Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica

Integrazione agente IA in un sito e-commerce con interfaccia conversazionale

Tesi di Laurea

Relatore
Prof. Vardanega Tulio

Laureando
Eghosa Matteo Igbinedion-Osamwonyi
Matricola 2042888

Anno Accademico 2024-2025

Sommario

Il presente elaborato documenta le attività svolte durante il periodo di stage (circa 300 ore) presso *Halue S.r.l.* e descrive l'inserimento nel contesto di stage, la progettazione e la realizzazione di un prototipo di interfaccia conversazionale per un sito e-commerce nel settore skincare. Il lavoro comprende l'analisi dei requisiti, la progettazione architetturale di un sistema agentico integrato con modelli di linguaggio e retrieval-augmented generation (RAG), l'implementazione di connettori verso piattaforme (ad es. Shopify e Sanity), lo sviluppo del proof-of-concept e la verifica tramite test end-to-end.

Struttura del documento. Il testo è organizzato nei seguenti capitoli e sezioni principali:

1. **Capitolo 1 — Azienda:** contesto aziendale, obiettivi, tecnologie e processi interni.
2. **Capitolo 2 — Stage:** descrizione del progetto di stage, motivazioni, pianificazione, metodo di lavoro e strumenti adottati.
3. **Capitolo 3 — Sviluppo:** requisiti funzionali e non funzionali, progettazione dell'architettura (storefront, integrazioni, database vettoriale), descrizione delle funzionalità implementate (chat, function calling, chaining, multi-agent, token streaming), test effettuati e problemi riscontrati.
4. **Capitolo 4 — Conclusioni:** valutazione degli obiettivi raggiunti lato progetto e lato personale, retrospettiva e la valutazione dell'esperienza di stage.
5. **Appendici e materiali complementari:** glossario, elenchi di figure e tabelle.

Convenzioni tipografiche. Per garantire chiarezza e uniformità nella scrittura del documento sono state adottate le seguenti convenzioni tipografiche:

Lingua: italiano.

Carattere e corpo: Times New Roman (o equivalente); corpo del testo 12 pt; titoli dei capitoli 14 pt (o come richiesto dal regolamento), interlinea 1.5.

Intestazioni: numerazione gerarchica (es. 1, 1.1, 1.1.1), titoli dei paragrafi in grassetto.

Allineamento e margini: testo giustificato; margini standard (ad es. 2.5 cm su tutti i lati) salvo diversa indicazione.

Numerazione pagine: numerazione romana per frontespizi e sommario (i, ii, ...); numerazione araba a partire dall'introduzione / Capitolo 1.

Figure e tabelle: didascalie concise sotto la figura/tabella; numerazione progressiva (Figura 1.1, Tabella 2.1); riferimenti alle figure nel testo.

Codice e output: font monospace (es. `Consolas` o `Courier New`), corpo 10 pt, blocchi di codice delimitati e con caption descrittiva se presenti.

Abbreviazioni e termini tecnici: alla prima occorrenza la forma estesa seguita dall'acronimo tra parentesi; uso coerente dell'acronimo in seguito.

Citazioni e bibliografia: seguire lo stile indicato dal relatore (se non specificato, mantenere uno stile coerente come APA o IEEE in tutto il documento).

Note tipografiche: termini in lingua straniera o parole chiave in corsivo; evitare uso eccessivo di maiuscole e colori non necessari.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Azienda

1.1 Descrizione generale

Le principali fonti di informazione relative all'azienda ospitante derivano dalla consultazione del *sito web* aziendale, effettuata durante la fase di scelta dello stage, e dalle interazioni con il tutor aziendale e i membri del team nelle prime fasi conoscitive. Poiché la mia postazione era nella stessa stanza del team nei giorni di presenza in azienda, ho inoltre raccolto informazioni di tipo informale osservando il lavoro quotidiano e ascoltando le conversazioni del gruppo. Le informazioni non sono state tutte disponibili prima dell'inizio del progetto, ma si sono accumulate gradualmente durante l'intero periodo di stage.

Halue S.r.l. è una società benefit di consulenza tecnica orientata all'adozione di soluzioni basate sull'intelligenza artificiale e alla fornitura di servizi sia *B2C* (business-to-consumer) sia *B2B* (business-to-business). Le attività dell'azienda si articolano in diversi servizi, tra cui: progettazione e gestione di soluzioni per l'*e-commerce* (commercio elettronico), implementazione di sistemi di gestione delle relazioni con i clienti (*CRM*, Customer Relationship Management), servizi legati all'IA (ad es. sviluppo di agenti intelligenti) e percorsi di formazione rivolti ai clienti.

