



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Ingegneria, Gestione ed Evoluzione del Software

Post-Modification Test Case Specification

TEAM MEMBER

Donia Daniele - 0522501575

La Marca Antonio - 0522501557

Somma Pasquale - 0522501543

| | | |
|----------|---|----------|
| 1 | Introduzione | 1 |
| 2 | Categorie e scelte | 2 |
| 2.1 | Test Case Specification di unità | 2 |
| 2.1.1 | Repo Mining | 2 |
| 2.1.2 | Text Mining | 26 |
| 2.1.3 | Software Metrics | 64 |
| 2.1.4 | Rifinitura dati analisi statica | 142 |
| 2.1.5 | Union | 179 |
| 2.1.6 | Main | 181 |
| 2.2 | Test Case Specification di integrazione | 218 |
| 2.2.1 | Repo Mining | 218 |
| 2.2.2 | Text Mining | 226 |
| 2.2.3 | Software Metrics | 237 |
| 2.2.4 | Rifinitura dati analisi statica | 247 |
| 2.2.5 | RunPrediction | 251 |
| 2.3 | Test Case Specification di sistema | 258 |

CAPITOLO 1

Introduzione

Il seguente documento fornisce la specifica dettagliata dei test sviluppati dopo l'implementazione delle change request. I test case sono organizzati in base alle categorie chiave del sistema e coprono vari livelli di test, inclusi test di unità, integrazione e sistema. Ogni test case include una descrizione dettagliata degli input, degli output attesi e delle condizioni necessarie per il corretto funzionamento, facilitando così la verifica e la validazione delle modifiche apportate al sistema.

 Categorie e scelte

Di seguito sono riportate le categorie e le scelte individuate per gli oggetti dell'ambiente citati nella sezione precedente.

2.1 Test Case Specification di unità

2.1.1 Repo Mining

Divide Dataset

| | |
|--|---------------------|
| Test case ID: TC_1.1_1 | Test Frame: EI2 |
| Input | |
| Il file initial_dataset.csv e la directory Dataset_Divided non esistono. | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestDatasetDivider | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.1: Metodo della classe DatasetDivider

| | |
|--|--|
| Test case ID: TC_1.1_2 | Test Frame: EI1 N1 ED1 S1 F1 E1 |
| Input | |
| Il file <code>initial_dataset.csv</code> contiene solo l'header, senza righe di dati. | |
| Output | |
| La directory <code>Dataset_Divided</code> contiene il file <code>1.csv</code> con solo l'header. | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_2</code> |
| Oracolo | |
| La directory <code>Dataset_Divided</code> contiene il file <code>1.csv</code> con solo l'header. | |

Tabella 2.2: Metodo della classe `DatasetDivider`

| | |
|---|----------------------------------|
| Test case ID: TC_1.1_3 | Test Frame: S2 |
| Input | |
| Il file <code>initial_dataset.csv</code> esiste e contiene 50 record. La directory <code>Dataset_Divided</code> esiste, ma non vi sono i permessi di scrittura. | |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_3</code> |
| Oracolo | |
| PermissionError | |

Tabella 2.3: Metodo della classe `DatasetDivider`

| | |
|---|--|
| Test case ID: TC_1.1_4 | Test Frame: EI1 N2 ED1 S1 F1 E1 |
| Input | |
| Il file <code>initial_dataset.csv</code> esiste e contiene 50 record. La directory <code>Dataset_Divided</code> esiste. | |
| Output | |
| <code>Dataset_Divided</code> contiene <code>1.csv</code> con 50 record | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_4</code> |

| Oracolo |
|--|
| Dataset_Divided contiene 1.csv con 50 record |

Tabella 2.4: Metodo della classe DatasetDivider

| Test case ID: TC_1.1_5 | Test Frame: EI1 N3 ED1 S1 F1 E1 |
|--|----------------------------------|
| Input | |
| Il file <code>initial_dataset.csv</code> contiene 60 record. La directory <code>Dataset_Divided</code> esiste | |
| Output | |
| Dataset_Divided contiene i file <code>1.csv</code> e <code>2.csv</code> . <code>1.csv</code> contiene l'header e 50 record, <code>2.csv</code> è sprovvisto di header e contiene 10 record | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_5</code> |
| Oracolo | |
| Dataset_Divided contiene i file <code>1.csv</code> e <code>2.csv</code> . <code>1.csv</code> contiene l'header e 50 record, <code>2.csv</code> è sprovvisto di header e contiene 10 record | |

Tabella 2.5: Metodo della classe DatasetDivider

| Test case ID: TC_1.1_6 | Test Frame: EI1 N3 ED2 S1 F1 E1 |
|---|----------------------------------|
| Input | |
| Il file <code>initial_dataset.csv</code> contiene 60 record. La directory <code>Dataset_Divided</code> non esiste | |
| Output | |
| È creata la directory <code>Dataset_Divided</code> contenente file <code>1.csv</code> e <code>2.csv</code> . <code>1.csv</code> contiene l'header e 50 record, <code>2.csv</code> è sprovvisto di header e contiene 10 record | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_6</code> |
| Oracolo | |

È creata la directory `Dataset_Divided` contenente file `1.csv` e `2.csv`.
`1.csv` contiene l'header e 50 record,
`2.csv` è sprovvisto di header e contiene 10 record

Tabella 2.6: Metodo della classe `DatasetDivider`

| Test case ID: TC_1.1_7 | Test Frame: F3 |
|---|----------------------------------|
| Input | |
| La variabile <code>base_dir</code> non è una stringa | |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_7</code> |
| Oracolo | |
| La directory <code>Dataset_Divided</code> deve essere aggiornata correttamente con il nuovo file CSV. | |

Tabella 2.7: Metodo della classe `DatasetDivider`

| Test case ID: TC_1.1_8 | Test Frame: F2 |
|---|----------------------------------|
| Input | |
| La variabile <code>base_dir</code> non ha un formato valido | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: <code>TestDatasetDivider</code> | Metodo: <code>test_case_8</code> |
| Oracolo | |
| OSError | |

Tabella 2.8: Metodo della classe `DatasetDivider`

| Test case ID: TC_1.1_9 | Test Frame: F1 EB2 |
|--|--------------------|
| Input | |
| La variabile <code>base_dir</code> ha un formato valido ma indica path non esistente | |
| Output | |

| | |
|----------------------------|---------------------|
| FileNotFoundException | |
| Test Driver | |
| Classe: TestDatasetDivider | Metodo: test_case_9 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.9: Metodo della classe DatasetDivider**Initialize**

| | |
|---|------------------------|
| Test case ID: TC_1.2_1 | Test Frame: FM2 |
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato invalido.</p> <p>Il dataset da minare esiste e ha un formato valido.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_1 |
| Oracolo | |
| OSError | |

Tabella 2.10: Metodo della classe RepoMiner

| | |
|--|----------------------------|
| Test case ID: TC_1.2_2 | Test Frame: FM1 ED2 |
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare non esiste.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| FileNotFoundException | |

| Test Driver | |
|--|---------------------|
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_2 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.11: Metodo della classe RepoMiner

| Test case ID: TC_1.2_3 | Test Frame: FM1 ED1 FD1 ND3 EM2 ER1 |
|--|-------------------------------------|
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste ed è valido.</p> <p>La directory <code>mining_results</code> non esiste.</p> | |
| Output | |
| Il dizionario data contiene i record contenuti nel dataset e la directory è creata | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_3 |
| Oracolo | |
| Il dizionario data contiene i record contenuti nel dataset e la directory è creata | |

Tabella 2.12: Metodo della classe RepoMiner

| Test case ID: TC_1.2_4 | Test Frame: FM1 ED1 EM1 FD2 |
|--|-----------------------------|
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste ma ha un formato invalido.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| ValueError | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_4 |
| Oracolo | |
| ValueError | |

Tabella 2.13: Metodo della classe RepoMiner

| | |
|--|--|
| Test case ID: TC_1.2_5 | Test Frame: FM1 ED1 FD1 ND3 EM1 ER1 |
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e contiene 3 record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| Il dizionario data contiene i record contenuti nel dataset | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_5 |
| Oracolo | |
| Il dizionario data contiene i record contenuti nel dataset | |

Tabella 2.14: Metodo della classe RepoMiner

| | |
|--|--|
| Test case ID: TC_1.2_6 | Test Frame: FM1 ED1 FD1 ND1 EM1 ER2 |
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e non contiene record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| Il dizionario data è vuoto e la repository è creata. | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_6 |
| Oracolo | |
| Il dizionario data è vuoto. | |

Tabella 2.15: Metodo della classe RepoMiner

| | |
|--|--|
| Test case ID: TC_1.2_7 | Test Frame: FM1 ED1 FD1 ND1 EM1 ER1 |
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e non contiene record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| Il dizionario data è vuoto | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_7 |
| Oracolo | |
| Il dizionario data è vuoto | |

Tabella 2.16: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.2_8 | Test Frame: FM1 ED1 FD1 ND2 EM1 ER1 |
| Input | |
| <p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e contiene un record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p> | |
| Output | |
| Il dizionario <code>data</code> contiene i commit specificati nel dataset. | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_8 |
| Oracolo | |
| Il dizionario <code>data</code> contiene i commit specificati nel dataset. | |

Tabella 2.17: Metodo della classe RepoMiner

| | |
|---|------------------------|
| Test case ID: TC_1.2_9 | Test Frame: FB2 |
| Input | |
| La variabile <code>base_dir</code> ha un formato non valido | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestInitialize | Metodo: test_case_9 |
| Oracolo | |
| OSError | |

Tabella 2.18: Metodo della classe RepoMiner

| | |
|---|----------------------------|
| Test case ID: TC_1.2_10 | Test Frame: FB1 EB2 |
| Input | |
| La variabile <code>base_dir</code> ha un formato valido ma indica un percorso non esistente | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestInitialize | Metodo: test_case_10 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.19: Metodo della classe RepoMiner**Start Mining Repo**

| | |
|---|---|
| Test case ID: TC_1.3_1 | Test Frame: FD1 N1 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato corretto e non contiene alcun record.</p> <p>Il parametro <code>cwd</code> ha un formato valido e indica un percorso esistente.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| File di log è vuoto. | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_1 |
| Oracolo | |
| File di log è vuoto. | |

Tabella 2.20: Metodo della classe RepoMiner

| | |
|--|---------------------|
| Test case ID: TC_1.3_2 | Test Frame: FD2 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato non valido.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_2 |
| Oracolo | |
| KeyError | |

Tabella 2.21: Metodo della classe RepoMiner

| | |
|--|---|
| Test case ID: TC_1.3_3 | Test Frame: FD1 N3 CL3 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido, ha 2 record, ma contiene link non valido.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log indica: Invalid URL | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: <code>test_case_3</code> |
| Oracolo | |
| Log indica errore: Invalid URL | |

Tabella 2.22: Metodo della classe RepoMiner

| | |
|---|---|
| Test case ID: TC_1.3_4 | Test Frame: FD1 N3 CL2 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido, ha 2 record e contiene link valido ma non esistente.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log indica: Connection Error | |
| Test Driver | |
| Classe: TestStartMiningRepo | Metodo: <code>test_case_4</code> |
| Oracolo | |
| Log indica errore: ConnectionError | |

Tabella 2.23: Metodo dello script `test_start_mining_repo.py`

| | |
|---|---|
| Test case ID: TC_1.3_5 | Test Frame: FD1 N3 CL1 CR2 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido ma contiene link a una repository non valida.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log indica errore: la repository non è disponibile. | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_5 |
| Oracolo | |
| Log indica errore: la repository non è disponibile. | |

Tabella 2.24: Metodo della classe RepoMiner

| | |
|---|---|
| Test case ID: TC_1.3_6 | Test Frame: FD1 N3 CL1 CR1 CC2 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido ma contiene link a un commit con path eccedente la lunghezza massima.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log indica errore: GitCommandError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_7 |
| Oracolo | |

| |
|------------------------------------|
| Log indica errore: GitCommandError |
|------------------------------------|

Tabella 2.25: Metodo della classe RepoMiner

| | |
|---|---|
| Test case ID: TC_1.3_7 | Test Frame: FD1 N3 CL1 CR1 CC3 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido ma contiene link a un commit non esistente.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log indica errore: commit non è esistente. | |
| Test Driver | |
| <p>Classe: TestRepoMiner</p> <p>Sottoclasse: TestStartMiningRepo</p> | Metodo: <code>test_case_7</code> |
| Oracolo | |
| Log indica errore: commit non è esistente. | |

Tabella 2.26: Metodo della classe RepoMiner

| | |
|--|---|
| Test case ID: TC_1.3_8 | Test Frame: FD1 N3 CL1 CR1 CC4 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido ma contiene link a due repo con commit non definiti.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log indica errore: commit non definito | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_8 |
| Oracolo | |
| Log indica errore: commit non definito | |

Tabella 2.27: Metodo della classe RepoMiner

| | |
|--|---|
| Test case ID: TC_1.3_9 | Test Frame: FD1 N3 CL1 CR1 CC1 FW1 EW1 FR1 ER1 ERI1 ERC1 SL1 FB1 EB1 |
| Input | |
| <p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, senza modifiche.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>La variabile <code>base_dir</code> è valida e indica path esistente</p> | |
| Output | |
| Log informa: status ok | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_9 |
| Oracolo | |
| Log informa: status ok | |

Tabella 2.28: Metodo della classe RepoMiner

| | |
|--|-----------------|
| Test case ID: TC_1.3_10 | Test Frame: FR3 |
| Input | |
| Il parametro <code>repoName</code> non è una stringa | |
| Output | |
| TypeError | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_10 |
| Oracolo | |
| TypeError | |

Tabella 2.29: Metodo della classe RepoMiner

| | |
|---|------------------------|
| Test case ID: TC_1.3_11 | Test Frame: FR2 |
| Input | |
| Il parametro repoName ha un formato non valido | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_11 |
| Oracolo | |
| OSError | |

Tabella 2.30: Metodo della classe RepoMiner

| | |
|--|----------------------------|
| Test case ID: TC_1.3_12 | Test Frame: FR1 ER2 |
| Input | |
| Il parametro repoName ha un formato valido (FR1), ma la repository non esiste (ER2). | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: test_case_12 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.31: Metodo della classe RepoMiner

| | |
|--------------------------------|------------------------|
| Test case ID: TC_1.3_13 | Test Frame: FB2 |
|--------------------------------|------------------------|

| Input | |
|---|-----------------------------------|
| La variabile <code>base_dir</code> ha un formato non valido | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: <code>test_case_13</code> |
| Oracolo | |
| OSError | |

Tabella 2.32: Metodo della classe RepoMiner

| Test case ID: TC_1.3_14 | Test Frame: FB1 EB2 |
|---|-----------------------------------|
| Input | |
| Il parametro <code>base_dir</code> ha un formato valido, ma indica un percorso non esistente. | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: <code>test_case_14</code> |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.33: Metodo della classe RepoMiner

| Test case ID: TC_1.3_15 | Test Frame: EC1 SC2 |
|---|-----------------------------------|
| Input | |
| Il file log non ha permessi di scrittura | |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestStartMiningRepo | Metodo: <code>test_case_15</code> |

| Oracolo |
|-----------------|
| PermissionError |

Tabella 2.34: Metodo della classe RepoMiner**Analyze Commit**

| Test case ID: TC_1.4_1 | Test Frame: CE2 |
|---|---------------------|
| Input | |
| Il parametro commit è nullo cve_path = "test/path/mining_results/1", commit_path = "test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2" | |
| Output | |
| AttributeError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_1 |
| Oracolo | |
| AttributeError | |

Tabella 2.35: Metodo della classe RepoMiner

| Test case ID: TC_1.4_2 | Test Frame: FCV2 |
|---|---------------------|
| Input | |
| Il commit è valido cve_path = "test«304" commit_path = "test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2" | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_2 |

| Oracolo |
|---------|
| OSError |

Tabella 2.36: Metodo della classe RepoMiner

| Test case ID: TC_1.4_3 | Test Frame: FCV1 ECV2 |
|--|-----------------------|
| Input | |
| Il commit è valido cve_path = "test/not-exist-path" commit_path = "test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2" | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_3 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.37: Metodo della classe RepoMiner

| Test case ID: TC_1.4_4 | Test Frame: FCM2 |
|---|---------------------|
| Input | |
| Il commit è valido cve_path = "test/path/mining_results/1" commit_path = "test«304" | |
| Output | |
| OSError | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_4 |
| Oracolo | |
| OSError | |

Tabella 2.38: Metodo della classe RepoMiner

| | |
|--|-----------------------|
| Test case ID: TC_1.4_5 | Test Frame: FCM1 ECM2 |
| Input | |
| Il commit è valido cve_path = "test/path/mining_results/1" commit_path = "test/not-exist-path" | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_5 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.39: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.4_6 | Test Frame: CE1 CM1 FCV1 ECV1 FCM1 ECM1 EDCV2 EDCM2 |
| Input | |
| Il commit è valido ma non contiene modifiche cve_path = "test/path/mining_results/1" commit_path = "test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2" Le directory '1' e 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' non esistono | |
| Output | |
| Dir cve e commit non sono create | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_6 |
| Oracolo | |
| Dir cve e commit non sono create | |

Tabella 2.40: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.4_7 | Test Frame: CE1 CM2 FCV1 ECV1 FCM1 ECM2 EDCV2 EDCM2 |
| Input | |
| <p>Il commit è valido e le modifiche non contengono classi .java</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>Le directory '1' e 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' non esistono</p> | |
| Output | |
| Dir cve e commit non sono create | |
| Test Driver | |
| <p>Classe: TestRepoMiner</p> <p>Sottoclasse: TestAnalyzeCommit</p> | Metodo: test_case_7 |
| Oracolo | |
| Dir cve e commit non sono create | |

Tabella 2.41: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.4_8 | Test Frame: CE1 CM3 FCV1 ECV1 FCM1 ECM1 EDCV1 EDCM1 |
| Input | |
| <p>Il commit è valido e introduce classi .java</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>Le directory '1' e 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' esistono</p> | |
| Output | |
| Dir commit è vuota | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_8 |
| Oracolo | |
| Dir commit è vuota | |

Tabella 2.42: Metodo della classe RepoMiner

| | |
|--|--|
| Test case ID: TC_1.4_9 | Test Frame: CE1 CM3 FCV1 ECV1 FCM1 ECM1 EDCV1 EDCM2 |
| Input | |
| <p>Il commit è valido e introduce classi .java</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>La directory '1' esiste</p> <p>La directory 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' non esiste</p> | |
| Output | |
| Dir commit è creata e vuota | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_9 |
| Oracolo | |
| Dir commit è creata e vuota | |

Tabella 2.43: Metodo della classe RepoMiner

| | |
|-------------------------|--|
| Test case ID: TC_1.4_10 | Test Frame: CE1 CM3 FCV1 ECV1 FCM1 ECM1 EDCV2 EDCM1 |
| Input | |

| | |
|--|----------------------|
| <p>Il commit è valido e introduce classi .java</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>La directory '1' non esiste</p> <p>La directory 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' esiste</p> | |
| Output | |
| Dir CVE è creata e dir commit è vuota | |
| Test Driver | |
| <p>Classe: TestRepoMiner</p> <p>Sottoclasse: TestAnalyzeCommit</p> | Metodo: test_case_10 |
| Oracolo | |
| Dir CVE è creata e dir commit è vuota | |

Tabella 2.44: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.4_11 | Test Frame: CE1 CM3 FCV2 ECV2 FCM1 ECM1 EDCV1 EDCM1 |
| Input | |
| <p>Il commit è valido e introduce classi non .java</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>Le directory '1' e 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' esistono</p> | |
| Output | |
| <p>Dir cve e commit sono create</p> <p>e dir commit è vuota</p> | |
| Test Driver | |
| <p>Classe: TestRepoMiner</p> <p>Sottoclasse: TestAnalyzeCommit</p> | Metodo: test_case_11 |
| Oracolo | |
| <p>Dir cve e commit sono create</p> <p>e dir commit è vuota</p> | |

Tabella 2.45: Metodo della classe RepoMiner

| | |
|--|--|
| Test case ID: TC_1.4_12 | Test Frame: CE1 CM4 FCV1 ECV1 FCM1 ECM1 EDCV1 EDCM1 |
| Input | |
| <p>Il commit è valido e introduce file .java modificati</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>Le directory '1' e 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' esistono</p> | |
| Output | |
| Dir commit contiene java file modificati | |
| Test Driver | |
| <p>Classe: TestRepoMiner</p> <p>Sottoclasse: TestAnalyzeCommit</p> | Metodo: test_case_12 |
| Oracolo | |
| Dir commit contiene java file modificati | |

Tabella 2.46: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.4_13 | Test Frame: CE1 CM4 FCV1 ECV1 FCM1 ECM1 EDCV1 EDCM2 |
| Input | |
| <p>Il commit è valido e introduce file .java modificati</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>La directory '1' esiste</p> <p>La directory 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' non esiste</p> | |
| Output | |
| Dir commit è creata e contiene java file modificati | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_13 |
| Oracolo | |
| Dir commit è creata e contiene java file modificati | |

Tabella 2.47: Metodo della classe RepoMiner

| | |
|---|--|
| Test case ID: TC_1.4_14 | Test Frame: CE1 CM4 FCV1 ECV1 FCM1 ECM1 EDCV2 EDCM1 |
| Input | |
| <p>Il commit è valido e introduce file .java modificati</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>La directory '1' non esiste</p> <p>La directory 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' esiste</p> | |
| Output | |
| Dir cve è creata e dir commit contiene file java modificati | |
| Test Driver | |
| Classe: TestRepoMiner Sottoclasse: TestAnalyzeCommit | Metodo: test_case_14 |
| Oracolo | |
| Dir cve è creata e dir commit contiene file java modificati | |

Tabella 2.48: Metodo della classe RepoMiner

| | |
|-------------------------|--|
| Test case ID: TC_1.4_15 | Test Frame: CE1 CM4 FCV1 ECV1 FCM1 ECM1 EDCV2 EDCM2 |
| Input | |

| | |
|--|----------------------|
| <p>Il commit è valido e introduce file .java modificati</p> <p>cve_path = "test/path/mining_results/1"</p> <p>commit_path =</p> <p>"test/path/mining_results/1/d72bd78c19dfb7b57395a66ae8d9269d59a87bd2"</p> <p>Le directory '1' e 'd72bd78c19dfb7b57395a66ae8d9269d59a87bd2' non esistono</p> | |
| Output | |
| Dir cve e commit sono create e dir commit contiene file java modificati | |
| Test Driver | |
| <p>Classe: TestRepoMiner</p> <p>Sottoclasse: TestAnalyzeCommit</p> | Metodo: test_case_15 |
| Oracolo | |
| Dir cve e commit sono create e dir commit contiene file java modificati | |

Tabella 2.49: Metodo della classe RepoMiner

2.1.2 Text Mining

JavaTextMining

| | |
|--|------------------------|
| Test case ID: TC_2.1_1 | Test Frame: CN1 |
| Input | |
| Parametro: tokens | [] |
| Output | |
| Restituisce una lista vuota | |
| Test Driver | |
| <p>Classe: TestJavaTextMining</p> <p>SottoClasse: TestRemoveNotAlpha</p> | Metodo: test_case_1 |
| Oracolo | |
| Restituisce una lista vuota | |

Tabella 2.50: Metodo della classe JavaTextMining

| Test case ID: TC_2.1_2 | Test Frame: CN2 |
|---|--|
| Input | |
| Parametro: tokens | ["class", "public", "int", "String", "try", "catch", "main", "final", "while", "return"] |
| Output | |
| Restituisce una lista identica a quella di input | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestRemoveNotAlpha | Metodo: test_case_2 |
| Oracolo | |
| Restituisce una lista identica a quella di input | |

Tabella 2.51: Metodo della classe JavaTextMining

| Test case ID: TC_2.1_3 | Test Frame: CN3 |
|-----------------------------|--|
| Input | |
| Parametro: tokens | ["main1", "while2", "return4", "if1else", "for3", "abstract4"] |
| Output | |
| Restituisce una lista vuota | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestJavaTextMining SottoClasse: TestRemoveNotAlpha | Metodo: test_case_3 |
| Oracolo | |
| Restituisce una lista vuota | |

Tabella 2.52: Metodo della classe JavaTextMining

| | |
|--|--|
| Test case ID: TC_2.1_4 | Test Frame: CN4 |
| Input | |
| Parametro: tokens | ["class", "public", "int", "String", "try", "catch", "main1", "final", "while2", "return", "if1else", "for3", "abstract4"] |
| Output | |
| ["class", "public", "int", "String", "try", "catch", "final", "return"] | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestRemoveNotAlpha | Metodo: test_case_4 |
| Oracolo | |
| Restituisce una lista contenente le parole con soli caratteri alfabetici | |

Tabella 2.53: Metodo della classe JavaTextMining

| | |
|-------------------------------|------------------------|
| Test case ID: TC_2.2_1 | Test Frame: CN1 |
| Input | |
| Parametro: s | "" |

| Output | |
|--|---------------------|
| Restituisce una lista vuota | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestStringTokenizer | Metodo: test_case_1 |
| Oracolo | |
| Restituisce una lista vuota | |

