Test Planning

# **1 Introduzione**

Il processo di testing, nel contesto dello sviluppo software, si propone di

raggiungere due obiettivi cruciali: dimostrare la conformità del software ai requisiti

stabiliti e individuare eventuali deviazioni dal comportamento desiderato. Pur

riconoscendo che i test non possono eliminare completamente i difetti, essi

costituiscono un pilastro essenziale per garantire che il software rispetti le

specifiche previste.

# **2 Breve riepilogo del Sistema**

L'applicazione è una rubrica che permette di gestire i contatti in modo semplice ed efficace. L'utente può creare, modificare ed eliminare contatti, visualizzarli in ordine alfabetico o singolarmente, importare una rubrica esistente e salvare quella attuale.

# **3 Funzionalità da testare**

Il focus dei test si è concentrato sul:

1. **Gestione dei contatti:** verificare la corretta gestione dei contatti, con le operazioni principali.
   * **Creazione di un nuovo contatto**: verificare che appaia correttamente nella lista contatti considerando anche i casi limite (come l’inserimento solo di un campo nome/cognome oppure l’inserimento da 0 a 3 numeri telefonici/email)
   * **Modifica di un contatto**: verificare che avvenga correttamente la modifica, o aggiunta, di uno o più campi, di un contatto precedentemente salvato.
   * **Eliminazione di un contatto**: verificare che avvenga correttamente l’eliminazione definitiva di un contatto precedentemente salvato.
2. **Ricerca e ordinamento**: verificare la corretta manipolazione della lista dei contatti.
   * **Funzionalità di ricerca**: verificare che inserendo una sottostringa nella barra di ricerca, si visualizzi nessuno, uno o più contatti correlati alla sottostringa.
   * **Funzionalità di ordinamento**: verificare che i contatti vengano ordinati alfabeticamente per cognome (e per nome in caso di cognomi uguali), dopo che l’utente ha pigiato sul bottone “*Visualizza contatti ordinati*”.
3. **Persistenza dei dati**: verificare la corretta interazione con i file system.
   * **Salvataggio rubrica**: verificare che il file .dot venga creato correttamente.
   * **Caricare rubrica**: verificare che venga importata correttamente una rubrica precedentemente salvata, senza sovrascrivere contatti già presenti.
4. **Validazione input e gestione degli errori**: verificare che i campi inseriti in input siano corretti e in caso di errore, che vengano segnalati e gestiti correttamente.

# **4 Functional Test Cases**

|  |  |
| --- | --- |
| **FTC-1.1** | **Creazione Contatto** |
| **Use Case relativo** | UC.1.1 Creare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti può essere vuota o contenere altri contatti |
| **Flusso di eventi** | 1. L’utente clicca sul pulsante “Aggiungi Contatto”. 2. Si apre la finestra di DialogoContatto. 3. L’utente compila i campi di num. di telefono/email, ma non inserisce alcun Nome o Cognome. 4. L’utente compila i campi con dati validi. 5. L’utente preme sul pulsante “Ok”. |
| **Condizioni di uscita** | 1. La finestra di dialogo si chiude 2. La MainView si aggiorna mostrando la lista dei contatti. |
| **Input** | Nome: Mario  Cognome: Rossi  Telefono 1: 3331234567  Email 1: mario.rossi@email.com |

|  |  |
| --- | --- |
| **FTC-1.2** | **Creazione Contatto** |
| **Use Case relativo** | UC.1.1 Creare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti può essere vuota o contenere altri contatti |
| **Flusso di eventi** | 1. L’utente clicca sul pulsante “Aggiungi Contatto”. 2. Si apre la finestra di DialogoContatto. 3. L’Utente non inserisce nessun dato e preme il tasto “Ok”.   L’utente ripete il passo 3 finché non inserisce dati nel formato corretto. |
| **Condizioni di uscita** | 1. Appare un messaggio di errore (JOptionPane) con il testo “E’ obbligatorio compilare almeno il nome o il cognome.” 2. La finestra di DialogoContatto non si chiude.   Nel momento in cui l’utente inserisce i dati nel formato corretto:   * La finestra di dialogo si chiude * La MainView si aggiorna mostrando la lista dei contatti. |
| **Input 1** | Nome: “”. Cognome: “”. |
| **Input 2** | Nome: “Paolo”. Cognome: “Cobolli”. |

