

# Scomposizione in moduli

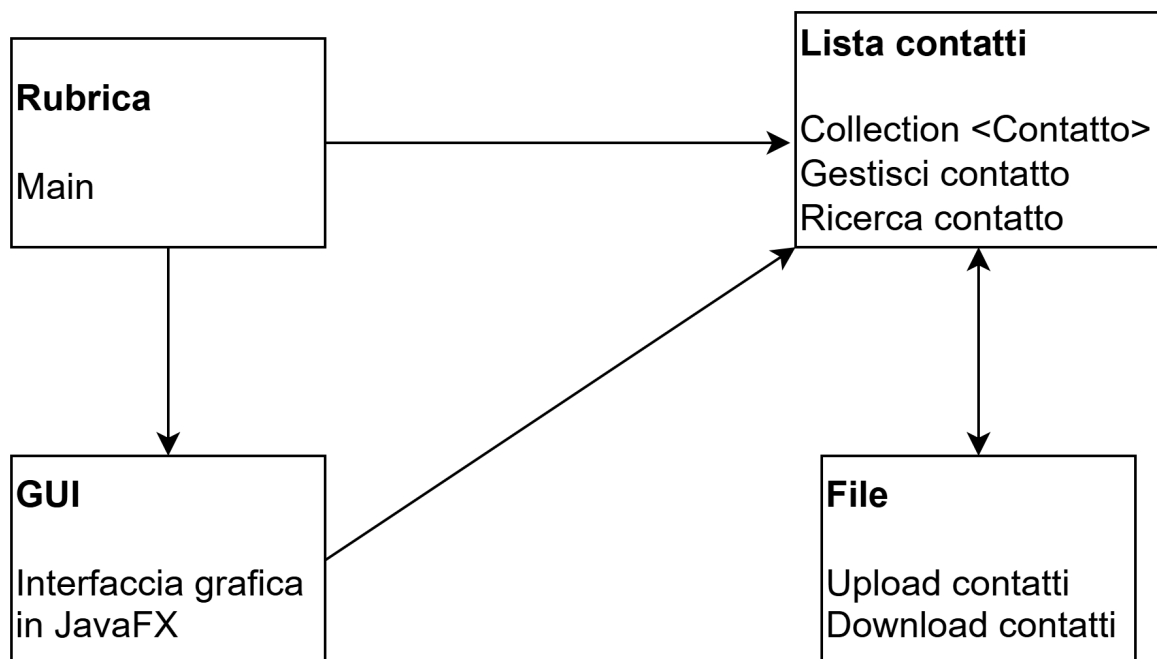
Aspetti principali del programma:

- una classe **main** che gestisce il funzionamento della rubrica, l'upload ed il download da file;
- una **interfaccia grafica** realizzata in JavaFX;
- una struttura dati che gestisca la **lista dei contatti**;
- un **file** per upload e download dei contatti.

Rubrica conterrà sia una struttura dati che gestisca i contatti, sia un oggetto interfaccia grafica.

Interfaccia grafica dovrà accedere alla lista contatti per poterli mostrare.

Lista contatti e file interagiranno reciprocamente per l'upload e il download dei contatti.



# Diagramma delle classi

Le classi principali che abbiamo individuato sono tre:

