POLITECNICO DI TORINO

Collegio di Ingegneria Gestionale

Corso di Laurea Magistrale
in Ingegneria Gestionale

Tesi di Laurea Magistrale

Progettazione e sviluppo di una applicazione mobile per il monitoraggio della qualità dell'aria nel contesto urbano. Caso di studio: la città di Milano.

Relatori
prof. Silvia Chiusano
prof. Tania Cerquitelli

Candidato
Nunzio Caggiano

Aprile 2018

Sommario

1. Introduzione

Uno dei temi più importanti a livello nazionale e globale risulta essere quello dell'inquinamento atmosferico che sta diventando sempre più un problema per le varie città, e che quindi le amministrazioni locali cercano di contrastare in vari modi(blocchi sul traffico, maggior controllo sulle emissioni delle fabbriche ecc.). Ovviamente questo è un problema che riguarda tutte le città, ma quelle che preoccupano maggiormente risultano essere le città con elevata densità demografica e un alto numero di fabbriche. Ad esempio Milano è una di queste e proprio su di essa viene sviluppata questa tesi.

Questo tema riscuote enorme interesse anche per il singolo cittadino, in quanto l'inquinamento atmosferico può influire in maniera positiva o negativa sulla qualità della vita. Proprio per questo, per migliorare il benessere dei cittadini è importante monitorare costantemente l'inquinamento atmosferico e prendere le dovute decisioni e precauzioni.

Per monitorare in maniera adeguata tale fenomeno è necessario conoscere dettagliatamente l'argomento, quindi le sue cause e gli inquinanti da prendere in considerazione. L'operazione di monitoraggio può avvenire solamente dopo aver raccolto i dati giusti e averli poi rappresentati mediante indicatori di facile lettura.

Questa tesi si pone come obiettivo quello di sviluppare degli strumenti che possano essere consultabili da tutti, relativi al fenomeno dell'inquinamento della città di Milano.

Per raggiungere tale obiettivo, è stato necessario progettare il Data Warehouse (DW). Progettare il DW è un processo che richiede varie fasi.

Per prima cosa sono state individuate le fonti dati su cui lavorare. Tali dati sono tutti open e disponibili online. Per far sì che il data warehouse progettato sia completo, oltre ai dati sugli inquinanti, sono stati raccolti anche i dati su meteo e traffico, in quanto fenomeni strettamente correlati al problema dell'inquinamento atmosferico. Nel dettaglio, i dati sulle concentrazioni di inquinanti misurate dalle 8 stazioni presenti a Milano sono state recuperate dal portale di ARPA Lombardia, quelli sulle condizioni metereologiche son state fornite dal portale Weather Underground, e infine quelli riguardanti il traffico nella città sono stati reperiti dal portale del comune di Milano.

Il secondo passo effettuato per la progettazione del DW è stata quella di capire come e cosa voler mostrare all'utente finale, quindi si è deciso quella che deve essere la granularità con cui mostrare le varie informazioni e i KPI (Key Performance Indicator) fondamentali, che permettono di monitorare i valori di inquinamento atmosferico. Quindi, dopo aver progettato il data warehouse, sono stati sviluppati due strumenti di visualizzazione pensati per due tipologie di utenti diversi.

In particolare sono state sviluppate: un'applicazione Android e una dashboard (usando php/mysql).

L'applicazione Android è stata realizzata con l'obiettivo di andare incontro a quelle che possono essere le esigenze del cittadino riguardo questo argomento. I cittadini sono interessati a valutare localmente la qualità dell'aria in alcune specifiche aree urbane, ad esempio dove vivono o lavorano. Questi utenti possono essere interessati alla visualizzazione di alcuni indicatori con diverse granularità spaziali e temporali, ma possibilmente forniti in modo facilmente comprensibile e intuitivo.

La dashboard invece è stata sviluppata pensando che debba essere rivolta principalmente al personale dell'amministrazione locale. Il personale dell'amministrazione locale è interessato a comprendere le principali cause di inquinamento che interessano la città cercando poi di garantire un buon livello di qualità dell'aria. Questi cercano di capire in quali punti della città

l'inquinamento risulta aver raggiunto condizioni critiche. Sono anche interessati a valutare gli aspetti contestuali che possono contribuire all'inquinamento atmosferico come le condizioni climatiche e il traffico di veicoli nelle aree urbane. Il personale dell'amministrazione locale quindi deve analizzare i flussi completi dei dati raccolti (anche a diverse granularità spaziali e temporali), per monitorare e comprendere il fenomeno osservato, valutare le diverse componenti e identificare le possibili cause.

La tesi è strutturata nel seguente modo: il capitolo 2 riguarda gli Aspetti teorici della progettazione di un data warehouse. Questo capitolo è suddiviso in vari paragrafi contenenti aspetti teorici sulla basi dati, e appunto il data warehouse, Di esso vengono descritti i vari tipi di architetture, le fasi di progettazione necessarie per la costruzione di un data warehouse, viene descritto il dimensional fact model e vengono introdotti a livello teorico i KPI. Viene poi descritta la disciplina della Business Intelligence.

Nel capitolo 3 viene invece contestualizzato a livello teorico il problema dell'inquinamento atmosferico e vengono descritti gli effetti dei vari inquinanti sia dal punto di vista ambientale, sia sulla salute dell'uomo. Questo capitolo quindi fa una panoramica a livello teorico di quello che è l'ambito di sviluppo della tesi.

Nel capitolo 4 viene descritto lo stato dell'arte. In questo capitolo viene fatta un'analisi su quelle che sono le app e le dashboard esistenti riguardo tale fenomeno. Inoltre, vengono anche descritte le varie tecnologie prese in considerazione per lo sviluppo dell'applicazione e della dashboard e ne viene fatto un confronto che spiega i motivi per cui è stata adottata una certa tecnologia piuttosto che un'altra.

Nel capitolo 5 si descrive quello che è il framework dell'intero progetto. Per cui questo capitolo contiene le fonti dati usate, i dimensional fact model progettati che vanno a definire quella che è la progettazione concettuale, la spiegazione di come le fonti dati sono state pulite e modificate (in base alle necessità individuate nella progettazione concettuale) per poterle poi inserire nel database (rappresenta la progettazione logica), la descrizione dettagliata di tutte le tabelle presenti nel database e la spiegazione della vista creata che ha permesso di rendere più immediato e performante l'accesso e l'utilizzo del data warehouse. Inoltre vengono anche indicati i vari KPI di interesse nella tesi. Uno dei paragrafi di questo capitolo contiene poi la spiegazione nel dettaglio di come è stata sviluppata sia l'applicazione, sia la sezione della dashboard creata.[I]

Il capitolo 6 è il capitolo in cui vengono effettuate delle osservazioni sul fenomeno dell'inquinamento in base ai grafici creati nella dashboard.

Il capitolo 7 è quello in cui sono inserite le conclusioni e i possibili sviluppi futuri.

Il capitolo 8 contiene la bibliografia e sitografia consultata nello sviluppo della tesi.

[I] La progettazione e realizzazione del Data Warehouse è in comune con la tesi magistrale "Progettazione e sviluppo di una dashboard per l'analisi della qualità dell'aria nel contesto urbano. Caso di studio: la città di Milano" di Francesco Cagnazzo.

2. Aspetti teorici della progettazione e analisi di basi di dati

In questo capitolo vengono presentati gli aspetti teorici relativi a sistemi e strumenti utilizzati nello sviluppo del progetto, in particolare quindi le basi di dati, il Data Warehouse e gli strumenti di Business Intelligence.

2.1 Basi di Dati

La base di dati rappresenta una collezione di dati riguardanti uno o più argomenti correlati tra loro, ed è strutturata in maniera tale da consentire ai dati di essere utilizzati per diverse applicazioni.

Il DBMS o Data Base Management System è un sistema software progettato per assistere, mantenere e utilizzare una grande collezioni di dati, assicurando condivisione, persistenza e affidabilità. Tale collezione di dati è gestita da un DBMS. Le caratteristiche generali di un DBMS sono le seguenti:

- possibilità di creare nuovi database e di specificare i rispettivi schemi mediante un linguaggio specializzato chiamato data definition language;
- permettere agli utenti di eseguire query ai dati e di modificarli, facendo uso di un apposito linguaggio detto query language o data-manipulation language;
- consente di salvare una enorme quantità di dati per un lungo periodo di tempo, rendendoli sicuri;
- garantisce l'integrità evitando accessi simultanei allo stesso dato, controllando l'accesso da più utenti alla volta. [1]

2.2 Data Warehouse

Un data warehouse (DW) è un archivio informatico contenente i dati di un'organizzazione, necessari per eventuali analisi di dati con lo scopo dell'attuazione di processi decisionali e del miglioramento del patrimonio informativo.

Alla base del Data warehousing ci sono alcuni strumenti tra cui quelli per localizzare, estrarre, trasformare e caricare i dati. Altri componenti importanti in un sistema di data warehouse sono quelli di business intelligence e quelli per gestire e recuperare i metadati.

Nel 1996, fu data da William H. Inmon per la prima volta la definizione di data warehouse:

"*Un Data Warehouse è una collezione di dati di supporto per il processo decisionale che presenta le seguenti caratteristiche:*

- è integrata e consistente;

- è orientata ai soggetti di interesse;

- è rappresentativa dell'evoluzione temporale e non volatile."

Andando nel dettaglio della definizione di DW di Inmon, si analizzano di seguito le caratteristiche:

- Integrata e consistente: in un DW possono confluire dati provenienti da varie fonti eterogenee: dati estratti dall'ambiente di produzione e dati provenienti da sistemi informativi esterni all'azienda. Tramite l'utilizzo di metodi di codifica uniformi, il perseguimento di una omogeneità semantica di tutte le variabili, l'utilizzo delle stesse unità di misura, può essere raggiunta l'integrazione.

- Orientata al soggetto: il DW è incentrato sui concetti di interesse dell'azienda, quali clienti, prodotti, vendite e ordini. Si passa dalla progettazione per funzioni ad una modellazione dei dati che consenta una visione multidimensionale degli stessi.
- Variabile nel tempo: I dati contenuti in un DW sono aggiornati fino ad una certa data che di solito è antecedente a quella in cui l'utente interroga il sistema. Questa è una caratteristica che lo differenza da un sistema transazionale, il quale invece è incapace di fornire un quadro storico di ciò che viene analizzato e in cui i dati corrispondono sempre ad una situazione aggiornata.
- Non volatile: in confronto alla progettazione di un'applicazione transnazionale il DW ha una facilità di progettazione maggiore. Questa consente degli accessi in sola lettura per cui i dati contenuti nel DW non sono modificabili. Questo fa sì che non ci siano più più le possibili anomalie dovute agli aggiornamenti, né si ricorre a strumenti complessi per gestire l'integrità referenziale o per bloccare record a cui possono accedere altri utenti in fase di aggiornamento.

Il data warehouse viene usato come supporto ai decision maker andando a descrive quello che è il processo di acquisizione, trasformazione e distribuzione di informazioni presenti all'interno o all'esterno delle aziende. Questo lo differenzia dai normali sistemi gestionali i quali, al contrario, hanno il compito di automatizzare le operazioni di routine. Un data warehouse può essere costruito secondo diverse modalità che spaziano da una logica completamente accentrata ad una logica completamente distribuita, questo quando il focus viene posto sulla capacità di supportare il processo decisionale

Un concetto fondamentale del DW, da analizzare prima di entrare nel dettaglio delle possibili architetture, è quella di multidimensionalità. Dato che gli eventi che si susseguono in un'azienda sono molteplici per poterli raggruppare e selezionare si immagina di collocarli in uno spazio n-dimensionale i cui assi (dimensioni di analisi) definiscono le diverse prospettive per la loro identificazione. Dal concetto di dimensione nasce la diffusissima metafora del cubo per la rappresentazione dei dati multidimensionali: si immaginano gli eventi come celle di un cubo i cui spigoli rappresentano le dimensioni di analisi. Queste celle contengono valori per ciascuna misura, per cui un cubo multidimensionale è incentrato su un fatto di interesse per il processo decisionale e che indica un insieme di eventi, descritti in maniera quantitativa da misure numeriche. Gli assi di tale cubo rappresentano una possibile dimensione di analisi; ed ogni dimensione può avere più livelli di dettaglio individuati da attributi strutturati in gerarchie. [2]

2.2.1 Architettura del Data Warehouse

In un sistema di data warehousing le caratteristiche fondamentali sono le seguenti:

- Separazione: mantenere separate l'elaborazione analitica e quella transazionale.
- Scalabilità: con la crescita dei volumi di dati da gestire ed elaborare e del numero di utenti da soddisfare, l'architettura hardware e quella software devono poter essere facilmente ridimensionate.
- Estendibilità: deve essere in grado di supportare nuove applicazioni e tecnologie senza riprogettare integralmente il sistema.
- Sicurezza: i dati memorizzati hanno natura strategica per cui il controllo sugli accessi è essenziale.
- Amministrabilità: la complessità dell'attività di amministrazione non deve risultare eccessiva.

Esistono diverse architetture per la progettazione di un sistema data warehousing; tuttavia, quella più utilizzata risulta essere l'architettura a due livelli che evidenzia la separazione tra il livello delle sorgenti e quello del DW. In realtà, questo si estende su quattro livelli distinti che descrivono stadi successivi del flusso di dati:

1. *Livello delle sorgenti*. Le fonti di dati utilizzate dal DW sono eterogenee e possono essere estratte dall'ambiente di produzione (originariamente archiviati in database aziendali relazionali) oppure provenire da sistemi informativi non interni all'azienda.

2. *Livello dell'alimentazione*. Per eliminare le inconsistenze e completare eventuali parti mancanti, i dati memorizzati nelle sorgenti devono essere estratti e ripuliti. Si parla così di strumenti ETL (Extraction, Transformation and Loading) i quali permettono di integrare schemi eterogenei, nonché di estrarre, trasformare, pulire, validare, filtrare e caricare i dati dalle sorgenti nel DW.

3. *Livello del warehouse*. Tutte le informazioni raccolte dalla varie sorgenti vanno a caratterizzare il DW. Esso può essere consultato e usato come sorgente per costruire data mart ossia un sottoinsieme o un'aggregazione dei dati presenti nel DW primario.

4. *Livello di analisi*. Permette la consultazione efficiente e flessibile dei dati integrati a fini di stesura di report, di analisi, di simulazione.

Le principali motivazioni a sostegno dell'utilizzo di questa architettura sono:

- E' sempre disponibile informazione di buona qualità anche quando è precluso l'accesso alle sorgenti a livello del warehouse.
- L'interrogazione analitica effettuata sul DW non interferisce con la gestione delle transazioni a livello operazionale
- L'organizzazione logica del DW si basa sul modello descritto in precedenza ovvero quello multidimensionale
- Tra i sistemi OLTP (trattano dati correnti al massimo livello di dettaglio), e sistemi OLAP (operano su dati storici e di sintesi), c'è una importante discordanza temporale e di granularità.
- per ottimizzare le prestazioni per applicazioni di analisi e reportistica si possono impiegare tecniche specifiche.

Nella Figura 2.1 è riportata la struttura dell'architettura a due livelli appena descritta.

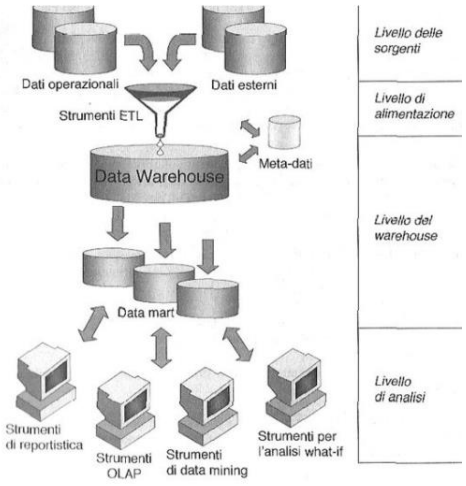


Figura 2.1: Architettura data warehouse a due livelli. [2]

Nei successivi paragrafi sono analizzati nel dettaglio alcune delle componenti presenti in questa architettura. [2]

2.2.1.1 Strumenti ETL

Componenti fondamentali per alimentare una sorgente dati singola, esauriente e di qualità in grado di alimentare il DW risultano essere gli strumenti ETL (Extraction, Transformation and Loading) in una architettura a due livelli.

Questa operazione si compone di quattro fasi:

1. *Estrazione*. Fase in cui i dati rilevanti vengono estratti dalle sorgenti. Se il DW viene popolato per la prima volta si parla di estrazione statica, se invece c'è un aggiornamento periodico del DW che cattura solamente i cambiamenti avvenuti nelle sorgenti dall'ultima estrazione, si parla di estrazione incrementale.

2. *Pulitura*. È una fase importantissima nel processo di data warehousing, in quanto si occupa di migliorare la qualità dei dati. I dati possono essere sporchi a causa di alcuni fattori: dati duplicati, dati mancanti, valori errati, valori inconsistenti.

3. *Trasformazione*. Fase in cui i dati vengono convertiti dal formato operazionale sorgente a quello del DW. La presenza di più fonti distinte eterogenee rende complicata la corrispondenza con il livello sorgente, e questo richiede durante la progettazione una complessa fase di integrazione.

4. *Caricamento*. E' l'ultima fase da eseguire e si tratta di caricare i dati nel DW. Questo può avvenire secondo due modalità:

- Refresh. I dati del DW vengono riscritti integralmente, sostituendo quelli precedenti.
- Update. I soli cambiamenti occorsi nei dati sorgente vengono aggiunti nel DW, tipicamente senza distruggere o alterare i dati esistenti.

2.2.1.2 Strumenti di Reportistica e OLAP

Nell'architettura a due livelli prima descritta, a livello di analisi vengono utilizzati differenti strumenti. In particolare, per quanto riguarda quelli di data mining e what-if si rimanda al paragrafo riguardante BI; invece di seguito sono descritti quelli di reportistica e OLAP.

Gli strumenti di reportistica sono principalmente orientati agli utenti con necessità di accedere a informazioni strutturate in modo invariabile e a intervalli di tempo stabiliti. Un rapporto (report) è invece caratterizzato da un'interrogazione in grado di generare la selezione e l'aggregazione dei dati multidimensionali e da una presentazione che può essere in forma tabellare o grafica.

Tuttavia, la principale modalità di fruizione delle informazioni contenute in un DW è rappresentata dagli strumenti OLAP: essi consentono agli utenti con difficoltà nell'effettuare le analisi da fare a priori, di esplorare in maniera interattiva i dati sulla base del modello multidimensionale. Quindi, gli utenti degli strumenti di reportistica svolgono un ruolo passivo, gli utenti OLAP sono in grado di costruire attivamente una sessione di analisi complessa in cui ogni passo effettuato è conseguenza dei risultati ottenuti al passo precedente.

Le funzioni di base di uno strumento OLAP sono:

• *Slicing*: permette di ridurre la dimensionalità del cubo andando a fissare un valore per una delle dimensioni;

• *Dicing*: mediante un determinato criterio di selezione viene ridotto l'insieme dei dati oggetto di analisi;

• *Roll-up*: tramite l'eliminazione di un livello di dettaglio da una gerarchia, c'è un aumento nell'aggregazione dei dati

• *Drill-down*: riduce l'aggregazione dei dati introducendo un altro livello di dettaglio in una gerarchia;

• *Drill-across*: permette un collegamento tra due o più cubi correlati al fine di compararne i dati;

• *Drill-through*: consiste nel passaggio dai dati aggregati multidimensionali del DW ai dati operazionali presenti nelle sorgenti;

• *Pivoting*: ha l'obiettivo di analizzare le stesse informazioni sotto diversi punti di vista con un cambiamento nella modalità di presentazione.

2.2.2 Fasi di progettazione di un Data Warehouse

2.2.2.1 Analisi e riconciliazione delle fonti dati

I dati presenti in un DW sono ricavati da varie sorgenti diverse sia per la tecnologia che le gestisce e sia per il modello con cui viene rappresentata la realtà aziendale. Per cui, durante la prima fase di progettazione è necessario andare a definire lo schema dei dati operazionali a partire dal quale verrà alimentato il data mart. Per fare questo si analizzano gli schemi delle sorgenti disponibili, le correlazione tra le varie sorgenti e si valuta quella che è la qualità dei dati. Ad essere coinvolti in questa fase risultano essere il progettista e gli amministratori dei database operazionali capaci di dare un significato a schemi e tracciati record non facilmente comprensibili ad altri. Un aspetto molto importante del data warehousing è il concetto di dato integrato che permette di derivare informazioni consistenti e prive di errori, dalle quali poi è necessario un processo di riconciliazione che comporta integrazione, pulizia e trasformazione dei dati.

2.2.2.2 Analisi dei requisiti

Fase adibita alla raccolta, filtraggio e analisi dei requisiti degli utenti finali da parte del progettista, con l'obiettivo di capire quali sono le informazioni di interesse strategico. Da questa fase vengono poi individuati i fatti da modellare e alcune indicazioni sul carico di lavoro. La scelta dei fatti dipende dall'utente finale e per ognuno di essi è fondamentale definire l'intervallo di storicizzazione, ossia l'arco temporale che gli eventi memorizzati dovranno abbracciare. La definizione del carico di lavoro ha lo scopo di consentire al progettista di identificare dimensioni e misure per la progettazione concettuale.

Dopo aver determinato il carico di lavoro viene definita la granularità (dipende da velocità di risposta richiesta e livello di dettaglio delle interrogazioni) nella rappresentazione dei fatti, cosa molto importante per la riuscita dell'intero progetto in quanto va a determinare la flessibiltà di interrogazione del data mart.

Ci sono varie tecniche per l'analisi dei requisiti utente, tra le quali si distingue l'approccio basato sull'utilizzo del formalismo di Tropos. Quindi nel contesto specifico dei data warehouse, è necessario introdurre alcuni nuovi concetti:

- Fatti. Un fatto modella un insieme di eventi che si verificano quando un obiettivo viene raggiunto.
- Attributi. Sono dei campi la cui valorizzazione si accompagna alla registrazione di un fatto ad opera di un obiettivo.
- Dimensioni. Una dimensione è una proprietà di un fatto che ne descrive una possibile coordinata di analisi.
- Misure. Una misura è una proprietà numerica di un fatto che ne descrive un aspetto quantitativo.

2.2.2.3 Progettazione concettuale

E' la fase in cui viene disegnato uno schema concettuale per il data mart. Questo schema è il così detto Dimensional Fact Model (DFM), in cui per ogni fatto di interesse evidenziato dall'utente deve essere creato uno schema che permetta di descrivere in maniera grafica tutti i concetti del modello multidimensionale: fatti, misure, dimensione e gerarchie. Ci sono tre fasi per quanto riguarda la progettazione concettuale:

1. *Mappatura dei requisiti*. Lo scopo principale di questa fase è quello di creare una corrispondenza tra fatti, dimensioni e misure individuate precedentemente e le relazioni e attributi presenti nello schema operazionale.

2. *Costruzione dello schema di fatto.* Ciascuna dimensione e misura mappati con successo da un diagramma di ragionamento esteso allo schema operazionale vengono inclusi nello schema di fatto. Successivamente vengono aggiunti anche gli attributi, eliminando quelli inutilizzati.

3. *Raffinamento*. Avviene una affinazione dello schema di fatto per renderlo sempre più vicino ai fabbisogni dell'utente.

Nel paragrafo 2.2.3 verrà analizzato più nel dettaglio il DFM

2.2.2.4 Progettazione logica

E' la fase in cui vengono determinate le procedure necessarie per andare a determinare lo schema logico del data mart, partendo dallo schema concettuale precedentemente estratto.

Il principale obiettivo di questa fase è la massimizzazione della velocità di reperimento dati, autorizzando il ripetuto utilizzo di dati ridondanti e denormalizzati. Essa può essere riassunta in tre fasi:

- Traduzione degli schemi di fatto in schemi logici. Se schema logico a stella, la tabella dei fatti presenta tutti gli attributi e tutte le misure collegate al fatto in maniera diretta ed inoltre per le gerarchie vengono create tabelle delle dimensioni contenenti tutti gli attributi.
- Materializzazione delle viste. Processo con cui vengono selezionate un insieme di viste con lo scopo di esaltare gli obiettivi di un progetto.
- Frammentazione delle viste. La tabella viene suddivisa in più tabelle chiamate frammenti in maniera tale da aumentare le prestazioni del sistema

2.2.2.5 Progettazione fisica

La fase di progettazione fisica è una fase in cui vengono individuati gli indici che ottimizzino le prestazioni. In questa fase un ruolo importante lo svolgono il carico di lavoro e il volume dati e inoltre occorre anche riferirsi ad uno specifico DBMS.

2.2.2.6 Progettazione dell'alimentazione

E' una fase in cui si prendono le decisioni sul processo di alimentazione del livello riconciliato e del data mart.

Fondamentale è la scelta, a partire dalle sue sorgenti, dell'intervallo di aggiornamento periodico del data mart.

2.2.3 Dimensional Fact Model

Il Dimensional Fact Model (DFM) è una vera e propria specializzazione del modello multidimensionale discusso in precedenza per le applicazioni di data warehousing. E' di tipo grafico e altro non è che un modello concettuale creato per fare da supporto alla progettazione di data mart. Il DFM si pone i seguenti obiettivi:

- dare supporto efficace alla progettazione concettuale;

- creare un ambiente in cui le interrogazioni utente possano essere formulate in modo intuitivo;
- rendere possibile la comunicazione tra il progettista e l'utente finale con l'obiettivo di raffinare le specifiche dei requisiti;
- costituire una piattaforma stabile per la progettazione logica;
- fornire una documentazione di progetto espressiva e non ambigua.

Il DFM genera una rappresentazione concettuale costituita da un insieme di schemi di fatto. Gli elementi di base modellati da questi schemi sono:

- Fatto: permette di modellare un insieme di eventi che avvengono nell'impresa ed esprime il concetto di interesse per il processo decisionale. Il fatto deve evolvere nel tempo quindi è fondamentale che abbia aspetti dinamici.
- Misura: descrive un aspetto di interesse quantitativo per l'analisi ed è quindi una vera e propria proprietà numerica di un fatto.
- Dimensione: va a descrivere le coordinate di analisi e indica la proprietà con dominio finito di un fatto. Ogni fatto di solito ha più dimensioni che ne vanno a determinare la minima granularità di rappresentazione.
- Evento primario: è l'occorrenza di un fatto, realizzata da una ennupla formata da un valore per ogni dimensione. Ad ogni evento primario viene associato un valore per ciascuna misura.
- Attributo dimensionale: sono gli attributi e le dimensioni, a valori discreti che le descrivono.
- Gerarchia: albero in cui i nodi sono attributi dimensionali e gli archi vanno a modellare associazioni di tipo molti a uno tra coppie di attributi dimensionali. Essa comprende una dimensione indicata sulla radice dell'albero oltre agli attributi dimensionali che la descrivono.
- Evento secondario: considerando un blocco di attributi dimensionali, ogni ennupla di loro valori porta ad un evento secondario che aggrega tutti gli eventi primari ad essa associato. Ad ognuno di questi eventi è anche associato un valore per misura, che sintetizza tutti i valori della stessa misura negli eventi primari corrispondenti.

Vengono ora introdotti alcuni concetti di modellazione avanzata del DFM:

- Attributi descrittivi: sono attributi che non sono usati per l'aggregazione in quanto hanno spesso domini con valori continui. Questi attributi specificano una proprietà di un attributo dimensionale e di una gerarchia ed è da quest'ultimo determinato mediante una dipendenza funzionale.
- Attributi cross-dimensionali: attributi di tipo dimensionale o descrittivo, il cui valore è determinato dalla combinazione di due o più attributi dimensionali, che possono anche appartenere a gerarchie distinte.
- Convergenza: avviene quando all'interno di una gerarchia due attributi dimensionali risultano essere collegati da due o più percorsi alternativi di associazioni molti a uno.
- Archi multipli: sono una connessione tra due attributi dimensionali che modellano un'associazione molti a molti
- Archi opzionali: vengono usati nel caso in cui un'associazione rappresentata in uno schema di fatto non viene definita per un sottoinsieme di eventi. [2]

2.2.4 Key Processor Indicator

I Key Process Indicator (KPI), sono indicatori numerici che consentono di valutare diversi aspetti dei processi aziendali. Tramite il loro utilizzo i manager capiscono come i vari processi gestiti dal sistema informativo stanno procedendo.

I KPI devono possedere cinque caratteristiche (SMART) [3]:

• *Specific*: il KPI deve essere specifico per ogni processo, e chiaramente dipende dall'azienda in questione e dal modo di lavorare. Non è possibile definire a priori un indicatore numerico buono in assoluto, ma dipendono dal singolo processo, ad esempio le soglie di minimo e di massimo che non devono essere superate non possono essere definite.

• *Measurable*: il KPI deve essere facile da misurare e la sua misurazione deve essere fattibile sia a livello economico sia a livello pratico.

• *Achievable*: per ogni indicatore le soglie stabilite devono sempre essere realisticamente raggiungibili oltre che essere sotto il controllo di chi valuta l'indicatore.

