

# Project work 2024

## Contesto di riferimento

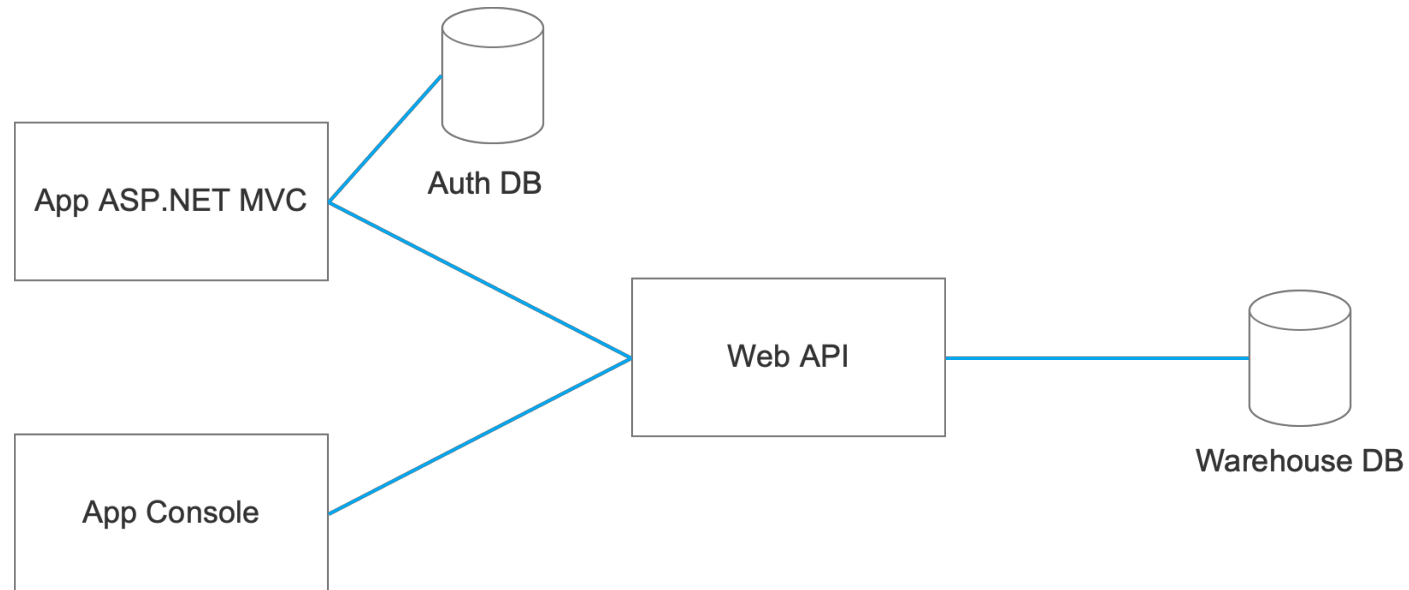
### Applicazione distribuita per la gestione del magazzino

Una azienda di produzione vuole realizzare due applicazioni per la gestione manuale dei movimenti di magazzino e per la visualizzazione delle giacenze dei singoli prodotti:

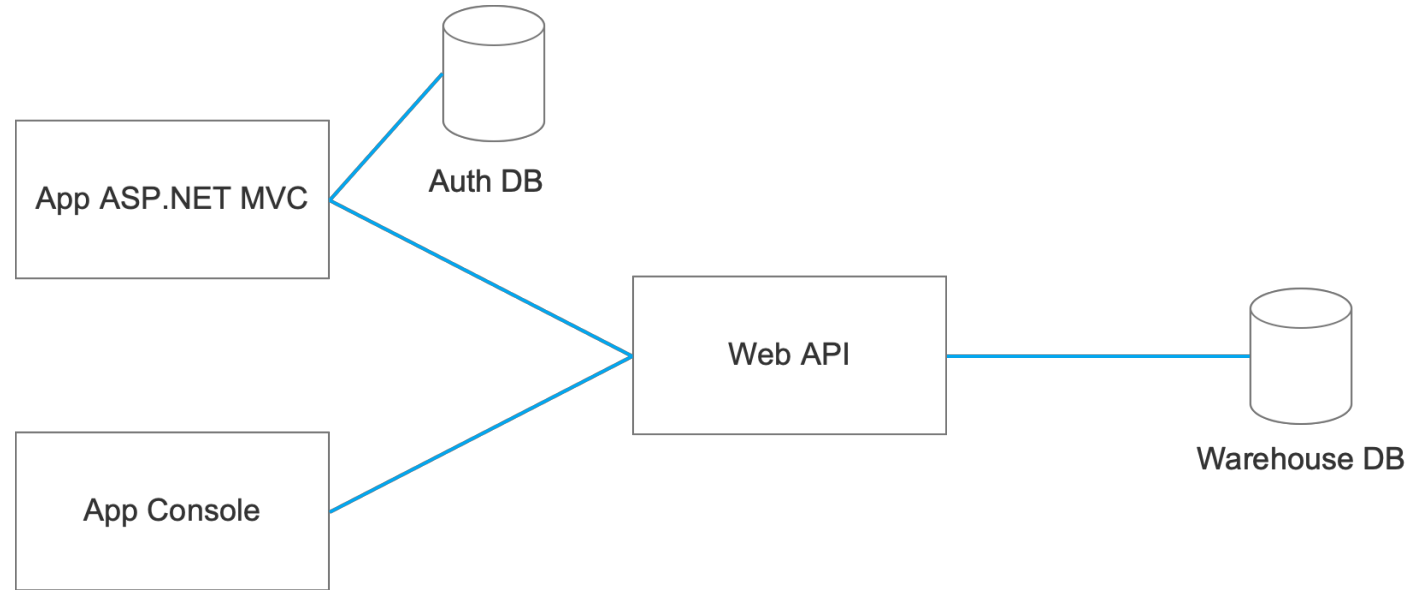
1. Applicazione web per la gestione del magazzino da pc (funzionalità complete)
2. Applicazione console, installata su bar code reader e utilizzata dagli operatori di magazzino (funzionalità ridotte)

