



TheFellowshipOfTheCode

Piano di Qualifica

Informazioni sul documento

Nome Documento	PianoDiQualifica_v4_0_0.pdf
Versione	4
Data di Creazione	12 Dicembre 2015
Data ultima modifica	9 Aprile 2016
Stato	Approvato
Redazione	Marco Prelaz Mattia Varotto Franco Berton Alberto Ferrara Matteo Gnoato Matteo Granzotto
Verifica	Marco Prelaz Matteo Granzotto Mattia Varotto
Approvazione	Alberto Ferrara
Uso	Esterno
Distribuzione	TheFellowshipOfTheCode
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, Zucchetti S.P.A.
Email di riferimento	thefellowshipofthecode@gmail.com

Sommario

Documento contenente le strategie adottate dal gruppo TheFellowshipOfTheCode per raggiungere gli obiettivi qualitativi richiesti per il prodotto QuizziPedia.



Registro delle modifiche

Versione	Descrizione	Autore e Ruolo	Data
4.0.0	Approvazione del documento	Alberto Ferrara Responsabile	2016-04-09
3.1.0	Verifica del documento	Matteo Granzotto Verificatore	2016-04-09
3.0.2	Aggiornata l'appendice A	Marco Prelaz Progettista	2016-04-08
3.0.1	Aggiunti Test di Unità	Marco Prelaz Progettista	2016-04-07
3.0.0	Approvazione del documento	Simone Magagna Responsabile	2016-03-19
2.1.0	Verifica del documento	Alberto Ferrara Verificatore	2016-03-19
2.0.5	Aggiunti Test di Sistema	Mattia Varotto Progettista	2016-03-18
2.0.4	Aggiunti Test di Integrazione	Franco Berton Progettista	2016-03-18
2.0.3	Aggiunti Test di Validazione	Franco Berton Progettista	2016-03-17
2.0.2	Creazione sezione Tracciamento dei test	Marco Prelaz Progettista	2016-03-16
2.0.1	Stesura sezione Specifica dei test	Mattia Varotto Progettista	2016-03-16
2.0.0	Approvazione del documento	Matteo Granzotto Responsabile	2016-02-21
1.1.0	Verifica del documento	Simone Magagna Verificatore	2016-02-21
1.0.5	Rimozione sezione Gestione amministrativa della revisione	Mattia Varotto Analista	2016-02-19
1.0.4	Rimozione appendici PDCA, SPICE e Standard ISO/IEC 9126	Marco Prelaz Analista	2016-02-19
1.0.3	Stesura sezione Qualità di prodotto	Mattia Varotto Analista	2016-02-19
1.0.2	Stesura sezione Qualità di processo	Marco Prelaz Analista	2016-02-19
1.0.1	Sostituzione della sezione Strategie di Verifica con le sezioni Qualità di processo e Qualità di prodotto	Mattia Varotto Analista	2016-02-18
1.0.0	Approvazione del documento	Matteo Granzotto Responsabile	2016-01-20



Versione	Descrizione	Autore e Ruolo	Data
0.3.0	Verifica del documento	Simone Magagna Verificatore	2016-01-19
0.2.1	Aggiunti valori delle verifiche automatizzate	Marco Prelaz Verificatore	2016-01-19
0.2.0	Verifica del documento	Simone Magagna Verificatore	2015-12-29
0.1.1	Modifica paragrafi Scopo del prodotto e Glossario	Mattia Varotto Analista	2015-12-28
0.1.0	Verifica del documento	Simone Magagna Verificatore	2015-12-28
0.0.4	Stesura sezione Gestione amministrativa della revisione e appendici A,B,C,D	Marco Prelaz Analista	2015-12-14
0.0.3	Stesura sezione Strategie di verifica	Mattia Varotto Analista	2015-12-13
0.0.2	Stesura sezione Introduzione	Mattia Varotto Analista	2015-12-12
0.0.1	Creato template	Matteo Granzotto Responsabile	2015-12-12



Indice

1	Introduzione	2
1.1	Scopo del Documento	2
1.2	Scopo del prodotto	2
1.3	Glossario	2
1.4	Riferimenti	2
1.4.1	Normativi	2
1.4.2	Informativi	2
2	Qualità di processo	3
2.1	Infrastructure Management Process (6.2.2)	4
2.1.1	Obiettivi di qualità	4
2.1.2	Strategie	4
2.1.3	Metriche	5
2.1.3.1	Disponibilità DocumentsDB	5
2.1.3.2	Tempo di correzione incoerenze in DocumentsDB	5
2.2	Project Planning, Assessment & Control Process (6.3.1 - 6.3.2)	5
2.2.1	Obiettivi di qualità	5
2.2.2	Strategie	5
2.2.3	Metriche	6
2.2.3.1	Schedule Variance	6
2.2.3.2	Budget Variance	6
2.3	Risk Management Process (6.3.4)	6
2.3.1	Obiettivi di qualità	6
2.3.2	Strategie	6
2.3.3	Metriche	6
2.3.3.1	Rischi non preventivati	6
2.3.3.2	Efficienza di gestione dei rischi	7
2.4	System/Software Requirements Analysis Process (6.4.2 - 7.1.2)	7
2.4.1	Obiettivi di qualità	7
2.4.2	Strategie	7
2.4.3	Metriche	7
2.4.3.1	Requisiti obbligatori soddisfatti	7
2.5	System/Software Architectural Design Process (6.4.3 - 7.1.3)	8
2.5.1	Obiettivi di qualità	8
2.5.2	Strategie	8
2.5.3	Metriche	8
2.5.3.1	Structural Fan-In	8
2.5.3.2	Structural Fan-Out	8
2.6	Software Detailed Design Process (7.1.4)	8
2.6.1	Obiettivi di qualità	9
2.6.2	Strategie	9
2.6.3	Metriche	9
2.6.3.1	Numero di metodi per classe	9
2.6.3.2	Numero di parametri per metodo	9
2.7	Software Construction Process (7.1.5)	9
2.7.1	Obiettivi di qualità	9
2.7.2	Strategie	10
2.7.3	Metriche	10
2.7.3.1	Produttività di codifica	10



2.7.3.2	Complessità Ciclomatica	10
2.7.3.3	Numero di livelli di annidamento	10
2.7.3.4	Linee di codice per linee di commento	11
2.7.3.5	Variabili inutilizzate	11
2.7.3.6	Dipendenze	11
2.7.3.7	Halstead Difficulty per-function	11
2.7.3.8	Halstead Volume per-function	11
2.7.3.9	Halstead Effort per-function	12
2.7.3.10	Indice di manutenibilità	12
2.8	System/Software Integration Process (6.4.5 - 7.1.6)	12
2.8.1	Obiettivi di qualità	12
2.8.2	Strategie	12
2.8.3	Metriche	12
2.8.3.1	Componenti integrate	12
2.9	System/Software Qualification Testing Process (6.4.6 - 7.1.7)	13
2.9.1	Obiettivi di qualità	13
2.9.2	Strategie	13
2.9.3	Metriche	13
2.9.3.1	Test di Unità eseguiti	13
2.9.3.2	Test di Integrazione eseguiti	13
2.9.3.3	Test di Sistema eseguiti	13
2.9.3.4	Test di Validazione eseguiti	14
2.9.3.5	Test superati	14
2.10	Software Documentation Management Process (7.2.1)	14
2.10.1	Obiettivi di qualità	14
2.10.2	Strategie	14
2.10.3	Metriche	14
2.10.3.1	Indice Gulpease	14
2.11	Software Verification Process (7.2.4)	15
2.11.1	Obiettivi di qualità	15
2.11.2	Strategie	15
2.11.3	Metriche	15
2.11.3.1	Branch Coverage	15
2.11.3.2	Code Coverage	15
3	Qualità di prodotto	16
3.1	Funzionalità (6.1)	16
3.1.1	Obiettivi di qualità	16
3.1.2	Metriche	16
3.1.2.1	Completezza dell'implementazione funzionale	16
3.1.2.2	Accuratezza rispetto alle attese	16
3.1.2.3	Controllo degli accessi	16
3.2	Affidabilità (6.2)	17
3.2.1	Obiettivi di qualità	17
3.2.2	Metriche	17
3.2.2.1	Densità di failure	17
3.2.2.2	Blocco di operazioni non corrette	17
3.3	Usabilità (6.3)	17
3.3.1	Obiettivi di qualità	17
3.3.2	Metriche	18
3.3.2.1	Comprensibilità delle funzioni offerte	18
3.3.2.2	Facilità di apprendimento delle funzionalità	18



3.3.2.3	Consistenza operativa in uso	18
3.4	Efficienza (6.4)	18
3.4.1	Obiettivi di qualità	18
3.4.2	Metriche	19
3.4.2.1	Tempo di risposta	19
3.5	Manutenibilità (6.5)	19
3.5.1	Obiettivi di qualità	19
3.5.2	Metriche	19
3.5.2.1	Capacità di analisi di failure	19
3.5.2.2	Impatto delle modifiche	19
3.6	Portabilità (6.6)	20
3.6.1	Obiettivi di qualità	20
3.6.2	Metriche	20
3.6.2.1	Versioni dei browser supportate	20
3.6.2.2	Inclusione di funzionalità da altri prodotti	20
4	Specifica dei test	20
4.1	Tipi di test	21
4.1.1	Test di Validazione	21
4.1.2	Test di Sistema	30
4.1.3	Test di Integrazione	35
4.1.4	Test di Unità	37
5	Tracciamento dei test	37
5.1	Tracciamento Test di Validazione-Requisiti	37
5.2	Tracciamento Requisiti-Test di Validazione	38
5.3	Tracciamento Test di Sistema-Requisiti	39
5.4	Tracciamento Requisiti-Test di Sistema	41
5.5	Tracciamento Test di Integrazione-Componenti	43
5.6	Tracciamento Componenti-Test di Integrazione	43
A	Resoconto attività di verifica	44
A.1	Revisione dei Requisiti	44
A.1.1	Tracciamento casi d'uso e requisiti	44
A.1.2	Analisi statica dei documenti	44
A.1.3	Esiti verifiche automatizzate	44



Elenco delle tabelle

1	Test di Validazione	29
2	Test di Sistema	34
3	Test di Integrazione	37
4	Tracciamento Test di Validazione-Requisiti	38
5	Tracciamento Requisiti-Test di Validazione	39
6	Tracciamento Test di Sistema-Requisiti	40
7	Tracciamento Requisiti-Test di Sistema	42
8	Tracciamento Test di Integrazione-Componenti	43
9	Tracciamento Componenti-Test di Integrazione	43
10	Resoconto verifiche automatizzate - Revisione dei Requisiti	44



1 Introduzione

1.1 Scopo del Documento

Lo scopo del presente documento è quello di illustrare le strategie di verifica e validazione che il gruppo TheFellowshipOfTheCode ha deciso di adottare per raggiungere gli obiettivi qualitativi prefissati sul prodotto da sviluppare. Per raggiungere tali obiettivi è necessaria una verifica continua sulle attività svolte. In questo modo sarà possibile individuare rapidamente eventuali anomalie che si potrebbero riscontrare e quindi risolverle in modo tempestivo e senza spreco di risorse, garantendo la correttezza del prodotto e il soddisfacimento del cliente.

1.2 Scopo del prodotto

Lo scopo del prodotto è di permettere la creazione e gestione di questionari in grado di identificare le lacune dei candidati prima, durante e al termine di un corso di formazione.

Il sistema dovrà offrire le seguenti funzionalità:

- Archiviare questionari in un server suddivisi per argomento;
- Somministrare all'utente, tramite un'interfaccia, questionari specifici per argomento scelto;
- Verificare e valutare i questionari scelti dagli utenti in base alle risposte date.

La parte destinata ai creatori di questionari dovrà essere fruibile attraverso un *browser_G* desktop, abilitato all'utilizzo delle tecnologie *HTML5_G*, *CSS3_G* e *JavaScript_G*. La parte destinata agli esaminandi sarà utilizzabile su qualunque dispositivo: dal personal computer ai tablet e smartphone.

1.3 Glossario

Al fine di evitare ogni ambiguità i termini tecnici del dominio del progetto, gli acronimi e le parole che necessitano di ulteriori spiegazioni saranno nei vari documenti marcate con il pedice *G* e quindi presenti nel documento *Glossario*.

1.4 Riferimenti

1.4.1 Normativi

- *NormeDiProgetto_v_3_0_0*;
- **Capitolato**: <http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf>.

1.4.2 Informativi

- **Piano di Progetto**: *Piano di Progetto*;
- **PDCA (Plan-Do-Check-Act)**: <http://it.wikipedia.org/wiki/PDCA>;
- **Standard ISO/IEC 12207:2008 - IEEE Std 12207-2008**: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4475822>;
- **Standard ISO/IEC 15504**: https://en.wikipedia.org/wiki/ISO/IEC_15504;
- **Standard ISO/IEC 9126**: http://it.wikipedia.org/wiki/ISO/IEC_9126;
- **Indice di Gulpease**: http://it.wikipedia.org/wiki/Indice_Gulpease;
- **Analisi dinamica** <http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/L13.pdf>;



2 Qualità di processo

Per garantire la qualità del prodotto è necessario perseguire la qualità dei processi che lo definiscono. Per fare ciò si è deciso di adottare lo standard *ISO/IEC 15504* denominato *SPICE_G* che definisce un modello di riferimento per la valutazione del livello di maturità dei processi.

Per l'esattezza sono previsti sei livelli di maturità, per i quali vengono definiti degli attributi che permettano di misurarla:

- **Level 0 - Incomplete process:** il processo non è implementato o non riesce a raggiungere i suoi obiettivi;
- **Level 1 - Performed process:** il processo viene messo in atto e raggiunge i suoi scopi. Viene misurato tramite:
 - **Process performance:** capacità di raggiungere i propri obiettivi e di ottenere risultati identificabili.
- **Level 2 - Managed process:** il processo viene eseguito sulla base di obiettivi ben definiti. Viene misurato tramite:
 - **Performance management:** capacità di elaborare un prodotto coerente con gli obiettivi attesi;
 - **Work product management:** capacità di elaborare un prodotto appropriatamente documentato, controllato e verificato.
- **Level 3 - Established process:** il processo viene eseguito in base ai principi dell'ingegneria del software. Viene misurato tramite:
 - **Process definition:** capacità di raggiungere i propri obiettivi aderendo agli standard;
 - **Process resource:** capacità di sfruttare risorse adeguate che gli permettano di essere attuato efficacemente.
- **Level 4 - Predictable process:** Il processo è attuato all'interno di limiti ben definiti. Viene misurato tramite:
 - **Process measurement:** capacità di utilizzare i risultati raggiunti e le misure ricavate durante l'esecuzione per garantire il raggiungimento dei traguardi definiti;
 - **Process control:** capacità di correggere o migliorare, se necessario, le sue modalità di esecuzione, in seguito a controlli basati sulle misurazioni rilevate.
- **Level 5 - Optimizing process:** Il processo è predicibile ed in grado di adattarsi per raggiungere obiettivi specifici e rilevanti. Viene misurato tramite:
 - **Process change:** capacità di tenere sotto controllo tutti i cambiamenti strutturali e di esecuzione;
 - **Continuous improvement:** capacità di identificare e implementare le modifiche effettuate, per garantire un miglioramento continuo nella realizzazione degli obiettivi fissati.

Ogni attributo definito precedentemente è misurabile e lo standard stabilisce 4 differenti livelli:

- **N:** non posseduto (0% - 15%);
- **P:** parzialmente posseduto (16% - 50%);
- **L:** largamente posseduto (51% - 85%);



- **F**: completamente posseduto (86% - 100%).

Le misurazioni ottenute vengono utilizzate all'interno della strategia di miglioramento continuo della qualità, realizzata attraverso il ciclo $PDCA_G$, che definisce una metodologia di controllo dei processi durante il loro ciclo di vita. Tale approccio è suddiviso in 4 fasi:

- **Plan**: fase di pianificazione dove si individuano gli obiettivi e i processi necessari per il raggiungimento dei risultati attesi;
- **Do**: fase di attuazione del piano individuato al passo precedente e raccolta di dati sulla qualità ottenuta;
- **Check**: fase di verifica dove si confrontano i risultati ottenuti (fase di Do) ed i risultati attesi (fase di Plan);
- **Act**: fase in cui si determinano le cause delle differenze fra risultati ottenuti e risultati attesi, per decidere dove attuare eventuali azioni correttive per avere un effettivo miglioramento della qualità.

Per garantire una buona qualità di processo, il $team_G$ ha individuato dallo standard $ISO/IEC\ 12207:2008_G$ i processi che ritiene più importanti nell'arco del ciclo di vita del prodotto e li ha istanziati individuando obiettivi e metriche coerenti con i livelli di qualità perseguiti.

2.1 Infrastructure Management Process (6.2.2)

Il processo si pone come obiettivo quello di fornire, mantenere ed aggiornare l'infrastruttura ed i servizi necessari allo svolgimento del progetto nell'arco di tutto il suo ciclo di vita. Con il termine infrastruttura si intendono elementi hardware, software, metodi, strumenti, tecniche e standard impiegati nello sviluppo del prodotto.

2.1.1 Obiettivi di qualità

Per tutta la durata del progetto, l'infrastruttura impiegata nello sviluppo dovrà raggiungere determinati obiettivi; in particolare:

- tutte le procedure riguardanti le attività svolte più frequentemente durante lo sviluppo del progetto saranno descritte esaustivamente nel documento *Norme di Progetto*;
- tutti i riferimenti normativi e informativi saranno completi di informazioni utili al loro reperimento;
- la piattaforma DocumentsDB sarà disponibile all'uso ogniqualvolta un componente del $team_G$ avesse bisogno di accedere ai dati in essa contenuti;
- i dati ottenuti da DocumentsDB saranno sempre coerenti e aggiornati;
- nell'arco dello sviluppo del progetto, DocumentsDB si estenderà fornendo nuove funzionalità in relazione alle nuove attività intraprese; esse risulteranno implementate e funzionanti alla prima necessità di utilizzo;

2.1.2 Strategie

L'infrastruttura necessaria allo svolgimento del progetto dovrà essere mantenuta costantemente aggiornata; in particolare l'utilizzo delle metriche sotto indicate permetterà l'individuazione di eventuali errori all'interno degli strumenti utilizzati, la cui correzione permetterà di ripristinare l'erogazione di dati corretti e coerenti.



2.1.3 Metriche

2.1.3.1 Disponibilità DocumentsDB

Indica la percentuale di disponibilità di utilizzo della piattaforma DocumentsDB rispetto alle richieste di accesso.

