

Programmazione ad Oggetti

Relazione Progetto

“Prodevice”

- 1 - Introduzione
- 2 - Modalità consegna
- 3 - Gerarchia e classi
 - 3.1 - Gerarchia
 - 3.2 - Campi delle classi
 - 3.3 - Prezzi
 - 3.4 - Uguaglianza
 - 3.5 - Metodi polimorfi
 - 3.6 - Metodi polimorfi
- 4 - Contenitore
- 5 - Graphical User Interface
 - 5.1 - Homepage
 - 5.2 - Aggiungere prodotti
 - 5.3 - Ricerca di un dispositivo
 - 5.4 - Catalogo e modifica
- 6 - Model Control View
- 7 - Estensibilità del codice
- 8 - Conteggio ore

1 - Introduzione

Il progetto "ProDevice" consiste in un gestionale di dispositivi per un rivenditore. Permette di visualizzare un catalogo ed aggiungere, modificare o rimuovere i prodotti.

Il programma è stato sviluppato su sistema operativo Windows 10 con QtCreator 4.8.1 e successivamente testato su Linux Ubuntu 18.04.1 LTS tramite macchina virtuale Oracle VirtualBox.

2 - Modalità consegna

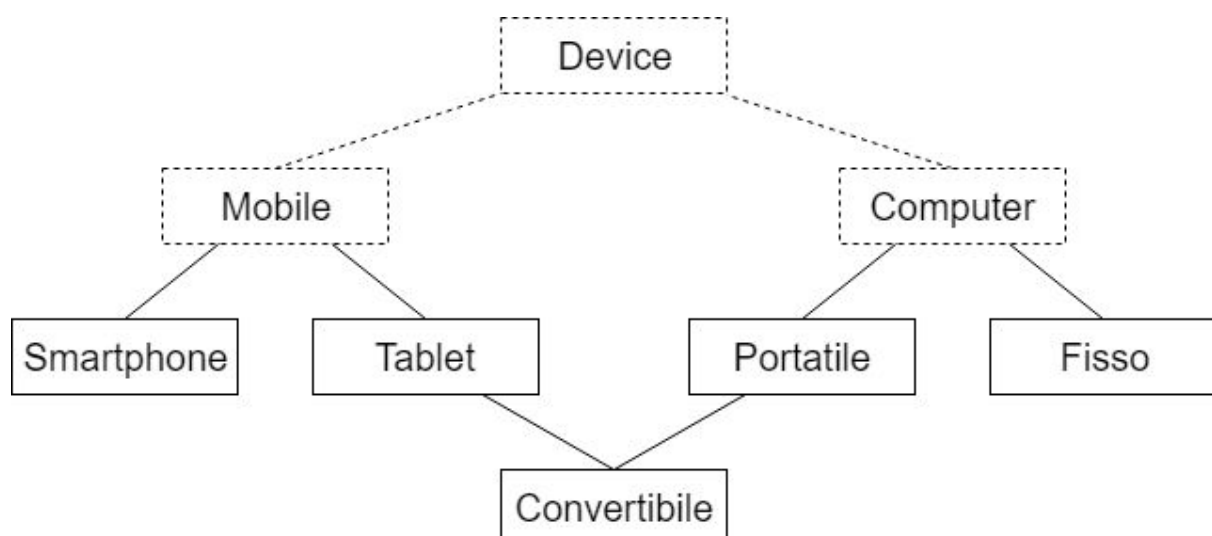
Come indicato il progetto è stato consegnato col comando **consegna progetto-pao-2019**.

Sono presenti tutti i file necessari incluso il file **ProDevice.pro** generato automaticamente da Qt all'avvio del progetto.: poiché la compilazione del progetto necessita di tale file per qmake, diverso da quello ottenibile tramite l'invocazione di **qmake -project**, che permette la generazione automatica tramite qmake del Makefile.

