



byteyourdreams.swe@gmail.com

Valutazione Capitolati

Informazioni documento

Data	28/10/2024
Redattore	A.M. Margarit L. Zanesco
Verificatore	Y. Huang A. Mio O.F. Stiglet
Amministratore	L. Albertin
Destinatari	T. Vardanega R. Cardin

Il responsabile: L. Albertin



Registro delle modifiche

Versione	Data	Autore	Verificatore	Dettaglio
0.0.1	28/10/2024	L. Zanesco A.M. Margarit	L. Albertin O.F. Stiglet A. Mio Y. Huang	Redazione

Indice

Byte Your Dreams

ottobre 28, 2024

Contents

1	Valutazione del capitolato scelto	4
1.1	Capitolato C2 - Vimar GENIALE	4
2	Valutazione sui capitolati rimanenti	5
2.1	Capitolato C1 - Artificial QI	5
2.2	Capitolato C3 - Automatizzare le routine digitali tramite l'intelligenza generativa	6
2.3	Capitolato C4 - NearYou - Smart custom advertising platform	7
2.4	Capitolato C5 - 3DataViz	8
2.5	Capitolato C6 - Sistema di gestione di un magazzino distribuito	9
2.6	Capitolato C7 - LLM: Assistente virtuale	10
2.7	Capitolato C8 - Requirement Tracker - Plug-in VS Code	11
2.8	Capitolato C9 - BuddyBot	12
3	Conclusioni	13



1 Valutazione del capitolato scelto

1.1 Capitolato C2 - Vimar GENIALE

Proponente: *Vimar S.p.A*

Obiettivo: Sviluppare un *applicativo_G* che gli installatori possono interrogare per reperire informazioni testuali e grafiche sui prodotti Vimar presenti all'interno del sito ufficiale. L'azienda, con il *progetto_G* in questione, mira a supportare gli installatori nella fase di progettazione dei diversi tipi di impianti, integrando e completando le loro conoscenze oppure, semplicemente, chiarendo qualche loro dubbio durante l'effettiva fase di installazione dei dispositivi.

Tecnologie Suggerite:

- **Flask, Angular o VueJS:** utilizzati a livello di applicativo web *responsive_G* per lo sviluppo *front-end_G*.
- **Python:** consigliato come linguaggio di programmazione per lo sviluppo del *progetto_G*, data la sua semplicità di utilizzo e apprendimento.
- **Scrapy e OCRmyPDF:** librerie di Web Scraping e OCR utilizzate per la fase di estrazione e reperimento delle informazioni dal sito web.
- **PostgreSQL:** database *relazionale_G* per immagazzinare i dati, unito all'uso dell'estensione *pgvector* per realizzare indici vettoriali. In **alternativa** si possono utilizzare database *NoSQL_G* come *TimescaleDB* o *InfluxDB*.
- **Llama 3.1, Mistral, Bert o Phi:** modelli open source_G di *LLM_G* utilizzati come componente di interrogazione.
- **AWS LightSail o AWS EC2:** servizi utilizzati per soluzioni containerizzate con sviluppo su *AWS_G*.
- **Github Runners o altro:** *CI_G* per automatizzare l'esecuzione di *test_G* e l'analisi statica del codice.
- **Github Copilot o Amazon Q:** per lo sviluppo *software_G*.

Considerazioni:

- Stimolante l'utilizzo degli *LLM_G* in un ambito così pratico e richiesto anche in diversi settori lavorativi.
- Con un'interfaccia *user-friendly_G* progettata per i professionisti del settore, Vimar GENIALE consente agli installatori di reperire rapidamente dettagli tecnici e visualizzare schemi elettrici, semplificando notevolmente il loro lavoro quotidiano.
- Il *progetto_G* prevede un'applicazione web *responsive_G*, accessibile da smartphone, tablet e desktop, permettendo la consultazione in mobilità e direttamente sul luogo di installazione.
- La natura *open source_G* del *progetto_G* garantisce trasparenza e facilita il contributo della *community_G*, permettendo la futura espandibilità e il miglioramento continuo dell'*applicativo_G*.
- Vimar mette a disposizione strumenti e risorse, inclusa una *base dati_G*, supporto tecnico per la fase iniziale e sessioni di avanzamento lavori regolari, facilitando un progresso continuo del *progetto_G*.
- Ogni membro del gruppo ha ritenuto molto utile ed interessante l'apprendimento delle tecnologie proposte.
- Con una copertura di *test_G* richiesta, il *progetto_G* garantisce affidabilità e robustezza del *system_G*, migliorando l'esperienza e la soddisfazione dell'*utente_G* finale.
- Vimar propone una gestione degli avanzamenti che supporta notevolmente la fase di sviluppo del *progetto_G* con conseguenti migliorie generali.

