

Piano di Qualifica

RAMtastic6

11 giugno 2024



email: ramtastic6@gmail.com

Informazioni sul documento

Versione: 2.0.0
Redattori: Zaupa R. Zambon M. Brotto D. Visentin S. Basso L. Tonietto F.
Verificatori: Brotto D. Visentin S. Zaupa R. Tonietto F. Basso L. Zambon M.
Destinatari: T. Vardanega, R. Cardin, Imola Informatica
Uso: Esterno

Registro dei Cambiamenti - Changelog

Versione	Data	Autore	Verificatore	Dettaglio
v 2.0.0	2024-06-11	Zambon M.	Zambon M.	Approvazione e validazione del documento.
v 1.5.0	2024-06-11	Tonietto F.	Zambon M.	Aggiunte le metriche MPD01-CRO, MPC07-FD, MPC09-RSI ed MPC15-PCTS.
v 1.4.1	2024-06-11	Basso L.	Zambon M.	Aggiornati grafici MPC06, MPC07, MPC08 e MPC19 fino al dodicesimo periodo.
v 1.4.0	2024-06-10	Brotto D.	Basso L.	Aggiornato grafico MPC02 e MPC04 fino al dodicesimo periodo
v 1.3.2	2024-06-09	Zambon M.	Basso L.	Aggiornato MPC-01: grafico fino all'undicesimo periodo con considerazioni della fase PB_G
v 1.3.1	2024-06-08	Tonietto F.	Basso L.	Inserite le sezioni MPC-16 e MPC-17 relative a Statement e $Branch_G$ Coverage con relativi grafici.
v 1.3.0	2024-06-01	Zambon M.	Tonietto F.	Cambiate le descrizioni del TS-04 e TS-44, cambiati i requisiti dei $test_G$ di $sistema_G$ TS-08, TS-18, TS-35, TS-70 e TS-72 da ROF a RDF
v 1.2.5	2024-05-29	Visentin S.	Tonietto F.	Segnati i $test_G$ di accettazione come "superati" dopo l'approvazione del proponente
v 1.2.4	2024-05-28	Zaupa R. Visentin S.	Tonietto F.	Aggiunti i $test_G$ di accettazione.
v 1.2.3	2024-05-28	Brotto D.	Tonietto F.	Inseriti i $test_G$ di unità dei moduli $backend_G$ daysopen, food, menu, restaurant e staff. Aggiornamento della tabella di tracciabilità
v 1.2.2	2024-05-27	Brotto D.	Tonietto F.	Inseriti tutti i $test_G$ di unità (sezioni 3.1.1, 3.1.2, 3.1.3) (lato $backend_G$, $frontend_G$ e $websocket_G$) con relative tabelle di tracciabilità
v 1.2.1	2024-05-18	Brotto D.	Tonietto F.	Inserita sezione 3.1.1 con relativa tabella di tracciamento

v 1.2.0	2024-05-18	Zaupa R.	Tonietto F.	Inserita sezione 2.1.2.2 (Verifica) e inserite metriche MPC15-PCTS, MPC16-SC, MPC17-BC e MPC18-CC. Cambiato RDF33 a ROF33 e ROF76 a RDF76.
v 1.1.3	2024-05-17	Zaupa R.	Tonietto F.	Inserita sezione 1.4 (<i>Codifica_G</i> dei test) e impostata la sezione 3.1 (<i>Test_G</i> di unità).
v 1.1.2	2024-05-15	Brotto D.	Tonietto F.	Aggiunto <i>test_G</i> di sistema TS-77. In Requisiti di <i>Sistema_G</i> : Aggiornati Requisiti Funzionali 5,7 da desiderabili a obbligatori. Aggiornati Requisiti Funzionali 52,53,54,56,60 da obbligatori a desiderabili.
v 1.1.1	2024-05-13	Basso L.	Tonietto F.	Aggiunto <i>test_G</i> di <i>sistema_G</i> TS-76.
v 1.1.0	2024-05-04	Visentin S.	Zaupa R.	Aggiornata la sezione dei <i>Test_G</i> di <i>Sistema_G</i> in base alla versione 2.1.0 dell'Analisi dei Requisiti.
v 1.0.0	2024-04-29	Zambon M.	Zambon M.	Approvazione e validazione del documento
v 0.3.0	2024-04-28	Zambon M.	Visentin S.	Aggiunti grafici nella sezione 5
v 0.2.1	2024-04-28	Zambon M.	Visentin S.	Stesura sezione 5. (Analisi dei grafici)
v 0.2.0	2024-04-24	Brotto D.	Visentin S.	Aggiornata la sezione dei <i>Test_G</i> di <i>Sistema_G</i> (messa in linea con i ROF corretti nell'Analisi dei Requisiti). Inizio della sezione "Modifiche Migliorative".
v 0.1.3	2024-04-04	Zambon M.	Visentin S.	Cambiate alcuni termini che sono presenti nel glossario per risolvere alcuni problemi con l'automazione dei riferimenti
v 0.1.2	2024-03-26	Zambon M.	Visentin S.	Completata stesura della sottosezione " <i>Test_G</i> di <i>sistema_G</i> " nella sezione testing
v 0.1.1	2024-03-25	Zambon M.	Visentin S.	Iniziata stesura della sottosezione " <i>Test_G</i> di <i>sistema_G</i> " nella sezione testing. Definiti 25/85 test di <i>sistema_G</i>
v 0.1.0	2024-03-21	Zambon M.	Brotto D.	Cambiata metrica nella sezione "obiettivi di qualità": "sviluppo" (da "Complessità ciclomatica" a "Linee di codice")

v 0.0.4	2024-03-20	Zambon M.	Brotto D.	Stesura delle prime versioni delle sottosezioni di "Qualità di prodotto"
v 0.0.3	2024-03-13	Zaupa R.	Brotto D.	Stesura delle prime versioni della sezione "Testing" e della sezione "Checklist"
v 0.0.2	2024-03-12	Zaupa R.	Brotto D.	Stesura delle prime versioni delle sottosezioni di "Obiettivi di qualità": "Documentazione" e "Sviluppo"
v 0.0.1	2024-03-11	Zaupa R.	Brotto D.	Prima versione, identificazione delle sezioni principali, stesura della sezione 1 "Introduzione", della sezione introduttiva di "Qualità di <i>processo_G</i> " e di "Qualità di prodotto" e della sottosezione " <i>Fornitura_G</i> "

Indice

1	Introduzione	7
1.1	Scopo del documento	7
1.2	Scopo del prodotto	7
1.3	Riferimenti	7
1.3.1	Riferimenti normativi	7
1.3.2	Riferimenti informativi	7
1.4	Codifica delle metriche	8
1.5	Codifica dei test	8
2	Obiettivi di qualità	9
2.1	Qualità di processo	9
2.1.1	Processi primari	9
2.1.1.1	Fornitura	9
2.1.1.2	Sviluppo	10
2.1.2	Processi di supporto	11
2.1.2.1	Documentazione	11
2.1.2.2	Verifica	11
2.1.3	Processi organizzativi	12
2.2	Qualità di prodotto	13
2.2.1	Adeguatezza funzionale	13
2.2.2	Efficienza	15
2.2.3	Usabilità	15
2.2.4	Affidabilità	16
2.2.5	Manutenibilità	16
2.2.6	Portabilità	17
3	Testing	18
3.1	Test di unità	19
3.1.1	Test di unità - FrontEnd	19
3.1.2	Test di unità - BackEnd	24
3.1.3	Test di unità - WebSocket Server	36
3.2	Test di sistema	37
3.3	Test di accettazione	41
4	Checklist	44
4.1	Documentazione	44
4.1.1	Struttura	44
4.1.2	Errori ortografici	44
4.1.3	Analisi Dei Requisiti	45
5	Cruscotto di valutazione della qualità	46
5.1	Qualità di processo	46
5.1.1	MPC01-EAC (Estimated at Completion)	46
5.1.2	MPC02-PV (Planned Value) e MPC04-EV (Earned Value)	47
5.1.3	MPC05-ETC (Estimated to Complete)	48

5.1.4	MPC06-CV (Cost Variance), MPC07-SV (Schedule Variance) e MPC08-BV (Budget Variance)	48
5.1.5	MPC09-RSI (Requirements stability index)	49
5.1.6	MPC15-PCTS (Percentuale dei test superati)	49
5.1.7	MPC16-SC (Statement Coverage)	50
5.1.8	MPC17-BC (Branch Coverage)	50
5.1.9	MPC19-NCR (Non Calculated Risk)	51
5.2	Qualità di prodotto	52
5.2.1	MPD01-CRO (Copertura dei requisiti obbligatori)	52
5.2.2	MPC07-FD (Failure Density)	52

1 Introduzione

1.1 Scopo del documento

Il presente documento si propone di definire le metriche e le metodologie di controllo e misurazione necessarie per garantire la qualità del prodotto e del *processo_G*. In particolare, le metriche di valutazione del prodotto sono correlate ai requisiti e alle aspettative del fornitore. Il *Piano di Qualifica_G* è concepito per essere dinamico ed incrementale, in particolar modo per quanto riguarda le metriche descritte e mira a fornire una valutazione il più obiettiva possibile di ciò che è stato realizzato. Le procedure del *Way Of Working_G* devono essere costantemente osservate e migliorate, al fine di garantire che il prodotto soddisfi le aspettative del cliente e mantenga gli standard di qualità richiesti. Eventuali termini tecnici sono definiti all'interno del documento "Glossario Tecnico".

1.2 Scopo del prodotto

Il prodotto finale, realizzato tramite un'*Applicazione Web Responsive_G*, si propone di realizzare un *software_G* innovativo volto a semplificare il *processo_G* di *prenotazione_G* e *ordinazione_G* nei ristoranti, contribuendo a migliorare l'esperienza per clienti e ristoratori. In particolare, *Easy Meal* dovrà consentire agli utenti di personalizzare gli ordini in base alle proprie preferenze, allergie ed esigenze alimentari; interagire direttamente con lo staff del ristorante attraverso una chat integrata e in ultimo, consentire di dividere il conto tra i partecipanti al tavolo.

1.3 Riferimenti

1.3.1 Riferimenti normativi

1. *Norme di Progetto_G* v2.0.0
2. Presentazione del *capitolato_G* d'appalto C3 - Progetto *Easy Meal_G*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C3.pdf>
3. Regolamento del progetto didattico:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>

1.3.2 Riferimenti informativi

1. Lezione "*Progettazione software_G (T6)*" del corso di *Ingegneria del software_{GG}*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T6.pdf>
2. Lezione "*Qualità del software_G (T7)*" del corso di *Ingegneria del software_{GG}*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T7.pdf>
3. Lezione "*Qualità di processo_G (T8)*" del corso di *Ingegneria del software_G*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T8.pdf>
4. Lezione "*Verifica e validazione: introduzione (T9)*" del corso di *Ingegneria del software_G*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T9.pdf>
5. Lezione "*Verifica e validazione: analisi statica (T10)*" del corso di *Ingegneria del software_G*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T10.pdf>

6. Lezione "Verifica e validazione: analisi dinamica (T11)" del corso di *Ingegneria del software_G*:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T11.pdf>
7. Documento "Dichiarazione impegni v1.2":
https://github.com/RAMtastic6/Project14/blob/main/documenti/CANDIDATURA/documento_impegni_v1.2.pdf
8. Metriche di progetto (*Earned Value Analysis*):
https://it.wikipedia.org/wiki/Metriche_di_progetto
9. Glossario v2.0.0;
10. Analisi dei Requisiti v3.0.0.

1.4 Codifica delle metriche

In questa sottosezione verranno definite le metriche che utilizzeremo, utilizzando un codice standardizzato.

Una metrica è identificata dal seguente formato di codice:

$$M[\text{Tipo}][\text{Id}]-[\text{Acronimo}]$$

Dove:

- **M** sta per "Metrica"
- **Tipo** può essere PC (per un *processo_G*) o PD (per un prodotto)
- **Id** rappresenta un identificativo all'interno di una metrica di un certo tipo
- **Acronimo** indica l'acronimo del nome della metrica utilizzata

Per ciascuna metrica verranno fornite descrizioni, valori accettabili e valori preferibili.

