



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Sistema IoT wireless

per il rilevamento di variabili ambientali,
l'archiviazione remota e la
visualizzazione dei dati

Candidato
Paula Mihalcea

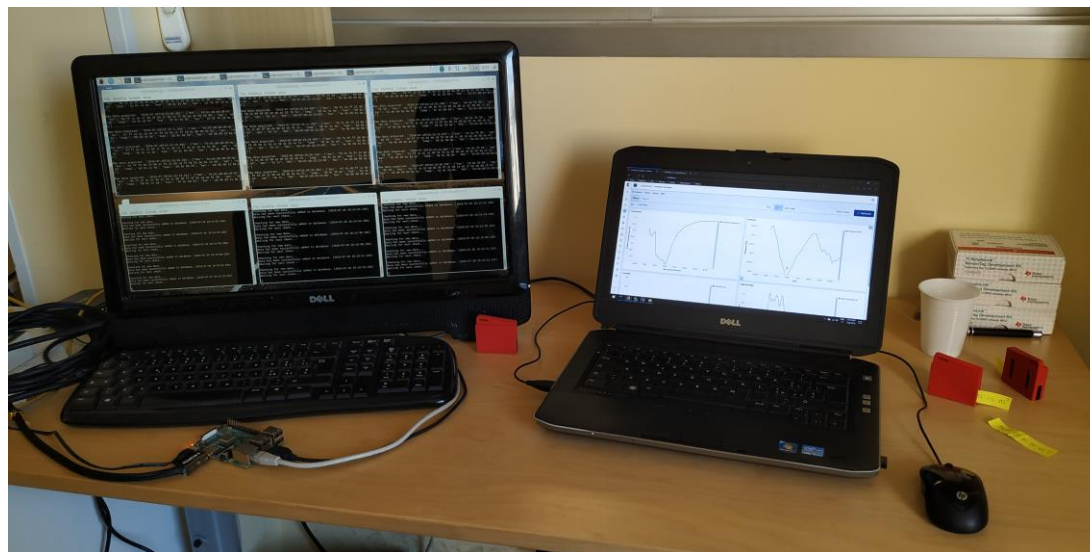
26 novembre 2019
Scuola di Ingegneria - Santa Marta

Relatore
Prof. Andrew D. Bagdanov

Correlatore
Walter Nunziati

Sommario

- Internet of Things
 - Smartyard – SY 4.0
- Strumenti utilizzati
- Obiettivi e architettura
- Connessione ed automatizzazione
- Elaborazione dei dati
- Archiviazione e visualizzazione
- Risultati, conclusioni e sviluppi futuri



magenta
softwarelab

Postazione di lavoro presso gli uffici di **Magenta s.r.l.**, dove ha preso vita il prototipo del progetto Smartyard – SY 4.0, ovvero la presente tesi.

