



BugBusters

Email: bugbusters.unipd@gmail.com

Gruppo: 4

Università degli Studi di Padova

Laurea in Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/2026

Resoconto Capitolati

Versione 0.5.0

Redattori	Alberto Autiero, Linor Sadè
Verificatori	[Nome Cognome]
Uso	Interno
Destinatari	Prof. Tullio Vardanega, Prof. Riccardo Cardin
Data	20/10/2025

Abstract

Documento di analisi e valutazione dei capitolati proposti per l'anno accademico 2025/2026. Il documento include una valutazione dettagliata del capitolato scelto e un'analisi comparativa degli altri capitolati disponibili.

1 Versioni del documento

Versione	Data	Descrizione	Redatto	Verificato	Approvato
0.5.0	23/10/2025	Stesura di C6	Linor Sadè	-	-
0.4.1	22/10/2025	Correzione indice	Linor Sadè	-	-
0.4.0	22/10/2025	Iniziato e terminato C7	Linor Sadè	-	-
0.3.1	22/10/2025	Correzioni minori stile	Linor Sadè	-	-
0.3.0	21/10/2025	Terminato C5, iniziato C8, C9	Linor Sadè	-	-
0.2.0	21/10/2025	Aggiunta introduzione al documento, aggiunti punti di forza e debolezza per ogni capitolato	Linor Sadè	-	-
0.1.0	20/10/2025	Prima stesura della struttura del documento	Alberto Autiero	Marco Favero	-

Indice

1	Versioni del documento	2
2	Metodologia di valutazione	5
3	Elenco capitolati analizzati	5
3.1	C1 - Automated EN18031 Compliance Verification	5
3.1.1	Breve descrizione	5
3.1.2	Requisiti funzionali e non funzionali	5
3.1.3	Tecnologie proposte	5
3.1.4	Chiarimenti e colloqui con l'azienda	5
3.1.5	Interesse del team	5
3.1.6	Punti di forza e di debolezza	5
3.2	C2 - Code Guardian	6
3.2.1	Breve descrizione	6
3.2.2	Requisiti funzionali e non funzionali	6
3.2.3	Tecnologie proposte	6
3.2.4	Chiarimenti e colloqui con l'azienda	6
3.2.5	Interesse del team	6
3.2.6	Punti di forza e di debolezza	6
3.3	C3 - DIPReader	7
3.3.1	Breve descrizione	7
3.3.2	Requisiti funzionali e non funzionali	7
3.3.3	Tecnologie proposte	7
3.3.4	Chiarimenti e colloqui con l'azienda	7
3.3.5	Interesse del team	7
3.3.6	Punti di forza e di debolezza	7
3.4	C4 - L'app che Protegge e Trasforma	8
3.4.1	Breve descrizione	8
3.4.2	Requisiti funzionali e non funzionali	8
3.4.3	Tecnologie proposte	8
3.4.4	Chiarimenti e colloqui con l'azienda	8
3.4.5	Interesse del team	8
3.4.6	Punti di forza e di debolezza	8
3.5	C5 - Nexum	9
3.5.1	Breve descrizione	9
3.5.2	Requisiti funzionali e non funzionali	9
3.5.3	Tecnologie proposte	9
3.5.4	Chiarimenti e colloqui con l'azienda	10
3.5.5	Interesse del team	11
3.5.6	Punti di forza e di debolezza	11
4	C6 - Second Brain	12
4.1	Breve descrizione	12
4.2	Requisiti funzionali e non funzionali	12
4.3	Tecnologie proposte	12
4.4	Chiarimenti e colloqui con l'azienda	13
4.5	Interesse del team	13
4.6	Punti di forza e di debolezza	13

5	C7 - Sistema di acquisizione dati da sensori	13
5.1	Breve descrizione	13
5.2	Requisiti funzionali e non funzionali	14
5.3	Tecnologie proposte	15
5.4	Chiarimenti e colloqui con l'azienda	15
5.5	Interesse del team	17
5.6	Punti di forza e di debolezza	17
5.7	C8 - Smart Order	18
5.7.1	Breve descrizione	18
5.7.2	Requisiti funzionali e non funzionali	18
5.7.3	Tecnologie proposte	19
5.7.4	Chiarimenti e colloqui con l'azienda	19
5.7.5	Interesse del team	19
5.7.6	Punti di forza e di debolezza	19
5.8	C9 - View4Life	21
5.8.1	Breve descrizione	21
5.8.2	Requisiti funzionali e non funzionali	21
5.8.3	Tecnologie proposte	21
5.8.4	Chiarimenti e colloqui con l'azienda	21
5.8.5	Interesse del team	21
5.8.6	Punti di forza e di debolezza	21

2 Metodologia di valutazione

Per la valutazione dei capitoli proposti, il team ha adottato una metodologia strutturata che prevede i seguenti passaggi:

1. **Discussione:** Dopo le presentazioni delle aziende, il team si è trovato “a caldo” per discutere di ciascun capitolo esaminando l’interesse verso ciascun progetto (propensione, curiosità, motivazione) e le competenze richieste (tecnologie note, tecnologie da apprendere).
2. **Analisi individuale:** Ogni membro del team ha condotto un’analisi individuale di ogni capitolo, leggendone il relativo documento di proposta, valutandone più attentamente i punti di forza e di debolezza in base a criteri predefiniti.
3. **Richiesta di chiarimenti:** In caso di dubbi o incertezze riguardo a specifici aspetti dei capitoli, il team ha preparato una lista di domande da inviare ai proponenti per ottenere chiarimenti. In seguito sono stati fissati incontri con le aziende per discutere questi punti (se necessario). In particolare ci interessava capire:
 - Il livello di supporto che l’azienda fornirà durante lo sviluppo del progetto.
 - L’importanza nella conoscenza pregressa delle tecnologie richieste.
 - ALTRO DA DEFINIRE
4. **Compilazione tabelle:** Il team si è diviso il compito di compilare le tabelle dei punti di forza e di debolezza per ogni capitolo, assicurando che ogni membro contribuisse in modo equo.

I punti 3 e 4 sono stati ripetuti più volte fino a quando il team non si è sentito sufficientemente preparato per prendere una decisione informata sul capitolo da scegliere.

