



M3I

**GC-0003.02**

# **Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto**

Authors: Luca Cossaro

Confidentiality: Public

Last Modified: 25/09/2024 10:04

Release Status: Released

M3I-UUID: NA



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	2/13

## CONTROLLO DOCUMENTALE

<b>Titolo</b>	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto
<b>Classificazione</b>	Public
<b>Stato del Rilascio</b>	Released
<b>Data</b>	25/09/2024
<b>Riferimento M31</b>	Luca Cossaro E-mail: luca.cossaro@m31.com Tel: +39 049 7811 811

## LISTA DI DISTRIBUZIONE

Società	Nome Cognome	Società	Nome Cognome
M31 S.r.l.			

## TABELLA DELLE REVISIONI

Rev	Data	Descrizione	Autori	Verificata	Approvata
01	24/09/2024	Prima revisione	Luca Cossaro	Sandro Lain	Luca Cossaro
02	25/09/2024	Seconda revisione	Luca Cossaro	Sandro Lain	Luca Cossaro
<b>Nome File:</b>	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto.docx				



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	3/13

## INDICE

<b>CONTROLLO DOCUMENTALE</b> .....	<b>2</b>
<b>LISTA DI DISTRIBUZIONE</b> .....	<b>2</b>
<b>TABELLA DELLE REVISIONI</b> .....	<b>2</b>
<b>INDICE</b> .....	<b>3</b>
<b>1 CONTESTO E OBIETTIVO DEL PROGETTO</b> .....	<b>4</b>
<b>2 DESCRIZIONE DEL SISTEMA</b> .....	<b>4</b>
<b>3 ARCHITETTURA</b> .....	<b>5</b>
<b>4 TECNOLOGIE DI RIFERIMENTO</b> .....	<b>7</b>
4.1 RIFERIMENTI.....	7
<b>5 REQUISITI</b> .....	<b>8</b>
5.1 REQUISITI FUNZIONALI MINIMI .....	8
5.2 REQUISITI FUNZIONALI OPZIONALI.....	9
5.3 REQUISITI DI SICUREZZA .....	9
5.4 REQUISITI DI SICUREZZA OPZIONALI.....	9
5.5 REQUISITI NON FUNZIONALI .....	10
<b>6 DELIVERABLE DEL PROGETTO</b> .....	<b>10</b>
6.1 ANALISI DELLO STATO DELL'ARTE.....	10
6.2 PROOF OF CONCEPT (POC) .....	10
6.3 MINIMUM VIABLE PRODUCT (MVP) .....	11
<b>7 TEST E VALIDAZIONE</b> .....	<b>11</b>
<b>8 DOCUMENTAZIONE</b> .....	<b>12</b>



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	4/13

## 1 CONTESTO E OBIETTIVO DEL PROGETTO

La gestione ottimale dell'inventario in una rete logistica distribuita è fondamentale per garantire la disponibilità continua delle risorse lungo tutta la catena operativa. In particolare, un contesto in cui magazzini geograficamente distribuiti devono mantenere un flusso costante di materiali e prodotti, richiede un sistema di gestione in grado di minimizzare i tempi di risposta e di ottimizzare la distribuzione delle scorte.

Ad esempio, immaginiamo di avere tre magazzini diversi: A, B e C. Se il magazzino di A sta per esaurire un prodotto critico e quello di C ne ha in eccesso, un sistema ben progettato dovrebbe essere in grado di rilevare rapidamente questa situazione e trasferire le scorte da C ad A prima che si verifichi una rottura di stock. Questo garantisce che tutti i magazzini siano bilanciati e che le operazioni possano proseguire senza interruzioni. Un altro esempio significativo consiste nel caso in cui il magazzino B non sia più raggiungibile e che quindi si debba procedere ad una movimentazione di merci per rifornire altri magazzini, localizzati nelle sue vicinanze, in modo che questi possano sopperire al disservizio di B.

