

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica Per il Management

**Progettazione ed implementazione di
un'applicazione mobile per la
visualizzazione di Open Data ambientali**

Relatore:
Chiar.mo Prof.
Marco Di Felice

Presentata da:
Fabrizio Camassa

Sessione II
Anno Accademico 2015/2016

Ai miei amici ...

Indice

<i>Introduzione</i>	1
1 Open Data - Le Origini	3
1.1 L'era dell'ipertesto	4
1.2 L'era del WWW	6
1.3 L'era del Web 2.0	9
1.4 L'era delle App e dei social	10
1.5 Gli Open Data - I dati aperti	13
1.6 Gli Open Data - I principi	17
1.7 Gli Open Data - La catalogazione	18
1.8 Gli Open Data - Esempi pratici & competitors	20
1.9 I dati ambientali	25
2 Progettazione dell'app	31
2.1 Architettura	31
2.2 CKAN	33
2.3 MongoDB	34
3 Implementazione - parte 1	37
3.1 Il Web Service	37
3.2 I dati	38
4 Implementazione - parte 2	45
4.1 Android	45

<i>4.2 Implementazione dell'app</i>	<i>46</i>
<i>Conclusioni</i>	<i>52</i>
<i>Ringraziamenti</i>	<i>53</i>
<i>Bibliografia</i>	<i>55</i>

Introduzione

Al giorno d'oggi i dati rappresentano una parte fondamentale della nostra vita, sia dal punto di vista teorico (in quanto maggiori sono i dati in nostro possesso, maggiore è la nostra conoscenza), sia dal punto di vista pratico (ricerche scolastiche, navigazione online, social media, ecc.).

Parte integrante di questo studio sono stati i dati in formato aperto, messi a disposizione gratuitamente (quindi open data), dall'ente ARPA Emilia Romagna e l'utilizzo di un sistema di basi di dati open source per la loro corretta gestione di essi, chiamato MongoDB.

Gli Open Data rappresentano una tematica ad oggi molto poco conosciuta ma in fortissima espansione. La loro importanza è sottovalutata e soltanto da pochi anni si stanno vedendo i primi investimenti in questo campo.

I servizi che trattano questa tipologia di dati derivano fundamentalmente da enti pubblici, ma negli ultimi tempi stanno nascendo portali e piattaforme online dove anche utenti comuni, possono mettere a disposizione le proprie rilevazioni, per esempio ThingSpeak, CKAN ecc.

In questa tesi verranno anche illustrati numerosi esempi di applicazioni mobile basati sugli Open Data, con particolare attenzione a quelle riguardanti i dati ambientali.

Durante l'elaborazione, ho cercato di approfondire ed incrementare la mia conoscenza

in merito agli open data, cercando prima di tutto nelle fonti consigliate dal mio relatore ed in seguito svolgendo alcune ricerche per conto mio.

Alla fine di questo studio ho pertanto deciso di affrontare tre aspetti fondamentali per la stesura del mio lavoro:

- Parleremo anzitutto di come si sono sviluppati i dati nel corso degli anni, per poi specificare bene il ruolo degli open data all'interno di questa categoria e come vengono utilizzati al giorno d'oggi.
- La seconda parte introdurrà il mio lavoro in fase di progettazione, le scelte che ho fatto per arrivare al risultato finale, gli errori ed i problemi riscontrati per via della mancanza di conoscenza degli open data messi a disposizione e l'uso di MongoDB come sistema di basi di dati.
- La terza parte tratterà infine dello sviluppo vero e proprio, con particolare attenzione all'elaborazione del Webservice a cui far appoggiare l'app creata, all'utilizzo di strumenti messi a disposizione per la costruzione dell'interfaccia grafica quali le API di google e le librerie Python per la comunicazioni e le richieste al Webservice.

Capitolo 1

Open Data - Le Origini

Fabrizio Camassa

La definizione di open data è presto deducibile dal nome stesso, si tratta di dati messi a disposizione liberamente da colui che li ha creati e resi accessibili a chiunque, con le uniche possibili restrizioni di mantenere la banca dati sempre aperta a tutti, o piu' semplicemente l'obbligo di citare la fonte di provenienza per dare un minimo di credito e riconoscimento al loro creatore. [1]

L'open data si rifà ad una disciplina più grande che è l'open government, cioè una dottrina in base alla quale la pubblica amministrazione dovrebbe essere aperta ai cittadini, sia dal punto di vista della trasparenza che da quello della partecipazione diretta al processo decisionale, che all'etica di altri movimenti di tipo "open" come l'open source, l'open access e l'open content.

Per le origini degli open data dobbiamo tornare indietro negli anni '80 ed analizzare con attenzione l'evoluzione di Internet.

1.1 L'era dell'ipertesto

Negli anni '80 e '90 quando Internet, successore di ArpaNet, cominciava la sua rapida diffusione, le prime informazioni scritte dall'uomo su questa nuova tecnologia furono gli ipertesti, derivanti dall'umana necessità di collezionare, gestire ed infine manipolare informazioni al fine di renderle più accessibili e a disposizione di tutti.

Prima di questa invenzione, l'unico modo conosciuto per tramandare informazioni era la scrittura e prima ancora, ai tempi dei romani, si compivano studi per aumentare artificialmente la memoria al fine di tramandare conoscenze ed informazioni alle generazioni future.

Un ipertesto è un insieme di documenti messi in relazione tra loro per mezzo di parole chiave, pertanto altro non è che una ragnatela di informazioni connesse tra loro, la cui lettura è variabile a seconda del tipo di documento o informazione che si sta cercando. [2]

Il termine fu coniato dal sociologo, filosofo e pioniere dell'informatica statunitense Ted Nelson nel 1963 e successivamente questo termine venne pubblicato ed ufficializzato nel 1965. Con questo termine si voleva sottolineare la mole di informazioni che era possibile collegare e consultare con estrema facilità.[3]

Inizialmente, i sistemi ipertestuali, vennero mantenuti solo ed esclusivamente nei laboratori o nei settori scientifici a scopo di semplice documentazione, ma verso la metà degli anni '80 le prime aziende cominciarono ad interessarsi per la loro commercializzazione, intravedendo il potenziale del loro utilizzo e cominciando ad elargire finanziamenti atti al loro sviluppo.

Nel 1980 Bill Atkinson realizzò Hypercard, un'applicazione software per la Apple che gestiva grandi quantità di informazioni sotto forma di testi e foto, attraverso un linguaggio di programmazione proprietario del software, chiamato Hyper Talk e malgrado fosse un software disponibile solo per la piattaforma MacOS, Hypercard passò alla storia come uno dei più famosi sistemi di realizzazioni di ipertesti prima dell'avvento del World Wide

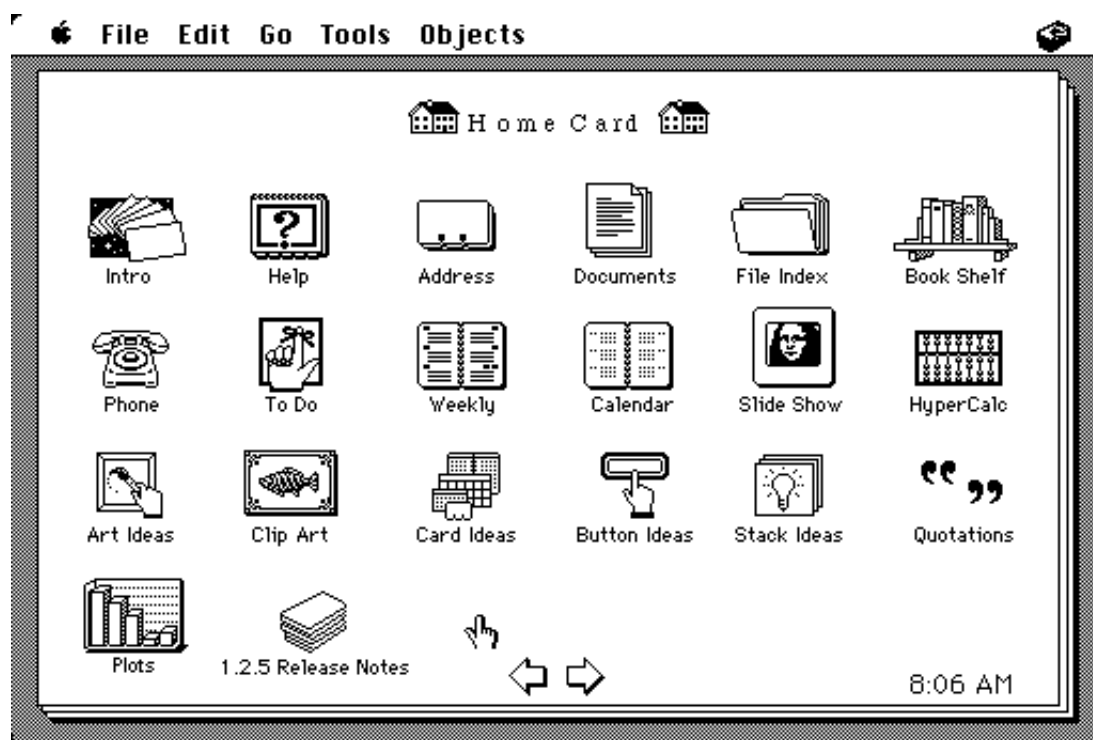


Figura 1.1: Hypercard

Web, che di lì a poco avrebbe dato il vita ad una rivoluzione in campo tecnologico. [4] [5]

Hypercard continuò a ricevere aggiornamenti da parte di Apple fino al 2004, anno nel quale il software fu ritirato ufficialmente dal mercato per problemi di compatibilità con l'architettura delle successive versioni di MacOS.

Per come erano concepiti all'epoca gli ipertesti, non esisteva uno standard comune; i dati erano memorizzati su diverse macchine con sistemi operativi e software differenti tra loro. Per consultare diversi documenti da diverse macchine occorre infatti imparare nuovi linguaggi e studiare nuovi software, il che rallentava costantemente il lavoro di ricercatori e scienziati.