• *Relevant*: un indicatore deve essere rilevante. Questo vuol dire che si deve essere in grado di interpretare l'indicatore quindi capire da esso cosa non funziona, fornendo indicazioni per intervenire, correggendo, sul processo. Per cui, in generale, risulta essere utile misurare gli indicatori solo se le informazioni che ricavo sono utili per le esigenze informative. E' importante definire quelli che sono gli stakeholders e gli utenti a cui interessa il KPI ed è necessario definirli in maniera tale che essi possano valutare il processo.

• *Timely*: il KPI deve essere disponibile nei tempi prestabiliti: infatti se un indicatore non risulta essere disponibile nel momento in cui serve risulta essere inutile, anche se è un indicatore fondamentale. Qualsiasi indicatore ha un intervallo di tempo in cui è utile e rilevante.

Per fornire una visione completa e multidimensionale è di grande importanza andare a scegliere indicatori che abbraccino diverse categorie e che possano bilanciarsi a vicenda. Ci sono 4 principali categorie di KPI:

1. *KPI Generali o di Volume*: sono la base per nuovi indicatori e consentono di comprendere il contesto in cui avviene il processo.

2. *KPI di Efficienza*: ci dà informazioni sulla bontà o meno del processo rispetto alle risorse utilizzate e alla quantità di output prodotto.

3. *KPI* di Qu*alità*: verificano la qualità del prodotto come conseguenza del processo andandosi a focalizzare sui prodotti del processo finito.

4. *KPI di Servizio*: cercano di capire la relazione che c'è tra una richiesta e la relativa risposta ottenuta in termini di prodotto, di tempestività etc. Per cui questo tipo di KPI sono legati agli input e agli output del processo.

2.3 Business Intelligence

Tra gli strumenti informatici di supporto alle attività decisionali, i sistemi di data warehousing, prima descritti, sono sicuramente i più diffusi. Tuttavia esistono insiemi più ampi ed eterogenei di soluzioni, i quali prendono il nome di sistemi di Business Intelligence (BI). La Business Intelligence è una disciplina che consente ai decisori aziendali, di capire attraverso soluzioni software, i fattori chiave del business e in seguito di prendere le migliori decisioni in quel momento.

Il DW fino ad oggi è stato lo strumento principale per la BI, grazie ai vantaggi e le potenzialità descritti nei paragrafi precedenti. Tuttavia, andare ad utilizzare i DW comporta alcuni limiti, sono qui di seguito elencati quelli principali:

- L'intervallo di aggiornamento dei dati difficilmente è minore della settimana/giorno
- Se utilizzato per formulare interrogazioni complesse che non fanno parte del modello multidimensionale, il DW è poco efficiente.
- Non permette di formulare scenari di previsione in quanto consente di registrare solo il passato

Con il passare degli anni le aziende sono cambiate e ciò ha reso necessaria l'adozione di nuove soluzioni che consentano di rispondere alle sempre più crescenti richieste informative, che si possono riassumere nei seguenti punti:

- Tecniche di analisi più potenti e non basate sul modello multidimensionale;
- Possibilità di effettuare analisi sui dati provenienti da sorgenti informative eterogenee e con aggiornamenti rapidi
- Possibilità di prevedere il futuro e non solo di analizzare il passato.

Tre possibili proposte per il soddisfacimento dei punti precedenti sono: l'analisi what-if, il data maning e il Business Performance Management. [2]

2.3.1 Analisi What-If

L'analisi what-if supera uno dei limiti degli strumenti di reportistica e OLAP ossia quello di andare a tenere traccia solo del passato non consentendo di analizzare scenari futuri. Il what if è l'insieme di tecniche di analisi previsionale che consentono di valutare il comportamento di un sistema reale assumendo un particolare insieme di condizioni iniziali.

Si possono classificare le tecniche di analisi what-if in base all'approccio utilizzato quando si elabora il modello:

- Tecniche induttive: in base al comportamento che il sistema ha avuto durante un certo intervallo temporale si cerca di ricavarne il modello.
- Tecniche deduttive: puntano ad identificare e caratterizzare i rapporti di tipo causa-effetto tra i vari componenti del sistema.

2.3.2 Data Mining

Il Data Mining è quel processo che permette di estrarre conoscenza da banche dati di dimensioni molto elevate mediante l'applicazione di algoritmi che individuano le associazioni nascoste tra le informazioni e le rendono visibili. L'utente non riesce ad individuare tutti i pattern significativi quando ci sono moli di dati molto alte. Per pattern si intende la rappresentazione sintetica e ricca di semantica di un insieme di dati. Il Data Mining ha come obiettivo quello di aiutare l'utente nella ricerca del pattern e per farlo raccoglie un insieme di tecniche e metodologie dalle aree dell'intelligenza artificiale e del pattern recognition. Questo fa sì che non occorre più effettuare manualmente l'analisi dei dati, ma basta indicare con una certa approssimazione cosa e dove si vuole ricercare e lasciare che uno strumento automatico si accolli il peso della computazione.

Le tipologie di pattern più usate ed estraibili dai dati con tecniche di data mining sono:

- Regole Associative: consentono di individuare i gruppi di affinità tra oggetti, quindi determina le regole di implicazione logica presente nelle basi di dati.
- Clustering: è un insieme di tecniche di analisi multivariata dei dati con lo scopo di selezionare e raggruppare elementi omogenei in un insieme di dati.
- Alberi Decisionali: rappresentano un particolare tipo di classificatore, vengono usati per la comprensione di un particolare fenomeno poiché permettono di classificare in ordine di importanza, le cause che portano al verificarsi di un evento;

- Serie Temporali: a partire da sequenze di dati complesse, individuano i pattern ricorrenti o atipici.

2.3.3 Business Performance Management

Il Business Performance Management (BPM) rappresenta tutte le attività che permettono di misurare le proprie prestazioni andando ad incoraggiare l'uso efficiente delle risorse umane e l'efficacia dei processi aziendali

3. Inquinamento atmosferico

Questo capitolo è una introduzione teorica di quello che è il fenomeno dell'inquinamento atmosferico, le sue fonti e i suoi principali inquinanti. Inquinamento atmosferico che è l'argomento di fondo su cui vengono applicati i vantaggi del Data Warehouse.

3.1 Contesto

L'aria è una miscela di gas costituita da azoto (circa 78%), ossigeno (circa 21%) e da altri gas (argon, anidride carbonica, metano, ecc.) presenti in concentrazioni molto inferiori.
A causa di fenomeni di origine naturale (ad esempio eruzioni vulcaniche) e di origine antropica (attività umane), una sostanza presente nell'aria può avere concentrazioni diverse rispetto a quelle naturali. Quando si presentano queste alterazioni si parla di **inquinamento atmosferico**.

Le sorgenti artificiali di inquinamento atmosferico sono: le emissioni degli inquinanti industriali, delle centrali termiche, degli impianti di riscaldamento e dei mezzi di trasporto.
Gli inquinanti possono essere divisi in primari e secondari. I primi si hanno quando le sostanze contenute nelle emissioni si ritrovano direttamente nell'ambiente, le seconde invece si hanno quando queste sostanze subiscono in atmosfera dei processi di trasformazione. La presenza di tali inquinanti nell'aria va ad influire in maniera negativa su di essa, colpendo di conseguenza la salute delle persone. Per questo, molte città si adoperano per monitorare la qualità dell'aria urbana con l'obiettivo di prendere le dovute precauzioni e nello stesso tempo informare i cittadini. [4]

3.1.1 Fonti di inquinamento

Le cause dell'inquinamento atmosferico, come detto in precedenza, possono essere distinte in: fonti naturali (vulcani (SO2), incendi (PM10), ghiaioni (amianto), processi biologici (allergeni)) e fonti antropiche (traffico veicolare, riscaldamento domestico, industrie e attività artigianali, agricoltura e altre attività).

In particolare, per quanto riguarda il traffico veicolare, le emissioni dipendono dal tipo di combustibile, dal tipo di veicolo e da quando è stato prodotto. I veicoli alimentati a combustibili fossili provocano principalmente la produzione di anidride carbonica (CO2), particolato (PM10 e inferiori), idrocarburi (HC), ossidi di azoto (NOx), mentre i veicoli a metano e GPL emettono NOx, particolato ultrafine e scarsi idrocarburi. Negli ultimi anni si cerca di diminuire le emissioni da parte dei veicoli con l'introduzione delle fasce euro 1,2,3 ecc..

Un'altra fonte importante di sostanze inquinanti nell'aria è il riscaldamento domestico in cui gli inquinanti emessi dipendono dal combustibile utilizzato, dalla tipologia di riscaldamento, dalla tipologia, dalla vetustà e dalla manutenzione dello stesso.

La terza fonte più importante di inquinamento atmosferico riguarda l'industria e l'artigianato. Sono tantissime le lavorazioni in campo industriale che determinano inquinanti molto diversi tra loro in base alla lavorazione eseguita, e vanno da solventi, nebbie acide a metalli polveri ecc.. Tuttavia nel settore industriale le emissioni sono fortemente regolamentate per cui le industrie utilizzano diversi sistemi di abbattimento degli inquinanti.

Tra i principali contribuenti alle emissioni di sostanze inquinanti ci sono le industrie e la produzione di energia. Tuttavia, queste emissioni si verificano principalmente a notevoli altezze. Le emissioni sono mescolate con grandi volumi d'aria e il loro contributo alle concentrazioni ambientali è spesso piccolo. Le emissioni da traffico invece avvengono a basse altezze, in strada, dove la dispersione e la diluizione è lenta. Quindi questo, potrebbe causare

un gravissimo inquinamento dell'aria, anche quando le emissioni sono di piccole dimensioni. [4]

3.2 Principali inquinanti

Le sostanze emesse da industrie, veicoli, centrali elettriche ecc vanno a contaminare l'aria che si respira. Queste sostanze sono un grosso problema poiché portano ad effetti dannosi sulla salute o sull'ambiente. Ovviamente in base alla quantità, alla durata e alla pericolosità dell'inquinante, gli effetti e l'impatto possono essere diversi. Ad esempio, sulla salute possono essere di piccola entità e reversibili, come può essere ad esempio una irritazione agli occhi, oppure debilitanti se non addirittura fatali, ad esempio il cancro.

Di seguito sono analizzati i principali inquinanti dell'aria. [5][6]

- *Particolato Atmosferico (PM10/PM2.5)*

A causa delle loro piccolissime dimensioni (<10 micron, <2,5 micron), il PM10 e il PM2.5 sono di grande interesse poiché entrano facilmente in profondità nei polmoni e quindi potenzialmente pongono rischi significativi per la salute.

Le particelle in sospensione nell'aria sono variabili sia nella forma fisica sia nella composizione chimica.

Le cause principale dell'emissione di PM10 e PM2.5 sono: le lavorazioni industriali, gli impianti di riscaldamento, l'usura dell'asfalto, degli pneumatici e dei freni e le emissioni di scarico degli autoveicoli, in particolare quelli con motore diesel.

Il PM10 viene trasportato a distanze molto elevate e costituisce uno dei principali mezzi di diffusione di composti tossici. Il PM10 ha anche una componente secondaria, formatasi direttamente in atmosfera a partire da altri inquinanti gassosi già presenti.

Dal punto di vista della salute, queste particelle possono arrivare in profondità nei polmoni provocando infiammazioni e peggioramenti nelle persone con malattie cardiache e polmonari e possono portare in superficie composti cancerogeni.

Inoltre le polveri sottili possono agire pesantemente su statue, monumenti ed edifici sporchi, aumentandone costi di pulizia e manutenzione.

Molto alta risulta essere la concentrazione di tale inquinante nelle zone maggiormente urbanizzate dove si verificano numerosi superamenti del limite giornaliero di 50 $\mu g/m^3$. Un grosso problema è che la componente secondaria di tale inquinante è un "fondo" su cui risulta difficile incidere in maniera significativa.

- *Ossidi di Azoto (NOx)*

Con il termine NOX si descrive una miscela di ossido nitrico (NO) e biossido di azoto (NO2). NO è prodotto in quantità molto maggiori di NO2, ma si ossida a NO2 nell'atmosfera. L'emissione dell'NOx avviene quando il carburante viene bruciato ad esempio nel settore dei trasporti, nei processi industriali e nella produzione di energia.

Nel caso di un elevato irraggiamento solare porta allo smog fotochimico il quale a sua volta crea altre sostanze inquinanti. Per questo è da considerarsi uno degli inquinanti atmosferici più pericolosi.

I gas di scarico delle auto risultano essere una delle principali fonti di produzione di NOx. La quantità delle emissioni dipende dalle caratteristiche del motore e dalla modalità del suo utilizzo, questo poiché se il motore lavora ad un elevato numero di giri la produzione di NOx aumenta.

Esso dipende anche dagli impianti di riscaldamento infatti durante la stagione invernale a causa degli impianti di riscaldamento, le concentrazioni di tale inquinante aumentano,

Dal punto di vista della salute, il biossido di azoto è in grado di irritare i polmoni e dare

minore resistenza alle infezioni respiratorie come l'influenza. Ad essere molto soggetti ad eventuali problemi causati da questo inquinante sono i bambini.

Inoltre, gli ossidi di azoto contribuiscono in maniera significativa alla formazione di piogge acide e all'aumento della concentrazione di nitrati nel suolo.

- *Ozono (O3)*

L'ozono troposferico (O3) è un inquinante che non viene emesso direttamente in atmosfera, ma è prodotto dalla reazione tra il biossido di azoto (NO2), idrocarburi e luce solare ed è quindi di tipo secondario. La luce solare fornisce l'energia per iniziare la formazione di ozono, per cui elevati livelli di ozono sono osservati durante il periodo estivo.

I livelli di ozono sono molto elevati nelle aree urbane (dove alti livelli di NO sono emessi dai veicoli), più che nelle zone rurali.

Dal punto di vista della salute, l'ozono è causa di irritazione alle vie respiratorie dei polmoni, aumentando i sintomi di chi soffre di asma e malattie polmonari.

Nell'ambiente invece, l'ozono provoca danni ad una serie di colture commerciali ed alla vegetazione naturale. Esso è un potente agente ossidante nel deterioramento della qualità della gomma, dei tessuti.

- *Biossido di Zolfo (SO2)*

L'SO2 è causata da combustibili fossili bruciati che contengono tracce di composti di zolfo ed è dovuto alla produzione di energia e a fonti di trasporto di piccole dimensioni.

Le emissioni di SO2 sono molto variabili a seconda della stagione, Infatti presenta valori massimi nella stagione invernale, laddove sono in funzione impianti di riscaldamento domestici.

L'esposizione all'SO2 può danneggiare la salute causando una diminuzione della funzione polmonare negli asmatici. L'inquinamento dovuto all'anidride solforosa è considerato più dannoso quando sono elevati le concentrazioni da inquinamento di particolato e da altri composti.

Dal punto di vista ambientale, il principale costituente delle piogge acide è proprio l'acido solforico generato da reazioni atmosferiche di SO2. Inoltre le particelle di solfato di ammonio sono quelle più abbondanti che si trovano in aria. Questo inquinante risulta anche essere una minaccia per determinati tipi di pietra utilizzati negli edifici.

- *Monossido di Carbonio (CO)*

L'incompleta combustione dei materiali contenenti carbonio (tra cui quelli usati nei mezzi di trasporto) producono il CO, che è un gas inodore, insapore e incolore. Questo inquinante ha concentrazioni più elevate laddove c'è un elevato traffico urbano intenso e rallentato, infatti i valori risultano più elevati con motore al minimo e in fase di decelerazione. Tuttavia, un aspetto positivo, è quello che tali concentrazioni raramente superano gli standard relativi alla salute, anche nei centri urbani più affollati.

Il CO va a ridurre la capacità di trasporto dell'ossigeno nel sangue, agendo mediante reazione con emoglobina.

Inoltre il monossido di carbonio ha anche un importante ruolo nella formazione di ozono ed è trasformato in anidride carbonica.

- *Benzene (C6H6)*

Il benzene (C6H6), emesso dall'uso del petrolio e da oli minerali, altro non è che un idrocarburo aromatico incolore, liquido e infiammabile.

Il benzene è una sostanza cancerogena per l'uomo. Con esposizione a concentrazioni elevate, si verificano danni al midollo osseo, mentre con una esposizione cronica professionale sono

frequenti i casi di leucemia (molto frequenti nei lavoratori dell'industria manifatturiera, dell'industria della gomma e dell'industria petrolifera).

4. Stato dell'arte

In questo capitolo viene fatta una panoramica su quello che è lo stato dell'arte relativo allo sviluppo di applicazioni Android e Dashboard riguardo l'argomento dell'inquinamento atmosferico e sulle tecnologie disponibili per sviluppare queste metodologie di visualizzazione. Inquinamento atmosferico che è uno dei problemi più importanti a livello globale e che interessa tutti i paesi del mondo.
Ad esserne interessati sono i comuni cittadini e anche le singole amministrazioni locali. I primi sono interessati a conoscere la qualità dell'aria che respirano in determinate zone della città, verosimilmente dove lavorano e dove vivono, prendendo le dovute precauzioni. L'interesse dell'amministrazione locale invece è legata alla volontà di capire quelle che sono le cause che portano alla crescita dell'inquinamento atmosferico e di conseguenza prendere le dovute decisioni per limitare questo problema.

Per rispondere a queste esigenze sono sempre più frequenti sistemi di monitoraggio dell'inquinamento atmosferico che raccolgano le informazioni sui vari inquinanti, ai quali vengono anche associati dati metereologici e dati sul traffico poiché strettamente correlati con quelli che sono i valori delle concentrazioni di inquinanti. Questi sistemi di monitoraggio permettono analisi più approfondite in merito al fenomeno dell'inquinamento.

Con l'avvento di Internet e degli smartphone si è avuto un aumento esponenziale di dashboard e applicazioni riguardanti tale tema, i quali in maniera grafica e numerica mostrano i risultati delle analisi dei dati raccolti.

Per cui, prima di realizzare l'applicazione riguardante l'inquinamento atmosferico nel comune di Milano, è stato utile scaricare alcune applicazioni esistenti per capire quali possono essere gli indicatori più importanti e come questi vengono solitamente rappresentati.

Oltre all'applicazione si è anche integrata una sezione relativa al calcolo dell'AQI su una dashboard esistente per cui è stato anche necessario consultare varie dashboard che trattassero il problema per capire come queste lo visualizzassero sui propri siti.

E' stata infine fatta una analisi sulle tecnologie a disposizione per lo sviluppo di applicazioni per smartphone e lo sviluppo Web [7] cercando di individuare la tecnologia più idonea per la realizzazione di esse.

4.1 App esistenti

- *EPA's AIRNow*

L'Agenzia per la protezione dell'ambiente statunitense fornisce gratuitamente un'app Android che consente agli utenti di ottenere informazioni sul valore di AQI della zona localizzata tramite GPS o tramite l'inserimento manuale del codice postale della zona, sia per l'ozono che per le polveri fini (PM2,5). Vengono anche mostrate mappe di qualità dell'aria corrente e previsioni a livello nazionale della giornata successiva a quella odierna.

- *Air Quality: Real-Time AQI*

Quest'app permette di monitorare in tempo reale l'Air Quality Index (AQI) per più di 60 paesi nel mondo. Le fonti di dati di qualità dell'aria varia a seconda delle città:

- I dati sul PM2.5 sono presi dall'ambasciata statunitense e sono utilizzati per Shanghai, Pechino, Chengdu e Guangzhou.
- I dati del Ministero dell'Ambiente cinese relativi al PM10 sono utilizzati per tutte le altre città della Cina.

- Per Hong Kong, Taiwan e Singapore vengono utilizzati i dati delle rispettive agenzie relative al PM2.5.

L'AQI generale, e quello degli inquinanti che misura (PM10, PM2.5, SO2, NO2, CO e O3) vengono aggiornati con frequenza oraria e per ognuno è possibile vedere tramite grafico a barre il loro valore di AQI nelle 24 ore. Inoltre quest'app contiene anche le informazioni sui fenomeni metereologici, anch'essi con granularità oraria e relativi a temperatura, pressione, velocità del vento e umidità. Inoltre permette anche di localizzarsi e fornire le informazioni della stazione più vicina alla posizione dell'utente.

- *Ambience Data*

Questa azienda fornisce gratuitamente un'applicazione mobile scaricabile in cui l'utente selezionando la città, ha accesso al valore più recente di AQI, temperatura, pressione e velocità del vento, dicendo anche qual è l'inquinante che provoca quel dato valore di AQI e indicando l'orario giornaliero in cui l'AQI ha avuto il valore più alto.

- *Air Quality in Europe*

E' un'app che mostra l'AQI (con scala europea) in 100 grandi città europee (Parigi, Londra, Berlino, Bruxelles, Amsterdam, Roma ecc.).

L'applicazione offre la possibilità di visualizzare l'AQI corrente in città, quello della giornata precedente e una previsione per quella successiva. Inoltre l'AQI giornaliero è aggiornato ogni ora. Il valore di AQI viene mostrato tramite la mappa dell'Europa, dove per ogni città è indicato oltre che l'AQI di fondo della città (qualità dell'aria esterna vissuta dal cittadino medio), anche l'AQI lungo la strada (i cittadini interessati sono coloro che vivono, lavorano e visitano queste strade, incluse le persone in auto o bus).

Tutte le informazioni riportate su mappa, possono essere consultate anche in forma tabellare, permettendo così un confronto più veloce e immediato tra le varie città europee.

- *AIR QUALITY | AIR VISUAL*

E' sicuramente una delle app più complete riguardo l'inquinamento atmosferico. Fornisce in tempo reale i dati di inquinamento atmosferico, meteorologiche e climatiche per oltre 10.000 città in tutto il mondo.

Una volta localizzati, l'app mostra l'AQI della stazione più vicina con i dati su precipitazioni, temperatura e velocità del vento, fornendo di esse la previsione nei 7 giorni successivi anche con dettagli orari. Per una visualizzazione veloce di quella che è la situazione della qualità dell'aria nel mondo si può accedere alla sezione della mappa in cui per ogni stazione viene mostrato un marker con definizione cromatica corrispondente alla pericolosità dell'aria e con il rispettivo valore di AQI. Cliccando su ogni marker si possono avere informazioni specifiche su quella determinata stazione di rilevamento.

Inoltre è anche presente una sezione in cui vi è la classifica giornaliera delle città a livello mondiali effettuata in base all'AQI, a partire da quella peggiore, oltre ad una sezione News in cui sono inserite tutte le notizie online riguardanti aggiornamenti sul fenomeno dell'inquinamento.

- *Plume Air Report – Live and Forecast smog reports*

Plume Air Report è un'app che offre i livelli di inquinamento in tempo reale nella tua zona e le previsioni su come la qualità dell'aria si evolverà ora per ora nelle prossime 24 ore, proprio come le previsioni del tempo. Inoltre offre consigli personalizzati sul momento migliore per svolgere un'attività all'aperto senza essere sovraesposto all'inquinamento. Fornisce sia misurazioni con l'indice AQI sia in microgrammi/metrocubo. Contiene uno storico su quello che è stato il livello di qualità dell'aria nei giorni precedenti, ed inoltre per ogni città fornisce

la media annuale di AQI sia a livello generale, che di singolo inquinante, indicandone anche il valore più alto e quello più basso nell'arco dell'anno.

- *Qualità dell'aria (FFZ srl)*

Tramite quest'app, si potrà monitorare la qualità dell'aria del luogo in cui ci si trova e verificare quali attività sono sconsigliate in caso di inquinamento. Si potranno inoltre verificare le condizioni meteo con molta accuratezza e ricercare tutte le località del mondo per visionare in tempo reale la qualità dell'aria. Molto interessante è lo scatta foto con i dati sovraimpressi e con la possibilità di condividerla con gli amici sui principali social network.

4.2 Dashboard esistenti

Oltre ad un'applicazione, è stata sviluppata una sezione relativa al calcolo dell'AQI su una dashboard esistente per cui si sono consultati vari siti che visualizzassero il tutto.

Molte delle app viste in precedenza, hanno anche un sito, che chiaramente mostra più informazioni rispetto a quelle mostrate dalle app. Tra queste c'è il sito dell' Agenzia per la protezione dell'ambiente statunitense che dà all'utente l'opportunità di accedere ai dati riguardanti i livelli dei diversi inquinanti nelle differenti zone degli USA, con diverse granularità temporali (giornaliera e annuale). Tali dati possono essere scaricati oppure consultabili tramite reports che forniscono informazioni di sintesi in forma tabellare o tramite rappresentazione su grafici. Su questa dashboard, l'utente può selezionare l'anno e la zona di interesse e per ogni città della zona scelta vengono mostrati i valori medi e massimi di AQI con la possibilità conoscere il numero di giorni in cui il valore di tale indice supera una certa soglia, i valori giornalieri dell'indice AQI dell'inquinante selezionato,
e può anche visualizzare uno Scatter Plot con i valori di AQI giornalieri durante uno specifico anno dell'inquinante di riferimento e di un altro sempre scelto dall'utente.

Altro sito consultato, che mette a disposizione anche una delle app descritte in precedenza è quello di Real Time AQI. Nella dashboard di questo sito, l'utente può consultare una mappa della città selezionata, visualizzandone per ogni stazione di monitoraggio il livello di AQI giornaliero.
Vengono anche visualizzati una serie di istogrammi relativi ai singoli inquinanti, si mostrano i trend dell'AQI nelle precedenti 48 ore e i rispettivi valori massimi e minimi e sempre con la stessa rappresentazione grafica e con lo stesso periodo temporale vengono monitorati i trend dei principali fenomeni atmosferici: Temperatura, Pressione, Umidità e Velocità del vento. Inoltre, tramite le previsioni metereologiche, vengono visualizzate le previsioni settimanali sul valore di AQI della città scelta.

Tra le app descritte, anche Air Quality in Europe presenta una dashboard molto simile a quella che è l'app per cui anche qui vengono riportati su mappa e in forma tabellare i valori di AQI sia lungo la strada, sia su fondo.

Inoltre sono state consultate anche alcune dashboard che non hanno un'app associata. Tra queste merita una menzione quella realizzata da OpenPuglia, relativa alla città di Bari, dove un grafico molto interessante è il grafico a linee relativo ai 60 giorni precedenti all'ultimo a disposizione riguardante tutte le stazioni di monitoraggio di Bari.

4.3 Tecnologie disponibili

- *Android*

E' basato sul kernel Linux ed è un sistema operativo per dispositivi mobili sviluppato da Google Inc. E' stato progettato soprattutto per smartphone e tablet ed è il sistema operativo per mobili più diffuso al mondo. Android ha una comunità di sviluppatori vastissima che permette di realizzare applicazioni con l'obiettivo di aumentare le funzionalità dei dispositivi. Il

linguaggio di programmazione utilizzato è Java. Per quanto riguarda l'interfaccia utente di Android si utilizzano gli ingressi mono e multi-touch come strisciate, tocchi e pizzichi sullo schermo per manipolare gli oggetti visibili sullo stesso.
Le applicazioni (scaricabili da Google Play e da Amazon Appstore e F-Droid) sono la forma più generica per indicare i software applicativi installabili su Android. Per installare queste applicazione, il distributore del software può fornire direttamente un file APK dal quale installare l'applicazione.

Queste applicazioni hanno un sistema di certificazione che verifica l'integrità del pacchetto da installare e la sua paternità.

- *iOS*

E' un sistema operativo sviluppato da Apple ed utilizzabile solo per i suoi prodotti (iPod, iPhone e iPad). Il linguaggio di programmazione necessario per realizzare applicazoni iOS è l'Objective C ossia una estensione del linguaggio C. Per realizzare queste applicazioni l'IDE offerto agli sviluppatori è l'XCode. Tuttavia ultimamente è stato creato un nuovo linguaggio di programmazione per sviluppare applicazioni iOS che è il linguaggio SWIFT, concepito per coesistere con il linguaggio Objective C, semplificandone la scrittura del codice.

- *MySQL & PHP*

Nel mondo dell'IT esistono differenti software di tipo server, detti Database management System(DBMS), utili per la gestione dei database. Tra questi, ha riscosso notevole successo, anche molto velocemente, MySQ: un RDBMS (Relational Databse Management System) open source composto da un client a riga di comando e un server.I punti di forza che hanno permesso a MySQL di raggiungere il successo, sono stati:

• alta efficienza nonostante le moli di dati affidate;

• integrazione di tutte le funzionalità che offrono i migliori DBMS: indici, trigger e stored procedure, ecc.

• altissima capacità di integrazione con i principali linguaggi di programmazione, ambienti di sviluppo e suite di programmi da ufficio (ODBC, Java, Mono, .NET, PHP, Python).

Per utilizzare e amministrare i database MySQL sono stati ideati dei Manger appositi Il più importante è phpMyAdmin; esso permette di creare database e tabelle e di eseguire operazioni di ottimizzazione. Per l'utilizzo di phpMyAdmin è necessario un server web come Apache HTTP Server ed il supporto del linguaggio PHP. MYSQL, al momento della creazione delle tabelle fornisce un feedback per evitare eventuali errori. Inoltre, mette a disposizoni altre funzionalità riguardanti l'inserimento dei dati (popolazione del database), per le query, per il backup dei dati, ecc. Com detto in precedenza, per utilizzare il DBMS MySQL è necessario phpMyAdmin che a sua volta deve supportare il linguaggio server-side PHP. Esso risiede in un server in remoto e tramite la richiesta HTTP da parte del client, in fase di esecuzione interpreta le informazioni ricevute,le elabora e restituisce un risultato al client che ha formulato la richiesta. Il PHP è stato sviluppato principalmente per il Web, infatti consente di accedere alle richieste HTTP di tipo GET e POST; questo è solo un esempio delle caratteristiche di questo linguaggio implementate in funzione del Web. Un altro esempio importante è l'accesso in lettura/scrittura ai cookie del browser e il supporto alle sessioni sul server.

