

Web-Mobile-GIS e Open Data geografici: due casi di studio pratici

Gianfranco Di Pietro (Geofunction Srls - dipietro@geofunction.it)

Fabio Rinnone (Geofunction Srls - fabiorinnone@geofunction.it)

Introduzione

Definiamo *WebGIS* l'estensione web degli applicativi nati e sviluppati per gestire la cartografia numerica. La principale finalità di un *WebGIS* è la comunicazione e la condivisione con altri utenti di informazioni georiferite. Tuttavia, la recente spinta verso i sistemi *mobile*, dovuta alla crescente necessità di condividere e scambiare le informazioni ovunque ci si trovi - e non necessariamente attraverso una postazione fissa - ha promosso lo sviluppo di applicativi orientati a dispositivi *mobile*, detti appunto *Web-Mobile-GIS*.



Realizzare un Web-Mobile-GIS

Un *Web-Mobile-GIS* è un applicativo software che consente la consultazione di informazioni geografiche *in situ*, su dispositivi mobili di largo consumo dotati di sistemi di posizionamento e connessi ad Internet.

Il processo di realizzazione di un *Web-Mobile-GIS* comporta diverse fasi che possiamo riassumere in tre punti principali:

- pre-elaborazione dei dati;
- creazione del database cartografico;
- implementazione delle applicazioni software.

Pre-elaborazione dei dati

Durante la prima fase, i dati geografici e spaziali, provenienti da diverse fonti e aventi *datum* differenti, devono essere opportunamente trasformati per un utilizzo *GIS-ready*. In questa fase possono essere utilizzati applicativi GIS open source, quali ad esempio Quantum GIS oppure dei software proprietari in grado di gestire e processare i dati che si vogliono utilizzare.

Creazione del database cartografico

I dati geografici in ambito GIS vengono organizzati in due diverse famiglie di formati: dati di tipo RASTER e dati di tipo VETTORIALE. Essi a loro volta si suddividono ciascuno in base a diversi formati informatici, tipologie e caratteristiche di utilizzo.

Per quanto riguarda i dati VETTORIALI, ai fini di un utilizzo ottimale all'interno di un *Web-mobile-GIS*, si suggerisce l'utilizzo di due formati: quelli basati su XML e quelli testuali. Entrambi sono realizzati da strutture file di tipo aggregato: esiste un solo file per l'intero dataset contenente le specifiche (geometrie e tabella attributi).

I principali formati di rappresentazione basati su XML sono:

- KML: *Keyhole Markup Language*;
- GML: *Geography Markup Language*.

KML è un linguaggio basato su XML, sviluppato da Keyhole, Inc. (società acquisita da Google nel 2004), usato per la gestione di dati spaziali su Google Earth e Google Maps. Oggi costituisce uno standard aperto riconosciuto dall'OGC (*Open Geospatial Consortium*). GML è invece una grammatica XML, anch'essa riconosciuta dall'OGC, che consente di esprimere e modellare oggetti geografici, permettendo di descrivere geometrie, *features* e coordinate.

I file di dati in questo formato possiedono estensioni del tipo .kml e .gml.

I formati di rappresentazione testuali risultano facilmente manipolabili e sono adatti all'importazione in DBMS (*Database Managment Systems*). I principali formati utilizzati sono:

- CSV: *Comma-separated Values*;
- SQL: *Structured Query Language*.

CSV è un formato testuale di file di testo, utilizzato tipicamente per l'importazione e l'esportazione di dati tabulari in fogli di calcolo elettronici. In questo formato ogni riga (*record*) è costituita da una linea di testo suddivisa in campi attraverso caratteri separatori, che definiscono la suddivisione in colonne. Il formato si adatta particolarmente all'importazione e all'esportazione in DBMS relazionali. I file di CSV possono avere estensioni .csv o .txt.

Il formato SQL, basato sull'omonimo linguaggio standardizzato per i database relazionali, permette di definire schemi di database e tabelle, di gestirne i dati (inserimento, modifica e cancellazione), oltre che di interrogare i dati stessi. Il formato si adatta particolarmente alla gestione di database cartografici i cui attributi possono anche descrivere geometrie.

