



**CORSO DI INGEGNERIA DEL SOFTWARE PROF. A. DE LUCIA**

**PROGETTO EMC**

**TEST PLAN**

**2020/2021**

|  |  |
| --- | --- |
| PARTECIPANTI | MATRICOLA |
| ALESSANDRA POTESTA’ | **06188** |
| ROSARIO ANNUNZIATA | **05810** |
| GIOVANNI TAVOLO | **05912** |



INTRODUZIONE

Lo scopo di questo documento è quello di gestire lo sviluppo e le attività di test riguardanti il sito web “European Metals Corporation”. Saranno identificati: gli elementi e le funzionalità da testare, le strategie di testing e gli strumenti utilizzati, il personale responsabile dei test, le risorse e le attività richieste per completare i test e i rischi associati al piano. Lo scopo del testing è quello di rilevare errori in maniera pianificata all'interno del codice realizzato. L’obiettivo del testing consiste nell’evitare che gli errori si presentino durante l’utilizzo del sistema dell'utente finale. I risultati prodotti dai test saranno utilizzati per comprendere dove intervenire per correggere gli errori o apportare modifiche per il migliorare il sistema. In questo documento verranno analizzate, in particolar modo, le seguenti attività:

* Gestione Utente
* Gestione Categoria
* Gestione Prodotto

# **Relazioni Con Altri Documenti**

Il Test Plan ha una stretta relazione con i documenti prodotti finora, dato che il sistema è stato pianificato nelle precedenti documentazioni. Per verificare il corretto funzionamento del sito web “European Metals Corporation” saranno usati i test cases individuati e documentati precedentemente nel processo di sviluppo del sistema. I test cases sono basati sulle funzionalità individuate nel documento di raccolta ed analisi dei requisiti.

# **Relazioni Con Il Documento Di Analisi Dei Requisiti (Rad)**

La relazione tra test plan e RAD (Requirement Analysis Document) riguarda in particolare i requisiti funzionali e non funzionali del sistema visto che i test saranno eseguiti su quelle funzionalità tenendo conto delle specifiche espresse nel documento precedente. In particolare, il RAD contiene lo scopo del sistema, l’ambito del sistema e gli obiettivi, evidenziando una panoramica di requisiti funzionali, requisiti non funzionali, scenari, casi d’uso, diagrammi e mock-up del sistema.

# **Relazioni Con Il System Design Document (Sdd)**

Tramite il System Design Document sarà possibile definire i sottosistemi e i servizi da porre a Testing.

# **Relazioni Con L’object Design Document (Odd)**

Tramite l’Object Design Document sarà possibile definire le classi e le componenti del sistema da porre a Testing.

# **Funzionalita’ Da Testare E Da Non Testare**

Di seguito saranno elencate le funzionalità introdotte nel sistema che saranno sottoposte a test, suddivise per ogni gestione del sistema.

**Feature da testare:**

## Gestione Utente

* Registrazione utente
* Login utente
* Modifica dati utenti

## Gestione Categoria

* Aggiungi categoria
* Modifica categoria

## Gestione Prodotto

* Aggiungi prodotto
* Modifica prodotto

**Feature da non testare:**

Le funzionalità escluse dal testing riguardano i requisiti funzionali di bassa e media priorità o i

requisiti per cui non è necessario creare dei casi di test perché non accettano input utente

particolari.

## Gestione Utente

* Visualizzazione pagine informazioni (chi Siamo, About, sedi)
* Visualizzazione ordine

## Gestione Categoria

* Visualizzazione categoria

## Gestione Prodotto

* Visualizzazione prodotto

# **Criteri Pass/Fail Testing**

Lo scopo del testing è quello di trovare delle failure durante l’esecuzione del sistema. Il testing ha successo se l’output osservato (finale) è diverso dall’output atteso: ciò significa che la fase di testing avrà successo se individua una failure. Nel caso verrà riscontrata una failure, bisognerà verificare a che tipo di fault è legata, se di tipo meccanico o algoritmico. Al termine dell’individuazione del fault si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. La failure quindi è uno stato di condizione nel quale non si trova l’output desiderato.

# **Approccio**

L’approccio alla fase di testing si compone di tre fasi. Si inizia con il testing di unità che ha lo scopo di testare le componenti del sistema singolarmente, poi si passa al testing di integrazione in cui le componenti del sistema verranno combinate e testate come un unico gruppo, ed infine si effettua il testing di sistema che mira a verificare il funzionamento dell’intero sistema.

