DOI: 10.26314/GARR-Conf17-proceedings-26

WeatherLink una piattaforma per l'integrazione e la visualizzazione dei dati meteo

Riccardo La Grassa[1], Marco Alfano[1,2], Biagio Lenzitti[1], Davide Taibi[3]

[1]Dipartimento di Matematica e Informatica, Università degli Studi di Palermo, [2]Anghelos Centro Studi Sulla Comunicazione, [3]Consiglio Nazionale delle Ricerche, Istituto per le Tecnologie Didattiche

Abstract. WeatherLink è una piattaforma per la raccolta, l'elaborazione e la pubblicazione di dati relativi a misure atmosferiche. WeatherLink si compone di una parte server che memorizza l'enorme quantità di dati provenienti dalle diverse board dislocate nel territorio che fungono da client. Ogni board tramite i suoi sensori raccoglie dati su temperatura, pressione atmosferica, qualità dell'aria e li invia al server. Il server si occupa della memorizzazione dei dati, e offre anche delle API che possono essere interrogate per estrarre i dati ed effettuare elaborazioni successive. I dati vengono esposti come dati aperti nel formato JSON e JSON-LD. Le API sono state utilizzate per la realizzazione di una Web Application che consente il monitoraggio delle board e la visualizzazione dei dati aggregati o per singola board attraverso grafici relativi ai diversi parametri metereologici catturati dai sensori.

Keywords. Open Data, IoT, Data integration and visualization, monitoraggio ambientale.

Introduzione

WeatherLink è una piattaforma per la raccolta, l'elaborazione e la pubblicazione di dati di natura metereologica. L'idea di WeatherLink è quella di creare una fitta rete di stazioni meteo amatoriali e non, al fine di integrare e rendere disponibili per successive elaborazioni i dati, in formato aperto, provenienti da diverse sorgenti. Come mostrato in figura 1, la piattaforma si compone di due elementi principali: i) il server che riceve dalle stazioni i dati me-

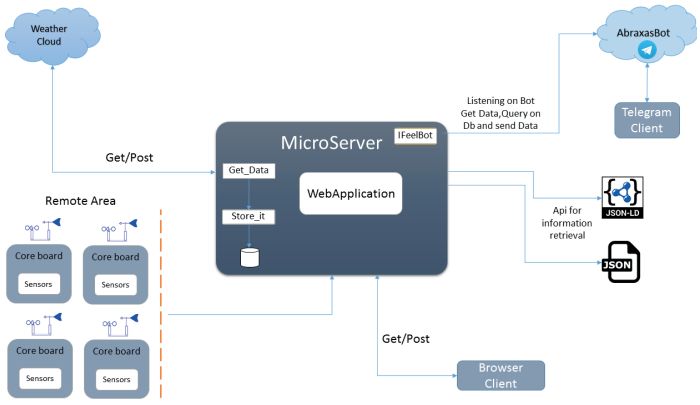


Fig. 1
Architettura di WeatherLink

Venezia, 15-17 novembre 2017 - ISBN: 978-88-905077-7-9

teo acquisiti dai sensori ed espone, tramite servizi implementati con metodi REST, i dati in formato aperto; ii) le stazioni meteo, realizzate tramite Board specifiche, sono dislocate nel territorio e raccolgono i dati mediante sensori di temperatura, umidità, pressione.

Il server è stato implementato attraverso diversi moduli Python per l'acquisizione dei dati e la loro memorizzazione. Per quanto riguarda le board, il codice è stato scritto interamente in C. Il codice è stato reso pubblico in modo da consentirne il riuso . Gli utenti che vogliono contribuire alla rete di sensori WeatherLink possono registrare sul server la propria stazione meteo e installare sulle board il codice necessario. L'utente, inoltre, può accedere sia ai dati della propria board in locale, per via dell'aggiuntivo sviluppo del frontend lato board, che ai dati storicizzati sul server. Uno dei punti di forza di WeatherLink è offrire un pieno servizio a chiunque voglia utilizzare le API, esponendo dati in formato aperto (JSON e JSON-LD) e senza restrizioni temporali. Questo approccio rende WeatherLink differente da altre piattaforme disponibili in rete che limitano il riuso dei dati o ne restringono l'accesso solo a un ristretto arco temporale. Sulla base dei dati resi disponibili dal server WeatherLink è stata costruita una Web Application appositamente progettata per la visualizzazione dei dati meteo, descritta in dettaglio nella seguente sezione.