Tim Berners-Lee, un informatico britannico, lavorava come software engineer per il CERN; notata la problematica inerente a questi sistemi informativi ed infastidito da



Figura 1.2: Tim Berners-Lee

quest'ultima, scrisse nel 1989 un memorandum in cui proponeva un Sistema Iper-testuale Globale, che avrebbe unificato quella gigantesca quantità di informazioni rendendole accessibili a chiunque con semplicità.[6]

La sua idea non diede immediatamente i risultati sperati, non era semplice immaginare un sistema virtuale di comunicazione che collegasse tutti questi documenti anche a grandissime distanze, così inizialmente nessuno lavorò a questa possibile via, sicché fu Berners stesso a scrivere il programma.[7]

Cominciò con il definire i concetti base del linguaggio HTML, definì il protocollo HTTP e introdusse il concetto di URL per cercare dati e documenti, senza renderse ne conto Berners-Lee aveva appena dato vita al World Wide Web come lo conosciamo adesso.[6]

1.2 L'era del WWW

L'invenzione di Tim Berners-Lee diede il via ad una corsa allo sviluppo, nacquero i primi browser per la navigazione Web. Il funzionamento era semplice quanto geniale, tramite l'indirizzo URL si accedeva ad un documento visualizzato tramite una pagina

HTML statica e contenente links che permettevano di passare ad altri documenti visualizzati allo stesso modo.[6]

Nel 1993 il CERN rese accessibile il web al pubblico e subito internet divenne un vero e proprio portale dell'informazione. La facilità con cui era possibile generare collegamenti tra i vari documenti non era utile soltanto a sviluppatori e ricercatori, ma anche a coloro che avevano fame di conoscenza e volevano apprendere nozioni a cui altrimenti non avrebbero avuto accesso.

Il numero di informazioni disponibili ebbe una crescita esponenziale, d'un tratto l'intero Web era popolato da milioni di documenti con altrettanti collegamenti; in poco tempo, nacque la necessità di ordinare le informazioni, catalogarle e ricercarne determinate specifiche attraverso l'uso di parole chiave, onde facilitare la navigazione degli utenti.

Queste necessità diedero vita ai motori di ricerca, attraverso i quali, con il semplice ausilio di una parola chiave, consentivano al browser di scremare i contenuti e restituire al richiedente le informazioni ricercate. I primi motori ad essere creati furono Archie e Gopher; il primo di essi derivava da un progetto scolastico del 1990 della McGill University di Montreal, creato dallo studente Alan Emtage, mentre il secondo fu sviluppato l'anno successivo dall'università del Minnesota da Mark McCahill.

Il funzionamento dei due motori di ricerca era radicalmente diverso, in quanto il primo memorizzava un archivio nel quale venivano raccolte pagine e file consultabili attraverso una parola chiave, mentre il secondo utilizzava un paradigma ipertestuale per trovare i riferimenti ricercati nei vari file che analizzava.





Figura 1.3: Larry Page

Sfortunatamente nessuno dei due motori consentiva la ricerca di parole in linguaggio naturale, per questa feature si dovette attendere Wandex (il primo motore di ricerca in un formato simile a quello che usiamo noi oggi) e successivamente WebCrawler che l'anno successivo introdusse l'indicizzazione del testo completo dei siti web.

Nel 1996 Larry Page e Sergey Brin, introdussero il concetto di ranking di pertinenza all'interno del loro motore di ricerca ancora in fase di sviluppo, BackRub. Tale sistema consentiva alle pagine di essere suddivise in ranghi a seconda della loro notorietà all'interno dei motori di ricerca, in questo modo le pagine più gettonate venivano messe in risalto rispetto a quelle comuni. [8] [9]

BackRub ebbe vita breve e visse solo all'interno dell'università dov'era stato sviluppato, in quanto la crescita esponenziale dei dati rendeva impossibile un suo ingresso al fianco di altri motori di ricerca. I due sviluppatori decisero quindi di distaccarsi dalla Stanford University e cominciarono a lavorare al progetto autonomamente.

Dopo poco tempo il nome BackRub fu cambiato e nacque così Google. Questo nome derivava dal termine Googol, un'espressione matematica atta ad indicare un 1 iniziale seguito da 100 zeri. Questo termine rispecchiava l'obiettivo dei due sviluppatori di or-



Figura 1.4: Sergey Brin

ganizzare una quantità illimitata di informazioni sul web. [10]

Andy Bechtolsheim, cofondatore di Sun Microsystem finanziò immediatamente questo motore di ricerca e nel 1998 Larry Page e Sergey Brin fondarono l'odierna Google Inc, un'azienda che conta oggi più di 50.000 dipendenti e con una capitalizzazione azionaria superiore ai 500 miliardi di dollari (a seguito di una ristrutturazione interna, dal 5 ottobre 2015 Google Inc. è controllata dalla holding Alphabet con a capo Larry Page).

Ciò che google offriva rispetto ai suoi concorrenti, furono la precisione del suo l'algoritmo di ricerca e l'interfaccia semplice e ottimizzata, grazie alla quale anche un utente inesperto poteva realizzare ricerche complesse.

1.3 L'era del Web 2.0

Storicamente possiamo parlare di web 2.0 come termine per identificare ciò che avvenne verso la fine degli anni '90, quando si cominciarono a sviluppare applicazioni online che permettessero un alto livello di interazione tra il sito web e l'utente. [11]

Si cominciò con applicazioni come blog, forum, sistemi di chat e le wiki, per poi passare a piattaforme di condivisione di media come Flickr, YouTube, Vimeo e le varie applicazioni social come Facebook, MySpace, Twitter, Google+, LinkedIn, Foursquare ecc.

A differenza degli anni precedenti, ci fu un notevole sviluppo nel modo di concepire le pagine web. L'esigenza di poter interagire con esse in modo creativo, rese necessario da parte degli sviluppatori il passaggio da uno stile prettamente statico e semplice ad uno dinamico e complesso. La pagina doveva essere piena di contenuti, offerti in uno stile visivo piacevole, con la possibilità di potervi interagire.

Complici di questa nuova concezione, furono lo sviluppo dei calcolatori, la diffusione della connessione internet (resa disponibile alla stragrande maggioranza delle popolazioni) e l'uso di linguaggi di programmazione quali CSS (Cascading Style Sheet) e Javascript grazie ai quali era possibile inserire animazioni ed effetti che andavano ben al di là di quello che il semplice HTML offriva.

Il termine Web 2.0 è associato a Tim O'Reilly per via di una conferenza da lui tenuta nel 2004, dove identificava "Web 1.0" il semplice HTML statico, il Web 1.5 l'uso combinato delle pagine web con primordiali sistemi di basi di dati e di sistemi di gestione di contenuti ed infine il Web 2.0 come nuova frontiera dove l'utente era effettivamente in grado di avere una vera e propria interazione con i siti e con gli altri utenti nel mondo. Non erano più i contenuti ad essere protagonisti delle pagine Web, l'utente era il nuovo protagonista: poteva creare dei contenuti, comunicare con altre persone, creare intere comunità di utenti con interessi e bisogni comuni anche a grandissime distanze.[12] [15] [?]

Una delle più grandi innovazioni di quegli anni fu la nascita di portali come Wikipedia, la più grande enciclopedia online del mondo, gestita e modificata dagli utenti stessi.

1.4 L'era delle App e dei social

La panoramica offerta finora, rende in modo chiaro l'idea di come la continua evoluzione dei nostri sistemi non sia stata una banale casualità ma un vero processo di ricerca, nel quale ogni innovazione è stata generata dal soddisfacimento di un'esigenza che ha

generato nuove esigenze fino ad arrivare ai nostri giorni.

Grazie a questo processo l'esigenza primordiale che originariamente era alla base di tutto, ovvero la catalogazione di documenti e del sapere al livello globale ha prima raggiunto il suo obiettivo e poi ha virato su un altro aspetto: l'abbattimento delle frontiere e dei confini; se prima la scrittura tramite lettera era l'unico modo di poter contattare e comunicare con una persona, con l'avvento dei social network era possibile creare rapporti con perfetti sconosciuti ubicati in altri stati, in tempo reale e senza ritardi nella comunicazioni.

Nel 2002 il primo social network mai creato si chiamava Friendster, nel quale un utente poteva creare un profilo e stringere amicizia con altri utenti sulla base di interessi comuni. Successivamente fu creato MySpace nel 2003, che consentiva agli utenti una maggiore personalizzazione del profilo e delle loro pagine, per poi arrivare nel 2004 al più grande social network mai creato e che ancora oggi detiene il record per il maggior numero di utenti registrati ed attivi, Facebook.[13]

Facebook ha rivoluzionato il concetto di social network conosciuto fino a quel momento, tramite esso, un utente può trasferire online le proprie amicizie, relazioni della vita reale e condividere con essi contenuti di tutti i tipi attraverso gli strumenti messi a disposizione dal sito. Ciò che fu creato in seguito come Twitter, Google+ o LinkedIn non ha fatto altro che scremare ulteriormente le comunità di utenti a seconda dei loro bisogni.

Con questa nuova forma di espressione libera e condivisione di contenuti, molte applicazioni web e sistemi di chat sono stati chiusi poiché ritenuti obsoleti, basti pensare alle ultime statistiche sull'uso dei social rispetto ai blog e alle altre applicazioni web utilizzate negli anni '90. [15]

- Più del 75% di tutti gli utenti di internet (3.176 miliardi) utilizzano i social media.
- Il 72% di tutti gli utenti di internet possiede un account su facebook.

- Facebook è ancora in espansione e si stimano circa mezzo milione di nuovi utenti al giorno.
- Youtube è usato dall'82% di utenti appartenenti alla fascia 18-49 anni.
- Reddit è la piu' grande community del mondo (anche se in Italia non esiste).

Da questi dati si comprende facilmente quanto ormai queste piattaforme siano nella vita di tutti noi, ma le nostre esigenze non si sono erano ancora fermate. [14]

Controllare la posta elettronica, i vari social e consultare internet per banali ricerche richiedeva comunque avere accesso ad una macchina dal quale potersi connettere, nacque così una nuova esigenza, poter compiere queste attività da qualunque parte del globo in qualsiasi momento.

Nel 2007 Steve Jobs presentò l'iPhone: un telefono portatile con la capacità di andare su internet e compiere molteplici funzioni mediate l'uso di applicazioni installabili sul dispositivo dette app. Nacquero così gli smartphone, aprendo una nuova frontiera nel mondo delle comunicazioni e dei social.

L'attuale situazione nel mondo della tecnologia è ulteriormente mutata rispetto al 2007, ora con il semplice ausilio di uno di questi dispositivi, un utente è perfettamente in grado di poter compiere ricerche, consultare i propri account social, effettuare pagamenti online consultando appositi store, mandare messaggi ed effettuare chiamate, anche senza sfruttare direttamente sms e minuti, bensì appoggiandosi direttamente ad una connessione di rete.

