

Analisi dei Requisiti

versione 2.0.0



7commits@gmail.com

Progetto di Ingegneria del Software

A.A. 2024/2025

Destinatari	Responsabile	Redattori	Verificatori
Prof. Tullio Vardanega	Marco Cola	Stefano Dal Poz	Giulia Hu
Prof. Riccardo Cardin		Marco Cola	Ruize Lin
Gruppo 🇮🇹 7Commits		Ruize Lin	Stefano Dal Poz
		Mattia Piva	Giada Rossi
		Giada Rossi	Marco Cola

Registro delle modifiche

Versione	Data	Autori	Verificatori	Descrizione
v2.0.0	2025-08-25	Marco Cola	Ruize Lin	Approvazione per PB
v1.2.2	2025-08-21	Stefano Dal Poz, Marco Cola	Ruize Lin	Ampliate le descrizioni dei Casi d'Uso, corretta la grammatica e alcuni errori nei nomi delle sezioni
v1.2.1	2025-08-18	Stefano Dal Poz, Marco Cola	Ruize Lin	Aggiunti riferimenti al Glossario v2.0.0
v1.2.0	2025-08-14	Stefano Dal Poz	Marco Cola	Modifica del capitolo 1: aggiunti sezione 1.2, 1.3, 1.4
v1.1.0	2025-07-17	Marco Cola	Mattia Piva, Giada Rossi, Ruize Lin	Approvazione modifiche conseguenti ai commenti emersi dalla revisione RTB
v1.0.4	2025-07-16	Giulia Hu, Marco Cola	Mattia Piva, Giada Rossi	Correzione diagrammi UML
v1.0.3	2025-07-15	Ruize Lin, Marco Cola	Stefano Dal Poz, Mattia Piva	Modifiche di UC9, UC10, UC11, UC12, UC14, UC27
v1.0.2	2025-07-14	Giada Rossi, Ruize Lin, Mattia Piva, Marco Cola	Stefano Dal Poz, Giulia Hu	Revisione completa, modifica e integrazione dei requisiti funzionali, di qualità e di vincolo
v1.0.1	2025-07-14	Mattia Piva	Marco Cola, Stefano Dal Poz	separazione dei sottocasi UC1.1, 1.1.1, 1.1.2, 1.1.3, 8.1, 8.1.1, 8.1.2, 8.1.3, 17.1, 17.2 nei rispettivi UC separati. Rielaborazione UC3, UC4, UC6, UC15, Abbozzato UC29
v1.0.0	2025-06-12	Marco Cola	Michele Ogniben	Approvazione per colloquio RTB
v0.2.6	2025-06-11	Giulia Hu	Marco Cola, Giada Rossi	Corrette immagini UML modificati
v0.2.5	2025-06-10	Giada Rossi	Marco Cola, Giulia Hu, Ruize Lin	Aggiunti Casi d'Uso UC7, UC12, UC14, UC16, UC19 e UC25 in seguito a revisione interna e rinumerazione Casi d'Uso di conseguenza

v0.2.4	2025-05-17	Marco Cola	Marco Cola, Giada Rossi, Stefano Dal Poz	Corrette tabelle dei requisiti
v0.2.3	2025-05-15	Stefano Dal Poz	Marco Cola, Giada Rossi, Ruize Lin	Aggiunta priorità nella notazione dei requisiti
v0.2.2	2025-05-15	Giulia Hu, Marco Cola	Marco Cola, Giada Rossi, Ruize Lin	Aggiunte immagini corrette UML Casi d'Uso
v0.2.1	2025-05-10	Marco Cola, Ruize Lin, Mattia Piva	Marco Cola, Giada Rossi	Aggiunti Casi d'Uso UC22: «Salvataggio risultato test» e UC21: «Visualizzazione risultati test»
v0.2.0	2025-05-06	Marco Cola, Ruize Lin	Marco Cola, Giada Rossi, Ruize Lin	Revisione casi d'uso in seguito a colloquio con proponente
v0.1.4	2025-04-28	Marco Cola	Giada Rossi, Michele Ogniben, Ruize Lin, Marco Cola	Corrette tabelle sezioni 4 e 5
v0.1.3	2025-04-26	Marco Cola, Giulia Hu	Giada Rossi, Michele Ogniben, Ruize Lin, Marco Cola	Rimosso UC6 per ridondanza, Aggiunte immagini dei Casi d'Uso da UC1 a UC21
v0.1.2	2025-04-16	Marco Cola	Giada Rossi, Michele Ogniben, Ruize Lin, Marco Cola	Rinumerati UC in seguito all'eliminazione di UC8, rimossi UC26 e UC27 per ridondanza
v0.1.1	2025-04-15	Ruize Lin	Giada Rossi, Michele Ogniben, Ruize Lin, Marco Cola	Corretti casi d'uso UC4, UC8, UC16, Aggiunti UC26, UC27
v0.1.0	2025-04-15	Marco Cola	Giulia Hu, Giada Rossi, Michele Ogniben, Ruize Lin, Marco Cola	Aggiunti link al Glossario
v0.0.9	2025-04-14	Marco Cola	Giulia Hu, Giada Rossi, Michele Ogniben, Ruize Lin, Marco Cola	Iniziate tabelle sezione 4: 4.1, 4.2, 4.3
v0.0.8	2025-04-14	Marco Cola, Ruize Lin	Michele Ogniben, Ruize Lin, Marco Cola	Aggiunti UC18, UC18.1, UC19, UC19.1, UC19.2, UC20, UC21, UC21.1, UC22, UC23, UC24, UC25
v0.0.7	2025-04-13	Marco Cola, Ruize Lin	Michele Ogniben, Ruize Lin, Marco Cola	Aggiunti UC7, UC8, UC9, UC10, UC11, UC12, UC13, UC13.1,

				UC14, UC15, UC16, UC17
v0.0.6	2025-04-13	Marco Cola	Michele Ogniben, Ruize Lin, Marco Cola	Aggiunte generalizzazioni di UC1: UC1.1, 1.2, 1.3, 1.4, 1.5 in quanto relativi alle domande
v0.0.5	2025-04-13	Marco Cola	Ruize Lin, Marco Cola	Aggiunta Sezione 5.2 e tabelle
v0.0.4	2025-04-11	Mattia Piva	Ruize Lin, Marco Cola	Aggiunti Casi d'Uso UC1, UC2, UC3, UC4, UC5
v0.0.3	2025-04-09	Marco Cola	Ruize Lin, Marco Cola	Aggiunte sezioni da 3.1 a 3.3 dei Casi d'Uso
v0.0.2	2025-04-09	Marco Cola	Ruize Lin, Marco Cola	Aggiunta 1) Introduzione e 2) Descrizione generale del Prodotto
v0.0.1	2025-04-08	Marco Cola	Ruize Lin, Marco Cola	Prima Bozza

Indice

1. Introduzione	7
1.1. Scopo del documento	7
1.2. Prospettiva del prodotto	7
1.3. Funzioni del prodotto	8
1.4. Caratteristiche dell'utente	8
1.5. Glossario	8
1.6. Riferimenti	9
1.6.1. Riferimenti normativi	9
1.6.2. Riferimenti informativi	9
2. Descrizione generale del prodotto richiesto da Zucchetti	10
2.1. Obiettivo fissato	10
2.2. Funzionalità del prodotto e requisiti	10
2.2.1. Caratteristiche e requisiti obbligatori:	10
2.2.2. Requisiti opzionali suggeriti:	11
2.3. Caratteristiche utente	11
2.3.1. Utenti del nostro prodotto	11
2.4. Tecnologie ed analisi della struttura di progetto	12
3. Casi d'uso	13
3.1. Obiettivi	13
3.2. Introduzione ai casi d'uso	13
3.3. Attori	13
3.4. Elenco dei casi d'uso	14
3.4.1. UC: Gestione domande e risposte attese	14
3.4.1.1. UC1: Visualizzazione lista domande	14
3.4.1.1.1. UC1.1: Visualizzazione singola domanda	14
3.4.1.2. UC2: Modifica domanda	15
3.4.1.3. UC3: Modifica risposta attesa	15
3.4.1.4. UC4: Eliminazione domanda	16
3.4.1.5. UC5: Annullamento eliminazione domanda	16
3.4.1.6. UC6: Ricerca domande per categoria	17
3.4.1.7. UC7: Aggiunta manuale coppia domanda/risposta	18
3.4.1.8. UC8: Errore «Sono necessarie sia la domanda che la risposta attesa»	18
3.4.1.9. UC9: Importazione domande da file CSV	19
3.4.1.10. UC10: Importazione domande da file JSON	19
3.4.1.11. UC11: Errore caricamento file strutturato	20
3.4.2. UC: Gestione dei set di domande	20
3.4.2.1. UC12: Visualizzazione lista dei set	20
3.4.2.1.1. UC12.1: Visualizzazione singolo set	21
3.4.2.2. UC13: Modifica set	21
3.4.2.3. UC14: Eliminazione set	22
3.4.2.4. UC15: Annullamento eliminazione set	22
3.4.2.5. UC16: Ricerca dei set per categoria di appartenenza	23
3.4.2.6. UC17: Aggiunta manuale set	23
3.4.2.7. UC18: Errore «Il nome del set è obbligatorio.»	24
3.4.2.8. UC19: Aggiunta set da file JSON	24
3.4.2.9. UC20: Aggiunta set da file CSV	25
3.4.2.10. UC21: Errore «Formato non valido.»	25
3.4.3. UC: Configurazione API LLM	26

3.4.3.1.	UC22: Creazione di preset per la configurazione API LLM	26
3.4.3.1.1.	UC22.1: Test di connessione API	26
3.4.3.2.	UC23: Errore «Il nome del preset non può essere vuoto.»	27
3.4.3.3.	UC24: Visualizzazione lista dei preset configurati	27
3.4.3.4.	UC25: Modifica preset	28
3.4.3.5.	UC26: Elimina preset	28
3.4.4.	UC: Avvio test	29
3.4.4.1.	UC27: Esecuzione test	29
3.4.4.1.1.	UC27.1: Visualizzazione preset di configurazione API salvati . .	30
3.4.4.1.2.	UC27.2: Visualizzazione set di domande salvate	30
3.4.4.2.	UC28: Errore «Incorretta chiave API fornita»	31
3.4.4.3.	UC29: Valutazione automatica	31
3.4.5.	UC: Visualizzazione dei risultati	32
3.4.5.1.	UC30: Visualizzazione risultati	32
3.4.5.1.1.	UC30.1: Visualizzazione risultati per singola domanda	33
3.4.5.1.2.	UC30.2: Visualizzazione metrica per singola risposta	34
3.4.5.2.	UC31: Salvataggio risultati del test	34
3.4.5.3.	UC32: Download risultati del test	35
3.4.5.4.	UC33: Caricamento risultati da file	35
3.4.5.5.	UC34: Errore «Caricamento risultati da file non riuscito»	36
3.4.5.6.	UC35: Visualizzazione storico dei test	36
3.4.5.6.1.	UC35.1: Visualizzazione elemento esecuzione test	37
3.4.5.7.	UC36: Ricerca con applicazione del filtro su risultati test	37
3.4.5.7.1.	UC36.1: Ricerca storico dei test per nome del set	38
3.4.5.7.2.	UC36.2: Ricerca storico dei test per LLM	39
3.4.5.7.3.	UC36.3: Ricerca storico dei test per risultati di valutazione	39
3.4.5.8.	UC37: Confronto tra esecuzioni diverse di test	40
3.4.5.9.	UC38: Visualizzazione risultato confronto esecuzioni	40
3.4.5.9.1.	UC38.1: Visualizzazione elemento confronto esecuzione	41
3.4.5.9.2.	UC38.2: Visualizzazione confronto esecuzione singola domanda .	42
4.	Requisiti	43
4.1.	Requisiti di funzionalità	44
4.2.	Requisiti di qualità	47
4.3.	Requisiti di vincolo	48
5.	Tracciamento dei requisiti	49
5.1.	Riepilogo	50
5.2.	Conclusioni	51

1. Introduzione

1.1. Scopo del documento

L'*Analisi dei Requisiti_G* è un documento essenziale in ogni progetto di sviluppo software che intenda aderire agli standard di qualità previsti dalla disciplina dell'Ingegneria del Software. Il suo scopo principale è definire in modo chiaro e strutturato le *funzionalità_G* che il sistema dovrà offrire, ossia i *requisiti_G* obbligatori, desiderati e opzionali che il prodotto dovrà soddisfare per risultare conforme alle esigenze espresse dal *proponente_G*.

L'analisi non deve proporre direttamente una soluzione al problema, ma deve tener conto della fattibilità tecnologica, tracciando così un confine netto tra la definizione del problema e la sua progettazione (design).

In particolare, le finalità di questo documento possono essere sintetizzate nei seguenti punti:

- **Esplicitare le esigenze del proponente:** il documento raccoglie e formalizza le aspettative che il proponente ha nei confronti del prodotto software, includendo sia i requisiti esplicitamente indicati nel *capitolato_G* sia quelli impliciti emersi durante le interviste e l'analisi.
- **Ridurre le ambiguità comunicative:** fornisce un punto di riferimento condiviso tra il team di sviluppo e il proponente, assicurando una comprensione univoca dei requisiti e delle funzionalità attese.
- **Garantire il tracciamento dei requisiti:** consente di mappare le esigenze del proponente attraverso l'identificazione, classificazione e organizzazione dei *casi d'uso_G* e dei relativi requisiti, suddivisi in funzionali, qualitativi e di vincolo.
- **Fornire una base solida per la progettazione del sistema:** il documento rappresenta un input fondamentale per l'attività di progettazione, in quanto definisce cosa il sistema deve fare, senza però specificare come. Questo permette al team di identificare soluzioni tecnicamente adeguate.
- **Verifica e Validazione dei requisiti:** la *verifica dei requisiti_G* mira a garantire che i requisiti siano formalmente corretti, completi e privi di contraddizioni. La *validazione dei requisiti_G* invece, si concentra sull'allineamento tra quanto definito nei requisiti e le reali esigenze del proponente, assicurandosi che il prodotto finale rispecchi le aspettative.
- **Minimizzare i rischi e ottimizzare i costi di sviluppo:** una definizione accurata e condivisa dei requisiti riduce il rischio di errori e ritardi nelle fasi successive, contribuendo a contenere i costi e migliorare l'efficienza del progetto.
- **Formalizzare graficamente i requisiti:** al termine dell'analisi, i risultati vengono rappresentati in maniera formale tramite diagrammi dei casi d'uso, utili per visualizzare le interazioni tra gli *attori_G* e le funzionalità del *sistema software_G*.