1. Data Visualization

La Web Application è stata realizzata con l'utilizzo del framework Django, ed è basata sul paradigma MVC (Model View Control) che ha permesso la creazione di modelli di dati efficienti utilizzati per lo storage su database. Una volta creati i modelli dei dati, specifici comandi estrapolano il codice SQL necessario per eseguire le query su un database di tipo SQLite. Inoltre si è preferito avere due database indipendenti, uno gestito da Django per la gestione delle autenticazioni degli utenti, ed uno per la memorizzazione dei dati provenienti dalle stazioni meteo alimentato anche da moduli python in esecuzione periodica per il recupero dei dati da fonti di dati esterni come OpenWeatherMap (OWM) . Nello specifico i moduli scritti in python, Get_data e Store_it, hanno il compito di recuperare i dati da OWM e di memorizzarli in db_weather.sqlite. Una volta registrati su OpenWeatherMap e ottenuto un token, si sono utilizzate le API messe a disposizione della piattaforma OWM (con licenza CC-BY-SA), al fine di creare un proprio storico da utilizzare per successive elaborazioni e da integrare con i dati raccolti dalle stazioni meteo. La finalità di WeatherLink, è quella, di creare un proprio network di weather station, integrare i dati con altre fonti di dati aperti relativi alle misurazioni meteo e fornire, dopo successiva aggregazione ed elaborazione, nuove visualizzazione dei dati.

In fase di progettazione del database, si sono considerati diversi aspetti legati alla pertinenza dei dati nel contesto meteorologico, come la gestione dei diversi tipi di dati, o le problematiche su possibili errori dovuti all'invio di dati parziali. Un utente ha la facoltà di creare più stazioni meteo legate al proprio account, registrando ogni stazione meteo sul server WeatherLink e ottenendo uno specifico token. Uno dei principali servizi offerti da WeatherLink è la possibilità di visualizzare in tempo reale i dati raccolti dalla propria stazione meteo direttamente plottata su OpenStreetMap (OSM).
La piattaforma presenta due tipologie di plotting: Mappa e Grafici. Per la visualizzazione

Mappa è stata utilizzata la libreria folium (leaflet derivata per python) per plottare degli oggetti di tipo circle direttamente su mappa OpenStreetMap (OSM). Inizialmente, si effettua un recupero tramite query, nel database db_weather.sqlite dei dati più recenti, ma l'utente può estendere l'arco temporale da visualizzare. La figura 2 mostra un esempio dei grafici che possono essere visualizzati dalla Web Application di WeatherLink.

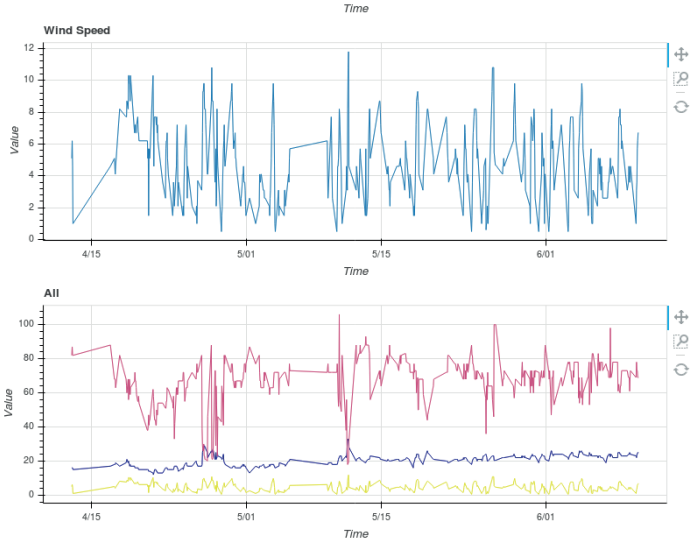
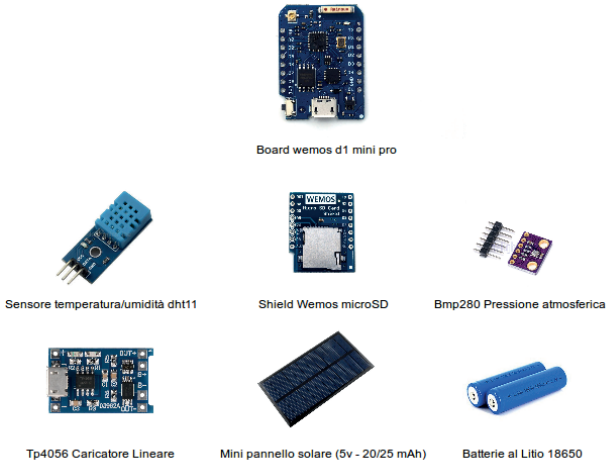


Fig. 2
Esempio di visualizzazione dei dati

2. Realizzazione e dati tecnici

L'architettura di WeatherLink è composta da un server centrale e numerose boards che compongono la rete di stazioni meteo. Come mostrato in Figura 3, al momento la rete è composta da board Wemos d1 mini pro con Esp8266 (Datasheet per ESP8266), sensore DHT11 (Temperatura/Umidità) e Bmp 280 (Pressione). Inoltre le board sono dotate di Caricatore Lineare Tp4056, Mini pannello Solare (5v - 20/25 mAh) e Batteria al litio. I2C/

Fig. 3
Stazione meteo e sensori utilizzati



OneWire/SPi sono i tre possibili protocolli per la comunicazione con sensori esterni. La board ha solo un canale analogico e 11 digitali, Flash 16 MB, 80KB di ram e 80/160 MHz di clocks.
Il flusso di esecuzione implementato su ciascuna board è il seguente:

- Setup del web Server e connessione alla rete.
- Inizializzazione dei vari sensori e della porta d'ascolto
- Acquisizione dati ogni 5 secondi e invio al server. (Funzionalità one shoot)

Inoltre in locale gli utenti possono accedere ai dati raccolti tramite normale protocollo HTTP (attraverso la funzionalità di web server locale). In base alla richiesta fatta dagli utenti (tramite HTTP GET), la board risponde con l'informazione desiderata per ciascun sensore.

