

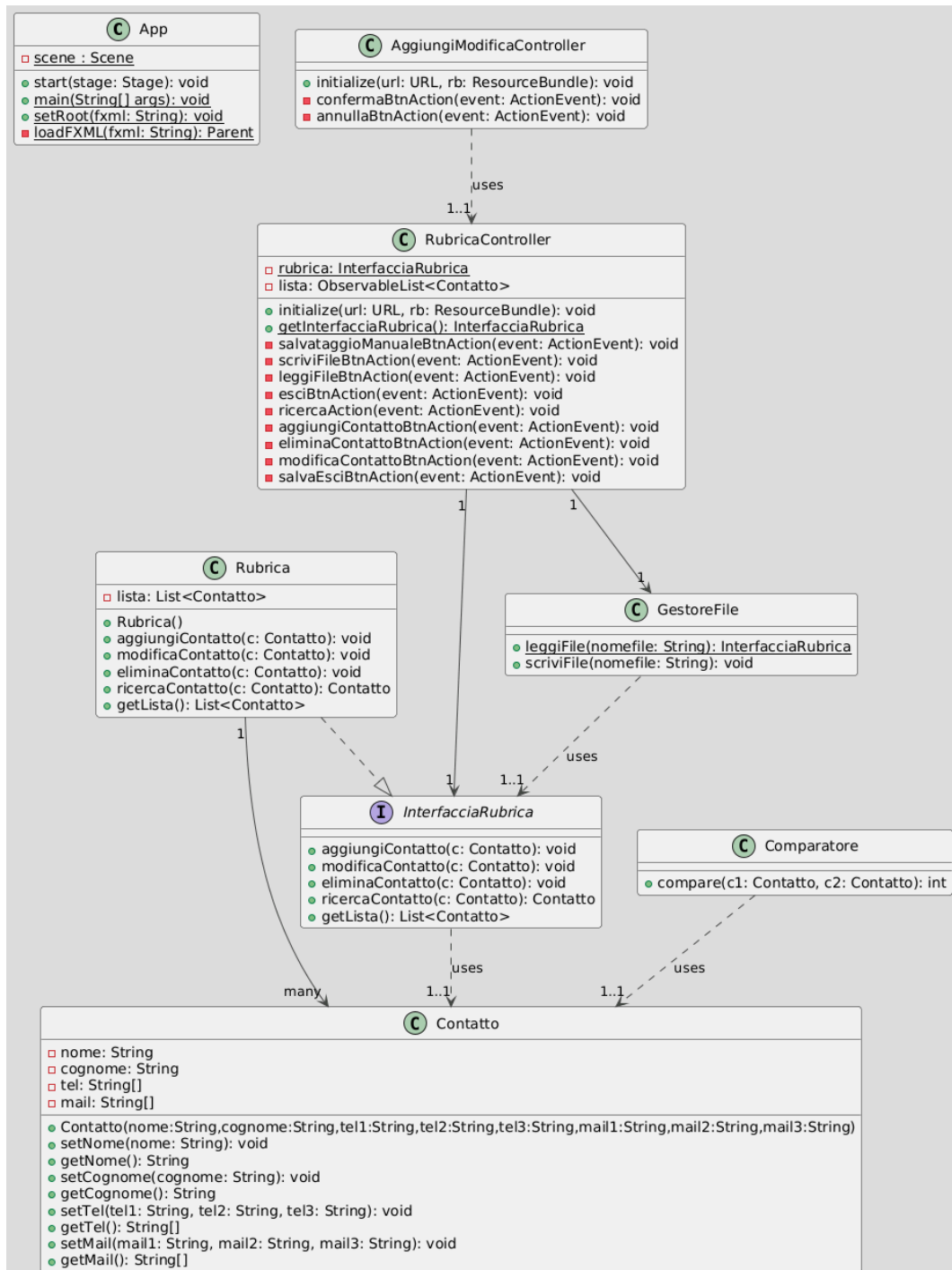
# Documento Design

## Approccio di Scomposizione

La decisione sulla scomposizione del Sistema, è stata fatta tramite l'approccio di Scomposizione Object-Oriented, esso ci permette di poter suddividere il problema da dover risolvere in diversi componenti (classi/oggetti) che rappresentano singolarmente una parte del dominio del problema.

