

Mille Miglia In Viaggio
Unit Testing
Versione 1.0



Data: 05/03/2019

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

Coordinatore del progetto:

Nome	Matricola
De Lucia Andrea	

Partecipanti:

Nome	Matricola
Antonio Scognamiglio	0512104868
Giuliana Muto	0512104598
Matteo Fasolino	0512103038

Scritto da:	Antonio Scognamiglio
--------------------	----------------------

Revision History

Data	Versione	Descrizione	Autore
05/03/2019	1.0	Glossario	Giuliana Muto
05/03/2019	1.0	Revisione	Giuliana Muto
05/03/2019	1.0	Descrizione del documento	Antonio Scognamiglio

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

Sommario

1. DESCRIZIONE DEL DOCUMENTO	4
2. RELAZIONE CON ALTRI DOCUMENTI	4
2.1 Relazione con RAD.....	4
2.2 Relazione con SDD	4
2.3 Relazione con ODD	4
3. DETTAGLI DEL LEVEL TESTING	5
3.1 Approccio di Unit Testing	5
3.2 Testing di unità	5
4. PASS / FAIL CRITERIA	5
5. RISULTATI TEST D'UNITA	6
5.1 Test Gestione Utente.....	6
5.2 Test Gestione Acquisto	8
5.3 Test Gestione Vendita	11
6. Glossario	12

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

1. DESCRIZIONE DEL DOCUMENTO

Il testing di unità rappresenta una delle fasi di testing più importanti, in quanto consiste nella verifica e il collaudo di singole unità software di un sistema. Tuttavia, questa fase è anche molto delicata e può essere causa di imprevisti e slittamenti dei tempi a causa di errori e malfunzionamento del sistema.

2. RELAZIONE CON ALTRI DOCUMENTI

Essendo in relazione con altri documenti, ne derivano vari criteri di accettazione del test. I documenti di test case Specification devono essere disponibili prima dell'inizio della fase di progettazione del test.

Per individuare correttamente i test case si terrà conto dei documenti prodotti precedentemente. Infatti, la fase di testing è strettamente legata alle fasi precedenti, in quanto saranno un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema desiderato sia simile a quello proposto.

2.1 Relazione con RAD

La relazione tra Unit Test Plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse in tale documento.

2.2 Relazione con SDD

Nel SDD abbiamo suddiviso il nostro sistema in sottosistemi e l'architettura in tre livelli: Presentation Layer, Application Layer e Storage Layer. Il test dei vari componenti deve rimanere fedele a queste suddivisioni il più possibile.

2.3 Relazione con ODD

Il test d'integrazione farà quanto più riferimento possibile alle interfacce delle classi e i package definiti nell'ODD.

Inoltre, lo sviluppo deve essere completato ed i risultati condivisi tra il team di test per evitare i difetti duplicati. Infine, l'ambiente di test deve essere installato, configurato e pronto per l'uso dell'applicazione.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

3. DETTAGLI DEL LEVEL TESTING

3.1 Approccio di Unit Testing

Il testing si compone di tre fasi. Inizialmente verranno eseguiti i test di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità andando a constatare il corretto funzionamento di tutte le singole unità di codice. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.

3.2 Testing di unità

Per effettuare il testing di ogni singola componente del sistema verrà utilizzata la tecnica “Black-Box testing” attraverso il framework JUnit. In questa fase saranno analizzate le funzionalità dell’applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura interna. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema. Se si verifica un errore con dei risultati inattesi si interviene in maniera tempestiva sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