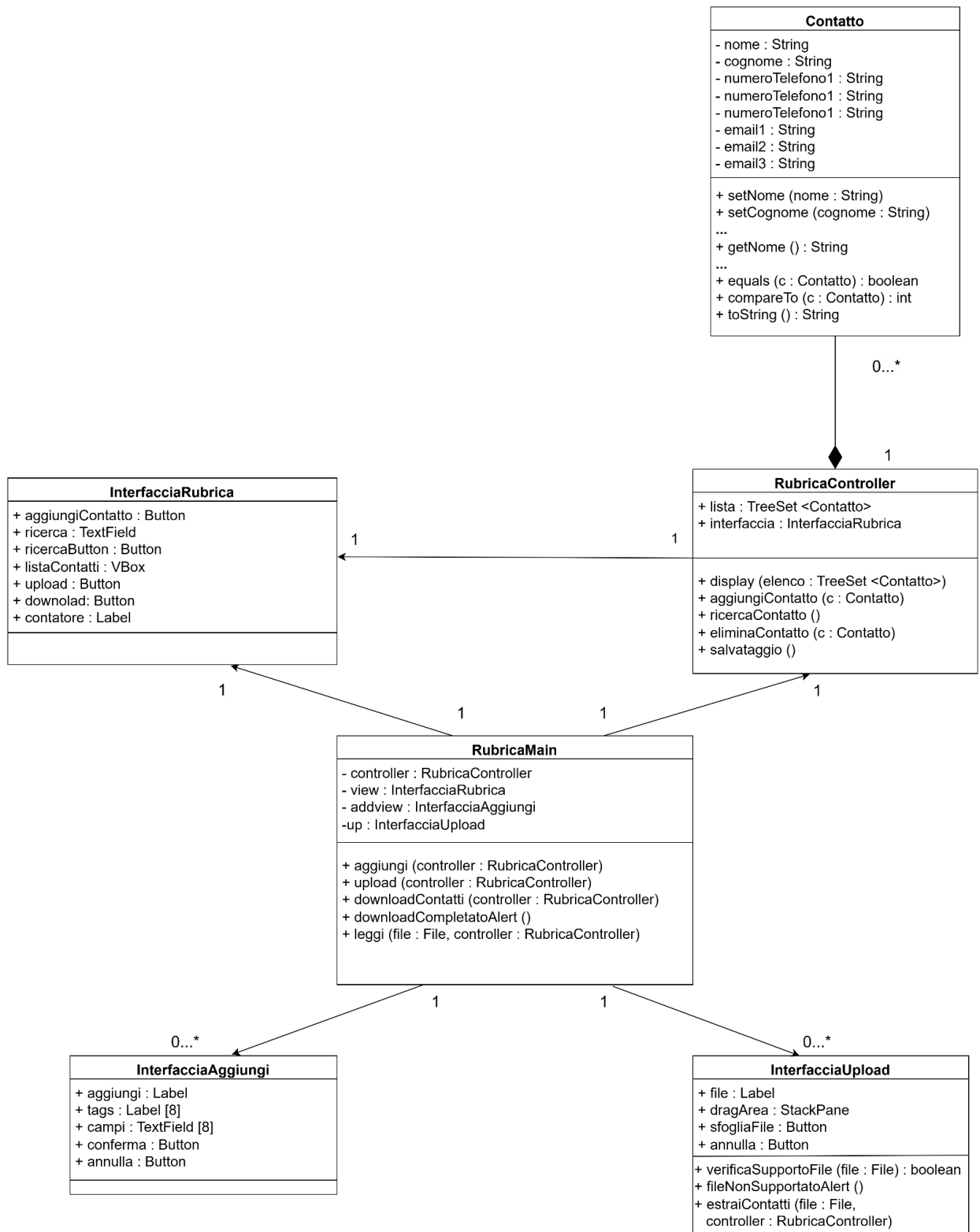
- una classe **Contatto** che contiene tutte le info (nome, cognome, tel, email) e un metodo per modificarlo; numero di telefono e email possono avere fino a 3 campi, quindi abbiamo scelto di implementarli come array di lunghezza 3
- **RubricaMain**, classe main in JavaFX con il compito di avviare l'applicazione, gestire le interazioni dell'utente con l'interfaccia (metodo aggiungi) principale e le varie azioni di upload e download.
- **RubricaController** che si occupa di gestire la lista dei contatti. Contiene una struttura dati di tipo TreeSet (così da poter mostrare i contatti in ordine alfabetico) e l'interfaccia principale (passata tramite costruttore), un metodo per mostrare l'intera lista sull'interfaccia, un metodo per aggiungere un contatto alla rubrica, un metodo per eliminare contatti e un metodo ricerca che prende in input la stringa presente nella barra di ricerca e mostra i soli contatti che contengono quella stringa.
- una classe **InterfacciaRubrica** che mostri la lista dei contatti (sotto forma di più HBox) e i bottoni utili a gestirla: in particolare, un bottone per aggiungere un contatto, un bottone modifica e uno elimina accanto ad ogni contatto, una casella di ricerca, un bottone per il download e uno per l'upload.
- altre classi Interfaccia per guidare l'utente nella gestione della rubrica, come:
  - **InterfacciaAggiungi**, che mostra un'interfaccia con caselle vuote da riempire con le info del nuovo contatto
  - **InterfacciaUpload**, che permette all'utente di caricare (attraverso drag and drop oppure navigazione) il file dal quale estrarre i contatti da salvare in rubrica

