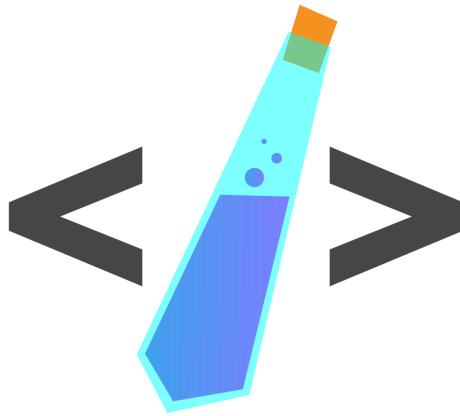




CODE ALCHEMISTS
TURNING IDEAS INTO CODE, LIKE MAGIC

Manuale Utente

Stato

Approvato

Versione

1.0.0

Distribuzione

Code Alchemists

M31



Registro delle Modifiche

Vers.	Data	Autore	Verificatore	Descrizione
1.0.0	12/09/2025	R. Zangla	-	Approvazione interna del documento.
0.1.0	12/09/2025	N. Bolzon	R. Zangla	Inserimento contenuto del documento.
0.0.1	08/09/2025	N. Bolzon	R. Zangla	Creazione template e struttura del documento.

Indice

1. Introduzione	5
1.1. Scopo del documento	5
1.2. Scopo dell'applicativo	5
1.3. Glossario	5
1.4. Riferimenti	5
1.4.1. Riferimenti normativi	5
1.4.2. Riferimenti informativi	5
2. Requisiti di sistema	6
2.1. Requisiti hardware	6
2.2. Requisiti software	6
3. Installazione del sistema	7
3.1. Clonazione della repository	7
3.2. Avvio del sistema	7
3.2.1. Avviare Docker	7
3.2.2. Avviare Docker Compose	7
3.3. Spegnimento	7
3.4. Ripristino	7
4. Funzionalità implementate	9
4.1. Autenticazione	9
4.1.1. Registrarsi come supervisore globale	9
4.1.2. Registrarsi come supervisore locale	9
4.1.3. Accedere come supervisore globale	9
4.1.4. Accedere come supervisore locale	10
4.2. Gestione di magazzini	10
4.2.1. Creazione di un magazzino	10
4.2.2. Visualizzazione dello stato di un magazzino	10
4.3. Gestione prodotti	10
4.3.1. Inserimento prodotto in un Inventario	10
4.3.2. Visualizzazione dati di un singolo prodotto	11
4.3.3. Rimozione prodotti da un Inventario	11
4.3.4. Modifica quantità disponibile di un prodotto da un Inventario	11
4.3.5. Modifica prezzo unitario di un prodotto	12
4.3.6. Modifica dei valori di soglia minima e/o soglia massima di un prodotto	12
4.3.7. Visualizzazione dei prodotti presenti in un magazzino	13
4.3.8. Visualizzazione di tutti i prodotti presenti in rete	13
4.4. Gestione ordini	13
4.4.1. Creazione di un nuovo ordine interno	13
4.4.2. Creazione di un nuovo ordine di vendita	14
4.4.3. Visualizzazione dati di un singolo ordine	15
4.4.3.1. Visualizzazione dello stato attuale di un ordine	15
4.4.4. Visualizzazione di tutti gli ordini in uno specifico magazzino	15
4.4.5. Visualizzazione di tutti gli ordini presenti in rete	15
4.4.6. Annullamento di un ordine	16
5. Esecuzione Test	17
5.1. Test di unità e di integrazione	17
5.2. Test di accettazione	17
6. Telemetria	18

6.1. Monitoraggio delle performance	18
6.2. Visualizzazione dei dati di telemetria	18
6.3. Primo accesso a Grafana	18
6.4. Aggiungere Prometheus come fonte dati in Grafana	18
6.5. Importare una dashboard predefinita	18

1. Introduzione

1.1. Scopo del documento

Lo scopo di questo documento è di riportare le istruzioni per installare correttamente l'applicativo, spiegarne lo scopo generale e illustrare le funzionalità implementate al suo interno.