Tabella 2.54: Metodo della classe JavaTextMining

| Test case ID: TC_2.2_2 | Test Frame: CN2 |
|---|-------------------------|
| Input | |
| Parametro: s | "int main() return 0; " |
| Output | |
| ["int", "main", "return"] | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestStringTokenizer | Metodo: test_case_2 |
| Oracolo | |
| Restituisce una lista con tutte e sole le parole all'interno della stringa di input | |

Tabella 2.55: Metodo della classe JavaTextMining

| Test case ID: TC_2.2_3 | Test Frame: CN3 |
|------------------------|-----------------|
|------------------------|-----------------|

| Input | |
|---|--|
| Parametro: s | "/ / this is a comment\n /* This is another block comment \n that spans multiple lines */" |
| Output | |
| ['This', 'is', 'another', 'block', 'comment', 'that', 'spans', 'multiple', 'lines'] | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestStringTokenizer | Metodo: test_case_3 |
| Oracolo | |
| Restituisce una lista delle parole dei commenti multi-line | |

Tabella 2.56: Metodo della classe JavaTextMining

| Test case ID: TC_2.2_4 | Test Frame: CN4 |
|--|--------------------------------------|
| Input | |
| Parametro: s | " \"Testiamo le costanti stringhe\"" |
| Output | |
| [] | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestStringTokenizer | Metodo: test_case_4 |
| Oracolo | |
| Restituisce una lista vuota | |

Tabella 2.57: Metodo della classe JavaTextMining

| Test case ID: TC_2.2_5 | Test Frame: CN5 |
|---|--|
| Input | |
| Parametro: s | "string main() { // comment \n return Test passato} " |
| Output | |
| ["string", "main", "return"] | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestStringTokenizer | Metodo: test_case_5 |
| Oracolo | |
| Restituisce una lista con tutte e sole le parole all'interno della stringa di input, tranne dei commenti single-line | |

Tabella 2.58: Metodo della classe JavaTextMining

| Test case ID: TC_2.2_6 | Test Frame: CN6 |
|------------------------------|---|
| Input | |
| Parametro: s | "string main() { // comment \n return " \\Prova "} " |
| Output | |
| ["string", "main", "return"] | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestJavaTextMining SottoClasse: TestStringTokenizer | Metodo: test_case_6 |
| Oracolo | |
| Restituisce una lista con tutte e sole le parole all'interno della stringa di input, tranne dei commenti single-line | |

Tabella 2.59: Metodo della classe JavaTextMining

| | |
|---|------------------------|
| Test case ID: TC_2.3_1 | Test Frame: AC2 |
| Input | |
| Parametro: java_file | Non accessibile |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestRemoveComments | Metodo: test_case_1 |
| Oracolo | |
| PermissionError | |

Tabella 2.60: Metodo della classe JavaTextMining

| | |
|-------------------------------|----------------------------|
| Test case ID: TC_2.3_2 | Test Frame: AC1 CN1 |
| Input | |
| Parametro: java_file | "" |
| Output | |

| | |
|---|---------------------|
| " | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestRemoveComments | Metodo: test_case_2 |
| Oracolo | |
| Restituisce una stringa vuota | |

Tabella 2.61: Metodo della classe JavaTextMining

| | |
|---|---|
| Test case ID: TC_2.3_3 | Test Frame: AC1 CN2 |
| Input | |
| Parametro: java_file | 'public class Test { public static void main(String[] args) { System.out.println("Hello World"); } }' |
| Output | |
| 'public class Test { public static void main(String[] args) { System.out.println("Hello World"); } }' | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestRemoveComments | Metodo: test_case_3 |
| Oracolo | |
| Restituisce una stringa uguale al contenuto del file | |

Tabella 2.62: Metodo della classe JavaTextMining

| | |
|---|---|
| Test case ID: TC_2.3_4 | Test Frame: AC1 CN3 |
| Input | |
| Parametro: java_file | ''' \n /* This is a comment */ \n // Another comment \n ''' |
| Output | |
| ' \n \n // Another comment \n ' | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestRemoveComments | Metodo: test_case_4 |
| Oracolo | |
| Restituisce una stringa contenente il codice e i commenti single-line | |

Tabella 2.63: Metodo della classe JavaTextMining

| | |
|--|--|
| Test case ID: TC_2.3_5 | Test Frame: AC1 CN4 |
| Input | |
| Parametro: java_file | ''' \n public class Test { \n /* This is a comment */ \n public static void main(String[] args) { \n // Single line comment\n System.out.println("Hello World"); /* Inline comment */\n } \n }\n ''' |
| Output | |
| 'public class Test {n {n public static void main(String[] args) {n // Single line comment{n System.out.println("Hello World"); {n {n ' | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestJavaTextMining SottoClasse: TestRemoveComments | Metodo: test_case_5 |
| Oracolo | |
| Restituisce una stringa con solo il codice e commenti single-line | |

Tabella 2.64: Metodo della classe JavaTextMining

| | |
|--|------------------------|
| Test case ID: TC_2.4_1 | Test Frame: ST1 |
| Input | |
| Parametro: java_file_name | File inesistente |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestTakeJavaClass | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.65: Metodo della classe JavaTextMining

| | |
|-------------------------------|--------------------------------------|
| Test case ID: TC_2.4_2 | Test Frame: ST2 |
| Input | |
| Parametro: java_file_name | File esistente con permessi limitati |
| Output | |

| | |
|--|---------------------|
| PermissionError | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestTakeJavaClass | Metodo: test_case_2 |
| Oracolo | |
| PermissionError | |

Tabella 2.66: Metodo della classe JavaTextMining

| | |
|--|----------------------------|
| Test case ID: TC_2.4_3 | Test Frame: ST3 CN1 |
| Input | |
| Parametro: java_file_name | Contenuto file: "" |
| Output | |
| {} | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestTakeJavaClass | Metodo: test_case_3 |
| Oracolo | |
| Dizionario vuoto | |

Tabella 2.67: Metodo della classe JavaTextMining

| | |
|-------------------------------|----------------------------|
| Test case ID: TC_2.4_4 | Test Frame: ST3 CN2 |
| Input | |

| | |
|--|--------------------------------------|
| Parametro: java_file_name | Contenuto file: ""void public main"" |
| Output | |
| "void": 1, "public": 1, "main": 1 | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestTakeJavaClass | Metodo: test_case_4 |
| Oracolo | |
| Restituisce un dizionario con tutte le parole e frequenza 1 | |

Tabella 2.68: Metodo della classe JavaTextMining

| | |
|--|---------------------------------------|
| Test case ID: TC_2.4_5 | Test Frame: ST3 CN3 |
| Input | |
| Parametro: java_file_name | Contenuto file: 'word word2 word int' |
| Output | |
| "word": 2, "int": 1 | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestTakeJavaClass | Metodo: test_case_5 |
| Oracolo | |
| Restituisce un dizionario con tutte le parole e relativa frequenza | |

Tabella 2.69: Metodo della classe JavaTextMining

| | |
|--|---------------------------------|
| Test case ID: TC_2.4_6 | Test Frame: ST4 |
| Input | |
| Parametro: java_file_name | Fa riferimento ad una directory |
| Output | |
| IsADirectoryError | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestTakeJavaClass | Metodo: test_case_6 |
| Oracolo | |
| IsADirectoryError | |

Tabella 2.70: Metodo della classe JavaTextMining

| | |
|--|--------------------------------|
| Test case ID: TC_2.6_1 | Test Frame: CV1 CVA1 |
| Input | |
| Parametri: dict1, dict2 | [1,2,3], "non è un dizionario" |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.71: Metodo della classe JavaTextMining

| | |
|--|--|
| Test case ID: TC_2.6_2 | Test Frame: CV1 CVA2 |
| Input | |
| Parametri: dict1, dict2 | [1,2,3], {"word": 1, "public": 1, "main": 1} |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_2 |
| Oracolo | |
| TypeError | |

Tabella 2.72: Metodo della classe JavaTextMining

| | |
|--|----------------------|
| Test case ID: TC_2.6_3 | Test Frame: CV1 CVA3 |
| Input | |
| Parametri: dict1, dict2 | "ciao", {} |
| Output | |
| "ciao" | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_3 |

| Oracolo |
|---|
| Restituisce il valore del primo parametro |

Tabella 2.73: Metodo della classe JavaTextMining

| Test case ID: TC_2.6_4 | Test Frame: CV3 CVA1 |
|--|---------------------------|
| Input | |
| Parametri: dict1, dict2 | {}, "non è un dizionario" |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_4 |
| Oracolo | |
| TypeError | |

Tabella 2.74: Metodo della classe JavaTextMining

| Test case ID: TC_2.6_5 | Test Frame: CV3 CVA2 |
|-------------------------------------|---|
| Input | |
| Parametri: dict1, dict2 | {}, {"word": 1, "public": 1, "main": 1} |
| Output | |
| {"word": 1, "public": 1, "main": 1} | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_5 |
| Oracolo | |
| Restituire il valore del secondo parametro. | |

Tabella 2.75: Metodo della classe JavaTextMining

| | |
|--|-----------------------------|
| Test case ID: TC_2.6_6 | Test Frame: CV3 CVA3 |
| Input | |
| Parametri: dict1, dict2 | {}, {} |
| Output | |
| {} | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_6 |
| Oracolo | |
| Restituisce un dizionario vuoto. | |

Tabella 2.76: Metodo della classe JavaTextMining

| | |
|-------------------------------|---|
| Test case ID: TC_2.6_7 | Test Frame: CV2 CVA1 |
| Input | |
| Parametri: dict1, dict2 | {"word": 2, "int": 1}, "non sono un dizionario" |
| Output | |

| | |
|--|---------------------|
| TypeError | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_7 |
| Oracolo | |
| TypeError | |

Tabella 2.77: Metodo della classe JavaTextMining

| | |
|---|---|
| Test case ID: TC_2.6_8 | Test Frame: CV2 CVA2 VC1 |
| Input | |
| Parametri: dict1, dict2 | {"word": 2, "int": 1}, {"package": 1, "public": 1, "main": 1} |
| Output | |
| {"word": 2, "int": 1, "package": 1, "public": 1, "main": 1} | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_8 |
| Oracolo | |
| Restituire l'unione dei due dizionari. | |

Tabella 2.78: Metodo della classe JavaTextMining

| | |
|-------------------------------|---------------------------------|
| Test case ID: TC_2.6_9 | Test Frame: CV2 CVA2 VC2 |
| Input | |

| | |
|---|--|
| Parametri: dict1, dict2 | {"word": 2, "int": 1}, {"word": 1, "public": 1, "main": 1} |
| Output | |
| {"word": 3, "int": 1, "public": 1, "main": 1} | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_9 |
| Oracolo | |
| Restituire l'unione dei due dizionari, sommando le frequenze delle parole in comune. | |

Tabella 2.79: Metodo della classe JavaTextMining

| | |
|--|-----------------------------|
| Test case ID: TC_2.6_10 | Test Frame: CV2 CVA3 |
| Input | |
| Parametri: dict1, dict2 | {"word": 2, "int": 1}, {} |
| Output | |
| "word": 2, "int": 1 | |
| Test Driver | |
| Classe: JavaTextMining SottoClasse: TestMergeDict | Metodo: test_case_10 |
| Oracolo | |
| Restituisce il valore del primo dizionario. | |

Tabella 2.80: Metodo della classe JavaTextMining

| | |
|--|----------------------------|
| Test case ID: TC_2.7_1 | Test Frame: CN1 CD1 |
| Input | |
| Parametro: fake_dic | { } |
| Output | |
| { } | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_1 |
| Oracolo | |
| Restituisce un dizionario vuoto | |

Tabella 2.81: Metodo della classe JavaTextMining

| | |
|--|---|
| Test case ID: TC_2.7_2 | Test Frame: CN1 CD2 |
| Input | |
| Parametro: fake_dic | { 'CamelCase': 1, 'SplitThis': 2, 'AnotherTest': 3 } |
| Output | |
| { 'camel': 1, 'case': 1, 'split': 2, 'this': 2, 'another': 3, 'test': 3 } | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_2 |
| Oracolo | |

Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore

Tabella 2.82: Metodo della classe JavaTextMining

| Test case ID: TC_2.7_3 | Test Frame: CN1 CD3 |
|--|--|
| Input | |
| Parametro: fake_dic | {'lowercase': 1, 'uppercase': 2, 'allcaps': 3} |
| Output | |
| {'lowercase': 1, 'uppercase': 2, 'allcaps': 3} | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_3 |
| Oracolo | |
| Restituisce un dizionario uguale a quello in input | |

Tabella 2.83: Metodo della classe JavaTextMining

| Test case ID: TC_2.7_4 | Test Frame: CN1 CD4 |
|--|---|
| Input | |
| Parametro: fake_dic | {'CamelCaseKey': 4, 'lowercasekey': 2, 'AnotherCamelCase': 3} |
| Output | |
| {'camel': 7, 'case': 7, 'key': 4, 'lowercasekey': 2, 'another': 3} | |

| Test Driver | |
|---|---------------------|
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_4 |
| Oracolo | |
| Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore | |

Tabella 2.84: Metodo della classe JavaTextMining

| Test case ID: TC_2.7_5 | Test Frame: CN2 CD2 |
|---|--|
| Input | |
| Parametro: fake_dic | {'CamelCase': None, 'AnotherTest': 'string_value', 'validKey': 5} |
| Output | |
| {'camel': None, 'case': None, 'another': 'string_value', 'test': 'string_value', 'valid': 5, 'key': 5} | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_5 |
| Oracolo | |
| Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore | |

Tabella 2.85: Metodo della classe JavaTextMining

| Test case ID: TC_2.7_6 | Test Frame: CN2 CD3 |
|------------------------|---------------------|
| Input | |

| | |
|--|--|
| Parametro: fake_dic | {'lowercase': None, 'uppercase': 2, '2': 'ciao'} |
| Output | |
| {'lowercase': None, 'uppercase': 2, '2': 'ciao'} | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_6 |
| Oracolo | |
| Restituisce un dizionario uguale a quello in input | |

Tabella 2.86: Metodo della classe JavaTextMining

| | |
|--|---|
| Test case ID: TC_2.7_7 | Test Frame: CN2 CD4 |
| Input | |
| Parametro: fake_dic | {'CamelCase': None, 'AnotherCaseTest': 'string_value', 'validKey': 5} |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_7 |
| Oracolo | |
| TypeError | |

Tabella 2.87: Metodo della classe JavaTextMining

| Test case ID: TC_2.7_8 | Test Frame: CN3 |
|--|---|
| Input | |
| Parametro: fake_dic | {1: 2, 'AnotherCaseTest': 3, 'validKey': 5} |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_8 |
| Oracolo | |
| TypeError | |

Tabella 2.88: Metodo della classe JavaTextMining

| Test case ID: TC_2.7_9 | Test Frame: CN4 |
|--|---------------------|
| Input | |
| Parametro: fake_dic | ['ad', 1] |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestJavaTextMining SottoClasse: TestSplitDict | Metodo: test_case_9 |

| Oracolo |
|-----------|
| TypeError |

Tabella 2.89: Metodo della classe JavaTextMining**CSVWriter**

| Test case ID: TC_2.10_1 | Test Frame: CV1 |
|---------------------------------------|--|
| Input | |
| Parametri: filtered_dict, mining_dict | [1, 2, 3], {"commit1": {"classname": 3, "methodname": 2, "variablename": 1}, "commit2": {"classname": 0, "methodname": 1, "variablename": 0}} |
| Output | |
| AttributeError | |
| Test Driver | |
| Classe: TestCSVWriter | Metodo: test_case_1 |
| Oracolo | |
| AttributeError | |

Tabella 2.90: Metodo della classe CSVWriter

| Test case ID: TC_2.10_2 | Test Frame: CV2 CVA2 CN1 |
|-------------------------|--------------------------|
| Input | |

| | |
|--|--|
| Parametri: filtered_dict, mining_dict | { "className": 3, "methodName": 3, "variableName": 1 }, { "commit1": {"classname": 3, "methodname": 2, "variablename": 1}, "commit2": {"classname": 0, "methodname": 1, "variablename": 0} } |
| Output | |
| L'header contiene le colonne Name, className, methodName, variableName | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_1 |
| Oracolo | |
| L'header viene scritto correttamente (con Name e le chiavi) | |

Tabella 2.91: Metodo della classe CSVWriter

| | |
|--|---|
| Test case ID: TC_2.10_3 | Test Frame: CV2 CVA3 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {"className": 3, "methodName": 3, "variableName": 1}, {}} |
| Output | |
| L'header contiene le colonne Name, className, methodName, variableName | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_2 |

| Oracolo |
|---|
| L'header viene scritto correttamente (con Name e le chiavi) |

Tabella 2.92: Metodo della classe CSVWriter

| Test case ID: TC_2.10_4 | Test Frame: CV2 CVA1 CN1 |
|--|--|
| Input | |
| Parametri: filtered_dict, mining_dict | {"className": 3, "methodName": 3, "variableName": 1}, "prova con una variabile non dizionario" |
| Output | |
| L'header contiene le colonne Name, className, methodName, variableName | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_3 |
| Oracolo | |
| L'header viene scritto correttamente (con Name e le chiavi) | |

Tabella 2.93: Metodo della classe CSVWriter

| Test case ID: TC_2.10_5 | Test Frame: CV3 CVA2 CN1 |
|---------------------------------------|---|
| Input | |
| Parametri: filtered_dict, mining_dict | {}, {"commit1": {"classname": 3, "methodname": 2, "variablename": 1}, "commit2": {"classname": 0, "methodname": 1, "variablename": 0}} |

| Output | |
|--|---------------------|
| L'header contiene le colonne Name | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_4 |
| Oracolo | |
| L'header viene scritto correttamente (solo colonna Name) | |

Tabella 2.94: Metodo della classe CSVWriter

| Test case ID: TC_2.10_6 | Test Frame: CV3 CVA3 CN1 |
|--|--------------------------|
| Input | |
| Parametri: filtered_dict, mining_dict | {}, {} |
| Output | |
| L'header contiene le colonne Name | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_5 |
| Oracolo | |
| L'header viene scritto correttamente (solo colonna Name) | |

Tabella 2.95: Metodo della classe CSVWriter

| Test case ID: TC_2.10_7 | Test Frame: CV3 CVA1 CN1 |
|-------------------------|--------------------------|
|-------------------------|--------------------------|

| Input | |
|--|---|
| Parametri: filtered_dict, mining_dict | {}, "prova con una variabile non dizionario" |
| Output | |
| L'header contiene le colonne Name | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_6 |
| Oracolo | |
| L'header viene scritto correttamente (solo colonna Name) | |

Tabella 2.96: Metodo della classe CSVWriter

| Test case ID: TC_2.10_8 | Test Frame: CV3 CVA3 CN2 |
|---|--|
| Input | |
| Parametri: filtered_dict, mining_dict | {"className": 3, "methodName": 3, "variableName": 1}, {"commit1": {"classname": 3, "methodname": 2, "variablename": 1}} |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_7 |

| Oracolo |
|-----------------|
| PermissionError |

Tabella 2.97: Metodo della classe CSVWriter

| Test case ID: TC_2.10_9 | Test Frame: CV3 CVA3 CN3 |
|---|--|
| Input | |
| Parametri: filtered_dict, mining_dict | <pre>{"className": 3, "methodName": 3, "variableName": 1}, {"commit1": {"classname": 3, "methodname": 2, "variablename": 1}}</pre> |
| Output | |
| IOError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteHeader | Metodo: test_case_8 |
| Oracolo | |
| IOError | |

Tabella 2.98: Metodo della classe CSVWriter

| Test case ID: TC_2.11_1 | Test Frame: CV2 CVA2 CJ2 CN1 |
|-------------------------|------------------------------|
| Input | |

| | |
|---|---|
| Parametri: filtered_dict, mining_dict | { "className": 3, "methodName": 3, "variableName": 1 }, { "commit1": {"className": 3, "methodName": 2, "variableName": 1}, "commit2": {"className": 0, "methodName": 1, "variableName": 0} } |
| Output | |
| La riga viene scritto col nome del file e tutti 0 | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_1 |
| Oracolo | |
| La riga viene scritto col nome del file e tutti 0 | |

Tabella 2.99: Metodo della classe CSVWriter

| | |
|--|--|
| Test case ID: TC_2.11_2 | Test Frame: CV1 CVA2 CJ2 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {'package': 723, 'java': 16, 'apache': 32, 'sdk': 24, 'web': 1650, 'util': 986, 'import': 584, 'com': 180, 'static': 70}, { "commit1": {'package': 1, 'com': 18, 'apache': 5, 'sdk': 2, 'import': 23, 'java': 6, 'util': 13}, "commit2": {'package': 1, 'com': 18, 'apache': 5, 'sdk': 2, 'import': 23, 'java': 6, 'util': 13} } |
| Output | |
| La riga viene scritta col nome del file e le relative frequenze delle chiavi | |

| Test Driver | |
|--|---------------------|
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_2 |
| Oracolo | |
| La riga viene scritta col nome del file e le relative frequenze delle chiavi | |

Tabella 2.100: Metodo della classe CSVWriter

| Test case ID: TC_2.11_3 | Test Frame: CV1 CVA2 CJ3 CN1 |
|---|--|
| Input | |
| Parametri: filtered_dict, mining_dict | {'package': 723, 'java': 16, 'apache': 32, 'sdk': 24, 'web': 1650, 'util': 986, 'import': 584, 'com': 180, 'static': 70}, { "commit1": "non è corretto", "commit2": {'package': 1, 'com': 18, 'apache': 5, 'sdk': 2, 'import': 23, 'java': 6, 'util': 13} } |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_3 |
| Oracolo | |
| TypeError | |

Tabella 2.101: Metodo della classe CSVWriter

| | |
|---|--|
| Test case ID: TC_2.11_4 | Test Frame: CV1 CVA2 CJ1 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {'package': 723, 'java': 16, 'apache': 32, 'sdk': 24, 'web': 1650, 'util': 986, 'import': 584, 'com': 180, 'static': 70}, { "commit1": {}, "commit2": {'package': 1, 'com': 18, 'apache': 5, 'sdk': 2, 'import': 23, 'java': 6, 'util': 13} } |
| Output | |
| La riga viene scritta col nome del file e tutti 0 | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_4 |
| Oracolo | |
| La riga viene scritta col nome del file e tutti 0 | |

Tabella 2.102: Metodo della classe CSVWriter

| | |
|---------------------------------------|---|
| Test case ID: TC_2.11_5 | Test Frame: CV1 CVA3 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {'package': 723, 'java': 16, 'apache': 32, 'sdk': 24, 'web': 1650, 'util': 986, 'import': 584, 'com': 180, 'static': 70}, {"classname": 3, "methodname": 2, "variablename": 1} |
| Output | |

| | |
|---|---------------------|
| TypeError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_5 |
| Oracolo | |
| TypeError | |

Tabella 2.103: Metodo della classe CSVWriter

| | |
|---|---|
| Test case ID: TC_2.11_6 | Test Frame: CV1 CVA4 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | { "classname": 3, "methodname": 3, "variablename": 1 }, } |
| Output | |
| La riga non viene scritta. | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_6 |
| Oracolo | |
| La riga non viene scritta | |

Tabella 2.104: Metodo della classe CSVWriter

| | |
|--------------------------------|---------------------------------|
| Test case ID: TC_2.11_7 | Test Frame: CV1 CVA1 CN1 |
|--------------------------------|---------------------------------|

| Input | |
|---|---|
| Parametri: filtered_dict, mining_dict | { "classname": 3, "methodname": 3, "variablename": 1 }, [1, 2, 3] |
| Output | |
| AttributeError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_7 |
| Oracolo | |
| AttributeError | |

Tabella 2.105: Metodo della classe CSVWriter

| Test case ID: TC_2.11_8 | Test Frame: CV3 CVA2 CJ2 CN1 |
|---|---|
| Input | |
| Parametri: filtered_dict, mining_dict | {}, { "commit1": {"classname": 3, "methodname": 2, "variablename": 1}, "commit2": {"classname": 0, "methodname": 1, "variablename": 0} } |
| Output | |
| La riga non viene scritta | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_8 |

| Oracolo |
|---------------------------|
| La riga non viene scritta |

Tabella 2.106: Metodo della classe CSVWriter

| Test case ID: TC_2.11_9 | Test Frame: CV3 CVA2 CJ3 CN1 |
|---|--|
| Input | |
| Parametri: filtered_dict, mining_dict | <pre> {}, { "commit1": "non è corretto", "commit2": {'package': 1, 'com': 18, 'apache': 5, 'sdk': 2, 'import': 23, 'java': 6, 'util': 13} } </pre> |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_9 |
| Oracolo | |
| TypeError | |

Tabella 2.107: Metodo della classe CSVWriter

| Test case ID: TC_2.11_10 | Test Frame: CV3 CVA2 CJ1 CN1 |
|--------------------------|------------------------------|
| Input | |

| | |
|---|--|
| Parametri: filtered_dict, mining_dict | {}, { "commit1": {}, "commit2": {'package': 1, 'com': 18, 'apache': 5, 'sdk': 2, 'import': 23, 'java': 6, 'util': 13} } |
| Output | |
| La riga viene scritta con solo il nome del file | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_10 |
| Oracolo | |
| La riga viene scritta con solo il nome del file | |

