

Sensor hub

Riccardo Carta , mat. 1231954

Relazione progetto Programmazione a Oggetti

gruppo:

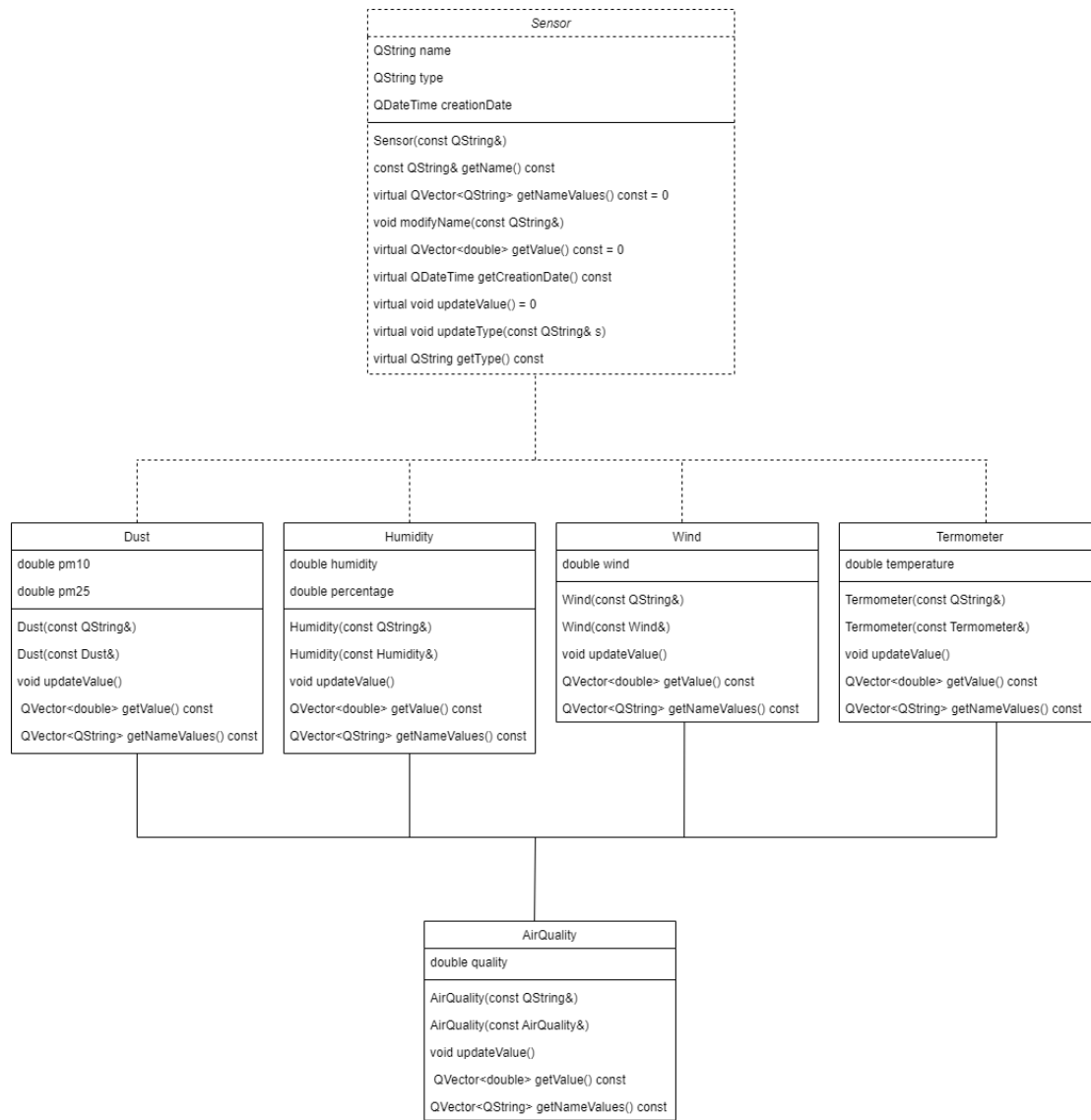
Bressan Alessandro, mat. 1224823

Carta Riccardo, mat. 1231954

1. Introduzione

Il progetto "Sensor hub" si propone di fornire un'interfaccia grafica per la gestione e la visualizzazione di dati provenienti da diversi tipi di sensori ambientali. L'applicazione permette agli utenti di monitorare in tempo reale vari parametri ambientali come PM10, PM2.5, umidità, temperatura, velocità del vento e qualità dell'aria attraverso un'interfaccia suddivisa in 2 macroaree: una per la gestione dei sensori che permette la loro aggiunta eliminazione e modifica, ed una per la loro simulazione in un grafico.