Il team ha deciso di dare particolare importanza all’interesse verso l’argomento e le tecnologie rispetto alla complessità del progetto, senza tuttavia trascurare la valutazione della sua fattibilità. Crediamo che lavorare su un progetto che suscita entusiasmo e curiosità porterà a un’esperienza di apprendimento più significativa e gratificante, anche se ciò comporta affrontare sfide tecniche più impegnative.

3 Elenco capitoli analizzati

Di seguito sono elencati i capitoli analizzati dal team BugBusters. Per ogni capitolo, sono riportati i punti di forza e di debolezza identificati durante la fase di valutazione.

3.1 C1 - Automated EN18031 Compliance Verification

3.1.1 Breve descrizione

3.1.2 Requisiti funzionali e non funzionali

3.1.3 Tecnologie proposte

3.1.4 Chiarimenti e colloqui con l’azienda

3.1.5 Interesse del team

3.1.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none"> • Requisiti Ben Definiti: I requisiti obbligatori e opzionali sono elencati in modo chiaro e strutturato. Questo riduce l'ambiguità e fornisce un'ottima checklist per la pianificazione e la verifica finale del progetto. • Caso Studio Concreto: La presenza di un caso studio specifico (la macchina da caffè connessa via Wi-Fi) fornisce un dominio applicativo tangibile per testare le funzionalità, evitando di lavorare in modo troppo astratto. • Bluewind si impegna esplicitamente per un supporto "ibrido" (online e in presenza), con incontri periodici. Questo è un enorme vantaggio, poiché fornisce un canale diretto per chiarire dubbi, ottenere feedback e allinearsi con le aspettative dell'azienda. 	<ul style="list-style-type: none"> • Complessità del Dominio: Il dominio normativo (EN 18031, direttiva RED) è intrinsecamente complesso. Comprendere appieno la logica dei "Decision Tree" e le interdipendenze tra i requisiti richiederà uno sforzo iniziale significativo di analisi e studio.

3.2 C2 - Code Guardian

3.2.1 Breve descrizione

3.2.2 Requisiti funzionali e non funzionali

3.2.3 Tecnologie proposte

3.2.4 Chiarimenti e colloqui con l'azienda

3.2.5 Interesse del team

3.2.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none"> • Tema Innovativo e ad Alto Potenziale: L'uso di un'architettura multi-agente per l'analisi automatizzata del codice è estremamente attuale e all'avanguardia. • Dominio Concreto e Utile: La piattaforma risolve problemi reali di qualità del codice, sicurezza e manutenzione dei repository. • Sessione di mentoring sulle tecnologie che verranno utilizzate 	<ul style="list-style-type: none"> • Alta Complessità Concettuale: L'architettura multi-agente è concettualmente avanzata. Progettare un sistema dove agenti specializzati comunicano efficacemente attraverso un orchestratore richiede una solida comprensione di pattern complessi. • Requisiti di Testing Stringenti: La richiesta del 70% di test coverage (obbligatorio) è apprezzabile professionalmente ma può essere impegnativa da raggiungere in un progetto universitario, specialmente per componenti AI. • Non sembra esserci flessibilità nella scelta delle tecnologie da utilizzare

3.3 C3 - DIPReader

3.3.1 Breve descrizione

3.3.2 Requisiti funzionali e non funzionali

3.3.3 Tecnologie proposte

3.3.4 Chiarimenti e colloqui con l'azienda

3.3.5 Interesse del team

3.3.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none"> • Dominio Specializzato e di Alto Valore: La conservazione digitale è un campo di nicchia ma cruciale, specialmente in ambito legale e amministrativo. • Problema Concreto e Ben Definito: L'esigenza di accedere a documenti conservati digitalmente in modalità offline è un requisito reale per molti professionisti. Il progetto risolve un problema tangibile. • Interazione con l'azienda: offre esempi di pacchetti estratti dal sistema di conservazione e la relativa documentazione 	<ul style="list-style-type: none"> • Complessità del Dominio Normativo: La conservazione digitale è regolata da standard e normative complesse. Comprendere appieno il formato dei DIP e i requisiti di compliance richiederà uno sforzo iniziale significativo. • Sfide Tecniche per l'Offline: Implementare ricerche efficienti e visualizzazioni di anteprima completamente offline, specialmente per grandi volumi di dati, presenta sfide non banali di performance e gestione della memoria. • Ambiguità Architetturale: Non è completamente chiaro se l'applicazione debba essere una PWA, un'app desktop (Electron) o entrambe. Questa decisione avrà impatti significativi sull'architettura. • Scope Potenzialmente Ampio: Le funzionalità opzionali come la ricerca semantica (con FAISS) e la verifica delle firme digitali sono progetti ambiziosi che potrebbero distrarre dallo sviluppo del MVP. • Testing su Grandi Volumi: Garantire le performance con "grandi volumi" di dati potrebbe essere difficile da testare e validare in ambiente universitario.

3.4 C4 - L'app che Protegge e Trasforma

3.4.1 Breve descrizione

3.4.2 Requisiti funzionali e non funzionali

3.4.3 Tecnologie proposte

3.4.4 Chiarimenti e colloqui con l'azienda

3.4.5 Interesse del team

3.4.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none"> • Impatto Sociale Elevatissimo: Il progetto ha uno scopo nobile e concretamente utile - prevenire e supportare vittime di violenza di genere. Sviluppare un'app che può potenzialmente salvare vite fornisce una motivazione etica molto forte. • Supporto Aziendale Eccezionale e Strutturato: Miriade offre un supporto completo: o Referenti specializzati per ogni area (tecnica, design, dominio sociale) o Formazione iniziale sulla tematica della violenza di genere o Strumenti professionali (Jira, Bitbucket) o Supporto multidisciplinare continuo o Possibilità di incontri in sede • Requisiti Chiari e Dettagliati: Sia i requisiti funzionali che non funzionali sono ben specificati, con una chiara distinzione tra obbligatori e opzionali. • Ottima possibilità riguardo il ciclo di vita dell'applicazione: analisi, progettazione, sviluppo, test di sicurezza e controllo dei contenuti etici devono essere fatti in modo chiaro e preciso, apprendendo così competenze multidisciplinari. 	<ul style="list-style-type: none"> • Alta Complessità Tecnica e Progettuale: o L'architettura proposta è ambiziosa e forse eccessiva essendo per alcuni il primo affronto verso certe conoscenze o L'integrazione di AI/ML per l'analisi comportamentale richiede competenze specialistiche o L'architettura serverless su AWS con microservizi è complessa da gestire o La sicurezza dei dati è critica e richiede implementazioni robuste • Responsabilità e Sensibilità del Dominio: L'errore in un'app di questo tipo può avere conseguenze gravi, soprattutto dal punto di vista etico. La progettazione deve essere impeccabile sotto il profilo della sicurezza e dell'affidabilità. • Scope Molto Ampio: Le funzionalità previste sono numerose e ambiziose (rilevamento AI, allarmi silenziosi, diario criptato, moduli educativi, community). Il rischio di sovra-estendere il progetto è alto.