# Architettura

Per un corretto funzionamento delle due applicazioni, e per evitare duplicazioni di dati e codice, si è scelto di utilizzare la seguente architettura:



# Architettura



- I dati del magazzino sono memorizzati nel Warehouse DB
- Solamente la Web API accede al Warehouse DB
- L'applicazione Web e l'applicazione Console accedono ai dati utilizzando la Web API
- La Web API e l'applicazione Console non richiedono autenticazione (non sono esposte sulla internet pubblica)
- L'applicazione Web richiede invece l'autenticazione e mantiene i dati degli utenti in un proprio database

# Warehouse DB

Il Warehouse DB mantiene le seguenti informazioni:

- Informazioni sui prodotti (codice, descrizione, immagini...)
- Informazioni sui movimenti di magazzino (data del movimento, prodotto, quantità entrata...)

Il database non contiene informazioni sulle giacenze. Possono però essere ricavate tramite sommatoria di tutti i movimenti.

# Auth DB

L'Auth DB mantiene le informazioni minimali sugli utenti autorizzati ad accedere al sistema:

- Username
- Password (possibilmente cifrata con SHA2-512)

# Applicazione Console - Requisiti

L'applicazione console deve implementare una sola funzionalità:

- Dato un codice di prodotto (letto da tastiera) stampare a video la giacenza di tale articolo e l'elenco degli ultimi 5 movimenti (in ordine cronologico inverso, stampando per primo il più recente)

# Applicazione ASP.NET MVC - Requisiti

L'applicazione ASP.NET MVC deve implementare le seguenti funzionalità:

- Login utente.
- Menù di navigazione con tre voci: Prodotti, Giacenze, Movimenti. Ogni voce rimanda ad una vista che permette di gestire i relativi dati
- Gestione prodotti
  - Vista paginata con l'elenco di ogni prodotto, riportando solo le informazioni principali
    - La vista deve avere un campo di ricerca per filtrare sia per codice articolo che per descrizione
  - Prevedere la possibilità di visualizzare la vista di dettaglio del singolo prodotto, riportando tutte le informazioni, anche le foto
  - I prodotti sono in sola lettura. Non deve pertanto essere possibile modificarli.



# Applicazione ASP.NET MVC - Requisiti

- Gestione giacenze
  - Unica vista paginata con l'elenco di tutti i prodotti, corredato dalla giacenza
  - La vista deve avere un campo di ricerca per filtrare sia per codice articolo che per descrizione
  - I prodotti con giacenza negativa devono essere evidenziati in rosso
- Gestione movimenti
  - Vista paginata con l'elenco di tutti i movimenti ordinati cronologicamente (dal più recente)
    - La vista deve avere un campo di ricerca per filtrare sia per codice articolo che per descrizione
  - Prevedere la possibilità di inserire, cancellare, modificare i movimenti
  - Non è necessaria la vista di dettaglio del singolo movimento