Contesto organizzativo

Le informazioni relative al contesto organizzativo derivano sia dalle discussioni con il tutor aziendale sull'andamento del progetto, sia dalle conversazioni con altri membri del team. L'organizzazione concilia modalità di lavoro ibride e in *full remote*. Il team tiene *stand-up meeting* quotidiani per allinearsi sul lavoro svolto nella giornata precedente e sugli obiettivi della giornata corrente; periodicamente vengono organizzati incontri informali per mantenere buoni rapporti fra i membri e rinsaldare i valori aziendali.

La comunicazione interna si sviluppa su più livelli, con strumenti scelti in funzione delle esigenze organizzative:

- richieste di aiuto di carattere informale vengono spesso rivolte verbalmente; richieste più dettagliate o tecniche vengono inserite nello strumento di gestione dei progetti, dove ogni membro può *postare* dubbi o proposte all'interno del proprio ramo di lavoro;
- richieste di *meeting* o chiarimenti che richiedono l'accordo su un orario sono gestite tramite la piattaforma di messaggistica condivisa dal team;
- infine, le comunicazioni di natura amministrativa o burocratica vengono inviate tramite Gmail.

Contesto produttivo

Le informazioni sul contesto produttivo provengono dalle interazioni con il tutor e dai colloqui con i colleghi. Dal punto di vista produttivo, l'azienda privilegia il rapido

ingresso delle soluzioni nel mercato, adottando architetture che riducano gli ostacoli al *deployment* (messa in distribuzione del software in ambiente operativo). Ho riscontrato l'uso dell'approccio *Agile* (metodologie iterative e incrementali per adattarsi rapidamente ai cambiamenti dei requisiti) con gestione tramite *backlog* (elenco prioritizzato di attività) e strumenti di tracciamento. La comunicazione interna è adattata alle necessità operative.

Sono presenti ambienti separati per il *testing* (collaudo) e per la produzione, oltre a procedure di rilascio automatizzate predisposte per i singoli progetti. Si alternano attività di *delivery* (consegna/erogazione del servizio) su progetti custom e attività di supporto e manutenzione post-consegna.

Clientela

Le informazioni sulla clientela derivano dall'analisi del sito web aziendale condotta nella fase di scelta dello stage e dall'osservazione dei progetti a cui ho partecipato. La clientela va dalle piccole e medie imprese con esigenze di commercio elettronico fino a grandi committenti che richiedono integrazioni complesse e soluzioni *CRM* articolate. Sono presenti commesse nel settore privato (retail, distributori, operatori *B2B*) e interventi rivolti a processi interni di organizzazioni di maggiori dimensioni.

Propensione all'innovazione

La propensione all'innovazione dell'azienda è emersa chiaramente dall'analisi del progetto a me assegnato e dal modo in cui esso si è integrato nel contesto produttivo. Questa inclinazione si manifesta attraverso un'attenzione costante alla formazione interna, l'adozione di architetture e pratiche tecnologiche moderne e la sperimentazione su iniziative legate all'intelligenza artificiale. Al contempo, l'approccio rimane pragmatico: quando i vincoli operativi, i requisiti di affidabilità o i tempi di consegna lo richiedono, l'azienda privilegia soluzioni consolidate per garantire stabilità e robustezza.

Ho osservato una chiara divisione tra attività orientate allo sviluppo operativo e attività focalizzate sulla ricerca e sperimentazione. L'attività di ricerca comprende indagini tecniche per valutare nuove tecnologie e framework, lo sviluppo di *proof-of-concept* (*PoC*: prototipo sperimentale) e prototipi per dimostrare la fattibilità di idee innovative, sperimentazioni comparative e benchmark per valutare prestazioni, scalabilità e costi, oltre alla preparazione di dimostrazioni e documentazione tecnica da sottoporre a *stakeholder* (persona o gruppo che ha interesse e influenza attività e decisioni di un'organizzazione o di un progetto).

La parte del team dedita alla ricerca crea e mette a disposizione un ambienti di sperimentazione controllato con branch sperimentali nel sistema di controllo versione e procedure formali per la validazione dei PoC prima di un eventuale inserimento in produzione.

Nel complesso, la propensione all'innovazione risulta essere sia presente che ben strutturata: l'azienda favorisce l'emergere di nuove soluzioni e competenze mantenendo al contempo processi e controlli tecnici volti a garantire la qualità e la stabilità dei prodotti e dei servizi offerti.

