



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 2.0.0

Redattore_g	Linor Sadè
Verificatori_g	Alberto Pignat, Alberto Autiero
Destinatari	Prof. Tullio Vardanega, Prof. Riccardo Cardin, BugBusters
Data ultima modifica	30/10/2025

Abstract

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

Registro delle modifiche

Versione	Data	Descrizione	Redatto	Verificato	Approvato
2.0.0	30/01/2026	Approvazione _c del documento	-	-	Luca Slongo
1.1.0	30/01/2026	Verifica _c del documento	-	Alberto Pignat	-
1.0.1	29/01/2026	Aggiunti ai termini presenti nel Glossario _c la G	Luca Slongo	-	-
1.0.0	30/10/2025	Approvazione _c del documento.	-	-	Alberto Autiero
0.2.0	30/10/2025	Verificati i capitoli _c C5, C6, C7, C8, C9.	-	Alberto Pignat	-
0.1.0	30/10/2025	Verificati i capitoli _c C1, C2, C3, C4.	-	Alberto Autiero	-
0.0.12	29/10/2025	Aggiunta tabella conclusioni.	Linor Sadè	-	-
0.0.11	29/10/2025	Correzioni minime a C9.	Alberto Pignat	-	-
0.0.10	26/10/2025	Correzioni minime a C4, C5, C6, C7.	Linor Sadè	-	-
0.0.9	24/10/2025	Continuo del documento con modifiche di C5 e C7 relative ai meeting svolti successivamente. Minori modifiche di stile.	Linor Sadè	-	-
0.0.8	24/10/2025	Stesura di C4 e altre piccole modifiche.	Linor Sadè	-	-
0.0.7	23/10/2025	Stesura di C6.	Linor Sadè	-	-
0.0.6	22/10/2025	Correzione indice.	Linor Sadè	-	-
0.0.5	22/10/2025	Iniziato e terminato C7.	Linor Sadè	-	-
0.0.4	22/10/2025	Correzioni minori stile.	Linor Sadè	-	-
0.0.3	21/10/2025	Terminato C5, iniziato C8, C9.	Linor Sadè	-	-
0.0.2	21/10/2025	Aggiunta introduzione, punti di forza/debolezza.	Linor Sadè	-	-
0.0.1	20/10/2025	Prima stesura della struttura del documento.	Linor Sadè	-	-

Indice

1	Metodologia di valutazione	4
2	Elenco capitolati_G analizzati	5
2.1	C1 - Automated EN18031 Compliance Verification	6
2.1.1	Breve descrizione	6
2.1.2	Caratteristiche funzionali _G	6
2.1.3	Tecnologie proposte	6
2.1.4	Chiarimenti e colloqui con l'azienda	6
2.1.5	Interesse del team	6
2.1.6	Punti di forza e di debolezza	7
2.2	C2 - Code Guardian	7
2.2.1	Breve descrizione	7
2.2.2	Caratteristiche funzionali _G	7
2.2.3	Tecnologie proposte	8
2.2.4	Chiarimenti e colloqui con l'azienda	8
2.2.5	Interesse del team	8
2.2.6	Punti di forza e di debolezza	8
2.3	C3 - DIPReader	9
2.3.1	Breve descrizione	9
2.3.2	Caratteristiche funzionali _G	9
2.3.3	Tecnologie proposte	9
2.3.4	Chiarimenti e colloqui con l'azienda	9
2.3.5	Interesse del team	9
2.3.6	Punti di forza e di debolezza	10
2.4	C4 - L'app che Protegge e Trasforma	10
2.4.1	Breve descrizione	10
2.4.2	Caratteristiche funzionali _G	10
2.4.3	Tecnologie proposte	11
2.4.4	Chiarimenti e colloqui con l'azienda	12
2.4.5	Interesse del team	12
2.4.6	Punti di forza e di debolezza	12
2.5	C5 - Nexum	14
2.5.1	Breve descrizione	14
2.5.2	Caratteristiche funzionali _G	14
2.5.3	Tecnologie proposte	14
2.5.4	Chiarimenti e colloqui con l'azienda	15
2.5.5	Interesse del team	17
2.5.6	Punti di forza e di debolezza	18
2.6	C6 - Second Brain	18
2.6.1	Breve descrizione	18
2.6.2	Caratteristiche funzionali _G	18
2.6.3	Tecnologie proposte	19
2.6.4	Chiarimenti e colloqui con l'azienda	19
2.6.5	Interesse del team	20
2.6.6	Punti di forza e di debolezza	20
2.7	C7 - Sistema di acquisizione dati da sensori	21
2.7.1	Breve descrizione	21
2.7.2	Caratteristiche funzionali _G	21
2.7.3	Tecnologie proposte	22

2.7.4	Chiarimenti e colloqui con l'azienda	22
2.7.5	Interesse del team	24
2.7.6	Punti di forza e di debolezza	25
2.8	C8 - Smart Order	25
2.8.1	Breve descrizione	25
2.8.2	Caratteristiche funzionali _G	25
2.8.3	Tecnologie proposte	26
2.8.4	Chiarimenti e colloqui con l'azienda	26
2.8.5	Interesse del team	26
2.8.6	Punti di forza e di debolezza	27
2.9	C9 - View4Life	28
2.9.1	Breve descrizione	28
2.9.2	Caratteristiche funzionali _G	28
2.9.3	Tecnologie proposte	28
2.9.4	Chiarimenti e colloqui con l'azienda	29
2.9.5	Interesse del team	30
2.9.6	Punti di forza e di debolezza	30
3	Conclusioni	31

1 Metodologia di valutazione

Per la valutazione dei capitolati_e 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 capitolato_e esaminando l'interesse verso ciascun progetto_e (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 capitolato_e, 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 capitolati_e, il team ha preparato una lista di domande da inviare ai proponenti_e 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_e.
 - L'importanza nella conoscenza pregressa delle tecnologie richieste.
4. **Compilazione tabelle:** Il team ha redatto congiuntamente le tabelle dei punti di forza e di debolezza, sintetizzando le osservazioni emerse nelle fasi precedenti.

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 capitolato_e da scegliere.

È stata data particolare importanza all'interesse verso l'argomento e le tecnologie rispetto alla complessità del progetto_e, senza tuttavia trascurare la valutazione della sua fattibilità.

Si ritiene che lavorare a un progetto_e capace di suscitare entusiasmo e curiosità possa offrire un'esperienza di apprendimento più significativa e gratificante, anche qualora comporti l'affrontare sfide tecniche più impegnative.

Qui di seguito i nomi degli analisti_e e i corrispondenti capitolati_e analizzati:

Analista _e	Capitolato _e	Verificatore _e
Luca Slongo	C1, C2	Alberto Autiero
Marco Piro	C3, C4	Alberto Autiero
Linor Sadè	C5, C6, C7	Alberto Pignat
Leonardo Salviato	C8, C9	Alberto Pignat

2 Elenco capitoli_c analizzati

Di seguito sono elencati i capitoli_c analizzati dal team BugBusters. Per ogni capitolo_c è riportata una breve descrizione e le caratteristiche principali. Infine, vengono presentati l'interesse del team e i punti di forza e di debolezza emersi dall'analisi.

