

Test Execution Report for Mark

a.prisco50@studenti.unisa.it 0522501976

c.ranieri7@studenti.unisa.it 0522501977

June 30, 2025

1 Version History

Versione	Data	Descrizione
0.1	25/06/2025	Stesura della struttura del documento.
1.0	25/06/2025	Prima stesura del documento.

2 Riferimenti

All'interno del documento vengono citate le seguenti informazioni.

Riferimento	Descrizione
IGES_Mark_TPD	Documento relativo alla pianificazione della fase di testing.
IGES_Mark_TCS	Documento relativo alla specifica dei casi di test.
IGES_Mark_CCR	Documento relativo alla comprensione del codice in cui sono stati identificati una serie di problemi relativi ai dettagli tecnici del sistema.

3 Introduzione

Di seguito sono riportate tutte le informazioni relative all'esecuzione del piano di testing prima dell'intervento di reengineering.

Obiettivo del Documento Questo documento ha l'obiettivo di tenere traccia di tutte le informazioni relative ai risultati della fase di testing, le quali sono state raccolte in tabelle dettagliate e presentate nelle sezioni seguenti.

Contesto del Progetto Mark è un tool scritto in Python che dato un progetto, tramite l'ausilio di una Knowledge Base e delle detection rules, può classificare progetti ML in tre diverse categorie: ML-Consumer, ML-Producer e ML-Producer e Consumer.

4 Risultati della Fase di Testing Funzionale

Di seguito sono riportati i risultati dei test descritti nei documenti **IGES_Mark_TPD** e **IGES_Mark_TCS**.

Nota. I test sono stati eseguiti una sola volta perchè le funzioni implementate non presentavano errori.

ID	Comportamento Rilevato	Esito Test
EA_0	Il sistema non va in crash a causa di path errato e avverte l'utente.	PASSED
EA_1	Nessuna classificazione effettuata, creati solo <i>results_first_step.csv</i> e <i>results_consumer.csv</i> .	PASSED
EA_2	Corretta classificazione del progetto come producer.	PASSED
EA_3	Corretta classificazione del progetto come consumer.	PASSED
EA_4	Corretta classificazione del progetto come producer e consumer.	PASSED
EA_5	Corretta classificazione del progetto come no-producer-consumer.	PASSED
EA_6	Corretta classificazione dei progetti come producer e consumer.	PASSED
EA_7	Corretta classificazione dei progetti come consumer e no-producer-consumer.	PASSED
EA_8	Corretta classificazione dei progetti come producer-consumer e producer.	PASSED
EA_9	Corretta classificazione dei progetti come no-producer-consumer, producer, consumer e producer-consumer.	PASSED

ID	Comportamento Rilevato	Esito Test
CL_0	Il sistema va in crash a causa dell'assenza del file indicato come input.	FAILED
CL_1	Nessuna clonazione effettuata, come atteso in caso di file vuoto.	PASSED
CL_2	Nessuna clonazione effettuata per l'unica repository elencata, come previsto.	PASSED
CL_3	Nessuna clonazione effettuata per le repository elencate, come previsto.	PASSED

5 Risultati della Fase di Testing Strutturale

Di seguito sono riportati i risultati dei test descritti nei documenti **IGES_Mark_TPD** e **IGES_Mark_TCS**

Nota. I test sono stati eseguiti una sola volta perchè per le funzioni implementate si è raggiunta immediatamente una coverage media dell'80%.

Modulo	Tot	Non	Parz	Salt	Cov.	Righe non coperte / Note
<i>library_extractor.py</i>	33	10	12	0	78%	11–20
<i>consumer_classifier_by_dict.py</i>	135	20	60	9	85%	12→14, 31→34, 40–41, 50, 59, 87–89, 103, 119–124, 138, 174, 200, 204–208
<i>exec_analysis.py</i>	56	10	18	7	77%	25→24, 35–36, 40–41, 70–71, 73–74, 78–79, 89→exit
<i>producer_classifier_by_dict.py</i>	113	16	46	6	86%	20→23, 29–30, 52–54, 76–81, 95, 129, 134→131, 156–158
<i>Results_Analysis.py</i>	99	0	38	9	93%	31→30, 57→56, 86→85, 146→145, 172→171, 198→197, 223→222, 253→252, 266→exit
<i>merge.py</i>	50	0	2	1	98%	110→exit
<i>cloning_check.py</i>	50	2	22	3	93%	10, 44, 61→exit
<i>Dopo esecuzione di notebook_converter</i>						
<i>notebook_converter.py</i>	29	5	12	2	83%	18, 29–30, 34→exit, 43–44
<i>Su 341 casi di test</i>						
<i>cloner.py</i>	86	20	18	6	71%	18, 38, 47, 98–100, 119–133, 136→exit, 147–149, 152

Nota. Di seguito si riporta il significato delle colonne:

- **Tot** – Numero totale di righe del modulo.
- **Non** – Righe non coperte da alcun test.
- **Parz** – Righe parzialmente coperte (ad esempio, condizioni booleane valutate solo in parte).
- **Salt** – Righe saltate esplicitamente durante l'esecuzione.
- **Cov.** – Percentuale di copertura totale raggiunta sul modulo.
- **Righe non coperte / Note** – Dettagli sulle righe non eseguite e note di copertura generate dallo strumento di analisi (es. salti condizionali o blocchi non attraversati).