Infine, PHP mette a disposizione delle librerie per accedere ai database (MySQL, Postgres, SQLite, ecc), per interagire con i server Web (Apache), per manipolare le immagini (GD) e per effettuare connessioni remote (cUrl). [7] [8]

- *Google Data Studio 360*

Google Data Studio 360 è un servizio che trasforma i dati a disposizione in rapporti e dashboard dinamici e facili da leggere, da condividere e da personalizzare. Con Data Studio si può:

- Connettere i dati ai rapporti provenienti da database o da Fogli Google, Google Analytics.
- Creare dimensioni, metriche e calcoli per ripulire e trasformare i dati senza aggiornare i dati non elaborati.
- Visualizzare differenti tipologie di grafici come serie temporali, grafici a barre, grafici a torta, ecc.
- Personalizzare graficamente la dashboard
- Condividere l'accesso ai rapporti con altri, dando l'opportunità di modificare tali rapporti.

- *Javascript*

Javascript è un linguaggio lato client orientato ad oggetti e ad eventi, che tramite funzioni di script permette di creare effetti dinamici interattivi su siti web e applicazioni web. Tali effetti scaturiscono da eventi innescati dall'utente stesso della pagina web. Tali funzioni di script, utilizzati dunque nella logica di presentazione, possono essere opportunamente inserite in file HTML, in pagine JSP o in appositi file separati con estensione .js poi richiamati nella logica di business.

L'accesso ad operazioni specifiche è fornito dal programma ospite (quello che ospita ed esegue lo script) tramite un'API ben definita implementata dal programma ospite e fornita allo script. Per uno script JavaScript, l'esempio più noto di programma ospite è quello del browser. Un browser tipicamente incorpora un interprete JavaScript; questo quando viene visitata una pagina web interpreta il codice JavaScript che è stato portato in memoria primaria, e lo esegue. JavaScript riesce a rapportarsi con un browser tramite delle interfacce chiamate DOM (Document Object Model in italiano Modello a Oggetti del Documento). In ambito Web, nelle pagine HTML, viene integrato codice JavaScript basato sulla scrittura di piccole funzioni che interagiscono con il DOM del browser permettendo azioni come controllare i valori nei campi di input, nascondere o visualizzare determinati elementi, ecc., non possibili con il semplice HTML statico. [9]

- *AngularJS*

Per semplificare lo sviluppo di applicazioni web, Google ha sviluppato un web framework open source denominato AngularJS. Tale framework legge prima la pagina HTML, che ha incapsulato degli attributi personalizzati addizionali (esempio: ng-controller), interpreta questi attributi come delle direttive (comandi) e lega le parti di ingresso e uscita della pagina al modello. QUesto modello è rappresentato da variabili Javascript a cui si può impostare manualmente il valore, tramite codice o recuperandolo da risorse JSON statiche o dinamiche.

AngularJS in pratica fornisce strumenti per costruire un'architettura modulare e testabile della logica applicativa dell'applicazione, ma allo stesso tempo si basa sull'approccio dichiarativo dell'HTML nella definizione dell'interfaccia grafica.

AngularJS permette di creare le Single Page Application, applicazioni moderne che sfruttano le più recenti tecnologie, esse hanno la particolarità di avere risorse che vengono caricate dinamicamente su richiesta, senza necessità di ricaricare l'intera pagina.

Gran parte del codice da scrivere viene eliminato grazie all'associazione di dati in AngularJS e all'iniezione di dipendenze. In questo modo, si risparmia molto tempo e l'interpretazione del tutto avviene all'interno del browser. Questo framework strutturale raggiunge quindi l'obiettivo di aumentare la capacità MVC (Model View Controller) delle applicazioni web. [10]

- *Analisi e scelta delle tecnologie*

Per quanto riguarda la scelta tra creare un'applicazione Android o un'applicazione iOS ci si basa soprattutto sulle competenze dello sviluppatore e sui dispositivi a disposizione. Questi fattori hanno portato alla scelta dello sviluppo di un'applicazione Android e non iOS.

Dal punto di vista della realizzazione della dashboard, in questa tesi si è andata ad integrare una dashboard già esistente e realizzata con MySQL e php. Tuttavia, sempre più frequenti sono l'utilizzo di alcuni software (tra cui Google Data Studio 360) per realizzare le dashboard. Per cui è doveroso fare un confronto tra Google Data Studio 360 e il DBMS MySQL con php e da esse capirne il motivo per cui in questo caso si è optato per la seconda opzione.

In particolare, MySQL e PHP permettono la gestione delle relazioni tra tabelle diverse (Join) a differenza di Google Data Studio in cui la consultazione dei dati da parte dell'utente può avvenire solo tramite reports relativi ad una singola tabella. Un'altra importante differenza è che con Google Data Studio i report vengono realizzati senza dover programmare scrivendo righe di codice lato server e senza dover scrivere query SQL. Questo può risultare un vantaggio per chi non ha competenze di programmazione poiché permette comunque di creare report in maniera rapida e veloce. Tuttavia questo può rappresentare un limite, e per lo sviluppo preposto in questa tesi della sezione relativa alla dashboard, effettivamente lo è, poiché per il calcolo dell'AQI è stato necessario scrivere del codice php in quanto nelle tabelle caricate non c'è il valore di AQI, ma c'è il valore di concentrazione di ogni singolo inquinante, il quale in php viene trasformato nel rispettivo valore di AQI. Questo in Google Data Studio non sarebbe possibile.

Altro fattore che ha influenzato la scelta, è quello relativo alla visualizzazione su mappa, che è uno dei metodi di visualizzazione scelti nello sviluppo della sezione da integrare nella dashboard. Entrambe le tecnologie permettono l'utilizzo di esse, con la differenza che con Google Data Studio il massimo focus della mappa è per città. Invece, utilizzando MySQL e PHP, si possono sfruttare le API di Google per mostrare informazioni dettagliate sulla singola città, caratteristica idonea a quello che è stato sviluppato in quanto si vuole andare nel dettaglio della città di Milano.

In aggiunta, con MySQL e PHP rispetto a Google Data Studio, è possibile indirizzare maggiormente l'utente rispetto ai parametri da inserire in input per consultare i reports.

In virtù di quanto detto, si è optato per implementare il tutto con MySQL e php.

Dal punto di vista dello sviluppo lato client si è preferito usare JavaScript piuttosto che AngularJS, in quanto la quantità e complessità di codice lato client non era tale da usufruire dei vantaggi della tecnologia AngularJS.

5. Framework proposto

Questo capitolo spiega in maniera dettagliata tutte le fasi da percorrere per raggiungere l'obiettivo finale di creare l'applicazione Android e sviluppare la sezione riguardante la dashboard. Dopo aver visto gli aspetti teorici legati all'obiettivo prefissato, si andrà a progettare e realizzare il DW di riferimento e successivamente si realizzerà la dashboard e l'applicazione mobile per la visualizzazione e l'analisi delle informazioni contenute nella base dati, da mostrare agli utenti.

Il primo passo è quello di definire le tipologie di dati su cui costruire il DW e di conseguenza gli indicatori di interesse (KPI). I dati presi in considerazione sono di tre tipi e riguardano la città di Milano per il periodo che va dal 2015 al 2017:

- Misurazioni sulle concentrazioni di inquinanti nelle aree urbane
- Condizioni climatiche nelle aree urbane
- Dati relativi al traffico di veicoli

Nei paragrafi successivi, verranno descritti nel dettaglio le fonti dei dati sopra citati, verrà effettuata la progettazione concettuale e logica del DW basata su questi dati e verranno indicati i KPI scelti per presentare le informazioni di interesse agli utenti che andranno ad usare la dashboard e l'applicazione mobile.

Nella figura 5.1 è riportato in maniera grafica il framework del progetto.

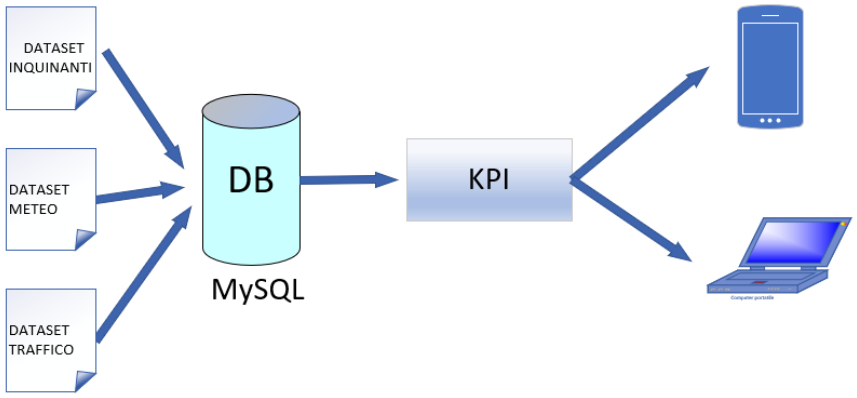


Figura 5.1: Framework

5.1 Fonti dati

Questo paragrafo presenta le tre fonti di dati considerate per il progetto, ognuna di esse fornisce una delle tipologie di dati prima citate. L'analisi è stata focalizzata sui dati della città di Milano.

1. *Misurazioni sulle concentrazioni di inquinanti*: questi dati sono estratti dal dataset di ARPA Lombardia e comprende i valori di concentrazione per un insieme di inquinanti (PM10, PM2.5, NO2, CO, O3, SO2, C6H6) raccolti attraverso alcune stazioni di monitoraggio situate nel Comune di Milano. Ogni stazione è dotata di un set di sensori, ognuno dei quali misura un determinato inquinante I dati forniti consistono in una serie di letture orarie o giornaliere, a seconda del tipo di inquinante (PM10 e PM2.5 solo giornaliere). Ogni lettura è caratterizzata dall'identificatore della stazione di monitoraggio, dall'identificatore del

sensore, dal nome dell'inquinante misurato, dalla concentrazione misurata dell'inquinante e dalla data e ora della lettura.

2. *Condizioni climatiche*: Il set di dati riguardanti le misurazioni metereologiche sono estratti da Weather Underground e sono raccolti dagli utenti attraverso tutti i registri di Personal Weather Station (PWS) più vicini alla città di Milano. Per ciascuna PWS selezionata, le misurazioni meteorologiche considerate (Temperatura, Pressione, Umidità, Precipitazioni, Velocità del vento) sono relative all'anno 2015. Ogni registrazione include l'identificatore della PWS, il timestamp della misurazione e per ogni fenomeno meteorologico il corrispondente valore meteorologico misurato. I dati meteorologici raccolti sono stati poi aggregati attraverso diverse PWS per ottenere un valore rappresentativo alla granularità giornaliera associata a ciascuna stazione di monitoraggio degli inquinanti.
3. *Dati relativi al traffico di veicoli*: questi dati sono forniti dal comune di Milano sotto forma di informazioni aggregate sul numero di veicoli che entrano nella zona centrale di Milano giornalmente. Il set di dati contiene il numero totale giornaliero di veicoli in entrata e il numero giornaliero di veicoli separatamente per ciascuna categoria di veicolo. I veicoli sono classificati in base al tipo di carburante (Benzina, Diesel, Elettrico, GPL-Metano, Ibrido, GPL, Miscela, Ibrido Gasolio, Metano, Ibrido Benzina), al tipo di veicolo (Bus, Merci, Privati, Altro) e alla categoria di Euro. I dati sul traffico qui descritti sono relativi al periodo 2015-2016.

5.2 Progettazione del Dimensional Fact Model

I dati sulla qualità dell'aria, delle condizioni metereologiche e del traffico vengono quindi archiviati in un unico data warehouse, la cui rappresentazione concettuale, secondo il Dimensional Fact Model, è riportata in figura 5.2.

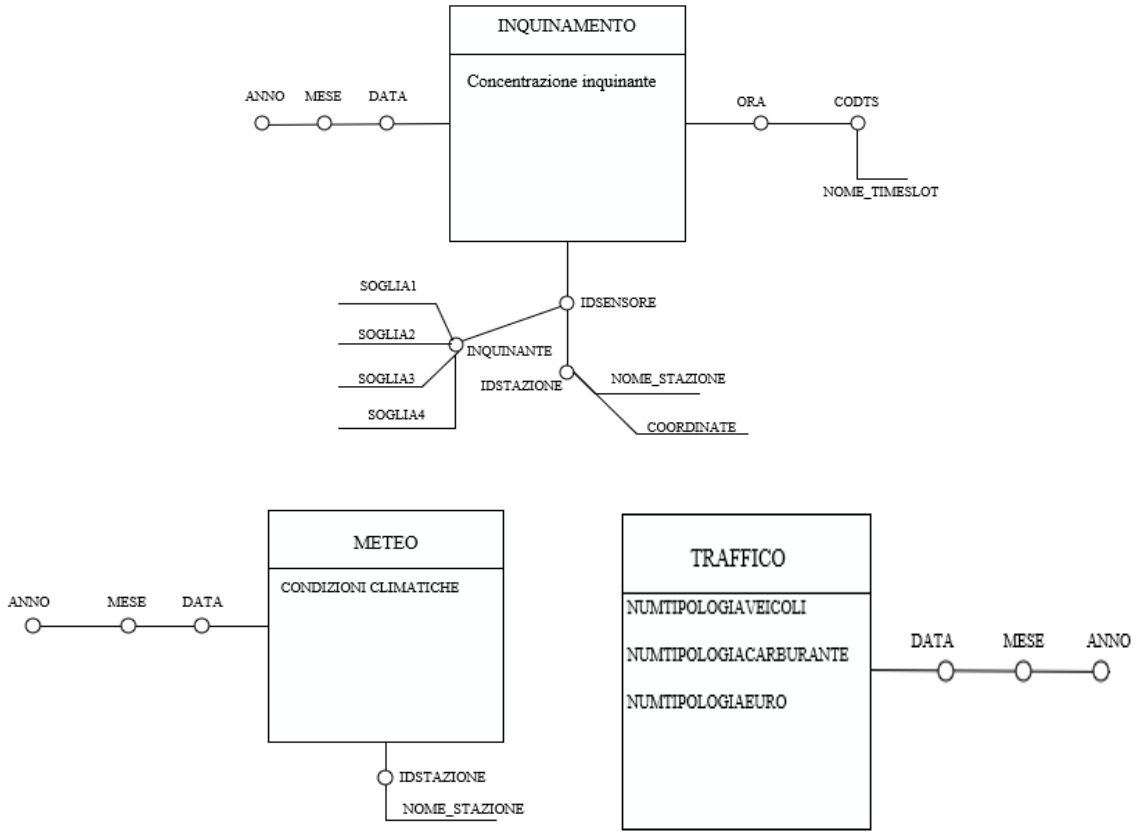


Figura 5.2: Dimensional Fact Model

Come si può notare, il DW progettato è composto da tre tabelle dei fatti, una riguardante l'inquinamento atmosferico, una riguardante le condizioni climatiche e una relativa al traffico.

Per quanto riguarda la tabella dei fatti relativa all'inquinamento, essa fornisce una misura sulla concentrazione dell'inquinante; inoltre, definisce tre gerarchie di dimensioni per analizzare la distribuzione spaziale e temporale della qualità dell'aria. Nello specifico, sono state definite una gerarchia delle dimensioni legate alla posizione collegata al dispositivo sensore che monitora la concentrazione di un determinato inquinante e due gerarchie di dimensioni temporali: una relativa alla data della misurazione e una relativa all'ora.

La tabella dei fatti corrispondente al meteo, ha come misura la condizione climatica e due gerarchie di dimensioni, una spaziale e una temporale. La prima indica la stazione in cui avviene il rilevamento climatico, la seconda indica la data in cui questa rilevazione viene effettuata.

Infine, la tabella dei fatti relativa al traffico urbano permette la misurazione del traffico da tre punti di vista: rispetto al tipo di veicolo, al tipo di carburante e rispetto alla categoria Euro. Questa tabella definisce una sola dimensione temporale con granularità giornaliera.

Successivamente al Dimensional Fact Model, viene mostrata la relativa Progettazione Logica, per avere maggiore facilità e intuitività per la costruzione effettiva delle tabelle che dovranno essere realizzare per lo sviluppo del progetto.

TEMPO-DATA (IdData, Data, Mese, Anno)

TEMPO-ORA (IdOra, Ora, TimeSlot, Nome_TS)

SENSORE (IdSensore, IdStazione, Nome_Stazione, Latitudine, Longitudine, Inquinante, Soglia1, Soglia2, Soglia3, Soglia4)

INQUINAMENTO (IdData, IdOra, IdSensore, Concentrazione_Inquinante)

TRAFFICO (IdData, NumTipoVeicolo, NumTipoAlimentazione, NumTipoEuro)

STAZIONE (IdStazione, Nome_Stazione)

METEO (IdStazione, IdData, Condizione_Climatica)

Le gerarchie relative alle dimensioni temporali forniscono diverse combinazioni di intervalli di tempo. In una gerarchia, il tempo si estende da una data all'altra, con l'indicazione del mese e dell'anno di appartenenza. Nell'altra gerarchia, il tempo si estende da ore a fasce orarie (mattina, pomeriggio, sera, notte).

La gerarchia delle dimensioni basata sulla posizione, prende come riferimento la posizione del sensore fisico usato nella misurazione e si estende fino all'indirizzo della stazione di appartenenza. In questa dimensione, ogni sensore misura uno specifico inquinante con le relative soglie di pericolosità.

Per i dati riguardanti le misurazioni delle condizioni climatiche, la gerarchia di dimensione basata sulla posizione ha una granularità diversa dalla precedente, in particolare prende come riferimento non più il sensore fisico ma la stazione di appartenenza del sensore che ha effettuato la rilevazione.

In realtà, quanto appena descritto, differisce dall'effettiva realizzazione delle tabelle utilizzate per il progetto. Per alcune dimensioni non sono state create delle tabelle dedicate ma si è preferito integrarle con quelle relative ai fatti, per rendere più semplice e veloce l'interrogazione della base dati. Quanto appena detto, si vedrà in maniera approfondita nei paragrafi successivi.

5.3 Progettazione dei Key Performance Indicator

I dati che compongono il Data Warehouse vengono analizzati per ottenere informazioni sulla condizione di inquinamento dell'aria nell'area urbana di Milano. Lo scopo dell'analisi è di produrre feedback utili agli utenti finali utilizzando Key Performance Indicator(KPI). Nel contesto del progetto sviluppato, i KPI sono indicatori quantitativi della condizione di inquinamento atmosferico monitorata. Inoltre, vengono utilizzati ulteriori KPI per quantificare le condizioni climatiche e di traffico dei veicoli. I KPI individuati sono:

A. *KPI di concentrazione di inquinanti spazio-temporali*: segnalano la concentrazione di inquinanti per stazione, a diversi livelli di granularità nel tempo, in una data specificata dall'utente:

    A1. Concentrazione media giornaliera degli inquinanti per stazione

    A2. Concentrazione media degli inquinanti per fascia oraria giornaliera per stazione

    A3. Concentrazione media oraria degli inquinanti per stazione

    A4. Concentrazione media mensile degli inquinanti per stazione

B. *KPI sul clima*: segnalano i valori medi climatici per stazione, a diversi livelli di granularità nel tempo, in un intervallo temporale specificato dall'utente

    B1. Valori medi giornalieri per i diversi fenomeni metereologici

    B2. Valori medi mensili per i diversi fenomeni metereologici

C. *KPI di traffico veicolare*: riportano il numero totale di veicoli giornalieri suddivisi per diverse categorie, entro un periodo temporale specificato dall'utente, nell'area del centro di Milano.

    C1. Valori totali giornalieri per categoria di veicolo

    C2. Valori totali giornalieri per tipologia di carburante

    C3. Valori totali giornalieri per categoria Euro

5.3.1 Air Quality Index (AQI)

L'Air Quality Index è un indice a 5 livelli che utilizza una scala che va da 0 a >100 nella notazione europea, in quella americana invece va da 0 a >500. Per ovvi motivi in questa tesi è stata utilizzata la scala europea. L'AQI è calcolato per ogni sensore utilizzando il relativo valore del KPI A1, per una data specificata dall'utente. Il valore di AQI di riferimento di ogni stazione è quello più alto tra i valori di AQI dei sensori presenti nella stazione. Per il calcolo dell'AQI di ogni inquinante si utilizza la formula (5.3.1).

$$AQI = \frac{(I_H - I_L)}{(C_H - C_{L)}} (C - C_L) + I_L \qquad (5.3.1)$$

dove:

AQI è l'indice che si vuole calcolare;

C è l'effettivo valore dell'inquinante (calcolato con il KPI A1);

C_H e C_L sono i limiti superiore e inferiore in cui ricade il valore di C (saranno diversi per ogni inquinante)

I_H e I_L sono i limiti superiore e inferiore corrispondenti ai valori di C_H e C_L presi in considerazione.

L'unità di misura di C e di C_H e C_L sono in $\mu g/m^3$

Si riportano nella tabella 5.3.1 le soglie dei singoli inquinanti e i relativi livelli di AQI:

Tabella 5.3.1. Livelli AQI

Livelli	**I_L e I_H**	**C_L e C_H di NO2**	**C_L e C_H di PM10**	**C_L e C_H di PM2.5**	**C_L e C_H di CO**	**C_L e C_H di O3**	**C_L e C_H di SO2**
Molto basso	$I_L = 0$ $I_H = 25$	$C_L = 0$ $C_H = 50$	$C_L = 0$ $C_H = 12.5$	$C_L = 0$ $C_H = 10$	$C_L = 0$ $C_H = 5000$	$C_L = 0$ $C_H = 60$	$C_L = 0$ $C_H = 50$
Basso	$I_L = 25$ $I_H = 50$	$C_L = 50$ $C_H = 100$	$C_L = 12.5$ $C_H = 25$	$C_L = 10$ $C_H = 20$	$C_L = 5000$ $C_H = 7500$	$C_L = 60$ $C_H = 120$	$C_L = 50$ $C_H = 100$
Medio	$I_L = 50$ $I_H = 75$	$C_L = 100$ $C_H = 200$	$C_L = 25$ $C_H = 50$	$C_L = 20$ $C_H = 30$	$C_L = 7500$ $C_H = 10000$	$C_L = 120$ $C_H = 180$	$C_L = 100$ $C_H = 300$
Alto	$I_L = 75$ $I_H = 100$	$C_L = 200$ $C_H = 400$	$C_L = 50$ $C_H = 100$	$C_L = 30$ $C_H = 60$	$C_L = 10000$ $C_H = 20000$	$C_L = 180$ $C_H = 240$	$C_L = 300$ $C_H = 500$
Molto alto	I>100	C > 400	C > 100	C > 60	C > 20000	C > 240	C > 500

E' di seguito riportato un esempio più chiaro su come calcolare l'AQI:

Si supponga di avere un valore di NO_2 (quindi nella formula sarebbe la C) pari a 113. La colonna di C_L e C_H di NO_2 di riferimento è quella in cui ricade il valore 113(C_L=100 e C_H=200). I corrispondenti valori di I_L e I_H sono 50 e 75. Quindi in base alla formula (5.3.1) si avrà il risultato nella equazione (5.3.2).

$$AQI = \frac{(75-50)}{(200-100)}\,(113-100) + 50 = 53.25 \qquad (5.3.2)$$

Il valore di AQI è 53.25 e ricade quindi nel livello Medio.
Se l'AQI è compreso tra 0 e 25 vuol dire che la qualità dell'aria che si respira è eccellente, se l'AQI è compresa tra 25 e 50 l'aria è buona, da 50 a 75 inizia a essere non salutare per l'uomo, da 75 a 100 è pericolosa, e se supera i 100 diventa molto pericolosa.

A partire dal calcolo dell'AQI si è creato un altro KPI rilevato come *KPI di concentrazione di inquinanti critica* il quale va ad analizzare la frequenza con cui i valori dell'Air Quality Index per inquinante sono critici. In particolare, riporta la percentuale di giorni entro un particolare periodo di tempo in cui il valore AQI per inquinante è critico in base ad una soglia specificata dall'utente.

5.4 Preparazione dei dati e creazione del Data Warehouse

In questa fase sono stati utilizzati gli strumenti ETL (Extraction, Transformation, Loading), utili per alimentare una sorgente dati singola, dettagliata, esauriente e di alta qualità che possa a sua volta alimentare il DW. Durante il processo di alimentazione del DW, la riconciliazione avviene in due occasioni: quando il DW viene popolato per la prima volta e periodicamente quando il DW viene aggiornato. Questa fase di riconciliazione consiste in quattro processi distinti:

- *Estrazione*: consiste nell'estrarre i dati rilevanti dalle sorgenti
- *Pulitura*: consiste nel migliorare la qualità dei dati (normalmente piuttosto scarsa nelle sorgenti)

- *Trasformazione*: consiste nella conversione dei dati dal formato operazionale sorgente a quello del DW
- *Caricamento*: consiste nel caricamento dei dati puliti e trasformati nel DW

Nella realizzazione di questo progetto, la fase di Trasformazione dei dati viene effettuata sia prima che dopo la fase di Caricamento, per una maggiore praticità nella gestione e conversione dei dati nel formato desiderato.

Si descrivono ora nel dettaglio, per ogni differente fonte dati descritta nei paragrafi precedenti (ARPA Lombardia, Wheater Underground e il portale del Comune di Milano) , l'applicazione degli strumenti ETL.

5.4.1 ARPA Lombardia

Per quanto riguarda i dati estratti da ARPA Lombardia, è stato necessario l'utilizzo del relativo sito per effettuare le richieste per lo scaricamento delle informazioni riguardanti i livelli di inquinamento. Tali richieste sono state fatte singolarmente per ogni stazione di monitoraggio presente nella città di interesse indicata (Milano) e per il periodo temporale indicato(2015-2017); le risposte di ARPA Lombardia alle richieste ricevute, sono state fornite tramite email in formato csv, suddividendo le misurazioni effettuate dalla stazione di monitoraggio in base ai sensori che la compongono (ogni sensore misura un differente inquinante).

Per facilitare la comprensione della struttura dei dati forniti da ARPA Lombardia, viene ora analizzata la risposta alla richiesta delle misurazioni dell'anno 2016 relative alla stazione di monitoraggio sita in Viale Marche.

Nella figura 5.3 è mostrata una parte del file csv ricevuto da ARPA Lombardia contenente le rilevazioni.

	A	B	C
1	5504,2016/01/01 01:00,61.0		
2	5504,2016/01/01 02:00,60.4		
3	5504,2016/01/01 03:00,64.4		
4	5504,2016/01/01 04:00,63.9		
5	5504,2016/01/01 05:00,62.4		
6	5504,2016/01/01 06:00,61.7		

Figura 5.3: file csv ARPA Lombardia

Come si può notare, sono presenti tre informazioni: le prime due indicano il codice del sensore che ha effettuato la rilevazione e quando essa è stata effettuata (data e ora), invece il terzo valore indica l'effettiva misurazione della concentrazione dell'inquinante. In allegato a tali rilevazioni, ARPA Lombardia fornisce una legenda composta come in figura 5.4.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Legenda											
2	-999 dato non disponibile											
3												
4	Riepilogo estrazione											
5	Id_Stazione,Nome_Stazione,Id_Sensore,Nome_Sensore,Unita_Misura,Periodo_Dal,Periodo_Al,Operatore											
6	501,Milano - viale Marche,5504,Milano - viale Marche Biossido di Azoto,µg/m³,2016/01/01 01:00,2017/01/01 00:00,Media											
7	501,Milano - viale Marche,5827,Milano - viale Marche Monossido di Carbonio,mg/m³,2016/01/01 01:00,2017/01/01 00:00,Media											
8	501,Milano - viale Marche,6328,Milano - viale Marche Ossidi di Azoto,µg/m³,2016/01/01 01:00,2017/01/01 00:00,Media											
9	501,Milano - viale Marche,17127,Milano - viale Marche Benzene,µg/m³,2016/01/01 01:00,2017/01/01 00:00,Media											
10												

Figura 5.4: legenda dati ARPA Lombardia

Da tale legenda è possibile ricavare l'identificativo e il nome della stazione, l'identificativo e i nomi dei sensori ad essa associati, l'unità di misura in cui sono effettuate le misurazioni, il periodo temporale di riferimento e la tipologia di misurazione effettuata. Inoltre, tale legenda indica che valori di misurazioni pari a -999 corrispondono all'assenza del dato stesso.

Per quanto riguarda il file mostrato in figura 5.3, contenente le misurazioni, le tre tipologie di informazioni sono state suddivise in altrettante colonne; in questo modo, è stato possibile lavorare singolarmente sulla colonna contenente le informazioni temporali, suddividendola ulteriormente in due colonne distinte: una contenente la data e l'altra contenente l'ora della misurazione. Con la singola Data, è stato possibile creare ulteriori due colonne contenenti rispettivamente le informazioni sul mese e l'anno della misurazione; queste informazioni saranno utili quando si vorranno estrarre i KPI con una granularità diversa da quella giornaliera.