Il sistema richiesto deve fornire a chi gestisce la rete di magazzini una visibilità centralizzata e in tempo reale dell'inventario distribuito tra più sedi, consentendo una gestione fluida delle operazioni logistiche, tra cui il riassortimento, il trasferimento di prodotti tra le diverse sedi e la riduzione dei livelli medi di scorte, pur mantenendo scorte di sicurezza per prevenire carenze dovute a variazioni della domanda o ritardi nella consegna. Sarà altresì fondamentale integrare meccanismi che permettano di affrontare la variabilità della domanda e i tempi di consegna con modelli predittivi e ottimizzati per la gestione delle scorte. Questa attività sarà subordinata al reperimento da parte di M31 di almeno un dataset pubblico su cui basare l'analisi e lo sviluppo delle soluzioni, M31 si occuperà di mettere a disposizione una figura tecnica esperta in materia che possa guidare questa attività.

Il progetto si propone di sviluppare un sistema distribuito, scalabile, basato su architettura a microservizi, che favorisca l'interoperabilità tra i diversi magazzini e la centralizzazione delle informazioni in modo efficiente e sicuro. Tale sistema sarà capace di gestire operazioni distribuite, garantendo prestazioni elevate anche in scenari di elevato carico di dati e richieste simultanee.

## 2 DESCRIZIONE DEL SISTEMA

Il sistema di gestione dovrà essere in grado di:

- Ottimizzare i livelli di scorte:** Il sistema dovrà monitorare costantemente i livelli di inventario, suggerendo o automatizzando azioni di riassortimento o trasferimento tra magazzini al fine di mantenere scorte minime senza compromettere l'operatività. M31 si occuperà di produrre e mettere a disposizione un simulatore di movimenti di inventario.
- Gestire la condivisione dei dati in tempo reale:** Attraverso una sincronizzazione continua, sarà possibile avere una visione chiara e centralizzata delle scorte presenti in ogni magazzino, con aggiornamenti in tempo reale riguardo alle operazioni di prelievo, ricezione e trasferimento.