Tabella 2.108: Metodo della classe CSVWriter

| | |
|---|---|
| Test case ID: TC_2.11_11 | Test Frame: CV3 CVA3 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {}, {"classname": 0, "methodname": 1, "variablename": 0} |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_11 |
| Oracolo | |

| |
|-----------|
| TypeError |
|-----------|

Tabella 2.109: Metodo della classe CSVWriter

| | |
|---|--------------------------|
| Test case ID: TC_2.11_12 | Test Frame: CV3 CVA4 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {}, {} |
| Output | |
| La riga non viene scritta | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_12 |
| Oracolo | |
| La riga non viene scritta | |

Tabella 2.110: Metodo della classe CSVWriter

| | |
|---------------------------------------|------------------------------|
| Test case ID: TC_2.11_13 | Test Frame: CV3 CVA1 CN1 |
| Input | |
| Parametri: filtered_dict, mining_dict | {}, "test non dizionario" |
| Output | |
| AttributeError | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_13 |
| Oracolo | |
| AttributeError | |

Tabella 2.111: Metodo della classe CSVWriter

| | |
|---|--|
| Test case ID: TC_2.11_14 | Test Frame: CV3 CVA2 CJ2 CN2 |
| Input | |
| Parametri: filtered_dict, mining_dict | {"classname": 3, "methodname": 3, "variablename": 1}, {"commit1": {"classname": 3, "methodname": 2, "variablename": 1}} |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_14 |
| Oracolo | |
| PermissionError | |

Tabella 2.112: Metodo della classe CSVWriter

| | |
|--------------------------|------------------------------|
| Test case ID: TC_2.11_15 | Test Frame: CV3 CVA2 CJ2 CN3 |
| Input | |

| | |
|---|--|
| Parametri: filtered_dict, mining_dict | {"classname": 3, "methodname": 3, "variablename": 1}, {"commit1": {"classname": 3, "methodname": 2, "variablename": 1}} |
| Output | |
| IOError | |
| Test Driver | |
| Classe: TestCSVWriter SottoClasse: TestWriteRows | Metodo: test_case_15 |
| Oracolo | |
| IOError | |

Tabella 2.113: Metodo della classe CSVWriter

2.1.3 Software Metrics

SoftwareMetrics

| | |
|-----------------------------|-------------------------|
| Test case ID: TC_5.1_1 | Test Frame: ED1 |
| Input | |
| Parametri: base_dir, text | "invalid_directory", "" |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |

| |
|-----------------------|
| FileNotFoundException |
|-----------------------|

Tabella 2.114: Metodo della classe SoftwareMetrics

| | |
|--|---------------------|
| Test case ID: TC_5.1_2 | Test Frame: ED2 CN1 |
| Input | |
| Parametri: base_dir, text | "" , "" |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestRemoveComments | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.115: Metodo della classe SoftwareMetrics

| | |
|---------------------------|--|
| Test case ID: TC_5.1_3 | Test Frame: ED2 CN2 |
| Input | |
| Parametri: base_dir, text | "/Dataset2/tests/SoftwareMetrics", ""/* Questo è un commento multilinea in Java che copre più righe */"" |
| Output | |
| FileNotFoundException | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestRemoveComments | Metodo: test_case_2 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.116: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.1_4 | Test Frame: ED2 CN3 |
| Input | |
| Parametri: base_dir, text | "/Dataset2/tests/SoftwareMetrics", "//Questo è un commento single-line in Java}" |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestRemoveComments | Metodo: test_case_3 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.117: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|----------------------------|
| Test case ID: TC_5.1_5 | Test Frame: ED2 CN4 |
| Input | |

| | |
|--|--|
| Parametri: base_dir, text | "/Dataset2/tests/SoftwareMetrics", "System.out.println(Questo non è un commento // ma parte della stringa);" |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestRemoveComments | Metodo: test_case_4 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.118: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.1_6 | Test Frame: ED2 CN5 |
| Input | |
| Parametri: base_dir, text | "/Dataset2/tests/SoftwareMetrics", "public class Test \n public void test() System.out.println(Hello World); \n " |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestRemoveComments | Metodo: test_case_5 |
| Oracolo | |

| |
|-----------------------|
| FileNotFoundException |
|-----------------------|

Tabella 2.119: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.1_7 | Test Frame: ED2 CN6 |
|--|---|
| Input | |
| Parametri: base_dir, text | "/Dataset2/tests/SoftwareMetrics", ""public class Test { // Questo è un commento single-line public void test() { /* Questo è un commento multilinea */ System.out.println("Hello World"); } /** Questo è un commento docstring */ }"" |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestRemoveComments | Metodo: test_case_6 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.120: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.2_1 | Test Frame: ED1 |
|------------------------|---------------------|
| Input | |
| Parametro: base_dir | "invalid_directory" |
| Output | |

| | |
|-----------------------------|---------------------|
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.121: Metodo della classe SoftwareMetrics

| | |
|---|---------------------------------------|
| Test case ID: TC_5.2_2 | Test Frame: ED2 FC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" |
| Output | |
| Restituisce 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_1 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.122: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|------------------------------------|
| Test case ID: TC_5.2_3 | Test Frame: ED2 FC2 CN1 SC2 |
| Input | |

| | |
|--|--|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "if () { " |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_2 |
| Oracolo | |
| Restituisce il livello di annidamento massimo dei controlli che rileva | |

Tabella 2.123: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.2_4 | Test Frame: ED2 FC3 CN1 SC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = 'a'; @InvalidToken" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_3 |
| Oracolo | |
| Restituisce il livello di annidamento massimo dei controlli che rileva | |

Tabella 2.124: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.2_5 | Test Frame: ED2 FC4 CN1 SC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = 0;" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_4 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.125: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.2_6 | Test Frame: ED2 FC4 CN1 SC2 CE1 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "if (x > 0) { x++; }" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_5 |
| Oracolo | |

Restituisce il livello di annidamento massimo dell'if che rileva

Tabella 2.126: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.2_7 | Test Frame: ED2 FC4 CN1 SC2 CE2 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""if (x > 0) { x++; } else { x--; }"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_6 |
| Oracolo | |
| Restituisce il livello di annidamento massimo dell'if o else che rileva | |

Tabella 2.127: Metodo della classe SoftwareMetrics

| | |
|-----------------------------------|--|
| Test case ID: TC_5.2_8 | Test Frame: ED2 FC4 CN1 SC2 CE2 AN2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""if (x > 0) { x--; } else { if (y > 0) { y++; } }"" |
| Output | |
| 2 | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_7 |
| Oracolo | |
| Restituisce il livello di annidamento massimo dell'if o else che rileva | |

Tabella 2.128: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.2_9 | Test Frame: ED2 FC4 CN1 SC2 CE3 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""if (x > 0) x++; else x-;"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_8 |
| Oracolo | |
| Restituisce il livello di annidamento massimo individuato, non considerando l'else come livello | |

Tabella 2.129: Metodo della classe SoftwareMetrics

| | |
|--------------------------------|--|
| Test case ID: TC_5.2_10 | Test Frame: ED2 FC4 CN1 SC2 CE3 AN2 |
| Input | |

| | |
|---|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""if (x > 0) { y++; } else if (y > 0) x--; "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_9 |
| Oracolo | |
| Restituisce il livello di annidamento massimo individuato, non considerando l'else come livello | |

Tabella 2.130: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.2_11 | Test Frame: ED2 FC4 CN1 SC3 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""for (int i = 0; i < 10; i++) { System.out.println(i); }"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_10 |
| Oracolo | |

Restituisce il livello di annidamento massimo del for che rileva

Tabella 2.131: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.2_12 | Test Frame: ED2 FC4 CN1 SC4 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""while (x < 10) { x++; }"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_11 |
| Oracolo | |
| Restituisce il livello di annidamento massimo del while che rileva | |

Tabella 2.132: Metodo della classe SoftwareMetrics

| | |
|-----------------------------------|---|
| Test case ID: TC_5.2_13 | Test Frame: ED2 FC4 CN1 SC5 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""do { x++; } while (x < 10);"" |
| Output | |
| 1 | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_12 |
| Oracolo | |
| Restituisce il livello di annidamento massimo del do che rileva | |

Tabella 2.133: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.2_14 | Test Frame: ED2 FC4 CN1 SC6 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""switch (x) { case 1: x++; break; case 2: x-; break; }"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_13 |
| Oracolo | |
| Restituisce il livello di annidamento massimo del switch che rileva | |

Tabella 2.134: Metodo della classe SoftwareMetrics

| | |
|--------------------------------|--|
| Test case ID: TC_5.2_15 | Test Frame: ED2 FC4 CN1 SC7 AN1 |
| Input | |

| | |
|---|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""try { x++; } catch (Exception e) { x--; }"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_14 |
| Oracolo | |
| Restituisce il livello di annidamento massimo del try che rileva | |

Tabella 2.135: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.2_16 | Test Frame: ED2 FC4 CN1 SC8 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""if (x > 0) { x++; } for (int i = 0; i < 10; i++) { x--; } while (x < 5) { x++; } do { x--; } while (x > 0); switch (x) { case 1: x++; break; } try { x++; } catch (Exception e) { x--; } "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_15 |
| Oracolo | |

Restituisce il livello di annidamento massimo dei controlli che rileva

Tabella 2.136: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.2_17 | Test Frame: ED2 FC4 CN1 SC8 AN2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" if (x > 0) { for (int i = 0; i < 10; i++) { if (x%2 == 0) { while (x < 5) { x++; } } } else { switch (x) { case 1: try { x++; } catch (Exception e) { x-; } break; } } "" |
| Output | |
| 4 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_16 |
| Oracolo | |
| Restituisce il livello di annidamento massimo dei controlli che rileva | |

Tabella 2.137: Metodo della classe SoftwareMetrics

| | |
|--------------------------------|--|
| Test case ID: TC_5.2_18 | Test Frame: ED2 FC4 CN1 SC8 AN3 |
| Input | |

| | |
|--|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" if (x > 0) { for (int i = 0; i < 10; i++) { x++; } if (x > 1) { while (x < 5) { x--; } } else { switch (x) { case 1: try { x++; } catch (Exception e) { x--; } break; } }"" |
| Output | |
| 3 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_17 |
| Oracolo | |
| Restituisce il livello di annidamento massimo dei controlli che rileva | |

Tabella 2.138: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.2_19 | Test Frame: ED2 FC4 CN2 SC2 CE2 AN1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" if (x > 0) { /* commento */ x++; } else { x--; }"" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCalculateMaxNesting | Metodo: test_case_18 |

| Oracolo |
|--|
| Restituisce il livello di annidamento massimo dei controlli che rileva |

Tabella 2.139: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.3_1 | Test Frame: ED1 |
|-----------------------------|---------------------|
| Input | |
| Parametro: base_dir | "invalid_directory" |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.140: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.3_2 | Test Frame: ED2 FC1 |
|-----------------------------------|---------------------------------------|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" |
| Output | |
| Restituisce 0 | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_1 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.141: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.3_3 | Test Frame: ED2 FC2 TD4 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "public class { int x = 10 }" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_2 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni rilevate correttamente | |

Tabella 2.142: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--------------------------------|
| Test case ID: TC_5.3_4 | Test Frame: ED2 FC3 TD4 |
| Input | |

| | |
|--|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "public class MyClass { int x = 'text' @#InvalidToken }" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_3 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni rilevate correttamente | |

Tabella 2.143: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.3_5 | Test Frame: ED2 FC4 TD1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "System.out.println('Hello World');" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_4 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.144: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.3_6 | Test Frame: ED2 FC4 TD2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "public class MyClass {}" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_5 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di classi | |

Tabella 2.145: Metodo della classe SoftwareMetrics

| | |
|-----------------------------------|---|
| Test case ID: TC_5.3_7 | Test Frame: ED2 FC4 TD3 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "public interface NomeInterfaccia {}" |
| Output | |
| 1 | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_6 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di interfacce | |

Tabella 2.146: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.3_8 | Test Frame: ED2 FC4 TD4 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public class MyClass { public void myMethod() {} } "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_7 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di classi e metodi | |

Tabella 2.147: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--------------------------------|
| Test case ID: TC_5.3_9 | Test Frame: ED2 FC4 TD5 |
| Input | |

| | |
|---|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public class MyClass { private int x = 10; } "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_8 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di variabili | |

Tabella 2.148: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.3_10 | Test Frame: ED2 FC4 TD6 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" @Option public class MyClass {} "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_9 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di annotazioni | |

Tabella 2.149: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.3_11 | Test Frame: ED2 FC4 TD7 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public static final int MAX_VALUE = 100; "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_10 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di parole chiave | |

Tabella 2.150: Metodo della classe SoftwareMetrics

| | |
|-----------------------------------|--|
| Test case ID: TC_5.3_12 | Test Frame: ED2 FC4 TD8 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" import java.util.List; "" |
| Output | |
| 1 | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_11 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazioni di import | |

Tabella 2.151: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.3_13 | Test Frame: ED2 FC4 TD9 |
| Input | |
| Parametro: base_dir, file_content | <pre>"/Dataset2/tests/SoftwareMetrics", "" import java.util.List; @Deprecated public class MyClass { private int x; public void myMethod() {} } public interface NomeInterfaccia { void metodoAstratto(); }""</pre> |
| Output | |
| 7 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountDeclarativeLines | Metodo: test_case_12 |
| Oracolo | |
| Restituisce il numero di linee di dichiarazione | |

Tabella 2.152: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|------------------------|
| Test case ID: TC_5.4_1 | Test Frame: ED1 |
| Input | |
| Parametro: base_dir | "invalid_directory" |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.153: Metodo della classe SoftwareMetrics

| | |
|--|---------------------------------------|
| Test case ID: TC_5.4_2 | Test Frame: ED2 GF1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" |
| Output | |
| Restituisce 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_1 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.154: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.4_3 | Test Frame: ED2 GF2 FS1 CF1 PC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = ;" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_2 |
| Oracolo | |
| Restituisce il numero di righe con codice della stringa data in input | |

Tabella 2.155: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.4_4 | Test Frame: ED2 GF2 FS2 CF1 PC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = 'string' @#InvalidToken" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_3 |
| Oracolo | |

Restituisce il numero di righe con codice della stringa data in input

Tabella 2.156: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.4_5 | Test Frame: ED2 GF2 FS3 CF1 PC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = 10;" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_4 |
| Oracolo | |
| Restituisce il numero di righe con codice della stringa data in input | |

Tabella 2.157: Metodo della classe SoftwareMetrics

| | |
|-----------------------------------|---|
| Test case ID: TC_5.4_6 | Test Frame: ED2 GF2 FS3 CF2 PC2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "// Questo è un commento" |
| Output | |
| 0 | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_5 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.158: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.4_7 | Test Frame: ED2 GF3 FS3 CF1 PC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" int x = 10; int y = 20; "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_6 |
| Oracolo | |
| Restituisce il numero di righe con codice della stringa data in input | |

Tabella 2.159: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--|
| Test case ID: TC_5.4_8 | Test Frame: ED2 GF3 FS3 CF2 PC1 |
| Input | |

| | |
|---|--|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" // Questo è un commento int x = 10; "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_7 |
| Oracolo | |
| Restituisce il numero di righe con codice della stringa data in input | |

Tabella 2.160: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.4_9 | Test Frame: ED2 GF3 FS3 CF2 PC2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" // Commento 1 // Commento 2 "" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_8 |
| Oracolo | |

| |
|---------------|
| Restituisce 0 |
|---------------|

Tabella 2.161: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.4_10 | Test Frame: ED2 GF3 FS3 CF3 PC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" /* Questo è un commento multilinea */ int x = 10; "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_9 |
| Oracolo | |
| Restituisce il numero di righe con codice della stringa data in input | |

Tabella 2.162: Metodo della classe SoftwareMetrics

| | |
|-----------------------------------|--|
| Test case ID: TC_5.4_11 | Test Frame: ED2 GF3 FS3 CF3 PC2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" /* Commento multilinea */ "" |
| Output | |

| | |
|--|----------------------|
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_10 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.163: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.4_12 | Test Frame: ED2 GF3 FS3 CF4 PC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ,,,,, int x = 10; int y = 20; ,,,,, |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountLinesOfCode | Metodo: test_case_11 |
| Oracolo | |
| Restituisce il numero di righe con codice della stringa data in input | |

Tabella 2.164: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|------------------------|
| Test case ID: TC_5.5_1 | Test Frame: ED1 |
| Input | |
| Parametro: base_dir | "invalid_directory" |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.165: Metodo della classe SoftwareMetrics

| | |
|---|---------------------------------------|
| Test case ID: TC_5.5_2 | Test Frame: ED2 FC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" |
| Output | |
| Restituisce 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_1 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.166: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.5_3 | Test Frame: ED2 FC2 TD2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", ""void myMethod({" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_2 |
| Oracolo | |
| Restituisce il numero di metodi individuati correttamente | |

Tabella 2.167: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.5_4 | Test Frame: ED2 FC3 TD2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "void myMethod() { return @InvalidToken 'string'; }" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_3 |

| Oracolo |
|---|
| Restituisce il numero di metodi individuati correttamente |

Tabella 2.168: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.5_5 | Test Frame: ED2 FC4 TD1 |
|---|---|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = 10;" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_4 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.169: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.5_6 | Test Frame: ED2 FC4 TD2 |
|-----------------------------------|---|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", """" public void method1() {} private int method2() { return 0; } """" |
| Output | |

| | |
|---|---------------------|
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_5 |
| Oracolo | |
| Restituisce il numero di metodi individuati correttamente | |

Tabella 2.170: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.5_7 | Test Frame: ED2 FC4 TD3 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public MyClass() {} public MyClass(int x) { this.x = x; } "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_6 |
| Oracolo | |
| Restituisce il numero di metodi individuati correttamente | |

Tabella 2.171: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--------------------------------|
| Test case ID: TC_5.5_8 | Test Frame: ED2 FC4 TD4 |
|-------------------------------|--------------------------------|

| Input | |
|---|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" Runnable r = () -> System.out.println("Hello"); Callable<Integer> c = () -> 42; "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_7 |
| Oracolo | |
| Restituisce il numero di metodi individuati correttamente | |

Tabella 2.172: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.5_9 | Test Frame: ED2 FC4 TD5 |
|---|--|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" Runnable r = new Runnable() { public void run() { System.out.println("Running"); } }; "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_8 |

| Oracolo |
|---|
| Restituisce il numero di metodi individuati correttamente |

Tabella 2.173: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.5_10 | Test Frame: ED2 FC4 TD6 |
|---|--|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public MyClass() {} public void method1() {} Runnable r = () -> System.out.println("Hello"); "" |
| Output | |
| 3 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountMethodsDeclaration | Metodo: test_case_9 |
| Oracolo | |
| Restituisce il numero di metodi individuati correttamente | |

Tabella 2.174: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.6_1 | Test Frame: ED1 |
|------------------------|---------------------|
| Input | |
| Parametro: base_dir | "invalid_directory" |
| Output | |

| | |
|-----------------------------|---------------------|
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.175: Metodo della classe SoftwareMetrics

| | |
|---|---------------------------------------|
| Test case ID: TC_5.6_2 | Test Frame: ED2 FC1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" |
| Output | |
| Restituisce 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_1 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.176: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--------------------------------|
| Test case ID: TC_5.6_3 | Test Frame: ED2 FC2 TD2 |
| Input | |

| | |
|---|---|
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "class MyClass {" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_2 |
| Oracolo | |
| Restituisce il numero di classi individuate correttamente | |

Tabella 2.177: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.6_4 | Test Frame: ED2 FC3 TD1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "int x = 'string'; @InvalidToken" |
| Output | |
| 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_3 |
| Oracolo | |
| Restituisce il numero di classi individuate correttamente | |

Tabella 2.178: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.6_5 | Test Frame: ED2 FC4 TD1 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "import java.util.List;" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_4 |
| Oracolo | |
| Restituisce 0 | |

Tabella 2.179: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.6_6 | Test Frame: ED2 FC4 TD2 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", """" public class MyClass { private int x; } """" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_5 |
| Oracolo | |

| |
|--|
| Restituisce il numero di classi dichiarate |
|--|

Tabella 2.180: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.6_7 | Test Frame: ED2 FC4 TD3 |
|---|---|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public interface MyInterface { void method1(); } "" |
| Output | |
| 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_6 |
| Oracolo | |
| Restituisce il numero di interfacce dichiarate | |

Tabella 2.181: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.6_8 | Test Frame: ED2 FC4 TD4 |
|-----------------------------------|---|
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public class MyClass { Runnable r = new Runnable() { public void run() { System.out.println("Running"); } } } "" |
| Output | |

| | |
|---|---------------------|
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_7 |
| Oracolo | |
| Restituisce il numero di classi anonime dichiarate | |

Tabella 2.182: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.6_9 | Test Frame: ED2 FC4 TD5 |
| Input | |
| Parametro: base_dir, file_content | "/Dataset2/tests/SoftwareMetrics", "" public class OuterClass { private class InnerClass { private int y; } } "" |
| Output | |
| 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_8 |
| Oracolo | |
| Restituisce il numero di classi dichiarate | |

Tabella 2.183: Metodo della classe SoftwareMetrics

| | |
|--------------------------------|--------------------------------|
| Test case ID: TC_5.6_10 | Test Frame: ED2 FC4 TD6 |
|--------------------------------|--------------------------------|

| Input | |
|---|--|
| Parametro: base_dir, file_content | <pre> "/Dataset2/tests/SoftwareMetrics", "" // Dichiarazione di classe public class OuterClass { // Classe nidificata private class InnerClass { private int y; } // Classe anonima Runnable r = new Runnable() { public void run() { System.out.println("Running"); } }; // Dichiarazione di metodo public void outerMethod() { // Metodo che utilizza una classe anonima Callable<Integer> c = new Callable<Integer>() { public Integer call() { return 42; } }; } // Dichiarazione di interfaccia public interface MyInterface { void myMethod(); } "" </pre> |
| Output | |
| 5 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestCountClassDeclaration | Metodo: test_case_9 |
| Oracolo | |
| Restituisce il numero di classi di ogni tipo dichiarate | |

Tabella 2.184: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.8_1 | Test Frame: ED1 |
|------------------------|-----------------|
| Input | |

| | |
|-----------------------------|---------------------|
| Parametro: base_dir | "invalid_directory" |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.185: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.8_2 | Test Frame: ED2 EP2 FC1 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "" |
| Output | |
| Restituisce 0 come massimo e somma della cyclomatic complexity | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_1 |
| Oracolo | |
| Restituisce 0 come massimo e somma della cyclomatic complexity | |

Tabella 2.186: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.8_3 | Test Frame: ED2 EP2 FC2 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "public class Test {" |
| Output | |
| "Errore nell'analisi del file existing_file.java: Il file non presenta una sintassi java valida." | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_2 |
| Oracolo | |
| Il log presenta un messaggio di errore | |

Tabella 2.187: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.8_4 | Test Frame: ED2 EP2 FC3 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", """"public class Test { void method() { int x = 10 @#invalid_token } }"""" |
| Output | |
| "Errore nell'analisi del file existing_file.java: Il file presenta un carattere o una sequenza di caratteri non valida." | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_3 |
| Oracolo | |
| Il log presenta un messaggio di errore | |

Tabella 2.188: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.8_5 | Test Frame: ED2 EP2 FC4 NF1 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "public class Test {}" |
| Output | |
| 0 come massimo e 0 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_4 |
| Oracolo | |
| Restituisce 0 come massimo e somma della cyclomatic complexity | |

Tabella 2.189: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--|
| Test case ID: TC_5.8_6 | Test Frame: ED2 EP2 FC4 NF2 CC1 |
| Input | |

| | |
|--|--|
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "public class Test { public void method() { int x = 'string'; } }" |
| Output | |
| 1 come massimo e 1 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_5 |
| Oracolo | |
| Restituisce 1 come massimo e somma della cyclomatic complexity | |

Tabella 2.190: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.8_7 | Test Frame: ED2 EP2 FC4 NF2 CC2 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "public class Test { public void method() { if (true) {} } }" |
| Output | |
| 2 come massimo e 2 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_6 |

| Oracolo |
|--|
| Restituisce 2 come massimo e somma della cyclomatic complexity |

Tabella 2.191: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.8_8 | Test Frame: ED2 EP2 FC4 NF2 CC3 |
|--|--|
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "" public class Test { public void method() { if (true) { while (true) {} } } } "" |
| Output | |
| 3 come somma e 3 come massimo | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_7 |
| Oracolo | |
| Restituisce il numero di livelli di controllo | |

Tabella 2.192: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.8_9 | Test Frame: ED2 EP2 FC4 NF3 CC1 |
|------------------------|---------------------------------|
| Input | |

| | |
|---|---|
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "" public class Test { public void method1() {} public void method2() {} } "" |
| Output | |
| 1 come massimo e 2 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_8 |
| Oracolo | |
| Restituisce 1 come massimo e come somma il numero di funzioni | |