## Diagramma delle Classi

Tramite il Diagramma delle Classi mostriamo, come nell'effettivo abbiamo scomposto il Sistema, con le annesse informazioni di Dipendenza tra i singoli componenti e le Interfacce fornite da ognuno di essi.



# Valutazione Scomposizione

## Coesione

Classe	Livello di Coesione	Descrizione
Comparatore	Coesione Funzionale	La classe Comparatore permette di confrontare due contatti in modo tale che si possano ordinare su cognome e nome.
Contatto	Coesione Funzionale	Tale classe gestisce tutte le informazioni di un contatto, in modo tale che si possano leggere e modificare singolarmente.
RubricaController	Coesione Funzionale	Cattura le azioni dell'utente sulla vista, in modo tale che sia possibile effettuare le operazioni di ricerca, eliminazione, aggiunta e modifica di un contatto e le operazioni di salvataggio, import/export e inizializzazione della Rubrica che vengono implementate da altre classi.
AggiungiModificaController	Coesione Funzionale	Cattura le azioni, dell'utente sulla vista, di conferma o annullamento delle operazioni di aggiunta o modifica di un contatto.
GestoreFile	Coesione Funzionale	Gestisce le operazioni di salvataggio, import/export e inizializzazione della rubrica.
InterfacciaRubrica	Coesione Funzionale	Definisce le operazioni di: Aggiunta, Modifica, Eliminazione e Ricerca di un contatto e l'operazione di restituzione della lista dei contatti.
Main	Coesione Funzionale	Il suo compito è quello di richiamare e visualizzare la scena da un File FXML.
Rubrica	Coesione Funzionale	Gestisce una lista di contatti tramite le diverse operazioni definite nell'InterfacciaRubrica.

## Accoppiamento

Classi	Livello di Accoppiamento	Descrizione
Contatto e Comparatore	Accoppiamento per Dati	La classe Comparatore utilizza gli oggetti della classe Contatto per confrontarli.
Contatto e InterfacciaRubrica	Accoppiamento per Dati	L'InterfacciaRubrica utilizza, come parametro dei metodi, un oggetto della classe Contatto.
Contatto e Rubrica	Accoppiamento per Dati	La classe Rubrica richiama i soli metodi pubblici della classe contatto.
Controller1 e Controller2	Accoppiamento per Dati	I Controller si scambiano le informazioni necessarie per aggiungere/modificare un contatto nella Rubrica.
GestoreFile e Controller1	Accoppiamento per Dati	Il Controller1 istanzia un oggetto di GestoreFile, sul quale richiama i soli metodi pubblici passandogli il solo nome del file.
InterfacciaRubrica e Controller1	Accoppiamento per Dati	Il Controller1 istanzia un oggetto InterfacciaRubrica sul quale richiama i soli metodi pubblici.
InterfacciaRubrica e Controller2	Accoppiamento per Dati	Controller2 utilizza i metodi pubblici della classe InterfacciaRubrica.
Controller1 e Rubrica	Accoppiamento per Dati	Il Controller1 utilizza l'implementazione dei metodi in Rubrica quando istanzia un oggetto di classe InterfacciaRubrica.
GestoreFile e Contatto	Accoppiamento per Dati	Il GestoreFile istanzia oggetti Contatto e usa i suoi metodi pubblici.
InterfacciaRubrica e GestoreFile	Accoppiamento per Dati	Il GestoreFile usa InterfacciaRubrica istanziando un oggetto in un suo metodo all'interno del quale scrive, ciò che viene letto da file.

### Nota:

La classe Main non presenta alcun accoppiamento con le altre classi

## Requisiti Di Buona Progettazione:

Con l'implementazione presentata, possiamo affermare di aver seguito diversi principi di buona progettazione e diversi attributi di qualità (**QA**). In particolare, abbiamo applicato i principi del **S.O.L.I.D.**.

Come il **Principio della Singola Responsabilità** (Separation of Concerns). Questo è stato possibile grazie alla suddivisione delle diverse tipologie di problemi in più classi, un approccio che riesce anche a garantire due attributi di qualità interna, ovvero la **manutenibilità** e la **riusabilità** del sistema.

Abbiamo inoltre rispettato il **Principio Aperto/Chiuso**, implementando l'interfacciaRubrica e usando l'incapsulamento. Questo perché ci consente di definire diverse implementazioni senza modificare il sistema esistente, e anche un limite alle modifiche del sistema. In questo modo, il sistema rimane chiuso alle modifiche, ma aperto all'estensione.

