

Prova di Laboratorio - Sistemi e Applicazioni di Rete

Appello 23 luglio 2018 - Modalità di esame 2017/2018

Si realizzi un'applicazione Web su Google App Engine per la gestione di un sistema di domotica per il controllo centralizzato dell'irrigazione in ambito smart city, dove l'applicazione Web mantiene delle regole di irrigazione e le mette a disposizione degli utenti e ai dispositivi IoT tramite opportune interfacce Web ed API.

L'applicazione gestisce l'irrigazione di diverse **città**, ognuna delle quali include diverse **zone** configurate con molteplici **regole di irrigazione**. L'applicazione deve mantenere lo stato delle regole definite e deve implementare le interfacce necessarie per l'accesso alle funzionalità da parte di utenti e applicazioni esterne.

Ogni città è identificata da un nome univoco, mentre ogni zona è identificata da un nome univoco all'interno della stessa città. Per ogni zona possono esistere più regole di irrigazione, anch'esse definite da nomi univoci all'interno della stessa zona. Ogni regola include le seguenti informazioni:

- orario di attivazione della regola:
 - l'orario di attivazione di una regola può assumere un valore multiplo di 3, (e.g., 00:00, 03:00, 06:00) e avrà durata fissa di 3 ore (e.g., la regola che si attiva alle 03:00 dura dalle 03:00 alle 05:59);
 - può esistere solo una regola per la stessa zona e lo stesso orario di attivazione;
- intensità di irrigazione:
 - l'intensità di irrigazione è un valore percentuale da 0 a 100, che corrispondono rispettivamente ad assenza di irrigazione e flusso di acqua massimo per il valore.

Ad esempio: una zona può avere nome "giardino", essere associata alla città "Mantova" ed essere associata alle seguenti regole:

- nome: "notte", orario: 3, intensità: 80
- nome: "alba", orario: 6, intensità: 40
- nome: "sera", orario: 18, intensità: 20

Si implementino le seguenti funzionalità:

1. interfaccia di visualizzazione e manipolazione tramite API REST;
2. client per la gestione di malfunzionamenti degli irrigatori realizzato tramite protocollo MQTT;
3. interfaccia di visualizzazione delle zone esistenti e delle relative regole impostate tramite pagine HTML.

API REST

L'applicazione deve esporre le seguenti funzionalità tramite opportune API:

- allo URI `/api/v0.1/city/<city>` è possibile registrare una nuova città o ottenere la lista delle zone associate ad una città esistente;
- allo URI `/api/v0.1/city/<city>/zone/<zone>` è possibile registrare una nuova zona o ottenere la lista delle zone associate ad una città;
- allo URI `/api/v0.1/city/<city>/zone/<zone>/<rule>` è possibile ottenere, creare o modificare le regole associate ad una zona.

L'interfaccia di utilizzo delle API deve soddisfare **rigorosamente** il file di specifica *OpenAPI* disponibile presso il sito del corso¹.

¹<https://weblab.ing.unimore.it/sar/1718/lab/openapi-irrigation.yaml>

Interfaccia MQTT

Si richiede di implementare un sistema Pub/Sub basato su MQTT per il controllo diretto dei dispositivi di irrigazione. Nell'ambito dell'esame il sistema deve essere emulato localmente. Si richiede di modellare degli opportuni topic per rappresentare al meglio la realtà in esame.

Il sistema implementato è composto da due moduli python:

- uno script eseguibile `irrigate_pub.py` con interfaccia da linea di comando che permette di attivare e disattivare tutti gli irrigatori di una zona o un singolo irrigatore in base ai parametri passati, tramite pubblicazione di questa informazione sulla rete MQTT. In dettaglio:
 - l'interfaccia dello script deve essere la seguente:

```
irrigate_pub.py {<irrigator>,<zone>} <action={START,STOP}>
```

Ad esempio:

```
irrigate_pub.py <irrigator> START
irrigate_pub.py <zone> STOP
```

- uno script eseguibile `irrigate_sub.py` che avvia un servizio di monitoraggio sulla zona o sull'irrigatore specificato. Il servizio è implementato tramite un subscriber MQTT che mostra a video i comandi ricevuti. L'interfaccia dello script deve essere la seguente:

```
irrigate_sub.py {<irrigator>,<zone>}
```

Interfaccia HTML

L'interfaccia HTML deve prevedere due pagine:

1. visualizzazione di tutte le zone presenti nell'applicazione;
2. visualizzazione dei dettagli delle regole per una città.

La prima pagina deve mostrare agli utenti la lista delle zone raggruppandole per le relative città. Per ogni città deve essere presente un link per accedere alla seconda pagina.

La seconda pagina deve mostrare le previsioni meteo della città richiesta per i cinque giorni successivi e le regole della città aggregate per orario. In dettaglio, la pagina deve mostrare:

- nome della città;
- lista di primo livello per il giorno descritto, indicando la temperatura massima e la temperatura minima;
- lista di secondo livello degli intervalli di tempi per cui esistono delle regole. Colorare di blu l'intervallo di tempo se è prevista pioggia (ovvero se in questo intervallo di tempo si ottiene un valore `precip > 0` dalle API, come descritto sotto);
- lista di terzo livello che mostra il nome della regola, il nome della zona e l'intensità di irrigazione indicata nella regola.

Per ottenere le informazioni delle previsioni si utilizzino i servizi di terze parti City Geo-Location Lookup² e Weather³. In particolare, il primo servizio permette di ottenere le coordinate della città e il secondo di ottenere le previsioni meteo dei 5 giorni seguenti con un dettaglio di 3 ore.

Note:

- è necessario disporre di un account Mashape valido per l'accesso ai servizi di terze parti;
- si supponga come valido il primo elemento ritornato dalle API esportate dal servizio City Geo-Location Lookup.

²<https://market.mashape.com/dev132/city-geo-location-lookup>

³<https://market.mashape.com/weatherbit/weather>