



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

Corso di

Ingegneria, Gestione ed Evoluzione del Software



Test Report

DOCENTE

Prof. Andrea De Lucia

LINK REPOSITORY - BRANCH "DEV"

CADOCS: [https://github.com/](https://github.com/alfcan/CADOCS/tree/dev)

[alfcan/CADOCS/tree/dev](https://github.com/alfcan/CADOCS/tree/dev)

CADOCS_NLU: [https://github.com/](https://github.com/alfcan/CADOCS_NLU_Model/tree/dev)

[alfcan/CADOCS_NLU_Model/tree/dev](https://github.com/alfcan/CADOCS_NLU_Model/tree/dev)

csDetector: [https://github.com/](https://github.com/alfcan/csDetector/tree/dev)

[alfcan/csDetector/tree/dev](https://github.com/alfcan/csDetector/tree/dev)

AUTORI

Alfonso Cannavale

Matricola: 0522501597

Davide La Gamba

Matricola: 0522501464

Kevin Pacifico

Matricola: 0522501501

Anno Accademico 2022-2023

Indice

1	Pre-Maintenance Tests	1
1.1	Test di unità e di integrazione	1
1.2	Test di sistema	1
1.3	Test di regressione	5
2	Post-Maintenance Tests	6
2.1	CR_1: Refactoring	6
2.1.1	Test di unità e di integrazione	6
2.1.2	Test di sistema	7
2.1.3	Conclusioni	12
2.2	CR_2: Italian Conversation	13
2.2.1	Test di unità e di integrazione	13
2.2.2	Test di sistema	13
2.2.3	Input Testing	20
2.2.4	Conclusioni	20
2.3	CR_3: Docker Porting csDetector	21
2.3.1	Test di unità e di integrazione	21
2.3.2	Test di sistema	21
2.3.3	Conclusioni	27

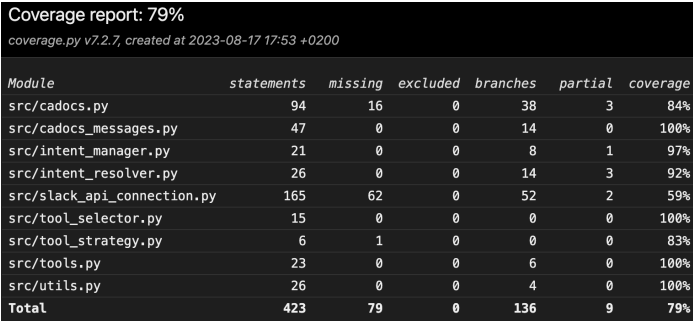
Capitolo 1

Pre-Maintenance Tests

1.1 Test di unità e di integrazione

In Figura 1.1 sono riportati i risultati in termini di branch coverage ottenuti dai test di unità e di integrazione implementati. Il requisito di copertura di branch richiesto è del 75%. Sono state utilizzate le librerie Python: *pytest* e *unittest*.

La test suite realizzata è composta da 102 test case.



Coverage report: 79%
coverage.py v7.2.7, created at 2023-08-17 17:53 +0200

Module	statements	missing	excluded	branches	partial	coverage
src/cadocs.py	94	16	0	38	3	84%
src/cadocs_messages.py	47	0	0	14	0	100%
src/intent_manager.py	21	0	0	8	1	97%
src/intent_resolver.py	26	0	0	14	3	92%
src/slack_api_connection.py	165	62	0	52	2	59%
src/tool_selector.py	15	0	0	0	0	100%
src/tool_strategy.py	6	1	0	0	0	83%
src/tools.py	23	0	0	6	0	100%
src/utls.py	26	0	0	4	0	100%
Total	423	79	0	136	9	79%

Figura 1.1: Pre-Maintenance branch coverage

1.2 Test di sistema

Gli identificativi dei test case utilizzati sono riferiti al documento di Test Case Specification. In fase di realizzazione dei test di sistema, abbiamo tentato di utilizzare il tool Selenium IDE, come stabilito nel Test Plan. Tuttavia, questo si è però rilevato poco efficiente e pratico da utilizzare a causa della natura delle funzionalità da te-

stare. Infatti, le richieste sono singoli messaggi digitati in una chat Slack e Selenium registra l'attesa della risposta del chatbot, causando un'impossibilità dell'automatizzazione dei test di sistema. Per questo motivo è stato deciso di eseguire questi test manualmente e di raccogliere i risultati ottenuti.

