

# Sistema di una Biblioteca Digitale

Gennaro Nappo - Giovanni Riccio  
N86004294 - N86004316

Anno accademico 2022-2023



UNIVERSITÀ DEGLI STUDI  
DI NAPOLI FEDERICO II

# Indice

<b>1</b>	<b>Traccia</b>	<b>3</b>
<b>2</b>	<b>Progettazione</b>	<b>4</b>
2.1	Analisi dei requisiti . . . . .	4
2.1.1	Identificazione delle classi . . . . .	4
2.1.2	Identificazione delle associazioni e degli attributi . . . . .	4
2.1.3	Identificazione delle generalizzazioni . . . . .	5
2.1.4	Assegnazione delle responsabilità . . . . .	5
2.2	Diagramma delle classi del dominio del problema . . . . .	7
2.3	Diagramma di dettaglio delle classi nel dominio della soluzione . . . . .	8
2.4	Sequence Diagram . . . . .	9
2.4.1	Login . . . . .	9
2.4.2	Elimina Notifica . . . . .	9

# 1 Traccia

*"Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di una biblioteca digitale. Gli elementi che possono essere inclusi nella biblioteca digitali sono di due tipi: articoli scientifici (o pubblicazioni) e libri (didattici o romanzi). Per ognuno di essi, devono essere specificati il titolo, l'anno di pubblicazione, l'autore (o gli autori), l'editore e definire le modalità di fruizione (cartaceo, digitale o audiolibro). Per i libri è importante definire la data di uscita del libro e la sala/libreria in cui è fatta una eventuale presentazione. Un libro può anche far parte di una collana, la quale può raggruppare tutte le pubblicazioni che condividono una determinata caratteristica (non tutti i libri fanno parte di collane). Per le pubblicazioni, andare a definire in quale rivista (nome, argomento, anno di pubblicazione, responsabile della rivista) o in quale conferenza (luogo della conferenza, data di inizio e data fine conferenza, struttura organizzatrice e responsabile) è stato pubblicato.*

*Per ogni libro, è necessario specificare dove può essere acquistato (libreria, online). Un romanzo può avere anche uno o più seguiti. In tal caso, è importante prevedere un'interrogazione che permette di recuperare tutte le librerie (o siti internet) dai quali è possibile acquistare l'intera serie dei libri. Non appena una serie sarà disponibile per l'acquisto da almeno una libreria, il sistema notificherà la disponibilità all'utente."*

## 2 Progettazione

### 2.1 Analisi dei requisiti

In questa sezione sono individuate le informazioni rilevanti per poter soddisfare le richieste informative e funzionali del sistema, in particolare sono individuate le *Classi* e le loro *Associazioni* e *Responsabilità*.

#### 2.1.1 Identificazione delle classi

Come da richiesta, consideriamo di gestire un'unica grande biblioteca digitale, cioè un catalogo di elementi che gli utenti possono consultare, quindi non necessiteremo di una classe *Biblioteca* per poter rappresentare la biblioteca del mondo reale.

Abbiamo la necessità di dover rappresentare gli elementi inclusi nella biblioteca, cioè gli articoli scientifici e i libri, quindi avremo le classi *Articolo\_Scientifico* e *Libro*.

Devono essere specificati gli autori dei libri e degli articoli scientifici, quindi necessiteremo della classe *Autore*.

Gli articoli scientifici saranno introdotti in dei fascicoli, che a loro volta faranno parte di riviste, quindi avremo bisogno delle classi *Rivista* e *Fascicolo*, in oltre ogni articolo scientifico può essere esposto in una conferenza, quindi necessiteremo della classe *Conferenza*. Ogni libreria sarà gestita da un utente che possiede una partita IVA, quindi avremo bisogno di una classe *Utente*. Le librerie saranno rappresentate dalla classe *Libreria* e gli elementi che possiedono, sono rappresentati dalle sottoclassi della classe *Elemento*.

Si richiede di gestire eventuali presentazioni dei libri, ed eventuali collane alle quali apparterranno dei libri, quindi necessiteremo delle classi *Collana* e *Presentazione*.