- **Misurazione:** $D = \frac{A}{R} \cdot 100$, dove A corrisponde al numero di accessi avvenuti correttamente alla pagina di login della piattaforma e R il numero totale di richieste di accesso alla pagina di login inoltrate alla piattaforma;
- **Range-ottimale:** 90 – 100;
- **Range-accettazione:** 80 – 100.

2.1.3.2 Tempo di correzione incoerenze in DocumentsDB

Indica il periodo medio intercorso fra l'individuazione di un'incoerenza nella piattaforma DocumentsDB da parte di un verificatore ed il suo aggiornamento da parte di un altro componente del $team_G$.

- **Misurazione:** $T = \frac{\sum_{i=1}^n C_i}{n}$ (con T espresso in giorni) dove C_i è il tempo intercorso fra il momento di individuazione dell'incoerenza i in DocumentsDB e l'istante in cui tale dato viene corretto;
- **Range-ottimale:** 0 – 1;
- **Range-accettazione:** 0 – 3.

2.2 Project Planning, Assessment & Control Process (6.3.1 - 6.3.2)

Il macro-processo (derivante dall'unione dei processi Project Planning Process e Project Assessment & Control Process) ha lo scopo di produrre dei piani di sviluppo per il progetto, comprendenti la scelta del modello di ciclo di vita del prodotto, descrizioni delle attività e dei compiti da svolgere, pianificazione temporale del lavoro e dei costi da sostenere, allocazione di compiti e responsabilità, e misurazioni per rilevare lo stato del progetto rispetto alle pianificazioni prodotte.

2.2.1 Obiettivi di qualità

L'intero sviluppo del progetto dovrà seguire la pianificazione prodotta, in particolare:

- ogni attività verrà svolta da parte di colui al quale è stata assegnata, rispettando le tempistiche fissate e svolgendo tutti i compiti nei quali è stata suddivisa;
- il costo necessario allo svolgimento di una fase di progetto non dovrà eccedere quanto preventivato per tale fase.

2.2.2 Strategie

La pianificazione effettuata dovrà essere aggiornata costantemente durante tutta l'attività di progetto per essere sempre coerente con la situazione corrente. Qualsiasi eventuale valore negativo a livello di Schedule Variance o Budget Variance rilevato in una fase di lavoro dovrà essere assolutamente compensato entro la fine dell'attività di progetto, in quanto non è assolutamente ammesso eccedere le ore di lavoro finali e il preventivo dei costi finale indicato nella pianificazione.



2.2.3 Metriche

2.2.3.1 Schedule Variance

Indica se si è in linea, in anticipo o in ritardo rispetto la pianificazione temporale delle attività nella $baseline_G$.

- **Misurazione:** $SV = BCWP - BCWS$, dove $BCWP$ sono le attività completate ad un certo momento e $BCWS$ le attività che, secondo la pianificazione, dovrebbero essere state completate a quel momento;
- **Range-ottimale:** ≥ 0 ;
- **Range-accettazione:** ≥ 0 .

2.2.3.2 Budget Variance

Indica se alla data corrente si è speso di più o di meno rispetto a quanto pianificato.

- **Misurazione:** $BV = BCWS - ACWP$, dove $BCWS$ è il costo pianificato per realizzare le attività di progetto alla data corrente e $ACWP$ è il costo effettivamente sostenuto alla data corrente;
- **Range-ottimale:** ≥ 0 ;
- **Valore di accettazione:** ≥ 0 .

2.3 Risk Management Process (6.3.4)

L'obiettivo del processo è quello di identificare, analizzare, trattare e monitorare continuamente i rischi che possono insorgere durante l'intera attività di progetto.

2.3.1 Obiettivi di qualità

Il $team_G$ dovrà gestire correttamente i rischi, in particolare:

- all'inizio dell'attività di progetto, verranno individuati i principali fattori di rischio riguardanti l'organizzazione delle attività;
- all'inizio di ogni fase, l'analisi dei rischi porterà all'individuazione di nuovi rischi specifici per tale fase;
- i rischi analizzati che si paleseranno saranno trattati secondo le strategie individuate in fase di individuazione e il loro impatto sarà controllato.

2.3.2 Strategie

Il livello di probabilità dei rischi analizzati dovrà sempre essere tenuto sotto controllo. Anche se a basso livello di pericolosità, in caso il rischio si manifestasse, il $team_G$ dovrà attuare le contromisure previste al fine di mitigare i suoi effetti ed evitare che la sua pericolosità aumenti.

2.3.3 Metriche

2.3.3.1 Rischi non preventivati

Indicatore che evidenzia i rischi non preventivati.

- **Misurazione:** indice numerico che viene incrementato nel momento in cui si manifesta un rischio non individuato nell'attività di analisi dei rischi;



- **Range-ottimale:** 0;
- **Range-accettazione:** 0 – 5.

2.3.3.2 Efficienza di gestione dei rischi

Misura il tempo medio trascorso fra l'individuazione di un rischio e il momento in cui manifesta in modo problematico i suoi effetti.

- **Misurazione:** $E = \frac{\sum_{i=1}^n (M_i \cdot P_i)}{\sum_{i=1}^n P_i}$, (con E espresso in giorni) dove M_i è il tempo intercorso fra l'individuazione del rischio i e l'istante in cui manifesta in modo problematico i suoi effetti, espresso in giorni, e P_i corrisponde al grado di pericolosità del rischio i , valutato in scala crescente [1-5];
- **Range-ottimale:** ≥ 60 ;
- **Range-accettazione:** ≥ 20 .

2.4 System/Software Requirements Analysis Process (6.4.2 - 7.1.2)

Il processo punta a trasformare i requisiti individuati dalle fonti in un set di requisiti tecnici che fungerà da linea guida nella progettazione del sistema.

2.4.1 Obiettivi di qualità

I requisiti identificati dal *team_G* dovranno essere gestiti in maniera tale da raggiungere i seguenti traguardi:

- per ogni requisito verrà tenuta traccia della fonte da cui è stato ricavato;
- per ogni requisito dovrà essere possibile indicare dei test, da effettuare per verificarne il soddisfacimento da parte del prodotto;
- per ogni requisito sarà possibile ricostruire i cambiamenti principali effettuati nella sua formulazione, durante tutto il ciclo di sviluppo del prodotto;
- nessun requisito dovrà risultare superfluo o ambiguo;
- tutti i requisiti che il prodotto andrà a soddisfare saranno stati precedentemente approvati dai committenti.

2.4.2 Strategie

Tutti i requisiti individuati dovranno essere correttamente inseriti nella piattaforma DocumentsDB, la quale si occuperà di mantenere traccia delle fonti dalle quali derivano, delle modifiche effettuate e della loro implementazione nel prodotto.

2.4.3 Metriche

2.4.3.1 Requisiti obbligatori soddisfatti

Indica la percentuale dei requisiti obbligatori soddisfatti dal prodotto.

- **Misurazione:** $S = \frac{N_S}{N_O} \cdot 100$, dove N_S è il numero dei requisiti obbligatori soddisfatti dal sistema e N_O è il numero dei requisiti obbligatori identificati;
- **Range-ottimale:** 100;
- **Range-accettazione:** 100.



2.5 System/Software Architectural Design Process (6.4.3 - 7.1.3)

Il processo si pone come obiettivo quello di identificare una corrispondenza fra requisiti di sistema ed elementi del sistema.

2.5.1 Obiettivi di qualità

Durante lo svolgimento delle attività previste da questo processo, il *team_G* punterà a definire un'architettura adatta agli scopi del progetto:

- ogni componente progettato come parte del sistema risulterà essere necessario per il funzionamento del prodotto e, quindi, costantemente tracciabile ai requisiti che soddisfa;
- il sistema dovrà presentare basso accoppiamento ed alta coesione
- ogni componente dovrà essere progettato puntando su incapsulamento, modularizzazione e riuso di codice.

2.5.2 Strategie

Nel corso dell'attività di progettazione, sia ad alto livello che di dettaglio, le componenti verranno inserite nella piattaforma DocumentsDB, la quale si occuperà di mantenere aggiornati i tracciamenti fra esse ed i requisiti che soddisfano, oltre alle relazioni presenti fra le varie componenti.

2.5.3 Metriche

2.5.3.1 Structural Fan-In

In riferimento ad un modulo del software, misura quanti altri moduli lo utilizzano durante la loro esecuzione; tale indicazione permette di stabilire il livello di riuso implementato.

- **Misurazione:** indice numerico che incrementa nel momento in cui viene individuato un modulo che, durante la sua esecuzione, chiama il modulo in oggetto;
- **Range-ottimale:** ≥ 2 ;
- **Range-accettazione:** ≥ 0 .

2.5.3.2 Structural Fan-Out

In riferimento ad un modulo del software, misura quanti moduli vengono utilizzati durante la sua esecuzione; tale indicazione permette di stabilire il livello di accoppiamento implementato.

- **Misurazione:** indice numerico che incrementa nel momento in cui viene individuato un modulo utilizzato dal modulo in oggetto durante la sua esecuzione;
- **Range-ottimale:** $0 - 1$;
- **Range-accettazione:** $0 - 5$.

2.6 Software Detailed Design Process (7.1.4)

Lo scopo del processo è fornire una progettazione di dettaglio del prodotto che andrà ad implementare i requisiti individuati.



2.6.1 Obiettivi di qualità

Le attività svolte dovranno raggiungere i seguenti obiettivi:

- il livello di dettaglio della progettazione dovrà essere tale da guidare codifica e testing senza bisogno di informazioni aggiuntive, indicando metodi con i relativi parametri e campi dati forniti da ciascuna classe;
- la struttura a basso livello dell'architettura e le relazioni fra le varie unità software concepite saranno esposte chiaramente nel documento di *Definizione di Prodotto*, che definirà dettagliatamente cosa implementare;
- oltre alle unità software individuate, le attività permetteranno di definire dettagliatamente le interfacce fra esse costituite.

2.6.2 Strategie

Sarà necessario effettuare un'analisi dettagliata delle componenti individuate in progettazione architeturale, suddividendole in unità che siano facilmente codificabili e testabili per le attività successive.

2.6.3 Metriche

2.6.3.1 Numero di metodi per classe

Indica il numero di metodi definiti in una classe; un valore molto alto potrebbe indicare una cattiva decomposizione delle funzionalità a livello di progettazione.

- **Misurazione:** indice numerico che indica il numero di metodi definiti in una classe;
- **Range-ottimale:** 1 – 7;
- **Range-accettazione:** 1 – 10.

2.6.3.2 Numero di parametri per metodo

Indica il numero di parametri passati ad un metodo; un valore molto alto potrebbe indicare un metodo troppo complesso e non efficacemente suddiviso in sotto-metodi.

- **Misurazione:** indice numerico che indica il numero di parametri passato ad un metodo;
- **Range-ottimale:** 0 – 4;
- **Range-accettazione:** 0 – 8.

2.7 Software Construction Process (7.1.5)

Il processo definisce le attività principali volte alla produzione di unità software eseguibili che riflettano quanto identificato a livello di progettazione.

2.7.1 Obiettivi di qualità

Le unità software prodotte dovranno risultare di qualità; a questo fine il $team_G$ si è posto i seguenti obiettivi:

- l'implementazione delle classi e dei metodi definiti in progettazione dovrà puntare a produrre codice a bassa complessità, in modo tale che quanto prodotto risulti facilmente comprensibile e testabile;



- l'uso di costrutti e tecniche che creano sdoppiamenti del flusso di esecuzione verrà valutato attentamente ed attuato solo se strettamente necessario;
- il codice prodotto dovrà risultare facilmente manutenibile;
- il codice prodotto risulterà privo di elementi inutilizzati.

2.7.2 Strategie

Durante l'attività di codifica, il *Programmatore* dovrà attenersi a quanto indicato nel documento *Definizione di Prodotto*, concentrandosi (in particolare) nel limitare la complessità del codice prodotto. Sarà necessario inoltre procedere con la codifica dei test individuati nell'attività di progettazione, in modo tale da consentire la verifica del corretto funzionamento delle varie unità prodotte.

2.7.3 Metriche

2.7.3.1 Produttività di codifica

Indica il numero medio di linee di codice prodotto per ora/persona.

- **Misurazione:** $P = \frac{N_{SLOC}}{h_P}$, dove N_{SLOC} è il numero di Source Lines Of Code prodotte e h_P è il numero di ore/persona utilizzate per la codifica;
- **Range-ottimale:** ≥ 10 ;
- **Range-accettazione:** ≥ 3 .

2.7.3.2 Complessità Ciclomatica

Indica la complessità di funzioni, moduli, metodi o classi di un programma misurando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso. Alti valori di *complessità ciclomatica_G* implicano una ridotta manutenibilità del codice. Valori bassi di *complessità ciclomatica_G* potrebbero però determinare scarsa efficienza dei metodi.

- **Misurazione:** indice numerico che indica il numero cammini percorribili nel grafo di controllo di flusso di un metodo;
- **Range-ottimale:** $1 - 10$;
- **Range-accettazione:** $1 - 15$.

2.7.3.3 Numero di livelli di annidamento

Indica il numero di funzioni o procedure chiamate all'interno di un metodo; un valore elevato di tale indice implica un'alta complessità ed un basso livello di astrazione del codice.

- **Misurazione:** indice numerico che indica il numero di chiamate a funzioni o procedure presenti all'interno di un metodo;
- **Range-ottimale:** $1 - 3$;
- **Range-accettazione:** $1 - 6$.



2.7.3.4 Linee di codice per linee di commento

Indica la percentuale di linee di commento presenti all'interno del codice sorgente; la loro presenza permette una più semplice comprensione ed un maggior livello di manutenibilità di quanto prodotto.

- **Misurazione:** $P = \frac{N_C}{N_{SLOC}} \cdot 100$, dove N_C è il numero di linee di commento presenti nel codice e N_{SLOC} è il numero di Source Lines Of Code prodotte;
- **Range-ottimale:** ≥ 30 ;
- **Range-accettazione:** ≥ 25 .

2.7.3.5 Variabili inutilizzate

Indica la percentuale di variabili dichiarate che non vengono mai utilizzate durante l'esecuzione.

- **Misurazione:** $P = \frac{N_{VI}}{N_{VD}} \cdot 100$, dove N_{VI} è il numero di variabili che non vengono mai utilizzate e N_{VD} è il numero di variabili dichiarate;
- **Range-ottimale:** 0;
- **Range-accettazione:** 0.

2.7.3.6 Dipendenze

Misura il numero medio di chiamate require di una funzione, analizzate staticamente considerando la firma del metodo.

- **Misurazione:** $D = \frac{\sum_{i=1}^n R_i}{n}$, dove R_i è il numero di chiamate require effettuate dalla funzione i ;
- **Range-ottimale:** 0 – 5;
- **Range-accettazione:** 0 – 10.

2.7.3.7 Halstead Difficulty per-function

Misura il livello di complessità di una funzione.

- **Misurazione:** $DIF = \frac{UOP}{2} \cdot \frac{OD}{UOD}$, dove UOP è il numero di operatori distinti, OD è il numero totale di operandi e UOD è il numero di operandi distinti;
- **Range-ottimale:** 0 – 15;
- **Range-accettazione:** 0 – 25.

2.7.3.8 Halstead Volume per-function

Indica la dimensione dell'implementazione di un algoritmo; si basa sul numero di operazioni eseguite e sugli operandi di una funzione.

- **Misurazione:** $VOL = (OP + OD) \cdot \log_2(UOP + UOD)$, dove OP è il numero totale di operatori, OD è il numero totale di operandi, UOP è il numero di operatori distinti e UOD è il numero di operandi distinti;
- **Range-ottimale:** 20 – 1000;
- **Range-accettazione:** 20 – 1500.



2.7.3.9 Halstead Effort per-function

Rappresenta il costo necessario a scrivere il codice di una funzione.

- **Misurazione:** $E = DIF \cdot VOL$, dove DIF indica l'Halstead Difficulty e VOL è l'Halstead Volume;
- **Range-ottimale:** 0 – 300;
- **Range-accettazione:** 0 – 400.

2.7.3.10 Indice di manutenibilità

Permette di stabilire quanto sarà semplice mantenere il codice prodotto.

- **Misurazione:** $MI = 171 - 3.42 \cdot \ln(aveE) - 0.23 \cdot \ln(aveV) - 16.2 \cdot \ln(aveLOC)$, dove $aveE$ è l'Halstead Effort medio per modulo, $aveV$ è la complessità ciclomatica media per modulo, e $aveLOC$ è il numero medio di linee di codice per modulo;
- **Range-ottimale:** 120 – 171;
- **Range-accettazione:** 100 – 171.

2.8 System/Software Integration Process (6.4.5 - 7.1.6)

Il processo si occupa di integrare fra loro gli elementi del sistema, rispettando quanto stabilito nell'attività di progettazione, al fine di produrre un prodotto completo tale da soddisfare quanto espresso dai requisiti identificati.

2.8.1 Obiettivi di qualità

Le attività previste da questo processo dovranno puntare a raggiungere un alto livello di automazione, in particolare:

- l'integrazione delle varie parti del sistema sarà completamente automatizzata utilizzando lo strumento di continuous integration Jenkins;
- il livello di integrazione raggiunto del sistema sarà sempre consultabile grazie all'utilizzo dello strumento di continuous integration Jenkins.

2.8.2 Strategie

Sarà necessario configurare accuratamente lo strumento di continuous integration Jenkins affinché esegua dei test di integrazione di quanto prodotto prima che le ultime modifiche diventino parte del sistema.

2.8.3 Metriche

2.8.3.1 Componenti integrate

Indica la percentuale di componenti progettate, attualmente implementate e correttamente integrate nel sistema.

- **Misurazione:** $I = \frac{N_{CI}}{N_{CP}} \cdot 100$, dove N_{CI} è il numero di componenti attualmente integrate nel sistema e N_{CP} è il numero di componenti delineate nell'attività di progettazione;
- **Range-ottimale:** 100;
- **Range-accettazione:** 100.



2.9 System/Software Qualification Testing Process (6.4.6 - 7.1.7)

Lo scopo del processo è quello di assicurare che ogni requisito individuato sia stato implementato nel prodotto.

2.9.1 Obiettivi di qualità

Durante lo svolgimento delle attività, ci si impegnerà affinché:

- le attività di test previste dal processo verranno svolte su un sistema le cui componenti sono verificate e correttamente integrate fra loro;
- il sistema dovrà implementare tutti i requisiti obbligatori individuati nell'attività di analisi.