get_smells

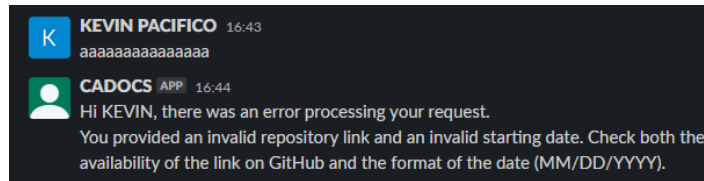


Figura 1.2: TC_GS_1_ENG

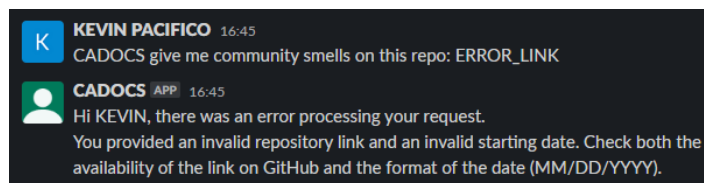


Figura 1.3: TC_GS_2_ENG

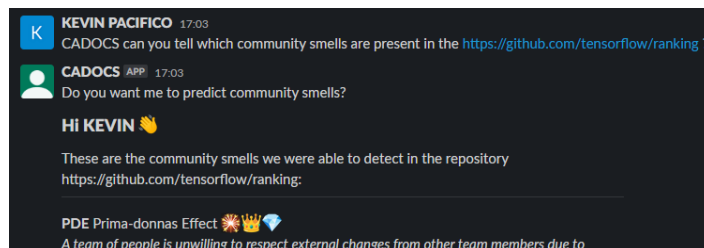


Figura 1.4: TC_GS_3_ENG

get_smells_date

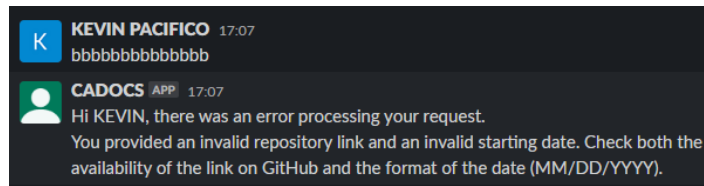


Figura 1.5: TC_GSD_1_ENG

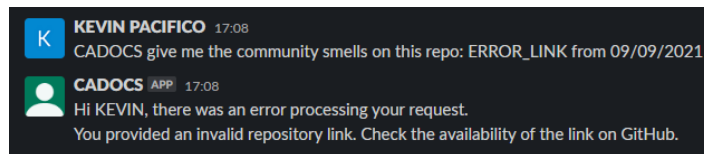


Figura 1.6: TC_GSD_2_ENG

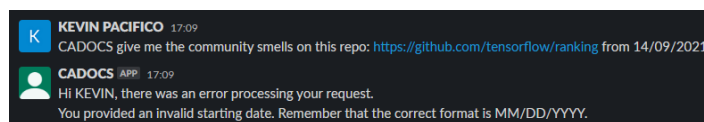


Figura 1.7: TC_GSD_3_ENG

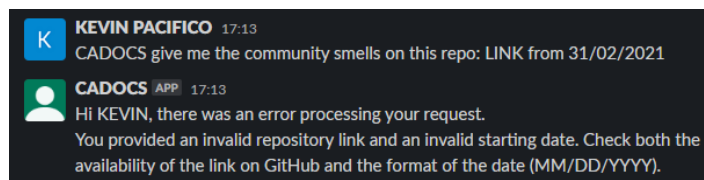


Figura 1.8: TC_GSD_4_ENG

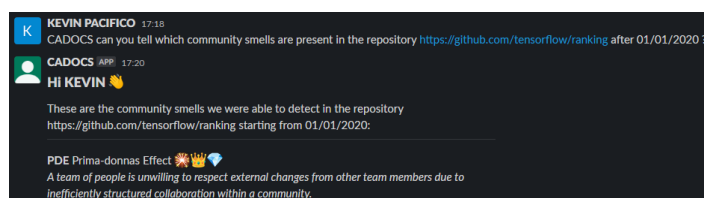


Figura 1.9: TC_GSD_5_ENG

report

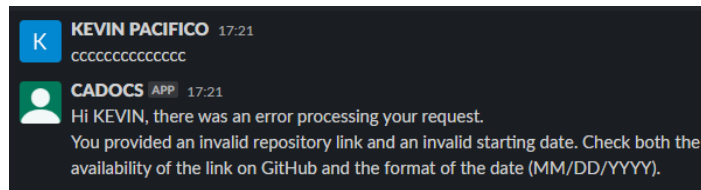


Figura 1.10: TC_RP_1_ENG

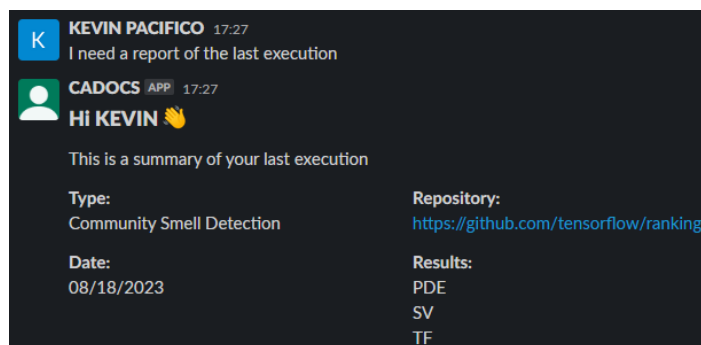


Figura 1.11: TC_RP_2_ENG

info

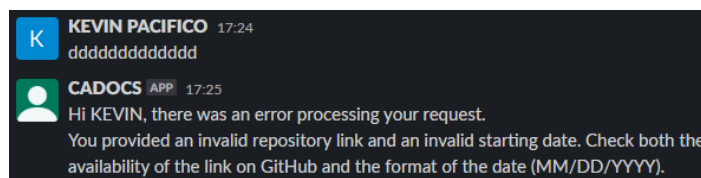


Figura 1.12: TC_IN_1_ENG

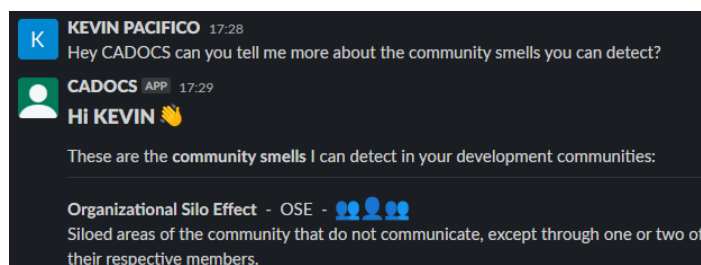


Figura 1.13: TC_IN_2_ENG

L'output dei test case TC_GS_1_ENG, TC_GSD_1_ENG, TC_RP_1_ENG e TC_IN_1_ENG è considerato conforme all'oracolo in quanto la richiesta viene classificata con un intent errato. Di conseguenza questi test vengono considerati passati. Da evidenziare il comportamento del TC_GSD_1_ENG, nel quale il modello interpreta correttamente l'intent della categoria a cui appartiene; in questo caso è ritenuto comunque passato perchè viene restituito un messaggio di errore per la mancanza della data.

Possiamo concludere che tutti i test di sistema eseguiti sono considerati passati.

1.3 Test di regressione

Per l'implementazione del test di regressione, è stata realizzata una GitHub Action che esegue i test di unità e di integrazione creati e verifica la loro corretta esecuzione. Inoltre, considerando possibili miglioramenti e sviluppi futuri, è stato implementato un controllo sulla copertura dei branch dei test. Questo controllo assicura che la copertura dei branch sia mantenuta al di sopra del 75%. Nella Tabella 1.1 è mostrato un riassunto del test report della fase di Pre-Maintenance.

	Branch coverage	Totale TC	Totale passati	Percentuale passati
Test di unità e integrazione	79%	102	102	100%
Test di sistema		12	12	100%

Tabella 1.1: Tabella riassuntiva test report pre-maintenance

Capitolo 2

Post-Maintenance Tests

2.1 CR_1: Refactoring

2.1.1 Test di unità e di integrazione

Le modifiche di refactoring hanno avuto un impatto sulla nostra test suite, che di conseguenza è stata modificata per adattarsi alla change request. Una volta terminate le modifiche abbiamo eseguito i test di unità ed integrazione realizzati con pytest in fase di pre-maintenance. La nuova test suite è composta da 98 test rispetto ai 102 test case della test suite iniziale, in quanto quattro test case sono stati rimossi in fase di refactoring perchè testavano più volte gli stessi metodi. Come possiamo notare dalla figura 2.1 la branch coverage totale è ancora del 79%.