1.2. Scopo dell'applicativo

Il sistema sviluppato consiste in una piattaforma centralizzata per la gestione ottimizzata e in tempo reale di inventari distribuiti di magazzini che ricevono ordini di vendita verso l'esterno. Esso è specializzato nel prevenire le rotture di stock e bilanciare dinamicamente le scorte tra magazzini geograficamente dislocati, garantendo la continuità operativa delle operazioni di vendita di merce. Inoltre, il sistema mitiga i rischi di interruzione (es. indisponibilità di un magazzino) e ottimizza la distribuzione della merce tra i vari magazzini, mantenendo comunque adeguate scorte di sicurezza.

1.3. Glossario

Al fine di garantire chiarezza ed evitare incomprensioni, è stato realizzato un glossario (consultabile all'indirizzo https://teamcodealchemists.github.io/docs/glossario/Gls_1.0.0.pdf) contenente le definizioni dei vari termini tecnici e concetti specifici del progetto. Tali parole consultabili nel Glossario vengono contrassegnate nel documento con una G in apice (ad es., «autenticazione^G»).

1.4. Riferimenti

1.4.1. Riferimenti normativi

Regolamento del progetto didattico:

<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T01.pdf>

1.4.2. Riferimenti informativi

Capitolato C6: Sistema di Gestione di un Magazzino Distribuito

<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C6.pdf>

Glossario:

https://teamcodealchemists.github.io/docs/glossario/Gls_1.0.0.pdf

2. Requisiti di sistema

2.1. Requisiti hardware

I requisiti hardware per l'installazione e l'esecuzione del sistema sono i seguenti:

Componente	Requisito minimo
«Processore»	Quad-core 2.0 GHz o superiore
«RAM»	8 GB o superiore
«Spazio su disco»	20 GB di spazio libero o superiore
«Connessione di rete»	Connessione Internet stabile per la corretta esecuzione del sistema e per l'accesso ai servizi cloud.

Tabella 1: Requisiti hardware minimi per il sistema

2.2. Requisiti software

In merito al Sistema Operativo, le componenti del sistema non presentano particolari vincoli e possono essere installate su qualsiasi sistema operativo moderno (Windows, macOS, Linux), in quanto il sistema si appoggia completamente a **Docker** per l'esecuzione dei vari servizi.

Per assicurare quindi il corretto funzionamento del sistema, è necessario che il sistema operativo supporti **Docker** e che quest'ultimo sia installato.

Di seguito è riportata una tabella con le versioni minime richieste per i vari software necessari al funzionamento del sistema:

Software	Versione minima richiesta
«Docker»	$\geq 22.16.0$
«Docker Compose»	$\geq 2.37.x$
«Ambiente Docker»	node:22-alpine
«Node.js»	$\geq 22.16.0$
«npm»	$\geq 9.x$

Tabella 2: Requisiti software minimi per il sistema

3. Installazione del sistema

3.1. Clonazione della repository

Per ottenere il codice sorgente del sistema, è necessario clonare la repository GitHub del progetto. Questo può essere fatto utilizzando il comando `git clone` seguito dall'URL della repository.

Esempio di comando:

```
git clone https://github.com/teamcodealchemists/MVP.git
```

Dopodiché, è necessario navigare nella cartella del progetto clonata utilizzando il comando `cd`:

```
cd MVP
```

3.2. Avvio del sistema

3.2.1. Avviare Docker

Avviare **Docker Desktop**, o **Docker Engine** a seconda del sistema operativo in uso.

3.2.2. Avviare Docker Compose

Per avviare il sistema, è necessario eseguire il comando `docker compose up --build -d` dalla cartella principale del progetto (dove è presente il file `docker-compose.yml`):

```
docker compose up --build -d
```

In questo modo, **Docker Compose** costruirà le immagini dei vari servizi (se non sono già state costruite) e avvierà i container in modalità detached (in background). Per verificare che i container siano stati avviati correttamente, è possibile utilizzare il comando:

```
docker ps
```

Questo comando mostrerà una lista dei container in esecuzione. Assicurarsi che tutti i container necessari siano presenti e in stato «Up».

Da questo momento, il sistema è avviato e funzionante.

3.3. Spegnimento

Per arrestare il sistema, è possibile eseguire il comando `docker compose down` dalla cartella principale del progetto:

```
docker compose down
```

Questo comando arresterà e rimuoverà i container, le reti e i volumi creati da **Docker Compose**.