La relazione tra RubricaController e Contatto è una **Composizione** poiché Rubrica è una aggregazione di contatti che però può esistere pur essendo vuota, mentre Contatto per esistere deve appartenere alla rubrica.

Le relazioni tra RubricaController, RubricaMain e InterfacciaRubrica sono di **Associazione**:

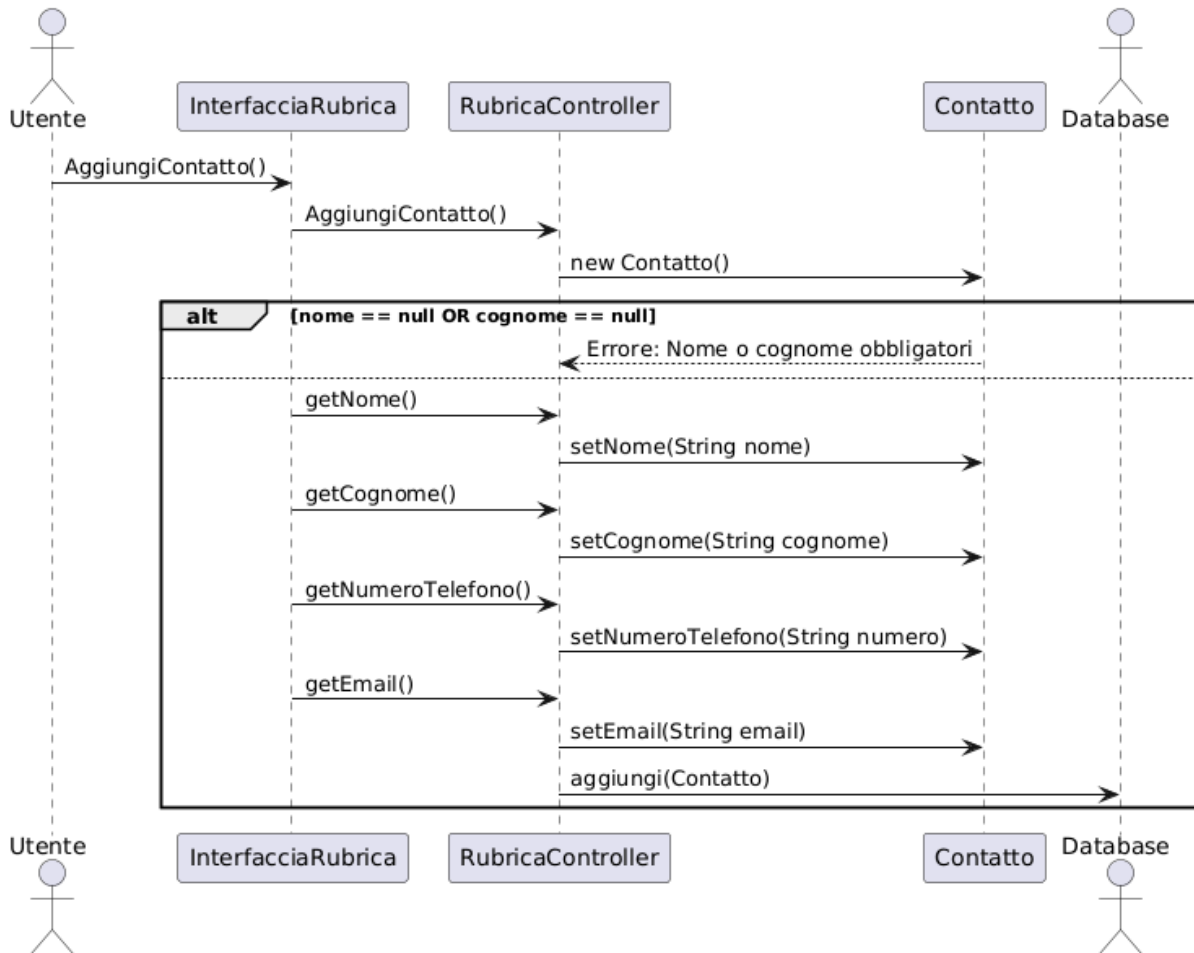
RubricaMain contiene sia un'istanza di RubricaController che una di InterfacciaRubrica, RubricaController contiene un'istanza di InterfacciaRubrica (cioè quella istanziata da RubricaMain e passata come parametro del costruttore).