1.2 Obiettivi e valori

Obiettivi

L'azienda persegue obiettivi strategici chiari e complementari: da un lato il rafforzamento e l'aggiornamento continuo delle competenze interne attraverso progetti di ricerca e sviluppo e percorsi formativi strutturati; dall'altro l'orientamento al cliente mediante offerte modulari e scalabili, pensate per ampliare la base clienti includendo sia piccole e medie imprese sia grandi committenti e, quando opportuno, pubbliche amministrazioni. La strategia tecnologica è focalizzata sull'innovazione applicata: lo sviluppo di *proof of concept* e soluzioni progettate per il *cloud* favorisce scalabilità e tempi di immissione sul mercato ridotti. Infine, la sostenibilità operativa e l'efficien-

za dei processi sono obiettivi trasversali, perseguiti con l'ottimizzazione dell'uso delle risorse per contenere i costi e ridurre l'impatto ambientale dei servizi erogati.

Valori promossi

I valori aziendali sostengono e rendono praticabili gli obiettivi strategici: la trasparenza e la responsabilità sono poste come fondamento dei rapporti interni e delle relazioni con i clienti, favorendo chiarezza nelle comunicazioni e responsabilità nelle decisioni. La collaborazione e la condivisione delle conoscenze sono considerate essenziali per evitare *silos* informativi e per accelerare la diffusione delle competenze; ciò si traduce in pratiche quotidiane come revisioni tecniche, sessioni di formazione interna e condivisione di documentazione. L'etica professionale e la *compliance* normativa rivestono un ruolo centrale soprattutto nei progetti con vincoli regolamentari: si presta particolare attenzione alla protezione dei dati personali, alla sicurezza e alle procedure richieste per operare in settori regolamentati, garantendo così affidabilità e conformità.

Collegamento tra obiettivi, valori e contesto operativo

Le scelte tecnologiche e i processi interni sono allineati agli obiettivi e ai valori dichiarati: l'adozione di architetture basate su microservizi e *container*, *pipeline CI/CD* e sistemi di monitoraggio centralizzato facilita il rilascio rapido e controllato di nuove funzionalità, mantenendo al contempo la qualità e la stabilità di sistema. L'adozione di pratiche *Agili* e la gestione delle richieste tramite sistemi di *ticketing* consentono di prioritizzare il lavoro, tracciare il debito tecnico e coordinare interventi di manutenzione e *delivery*. Questa combinazione di agilità operativa e rigore procedurale permette di conciliare la necessità di innovare rapidamente con l'esigenza di rispettare vincoli normativi e contrattuali imposti da una *clientela* eterogenea. Inoltre, gli investimenti in *PoC* e le collaborazioni esterne con partner tecnologici e realtà accademiche rafforzano la cultura dell'apprendimento e della sperimentazione, offrendo percorsi strutturati per valutare nuove idee prima della loro integrazione in produzione.

Impatto sulle relazioni interne e sul mio inserimento

Gli obiettivi e i valori aziendali hanno influenzato direttamente il mio inserimento operativo: mi è stata concessa ampia autonomia nello svolgimento delle attività, pur mantenendo un contatto costante con il *team* attraverso momenti di sincronizzazione e confronto. L'*onboarding* ha previsto l'assegnazione di una postazione in ufficio e l'integrazione nei canali di comunicazione del *team* (ad esempio *Slack*), strumenti che hanno facilitato l'accesso alle informazioni, la partecipazione a *stand-up* e la visibilità sul *backlog* di progetto. Questo ambiente ha favorito sia il mio coinvolgimento in attività di sviluppo pratico sia l'esposizione a fasi di ricerca e *PoC*, permettendomi di apprendere rapidamente le pratiche aziendali—dal flusso di rilascio alle revisioni tecniche—e di ricevere *feedback* strutturati volti al miglioramento continuo.

1.3 Tecnologie utilizzate

Dal punto di vista delle piattaforme, degli strumenti operativi e delle pratiche di deployment, l'ambiente tecnologico osservato durante lo stage combina soluzioni *enterprise* consolidate con strumenti e approcci moderni orientati alla scalabilità, all'integrazione e all'automazione. Di seguito viene fornita una descrizione estesa delle tecnologie principali e del loro impatto operativo.