Per i viaggi di lavoro i computer portatili stanno venendo rimpiazzati sempre più dai tablet, più semplici e sottili da trasportare, inoltre nel campo degli accessori abbiamo assistito alla nascita degli indossabili e degli smartwatch, orologi in grado di comunicare con il nostro smartphone in collegamento bluetooth, consentendoci di controllare agevolmente le notifiche ed alcune app.

Anche il campo dei database e dello spazio online ha subito un ulteriore cambiamento;



Figura 1.5: Social Media

gli utenti avevano esigenza di avere un proprio recipiente sempre reperibile e che non fosse strettamente fisico. Questo portò alla nascita dei sistemi di Cloud, spazi virtuali dove un utente può memorizzare i dati più disparati su di un server, dal quale può, in qualunque momento, riscargarli, manipolarli ed addirittura metterli in condivisione con altri utenti.

Da quanto illustrato finora è facilmente comprensibile come i dati siano stati fondamentali nella storia umana e come il perseguire la strada per renderli sfruttabili ed accessibili a tutti, abbiano spinto l'intelletto umano a compiere invenzioni e innovazioni sempre più importanti in questo ambito.

1.5 Gli Open Data - I dati aperti

Come accennato all'inizio, gli open data sono dati liberi messi a disposizione da una persona, un ente, uno stato ecc. per tutti coloro che vogliono fruirne ma qual'è la legislazione a cui sono sottoposti?



Figura 1.6: Comprehensive Knowledge Archive Network

Per rispondere a questa domanda, dobbiamo indagare prima di tutto sulla forma di dato che più si avvicina a quella che ci interessa, ovvero i dati pubblici. Inizialmente questi ultimi erano soggetti ad una burocrazia molto restrittiva che molte volte rendeva difficile sia la loro consultazione che la loro comprensione.

Nel 2003 è stata approvata una direttiva europea che individuava nelle informazioni del settore pubblico un'importante materia prima per prodotti e servizi incentrati su contenuti digitali. [18]

Tale direttiva cercava di invitare gli stati a rendere i dati più comprensibili e fruibili da tutti, sfruttando gli enti pubblici e provvedendo a fornire licenze standard per il riuso di questi ultimi.

Le iniziative in questa particolare tematica sono sempre state molto scarse, anche in ambito online. Per poter avere un'idea precisa partiamo dagli anni 2000 e vediamo come si sono sviluppati ed evoluti, cominciando con il definire per bene il concetto di Open Government.

Per Open Government intendiamo un concetto di “governance” a livello centrale e locale, basato su modelli, strumenti e tecnologie che consentono alle amministrazioni di essere “aperte” e “trasparenti” nei confronti dei cittadini. [18]

Da questa definizione si comprende facilmente che tutte le attività dello stato devono essere disponibili al fine di aiutare coloro che ne fanno richiesta.

Originariamente l'open government risale a concetti elaborati in epoca illuminista da tesi politiche di Charles-Louis Montesquieu e Antonio Genovesi.

Nel 2009 gli Stati Uniti dichiararono la creazione di un livello senza precedenti di apertura del loro governo. Per fare ciò fu creato un portale chiamato data.gov, destinato a contenere qualsiasi informazione proveniente dagli enti statali in modo tale da avere una panoramica completa di tutto ciò che riguardava il paese in termini di beni, servizi, dati ambientali eccetera.

Nel 2011 in Italia viene creato un Vademecum sugli Open Data, destinato a tutti i dipendenti del settore pubblico (Amministratori, dirigenti, impiegati ecc.) e contenente, tutti i principi di Open Government e l'indicazione degli elementi necessari ad avviare il processo di apertura dei dati pubblici.

I principi dell'open Government di seguito riportati, fondano le loro basi su tre elementi:

- Government should be transparent.

La trasparenza favorisce e promuove la responsabilità fornendo ai cittadini le informazioni sulle attività dell'Amministrazione. Le amministrazioni sono quindi invitate a disporre le informazioni in modo da essere facilmente reperibili, riutilizzabili ed aperte.

- Government should be participatory.

La partecipazione dei cittadini alle scelte aumenta l'efficacia dell'azione amministrativa e migliora la qualità delle decisioni dell'amministrazione. Si invita pertanto a coinvolgere i cittadini nei processi decisionali per contribuire attivamente anche grazie alle

tecnologie di comunicazione attualmente disponibili e diffuse.

- Government should be collaborative.

La collaborazione prevede un coinvolgimento diretto dei cittadini nelle attività dell'amministrazione, tracciando uno scenario nuovo nelle dinamiche di relazione tra l'amministrazione e i diretti interessati.

I tratti distintivi dell'Open Government appaiono quindi la centralità del cittadino, l'amministrazione partecipata e collaborativa, la trasparenza, l'apertura e la condivisione.

Per attuare questi principi è necessario che ogni cittadino, sia in grado di disporre degli strumenti per poter valutare le decisioni prese dall'amministrazione. Per alcuni aspetti l'Open government può essere considerato l'evoluzione dell'e-government, cioè un processo cominciato alla fine degli anni '90, nel quale l'amministrazione si è dotata di tecnologie dell'informazione e della comunicazione.

La pubblicazione su internet dei dati provenienti da enti governativi è diventato un aspetto molto importante dell'open government, poiché con il fatto di essere riutilizzabili consentono a terzi di poterli rielaborare e sfruttare in determinati ambiti, questa stessa tesi ne è un esempio esplicativo.

L'apertura dei dati pubblici risponde dunque a molteplici esigenze:

- rendere l'amministrazione trasparente.
- migliorare la qualità della vita dei cittadini.
- da la possibilità di sviluppare economicamente i dati.
- incentiva pratiche di civic hacking.

1.6 Gli Open Data - I principi

- Gli Open Data - I principi

L'OKF (Open Knowledge Foundation) esprime il concetto di conoscenza aperta attraverso i seguenti principi: [19]

- **Disponibilità e Accesso:** i dati devono essere disponibili in un formato utile e modificabile, ad un prezzo ragionevole e preferibilmente mediante scaricamento da internet.
- **Riutilizzo e Ridistribuzione:** i dati devono essere forniti a condizioni che permettano il riutilizzo e la redistribuzione, con la possibilità di essere combinati ad altre basi di dati.
- **Partecipazione universale:** tutti devono essere in grado di usare, riutilizzare e ridistribuire i dati, anche per scopi commerciali, senza alcuna discriminazione.

Il formato dei dati per essere considerati aperti in base agli standard internazionali devono essere:

- **Completi:** i dati devono comprendere tutte le componenti per essere esportati, utilizzati on-line ed off-line, integrati e aggregati con altre risorse e diffusi in rete.
- **Primari:** le risorse digitali devono essere strutturate in modo tale che i dati siano presentati in maniera abbastanza granulare da poter essere utilizzati dagli utenti per diversi scopi, anche integrandoli con altri dati e contenuti in formato digitale.
- **Disponibili:** gli utenti devono poter accedere e utilizzare i dati presenti in rete in modo rapido e immediato con la massimizzazione del valore e dell'utilità derivanti dall'uso di queste risorse.
- **Accessibili:** devono essere resi disponibili al maggior numero possibile di utenti senza restrizioni, attraverso il protocollo HTTP, senza alcuna sottoscrizione di contratto, pagamento, registrazione o richiesta.

- Leggibili da computer: devono essere processabili in automatico dal computer per garantire agli utenti la massima libertà di accesso e di utilizzo.
- In formati non proprietari: la codifica deve essere aperta e pubblica, preferibilmente semplice e maggiormente supportata. Nessuno deve avere il controllo esclusivo dei dati.
- Liberi da licenze che ne limitino l'uso: l'utilizzo, la diffusione o la redistribuzione dei dati aperti non possono essere soggetti a limitazione di licenza.
- Riutilizzabili: gli utenti devono essere messi in condizione di poter riutilizzare e integrare i dati aperti, fino a creare nuove risorse, applicazioni e servizi di pubblica utilità.
- Permanenti: le peculiarità dei dati aperti devono caratterizzarli per l'intero ciclo di vita degli stessi.

1.7 Gli Open Data - La catalogazione

Il W3C ha proposto un modello di catalogazione per classificare gli Open Data in base alle loro caratteristiche su una scala di valori da 1 a 5 stelle: [20]

- Una Stella: è il livello più basso. Fanno parte di questa categoria quei file non strutturati (immagini, file.doc e .pdf). Indica semplicemente la disponibilità di un dato online in un formato qualunque ma distribuito in ogni caso in modo leggibile, stampabile dagli utenti, conservabile e pubblicabile.
- Due Stelle: indica che i dati sono strutturati ma codificati in un formato proprietario (ad esempio microsoft Excel). Due stelle indicano, oltre alla possibilità fornite



Figura 1.7: Modello 5-star

dai dati con una sola stella, la possibilità di effettuare elaborazioni su tali dati, a patto di disporre del software necessario a gestire un file codificato con un formato proprietario. I dati codificati con due stelle non sono ancora in un formato aperto in quanto per elaborarli è necessario un software proprietario. Tuttavia, essendo dati strutturati, possono essere convertiti in dati aperti.

- Tre Stelle: questo livello contiene dati strutturati e codificati in un formato non proprietario (es. csv). Tre stelle indicano la possibilità di effettuare elaborazioni sui dati senza essere costretti ad utilizzare software proprietario. E' il formato più semplice di dato aperto.
- Quattro stelle: indica dati strutturati e codificati in un formato non proprietario che sono dotati di un URI (Uniform Resource Identifier) che li rende indirizzabili sulla rete e quindi utilizzabili direttamente online attraverso l'inclusione in una struttura basata sul modello RDF (Resource Description Framework). Quattro stelle indicano che un singolo dato di un dataset disponibile online, in un formato aperto, può essere richiamato attraverso un URL specifico. Ciò consente di puntare

al dato, o ad un insieme di dati, da un'applicazione e di accedervi dall'interno di un programma che può poi elaborarlo in vari modi.

- Cinque stelle: indica i dati che vengono definiti Linked Open Data (LOD), ovvero quei dati aperti che, dal punto di vista del formato, presentano anche collegamenti ad altri dataset. In altri termini è possibile collegare dinamicamente tra loro più dataset, incrociando informazioni provenienti da fonti diverse, anche gestite da diverse Amministrazioni.

