

Norme di Progetto

2025-05-20

Responsabile	Manuel Cinnirella
	Giovanni Battista Matteazzi
	Alessandro Di Pasquale
Redattori	Giovanni Battista Matteazzi
	Alessandro Di Pasquale
	Nicolò Bovo
	Romeo Calero
Validatori	Massimo Chioru
	Manuel Cinnirella
	Elia Leonetti
	Nicolò Bovo

Registro delle modifiche

Vers.	Data	Descrizione	Autore	Verificatore
0.5.0	2025-05-20	Estensione del documento	Giovanni Battista Matteazzi, Alessandro Di Pasquale, Nicolò Bovo, Romeo Calearo	Massimo Chioru, Manuel Cinnirella, Elia Leonetti
0.1.0	2025-04-10	Bozza del documento	Giovanni Battista Matteazzi, Alessandro Di Pasquale, Nicolò Bovo, Romeo Calearo	Massimo Chioru, Manuel Cinnirella, Elia Leonetti

Indice

I. Introduzione	5
I - 1. Scopo del documento	5
I - 2. Scopo del progetto	5
I - 3. Riferimenti	5
I - 3.1. Riferimenti normativi	5
I - 3.2. Riferimenti informativi	6
II. Processi primari	6
II - 1. Fornitura	6
II - 1.1. Scopo	6
II - 1.2. Descrizione	6
II - 1.3. Aspettative	6
II - 1.4. Comunicazioni con il proponente	7
II - 1.5. Documentazione fornita	7
II - 1.5.1. Valutazione dei Capitolati	7
II - 1.5.2. Analisi dei Requisiti	8
II - 1.5.3. Piano di Progetto	8
II - 1.5.4. Piano di Qualifica	9
II - 2. Sviluppo	10
II - 2.1. Scopo	10
II - 2.2. Descrizione	10
II - 2.2.1. Implementazione del processo	11
II - 2.2.2. Analisi dei requisiti di sistema	11
II - 2.2.3. Progettazione architettuale del sistema	11
II - 2.2.4. Analisi dei requisiti software	12
II - 2.3. Progettazione architettuale del software	12
II - 2.4. Progettazione dettagliata del software	13
II - 2.5. Codifica e testing del software	14
II - 2.6. Integrazione del software	14
II - 2.7. Test di qualifica del software	15
II - 2.8. Integrazione di sistema	15
II - 2.9. Test di qualifica del sistema	16
II - 2.10. Installazione del software	16
II - 2.11. Supporto all'accettazione del software	17
III. Processi di supporto	17
III - 1. Documentazione	17
III - 1.1. Scopo	17
III - 1.2. Descrizione	17
III - 1.3. Aspettative	18
III - 1.4. Ciclo di vita	18
III - 1.5. Sistema di composizione tipografica	19
III - 1.6. Struttura dei documenti	19
III - 1.6.1. Intestazione	19

III - 1.6.2. Registro delle modifiche	19
III - 1.6.3. Indice	19
III - 1.6.4. Corpo del documento	20
III - 1.6.5. Corpo del verbale	20
III - 1.6.6. Documenti del progetto	20

I. Introduzione

I - 1. Scopo del documento

Il presente documento si propone di definire in modo organico e condiviso il “Way of Working” adottato dal gruppo AlphaCode_G per l’esecuzione del progetto didattico, allineandosi ai processi di Project Planning, Project Monitoring & Control e Organizational Project-Enabling così come descritti nella norma ISO/IEC 12207:1995_G. In particolare, esso raccoglie e formalizza le best practice, i ruoli, le responsabilità e le attività che ciascun membro del team dovrà seguire, al fine di garantire un approccio professionale, coerente e riproducibile lungo tutto il ciclo di vita del progetto.

Per la sua redazione si è seguito un metodo incrementale: inizialmente è stato sviluppato un impianto base che descrive i processi fondamentali e i deliverable principali; successivamente, con l’avanzare del lavoro e il confronto continuo all’interno del team, sono state aggiunte e raffinate procedure operative, modelli di documentazione e linee guida per la gestione delle modifiche e dei rischi. Ogni nuova versione di questo documento terrà conto delle decisioni prese dal gruppo durante le revisioni periodiche, garantendo così flessibilità e capacità di adattamento senza compromettere la coerenza con gli standard internazionali.

I membri del gruppo AlphaCode si impegnano a consultare regolarmente questo documento e a rispettarne le indicazioni, assicurando così uniformità nei processi di pianificazione, monitoraggio e controllo del progetto. Tale impegno non solo favorisce la trasparenza nei confronti del proponente, ma consente anche di mantenere alta la qualità del lavoro e di tracciare in modo puntuale l’avanzamento delle attività, come prescritto dal processo di Configuration Management della ISO/IEC 12207:1995_G.

I - 2. Scopo del progetto

Lo scopo di NearYou è progettare, realizzare e validare una piattaforma di advertising personalizzata in tempo reale, capace di unire flussi GPS, profilazione utente e generative AI per produrre annunci altamente contestuali. Oltre a implementare simulatori di posizionamento, pipeline Kafka_G → ClickHouse_G → PostGIS_G e modelli LLM_G, il team AlphaCode condurrà misurazioni mirate per valutare throughput e latenza delle componenti di streaming (Kafka)_G e orchestrazione (Airflow), l’efficacia e i tempi di risposta delle query geospaziali in PostGIS_G, e la velocità di generazione e consegna degli annunci via FastAPI_G. Analizzeremo inoltre l’impatto delle operazioni AI su CPU e memoria e testeremo la resilienza dei microservizi (producer, consumer, webapp) in caso di picchi di traffico o failure di rete. Per confermare scalabilità, robustezza delle logiche di prossimità e manutenibilità del codice, AlphaCode eseguirà stress test ad alto carico su tutte le pipeline, verificando la correttezza funzionale e le performance sotto condizioni estreme.