2.1 C1 - Automated EN18031 Compliance Verification

2.1.1 Breve descrizione

Proponente_g : Bluewind S.r.l

Il progetto_g mira ad automatizzare la verifica_g di conformità alla norma EN 18031, legata ai dispositivi radio (Wi-Fi, Bluetooth, IoT, ecc.). L'obiettivo è quello di garantire che le apparecchiature radio non danneggino o abusino delle risorse, che salvaguardino la privacy e siano in grado di garantire sicurezza_g nelle transazioni.

2.1.2 Caratteristiche funzionali_g

Il progetto_g prevede lo sviluppo di un'interfaccia_g grafica che guida nella compilazione delle domande presenti nei decision tree relative ai requisiti_g. Il software deve essere in grado di:

- Importare documenti contenenti le informazioni preliminari necessarie, che descrivono le componenti di rete del caso da analizzare e i file che descrivono i decision tree.
- Eseguire i decision tree associati ai requisiti_g, rispettando la loro struttura gerarchica e le dipendenze_g, in modo interattivo, chiedendo all'utente di inserire le informazioni richieste in input.
- Restituire per ogni requisito_g un output che sia "Non applicabile", "Pass" o "Fail".
- Fornire una dashboard_g interattiva per visualizzare lo stato di ciascun requisito_g del decision tree, con la possibilità di:
 - modificare i documenti associati;
 - aggiornare i documenti e i decision tree tramite un editor;
 - salvare i decision tree modificati in formati come XML o JSON.

2.1.3 Tecnologie proposte

Non sono stati presentati vincoli sulle tecnologie da utilizzare e sulla tipologia di applicazione da sviluppare, l'unica preferenza espressa è sull'uso di Python per la parte di back-end_g.

2.1.4 Chiarimenti e colloqui con l'azienda

Il team non ha contattato l'azienda né per ricevere ulteriori informazioni né per richiedere colloqui.

2.1.5 Interesse del team

Il team non ha mostrato particolare interesse verso questo capitolato_g, principalmente per il motivo che non è stato ritenuto un progetto_g particolarmente accattivante.

2.1.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. Requisiti_e ben definiti: I requisiti_e obbligatori_e e opzionali_e sono elencati in modo chiaro e strutturato. Questo riduce l'ambiguità e fornisce un'ottima checklist per la pianificazione e la verifica_e finale del progetto_e. 2. 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_e, evitando di lavorare in modo troppo astratto. 3. 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.	1. 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_e richiederà uno sforzo iniziale di analisi e studio.

2.2 C2 - Code Guardian

2.2.1 Breve descrizione

Proponente_e : Var Group S.p.A

Il progetto_e mira alla realizzazione di una piattaforma web modulare, basata su un sistema multi-agente, per:

- Analizzare repository_e GitHub_e in termini di qualità_e, sicurezza_e e manutenzione;
- Fornire report automatici e suggerimenti di remediation;
- Evidenziare vulnerabilità;
- Offrire una dashboard_e che mostra lo stato dei progetti_e analizzati.

2.2.2 Caratteristiche funzionali_e

Per il progetto_e è richiesto di rispettare i seguenti vincoli:

- Svolgere un design thinking iniziale con il committente_e per raccogliere i requisiti_e;
- Redigere il documento dei requisiti_e di business e lo user story mapping;
- Realizzare diagrammi UML relativi agli Use Cases_e di progetto_e;
- Definire lo schema di design relativo alla base dati;
- Presentare un MVP funzionante con demo live e documentazione tecnica.

Tramite questi passaggi si intende arrivare a un software in grado di svolgere le attività illustrate nella descrizione.

2.2.3 Tecnologie proposte

L'azienda proponente_G ha specificato chiaramente le tecnologie da utilizzare per lo sviluppo del progetto_G e ha fornito risorse informative per facilitare il "Get started".

Lista delle tecnologie e dei framework richiesti:

- **back-end_G / orchestratore:** Node.js, Python
- **front-end_G :** React.js
- **Database:** MongoDB o PostgreSQL_G
- **CI/CD e integrazioni:** GitHub_G Actions
- **Architettura cloud:** AWS_G

2.2.4 Chiarimenti e colloqui con l'azienda

Non sono state richieste ulteriori informazioni o colloqui all'azienda proponente_G .

2.2.5 Interesse del team

Il team non ha preso in considerazione fin da subito il progetto_G perché è stato ritenuto particolarmente complesso. L'utilizzo di uno strumento come quello che verrebbe realizzato da questo progetto_G sarebbe sicuramente molto utile, ma crearlo è tutt'altra cosa e non ha suscitato l'interesse del gruppo.

2.2.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. 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. 2. Dominio Concreto e Utile: La piattaforma risolve problemi reali di qualità_G del codice, sicurezza_G e manutenzione dei repository_G . 3. Sessione di mentoring sulle tecnologie che verranno utilizzate.	1. 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. 2. Requisiti_G di Testing Stringenti: La richiesta del 70% di test coverage (obbligatorio) è apprezzabile professionalmente ma può essere impegnativa da raggiungere in un progetto_G universitario, specialmente per componenti AI_G . 3. Non sembra esserci flessibilità_G nella scelta delle tecnologie da utilizzare.

2.3 C3 - DIPReader

2.3.1 Breve descrizione

Proponente_e : Sanmarco Informatica SPA

DIPReader propone la realizzazione di un'applicazione per la consultazione locale di pacchetti di conservazione digitale (DIP). L'obiettivo è permettere a professionisti e operatori di accedere offline a documenti conservati, effettuare ricerche e visualizzare anteprime, con funzionalità_e opzionali avanzate quali ricerca semantica e verifica_e delle firme. Il progetto_e risponde a un'esigenza pratica e concreta.

2.3.2 Caratteristiche funzionali_e

Il software dovrà consentire:

- Accesso offline a pacchetti DIP locali;
- Ricerca tra i documenti e visualizzazione di anteprime;
- Funzionalità_e opzionali avanzate quali ricerca semantica e verifica_e delle firme digitali;
- Prestazioni efficienti anche con grandi volumi di documenti;
- Interfaccia_e intuitiva per utenti professionisti e operatori.

I requisiti_e non funzionali_e includono la gestione efficiente della memoria, tempi di risposta rapidi e la conformità a standard di conservazione digitale.

2.3.3 Tecnologie proposte

- Linguaggi di programmazione: Python o Node.js per il back-end_e ;
- Librerie per ricerca e indicizzazione dei documenti (ad esempio Elasticsearch o simili);
- Framework front-end_e : React.js o equivalente per interfaccia_e utente;
- Database: MongoDB o PostgreSQL_e per memorizzazione metadati;
- Gestione dei pacchetti DIP e formati standard (es. XML, PDF/A);
- Strumenti per verifica_e firme digitali.

2.3.4 Chiarimenti e colloqui con l'azienda

Il team non ha contattato l'azienda né per ricevere ulteriori informazioni né per richiedere colloqui.

2.3.5 Interesse del team

Il team valuta positivamente il progetto_e per i seguenti motivi:

- Il dominio è specializzato e di alto valore, con competenze spendibili professionalmente;
- Il progetto_e affronta un problema concreto e ben definito, rendendo immediatamente applicabili i risultati sviluppati.

