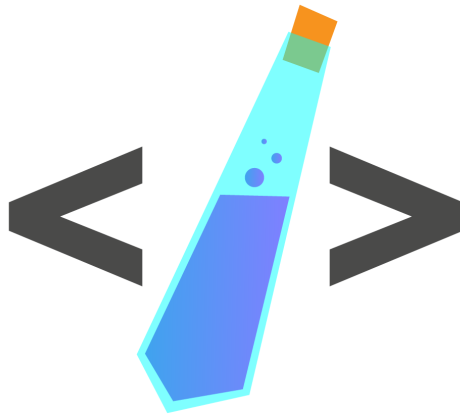




CODE ALCHEMISTS
TURNING IDEAS INTO CODE, LIKE MAGIC

Piano di Qualifica

Stato

Approvato

Versione

2.0.0

Distribuzione

Code Alchemists
Prof. Tullio Vardanega
Prof. Riccardo Cardin



Registro delle Modifiche

Vers.	Data	Autore	Verificatore	Descrizione
2.0.0	14/09/2025	A. Shu	S. Speranza	Approvazione del documento
1.4.0	14/09/2025	N. Moretto	S. Speranza	Aggiornamento dei grafici del Cruscotto di Valutazione
1.3.0	11/09/2025	N. Bolzon	N. Moretto	Completata sezione Test di Accettazione.
1.2.0	18/08/2025	N. Bolzon	M. Dalla Pozza	Aggiornamento dati dei grafici «valutazione dei costi» e «grafico valore atteso ed effettivo» relativi allo Sprint 10.
1.1.0	01/08/2025	N. Bolzon	S. Marana	Aggiornamento dati dei grafici «valutazione dei costi» e «grafico valore atteso ed effettivo» relativi allo Sprint 9.
1.0.0	17/07/2025	-	N. Bolzon	Verifica e Validazione del Documento
0.5.0	16/07/2025	S. Marana	N. Bolzon	Aggiunta test di sistema e grafici del Cruscotto di Valutazione
0.4.0	14/07/2025	N. Moretto	N. Bolzon	Aggiunta sezione 5 e relativi grafici
0.3.0	10/07/2025	S. Marana N. Moretto	N. Bolzon	Aggiunta sezione 3 e relative metriche di qualità per il prodotto ^G software
0.2.0	27/05/2025	S. Marana	R. Zangla	Aggiunta delle tabelle di continuo miglioramento e delle metriche di qualità per il processo di sviluppo
0.1.0	10/05/2025	S. Marana	N. Moretto	Compilazione base del Documento e creazione tabelle
0.0.1	22/04/2025	S. Marana	R. Zangla	Creazione template e struttura del documento

Indice

1. Introduzione	6
1.1 Scopo del documento	6
1.1.1 Piano della Qualità	6
1.1.2 Controllo di Qualità	6
1.2 Glossario	6
1.3 Riferimenti	6
1.3.1 Riferimenti normativi	6
1.3.2 Riferimenti informativi	6
2. Qualità di Processo	8
2.1 Processi Primari	8
2.1.1 Fornitura	8
2.1.2 Sviluppo	9
2.2 Processi di Supporto	9
2.2.1 Documentazione	9
2.2.2 Verifica	9
2.2.3 Gestione della Qualità	10
2.3 Processi Organizzativi	10
2.3.1 Gestione dei processi	10
3 Qualità di Prodotto	11
3.1 Funzionalità	11
3.2 Affidabilità	12
3.3 Manutenibilità	12
3.4 Efficienza	12
3.5 Usabilità	12
3.6 Portabilità	13
4. Metodologie di Testing	14
4.1 Test di unità	14
4.1.1 Microservizio di Autenticazione	14
4.1.2 Microservizio di Sistema Centralizzato	20
4.1.3 Microservizio di Stato Aggregato	32
4.1.4 Microservizio di Routing	35
4.1.5 Microservizio di Inventario Aggregato	41
4.1.6 Microservizio di Ordine Aggregato	45
4.1.7 Microservizio di Inventario	52
4.1.8 Microservizio di Ordini	58
4.1.9 Microservizio di State	73
4.2 Test di integrazione	77
4.3 Test di sistema	78
4.4 Test di accettazione	86
5. Cruscotto di Valutazione	89
5.1 Valutazione dei costi	89
5.2 Grafico valore atteso ed effettivo	89
5.3 Indice di prestazione dei tempi (tempo attuale e tempo pianificato per ogni sprint) .	90
5.4 Grafico indice di stabilità dei requisiti	91
5.5 Grafico indice di Gulpease	92

Indice Tabelle

Tabella 1	Metriche di qualità per il processo di fornitura	8
Tabella 2	Metriche di qualità per il processo di sviluppo	9
Tabella 3	Metriche di qualità per il processo di documentazione	9
Tabella 4	Metriche di qualità per il processo di verifica ^G	9
Tabella 5	Metriche di qualità per il processo di gestione della qualità	10
Tabella 6	Metriche di qualità per il processo di gestione dei processi	10
Tabella 7	Metriche per la funzionalità del prodotto software	11
Tabella 8	Metriche per l'affidabilità del prodotto software	12
Tabella 9	Metriche per la manutenibilità del prodotto software	12
Tabella 10	Metriche per l'efficienza del prodotto software	12
Tabella 11	Metriche per l'usabilità del prodotto software	12
Tabella 12	Metriche per la portabilità del prodotto software	13
Tabella 13	Test di unità di Autenticazione	14
Tabella 14	Test di unità di Sistema Centralizzato	20
Tabella 15	Test di unità di Stato Aggregato	32
Tabella 16	Test di unità di Routing	35
Tabella 17	Test di unità di Inventario Aggregato	41
Tabella 18	Test di unità di Ordine Aggregato	45
Tabella 19	Test di unità di Inventario	52
Tabella 20	Test di unità di Ordini	58
Tabella 21	Test di unità di State	73
Tabella 22	Test di integrazione	77
Tabella 23	Test di sistema	78
Tabella 24	Test di Accettazione	86

Indice Grafici

Figura 1	Valutazione dei costi	89
Figura 2	Valore atteso ed effettivo	90
Figura 3	Indice di prestazione dei tempi	91
Figura 4	Indice di stabilità dei requisiti	92
Figura 5	Indice di Gulpease	93

1. Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo di definire le modalità di qualifica e verifica del prodotto software. Esso rappresenta un indice di valutazione del prodotto software, delle sue componenti e degli strumenti utilizzati. L'obiettivo è quello di garantire la qualità del prodotto finale.

Il Piano di Qualifica^G si basa su tre aspetti principali:

1.1.1 Piano della Qualità

- Fissare gli standard del prodotto e i tipi di test da effettuare;
- Indicare gli obiettivi e le modalità di qualifica;
- Stabilire le regole a cui attenersi.

1.1.2 Controllo di Qualità

- Attuare e assicurare la conformità con le regole da seguire e il Way of Working^G ;
- L'operato viene monitorato attraverso il Cruscotto di Valutazione;
- Effettuare il Quality Assurance passo-passo per prevenire e mitigare i rischi e verificare la soddisfazione dei requisiti^G.

1.2 Glossario

Per una corretta comprensione del documento, viene fornito un glossario^G dei termini utilizzati. Ogni termine è indicato da una «^G» in apice alla parola. Per trovare il significato del termine, è possibile consultare il glossario^G al seguente indirizzo: <https://teamcodealchemists.github.io/glossario.html>

1.3 Riferimenti

1.3.1 Riferimenti normativi

- **Capitolato^G d'appalto**
C6 - Sistema di Gestione di un Magazzino Distribuito
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C6.pdf>
Ultimo Accesso: 14 Settembre 2025

1.3.2 Riferimenti informativi

- **Qualità del software**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T07.pdf>
Ultimo Accesso: 14 Settembre 2025
- **Qualità del processo**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T08.pdf>
Ultimo Accesso: 14 Settembre 2025
- **Verifica^G e Validazione^G**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T09.pdf>
Ultimo Accesso: 14 Settembre 2025
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T10.pdf>
Ultimo Accesso: 14 Settembre 2025
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T11.pdf>
Ultimo Accesso: 14 Settembre 2025

- **Glossario**^G
<https://teamcodealchemists.github.io/glossario.html>
Ultimo Accesso: 14 Settembre 2025

2. Qualità di Processo

La qualità di processo rappresenta l'insieme delle attività, delle metodologie e delle metriche adottate per garantire che lo sviluppo del prodotto^G software avvenga secondo standard definiti e controllabili. L'obiettivo è assicurare che ogni fase del ciclo di vita del progetto sia pianificata, monitorata e migliorata in modo continuo.

Le metriche di qualità di processo permettono di:

- Valutare l'efficacia e l'efficienza delle attività svolte nei diversi processi (primari, di supporto e organizzativi);
- Monitorare il rispetto degli standard e delle procedure stabilite nel Piano della Qualità;
- Fornire indicatori oggettivi utili per individuare aree di miglioramento e prevenire eventuali criticità.

L'adozione di un approccio sistematico e quantitativo consente di ridurre la soggettività nelle valutazioni e di supportare decisioni basate su dati concreti.

2.1 Processi Primari

I processi primari sono quelli che riguardano direttamente la realizzazione del prodotto software. Essi comprendono le attività di fornitura e sviluppo.

È stato deciso di utilizzare la seguente nomenclatura per identificare in modo univoco le metriche di qualità:

- **PPF**: Processi Primari di Fornitura;
- **PPS**: Processi Primari di Sviluppo.

Per ogni categoria, i codici definiti precedentemente sono seguiti da un numero progressivo che identifica la metrica specifica.

2.1.1 Fornitura

Codice	Nome	Valore Ammissibile	Valore Ottimale
PPF01	Valore atteso	≥ 0	$\geq \text{BAC}$
PPF02	Costo attuale	≥ 0	$\geq \text{EAC}$
PPF03	Valore effettivi	≥ 0	$\geq \text{EAC}$
PPF04	Indice di prestazione dei costi	≥ 0	1
PPF05	Indice di prestazione dei tempi	≥ 0	1
PPF06	Preventivo al completamento	≥ 0	$\geq \text{BAC}$
PPF07	Preventivo da completare	≥ 0	$\geq \text{BAC}$
PPF08	Tempo di completamento del preventivo	≥ 0	$\leq$ Durata pianificata

Tabella 1: Metriche di qualità per il processo di fornitura

2.1.2 Sviluppo

Codice	Nome	Valore Ammissibile	Valore Ottimale
PPS01	Indice di stabilità dei requisiti ^G	$\geq 70\%$	100%

Tabella 2: Metriche di qualità per il processo di sviluppo

2.2 Processi di Supporto

I processi di supporto sono quelli che non riguardano direttamente la realizzazione del prodotto software, ma che sono comunque necessari per il suo sviluppo. Essi comprendono le attività di documentazione, verifica e gestione della qualità. I processi di supporto sono identificati dalla seguente nomenclatura:

- **PSD**: Processi di Supporto alla Documentazione;
- **PSV**: Processi di Supporto alla Verifica;
- **PSQ**: Processi di Supporto alla Qualità.

Per ogni categoria, i codici definiti precedentemente sono seguiti da un numero progressivo che identifica la metrica specifica.

2.2.1 Documentazione

Codice	Nome	Valore Ammissibile	Valore Ottimale
PSD01	Indice di Gulpease	$\geq 50\%$	100%
PSD02	Parole ^G del glossario ^G contrassegnate con ^G	$\geq 60\%$	100%
PSD03	Glossario ^G ordinato	ordinato	ordinato

Tabella 3: Metriche di qualità per il processo di documentazione

2.2.2 Verifica

Codice	Nome	Valore Ammissibile	Valore Ottimale
PSV01	Code Coverage	$\geq 70\%$	$\geq 80\%$
PSV02	Grado di successo dei test	100%	100%

Tabella 4: Metriche di qualità per il processo di verifica^G

2.2.3 Gestione della Qualità

Codice	Nome	Valore Ammissibile	Valore Ottimale
PSQ01	Metriche di qualità soddisfatte	$\geq 70\%$	100%

Tabella 5: Metriche di qualità per il processo di gestione della qualità

2.3 Processi Organizzativi

I processi organizzativi sono quelli che riguardano la gestione del progetto e delle risorse. Essi comprendono le attività di pianificazione, monitoraggio^G e controllo del progetto, gestione delle risorse umane e gestione dei fornitori. I processi organizzativi sono identificati dalla seguente nomenclatura:

- **POG:** Gestione dei processi

Per ogni categoria, i codici definiti precedentemente sono seguiti da un numero progressivo che identifica la metrica specifica.

2.3.1 Gestione dei processi

Codice	Nome	Valore Ammissibile	Valore Ottimale
POG01	Efficienza in termini di tempo	$\geq 60\%$	100%

Tabella 6: Metriche di qualità per il processo di gestione dei processi

3 Qualità di Prodotto

La qualità di prodotto si riferisce alle caratteristiche e alle proprietà che rendono il software conforme alle aspettative degli stakeholder e agli standard prefissati. Essa viene valutata attraverso metriche oggettive che misurano aspetti come affidabilità, manutenibilità, efficienza e usabilità.

Punti chiave della qualità di prodotto:

- **Funzionalità:** capacità del software di fornire le funzioni richieste dagli utenti e di soddisfare i requisiti specificati nel documento di Analisi dei Requisiti^G.
- **Affidabilità:** capacità del software di funzionare senza errori in condizioni specificate per un periodo di tempo definito.
- **Manutenibilità:** facilità con cui il software può essere modificato per correggere difetti, migliorare prestazioni o adattarsi a nuovi requisiti.
- **Efficienza:** utilizzo ottimale delle risorse di sistema, come memoria e tempo di esecuzione.
- **Usabilità:** grado di facilità e soddisfazione con cui gli utenti possono utilizzare il prodotto.
- **Portabilità:** capacità del software di essere trasferito da un ambiente a un altro senza modifiche significative.

I tipi di processi di qualità di prodotto sono identificati dalla seguente nomenclatura:

- **QPF:** Qualità di Prodotto Funzionale;
- **QPA:** Qualità di Prodotto Affidabile;
- **QPM:** Qualità di Prodotto Manutenibile;
- **QPE:** Qualità di Prodotto Efficiente;
- **QPU:** Qualità di Prodotto Usabile;
- **QPP:** Qualità di Prodotto Portabile.

Per ogni categoria, i codici definiti precedentemente sono seguiti da un numero progressivo che identifica la metrica specifica.

3.1 Funzionalità

Codice	Nome	Valore Ammissibile	Valore Ottimale
QPF01	Requisiti obbligatori soddisfatti	100%	100%
QPF02	Requisiti desiderabili soddisfatti	$\geq 0\%$	100%
QPF03	Requisiti opzionali soddisfatti	$\geq 0\%$	100%

Tabella 7: Metriche per la funzionalità del prodotto software

3.2 Affidabilità

Codice	Nome	Valore Ammissibile	Valore Ottimale
QPA01	Branch coverage	$\geq 70\%$	$\geq 80\%$
QPA02	Statement coverage	$\geq 70\%$	$\geq 80\%$
QPA03	Failure density	≤ 0.5	≤ 0.1

Tabella 8: Metriche per l'affidabilità del prodotto software

3.3 Manutenibilità

Codice	Nome	Valore Ammissibile	Valore Ottimale
QPM01	Complessità ciclomatica	≤ 20	≤ 10
QPM02	Code smell rilevati	≤ 10	≤ 5
QPM03	Coefficient of coupling	≤ 0.5	≤ 0.3

Tabella 9: Metriche per la manutenibilità del prodotto software

3.4 Efficienza

Codice	Nome	Valore Ammissibile	Valore Ottimale
QPE01	Tempo di risposta medio	≤ 200 ms	≤ 100 ms
QPE02	Utilizzo della memoria	$\leq 80\%$	$\leq 60\%$
QPE03	Throughput	≥ 100 operazioni/s	≥ 200 operazioni/s

Tabella 10: Metriche per l'efficienza del prodotto software

3.5 Usabilità

Codice	Nome	Valore Ammissibile	Valore Ottimale
QPU01	Time on task	≤ 30 s	≤ 20 s
QPU02	Error rate	$\leq 5\%$	$\leq 2\%$

Tabella 11: Metriche per l'usabilità del prodotto software

3.6 Portabilità

Codice	Nome	Valore Ammissibile	Valore Ottimale
QPP01	Portability index	$\geq 70\%$	$\geq 80\%$
QPP02	Cross-platform compatibility	100%	100%

Tabella 12: Metriche per la portabilità del prodotto software

4. Metodologie di Testing

Per garantire la qualità del prodotto software, è fondamentale adottare metodologie di testing adeguate.

Ad ogni test è stato attribuito un codice univoco per facilitarne l'identificazione e la tracciabilità. Il codice è strutturato come segue: tipoXX, dove «tipo» rappresenta il tipo di test e «XX» è un numero progressivo che identifica il test specifico. I tipi di test sono i seguenti:

- **TdU**: Test di unità;
- **TdI**: Test di integrazione;
- **TdS**: Test di sistema;
- **TdA**: Test di accettazione.

Per ogni test viene specificato uno stato, che può essere:

- **NI**: Non Implementato;
- **IP**: In fase di Implementazione;
- **TS**: Test Superato;
- **TF**: Test Fallito.

4.1 Test di unità

I test di unità verificano il comportamento corretto delle singole unità di codice, come funzioni, metodi o classi. Vengono scritti e mantenuti dagli sviluppatori durante la fase di sviluppo e servono a garantire che ogni componente, isolato dal resto, produca il risultato atteso per diversi input. Sono fondamentali per individuare rapidamente errori logici a livello di codice sorgente.

Di seguito sono riportati i test di unità: per una migliore leggibilità, sono stati suddivisi in più tabelle, una per ogni microservizio.