Salesforce Commerce Cloud *Salesforce Commerce Cloud* e relativi moduli *Service Cloud* / *Marketing Cloud* (suite *enterprise* per commercio, servizio clienti e automazione marketing): queste piattaforme vengono impiegate per realizzare soluzioni di commercio elettronico e gestione clienti sia in ambito *B2B* che *B2C*. Nell'uso osservato, la suite supporta integrazioni con sistemi esterni per sincronizzare cataloghi, ordini e dati anagrafici; consente la personalizzazione della *customer experience* tramite regole di business e strumenti di segmentazione, e automatizza attività di marketing e customer care. Dal punto di vista operativo richiede competenze specifiche per la gestione

delle estensioni, delle API e dei processi di deployment verso ambienti proprietari della piattaforma.

Bloomreach *Bloomreach* (piattaforma per ricerca avanzata e personalizzazione della *customer journey*): utilizzata in progetti di scala per migliorare la rilevanza delle ricerche interne al sito e per orchestrare contenuti personalizzati lungo il percorso di acquisto. Bloomreach funge da livello di personalizzazione e ricerca ad alto throughput, spesso integrato con il *CMS* e il *CRM* per arricchire i profili utente e abilitare esperienze omnicanale.

deployment Cloud e piattaforme di *deployment*: *AWS*, *Google Cloud* e *Heroku* per hosting, provisioning di risorse scalabili e ambienti di *staging* e *production*. Queste piattaforme vengono impiegate per ospitare servizi applicativi, database e risorse di caching; la scelta tra le piattaforme è dettata da vincoli di progetto, integrazioni richieste e costi operativi. Sul cloud si realizzano inoltre configurazioni per il *scaling* automatico, il bilanciamento del carico e la gestione dei certificati TLS. Per l'infrastruttura è comune l'adozione di pratiche di *infrastructure as code* (ad es. strumenti che consentono il provisioning ripetibile delle risorse).

iPaaS Strumenti di integrazione: approcci punto a punto e soluzioni *iPaaS* (*Integration Platform as a Service*: piattaforme per orchestrare integrazioni) per orchestrare flussi dati fra *CRM*, *ERP*, piattaforme e-commerce e *CMS*. L'impiego di un *iPaaS* semplifica la gestione degli *endpoint*, la trasformazione dei payload e la gestione degli errori/ritentativi, riducendo la complessità rispetto a molteplici integrazioni punto-a-punto. Nei progetti osservati, le integrazioni includono sincronizzazione ordini, anagrafiche clienti, disponibilità di magazzino e tracking degli eventi di marketing.

Gestione progetto Strumenti di gestione progetto e comunicazione: *Jira* per il tracciamento delle attività e *Slack* per la comunicazione operativa quotidiana. *Jira* viene usato per definire le *issue*, assegnare priorità, documentare criteri di accettazione e tenere traccia dello stato di avanzamento; *Slack* è lo strumento preferito per aggiornamenti rapidi, notifiche dagli strumenti di integrazione (build, deploy, alert) e conversazioni sincrone o asincrone tra membri del *team*.

Git Controllo versione e *pipeline*: uso diffuso di *Git*, *pipeline* di integrazione continua (*CI*) e distribuzione continua (*CD*) e *test* automatici (ad esempio *unit tests* e *integration tests*) prima del rilascio. Il flusso di lavoro tipico prevede sviluppo su *feature branches*, apertura di *pull request* per la revisione del codice (*code review*), esecuzione automatica di test nella *pipeline* e successiva promozione verso ambienti di *staging* e infine *production*. Strategie di rilascio come *blue/green* o *canary* possono essere adottate per minimizzare il rischio in produzione, così come meccanismi di rollback automatizzato in caso di regressione.

A livello operativo queste tecnologie si traducono in pratiche concrete (osservate però solo in parte durante lo stage) che includono:

- sviluppo su rami funzionali con revisioni del codice (*code review*) formali e commenti tracciati tramite *pull request* o meccanismi equivalenti;
- creazione e gestione di ambienti distinti di *staging* e *production* sul cloud, con pipeline che automatizzano build, test e deployment;
- esecuzione di *test* automatici a livelli differenziati (*unit*, *integration*, *end-to-end*) per garantire qualità e ridurre regressioni;
- strumenti di monitoraggio e *logging* centralizzati per osservabilità e diagnosi (alerting su metriche critiche, raccolta dei log applicativi e centralizzazione dei tracciati di errore);
- pratiche di sicurezza e compliance applicate alle integrazioni e al trattamento dei dati (gestione di credenziali, accessi basati su ruoli, cifratura dei dati sensibili in transito e a riposo).

