

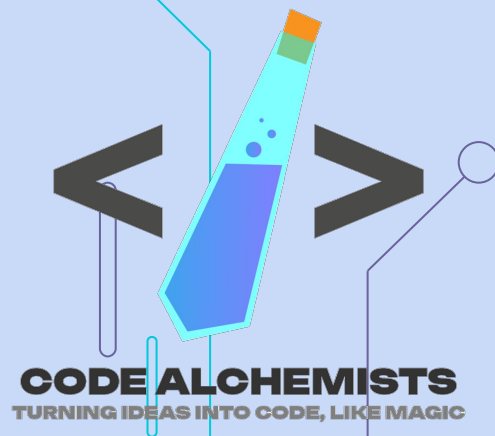


# **TECNOLOGIE**

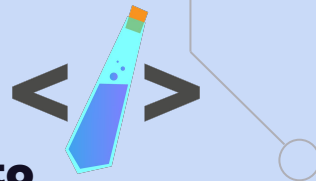
21 luglio 2025



team.codealchemists@gmail.com



# ► PRESENTAZIONE CAPITOLATO



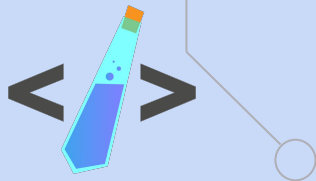
- **Capitolato C6: Sistema di Gestione di un Magazzino Distribuito**

- Architettura a microservizi: ogni magazzino locale deve essere indipendente, e i microservizi al suo interno sono indipendenti tra di loro ma al tempo stesso comunicano con tutto il sistema.

- **Obiettivi:**

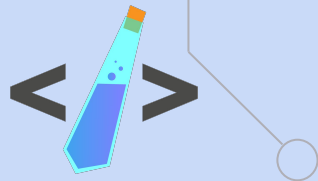
- Interoperabilità tra magazzini: ordini da e verso l'esterno, ordini interni tra magazzini, sistema di notifiche in caso di guasti/errori o in caso di merce sotto soglia.
- Sincronizzazione con il sistema centrale, che deve avere sempre una visione complessiva aggiornata per poter operare correttamente.
- Gestione utenti: Supervisore Locale che gestisce uno o più magazzini locali, Supervisore Globale che controlla il sistema centrale.

# ► TECNOLOGIE UTILIZZATE



- **TypeScript** con **NestJS** basato su **NodeJS**
  - Alternative considerate: **Express js, Fastify, Django/Flask, Go**
  - Motivo della scelta Typescript: è **tipizzato staticamente**, offre **ottimo supporto da IDE**, e ha **ampio ecosistema JavaScript**, garantendo **codice più sicuro e facilmente manutenibile** oltre che a tempi di sviluppo rapidi.
  - Motivo della scelta NestJS: fornisce una **struttura solida e modulare per sviluppare microservizi in TypeScript**, con **supporto nativo a concetti enterprise come l'iniezione delle dipendenze**, semplificando l'organizzazione del codice e rendendo il progetto **più scalabile, testabile e manutenibile**.

# ► TECNOLOGIE UTILIZZATE



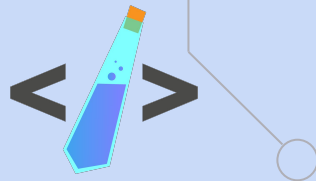
## ● Nats

- Alternative considerate: **Apache Kafka**
- Motivo della scelta: è **più leggero e semplice da configurare rispetto a Kafka**, offre **latenza estremamente bassa** e garantisce **comunicazione asincrona efficiente** tra microservizi, caratteristiche ideali per un'architettura distribuita e reattiva come quella dei nostri magazzini.

## ● MongoDB

- Alternative considerate: **PostgreSQL, Redis**
- Motivo della scelta: offre **flessibilità nella gestione dei dati non strutturati, scalabilità orizzontale e replicazione automatica**, risultando ideale per un sistema distribuito come quello dei nostri magazzini, dove i dati possono variare da un nodo all'altro e devono essere sincronizzati in tempo reale.

# ► TECNOLOGIE UTILIZZATE



- **Docker**

- Motivo della scelta: permette di **eseguire ogni componente del sistema in un ambiente isolato e replicabile**, facilitando la gestione dei microservizi, garantendo **portabilità tra ambienti** e semplificando il **deployment distribuito del sistema di magazzini**.

- **Docker-compose**

- Alternative: **Kubernetes**
- Motivo della scelta: si integra molto bene con Docker, è molto comodo e facile da utilizzare per la **fase di sviluppo e testing** .



## DOCKER COMPOSE