|  |  |
| --- | --- |
| **FTC-1.3** | **Creazione Contatto** |
| **Use Case relativo** | UC.1.1 Creare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti può essere vuota o contenere altri contatti |
| **Flusso di eventi** | 1. L’utente clicca sul pulsante “Aggiungi Contatto”. 2. Si apre la finestra di DialogoContatto. 3. Compila il campo nome e inserisce un’email non nel formato corretto. 4. L’utente preme “Ok”. 5. L’utente ripete il passo 3 finché non inserisce dati nel formato corretto. |
| **Condizioni di uscita** | 1. Appare un messaggio di errore (JOptionPane) con il testo “Formato Email non valido…” 2. La finestra di DialogoContatto non si chiude.   Nel momento in cui l’utente inserisce i dati nel formato corretto:   * La finestra di dialogo si chiude * La MainView si aggiorna mostrando la lista dei contatti. |
| **Input 1** | Nome: “Luigi”. email: “luigi.verdi”. |
| **Input 2** | Nome: “Luigi”. email: “luigi.verdi@gmail.com”. |

|  |  |
| --- | --- |
| **FTC-1.4** | **Creazione Contatto** |
| **Use Case relativo** | UC.1.1 Creare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti può essere vuota o contenere altri contatti |
| **Flusso di eventi** | 1. L’utente clicca sul pulsante “Aggiungi Contatto”. 2. Si apre la finestra di DialogoContatto. 3. Compila il campo nome e inserisce un numero di telefono non nel formato corretto. 4. L’utente preme “Ok”. 5. L’utente ripete il passo 3 finché non inserisce dati nel formato corretto. |
| **Condizioni di uscita** | 1. Appare un messaggio di errore (JOptionPane) con il testo “Formato numero di telefono non valido…” 2. La finestra di DialogoContatto non si chiude.   Nel momento in cui l’utente inserisce i dati nel formato corretto:   * La finestra di dialogo si chiude * La MainView si aggiorna mostrando la lista dei contatti. |
| **Input 1** | Nome: “Franco”. Telefono 1: “33366”. |
| **Input 2** | Nome: “Franco”. Telefono 1: “3336667849”. |

|  |  |
| --- | --- |
| **FTC-2.1** | **Modifica Contatto** |
| **Use Case relativo** | UC.1.2 Modificare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata e deve contenere almeno un contatto |
| **Flusso di eventi** | 1. L’utente individua la scheda del contatto “Mario Rossi” 2. Clicca sul pulsante “Modifica” relativo a quella scheda. 3. Si apre DialogoContatto con i dati pre-compilati. 4. L’utente modifica il campo cognome e aggiunge due numeri telefonici |
| **Condizioni di uscita** | 1. La finestra di dialogo si chiude. 2. La MainView si aggiorna. 3. La scheda del contatto ora mostra le modifiche apportate. |
| **Input** | Cognome: “Verdi”. Telefono 2: “3884283239”. Telefono 3: “051918906”. |