3.4. Ripristino

In caso di problemi con il sistema, è possibile eseguire il comando `docker compose down` dalla cartella principale del progetto per arrestare e rimuovere i container, le reti e i volumi creati da **Docker Compose**:

```
docker compose down
```

Successivamente, è possibile riavviare il sistema eseguendo nuovamente il comando `docker compose up --build -d`. In caso di problemi persistenti, è possibile eliminare i volumi associati ai container utilizzando il comando:

```
docker compose down -v
```

Questo comando rimuoverà anche i volumi, permettendo un ripristino completo del sistema. Tuttavia, si noti che l'eliminazione dei volumi comporta la perdita di tutti i dati memorizzati nei volumi stessi.

4. Funzionalità implementate

4.1. Autenticazione

Il sistema prevede due ruoli di supervisore: il supervisore locale, che gestisce un singolo magazzino, e il supervisore globale, che ha una visione d'insieme di tutti i magazzini in rete.

L'autenticazione avviene tramite token JWT (JSON Web Token), che vengono generati al momento del login e devono essere inclusi nelle richieste successive per accedere alle funzionalità protette del sistema.

4.1.1. Registrarsi come supervisore globale

Al primo avvio del sistema, non sono presenti utenti registrati. La prima registrazione deve essere effettuata come supervisore globale.

Per registrarsi come supervisore globale, è necessario inviare una richiesta POST all'endpoint `/api/auth/register/globalSupervisor` con i seguenti dati:

```
{
  «name»: «Mario»,
  «surname»: «Rossi»,
  «phone»: «+39 355372965»,
  «authentication»: {
    «email»: «email@example.com»,
    «password»: «{{password}}»
  }
}
```

4.1.2. Registrarsi come supervisore locale

Una volta registrato il supervisore globale, quest'ultimo ha la possibilità di registrare nuovi supervisori locali.

Per registrare un supervisore locale, è necessario inviare una richiesta POST all'endpoint `/api/auth/register/localSupervisor` con i seguenti dati:

```
{
  «name»: «Luigi»,
  «surname»: «Verdi»,
  «phone»: «+39 3661234567»,
  «authentication»: {
    «email»: «email@example.com»,
    «password»: «{{password}}»
  }
}
```

4.1.3. Accedere come supervisore globale

Per accedere come supervisore globale, è necessario inviare una richiesta POST all'endpoint `/api/auth/login` con i seguenti dati:

```
{
  «email»: «email@example.com»,
  «password»: «{{password}}»
}
```

4.1.4. Accedere come supervisore locale

Per accedere come supervisore locale, è necessario inviare una richiesta POST all'endpoint /api/auth/login con i seguenti dati:

```
{
  «email»: «email@example.com»,
  «password»: «{{password}}»
}
```

4.2. Gestione di magazzini

4.2.1. Creazione di un magazzino

Il sistema permette la creazione di un nuovo magazzino locale. Per creare un nuovo magazzino è necessario essere un supervisore globale (ovvero aver effettuato accesso come tale).

Presupponendo che i dati del magazzino da creare siano di questo tipo:

```
{
  «state»: «ONLINE»,
  «address»: «Via Roma 123, Padova»
},
```

il comando da inserire per creare un nuovo magazzino è del tipo: POST «http://localhost:8080/api/routing/warehouse/create».

4.2.2. Visualizzazione dello stato di un magazzino

Il sistema permette la visualizzazione dello stato (online/offline) di uno specifico magazzino locale.

Il comando da inserire per visualizzare lo stato di un magazzino specifico è del tipo: GET «http://localhost:8080/api/cloudState/warehouse/1/state/get», dove “1” corrisponde al numero del magazzino interessato.

4.3. Gestione prodotti

4.3.1. Inserimento prodotto in un Inventario

Il sistema permette l'inserimento di un nuovo prodotto nell'inventario di uno specifico magazzino locale. Per aggiungere un nuovo prodotto a un inventario è necessario essere il supervisore locale del magazzino in cui effettuare tale operazione.

Presupponendo che i dati del prodotto da inserire nel magazzino siano di questo tipo:

```
{
  «id»: 123,
  «name»: «Penne»,
  «unitPrice»: 1.99,
  «quantity»: 0,
  «minThres»: 0,
  «maxThres»: 100
},
```

il comando da inserire per aggiungere un nuovo prodotto all'interno di uno specifico inventario è del tipo: POST «http://localhost:8080/api/warehouse/1/stock/new», dove “1” corrisponde al numero di magazzino in cui aggiungere il nuovo prodotto.