A questo punto parliamo di Linked Open Data per avere anche una panoramica sulle regole che li rappresentano. Tali regole sono state espresse da Tim Berners-Lee in una conferenza del 2009 facendo notare come i dati di per sè non forniscono un'applicazione immediata, tuttavia giocano un ruolo importantissimo nelle nostre vite.

I Linked Data secondo Lee non sono una serie di dati messi in relazione, ma un mondo in cui tutti abbiamo caricato dei dati sul web, e dunque nel quale è possibile ritrovare qualsiasi informazione immaginabile.

Le tre regole che rappresentano i Linked Data sono:

- La prima ridefinizione del concetto di URL come indicatore, non soltanto di documenti, ma di cose di cui i documenti parlano, ovvero le persone, i luoghi, i prodotti, gli eventi. Ogni sorta di concetto sarebbe stato individuato attraverso un URL.
- La seconda regola è il formato standard dei dati, fondamentali per un riutilizzo da parte di un soggetto interessato ad essi.
- La terza, la capacità dei dati di creare relazioni. Ogni dato è relazionato a qualche altro dato e così via.

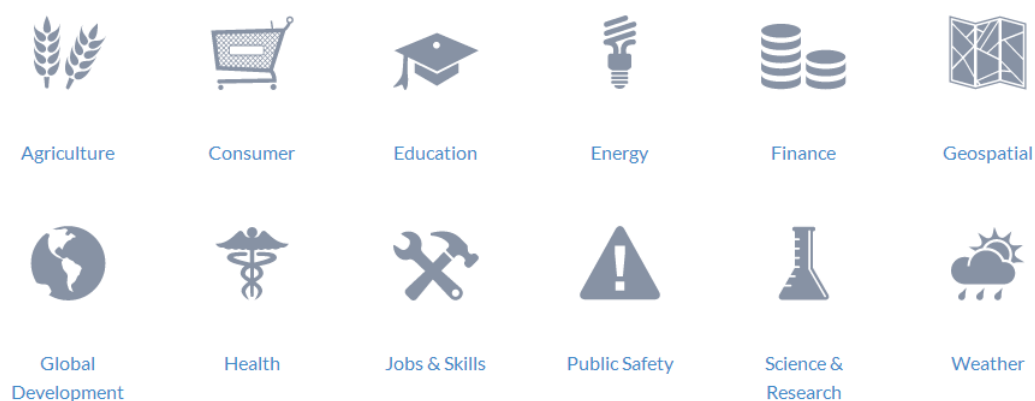
1.8 Gli Open Data - Esempi pratici & competitors

Al fine di raccogliere informazioni da rilasciare al pubblico si sono diffusi numerosi portali e siti web:

Data.gov

E' stato il primo portale per Open Data lanciato il 5 marzo del 2009 dall'amministra-

BROWSE TOPICS



MORE TOPICS



Figura 1.8: Data.gov

zione degli stati uniti d'America, in particolare dal neo assunto Vivek Kundra. Il sito ha introdotto la filosofia dei dati aperti in formato digitale.

Data.gov è cresciuto da 47 gruppi di dati al momento del lancio ad oltre 180.000.

DBPedia

Fu uno dei primi esempi di Linked Open Data, citato da Berners-Lee stesso nella conferenza del 2009. DBPedia è stato creato da Chris Bizer della Freie Universitat di Berlino che, notando le potenzialità e la mole di informazioni possedute da Wikipedia, decise di estrapolare questi dati e di metterli in un insieme di dati linkati sul web.

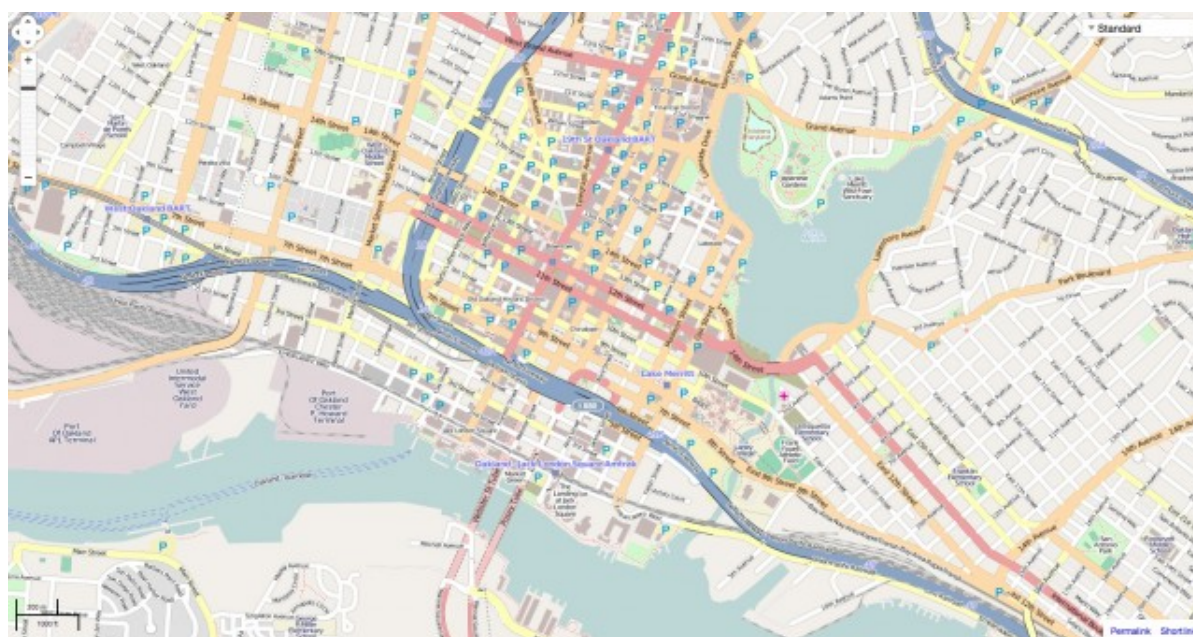


Figura 1.9: OpenStreetMap Example

In questo modo quando si cerca un'informazione in DBPedia, si può accedere facilmente a tanti altre tipologie di informazioni collegate a quella che stiamo consultando. Il formato di output dei Linked Open Data di DBPedia è l'RDF (Resource Description Framework)

OpenStreetMap

Il vantaggio dei Linked Open Data è la grande portata di informazioni e di contenuti che possono essere generati con il contributo di ogni singolo utente. OpenStreetMap altro non è che un progetto collaborativo la cui finalità è quella di creare mappe a contenuto libero, al fine di incoraggiare la crescita e lo sviluppo dei dati Geospaziali disponibili, utilizzabili e condivisibili da chiunque. L'esigenza di questi dati nasce fondamentalmente dal fatto che, a dispetto delle credenze comuni, la maggior parte delle mappe che si credono liberamente utilizzabili, hanno invece restrizioni legali o tecniche al loro utilizzo e ciò ne impedisce il libero utilizzo.

Dati.gov.it

Dopo l'introduzione di Dati.gov creato negli stati uniti, anche negli altri paesi si cercò di creare un sistema analogo, atto a rendere i propri dati governativi liberi e fruibili a tutti i cittadini.

Nell'ottobre del 2011 in Italia venne lanciato il portale dati.gov.it nell'ottica di un

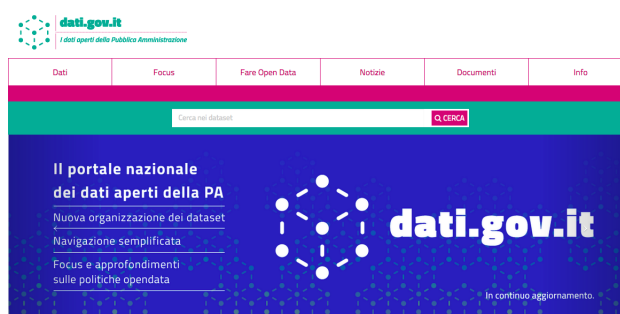


Figura 1.10: Dati.gov.it Interface

processo di gestione degli Open Data italiani, raccogliendo in un dataset aggiornato periodicamente i dati catalogati sul sito, rendendoli interpretabili attraverso un'infografica.

Istat

In Italia l'ente che rilascia il maggior numero di dataset consultabili risulta essere l'Istat, che mette a disposizione dei cittadini, delle imprese e delle istituzioni i risultati delle rilevazioni sui vari aspetti economici sociali, territoriali ed ambientali.

Le principali rilevazioni che questo ente compie sono:

- Censimenti sulla popolazione
- Censimenti sull'industria, sui servizi e sull'agricoltura



Figura 1.11: An Example of Istat Graphic

- Indagini campionarie sulle famiglie (consumi, forze di lavoro, aspetti della vita quotidiana, salute, sicurezza, tempo libero, famiglia e soggetti sociali, uso del tempo ecc.)
- Indagini economiche (contabilità nazionale, prezzi, commercio estero, imprese, occupazione ecc.)

Tale istituzione venne creata con il nome di Istituto Centrale di Statistica nel 1926 durante il periodo fascista. Dopo il decreto legislativo del 6 settembre 1989 è stato riorganizzato e rinominato Istituto Nazionale di Statistica.

Da notare che tutti i contenuti provenienti e pubblicati dall'ISTAT sono disponibili sotto licenza Creative Commons.

Comune di Bologna

Provenienti sempre dai dati ricavati sul portale data.gov.it è chiaro come anche comuni, regioni e province si occupino di raccogliere dati liberi. Il comune di Bologna, nello specifico, ha rilasciato 591 dataset ed è secondo solo al comune di Firenze per mole di dati con 610 insiemi di dati distribuiti.