4. PASS / FAIL CRITERIA

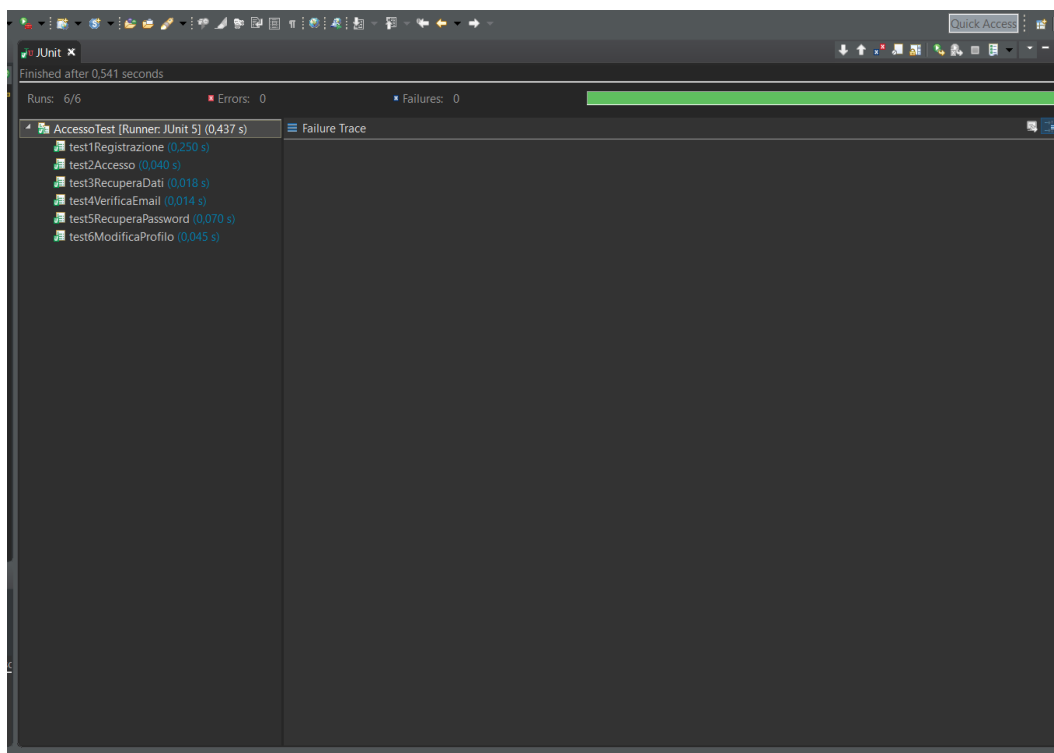
Il testing ha successo se l’output osservato è diverso dall’output atteso: ciò significa che la fase di testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. La fase di testing sarà iterata per verificare che la modifica non abbia portato impatti su altri componenti del sistema. Al contrario, il testing fallirà se l’output osservato sarà uguale all’oracolo.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

5. RISULTATI TEST D'UNITA

Di seguito sono stati riportati i risultati di tutti i test effettuati per le classi “Gestione” del sistema “MilleMigliaInViaggio” tramite testing JUNIT.