Tabella 2.193: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.8_10 | Test Frame: ED2 EP2 FC4 NF3 CC2 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "" public class Test { public void method1() { if (true) {} } public void method2() { if (true) {} } } "" |
| Output | |
| 2 come massimo e 4 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_9 |

| Oracolo |
|---|
| Restituisce la massimo cyclomati complexity tra le funzioni e la loro somma |

Tabella 2.194: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.8_11 | Test Frame: ED2 EP2 FC4 NF3 CC3 |
|---|---|
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "" public class Test { public void method1() { if (true) { while (true) {} } } public void method2() { for (int i = 0; i < 10; i++) {} if (x==0) x++; } } "" |
| Output | |
| 3 come massimo e 6 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_10 |
| Oracolo | |
| Restituisce la massimo cyclomati complexity tra le funzioni e la loro somma | |

Tabella 2.195: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.8_12 | Test Frame: ED2 EP1 FC4 NF2 CC3 |
|-------------------------|---------------------------------|
| Input | |

| | |
|---|---|
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "" "" public class Test { public void method() { if (true) { while (true) {} } } } "" |
| Output | |
| 3 come massimo e 3 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_11 |
| Oracolo | |
| Restituisce la massimo cyclomati complexity tra le funzioni e la loro somma | |

Tabella 2.196: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.8_13 | Test Frame: ED2 EP1 FC4 NF2 CC3 |
| Input | |
| Parametro: base_dir, file_path, file_content | "/Dataset2/tests/SoftwareMetrics", "existing_file.java", "" public class Test { public void method() { if (true) { while (true) {} } } } "" |
| Output | |
| 3 come massimo e 3 come somma | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestAnalyzer | Metodo: test_case_12 |

| Oracolo |
|---|
| Restituisce la massimo cyclomati complexity tra le funzioni e la loro somma |

Tabella 2.197: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.7_1 | Test Frame: L1 S1 N3 CC3 PC7 PB1 PS1 PTH1 PE1 PL1 PT1 PF2 NCH3 PCH2 TS1 NS1 NC2 PCS1 PBK2 O2 |
|---|--|
| Input | |
| Parametri: file_content | ''' public class Main { int §; }''' |
| Output | |
| LexerError | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_1 |
| Oracolo | |
| LexerError | |

Tabella 2.198: Metodo della classe SoftwareMetrics

| Test case ID: TC_5.7_2 | Test Frame: L1 S2 |
|-------------------------|---------------------------------|
| Input | |
| Parametri: file_content | " public class Main { int +; }" |
| Output | |

| | |
|--|---------------------|
| SyntaxError | |
| Test Driver | |
| Classe: TestSoftwareMetrics Sottoclasse: TestComputeEssentialComplexity | Metodo: test_case_2 |
| Oracolo | |
| SyntaxError | |

Tabella 2.199: Metodo della classe SoftwareMetrics

| | |
|---|-----------------------------|
| Test case ID: TC_5.7_3 | Test Frame: L1 S1 N1 |
| Input | |
| Parametri: file_content | " public class Main { }" |
| Output | |
| Max Essential Complexity = 0, Sum Essential Complexity = 0 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_3 |
| Oracolo | |
| Max Essential Complexity = 0, Sum Essential Complexity = 0 | |

Tabella 2.200: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.7_4 | Test Frame: L1 S1 N3 CC3 PC7 PB1 PS1 PTH1 PE1 PL1 PT1 PF2 NCH3 PCH2 TS1 NS1 NC2 PCS1 PBK2 O2 |
| Input | |
| <p>Il file non contiene blocchi singoli di codice e include un'istruzione sincronizzata</p> <p>È presente un if con un then non riducibile e un'istruzione else non riducibile.</p> <p>È presente un ciclo while non riducibile</p> <p>È presente un corpo try non riducibile e un solo catch con uno statement non riducibile</p> <p>È presente uno switch che include più di un caso, almeno uno dei quali contiene un'istruzione non riducibile, con break presente solo in un caso.</p> <p>Il flusso di controllo degli statement è sequenziale.</p> | |
| Output | |
| Max Essential Complexity = 10, Sum Essential Complexity = 12 | |
| Test Driver | |
| Classe: TestSoftwareMetrics Sottoclasse: TestComputeEssentialComplexity | Metodo: test_case_4 |
| Oracolo | |
| Max Essential Complexity = 10, Sum Essential Complexity = 12 | |

Tabella 2.201: Metodo della classe SoftwareMetrics

| | |
|------------------------|--------------------------------------|
| Test case ID: TC_5.1_5 | Test Frame: L1 S1 N2 CC1 PC8 PB1 PS2 |
| Input | |

| | |
|---|--|
| Parametri: file_content | " public class Main { int var1, var2; public void setVar1(int var1){ this.var1 = var1; } public int method1(){ int a = 1; int b = 0; { a = b + 1; b = a + 2*b; System.out.println(a); } return b; } } " |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_5 |
| Oracolo | |
| Max Essential Complexity = 1, Sum Essential Complexity = 2 | |

Tabella 2.202: Metodo della classe SoftwareMetrics

| | |
|-------------------------|--|
| Test case ID: TC_5.1_6 | Test Frame: L1 S1 N2 CC2 PC1 PB2 PS2 PTH2 PE2 O1 |
| Input | |
| Parametri: file_content | " public class Main { int var1, var2; public Main(int var1, int var2){ if(var1>var2) var1=var1+var2; else var2=var1+var2; this.var1=var1; this.var2=var2; } } " |
| Output | |

| | |
|---|---------------------|
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_6 |
| Oracolo | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |

Tabella 2.203: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.1_7 | Test Frame: L1 S1 N2 CC1 PC1 PB1 PS2 PTH1 PE1 NI1 O1 |
| Input | |
| Parametri: file_content | <pre>" public class Main { int var1, var2; public int method_1(){ int a, b; if(a>b){ a=a-b; setVar(a); return a; } else{ b=b+1; return b; } } public void setVar1(int var){ this.var1 = var; } }"</pre> |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_7 |
| Oracolo | |

Max Essential Complexity = 1, Sum Essential Complexity = 2

Tabella 2.204: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.1_8 | Test Frame: L1 S1 N2 CC1 PC1 PB1 PS2 PTH1 PE1 NI2 O1 |
| Input | |
| Parametri: file_content | <pre>public class Main { int var1, var2; public int method_1(){ int a, b; if(a > b){ a = a - b; setVar(a); return a; } if(a == b){ return b; } else { setVar(b); return b - a; } } public void setVar1(int var){ this.var1 = var; } }</pre> |
| Output | |
| Max Essential Complexity = 3, Sum Essential Complexity = 4 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_8 |
| Oracolo | |
| Max Essential Complexity = 3, Sum Essential Complexity = 4 | |

Tabella 2.205: Metodo della classe SoftwareMetrics

| | |
|-------------------------------|--|
| Test case ID: TC_5.1_9 | Test Frame: L1 S1 N2 CC1 PC2 PB1 PS2 PL2 O1 |
| Input | |

| | |
|---|---------------------|
| Il file include solo metodi e presenta cicli while con corpo del ciclo riducibile. Gli statement di controllo sono organizzati in modo sequenziale. | |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 3 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_9 |
| Oracolo | |
| Max Essential Complexity = 1, Sum Essential Complexity = 3 | |

Tabella 2.206: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.1_10 | Test Frame: L1 S1 N2 CC1 PC3 PB1 PS2 PL2 O1 |
| Input | |
| Il file include solo metodi e presenta cicli for con corpo del ciclo riducibile. Gli statement di controllo sono organizzati in modo sequenziale. | |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 2 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_10 |
| Oracolo | |

Max Essential Complexity = 1, Sum Essential Complexity = 2

Tabella 2.207: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.1_11 | Test Frame: L1 S1 N2 CC1 PC4 PB1 PS2 PL2 O1 |
| Input | |
| Il file include solo metodi e presenta cicli do while con corpo del ciclo riducibile. Gli statement di controllo sono organizzati in modo sequenziale. | |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_11 |
| Oracolo | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |

Tabella 2.208: Metodo della classe SoftwareMetrics

| | |
|-------------------------|--|
| Test case ID: TC_5.1_12 | Test Frame: L1 S1 N2 CC1 PC5 PB1 PS2 NS2 NC1 PCS2 PBK1 O1 |
| Input | |

| | |
|---|---|
| Parametri: file_content | ''' public class Main { int var1, var2; public int method_1(int a, int b){ switch(a){ case 1: return a; } switch(b){ case 2: return b; } } } ''' |
| Output | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_12 |
| Oracolo | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |

Tabella 2.209: Metodo della classe SoftwareMetrics

| | |
|--|---|
| Test case ID: TC_5.1_13 | Test Frame: L1 S1 N2 CC1 PC5 PB1 PS2 NS1 NC2 PCS1 PBK2 O1 |
| Input | |
| Parametri: file_content | ''' public class Main { int var1, var2; public int method_1(int a, int b){ switch(a){ case 1: a++; case 2: System.out.println(a); break; case 3: b++; case 4: return b; } } } ''' |
| Output | |
| Max Essential Complexity = 5, Sum Essential Complexity = 5 | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_13 |
| Oracolo | |
| Max Essential Complexity = 5, Sum Essential Complexity = 5 | |

Tabella 2.210: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.1_14 | Test Frame: L1 S1 N2 CC1 PC5 PB1 PS2 NS1 NC2 PCS1 PBK3 O1 |
| Input | |
| Parametri: file_content | <pre>''' public class Main { int var1, var2; public int method_1(int a, int b){ switch(a){ case 1: a++; break; case 2: System.out.println(a); break; case 3: b++; break; case 4: a++; return a; break; } return b; } } '''</pre> |
| Output | |
| Max Essential Complexity = 5, Sum Essential Complexity = 5 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_14 |
| Oracolo | |
| Max Essential Complexity = 5, Sum Essential Complexity = 5 | |

Tabella 2.211: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.1_15 | Test Frame: L1 S1 N2 CC1 PC5 PB1 PS2 NS1 NC2 PCS2 PBK2 O1 |
| Input | |
| Parametri: file_content | ''' public class Main { int var1, var2; public int method_1(int a, int b){ switch(a){ case 1: a++; case 2: System.out.println(a); case 3: b++; case 4: System.out.println(b); break; } return b; } } ''' |
| Output | |
| Max Essential Complexity = 5, Sum Essential Complexity = 5 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_15 |
| Oracolo | |
| Max Essential Complexity = 5, Sum Essential Complexity = 5 | |

Tabella 2.212: Metodo della classe SoftwareMetrics

| | |
|-------------------------|--|
| Test case ID: TC_5.1_16 | Test Frame: L1 S1 N2 CC1 PC5 PB1 PS2 NS1 NC2 PCS2 PBK3 O1 |
| Input | |

| | |
|---|--|
| Parametri: file_content | <pre>''' public class Main { int var1, var2; public int method_1(int a, int b){ switch(a){ case 1: a++; break; case 2: System.out.println(a); break; case 3: b++; break; case 4: System.out.println(b); break; } return b; } } '''</pre> |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_16 |
| Oracolo | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |

Tabella 2.213: Metodo della classe SoftwareMetrics

| | |
|-------------------------|--|
| Test case ID: TC_5.1_17 | Test Frame: L1 S1 N2 CC1 PC6 PB1 PS2 PT2 PF3 NCH1 O1 |
| Input | |
| Parametri: file_content | <pre>''' public class Main { int var1, var2; public int method_1(int a, int b){ try{ File myObj = new File("filename.txt"); Scanner myReader = new Scanner(myObj); String data = myReader.nextLine(); } finally { while(a<1) if(b) break; } } } '''</pre> |
| Output | |

| | |
|---|----------------------|
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_17 |
| Oracolo | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |

Tabella 2.214: Metodo della classe SoftwareMetrics

| | |
|--|--|
| Test case ID: TC_5.1_18 | Test Frame: L1 S1 N2 CC1 PC6 PB1 PS2 PT1 PF1 NCH3 PCH1 O1 |
| Input | |
| Parametri: file_content | <pre>''' public class Main { int var1, var2; public int method_1(int a, int b){ try{ File myObj = new File("filename.txt"); Scanner myReader = new Scanner(myObj); String data = myReader.nextLine(); return a; } catch(ArithmeticException e){ System.out.println("Exception detected"); } catch(NullPointerException e){ System.out.println("Exception detected"); } } }</pre> |
| Output | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_18 |
| Oracolo | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |

Tabella 2.215: Metodo della classe SoftwareMetrics

| | |
|---|--|
| Test case ID: TC_5.1_19 | Test Frame: L1 S1 N2 CC1 PC6 PB1 PS2 PT1 PF1 NCH2 PCH3 TS2 |
| Input | |
| Parametri: file_content | <pre>''' public class Main { int var1, var2; public int method_1(int a, int b){ try{ File myObj = new File("filename.txt"); Scanner myReader = new Scanner(myObj); String data = myReader.nextLine(); return a; } catch(ArithmeticException e){ System.out.println("Exception detected"); return 0; } } }'''</pre> |
| Output | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_19 |
| Oracolo | |
| Max Essential Complexity = 1, Sum Essential Complexity = 1 | |

Tabella 2.216: Metodo della classe SoftwareMetrics

| | |
|---|---|
| Test case ID: TC_5.1_20 | Test Frame: L1 S1 N2 CC1 PC6 PB1 PS2 PT1 PF1 NCH3 PCH3 TS2 |
| Input | |
| Parametri: file_content | <pre>''' public class Main { int var1, var2; public int method_1(int a, int b){ try{ File myObj = new File("filename.txt"); Scanner myReader = new Scanner(myObj); String data = myReader.nextLine(); return a; } catch(ArithmeticException e){ System.out.println("Exception detected"); return 0; } catch(NullPointerException e){ System.out.println("Exception detected"); return -1; } } }</pre> |
| Output | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |
| Test Driver | |
| Classe: TestSoftwareMetrics SottoClasse: TestComputeEssentialComplexity | Metodo: test_case_20 |
| Oracolo | |
| Max Essential Complexity = 3, Sum Essential Complexity = 3 | |

Tabella 2.217: Metodo della classe SoftwareMetrics**MetricsWriter**

| | |
|---|------------------------|
| Test case ID: TC_5.9_1 | Test Frame: CV1 |
| Input | |
| Parametro: csv_file | None |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteHeader | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.218: Metodo della classe MetricsWriter

| | |
|---|------------------------|
| Test case ID: TC_5.9_2 | Test Frame: CV2 |
| Input | |
| Parametro: csv_file | "" |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteHeader | Metodo: test_case_2 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.219: Metodo della classe MetricsWriter

| | |
|---|---------------------|
| Test case ID: TC_5.9_3 | Test Frame: CV3 EP2 |
| Input | |
| Parametro: csv_file | test.csv |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteHeader | Metodo: test_case_3 |
| Oracolo | |
| PermissionError | |

Tabella 2.220: Metodo della classe MetricsWriter

| | |
|---|---------------------|
| Test case ID: TC_5.9_4 | Test Frame: CV3 EP1 |
| Input | |
| Parametro: csv_file | test.csv |
| Output | |
| Il file viene creato correttamente con gli header impostati | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteHeader | Metodo: test_case_4 |

| Oracolo |
|---|
| Il file viene creato correttamente con gli header impostati |

Tabella 2.221: Metodo della classe MetricsWriter

| Test case ID: TC_5.9_5 | Test Frame: CV4 EP1 |
|---|---------------------|
| Input | |
| Parametro: csv_file | test.txt |
| Output | |
| Il file viene creato correttamente con gli header impostati | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteHeader | Metodo: test_case_5 |
| Oracolo | |
| Il file viene creato correttamente con gli header impostati | |

Tabella 2.222: Metodo della classe MetricsWriter

| Test case ID: TC_5.10_1 | Test Frame: CV1 CK3 CN4 CM3 |
|--|--|
| Input | |
| Parametri: csv_file, kind, name, metrics | None, "File", "path/file.java", {"CountLineCode": 10} |
| Output | |
| TypeError | |

| Test Driver | |
|---|---------------------|
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.223: Metodo della classe MetricsWriter

| Test case ID: TC_5.10_2 | Test Frame: CV2 CK3 CN4 CM3 |
|---|---|
| Input | |
| Parametri: csv_file, kind, name, metrics | "" , "File", "path/file.java", {"CountLineCode": 10} |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_2 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.224: Metodo della classe MetricsWriter

| Test case ID: TC_5.10_3 | Test Frame: CV3 EP2 CK3 CN4 CM3 |
|-------------------------|---------------------------------|
| Input | |

| | |
|---|--|
| Parametri: csv_file, kind, name, metrics | test.csv, "File", "path/file.java", {"CountLineCode": 10} |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_3 |
| Oracolo | |
| PermissionError | |

Tabella 2.225: Metodo della classe MetricsWriter

| | |
|--|---|
| Test case ID: TC_5.10_4 | Test Frame: CV3 EP1 CK1 CN1 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, None, None, {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a None, Name a None e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_4 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.226: Metodo della classe MetricsWriter

| | |
|--|--|
| Test case ID: TC_5.10_5 | Test Frame: CV3 EP1 CK1 CN2 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, None, [2,3,4], {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a None, Name a [2,3,4] e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_5 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.227: Metodo della classe MetricsWriter

| | |
|--|---|
| Test case ID: TC_5.10_6 | Test Frame: CV3 EP1 CK1 CN3 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, None, "", {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a None, Name a "" e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_6 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.228: Metodo della classe MetricsWriter

| | |
|--|--|
| Test case ID: TC_5.10_7 | Test Frame: CV3 EP1 CK1 CN4 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, None, "path/file.java", {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a None, Name a "path/file.java" e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_7 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.229: Metodo della classe MetricsWriter

| | |
|---|--|
| Test case ID: TC_5.10_8 | Test Frame: CV3 EP1 CK2 CN1 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, 123, None, {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a 123, Name a None e CountLineCode a 10 | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_8 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.230: Metodo della classe MetricsWriter

| | |
|--|---|
| Test case ID: TC_5.10_9 | Test Frame: CV3 EP1 CK2 CN2 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, 123, [2, 3, 4], {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a 123, Name a [2,3,4] e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_9 |
| Oracolo | |
| TypeError | |

Tabella 2.231: Metodo della classe MetricsWriter

| | |
|--|--|
| Test case ID: TC_5.10_10 | Test Frame: CV3 EP1 CK2 CN3 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, 123, "", {"CountLineCode": 10} |
| Output | |

| | |
|--|----------------------|
| Viene creato il file test.csv, con Kind impostato a 123, Name a "" e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_10 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.232: Metodo della classe MetricsWriter

| | |
|--|---|
| Test case ID: TC_5.10_11 | Test Frame: CV3 EP1 CK2 CN4 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, 123, "path/file.java", {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a 123, Name a "path/file.java" e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_11 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.233: Metodo della classe MetricsWriter

| | |
|---------------------------------|--|
| Test case ID: TC_5.10_12 | Test Frame: CV3 EP1 CK3 CN1 CM3 |
|---------------------------------|--|

| Input | |
|--|---|
| Parametri: csv_file, kind, name, metrics | test.csv, "File", None, {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a "File", Name a None e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_12 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.234: Metodo della classe MetricsWriter

| Test case ID: TC_5.10_13 | Test Frame: CV3 EP1 CK3 CN2 CM3 |
|--|--|
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, "File", [1,2,3], {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a "File", Name a [1,2,3] e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_13 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.235: Metodo della classe MetricsWriter

| | |
|--|---|
| Test case ID: TC_5.10_14 | Test Frame: CV3 EP1 CK3 CN3 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, "File", "", {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a "File", Name a "" e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_14 |
| Oracolo | |
| Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori | |

Tabella 2.236: Metodo della classe MetricsWriter

| | |
|--|---|
| Test case ID: TC_5.10_15 | Test Frame: CV3 EP1 CK3 CN4 CM3 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test.csv, "File", "path/file.java", {"CountLineCode": 10} |
| Output | |
| Viene creato il file test.csv, con Kind impostato a "File", Name a "path/file.java" e CountLineCode a 10 | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_15 |
| Oracolo | |

Il file viene creato con Kind, Name e le chiavi del dizionario come colonne, e i rispettivi valori

Tabella 2.237: Metodo della classe MetricsWriter

| | |
|---|---|
| Test case ID: TC_5.10_16 | Test Frame: CV3 EP1 CK3 CN4 CM1 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test_file.csv, "File", "SomeName", None |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_16 |
| Oracolo | |
| TypeError | |

Tabella 2.238: Metodo della classe MetricsWriter

| | |
|--|--|
| Test case ID: TC_5.10_17 | Test Frame: CV3 EP1 CK3 CN4 CM2 |
| Input | |
| Parametri: csv_file, kind, name, metrics | test_file.csv, "File", "SomeName", "InvalidMetrics" |
| Output | |
| TypeError | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestMetricsWriter SottoClasse: TestWriteRows | Metodo: test_case_17 |
| Oracolo | |
| TypeError | |

Tabella 2.239: Metodo della classe MetricsWriter

2.1.4 Rifinitura dati analisi statica

SonarAnalyzer

| | |
|---|---------------------|
| Test case ID: TC_8.1_1 | Test Frame: E1 |
| Input | |
| project_key è nulla e commit_dir non contiene file Java. | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestCreateSonarProperties | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.240: Test case per la funzione create_sonar_properties

| | |
|------------------------|-------------------|
| Test case ID: TC_8.1_2 | Test Frame: E2 V1 |
| Input | |

| | |
|---|---------------------|
| project_key non è nulla e commit_dir non contiene file Java. | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestCreateSonarProperties | Metodo: test_case_2 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.241: Test case per la funzione `create_sonar_properties`

| | |
|--|---------------------|
| Test case ID: TC_8.1_3 | Test Frame: E2 V2 |
| Input | |
| project_key non è nulla e commit_dir contiene file Java. | |
| Output | |
| La funzione crea il file <code>sonar-project.properties</code> nella directory <code>commit_dir</code> . | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestCreateSonarProperties | Metodo: test_case_3 |
| Oracolo | |
| La funzione crea il file <code>sonar-project.properties</code> . | |

Tabella 2.242: Test case per la funzione `create_sonar_properties`

| | |
|---|-------------------------|
| Test case ID: TC_8.2_1 | Test Frame: DIR1 |
| Input | |
| commit_dir non esistente | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.243: Test case per la funzione run_sonar_scanner

| | |
|---|------------------------------|
| Test case ID: TC_8.2_2 | Test Frame: DIR2 KEY1 |
| Input | |
| commit_dir esistente, sonar_project_key non valida | |
| Output | |
| Exception, con errore riportato nel logger | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: test_case_2 |
| Oracolo | |
| Exception, con errore riportato nel logger | |

Tabella 2.244: Test case per la funzione `run_sonar_scanner`

| | |
|---|-----------------------------------|
| Test case ID: TC_8.2_3 | Test Frame: DIR2 KEY1 LOG1 |
| Input | |
| <code>commit_dir</code> esistente, <code>sonar_project_key</code> non valida, <code>logger</code> non esistente | |
| Output | |
| AttributeError | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: <code>test_case_3</code> |
| Oracolo | |
| AttributeError | |

Tabella 2.245: Test case per la funzione `run_sonar_scanner`

| | |
|---|---------------------------------------|
| Test case ID: TC_8.2_4 | Test Frame: DIR2 KEY2 LOG2 SQ1 |
| Input | |
| <code>commit_dir</code> esistente, <code>sonar_project_key</code> valida, <code>logger</code> esistente, SonarQube non attivo | |
| Output | |
| Exception, con errore riportato nel logger | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: <code>test_case_4</code> |

| Oracolo |
|--|
| Exception, con errore riportato nel logger |

Tabella 2.246: Test case per la funzione `run_sonar_scanner`

| Test case ID: TC_8.2_5 | Test Frame: DIR2 KEY2 LOG2 SQ2 |
|--|--------------------------------|
| Input | |
| commit_dir esistente, sonar_project_key valida, logger esistente, SonarQube attivo | |
| Output | |
| Exception, con errore riportato nel logger | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: test_case_5 |
| Oracolo | |
| Exception, con errore riportato nel logger | |

Tabella 2.247: Test case per la funzione `run_sonar_scanner`

| Test case ID: TC_8.2_6 | Test Frame: DIR2 KEY2 LOG2 SQ2 H1 |
|--|-----------------------------------|
| Input | |
| commit_dir esistente, sonar_project_key valida, logger esistente, SonarQube attivo, sonar_host non valido | |
| Output | |
| Exception, con errore riportato nel logger | |

| Test Driver | |
|---|---------------------|
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: test_case_6 |
| Oracolo | |
| Exception, con errore riportato nel logger | |

Tabella 2.248: Test case per la funzione `run_sonar_scanner`

| Test case ID: TC_8.2_7 | Test Frame: DIR2 KEY2 LOG2 SQ2 H2 T2 SS1 |
|--|---|
| Input | |
| commit_dir esistente, sonar_project_key valida, logger esistente, SonarQube attivo, sonar_host valido, sonar_token valido, percorso SonarScanner errato | |
| Output | |
| Exception, con errore riportato nel logger | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: test_case_7 |
| Oracolo | |
| Exception, con errore riportato nel logger | |