Coverage report: 79%						
coverage.py v7.3.0, created at 2023-08-27 13:06 +0200						
Module	statements	missing	excluded	branches	partial	coverage
src/__init__.py	0	0	0	0	0	100%
src/api/__init__.py	0	0	0	0	0	100%
src/api/slack_api_connection.py	168	62	0	52	2	59%
src/chatbot/__init__.py	0	0	0	0	0	100%
src/chatbot/cadocs_slack.py	71	0	0	34	2	98%
src/chatbot/cadocs_utils.py	24	16	0	4	1	32%
src/chatbot/intent_manager.py	21	0	0	8	1	97%
src/intent_handling/__init__.py	0	0	0	0	0	100%
src/intent_handling/cadocs_intents.py	6	0	0	0	0	100%
src/intent_handling/intent_resolver.py	14	0	0	8	2	91%
src/intent_handling/tool_selector.py	15	0	0	0	0	100%
src/intent_handling/tool_strategy.py	6	1	0	0	0	83%
src/intent_handling/tools.py	23	0	0	6	0	100%
src/service/__init__.py	0	0	0	0	0	100%
src/service/cadocs_messages.py	59	0	0	20	1	99%
src/service/utils.py	22	0	0	4	0	100%
Total	429	79	0	136	9	79%

Figura 2.1: Branch coverage dopo la CR_1

2.1.2 Test di sistema

Di seguito sono riportati gli esiti dei Test di Sistema specificati nel documento Test Case Specification.

In particolare, è stato necessario considerare accettato un diverso formato del link e delle date rispetto a quelle indicate nel documento Test Plan, in modo da riflettere le modifiche apportate.

I test di sistema sono stati eseguiti prima di effettuare il merge del branch *CR1* (dove sono presenti i cambiamenti apportati) nel branch *dev* per verificare la qualità delle modifiche effettuate. In particolare, la modifica riguardante l'accettazione anche di date contenenti "." e "-" come separatori è stata testata in questa fase, per cui viene riportato l'esito ottenuto. Tali test non saranno però ripetuti in futuro poiché non facenti parte della TCS. In particolare, da questa modifica, anche le date espresse con "." e "-" fanno parte della categoria "formato data corretto".