1.2. Prospettiva del prodotto

Il sistema si colloca come strumento a supporto della valutazione e gestione di test basati su modelli linguistici (*LLM_G*). In particolare, il prodotto si propone come:

- Interfaccia di gestione delle domande e risposte attese, per la costruzione di set di test strutturati;
- Piattaforma di esecuzione dei test, che consente di sottoporre insiemi di domande a diversi modelli e ottenere risposte;
- Ambiente di analisi dei risultati, con possibilità di visualizzare, salvare ed esportare le metriche raccolte.

Il sistema non sostituisce l'infrastruttura dei *modelli LLM_G* ma funge da middleware applicativo che semplifica la gestione dei test e centralizza la reportistica. È previsto l'utilizzo in contesti accademici, di ricerca e aziendali, con particolare attenzione a team che necessitano di uno strumento unitario per la validazione dei modelli.

1.3. Funzioni del prodotto

Il sistema offrirà le seguenti funzioni principali:

- **Gestione domande e risposte attese:** creazione, modifica, eliminazione, ricerca e importazione/esportazione da file strutturati (*CSV_G*, *JSON_G*);
- **Gestione set di domande:** organizzazione delle domande in collezioni tematiche, con possibilità di aggiunta manuale o tramite file;
- **Configurazione delle API_G LLM_G:** definizione e salvataggio di preset di configurazione con possibilità di test di connessione;
- **Avvio ed esecuzione dei test:** selezione dei set e dei preset disponibili per lanciare un test automatico;
- **Gestione degli errori:** notifiche in caso di input mancanti, file malformati o *chiavi API_G* non valide;
- **Visualizzazione e analisi dei risultati:** report dettagliati con metrica per risposta, possibilità di salvataggio, download e caricamento da file esterni;
- **Storico dei test:** tracciamento delle esecuzioni precedenti e funzioni di ricerca e filtraggio sui risultati.

1.4. Caratteristiche dell'utente

Il sistema è progettato per essere utilizzato esclusivamente da utenti sviluppatori, i quali possiedono competenze tecniche di base o intermedie nell'ambito della programmazione e dell'integrazione di API. L'utente sviluppatore è interessato a:

- Gestire set di domande e risposte attese per la validazione di modelli linguistici (*LLM_G*);
- Configurare e testare preset di connessione alle *API_G* dei *modelli_G*
- Eseguire test automatizzati e interpretarne i risultati;
- Esportare e riutilizzare i dati prodotti dal sistema.

Poiché il prodotto è rivolto a una categoria di utenti con conoscenze tecniche, l'interfaccia privilegia la completezza delle funzionalità e la flessibilità di configurazione rispetto a esigenze di semplicità estrema. Tuttavia, sono previste funzionalità di supporto (messaggi di errore chiari, validazione degli input) per ridurre il rischio di utilizzo improprio e garantire un'esperienza fluida anche in fase di apprendimento dello strumento.

1.5. Glossario

Per evitare ambiguità o fraintendimenti legati alla terminologia utilizzata nel documento, si è deciso di introdurre un glossario contenente le definizioni dei termini specifici del dominio d'uso. In questo modo, tutti i concetti rilevanti verranno chiariti e resi accessibili. I termini inclusi nel glossario sono contrassegnati nel testo con una G a pedice, in *questo modo_G*, sono anche presenti nel documento **Glossario_v2.0.0** disponibile nel sito web del gruppo  7Commits.

1.6. Riferimenti

1.6.1. Riferimenti normativi

- Norme_di_Progetto_v2.0.0:
https://7commits.github.io/7Commits/docs/PB/Interni/Norme_di_Progetto_v2.0.0.pdf
- **Capitolato C1** (ultimo accesso: 2025-08-25):
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C1.pdf>

1.6.2. Riferimenti informativi

- **T5-Analisi dei requisiti** (ultimo accesso: 2025-08-25):
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T05.pdf>
- **P2-Diagrammi dei Casi d'Uso (UML)** (ultimo accesso: 2025-08-25):
<https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20Use%20Case.pdf>
- **Large Language Models: A Survey** (ultimo accesso: 2025-08-25):
<https://arxiv.org/pdf/2402.06196>
- **Retrieval-Augmented Generation for Large Language Models: A Survey** (ultimo accesso: 2025-08-25):
<https://arxiv.org/pdf/2312.10997>
- Glossario v2.0.0:
<https://7commits.github.io/7Commits/glossary.html>

2. Descrizione generale del prodotto richiesto da Zucchetti

Il progetto, proposto da Zucchetti S.p.A., prevede la realizzazione di una piattaforma *software_G* denominata **Artificial QI**, concepita per valutare le prestazioni di sistemi basati su *Large Language Models (LLM)_G*. L'obiettivo è quello di offrire uno strumento integrato che consenta di confrontare, in modo efficace e automatizzato, le risposte fornite da un LLM con quelle attese per una determinata serie di domande.

La piattaforma dovrà permettere all'utente di:

- inserire, modificare e gestire coppie di domande e risposte attese tramite un'*interfaccia_G user-friendly_G* ;
- selezionare un sistema LLM da testare (tramite chiamate a *API_G* conformi allo standard *OpenAPI 3.1_G*) e sottoporli le domande archiviate;
- valutare automaticamente la coerenza tra le risposte ricevute dal modello e le risposte attese, tramite metodi sia tradizionali (es. *BM25_G*) sia basati su altri LLM;
- visualizzare i risultati delle esecuzioni dei *test_G* attraverso una sintesi numerica e una presentazione dei casi migliori e peggiori;
- eventualmente archiviare i risultati per confrontare performance tra diverse esecuzioni e modelli.

L'intero sistema deve essere **organico** e **coerente**, evitando la separazione in strumenti indipendenti. Tutte le funzionalità richieste devono coesistere all'interno della stessa applicazione, al fine di supportare un flusso di lavoro continuo ed efficiente per lo sviluppatore.

2.1. Obiettivo fissato

L'obiettivo del progetto consiste nello sviluppo di una piattaforma che permetta:

- di testare e confrontare in modo sistematico le risposte generate da LLM su una varietà di domande significative;
- di fornire un ambiente di validazione oggettiva per sviluppatori, consentendo loro di analizzare l'impatto delle modifiche al sistema in fase di sviluppo;
- di disporre di uno strumento affidabile e riutilizzabile per valutare miglioramenti, regressioni o effetti collaterali nel comportamento di un modello linguistico rispetto alle richieste dell'utente.

Il prodotto finale dovrà essere realizzato in più fasi:

- **Proof of Concept (*PoC_G*)**: focalizzata sull'esplorazione delle tecnologie necessarie allo sviluppo.
- **Minimum Viable Product (*MVP_G*)**: realizzazione completa delle funzionalità obbligatorie.
- **Prodotto finale**: estensione mediante requisiti opzionali e ottimizzazione generale.

2.2. Funzionalità del prodotto e requisiti

2.2.1. Caratteristiche e requisiti obbligatori:

1. **Archiviazione di una lista di domande e delle risposte attese**: sistema di raccolta persistente (*database_G*) accessibile tramite interfaccia grafica per gestione *CRUD_G*.
2. **Programma di esecuzione del test che attraverso una API ponga le domande ad un programma esterno e ne registri la risposta**: sistema automatizzato di *test_G* che sottopone le domande a un LLM via *API Rest_G* (*OpenAPI 3.1_G*) e ne registra le risposte. I risultati possono

essere utilizzati uno ad uno o registrati su un apposito archivio. Tra i requisiti opzionali è descritta una proposta per gestire in modo completo questa fase.

3. **Programma di valutazione della correttezza / verosimiglianza delle risposte ricevute:** confronto tra risposte attese e risposte ottenute attraverso metodi tradizionali (es. *BM25_G*), tecniche semantiche e/o confronto mediato da un LLM.
4. **Procedura di presentazione dei risultati dell'esecuzione del test:** visualizzazione sintetica e dettagliata dei risultati (statistiche generali, errori, confronti testuali).
5. **I punti precedenti devono essere integrati in un unico sistema che permetta di utilizzarli come parte di un insieme e non forniti come utility separate:** dovrà essere costruito un sistema di aggregazione dei risultati con un indice sintetico della bontà generale di quanto ottenuto, una evidenziazione dei casi migliori e peggiori e la possibilità di verificare tutte le risposte ottenute. Di conseguenza, tutte le funzionalità devono essere contenute in un'unica piattaforma coerente e facilmente utilizzabile.

2.2.2. Requisiti opzionali suggeriti:

- **Archiviazione dei risultati di test:** dopo una esecuzione del test è interessante poter archiviare il risultato per poi confrontarlo con altre esecuzioni. Questo richiede una gestione delle esecuzioni con la possibilità di salvare, ricercare, cancellare il documento prodotto da una esecuzione.
- **Confronto automatico tra run di test diverse:** il primo confronto sarà dato dall'indice sintetico e dalle altre valutazioni, disponendo dei dati di una esecuzione precedente può essere sviluppato un programma di confronto per evidenziare quali parti sono migliorate e quali sono peggiorate. In questo modo si ha una valutazione dei risultati che permette di capire meglio come ogni scelta nel programma che espone le API ha influenzato il risultato finale.
- **Archiviazione di set di domande/risposte:** per come sono stati proposti i requisiti obbligatori viene richiesto un unico *set di domande/risposte_G* che viene eseguito nella sua interezza. Un'interessante estensione è avere un archivio che possa gestire più set di domande/risposte per verifiche di aspetti diversi del sistema o di più di un sistema.
- **Gestione della configurazione delle chiamate alle API esterne:** i programmi espongono API secondo le loro regole, per poter testare vari sistemi è opportuno poter configurare il punto di chiamata delle API in modo da potersi adeguare a prodotti diversi.

2.3. Caratteristiche utente

L'applicazione Artificial QI è destinata a un'utenza specializzata nell'ambito dello sviluppo software e dell'integrazione di modelli di Intelligenza Artificiale, in particolare Large Language Models (*LLM_G*). Sebbene non sia progettata per un uso commerciale diretto, il sistema ha finalità sperimentali e valutative, ed è pensato per assistere gli sviluppatori nella fase di validazione delle risposte generate da modelli linguistici.

2.3.1. Utenti del nostro prodotto

Gli utenti possono essere classificati nelle seguenti categorie:

1. **Utente sviluppatore/tecnico esperto:** È l'utente primario del sistema. Si tratta di un professionista o studente con competenze in:
 - sviluppo software e architetture web;
 - tecnologie *AI_G/ML_G* e utilizzo di LLM tramite API (es. *OpenAI_G, Mistral_G*);

- linguaggi di scripting e automazione di test;
- progettazione e *analisi dei requisiti*.

Questa tipologia di utente utilizzerà la piattaforma per testare le risposte di uno o più LLM in scenari specifici, confrontare le uscite dei modelli con le risposte attese, e valutare l'effetto delle scelte progettuali.

2. **Utente aziendale/valutatore di sistema:** Un utente secondario è rappresentato da figure aziendali o accademiche interessate alla valutazione comparativa di modelli linguistici per scopi di ricerca o adozione industriale. Pur non interagendo direttamente con il sistema a livello di configurazione tecnica, questo utente potrà:

- esaminare i risultati dei test effettuati;
- visualizzare indici di performance aggregati;
- comprendere il comportamento dei modelli sotto test, anche senza conoscenze approfondite di programmazione.

L'interfaccia dell'applicazione dovrà quindi garantire un buon equilibrio tra completezza funzionale e usabilità, permettendo a entrambi i profili di fruire delle funzionalità in modo coerente e fluido.

3. **Utente LLM (entità automatica):** Infine, un "utente" non umano ma funzionale al sistema è rappresentato dal modello LLM esterno con cui l'applicativo interagisce. Il modello:

- riceve le domande tramite chiamate API;
- fornisce risposte che verranno confrontate con quelle attese;
- viene monitorato per verificarne coerenza, affidabilità e variazioni di comportamento.

Sebbene non sia un utente nel senso tradizionale, l'LLM è un attore cruciale all'interno del sistema e le sue prestazioni costituiscono l'oggetto dell'analisi svolta dalla piattaforma.

2.4. Tecnologie ed analisi della struttura di progetto

La realizzazione della piattaforma **Artificial QI** richiede l'integrazione di componenti software orientati alla valutazione delle risposte fornite da modelli linguistici esterni (LLM_G) attraverso un'interfaccia utente accessibile, strumenti di automazione e componenti server per la gestione delle esecuzioni.

Il progetto sarà sviluppato secondo una struttura modulare, con particolare attenzione alla manutenibilità e alla possibilità di estendere le funzionalità in fasi successive (PoC_G , MVP_G , prodotto finale). Si prevede l'utilizzo di tecnologie moderne per garantire portabilità, scalabilità e coerenza tra ambienti di sviluppo e produzione.

3. Casi d'uso

3.1. Obiettivi

Un *caso d'uso*_G rappresenta formalmente una funzionalità del sistema, illustrando l'interazione tra uno o più attori esterni e il sistema stesso per il raggiungimento di uno scopo definito. Citando la presentazione P2 della sezione Riferimenti informativi, un caso d'uso è definito come segue:

«Un caso d'uso è un insieme di scenari (sequenze di azioni) che hanno in comune uno scopo finale (obiettivo) per un utente (attore)».

Questa sezione ha lo scopo di descrivere in maniera dettagliata i casi d'uso identificati a seguito dell'analisi del capitolato fornito da Zucchetti S.p.A., dei colloqui col proponente e delle discussioni interne al gruppo di progetto.