5.1 Test Gestione Utente

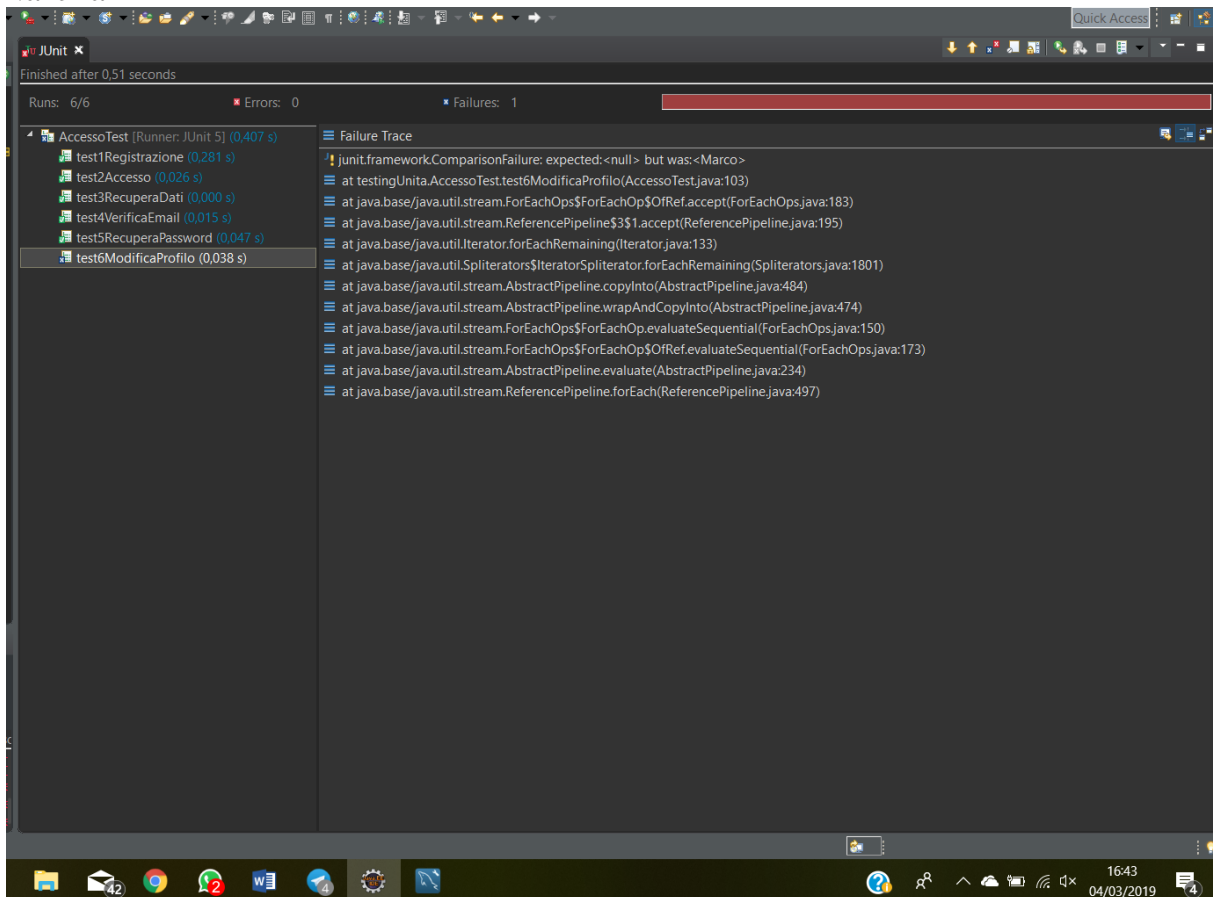


Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

Test Modifica Profilo:

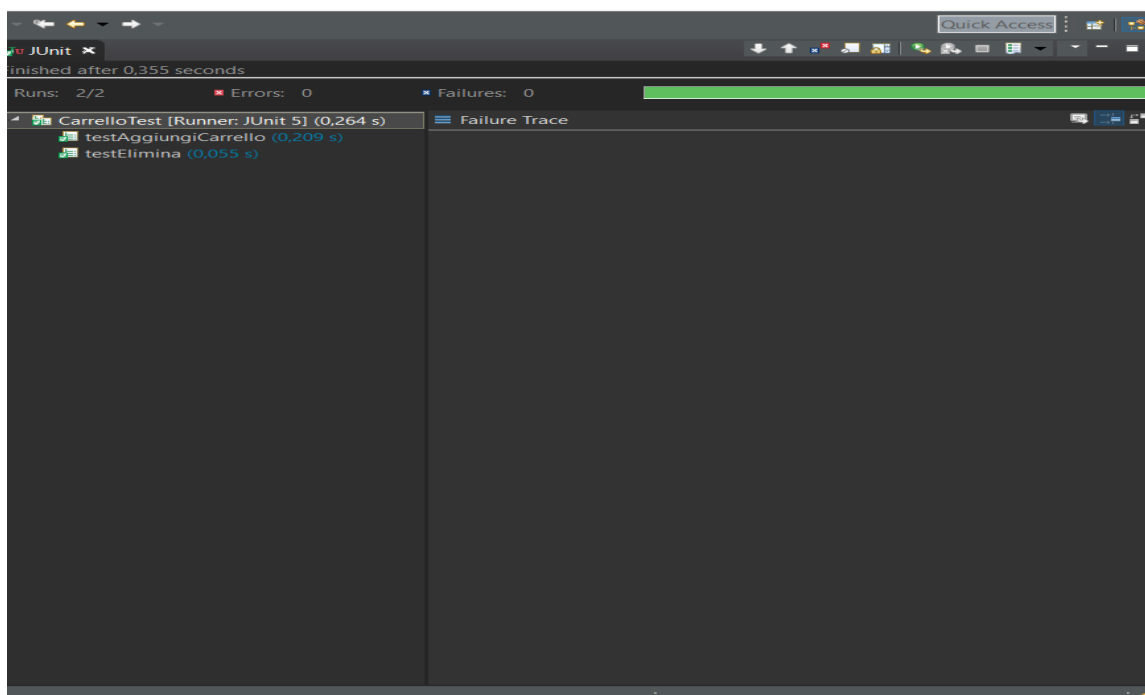
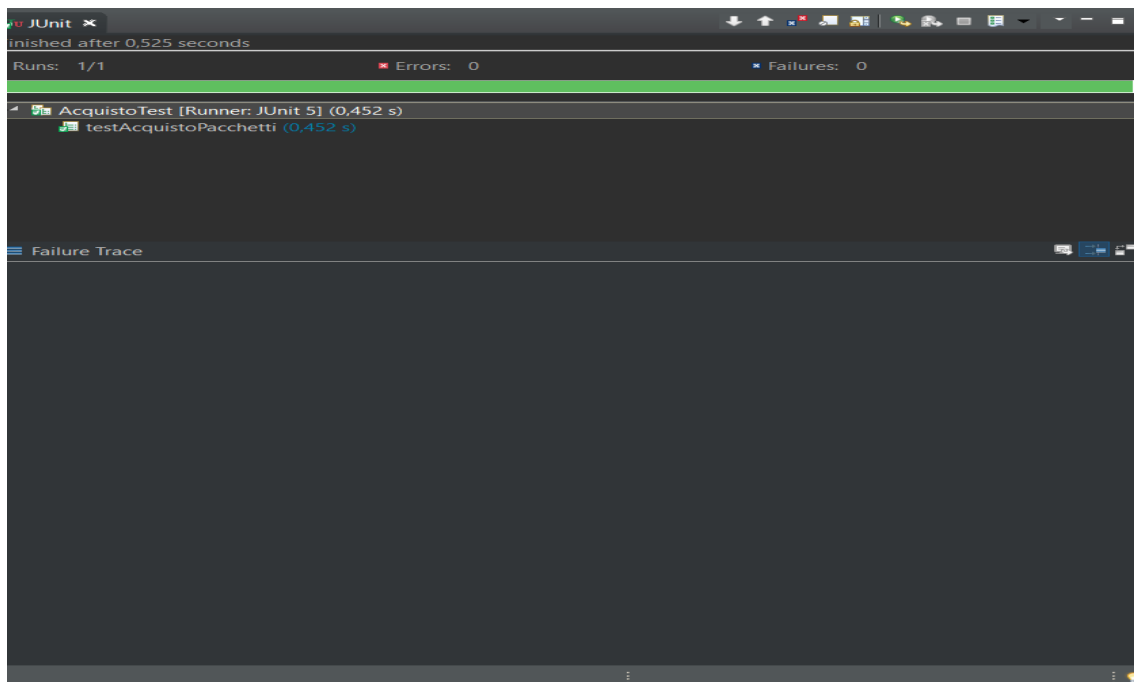
Errore: è stato riscontrato un errore in quanto non era possibile modificare il profilo dell'utente venditore, settando la data di nascita a "null" siccome il campo è di tipo Date.

Soluzione: L'errore è stato risolto con il cambio del tipo data di nascita con il tipo Varchar.