3 - Gerarchia e classi

3.1 - Gerarchia

La gerarchia definisce la suddivisione dei diversi tipi di dispositivi. **Device** è la base astratta dalla quale derivano **Mobile** e **Computer** anch'esse astratte. Al terzo livello di profondità troviamo le classi concrete: **Smartphone** e **Tablet** derivate da Mobile, **Portatile** e **Fisso** derivate da Computer. Infine la gerarchia forma un diamante con la sottoclasse **Convertibile** che deriva da Tablet e Portatile poiché rappresenta i portatili che possono essere utilizzati come tablet (ruotando lo schermo a 360° oppure staccandolo dalla tastiera).



3.2 - Campi delle classi

La classe base astratta **Device** contiene le informazioni comuni ad ogni tipologia di dispositivo:

- **Modello**: nome del dispositivo;
- **Produttore**: marca del dispositivo;
- Dimensione dello **schermo**;
- Informazioni del **processore**;
- Dimensione della **memoria ram**;
- Dimensione della **memoria interna**;
- **Prezzo** base* del prodotto;
- Percorso dell'immagine.

Le classi di secondo livello contengono le informazioni generiche relative ai dispositivi rappresentati dalle relative sottoclassi.

Classe **Mobile**:

- Se la memoria è espandibile con una scheda SD;
- Se il dispositivo è dotato di un ingresso Jack per periferiche audio;
- Se il dispositivo offre la funzionalità di riconoscimento facciale;
- Mpx della fotocamera frontale;
- Mpx della fotocamera posteriore.

Classe **Computer**:

- Se il computer è dotato di touchscreen;
- Se vi è un ingresso lettore CD;
- Numero di porte USB.

Le classe concrete contengono dati propri, in aggiunta a quelli elencati qui sopra, per fornire le seguenti informazioni aggiuntive:

Classe **Smartphone**:

- Se lo smartphone può possibile gestire una SIM secondaria.

Classe **Fisso**:

- Se il computer fisso ha il WiFi;
- Se il computer fisso ha il Bluetooth.

Classe **Tablet**:

- Se nel Tablet si può inserire una SIM (altrimenti è solo WiFi).

Classe **Portatile**:

- Se il portatile ha un ingresso per cavo Ethernet;
- Se ha la webcam;
- Mpx della webcam;
- Se la tastiera è retroilluminata.

Classe **Convertibile**:

- Se è dotato di apposita penna compatibile con lo schermo;
- Se schermo e tastiera si staccano.

Da notare che **Convertibile** derivando sia da *Mobile* sia da *Portatile* comprende informazioni ridondanti (Mpx della fotocamera frontale, Mpx della fotocamera posteriore e Mpx della Webcam): per risolvere tale situazione nel costruttore (come nella GUI) non viene richiesto nessun valore relativo alla Webcam, quest'ultima viene considerata la fotocamera frontale, quindi viene richiesto un solo valore che verrà assegnato ad entrambi i campi ereditati.

3.3 - Prezzi

Ogni dispositivo (nella base astratta *Device*) ha associato un **prezzo base**. Il gestionale di questo progetto identifica i dispositivi in base a modello, produttore e alcune caratteristiche che ne provocando un aumento del prezzo. Quindi ogni prodotto possiede un prezzo base e un **prezzo totale** che si ottiene sommando gli extra al prezzo base.

Nella gerarchia vi sono delle variabili statiche che indicano dell'aumento di prezzo:

- 20€ - *Mobile* con scheda SD ;
- 100€ - *Computer* Touchscreen;
- 30€ - *Smartphone* dual SIM;
- 30€ - Tablet con SIM;
- 70€ - Portatile con tastiera retroilluminata;

Per esempio uno smartphone Huawei P20 potrebbe esserci in due varianti, singola SIM e dual SIM. Il prezzo base è 280€ per entrambi: il prezzo totale della variante singola SIM resterà 280€ mentre il prezzo totale per la variante dual SIM risulterà essere 310€ (ovvero 280€ base + 30€ extra dual SIM).