3.5 C5 - Nexum

3.5.1 Breve descrizione

NEXUM nasce come piattaforma HR evoluta, in grado di connettere aziende, collaboratori e gli studi dei CdL. Il progetto prevede lo sviluppo di nuovi moduli per la piattaforma esistente, con l'obiettivo di migliorare l'efficienza dei processi HR e offrire funzionalità innovative ai suoi utenti. Il team si occuperà di sviluppare due moduli principali:

- **Ai Assistant Generativo per HR:**

Che dovrà permettere agli utenti di creare in autonomia contenuti accattivanti con titolo, descrizione e immagine di copertina attraverso l'uso di AI generativa, adeguando tono e stile della comunicazione a quello aziendale (formale, informale ...ecc).

- **Ai Co-Pilot per i CdL:**

Deve essere in grado di riconoscere la tipologia di documenti caricati (cedolini, comunicazioni, documenti da firmare, ecc.) e i destinatari, direttamente dal documento e consegnarli ai destinatari anche in modo massivo.

3.5.2 Requisiti funzionali e non funzionali

Tra i requisiti funzioni del progetto vi sono:

- Integrazione con la piattaforma esistente NEXUM
- Utilizzo di tecnologie cloud (AWS)
- Implementazione delle funzionalità di AI generativa e AI Co-Pilot
- Le funzionalità devono essere disponibili sulla dashboard o sulla PWA

Tra i requisiti non funzionali vi sono:

- fluidità d'uso e interfaccia user-friendly
- le operazioni time-consuming dovranno essere delegate a sistemi batch

Altri requisiti di prestazione saranno concordati con il team in base alle tecnologie utilizzate e soluzioni proposte

3.5.3 Tecnologie proposte

Le tecnologie proposte per lo sviluppo del progetto includono:

- **Frontend:** Angular (dashboard amministrativa), Next.js (PWA utenti finali, hosting su AWS Amplify o S3+CloudFront)
- **Backend e API:** Ruby on Rails (stateless su ECS Fargate dietro ALB)
- **Background Jobs:** Sidekiq (su Fargate + SQS)
- **Database e cache:** PostgreSQL (RDS Multi-AZ con snapshot automatici), ElastiCache for Redis (cache e gestione sessioni)
- **Storage:** Amazon S3 (bucket separati per "uploads" e "processed", Lifecycle e legal hold opzionale)
- **Sicurezza:** KMS (gestione chiavi per S3, RDS, Secrets), Secrets Manager (credenziali, API keys, JWT secrets), Amazon Cognito (gestione identità e accessi)
- **Comunicazioni:** SES (email), SNS (notifiche push/eventi)
- **Networking:** VPC (reti private e pubbliche per ECS/RDS/Redis, con ALB e NAT Gateway), Security Groups (accesso minimo privilegio)
- **Sicurezza web:** WAF + AWS Shield (protezione ALB/CloudFront), IAM policy granulari (permessi per ECS task roles, S3, SQS, Secrets, CloudWatch)
- **Osservabilità:** CloudWatch Logs/Metrics/Alarms/X-Ray (monitoring e tracing)

- **Compliance e threat detection:** AWS Config + GuardDuty

3.5.4 Chiarimenti e colloqui con l'azienda

Qui di seguito sono riportate alcune delle domande poste ad Eggon con le relative risposte fornite dall'azienda. In particolare abbiamo posto domande che riguardavano il supporto di Eggon durante lo sviluppo del progetto, l'importanza della conoscenza pregressa delle tecnologie richieste e il coinvolgimento del team nel processo SCRUM dell'azienda, in quanto volevamo comprendere come l'azienda avrebbe affrontato la gestione di un progetto di questo tipo, trattandosi del primo anno in cui Eggon vi partecipava.

Domande	Risposte
Le tecnologie richieste per lo sviluppo del progetto sono diverse e molte sono completamente nuove per noi: quanto è rilevante per voi la conoscenza pregressa e quali sono le vostre aspettative rispetto al nostro apprendimento progressivo durante il progetto? Verrà fornito supporto o affiancamento nell'utilizzo di queste tecnologie?	<ul style="list-style-type: none"> • Conoscenza pregressa: non vincolante. Valutiamo impegno, qualità del codice e velocità di apprendimento. • Aspettative: avanzamento sprint-by-sprint, PR piccole e frequenti, test minimi, documentazione essenziale. • Supporto Eggon: kickoff e seed repository, canale e-mail/Telegram, code review e pairing su temi critici, sandbox (API mock, S3, chiavi temporanee).
Nella vostra esperienza avete già avuto modo di affidare una certa responsabilità operativa o decisionale a team che non avevano ancora esperienza nel mondo del lavoro? Se sì, quali risultati o insegnamenti ne avete tratto in termini di autonomia, qualità del lavoro e collaborazione con il vostro team interno?	<ul style="list-style-type: none"> • Ci lavoriamo spesso: funziona quando suddividiamo il lavoro in milestone piccole con demo frequenti, manteniamo standard chiari (lint/test/review) e i blocchi emergono subito. • Obiettivo: autonomia crescente — più guida all'inizio, più ownership col passare degli sprint.
Avete parlato di includere il team di lavoro nel vostro processo SCRUM e nelle riunioni o stand-up periodiche: quale cadenza hanno questi incontri e come si svolgono concretamente? Considerando che abbiamo anche impegni universitari, ci potete chiarire se è previsto che partecipiamo a tutte le daily stand-up o solo ad alcune delle cerimonie principali (ad esempio sprint review o retrospettive)?	<ul style="list-style-type: none"> • Sprint: 2 settimane. • Cerimonie: <ul style="list-style-type: none"> – Grooming/Planning ($\approx 1h$; nei primi sprint può servire più tempo) — obbligatoria. – Check-in asincroni su Telegram (daily in 3 righe: fatto / da fare / blocchi). – Review + Retro ($\approx 1h$) — obbligatorie con demo. • Calendario condiviso. <ul style="list-style-type: none"> – Lo costruiamo insieme attorno ai vostri impegni di studio (lezioni, esami, sessioni). – Una volta concordate milestone e scadenze, ci si impegna a rispettarle: fa parte del patto professionale azienda-fornitore e ci permette di coordinare bene tutto il team.