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

5.2 Test Gestione Acquisto

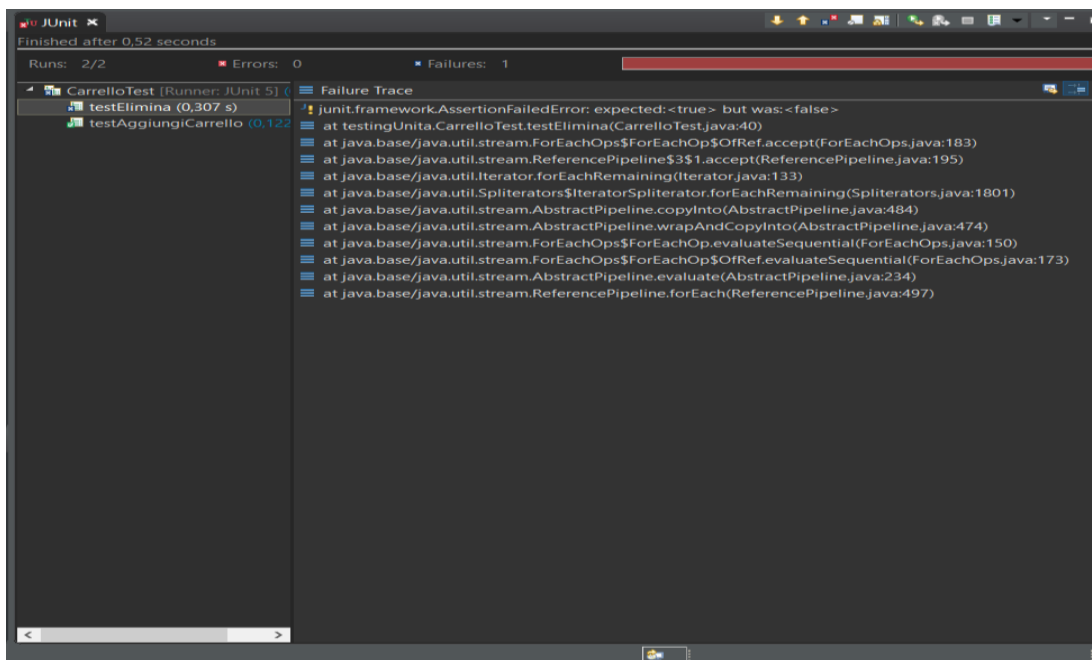


Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

Test Elimina Pacchetto Dal Carrello:

Errore: è stato riscontrato un errore all'interno della query, in quanto ,non permetteva la rimozione del pacchetto dal Database.

Soluzione: per risolvere l'errore e verificare che il metodo di rimozione funzioni è stata modificata la query ed è stato inserito un valore di tipo boolean che ritorna vero o falso a seconda dell'esito della query.