Anche tra RubricaMain e le classi InterfacciaAggiungi e InteerfacciaUpload c'è **Associazione**. RubricaMain non presenta veri e propri attributi di queste classi, ma esse vengono istanziate nei metodi aggiungi e upload di RubricaMain.



# Diagramma di sequenza

- **AggiungiContatto**



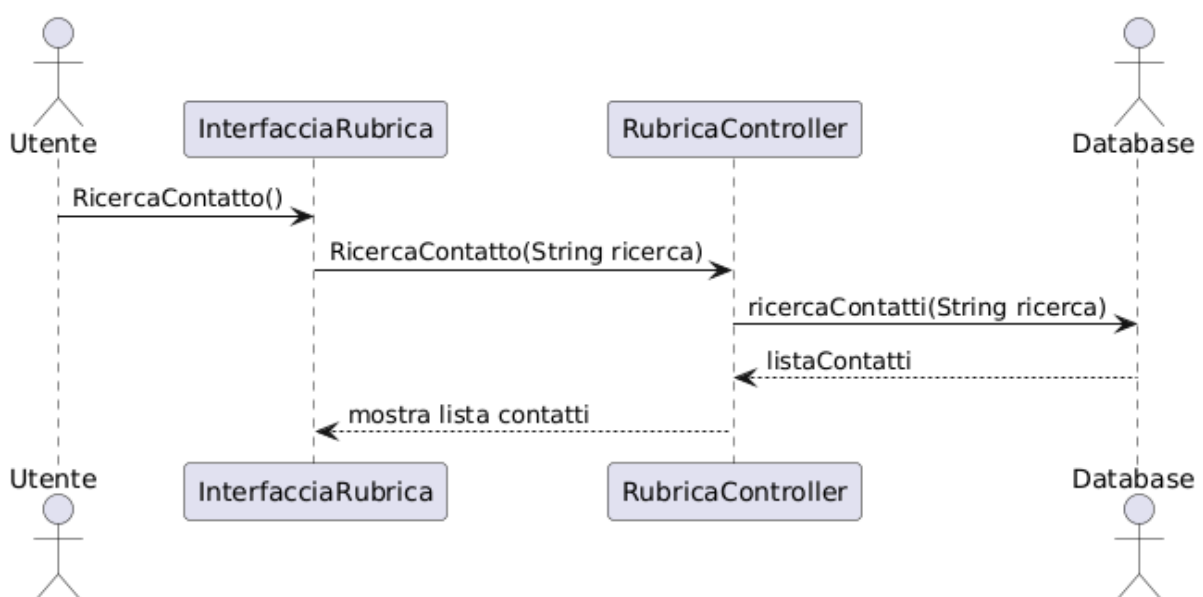
Il diagramma rappresenta il flusso delle chiamate tra i vari componenti del sistema per aggiungere un nuovo contatto alla rubrica.