Tuttavia, il team non ha ritenuto opportuno proseguire con un'analisi più approfondita, che avrebbe previsto anche un confronto diretto con l'azienda per eventuali chiarimenti, poiché il progetto_e non è stato considerato sufficientemente stimolante dal punto di vista tecnico.

2.3.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. Dominio specializzato e di alto valore: La conservazione digitale è un campo di nicchia ma cruciale, specialmente in ambito legale e amministrativo. 2. Problema concreto e ben definito: L'esigenza di accedere a documenti conservati digitalmente in modalità offline è un requisito_e reale per molti professionisti. Il progetto_e risolve un problema tangibile. 3. Interazione con l'azienda: offre esempi di pacchetti estratti dal sistema di conservazione e la relativa documentazione.	1. Complessità del Dominio Normativo: La conservazione digitale è regolata da standard e normative complesse. Comprendere appieno il formato dei DIP e i requisiti_e di compliance richiederà uno sforzo iniziale. 2. 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.

2.4 C4 - L'app che Protegge e Trasforma

2.4.1 Breve descrizione

Proponente_e : Miriade S.r.l.

Il progetto_e "*L'app che Protegge e Trasforma*", proposto da Miriade S.r.l., ha come obiettivo la realizzazione di un'applicazione mobile innovativa per i sistemi operativi iOS e Android, finalizzata alla prevenzione e al supporto delle vittime di violenza di genere.

L'app è concepita come uno strumento intelligente, sicuro e facilmente accessibile, in grado di:

- riconoscere segnali di pericolo e situazioni di rischio_e tramite analisi del linguaggio e dei comportamenti;
- offrire risorse immediate e personalizzate per il supporto psicologico, legale e d'emergenza;
- promuovere la consapevolezza e l'autonomia dell'utente attraverso moduli_e educativi e interattivi;
- garantire la sicurezza_e e la privacy dei dati sensibili mediante crittografia e modalità di utilizzo anonima.

L'obiettivo finale è creare un prodotto_e tecnologicamente avanzato ma anche eticamente responsabile, in grado di avere un impatto sociale positivo e concreto.

2.4.2 Caratteristiche funzionali_e

Le funzionalità_e principali richieste dall'azienda comprendono:

- **Rilevamento e Alert**: algoritmi di intelligenza artificiale per individuare pattern di rischio e situazioni di pericolo, con invio di notifiche silenziose a contatti fidati o centri di aiuto;

- **Risorse e Supporto:** accesso geolocalizzato a servizi di assistenza (centri antiviolenza, supporto legale, psicologico) e invio rapido di richieste di aiuto;
- **Funzionalità_G di Sicurezza_G Personalizzate:** modalità stealth, diario criptato e pianificazione di percorsi sicuri con mappe interattive;
- **Formazione e Prevenzione:** moduli_G multimediali, quiz e contenuti educativi per aumentare la consapevolezza e l'empowerment;
- **Community di Supporto (opzionale):** spazio anonimo e moderato per la condivisione di esperienze e il supporto reciproco.

Requisiti_G opzionali_G includono:

- Integrazione con dispositivi indossabili per l'invio discreto di allarmi;
- Funzionalità_G di "check-in" di sicurezza_G e modalità "fuga rapida";
- Analisi sonora ambientale per rilevare rumori associati a situazioni di pericolo;
- Gamification dei percorsi educativi e supporto multilingue esteso;
- Accessibilità per utenti con disabilità e conformità alle leggi locali e internazionali.

Dal punto di vista tecnico, l'app dovrà rispettare rigorosi requisiti_G di sicurezza_G (*security by design* e *privacy by design*), con crittografia AES-256, autenticazione a due fattori, audit trail e conformità GDPR_G.

2.4.3 Tecnologie proposte

Le tecnologie suggerite da Miriade per la realizzazione del progetto_G comprendono:

- **Mobile App:** sviluppo multi-piattaforma con il framework *Flutter* (fortemente consigliato).
- **Architettura:** basata su microservizi_G o su modello *serverless*, con deployment su cloud.
- **Cloud Provider:** preferibilmente *Amazon Web Services (AWS_G)*, con possibilità di alternative come Google Cloud.
- **back-end_G:** *AWS_G Lambda* per la logica applicativa, *Amazon API_G Gateway* per la gestione delle richieste e *AWS_G Step Functions* per i flussi complessi.
- **Database:** *Amazon DynamoDB* (NoSQL) per i dati ad alta scalabilità_G e *Amazon RDS* (PostgreSQL_G/MySQL) per dati strutturati.
- **Storage:** *Amazon S3* per contenuti multimediali e dati configurativi.
- **Autenticazione e Sicurezza_G:** *Amazon Cognito* per la gestione delle identità e l'autenticazione sicura.
- **AI_G e Machine Learning:** utilizzo di *Amazon SageMaker* e *Amazon Bedrock_G* per le funzionalità di analisi del linguaggio e del comportamento ("Detective delle Relazioni" e "Specchio Intelligente").
- **Comunicazione Asincrona:** *Amazon Kinesis* o *AWS_G SQS* per la gestione di code di messaggi.
- **Monitoraggio e Logging:** *AWS_G CloudWatch* per l'osservabilità e la gestione degli eventi di sistema.

Questa infrastruttura, basata su tecnologie cloud e AI_G, garantisce sicurezza_G, scalabilità_G e manutenzione semplificata nel tempo.

2.4.4 Chiarimenti e colloqui con l'azienda

Qui di seguito sono riportate alcune delle domande poste a Miriade in via telematica, con le relative risposte fornite dall'azienda.

Per ulteriori dettagli è possibile consultare il verbale esterno_e dell'incontro disponibile al seguente link: [Verbale Esterno_e - Miriade \(22/10/2025\)](#).

Nota: Le risposte dell'azienda sono state fornite in un colloquio in via telematica. Qui sono riportate riassunte e riformulate dal team in modo da essere più concise.

Domande	Risposte
C'è la possibilità di avere incontri online anche settimanalmente?	• L'azienda si organizza in sprint_e di 2 settimane. • è disponibile supporto anche al di fuori degli incontri programmati, tramite email o se richiesti anche tramite riunioni in sede a Padova.
L'azienda ha pensato ad un eventuale formazione iniziale rispetto alle tecnologie da utilizzare e alla tematica della violenza di genere?	• Sono disponibili corsi di formazione tecnica e il supporto di referenti specializzati per ogni area tecnica • In azienda lavora una referente che, purtroppo, ha vissuto in prima persona la violenza di genere e che può fornire un supporto prezioso per comprendere le esigenze delle vittime. • Miriade lavora con un'associazione che ha come fulcro il tipo di tematiche che l'applicazione tratta.
Considerata la complessità del progetto _e descritto nel capitolato _e , il team desidera sapere se è già stata effettuata una stima delle tempistiche per l'implementazione _e delle diverse funzionalità _e e se sono stati individuati dei requisiti _e minimi realizzabili entro il tempo assegnato.	• Il documento del capitolato_e fornisce una serie di aspettative per un'applicazione completa, ma i requisiti_e minimi non sono ancora stati definiti: in caso di conferma dell'assegnazione del capitolato_e, l'azienda lavorerà con il team per definire un MVP realistico. • Data la mancanza di requisiti_e minimi non possono ancora essere determinate le tempistiche per l'implementazione_e delle varie funzionalità_e.