1.5 Codifica dei test

In questa sottosezione verranno definiti i *test_G* che utilizzeremo, utilizzando un codice standardizzato.

Un *test_G* è identificato dal seguente formato di codice:

$$T[\text{Tipo}]-[\text{Id}]$$

Dove:

- **T** sta per "*Test_G*"
- **Tipo** può essere S (di *sistema_G*) o I (di integrazione) o U (di unità) oppure A (di accettazione)
- **Id** rappresenta un identificativo all'interno di un *test_G* di un certo tipo

Per ciascun *test_G* verranno fornite descrizioni e il loro stato se implementato o meno oltre che un loro tracciamento.

2 Obiettivi di qualità

2.1 Qualità di processo

La verifica della qualità si misura sul come sta andando il *processo_G*, si misura l'efficacia e l'efficienza. Dopo aver definito il *processo_G* in esame nel documento di *Norme di Progetto_G*, si deve dunque misurare l'efficacia e l'efficienza, imparando cosa trattenere e cosa correggere. Per farlo sceglieremo delle buone metriche di misurazione di qualità e anche buoni strumenti (serve oggettività e automazione).

2.1.1 Processi primari

2.1.1.1 Fornitura

Per definire le metriche adottate per i *processi primari_G* di *fornitura_G*, andremo a fare riferimento alla tecnica denominata *EVA (Earned Value Analysis)*, consente di calcolare in termini di tempo, denaro speso e valore del lavoro realizzato e di valutare la performance del progetto.

Tra queste individuiamo:

- ***Budget at Completion (BAC)***:
valore previsto iniziale per la realizzazione del progetto
- ***Estimated at Completion (EAC)***:
revisione del valore stimato per la realizzazione del progetto
- ***Planned Value (PV)***:
rappresenta il costo pianificato per realizzare le attività di progetto per la data stabilita (misurato in tempo o in denaro)
- ***Actual cost (AC)***:
rappresenta il costo effettivamente sostenuto alla data stabilita (misurato in tempo o in denaro)
- ***Earned Value (EV)***:
rappresenta il valore delle attività sostenute alla data stabilita (misurato in tempo o in denaro)
- ***Estimated to Complete (ETC)***:
valore stimato per completare le attività rimanenti necessarie per concludere il progetto
- ***Cost Variance (CV)***:
misura la variazione del valore ottenuto (EV) rispetto al costo effettivo (AC)
- ***Schedule Variance (SV)***:
misura la variazione del valore ottenuto (EV) rispetto al costo pianificato (PV)
- ***Budget Variance (BV)***:
misura la variazione dal costo attuale (AC) rispetto al costo atteso (CV)

Come riportato dal documento "*Dichiarazione impegni v1.2*" citato nella sezione 1.3.2 "*Riferimenti_G informativi*", il BAC corrisponde ad un valore di €11.520.

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPC01-EAC	Estimated at Completion	$\pm 5\%$ BAC	$= BAC$
MPC02-PV	Planned Value	≥ 0	$\leq BAC$
MPC03-AC	Actual Cost	≥ 0	$\leq EAC$
MPC04-EV	Earned Value	≥ 0	$\leq EAC$
MPC05-ETC	Estimated to Complete	≥ 0	$\leq EAC$
MPC06-CV	Cost Variance	$\geq -7.5\%$	≥ 0
MPC07-SV	Schedule Variance	$\geq -7.5\%$	≥ 0
MPC08-BV	Budget Variance	$\pm 10\%$	$= 0$

Tabella 2: Metriche per i processi di fornitura

2.1.1.2 Sviluppo

Analisi Dei Requisiti Nel caso dell'attività di *Analisi Dei Requisiti_G* andremo a definire le seguenti metriche:

- **Requirements stability index (RSI):**

misura la stabilità dei requisiti nel corso del tempo durante lo sviluppo del progetto. Si misura nel seguente modo:

$$RSI = 1 - \left(\frac{N_{\text{modifiche}}}{N_{\text{requisiti totali}}} \right) \times 100$$

Dove:

- $N_{\text{modifiche}}$ è il numero totale di modifiche apportate ai requisiti durante un periodo di tempo specifico.
- $N_{\text{requisiti totali}}$ è il numero totale di requisiti nel progetto.

Progettazione Nel caso delle attività di progettazione andremo a definire le seguenti metriche:

- **Structural fan-in (SFIN):**

misura la quantità dei moduli che utilizzano un modulo specifico. Un valore alto indica che molte parti del *sistema_G* dipendono da un modulo specifico.

- **Structural fan-out (SFOUT):**

misura il numero di moduli utilizzati da un modulo specifico. Un valore alto indica che il modulo preso in considerazione dipende da molti altri moduli.

Codifica Nel caso dell'attività di *codifica_G* andremo a definire le seguenti metriche:

- **Linee di codice (LOC):**

misura la complessità del *software_G* in base al numero di linee di codice sorgente.

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPC09-RSI	Requirements stability index	$\geq 70\%$	100%
MPC10-SFIN	Structural fan-in	-	Massimo
MPC11-SFOUT	Structural fan-out	-	Minimo
MPC12-LOC	Linee di codice	≤ 50.000	≤ 30.000

Tabella 3: Metriche per i processi di sviluppo

2.1.2 Processi di supporto

2.1.2.1 Documentazione

Per il $processo_G$ di documentazione andremo a definire le seguenti metriche:

- **Errori ortografici (EO):**
misura la quantità di errori ortografici individuati per documento
- **Indice Gulpease (IG):**
misura la leggibilità di un documento in lingua italiana. Si calcola nel seguente modo:

$$IG = 89 + \frac{300 \times (\text{num. frasi}) - 10 \times (\text{num. lettere})}{\text{num. parole}}$$

dove:

- num. frasi è il numero di frasi nel testo,
- num. lettere è il numero di lettere nel testo,
- num. parole è il numero di parole nel testo.

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPC13-EO	Errori ortografici	0	0
MPC14-IG	Indice Gulpease	30-100	50-100

Tabella 4: Metriche per i processi di documentazione

2.1.2.2 Verifica

Per il $processo_G$ di verifica andremo a definire le seguenti metriche:

- **Percentuale dei test_G superati (PCTS):**
misura la percentuale di $test_G$, definiti nella sezione *Testing*, superati. Si calcola nel seguente modo:

$$PCTS = \frac{\text{num. test}_G \text{ superati}}{\text{num. test}_G \text{ totali}} \times 100$$

- **Statement coverage (SC):**

misura la percentuale degli statement del codice coperti dai $test_G$, ovvero eseguiti almeno una volta. Si calcola nel seguente modo:

$$SC = \frac{\text{Statement eseguiti}}{\text{Statement totali}} \times 100$$

- **Branch coverage (BC):**

misura la percentuale dei rami decisionali, ovvero i percorsi di esecuzione, del codice coperti dai $test_G$, cioè eseguiti almeno una volta. Si calcola nel seguente modo:

$$BC = \frac{\text{Rami decisionali testati}}{\text{Rami decisionali totali}} \times 100$$

- **Condition coverage (CC):**

misura la percentuale di condizioni logiche del codice che sono state eseguite durante i $test_G$, ovvero eseguite almeno una volta. Si calcola nel seguente modo:

$$CC = \frac{\text{Condizioni logiche testate}}{\text{Condizioni logiche totali}} \times 100$$

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPC15-PCTS	Percentuale dei $test_G$ superati	$\geq 80\%$	100%
MPC16-SC	Statement coverage	$\geq 80\%$	100%
MPC17-BC	$Branch_G$ coverage	$\geq 80\%$	100%
MPC18-CC	Condition coverage	$\geq 80\%$	100%

Tabella 5: Metriche per i processi di verifica

2.1.3 Processi organizzativi

Per i *processi organizzativi $_G$* andremo ad utilizzare le seguenti metriche:

- **Non Calculated Risk (NCR):**

misura la quantità di *rischi $_G$* non previsti e non stimati

- **Efficienza temporale (ET):**

misura quanto tempo viene impiegato per attività produttive rispetto alle ore individuali:

$$ET = \frac{\text{Ore individuali}}{\text{Ore produttive}}$$

dove:

- *Ore individuali* rappresentano le ore che portano al raggiungimento di obiettivi
- *Ore produttive* rappresentano il tempo totale trascorso come ore di orologio

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPC19-NCR	Non Calculated Risk	≤ 4	0
MPC20-ET	Efficienza temporale	≤ 3	≤ 1

Tabella 6: Metriche per i processi organizzativi

2.2 Qualità di prodotto

In questa sezione si discute degli obiettivi che un prodotto $software_G$ di qualità dovrebbe avere. Di seguito vengono elencati gli obiettivi di qualità esterni.

- *Adeguatezza funzionale*: si riferisce alla capacità di fornire le funzionalità e le caratteristiche previste permettendo di soddisfare i requisiti funzionali.
- *Efficienza*: si riferisce alla capacità di fornire adeguate prestazioni e a quella di utilizzo delle risorse di $sistema_G$.
- *Usabilità*: si riferisce alla misura all'apprendimento, comprensione e all'operabilità del prodotto da parte dell'utente.
- *Affidabilità*: misura la capacità del prodotto di funzionare correttamente sotto determinate condizioni per un determinato periodo di tempo.
- *Sicurezza*: indica la capacità del $software_G$ di proteggere i dati sensibili e prevenire accessi non autorizzati oltre che violazioni della privacy.

Di seguito vengono elencati gli obiettivi di qualità interni.

- *Manutenibilità*: misura la facilità con cui il prodotto può essere modificato, aggiornato e corretto.
- *Portabilità*: misura la facilità con cui il prodotto può essere trasferito da un ambiente all'altro.

2.2.1 Adeguatezza funzionale

Per l'adeguatezza funzionale andremo ad utilizzare le seguenti metriche:

- ***Copertura dei requisiti obbligatori (CRO)***:
misura la percentuale dei requisiti obbligatori soddisfatti

$$CRO = \frac{N_{ros}}{N_{rot}} \times 100$$

dove:

- N_{ros} rappresentano il numero di requisiti obbligatori soddisfatti
- N_{rot} rappresentano il numero di requisiti obbligatori totali

- ***Copertura dei requisiti desiderabili (CRD)***:
misura la percentuale dei requisiti desiderabili soddisfatti

$$CRD = \frac{N_{rds}}{N_{rdt}} \times 100$$

dove:

- N_{rds} rappresentano il numero di requisiti desiderabili soddisfatti
- N_{rdt} rappresentano il numero di requisiti desiderabili totali

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPD01-CRO	Copertura dei requisiti obbligatori	100%	100%
MPD02-CRD	Copertura dei requisiti desiderabili	$\geq 50\%$	100%

Tabella 7: Metriche per l'adeguatezza funzionale

2.2.2 Efficienza

Per l'efficienza andremo ad utilizzare la seguente metrica:

- **Tempo di risposta media (TM):**
misura il tempo impiegato dal $software_G$ di gestire ed elaborare una richiesta fino al risultato finale.

Di seguito vengono riportati gli obiettivi di qualità per tale metrica:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPD03-TM	Tempo di risposta medio	3 secondi	2 secondi

Tabella 8: Metriche per l'efficienza

2.2.3 Usabilità

Per l'usabilità andremo ad utilizzare le seguenti metriche:

- **Tempo di apprendimento (TA):**
misura il tempo impiegato dall'utente di imparare ad utilizzare le funzionalità del $software_G$.
- **Raggiunta dell'obiettivo (RO):**
misura il numero di iterazioni necessarie all'utente per raggiungere il risultato voluto.
- **Errori dell'utente (EU):**
misura il numero di errori che l'utente compie prima di raggiungere l'obiettivo desiderato.