Dopo aver effettuato tali trasformazioni, nella prima riga del file sono stati inseriti i nomi di quelli che saranno gli attributi della tabella che verrà successivamente usata nel DW.

In figura 5.5 è riportata la stessa parte del file precedente, aggiornata e pronta per essere caricata.

	A	B	C	D
1	Idsensore,Data,Ora,Valore,Mese,Anno			
2	5504,01/01/2016,01:00:00,61.0,01-2016,2016			
3	5504,01/01/2016,02:00:00,60.4,01-2016,2016			
4	5504,01/01/2016,03:00:00,64.4,01-2016,2016			
5	5504,01/01/2016,04:00:00,63.9,01-2016,2016			
6	5504,01/01/2016,05:00:00,62.4,01-2016,2016			
7	5504,01/01/2016,06:00:00,61.7,01-2016,2016			

Figura 5.5: file csv ARPA Lombardia da caricare

Raccogliendo le legende delle stazioni di monitoraggio, sono state applicate delle trasformazioni sui dati con l'obiettivo di creare un nuovo file contenente informazioni sulle diverse stazioni di monitoraggio di Milano, i loro rispettivi sensori e gli inquinanti da quest'ultimi misurati. Tale file sarà caricato anch'esso nel DW e utilizzato per creare tramite l'identificativo del sensore una corrispondenza tra la stazione, l'inquinante e la misurazione effettuata. In figura 5.6 è mostrato una parte del file appena descritto.

	A	B	C	D	E
1	Id_Stazione,Nome_Stazione,Id_Sensore,Inquinante				
2	501,Milano - viale Marche,5504,NO2				
3	501,Milano - viale Marche,5827,CO				
4	501,Milano - viale Marche,6328,NO				
5	501,Milano - viale Marche,17127,C6H6				
6	549,Milano - P.zza Abbiategrasso,5552,NO2				
7	549,Milano - P.zza Abbiategrasso,6344,NO				

Figura 5.6: legenda ARPA Lombardia da caricare

Come si può vedere da un facile confronto, rispetto alla struttura delle legende ricevute da ARPA Lombardia, alcune informazioni sono state rimosse perché ritenute inutili. In particolare, sono stati eliminati i dati relativi al periodo temporale (perché già contenuto nel file riguardante le misurazioni), all'unità di misura(perché tutti gli inquinanti sono misurati in µg/m3 ad eccezione del CO misurato in mg/m3) e alla tipologia di misurazione effettuata (perché sempre corrispondente alla Media). Inoltre, per una maggiore facilità di lettura, si è preferito sostituire al nome del sensore (contenente il nome dell'inquinante misurato) la formula chimica dell'inquinante.

A questo punto, sia il file delle misurazioni, sia quello riguardante le stazioni sono pronti per essere caricati nel DW in formato csv.

Per quanto riguarda il primo file, ossia quello delle misurazioni, esso è stato caricato e convertito in una tabella del database con il nome "inquinanti". Al momento della creazione della tabella, alcuni campi riguardanti le informazioni temporali (Data e Mese) hanno subito una conversione di formato diversa da quella desiderata (varchar anziché date).

Di conseguenza, è stato necessario convertire questi due attributi nel formato corretto e per fare questo sono stati seguiti i seguenti passaggi:

- Aggiunta di due colonne temporanee ('DataTemp' e 'MeseTemp') con formato date
- Esecuzione delle istruzioni sql:

  UPDATE *'inquinanti'* ***SET*** *'DataTemp'=(****SELECT*** ***STR_TO_DATE****('Data', %d/%m/%Y));*

  UPDATE *'inquinanti'* ***SET*** *'MeseTemp' =(****SELECT STR_TO_DATE****('Mese', %m/%Y));*

- Cancellazione delle colonne 'Data' e 'Mese'
- Aggiornamento del nome delle colonne 'DataTemp' e 'MeseTemp' rispettivamente in 'Data' e 'Mese'

Tali conversioni sono state necessarie per permettere la futura estrazione dei KPI descritti nei paragrafi precedenti. Infatti, senza le informazioni temporali nel formato date non sarebbe stato possibile consultare il DW per avere informazioni con granularità giornaliera o mensile (come indicato nei KPI). Inoltre, sempre per la possibilità di estrarre i KPI prima descritti, è stato ritenuto utile inserire due ulteriori colonne nella tabella "inquinanti", con l'obiettivo di raggruppare i dati in fasce orarie. In particolare, sono stati aggiunti i campi 'CodTS' e 'TimeSlot': il primo è un valore da 1 a 4 che identifica la fascia oraria di appartenenza della misurazione, il secondo è una descrizione letterale della fascia oraria stessa. Usando le seguenti istruzioni sql è stato possibile assegnare i valori corrispondenti a tali campi:

- ***UPDATE*** *'inquinanti'* ***SET*** *'CodTS'=1, 'TimeSlot'=Morning (0-11)* ***WHERE*** *'Ora' >=00:00:00 AND 'Ora' <=11:00:00;*
- ***UPDATE*** *'inquinanti'* ***SET*** *'CodTS'=2, 'TimeSlot'=Afternoon (12-16)* ***WHERE*** *'Ora' >=12:00:00 AND 'Ora' <=16:00:00;*
- ***UPDATE*** *'inquinanti'* ***SET*** *'CodTS'=3, 'TimeSlot'=Evening (17-20)* ***WHERE*** *'Ora' >=17:00:00 AND 'Ora' <=20:00:00*
- ***UPDATE*** *'inquinanti'* ***SET*** *'CodTS'=4, 'TimeSlot'=Night (21-23)* ***WHERE*** *'Ora' >=21:00:00 AND 'Ora' <=23:00:00*

In seguito a queste trasformazioni degli attributi della tabella "inquinanti", sono state effettuate delle operazioni su alcuni dati presenti nella tabella. In particolare, nella colonna 'Valore', in cui sono presenti le misurazioni effettuate, sono presenti dei valori pari a -999.0

che indicano (come detto prima) l'assenza della misurazione. Per evitare che tale valore possa influire in qualche modo nelle analisi successive basate sui KPI da estrarre, si è deciso di trasformare tramite l'operazione sql di UPDATE, il valore -999.0 in NULL.

Nella figura 5.7 è riportata la struttura della tabella "inquinanti", mentre nella figura 5.8 è rappresentato un esempio di come essa è stata popolata.

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito
1	**Idsensore**	int(5)			Si	*NULL*
2	**Data**	date			No	*Nessuno*
3	**Mese**	date			No	*Nessuno*
4	**Ora**	varchar(8)	utf8_general_ci		Si	*NULL*
5	**CodTS**	int(11)			No	*Nessuno*
6	**TimeSlot**	varchar(50)	utf8_general_ci		No	*Nessuno*
7	**Valore**	decimal(6,1)			Si	*NULL*
8	**Anno**	int(4)			Si	*NULL*

Figura 5.7: Struttura tabella "inquinanti"

Idsensore	Data ▼ 1	Mese	Ora	CodTS	TimeSlot	Valore	Anno
5722	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	26.3	2017
5550	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	36.0	2017
5506	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	31.6	2017
10280	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	0.9	2017
10282	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	36.8	2017
20004	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	2.7	2017
10279	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	45.8	2017
6372	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	39.3	2017
5841	2017-10-01	2017-10-00	00:00:00	1	Morning (0-11)	0.6	2017

Figura 5.8: Popolazione tabella "inquinanti"

Prima di concludere la descrizione di tale tabella è importante sottolineare un aspetto concettuale riguardante i sensori relativi agli inquinanti PM10 e PM2.5. A differenza degli altri sensori, questi effettuano rilevazioni con frequenza giornaliera e non oraria; di conseguenza, i campi 'Ora', 'CodTS' e 'TimeSlot' avranno rispettivamente valori pari a NULL, 0 e stringa vuota. Per cui le analisi sui KPI basati sui dati raggruppati per fasce orarie, non prenderanno in considerazione i valori delle misurazioni relative al PM10 e al PM2.5.

L'altro file ancora da analizzare è quello prima descritto riguardante le stazioni di monitoraggio con i relativi sensori. A differenza del caso precedente, dopo il caricamento del file non risultano necessarie ulteriori trasformazioni sui campi della tabella creata ("stazioni").

Vengono riportate rispettivamente in figura 5.9 e in figura 5.10, la struttura ed una parte della popolazione di tale tabella.

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito
1	Id_Stazione	int(3)			Sì	*NULL*
2	Nome_Stazione	varchar(28)	utf8_general_ci		Sì	*NULL*
3	Id_Sensore	int(5)			Sì	*NULL*
4	Inquinante	varchar(5)	utf8_general_ci		Sì	*NULL*

Figura 5.9: Struttura tabella "stazioni"

Id_Stazione	Nome_Stazione	Id_Sensore	Inquinante
501	Milano - viale Marche	5504	NO2
501	Milano - viale Marche	5827	CO
501	Milano - viale Marche	6328	NO
501	Milano - viale Marche	17127	C6H6
549	Milano - P.zza Abbiategrasso	5552	NO2
549	Milano - P.zza Abbiategrasso	6344	NO
547	Milano - Parco Lambro	5550	NO2
547	Milano - Parco Lambro	5722	O3

Figura 5.10: Popolazione tabella "stazioni"

Successivamente alla creazione di queste due tabelle, si è ritenuta necessaria l'implementazione di ulteriori due tabelle che fornissero informazioni aggiuntive riguardo alle stazioni di monitoraggio e riguardo agli inquinanti misurati. In particolare, per le stazioni di monitoraggio è stata creata la tabella "latlngstazioni" che per ogni stazione indica le relative coordinate geografiche (latitudine e longitudine); questo perché nello sviluppo della dashboard e dell'applicazione mobile facente parte di questo progetto, risultano necessarie le informazioni sulla posizione delle stazioni di monitoraggio.

Questa tabella è legata a "stazioni" tramite l'identificativo della stazione; nelle figure 5.11 e 5.12 sono mostrate la struttura e i dati contenuti.

#	Nome	Tipo	Codifica caratteri	Attributi	Null
1	Id_Stazione	int(11)			No
2	Lat	double			No
3	Lng	double			No
4	Nome_Stazione	varchar(100)	latin1_swedish_ci		No

Figura 5.11: Struttura tabella "latlngstazioni"

Id_Stazione	Lat	Lng	Nome_Stazione
501	45.4954866	9.1921222	Viale Marche
549	45.4304475	9.1741345	P.zza Abbiategrasso
547	45.4957003	9.2465629	Parco Lambro
539	45.4437788	9.1695722	Viale Liguria
705	45.4785456	9.2326961	Via Pascal
548	45.4704374	9.1952129	Via Senato
528	45.4631078	9.1938893	Via Verziere
503	45.4759561	9.1408294	P.zza Zavattari

Figura 5.12: Popolazione tabella "latlngstazioni"

La seconda tabella che è risultato utile creare è stata “soglie”, essa identifica per ogni inquinante le soglie di pericolosità da prendere come riferimento. Questa tabella è stata necessaria per fornire agli utenti della dashboard un indicatore cromatico differente a seconda della soglia in cui ricade il valore dell’inquinante. Essa è legata con la tabella “stazioni” tramite il nome univoco dell’inquinante.

Nelle figure 5.13 e 5.14 sono mostrate rispettivamente la struttura e i dati in essa contenuti.

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito
1	**Nome_Inquinante**	varchar(5)	utf8_general_ci		Sì	*NULL*
2	**Soglia1**	int(3)			Sì	*NULL*
3	**Soglia2**	int(3)			Sì	*NULL*
4	**Soglia3**	int(3)			Sì	*NULL*
5	**Soglia4**	int(3)			Sì	*NULL*

Figura 5.13: Struttura tabella “soglie”

Nome_Inquinante	Soglia1	Soglia2	Soglia3	Soglia4
PM10	25	50	75	100
PM2.5	20	40	60	80
NO2	100	200	300	400
O3	90	180	240	300
CO	5	10	15	20
SO2	62	125	187	250
C6H6	4	8	12	16

Figura 5.14: Popolazione tabella “soglie”

5.4.2 Portale della città di Milano

Per quanto riguarda i dati del traffico, tramite il portale del Comune di Milano è stato possibile scaricare le informazioni relative agli ingressi dei veicoli nell’Area C di Milano (corrispondente al centro cittadino). Questi dati riguardano gli ultimi due anni più recenti (2015 e 2016) e forniscono i valori giornalieri degli accessi in quest’area.

Al momento dello scaricamento dei dati, dal portale sarà possibile specificare la classificazione per cui i valori devono essere suddivisi; in particolare, per le necessità del progetto, quelle ritenute utili sono: per categoria veicolo, per tipo di carburante e per categoria Euro. In figura 5.15 sono mostrati una parte dei file (uno per ogni classificazione) in formato csv, contenenti i dati appena descritti.

	A	B	C	D
1	timeIndex,NC,ALTRO,BUS,MERCI,PERSONE			
2	19/08/2016,2155,235,506,1712,13667			
3	20/08/2016,2423,164,513,816,12923			
4	21/08/2016,2027,79,460,485,14661			
5	22/08/2016,2505,332,574,1963,13328			
6	23/08/2016,2640,419,517,2514,16355			
7	24/08/2016,2858,359,452,2176,14026			

	A	B	C	D	E	F	G
1	timeIndex,EURO0,EURO1,EURO2,EURO3,EURO4,EURO5,EURO6,NC						
2	19/08/2016,18,41,438,1157,5247,6713,2452,2209						
3	20/08/2016,34,57,365,1354,4109,5751,2739,2430						
4	21/08/2016,26,41,369,1073,4735,6902,2531,2035						
5	22/08/2016,24,13,392,1117,4540,7289,2756,2571						
6	23/08/2016,18,24,334,980,5325,10337,2755,2672						
7	24/08/2016,27,23,277,1024,4818,8030,2746,2926						

	A	B	C	D	E	F	G	H	I	J	K	L
1	timeIndex,BENZINA,DIESEL,ELETTRICO,GPL_METANO,IBRIDO,NC,GPL,MISCELA,IBRIDO GASOLIO,METANO,IBRIDO BENZINA											
2	19/08/2016,5640,7718,56,,,2159,1115,,6,582,999											
3	20/08/2016,5358,7024,8,,,2428,944,,17,278,782											
4	21/08/2016,5600,7725,10,,,2029,1050,,7,237,1054											
5	22/08/2016,4752,8114,74,,,2509,1395,,20,594,1244											
6	23/08/2016,5248,10767,41,,,2641,1691,,10,667,1380											
7	24/08/2016,5572,8189,70,,,2859,1460,,19,551,1151											
8	25/08/2016,5524,9863,53,,,2026,1643,,18,790,2374											

Figura 5.15: file csv Portale città di Milano

Come si può notare, i tre file hanno una struttura molto simile, infatti tutti e tre hanno una informazione temporale indicante la data della rilevazione e la numerosità dei veicoli entranti nell'area C, suddivisi per categoria a seconda della classificazione scelta.

Tutti questi file csv sono stati suddivisi per colonne, per distinguere al meglio i valori di ogni categoria e soprattutto per aggiungere ulteriori due colonne, contenenti informazioni temporali riguardo al mese e all'anno; informazioni che risulteranno utili in seguito, durante le analisi, per l'estrazione di alcuni KPI. In figura 5.16 sono riportate le stesse parti di file precedenti aggiornati e pronti per essere caricati.

	A	B	C	D	E
1	Data,NC,ALTRO,BUS,MERCI,PERSONE,Mese,Anno				
2	01/01/2015,9312,350,675,1352,63652,01-2015,2015				
3	02/01/2015,10793,1085,891,6520,69847,01-2015,2015				
4	03/01/2015,12293,890,729,3967,81270,01-2015,2015				
5	04/01/2015,12532,358,709,1911,78579,01-2015,2015				
6	05/01/2015,13239,1384,852,7981,87271,01-2015,2015				
7	06/01/2015,12381,520,780,2540,87908,01-2015,2015				

	A	B	C	D	E	F	G
1	Data,EURO0,EURO1,EURO2,EURO3,EURO4,EURO5,EURO6,NC,Mese,Anno						
2	01/01/2015,67,202,2691,7516,26085,28086,1377,9317,01-2015,2015						
3	02/01/2015,71,186,2598,7596,30179,35784,1884,10838,01-2015,2015						
4	03/01/2015,93,207,2939,8911,33425,39125,2127,12322,01-2015,2015						
5	04/01/2015,73,162,2680,8261,31452,36857,2047,12557,01-2015,2015						
6	05/01/2015,65,147,2748,8261,36255,47320,2639,13292,01-2015,2015						
7	06/01/2015,92,181,2830,8504,34252,43288,2583,12399,01-2015,2015						

	A	B	C	D	E	F	G	H	I	J	K	L
1	Data,BENZINA,DIESEL,ELETTRICO,GPL_METANO,IBRIDO,NC,GPL,MISCELA,IBRIDO GASOLIO,METANO,IBRIDO BENZINA,Mese,Anno											
2	01/01/2015,27461,29968,12,5515,3073,9312,0,0,0,0,0,01-2015,2015											
3	02/01/2015,29222,38132,47,7622,3320,10793,0,0,0,0,0,01-2015,2015											
4	03/01/2015,35091,41227,42,7216,3280,12293,0,0,0,0,0,01-2015,2015											
5	04/01/2015,33605,38512,29,6346,3066,12531,0,0,0,0,0,01-2015,2015											
6	05/01/2015,35797,47802,58,9269,4562,13239,0,0,0,0,0,01-2015,2015											
7	06/01/2015,36840,43655,28,7213,4013,12380,0,0,0,0,0,01-2015,2015											

Figura 5.16: file csv Portale città di Milano da caricare

A questo punto, i tre file così composti sono pronti per essere caricati nel DW in formato csv; alla conclusione del loro caricamento verranno create tre tabelle così definite. “tipalimentazione”, “catveicolo” e “euro”. Anche in questo caso, come in quello della tabella “inquinanti”, al momento della creazione, alcuni campi riguardanti le informazioni temporali (Data e Mese) subiscono una conversione di formato diversa da quella desiderata (varchar anziché date). Di conseguenza, è stato necessario convertire questi due attributi nel formato corretto nelle stesse modalità e medesimi passaggi descritti in precedenza. Dopo tali conversioni sarà possibile estrarre in una futura analisi alcuni KPI descritti nei paragrafi precedenti. Infatti, senza le informazioni temporali nel formato date non sarebbe stato possibile consultare il DW per avere informazioni con granularità giornaliera o mensile (come indicato nei KPI). Vengono ora riportate, rispettivamente nelle figure 5.17 e 5.18 la struttura di una delle tre tabelle ed un esempio di come essa è stata popolata (non vengono riportate le altre perché molto simili a quella mostrata).

	#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito
☐	1	**Data**	date			No	*Nessuno*
☐	2	**NC**	int(5)			Sì	*NULL*
☐	3	**ALTRO**	int(4)			Sì	*NULL*
☐	4	**BUS**	int(4)			Sì	*NULL*
☐	5	**MERCI**	int(5)			Sì	*NULL*
☐	6	**PERSONE**	int(6)			Sì	*NULL*
☐	7	**Mese**	date			No	*Nessuno*
☐	8	**Anno**	int(4)			Sì	*NULL*

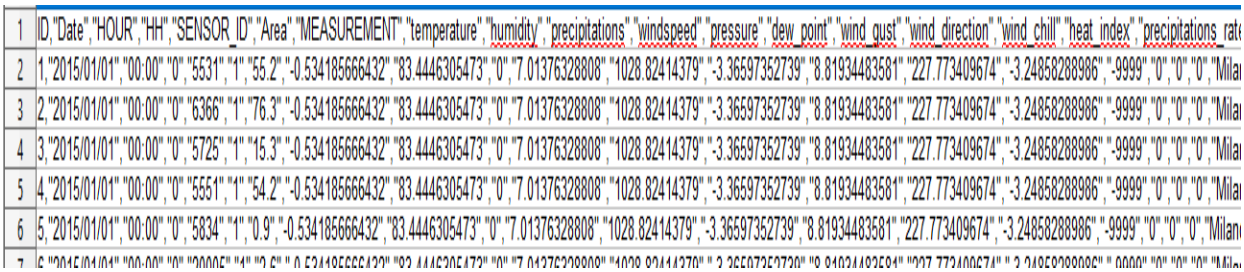
Figura 5.17: Struttura tabella “catveicolo”

Data	NC	ALTRO	BUS	MERCI	PERSONE ▲ 1	Mese	Anno
2016-01-07	1133	154	171	655	5057	2016-01-00	2016
2016-01-28	1043	182	212	1051	7829	2016-01-00	2016
2016-08-15	1283	71	384	365	10316	2016-08-00	2016
2016-08-17	1698	271	392	1804	11184	2016-08-00	2016
2016-08-14	2773	162	380	367	11515	2016-08-00	2016
2015-10-24	3941	119	154	509	11645	2015-10-00	2015
2016-08-16	2225	246	315	1566	12251	2016-08-00	2016
2016-08-18	2046	262	490	1601	12857	2016-08-00	2016
2016-08-13	2246	172	426	844	12891	2016-08-00	2016

Figura 5.18: Popolazione tabella “catveicolo”

5.4.3 Wheater Underground

Per quanto riguarda le informazioni metereologiche, queste sono state reperite tramite il portale Wheater Underground. Questo sito si occupa di fornire previsioni dettagliate del meteo di qualsiasi città a livello mondiale e mettendo a disposizione degli utenti la possibilità di scaricare dati o report sulle condizioni climatiche di una determinata città in uno specifico periodo temporale. In particolare, questa ultima funzione è risultata utile per avere a disposizione i dati del meteo della città di Milano dell’anno 2015. Il file scaricato contenente queste informazioni si presenta come nella figura 5.19:



1	ID,"Date","HOUR","HH","SENSOR_ID","Area","MEASUREMENT","temperature","humidity","precipitations","windspeed","pressure","dew_point","wind_gust","wind_direction","wind_chill","heat_index","precipitations_rat
2	1,"2015/01/01","00:00","0","5531","1","55.2","-0.534185666432","83.4446305473","0","7.01376328808","1028.82414379","-3.36597352739","8.81934483581","227.773409674","-3.24858288986","-9999","0","0","0","Mila
3	2,"2015/01/01","00:00","0","6366","1","76.3","-0.534185666432","83.4446305473","0","7.01376328808","1028.82414379","-3.36597352739","8.81934483581","227.773409674","-3.24858288986","-9999","0","0","0","Mila
4	3,"2015/01/01","00:00","0","5725","1","15.3","-0.534185666432","83.4446305473","0","7.01376328808","1028.82414379","-3.36597352739","8.81934483581","227.773409674","-3.24858288986","-9999","0","0","0","Mila
5	4,"2015/01/01","00:00","0","5551","1","54.2","-0.534185666432","83.4446305473","0","7.01376328808","1028.82414379","-3.36597352739","8.81934483581","227.773409674","-3.24858288986","-9999","0","0","0","Mila
6	5,"2015/01/01","00:00","0","5834","1","0.9","-0.534185666432","83.4446305473","0","7.01376328808","1028.82414379","-3.36597352739","8.81934483581","227.773409674","-3.24858288986","-9999","0","0","0","Milan
7	6,"2015/01/01","00:00","0","30005","1","0.6","-0.534185666432","83.4446305473","0","7.01376328808","1028.82414379","-3.36597352739","8.81934483581","227.773409674","-3.24858288986","-9999","0","0","0","Mila

Figura 5.19: file csv Wheater Underground

Le informazioni contenute sono risultate essere maggiori rispetto a quelle necessarie per le analisi da effettuare in questo progetto; per questo dopo aver suddiviso i dati per colonna, si è deciso di prendere in considerazione solo le condizioni climatiche ritenute correlabili con i livelli di inquinamento (temperatura, umidità, velocità del vento, pressione, precipitazioni).

La granularità dei dati a livello temporale è oraria, a livello spaziale riguarda i singoli sensori di ogni stazione presente nella città di Milano. Dopo aver osservato i valori metereologici misurati da ogni sensore della stessa stazione di monitoraggio, si è notato che questi fossero pressoché identici tra loro e di conseguenza si è deciso di portare la granularità ad un livello minore di dettaglio, cioè andando a considerare per ogni stazione nella medesima ora del medesimo giorno, solo un singolo valore rappresentativo dell'intera stazione di monitoraggio; andando quindi a rimuovere tutte le informazioni relative ai sensori.

Quindi riassumendo, le informazioni ritenute importanti da inserire nel DW del progetto, sono la data, l'ora, le condizioni climatiche prima citate e il nome della stazione che ha effettuato le rilevazioni. Inoltre, come visto anche in precedenza, sono state aggiunte altre due colonne: Mese e Anno, sempre con l'obiettivo di avere la possibilità di effettuare analisi con granularità differenti. Il file csv pronto per il caricamento nel DW è mostrato in figura 5.20.

	A	B	C	D	E	F	G	H	I	J	K
1	Date,Mese,Anno,Ora,temperature,humidity,wind_speed,pressure,precipitation_rate,Nome_Stazione										
2	01/01/2015,01-2015,2015,00:00,1.265838858,79.26020396,3.830803563,1030.76711,0,Milano - via Verziere										
3	01/01/2015,01-2015,2015,00:00,1.243883747,79.30366993,3.929370982,1030.369198,0,Milano - via Senato										
4	01/01/2015,01-2015,2015,00:00,1.200327454,78.85514945,4.694661182,1027.008018,0,Milano - Parco Lambro										
5	01/01/2015,01-2015,2015,00:00,1.226741512,78.86289216,4.512110389,1027.775475,0,Milano -via Carlo Pascal										

Figura 5.20: file csv Wheater Underground da caricare

Dopo essere stato caricato sono state necessarie altre operazioni per rendere questi dati utilizzabili. Per prima cosa, anche in questo caso, come in quello delle tabelle precedentemente descritte, al momento della creazione alcuni campi riguardanti le informazioni temporali (Data e Mese) subiscono una conversione di formato diversa da quella desiderata (varchar anziché date). Di conseguenza, è stato necessario convertire questi due attributi nel formato corretto nelle stesse modalità e medesimi passaggi descritti in precedenza.

Dopodiché, si è notato come alcuni valori delle misurazioni dei fenomeni metereologici fossero concettualmente errati: sono stati rilevati valori al di fuori dei range possibili per questi fenomeni (es. temperatura -300°C); in questi casi, si è provveduto alla sostituzione di tali valori con il valore di default NULL, per evitare che questi influissero sul valore dei KPI che saranno estratti tramite questa tabella.

Infine, dal punto di vista temporale, come già detto in precedenza nel paragrafo relativo al DFM, l'interesse era quello di avere dati metereologici con granularità giornaliera; quindi, in virtù del fatto che le informazioni fornite dal portale hanno una granularità maggiore (oraria),

la scelta è stata quella di effettuare una media giornaliera per ogni stazione per ogni condizione metereologica misurata; questo è stato possibile tramite un'interrogazione sql. In base al risultato è stata aggiornata l'intera tabella "meteo", passando così da una granularità oraria ad una giornaliera. Nelle figure 5.21 e 5.22 sono riportate rispettivamente la struttura di tale tabella e una parte della popolazione di essa, successive alle operazioni prima descritte.

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito
1	Data	date			No	*Nessuno*
2	Mese	date			No	*Nessuno*
3	Anno	varchar(50)	latin1_swedish_ci		No	*Nessuno*
4	temperature	double			Sì	*NULL*
5	humidity	double			Sì	*NULL*
6	wind_speed	double			Sì	*NULL*
7	pressure	double			Sì	*NULL*
8	precipitation_rate	double			Sì	*NULL*
9	Nome_Stazione	varchar(50)	latin1_swedish_ci		No	*Nessuno*

Figura 5.21: Struttura tabella "meteo"

Data	Mese	Anno	temperature	humidity	wind_speed	pressure	precipitation_rate	Nome_Stazione
2015-01-01	2015-01-00	2015	1.265838858	79.26020396	3.830803563	1030.76711	0	Milano - via Verziere
2015-01-01	2015-01-00	2015	1.243883747	79.30366993	3.929370982	1030.369198	0	Milano - via Senato
2015-01-01	2015-01-00	2015	1.200327454	78.85514945	4.694661182	1027.008018	0	Milano - Parco Lambro
2015-01-01	2015-01-00	2015	1.226741512	78.86289216	4.512110389	1027.775475	0	Milano -via Carlo Pascal
2015-01-01	2015-01-00	2015	1.323304423	79.35583162	3.349713785	1032.815653	0	Milano - P.zza Abbiategrasso
2015-01-01	2015-01-00	2015	1.290933842	79.66544131	3.233451539	1033.409764	0	Milano - viale Liguria
2015-01-01	2015-01-00	2015	1.199877418	80.29196117	3.166798345	1033.905607	0	Milano - P.zza Zavattari
2015-01-01	2015-01-00	2015	1.19730829	79.46449143	4.065368521	1029.855049	0	Milano - viale Marche

Figura 5.22: Popolazione tabella "meteo"

Le unità di misura utilizzate per i valori delle condizioni climatiche della tabella "meteo" sono: °C per la temperatura, % per l'umidità, Kmh per la velocità del vento, hPa per la pressione e % per i livelli di precipitazione.