Quali modelli LLM specifici prevedete di utilizzare? Oppure possiamo testare con diversi provider?	<ul style="list-style-type: none"> • Preferenza: AWS Bedrock (integrazione e governance). Tramite Bedrock possiamo usare più modelli (Claude, Llama, Mistral). • Apertura ad alternative: via libera a provider/idee creative, purché valutate su qualità output, aderenza al prompt, performance, costi e manutenibilità. • Requisito: adapter per evitare lock-in
--	---

3.5.5 Interesse del team

Il team ha mostrato un interesse significativo verso la proposta di Eggon. Inanzitutto, i rappresentanti dell'azienda hanno subito mostrato uno stile giovanile e informale, pur mantenendo professionalità e struttura. Ciò ha avuto un impatto positivo come prima impressione.

Inoltre, il progetto proposto si allinea bene con le aspirazioni del team di acquisire esperienza pratica nello sviluppo di applicazioni reali, specialmente in un contesto aziendale. L'opportunità di lavorare su una piattaforma HR esistente come NEXUM, che ha una base di utenti reale, ha particolare attrattiva.

Infine, come già specificato all'inizio di questo documento, il team pone molta importanza al supporto che l'azienda può offrire durante lo sviluppo del progetto. Le risposte fornite da Eggon alle nostre domande hanno confermato che l'azienda è disposta a fornire un supporto strutturato e continuo, il che aumenta ulteriormente l'interesse del team verso questo capitolato.

3.5.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none"> • Prodotto Reale e Integrazione con Piattaforma Esistente: NEXUM è una piattaforma HR già operativa. Sviluppare moduli che si integreranno in un prodotto commerciale fornisce un'esperienza di lavoro su codice legacy e integrazione con sistemi esistenti. • Processo SCRUM realistico (rispettare le scadenze dell'azienda) • Ottimo supporto dell'azienda: l'azienda ha un piano ben strutturato quali le cerimonie a cui il team deve necessariamente partecipare (con relativa durata) e il tipo di dialogo che vuole con il team (giornalmente). Inoltre siamo stati informati che il supporto al progetto è continuo (anche formazione e mentoring). • Tecnologie definite esaustivamente e requisiti del progetto chiari, con due requisiti opzionali definiti. • L'azienda ha fornito un documento per il capitolato completo di casi d'uso e obiettivi misurabili in milestone, rendendo la pianificazione più semplice • Integrazione di AI. Lavorare su queste tecnologie fornisce un'esperienza preziosa vista la loro crescente importanza nel settore. • Valori dell'azienda?? 	<ul style="list-style-type: none"> • NEXUM è una piattaforma già esistente e operativa, lo sviluppo di nuovi moduli (che verranno realmente utilizzati) comporta un livello di rischio elevato. L'integrazione in un prodotto reale, destinato a un mercato effettivo e a clienti concreti, implica vincoli progettuali e tecnologici particolarmente rigidi, dettati dalle esigenze e dagli standard aziendali. Questo può limitare la flessibilità del team e aumentare la complessità delle attività di sviluppo e validazione • Alta complessità riguardante le tecnologie da utilizzare con AWS, OCR (Optical Character Recognition), e integrazione dell'AI.

4 C6 - Second Brain

4.1 Breve descrizione

Sviluppare una **web app** (in **HTML** e altre tecnologie web) che rappresenti un **editor basato su Markdown** con l'integrazione di strumenti di **Intelligenza Artificiale (AI)** per assistere l'utente nella scrittura e revisione dei testi. In particolare l'applicazione dovrà essere in grado di interagire con l'AI per eseguire operazioni come riassumere, riscrivere, tradurre e criticare il testo scritto dall'utente. Inoltre, l'applicazione dovrà supportare un prompt generativo associato all'editor per permettere la generazione automatica di testi.

4.2 Requisiti funzionali e non funzionali

Tra i requisiti funzionali principali del progetto, l'applicazione deve includere:

1. **Editing del testo:** area di scrittura che accetta testo e marcatori in formato **Markdown**.
2. **Rendering grafico:** visualizzazione del testo formattato, applicando lo stile corrispondente ai marcatori Markdown.
3. **Accesso a un LLM:** integrazione con un *Large Language Model* in grado di operare sull'intero testo o su porzioni selezionate.
4. **Comandi di base basati su AI:**
 - Riassunto del testo.
 - Riscrittura automatica del testo.
 - Traduzione del testo in una lingua differente.
5. **Critica del testo:** implementazione di comandi di analisi secondo il modello dei “*sei cappelli per pensare*” di *Edward De Bono*.
6. **Prompt generativo:** gestione di un prompt associato che consente la generazione automatica di un intero testo all'interno della finestra di editing.
7. **Persistenza dei dati:** possibilità di salvare e leggere le note come file di testo.

Zucchetti è più interessata all'idea di esplorare le caratteristiche dell'AI (e in particolare del concetto dei ‘Sei cappelli per pensare’) piuttosto che ad un concreto uso dell'applicazione. Per questo motivo, presentazione e salvataggio nel database, o un sistema di autenticazione utenti, sono opzionali e verrà come minimo richiesta la possibilità di salvare e leggere note dal file system.

Eventualmente l'Azienda potrebbe fornire un database volto all'integrazione dell'applicazione con un sistema di autenticazione utenti e salvataggio delle note. Estendendo il progetto in questa direzione, sarebbe possibile anche costruire un sistema di *linking* tra le note.