Per poter verificare l'avvenuto inserimento del prodotto nell'inventario, è consigliato consultare la sezione «4.3.2 - Visualizzazione dati di un singolo prodotto».

4.3.2. Visualizzazione dati di un singolo prodotto

Il sistema permette la visualizzazione dei dati di un singolo prodotto, presente all'interno di uno specifico magazzino locale.

Il comando da inserire per visualizzare i dati di un prodotto esistente all'interno di un inventario locale è del tipo: GET «http://localhost:8080/api/warehouse/1/stock/123», dove “1” corrisponde al numero di magazzino da cui prelevare i dati del prodotto e “123” è l'id del prodotto interessato.

4.3.3. Rimozione prodotti da un Inventario

Il sistema permette la rimozione di prodotti all'interno dell'Inventario di uno specifico magazzino locale. Per rimuovere un prodotto da un Inventario è necessario essere il supervisore locale del magazzino in cui effettuare tale operazione.

Il comando da inserire per rimuovere un prodotto esistente dall'inventario è del tipo: DELETE «http://localhost:8080/api/warehouse/1/stock/123/delete», dove “1” corrisponde al numero di magazzino da cui rimuovere il prodotto e “123” è l'id del prodotto da eliminare.

Per verificare l'avvenuta rimozione del prodotto dall'inventario, è consigliato consultare la sezione «4.3.2 - Visualizzazione dati di un singolo prodotto» tentando di visualizzare il prodotto rimosso.

4.3.4. Modifica quantità disponibile di un prodotto da un Inventario

Il sistema permette la modifica della quantità disponibile di un prodotto a un Inventario di uno specifico magazzino locale.

Per modificare la quantità di un prodotto è necessario essere il supervisore locale del magazzino in cui effettuare tale operazione.

Presupponendo che i dati già presenti nel magazzino del prodotto da modificare siano del tipo:

```
{  
  «name»: «Penne»,  
  «unitPrice»: 45,  
  «quantity»: 10,  
  «quantityReserved»: 0,  
  «minThres»: 0,  
  «maxThres»: 20  
}
```

e che i dati aggiornati del prodotto da modificare nel magazzino siano del tipo:

```
{  
  «name»: «Penne»,  
  «unitPrice»: 45,  
  «quantity»: 15,  
  «quantityReserved»: 0,  
  «minThres»: 0,  
  «maxThres»: 20  
},
```

il comando da inserire per modificare la quantità disponibile di un prodotto esistente è del tipo: POST «http://localhost:8080/api/warehouse/1/stock/123/set», dove “1” corrisponde al numero di magazzino in cui modificare il prodotto e “123” è l’id del prodotto da aggiornare.

Per verificare l’avvenuta modifica della quantità del prodotto, è consigliato consultare la sezione «4.3.2 - Visualizzazione dati di un singolo prodotto».

4.3.5. Modifica prezzo unitario di un prodotto

Il sistema prevede la modifica del prezzo unitario di un prodotto all’interno del catalogo del magazzino. Per modificare il prezzo unitario di un prodotto è necessario essere il supervisore locale del magazzino in cui effettuare tale operazione.

Presupponendo che i dati già presenti nel magazzino del prodotto da modificare siano del tipo:

```
{
  «name»: «Penne»,
  «unitPrice»: 1.99,
  «quantity»: 10,
  «quantityReserved»: 0,
  «minThres»: 0,
  «maxThres»: 20
}
```

e che i dati una volta aggiornati del prodotto da modificare nel magazzino siano del tipo:

```
{
  «name»: «Penne»,
  «unitPrice»: 4.99,
  «quantity»: 10,
  «quantityReserved»: 0,
  «minThres»: 0,
  «maxThres»: 20
},
```

il comando da inserire per modificare il prezzo unitario di un prodotto esistente è del tipo: POST «http://localhost:8080/api/warehouse/1/stock/123/set», dove “1” corrisponde al numero di magazzino in cui modificare il prodotto e “123” è l’id del prodotto da aggiornare.

Per verificare l’avvenuta modifica del prezzo del prodotto, è consigliato consultare la sezione «4.3.2 - Visualizzazione dati di un singolo prodotto».

4.3.6. Modifica dei valori di soglia minima e/o soglia massima di un prodotto

Il sistema prevede la modifica dei valori di soglia minima e soglia massima di un tipo di prodotto per un magazzino specifico. Per modificare la soglia minima e massima di un prodotto è necessario essere il supervisore locale del magazzino in cui effettuare tale operazione.