I formati RASTER consentono invece la rappresentazione delle informazioni geografiche attraverso una matrice di pixel di forma quadrata o rettangolare: a ciascun pixel è associato un attributo che definisce le caratteristiche dell'elemento rappresentato. L'attributo rappresentato per ogni pixel può essere o singola informazione o un vettore di informazioni.

Tipicamente un file raster è costituito da due blocchi: un'intestazione (*header*, contenente le informazioni su natura, struttura e dimensioni delle celle del file) e un blocco contenente la matrice dati; entrambi i blocchi sono inseriti in un unico file (struttura aggregata).

Sovente le informazioni relative ai dati (metadati) contenute nell'*header* vengono inserite in un file diverso, aggiuntivo. Nel caso in cui vengono memorizzati i parametri di georeferenziazione della matrice dati in un file esterno, essi verranno inseriti in un file denominato *world file*.

Il formato più diffuso per la rappresentazione di dati geografici raster è il *GeoTIFF*, ma altri formati tipicamente utilizzati sono PNG (*Portable Network Graphics*) e GIF (*Graphics Interchange Format*).

I dati cartografici raster, per essere resi fruibili agli utenti di applicativi *Web-Mobile-GIS*, devono essere pubblicati in server cartografici opportunamente configurati per un utilizzo in ambito web con servizi di mappa dedicati (WMS, WCS). Per la configurazione di tali server si può optare per le piattaforme *MapServer* o *GeoServer*.

MapServer è un ambiente di sviluppo che consente il trattamento di informazioni geospaziali ed è utilizzato per la pubblicazione di servizi web conformi alle raccomandazioni dell'OGC.

Geoserver, interamente sviluppato in Java, supporta i più diffusi standard OGC, ma permette anche un'interazione trasparente con client come Google Earth e altri software commerciali.

Implementazione dell'APP

La terza fase di realizzazione di un *Web-Mobile-GIS* consiste, infine, nell'implementazione dell'APP per dispositivi *mobile*.

Le strategie di implementazione, in questo caso, possono essere differenti. Avremo, in particolare, le seguenti tipologie di APP:

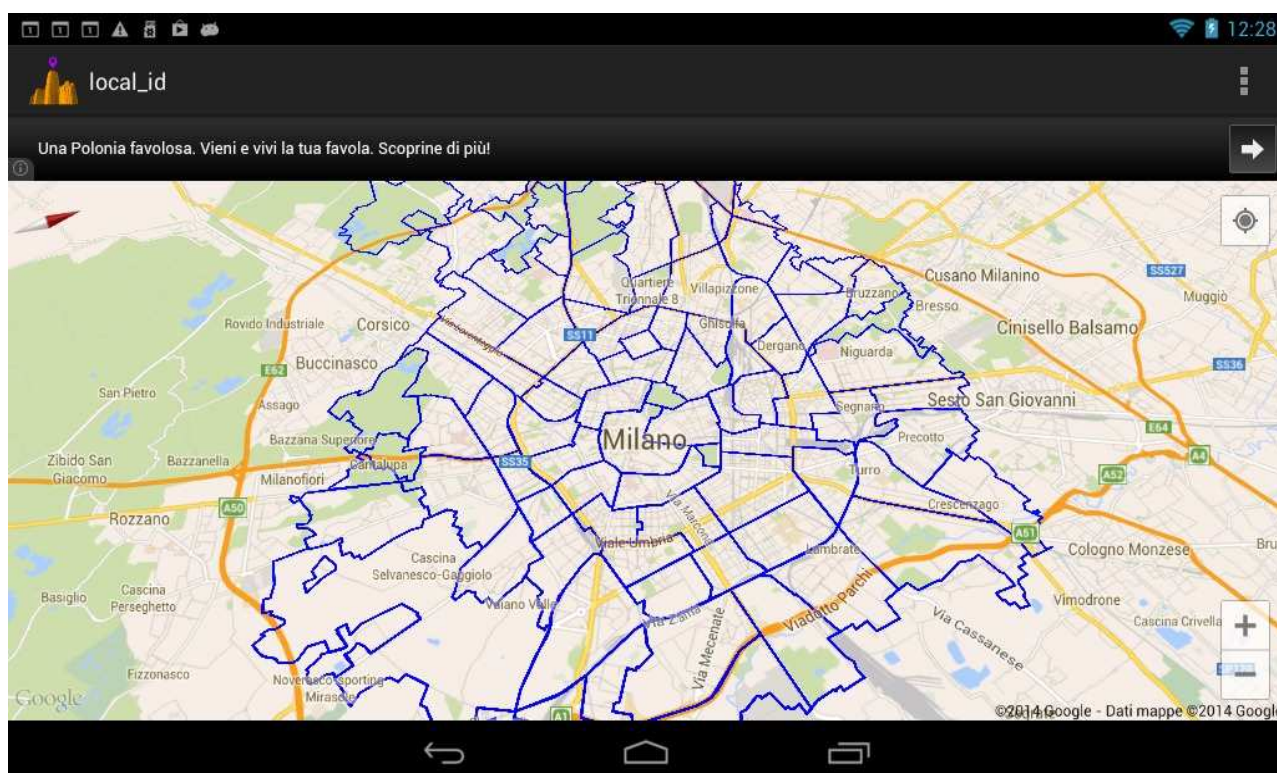
- *web-app*: sviluppata interamente con tecnologie web (tipicamente HTML, Javascript, CSS), orientata all'utilizzo mediante i browser integrati nei dispositivi *mobile*;
- *app nativa*: sviluppata specificatamente per un sistema operativo mobile, utilizzando uno dei linguaggi di programmazione supportati dal particolare sistema operativo (Java, Objective C, C#, etc.);
- *app ibrida*: sviluppata con tecnologie web, installabile ed eseguibile nativamente sui principali *devices* (Android, iOS, Windows Phone, etc.).