4.3 Tecnologie proposte

- **Frontend:** HTML5 per l'interfaccia web dell'editor; linguaggio di markup *Markdown* per la scrittura; librerie di rendering come *marked.js*, *Showdown* o *markdown-it*.
- **Integrazione LLM:** API compatibili OpenAI per chiamate a modelli linguistici; supporto a modelli come *Gemini* (Google), *Mistral*, *Gemma*; uso opzionale di *llama.cpp* per esecuzione locale.
- **Backend e API (opzionale):** server-side in *Python* o *Java*; gestione delle richieste HTTP e delle interazioni con il database; middleware per superare la same-origin policy.
- **Database (opzionale):** archivio note e metadati su DB; supporto per collegamenti tra note (link nel markup).

- **Modelli ispiratori:** *Joplin*, *Obsidian*, *Logseq* — esempi di editor Markdown; *ChatGPT* come riferimento per l'interfaccia conversazionale.

4.4 Chiarimenti e colloqui con l'azienda

4.5 Interesse del team

Troviamo che il capitolato proposto da Zucchetti sia realistico per un team di studenti universitari che si interfacciano per la prima volta ad un progetto di questo calibro, sia dal punto di vista delle tempistiche e delle competenze richieste. Inoltre, l'argomento dell'integrazione di modelli di linguaggio di grandi dimensioni (LLM) in applicazioni pratiche è estremamente attuale e rilevante nel contesto tecnologico odierno.

4.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none"> • L'integrazione degli LLM in un'applicazione fornisce un'esperienza preziosa vista la loro crescente importanza nel settore. Lavorare su questo progetto fornisce un'esperienza diretta con questo tipo di tecnologia. • Ampio spazio per la progettazione e la ricerca: Il capitolato non specifica esattamente come implementare le funzionalità, ma si concentra sul cosa. Questo lascia al team una grande libertà nelle scelte architetturali, nella progettazione dell'interfaccia utente (UI/UX) e, soprattutto, nell'ingegnerizzazione dei prompt per l'LLM, che è il cuore del progetto. • Zucchetti si offre esplicitamente di supportare il team nelle parti più complesse (es. configurazione API LLM) e metterà a disposizione gli LLM stessi. • Tempi di realizzazione adeguati: Zucchetti ha esplicitamente confermato che aggiungere altre funzionalità (es. tema della sicurezza) potrebbe impegnare al punto da sfiorare i tempi previsti per il progetto. Questo indica una consapevolezza realistica delle tempistiche e delle capacità del team. 	<ul style="list-style-type: none"> • Le tecnologie non risultano chiaramente definite e i requisiti di progetto sono ampi. Inoltre, l'assenza di specifiche vincolanti e di requisiti opzionali chiaramente delineati comporta un'eccessiva libertà decisionale, che può generare incertezza nella pianificazione e aumentare il rischio di deviazioni dagli obiettivi progettuali. • L'azienda propone lo sviluppo di un'applicazione web basata su HTML per la parte front-end. Tuttavia, il team di progetto, è interessato anche all'utilizzo di nuove tecnologie più moderne e avanzate per lo sviluppo web, come i framework <i>React</i> e <i>Angular</i>.

5 C7 - Sistema di acquisizione dati da sensori

5.1 Breve descrizione

Il progetto mira a sviluppare un sistema distribuito di acquisizione e smistamento dati dai sensori BLE. La piattaforma deve essere in grado di ricevere, aggregare, normalizzare e smistare tali informazioni in modo affidabile, sicuro e scalabile.

La piattaforma deve essere articolata in tre livelli principali:

- **Sensori BLE:** dispositivi periferici che raccolgono dati dal campo (es. temperatura, umidità, movimento, segnali biometrici);

- **Gateway BLE–WiFi:** nodi intermedi che si connettono ai sensori, raccolgono i dati tramite profili standard o personalizzati, li formattano secondo un modello interno e li inviano al cloud;
- **Cloud:** piattaforma centrale che gestisce la connessione sicura dei gateway, riceve e bufferizza i dati, li rende disponibili tramite API e interfacce di visualizzazione, garantendo segregazione tra diversi tenant.

I primi due livelli (sensori BLE e gateway) non sono oggetto di questo capitolato e devono essere considerati come strumenti esterni. Nel contesto del progetto, il team non deve realizzare un gateway fisico, ma sviluppare un **simulatore di gateway** in grado di riprodurre in modo verosimile il comportamento di un nodo BLE–WiFi. Tale simulatore sarà utilizzato per validare l’infrastruttura cloud e testare i flussi di comunicazione previsti.

5.2 Requisiti funzionali e non funzionali

Tra i requisiti funzionali del progetto vi sono:

1. **Acquisizione e generazione dati da sensori:** il sistema deve gestire dati provenienti da sensori Bluetooth Low Energy (BLE). Poiché i sensori reali non rientrano nell’ambito del progetto, i dati saranno generati dal simulatore di gateway, che produrrà valori realistici per 4–5 tipologie di sensori da concordare.
2. **Simulatore di gateway:** in sostituzione di un gateway fisico, deve essere sviluppato un simulatore in grado di riprodurre il comportamento principale.
 - (a) generazione diretta dei dati simulati pronti alla trasmissione;
 - (b) invio al cloud tramite protocolli sicuri (SSL/TLS);
 - (c) gestione di più sensori simulati in parallelo;
 - (d) persistenza delle informazioni di commissioning (sensori/gateway) per garantire consistenza;
 - (e) capacità di rispondere ai messaggi dal cloud, anche in forma semplificata (con supporto opzionale a flag di debug che contengano la risposta attesa).
3. **Gestione multi-tenant:** il livello cloud deve garantire isolamento logico tra diversi tenant, permettendo a ciascun cliente/utente di accedere solo ai propri dati.
4. **Esposizione API centralizzate:** i dati raccolti dal simulatore e gestiti dal cloud devono essere accessibili tramite API sicure, documentate e versionate (test).
 - (a) ogni client deve essere identificato univocamente e certamente, per garantire il controllo dell’accesso ai dati e la loro segregazione;
 - (b) le API devono fornire accesso ai dati nelle seguenti modalità:
 - i. *on demand:* accesso su richiesta a uno o più dati di un gateway, con possibilità di filtrare i dati per sensore e data/ora di ricezione;
 - ii. *stream:* accesso ai dati in tempo reale attraverso uno stream continuo.
5. **Interfaccia utente:** deve essere fornita una UI web che consenta la consultazione e l’esplorazione dei dati acquisiti, con procedure semplificate per registrare e configurare nuovi sensori o gateway simulati.