Recentemente, una nuova versione è in fase di testing per ridurre drasticamente i consumi energetici da parte della board, introducendo un pannello fotovoltaico, un caricatore lineare con modalità trickle, ed una batteria al litio. Il software è stato modificato, eliminando la funzionalità di web server locale, e decrementando il consumo di corrente ponendo la board in modalità deep-sleep ogni qualvolta si effettui un invio dei dati (one shoot). Il risveglio dalla modalità, viene effettuato per mezzo di un interrupt causato da un timer rtc interno. La board è stata pensata per operare sia in rete che in zone sprovviste di copertura wireless. In quest'ultimo caso la board implementa un file system e conserva i dati dei sensori in una microSD.

3. Conclusioni e sviluppi futuri

La piattaforma WeatherLink è stata progettata con l'obiettivo di integrare dati provenienti da stazioni meteo realizzate attraverso apparecchiature IoT, con altre sorgenti di dati pubblicati in formato aperto, in modo da creare una fonte completa di dati meteo su cui effettuare nuove elaborazioni e visualizzazioni.

Al momento si è in procinto di costruire una rete di stazioni di rilevamento sparse per la città di Palermo che conterranno, oltre al sensore per il rilevamento della temperatura e umidità, un rilevatore del livello di qualità dell'aria in modo che si riesca a fornire una sorta di indice di "qualità ambientale" delle varie zone della città. Inoltre sono in via sviluppo dei connettori per i dati, disponibili in formato aperto, messi a disposizione da ARPA Sicilia . In questo modo i dati raccolti dalle stazioni meteo verranno integrati non solo con i dati di OWM ma anche altre fonti di dati aperti spianando la strada ad ulteriori elaborazioni (Bannayan 2008, Holmstrom 2016).

Riferimenti bibliografici

Bannayan M., Hoogenboom G. (2008). Weather analogue: A tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach, In Environmental Modelling & Software, 23, (6), 703-713.

Datasheet per ESP8266, disponibile in rete: http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

Holmstrom H., Liu D., and Vo C. (2016). Machine Learning Applied to Weather Foreca-

sting. Stanford University.

http://www.arpa.sicilia.it/temi-ambientali/monitoraggi-ambientali

https://github.com/pulsar2468/IoT-project

https://www.djangoproject.com

https://openweathermap.org

https://www.youtube.com/watch?v=pKFKAl9XQC4

Autori

Riccardo La Grassa riccardo.lagrassa@community.unipa.it

Laureato in scienze informatiche nel 2015, da alcuni anni si occupa di progetti in cui vengono applicate le tecnologie IoT, embedded programming e machine learning in diversi settori. Recentemente ha collaborato con esperti del CNR-ISSIA sull'interfacciamento hardware delle board programmabili. Attualmente è laureando magistrale in informatica, la tesi di laurea riguarda la realizzazione di un sistema intelligente per la distribuzione di energia elettrica mediante tecniche di machine learning.

Marco Alfano marco.alfano@anghelos.org

Ingegnere Elettronico e Dottore di Ricerca in Ingegneria Elettronica, Informatica e delle Telecomunicazioni con un'esperienza professionale trentennale nel campo dell'ICT maturata (sia in Italia che all'estero) in diversi ambiti lavorativi quali quello della pubblica amministrazione, aziendale, universitario e di ricerca e con diverse mansioni quali manager di struttura, coordinatore di progetto, consulente, sistemista, progettista, programmatore, docente e tutor. Svolge ricerca nel campo Open Data e servizi socio-sanitari.

Biagio Lenzitti biagio.lenzitti@unipa.it

Ricercatore presso il Dipartimento di Matematica e Informatica dell'Università degli studi di Palermo. Attualmente docente del corso di programmazione per il corso di laurea in Informatica e già docente dei corsi di Reti di calcolatori e Sistemi Operativi. Responsabile locale di vari progetti Europei (ReBus, Fetch, Trice) le sue attività di ricerca sono andate dal Calcolo parallelo ai linguaggi Pittorici fino al Pattern Recognition. Negli ultimi anni si è interessato di E-Learning e di Smart Health.

Davide Taibi davide.taibi@itd.cnr.it

Ricercatore a tempo indeterminato dell'Istituto per le Tecnologie Didattiche del Consiglio Nazionale delle Ricerche. Le sue attività di ricerca sono focalizzate principalmente sull'applicazione delle tecnologie innovative nel campo dell'e-learning, con particolare riguardo all'apprendimento con dispositivi mobili, il Web semantico e i Linked Data, gli standard per la progettazione dei processi educativi. Professore a contratto presso l'Università degli Studi di Palermo per il corso di Tecniche per la gestione degli Open Data.