

# **Relazione progetto di Programmazione e modellazione ad Oggetti (PMO)**

**ANTONIO CARDELLINI  
NUMERO MATRICOLA : 293475**

---

## Specifica

---

Una applicazione utile alla gestione della dispensa, con il fine di ridurre gli sprechi di cibo. Salva in memoria i prodotti aggiunti alla dispensa , calcola quando dovranno essere ricomprati , e mostra la data di scadenza. Inoltre mostra in altre pagine i prodotti che stanno per terminare e i prodotti che in teoria dovrebbero essere già terminati. Ha una sezione dedicata alle ricette. le varie ricette culinarie che vengono divise in antipasti, primi, secondi ,dolci. Mostrate cliccando in dei pulsanti dedicati.

## Studio del problema

---

### Punti critici:

- Creare una UI
- Passaggi di dati tra la partial class xaml e la partial class in C#
- Gestire gli alimenti inseriti nella dispensa
- Fare in modo che i dati non vadano persi
- Riempire le 3 diverse sezioni dell'applicazione
- Mettere a disposizione un ricettario

### Soluzione ai punti critici:

- Per Creare la interfaccia utente per la mia applicazione ho imparato e utilizzato Xamarins. È uno strumento messo a disposizione dalla Piattaforma .Net che permette di creare Applicazioni. Utilizza come linguaggio per la logica del programma c# e come linguaggio per la parte grafica xaml.
- Inizialmente ho riscontrato problemi con il passaggio di dati. Servendomi della documentazione messa disposizione da Microsoft . Ho risolto il mio problema impiegando il Data Binding. come dice Il sito di documentazione di Xamarin. Form :” Il data binding consente di automatizzare questo processo ed elimina la necessità di usare gestori degli eventi. I data binding possono essere implementati nel codice o in XAML, ma sono molto più comuni in XAML dove contribuiscono alla riduzione delle dimensioni del file code-behind. Sostituendo il codice procedurale nei gestori degli eventi con codice dichiarativo o markup, l'applicazione diventa più semplice e chiara.”
- Per gestire gli alimenti che vengono inseriti dall'utente , ho utilizzato tre tipi diversi di Design pattern. Ho fatto in modo che gli alimenti vengono inizializzati dal utente il quale deve inserire anche varie caratteristiche relative al'prodotto.
- Ho utilizzato un database interno all'applicazione per poter salvare i dati e renderli usufruibili nel tempo. Questo database è amministrato direttamente da dei pulsanti presenti nella UI(Add new item, Delete item ), e indirettamente da delle regole scritte in codice , che eliminano il prodotto dal'database dopo un certo lasso di tempo, variabile a seconda dell'prodotto.
- Per Riempire le tre diverse sezioni dell'applicazione ho deciso di utilizzare un solo database . Poi i dati raccolti vengono smistati nelle opportune code . Questa soluzione mi è sembrata la migliore , poiché necessita meno spazio di archiviazione rispetto tre database separati ed è più semplice la gestione.
- Per Disporre di un ricettario nella mia applicazione ho pensato di creare un collegamento alla pagina web di Giallo Zafferano. Ho creato diversi collegamenti per rendere più immediata la

navigazione nel blog, infatti ho suddiviso il ricettario in antipasti, primi, secondi e Dolci. Inoltre cliccando sul logo di Giallo zafferano presente nella mia applicazione , il collegamento si apre nella main page del blog , così da rendere disponibile in modo immediato la ricerca di ricette specifiche.

# Scelte architettoniche

Questa applicazione Utilizza un DataBase locale, sul quale vengono aggiunti gli Item. Ogni item per essere creato utilizza il **Builder Pattern**. Ho utilizzato il pattern creazionale per poter effettuare più semplicemente manutenzione del programma , e per agevolare anche futuri Upgrade, per esempio aggiungere nuovi tipi di Item oltre ItemSemplice.

