Relazione progetto Programmazione ad Oggetti

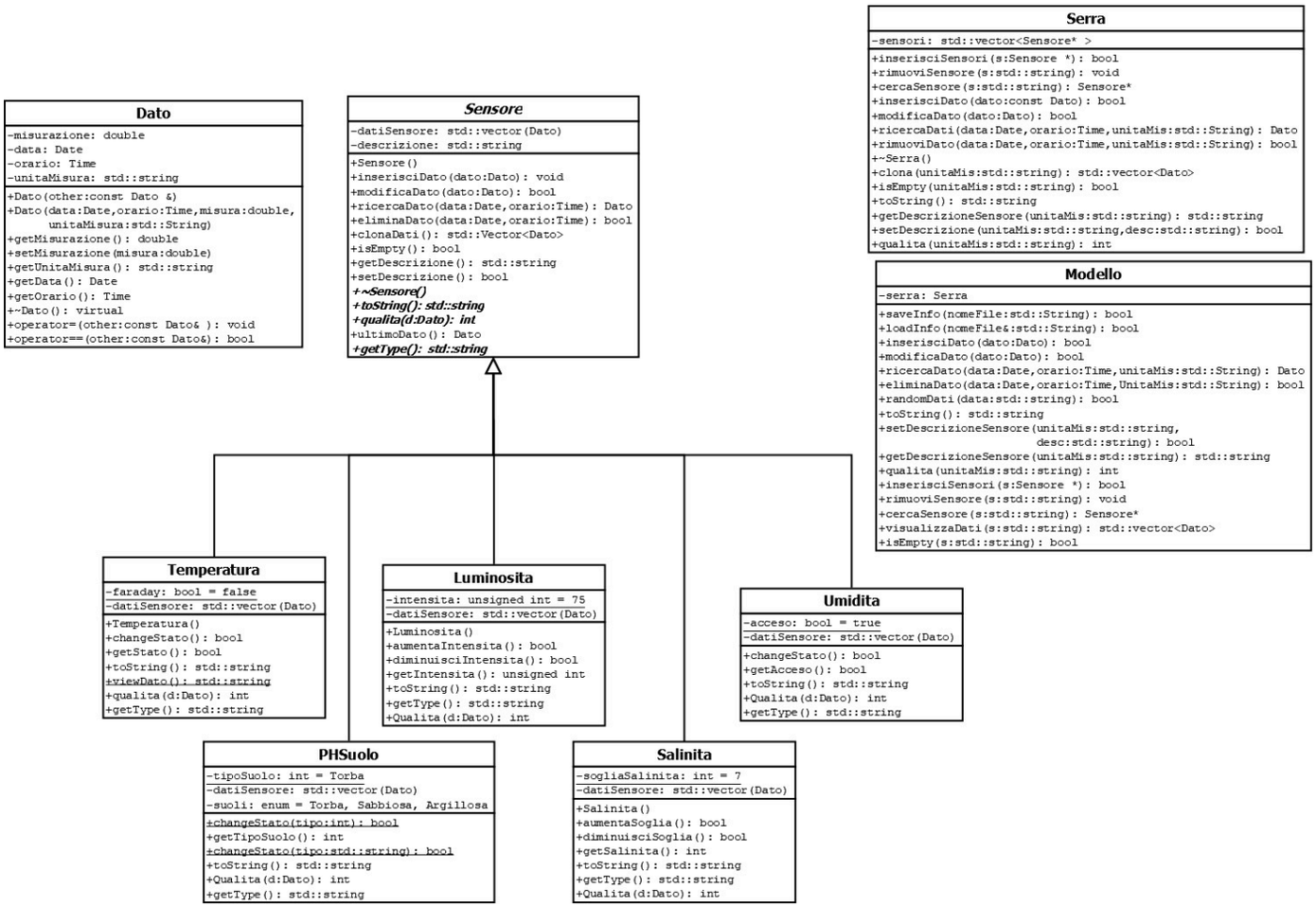
Alessandro Benin 2042356
Filippo Rizzolo 2042377

Nessuna modifica

Introduzione

Questo progetto è il frutto della nostra consapevolezza sull'efficacia dell'utilizzo di sensori specifici all'interno di un ambiente di coltivazione come una serra, al fine di migliorare la produttività e la qualità di vari prodotti alimentari.
Questi aiuti sono garantiti dai diversi sensori pensati per misurare tutti i valori che possono influenzare la nascita e crescita delle diverse piante, diminuendo o aumentandone il tempo di produzione. Inoltre, il controllo più accurato delle condizioni delle piante tramite l'uso di sensori potrebbe ridurre la necessità di utilizzare prodotti chimici per mantenere la salute delle piante, contribuendo così a prevenire la contaminazione del terreno da pesticidi e concimi.
In particolare, abbiamo sviluppato sensori per monitorare fattori come la luminosità all'interno della serra, oltre a raccogliere dati dal terreno, come il pH del suolo, la temperatura, la salinità e l'umidità."

Modello



Il modello comprende la gestione dei diversi sensori e la loro visualizzazione. Una parte riguarda la definizione delle classi utilizzate per descrivere i sensori e alcune classi per convertirle nel formato JSON, consentendo così il salvataggio dei dati, mentre l'altra parte si occupa della visualizzazione dei sensori. Nella base della nostra gerarchia c'è la classe Sensore, che include un vettore di tipo Dato in cui vengono memorizzati tutti i diversi valori relativi ad esso.
Il design utilizzato si ispira al modello MVC (Model-View-Controller), al fine di mantenere separate le diverse parti dell'applicazione. Nel modello sono contenuti i vari metodi necessari per accedere ai dati, mentre nella vista viene gestita l'interfaccia tra l'utente e il sottostante sistema. Infine, nel controller, che riceve i comandi dell'utente tramite la vista, vengono attivati i metodi appropriati nel modello in risposta alle azioni dell'utente.