get_smells

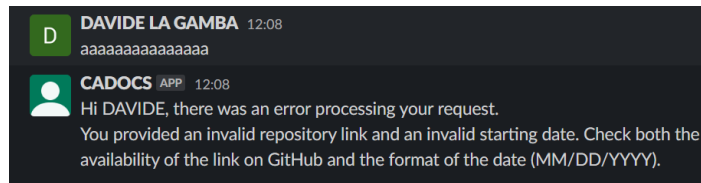


Figura 2.2: TC_GS_1_ENG

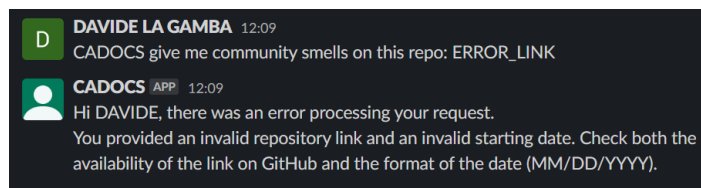


Figura 2.3: TC_GS_2_ENG

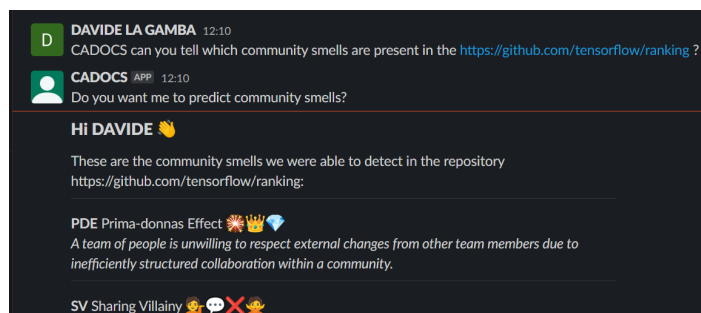


Figura 2.4: TC_GS_3_ENG

get_smells_date

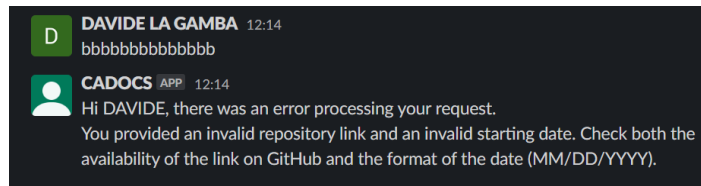


Figura 2.5: TC_GSD_1_ENG

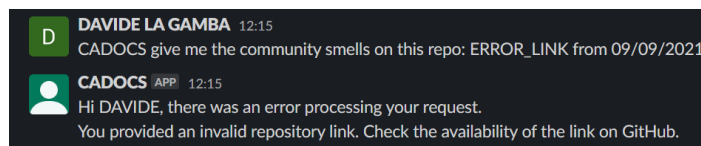


Figura 2.6: TC_GSD_2_ENG

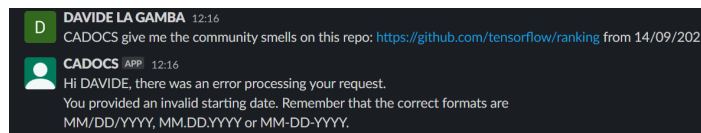


Figura 2.7: TC_GSD_3_ENG

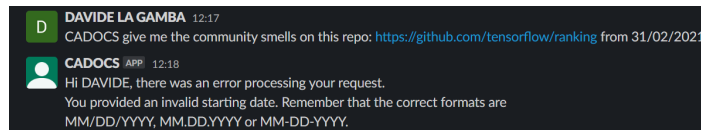


Figura 2.8: TC_GSD_4_ENG

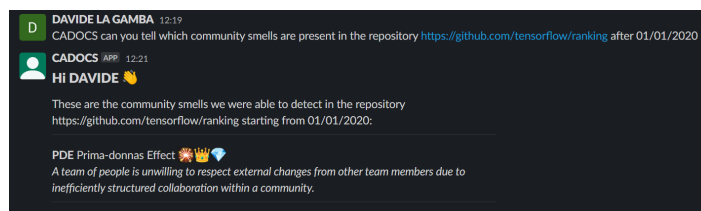


Figura 2.9: TC_GSD_5_ENG

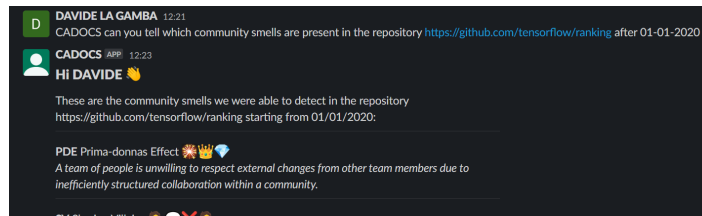


Figura 2.10: TC_GSD_5_ENG con data avente "-" come separatore

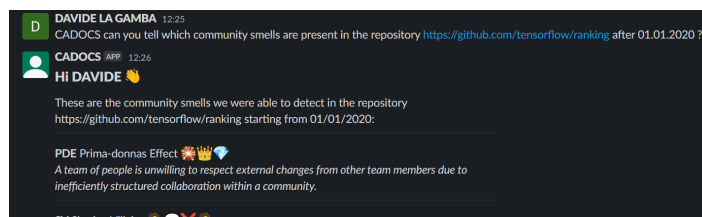
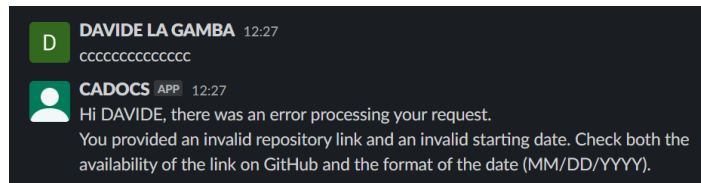
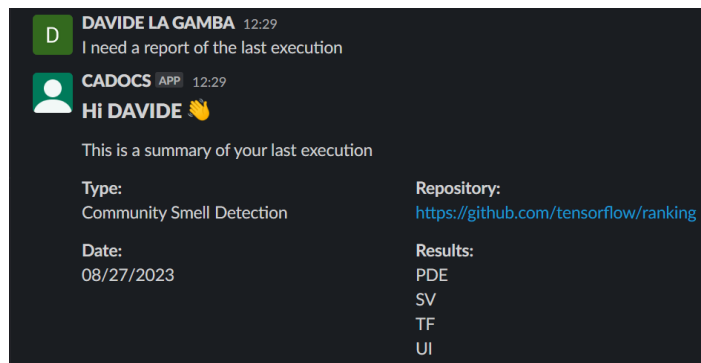
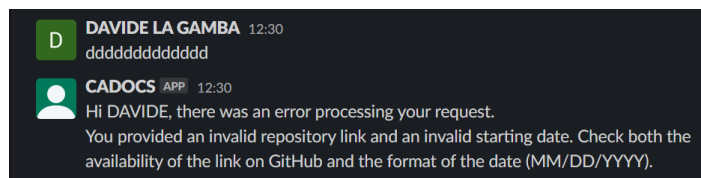


Figura 2.11: TC_GSD_5_ENG con data avente "." come separatore

report**Figura 2.12:** TC_RP_1_ENG**Figura 2.13:** TC_RP_2_ENG**info****Figura 2.14:** TC_IN_1_ENG

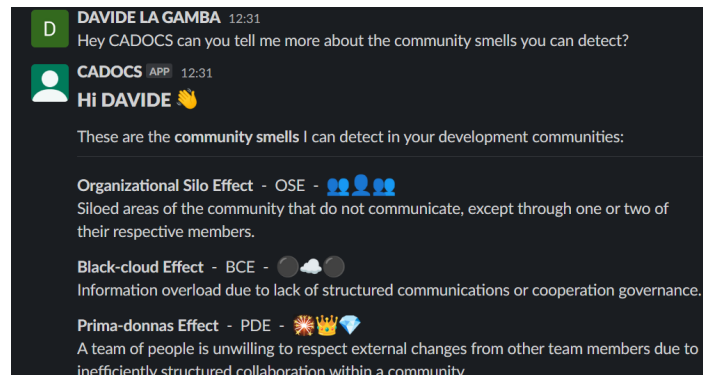


Figura 2.15: TC_IN_2_ENG

Per i motivi riportati nella Sezione 1.2, tutti i test di sistema sono considerati passati.

2.1.3 Conclusioni

Nella tabella 2.1 è mostrato un riassunto dei test eseguiti dopo l'implementazione della change request 1. In particolare sono riportati i risultati relativi all'esecuzione dei test di unità ed integrazione della test suite ed i test di sistema.

	Branch coverage	Totale TC	Totale passati	Percentuale passati
Test di unità e integrazione	79%	98	98	100%
Test di sistema		14	14	100%

Tabella 2.1: Test report post-maintenance della CR_1

2.2 CR_2: Italian Conversation

2.2.1 Test di unità e di integrazione

Le modifiche per l'integrazione della lingua italiana all'interno del sistema hanno avuto un impatto sulla nostra test suite, che di conseguenza è stata modificata con l'aggiunta di nuovi test per testare le funzionalità aggiunte all'interno delle varie componenti. Inoltre, sono stati rimossi alcuni test, in quanto alcune delle funzionalità fornite dal vecchio modello NLU erano diventate inutilizzate. In particolare, oltre ai 98 test case della test suite realizzata per la CR_1, sono stati aggiunti 16 test case per verificare il codice sorgente aggiunto nelle componenti per le nuove funzionalità. La test suite finale è composta da 114 test case. Come possiamo notare dalla figura 2.16 la branch coverage totale è aumentata al 86%.

Coverage report: 86%

coverage.py v7.3.0, created at 2023-09-04 16:59 +0200

Module	statements	missing	excluded	branches	partial	coverage
src__init__.py	0	0	0	0	0	100%
src\api__init__.py	0	0	0	0	0	100%
src\api\slack_api_connection.py	138	33	0	46	4	71%
src\chatbot__init__.py	0	0	0	0	0	100%
src\chatbot\cadocs_slack.py	70	0	0	36	2	98%
src\chatbot\cadocs_utils.py	24	16	0	4	1	32%
src\chatbot\intent_manager.py	21	0	0	8	1	97%
src\intent_handling__init__.py	0	0	0	0	0	100%
src\intent_handling\cadocs_intents.py	6	0	0	0	0	100%
src\intent_handling\intent_resolver.py	17	0	0	10	2	93%
src\intent_handling\tool_selector.py	15	0	0	0	0	100%
src\intent_handling\tool_strategy.py	6	1	0	0	0	83%
src\intent_handling\tools.py	23	0	0	6	0	100%
src\service__init__.py	0	0	0	0	0	100%
src\service\cadocs_messages.py	114	0	0	72	14	92%
src\service\language_handler.py	24	0	0	4	0	100%
src\service\utils.py	22	0	0	4	0	100%
Total	480	50	0	190	24	86%

Figura 2.16: Branch coverage dopo la CR_2

2.2.2 Test di sistema

A seguito delle modifiche implementate in CR_2, è stato necessario aggiungere anche i test con input in lingua italiana presenti nel documento Test Case Specification. Inoltre, è stato necessario considerare un nuovo formato per i link accettati, permettendo solo link di GitHub che iniziassero con "https://".

Durante l'esecuzione di questi, è stato notato un comportamento diverso rispetto alla precedente esecuzione per i test case TC_GS_1, TC_GSD_1, TC_RP_1 e TC_IN_1,

i quali, presentando input non significativi come "aaaaaaaaaaaaaaaa", hanno ricevuto una risposta da parte del modello NLU di tipo "info" o "get_smells". Questo comportamento era stato notato in modo simile anche durante i precedenti test di sistema con il precedente modello NLU; tuttavia in quel caso l'intent restituito era di tipo "get_smell_date", restituendo quindi un messaggio di errore derivante dall'assenza di link e data. Per questo motivo, si ritiene che il comportamento ottenuto dal modello non si sia discostato da quello ottenuto precedentemente, risultando solo più evidente, per cui è stato deciso di considerare tali test come passati. In ogni caso, il focus di questa fase di test non è sul modello di NLU, per cui è stata effettuata una fase di input testing.