Di seguito vengono riportati gli obiettivi di qualità per tali metriche:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPD04-TA	Tempo di apprendimento	20 minuti	10 minuti
MPD05-RO	Raggiunta dell'obiettivo	7 click	4 click
MPD06-EU	Errori dell'utente	3	0

Tabella 9: Metriche per l'usabilità

2.2.4 Affidabilità

Per l'affidabilità andremo ad utilizzare la seguente metrica:

- **Failure density (FD):**
misura in percentuale l'affidabilità del $software_G$.

$$FD = \frac{N_{tf}}{N_{te}} \times 100$$

dove:

- N_{tf} rappresentano il numero di $test_G$ falliti
- N_{te} rappresentano il numero di $test_G$ effettuati

Di seguito vengono riportati gli obiettivi di qualità per tale metrica:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPD07-FD	Failure density	10%	0%

Tabella 10: Metriche per l'affidabilità

2.2.5 Manutenibilità

per la manutenibilità andremo ad utilizzare le seguente metrica:

- **Complessità ciclomatica (CC):**
misura la complessità del $software_G$ utilizzando il grafo di controllo del flusso

$$v(G) = e - n + p$$

dove:

- G rappresenta il grafo,
- e rappresenta il numero di archi in G,
- n rappresenta il numero di nodi in G,
- p rappresenta il numero delle componenti connesse da ogni arco.

Di seguito vengono riportati gli obiettivi di qualità per tale metrica:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPD08-CC	Complessità ciclomatica	≤ 15	≤ 10

Tabella 11: Metriche per la manutenibilità

2.2.6 Portabilità

per la portabilità andremo ad utilizzare le seguente metrica:

- **Browser supportati (BS):**

misura in percentuale le versioni dei browser supportati

$$BS = \frac{N_{vbs}}{N_{vbp}} \times 100$$

dove:

- N_{vbs} rappresenta il numero di versioni di browser supportate
- N_{vbp} rappresenta il numero di versioni di browser previste da supportare

Di seguito vengono riportati gli obiettivi di qualità per tale metrica:

Metrica	Nome	Valore Accettabile	Valore Preferibile
MPD09-BS	Browser supportati	100%	100%

Tabella 12: Metriche per la portabilità

3 Testing

In questa sezione vengono esplorate le metodologie di $test_G$ ing e la loro specifica. L'obiettivo è quello di seguire il "Modello a V_G " in cui ad ogni fase di sviluppo corrisponde una tipologia di $test_G$ da eseguire.

I $test_G$ si dividono in:

- **Test di unità, TU:**
vengono eseguiti sulle unità più semplici del codice. Viene fatta corrispondere all'attività di $codifica_G$ (*implementation*).
- **Test di integrazione, TI:**
vengono eseguiti per verificare la corretta integrazione tra le diverse unità $software_G$. Viene fatta corrispondere all'attività di progettazione.
- **Test di $sistema_G$, TS:**
verificano il corretto funzionamento dell'intero $sistema_G$ e, in particolare, che tutti i requisiti individuati siano soddisfatti. Viene fatta corrispondere all'attività di *Analisi Dei Requisiti*.
- **Test di accettazione, TA:**
verificano, alla presenza del committente, che il prodotto finale soddisfi tutti i requisiti. Se superati, si può procedere al $rilascio_G$ dello stesso.

Per ogni $test_G$ verrà fornito uno stato, codificato nel modo seguente:

- **S:** superato. Il $test_G$ è stato implementato ed ha avuto esito positivo;
- **NS:** non superato. Il $test_G$ è stato implementato ed ha avuto esito negativo;
- **NI:** non implementato.

3.1 Test di unità

I $test_G$ di unità sono una fase del $processo_G$ di $test_G$ ing il cui scopo è quello di verificare il corretto funzionamento delle singole componenti di codice. Viene intesa un'unità come singole funzioni o classi oppure, in generale, ogni singola entità di codice responsabile di svolgere specifici compiti all'interno del $sistema_G$. Per farlo viene utilizzato il $framework_G$ per il $test_G$ ing *Jest*. Di seguito vengono elencati i $test_G$, i quali avranno un codice identificativo, una descrizione e il corrispondente stato.

3.1.1 Test di unità - FrontEnd

Per ogni componente viene descritto il $test_G$ effettuato e se è stato soddisfatto

Codice	Descrizione	Stato
TU-01	FoodList - Verificare che il titolo del menù venga visualizzato correttamente	S
TU-02	FoodList - Verificare che gli item nel menù vengano visualizzati correttamente	S
TU-03	FoodList - Verificare i nomi ed i prezzi dei pasti nel menù vengano visualizzati correttamente	S
TU-04	FoodList - Verificare che se il menù non contiene item al suo interno, visualizzi una lista vuota	S
TU-05	Header - Verificare che l'header venga visualizzato correttamente	S
TU-06	HomePageCards - Verificare che l'homepage venga visualizzata correttamente	S
TU-07	IngredientChart - Verificare che gli ingredienti di un pasto vengano visualizzati correttamente	S
TU-08	IngredientChart - Verificare che la rimozione dei un ingrediente da un pasto avvenga correttamente	S
TU-09	IngredientChart - Verificare che l' $ordinazione_G$ venga aggiornata correttamente	S
TU-10	IngredientChart - Verificare che vengano chiamati gli eventi relativi al socket correttamente	S
TU-11	Login - Verificare che la schermata di login venga visualizzata correttamente	S
TU-12	Login - Verificare che il login ottenga gli input correttamente	S
TU-13	MenuTable - Verificare che il menù venga visualizzato correttamente	S
TU-14	MenuTable - Verificare che l'aumento di quantità di un pasto venga visualizzato correttamente	S
TU-15	MenuTable - Verificare che l'abbassamento di quantità di un pasto venga visualizzato correttamente	S
TU-16	MenuTable - Verificare che il prezzo totale venga visualizzato correttamente	S
TU-17	MenuTable - Verificare che vengano chiamati gli eventi relativi al socket correttamente	S
TU-18	Notification - Verificare che le notifiche vengano visualizzate correttamente	S
TU-19	Notification - Verificare che il click sul bottone nella notifica avvenga correttamente	S
TU-20	OrderCart - Verificare che la visualizzazione di un ordine avvenga correttamente	S
TU-21	OrderCart - Verificare il cambio del metodo di pagamento avvenga e venga visualizzato correttamente	S
TU-22	Pagination - Verificare che i link alle pagine corrispondano correttamente	S
TU-23	Pagination - Verificare la freccia alla pagina successiva nell'ultima pagina venga disabilitata correttamente	S

TU-24	Pagination - Verificare la freccia alla pagina precedente nella prima pagina venga disabilitata correttamente	S
TU-25	Pagination - Verificare che dopo un click si navighi verso la pagina corretta	S
TU-26	ReservationAdmin - Verificare la corretta visualizzazione del caricamento	S
TU-27	ReservationAdmin - Verificare che i dettagli di una <i>prenotazione_G</i> vengano visualizzati correttamente dall'amministratore	S
TU-28	ReservationAdmin - Verificare lo stato "In attesa di conferma" di una <i>prenotazione_G</i> dall'amministratore	S
TU-29	ReservationAdmin - Verificare lo stato "Rifiutata" di una <i>prenotazione_G</i> dall'amministratore	S
TU-30	ReservationAdmin - Verificare lo stato "Completata" di una <i>prenotazione_G</i> dall'amministratore	S
TU-31	ReservationForm - Verificare che il form per effettuare una <i>prenotazione_G</i> venga visualizzato correttamente	S
TU-32	ReservationForm - Verificare che il form per effettuare una <i>prenotazione_G</i> ottenga gli input correttamente	S
TU-33	ReservationUser - Verificare la corretta visualizzazione del caricamento della <i>prenotazione_G</i>	S
TU-34	ReservationUser - Verificare che la <i>prenotazione_G</i> è in attesa di conferma da parte dell'amministratore	S
TU-35	ReservationUser - Verificare che la <i>prenotazione_G</i> è stata accettata e le ordinazioni sono in attesa di conferma	S
TU-36	ReservationUser - Verificare che la <i>prenotazione_G</i> è stata rifiutata dall'amministratore	S
TU-37	ReservationUser - Verificare che le ordinazioni sono state confermate e che la <i>prenotazione_G</i> è in attesa di pagamento	S
TU-38	ReservationUser - Verificare che la <i>prenotazione_G</i> è stata pagata e completata	S
TU-39	ReservationsAdmin - Verificare la corretta visualizzazione del caricamento della lista di prenotazioni all'amministratore	S
TU-40	ReservationsAdmin - Verificare che ,se non ci sono prenotazioni, venga visualizzato correttamente "Non è stata effettuata nessuna <i>prenotazione_G</i> "	S
TU-41	ReservationsAdmin - Verificare la corretta visualizzazione della lista di prenotazioni all'amministratore dopo il caricamento	S
TU-42	ReservationsUser - Verificare la corretta visualizzazione del caricamento della lista prenotazioni all'utente	S
TU-43	ReservationsUser - Verificare che ,se non ci sono prenotazioni, venga visualizzato correttamente "Non è stata effettuata nessuna <i>prenotazione_G</i> "	S
TU-44	ReservationsUser - Verificare la corretta visualizzazione della lista prenotazioni all'utente dopo il caricamento	S
TU-45	RestaurantSearch - Verificare che la ricerca di un ristorante venga visualizzata correttamente	S
TU-46	RestaurantSearch - Verificare che la ricerca di un ristorante ottenga gli input correttamente	S
TU-47	RestaurantsTable - Verificare la corretta visualizzazione del caricamento	S
TU-48	RestaurantsTable - Verificare la corretta visualizzazione dopo il caricamento	S

TU-49	SignupAdmin - Verificare la corretta visualizzazione dopo il caricamento	S
TU-50	SignupAdmin - Verificare che gli input vengano ottenuti correttamente	S
TU-51	Signup - Verificare la corretta visualizzazione dopo il caricamento	S
TU-52	Signup - Verificare che gli input vengano ottenuti correttamente	S
TU-53	validateSignUp - Verificare che , se manca l'email, venga visualizzato un errore correttamente	S
TU-54	validateSignUp - Verificare che , se l'email non è valida, venga visualizzato un errore correttamente	S
TU-55	validateSignUp - Verificare che , se manca il nome, venga visualizzato un errore correttamente	S
TU-56	validateSignUp - Verificare che , se manca il cognome, venga visualizzato un errore correttamente	S
TU-57	validateSignUp - Verificare che , se manca la password, venga visualizzato un errore correttamente	S
TU-58	validateSignUp - Verificare che , se manca la conferma password, venga visualizzato un errore correttamente	S
TU-59	validateSignUp - Verificare che , se le password non coincidono, venga visualizzato un errore correttamente	S
TU-60	validateSignUp - Verificare che la registrazione venga effettuata con successo	S
TU-61	validateSignUpAdmin - Verificare che , se l'email non è valida, venga visualizzato un errore correttamente	S
TU-62	validateSignUpAdmin - Verificare che , se manca il nome, venga visualizzato un errore correttamente	S
TU-63	validateSignUpAdmin - Verificare che , se manca il cognome, venga visualizzato un errore correttamente	S
TU-64	validateSignUpAdmin - Verificare che , se le password non coincidono, venga visualizzato un errore correttamente	S
TU-65	validateSignUpAdmin - Verificare che , se manca il nome del ristorante, venga visualizzato un errore correttamente	S
TU-66	validateSignUpAdmin - Verificare che , se manca il nome della città, venga visualizzato un errore correttamente	S
TU-67	validateSignUpAdmin - Verificare che , se manca l'indirizzo del ristorante, venga visualizzato un errore correttamente	S
TU-68	validateSignUpAdmin - Verificare che , se manca il numero di coperti, venga visualizzato un errore correttamente	S
TU-69	validateSignUpAdmin - Verificare che , se manca il numero di telefono, venga visualizzato un errore correttamente	S
TU-70	validateSignUpAdmin - Verificare che , se manca l'email del ristorante, venga visualizzato un errore correttamente	S
TU-71	validateSignUpAdmin - Verificare che , se manca la tipologia di cucina, venga visualizzato un errore correttamente	S
TU-72	validateSignUpAdmin - Verificare che che la registrazione venga effettuata con successo	S
TU-73	ReservationAdmin - Verificare lo stato "Da pagare" di una <i>prenotazione_G</i> dall'amministratore	S

TU-74	ReservationAdmin - Verificare lo stato "Accettata" di una <i>prenotazione_G</i> dall'amministratore	S
--------------	---	---