2.4.5 Interesse del team

Il team ha riconosciuto l'importanza sociale del progetto_e proposto da Miriade e si è dimostrato, anche per questo, interessato. Inoltre, il team ha riconosciuto l'opportunità di implementare un'applicazione mobile completa, che copre l'intero ciclo di vita dello sviluppo software, dall'analisi dei requisiti_e allo sviluppo_e di un'applicazione non ancora esistente.

Tuttavia, sono emerse alcune preoccupazioni riguardo alla complessità del dominio delle richieste: sebbene le funzionalità_e indicate nel capitolato_e siano presentate come preferenziali e non come requisiti_e minimi, non è ancora chiaro quanto margine di contrattazione vi sia effettivamente con l'azienda. Ciò potrebbe rendere rischiosa una pianificazione basata sull'ipotesi di poter ridurre successivamente gli obiettivi del progetto_e.

2.4.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. Impatto sociale elevatissimo: Il progetto_e ha uno scopo nobile e concretamente utile, prevenire e supportare vittime di violenza di genere. 2. Buon supporto aziendale: Miriade offre un supporto completo: • Referenti specializzati per ogni area (tecnica, design, dominio sociale); • Formazione sulla tematica della violenza di genere, se richiesta; • Supporto multidisciplinare continuo; • Possibilità di incontri in sede. 3. Ottima possibilità riguardo il ciclo di vita_e dell'applicazione: analisi_e , progettazione_e , sviluppo_e , test_e di sicurezza_e e controllo dei contenuti etici devono essere fatti in modo chiaro e preciso, apprendendo così competenze multidisciplinari. 4. Integrazione di AI_e /LLM_e . Lavorare su queste tecnologie fornisce un'esperienza preziosa vista la loro crescente importanza nel settore.	1. Alta complessità tecnica e progettuale: • L'architettura proposta è ambiziosa e forse eccessiva essendo per alcuni il primo approccio verso certe conoscenze; • La sicurezza_e dei dati è critica e richiede implementazioni robuste. 2. Responsabilità e sensibilità del dominio: L'errore_e in un'app di questo tipo può avere conseguenze gravi, soprattutto dal punto di vista etico. La progettazione_e deve essere impeccabile sotto il profilo della sicurezza_e e dell'affidabilità_e. 3. Scope_e molto ampio: Le funzionalità_e previste sono numerose e ambiziose (rilevamento AI_e , allarmi silenziosi, diario criptato, moduli_e educativi, community). Il rischio_e di sovra-estendere il progetto_e è alto.

2.5 C5 - Nexum

2.5.1 Breve descrizione

Proponente_g : Eggon S.r.l.

Nexum nasce come piattaforma HR evoluta, in grado di connettere aziende, collaboratori e gli studi dei CdL_g. Il progetto_g prevede lo sviluppo_g di nuovi moduli_g per la piattaforma esistente, con l'obiettivo di migliorare l'efficienza_g dei processi HR e offrire funzionalità_g innovative ai suoi utenti. Il team si occuperà di sviluppare due moduli_g principali:

- **AI_g 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_g generativa_g, adeguando tono e stile della comunicazione a quello aziendale (formale, informale ...ecc).

- **AI_g Co-Pilot_g per i CdL_g:**

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

2.5.2 Caratteristiche funzionali_g

Tra i requisiti_g funzionali_g del progetto_g vi sono:

- Integrazione con la piattaforma esistente Nexum;
- Utilizzo di tecnologie cloud (AWS_g);
- Implementazione_g delle funzionalità_g di AI_g generativa_g e AI_g Co-Pilot_g;
- Le funzionalità_g devono essere disponibili sulla dashboard_g o sulla PWA_g.

Tra i requisiti_g non funzionali_g vi sono:

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

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

2.5.3 Tecnologie proposte

Le tecnologie proposte per lo sviluppo_g del progetto_g includono:

- **front-end_g** : Angular_g (dashboard_g amministrativa), Next.js (PWA_g utenti finali, hosting su AWS_g Amplify o S3+CloudFront);
- **back-end_g e API_g** : Ruby on Rails_g (stateless su ECS Fargate dietro ALB);
- **Background Jobs**: Sidekiq (su Fargate + SQS);
- **Database e cache**: PostgreSQL_g (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_g**: KMS (gestione chiavi per S3, RDS, Secrets), Secrets Manager (credenziali, API_g 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_g 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_g Config + GuardDuty.

2.5.4 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, sono state formulate domande riguardanti il supporto di Eggon durante lo sviluppo_g del progetto_g, l'importanza della conoscenza pregressa delle tecnologie richieste e il coinvolgimento del team nel processo SCRUM_g dell'azienda, al fine di comprendere come l'azienda avrebbe affrontato la gestione di un progetto_g di questo tipo, trattandosi del primo anno in cui Eggon vi partecipava.

Il team ha ritenuto opportuno fissare un incontro conoscitivo in modalità telematica, per discutere alcuni dettagli poco chiari e ottenere una visione più concreta e completa del progetto_g. Solo una parte delle domande riportate è stata trattata durante tale colloquio, mentre le restanti sono state effettuate per via telematica.

Per ulteriori dettagli è possibile consultare il verbale esterno_g dell'incontro disponibile al seguente link: [Verbale Esterno_g - Eggon \(24/10/2025\)](#).

Domande	Risposte
Le tecnologie richieste per lo sviluppo _g del progetto _g 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 _g ? Verrà fornito supporto o affiancamento nell'utilizzo di queste tecnologie?	• Conoscenza pregressa: non vincolante. Valutiamo impegno, qualità_g del codice e velocità di apprendimento. • Aspettative: avanzamento sprint_g -by-sprint_g, PR piccole e frequenti, test_g minimi, documentazione_g essenziale. • Supporto Eggon: kickoff e seed repository_g, canale e-mail/Telegram, code review e pairing su temi critici, sandbox_g (API_g 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 _g del lavoro e collaborazione con il vostro team interno?	• Ci lavoriamo spesso: funziona quando suddividiamo il lavoro in milestone_g piccole con demo frequenti, manteniamo standard chiari (lint/test_g/review) e i blocchi emergono subito. • Obiettivo: autonomia crescente — più guida all'inizio, più ownership col passare degli sprint_g.

Avete parlato di includere il team di lavoro nel vostro processo SCRUM_g 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_g principali (ad esempio sprint_g review o retrospective)?	• Sprint_g : 2 settimane. • Cerimonie_g : – Grooming/Planning ($\approx 1h$; nei primi sprint_g 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. – Lo costruiamo insieme attorno ai vostri impegni di studio (lezioni, esami, sessioni). – Una volta concordate milestone_g 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_g specifici prevedete di utilizzare? Oppure possiamo testare con diversi provider?	• Preferenza: AWS_g Bedrock_g (integrazione e governance). Tramite Bedrock_g possiamo usare più modelli (Claude, Llama, Mistral). • Apertura ad alternative: via libera a provider/idee creative, purché valutate su qualità_g output, aderenza al prompt_g, performance, costi_g e manutenibilità_g. • Requisito_g : adapter per evitare lock-in.

