

Testing

Gabriele Chignoli

Luglio 2025

Contents

1	Introduzione	3
2	Attività di Test	3
2.1	Costruttore classe Product - 25/07/2025	3
	Obiettivo	3
	Input	3
	Aspettative e risultati	3
	Directory Componente	3
	Directory Test	3
2.2	ProductManager - 26/08/2025	3
	Obiettivo	3
	Input	4
	Aspettative e risultati	4
	Directory Componente	4
	Directory Test	4

1 Introduzione

Nel seguente documento vengono mostrate le varie attività di test eseguito sul software Pantrymanager. Si intende documentare per ogni componente testata il nome e la posizione del componente, la data di esecuzione, il tipo e l'obiettivo del test ed i risultati ottenuti.

2 Attività di Test

2.1 Costruttore classe Product - 25/07/2025

Obiettivo Viene testato il costruttore di oggetti *Product*

```
public Product(String name, float weight, int qty,
               float calories, LocalDate expiration_date) {
    ...
}
```

Input Sono stati utilizzati 5 valori differenti per testare il nome del prodotto, 5 per il peso, e solamente 1 per gli altri attributi. Sono stati inclusi anche valori che non hanno senso per quello che rappresentano, ma verranno vincolati più avanti e ad un livello superiore della struttura, verso l'interfaccia utente.

Aspettative e risultati Il test era molto semplice e testava una parte molto piccola di codice che difficilmente risulta problematica. L'obiettivo era prendere confidenza con JUnit, per poi provare a seguire in futuro anche un approccio guidato dei test.

I risultati del test vengono caricati attraverso un file xml esportato direttamente da Eclipse.

Directory Componente

```
../src/main/java/controller/Product.java
```

Directory Test

```
../src/test/java/controller/ProductTest.java
```

2.2 ProductManager - 26/08/2025

Obiettivo Viene testata la classe ProductManager che effettua le operazioni sul DataBase, in particolare i seguenti metodi:

```
public static int saveProduct(Product to_save) {
    return run(OperationMode.SAVE, to_save);
}

public static int deleteProduct(String name) {
    ...
    return run(OperationMode.DELETE, to_delete);
}

public static int modifyProduct(Product p) {
```

```

        return run(OperationMode.MODIFY, p);
    }

    public static List<Product> getProducts() {
        ...
        return entity_manager.createQuery(criteria).getResultList();
    }

```

Input Sono stati utilizzati diversi prodotti, per i quali viene verificato che le operazioni abbiano correttamente salvato, eliminato o modificato nel database `PANTRY.mv.db`.

Aspettative e risultati Il test voleva verificare principalmente una corretta interazione con il database, per la quale si considera il test superato; tuttavia ha anche evidenziato come gli input vengano semplicemente salvati senza essere controllati. In particolare, a parte il nome che non può essere omesso e che deve essere unico, gli altri parametri numerici possono assumere valori negativi o comunque non voluti se si agisce a questo livello.

Passando per l'interfaccia grafica questo problema viene evitato non permettendo all'utente di inserire certi valori; rimane da decidere se introdurre un'ulteriore verifica a questo stadio.

Directory Componente

```
../src/main/java/controller/ProductManager.java
```

Directory Test

```
../src/test/java/controller/ProductManagerTest.java
```