4.1.1 Microservizio di Autenticazione

Codice	Descrizione	Valore atteso	Stato
TdU01	loginAccess dovrebbe restituire login,logout se global è impostato	ritorna login,logout	TS
TdU02	loginAccess dovrebbe restituire accessDenied se global non è impostato	ritorna accessDenied	TS
TdU03	registerAccess dovrebbe restituire globalSupervisor se global non è impostato	ritorna globalSupervisor	TS
TdU04	registerAccess dovrebbe restituire localSupervisor,globalSupervisor se global è impostato e token.isGlobal	ritorna localSupervisor,globalSupervisor	TS

Codice	Descrizione	Valore atteso	Stato
TdU05	registerAccess dovrebbe restituire get:false se global è impostato e token non è globale	ritorna get:false	TS
TdU06	registerAccess dovrebbe restituire accessDenied se token.error è presente	ritorna accessDenied	TS
TdU07	login dovrebbe chiamare inboundPortsAdapter.login e restituire il risultato	ritorna token	TS
TdU08	login dovrebbe gestire errori e restituire risposta di errore	ritorna system.internalError con messaggio fail	TS
TdU09	logout dovrebbe chiamare logout con SubDTO se token.sub non è presente	ritorna ok	TS
TdU10	logout dovrebbe chiamare logout con sub dal token	ritorna ok	TS
TdU11	logout dovrebbe gestire errori e restituire risposta di errore	ritorna system.internalError con messaggio logout error	TS
TdU12	registerGlobalSupervisor dovrebbe chiamare registerGlobalSupervisor e restituire il risultato	ritorna registered	TS
TdU13	registerGlobalSupervisor dovrebbe gestire errori e restituire risposta di errore	ritorna system.internalError con messaggio reg error	TS
TdU14	registerLocalSupervisor dovrebbe chiamare registerLocalSupervisor e restituire il risultato	ritorna registered	TS
TdU15	registerLocalSupervisor dovrebbe gestire errori e restituire risposta di errore	ritorna system.internalError con messaggio reg error	TS
TdU16	UserIdDTO dovrebbe creare un UserIdDTO con userId	userId impostato a 123	TS

Codice	Descrizione	Valore atteso	Stato
TdU17	findByEmail dovrebbe restituire GlobalSupervisor se isGlobal è true	ritorna istanza di GlobalSupervisor con nome Mario	TS
TdU18	findByEmail dovrebbe restituire LocalSupervisor se isGlobal è false	ritorna istanza di LocalSupervisor con nome Luigi	TS
TdU19	findByEmail dovrebbe restituire null se non trovato	ritorna null	TS
TdU20	getIdByEmail dovrebbe restituire id se trovato	ritorna id123	TS
TdU21	getIdByEmail dovrebbe restituire null se non trovato	ritorna null	TS
TdU22	newProfile dovrebbe lanciare eccezione se email esiste	lancia eccezione "Profile with this email already exists"	TS
TdU23	getGlobalSupervisor dovrebbe restituire GlobalSupervisor se trovato	ritorna istanza di GlobalSupervisor con nome Mario	TS
TdU24	getGlobalSupervisor dovrebbe restituire null se non trovato	ritorna null	TS
TdU25	getAllProfiles dovrebbe restituire array di utenti	ritorna array con GlobalSupervisor e LocalSupervisor	TS
TdU26	storeToken dovrebbe salvare il token	save del token chiamato	TS
TdU27	getToken dovrebbe restituire Token se trovato	ritorna istanza di Token con sub=sub123 e status=ACTIVE	TS
TdU28	getToken dovrebbe restituire null se non trovato	ritorna null	TS
TdU29	updateToken dovrebbe aggiornare lo status del token	chiamato updateOne con status=REVOKED	TS
TdU30	login dovrebbe funzionare con credenziali valide	ritorna "Login successful"	TS

Codice	Descrizione	Valore atteso	Stato
TdU31	login fallisce se email non esiste	ritorna "system.accessDenied: Email does not exist"	TS
TdU32	login fallisce se password non valida	ritorna "system.accessDenied: Password is not valid"	TS
TdU33	login fallisce se già loggato	ritorna "system.accessDenied: already logged in"	TS
TdU34	logout dovrebbe funzionare correttamente	ritorna "Logout successful"	TS
TdU35	logout fallisce se token è null	ritorna errore "You are already logged out"	TS
TdU36	logout fallisce se token è già revocato	ritorna "Token has already been Revoked"	TS
TdU37	authenticate dovrebbe autenticare ed emettere token	ritorna "«result»:null" e chiama emitAccessToken	TS
TdU38	authenticate fallisce se JWT non fornito	ritorna "No JWT provided"	TS
TdU39	authenticate fallisce se token è loggato out	ritorna "Token has been logged out"	TS
TdU40	authenticate fallisce se verifica JWT fallisce	ritorna errore "invalid jwt" e chiama updateToken e emitAccessToken	TS
TdU41	registerGlobalSupervisor registra se non già impostato	ritorna "Global Supervisor registered successfully"	TS
TdU42	registerGlobalSupervisor fallisce se già impostato	ritorna "system.accessDenied: already signed in"	TS
TdU43	registerLocalSupervisor registra con successo	ritorna "Local Supervisor registered successfully"	TS
TdU44	registerLocalSupervisor fallisce se throw errore	ritorna "system.accessDenied: fail"	TS
TdU45	isGlobalSet ritorna true se global supervisor esiste	ritorna true	TS

Codice	Descrizione	Valore atteso	Stato
TdU46	isGlobalSet ritorna false se global supervisor non esiste	ritorna false	TS
TdU47	onModuleInit dovrebbe chiamare natsClient.connect	natsClient.connect chiamato	TS
TdU48	emitAccessToken dovrebbe chiamare natsClient.emit con topic e token	natsClient.emit chiamato con "conn.cid456.token" e "token123"	TS
TdU49	globalSupervisorToDomain mappa DTO in entità dominio	domain è istanza di GlobalSupervisor con proprietà corrette	TS
TdU50	localSupervisorToDomain mappa DTO in entità dominio	domain è istanza di LocalSupervisor con proprietà corrette	TS
TdU51	warehouseToDomain mappa DTO in entità dominio	domain è istanza di WarehouseId con id corretto	TS
TdU52	authenticationToDomain mappa DTO in entità dominio	domain è istanza di Authentication con email e password corrette	TS
TdU53	globalSupervisorToDTO mappa entità dominio in DTO	dto ha nome, cognome, telefono e authentication corrette	TS
TdU54	localSupervisorToDTO mappa entità dominio in DTO	dto ha nome, cognome, telefono, authentication e warehouseAssigned corrette	TS
TdU55	warehouseIdToDTO mappa entità dominio in DTO	dto ha warehouseId corretto	TS
TdU56	authenticationToDTO mappa entità dominio in DTO	dto ha email e password corrette	TS
TdU57	userIdToDTO mappa entità dominio in DTO	dto ha userId corretto	TS
TdU58	L'entità Authentication deve memorizzare email e password	email e password correttamente memorizzate	TS
TdU59	L'entità GlobalSupervisor deve memorizzare nome, cognome, telefono e authentication	i dati vengono correttamente memorizzati	TS

Codice	Descrizione	Valore atteso	Stato
TdU60	L'entità LocalSupervisor deve memorizzare nome, cognome, telefono, authentication e magazzini	i dati vengono correttamente memorizzati	TS
TdU61	L'entità Role deve avere valori statici	Role.GLOBAL e Role.LOCAL definiti	TS
TdU62	L'entità Token deve memorizzare sub e status	sub e status correttamente memorizzati	TS
TdU63	L'entità TokenStatus deve avere valori statici	TokenStatus.ACTIVE e REVOKED definiti	TS
TdU64	L'entità UserId deve memorizzare l'id	id correttamente memorizzato	TS
TdU65	L'entità WarehouseId deve memorizzare warehouseId	id del magazzino correttamente memorizzato	TS
TdU66	login chiama authService.login con il DTO mappato	DTO mappato passato e token restituito	TS
TdU67	logout chiama authService.logout con sub	sub passato e risposta ok	TS
TdU68	authenticate chiama authService.authenticate con jwt e cid	jwt e cid passati e autenticazione restituita	TS
TdU69	registerGlobalSupervisor chiama authService.registerGlobalSupervisor con DTO mappato	DTO mappato passato e registrazione riuscita	TS
TdU70	registerLocalSupervisor chiama authService.registerLocalSupervisor con DTO mappato	DTO mappato passato e registrazione riuscita	TS
TdU71	estrazione JWT e chiamata ad authenticate	JWT estratto e chiamata ad authenticate restituisce auth-ok	TS
TdU72	gestione header Authorization come array	JWT estratto da array e chiamata ad authenticate restituisce auth-ok	TS

Codice	Descrizione	Valore atteso	Stato
TdU73	restituisce errore se authenticate fallisce	eccezione di authenticate catturata, errore restituito	TS
TdU74	restituisce errore se manca token Bearer	nessun Bearer token, accesso negato restituito	TS
TdU75	emitAccessToken chiama la porta con token e cid	token e cid passati correttamente	TS
TdU76	AuthenticationSchema should be defined	schema di Authentication definito	TS
TdU77	AuthenticationSchema should have expected fields	campi definiti siano presenti	TS
TdU78	tokenListSchema should be defined	schema tokenList definito	TS
TdU79	tokenListSchema should have expected fields	campi sub e status presenti	TS

Tabella 13: Test di unità di Autenticazione

4.1.2 Microservizio di Sistema Centralizzato

Codice	Descrizione	Valore atteso	Stato
TdU80	Dovrebbe connettersi all'inizializzazione del modulo	connect chiamato	TS
TdU81	Dovrebbe emettere una notifica	emit chiamato	TS
TdU82	Dovrebbe emettere un ordine interno	emit chiamato	TS
TdU83	Dovrebbe inviare una richiesta di inventario al cloud e restituire un Inventory	send chiamato + ritorno Inventory	TS
TdU84	Dovrebbe inviare una richiesta di ordini al cloud e restituire un Orders o null	send chiamato + ritorno null	TS

Codice	Descrizione	Valore atteso	Stato
TdU85	Dovrebbe richiedere la distanza dei magazzini e restituire un array di WarehouseId[]	send chiamato, e ritorno WarehouseId[]	TS
TdU86	Dovrebbe compilare il modulo	module definito	TS
TdU87	Dovrebbe risolvere CentralSystemService	istanza di CentralSystemService	TS
TdU88	Dovrebbe risolvere centralSystemController	istanza di centralSystemController	TS
TdU89	Dovrebbe risolvere InboundPortsAdapter	istanza di InboundPortsAdapter	TS
TdU90	Dovrebbe risolvere OutboundPortsAdapter	istanza di OutboundPortsAdapter	TS
TdU91	Dovrebbe risolvere centralSystemHandler	istanza di centralSystemHandler	TS
TdU92	Dovrebbe creare un ordine interno se residualQty >= minThres	createInternalOrder chiamato	TS
TdU93	Non dovrebbe creare un ordine interno se availableQty < minThres	createInternalOrder non chiamato	TS
TdU94	Non dovrebbe creare un ordine interno se residualQty < minThres	createInternalOrder non chiamato	TS
TdU95	Non dovrebbe creare un ordine interno se il prodotto non è in inventario	createInternalOrder non chiamato	TS
TdU96	Dovrebbe creare ordini interni se i prodotti hanno quantità sufficiente	createInternalOrder chiamato	TS
TdU97	Non dovrebbe creare un ordine interno se availableQty < minThres	createInternalOrder non chiamato	TS

Codice	Descrizione	Valore atteso	Stato
TdU98	Non dovrebbe creare un ordine interno se residualQty < minThres	createInternalOrder non chiamato	TS
TdU99	Non dovrebbe creare un ordine interno se il prodotto richiesto non è nei magazzini	createInternalOrder non chiamato	TS
TdU100	Dovrebbe inviare una notifica per magazzini inattivi	SendNotification chiamato	TS
TdU101	Non dovrebbe fare nulla se tutti i magazzini sono attivi	SendNotification non chiamato	TS
TdU102	Non dovrebbe fare nulla se la lista magazzini è vuota	SendNotification non chiamato	TS
TdU103	Dovrebbe creare un ordine interno se residualQty <= maxThres	createInternalOrder chiamato	TS
TdU104	Non dovrebbe creare un ordine interno se availableQty > maxThres	createInternalOrder non chiamato	TS
TdU105	Non dovrebbe creare un ordine interno se residualQty > maxThres	createInternalOrder non chiamato	TS
TdU106	Non dovrebbe creare un ordine interno se il prodotto non è in inventario	createInternalOrder non chiamato	TS
TdU107	Dovrebbe recuperare inventario, ordini e stati dei magazzini e mapparli al dominio	CloudInventoryRequest, CloudOrderRequest e RequestDistanceWarehouse chiamati + ritorno con proprietà inv, order, dist	TS
TdU108	Dovrebbe trasformare il payload e chiamare handleInsufficientQuantity	handleInsufficientQuantity chiamato + corretta conversione di OrderQuantityDTO e WarehouseIdDTO	TS

Codice	Descrizione	Valore atteso	Stato
TdU109	Dovrebbe trasformare il payload e chiamare handleCriticalQuantityMin	handleCriticalQuantityMin chiamato + corretta conversione di ProductDTO	TS
TdU110	Dovrebbe trasformare il payload e chiamare handleCriticalQuantityMax	handleCriticalQuantityMax chiamato + corretta conversione di ProductDTO	TS
TdU111	Dovrebbe trasformare il payload e chiamare getWarehouseState	getWarehouseState chiamato + trasformazione in array di WarehouseStateDTO corretti	TS
TdU112	Dovrebbe chiamare handleNotification	handleNotification chiamato con messaggio	TS
TdU113	Dovrebbe convertire InternalOrder in DTO e chiamare handleOrder	internalOrderToDTO chiamato + handleOrder chiamato	TS
TdU114	Dovrebbe chiamare handler e mappare l'inventario in DTO	handleCloudInventoryRequest chiamato + toDtoInventory chiamato	TS
TdU115	Dovrebbe chiamare handler e mappare gli ordini in DTO	handleCloudOrdersRequest chiamato + ordersToDTO chiamato	TS
TdU116	Dovrebbe restituire null se handler ritorna null	handleCloudOrdersRequest chiamato + ritorno null	TS
TdU117	Dovrebbe chiamare handler, mappare in DTO e restituire array DTO	handleWarehouseDistance chiamato + ritorno array DTO corretti	TS
TdU118	Dovrebbe chiamare handleRequestOrdResult	handleRequestOrdResult chiamato con messaggio, orderId, warehouseId	TS
TdU119	Dovrebbe chiamare handleRequestInvResult	handleRequestInvResult chiamato con messaggio, productId, warehouseId	TS
TdU120	Dovrebbe mappare productDto in Product	Product creato con nome, quantità e warehouseId coerenti	TS

Codice	Descrizione	Valore atteso	Stato
TdU121	Dovrebbe mappare Product in productDto	DTO con nome, unitPrice, quantity e warehouseId coerenti	TS
TdU122	Dovrebbe mappare WarehouseId in warehouseIdDto	DTO con warehouseId coerente	TS
TdU123	internalOrderToDomain deve lanciare errore se warehouseDeparture = warehouseDestination	eccezione lanciata con messaggio corretto	TS
TdU124	orderIdToDomain deve lanciare errore su formato non valido	eccezione lanciata con messaggio corretto	TS
TdU125	orderStateToDomain deve lanciare errore su stato non valido	eccezione lanciata con messaggio corretto	TS
TdU126	orderItemDetailToDomain deve lanciare errore se reserved > ordered	eccezione lanciata con messaggio corretto	TS
TdU127	orderQuantityToDomain deve mappare DTO in dominio	OrderQuantity creato con ID e items coerenti	TS
TdU128	warehouseIdToDomain deve lanciare errore se DTO invalido	eccezione lanciata con messaggio corretto	TS
TdU129	warehouseStatetoDto deve mappare WarehouseState in DTO	DTO con state e warehouseId coerenti	TS
TdU130	toDomainProduct deve mappare productDto in Product	Product creato con nome, quantità e warehouseId coerenti	TS
TdU131	toDtoProduct deve mappare Product in productDto	DTO con id, nome, unitPrice, quantity e warehouseId coerenti	TS
TdU132	toBelowMinDTO deve mappare Product in belowMinThresDto	DTO con id, quantity e minThres coerenti	TS

Codice	Descrizione	Valore atteso	Stato
TdU133	toAboveMaxDTO deve mappare Product in aboveMaxThresDto	DTO con id, quantity e maxThres coerenti	TS
TdU134	toDTOProductQuantity deve mappare ProductId + quantity in DTO	DTO con productId e quantity coerenti	TS
TdU135	orderItemToDTO deve mappare OrderItem in DTO	DTO con itemId e quantity coerenti	TS
TdU136	orderIdToDTO deve mappare OrderId in DTO	DTO con id coerente	TS
TdU137	orderStateToDTO deve mappare OrderState in DTO	DTO con orderState coerente	TS
TdU138	orderItemDetailToDTO deve mappare OrderItemDetail in DTO	DTO con itemId, quantityReserved e unitPrice coerenti	TS
TdU139	orderQuantityToDTO deve mappare OrderQuantity in DTO	DTO con id e items coerenti	TS
TdU140	productQuantityToDTO deve mappare entity in productQuantityDto	DTO con productId e quantity coerenti	TS
TdU141	warehouseIdToDomain deve mappare DTO in WarehouseId	WarehouseId creato con ID coerente	TS
TdU142	Dovrebbe mappare DTO in dominio e chiamare CheckInsufficientQuantity	orderQuantityToDomain e warehouseIdToDomain chiamati + CheckInsufficientQuantity con oggetti dominio	TS
TdU143	Dovrebbe mappare DTO in dominio e chiamare ManageCriticalMinThres	toDomainProduct chiamato + ManageCriticalMinThres con prodotto dominio	TS
TdU144	Dovrebbe mappare DTO in dominio e chiamare ManageOverMaxThres	toDomainProduct chiamato + ManageOverMaxThres con prodotto dominio	TS

Codice	Descrizione	Valore atteso	Stato
TdU145	Dovrebbe mappare DTOs in dominio e chiamare CheckWarehouseState	warehouseStatetoDomain chiamato + CheckWarehouseState con array dominio	TS
TdU146	Dovrebbe creare il microservice con opzioni corrette e applicare ValidationPipe	NestFactory.createMicroservice chiamato con opzioni corrette + useGlobalPipes applicato con ValidationPipe + listen chiamato	TS
TdU147	Dovrebbe usare RpcException in ValidationPipe exceptionFactory	exceptionFactory ritorna RpcException con errori di validazione	TS
TdU148	Dovrebbe inizializzare correttamente le proprietà di InternalOrder	orderId, itemsDetail, orderState, warehouseDeparture e warehouseDestination valorizzati correttamente	TS
TdU149	Dovrebbe aggiornare warehouseDestination usando setWarehouseDestination	warehouseDestination aggiornato al valore passato	TS
TdU150	Dovrebbe permettere di aggiornare orderState	orderState aggiornato correttamente al nuovo valore	TS
TdU151	Dovrebbe restituire tutti i prodotti tramite getInventory	getInventory ritorna array con tutti i prodotti presenti	TS
TdU152	Dovrebbe aggiungere un nuovo prodotto tramite addProductItem	prodotto aggiunto correttamente all'inventario e presente in getInventory	TS
TdU153	Dovrebbe restituire il corretto id tramite getId di ItemId	getId ritorna l'id corretto	TS
TdU154	Dovrebbe memorizzare correttamente diversi id	getId ritorna valori coerenti per item1 e item2	TS
TdU155	Dovrebbe restituire il corretto orderId	getOrderId ritorna l'id corretto	TS
TdU156	Dovrebbe ottenere e settare gli items di Order	setItemsDetail aggiorna correttamente gli items e getItemDetail li restituisce	TS