2 Valutazione sui capitolati rimanenti

2.1 Capitolato C1 - Artificial QI

Proponente: *Zucchetti SPA*

Obiettivo: realizzare un *programma_G* che permetta di valutare la capacità di rispondere ad una lista di problemi/domande di un *sistema_G* che usa tecniche di *AI_G* con il fine di far comprendere, in modo esaustivo, al *programmatore_G* che utilizza il *sistema_G* di test, gli effetti delle varie scelte che tale *infrastruttura_G* farà di volta in volta, pervenendo a quella più performante.

Tecnologie Suggerite:

Per lo sviluppo dell'*applicazione_G* proposta, orientata alla valutazione e al testing di modelli basati su *LLM_G*, è importante scegliere tecnologie affidabili, scalabili e orientate al lavoro con *AI_G* e *NLP_G* (Natural Language Processing). Di seguito sono riportate alcune tecnologie utili:

- **Python:** Le librerie *Python_G* facilitano l'integrazione con *LLM_G*, l'elaborazione del linguaggio e la gestione delle *API_G*.
- **FastAPI o Flask:** *framework_G Python_G* leggeri, ideali per creare *API REST_G*.
- **PostgreSQL o MySQL:** database *relazionali_G* per archiviare in modo strutturato domande, risposte attese, risposte ottenute e risultati dei *test_G*.
- **MongoDB:** database *NoSQL_G*, che permette maggiore flessibilità nel caso sia necessario archiviare dati *non strutturati_G*.
- **spaCy e NLTK:** librerie per l'elaborazione del linguaggio *naturale_G* che includono strumenti per tokenizzazione_G, estrazione di entità, e gestione di sinonimi.
- **React o Vue.js:** *framework_G frontend_G* popolari per creare interfacce utente_G interattive, adatte per applicazioni *web_G*. Consentono la creazione di *dashboard_G* per la visualizzazione dei risultati dei *test_G* e l'interazione con i vari set di domande/risposte.
- **Plotly o Chart.js:** librerie di visualizzazione grafica per rappresentare in modo efficace i risultati dei *test_G*, come l'accuratezza delle risposte, gli indici di *performance_G* e i confronti tra esecuzioni diverse.
- **Redis o SQLite:** *Redis* può essere usato per archiviare in *cache_G* i risultati dei *test_G*, mentre *SQLite* può essere utilizzato come archivio semplice e veloce.
- **Pytest o Unittest:** strumenti per l'automazione dei *test_G* dell'*applicazione_G*.
- **Docker:** facilita la *containerizzazione_G* e il *deployment_G* dell'*applicazione_G* su ambienti locali o server_G remoti, garantendo la consistenza dell'ambiente.

Considerazioni:

- Attraverso *test_G* strutturati e *report_G* dettagliati, il *progetto_G* consente di individuare punti di forza e debolezza dei modelli, ottimizzando così la qualità delle risposte e l'affidabilità del *system_G*, elemento cruciale per l'esperienza utente.
- Questo *progetto_G* permette di lavorare con tecnologie all'avanguardia, rafforzando competenze in un settore in forte espansione.
- La soluzione può essere applicata in diversi settori grazie alla flessibilità e adattabilità nella valutazione di linguaggi naturali.
- Se adottato in contesti aziendali, un *sistema_G* che consente una valutazione preventiva può ridurre i costi di integrazione di modelli meno performanti.



2.2 Capitolato C3 - Automatizzare le routine digitali tramite l'intelligenza generativa

Proponente: Var Group S.p.A.