Tabella 2.249: Test case per la funzione `run_sonar_scanner`

| | |
|------------------------|---|
| Test case ID: TC_8.2_8 | Test Frame: DIR2 KEY2 LOG2 SQ2 H2 T2 SS2 |
|------------------------|---|

| Input | |
|---|---------------------|
| commit_dir esistente, sonar_project_key valida, logger esistente, SonarQube attivo, sonar_host valido, sonar_token valido, percorso SonarScanner corretto | |
| Output | |
| Il comando SonarScanner viene eseguito con successo | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestRunSonarScanner | Metodo: test_case_8 |
| Oracolo | |
| Il comando SonarScanner viene eseguito con successo | |

Tabella 2.250: Test case per la funzione run_sonar_scanner

| Test case ID: TC_8.3_1 | Test Frame: KEY1 |
|---|---------------------|
| Input | |
| project_key non valido | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetAnalysisId | Metodo: test_case_1 |
| Oracolo | |
| Exception | |

Tabella 2.251: Test case per la funzione get_analysis_id

| | |
|---|-----------------------------|
| Test case ID: TC_8.3_2 | Test Frame: KEY2 SQ1 |
| Input | |
| project_key valida, SonarQube non attivo | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetAnalysisId | Metodo: test_case_2 |
| Oracolo | |
| Exception | |

Tabella 2.252: Test case per la funzione get_analysis_id

| | |
|---|--------------------------------|
| Test case ID: TC_8.3_3 | Test Frame: KEY2 SQ2 H1 |
| Input | |
| project_key valida, SonarQube attivo, sonar_host non valido | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetAnalysisId | Metodo: test_case_3 |
| Oracolo | |
| Exception | |

Tabella 2.253: Test case per la funzione `get_analysis_id`

| | |
|---|----------------------------|
| Test case ID: TC_8.3_4 | Test Frame: KEY2 SQ2 H2 T1 |
| Input | |
| project_key valida, SonarQube attivo, sonar_host valido, sonar_token non valido | |
| Output | |
| None | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetAnalysisId | Metodo: test_case_4 |
| Oracolo | |
| None | |

Tabella 2.254: Test case per la funzione `get_analysis_id`

| | |
|---|----------------------------|
| Test case ID: TC_8.3_5 | Test Frame: KEY2 SQ2 H2 T2 |
| Input | |
| project_key valida, SonarQube attivo, sonar_host valido, sonar_token valido | |
| Output | |
| La funzione restituisce l'ID del progetto del server SonarQube | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetAnalysisId | Metodo: test_case_5 |

| Oracolo |
|-----------------|
| ID del progetto |

Tabella 2.255: Test case per la funzione `get_analysis_id`

| Test case ID: TC_8.4_1 | Test Frame: ID1 |
|---|----------------------------------|
| Input | |
| <code>analysis_id</code> non valido o di un progetto non analizzato | |
| Output | |
| None | |
| Test Driver | |
| Classe: <code>TestSonarAnalyzer</code> Sottoclasse: <code>TestCheckAnalysisStatus</code> | Metodo: <code>test_case_1</code> |
| Oracolo | |
| None | |

Tabella 2.256: Test case per la funzione `check_analysis_status`

| Test case ID: TC_8.4_2 | Test Frame: ID2 H1 |
|---|--------------------|
| Input | |
| <code>analysis_id</code> valido (progetto analizzato), <code>sonar_host</code> non valido | |
| Output | |
| Exception | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestSonarAnalyzer Sottoclasse: TestCheckAnalysisStatus | Metodo: test_case_2 |
| Oracolo | |
| Exception | |

Tabella 2.257: Test case per la funzione `check_analysis_status`

| | |
|---|-----------------------|
| Test case ID: TC_8.4_3 | Test Frame: ID2 H2 T1 |
| Input | |
| analysis_id valido, sonar_host valido, sonar_token non valido | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestCheckAnalysisStatus | Metodo: test_case_3 |
| Oracolo | |
| Exception | |

Tabella 2.258: Test case per la funzione `check_analysis_status`

| | |
|---|-----------------------|
| Test case ID: TC_8.4_4 | Test Frame: ID2 H2 T2 |
| Input | |
| analysis_id valido, sonar_host valido, sonar_token valido | |
| Output | |

| | |
|--|---------------------|
| La funzione restituisce lo stato del progetto ('SUCCESS' o 'FAILED') | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestCheckAnalysisStatus | Metodo: test_case_4 |
| Oracolo | |
| 'SUCCESS' o 'FAILED' | |

Tabella 2.259: Test case per la funzione `check_analysis_status`

| | |
|--|------------------------|
| Test case ID: TC_8.5_1 | Test Frame: ID1 |
| Input | |
| project_key non valido o di un progetto non analizzato | |
| Output | |
| La lista delle issues è vuota | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetProjectIssues | Metodo: test_case_1 |
| Oracolo | |
| La lista delle issues è vuota | |

Tabella 2.260: Test case per la funzione `get_project_issues`

| | |
|-------------------------------|-----------------------------|
| Test case ID: TC_8.5_2 | Test Frame: ID2 LOG1 |
| Input | |

| | |
|--|---------------------|
| project_key valido (progetto analizzato), logger non esistente | |
| Output | |
| AttributeError | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetProjectIssues | Metodo: test_case_2 |
| Oracolo | |
| AttributeError | |

Tabella 2.261: Test case per la funzione get_project_issues

| | |
|--|-------------------------|
| Test case ID: TC_8.5_3 | Test Frame: ID2 LOG2 H1 |
| Input | |
| project_key valido, logger esistente, sonar_host non valido | |
| Output | |
| La lista delle issues è vuota | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetProjectIssues | Metodo: test_case_3 |
| Oracolo | |
| La lista delle issues è vuota | |

Tabella 2.262: Test case per la funzione get_project_issues

| | |
|---|-----------------------------------|
| Test case ID: TC_8.5_4 | Test Frame: ID2 LOG2 H2 T1 |
| Input | |
| project_key valido, logger esistente, sonar_host valido, sonar_token non valido | |
| Output | |
| La lista delle issues è vuota | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetProjectIssues | Metodo: test_case_4 |
| Oracolo | |
| La lista delle issues è vuota | |

Tabella 2.263: Test case per la funzione get_project_issues

| | |
|---|-----------------------------------|
| Test case ID: TC_8.5_5 | Test Frame: ID2 LOG2 H2 T2 |
| Input | |
| project_key valido, logger esistente, sonar_host valido, sonar_token valido | |
| Output | |
| La funzione restituisce le issue del progetto | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestGetProjectIssues | Metodo: test_case_5 |
| Oracolo | |
| La funzione restituisce le issue del progetto | |

Tabella 2.264: Test case per la funzione `get_project_issues`

| | |
|---|---------------------------|
| Test case ID: TC_8.6_1 | Test Frame: L1 LP2 |
| Input | |
| Lista di issue vuota, directory non esistente | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestSaveIssuesToCSV | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.265: Test case per la funzione `save_issues_to_csv`

| | |
|---|-------------------------------------|
| Test case ID: TC_8.6_2 | Test Frame: L1 LP1 CP2 ECSV2 |
| Input | |
| Lista di issue vuota, directory esistente senza file Java, file CSV non esistente | |
| Output | |
| Viene creato un nuovo file, ed aggiunta l'intestazione | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestSaveIssuesToCSV | Metodo: test_case_2 |

| Oracolo |
|--|
| Viene creato un nuovo file, ed aggiunta l'intestazione |

Tabella 2.266: Test case per la funzione `save_issues_to_csv`

| Test case ID: TC_8.6_3 | Test Frame: L1 LP1 CP1 ECSV1 |
|--|----------------------------------|
| Input | |
| Lista di issue non vuota, directory esistente con file Java, file CSV esistente | |
| Output | |
| Viene creato un nuovo file, aggiunta l'intestazione, e una issue inerente al progetto contenuto in <i>project</i> | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestSaveIssuesToCSV | Metodo: <code>test_case_3</code> |
| Oracolo | |
| Viene creato un nuovo file, aggiunta l'intestazione, e una issue inerente al progetto contenuto in <i>project</i> | |

Tabella 2.267: Test case per la funzione `save_issues_to_csv`

| Test case ID: TC_8.6_4 | Test Frame: L1 LP1 CP1 ECSV2 |
|---|------------------------------|
| Input | |
| Lista di issue non vuota, directory esistente con file Java, file CSV esistente | |
| Output | |
| Viene aggiunta la issue in append | |

| Test Driver | |
|---|---------------------|
| Classe: TestSonarAnalyzer Sottoclasse: TestSaveIssuesToCSV | Metodo: test_case_4 |
| Oracolo | |
| Viene aggiunta la issue in append | |

Tabella 2.268: Test case per la funzione `save_issues_to_csv`

| Test case ID: TC_8.6_5 | Test Frame: L2 LP1 ECSV1 |
|---|--------------------------|
| Input | |
| Lista di issue non vuota, directory esistente con file Java, file CSV non esistente | |
| Output | |
| Viene creato un nuovo file, aggiunta l'intestazione, e le issue contenute in <i>issue</i> | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestSaveIssuesToCSV | Metodo: test_case_5 |
| Oracolo | |
| Viene creato un nuovo file, aggiunta l'intestazione, e le issue contenute in <i>issue</i> | |

Tabella 2.269: Test case per la funzione `save_issues_to_csv`

| Test case ID: TC_8.6_6 | Test Frame: L2 LP1 ECSV2 |
|---|--------------------------|
| Input | |
| Lista di issue non vuota, directory esistente con file Java, file CSV esistente | |

| Output | |
|---|---------------------|
| Viene aggiunta la issue in append | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestSaveIssuesToCSV | Metodo: test_case_6 |
| Oracolo | |
| Viene aggiunta la issue in append | |

Tabella 2.270: Test case per la funzione save_issues_to_csv

| Test case ID: TC_8.7_1 | Test Frame: L1 |
|---|---------------------|
| Input | |
| MR2 | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.271: Test case per la funzione process_repositories

| Test case ID: TC_8.7_2 | Test Frame: L1 |
|------------------------|----------------|
|------------------------|----------------|

| Input | |
|---|---------------------|
| MR1 RM2 | |
| Output | |
| Non viene generato alcun file | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_2 |
| Oracolo | |
| Non viene generato alcun file | |

Tabella 2.272: Test case per la funzione process_repositories

| Test case ID: TC_8.7_3 | Test Frame: L1 |
|---|---------------------|
| Input | |
| MR1 RM1 CVE2 | |
| Output | |
| Non viene generato alcun file | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_3 |
| Oracolo | |
| Non viene generato alcun file | |

Tabella 2.273: Test case per la funzione process_repositories

| | |
|---|-----------------------|
| Test case ID: TC_8.7_4 | Test Frame: L1 |
| Input | |
| MR1 RM1 CVE1 CI2 | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_4 |
| Oracolo | |
| Exception | |

Tabella 2.274: Test case per la funzione process_repositories

| | |
|---|-----------------------|
| Test case ID: TC_8.7_5 | Test Frame: L1 |
| Input | |
| MR1 RM1 CVE1 CI1 SQ2 | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_5 |
| Oracolo | |
| Exception | |

Tabella 2.275: Test case per la funzione process_repositories

| | |
|---|-----------------------|
| Test case ID: TC_8.7_6 | Test Frame: L1 |
| Input | |
| MR1 RM1 CVE1 CI1 SQ2 I1 | |
| Output | |
| Viene creato il file di output ed inserito il progetto java | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_6 |
| Oracolo | |
| Viene creato il file di output ed inserito il progetto java | |

Tabella 2.276: Test case per la funzione process_repositories

| | |
|--|-----------------------|
| Test case ID: TC_8.7_7 | Test Frame: L1 |
| Input | |
| MR1 RM1 CVE1 CI1 SQ2 I2 | |
| Output | |
| Viene creato il file di output ed inserite le vulnerabilità prese da SonarQube | |
| Test Driver | |
| Classe: TestSonarAnalyzer Sottoclasse: TestProcessRepositories | Metodo: test_case_7 |

| Oracolo |
|--|
| Viene creato il file di output ed inserite le vulnerabilità prese da SonarQube |

Tabella 2.277: Test case per la funzione `process_repositories`**DictGenerator**

| Test case ID: TC_3.2_1 | Test Frame: E1 |
|---|---------------------|
| Input | |
| <p>file non è presente.</p> <p>RepositoryMining_ASAResults_pos.csv è presente, è valido e contiene vulnerabilità.</p> | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.278: Metodo della classe DictGenerator

| Test case ID: TC_3.2_2 | Test Frame: E1 NR1 |
|-------------------------|--------------------|
| Input | |
| file esiste ma è vuoto. | |
| Output | |

| | |
|---|---------------------|
| Dizionario delle rules vuoto. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_2 |
| Oracolo | |
| Dizionario delle rules vuoto | |

Tabella 2.279: Metodo della classe DictGenerator

| | |
|---|------------------------------|
| Test case ID: TC_3.2_3 | Test Frame: E1 NR2 L1 |
| Input | |
| file esiste con 1 record e meno di 9 colonne. | |
| Output | |
| Dizionario delle rules vuoto. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_3 |
| Oracolo | |
| Dizionario delle rules vuoto | |

Tabella 2.280: Metodo della classe DictGenerator

| | |
|-------------------------------|------------------------------|
| Test case ID: TC_3.2_4 | Test Frame: E1 NR2 L2 |
| Input | |

| | |
|---|---------------------|
| file esiste con 1 record e esattamente 9 colonne. | |
| Output | |
| Dizionario delle rules vuoto. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_4 |
| Oracolo | |
| Dizionario delle rules vuoto | |

Tabella 2.281: Metodo della classe DictGenerator

| | |
|---|-----------------------|
| Test case ID: TC_3.2_5 | Test Frame: E1 NR2 L3 |
| Input | |
| file esiste con 1 record e più di 9 colonne. | |
| Output | |
| Dizionario delle rules vuoto. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_5 |
| Oracolo | |
| Dizionario delle rules vuoto | |

Tabella 2.282: Metodo della classe DictGenerator

| | |
|--|--------------------------|
| Test case ID: TC_3.2_6 | Test Frame: E1 NR3 L1 C1 |
| Input | |
| file esiste con più di 1 record, meno di 9 colonne e contiene vulnerabilità. | |
| Output | |
| IndexError | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_6 |
| Oracolo | |
| IndexError | |

Tabella 2.283: Metodo della classe DictGenerator

| | |
|--|--------------------------|
| Test case ID: TC_3.2_7 | Test Frame: E1 NR3 L1 C2 |
| Input | |
| file esiste con più di 1 record, meno di 9 colonne e non contiene vulnerabilità. | |
| Output | |
| IndexError | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_7 |
| Oracolo | |
| IndexError | |

Tabella 2.284: Metodo della classe DictGenerator

| | |
|--|--------------------------|
| Test case ID: TC_3.2_8 | Test Frame: E1 NR3 L2 C1 |
| Input | |
| file esiste con più di 1 record, esattamente 9 colonne e contiene vulnerabilità. | |
| Output | |
| Dizionario delle rules contenute in file. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_8 |
| Oracolo | |
| Dizionario delle rules contenute in file | |

Tabella 2.285: Metodo della classe DictGenerator

| | |
|--|--------------------------|
| Test case ID: TC_3.2_9 | Test Frame: E1 NR3 L2 C2 |
| Input | |
| file esiste con più di 1 record, esattamente 9 colonne e non contiene vulnerabilità. | |
| Output | |
| Dizionario delle rules vuoto. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_9 |

| Oracolo |
|------------------------------|
| Dizionario delle rules vuoto |

Tabella 2.286: Metodo della classe DictGenerator

| Test case ID: TC_3.2_10 | Test Frame: E1 NR3 L3 C1 |
|---|--------------------------|
| Input | |
| file esiste con più di 1 record, più di 9 colonne e contiene vulnerabilità. | |
| Output | |
| Dizionario delle rules contenute in file. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_10 |
| Oracolo | |
| Dizionario delle rules contenute in file | |

Tabella 2.287: Metodo della classe DictGenerator

| Test case ID: TC_3.2_11 | Test Frame: E1 NR3 L3 C2 |
|---|--------------------------|
| Input | |
| file esiste con più di 1 record, più di 9 colonne e non contiene vulnerabilità. | |
| Output | |
| Dizionario delle rules vuoto. | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestDictGenerator Sottoclasse: TestGenerateRulesDict | Metodo: test_case_11 |
| Oracolo | |
| Dizionario delle rules vuoto | |

Tabella 2.288: Metodo della classe DictGenerator

| | |
|---|---------------------|
| Test case ID: TC_3.1_1 | Test Frame: E1 |
| Input | |
| file non esiste. | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.289: Metodo della classe DictGenerator

| | |
|-------------------------|--------------------|
| Test case ID: TC_3.1_2 | Test Frame: E1 NR1 |
| Input | |
| file esiste ed è vuoto. | |
| Output | |

| | |
|---|---------------------|
| La lista di dizionari delle vulnerabilità risulterà vuota. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_2 |
| Oracolo | |
| Lista vuota | |

Tabella 2.290: Metodo della classe DictGenerator

| | |
|---|------------------------------|
| Test case ID: TC_3.1_3 | Test Frame: E1 NR2 L3 |
| Input | |
| file esiste con 1 record e più di 10 colonne. | |
| Output | |
| La lista di dizionari delle vulnerabilità risulterà vuota. | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_3 |
| Oracolo | |
| Lista vuota | |

Tabella 2.291: Metodo della classe DictGenerator

| | |
|-------------------------------|--------------------------------------|
| Test case ID: TC_3.1_4 | Test Frame: E1 NR3 L1 CV1 CN1 |
| Input | |

| | |
|--|---------------------|
| file esiste con più di un record, meno di 10 colonne e contiene vulnerabilità di tipo "VULNERABILITY" e "NO_ISSUES_FOUND". | |
| Output | |
| IndexError | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_4 |
| Oracolo | |
| IndexError | |

Tabella 2.292: Metodo della classe DictGenerator

| | |
|---|--------------------------------------|
| Test case ID: TC_3.1_5 | Test Frame: E1 NR3 L1 CV1 CN2 |
| Input | |
| file esiste con più di un record, meno di 10 colonne e contiene vulnerabilità di tipo "VULNERABILITY" ma non di tipo "NO_ISSUES_FOUND". | |
| Output | |
| IndexError | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_5 |
| Oracolo | |
| IndexError | |

Tabella 2.293: Metodo della classe DictGenerator

| | |
|--|-------------------------------|
| Test case ID: TC_3.1_6 | Test Frame: E1 NR3 L1 CV2 CN1 |
| Input | |
| file esiste con più di un record, meno di 10 colonne, senza vulnerabilità di tipo "VULNERABILITY", ma con "NO_ISSUES_FOUND". | |
| Output | |
| IndexError | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_6 |
| Oracolo | |
| IndexError | |

Tabella 2.294: Metodo della classe DictGenerator

| | |
|--|-------------------------------|
| Test case ID: TC_3.1_7 | Test Frame: E1 NR3 L2 CV1 CN1 |
| Input | |
| file esiste con più di un record, esattamente 10 colonne e contiene vulnerabilità di tipo "VULNERABILITY" e "NO_ISSUES_FOUND". | |
| Output | |
| La lista conterrà un dizionario per ogni issue di tipo "VULNERABILITY" o "NO_ISSUES_FOUND". | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_7 |

| Oracolo |
|---|
| Lista contenente un dizionario per ogni issue |

Tabella 2.295: Metodo della classe DictGenerator

| Test case ID: TC_3.1_8 | Test Frame: E1 NR3 L2 CV1 CN2 |
|---|-------------------------------|
| Input | |
| file esiste con più di un record, esattamente 10 colonne e contiene vulnerabilità di tipo "VULNERABILITY" ma non di tipo "NO_ISSUES_FOUND". | |
| Output | |
| La lista conterrà un dizionario per ogni issue di tipo "VULNERABILITY". | |
| Test Driver | |
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_8 |
| Oracolo | |
| Lista contenente un dizionario per ogni issue "VULNERABILITY" | |

Tabella 2.296: Metodo della classe DictGenerator

| Test case ID: TC_3.1_9 | Test Frame: E1 NR3 L2 CV2 CN1 |
|---|-------------------------------|
| Input | |
| file esiste con più di un record, esattamente 10 colonne e contiene solo vulnerabilità di tipo "NO_ISSUES_FOUND". | |
| Output | |
| Lista vuota | |

| Test Driver | |
|---|---------------------|
| Classe: TestDictGenerator Sottoclasse: TestGenerateVulnerabilityDict | Metodo: test_case_9 |
| Oracolo | |
| Lista vuota | |

Tabella 2.297: Metodo della classe DictGenerator

CsvCreatorForAsa

| Test case ID: TC_3.3_1 | Test Frame: E1 |
|---|---------------------|
| Input | |
| vulnerabilities non esiste | |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestProcessVulnerabilities | Metodo: test_case_1 |
| Oracolo | |
| TypeError | |

Tabella 2.298: Metodo della classe CsvCreatorForAsa

| Test case ID: TC_3.3_2 | Test Frame: E2 F2 |
|------------------------|-------------------|
| Input | |

| | |
|---|---------------------|
| vulnerabilities esiste, ed è mal formattata | |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestProcessVulnerabilities | Metodo: test_case_2 |
| Oracolo | |
| TypeError | |

Tabella 2.299: Metodo della classe CsvCreatorForAsa

| | |
|---|----------------------|
| Test case ID: TC_3.3_3 | Test Frame: E2 F1 L1 |
| Input | |
| vulnerabilities esiste ed è vuota | |
| Output | |
| Dizionario delle vulnerabilità di output vuoto | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestProcessVulnerabilities | Metodo: test_case_3 |
| Oracolo | |
| Dizionario delle vulnerabilità di output vuoto | |

Tabella 2.300: Metodo della classe CsvCreatorForAsa

| | |
|---|-----------------------------|
| Test case ID: TC_3.3_4 | Test Frame: E2 F1 L2 |
| Input | |
| vulnerabilities esiste e contiene vulnerabilità | |
| Output | |
| Dizionario delle vulnerabilità con il numero di occorrenze di ogni rule java, per ogni progetto | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestProcessVulnerabilities | Metodo: test_case_4 |
| Oracolo | |
| Dizionario delle vulnerabilità con il numero di occorrenze di ogni rule java, per ogni progetto | |

Tabella 2.301: Metodo della classe CsvCreatorForAsa

| | |
|--|-----------------------|
| Test case ID: TC_3.4_1 | Test Frame: E1 |
| Input | |
| final_csv non esiste | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestWriteFinalCsv | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.302: Metodo della classe CsvCreatorForAsa

| | |
|--|---------------------------------|
| Test case ID: TC_3.4_2 | Test Frame: E2 CBD1 CRD1 |
| Input | |
| final_csv esiste big_dict è vuoto rules_dict è vuoto | |
| Output | |
| Il file di output non contiene nessuna vulnerabilità | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestWriteFinalCsv | Metodo: test_case_2 |
| Oracolo | |
| Il file di output non contiene nessuna vulnerabilità | |

Tabella 2.303: Metodo della classe CsvCreatorForAsa

| | |
|---|---------------------------------|
| Test case ID: TC_3.4_3 | Test Frame: E2 CBD1 CRD2 |
| Input | |
| final_csv esiste big_dict è vuoto rules_dict contiene rules | |
| Output | |
| Il file di output contiene le regole come intestazione | |

| Test Driver | |
|--|---------------------|
| Classe: TestCsvCreatorForAsa Sottoclasse: TestWriteFinalCsv | Metodo: test_case_3 |
| Oracolo | |
| Il file di output contiene le regole come intestazione | |

Tabella 2.304: Metodo della classe CsvCreatorForAsa

| Test case ID: TC_3.4_4 | Test Frame: E2 CBD1 CRD2 |
|---|--------------------------|
| Input | |
| final_csv esiste big_dict contiene progetti java rules_dict è vuoto | |
| Output | |
| Il file di output non contiene nessuna vulnerabilità | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestWriteFinalCsv | Metodo: test_case_4 |
| Oracolo | |
| Il file di output non contiene nessuna vulnerabilità | |

Tabella 2.305: Metodo della classe CsvCreatorForAsa

| Test case ID: TC_3.4_5 | Test Frame: E2 CBD2 CRD2 |
|------------------------|--------------------------|
| Input | |

| | |
|---|---------------------|
| final_csv esiste big_dict contiene progetti java rules_dict contiene rules | |
| Output | |
| Il file di output contiene le vulnerabilità e i progetti di big_dict in corrispondenza delle java rules presenti in rules_dict | |
| Test Driver | |
| Classe: TestCsvCreatorForAsa Sottoclasse: TestWriteFinalCsv | Metodo: test_case_5 |
| Oracolo | |
| Il file di output contiene le vulnerabilità e i progetti di big_dict in corrispondenza delle java rules presenti in rules_dict | |

Tabella 2.306: Metodo della classe CsvCreatorForAsa

2.1.5 Union

DatasetCombiner

| | |
|-----------------------------|-----------------------|
| Test case ID: TC_4_1 | Test Frame: E1 |
| Input | |
| csv_files è vuota | |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestDatasetCombiner | Metodo: test_case_1 |

| Oracolo |
|------------|
| ValueError |

Tabella 2.307: Metodo della classe DatasetCombiner

| Test case ID: TC_4_2 | Test Frame: E2 F1 |
|--|---------------------|
| Input | |
| csv_files contenente file non formattato correttamente | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestDatasetCombiner | Metodo: test_case_2 |
| Oracolo | |
| KeyError | |

