

Test Report

NICOLA TORTORA, GASPARE GALASSO

June 2025

1 Test Summary

Il processo di verifica è stato articolato in tre fasi principali:

1.1 Prima esecuzione

In una prima fase è stata eseguita l'intera test suite costituita da test unitari, di integrazione e di sistema. Su un totale di **72 test**, **58 sono risultati superati** mentre **14 hanno fallito**, evidenziando diverse anomalie. Le principali criticità rilevate riguardavano:

- la mancanza di gestione delle eccezioni in presenza di input errati o risorse non disponibili;
- la presenza di messaggi di log inattesi nei test di sistema;
- mancata gestione di input vuoti o errati

1.2 Correzione e miglioramento

A seguito dell'analisi delle anomalie, sono state introdotte le seguenti modifiche:

- aggiornamento dei test di sistema per validare correttamente i messaggi prodotti dal programma principale (`main.py`);
- revisione della gestione degli errori nei moduli `retrieval_engine`, `embedding_model` e `vuln_analysis`;
- validazione preventiva degli input prima di entrare nella pipeline;

1.3 Seconda esecuzione

Dopo l'applicazione delle correzioni, è stata eseguita nuovamente l'intera suite di test. Questa seconda esecuzione ha portato a risultati positivi: **tutti i test sono stati superati con successo**.

Il processo di testing ha dunque permesso di rilevare malfunzionamenti preesistenti e di migliorare la qualità del software esistente prima dell'introduzione di modifiche evolutive.

2 Anomalie riscontrate durante i test

Durante la prima esecuzione della test suite, sono stati rilevati 14 test falliti su un totale di 72. Di seguito si riporta una tabella riepilogativa delle anomalie:

Test Case	Test Item	Motivo potenziale
test_get_vuln_analysis_empty_code	TI-03	Analisi su codice vuoto non solleva eccezione
test_encode_exceeds_max_tokens	TI-04	Nessuna eccezione sollevata oltre il limite token
test_get_similar_contracts_http_error	TI-05	L'errore HTTP non viene catturato correttamente
test_generate_invalid_apikey	TI-06	Eccezione <code>openai.AuthenticationError</code> non catturata correttamente
test_generate_invalid_modelname	TI-06	<code>APIConnectionError</code> non catturato correttamente
test_generate_invalid_url	TI-06	Connessione fallita verso URL errato
test_generate_long_prompt (OpenAI)	TI-06	Errore prompt troppo lungo non viene gestito.
test_generate_long_prompt (HFLLM)	TI-07	Errore prompt troppo lungo non viene gestito.
test_invalid_collection_name	TI-08	Eccezione non gestita correttamente
test_retrieval_empty_code	TI-09	L'engine ritorna contratti anche con input vuoto.
test_code_analysis_llm_failure	TI-11	L'eccezione del LLM non viene sollevata come previsto
test_empty_contract	SYS	Il main non rileva che il codice è vuoto e continua l'analisi.
test_invalid_contract	SYS	Se il codice ha sintassi errata o appartiene ad un linguaggio diverso, il sistema continua comunque l'analisi.

Table 1: Test case che hanno fallito e relative anomalie

3 Copertura del Codice

La misurazione della code coverage è stata condotta per verificare il livello di esercizio del codice sorgente durante l'esecuzione della test suite. L'obiettivo è garantire che tutti i rami logici significativi del sistema siano stati testati, minimizzando il rischio di malfunzionamenti in aree non testate.

Per questa analisi è stato utilizzato lo strumento `coverage.py`, integrato con il framework `pytest`.

```
coverage run --branch --source=src/ -m pytest tests/ -v
coverage html
```

Il report è stato generato in formato HTML ed è disponibile per l'analisi visuale aprendo il file `docs/test/coverage/index.html` presente nel repository.