Codice	Descrizione	Valore atteso	Stato
TdU157	Dovrebbe ottenere e settare lo stato dell'ordine	setOrderState aggiorna lo stato e getOrderState lo restituisce	TS
TdU158	Dovrebbe ottenere e settare la data di creazione dell'ordine	setCreationDate aggiorna la data e getCreationDate la restituisce	TS
TdU159	Dovrebbe ottenere e settare warehouseDeparture	setWarehouseDeparture aggiorna il valore e getWarehouseDeparture lo restituisce	TS
TdU160	Dovrebbe restituire l'OrderItem corretto da OrderItemDetail	getItem ritorna l'oggetto OrderItem corretto	TS
TdU161	Dovrebbe restituire la quantityReserved corretta	getQuantityReserved ritorna il valore corretto	TS
TdU162	Dovrebbe restituire l'unitPrice corretto	getUnitPrice ritorna il valore corretto	TS
TdU163	Dovrebbe aggiornare quantityReserved	setQuantityReserved aggiorna il valore e getQuantityReserved lo restituisce	TS
TdU164	Dovrebbe aggiornare unitPrice	setUnitPrice aggiorna il valore e getUnitPrice lo restituisce	TS
TdU165	Dovrebbe restituire gli items corretti di OrderQuantity	getItemId ritorna array corretto degli items	TS
TdU166	Dovrebbe restituire le quantità corrette	getQuantity ritorna array corretto delle quantità	TS
TdU167	Dovrebbe aggiornare la quantità per un item specifico	setQuantity aggiorna il valore corretto e getQuantity lo restituisce	TS
TdU168	Dovrebbe restituire il corretto id dell'ordine	getId ritorna l'ID corretto	TS
TdU169	Dovrebbe restituire il corretto tipo di ordine	getOrderType ritorna il tipo corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU170	Dovrebbe aggiornare sellOrders quando setSellOrders è chiamato	sellOrders aggiornati correttamente e diversi dai precedenti	TS
TdU171	Dovrebbe aggiornare internalOrders quando setInternalOrders è chiamato	internalOrders aggiornati correttamente e diversi dai precedenti	TS
TdU172	Dovrebbe restituire il nome corretto tramite getName	getName ritorna il nome iniziale del prodotto	TS
TdU173	Dovrebbe aggiornare il nome con setName	setName aggiorna il nome e getName lo restituisce	TS
TdU174	Dovrebbe aggiornare unitPrice con setUnitPrice	setUnitPrice aggiorna il valore e getUnitPrice lo restituisce	TS
TdU175	Dovrebbe aggiornare quantity con setQuantity	setQuantity aggiorna il valore e getQuantity lo restituisce	TS
TdU176	Dovrebbe aggiornare minThres con setMinThres	setMinThres aggiorna il valore e getMinThres lo restituisce	TS
TdU177	Dovrebbe aggiornare maxThres con setMaxThres	setMaxThres aggiorna il valore e getMaxThres lo restituisce	TS
TdU178	Dovrebbe restituire il corretto ProductId da ProductQuantity	getId ritorna l'id corretto	TS
TdU179	Dovrebbe restituire la quantità corretta da ProductQuantity	getQuantity ritorna il valore corretto	TS
TdU180	Dovrebbe restituire correttamente l'indirizzo di destinazione di SellOrder	getDestinationAddress ritorna l'indirizzo corretto	TS
TdU181	Dovrebbe aggiornare l'indirizzo di destinazione di SellOrder	setDestinationAddress aggiorna l'indirizzo e getDestinationAddress lo restituisce	TS
TdU182	Dovrebbe restituire il corretto orderId di SellOrder	getOrderId ritorna l'id corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU183	Dovrebbe restituire correttamente gli items di SellOrder	getItemsDetail ritorna l'array corretto di items	TS
TdU184	Dovrebbe restituire il corretto stato dell'ordine di SellOrder	getOrderState ritorna lo stato corretto	TS
TdU185	Dovrebbe restituire correttamente la data di creazione di SellOrder	getCreationDate ritorna la data corretta	TS
TdU186	Dovrebbe restituire correttamente warehouseDeparture di SellOrder	getWarehouseDeparture ritorna il valore corretto	TS
TdU187	Dovrebbe validare un DTO aboveMaxThres corretto	validate ritorna zero errori per un DTO valido	TS
TdU188	Dovrebbe fallire se id non è un UUID valido in aboveMaxThresDto	validate ritorna errore sulla proprietà id	TS
TdU189	Dovrebbe fallire se quantity è negativa in aboveMaxThresDto	validate ritorna errore sulla proprietà quantity	TS
TdU190	Dovrebbe fallire se maxThres è negativa in aboveMaxThresDto	validate ritorna errore sulla proprietà maxThres	TS
TdU191	Dovrebbe validare un DTO belowMinThres corretto	validate ritorna zero errori per un DTO valido	TS
TdU192	Dovrebbe fallire se id non è un UUID valido in belowMinThresDto	validate ritorna errore sulla proprietà id	TS
TdU193	Dovrebbe fallire se quantity è negativa in belowMinThresDto	validate ritorna errore sulla proprietà quantity	TS
TdU194	Dovrebbe fallire se minThres è negativa in belowMinThresDto	validate ritorna errore sulla proprietà minThres	TS

Codice	Descrizione	Valore atteso	Stato
TdU195	Dovrebbe validare un InternalOrderDTO corretto	validate ritorna zero errori per un DTO valido	TS
TdU196	Dovrebbe fallire se orderId manca in InternalOrderDTO	validate ritorna errore sulla proprietà orderId	TS
TdU197	Dovrebbe fallire se items non è un array in InternalOrderDTO	validate ritorna errore sulla proprietà items	TS
TdU198	Dovrebbe fallire se warehouseDeparture non è un intero in InternalOrderDTO	validate ritorna errore sulla proprietà warehouseDeparture	TS
TdU199	Dovrebbe validare un inventoryDto corretto	validate ritorna zero errori per un DTO valido	TS
TdU200	Dovrebbe fallire se productList non è un array in inventoryDto	validate ritorna errore sulla proprietà productList	TS
TdU201	Dovrebbe fallire se productList contiene productDto non valido	validate ritorna errori per i prodotti invalidi	TS
TdU202	Dovrebbe validare un OrderItemDetailDTO corretto	validate ritorna zero errori per un DTO valido	TS
TdU203	Dovrebbe fallire se quantityReserved è negativo in OrderItemDetailDTO	validate ritorna errore sulla proprietà quantityReserved	TS
TdU204	Dovrebbe fallire se unitPrice ha più di due decimali in OrderItemDetailDTO	validate ritorna errore sulla proprietà unitPrice	TS
TdU205	Dovrebbe fallire se item manca in OrderItemDetailDTO	validate ritorna errore sulla proprietà item	TS
TdU206	Dovrebbe validare un OrdersDTO corretto	validate ritorna zero errori per un OrdersDTO valido	TS
TdU207	Dovrebbe validare un productQuantityDto corretto	validate ritorna zero errori per un productQuantityDto valido	TS

Codice	Descrizione	Valore atteso	Stato
TdU208	Dovrebbe fallire se quantity è negativa in productQuantityDto	validate ritorna errore sulla proprietà quantity	TS
TdU209	Dovrebbe validare un productQuantityArrayDto corretto	validate ritorna zero errori per un productQuantityArrayDto valido	TS
TdU210	Dovrebbe fallire se productQuantityArray non è un array	validate ritorna errore sulla proprietà productQuantityArray	TS
TdU211	Dovrebbe fallire se un elemento di productQuantityArray è invalido	validate ritorna errori per elementi invalidi	TS
TdU212	Dovrebbe validare un SellOrderDTO corretto	validate ritorna zero errori per un SellOrderDTO valido	TS
TdU213	Dovrebbe fallire se items non è un array in SellOrderDTO	validate ritorna errore sulla proprietà items	TS
TdU214	Dovrebbe fallire se destinationAddress è vuoto in SellOrderDTO	validate ritorna errore sulla proprietà destinationAddress	TS
TdU215	Dovrebbe validare un WarehouseStateDTO corretto	validate ritorna zero errori per un WarehouseStateDTO valido	TS
TdU216	Dovrebbe fallire se warehouseId manca in WarehouseStateDTO	validate ritorna errore sulla proprietà warehouseId	TS
TdU217	Dovrebbe fallire se warehouseId è invalido in WarehouseStateDTO	validate ritorna errore sulla proprietà warehouseId	TS
TdU218	Dovrebbe emettere l'evento corretto per handleRequestInvResult	natsClient.emit chiamato e message	TS
TdU219	Dovrebbe emettere l'evento corretto per handleRequestOrdResult	natsClient.emit chiamato e message	TS

Tabella 14: Test di unità di Sistema Centralizzato

4.1.3 Microservizio di Stato Aggregato

Codice	Descrizione	Valore atteso	Stato
TdU220	publishHeartbeat chiama outboundService.publishHeartbeat con DTO	publishHeartbeat invocato correttamente	TS
TdU221	publishState chiama outboundService.publishState con DTO	publishState invocato correttamente	TS
TdU222	stateUpdated chiama outboundService.stateUpdated con DTO	stateUpdated invocato correttamente	TS
TdU223	getState restituisce CloudWarehouseState se trovato	ritorna stato ONLINE	TS
TdU224	getState restituisce null se non trovato	ritorna null	TS
TdU225	updateState restituisce true se acknowledged	ritorna true	TS
TdU226	updateState restituisce false se non acknowledged	ritorna false	TS
TdU227	getAllWarehouseIds restituisce array di CloudWarehouseId	ritorna array di ID	TS
TdU228	getState chiama il repository	repository invocato e ritorna ONLINE	TS
TdU229	updateState chiama il repository	repository invocato e ritorna true	TS
TdU230	handleHeartbeatResponse chiama tutte le callback e le pulisce	callback invocate e lista svuotata	TS
TdU231	checkHeartbeat imposta lo stato ONLINE se isAlive è true	ritorna ONLINE	TS

Codice	Descrizione	Valore atteso	Stato
TdU232	checkHeartbeat imposta lo stato OFFLINE se non arriva risposta	ritorna OFFLINE	TS
TdU233	startPeriodicHeartbeatCheck chiama checkHeartbeat per ogni magazzino	checkHeartbeat chiamato per tutti gli ID	TS
TdU234	chiama handleHeartbeatResponse con i parametri corretti	ritorna OK	TS
TdU235	gestisce errori e restituisce un JSON di errore	ritorna errore system.invalidParams	TS
TdU236	updateState aggiorna lo stato se diverso e chiama notifyStateUpdated	ritorna Address updated successfully	TS
TdU237	updateState non aggiorna lo stato se uguale	ritorna Address updated successfully	TS
TdU238	updateState gestisce errori e restituisce un JSON di errore	ritorna errore system.invalidParams	TS
TdU239	getState restituisce lo stato e chiama publishState	ritorna stato ONLINE	TS
TdU240	getState restituisce stringa se non trova lo stato	ritorna No state found	TS
TdU241	getState gestisce errori e restituisce un JSON di errore	ritorna errore system.invalidParams	TS
TdU242	chiama logger.verbose con il messaggio corretto	logger.verbose invocato con messaggio e dati	TS
TdU243	restituisce un oggetto con pattern, data e id	ritorna oggetto con data e id stringa	TS
TdU244	cloudWarehouseIdToDomain converte DTO in Domain	ritorna CloudWarehouseId	TS
TdU245	cloudWarehouseStateToDomain converte DTO in Domain	ritorna CloudWarehouseState ONLINE	TS

Codice	Descrizione	Valore atteso	Stato
TdU246	cloudHeartbeatToDomain converte DTO in Domain	ritorna CloudHeartbeat ALIVE	TS
TdU247	cloudWarehouseIdToDTO converte Domain in DTO	ritorna DTO con warehouseId	TS
TdU248	cloudWarehouseStateToDTO converte Domain in DTO	ritorna DTO con warehouseId e state	TS
TdU249	cloudHeartbeatToDTO converte Domain in DTO	ritorna DTO con heartbeatMsg e timestamp	TS
TdU250	publishHeartbeat chiama natsService.publish con il topic corretto	natsService.publish invocato con heartbeat	TS
TdU251	publishState chiama natsService.publish con il topic corretto	natsService.publish invocato con state JSON	TS
TdU252	stateUpdated chiama natsService.publish con il topic corretto	natsService.publish invocato con state	TS
TdU253	CloudWarehouseStateDTO valida un DTO corretto	nessun errore	TS
TdU254	CloudWarehouseStateDTO non valida se warehouseId è negativo	errore su warehouseId	TS
TdU255	CloudWarehouseStateDTO non valida se warehouseId è mancante	errore su warehouseId	TS
TdU256	CloudWarehouseStateDTO non valida se state è mancante	errore su state	TS
TdU257	CloudWarehouseStateDTO non valida se state non è una stringa	errore su state	TS
TdU258	crea la microservice app con le opzioni corrette	microservice creato con NATS e server configurato	TS

Codice	Descrizione	Valore atteso	Stato
TdU259	deve avere la struttura corretta	config NATS contiene tutte le proprietà richieste	TS
TdU260	deve usare NATS_URL se presente	config NATS usa URL custom	TS
TdU261	deve usare il valore di default se NATS_URL non è presente	config NATS usa URL di default	TS
TdU262	onModuleInit inizializza la connessione NATS	connessione NATS inizializzata	TS
TdU263	onModuleDestroy chiude la connessione NATS	connessione NATS chiusa	TS
TdU264	publish invia il messaggio codificato	messaggio codificato e pubblicato	TS
TdU265	serializza correttamente un oggetto con proprietà data	ritorna oggetto con proprietà data	TS
TdU266	serializza un oggetto senza proprietà data	ritorna oggetto con data undefined	TS
TdU267	logga la serializzazione	logger.verbose chiamato con messaggi di serializzazione	TS
TdU268	serializza una risposta con response	ritorna data con response e loggata	TS
TdU269	serializza una risposta con err come stringa	ritorna error JSON e loggata	TS
TdU270	serializza una risposta con err come array di errori di validazione	ritorna error JSON concatenato e loggata	TS
TdU271	serializza una risposta senza response né err	ritorna data originale e loggata	TS

Tabella 15: Test di unità di Stato Aggregato

4.1.4 Microservizio di Routing

Codice	Descrizione	Valore atteso	Stato
TdU272	deve chiamare outboundService.sendWarehouseAndState con il DTO corretto	chiamata a outboundService	TS
TdU273	Dovrebbe restituire un output valorizzato quando l'aggiornamento di un indirizzo di magazzino ha successo	result è true	TS
TdU274	Dovrebbe restituire false quando l'aggiornamento di un indirizzo di magazzino fallisce	result è false	TS
TdU275	Dovrebbe restituire un output valorizzato quando la rimozione di un indirizzo di magazzino ha successo	result è true	TS
TdU276	Dovrebbe restituire un output false quando la rimozione di un indirizzo di magazzino fallisce	result è false	TS
TdU277	Dovrebbe restituire un output valorizzato quando l'aggiornamento dello stato di un magazzino ha successo	result è true	TS
TdU278	Dovrebbe restituire false quando l'aggiornamento dello stato di un magazzino fallisce	result è false	TS
TdU279	Dovrebbe restituire un output valorizzato quando il salvataggio di un magazzino ha successo	result è true	TS
TdU280	Dovrebbe restituire false quando il salvataggio di un magazzino fallisce	result è false	TS
TdU281	Dovrebbe restituire l'array di magazzini in ordine per distanza più vicina al magazzino richiedente	array contenente WarehouseId	TS

Codice	Descrizione	Valore atteso	Stato
TdU282	equals ritorna true se gli id sono uguali	true	TS
TdU283	equals ritorna false se gli id sono diversi	false	TS
TdU284	serializza una risposta con response	result = { data: { foo: "bar" } }	TS
TdU285	serializza una risposta con err come stringa	result.data contiene error con code "system.invalidParams" e message	TS
TdU286	serializza una risposta con err come array di errori di validazione	result.data contiene error con code "system.invalidParams" e message	TS
TdU287	serializza una risposta senza response né err	result = { data: "test" }	TS
TdU288	deve avere la struttura corretta	natsConfig con proprietà e valori attesi	TS
TdU289	deve usare NATS_URL se presente	servers[0] = "nats://custom:4222"	TS
TdU290	deve usare il valore di default se NATS_URL non è presente	servers[0] = "nats://localhost:4222"	TS
TdU291	onModuleInit inizializza la connessione NATS	service["nc"] definito	TS
TdU292	onModuleDestroy chiude la connessione NATS	close chiamato	TS
TdU293	publish invia il messaggio codificato	publish chiamato con subject e buffer	TS
TdU294	crea la microservice app con le opzioni corrette	NestFactory.createMicroservice chiamato con RoutingModule e opzioni	TS
TdU295	warehouseIdToDomain converte DTO in entity	entity è WarehouseId, id = 42	TS

Codice	Descrizione	Valore atteso	Stato
TdU296	warehouseStateToDomain converte DTO in entity	entity è WarehouseState, id = 7, state = "ATTIVO"	TS
TdU297	warehouseAddressToDomain converte DTO in entity	entity è WarehouseAddress, id = 5, state = "DISATTIVO", address = "Via Roma 5"	TS
TdU298	warehouseAddressToDTO converte entity in DTO	DTO con warehouseState e address	TS
TdU299	warehouseIdToDTO converte entity in DTO	DTO con warehouseId	TS
TdU300	warehouseStateToDTO con- verte entity in DTO	DTO con warehouseId e state	TS
TdU301	sendAddress chiama natsService.publish con il to- pic e il payload giusto	publish chiamato con "warehouse.address" e address	TS
TdU302	sendWarehouseDistance chia- ma natsService.publish con il topic e il payload giusto	publish chiamato con "warehouse.distance" e array	TS
TdU303	deve chiamare outboundService.sendAddress con il DTO corretto	sendAddress chiamato con DTO corretto	TS
TdU304	saveWarehouse chiama create con warehouseId	create chiamato con warehou- seId	TS
TdU305	getWarehouseById ritorna WarehouseId se trovato	WarehouseId(2)	TS
TdU306	getWarehouseById ritorna null se non trovato	null	TS
TdU307	getAllWarehouses ritorna ar- ray di WarehouseId	[WarehouseId(1), Warehou- seId(2)]	TS
TdU308	saveWarehouseAddress chia- ma create con dati corretti	create chiamato con warehou- seId, state, address	TS
TdU309	removeWarehouseAddress chiama deleteOne con ware- houseId	deleteOne chiamato con ware- houseId	TS