Nota: Le successive domande sono state esposte durante un incontro di approfondimento e conoscitivo con Eggon sulla piattaforma Google Meet_g. Le risposte dell'azienda non sono citazioni dirette, ma sono state riassunte e riformulate dal team in modo da essere più concise.

Domande	Risposte
Dovremo fare una parte di formazione iniziale per comprendere bene la piattaforma Nexum e l'integrazione _g tra i moduli _g ?	• Il dominio HR e CdL_g potrà essere appreso tramite interazioni con Eggon, in quanto conosce nel dettaglio come intende supportare i manager HR e i CdL_g nel loro lavoro. • Ci sarà una fase di onboarding iniziale in cui vi verrà spiegata la code base delle componenti_g Nexum che userete. • Ci aspettiamo una fase iniziale di studio e apprendimento delle tecnologie proposte.
Quali sono le vostre aspettative su questo progetto _g ?	Il progetto_g non è orientato esclusivamente al risultato finale, ma punta sul processo e sul confronto tra idee. La presenza di più team universitari è considerata un valore aggiunto, poiché favorisce la diversità di approcci e soluzioni. Inoltre, i requisiti_g e gli use case riportati nel documento sono solo abbozzati e ci aspettiamo che vengano ridefiniti sulla base delle vostre analisi e proposte.

Potreste spiegarci come vi aspettate che vengano utilizzati gli LLM _e nel progetto _e ? In particolare come debba l'AI _e rispecchiare i toni aziendali di ciascun cliente.	Si utilizzano modelli pre-addestrati come base, che vengono successivamente adattati al dominio specifico tramite gli strumenti messi a disposizione da alcune piattaforme, come ad esempio Amazon Bedrock _e , attraverso funzionalità _e quali Knowledge Base e Guardrails. A questo proposito, è fondamentale considerare gli aspetti di sicurezza _e , in particolare per prevenire possibili leak di informazioni tra diverse knowledge base aziendali, ad esempio causati da attacchi di prompt _e injection.
Quali tecnologie ritenete possano presentare le principali criticità o difficoltà di implementazione _e per il team? Avete dei consigli a riguardo?	La documentazione di AWS _e risulta in alcuni casi poco chiara. Il codice di back-end _e attualmente disponibile deriva da un progetto _e sviluppato alcuni anni fa e successivamente aggiornato alle versioni più recenti di Ruby on Rails _e . Tale codice presenta alcune parti ridondanti o non più necessarie, che potrebbero generare confusione durante lo sviluppo. È prevedibile che l'integrazione con AWS _e rappresenti una delle principali difficoltà; si consiglia pertanto di esaminare le diverse librerie e SDK disponibili per Bedrock _e , preferibilmente in linguaggi già noti al team, per comprendere il funzionamento del servizio. Una volta acquisite le necessarie competenze, sarà possibile effettuare una migrazione del codice in Ruby on Rails _e .

2.5.5 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_e 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_e. 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_e.

2.5.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. Prodotto_e reale e integrazione con piattaforma esistente: Nexum è una piattaforma HR già operativa. Sviluppare moduli_e che si integreranno in un prodotto_e commerciale fornisce un'esperienza di lavoro su codice legacy e integrazione con sistemi esistenti. 2. Processo SCRUM_e realistico (rispettare le scadenze dell'azienda). 3. Ottimo supporto dell'azienda: l'azienda ha un piano ben strutturato quali le cerimonie_e 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_e è continuo (anche formazione e mentoring). 4. Tecnologie definite esaustivamente e requisiti_e del progetto_e chiari, con due requisiti_e opzionali definiti. 5. L'azienda ha fornito un documento per il capitolato_e completo di casi d'uso_e e obiettivi misurabili in milestone_e, rendendo la pianificazione. 6. Integrazione di AI_e. Lavorare su queste tecnologie fornisce un'esperienza preziosa vista la loro crescente importanza nel settore. 7. Abbiamo particolarmente apprezzato l'obiettivo finale di Eggon, volto a coinvolgere giovani per permettere loro di conoscere l'azienda e farsi conoscere, offrendo anche un punto di vista fresco e diverso.	1. Integrazione con una piattaforma esistente: Nexum è già una piattaforma operativa; sviluppare nuovi moduli_e comporta difficoltà legate alla comprensione e modifica di codice preesistente, con rischi di incompatibilità e maggiore complessità tecnica. 2. Vincoli progettuali e tecnologici: L'integrazione in un prodotto_e reale impone vincoli rigidi, definiti dagli standard e dalle esigenze aziendali, che limitano la flessibilità_e del team e aumentano la complessità dello sviluppo. 3. Alta complessità riguardante le tecnologie da utilizzare con AWS_e, OCR_e (Optical Character Recognition), e integrazione dell'AI_e.

2.6 C6 - Second Brain

2.6.1 Breve descrizione

Proponente_e : Zucchetti s.p.a.

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_e) per assistere l'utente nella scrittura e revisione dei testi. In particolare l'applicazione dovrà essere in grado di interagire con l'AI_e per eseguire operazioni come riassumere, riscrivere, tradurre e criticare il testo scritto dall'utente. Inoltre, l'applicazione dovrà supportare un prompt_e generativo associato all'editor per permettere la generazione automatica di testi.

2.6.2 Caratteristiche funzionali_e

Tra i requisiti_e funzionali_e principali del progetto_e, 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_G** : integrazione con un *Large Language Model_G* in grado di operare sull'intero testo o su porzioni selezionate.
4. **Comandi di base basati su AI_G** :
 - Riassunto del testo.
 - Riscrittura automatica del testo.
 - Traduzione del testo in una lingua differente.
5. **Critica del testo:** implementazione_G di comandi di analisi secondo il modello dei "*sei cappelli per pensare*" di *Edward De Bono*.
6. **Prompt_G generativo:** gestione di un prompt_G 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_G (e in particolare del concetto dei 'Sei cappelli per pensarÈ) 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_G in questa direzione, sarebbe possibile anche costruire un sistema di *linking* tra le note.

2.6.3 Tecnologie proposte

- **front-end_G** : HTML5 per l'interfaccia_G web dell'editor; linguaggio di markup *Markdown* per la scrittura.
- **Integrazione LLM_G** : API_G compatibili OpenAI per chiamate a modelli linguistici; supporto a modelli come *Gemini* (Google), *Mistral*, *Gemma*; uso opzionale di *llama.cpp* per esecuzione locale.
- **back-end_G e API_G (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_G conversazionale.

2.6.4 Chiarimenti e colloqui con l'azienda

Domande	Risposte
Quali modelli LLM _G saranno disponibili e con che modalità potremo accedervi? Possiamo utilizzare modelli diversi durante il progetto _G ?	• Modelli forniti: Claude e Gemini accessibili dall'esterno; Gemma, Deepseek e Qwen disponibili su infrastrutture interne BugBuster (tra 7B e 32B, versioni quantizzate). • Accesso: verrà fornita una API_G key per connettersi a un endpoint dedicato gestito dall'azienda. • Utilizzo: i modelli potranno essere usati anche per attività di traduzione e analisi linguistica.