2.9.2 Strategie

Bisognerà cercare di implementare il maggior livello possibile di automazione nell'esecuzione dei test di sistema, in modo tale che la loro esecuzione non richieda costi eccessivi (soprattutto in termini temporali) e sia possibile eseguirne un numero sufficiente a garantire un'ottima copertura dei requisiti; a tal fine molto importante sarà lo strumento di continuous integration Jenkins, il quale andrà opportunamente configurato per eseguire i test stabiliti.

2.9.3 Metriche

2.9.3.1 Test di Unità eseguiti

Indica la percentuale di test di unità eseguiti.

- **Misurazione:** $UE = \frac{N_{TUE}}{N_{TUP}} \cdot 100$, dove N_{TUE} è il numero di test di unità eseguiti e N_{TUP} è il numero di test di unità pianificati;
- **Range-ottimale:** 100;
- **Range-accettazione:** 90 – 100.

2.9.3.2 Test di Integrazione eseguiti

Indica la percentuale di test di integrazione eseguiti.

- **Misurazione:** $IE = \frac{N_{TIE}}{N_{TIP}} \cdot 100$, dove N_{TIE} è il numero di test di integrazione eseguiti e N_{TIP} è il numero di test di integrazione pianificati;
- **Range-ottimale:** 70 – 100;
- **Range-accettazione:** 60 – 100.

2.9.3.3 Test di Sistema eseguiti

Indica la percentuale di test di sistema eseguiti in modo automatico.

- **Misurazione:** $SE = \frac{N_{TSE}}{N_{TSP}} \cdot 100$, dove N_{TSE} è il numero di test di sistema eseguiti e N_{TSP} è il numero di test di sistema pianificati;
- **Range-ottimale:** 80 – 100;
- **Range-accettazione:** 70 – 100.



2.9.3.4 Test di Validazione eseguiti

Indica la percentuale di test di validazione eseguiti manualmente.

- **Misurazione:** $VE = \frac{N_{TVE}}{N_{TVP}} \cdot 100$, dove N_{TVE} è il numero di test di validazione eseguiti e N_{TVP} è il numero di test di validazione pianificati;
- **Range-ottimale:** 100;
- **Range-accettazione:** 100.

2.9.3.5 Test superati

Indica la percentuale di test superati.

- **Misurazione:** $S = \frac{N_{TS}}{N_{TE}} \cdot 100$, dove N_{TS} è il numero di test superati e N_{TE} è il numero di test eseguiti;
- **Range-ottimale:** 100;
- **Range-accettazione:** 90 – 100.

2.10 Software Documentation Management Process (7.2.1)

Il processo punta a produrre e mantenere le informazioni sul software prodotte dai processi attuati.

2.10.1 Obiettivi di qualità

Il processo di documentazione dovrà perseguire le seguenti direttive:

- la documentazione prodotta dovrà essere chiara e comprensibile a tutti gli *stakeholder_G* e sarà resa disponibile alle parti interessate per la consultazione;
- ogni forma di ambiguità sul significato di un termine utilizzato verrà eliminata grazie al *Glossario*;
- la documentazione prodotta sarà sempre aggiornata ed allineata allo stato attuale del processo di sviluppo del prodotto.

2.10.2 Strategie

Durante la stesura della documentazione, ogni termine con significato ambiguo deve essere indicato (corredato di definizione) nel *Glossario*; gli script automatici provvederanno alla sua segnalazione all'interno del documento.

Ogni documento sarà dotato di numero di versione e corredato da un diario delle modifiche che consente di prendere visione di tutte le azioni effettuate sul testo in oggetto.

2.10.3 Metriche

2.10.3.1 Indice Gulpease

L'*indice Gulpease_G* è un indice di leggibilità di un testo tarato sulla lingua italiana che permette di misurare la complessità dello stile di un documento considerando due variabili linguistiche, la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

- **Misurazione:** $G = 89 + \frac{300 \cdot N_F - 10 \cdot N_L}{N_P}$, dove N_F è il numero di frasi, N_L è il numero di lettere e N_P è il numero di parole presenti nel testo;



- **Range-ottimale:** 50 – 100;
- **Range-accettazione:** 40 – 100.

2.11 Software Verification Process (7.2.4)

Il processo punta a verificare se qualsiasi elemento del sistema soddisfa completamente i requisiti ad esso correlati.

2.11.1 Obiettivi di qualità

Al fine di garantire qualità nell'attuazione del processo:

- la documentazione verrà verificata attraverso *inspection_G*, poiché permette risparmio in termini di tempi e costi;
- i test dinamici effettuati sui vari elementi saranno il più possibile automatizzabili;
- i test dinamici effettuati sui vari elementi del software copriranno una grande parte delle possibili casistiche d'utilizzo.

2.11.2 Strategie

Durante le attività di correzione della documentazione, gli errori più frequenti rilevati saranno riportati in un documento, in modo tale da diventare uno dei punti chiave delle successive attività di *inspection_G*.

Per ogni test effettuato verrà tenuto tracciamento del suo esito.

2.11.3 Metriche

2.11.3.1 Branch Coverage

Indica la percentuale di rami decisionali percorsi dai test di unità utilizzati.

- **Misurazione:** $BC = \frac{R_P}{R_T} \cdot 100$, dove R_P è il numero dei rami decisionali percorsi dai test e R_T è il numero di rami decisionali definiti nel software;
- **Range-ottimale:** 80 – 100;
- **Range-accettazione:** 70 – 100.

2.11.3.2 Code Coverage

Indica la percentuale di istruzioni che sono eseguite durante i test. Maggiore è la percentuale di istruzioni coperte dai test eseguiti, maggiore sarà la probabilità che le componenti testate abbiano una ridotta quantità di errori. Il valore di tale indice può essere abbassato da metodi molto semplici che non richiedono testing. Esempi di questi metodi sono: get e set.

- **Misurazione:** $CC = \frac{L_M}{L_T} \cdot 100$, dove L_M è il numero di linee di codice monitorata dai test e L_T è il numero di linee di codice implementate nel software;
- **Range-ottimale:** 65 – 100;
- **Range-accettazione:** 42 – 100.



3 Qualità di prodotto

Per garantire una buona qualità di prodotto, il $team_G$ ha individuato dallo standard *ISO/IEC 9126_G* le qualità che ritiene più importanti nell'arco del ciclo di vita del prodotto e le ha istanziate individuando obiettivi e metriche coerenti con i livelli di qualità perseguiti.

3.1 Funzionalità (6.1)

Rappresenta la capacità del prodotto di fornire tutte le funzioni che sono state individuate attraverso l'*Analisi dei Requisiti*

3.1.1 Obiettivi di qualità

Il $team_G$ si impegnerà affinché:

- **Adeguatezza (6.1.1):** le funzionalità fornite siano conformi rispetto le aspettative;
- **Accuratezza (6.1.2):** il prodotto fornisca i risultati attesi, con il livello di dettaglio richiesto;
- **Sicurezza (6.1.4):** il prodotto protegga le informazioni e i dati da accessi e modifiche non autorizzati.

3.1.2 Metriche

3.1.2.1 Completezza dell'implementazione funzionale

Indica la percentuale di requisiti funzionali coperti dall'implementazione.

- **Misurazione:** $C = (1 - \frac{N_{FM}}{N_{FI}}) \cdot 100$, dove N_{FM} è il numero di funzionalità mancanti nell'implementazione e N_{FI} è il numero di funzionalità individuate nell'attività di analisi;
- **Range-ottimale:** 100;
- **Range-accettazione:** 100.

3.1.2.2 Accuratezza rispetto alle attese

Indica la percentuale di risultati concordi alle attese.

- **Misurazione:** $A = (1 - \frac{N_{RD}}{N_{TE}}) \cdot 100$, dove N_{RD} è il numero di test che producono risultati discordanti rispetto alle attese e N_{TE} è il numero di test-case eseguiti;
- **Range-ottimale:** 100;
- **Range-accettazione:** 90 – 100.

3.1.2.3 Controllo degli accessi

Indica la percentuale di operazioni illegali non bloccate.

- **Misurazione:** $I = \frac{N_{IE}}{N_{II}} \cdot 100$, dove N_{IE} è il numero di operazioni illegali effettuabili dai test e N_{II} è il numero di operazioni illegali individuate;
- **Range-ottimale:** 0;
- **Range-accettazione:** 0 – 10.



3.2 Affidabilità (6.2)

Rappresenta la capacità del prodotto software di svolgere correttamente le sue funzioni durante il suo utilizzo, anche nel caso in cui si presentino situazioni anomale.

3.2.1 Obiettivi di qualità

L'esecuzione del prodotto dovrà presentare le seguenti caratteristiche:

- **Maturità (6.2.1):** evitare che si verifichino malfunzionamenti, operazioni illegali e restituzione di risultati errati (failure) in seguito a fault;
- **Tolleranza agli errori (6.2.2):** nel caso in cui si presentino degli errori, dovuti a guasti o ad un uso scorretto dell'applicativo, questi devono essere gestiti in modo da mantenere alto il livello di prestazione.

3.2.2 Metriche

3.2.2.1 Densità di failure

Indica la percentuale di operazioni di testing che si sono concluse in failure.

- **Misurazione:** $F = \frac{N_{FR}}{N_{TE}} \cdot 100$, dove N_{FR} è il numero di failure rilevati durante l'attività di testing e N_{TE} è il numero di test-case eseguiti;
- **Range-ottimale:** 0;
- **Range-accettazione:** 0 – 10.

3.2.2.2 Blocco di operazioni non corrette

Indica la percentuale di funzionalità in grado di gestire correttamente i fault che potrebbero verificarsi.

- **Misurazione:** $B = \frac{N_{FE}}{N_{ON}} \cdot 100$, dove N_{FE} è il numero di failure evitati durante i test effettuati e N_{ON} è il numero di test-case eseguiti che prevedono l'esecuzione di operazioni non corrette, causa di possibili failure;
- **Range-ottimale:** 100;
- **Range-accettazione:** 80 – 100.

3.3 Usabilità (6.3)

Rappresenta la capacità del prodotto di essere facilmente comprensibile e attraente in ogni sua parte per qualsiasi utente che lo andrà ad utilizzare.

3.3.1 Obiettivi di qualità

Il prodotto dovrà puntare ai seguenti obiettivi di usabilità:

- **Comprensibilità (6.3.1):** l'utente deve essere in grado di riconoscerne le funzionalità offerte dal software e deve comprenderne le modalità di utilizzo per riuscire a raggiungere i risultati attesi;
- **Apprendibilità (6.3.2):** deve essere data la possibilità all'utente di imparare ad utilizzare l'applicazione senza troppo impegno;



- **Operabilità (6.3.3):** le funzionalità presenti devono essere coerenti con le aspettative dell'utente;
- **Attrattiva (6.3.4):** il software deve essere piacevole per chi ne fa uso.

3.3.2 Metriche

3.3.2.1 Comprensibilità delle funzioni offerte

Indica la percentuale di operazioni comprese in modo immediato dall'utente, senza la consultazione del manuale.

- **Misurazione:** $C = \frac{N_{FC}}{N_{FO}} \cdot 100$, dove N_{FC} è il numero di funzionalità comprese in modo immediato dall'utente durante l'attività di testing del prodotto e N_{FO} è il numero di funzionalità offerte dal sistema;
- **Range-ottimale:** 90 – 100;
- **Range-accettazione:** 80 – 100.

3.3.2.2 Facilità di apprendimento delle funzionalità

Indica il tempo medio impiegato dall'utente nell'imparare ad usare correttamente una data funzionalità.

- **Misurazione:** indicatore numerico, espresso in minuti che tiene traccia del tempo medio impiegato dall'utente nell'apprendere il corretto utilizzo di una funzionalità offerta dal sistema;
- **Range-ottimale:** 0 – 15;
- **Range-accettazione:** 0 – 30.

3.3.2.3 Consistenza operativa in uso

Indica la percentuale di messaggi e funzionalità offerte all'utente che rispettano le sue aspettative riguardo al comportamento del software.

- **Misurazione:** $C = (1 - \frac{N_{MFI}}{N_{MFO}}) \cdot 100$, dove N_{MFI} è il numero di messaggi e funzionalità che non rispettano le aspettative dell'utente e N_{MFO} è il numero di messaggi e funzionalità offerti dal sistema;
- **Range-ottimale:** 90 – 100;
- **Range-accettazione:** 80 – 100.

3.4 Efficienza (6.4)

Rappresenta la capacità di eseguire le funzionalità offerte dal software nel minor tempo possibile utilizzando al tempo stesso il minor numero di risorse possibili.

3.4.1 Obiettivi di qualità

Il prodotto dovrà essere efficiente, in particolare:

- **Comportamento rispetto al tempo (6.4.1):** per svolgere le sue funzioni il software deve fornire adeguati tempi di risposta ed elaborazione;
- **Utilizzo delle risorse (6.4.2):** il software quando esegue le sue funzionalità deve utilizzare un appropriato numero e tipo di risorse.



3.4.2 Metriche

3.4.2.1 Tempo di risposta

Indica il periodo temporale medio che intercorre fra la richiesta al software di una determinata funzionalità e la restituzione del risultato all'utente.

- **Misurazione:** $T_{RISP} = \frac{\sum_{i=1}^n T_i}{n}$ (con T_{RISP} espresso in *secondi*) dove T_i è il tempo intercorso fra la richiesta i di una funzionalità ed il completamento delle operazioni necessarie a restituire un risultato a tale richiesta;
- **Range-ottimale:** 0 – 3;
- **Range-accettazione:** 0 – 8.

3.5 Manutenibilità (6.5)

Rappresenta la capacità del prodotto di essere modificato, tramite correzioni, miglioramenti o adattamenti del software a cambiamenti negli ambienti, nei requisiti e nelle specifiche funzionali.

3.5.1 Obiettivi di qualità

Le operazioni di manutenzione andranno agevolate il più possibile adottando le seguenti caratteristiche:

- **Analizzabilità (6.5.1):** il software deve consentire una rapida identificazione delle possibili cause di errori e malfunzionamenti;
- **Modificabilità (6.5.2):** il prodotto originale deve permettere eventuali cambiamenti in alcune sue parti;
- **Stabilità (6.5.3):** non devono insorgere effetti indesiderati in seguito a modifiche effettuate sul software;
- **Testabilità (6.5.4):** il software deve poter essere facilmente testato per validare le modifiche effettuate.

3.5.2 Metriche

3.5.2.1 Capacità di analisi di failure

Indica la percentuale di failure registrate, delle quali sono state individuate le cause.

- **Misurazione:** $I = \frac{N_{FI}}{N_{FR}} \cdot 100$, dove N_{FI} è il numero di failure delle quali sono state individuate le cause e N_{FR} è il numero di failure rilevate;
- **Range-ottimale:** 80 – 100;
- **Range-accettazione:** 60 – 100.

3.5.2.2 Impatto delle modifiche

Indica la percentuale di modifiche effettuate in risposta a failure che hanno portato all'introduzione di nuove failure in altre componenti del sistema.

- **Misurazione:** $I = \frac{N_{FRF}}{N_{FR}} \cdot 100$, dove N_{FRF} è il numero di failure risolte con l'introduzione di nuove failure e N_{FR} è il numero di failure risolte;
- **Range-ottimale:** 0 – 10;
- **Range-accettazione:** 0 – 20.



3.6 Portabilità (6.6)

Rappresenta la capacità del software di poter essere utilizzato su diversi ambienti.

3.6.1 Obiettivi di qualità

Sarà agevolata la portabilità del prodotto adottando i seguenti obiettivi:

- **Adattabilità (6.6.1):** il prodotto deve adattarsi a tutti quegli ambienti di lavoro nei quali è stato previsto un suo utilizzo, senza dover apportare modifiche allo stesso;
- **Sostituibilità (6.6.4):** l'applicativo deve poter sostituire un altro software che ha lo stesso scopo e lavora nel medesimo ambiente.

3.6.2 Metriche

3.6.2.1 Versioni dei browser supportate

Indica la percentuale di versioni di *browser_G* attualmente supportate, fra quelle individuate dai requisiti.

- **Misurazione:** $S = \frac{N_{VS}}{N_{VI}} \cdot 100$, dove N_{VS} è il numero di versioni di *browser_G* supportate dal prodotto e N_{VI} è il numero di versioni di *browser_G* che devono essere supportate dal prodotto;
- **Range-ottimale:** 100;
- **Range-accettazione:** 100.

3.6.2.2 Inclusione di funzionalità da altri prodotti

Indica la percentuale di funzionalità del software utilizzato in precedenza dall'utente che produce risultati simili a quelli ottenuti dal prodotto in oggetto.

- **Misurazione:** $I = \frac{N_{FPA}}{N_{FPP}} \cdot 100$, dove N_{FPA} è il numero di funzionalità del software utilizzato in precedenza dall'utente che produce risultati simili a quelli ottenuti dal prodotto in oggetto e N_{FPP} è il numero di funzionalità offerte dal software utilizzato in precedenza dall'utente;
- **Range-ottimale:** 90 – 100;
- **Range-accettazione:** 80 – 100.

4 Specifica dei test

Il *team_G* TheFellowshipOfTheCode, al fine di implementare del software che sia di qualità, ha strutturato dei test atti a verificare che il software prodotto rispecchi le funzionalità a fronte di risultati attesi. Questi test sono il frutto dell'applicazione delle tecniche di verifica dinamica che sono state introdotte nel documento Norme di Progetto. Tutte le attività di testing prodotte devono poter essere ripetibili e devono essere deterministiche, al fine di poter fornire delle informazioni utili a intraprendere azioni di correzione, nel caso in cui i risultati ottenuti siano diversi da quelli attesi. Per avere un tracciamento dei test prodotti e dei risultati ottenuti si è scelto di classificare il tutto producendo dei log che siano di facile consultazione e che possano fornire una precisa indicazione di quelli che sono stati gli output di queste attività di verifica, eventuali errori o eventuali risultati che siano non coerenti con quanto in precedenza fissato.



4.1 Tipi di test

Sono stati individuati quattro livelli di testing e sono rispettivamente:

- **Test di unità [TU]:** con questa tipologia di test si cerca di verificare la più piccola parte di lavoro prodotta da un programmatore. Questo si traduce tendenzialmente a verificare i metodi e le funzioni scritte;
- **Test di integrazione [TI]:** con questa tipologia di test si cerca di verificare le componenti di sistema. Più precisamente, l'obiettivo è quello di testare il funzionamento dei vari package prodotti, sia singolarmente che nel loro insieme;
- **Test di sistema [TS]:** con questa tipologia di test si cerca di verificare che il comportamento e il funzionamento dell'architettura siano corretti;
- **Test di validazione [TV]:** con questa tipologia di test si vuole verificare che il lavoro prodotto soddisfi quanto richiesto dal proponente.