I - 3. Riferimenti

I - 3.1. Riferimenti normativi

- Capitolato d’appalto C4: NearYou, sistema di advertising
- Standard ISO/IEC 12207:1995

I - 3.2. Riferimenti informativi

- I processi di ciclo di vita del software
- Glossario

II. Processi primari

II - 1. Fornitura

II - 1.1. Scopo

Il processo di fornitura, così come previsto dal capitolato NearYou e in linea con i principi della ISO/IEC 12207:1995₆, costituisce il momento in cui il team AlphaCode definisce formalmente i termini della collaborazione con il proponente (*Sync Lab*₆). Attraverso un contratto condiviso, vengono stabiliti i requisiti funzionali e non funzionali - dal simulatore GPS fino all'integrazione con i modelli di generative AI - i vincoli tecnologici e di sicurezza (ad esempio l'adozione di Mutual TLS₆ e di database PostGIS/ClickHouse₆) e le milestone temporali per la consegna di ciascun deliverable. Questo accordo non è un semplice atto burocratico, ma la bussola che guida ogni attività del gruppo, permettendo di tracciare costantemente lo stato di avanzamento, individuare eventuali scostamenti e riallinearsi alle attese del proponente. Solo dopo aver formalizzato questi punti, AlphaCode procede alla redazione del Piano di Progetto, che dettaglia in modo sistematico tempi, risorse, ruoli e modalità di controllo necessari per una realizzazione ordinata e trasparente.

II - 1.2. Descrizione

Questo processo è diviso nelle seguenti fasi:

- Inizializzazione;
- Preparazione delle risposte alle richieste;
- Contrattazione;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Consegna e completamento.

II - 1.3. Aspettative

Il gruppo *AlphaCode* si impegna a mantenere una collaborazione costante e professionale con l'azienda proponente *Sync Lab*₆, facendo riferimento ai contatti indicati nel capitolato, in particolare i referenti tecnici Davide Zorzi e Andrea Dorigo, con Federico Pallaro sempre in copia nelle comunicazioni.

Attraverso un dialogo continuo, il gruppo intende assicurarsi che il lavoro svolto sia allineato alle richieste del capitolato, ricevendo feedback regolari e chiarendo tempestivamente eventuali dubbi. L'obiettivo è garantire che vincoli, requisiti e scadenze siano rispettati, offrendo trasparenza sullo stato di avanzamento del progetto e favorendo una collaborazione efficace durante tutte le fasi dello sviluppo.

II - 1.4. Comunicazioni con il proponente

Per garantire un flusso comunicativo efficace e trasparente, il gruppo AlphaCode ha concordato con l'azienda proponente *Sync Lab* una cadenza settimanale per gli incontri esterni tramite piattaforma Google Meet, durante i quali vengono affrontate le principali criticità e discussi i progressi del progetto. Tali riunioni avvengono generalmente in modalità telematica e vedono la partecipazione attiva dei referenti tecnici indicati nel capitolato.

Gli incontri hanno lo scopo di:

- chiarire dubbi sui requisiti o sui vincoli progettuali;
- discutere le tecnologie adottate o future integrazioni;
- raccogliere feedback sulle componenti sviluppate.

Ogni colloquio viene documentato attraverso la stesura di un Verbale Esterno, che include data, partecipanti e contenuti rilevanti emersi dalla discussione. I verbali vengono salvati in una cartella dedicata all'interno della repository ufficiale del progetto, in modo da garantirne la tracciabilità e l'accessibilità per tutte le parti coinvolte.

II - 1.5. Documentazione fornita

Il gruppo AlphaCode si impegna a produrre e consegnare una documentazione completa e strutturata, rivolta sia all'azienda proponente *Sync Lab*, sia ai committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin.

II - 1.5.1. Valutazione dei Capitolati

Il documento di Valutazione dei Capitolati ha lo scopo di analizzare in maniera approfondita i progetti proposti dalle aziende nella giornata del 2025/03/04. Per ogni capitolato presentato, il gruppo AlphaCode ha identificato le esigenze del proponente, le potenziali soluzioni tecniche e le eventuali criticità che avrebbero potuto ostacolare lo sviluppo.

La valutazione si articola in sezioni ben definite che comprendono:

- **Descrizione:** include il nome del progetto, l'azienda proponente e le informazioni principali relative al prodotto, così come presentato;
- **Dominio applicativo:** definisce il contesto in cui il progetto si colloca, con particolare attenzione agli utenti finali e agli scenari d'uso;
- **Dominio tecnologico:** elenca gli strumenti, linguaggi, framework e infrastrutture richiesti;
- **Aspetti positivi:** evidenzia le opportunità di apprendimento e di innovazione legate al progetto;
- **Fattori critici:** analizza eventuali rischi o complessità, sia tecniche che organizzative;
- **Conclusione:** motiva la scelta o la non scelta del capitolato sulla base di criteri condivisi dal gruppo.

Nel caso specifico, il gruppo AlphaCode ha scelto di aderire al Capitolato C4 - NearYou, proposto dall'azienda *Sync Lab*, per l'interesse dimostrato verso le tecnologie di streaming dati, la componente geospaziale e l'integrazione con tecniche di AI generativa.

II - 1.5.2. Analisi dei Requisiti

Il documento di Analisi dei Requisiti ha il compito di descrivere nel dettaglio le funzionalità che il sistema NearYou dovrà offrire, oltre ai vincoli da rispettare affinché il prodotto risulti conforme alle aspettative espresse nel capitolato e alle necessità del proponente *Sync Lab*.