È previsto un database specifico per l'applicazione? Ci sono preferenze o vincoli tecnologici?	• Database consigliato: PostgreSQL_G. • Vincoli: nessuno — il team può scegliere liberamente soluzioni alternative se più adatte.
È richiesta una gestione dell'autenticazione utenti? Se sì, quale approccio consigliate?	• Autenticazione: non richiesta come requisito_G obbligatorio_G. • Suggerimento: se si desidera includerla, utilizzare un sistema basato su token <i>OAuth</i>.
L'applicazione dovrà includere funzioni di traduzione automatica o comparazione tra modelli linguistici?	• Traduzione: gli LLM_G forniti possono essere impiegati anche per traduzioni automatiche. • Comparazione: non richiesta, ma la possibilità di scegliere il modello di LLM_G da utilizzare è considerata una funzionalità_G opzionale_G interessante. • Sistema di confronto: non richiesto — era parte del progetto_G dell'anno precedente.
Avete preferenze sui framework di sviluppo o sull'architettura (container, PWA _G , app nativa)?	• Framework: completa libertà di scelta. • Applicazione: non è richiesta un'app nativa. È sufficiente che l'interfaccia_G sia <i>responsiva</i> e utilizzabile su PC e tablet. • PWA_G: facoltativa; può essere sviluppata a discrezione del team. • Container: scelta libera. L'app può essere eseguita in modo tradizionale oppure su container. BugBuster utilizza <i>Docker</i> o <i>Podman</i> senza orchestratore, oppure <i>Kubernetes</i> in ambienti completi.
Il progetto _G prevede aspetti di sicurezza _G applicativa o gestione avanzata degli accessi?	• Sicurezza_G: non richiesta come parte obbligatoria del progetto_G. • Motivazione: una trattazione completa del tema richiederebbe un impegno eccessivo rispetto ai tempi previsti.

2.6.5 Interesse del team

Il team ritiene che il capitolato_G proposto da Zucchetti sia realistico per un gruppo di studenti universitari che si confrontano per la prima volta con un progetto_G di questo calibro, sia dal punto di vista delle tempistiche sia delle competenze richieste. Inizialmente, l'interesse del team era elevato, in quanto l'argomento dell'integrazione di modelli di linguaggio di grandi dimensioni (LLM_G) in applicazioni pratiche risulta estremamente attuale e rilevante nel contesto tecnologico odierno. Tuttavia, il team non lo ha ritenuto sufficientemente stimolante.

2.6.6 Punti di forza e di debolezza

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

1. L'integrazione degli LLM_e in un'applicazione fornisce un'esperienza preziosa vista la loro crescente importanza nel settore. Lavorare su questo progetto_e fornisce un'esperienza diretta con questo tipo di tecnologia. 2. Ampio spazio per la progettazione e la ricerca: Il capitolato_e non specifica esattamente come implementare le funzionalità_e, ma si concentra sul cosa. Questo lascia al team una grande libertà nelle scelte architetturali, nella progettazione dell'interfaccia_e utente (UI/UX) e, soprattutto, nell'ingegnerizzazione dei prompt_e per l'LLM_e, che è il cuore del progetto_e. 3. Zucchetti si offre esplicitamente di supportare il team nelle parti più complesse (es. configurazione API_e LLM_e) e metterà a disposizione gli LLM_e stessi. 4. Tempi di realizzazione adeguati: Zucchetti ha esplicitamente confermato che aggiungere altre funzionalità_e (es. tema della sicurezza_e) potrebbe impegnare al punto da sfiorare i tempi previsti per il progetto_e. Questo indica una consapevolezza realistica delle tempistiche e delle capacità del team.	1. Le tecnologie non risultano chiaramente definite e i requisiti_e di progetto_e sono ampi. Inoltre, l'assenza di specifiche vincolanti e di requisiti_e opzionali_e chiaramente delineati comporta un'eccessiva libertà decisionale, che può generare incertezza nella pianificazione e aumentare il rischio_e di deviazioni dagli obiettivi progettuali. 2. L'azienda propone lo sviluppo di un'applicazione web basata su HTML per la parte front-end_e. Tuttavia, il team di progetto_e, 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>_e.
---	---

2.7 C7 - Sistema di acquisizione dati da sensori

2.7.1 Breve descrizione

Proponente_e : M31 S.r.l.

Il progetto_e 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 e interfacce_e di visualizzazione, garantendo segregazione tra diversi tenant.

I primi due livelli (sensori BLE e gateway) non sono oggetto di questo capitolato_e e devono essere considerati come strumenti esterni. Nel contesto del progetto_e, 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.

2.7.2 Caratteristiche funzionali_e

Tra i requisiti_e funzionali_e del progetto_e 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_e, 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 l'isolamento logico tra diversi tenant, permettendo a ciascun cliente/utente di accedere solo ai propri dati.
4. **Esposizione API_c centralizzate:** i dati raccolti dal simulatore e gestiti dal cloud devono essere accessibili tramite API_c sicure, documentate e versionate (test_c).
 - (a) ogni client deve essere identificato univocamente e certamente, per garantire il controllo dell'accesso ai dati e la loro segregazione;
 - (b) le API_c 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_c 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_c non funzionali_c vi sono:

- **Scalabilità_c:** il sistema cloud deve progettato per essere scalabile orizzontalmente per gestire un numero crescente di sensori, gateway e tenant senza degrado delle prestazioni;
- **Test_c automatizzati e code coverage_c:** lo sviluppo deve includere test_c unitari e di integrazione con un livello minimo di copertura da concordare ad inizio progetto_c ;
- **Monitoraggio:** devono essere predisposti strumenti di monitoraggio in tempo reale per le performance del sistema. In aggiunta, devono essere presenti almeno funzionalità_c di alert di base che consentano di capire rapidamente se un gateway sta funzionando correttamente o risulta non raggiungibile;
- **Versionamento_c /versioning_c e DevOps:** il codice deve essere gestito tramite Git con pipeline CI/CD per il deploy su ambienti di test_c e produzione.

2.7.3 Tecnologie proposte

Le tecnologie proposte per lo sviluppo del progetto_c includono:

- **Sviluppo:** interfaccia_c utente in **Angular_c** per la creazione di una **SPA** (Single Page Application); microservizi, sviluppati con **Node.js** e **Nest.js** in **TypeScript** per la parte di API_c .
- **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_c* 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.

2.7.4 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_c , poiché 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_c .

Per ulteriori dettagli è possibile consultare il verbale esterno_c dell'incontro disponibile al seguente link: [Verbale Esterno_c - M31 \(24/10/2025\)](#).

Nota: Le risposte dell'azienda sono state fornite in un colloquio in via telematica. Qui sono riportate riassunte e riformulate dal team in modo da essere più concise.