4.1.1 Test di Validazione

I test di validazione hanno lo scopo di accertare che tutte le funzionalità richieste dal proponente siano state soddisfatte. Per questo motivo, attraverso delle macro azioni, si andrà a simulare il comportamento generale dell'applicativo e dell'utente che interagisce con esso. I test di validazione saranno organizzati nel modo seguente:

$$TV[TipoRequisito][ImportanzaRequisito][IdRequisito]$$

dove:

- **TipoRequisito** può assumere valori tra:
 - F per i requisiti funzionali;
 - Q per i requisiti di qualità;
 - V per i requisiti di vincolo;
 - P per i requisiti prestazionali.
- **ImportanzaRequisito** può assumere valori tra:
 - D per i requisiti desiderabili;
 - O per i requisiti di obbligatori;
 - F per i requisiti di facoltativi.
- **IdRequisito** assume un valore gerarchico che identifica il singolo requisito.



Id Test	Descrizione	Stato
TVFO1	L'utente intende registrarsi al sistema. All'utente è richiesto di: <ul style="list-style-type: none">• Trovarsi nella sezione apposita;• Compilare il form di registrazione;• Premere il pulsante di conferma;• Verificare attraverso l'autenticazione che la registrazione sia effettivamente avvenuta.	<i>Non Implementato</i>
TVFO2	L'utente intende autenticarsi al sistema. All'utente è richiesto di: <ul style="list-style-type: none">• Trovarsi nella sezione apposita;• Inserire le credenziali nell'apposito form;• Premere il pulsante di autenticazione;• Verificare che l'autenticazione sia effettivamente avvenuta.	<i>Non Implementato</i>
TVFO3	L'utente intende disconnettersi dal sistema. All'utente è richiesto di: <ul style="list-style-type: none">• Essere Autenticato;• Trovarsi nella sezione apposita;• Premere il pulsante di logout;• Verificare che la disconnessione sia effettivamente avvenuta.	<i>Non Implementato</i>
TVFD4	L'utente autenticato intende gestire i propri dati. All'utente è richiesto di: <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Modificare i campi dati consentiti;• Premere il tasto conferma modifica;• Visualizzare il profilo dell'utente modificato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFD4.3	<p>L'utente autenticato intende impostare la propria foto profilo . All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Impostare la foto da inserire; • Visualizzare il profilo dell'utente modificato. 	<i>Non Implementato</i>
TVFD4.8	<p>L'utente autenticato intende modificare la tipologia di utenza. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Cambiare la tipologia di utenza; • Verificare la modifica effettuata. 	<i>Non Implementato</i>
TVFD4.9	<p>L'utente autenticato intende eliminare il proprio account. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Premere il tasto di eliminazione; • Verificare la disconnessione della sessione sia avvenuta; • Verificare che l'autenticazione con le credenziali dell'utente eliminato non avvenga. 	<i>Non Implementato</i>
TVFD5	<p>L'utente autenticato intende ricercare un questionario per iscriversi. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Ricercare un questionario; • Iscrizione ad un questionario trovato mediante l'apposito tasto; • Verificare l'operazione appena effettuata. 	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFO6	<p>L'utente autenticato intende compilare un questionario a cui si è iscritto. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Selezionare il questionario da compilare; • Rispondere alle domande previste; • Confermare le risposte date al questionario; • Verificare il risultato del questionario. 	<i>Non Implementato</i>
TVFD7.1	<p>L'utente autenticato intende creare una nuova domanda tramite procedura guidata. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Premere il pulsante Crea nuova domanda; • Inserire i dati necessari alla creazione di una nuova domanda; • Premere il pulsante di conferma creazione nuova domanda; • Verificare che sia stato creata la nuova domanda. 	<i>Non Implementato</i>
TVFD7.2	<p>L'utente autenticato intende modificare una domanda esistente tramite procedura guidata. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Premere il pulsante modifica domanda; • Modificare i dati della domanda selezionata; • Premere il pulsante di conferma modifica; • Verificare che sia stata modificata la domanda. 	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFO7.3	<p>L'utente autenticato intende creare una nuova domanda tramite QML. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Premere il pulsante Crea nuova domanda;• Inserire i dati necessari alla creazione di una nuova domanda;• Premere il pulsante di conferma creazione nuova domanda;• Verificare che sia stata creata la nuova domanda.	<i>Non Implementato</i>
TVFD7.4	<p>L'utente autenticato intende modificare una domanda esistente tramite QML. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Premere il pulsante modifica domanda;• Modificare i dati della domanda selezionata;• Premere il pulsante di conferma modifica;• Verificare che sia stata modificata la domanda.	<i>Non Implementato</i>
TVFD8.1	<p>L'utente autenticato intende visualizzare i questionari creati. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Verificare che vengano visualizzati tutti i questionari creati.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFD8.2	<p>L'utente autenticato pro intende modificare un questionario esistente. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Selezionare il questionario da modificare tra quelli creati; • Dalla sezione di modifica effettuare tutte le modifiche consentite dal sistema; • Premere il pulsante di conferma modifiche; • Verificare che sia stato modificato il questionario correttamente. 	<i>Non Implementato</i>
TVFD8.3	<p>L'utente autenticato pro intende eliminare un questionario esistente. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Selezionare un questionario esistente da eliminare; • Premere il tasto di conferma eliminazione; • Verificare l'operazione appena effettuata. 	<i>Non Implementato</i>
TVFD8.4	<p>L'utente autenticato pro intende controllare i risultati degli esaminandi di un suo questionario. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Selezionare il questionario da controllare gli esiti; • Verificare che si ci siano gli esiti di tutti gli utenti iscritti. 	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFD8.5	<p>L'utente autenticato pro intende attivare la modalità esame di un suo questionario. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Selezionare il questionario da attivare; • Verificare l'operazione effettuata. 	<i>Non Implementato</i>
TVFO8.6	<p>L'utente autenticato pro intende creare un nuovo questionario. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Premere il pulsante Crea nuovo questionario; • Inserire il titolo del nuovo questionario; • Inserire l'argomento del nuovo questionario; • Ricercare e selezionare le domande che andranno a comporre il questionario; • Premere il pulsante di conferma creazione nuovo questionario; • Verificare che sia stato creato il nuovo questionario. 	<i>Non Implementato</i>
TVFO8.7	<p>L'utente autenticato pro intende gestire le domande del questionario. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Premere il pulsante per gestire il questionario; • Effettuare le operazioni consentite; • Verificare che le operazioni siano effettivamente avvenute. 	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFD8.8	<p>L'utente autenticato pro intende gestire le iscrizioni di un suo questionario. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato pro; • Trovarsi nella sezione apposita; • Selezionare il questionario di cui gestire le iscrizioni; • Effettuare le operazioni consentite; • Verificare le operazioni apportate. 	<i>Non Implementato</i>
TVFO9	<p>L'utente intende esercitarsi effettuando un allenamento. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Trovarsi nella sezione apposita; • Selezionare gli appositi filtri per focalizzare l'allenamento che si vuole effettuare; • Premere il pulsante per iniziare l'allenamento. • Rispondere alle domande proposte e controllare l'esito delle risposte date. 	<i>Non Implementato</i>
TVFD10	<p>L'utente autenticato intende visualizzare il proprio profilo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Visualizzare il proprio profilo. 	<i>Non Implementato</i>
TVFD10.4	<p>L'utente autenticato intende visualizzare la cronologia dei questionari svolti. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Visualizzare la cronologia dei questionari svolti. 	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TVFD10.4.1	L'utente autenticato intende visualizzare il riepilogo di un questionario svolto. All'utente è richiesto di: <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Selezionare il questionario svolto da visualizzare il riepilogo.	<i>Non Implementato</i>
TVFO11	L'utente intende rispondere alle domande. All'utente è richiesto di: <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Rispondere alle domande;• Verificare la funzionalità dell'operazione.	<i>Non Implementato</i>
TVFO12	L'utente autenticato intende ricercare un utente. All'utente è richiesto di: <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Inserire le keywords nella barra di ricerca;• Premere l'apposito tasto per effettuare la ricerca;• Verificare l'operazione appena effettuata.	<i>Non Implementato</i>

Tabella 1: Test di Validazione



4.1.2 Test di Sistema

Con questa tipologia di test si vuole verificare il corretto funzionamento delle componenti architetture.

I test di sistema saranno descritti nel modo seguente:

$$TS[TipoRequisito][ImportanzaRequisito][IdRequisito]$$

dove:

- **TipoRequisito** può assumere valori tra:
 - *F* per i requisiti funzionali;
 - *Q* per i requisiti di qualità;
 - *V* per i requisiti di vincolo;
 - *P* per i requisiti prestazionali.
- **ImportanzaRequisito** può assumere valori tra:
 - *D* per i requisiti desiderabili;
 - *O* per i requisiti di obbligatori;
 - *F* per i requisiti di facoltativi.
- **IdRequisito** può assumere un valore gerarchico che identifica il singolo requisito.

Id Test	Descrizione	Stato
TSFO1	Viene verificato che il sistema registri correttamente un utente.	<i>Non Implementato</i>
TSFO2	Viene verificato che il sistema riesca a far effettuare correttamente il login ad un utente attraverso QuizziPedia.	<i>Non Implementato</i>
TSFO3	Viene verificato che il sistema riesca a far effettuare correttamente il logout ad un utente.	<i>Non Implementato</i>
TSFD4	Viene verificato che il sistema permetta di gestire correttamente ad un utente le informazioni del proprio profilo.	<i>Non Implementato</i>
TSFD4.6	Viene verificato che il sistema modifichi correttamente le informazioni di un utente.	<i>Non Implementato</i>
TSFD4.8	Viene verificato che il sistema cambi correttamente la tipologia di account di un utente.	<i>Non Implementato</i>
TSFD4.9	Viene verificato che il sistema elimini correttamente un account.	<i>Non Implementato</i>
TSFD5	Viene verificato che il sistema ricerchi correttamente un questionario.	<i>Non Implementato</i>
TSFO6	Viene verificato che il sistema permetta la compilazione corretta di un questionario da parte di un utente.	<i>Non Implementato</i>
TSFD7.1	Viene verificato che il sistema crei correttamente una domanda attraverso la procedura guidata.	<i>Non Implementato</i>
TSFD7.2	Viene verificato che il sistema modifichi correttamente una domanda attraverso la procedura guidata.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TSFO7.3	Viene verificato che il sistema crei correttamente una domanda attraverso l'editor QML.	<i>Non Implementato</i>
TSFD7.4	Viene verificato che il sistema modifichi correttamente una domanda attraverso l'editor QML.	<i>Non Implementato</i>
TSFD8.1	Viene verificato che il sistema visualizzi correttamente i questionari creati da un utente.	<i>Non Implementato</i>
TSFD8.2	Viene verificato che il sistema modifichi correttamente un questionario.	<i>Non Implementato</i>
TSFD8.3	Viene verificato che il sistema elimini correttamente un questionario.	<i>Non Implementato</i>
TSFD8.4	Viene verificato che il sistema visualizzi correttamente i risultati degli esaminandi.	<i>Non Implementato</i>
TSFD8.5	Viene verificato che il sistema attivi correttamente la modalità esame.	<i>Non Implementato</i>
TSFO8.6	Viene verificato che il sistema crei correttamente un questionario.	<i>Non Implementato</i>
TSFD8.6.4.1	Viene verificato che il sistema generi correttamente un riepilogo e lo renda visualizzabile dall'utente al termine della compilazione di un questionario.	<i>Non Implementato</i>
TSFO8.7.1	Viene verificato che il sistema inserisca correttamente le domande all'interno di un questionario.	<i>Non Implementato</i>
TSFO8.7.1.1	Viene verificato che il sistema ricerchi correttamente una domanda all'interno del database delle domande.	<i>Non Implementato</i>
TSFO8.7.2	Viene verificato che il sistema rimuova correttamente una domanda da un questionario.	<i>Non Implementato</i>
TSFD8.8	Viene verificato che il sistema gestisca correttamente le iscrizioni ad un questionario.	<i>Non Implementato</i>
TSFD9.4	Viene verificato che il sistema avvii correttamente un allenamento.	<i>Non Implementato</i>
TSFD10	Viene verificato che il sistema visualizzi correttamente il profilo di un utente.	<i>Non Implementato</i>
TSFD10.4	Viene verificato che il sistema visualizzi correttamente la cronologia di tutti i questionari svolti da un utente.	<i>Non Implementato</i>
TSFD10.4.1	Viene verificato che il sistema visualizzi correttamente le statistiche di un questionario scelto dalla cronologia dei questionari svolti da un utente.	<i>Non Implementato</i>
TSFD10.5	Viene verificato che il sistema visualizzi correttamente la lista dei questionari abilitati per un utente.	<i>Non Implementato</i>
TSFO11	Viene verificato che il sistema visualizzi e permetta di rispondere ad una domanda da parte di un utente.	<i>Non Implementato</i>
TSFO12	Viene verificato che il sistema ricerchi correttamente un utente.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TSFD18	Viene verificato che il sistema iscriva correttamente un utente ad un questionario.	<i>Non Implementato</i>
TSFF19	Viene verificato che il sistema recuperi correttamente la password dimenticata da un utente.	<i>Non Implementato</i>
TSFO20	Viene verificato che il sistema proponga dei questionari da sottoporre all'utente.	<i>Non Implementato</i>
TSFO21	Viene verificato che il sistema proponga dei questionari specifici per l'argomento scelto.	<i>Non Implementato</i>
TSFO22	Viene verificato che il sistema generi dei questionari a partire dalle domande archiviate.	<i>Non Implementato</i>
TSFO23	Viene verificato che il sistema archivi correttamente le domande attraverso uno specifico linguaggio chiamato QML_G .	<i>Non Implementato</i>
TSFO23.8	Viene verificato che il sistema archivi le domande suddivise per argomento.	<i>Non Implementato</i>
TSFO24	Viene verificato che il sistema archivi i questionari creati con le domande.	<i>Non Implementato</i>
TSFO25	Viene verificato che il sistema proponga all'utente dei questionari preconfezionati.	<i>Non Implementato</i>
TSFO26	Viene verificato che il sistema permetta di creare domande e questionari da dispositivi desktop.	<i>Non Implementato</i>
TSFO27	Viene verificato che il sistema permetta di rispondere alle domande e ai questionari da dispositivi desktop, tablet e mobile.	<i>Non Implementato</i>
TSFD28	Viene verificato che il sistema archivi correttamente i risultati dei questionari.	<i>Non Implementato</i>
TSFD29	Viene verificato che il sistema archivi le statistiche delle risposte date ad ogni domanda.	<i>Non Implementato</i>
TSFF30	Viene verificato che il sistema valuti il candidato rispetto agli altri candidati che hanno svolto lo stesso quiz.	<i>Non Implementato</i>
TSFD31	Viene verificato che il sistema crei questionari dinamicamente (modalità allenamento) per un argomento scegliendo le domande in modo casuale.	<i>Non Implementato</i>
TSFD32	Viene verificato che il sistema crei questionari dinamicamente (modalità allenamento) scegliendo le domande in base alle risposte date dai questionari precedenti.	<i>Non Implementato</i>
TSFD33	Viene verificato che il sistema crei questionari dinamicamente (modalità allenamento) scegliendo tra le domande più difficili.	<i>Non Implementato</i>
TSFD34	Viene verificato che il sistema crei questionari dinamicamente (modalità allenamento) scegliendo tra le lacune dei partecipanti.	<i>Non Implementato</i>
TSFD35	Viene verificato che il sistema permetta agli utilizzatori di proporre nuove domande.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TSFF36	Viene verificato che il sistema permetta agli utilizzatori di segnalare positivamente una domanda.	<i>Non Implementato</i>
TSFF37	Viene verificato che il sistema permetta agli utilizzatori di commentare una domanda.	<i>Non Implementato</i>
TSFF38	Viene verificato che il sistema permetta agli utilizzatori di rispondere più volte ad una domanda.	<i>Non Implementato</i>
TSVO1	Viene verificato che il sistema utilizzi il linguaggio <i>javascript_G</i> .	<i>Non Implementato</i>
TSVO2	Viene verificato che il sistema utilizzi il <i>linguaggio di markup_G HTML5_G</i> .	<i>Non Implementato</i>
TSVO3	Viene verificato che il sistema utilizzi <i>fogli di stile_G</i> in <i>CSS/ped_G</i> .	<i>Non Implementato</i>
TSVO5	Viene verificato che il sistema funzioni su <i>Mozilla Firefox_G</i> versione 33.0 o superiore	<i>Non Implementato</i>
TSVO6	Viene verificato che il sistema funzioni su <i>Google Chrome_G</i> versione 31.0 o superiore	<i>Non Implementato</i>
TSVO7	Viene verificato che il sistema funzioni su <i>Safari_G</i> versione 7.1 o superiore	<i>Non Implementato</i>
TSVO8	Viene verificato che il sistema funzioni su <i>Opera</i> versione 26.0 o superiore	<i>Non Implementato</i>
TSVD9	Viene verificato che il sistema funzioni su <i>Internet Explorer_G</i> versione 11.0 o superiore	<i>Non Implementato</i>
TSVO10	Viene verificato che il sistema funzioni su <i>Microsoft Edge_G</i> versione 25 o superiore	<i>Non Implementato</i>
TSVD11	Viene verificato che il sistema funzioni su <i>Android browser_G</i> versione 4.4 o superiore per le funzionalità che riguardano la compilazione dei questionari e delle domande	<i>Non Implementato</i>
TSVO12	Viene verificato che il sistema funzioni su <i>Safari per iOS 8_G</i> o versioni superiori per le funzionalità che riguardano la compilazione dei questionari e delle domande	<i>Non Implementato</i>
TSVD13	Viene verificato che il sistema funzioni su <i>Google Chrome per iOS_G</i> versione 39 o superiore per le funzionalità che riguardano la compilazione dei questionari e delle domande	<i>Non Implementato</i>
TSVO14	Viene verificato che il sistema funzioni su <i>Google Chrome per Android_G</i> versione 39 o superiore per le funzionalità che riguardano la compilazione dei questionari e delle domande	<i>Non Implementato</i>
TSVF15	Viene verificato che il sistema funzioni su <i>Mozilla Firefox per Android_G</i> versione 33 o superiore per le funzionalità che riguardano la compilazione dei questionari e delle domande	<i>Non Implementato</i>
TSVF16	Viene verificato che il sistema funzioni su <i>Microsoft Edge per Windows 10 mobile_G</i> versione 25 o superiore per le funzionalità che riguardano la compilazione dei questionari e delle domande	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TSVF17	Viene verificato che il sistema funzioni su <i>Opera Mini per iOS_G</i> versione 12 o superiore per le funzionalità che riguardano la compilazione dei questionari e delle domande.	<i>Non Implementato</i>
TSVD18	Viene verificato che il sistema funzioni su <i>Google Chrome per Android_G</i> versione 39 o superiore per le funzionalità che riguardano la creazione dei questionari e delle domande.	<i>Non Implementato</i>
TSVF19	Viene verificato che il sistema funzioni su <i>Mozilla Firefox per Android_G</i> versione 33 o superiore per le funzionalità che riguardano la creazione dei questionari e delle domande.	<i>Non Implementato</i>
TSVF20	Viene verificato che il sistema funzioni su <i>Microsoft Edge per Windows 10 mobile_G</i> versione 25 o superiore per le funzionalità che riguardano la creazione dei questionari e delle domande.	<i>Non Implementato</i>
TSVF21	Viene verificato che il sistema funzioni su <i>Browser Opera per Android_G</i> versione 34 o superiore per le funzionalità che riguardano la creazione dei questionari e delle domande.	<i>Non Implementato</i>
TSVF22	Viene verificato che il sistema funzioni su <i>Opera Mini per iOS_G</i> versione 12 o superiore per le funzionalità che riguardano la creazione dei questionari e delle domande.	<i>Non Implementato</i>

Tabella 2: Test di Sistema



4.1.3 Test di Integrazione

Con questa tipologia di test si vuole determinare il corretto funzionamento delle componenti progettate durante la definizione dell'architettura ad alto livello.