Codice	Descrizione	Valore atteso	Stato
TdU310	updateWarehouseAddress chiama updateOne con dati corretti	updateOne chiamato con warehouseId e address	TS
TdU311	getWarehouseAddressById ritorna WarehouseAddress se trovato	WarehouseAddress con dati attesi	TS
TdU312	getWarehouseAddressById ritorna null se non trovato	null	TS
TdU313	getAllWarehouseAddresses ritorna array di WarehouseAddress	array di WarehouseAddress	TS
TdU314	saveWarehouseState chiama create con dati corretti	create chiamato con warehouseId e state	TS
TdU315	getWarehouseStateById ritorna WarehouseState se trovato	WarehouseState con dati attesi	TS
TdU316	getWarehouseStateById ritorna null se non trovato	null	TS
TdU317	getAllWarehouseStates ritorna array di WarehouseState	array di WarehouseState	TS
TdU318	updateWarehouseState chiama updateOne con dati corretti	updateOne chiamato con warehouseId e state	TS
TdU319	chiama logger.verbose con il messaggio corretto	logger.verbose chiamato con messaggio	TS
TdU320	restituisce un oggetto con pattern, data e id	oggetto con pattern, data, id	TS
TdU321	usa la cache se l'indirizzo è già presente	axios.get non chiamato, risultato da cache	TS
TdU322	lancia errore se la risposta è vuota	throw errore	TS
TdU323	geocodifica e salva in cache se non presente	axios.get chiamato, risultato corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU324	calcola la distanza tra due coordinate note	distanza tra Milano e Roma 477km	TS
TdU325	la distanza tra due punti uguali è zero	distanza = 0	TS
TdU326	updateAddress chiama il service e ritorna il risultato	updateWarehouseAddress chiamato, risultato "ok"	TS
TdU327	updateAddress gestisce errori	result contiene error code "system.invalidParams"	TS
TdU328	removeAddress chiama il service e ritorna il risultato	removeWarehouseAddress chiamato, risultato "removed"	TS
TdU329	removeAddress gestisce errori	result contiene error code "system.invalidParams"	TS
TdU330	receiveRequest chiama il service e ritorna la risposta	calculateDistance chiamato, risultato array	TS
TdU331	receiveRequest gestisce errori	result contiene error code "system.invalidParams"	TS
TdU332	updateWarehouseState chiama il service e ritorna il risultato	updateWarehouseState chiamato, risultato "updated"	TS
TdU333	updateWarehouseState gestisce errori	result contiene error code "system.invalidParams"	TS
TdU334	createWarehouse chiama il service e ritorna il risultato	saveWarehouse chiamato, risultato "created"	TS
TdU335	createWarehouse gestisce errori	result contiene error code "system.invalidParams"	TS
TdU336	should return access granted for global token	result contiene get: true, call: "*"	TS
TdU337	should return access denied for token with error	result contiene error code "system.accessDenied"	TS
TdU338	should return access denied for missing token	result contiene error code "system.accessDenied"	TS

Tabella 16: Test di unità di Routing

4.1.5 Microservizio di Inventario Aggregato

Codice	Descrizione	Valore atteso	Stato
TdU339	gestisce errori e restituisce JSON formattato	estrae messaggi da array di errori o usa il messaggio di un errore generico e ritorna JSON con code e meta appropriati	TS
TdU340	Verifica creazione istanza ProductQuantity con valori corretti	L'istanza deve avere l'ID e la quantità impostati	TS
TdU341	Verifica lettura proprietà di Product	Tutte le proprietà devono essere restituite correttamente	TS
TdU342	Verifica setName su Product	Il nome deve essere aggiornato correttamente	TS
TdU343	Verifica setUnitPrice su Product	Il prezzo deve essere aggiornato correttamente	TS
TdU344	Verifica setQuantity su Product	La quantità deve essere aggiornata correttamente	TS
TdU345	Verifica setQuantityReserved su Product	La quantità riservata deve essere aggiornata correttamente	TS
TdU346	Verifica setMinThres su Product	La soglia minima deve essere aggiornata correttamente	TS
TdU347	Verifica setMaxThres su Product	La soglia massima deve essere aggiornata correttamente	TS
TdU348	Verifica aggiunta prodotto a InventoryAggregated	Il prodotto deve essere aggiunto all'inventario	TS
TdU349	Verifica addProduct su InventoryAggregatedService	Il repository deve essere chiamato con il prodotto	TS
TdU350	Verifica updateProduct su InventoryAggregatedService	Il repository deve essere chiamato con il prodotto	TS
TdU351	Verifica removeProduct su InventoryAggregatedService	Il repository deve essere chiamato con ID e warehouse	TS
TdU352	Verifica getAllProducts su InventoryAggregatedService	Restituisce l'inventario aggregato	TS

Codice	Descrizione	Valore atteso	Stato
TdU353	Verifica getAll su InventoryAggregatedService	Restituisce l'inventario aggregato	TS
TdU354	Verifica getProduct su InventoryAggregatedService se trovato	Restituisce il prodotto	TS
TdU355	Verifica getProduct su InventoryAggregatedService se non trovato	Lancia errore «Product not found»	TS
TdU356	Verifica getProductAggregated su InventoryAggregatedService se trovato	Restituisce il prodotto aggregato	TS
TdU357	Verifica getProductAggregated su InventoryAggregatedService se non trovato	Lancia errore «Product not found»	TS
TdU358	Verifica addProduct su InventoryAggregatedRepositoryImpl	Salva il prodotto nel database	TS
TdU359	Verifica removeProduct su InventoryAggregatedRepositoryImpl	Rimuove il prodotto dal database	TS
TdU360	Verifica updateProduct su InventoryAggregatedRepositoryImpl	Aggiorna il prodotto nel database	TS
TdU361	Verifica getAllProducts su InventoryAggregatedRepositoryImpl	Restituisce l'inventario aggregato	TS
TdU362	Verifica getAll su InventoryAggregatedRepositoryImpl	Restituisce l'inventario aggregato	TS
TdU363	Verifica getProduct su InventoryAggregatedRepositoryImpl	Restituisce il prodotto se trovato	TS
TdU364	Verifica getProduct su InventoryAggregatedRepositoryImpl se non trovato	Restituisce null	TS

Codice	Descrizione	Valore atteso	Stato
TdU365	Verifica getProductAggregated su InventoryAggregatedRepositoryImpl	Restituisce il prodotto aggregato	TS
TdU366	Verifica getProductAggregated su InventoryAggregatedRepositoryImpl se non trovato	Restituisce null	TS
TdU367	Verifica syncAddedStock su CloudInventoryEventAdapter	Chiama il service con il prodotto	TS
TdU368	Verifica syncEditedStock su CloudInventoryEventAdapter	Chiama il service con il prodotto	TS
TdU369	Verifica syncRemovedStock su CloudInventoryEventAdapter	Chiama il service con ID e warehouse	TS
TdU370	Verifica getAll su CloudInventoryEventAdapter	Restituisce DTO inventario aggregato	TS
TdU371	Verifica getAllProducts su CloudInventoryEventAdapter	Restituisce DTO inventario aggregato	TS
TdU372	Verifica getProduct su CloudInventoryEventAdapter se trovato	Restituisce DTO prodotto	TS
TdU373	Verifica getProduct su CloudInventoryEventAdapter se non trovato	Restituisce null	TS
TdU374	Verifica getProductAggregated su CloudInventoryEventAdapter se trovato	Restituisce DTO prodotto	TS
TdU375	Verifica getProductAggregated su CloudInventoryEventAdapter se non trovato	Restituisce null	TS
TdU376	Verifica mapping SyncProductDTO -> Product	Restituisce entità Product corretta	TS
TdU377	Verifica mapping SyncProductIdDTO -> ProductId	Restituisce entità ProductId corretta	TS

Codice	Descrizione	Valore atteso	Stato
TdU378	Verifica mapping SyncWarehouseIdDTO -> WarehouseId	Restituisce entità WarehouseId corretta	TS
TdU379	Verifica mapping SyncInventoryDTO -> InventoryAggregated	Restituisce entità InventoryAggregated corretta	TS
TdU380	Verifica mapping Product -> SyncProductDTO	Restituisce DTO prodotto corretto	TS
TdU381	Verifica mapping ProductId -> SyncProductIdDTO	Restituisce DTO ProductId corretto	TS
TdU382	Verifica mapping WarehouseId -> SyncWarehouseIdDTO	Restituisce DTO WarehouseId corretto	TS
TdU383	Verifica mapping InventoryAggregated -> SyncInventoryDTO	Restituisce DTO inventario corretto	TS
TdU384	Verifica creazione StockRemovedDTO	Proprietà impostate correttamente	TS
TdU385	Verifica creazione SyncEventDTO	Proprietà impostate correttamente	TS
TdU386	Verifica creazione SyncInventoryDTO	Proprietà impostate correttamente	TS
TdU387	Verifica creazione SyncProductDTO	Proprietà impostate correttamente	TS
TdU388	Verifica creazione SyncProductIdDTO	Proprietà impostate correttamente	TS
TdU389	Verifica creazione SyncWarehouseIdDTO	Proprietà impostate correttamente	TS
TdU390	Verifica AccessController definito	Il controller deve essere definito	TS
TdU391	Verifica loginAccess su AccessController	Restituisce JSON con accesso completo	TS
TdU392	Verifica syncAddedStock su CommandHandler	Chiama adapter con DTO corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU393	Verifica syncRemovedStock su CommandHandler	Chiama adapter con DTO corretto	TS
TdU394	Verifica syncEditedStock su CommandHandler	Chiama adapter con DTO corretto	TS
TdU395	Verifica getProductAggregated su CommandHandler	Restituisce prodotto aggregato come JSON	TS
TdU396	Verifica getProduct su CommandHandler	Restituisce prodotto per warehouse come JSON	TS
TdU397	Verifica getAllProducts su CommandHandler	Restituisce tutti i prodotti come JSON	TS
TdU398	Verifica getAll su CommandHandler	Restituisce tutto l'inventario come JSON	TS
TdU399	Verifica getInventory su CommandHandler	Restituisce inventario come JSON	TS

Tabella 17: Test di unità di Inventario Aggregato

4.1.6 Microservizio di Ordine Aggregato

Codice	Descrizione	Valore atteso	Stato
TdU400	stockReserved chiama ordersService con id e items	syncUpdateReservedStock invocato con dto.id e dto.items	TS
TdU401	syncAddSellOrder chiama ordersService con ordine	syncCreateSellOrder invocato con dto	TS
TdU402	syncAddInternalOrder chiama ordersService con ordine interno	syncCreateInternalOrder invocato con dto	TS
TdU403	updateOrderState aggiorna lo stato dell'ordine	syncUpdateOrderState invocato con id e stato	TS
TdU404	cancelOrder cancella un ordine	syncCancelOrder invocato con id	TS
TdU405	completeOrder completa un ordine	syncUpdateOrderState invocato con id e COMPLETED	TS

Codice	Descrizione	Valore atteso	Stato
TdU406	getOrderState recupera stato ordine dal repository e converte DTO	repository e dataMapper restituiscono DTO con stato	TS
TdU407	getOrder recupera un ordine interno	repository e dataMapper restituiscono InternalOrderDTO	TS
TdU408	getOrder recupera un ordine di vendita	repository e dataMapper restituiscono SellOrderDTO	TS
TdU409	getOrder lancia errore se tipo non riconosciuto	errore lanciato	TS
TdU410	getAllFilteredOrders recupera ordini filtrati	repository e dataMapper restituiscono DTO array	TS
TdU411	getAllOrders recupera tutti gli ordini	repository e dataMapper restituiscono DTO array	TS
TdU412	orderUpdated pubblica aggiornamento ordine su NATS	natsService.publish chiamato con topic e ordine	TS
TdU413	publishAllProducts pubblica tutti i prodotti	natsService.publish chiamato con topic e oggetti orders	TS
TdU414	orderCancelled pubblica cancellazione ordine	natsService.publish chiamato con topics e orderId	TS
TdU415	orderCompleted pubblica completamento ordine	natsService.publish chiamato con topics e orderId	TS
TdU416	grant access per token globale	accesso consentito	TS
TdU417	grant access per warehouse assegnato	accesso consentito	TS
TdU418	deny access per warehouse non assegnato	accesso negato	TS
TdU419	deny access se token ha errore	accesso negato con codice e messaggio di errore	TS
TdU420	deny access se token mancante	accesso negato con codice e messaggio di errore	TS

Codice	Descrizione	Valore atteso	Stato
TdU421	chiama stockReserved con inboundPortsAdapter.stockReserved payload	inboundPortsAdapter.stockReserved invocato con payload	TS
TdU422	chiama syncAddSellOrder con inboundPortsAdapter.syncAddSellOrder payload	inboundPortsAdapter.syncAddSellOrder invocato con payload	TS
TdU423	chiama syncAddInternalOrder con inboundPortsAdapter.syncAddInternalOrder payload	inboundPortsAdapter.syncAddInternalOrder invocato con payload	TS
TdU424	chiama updateOrderState con inboundPortsAdapter.updateOrderState payload	inboundPortsAdapter.updateOrderState invocato con id e stato	TS
TdU425	chiama cancelOrder con payload	inboundPortsAdapter.cancelOrder invocato con id	TS
TdU426	chiama completeOrder con inboundPortsAdapter.completeOrder payload	inboundPortsAdapter.completeOrder invocato con id	TS
TdU427	chiama getOrderState e ritorna valore	inboundPortsAdapter.getOrderState restituisce DTO con stato	TS
TdU428	chiama getOrder per InternalOrder	repository e dataMapper restituiscono InternalOrderDTO	TS
TdU429	chiama getOrder per SellOrder	repository e dataMapper restituiscono SellOrderDTO	TS
TdU430	getOrder lancia errore se tipo non riconosciuto	errore lanciato	TS
TdU431	getAllFilteredOrders chiama repository e dataMapper	restituisce collection con rid	TS
TdU432	getAllOrders chiama repository e dataMapper	restituisce collection con rid	TS
TdU433	getAllOrdersForCentralized chiama repository e restituisce collection	restituisce collection con orderId	TS
TdU434	getAllOrdersForCentralized restituisce null se repository vuota	result null	TS
TdU435	SyncOrderIdDTO valido	nessun errore	TS

Codice	Descrizione	Valore atteso	Stato
TdU436	SyncOrderIdDTO non valido	errore di validazione	TS
TdU437	SyncItemIdDTO valido	nessun errore	TS
TdU438	SyncItemIdDTO non valido	errore di validazione	TS
TdU439	SyncOrderItemDTO valido	nessun errore	TS
TdU440	SyncOrderItemDTO non valido	errore di validazione	TS
TdU441	SyncOrderItemDetailDTO valido	nessun errore	TS
TdU442	SyncOrderItemDetailDTO non valido	errore di validazione	TS
TdU443	SyncOrderStateDTO valido	nessun errore	TS
TdU444	SyncOrderStateDTO non valido	errore di validazione	TS
TdU445	SyncSellOrderDTO valido	nessun errore	TS
TdU446	SyncSellOrderDTO non valido	errore di validazione	TS
TdU447	SyncInternalOrderDTO valido	nessun errore	TS
TdU448	SyncInternalOrderDTO non valido	errore di validazione	TS
TdU449	SyncOrderQuantityDTO valido	nessun errore	TS
TdU450	SyncOrderQuantityDTO non valido	errore di validazione	TS
TdU451	SyncOrdersDTO valido	nessun errore	TS
TdU452	SyncOrdersDTO non valido	errore di validazione	TS
TdU453	Logger verbose log del deserializer	log corretto chiamato	TS
TdU454	Deserializer restituisce oggetto con pattern, data e id	oggetto con pattern/data/id	TS

Codice	Descrizione	Valore atteso	Stato
TdU455	Bootstrap crea microservice con opzioni corrette	app creata con NATS e server	TS
TdU456	syncOrderIdToDomain converte DTO in dominio	istanza SyncOrderId con id	TS
TdU457	syncOrderStateToDomain accetta solo stati validi	stato valido o errore	TS
TdU458	syncOrderItemToDomain converte DTO in dominio	istanza SyncOrderItem	TS
TdU459	syncOrderItemDetailToDomain valida quantityReserved	istanza o errore se invalido	TS
TdU460	syncInternalOrderToDomain valida magazzini diversi	istanza o errore se magazzini uguali	TS
TdU461	syncSellOrderToDomain converte correttamente	istanza SyncSellOrder	TS
TdU462	syncOrderIdToDTO converte dominio in DTO	oggetto con id	TS
TdU463	syncOrderStateToDTO converte dominio in DTO	oggetto con orderState	TS
TdU464	syncOrderItemToDTO converte dominio in DTO	oggetto itemId/quantity	TS
TdU465	syncOrderItemDetailToDTO converte dominio in DTO	oggetto item/quantity/unit-Price	TS
TdU466	syncInternalOrderToDTO converte dominio in DTO	oggetto con id e magazzini	TS
TdU467	syncSellOrderToDTO converte dominio in DTO	oggetto con id e magazzino/destinazione	TS
TdU468	syncOrdersToDTO converte collezione di ordini	array sell/internal con id	TS
TdU469	NATS config struttura corretta	tutte le proprietà presenti	TS

Codice	Descrizione	Valore atteso	Stato
TdU470	NATS usa NATS_URL se presente	server custom	TS
TdU471	NATS usa default se NATS_URL non presente	server localhost	TS
TdU472	NatsService onModuleInit inzializza connessione	nc definito	TS
TdU473	NatsService onModuleDestroy chiude connessione	close chiamato	TS
TdU474	NatsService publish invia messaggio codificato	messaggio codificato pubblicato	TS
TdU475	syncUpdateOrderState chiama repository con parametri corretti	chiamata repository con id e stato	TS
TdU476	syncCreateSellOrder chiama repository con nuovo ordine	repository chiamata con ordine con id corretto	TS
TdU477	syncCreateInternalOrder chiama repository con nuovo ordine	repository chiamata con ordine con id corretto	TS
TdU478	syncCancelOrder chiama repository e logga corretto	repository chiamata con id	TS
TdU479	syncCancelOrder logga errore se repository restituisce false	repository chiamata con id	TS
TdU480	syncUpdateReservedStock chiama repository con parametri corretti	repository chiamata con id e items	TS
TdU481	SyncOrderId restituisce id	ritorna id corretto	TS
TdU482	SyncOrderId restituisce tipo ordine	ritorna prima lettera dell'id	TS
TdU483	SyncItemId restituisce id	ritorna id corretto	TS
TdU484	SyncOrderItem restituisce quantity	ritorna quantity corretta	TS

