

# OpenFarmacie@Ravenna

Il presente documento illustrerà la progettazione e l'implementazione di un sistema backend, nonché di una controparte client web, per la gestione, manipolazione e utilizzo dei dati relativi alle farmacie della provincia di Ravenna.

Il backend, com'è ovvio, si occupa di gestire i dati tramite un qualche sistema database-oriented. Dal lato client si ha invece un'interfaccia web che fornisce mappe interattive, consultabili da un browser desktop così come da un browser mobile. Inoltre, è possibile visualizzare i dati tramite grafici e mappe di densità, utili per analisi e studi statistici.

## Parte 1: raccolta dei dati

A partire dal file kml disponibile al link

<http://opendata.comune.ravenna.it/dataset/farmacieravenna/resource/9f775726-9eee-479a-b7df-ba1fd44fe0ce>

si è proceduto alla normalizzazione e strutturazione dei dati in esso contenuti, arricchendoli con altri disponibili presso

<http://www.farmacieravenna.com/indiFarmacie.asp>,

ed eventualmente correggendo il tutto nel caso di imprecisioni (molto frequenti nel caso delle coordinate geospaziali e/o indirizzi, non sempre funzionanti con i motori di geocoding disponibili online, in primis quello di GoogleMaps).

I dati sono stati inseriti in un foglio di calcolo online, ovvero Google Spreadsheet, il cui url è:

[https://docs.google.com/spreadsheets/d/1uYVT60vBR2gdR8zVisQHOq6yPOC9o8qTwgFw\\_hJuVCo/edit#gid=1588294032](https://docs.google.com/spreadsheets/d/1uYVT60vBR2gdR8zVisQHOq6yPOC9o8qTwgFw_hJuVCo/edit#gid=1588294032).

Il foglio di calcolo permette una facile manipolazione (aggiunta/correzione) di nuovi dati, senza dover far ricorso a un database sql e a qualche modalità di inserimento nel suddetto; inoltre, permette una facile aggregazione di alcuni dati, utili successivamente per analisi tramite grafici o mappe (vedasi in seguito).

## Parte 2: Backend

La scelta di uno spreadsheet nel cloud, e in particolare quello di Google, non è casuale, poichè, tramite la tecnologia Apps Script, si possono manipolare i dati dello spreadsheet tramite un linguaggio lato server compatibile con ECMA5/Javascript.

In questo specifico caso, AppsScript permette di sincronizzare l'intero contenuto dello sheet in database NoSQL, sempre fornito dal cloud di Google, chiamato ScriptDB, il quale può essere interrogato/manipolato con una sintassi SQL-like, nonostante i dati siano memorizzati sottoforma di array javascript: quest'ultima proprietà è assai interessante, perchè, con alcuno sforzo, si ottiene l'intero database (ovvero l'intero contenuto dello sheet) sottoforma di oggetto JSON: richiesta soddisfatta da un webservice incluso nel motore di AppsScript.

Sempre tramite AppsScript, si può popolare una FusionTable (un altro content editor fornito dal cloud di Google), che a sua volta fornisce mappe (GoogleMaps), i cui markers (e relativi infowindows) sono automaticamente creati a partire dai dati contenuti nella table.

### HowTo backoffice-oriented:

Lo sheet offre un menu aggiuntivo "Export" che permette di:

1. aggiornare la fusiontable e lo ScriptDB con i dati dello sheet medesimo
2. ottenere l'url del webservice che fornisce il JSON

L'aggiornamento della fusiontable è obbligatorio ogni qualvolta si modifichi qualcosa nello sheet, affinché le tabelle mostrate nell'interfaccia web mostrino i dati coerentemente; l'aggiornamento dello scriptdb è invece necessario affinché l'oggetto JSON fornisca dati sempre coerenti con lo sheet.

Il webservice che espone JSON, già incluso nel menu "dettagli tecnici" nell'interfaccia web, è anche disponibile all'indirizzo:

[https://script.google.com/macros/s/AKfycbzTwsoaDo3iWolKhSwJRR\\_QkTHVM6HvXXpmU-Kx0EO4oKQJZU/exec](https://script.google.com/macros/s/AKfycbzTwsoaDo3iWolKhSwJRR_QkTHVM6HvXXpmU-Kx0EO4oKQJZU/exec)

### HowTo developer-oriented:

a partire dall'url dello sheet (già precedentemente citato)