Ogni Item mette a disposizione una data di *Rebuy* , cioè una data ipotetica in cui l'oggetto andrebbe ricomprato. Questa data è calcolata tenendo conto del numero di oggetti dello stesso tipo (esempio 3 Arance) moltiplicato per il numero di giorni che ogni singolo oggetto approssimativamente sta in dispensa (3 arance, ogni arancia dura 1 giorno , quindi la data di *Rebuy* sarà data di oggi sommata di 3 giorni).

Gli item raccolti nel Database vengono poi Trasferiti su una lista per gestirli più agevolmente , questa lista che chiameremo Listatem contiene tutti gli item aggiunti. Listatem viene mostrata per intero nella pagina MainPage , in una struttura ListView.

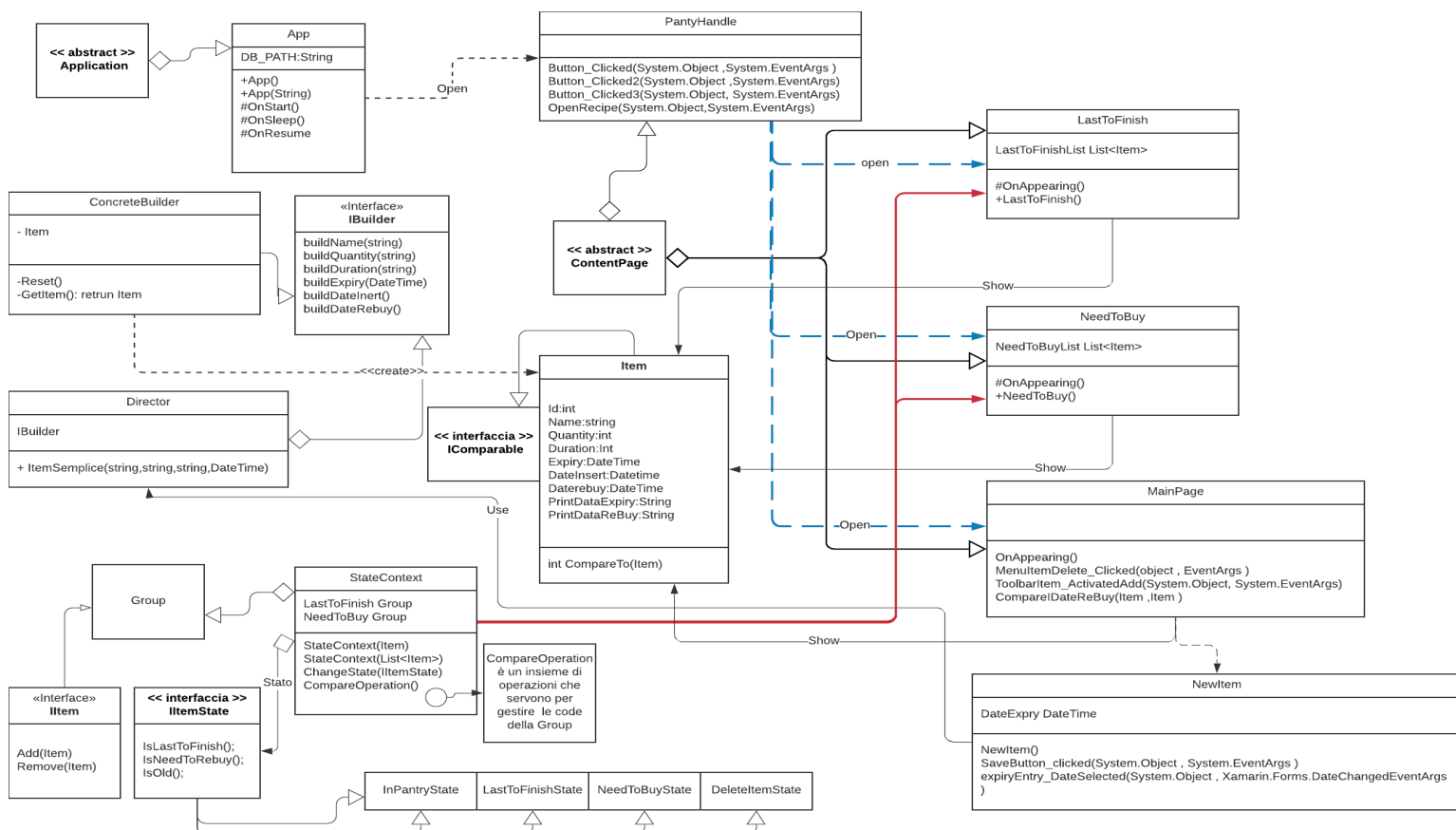
Nelle pagine *LastToFinish* e *NeedToBuy* la Listaltem viene filtrata. Il filtraggio della lista viene deferito allo **State Pattern** il quale differenzia il filtro da usare a seconda dello stato che vogliamo visualizzare.