Tra i requisiti non funzionali vi sono:

- **Scalabilità;**
- **Test automatizzati e code coverage;**
- **Monitoraggio:** devono essere predisposti strumenti di monitoraggio in tempo reale per le performance del sistema. In aggiunta, devono essere presenti almeno funzionalità di alert di

base che consentano di capire rapidamente se un gateway sta funzionando correttamente o risulta non raggiungibile;

- **Versionamento e DevOps:** il codice deve essere gestito tramite Git con pipeline CI/CD per il deploy su ambienti di test e produzione.

5.3 Tecnologie proposte

Le tecnologie proposte per lo sviluppo del progetto includono:

- **Sviluppo:** interfaccia utente in **Angular** per la creazione di una **SPA** (Single Page Application); microservizi sviluppati con **Node.js** e **Nest.js** in **TypeScript** per la parte di API.
- **Orchestrazione:** il sistema sarà ospitato su *Google Cloud Platform* e orchestrato tramite **Kubernetes**.
- **Archiviazione dati:** *MongoDB* per la memorizzazione di dati non strutturati (quelli grezzi raccolti dai sensori); *PostgreSQL* per la persistenza dei dati strutturati (quelli raccolti e manipolati per renderli *valuable*).
- **Opzionale:** *Redis* come sistema di caching per ridurre la latenza e migliorare le prestazioni.

Tuttavia le tecnologie sopra elencate non sono vincolanti: il team può proporre alternative ma devono in ultimo essere approvate da M31.

5.4 Chiarimenti e colloqui con l'azienda

Qui di seguito sono riportate alcune delle domande poste a M31 con le relative risposte fornite dall'azienda. Le domande vertono principalmente su aspetti tecnici del progetto, poichè il team voleva chiarire alcuni dubbi riguardanti l'architettura del sistema e le tecnologie da utilizzare.

Il team ha deciso di fissare una riunione per discutere di questi aspetti, in modo da poter porre domande specifiche e ricevere risposte dettagliate direttamente dai referenti dell'azienda. Questo approccio ha permesso di ottenere chiarimenti più approfonditi e di comprendere meglio le aspettative dell'azienda riguardo al progetto.

Le risposte dell'azienda non sono citazioni dirette, ma sono state riassunte e riformulate dal team in modo da essere più concise e chiare.

Domande	Risposte
---------	----------

<p>Modello di identità e provisioning dei gateway simulati</p> <ul style="list-style-type: none"> • Quale modello di identità volete per i gateway simulati? • Come è previsto che avvenga il provisioning dei gateway nel cloud (flusso tipico)? • Immaginiamo debba simulare il provisioning reale di un gateway fisico; tuttavia, da quanto abbiamo visto, non esiste un'unica modalità. Quale approccio ritenete più in linea con i vostri gateway reali? • Alcuni possibili tipi di provisioning che abbiamo considerato: <ul style="list-style-type: none"> – Provisioning manuale (statico) – Provisioning basato su certificati X.509 (autogenerati) – Provisioning Just-In-Time (JIT) – Provisioning Just-Enough-Time (JET) 	
<p>Simulazione sensori e profili BLE</p> <ul style="list-style-type: none"> • Potete indicarci quali sensori dobbiamo simulare e, di conseguenza, quali profili BLE standard è opportuno utilizzare? • Avete già definito eventuali profili custom? 	
<p>Persistenza delle informazioni di commissioning</p> <p>Avete qualche consiglio sulla modalità di persistenza delle informazioni di commissioning dei sensori (nel gateway) e del gateway (nel cloud)?</p>	
<p>Gestione dei tenant</p> <p>I tenant sono assegnati a un amministratore (con pieno accesso ai gateway), a un cliente generico (con permessi limitati), oppure ad entrambi?</p>	
<p>Sicurezza</p> <ul style="list-style-type: none"> • Sono citate tre tecnologie di sicurezza: JWT, OAuth2 e mTLS. Nel contesto di sensori e gateway simulati, quali di queste devono essere utilizzate in modo “diverso” rispetto al consueto? • Per il lato dispositivi simulati (sensori + gateway), quali tecnologie delle proposte consigliate? • Per il lato client (cloud + API), quali tecnologie delle proposte consigliate? 	
<p>Supporto e modalità di collaborazione</p> <ul style="list-style-type: none"> • Che tipo di supporto intendete fornirci durante il progetto? • La gestione sarà lasciata interamente a noi oppure è prevista una forma di supporto strutturata (ad esempio con riunioni periodiche, review o retrospective in stile Scrum)? 	

Le tecnologie richieste per lo sviluppo del progetto sono diverse e molte sono completamente nuove per noi: quanto è rilevante per voi la conoscenza pregressa e quali sono le vostre aspettative rispetto al nostro apprendimento progressivo durante il progetto?	<ul style="list-style-type: none"> Le tecnologie indicate sono quelle da noi suggerite e utilizzate anche internamente. Questo ci permetterebbe di offrirvi un supporto più puntuale ed efficace. Le stesse tecnologie sono state consigliate anche ai gruppi che hanno lavorato al progetto proposto l'anno scorso, di complessità analoga. Siamo consapevoli del numero limitato di ore a vostra disposizione e della complessità — sicuramente stimolante — del progetto, ma grazie all'esperienza pregressa sappiamo cosa aspettarci in termini di output e possiamo venirvi incontro. Ad esempio, non è necessario approfondire GCP o Kubernetes: potete limitarvi a utilizzare Docker in locale o, se lo ritenete opportuno, Minikube. Un'altra possibilità è quella di utilizzare un'unica tecnologia per lo sviluppo dei vari componenti, sfruttando ad esempio TypeScript — con Node.js per la parte cloud e per il simulatore gateway, e Angular per la dashboard. Questi dettagli, tuttavia, li definiremo progressivamente, dopo che avrete avuto modo di condurre un adeguato studio preliminare del progetto, come è naturale che sia.
Verrà fornito supporto o affiancamento nell'utilizzo di queste tecnologie? Potreste indicarci una lista di conoscenze o competenze di base da acquisire per affrontare al meglio il capitolato?	Non è previsto un supporto attivo, come corsi o lezioni dirette, e riteniamo che non sia necessario, poiché parte integrante del progetto consiste proprio nello studio e nella comprensione di queste tecnologie da parte vostra. Per questo motivo non riteniamo indispensabili competenze pregresse. Possiamo tuttavia suggerirvi di esplorare e analizzare soluzioni esistenti già presenti sul mercato, ad esempio nel mondo open source. Naturalmente resteremo a disposizione per domande e supporto mirato e, se necessario, potremo organizzare incontri dedicati per aiutarvi a chiarire dubbi o risolvere eventuali blocchi.