I test di integrazione saranno descritti nel modo seguente:

$$TI[IdComponente]$$

dove:

- **IdComponente** rappresenta il codice identificativo crescente del componente considerato.

È stato scelto di utilizzare un approccio top-down nel determinare i test di integrazione. Di seguito viene riportato un diagramma informale per rendere chiara la struttura dei test identificati.

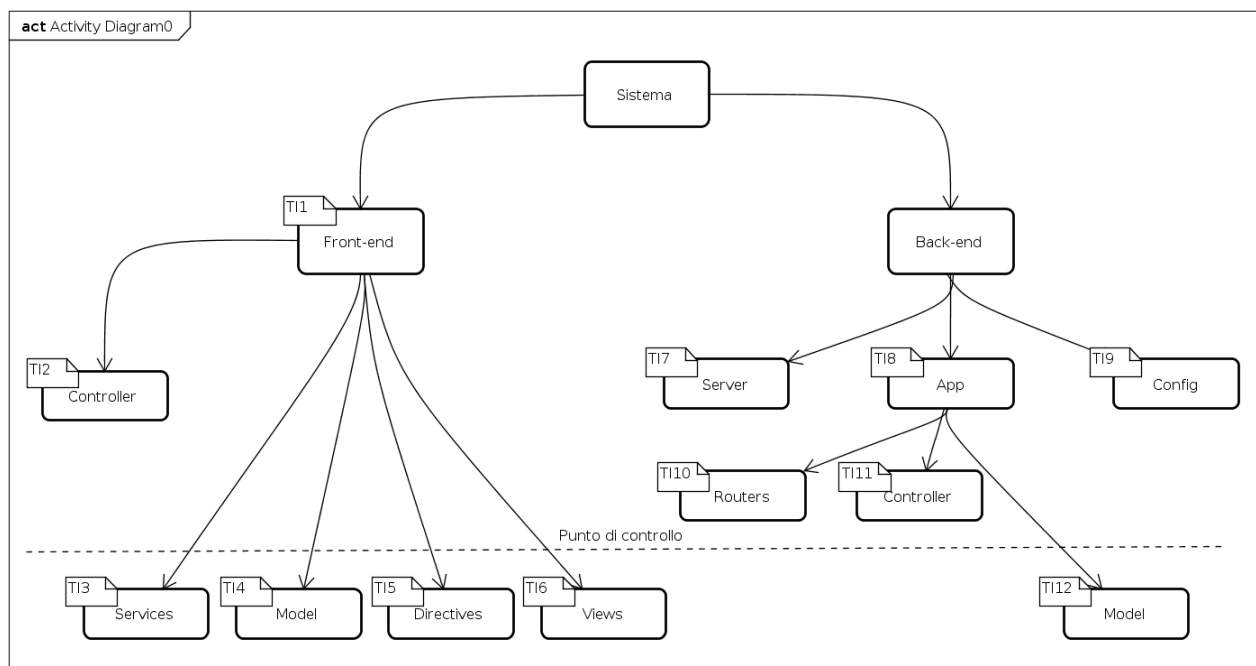


Figura 1: Albero di integrazione delle componenti

Nell'approccio top-down dei test di integrazione i moduli di livello più alto vengono sottoposti a test e integrati per primi. Così facendo anche la logica di alto livello e il flusso di dati vengono sottoposti a test fin da subito; sarà perciò necessario simulare le componenti di livello più basso con degli stub. Una volta codificate, le componenti di più basso livello dovranno a loro volta essere integrate e testate. L'approccio top-down rientra tra le strategie di integrazione incrementali, che conferiscono il vantaggio di poter determinare in modo più immediato quale componente causa problemi: i difetti rilevati dai test, infatti, nella maggioranza dei casi saranno da attribuirsi all'ultima componente aggiunta.

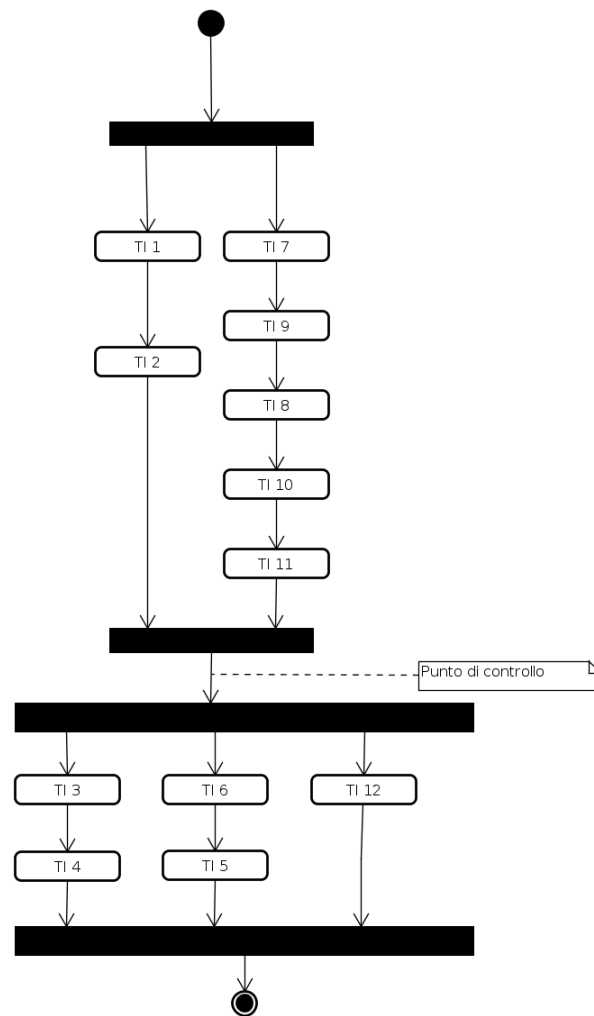


Figura 2: Diagramma di attività dei test di integrazione

Id Test	Descrizione	Stato
TI1	Viene verificato che l'applicazione Web gestisca correttamente il Front-End del prodotto e le sue interazioni con il Back-End.	<i>Non Implementato</i>
TI2	Viene verificato che i Controllers del Front-End si integrino correttamente nell'applicazione Web.	<i>Non Implementato</i>
TI3	Viene verificato che i Services permettano di interagire correttamente con il Back-End.	<i>Non Implementato</i>
TI4	Viene verificato che il Model si integri correttamente con i Services e con le componenti dell'applicazione che lo utilizzano.	<i>Non Implementato</i>
TI5	Viene verificato che le Directives si integrino correttamente con le Views.	<i>Non Implementato</i>
TI6	Viene verificato che le Views si integrino correttamente con i Controllers e che visualizzino in modo corretto i dati da essi ricevuti.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TI7	Viene verificato che il server si avvii correttamente, utilizzando Config per effettuare le configurazioni dell'applicazione, e che l'applicazione Web gestisca correttamente il Back-End del prodotto in modo tale da fornire al Front-End tutte le informazioni richieste.	<i>Non Implementato</i>
TI8	Viene verificato che App si integri correttamente con le librerie di Node.js utilizzate.	<i>Non Implementato</i>
TI9	Viene verificato che Config si integri con Server, carichi correttamente tutte le librerie per Node.js che utilizzerà e che istanzi le classi del package App in modo corretto.	<i>Non Implementato</i>
TI10	Viene verificato che Config si integri con Server, carichi correttamente tutte le librerie per Node.js che utilizzerà e che istanzi le classi del package App in modo corretto.	<i>Non Implementato</i>
TI11	Viene verificato che i Controllers si integrino correttamente e gestiscano le richieste inoltrate dai Routers.	<i>Non Implementato</i>
TI12	Viene verificato che il Model si integri correttamente con i Controllers per la gestione dell'inserimento, della modifica e dell'eliminazione dei dati.	<i>Non Implementato</i>

Tabella 3: Test di Integrazione



4.1.4 Test di Unità

Con questa tipologia di test si vuole verificare il corretto funzionamento delle unità individuate durante la definizione dell'architettura ad alto livello. I test di unità saranno descritti nel modo seguente:

$$\mathbf{TU}[IdTest]$$

dove:

- **IdTest** rappresenta il codice identificativo crescente dell'unità considerata.

Id Test	Descrizione	Stato
TU1	Verificare che l'istanza del server venga creata correttamente e che si metta in ascolto su localhost:3333 o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU2	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al concreteHandler corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU3	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al concreteHandler corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU4	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al concreteHandler corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU5	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al concreteHandler corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU6	Verificare che, in base ai parametri forniti in input, il JSON ritornato contenga il messaggio d'errore corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU7	Verificare che, in base ai parametri forniti in input, il JSON ritornato contenga il messaggio d'errore corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU8	Verificare che il messaggio d'errore venga costruito in modo coerente rispetto ai dati passati al costruttore e che i metodi disponibili restituiscano l'informazione nel formato desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU9	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti registrazione ed autenticazione dell'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU10	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, verificando che l'utente che esegue la richiesta sia effettivamente un utente autenticato e mantenendo aggiornate le informazioni riguardanti l'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU11	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati i dati dell'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU12	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati le statistiche relative all'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU13	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornata la cronologia dei questionari svolti o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU14	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati la gestione dei topic o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU15	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le statistiche delle domande uscite o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU16	Verificare che, in base ai parametri forniti in input, il metodo crei il riepilogo del questionario svolto dall'utente, restituisca il relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU17	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, verificando che il questionario creato dall'utente sia effettivamente creato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU18	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti i questionari creati dall'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU19	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti le domande create dall'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU20	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, verificando che una domanda creata dall'utente sia effettivamente creata o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU21	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, verificando che le informazioni ottenute per effettuare una traduzione siano corrette o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU22	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni dell'utente in modo corretto, creando un nuovo utente, aggiornandone il contenuto e restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU23	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, verificando la password inserita in fase di autenticazione e restituendone un messaggio di conferma o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU24	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni di una domanda in modo corretto, creando una nuova domanda, aggiornandone il contenuto e restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU25	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni di un quiz in modo corretto, creando un nuovo quiz, aggiornandone il contenuto e restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU26	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni dell'argomento in modo corretto, e restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU27	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni di un questionario svolto in modo corretto, creando il riepilogo di un questionario svolto e restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU28	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni delle traduzioni in modo corretto, restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU29	Verificare che il metodo effettui le operazioni richieste, costruendo l'oggetto corrispondente e, per mezzo di \$routeProvider, associando ad ogni route un controller ed una view o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU30	Verificare che il metodo effettui le operazioni richieste, costruendo l'oggetto corrispondente, il quale funge da gestore per gli eventi globali o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU31	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo i dettagli di un questionario richiamando i metodi del relativo Service. Nello specifico restituendo i dettagli dei questionari creati da un utente e i questionari a cui un utente è iscritto o che, nel caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU32	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo tutti i questionari creati da un utente richiamando i metodi del relativo Service. Nello specifico restituendo tutti i questionari creati da un utente o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU33	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le risposte inserite dall'utente, salvando la domanda inserita dall'utente per poi visualizzarla nella pagina o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU34	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo l'autenticazione al sistema, in particolare la richiesta di autenticazione, la richiesta di registrazione e la possibilità di recuperare la password dimenticata o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU35	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di una domanda ad area cliccabile; in specifico venga controllata la costruzione della relativa domanda e gli eventi che generano la creazione/modifica della domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU36	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di una domanda a collegamento; in specifico venga controllata la costruzione della relativa domanda e gli eventi che generano la creazione/modifica della domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU37	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il menù fisso per ogni pagina; in particolare deve gestire il logout dal sistema, il login dal sistema, la registrazione al sistema, la possibilità di effettuare il redirect alla pagina di visualizzazione utente, di gestione profilo utente, di gestione delle domande, di gestione dei questionari e infine gestire la visualizzazione dei giusti collegamenti della barra per ogni pagina o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU38	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la creazione e modifica di domande a risposta multipla, in particolare deve gestire la chiamata al service per l'invio della nuova domanda o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU39	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il redirect alla pagina di creazione di una nuova domanda o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU40	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione di un nuovo questionario e la modifica di uno esistente; in specifico venga controllata la costruzione del questionario, le operazioni che permettano la conferma creazione/modifica/eliminazione del questionario o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU41	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il recupero della password dimenticata, in particolare la comunicazione col service per l'invio della mail e la gestione dell'evento redirect alla pagina di login o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU42	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il recupero delle domande per far sì che possano essere visualizzate nella modalità allenamento e nella compilazione di questionari, richiamando i metodi del relativo service. Nello specifico gestendo tutti gli eventi che hanno a che fare con la selezione e la visualizzazione di una nuova domanda o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU43	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il profilo personale dell'utente, in particolare l'invio delle nuove informazioni al service tramite l'apposito metodo.	<i>Non Implementato</i>
TU44	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la registrazione di un utente al sistema; in particolare deve richiamare il metodo signup del service AuthService passandogli un oggetto di tipo SignUpModelView; nel caso di buona riuscita deve essere mostrato un messaggio di successo e gestito il redirect alla pagina di login mentre nel caso contrario deve essere mostrato un messaggio di errore.	<i>Non Implementato</i>
TU45	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di domande create tramite editor QML; in specifico venga controllata la costruzione della domanda e le operazioni che permettano la conferma creazione/modifica della domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU46	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo tutte le domande create da un utente e creandone di nuove richiamando i metodi del relativo service. Nello specifico recuperando tutte le domande create da un utente e gestendo l'evento click sul pulsante di modifica di una domanda o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU47	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la compilazione del questionario; in specifico venga controllata la visualizzazione del questionario e le operazioni che permettano di rispondere alle varie domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU48	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, reagendo ai comandi dell'utente durante la gestione dei suoi questionari richiamando i metodi del relativo service. Nello specifico gestendo l'evento click di modifica/eliminazione/gestione iscrizioni/attivazione modalità esame/allenamento di un questionario o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU49	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le iscrizioni degli utenti ai questionari richiamando i metodi del relativo service. Nello specifico permettendo di iscriversi ad un questionario o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU50	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la visualizzazione dei risultati di un singolo questionario richiamando i metodi del relativo service. Nello specifico restituendo il risultato di un questionario, restituendo tutti gli utenti che hanno eseguito un questionario ed il loro risultato o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU51	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le statistiche di un utente; in particolare deve ottenere tali statistiche attraverso il StatisticsService, passandogli l'username dell'utente ricercato o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU52	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di una domanda a riempimento di spazi; in specifico venga controllata la costruzione della domanda e le operazioni che permettano la creazione/modifica della domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU53	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo la visualizzazione della homepage; in specifico venga controllata che siano presenti all'interno della home tutte le features previste o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU54	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di una domanda a ordinamento immagini; in specifico venga controllata la costruzione della domanda e le operazioni che permettano la creazione/modifica della domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU55	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di una domanda a ordinamento di stringhe; in particolare deve gestire l'evento click sul pulsante di conferma sulla domanda, raccogliendo i dati di questa dal modelview e mandandoli al server attraverso il Question-Service; dovrà gestire infine il giusto redirect alla pagina di gestione delle domande oppure al questionario che si stava creando o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU56	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la ricerca di questionari e utenti all'interno dell'applicazione richiamando i metodi del relativo service. Nello specifico gestendo l'evento click sui bottoni per visualizzare l'utente selezionato, per registrarsi ad un questionario, per effettuare una ricerca o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU57	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il recupero delle parole chiave di un questionario; in particolare data una stringa di ricerca deve restituire la parole chiave associata ad essa o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU58	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la modalità allenamento sottoponendo all'utente le giuste domande adatte al suo livello; in particolare deve gestire l'evento per inserire una domanda nella cronologia delle domande, gestire l'evento per scaricare una nuova domanda in base ai parametri passati facendo una richiesta al QuestionsController che andrà a scaricare la domanda e ad inserirla in TrainingMode-Model nello \$scope mediante il metodo addQuestion, gestire l'evento per inserire il risultato di una domanda e gestire l'evento per iniziare l'allenamento o, nel caso di un errore, deve garantire che la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU59	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire la creazione e la modifica di una domanda vero e falso; in specifico venga controllata la costruzione della domanda e le operazioni che permettano la creazione/modifica della domanda o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU60	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo i dati di un utente da mostrare nella pagina di un profilo; in particolare deve i dati con una chiamata a UserDetailsService, passandogli l'username dell'utente ricercato o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU61	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo i dati di un questionario; in particolare i metodi devono permettere di settare il titolo del questionario, settare l'autore del questionario, ottenere il titolo del questionario, ottenere l'autore del questionario, ottenere la lista di tutte le domande del questionario, aggiungere una domanda al questionario e infine rimuovere una domanda dal questionario o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU62	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, memorizzando i dati relativi ad un utente e gestendoli richiamando i relativi Controller. Nello specifico gestendo l'autenticazione dell'utente al sistema, la ricerca di questionari e utenti all'interno dell'applicazione, i dati e le statistiche riguardanti un utente o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU63	Verificare che i metodi getter effettuino le operazioni richieste, restituendo i dati necessari per la creazione dinamica della barra menù posizionata in modo fisso su ogni pagina; o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU64	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto contenente le informazioni sull'errore, ritornando correttamente il titolo, il messaggio ed il codice dell'errore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU65	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo i dati di una domanda; in particolare i metodi devono permettere di settare l'autore, settare il metodo con il quale è stata creata, settare la lingua, settare tutte le parti della domanda, settare l'id della domanda con quello ottenuto automaticamente, ottenere l'autore, ottenere il metodo con il quale è stata creata, ottenere la lingua, ottenere l'oggetto domanda, ottenere l'id della domanda, comparare le risposte date con quelle corrette, aggiungere un pezzo nuovo di domanda e creare una domanda o, nel caso di un errore, deve garantire la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU66	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, memorizzando i dati di un allenamento e gestendoli richiamando il relativo Controller. Nello specifico settando/recuperando argomento, numero di domande dell'allenamento, inserendo/rimuovendo una domanda, aggiungendo/rimuovendo una parola chiave, recuperando il riepilogo dell'allenamento o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU67	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto contenente riguardanti la giusta traduzione dell'applicazione, ritornando correttamente la lingua, la lista di keywords o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU68	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto contenente riguardanti la giusta traduzione dell'applicazione, impostando correttamente la nuova lingua inserita o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU69	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le domande, ossia il loro recupero e salvataggio; in particolare i metodi devono permettere di inviare una richiesta di salvataggio di una domanda, ottenere tutte le domande di un utente, ottenere le keywords a partire da una stringa in input, ritornare una singola domanda avendo come input l'id e infine ottenere la domanda successiva per la modalità allenamento o, nel caso di un errore, deve garantire che la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU70	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la registrazione e l'autenticazione di un utente fornendone le funzionalità ai Controller. Nello specifico permettendo di controllare se un utente è già autenticato, di effettuare il login, il logout, la registrazione e il recupero della password o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>