Il vantaggio delle *web-app* consiste nella portabilità dell'APP sulla maggior parte dei dispositivi e nei tempi di sviluppo, tipicamente più bassi. Inoltre esse non devono essere sottoposte al processo di valutazione del market ove verranno pubblicate, qualora sia previsto. Il vantaggio delle *app native* consiste invece nella possibilità che esse hanno di interagire con le features del dispositivo mobile e di poter essere pubblicate sui market ufficiali. Quest'ultima possibilità consente alle APP di poter garantire di una maggiore visibilità e diffusione tra gli utenti.

Il vantaggio delle *app ibride* consiste infine nella possibilità di realizzare applicazioni multi-piattaforma in maniera semplice, ma al tempo stesso pubblicabili sui diversi marketplace ufficiali.

Caso di studio 1: *local_id*

local_id è una APP per dispositivi Android che permette di valutare le *performance* urbanistiche e territoriali di aree metropolitane al fine di individuare la zona ideale secondo le aspettative dell'utente. La prima versione ufficiale rilasciata (v0.1) fa uso dei dati aperti del Comune di Milano, estrapolati dal portale dati.comune.milano.it.

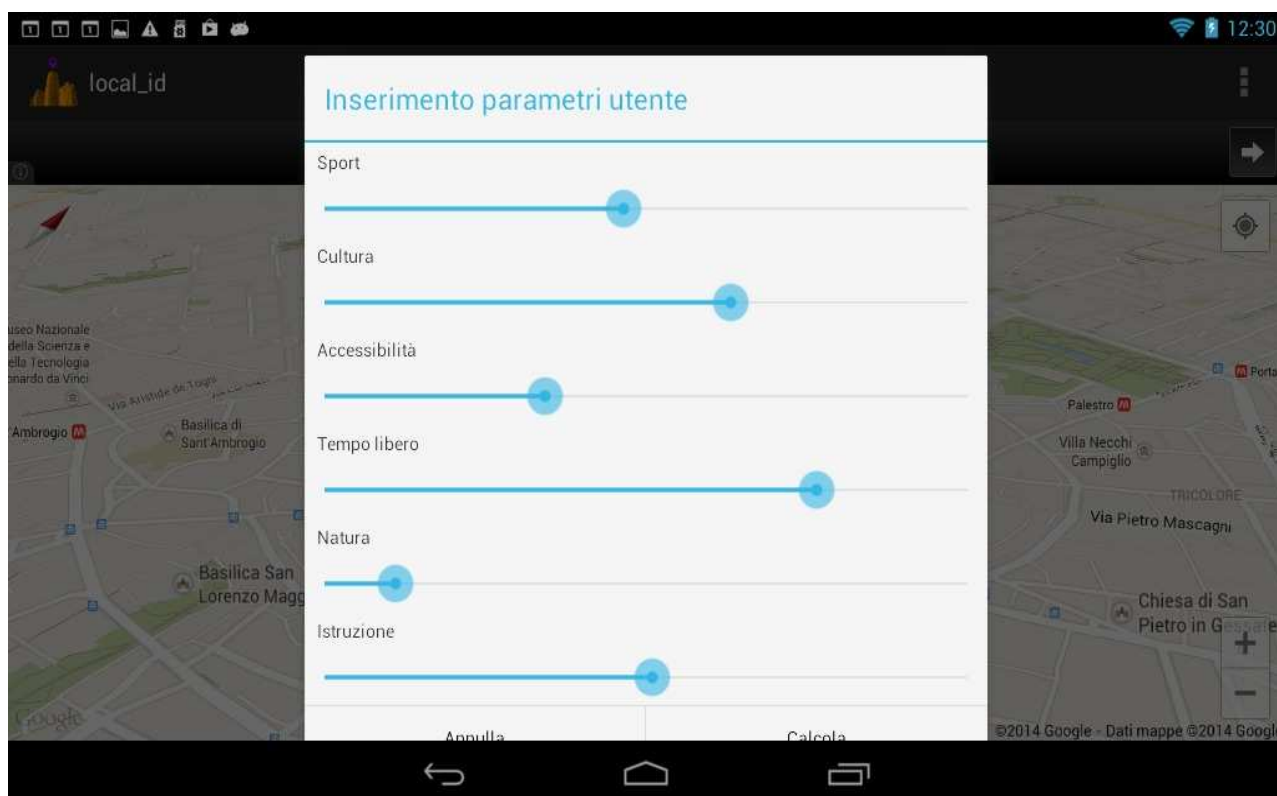


Le analisi territoriali sono state effettuate sfruttando le mappe di OpenStreetMap e gli Open Geo Data del Comune di Milano e della Regione Lombardia.

I dati acquisiti ed elaborati, consistenti in punti di interesse e nel *rating* della zona in base alle proprie aspettative, sono stati poi rappresentati in *mashup* con le mappe di Google e con OpenStreetMap.

Scopo dell'applicazione è valutare la propria *best zone* in base alla personale *local identity*. La APP sviluppata è in grado di effettuare analisi di *benchmarking* in *real-time* tra elementi geografici rappresentanti i poligoni dei Nuclei di Identità Locale (NIL) di Milano. All'utente è lasciata la possibilità di “agire” sulle ponderazioni delle singole performance analizzate, attraverso una metodologia di input assistita. In particolare l'utente fornisce valori di input che rappresentano le proprie aspettative di nucleo di identità locale in termini di:

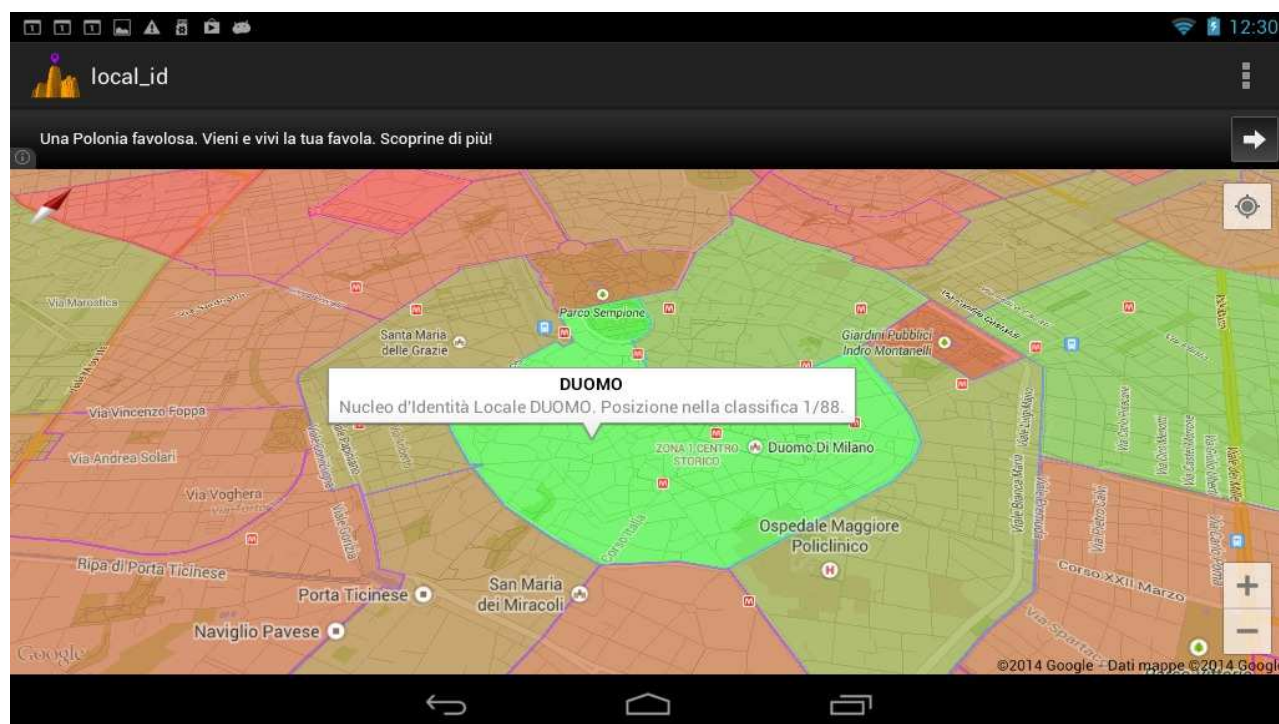
- sport: fare sport in centri sportivi, piscine e parchi pubblici;
- cultura: presenza di cinema, musei, teatri e sedi per le attività culturali;
- accessibilità: possibilità di accesso con i mezzi pubblici, collegamenti e fruibilità;
- tempo libero: presenza di diverse occasioni per il tempo libero;
- natura: un nucleo con tanto verde ed ecosostenibilità;
- istruzione: presenza di scuole ed istruzione;
- vivibilità: nucleo urbanisticamente accogliente.



Nella fase di pre-elaborazione dei dati, per ogni zona - denominata NIL - sono state calcolate attraverso geo-algoritmi del software QGIS le seguenti informazioni:

- presenza e consistenza dei servizi (aggregazione di info puntuali);
- analisi demografica della popolazione, attraverso i dati degli edifici (area, altezza, etc.) per ricavare le densità abitative;
- superfici disponibili per abitante per ogni attributo disponibile (superfici di parchi giochi, zone verdi, etc.).

È stato quindi implementato un algoritmo di calcolo per il *rating* personalizzato del NIL che prevede la predisposizione di apposite matrici per i dati e per i 7 parametri di input (che rappresentano le aspettative dell'utente di NIL ideale). Successivamente vengono analizzate linearmente le performance dei singoli NIL e vengono calcolati punteggi pesati in base alle esigenze di input. Il punteggio finale è un vettore che viene utilizzato per il *rating* comparativo tra le zone. All'utente viene fornita una mappa tematica con delle colorazioni che evidenziano i luoghi e le zone con il miglior punteggio e quelli con il peggior risultato. L'operazione di calcolo e vestizione della mappa è dinamica e *real time*.