All'interno del documento vengono fornite le seguenti informazioni:

- **Descrizione del prodotto:** viene delineato l'obiettivo generale del sistema, ovvero la creazione di una piattaforma per l'advertising personalizzato basata su flussi GPS, profilazione utenti e AI generativa, con evidenza sulle sue principali funzionalità in ambito di raccolta, trasformazione e analisi dei dati.
- **Casi d'uso:** il sistema viene descritto attraverso una serie di scenari d'uso concreti che illustrano come gli utenti interagiscono con la piattaforma. Ogni caso d'uso evidenzia lo scenario, gli attori coinvolti (ad esempio: utenti, amministratori, processi automatici) e le azioni che questi possono compiere.
- **Requisiti funzionali e non funzionali:** sono elencati tutti i requisiti identificati, siano essi forniti esplicitamente dal proponente o dedotti dal gruppo sulla base delle funzionalità richieste. Tali requisiti vengono classificati come:
 - **Obbligatori:** indispensabili per il funzionamento minimo del sistema;
 - **Desiderabili:** utili al miglioramento della qualità e dell'esperienza utente;
 - **Opzionali:** funzionalità aggiuntive implementabili in funzione del tempo disponibile.

L'analisi dei requisiti rappresenta la base per l'intero processo di sviluppo e verrà aggiornata in modo incrementale in caso emergano nuove necessità o cambiamenti durante le fasi progettuali.

II - 1.5.3. Piano di Progetto

Il Piano di Progetto è un documento strategico, redatto e mantenuto dal Responsabile di Progetto con il supporto dell'Amministratore, che ha lo scopo di guidare lo sviluppo del sistema NearYou attraverso una pianificazione strutturata, realistica e monitorabile. Viene costantemente aggiornato per riflettere l'evoluzione delle attività, dei vincoli e delle decisioni progettuali.

Il documento comprende:

- **Analisi dei rischi:** il gruppo identifica e classifica le possibili problematiche che potrebbero compromettere il corretto svolgimento delle attività, distinguendo tra:
 - *Rischi organizzativi*, legati a dinamiche di gruppo, pianificazione e comunicazione;
 - *Rischi tecnologici*, relativi all'integrazione di strumenti complessi come Kafka_®, ClickHouse_®, PostGIS_® e modelli AI generativi.

Per ogni rischio vengono definite misure preventive e correttive per contenerne l'impatto.

- **Modello di sviluppo:** viene illustrata la strategia metodologica adottata dal team AlphaCode, che segue un approccio incrementale ed iterativo, ispirato a pratiche agili, per garantire una costante validazione degli obiettivi raggiunti e la massima adattabilità ai feedback.

- **Pianificazione:** il lavoro viene suddiviso in periodi chiave (milestone), ognuno dei quali è corredato da una panoramica delle attività previste, dei ruoli assegnati, delle responsabilità individuali e delle stime di impegno in termini di ore/uomo.
- **Preventivo:** include una stima temporale dettagliata del carico di lavoro previsto per ogni membro del gruppo, tenendo conto delle complessità tecniche e delle dipendenze tra le attività.
- **Consuntivo:** al termine di ciascun periodo pianificato, viene effettuata un'analisi delle attività realmente svolte rispetto a quanto previsto. Questo confronto consente di individuare eventuali scostamenti, valutarne le cause ed effettuare un aggiornamento del piano nelle fasi successive.

Il Piano di Progetto è uno strumento fondamentale per mantenere il controllo sul progresso del lavoro e per garantire il raggiungimento degli obiettivi nei tempi e modi concordati con il proponente.

II - 1.5.4. Piano di Qualifica

Il Piano di Qualifica è il documento che delinea le strategie, le metriche e le attività adottate dal gruppo AlphaCode per garantire la qualità del sistema NearYou, sia a livello di processo che di prodotto. Rappresenta un riferimento centrale per tutti i membri del team durante lo sviluppo, al fine di assicurare che il risultato finale sia conforme alle aspettative del proponente e agli standard richiesti.

Il documento è strutturato in quattro sezioni principali:

- **Qualità di processo:** vengono definite le metriche da rispettare per monitorare l'efficacia e l'efficienza del ciclo di sviluppo. Tra questi parametri figurano la tracciabilità dei requisiti, la copertura della verifica e il rispetto delle scadenze prefissate.
- **Qualità di prodotto:** si stabiliscono i criteri per valutare le caratteristiche del software, come manutenibilità, affidabilità, usabilità e performance. Questi aspetti saranno misurati tramite indicatori oggettivi e soggettivi, per garantire un risultato stabile e robusto.
- **Test:** la sezione include la descrizione delle attività di test previste (test funzionali) e i criteri di accettazione per ogni funzionalità. I test saranno fondamentali per verificare che ogni componente del sistema rispetti i requisiti.
- **Valutazioni per il miglioramento:** al termine di ogni ciclo di verifica, verranno registrati i risultati ottenuti, segnalate le criticità emerse e proposte azioni correttive. Questo approccio incrementale permetterà di migliorare costantemente la qualità complessiva del progetto.

Il Piano di Qualifica assume quindi un ruolo chiave nell'assicurare che il progetto NearYou sia sviluppato secondo metodologie rigorose e orientate alla qualità, permettendo di mantenere sotto controllo tutti gli aspetti critici legati alla verifica e alla validazione del sistema.