Id Test	Descrizione	Stato
TU71	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la lingua in cui un utente ha deciso di utilizzare un'applicazione e fornendo le funzionalità per recuperare la giusta traduzione delle pagine. Nello specifico recuperando la lista di tutte le keywords nella lingua richiesta o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU72	Verificare che, in base ai parametri forniti in input, utilizzando il service SearchService, i metodi richiedano in maniera corretta le informazioni inerenti alla ricerca degli utenti e dei questionari, che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU73	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo tutto ciò che riguarda i questionari; in particolare i metodi devono permettere di cercare un questionario, iscrivere un utente ad un questionario, ottenere tutti i dettagli di un questionario, ottenere tutti i questionari eseguiti da un utente, creare un nuovo questionario, ottenere tutti i questionari creati da un utente, ottenere tutti i questionari a cui un utente è iscritto, approvare l'iscrizione di un utente ad un questionario, ottenere tutti gli utenti che hanno eseguito un particolare questionario, ottenere i risultati di un questionario e infine ottenere i risultati di un singolo utente ad un questionario o, nel caso di un errore, deve garantire che la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU74	Verificare che, in base ai parametri forniti in input, utilizzando il service StatisticsService, i metodi richiedano in maniera corretta le informazioni inerenti alle statistiche degli utenti e dei questionari, che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU75	Verificare che, in base ai parametri forniti in input, utilizzando il service UserDetailsService, i metodi richiedano in maniera corretta l'ottenimento e il salvataggio dei dati dell'utente, che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Tabella 4: Test di Unità



5 Tracciamento dei test

5.1 Tracciamento Test di Validazione-Requisiti

Test	Requisito
TVFO1	RFO1
TVFO2	RFO2
TVFO3	RFO3
TVFD4	RFD4
TVFD4.3	RFD4.3
TVFD4.8	RFD4.8
TVFD4.9	RFD4.9
TVFD5	RFD5
TVFO6	RFO6
TVFD7.1	RFD7.1
TVFD7.2	RFD7.2
TVFO7.3	RFO7.3
TVFD7.4	RFD7.4
TVFD8.1	RFD8.1
TVFD8.2	RFD8.2
TVFD8.3	RFD8.3
TVFD8.4	RFD8.4
TVFD8.5	RFD8.5
TVFO8.6	RFO8.6
TVFO8.7	RFO8.7
TVFD8.8	RFD8.8
TVFO9	RFO9
TVFD10	RFD10
TVFD10.4	RFD10.4
TVFD10.4.1	RFD10.4.1
TVFO11	RFO11
TVFO12	RFO12

Tabella 5: Tracciamento Test di Validazione-Requisiti



5.2 Tracciamento Requisiti-Test di Validazione

Requisito	Test
RFO1	TVFO1
RFO2	TVFO2
RFO3	TVFO3
RFD4	TVFD4
RFD4.3	TVFD4.3
RFD4.8	TVFD4.8
RFD4.9	TVFD4.9
RFD5	TVFD5
RFO6	TVFO6
RFD7.1	TVFD7.1
RFD7.2	TVFD7.2
RFO7.3	TVFO7.3
RFD7.4	TVFD7.4
RFD8.1	TVFD8.1
RFD8.2	TVFD8.2
RFD8.3	TVFD8.3
RFD8.4	TVFD8.4
RFD8.5	TVFD8.5
RFO8.6	TVFO8.6
RFO8.7	TVFO8.7
RFD8.8	TVFD8.8
RFO9	TVFO9
RFD10	TVFD10
RFD10.4	TVFD10.4
RFD10.4.1	TVFD10.4.1
RFO11	TVFO11
RFO12	TVFO12

Tabella 6: Tracciamento Requisiti-Test di Validazione



5.3 Tracciamento Test di Sistema-Requisiti

Test	Requisito
TSFO1	RFO1
TSFO2	RFO2
TSFO3	RFO3
TSFD4	RFD4
TSFD4.6	RFD4.6
TSFD4.8	RFD4.8
TSFD4.9	RFD4.9
TSFD5	RFD5
TSFO6	RFO6
TSFD7.1	RFD7.1
TSFD7.2	RFD7.2
TSFO7.3	RFO7.3
TSFD7.4	RFD7.4
TSFD8.1	RFD8.1
TSFD8.2	RFD8.2
TSFD8.3	RFD8.3
TSFD8.4	RFD8.4
TSFD8.5	RFD8.5
TSFO8.6	RFO8.6
TSFD8.6.4.1	RFD8.6.4.1
TSFO8.7.1	RFO8.7.1
TSFO8.7.1.1	RFO8.7.1.1
TSFO8.7.2	RFO8.7.2
TSFD8.8	RFD8.8
TSFD9.4	RFD9.4
TSFD10	RFD10
TSFD10.4	RFD10.4
TSFD10.4.1	RFD10.4.1
TSFD10.5	RFD10.5
TSFO11	RFO11
TSFO12	RFO12
TSFD18	RFD18
TSFF19	RFF19
TSFO20	RFO20
TSFO21	RFO21
TSFO22	RFO22
TSFO23	RFO23
TSFO23.8	RFO23.8
TSFO24	RFO24
TSFO25	RFO25
TSFO26	RFO26
TSFO27	RFO27
TSFD28	RFD28
TSFD29	RFD29
TSFF30	RFF30
TSFD31	RFD31



Test	Requisito
TSFD32	RFD32
TSFD33	RFD33
TSFD34	RFD34
TSFD35	RFD35
TSFF36	RFF36
TSFF37	RFF37
TSFF38	RFF38
TSVO1	RVO1
TSVO2	RVO2
TSVO3	RVO3
TSVO5	RVO5
TSVO6	RVO6
TSVO7	RVO7
TSVO8	RVO8
TSVD9	RVD9
TSVO10	RVO10
TSVD11	RVD11
TSVO12	RVO12
TSVD13	RVD13
TSVO14	RVO14
TSVF15	RVF15
TSVF16	RVF16
TSVF17	RVF17
TSVD18	RVD18
TSVF19	RVF19
TSVF20	RVF20
TSVF21	RVF21
TSVF22	RVF22

Tabella 7: Tracciamento Test di Sistema-Requisiti



5.4 Tracciamento Requisiti-Test di Sistema

Requisito	Test
RFO1	TSFO1
RFO2	TSFO2
RFO3	TSFO3
RFD4	TSFD4
RFD4.6	TSFD4.6
RFD4.8	TSFD4.8
RFD4.9	TSFD4.9
RFD5	TSFD5
RFO6	TSFO6
RFD7.1	TSFD7.1
RFD7.2	TSFD7.2
RFO7.3	TSFO7.3
RFD7.4	TSFD7.4
RFD8.1	TSFD8.1
RFD8.2	TSFD8.2
RFD8.3	TSFD8.3
RFD8.4	TSFD8.4
RFD8.5	TSFD8.5
RFO8.6	TSFO8.6
RFD8.6.4.1	TSFD8.6.4.1
RFO8.7.1	TSFO8.7.1
RFO8.7.1.1	TSFO8.7.1.1
RFO8.7.2	TSFO8.7.2
RFD8.8	TSFD8.8
RFD9.4	TSFD9.4
RFD10	TSFD10
RFD10.4	TSFD10.4
RFD10.4.1	TSFD10.4.1
RFD10.5	TSFD10.5
RFO11	TSFO11
RFO12	TSFO12
RFD18	TSFD18
RFF19	TSFF19
RFO20	TSFO20
RFO21	TSFO21
RFO22	TSFO22
RFO23	TSFO23
RFO23.8	TSFO23.8
RFO24	TSFO24
RFO25	TSFO25
RFO26	TSFO26
RFO27	TSFO27
RFD28	TSFD28
RFD29	TSFD29
RFF30	TSFF30
RFD31	TSFD31



Requisito	Test
RFD32	TSFD32
RFD33	TSFD33
RFD34	TSFD34
RFD35	TSFD35
RFF36	TSFF36
RFF37	TSFF37
RFF38	TSFF38
RVO1	TSVO1
RVO2	TSVO2
RVO3	TSVO3
RVO5	TSVO5
RVO6	TSVO6
RVO7	TSVO7
RVO8	TSVO8
RVD9	TSVD9
RVO10	TSVO10
RVD11	TSVD11
RVO12	TSVO12
RVD13	TSVD13
RVO14	TSVO14
RVF15	TSVF15
RVF16	TSVF16
RVF17	TSVF17
RVD18	TSVD18
RVF19	TSVF19
RVF20	TSVF20
RVF21	TSVF21
RVF22	TSVF22

Tabella 8: Tracciamento Requisiti-Test di Sistema



5.5 Tracciamento Test di Integrazione-Componenti

Test	Componente
TI1	Quizzipedia::Front-End
TI2	Quizzipedia::Front-End::Controllers
TI3	Quizzipedia::Front-End::Services
TI4	Quizzipedia::Front-End::Model
TI5	Quizzipedia::Front-End::Directives
TI6	Quizzipedia::Front-End::Views
TI7	Quizzipedia::Back-End
TI8	Quizzipedia::Back-End::App
TI9	Quizzipedia::Back-End::Config
TI10	Quizzipedia::Back-End::App::Routers
TI11	Quizzipedia::Back-End::App::Controller
TI12	Quizzipedia::Back-End::App::Model

Tabella 9: Tracciamento Test di Integrazione-Componenti



5.6 Tracciamento Componenti-Test di Integrazione

Componente	Test
Quizzipedia::Back-End	TI7
Quizzipedia::Back-End::App	TI8
Quizzipedia::Back-End::App::Controller	TI11
Quizzipedia::Back-End::App::Model	TI12
Quizzipedia::Back-End::App::Routers	TI10
Quizzipedia::Back-End::Config	TI9
Quizzipedia::Front-End	TI1
Quizzipedia::Front-End::Controllers	TI2
Quizzipedia::Front-End::Directives	TI5
Quizzipedia::Front-End::Model	TI4
Quizzipedia::Front-End::Services	TI3
Quizzipedia::Front-End::Views	TI6

Tabella 10: Tracciamento Componenti-Test di Integrazione



5.7 Tracciamento Test di Unità-Metodi

Test	Metodi
TU1	Quizzipedia::Back-End::Server::start()
TU2	Quizzipedia::Back-End::App::Routers::- UserRouter::router()
TU3	Quizzipedia::Back-End::App::Routers::- QuizRouter::router()
TU4	Quizzipedia::Back-End::App::Routers::- QuestionRouter::router()
TU5	Quizzipedia::Back-End::App::Routers::- LangRouter::router()
TU6	Quizzipedia::Back-End::App::Controller::- NotFoundHandler::handle()
TU7	Quizzipedia::Back-End::App::Controller::- ErrorsHandler::handle()
TU8	Quizzipedia::Back-End::App::Controller::- Errors::QuizziPediaError::QuizziPediaError() Quizzipedia::Back-End::App::Controller::- Errors::QuizziPediaError::toJSON() Quizzipedia::Back-End::App::Controller::- Errors::QuizziPediaError::toString()
TU9	Quizzipedia::Back-End::App::Controller::- Users:: AuthenticationController::signin() Quizzipedia::Back-End::App::Controller::- Users:: AuthenticationController::signout() Quizzipedia::Back-End::App::Controller::- Users:: AuthenticationController::signup()
TU10	Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::deleteUser() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getInfo() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updateDataUser() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updatePasswordUser()
TU11	Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::deleteUser() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getInfo() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getSummaries() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getSummary() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updateDataUser() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updatePasswordUser() Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updateSummary()



Test	Metodi
TU12	Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::getStatistics() Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::updateStatisticsUser()
TU13	Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::getSummaries() Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::getSummary() Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::updateSummary()
TU14	Quizzipedia::Back-End::App::Controller::- TopicController::addQuestion() Quizzipedia::Back-End::App::Controller::- TopicController::updateTopic()
TU15	Quizzipedia::Back-End::App::Controller::- QuestionController::updateStatistic()
TU16	Quizzipedia::Back-End::App::Controller::- SummaryController::createSummary()
TU17	Quizzipedia::Back-End::App::Controller::- QuizController::getPersonalQuizzes()
TU18	Quizzipedia::Back-End::App::Controller::- QuizController::addActiveUser() Quizzipedia::Back-End::App::Controller::- QuizController::addUser() Quizzipedia::Back-End::App::Controller::- QuizController::editQuiz() Quizzipedia::Back-End::App::Controller::- QuizController::removeUser() Quizzipedia::Back-End::App::Controller::- QuizController::updateStatistic()
TU19	Quizzipedia::Back-End::App::Controller::- QuestionController::editQuestion() Quizzipedia::Back-End::App::Controller::- QuestionController::updateStatistic()
TU21	Quizzipedia::Back-End::App::Controller::- LangController::getVarList()
TU22	Quizzipedia::Back-End::App::Model::- UserModel::deleteUser() Quizzipedia::Back-End::App::Model::- UserModel::editPassword() Quizzipedia::Back-End::App::Model::- UserModel::editUser() Quizzipedia::Back-End::App::Model::- UserModel::getSummaries() Quizzipedia::Back-End::App::Model::- UserModel::getUsers() Quizzipedia::Back-End::App::Model::- UserModel::setImg() Quizzipedia::Back-End::App::Model::- UserModel::updateStatistics()



Test	Metodi
	Quizzipedia::Back-End::App::Model::- UserModel::updateSummary() Quizzipedia::Back-End::App::Model::- UserModel::upLevel() Quizzipedia::Back-End::App::Model::- UserProModel::addUserPro() Quizzipedia::Back-End::App::Model::- UserProModel::deleteUserPro()
TU23	Quizzipedia::Back-End::App::Model::- UserModel::hashPassword() Quizzipedia::Back-End::App::Model::- UserModel::validPassword()
TU24	Quizzipedia::Back-End::App::Model::- QuestionModel::addCorrect() Quizzipedia::Back-End::App::Model::- QuestionModel::addKeyword() Quizzipedia::Back-End::App::Model::- QuestionModel::addTotal() Quizzipedia::Back-End::App::Model::- QuestionModel::createQuestion() Quizzipedia::Back-End::App::Model::- QuestionModel::editQuestion() Quizzipedia::Back-End::App::Model::- QuestionModel::getQuestions() Quizzipedia::Back-End::App::Model::- QuestionModel::updateLevel()
TU25	Quizzipedia::Back-End::App::Model::- QuizModel::addActiveUser() Quizzipedia::Back-End::App::Model::- QuizModel::addCorrect() Quizzipedia::Back-End::App::Model::- QuizModel::addQuestion() Quizzipedia::Back-End::App::Model::- QuizModel::addUser() Quizzipedia::Back-End::App::Model::- QuizModel::createQuiz() Quizzipedia::Back-End::App::Model::- QuizModel::getPersonalQuizzes() Quizzipedia::Back-End::App::Model::- QuizModel::getQuiz() Quizzipedia::Back-End::App::Model::- QuizModel::removeQuestion() Quizzipedia::Back-End::App::Model::- QuizModel::removeUser() Quizzipedia::Back-End::App::Model::- QuizModel::searchQuiz() Quizzipedia::Back-End::App::Model::- QuizModel::setTitle()
TU26	Quizzipedia::Back-End::App::Model::- TopicModel::addQuestion()



Test	Metodi
	Quizzipedia::Back-End::App::Model::- TopicModel::getNextQuestion() Quizzipedia::Back-End::App::Model::- TopicModel::getQuestions() Quizzipedia::Back-End::App::Model::- TopicModel::updateCorrect() Quizzipedia::Back-End::App::Model::- TopicModel::updateTotal()
TU28	Quizzipedia::Back-End::App::Model::- LangModel::getVarlist()
TU29	Quizzipedia::Front-End::AppRouter::- appRouter()
TU30	Quizzipedia::Front-End::AppRun::- AppRun() Quizzipedia::Front-End::AppRun::- getLag() Quizzipedia::Front-End::AppRun::- getUserDetails()
TU31	Quizzipedia::Front-End::Controllers::- QuestionnaireDetailsController::getQuestionnaireDetails() Quizzipedia::Front-End::Controllers::- QuestionnaireDetailsController::getSubscribedQuestionnaire() Quizzipedia::Front-End::Controllers::- QuestionnaireDetailsController::QuestionnaireDetailsController()
TU32	Quizzipedia::Front-End::Controllers::- QuestionnaireManagementController::getUserQuestionnaire() Quizzipedia::Front-End::Controllers::- QuestionnaireManagementController::QuestionnaireManagementController()
TU33	Quizzipedia::Front-End::Controllers::- InputToListController::InputToListController() Quizzipedia::Front-End::Controllers::- InputToListController::putDownAnswer()
TU34	Quizzipedia::Front-End::Controllers::- LoginController::logIn() Quizzipedia::Front-End::Controllers::- LoginController::LoginController() Quizzipedia::Front-End::Controllers::- LoginController::recoveryPassword() Quizzipedia::Front-End::Controllers::- LoginController::signUp()
TU35	Quizzipedia::Front-End::Controllers::- ClickableAreaQuestionsController::chosedThatPoint() Quizzipedia::Front-End::Controllers::- ClickableAreaQuestionsController::ClickableAreaQuestionsController() Quizzipedia::Front-End::Controllers::- ClickableAreaQuestionsController::submitQuestion()
TU36	Quizzipedia::Front-End::Controllers::- ConnectionQuestionsController::ConnectionQuestionsController() Quizzipedia::Front-End::Controllers::- ConnectionQuestionsController::submitQuestion()