Esempio: se Nella pagina principale clicchiamo sul bottone che ci porta alla pagina LastToFinish , allora la ListItem verra filtrata per mostrare solamente i prodotti che stanno per terminare e che soddisfano certe condizioni.

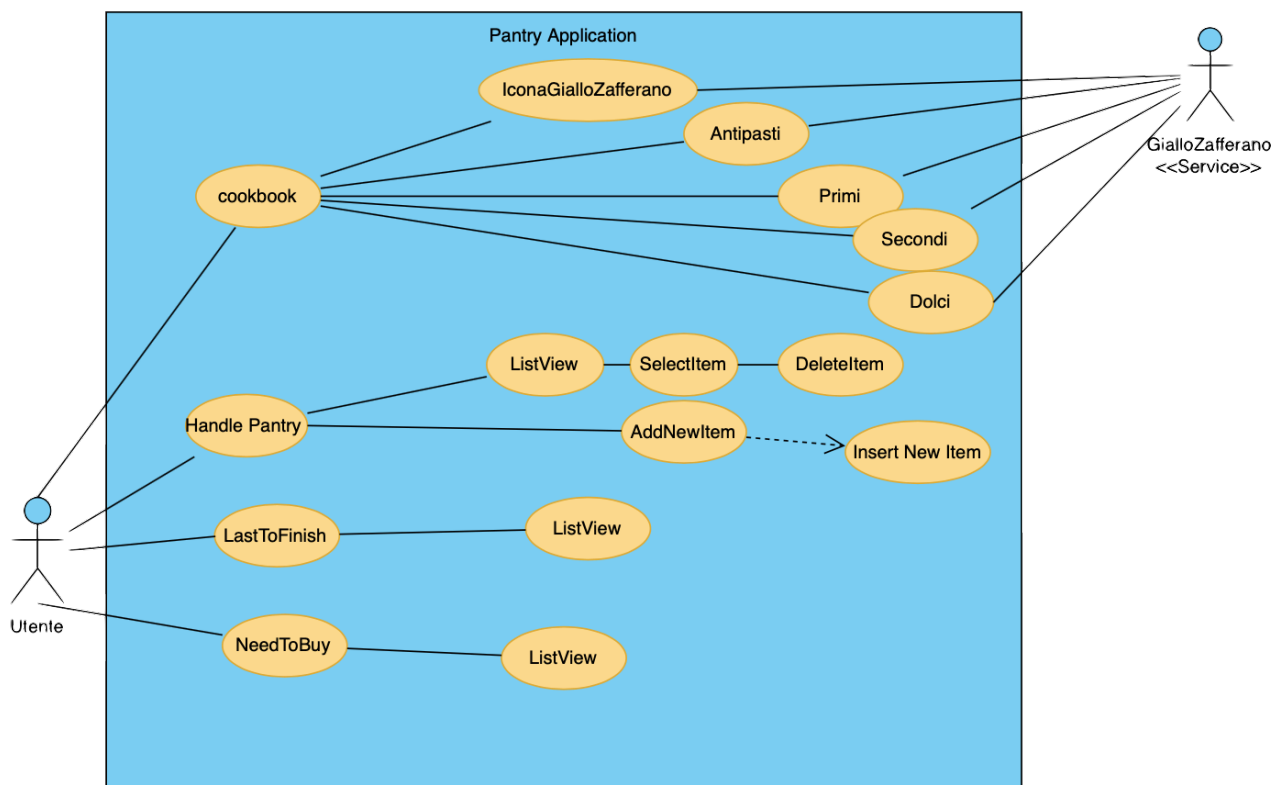
Si possono immaginare le 3 pagine iniziali come Stati diversi di un prodotto, quindi lo StatePattern mi è sembrato adeguato per l'utilizzo.