Obiettivo: Sviluppare un *servizio_G* ad agenti dove gli *utenti_G* possono disegnare localmente un *workflow_G* sfruttando le *API_G* dei *software_G* locali e l'intelligenza artificiale_G in *cloud_G* per automatizzare attività quotidiane che l'*utente_G* svolge manualmente. Scendendo, quindi, nei particolari, l'obiettivo del *progetto_G* si suddivide nei seguenti punti:

- Realizzare un *sistema_G* di *Gen AI_G* in *cloud_G* in grado di ricevere dati *software_G* locali e produrre delle *automazioni_G* da eseguire localmente o direttamente tramite *API_G* ai *servizi_G* finali.
- Creare un *applicativo_G* cliente per Mac o PC che permetta agli *utenti_G* di disegnare dei *workflow_G*.
- Creare un *repository_G* di "blocchi" funzionali che accedono alle *API_G* applicative.
- Per ogni passo del *workflow_G*, l'*utente_G* in *linguaggio naturale_G* può descrivere l'attività di logica e il *software_G* deve creare la logica di *automazione_G* in autonomia.

Tecnologie Suggeste: Per la creazione di *agenti_{software G}*, il *proponente_G* consiglia l'utilizzo delle seguenti tecnologie:

- **Python** o **C** (in caso di ambiente Apple: **Swift**).
- **MongoDB** o altro *database_G* locale.
- **React** (in caso di ambiente Apple: **Swift UI**): per la realizzazione di interfacce applicative *G* web.

Mentre, per quanto riguarda i linguaggi di programmazione per lo sviluppo *API_G* *cloud_G*, l'azienda *proponente_G* suggerisce:

- **NodeJS**
- **Python**
- **Typescript**

Considerazioni:

- La partecipazione a questo *progetto_G* consente di acquisire competenze avanzate in *cloud_G*, *AI_G*, sviluppo *API_G*, e interfacce utente cross-platform, conoscenze oggi molto richieste sul mercato.
- La flessibilità consentita e richiesta nello sviluppo di questo *progetto_G* renderà l'applicazione_G adattabile a diverse esigenze e contesti aziendali.
- Il *team_G* sarà coinvolto in tutte le fasi di sviluppo del *progetto_G*: questa esperienza completa permette di acquisire una visione olistica dello sviluppo del *software_G* e delle buone pratiche dell'ingegneria del *software_G*.
- Gli *utenti_G* potranno definire flussi specifici attraverso *linguaggio naturale_G*, permettendo al *sistema_G* di interpretare e automatizzare task con precisione. Questa capacità di personalizzazione permette alle aziende di ottenere strumenti su misura, risparmiando tempo e risorse, e aumentando l'efficacia operativa.

2.3 Capitolato C4 - NearYou - Smart custom advertising platform

Proponente: Sync Lab

Obiettivo: Sviluppare una *piattaforma_G* di advertising personalizzata tramite *AI_G*, in grado di creare annunci su misura, analizzando dati di localizzazione, di profilazione e interazioni pregresse. L'intento del *progetto_G* è quello di migliorare l'esperienza utente, di elevare il valore percepito delle pubblicità e di permettere agli inserzionisti di migliorare la precisione nel targeting. Il *sistema_G* si basa sull'uso di tecnologie per *data stream processing_G* e *LLM_G*.

Tecnologie Suggestite:

- **Python** (o altri linguaggi): per l'utilizzo di *framework_G* per la simulazione dei dati quanto più possibile realistica.
- **Apache Kafka** (in alternativa: **RabbitMQ** o **HiveMQ**): piattaforme di event streaming, message brokers, o MQTT brokers con l'obiettivo di utilizzare broker per disaccoppiare lo stream di informazioni provenienti dai simulatori.
- **Apache Airflow, Apache NiFi** (in alternativa: framework come **Apache Spark, Apache Flink**): strumenti utilizzati per il *data stream processing_G* in grado di prelevare i dati e processarli, per fornirli in pasto al *framework_G* per la *generative AI_G*.
- **LangChain** (oppure **Flow** o simili): per processare i messaggi in input e fornire una risposta tramite *LLM_G*.
- **PostGIS** (in alternativa: **ClickHouse, Timescale** o simili): database *relazionali_G* in grado di soddisfare le esigenze specifiche di *progetto_G*, ovvero:
 - Capacità di sopportare ingenti quantità di dati in *INSERT_G*
 - Capacità di gestire comodamente messaggi di tipo *timeseries_G*
 - Capacità di offrire funzioni e metodi per l'elaborazione di dati *geospaziali_G*.
- **Superset, Grafana, Tableau**: strumenti utilizzati per la *data visualization_G* delle informazioni, sia lato utente che lato cliente.