Implicazioni operative e suggerimenti

L'adozione di questo insieme di tecnologie determina alcuni vincoli e opportunità operative: la complessità delle integrazioni richiede governance dei *data contracts* e politiche chiare di versioning delle API; le piattaforme *enterprise* come *Salesforce Commerce Cloud* richiedono competenze specifiche per personalizzazioni e upgrade; le risorse cloud implicano attenzione alla gestione dei costi e al monitoraggio del consumo. Per migliorare l'efficacia complessiva è raccomandabile formalizzare convenzioni comuni (naming delle *branches*, standard per le *code review*, soglie di qualità per le *pipeline*) e prevedere un minimo di documentazione operativa a supporto dell'onboarding e della manutenzione.

In conclusione, la suite tecnologica osservata mette a disposizione strumenti potenti per costruire soluzioni integrate e scalabili; la sfida principale risiede nell'armonizzare pratiche, automazioni e governance per trasformare la tecnologia in valore ripetibile e manutenibile nel tempo.

1.4 Processi interni

Le informazioni qui riportate derivano da osservazione diretta del mio ambiente di lavoro e da confronti brevi con il tutor interno; durante i due mesi di stage ho svolto il progetto in larga parte in autonomia, perciò la mia percezione dei processi si basa prevalentemente su quanto mi è stato illustrato dal tutor aziendale e su alcune conversazioni avute con i membri del *team*, e non pretende di essere esaustiva.

Processo di sviluppo

Il processo di sviluppo si articola in fasi distinte ma integrate: individuazione e definizione del problema, scomposizione in attività eseguibili, implementazione, verifica e rilascio. Per la gestione del codice e del *versioning* si utilizzano strumenti dedicati (controllo di versione distribuito, repository remoti) e, parallelamente, si impiegano sistemi per tracciare le attività e le priorità. Dalle conversazioni tra i membri del *team* ho rilevato l'esistenza di una fase iniziale di analisi in cui il problema viene chiarito e suddiviso in *task* di dimensione tale da poter essere assegnata a un singolo sviluppatore; questa scomposizione agevola la responsabilizzazione dei singoli e il monitoraggio dei progressi.

Lo strumento visivamente riconoscibile per la creazione, l'assegnazione e il monitoraggio delle *tasks* è *Jira*, che funge da punto di riferimento per lo stato dei lavori, le descrizioni delle attività e i commenti di avanzamento. Durante la fase di sviluppo giornaliera sono previste brevi *stand-up meetings* (riunioni di sincronizzazione) con l'obiettivo di condividere quanto svolto nella giornata precedente, segnalare impedimenti e pianificare le attività immediate; queste riunioni favoriscono l'allineamento del *team* e la rapida emersione di blocchi tecnici.

Al completamento di ogni *task* è prevista una fase di verifica da parte di un altro membro del *team* (code review): tale pratica viene svolta sia per assicurare qualità e conformità agli standard interni sia per favorire la diffusione delle conoscenze. Quando possibile, il rilascio del codice avviene seguendo procedure automatizzate che includono build e test automatici; ove presenti, pipeline di integrazione continua vengono attivate prima del merge verso i rami principali, riducendo il rischio di regressioni.

Processo di organizzazione

L'organizzazione della comunicazione è stratificata in base al livello di formalità e alla finalità: comunicazioni rapide e operative avvengono su *Slack*, che viene utilizzato per la pianificazione quotidiana delle presenze, richieste logistiche e notifiche veloci; per questioni tecniche più specifiche e per lo scambio su singole *tasks* si sfruttano i *branches* e i commenti collegati alle issue in *Jira*, dove rimane tracciata la cronologia delle decisioni e delle discussioni tecniche. Le comunicazioni formali, amministrative o di carattere aziendale più strutturato vengono invece inviate tramite *email*.

La documentazione di progetto è mantenuta in repository condivisi o in spazi dedicati (wiki, documenti di progetto): questo facilita il reperimento di informazioni, la consultazione delle linee guida e la storage delle specifiche. Ho notato che, nonostante

esista una struttura di comunicazione definita, la pratica quotidiana lascia spazio a scambi informali che spesso risolvono rapidamente piccoli problemi ma che possono anche richiedere successiva formalizzazione quando le decisioni impattano sul piano di rilascio.

Non sono stato coinvolto nelle attività di manutenzione continuativa del sistema, né ho avuto discussioni approfondite con il tutor o con i membri del *team* su procedure operative relative al supporto post-rilascio; pertanto non posso fornire dettagli completi su quel processo.

