

WINESENSE



Progetto di Programmazione ad Oggetti
A.a. 2023/2024

Moretto Nicholas - 1226318



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

1. Abstract

WineSense è un programma pensato per la gestione delle Autoclavi contenenti vino.

Tramite l'utilizzo dei sensori si occupa di raccogliere dati riguardanti Temperatura, Pressione, Umidità, Ph e livello dell'autoclave stessa, e rappresentare questi ultimi sottoforma di lista di dati, e sottoforma di grafico.

In un'ottica di commercializzazione e di sviluppo futuro, questo programma è pensato per gestire le varie fasi di fermentazione del vino, e quindi per tenere sotto controllo le soglie dei sensori definite dall'utente correggendo e mettendo in funzione sistemi per mantenere ad esempio il range di temperatura scelto, piuttosto che il range di pressione.

2. Descrizione del programma

WineSense dà la possibilità all'utente di gestire un'autoclave fisica partendo dalla creazione della stessa, indicando nome e capacità. Ogni autoclave è univoca: questo è reso possibile dal salvataggio dell'id ogni qualvolta viene creata un'autoclave. Anche se non salvata, l'id verrà incrementato lo stesso per evitare qualsiasi problema di ambiguità che si creerebbe lavorando su più autoclavi non salvate e magari salvate in un secondo momento.

Per ogni autoclave si ha la possibilità di inserire fino a cinque sensori:

- Temperatura
- Pressione
- Umidità
- PH
- Livello (livello di liquido contenuto nell'autoclave)

Per ogni sensore l'utente deve indicare la soglia minima e massima che l'autoclave dovrebbe mantenere. L'utente ha la possibilità di modificare le soglie inserite in precedenza, di eliminare il sensore, di generare dei dati random per simulare la raccolta dati del sensore stesso, e di visualizzare i dati generati sottoforma di grafico.

Inoltre il grafico può essere visualizzato sottoforma di **LineChart** o di **BarChart**. Nella generazione di dati, dato che possono esserci delle anomalie (ovvero dati registrati che non sono inclusi nel range di soglia), tramite il metodo *controllaAnomalia* vengono evidenziati i valori fuori dal range.

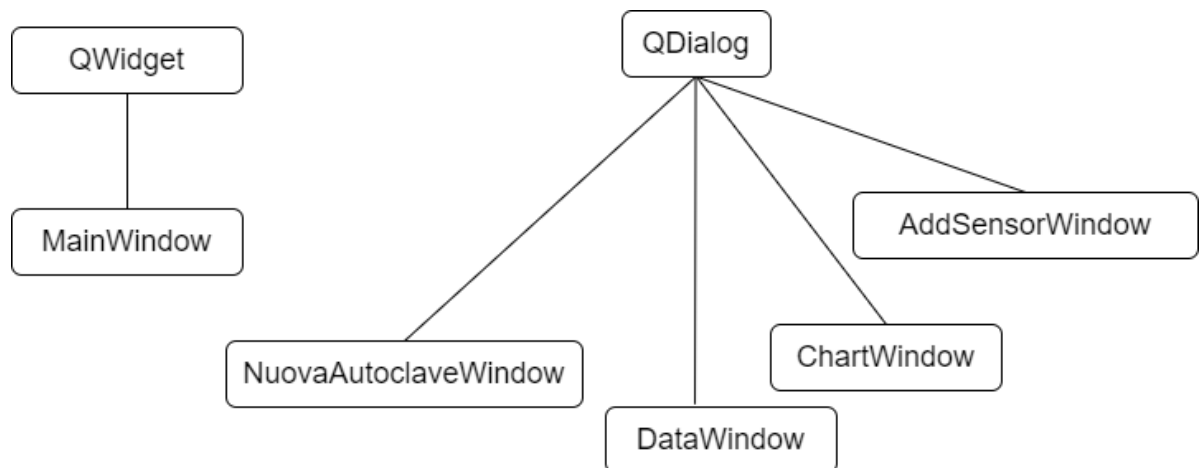
Infine, l'utente può lavorare contemporaneamente su più autoclavi, oltre che salvarle o caricarle.

3. Funzionalità

- Elaborazione dei dati e visualizzazione tramite grafici: sono offerti i tipi LineChart e BarChart. Questi due grafici rappresentano gli stessi dati, ma sono stati implementati entrambi per dare all'utente la possibilità di scegliere in base alla preferenza di visualizzazione.
- Salvataggio dei dati in formato .csv: l'utente può salvare o meno l'autoclave sulla quale lavora. In caso di salvataggio, viene generato un file .csv con i dati dell'autoclave e dei sensori. È stato scelto appositamente di dare la possibilità all'utente di salvare le autoclavi in qualsiasi stato (ovvero incomplete, senza alcun sensore, con i sensori ma senza dati raccolti, con i dati raccolti) in modo che sia possibile effettuare sempre il salvataggio. Questo per dar libertà all'utente di poter fermare il lavoro e riprenderlo in seguito nel medesimo stato.
- Salvataggio automatico di *id_autoclave*. Il salvataggio dell'id dell'autoclave menzionato in precedenza viene effettuato automaticamente alla creazione di una nuova autoclave. In caso di aggiunta di una nuova autoclave, viene prelevato il dato salvato, viene incrementato e poi sovrascritto.