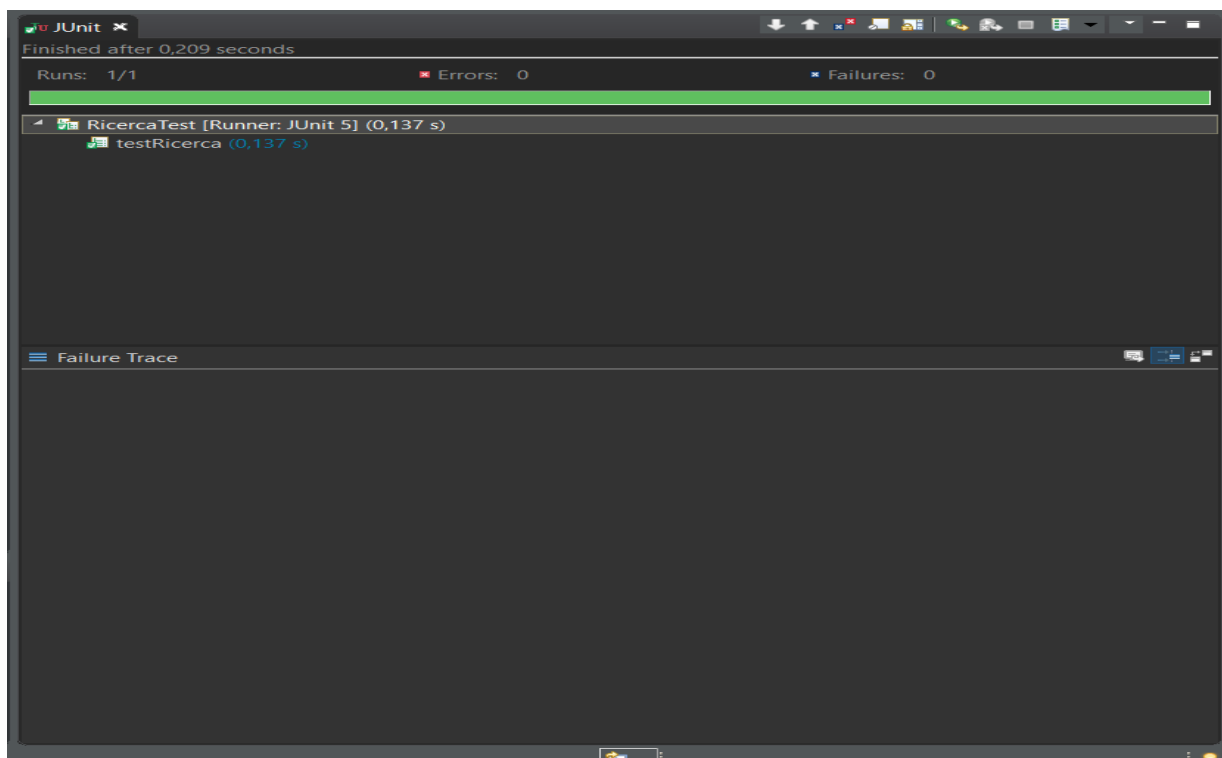
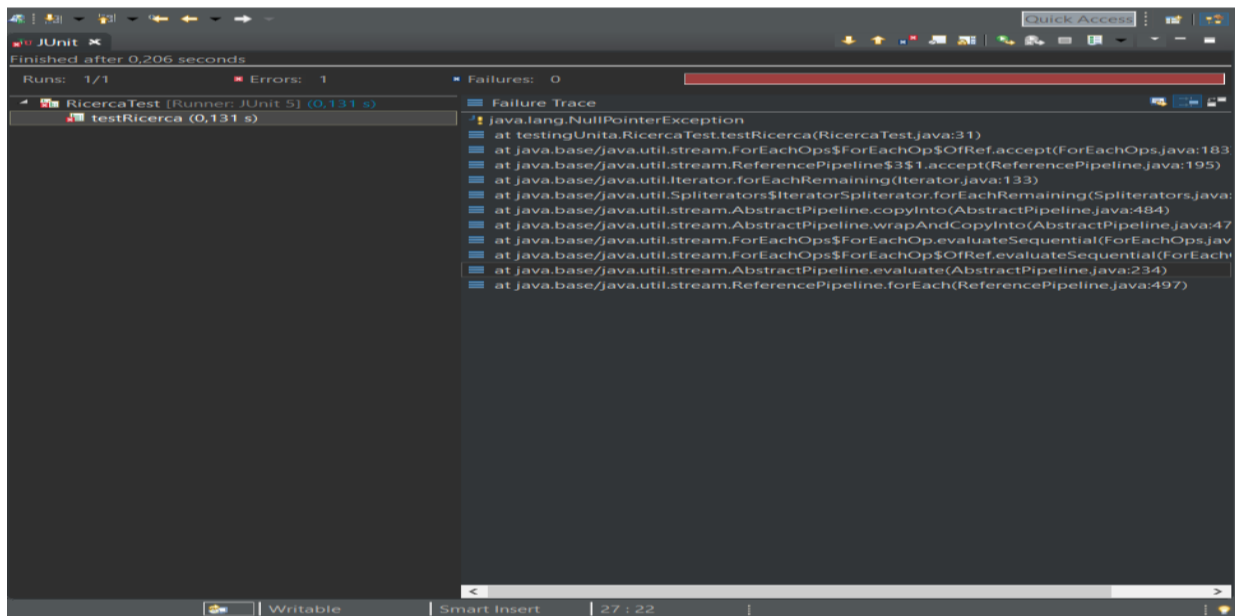


Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

Test Ricerca:

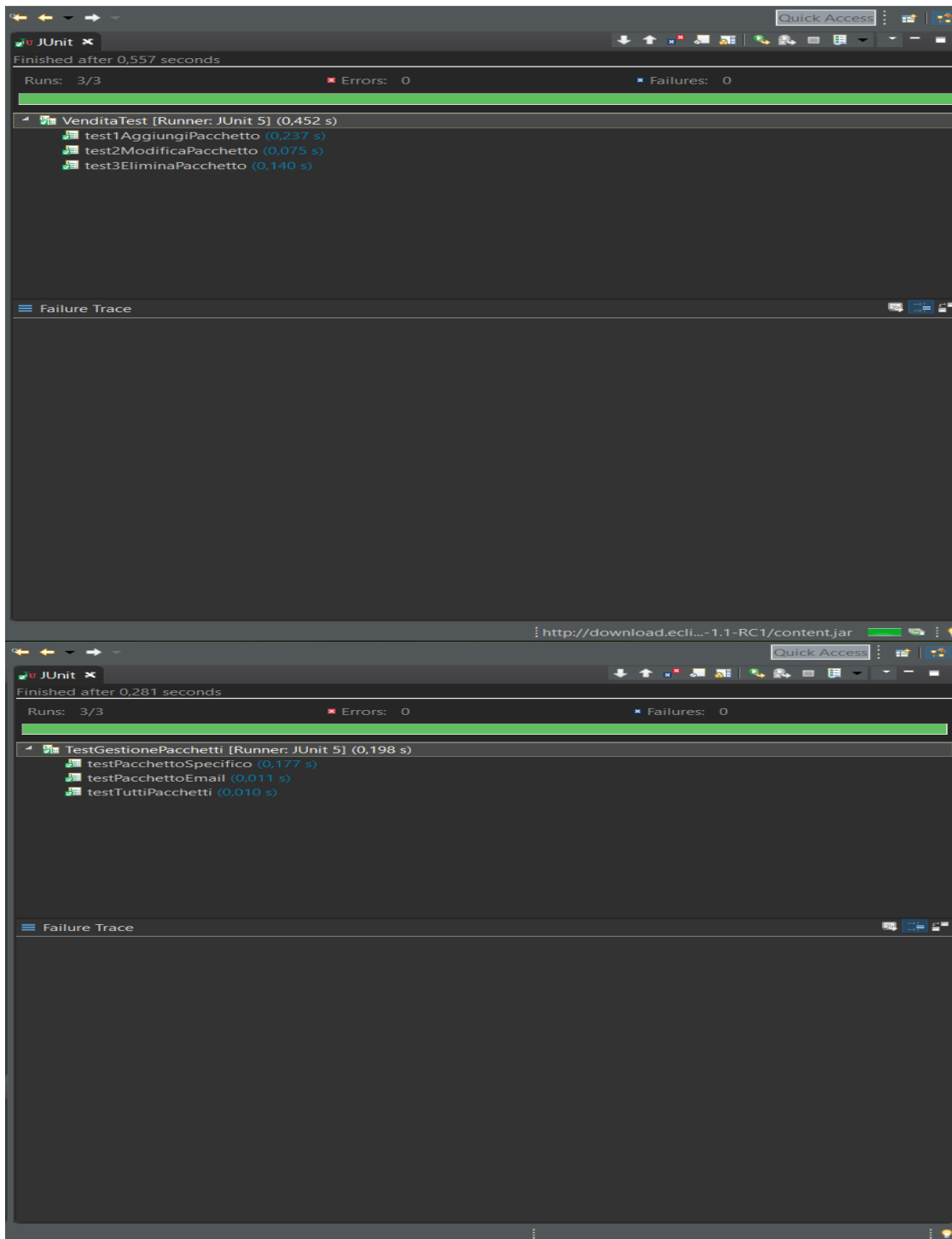
Errore: è stato riscontrato un errore nel metodo Ricerca(), in quanto i parametri (data) passati come input non erano corretti e quindi la query ritornava un valore null.

Soluzione: il problema è stato risolto facendo un subString alla data, estraendo così il mese e l'anno dalla data passandoli come parametri separatamente.



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

5.3 Test Gestione Vendita



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Unit Testing	Data: 05/03/2019

6.Glossario

Black-Box testing: i test case sono derivati dalle specifiche di una unità. Non presuppone conoscenza del codice sorgente;

JUnit: : framework di unit testing per il linguaggio di programmazione Java.;

Testing: processo o metodo per trovare gli errori in un'applicazione o un programma software in modo che l'applicazione funzioni in base ai requisiti dell'utente finale;

Componente: un blocco di programma riutilizzabile anche in combinazione con altre componenti;

Failure: Fallimento in una o più fasi di test;

Pass: Successo in una o più fasi di test;

Specific: Nota, distinta in cui sono specificati uno per uno gli elementi utili a un determinato fine;

Successo: Esito positivo;

Test: Esperimento variamente espletato allo scopo di saggiare, mediante determinate reazioni, l'entità o la consistenza di un'attitudine o di una capacità individuale;

Verifica: Operazione di controllo per mezzo della quale si procede all'accertamento di determinati risultati;

Unit-testing: attività di testing di singole unità software.