Tabella 13: Test di unità BackEnd

Tracciamento *test_G* di unità FrontEnd

Codice	Fonte
TU-01	food_list.test.jsx
TU-02	
TU-03	
TU-04	
TU-05	header.test.jsx
TU-06	homepage_cards.test.jsx
TU-07	ingridient_chart.test.jsx
TU-08	
TU-09	
TU-10	
TU-11	login.test.jsx
TU-12	
TU-13	menu_table.test.jsx
TU-14	
TU-15	
TU-16	
TU-17	
TU-18	notification.test.jsx
TU-19	
TU-20	order_cart.test.jsx
TU-21	
TU-22	pagination.test.jsx
TU-23	
TU-24	
TU-25	
TU-26	reservation_admin.test.jsx
TU-27	
TU-28	
TU-29	
TU-30	
TU-73	
TU-74	
TU-31	reservation_form.test.jsx
TU-32	

TU-33 TU-34 TU-35 TU-36 TU-37 TU-38	<code>reservation_user.test.jsx</code>
TU-39 TU-40 TU-41	<code>reservations_admin.test.jsx</code>
TU-42 TU-43 TU-44	<code>reservations_user.test.jsx</code>
TU-45 TU-46	<code>restuarant_search.test.jsx</code>
TU-47 TU-48	<code>restuarant_table.test.jsx</code>
TU-49 TU-50	<code>sign_up_admin.test.jsx</code>
TU-51 TU-52	<code>sign_up.test.jsx</code>
TU-53 TU-54 TU-55 TU-56 TU-57 TU-58 TU-59 TU-60	<code>ValidateSignUp.test.jsx</code>
TU-61 TU-62 TU-63 TU-64 TU-65 TU-66 TU-67 TU-68 TU-69 TU-70 TU-71 TU-72	<code>ValidateSignUpAdmin.test.jsx</code>

Tabella 14: Tracciamento dei test di unità FrontEnd

3.1.2 Test di unità - BackEnd

Per ogni funzione viene descritto il $test_G$ effettuato e se è stato soddisfatto

Codice	Descrizione	Stato
TU-75	SignIn - Verificare che ritorni un token quando vengono immesse delle credenziali valide	S
TU-76	SignIn - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException quando l'utente non è stato trovato oppure la password non è corretta	S
TU-77	decodeToken - Verificare che ritorni un token decodificato quando viene immesso un token valido	S
TU-78	decodeToken - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException quando manca il token	S
TU-79	decodeToken - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException quando viene immesso un token non valido	S
TU-80	generateToken - Verificare che generi un token	S
TU-81	signinUser - Verificare che ritorni nullo quando l'utente non esiste	S
TU-82	signinUser - Verificare che ritorni nullo quando la password non corrisponde	S
TU-83	signinUser - Verificare che ritorni un token se l'utente esiste la password corrisponde	S
TU-84	verifyToken - Verificare che ritorni true se il token è valido	S
TU-85	verifyToken - Verificare che ritorni false se il token non è valido	S
TU-86	DaysopenController - Verificare che il controller sia definito	S
TU-87	create - Verificare che chiami il servizio con i corretti parametri	S
TU-88	DaysopenService - Verificare che il servizio sia definito	S
TU-89	create - Verificare che chiami il servizio con i corretti parametri	S
TU-90	create_manager - Verificare che crei le voci dei giorni di apertura con uno specifico manager	S
TU-91	findOne - Verificare che ritorni un cibo da un id dato	S
TU-92	findOne - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se il cibo non è stato trovato	S
TU-93	FoodService - Verificare che il servizio sia definito	S
TU-94	findOne - Verificare che chiami la funzione foodRepository.findOne con un id dato	S
TU-95	MenuController - Verificare che il controller sia definito	S
TU-96	create - Verificare che chiami la funzione menuRepository.create con i corretti parametri	S
TU-97	MenuService - Verificare che il servizio sia definito	S
TU-98	create - Verificare che crei un nuovo menù	S
TU-98	findAllByUserId - Verificare che ritorni le notifiche per un utente valido	S
TU-100	findAllByUserId - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException se il token non è valido	S
TU-101	updateStatus - Verificare che aggiorni lo stato di una notifica per un utente valido	S
TU-102	updateStatus - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException per un token o un id notifica non validi	S
TU-103	NotificationService - Verificare che il servizio sia definito	S
TU-104	create - Verificare che crei una nuova notifica	S
TU-105	findAllByUserId - Verificare che trovi tutte le notifiche dato un id utente	S
TU-106	updateStatus - Verificare che aggiorni lo stato di una notifica	S

TU-107	updateStatus - Verificare che ritorni null se la notifica non è stata trovata	S
TU-108	findOne - Verificare che ritorni una notifica dato il suo id	S
TU-109	create - Verificare che crei un ordine	S
TU-110	create - Verificare che lanci l' <i>eccezione_G</i> BadRequestException se il token non è valido	S
TU-111	create - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se l'ordine esiste già	S
TU-112	findAll - Verificare che ritorni tutti gli ordini	S
TU-113	create - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se l'ordine non è stato trovato	S
TU-114	findOne - Verificare che ritorni un ordine	S
TU-115	findOne - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se l'ordine non è stato trovato	S
TU-116	update - Verificare che aggiorni un ordine	S
TU-117	update - Verificare che lanci l' <i>eccezione_G</i> BadRequestException se il token non è valido	S
TU-118	update - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se l'ordine non è stato trovato	S
TU-119	updateIngredients - Verificare che aggiorni gli ingredienti di un ordine	S
TU-120	updateIngredients - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se l'ordine non è stato trovato	S
TU-121	remove - Verificare che rimuova un ordine	S
TU-122	remove - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se l'ordine non è stato trovato	S
TU-123	partialBill - Verificare che calcoli il conto parziale di un cliente	S
TU-124	RomanBill - Verificare che calcoli il conto intero di un cliente	S
TU-125	updateListOrders - Verificare che aggiorni la lista degli ordini per una <i>prenotazione_G</i>	S
TU-126	updateListOrders - Verificare che lanci l' <i>eccezione_G</i> BadRequestException se il token non è valido	S
TU-127	getReservationOrders - Verificare che ritorni gli ordini di una <i>prenotazione_G</i>	S
TU-128	getReservationOrders - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se nono sono stati trovati ordini per una <i>prenotazione_G</i>	S
TU-129	pay - Verificare che ritorni true se il pagamento è stato effettuato con successo	S
TU-130	pay - Verificare che ritorni null l'ordine non esiste	S
TU-131	create - Verificare che crei un ordine e lo salvi nel database	S
TU-132	remove - Verificare che rimuova un ordine dal database	S
TU-133	findAll - Verificare che ritorni tutti gli ordini nel database	S
TU-134	findOne - Verificare che ritorni uno specifico ordine dal database	S
TU-135	addQuantity - Verificare che aumenti la quantità per un ordine esistente	S
TU-136	updateIngredients - Verificare che aggiorni gli ingredienti per un ordine esistente	S
TU-137	getPartialBill - Verificare che ritorni correttamente il costo totale per un cliente	S
TU-138	getPartialBill - Verificare che ritorni 0 se il cliente non ha ordini nella <i>prenotazione_G</i>	S
TU-139	getRomanBill - Verificare che ritorni correttamente il costo per persona	S
TU-140	getRomanBill - Verificare che ritorni 0 se non ci sono ordini	S
TU-141	getReservationOrders - Verificare che ritorni tutti gli ordini di una specifica <i>prenotazione_G</i>	S

TU-142	updateListOrders - Verificare che ritorni la lista degli ordini per una specifica persona e <i>prenotazione_G</i>	S
TU-143	pay - Verificare che segni gli ordini come pagati e aggiorni lo stato della <i>prenotazione_G</i> se tutti gli ordini sono stati pagati	S
TU-144	pay - Verificare che segni gli ordini come pagati ma non aggiorni lo stato della <i>prenotazione_G</i> se tutti gli ordini non sono stati pagati	S
TU-145	pay - Verificare che ritorni null se l'ordine non esiste	S
TU-146	ReservationController - Verificare che il controller sia definito	S
TU-147	create - Verificare che crei una <i>prenotazione_G</i>	S
TU-148	create - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException se il token non è valido	S
TU-149	create - Verificare che lanci l' <i>eccezione_G</i> BadRequestException se la <i>prenotazione_G</i> non è valida	S
TU-150	addCustomer - Verificare che aggiunga un utente alla prenotazione	S
TU-151	addCustomer - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-152	findAll - Verificare che ritorni tutte le prenotazioni	S
TU-153	findAll - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-154	findOne - Verificare che ritorni una <i>prenotazione_G</i> dato un id	S
TU-155	findOne - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-156	getMenuWithOrdersQuantityByIdReservation - Verificare che ritorni il menù con la quantità di ordini dato un id <i>prenotazione_G</i>	S
TU-157	getMenuWithOrdersQuantityByIdReservation - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-158	getReservationsByRestaurantId - Verificare che ritorni delle prenotazioni dati degli id	S
TU-159	acceptReservation - Verificare che accetti una <i>prenotazione_G</i>	S
TU-160	acceptReservation - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-161	rejectReservation - Verificare che rifiuti una <i>prenotazione_G</i>	S
TU-162	rejectReservation - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-163	completeReservation - Verificare che completi una <i>prenotazione_G</i>	S
TU-164	completeReservation - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S
TU-165	getReservationsByUserId - Verificare che ritorni una <i>prenotazione_G</i> tramite un id utente	S
TU-166	verifyReservation - Verificare che verifichi una <i>prenotazione_G</i>	S
TU-167	verifyReservation - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException se il token non è valido	S
TU-168	verifyReservation - Verificare che lanci l' <i>eccezione_G</i> NotFoundException se la <i>prenotazione_G</i> non è stata trovata	S

TU-169	getReservationsByAdminId - Verificare che ritorni una <i>prenotazione_G</i> tramite un id admin	S
TU-170	getReservationsByAdminId - Verificare che lanci l' <i>eccezione_G</i> UnauthorizedException se il token non è valido	S
TU-171	ReservationService - Verificare che il servizio sia definito	S
TU-172	ReservationService - Verificare che reservationRepo sia definito	S
TU-173	create - Verificare che crei una <i>prenotazione_G</i>	S
TU-174	create - Verificare che ritorni null se il ristorante non p stato trovato	S
TU-175	create - Verificare che ritorni status:null e message:"Restaurant is full" se il ristorante è pieno	S
TU-176	create - Verificare che ritorni null se la data di <i>prenotazione_G</i> è passata	S
TU-177	addCustomer - Verificare che aggiunga una persona a una <i>prenotazione_G</i>	S
TU-178	addCustomer - Verificare che ritorni null se la <i>prenotazione_G</i> non è stata trovata	S
TU-179	findAll - Verificare che ritorni tutte le prenotazioni	S
TU-180	findOne - Verificare che ritorni una <i>prenotazione_G</i> tramite un id	S
TU-181	findOne - Verificare che ritorni null se la <i>prenotazione_G</i> non è stata trovata	S
TU-182	getMenuWithOrdersQuantityByIdReservation - Verificare che ritorni gli ordini di una <i>prenotazione_G</i> con la loro quantità	S
TU-183	getMenuWithOrdersQuantityByIdReservation - Verificare che ritorni null se la <i>prenotazione_G</i> non è stata trovata	S
TU-184	getReservationsByRestaurantId - Verificare che ritorni delle prenotazioni dall'id di una ristorante	S
TU-185	completeReservation - Verificare che completi una prenotazioni	S
TU-186	completeReservation - Verificare che ritorni null se la <i>prenotazione_G</i> non è stata trovata	S
TU-187	updateStatus - Verificare che aggiorni lo stato di una <i>prenotazione_G</i>	S
TU-188	updateStatus - Verificare che ritorni false se la <i>prenotazione_G</i> non è stata trovata	S
TU-189	getReservationsByUserId - Verificare che ritorni una <i>prenotazione_G</i> da un id utente	S
TU-190	verifyReservation - Verificare che ritorni una <i>prenotazione_G</i> se esiste e se l'utente è associato ad essa	S
TU-191	verifyReservation - Verificare che ritorni null se una <i>prenotazione_G</i> non esiste	S
TU-192	getReservationsByAdminId - Verificare che ritorni una <i>prenotazione_G</i> tramite id admin	S
TU-193	RestaurantController - Verificare che il controller sia definito	S
TU-194	getFilteredRestaurants - Verificare che chiami la funzione restaurantService.getFilteredRestaurants con un query data	S
TU-195	create - Verificare che chiami la funzione restaurantService.create con un elemento createRestaurantDto dato	S
TU-196	create - Verificare che lanci l' <i>eccezione_G</i> BadRequestException se il risultato è null	S
TU-197	findAll - Verificare che chiami la funzione restaurantService.findAll	S
TU-198	findAllCuisines - Verificare che chiami la funzione restaurantService.findAllCuisines	S
TU-199	findAllCities - Verificare che chiami la funzione restaurantService.findAllCities	S
TU-200	getNumberOfFilteredRestaurants - Verificare che chiami la funzione restaurantService.getNumberOfFilteredRestaurants con una query data	S
TU-201	findOne - Verificare che chiami la funzione restaurantService.findOne con un id dato	S