4. Architettura del progetto

4.1. Gerarchia Vista

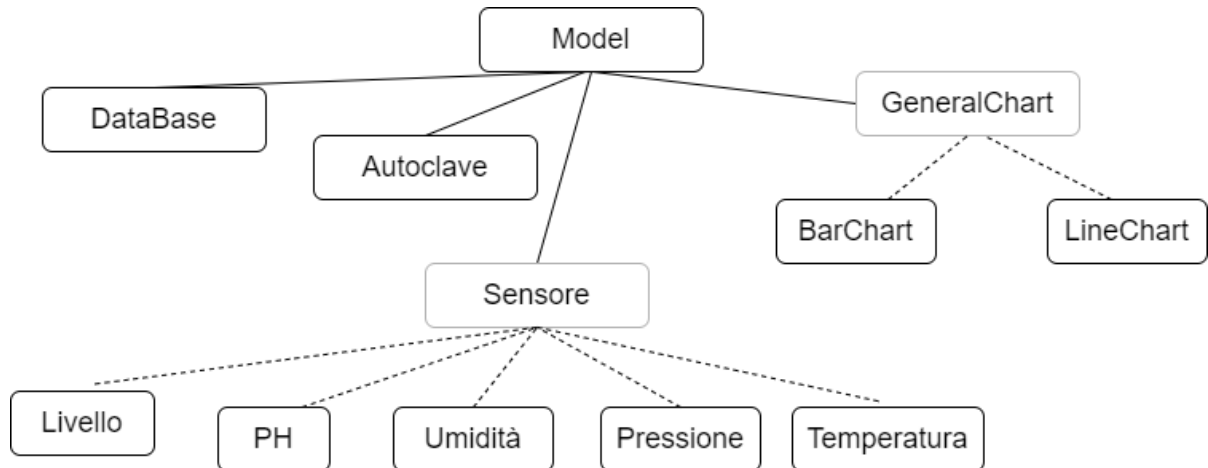


La classe **MainWindow** deriva dalla classe **QWidget** in quanto permette l'utilizzo dei widget per costruire le varie finestre. La **MainWindow** è la finestra principale del programma che include: la schermata principale (dove l'utente può creare una nuova autoclave o caricarne una) e la *manageView*, che mostra i dettagli dell'autoclave e dei relativi sensori, se presenti.

Le classi **NuovaAutoclaveWindow**, **DataWindow**, **ChartWindow** e **AddSensorWindow** derivano da **QDialog** in quanto sono finestre di dialogo che appaiono dopo aver premuto un tasto per svolgere un'operazione.

Rispettivamente mostrano le finestre di creazione di una nuova autoclave, di visualizzazione dei dati generati, di visualizzazione del grafico, e di aggiunta del sensore.

4.2. Gerarchia Modello



Il Modello è rappresentato dalla classe **Model**. Le classi Database, Autoclave, Sensori e GeneralChart non sono derivate da model, ma si appoggiano direttamente a quest'ultimo.

La classe **Database** gestisce il salvataggio e il caricamento delle Autoclavi e dei sensori.

La classe **Autoclave** gestisce le funzionalità delle Autoclavi.

La classe **Sensore** è una classe base astratta, che gestisce le funzionalità dei sensori per ogni autoclave. *Livello*, *Ph*, *Umidità*, *Pressione* e *Temperatura* sono le classi derivate da Sensore.

GeneralChart è una classe base astratta, la quale ha come classi figlie *BarChart* e *LineChart*, e si occupa della visualizzazione dei grafici.

4.3. Polimorfismo

Il polimorfismo è stato utilizzato nelle classi Sensore e GeneralChart. Vista la natura astratta di queste classi, è stato fatto l'*override* di vari metodi, tra cui **clone**, **getDati**, **setDati**, **controllaAnomalia**, **getLimiti** e **getUnitàMisura**.

4.4. Note

Per i test e il debug in fase di sviluppo, ho utilizzato principalmente QDebug di Qt, che mi ha permesso spesso di intercettare gli errori a run-time, e, di conseguenza, di correggerli.

5. Conclusioni

5.1. Ore di lavoro

Attività	Ore impiegate
Analisi delle specifiche	3
Progettazione della GUI	6
Studio Framework Qt	7
Progettazione e implementazione della struttura dati	10
Progettazione e implementazione del modello e dei grafici	12
Progettazione e implementazione della parte grafica	15
Test e debugging	4
Stesura della relazione	3
Totale ore	60

5.2. Ambiente di sviluppo

Per lo sviluppo è stato utilizzato Visual Studio Code. Per la compilazione è stato utilizzato QMake e Make. Lo sviluppo è avvenuto utilizzando Qt6(6.2.4) e qmake 3.1.

5.3. Compilazione ed esecuzione

La compilazione avviene con **QMake** mediante il file .pro.

Utilizzando i comandi *qmake* e *make* viene creato l'eseguibile con il quale si può avviare il programma.

5.4. Note

Questa attività è stata molto interessante e formativa. Mi ha permesso per la prima volta di sviluppare un'applicazione completa, a differenza di altri lavori e progettini passati. Soprattutto ho trovato stimolante lo sviluppo della parte grafica. Inizialmente ho riscontrato qualche difficoltà in questo, in particolare a causa di alcune librerie di Qt a me sconosciute. Ma dopo un approfondito studio, sono riuscito perfettamente nell'intento.