Una serie sarà formata da un certo numero di libri e sarà rappresentata dalla classe *Serie*.

#### 2.1.2 Identificazione delle associazioni e degli attributi

La classe *Utente*, con attributi *Username*, *Password*, *Email*, *Nome*, *Cognome*, *DataNascita* e *PartitaIVA*, avrà un'associazione una a molti *Gestisce* con la classe *Libreria*, con attributi *Nome*, *NumeroTelefonico*, *Indirizzo* e *SitoWeb*, in questa associazione l'Utente ha il ruolo di Gestore. Per permettere al sistema di specificare dove può essere acquistato ogni elemento (fascicolo di una rivista, libro o serie di libri) posseduto da una libreria, la classe *Libreria* avrà anche un'associazione molti a molti *Possesso* con la classe *Elemento*, che ha l'attributo *AnnoPubblicazione*, questa associazione avrà come attributi *Quantita* e *Fruizione* per avere traccia della quantità e delle modalità di fruizione degli elementi disponibili di ogni libreria. La classe *Fascicolo*, con l'attributo *Numero* è un'aggregazione della classe *Articolo\_Scientifico* con attributi *DOI*, *Titolo* e *AnnoPubblicazione*, mentre la classe *Rivista* è una composizione di *Fascicolo*.

Per gli articoli scientifici è importante avere traccia in quale rivista sono introdotti, ovvero in quale rivista è contenuto il fascicolo in cui è stato pubblicato l'articolo, e in quale conferenza sono stati esposti, quindi la classe *Rivista* avrà i seguenti attributi: *ISSN*; *Titolo*; *Editore*; *AnnoPubblicazione*; *Responsabile* e *Argomento*. In oltre la classe *Conferenza* avrà i seguenti attributi: *Struttura*, che conterrà la struttura organizzatrice; *Luogo*, che sarà formato dall'indirizzo; *DataInizio* e *DataFine*, questa classe sarà in relazione con l'entità *Articolo\_Scientifico* tramite l'associazione molti a molti *Esposto*.

La classe Libro, con attributi *ISBN*, *Genere*, *Editore*, *Lingua* e *Titolo*, avrà un'associazione una a molti *Partecipa* con la classe Presentazione, la quale avrà come attributi *Luogo*, *Data*, *Ora* e *Struttura*.

La classe Autore avrà due associazioni molti a molti *Scrivi\_L* e *Scrivi\_A* rispettivamente con le classi Libro e Articolo\_Scientifico, in oltre avrà i seguenti attributi: *Nome*, *Cognome*, *Nazionalità* e *DataNascita*.

Una collana raggruppa dei libri con una caratteristica in comune, quindi sarà una composizione della classe Libro e avrà i seguenti attributi: *Caratteristica*; *Nome* e *ISSN*.

Una serie sarà formata da un certo numero di libri, quindi la classe Serie sarà una composizione della classe Libro e avrà i seguenti attributi: *ISBN*; *Titolo* e *Nlibri*. Per inviare la notifica agli utenti interessati ad una serie disponibile, utilizzeremo:

- Un'associazione molti a molti *Recensione* tra Utente ed Elemento: permetterà di avere traccia di una valutazione e una recensione, presenti rispettivamente negli attributi *Valutazione* e *Testo*, fatte dall'utente;
- Un'associazione *Notifica* tra Utente e Serie: che rappresenterà tutte le notifiche inviate agli utenti che partecipano alla relazione *Recensione* e che hanno attivato le notifiche per una certa serie, ogni volta che una serie associata a tali utenti è disponibile presso qualche libreria. Questa associazione avrà i seguenti attributi: *Testo*; *Libreria*; *DataInvio*; *OraInvio* e *Lettura*.

### 2.1.3 Identificazione delle generalizzazioni

L'unica generalizzazione identificata è quella della classe *Elemento* e delle sue sottoclassi *Fascicolo*, *Libro* e *Serie*, le quali condividono l'attributo *AnnoPubblicazione*. Questa generalizzazione sarà utilizzata per specificare gli elementi (fascicoli, libri e serie di libri) che possono essere acquistati in qualche libreria.