II - 2. Sviluppo

II - 2.1. Scopo

In conformità allo standard ISO/IEC 12207:1995_G, il processo di sviluppo ha lo scopo di definire e gestire le attività tecniche necessarie per trasformare i requisiti del proponente in un prodotto software conforme e funzionante. Per il progetto NearYou, il processo include l'analisi dei requisiti, la progettazione architetturale e di dettaglio, l'implementazione delle componenti, l'integrazione dei vari moduli, il collaudo del sistema, la sua installazione in ambiente operativo e l'accettazione finale da parte del committente. Tutte le attività sono guidate da pratiche di sviluppo incrementali e iterative, con l'obiettivo di assicurare qualità, tracciabilità dei requisiti e aderenza ai vincoli tecnologici stabiliti nel capitolato.

II - 2.2. Descrizione

Le attività che compongono il processo di sviluppo del progetto NearYou, strutturate secondo lo standard ISO/IEC 12207:1995_G, sono organizzate in fasi sequenziali e iterabili, ciascuna mirata alla costruzione incrementale del sistema:

- **Definizione del processo di sviluppo:** formalizzazione del flusso operativo e delle pratiche adottate dal gruppo, incluso l'uso di strumenti e metodologie agili.
- **Analisi dei requisiti di sistema:** studio dei requisiti funzionali e non funzionali forniti dal capitolato, con valutazione di vincoli architetturali e tecnologici.
- **Progettazione dell'architettura di sistema:** definizione dei macro-componenti e della loro interazione (es. Kafka, ClickHouse, PostGIS, WebApp).
- **Analisi dei requisiti software:** individuazione delle funzionalità specifiche di ogni modulo software, come il consumer Kafka, i DAG_G di Airflow, o la web app FastAPI_G.
- **Progettazione architetturale del software:** suddivisione in microservizi, definizione delle interfacce, scelte tecnologiche (containerizzazione, sicurezza TLS, ecc.).
- **Progettazione dettagliata del software:** specifica dei componenti interni, logiche di business, gestione degli errori, logging, sicurezza e configurabilità.
- **Codifica e testing unitario:** sviluppo del codice con pratiche di qualità e test delle singole unità.
- **Integrazione progressiva:** combinazione delle componenti in ambienti Docker tramite Docker Compose_G.
- **Test di qualifica del software:** verifica che ogni modulo soddisfi i requisiti previsti (es. throughput del producer, correttezza del consumer, efficienza dei DAG).
- **Integrazione di sistema:** test del sistema completo, inclusi pipeline Kafka→ClickHouse→PostGIS e moduli AI.
- **Test di qualifica di sistema:** valutazione delle prestazioni, affidabilità, sicurezza e scalabilità del sistema.
- **Installazione del sistema in ambiente target:** configurazione degli ambienti Gitpod e/o locali tramite Docker.
- **Supporto all'accettazione:** fornitura di documentazione tecnica e dimostrazione del prodotto, con eventuali sessioni di collaudo da parte del proponente.

II - 2.2.1. Implementazione del processo

L'implementazione del processo di sviluppo per il progetto NearYou ha avuto inizio con la scelta di un modello di ciclo di vita incrementale ed iterativo, compatibile con i vincoli temporali e con l'evoluzione graduale dei requisiti. Il team AlphaCode ha definito in modo chiaro i ruoli e le responsabilità di ogni membro, assicurando la copertura di aspetti fondamentali come la redazione della documentazione, la gestione delle configurazioni (tramite Git), la risoluzione dei problemi, e il supporto alla qualità.

Per mantenere coerenza e tracciabilità nel processo, sono stati adottati standard di codifica, strumenti di automazione (come Docker, Gitpod e Airflow), metodi di verifica continua e linguaggi adeguati allo scopo.

Il gruppo ha anche pianificato l'utilizzo di risorse condivise, ambienti di sviluppo replicabili e tool di gestione collaborativa (come GitHub e Discord), per garantire trasparenza, coordinamento e continuità.

II - 2.2.2. Analisi dei requisiti di sistema

Nel contesto dell'Analisi dei Requisiti di Sistema, il processo ha coinvolto il team AlphaCode nell'esecuzione e nel supporto delle attività previste dal capitolato NearYou. La prima fase ha richiesto l'analisi dell'uso specifico del sistema in sviluppo, volto alla realizzazione di una piattaforma per l'advertising personalizzato basata sull'integrazione di dati GPS, profilazione utente e generative AI. Tale analisi ha permesso di definire accuratamente i requisiti di sistema.

Questi requisiti includono funzionalità come il consumo e la gestione in tempo reale di flussi Kafka, la persistenza efficiente in ClickHouse, l'esecuzione di query spaziali in PostGIS, la generazione dinamica di annunci tramite modelli LLM, nonché la comunicazione sicura tramite certificati mutual TLS. Sono stati inoltre definiti i requisiti relativi alle performance del sistema, all'affidabilità, alla manutenibilità, alle interfacce, all'operatività e alla sicurezza, insieme ai vincoli di progettazione, ambienti d'esecuzione (containerizzati) e requisiti di qualifica.

La fase successiva ha previsto la valutazione di tali requisiti in base alla loro rintracciabilità rispetto ai bisogni espressi dal proponente, alla coerenza interna, alla testabilità e alla fattibilità architettuale. È stata inoltre verificata la sostenibilità delle operazioni e della manutenzione a lungo termine. I risultati delle analisi sono stati accuratamente documentati nel relativo documento di Analisi dei Requisiti, che costituisce base fondante per le attività di progettazione e sviluppo.