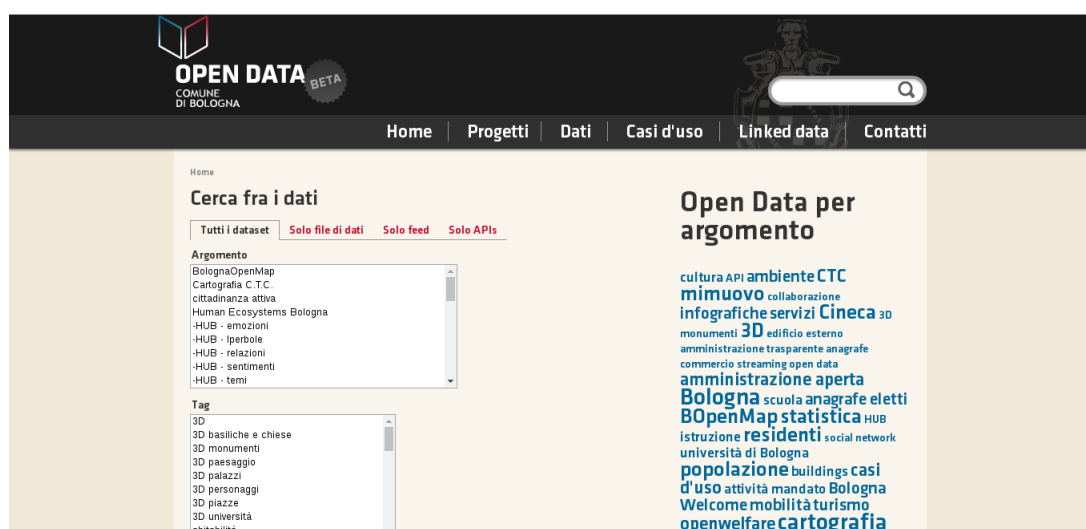


Figura 1.12: Dataset Bologna

I dati del comune di Bologna vengono rilasciati dal sito <http://dati.comune.bologna.it>. Se è vera l'espressione che afferma che i dati acquisiscono valore in base all'uso di cui se ne fa, allora non si può non riconoscere un merito ai servizi offerti dal comune di Bologna e generati dallo sfruttamento di questi dati:

1.9 I dati ambientali

Questa parte mi riguarda personalmente in quanto la mia tesi si basa proprio su questo tipo di dati. Ogni regione dell'Italia gestisce in modo autonomo i propri dataset di dati ambientali, adottando ognuno la propria strategia sul formato e sui metodi di rilascio, ma fanno tutti parte di unico macro-ente chiamato Arpa (agenzia regionale per la protezione ambientale).

Queste agenzie sono state istituite nel 1994, inizialmente come 100 PMP sparsi su tutto il territorio italiano e successivamente il potere fu gestito da 19 agenzie regionali e 2 delle province autonome.

Le principali funzioni compiute da Arpa sono le seguenti:

- Controllo di fonti e fattori di inquinamento dell'aria, dell'acqua, del suolo, acustico ed elettromagnetico.
- Monitoraggio delle diversi componenti ambientali: clima, qualità dell'aria, delle acque, caratterizzazione del suolo, livello sonoro dell'ambiente.
- Controllo e vigilanza del rispetto della normativa vigente e delle prescrizioni contenute nei provvedimenti emanati dalle Autorità competenti in materie ambientali;
- Supporto tecnico-scientifico, strumentale ed analitico agli Enti titolari con funzioni di programmazione e amministrazione attiva in campo ambientale (Regioni, Provincie e Comuni);
- Sviluppo di un sistema informativo ambientale che sia di supporto agli Enti istituzionali e a disposizione delle organizzazioni sociali interessate.

I dati ambientali sono quelli di più alto interesse per le persone, poiché rappresentano e descrivono una serie di fattori riconducibili alla loro salute e benessere. Sono state create numerose applicazioni per la loro visualizzazione, qui vediamo alcuni esempi:

Aircheck

Lo scopo di aircheck è fornire informazioni aggiornate sulla qualità dell'aria in Svizzera e nel principato del Liechtenstein a livello informativo, suggerendo comportamenti a cui attenersi a scopo preventivo.

Le principali caratteristiche di quest'app sono:

- Informa sulla qualità dell'aria corrente presso la sede o in qualsiasi altro sito arbitrario in Svizzera e Lichtenstein.
- Mostra l'attuale indice di inquinamento dell'aria e le concentrazioni di particelle , ozono e biossido di azoto sottili.
- consente l'accesso aperto alle ultime misurazioni delle stazioni di monitoraggio cantonali e la rete di monitoraggio nazionale.
- Offre informazioni di base sulla formazione e le fonti dei principali inquinanti atmosferici.
- consiglia come è possibile proteggersi in caso di inquinamento atmosferico elevato.



Figura 1.13: Aircheck

ArpaUmbria

Un altro esempio di dati ambientali regionali è l'applicazione della agenzia regionale per la protezione ambientale dell'umbria.

Le sue funzionalità sono:

- Bollettino quotidiano della qualità dell'aria in Umbria.
- Previsioni quotidiane sull'andamento degli inquinanti PM10 e Ozono.
- Bollettino settimanale dei pollini.
- Sistema di ricerca cartografica delle stazioni di rilevamento.
- Tra le altre features mostra anche lo stato di balneabilità dei laghi, qualità delle acque potabili, informazioni sui campi elettromagnetici.



Figura 1.14: Arpaumbria

Arpat

L'agenzia regionale per la protezione ambientale della toscana ha sviluppato un'applicazione di visualizzazione di dati ambientali a livello regionale.

Le principali features di quest'app sono le seguenti:

- Stato delle acque di balneazione.
- Percentuale di presenza dell'alga tossica *Ostreopsis Ovata*.
- Qualità dell'aria e dei livelli di ozono.
- Percentuale di pollini e altri allergeni.



Figura 1.15: Arpat

ArpaVeneto

Anche L'Arpav (Agenzia Regionale per la Prevenzione e Protezione Ambientale del Veneto) ha realizzato la propria applicazione Mobile per i dati ambientali.

Tale applicazione mostra le seguenti informazioni:

- Previsioni Meteo.
- Balneazione (stato delle acque).
- Livelli idrometrici.
- Pollini.
- Qualità aria.

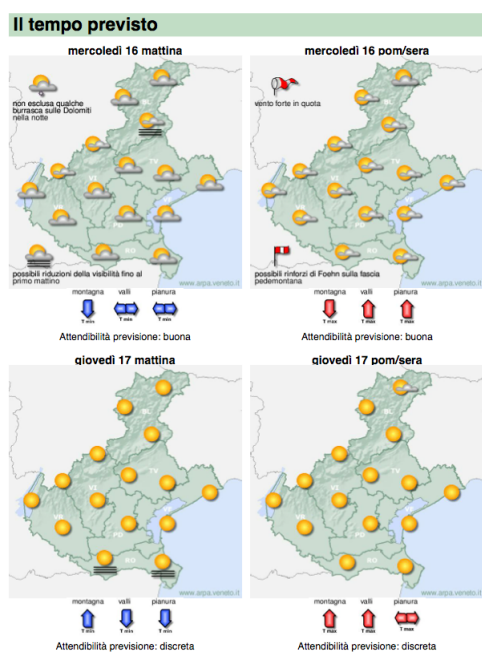


Figura 1.16: Arpaveneto

Capitolo 2

Progettazione dell'app

Il mio lavoro in questa tesi è suddiviso in due parti, nella prima mi sono concentrato sulla realizzazione di un Web Service, che estraesse ed analizzasse i dati resi disponibili dall'ente regionale per i dati ambientali dell'Emilia Romagna, ArpaER. Nella seconda ho cercato di renderla funzionale, sfruttando i dati derivanti da quest'ultimo, per realizzare un'applicazione mobile che mettesse in pratica quanto illustrato finora, ovvero un'app che ho chiamato ER-Pollen.[27]

ER-Pollen è un'applicazione mobile sviluppata su piattaforma Android, che permette di visualizzare i dati ambientali estratti dal Web Service e resi disponibili, in formato aperto dall'ARPA (Agenzia Regionale per la Protezione Ambientale) relativi alla regione Emilia Romagna.

Bisogna precisare che, benché i dati siano disponibili da qualche anno, il loro sistema informativo ed il formato dei loro dati è in continuo mutamento.

2.1 Architettura

L'architettura del mio progetto è riassumibile nel seguente schema:

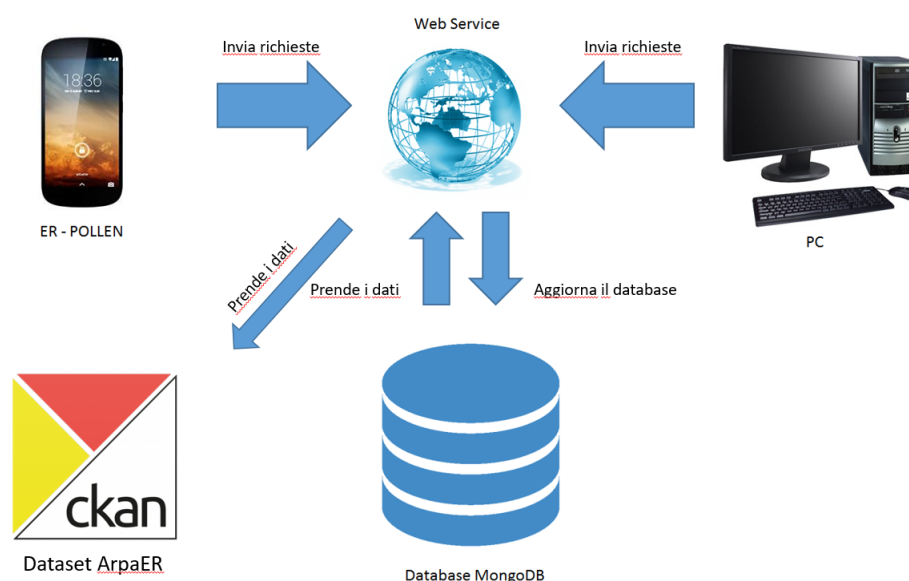


Figura 2.1: Architettura del progetto

Come si può facilmente notare il cuore del progetto è il Web Service, in grado di comunicare con l'app, i dataset dell'arpa, il proprio database mongodb e i pc. Ho scelto di sviluppare il Webservice in linguaggio Python per via del gran numero di librerie disponibili e la sua versatilità; le librerie includevano già tutto quello che occorreva per compiere http request e comunicare con un database MongoDB.

Quest'ultimo è stato scelto per via del formato di output dei dataset dell'Arpa e cioè il JSON, scegliendo un database che opera e lavora dati di quel formato ho eliminato inutili processi di parsing.

Il WS in è in grado di manipolare e fornire i dati in formato JSON. La scelta di crearlo standalone anziché integrarlo direttamente nella mia app è finalizzata a rendere disponibile questo strumento a futuri sviluppatori, in modo da poter continuare e rendere più semplice la loro elaborazione e manipolazione.[28]

Per quanto riguarda l'app, ho deciso di scegliere Android come linguaggio di programmazione in quanto l'omonimo sistema operativo è molto più diffuso, gratuito e

meno severo per quanto ne concerne l'immissione sul mercato rispetto ad IO'S.

ER-Pollen nasce per essere utilizzata sia da persone comuni, che da utenti esperti, rappresentando i concetti enunciati finora sul fatto che i dati acquisiscono valore a seconda del modo in cui se ne fa uso.

