

Si implementi un'applicazione Web-based per la gestione di nodi sensore nell'ambito dell'Internet of Things (IoT). L'applicazione deve essere costruita a partire da un framework di base costituito da un IoT Agent al quale è possibile collegare sia nodi sensore che un'interfaccia utente per la gestione dei nodi e il monitoraggio degli ambienti in cui i nodi sono collocati (vedi Figura 1).

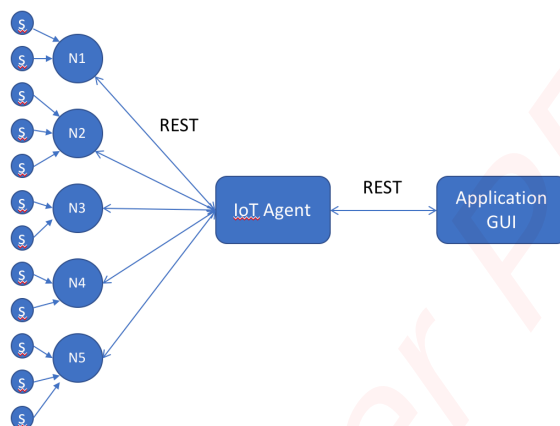


Figura 1 - Architettura del sistema

Un IoT Agent è un componente software che si colloca tra dispositivi IoT (nodi sensore o attuatori) e l'applicazione. Può gestire gruppi di dispositivi, raccogliere i loro dati e restituirli in forma aggregata ad altre componenti applicative.

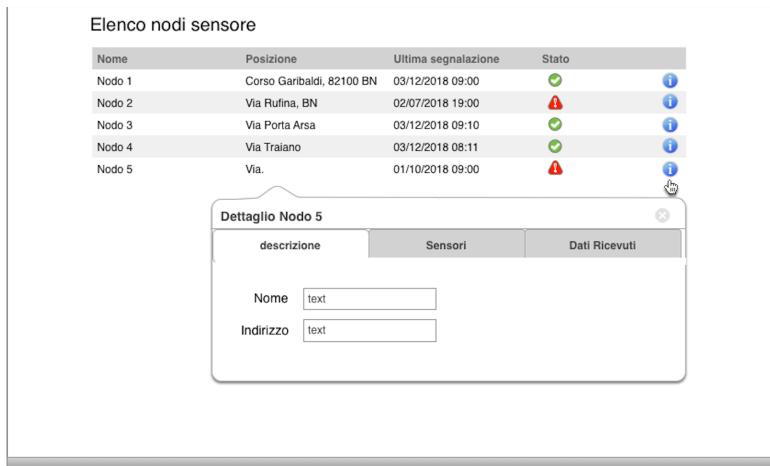
Assumiamo che i dispositivi IoT siano solo nodi sensore caratterizzati da una posizione geografica e composti di uno o più sensori associabili ad attributi di contesto (es. temperatura, umidità, pressione atmosferica, luminosità, ecc.).

Ogni sensore può acquisire periodicamente dati (campioni) dall'ambiente e inviarli all'IoT Agent. Interrogando l'IoT Agent possiamo conoscere sia l'ultimo campione per ciascun attributo di contesto di un nodo sensore sia la media di una finestra di N campioni. Inoltre, l'IoT Agent può fornire informazioni sui nodi gestiti e sui sensori presenti sui diversi nodi.

Si memorizzino i nodi, i sensori e i dati inviati all'IoT Agent in memoria¹. Si implementino semplici nodi sensore virtuali in grado di generare campioni periodicamente. Un nodo può essere un processo caratterizzato da una posizione geografica e da un insieme di sensori (uno per attributo di contesto). L'attività di ogni sensore può essere modellata con un thread di controllo. Un sensore sarà caratterizzato dall'attributo di contesto (es. temperatura) e da alcuni parametri di configurazione come la frequenza di acquisizione e di spedizione dei dati acquisiti.

Si realizzi in HTML, CSS e JavaScript un'interfaccia utente che consenta di gestire i nodi sensore registrati all'IoT Agent e di osservare i campioni degli attributi di contesto che ciascun sensore acquisisce e invia all'IoT Agent. Si vedano gli esempi di GUI mostrati di seguito.

¹ Opzionalmente è possibile gestire i dati in modo persistente in un DB.



Elenco dei sensori registrati all'IoT Agent con la visualizzazione di alcune informazioni di sintesi, quali nome del nodo, posizione del nodo, ultima segnalazione ricevuta, stato del nodo.