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	5/13

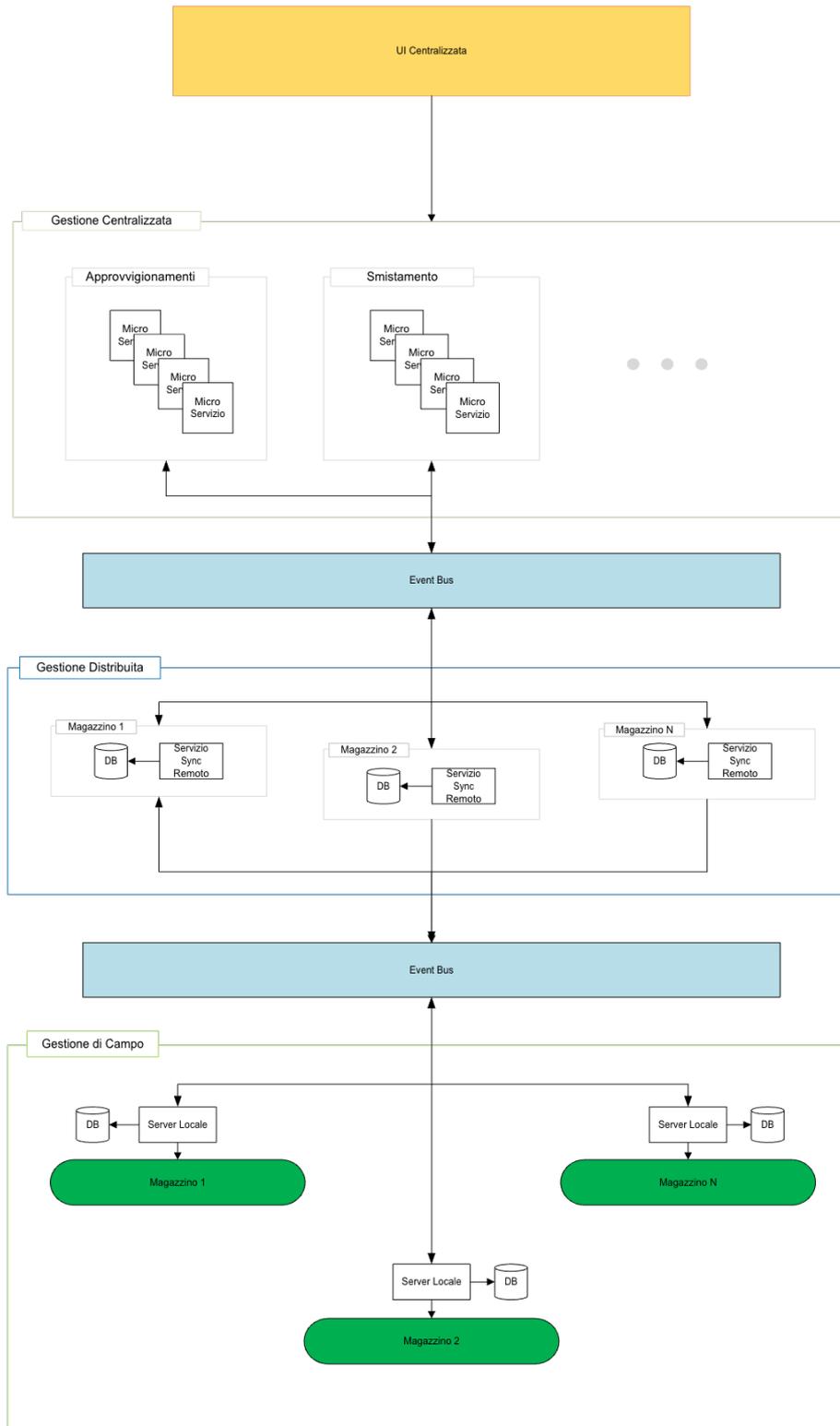
- Implementare riassortimento predittivo:** L'utilizzo di algoritmi di machine learning permetterà di prevedere la domanda futura in base a dati storici, stagionalità e pattern di consumo. Questo approccio ridurrà il rischio di esaurimento scorte e ottimizzerà l'allocazione dei materiali tra magazzini.
- Risolvere i conflitti di aggiornamento simultaneo:** Saranno implementati meccanismi che gestiranno le situazioni in cui aggiornamenti simultanei dell'inventario, provenienti da magazzini differenti, potrebbero creare discrepanze nei dati (ad esempio utilizzando tecniche di versionamento o timestamp). Le discrepanze nei dati possono essere generate, ad esempio, quando un magazzino riceve contemporaneamente due ordini distinti per lo stesso prodotto e quindi cerca di riservare le scorte contemporaneamente, portando ad una errata valutazione della possibilità di soddisfare entrambi gli ordini.

### 3 ARCHITETTURA

L'architettura attualmente ipotizzata è basata su microservizi. Ogni magazzino fungerà da nodo autonomo (edge node, cioè un server situato ai margini della rete, vicino agli utenti o alle risorse fisiche, il suo ruolo è elaborare i dati localmente, riducendo la latenza e il carico sulla rete centrale, migliorando l'efficienza e la velocità delle operazioni), gestito da un insieme di servizi (sia su edge che su cloud) distribuiti che permetteranno di gestire le operazioni di inventario in maniera indipendente e allo stesso tempo sincronizzata con il sistema centrale. I servizi saranno liberamente accoppiati, garantendo che ogni magazzino possa operare in maniera autonoma, gestendo il proprio database locale, e sincronizzandosi con gli altri nodi per garantire l'allineamento globale dei dati.

Ogni magazzino sarà sincronizzato con la sua copia remota (servizio autonomo, facente parte della gestione distribuita, cfr. Figura 1, che risiede nell'infrastruttura cloud) che fornirà i dati raccolti al sistema centralizzato di gestione del magazzino globale. Anche questa applicazione di gestione remota dovrà essere realizzata mediante microservizi, al fine di garantire la scalabilità ed affidabilità del sistema.