|  |  |
| --- | --- |
| **FTC-2.2** | **Modifica Contatto** |
| **Use Case relativo** | UC.1.2 Modificare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata e deve contenere almeno un contatto |
| **Flusso di eventi** | 1. L’utente individua la scheda del contatto “Paolo Cobolli” 2. Clicca sul pulsante “Modifica” relativo a quella scheda. 3. Si apre DialogoContatto con i dati pre-compilati. 4. L’utente modifica il campo nome e cognome; modifica il secondo numero di telefono e aggiunge un indirizzo email. 5. L’utente ripete il passo 4 finché non inserisce dati nel formato corretto. |
| **Condizioni di uscita** | 1. Appare un messaggio di errore (JOptionPane) con un messaggio “E’ obbligatorio compilare almeno il nome o il cognome.”   La finestra di DialogoContatto non si chiude. Nel momento in cui l’utente inserisce i dati nel formato corretto:   * La finestra di dialogo si chiude  1. La MainView si aggiorna mostrando la lista dei contatti. |
| **Input 1** | Nome:””. Cognome: “”. Telefono 2: “3884”. email 2: “ciao”. |
| **Input 2** | Nome:”Mattia”. Cognome: “Vitali”. Telefono 2: “3884283235”. email 2: “mattia.vitali@gmail.com”. |

|  |  |
| --- | --- |
| **FTC-3.1** | **Eliminazione Contatto** |
| **Use Case relativo** | UC.1.3 Eliminare un contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti deve avere almeno un contatto. |
| **Flusso di eventi** | 1. L’utente individua la scheda di contatto “Luigi”. 2. Clicca sul pulsante “Elimina” relativo a quella scheda. 3. Appare un pop up di conferma “Sei sicuro di…?”. 4. L’utente clicca “Si”. |
| **Condizioni di uscita** | 1. Il pop up si chiude. 2. La scheda del contatto scompare dalla lista. |
| **Input** | Click sul pulsante “Si”. |

|  |  |
| --- | --- |
| **FTC-4.1** | **Ricerca Contatti** |
| **Use Case relativo** | UC.2.2 Cercare contatto |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti deve avere almeno un contatto. |
| **Flusso di eventi** | 1. L’utente digita “ver” nella barra di ricerca. 2. L’utente preme il tasto cerca. |
| **Condizioni di uscita** | 1. La lista si aggiorna mostrando tutte le schede contenente la sottostringa “ver” (Il contatto “Mario Verdi”) 2. I contatti non contenenti la sottostringa “ver”, non vengono visualizzati. |
| **Input** | Query “ver”. |

|  |  |
| --- | --- |
| **FTC-5.1** | **Ordinamento Contatti** |
| **Use Case relativo** | UC.2.1 Visualizzare rubrica ordinata. |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti deve avere almeno un contatto. |
| **Flusso di eventi** | 1. L’utente preme sul bottone relativo a “Visualizza lista ordinata”. |
| **Condizioni di uscita** | 1. La lista dei contatti viene ri-visualizzata contenente la lista dei contatti in ordine ( Mario Verdi, Mattia Vitali e Franco) |
| **Input** | Nessuno |

|  |  |
| --- | --- |
| **FTC-6.1** | **Salvataggio Rubrica** |
| **Use Case relativo** | UC.3.1 Salvare rubrica. |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. La lista dei contatti deve avere almeno un contatto. |
| **Flusso di eventi** | 1. L’utente clicca su “Salva rubrica”. 2. L’utente scegli il percorso file dove salvare il file, nomina il file(prova.dot) e clicca “Salva”. |
| **Condizioni di uscita** | 1. Appare un messaggio di successo. |
| **Input** | Interazione con il file system. |

|  |  |
| --- | --- |
| **FTC-7.1** | **Caricamento Rubrica** |
| **Use Case relativo** | UC.3.2 Carica rubrica. |
| **Condizioni di ingresso** | L’applicazione deve esser avviata. |
| **Flusso di eventi** | Prima di procedere eliminiamo tutti i contatti.   1. L’utente clicca su “Carica Rubrica”. 2. L’utente sceglie il file (prova.dot) e clicca su “Carica”. |
| **Condizioni di uscita** | 1. Appare un messaggio di successo. 2. I contatti salvati nel file “prova.dot” vengono importati. |
| **Input** | Interazione con il file system. |