### 2.1.4 Assegnazione delle responsabilità

La classe Articolo\_Scientifico dovrà: *fornire le informazioni e gli autori degli articoli scientifici; cercare tutti gli articoli scientifici presenti nel sistema e gestire la creazione e l'eliminazione degli articoli scientifici*. La classe Libro, invece, avrà le seguenti responsabilità: *fornire le informazioni e gli autori dei libri; cercare tutti i libri presenti nel sistema, e anche solo quelli preferiti dell'utente; cercare tutti i libri di una collana/serie e gestire la creazione e l'eliminazione dei libri*.

La classe Autore dovrà: *fornire i nomi e i cognomi dei libri e degli articoli scientifici; cercare tutti gli autori di un libro/articolo; assegnare gli autori ai loro libri/articoli*.

La classe Rivista avrà le seguenti responsabilità: *fornire le informazioni delle riviste; cercare tutte le riviste presenti nel sistema e gestire la creazione e l'eliminazione delle riviste*. La classe Fascicolo dovrà: *fornire le informazioni dei fascicoli; cercare tutti i fascicoli presenti nel sistema, e anche solo quelli preferiti dell'utente e gestire la creazione e l'eliminazione dei fascicoli*. Le responsabilità della classe Conferenza saranno le seguenti: *fornire le informazioni delle conferenze; cercare tutte le conferenze presenti nel sistema, e anche solo quelle di un certo articolo scientifico; gestire la creazione delle conferenze, inserendone anche i rispettivi articoli scientifici*. La classe utente avrà le seguenti responsabilità: *fornire e impostare l'username, la password, l'email, il nome, il cognome ed eventualmente la Partita IVA, verificando la correttezza di quest'ultima; registrare gli utenti nel sistema e gestire le modifiche delle informazioni e gli accessi degli utenti*. Le responsabilità della classe Libreria sono le seguenti: *fornire le informazioni delle librerie; cercare le librerie che*

*possiedono un certo elemento; gestire l'eliminazione e la creazione delle librerie evitando che venga registrata una libreria già presente nel sistema. La classe elemento dovrà fornire le informazioni degli elementi.*

