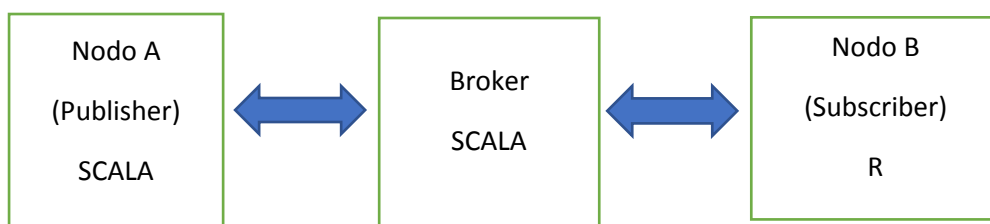


**Titolo progetto:** “Agriculture’s monitoring and health status

**Anno accademico:** 2018/2019

**Elaborato da:** Biagio Fornitto (parte in Scala) e Matteo Bonaccorso (parte in R)



L'architettura del sistema è composta da 3 nodi di cui 2 nodi realizzati attraverso il linguaggio Scala e uno realizzato in R. Il sistema emula un protocollo publish-subscribe in cui ogni nodo assolve una particolare funzione e comunica con gli altri attraverso delle API REST.

Il nodo A è un simulatore di sensori (temperatura, luminosità e umidità) e per ogni classe di sensori espone le seguenti API:

- Accensione sensori (GET /api/classe\_sensori/on)  
Accende tutti i sensori di una determinata classe (es. per la temperatura l'API invocata sarà /api/temp/on) e avvia la connessione (se non è ancora stata fatta) e la pubblicazione dei dati verso il broker invocando le relative API di quest'ultimo.
- Spegnimento sensori (GET /api/classe\_sensori/off)  
Spegne tutti i sensori di una determinata classe e arresta la pubblicazione dei relativi dati verso il broker. Se nessun altro sensore di nessun'altra classe è attivo sarà invocata le API di disconnessione del broker.
- Lettura (GET /api/classe\_sensori/read)  
Ritorna il valore generato dall'ultimo sensore della classe
- Settaggio numero sensori (POST /api/classe\_sensori/setSensors)  
Permette il settaggio del numero di sensori desiderati per classe, tale dato viene passato nel body della richiesta.

Il broker espone le seguenti API:

- **Connessione (POST /api/connect)**  
Viene invocata da tutti i client prima di pubblicare o sottoscrivere. Nel body della richiesta ogni client dovrà passare due parametri: un clientID che lo identifica univocamente e il suo indirizzo. Se al momento dell'invocazione dell'API il broker riconosce che ci sono due clientID uguali rifiuta la seconda connessione.
- **Pubblicazione (POST /api/publish)**  
Può venire invocata solo dai clients che si sono connessi con successo. Nel body trasporta i seguenti parametri: il ClientID che serve per identificare se il client è connesso, il topic che indica l'argomento su cui il client sta pubblicando e il dato del sensore. Appena il broker la riceve verifica se quel client si è connesso e verifica se esistono dei subscribers per quel topic. Se non ci sono subscribers scarta il dato, altrimenti invia loro il dato.
- **Sottoscrizione (POST /api/subscribe)**  
Anche questa API può venire invocata solo dai clients che si sono connessi. Nel body trasporta i seguenti parametri: il ClientID che serve per identificare se il client è connesso e il topic che indica l'argomento a cui è interessato il client. Appena viene invocata questa API il broker, dopo aver verificato la connessione del client, salva in una mappa l'informazione ricevuta (relazione topic->subscribers).
- **Annulamento sottoscrizione (POST /api/unsubscribe)**  
Il client che non è più interessato ai dati di un determinato topic invoca questa API. Nel body della richiesta sono presenti due parametri: il ClientID che serve per identificare se il client è connesso e il topic che indica l'argomento da cui il client vuole annullare la sottoscrizione. Il broker eliminerà l'informazione relativa all'associazione del client con il topic ricevuto nella richiesta.
- **Disconnessione (POST /api/disconnect)**  
Il client che ha finito di pubblicare i dati (publisher) o che non ha più interesse per un determinato topic (subscriber) invoca questa API del broker. Nel body della richiesta è presente il clientID del client che la invoca, così il broker può eliminarlo dalla lista dei suoi client. Nel caso in cui il client che si disconnette è un subscriber, il broker eliminerà anche l'informazione relativa ai topic a cui si era sottoscritto. Dopo di che, qualsiasi client che vorrà tornare ad interagire con il broker, dovrà invocare nuovamente l'API per la connessione.

Il nodo B svolge la funzione di subscriber esponendo le seguenti API:

- **Connect (POST /connect):** tramite servizio REST il nodo comunica al broker la volontà di connettersi passando anche il suo id e l'api su cui comunicare: un servizio REST che il broker chiamerà se il nodo è sottoscritto ad un topic.

- **Subscribe (POST /subscribe):** tramite servizio REST il subscriber comunica al broker il topic al quale si vuole sottoscrivere, il topic è comunicato nel body della richiesta in formato JSON.
- **Publish (POST /publish):** questa risorsa viene esposta durante la Connect quando il nodo è sottoscritto ad un topic, questo servizio viene richiamato per passare i dati pubblicati sul broker, una volta ricevuti vengono organizzati in un dataframe e salvati in formato "csv". I dati sono comunicati nel body della richiesta in formato JSON.
- **LastData (GET /lastdata):** questo servizio rest recupera l'ultimo dato inserito nel file "csv".
- **Grafico per topic (GET /plot):** questo servizio si occupa di plottare i dati prelevati dal file "csv" appartenenti al topic specificato come parametro, nel grafico viene tracciata una linea per ogni sensore di cui si ha il dato con colore differente, se non ci sono dati per quel topic viene restituito un grafico vuoto. Il topic viene passato come parametro nell'URL della richiesta con nome del parametro per l'appunto "topic", il risultato è restituito in formato PNG.
- **Grafico per topic e id (GET /plotid):** questo servizio si occupa di plottare i dati relativi a un sensore per un determinato topic quindi si avrà un unico insieme di valori per il grafico. Il topic e l'id del sensore vengono passati come parametro nell' URL della richiesta, il risultato è restituito in formato PNG.
- **Salute (GET /salute):** tramite un classificatore allenato all'avvio del servizio Rest questa risorsa restituisce lo stato di salute della coltivazione analizzando gli ultimi dati raccolti dal file "csv" lo stato di salute è indicato tramite parole le chiave "low", "med" (medium), "high".
- **Unsubscribe (POST /unsubscribe):** tramite questo servizio si comunica al broker il topic dal quale non si vogliono più ricevere dati. Il topic è comunicato nel body della richiesta in formato JSON.
- **Disconnect (POST /disconnect):** con questa risorsa il nodo si disconnette dal broker, da notare che la disconnessione comporta l'inevitabile annullamento dell'iscrizione da tutti i topic ai quale ci si era sottoscritti.

All'avvio del nodo viene settato un id casuale e viene fatto il training del classificatore. Quest'ultimo usa come dataset il file "dataset.csv" precedentemente creato. Inoltre, viene stampata a schermo la confusion matrix, se si vuole vedere com'è stato costruito il dataset si può controllare la funzione Dataset che non è esposta né come servizio REST né è richiamata all'interno del codice è stata mantenuta a livello puramente consultivo.