Codice	Descrizione	Valore atteso	Stato
TdU485	Serializer serializza risposta con response	data correttamente formattata	TS
TdU486	Serializer serializza risposta con err come stringa	data con errore formattato	TS
TdU487	Serializer serializza risposta con err come array	data con errori concatenati	TS
TdU488	Serializer serializza risposta senza response né err	data invariata	TS
TdU489	getId restituisce InternalOrder se trovato	ritorna InternalOrder con id corretto	TS
TdU490	getId restituisce SellOrder se InternalOrder non trovato	ritorna SellOrder con id corretto	TS
TdU491	getId lancia errore se nessun ordine trovato	errore «Ordine con ID ... non trovato»	TS
TdU492	getState restituisce stato di InternalOrder se trovato	ritorna stato corretto	TS
TdU493	getState restituisce stato di SellOrder se InternalOrder non trovato	ritorna stato corretto	TS
TdU494	getState lancia errore se nessun ordine trovato	errore «Stato per l'ordine ... non trovato»	TS
TdU495	getAllFilteredOrders restituisce SyncOrders filtrati	ritorna collezione interna e sell con lunghezza corretta	TS
TdU496	syncRemoveById ritorna false se ordine già cancellato	ritorna false	TS
TdU497	syncRemoveById ritorna true se cancellazione riuscita	ritorna true	TS
TdU498	syncUpdateOrderState aggiorna InternalOrder	ritorna InternalOrder aggiornato	TS
TdU499	syncUpdateOrderState aggiorna SellOrder se InternalOrder non trovato	ritorna SellOrder aggiornato	TS

Codice	Descrizione	Valore atteso	Stato
TdU500	syncUpdateOrderState lancia errore se nessun ordine trovato	errore	TS
TdU501	syncUpdateReservedStock lancia errore se nessun ordine trovato	errore	TS
TdU502	syncUnreservedStock aggiorna tutti gli items di InternalOrder	bulkWrite chiamato, stock resettato	TS
TdU503	syncUnreservedStock aggiorna tutti gli items di SellOrder	bulkWrite chiamato, stock resettato	TS
TdU504	syncUnreservedStock lancia errore se nessun documento trovato dopo bulkWrite	errore «Impossibile trovare l'ordine ...»	TS
TdU505	syncUpdateReservedStock lancia errore se item non trovato	errore «Impossibile trovare l'ordine ...»	TS

Tabella 18: Test di unità di Ordine Aggregato

4.1.7 Microservizio di Inventario

Codice	Descrizione	Valore atteso	Stato
TdU506	Verifica creazione ProductSchema e modello Mongoose	Il modello deve essere definito	TS
TdU507	Verifica creazione istanza prodotto tramite ProductSchema	Le proprietà devono essere valorizzate correttamente	TS
TdU508	Verifica presenza campi obbligatori in ProductSchema	Tutti i campi richiesti devono essere presenti	TS
TdU509	Verifica onModuleInit di OutboundEventHandler	Deve chiamare connect sul client NATS	TS
TdU510	Verifica emissione evento belowMinThres	Deve chiamare emit con il payload corretto	TS
TdU511	Verifica emissione evento aboveMaxThres	Deve chiamare emit con il payload corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU512	Verifica emissione evento stockAdded	Deve chiamare emit con il payload corretto	TS
TdU513	Verifica emissione evento stockRemoved	Deve chiamare emit con il payload corretto	TS
TdU514	Verifica emissione evento stockUpdated	Deve chiamare emit con il payload corretto	TS
TdU515	Verifica emissione evento sufficientProductAvailability	Deve chiamare emit con il payload corretto	TS
TdU516	Verifica emissione evento reservationQuantities	Deve chiamare emit con il payload corretto	TS
TdU517	Verifica emissione evento stockShipped	Deve chiamare emit con il payload corretto	TS
TdU518	Verifica chiamata handlerBelowMinThres su OutboundEventAdapter	Deve chiamare handlerBelowMinThres	TS
TdU519	Verifica chiamata handlerStockAdded su OutboundEventAdapter	Deve chiamare handlerStockAdded	TS
TdU520	Verifica chiamata handlerStockRemoved su OutboundEventAdapter	Deve chiamare handlerStockRemoved	TS
TdU521	Verifica chiamata handlerStockUpdated su OutboundEventAdapter	Deve chiamare handlerStockUpdated	TS
TdU522	Verifica chiamata handlerSufficientProductAvailability su OutboundEventAdapter	Deve chiamare handlerSufficientProductAvailability	TS
TdU523	Verifica chiamata handlerReservationQuantities su OutboundEventAdapter	Deve chiamare handlerReservationQuantities	TS
TdU524	Verifica chiamata handlerStockShipped su OutboundEventAdapter	Deve chiamare handlerStockShipped	TS

Codice	Descrizione	Valore atteso	Stato
TdU525	Verifica chiamata handler-StockReceived su OutboundEventAdapter	Deve chiamare handlerStockReceived	TS
TdU526	Verifica addProduct su InventoryService	Aggiunge prodotto e pubblica evento stockAdded	TS
TdU527	Verifica errore se prodotto già esistente in addProduct	Lancia eccezione se il prodotto esiste	TS
TdU528	Verifica removeProduct su InventoryService	Rimuove prodotto e pubblica evento stockRemoved	TS
TdU529	Verifica errore se prodotto non esiste in removeProduct	Lancia eccezione se il prodotto non esiste	TS
TdU530	Verifica editProduct su InventoryService	Aggiorna prodotto e pubblica evento stockUpdated	TS
TdU531	Verifica errore se prodotto non esiste in editProduct	Lancia eccezione se il prodotto non esiste	TS
TdU532	Verifica getProduct su InventoryService	Restituisce il prodotto se esiste	TS
TdU533	Verifica errore se prodotto non trovato in getProduct	Lancia eccezione se il prodotto non esiste	TS
TdU534	Verifica checkProductExistence true	Restituisce true se il prodotto esiste	TS
TdU535	Verifica checkProductExistence false	Restituisce false se il prodotto non esiste	TS
TdU536	Verifica checkProductThres true	Restituisce true se il prodotto è nei limiti	TS
TdU537	Verifica checkProductThres false	Restituisce false se il prodotto è fuori soglia	TS
TdU538	Verifica addProductQuantity su InventoryService	Aumenta la quantità del prodotto	TS
TdU539	Verifica errore se prodotto non trovato in addProductQuantity	Lancia eccezione se il prodotto non esiste	TS

Codice	Descrizione	Valore atteso	Stato
TdU540	Verifica checkProductAvailability true	Restituisce true se tutti i prodotti sono disponibili	TS
TdU541	Verifica checkProductAvailability false	Restituisce false se almeno un prodotto non è disponibile	TS
TdU542	Verifica shipOrder su InventoryService	Aggiorna prodotti e pubblica evento stockShipped	TS
TdU543	Verifica reserveStock sufficiente	Riserva stock e pubblica evento sufficientProductAvailability	TS
TdU544	Verifica reserveStock non sufficiente	Riserva stock e pubblica evento reservedQuantities	TS
TdU545	Verifica receiveStock su InventoryService	Aggiorna prodotti e pubblica evento stockReceived	TS
TdU546	Verifica chiamata aboveMaxThres se quantità supera max in receiveStock	Pubblica evento aboveMaxThres	TS
TdU547	Verifica addProduct su InventoryRepositoryMongo	Salva il prodotto chiamando save	TS
TdU548	Verifica removeById su InventoryRepositoryMongo	Rimuove il prodotto e restituisce true	TS
TdU549	Verifica removeById restituisce false se prodotto non trovato	Restituisce false se deleted-Count è 0	TS
TdU550	Verifica updateProduct su InventoryRepositoryMongo	Aggiorna il prodotto chiamando updateOne	TS
TdU551	Verifica getById restituisce Product se trovato	Restituisce istanza Product corretta	TS
TdU552	Verifica getById restituisce null se non trovato	Restituisce null	TS
TdU553	Verifica getAllProducts restituisce Inventory con tutti i prodotti	Restituisce istanza Inventory con prodotti	TS

Codice	Descrizione	Valore atteso	Stato
TdU554	Verifica InventoryRepository-Module definito	Il repository deve essere definito	TS
TdU555	Verifica chiamata addProduct su InventoryRepositoryModule	Chiama addProduct con il prodotto	TS
TdU556	Verifica chiamata getById su InventoryRepositoryModule	Chiama getById con l'id	TS
TdU557	Verifica compilazione InventoryModule e provider	Service, repository e adapter devono essere definiti	TS
TdU558	Verifica addProduct su InventoryModule	Aggiunge prodotto e chiama stockAdded	TS
TdU559	Verifica errore se prodotto già esistente in InventoryModule	Lancia eccezione se il prodotto esiste	TS
TdU560	Verifica removeProduct su InventoryModule	Rimuove prodotto e chiama stockRemoved	TS
TdU561	Verifica errore se prodotto non esiste in InventoryModule	Lancia eccezione se il prodotto non esiste	TS
TdU562	Verifica creazione istanza InventoryDto con lista prodotti	La lista deve essere valorizzata correttamente	TS
TdU563	Verifica validazione positiva InventoryDto con lista corretta	Non devono esserci errori di validazione	TS
TdU564	Verifica validazione negativa InventoryDto se productList non è array	Deve fallire la validazione	TS
TdU565	Verifica handleAddQuantity su InboundEventController	Chiama addQuantity con il payload corretto	TS
TdU566	Verifica handleOrderRequest su InboundEventController	Chiama orderRequest con il payload corretto	TS
TdU567	Verifica handleShipOrderRequest su InboundEventController	Chiama shipOrderRequest con il payload corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU568	Verifica handleReceiveShipment su InboundEventController	Chiama receiveShipment con il payload corretto	TS
TdU569	Verifica unreserveStock su InboundEventController	Chiama unreserveStock con il payload corretto	TS
TdU570	Verifica newStock su InboundEventListener	Chiama addProduct con il prodotto	TS
TdU571	Verifica removeStock su InboundEventListener	Chiama removeProduct con l'id	TS
TdU572	Verifica editStock su InboundEventListener	Chiama editProduct con il prodotto	TS
TdU573	Verifica handleGetProduct su InboundEventListener	Chiama getProduct con l'id	TS
TdU574	Verifica getInventory su InboundEventListener	Chiama getInventory	TS
TdU575	Verifica orderRequest su InboundEventListener	Chiama reserveStock con i parametri corretti	TS
TdU576	Verifica addProductQuantity su InboundEventListener	Chiama addProductQuantity con il payload	TS
TdU577	Verifica shipOrderRequest su InboundEventListener	Chiama shipOrder con i parametri corretti	TS
TdU578	Verifica receiveShipment su InboundEventListener	Chiama receiveStock con i parametri corretti	TS
TdU579	Verifica toDomainProductQuantityArray su DataMapper	Restituisce orderId e array ProductQuantity corretti	TS
TdU580	Verifica toDTOProductId su DataMapper	Restituisce DTO ProductId corretto	TS
TdU581	Verifica toDtoInventory su DataMapper	Restituisce DTO Inventory con lista prodotti	TS
TdU582	Verifica creazione BelowMinThresDto	Proprietà valorizzate correttamente	TS

Codice	Descrizione	Valore atteso	Stato
TdU583	Verifica validazione positiva BelowMinThresDto	Non devono esserci errori di validazione	TS
TdU584	Verifica validazione negativa BelowMinThresDto per numeri negativi	Deve fallire la validazione	TS
TdU585	Verifica creazione AboveMaxThresDto	Proprietà valorizzate correttamente	TS
TdU586	Verifica validazione positiva AboveMaxThresDto	Non devono esserci errori di validazione	TS
TdU587	Verifica validazione negativa AboveMaxThresDto per numeri negativi	Deve fallire la validazione	TS
TdU588	Verifica handleNewStock su CommandHandler	Chiama newStock con il payload corretto	TS
TdU589	Verifica handleRemoveStock su CommandHandler	Chiama removeStock con l'id corretto	TS
TdU590	Verifica handleEditStock su CommandHandler	Chiama editStock con il payload corretto	TS
TdU591	Verifica handleGetProduct su CommandHandler	Chiama handleGetProduct con l'id corretto	TS

Tabella 19: Test di unità di Inventario

4.1.8 Microservizio di Ordini

Codice	Descrizione	Valore atteso	Stato
TdU592	Verifica che l'ordine interno sia inizializzato correttamente con tutte le proprietà	L'ordine deve avere ID "1233dcc2d-6637-4120-b80c-199f18e0ff2b", 1 articolo, stato PENDING, magazzino partenza 1 e destinazione 2	TS
TdU593	Verifica l'aggiornamento del magazzino di destinazione in un ordine interno	Il magazzino di destinazione deve essere aggiornato a 5	TS

Codice	Descrizione	Valore atteso	Stato
TdU594	Verifica l'aggiornamento dello stato dell'ordine in un ordine interno	Lo stato dell'ordine deve essere aggiornato a PROCES-SING	TS
TdU595	Verifica che ItemId restituisca l'ID corretto	L'ID dell'articolo deve essere 42	TS
TdU596	Verifica che ItemId memorizzi correttamente ID diversi	I due articoli devono avere ID 1 e 999 rispettivamente	TS
TdU597	Verifica che Order restituisca l'ID ordine corretto	L'ID dell'ordine deve essere "I233dcc2d-6637-4120-b80c-199f18e0ff2b"	TS
TdU598	Verifica l'impostazione e il recupero degli articoli in Order	Gli articoli devono essere aggiornati correttamente	TS
TdU599	Verifica l'impostazione e il recupero dello stato dell'ordine in Order	Lo stato dell'ordine deve essere aggiornato a PROCES-SING	TS
TdU600	Verifica l'impostazione e il recupero della data di creazione in Order	La data di creazione deve essere aggiornata a "2025-01-01"	TS
TdU601	Verifica l'impostazione e il recupero del magazzino di partenza in Order	Il magazzino di partenza deve essere aggiornato a 5	TS
TdU602	Verifica che OrderId restituisca l'ID corretto	L'ID dell'ordine deve essere "S933dcc2d-6637-4120-b80c-199f18e0ff2b"	TS
TdU603	Verifica che OrderId restituisca il tipo ordine corretto	Il tipo ordine deve essere "S" (prima lettera dell'ID)	TS
TdU604	Verifica che OrderItem restituisca l'ID articolo corretto	L'ID dell'articolo deve essere 125	TS
TdU605	Verifica che OrderItem restituisca la quantità corretta	La quantità deve essere 23	TS
TdU606	Verifica che OrderItemDetail restituisca l'OrderItem corretto	Deve restituire l'OrderItem inizializzato	TS

Codice	Descrizione	Valore atteso	Stato
TdU607	Verifica che OrderItemDetail restituisca la quantità riservata corretta	La quantità riservata deve essere 2	TS
TdU608	Verifica che OrderItemDetail restituisca il prezzo unitario corretto	Il prezzo unitario deve essere 100	TS
TdU609	Verifica l'aggiornamento della quantità riservata in OrderItemDetail	La quantità riservata deve essere aggiornata a 10	TS
TdU610	Verifica l'aggiornamento del prezzo unitario in OrderItemDetail	Il prezzo unitario deve essere aggiornato a 200	TS
TdU611	Verifica l'aggiornamento degli ordini di vendita in Orders	Gli ordini di vendita devono essere aggiornati correttamente	TS
TdU612	Verifica l'aggiornamento degli ordini interni in Orders	Gli ordini interni devono essere aggiornati correttamente	TS
TdU613	Verifica che SellOrder restituisca l'indirizzo di destinazione corretto	L'indirizzo deve essere "Via Roma 1"	TS
TdU614	Verifica l'aggiornamento dell'indirizzo di destinazione in SellOrder	L'indirizzo deve essere aggiornato a "Via Milano 2"	TS
TdU615	Verifica che SellOrder restituisca l'ID ordine corretto	L'ID ordine deve essere "S933dcc2d-6637-4120-b80c-199f18e0ff2b"	TS
TdU616	Verifica che SellOrder restituisca gli articoli corretti	Deve restituire l'array di OrderItemDetail inizializzato	TS
TdU617	Verifica che SellOrder restituisca lo stato ordine corretto	Lo stato ordine deve essere PENDING	TS
TdU618	Verifica che SellOrder restituisca la data di creazione corretta	La data di creazione deve essere "2025-09-02"	TS

Codice	Descrizione	Valore atteso	Stato
TdU619	Verifica che SellOrder restituisca il magazzino di partenza corretto	Il magazzino di partenza deve essere 1	TS
TdU620	Verifica la validazione di un InternalOrderDTO corretto	Non devono esserci errori di validazione	TS
TdU621	Verifica che la validazione fallisca se orderId non inizia con I o S	Devono essere presenti errori di validazione	TS
TdU622	Verifica che la validazione fallisca se orderState non è uno stato valido	Devono essere presenti errori di validazione per orderState	TS
TdU623	Verifica che la validazione fallisca se items non è un array	Devono essere presenti errori di validazione per items	TS
TdU624	Verifica che la validazione fallisca se warehouseDeparture non è un intero	Devono essere presenti errori di validazione per warehouseDeparture	TS
TdU625	Verifica che la validazione fallisca se warehouseDestination non è un intero	Devono essere presenti errori di validazione per warehouseDestination	TS
TdU626	Verifica la validazione di un OrderItemDetailDTO corretto	Non devono esserci errori di validazione	TS
TdU627	Verifica che la validazione fallisca se quantityReserved è negativo	Devono essere presenti errori di validazione per quantityReserved	TS
TdU628	Verifica che la validazione fallisca se unitPrice ha più di 2 decimali	Devono essere presenti errori di validazione per unitPrice	TS
TdU629	Verifica che la validazione fallisca se item è mancante	Devono essere presenti errori di validazione per item	TS
TdU630	Verifica la validazione di un OrderQuantityDTO corretto	Non devono esserci errori di validazione	TS

Codice	Descrizione	Valore atteso	Stato
TdU631	Verifica che la validazione fallisca se id è mancante	Devono essere presenti errori di validazione per id	TS
TdU632	Verifica che la validazione fallisca se items è vuoto	Devono essere presenti errori di validazione per items	TS
TdU633	Verifica che la validazione fallisca se la quantità dell'item è negativa	Devono essere presenti errori di validazione	TS
TdU634	Verifica che la validazione fallisca se itemId è mancante	Devono essere presenti errori di validazione	TS
TdU635	Verifica la validazione di un OrdersDTO corretto	Non devono esserci errori di validazione	TS
TdU636	Verifica che la validazione fallisca se sellOrders non è un array	Devono essere presenti errori di validazione per sellOrders	TS
TdU637	Verifica che la validazione fallisca se internalOrders non è un array	Devono essere presenti errori di validazione per internalOrders	TS
TdU638	Verifica la validazione di un SellOrderDTO corretto	Non devono esserci errori di validazione	TS
TdU639	Verifica che la validazione fallisca se orderState non è uno stato valido	Devono essere presenti errori di validazione per orderState	TS
TdU640	Verifica che la validazione fallisca se items non è un array	Devono essere presenti errori di validazione per items	TS
TdU641	Verifica che la validazione fallisca se warehouseDeparture non è un intero	Devono essere presenti errori di validazione per warehouseDeparture	TS
TdU642	Verifica che la validazione fallisca se destinationAddress è vuoto	Devono essere presenti errori di validazione per destinationAddress	TS
TdU643	Verifica accesso consentito con token globale in commandOrder	Deve restituire accesso completo (get: true, call: «»)	TS