3.4 - Uguaglianza

Nella gerarchia, per ogni classe, è stato ridefinito l'operatore di uguaglianza *operator==*. In riferimento a quanto spiegato riguardo i prezzi: due dispositivi si considerano uguali se hanno stesso modello, stesso produttore e stessi extra. Il resto delle informazioni non viene confrontato.

3.5 - Metodi polimorfi

I metodi virtuali della gerarchia che sfruttano il polimorfismo:

- *getTipo()* ritorna una stringa col tipo del dispositivo;
- *stampaSpecifiche()* fornisce una stringa contenente tutte le informazioni relative ad un dispositivo;
- *calcolaPrezzo()* considera tutti gli extra di un prodotto per fornire il prezzo totale;
- *operator==* operatore di uguaglianza per confrontare due dispositivi;
- il *distuttore* virtuale.

4 - Contenitore

Come richiesto nelle specifiche è stato definito un **Container** (*qonteiner.h*). Tale template rappresenta un contenitore di dati con le funzionalità richieste:

- Inserimento all’inizio *insertFront()* e alla fine *insertBack()*;
- Ricerca per oggetto *cerca(const T&)*;
- Eliminazione di un elemento *elimina(const T&)*;
- Iteratori costanti e non.

La lista Container sfrutta SmartPointer che puntano a Nodi e servono ad ottimizzare l’uso della memoria e prevengono la cancellazione involontaria di elementi.

È stata scelta una doppia lista dato che in un gestionale di un negozio è alta la probabilità di rimuovere spesso oggetti in mezzo alla lista.

5 - Graphical User interface

Il progetto “ProDevice” è suddiviso nelle seguenti schermate:

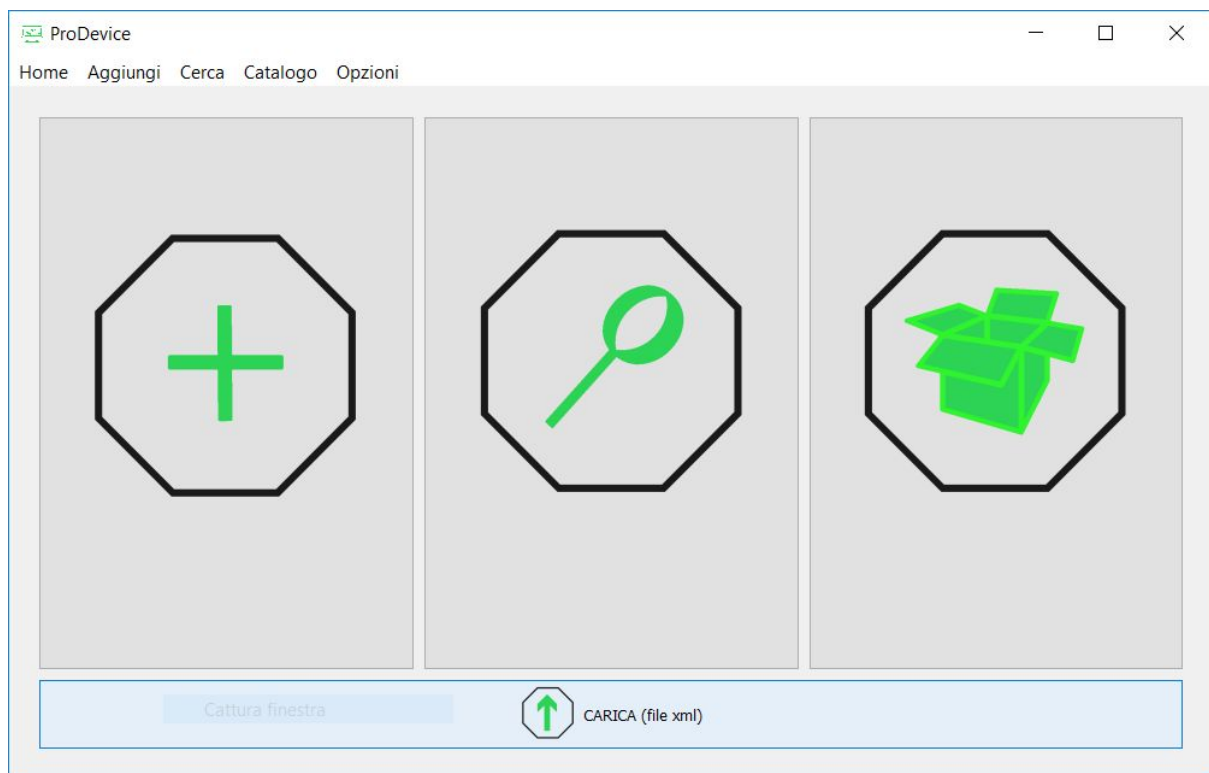
- **Home**: per una comoda navigazione all’interno del programma;
- **Aggiungi**: per inserire nuovi prodotti all’interno di un catalogo;
- **Cerca**: per cercare un prodotto e visualizzarne il prezzo;
- **Catalogo**: per visualizzare l’intero catalogo e modificarne o rimuoverne i prodotti.