Il **Principio di Sostituzione di Liskov** è stato rispettato poiché l'oggetto rubrica, presente all'interno del Controller può essere sostituito con diverse implementazioni dell'InterfacciaRubrica, perché viene usato un UP-Cast, e questo permette di mantenere un corretto funzionamento del sistema.

Inoltre l'implementazione dell'interfacciaRubrica contribuisce anche all'introduzione del **QA Interno** per la **Modularità** del sistema. Poiché grazie ad essa possiamo ridefinire nuove Rubriche con modifiche o aggiunte di funzionalità senza avere alcun problema sul funzionamento complessivo del sistema.

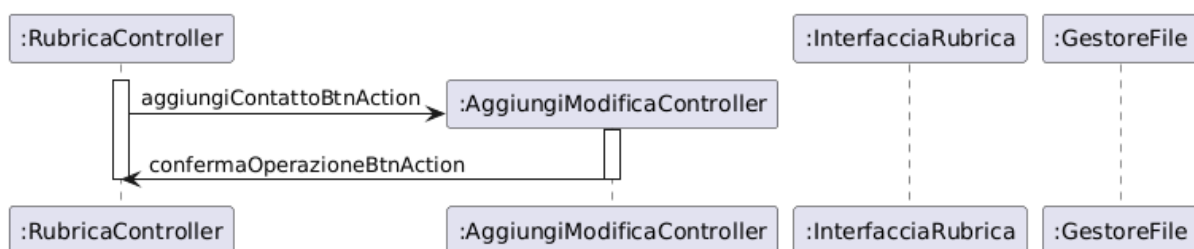
Oltre al S.O.L.I.D. sono rispettati anche altri Principi di buona progettazione come il **Dry - Don't Repeat Yourself** grazie alla definizione della classe Comparatore che ci permette di evitare di ripetere in diversi punti del codice l'implementazione di una classe anonima per andare a comparare i contatti.

Il **Principio della Minima Sorpresa** grazie all'utilizzo delle convenzioni CamelCase (per metodi e attributi) e Upper CamelCase (per classi e interfacce) e all'utilizzo di nomi per classi,metodi e attributi abbastanza esplicativi.

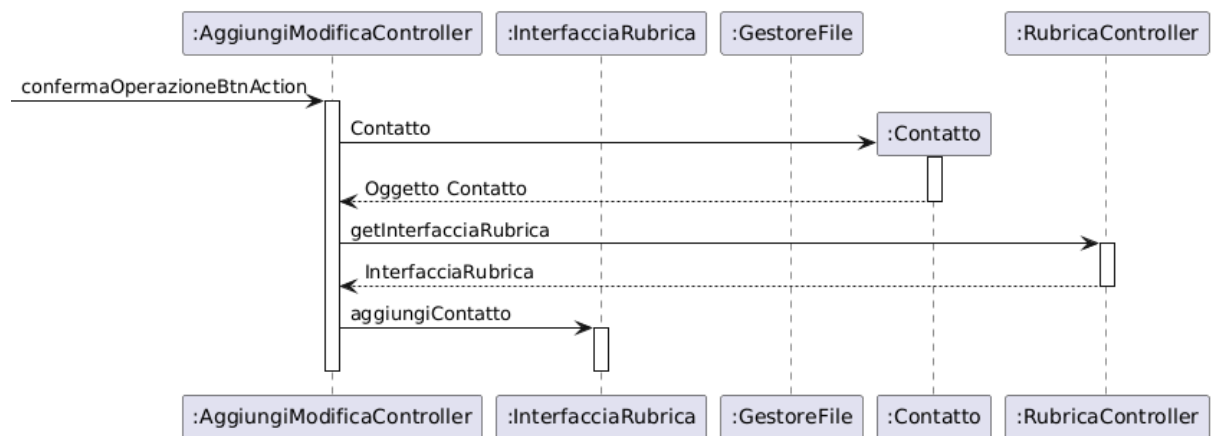
Infine abbiamo favorito l'utilizzo delle **relazioni di associazione rispetto alla Ereditarietà** tra le classi.