Codice	Descrizione	Valore atteso	Stato
TdU644	Verifica accesso consentito con warehouseAssigned corrispondente	Deve restituire accesso completo (get: true, call: «»)	TS
TdU645	Verifica accesso negato con warehouseAssigned non corrispondente	Deve restituire accesso negato (get: false)	TS
TdU646	Verifica accesso negato senza token	Deve restituire errore di accesso negato	TS
TdU647	Verifica accesso negato con errore nel token	Deve restituire errore di accesso negato con messaggio	TS
TdU648	Verifica che orderAccess deleghi a checkAccess	Deve chiamare il metodo checkAccess con i dati corretti	TS
TdU649	Verifica che ordersAccess deleghi a checkAccess	Deve chiamare il metodo checkAccess con i dati corretti	TS
TdU650	Verifica conversione InternalOrderDTO in InternalOrder	Deve creare un'istanza corretta di InternalOrder con tutte le proprietà	TS
TdU651	Verifica errore quando magazzino partenza e destinazione sono uguali	Deve lanciare un errore di validazione	TS
TdU652	Verifica conversione SellOrderDTO in SellOrder	Deve creare un'istanza corretta di SellOrder con tutte le proprietà	TS
TdU653	Verifica conversione OrderItemDTO in OrderItem	Deve creare un'istanza corretta di OrderItem con ID e quantità	TS
TdU654	Verifica conversione OrderIdDTO in OrderId	Deve creare un'istanza corretta di OrderId con ID	TS
TdU655	Verifica conversione OrderIdDTO in OrderId per riferimento ordine vendita	Deve creare un'istanza corretta di OrderId con ID	TS
TdU656	Verifica conversione OrderStateDTO valido in OrderState	Deve restituire lo stato ordine corretto	TS

Codice	Descrizione	Valore atteso	Stato
TdU657	Verifica errore per OrderStateDTO non valido	Deve lanciare un errore di stato non valido	TS
TdU658	Verifica conversione OrderItemDetailDTO in OrderItemDetail	Deve creare un'istanza corretta con tutte le proprietà	TS
TdU659	Verifica errore quando quantityReserved > quantity	Deve lanciare un errore di quantità non valida	TS
TdU660	Verifica conversione InternalOrder in InternalOrderDTO	Deve creare un DTO corretto con tutte le proprietà	TS
TdU661	Verifica conversione SellOrder in SellOrderDTO	Deve creare un DTO corretto con tutte le proprietà	TS
TdU662	Verifica conversione OrderItem in OrderItemDTO	Deve creare un DTO corretto con ID e quantità	TS
TdU663	Verifica conversione OrderId in OrderIdDTO	Deve creare un DTO corretto con ID	TS
TdU664	Verifica conversione OrderId in OrderIdDTO per riferimento ordine vendita	Deve creare un DTO corretto con ID	TS
TdU665	Verifica conversione OrderState in OrderStateDTO	Deve creare un DTO corretto con stato	TS
TdU666	Verifica gestione di tutti i valori OrderState in conversione DTO	Deve gestire correttamente tutti gli stati	TS
TdU667	Verifica conversione OrderItemDetail in OrderItemDetailDTO	Deve creare un DTO corretto con tutte le proprietà	TS
TdU668	Verifica conversione OrderId e OrderItem[] in OrderQuantityDTO	Deve creare un DTO corretto con ID e items	TS
TdU669	Verifica gestione array vuoto in orderQuantityToDTO	Deve creare un DTO con array items vuoto	TS
TdU670	Verifica conversione Orders in OrdersDTO	Deve creare un DTO corretto con ordini vendita e interni	TS

Codice	Descrizione	Valore atteso	Stato
TdU671	Verifica gestione ordini vuoti in ordersToDTO	Deve creare un DTO con array vuoti	TS
TdU672	Verifica gestione errori conversione in ordini vendita	Deve gestire correttamente gli errori di conversione	TS
TdU673	Verifica gestione errori conversione in ordini interni	Deve gestire correttamente gli errori di conversione	TS
TdU674	Verifica gestione valori null/undefined in conversione DTO to Domain	Deve gestire correttamente i valori mancanti	TS
TdU675	Verifica gestione di tutti i valori OrderState	Deve gestire correttamente tutti gli stati	TS
TdU676	Verifica gestione valori zero in orderItemDetailToDomain	Deve gestire correttamente quantità e prezzo zero	TS
TdU677	Verifica gestione valori decimali in unitPrice	Deve gestire correttamente i prezzi decimali	TS
TdU678	Verifica conversione round-trip per InternalOrder	Deve mantenere l'integrità dei dati nella conversione avanti e indietro	TS
TdU679	Verifica conversione round-trip per SellOrder	Deve mantenere l'integrità dei dati nella conversione avanti e indietro	TS
TdU680	Verifica aggiornamento stock riservato in stockReserved	Deve chiamare il servizio correttamente con i parametri	TS
TdU681	Verifica aggiornamento stock completo riservato in sufficientProductAvailability	Deve chiamare il servizio correttamente	TS
TdU682	Verifica aggiunta ordine vendita in addSellOrder	Deve chiamare il servizio correttamente e restituire ID	TS
TdU683	Verifica aggiunta ordine interno in addInternalOrder	Deve chiamare il servizio correttamente e restituire ID	TS
TdU684	Verifica spedizione ordine in waitingForStock	Deve chiamare il servizio correttamente	TS

Codice	Descrizione	Valore atteso	Stato
TdU685	Verifica completamento ordine vendita in stockShipped	Deve aggiornare stato e completare ordine	TS
TdU686	Verifica ricezione ordine interno in stockShipped	Deve ricevere l'ordine interno correttamente	TS
TdU687	Verifica completamento ordine in stockReceived	Deve completare l'ordine correttamente	TS
TdU688	Verifica gestione rifornimento per ordine vendita in replenishmentReceived	Deve aggiornare stock riservato e verificare quantità	TS
TdU689	Verifica gestione rifornimento per ordine interno in replenishmentReceived	Deve aggiornare stock riservato e verificare quantità	TS
TdU690	Verifica aggiornamento stato ordine in updateOrderState	Deve chiamare il servizio correttamente con stato	TS
TdU691	Verifica cancellazione ordine in cancelOrder	Deve aggiornare stato a CANCELLED	TS
TdU692	Verifica completamento ordine in completeOrder	Deve completare l'ordine correttamente	TS
TdU693	Verifica recupero stato ordine in getOrderState	Deve restituire lo stato corretto come DTO	TS
TdU694	Verifica recupero ordine interno in getOrder	Deve restituire l'ordine interno come DTO	TS
TdU695	Verifica recupero ordine vendita in getOrder	Deve restituire l'ordine vendita come DTO	TS
TdU696	Verifica errore per tipo ordine sconosciuto in getOrder	Deve lanciare errore per tipo non riconosciuto	TS
TdU697	Verifica recupero tutti gli ordini in getAllOrders	Deve restituire tutti gli ordini come DTO	TS
TdU698	Verifica gestione items vuoti in replenishmentReceived	Deve gestire correttamente array vuoto	TS

Codice	Descrizione	Valore atteso	Stato
TdU699	Verifica gestione items non corrispondenti in replenishmentReceived	Deve gestire items non presenti nell'ordine	TS
TdU700	Verifica deserializzazione messaggio in InboundResponse-Deserializer	Deve loggare e restituire oggetto con pattern, data e id	TS
TdU701	Verifica creazione microservizio con opzioni corrette in bootstrap	Deve creare il microservizio con transport NATS e server corretti	TS
TdU702	Verifica recupero InternalOrder in getById	Deve restituire InternalOrder se trovato	TS
TdU703	Verifica recupero SellOrder se InternalOrder non trovato in getById	Deve restituire SellOrder se InternalOrder non esiste	TS
TdU704	Verifica errore se ordine non trovato in getById	Deve lanciare NotFoundException	TS
TdU705	Verifica salvataggio ordine vendita in addSellOrder	Deve salvare correttamente l'ordine vendita	TS
TdU706	Verifica salvataggio ordine interno in addInternalOrder	Deve salvare correttamente l'ordine interno	TS
TdU707	Verifica aggiornamento stato InternalOrder in updateOrderState	Deve aggiornare correttamente lo stato	TS
TdU708	Verifica aggiornamento stato SellOrder in updateOrderState	Deve aggiornare correttamente lo stato	TS
TdU709	Verifica errore se ordine COMPLETED in removeById	Deve lanciare RpcException	TS
TdU710	Verifica return false se ordine CANCELED in removeById	Deve restituire false	TS
TdU711	Verifica errore se ordine non trovato in updateReservedStock	Deve lanciare errore	TS

Codice	Descrizione	Valore atteso	Stato
TdU712	Verifica errore se quantità insufficiente per ordine vendita in <code>checkReservedQuantityForSellOrder</code>	Deve lanciare errore	TS
TdU713	Verifica errore se quantità insufficiente per ordine interno in <code>checkReservedQuantityForInternalOrder</code>	Deve lanciare errore	TS
TdU714	Verifica esistenza ordine	Restituisce true se l'ordine esiste	TS
TdU715	Aggiorna stock per <code>SellOrder</code>	<code>updateReservedStock</code> e <code>publishUpdatedReservedStock</code> vengono chiamati	TS
TdU716	Aggiorna stock per <code>InternalOrder</code>	<code>updateReservedStock</code> e <code>publishUpdatedReservedStock</code> vengono chiamati	TS
TdU717	Errore durante <code>updateReservedStock</code>	Viene sollevata un'eccezione "DB error"	TS
TdU718	Aggiorna stock completo per <code>SellOrder</code>	<code>updateReservedStock</code> viene chiamato	TS
TdU719	Aggiorna stock completo per <code>InternalOrder</code>	<code>updateReservedStock</code> viene chiamato	TS
TdU720	Aggiorna stato ordine con successo	<code>updateOrderState</code> e <code>orderStateUpdated</code> vengono chiamati	TS
TdU721	Transizione stato non valida	Viene sollevata un'eccezione per flusso non corretto	TS
TdU722	Aggiorna stato da magazzino non di partenza	<code>orderStateUpdated</code> NON viene chiamato	TS
TdU723	Crea <code>InternalOrder</code> in magazzino destinazione	Restituisce ID dell'ordine e <code>waitingForStock</code> viene chiamato	TS
TdU724	Annulla ordine <code>InternalOrder</code>	<code>updateOrderState</code> e <code>unreserveStock</code> vengono chiamati	TS

Codice	Descrizione	Valore atteso	Stato
TdU725	Stock riservato per SellOrder	updateReservedStock viene chiamato	TS
TdU726	Controllo riserva completa SellOrder	publishShipment viene chiamato	TS
TdU727	Controllo riserva incompleta SellOrder	publishStockRepl viene chiamato	TS
TdU728	Controllo riserva completa InternalOrder	publishInternalOrder viene chiamato	TS
TdU729	Controllo riserva incompleta InternalOrder	cancelOrder viene chiamato	TS
TdU730	Spedizione SellOrder	updateOrderState(SHIPPED) e publishShipment vengono chiamati	TS
TdU731	Spedizione InternalOrder	orderStateUpdated viene chiamato	TS
TdU732	Ricezione InternalOrder	receiveShipment viene chiamato	TS
TdU733	Ricezione SellOrder	receiveShipment NON viene chiamato	TS
TdU734	Completamento SellOrder	updateOrderState(COMPLETED) viene chiamato	TS
TdU735	Completamento InternalOrder nel magazzino di destinazione	orderCompleted e publishReserveStock vengono chiamati	TS
TdU736	Completamento InternalOrder in altro magazzino	orderCompleted NON viene chiamato	TS
TdU737	updateReservedStock con items vuoto	updateReservedStock viene comunque chiamato	TS
TdU738	updateReservedStock con item non corrispondenti	updateReservedStock viene chiamato comunque	TS
TdU739	Gestione evento stockReserved	stockReserved viene chiamato con il DTO	TS

Codice	Descrizione	Valore atteso	Stato
TdU740	Aggiunta SellOrder	Restituisce ID e viene chiamato addSellOrder	TS
TdU741	Errore durante aggiunta SellOrder	Restituisce messaggio di errore con "system.internalError"	TS
TdU742	Aggiunta InternalOrder	Restituisce ID e viene chiamato addInternalOrder	TS
TdU743	Errore durante aggiunta InternalOrder	Restituisce messaggio di errore con "system.internalError"	TS
TdU744	Gestione evento addInternalOrderEvent	Restituisce ID ordine correttamente	TS
TdU745	Disponibilità sufficiente prodotti	Viene chiamato sufficientProductAvailability	TS
TdU746	Gestione evento waitingForStock	Viene chiamato waitingForStock con ID ordine	TS
TdU747	Gestione evento stockShipped	Viene chiamato stockShipped con ID ordine	TS
TdU748	Gestione evento stockReceived	Viene chiamato stockReceived	TS
TdU749	Gestione evento replenishmentReceived	Viene chiamato replenishmentReceived	TS
TdU750	Aggiornamento stato ordine riuscito	Restituisce messaggio di stato aggiornato	TS
TdU751	Errore aggiornamento stato ordine	Restituisce "system.internalError"	TS
TdU752	Aggiornamento stato da evento	Viene chiamato updateOrderState con ID e stato	TS
TdU753	Errore aggiornamento stato da evento	Sollevata eccezione con messaggio "Update error"	TS
TdU754	Annullamento ordine riuscito	Viene chiamato cancelOrder con ID	TS
TdU755	Errore durante annullamento ordine	Sollevata RpcException	TS

Codice	Descrizione	Valore atteso	Stato
TdU756	Completamento ordine	Viene chiamato completeOrder con ID	TS
TdU757	Recupero stato ordine	Restituisce stato "PENDING" e chiama getOrderState	TS
TdU758	Recupero ordine SellOrder	Restituisce S123 e "Via Roma"	TS
TdU759	Recupero ordine InternalOrder	Restituisce I123 e contiene "warehouseDestination"	TS
TdU760	Recupero lista ordini	Restituisce entrambi SellOrder e InternalOrder	TS
TdU761	InternalOrder con sellOrderReference vuoto	Ordine aggiunto con successo	TS
TdU762	Parsing contesto con warehouse diverso	updateOrderState chiamato con ID e stato corretti	TS
TdU763	Connessione a NATS riuscita	connect viene chiamato senza errori	TS
TdU764	Errore durante connessione NATS	Log dell'errore con messaggio "Error connecting to NATS service"	TS
TdU765	Evento waitingForStock pubblicato	emit chiamato con subject corretto	TS
TdU766	Pubblicazione evento reserveStock	emit chiamato con subject e payload corretti	TS
TdU767	Pubblicazione evento updatedReservedStock	emit chiamato con subject e payload corretti	TS
TdU768	Pubblicazione evento unreserveStock	emit chiamato due volte per inventario e aggregate	TS
TdU769	Pubblicazione evento shipment	emit chiamato con subject e payload corretti	TS
TdU770	Pubblicazione evento stock replenishment	emit chiamato con subject e payload corretti	TS

Codice	Descrizione	Valore atteso	Stato
TdU771	Pubblicazione evento receive-Shipment	emit chiamato con subject corretto basato su warehouse	TS
TdU772	Aggiornamento stato ordine verso aggregate	emit chiamato con subject aggregate e payload vuoto	TS
TdU773	Aggiornamento stato ordine verso warehouse	emit chiamato con subject warehouse e payload vuoto	TS
TdU774	Errore durante aggiornamento stato ordine	Viene sollevato errore con messaggio "NATS error"	TS
TdU775	Pubblicazione evento ordine annullato	emit chiamato con subject cancel	TS
TdU776	Errore pubblicazione ordine annullato	Viene sollevato errore "NATS error"	TS
TdU777	Pubblicazione ordine completato	emit chiamato con subject complete	TS
TdU778	Errore pubblicazione ordine completato	Viene sollevato errore "Mapping error"	TS
TdU779	Pubblicazione InternalOrder verso aggregate	emit chiamato con subject aggregate e payload DTO	TS
TdU780	Pubblicazione InternalOrder verso warehouse	emit chiamato con subject warehouse e payload DTO	TS
TdU781	Pubblicazione SellOrder verso aggregate	emit chiamato con subject aggregate e contiene ID ordine	TS
TdU782	SellOrder restituisce ID come stringa	Risultato è "«S123»"	TS
TdU783	Gestione items array vuoto in publishReserveStock	emit chiamato con itemsDTO vuoto	TS
TdU784	Gestione items multipli in publishReserveStock	emit chiamato con due itemsDTO	TS
TdU785	Gestione warehouseId diverso da 1	emit chiamato con subject "warehouse.3.order.request"	TS

Codice	Descrizione	Valore atteso	Stato
TdU786	Serializzazione richiesta con proprietà "data" oggetto	Risultato uguale all'oggetto iniziale	TS
TdU787	Serializzazione richiesta con "data" vuoto	Risultato è { data: {} }	TS
TdU788	Serializzazione richiesta con "data" come stringa	Risultato è { data: "test" }	TS
TdU789	Serializzazione con proprietà "response"	Ritorna { data: { foo: "bar" } } + logging verbose	TS
TdU790	Serializzazione con errore come stringa	Ritorna JSON con errore (code: system.invalidParams, message: errore generico)	TS
TdU791	Serializzazione con errore come array di errori di validazione	Ritorna JSON con errore aggregato dai constraint + logging	TS
TdU792	Serializzazione fallback (nessun response né err)	Ritorna { data: "test" } + logging verbose	TS
TdU793	startSellOrderSaga chiama publishReserveStock con items corretti	Chiama repository, estrae item e chiama publishReserveStock	TS
TdU794	startInternalOrderSaga chiama publishReserveStock con items corretti	Chiama repository, estrae item e chiama publishReserveStock	TS
TdU795	Gestione errore da getById	startSellOrderSaga e startInternalOrderSaga lanciano errore se getById fallisce	TS
TdU796	Gestione errore da publishReserveStock	startSellOrderSaga e startInternalOrderSaga lanciano errore se publishReserveStock fallisce	TS

