

Internet of Things, a.a. 2019-2020

Proposte di Progetti

Docenti: Prof. Luciano Bononi, Prof. Marco Di Felice
{luciano.bononi, marco.difelice3}@unibo.it

May 19, 2020

Regole generali

- **COME** Il progetto può essere svolto singolarmente o in gruppi di massimo 3 unità.
- **COME** Il progetto può essere svolto su traccia proposte dallo studente (previa approvazione dai docenti) oppure sviluppando le tracce descritte qui sotto. Quest'ultime devono essere considerate delle specifiche di massima; è possibile estenderle/modificarle/customizzarle a piacimento, a patto di non ridurre la complessità.
- **COME** La consegna avviene attraverso la piattaforma Insegnamenti On Line (IOL) di UNIBO, nella pagina del corso.
- **QUANDO** Non sono previste deadline di consegna.
- **COSA** Occorre consegnare i **sorgenti** ed una **relazione** (strutturata nelle seguenti sezioni: Introduzione, Progettazione, Implementazione, Risultati), lingua a scelta (inglese preferibile).
- **COSA** A valle della consegna, occorre preparare una **presentazione con slides** per la discussione del progetto: durante la presentazione, verrà chiesto di effettuare una demo del progetto.
- **NOTA** Tutti i membri del gruppo devono essere presenti durante la discussione, e devono conoscere il 100% del progetto svolto.

1 Proposta 1 (Safe Lab)

Si vuole realizzare un sistema di **ambient intelligence** basato su dispositivi e tecnologie IoT per il monitoraggio di parametri ambientali e di persone all'interno di una stanza (es. laboratorio di Via Ranzani). Nello specifico, il sistema deve essere in grado di:

- **Contare** il numero di persone attualmente presenti nella stanza. Il rilevamento deve essere effettuato utilizzando opportuna sensoristica (es. sensori PIR) posti in prossimità del varco/porta d'accesso; ovviamente, occorre distinguere eventi di entrata/uscita dal varco. *Non si richiede l'utilizzo di camere e di tecniche di computer vision.*
- **Rilevare** ed acquisire parametri ambientali della stanza, quali: temperatura, umidità, pressione.
- Prevedere lo **storage dei dati** acquisiti (numero di persone nell'ambiente, parametri ambientali di cui sopra) in opportuno time-series database realizzato in INFLUX 2.0.
- Prevedere opportune **interfacce di visualizzazione dei dati** dei sensori mediante il tool GRAFANA, o mediante interfacce custom.
- Fornire **tecniche di forecasting** del numero di utenti nella stanza (in altre parole: prevedere quanti utenti saranno presenti all'interno della stanza nei prossimi X minuti) e del valore dei sensori ambientali. I valori della previsione devono di nuovo essere memorizzati in INFLUX.
- Prevedere **meccanismi di alerting** nel caso in cui vengano rilevate le seguenti situazioni di allarme: (i) il numero di utenti è maggiore di una soglia prestabilita; (ii) i valori di temperatura/umidità/pressione sono superiori/inferiori a soglie di riferimento. Le notifiche di allarme sono indirizzate all'amministratore della struttura e possono consistere nell'invio di un'email, di un'invocazione di API REST, di un messaggio su profilo TELEGRAM etc (il tipo di notifica è a scelta del gruppo).

1.1 Componenti aggiuntive

- +1 punto. **Valutazione delle prestazioni:** quanto è accurato il sistema di contatore degli accessi? qual è il Mean Square Error (MSE) del sistema di forecasting per i singoli sensori?
- +0.5 punto. **Web of Things:** progettare il sistema mediante l'architettura W3C Web of Things presentata a lezione. Individuare le Things ed implementare le rispettive Thing Description (TD).
- +0.5 punto. **Multi-sensor forecasting:** Studiare eventuali correlazioni tra i valori dei sensori ambientali e del contatore di presenza. Es. è possibile prevedere il numero di persone nella stanza sulla base delle variazioni dei sensori ambientali?

1.2 Tecnologie da utilizzare

E' possibile utilizzare un qualsiasi linguaggio di programmazione o librerie a scelta dello studente. In maniera simile per quanto riguarda la

parte hardware, è possibile utilizzare una qualsiasi piattaforma di prototipizzazione IoT. Se si vuole limitare al minimo la prototipizzazione circuitale, si consiglia l'utilizzo di board integrate con sensoristica (es. Arduino BLE Sense).