5.5 Interesse del team

L'interesse del team verso questo capitolato è stato inizialmente moderato, ma è cresciuto significativamente dopo aver approfondito i dettagli del progetto e aver interagito con l'azienda proponente, M31.

Inizialmente il team era attratto dall'idea di lavorare su un sistema distribuito e scalabile, ma aveva delle riserve riguardo alla complessità tecnica e alla mancanza di esperienza con molte delle tecnologie proposte. Tuttavia, dopo aver avuto l'opportunità di porre domande specifiche a M31 e ricevere chiarimenti, il team ha acquisito una maggiore comprensione delle sfide e delle opportunità offerte dal progetto.

5.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
----------------	--------------------

<ul style="list-style-type: none"> • Il progetto affronta una sfida attuale e molto concreta nel campo IoT: acquisire, gestire e rendere fruibili dati da sensori in modo sicuro, scalabile e multi-tenant. Questo fornisce un'esperienza su problematiche industriali reali, non solo accademiche • supporto?? • Forte opportunità di apprendimento: il progetto introduce tecnologie e pratiche non ancora esplorate dal team, accelerando la crescita tecnica. Nello specifico, il sistema di acquisizione dati da sensori si basa su un'architettura a microservizi e introduce tecnologie come Node.js e Nest.js con TypeScript (backend), Go (componenti ad alte prestazioni), Angular (frontend SPA), MongoDB (database NoSQL), PostgreSQL (database SQL), Redis (caching), Google Cloud Platform (infrastruttura cloud), Kubernetes (orchestrazione container), NATS o Kafka (comunicazione asincrona), e Prometheus con Grafana (monitoring) 	<ul style="list-style-type: none"> • Progettare e implementare un'architettura distribuita, sicura e multi-tenant è una sfida tecnica notevole. La gestione della concorrenza, della consistenza dei dati, della comunicazione asincrona e della tolleranza ai guasti richiede competenze solide e un'attenta progettazione. • I requisiti sono molti e articolati (dalle API on-demand e streaming, alla UI, al monitoring, alla sicurezza). C'è il concreto rischio di dover sacrificare la profondità di alcune funzionalità per coprirne la quantità entro i tempi del progetto. • Testare un'architettura distribuita è intrinsecamente difficile. Configurare ambienti di test end-to-end, simulare guasti e testare la scalabilità richiederà sforzi notevoli e una buona pianificazione. • Il gateway e i dispositivi devono essere simulati. I dati grezzi che vengono generati dai sensori devono essere realistici. Inoltre deve essere posta particolare attenzione alla coerenza (p.es i dati delle temperature devono essere non eccessive e adatte al contesto). Eventualmente i dati devono essere normalizzati. Rispetto al maggiore competitor (Vimar), M31 non prevede utilizzo di veri sensori per aiutare il team nello sviluppo.
--	--

5.7 C8 - Smart Order

5.7.1 Breve descrizione

Analisi multimodale per la creazione automatica di ordini, proposto da Ergon Informatica Srl, prevede la realizzazione di una piattaforma intelligente in grado di ricevere, interpretare e strutturare automaticamente ordini provenienti da input testuali, vocali e visivi, trasformando dati non strutturati in ordini cliente completi pronti per l'inserimento nei sistemi gestionali aziendali. L'obiettivo del progetto è quello di migliorare l'efficienza dei processi aziendali, riducendo al minimo l'intervento umano nelle fasi più ripetitive e soggette a errore.

5.7.2 Requisiti funzionali e non funzionali

Tra i requisiti funzionali del progetto vi sono:

- **Integrazione multimodale completa:** il sistema deve poter ricevere e gestire input di testo, immagini e audio, integrandoli in un unico flusso coerente.
- **Pipeline di elaborazione dati:** implementazione di tutti i layer funzionali descritti nel capitolato.
- **Output strutturato e integrazione gestionale:** gli ordini generati devono essere completi, coerenti e in formato strutturato (ad esempio JSON, XML o tabelle relazionali), pronti per l'inserimento in un sistema ERP.
- **Accuratezza e validazione automatica:** devono essere previsti controlli di coerenza e integrità dei dati, per garantire che gli ordini generati siano affidabili e corretti.
- **Interfaccia e API di comunicazione:** il sistema deve prevedere API REST per l'interazione tra il modello AI e i componenti esterni, in particolare con il database o l'applicativo gestionale.

- **Caso di studio e testing:** il sistema deve essere testato su un caso di studio reale fornito dall'azienda, includendo verifiche di performance e accuratezza.

Tra i requisiti non funzionali vi sono:

- **Modularità e scalabilità del sistema:** ogni componente della pipeline deve essere progettato come modulo indipendente, aggiornabile senza impattare l'intero sistema.
- **Logging e feedback continuo:** implementazione di un sistema di monitoraggio e logging che consenta di raccogliere errori, aggiornare le regole aziendali e migliorare le prestazioni tramite retraining periodico.
- **Documentazione tecnica completa:** il sistema deve essere accompagnato da una documentazione chiara, aggiornata e completa di istruzioni per l'uso e la manutenzione.
- **Supporto e replicabilità:** il sistema deve essere facilmente replicabile in ambienti diversi e supportato da istruzioni di installazione e configurazione.

5.7.3 Tecnologie proposte

- **Database relazionale:** SQL Server Express, MySQL o MariaDB.
- **Modelli di linguaggio e NLP:** BERT, RoBERTa, GPT.
- **Visione computazionale e OCR:** Tesseract OCR, EasyOCR, Convolutional Neural Networks (CNN), Vision Transformer (ViT).
- **Riconoscimento vocale e trascrizione:** Whisper (OpenAI), Google Speech-to-Text.
- **API REST.**
- **Comunicazione da/per il database:** connettori da una fonte dati ODBC o indipendenza del modello LLM dal database.
- **Interfaccia utente:** .NET Blazor, React.js, Angular.