Osservazioni riassuntive e suggerimenti

Dall'esperienza osservativa emergono alcuni punti di forza e margini di miglioramento: la presenza di pratiche consolidate (uso di *Jira*, *stand-up meetings*, code review) favorisce trasparenza e tracciabilità, mentre l'adozione più ampia di procedure formali per la documentazione delle decisioni tecniche potrebbe ridurre la dipendenza dalle comunicazioni informali. Sarebbe utile, a mio avviso, integrare nell'onboarding materiali che esplicitino il ciclo di vita di una *task*, le convenzioni di *versioning* adottate e le regole per le code review, in modo da velocizzare l'inserimento operativo dei nuovi membri del *team*.

Capitolo 2

Stage

Introduzione al capitolo sullo stage

2.1 Strategia

Sezione che riporterà come lo stage si inserisce nella visione strategica da parte dell'aziendale (e dunque la propensione dell'azienda per l'innovazione). Qui descriverò parzialmente il punto 2 (perché) riportato nel file Struttura relazione finale.pdf. e lo concluderò nella sezione successiva.

2.2 Progetto di stage

Sezione in cui verrà illustrato il progetto di stage ricevuto, esplicitando le problematiche applicative che l'organizzazione intende affrontare con il tirocinio, gli obiettivi specifici prefissati e i vincoli operativi e temporali associati. Verrà inoltre evidenziato il rapporto tra la proposta di stage e la strategia più ampia dell'ente ospitante in materia di innovazione (con riferimento al ruolo e alla posizione assunta dal tutor aziendale emerse nel primo incontro) nonché le attività di supporto previste prima, durante e dopo il periodo di tirocinio. Qui concluderò la trattazione del punto 2 (perché) riportato nel file Struttura relazione finale.pdf.

2.3 Motivo della scelta

Sotto-sezione in cui verrà descritto il motivo per cui ho preferito scegliere di fare lo stage presso questa azienda rispetto ad altre, quali sono i miei obiettivi personali che mi sono auto-assegnato nello svolgimento del progetto e come si interconnettono con gli obiettivi dell'azienda.

2.4 Pianificazione

Sezione che descriverà la pianificazione riportata nel piano di lavoro.

2.4.1 Pianificazione settimanale

Sotto-sezione che riporterà in lista il contenuto del lavoro pianificato per lo stage, suddiviso nelle settimane definite a priori.

2.4.2 Requisiti

Sotto-sezione che riporterà la lista dei requisiti per il progetto presenti nel piano di lavoro.

2.5 Metodo di lavoro

Sezione che riporterà il flusso di lavoro uilizzato per lo sviluppo del progetto in accordo con il tutor aziendale. Verranno riportati pianificazione, interazioni con il tutor aziendale, revisioni di progresso, uso di diagrammi, tecniche di analisi e tracciamento dei requisiti, strumenti di verifica, ecc. Qui descriverò il punto 3.a (cosa e come) riportato nel file Struttura relazione finale.pdf.

2.6 Tecnologie utilizzate

2.6.1 Sanity

Sotto-sezione che riporterà la spiegazione e la logica della scelta del CMS Sanity.

2.6.2 Shopify

Sotto-sezione che riporterà la spiegazione e la logica della scelta delle API di Shopify.

Capitolo 3

Sviluppo

Introduzione al capitolo sullo sviluppo

3.1 Requisiti

3.1.1 Obbligatori

Sotto-sezione che riporterà la lista dei requisiti obbligatori discussi e studiati nel dettaglio.

3.1.2 Facoltativi

Sotto-sezione che riporterà la lista dei requisiti facoltativi discussi e studiati nel dettaglio.

3.2 Progettazione

3.2.1 Architettura storefronte

Sotto-sezione che riporterà la descrizione dell'architettura sviluppata per lo storefront. verranno descritti le componenti, le classi e le interazioni tra loro.

3.2.2 Comunicazione con shopify

Sotto-sezione che riporterà la descrizione delle comunicazione con le API di Shopify da parte del sistema.

3.2.3 Comunicazione con sanity

Sotto-sezione che riporterà la descrizione delle comunicazione con le API di Sanity da parte del sistema.

3.2.4 Scleta LLM

Sotto-sezione che riporterà la spiegazione e la logica relativa alla scelta dell'LLM.

3.2.5 Comunicazione con agente

Sotto-sezione che riporterà la descrizione delle comunicazione con l'agente da parte del sistema.

3.2.6 Flusso agentico

Sotto-sezione che riporterà la descrizione del flusso agentico scelto per l'agente.

3.2.7 Database vettoriale

Sotto-sezione che riporterà la spiegazione e la logica della scelta del database vettoriale.

3.2.8 Architettura applicazione

Sotto-sezione che riporterà la descrizione arichietturale dell'applicazione con citate tutte le componenti architetturali principali e descritto come si legano tra loro. Verrà riportato un diagramma rappresentativo del flusso dell'applicazione.