II - 2.2.3. Progettazione architettuale del sistema

Nel contesto della progettazione architettuale del sistema NearYou, la prima fase ha previsto la definizione di un'architettura ad alto livello basata su una composizione di microservizi containerizzati, orchestrati tramite Docker Compose. Sono stati identificati i principali componenti del sistema, tra cui: generatori di dati GPS, broker Kafka per lo streaming sicuro (mutual TLS), database analitico ClickHouse, sistema geospaziale PostGIS, orchestratore Airflow, API FastAPI per la generazione di annunci personalizzati tramite modelli LLM e interfaccia Grafana per la visualizzazione dei dati.

Tutti i requisiti del sistema sono stati allocati con precisione ai rispettivi elementi architetturali, distinguendo tra componenti software, operazioni automatizzate (es. DAG_G di Airflow) e interazioni manuali (es. generazione certificati, monitoraggio).

Successivamente è stata condotta una valutazione dell'architettura proposta, con particolare attenzione alla tracciabilità dei requisiti, alla coerenza tra i moduli, all'adeguatezza degli standard tecnologici adottati, alla fattibilità implementativa dei singoli elementi software e alla manutenibilità del sistema.

II - 2.2.4. Analisi dei requisiti software

Nel contesto del progetto NearYou, l'analisi dei requisiti software ha avuto lo scopo di identificare e documentare in maniera strutturata tutte le caratteristiche funzionali e non funzionali necessarie alla realizzazione degli elementi software che compongono la piattaforma. Per ciascun modulo identificato (es. microservizi Kafka, consumer, producer, FastAPI, DAG Airflow), il team ha definito:

- *Specifiche funzionali e prestazionali*, tenendo conto delle capacità di throughput, latenza di risposta, volume dati gestito e resilienza;
- *Interfacce esterne*, in particolare tra Kafka ↔ ClickHouse, ClickHouse ↔ FastAPI e tra API ↔ frontend;
- *Requisiti di sicurezza*, relativi alla comunicazione cifrata tramite TLS, gestione dei certificati mutual TLS e protezione dei dati trasmessi e memorizzati;
- *Aspetti legati all'usabilità e all'interazione utente-sistema nella webapp*, anche in scenari con accesso simultaneo e consultazione dei dati tramite dashboard;
- *Definizione dei dati gestiti*, tra cui eventi GPS, utenti, POI, annunci e le strutture dei relativi database (ClickHouse e PostGIS);
- *Requisiti di deploy*, operatività e manutenzione dei container Docker e dei servizi;
- *Documentazione prevista per l'utente e per lo sviluppatore* (es. API reference, struttura delle query in Grafana);

A completamento dell'analisi, ogni requisito è stato valutato in base a:

- Rintracciabilità rispetto ai requisiti di sistema;
- Coerenza interna ed esterna con le specifiche architetturali;
- Fattibilità di sviluppo e mantenimento.

II - 2.3. Progettazione architetturale del software

Per ciascun elemento software identificato nel progetto NearYou, l'attività di progettazione architetturale comprende i seguenti compiti:

- **Conversione dei requisiti in architettura:** Gli sviluppatori traducono i requisiti software in una struttura architetturale ad alto livello, identificando chiaramente i componenti software necessari (ad esempio: modulo simulazione GPS, stream processor Kafka/Flink, servizio LLM con LangChain, dashboard analitica Grafana, web-app frontend). Ogni requisito viene assegnato a un componente specifico, facilitando la successiva progettazione dettagliata. L'intera architettura viene documentata tramite diagrammi C4 (System, Container, Component).

- **Progettazione di alto livello delle interfacce:** Viene definito e documentato il progetto preliminare delle interfacce tra i componenti software (REST API, WebSocket, formati di messaggi JSON/Avro_g) e le interfacce esterne al sistema (ad esempio database geospaziali).
- **Progettazione di alto livello del database:** Viene progettata la struttura preliminare del database geospaziale (PostGIS) e analitico (ClickHouse), definendo schemi tabellari principali, indici geospaziali e strategie di partizionamento dei dati temporali per ottimizzare query e performance.
- **Valutazione dell'architettura e interfacce:** Viene effettuata una valutazione strutturata dell'architettura proposta e dei progetti preliminari delle interfacce e dei database, verificando in particolare la rintracciabilità dei requisiti, la coerenza interna ed esterna, l'uso adeguato delle tecnologie suggerite (Kafka, Flink, LangChain, PostGIS), la fattibilità della progettazione dettagliata e la semplicità di manutenzione.
- **Revisione formale:** Si conducono una o più revisioni formali (peer review) con tutto il team AlphaCode e l'azienda proponente, validando l'architettura definita e apportando eventuali miglioramenti derivanti dai feedback raccolti.

II - 2.4. Progettazione dettagliata del software

Per ciascun componente software identificato nel progetto NearYou, l'attività prevede:

- **Progettazione dettagliata dei componenti software:** Definizione puntuale di unità software (moduli Python, funzioni, classi, query SQL) per simulazione dati, stream processing (Kafka/Flink_g), generazione annunci (LangChain_g), dashboard analitica e web-app, in modo che ciascuna unità sia immediatamente codificabile e testabile.
- **Progettazione dettagliata delle interfacce:** Documentazione precisa delle API REST, WebSocket e schemi JSON/Avro, incluse specifiche tecniche complete che permettono la codifica diretta senza ambiguità.
- **Progettazione dettagliata del database:** Struttura dettagliata delle tabelle PostGIS (positions, places, ads) e ClickHouse, definendo esattamente campi, tipi, indici e partizioni.
- **Aggiornamento preliminare della documentazione utente:** Revisione e ampliamento della documentazione utente con dettagli tecnici aggiornati e indicazioni operative più complete.
- **Aggiornamento requisiti e pianificazione integrazione software:** Revisione delle strategie e criteri per test di integrazione software (Docker Compose_g).
- **Valutazione della progettazione dettagliata e test:** Analisi documentata della progettazione dettagliata sulla base di rintracciabilità requisiti, coerenza architetturale, semplicità di manutenzione, fattibilità dei test unitari e integrazione.
- **Revisione formale interna:** Una o più sessioni di revisione obbligatorie del team AlphaCode per verificare la progettazione.