Al click sull'icona "info" dovrà essere visualizzata la scheda di dettaglio del nodo. La scheda riporta la descrizione del nodo sensore, l'elenco dei sensori collegati al nodo e i dati inviati dal nodo con riferimento ai sensori ad esso associati.

Figura 2 - Interfaccia principale per l'accesso all'elenco dei nodi sensore

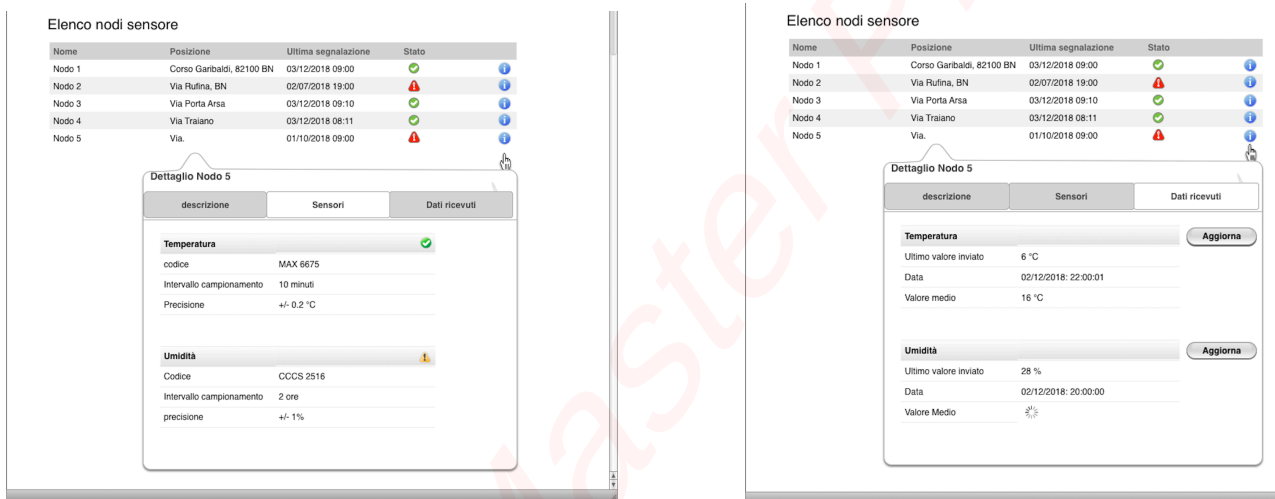


Figura 3 - Dettaglio di un nodo: (a) caratteristiche dei sensori; (b) ultimo dato acquisito da ciascun sensore

La Figura 3 riporta i contenuti dei tab "Sensori" e "Dati ricevuti" della scheda relativa ad uno specifico sensore. Le interfacce utente mostrate in questo documento sono indicative. E' possibile personalizzarle e arricchirle compatibilmente con le funzionalità offerte dall'IoT Agent ed esposte attraverso la sua REST API. Per completezza, di seguito si riporta l'interfaccia REST dell'IoT Agent.

```
package it.unisannio.iot;
```

```
public interface IoTAgentService {
```

```
// @GET /nodes /* invoked by consumers */
// @GET /nodes/{nodeId} /* invoked by consumers */
// @GET /nodes/{latitude}&{longitude}&{radius} /* invoked by consumers */
// @GET /nodes/{nodeId}/sensors/{attribute} /* invoked by consumers */
// @GET /nodes/{nodeId}/sensors/{attribute}/samples/{sampleId} /* invoked by consumers */
// @GET /nodes/{nodeId}/sensors/{attribute}/samples_average /* invoked by consumers */
// @POST /nodes /* invoked by nodes */
// @POST /nodes/{nodeId}/sensors /* invoked by nodes */
// @POST /nodes/{nodeId}/sensors/{attribute}/samples /* invoked by nodes */
// @PUT /nodes/{nodeId} /* invoked by nodes */
// @PUT /nodes/{nodeId}/sensors/{attribute} /* invoked by nodes */
// @DELETE /nodes/{nodeId} /* invoked by nodes or infrastructure */
// @DELETE /nodes/{nodeId}/sensors/{attribute} /* invoked by nodes */
// @DELETE /nodes/{nodeId}/sensors/{attribute}/samples/{sampleId} /* invoked by nodes or consumers */
}
```