Tabella 2.308: Metodo della classe DatasetCombiner

| Test case ID: TC_4_3 | Test Frame: E2 F2 H1 |
|--|----------------------|
| Input | |
| csv_files contenente file formattato correttamente, con la presenza della colonna Name | |
| Output | |
| Combinazione dei file in corrispondenza della colonna <i>Name</i> | |
| Test Driver | |
| Classe: TestDatasetCombiner | Metodo: test_case_3 |

| Oracolo |
|---|
| Combinazione dei file in corrispondenza della colonna <i>Name</i> |

Tabella 2.309: Metodo della classe DatasetCombiner

| Test case ID: TC_4_4 | Test Frame: E2 F2 H2 |
|--|----------------------|
| Input | |
| csv_files contenente file formattato correttamente, con l'assenza della colonna Name | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestDatasetCombiner | Metodo: test_case_4 |
| Oracolo | |
| KeyError | |

Tabella 2.310: Metodo della classe DatasetCombiner

2.1.6 Main

RunTextMining

| Test case ID: TC_2.5_1 | Test Frame: DF1 |
|--|-----------------|
| Input | |
| La directory Dataset_Divided non esiste. | |
| Output | |

| | |
|---|---------------------|
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_1 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.311: Metodo della classe Main

| | |
|---|----------------------------|
| Test case ID: TC_2.5_2 | Test Frame: DF2 ES1 |
| Input | |
| La directory Dataset_Divided esiste, ma la directory di lavoro repo_path non è presente. | |
| Output | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_2 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.312: Metodo della classe Main

| | |
|---|--------------------------------|
| Test case ID: TC_2.5_3 | Test Frame: DF2 ES2 CN1 |
| Input | |
| La directory Dataset_Divided esiste e la directory di lavoro repo_path è presente ma vuota. | |
| Output | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_3 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.313: Metodo della classe Main

| | |
|---|-------------------------------------|
| Test case ID: TC_2.5_4 | Test Frame: DF2 ES2 CN2 STC1 |
| Input | |
| La directory di lavoro repo_path è presente e non vuota. La variabile cvd_id è un file. | |
| Output | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_4 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.314: Metodo della classe Main

| | |
|---|-------------------------------------|
| Test case ID: TC_2.5_5 | Test Frame: DF2 ES2 CN2 STC2 |
| Input | |
| La directory esiste e non è vuota, ma cvd_id è un file non ammesso di nome CHECK.txt. | |
| Output | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_5 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.315: Metodo della classe Main

| | |
|-------------------------------|--|
| Test case ID: TC_2.5_6 | Test Frame: DF2 ES2 CN2 STC3 CNC1 |
| Input | |

| | |
|--|----------------------------------|
| La directory esiste e non è vuota, <code>cvd_id</code> è una directory vuota. | |
| Output | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: <code>test_case_6</code> |
| Oracolo | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.316: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_2.5_7 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF1 |
| Input | |
| La directory esiste e non è vuota, <code>cvd_id</code> è una directory non vuota. La variabile <code>folder</code> è un file. | |
| Output | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: <code>test_case_7</code> |
| Oracolo | |

| |
|---|
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione |
|---|

Tabella 2.317: Metodo della classe Main

| | |
|---|---|
| Test case ID: TC_2.5_8 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF2 |
| Input | |
| La variabile <code>folder</code> è un file non ammesso .DS_Store. | |
| Output | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_8 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.318: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_2.5_9 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF1 |
| Input | |
| La variabile <code>folder</code> è una directory vuota. | |
| Output | |

| | |
|---|---------------------|
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_9 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.319: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_2.5_10 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES2 SV1 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota e la variabile <code>file</code> è un file Java accessibile. | |
| Output | |
| Vengono creati i file text_mining.txt, text_mining_dict.txt e FilteredTextMining.txt con i relativi dizionari e il csv con intestazione e una riga per ogni file java | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_10 |
| Oracolo | |
| Vengono creati i file text_mining.txt, text_mining_dict.txt e FilteredTextMining.txt con i relativi dizionari e il csv con intestazione e una riga per ogni file java | |

Tabella 2.320: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_2.5_11 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES1 SV1 |
| Input | |
| La directory <code>folder</code> non è vuota e contiene sia un file java che il relativo file di text mining | |
| Output | |
| Vengono creati i file <code>text_mining.txt</code> , <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> con i relativi dizionari e il csv con intestazione e una riga per ogni file java. | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: <code>test_case_11</code> |
| Oracolo | |
| Vengono creati i file <code>text_mining.txt</code> , <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> con i relativi dizionari e il csv con intestazione e una riga per ogni file java. | |

Tabella 2.321: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_2.5_12 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES2 SV2 |
| Input | |
| La variabile <code>file</code> è un file Java accessibile. I permessi impediscono il salvataggio del file <code>text_mining.txt</code> . | |
| Output | |
| PermissionError | |
| Test Driver | |

| | |
|--|----------------------|
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_12 |
| Oracolo | |
| PermissionError | |

Tabella 2.322: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_2.5_13 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES2 SV3 |
| Input | |
| La variabile <code>file</code> è un file Java accessibile. Si cerca però di scrivere un tipo di valore errato nel nuovo file. | |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_13 |
| Oracolo | |
| ValueError | |

Tabella 2.323: Metodo della classe Main

| | |
|-------------------------|--|
| Test case ID: TC_2.5_14 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC2 |
| Input | |

| | |
|--|-----------------------------------|
| La variabile <code>file</code> è un file Java non accessibile. | |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: <code>test_case_14</code> |
| Oracolo | |
| PermissionError | |

Tabella 2.324: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_2.5_15 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI2 |
| Input | |
| La variabili <code>file</code> è un file <code>.DS_Store</code> che non è ammesso. | |
| Output | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: <code>test_case_15</code> |
| Oracolo | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.325: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_2.5_16 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 |
| Input | |
| La variabile <code>file</code> è una cartella. | |
| Output | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |
| Test Driver | |
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: <code>test_case_16</code> |
| Oracolo | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.326: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_2.5_17 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI4 |
| Input | |
| La variabile <code>file</code> è una file <code>.txt</code> . | |
| Output | |
| Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

| Test Driver | |
|---|----------------------|
| Classe: TestMain SottoClasse: TestRunTextMining | Metodo: test_case_17 |
| Oracolo | |
| Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione | |

Tabella 2.327: Metodo della classe Main**RunSoftwareMetrics**

| Test case ID: TC_5.11_1 | Test Frame: DF1 |
|--|---------------------|
| Input | |
| La directory Dataset_Divided non esiste. | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.328: Metodo della classe Main

| Test case ID: TC_5.11_2 | Test Frame: DF2 ES1 |
|-------------------------|---------------------|
| Input | |

| | |
|--|---------------------|
| La directory Dataset_Divided esiste, ma la directory di lavoro repo_path non è presente. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: test_case_2 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.329: Metodo della classe Main

| | |
|---|--------------------------------|
| Test case ID: TC_5.11_3 | Test Frame: DF2 ES2 CN1 |
| Input | |
| La directory Dataset_Divided esiste e la directory di lavoro repo_path è presente ma vuota. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: test_case_3 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.330: Metodo della classe Main

| | |
|---|----------------------------------|
| Test case ID: TC_5.11_4 | Test Frame: DF2 ES2 CN2 STC1 |
| Input | |
| La directory di lavoro <code>repo_path</code> è presente e non vuota. La variabile <code>cvd_id</code> è un file. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_4</code> |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.331: Metodo della classe Main

| | |
|--|----------------------------------|
| Test case ID: TC_5.11_5 | Test Frame: DF2 ES2 CN2 STC2 |
| Input | |
| La directory esiste e non è vuota, ma <code>cvd_id</code> è un file non ammesso di nome CHECK.txt. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_5</code> |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.332: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_6 | Test Frame: DF2 ES2 CN2 STC3 CNC1 |
| Input | |
| La directory esiste e non è vuota, <code>cvd_id</code> è una directory vuota. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_6</code> |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.333: Metodo della classe Main

| | |
|---|---|
| Test case ID: TC_5.11_7 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF1 |
| Input | |
| La directory esiste e non è vuota, <code>cvd_id</code> è una directory non vuota. La variabile <code>folder</code> è un file. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_7</code> |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.334: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_8 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF2 |
| Input | |
| La variabile <code>folder</code> è un file non ammesso <code>.DS_Store</code> . | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_8</code> |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.335: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_5.11_9 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF1 |
| Input | |
| La variabile <code>folder</code> è una directory vuota. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_9</code> |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.336: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_10 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ1 CNM1 PF1 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java accessibile. La funzione del calcolo delle metriche contiene un dizionario pieno e corretto. Il csv è accessibile | |
| Output | |
| Scrive l'header nel csv finale e il valore delle metriche per ogni file java | |
| Test Driver | |
| Classe: <code>TestRunsoftwareMetrics</code> | Metodo: <code>test_case_10</code> |
| Oracolo | |
| Scrive l'header nel csv finale e il valore delle metriche per ogni file java | |

Tabella 2.337: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_11 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ1 CNM2 PF1 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java accessibile. La funzione del calcolo delle metriche restituisce un dizionario vuoto. Il csv è accessibile | |
| Output | |
| Scrive l'header nel csv finale e lascia uno spazio vuoto per ogni metrica | |
| Test Driver | |
| Classe: <code>TestRunsoftwareMetrics</code> | Metodo: <code>test_case_11</code> |
| Oracolo | |

Scrivo l'header nel csv finale e lascio uno spazio vuoto per ogni metrica

Tabella 2.338: Metodo della classe Main

| | |
|--|---|
| Test case ID: TC_5.11_12 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ1 CNM3 PF1 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java accessibile. La funzione del calcolo delle metriche contiene un dizionario pieno ma con una metrica mancante. Il csv è accessibile | |
| Output | |
| Scrivo solo l'header nel csv finale, scrivo il valore delle metriche presenti e uno spazio vuoto per quella mancante | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_12</code> |
| Oracolo | |
| Scrivo solo l'header nel csv finale, scrivo il valore delle metriche presenti e uno spazio vuoto per quella mancante | |

Tabella 2.339: Metodo della classe Main

| | |
|---------------------------------|---|
| Test case ID: TC_5.11_13 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ1 CNM4 PF1 |
| Input | |

| | |
|--|-----------------------------------|
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java accessibile. La funzione del calcolo delle metriche contiene un dizionario pieno ma con una metrica in più. Il csv è accessibile | |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_13</code> |
| Oracolo | |
| ValueError | |

Tabella 2.340: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_14 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ1 CNM5 PF1 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java accessibile. La funzione del calcolo delle metriche non restituisce un dizionario. Il csv è accessibile | |
| Output | |
| TypeError | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_14</code> |
| Oracolo | |
| TypeError | |

Tabella 2.341: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_15 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ1 CNM1 PF2 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java accessibile. La funzione del calcolo delle metriche restituisce un dizionario pieno e corretto. Il csv non è accessibile. | |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_15</code> |
| Oracolo | |
| PermissionError | |

Tabella 2.342: Metodo della classe Main

| | |
|--|---|
| Test case ID: TC_5.11_16 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 PFJ2 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java non accessibile. | |
| Output | |
| PermissionError | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_16</code> |
| Oracolo | |

| |
|-----------------|
| PermissionError |
|-----------------|

Tabella 2.343: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_5.11_17 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 |
| Input | |
| La variabili <code>file</code> è un file <code>.DS_Store</code> che non è ammesso. | |
| Output | |
| Scrive solo l'header previsto nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: <code>test_case_17</code> |
| Oracolo | |
| Scrive solo l'header previsto nel csv finale | |

Tabella 2.344: Metodo della classe Main

| | |
|--|--|
| Test case ID: TC_5.11_18 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI2 |
| Input | |
| La variabile <code>file</code> è una cartella. | |
| Output | |
| Scrive solo l'header previsto nel csv finale | |
| Test Driver | |

| | |
|--|----------------------|
| Classe: TestRunsoftwareMetrics | Metodo: test_case_18 |
| Oracolo | |
| Scrive solo l'header previsto nel csv finale | |

Tabella 2.345: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_5.11_19 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI4 |
| Input | |
| La variabile <code>file</code> è una file .txt. | |
| Output | |
| Scrive solo l'header previsto nel csv finale | |
| Test Driver | |
| Classe: TestRunsoftwareMetrics | Metodo: test_case_19 |
| Oracolo | |
| Scrive solo l'header previsto nel csv finale | |

Tabella 2.346: Metodo della classe Main**RunPrediction**

| | |
|-----------------------|-----------------|
| Test case ID: TCI_7.1 | Test Frame: EV1 |
| Input | |

| | |
|---|---|
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab_wrong.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" directory vocabolario non esiste |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.347: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_7.2 | Test Frame: EV2 EFV1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab_wrong.txt", "model.pkl", "encoder.pkl", "input.csv", "output.csv" |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_2 |
| Oracolo | |
| ValueError | |

Tabella 2.348: Metodo della classe Main

| Test case ID: TCI_7.3 | Test Frame: EV2 EFV2 EM1 |
|---|---|
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" directory modello non esiste |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_3 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.349: Metodo della classe Main

| Test case ID: TCI_7.4 | Test Frame: EV2 EFV2 EM2 EFM1 |
|---|---|
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.txt", "encoder.pkl", "input.csv", "output.csv" |
| Output | |
| ValueError | |
| Test Driver | |

| | |
|--|---------------------|
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_4 |
| Oracolo | |
| ValueError | |

Tabella 2.350: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_7.5 | Test Frame: EV2 EFV2 EM2 EFM2 EE1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" directory encoder non esiste |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_5 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.351: Metodo della classe Main

| | |
|-----------------------------|---|
| Test case ID: TC_7.6 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE1 |
| Input | |

| | |
|---|---|
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.txt", "input.csv", "output.csv" |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_6 |
| Oracolo | |
| ValueError | |

Tabella 2.352: Metodo della classe Main

| | |
|---|---|
| Test case ID: TC_7.7 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" directory input non esiste |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_7 |
| Oracolo | |

FileNotFoundError

Tabella 2.353: Metodo della classe Main

| | |
|---|---|
| Test case ID: TC_7.8 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI2 EFI1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.txt", "output.csv" |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_8 |
| Oracolo | |
| ValueError | |

Tabella 2.354: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_7.9 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI2 EFI2 CI1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" input.csv è vuoto |
| Output | |

| | |
|--|---------------------|
| KeyError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_9 |
| Oracolo | |
| KeyError | |

Tabella 2.355: Metodo della classe Main

| | |
|---|---|
| Test case ID: TC_7.10 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI2 EFI2 CI2 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" input.csv non contiene la colonna name |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_10 |
| Oracolo | |
| KeyError | |

Tabella 2.356: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_7.11 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI2 EFI2 CI3 CIM1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" input non corrisponde ai dati di addestramento |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_11 |
| Oracolo | |
| ValueError | |

Tabella 2.357: Metodo della classe Main

| | |
|---|--|
| Test case ID: TC_7.12 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI2 EFI2 CI3 CIM2 EP1 |
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" Le etichette delle predizioni non sono previste |
| Output | |
| ValueError | |

| Test Driver | |
|--|----------------------|
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_12 |
| Oracolo | |
| ValueError | |

Tabella 2.358: Metodo della classe Main

| Test case ID: TC_7.13 | Test Frame: EV2 EFV2 EM2 EFM2 EE2 EFE2 EI2 EFI2 CI3 EP2 |
|---|---|
| Input | |
| Parametri: vocab_path, model_path, encoder_path, input_csv_path, output_path | "vocab.pkl", "model.pkl", "encoder.pkl", "input.csv", "output.csv" Le etichette sono previste |
| Output | |
| Il file csv viene creato con il nome dei file e le relative classi predette | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunPrediction | Metodo: test_case_13 |
| Oracolo | |
| Il file csv viene creato con il nome dei file e le relative classi predette | |

Tabella 2.359: Metodo della classe Main**RunAsa**

| | |
|--------------------------------|---------------------|
| Test case ID: TC_8.8_1 | Test Frame: L1 |
| Input | |
| H1 | |
| Output | |
| Exception, Host unreachable | |
| Test Driver | |
| Classe: TestMain TestRunASA | Metodo: test_case_1 |
| Oracolo | |
| Exception, Host unreachable | |

Tabella 2.360: Test case per la funzione `run_ASA`

| | |
|--------------------------------|---------------------|
| Test case ID: TC_8.8_2 | Test Frame: L1 |
| Input | |
| H2 T1 | |
| Output | |
| Exception, Host unreachable | |
| Test Driver | |
| Classe: TestMain TestRunASA | Metodo: test_case_2 |
| Oracolo | |
| Exception, Host unreachable | |

Tabella 2.361: Test case per la funzione run_ASA

| | |
|--------------------------------|---------------------|
| Test case ID: TC_8.8_3 | Test Frame: L1 |
| Input | |
| H2 T2 SS1 | |
| Output | |
| Exception, Host unreachable | |
| Test Driver | |
| Classe: TestMain TestRunASA | Metodo: test_case_3 |
| Oracolo | |
| Exception, Host unreachable | |

Tabella 2.362: Test case per la funzione run_ASA

| | |
|--------------------------------|---------------------|
| Test case ID: TC_8.8_4 | Test Frame: L1 |
| Input | |
| H2 T2 SS2 BD1 | |
| Output | |
| Exception | |
| Test Driver | |
| Classe: TestMain TestRunASA | Metodo: test_case_4 |

| Oracolo |
|-----------|
| Exception |

Tabella 2.363: Test case per la funzione `run_ASA`

| Test case ID: TC_8.8_5 | Test Frame: L1 |
|------------------------------------|---------------------|
| Input | |
| H2 T2 SS2 BD2 ER1 CR1 | |
| Output | |
| File di output senza vulnerabilità | |
| Test Driver | |
| Classe: TestMain TestRunASA | Metodo: test_case_5 |
| Oracolo | |
| File di output senza vulnerabilità | |

Tabella 2.364: Test case per la funzione `run_ASA`

| Test case ID: TC_8.8_6 | Test Frame: L1 |
|----------------------------------|----------------|
| Input | |
| H2 T2 SS2 BD2 ER1 CR2 | |
| Output | |
| File di output con vulnerabilità | |
| Test Driver | |

| | |
|----------------------------------|---------------------|
| Classe: TestMain TestRunASA | Metodo: test_case_6 |
| Oracolo | |
| File di output con vulnerabilità | |

Tabella 2.365: Test case per la funzione run_ASA

| | |
|------------------------------------|---------------------|
| Test case ID: TC_8.8_7 | Test Frame: L1 |
| Input | |
| H2 T2 SS2 BD2 ER2 | |
| Output | |
| File di output senza vulnerabilità | |
| Test Driver | |
| Classe: TestMain TestRunASA | Metodo: test_case_7 |
| Oracolo | |
| File di output senza vulnerabilità | |

Tabella 2.366: Test case per la funzione run_ASA**RunRepoMining**

| | |
|------------------------|----------------|
| Test case ID: TC_1 | Test Frame: E2 |
| Input | |
| Il dataset non esiste. | |

| Output | |
|--|---------------------|
| FileNotFoundError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunRepoMining | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.367: Metodo della classe Main

| Test case ID: TC_2 | Test Frame: E1 F2 |
|---|---------------------|
| Input | |
| Il dataset esiste, ma il formato non è valido (l'header del dataset non contiene il nome di una colonna) | |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunRepoMining | Metodo: test_case_2 |
| Oracolo | |
| ValueError | |

Tabella 2.368: Metodo della classe Main

| Test case ID: TC_3 | Test Frame: E1 F1 C1 |
|--------------------|----------------------|
|--------------------|----------------------|

| Input | |
|---|---------------------|
| Il dataset esiste, ha un formato valido e contiene 0 righe. | |
| Output | |
| Nessuna repository è inizializzata. | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunRepoMining | Metodo: test_case_3 |
| Oracolo | |
| Nessuna repository è inizializzata. | |

Tabella 2.369: Metodo della classe Main

| Test case ID: TC_4 | Test Frame: E1 F1 C2 |
|--|----------------------|
| Input | |
| Il dataset esiste, ha un formato valido e contiene 1 riga. | |
| Output | |
| Il programma inizializza una sola repository | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunRepoMining | Metodo: test_case_4 |
| Oracolo | |
| Il programma inizializza una sola repository | |

Tabella 2.370: Metodo della classe Main

| | |
|--|-----------------------------|
| Test case ID: TC_5 | Test Frame: E1 F1 C3 |
| Input | |
| Il dataset esiste, ha un formato valido e contiene più di 1 riga. | |
| Output | |
| Il numero di repository inizializzate è pari al numero di righe del dataset. | |
| Test Driver | |
| Classe: TestMain Sottoclasse: TestRunRepoMining | Metodo: test_case_5 |
| Oracolo | |
| Il numero di repository inizializzate è pari al numero di righe del dataset | |

Tabella 2.371: Metodo della classe Main

2.2 Test Case Specification di integrazione

2.2.1 Repo Mining

| | |
|--|---------------------|
| Test case ID: TCI_1.1 | Test Frame: EI2 |
| Input | |
| initial_dataset.csv e la directory Dataset_Divided non esistono. La directory mining_results esiste. | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundException | |

Tabella 2.372: Metodo della classe Main

| | |
|--|---|
| Test case ID: TCI_1.2 | Test Frame: EI1 FI1 N2 CL1 CC3 ED1 EM2 EL1 |
| Input | |
| initial_dataset.csv ha un formato valido e 50 record. La directory Dataset_Divided esiste. La directory mining_results non esiste. | |
| Output | |
| Log indica errore: repo non esistente | |
| Test Driver | |

| | |
|---------------------------------------|---------------------|
| Classe: TestRepoMiningIntegration | Metodo: test_case_2 |
| Oracolo | |
| Log indica errore: repo non esistente | |

Tabella 2.373: Metodo della classe Main

| | |
|--|---------------------|
| Test case ID: TCI_1.3 | Test Frame: EI1 FD2 |
| Input | |
| <p><code>initial_dataset.csv</code> esiste ma ha un formato non valido.</p> <p>La directory <code>Dataset_Divided</code> esiste.</p> <p>La directory <code>mining_results</code> esiste.</p> | |
| Output | |
| ValueError | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_3 |
| Oracolo | |
| ValueError | |

Tabella 2.374: Metodo della classe Main

| | |
|--|------------------------------------|
| Test case ID: TCI_4 | Test Frame: EI1 FI1 N1 ED2 EM1 EL2 |
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ma nessun record.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |

| | |
|--|---------------------|
| Dataset_Divided creata e contiene 1.csv e mining_results contentente dir repo vuota | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_4 |
| Oracolo | |
| Dataset_Divided creata e contiene 1.csv e mining_results contentente dir repo vuota | |

Tabella 2.375: Metodo della classe Main

| | |
|---|-----------------------------------|
| Test case ID: TCI_5 | Test Frame: EI1 FI1 N2 CL3 |
| Input | |
| initial_dataset.csv ha un formato valido, ha 50 record e contiene link non valido. La directory Dataset_Divided esiste. La directory mining_results esiste. | |
| Output | |
| Log indica errore: Invalid URL | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_5 |
| Oracolo | |
| Log indica errore: Invalid URL | |

Tabella 2.376: Metodo della classe Main

| | |
|----------------------------|-----------------------------------|
| Test case ID: TCI_6 | Test Frame: EI1 FI1 N2 CL2 |
|----------------------------|-----------------------------------|

| Input | |
|--|----------------------------------|
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link non esistente.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |
| Log indica errore: Connection Error | |
| Test Driver | |
| Classe: <code>TestRepoMiningIntegration</code> | Metodo: <code>test_case_6</code> |
| Oracolo | |
| Log indica errore: Connection Error | |

Tabella 2.377: Metodo dello script `test_repo_mining_integration.py`

| Test case ID: TCI_1.7 | Test Frame: EI1 FI1 N3 CL1 CR2 ED1 EM1 EC2 EE2 |
|---|---|
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 150 record e contiene link esistente con repo non valida.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |
| Log indica errore: la repo non è disponibile. | |
| Test Driver | |
| Classe: <code>TestRepoMiningIntegration</code> | Metodo: <code>test_case_7</code> |
| Oracolo | |
| Log indica errore: la repo non è disponibile. | |

Tabella 2.378: Metodo della classe Main

| | |
|--|---|
| Test case ID: TCI_1.8 | Test Frame: EI1 FI1 N2 CL1 CR1 CC3 ED1 EM1 EL2 |
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida ma commit non valido senza risposta.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |
| Log indica errore: il commit non è esistente. | |
| Test Driver | |
| Classe: <code>TestRepoMiningIntegration</code> | Metodo: <code>test_case_8</code> |
| Oracolo | |
| Log indica errore: il commit non è esistente. | |