Considerazioni:

- Interessanti i diversi campi di lavoro dell'*azienda_G* proponente_G.
- Supporto fornito dalla *proponente_G* attraverso personale altamente professionale.
- Creazione di contenuti pubblicitari mirati così da evitare pubblicità indesiderate da parte dell'*utente_G*, garantendo uno stretto legame *utente*-contenuti.
- *Progetto_G* con elevata utilità pratica effettiva.
- Alto stimolo, da parte del *team_G*, nell'imparare ad utilizzare tecnologie nate per *data stream processing_G*.

2.4 Capitolato C5 - 3DataViz

Proponente: Sanmarco Informatica S.P.A.

Obiettivo: realizzare in modo navigabile e interattivo un'interfaccia web per la visualizzazione in forma tridimensionale di dati tramite *istogrammi_G* e i relativi dati di origine.

Tecnologie Suggeste:

- **Three.js:** libreria *JavaScript_G* che consente di creare grafica 3D nel *browser_G*, utilizzando *WebGL_G*. *WebGL_G* è una tecnologia che permette al *browser_G* di visualizzare grafica 3D senza bisogno di *plug-in_G* esterni.
- **D3.js:** libreria *JavaScript_G* progettata per creare visualizzazioni di dati dinamiche e interattive nel *browser_G*, utilizzando standard_G web come *SVG_G*, *HTML_G* e *CSS_G*.
- **Angular:** per lo sviluppo di applicazioni web dinamiche, modulari e riutilizzabili.
- **React:** per la realizzazione di interfacce *utente_G* interattive, usato per costruire componenti dell'interfaccia che si aggiornano efficientemente al cambiare dei dati.

Considerazioni:

- *Progetto_G* stimolante in quanto non si limita alla gestione dei dati ma include anche la loro rappresentazione grafica in maniera interattiva. Questo permette di lavorare sia sul trattamento dei dati che sul loro aspetto visivo, rendendo l'esperienza di sviluppo completa e dinamica.
- La visualizzazione dei dati è resa molto più intuitiva e *accessibile_G*, grazie ad una rappresentazione grafica consapevole, risulta più semplice comprendere informazioni che altrimenti sarebbero difficili da interpretare.
- Alto grado di libertà nella gestione dei dati, che consente di esplorare diverse modalità di rappresentazione e manipolazione delle informazioni.

2.5 Capitolato C6 - Sistema di gestione di un magazzino distribuito

Proponente: M31 S.r.l.

Obiettivo: Sviluppare un *sistema_G* distribuito, scalabile_G, basato su architettura a *microservizi_G*, che favorisca l'interoperabilità_G tra i diversi magazzini e la centralizzazione delle informazioni in modo efficiente e sicuro. Il *sistema_G* in questione dovrà essere in grado di *ottimizzare_G* i livelli delle scorte, gestendo le scorte minime in ciascun magazzino senza compromettere l'operatività, gestire la condivisione dei dati in tempo reale, implementare un riassortimento predittivo e risolvere i conflitti di aggiornamento simultaneo, che potrebbero portare ad errori di valutazione da parte dei magazzini coinvolti.

Tecnologie Suggerite:

- **Node.js** e **Nest.js**: per lo sviluppo di *microservizi_G*, grazie alla loro capacità di creare applicazioni *server-side_G* performanti e modulari.
- **Go**: utilizzato per eventuali componenti ad alte prestazioni (es. servizi di sincronizzazione).
- **NATS** o **Apache Kafka**: utilizzate per la gestione di messaggi distribuiti e asincroni.
- **Kubernetes**: servizio *cloud_G* affidabile e scalabile_G offerto da Google Cloud Platform_G.
- **MongoDB**: per la memorizzazione di dati non strutturati.
- **PostgreSQL**: per la persistenza di dati strutturati.
- **Redis**: utilizzato come *sistema_G* di caching per migliorare le prestazioni e ridurre la latenza.
- **Angular**: utilizzato nell'interfaccia utente per fornire agli utenti un'esperienza il più possibile simile a quella di un'applicazione *desktop_G*.