Un'attenzione particolare sarà dedicata alla progettazione architeturale, che dovrà essere ben documentata e presentata con appositi **diagrammi a blocchi**. L'architettura finale non sarà imposta dall'azienda proponente, ma dovrà essere il frutto di un processo decisionale basato sull'analisi dei requisiti e delle esigenze tecniche del contesto distribuito.



**Figura 1- Ipotesi di Architettura**



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	7/13

## 4 TECNOLOGIE DI RIFERIMENTO

Nel progetto proposto, sarà fondamentale utilizzare tecnologie scalabili e adatte alla gestione di sistemi distribuiti, in linea con quelle già adottate da M31. **Node.js** e **Nest.js** (usando **TypeScript** come linguaggio) verranno impiegati per lo sviluppo dei microservizi, grazie alla loro capacità di creare applicazioni server-side performanti e modulari. **Go** potrà essere utilizzato per eventuali componenti ad alte prestazioni, come i servizi di sincronizzazione. La comunicazione tra i microservizi sarà facilitata da **NATS** o **Apache Kafka**, entrambe tecnologie robuste per la gestione di messaggi distribuiti e asincroni. Il sistema di orchestrazione e gestione centralizzata sarà ospitato su **Google Cloud Platform**, che offre servizi cloud affidabili e scalabili in particolare **Kubernetes**. Per l'archiviazione dei dati, si farà uso di **MongoDB** per la memorizzazione di dati non strutturati, mentre **PostgreSQL** potrà essere impiegato per la persistenza di dati strutturati. **Redis** potrà essere utilizzato come sistema di caching per migliorare le prestazioni e ridurre la latenza. Per quanto riguarda la parte di interfaccia utente si utilizza **Angular** e generalmente SPAs (Single Page Applications) per fornire agli utenti una esperienza il più possibile simile a quella di un'applicazione desktop.

Le tecnologie qui presentate non sono vincolanti, ma nel caso ne venissero proposte di alternative, queste scelte dovranno essere motivate ed esplicitamente accettate da M31. Lo stesso vale per eventuali aree tecnologiche non coperte da strumenti già in utilizzo presso M31.

### 4.1 Riferimenti

Tecnologie, linguaggi, strumenti e framework:

- NodeJS: <https://nodejs.org/>
- Nest.js: <https://nestjs.com/>
- TypeScript: <https://www.typescriptlang.org/>
- Go: <https://go.dev/>
- NATS: <https://nats.io/>
- Kafka: <https://kafka.apache.org/>
- Google Cloud Platform: <https://cloud.google.com/gcp>
- Kubernetes: <https://kubernetes.io/>
- PostgreSQL: <https://www.postgresql.org/>
- MongoDB: <https://www.mongodb.com/>
- Redis: <https://redis.io/>
- Angular: <https://angular.dev/>
- Architettura a Microservizi: <https://microservices.io/>

Dataset e progetti simili:

- <https://github.com/austinlasseter/datasets-shipping-logistics?tab=readme-ov-file>
- <https://www.kaggle.com/datasets/aikinomichi/mega-star-distribution-centre>
- <https://www.kaggle.com/datasets/muhammadshahrayar/transactional-retail-dataset-of-electronics-store>

M31 S.r.l.

via Tommaseo, 77  
35131 Padova  
Italia

E-mail info@m31.com  
PEC [m31.srl@legalmail.it](mailto:m31.srl@legalmail.it)  
Tel +39 049 781 1811