2 Proposta 2 (IoT Bridge Platform)

Si vuole realizzare un framework software per facilitare l'acquisizione dati da dispositivi IoT mappati su protocolli di sessione differenti (MQTT, CoAP, HTTP), con funzionalità avanzate di **protocol bridging** e di **data persistence**, sul modello (molto semplificato) del tool PONTE¹. Nello specifico, il framework deve consentire l'acquisizione dati da una sorgente IoT generica, con supporto verso i seguenti protocolli:

- **MQTT**: in questo caso, l'utente specifica l'indirizzo del broker (si può assumere sia MOSQUITTO), la porta, ed il topic di interesse;
- **CoAP**: in questo caso, l'utente specifica l'URL del server CoAP, la porta, e la modalità di messaging (confirmable/non confirmable);
- **HTTP**: in questo caso, l'utente specifica l'URL del server HTTP, la porta e la primitiva di lettura (GET/POST).

Si assume che -indipendentemente dalla sorgente e dal protocollo- il formato dei messaggi sia noto a priori e consista in una singola misurazione (ossia il valore prodotto da un sensore, senza ulteriori informazioni).

A valle dell'acquisizione, l'utente può scegliere l'output del flusso dati IoT, considerando tre alternative messe a disposizione dal framework:

- **Data visualization**. Stampa a video dei dati acquisiti in real-time.
- **Data storage**. Memorizzazione dei dati acquisiti su time-series database realizzato in INFLUX 2.0. In questo caso, l'utente deve specificare l'host su cui è installata l'istanza di INFLUX (può non essere la stessa macchina su cui viene eseguito il framework), le credenziali d'accesso, il nome del bucket, eventuali tag aggiuntivi definiti dall'utente. I dati acquisiti devono essere automaticamente salvati all'interno del database INFLUX.
- **Data bridging** verso altri protocolli di sessione. In questo caso, l'utente specifica il protocollo di destinazione (diverso ovviamente da quello di acquisizione) ed i parametri di configurazione precedentemente specificati per quel protocollo. Viene quindi costruito un flusso dati in output basato sul protocollo selezionato, al quale possono connettersi eventuali client esterni (indipendenti dal framework oggetto della proposta). Si chiede di prevedere meccanismi di bridge tra almeno due protocolli distinti (es. acquisizione in CoAP, output in HTTP o viceversa); ovviamente il caso ideale è quello in cui il sistema è perfettamente ortogonale in riferimento ai protocolli di acquisizione in ingresso e di bridge in output.

¹<https://www.eclipse.org/ponte/>

Il tool deve prevedere un'opportuna lista di comandi in modalità **CLI (Command Line Interface)** per consentire all'utente di eseguire le funzionalità richieste. Esempio di comandi:

```
./bridgeIoT -protocol mqtt -host 130.136.2.70  
  
-topic temperature -command visualize
```

Il comando di cui sopra consente di acquisire dati da sorgente MQTT e di visualizzare a video i dati in tempo reale, man mano che vengono prodotti dalla sorgente dati MQTT.

```
./bridgeIoT -protocol mqtt -host 130.136.2.70  
  
-topic temperature -influxconf influx.json -command save
```

Il comando di cui sopra consente di acquisire dati da sorgente MQTT e di memorizzarli su database INFLUX. I parametri di configurazione dell'istanza di INFLUX sono dettagliati nel file `influx.json`.

```
./bridgeIoT -protocol mqtt -host 130.136.2.70  
  
-topic temperature -command translate  
  
-dstprotocol coap -config coap.json
```

Il comando di cui sopra consente di acquisire dati da sorgente MQTT e di creare un flusso in uscita basato su protocollo CoAP. I parametri di configurazione del servizio CoAP sono dettagliati nel file `coap.json`.

2.1 Componenti aggiuntive

- +1 punto. **Interfaccia Grafica**: prevedere lo sviluppo di una GUI (oltre alla CLI) per facilitare l'inserimento dei comandi da parte dell'utente.
- +1 punto. **Filtering**: oltre alle funzionalità già menzionate di data visualization, storage e bridge, si vuole prevedere una funzionalità di Filtering, attraverso la quale vengono filtrati i dati in ingresso PRIMA di essere visualizzati, salvati su INFLUX, o trasferiti su altro protocollo di sessione. L'utente può specificare in questo caso delle soglie (minimo/massimo) per filtrare i dati acquisiti dalla sorgente IoT; i dati filtrati sono automaticamente scartati.

2.2 Tecnologie da utilizzare

E' possibile utilizzare un qualsiasi linguaggio di programmazione o librerie a scelta dello studente.