## FLUSSO DI INTERAZIONI

1. L'interazione inizia dall'Utente, che interagisce con **InterfacciaRubrica**, richiedendo l'aggiunta di un nuovo contatto tramite il metodo *AggiungiContatto()*.
2. **InterfacciaRubrica** invoca poi il metodo *AggiungiContatto()* nella classe **RubricaController**.
3. **RubricaController** crea un'istanza di un nuovo contatto tramite il costruttore *new Contatto()*.

4. Il **RubricaController** richiede all'utente i dettagli del contatto tramite i metodi:
- *getNome()* per ottenere il nome;
  - *getCognome()* per ottenere il cognome;
  - *getNumeroTelefono()* per ottenere il numero di telefono;
  - *getEmail()* per ottenere l'email.
5. Una volta raccolti i dettagli, **RubricaController** utilizza i metodi setter di **Contatto**:
- *setNome()* per impostare il nome;
  - *setCognome()* per impostare il cognome;
  - *setNumeroTelefono()* per impostare il numero di telefono;
  - *setEmail()* per impostare l'email.
6. Se i dati del contatto (nome o cognome) non sono validi, viene gestito un errore (esempio: nome o cognome obbligatori).
7. Una volta completata la configurazione del contatto, **RubricaController** salva il nuovo contatto nel Database chiamando il metodo *setContatto()*.

• **RicercaContatto**



Il diagramma rappresenta il flusso delle chiamate tra i vari componenti del sistema per cercare un contatto presente all'interno della rubrica.

## FLUSSO DI INTERAZIONI

1. L'interazione inizia dall'Utente, che interagisce con **InterfacciaRubrica**, richiedendo la ricerca di un contatto tramite il metodo *RicercaContatto()*.
  2. **InterfacciaRubrica** invoca il metodo *RicercaContatto(String ricerca)* in **RubricaController**, passando la stringa di ricerca inserita dall'utente.
  3. La stringa di ricerca viene passata a **RubricaController**, che effettua un'operazione di *Select* per ottenere una *lista di contatti* corrispondenti alla stringa di ricerca dal Database.
  4. Il Database fornisce l'elenco di contatti corrispondenti ai criteri di ricerca trovati, il quale viene restituito a **RubricaController**.
  5. **RubricaController** invia i risultati a **InterfacciaRubrica**, che li mostra all'utente sotto forma di lista.
-

# Render dell'applicazione

In questa sezione vengono descritte e illustrate le interfacce che appariranno nel programma.

La prima interfaccia, quella principale che corrisponde a **InterfacciaRubrica**, presenta in alto un **tasto** per aggiungere un nuovo contatto alla rubrica e una **casella di testo** dove andare a ricercare i contatti desiderati.

Segue, più in basso, una **lista** contenente tutti i contatti memorizzati in rubrica, e accanto ad ogni contatto è presente un tasto per **modificare** e uno per **eliminare** il contatto indicato.

Infine, sul fondo dell'interfaccia, sono presenti due tasti: uno atto al **download** dei contatti, e un altro atto all'**upload**.