Arpa, per aiutare gli sviluppatori e coloro che vogliono usare i suoi dati, fornisce agli utenti un post processor a livello server, che traduce i dati di formato binario, in formato JSON, rispondendo alle richieste con i dati già tradotti, quindi più facilmente elaborabili ed interpretabili dal Webservice e di conseguenza dall'applicazione.

2.2 CKAN

CKAN

la piattaforma utilizzata da ArpaER per il rilascio dei dati è CKAN (Comprehensive Knowledge Archive Network), un sistema Open Source basato sul Web nel quale un utente può caricare i suoi dati specificandone il formato e sfruttando le API messe a disposizione per mostrare agli altri utenti delle preview del proprio lavoro.[23] [24]

CKAN è ispirato dal sistema di gestione dei pacchetti, utilizzato da alcuni sistemi operativi open source come Linux. Attualmente la versione di sviluppo di CKAN è la 2.2 rilasciata a febbraio di quest'anno ed è in continuo miglioramento.

I dataset disponibili su CKAN per quello che riguarda ArpaER si trovano all'indirizzo:

<http://ckan.arpa.emr.it/dataset>

da cui è possibile selezionare, visualizzare i dati e scaricarli attraverso delle richieste HTTP, nel mio caso compiute dal Webservice.

Il tipo di dati di cui mi sono interessato per il mio progetto sono quelli inerenti allo stato dell'aria e quello dei pollini sul territorio dell'Emilia Romagna, disponibili nei dataset a questi link:

Pollini:

http://ckan.arpa.emr.it/api/action/datastore_search?resource_id=c08df772-8c32-4f11

Aria:

http://ckan.arpa.emr.it/api/action/datastore_search?resource_id=301ba629-da56-49e3

2.3 MongoDB

MongoDB

Il WebService effettua delle chiamate GET ai dataset su CKAN, e prende i dati a seconda della richiesta che viene inoltrata. Tuttavia il WebService, per poter svolgere la funzione richiesta, necessitava di uno spazio per immagazzinare l'output delle richieste; doveva cioè agganciarsi ad un sistema di basi di dati, con lo scopo di memorizzare e tenere pronte all'uso tutte le informazioni. [21] [22]

La mia scelta per questo passaggio è ricaduta su MongoDB. Questo sistema di basi di dati è un DBMS non relazionale, orientato ai documenti. A differenza degli altri database esso si allontana radicalmente dalla struttura tradizionale basata su tabelle dei database relazionali, in quanto classificato come database NoSQL.

Questa differenziazione lo rende un database nettamente dinamico rispetto agli altri, in quanto è possibile gestire i dati in formato JSON, rendendo l'integrazione di dati di alcuni tipi di applicazione più facile e veloce; inoltre va specificato che è un tipo di

database open source quindi si presta benissimo a qualsiasi utilizzo in materia di sviluppo.

MongoDB deriva dalla parola “humongous” che significa enorme ed è stato sviluppato inizialmente dalla società di software 10gen (poi rinominata MongoDB inc.) nell'ottobre del 2007 come un componente di un prodotto di platform as a service.

Successivamente l'azienda si è spostata verso un modello di sviluppo open source nel 2009 offrendo un supporto commerciale ed altri servizi. Da quel momento MongoDB è stato adottato come backend da numerosissimi siti web, divenendo tra gli altri, il più popolare database NoSQL.

Le caratteristiche principali di questo sistema di basi di dati sono:

- Query ad hoc

MongoDB supporta ricerche per campi, intervalli e regular expression. Le query possono restituire campi specifici del documento ed includere funzioni definite dall'utente in Javascript.

- Indicizzazione

qualunque campo in MongoDB può essere indicizzato, tradizionalmente ai RDBMS, inoltre sono disponibili anche indici secondari, indici unici, indici sparsi, indici geospaziali e indici full text.

- Alta Affidabilità

MongoDB fornisce alta disponibilità e aumento del carico gestito attraverso i replica set. Un replica set consiste in due o più copie dei dati. Ogni replica può avere il ruolo di copia primaria o secondaria in qualunque momento. Qualora una replica primaria fallisca, il replica set inizia automaticamente un processo di elezione per determinare quale

replica secondaria deve diventare primaria. Le copie secondarie possono anche effettuare letture, con dati eventualmente consistenti di default.

- File Storage

MongoDB può essere usato anche come un file system, traendo vantaggio dalle caratteristiche di replicazione e bilanciamento su più server per memorizzare file.

- Aggregazione

MongoDB supporta due modalità di aggregazione dei dati: il MapReduce e l'Aggregation Framework. Quest'ultimo lavora come un pipeline e permette di ottenere risultati molto più rapidamente del MapReduce grazie all'implementazione in C++.

- Capped Collection

MongoDB supporta collection a dimensioni fisse chiamate capped collection. Questo tipo di collection mantengono l'ordine di inserimento e una volta raggiunta la dimensione definita, si comportano come code circolari.

Come abbiamo già accennato precedentemente MongoDB offre una serie di consulenze e servizi supplementari qualora l'utente decida di abbonarsi al sito tra cui opzioni per aumentare la sicurezza dei database, assistenza SNMP, certificazioni SO e molto altro ancora; va inoltre accennata l'interessante opzione di training e formazione del personale.

Capitolo 3

Implementazione - parte 1

3.1 Il Web Service

In questa sezione ci interesseremo del funzionamento del Web Service. la definizione di Web Service, rilasciata dal World Wide Web Consortium, altro non è che un sistema software progettato per supportare l'interoperabilità tra diversi elaboratori su di una medesima rete, ovvero in un contesto distribuito.[27] [28]

In questo caso, il Web Service esegue 3 funzioni principali:

- Ascolta: il WS resta in ascolto su una porta in attesa di ricevere richieste da un utente.
- Scarica: il WS si collega al database dell'ArpaER, scarica i dati delle rilevazioni 1 volta al giorno, li processa e li memorizza database.
- Invia: una volta ricevuta la richiesta, il WS effettua una query sul database MongoDB filtrando le informazioni e restituisce all'utente i dati richiesti.

3.2 I dati

la connessione a MongoDB avviene con il seguente codice:

```
1 from flask import Flask
2 import json
3 from pymongo import MongoClient
4 import urllib.request as ur
5 from bson import ObjectId
6
7 app = Flask(__name__)
8
9 try:
10     client = MongoClient('localhost', 27017)
11     print("ho stabilito la connessione a mongodb")
12     db = client.dati_arpaeer #connessione al db
13     print("accesso effettuato al database")
14
15 except:
16     print("problemi nella connessione mongodb")
```

Figura 3.1: Python Connection to MongoDB

le richieste al Web service vengono lanciate attraverso dei file Python, che utilizzano request per ottenere i dati che ci interessano. Il Web Service, ricevuta la risposta dai file Python, esegue una query sul database, nel quale ha memorizzato tutti i dati dell'aria e dei pollini e restituisce al client i risultati desiderati. Vediamo cosa succede con i dati sui pollini.[29]

```
"help": "http://ckan.arpa.emr.it/api/3/action/help_show?name=datastore_search",
"success": true,
"result": {
"resource_id": "c08df772-8c32-4f11-8e24-5782fa9aaaf",
"fields": [],
"records": [],
"_links": {
"start": "/api/action/datastore_search?limit=5&resource_id=c08df772-8c32-4f11-8e24-5782fa9aaaf",
"next": "/api/action/datastore_search?offset=5&limit=5&resource_id=c08df772-8c32-4f11-8e24-5782fa9aaaf",
}
}
```



```
"total":3328
}
}
```

- help: contiene un link dove è possibile consultare tutti i comandi per muoversi all'interno del dataset.
- Success: verifica che la richiesta sia andata a buon fine e il PostProcessor abbia portato correttamente a termine il suo lavoro.
- Result: è un oggetto di tipo Array che contiene effettivamente le informazioni che ci interessano.

L'array Result contiene:

- resource_id: è un campo che mostra l'id della risorsa richiesta
- fields: è un oggetto di tipo Array che contiene la dichiarazione di tutti i campi sottostanti ed il loro formato.
- Records: è un oggetto di tipo Array che contiene tutte le misurazioni effettuate e registrate nel database a seconda.
- Links: contiene due campi: Start e Next che altro non sono che i collegamenti alle risorse a cui si è fatta la richiesta.
- Limit: eventuale limitazione ai valori che abbiamo richiesto (ad esempio limit : 5 sta ad indicare che chiediamo solo gli ultimi 5 risultati).
- Total: è il totale del numero di risultati.

```
"records":[
{
"ident":null,
"network":"pollini",
"lon":970554,
```

```
"date": "2015-09-17T00:00:00",  
"version": "0.1",  
"lat": 4506052,  
"_id": 177,  
"data": []  
},
```

L'array Records contiene:

- ident : rappresenta l'identificativo opzionale della stazione.
- network : nome della rete da cui proviene la stazione.
- lon : rappresenta la longitudine della rilevazione.
- date: rappresenta la data in cui è stata fatta la rilevazione.
- version: è la versione dell'algoritmo che prende i dati.
- lat: rappresenta la latitudine della rilevazione.
- _id: rappresenta l'id della rilevazione.
- data: è un array contenente tre valori statici che sono rispettivamente il nome della stazione, l'altezza della stazione e l'altezza barometrica della stazione.

Simile è la risposta del server alla richiesta dei dati dell'aria che sono pertanto così strutturati:

```
{  
"help": "http://ckan.arpa.emr.it/api/3/action/help_show?name= datastore_search",  
"success": true,  
"result": {},  
}
```

- help: contiene un link dove è possibile consultare tutti i comandi per muoversi all'interno del dataset.
- Success: verifica che la richiesta sia andata a buon fine e il PostProcessor abbia portato correttamente a termine il suo lavoro.
- Result: è un oggetto di tipo Array che contiene effettivamente le informazioni che ci interessano.