Polimorfismo

Il polimorfismo viene implementato all'interno del progetto attraverso il design pattern MVC, in particolare per i seguenti metodi:

- Metodo "int Qualità(Dato d)": questo metodo è utilizzato per determinare la qualità dell'ultimo dato inserito. Restituisce un valore appropriato in base ai diversi controlli eseguiti su ogni tipologia di sensore. I valori di ritorno sono i seguenti:
  A. 0 indica che il dato è BASSO (colore blu)
  B. 1 indica che il dato è BUONO (colore verde)
  C. 2 indica che il dato è ALTO (colore rosso)

  Questo metodo sfrutta il polimorfismo per adattarsi dinamicamente ai diversi tipi di sensori e eseguire i controlli appropriati sulla qualità dei dati in base alle specifiche caratteristiche di ciascun sensore.

- Metodo "string toString()": questo metodo è utilizzato per stampare le informazioni presenti e, inoltre, per inserire i dati nel file Dati.json dopo aver premuto il tasto "salva" nella GUI. Anche questo metodo sfrutta il polimorfismo per adattarsi alla specifica implementazione di ciascun tipo di sensore e restituire una rappresentazione sotto forma di stringa delle informazioni relative al sensore.

In entrambi i casi, il polimorfismo consente di trattare oggetti di classi diverse in modo uniforme, poiché i metodi vengono implementati in modo diverso da ciascuna sottoclasse, garantendo una maggiore flessibilità e modularità del codice.

Persistenza dei Dati

I dati vengono salvati nel file "Dati.json" utilizzando il formato JSON. Ogni volta che si effettua il salvataggio, il file viene sovrascritto per evitare la creazione di copie aggiuntive. È anche possibile caricare i dati da questo file tramite un'apposita funzione nella GUI. Quando si carica il file, i dati esistenti vengono sostituiti con quelli presenti nel file JSON, eliminando eventuali dati precedenti.

Introduzione alla GUI

La nostra interfaccia *(FIGURA 2)* presenta dei bottoni che consentono all'utente di inserire i diversi sensori a sua scelta, da uno solo fino a tutti e cinque. Inoltre, offre la possibilità di caricare o salvare i dati e di generare dati casuali all'interno di un range specificato durante la fase di progettazione. Infine, include una funzione di ricerca del sensore tramite la sua unità di misura nell'apposita sezione.
Nella visualizzazione dei grafici *(FIGURA 3)*, nella legenda in fondo alla pagina, è presente un quadrato che, attraverso il colore e la descrizione, indica la qualità dell'ultimo dato inserito:

- Il colore blu indica un valore BASSO
- Il colore verde indica un valore BUONO
- Il colore rosso indica un valore ALTO

FIGURA 2

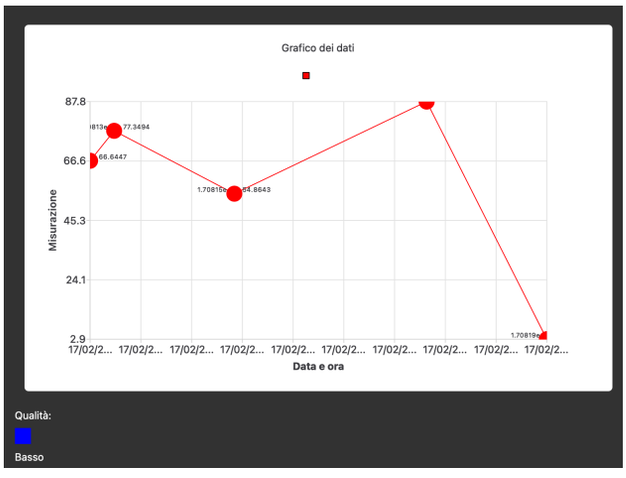


FIGURA 3

Funzionalità implementate

- Creazione dei diversi sensori
- Rimozione dei diversi sensori
- Visualizzazione dei diversi sensori
- Per ogni sensore:
    A. Inserimento dei dati
    B. Visualizzazione del grafico
    C. Modifica della descrizione del sensore
    D. Visualizzazione dei dati:
        1. Modifica del singolo dato
        2. Rimozione del singolo dato
- Caricamento e salvataggio dei dati in un file JSON
- Generazione di dati casuali
- Ricerca di un sensore tramite unità di misura

Rendicontazione ore

Attività	Ore Previste	Ore Effettive
Progettazione	10	10
Sviluppo del codice del modello	20	24
Studio del framework QT	5	8
Sviluppo del codice della GUI	10	15
Test e debug	5	10
Stesura della relazione	5	5
Totale	55	72

Suddivisione Attività

Attività	Assegnazione
Progettazione	Alessandro e Filippo
Sviluppo del codice del modello	Alessandro
Sviluppo del codice della GUI	Filippo
Sviluppo del grafico	Filippo
Sviluppo del Controller	Alessandro
Salvataggio su file JSON	Alessandro e Filippo
Caricamento da file JSON	Alessandro e Filippo