Implementazione

local-id è un applicativo sviluppato per dispositivi Android, compatibile con le versioni 3.0 Honeycomb e successive. Per la rappresentazione delle matrici dei dati territoriali, utilizzate nella fase di analisi, si è fatto uso delle librerie JAMA (*Java Matrix Package*). JAMA offre un set di API di algebra lineare e fornisce classi a livello utente per la creazione e manipolazione di numeri reali e di matrici.

Per la scelta del DBMS (*Database Management System*) si è optato per MySQL: proprietà di Oracle e rilasciato sia in licenza GPL che commerciale, supporta, a partire dalla versione 5.0.16, estensioni spaziali che consentono la generazione, la conservazione, l'analisi e l'interrogazione di elementi geometrici e spaziali. In particolare possono essere gestiti punti, linee, poligoni e la loro rappresentazione è in formato WKT (*Well Known Text*). Quest'ultimo è un linguaggio creato appositamente per rappresentare oggetti geometrici semplici (*Geometry, Point, LineString, Polygon, etc.*) e multi-geometrie, ovvero collezioni di geometrie (*MultiPoint, MultiLineString, MultiPolygon, etc.*). I dati immagazzinati nel database sono interrogati mediante appositi script PHP, implementati lato server, che restituiscono in output una loro estrapolazione in formato JSON (*JavaScript Object Notation*), un linguaggio basato su JavaScript, tipicamente usato per lo scambio di dati tra client e server.