Test	Metodi
	Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::addQuestion()
TU37	Quizzipedia::Front-End::Controllers::- MenuBarController::\$on() Quizzipedia::Front-End::Controllers::- MenuBarController::goToQuestionsManagementPage() Quizzipedia::Front-End::Controllers::- MenuBarController::goToQuizManagementPage() Quizzipedia::Front-End::Controllers::- MenuBarController::goToUserManagementPage() Quizzipedia::Front-End::Controllers::- MenuBarController::goToUserPage() Quizzipedia::Front-End::Controllers::- MenuBarController::login() Quizzipedia::Front-End::Controllers::- MenuBarController::logout() Quizzipedia::Front-End::Controllers::- MenuBarController::MenuBarController() Quizzipedia::Front-End::Controllers::- MenuBarController::signUp()
TU38	Quizzipedia::Front-End::Controllers::- MultipleQuestionsController::MultipleQuestionsController() Quizzipedia::Front-End::Controllers::- MultipleQuestionsController::submitQuestion()
TU39	Quizzipedia::Front-End::Controllers::- NewQuestionsButtonController::newQuestion() Quizzipedia::Front-End::Controllers::- NewQuestionsButtonController::NewQuestionButtonsController()
TU40	Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::addQuestion() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::createQuestionnaire() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::CreateQuestionnaireController() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::deleteQuestionnaire() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestion() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestionnaire() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestionnairePreview() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestions() Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::modifyQuestionnaire()
TU41	Quizzipedia::Front-End::Controllers::- PasswordForgotController::login() Quizzipedia::Front-End::Controllers::- PasswordForgotController::passwordForgot()



Test	Metodi
	Quizzipedia::Front-End::Controllers::- PasswordForgotController::PasswordForgotController()
TU42	Quizzipedia::Front-End::Controllers::- QuestionsController::addAnswer() Quizzipedia::Front-End::Controllers::- QuestionsController::answerGiven() Quizzipedia::Front-End::Controllers::- QuestionsController::checkAnswer() Quizzipedia::Front-End::Controllers::- QuestionsController::getQuestionBy() Quizzipedia::Front-End::Controllers::- QuestionsController::linkingMade() Quizzipedia::Front-End::Controllers::- QuestionsController::loadNewQuestion() Quizzipedia::Front-End::Controllers::- QuestionsController::loadNewQuestionBy() Quizzipedia::Front-End::Controllers::- QuestionsController::orderChosen() Quizzipedia::Front-End::Controllers::- QuestionsController::QuestionsController()
TU43	Quizzipedia::Front-End::Controllers::- ProfileManagementController::confirm() Quizzipedia::Front-End::Controllers::- ProfileManagementController::ProfileManagementController()
TU44	Quizzipedia::Front-End::Controllers::- SignUpController::logIn() Quizzipedia::Front-End::Controllers::- SignUpController::signUp() Quizzipedia::Front-End::Controllers::- SignUpController::SignUpController()
TU45	Quizzipedia::Front-End::Controllers::- EditorQMLController::EditorQMLController() Quizzipedia::Front-End::Controllers::- EditorQMLController::submitQuestion()
TU46	Quizzipedia::Front-End::Controllers::- QuestionsManagementController::editQuestion() Quizzipedia::Front-End::Controllers::- QuestionsManagementController::getQuestionsByUser() Quizzipedia::Front-End::Controllers::- QuestionsManagementController::QuestionsManagementsController()
TU47	Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::FillingQuestionnaireController() Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::getNextQuestion() Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::getQuestionnaire() Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::startQuiz()
TU48	Quizzipedia::Front-End::Controllers::- QuizEventController::deleteQuestionnaire()



Test	Metodi
	Quizzipedia::Front-End::Controllers::- QuizEventController::examModalityquizId() Quizzipedia::Front-End::Controllers::- QuizEventController::modifyQuestionnaire() Quizzipedia::Front-End::Controllers::- QuizEventController::QuizEventController() Quizzipedia::Front-End::Controllers::- QuizEventController::resultsQuestionnaire() Quizzipedia::Front-End::Controllers::- QuizEventController::subscribeManagement()
TU49	Quizzipedia::Front-End::Controllers::- RegistrationManagementController::RegistrationmanagementController() Quizzipedia::Front-End::Controllers::- RegistrationManagementController::subscribeQuestionnaire()
TU50	Quizzipedia::Front-End::Controllers::- ResultsQuestionnaireController::getQuizResults() Quizzipedia::Front-End::Controllers::- ResultsQuestionnaireController::getUserForThisQuestionnaire() Quizzipedia::Front-End::Controllers::- ResultsQuestionnaireController::ResultsQuestionnaireController()
TU51	Quizzipedia::Front-End::Controllers::- StatisticsController::getStatistics() Quizzipedia::Front-End::Controllers::- StatisticsController::StatisticsController()
TU52	Quizzipedia::Front-End::Controllers::- FillingQuestionsController::choseThatWord() Quizzipedia::Front-End::Controllers::- FillingQuestionsController::FillingQuestionsController() Quizzipedia::Front-End::Controllers::- FillingQuestionsController::submitQuestion()
TU53	Quizzipedia::Front-End::Controllers::- HomeController::HomeController() Quizzipedia::Front-End::Controllers::- HomeController::trainingMode()
TU54	Quizzipedia::Front-End::Controllers::- ImagesSortingQuestionsController::ImageSortingQuestionsController() Quizzipedia::Front-End::Controllers::- ImagesSortingQuestionsController::submitQuestion()
TU55	Quizzipedia::Front-End::Controllers::- StringsSortingQuestionsController::StringsSortingQuestionsController() Quizzipedia::Front-End::Controllers::- StringsSortingQuestionsController::submitQuestion()
TU56	Quizzipedia::Front-End::Controllers::- SearchController::getSearch() Quizzipedia::Front-End::Controllers::- SearchController::goToResultsPage() Quizzipedia::Front-End::Controllers::- SearchController::goToUser() Quizzipedia::Front-End::Controllers::- SearchController::registrationToQuiz()



Test	Metodi
	Quizzipedia::Front-End::Controllers::- SearchController::SearchController()
TU57	Quizzipedia::Front-End::Controllers::- TopicKeywordsController::getKeywords() Quizzipedia::Front-End::Controllers::- TopicKeywordsController::TopicKeywordsController()
TU58	Quizzipedia::Front-End::Controllers::- TrainingController::addQuestion() Quizzipedia::Front-End::Controllers::- TrainingController::addResult() Quizzipedia::Front-End::Controllers::- TrainingController::loadNewQuestionBy() Quizzipedia::Front-End::Controllers::- TrainingController::startTraining() Quizzipedia::Front-End::Controllers::- TrainingController::TrainingController()
TU59	Quizzipedia::Front-End::Controllers::- TrueFalseQuestionsController::submitQuestion() Quizzipedia::Front-End::Controllers::- TrueFalseQuestionsController::TrueFalseQuestionsController()
TU60	Quizzipedia::Front-End::Controllers::- UserDetailsController::getUserDetails() Quizzipedia::Front-End::Controllers::- UserDetailsController::UserDetailsController()
TU61	Quizzipedia::Front-End::Model::- QuestionnaireModel::addQuestion() Quizzipedia::Front-End::Model::- QuestionnaireModel::getAuthor() Quizzipedia::Front-End::Model::- QuestionnaireModel::getQuestion() Quizzipedia::Front-End::Model::- QuestionnaireModel::getTitle() Quizzipedia::Front-End::Model::- QuestionnaireModel::removeQuestion() Quizzipedia::Front-End::Model::- QuestionnaireModel::setAuthor() Quizzipedia::Front-End::Model::- QuestionnaireModel::setTitle()
TU62	Quizzipedia::Front-End::Model::UserDetailsModel::- getEmail() Quizzipedia::Front-End::Model::UserDetailsModel::- getId() Quizzipedia::Front-End::Model::UserDetailsModel::- getLevelUser() Quizzipedia::Front-End::Model::UserDetailsModel::- getName() Quizzipedia::Front-End::Model::UserDetailsModel::- getPrivilege() Quizzipedia::Front-End::Model::UserDetailsModel::- getStatistics()



Test	Metodi
	<p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>getSurname()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>getUserImg()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>getUsername()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setEmail()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setId()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setLevel()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setName()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setPrivilege()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setStatistics()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setSurname()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setUserImg()</p> <p>Quizzipedia::Front-End::Model::UserDetailsModel::-</p> <p>setUsername()</p>
TU63	<p>Quizzipedia::Front-End::Model::MenuBarModel::-</p> <p>getDirective()</p>
TU64	<p>Quizzipedia::Front-End::Model::ErrorInfoModel::-</p> <p>ErrorModel()</p> <p>Quizzipedia::Front-End::Model::ErrorInfoModel::-</p> <p>getCode()</p> <p>Quizzipedia::Front-End::Model::ErrorInfoModel::-</p> <p>getMessage()</p> <p>Quizzipedia::Front-End::Model::ErrorInfoModel::-</p> <p>getTitle()</p>
TU65	<p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::addPieceOfQuestion()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::compareAnswers()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::createQuestion()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::getAuthor()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::getId()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::getLanguage()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::getMadeWith()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::getQuestion()</p>



Test	Metodi
	<p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::setAuthor()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::setId()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::setLanguage()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::setMadeWith()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>QuestionItemModel::setQuestion()</p>
TU66	<p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::addKeyword()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::addQuestion()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::getArgument()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::getNumeberOfQuestions()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::getResult()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::removeKeyword()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::removeQuestion()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::setArgument()</p> <p>Quizzipedia::Front-End::Model::-</p> <p>TrainingModeModel::setNumberOfQuestions()</p>
TU67	<p>Quizzipedia::Back-End::App::Model::-</p> <p>LangModel::getVarlist()</p> <p>Quizzipedia::Front-End::Model::LangModel::-</p> <p>getLang()</p> <p>Quizzipedia::Front-End::Model::LangModel::-</p> <p>LangModel()</p>
TU68	<p>Quizzipedia::Front-End::Model::LangModel::-</p> <p>LangModel()</p> <p>Quizzipedia::Front-End::Model::LangModel::-</p> <p>setNewLang()</p>
TU69	<p>Quizzipedia::Front-End::Services::-</p> <p>QuestionsService::getKeywords()</p> <p>Quizzipedia::Front-End::Services::-</p> <p>QuestionsService::getNextQuestion()</p> <p>Quizzipedia::Front-End::Services::-</p> <p>QuestionsService::getQuestion()</p> <p>Quizzipedia::Front-End::Services::-</p> <p>QuestionsService::getUserQuestions()</p> <p>Quizzipedia::Front-End::Services::-</p> <p>QuestionsService::QuestionService()</p> <p>Quizzipedia::Front-End::Services::-</p> <p>QuestionsService::sendQuestion()</p>



Test	Metodi
TU70	Quizzipedia::Front-End::Services:- AuthService::AuthService() Quizzipedia::Front-End::Services:- AuthService::getNewPassword() Quizzipedia::Front-End::Services:- AuthService::isLoggedIn() Quizzipedia::Front-End::Services:- AuthService::logout() Quizzipedia::Front-End::Services:- AuthService::signIn() Quizzipedia::Front-End::Services:- AuthService::signup()
TU71	Quizzipedia::Front-End::Services:- LangService::getKeywords() Quizzipedia::Front-End::Services:- LangService::LangService()
TU72	Quizzipedia::Front-End::Services:- SearchService::searchQuestionnaire() Quizzipedia::Front-End::Services:- SearchService::SearchService() Quizzipedia::Front-End::Services:- SearchService::searchUsers()
TU73	Quizzipedia::Front-End::Services:- QuizService::approveSubscribeQuestionnaire() Quizzipedia::Front-End::Services:- QuizService::createQuestionnaire() Quizzipedia::Front-End::Services:- QuizService::getDoneQuestionnaire() Quizzipedia::Front-End::Services:- QuizService::getQuestionnaireDetails() Quizzipedia::Front-End::Services:- QuizService::getQuiz() Quizzipedia::Front-End::Services:- QuizService::getQuizResults() Quizzipedia::Front-End::Services:- QuizService::getSubscribedQuestionnaire() Quizzipedia::Front-End::Services:- QuizService::getUserForThisQuestionnaire() Quizzipedia::Front-End::Services:- QuizService::getUserQuestionnaire() Quizzipedia::Front-End::Services:- QuizService::QuizService() Quizzipedia::Front-End::Services:- QuizService::search() Quizzipedia::Front-End::Services:- QuizService::subscribeQuestionnaire()
TU74	Quizzipedia::Front-End::Services:- StatisticsService::getQuestionnaireStatistics() Quizzipedia::Front-End::Services:- StatisticsService::getStatistics()



Test	Metodi
	Quizzipedia::Front-End::Services::- StatisticsService::StatisticsService()
TU75	Quizzipedia::Front-End::Services::- UserDetailsService::confirmSaveUser() Quizzipedia::Front-End::Services::- UserDetailsService::getUserDetails() Quizzipedia::Front-End::Services::- UserDetailsService::UserDetailsService()

Tabella 11: Tracciamento Test di Unità-Metodi



5.8 Tracciamento Metodi-Test di Unità

Metodo	Test
Quizzipedia::Back-End::App::Controller::- ErrorsHandler::handle()	TU7
Quizzipedia::Back-End::App::Controller::- LangController::getVarList()	TU21
Quizzipedia::Back-End::App::Controller::- NotFoundHandler::handle()	TU6
Quizzipedia::Back-End::App::Controller::- QuestionController::editQuestion()	TU19
Quizzipedia::Back-End::App::Controller::- QuestionController::updateStatistic()	TU15 TU19
Quizzipedia::Back-End::App::Controller::- QuizController::addActiveUser()	TU18
Quizzipedia::Back-End::App::Controller::- QuizController::addUser()	TU18
Quizzipedia::Back-End::App::Controller::- QuizController::editQuiz()	TU18
Quizzipedia::Back-End::App::Controller::- QuizController::getPersonalQuizzes()	TU17
Quizzipedia::Back-End::App::Controller::- QuizController::removeUser()	TU18
Quizzipedia::Back-End::App::Controller::- QuizController::updateStatistic()	TU18
Quizzipedia::Back-End::App::Controller::- SummaryController::createSummary()	TU16
Quizzipedia::Back-End::App::Controller::- Errors::QuizziPediaError::QuizziPediaError()	TU8
Quizzipedia::Back-End::App::Controller::- Errors::QuizziPediaError::toJSON()	TU8
Quizzipedia::Back-End::App::Controller::- Errors::QuizziPediaError::toString()	TU8
Quizzipedia::Back-End::App::Controller::- TopicController::addQuestion()	TU14
Quizzipedia::Back-End::App::Controller::- TopicController::updateTopic()	TU14
Quizzipedia::Back-End::App::Controller::- Users:: AuthenticationController::signin()	TU9
Quizzipedia::Back-End::App::Controller::- Users:: AuthenticationController::signout()	TU9
Quizzipedia::Back-End::App::Controller::- Users:: AuthenticationController::signup()	TU9
Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::deleteUser()	TU10 TU11
Quizzipedia::Back-End::App::Controller::- Users:: UserManagementController::getInfo()	TU10 TU11



Metodo	Test
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getStatistics()	TU12
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getSummaries()	TU11 TU13
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::getSummary()	TU11 TU13
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updateDataUser()	TU10 TU11
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updatePasswordUser()	TU10 TU11
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updateStatisticsUser()	TU12
Quizzipedia::Back-End::App::Controller::- Users:: UserManagerController::updateSummary()	TU11 TU13
Quizzipedia::Back-End::App::Model::- LangModel::getVarlist()	TU28 TU67
Quizzipedia::Back-End::App::Model::- QuestionModel::addCorrect()	TU24
Quizzipedia::Back-End::App::Model::- QuestionModel::addKeyword()	TU24
Quizzipedia::Back-End::App::Model::- QuestionModel::addTotal()	TU24
Quizzipedia::Back-End::App::Model::- QuestionModel::createQuestion()	TU24
Quizzipedia::Back-End::App::Model::- QuestionModel::editQuestion()	TU24
Quizzipedia::Back-End::App::Model::- QuestionModel::getQuestions()	TU24
Quizzipedia::Back-End::App::Model::- QuestionModel::updateLevel()	TU24
Quizzipedia::Back-End::App::Model::- QuizModel::addActiveUser()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::addCorrect()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::addQuestion()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::addUser()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::createQuiz()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::getPersonalQuizzes()	TU25



Metodo	Test
Quizzipedia::Back-End::App::Model::- QuizModel::getQuiz()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::removeQuestion()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::removeUser()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::searchQuiz()	TU25
Quizzipedia::Back-End::App::Model::- QuizModel::setTitle()	TU25
Quizzipedia::Back-End::App::Model::- TopicModel::addQuestion()	TU26
Quizzipedia::Back-End::App::Model::- TopicModel::getNextQuestion()	TU26
Quizzipedia::Back-End::App::Model::- TopicModel::getQuestions()	TU26
Quizzipedia::Back-End::App::Model::- TopicModel::updateCorrect()	TU26
Quizzipedia::Back-End::App::Model::- TopicModel::updateTotal()	TU26
Quizzipedia::Back-End::App::Model::- UserModel::deleteUser()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::editPassword()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::editUser()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::getSummaries()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::getUsers()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::hashPassword()	TU23
Quizzipedia::Back-End::App::Model::- UserModel::setImg()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::updateStatistics()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::updateSummary()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::upLevel()	TU22
Quizzipedia::Back-End::App::Model::- UserModel::validPassword()	TU23
Quizzipedia::Back-End::App::Model::- UserProModel::addUserPro()	TU22
Quizzipedia::Back-End::App::Model::- UserProModel::deleteUserPro()	TU22
Quizzipedia::Back-End::App::Router::- LangRouter::router()	TU5