# **5 Unit Test Case**

**5.1 Classe ValidatoreDati**

|  |  |  |
| --- | --- | --- |
| **UTC 1** | **TEST ITEM:** *validaNomeCognome(String nome, String cognome)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.1.1 | Nome: ”Mario”  Cognome: ”Rossi” | void (termina senza eccezzioni) |
| UTC.1.2 | Nome: ”Mario”  Cognome: ”null” | void (termina senza eccezzioni) |
| UTC.1.3 | Nome: ”null”  Cognome: ”Rossi” | void (termina senza eccezzioni) |
| UTC.1.4 | Nome: ””  Cognome: ”” | throws CampoObbligatorioexception |
| UTC.1.5 | Nome: ”null”  Cognome: ”null” | throws CampoObbligatorioexception |

|  |  |  |
| --- | --- | --- |
| **UTC 2** | **TEST ITEM:** *validaNumeroTelefono(String numero)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.2.1 | Numero: ”3884283235” | void (termina senza eccezzioni) |
| UTC.2.2 | Numero: ”+39 3391736588” | void (termina senza eccezzioni) |
| UTC.2.3 | Numero: ”abcd” | throws FormatoDatiException |

|  |  |  |
| --- | --- | --- |
| **UTC 3** | **TEST ITEM:** *validaEmail(String email)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.3.1 | Email: “mario.rossi@test.edu” | void (termina senza eccezzioni) |
| UTC.3.2 | Email: “mario.rossi.test.edu” | throws FormatoDatiException |

|  |  |  |
| --- | --- | --- |
| **UTC 4** | **TEST ITEM:** *validaLista(List <?>, list, int max)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.4.1 | lista.size() == 3, max: 3 | void (termina senza eccezzioni) |
| UTC.4.2 | lista.size() == 4, max: 3 | throws VincoloQuantitaException |

**5.2 Classe GestoreFile**

|  |  |  |
| --- | --- | --- |
| **UTC 5** | **TEST ITEM:** *scrivi(Rubrica rubrica, String filePath)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.5.1 | *filePath* è un percorso valido e scrivibile. | Il metodo termina con void. Un file viene creato/sovrascritto al *filePath*. |
| UTC.5.2 | *filePath* è un percorso non scrivibile. | throws FileNonValidoException |

|  |  |  |
| --- | --- | --- |
| **UTC 6** | **TEST ITEM:** *leggi(String filePath)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.6.1 | *filePath* punta a un file valido contenente un oggetto *Rubrica*. | return rubricaObject |
| UTC.6.2 | *filePath* punta a un file non esistente. | throws FileNonValidoException |

**5.3 Classe Contatto**

|  |  |  |
| --- | --- | --- |
| **UTC 7** | **TEST ITEM:** *valida()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.7.1 | Nome: “Mario”  Cognome: “Rossi” | return true |
| UTC.7.2 | Nome: “Mario”  Cognome: “” | return true |
| UTC.7.3 | Nome: “”  Cognome: “Rossi” | return true |
| UTC.7.4 | Nome: “”  Cognome: “” | return false |

|  |  |  |
| --- | --- | --- |
| **UTC 8** | **TEST ITEM:** *contatto(String nome, String cognome)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.8.1 | Nome: “Mario”  Cognome: “Rossi” | this.nome == “Mario”, this.cognome == “Rossi”, this.telefoni.size() == 0, this.email.size() == 0 |

|  |  |  |
| --- | --- | --- |
| **UTC 9** | **TEST ITEM:** *getNome()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.9.1 | this.nome = “Mario” | return “Mario” |
| UTC.9.2 | this.nome = “” | return null |

|  |  |  |
| --- | --- | --- |
| **UTC 10** | **TEST ITEM:** *getCognome()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.10.1 | this.cognome = “Rossi” | return “Rossi” |
| UTC.10.2 | this.cognome = “” | return “” |

|  |  |  |
| --- | --- | --- |
| **UTC 11** | **TEST ITEM:** *getTelefoni()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.11.1 | this.telefoni = [3884283235]  List <String> listaEsterna = c.getTelefoni()  listaEsterna.add(“999999999”) | this.telefoni.size() == 1 |