Si può navigare tra le schermate tramite il menù in alto, oppure dalla Home.

5.1 - Homepage

All'**avvio del programma** viene chiesto per prima cosa di selezionare un file in formato .xml da leggere, ovvero di caricare un catalogo. Senza tale file non è possibile visualizzare, cercare, aggiungere, modificare o eliminare prodotti per cui se non viene scelto un xml si apre un popup che avvisa l'utente.

Viene quindi mostrata la pagina Home, dalla quale è possibile spostarsi su ogni altra schermata. Il pulsante in basso permette di **caricare** un xml dall'interno del programma, per cambiare il catalogo o caricarlo nuovamente se ci sono stati errore Inoltre questa operazione può essere effettuata da qualunque pagina tramite le opzioni nel menù.



In seguito il file xml viene **automaticamente salvato** in seguito a ogni modifica effettuata sui prodotti, mantenendo sempre coerenza tra la visualizzazione grafica e il suo contenuto.

5.2 - Aggiungere prodotti

Dalla pagina Aggiungi è possibile inserire nuovi prodotti. Per l'aggiunta di un dispositivo vi sono un layout per ogni classe della gerarchia. Subito sono disponibili gli elementi grafici per impostare i campi della classe base astratta *Device* e alla selezione del tipo che si vuole inserire appaiono tutti i campi necessari.