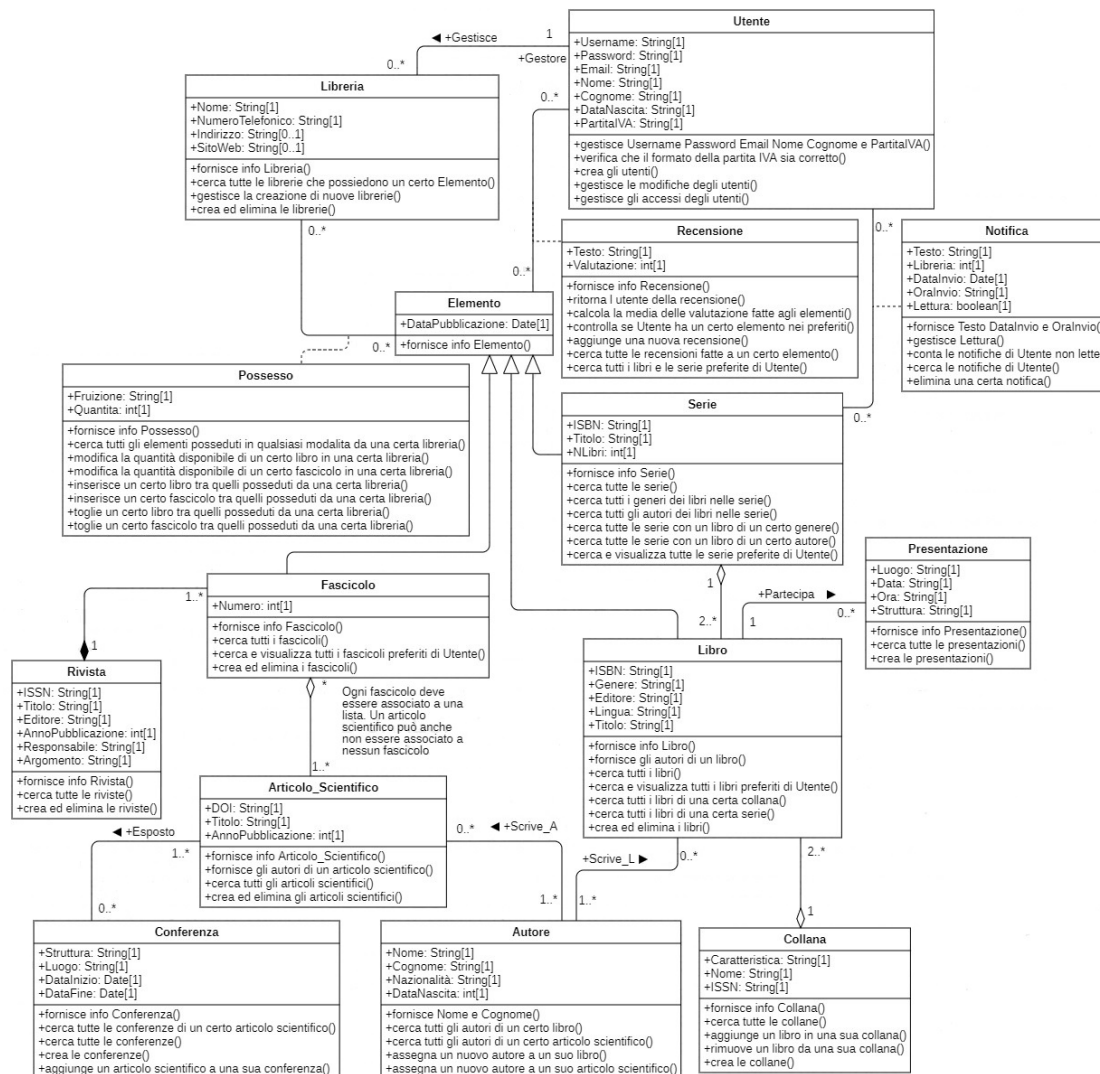
*La classe Collana dovrà: fornire le informazioni delle collane; cercare tutte le collane presenti nel sistema; aggiungere e rimuovere un libro da una collana e gestire la creazione delle collane. La classe Presentazione, invece, avrà le seguenti responsabilità: fornire le informazioni delle presentazioni; cercare tutte le presentazioni presenti nel sistema e gestire la creazione delle presentazioni.*

*La classe Serie dovrà fornire le informazioni delle serie; cercare tutte le serie presenti nel sistema, e anche solo quelle preferite dell'utente; cercare tutti i generi e gli autori dei libri nelle serie e cercare tutte le serie con un libro di un certo genere/autore.*