R.E.A. PD-0390285  
C.F./P.I. 04447620289  
www.m31.com



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	8/13

## 5 REQUISITI

### 5.1 Requisiti Funzionali Minimi

- Sincronizzazione in tempo reale dei dati di inventario:** il sistema deve permettere la sincronizzazione in tempo reale dei dati di inventario tra i vari magazzini distribuiti e il sistema centrale. Questo significa che ogni cambiamento nelle quantità di stock, come entrate o uscite di merci, deve essere immediatamente registrato e comunicato, garantendo che ogni magazzino e il sistema centrale abbiano una visione coerente e aggiornata dell'inventario.
- Gestione autonoma dei magazzini tramite microservizi:** ogni magazzino sarà responsabile della gestione locale del proprio inventario, utilizzando un set di microservizi autonomi che possono operare indipendentemente. I servizi locali dovranno gestire le operazioni di inserimento, modifica, trasferimento e prelievo delle merci, aggiornando contemporaneamente il database locale e trasmettendo i dati al sistema centrale per la sincronizzazione.
- Risoluzione dei conflitti negli aggiornamenti concorrenti:** in scenari in cui due o più magazzini modificano simultaneamente lo stesso elemento di inventario (ad esempio, due ordini contemporanei che richiedono lo stesso prodotto per cui non c'è una riserva sufficiente per entrambi gli ordini), il sistema dovrà gestire automaticamente i conflitti, utilizzando tecniche di controllo di versione o timestamp per determinare quale aggiornamento prevale o per conciliare le modifiche.
- Riassortimento predittivo basato su machine learning:** il sistema dovrà implementare algoritmi di machine learning per analizzare i dati storici e prevedere la domanda futura di ciascun prodotto. Questi algoritmi dovranno essere in grado di considerare vari fattori, come la stagionalità e le tendenze di consumo, al fine di suggerire quando e quanto riassortire le scorte, riducendo al minimo il rischio di esaurimento o sovrastoccaggio.
- Monitoraggio centralizzato e reportistica:** il sistema centrale dovrà fornire una piattaforma di monitoraggio che consenta di visualizzare in tempo reale lo stato delle scorte in tutti i magazzini della rete. Questa piattaforma dovrà includere la generazione automatica di report dettagliati sugli stock e avvisi personalizzabili, ad esempio quando il livello di un prodotto scende sotto una soglia predefinita.
- Gestione delle operazioni di trasferimento tra magazzini:** il sistema deve permettere la gestione delle operazioni di trasferimento di merce da un magazzino all'altro. In particolare, sarà necessario poter generare e gestire richieste di trasferimento di prodotto, visualizzando lo stato del trasferimento in tempo reale e aggiornando automaticamente l'inventario sia del magazzino di partenza che di quello di destinazione.



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	9/13

## 5.2 Requisiti Funzionali Opzionali

- Integrazione con sistemi di notifica e alert:** il sistema deve essere in grado di integrarsi con strumenti di notifica per inviare avvisi agli operatori in caso di eventi critici, come il raggiungimento di livelli minimi di scorte o la necessità di approvare trasferimenti di merci tra magazzini. Gli alert dovranno essere configurabili in base a regole predefinite e dovranno supportare canali di comunicazione multipli (e-mail, SMS, dashboard interna).
- Sicurezza e controllo degli accessi:** il sistema dovrà prevedere meccanismi di autenticazione e autorizzazione per garantire che solo personale autorizzato possa accedere e modificare i dati di inventario. Questo include la gestione di ruoli e permessi granulari per operatori, supervisor e amministratori del sistema.
- Backup e ripristino dei dati:** sarà implementata una strategia di backup regolare che coinvolge sia i sistemi distribuiti dei magazzini sia il cloud centralizzato. Il sistema centrale gestirà i processi di integrità e riconciliazione dei dati tra i magazzini, mentre ciascun sistema manterrà copie locali dei propri dati. In caso di guasti o perdite di dati, sarà possibile ripristinare lo stato di tutti i sistemi, garantendo l'integrità e la coerenza complessiva delle informazioni critiche.

## 5.3 Requisiti di Sicurezza

- Protezione dei dati e confidenzialità nelle comunicazioni:** Implementare la cifratura end-to-end per tutti i dati scambiati tra il sistema centrale e i sistemi locali, utilizzando protocolli sicuri come TLS.
- Monitoraggio del sistema:** Implementare sistemi di monitoraggio continuo per rilevare e rispondere tempestivamente a tentativi di accesso non autorizzato o attività sospette.