# **Testing Di Unità**

Per il testing di unità si utilizzerà la tecnica BLACK-BOX. Con il BLACK-BOX testing ci focalizzeremo sul comportamento dell’input/output delle singole componenti senza tener conto della loro struttura interna. A causa della mancanza di fattibilità di effettuare un test esaustivo per l’ingente quantitativo di dati di input, verrà utilizzata la strategia del Category Partition, che consente di decomporre lo spazio di input in categorie per poi partizionare le “categorie” in classi di equivalenza chiamate “scelte”. Al termine saranno specificate le “combinazioni” delle scelte da testare creando delle istanze di casi di test specificando i valori dei dati effettivi per ciascuna scelta e determinare i risultati corrispondenti. Mediante il Category Partition otterremo, quindi, un test efficiente e privo di ridondanze.

# **Testing Di Sistema**

Il testing di sistema è stato realizzato tramite il tool Selenium.

# **Sospensione E Ripresa**

## **Criteri di sospensione**

La fase di testing sarà sospesa nel caso verrà rilevato un difetto che può limitare il processo di test. La sospensione del processo dovrà incidere il meno possibile sulle risorse disponibili. La fase di testing può essere sospesa qualora si raggiungeranno gli obiettivi dichiarati, rispettando i tempi fissati.

## **Criteri di ripresa**

La fase di testing riprenderà quando il difetto verrà risolto con successo. I test verranno ripetuti per controllare se le modifiche non hanno generato nuovi errori.

## **Materiale Per Il Testing**

Gli strumenti necessari per svolgere le attività di testing sono:

* WebServer Apache Tomcat 9 in locale che gira sul sistema.
* Un DBMS mySQL che gestisce l’utilizzo del database.
* Selenium IDE per il test di integrazione.
* JUnit per il test di unità.

# Test Case Registrazione Utente

TC\_RegistrazioneUtente

|  |  |
| --- | --- |
| Parametro: e-mail  Formato: ^\\w+([\\.-]?\\w+)\*@\\w+([\\.-]?\\w+)\*(\\.\\w+)+$ | |
| CATEGORIE | SCELTE |
| Lunghezza e-mail - LE | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza > 1 - [property LE\_OK] |
| Formato e-mail - FE | 1. Non rispetta il formato   [if LE\_OK] [error]   1. Rispetta il formato [if LE\_OK] [property FE\_OK] |
| Esiste email - EE | 1. Esiste nel DB [if LE\_OK AND FE\_OK] [error] 2. Non esiste nel DB [if LE\_OK AND FE\_OK] [property EE\_OK] |

|  |  |
| --- | --- |
| Parametro: nome  Formato: ^[ a-zA-Z\u00C0-\u00ff]+$ | |
| CATEGORIE | SCELTE |
| Lunghezza nome - LN | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >1 - [property LN\_OK] |
| Formato nome - FN | 1. Non rispetta il formato [if LN\_OK AND FN\_OK] [error] 2. Rispetta il formato [if LN\_OK AND FN\_OK] [Property FN\_OK] |

|  |  |
| --- | --- |
| Parametro: username  Formato: ^[0-9a-zA-Z]+$ | |
| CATEGORIE | SCELTE |
| Lunghezza username - LU | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >0 and <6 - [error] 3. Lunghezza >=6 - [property LU\_OK] |
| Formato username - FU | 1. Non rispetta il formato [if LU\_OK][error] 2. Rispetta il formato [if LU\_OK][property FU\_OK] |
| Esiste username - EU | 1. Esiste nel DB [if LU\_OK AND FU\_OK] [error] 2. Non esiste nel DB [if LU\_OK AND FU\_OK] [property EU\_OK] |