Per quanto riguarda i test case TC_GS_1, TC_GSD_1, TC_RP_1 e TC_IN_1, è stato eseguito un unico test sia per la lingua inglese che italiana.

Di seguito sono riportati tutti i risultati ottenuti in fase di test di sistema:

get_smells

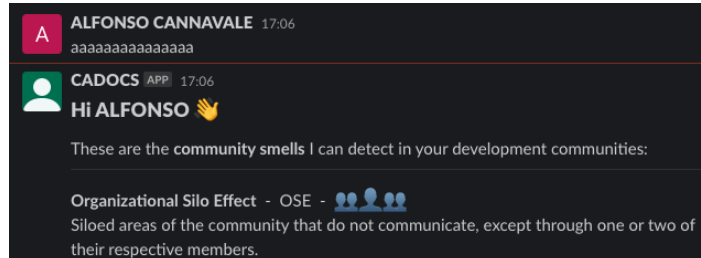


Figura 2.17: TC_GS_1_ENG_ITA

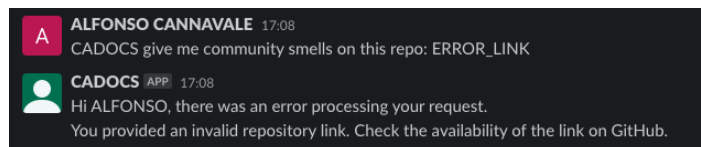


Figura 2.18: TC_GS_2_ENG

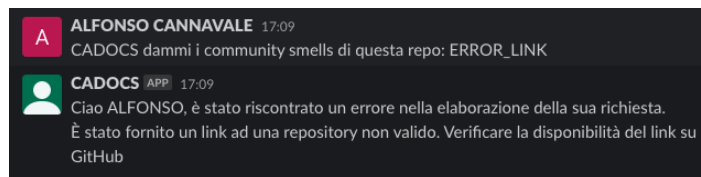


Figura 2.19: TC_GS_2_ITA

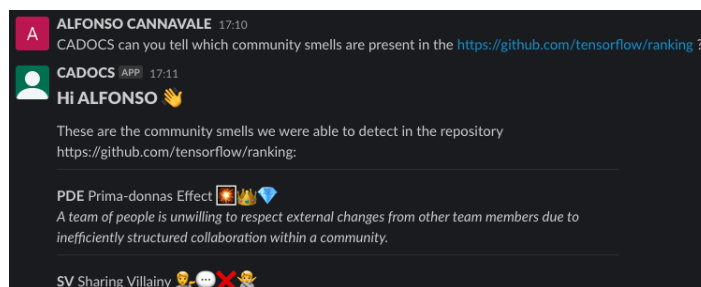


Figura 2.20: TC_GS_3_ENG



Figura 2.21: TC_GS_3_ITA

get_smells_date

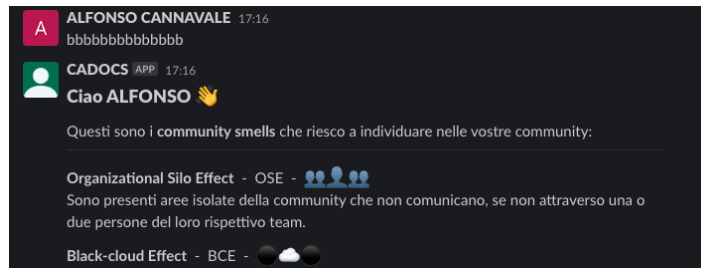


Figura 2.22: TC_GSD_1_ENG_ITA

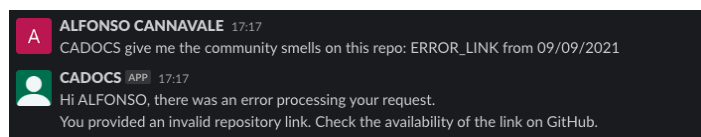


Figura 2.23: TC_GSD_2_ENG

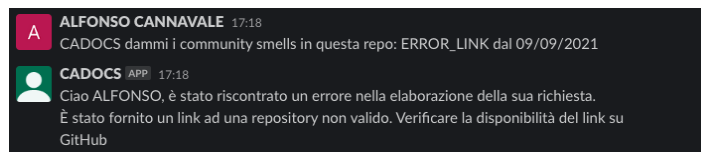


Figura 2.24: TC_GSD_2_ITA

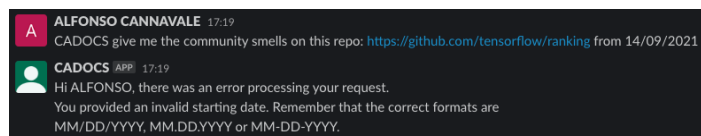


Figura 2.25: TC_GSD_3_ENG

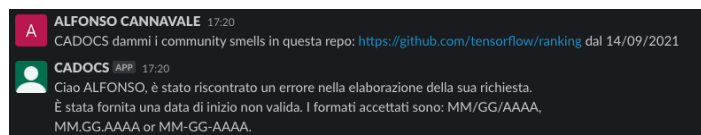


Figura 2.26: TC_GSD_3_ITA

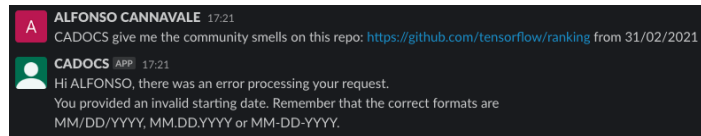


Figura 2.27: TC_GSD_4_ENG

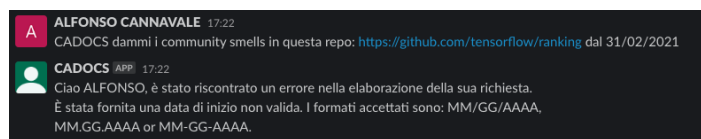


Figura 2.28: TC_GSD_4_ITA

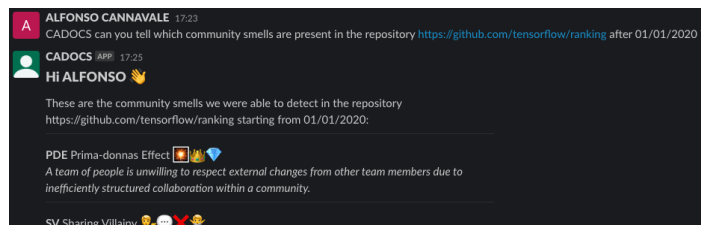


Figura 2.29: TC_GSD_5_ENG

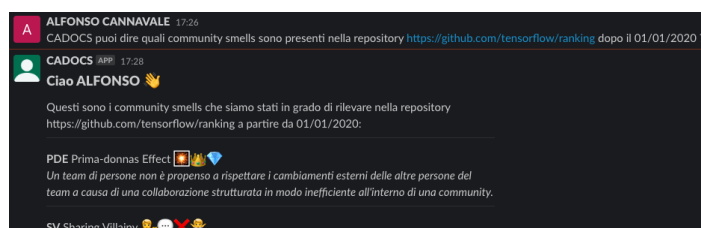


Figura 2.30: TC_GSD_5_ITA

report

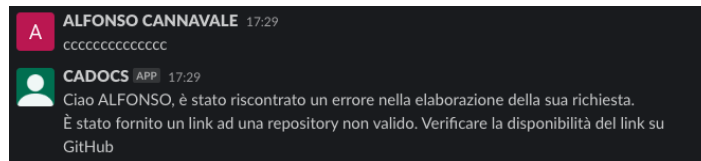


Figura 2.31: TC_RP_1_ENG_ITA



Figura 2.32: TC_RP_2_ENG

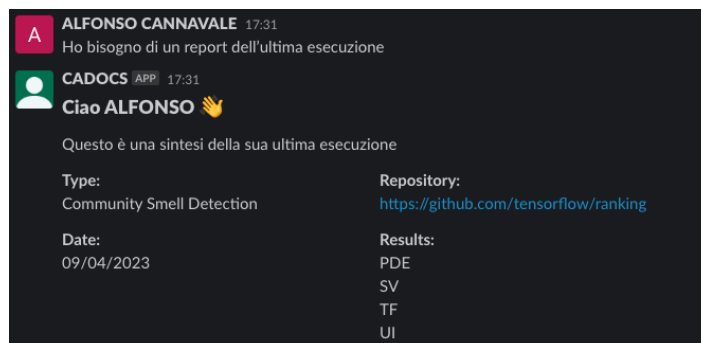


Figura 2.33: TC_RP_2_ITA

info

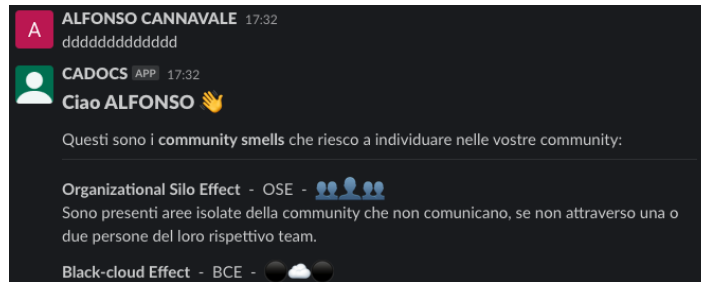


Figura 2.34: TC_IN_1_ENG_ITA

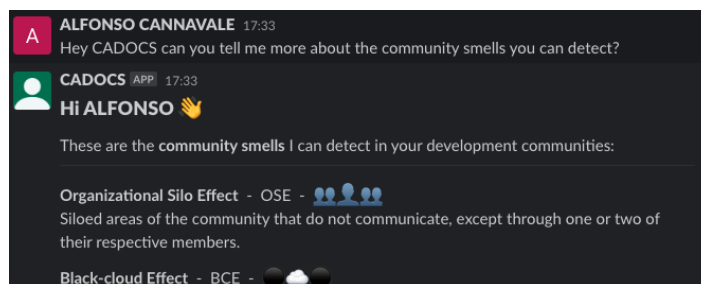


Figura 2.35: TC_IN_2_ENG

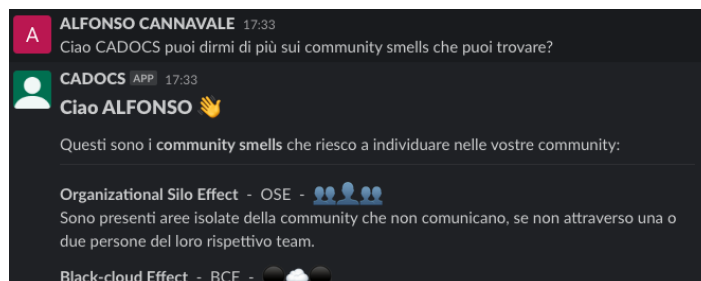


Figura 2.36: TC_IN_2_ITA

2.2.3 Input Testing

Per questa CR sono stati modificati anche i modelli NLU su cui si basa il sistema CADOCS. Tuttavia, non essendo la realizzazione dei modelli parte di questo progetto, i risultati del testing effettuato a seguito della realizzazione non sono stati qui documentati.

I risultati dell'input testing sono mostrati ai seguenti link:

- **modello inglese:** https://github.com/alfcan/CADOCS_NLU_Model/blob/dev/jupiters/input_testing_english.ipynb
- **modello italiano:** https://github.com/alfcan/CADOCS_NLU_Model/blob/dev/jupiters/input_testing_italian.ipynb

2.2.4 Conclusioni