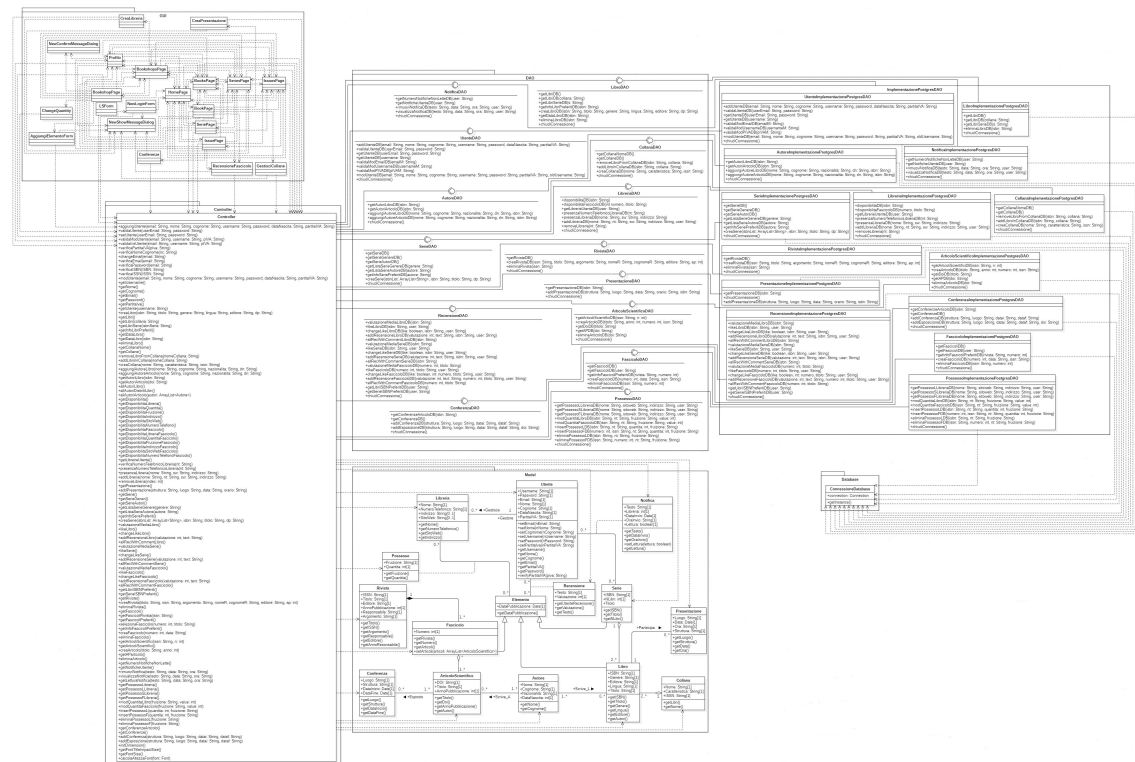
*Possesso dovrà: fornire la modalità di fruizione e la quantità disponibile degli elementi posseduti dalle librerie; cercare tutti gli elementi posseduti da una certa libreria; modificare la quantità disponibile in un certo libro/fascicolo e aggiungere e rimuovere un certo libro/fascicolo tra quelli posseduti da una certa libreria.*

*Recensione avrà le seguenti responsabilità: fornire i testi, le valutazioni e gli autori delle recensioni; calcolare la media delle valutazioni fatte agli elementi; controllare se l'utente ha un certo elemento nei preferiti; gestire la pubblicazione delle recensioni e cercare tutte le recensioni fatte a un elemento e tutte quelle fatte dall'utente. Notifica avrà le seguenti responsabilità: fornire il testo la data e l'ora di invio; segnalare se una notifica è stata letta (o non è stata letta); contare le notifiche non lette dell'utente; cercare tutte le notifiche dell'utente e gestire l'eliminazione delle notifiche.*