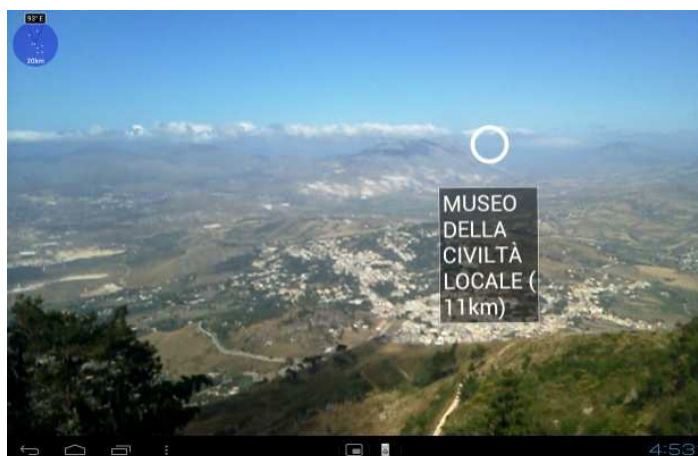
L'applicativo Android, mediante un opportuno *parser* JSON, implementato in linguaggio Java, estrapola le informazioni dal listato JSON affinché queste possano essere gestite da apposite librerie in grado di gestire efficientemente dati spaziali. Nel nostro caso si è fatto uso di JTS (*Java Topology Suite*), un framework open source, sviluppato da Vivid Solutions, Inc., che fornisce un set di API per la gestione di geometrie 2D. Esso mette inoltre a disposizione apposite classi per il *parsing* delle stringhe in formato WKT. Per quanto riguarda infine l'integrazione delle mappe sul dispositivo si è fatto uso delle librerie Google Maps Android API v2. Esse gestiscono autonomamente l'accesso ai server di Google Maps e consentono di rappresentare in una mappa punti, linee e poligoni, oltre che immagini sovrapponibili alla mappa di base.

Caso di studio 2: MobileMap Enna

MobileMap Enna è una APP per dispositivi Android che permette di consultare ed interrogare su *smartphone* e *tablet* le cartografie e i *dataset* istituzionali della Provincia Regionale di Enna, attraverso il collegamento con i server WMS. Interfacendosi con la posizione del dispositivo, offerta dal sensore GPS, permette la consultazione “in situ” delle cartografie, funzionalità molto utile per il supporto ai rilievi (*surveying*) o alle operazioni di controllo del territorio.



La APP consente anche la consultazione di informazioni di carattere turistico e naturalistico. La APP offre funzionalità di *pan & zoom*, *geocoding*, localizzazione, selezione ed interrogazione dei livelli.



Implementazione

Sono state adottate due differenti strategie di implementazione: *web-app* e *app ibrida*, quest'ultima disponibile per dispositivi Android.

La *web-app* è sviluppata interamente con tecnologia web, è accessibile da qualsiasi dispositivo *mobile* ed è compatibile con la maggior parte dei sistemi operativi per *smartphone* e *tablet* dall'indirizzo <http://bit.ly/map-en>.

Per lo sviluppo della *web-app* sono state utilizzate le seguenti tecnologie web:

- OpenLayers;
- jQuery Mobile;

OpenLayers è una libreria implementata in linguaggio Javascript, rilasciata in licenza open *BSD 2-Clause License*, che permette di visualizzare mappe interattive in browser web, consentendo di accedere, attraverso apposite API, a fonti di informazioni geografiche presenti in rete quali:

- WMS: *Web Map Service*;
- WFS: *Web Feature Service*;
- Google Maps;
- Bing Maps;
- OpenStreetMap.

jQuery Mobile è un framework *touch-optimized* implementato in Javascript, derivato da jQuery, ottimizzato per dispositivi *mobile*. Supporta la creazione di temi grafici personalizzati ed è progettato per la costruzione di interfacce user friendly.