Nella tabella 2.2 è mostrato un riassunto dei test eseguiti a seguito dell'implementazione della change request 2. In particolare sono riportati i risultati relativi all'esecuzione dei test di regressione, dei test di unità e integrazione aggiunti e dei test di sistema.

In seguito all'esecuzione dei test qui riportati, è stata modificata la descrizione del community smell **Toxic Communication** nei file *community_smell.json* e *community_smell_ita.json*, rieseguendo i test di unità e di integrazione con gli stessi risultati. È stato scelto di non rieseguire i test di sistema poiché questa modifica non impatta sul flusso del progetto.

	Branch coverage	Totale TC	Totale passati	Percentuale passati
Test di unità e integrazione	86%	114	114	100%
Test di sistema		20	20	100%

Tabella 2.2: Test report post-maintenance della CR_2

2.3 CR_3: Docker Porting csDetector

2.3.1 Test di unità e di integrazione

Le modifiche per il Docker Porting del modulo csDetector non hanno richiesto di modificare la test suite. Al termine dell'implementazione della CR_3 abbiamo eseguito i test di unità ed integrazione, i quali sono tutti passati. Come è possibile notare in figura 2.37 la coverage è ancora dell'86%.

Coverage report: 86%
coverage.py v7.3.0, created at 2023-09-08 16:04 +0200

Module	statements	missing	excluded	branches	partial	coverage
src__init__.py	0	0	0	0	0	100%
src\api__init__.py	0	0	0	0	0	100%
src\api\slack_api_connection.py	139	33	0	46	4	71%
src\chatbot__init__.py	0	0	0	0	0	100%
src\chatbot\cadocs_slack.py	70	0	0	36	2	98%
src\chatbot\cadocs_utils.py	24	16	0	4	1	32%
src\chatbot\intent_manager.py	21	0	0	8	1	97%
src\intent_handling__init__.py	0	0	0	0	0	100%
src\intent_handling\cadocs_intents.py	6	0	0	0	0	100%
src\intent_handling\intent_resolver.py	17	0	0	10	2	93%
src\intent_handling\tool_selector.py	15	0	0	0	0	100%
src\intent_handling\tool_strategy.py	6	1	0	0	0	83%
src\intent_handling\tools.py	25	1	0	8	1	94%
src\service__init__.py	0	0	0	0	0	100%
src\service\cadocs_messages.py	114	0	0	72	14	92%
src\service\language_handler.py	24	0	0	4	0	100%
src\service\utils.py	22	0	0	4	0	100%
Total	483	51	0	192	25	86%

Figura 2.37: Branch coverage dopo la CR_3

2.3.2 Test di sistema

Durante l'esecuzione dei test di sistema è stata cambiata l'immagine Docker utilizzata, a causa di problemi di compatibilità derivanti dai processori utilizzati per il processo di build. Il Dockerfile non ha però subito modifiche.