## InterfacciaRubrica:

Rubrica  
Numero di contatti: 4

+ Aggiungi contatto

Cerca

Cognome	Nome	Numero di telefono 1 Numero di telefono 2 Numero di telefono 3	Indirizzo email 1 Indirizzo email 2 Indirizzo email 3	Modifica	Elimina
Cognome	Nome	Numero di telefono 1 Numero di telefono 2 Numero di telefono 3	Indirizzo email 1 Indirizzo email 2 Indirizzo email 3	Modifica	Elimina
Cognome	Nome	Numero di telefono 1 Numero di telefono 2 Numero di telefono 3	Indirizzo email 1 Indirizzo email 2 Indirizzo email 3	Modifica	Elimina
Cognome	Nome	Numero di telefono 1 Numero di telefono 2 Numero di telefono 3	Indirizzo email 1 Indirizzo email 2 Indirizzo email 3	Modifica	Elimina

Download file

Upload file

Andando a cliccare sul tasto **"Aggiungi contatto"** si aprirà questa schermata, ossia **InterfacciaAggiungiModifica**, che presenta un **label** superiore, una casella di testo per indicare il **nome** del contatto, una per il **cognome**, tre per i **numeri di telefono** e tre per le **email**. Una volta terminato l'inserimento delle info, l'utente potrà scegliere se **confermare** o **annullare** il salvataggio del nuovo contatto in rubrica tramite i tasti appositi.



## InterfacciaAggiungiModifica (1):

The image shows a web interface for managing contacts. On the left, there's a sidebar titled 'Rubrica' with the text 'Numero di contatti: 4'. Below this is a table with two columns: 'Cognome' and 'Nome'. The table has four rows, each containing the text 'Cognome' and 'Nome'. In the center, a modal form titled 'Nuovo contatto' is open. This form contains several input fields: 'Cognome', 'Nome', 'Numero di telefono 1', 'Numero di telefono 2', 'Numero di telefono 3', 'Indirizzo email 1', 'Indirizzo email 2', and 'Indirizzo email 3'. Each field has a small 'I' icon inside the input box. At the bottom of the modal are two buttons: 'Salva' and 'Annulla'. On the right side of the background interface, there's a search bar with a 'Cerca' button. Below the search bar is a table with two columns: 'Modifica' and 'Elimina'. This table has four rows, each containing the text 'Modifica' and 'Elimina'. At the bottom right, there's a button labeled 'Upload file'.

Per una questione di riusabilità, possiamo usare l'interfaccia precedente anche per modificare un contatto, andando a cliccare sul tasto **“Modifica”** presente accanto ad ogni contatto, con due sole differenze. La prima differenza è che il **label** superiore indica che stiamo eseguendo la **modifica** del contatto (e non l'aggiunta), per indicare all'utente quale azione sta eseguendo. La seconda differenza è che i **campi** sono già pre-inseriti in base alle info esistenti in rubrica per quel contatto.

## InterfacciaAggiungiModifica (2):

**Rubrica**  
Numero di contatti: 4

Cognome	Nome

**Modifica contatto**

Cognome

Nome

Numero di telefono 1

Numero di telefono 2

Numero di telefono 3

Indirizzo email 1

Indirizzo email 2

Indirizzo email 3

Quando invece l'utente clicca sul tasto **"Elimina"**, presente accanto ad ogni contatto, apparirà questa interfaccia, dove il programma chiede conferma all'utente prima di eliminare il contatto dalla rubrica. Se l'utente **annulla** l'operazione, il contatto rimane in rubrica, se invece l'utente clicca su **conferma**, il contatto viene eliminato.

## InterfacciaElimina:

**Rubrica**  
Numero di contatti: 4

+ Aggiungi contatto

Cerca

Cognome	Nome	Numero di telefono 1	Numero di telefono 2	Numero di telefono 3	Indirizzo email 1	Indirizzo email 2	Indirizzo email 3	Modifica	Elimina

**Vuoi eliminare il contatto selezionato?**

L'interfaccia di upload, che compare quando si clicca sul tasto **“Upload file”**, è composta da un **label**, una **sezione** dove trascinare il file che contiene i contatti, un tasto **“sfoglia”** per creare graficamente un percorso così da aiutare l'utente a raggiungere facilmente la locazione del file, e un tasto **“annulla”** per annullare l'operazione di upload.