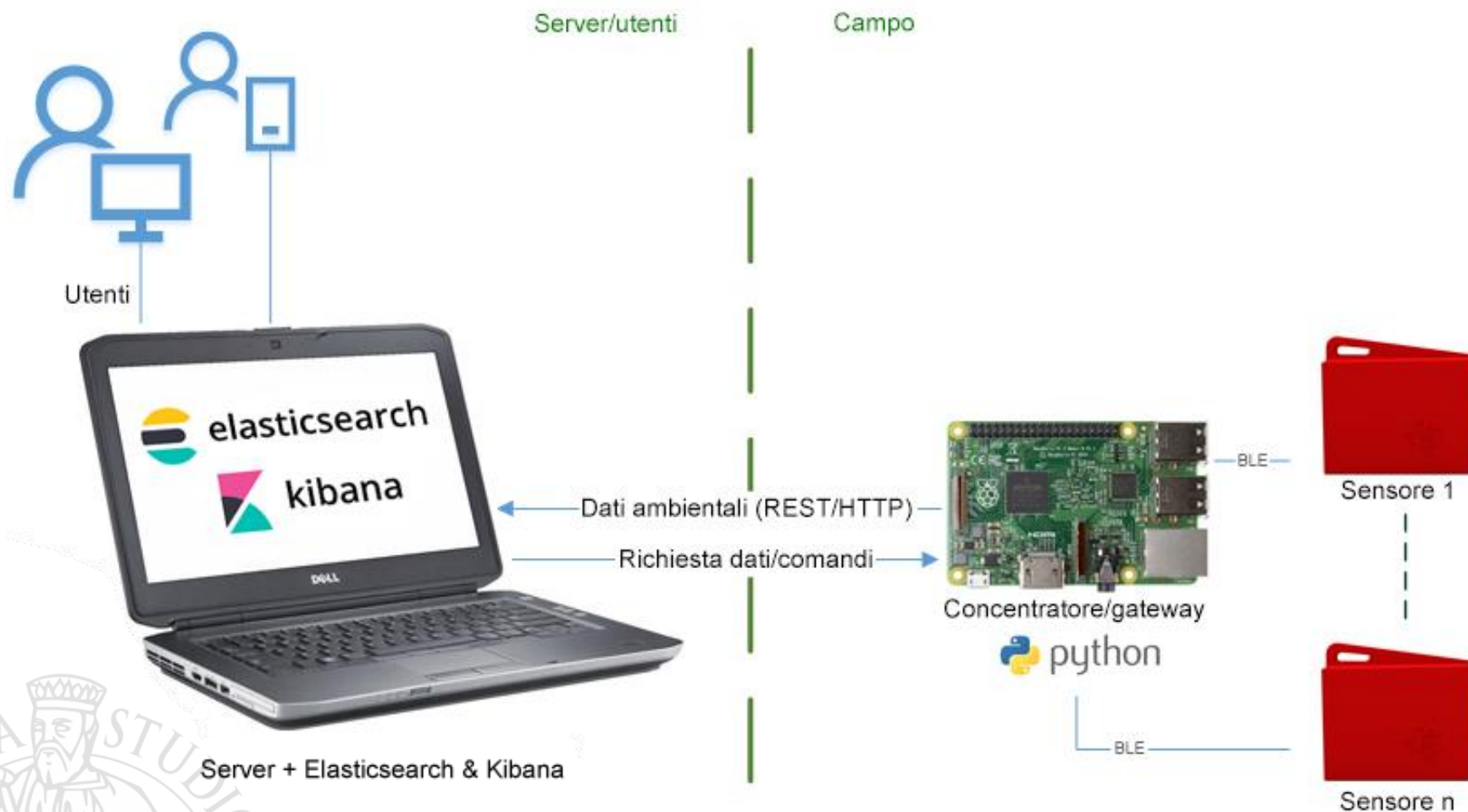
Internet of Things

- **Dispositivi interconnessi** tramite Internet o reti ad hoc
- Inseriti in oggetti di uso comune e dotati di **sensori**
- Utilizzati per **raccogliere dati** ed inviarli attraverso la rete
- I dati vengono salvati ed **analizzati** per ricavarne informazioni
- Utili anche per **inviare comandi** ad altri dispositivi
- Analisi statistiche prevedono 20 miliardi di dispositivi connessi in rete nel 2020

Smartyard SY 4.0

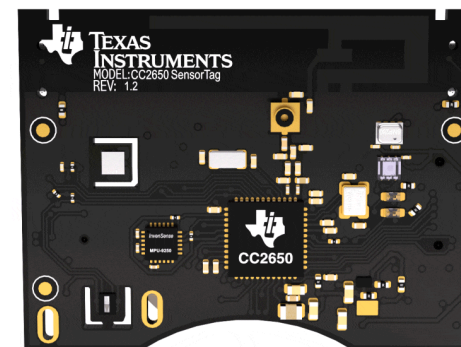
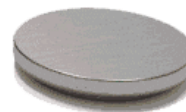
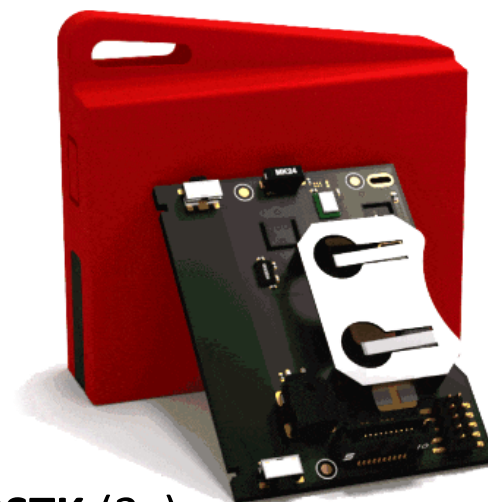
- Necessità nei luoghi di lavoro di **rilevare potenziali pericoli** ed **allertare in tempi utili** gli operatori per impedire danni
- **Smartyard SY 4.0**: progetto IoT dell'azienda **Magenta s.r.l.** nell'ambito della **cantieristica edile** e dell'**Industria 4.0** per:
 - evolvere il cantiere in una fabbrica intelligente integrando varie discipline ingegneristiche
 - studiare e sviluppare dispositivi e software innovativi
 - creare un **ambiente di lavoro sicuro** attraverso l'installazione di sensori per rilevare ed analizzare variabili ambientali