Considerazioni:

- Il *progetto_G* si concentra sulla gestione efficiente delle scorte attraverso la sincronizzazione in tempo reale, riducendo i rischi di esaurimento e sovraccarico. Questo contribuisce a minimizzare i costi operativi e a migliorare il servizio al cliente.
- L'implementazione di algoritmi di machine learning permette di anticipare la domanda e pianificare il riassortimento in modo intelligente, basandosi su dati storici e tendenze di consumo. Questo riduce la necessità di interventi manuali e migliora l'accuratezza delle previsioni.
- Il *progetto_G* offre la libertà di scegliere tecnologie alternative, purché motivate. Questo consente di adottare strumenti innovativi senza rigidità, adattandosi a specifiche esigenze tecniche e ambientali, e promuovendo l'uso di tecnologie all'avanguardia già presenti in azienda.

2.6 Capitolato C7 - LLM: Assistente virtuale

Proponente: Ergon Informatica Srl

Obiettivo: Sviluppare un *Assistente Virtuale*_G che assista i *clienti*_G nella ricerca delle informazioni sui prodotti disponibili e risponda alle domande più frequenti. Tale *progetto*_G prende in considerazione tutte quelle aziende il cui core business è dato dalla vendita di prodotti dove i *clienti*_G effettuano ordini di acquisto multiprodotto all'azienda. In quest'ottica, l'azienda si propone di favorire l'utilizzo di un *assistente virtuale*_G basato su LLM_G, a sfavore degli "*specialist*_G", ovvero persone preparate accuratamente sui dettagli di alcuni prodotti appartenenti alla loro area di competenza, per l'assistenza ai loro clienti, cosicché da "facilitare" il rapporto tra azienda e cliente.

Tecnologie Suggerite:

- **Database Relazionale:** per la storicizzazione dei dati (per il *progetto*_G in questione può essere utilizzato un qualsiasi database relazionale_G).
- **LLM:** BLOOM, Falcon IA, Pythia, Italia by iGenius, Minerva.
- **API REST:** per la comunicazione tra il modello LLM_G e l'app_G di interazione con l'utente_G.
- **Comunicazione da/per il database:** in funzione del componente scelto per lo sviluppo del modello LLM_G, si possono optare per diverse strade per comunicare con il database_G:
 - attraverso connettori standard_G di una fonte dati ODBC.
 - rendere indipendente il modello LLM_G dal database_G implementando un middleware_G che gestisca la comunicazione tra i componenti.
- **Interfaccia utente:** per lo sviluppo di un'app_G mobile che implementa la chat_G di dialogo tra l'AI_G e l'utente_G su piattaforma_G .NET, MAUI o Android.

Considerazioni:

- Grazie al virtual assistant, i clienti possono ottenere risposte in tempo reale alle loro domande senza dover attendere un operatore umano.
- L'LLM_G consente di personalizzare l'esperienza di ogni cliente, anche grazie all'analisi comportamentale che permette di consigliare prodotti e fornire risposte personalizzate, supportando strategie di marketing mirate.
- Automatizzando le risposte e riducendo il tempo di interazione richiesto dal personale, l'azienda può *ottimizzare*_G i costi di servizio, destinando risorse verso attività a più alto valore aggiunto.

2.7 Capitolato C8 - Requirement Tracker - Plug-in VS Code

Proponente: *Bluewind s.r.l.*

Obiettivo: Sviluppare un *plug-in*_G per Visual Studio Code_G denominato *Requirement Tracker - VS Code Plug-in*_G con l'obiettivo di automatizzare il tracciamento dei *requisiti*_G di *progetto*_G nel *codice sorgente*_G, fornire suggerimenti per migliorare la qualità dei *requisiti*_G rispettando la clausola di sottostare ad una metodologia di progettazione in grado di rendere agevoli future estensioni capaci di introdurre nuove funzionalità. Lo scopo finale del *progetto*_G sarà, quindi, quello di facilitare utenti, *programmatori*_G o qualsiasi componente all'interno di un *team*_G di sviluppo, nella valutazione, nella stesura e nel tracciamento dei *requisiti*_G stipulati, andando così ad aumentare le *performance*_G durante tutte le fasi di sviluppo del *codice*_G.