5.4.4 Viste

Dopo aver effettuato la descrizione dettagliata delle tabelle che compongono il DW del progetto, sono necessarie alcune considerazioni sull'utilizzo di tali tabelle. Analizzando i KPI precedentemente descritti, si nota che molte delle interrogazioni che verranno effettuate sul DW, per estrarre dati utili agli utenti, riguardano principalmente tre tabelle: "inquinanti", "stazioni" e "latlngstazioni".

In particolare, molte delle informazioni e analisi che verranno mostrate all'utente tramite la dashboard e l'applicazione mobile, richiedono l'interrogazione della tabella "inquinanti" per l'estrazione dei dati necessari per il calcolo dei KPI. La maggior parte di quest'ultimi richiedono una granularità giornaliera ma la tabella in questione, ha una granularità oraria; per cui, ogni qualvolta un utente accede a report o analisi giornalieri, sarebbe necessaria una interrogazione al DW che comporti una elaborazione per raggruppare i dati orari presenti nella tabella e calcolare i rispettivi valori giornalieri. Naturalmente, affinché questi calcoli siano disponibili all'utilizzo su dashboard o app, è necessario un certo tempo di elaborazione, dovuto all'elevato numero di record presenti nella tabella "inquinanti" (oltre 700 mila righe).

Considerando la frequenza con il quale si interrogherebbe tale tabella e l'elevata numerosità di essa, la soluzione trovata è quella di creare e utilizzare una così detta Vista.

5.4.4.1 Cosa è una vista

Le viste sono un elemento utilizzato dalla maggior parte dei DBMS. Una vista è rappresentata da una query (SELECT), il cui risultato può essere utilizzato come se fosse una tabella. Generalmente i DBMS rielaborano le query sulle viste in modo che agiscano sulle tabelle che fanno parte della vista stessa. Le viste generalmente vengono utilizzate per semplificare le query. Leggere un insieme di dati avente un significato potrebbe essere complesso, perché potrebbe richiedere eccessive JOIN fra tabelle; con una vista è possibile semplificare molto la stesura di query che leggono le informazioni.

Un altro scopo delle viste potrebbe essere semplificare o potenziare la gestione dei permessi. Ad esempio, si potrebbe creare una query che legge solo alcuni dati da una tabella (tramite la clausola WHERE), per poi assegnare il permesso in lettura ad un certo utente sulla vista, ma non sulla tabella di base. In questo modo l'utente non vedrà i dati che non vengono estratti dalla vista.

Le viste possono essere aggiornabili, cioè è possibile eseguire su di esse comandi DML come INSERT, UPDATE e DELETE.

5.4.4.2 Utilizzo della vista

Quindi, dopo aver capito cosa è e a cosa serva una vista, si è passati a crearne una. Come già detto, la vista da dover creare dovrà permettere il raggruppamento dei dati contenuti nella tabella "inquinanti", passando ad una granularità giornaliera e di conseguenza diminuendo il numero di righe. Inoltre, con lo scopo di facilitare la futura stesura di query relative alle informazioni su inquinanti e stazioni di monitoraggio, all'interno della vista sono stati inseriti i dati provenienti dalle tabelle "latlngstazioni" e "stazioni", al fine di evitare il frequente uso dell'operatore JOIN per accedere a dati legati tra loro ma contenuti in differenti tabelle.

Per cui, dopo la creazione della vista, ogni qualvolta tramite dashboard o applicazione mobile sarà necessario l'accesso ai dati relativi alle misurazioni giornaliere degli inquinanti e alle stazioni (con relativa posizione geografica) che hanno effettuato tali misurazioni, sarà sufficiente interrogare questa vista con inoltre l'utilizzo di query molto più semplici da scrivere.

Quindi, la vista che si è deciso di creare prenderà il nome di "viewdaily" e sarà possibile realizzarla tramite la seguente query sql:

```
CREATE VIEW viewdaily AS
   (SELECT latlngstazioni.Lat, latlngstazioni.Lng, stazioni.Nome_Stazione, inquinanti.Data,
           stazioni.Inquinante, AVG(inquinanti.Valore) AS Media_Giornaliera,
           inquinanti.Mese, inquinanti.Anno
    FROM inquinanti, stazioni, latlngstazioni
    WHERE stazioni.Id_Sensore = inquinanti.Idsensore
        AND stazioni.Id_Stazione = latlngstazioni.Id_Stazione
    GROUP BY stazioni.Nome_Stazione, inquinanti.Data, stazioni.Inquinante,
             inquinanti.Mese, inquinanti.Anno )
```

Il risultato di questa query è quindi una vista composta come mostrato in figura 5.23.

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito
1	**Lat**	double			No	*Nessuno*
2	**Lng**	double			No	*Nessuno*
3	**Nome_Stazione**	varchar(28)	latin1_swedish_ci		Sì	*NULL*
4	**Data**	date			No	*Nessuno*
5	**Inquinante**	varchar(5)	latin1_swedish_ci		Sì	*NULL*
6	**Media_Giornaliera**	decimal(10,5)			Sì	*NULL*
7	**Mese**	date			No	*Nessuno*
8	**Anno**	int(4)			Sì	*NULL*

Figura 5.23: Struttura vista "viewdaily"

Tale vista è formata da circa 35 mila record, numerosità molto minore rispetto alle oltre 700 mila righe della sola tabella "inquinanti". Un esempio di tali record è riportato nella figura 5.24.

Lat	Lng	Nome_Stazione	Data	Inquinante	Media_Giornaliera	Mese	Anno
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-01	C6H6	*NULL*	2015-01-00	2015
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-01	CO	0.97391	2015-01-00	2015
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-01	NO	97.04783	2015-01-00	2015
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-01	NO2	56.52609	2015-01-00	2015
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-02	C6H6	*NULL*	2015-01-00	2015
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-02	CO	1.25000	2015-01-00	2015
45.4759561	9.1408294	Milano - P.zza Zavattari	2015-01-02	NO	146.09167	2015-01-00	2015

Figura 5.24: Popolazione vista "viewdaily"

Con la creazione e l'utilizzo della vista "viewdaily", uno dei problemi che si aveva necessità di risolvere, cioè diminuire il tempo di estrazione dei dati utili alle analisi, non ha trovato soluzione. La causa di ciò è che questo tipo di vista non viene scritta fisicamente su disco da parte del DBMS, ma viene creata ogni qualvolta essa viene richiamata in una interrogazione sql. Di conseguenza, permette di ridurre il numero di righe da consultare e diminuisce la complessità delle query da scrivere ma non influisce positivamente sul tempo di esecuzione della query stessa, andando a volte ad aumentarne il tempo di attesa.

Per ovviare anche a questo problema, si passa all'utilizzo delle viste materializzate: esse sono viste scritte fisicamente su disco per consentirne una lettura più rapida. In pratica, essa non è altro che la copia fisica della vista originale; in questo modo, richiamando nelle interrogazioni sql la vista materializzata anziché quella originale, si diminuisce drasticamente il tempo di esecuzione perché non sarà necessario al DBMS ricreare ogni volta la vista. Nel caso in questione, la vista materializzata creata sulla base di "viewdaily" prende il nome di "vistagiornaliera".

Uno degli aspetti da prendere in considerazione quando si parla di viste non materializzate ("viewdaily"), riguarda il loro aggiornamento: infatti, ogni qualvolta vengono inseriti nuovi dati all'interno di tabelle utilizzate per la creazione della vista, quest'ultima si aggiorna automaticamente inserendo i nuovi record.

Per le viste materializzate ("vistagiornaliera") invece, l'aggiornamento non avviene nello stesso momento di quello della vista originale, ma sarà il DBMS a intervalli regolari ad aggiornarla inserendo i nuovi record della vista originale.

5.5 Estrazione e calcolo dei KPI dal data warehouse

Come visto nel paragrafo 5.3, l'inserimento dei dati all'interno del DW aveva come obiettivo quello di estrarre determinati KPI. Dopo aver descritto nel dettaglio la struttura delle tabelle del DW, si possono ora scrivere le query sql per l'estrazione di tali KPI.

In alcuni casi, è necessario una elaborazione del risultato della relativa query per permettere l'effettivo calcolo del KPI. Tale elaborazione è stata sviluppata in linguaggio PHP.

A1. Concentrazione media giornaliera degli inquinanti per stazione

```
SELECT 'Nome_Stazione','Data','Inquinante','MediaGiornaliera'
FROM 'vistagiornaliera'
GROUP BY 'Nome_Stazione','Data','Inquinante'
```

A2. Concentrazione media degli inquinanti per fascia oraria giornaliera per stazione

```
SELECT S.'Nome_Stazione',I.'Data',S.'Inquinante',I.'TimeSlot',AVG(I.'Valore')
FROM 'stazioni' as S,'inquinanti' as I
WHERE S.'Id_Sensore'=I.'Idsensore'
GROUP BY S.'Nome_Stazione',I.'Data',S.'Inquinante',I.'TimeSlot'
```

A3. Concentrazione media oraria degli inquinanti per stazione

```
SELECT S.'Nome_Stazione',I.'Data',S.'Inquinante',I.'Ora',I.'Valore'
FROM 'stazioni' as S,'inquinanti' as I
WHERE S.'Id_Sensore'=I.'Idsensore'
GROUP BY S.'Nome_Stazione',I.'Data',S.'Inquinante',I.'Ora'
```

A4. Concentrazione media mensile degli inquinanti per stazione

```
SELECT 'Nome_Stazione','Mese','Inquinante',AVG('MediaGiornaliera')
FROM 'vistagiornaliera'
GROUP BY 'Nome_Stazione','Mese','Inquinante'
```

Nel paragrafo 5.3.1 è descritto l'Air Quality Index (AQI): per l'estrazione di questo KPI è stato necessario, anche in questo caso, utilizzare del codice PHP per elaborare il risultato di una query sql. Tale query restituirà i valori del KPI A1 e tramite istruzioni php sarà possibile di volta in volta identificare l'inquinante e di conseguenza le soglie (come nella tabella 2.1) relative a tali valori per poi calcolare l'Air Quality Index. Un esempio del calcolo AQI effettuato su php è riportato in figura 5.25 ed è relativo all'inquinante NO2:

```
if($r['Inquinante']=="NO2"){
if($avg<50){
$aqi=((25-0)/(50-0)*($avg-0))+0;
}else if($avg>=50 && $avg<100){$aqi=((50-25)/(100-50)*($avg-50))+25;
}else if($avg>=100 && $avg<200){$aqi=((75-50)/(200-100)*($avg-100))+50;
}else if($avg>=200 && $avg<400){$aqi=((100-75)/(400-200)*($avg-200))+75;
}else if($avg>400){$aqi=((125-100)/(600-400)*($avg-400))+100;}
```

Figura 5.25: Frammento di codice php per il calcolo dell'AQI

5.6 Sviluppo del framework

Dopo aver caricato i dati nel DB e definito quelli che possono essere i KPI di interesse, il passo successivo è stato quello di andare a sviluppare in maniera pratica questo progetto.

In particolare, è stata sviluppata un'applicazione Android, rivolta principalmente al cittadino. Tale applicazione si basa sull'indice AQI, poiché è stato ritenuto essere, tra i KPI definiti precedentemente, quello di maggiore impatto e di più facile lettura per il cittadino, interessato ad avere una visione veloce della qualità dell'aria in una zona di interesse o nella zona in cui esso si trova.

In linea generale, quindi, quest'app è rivolta al cittadino, il quale avrà la possibilità di monitorare, per ognuna delle stazioni di riferimento (P.zza Zavattari, P.zza Abbiategrasso, Parco Lambro, Via Verziere, Via Pascal, Via Senato, Viale Liguria, Viale Marche), il rispettivo valore di AQI, misurato sul giorno più recente inserito nel database. Inoltre potrà vedere nel dettaglio il valore di AQI di ogni singolo inquinante misurato dalla stazione scelta, potrà visualizzare i valori di AQI di tutte le stazioni (sempre relative al giorno più recente) e potrà monitorare i valori di AQI nei dieci giorni precedenti all'ultimo inserito nel database della stazione di riferimento.

Inoltre, quest'applicazione sarà in grado di rilevare la posizione corrente dell'utente che la usa, segnalare quella che è la stazione più vicina alla sua posizione, e quindi quella di maggior interesse per la posizione in cui si trova l'utente, e mostrarne i relativi valori di AQI.

Oltre alla creazione dell'applicazione, rivolta in particolar modo al cittadino, è stata anche sviluppata una sezione relativa all'indice AQI, integrata in una dashboard esistente, pensata prevalentemente per il personale dell'amministrazione locale, i quali rispetto al cittadino richiedono una visione più ampia di quello che è l'andamento della qualità dell'aria nella città, e che l'app non è in grado di fornire. Per questo è stata sviluppata una sezione sulla dashboard relativa all'indice AQI che mostri informazioni più complete e generali rispetto a quelle mostrate nell'app.

In particolar modo, la sezione aggiunta alla dashboard e chiamata "Air Quality Index" si compone di due sottosezioni: AQI REPORT e DAYS ABOVE THRESHOLD.

In linea di massima la prima, una volta selezionata una data, permette di visualizzare su mappa (in modo da avere un impatto visivo immediato e ad effetto) la posizione precisa delle varie stazioni con il relativo valore di AQI.

Inoltre, consente di visualizzare i valori di AQI di ogni singolo inquinante, sempre relativo alla data scelta, di tutte le stazioni di riferimento.

Infine questa sottosezione, dà anche la possibilità di monitorare l'andamento, sotto forma di grafico a linee, dei valori di AQI di tutte le stazioni, negli ultimi sessanta giorni rispetto al giorno selezionato.

La seconda sottosezione invece, è pensata per risolvere uno dei KPI precedentemente individuati, grazie al quale, in base all'intervallo di giorni specificato dall'utente e all'inquinante scelto (PM10, PM2.5, CO, NO2, SO2, O3), vengono mostrate su mappa le stazioni che misurano quell'inquinante, con la rispettiva percentuale di giorni in cui ogni stazione ha superato il livello di soglia inserito dall'utente.

In definitiva quindi, nello sviluppo si è cercato di creare un qualcosa che potesse essere di utilità sia per il cittadino interessato alla valutazione locale della qualità dell'aria in alcune specifiche aree urbane, ad esempio dove vivono o lavorano (per questo è stata realizzata l'applicazione Android), sia per il personale dell'amministrazione interessati a monitorare l'inquinamento atmosferico nell'area urbana per indicare quando e dove le concentrazioni di inquinanti sono diventate critiche (per questo è stata realizzata una sezione apposita di una dashboard).

Nei prossimi paragrafi saranno spiegati nel dettaglio sia l'applicazione Android sia questa sezione inserita nella Dashboard.

5.6.1 Applicazione Android

Come accennato precedentemente, l'applicazione Android "Monitoring Air Pollution" è stata creata prevalentemente per il cittadino, e quindi si focalizza in particolare sulle singole stazioni, piuttosto che sulla visione generale dell'intero comune di Milano, in quanto il cittadino è interessato principalmente ad avere informazioni su aree specifiche della città.

Per la creazione di quest'applicazione, è stato usato come ambiente di sviluppo Android Studio. Questo ambiente di sviluppo divide il progetto in tre parti principali:

- Una cartella con il codice Java e che conterrà tutta la logica usata per far funzionare l'app;
- Un file di configurazione AndroidManifest.xml che dà il nome al package Java dell'applicazione, determina i processi che ospitano i componenti dell'applicazione, elenca le librerie necessarie, dichiara il livello minimo di API Android che l'app richiede e dichiara i permessi delle app e i permessi necessari alle altre app per interagire con essa;
- Una cartella res contenente risorse per lo più realizzate in xml, e nella quale vengono prevalentemente definiti gli aspetti grafici dell'app.

Per quanto riguarda il server utilizzato dall'app per interfacciarsi con il DW spiegato nei paragrafi precedenti, è stata utilizzata la piattaforma web Altervista, la quale permette di caricare in remoto il database creato e di gestire le pagine php necessarie per estrarre i dati dal database. In particolare, nel DB creato su Altervista sono state inserite tutte le tabelle descritte precedentemente (Traffico, inquinanti, meteo, vistagiornaliera) in prospettiva di eventuali sviluppi futuri dell'applicazione, ma quella con cui l'app si è interfacciata è la tabella "vistagiornaliera" che contiene tutti i dati necessari per l'applicazione creata. In linea di massima, quest'app permette all'utente di visualizzare il valore di AQI dell'ultimo giorno disponibile nel DB di una stazione di riferimento scelta dall'utente, il valore di AQI di ogni singolo inquinante misurato dalla stazione scelta, il valore di AQI di tutte le stazioni di Milano e il valore di AQI della stazione scelta negli ultimi dieci giorni a partire dall'ultimo inserito nel DB. Inoltre, sarà in grado di capire qual è la posizione dell'utente, indicarne la stazione più vicina, e di essa mostrarne le cose dette precedentemente. Di seguito vengono analizzate nel dettaglio le singole schermate dell'applicazione "Monitoring Air Pollution" spiegandone anche la logica di programmazione usata per crearla. La prima schermata che l'utente visualizza quando usa per la prima volta l'applicazione è quella riportata in figura 5.26.

Figura 5.26: Schermata Richiesta permesso per accedere alla posizione del dispostivo

Quindi, la prima cosa che viene richiesta all'utente è quella di consentire all'app di accedere alla posizione del dispositivo, poiché come si vedrà in seguito per il corretto utilizzo di tutte le funzionalità dell'app è necessario conoscere la posizione corrente dell'utente.

A livello di codice Android, per realizzare questa schermata, è stato necessario far uso delle "Permission", ossia permessi speciali che è necessario dichiarare nel file AndroidManifest.xml. In questa app la permission richiesta è la seguente:

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

Per fare in modo che le nostre app funzionino su versioni Android con API 23 o superiori è tuttavia necessario verificare la disponibilità di una permission prima di utilizzare il metodo che ne ha bisogno. Per far comparire la finestra di dialogo che richieda l'accettazione della permission, è stato poi scritto del codice Java apposito nell'activity principale del programma ossia "MainActivity".

L'utente dopo avere accettato o negato il consenso all'accesso alla posizione del dispositivo sarà rimandato alla schermata iniziale dell'app riportata in figura 5.27:

Figura 5.27: Schermata Iniziale

E' una semplice schermata iniziale in cui viene mostrato all'utente quella che è la tematica su cui si basa l'app.

Scorrendo verso destra o cliccando sul simbolo del menù in alto a sinistra l'utente ha la possibilità di accedere al menù laterale composto dalle seguenti voci; My Position, P.zza Zavattari, P.zza Abbiatrgrasso, Parco Lambro, Via Verziere, Via Pascal, Via Senato, Viale Liguria e Viale Marche. Una rappresentazione più chiara è visualizzata in figura 5.28

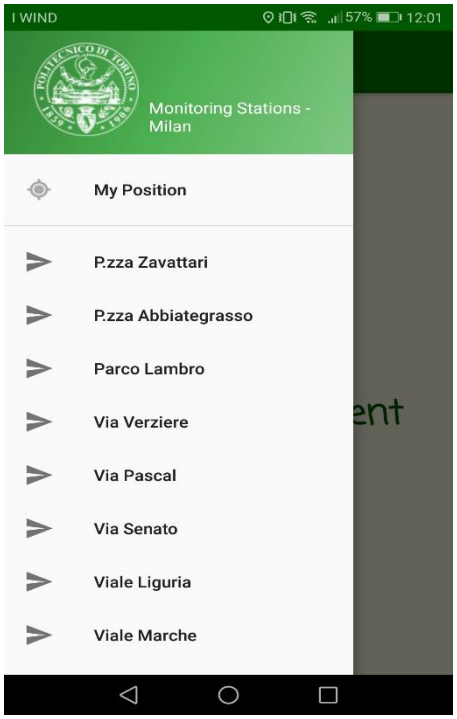


Figura 5.28: Schermata Menù

Graficamente, quindi, la scelta adottata è stata quella di creare un moderno menu laterale a scomparsa chiamato anche "Navigation Drawer", che a livello di codice sarà parte integrante dell'activity principale "MainActivity" creata in Android Studio. Come si può notare dalla figura 5.28, l'utente potrà cliccare su una delle voci del menu, che comprende My Position, e le varie stazioni di Milano. Cliccando su MyPosition gli scenari che si possono presentare all'utente sono due. Il primo è nel caso in cui l'utente non abbia attivato il GPS e/o abbia negato il consenso all'accesso alla posizione del dispositivo. In questo caso la schermata che visualizzerebbe è riportata in figura 5.29

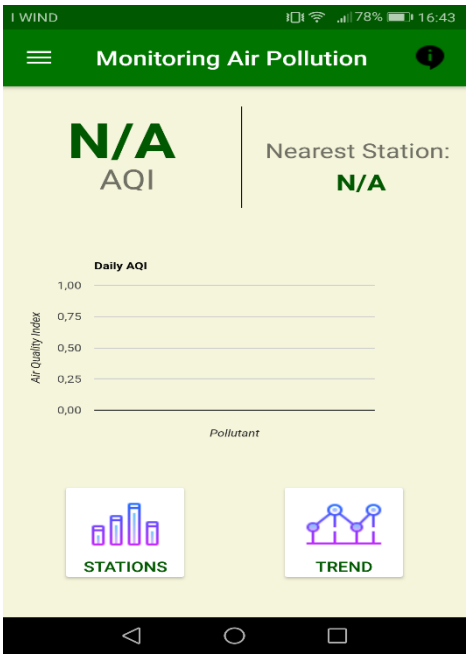


Figura 5.29: Schermata MyPosition in caso di GPS non attivato

Come si può notare, in questo caso l'utente visualizzerà una schermata in cui non sarà possibile avere alcun tipo di informazione poiché affinchè questa schermata si presenti con delle informazioni utili è necessaria l'attivazione del GPS;

Il secondo è nel caso in cui l'utente attiva il GPS e ha dato il suo consenso all'accesso alla posizione del dispositivo. In questa situazione l'utente si troverà dinanzi alla schermata di figura 5.30

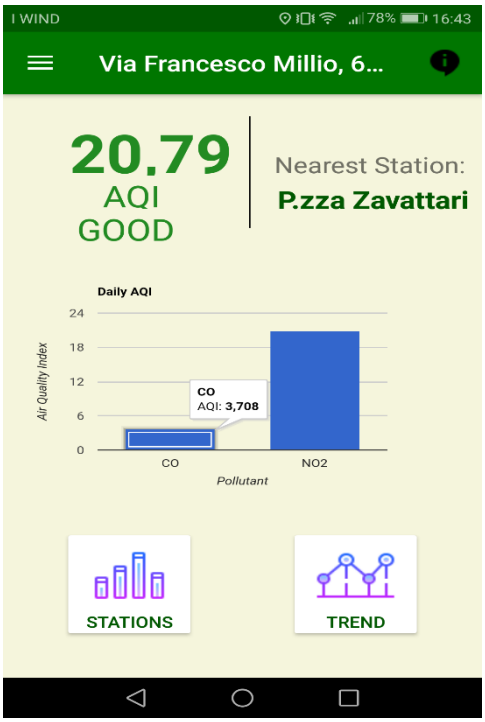


Figura 5.30: Schermata MyPosition in caso di GPS attivato

Si spiega ora nel dettaglio, sia a livello di codice che a livello di visualizzazione la schermata 5.30.

Si può notare come nella action bar dell'applicazione venga visualizzata una via. Nello specifico è riportata la via in cui è posizionato in tempo reale l'utente. Nell'esempio in questione "Via Francesco Millio".

A seconda della posizione in cui si trova l'utente, tramite la distanza euclidea misurata tra la latitudine e longitudine della posizione corrente e quella di ogni singola stazione, si vede qual è la stazione più vicina rispetto alla posizione dell'utente. Questa stazione viene indicata sotto la voce Nearest Station. In questo esempio la stazione più vicina a Via Francesco Millio risulta essere P.zza Zavattari.

Accanto alla voce Nearest Station verrà quindi mostrato il valore di AQI della data più recente presente nel database, calcolato dalla stazione di riferimento, in questo caso, per P.zza Zavattari pari a 20.79, con l'indicazione della fascia in cui ricade, in questo caso "GOOD". Sotto a queste informazioni è presente un grafico in cui è indicato il valore di AQI di tutti gli inquinanti rilevati dalla stazione presa in considerazione.

Viene ora spiegato nel dettaglio il codice scritto per visualizzare quanto detto. Nella classe MainActivity del codice, per prima cosa vengono individuate le coordinate (latitudine e longitudine) della posizione corrente dell'utente. Per ottenere tali informazioni è stato necessario far ricorso ad uno dei servizi messi a disposizione da Google, ossia i Location Services. Per l'uso dei Location Services i prerequisiti necessari richiesti sono: i Google Play

Services e la classe GoogleApiClient.

I Google Play Services fungono da intermediari necessari alle app Android per interagire con un servizio remoto Google e in particolare in Android Studio vanno scaricati dall'Android SDK Manager. Una volta scaricati, per integrarli nel programma sarà necessario inserire nel file build.gradle tra le dependencies la seguente istruzione:

compile 'com.google.android.gms:play-services:10.2.6'

La classe GoogleApiClient invece, permette un accesso uniforme a qualsiasi set di API offerte da Google. L'approccio da utilizzare per un servizio Google è quello di avere una fase di connessione (metodo Connect), seguita dalla vera interazione con il servizio, ed infine una fase di disconnessione (metodo Disconnect). I metodi di connessione e disconnessione sono stati opportunamente inseriti nell'activity principale creata, ossia "MainActivity".

Una volta implementati tali metodi, è stato necessario usare la classe LocationServices, al cui interno viene usato il membro API in fase di configurazione di GoogleApiClient e l'oggetto FusedLocationApi che fornisce il riferimento al provider di localizzazione da utilizzare. I metodi richiamati da questo oggetto, nell'applicazione creata, sono:

- requestLocationUpdates: permette di effettuare l'iscrizione al sistema di notifiche del servizio di localizzazione;
- removeLocationUpdates: permette di annullare l'iscrizione al sistema di notifiche del servizio di localizzazione;
- getLastLocation: che fornisce il migliore e più recente valore della posizione, basandosi sulle permission richieste dall'applicazione e dai diversi sensori di localizzazione attivi.

E' proprio da quest'ultimo metodo che si riesce quindi a ottenere le coordinate di latitudine e longitudine relative alla posizione corrente. L'istruzione utilizzata è la seguente:

mCurrentLocation=LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient)

Dopo aver ottenuto queste coordinate, l'obiettivo è trasformarle nel corrispondente indirizzo stradale. Per fare questo si è ricorsi all'operazione di Geocoding inverso, attuabile in Android mediante un oggetto di classe Geocoder, il cui utilizzo è basato sull'invocazione del metodo getFromLocation che richiede come parametri i valori di latitudine e longitudine. Gli indirizzi restituiti vengono poi collocati in una lista di oggetti Address. Il codice scritto per effettuare tale operazione è presente in figura 5.31:

```
double lat = mCurrentLocation.getLatitude();
double lng = mCurrentLocation.getLongitude();
try {
   Geocoder geocoder = new Geocoder(this, Locale.getDefault());
   List<Address> addresses = geocoder.getFromLocation(lat, lng, 1);
 via = addresses.get(0).getAddressLine(0);
} catch (IOException e) {
   e.printStackTrace();
}}
```

Figura 5.31: Frammento di codice per ottenere l'indirizzo stradale dalle coordinate

Ottenuta l'informazione sul nome della via in cui si trova l'utente, l'operazione successiva è stata quella di capire quale fosse la stazione più vicina rispetto alla posizione dell'utente. Come detto, è stata usata la distanza euclidea. In particolare, è stato creato un metodo chiamato

getNearest in cui i parametri passati sono stati la latitudine e la longitudine della posizione corrente. In questo metodo, per ogni stazione, usando una HashMap sono state salvate tutte le stazioni con le rispettive coordinate corrispondenti ad esse. A questo punto, con un banale algoritmo per il calcolo del minimo, è stato possibile calcolare la stazione più vicina alla posizione dell'utente. Questo metodo quindi, restituisce il nome della stazione più vicina.

```
nome_stazione=getNearest(mCurrentLocation.getLatitude(),
mCurrentLocation.getLongitude());
```

Queste due istruzioni mostrano il richiamo del metodo getNearest in cui si passano le coordinate della posizione dell'utente e come risultato restituisce il nome della stazione più vicina.