---

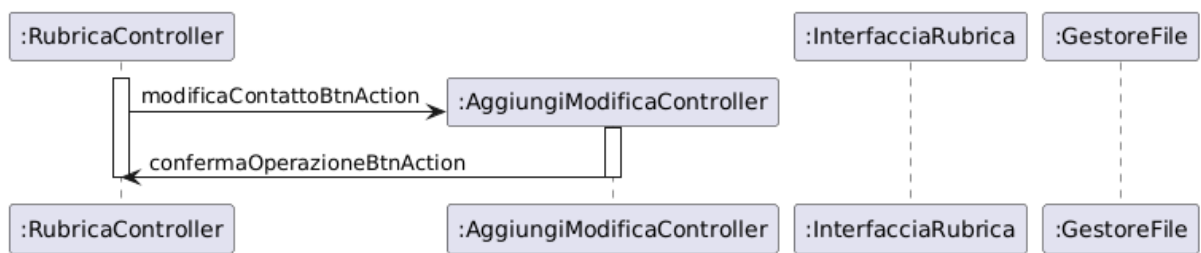
## Diagrammi Delle Interazioni



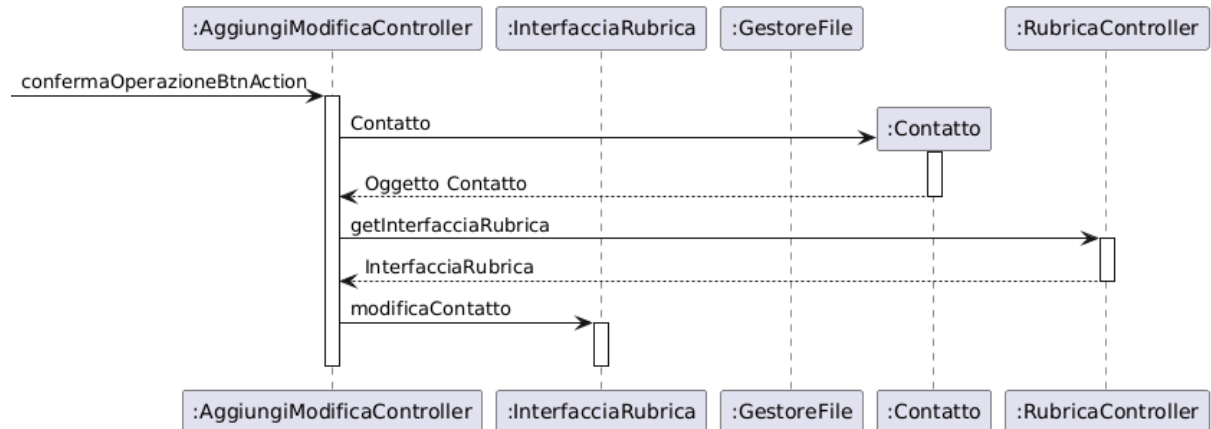
Operazione Aggiungi Contatto



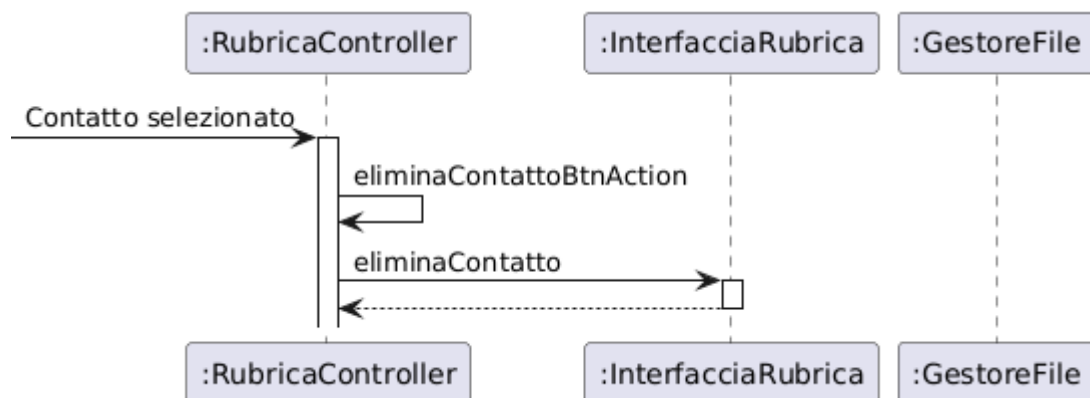
Dettaglio Operazione di Aggiunta



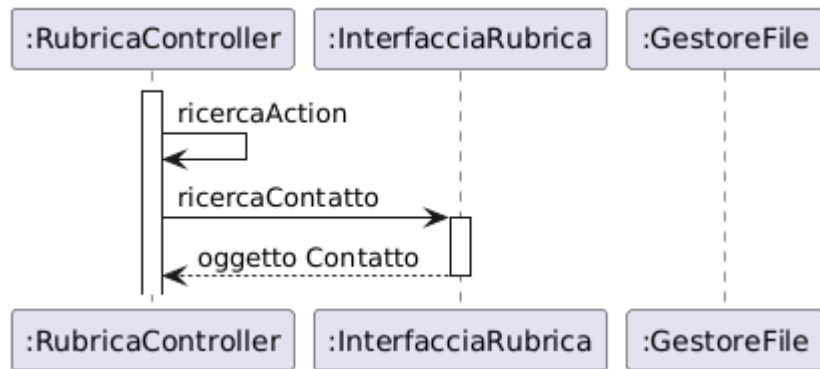
Operazione Modifica Contatto