I casi d'uso costituiscono la base per l'individuazione dei requisiti funzionali e guidano le successive fasi di progettazione, implementazione e validazione del sistema.

3.2. Introduzione ai casi d'uso

Ogni caso d'uso sarà descritto secondo la seguente struttura:

- **Attori coinvolti:** entità esterne che interagiscono con il sistema;
- **Descrizione:** breve spiegazione del contesto e degli obiettivi del caso d'uso;
- **Pre-condizioni:** stato del sistema necessario per avviare l'interazione;
- **Post-condizioni:** stato del sistema al termine dell'interazione;
- **Scenario principale:** sequenza tipica di azioni svolte dall'attore e dal sistema;
- **Inclusioni/Estensioni/Generalizzazioni** (se applicabili):
 - **Inclusione** = Funzionalità comune fra più use case;
 - **Estensione** = Aumento delle funzionalità di un use case;
 - **Generalizzazione** = Aggiungere o modificare caratteristiche base.

L'identificatore dei casi d'uso seguirà il prefisso UC (Use Case), seguito da un numero naturale, eventualmente seguito da numerazione gerarchica (es. UC1.2, UC1.2.1).

3.3. Attori

Nel capitolato **Artificial QI** si identificano tre attori principali:

1. Utente sviluppatore (principale):

È l'attore primario del sistema. Rappresenta la persona che utilizza l'applicazione per caricare domande e risposte attese, configurare le API dei modelli LLM da testare, avviare le sessioni di valutazione e analizzare i risultati.

Non è previsto un sistema di autenticazione, poiché l'applicativo ha finalità puramente sperimentali e si prevede un utilizzo locale o comunque controllato.

2. LLM esterno (attore secondario):

Il modello linguistico (es. *ChatGPT*_G, *Mistral*_G, *LLaMA*_G) rappresenta un attore passivo ma centrale nel processo. Riceve richieste di input in linguaggio naturale e produce una risposta da confrontare con quella attesa.

3. Sistema di valutazione automatica (attore funzionale):

Si tratta di un modulo interno che svolge una funzione critica di valutazione ma che, per coerenza modellistica, viene trattato come attore funzionale: riceve le due risposte (attesa e ottenuta) ed esegue il confronto semantico o statistico per determinare il grado di corrispondenza.

3.4. Elenco dei casi d'uso

3.4.1. UC: Gestione domande e risposte attese

3.4.1.1. UC1: Visualizzazione lista domande

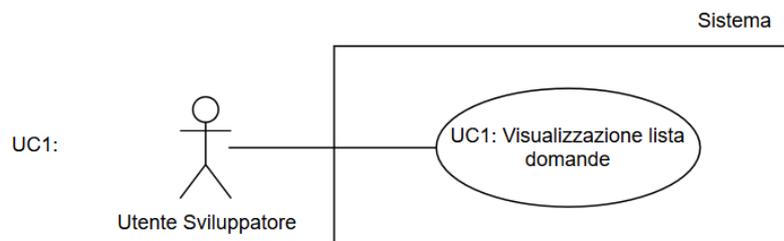


Figura 1: UC1 - Visualizzazione lista domande

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza la lista di tutte le domande presenti nel sistema.

Pre-condizioni: L'utente deve entrare nella sezione di «Gestione domande»

Post-condizioni: Viene visualizzata una lista con gli elementi domanda presenti o una lista vuota

Scenario principale:

1. L'utente entra nella sezione «Gestione domande».
2. L'utente visualizza la lista di domande presenti.

3.4.1.1.1. UC1.1: Visualizzazione singola domanda

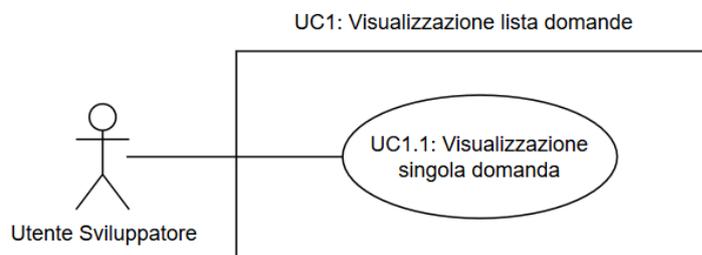


Figura 2: UC1.1 - Visualizzazione singola domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza il dettaglio di una domanda esistente, con la risposta attesa e la categoria.

Pre-condizioni: Almeno una domanda presente nel sistema.

Post-condizioni: L'utente visualizza la domanda con la risposta attesa associata e la categoria di appartenenza.

Scenario principale:

1. L'utente seleziona la domanda.
2. L'utente visualizza la domanda con la risposta attesa associata e la categoria di appartenenza.

3.4.1.2. UC2: Modifica domanda

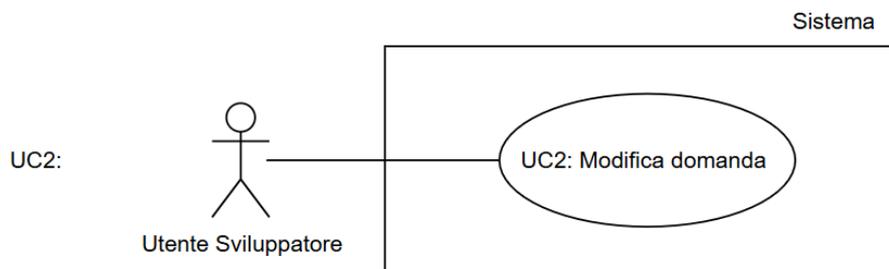


Figura 3: UC2 - Modifica domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente seleziona una domanda e ne modifica il testo salvando le modifiche.

Pre-condizioni: Almeno una domanda presente nel sistema.

Post-condizioni: Il testo della domanda viene aggiornato.

Scenario principale:

1. L'utente seleziona la domanda.
2. Modifica il testo della domanda.
3. Salva le modifiche.

3.4.1.3. UC3: Modifica risposta attesa

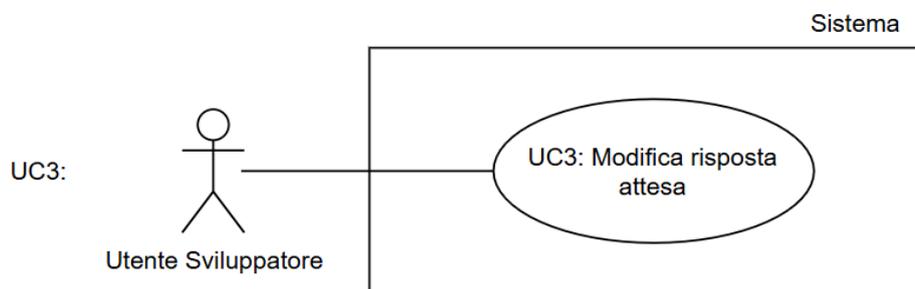


Figura 4: UC3 - Modifica risposta attesa

Attore principale: Utente sviluppatore

Descrizione: L'utente modifica il testo di una risposta attesa associata alla domanda selezionata.

Pre-condizioni: Almeno una domanda presente nel sistema.

Post-condizioni: La risposta viene aggiornata.

Scenario principale:

1. L'utente seleziona la domanda.
2. Modifica il testo della risposta attesa.
3. Salva le modifiche.

3.4.1.4. UC4: Eliminazione domanda

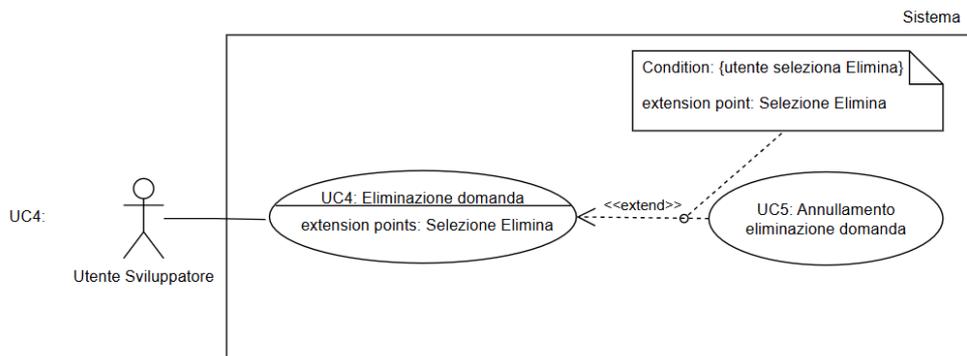


Figura 5: UC4 - Eliminazione domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente seleziona la domanda che desidera eliminare e conferma l'operazione, eliminando la domanda esistente dal sistema.

Pre-condizioni: Almeno una domanda nel sistema.

Post-condizioni: La domanda selezionata viene eliminata.

Scenario principale:

1. L'utente seleziona la domanda da eliminare.
2. Conferma l'azione.
3. Il sistema rimuove i dati.

3.4.1.5. UC5: Annullamento eliminazione domanda

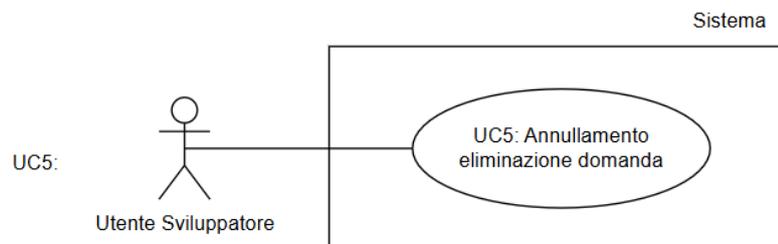


Figura 6: UC5 - Annullamento eliminazione domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente, dopo aver avviato la procedura di eliminazione di una domanda, decide di annullarla prima della conferma definitiva.

Pre-condizioni: L'utente ha selezionato l'eliminazione di una domanda ed è nella fase di conferma dell'eliminazione.

Post-condizioni: Nessuna modifica è applicata al sistema; il set rimane disponibile.

Scenario principale:

1. L'utente seleziona la domanda.
2. L'utente seleziona il pulsante di eliminazione della domanda.
3. Il sistema mostra una finestra di conferma.
4. L'utente sceglie l'opzione «Annulla» o chiude la finestra.
5. L'operazione viene annullata e la domanda non viene eliminata.

3.4.1.6. UC6: Ricerca domande per categoria

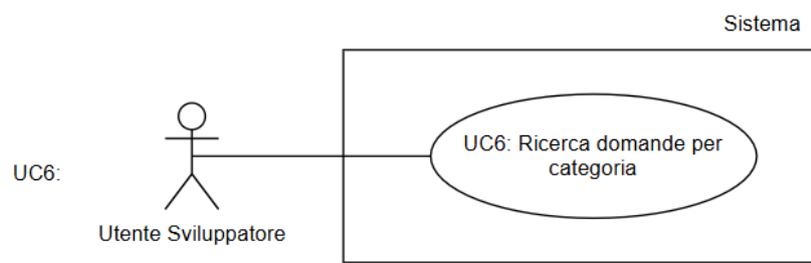


Figura 7: UC6 - Ricerca domande per Categoria di appartenenza

Attore principale: Utente sviluppatore

Descrizione: L'utente esegue una ricerca sulla lista delle domande digitando o selezionando il nome di una categoria, in modo da visualizzare solo le domande appartenenti a quella categoria.

Pre-condizioni: È presente almeno una domanda nella lista, con il campo «Categoria» non vuoto.

Post-condizioni: Il sistema mostra solo le domande selezionate.

Scenario principale:

1. L'utente apre il menu dei filtri.
2. Seleziona il nome della categoria desiderata.
3. Visualizza i risultati filtrati.

3.4.1.7. UC7: Aggiunta manuale coppia domanda/risposta

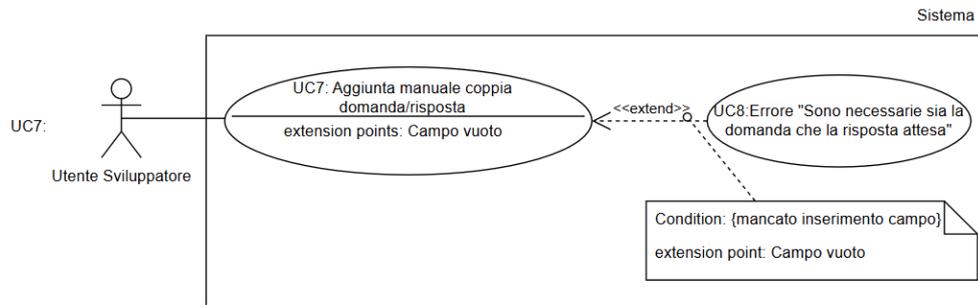


Figura 8: UC7 - Aggiunta manuale domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente aggiunge una nuova coppia domanda/risposta fornendo il testo della domanda e la risposta attesa. La categoria, se presente, può essere selezionata da un elenco predefinito ma non rappresenta un campo obbligatorio.

Pre-condizioni: L'utente ha accesso alla sezione gestione domande della webapp

Post-condizioni: Il sistema salva la nuova entità domanda/risposta, associandole eventualmente la categoria selezionata.

Scenario principale:

1. L'utente apre il modulo di inserimento.
2. Nel modulo inserisce il testo della domanda e la risposta attesa.
3. L'utente conferma l'operazione.
4. Il sistema verifica che testo e risposta siano compilati.
5. Se la validazione ha esito positivo, il sistema salva i dati.

3.4.1.8. UC8: Errore «Sono necessarie sia la domanda che la risposta attesa»

Attore principale: Utente sviluppatore

Descrizione: L'utente non inserisce manualmente una domanda o la sua risposta attesa (o entrambe) e ne conferma comunque l'inserimento.

Pre-condizioni: L'utente ha accesso alla sezione gestione domande della webapp

Post-condizioni: L'utente riceve un messaggio di errore che specifica che nell'inserimento di una domanda sono necessarie sia la domanda che la risposta attesa.

Scenario principale:

1. L'utente apre il modulo di inserimento.
2. Conferma l'inserimento senza compilare uno dei campi oppure entrambi i campi «Domanda» e «Risposta attesa».
3. L'utente riceve un messaggio di errore che specifica il motivo del mancato inserimento.