TU-202	getBookedTables - Verificare che chiami la funzione restaurantService.getBookedTables dati un id ed una data	S
TU-203	getRestaurantAndMenuByRestaurantId - Verificare che chiami la funzione restaurantService.getRestaurantAndMenuByRestaurantId dato un id	S
TU-204	getRestaurantAndMenuByRestaurantId - Verificare che lanci l' <i>eccezione</i> _G NotFoundException se il risultato è nullo	S
TU-205	RestaurantService - Verificare che il servizio sia definito	S
TU-206	RestaurantService - Verificare che la repo sia definita	S
TU-207	getFilteredRestaurants - Verificare che ritorni un array di ristoranti	S
TU-208	create - Verificare che crei un ristorante	S
TU-209	create - Verificare che lanci un errore se il ristorante esiste già	S
TU-210	create - Verificare che lanci un errore se l'input non è valido	S
TU-211	findAll - Verificare che ritorni un array di ristoranti	S
TU-212	findAllCuisines - Verificare che ritorni un array di cucine	S
TU-213	findAllCities - Verificare che ritorni un array di città	S
TU-214	findOne - Verificare che ritorni un ristorante	S
TU-215	getBookedTables - Verificare che ritorni un numero di tavoli prenotati in un ristorante in una data specifica	S
TU-216	getRestaurantAndMenuByRestaurantId - Verificare che ritorni un ristorante ed il suo menù	S
TU-217	getNumberOfFilteredRestaurants - Verificare che ritorni il numero di ristoranti filtrati	S
TU-218	createManager - Verificare che crei un manager del ristorante	S
TU-219	createManager - Verificare che lanci un errore se l'input non è valido	S
TU-220	StaffController - Verificare che il controller sia definito	S
TU-221	StaffController - Verificare che il servizio sia definito	S
TU-222	create - Verificare che crei un membro dello staff	S
TU-223	create - Verificare che lanci l' <i>eccezione</i> _G BadRequestException se la creazione del membro dello staff fallisce	S
TU-224	getRestaurantIdByAdminId - Verificare che ritorni l'admin dato un id ristorante	S
TU-225	getRestaurantIdByAdminId - Verificare che lanci l' <i>eccezione</i> _G NotFoundException se il ristorante non viene trovato	S
TU-226	StaffService - Verificare che il servizio sia definito	S
TU-227	StaffService - Verificare che crei un <i>repository</i> _G di staff	S
TU-228	create - Verificare che crei nuovo membro dello staff	S
TU-229	create - Verificare che ritorni nulla se l'elemento staffDto non è valido	S
TU-230	create - Verificare che ritorni null se il membro dello staff esiste già	S
TU-231	getAdminByRestaurantId - Verificare che ritorni un admin dato un id ristorante	S
TU-232	getRestaurantByAdminId - Verificare che ritorni il membro dello staff del ristorante dato un id utente admin	S
TU-233	create_manager - Verificare che crei un nuovo manager dello staff	S
TU-234	create_manager - Verificare che ritorni null se l'elemento staffDto non è valido	S
TU-235	create_manager - Verificare che ritorni null se il membro dello staff esiste già	S
TU-236	UserController - Verificare che il controller sia definito	S
TU-237	UserController - Verificare che il servizio sia definito	S

TU-238	create - Verificare che chiami il metodo create del servizio con i corretti parametri	S
TU-239	create - Verificare che ritorni il risultato del metodo create del servizio	S
TU-240	findOne - Verificare che chiami il metodo findOne del servizio con i corretti parametri	S
TU-241	findOne - Verificare che il risultato del metodo findOne del servizio	S
TU-242	createAdmin - Verificare che chiami il metodo create_admin del servizio con i corretti parametri	S
TU-243	createAdmin - Verificare che ritorni il risultato del metodo create_admin	S
TU-244	createAdmin - Verificare che lanci un errore se il metodo create_admin ritorna null	S
TU-245	create_user - Verificare che crei un nuovo utente	S
TU-246	create_user - Verificare che ritorni null se l'email è già registrata	S
TU-247	create_user - Verificare che ritorni null se qualche campo è mancante	S
TU-248	findOne - Verificare che ritorni un utente dato un id	S
TU-249	findUserByEmail - Verificare che ritorni un utente data una email	S
TU-250	create_admin - Verificare che crei un admin correttamente	S
TU-251	create_admin - Verificare che ritorni null se si incontra un errore durante la creazione di un admin	S
TU-252	create_user_manager - Verificare che crei un nuovo utente manager	S
TU-253	create_user_manager - Verificare che ritorni null se l'email è già registrata	S
TU-254	AppController - Verificare che il controller sia definito	S
TU-255	AppController - Verificare che il servizio sia definito	S
TU-281	DaysopenModule - Verificare che il modulo sia definito	S
TU-282	DaysopenModule - Verificare che il modulo abbia il DaysopenService	S
TU-283	DaysopenModule - Verificare che il modulo abbia il DaysopenController	S
TU-284	FoodModule - Verificare che il modulo sia definito	S
TU-285	findOne - Verificare che chiami la funzione foodService.findOne e ritorni il risultato corretto	S
TU-286	findOne - Verificare che lanci l'eccezione $Not\ Found\ Exception$ se il cibo non è stato trovato	S
TU-287	FoodService - Verificare che chiami la funzione foodRepository.findOne con un id dato	S
TU-288	MenuModule - Verificare che il modulo sia definito	S
TU-289	create - Verificare che chiami la funzione menuService.create e ritorni il risultato corretto	S
TU-290	create - Verificare che chiami la funzione menuRepository.create e la funzione menuRepository.save	S
TU-291	RestaurantModule - Verificare che il servizio RestaurantService sia definito	S
TU-292	RestaurantModule - Verificare che il controller RestaurantController sia definito	S
TU-293	StaffModule - Verificare che il servizio StaffService sia definito	S
TU-294	StaffModule - Verificare che il controller StaffController sia definito	S
TU-295	StaffModule - Verificare che StaffRepository sia definito	S

Tabella 15: Test di unità BackEnd

Tracciamento $test_G$ di unità BackEnd

Codice	Fonte
TU-75 TU-76 TU-77 TU-78 TU-79	autetication.controller.spec.ts
TU-80 TU-81 TU-82 TU-83 TU-84 TU-85	autetication.service.spec.ts
TU-86 TU-87	daysopen.controller.spec.ts
TU-88 TU-89 TU-90	daysopen.service.spec.ts
TU-91 TU-92	food.controller.spec.ts
TU-93 TU-94	food.service.spec.ts
TU-95 TU-96	menu.controller.spec.ts
TU-97 TU-98	menu.service.spec.ts
TU-99 TU-100 TU-101 TU-102	notification.controller.spec.ts
TU-103 TU-104 TU-105 TU-106 TU-107 TU-108	notification.service.spec.ts

TU-109	
TU-110	
TU-111	
TU-112	
TU-113	
TU-114	
TU-115	
TU-116	
TU-117	
TU-118	
TU-119	
TU-120	<code>orders.controller.spec.ts</code>
TU-121	
TU-122	
TU-123	
TU-124	
TU-125	
TU-126	
TU-127	
TU-128	
TU-129	
TU-130	
TU-131	
TU-132	
TU-133	
TU-134	
TU-135	
TU-136	
TU-137	
TU-138	<code>orders.service.spec.ts</code>
TU-139	
TU-140	
TU-141	
TU-142	
TU-143	
TU-144	
TU-145	

TU-146	
TU-147	
TU-148	
TU-149	
TU-150	
TU-151	
TU-152	
TU-153	
TU-154	
TU-155	
TU-156	
TU-153	
TU-158	<code>reservation.controller.spec.ts</code>
TU-159	
TU-160	
TU-161	
TU-162	
TU-163	
TU-164	
TU-165	
TU-166	
TU-167	
TU-168	
TU-169	
TU-170	

TU-171	<code>reservation.service.spec.ts</code>
TU-172	
TU-173	
TU-174	
TU-175	
TU-176	
TU-177	
TU-178	
TU-179	
TU-180	
TU-181	
TU-182	
TU-183	
TU-184	
TU-185	
TU-186	
TU-187	
TU-188	
TU-189	
TU-190	
TU-191	
TU-192	
TU-193	<code>restaurant.controller.spec.ts</code>
TU-194	
TU-195	
TU-196	
TU-197	
TU-198	
TU-199	
TU-200	
TU-201	
TU-202	
TU-203	
TU-204	

TU-205 TU-206 TU-207 TU-208 TU-209 TU-210 TU-211 TU-212 TU-213 TU-214 TU-215 TU-216 TU-217 TU-218 TU-219	<code>restaurant.service.spec.ts</code>
TU-220 TU-221 TU-222 TU-223 TU-224 TU-225	<code>staff.controller.spec.ts</code>
TU-226 TU-227 TU-228 TU-229 TU-230 TU-231 TU-232 TU-233 TU-234 TU-235	<code>staff.service.spec.ts</code>
TU-236 TU-237 TU-238 TU-239 TU-240 TU-241 TU-242 TU-243 TU-244	<code>user.controller.spec.ts</code>

TU-245 TU-246 TU-247 TU-248 TU-249 TU-250 TU-251 TU-252 TU-253	<code>user.service.spec.ts</code>
TU-254 TU-255	<code>app.controller.spec.ts</code>
TU-281 TU-282 TU-283	<code>daysopen.module.spec.ts</code>
TU-284 TU-285 TU-286 TU-287	<code>food.module.spec.ts</code>
TU-288 TU-289 TU-290	<code>menu.module.spec.ts</code>
TU-291 TU-292	<code>restaurant.module.spec.ts</code>
TU-293 TU-294 TU-295	<code>staff.module.spec.ts</code>

Tabella 16: Tracciamento dei test di unità BackEnd

3.1.3 Test di unità - WebSocket Server

Codice	Descrizione	Stato
TU-256	GatewayModule - Verificare che il modulo sia definito	S
TU-257	GatewayModule - Verificare che il modulo abbia il MyGateway provider	S
TU-258	handleConnection - Verificare che disconnetta il socket se il campo dati id_prenotazione è mancante	S
TU-259	handleConnection - Verificare che disconnetta il socket se il token è mancante	S
TU-260	handleConnection - Verificare che disconnetta il socket se il token è non valido	S
TU-261	handleConnection - Verificare che il socket faccia un join alla stanza se id_prenotazione è fornito e il token è valido	S
TU-262	handleConnection - Verificare che venga mostrato il messaggio "onMessage" alla stanza specificata	S
TU-263	handleConnection - Verificare che venga mostrato il messaggio "onIngredient" alla stanza specificata	S
TU-264	handleConnection - Verificare che venga mostrato il messaggio "onConfirm" alla stanza specificata	S
TU-265	NotificationController - Verificare che il controller sia definito	S
TU-266	NotificationController - Verificare che chiami la funzione emitAll con il corpo della notifica	S
TU-267	NotificationModule - Verificare che il modulo sia definito	S
TU-268	NotificationModule - Verificare che il modulo abbia il NotificationController	S
TU-269	NotificationModule - Verificare che il modulo abbia il NotificationGateway	S
TU-270	NotificationGateway - Verificare che mandi notifiche a tutti coloro che devono riceverle	S
TU-271	handleConnection - Verificare che si disconnetta se non è fornito alcun token	S
TU-272	handleConnection - Verificare che si disconnetta se il token non è valido	S
TU-273	handleConnection - Verificare che avvenga la join del socket ad un stanza se il token è valido	S
TU-274	handleConnection - Verificare che avvenga il log quando il socket si disconnette	S
TU-275	AppController - Verificare che il controller sia definito	S
TU-276	AppModule - Verificare che il modulo sia definito	S
TU-277	AppModule - Verificare che il modulo abbia l'AppController	S
TU-278	AppModule - Verificare che il modulo abbia l'AppService	S
TU-279	AppModule - Verificare che il modulo abbia il GatewayModule	S
TU-280	AppModule - Verificare che il modulo abbia il NotificationModule	S

Tabella 17: Test di unità WebSocket Server

Tracciamento $test_G$ di unità $WebSocket_G$ Server

Codice	Fonte
TU-256	gateway.module.spec.ts
TU-257	

TU-258 TU-259 TU-260 TU-261 TU-262 TU-263 TU-264	gateway.spec.ts
TU-265 TU-266	notification.controller.spec.ts
TU-267 TU-268 TU-269	notification.module.spec.ts
TU-270 TU-271 TU-272 TU-273 TU-274	notificationGateway.spec.ts
TU-275	app.controller.spec.ts
TU-276 TU-277 TU-278 TU-279 TU-280	app.module.spec.ts

Tabella 18: Tracciamento dei test di unità WebSocket

3.2 Test di sistema

I $test_G$ di $sistema_G$ sono una fase del $processo_G$ di $test_G$ ing il cui scopo è quello di verificare che il $sistema_G$ $software_G$ rispetti i requisiti specificati nel documento *Analisi Dei Requisiti*. Di seguito verranno elencati i vari $test_G$, i quali avranno un codice identificativo, una descrizione, il $requisito_G$ a cui fa riferimento e lo stato del $test_G$.