Smartyard – SY 4.0



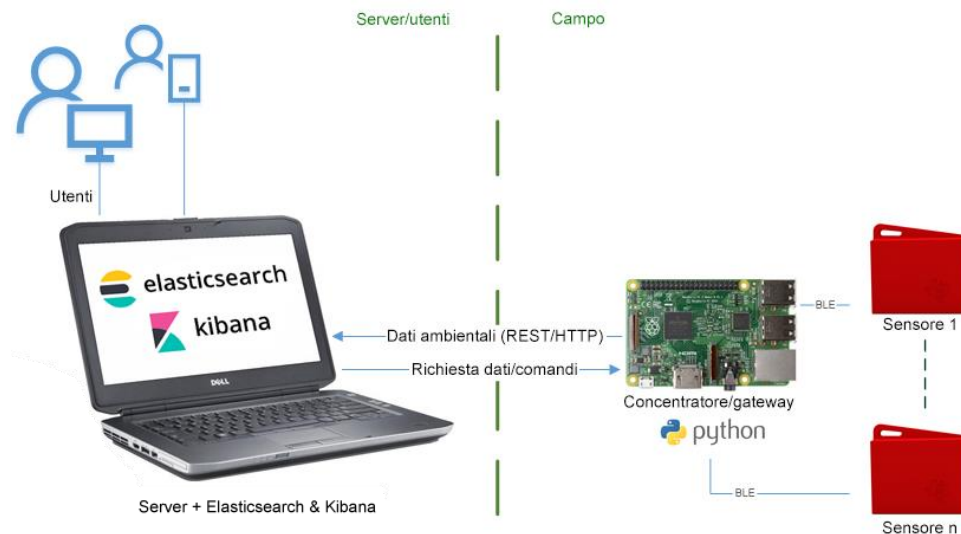
Strumenti

- **Computer** (*Dell Latitude E5430, Windows 10*)
 - Python 3
 - Elasticsearch
 - Kibana
- **Raspberry Pi 3 B+** (*Raspbian GNU/Linux 9*)
 - Stack Bluez + relative librerie
 - gatttool
 - Python 3
- **Texas Instruments SimpleLink SensorTag CC2650STK (3x)**
 - microcontrollore con processore ARM Cortex M3
 - sensori di temperatura, umidità, pressione, intensità luminosa e movimento (giroscopio, accelerometro, magnetometro)
 - connessione Bluetooth Low Energy (BLE)
 - batteria rimovibile (CR2032)
 - dimensioni: 6,7 x 5,0 x 1,4 cm



Obiettivi

- Stabilire ed automatizzare **connessione BLE e raccolta dati**
- **Elaborare** i dati grezzi e salvarli in un formato adeguato
- **Archiviare** i dati raccolti
- **Visualizzare i dati** archiviati, eventualmente fornendo un'interfaccia grafica all'utente finale



Architettura software

1. INIZIALIZZAZIONE

- Avvio RPi, SensorTag, server e relativi script

2. CONNESSIONE

- Acquisizione impostazioni
- Connessione
- Configurazione sensori
- Lettura sensori
- Salvataggio dati

3. ELABORAZIONE e ARCHIVIAZIONE

- Acquisizione impostazioni database
- Connessione al database
- Selezione log
- Inserimento log vecchi
- Aggiornam. database
 - lettura log grezzi
 - dataframe Pandas
 - dizionario JSON
 - invio dati al server
 - aggiornamento data ultimo invio

4. VISUALIZZAZIONE

- Creazione visualizzazioni e dashboard (Kibana)

Connessione e automatizzazione

- **gatttool** (Linux) permette di individuare i dispositivi BLE presenti nel range del Raspberry Pi
- Una volta stabilita la connessione è possibile **attivare** i sensori del SensorTag e richiedere loro una lettura delle variabili ambientali attraverso **comandi da terminale**, che restituiscono i dati in codifica esadecimale little-endian:

Characteristic/value descriptor:
04 95 73 59

- L'interpretazione dei dati è possibile grazie alle tabelle ed algoritmi presenti nella **documentazione ufficiale** del SensorTag
- **Poco pratico**, adatto solo per test e debug

