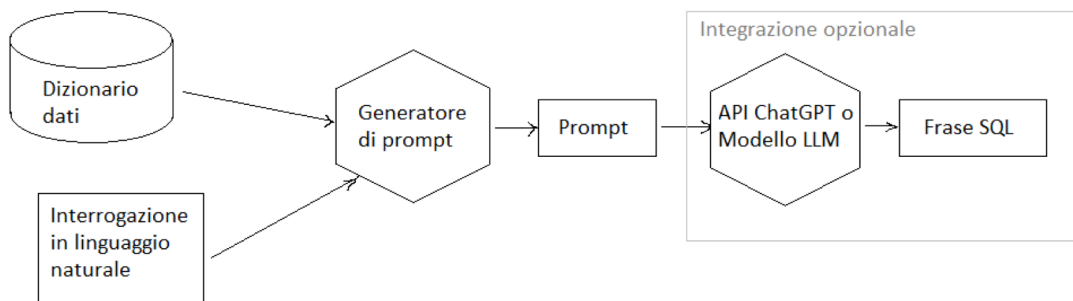
- **Proponente:** Zucchetti
- **Committenti:** Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 9.1 Riassunto

Il “prompt engineering” è un processo di strutturazione del testo interpretato da modelli generativi di Intelligenza Artificiale. Attraverso questo processo è possibile formulare dei prompt che guidano i modelli di AI nella produzione di risposte di maggiore qualità e coerenza tramite la scelta della sintassi, della struttura e del contesto, descritti in linguaggio naturale. Il problema proposto consiste nella scrittura di frasi SQL a partire dalla definizione della struttura del database e della specifica della frase in linguaggio naturale. In particolare, l'azienda chiede di sviluppare una applicazione che svolga i seguenti compiti:

- Archiviazione della descrizione della struttura di un database, possibilmente commentata in tutte le sue parti. Questo può essere fatto sia tramite informazioni strutturate, che attraverso documenti di testo;
- Maschera di richiesta di una frase di interrogazione del database in linguaggio naturale, realizzabile attraverso un semplice campo di testo;
- Procedura che combina la richiesta di interrogazione con le informazioni della struttura del database creando un “prompt” che, sottoposto ad un sistema di AI, fornisce l'interrogazione equivalente al linguaggio naturale in linguaggio SQL.

I tre punti precedenti devono essere integrati in un unico sistema che permetta di utilizzarli in modo integrato. L'intero sistema può essere riassunto dal seguente schema.



Requisiti opzionali:

- Visualizzare la frase SQL prodotta dal sistema di AI;
- Sviluppare l'input vocale della frase in linguaggio naturale;
- Verificare la correttezza della frase SQL prodotta dal sistema di AI tramite batterie di testing;
- Implementare la gestione di più basi di dati, archiviando diversi scenari e scegliendo quello opportuno nel momento in cui avviene una richiesta;
- Utilizzare modelli LLM alternativi a *ChatGPT* (Es: *Palm*, *LLaMa*, *Falcon*);
- Addestrare un modello in modo specifico per la traduzione di frasi di interrogazione in italiano a frasi SQL.





## 9.2 Obiettivo

Realizzare un sistema che, a partire da frasi in linguaggio naturale che descrivono un database e un'interrogazione da effettuare su di esso, generi dei prompt opportuni tramite i quali un LLM sarebbe in grado di generare query o comandi SQL che soddisfano le richieste dell'utente.

## 9.3 Dominio tecnologico

- *ChatGPT* e strumenti per il suo fine tuning;
- *GitHub* per la pubblicazione del progetto, con i relativi requisiti di natura open-source, in modo da favorire la continuità del prodotto risultante.

## 9.4 Aspetti Positivi

- Proposta molto allettante, argomenti ultimamente molto discussi e tecnologie sempre più utilizzate e richieste nell'ambito lavorativo;
- Dominio tecnologico non vincolante.

## 9.5 Aspetti Negativi

- L'efficacia dei prompt generati potrebbe non soddisfare le aspettative a causa del non determinismo di LLM;
- Integrazione complessa tra le varie possibili tecnologie da utilizzare.

## 9.6 Conclusioni

Il capitolo, sebbene abbia alcuni aspetti negativi, grazie anche alla possibilità di lavorare con l'azienda Zucchetti, ha ottenuto una buona valutazione da parte dei componenti del gruppo. Per questo motivo abbiamo deciso di porlo come seconda scelta.