APP

Sommario

[Da definire 2](#_Toc511976587)

[Teconolgie 2](#_Toc511976588)

[Front 2](#_Toc511976589)

[Back 2](#_Toc511976590)

[Database 2](#_Toc511976591)

[Versioning tool 2](#_Toc511976592)

[Macchina esposta per web api / db 3](#_Toc511976593)

[IDE 3](#_Toc511976594)

[Varie app 3](#_Toc511976595)

# Da definire

## Teconolgie

### Front

#### Angular?

* + PROS: più conosciuta, utilizzata in azienda quindi eventuali richieste di aiuto meglio indirizzabili; ionic
  + CONS: non direttamente nativa;

#### React?

* + PROS: più indirizzata per app native;
  + CONS: completamente sconosciuta;

### Back

#### Asp.net core web api?

* + PROS: multiplatform, nuova, a tendere più supportata?
  + CONS: non tutte le librerie di .net ancora supportate

#### Asp.net WebApi classiche?

* + PROS: conosciute
  + CONS: no multiplatform, serve a tutti i costi windows

### Database

#### MySql?

* + PROS: conosciuto, multiplatform
  + CONS: limitato rispetto a sql server in termini di dimensioni db

#### MariaDb?

* + PROS: boh
  + CONS: boh

#### SqlServer?

* + PROS: conosciuto
  + CONS: no multiplatform

#### Altri db?

Criteri per la scelta sono: famigliarità con database, semplicità di utilizzo, “richiesta” di risorse hardware, client user friendly per interrogazioni a db

### Versioning tool

#### Git:

* + PROS: buttare roba su web
  + CONS: sconosciuto, 7€ mese per avere versione privata, se no tutto pubblico / visibile, da impostare server se si vuole mantenere tutto privato

#### Subversion:

* + PROS: conosciuto ed utilizzato
  + CONS: necessario server?

### Macchina esposta per web api / db

#### Raspberry:

* + PROS: poco dispendio di energia
  + CONS: risorse limitate

#### Pc con windows server 2016:

* + PROS: risorse più elevate
  + CONS: dispendio di energia, ingombro, rumore (più costoso in generale)

### IDE

#### FRONT

* Visual studio code

#### BACK

* Visual Studio 2017
  + PROS: più dedicato al c#
  + CONS: utilizzando vs code per il front, avere un ide solo su cui lavorare sarebbe meglio!
* Visual studio code
  + PRONS: ide condiviso per back e front
  + CONS: al momento poco “famigliare” per lavorare in c#

## Varie app

#### Nome app (elencare proposte)

#### Funzionalità app (elencare proposte)

Login con gestione utente

* + info di base
  + registrazione
  + cancellazione
  + impostazione pwd

Gestione spese carburante auto

* + Anagrafica auto
  + Anagrafica carburanti
  + Anagrafica distributori
  + Gestione auto di default per utente
  + Gestione carburante di default per auto
  + Inserimento nuovo rifornimento, sulla base del modello cartaceo utilizzato. Dati da inserire:
    - Km auto nel momento del rifornimento
    - Data rifornimento
    - Distributore in cui si sta facendo rifornimento (da anagrafica)
    - Q.tà carburante
    - UM carburante (anagrafica unità di misura?)
    - Prezzo unitario €/UM (prevedere altra valuta?)
    - Spesa totale
    - Flag per ricevuta Y/N
    - Utente del rifornimento (chi è loggato)
  + Gestione pagina di statistiche rifornimenti:
    - Consumo medio / periodo (da data a data)
    - Percorrenza km / periodo (da data a data)
    - Consumo carburante / periodo (da data a data)