3.4.1.9. UC9: Importazione domande da file CSV

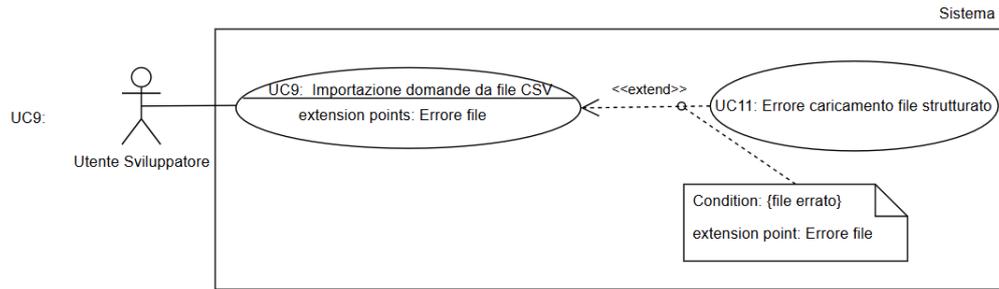


Figura 9: UC9 - Aggiunta da file strutturato

Attore principale: Utente sviluppatore

Descrizione: L'utente carica un file in formato CSV_G con colonne predefinite (question, answer, category). Il sistema verifica la correttezza delle intestazioni e importa i record.

Pre-condizioni: Il file è in formato valido CSV e rispetta le indicazioni sintattiche fornite.

Post-condizioni: Le domande sono salvate nel sistema.

Scenario principale:

1. L'utente apre la sezione "Importa domande".
2. Seleziona il file .csv.
3. Il sistema valida il formato.
4. I dati vengono salvati nel database.

3.4.1.10. UC10: Importazione domande da file JSON

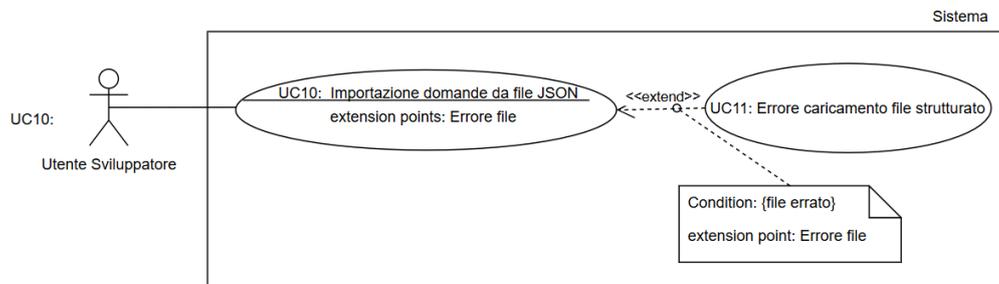


Figura 10: UC10 - Aggiunta da file strutturato

Attore principale: Utente sviluppatore

Descrizione: L'utente carica un file in formato $JSON_G$ con la sintassi corretta. Il sistema verifica la correttezza delle intestazioni e importa i record.

Pre-condizioni: Il file è in formato valido JSON e rispetta le indicazioni sintattiche fornite.

Post-condizioni: Le domande sono salvate nel sistema.

Scenario principale:

1. L'utente apre la sezione "Importa domande".

2. Seleziona il file .json.
3. Il sistema valida il formato.
4. I dati vengono salvati nel database.

3.4.1.11. UC11: Errore caricamento file strutturato

Attore principale: Utente sviluppatore

Descrizione: L'utente importa le coppie di domande e risposte da un file strutturato.

Pre-condizioni: L'utente ha accesso alla sezione gestione domande della webapp e inserisce tramite selezione del file un documento json oppure csv.

Post-condizioni: L'utente riceve un messaggio di errore che specifica che nell'inserimento di un file, la struttura del file deve essere come indicato nell'esempio fornito nella pagina di caricamento del file.

Scenario principale:

1. L'utente apre il modulo di inserimento.
2. Conferma l'inserimento del file.
3. L'utente riceve un messaggio di errore che specifica il motivo del mancato inserimento.

3.4.2. UC: Gestione dei set di domande

3.4.2.1. UC12: Visualizzazione lista dei set

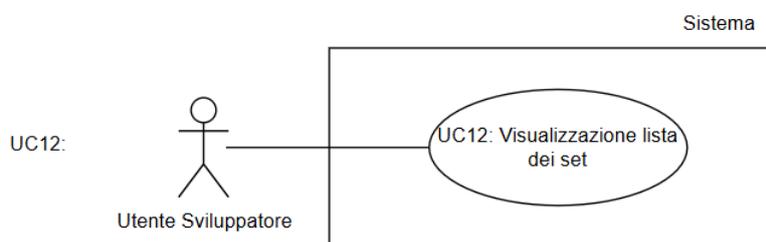


Figura 11: UC12 - Visualizzazione lista dei set

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza la lista dei set di domande salvati nel sistema per consultarli o selezionarli.

Pre-condizioni: Sono presenti uno o più set nel sistema.

Post-condizioni: I set sono visibili all'utente.

Scenario principale:

1. L'utente accede alla sezione "Gestione set".
2. Il sistema mostra l'elenco dei set disponibili.
3. L'utente può selezionare un set per ulteriori azioni (visualizzazione, modifica, eliminazione).

3.4.2.1.1. UC12.1: Visualizzazione singolo set

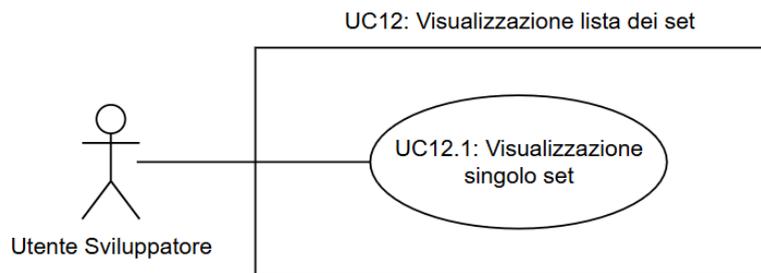


Figura 12: UC12.1 - Visualizzazione singolo set

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza i dettagli di un set selezionato, comprese le domande contenute.

Pre-condizioni: Almeno un set presente nel sistema.

Post-condizioni: L'utente visualizza le domande presenti all'interno del set selezionato.

Scenario principale:

1. L'utente seleziona il set.
2. L'utente visualizza le domande presenti all'interno del set selezionato.

3.4.2.2. UC13: Modifica set

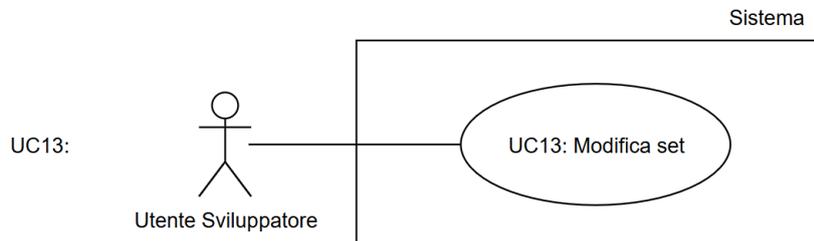


Figura 13: UC13 - Modifica set

Attore principale: Utente sviluppatore

Descrizione: L'utente modifica un set esistente cambiandone il nome e/o la composizione, rimuovendo o aggiungendo coppie di domande/risposte.

Pre-condizioni: Il set da modificare è presente e visualizzabile.

Post-condizioni: Il set viene aggiornato con il nuovo nome e/o la nuova lista di domande.

Scenario principale:

1. L'utente seleziona un set esistente.
2. Aggiunge o rimuove domande dal set.
3. Salva le modifiche.

3.4.2.3. UC14: Eliminazione set

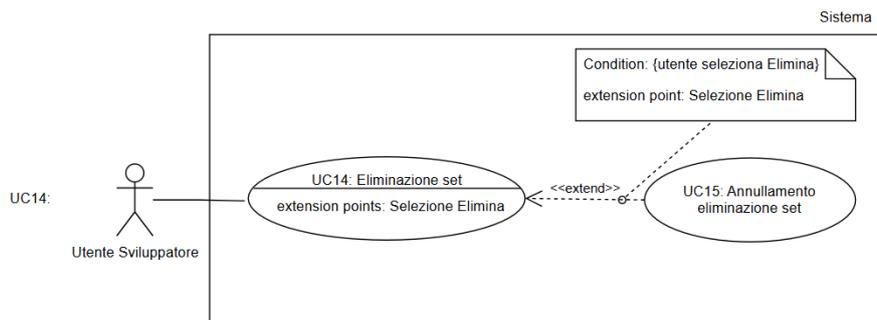


Figura 14: UC14 - Eliminazione set

Attore principale: Utente sviluppatore

Descrizione: L'utente seleziona un set da eliminare, conferma l'operazione ed elimina il set di domande dalla visualizzazione. Le domande contenute non vengono cancellate dal database.

Pre-condizioni: È presente almeno un set di domande.

Post-condizioni: Il set selezionato è rimosso dalla visualizzazione.

Scenario principale:

1. L'utente accede alla sezione "Gestione set".
2. Seleziona un set da eliminare.
3. Conferma l'eliminazione.
4. Il sistema rimuove il set, lasciando inalterate le domande.

3.4.2.4. UC15: Annullamento eliminazione set

Attore principale: Utente sviluppatore

Descrizione: L'utente, dopo aver avviato la procedura di eliminazione di un set, decide di annullarla prima della conferma definitiva.

Pre-condizioni: L'utente ha selezionato un set ed è nella fase di conferma dell'eliminazione.

Post-condizioni: Nessuna modifica è applicata al sistema; il set rimane disponibile.

Scenario principale:

1. L'utente seleziona un set da eliminare.
2. Il sistema mostra una finestra di conferma.
3. L'utente sceglie l'opzione «Annulla» o chiude la finestra.
4. L'operazione viene annullata e il set non viene eliminato.

3.4.2.5. UC16: Ricerca dei set per categoria di appartenenza

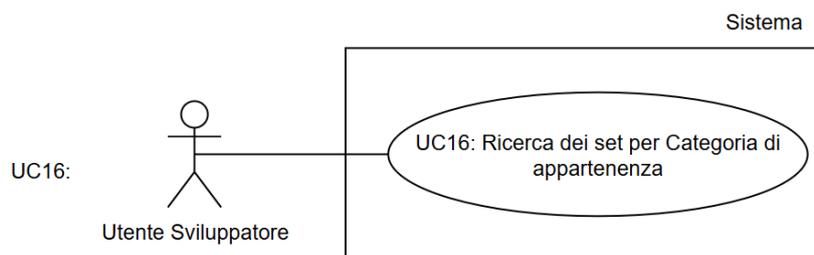


Figura 15: UC16 - Ricerca dei set per Categoria di appartenenza

Attore principale: Utente sviluppatore

Descrizione: L'utente fa la ricerca con il filtro applicato alla lista dei set per visualizzare solo quelli appartenenti ad una categoria specifica.

Pre-condizioni: È presente almeno un set nella lista.

Post-condizioni: Il sistema mostra solo i set con la categoria selezionata

Scenario principale:

1. L'utente apre il menu dei filtri.
2. Seleziona il nome della categoria desiderata.
3. Visualizza i risultati filtrati.

3.4.2.6. UC17: Aggiunta manuale set

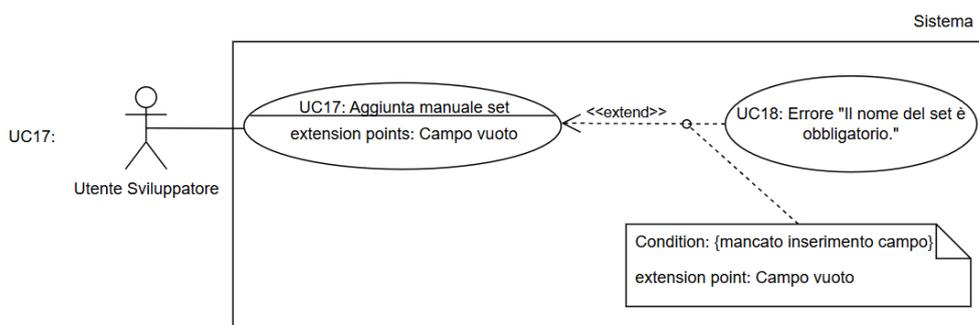


Figura 16: UC17 - Aggiunta manuale set

Attore principale: Utente sviluppatore

Descrizione: L'utente crea manualmente un set, inserendo il nome set, e aggiungendo le domande disponibili nel database al set.

Pre-condizioni: L'utente si trova nella sezione gestione set della webapp

Post-condizioni: Il nuovo set viene salvato e visualizzato correttamente.

Scenario principale:

1. L'utente apre il modulo di creazione set.

2. Aggiunge le domande desiderate al set.
3. Assegna un nome al set.
4. Preme crea set creando un nuovo set.

3.4.2.7. UC18: Errore «Il nome del set è obbligatorio.»

Attore principale: Utente sviluppatore

Descrizione: L'utente tenta di creare il set senza inserire il nome del set.

Pre-condizioni: L'utente si trova nella sezione gestione set della webapp

Post-condizioni: L'utente riceve un messaggio di errore che specifica che nell'inserimento di un nuovo set è necessario specificare il nome (NB: un set può essere vuoto, ma non può essere senza nome).

Scenario principale:

1. L'utente apre il modulo di creazione del set.
2. Conferma l'inserimento senza assegnare un nome al set.
3. L'utente riceve un messaggio di errore che specifica il motivo del mancato inserimento.