In particolare, l'immagine Docker presentava problemi di compatibilità con processori ARCH/arm64/v8 per cui è stata realizzata un'immagine Docker specifica. Questa immagine è stata testata solo sulle funzionalità principali e non sono stati rilevati errori, per cui per motivi di leggibilità i test dell'immagine non sono stati qui documentati. I risultati ottenuti dalla seconda immagine, infatti, sono identici ai risultati riportati di seguito.

get_smells

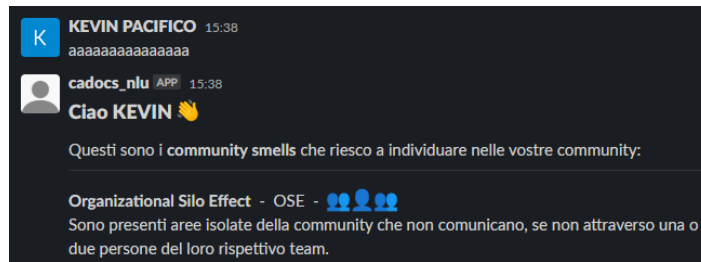


Figura 2.38: TC_GS_1_ENG_ITA

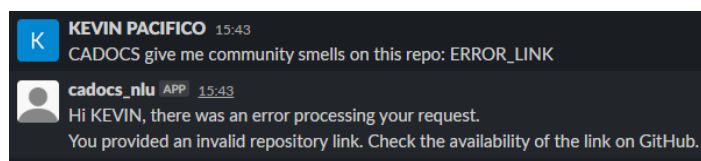


Figura 2.39: TC_GS_2_ENG

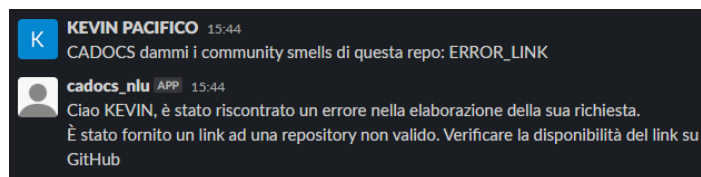


Figura 2.40: TC_GS_2_ITA

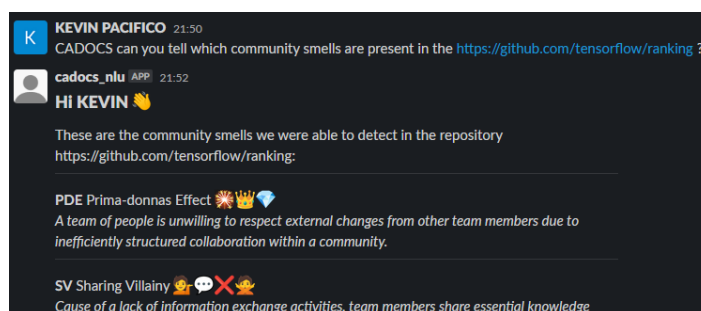


Figura 2.41: TC_GS_3_ENG

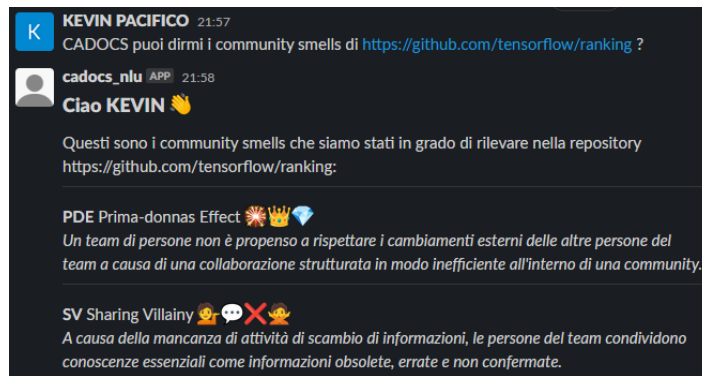


Figura 2.42: TC_GS_3_ITA

get_smells_date

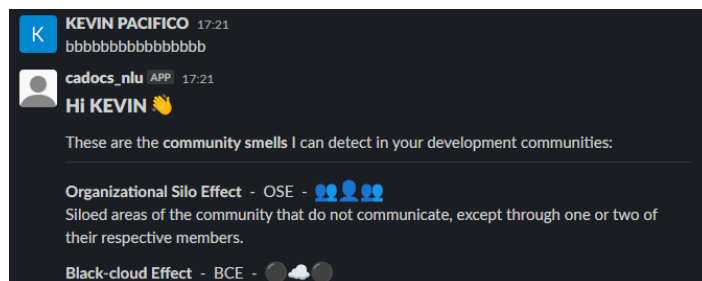


Figura 2.43: TC_GSD_1_ENG_ITA

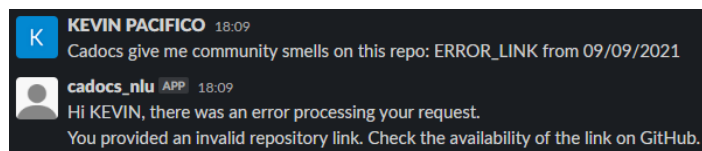


Figura 2.44: TC_GSD_2_ENG

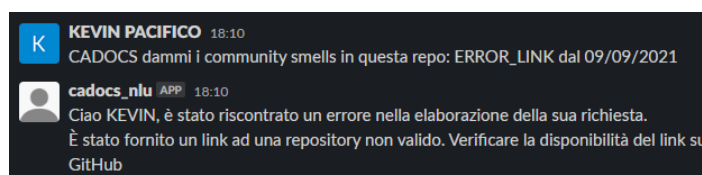


Figura 2.45: TC_GSD_2_ITA

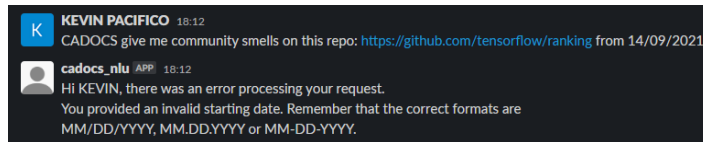


Figura 2.46: TC_GSD_3_ENG

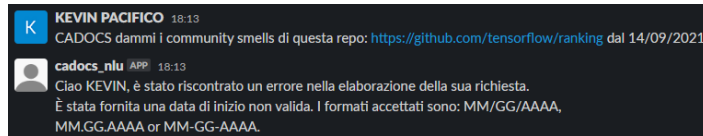