Tabella 20: Test di unità di Ordini

4.1.9 Microservizio di State

Codice	Descrizione	Valore atteso	Stato
TdU797	«Crea un DTO valido da oggetto plain»	«DTO valido, nessun errore di validazione»	TS
TdU798	«Fallisce la validazione se manca lo stato»	«errors[0].property = "state"»	TS
TdU799	«Fallisce la validazione se lo stato non è una stringa»	«errors[0].property = "state"»	TS
TdU800	«Fallisce la validazione se manca warehouseId»	«errors[0].property = "warehouseId"»	TS
TdU801	«Service definito»	«service definito»	TS
TdU802	«Crea Heartbeat con messaggio corretto usando syncHeartbeat»	«heartbeat contiene "ONLINE", id corretto, timestamp è Date»	TS
TdU803	«Chiama publishHeartbeat con Heartbeat usando sendHeartBeat»	«publishHeartbeat chiamato con Heartbeat, dati corretti»	TS
TdU804	«Crea documento State con tutte le proprietà»	«doc ha proprietà warehouseId, state, lastHeartbeat, lastHeartbeatMsg»	TS
TdU805	«Restituisce WarehouseState dal documento»	«result è WarehouseState, stato = "ACTIVE"»	TS
TdU806	«Restituisce stato "unknown" se non trovato»	«result è WarehouseState, stato = "unknown"»	TS
TdU807	«Restituisce true se update acknowledged»	«result = true»	TS
TdU808	«Restituisce false se update non acknowledged»	«result = false»	TS
TdU809	«Chiama getSyncedState con input oggetto»	«getSyncedState chiamato con { id: 5 }»	TS
TdU810	«Chiama getSyncedState con input stringa»	«getSyncedState chiamato con { id: 10 }»	TS
TdU811	«Default a warehouseId 0 se input malformato»	«getSyncedState chiamato con { id: 0 }»	TS

Codice	Descrizione	Valore atteso	Stato
TdU812	«Gestisce stringa JSON non valida»	«logger.error chiamato, getSyncedState con { id: 0 }»	TS
TdU813	«Chiama connect su onModuleInit»	«connect chiamato»	TS
TdU814	«Pubblica heartbeat»	«emit chiamato con topic heartbeat e dto»	TS
TdU815	«Pubblica stato magazzino»	«emit chiamato con topic stato e dati»	TS
TdU816	«Pubblica stato aggiornato»	«emit chiamato con topic updated e dati»	TS
TdU817	«Pubblica stato magazzino tramite adapter»	«publishState chiamato con warehouseId e state»	TS
TdU818	«Pubblica heartbeat tramite adapter»	«publishHeartbeat chiamato con HeartbeatDTO»	TS
TdU819	«Logga errore se publishState fallisce»	«logger.error chiamato»	TS
TdU820	«Logga errore se publishHeartbeat fallisce»	«logger.error chiamato»	TS
TdU821	«Chiama sendHeartBeat con oggetti dominio corretti»	«sendHeartBeat chiamato con WarehouseId e WarehouseState "ONLINE"»	TS
TdU822	«Crea entità Heartbeat»	«getHeartbeatMsg = "ALIVE", getId = 1»	TS
TdU823	«Crea entità WarehouseState»	«getState = "ACTIVE", dopo setState = "INACTIVE"»	TS
TdU824	«Crea entità WarehouseId»	«getId = 42»	TS
TdU825	«Mappa WarehouseIdDTO a WarehouseId»	«entity è WarehouseId, id = 5»	TS
TdU826	«Mappa WarehouseStateDTO a WarehouseState»	«entity è WarehouseState, stato = "ACTIVE"»	TS

Codice	Descrizione	Valore atteso	Stato
TdU827	«Mappa HeartbeatDTO a Heartbeat»	«entity è Heartbeat, msg = “ALIVE”, timestamp corretto, id = 1»	TS
TdU828	«Mappa WarehouseId a WarehouseIdDTO»	«dto = { id: 10 }»	TS
TdU829	«Mappa WarehouseState a WarehouseStateDTO»	«dto.state = “ACTIVE”, dto.warehouseId.id = 3»	TS
TdU830	«Mappa Heartbeat a HeartbeatDTO»	«dto è HeartbeatDTO, msg = “ALIVE”, warehouseId = 7, timestamp è Date»	TS

Tabella 21: Test di unità di State

4.2 Test di integrazione

I test di integrazione servono a verificare che più unità di codice funzionino correttamente insieme. L'obiettivo è individuare problemi che emergono quando i vari moduli interagiscono tra loro, ad esempio errori di comunicazione tra componenti o malintesi sui formati di dati scambiati. Questi test sono particolarmente utili quando si integrano sistemi esterni come database, API o servizi^G di terze parti.

Codice	Descrizione	Codice Requisito	Stato
TdI01	Testare la verifica del JWT(Json Web Token) tramite il microservizio autenticazione, Resgate e accesso ad un dato microservizio		S
TdI02	Testare la sincronizzazione da Inventario a Inventario Aggregato dopo un operazione di aggiunta/modifica/rimozione		S
TdI03	Testare la sincronizzazione da Ordine a Ordine Aggregato dopo un operazione di aggiunta/modifica		S
TdI04	Testare interazione tra microservizi stato e microservizio stato Aggregato		S
TdI05	Testare interazione tra Magazzini differenti in caso di ordine interno		S
TdI06	Testare interazione tra microservizio Ordini e Inventario		S
TdI07	Testare interazione tra microservizio Inventario a Inventario		S
TdI08	Testare interazione tra microservizio Inventario e Sistema centralizzato in caso di criticità soglia Min o Max		S
TdI09	Testare interazione tra microservizio Ordine e Sistema centralizzato in caso di insufficienza Prodotti		S

Codice	Descrizione	Codice Requisito	Stato
TdI09	Testare interazione tra microservizio Sistema centralizzato e Inventario aggregato per ottenere inventario		S
TdI10	Testare interazione tra microservizio Sistema centralizzato e Ordine aggregato per ottenere ordini		S
TdI11	Testare interazione tra microservizio Sistema centralizzato e Routing per ottenere informazioni sulla distanza dei magazzini		S
TdI12	Testare interazione tra Inventario e MongoDB per archiviazione prodotti		S
TdI13	Testare interazione tra Ordini e MongoDB per archiviazione ordini		S

Tabella 22: Test di integrazione

4.3 Test di sistema

I test di sistema valutano il comportamento del software nel suo complesso, simulando l'uso reale da parte dell'utente in un ambiente quanto più possibile simile alla produzione. Questo tipo di test include tutti i componenti integrati e mira a verificare che il sistema nel suo insieme soddisfi i requisiti^G funzionali e non funzionali. È un passaggio fondamentale prima del collaudo finale. Questi test sono stati testati manualmente tramite blocchi di chiamate API che simulavano interazioni realistiche con il sistema.

Codice	Descrizione	Codice Requisito	Stato
TdS01	Verificare che il supervisore globale si possa registrare al primo avvio del sistema.	RF01/01	S
TdS02	Verificare che il supervisore globale abbia inserito l'email.	RF01/01	S
TdS03	Verificare che il supervisore globale abbia inserito il numero di cellulare.	RF01/01	S

Codice	Descrizione	Codice Requisito	Stato
TdS04	Verificare che il supervisore globale abbia inserito la password.	RF01/01	S
TdS05	Verificare che il supervisore globale riceva un avviso nel caso in cui l'email è errata dal punto di vista della forma.	RF01/02	S
TdS06	Verificare che il supervisore globale riceva un avviso nel caso in cui il numero di cellulare è errato dal punto di vista della forma.	RF01/03	S
TdS07	Verificare che il supervisore globale riceva un avviso nel caso in cui la password non rispetti la sintassi.	RF01/04	S
TdS08	Verificare che il supervisore globale riceva un avviso nel caso in cui la password e conferma password non corrispondono.	RF01/05	S
TdS09	Verificare che l'utente non autenticato possa effettuare l'autenticazione ^G .	RF02/01	S
TdS10	Verificare che l'utente non autenticato abbia inserito l'email.	RF02/01	S
TdS11	Verificare che l'utente non autenticato abbia inserito la password.	RF02/01	S
TdS12	Verificare che l'utente non autenticato riceva un avviso in caso di errore di autenticazione ^G .	RF02/03	S
TdS13	Verificare che l'utente possa effettuare il logout dal sistema.	RF02/02	S
TdS14	Verificare che il supervisore globale possa registrare nuovi supervisori locali.	RF03/01	S

Codice	Descrizione	Codice Requisito	Stato
TdS15	Verificare che il supervisore globale abbia inserito l'email per la registrazione di un nuovo supervisore locale.	RF03/01	S
TdS16	Verificare che il supervisore globale abbia inserito il numero di cellulare per la registrazione di un nuovo supervisore locale.	RF03/01	S
TdS17	Verificare che il supervisore globale abbia inserito la password per la registrazione di un nuovo supervisore locale.	RF03/01	S
TdS18	Verificare che il supervisore globale riceva un avviso nel caso in cui l'email è errata dal punto di vista della forma per la registrazione di un nuovo supervisore locale.	RF03/03	S
TdS19	Verificare che il supervisore globale riceva un avviso nel caso in cui il numero di cellulare è errato dal punto di vista della forma per la registrazione di un nuovo supervisore locale.	RF03/05	S
TdS20	Verificare che il supervisore globale riceva un avviso nel caso in cui la password non rispetti la sintassi per la registrazione di un nuovo supervisore locale.	RF03/04	S
TdS21	Verificare che il supervisore globale assegni il/i magazzino/i a cui il supervisore locale ha accesso in fase di registrazione di un nuovo supervisore locale.	RF03/02	S
TdS22	Verificare che il supervisore globale abbia inserito l'indirizzo fisico del magazzino aggiunto.	RF04/01	S

Codice	Descrizione	Codice Requisito	Stato
TdS23	Verificare che il supervisore globale possa rimuovere un magazzino dal sistema.	RF04/02	S
TdS24	Verificare che il supervisore globale abbia inserito il supervisore locale al magazzino.	RF04/03	S
TdS25	Verificare che ogni magazzino sia identificato univocamente.	RF04/04	S
TdS26	Verificare che il supervisore globale possa modificare le informazioni del magazzino	RF04/03	S
TdS27	Verificare che, per l'inserimento di un nuovo tipo di merce nell'inventario ^G , questa abbia un codice EAN univoco.	RF05/01	S
TdS28	Verificare che, per l'inserimento di un nuovo tipo di merce nell'inventario ^G , questa abbia un nome.	RF05/01	S
TdS29	Verificare che, per l'inserimento di un nuovo tipo di merce nell'inventario ^G , sia presente un avviso nel caso in cui il codice EAN non rispetti la sintassi prevista.	RF05/02	S
TdS30	Verificare che, per l'inserimento di un nuovo tipo di merce nell'inventario ^G , in caso di inserimento o modifica del prezzo unitario, sia presente un avviso nel caso in cui quest'ultimo non rispetti la forma corretta.	RF05/03 RF05/04 RF05/10	S
TdS31	Verificare che, per l'inserimento di un nuovo tipo di merce nell'inventario ^G , in caso di inserimento o modifica della quantità, sia presente un avviso nel caso	RF05/05 RF05/06 RF05/08 RF05/09	S

Codice	Descrizione	Codice Requisito	Stato
	in cui quest'ultima non rispetti la forma corretta.		
TdS32	Verificare che i supervisori possano rimuovere un tipo di prodotto ^G dall'inventario ^G solo se la quantità di quel prodotto ^G è zero.	RF05/07	S
TdS33	Verificare che, per l'inserimento di un ordine ^G di trasferimento ^G interno, sia selezionato il magazzino di partenza.	RF06/01	S
TdS34	Verificare che, per l'inserimento di un ordine ^G di vendita verso l'esterno, sia inserito l'indirizzo del destinatario.	RF06/02	S
TdS35	Verificare che, per l'inserimento di un ordine ^G , siano inseriti uno o più prodotti.	RF06/01 RF06/02	S
TdS36	Verificare che, per l'inserimento di un ordine ^G , sia presente un avviso se la quantità di prodotto ^G non è corretta nella forma.	RF06/03	S
TdS37	Verificare che i supervisori possano annullare ordini "in attesa" o "in elaborazione".	RF06/04	S
TdS38	Verificare che il supervisore globale possa visualizzare correttamente l'inventario ^G globale.	RF07/01	S
TdS39	Verificare che il supervisore globale possa visualizzare correttamente l'inventario ^G di ciascun magazzino.	RF07/02	S
TdS40	Verificare che il supervisore globale possa visualizzare correttamente un report degli ordini globali.	RF07/03	S

Codice	Descrizione	Codice Requisito	Stato
TdS41	Verificare che il supervisore globale possa visualizzare correttamente un report degli ordini di ciascun magazzino.	RF07/04	S
TdS42	Verificare che i supervisori locali possano visualizzare correttamente l'inventario ^G del/dei magazzino/i a loro assegnato/i.	RF07/05	S
TdS43	Verificare che i supervisori locali possano visualizzare correttamente un report degli ordini del/dei magazzino/i a loro assegnato/i.	RF07/06	S
TdS44	Verificare che i supervisori possano visualizzare correttamente lo stato di tutti i magazzini.	RF08	S
TdS45	Verificare che i supervisori possano visualizzare correttamente le notifiche informative prodotte dal sistema.	RF09/01	S
TdS46	Verificare che il supervisore globale possa accettare le richieste decisionali prodotte dal sistema.	RF09/02	S
TdS47	Verificare che il supervisore globale possa rifiutare le richieste decisionali prodotte dal sistema.	RF09/03	S
TdS48	Verificare che, in caso di inserimento o modifica del valore di soglia minima in un prodotto ^G da parte dei supervisori, si riceva un avviso nel caso in cui il valore di soglia minima non sia nella forma corretta.	RF10/01 RF10/03 RF10/04 RF10/05 RF10/06 RF10/07	S
TdS49	Verificare che, in caso di inserimento o modifica del valore di soglia massima in un prodotto ^G da parte dei supervisori, si riceva un avviso nel caso in cui il valore di	RF10/02 RF10/03 RF10/04 RF10/05	S

Codice	Descrizione	Codice Requisito	Stato
	soglia massima non sia nella forma corretta.	RF10/06 RF10/07	
TdS50	Verificare che ogni magazzino identifichi quando le scorte ^G di un determinato prodotto ^G scendono sotto una certa soglia.	RF11/01	S
TdS51	Verificare che ogni magazzino identifichi quando le scorte ^G di un determinato prodotto ^G salgono sopra una certa soglia.	RF11/02	S
TdS52	Verificare che ogni magazzino verifichi la disponibilità delle scorte ^G prima di soddisfare un ordine ^G .	RF12/01	S
TdS53	Verificare che ogni magazzino identifichi l'impossibilità di soddisfare un ordine ^G .	RF12/02	S
TdS54	Verificare che ogni magazzino identifichi l'insufficienza di scorte ^G per ciascun prodotto ^G .	RF12/03	S
TdS55	Verificare che, in caso di insufficienza di scorte ^G per un determinato prodotto ^G , il magazzino in questione richieda un riassortimento ^G al sistema.	RF12/03	S
TdS56	Verificare che ogni magazzino abbia un tracciamento dello stato degli ordini.	RF12/04	S
TdS57	Verificare che i magazzini possono inviare merce tramite un ordine ^G .	RF12/05	S
TdS58	Verificare che i magazzini possono ricevere un approvvigionamento ^G di merce dall'esterno tramite un ordine ^G .	RF12/06	S

Codice	Descrizione	Codice Requisito	Stato
TdS59	Verificare che i magazzini possono inviare merce verso l'esterno tramite un ordine ^G .	RF12/07	S
TdS60	Verificare che il magazzino di destinazione notifichi l'arrivo della merce al magazzino di origine.	RF12/08	S
TdS61	Verificare che venga rispettata la forma dello stato dell'ordine ^G . Sono ammessi i seguenti stati: "in attesa", "in elaborazione", "in transito", "annullato", "consegnato".	RF12/09	S
TdS62	Verificare che venga rispettata la forma dello stato del magazzino. Sono ammessi i seguenti stati: "online", "offline".	RF14/01	S
TdS63	Verificare che il sistema mostri un avviso di un eventuale cambio di stato del magazzino.	RF14/02	S
TdS64	Verificare che il sistema identifichi eventuali disconnessioni di uno o più magazzini segnandoli come offline.	RF14/03	S
TdS65	Verificare che il sistema sia in grado di programmare trasferimenti interni in caso di raggiungimento di una soglia critica di un prodotto ^G in un magazzino.	RF13/01	S
TdS66	Verificare che, nel caso di un magazzino offline, quest'ultimo annulli gli ordini che non può soddisfare dopo un periodo di tempo predefinito.	RF15/01	S
TdS67	Verificare che il supervisore globale possa definire il periodo di tempo dopo cui l'ordine ^G di un magazzino offline viene annullato.	RF15/02	S

Codice	Descrizione	Codice Requisito	Stato
TdS68	Verificare che il sistema riconosca il ritorno di un magazzino da “offline” a “online”.	RF16/01	S
TdS69	Verificare che un ordine ^G “in transito” cambi il suo stato in “annullato” dopo il superamento di una soglia temporale predefinita.	RF17/01	S
TdS70	Verificare che il supervisore globale possa definire la soglia temporale dopo cui un ordine ^G “in transito” cambi il suo stato in “annullato”.	RF17/02	S

Tabella 23: Test di sistema

4.4 Test di accettazione

I test di accettazione vengono eseguiti per verificare che il software soddisfi i requisiti^G concordati con il cliente^G o l'utente finale. Sono spesso definiti sulla base di scenari realistici o specifiche formali (come le User Story o i Criteri di Accettazione). Questi test rappresentano l'ultima fase di verifica prima del rilascio del prodotto^G e determinano se esso può essere considerato “pronto per l'uso”.