Presupponendo che i dati già presenti nel magazzino del prodotto da modificare siano del tipo:

```
{
  «name»: «Penne»,
  «unitPrice»: 1.99,
  «quantity»: 10,
  «quantityReserved»: 0,
  «minThres»: 0,
```

```
    «maxThres»: 20
  }
```

e che i dati aggiornati del prodotto da modificare nel magazzino siano del tipo:

```
{
  «name»: «Penne»,
  «unitPrice»: 1.99,
  «quantity»: 10,
  «quantityReserved»: 0,
  «minThres»: 5,
  «maxThres»: 30
},
```

il comando da inserire per modificare le soglie di un prodotto esistente è del tipo: POST «http://localhost:8080/api/warehouse/1/stock/123/set», dove “1” corrisponde al numero di magazzino in cui modificare il prodotto e “123” è l’id del prodotto da aggiornare.

Per verificare l’avvenuta modifica delle soglie del prodotto, è consigliato consultare la sezione «4.3.2 - Visualizzazione dati di un singolo prodotto».

4.3.7. Visualizzazione dei prodotti presenti in un magazzino

Il sistema prevede la visualizzazione dei prodotti presenti in un magazzino specifico.

Il comando da inserire per visualizzare tutti i prodotti presenti in un inventario locale è del tipo: GET «http://localhost:8080/api/warehouse/1/inventory», dove “1” corrisponde al numero di magazzino di cui visualizzare l’intero inventario.

4.3.8. Visualizzazione di tutti i prodotti presenti in rete

Il sistema prevede la visualizzazione di tutti i prodotti presenti in rete, aggregando i dati da tutti i magazzini.

Il comando da inserire per visualizzare tutti i prodotti presenti nella rete di magazzini è del tipo: GET «http://localhost:8080/api/aggregatedWarehouses/all», che restituirà l’elenco completo di tutti i prodotti disponibili in tutti i magazzini.

4.4. Gestione ordini

4.4.1. Creazione di un nuovo ordine interno

Il sistema prevede la creazione di un nuovo ordine interno tra magazzini esistenti in rete. Per creare un ordine interno è necessario essere il supervisore locale del magazzino di partenza.

Presupponendo che i dati dell’ordine interno da creare siano di questo tipo:

```
{
  «orderId»: {
    «id»: «»
  },
  «items»: [
    {
      «item»: {
        «itemId»: {
          «id»: «1»
        },
        «quantity»: 50
      }
    }
  ]
}
```

```

    },
    «quantityReserved»: 0,
    «unitPrice»: 1.22
  }
],
«orderState»: {
  «orderState»: «PENDING»
},
«creationDate»: «2024-01-16T14:20:00.000Z»,
«warehouseDeparture»: 1,
«warehouseDestination»: 2,
«sellOrderReference»: {
  «id»: «S35d99099-5818-4fee-8df7-d93129eaa869»
}
},

```

il comando da inserire per creare un nuovo ordine interno è del tipo: POST «http://localhost:8080/api/warehouse/1/order/internal/new», dove “1” corrisponde al numero del magazzino di partenza.

Note: l’id dell’orderId da inserire alla creazione è sempre vuoto in quanto viene compilato durante l’esecuzione della creazione stessa. Inoltre, l’id di sellOrderReference dipende dall’id dell’ordine di vendita corrispondente per cui è stato generato questo ordine interno. Infine, l’orderState, al momento della creazione, è sempre «PENDING».

4.4.2. Creazione di un nuovo ordine di vendita

Il sistema prevede la creazione di un nuovo ordine di vendita verso clienti esterni.

Presupponendo i dati dell’ordine di vendita da creare siano di questo tipo:

```

{
  «orderId»: {
    «id»: «»
  },
  «items»: [
    {
      «item»: {
        «itemId»: {
          «id»: «1»
        },
        «quantity»: 50
      },
      «quantityReserved»: 0,
      «unitPrice»: 1.22
    }
  ],
  «orderState»: {
    «orderState»: «PENDING»
  },
  «creationDate»: «2024-01-16T14:20:00.000Z»,
  «warehouseDeparture»: 1,

```

```
    «destinationAddress»: «Via Roma, 123»,  
  },
```

il comando da inserire per creare un nuovo ordine di vendita è del tipo: POST «http://localhost:8080/api/warehouse/1/order/sell/new», dove “1” corrisponde al numero del magazzino di partenza.