5.7.4 Chiarimenti e colloqui con l'azienda

5.7.5 Interesse del team

Il gruppo ha valutato positivamente la modernità e la rilevanza del tema, che combina più ambiti dell'intelligenza artificiale offrendo l'opportunità di approfondire tecniche avanzate di Machine Learning e integrazione multimodale. L'architettura proposta è risultata flessibile e ben strutturata, permettendo l'uso di tecnologie moderne come LLM, OCR e sistemi speech-to-text, oltre a framework web come React o Blazor. È stato inoltre ritenuto molto positivo il supporto previsto da parte dell'azienda proponente e la possibilità di lavorare su un caso di studio reale fornito da Ergon.

5.7.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
----------------	--------------------

<ul style="list-style-type: none">• L'integrazione di NLP, Computer Vision e Speech-to-Text in un'unica pipeline è un campo di ricerca e sviluppo estremamente attuale e complesso. Offre un'esperienza di apprendimento su tecnologie AI d'avanguardia.• Viene fornito un ricco elenco di tecnologie suggerite per ogni componente (LLM, OCR, Speech-to-Text, UI), ma con la libertà di scegliere alternative. Questo permette al team di adattare lo stack tecnologico alle proprie competenze.	<ul style="list-style-type: none">• Integrare e far collaborare modelli di AI diversi (LLM, Vision, Audio) in una pipeline coerente è una sfida tecnica notevolissima.• Il team dovrebbe padroneggiare o imparare rapidamente una vasta gamma di tecnologie AI e di framework, ciascuna con le sue complessità.• Il pre-processing di dati multimodali (pulizia del testo, elaborazione di immagini, trascrizione audio) è un compito laborioso e critico. La qualità dell'output finale dipende fortemente da questa fase.• I modelli di AI, specialmente gli LLM, possono essere imprevedibili. Validare l'accuratezza e l'affidabilità del sistema in scenari reali, dove un errore può significare un ordine sbagliato, è una sfida significativa.
--	---

5.8 C9 - View4Life

5.8.1 Breve descrizione

Il capitolato proposto da Vimar S.p.A. ha come obiettivo la realizzazione di una piattaforma per la gestione intelligente di impianti domotici. all'interno di residenze protette per anziani, con l'intento di migliorare la sicurezza, il comfort e l'efficienza energetica degli ambienti. Il progetto prevede lo sviluppo di un'infrastruttura cloud e di un'applicazione web responsive dedicata al personale sanitario per il monitoraggio e il controllo dei dispositivi

5.8.2 Requisiti funzionali e non funzionali

5.8.3 Tecnologie proposte

- **Infrastruttura Cloud:** deve utilizzare **Docker** con **docker-compose**, in modo da rispettare il principio di *Infrastructure as Code*.
- **Applicativo web:** deve prevedere l'utilizzo della tecnologia **KNX IoT 3rd Party API**.
- **Autenticazione:** l'interfaccia KNX IoT richiede l'uso del protocollo **OAuth2** per la gestione sicura dell'autenticazione.
- **Notifiche di impianto:** l'applicativo deve utilizzare il meccanismo di ricezione delle notifiche *push* previsto dallo standard KNX IoT. Meccanismi di aggiornamento periodico tramite *polling* non sono ammessi.
- **Versionamento:** il repository di lavoro deve essere gestito con **Git** e reso **pubblicamente accessibile**. I sorgenti dovranno essere rilasciati con **licenza open source**.
- **Compatibilità delle licenze:** il gruppo di lavoro deve prestare attenzione alla compatibilità delle licenze open-source adottate rispetto alle librerie o ai framework utilizzati.

5.8.4 Chiarimenti e colloqui con l'azienda

5.8.5 Interesse del team

Il gruppo ha valutato molto positivamente questo capitolato per la concretezza e l'impatto sociale: lavorare su un sistema capace di migliorare la qualità della vita e la sicurezza degli anziani rappresenta una forte motivazione, dando al progetto un valore tangibile e significativo. Sono stati inoltre apprezzati l'utilizzo di tecnologie moderne e versatili, come Docker, AWS e framework web come Angular o React, che offrono ampie opportunità di apprendimento e applicazione pratica. Un ulteriore punto di forza è rappresentato dal supporto fornito da Vimar, che garantisce incontri di avanzamento regolari e fornisce un kit hardware fisico per testare l'integrazione con dispositivi reali, riducendo il divario tra teoria e pratica e permettendo di affrontare concretamente le sfide dell'IoT.

La principale complessità del progetto risiede nella sua ampiezza e completezza, che richiede una pianificazione accurata e una buona capacità di prioritizzazione. Tuttavia, il gruppo ha interpretato questa caratteristica non come una difficoltà, ma come un'opportunità di crescita: gestire un progetto così articolato permetterà infatti di sviluppare competenze di organizzazione, coordinamento e gestione tecnica avanzata

5.8.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
<ul style="list-style-type: none">• Dominio Applicativo Concreto e Socialmente Utile: Il contesto delle "residenze protette" fornisce uno scopo nobile e tangibile. Sviluppare un sistema che può migliorare la sicurezza e il benessere degli anziani è un forte motivatore e rende il progetto molto più significativo rispetto a un dominio astratto.• Tecnologie Moderne e Ricerche• Supporto Aziendale Eccezionale e Materiale Fornito: Vimar fornisce un supporto strutturato con incontri bisettimanali/settimanali (SAL) e, aspetto cruciale, fornisce un kit hardware fisico per testare con dispositivi reali. Questo riduce enormemente il gap tra teoria e pratica e permette di affrontare problematiche reali di integrazione IoT.	<ul style="list-style-type: none">• Elevata Complessità e Ampio Scope: La completezza del progetto è anche la sua principale sfida. Il team deve essere bravo a gestire la complessità e a priorizzare le funzionalità obbligatorie, per non perdersi negli optional. Una buona pianificazione iniziale è fondamentale.• Punto di debolezza 2• Punto di debolezza 3