ProDevice

Home Aggiungi Cerca Catalogo Opzioni

Portatile (Computer) Prezzo: Inserisci prezzo base

Modello: Inserisci il modello Produttore: Inserisci il produttore

Schermo: Inserisci dimensione schermo Processore: Inserisci processore

Ram: 2 GB Mem: 8 GB

☐ TouchScreen ☐ Lettore CD Porte USB: Inserisci numero porte usb

☐ Porta ethernet ☐ Tastiera retroilluminata

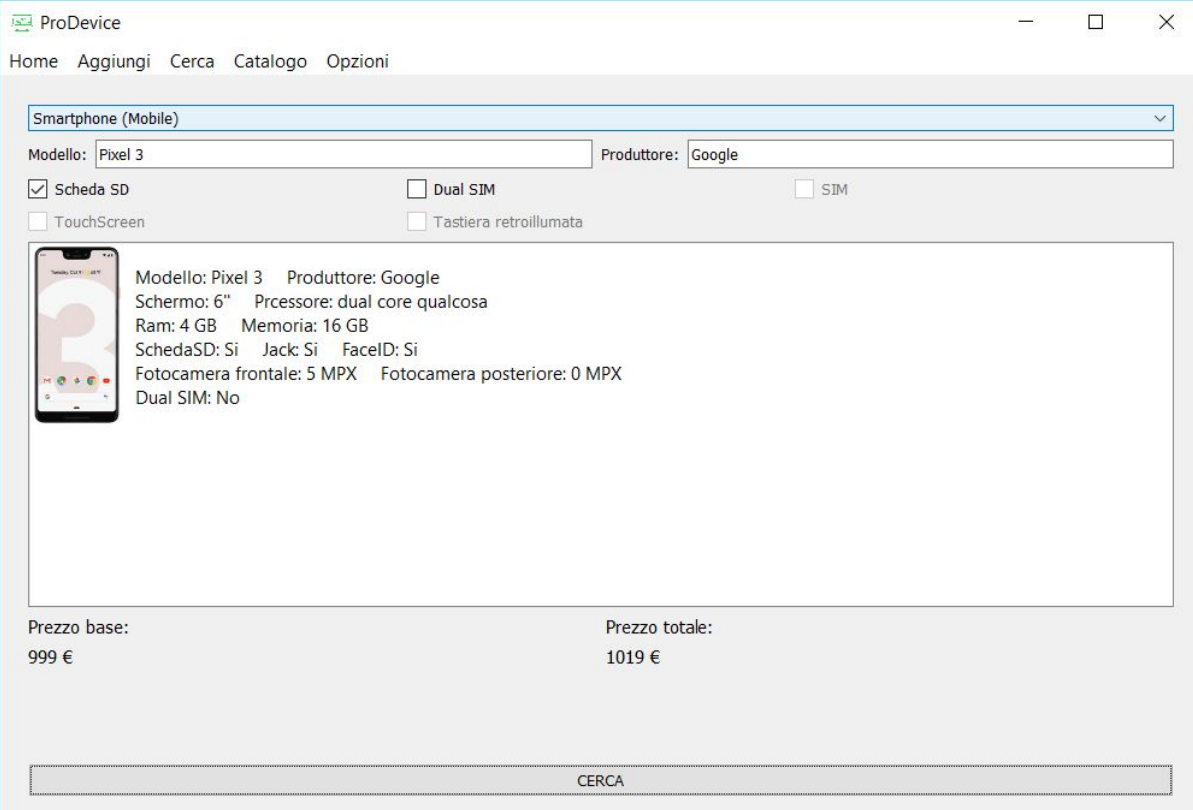
☐ Webcam Pixel Webcam: Inserisci pixel webcam

Inserisci immagine AGGIUNGI

L'utente viene avvisato dell'errore se preme il pulsante di conferma senza selezionare il tipo di dispositivo o senza aver compilato dei dati. Inoltre viene anche segnalato se si inserisce un prodotto uguale ad un'altro già presente. Se l'aggiunta va a buon fine l'xml viene aggiornato e il nuovo prodotto è visualizzabile nella pagina Catalogo.

5.3 - Ricerca di un dispositivo

La Ricerca presenta pochi elementi che vengono abilitati o resi inutilizzabili in base al tipo di dispositivo selezionato per la ricerca. Per la definizione dei criteri di uguaglianza [sezione 3.4] può esserci solamente un risultato.



The screenshot shows the ProDevice application window. At the top, there is a navigation bar with 'Home', 'Aggiungi', 'Cerca', 'Catalogo', and 'Opzioni'. Below this is a dropdown menu set to 'Smartphone (Mobile)'. The search criteria are: 'Modello: Pixel 3' and 'Produttore: Google'. There are several checkboxes for features: 'Scheda SD' (checked), 'Dual SIM' (unchecked), 'SIM' (unchecked), 'TouchScreen' (unchecked), and 'Tastiera retroilluminata' (unchecked). The search results display a small image of a Pixel 3 phone next to its specifications: 'Modello: Pixel 3', 'Produttore: Google', 'Schermo: 6"', 'Processore: dual core qualcosa', 'Ram: 4 GB', 'Memoria: 16 GB', 'SchedaSD: Si', 'Jack: Si', 'FaceID: Si', 'Fotocamera frontale: 5 MPX', 'Fotocamera posteriore: 0 MPX', and 'Dual SIM: No'. At the bottom, the 'Prezzo base:' is listed as '999 €' and the 'Prezzo totale:' is listed as '1019 €'. A 'CERCA' button is located at the very bottom of the window.