2. Descrizione del Modello



Il modello logico del progetto si articola attorno a diverse classi chiave che rappresentano i concetti fondamentali dell'applicazione. Al centro vi è la classe `Sensor`, classe astratta che contiene le informazioni comuni a tutti i sensori, derivata in sotto-classi specifiche che vanno a rappresentare un determinato tipo di sensore. La classe `AirQuality` non è esattamente un sensore ma è da intendersi come un insieme di sensori che, messi insieme, danno una stima della qualità media dell'aria. Ogni sensore ha lo scopo di restituire gli ipotetici valori captati nell'ambiente e potenzialmente anche più di uno alla volta, come ad esempio un sensore `Dust` che rileva la concentrazione nell'aria sia di `pm10` che `pm25`, ma condividono comunque lo stesso metodo `getValue`.

3. Polimorfismo

L'intera schermata di simulazione viene costruita a partire da un vettore di sensori estrapolato dal json attualmente aperto. Sia la simulazione nel grafico sia la simulationbar, che contiene una visualizzazione semplificata dei sensori, vengono create e modificate interattivamente senza necessità di conoscere il tipo del sensore.

4. Persistenza dei Dati

I dati dei sensori sono gestiti attraverso file JSON utilizzando librerie integrate di Qt per l'I/O. Questo approccio offre una persistenza dei dati robusta consentendo agli utenti di salvare configurazioni personalizzate. E' presente un file JSON centrale, utilizzato per il salvataggio dei sensori senza necessità di creare ulteriori file esterni, che verrà utilizzato automaticamente all'avvio riportando lo stato dell'interfaccia di gestione dei sensori all'ultima sessione di utilizzo dell'applicazione. E' comunque possibile salvare o importare i sensori da un file esterno.

5. Funzionalità Implementate

Come detto in precedenza l'interfaccia è divisa in 2 schermate

-Hub di gestione dei sensori:

Offre una visualizzazione comoda ed ordinata dei sensori e la possibilità di crearne di nuovi o modificare quelli esistenti, presenta inoltre una barra di ricerca per nome che risponde in tempo reale.

-Schermata di simulazione:

La simulazione permette di visualizzare su un piano cartesiano i valori di uno o più sensori contemporaneamente (differenziandoli tramite colori diversi indicati dal colore del tasto corrispondente) oltre che aggiornarli anche singolarmente. E' presente uno zoom per permettere una visione dei valori captati in precedenza. La legenda in alto a sinistra indica le unità di misura utilizzate nella simulazione, alcuni di questi (come indicato anche nella legenda) sono stati moltiplicati o divisi per 10 per permettere a valori con scale molto diverse di rientrare nella stessa schermata

E' presente una menubar in alto che permette di navigare tra le schermate oltre che salvare o importare file tramite apposito menu a tendina

6. Suddivisione delle attività

Bressan Alessandro:

- persistenza dei dati
- interfaccia di gestione dei sensori
- menubar

Carta Riccardo:

- schermata di simulazione
- modello logico dei sensori

7. Rendicontazione ore

Attività	Ore Previste	Ore Effettive
Studio e progettazione	10	10
Sviluppo del codice del modello	10	10
Studio del framework Qt	20	30
Sviluppo del codice della GUI	20	30
Test e debug	10	5
Stesura della relazione	5	3
totale	75	88

Come previsto il la maggior parte delle ore sono state spese nell'imparare ad utilizzare la libreria di QT. Le ore di studio e sviluppo sono uguali dato che, pur avendo iniziato a scrivere solo dopo uno studio teorico delle basi della libreria, in fase di sviluppo è stato necessario più di una volta tornare a studiare rendendo la linea di separazione tra studio e sviluppo molto labile. In compenso non ci sono stati particolari bug o malfunzionamenti che, più che altro, sono stati trovati e risolti in fase di sviluppo.