|  |  |  |
| --- | --- | --- |
| **UTC 12** | **TEST ITEM:** *getEmail()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.12.1 | this.email = [ab@gmail.com]  List <String> listaEsterna = c.getEmail()  listaEsterna.add(“cd@gmail.com”) | this.email.size() == 1 |

|  |  |  |
| --- | --- | --- |
| **UTC 13** | **TEST ITEM:** *setNome(String nome)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.13.1 | this.nome = “Mario”  Nome: “Luigi” | this.nome == “Luigi” |

|  |  |  |
| --- | --- | --- |
| **UTC 14** | **TEST ITEM:** *setCognome(String cognome)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.14.1 | this.cognome = “Rossi”  Cognome: “Verdi” | this.cognome == “Verdi” |

|  |  |  |
| --- | --- | --- |
| **UTC 15** | **TEST ITEM:** *aggiungiTelefono(String numero)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.15.1 | this.telefoni.size() == 2  Numero : “3391736588” | this.telefoni.size() == 3 |
| UTC.15.2 | this.telefoni.size() == 3  Numero: “3582653788” | this.telefoni.size() == 3 |

|  |  |  |
| --- | --- | --- |
| **UTC 16** | **TEST ITEM:** *rimuoviTelefono(String numero)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.16.1 | this.telefoni == [“3884283235”, “3391736588”]  Numero: ”3391736588” | this.telefoni.size() == 1  this.telefoni.contains(“3391736588”) == false |

|  |  |  |
| --- | --- | --- |
| **UTC 17** | **TEST ITEM:** *aggiungiEmail(String email)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.17.1 | this.email.size() == 2  Email : “a@b.it” | this.email.size() == 3 |
| UTC.17.2 | this.email.size() == 3  Email : “c@d.it” | this.email.size() == 3 |

|  |  |  |
| --- | --- | --- |
| **UTC 18** | **TEST ITEM:** *rimuoviEmail(String email)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.18.1 | this.email() == [[a@b.it](mailto:a@b.it)]  Email : “a@b.it” | this.email.size() == 0  this.email.contains(“a@b.it”) == false |

|  |  |  |
| --- | --- | --- |
| **UTC 19** | **TEST ITEM:** *equals(Object o)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.19.1 | Contatto c1 = new Contatto(“A”, “B”)  o = c1 | return true |
| UTC.19.2 | Contatto c1 = new Contatto(“A”, “B”)  o = null | return false |
| UTC.19.3 | Contatto c1 = new Contatto(“A”, “B”)  o = new String (“Test”) | return false |
| UTC.19.4 | Contatto c1 = new Contatto(“A”, “B”)  Contatto c2 = new Contatto(“A”, “B”)  o = c2 | return true |
| UTC.19.5 | Contatto c1 = new Contatto(“A”, “B”)  Contatto c2 = new Contatto(“A”, “C”)  o = c2 | return false |
| UTC.19.6 | Contatto c1 = new Contatto(“A”, “B”)  c1.aggiungiTelefono(“1”)  Contatto c2 = new Contatto(“A”, “B”)  o = c2 | return false |

|  |  |  |
| --- | --- | --- |
| **UTC 20** | **TEST ITEM:** *hashCode()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.20.1 | Contatto c1 = new Contatto(“A”, “B”)  Contatto c2 = new Contatto(“A”, “B”) | c1.hashCode() == c2.hashCode() |
| UTC.21.2 | Contatto c1 = new Contatto(“A”, “B”)  Contatto c2 = new Contatto(“C”, “D”) | c1.hashCode() != c2.hashCode() |