Codice	Descrizione	Requisito	Stato
TS-01	Verificare che l'utente base possa visualizzare i pasti ordinabili	ROF 1	NI
TS-02	Verificare che il $sistema_G$ possa inviare una notifica se l'utente base sta aggiungendo un piatto che contiene elementi a cui è allergico/intollerante	RDF 2	NI
TS-03	Verificare che l'utente possa visualizzare i pasti con i loro ingredienti e che possa modificarli togliendo ingredienti	ROF 3	NI
TS-04	Verificare che l'utente base possa essere in grado di visualizzare il riepilogo di quanto ordinato e di possa confermarlo.	ROF 4	NI
TS-05	Il $sistema_G$ deve poter inviare una notifica di conferma dell'ordinazione collaborativa all'amministratore del ristorante	ROF 5	NI
TS-06	Verificare che l'utente possa inserire le informazioni (data, orario, persone) per effettuare una $prenotazione_G$	ROF 6	NI

TS-07	Il <i>sistema_G</i> deve poter inviare una notifica all'amministratore per comunicare la richiesta di <i>prenotazione_G</i> , che può accettare e rifiutare.	ROF 7	NI
TS-08	Verificare che l'utente possa essere in grado di cancellare la <i>prenotazione_G</i>	RDF 8	NI
TS-09	Verificare che il <i>sistema_G</i> possa negare la <i>prenotazione_G</i> se il ristorante non possiede abbastanza posti o tavoli	ROF 9	NI
TS-10	Verificare che l'utente base possa visualizzare una lista di ristoranti filtrata per nome, data, luogo o tipologia cucina	ROF 10	NI
TS-11	Verificare che l'utente base possa selezionare un ristorante	ROF 11	NI
TS-12	Verificare che l'amministratore possa visualizzare una lista di prenotazioni in attesa	ROF 12	NI
TS-13	Verificare che l'amministratore possa selezionare una specifica richiesta di <i>prenotazione_G</i> dalla lista visualizzata	ROF 13	NI
TS-14	Verificare che l'amministratore possa accettare la richiesta di <i>prenotazione_G</i> selezionata	ROF 14	NI
TS-15	Verificare che il <i>sistema_G</i> possa aggiungere la <i>prenotazione_G</i> nell'area "prenotazioni" dell'utente	ROF 15	NI
TS-16	Verificare che il <i>sistema_G</i> possa notificare gli utenti coinvolti (nel caso di <i>prenotazione_G</i> collaborativa) dell'accettazione della <i>prenotazione_G</i>	ROF 16	NI
TS-17	Verificare che il <i>sistema_G</i> possa fornire un'interfaccia per consentire all'amministratore di verificare la disponibilità di posti in base alle specifiche della <i>prenotazione_G</i> selezionata	RDF 17	NI
TS-18	Verificare che il <i>sistema_G</i> possa ridurre il numero di posti disponibili in base alle specifiche della <i>prenotazione_G</i> accettata	RDF 18	NI
TS-19	Verificare che l'amministratore possa rifiutare la richiesta di <i>prenotazione_G</i> selezionata	ROF 19	NI
TS-20	Verificare che il <i>sistema_G</i> possa notificare gli utenti coinvolti (nel caso di <i>prenotazione_G</i> collaborativa) del rifiuto della <i>prenotazione_G</i>	ROF 20	NI
TS-21	Verificare che il <i>sistema_G</i> possa creare un canale di comunicazione tra l'utente e l'amministratore del ristorante quando l'utente lo richiede	RDF 21	NI
TS-22	Verificare che l'utente e l'amministratore possano scambiarsi messaggi in modo bidirezionale tramite l'interfaccia di comunicazione	RDF 22	NI
TS-23	Verificare che durante la comunicazione, il <i>sistema_G</i> possa inviare notifiche <i>push_G</i> per informare l'utente e l'amministratore dei nuovi messaggi ricevuti	RDF 23	NI
TS-24	Verificare che la cancellazione della <i>prenotazione_G</i> possa essere effettuata con al massimo un giorno di anticipo rispetto alla data della <i>prenotazione_G</i>	ROF 24	NI
TS-25	Verificare che il <i>sistema_G</i> permetta di pagare il conto in base alla modalità scelta (divisione equa, divisione proporzionale) da chi ha creato la <i>ordinazione_G</i> collaborativa	ROF 25	NI

TS-26	Verificare che l'utente base possa pagare tutto il conto se nessun utente ha pagato	RDF 26	NI
TS-27	Verificare che l'amministratore possa modificare il menù del proprio ristorante, aggiungendo, rimuovendo pietanze e modificando le informazioni del singolo piatto (nome, ingredienti e prezzo)	RDF 27	NI
TS-28	Verificare che la modifica del menù da parte dell'amministratore non causi problemi di sincronizzazione nella visualizzazione, ricerca del menù e nell' <i>ordinazione_G</i> da parte dell'utente base	RDF 28	NI
TS-29	Verificare che l'utente base possa inserire un coupon prima di pagare il conto, che, se applicato, deve far ricalcolare al <i>sistema_G</i> il prezzo del conto	RDF 29	NI
TS-30	Verificare che l'amministratore possa consultare le prenotazioni associate al proprio ristorante	ROF 30	NI
TS-31	Verificare che l'amministratore possa visualizzare i dettagli completi di una specifica <i>prenotazione_G</i>	ROF 31	NI
TS-32	Verificare che l'amministratore possa visualizzare lo stato delle ordinazioni associate alla <i>prenotazione_G</i>	ROF 32	NI
TS-33	Verificare che il <i>sistema_G</i> possa fornire all'amministratore la possibilità di visualizzare la lista totale degli ingredienti inclusi nella <i>prenotazione_G</i>	ROF 33	NI
TS-34	Verificare che il <i>sistema_G</i> possa consentire all'amministratore di visualizzare tutti gli ordini confermati per il ristorante selezionato	ROF 34	NI
TS-35	Verificare che il <i>sistema_G</i> possa permettere di annullare l' <i>ordinazione_G</i> collaborativa nel tempo utile per farlo	RDF 35	NI
TS-36	Verificare che il <i>sistema_G</i> invii una notifica a tutti gli utenti associati alla <i>prenotazione_G</i> la cui <i>ordinazione_G</i> collaborativa è stata annullata	RDF 36	NI
TS-37	Verificare che il <i>sistema_G</i> fornisca un'interfaccia per consentire agli utenti non autenticati di registrarsi come utenti base	ROF 37	NI
TS-38	Verificare che l'utente base e l'amministratore possano inserire le proprie informazioni personali durante la registrazione come nome, cognome, email, password	ROF 38	NI
TS-39	Verificare che l'utente base e l'amministratore possano confermare di volersi registrare con le informazioni fornite prima di completare la registrazione	ROF 39	NI
TS-40	Verificare che il <i>sistema_G</i> gestisca correttamente gli errori nell'inserimento delle informazioni durante la registrazione	ROF 40	NI
TS-41	Verificare che l'utente base e l'amministratore possano visualizzare il menu' del ristorante selezionato	ROF 41	NI
TS-42	Verificare che l'amministratore possa modificare le informazioni del proprio ristorante (nome, indirizzo, orari, coperti e tipologia di cucina)	RDF 42	NI
TS-43	Verificare che la modifica delle informazioni del ristorante da parte dell'amministratore non causi problemi di sincronizzazione nella visualizzazione, ricerca del ristorante e nell' <i>ordinazione_G</i> da parte dell'utente base	RDF 43	NI

TS-44	Verificare che l'utente autenticato possa visualizzare una lista con le prenotazioni passate	RDF 44	NI
TS-45	Verificare che l'utente autenticato possa rilasciare una recensione (con anche una votazione di gradimento tramite stelle) sui ristoranti nei quali ha effettuato almeno una <i>prenotazione_G</i>	RDF 45	NI
TS-46	Verificare che l'utente autenticato possa visualizzare gli ordini di un tavolo	ROF 46	NI
TS-47	Verificare che l'utente autenticato possa visualizzare le recensioni rilasciate ed eventualmente eliminarle	RDF 47	NI
TS-48	Verificare che l'utente generico possa visualizzare le recensioni di un ristorante e visualizzare per ognuna di essa le relative informazioni	RDF 48	NI
TS-49	Verificare che l'amministratore possa inserire le informazioni del ristorante di cui è amministratore	ROF 49	NI
TS-50	Verificare che l'utente/l'amministratore possa inserire la propria email e la password durante la fase di login	ROF 50	NI
TS-51	Verificare che il <i>sistema_G</i> verifichi che le informazioni inserite nel login corrispondano ad un account esistente nel <i>sistema_G</i>	ROF 51	NI
TS-52	Il <i>sistema_G</i> deve predisporre un'opzione per il recupero della password	RDF 52	NI
TS-53	L'utente non autenticato deve poter inserire la propria email durante il <i>processo_G</i> di recupero password	RDF 53	NI
TS-54	Se l'email inserita dall'utente corrisponde ad un account nel <i>sistema_G</i> , il <i>sistema_G</i> deve inviare un'email contenente un link per il recupero della password	RDF 54	NI
TS-55	Verificare che, se l'email inserita dall'utente non corrisponde a un account nel <i>sistema_G</i> , il <i>sistema_G</i> lo mostri a schermo	ROF 55	NI
TS-56	L'utente deve poter accedere a una sezione dedicata per il recupero della password tramite il link fornito nell'email di recupero password	RDF 56	NI
TS-57	Verificare che il <i>sistema_G</i> mostri una lista di prenotazioni, ognuna con le seguenti informazioni: Nome del ristorante, Data, Ora, Stato della <i>prenotazione_G</i> (se è ancora attiva o già completata) e Numero di persone coinvolte	ROF 57	NI
TS-58	Verificare che il <i>sistema_G</i> ordini la lista delle prenotazioni per data della <i>prenotazione_G</i>	RDF 58	NI
TS-59	Verificare che l'utente possa inserire le proprie allergie/intolleranze, se presenti, durante la modifica del profilo	RDF 59	NI
TS-60	L'utente autenticato e l'amministratore devono poter inserire nome, cognome, email e password nell'area di modifica delle proprie informazioni.	RDF 60	NI
TS-61	Verificare che il <i>sistema_G</i> verifichi che l'email inserita dall'utente base/amministratore sia valida e non sia già presente nel <i>sistema_G</i>	ROF 61	NI
TS-62	Verificare che l'utente base autenticato possa accedere alla funzionalità di <i>ordinazione_G</i> di un piatto	ROF 62	NI