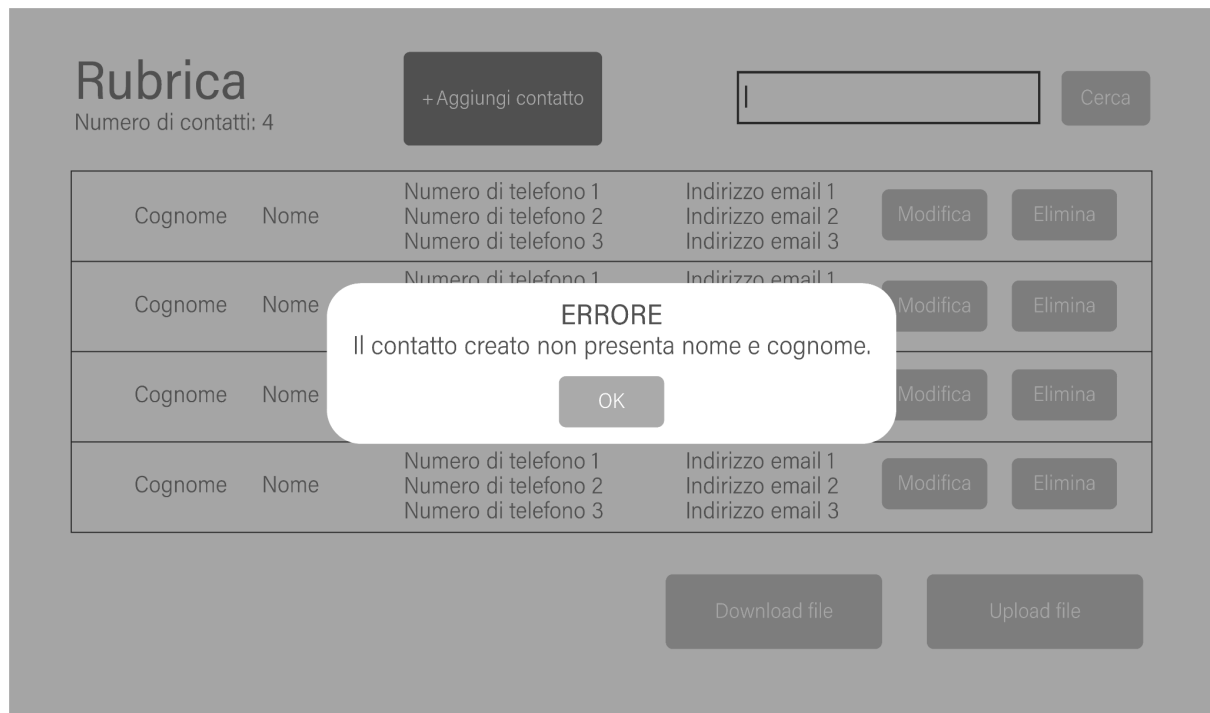
#### InterfacciaUpload:



Qualora l'utente cercasse di creare un contatto dove i campi obbligatori (nome e cognome) non sono stati compilati correttamente, apparirà il **Messaggio di Errore (1)**; l'utente viene così informato dell'errore e può correggersi nell'inserimento successivo. L'interfaccia è composta da un **label** e da un tasto **“ok”** per confermare la presa visione del messaggio.