Tecnologie Suggerite:

- **Visual Studio Code Extension API:** per la costruzione di un'*architettura*_G modulare, che consenta l'aggiunta di nuove funzionalità in maniera semplice.
- **API REST:** per la connessione a modelli di *AI*_G per l'*analisi del codice*_G e dei *requisiti*_G.
- **Python o Node.js:** per l'integrazione con le API *AI*_G, con un *design*_G flessibile che consenta di aggiungere nuovi *linguaggi*_G o componenti senza modifiche significative.
- **Modelli API pre-addestrati** (es. GPT o simili): per analisi semantiche, con la possibilità di integrare facilmente nuovi *modelli*_G o *algoritmi*_G in futuro.
- **Ollama** o alternative: per il *deployment*_G locale di *LLM*_G (opzionale).

Considerazioni:

- Aumento delle *performance*_G nella stesura del codice grazie al meccanismo automatizzato offerto dal *plug-in*_G nel tracciare i *requisiti*_G, con l'intento di migliorare e verificarne la qualità.
- Completo appoggio da parte dell'azienda durante la fase di sviluppo del *progetto*_G, sia con incontri in presenza e sia con incontri da remoto.

2.8 Capitolato C9 - BuddyBot

Proponente: *Azzurro Digitale*

Obiettivo: Sviluppare un assistente virtuale che sia in grado di ottenere in modo facile e veloce informazioni dalle fonti specificate e di fornirle in base alle domande poste tramite chat in *linguaggio naturale_G*. Con tale *progetto_G*, l'azienda mira a fornire a ciascun membro del proprio *team_G* la possibilità di migliorare la propria produttività, affidando la ricerca e l'ottenimento delle informazioni desiderate completamente all'assistente virtuale.

Tecnologie Suggeste:

- **OpenAI:** sarà il motore per le funzionalità di *NLP_G*, cioè di comprensione del testo e generazione delle risposte.
- **Langchain:** *progetto open source_G* che permette di integrare modelli di *AI_G* senza conoscerne i dettagli interni. Usa i modelli come delle *blackbox_G*, quindi è perfetto per chi vuole integrare funzionalità di *AI_G*.
- **Angular:** per la creazione di *interfacce utente_G* modulari e riutilizzabili.
- **Node/NestJS:** facilita la creazione di applicazioni scalabili e manutenibili, seguendo i principi del design orientato ai *microservizi_G* e alle API RESTful.
- **Spring Boot:** offre un ambiente preconfigurato e un set di strumenti per creare applicazioni standalone_G e pronte per la produzione, con supporto integrato per database, sicurezza, gestione delle dipendenze e *microservizi_G*.

Considerazioni:

- L'utilizzo delle tecnologie suggerite dall'azienda non è obbligatorio, con una conseguente possibilità di impiego di differenti tecnologie che si possono adattare al meglio alle caratteristiche assunte dal *progetto_G* durante il suo avanzamento.
- Evidente aumento delle *performance_G* all'interno dei *team_G* aziendali e quindi dell'azienda stessa, la quale può allocare le risorse risparmiate per il completamento di compiti di maggiore priorità.
- L'utilizzo dell'AI per il reperimento delle informazioni può aumentare e migliorare lo schema mentale dei vari utenti aziendali per quanto riguarda la struttura e la gestione delle informazioni dell'azienda stessa.

3 Conclusioni

Dopo un'attenta valutazione di tutti i capitolati proposti, ogni membro del gruppo ha espresso le proprie preferenze ed è emerso che il **capitolato C2** è quello che ha suscitato maggiore stimolo ed interesse. La scelta di questo capitolato vuole ampliare le conoscenze tecniche e pratiche di tutto il gruppo, grazie alle interessanti tecnologie proposte dall'azienda e grazie alla gestione degli avanzamenti che il *proponente_G* offre durante il percorso di sviluppo del *progetto_G*. L'utilizzo dell'*AI_G*, in particolare degli *LLM_G*, per la realizzazione di un assistente *virtuale_G*, strettamente legato alla fase di progettazione di un impianto e di installazione di dispositivi elettrici all'interno di esso, è in grado di semplificare largamente il lavoro quotidiano di ciascun installatore, apportando, a questi ultimi, benefici anche al di fuori dell'ambito lavorativo. Inoltre, la visione di un assistente *virtuale_G* simile, può essere espansa in svariati settori lavorativi. Profondamente attratti dall'argomento proposto, intendiamo realizzare un prodotto professionale, in grado di fondere alla perfezione robustezza tecnologica e semplicità di utilizzo, dandogli il corretto spazio e possibilità di crescita futura.