TS-63	Verificare che il <i>sistema_G</i> consenta all'utente di visualizzare e togliere ingredienti del piatto selezionato	ROF 63	NI
TS-64	Verificare che l'amministratore possa visualizzare le recensioni del proprio ristorante	RDF 64	NI
TS-65	Verificare che l'amministratore possa rispondere alle recensioni del proprio ristorante	RDF 65	NI
TS-66	Verificare che l'utente possa visualizzare le risposte alla propria recensione	RDF 66	NI
TS-67	Verificare che il <i>sistema_G</i> consenta di modificare l'ordine ad un utente prima della scadenza del tempo previsto, inserendo piatti, modificando ingredienti e quantità delle pietanze ordinate	ROF 67	NI
TS-68	Verificare che il <i>sistema_G</i> permetta di far visualizzare all'amministratore il dettaglio degli ingredienti necessari per ogni giornata	RDF 68	NI
TS-69	Verificare che il <i>sistema_G</i> consenta all'utente autenticato di selezionare l'opzione di logout	ROF 69	NI
TS-70	Verificare che, dopo che l'utente ha selezionato l'opzione di logout, il <i>sistema_G</i> richieda una conferma esplicita dall'utente prima di procedere con la disconnessione	RDF 70	NI
TS-71	Verificare che, dopo aver terminato la sessione dell'utente, il <i>sistema_G</i> reindirizzi l'utente alla pagina di accesso o a una pagina di destinazione predefinita	ROF 71	NI
TS-72	Verificare che l'utente autenticato possa visualizzare le informazioni del suo profilo	RDF 72	NI
TS-73	Verificare che l'amministratore possa visualizzare tutte le chat aperte in precedenza da altri utenti	RDF 73	NI
TS-74	Verificare che l'amministratore possa rispondere alle chat con gli utenti	RDF 74	NI
TS-75	Verificare che il <i>sistema_G</i> generi un link valido per invitare altri utente alla <i>prenotazione_G</i>	ROF 75	NI
TS-76	Il <i>sistema_G</i> permette di far visualizzare all'amministratore in dettaglio la lista delle prenotazioni di ogni giornata.	RDF 76	NI
TS-77	Verificare che il <i>sistema_G</i> invii una notifica all'amministratore in base agli aggiornamenti sul conto di una <i>prenotazione_G</i> del suo ristorante.	ROF 77	NI

Tabella 19: Test di sistema

3.3 Test di accettazione

I *test_G* di accettazione servono per garantire che il prodotto soddisfi i requisiti utente come specificati nel *capitolato_G*. Essi vengono eseguiti in presenza del committente e dimostrano la conformità del prodotto alle aspettative attraverso l'esecuzione dei casi di prova previsti nel *capitolato_G*.

Codice	Descrizione	Stato
TA-01	Verificare che l'utente si possa registrare come utente base o amministratore	S
TA-02	Inserimento nome, cognome, mail e password	S
TA-03	Inserimento nome ristorante, città e recapiti in caso di registrazione dell'amministratore	S
TA-04	Inserimento orari di apertura in caso di registrazione dell'amministratore	S
TA-05	Inserimento coperti in caso di registrazione dell'amministratore	S
TA-06	Inserimento tipologia di cucina in caso di registrazione dell'amministratore	S
TA-07	Richiesta login per l'utente base in caso di prenotazione	S
TA-08	Ricerca dei ristoranti per nome e città	S
TA-09	Ricerca dei ristoranti per tipologia di cucina	S
TA-10	Possibilità di creare una <i>prenotazione_G</i> per l'utente base	S
TA-11	Notificare l'amministratore della nuova <i>prenotazione_G</i>	S
TA-12	Notificare l'utente base della conferma o del rifiuto della <i>prenotazione_G</i>	S
TA-13	Visualizzazione delle prenotazioni per l'utente base	S
TA-14	Richiesta login utente base per l' <i>ordinazione_G</i>	S
TA-15	Visualizzazione della lista dei piatti e degli ingredienti per l' <i>ordinazione_G</i>	S
TA-16	Modifica della quantità del piatto	S
TA-17	Rimozione degli ingredienti del piatto	S
TA-18	Riepilogo dell'ordine	S
TA-19	Notifica all'amministratore per la conferma delle ordinazioni	S
TA-20	Richiesta del login all'utente base in fase di pagamento	S
TA-21	Scelta della modalità di divisione del conto tra equa e proporzionale	S
TA-22	Il primo utente seleziona la modalità di pagamento	S
TA-23	Pagamento in parti uguali	S
TA-24	Pagamento in parti proporzionali	S
TA-25	Notifiche dell'aggiornamento dello stato di pagamento verso l'amministratore	S
TA-26	Richiesta login per l'utente amministratore nella fase di consultazione delle prenotazioni del ristorante	S

TA-27	Vista delle prenotazioni effettuate nella fase di consultazione delle prenotazioni del ristorante	S
TA-28	Visualizzazione dello stato nella fase di consultazione delle prenotazioni del ristorante	S
TA-29	Visualizzazione della lista degli ingredienti per ogni <i>prenotazione_G</i> nella fase di consultazione delle prenotazioni del ristorante	S
TA-30	Possibilità di comunicare con lo staff del ristorante	NI
TA-31	Possibilità di rilasciare recensioni e <i>feedback_G</i> al ristorante	NI
TA-32	Inserimento allergie	NI
TA-33	Notifiche <i>push_G</i>	NI
TA-34	Coverage dei <i>test_G</i> minimo di 80%	S
TA-35	Sito accessibile da mobile	S

4 Checklist

La verifica tramite analisi statica è preferibile che avvenga tramite *Inspection_G* anziché *Walkthrough_G*, per questo motivo sono definite delle liste di controllo aggiornate progressivamente dai verificatori che permettono rapidamente di analizzare gli errori più comuni e di verificare selettivamente.

4.1 Documentazione

4.1.1 Struttura

Errore	Descrizione
Manca la <i>caption_G</i> per le tabelle o le immagini	Ogni tabella o immagine deve possedere una <i>caption_G</i>
Sezione vuota	Le sezioni vuote devono essere eliminate
Parola non coincide con il glossario	Tale parola deve essere riportata con le stesse lettere maiuscole e minuscole rispetto al glossario eccetto la lettera iniziale

Tabella 21: Checklist dei possibili errori nella struttura della documentazione

4.1.2 Errori ortografici

Errore	Descrizione
Errore di sintassi	Errori di battitura o di distrazione devono essere rimossi
Errore di coniugazione	Gli errori di coniugazione devono essere rimossi

Tabella 22: Checklist dei possibili errori ortografici nella documentazione

4.1.3 Analisi Dei Requisiti

Errore	Descrizione
Requisiti per un <i>caso d'uso_G</i> assenti	Ad ogni <i>caso d'uso_G</i> deve corrispondere almeno un requisito
Diagrammi dei casi d'uso erronei	I diagrammi dei casi d'uso devono riportare la corretta numerazione e nomenclatura del <i>caso d'uso_G</i> stesso e riferirsi allo stesso modo ad altri casi d'uso

Tabella 23: Checklist dei possibili errori per l'Analisi Dei Requisiti

5 Cruscotto di valutazione della qualità

5.1 Qualità di processo

5.1.1 MPC01-EAC (Estimated at Completion)

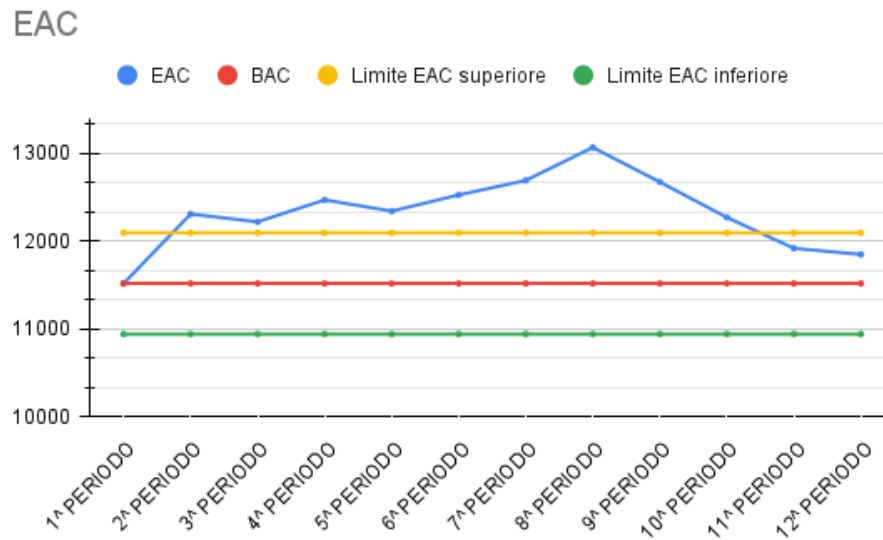


Figura 1: Estimated at Completion

RTB: Dal grafico si può notare che l'EAC supera il valore accettabile, questo a causa di una scorretta divisione tra ore produttive e individuali, oltre che a lunghi periodi di tempo impiegati nello studio delle *tecnologie_G* necessarie per lo svolgimento del progetto. Si può notare un andamento finale decrescente, quindi si ritiene che dopo la fase iniziale, il gruppo possa rientrare nei valori ottimali. Infine il team è consapevole che l'andamento iniziale del valore è dovuto ad una gestione errata del progetto e quindi i membri dovranno impegnarsi per rientrare nel valore ottimale e garantire il raggiungimento degli obiettivi fissati.

PB: Dal grafico si può notare che, durante la fase *PB_G* (decimo periodo in poi), il valore dell'EAC è sceso, rientrando nei valori accettabili. Ciò è dovuto ad una migliore divisione e assegnazione delle tasks e una valutazione del tempo di lavoro più precisa.

5.1.2 MPC02-PV (Planned Value) e MPC04-EV (Earned Value)

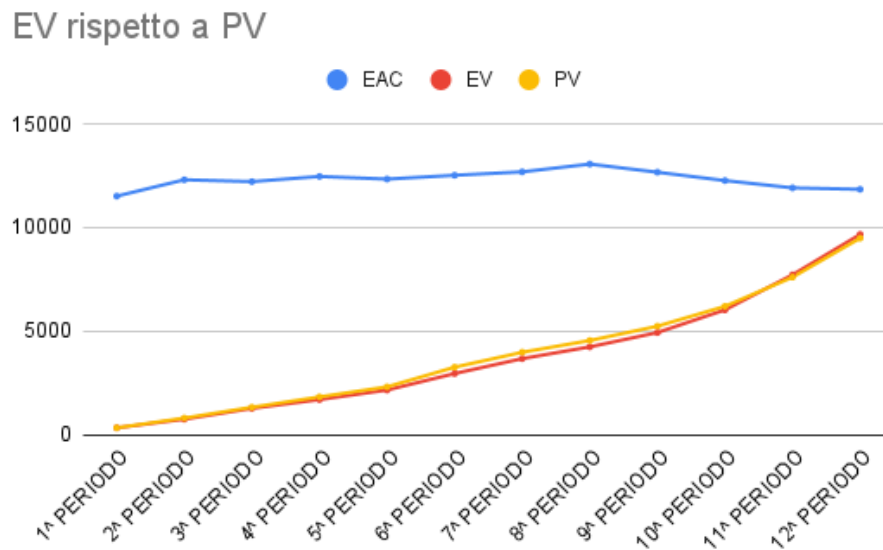


Figura 2: Planned Value - Earned Value

RTB: Dal grafico si può notare che le linee del PV e dell'EV sono molto vicine anche se si può notare un allontanamento dei due valori negli ultimi periodi. Questo denota che il costo preventivato è maggiore del costo sostenuto dal gruppo e quindi il team dovrà impegnarsi maggiormente per terminare le attività stabilite.

PB: Si può notare che il valore PV e EV sono molto vicini tra di loro e quindi il costo effettivo rispetta quanto preventivato.