|  |  |
| --- | --- |
| Parametro: password  Formato: .\*[0-9].\* | |
| CATEGORIE | SCELTE |
| Lunghezza password - LP | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >0 and <7 - [error] 3. Lunghezza >7 [property LP\_OK] |
| Formato password - FP | 1. Non rispetta il formato [if LP\_OK] [error] 2. Rispetta il formato [if LP\_OK] [property LP\_OK] |
| Corrispondenza password - CP | 1. Lunghezza ConfermaPassword = 0   - [error]   1. ConfermaPassword non corrisponde a password [if LP\_OK AND FP\_OK] [error] 2. ConfermaPassword corrisponde a password [if LP\_OK AND FP\_OK] [property CP\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_RegistrazioneUtente\_1 | LN1 | **Errore:campo email vuoto** |
| TC\_RegistrazioneUtente\_2 | LN2, FN1 | **Errore:formato non valido** |
| TC\_RegistrazioneUtente\_3 | LN2, FN2, EE1 | **Errore:email esistente** |
| TC\_RegistrazioneUtente\_4 | LN2, FN2, EE2, LN1 | **Errore:campo nome vuoto** |
| TC\_RegistrazioneUtente\_5 | LN2, FN2, EE2, LN2, FN1 | **Errore:formato non valido** |
| TC\_RegistrazioneUtente\_6 | LN2, FN2, EE2, LN2, FN2, LU1, | **Errore:campo username vuoto** |
| TC\_RegistrazioneUtente\_7 | LN2, FN2, EE2, LN2, FN2, LU2, | **Errore:lunghezza username non valida** |
| TC\_RegistrazioneUtente\_8 | LN2, FN2, EE2, LN2, FN2, LU3, FU1 | **Errore:formato non valido** |
| TC\_RegistrazioneUtente\_9 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU1 | **Errore:username esistente** |
| TC\_RegistrazioneUtente\_10 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP1 | **Errore:campo password vuoto** |
| TC\_RegistrazioneUtente\_11 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP2 | **Errore:lunghezza password non valida** |
| TC\_RegistrazioneUtente\_12 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP1 | **Errore:formato non valido** |
| TC\_RegistrazioneUtente\_13 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP2, CP1 | **Errore:corrispondenza password vuoto** |
| TC\_RegistrazioneUtente\_14 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP2, CP2 | **Errore:corrispondenza password lunghezza non valida** |
| TC\_RegistrazioneUtente\_15 | LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP2, CP3 | **Corretto, registrazione avvenuta** |

# Test Case Login Utente

TC\_LoginUtente

|  |  |
| --- | --- |
| Parametro: username  Formato: ^[0-9a-zA-Z]+$ | |
| CATEGORIE | SCELTE |
| Lunghezza username - LU | 1. Lunghezza = 0 [error] 2. Lunghezza >=6 [property LU\_OK] |
| Esiste username - EU | 1. Esiste nel DB [ifLU\_OK] [error] 2. Non esiste nel DB [if LU\_OK] [property EU\_OK] |

|  |  |
| --- | --- |
| Parametro: password  Formato: .\*[0-9].\* | |
| CATEGORIE | SCELTE |
| Lunghezza password - LP | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >= 8 - [LP\_OK] |
| Corrispondenza password e-mail - CPE | 1. Non c’è corrispondenza tra password ed e-mail nel database [if LP\_OK] [error] 2. C’è corrispondenza tra password ed e-mail nel database [if LP\_OK] [property CPE\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_LoginUtente\_1 | LE1 | **Errore: campo e-mail vuoto** |
| TC\_LoginUtente\_2 | LE2, EE1 | **Errore: e-mail esiste nel database** |
| TC\_LoginUtente\_3 | LE2, EE2, LP1 | **Errore: campo password vuoto** |
| TC\_LoginUtente\_4 | LE2, EE2, LP2, CPE1 | **Errore: corrispondenza non valida** |
| TC\_LoginUtente\_5 | LE, EE2, LP2, CPE2 | **Corretto: login avvenuto** |

# Test Case Modifica Dati Utenti

TC\_ModificaDatiUtenti

|  |  |
| --- | --- |
| Parametro: username  Formato: ^[0-9a-zA-Z]+$ | |
| CATEGORIE | SCELTE |
| Lunghezza username - LU | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >0 AND <6 - [error] 3. Lunghezza >=6 - [property LU\_OK] |
| Formato username - FU | 1. Non rispetta il formato [if LU\_OK] [error] 2. Rispetta il formato [if LU\_OK] [property FU\_OK] |
| Esiste username - EU | 1. Esiste nel DB [if LU\_OK AND FU\_OK] [error] 2. Non esiste nel DB [if   LU\_OK AND FU\_OK] [property EU\_OK] |

|  |  |
| --- | --- |
| Parametro: nome  Formato: ^[ a-zA-Z\u00C0-\u00ff]+$ | |
| CATEGORIE | SCELTE |
| Lunghezza nome - LN | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LN\_OK] |
| Formato nome - FN | 1. Non rispetta il formato [if LN\_OK] [error] 2. Rispetta il formato [if   LN\_OK] [property  FN\_OK] |