Per agevolare ulteriormente il lavoro futuro ho voluto implementare anche il **CompositePattern**. Infatti ho utilizzato diverse classi per ogni stato in modo da avere una struttura Lista separata semplice da gestire e da mantenere dove collocare gli oggetti già filtrati.

Il *Composite Pattern* e lo *State Pattern* sono in stretta relazione e lavorano insieme.



## Utilizzo



L'applicazione ha un'interfaccia abbastanza intuitiva. Appena viene aperta ci viene mostrata una schermata avente tre bottoni principali, più un bottone sulla barra superiore il quale ci porta alla sezione CookBook in cui sono presenti le ricette usufruibili dall'utente utilizzando la connessione dati e accedendo alla pagina ufficiale di GialloZafferano.

I Tre pulsanti principali sono HandlePantry, LastToFinish, NeedToBuy. I nomi sono abbastanza esplicativi ma analizziamo nel dettaglio ogni bottone per avere un'idea più chiara su come utilizzare questo programma di gestione.

### o HandlePantry

premendo questo bottone ci viene aperta una pagina, la quale è una ListView popolata con tutti gli item inseriti nell'applicazione. In questa pagina e solo in questa pagina è possibile modificare direttamente la *ItemList*, insomma, la lista principale che contiene tutti i prodotti. Attraverso un Bottone posto sulla parte in alto a destra nominato "Add" è possibile aggiungere un nuovo Item. Se cliccato aprirà una pagina finalizzata all'inserimento dei dati relativi all'oggetto che vogliamo aggiungere. Questa applicazione mira ad evitare lo spreco del cibo e ad avere una dispensa organizzata.

Quindi i campi di inserimento degni di un attimo di attenzione sono "*Quantity*" e "*Numer of days this product stay in your pantry*". Infatti Questi due valori sono fondamentali per una previsione accurata della data di ReBuy. Maggiore è l'attenzione dell'utente nell'inserire valori più possibile vicini alla realtà e migliori saranno i risultati.

Se nella pagina HandlePantry selezioniamo un oggetto della lista cliccandoci sopra, allora apparirà un nuovo bottone, sempre nella stessa posizione di "Add". Questa volta il bottone si chiama "Delete" e serve per eliminare un oggetto.

○ LastToFinish

In questa pagina è possibile consultare gli alimenti che stanno per finire , viene mostrato per ogni oggetto il nome, la data di scadenza, la data di ReBuy, e il numero di oggetti inseriti.

○ NeedToBuy

Questa pagina è una sorta di Lista della Spesa poiché se il valore di Rebuy è vicino alla realtà , allora qua ci saranno effettivamente tutti gli oggetti che sono terminati nella dispensa e che necessitano di essere ricomprati. Gli oggetti rimangono in questa pagina per un certo periodo, poi vengono eliminati automaticamente .

○ CookBook

Viene aperta una pagina locale. Se si clicca sulla icona di Giallo Zafferano viene aperta la main page del blog stabilendo una connessione ad internet. Subito sotto ci sono altri tre pulsanti che ho voluto inserire per rendere l'accesso alle ricette più immediato. Ogni pulsante apre la rispettiva pagina nel Blog di GialloZafferano i pulsanti sono: antipasti, primi, secondi, ,dolci.