Domande	Risposte
Modello di identità e provisioning dei gateway simulati • 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: – Provisioning manuale (statico) – Provisioning basato su certificati X.509 (auto-generati) – Provisioning Just-In-Time (JIT) – Provisioning Just-Enough-Time (JET)	Ci aspettiamo che lo decidiate voi, dopo un attento studio. L'idea è far sì che per il PoC_g questo avvenga manualmente attraverso l'interfaccia_g.
Simulazione sensori e profili BLE • Potete indicarci quali sensori dobbiamo simulare e, di conseguenza, quali profili BLE standard è opportuno utilizzare? • Avete già definito eventuali profili custom?	Alcuni esempi di sensori sono di: heart rate, temperatura, pressione sanguigna, saturazione ossigeno, ECG, glicemia.

Le tecnologie richieste per lo sviluppo del progetto_e 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_e ?	• 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_e proposto l'anno scorso, di complessità analoga. • Siamo consapevoli del numero limitato di ore a vostra disposizione e della complessità — sicuramente stimolante — del progetto_e , 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_e per la dashboard_e . • Questi dettagli, tuttavia, li definiremo progressivamente, dopo che avrete avuto modo di condurre un adeguato studio preliminare del progetto_e , 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_e ?	Non è previsto un supporto attivo, come corsi o lezioni dirette, e riteniamo che non sia necessario, poiché parte integrante del progetto_e 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.

Oltre alle domande già discusse, la proponente_e ha ridefinito gli obiettivi del progetto_e in seguito a una richiesta esplicita del team. L'azienda ha inoltre sottolineato che lo studio architettuale e documentale riveste per loro un ruolo centrale, mentre l'applicazione rappresenta principalmente il risultato di tale studio; l'interfaccia_e utente è considerata tecnicamente di minor rilevanza.

2.7.5 Interesse del team

L'interesse del team verso questo capitolato_e è stato inizialmente moderato, ma è cresciuto significativamente dopo aver approfondito i dettagli del progetto_e e aver interagito con l'azienda proponente_e , 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_e .

2.7.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. Il progetto_e 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. 2. Buon supporto da parte dell'azienda; eventualmente anche incontri in sede molto vicini alle nostre sedi di studio. 3. Forte opportunità di apprendimento: il progetto_e 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 e introduce tecnologie come Node.js e Nest.js con TypeScript (back-end_e), Go (componenti ad alte prestazioni), Angular_e (front-end_e SPA), MongoDB (database NoSQL), PostgreSQL_e (database SQL), Redis (caching), Google Cloud Platform (infrastruttura cloud), Kubernetes (orchestrazione container), NATS o Kafka (comunicazione asincrona), e Prometheus con Grafana (monitoring).	1. 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_e richiede competenze solide e un'attenta progettazione. 2. I requisiti_e sono molti e articolati (dalle API_e on-demand e streaming, alla UI, al monitoring, alla sicurezza_e). C'è il concreto rischio_e di dover sacrificare la profondità di alcune funzionalità_e per coprirne la quantità entro i tempi del progetto_e. 3. Testare un'architettura distribuita è intrinsecamente difficile. Configurare ambienti di test_e end-to-end, simulare guasti_e e testare la scalabilità_e richiederà sforzi notevoli e una buona pianificazione. 4. 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 (ad esempio 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.

2.8 C8 - Smart Order

2.8.1 Breve descrizione

Proponente_e : Ergon Informatica S.r.l.

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_e è quello di migliorare l'efficienza_e dei processi aziendali, riducendo al minimo l'intervento umano nelle fasi più ripetitive e soggette a errore_e.

2.8.2 Caratteristiche funzionali_e

Tra i requisiti_e funzionali_e del progetto_e 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_e di tutti i layer funzionali descritti nel capitolato_e.
- **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_e automatica:** devono essere previsti controlli di coerenza e integrità dei dati, per garantire che gli ordini generati siano affidabili e corretti.
- **Interfaccia_e e API_e di comunicazione:** il sistema deve prevedere API_e REST per l'interazione tra il modello AI_e 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_c non funzionali_c vi sono:

- **Modularità_c e scalabilità_c del sistema:** ogni componente della pipeline deve essere progettato come modulo_c indipendente, aggiornabile senza impattare l'intero sistema.
- **Logging e feedback continuo:** implementazione_c di un sistema di monitoraggio e logging che consenta di raccogliere errori_c, aggiornare le regole aziendali e migliorare le prestazioni tramite retraining periodico.
- **Documentazione_c 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.

2.8.3 Tecnologie proposte

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

2.8.4 Chiarimenti e colloqui con l'azienda

Il team non ha contattato l'azienda né per ricevere ulteriori informazioni né per richiedere colloqui.

2.8.5 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 multimediale. L'architettura proposta è risultata flessibile e ben strutturata, permettendo l'uso di tecnologie moderne come LLM_c, OCR_c 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_c e la possibilità di lavorare su un caso di studio reale fornito da Ergon.

Tuttavia nel complesso il capitolato_c non ha suscitato particolare interesse al team.

2.8.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. 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_e d'avanguardia. 2. Viene fornito un ricco elenco di tecnologie suggerite per ogni componente (LLM_e , OCR_e , Speech-to-Text, UI), ma con la libertà di scegliere alternative. Questo permette al team di adattare lo stack tecnologico alle proprie competenze.	1. Integrare e far collaborare modelli di AI_e diversi (LLM_e , Vision, Audio) in una pipeline coerente è una sfida tecnica notevolissima. 2. Il team dovrebbe padroneggiare o imparare rapidamente una vasta gamma di tecnologie AI_e e di framework, ciascuna con le sue complessità. 3. Il pre-processing di dati multimodali (pulizia del testo, elaborazione di immagini, trascrizione audio) è un compito laborioso e critico. La qualità_e dell'output finale dipende fortemente da questa fase. 4. I modelli di AI_e , specialmente gli LLM_e , possono essere imprevedibili. Validare l'accuratezza e l'affidabilità_e del sistema in scenari reali, dove un errore_e può significare un ordine sbagliato, è una sfida significativa.

2.9 C9 - View4Life

2.9.1 Breve descrizione

Proponente_e : Vimar s.p.a

Il capitolato_e 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, con l'intento di migliorare la sicurezza_e, il comfort e l'efficienza energetica degli ambienti. Il progetto_e 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.

2.9.2 Caratteristiche funzionali_e

Tra i requisiti_e funzionali_e vi sono:

- L'applicativo web deve essere **responsive**, adattandosi a tutti i dispositivi.
- Deve possedere un **design intuitivo e semplice**.
- L'applicativo deve **interfacciarsi con uno o più impianti Vimar View Wireless**.
- Deve **utilizzare l'interfaccia_e API_e KNX IoT** fornita da Vimar per comunicare con gli impianti.
- Deve includere un **sistema di accesso utente** dedicato al personale sanitario.
- Deve prevedere un **sistema di gestione degli allarmi** per gli operatori sanitari, ad esempio:
 - Allarme per **rilevamento cadute**.
 - Allarme per **accesso indesiderato a un'area in una certa fascia oraria**.
 - Allarme **azionato manualmente dall'utente**.
- Deve avere una **sezione dedicata ai dispositivi dell'impianto**, mostrando le informazioni relative a ciascuno.
- Deve includere una **sezione "Analytics"** che presenti:
 - **Statistiche della piattaforma e dell'impianto** sotto forma di grafici.
 - **Suggerimenti per la riduzione dei consumi energetici**, basati sui dati raccolti dai sensori.
- Deve fornire un **cruscotto_e /dashboard_e informativo (dashboard_e)** per l'utente.