|  |  |
| --- | --- |
| Parametro: e-mail  Formato: ^\\w+([\\.-]?\\w+)\*@\\w+([\\.-]?\\w+)\*(\\.\\w+)+$ | |
| CATEGORIE | SCELTE |
| Lunghezza e-mail - LE | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >= 1 - [property   LE\_OK] |
| Formato e-mail - FE | 1. Non rispetta il formato [if LE\_OK] [error] 2. Rispetta il formato [if   LE\_OK] [property  FE\_OK] |
| Esiste email - EE | 1. Esiste nel DB [if LE\_OK AND FE\_OK] [error] 2. Non esiste nel DB [if LE\_OK AND FE\_OK] [property EE\_OK ] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_ModificaDatiPersonali\_1 | LU1 | **Errore: campo username vuoto** |
| TC\_ModificaDatiPersonali\_2 | LU2 | **Errore: lunghezza username non valida** |
| TC\_ModificaDatiPersonali\_3 | LU3, FU1 | **Errore: formato username non valido** |
| TC\_ModificaDatiPersonali\_4 | LU3, FU2, EU1 | **Errore: username esiste nel database** |
| TC\_ModificaDatiPersonali\_5 | LU3, FU2, EU2, LN1 | **Errore: campo nome vuoto** |
| TC\_ModificaDatiPersonali\_6 | LU3, FU2, EU2, LN2, FN1 | **Errore: formato nome non valido** |
| TC\_ModificaDatiPersonali\_7 | LU3, FU2, EU2, LN2, FN2, LE1 | **Errore: campo e-mail vuoto** |
| TC\_ModificaDatiPersonali\_8 | LU3, FU2, EU2, LN2, FN2, LE2, FE1 | **Errore: formato e-mail non valido** |
| TC\_ModificaDatiPersonali\_9 | LU3, FU2, EU2, LN2, FN2, LE2, FE2, EE1 | **Errore: e-mail esiste nel database** |
| TC\_ModificaDatiPersonali\_10 | LU3, FU2, EU2, LN2, FN2, LE2, FE2, EE2 | **Corretto: modifica dati personali avvenuta con successo** |

# Test Case Aggiunta Categoria

TC\_AggiuntaCategoria

|  |  |  |
| --- | --- | --- |
| Parametro: nome | | |
| CATEGORIE | SCELTE | |
| Lunghezza nome - LN | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LN\_OK] | |
| Esiste nome - EN | 1. Esiste nel DB [if LN\_OK] [error] 2. Non esiste nel DB [if LN\_OK] [property EN\_OK] |

|  |  |
| --- | --- |
| Parametro: descrizione | |
| CATEGORIE | SCELTE |
| Lunghezza descrizione - LD | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LD\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_AggiuntaCategoria\_1 | LN1 | Errore: campo nome vuoto |
| TC\_AggiuntaCategoria\_2 | LN2, EN1 | Errore: nome esiste nel database |
| TC\_AggiuntaCategoria\_3 | LN2, EN2, LD1 | Errore: campo descrizione vuoto |
| TC\_AggiuntaCategoria\_4 | LN2, EN2, LD2 | Corretto: aggiunta categoria avvenuta con successo |

# Test Case Modifica Categoria

TC\_ModificaCategoria

|  |  |
| --- | --- |
| Parametro: nome | |
| CATEGORIE | SCELTE |
| Lunghezza nome - LN | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LN\_OK] |

|  |  |
| --- | --- |
| Parametro: nome e descrizione | |
| CATEGORIE | SCELTE |
| Esiste categoria - EC | 1. Esiste nel DB [error] 2. Non esiste nel DB [property EC\_OK] |

|  |  |
| --- | --- |
| Parametro: descrizione | |
| CATEGORIE | SCELTE |
| Lunghezza descrizione - LD | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LD\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_ModificaCategoria\_1 | LN1 | Errore: campo nome vuoto |
| TC\_ModificaCategoria\_2 | LN2, EC1 | Errore: categoria esiste nel database |
| TC\_ModificaCategoria\_3 | LN2, EC2, LD1 | Errore: campo descrizione vuoto |
| TC\_ModificaCategoria\_4 | LN2, EC2, LD2 | Corretto: modifica categoria avvenuta con successo |