La *app ibrida* è sviluppata sia con tecnologie web che nativamente. È implementata in linguaggio Java per dispositivi Android, di cui supporta tutte le versioni a partire dalla Froyo 2.2.2. L'interfaccia web è implementata mediante OpenLayers e jQuery Mobile. Inoltre, al fine di consentire l'interfacciamento tra il codice Javascript dell'interfaccia web con le componenti native del dispositivo, si è fatto uso di Apache Cordova., framework di sviluppo *open source*, distribuito in licenza Apache 2.0.

Sviluppi futuri

MobileMap Enna sarà prossimamente rilasciata in una versione interamente nativa, analogamente a quanto già realizzato con *app* simili (ad es. MobileMap Agrigento), sarà dotata di funzionalità di realtà aumentata, facendo uso, in questo caso, della piattaforma *open source Mixare*.

Riferimenti

- Di Pietro G., Rinnone F. (2013). *Web-mobile-GIS, la nuova frontiera di pubblicazione dei dati cartografici*. In: Biallo G. (eds.) *Dati Geografici Aperti – Istruzioni per l'uso*.
- QGIS Development Team (2013). *Quantum GIS geographic information system*. Open Source Geospatial Foundation Project.
- Google, Inc. (2010). *KML Reference*. <https://developers.google.com/kml/documentation/kmlreference>.
- Open Geospatial Consortium, Inc. (2008). *KML*. <http://www.opengeospatial.org/standards/kml>.
- Open Geospatial Consortium, Inc. (2012). *Geography Markup Language*. <http://www.open-geospatial.org/standards/gml>
- Shafranovich Y. (2005). *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180, October 2005.
- Atzeni P., Ceri S., Paraboschi S., Torlone R. (2009). *Basi di dati. Modelli e linguaggi di interrogazione*. Terza edizione. McGraw-Hill.
- Ritter N. (2000). *GeoTIFF Format Specification. GeoTIFF Revision 1.0*. <http://www.remotesensing.org/geotiff/spec/geotiffhome.html>.
- Kropla, B. (2005). *Beginning MapServer: open source GIS development*. Apress.

- Giannecchini S., Aime A. (2011). *Geoserver, il server open source per la gestione interoperabile dei dati geospaziali*. Atti della XV Conferenza Nazionale ASITA, 15-18 novembre 2011, Reggio di Colorno.
- Hicklin, J., Moler, C., Webb, P., Boisvert, R. F., Miller, B., Pozo, R., Remington, K. (2012). *Jama: A Java matrix package*. <http://math.nist.gov/javanumerics/jama>.
- Du Bois, P. (2004). *MySQL*. Pearson Italia Spa.
- Open Geospatial Consortium, Inc. (n.d.). *Well Known Text format*. <http://www.geo-api.org/2.0/javadoc/org/opengis/referencing/doc-files/WKT.html>.
- Bray, T. (2013). *The JSON Data Interchange Format*. Standard ECMA-404.
- Vivid Solutions, Inc. (2006). *Java Topology Suite*. <http://www.vivid-solutions.com/jts/JTSHome.htm>.
- Geofunction Srls (2013). *local_id: where is your local identity?* <http://www.slide-share.net/geofunction/localid>.
- Provincia Regionale di Enna (2013). *Open Data Territorio Enna*. <http://www.open-dataterriorioenna.it/>
- OpenLayers Contributors (2013). *OpenLayers*. <http://www.openlayers.org/>.
- The BSD 2-Clause License. Open Source Initiative. <http://opensource.org/licenses/BSD-2-Clause>.
- Open Geospatial Consortium, Inc. (2006). *Web Map Server*. <http://www.opengeospatial.org/standards/wms>.
- Open Geospatial Consortium, Inc. (2010). *Web Feature Service*. <http://www.opengeospatial.org/standards/wfs>.
- Reid, J. D. (2012). *jQuery Mobile*. O'Reilly Germany.
- Open Source Initiative. (2006). *The MIT License*. <http://opensource.org/licenses/mit-license.php>.
- Apache Cordova Foundation (2012). *Apache Cordova*. <https://cordova.apache.org/>.
- Di Pietro G., Rinnone F., Casano G., Sferlazza E. (2013). *MobileMap-Agrigento: gli OpenGeo-Data della Pubblica Amministrazione su smartphone*. Atti della XVII Conferenza Nazionale ASITA, 5-7 novembre 2013, Riva del Garda.
- Gobbetti, D. (2011). *mixare-Augmented Reality Engine*. <http://www.mixare.org>.