## 5.4 Requisiti di Sicurezza Opzionali

- Sicurezza dei microservizi:** Implementare l'autenticazione tra microservizi utilizzando, ad esempio, JWT e mTLS per garantire che solo microservizi autorizzati possano comunicare tra loro. Cifrare le comunicazioni tra i microservizi per proteggere i dati in transito. Isolare i microservizi con funzioni distinte, applicando il principio del minimo privilegio per ridurre l'impatto di eventuali compromissioni. Monitorare e gestire le dipendenze tra microservizi per prevenire vulnerabilità derivanti da componenti esterni.
- Resilienza del sistema:** Garantire la resilienza del sistema con backup regolari, ridondanza e piani di ripristino in caso di attacchi o malfunzionamenti, mantenendo la continuità operativa e minimizzando l'impatto di eventuali incidenti.
- Protezione dei dati avanzata:** Assicurare che l'autenticazione sia robusta tramite MFA e certificati crittografici, garantendo che solo utenti e dispositivi autorizzati abbiano accesso. Verificare l'integrità dei dati attraverso firme digitali o hashing, per prevenire manomissioni.



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	10/13

## 5.5 Requisiti non Funzionali

15. **Unit test e code coverage:** lo sviluppo del software dovrà includere test unitari per garantire la correttezza delle funzioni sviluppate. I test dovranno coprire le funzionalità critiche, con una percentuale minima di copertura del codice concordata con M31. I test dovranno essere eseguibili in maniera automatizzata per garantire che ogni nuovo cambiamento o aggiornamento del codice non introduca errori.
16. **Test book dettagliato:** la fornitura di un **test book** sarà essenziale per documentare i test eseguiti e i risultati ottenuti. Questo documento dovrà includere una descrizione dei casi di test, delle condizioni iniziali, dei passaggi eseguiti, dei risultati attesi e ottenuti, nonché dei criteri di validazione utilizzati per verificare il corretto funzionamento del sistema. Il test book sarà la base per la verifica finale e l'accettazione del progetto.
17. **Scalabilità dei microservizi:** l'architettura del sistema dovrà essere progettata per essere scalabile orizzontalmente, in modo che ogni magazzino possa aumentare o diminuire le proprie risorse in base al carico di lavoro senza compromettere le prestazioni del sistema complessivo. Questo richiede che i microservizi siano progettati con l'obiettivo di essere modulari e distribuibili su più istanze in cloud.
18. **Versionamento del codice:** il codice prodotto dovrà essere sottoposto a versionamento mediante Git. M31 metterà a disposizione i repository necessari allo svolgimento del progetto.

## 6 DELIVERABLE DEL PROGETTO

### 6.1 Analisi dello Stato dell'Arte

- La prima fase del progetto consisterà nell'analisi dei sistemi di gestione dell'inventario esistenti, con un focus specifico su quelli distribuiti e basati su microservizi. Saranno valutati diversi approcci tecnologici, architetturali e matematici per l'ottimizzazione delle scorte.
- L'analisi sarà supervisionata e guidata dall'azienda proponente, che offrirà supporto per garantire un rapido completamento di questa attività.

### 6.2 Proof of Concept (PoC)

- Un primo prototipo che dimostri la fattibilità tecnologica del sistema. Questo prototipo si concentrerà su una piccola porzione delle funzionalità previste, come la gestione di un singolo magazzino e la sincronizzazione dei dati con un nodo centrale.
- Il PoC dovrà dimostrare la capacità di sincronizzazione dei dati in tempo reale, con una prima implementazione delle tecnologie e dell'architettura scelta.



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	11/13

## 6.3 Minimum Viable Product (MVP)

- Il deliverable finale sarà un MVP completo che simulerà una rete di magazzini distribuiti. Il sistema dovrà gestire l'inventario in modo ottimizzato, garantendo l'operatività in scenari di elevata complessità logistica e fornendo un'interfaccia centrale per la gestione e il monitoraggio delle scorte.
- Il sistema dovrà essere in grado di affrontare operazioni simultanee di prelievo e aggiornamento inventario da parte di più magazzini e gestire i conflitti di sincronizzazione.