- La libreria Python **Pexpect** permette di aprire un processo interattivo da terminale
- Lo script **connect.py** (sviluppato *ad hoc*) esegue la completa automatizzazione della connessione in cinque fasi:

2. CONNESSIONE

- Acquisizione impostazioni
- Connessione
- Configurazione sensori
- Lettura sensori
- Salvataggio dati

Elaborazione dei dati

3. ELABORAZIONE e ARCHIVIAZIONE

- Acquisizione impostazioni database
- Connessione al database
- Selezione log
- Inserimento log vecchi

- Aggiornam. database

- lettura log grezzi
- dataframe Pandas
- dizionario JSON
- invio dati al server
- aggiornamento data ultimo invio

- Lo script ***process_data.py*** realizza un loop infinito di elaborazione dei dati ed aggiornamento del database
- **Elabora i dati grezzi, decodificandoli e trasformandoli** prima in un **dataframe Pandas**, poi in formato **JSON**, adatto ad essere inviato al server.
- Può essere **avviato in qualunque momento**, in quanto controlla anche i log vecchi e riprende l'aggiornamento del database dagli ultimi dati inseriti

Archiviazione e visualizzazione

elasticsearch

kibana

- **Server e database** open-source basato su Apache Lucene
- Senza schema, orientato ai documenti
- Ricerca in **tempo quasi reale**, indicizzazione automatica
- API **RESTful**, dati in formato **JSON**
- Distribuito, facilmente scalabile
- API Python: integrazione perfetta con lo script di elaborazione dati

- Strumento open-source per la **visualizzazione grafica** dei dati
- Parte di **The ELK Stack**, un insieme di applicazioni per la gestione dei server prodotto da Elastic.co
- **Perfettamente integrato** con il database Elasticsearch (esplorazione dei dati, query JSON, ecc.)
- Potente ed **altamente configurabile**, adatto sia agli analisti che agli utenti finali

Risultati

```
pi@raspberrypi:~/Smartyard-SY4.0 $ python3 connect.py 2
Connecting to 98:07:2D:32:23:80... (attempt no. 1/5)
Connection successful.
Starting sensor configuration...
Sensor configuration successful.
Data retrieval cycle started. Press CTRL+C to stop and disconnect.

Raw data acquired:  '2019-07-26T10:39:52.773': {'hum': 'dc 69 68 4a', 'bar'
: '1d 0b 00 4a 8a 01', 'temp': '1c 0c 00 0e', 'MAC': '98:07:2D:32:23:80', '
opt': '20 00', 'mov': '93 fe 6f ff 22 00 17 00 09 00 ce f7 4c 02 42 03 f3 f
e'}
```

