

Il Gestore IoT è un eseguibile a sé stante che svolge le seguenti operazioni:

Potremmo avere un oggetto “campo” che contiene un oggetto sense\_umid e un oggetto att\_irrig, inoltre ha un oggetto che salva il livello di scorta d’acqua per la giornata e quella attualmente disponibile (o quella già consumata). Espone metodi come “getUmidita” o “setStatoAtt”

sense\_umid è un oggetto capace di dialogare con i veri sensori, se li abbiamo, oppure un oggetto che emula un sensore se non abbiamo quello reale. att\_irrig è in grado di comunicare con l’attuatore reale, oppure emula l’attuatore mantenendo solo in memoria lo stato attuale.

```
while (true)
```

```
{    if <<inizio_giornata>> then reset clock e rinnova scorte acqua // prima del reset potremmo registrare se si è avanzata acqua o se c’è stata carenza. Il rinnovo delle scorte può avvenire in base ad una interrogazione fatta al backend oppure ricevendo un messaggio su MQTT
```

```
        valUmid = campo.getUmidita(); // eventualmente da ripetere su un array di campi se il Gestore IoT ne gestisce più di uno
```

```
        invia al broker su topic farmAPI/<IDcampo>/sensUmid il messaggio contenente {clock, valUmid }
```

```
        newStatoAtt = f(valUmid,altri parametri)
```

```
        if newStatoAtt <> campo.getStatoAtt() // è cambiato lo stato
```

```
            campo.setStatoAtt(newStatoAtt)
```

```
            invia al broker su topic farmAPI/<IDcampo>/statoAtt il messaggio contenente {clock, newStatoAtt }
```

```
            campo.aggiornaScortaAcqua; // se l’attuatore è ON la scorta viene decrementata di un tot da definire, altrimenti rimane invariata
```

```
            clock++;
```

```
            sleep(DELTA T) // passo di avanzamento del clock
```

```
}
```

Nota: quando leggiamo l’umidità, se il metodo è implementato da un emulatore, dovremmo con qualche criterio anche aggiornarla (es. se attuatore ON o piove l’umidità aumenta di un tot, se OFF e piove aumenta un po’ meno, se OFF e c’è sole diminuisce), se invece è implementato da una classe capace di interagire con i veri sensori il valore restituito sarà quello reale letto dai sensori e se analogamente l’attuatore questo sarà comandato da una classe capace di interagire con l’attuatore per accenderlo e spegnerlo.

Nota: la funzione f(valUmid,altri parametri) racchiude il criterio di controllo degli attuatori: dovrà anche tenere conto della riserva d’acqua disponibile. Eventualmente possiamo usare un terzo stato BLOCK per indicare che l’attuatore è bloccato in attesa che vengano ripristinate le scorte d’acqua.

L’inizio giornata è impostato per esempio tramite un messaggio su un topic farmAPI/segnaleOrario