L'array Result contiene:

- resource_id: è un campo che mostra l'id della risorsa richiesta.
- fields: è un oggetto di tipo Array che contiene la dichiarazione di tutti i campi sottostanti ed il loro formato.
- Records: è un oggetto di tipo Array che contiene tutte le misurazioni effettuate e registrate nel database a seconda.
- Links: contiene due campi: Start e Next che altro non sono che i collegamenti alle risorse a cui si è fatta la richiesta.
- Limit: eventuale limitazione ai valori che abbiamo richiesto (ad esempio limit : 5 sta ad indicare che chiediamo solo gli ultimi 5 risultati).
- Total: è il totale del numero di risultati.

```
"result":{  
"resource_id":"301ba629-da56-49e3-94b6-d3b815229ca8",  
"fields":[]
```

```
"records":[],
"_links":
"start":":":"/api/action/datastore_search?resource_id=301ba629-da56-49e3-94b6-d3b8152
"next":":":"/api/action/datastore_search?offset=100&resource_id=301ba629-da56-49e3-94
},
"total":1243161
}
```

L'array Records contiene:

- variable_id: rappresenta il parametro a cui si riferisce la rilevazione.
- value: rappresenta il valore della misurazione.
- _id: rappresenta l'id della rilevazione.
- reftime: rappresenta la data in cui è stata fatta la rilevazione.
- Station_id: rappresenta l'id della stazione.

```
"records":[
{
"variable_id":7,
"value":44.0,
"_id":1008358,
"reftime":":2016-04-13T13:00:00",
"station_id":6000031
},
]
```

Le richieste a cui il Webservice risponde sono le seguenti:

/Aria: richiesta di tipo GET che restituisce tutti i dati dell'aria presenti nel database
/Pollini: richiesta di tipo GET che restituisce tutti i dati dei pollini presenti nel database.

Tutte le richieste che il client fa al Webservice fanno sì che quest'ultimo si colleghi al database MongoDB, contenente i dati interessati e li restituisca nel modo desiderato attraverso una query ben precisa.

Nel caso del Webservice usato in standalone, il richiedente riceverà nel proprio terminale i dati richiesti in formato JSON, tuttavia nel caso si possieda ER-Pollen, molto semplicemente i dati verranno restituiti all'app che li riceverà, li interpreterà e li restituirà sottoforma di zone colorate all'interno della mappa, con un colore che identificherà lo status delle zone esaminate.

Utilizzare le API di Google Maps all'interno di Android o meccanismi di geolocalizzazione, è diventato uno standard per tantissime app nei più svariati ambiti di utilizzo (Facebook, Foursquare, BlaBlaCar, UbiSafe, LifePro360 ecc.).

Una API (application programming interface) è un insieme di procedure disponibili al programmatore, di solito sono raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito, all'interno di un certo programma. Spesso e volentieri con questa terminologia si intende indicare una libreria software disponibili in un certo linguaggio di programmazione.

Google mette a disposizione le proprie API agli sviluppatori, a patto che siano registrati e che facciano richiesta di una API key per poter usufruire appieno dei suoi strumenti.

E' inoltre possibile accedere ad ulteriori contenuti, guide, assistenza e tantissimo altro, attivando il piano premium di google maps.

Capitolo 4

Implementazione - parte 2

4.1 Android

Android è un sistema operativo per dispositivi mobili sviluppato da Google Inc e basato sul kernel Linux; dalla sua nascita sono state sviluppate versioni ufficiali, custom rom (ovvero versioni personalizzate da utenti o aziende produttrici di telefonia), versioni trial e versioni developer.[30]

Le potenzialità di questo sistema operativo sono molteplici, a partire dal semplice market che fornisce app sviluppate da programmatori di tutto il mondo e scaricabili gratuitamente nella maggior parte dei casi, fino ad arrivare ad applicazioni che interagiscono con la sensoristica del telefono, permettendo così all'utente, il massimo dell'interazione con il device.

Il costante sviluppo delle varie versioni ha portato anche all'aumento del numero di features disponibili (la compatibilità con HTML5, ritenuto ormai il nuovo standard del web, nonché la compatibilità con i nuovissimi supporti wearable che stanno uscendo in questo periodo). La versione attualmente più usata è la 6.0 - 6.1.0 Marshmallow ma è da poco uscita la nuova versione 7.0 Nougat e verso dicembre è prevista l'uscita della 8.0 Oreo, dalla quale si vocifera si avranno moltissime novità per quello che riguarda i supporti esterni al semplice smartphone.

4.2 Implementazione dell'app

La seconda metà della mia fase di implementazione si è concentrata sullo sviluppo di un app, che si appoggiasse al Web Service ed elaborasse i suoi dati per visualizzarli attraverso le API di google maps.

Er pollen è un applicazione mobile che sfrutta le API di google maps, per integrare nella schermata principale, una mappa focalizzata sulla regione Emilia Romagna. All'avvio dell'app la mappa caricherà automaticamente lo status generale dell'aria a seconda di quelli che sono i dati che ha elaborato. È inoltre possibile monitorare lo stato dei pollini oppure visualizzare nello specifico le rilevazioni dell'aria premendo gli appositi pulsanti SO₂ (biossido di zolfo), PM₁₀ (polveri sottili) oppure CO (monossido di carbonio).

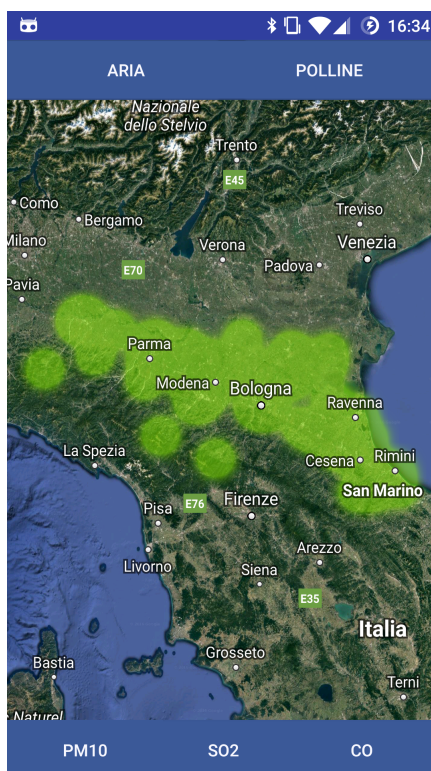


Figura 4.1: ER-POLLEN - stato dell'aria del 20/09/2016

Il funzionamento dei bottoni è semplice:

ogni pulsante invia un http request al Web service. Non appena la richiesta arriva, il Web service lancia delle query al database ed estrae i dati richiesti inviandoli poi all'app;

quest'ultima elabora i dati e colora le zone della mappa a seconda della rilevazione, in tre colorazioni possibili:

- Verde per indicare una rilevazione positiva dello stato dell'aria o dei pollini.
- Gialla per una rilevazione media ma comunque accettabile.
- Rossa per identificare una situazione dove la rilevazione dell'aria e dei pollini potrebbe costituire un pericolo per chi dovesse trovarsi in quella zona.

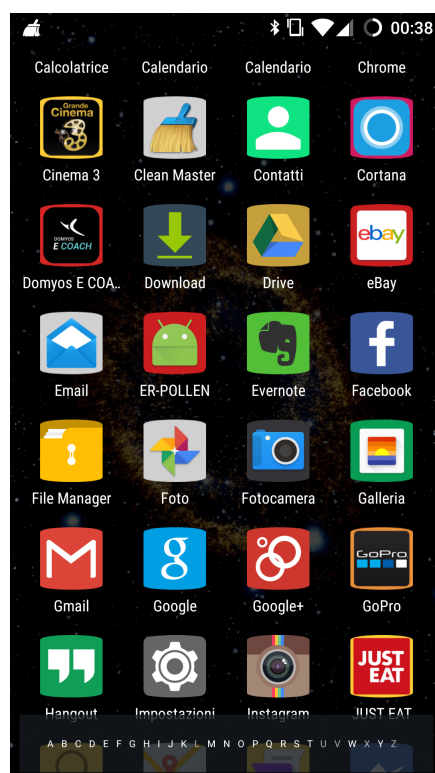


Figura 4.2: ER-POLLEN - icona dal menù app

L'http request viene lanciata tramite questo codice:

```
protected Object doInBackground(URL... urls) {
    final String[] result = new String[100];
    try {
        URL url = new URL("http://192.168.1.102:5000/Aria");
        HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
        urlConnection.connect();
        BufferedReader in = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));
        String inputLine;
        while ((inputLine = in.readLine()) != null)
        {
            JSONObject obj = new JSONObject(inputLine);
            JSONArray arr = obj.getJSONObject("result").getJSONArray("records");
            for (int i = 0; i < arr.length(); i++)
            {
                String _id = arr.getJSONObject(i).getString("_id");
                double value = arr.getJSONObject(i).getDouble("value");
            }
        }
    }
}
```

Figura 4.3: Http request

dove grazie al metodo `HttpURLConnection`, viene creata la connessione con il Web Service che riceve l'http request; successivamente viene creato un buffer che effettua la lettura di ciò che il web service invia come risposta e salva in un array l'output, dal quale andremo infine a ripescare le informazioni utili ad analizzare le rilevazioni.

L'interfaccia grafica è stata realizzata utilizzando le utility di google maps, in particolari le librerie che vengono utilizzate solitamente per generare le mappe che misurano l'intensità dei dati (solitamente il calore), comunemente note come heatmaps.

Una Heatmap è una rappresentazione grafica di dati, dove i singoli colori contenuti in una matrice (nel nostro caso in un array) sono rappresentati da colori. Il termine fu coniato da Cormac Kinney nel 1991, per descrivere il suo software che forniva una visualizzazione 2d raffigurante informazioni in tempo reale sui mercati finanziari. Successivamente anche altri programmatori cominciarono ad includere questo sistema di visualizzazione grafica come Jacques Bertin e Leland Wilkinson (ad oggi detentore del primo programma per la produzione di mappe di calore con grafica ad alta risoluzione a colori a grappolo).

In questo caso la heatmap viene generata con le utility messe a disposizione per gli sviluppatori android. Per poter inserire le rilevazioni ed associarle alla stazione a cui sono riferite ho creato il metodo `readItems()`. Questo metodo non fa altro che estrarre da un

```
public ArrayList<LatLng> readItems() {
    ArrayList<LatLng> coordinatesStaz = new ArrayList<>();
    try {
        InputStream is = getResources().openRawResource(R.raw.stazioni);
        BufferedReader in = new BufferedReader(new InputStreamReader(is));
        String inputStaz = "", temp;
        while ((inputStaz = in.readLine()) != null) {
            JSONObject json = new JSONObject(inputStaz);
            JSONArray arr = json.getJSONArray("records");
            for (int i = 0; i < arr.length(); i++) {
                String LatString = arr.getJSONObject(i).getString("lat");
                String LngString = arr.getJSONObject(i).getString("lon");
                Integer id = arr.getJSONObject(i).getInt("_id");
                double lat, lng;
                lat = Double.parseDouble(LatString);
                lng = Double.parseDouble(LngString);
                LatLng station = new LatLng(lat, lng);
                dictionary.put(station, id);
                coordinatesStaz.add(station);
            }
        }
    } catch (IOException ex) {
        ex.printStackTrace();
        return null;
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return coordinatesStaz;
}
```

Figura 4.4: `readItems()`

file json tutte le informazioni inerenti alle stazioni (a noi interessano solo le coordinate e l'id della stazione). Queste informazioni vengono poi salvate in un dizionario Java dove viene creata l'associazione chiave valore id-coordinate. Questo è un passaggio cruciale in quanto la tabella delle rilevazioni contiene oltre alle informazioni della rilevazione, l'id della stazione di riferimento, quindi grazie a questo dizionario ogni rilevazione viene associata alla stazione corrispondente ed al momento della creazione della heatmap ogni rilevazione viene posizionata ed associata nel modo corretto.