Risultati

KQL



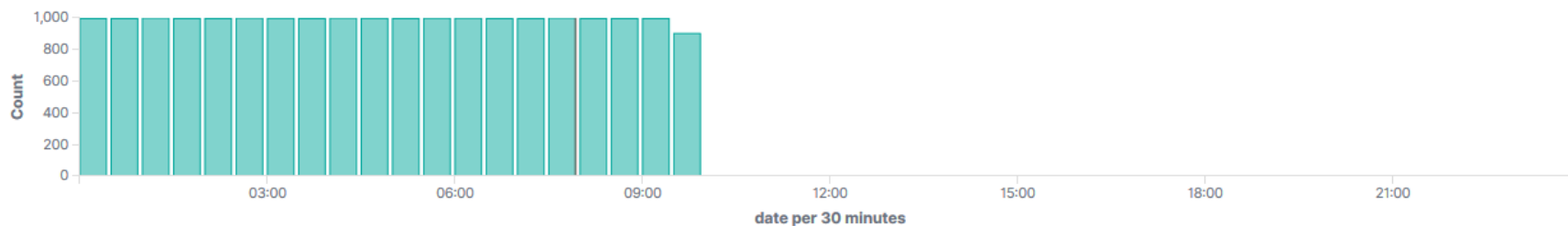
Today

Show dates

Refresh

Jul 26, 2019 @ 00:00:00.000 - Jul 26, 2019 @ 23:59:59.999 —

Auto

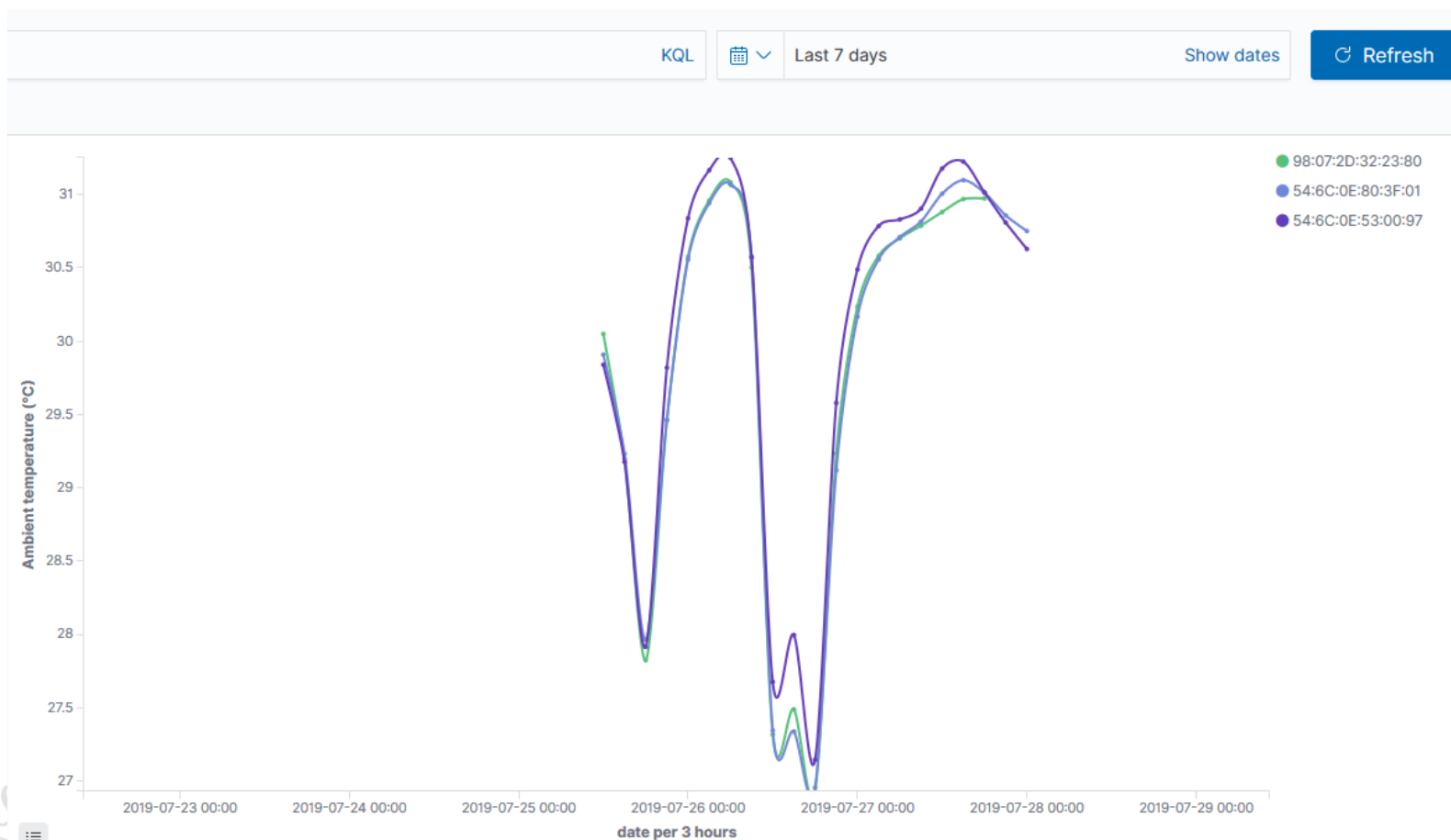


Time

_source

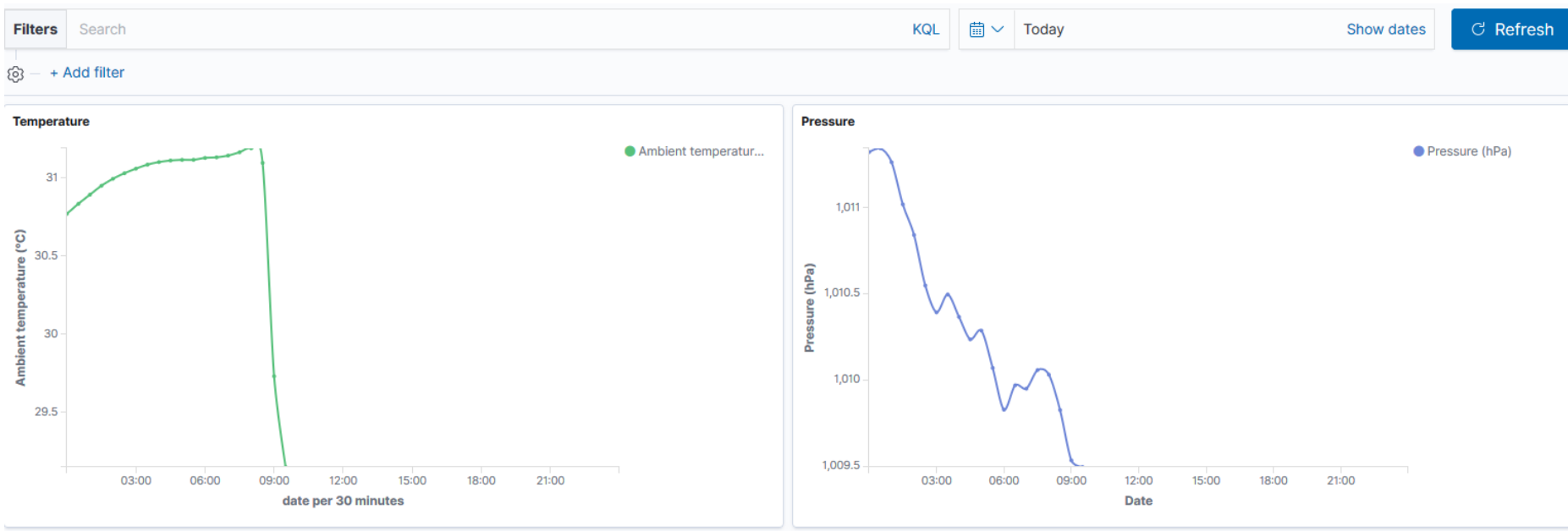
- > Jul 26, 2019 @ 09:55:31.189 `date: Jul 26, 2019 @ 09:55:31.189 hum: 35.52 opt: 30.52 bar: 1,009.03 temp: 28.5 MAC: 54:6C:0E:53:00:97 _id: 16JHLWwBDAlZtBXVN-J5`
`_type: sreading _index: sensortag _score: -`
- > Jul 26, 2019 @ 09:55:27.233 `temp: 28.56 bar: 1,009.96 hum: 33.28 opt: 12.65 MAC: 98:07:2D:32:23:80 date: Jul 26, 2019 @ 09:55:27.233 _id: 1KJHLWwBDAlZtBXVLOJd`
`_type: sreading _index: sensortag _score: -`
- > Jul 26, 2019 @ 09:55:26.843 `MAC: 54:6C:0E:80:3F:01 bar: 1,009.07 temp: 28.78 date: Jul 26, 2019 @ 09:55:26.843 opt: 31.27 hum: 33.39 _id: 1aJHLWwBDAlZtBXVLIy`
`_type: sreading _index: sensortag _score: -`
- > Jul 26, 2019 @ 09:55:23.979 `date: Jul 26, 2019 @ 09:55:23.979 hum: 35.52 opt: 30.64 bar: 1,009.04 temp: 28.5 MAC: 54:6C:0E:53:00:97 _id: 1qJHLWwBDAlZtBXVN-Jp`
`_type: sreading _index: sensortag _score: -`

Risultati





Risultati



Conclusioni

- **Obiettivi iniziali raggiunti**
- **Programma pronto all'uso** altamente configurabile, anche da utenti non esperti
- Possibilità di **riadattare facilmente gli script** sviluppati ad altri dispositivi BLE
- **Gestione automatica** della **perdita del segnale**, **riconnessione**, e **logging** delle (dis)connessioni
- **Range del SensorTag limitato**, connessione instabile sopra gli 8-9 m (ridotta a circa la metà con il segnale ostacolato da oggetti in linea d'aria)
- **Durata batteria insufficiente** (circa 55 ore, intervallo di polling di 5 secondi)
- Complessivamente **poco adatto ad usi esterni** e/o in condizioni non ideali

Sviluppi futuri

- Test del prototipo in **condizioni non ideali**
- Valutazione di **dispositivi diversi** dal SensorTag
- Integrazione del rilevamento dei **dati di movimento** ed altre variabili



Grazie per l'attenzione!



Tutto il software creato per questo progetto, assieme alla tesi contenente le istruzioni necessarie per l'installazione e l'utilizzo, è disponibile al seguente indirizzo:

<https://github.com/PaulaMiha1cea/Smartyard-SY4.0>