**5.4 Classe Rubrica**

|  |  |  |
| --- | --- | --- |
| **UTC 22** | **TEST ITEM:** *aggiungiContatto(Contatto contatto)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.22.1 | this.contatti.size() == 0  Parametro: c1 (nuovo contatto) | this.contatti.size() == 1  this.contatti.get(0) == c1 |
| UTC.22.2 | this.contatti.size() == 1  Parametro: cc (nuovo contatto) | this.contatti.size() == 2  this.contatti.get(0) == c2 |

|  |  |  |
| --- | --- | --- |
| **UTC 23** | **TEST ITEM:** *modificaContatto(int id, Contatto nuovoContatto)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.23.1 | this.contatti = [c1, c2]  id: 0, nuovoContatto : c3 | this.contatti.get(0) == c3 |
| UTC.23.2 | this.contatti.size() == 2  id: 5, nuovoContatto : c3 | throws ContattoNonTrovatoException |
| UTC.23.3 | this.contatti.size() == 2  id: -1, nuovoContatto : c3 | throws ContattoNonTrovatoException |

|  |  |  |
| --- | --- | --- |
| **UTC 24** | **TEST ITEM:** *eliminaContatto(int id)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.24.1 | this.contatti = [c1, c2, c3]  id: 1 | this.contatti.size() == 2 this.contatti.get(1) == c3 |
| UTC.24.2 | this.contatti = [c1]  id: 5 | throws ContattoNonTrovatoException |

|  |  |  |
| --- | --- | --- |
| **UTC 25** | **TEST ITEM:** *getContatto(int id)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.25.1 | this.contatti = [c1, c2]  id: 1 | return c2 |
| UTC.25.2 | this.contatti = [c1, c2]  id: 2 | throws ContattoNonTrovatoException |
| **UTC 26** | | **TEST ITEM:** *getContattiOrdinati()* | |
| **ID TEST CASE** | | **INPUT** | **ORACLE** |
| UTC.26.1 | | this.contatti = [Contatto(“Mario”, “Zeta”), Contatto(“Luca”, “Aquino”)] | return List Contenente : [Contatto(“Luca”, “Aquino”), Contatto(“Mario”, “Zeta”)] |
| UTC.26.2 | | this.contatti = [Contatto(“Bruno”, “Rossi”), Contatto(“Anna”, “Rossi”)] | return List Contenente : [Contatto(“Anna”, “Rossi”), Contatto(“Bruno”, “Rossi”)] |
| UTC.26.3 | | this.contatti.size() == 0 | return List  list.size() == 0 |

|  |  |  |
| --- | --- | --- |
| **UTC 26** | **TEST ITEM:** *cercaContatti(String sottoStringa)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.26.1 | contatti = [c1(“Mario”, “”Rossi)]  Parametro: “ros” | return List Contenente: c1 |
| UTC.26.2 | contatti = [c1(“Mario”, “”Rossi)]  Parametro: “verdi” | return List list.size() == 0 |

|  |  |  |
| --- | --- | --- |
| **UTC 27** | **TEST ITEM:** *isVuota()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.27.1 | this.contatti.size() == 0 | return true |
| UTC.27.2 | this.contatti.size() == 1 | return false |

|  |  |  |
| --- | --- | --- |
| **UTC 28** | **TEST ITEM:** *aggiungiContatti(List <Contatto> nuoviContatti)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.7.1 | this.contatti = [c1]  listaNuovi = [c2, c3] | this.contatti.size() == 3 |
| UTC.7.2 | this.contatti = [c1]  listaNuovi = null | this.contatti.size() == 1 |

|  |  |  |
| --- | --- | --- |
| **UTC 29** | **TEST ITEM:** *getIndiceContatto(Contatto contatto)* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.29.1 | this.contatti = [c1, c2]  Parametro: c2 | return 1 |
| UTC.29.2 | this.contatti = [c1, c2]  Parametro: c3 | return -1 |

**5.5 Classe RubricaController**

|  |  |  |
| --- | --- | --- |
| **UTC 30** | **TEST ITEM:** *valida()* | |
| **ID TEST CASE** | **INPUT** | **ORACLE** |
| UTC.30.1 |  |  |
| UTC.30.2 |  |  |
| UTC.7.3 |  |  |
| UTC.7.4 |  |  |