

Relazione progetto Programmazione a Oggetti

Tiozzo Matteo, matr. 2042882

Eghosa Matteo Igbiniedion Osamwonyi, matr. 2042888

Sensors Data Factory

Introduzione

Sensors Data Factory è un'applicazione scritta interamente in linguaggio C++ e progettata per aziende che impiegano una varietà di sensori, tra cui quelli per gas, temperatura e movimento. Questa applicazione consente la gestione efficiente di diversi tipi di sensori, quali accelerometri, giroscopi, sensori elettrochimici, sensori semiconduttori, termoresistenze e termocoppie. Ogni sensore è identificato univocamente per nome, modello e tipologia di misurazioni effettuate.

Descrizione del modello

Il modello si articola in:

- Gerarchia dei sensori
- Classe `SensorsDashboard`: permette la gestione dei vari sensori. Accompagnata da scorciatoie da tastiera.
- Classe `CreatorSensors`: permette la creazione dei sensori
- Classe `Model`: per la comunicazione con il controller

Modello Logico

Il modello logico comprende la gestione dei tipi di sensori e la visualizzazione dei sensori stessi. La prima parte include le classi che descrivono i sensori e alcune classi di servizio per convertire i sensori nel formato JSON e salvarli su file. Il modello parte da una classe astratta `Sensor` che rappresenta le informazioni comuni a tutti gli oggetti cercabili. Le classi concrete rappresentano tipi specifici di sensori. Tutti i sensori hanno un metodo `addMisuration` per effettuare misurazioni. Il design si ispira al pattern Model-view Controller. Un `Data Sensor` funge da contenitore in cui è possibile aggiungere, rimuovere e cercare sensori. La ricerca avviene attraverso un oggetto `Sensors`, che contiene informazioni relative alle parole cercate e ai filtri. Gli oggetti ottenuti vengono visualizzati nella pagina `CreateView`.

Polimorfismo

Il polimorfismo è utilizzato nel design pattern Model-View Controller, più precisamente all'interno della gerarchia `Sensors`. Viene utilizzato per la simulazione del calcolo delle misurazioni, la conversione in JSON e la costruzione dei widget per mostrare diverse tipologie di elementi. La visualizzazione prevede tre possibili rese grafiche per ciascun sensore: `CreateView`, `Modify`, `Graph View`.

Persistenza dei Dati

I dati vengono salvati in formato JSON in un unico file per tutti i tipi di sensori. Le misurazioni sono salvate in un dizionario. Vi è inoltre la possibilità di salvare e caricare in qualsiasi cartella presente nel proprio dispositivo.