In seguito viene mostrata la figura 5.32 contenente il codice usato per calcolare la stazione con la distanza minima da quella attuale dell'utente.

```
float distancemin=0;
String staznearest= null;
Iterator it = stazioni.entrySet().iterator();
int i=0;
while(it.hasNext()) {
   Map.Entry entry = (Map.Entry)it.next();
  float d = mia.distanceTo((Location) entry.getValue());     if(i==0){
      distancemin = d;
      staznearest= (String) entry.getKey();
   }
   if(d<distancemin) {
      distancemin = d;
      staznearest= (String) entry.getKey();
   }
   i++;
}
return staznearest;
```

F**igura 5.32**: Frammento di codice per calcolare la stazione con distanza minima da quella dell'utente

Fatto questo, sono state calcolate due informazioni nell' Activity "MainActivity" che vengono visualizzate nella schermata di figura 5.30. Tuttavia, tale schermata contiene le informazioni calcolate in un fragment creato appositamente e chiamato "Position". Per cui, le informazioni relative al nome della stazione più vicina alla posizione corrente, e la posizione corrente stessa, verranno inviate a questo fragment e saranno utilizzabili da esso. Tale fragment viene richiamato solamente quando l'utente clicca sulla voce My Position del menu. Il codice che intercetta tale evento è in figura 5.33 ed è il seguente:

```
        if (id == R.id.nav_position) {
    Bundle args = new Bundle();
    args.putString("stazione",nome_stazione);
    args.putString("Via",via);
    Position p = new Position();
    p.setArguments(args);
  FragmentManager fragmentManager=getFragmentManager();
  fragmentManager.beginTransaction().replace(R.id.content_frame,p).commit();
  }
```

Figura 5.33: Codice relativo al passaggio dall'activity al fragment

Spiegandolo nel dettaglio, la if intercetta il click dell'utente sulla voce del menu My Position, si definisce un oggetto args di tipo Bundle in cui inserire i parametri da inviare al fragment (nome_stazione e via), si crea un oggetto di tipo Position in cui inserire i due parametri e si inizia la transazione con il fragment.

A questo punto, passando al fragment "Position", basterà recuperare i parametri inviati dal "MainActivity", con le seguenti istruzioni:

```
 name = getArguments().getString("stazione");
myvia = getArguments().getString("Via");
```

e stamparli. Infatti:

```
((AppCompatActivity)getActivity()).getSupportActionBar().setTitle(""+myvia);
```

è l'istruzione che setta sulla action Bar la via dell'utente, mentre:

```
ts = (TextView) getActivity().findViewById(R.id.text4);
if(name!=null) {
ts.setText(name);
```

sono le istruzioni con cui viene stampato a video la stazione più vicina.

A questo punto, si passa alla spiegazione del codice che ha portato al calcolo dell'AQI. Per prima cosa, è necessario dire che l'AQI calcolato si riferisce al giorno più vicino a quello odierno, inserito nel Database. Per reperire le informazioni sull'AQI è necessario creare un collegamento tra l'applicazione Android e il server remoto di Altervista, contenente la tabella da cui estrarre i dati che ci interessano per il calcolo dell'AQI.

In particolare, Android non può interagire direttamente con il DB remoto, ed è per questo che si collega tramite protocollo http ad un web server. Volendo generare pagine dinamiche, sul server viene inserita una pagina php che interagisce con il database, esegue l'interrogazione e ottiene il risultato dal database. Dai dati ottenuti, viene calcolato il valore di AQI in php, e tale valore viene poi trasportato in formato JSON all'applicazione Android.

Per prima cosa, per comunicare con un server remoto è necessario aggiungere nel manifest il seguente permesso:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Tale permesso consente di accedere a Internet da un'applicazione Android.

L'operazione di scaricamento dei dati da un database va eseguita in background. E' per questo che per gestire tali operazioni è stata usata la classe AsyncTask. Nello specifico è stata creata una classe BackGround che estende AsyncTask. Sono stati implementati due metodi all'interno di tale classe. Il primo è chiamato doInBackground ed è il metodo in cui vengono collocate tutte le operazioni "lente" ed è l'unico che viene eseguito su un thread secondario e che quindi non può aggiornare l'interfaccia utente.

Il secondo metodo invece è l'onPostExecute, il quale viene eseguito sul thread principale ed è richiamato automaticamente alla fine di doInBackground e svolge operazioni collegate all'interfaccia utente. E' qui che arrivano i risultati del doInBackground.

Quindi, all'interno del codice scritto, dopo aver inserito il permesso per accedere ad Internet, è stata estesa la classe AsyncTask ed è stato implementato il metodo doInBackground. L'operazione fondamentale contenuta in tale metodo è quella di creare la connessione al web server e alla pagina php con cui interagire, definendo il parametro da inviare ad essa. Nel caso in questione il parametro inviato è quello relativo al nome della stazione di cui si vuole calcolare l'indice AQI. In figura 5.34 è riportato un frammento di codice che fa questo:

```
protected String doInBackground(String... params) {
String station = params[0];
int tmp;
try {
URL url = newURL("http://monitoringairpollution.altervista.org/aqi.php");
String urlParams = "stat=" + station;
HttpURLConnection httpURLConnection = (HttpURLConnection)
url.openConnection();
httpURLConnection.setDoOutput(true);
```

Figura 5.34: Codice relativo alla connessione al db

Si passa ad esaminare nel dettaglio la logica usata nel file php per calcolare l'AQI della stazione passata come parametro dal programma Android. Per prima cosa, viene recuperato il parametro inviato, quindi il nome della stazione che ci interessa, e viene fatta la seguente query:

```
SELECT `Inquinante`,`Media_Giornaliera`
FROM `vistagiornaliera`
WHERE `Nome_Stazione`='".$station."' AND `Data`=(SELECT MAX(`Data`)
                                                FROM `vistagiornaliera` )
```

Questa query permette di estrarre dalla tabella vistagiornaliera, i valori degli inquinanti misurati dalla stazione di riferimento dell'ultimo giorno disponibile.

A questo punto per ogni riga del resultset della query, viene chiesto qual è l'inquinante in questione, e in base a questo valore, viene calcolato opportunamente l'AQI rispettivo. Un frammento di codice che mostra questo è in figura 5.35:

```
while($row = mysqli_fetch_array($result)){
if($row[0]=="NO2"){
$aqi=0;
$avg=$row[1];
if($avg<50){
$aqi=((25-0)/(50-0)*($avg-0))+0;
```

```
        }else if($avg>=50 && $avg<100){$aqi=((50-25)/(100-50)*($avg-
50))+25;
        }else if($avg>=100 && $avg<200){$aqi=((75-50)/(200-100)*($avg-
100))+50;
        }else if($avg>=200 && $avg<400){$aqi=((100-75)/(400-200)*($avg-
200))+75;
        }else if($avg>400){$aqi=((125-100)/(600-400)*($avg-400))+100;
        }}
        if($row[0]=="CO"){
        $aqi=0;
        $avg=$row[1];
        if($avg<5){
        $aqi=((25-0)/(5-0)*($avg-0))+0;
        }else if($avg>=5 && $avg<7.5){$aqi=((50-25)/(7.5-5)*($avg-5))+25;
        }else if($avg>=7.5 && $avg<10){$aqi=((75-50)/(10-7.5)*($avg-7.5))+50;
 }else if($avg>=10 && $avg<20){$aqi=((100-75)/(20-10)*($avg-10))+75;
        }else if($avg>20){$aqi=((125-100)/(30-20)*($avg-20))+100;}}
```

Figura 5.35: Codice relativo al calcolo dell'AQI

Si può notare come a seconda del tipo di inquinate, per il calcolo dell'AQI vengono usati parametri diversi.

Ad ogni ciclo che scandisce la tabella del risultato ottenuto dalla query, viene inserito in un array il nome dell'inquinante di cui si è calcolato l'AQI e il rispettivo valore di AQI, con la seguente istruzione:

```
$response[] = array("inquinante"=>$row[0],"aqi"=>$aqi);
```

A questo punto, una volta riempito il vettore "$response", esso viene inviato in formato JSON al programma Android. Termina così l'operazione di doInBackground e viene automaticamente richiamato il metodo onPostExecute. In questo metodo, viene recuperato il vettore in formato JSON proveniente dal php, e vengono inserite in due liste diverse i risultati. In una lista, chiamata "inq", ci saranno tutte le stringhe contenenti il nome degli inquinanti, e nell'altra chiamata "aqi" ci saranno tutti i rispettivi valori di AQI. In figura 5.36 è riportato il codice che fa questo:

```
JSONObject jObj = new JSONObject(s);
JSONArray jr =jObj.optJSONArray("Aqidb");
if(jr!=null){
 for (int i=0;i<jr.length();i++){
    JSONObject jo=jr.getJSONObject(i);
    inq[i]=jo.getString("inquinante");
    aqi[i]=jo.getDouble("aqi");
    lista.add(aqi[i]);
 }
```

Figura 5.36: Trasformazione da JSON a codice Java

Per visualizzare il valore di AQI maggiore tra quelli degli inquinanti della stazione di riferimento, basterà estrarre il massimo dei valori presenti nella lista aqi e stamparlo a video.

```
maxaqi = Collections.max(lista);
t1.setText(String.format("%.2f",maxaqi));
```

Dopo questo, viene ora spiegato come è stato possibile creare il grafico presente in figura 5.30 relativo ai valori di AQI degli inquinanti della stazione più vicina alla posizione degli utenti. Per realizzare questo grafico a barre in Android si è ricorsi all'uso di un'API Google che sono i Google Chart.

Questo è un servizio che permette di creare dei grafici online. Il grande vantaggio è proprio quello di avere uno strumento molto potente senza dover ricorrere ad altre librerie apposite per i grafici. Google Chart dà la possibilità di costruire diversi tipi di grafici, scegliere i colori degli elementi e dello sfondo e le rispettive dimensioni. Ciò che viene prodotto in output è una URL contenente l'immagine del grafico creato. Il grafico ottenuto con i Google chart è contenuto nel widget Android "webView". Tale widget permette di integrare una web application (in questo caso le Google Chart) nella propria applicazione.

Per l'uso dei Google Chart, come per l'uso di gran parte delle cose che si interfacciano con il web, fondamentale risulta essere l'uso di JavaScript, che viene integrato quindi nel codice dell'applicazione Android creata. Di default, l'uso di JavaScript in una webView non è attivo. Per attivarlo è stato necessario inserire le seguenti righe di codice di figura 5.37:

```
WebViewwebView=(WebView)getActivity().findViewById(R.id.web);
webView.addJavascriptInterface(new WebAppInterface(), "Android");
webView.setBackgroundColor(TRANSPARENT);
webView.getSettings().setJavaScriptEnabled(true);
```

Figura 5.37: Codice per attivareJavaScript in una webView

Il widget webView è molto potente in quanto permette anche a JavaScript di interagire con il codice Java presente nell'applicazione. Per far sì che questo accada è necessario che al codice Java che si vuole richiamare in JavaScript si apponga la notazione *@JavascriptInterface* sui metodi che devono essere visibili "da Javascript". In figura 5.38 è riportato il codice Android scritto per l'applicazione creata che si occupa di quanto appena detto:

```
   public class WebAppInterface {
@JavascriptInterface
 public int lunghezza(){
   return lista.size();
  }
  @JavascriptInterface
public String getInquinanti(int i){
  return inq[i];
}
@JavascriptInterface
public String getValori(int i){
  return ""+aqi[i];
}
}
```

Figura 5.38: Codice per richiamare codice Java in JavaScript

Come si può notare sono stati creati tre metodi in codice Java che saranno poi usati dal codice JavaScript per creare il grafico. Questi tre metodi sono:

- "lunghezza" che restituisce il numero di inquinanti misurati dalla stazione. Nel nostro esempio sono 2.
- "getInquinanti" che restituisce le stringhe presenti nella lista "inq" descritta in precedenza, e nel nostro esempio sono 2 ossia "CO" ed "NO2";
- "getValori" che restituisce i valori di AQI calcolati dalla pagina php sopra descritta e che nel nostro esempio contiene appunto i valori di AQI di "CO" ed "NO2", rispettivamente 3.7 e 20.79.

A questo punto, si è passati alla scrittura del codice JavaScript utile per la creazione del grafico in questione. In particolare, è stata creata la funzione drawChart in cui un frammento di codice è visibile in figura 5.39:

```
var data = new google.visualization.DataTable();
data.addColumn('string', 'Pollutant');
data.addColumn('number', 'AQI');
for(i=0;i<Android.lunghezza();i++)
data.addRow([Android.getInquinanti(i),parseFloat(Android.getValori(i))]);
```

Figura 5.39: Codice Javascript per disegnare il grafico

Con addColumn vengono impostati quelli che sono i valori delle ascisse e delle ordinate del grafico. Nell'esempio in questione sulle ascisse ci sono gli inquinanti presenti, e sulle ordinate i rispettivi valori di AQI. Dopodichè viene inserito un ciclo for che si ripeterà un numero di volte pari al numero di inquinanti presenti (questa informazione viene reperita dal metodo lunghezza() scritto in Java e spiegato in precedenza) e per ogni ciclo viene aggiunto il grafico a barre relativo all'inquinante presente (informazione reperita dal metodo getInquinanti presente in Java) con il rispettivo valore (informazione reperita dal metodo getValori presente in Java).

Sempre in JavaScript sono state poi inserite tutte le impostazioni grafiche del grafico visualizzato.

Infine, una volta che tutto questo è stato completato, la webView è pronta per essere visualizzata e lo fa, come detto in precedenza, mostrando una URL contenente l'immagine del grafico creato.

```
webView.loadUrl("file:///android_asset/chart1.html");
```

In figura 5.30 in basso ci sono 2 bottoni che permettono all'utente di accedere a due sezioni diverse dell'app. L'utente cliccando il bottone in basso a sinistra "Stations", accederà ad una nuova schermata che mostrerà il valore di AQI, calcolato sempre sull'ultimo giorno disponibile nel database, di tutte le stazioni presenti a Milano, in modo tale da farsi un'idea più generale e immediata su quella che è la situazione della qualità dell'aria in tutto il comune e non su una singola zona.

In figura 5.40 è riportata la schermata che mostra questo.

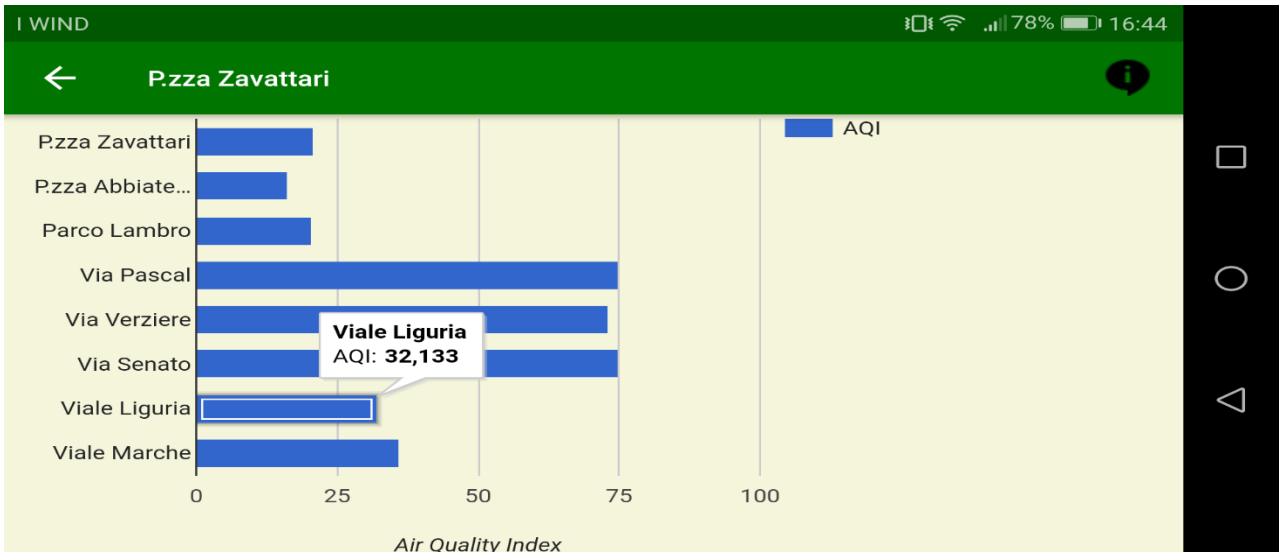


Figura 5.40: Schermata Stations

Come si può notare si tratta di un grafico a barre in cui i due assi contengono le informazioni sulle stazioni presenti e i rispettivi valori di AQI. Nell'esempio si nota come cliccando su una delle barre presenti, in questo caso su "Viale Liguria", l'utente possa visualizzare il valore preciso di AQI.

Per realizzare questa schermata, nel codice scritto è stato necessario creare una nuova activity "confronto" che viene richiamata ogni qualvolta l'utente clicca il bottone stations. In questa activity la prima operazione che è stata effettuata è stata quella di connettersi al server in remoto Altervista e quindi di interagire con una pagina php, che interroga la base dati, estrapola i risultati e li invia all'applicazione Android sotto forma di risposta JSON.

Il processo di connessione al server web rimane lo stesso di quello descritto in precedenza, per cui si passa ora ad esaminare direttamente la pagina php creata che permette di ricevere le informazioni che ci interessano.

La query presente in questa pagina è la seguente:

```
SELECT Nome_Stazione,Inquinante,Media_Giornaliera
FROM vistagiornaliera
WHERE Data=(SELECT MAX(`Data`)
            FROM `vistagiornaliera` )
```

Semplicemente questa query, estrae dalla tabella vistagiornaliera, tutti i valori giornalieri di tutti gli inquinanti di ogni singola stazione corrispondenti all'ultimo giorno disponibile nel database.

A questo punto, per ogni inquinante di ogni stazione di monitoraggio verrà calcolato l'AQI e il valore più alto tra quelli della medesima stazione rappresenterà l'AQI di riferimento della stazione stessa. In particolare, la logica usata in PHP per l'estrazione di tale KPI è la seguente:

1. Creazione di un vettore per ogni stazione di monitoraggio
2. Creazione di un vettore per i valori di AQI di riferimento di ogni stazione
3. Identificazione della stazione di monitoraggio
4. Identificazione dell'inquinante della stazione identificata
5. Calcolo dell'AQI per l'inquinante identificato
6. Salvataggio dell'AQI, calcolato al punto 5, nel vettore della stazione identificata
7. Ripetizione dei punti 4-5-6 per tutti gli inquinanti della stazione identificata

8. Inserimento nel vettore creato al punto 2 del valore massimo di AQI presente nel vettore della stazione identificata
9. Ripetizione dal punto 3 al punto 8, per ogni stazione

In figura 5.41 è riportato una parte del codice php riguardante quanto detto rispetto ad una singola stazione di monitoraggio:

```
    while($r = mysqli_fetch_array($result)){
        $avg=$r['Media_Giornaliera'];
       if($r['Nome_Stazione']=="P.zza Zavattari")
       {
     $array_tit[0] = $r['Nome_Stazione'];
       if($r['Inquinante']=="NO2"){
       $aqi=0;
if($avg<50){
       $aqi=((25-0)/(50-0)*($avg-0))+0;
}else if($avg>=50 && $avg<100){$aqi=((50-25)/(100-50)*($avg-50))+25;
        }else if($avg>=100 && $avg<200){$aqi=((75-50)/(200-100)*($ avg-100))+50;
        }else if($avg>=200 && $avg<400){$aqi=((100-75)/(400-200)*($avg-200))+75;
    }else if($avg>400){$aqi=((125-100)/(600-400)*($avg-400))+100;
    }
        $aqizav['NO2']=$aqi;
}else if($r['Inquinante']=="CO"){$aqi=0;
        if($avg<5){
 $aqi=((25-0)/(5-0)*($avg-0))+0;
          }else if($avg>=5 && $avg<7.5){$aqi=((50-25)/(7.5-5)*($avg-5))+25;
          }else if($avg>=7.5 && $avg<10){$aqi=((75-50)/(10-7.5)*($avg-7.5))+50;
      }else if($avg>=10 && $avg<20){$aqi=((100-75)/(20-10)*($avg-10))+75;
          }else if($avg>20){$aqi=((125-100)/(30-20)*($avg-20))+100;
       }
          $aqizav['CO']=$aqi;
       }
          $array_aqiday[0]=max($aqizav);
```

Figura 5.41: Codice AQI

Il vettore $array_aqiday conterrà tutti i valori massimi di AQI di ogni singola stazione. Ad ogni ciclo che scandisce la tabella risultante dalla query, viene inserito in un vettore, la stazione di riferimento e il valore di AQI corrispondente. Questo avviene mediante la seguente istruzione:

```
$response[] = array("stazioni"=>$array_tit[$i],"aqi"=>$array_aqiday[$i]);
```

$response sarà quindi il vettore che tornerà tramite risposta JSON al programma Android.

A questo punto, la risposta JSON è stata convertita in Android. Si visualizzi il codice di figura 5.42:

```
    JSONObject jObj = new JSONObject(s);
    JSONArray jr = jObj.optJSONArray("Aqistadb");
if(jr!=null){
   for (int i=0;i<jr.length();i++){
        JSONObject jo=jr.getJSONObject(i);
        sta[i]=jo.getString("stazioni");
        aqi[i]=jo.getDouble("aqi");
         lista.add(aqi[i]);
     }
```

Figura 5.42: Codice per convertire da formato JSON a Java

Sono state quindi create due liste, “sta” che contiene tutte le stazioni, e “aqi” che contiene invece i rispettivi valori di AQI. Queste due liste vengono create per essere poi inserite nei metodi “getStazioni” e “getValori” che sono due metodi Java richiamati da JavaScript (nello stesso modo descritto precedentemente) per creare il grafico che viene visualizzato in figura 5.40. In questo caso il codice JavaScript per la realizzazione del grafico sarà molto simile al codice del grafico spiegato precedentemente, con la differenza che al posto degli inquinanti verranno prese in considerazione le stazioni.

Oltre al bottone “Stations”, in figura 5.30 in basso a destra è possibile cliccare il bottone “Trend”. L’utente cliccando tale bottone accede ad una nuova schermata in cui viene mostrato un grafico a linee contenente il valore di AQI degli ultimi dieci giorni a partire dall’ultimo presente nel database, chiaramente riferito alla stazione presa in considerazione.

In figura 5.43 è riportata tale schermata.

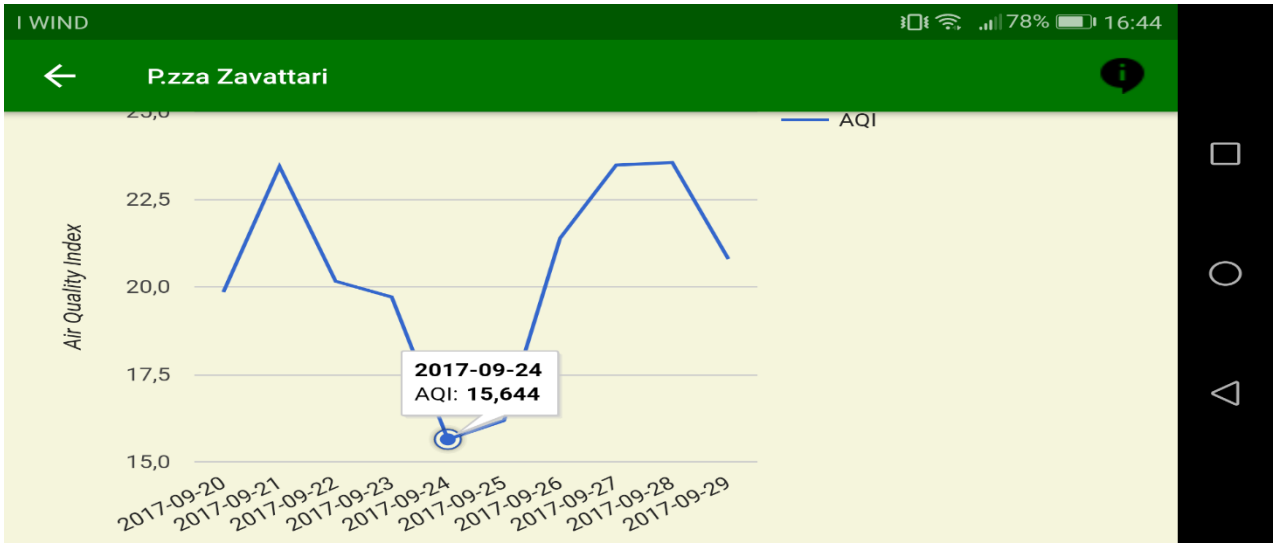


Figura 5.43: Schermata AQI 10 giorni precedenti

Si tratta quindi di un grafico a linee in cui sui due assi sono presenti la data di riferimento e il rispettivo valore di AQI. Nell’esempio si nota come cliccando su uno dei punti di queste linee, l’utente possa visualizzare il valore preciso di AQI del giorno in questione.

Per realizzare questa schermata, nel codice scritto è stato necessario creare una nuova activity chiamata “diecigg” che chiaramente viene richiamata ogni qualvolta viene cliccato il bottone trend.

Anche in questa activity, come quella descritta in precedenza, la prima operazione che è stata effettuata è stata quella di connettersi al server in remoto Altervista e quindi di interagire con una pagina php, che interroga la base dati, estrapola i risultati e li invia all'applicazione Android sotto forma di risposta JSON.

Chiaramente, anche in questo caso la connessione al server web è stata fatta nelle stesse modalità descritte precedentemente per cui si passa direttamente alla spiegazione della realizzazione della pagina php.

La prima operazione effettuata in questa pagina php è stata quella di recuperare, tramite il metodo POST, il parametro inviato dal programma Android relativo alla stazione di riferimento poiché sarà su di essa che si fonderà il vincolo della query.

```
$station=$_POST["stat"];
```

In seguito viene quindi riportata la query scritta per estrarre dalla tabella "vistagiornaliera" le informazioni necessarie per realizzare la schermata di figura 5.43.

```
SELECT  `Data`,`Inquinante`,`Media_Giornaliera`
FROM `vistagiornaliera`
WHERE `Nome_Stazione`='".$station."'
        AND `Data` BETWEEN adddate((SELECT MAX(`Data`)
                                    FROM `vistagiornaliera`),-9)
        AND (SELECT MAX(`Data`)
              FROM `vistagiornaliera`)
GROUP BY `Data`,`Inquinante`,`Media_Giornaliera`
```

La seguente query non fa altro che restituire la data, l'inquinante, e il rispettivo valore, dalla tabella vistagiornaliera, della stazione di riferimento, e nelle date che sono comprese tra l'ultimo giorno disponibile e i 9 giorni precedenti ad esso.

In php, ottenuta la tabella risultante da questa query, si è proceduto al calcolo dell'AQI per ogni data, quindi in questo caso è stato fatto un controllo sul tipo di inquinante e in base ad esso sono stati effettuati i rispettivi calcoli estraendone i valori di AQI. L'AQI maggiore relativo ad un singolo giorno è stato poi inserito in un vettore che conterrà tutti gli AQI di riferimento di ogni giorno. In seguito, questo vettore e il vettore contenente le date vengono inserite in un unico vettore che sarà quello che poi sarà trasmesso al programma Android in formato JSON. Questo avviene mediante la seguente istruzione:

```
for($n=0;$n<10;$n++){
 $response[] = array("data"=>$dat[$n],"aqi"=>$maxstaz[$n]);
}
```

$maxstaz è appunto il vettore che contiene tutti gli aqi dei vari giorni.

Il programma Android nel metodo onPostExecute della classe che estende AsyncTask, riceve la risposta Json, e la converte in due liste che vengono utilizzate in seguito, chiamate "dat" e "aqi". Il codice è quello di figura 5.44:

```
JSONObject jObj = new JSONObject(s);
        JSONArray jr = jObj.optJSONArray("Aqi10db");
        if(jr!=null){
        for (int i=0;i<jr.length();i++){
JSONObject jo=jr.getJSONObject(i);
dat[i]=jo.getString("data")
aqi[i]=jo.getDouble("aqi");
lista.add(aqi[i]);
```

Figura 5.44: Codice per convertire da JSON a Java

Anche queste due liste, sono state create per essere poi inserite nei metodi "getData" e "getValori" che sono due metodi Java richiamati da JavaScript (nello stesso modo descritto precedentemente) per creare il grafico che viene visualizzato in figura 5.43. In questo caso il codice JavaScript per la realizzazione del grafico sarà molto simile al codice dei grafici spiegati precedentemente, con la differenza che al posto degli inquinanti e delle stazioni verranno prese in considerazione le date e i valori di AQI.

Ritornando alla figura 5.28 che rappresenta la schermata del menù si può vedere come oltre alla voce My Position, che è stata appena spiegata, l'utente ha anche la possibilità di accedere singolarmente alle informazioni riguardanti tutte le stazioni presenti a Milano. L'utente cliccando su una delle stazioni presenti nel menù verra reindirizzato alla schermata di figura 5.45.

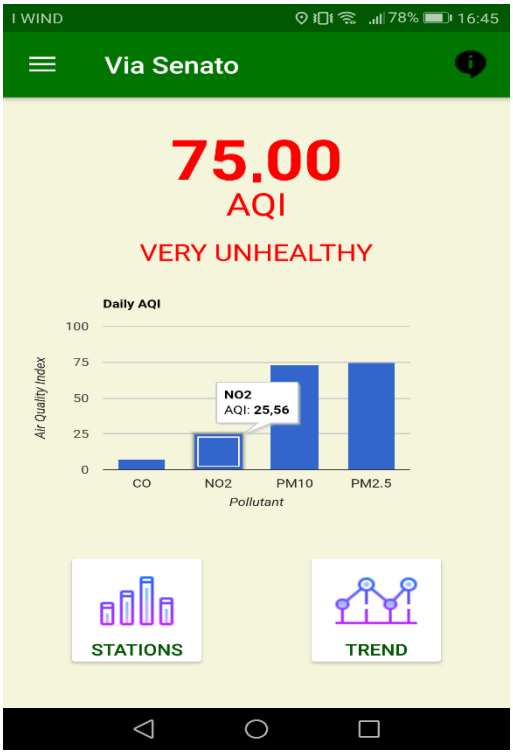


Figura 5.45: Schermata mostrata nel caso l'utente abbia cliccato su Via Senato tra le voci del menu

La schermata che viene presentata all'utente è simile a quella di My Position, con le uniche differenze che nella action Bar questa volta non viene mostrata la posizione corrente dell'utente ma il nome della stazione selezionata, e che a differenza della schermata di figura 5.30, viene

mostrato direttamente il valore di AQI calcolato rispetto alla stazione scelta, per cui non apparirà la voce Nearest Station in quanto inutile in questo caso.