3.3 Funzionalità

3.3.1 Chat page

Sotto-sezione che riporterà la descrizione dell'implementazione della pagine sul frontend per l'interazione con l'agente. Verranno riportati porzioni di codice e immagini per agevolare l'esposizione.

3.3.2 Sanity function calling

Sotto-sezione che riporterà lo scopo dei tool utili all'interazione con le api di sanity, il perché sono stati differenziati per come sono nel progetto e il perché nell'uso di wrappers. Verranno riportati porzioni di codice per agevolare l'esposizione.

3.3.3 Shopify function calling

Sotto-sezione che riporterà lo scopo dei tool utili all'interazione con le api di Shopify, il perché sono stati differenziati per come sono nel progetto e il perché nell'uso di wrappers. Verranno riportati porzioni di codice per agevolare l'esposizione.

3.3.4 Prompt engineering

Sotto-sezione che riporterà la descrizione del prompt engineering usato ai fini di ricevere risposte più consone e coerenti possibili con i requisiti desiderabili. Verranno riportati esempi per agevolare l'esposizione.

3.3.5 Chaining

Sotto-sezione che riporterà la descrizione dell'implementazione del pattern chaining all'interno del flusso agentico. Verranno riportati porzioni di codice e immagini rappresentative del pattern per agevolare l'esposizione.

3.3.6 Multi-agents

Sotto-sezione che riporterà la descrizione dell'implementazione del pattern multi-agent all'interno del flusso agentico. Verranno riportati porzioni di codice e immagini rappresentative del pattern per agevolare l'esposizione.

3.3.7 Token streaming

Sotto-sezione che riporterà la descrizione dell'implementazione dello streaming dei token di risposta generati dall'agente e dei log ai fini dell' UX Verranno riportati porzioni di codice per agevolare l'esposizione.

3.3.8 Logs

Sotto-sezione che riporterà la descrizione dell'output dei logs riassuntivi degli step compiuti e del relativo tempo impiegato dall'agente, il perché, il come (in streaming e non) e il loro ruolo nel contesto dell'esperienza utente. Verranno riportati porzioni di codice per agevolare l'esposizione.

3.3.9 Creazione interfaccia

Sotto-sezione che riporterà la descrizione del processo decisionale nel modo e nel perché riguardo alla creazione di un formato strutturato come output dell'agente e la sua conversione in un'interfaccia sul frontend. Verranno riportati porzioni di codice per agevolare l'esposizione.

3.4 Test

Sezione che riporterà la descrizione dei test scritti ed eseguiti per i tool chiamabili dal sistema. Verranno riportati porzioni di codice per agevolare l'esposizione.

3.4.1 Sanity test

Sotto-sezione che riporterà la spiegazione e il codice relativo ai test scritti e svolti per i tool finalizzati all'interazione tra il sistema e le API di Sanity.

3.4.2 Shopify test

Sotto-sezione che riporterà la spiegazione e il codice relativo ai test scritti e svolti per i tool finalizzati all'interazione tra il sistema e le API di Shopify.

3.5 Problemi riscontrati

Sezione che riporterà I problemi progettuali, tecnologici e applicativi che ho affrontato. Qui descriverò il punto 3.b (cosa e come) relativo al file Struttura relazione finale.pdf.

3.5.1 Tempo invio risposta

Sotto-sezione che riporterà la problematica e la soluzione adottata relativa al tempo richiesto per l'invio di una riposta da parte dell'agente in questione, dall'istante in cui è stato interrogato.

3.5.2 Tempo creazione interfaccia

Sotto-sezione che riporterà la problematica e la soluzione adottata relativa al tempo richiesto per la creazione di un interfaccia dall'istante in cui è stato interrogato l'agente in questione.

3.6 Prodotto finale

Sotto-sezione che riporterà i risultati che ho raggiunto, sia sul piano qualitativo che su quello quantitativo. Qui descriverò il punto 3.c (cosa e come) relativo al file Struttura relazione finale.pdf.

Capitolo 4

Conclusioni

Introduzione al capitolo delle conclusioni

4.1 Obiettivi soddisfatti

Sezione che riporterà gli obbiettivi e i risultati raggiunti sia sul piano qualitativo che su quello quantitativo sia su base personale che sui dati di fatto. Qui descriverò il punto 4.a relativo al file Struttura relazione finale.pdf.

4.1.1 Obiettivi effettivi

Sotto-sezione che riporterà gli obbiettivi e i risultati raggiunti sui dati di fatto.