3.4.2.8. UC19: Aggiunta set da file JSON

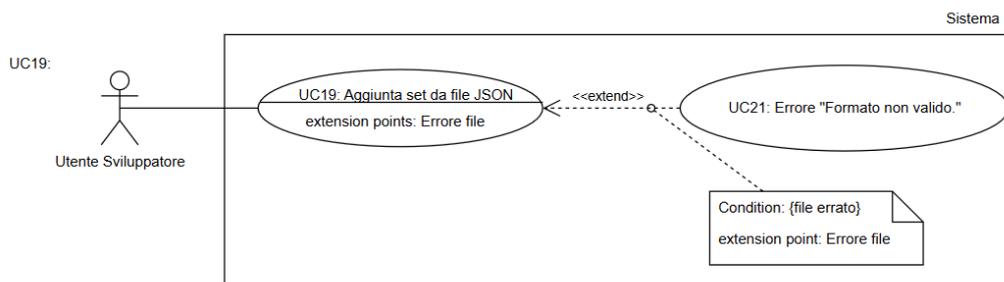


Figura 17: UC19 - Aggiunta set da file JSON

Attore principale: Utente sviluppatore

Descrizione: L'utente importa un file *JSON_G* contenente un set di domande già impostato.

Pre-condizioni: Il file è in formato JSON e rispetta le indicazioni sintattiche fornite.

Post-condizioni: Il set viene salvato nel sistema.

Scenario principale:

1. L'utente seleziona il file da caricare.
2. Il sistema valida il formato.
3. I dati vengono salvati nel database.

3.4.2.9. UC20: Aggiunta set da file CSV

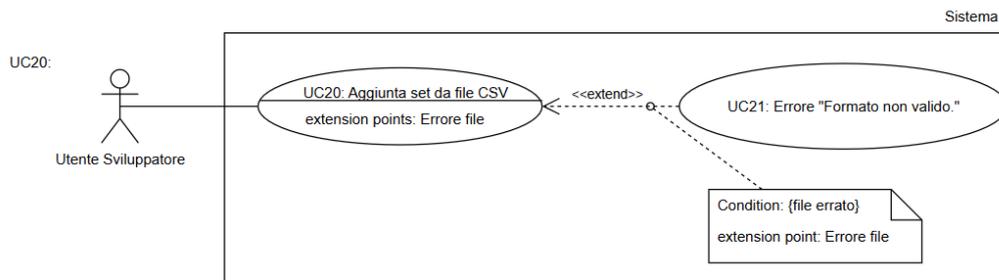


Figura 18: UC20 - Aggiunta set da file CSV

Attore principale: Utente sviluppatore

Descrizione: L'utente importa un file CSV_G contenente un set di domande già impostato.

Pre-condizioni: Il file è in formato CSV e rispetta le indicazioni sintattiche fornite.

Post-condizioni: Il set viene salvato nel sistema.

Scenario principale:

1. L'utente seleziona il file da caricare.
2. Il sistema valida il formato.
3. I dati vengono salvati nel database.

3.4.2.10. UC21: Errore «Formato non valido.»

Attore principale: Utente sviluppatore

Descrizione: L'utente importa un set da un file strutturato non rispettando la struttura descritta nella sezione di import set.

Pre-condizioni: L'utente si trova nella sezione gestione set della webapp e inserisce tramite selezione del file un documento json/csv.

Post-condizioni: L'utente riceve un messaggio di errore che specifica che nell'inserimento di un file, la struttura del file deve essere come indicato nell'esempio fornito nella pagina di caricamento del file.

Scenario principale:

1. L'utente apre il modulo di inserimento.
2. Seleziona il file da pc.
3. Preme importa set per importare i set dal file.
4. L'utente riceve un messaggio di errore che specifica il motivo del mancato inserimento.

3.4.3. UC: Configurazione API LLM

3.4.3.1. UC22: Creazione di preset per la configurazione API LLM

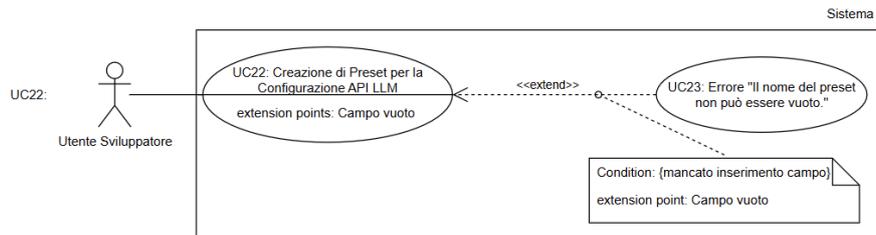


Figura 19: UC22 - Creazione di Preset per la Configurazione API LLM

Attore principale: Utente sviluppatore

Descrizione: L'utente accede al form di creazione di un preset e inserisce un nome per identificare il preset, la chiave API, il link al provider dell'endpoint, il modello utilizzato, la temperatura e i massimi token di output. L'utente può inoltre eseguire un test di connessione (UC22.1).

Pre-condizioni: L'utente conosce i dettagli dell'API del modello (es. URL_G , $token_G$, $temperatura_G$), il modello deve essere funzionante, il database deve essere impostato correttamente.

Post-condizioni: Il sistema crea un preset utilizzabile per effettuare test di valutazione tramite LLM.

Scenario principale:

1. L'utente entra nella sezione "Configurazione API" e sceglie "Crea nuovo preset".
2. Nel form digita il nome del preset, la chiave API, il link al provider dell'endpoint, il modello utilizzato, la temperatura e i massimi token di output.
3. Clicca su "Salva".
4. Il sistema verifica la presenza e la validità dei campi immessi, in assenza di errori salva il nuovo preset.

3.4.3.1.1. UC22.1: Test di connessione API

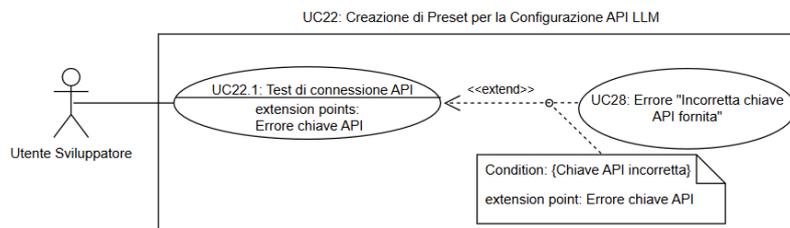


Figura 20: UC22.1 - Test di connessione API

Attore principale: Utente sviluppatore

Descrizione: L'utente al termine della compilazione dei campi richiesti per la creazione di un preset API, seleziona l'opzione di Test connessione per verificare che la connessione sia valida.

Pre-condizioni: L'utente accede alla compilazione del preset.

Post-condizioni: Il sistema restituisce una valutazione della connessione API.

Scenario principale:

1. L'utente accede alla sezione di configurazione API.
2. L'utente seleziona l'opzione di creazione nuovo preset.
3. Compila le richieste.
4. Esegue un test della connessione.
5. Ottiene una valutazione positiva o negativa del test.

3.4.3.2. UC23: Errore «Il nome del preset non può essere vuoto.»**Attore principale:** Utente sviluppatore**Descrizione:** L'utente compila un preset di configurazione API senza fornire un nome.**Pre-condizioni:** L'utente ha accesso alla sezione «Configurazione API» della webapp e non inserisce un nome al preset.**Post-condizioni:** L'utente riceve un messaggio di errore che specifica il motivo per cui il preset non è stato salvato.**Scenario principale:**

1. L'utente apre il modulo di compilazione preset.
2. Conferma l'inserimento del preset senza fornire un nome.
3. L'utente riceve un messaggio di errore che specifica il motivo del mancato inserimento.

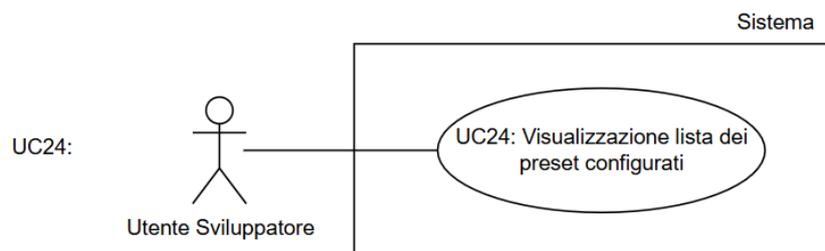
3.4.3.3. UC24: Visualizzazione lista dei preset configurati

Figura 21: UC24 - Visualizzazione lista dei preset configurati

Attore principale: Utente sviluppatore**Descrizione:** L'utente visualizza la lista dei preset di configurazione API per LLM salvati.**Pre-condizioni:** Sono presenti uno o più preset nel sistema.**Post-condizioni:** Nessuna modifica allo stato del sistema.**Scenario principale:**

1. L'utente accede alla sezione «Configurazione API».
2. Il sistema mostra l'elenco dei preset disponibili.

3.4.3.4. UC25: Modifica preset

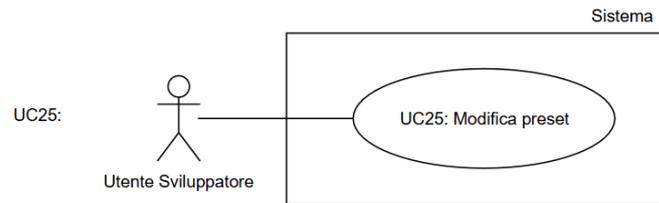


Figura 22: UC25 - Modifica preset

Attore principale: Utente sviluppatore

Descrizione: L'utente modifica uno o più parametri di un preset esistente.

Pre-condizioni: Il preset da modificare è presente e visualizzabile.

Post-condizioni: Il preset viene aggiornato con la nuova composizione.

Scenario principale:

1. L'utente entra nella sezione "Configurazione API" e sceglie «Modifica preset».
2. Nel form può modificare il nome del preset, la chiave API, il link al provider dell'endpoint, il modello utilizzato, la temperatura e i massimi token di output.
3. Clicca su "Salva".
4. Il sistema salva le modifiche.

3.4.3.5. UC26: Elimina preset

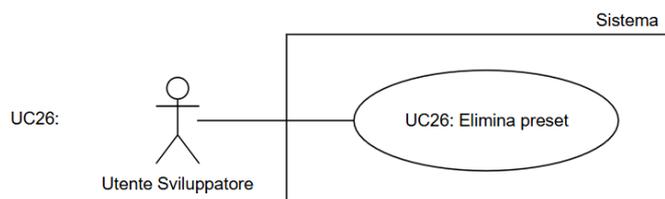


Figura 23: UC26 - Elimina preset

Attore principale: Utente sviluppatore

Descrizione: L'utente elimina un preset dal sistema. N.B. non è richiesta una conferma per l'eliminazione.

Pre-condizioni: È presente almeno un preset nel sistema.

Post-condizioni: Il preset selezionato è rimosso.

Scenario principale:

1. L'utente accede alla sezione "Configurazione API".
2. Elimina il preset.
3. Il sistema rimuove il preset.

3.4.4. UC: Avvio test

3.4.4.1. UC27: Esecuzione test

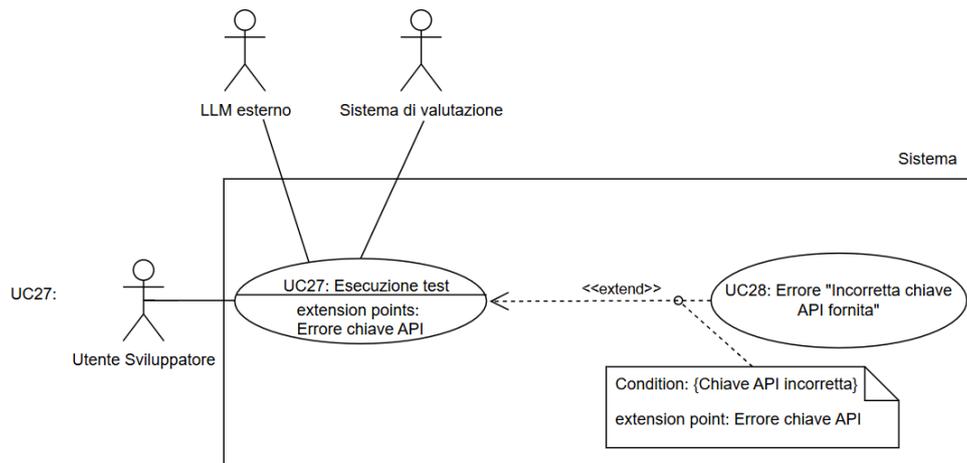


Figura 24: UC27 - Esecuzione test

Attore principale: Utente sviluppatore

Attori secondari: LLM esterno, Sistema di valutazione

Descrizione: L'utente seleziona un preset ed il set di domande da valutare; il sistema invia le domande all'LLM configurato nel preset, raccoglie le risposte e le confronta con quelle attese attraverso il sistema di valutazione automatica.

Pre-condizioni: È presente almeno una domanda con risposta attesa, è presente almeno un preset di configurazione API.

Post-condizioni: I risultati vengono salvati e mostrati.

Scenario principale:

1. L'utente seleziona un set di domande da valutare.
2. Seleziona il preset da utilizzare per la valutazione.
2. Avvia il test.
3. Il sistema invia domande all'LLM.
4. Riceve e registra le risposte.
5. Avvia la valutazione automatica.
6. Salva i risultati.

3.4.4.1.1. UC27.1: Visualizzazione preset di configurazione API salvati

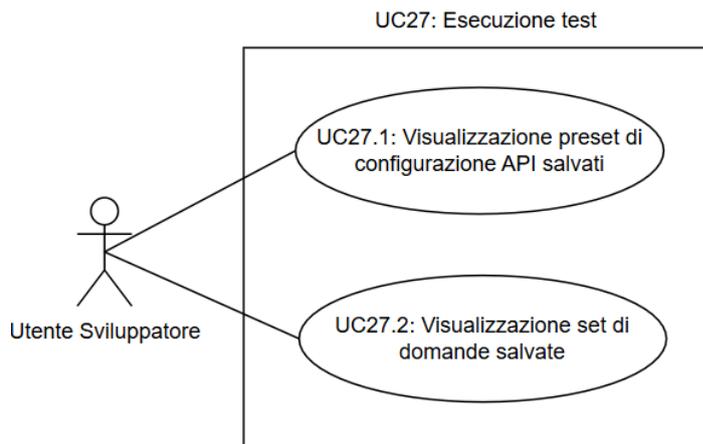


Figura 25: UC27.1 - Visualizzazione preset di configurazione API salvati

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza tutti i preset salvati all'interno della sezione di esecuzione test, in modo da poter selezionare il preset desiderato in fase di test.