Tabella 2.379: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_1.9 | Test Frame: EI1 FI1 N2 CL1 CR1 CC4 ED1 EM1 EL1 |
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida ma commit non valido senza risposta.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono. File di log è presente</p> | |
| Output | |
| Log indica errore: il commit non è definito. | |

| Test Driver | |
|--|---------------------|
| Classe: TestRepoMiningIntegration | Metodo: test_case_9 |
| Oracolo | |
| Log indica errore: il commit non è definito. | |

Tabella 2.380: Metodo della classe Main

| Test case ID: TCI_1.10 | Test Frame: EI1 FI1 N2 CL1 CR1 CC2 ED1 EM1 EL2 |
|---|---|
| Input | |
| <p>initial_dataset.csv ha un formato valido, ha 50 record e contiene link esistente con repo valida ma commit valido con percorso file eccedente limite Windows.</p> <p>Le directory Dataset_Divided e mining_results esistono.</p> | |
| Output | |
| Log indica errore: GitCommandError | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_10 |
| Oracolo | |
| Log indica errore: GitCommandError | |

Tabella 2.381: Metodo della classe Main

| Test case ID: TCI_1.11 | Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM1 ED1 EM1 EL2 |
|------------------------|---|
| Input | |

| | |
|--|-----------------------------------|
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit senza modifiche.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |
| Log informa: status ok. | |
| Test Driver | |
| Classe: <code>TestRepoMiningIntegration</code> | Metodo: <code>test_case_11</code> |
| Oracolo | |
| Log informa: status ok. | |

Tabella 2.382: Metodo della classe Main

| | |
|--|---|
| Test case ID: TCI_1.12 | Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM2 ED1 EM1 EL2 |
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit senza modifiche a classi . java.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |
| Log informa: status ok. | |
| Test Driver | |
| Classe: <code>TestRepoMiningIntegration</code> | Metodo: <code>test_case_12</code> |
| Oracolo | |
| Log informa: status ok. | |

Tabella 2.383: Metodo della classe Main

| | |
|--|--|
| Test case ID: TCI_1.13 | Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM3 ED1 EM1 EL2 |
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit con modifiche solo a classi <code>.java</code> non presenti prima.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |
| Output | |
| <p>Log informa: status ok</p> <p>dir. con id commit creata e vuota</p> | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_13 |
| Oracolo | |
| <p>Log informa: status ok</p> <p>dir. con id commit creata e vuota</p> | |

Tabella 2.384: Metodo della classe Main

| | |
|---|--|
| Test case ID: TCI_14 | Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM4 ED1 EM1 EL2 |
| Input | |
| <p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit con modifiche a classi <code>.java</code> già presenti.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> | |

| Output | |
|--|----------------------|
| <pre>log informa: status ok e dir. con id commit contiene java file modificati e già esistenti</pre> | |
| Test Driver | |
| Classe: TestRepoMiningIntegration | Metodo: test_case_14 |
| Oracolo | |
| <pre>Log informa: status ok dir. con id commit contiene java file modificati e già esistenti</pre> | |

Tabella 2.385: Metodo dello script test_repo_mining_integration.py

2.2.2 Text Mining

| Test case ID: TCI_2.1 | Test Frame: DF1 |
|---|---------------------|
| Input | |
| La directory "/Divided_Dataset " non esiste. | |
| Output | |
| Viene lanciata l'eccezione FileNotFoundError. | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_1 |
| Oracolo | |
| Viene lanciata l'eccezione FileNotFoundError. | |

Tabella 2.386: Metodo della classe Main

| | |
|---|----------------------------------|
| Test case ID: TCI_2.2 | Test Frame: DF2 ES1 |
| Input | |
| La directory <code>"/Divided_Dataset"</code> esiste, ma è vuota. | |
| Output | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |
| Test Driver | |
| Classe: <code>TestTextMiningIntegration</code> | Metodo: <code>test_case_2</code> |
| Oracolo | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |

Tabella 2.387: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.3 | Test Frame: DF2 ES2 CN1 ESD2 ESF2 ESC2 |
| Input | |
| La directory <code>"/Divided_Dataset"</code> esiste e non è vuota. Repo è presente, ma vuota. | |
| Output | |
| Il file di <code>text_mining.txt</code> non viene creato, mentre tutti gli altri file vengono creati vuoti. | |
| Test Driver | |
| Classe: <code>TestTextMiningIntegration</code> | Metodo: <code>test_case_3</code> |
| Oracolo | |
| Il file di <code>text_mining.txt</code> non viene creato, mentre tutti gli altri file vengono creati vuoti. | |

Tabella 2.388: Metodo della classe Main

| | |
|---|--|
| Test case ID: TCI_2.4 | Test Frame: DF2 ES2 CN2 STC1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>cvd_id</code> è un file di testo. | |
| Output | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: <code>test_case_4</code> |
| Oracolo | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |

Tabella 2.389: Metodo della classe Main

| | |
|---|--|
| Test case ID: TCI_2.5 | Test Frame: DF2 ES2 CN2 STC2 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>cvd_id</code> è un file non ammesso, denominato CHECK.txt. | |
| Output | |
| Il file di <code>text_mining.txt</code> non viene creato, ma tutti gli altri file vengono creati vuoti. | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestTextMiningIntegration | Metodo: test_case_5 |
| Oracolo | |
| Il file di <code>text_mining.txt</code> non viene creato, ma tutti gli altri file vengono creati vuoti. | |

Tabella 2.390: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.6 | Test Frame: DF2 ES2 CN2 STC3 CNC1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>cvd_id</code> è una directory vuota. | |
| Output | |
| Il file di <code>text_mining.txt</code> non viene creato, ma tutti gli altri file vengono creati vuoti. | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_6 |
| Oracolo | |
| Il file di <code>text_mining.txt</code> non viene creato, ma tutti gli altri file vengono creati vuoti. | |

Tabella 2.391: Metodo della classe Main

| | |
|--|--|
| Test case ID: TCI_2.7 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>cvd_id</code> è una directory non vuota e contiene un file (<code>folder</code>). | |
| Output | |

| | |
|---|---------------------|
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_7 |
| Oracolo | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |

Tabella 2.392: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.8 | Test Frame: DF2 ES2 CN2 STC3 STF2 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>cvd_id</code> è una directory non vuota e contiene solo un file non ammesso (folder ha .DS_Store come nome). | |
| Output | |
| Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti. | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_8 |
| Oracolo | |
| Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti. | |

Tabella 2.393: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.9 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>folder</code> è una directory vuota. | |
| Output | |
| Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti. | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: <code>test_case_9</code> |
| Oracolo | |
| Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti. | |

Tabella 2.394: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.10 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota, che contiene un file non ammesso (<code>file</code> denominato <code>.DS_Store</code>). | |
| Output | |
| Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti. | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: <code>test_case_10</code> |
| Oracolo | |

Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti.

Tabella 2.395: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.11 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI2 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota, che contiene una sottocartella (<code>file</code>). | |
| Output | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: <code>test_case_11</code> |
| Oracolo | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |

Tabella 2.396: Metodo della classe Main

| | |
|--|--|
| Test case ID: TCI_2.12 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 ESTM2 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>file</code> è un file Java vuoto. <code>text_mining_dict.txt</code> , <code>Filtered_text_mining.txt</code> e <code>csv_mining_final.txt</code> non esistono. | |
| Output | |

| | |
|---|----------------------|
| Il file text_mining.txt viene creato vuoto, come anche gli altri file | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_12 |
| Oracolo | |
| Il file text_mining.txt viene creato vuoto, come anche gli altri file | |

Tabella 2.397: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_2.13 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 ESTM1 CFTM1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>folder</code> contiene un file java vuoto e il rispettivo file di text mining vuoto. | |
| Output | |
| Il file text_mining.txt viene sovrascritto vuoto, come anche gli altri file | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_13 |
| Oracolo | |
| Il file text_mining.txt viene sovrascritto vuoto, come anche gli altri file | |

Tabella 2.398: Metodo della classe Main

| | |
|------------------------|---|
| Test case ID: TCI_2.14 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 ESTM1 CFTM2 ESD2 ESF2 ESC2 |
|------------------------|---|

| Input | |
|---|-----------------------------------|
| La variabile <code>folder</code> contiene un file java vuoto e il rispettivo file di text mining non vuoto. | |
| Output | |
| Il file <code>text_mining.txt</code> viene sovrascritto vuoto, come anche gli altri file | |
| Test Driver | |
| Classe: <code>TestTextMiningIntegration</code> | Metodo: <code>test_case_14</code> |
| Oracolo | |
| Il file <code>text_mining.txt</code> viene sovrascritto vuoto, come anche gli altri file | |

Tabella 2.399: Metodo della classe Main

| Test case ID: TCI_2.15 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM2 ESD2 ESF2 ESC2 |
|--|--|
| Input | |
| La variabile <code>file</code> è un file Java non vuoto. <code>text_mining_dict.txt</code> , <code>Filtered_text_mining.txt</code> e <code>csv_mining_final.txt</code> non esistono. | |
| Output | |
| Il file <code>text_mining.txt</code> viene sovrascritto, col dizionario relativo, e anche gli altri file vengono creati | |
| Test Driver | |
| Classe: <code>TestTextMiningIntegration</code> | Metodo: <code>test_case_15</code> |
| Oracolo | |
| Il file <code>text_mining.txt</code> viene creato, col dizionario relativo, e anche gli altri file vengono creati | |

Tabella 2.400: Metodo della classe Main

| | |
|--|--|
| Test case ID: TCI_2.16 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM2 ESD1 ESF1 ESC1 |
| Input | |
| La variabile <code>file</code> è un file Java non vuoto. <code>text_mining_dict.txt</code> , <code>Filtered_text_mining.txt</code> e <code>csv_mining_final.txt</code> esistono. | |
| Output | |
| Il file <code>text_mining.txt</code> viene creato, col dizionario relativo, e anche gli altri file vengono sovrascritti. | |
| Test Driver | |
| Classe: <code>TestTextMiningIntegration</code> | Metodo: <code>test_case_16</code> |
| Oracolo | |
| Il file <code>text_mining.txt</code> viene creato, col dizionario relativo, e anche gli altri file vengono sovrascritti. | |

Tabella 2.401: Metodo della classe Main

| | |
|--|---|
| Test case ID: TCI_2.17 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM1 CFTM1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>folder</code> contiene un file java non vuoto e il rispettivo file di text mining vuoto. | |
| Output | |
| Il file <code>text_mining.txt</code> viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv ha 68 righe, una per ogni file java e il relativo text mining. | |

| Test Driver | |
|---|----------------------|
| Classe: TestTextMiningIntegration | Metodo: test_case_17 |
| Oracolo | |
| Il file text_mining.txt viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv deve avere 34 righe. | |

Tabella 2.402: Metodo della classe Main

| Test case ID: TCI_2.18 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM1 CFTM2 ESD2 ESF2 ESC2 |
|---|---|
| Input | |
| La variabile <code>folder</code> contiene un file java non vuoto e il rispettivo file di text mining vuoto. | |
| Output | |
| Il file text_mining.txt viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv ha 68 righe, una per ogni file java e il relativo text mining. | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: test_case_18 |
| Oracolo | |
| Il file text_mining.txt viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv deve avere 34 righe. | |

Tabella 2.403: Metodo della classe Main

| | |
|---|--|
| Test case ID: TCI_2.19 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI4 CFJ2 ESTM1 ESD2 ESF2 ESC2 |
| Input | |
| La variabile <code>file</code> è un file di testo non vuoto. | |
| Output | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |
| Test Driver | |
| Classe: TestTextMiningIntegration | Metodo: <code>test_case_19</code> |
| Oracolo | |
| Il file di text mining non viene creato, gli altri .txt vengono creati con dizionari vuoti e il csv solo l'intestazione | |

Tabella 2.404: Metodo della classe Main

2.2.3 Software Metrics

| | |
|---|----------------------------------|
| Test case ID: TCI_7.1 | Test Frame: DF1 |
| Input | |
| La directory <code>Dataset_Divided</code> non esiste. | |
| Output | |
| FileNotFoundException | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: <code>test_case_1</code> |

| Oracolo |
|-----------------------|
| FileNotFoundException |

Tabella 2.405: Metodo della classe Main

| Test case ID: TCI_7.2 | Test Frame: DF2 ES1 |
|--|---------------------|
| Input | |
| La directory Dataset_Divided esiste, ma la directory di lavoro repo_path non è presente. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_2 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.406: Metodo della classe Main

| Test case ID: TCI_7.3 | Test Frame: DF2 ES2 CN1 |
|---|-------------------------|
| Input | |
| La directory Dataset_Divided esiste e la directory di lavoro repo_path è presente ma vuota. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |

| Test Driver | |
|--|---------------------|
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_3 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.407: Metodo della classe Main

| Test case ID: TCI_7.4 | Test Frame: DF2 ES2 CN2 STC1 |
|---|------------------------------|
| Input | |
| La directory di lavoro repo_path è presente e non vuota. La variabile cvd_id è un file. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_4 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.408: Metodo della classe Main

| Test case ID: TCI_7.5 | Test Frame: DF2 ES2 CN2 STC2 |
|---|------------------------------|
| Input | |
| La directory esiste e non è vuota, ma cvd_id è un file non ammesso di nome CHECK.txt. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |

| Test Driver | |
|--|---------------------|
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_5 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.409: Metodo della classe Main

| Test case ID: TCI_7.6 | Test Frame: DF2 ES2 CN2 STC3 CNC1 |
|--|-----------------------------------|
| Input | |
| La directory esiste e non è vuota, cvd_id è una directory vuota. | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_6 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.410: Metodo della classe Main

| Test case ID: TCI_7.7 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF1 |
|---|--|
| Input | |
| La directory esiste e non è vuota, cvd_id è una directory non vuota. La variabile folder è un file. | |
| Output | |

| | |
|--|---------------------|
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_7 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.411: Metodo della classe Main

| | |
|---|---|
| Test case ID: TCI_7.8 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF2 |
| Input | |
| La variabile <code>folder</code> è un file non ammesso <code>.DS_Store</code> . | |
| Output | |
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_8 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.412: Metodo della classe Main

| | |
|---|--|
| Test case ID: TCI_7.9 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF1 |
| Input | |
| La variabile <code>folder</code> è una directory vuota. | |

| Output | |
|--|---------------------|
| Scrive solo l'header con tutte le metriche previste nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_9 |
| Oracolo | |
| Scrive solo l'header nel csv finale | |

Tabella 2.413: Metodo della classe Main

| Test case ID:TCI_7.10 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 |
|---|--|
| Input | |
| La variabili <code>file</code> è un file .DS_Store che non è ammesso. | |
| Output | |
| Scrive solo l'header previsto nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_10 |
| Oracolo | |
| Scrive solo l'header previsto nel csv finale | |

Tabella 2.414: Metodo della classe Main

| | |
|------------------------|--|
| Test case ID: TCI_7.11 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI2 |
|------------------------|--|

| Input | |
|---|-----------------------------------|
| La variabile <code>file</code> è una cartella. | |
| Output | |
| Scrive solo l'header previsto nel csv finale | |
| Test Driver | |
| Classe: <code>TestSoftwareMetricsIntegration</code> | Metodo: <code>test_case_11</code> |
| Oracolo | |
| Scrive solo l'header previsto nel csv finale | |

Tabella 2.415: Metodo della classe Main

| Test case ID: TCI_7.12 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 |
|--|---|
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java vuoto. | |
| Output | |
| Il file di log viene creato vuoto e i file vengono inserit nel csv con tutti 0 per le metriche | |
| Test Driver | |
| Classe: <code>TestSoftwareMetricsIntegration</code> | Metodo: <code>test_case_12</code> |
| Oracolo | |
| Il file di log viene creato vuoto e i file vengono inserit nel csv con tutti 0 per le metriche | |

Tabella 2.416: Metodo della classe Main

| | |
|--|---|
| Test case ID: TCI_7.13 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM2 EL2 |
| Input | |
| <p>La variabile <code>folder</code> è una directory non vuota. Contiene un file java con codice non corretto:</p> <pre>"""class Persona: def __init__(self, nome, eta): self.nome = nome self.eta = eta def saluta(self): return f"Ciao, mi chiamo self.nome e ho self.eta anni."""</pre> <p>Il file di log non è ancora stato creato, come anche il csv finale.</p> | |
| Output | |
| <p>Al log viene aggiunto un errore per ogni file errato: ERROR - Errore nell'analisi del file file.java: Il file presenta un carattere o una sequenza di caratteri non valida.'. I file vengono aggiunti al csv con i valori individuati prima dell'errore: 'File,file.java,6,1,0,7,0,0,0,0,0'</p> | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_13 |
| Oracolo | |
| <p>Il log viene creato con un errore per ogni file errato e i file vengono aggiunti al csv con i valori individuati prima dell'errore</p> | |

Tabella 2.417: Metodo della classe Main

| | |
|--|---|
| Test case ID: TCI_7.14 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM2 EL1 |
| Input | |
| <p>La variabile <code>folder</code> è una directory non vuota. Contiene un file java con codice non corretto:</p> <pre>"""class Persona: def __init__(self, nome, eta): self.nome = nome self.eta = eta def saluta(self): return f"Ciao, mi chiamo self.nome e ho self.eta anni."""</pre> <p>Il file di log è già stato creato, a differenza del file csv.</p> | |

| Output | |
|---|----------------------|
| Al log viene aggiunto un errore per ogni file errato: ERROR - Errore nell'analisi del file file.java: Il file presenta un carattere o una sequenza di caratteri non valida.'. I file vengono aggiunti al csv con i valori individuati prima dell'errore: 'File,file.java,6,1,0,7,0,0,0,0' | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_14 |
| Oracolo | |
| Al log viene aggiunto un errore per ogni file errato e i file vengono aggiunti al csv con i valori individuati prima dell'errore | |

Tabella 2.418: Metodo della classe Main

| Test case ID: TCI_7.15 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ3 ESTM2 EL2 |
|--|---|
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java con codice scritto correttamente. Il file log non è ancora stato creato, come anche il file csv. | |
| Output | |
| Il log viene creato vuoto i file vengono aggiunti al csv con i valori individuati per le metriche: 'File,file.java,62,3,6,15,4,13,1,5,3' | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_15 |
| Oracolo | |
| Il log viene creato vuoto i file vengono aggiunti al csv con i valori individuati per le metriche | |

Tabella 2.419: Metodo della classe Main

| | |
|---|--|
| Test case ID: TCI_7.16 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 ESTM1 EL2 |
| Input | |
| La variabile <code>folder</code> è una directory non vuota. Contiene un file java con codice scritto correttamente. Il file log non è ancora stato creato, a differenza del file csv finale che è già presente. | |
| Output | |
| Il log viene creato vuoto i file vengono aggiunti al csv, che viene sovrascritto con i valori individuati per le metriche: 'File,file.java,62,3,6,15,4,13,1,5,3' | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_16 |
| Oracolo | |
| Il log viene creato vuoto i file vengono aggiunti al csv con i valori individuati per le metriche | |

Tabella 2.420: Metodo della classe Main

| | |
|--|--|
| Test case ID: TCI_7.17 | Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI4 |
| Input | |
| La variabile <code>file</code> è un file .txt. | |
| Output | |
| Scrive solo l'header previsto nel csv finale | |
| Test Driver | |
| Classe: TestSoftwareMetricsIntegration | Metodo: test_case_17 |

| Oracolo |
|--|
| Scrive solo l'header previsto nel csv finale |

Tabella 2.421: Metodo della classe Main

2.2.4 Rifinitura dati analisi statica

| Test case ID: TCI_3.1 | Test Frame: H1 |
|---------------------------------------|---------------------|
| Input | |
| L'host di SonarQube non è accessibile | |
| Output | |
| Exception, Host unreachable | |
| Test Driver | |
| Classe: TestASAIntegration | Metodo: test_case_1 |
| Oracolo | |
| Exception, Host unreachable | |

Tabella 2.422: Metodo dello script TestAsaIntegration.py

| Test case ID: TCI_3.2 | Test Frame: H1 T1 |
|------------------------------------|-------------------|
| Input | |
| Il token di SonarQube non è valido | |
| Output | |
| Exception, Host unreachable | |

| Test Driver | |
|-----------------------------|---------------------|
| Classe: TestASAIntegration | Metodo: test_case_2 |
| Oracolo | |
| Exception, Host unreachable | |

Tabella 2.423: Metodo dello script TestAsaIntegration.py

| Test case ID: TCI_3.3 | Test Frame: H1 T2 SS1 |
|--------------------------------------|-----------------------|
| Input | |
| Il path di SonarScanner non è valido | |
| Output | |
| Exception, Host unreachable | |
| Test Driver | |
| Classe: TestASAIntegration | Metodo: test_case_3 |
| Oracolo | |
| Exception, Host unreachable | |

Tabella 2.424: Metodo dello script TestAsaIntegration.py

| Test case ID: TCI_3.4 | Test Frame: H1 T2 SS2 BD1 |
|-----------------------|---------------------------|
| Input | |
| base_dir non esiste | |
| Output | |
| Exception | |

| Test Driver | |
|----------------------------|---------------------|
| Classe: TestASAIntegration | Metodo: test_case_4 |
| Oracolo | |
| Exception | |

Tabella 2.425: Metodo dello script TestAsaIntegration.py

| Test case ID: TCI_3.5 | Test Frame: H1 T2 SS2 BD2 ER1 CR1 |
|---|-----------------------------------|
| Input | |
| base_dir con sottodirectory con progetti java, che non contengono issue | |
| Output | |
| File di output csv_ASA_final.csv senza vulnerabilità | |
| Test Driver | |
| Classe: TestASAIntegration | Metodo: test_case_5 |
| Oracolo | |
| File di output csv_ASA_final.csv senza vulnerabilità | |

Tabella 2.426: Metodo dello script TestAsaIntegration.py

| Test case ID: TCI_3.6 | Test Frame: H1 T2 SS2 BD2 ER1 CR2 |
|---|-----------------------------------|
| Input | |
| base_dir con sottodirectory con progetti java, che contengono issue | |
| Output | |
| File di output csv_ASA_final.csv con vulnerabilità | |

| Test Driver | |
|--|---------------------|
| Classe: TestASAIntegration | Metodo: test_case_6 |
| Oracolo | |
| File di output csv_ASA_final.csv con vulnerabilità | |

Tabella 2.427: Metodo dello script TestAsaIntegration.py

| Test case ID: TCI_3.7 | Test Frame: H1 T2 SS2 BD2 ER2 |
|--|-------------------------------|
| Input | |
| base_dir senza sottodirectory con progetti java | |
| Output | |
| File di output csv_ASA_final.csv senza vulnerabilità | |
| Test Driver | |
| Classe: TestASAIntegration | Metodo: test_case_7 |
| Oracolo | |
| File di output csv_ASA_final.csv senza vulnerabilità | |

Tabella 2.428: Metodo dello script TestAsaIntegration.py

2.2.5 RunPrediction

| | |
|------------------------------------|---------------------|
| Test case ID: TCI_7.1 | Test Frame: EV1 |
| Input | |
| Directory vocab_path non esistente | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_1 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.429: Test case per la funzione run_prediction

| | |
|--|---------------------|
| Test case ID: TCI_7.2 | Test Frame: EV2 CV1 |
| Input | |
| Directory vocab_path esistente, vocabolario nel formato atteso | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_2 |
| Oracolo | |
| KeyError | |

Tabella 2.430: Test case per la funzione run_prediction

| | |
|--|--------------------------------|
| Test case ID: TCI_7.3 | Test Frame: EV2 CV2 EM1 |
| Input | |
| Directory vocab_path esistente, ma il vocabolario non contiene il formato atteso, Directory model_path non esistente | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_3 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.431: Test case per la funzione run_prediction

| | |
|--|------------------------------------|
| Test case ID: TCI_7.4 | Test Frame: EV2 CV2 EM2 CM1 |
| Input | |
| Directory vocab_path esistente, vocabolario non nel formato atteso, Directory model_path esistente, modello nel formato atteso | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_4 |
| Oracolo | |
| KeyError | |

Tabella 2.432: Test case per la funzione `run_prediction`

| | |
|--|--------------------------------------|
| Test case ID: TCI_7.5 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE1 |
| Input | |
| Directory <code>vocab_path</code> esistente, vocabolario non nel formato atteso, Directory <code>model_path</code> esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory <code>label_encoder_path</code> non esistente | |
| Output | |
| FileNotFoundError | |
| Test Driver | |
| Classe: <code>TestRunPrediction</code> | Metodo: <code>test_case_5</code> |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.433: Test case per la funzione `run_prediction`

| | |
|---|---|
| Test case ID: TCI_7.6 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI1 |
| Input | |
| Directory <code>vocab_path</code> esistente, vocabolario non nel formato atteso, Directory <code>model_path</code> esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory <code>label_encoder_path</code> esistente, Directory <code>input_csv_path</code> non esistente | |
| Output | |
| FileNotFoundError | |

| Test Driver | |
|---------------------------|---------------------|
| Classe: TestRunPrediction | Metodo: test_case_6 |
| Oracolo | |
| FileNotFoundError | |

Tabella 2.434: Test case per la funzione run_prediction

| Test case ID: TCI_7.7 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI2 CI1 |
|--|---|
| Input | |
| Directory vocab_path esistente, vocabolario non nel formato atteso, Directory model_path esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory label_encoder_path esistente, Directory input_csv_path esistente, file di input vuoto | |
| Output | |
| EmptyDataError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_7 |
| Oracolo | |
| EmptyDataError | |