4.1.2 Obiettivi personali

Sotto-sezione che riporterà gli obbiettivi e i risultati raggiunti dalla mia prospettiva.

4.2 Retrospettiva

Sezione in cui farò la retrospezione dell'esperienza di stage.

4.3 Valutazione esperienza

Sotto-sezione che riporterà una descrizione dell'esperienza dalla mia personale prospettiva. Qui descriverò il punto 4.C relativo al file Struttura relazione finale.pdf.

Glossario

A

Agile – Approccio allo sviluppo del software e alla gestione progetti basato su iterazioni brevi, feedback continui e adattamento ai cambiamenti. Include metodologie come Scrum e Kanban.

B

B2C Business-to-Consumer – Modello di business in cui un'azienda vende prodotti o servizi direttamente al consumatore finale. Esempi tipici: e-commerce al dettaglio, servizi in abbonamento per privati.

B2B Business-to-Business – Modello di business in cui le transazioni avvengono tra imprese; prodotti o servizi rivolti ad altre aziende (software aziendale, forniture industriali, consulenza).

Backlog – Elenco prioritizzato di attività, requisiti o user story che rappresentano il lavoro da svolgere su un prodotto o progetto. Nel contesto Agile esistono tipicamente un product backlog (tutte le feature/attività previste) e uno sprint backlog (selezione per uno sprint).

Branch Branch (ramo) – Linea di sviluppo separata in un sistema di controllo versione (es. Git). I branch permettono di lavorare contemporaneamente su feature, fix o release senza intaccare il ramo principale; tipiche operazioni correlate: commit, merge, rebase e pull request.

C

Cloud-native – Architettura e modalità di sviluppo che sfrutta appieno le caratteristiche del cloud (scalabilità, resilienza, containerizzazione, microservizi).

Code review – Revisione del codice sorgente da parte di uno o più membri del team, con l'obiettivo di migliorare qualità, leggibilità e sicurezza.

D

Delivery – Consegna di un prodotto, servizio o funzionalità al cliente o all'ambiente di produzione; include fasi di build, test, integrazione e deployment. In ambito Agile si parla spesso di "continuous delivery" per indicare rilascio frequente e affidabile.

Deploy Deploy (deployment) – Processo che porta il codice da un ambiente di sviluppo/testing all'ambiente di produzione (o ad un ambiente target). Comprende build, configurazione, esecuzione di script di migrazione, verifica post-deploy e possibilità di rollback. Spesso automatizzato tramite pipeline CI/CD.

F

G

Git – Sistema distribuito di controllo versione, utilizzato per tracciare modifiche al codice sorgente e coordinare il lavoro tra più sviluppatori.

H

I

M

Meeting Incontro (fisico o virtuale) tra partecipanti con uno scopo definito (allineamento, decisione, pianificazione); solitamente ha un'agenda, una durata prevista e produce output (verbale, action items).

O

P

Pipeline – Sequenza automatizzata di fasi che portano il codice sorgente dalla scrittura allo sviluppo, test, integrazione e distribuzione.

PoC Proof of Concept – Prototipo o esperimento preliminare volto a dimostrare la fattibilità tecnica o commerciale di un'idea o soluzione.

Production – Ambiente operativo in cui un'applicazione è effettivamente disponibile e utilizzata dagli utenti finali.

R

Repository Repo – Archivio centralizzato dove viene conservato e versionato il codice sorgente di un progetto.

S

Slack – Piattaforma di messaggistica aziendale in tempo reale per la comunicazione dei team, con canali, messaggi diretti, condivisione file e integrazioni con altri strumenti.

Script – Sequenza di comandi automatizzati che eseguono operazioni ripetitive (ad esempio build, deploy, manutenzione).

Stand-up meetings – Breve riunione quotidiana (tipica di Scrum) in cui i membri del team condividono progressi, piani e ostacoli.

T

Ticket – Registro/formalizzazione di una richiesta, problema o attività (supporto, sviluppo, manutenzione); contiene descrizione, priorità, assegnatario, stato e storico delle azioni intraprese.

Testing – Insieme di attività volte a verificare che un sistema o componente soddisfi i requisiti e funzioni correttamente. Tipologie comuni: unit testing, integration testing, system testing, end-to-end (E2E), acceptance testing e regression testing. Può essere manuale o automatizzato.

U

V

Versioning – Gestione e controllo delle versioni di software o documenti, che permette di tracciare le modifiche, collaborare in modo ordinato, ripristinare stati precedenti e rilasciare aggiornamenti.

W

X

Y

Z