Codice	Descrizione	Stato
TdA01	Verificare che all'avvio del sistema il supervisore globale possa registrarsi con successo	TS
TdA02	Verificare che alla registrazione del supervisore globale vengano gestiti gli errori di inserimento dati	TS
TdA03	Verificare la funzionalità di login con le credenziali corrette	TS
TdA04	Verificare che al login dell'utente vengano gestiti gli errori di inserimento dati	TS
TdA05	Verificare funzionalità logout dal sistema	TS
TdA06	Verificare funzionalità di registrazione di un nuovo Supervisore Locale con dati corretti	TS
TdA07	Verificare che alla registrazione del Supervisore Locale vengano gestiti gli errori di inserimento dati	TS

Codice	Descrizione	Stato
TdA08	Verificare l'assegnazione dei magazzini a un Supervisore Locale	TS
TdA09	Verificare l'aggiunta di un nuovo magazzino con indirizzo valido	TS
TdA10	Verificare la modifica dei dati di un magazzino	TS
TdA11	Verificare l'inserimento di una nuova merce con dati corretti	TS
TdA12	Verificare la definizione e la modifica della quantità di una merce	TS
TdA13	Verificare la rimozione di una merce dall'inventario	TS
TdA14	Verificare l'inserimento di un ordine interno (trasferimento tra magazzini)	TS
TdA15	Verificare l'inserimento di un ordine di vendita esterno	TS
TdA16	Verificare l'annullamento degli ordini in attesa o in elaborazione	TS
TdA17	Verificare il tracciamento e la corretta gestione degli stati degli ordini	TS
TdA18	Verificare la visualizzazione dell'inventario globale e per singolo magazzino	TS
TdA19	Verificare la visualizzazione dei report ordini globali e per singolo magazzino	TS
TdA20	Verificare la modifica dei valori di soglia minima e massima	TS
TdA21	Verificare la gestione dei magazzini online/offline	TS
TdA22	Verificare il riassortimento automatico in caso di ordine non soddisfacibile	TS
TdA23	Verificare il riassortimento automatico al raggiungimento della soglia minima di un prodotto	TS
TdA24	Verificare il riassortimento automatico al raggiungimento della soglia massima di un prodotto	TS

Codice	Descrizione	Stato
TdA25	Verificare la corretta gestione di ordini simultanei da parte di un magazzino	TS

Tabella 24: Test di Accettazione

5. Cruscotto di Valutazione

5.1 Valutazione dei costi

Il cruscotto di valutazione dei costi permette di monitorare l'andamento dei costi rispetto a quelli pianificati, evidenziando eventuali scostamenti e permettendo una gestione più efficace del budget. Le metriche principali includono il valore atteso, il costo attuale, il valore effettivo, l'indice di prestazione dei costi e il preventivo al completamento.

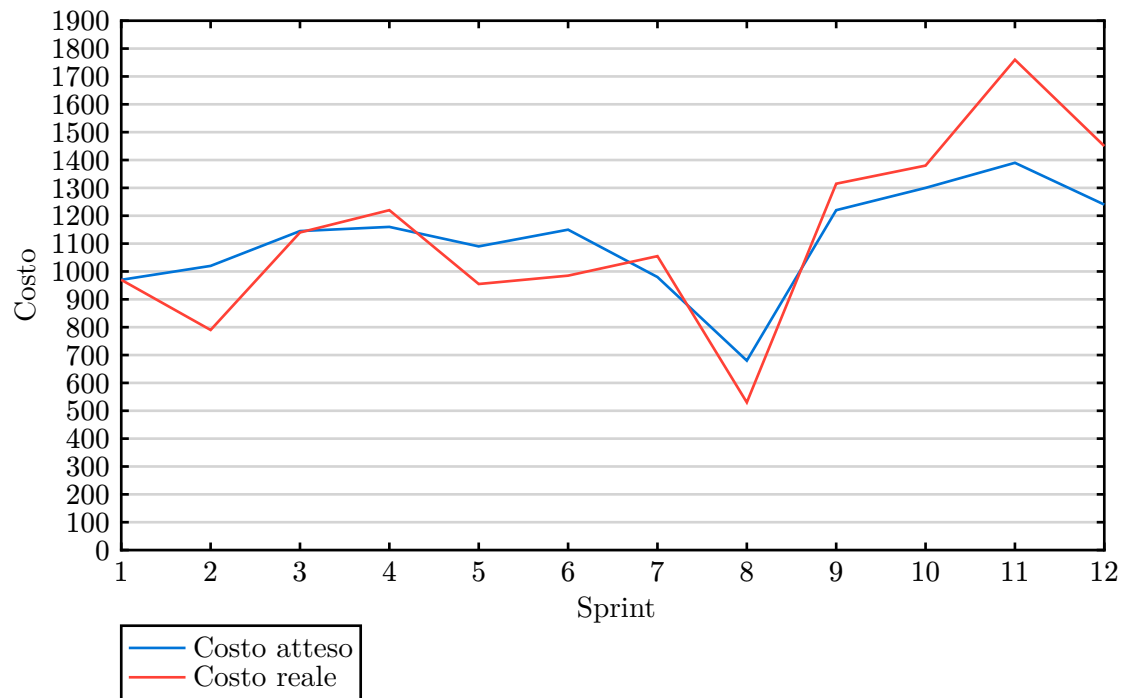


Figura 1: Valutazione dei costi

Possiamo osservare che il costo atteso e il costo reale sono molto vicini, con un leggero scostamento nei primi sprint. Questo indica una buona gestione dei costi fino a questo punto del progetto.

Gli scostamenti sono minimi e non superano il 10% rispetto al costo atteso, il che è un buon segno di stabilità finanziaria del progetto.

Nella fase PB notiamo uno scostamento più significativo tra costo atteso e costo reale, dovuto principalmente alla fase di programmazione dell'MVP.

5.2 Grafico valore atteso ed effettivo

Il grafico del valore atteso ed effettivo mostra l'andamento del valore pianificato rispetto al valore effettivo ottenuto fino ad oggi. Partendo dal budget iniziale di 14.430€, il valore atteso ed effettivo diminuisce progressivamente, in base ai costi sostenuti per ogni sprint. Il valore atteso rappresenta il budget pianificato per ogni sprint, mentre il valore effettivo rappresenta i costi reali sostenuti fino a quel momento.

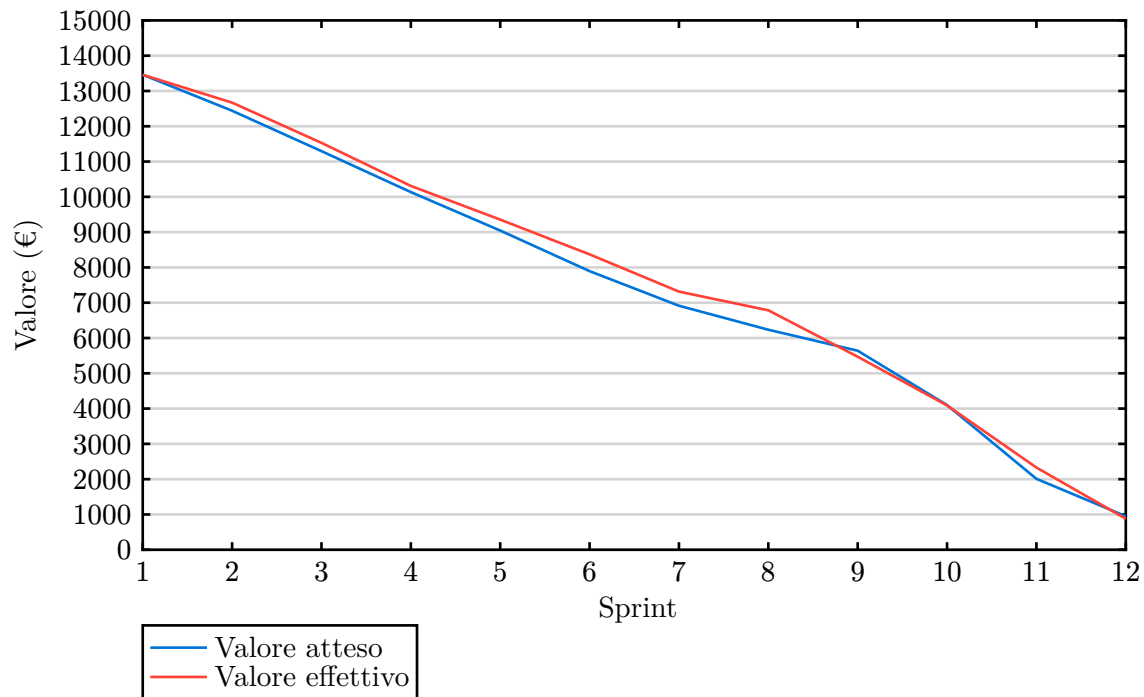


Figura 2: Valore atteso ed effettivo

Dal grafico si evince che il valore atteso diminuisce progressivamente, mentre il valore effettivo mostra una leggera fluttuazione. Questo indica che, sebbene i costi siano stati sostenuti in modo pianificato, ci sono state alcune variazioni nei costi reali rispetto a quelli previsti. Tuttavia, il valore effettivo rimane sempre al di sopra del valore atteso, il che è un segno positivo per la gestione del budget.

Abbiamo sempre preventivato in modo conservativo per evitare di sfiorare il budget iniziale, e questo approccio si è dimostrato efficace fino ad ora.

Infatti, mantenendo questa tendenza riguardo il valore effettivo, possiamo notare che il progetto dovrebbe rimanere entro i limiti del budget iniziale di 14.430€.

Dopo aver ultimato la fase di PB, notiamo che il valore effettivo si avvicina molto al valore atteso, segno che la pianificazione è stata accurata e che i costi sono stati gestiti in modo efficace, portando a un risparmio complessivo di 1.195€ rispetto al budget iniziale.

5.3 Indice di prestazione dei tempi (tempo attuale e tempo pianificato per ogni sprint)

Il grafico dell'indice di prestazione dei tempi mostra l'andamento del tempo pianificato rispetto al tempo effettivo impiegato per completare ogni sprint.

L'indice di prestazione dei tempi (SPI) è calcolato come il rapporto tra il valore pianificato e il valore effettivo. Se l'SPI è maggiore di 1, significa che il progetto sta procedendo più velocemente del previsto; se è inferiore a 1, significa che il progetto sta procedendo più lentamente del previsto.

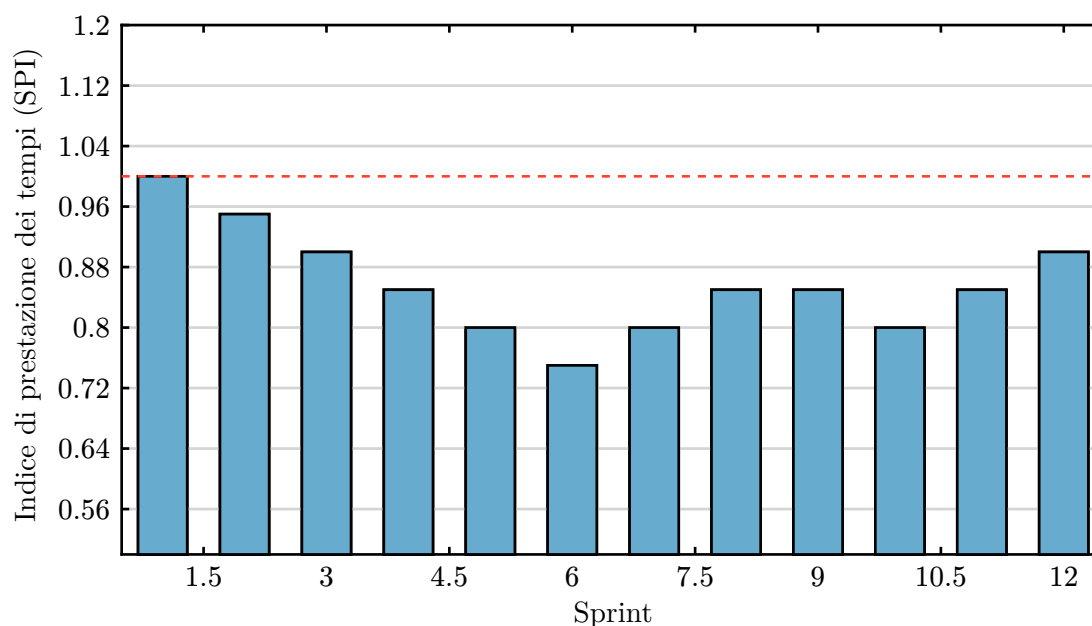


Figura 3: Indice di prestazione dei tempi

L'indice di prestazione dei tempi mostra una tendenza al ribasso, indicando che il progetto sta procedendo più lentamente del previsto. Tuttavia, l'SPI è ancora superiore a 0.5, il che significa che il progetto sta comunque rispettando i tempi pianificati in modo accettabile.

Come si può notare dal grafico, abbiamo subito dei rallentamenti negli sprint centrali, dovuti allo studio delle tecnologie non conosciute e alla complessità dell'Analisi dei Requisiti^G e della progettazione.

Tuttavia, nell'ultimo sprint, che riguarda la fase di sviluppo del PoC, abbiamo recuperato parte del tempo perso, portando l'indice di prestazione dei tempi a 0.8. Il team prevede di mantenere questo ritmo di recupero anche nei prossimi sprint, cercando di migliorare ulteriormente l'efficienza.

Durante la fase di PB, l'indice di prestazione dei tempi è aumentato rispetto agli sprint precedenti, raggiungendo un valore di 0.9 nell'ultimo sprint. Questo indica che il team è riuscito a recuperare parte del tempo perso in precedenza, grazie a una migliore pianificazione e gestione del lavoro, oltre che ad una fase di studio pressoché completata.

5.4 Grafico indice di stabilità dei requisiti

Il grafico dell'indice di stabilità dei requisiti mostra l'andamento della stabilità dei requisiti nel tempo. L'indice di stabilità dei requisiti è calcolato come il rapporto tra il numero di requisiti stabili e il numero totale di requisiti. Un indice più alto indica una maggiore stabilità dei requisiti, mentre un indice più basso indica una minore stabilità.

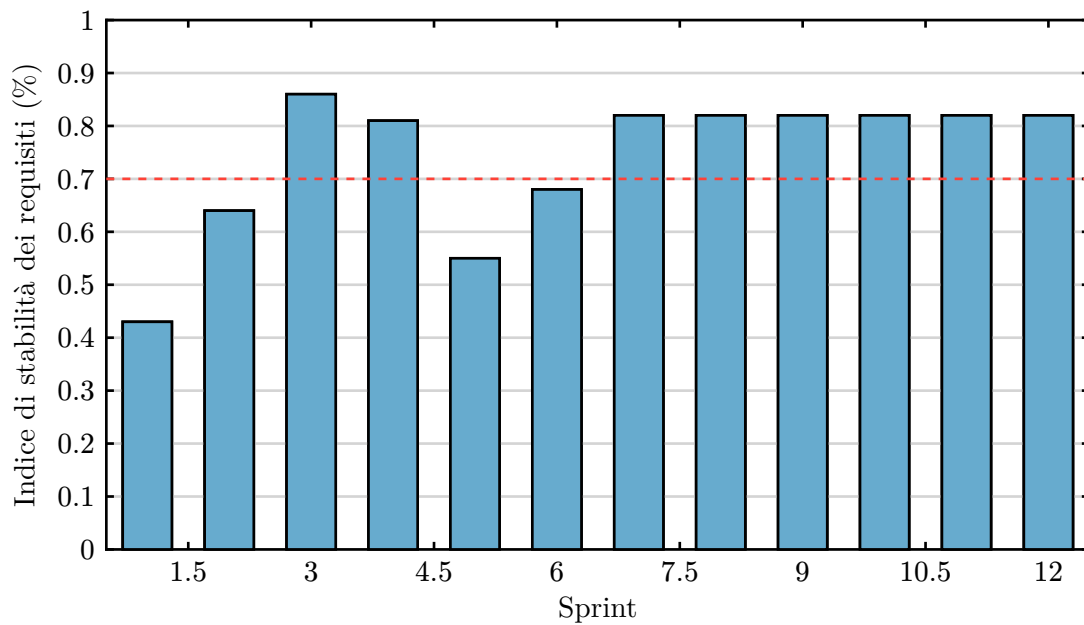


Figura 4: Indice di stabilità dei requisiti

L'indice di stabilità dei requisiti mostra una tendenza al ribasso nel quinto sprint, a causa delle modifiche apportate ai requisiti dopo il primo incontro con il Prof. Riccardo Cardin. Tuttavia, negli sprint successivi, l'indice è aumentato nuovamente, raggiungendo un picco dell'82% nell'ultimo sprint. Questo suggerisce che le modifiche apportate grazie ai consigli del Prof. Riccardo Cardin hanno avuto un impatto positivo sulla stabilità dei requisiti e che il team sta lavorando per mantenere questa stabilità anche nei prossimi sprint.

Durante la fase di PB, l'indice di stabilità dei requisiti è rimasto costante, indicando che non sono state apportate modifiche significative ai requisiti.

5.5 Grafico indice di Gulpease

Il grafico dell'indice di Gulpease mostra l'andamento della leggibilità del documento nel tempo. Un indice più alto indica una maggiore leggibilità, mentre un indice più basso indica una minore leggibilità. L'obiettivo è mantenere l'indice sopra il valore minimo di 50 per garantire che il documento sia facilmente comprensibile.

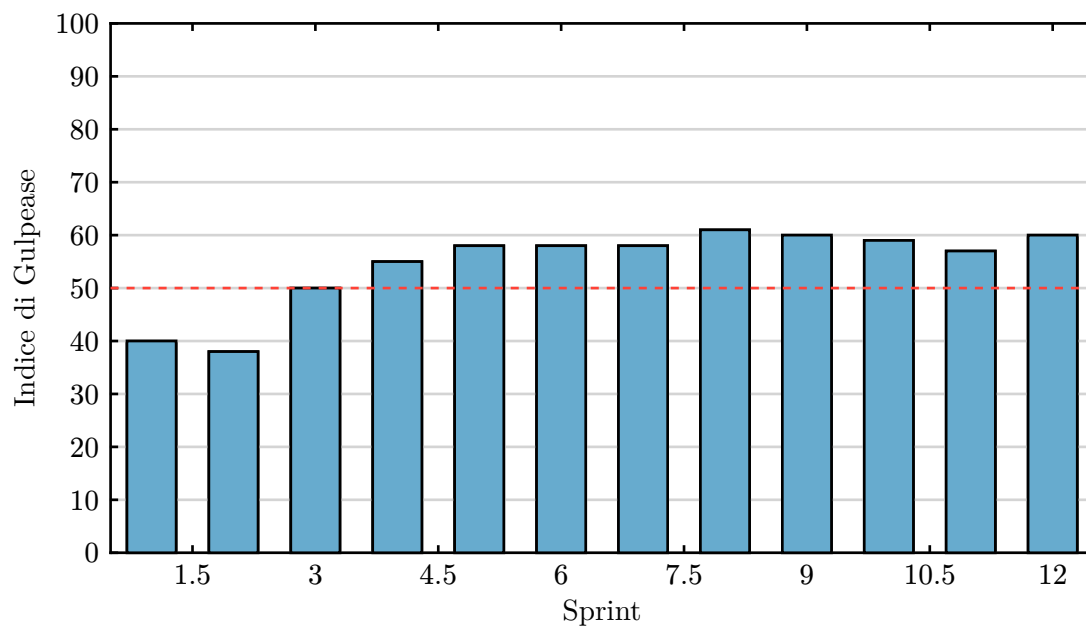


Figura 5: Indice di Gulpease

Inizialmente l'indice di Gulpease era sotto il valore minimo di 50, indicando una bassa leggibilità del documento. Tuttavia, grazie agli sforzi per migliorare la qualità della documentazione, l'indice è aumentato e si è stabilizzato sopra il valore minimo, raggiungendo un picco di 58 negli ultimi sprint. Questo suggerisce che le modifiche apportate hanno avuto un impatto positivo sulla leggibilità del documento.