[https://docs.google.com/spreadsheets/d/1uYVT60vBR2gdR8zVisQHOq6yPOC9o8qTwgFw\\_hJuVCo/edit#gid=1588294032](https://docs.google.com/spreadsheets/d/1uYVT60vBR2gdR8zVisQHOq6yPOC9o8qTwgFw_hJuVCo/edit#gid=1588294032), e clickando la voce di menu "Tools", e successivamente "Script Editor...", si accede all'editor di AppsScript, disponibile anche direttamente all'url:

[https://script.google.com/macros/d/MTIkumjjzcs6cX1OZFwSP7Bmb0qzzQGAX/edit?uiv=2&mid=ACjPJvH0QgQs\\_yk7y3Jk0RuH5vNkfXyGcfhnwi6XJEJ6348zMInG3v\\_BImDL9k1E4tRX9kpRbInlBnaJu9PvzJzKN4yKjOdEOfb8QV-aZUAxO582LRWLKpM1RUQmvKvyF\\_3cqQ66WMhW5A](https://script.google.com/macros/d/MTIkumjjzcs6cX1OZFwSP7Bmb0qzzQGAX/edit?uiv=2&mid=ACjPJvH0QgQs_yk7y3Jk0RuH5vNkfXyGcfhnwi6XJEJ6348zMInG3v_BImDL9k1E4tRX9kpRbInlBnaJu9PvzJzKN4yKjOdEOfb8QV-aZUAxO582LRWLKpM1RUQmvKvyF_3cqQ66WMhW5A)

Tramite l'ide si può modificare il codice apps-script che manipola i dati dello sheet, e quindi la modalità di esportazione verso il database NoSQL, verso JSON, nonché verso la FusionTables che genera le mappe disponibili nel url di cui all'introduzione.

A meno di bug non rilevati, le funzioni già disponibili sono sufficientemente complete e corrette. La scelta di un backend nel cloud fornisce evidenti vantaggi, in primis svincolando dai vari problemi di deploy e code hosting: in più, il cloud di Google permette di manipolare i dati in differenti e non banali modalità.

Per motivazioni legate all'identità digitale, lo spreadsheet, il progetto apps-script e la fusiontable sono di proprietà di un determinato account google (in questo caso, "Massimiliano Leone"), ma si può facilmente cedere la "proprietà" del tutto ad all'account google della persona responsabile del sistema; inoltre, ai fini del concorso, i tre "oggetti" sono pubblicamente accessibili in lettura, ma, in produzione, per coerenza con l'etica "opendata", è ragionevole voler esporre lo sheet, l'url che genera il JSON, la tabella della FusionTable e i vari grafici e mappe inclusi nello sheet nonché nella table, così come è ragionevole configurare un accesso ristretto per il progetto apps-script che contiene il codice server.

Infine, laddove Google AppsScript non riesce a soddisfare tutte le funzionalità richieste, interviene uno script php: in particolare, è utile per ottenere un formato strutturato e coerente dell'elenco delle farmacie di turno, già disponibile presso

<http://www.farmacieravenna.com/turni.asp> (mese corrente)

<http://www.farmacieravenna.com/turni2.asp> (mese successivo)

<http://www.farmacieravenna.com/turni3.asp> (2 mesi successivi)

Tuttavia, l'elenco è fornito tramite una pagina dinamica (asp, come si evince dall'url), e AppsScript non ha un metodo capace di invocare una richiesta HTTP GET e ottenere una RESPONSE presso una pagina dinamica. Lo script PHP fa esattamente questo, fornendo anche una modalità di query che permette di ottenere i nomi delle farmacie relative ad un giorno e/o mese arbitrario (sempre all'interno dell'intervallo trimestrale fornito dalle pagine asp).

### Parte 3: l'interfaccia client

Come già dall'introduzione, all'url <http://www.openfarmacieravenna.url.ph> è disponibile una mappa GoogleMaps interattiva, che fornisce una visione cartografica delle farmacie della provincia di Ravenna: l'interattività è fornita da una serie di checkbox che possono filtrare i risultati di ricerca.

L'engine è pure javascript/css/html5, escludendo, ovviamente, la parte di layer delle mappe fornite già da Google tramite l'esportazione delle FusionTable.

Infine, clickando il link "dettagli tecnici" è possibile visionare una serie di grafici ottenuti con GoogleChart, e generati grazie all'aggregazione di alcuni dati già presenti nello spreadsheet: i nomi dei grafici sono significativi.

I grafici possono essere utili per alcune analisi, ed eventualmente per migliorare i servizi già esistenti.

Bologna, 20-03-2014

Massimiliano Leone  
Antonio Notarangelo