## 7 TEST E VALIDAZIONE

L'azienda M31 pone grande enfasi sulla qualità del codice e sulla validazione del sistema sviluppato attraverso test strutturati. Al termine del progetto, dovrà essere prodotto un **test book** (cfr. req. 16) che descriva in dettaglio i test automatici implementati, i test manuali, i criteri di validazione e le metodologie utilizzate per garantire il corretto funzionamento del sistema in tutte le sue parti.

- **Test di sincronizzazione:** Saranno inclusi test che validino l'efficacia della sincronizzazione in tempo reale dei dati tra i diversi magazzini, valutando la capacità del sistema di gestire operazioni simultanee e conflitti di aggiornamento.
- **Test predittivi:** Test specifici valuteranno l'accuratezza degli algoritmi di riassortimento predittivo, simulando variazioni nei dati di input e valutando la risposta del sistema alle previsioni di domanda.
- **Unit test e code coverage:** Un vincolo imprescindibile sarà la definizione di una **copertura del codice** tramite unit test, che dovrà essere concordata con M31. Sarà necessario garantire che una percentuale significativa del codice sviluppato venga coperta da test automatici, con particolare attenzione alle componenti critiche del sistema, come la gestione delle scorte e la sincronizzazione dei dati.
- **Test di carico e scalabilità:** Il sistema dovrà essere sottoposto a test di carico per verificare la sua capacità di mantenere prestazioni elevate anche in scenari di elevato traffico dati. Questi test valuteranno l'efficienza del sistema nella gestione di grandi volumi di dati e richieste simultanee da parte di più magazzini.
- **Test di sicurezza:** Saranno inclusi test che verifichino la sicurezza del sistema, soprattutto per quanto riguarda la gestione dei dati sensibili e la protezione delle comunicazioni tra i magazzini.

M31 ed il committente si accorderanno su vincoli necessari a considerare il testbook come valido, in particolare verrà definita la quantità minima di test con risultato PASS che sarà necessaria per M31 per accettare quanto sviluppato.



Codice/Titolo	GC-0003.02 - Sistema di Gestione di un Magazzino Distribuito - Capitolato di Progetto		
Stato del Rilascio	Released	Classificazione	Public
Data	25/09/2024	Pagina	12/13

## 8 DOCUMENTAZIONE

Per tutto il ciclo di vita del progetto sarà necessario produrre una documentazione dettagliata che comprenda:

- **Diagrammi architetturali:** Sarà richiesta la produzione di diagrammi a blocchi per descrivere l'architettura del sistema e i vari componenti coinvolti. Si consiglia di utilizzare tool quali C4 (<https://c4model.com/>) per la descrizione dell'architettura nei suoi vari livelli e si consigliano tool quali Mermaid (<https://mermaid.js.org/>) per grafici più specifici o di processo.
- **Documentazione tecnica:** La documentazione del codice e delle funzionalità sviluppate dovrà essere completa, chiara e facilmente consultabile.
- **Manuale dei test:** Il test book dovrà essere redatto in modo esaustivo e dovrà descrivere tutte le procedure di test adottate, i criteri di successo e i risultati ottenuti durante la fase di validazione.

Copyright M31 © 2023 - This material is protected by copyright controlled by M31 S.r.l..

All rights are reserved. Unless otherwise stated, all product names, logos, and brands are property of their respective owners. All company, product and service names used in this material are for identification purposes only. Use of these names, logos, and brands does not imply endorsement. Copying, including reproducing, storing, adapting or translating, any or all of this material is strictly forbidden and requires the prior written consent of M31 S.r.l.. This material may also contain confidential information, which may not be disclosed to others without the prior written consent of M31 S.r.l..

M31 S.r.l.  
via Tommaseo, 77  
35131 Padova (PD) - Italy  
Tel: +39 049 78 11 811  
Fax: +39 049 78 11 899



[info@m31.com](mailto:info@m31.com) - [www.m31.com](http://www.m31.com)