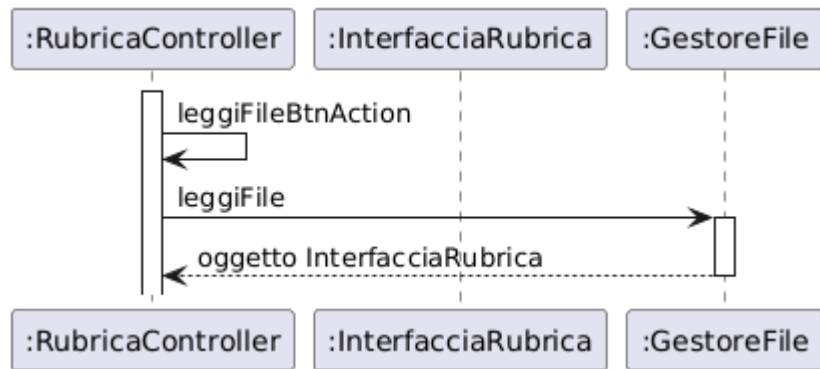
Dettaglio Operazione di Modifica



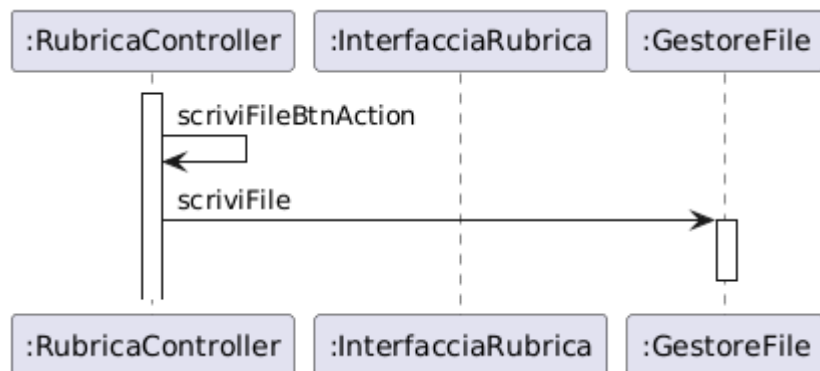
Elimina Contatto



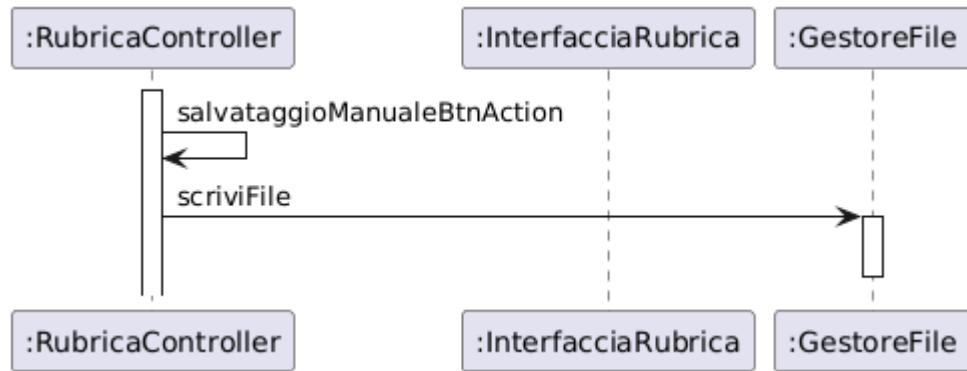
Ricerca Contatto



LeggiFile



ScriviFile



Salvataggio Manuale

## Diagramma Dei Package