Pre-condizioni: È disponibile almeno un preset.

Post-condizioni: Nessuna modifica allo stato del sistema.

Scenario principale:

1. L'utente accede alla sezione "Esecuzione test".
2. Il sistema mostra l'elenco dei preset disponibili.
3. L'utente può selezionare un preset per eseguire il test.

3.4.4.1.2. UC27.2: Visualizzazione set di domande salvate

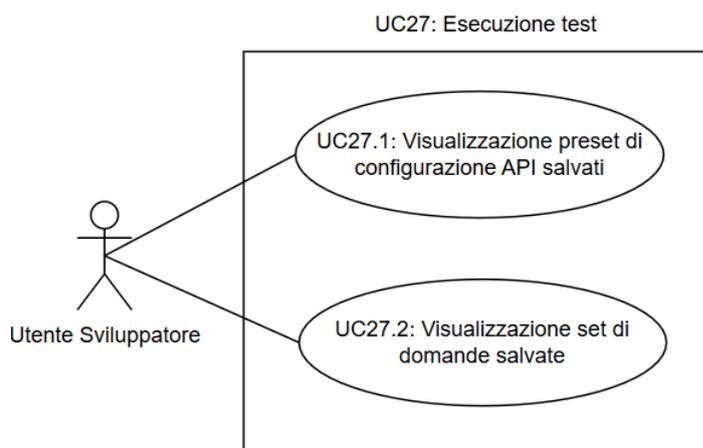


Figura 26: UC27.2 - Visualizzazione set di domande salvate

Attore principale: Utente sviluppatore

Descrizione: L'utente deve poter visualizzare tutti i set di domande salvati all'interno della sezione di esecuzione test, in modo da poter selezionare il set desiderato in fase di test.

Pre-condizioni: È stato caricato o creato almeno un set di domande.

Post-condizioni: Nessuna modifica allo stato del sistema.

Scenario principale:

1. L'utente accede alla sezione "Esecuzione test".
2. Il sistema mostra l'elenco dei set disponibili.
3. L'utente può selezionare un set su cui eseguire il test.

3.4.4.2. UC28: Errore «Incorretta chiave API fornita»

Attore principale: Utente sviluppatore

Descrizione: L'utente avvia un test tramite un preset configurato.

Pre-condizioni: L'utente ha accesso alla sezione «Esecuzione test» della webapp e avvia un test.

Post-condizioni: L'utente riceve un messaggio di errore che specifica il motivo per cui il test non è stato eseguito.

Scenario principale:

1. L'utente apre il modulo di esecuzione test.
2. Avvia l'esecuzione del test.
3. L'utente riceve un messaggio di errore che specifica il motivo della mancata esecuzione del test.

3.4.4.3. UC29: Valutazione automatica

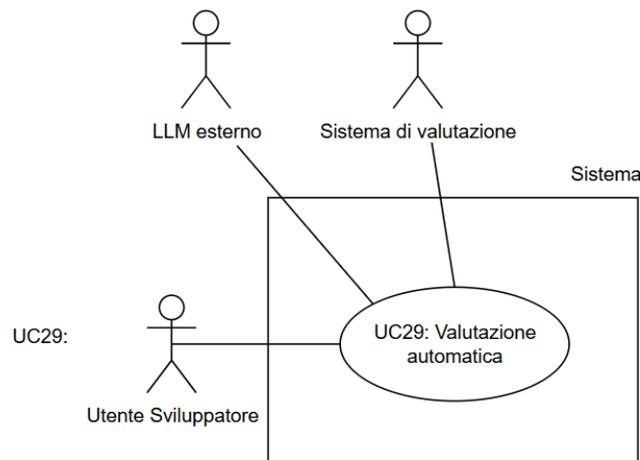


Figura 27: UC29 - Valutazione automatica

Attore principale: Sistema di valutazione automatica

Attori secondari: Utente sviluppatore, LLM esterno

Descrizione: Il sistema di valutazione automatica riceve una coppia di risposte (quella attesa e quella generata dall'LLM) e produce un punteggio o giudizio qualitativo in base a criteri semantici, lessicali o statistici.

Pre-condizioni: È disponibile una risposta attesa e una risposta generata dall'LLM.

Post-condizioni: Il risultato della valutazione è disponibile per la visualizzazione e l'analisi da parte dell'utente sviluppatore nella sezione «Visualizza risultati».

Scenario principale:

1. Il sistema riceve in input la risposta attesa e quella ottenuta.
2. Applica l'algoritmo di confronto.
3. Calcola un punteggio o una valutazione qualitativa.
4. Restituisce il risultato al sistema principale.

3.4.5. UC: Visualizzazione dei risultati

3.4.5.1. UC30: Visualizzazione risultati

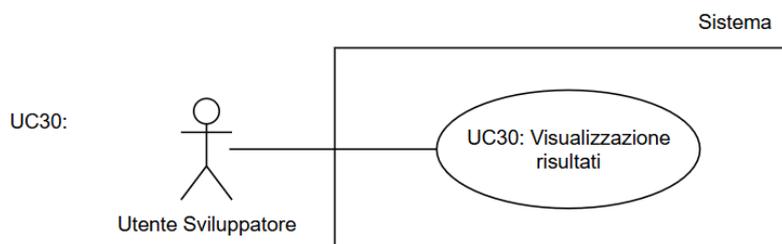


Figura 28: UC30 - Visualizzazione risultati

Attore principale: Utente sviluppatore

Descrizione: L'utente consulta i risultati ottenuti dal confronto tra risposte attese e risposte dell'LLM.

Pre-condizioni: È stata eseguita almeno una sessione di valutazione.

Post-condizioni: Nessuna modifica allo stato del sistema.

Scenario principale:

1. L'utente accede alla sezione di visualizzazione risultati.
2. Visualizza per ogni domanda la risposta attesa, la risposta dell'LLM e l'esito del confronto.

3.4.5.1.1. UC30.1: Visualizzazione risultati per singola domanda

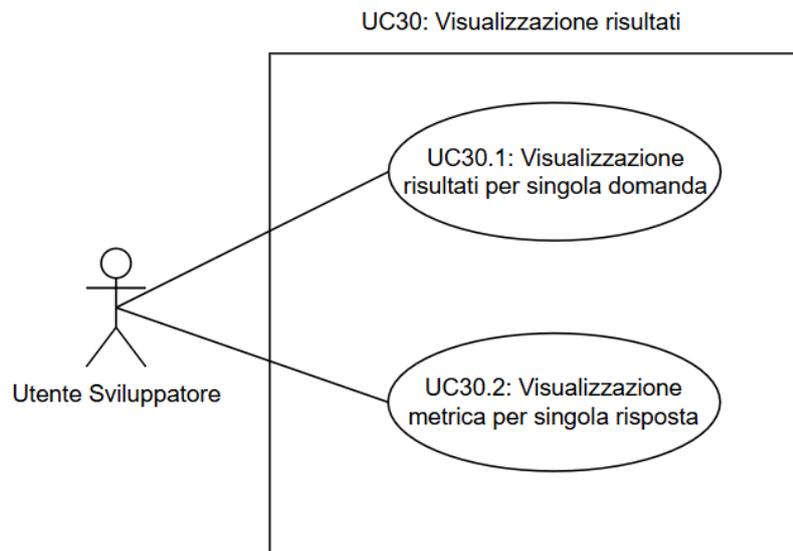


Figura 29: UC30.1 - Visualizzazione risultati per singola domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente accede al dettaglio del risultato relativo a una specifica domanda, osservando risposta generata, attesa e valutazione associata.

Pre-condizioni: È stato eseguito almeno un test.

Post-condizioni: Nessuna modifica allo stato.

Scenario principale:

1. L'utente entra nella sezione risultati.
2. Seleziona una domanda.
3. Visualizza risposta attesa, risposta dell'LLM e valutazione.

3.4.5.1.2. UC30.2: Visualizzazione metrica per singola risposta

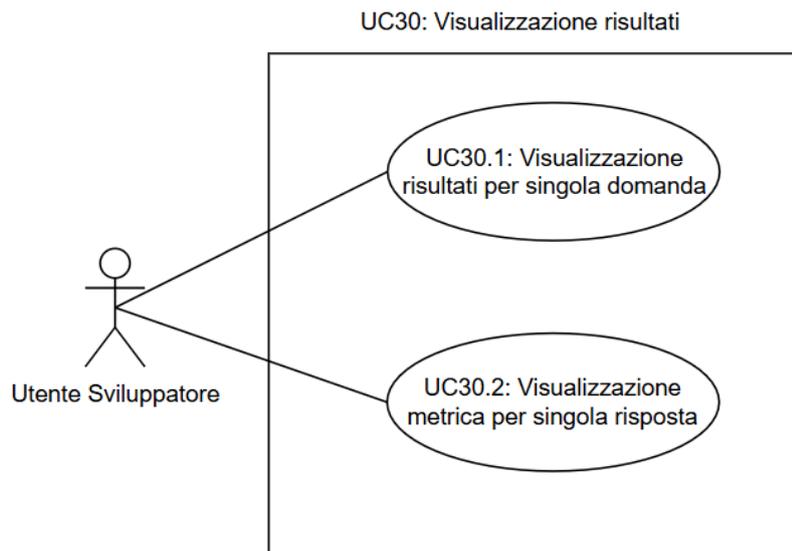


Figura 30: UC30.2 - Visualizzazione metrica per singola risposta

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza le singole metriche applicate (Somiglianza, Correttezza e Completezza) a una risposta.

Pre-condizioni: La valutazione è stata eseguita.

Post-condizioni: Nessuna modifica allo stato.

Scenario principale:

1. L'utente accede al dettaglio di una domanda.
2. Visualizza ogni metrica applicata (Somiglianza, Correttezza e Completezza) e il relativo punteggio.

3.4.5.2. UC31: Salvataggio risultati del test

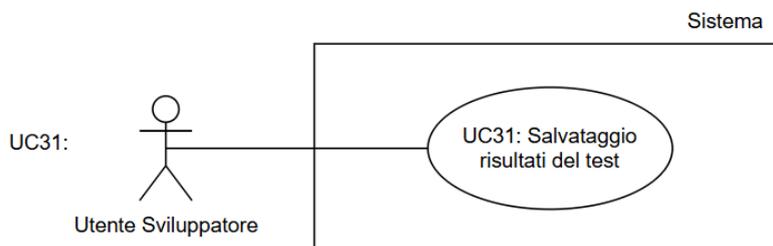


Figura 31: UC31 - Salvataggio risultati del test

Attore principale: Sistema

Attori secondari: Utente sviluppatore

Descrizione: Al termine di un'esecuzione, il sistema salva i risultati del test (risposte, valutazioni e metadati) per consentirne la consultazione successiva o l'esportazione.

Pre-condizioni: È stato eseguito un test su uno o più set.

Post-condizioni: I risultati vengono salvati nel sistema e associati a un identificativo univoco di esecuzione.

Scenario principale:

1. Il sistema riceve l'output del confronto tra risposte attese e risposte generate.
2. Per ogni domanda, associa: input, risposta LLM, valutazione e metadati.
3. Registra i dati nel database/storico.
4. Rende disponibili i risultati per UC30 (visualizzazione) e UC35 (storico).

3.4.5.3. UC32: Download risultati del test

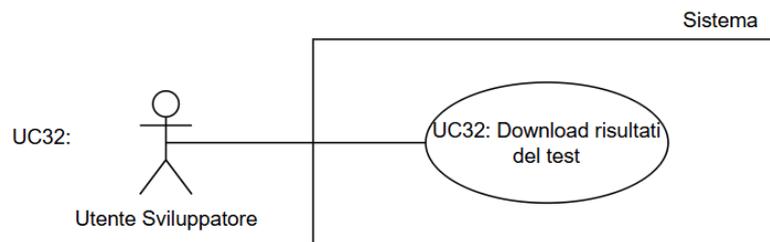


Figura 32: UC32 - Download risultati del test

Attore principale: Utente sviluppatore

Descrizione: L'utente scarica i risultati in un file strutturato contenente domande, risposte, valutazioni, metriche e metadati.

Pre-condizioni: È stato eseguito almeno un test.

Post-condizioni: Il file è scaricato localmente.

Scenario principale:

1. L'utente seleziona il test da esportare.
2. Sceglie il nome del file.
3. Avvia il salvataggio.

3.4.5.4. UC33: Caricamento risultati da file

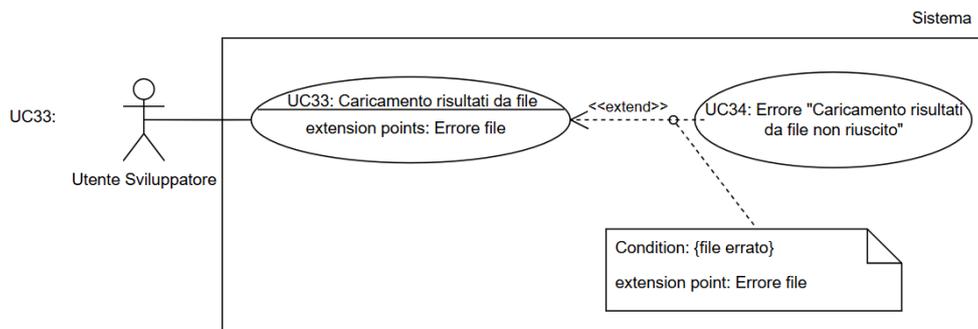


Figura 33: UC33 - Caricamento risultati da file

Attore principale: Utente sviluppatore

Descrizione: L'utente importa nel sistema i risultati di un test precedentemente salvato.

Pre-condizioni: Il file caricato è valido.

Post-condizioni: I risultati vengono aggiunti allo storico.

Scenario principale:

1. L'utente seleziona il file.
2. Il sistema lo analizza.
3. I dati vengono importati e associati.

3.4.5.5. UC34: Errore «Caricamento risultati da file non riuscito»

Attore principale: Utente sviluppatore

Descrizione: L'utente viene notificato della presenza di un errore dato da un formato errato del file

Pre-condizioni: L'utente carica un file contenente dei dati in formato errato.