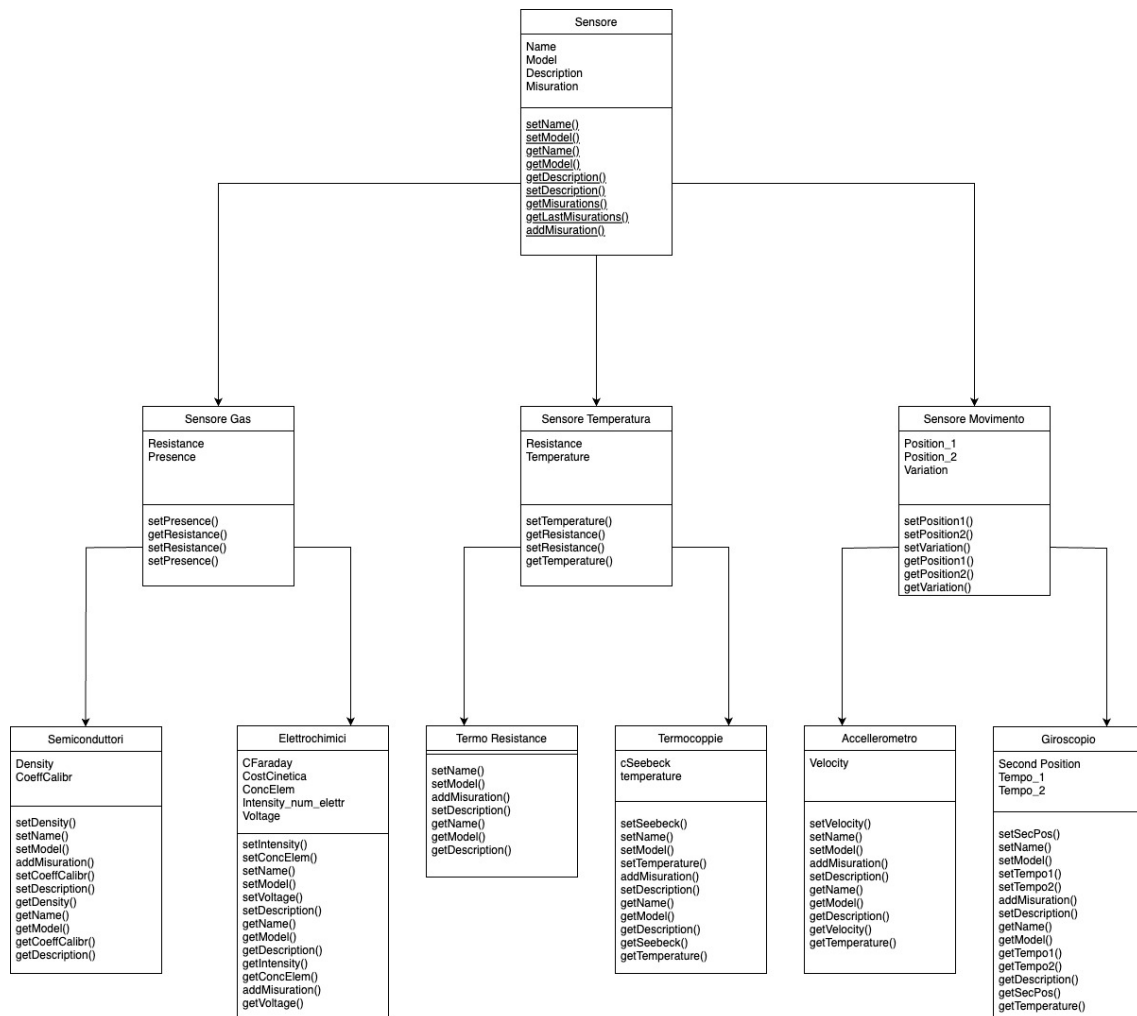


Figure 1: Schema delle classi

Funzionalità Implementate

Le funzionalità implementate includono la gestione di tre tipi di sensori, la conversione e il salvataggio in formato JSON, la ricerca con pertinenza, filtro e paginazione.

Funzionalità Grafiche

Le funzionalità grafiche comprendono una barra dei menù, shortcut da tastiera, un grafico per la visualizzazione delle misurazioni, l'utilizzo di icone per distinguere i tipi di sensori, gestione del ridimensionamento, navigazione attraverso diverse schermate tramite interfaccia, utilizzo di immagini e pulsanti nella visualizzazione degli oggetti, utilizzo di colori e stili grafici, effetti grafici come cambio del colore al passaggio del mouse.

Rendicontazione Ore

Attività	Ore Previste	Ore Effettive
Studio e progettazione	10	15
Sviluppo del codice del modello	10	20
Studio del framework Qt	10	10
Sviluppo del codice della GUI	10	13
Test e debug	5	7
Stesura della relazione	5	5
Totale	50	70

Suddivisione Attività

Attività	Assegnamento
Sviluppo del modello	Eghosa
Sviluppo della GUI	Matteo e Eghosa
Sviluppo del grafico	Matteo e Eghosa
Sviluppo del controller	Eghosa
Salvataggio su file	Matteo
Caricamento da file	Matteo