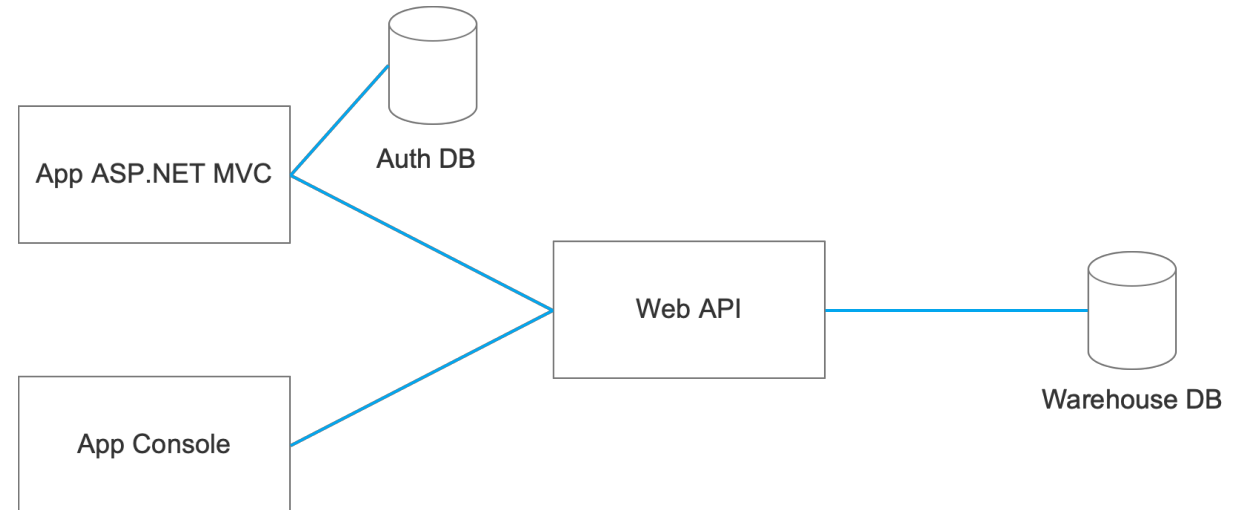
## Applicazione Web API - Requisiti

- L'applicazione Web API deve implementare tutti gli endpoint necessari a supportare le esigenze delle altre due applicazioni.
- Non devono essere implementati endpoint non necessari. Per esempio non devono essere implementate le operazioni di update, delete, insert sui prodotti
- L'applicazione deve essere conforme al pattern REST e deve scambiare dati in formato JSON

# Tecnologie e pattern

Le applicazioni utilizzano le seguenti tecnologie e pattern:

- .NET Framework
- ASP.NET API
- ASP.NET MVC
- Applicazioni Console
- EF
- MVC, ViewModels, Dto
- REST
- Database Sql Server
- Utilizzo dei metodi asincroni



# Obiettivi

- Rispetto delle specifiche tecniche
- Rispetto delle specifiche funzionali
- Processo di progettazione e sviluppo
  - Come affronto il progetto (quali passi)?
  - Come affronto i punti non trattati nelle specifiche?
  - Come progetto le UI?
  - Come strutturo il codice?
- Documentazione dell'intero processo
- Comunicazione dell'intero processo
  - Riesco a spiegare i motivi di tutte le decisioni prese?
- Codice funzionante (meglio incompleto ma funzionante che completo e non funzionante)

# Cosa realizzare?

Devono essere progettati e realizzati i seguenti componenti:

- Database Auth
- Applicazione API
- Applicazione Web ASP.NET MVC
- Applicazione console
- Slide di presentazione del lavoro

NB: Il database Warehouse è uguale per tutti e vi verrà fornito già popolato di dati

# Warehouse DB

Il database è composto da due tabelle con il seguente schema:

```
CREATE TABLE Products(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Title VARCHAR(100) NOT NULL,  
    Description VARCHAR(1024) NOT NULL,  
    Price FLOAT NOT NULL,  
    Images VARCHAR(1024)  
);
```

```
CREATE TABLE WarehouseMovements(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Date DateTime NOT NULL,  
    ProductId INT NOT NULL,  
    Qty FLOAT NOT NULL  
);
```

Le immagini sono ospitate su un cloud storage e sono raggiungibili al seguente indirizzo base: <https://storage.googleapis.com/e802024pw>

## Aggiunta (opzionale)

- Aggiungere una ulteriore pagina, chiamata "Statistiche" all'applicazione ASP.NET MVC, modificando l'API per recuperare i dati necessari.
- La pagina deve visualizzare, in forma di grafico, il totale dei movimenti mensili e la giacenza alla fine del mese, come nell'esempio che segue (dati per ProductId=1)
- L'articolo deve essere selezionabile tramite una combo box inserita nella parte alta della pagina
- Utilizzare la libreria chart.js (<https://www.chartjs.org>)