Post-condizioni: L'utente riceve un messaggio di errore che indica che il file è errato.

Scenario principale:

1. L'utente seleziona il file.
2. Il sistema lo analizza.
3. L'utente riceve un errore a causa del formato errato dei file.

3.4.5.6. UC35: Visualizzazione storico dei test

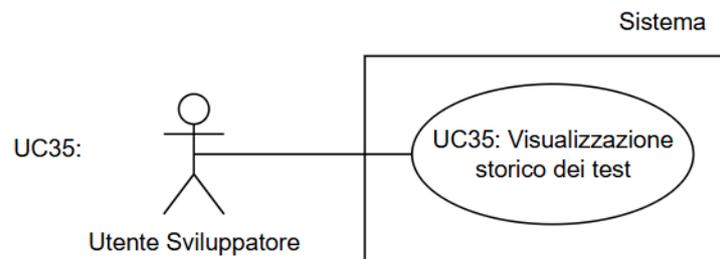


Figura 34: UC35 - Visualizzazione storico dei test

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza lo storico delle esecuzioni dei test precedentemente effettuati per poterli consultare e confrontare.

Pre-condizioni: È stato effettuato almeno un test.

Post-condizioni: L'elenco delle esecuzioni è visibile.

Scenario principale:

1. L'utente visualizza lo storico dei test.
2. Il sistema mostra una lista cronologica o filtrabile di tutte le esecuzioni passate.
3. L'utente può selezionare un'esecuzione per visualizzarne i dettagli (UC35.1) o confrontarla (UC37).

3.4.5.6.1. UC35.1: Visualizzazione elemento esecuzione test

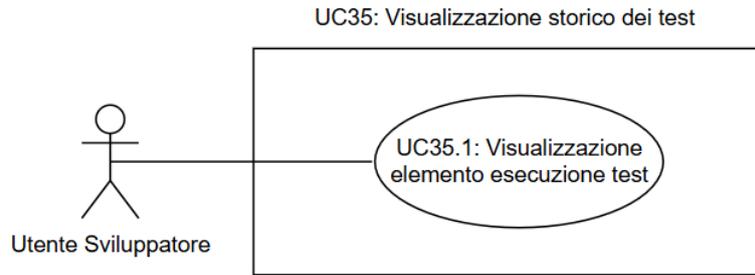


Figura 35: UC35.1 - Visualizzazione elemento esecuzione test

Attore principale: Utente sviluppatore

Descrizione: L'utente visualizza tutti i dati relativi a una specifica esecuzione, come set utilizzato, LLM testato, punteggio generale e risultati delle singole domande.

Pre-condizioni: Lo storico dei test è disponibile.

Post-condizioni: Nessuna modifica allo stato del sistema.

Scenario principale:

1. L'utente seleziona un test dallo storico.
2. Il sistema mostra:
 - Nome del set;
 - Modello LLM utilizzato;
 - Valutazione media;
 - Lista di tutte le domande, risposte e valutazioni;
3. L'utente può accedere al dettaglio di ogni domanda (vedi UC30.1/UC30.2).

3.4.5.7. UC36: Ricerca con applicazione del filtro su risultati test

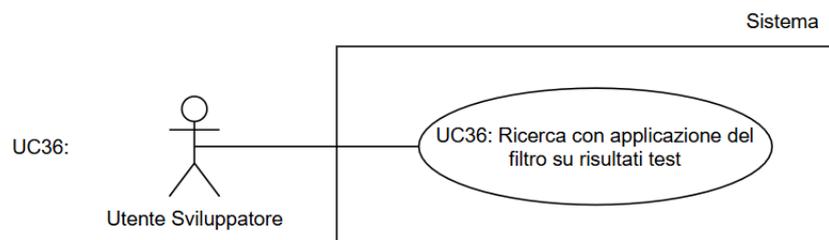


Figura 36: UC36 - Applicazione filtri storico dei test

Attore principale: Utente sviluppatore

Descrizione: L'utente effettua la ricerca su risultati test applicando il filtro dei test in base a criteri come nome del set utilizzato o modello LLM impiegato.

Pre-condizioni: Sono presenti almeno un'esecuzione testate nel sistema.

Post-condizioni: Il sistema visualizza solo i test che soddisfano i filtri applicati.

Scenario principale:

1. L'utente accede allo storico.
2. Seleziona uno o più criteri di filtro.
3. Il sistema aggiorna la visualizzazione.

Generalizzazioni:

- **UC36.1:** Ricerca storico dei test per nome del set;
- **UC36.2:** Ricerca storico dei test per LLM;
- **UC36.3:** Ricerca storico dei test per risultati di valutazione;

3.4.5.7.1. UC36.1: Ricerca storico dei test per nome del set

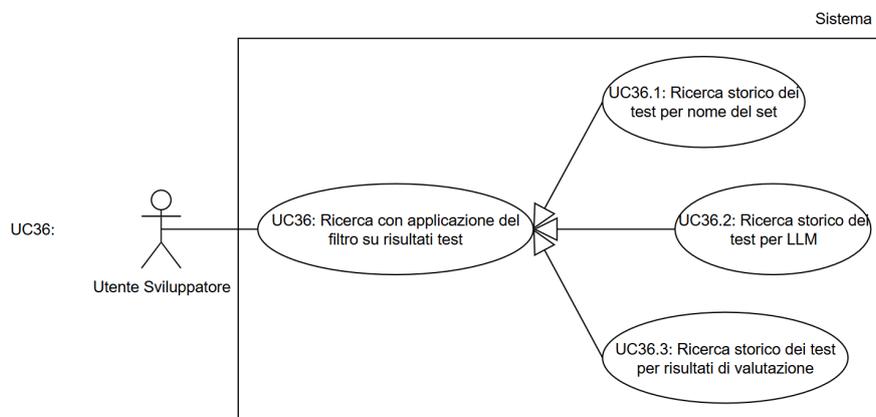


Figura 37: UC36.1 - Filtro storico dei test per nome del set

Attore principale: Utente sviluppatore

Descrizione: L'utente ricerca lo storico dei test per visualizzare solo quelli eseguiti su un determinato set di domande.

Pre-condizioni: Sono presenti test associati a set diversi.

Post-condizioni: Il sistema mostra solo i test del set selezionato.

Scenario principale:

1. L'utente apre il menu dei filtri.
2. Seleziona il nome del set.
3. Visualizza i risultati filtrati.

3.4.5.7.2. UC36.2: Ricerca storico dei test per LLM

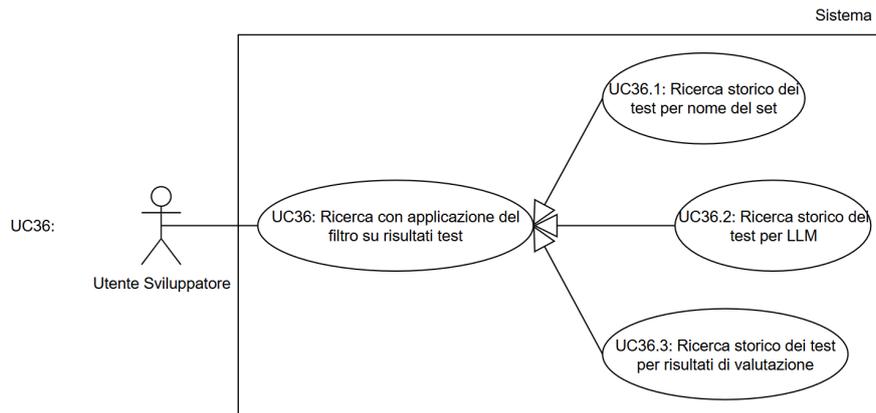


Figura 38: UC36.2 - Filtro storico dei test per LLM

Attore principale: Utente sviluppatore

Descrizione: L'utente ricerca lo storico per visualizzare solo i test eseguiti con uno specifico modello LLM.

Pre-condizioni: Sono presenti test eseguiti con più LLM.

Post-condizioni: Il sistema mostra solo i test eseguiti con il modello selezionato.

Scenario principale:

1. L'utente apre il menu dei filtri.
2. Seleziona il nome del LLM.
3. Visualizza i test associati.

3.4.5.7.3. UC36.3: Ricerca storico dei test per risultati di valutazione

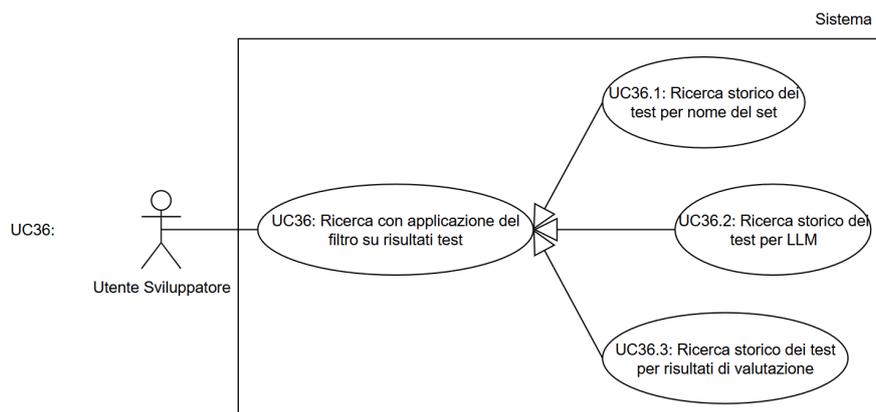


Figura 39: UC36.3 - Filtro storico dei test per risultati di valutazione

Attore principale: Utente sviluppatore

Descrizione: L'utente ordina i risultati di un test per individuare i peggiori o migliori in base alla valutazione.

Pre-condizioni: Esiste almeno un risultato disponibile.

Post-condizioni: I risultati sono visualizzati in ordine scelto.

Scenario principale:

1. L'utente accede ai risultati di un test.
2. Seleziona il criterio di ordinamento (es. punteggio crescente).
3. Visualizza l'elenco ordinato.

3.4.5.8. UC37: Confronto tra esecuzioni diverse di test

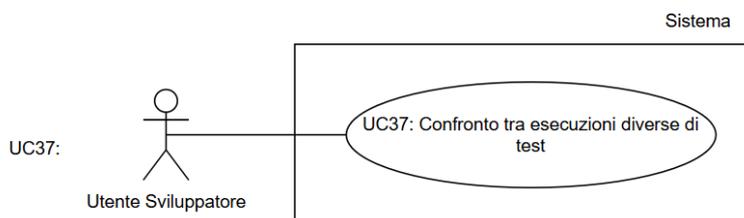


Figura 40: UC37 - Confronto tra esecuzioni diverse di test

Attore principale: Utente sviluppatore

Descrizione: L'utente seleziona due esecuzioni differenti per confrontarne i risultati, al fine di valutarne differenze e variazioni.

Pre-condizioni: Sono registrate almeno due esecuzioni precedenti di test.

Post-condizioni: Il sistema genera un confronto visualizzabile.

Scenario principale:

1. L'utente accede allo storico.
2. Seleziona due esecuzioni da confrontare.
3. Il sistema avvia il confronto ed elabora le differenze.
4. Viene mostrato il risultato (UC38).

3.4.5.9. UC38: Visualizzazione risultato confronto esecuzioni

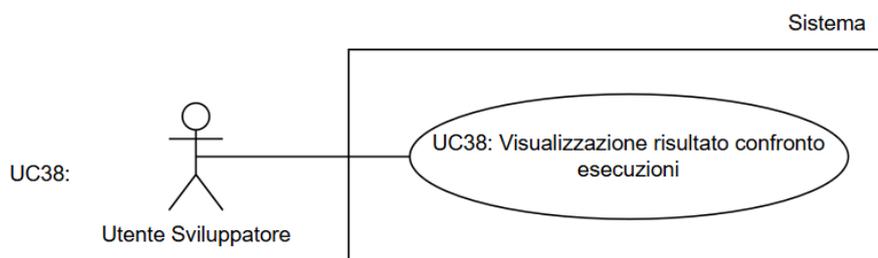


Figura 41: UC38 - Visualizzazione risultato confronto esecuzioni

Attore principale: Utente sviluppatore

Descrizione: Il sistema mostra il risultato del confronto tra due test, mostrando i valori in percentuale ottenuti in termini di Somiglianza, Correttezza e Completezza delle risposte ricevute per entrambi i set.

Pre-condizioni: Sono disponibili almeno due test validi nel sistema e l'utente ha selezionato due esecuzioni confrontabili tra loro.

Post-condizioni: Il confronto è mostrato all'utente.

Scenario principale:

1. L'utente seleziona due test per il confronto.
2. Il sistema confronta domande comuni tra i due test e ne valuta i risultati a confronto.
3. Per ogni domanda visualizza:
 - Risposte generate nei due test;
 - Differenza nei punteggi;
 - Somiglianza tra risposta attesa e ricevuta in percentuale;
 - Correttezza della risposta ricevuta in percentuale;
 - Completezza della risposta ricevuta in percentuale;
4. L'utente può accedere al dettaglio completo tramite UC38.1.

3.4.5.9.1. UC38.1: Visualizzazione elemento confronto esecuzione

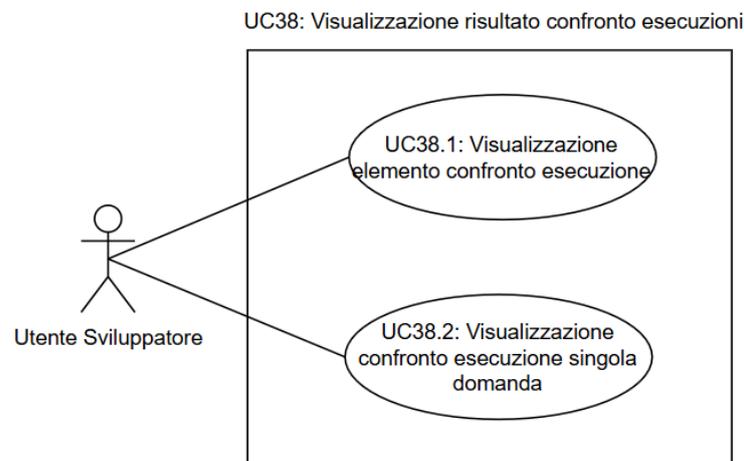


Figura 42: UC38.1 - Visualizzazione elemento confronto esecuzione

Attore principale: Utente sviluppatore

Descrizione: L'utente osserva il confronto tra due risposte LLM per una singola domanda specifica, incluso il dettaglio delle metriche.