Criterio	Valore
Modello	Pixel 3
Produttore	Google
Scheda SD	<input checked="" type="checkbox"/>
Dual SIM	<input type="checkbox"/>
SIM	<input type="checkbox"/>
TouchScreen	<input type="checkbox"/>
Tastiera retroilluminata	<input type="checkbox"/>

Smartphone (Mobile)

Modello: Pixel 3 Produttore: Google

Schermo: 6" Processore: dual core qualcosa

Ram: 4 GB Memoria: 16 GB

SchedaSD: Si Jack: Si FaceID: Si

Fotocamera frontale: 5 MPX Fotocamera posteriore: 0 MPX

Dual SIM: No

Prezzo base: 999 € Prezzo totale: 1019 €

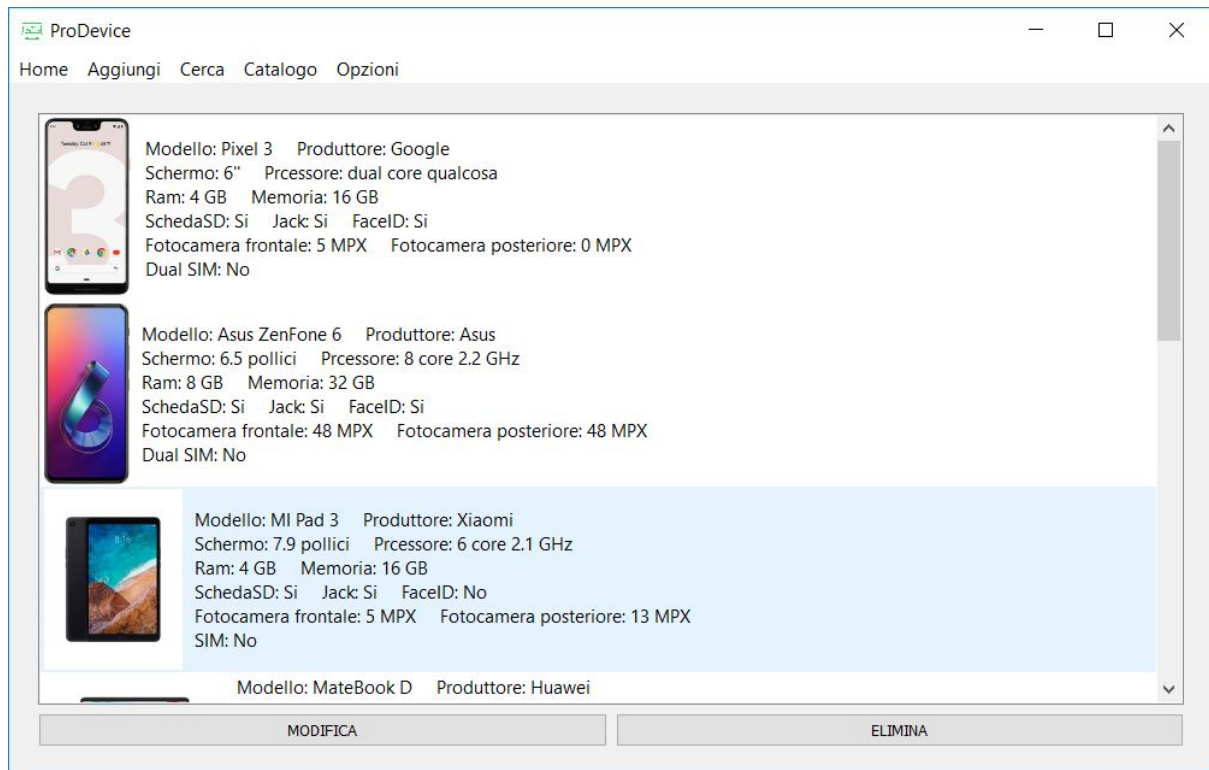
CERCA

Insieme al dispositivo risultante ne vengono visualizzati il prezzo base e il prezzo totale, calcolati come spiegato precedentemente [sezione 3.3].