Tra i requisiti_e non funzionali_e vi sono:

- L'applicativo deve essere di **facile ed efficace utilizzo ed essere accessibile da qualsiasi dispositivo**.
- L'infrastruttura cloud deve essere **basata su container**.
- L'infrastruttura deve essere progettata secondo il principio di **Infrastructure as Code (IaC)**, evitando configurazioni manuali e favorendo l'uso di file di configurazione versionati.
- L'infrastruttura deve essere **replicabile, tracciabile e facilmente manutenibile**.
- L'intera infrastruttura must essere **realizzata su Amazon Web Services (AWS_e)**.

2.9.3 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_e**.
- **Autenticazione:** l'interfaccia_e 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_e /versioning_e** : il repository_e di lavoro deve essere gestito con **Git** e reso **pubblicamente accessibile**. I sorgenti dovranno essere rilasciati con **licenza open source**.

2.9.4 Chiarimenti e colloqui con l'azienda

Le domande al team Vimar sono state inoltrate via email, tramite un altro gruppo di progetto_e, su richiesta esplicita dell'azienda, che ha preferito ricevere tutte le domande raggruppate per comodità.

Domande	Risposte
Abbiamo compreso che View4Life nasce come un dimostratore per mostrare quanto sia semplice e veloce realizzare un'integrazione con le tecnologie pubblicizzate da Vimar. Il contesto presentato, però, suggerisce uno scenario _e molto realistico e vicino a un possibile bisogno di mercato. Questo progetto _e deriva da richieste concrete o rimane principalmente un proof-of-concept dimostrativo? E, nel caso, pensate che possa avere in futuro uno spazio effettivo all'interno di Vimar?	• Il contesto presentato ha in realtà una duplice valenza, arrivata specialmente negli ultimi mesi. • Dimostratore per integrazioni 3rd party via Cloud: mostrare che si possono controllare i dispositivi Vimar View Wireless via Cloud con API_e REST. • Dimostratore per esigenza di mercato: esplorare cosa potrebbe essere utile in base alle richieste commerciali. • Se l'output del progetto_e farà l'effetto WOW, ci sarà uno spazio effettivo; altrimenti sarà comunque pubblico con il vostro nome sopra.
Quanto è rilevante per voi la conoscenza pregressa delle tecnologie richieste, o è previsto che ci sia spazio per apprendere progressivamente durante il progetto _e ?	• La conoscenza pregressa non è quasi mai prevista: ci sarà tempo per imparare insieme. • La curva di apprendimento seguirà l'effetto Dunning-Kruger. • Sessioni di approfondimento saranno organizzate durante il progetto_e, ad esempio su Docker e KNX IoT. • Framework di sviluppo e altri strumenti saranno affrontati secondo le esigenze del team.
È preferibile sviluppare parte di front-end _e , back-end _e (ed eventuali microservizi _e) singolarmente e in maniera seriale, oppure sviluppare tutto in parallelo?	• Generalmente conviene il parallelismo per ridurre il tempo totale. • La strategia seriale può essere una buona alternativa, considerando i ruoli da ruotare. • Dal front-end_e si può lavorare sul mocking delle API_e prima che siano implementate. • L'approccio finale sarà discusso in base allo skillset del gruppo.
La documentazione per integratori di terze parti per interfaccia _e KNX IoT richiede più o meno quanta mole di lavoro per poterla capire e padroneggiare? O comunque il livello di difficoltà/quantità di documentazione è molto consistente?	• La documentazione è approfondita, ma i concetti base devono essere padroneggiati bene. • Verrà organizzata una sessione di presentazione di KNX IoT e dei dispositivi via API_e. • Saranno forniti mini Jupiter Notebook o Postman per capire i primi passi.
Si sviluppa prima dall'idea o in maniera test-driven?	• Non esiste un giusto o sbagliato: dipende dallo stile del team. • Generalmente si parte dalla progettazione, poi implementazione_e test-driven o modulare.

Un requisito _c opzionale _c riguarda la possibilità di inserire una mappa degli appartamenti. L'azienda aveva pensato alla modalità di visualizzazione della mappa? Come viene aggiunta? È interattiva? (ad esempio cliccare i dispositivi, ingrandire etc)	• La mappa degli appartamenti è opzionale: può essere un rendering basico con framework tipo Floorspace.js o una foto con box cliccabili. • La mappa non deve essere modificabile, ma mostrare la possibilità di utilizzo.
--	---

2.9.5 Interesse del team

Il gruppo ha valutato molto positivamente questo capitolato_c per la concretezza e l'impatto sociale: lavorare su un sistema capace di migliorare la qualità_c della vita e la sicurezza_c delle persone rappresenta una forte motivazione, dando al progetto_c un valore tangibile e significativo. Sono stati inoltre apprezzati l'utilizzo di tecnologie moderne e versatili, come Docker, AWS_c e framework web come Angular_c 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_c 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_c così articolato permetterà infatti di sviluppare competenze di organizzazione, coordinamento e gestione tecnica avanzata.

2.9.6 Punti di forza e di debolezza

Punti di forza	Punti di debolezza
1. 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_c e il benessere delle persone è un forte motivatore e rende il progetto_c molto più significativo rispetto a un dominio astratto. 2. Tecnologie moderne e ricercate. 3. 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. 4. Opportunità reale che l'MVP venga effettivamente adottato da Vimar, qualora raggiungesse la qualità_c necessaria.	• Elevata complessità e ampio scope_c: La completezza del progetto_c è anche la sua principale sfida. Il team deve essere bravo a gestire la complessità e a prioritizzare le funzionalità_c obbligatorie_c, per non perdersi negli optional.

3 Conclusioni

La seguente tabella fornisce una sintesi immediata dei capitolati_c analizzati; per una descrizione più approfondita si rimanda alla Sezione 3.

Capitolato _c	Interesse iniziale	Analisi _c approfondita*	Motivo di scarto / proseguimento
C1 – Automated EN18031	Basso	No	Poco stimolante.
C2 – Code Guardian	Basso	No	Architettura troppo complessa.
C3 – DIPReader	Basso	No	Poco stimolante.
C4 – L'app che Protegge e Trasforma	Medio	Sì	Grande impatto sociale ma complessità tecnica elevata.
C5 – Nexum	Alto	Sì	Buon equilibrio tra sfida tecnica e supporto aziendale.
C6 – Second Brain	Medio	Sì	Interessante ma troppo libero e poco vincolato.
C7 – Sistema di acquisizione dati da sensori	Alto	Sì	Molto tecnico e infrastrutturale, buon supporto aziendale.
C8 – Smart Order	Basso	No	Interessante ma non particolarmente stimolante.
C9 – View4Life	Alto	Sì	Innovativo e utile, buon supporto aziendale.

* Per “Analisi approfondita” si intende il coinvolgimento dell'azienda per approfondire alcuni aspetti del progetto_c o porre domande su eventuali dubbi.