Note: l’id dell’orderId da inserire alla creazione è sempre vuoto in quanto viene compilato durante l’esecuzione della creazione stessa. Inoltre, l’orderState, al momento della creazione, è sempre «PENDING».

4.4.3. Visualizzazione dati di un singolo ordine

Il sistema prevede la visualizzazione dei dati di un singolo ordine, sia cercandolo dal suo magazzino locale di partenza, che cercandolo dall’aggregato cloud Ordini.

Il comando da inserire per visualizzare i dati di un ordine specifico cercandolo dal suo magazzino di partenza è del tipo: GET «http://localhost:8080/api/warehouse/1/order/I55e624c5-a87e-4a26-b956-5ed78586987a», dove “1” corrisponde al numero del magazzino di partenza, in cui risiede l’ordine, e “I55e624c5-a87e-4a26-b956-5ed78586987a” è l’id dell’ordine interessato.

Invece, il comando da inserire per visualizzare i dati di un ordine specifico cercandolo dall’aggregato cloud Ordini è del tipo: GET «http://localhost:8080/api/aggregate/order/I55e624c5-a87e-4a26-b956-5ed78586987a», dove “I55e624c5-a87e-4a26-b956-5ed78586987a” è l’id dell’ordine interessato.

4.4.3.1. Visualizzazione dello stato attuale di un ordine

Il sistema prevede la visualizzazione dello stato attuale di un ordine specifico, sia cercandolo dal suo magazzino locale di partenza, che cercandolo dall’aggregato cloud Ordini.

Il comando da inserire per visualizzare lo stato di un ordine specifico cercandolo dal suo magazzino di partenza è del tipo: GET «http://localhost:8080/api/warehouse/1/order/S35d99099-5818-4fee-8df7-d93129eaa869/state», dove “1” corrisponde al numero del magazzino in cui risiede l’ordine e “S35d99099-5818-4fee-8df7-d93129eaa869” è l’id dell’ordine interessato.

Invece, il comando da inserire per visualizzare lo stato di un ordine specifico cercandolo dall’aggregato cloud Ordini è del tipo: GET «http://localhost:8080/api/aggregate/order/S35d99099-5818-4fee-8df7-d93129eaa869/state», dove “1” corrisponde al numero del magazzino in cui risiede l’ordine e “S35d99099-5818-4fee-8df7-d93129eaa869” è l’id dell’ordine interessato.

4.4.4. Visualizzazione di tutti gli ordini in uno specifico magazzino

Il sistema prevede la visualizzazione di tutti gli ordini in uno specifico magazzino.

Il comando da inserire per visualizzare tutti gli ordini di un magazzino specifico è del tipo: GET «http://localhost:8080/api/warehouse/1/orders», dove “1” corrisponde al numero del magazzino di cui visualizzare tutti gli ordini.

4.4.5. Visualizzazione di tutti gli ordini presenti in rete

Il sistema prevede la visualizzazione di tutti gli ordini presenti in rete.

Il comando da inserire per visualizzare tutti gli ordini presenti in rete è del tipo: GET «http://localhost:8080/api/aggregate/orders», che restituirà l’elenco completo di tutti gli ordini in tutti i magazzini in rete.

4.4.6. Annullamento di un ordine

Il sistema prevede la possibilità di annullamento di un ordine esistente (effettuabile con successo se tale ordine si trova in stato PENDING, PROCESSING o SHIPPED). Per annullare un ordine è necessario essere il supervisore locale del magazzino in cui risiede l'ordine.

Il comando da inserire per annullare un ordine specifico è del tipo: POST «<http://localhost:8080/api/warehouse/1/order/I55e624c5-a87e-4a26-b956-5ed78586987a/cancel>», dove “1” corrisponde al numero del magazzino di partenza, in cui risiede l'ordine, e “I55e624c5-a87e-4a26-b956-5ed78586987a” è l'id dell'ordine da annullare.

Per verificare l'avvenuta cancellazione dell'ordine, è consigliato consultare la sezione «4.4.3 - Visualizzazione dati di un singolo ordine».

5. Esecuzione Test

5.1. Test di unità e di integrazione

I test di unità e di integrazione sono stati eseguiti utilizzando il framework **Jest**.