Metodo	Test
Quizzipedia::Back-End::App::Routers::- QuestionRouter::router()	TU4
Quizzipedia::Back-End::App::Routers::- QuizRouter::router()	TU3
Quizzipedia::Back-End::App::Routers::- UserRouter::router()	TU2
Quizzipedia::Back-End::Server::start()	TU1
Quizzipedia::Front-End::AppRouter::- appRouter()	TU29
Quizzipedia::Front-End::AppRun::- AppRun()	TU30
Quizzipedia::Front-End::AppRun::- getLag()	TU30
Quizzipedia::Front-End::AppRun::- getUserDetails()	TU30
Quizzipedia::Front-End::Controllers::- ClickableAreaQuestionsController::choseThatPoint()	TU35
Quizzipedia::Front-End::Controllers::- ClickableAreaQuestionsController::ClickableAreaQuestionsController()	TU35
Quizzipedia::Front-End::Controllers::- ClickableAreaQuestionsController::submitQuestion()	TU35
Quizzipedia::Front-End::Controllers::- ConnectionQuestionsController::ConnectionQuestionsController()	TU36
Quizzipedia::Front-End::Controllers::- ConnectionQuestionsController::submitQuestion()	TU36
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::addQuestion()	TU36 TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::createQuestionnaire()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::CreateQuestionnaireController()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::deleteQuestionnaire()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestion()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestionnaire()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestionnairePreview()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::getQuestions()	TU40
Quizzipedia::Front-End::Controllers::- CreateQuestionnaireController::modifyQuestionnaire()	TU40
Quizzipedia::Front-End::Controllers::- EditorQMLController::EditorQMLController()	TU45
Quizzipedia::Front-End::Controllers::- EditorQMLController::submitQuestion()	TU45



Metodo	Test
Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::FillingQuestionnaireController()	TU47
Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::getNextQuestion()	TU47
Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::getQuestionnaire()	TU47
Quizzipedia::Front-End::Controllers::- FillingQuestionnaireController::startQuiz()	TU47
Quizzipedia::Front-End::Controllers::- FillingQuestionsController::choseThatWord()	TU52
Quizzipedia::Front-End::Controllers::- FillingQuestionsController::FillingQuestionsController()	TU52
Quizzipedia::Front-End::Controllers::- FillingQuestionsController::submitQuestion()	TU52
Quizzipedia::Front-End::Controllers::- HomeController::HomeController()	TU53
Quizzipedia::Front-End::Controllers::- HomeController::trainingMode()	TU53
Quizzipedia::Front-End::Controllers::- ImagesSortingQuestionsController::ImageSortingQuestionsController()	TU54
Quizzipedia::Front-End::Controllers::- ImagesSortingQuestionsController::submitQuestion()	TU54
Quizzipedia::Front-End::Controllers::- InputToListController::InputToListController()	TU33
Quizzipedia::Front-End::Controllers::- InputToListController::putDownAnswer()	TU33
Quizzipedia::Front-End::Controllers::- LoginController::login()	TU34
Quizzipedia::Front-End::Controllers::- LoginController::LoginController()	TU34
Quizzipedia::Front-End::Controllers::- LoginController::recoveryPassword()	TU34
Quizzipedia::Front-End::Controllers::- LoginController::signUp()	TU34
Quizzipedia::Front-End::Controllers::- MenuBarController::\$on()	TU37
Quizzipedia::Front-End::Controllers::- MenuBarController::goToQuestionsManagementPage()	TU37
Quizzipedia::Front-End::Controllers::- MenuBarController::goToQuizManagementPage()	TU37
Quizzipedia::Front-End::Controllers::- MenuBarController::goToUserManagementPage()	TU37
Quizzipedia::Front-End::Controllers::- MenuBarController::goToUserPage()	TU37
Quizzipedia::Front-End::Controllers::- MenuBarController::login()	TU37
Quizzipedia::Front-End::Controllers::- MenuBarController::logout()	TU37



Metodo	Test
Quizzipedia::Front-End::Controllers::-MenuBarController::MenuBarController()	TU37
Quizzipedia::Front-End::Controllers::-MenuBarController::signUp()	TU37
Quizzipedia::Front-End::Controllers::-MultipleQuestionsController::MultipleQuestionsController()	TU38
Quizzipedia::Front-End::Controllers::-MultipleQuestionsController::submitQuestion()	TU38
Quizzipedia::Front-End::Controllers::-NewQuestionsButtonController::newQuestion()	TU39
Quizzipedia::Front-End::Controllers::-NewQuestionsButtonController::NewQuestionButtonsController()	TU39
Quizzipedia::Front-End::Controllers::-PasswordForgotController::login()	TU41
Quizzipedia::Front-End::Controllers::-PasswordForgotController::passwordForgot()	TU41
Quizzipedia::Front-End::Controllers::-PasswordForgotController::PasswordForgotController()	TU41
Quizzipedia::Front-End::Controllers::-ProfileManagementController::confirm()	TU43
Quizzipedia::Front-End::Controllers::-ProfileManagementController::ProfileManagementController()	TU43
Quizzipedia::Front-End::Controllers::-QuestionnaireDetailsController::getQuestionnaireDetails()	TU31
Quizzipedia::Front-End::Controllers::-QuestionnaireDetailsController::getSubscribedQuestionnaire()	TU31
Quizzipedia::Front-End::Controllers::-QuestionnaireDetailsController::QuestionnaireDetailsController()	TU31
Quizzipedia::Front-End::Controllers::-QuestionnaireManagementController::getUserQuestionnaire()	TU32
Quizzipedia::Front-End::Controllers::-QuestionnaireManagementController::QuestionnaireManagementController()	TU32
Quizzipedia::Front-End::Controllers::-QuestionsController::addAnswer()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::answerGiven()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::checkAnswer()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::getQuestionBy()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::linkingMade()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::loadNewQuestion()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::loadNewQuestionBy()	TU42
Quizzipedia::Front-End::Controllers::-QuestionsController::orderChosen()	TU42



Metodo	Test
Quizzipedia::Front-End::Controllers::- QuestionsController::QuestionsController()	TU42
Quizzipedia::Front-End::Controllers::- QuestionsManagementController::editQuestion()	TU46
Quizzipedia::Front-End::Controllers::- QuestionsManagementController::getQuestionsByUser()	TU46
Quizzipedia::Front-End::Controllers::- QuestionsManagementController::QuestionsManagementsController()	TU46
Quizzipedia::Front-End::Controllers::- QuizEventController::deleteQuestionnaire()	TU48
Quizzipedia::Front-End::Controllers::- QuizEventController::examModalityquizId()	TU48
Quizzipedia::Front-End::Controllers::- QuizEventController::modifyQuestionnaire()	TU48
Quizzipedia::Front-End::Controllers::- QuizEventController::QuizEventController()	TU48
Quizzipedia::Front-End::Controllers::- QuizEventController::resultsQuestionnaire()	TU48
Quizzipedia::Front-End::Controllers::- QuizEventController::subscribeManagement()	TU48
Quizzipedia::Front-End::Controllers::- RegistrationManagementController::RegistrationmanagementController()	TU49
Quizzipedia::Front-End::Controllers::- RegistrationManagementController::subscribeQuestionnaire()	TU49
Quizzipedia::Front-End::Controllers::- ResultsQuestionnaireController::getQuizResults()	TU50
Quizzipedia::Front-End::Controllers::- ResultsQuestionnaireController::getUserForThisQuestionnaire()	TU50
Quizzipedia::Front-End::Controllers::- ResultsQuestionnaireController::ResultsQuestionnaireController()	TU50
Quizzipedia::Front-End::Controllers::- SearchController::getSearch()	TU56
Quizzipedia::Front-End::Controllers::- SearchController::goToResultsPage()	TU56
Quizzipedia::Front-End::Controllers::- SearchController::goToUser()	TU56
Quizzipedia::Front-End::Controllers::- SearchController::registrationToQuiz()	TU56
Quizzipedia::Front-End::Controllers::- SearchController::SearchController()	TU56
Quizzipedia::Front-End::Controllers::- SignUpController::login()	TU44
Quizzipedia::Front-End::Controllers::- SignUpController::signUp()	TU44
Quizzipedia::Front-End::Controllers::- SignUpController::SignUpController()	TU44
Quizzipedia::Front-End::Controllers::- StatisticsController::getStatistics()	TU51



Metodo	Test
Quizzipedia::Front-End::Controllers::- StatisticsController::StatisticsController()	TU51
Quizzipedia::Front-End::Controllers::- StringsSortingQuestionsController::StringsSortingQuestionsController()	TU55
Quizzipedia::Front-End::Controllers::- StringsSortingQuestionsController::submitQuestion()	TU55
Quizzipedia::Front-End::Controllers::- TopicKeywordsController::getKeywords()	TU57
Quizzipedia::Front-End::Controllers::- TopicKeywordsController::TopicKeywordsController()	TU57
Quizzipedia::Front-End::Controllers::- TrainingController::addQuestion()	TU58
Quizzipedia::Front-End::Controllers::- TrainingController::addResult()	TU58
Quizzipedia::Front-End::Controllers::- TrainingController::loadNewQuestionBy()	TU58
Quizzipedia::Front-End::Controllers::- TrainingController::startTraining()	TU58
Quizzipedia::Front-End::Controllers::- TrainingController::TrainingController()	TU58
Quizzipedia::Front-End::Controllers::- TrueFalseQuestionsController::submitQuestion()	TU59
Quizzipedia::Front-End::Controllers::- TrueFalseQuestionsController::TrueFalseQuestionsController()	TU59
Quizzipedia::Front-End::Controllers::- UserDetailsController::getUserDetails()	TU60
Quizzipedia::Front-End::Controllers::- UserDetailsController::UserDetailsController()	TU60
Quizzipedia::Front-End::Model::ErrorInfoModel::- ErrorModel()	TU64
Quizzipedia::Front-End::Model::ErrorInfoModel::- getCode()	TU64
Quizzipedia::Front-End::Model::ErrorInfoModel::- getMessage()	TU64
Quizzipedia::Front-End::Model::ErrorInfoModel::- getTitle()	TU64
Quizzipedia::Front-End::Model::LangModel::- getLang()	TU67
Quizzipedia::Front-End::Model::LangModel::- LangModel()	TU67 TU68
Quizzipedia::Front-End::Model::LangModel::- setNewLang()	TU68
Quizzipedia::Front-End::Model::MenuBarModel::- getDirective()	TU63
Quizzipedia::Front-End::Model::- QuestionItemModel::addPieceOfQuestion()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::compareAnswers()	TU65



Metodo	Test
Quizzipedia::Front-End::Model::- QuestionItemModel::createQuestion()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::getAuthor()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::getId()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::getLanguage()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::getMadeWith()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::getQuestion()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::setAuthor()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::setId()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::setLanguage()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::setMadeWith()	TU65
Quizzipedia::Front-End::Model::- QuestionItemModel::setQuestion()	TU65
Quizzipedia::Front-End::Model::- QuestionnaireModel::addQuestion()	TU61
Quizzipedia::Front-End::Model::- QuestionnaireModel::getAuthor()	TU61
Quizzipedia::Front-End::Model::- QuestionnaireModel::getQuestion()	TU61
Quizzipedia::Front-End::Model::- QuestionnaireModel::getTitle()	TU61
Quizzipedia::Front-End::Model::- QuestionnaireModel::removeQuestion()	TU61
Quizzipedia::Front-End::Model::- QuestionnaireModel::setAuthor()	TU61
Quizzipedia::Front-End::Model::- QuestionnaireModel::setTitle()	TU61
Quizzipedia::Front-End::Model::- TrainingModeModel::addKeyword()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::addQuestion()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::getArgument()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::getNumbeberOfQuestions()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::getResult()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::removeKeyword()	TU66



Metodo	Test
Quizzipedia::Front-End::Model::- TrainingModeModel::removeQuestion()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::setArgument()	TU66
Quizzipedia::Front-End::Model::- TrainingModeModel::setNumberOfQuestions()	TU66
Quizzipedia::Front-End::Model::UserDetailsModel::- getEmail()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getId()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getLevelUser()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getName()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getPrivilege()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getStatistics()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getSurname()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getUserImg()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- getUsername()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setEmail()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setId()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setLevel()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setName()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setPrivilege()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setStatistics()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setSurname()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setUserImg()	TU62
Quizzipedia::Front-End::Model::UserDetailsModel::- setUsername()	TU62
Quizzipedia::Front-End::Services::- AuthService::AuthService()	TU70
Quizzipedia::Front-End::Services::- AuthService::getNewPassword()	TU70
Quizzipedia::Front-End::Services::- AuthService::isLoggedIn()	TU70



Metodo	Test
Quizzipedia::Front-End::Services::-AuthService::logout()	TU70
Quizzipedia::Front-End::Services::-AuthService::signIn()	TU70
Quizzipedia::Front-End::Services::-AuthService::signup()	TU70
Quizzipedia::Front-End::Services::-LangService::getKeywords()	TU71
Quizzipedia::Front-End::Services::-LangService::LangService()	TU71
Quizzipedia::Front-End::Services::-QuestionsService::getKeywords()	TU69
Quizzipedia::Front-End::Services::-QuestionsService::getNextQuestion()	TU69
Quizzipedia::Front-End::Services::-QuestionsService::getQuestion()	TU69
Quizzipedia::Front-End::Services::-QuestionsService::getUserQuestions()	TU69
Quizzipedia::Front-End::Services::-QuestionsService::QuestionService()	TU69
Quizzipedia::Front-End::Services::-QuestionsService::sendQuestion()	TU69
Quizzipedia::Front-End::Services::-QuizService::approveSubscribeQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-QuizService::createQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-QuizService::getDoneQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-QuizService::getQuestionnaireDetails()	TU73
Quizzipedia::Front-End::Services::-QuizService::getQuiz()	TU73
Quizzipedia::Front-End::Services::-QuizService::getQuizResults()	TU73
Quizzipedia::Front-End::Services::-QuizService::getSubscribedQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-QuizService::getUserForThisQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-QuizService::getUserQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-QuizService::QuizService()	TU73
Quizzipedia::Front-End::Services::-QuizService::search()	TU73
Quizzipedia::Front-End::Services::-QuizService::subscribeQuestionnaire()	TU73
Quizzipedia::Front-End::Services::-SearchService::searchQuestionnaire()	TU72



Metodo	Test
Quizzipedia::Front-End::Services::- SearchService::SearchService()	TU72
Quizzipedia::Front-End::Services::- SearchService::searchUsers()	TU72
Quizzipedia::Front-End::Services::- StatisticsService::getQuestionnaireStatistics()	TU74
Quizzipedia::Front-End::Services::- StatisticsService::getStatistics()	TU74
Quizzipedia::Front-End::Services::- StatisticsService::StatisticsService()	TU74
Quizzipedia::Front-End::Services::- UserDetailsService::confirmSaveUser()	TU75
Quizzipedia::Front-End::Services::- UserDetailsService::getUserDetails()	TU75
Quizzipedia::Front-End::Services::- UserDetailsService::UserDetailsService()	TU75

Tabella 12: Tracciamento Metodi-Test di Unità



A Resoconto attività di verifica

In questa sezione del documento vengono descritti e analizzati gli esiti delle attività di verifica svolte su tutti i documenti che vengono consegnati nelle varie revisioni di avanzamento del progetto.

A.1 Revisione dei Requisiti

A.1.1 Tracciamento

Il $team_G$ ha deciso di utilizzare il software interno *DocumentsDB* in modo da facilitare il tracciamento sia delle relazioni fra casi d'uso e requisiti, sia delle relazioni fra requisiti e fonti.

A.1.2 Analisi statica dei documenti

L'analisi dei documenti mediante *Walkthrough_G* ha portato all'individuazione di alcuni errori frequenti a partire dai quali è stata stilata una lista di controllo che è stata inserita all'interno dei Processi Organizzativi nelle *Norme di Progetto*. Grazie a questa sarà possibile applicare l'*Inspection_G* per le future attività di verifica.

A.1.3 Esiti verifiche automatizzate

Vengono qui riportati gli esiti delle verifiche automatizzate per il calcolo dell'*indice Gulpease_G*, alle quali sono stati sottoposti tutti i documenti.

Documento	Indice Gulpease	Esito
<i>Norme di Progetto</i>	68	Superato
<i>Studio di Fattibilità</i>	56	Superato
<i>Piano di Progetto</i>	60	Superato
<i>Piano di Qualifica</i>	57	Superato
<i>Analisi dei Requisiti</i>	72	Superato
<i>Glossario</i>	48	Superato
<i>Verbale Interno 2015-12-03</i>	75	Superato
<i>Verbale Interno 2015-12-11</i>	74	Superato
<i>Verbale Interno 2015-12-29</i>	75	Superato
<i>Verbale Interno 2016-01-12</i>	75	Superato
<i>Verbale Esterno 2016-01-11</i>	76	Superato

Tabella 13: Resoconto verifiche automatizzate - Revisione dei Requisiti

A.2 Revisione di Progettazione

A.2.1 Tracciamento

Il $team_G$ attraverso il software interno *DocumentsDB* è riuscito a effettuare il tracciamento sia delle relazioni fra requisiti e componenti che fra requisiti e classi. Questo software è stato utilizzato inoltre per generare le tabelle dei vari tipi di test e dei relativi tracciamenti con requisiti, componenti, classi e metodi.

A.2.2 Analisi statica dei documenti

L'analisi dei documenti mediante *Walkthrough_G* ha portato all'individuazione di alcuni errori frequenti a partire dai quali è stata stilata una lista di controllo che è stata inserita all'interno



dei Processi Organizzativi nelle *Norme di Progetto*. Grazie a questa sarà possibile applicare l'*Inspection_G* per le future attività di verifica.

A.2.3 Esiti verifiche automatizzate

Vengono qui riportati gli esiti delle verifiche automatizzate per il calcolo dell'*indice Gulpease_G*, alle quali sono stati sottoposti tutti i documenti.

Documento	Indice Gulpease	Esito
<i>Definizione di Prodotto</i>	61	Superato
<i>Norme di Progetto</i>	70	Superato
<i>Piano di Progetto</i>	61	Superato
<i>Piano di Qualifica</i>	68	Superato
<i>Analisi dei Requisiti</i>	70	Superato
<i>Glossario</i>	49	Superato
<i>Verbale Interno 2015-12-03</i>	75	Superato
<i>Verbale Interno 2015-12-11</i>	74	Superato
<i>Verbale Interno 2015-12-29</i>	75	Superato
<i>Verbale Interno 2016-01-12</i>	75	Superato
<i>Verbale Interno 2015-02-25</i>	76	Superato
<i>Verbale Interno 2015-03-01</i>	74	Superato
<i>Verbale Interno 2015-03-07</i>	76	Superato
<i>Verbale Esterno 2016-01-11</i>	75	Superato
<i>Verbale Esterno 2016-03-11</i>	76	Superato

Tabella 14: Resoconto verifiche automatizzate - Revisione di Progettazione