5.1.3 MPC05-ETC (Estimated to Complete)

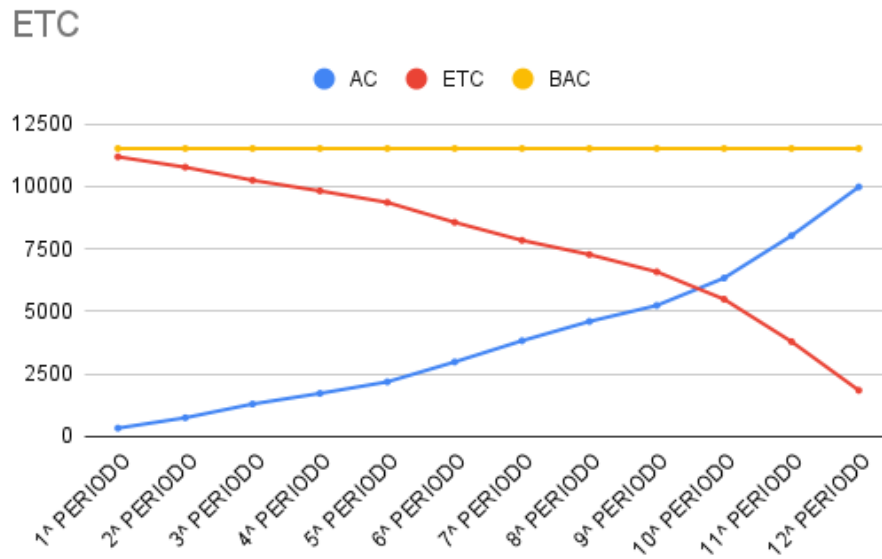


Figura 3: Estimated to Complete

RTB: Dal grafico si può notare che le linee dell'AC e dell'ETC nel corso dei vari periodi, mantengono un andamento costante. Di conseguenza si può affermare che il progetto stia mantenendo un ritmo regolare di avanzamento.

PB: Dal nono periodo si può notare un incremento dell'andamento del gruppo rispetto al periodo RTB_G .

5.1.4 MPC06-CV (Cost Variance), MPC07-SV (Schedule Variance) e MPC08-BV (Budget Variance)

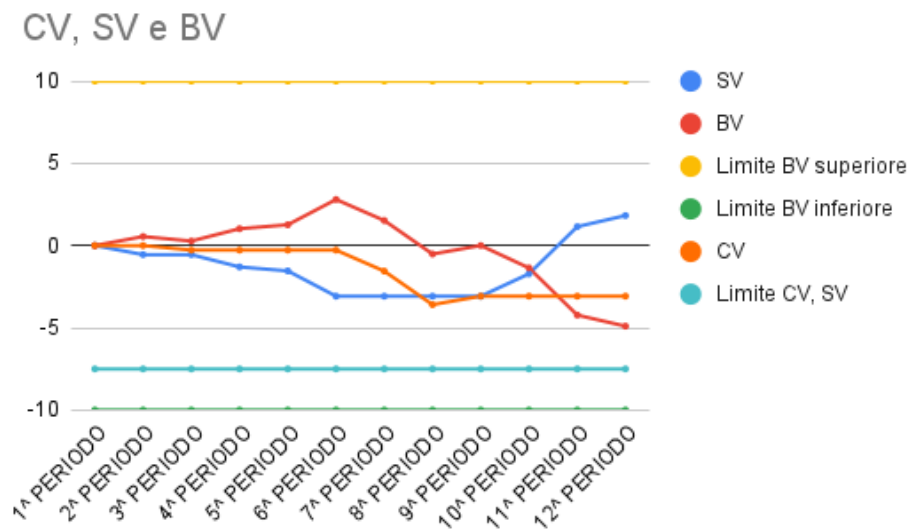


Figura 4: Cost Variance - Schedule Variance - Budget Variance

RTB: Dal grafico si può notare che la linea della Cost Variance ha mantenuto un andamento costante e lineare fino al sesto periodo, cosa positiva. Dal settimo periodo periodo però c'è stato un calo, ciò si può ricondurre alla quasi assente divisione delle ore produttive da quelle individuali. Osservando la Schedule Variance si nota invece che l'allontanamento dal valore preferibile è iniziato molto prima. Questa situazione è stata causata da un misto di mancata divisione delle ore produttive e ore individuali, e di sottostime delle ore da dare per ogni lavoro. Infine, osservando la Budget Variance, si nota che, nonostante si sia speso più di quanto previsto per la maggior parte del tempo, ciò si è stabilizzato negli ultimi periodi.

PB: Durante la PB_G (decimo periodo in poi) si nota un abbassamento della BV, con un aumento della SV; questo è dato dal fatto che durante questa fase il gruppo ha saputo gestire meglio le task. Nonostante ciò, i valori rientrano all'interno dei limiti imposti dal gruppo stesso. Per quanto riguarda la Cost Variance, si nota un calo e una stabilizzazione verso la fine; questo a causa di molte task implementative che hanno richiesto un tempo di completamento maggiore e quindi un costo maggiore rispetto a quanto preventivato inizialmente.

5.1.5 MPC09-RSI (Requirements stability index)

Per quanto riguarda l'attinenza dell'inquadramento dei requisiti funzionali del progetto, un incontro avuto nel corso del 10° periodo con l'azienda proponente ha portato a una importante revisione di quelli che erano stati individuati come "requisiti obbligatori" nel corso della fase RTB_G . A causa di ritardi nello sviluppo, infatti, e al fine di non sfiorare con la data di consegna prefissata, il gruppo ha concordato insieme al proponente di definire alcuni requisiti, precedentemente individuati come "obbligatori", "desiderabili". Nonostante dunque l'elevato numero di requisiti che hanno richiesto una revisione nel corso della PB_G , il *Requirements stability index* risulta comunque superiore al valore accettabile.

Metrica	Nome	Valore accettabile	Valore ottimale	Valore ottenuto
MPC09-RSI	<i>Requirements stability index</i>	70%	100%	72%

5.1.6 MPC15-PCTS (Percentuale dei test superati)

Per quanto riguarda la percentuale dei $test_G$ superati, dal momento della loro introduzione nel corso del 10° periodo, il loro superamento è sempre stato verificato prima dell'effettivo *merge* all'interno del *branch_G develop*. A parte dunque per un primo periodo in cui non era ancora completata appieno l'automazione di $test_G coverage$ all'interno del *repository*, si è sempre ottenuto una percentuale di $test_G$ superati pari al 100%, valore che corrisponde a quello ottenuto con la build definitiva.

Metrica	Nome	Valore accettabile	Valore ottimale	Valore ottenuto
MPC15-PCTS	Percentuale dei $test_G$ superati	80%	100%	100%

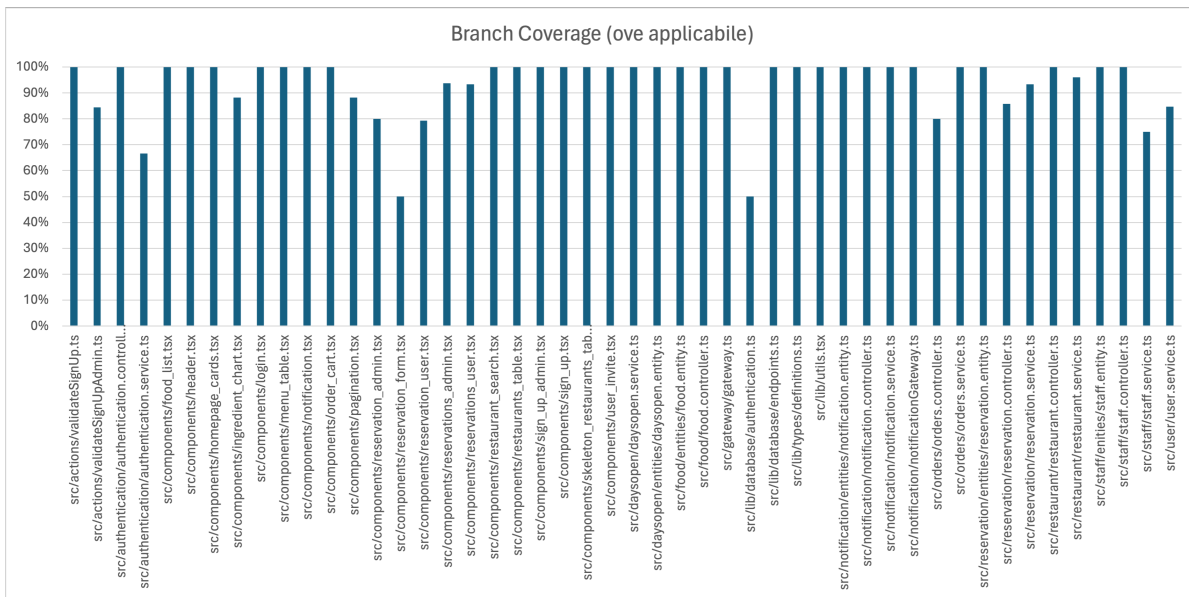


Figura 6: Branch Coverage

5.1.9 MPC19-NCR (Non Calculated Risk)

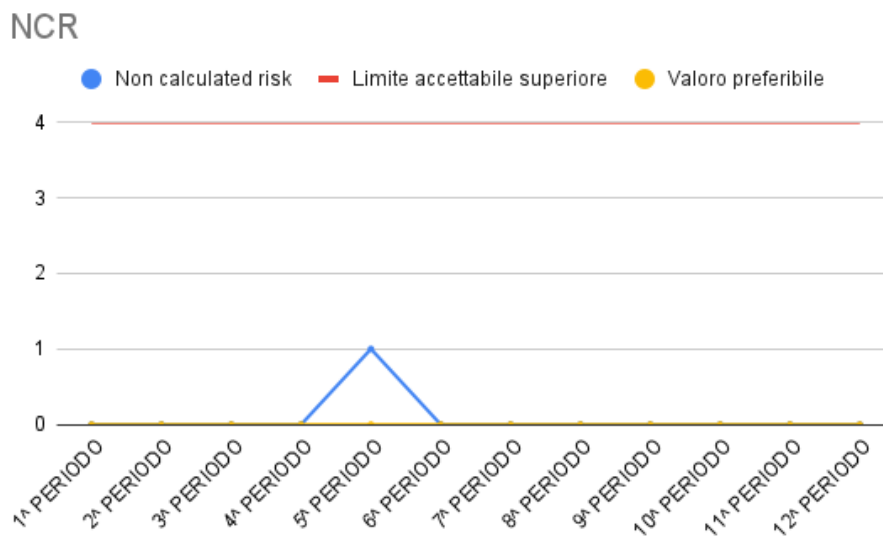


Figura 7: Non Calculated Risk

Dal grafico si può notare che per la maggior parte del tempo non sono comparsi *rischi_G* non previsti. Solo nel quinto periodo è emerso un rischio di cui non si era tenuto conto inizialmente, ovvero la sessione di esami, la quale ha portato via parecchio tempo ai vari membri del gruppo, rallentando di molto l'avanzamento del lavoro.

5.2 Qualità di prodotto

5.2.1 MPD01-CRO (Copertura dei requisiti obbligatori)

Il numero di requisiti obbligatori è stato ridotto a seguito di un colloquio avuto con il proponente nel 10° periodo. La necessità di ridurre il numero di requisiti obbligatori è stata dettata dal ritardo accumulato dal gruppo nel corso dello sviluppo del *software*. Al termine dell'11° periodo sono stati soddisfatti tutti i requisiti obbligatori individuati.

Metrica	Nome	Valore accettabile	Valore ottimale	Valore ottenuto
MPD01-CRO	Copertura dei requisiti obbligatori	100%	100%	100%

5.2.2 MPC07-FD (Failure Density)

A seguito del termine dello sviluppo *software*, per quanto riguarda il completamento dei *test_G*, il gruppo ha ottenuto una percentuale di successo di completamento dei *test_G* pari al 100%.

Nel corso dei vari *merge* nel *branch_G develop*, calcolati a partire dal 10° periodo (periodo durante il quale sono stati inizialmente inseriti dei *test_G*), solo in una prima fase in cui non era completata l'automazione di *test_G coverage* si registra un elevato numero di *test_G* falliti, arrivando comunque a un totale di solo 20 su 54 *merge* che abbiano dato come esito "fail". Ciò è verificabile andando a controllare la sezione "Code Coverage" della *repository_G* GitHub.

Metrica	Nome	Valore accettabile	Valore ottimale	Valore ottenuto
MPC07-FD	Failure Density	10%	0%	0%