II - 2.5. Codifica e testing del software

Per ciascuna unità software del progetto NearYou, il programmatore svolge le seguenti attività:

- **Codifica unità software e database:** Sviluppo e documentazione delle unità software (Python 3.12, TypeScript) e relativi schemi di database (PostGIS, ClickHouse).
- **Esecuzione e documentazione test unitari:** Conduzione sistematica dei test automatizzati per verificare conformità ai requisiti funzionali e tecnici, riportando chiaramente i risultati in report strutturati.
- **Aggiornamento documentazione utente:** Revisione e integrazione continua delle guide operative e manuali utente per riflettere tutte le modifiche effettuate nel software.
- **Aggiornamento requisiti e piano integrazione software:** Affinamento dei requisiti e del piano di test di integrazione basato su Docker Compose.
- **Valutazione codice e test:**
 - Tracciabilità codice-requisiti;
 - Coerenza interna ed esterna delle unità;
 - Adeguata copertura dei test ($\geq 80\%$);
 - Conformità agli standard di codifica adottati;
 - Fattibilità tecnica di integrazione e manutenzione.

II - 2.6. Integrazione del software

Per il progetto NearYou, l'integrazione software comprende:

- **Sviluppo del Piano di Integrazione:** Documentazione delle procedure e sequenze di integrazione delle unità software (moduli Python, servizi Docker), inclusi requisiti di test, dati, responsabilità assegnate e pianificazione delle attività.
- **Esecuzione integrazione e test:** Unione progressiva delle unità software secondo il piano, effettuando test di integrazione con ambienti Docker Compose per verificare il rispetto dei requisiti funzionali e non funzionali definiti nel capitolato.
- **Aggiornamento della documentazione utente:** Revisione e aggiornamento della documentazione operativa per riflettere i cambiamenti dovuti all'integrazione.
- **Preparazione Test di Qualificazione del Software:** Creazione di casi e procedure di test specifiche per ciascun requisito del software integrato, assicurando che l'insieme integrato soddisfi tutti i criteri di qualificazione previsti (copertura $\geq 80\%$).
- **Valutazione integrazione software:** Verifica con particolare attenzione a:
 - Tracciabilità completa ai requisiti di sistema;
 - Coerenza interna/esterna dell'integrazione;
 - Adeguata copertura dei test;
 - Rispetto dei metodi e standard adottati;
 - Conformità dei risultati ai requisiti previsti;
 - Fattibilità operativa e manutentiva della piattaforma.

II - 2.7. Test di qualifica del software

Per il progetto NearYou, l'attività comprende:

- **Esecuzione test di qualificazione:** Test approfonditi delle funzionalità software integrate, secondo i requisiti definiti nel Piano di Qualifica (copertura test $\geq 80\%$). I risultati sono formalmente documentati.
- **Aggiornamento documentazione utente:** Revisione della documentazione utente per riflettere tutte le modifiche e miglioramenti introdotti dopo i test.
- **Valutazione del prodotto software:** Revisione formale basata su:
 - Copertura completa dei requisiti software;
 - Conformità ai risultati attesi;
 - Fattibilità tecnica e operativa dell'integrazione e dei test di sistema;
 - Manutenibilità e operatività futura del sistema.

Preparazione per integrazione di sistema: Aggiornamento del prodotto software e definizione di una baseline stabile del codice e del design software, pronta per la fase successiva di integrazione di sistema.

II - 2.8. Integrazione di sistema

Inizialmente, le componenti software del sistema NearYou devono essere integrate tra loro e con gli elementi di simulazione hardware (ad esempio, generatori di dati GPS e sensori simulati), operazioni manuali (come la configurazione dei dataset utente) e altri sistemi esterni (ad esempio, modelli LLM e piattaforme di visualizzazione). Gli aggregati, come il flusso dati (dai simulatori al message broker, allo stream processing, al database e alla generative AI), devono essere testati incrementalmente durante lo sviluppo, verificando la conformità ai requisiti funzionali e non funzionali definiti nel capitolato. L'integrazione e i risultati dei test devono essere documentati in modo dettagliato, inclusi schemi logici del comportamento dei simulatori e report di copertura dei test ($\geq 80\%$).

Successivamente, per ciascun requisito di qualificazione del sistema (MVP o prodotto completo), è necessario sviluppare e documentare:

- **Casi di test:** input (es. percorsi GPS predefiniti, profili utente), output attesi (es. annunci contestuali generati da LLM), criteri di accettazione (es. latenza massima consentita, precisione geospaziale);
- **Procedure di test:** esecuzione manuale o automatizzata, convalidata in presenza della Proponente.

La valutazione del sistema integrato deve considerare i seguenti criteri, documentando i risultati:

- **Copertura dei test:** rispetto ai requisiti di sistema (es. supporto di più sorgenti dati, interpolazione contestuale, gestione di percorsi non predefiniti);
- **Appropriatezza dei metodi:** allineamento con le tecnologie proposte (Kafka, Flink, LangChain) e standard di settore per lo stream processing e l'AI;

- **Conformità ai risultati attesi:** coerenza tra annunci generati e contesto utente (es. target giovane vs. coppia), corretto salvataggio dei dati su storage ottimizzato (ClickHouse_c, PostGIS_c);
- **Fattibilità del test di qualificazione:** capacità di replicare scenari reali (es. picchi di dati, fallimenti di rete) e integrazione con strumenti di visualizzazione (Superset, web-app);
- **Fattibilità di operatività e manutenzione:** modularità dell'architettura, documentazione delle scelte progettuali (es. utilizzo di HiveMQ vs. RabbitMQ) e gestione degli eventuali problemi aperti.