Le rilevazioni vengono analizzate secondo un meticoloso processo di smistamento. Anzitutto viene fatta una Http request al web service per ottenere tutte le informazio-

ni sulle rilevazioni e viene inoltre creato un dizionario per ogni tipo di rilevazione che verrà riempito man mano che vengono analizzate le singole rilevazioni. Queste rilevazioni vengono analizzate in un ciclo While dove, per ognuna, viene prelevato l'identificativo della stazione dove è stata misurata, il suo tipo ed il valore della rilevazione. In base al tipo della rilevazione il suo valore verrà inserito all'interno del dizionario corrispondente, avente come chiave la stazione e come valore la lista di tutte le misurazioni trovati avvenute per quel tipo in quella stazione. In questo modo, al termine dell'analisi di tutte le rilevazioni ci ritroveremo ad avere per ogni dizionario una categorizzazione delle misurazioni in base alle stazioni in cui sono state rilevate, quindi ogni stazione avrà la sua lista di misurazioni per quel tipo, in quel giorno.

Questo ci permette di poter facilmente individuare la media dei valori delle misurazioni per ogni singola stazione e quindi identificare il valore medio giornaliero della componente analizzata. Dopo aver trovato tutti i valori medi giornalieri di tutti i componenti possiamo procedere ad analizzare la qualità dell'aria, considerando come valore finale, la componente che avrà il livello giornaliero pessimo tra tutte. È stato deciso quest'approccio in quanto, ad esempio, se il livello di biossido di zolfo è verde e l'ozono è giallo, la situazione globale dell'aria non potrà avere colore verde.

```
// Add the tile overlay to the map.
TileOverlay mOverlay = mMap.addTileOverlay(new TileOverlayOptions().tileProvider(mProvider));

mProvider.setOpacity(0.5);
mProvider.setRadius(50);
mOverlay.clearTileCache();
```

Figura 4.5: metodo che crea l'overlay per colorare la mappa

Dopo aver concluso il processo di analisi dei dati in questione, si potrà procedere con la loro immissione sulla nostra mappa, mediante l'uso delle utility di google maps e nello specifico delle heatmaps.

Il processo di creazione della heatmap segue questi passi. Le coordinate di ogni stazione vengono prelevate dal dizionario "stazioni" e viene creato un overlay della Heatmap per ogni possibile colore di rilevazione (verde, giallo, rosso). Ogni overlay viene riempito con le coordinate delle stazioni che sono risultate rientrare in quel colore specifico. In questo modo quando lanceremo la nostra app avremo una mappa con stazioni di colori diversi,

anche se va specificato, dopo molti test, che la mappa ha quasi sempre dato un esito uniforme su tutto il territorio.

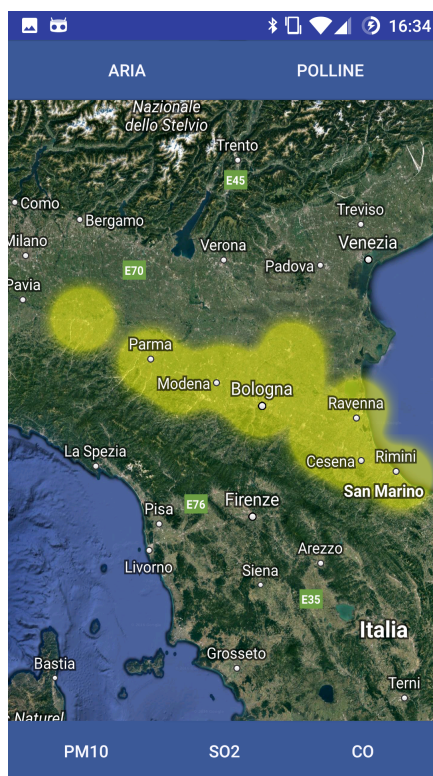


Figura 4.6: ER-POLLEN - status dei pollini al 20/08/2016

Conclusioni

Dall'analisi effettuata sugli Open Data si evince che pur essendo uno strumento ancora poco conosciuto, abbia comunque notevoli potenzialità. I metodi di sfruttamento di questa fonte, come anche il numero di utenti ne fanno uso, crescono ogni giorno.

Il fatto che questi dati siano prodotti liberamente e senza alcuno standard prefissato, rende difficile creare un software che possa sfruttare le diverse fonti di dati contemporaneamente. Le difficoltà maggiori riscontrate, sono state per l'appunto derivanti da questo fattore, poichè, come illustrato in precedenza, ogni dataset è diverso; perfino all'interno dello stesso ente pubblico, i vari tipi di rilevazioni sono stati trattati in modo differente tra loro ed hanno richiesto algoritmi diversi per la loro estrazione ed analisi.

È mia futura intenzione perfezionare questa app aggiungendo l'integrazione con la nota piattaforma di Open Data, ThingSpeak, al fine di rendere ancora più precise, estese e funzionali le rilevazioni dei dati.

Un'altra possibilità vagliabile che ho intenzione di valutare è quella di compiere uno studio sugli effetti dell'aria inquinata, con particolare attenzione alle zone industriali della città e far sì che l'app avvisi l'utente, nel caso in cui egli si trovi in una zona dove lo stato dell'aria non sia sicuro, invitandolo dunque a non permanere per troppo tempo in quell'area. Questa seconda ipotesi includerebbe ovviamente l'implementazione di un sistema di geolocalizzazione, basato su gps e sulle reti a disposizione del soggetto (wi-fi o rete mobile), tecnologia già sfruttata per moltissime altre app.

I motivi principali per cui, al momento questa possibilità è irrealizzabile, dipendono dal fatto che le stazioni di rilevamento sono poche; ne servirebbero molte di più, sparse in tutte le zone della città, il che includerebbe implicitamente moltissimi fondi da investire nel progetto ed altrettanti dati da gestire per l'ente.

Ringraziamenti

Per i ringraziamenti vorrei partire dalle persone che mi sono state più vicine durante la realizzazione di questa tesi: ringrazio moltissimo Federico Montori per i consigli sulla progettazione iniziale del mio lavoro, inoltre, vorrei ringraziare in modo particolare Andrea Betti, Matteo Trentin e Melissa Angelini per i consigli implementativi e gli innumerevoli pomeriggi e serate di delirio.

Ringrazio Chiara Saponaro per la "vicinanza" attraverso tutti i principali canali di comunicazione, la mia crew di Matera: Mariagrazia Giordano, Nicola Tarantino, Licia Rubino e Alessandro Cotrufo per avermi spinto e fatto forza ogni volta che ne ho avuto bisogno.

Un particolare grazie al mio gruppo musicale, per ogni riunione e momento in cui la musica era l'unica distrazione dal codice e per tutte le ore trascorse insieme in questi anni.

Ultima ma non meno importante, ringrazio la mia famiglia per avermi finanziato e sostenuto in tutti questi anni, malgrado le mie tempistiche siano andate ben oltre un normale ciclo di tre anni, grazie a tutti!

Bibliografia

- [1] Open Data: https://it.wikipedia.org/wiki/Dati_aperti
- [2] Iper testo: https://it.wikipedia.org/wiki/Iper_testo
- [3] Ted Nelson: https://it.wikipedia.org/wiki/Ted_Nelson
- [4] Hypercard: <https://it.wikipedia.org/wiki/HyperCard>
- [5] Hypercard: <http://hypercard.org/>
- [6] Tim Berners-Lee: https://it.wikipedia.org/wiki/Tim_Berners-Lee
- [7] Tim Berners-Lee: https://en.wikipedia.org/wiki/Tim_Berners-Lee
- [8] Larry Page: https://it.wikipedia.org/wiki/Larry_Page
- [9] Sergey Brin: https://it.wikipedia.org/wiki/Sergey_Brin
- [10] Google: <https://it.wikipedia.org/wiki/Google>
- [11] Tim Berners-Lee "MLA Berners-Lee, Tim. L'architettura del nuovo Web. Dall'inventore della rete il progetto di una comunicazione democratica, interattiva e intercreativa. Feltrinelli Editore, 2001."
- [12] https://it.wikipedia.org/wiki/Tim_O%27Reilly
- [13] Social Media: https://it.wikipedia.org/wiki/Social_media
- [14] Social Network: https://en.wikipedia.org/wiki/Social_network

-
- [15] Facebook: <https://it.wikipedia.org/wiki/Facebook>
 - [16] Web 2.0: https://it.wikipedia.org/wiki/Web_2.0
 - [17] Web 3.0: https://it.wikipedia.org/wiki/Web_3.0
 - [18] Open Government: https://it.wikipedia.org/wiki/Open_government
 - [19] OKF <https://okfn.org/>
 - [20] W3C <https://www.w3.org/>
 - [21] MongoDB: <https://it.wikipedia.org/wiki/MongoDB>
 - [22] MongoDB: <https://www.mongodb.com/it>
 - [23] CKAN: <https://it.wikipedia.org/wiki/CKAN>
 - [24] CKAN: <http://ckan.arpa.emr.it/organization/arpa-emilia-romagna>
 - [25] Aria: <http://ckan.arpa.emr.it/dataset/dati-pollini-regione>
 - [26] Pollini: <http://ckan.arpa.emr.it/dataset/qualita-dell-aria-rete-di-monitoraggio>
 - [27] Web Service: https://it.wikipedia.org/wiki/Web_service
 - [28] Web Service: https://en.wikipedia.org/wiki/Web_service
 - [29] Dati: <https://it.wikipedia.org/wiki/Dato>
 - [30] Android: <https://it.wikipedia.org/wiki/Android>