I test sono stati organizzati in file separati per ciascun modulo del sistema e sono eseguibili tramite il comando:

```
npm run test
```

Le informazioni riguardanti l'esecuzione dei test di unità e di integrazione sono consultabili nel **Piano di Qualifica**.

Per verificare la copertura del codice, è possibile eseguire il comando:

```
npm run test:cov
```

Questo comando genererà un report di copertura del codice per il modulo testato.

La copertura totale del codice è consultabile dal **README.md** della pagina di *GitHub* del **MVP**: tramite *CodeCov* è stato inserito un badge che riporta la percentuale di copertura totale del codice.

5.2. Test di accettazione

In accordo con **M31**, è stato stilato un **TestBook** contenente una serie di test di accettazione volti a verificare il corretto funzionamento del sistema secondo le specifiche concordate.

Il **TestBook** è consultabile all'indirizzo <https://daconfermare>.

I test di accettazione sfruttano NATS e sono eseguibili tramite **Postman**.

Poiché Postman richiede una versione a pagamento per poter essere eseguito in modalità automatica, abbiamo deciso di eseguire i test tramite **Newman**, un tool da linea di comando che permette di eseguire le collezioni di Postman.

Per eseguire i test di accettazione, è necessario installare **Newman** globalmente tramite npm:

```
npm install -g newman
```

Successivamente, è possibile eseguire i test utilizzando il comando:

```
newman run https://www.postman.com/collections/<collection-id>
```

Nota: Si può anche usare l'opzione *-verbose* per ottenere più informazioni sul ritorno delle richieste.

Le informazioni riguardanti l'esecuzione dei test di accettazione sono consultabili nel **TestBook**.

6. Telemetria

6.1. Monitoraggio delle performance

Il sistema integra **Prometheus** per il monitoraggio delle performance dei servizi di ordini aggregato e di inventario aggregato.

Prometheus raccoglie metriche in tempo reale, consentendo di analizzare l'utilizzo delle risorse e identificare eventuali colli di bottiglia.

6.2. Visualizzazione dei dati di telemetria

I dati raccolti da **Prometheus** possono essere visualizzati tramite Grafana, uno strumento di analisi e monitoraggio.

L'interfaccia di **Prometheus** è accessibile all'indirizzo `http://localhost:9090`, mentre l'interfaccia di **Grafana** è accessibile all'indirizzo `http://localhost:3210`.

6.3. Primo accesso a Grafana

Al primo accesso a **Grafana**, è necessario utilizzare le seguenti credenziali:

- Username: admin
- Password: admin

Al primo accesso, **Grafana** richiederà di cambiare la password. Successivamente, è possibile accedere alla dashboard di monitoraggio delle performance dei servizi.

6.4. Aggiungere Prometheus come fonte dati in Grafana

Per visualizzare i dati di telemetria in **Grafana**, è necessario aggiungere **Prometheus** come fonte dati.

Questo può essere fatto seguendo questi passaggi:

1. Accedere a **Grafana**.
2. Cliccare sull'icona dell'ingranaggio (⚙️) nel menu laterale per accedere alle impostazioni.
3. Selezionare «Data Sources» (Fonti dati).
4. Cliccare su «Add data source» (Aggiungi fonte dati).
5. Selezionare «Prometheus» dall'elenco delle fonti dati disponibili.
6. Configurare l'URL di **Prometheus** (di default `http://localhost:9090`).
7. Cliccare su «Save & Test» (Salva e testa) per verificare la connessione.

6.5. Importare una dashboard predefinita

Per visualizzare i dati di telemetria in modo efficace, è possibile importare una dashboard predefinita in **Grafana**.

Questo può essere fatto seguendo questi passaggi:

1. Accedere a **Grafana**.
2. Cliccare sull'icona «+» nel menu laterale e selezionare «Import» (Importa).
3. Inserire l'ID della dashboard predefinita o caricare un file JSON della dashboard.
4. Selezionare la fonte dati **Prometheus** appena configurata.
5. Cliccare su «Import» (Importa) per aggiungere la dashboard.

A questo punto, la dashboard sarà disponibile per la visualizzazione e l'analisi dei dati di telemetria raccolti da **Prometheus**. Si noti che le dashboard possono essere personalizzate ulteriormente in base alle esigenze specifiche di monitoraggio.