## 2.2 Diagramma delle classi del dominio del problema



### 2.3 Diagramma di dettaglio delle classi nel dominio della soluzione



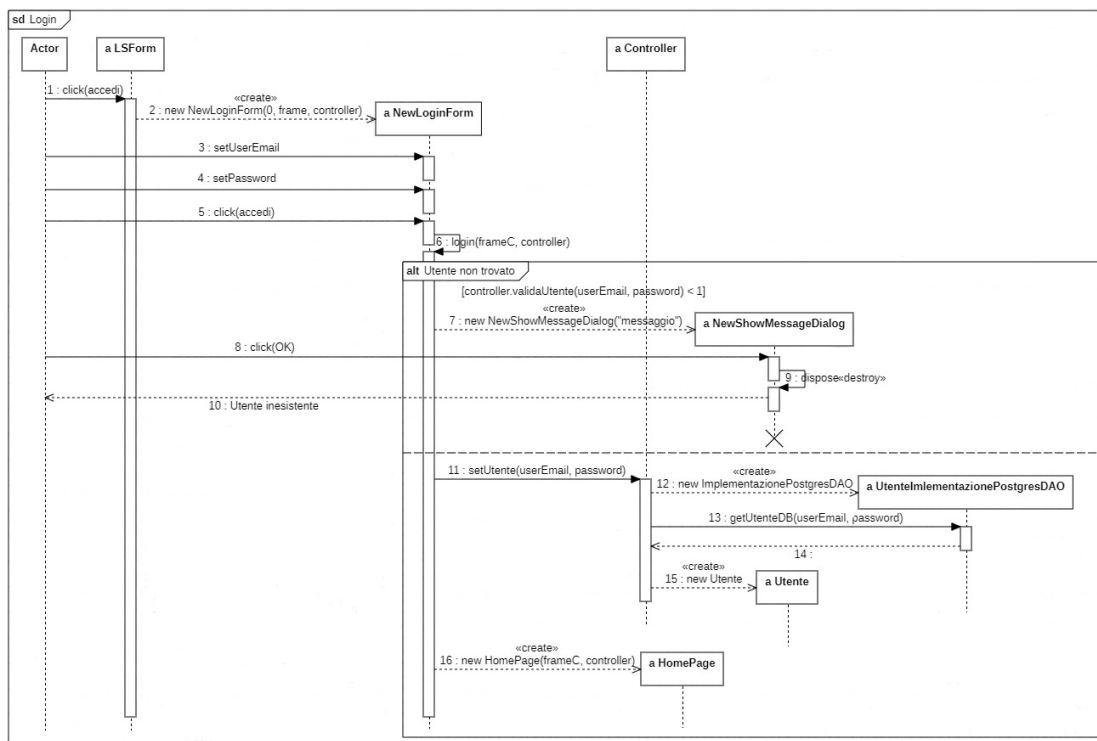


## 2.4 Sequence Diagram

### 2.4.1 Login

In questa sezione viene mostrato il sequence diagram della funzionalità "Login", che permette agli utenti di accedere al sistema.

Dalla pagina iniziale 'LSform' l'utente può cliccare su "accedi" per visualizzare il form presente nella pagina 'NewLoginForm', per inserire l'email o l'username e la password e cliccare su "accedi". Quando l'utente clicca sul bottone "accedi" si verifica se le credenziali sono errate, e nel caso viene mostrato un messaggio di errore da 'NewShowMessageDialog' all'utente, impedendo l'accesso all'utente, altrimenti, se le credenziali sono corrette, la funzione 'setUtente' del 'Controller' cerca tutte le informazioni nel database dell'utente che sta effettuando l'accesso tramite la funzione 'getUtenteDB' di 'UtenteImplementazionePostgresDAO', successivamente con queste informazioni si inizializza 'utente' di 'Controller'. Infine viene mostrata la pagina "HomePage".



### 2.4.2 Elimina Notifica

Questa funzionalità permette agli utenti di eliminare una notifica dalla tabella tra quelle che ha ricevuto.

Quando l'utente visualizza le notifiche che ha ricevuto nell'apposita tabella nella pagina 'Profilo', può scegliere di eliminare una notifica, cliccando con il tasto destro del mouse nella riga corrispondente

alla notifica da eliminare, successivamente può cliccare sulla voce "elimina", quindi viene mostrato all'utente un messaggio di conferma con "NewConfirmMessageDialog". Quando l'utente conferma di voler eliminare la notifica selezionata, allora la funzione 'rimuoviNotifica' del 'Controller' elimina dal database la notifica scelta dall'utente tramite la funzione 'rimuoviNotificaDB' di 'NotificaImplementazionePostgresDAO', successivamente si scorre 'listaNotifiche' di 'controller' finchè non viene trovata la notifica che si sta eliminando per poi rimuoverla dalla lista che si sta scorrendo. Infine la funzione 'setNumeroNotifiche' di 'Profilo' utilizza la funzione 'getNumeroNotificheNonLette' di 'Controller', che tramite la funzione 'getNumeroNotificheNonLetteDB' di 'NotificaImplementazionePostgresDAO' cerca nel database il nuovo numero di notifiche non lette dell'utente, se tale numero è minore o uguale a 0, allora non viene mostrato nessun numero, se altrimenti è maggiore o uguale di 100 viene mostrato il messaggio "99+", altrimenti viene mostrato il numero trovato.

