

# Capitolo 1

## Progettazione

La prima fase di questo progetto ha riguardato lo studio dei requisiti al fine di scegliere le migliori tecnologie da utilizzare. Nello specifico:

**L'app mobile deve prevedere:**

- Il riconoscimento dell'attività dell'utente
- Il riconoscimento della variazione dell'attività di un utente (WALKING-> DRIVING e viceversa)
- La possibilità di visualizzare una mappa con le varie aree poligonali di Bologna
- La possibilità di chiedere al backend la disponibilità di parcheggi all'interno di una certa area poligonale
- Il riconoscimento della posizione dell'utente
- L'invio al server degli eventi di entrata e uscita di un parcheggio

**Il server deve prevedere:**

- La possibilità di ricevere e rispondere alle richieste di disponibilità di parcheggi da parte dell'app
- La possibilità di ricevere le richieste di entrata e uscita di parcheggi da parte dell'app
- La possibilità di inviare le aree poligonali di Bologna all'app
- In caso di zone prive di dati (ad esempio per mancanza di utilizzo dell'app da parte di utenti in quella zona) il backend deve prevedere un meccanismo di interpolazione dei dati, per stimare il numero di parcheggi disponibili

### Il frontend deve prevedere:

- Visualizzare, per ogni zona della città, il numero totale di richieste di parcheggio provenienti da quella zona.
- Utilizzare il clustering K-Means per visualizzare gli eventi di parcheggio.
- Representare gli eventi di parcheggio mediante una heatmap.

Dopo aver studiato i requisiti siamo passati alla scelta dei linguaggi e delle librerie da utilizzare.

Per offrire un'app mobile che possa essere utilizzata sia su sistema operativo Android che iOS abbiamo deciso di utilizzare il framework *Flutter*, un framework open-source sviluppato di Google che si poggia su *Dart*. Il progetto è stato però testato solo su android poichè nessun componente del gruppo possedeva un dispositivo con MacOS. Per la gestione della mappa abbiamo utilizzato il pacchetto *flutter\_map* che permette di integrare comodamente delle mappe Leaflet all'interno delle propria app, mentre per quanto riguarda la libreria per fare Activity Recognition abbiamo utilizzato un pacchetto chiamato *flutter\_Activity\_Recognition*, un wrapper alle librerie di sistema Android e iOS per fare riconoscimento dell'attività dell'utente.

Per il backend abbiamo deciso di utilizzare *Node.js*, un runtime Javascript che permette di creare applicazioni web veloci e scalabili, il Database invece, è stato creato utilizzando *PostgreSQL* con estensione *PostGIS* per la gestione dei dati spaziali.

Per motivi di riusabilità del codice e dei componenti abbiamo deciso di utilizzare il framework React per il frontend, un framework *Javascript* open-source sviluppato da Facebook che permette di creare componenti riutilizzabili. Per la gestione delle mappe sia nel frontend abbiamo utilizzato la libreria *Leaflet*, un framework Javascript open-source che permette di creare mappe interattive, dal quale abbiamo preso anche due pacchetti per fare Clustering e per visualizzare la Heatmap dei dati. Infine per la parte grafica è stato utilizzato il framework *Material-UI*, un framework Javascript open-source che permette di creare componenti grafici in stile *Material Design*.