Figura 2.47: TC_GSD_3_ITA

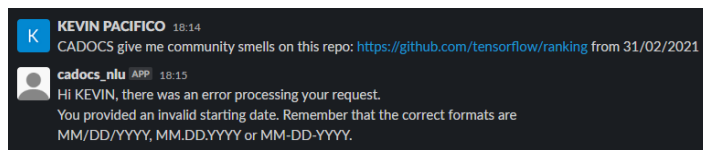


Figura 2.48: TC_GSD_4_ENG

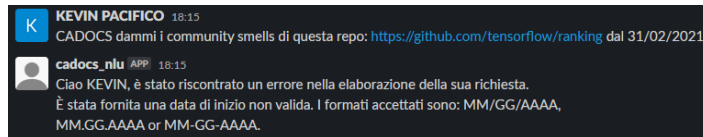


Figura 2.49: TC_GSD_4_ITA

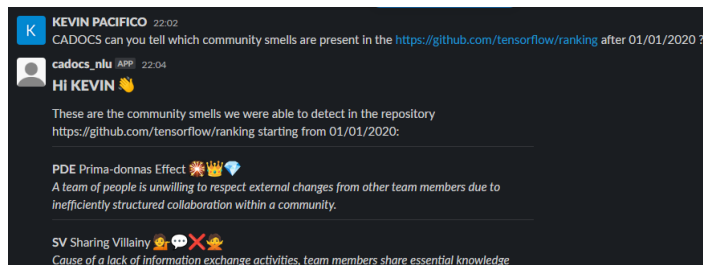


Figura 2.50: TC_GSD_5_ENG

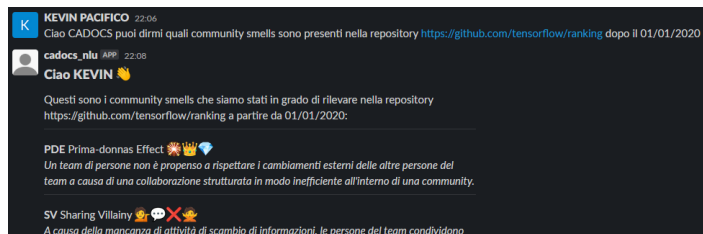


Figura 2.51: TC_GSD_5_ITA

report

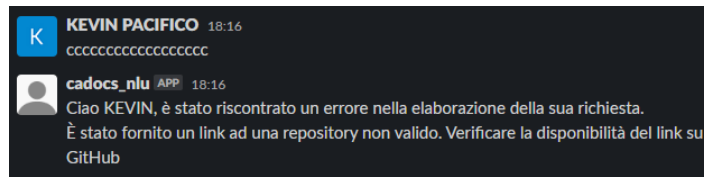


Figura 2.52: TC_RP_1_ENG_ITA

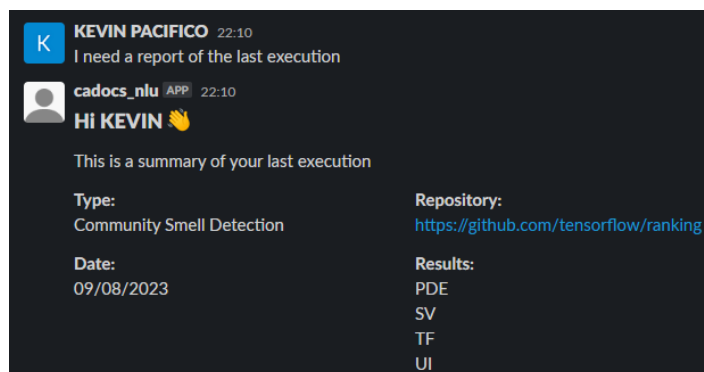


Figura 2.53: TC_RP_2_ENG



Figura 2.54: TC_RP_2_ITA

info

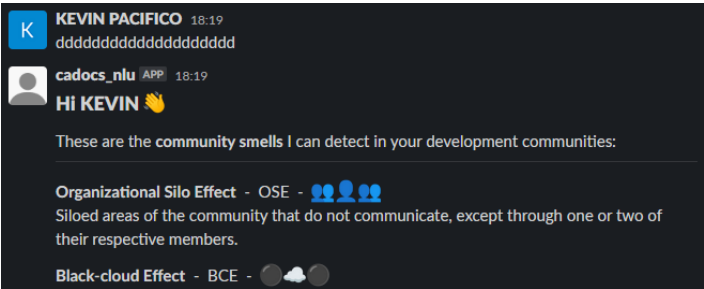


Figura 2.55: TC_IN_1_ENG_ITA

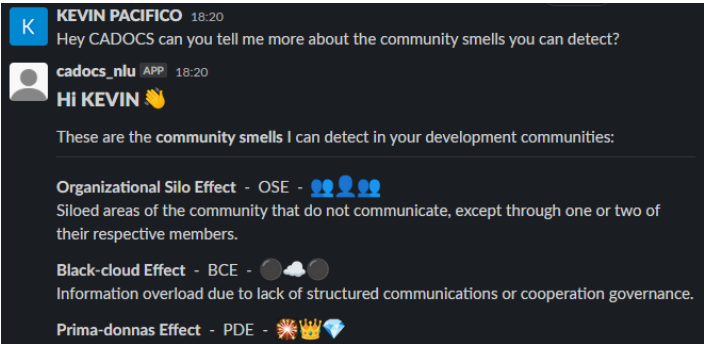


Figura 2.56: TC_IN_2_ENG

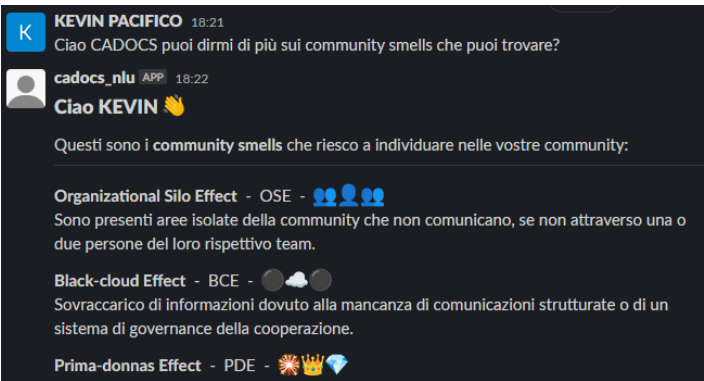


Figura 2.57: TC_IN_2_ITA

2.3.3 Conclusioni

Nella tabella 2.3 è mostrato un riassunto dei test eseguiti a seguito dell'implementazione della change request 3. In particolare sono riportati i risultati relativi all'esecuzione dei test di unità, dei test di integrazione e dei test di sistema.

	Branch coverage	Totale TC	Totale passati	Percentuale passati
Test di unità e integrazione	86%	114	114	100%
Test di sistema		20	20	100%

Tabella 2.3: Test report post-maintenance della CR_3