# Test Case Aggiunta Prodotto

TC\_AggiuntaProdotto

|  |  |
| --- | --- |
| Parametro: categoria | |
| CATEGORIE | SCELTE |
| Selezione categoria - SC | 1. Selezione = false [error] 2. Selezione = true   - [property SC\_OK] |

|  |  |  |
| --- | --- | --- |
| Parametro: nome | | |
| CATEGORIE | SCELTE | |
| Lunghezza nome - LN | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LN\_OK] | |
| Esiste nome - EN | 1. Esiste nel DB [if LN\_OK] [error] 2. Non esiste nel DB [if LN\_OK] [property EN\_OK] |  |

|  |  |
| --- | --- |
| Parametro: descrizione | |
| CATEGORIE | SCELTE |
| Lunghezza descrizione - LD | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LD\_OK] |

|  |  |
| --- | --- |
| Parametro: prezzo | |
| CATEGORIE | SCELTE |
| Inserimento prezzo - IP | 1. Inserimento = false [error] 2. Inserimento = true   - [property IP\_OK] |

|  |  |
| --- | --- |
| Parametro: iva | |
| CATEGORIE | SCELTE |
| Lunghezza iva - LI | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LI\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_AggiuntaProdotto\_1 | SC1 | Errore: Categoria non selezionata |
| TC\_AggiuntaProdotto\_2 | SC2,LN1 | Errore: Campo nome vuoto |
| TC\_AggiuntaProdotto\_3 | SC2,LN2,EN1 | Errore: Nome non trovato |
| TC\_AggiuntaProdotto\_4 | SC2,LN2,EN2,LD1 | Errore: Campo descrizione vuoto |
| TC\_AggiuntaProdotto\_5 | SC2,LN2,EN2,LD2,IP1 | Errore: Prezzo non inserito |
| TC\_AggiuntaProdotto\_6 | SC2,LN2,EN2,LD2,IP2,LI1 | Errore: Campo iva vuoto |
| TC\_AggiuntaProdotto\_7 | SC2,LN2,EN2,LD2,IP2,LI2 | Corretto: Prodotto aggiunto correttamente |

# Test Case Modifica Prodotto

TC\_ModificaProdotto

|  |  |
| --- | --- |
| Parametro: categoria | |
| CATEGORIE | SCELTE |
| Selezione categoria - SC | 1. Selezione = false [error] 2. Selezione = true   - [property SC\_OK] |

|  |  |
| --- | --- |
| Parametro: nome | |
| CATEGORIE | SCELTE |
| Lunghezza nome - LN | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LN\_OK] |

|  |  |
| --- | --- |
| Parametro: categoria, nome, descrizione, prezzo, iva | |
| CATEGORIE | SCELTE |
| Esiste prodotto- EP | 1. Esiste nel DB [error] 2. Non esiste nel DB [property EP\_OK] |

|  |  |
| --- | --- |
| Parametro: descrizione | |
| CATEGORIE | SCELTE |
| Lunghezza descrizione - LD | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LD\_OK] |

|  |  |
| --- | --- |
| Parametro: prezzo | |
| CATEGORIE | SCELTE |
| Inserimento prezzo - IP | 1. Inserimento = false [error] 2. Inserimento = true   - [property IP\_OK] |

|  |  |
| --- | --- |
| Parametro: iva | |
| CATEGORIE | SCELTE |
| Lunghezza iva - LI | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LI\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_ModificaProdotto\_1 | SC1 | Errore: Categoria non selezionata |
| TC\_ ModificaProdotto\_2 | SC2,LN1 | Errore: Campo nome vuoto |
| TC\_ ModificaProdotto\_3 | SC2,LN2,EP1 | Errore: Prodotto non trovato |
| TC\_ ModificaProdotto\_4 | SC2,LN2,EP2,LD1 | Errore: Campo lunghezza vuoto |
| TC\_ ModificaProdotto\_5 | SC2,LN2,EP2,LD2,IP1 | Errore: Prezzo non inserito |
| TC\_ ModificaProdotto\_6 | SC2,LN2,EP2,LD2,IP2,LI1 | Errore: Campo iva vuoto |
| TC\_ ModificaProdotto\_7 | SC2,LN2,EP2,LD2,IP2,LI2 | Corretto: Prodotto modificato correttamente |

# Test Case Procedi Acquisto

TC\_ProcediAcquisto

|  |  |
| --- | --- |
| Parametro: indirizzo | |
| CATEGORIE | SCELTE |
| Lunghezza indirizzo - LI | 1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1   - [property LI\_OK] |

|  |  |
| --- | --- |
| Parametro: utente | |
| CATEGORIE | SCELTE |
| Log utente - LU | 1. Login = false [error] 2. Login = true   - [property LU\_OK] |

|  |  |  |
| --- | --- | --- |
| CODICE | COMBINAZIONE | ESITO |
| TC\_ ProcediAcquisto\_1 | LI1 | Errore: Campo indirizzo vuoto |
| TC\_ ProcediAcquisto\_2 | LI2,LU1 | Errore: Utente non loggato |
| TC\_ ProcediAcquisto\_3 | LI2,LU2 | Corretto: Prodotto acquistato correttamente |