Per il resto le visualizzazioni sono uguali a quelle precedenti, quindi anche qui c'è il grafico a barre in cui è presente ogni inquinante della stazione con il rispettivo codice di AQI. E sono presenti i due bottoni "Stations" e "Trend" che cliccandoli daranno accesso alle stesse schermate presenti rispettivamente in figura 5.40 e 5.43, ovviamente relative alla stazione scelta.

Dal punto di vista del codice scritto, ogni qual volta l'utente clicca su una delle stazioni del menu viene richiamato un fragment che è stato chiamato "Station". Il codice che intercetta tale evento è presente in figura 5.46 e segna il passaggio dall'Activity "MainActivity" al fragment "Station" è il seguente:

```
else if (id == R.id.nav_sen) {
            String name_station="Via Senato";
            Bundle args = new Bundle();
            args.putString("stazione",name_station);
            Station s = new Station();
            s.setArguments(args);
            FragmentManager fragmentManager=getFragmentManager();

fragmentManager.beginTransaction().replace(R.id.content_frame,s).commit();
          }
```

Figura 5.46: codice che intercetta il passaggio da MainActiviy a Station

Spiegandolo nel dettaglio, la if intercetta il click dell'utente sulla voce del menu della stazione, nell'esempio mostrato su suppone che l'utente abbia cliccato su "via Senato", si definisce un oggetto args di tipo Bundle in cui inserire i parametri da inviare al fragment (nome_stazione), si crea un oggetto di tipo Station in cui inserire i due parametri e si inizia la transazione con il fragment.

A questo punto, passando al fragment "Station", basterà recuperare i parametri inviati dal "MainActivity" e si procederà alla stampa a video di tutte le informazioni presenti nella figura 5.45. Questo avverrà in maniera molto simile a quanto spiegato in precedenza per la schermata relativa a My Position per cui è inutile ripetere nuovamente la spiegazione di ogni singola visualizzazione.

5.6.2 Integrazione Dashboard Air Quality Index

Come spiegato nell'introduzione, oltra all'applicazione Android, pensata soprattutto per il cittadino, è stata anche sviluppata una sezione integrata in una dashboard esistente, pensata per un diverso tipo di utente e in particolare per il personale dell'amministrazione più interessato ad avere una visione globale della situazione in tutta la città.

Questa sezione è stata integrata in una dashboard esistente e nel menù di essa compare con la voce "Air Quality Index" dal quale si possono selezionare due sottosezioni diverse che permetteranno di accedere a due pagine diverse della dashboard. Queste due sottosezioni sono: "AQI Report" e "Days Above Threshold".

In figura 5.47 si riporta una immagine di come appare il menù relativo a questa sezione nella dashboard.

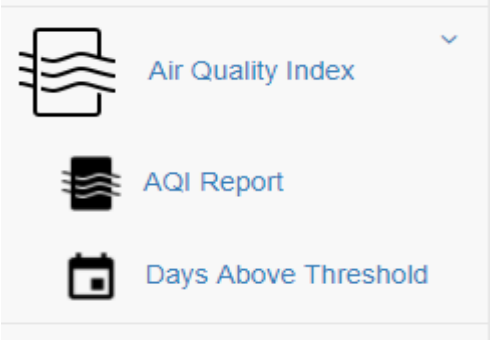


Figura 5.47: Schermata che mostra la sezione relativa all'AQI nel menu della dashboard

Per far sì che una dashboard sia consultabile dagli utenti, essa deve essere raggiungibile tramite qualsiasi browser. Per cui, affinchè il browser possa mostrare la dashboard e possa permettere l'interazione all'utente, devono essere utilizzati quei linguaggi di programmazione nati per lo sviluppo di applicazioni web.

Poiché le pagine create sono pagine dinamiche che richiedono l'interazione dell'utente con la dashboard e permettono di visualizzare dati diversi in base alle scelte di esso, per realizzare le due sottovoci elencate precedentemente, sono state realizzate due pagine php. Ovviamente in esse sono integrate, come in tutte le applicazioni web, delle istruzioni di HTML.

L'HTML è un linguaggio di formattazione che descrive le modalità di impaginazione o visualizzazione grafica (layout) del contenuto, testuale e non, di una pagina web attraverso tag di formattazione. Esso non prevede la definizione di variabili e strutture dati, ma il suo codice è in grado soltanto di strutturare e decorare dati testuali. Il linguaggio HTML ha come scopo quello di gestire i contenuti associandone o specificandone allo stesso tempo la struttura grafica (layout) all'interno della pagina web da realizzare grazie all'utilizzo di tag diversi.

All'interno di queste due pagine php create, quindi oltre ai tag HTML sono stati aggiunti anche i collegamenti a fogli di stile esterni contrassegnati dall'estensione .css tramite il tag link. Per cui, per curare la formattazione e gli aspetti grafici delle pagine realizzate si è fatto anche ricorso al CSS che è un linguaggio usato per definire la formattazione di documenti contenenti del codice HTML. Viene ora spiegato nel dettaglio come è stata realizzata la pagina a cui si accede cliccando la voce "AQI Report". Questa pagina permette di visualizzare tre informazioni diverse, in base alla data che verrà selezionata dall'utente. Agli occhi dell'utente questa possibile scelta della data viene mostrata in figura 5.48

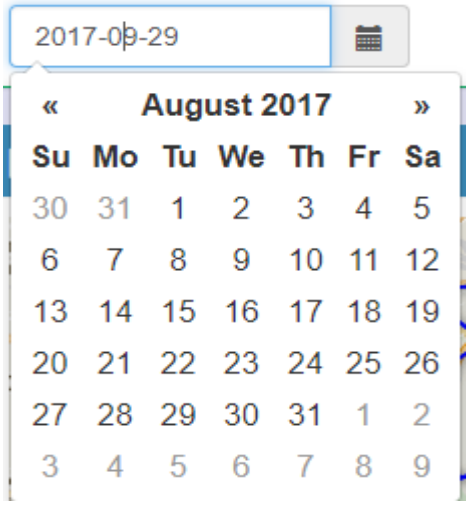


Figura 5.48: Schermata che mostra la scelta della data da parte dell'utente su cui visualizzare le informazioni

Per realizzare questa modalità di selezione della data per l'utente si è fatto ricorso ad un widget DatePicker nello stile Bootstrap. Data la diversità di utilizzo e la necessità di adattamento a vari formati di data (sia per sintassi che per lingua), datepicker offre un nutrito numero di opzioni che spaziano dall'impostazione del formato restituito, al restringmento delle date selezionabili a determinati range. Il tutto concorre ad ottimizzare l'esperienza utente e la gestione dell'input nella fase di rielaborazione dei dati. Di seguito è riportata la figura 5.49 con un frammento di codice relativo alla gestione del widget DatePicker:

```
        <script type="text/javascript">
        var end= '2017-09-29'
        var start= '2015-01-01'
        $('.input-group.date').datepicker({
format: "yyyy-mm-dd",
 autoclose: true,
            defaultViewDate: { year: 2017, month: 09, day: 29 },
    endDate: end,
            startDate: start
        });
        </script>
```

Figura 5.49: Codice di gestione del DatePicker

Come si può notare, con le variabili end e start si indicano la data iniziale e quella finale entro il quale l'utente può scegliere la data da selezionare. Una data al di fuori di questo range non sarà cliccabile dall'utente, e queste date sono proprio le date in cui sono disponibili i dati.

Dopo aver scelto la data su cui voler effettuare l'analisi, la dashboard mostrerà tre differenti visualizzazioni.

La prima è riportata in figura 5.50

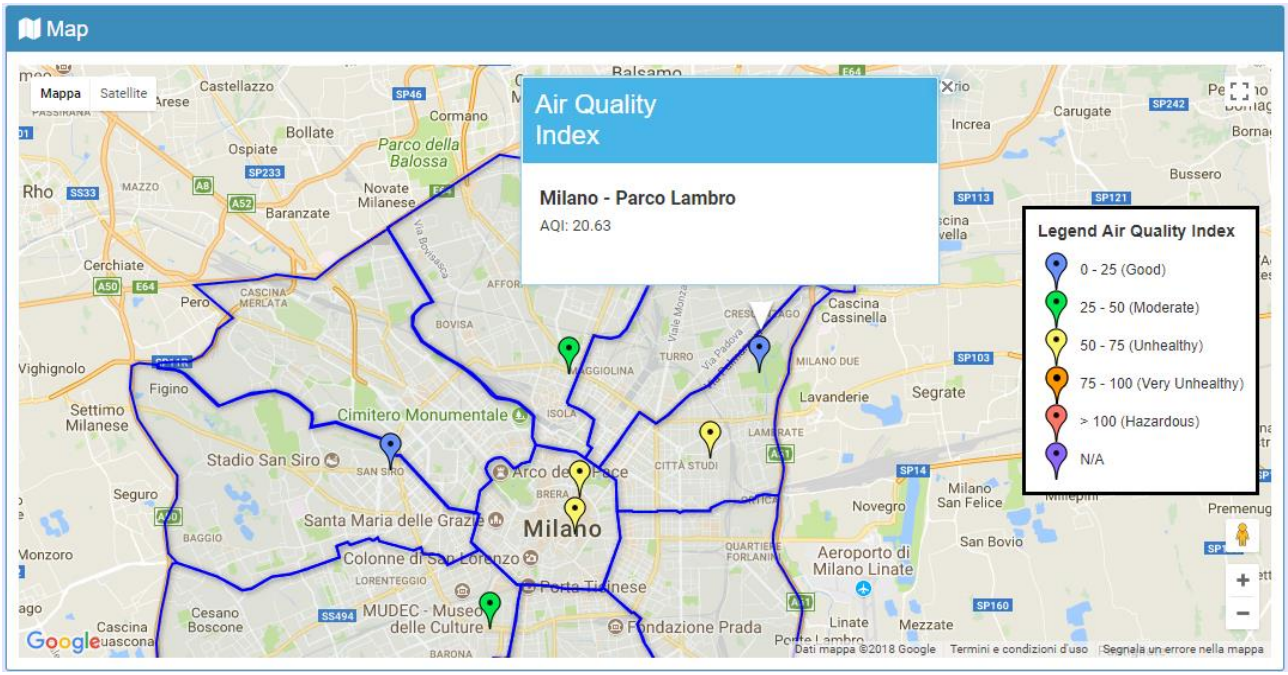


Figura 5.50: Schermata che mostra la mappa contenete le varie stazioni di Milano con il rispettivo valore di AQI

L'utente quindi, dopo aver scelto la data di interesse, visualizzerà la mappa della città di Milano in cui sono presenti 8 marker, ognuno nella posizione esatta in cui si trova realmente la stazione che effettua le misurazioni degli inquinanti. Quindi ogni marker rappresenta una stazione.

Questi marker avranno un colorazione diversa a seconda del valore di AQI ad essa associato. Ad esempio se una stazione ha un valore di AQI pari a 53, il relativo marker sarà colorato di giallo poiché ricade nell'intervallo 50-75, come riportato nella legenda sulla destra della mappa. Per cui, si avrà una percezione immediata su quella che è la situazione riguardo la qualità dell'aria in tutta Milano. Inoltre, cliccando su uno di questi marker, sarà possibile vedere il valore esatto dell'AQI ad esso corrispondente oltre che il nome della stazione associata.

Questa mappa è stata realizzata con l'ausilio delle API di Google relative a Google Maps. Affinchè si possa integrare in php questo servizio, è necessario possedere una API key e inserirla nella pagina php creata nel seguente modo:

```
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAwYar8NUmz1h6G0b4PhdMoRg
4yjkRS5s&sensor=false">
</script>
```

In seguito bisogna aggiungere un secondo tag script, contenete una funzione JavaScript che viene lanciata al caricamento della pagina HTML e che genera la mappa. In questo script sono contenute tutte le caratteristiche che deve avere la mappa, quindi ad esempio dove centrarla, la grandezza, lo zoom ecc. Un estratto di codice è in figura 5.51 che fa questo è il seguente:

```
var mapOptions = {
 zoom: 50,
 center: new google.maps.LatLng(41.896655,12.495918), //dove centro la mappa
 zoomControl: true,
 zoomControlOptions: {
 style: google.maps.ZoomControlStyle.LARGE
  },
   mapTypeId: google.maps.MapTypeId.ROADMAP
 };
```

Figura 5.51: Codice per l'impostazione grafica della mappa

In essa è anche presente una funzione adibita al caricamento dei singoli marker nelle posizioni esatte in base alle informazioni di latitudine e longitudine provenienti dal php. Questi marker assumono colori diversi a seconda del range in cui ricadono. Anche questi confronti, per stabilire in quale intervallo ricade l'AQI della stazione e quindi come colorarlo, vengono fatti nello script sopra citato, con i valori di AQI che provengono dal codice php, così come le informazioni di latitudine e longitudine. Per estrarre queste informazioni si è usata la seguente query:

```
SELECT  Lat, Lng, Nome_Stazione, Inquinante, Media_Giornaliera
FROM vistagiornaliera
WHERE Data ='".$giorno."'
```

Dalla query vengono estratte le informazioni relative a latitudine e longitudine (che servono per posizionare i marker nelle posizioni esatte in cui si trovano le stazioni), il nome della

stazione a cui corrispondono tali coordinate, l'inquinante misurato dalla stazione e la rispettiva concentrazione giornaliera (che servono per calcolare il valore di AQI nei modi descritti in precedenza). Il tutto viene estratto dalla vista materializzata "vistagiornaliera", ed è relativo al giorno scelto dall'utente.

Per l'algoritmo usato per calcolare l'AQI i passi sono stati esattamente quelli descritti in precedenza e presenti nella figura 5.41.

Dalla figura 5.50 si può notare come per la città di Milano, nella mappa messa a disposizione per l'utente, sono stati tracciati in blu dei confini che dividono la città nelle varie zone da cui essa è composta. Per fare questo si è fatto ricorso al servizio offerto da Google ossia le Google Fusion Tables.

Questo servizio permette di creare in maniera agevole delle mappe "arricchite" da dati, ovvero di "geotaggare" un insieme di dati esistente. È uno strumento molto potente e facile da utilizzare, che crea rapidamente dei mash-up tra fogli di calcolo e mappe.

Nel progetto in questione, come accennato precedentemente, si è fatto ricorso a tale servizio per dividere la città di Milano nelle varie zone da cui è formata. Questo è stato possibile grazie ad un file KML, che è un linguaggio basato su XML creato per gestire dati geospaziali in tre dimensioni, contenente le informazioni relative alla suddivisione di Milano nelle varie zone. Questo file è stato caricato nel servizio delle Google Fusion Tables, dal quale poi si è estratto del codice, che è stato inserito nel file php creato, in particolare nello script contenente le funzioni necessarie per visualizzare la mappa. Il codice integrato in php per far funzionare il servizio offerto dalle Google Fusion Tables è quello di figura 5.52:

```
layer = new google.maps.FusionTablesLayer({
map: map,
heatmap: { enabled: false },
query: {
 select: "col2",
 from: "1b0soXpXJEGElOb8iCRq72yMzl1YxjEYgCOLsKE95",
 where: ""
 },
 options: {
 suppressInfoWindows:true,
 styleId: 2,
 templateId: 2
}   });
```

Figura 5.52: Codice per l'integrazione delle Google Fusion Table in php

La seconda visualizzazione che viene mostrata all'utente nella sezione "AQI Report" è riportata in figura 5.53:

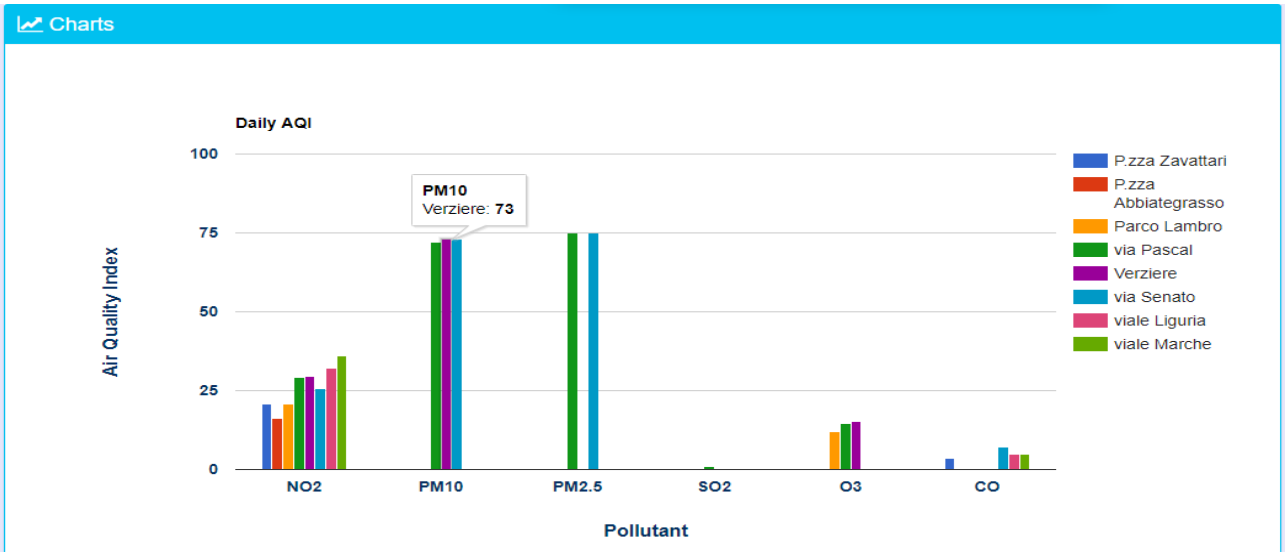


Figura 5.53: Schermata che mostra i grafici a barre dell'AQI per ogni inquinante di ogni stazione

Per cui l'utente dopo aver scelto la data di interesse, oltre a visualizzare la mappa di figura 5.50, avrà la possibilità di consultare un grafico costituito da una serie di istogrammi (suddivisi per inquinante) e per ogni inquinante ci sono tante barre verticali quante sono le stazioni di monitoraggio che misurano l'inquinante indicato sull'asse orizzontale. L'appartenenza di una determinata rilevazione ad una stazione è facilmente riconoscibile tramite la differente colorazione delle barre in funzione proprio della stazione di monitoraggio.

Queste barre verticali, quindi, mostrano il valore di AQI, calcolato per ogni inquinante da ogni stazione, nel giorno indicato dall'utente. Quindi ad esempio, l'NO2 che è misurato da tutte le stazioni, ha 8 barre verticali, ognuna per una stazione e con un proprio AQI, mentre ad esempio l'O3 che è misurata da solamente tre stazioni, ha solamente 3 barre verticali relative alle stazioni che lo misurano.

Per realizzare il seguente grafico è stato necessario scrivere la seguente query:

```
SELECT Nome_Stazione, Inquinante, Media_Giornaliera
FROM vistagiornaliera
WHERE Data ='".$giorno."'
```

Una semplicissima query in cui si estraggono dalla "vistagiornaliera" le informazioni sul Nome della Stazione, l'inquinante e la Media giornaliera e in cui l'unico vincolo è rappresentato dalla data che deve essere quella scelta dall'utente.

In seguito ne viene calcolato il valore di AQI per ogni stazione e per ogni inquinante, sempre nelle modalità descritte in precedenza, con l'unica differenza che questa volta per ogni stazione non viene preso in considerazione solo l'AQI massimo, ma vengono presi in considerazione tutti gli AQI poiché appunto nel grafico mostrato si vuole avere una panoramica generale su tutti gli AQI di tutti gli inquinanti.

Per realizzare questo grafico si è fatto ricorso, come per i grafici dell'applicazione Android, ai Google Charts. Queste, come è già stato detto, sono un insieme di API utilizzabili in qualsiasi contesto web che abbia un linguaggio server side.

Per realizzare dei grafici con Google Chart è necessario scrivere del codice JavaScript da inserire nell'head della pagina HTML creata.

Le istruzioni in JavaScript che permettono di disegnare il grafico sono le seguenti:

```
var chart = new
google.visualization.ColumnChart(document.getElementById('chart_div'));
chart.draw(data, options);
```

Dove in chart viene definito il tipo di grafico da disegnare (ColumnChart in questo caso) andando ad indicare l'elemento della pagina su cui accedere identificato tramite l'id, questo viene fatto tramite l'istruzione document.getElementById. A questo punto, viene disegnato il grafico tramite il metodo draw che contiene due parametri.

Uno di questi è options, il quale contiene tutte le caratteristiche grafiche da dare al grafico, quindi si definiscono gli assi, il carattere, il colore, il titolo ecc. In figura 5.54 un estratto di codice:

```
var options = {
  title: 'Daily AQI',
 is3D: 'true',
 width: 1000,
height: 500,
vAxis: { title: "Air Quality Index" ,
textStyle: {
 fontSize: 14,
 color: '#053061',
 bold: true,
 italic: false
},
```

Figura 5.54: codice di impostazione delle caratteristiche del grafico

L'altro parametro contenuto nel metodo draw è data. Data è una variabile che è stata definita sempre nel codice JavaScript scritto per disegnare il grafico in cui sono inseriti i dati su cui costruire il grafico.

```
var data = new google.visualization.DataTable(<?=$jsonTable?>);
```

E' questa l'istruzione con cui andiamo ad inserire in data la tabella proveniente dai dati lavorati in php e trasformati poi in formato Json affinchè sia leggibile dalle API di Google.

Per convertire in php una tabella, in una in formato JSON è stato usato il seguente codice:

```
$jsonTable = json_encode($table);
```

Si usa quindi il metodo json_encode. A sua volta $table è una tabella creata in php e che contiene tutte informazioni necessarie per costruire il grafico. E' necessario quindi definirne le colonne e le righe di tale tabella, al cui interno sono state inserite tutte le informazioni relative agli inquinanti, alle stazioni di appartenenza e ai rispettivi valori di AQI.

Dopo aver visto la visualizzazione su mappa e quella appena descritta, si passa ora all'ultimo grafico visualizzato dall'utente nella sezione "AQI Report". Questo è mostrato in figura 5.55.

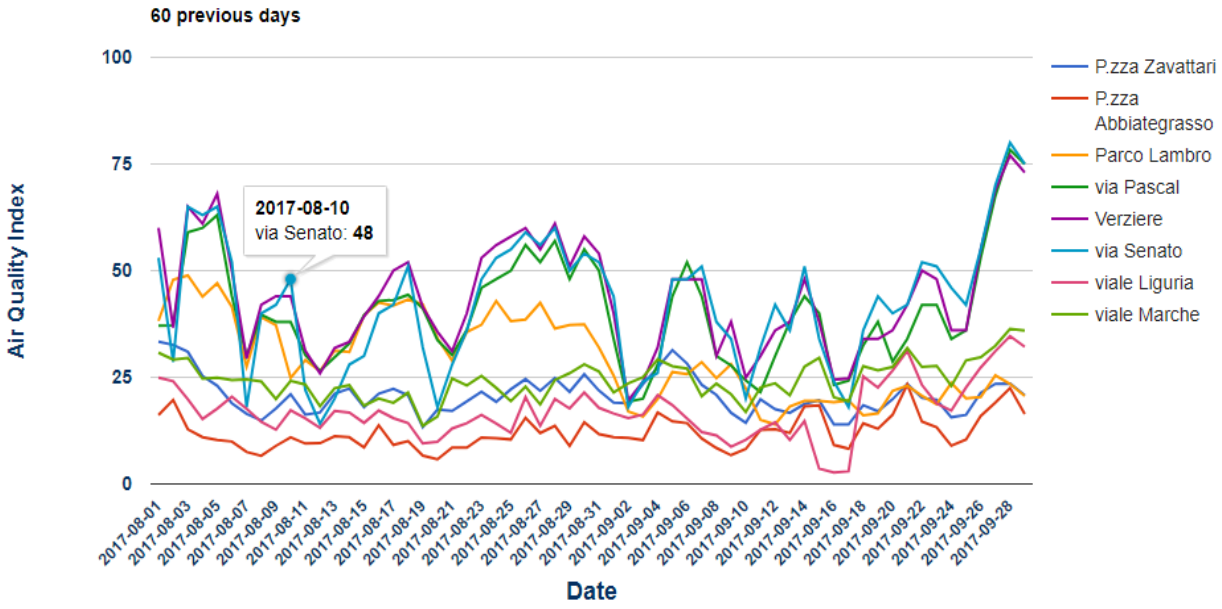


Figura 5.55: Schermata che mostra i grafici a linee dell'AQI di ogni stazione, negli ultimi 60 giorni

Come si può notare questo grafico permette all'utente che lo consulta di farsi un'idea su quello che è stato l'andamento della qualità dell'aria negli ultimi 60 giorni a partire da quello scelto dall'utente.

Ci sono 8 diverse linee, ognuna identifica una singola stazione, la legenda affianco associa il colore della linea alla stazione. L'utente passando il mouse su uno dei punti delle varie linee vedrà nello specifico, il giorno in questione, la stazione e il valore preciso di AQI (nella figura 5.55 "2017-08-10 Via Senato 48").

Anche in questo caso, come in quello precedentemente descritto, per la realizzazione del grafico si è ricorso al servizio offerto da Google ossia i Google Chart. La query usata per estrarre i dati necessari da usare per questo grafico è la seguente:

```
SELECT Nome_Stazione. Data, Inquinante, Media_Giornaliera
FROM vistagiornaliera
WHERE Data BETWEEN adddate('".$giorno."',-59) and '".$giorno."'
GROUP BY Data,Nome_Stazione,Inquinante
```

In questa query vengono estratti i nomi delle stazioni, le date, gli inquinanti e le rispettive misurazioni di concentrazione giornaliera dalla tabella "vistagiornaliera", per i giorni compresi tra quello selezionato dall'utente e i 59 giorni precedenti. Anche qui nel codice php vengono fatti gli opportuni calcoli per ottenere il valore di AQI massimo per ogni giorno preso in considerazione e per ogni stazione, seguendo le stesse logiche già spiegate in precedenza.

Come detto, vengono anche in questo caso usati i Google chart, quindi viene scritto del codice JavaScript dove la differenza sostanziale rispetto al caso precedente, oltre ai dati che chiaramente sono diversi, è quella che non si tratta più di una ColumnChart bensì di una LineChart. Dal punto di vista del codice questa volta si scrive:

```
var chart = new
google.visualization.LineChart(document.getElementById('60ggaqi'));
chart.draw(data, options);
```

Dopo aver descritto cosa visualizza l'utente, e come è stata realizzata, la pagina a cui si accede cliccando sulla voce del menu AQI Report, si passa ora a descrivere cosa avviene quando l'utente clicca sull'altra voce del menù, ossia "Days Above Threshold".

Questa sezione va a risolvere il KPI F, ossia quello di concentrazione di inquinanti critica, che analizza la frequenza con cui i valori dell'Air Quality Index per inquinante sono critici. In particolare, riporta la percentuale di giorni entro un particolare periodo di tempo in cui il valore AQI per inquinante è critico in base ad una soglia specificata dall'utente.

In questa sezione, l'utente dovrà compiere più operazioni rispetto alla sezione descritta in precedenza in cui doveva solamente selezionare la data. Infatti egli avrà di fronte la schermata di selezione mostrata in figura 5.56.

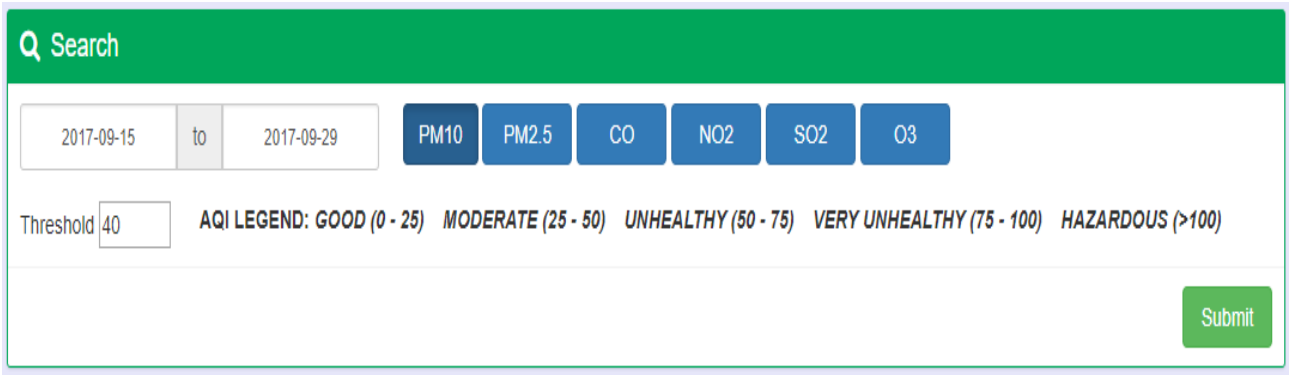


Figura 5.56: Schermata di selezione dell'utente dopo aver cliccato la voce del menu Days Above Threshold

Come intuibile dalla figura qui sopra, l'utente potrà per prima cosa scegliere un intervallo temporale entro cui effettuare l'analisi, scegliendo quindi una data iniziale e una finale. In seguito dovrà scegliere uno degli inquinanti a disposizione e su cui vorrà effettuare l'analisi (PM10, PM2.5, CO, NO2, SO2, O3). Fatto questo, l'utente può inserire manualmente, accanto alla voce "Threshold" la soglia di AQI da prendere come riferimento e su cui calcolare il KPI.