Pre-condizioni: Il confronto tra test è attivo (UC37).

Post-condizioni: Nessuna modifica allo stato.

Scenario principale:

1. L'utente seleziona una domanda tra quelle comuni.

2. Il sistema mostra:

- Risposta dell'LLM A;
- Risposta dell'LLM B;
- Valutazioni e differenze;
- Valori delle metriche;

3.4.5.9.2. UC38.2: Visualizzazione confronto esecuzione singola domanda

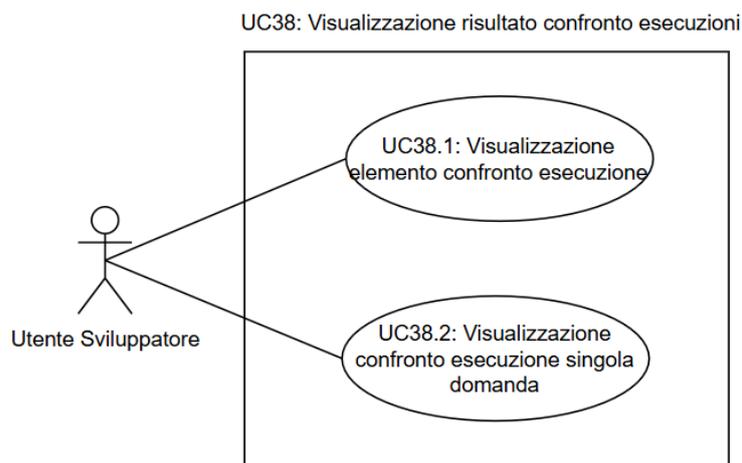


Figura 43: UC38.2 - Visualizzazione confronto esecuzione singola domanda

Attore principale: Utente sviluppatore

Descrizione: L'utente esplora nel dettaglio l'evoluzione di una singola domanda su più esecuzioni, utile per tracciare il comportamento nel tempo.

Pre-condizioni: La domanda è stata usata in più test.

Post-condizioni: Viene mostrato l'andamento storico della valutazione per la domanda.

Scenario principale:

1. L'utente cerca o seleziona una domanda.
2. Il sistema recupera tutti i test in cui è stata usata.
3. Visualizza in ordine cronologico:
 - Modello LLM utilizzato;
 - Risposta generata;
 - Valutazione e metrica;
 - Eventuali variazioni di contesto;

4. Requisiti

Ogni *requisito_G* analizzato sarà identificato univocamente da una sigla del tipo **R[Tipo][Priorità][Codice]** nella quale:

- **R** sta per Requisito;
- **[Tipo]** può essere:
 - **F** se Funzionale;
 - **Q** se di Qualità;
 - **V** se di Vincolo.
- **[Priorità]** può essere:
 - **O** per Obbligatorio;
 - **D** per Desiderabile;
 - **P** per Opzionale.
- **[Codice]**: identifica univocamente i requisiti per ogni tipologia. È un numero intero progressivo univoco assegnato in ordine di importanza se il *requisito_G* non ha padre, se invece si tratta di un sotto-requisito segue il formato **[Codice padre].[Numero figlio]** e trattandosi di una struttura ricorsiva non c'è limite alla profondità della gerarchia. I codici sono numerati in base alla sezione e alla classificazione (Es: RFO01 = Requisito Funzionale Obbligatorio 1, RQO01 = Requisito di Qualità Obbligatorio 1, RVO01 = Requisito di Vincolo Obbligatorio 1).

4.1. Requisiti di funzionalità

Codice	Descrizione	Classificazione	Fonti
RFO01	L'utente deve poter visualizzare la lista delle domande	Obbligatorio	UC1
RFO02	L'utente deve poter visualizzare una singola domanda	Obbligatorio	UC1.1
RFO03	L'utente deve poter modificare una singola domanda	Obbligatorio	UC2
RFO04	L'utente deve poter modificare una risposta attesa	Obbligatorio	UC3
RFO05	L'utente deve poter eliminare una singola domanda	Obbligatorio	UC4
RFD06	L'utente vuole poter annullare l'operazione di eliminazione di una singola domanda	Desiderabile	UC5
RFD07	L'utente deve poter filtrare le domande in base alla categoria di appartenenza	Desiderabile	UC6
RFO08	L'utente deve essere avvisato nel caso in cui l'aggiunta manuale della domanda non sia corretta e/o completa	Obbligatorio	UC8
RFO09	L'utente deve poter aggiungere manualmente una domanda	Obbligatorio	UC7
RFO10	L'utente deve poter aggiungere domande e relative risposte attese tramite un file CSV	Obbligatorio	UC9
RFO11	L'utente deve poter aggiungere domande e relative risposte attese tramite un file JSON	Obbligatorio	UC10
RFO12	L'utente vuole poter essere notificato nel caso in cui il file di importazione delle domande non sia corretto	Obbligatorio	UC11
RFO13	L'utente deve poter visualizzare la lista dei set di domande salvati nel sistema	Obbligatorio	UC12
RFO14	L'utente deve poter visualizzare un singolo set di domande	Obbligatorio	UC12.1
RFO15	L'utente deve poter modificare un set di domande	Obbligatorio	UC13
RFO16	L'utente deve poter eliminare un set di domande	Obbligatorio	UC14
RFD17	L'utente vuole poter annullare l'operazione di eliminazione di un set di domande	Desiderabile	UC15

RFD18	L'utente deve poter filtrare i set di domande in base alla categoria di appartenenza	Desiderabile	UC16
RFP19	L'utente deve poter aggiornare manualmente un set di domande	Opzionale	UC17
RFO20	L'utente vuole poter essere notificato nel caso in cui non inserisca correttamente il nome del set di domande	Obbligatorio	UC18
RFO21	L'utente deve poter aggiungere un set di domande da file JSON	Obbligatorio	UC19
RFO22	L'utente deve poter aggiungere un set di domande da file CSV	Obbligatorio	UC20
RFO23	L'utente vuole poter essere notificato nel caso in cui il file contenente il set da aggiungere non sia valido	Obbligatorio	UC21
RFO24	L'utente deve poter accedere alla sezione di configurazione dell'API e creare un preset	Obbligatorio	UC22
RFO25	L'utente deve poter essere notificato nel caso in cui il nome del preset non sia stato inserito	Obbligatorio	UC23
RFP26	L'utente deve poter effettuare un test di connessione dopo aver compilato i campi necessari per la creazione di un preset API	Opzionale	UC22.1
RFP27	L'utente deve poter essere notificato nel caso in cui l'esito del test sia negativo	Opzionale	UC23
RFO28	L'utente deve poter visualizzare la lista dei preset che sono stati configurati	Obbligatorio	UC24
RFO29	L'utente deve poter modificare un preset precedentemente configurato	Obbligatorio	UC25
RFO30	L'utente deve poter eliminare un preset precedentemente configurato	Obbligatorio	UC26
RFO31	Il sistema deve poter inviare domande a un LLM esterno tramite API e riceverne le risposte	Obbligatorio	UC27 Capitolato Zucchetti
RFO32	L'utente deve poter visualizzare tutti i preset salvati all'interno della sezione di esecuzione test	Obbligatorio	UC27.1
RFO33	L'utente deve poter visualizzare tutti i set di domande salvati all'interno della sezione di esecuzione test	Obbligatorio	UC27.2

RFO34	L'utente deve poter visualizzare il punteggio o il giudizio qualitativo assegnato dal sistema in seguito al confronto tra la risposta attesa e la risposta generata dall'LLM	Obbligatorio	UC29 Capitolato Zucchetti
RFO35	L'utente deve poter visualizzare i risultati ottenuti dal confronto tra risposte attese e risposte dell'LLM	Obbligatorio	UC29 Capitolato Zucchetti
RFO36	L'utente deve poter visualizzare i risultati ottenuti dal confronto tra risposte attese e risposte dell'LLM relativi ad una singola domanda	Obbligatorio	UC30.1
RFO37	L'utente deve poter visualizzare le metriche applicate ad una singola risposta	Obbligatorio	UC30.2
RFO38	L'utente deve poter salvare e consultare i risultati del test (risposte, valutazioni e metadati)	Obbligatorio	UC31 Capitolato Zucchetti
RFP39	L'utente deve poter scaricare i risultati di un test	Opzionale	UC32 Capitolato Zucchetti
RFP40	L'utente deve poter caricare nel sistema un file di risultati di test effettuati precedentemente	Opzionale	UC33 Capitolato Zucchetti
RFO41	L'utente vuole poter essere notificato nel caso in cui il caricamento del file contenente i risultati non vada a buon fine	Obbligatorio	UC34
RFO42	L'utente deve poter visualizzare i test effettuati precedentemente	Obbligatorio	UC35 Capitolato Zucchetti
RFP43	L'utente deve poter visualizzare tutti i dettagli di un determinato test effettuati precedentemente	Opzionale	UC35.1
RFD44	L'utente deve poter filtrare lo storico dei test in base a diversi criteri	Desiderabile	UC36, UC36.1, UC36.2, UC36.3
RFP45	L'utente deve poter confrontare due diverse esecuzioni di test	Opzionale	UC37, UC38 Capitolato Zucchetti
RFP46	L'utente deve poter confrontare due diverse risposte LLM per una singola domanda specifica	Opzionale	UC38.1, UC38.2

4.2. Requisiti di qualità

Codice	Descrizione	Classificazione	Fonti
RQO01	Deve essere redatto il documento Analisi dei requisiti comprensivo di diagrammi UML dei Casi d'Uso.	Obbligatorio	Decisione Interna
RQO02	Deve essere consegnata la documentazione relativa ai casi di test gestiti e le relative reportistiche, consultabile all'interno del documento Piano di Qualifica in una sezione dedicata.	Obbligatorio	Capitolato Zucchetti
RQO03	È necessario rispettare le convenzioni definite nel documento <i>Norme di Progetto v1.0.0</i> e seguire le regole imposte sullo stesso per l'intero periodo di svolgimento del progetto.	Obbligatorio	Decisione Interna
RQO04	Deve essere fornito il manuale per l'utilizzo dell'applicativo.	Obbligatorio	Capitolato Zucchetti
RQO05	Deve essere fornito il manuale per gli sviluppatori che intendono estendere l'applicativo.	Obbligatorio	Capitolato Zucchetti
RQO06	Il sistema deve poter essere avviato tramite una singola operazione.	Obbligatorio	Decisione interna
RQO07	Occorre realizzare il codice sorgente con un sistema di versionamento.	Obbligatorio	Decisione interna

4.3. Requisiti di vincolo

Codice	Descrizione	Classificazione	Fonti
RVO01	Il sistema deve essere sviluppato come Web Application accessibile da browser Chrome (da versione 137 in poi), Microsoft Edge (da versione 137 in poi), Mozilla Firefox (da versione 139 in poi), Apple Safari (da versione 16 in poi)	Obbligatorio	Decisione Interna
RVO02	La comunicazione tra il ciclo di esecuzione che legge domande e risposte e il sistema esterno deve avvenire tramite API rest secondo lo standard OpenAPI 3.1	Obbligatorio	Capitolato Zucchetti
RVD03	Il sistema può essere distribuito tramite un ambiente containerizzato, come Docker.	Desiderabile	Capitolato Zucchetti
RVO04	Le coppie domande/risposte inserite nel sistema devono poter essere formulate in linguaggio naturale.	Obbligatorio	Capitolato Zucchetti
RVO05	L'importazione delle coppie domande/risposte all'interno del sistema deve poter avvenire tramite file strutturato, JSON o CSV	Obbligatorio	Decisione Interna

5. Tracciamento dei requisiti

Fonte	Requisito
UC1 e sottocasi	RFO01, RFO02
UC2	RFO03
UC3	RFO04
UC4	RFO05
UC5	RFD06
UC6	RFD07
UC7	RFO09
UC8	RFO08
UC9	RFO10
UC10	RFO11
UC11	RFO12
UC12 e sottocasi	RFO13, RFO14
UC13	RFO15
UC14	RFO16,
UC15	RFD17
UC16	RFD18,
UC17	RFP19
UC18	RFO20
UC19	RFO21
UC20	RFO22, RFO34
UC21	RFO23
UC22 e sottocasi	RFO24, RFP26
UC23	RFO25, RFP27
UC24	RFO28
UC25	RFO29
UC26	RFO30
UC27 e sottocasi	RFO31, RFO32, RFO33
UC29	RFO34, RFO35
UC30 e sottocasi	RFO36, RFO37
UC31	RFO38
UC32	RFP39
UC33	RFP40
UC34	RFO41
UC35 e sottocasi	RFO42, RFP43
UC36 e sottocasi	RFD44
UC37	RFP45
UC38 e sottocasi	RFP45, RFP46
Capitolato Zucchetti	RFO31, RFO34, RFO35, RFO38, RFP39, RFP40, RFO42, RFP45, RVO02, RVD03, RVO04, RQO02, RQO04, RQO05

Decisione Interna	RVO01, RVO05, RQO01, RQO03, RQO06, RQO07
-------------------	--

5.1. Riepilogo

Tipologia	Obbligatorio	Opzionale	Desiderabile	Totale
Funzionali	33	8	5	46
Qualità	7	0	0	7
Vincolo	4	0	1	5
				58

5.2. Conclusioni

I requisiti del prodotto potranno essere modificati nel corso del suo ciclo di vita per introdurre miglioramenti o aggiornamenti. Qualora siano disponibili risorse sufficienti, il gruppo valuterà l'inserimento di nuovi requisiti volti a incrementare il valore e la qualità del prodotto. Eventuali aggiornamenti saranno quindi presi in considerazione nel tempo, in un'ottica di evoluzione continua.