



Laurea Triennale in informatica-Università di Salerno  
Corso di *Ingegneria del Software*- Prof. C. Gravino

# Test Summary Report



Riferimento	
Versione	1.0
Data	03/01/2026
Destinatario	Docente Ingegneria del Software Prof. Carmine Gravino
Presentato da	Gaetano Aprile, Luigi Artuso, Alessandro De Bonis & Marco Galdi
Approvato da	



## 1. Introduzione

Il presente documento, Test Summary Report, ha lo scopo di riassumere le attività di collaudo effettuate sul modulo backend del sistema software MatchPoint.

Il documento descrive l'ambiente di test, gli strumenti utilizzati, le metriche di copertura raggiunte e l'esito delle esecuzioni dei casi di test (Test Cases) definiti nel documento Test Case Specification (TCS).

L'obiettivo principale è certificare che i componenti di Business Logic (Service Layer) soddisfino i requisiti funzionali e di qualità prefissati.

## 2. Ambiente di Esecuzione

I test sono stati eseguiti in ambiente locale con la seguente configurazione:

- **Hardware:** Workstation locale (Intel/AMD, 16GB RAM).
- **Sistema Operativo:** Windows 11.
- **Framework di Sviluppo:** Java 25 (OpenJDK), Spring Boot 4.0.1.
- **Framework di Test:** JUnit 5 (Jupiter), Mockito 5.20.
- **IDE:** IntelliJ IDEA 2025.2.3.
- **Database:** MySQL 8.0

## 3. Metodologia e Copertura Coverage

Per la verifica del software è stata adottata una strategia di Unit Testing focalizzata sui componenti critici di Business Logic (Service Layer). Le metriche di copertura sono state rilevate utilizzando il motore di analisi integrato in IntelliJ IDEA.

Metrica	Valore Rilevato	Dettaglio (Elementi Coperti / Totali)
Class Coverage	50%	7/14 Classi
Method Coverage	51%	14/27 Metodi
Line Coverage	50%	115/230 Linee di codice
Branch Coverage	41%	47/112 Rami decisionali

Sebbene la copertura globale del progetto si attesti intorno al 50%, l'analisi di dettaglio conferma che:

1. **Business Logic (Service Layer):** Raggiunge una copertura prossima al **100%**. Tutti i metodi dei servizi GestioneEvento, GestioneUtenti, GestioneValutazione e PartecipazioneEvento sono stati testati (inclusi i flussi di eccezione).



2. **Codice non coperto:** Il restante 50% è costituito da:

- **DTO e Modelli:** Metodi getter e setter (codice generato/banale che non richiede test unitari espliciti).
- **Controller Layer:** Esclusi dallo scope di questo Unit Test (verranno verificati tramite test di sistema o manuali con Swagger).
- **Componenti Infrastrutturali:** Classi wrapper (MappeServiceFacade, EmailService) le cui dipendenze esterne sono state mockate.
- **Classi di Configurazione:** (es. MatchPointApplication, Configuration classi).

Il team ritiene il livello di copertura soddisfacente in quanto garantisce la correttezza di tutte le funzionalità "core" del sistema, isolando la logica di business da componenti infrastrutturali non oggetto di questo ciclo di test.

## 4. Riepilogo Esecuzioni

Di seguito vengono riportati i risultati delle iterazioni di test. Come da prassi (consultare *Test Incident Report*), dopo il fallimento rilevato nella prima esecuzione, è stato effettuato il fix del codice e una successiva esecuzione completa (Regression Test).

### Tabella Riepilogativa Iterazioni

Esecuzione ID	Data	# Test Totali	# Passati	# Falliti	Esito Globale
Esecuzione 1	02/01/2026	15	14	1	<b>FAILED</b>
Esecuzione 2	03/01/2026	15	15	0	<b>PASSED</b>

### Dettaglio Fallimenti Risolti

Durante l'Esecuzione 1 è emerso il seguente incidente critico, descritto nel documento *Test Incident Report*:

- **Incidente:** TI\_UC4\_1
- **Componente:** GestioneEventoService
- **Descrizione:** Mancata validazione delle date nel passato durante la creazione evento.
- **Risoluzione:** Implementazione check temporale e validazione input. Verificato con successo nell' Esecuzione 2.