II - 2.9. Test di qualifica del sistema

Il testing di qualificazione per NearYou deve verificare:

- Conformità ai requisiti (simulatori GPS, LLM, dashboard base);
- Funzionalità critiche: generazione annunci contestuali (es. target specifici), integrazione con broker (Kafka/HiveMQ), persistenza dati (ClickHouse/PostGIS).

Criteri di valutazione:

- **Copertura test:** Requisiti avanzati (interpolazione contestuale, visualizzazione lato utente).
- **Conformità:** Correlazione annunci-contesto utente, precisione geospaziale, prestazioni storage.
- **Fattibilità operativa:** Architettura modulare, documentazione chiara (scelte tecnologiche, es. LangChain_c).

Post-verifica:

- Consegnare pacchetto software con codice, configurazioni, dashboard (Superset/Grafana_c) e web-app demo.
- Stabilire baseline tramite versionamento (Git) e documentazione progettuale aggiornata.

II - 2.10. Installazione del software

Il fornitore deve sviluppare un piano di installazione del sistema NearYou, considerando:

- **Ambiente di destinazione:** Configurazione di server/cloud per componenti chiave (message broker, stream processing, database, LLM_c);
- **Risorse necessarie:** Librerie Python (es. Faker_c), container Docker (per Kafka, Flink, Superset), accesso a modelli LLM (es. API OpenAI, Groq Cloud o open-source);
- **Supporto alla configurazione:** Assistenza alla Proponente (*Sync Lab*) per integrare dataset predefiniti, percorsi simulati e parametri di personalizzazione annunci.

In caso di sostituzione di sistemi esistenti, garantendo la compatibilità dei formati dati. Il piano deve includere schemi logici e istruzioni dettagliate per l'avvio delle simulazioni (GPS_c, stato fisico).

Esecuzione dell'installazione: Il fornitore installerà il software conformemente al piano, garantendo:

- **Inizializzazione corretta:** Avvio dei broker (Kafka/HiveMQ), connessione allo stream processing (Flink), configurazione dello storage (ClickHouse/PostGIS) e integrazione con LLM_G (LangChain);
- **Funzionamento atteso:** Verifica che i dati simulati generino annunci contestuali e che le dashboard (Superset/Grafana/web-app) visualizzino posizioni e messaggi in tempo reale;

II - 2.11. Supporto all'accettazione del software

Il team di sviluppo dovrà coordinarsi in tutte le fasi di verifica e validazione della soluzione, ponendo particolare attenzione all'integrazione dei simulatori GPS, al corretto funzionamento del message broker per lo streaming dei dati e all'implementazione dei moduli di generative AI per la creazione di annunci personalizzati. Le prove, siano esse a livello unitario o integrato, devono dimostrare che l'intero sistema - dalla raccolta e memorizzazione dei dati alla visualizzazione in dashboard - soddisfi i requisiti tecnici e funzionali previsti.

Per ogni fase, è necessario documentare accuratamente i risultati, includendo le metriche prestazionali e il livello di copertura dei test (minimo 80%), in modo da identificare e risolvere eventuali criticità.

III. Processi di supporto

III - 1. Documentazione

III - 1.1. Scopo

Il processo di documentazione ha lo scopo di garantire tracciabilità, coerenza e qualità nelle attività del progetto **NearYou**, allineandosi alle specifiche del Capitolato C4 di *Sync Lab*. Le norme definiscono linee guida per la gestione della documentazione tecnica e di progetto, con particolare attenzione all'integrazione di tecnologie avanzate (IA, data streaming, geolocalizzazione) e al rispetto dei criteri di completamento definiti dal proponente.

III - 1.2. Descrizione

Questa sezione regola la creazione, revisione e approvazione di tutti i documenti relativi al progetto, inclusi:

- Documentazione tecnica (architettura, scelte tecnologiche, test);
- Manuali utente e cliente;
- Documentazione per gli stakeholder (demo, presentazioni finali).

Le norme garantiscono conformità con i requisiti del capitolato, come l'uso di **LangChain_G** per gli LLM, l'integrazione di **Apache Kafka_G** per lo streaming, e i criteri di testing (copertura $\geq 80\%$).

III - 1.3. Aspettative

- **Uniformità:** Utilizzo di template predefiniti per tutti i documenti, inclusi sezioni dedicate a:
 - Scelte tecnologiche (es. motivazioni per Kafka vs RabbitMQ);
 - Dettagli sull'integrazione LLM e personalizzazione annunci;
 - Risultati dei test end-to-end e metriche di copertura.
- **Conformità al Capitolato:** Rispetto delle specifiche *Sync Lab* (es. dashboard con Superset/Grafana, simulazione dati GPS).
- **Tracciabilità:** Registrazione delle decisioni progettuali e allineamento con i referenti aziendali.