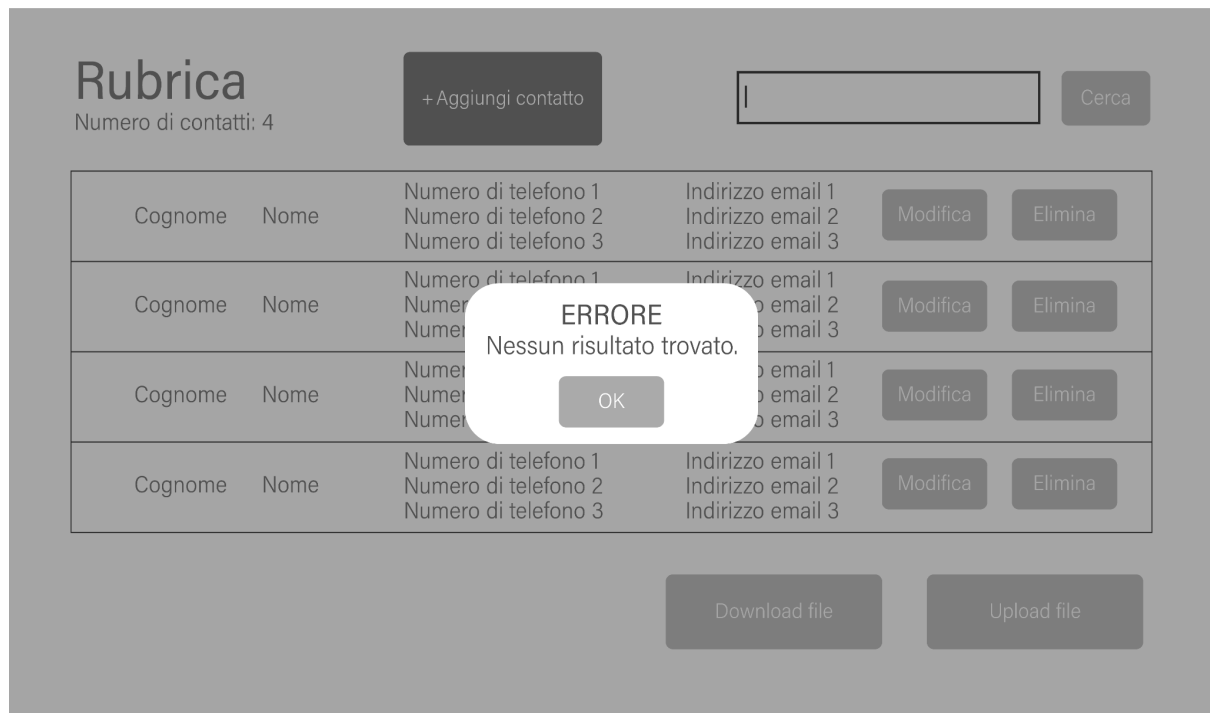
Il messaggio di errore è stato sostituito da un binding che non permette all'utente di salvare un contatto senza nome e cognome.

### Messaggio di Errore (1):



Qualora l'utente effettuasse una ricerca in rubrica senza ottenere corrispondenze tra i contatti, apparirà il **Messaggio di Errore (2)**; l'utente viene così informato che nessun risultato corrisponde ai criteri di ricerca inseriti e può modificarli per eseguire una nuova ricerca. L'interfaccia è composta da un **label** e da un tasto "**ok**" per confermare la presa visione del messaggio.

## Messaggio di Errore (2):



Qualora l'utente tentasse di caricare un file con dei contatti in un formato non supportato dalla rubrica, apparirà il **Messaggio di Errore (3)**; l'utente viene così avvisato del problema e può selezionare un file in un formato adeguato per completare l'operazione. L'interfaccia è composta da un **label** e da un tasto "ok" per confermare la presa visione del messaggio.

Messaggio di Errore (3):

Rubrica

Numero di contatti: 4

+ Aggiungi contatto

Cerca

Cognome	Nome	Numero di telefono 1 Numero di telefono 2 Numero di telefono 3	Indirizzo email 1 Indirizzo email 2 Indirizzo email 3	Modifica	Elimina
Cognome	Nome	Numero di telefono 1	Indirizzo email 1	Modifica	Elimina
Cognome	Nome			Modifica	Elimina
Cognome	Nome	Numero di telefono 1 Numero di telefono 2 Numero di telefono 3	Indirizzo email 1 Indirizzo email 2 Indirizzo email 3	Modifica	Elimina

Download file

Upload file

ERRORE

Il file inserito è di un formato non supportato.

OK

---