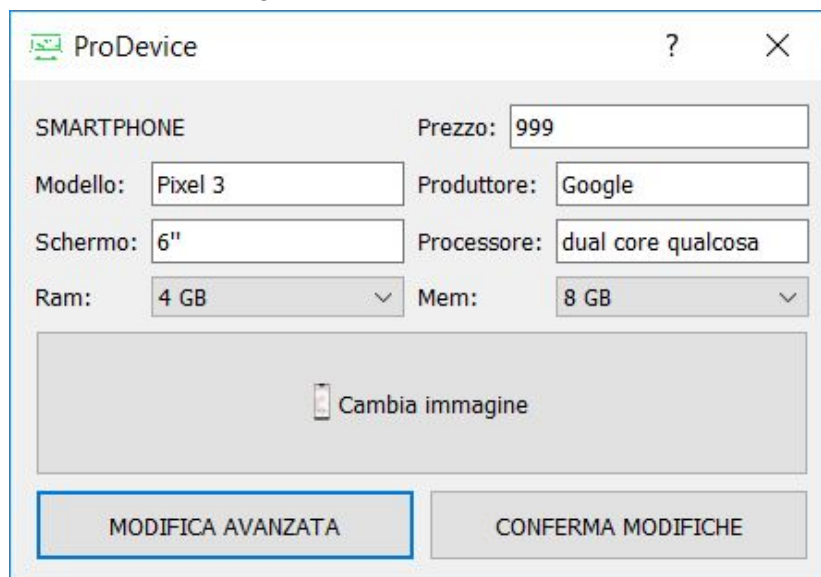
5.4 - Catalogo e modifica

Nella pagina Catalogo vengono visualizzati tutti i prodotti del catalogo (file .xml) con le relative informazioni e immagini.

I pulsanti in basso permettono di modificare o eliminare un singolo dispositivo quando è selezionato. L'utente viene avvisato se prova ad effettuare tali operazioni senza prima selezionare il prodotto.



Scegliendo di modificare un dispositivo appare una finestra aggiuntiva, con un layout simile a quello della pagina Aggiungi, già completo delle informazioni del prodotto che possono essere cambiate. Si vedono inizialmente solo le informazioni principali, premendo "modifica avanzata" è possibile vedere tutti gli elementi.



In seguito ad eliminazione o modifica il file xml viene automaticamente aggiornato e la visualizzazione del catalogo aggiornata.

6 - Model Control View

Lo schema di progettazione Model Control View, solitamente utilizzato per lo sviluppo di interfacce utente, separa la parte logica di presentazione dalla parte logica applicativa.

Suddivide in tre parti il funzionamento:

- Model (modello.h/modello.cpp) comprende i metodi per accedere ai dati;
- View (i layout): visualizza i dati raccolti nel modello e predispone l'interazione con l'utente;
- Controller (controller.h/controller.cpp) gestisce i comandi ricevuti dall'utente e comunica con modello e layout modificandone lo stato.

Nel modello le funzioni principali utilizzate sono salvataggio e caricamento che agiscono sul file .xml modificando così il catalogo. Queste funzioni vengono sempre chiamate in caso di aggiunta, modifica, rimozione di un prodotto. Tramite il controller vengono gestite le azioni e il trasferimento delle informazioni all'interno del programma.

7 - Estensibilità del codice

Sia le classi della gerarchia sia gli elementi grafici sono separati tra interfacce e implementazioni. Inoltre la gerarchia è suddivisa in più file: un .h e .cpp per ogni classe. Per gli elementi grafici sono anch'essi separati e i più complessi - layout aggiungi e modifica - che richiedono di essere mostrati o nascosti in parte rispetto all'azione dell'utente, oltre a essere suddivisi, sono inseriti in cartelle.

8 - Conteggio ore

Analisi specifiche del progetto	1 ora
Pianificazione gerarchia e container	2 ore
Scrittura gerarchia e container	10 ore
Apprendimento IDE Qt	15 ore
Scrittura GUI	25 ore
Test e debugging	6 ore
Stesura relazione	3 ore
TOTALE	62 ore