Nell'esempio riportato in figura 5.56 si suppone che l'utente voglia fare l'analisi per un periodo di giorni che va dal 15 Settembre 2017 al 29 Settembre 2017, avendo selezionato l'inquinante PM10 e inserito una soglia di AQI pari a 40.

Dal punto di vista del codice, chiaramente è stata creata una nuova pagina php. Anche in questo caso, per la selezione delle date si è fatto ricorso ad un widget DatePicker nello stile Bootstrap. A differenza di quello descritto in precedenza, in cui si doveva scegliere una data singola, in questo caso è necessario scegliere due date che indichino il range entro il quale fare le operazioni, quindi una data iniziale e una finale. Nella figura 5.57 è riportato il codice che fa questo.

```
<script type:"text/javascript">
  var end= '2017-09-29'
  var start= '2015-01-01'
  $('.input-daterange').datepicker({
    format: "yyyy-mm-dd",
     autoclose: true,
    defaultViewDate: { year: 2017, month: 09, day: 29 },
     endDate: end,
     startDate: star
```

Figura 5.57: Codice necessario per utilizzare il widget DatePicker

Per quanto riguarda invece la scelta degli inquinanti si tratta di una semplice serie di bottoni ottenuti con il tag dell'input type in cui ad ognuno viene assegnato il rispettivo valore. Ad esempio:

<input type="submit" class="btn btn-primary active" name="inquin" value="PM10" "style="width:75px">

Fatte queste operazioni, la schermata consultabile dall'utente, e su cui poter effettuare delle analisi sarà quella riportata in figura 5.58

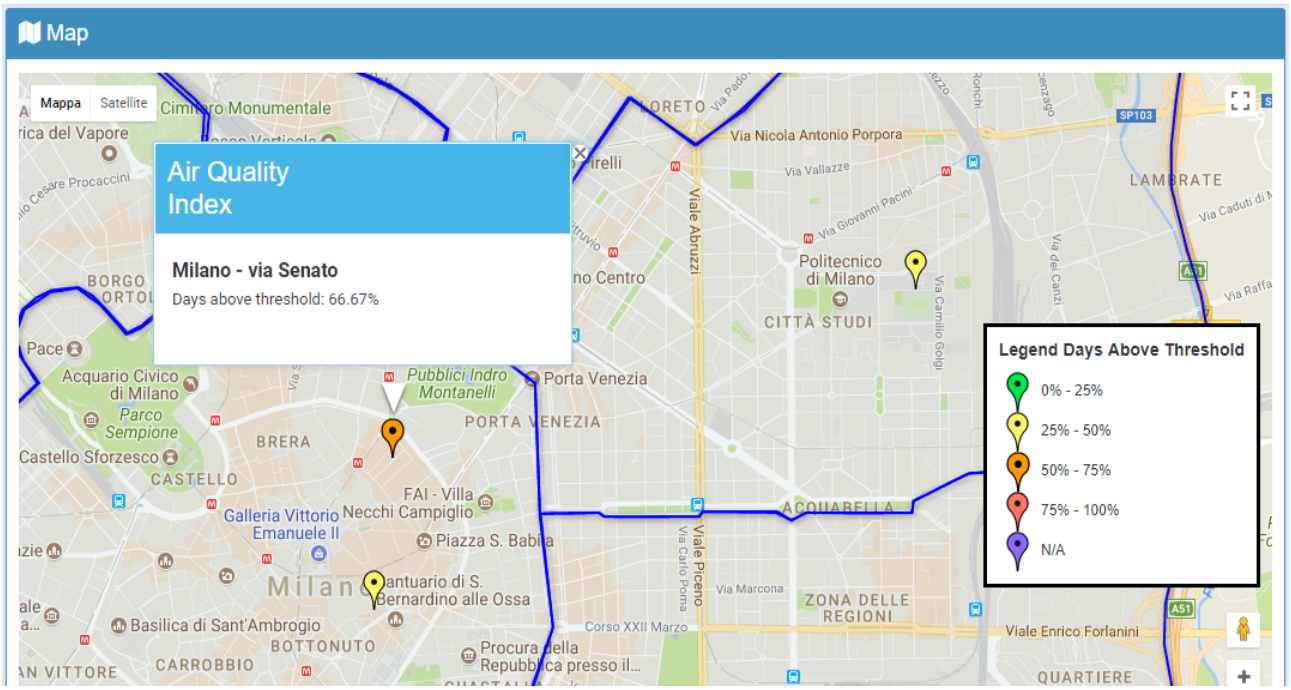


Figura 5.58: Schermata che mostra la mappa ottenuta cliccando sulla voce del menu Days Above Threshold

Come si vede da questa immagine, l'utente potrà visualizzare una mappa in cui sono riportati dei marker che rappresentano le singole stazioni. Chiaramente, nell'esempio, sono visualizzati solamente tre marker poiché sono solo tre le stazioni che misurano il PM10 che era, sempre nell'esempio, l'inquinante scelto dall'utente. Per cui, se l'utente avesse scelto ad esempio l'inquinante NO2, sarebbero stati visualizzati 8 marker.

Questi marker hanno una colorazione diversa, a seconda della fascia in cui essi ricadono ben descritta nella legenda presente a destra della mappa. Ad esempio, la stazione di Via Senato ha un colore arancione. Questo vuol dire che la percentuale di giorni, tra quelli nel range scelto dall'utente, in cui il valore di AQI misurato da quella stazione, è superiore al valore di riferimento scelto dall'utente (in questo caso 40), ricadrà nella fascia compresa tra il 50% e il 75%. Per vedere il valore preciso basta cliccare sul marker e leggere i dati presenti. Come si vede in figura 5.58 ad esempio, per la stazione di Via Senato, la percentuale di giorni in cui il valore di AQI scelto dall'utente come soglia è stato superato è pari al 66,67%. Questo chiaramente vale anche per tutte le altre stazioni.

Anche per la realizzazione e visualizzazione di questa mappa si è ricorsi alle API di Google di Google Maps integrate nella pagina php creata.

È stato già spiegato in precedenza come integrare queste API nella pagina php, per cui ora viene esposto solamente come è stata calcolata la percentuale di giorni che hanno superato la

soglia inserita dall'utente. Questa operazione è stata fatta con codice JavaScript presente in figura 5.59.

```
for(var h=0;h<vettore.length;h++){
if(vettore[h]>sogliautente)
count++;
}
if(vettore.length>0){
var media=parseFloat((count/vettore.length)*100).toFixed(2);
```

Figura 5.59: Calcolo della percentuale di giorni

vettore[h] è il vettore contenente tutti i valori di AQI di ogni singolo giorno del range stabilito dall'utente, per cui per ogni giorno, si vede se il valore di AQI è maggiore della soglia inserita dall'utente. Se si, viene incrementata la variabile count. A questo punto nella variabile media ci sarà la percentuale di giorni in cui il valore di AQI è stato superato.

In seguito per colorare i marker in base al valore presente nella variabile media calcolata sono state necessarie le istruzioni di figura 5.60:

```
        if(media<=soglia1)
        marker.setIcon('http://maps.google.com/mapfiles/ms/icons/green-
dot.png');
        else if(media>soglia1 && media<=soglia2)
        marker.setIcon('http://maps.google.com/mapfiles/ms/icons/yellow-
dot.png');
        else if(media>soglia2 && media<=soglia3)
        marker.setIcon('http://maps.google.com/mapfiles/ms/icons/orange-
dot.png');
        else if(media>soglia3 && media<=soglia4)
        marker.setIcon('http://maps.google.com/mapfiles/ms/icons/red-dot.png');
```

Figura 5.60: confronti con le varie soglie

dove soglia1, soglia2, soglia3 e soglia4 sono stati opportunamente definiti nella parte iniziale del codice JavaScript scritto.

Per estrarre i dati necessari da inserire nella mappa descritta, è stata eseguita la seguente query:

```
SELECT Lat, Lng, Nome_Stazione, Data, Media_Giornaliera
FROM vistagiornaliera
WHERE `Inquinante`='".$inqui."' AND Data BETWEEN '".$inizio."' and '".$fine."'
GROUP BY Data, Nome_Stazione
```

Dalla seguente query vengono estratti la latitudine e longitudine che servono per inserire nella mappa il marker nella posizione corretta, il nome della stazione, la data e il misurazione della sua concentrazione giornaliera. Il tutto proveniente dalla tabella "vistagiornaliera" con il vincolo sull'inquinante, che sarà quello scelto dall'utente in fase di selezione e sulla data che dovrà essere compresa tra quella iniziale e quella finale inserita dall'utente.

Come in tutti gli altri casi descritti precedentemente anche in questo caso in php sono stati effettuati i calcoli per trasformare la concentrazione dell'inquinante nel rispettivo valore di AQI.

6. Osservazioni finali

In questo capitolo vengono fatte delle osservazioni interessanti che mostrano l'effettivo utilizzo di quanto creato.

Dall'osservazione degli applicativi sviluppati, si è potuto giungere ad alcune considerazioni riguardanti il fenomeno dell'inquinamento.

In particolare, a risultare utile per effettuare analisi soggette ad interpretazione è la sezione "Air Quality Index" della dashboard. Questo è in linea con l'obiettivo iniziale, ossia quello che la dashboard è destinata principalmente al personale dell'amministrazione locale che vuole avere una panoramica generale di quelle che possono essere le cause dell'inquinamento atmosferico.

Dall'applicazione invece, che è rivolta principalmente al cittadino interessato a monitorare una determinata zona della città, è più difficile effettuare delle possibili analisi poiché non permette di visualizzare dati storici e quindi effettuare confronti particolarmente rilevanti.

Una prima osservazione rilevante è quella fatta nella sottosezione "Days Above Threshold". Questa sottosezione è quella che permette di definire un intervallo di giorni da valutare e l'inquinante, fornendo una mappa in cui ogni marker rappresenta una determinata stazione e questo avrà un colore diverso a seconda della percentuale di giorni nell'intervallo scelto, in cui l'AQI dell'inquinante selezionato, supera un valore definito dall'utente.

Si è provato ad effettuare questa analisi scegliendo come inquinante il PM10, e come intervallo di date due periodi, quello invernale e quello estivo. Per quanto riguarda quello invernale si è preso come riferimento l'intervallo di date che va dal 1 Ottobre al 31 Marzo. Il valore inserito di AQI entro cui valutare la percentuale di giorni, è il valore di 50, ossia il valore di AQI a partire dal quale la qualità dell'aria non è salutare. In figura 6.1 è riportata l'immagine di quello che viene visualizzato quando l'utente inserisce le date relative al periodo invernale.

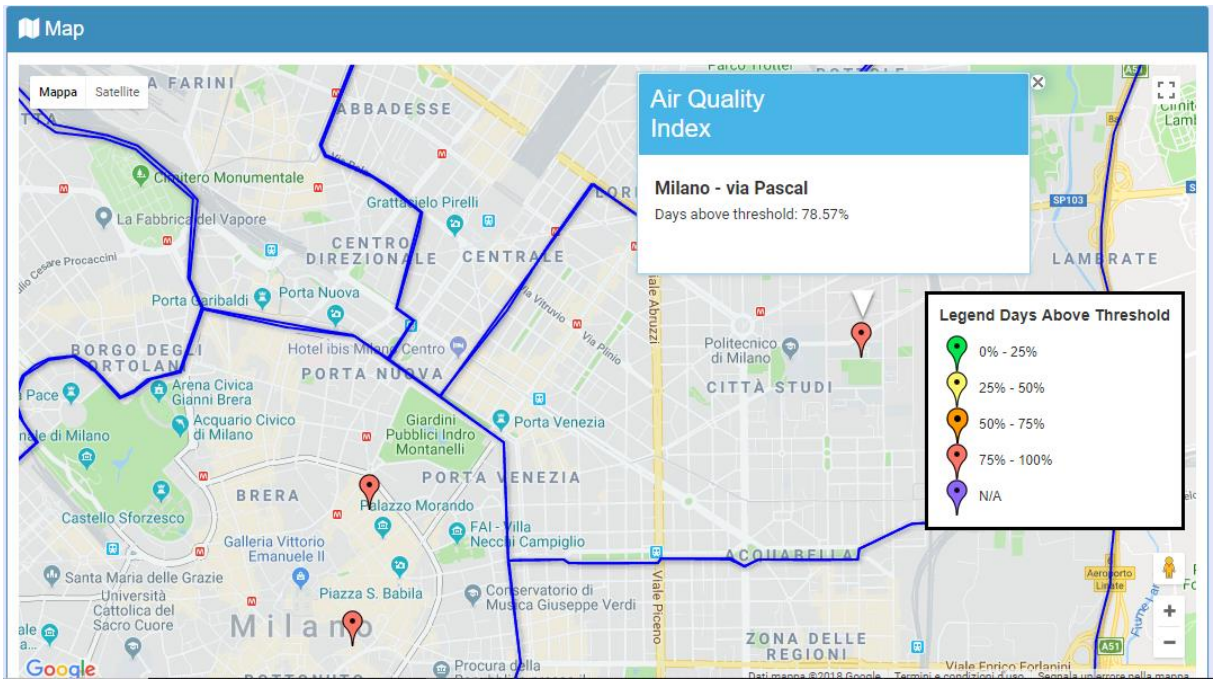


Figura 6.1: Schermata che mostra la mappa ottenuta nel periodo invernale

Come si può notare, tutte e tre le stazioni che misurano il PM10, nel periodo invernale, hanno una percentuale di giorni in cui l'AQI supera il valore di 50, pari ad un valore che va tra il 75 e

il 100% dei giorni. Ad esempio nella stazione di via Pascal il 78,57% dei giorni questo valore viene superato.

Viene riportata in figura 6.2 la mappa che si ottiene inserendo come inquinante PM10, come valore soglia 50, ma come periodo quello estivo, quindi dal 1 Aprile al 30 Settembre.

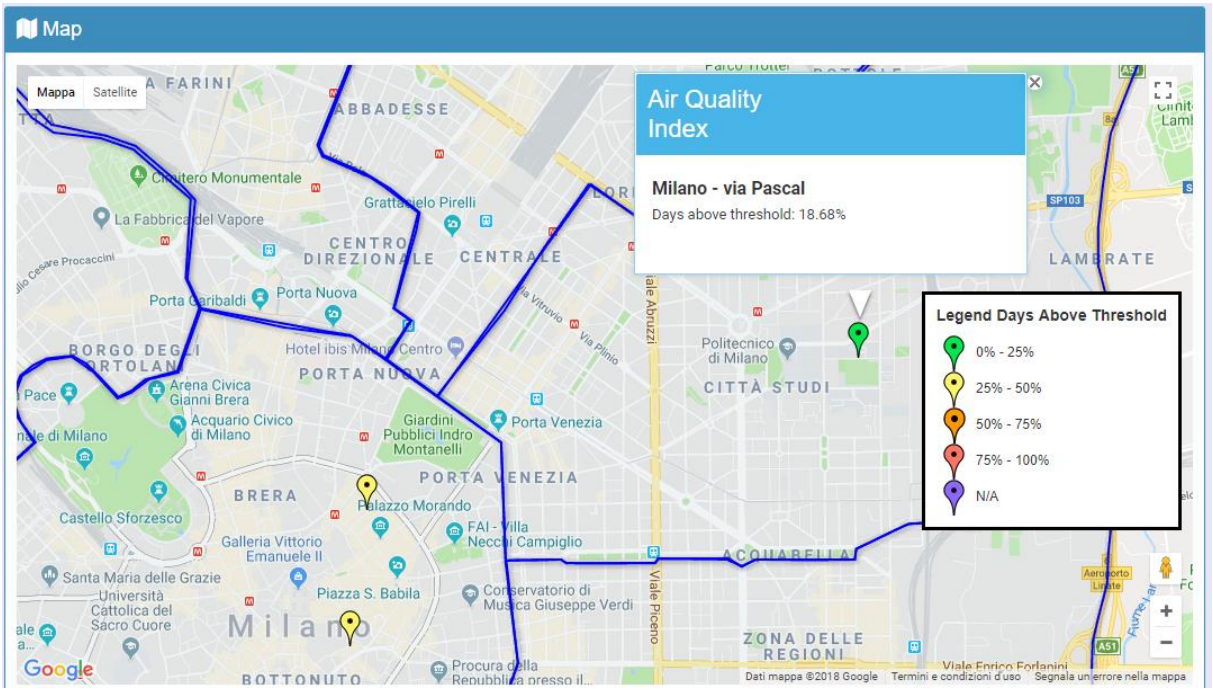


Figura 6.2: Schermata che mostra la mappa ottenuta nel periodo estivo

Si può notare come nel periodo estivo la situazione sia notevolmente migliorata. Infatti per le stazioni di Verziere e via Senato la percentuale di giorni in cui il valore di AQI supera il valore di 50 ricade nella fascia che va dal 25 al 50%, mentre per via Pascal è addirittura inferiore in quanto la percentuale di giorni è pari al 18,68%.

Si è quindi giunti a questa conclusione. Nel periodo invernale, la qualità dell'aria è molto peggiore rispetto a quella estiva. Questo probabilmente è da attribuire all'accensione dei riscaldamenti che va ad influire in negativo sulla qualità dell'aria. Altro fattore che potrebbe influire è il fatto che nel periodo estivo si è meno propensi ad usare la macchina a favore magari della bicicletta o altri mezzi che inquinano meno.

Un'altra importante osservazione è stata effettuata nella sottosezione "AQI Report" e in particolare il grafico che mostra gli AQI dei vari inquinanti. Si è visto infatti, che prendendo a campione varie date, gli inquinanti con AQI più alto sono sempre le polveri sottili (PM2.5 e PM10). Non a caso, sono proprio questi gli inquinanti di cui si sente parlare sempre più spesso e che le varie amministrazioni cercano di combattere poiché sono quelle che vanno ad influire maggiormente sulla qualità dell'aria. In figura 6.3 è mostrato un grafico a barre con gli AQI dei vari inquinanti in un giorno a caso in cui si nota come il PM10 e il PM2.5 siano gli inquinanti con AQI più alto.

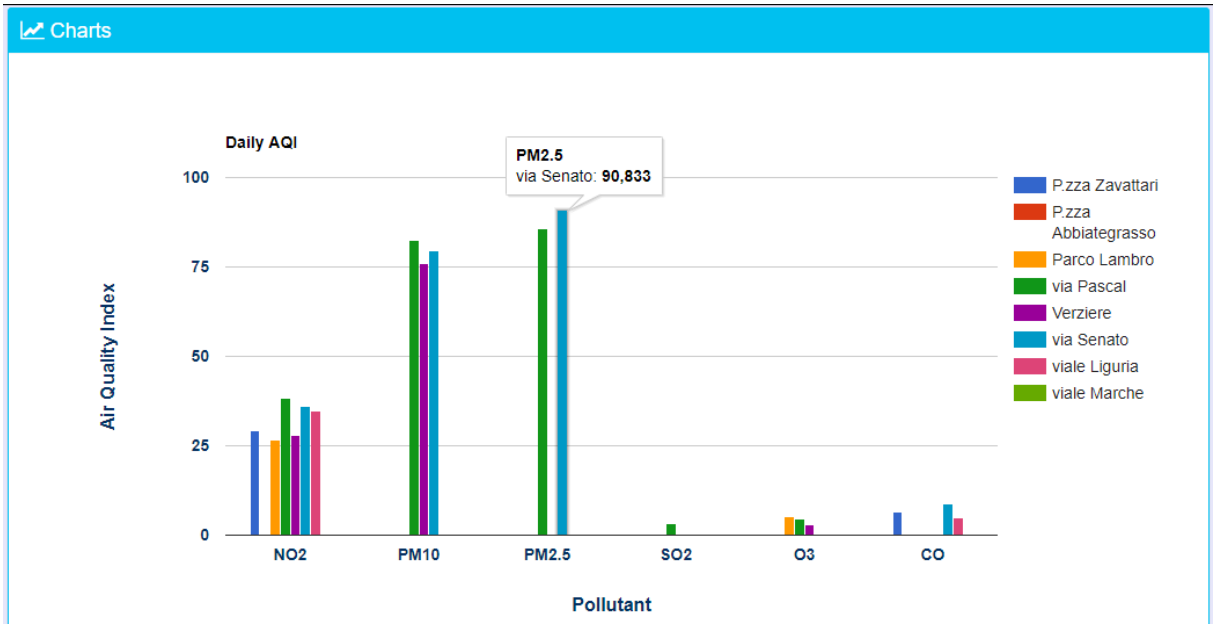


Figura 6.3: Schermata che mostra gli inquinanti che maggiormente influenzano l'AQI

A dimostrazione di quanto detto, è possibile visualizzare in figura 6.4 il grafico relativo ai valori di AQI nei 60 giorni precedenti rispetto a una data selezionata, in cui vi è il grafico a linee delle varie stazioni. Le stazioni che rilevano le concentrazioni di PM10 e PM2.5 sono Senato, Pascal e Verziere. Nella figura di 6.4 vediamo come le stazioni che hanno dei valori tendenzialmente più alti di AQI sono proprio quelli corrispondenti alle stazioni appena citate che sono quelle che misurano le concentrazioni di polveri sottili.

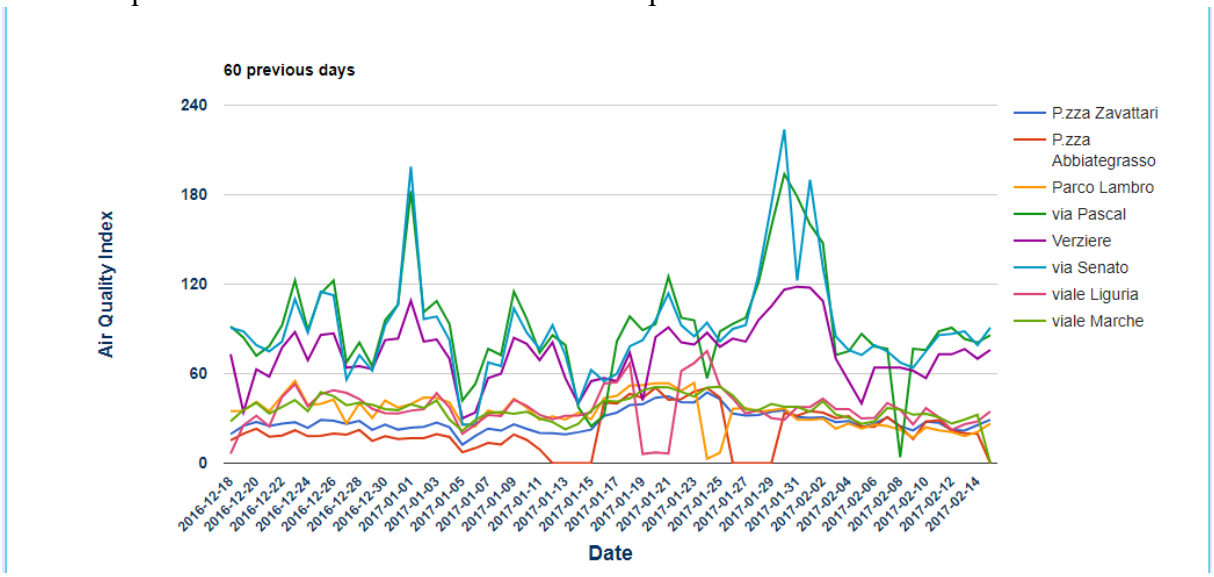


Figura 6.4: Schermata che mostra gli AQI nei 60 giorni precedenti a quello selezionato

7. Conclusioni e Sviluppi futuri

La tesi aveva come obiettivo quello di progettare un Data Warehouse contenente i dati sull'inquinamento atmosferico nell'ambiente urbano, messi a disposizione dal comune di Milano, e i dati relativi ad alcuni fenomeni che possono influenzare l'inquinamento come il traffico e il meteo. Da questo poi sono stati estratti indicatori importanti alla base di quello che viene poi visualizzato nella dashboard e applicazione mobile sviluppata.

L'app è stata pensata prevalentemente per il cittadino interessato ad avere informazioni su quella che è la qualità dell'aria nella zona di interesse. L'app sviluppata dà per il momento solo informazioni riguardanti appunto l'indice di qualità dell'aria relative alle varie stazioni di Milano. Tuttavia, si potrebbe in futuro espandere le funzionalità dell'app.

In particolare, in virtù anche del DW progettato e dei dati inseriti nel database la conseguenza naturale per accrescere le funzionalità dell'app sarebbe quella di permettere all'utente di ricevere anche le informazioni riguardanti fenomeni atmosferici (temperatura, pressione, velocità del vento, livello di precipitazioni, umidità) e il traffico in città.

Potrebbe essere anche utile cercare di trovare una possibile correlazione che c'è tra i fenomeni metereologici e il traffico, con i valori di AQI.

Altra possibile implementazione futura potrebbe essere quella di inserire anche sull'applicazione, come fatto per la dashboard, una visualizzazione su mappa delle stazioni presenti permettendo una visualizzazione grafica più moderna.

Attualmente l'app permette di monitorare solamente i dati di AQI relativi all'ultimo giorno inserito nel Database, e al massimo può visualizzarne l'andamento dell'AQI negli ultimi dieci giorni. Quindi, all'utente non viene data la possibilità di monitorare uno specifico giorno di interesse. Ecco perché potrebbe essere molto interessante potergli permettere una maggiore interazione con l'app. Per fare questo, si potrebbe dare la possibilità di scegliere una data e visualizzare l'andamento della qualità dell'aria nella data scelta.

Si potrebbe anche, prendendo spunto anche da altre app già esistenti, effettuare una previsione futura dei valori di AQI, quindi in base ai dati metereologici dei giorni successivi a quello corrente, è anche possibile fornire all'utente ad esempio una previsione di AQI nei 7 giorni successivi.

Per quanto riguarda la sezione di "Air Quality Index" inserita nella dashboard, essa è stata pensata principalmente per il personale della pubblica amministrazione.

Anche in questo caso potrebbe essere interessante descrivere alcuni dei possibili sviluppi futuri che potrebbero essere presi in considerazione per poter ampliare questa sezione. Si potrebbe inserire una sottosezione con le previsioni future dell'AQI, quindi una previsione ad esempio nei prossimi sette giorni sfruttando quelli che sono le previsioni del meteo dei giorni successivi.

Un'ulteriore sviluppo futuro che migliorerebbe l'esperienza dell'utente nell'usare tale dashboard sarebbe quello di permettere, anche sulla dashboard, un focus sulle singole stazioni, come già avviene per l'app, magari permettendo anche dei confronti tra due o più stazioni.

8. Bibliografia e Sitografia

[1] Ramakrishnan R., Gehrke J., “Sistemi di basi di dati”, McGraw-Hill, 2004

[2] Golfarelli M., Rizzi S., “Data Warehouse – teoria e pratica della progettazione”, McGraw-Hill, 2006

[3] Corno F., Torchiano M., “Sistemi Informativi Aziendali – Appunti per il corso”, 2018, https://elite.polito.it/files/courses/02CIX/libro/cap10-standalone.pdf

[4] Inquinamento atmosferico, https://it.wikipedia.org/wiki/Inquinamento_atmosferico, consultato il 05/02/2018

[5] Nozioni di base, http://www.airqualitynow.eu/it/pollution_home.php, consultato il 05/02/2018

[6] Inquinanti,http://www.arpa.piemonte.it/approfondimenti/temi-ambientali/aria/aria/cartella-qualita-inquinanti, consulato il 05/02/2018

[7] MySQL, https://it.wikipedia.org/wiki/MySQL, consultato il 06/02/2018

[8] PHP, http://www.html.it/guide/guida-php-di-base/, consultato il 06/02/2018

[9] Javascript, http://www.html.it/guide/guida-javascript-di-base/, consultato il 06/02/2018

[10] AngularJS, http://www.html.it/guide/guida-angularjs/, consultato il 06/02/2018

RINGRAZIAMENTI

Desidero innanzitutto ringraziare le Professoresse Silvia Chiusano e Tania Cerquitelli per i numerosi consigli e la grande disponibilità dimostrata durante lo sviluppo della tesi.
Ringrazio i miei genitori e mia sorella, per il prezioso e indispensabile sostegno morale ed economico che mi hanno fornito in questi anni e che mi hanno permesso di raggiungere questo importante traguardo.
Ringrazio i miei stupendi nonni, tutti i miei zii e i miei cugini.
Ringrazio tutti i miei amici, in particolare il gruppo "Corsa", diventato ora "lu gruppu Costagnigni", il miglior project manager del canavese Lucio, i miei cari amici di "giù", e infine un grande ringraziamento va al mio amico e collega Ciccio con cui nell'ultimo anno abbiamo condiviso la realizzazione di questa tesi e la realizzazione del "supermegaprogettogalattico""RealZone".