III - 1.4. Ciclo di vita

Il ciclo di vita di un documento prevede sette fasi principali:

1. **Definizione o modifica del template:** Nella prima fase si procede con la definizione o la modifica del modello (template) specifico per il documento. Questo template stabilisce la struttura, gli stili di formattazione e include elementi essenziali quali titolo, autore, data di creazione e altre informazioni basilari necessarie per la corretta identificazione del documento.
2. **Pianificazione e assegnazione delle sezioni:** Successivamente, le varie sezioni del documento vengono accuratamente pianificate e assegnate ai rispettivi Redattori, ciascuno responsabile per la redazione della propria parte in linea con le specifiche delle Norme di Progetto.
3. **Raccolta dei contenuti e stesura:** In questa fase intermedia, i Redattori raccolgono tutto il materiale necessario e procedono a realizzare una prima versione (bozza) del documento.
4. **Redazione definitiva del documento:** Durante la quarta fase, i Redattori elaborano ulteriormente le proprie sezioni, correggendo e perfezionando i contenuti, in caso di necessità, così da assicurare la conformità al modello stabilito e alle Norme di Progetto.
5. **Verifica dei contenuti:** La quinta fase prevede un controllo approfondito da parte dei Redattori sui contenuti redatti, assicurandosi che questi rispettino integralmente le Norme di Progetto e siano privi di errori logici, tecnici o di compilazione.
6. **Revisione generale del documento:** In questa fase, il documento viene sottoposto a una revisione da parte di un Verificatore incaricato, che ha il compito di assicurare la coerenza complessiva, la correttezza delle modifiche effettuate e la rispondenza del documento agli standard di qualità previsti.
7. **Approvazione finale e pubblicazione:** Infine, il documento giunge alla fase conclusiva, in cui un Responsabile effettua l'approvazione finale, certificandone la qualità e l'idoneità al rilascio. Dopo l'approvazione ufficiale, il documento viene pubblicato e distribuito nella sua versione definitiva.

III - 1.5. Sistema di composizione tipografica

Per garantire coerenza e semplicità nella produzione della documentazione, il gruppo ha adottato **Typst** come strumento principale, in sostituzione di LaTeX. La scelta è motivata dai seguenti vantaggi:

- **Semplicità d'uso:** Sintassi simile a Markdown, ideale per redattori con competenze eterogenee.
- **Programmabilità avanzata:** Supporto nativo per logiche condizionali e gestione dinamica dei contenuti, utile per documenti tecnici complessi (es. integrazione di diagrammi architetturali dal capitolato).
- **Velocità di compilazione:** Generazione immediata di output PDF, ottimizzando i tempi di revisione.
- **Coerenza grafica:** Template predefiniti nella cartella `template/` della repository `AlphaCode-docs-file`, allineati alle richieste del capitolato.

III - 1.6. Struttura dei documenti

Ogni documento segue una struttura standardizzata per garantire tracciabilità e chiarezza, conforme alle aspettative del proponente *Sync Lab*.

III - 1.6.1. Intestazione

La prima pagina include:

- **Logo del gruppo e dell'università** (`imgs/group_logo.png`).
- **Nome del documento** (es. «Analisi dei Requisiti»).
- **Data** della creazione del documento.
- **Responsabile, Redattori, Validatori:** È un elenco dei ruoli ricoperti dai vari membri nel corso della stesura del documento (es. risultano due responsabili elencati nel caso in cui una versione sia stata scritta sotto un determinato responsabile e la successiva in una rotazione dei ruoli differente). Ruoli definiti in base alle competenze (es. redattore esperto in A.I. per sezioni LLM_G).
- **Tabella versionamento:** Partendo dalla più recente in alto verso la prima in basso, contiene una breve descrizione delle modifiche fatte, chi le ha applicate e chi le ha successivamente verificate. In caso di unica versione, la suddetta tabella non risulterà visibile.
- **Versione attuale:**(e.g. `v0.3.1`).

III - 1.6.2. Registro delle modifiche

La seconda pagina è dedicata alla tabella di versionamento, la quale permette di tenere traccia delle modifiche applicate al documento. La tabella riporta le seguenti informazioni:

- **Versione:** il numero di versione del documento;
- **Data:** data di stesura della relativa versione del documento;
- **Descrizione:** un'introduzione sintetica alle modifiche effettuate.
- **Autori:** membri che hanno effettuato le modifiche;
- **Validatori:** membri che hanno approvato le modifiche;

III - 1.6.3. Indice

Organizzato per capitoli e sezioni, con riferimenti alle pagine.

III - 1.6.4. Corpo del documento

Strutturato in capitoli e sottosezioni con eventuali:

- **Descrizioni tecniche:** Dettagli su tecnologie (es. **PostGIS** per dati geospaziali).
- **Diagrammi:** Riproduzione degli schemi architettureali del capitolato.

III - 1.6.5. Corpo del verbale

Include:

- **Informazioni sulla riunione:**
 - Luogo (fisico o virtuale), data, ora, partecipanti (interni/esterni).
- **Ordine del giorno:** Elenco temi discussi (es. «Integrazione LLM con Apache Flink»).
- **Sintesi e decisioni:** Breve riassunto e accordi presi.
- **Consigli ricevuti:** Solo per i verbali esterni, riguarda quanto ci è stato suggerito dall'azienda su come procedere.
- **Attività individuate:** In formato tabulare, comprendendo l'eventuale Issue di GitHub associata, la descrizione dell'attività e gli assegnatari (stabiliti sulla base dei loro ruoli secondo la rotazione corrente)

III - 1.6.6. Documenti del progetto

Elenco dei documenti obbligatori, allineati al capitolato C4:

1. **Norme di Progetto** (*questo documento*).
2. **Piano di Progetto:** Timeline e allocazione risorse.
3. **Piano di Qualifica:** Metriche di testing (copertura $\geq 80\%$).
4. **Analisi dei Requisiti:** Casi d'uso (es. simulazione GPS, generazione annunci).
5. **Glossario:** Definizioni tecniche (es. «LLM», «stream processing»).
6. **Lettera di presentazione:** Lettera di presentazione ad RTB compiuta.
7. **Verbali Interni/Esterni:** Registri degli incontri con *Sync Lab*.