Tabella 2.435: Test case per la funzione run_prediction

| | |
|-----------------------|---|
| Test case ID: TCI_7.8 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI2 CI2 |
|-----------------------|---|

| Input | |
|--|---------------------|
| Directory vocab_path esistente, vocabolario non nel formato atteso, Directory model_path esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory label_encoder_path esistente, Directory input_csv_path esistente, file di input senza la colonna Name | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_8 |
| Oracolo | |
| KeyError | |

Tabella 2.436: Test case per la funzione run_prediction

| Test case ID: TCI_7.9 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI2 CI3 CFI1 |
|---|--|
| Input | |
| Directory vocab_path esistente, vocabolario non nel formato atteso, Directory model_path esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory label_encoder_path esistente, Directory input_csv_path esistente, file di input con la colonna Name e dati di tipo stringa | |
| Output | |
| Il csv viene creato col nome dei file e le relative predizioni | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_9 |

| Oracolo |
|--|
| Il csv viene creato col nome dei file e le relative predizioni |

Tabella 2.437: Test case per la funzione `run_prediction`

| Test case ID: TCI_7.10 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI2 CI3 CFI2 |
|--|--|
| Input | |
| Directory <code>vocab_path</code> esistente, vocabolario non nel formato atteso, Directory <code>model_path</code> esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory <code>label_encoder_path</code> esistente, Directory <code>input_csv_path</code> esistente, file di input con la colonna Name e dati di tipo intero | |
| Output | |
| Il csv viene creato col nome dei file e le relative predizioni | |
| Test Driver | |
| Classe: <code>TestRunPrediction</code> | Metodo: <code>test_case_10</code> |
| Oracolo | |
| Il csv viene creato col nome dei file e le relative predizioni | |

Tabella 2.438: Test case per la funzione `run_prediction`

| Test case ID: TCI_7.11 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI2 CI4 CFI1 |
|------------------------|--|
| Input | |

| | |
|---|----------------------|
| Directory vocab_path esistente, vocabolario non nel formato atteso, Directory model_path esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory label_encoder_path esistente, Directory input_csv_path esistente, file di input con la colonna Name e dati di tipo stringa | |
| Output | |
| Il csv viene creato col nome dei file e le relative predizioni | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_11 |
| Oracolo | |
| Il csv viene creato col nome dei file e le relative predizioni | |

Tabella 2.439: Test case per la funzione run_prediction

| | |
|--|--|
| Test case ID: TCI_7.12 | Test Frame: EV2 CV2 EM2 CM2 COM1 EE2 EI2 CI4 CFI2 |
| Input | |
| Directory vocab_path esistente, vocabolario non nel formato atteso, Directory model_path esistente, modello non nel formato atteso, modello non addestrato sul vocabolario scelto, Directory label_encoder_path esistente, Directory input_csv_path esistente, file di input con la colonna Name e dati di tipo intero | |
| Output | |
| KeyError | |
| Test Driver | |
| Classe: TestRunPrediction | Metodo: test_case_12 |
| Oracolo | |

| |
|----------|
| KeyError |
|----------|

Tabella 2.440: Test case per la funzione `run_prediction`

2.3 Test Case Specification di sistema

Per eseguire correttamente i test di sistema, è necessario specificare il path di SonarScanner, il token di SonarQube e l'indirizzo dell'host su cui è in esecuzione nel file `config.ini`.

| Test case ID: TS_1 | Test Frame: TI1 CC1 |
|---|--|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non selezionato, Non selezionato, Non selezionato |
| Il csv non è caricato | |
| Output | |
| Messaggio errore: You must upload a CSV to continue | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_1 |
| Oracolo | |
| Messaggio errore: You must upload a CSV to continue | |

Tabella 2.441: Test di sistema

| Test case ID: TS_2 | Test Frame: TI1 CC2 CTM2 CSM2 CAS2 |
|--------------------|------------------------------------|
| Input | |

| | |
|---|---|
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non selezionato, Non selezionato, Non selezionato |
| Il csv è vuoto | |
| Output | |
| Messaggio errore: You must select at least one option | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_2 |
| Oracolo | |
| Messaggio errore: You must select at least one option | |

Tabella 2.442: Test di sistema

| | |
|---|--|
| Test case ID: TS_3 | Test Frame: TI1 CC3 CTM1 CSM2 CAS1 SE1 TE1 HA1 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Non Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene dati formattati male | |
| Output | |
| Messaggio errore: Invalid header dataset | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_3 |
| Oracolo | |

Messaggio errore: Invalid header dataset

Tabella 2.443: Test di sistema

| | |
|---|--|
| Test case ID: TS_4 | Test Frame: TI1 CC4 NC1 CTM1 CSM1 CAS1 SE1 TE1 HA1 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Selezionato, Selezionato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e nessun commit | |
| Output | |
| Zip analisi contiene risultati per SM, TM, ASA, le unioni e file di log Zip predizioni è vuoto | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_4 |
| Oracolo | |
| Zip analisi contiene risultati per SM, TM, ASA, le unioni e file di log Zip predizioni è vuoto | |

Tabella 2.444: Test di sistema

| | |
|---|--|
| Test case ID: TS_5 | Test Frame: TI1 CC4 NC1 CTM1 CSM1 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Selezionato, Selezionato, Non selezionato |

| | |
|--|---------------------|
| Il csv contiene l'intestazione formattato correttamente e nessun commit | |
| Output | |
| Zip analisi contiene risultati per SM, TM, le unioni e file di log Zip predizioni è vuoto | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_5 |
| Oracolo | |
| Zip analisi contiene risultati per SM, TM, le unioni e file di log Zip predizioni è vuoto | |

Tabella 2.445: Test di sistema

| | |
|--|--|
| Test case ID: TS_6 | Test Frame: TI1 CC4 NC1 CTM1 CSM2 CAS1 SE1 TE1 HA1 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Non Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e nessun commit | |
| Output | |
| Zip analisi contiene risultati per TM e ASA, le unioni e file di log Zip predizioni è vuoto | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_6 |

| Oracolo |
|---|
| <p>Zip analisi contiene risultati per TM e ASA, le unioni e file di log</p> <p>Zip predizioni è vuoto</p> |

Tabella 2.446: Test di sistema

| Test case ID: TS_7 | Test Frame: TI1 CC4 NC1 CTM1 CSM2 CAS2 |
|--|--|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Selezionato, Non Selezionato, Non Selezionato |
| Il csv contiene l'intestazione formattata correttamente e nessun commit | |
| Output | |
| <p>Zip analisi contiene risultati per TM</p> <p>Zip predizioni è vuoto</p> | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_7 |
| Oracolo | |
| <p>Zip analisi contiene risultati per TM</p> <p>Zip predizioni è vuoto</p> | |

Tabella 2.447: Test di sistema

| Test case ID: TS_8 | Test Frame: TI1 CC4 NC1 CTM2 CSM1 CAS1 SE1 TE1 HA1 |
|--------------------|---|
| Input | |

| | |
|---|--|
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Non Selezionato, Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e nessun commit | |
| Output | |
| Zip analisi contiene risultati per SM e ASA, le unioni e file di log Zip predizioni è vuoto | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_8 |
| Oracolo | |
| Zip analisi contiene risultati per SM e ASA, le unionin e file di log Zip predizioni è vuoto | |

Tabella 2.448: Test di sistema

| | |
|---|---|
| Test case ID: TS_9 | Test Frame: TI1 CC4 NC1 CTM2 CSM1 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non Selezionato, Selezionato, Non Selezionato |
| Il csv contiene l'intestazione formattato correttamente e nessun commit | |
| Output | |
| Zip analisi contiene risultati per SM Zip predizioni è vuoto | |
| Test Driver | |

| | |
|---|---------------------|
| Classe: TestSystem | Metodo: test_case_9 |
| Oracolo | |
| Zip analisi contiene risultati per SM Zip predizioni è vuoto | |

Tabella 2.449: Test di sistema

| | |
|---|---|
| Test case ID: TS_10 | Test Frame: TI1 CC4 NC1 CTM2 CSM2 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non Selezionato, Non Selezionato, Non Selezionato |
| Il csv contiene l'intestazione formattato correttamente e nessun commit | |
| Output | |
| Messaggio errore: You must select at least one option | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_10 |
| Oracolo | |
| Messaggio errore: You must select at least one option | |

Tabella 2.450: Test di sistema

| | |
|----------------------------|---|
| Test case ID: TS_11 | Test Frame: TI1 CC4 NC2 CTM1 CSM1 CAS1 SE1 TE1 HA1 |
| Input | |

| | |
|--|--|
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Selezionato, Selezionato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Zip analisi contiene risultati per SM, TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_11 |
| Oracolo | |
| Zip analisi contiene risultati per SM, TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |

Tabella 2.451: Test di sistema

| | |
|---|--|
| Test case ID: TS_12 | Test Frame: TI1 CC4 NC2 CTM1 CSM1 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Selezionato, Selezionato, Non selezionato |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Zip analisi contiene risultati per SM, TM, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSystem | Metodo: test_case_12 |
| Oracolo | |
| Zip analisi contiene risultati per SM, TM, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |

Tabella 2.452: Test di sistema

| | |
|--|--|
| Test case ID: TS_13 | Test Frame: TI1 CC4 NC2 CTM1 CSM2 CAS1 SE1 TE1 HA1 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Non Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Zip analisi contiene risultati per TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_13 |
| Oracolo | |
| Zip analisi contiene risultati per TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |

Tabella 2.453: Test di sistema

| | |
|---------------------|--|
| Test case ID: TS_14 | Test Frame: TI1 CC4 NC2 CTM1 CSM2 CAS2 |
|---------------------|--|

| Input | |
|---|--|
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Selezionato, Non Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Zip analisi contiene risultati per TM, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_14 |
| Oracolo | |
| Zip analisi contiene risultati per TM, le unioni e file di log Zip predizioni contiene le predizioni per il commit | |

Tabella 2.454: Test di sistema

| Test case ID: TS_15 | Test Frame: TI1 CC4 NC2 CTM2 CSM1 CAS1 SE1 TE1 HA1 |
|---|--|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Non Selezionato, Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Zip analisi contiene risultati per SM, ASA, le unioni e il file di log Zip predizioni contiene le predizioni per il commit | |

| Test Driver | |
|---|----------------------|
| Classe: TestSystem | Metodo: test_case_15 |
| Oracolo | |
| Zip analisi contiene risultati per SM, ASA, le unioni e il file di log Zip predizioni contiene le predizioni per il commit | |

Tabella 2.455: Test di sistema

| Test case ID: TS_16 | Test Frame: TI1 CC4 NC2 CTM2 CSM1 CAS2 |
|---|--|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non Selezionato, Selezionato, Non Selezionato |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Zip analisi contiene risultati per SM e il file di log Zip predizioni contiene le predizioni per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_16 |
| Oracolo | |
| Zip analisi contiene risultati per SM e il file di log Zip predizioni contiene le predizioni per il commit | |

Tabella 2.456: Test di sistema

| | |
|---------------------|--|
| Test case ID: TS_17 | Test Frame: TI1 CC4 NC2 CTM2 CSM2 CAS2 |
|---------------------|--|

| Input | |
|--|--|
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non selezionato, Non selezionato, Non selezionato |
| Il csv contiene l'intestazione formattato correttamente e un solo commit | |
| Output | |
| Messaggio errore: You must select at least one option | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_17 |
| Oracolo | |
| Messaggio errore: You must select at least one option | |

Tabella 2.457: Test di sistema

| Test case ID: TS_18 | Test Frame: TI1 CC4 NC3 CTM1 CSM1 CAS1 SE1 TE1 HA1 |
|---|---|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Selezionato, Selezionato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e 60 commit | |
| Output | |
| Zip analisi contiene risultati per SM, TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSystem | Metodo: test_case_18 |
| Oracolo | |
| Zip analisi contiene risultati per SM, TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |

Tabella 2.458: Test di sistema

| | |
|--|--|
| Test case ID: TS_19 | Test Frame: TI1 CC4 NC3 CTM1 CSM1 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non selezionato, Non selezionato, Non selezionato |
| Il csv contiene l'intestazione formattato correttamente e 60 commit | |
| Output | |
| Zip analisi contiene risultati per SM, TM, le unioni e file di log, lo Zip predizioni contiene le predizioni per i commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_19 |
| Oracolo | |
| Zip analisi contiene risultati per SM, TM, le unioni e file di log, lo Zip predizioni contiene le predizioni per i commit | |

Tabella 2.459: Test di sistema

| | |
|----------------------------|---|
| Test case ID: TS_20 | Test Frame: TI1 CC4 NC3 CTM1 CSM2 CAS1 SE1 TE1 HA1 |
|----------------------------|---|

| Input | |
|---|---|
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Selezionato, Non Selezionato, Selezionato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e 60 commit | |
| Output | |
| Zip analisi contiene risultati per TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_20 |
| Oracolo | |
| Zip analisi contiene risultati per TM, ASA, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |

Tabella 2.460: Test di sistema

| Test case ID: TS_21 | Test Frame: TI1 CC4 NC3 CTM1 CSM2 CAS2 |
|--|--|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Selezionato, Non Selezionato, Non Selezionato |
| Il csv contiene l'intestazione formattato correttamente e 60 commit | |
| Output | |
| Zip analisi contiene risultati per TM, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |

| Test Driver | |
|--|----------------------|
| Classe: TestSystem | Metodo: test_case_21 |
| Oracolo | |
| Zip analisi contiene risultati per TM, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |

Tabella 2.461: Test di sistema

| Test case ID: TS_22 | Test Frame: TI1 CC4 NC3 CTM2 CSM1 CAS1 SE1 TE1 HA1 |
|--|--|
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "CSV", Non Selezionato, Selezionato, Selezio- nato, Esistente, Esistente, Esistente |
| Il csv contiene l'intestazione formattato correttamente e più commit | |
| Output | |
| Zip analisi contiene risultati per SM e ASA, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_22 |
| Oracolo | |
| Zip analisi contiene risultati per SM e ASA, le unioni e file di log Zip predizioni contiene le predizioni per i commit | |

Tabella 2.462: Test di sistema

| | |
|---|--|
| Test case ID: TS_23 | Test Frame: TI1 CC4 NC3 CTM2 CSM1 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non Selezionato, Selezionato, Non selezionato |
| Il csv contiene l'intestazione formattato correttamente e più commit | |
| Output | |
| Zip analisi contiene risultati per SM e file di log Zip predizioni contiene le predizioni per i commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_23 |
| Oracolo | |
| Zip analisi contiene risultati per SM e file di log Zip predizioni contiene le predizioni per i commit | |

Tabella 2.463: Test di sistema

| | |
|--|---|
| Test case ID: TS_24 | Test Frame: TI1 CC4 NC3 CTM2 CSM2 CAS2 |
| Input | |
| Parametri: tipo, check_tm, check_sm, check_asa | "CSV", Non Selezionato, Non elezionato, Non selezionato |
| Il csv contiene l'intestazione formattato correttamente e più commit | |
| Output | |
| Messaggio errore: You must select at least one option | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSystem | Metodo: test_case_24 |
| Oracolo | |
| Messaggio errore: You must select at least one option | |

Tabella 2.464: Test di sistema

| | |
|---|------------------------------|
| Test case ID: TS_25 | Test Frame: TI2 CID1 |
| Input | |
| Parametri: tipo, commit_id | "Single Commit", Campo vuoto |
| Commit ID non fornito | |
| Output | |
| Messaggio errore: "Inserimento commit necessario" | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_25 |
| Oracolo | |
| Messaggio errore: "You must enter Commit ID and GIT Repository URL to continue" | |

Tabella 2.465: Test di sistema

| | |
|---------------------|--|
| Test case ID: TS_26 | Test Frame: TI2 CID3 RU2 CTM1 CSM2 CAS2 |
| Input | |

| | | |
|---|---|------|
| Parametri: tipo, commit_id, repo_url, check_tm | Single Commit, https://github.com/spring- projects/spring-webflow, Selezionato | "a", |
| Commit ID non esistente e nessuna altra opzione selezionata | | |
| Output | | |
| Zip analisi contiene risultati vuoti per TM e file di log Zip predizioni vuoto | | |
| Test Driver | | |
| Classe: TestSystem | Metodo: test_case_26 | |
| Oracolo | | |
| Zip analisi contiene risultati vuoti per TM e file di log Zip predizioni vuoto | | |

Tabella 2.466: Test di sistema

| | |
|---|--|
| Test case ID: TS_27 | Test Frame: TI2 CID2 RU1 |
| Input | |
| Parametri: tipo, commit_id, repo_url | "Single Commit", 57f2ccb66946943fbf3b3f2165eac1c8eb6b1523, Campo vuoto |
| Commit ID valido ma Repo URL non fornito | |
| Output | |
| Messaggio errore: "Inserimento repo url necessario" | |
| Test Driver | |

| | |
|---|----------------------|
| Classe: TestSystem | Metodo: test_case_27 |
| Oracolo | |
| Messaggio errore: "Inserimento repo url necessario" | |

Tabella 2.467: Test di sistema

| | |
|---|--|
| Test case ID: TS_28 | Test Frame: TI2 CID2 RU2 CC1 NC1 CTM1 CSM1 CAS1 |
| Input | |
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", 57f2ccb66946943fbf3b3f2165eac1c8eb6b1523, https://github.com/spring-projects/spring-webflow , Selezionato, Non Selezionato, Selezionato, Selezionato, Selezionato |
| Commit valido e tutte le analisi selezionate tranne NC | |
| Output | |
| Zip analisi contiene risultati per TM, SM, ASA e file di log Zip predizioni vuoto | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_28 |
| Oracolo | |
| Zip analisi contiene risultati per TM, SM, ASA e file di log Zip predizioni vuoto | |

Tabella 2.468: Test di sistema

| | |
|--------------------------------------|--|
| Test case ID: TS_29 | Test Frame: TI2 CID2 RU2 CTM1 CSM1 CAS1 SE1 TE1 HA1 |
| Input | |
| Parametri: tipo, commit_id, repo_url | "Single Commit", "a", https://github.com/spring-projects/spring-webflow |
| Commit ID inesistente | |
| Output | |
| Zip analisi e Zip predizioni vuoti | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_29 |
| Oracolo | |
| Zip analisi e Zip predizioni vuoti | |

Tabella 2.469: Test di sistema

| | |
|---|--|
| Test case ID: TS_30 | Test Frame: TI2 CID2 RU2 CTM1 CSM1 CAS2 |
| Input | |
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", 57f2ccb66946943fbf3b3f2165eac1c8eb6b1523, https://github.com/spring-projects/spring-webflow, Selezionato, Non Selezionato, Selezionato, Selezionato, Non Selezionato |
| Commit valido e solo TM e SM selezionati | |

| Output | |
|--|----------------------|
| Zip analisi contiene risultati per TM, SM e file di log Zip predizioni contiene la predizione per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_30 |
| Oracolo | |
| Zip analisi contiene risultati per TM, SM e file di log Zip predizioni contiene la predizione per il commit | |

Tabella 2.470: Test di sistema

| Test case ID: TS_31 | Test Frame: TI2 CID2 RU2 CTM1 CSM2 CAS1 |
|---|--|
| Input | |
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", 57f2ccb66946943fbf3b3f2165eac1c8eb6b1523, https://github.com/spring-projects/spring-webflow , Selezionato, Non Selezionato, Selezionato, Non Selezionato, Selezionato |
| Commit valido e solo TM e ASA selezionati | |
| Output | |
| Zip analisi contiene risultati per TM, ASA e file di log Zip predizioni contiene la predizione per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_31 |

| Oracolo |
|--|
| <p>Zip analisi contiene risultati per TM, ASA e file di log</p> <p>Zip predizioni contiene la predizione per il commit</p> |

Tabella 2.471: Test di sistema

| Test case ID: TS_32 | Test Frame: TI2 CID2 RU2 CTM1 CSM2 CAS2 |
|---|--|
| Input | |
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", 57f2ccb66946943fbf3b3f2165eac1c8eb6b1523, https://github.com/spring-projects/spring-webflow , Selezionato, Non Selezionato, Selezionato, Non Selezionato, Non Selezionato |
| Commit valido e solo TM selezionato | |
| Output | |
| <p>Zip analisi contiene risultati per TM e file di log</p> <p>Zip predizioni contiene la predizione per il commit</p> | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_32 |
| Oracolo | |
| <p>Zip analisi contiene risultati per TM e file di log</p> <p>Zip predizioni contiene la predizione per il commit</p> | |

Tabella 2.472: Test di sistema

| | |
|---|--|
| Test case ID: TS_33 | Test Frame: TI2 CID2 RU2 CTM2 CSM1 CAS1 |
| Input | |
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", a5c3d21262e1448ea43eb0ae703ffb8d7eb6e982, https://github.com/spring-projects/spring- webflow, Non Selezionato, Non Selezionato, Selezionato, Selezionato, Selezionato |
| Commit valido e solo TM, SM, ASA selezionati | |
| Output | |
| Zip analisi contiene risultati per TM, SM, ASA e file di log Zip predizioni contiene la predizione per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_33 |
| Oracolo | |
| Zip analisi contiene risultati per TM, SM, ASA e file di log Zip predizioni contiene la predizione per il commit | |

Tabella 2.473: Test di sistema

| | |
|---------------------|--|
| Test case ID: TS_34 | Test Frame: TI2 CID2 RU2 CTM2 CSM1 CAS2 |
| Input | |

| | |
|--|--|
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", a5c3d21262e1448ea43eb0ae703ffb8d7eb6e982, https://github.com/spring-projects/spring- webflow, Non Selezionato, Non Selezionato, Selezionato, Selezionato, Non Selezionato |
| Commit valido e solo TM e SM selezionati | |
| Output | |
| Zip analisi contiene risultati per TM, SM e file di log Zip predizioni contiene la predizione per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_34 |
| Oracolo | |
| Zip analisi contiene risultati per TM, SM e file di log Zip predizioni contiene la predizione per il commit | |

Tabella 2.474: Test di sistema

| | |
|---|--|
| Test case ID: TS_35 | Test Frame: TI2 CID2 RU2 CTM2 CSM2 CAS1 |
| Input | |
| Parametri: tipo, commit_id, repo_url, check_cc, check_nc, check_tm, check_sm, check_asa | "Single Commit", a5c3d21262e1448ea43eb0ae703ffb8d7eb6e982, https://github.com/spring-projects/spring- webflow, Non Selezionato, Non Selezionato, Selezionato, Non Selezionato, Selezionato |
| Commit valido e solo TM e ASA selezionati | |

| Output | |
|---|----------------------|
| Zip analisi contiene risultati per TM, ASA e file di log Zip predizioni contiene la predizione per il commit | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_35 |
| Oracolo | |
| Zip analisi contiene risultati per TM, ASA e file di log Zip predizioni contiene la predizione per il commit | |

Tabella 2.475: Test di sistema

| Test case ID: TS_36 | Test Frame: TI2 CID2 RU2 CTM1 CSM2 CAS1 SE2 TE1 HA1 |
|--|---|
| Input | |
| Parametri: tipo, commit, url, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "Single commit", "57f2ccb66946943fbf3b3f2165eac1c8eb6b1523", "https://github.com/spring-projects/spring-webflow", Selezionato, Non Selezionato, Selezionato, Non Esistente, Esistente, Esistente |
| Output | |
| Messaggio errore: 'Error in static analysis, check the logs!' | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_36 |
| Oracolo | |
| Messaggio errore: 'Error in static analysis, check the logs!' | |

Tabella 2.476: Test di sistema

| | |
|--|---|
| Test case ID: TS_37 | Test Frame: TI2 CID2 RU2 CTM1 CSM2 CAS1 SE1 TE2 HA1 |
| Input | |
| Parametri: tipo, commit, url, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "Single commit", "57f2ccb66946943fbf3b3f2165eac1c8eb6b1523", "https://github.com/spring-projects/spring-webflow", Selezionato, Non Selezionato, Selezionato, Esistente, Non Esistente, Esistente |
| Output | |
| Messaggio errore: 'Error in static analysis, check the logs!' | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_37 |
| Oracolo | |
| Messaggio errore: 'Error in static analysis, check the logs!' | |

Tabella 2.477: Test di sistema

| | |
|---------------------|--|
| Test case ID: TS_38 | Test Frame: TI2 CID2 RU2 CTM1 CSM2 CAS1 SE1 TE1 HA2 |
| Input | |

| | |
|--|--|
| Parametri: tipo, commit, url, check_tm, check_sm, check_asa, sonar_path, sonar_token, sonar_host | "Single commit", "57f2ccb66946943fbf3b3f2165eac1c8eb6b1523", "https://github.com/spring-projects/spring- webflow", Selezionato, Non Selezionato, Selezionato, Esistente, Esistente, Non Esistente |
| Output | |
| Messaggio errore: 'Error in static analysis, check the logs!' | |
| Test Driver | |
| Classe: TestSystem | Metodo: test_case_38 |
| Oracolo | |
| Messaggio errore: 'Error in static analysis, check the logs!' | |

Tabella 2.478: Test di sistema