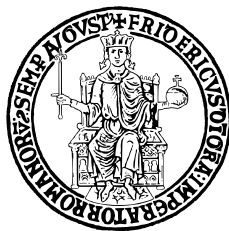


UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II



SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

PROGETTO D'ESAME DI OBJECT ORIENTATION

PROGETTAZIONE E SVILUPPO DI UN  
APPLICATIVO IN JAVA PER LA GESTIONE DI  
CONFERENZE SCIENTIFICHE

**Relatore**

Professore Sergio DI MARTINO

**Candidati**

Antonio CAPORASO

matr: N86003458

Giorgio DI FUSCO

matr: N86004389

Anno Accademico 2022-2023

*Questa pagina è stata lasciata intenzionalmente vuota.*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Definizione del problema . . . . .	5
1.2	Symposium: un applicativo per la gestione di conferenze scientifiche . . . . .	5
1.2.1	Caratteristiche principali di Symposium . . . . .	5
1.2.2	Interfaccia Utente . . . . .	6
<b>2</b>	<b>Progettazione del software</b>	<b>7</b>
2.1	Analisi delle entità . . . . .	7
<b>3</b>	<b>CRC Cards</b>	<b>9</b>
3.1	Suddivisione delle classi . . . . .	9
3.2	Il package <b>Model</b> . . . . .	9
3.2.1	Il subpackage <b>DAO</b> . . . . .	9
3.2.2	Il subpackage <b>Entities</b> . . . . .	9
3.2.3	Il subpackage <b>Utilities</b> . . . . .	9
3.3	Il package <b>Controller</b> . . . . .	9
<b>4</b>	<b>Sequence Diagram</b>	<b>11</b>

# Elenco delle figure

2.1	Class Diagram del modello di dominio . . . . .	8
-----	--	---

# Elenco delle tabelle

2.1	Entità del problema . . . . .	7
-----	-------------------------------	---

# Capitolo 1

## Introduzione

### 1.1 Definizione del problema

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 **conferenze scientifiche**.

Ogni conferenza ha una data di inizio e di fine, una collocazione (sede, indirizzo), uno o più enti che la organizzano, degli sponsor (che coprono in parte le spese), una descrizione, ed un gruppo di organizzatori, che può essere distinto in comitato scientifico e comitato locale (che si occupa cioè della logistica). Di ognuno degli organizzatori, così come di tutti i partecipanti, si riportano titolo, nome, cognome, email ed istituzione di afferenza.

Ogni conferenza può avere una o più sessioni, anche in parallelo fra loro. Ogni sessione ha una locazione all'interno della sede. Per ogni sessione c'è un programma, che prevede la presenza di un coordinatore (chair) che gestisce la sessione, ed eventualmente di un keynote speaker (un partecipante di particolare rilievo invitato dagli organizzatori). Ogni sessione avrà quindi una successione di interventi ad orari predefiniti e di specifici partecipanti. Per ogni intervento si conserva un abstract (un breve testo in cui viene spiegato il contenuto del lavoro presentato).

Si deve poter considerare la presenza di spazi di intervallo (coffee breaks, pranzo) ma anche la presenza di eventi sociali (cene, gite, etc).

### 1.2 Symposium: un applicativo per la gestione di conferenze scientifiche

*Symposium* è un applicativo per la gestione di conferenze scientifiche, sviluppato in JavaFX e basato su PostgreSQL15. Il sistema è progettato per fornire una piattaforma completa e intuitiva per gli organizzatori delle conferenze, consentendo loro di pianificare, gestire e visualizzare le varie attività svolte durante le conferenze.

#### 1.2.1 Caratteristiche principali di Symposium

All'interno di *Symposium* è possibile:

1. **Creare nuove conferenze:** Ogni conferenza viene registrata nel sistema con dettagli come la data di inizio e fine, la collocazione (sede e indirizzo), gli enti organizzatori, gli sponsor coinvolti e ogni eventuale sessione prevista durante la conferenza. Una descrizione dell'evento sarà disponibile per fornire informazioni generali.
2. **Gestione dei comitati:** Per ogni conferenza, sono registrati i dettagli dei membri del comitato scientifico e del comitato locale, che si occupano rispettivamente degli aspetti scientifici e logistici dell'evento.

3. **Gestione della conferenza:** Per ogni conferenza è possibile modificare i suoi dettagli generali (quali il titolo, l'inizio, la fine e la sede), modificare gli enti organizzatori e le varie sponsorizzazioni oppure slittare la conferenza.
4. **Gestione delle sessioni:** Per ogni sessione, è possibile creare un programma dettagliato con gli orari dei vari punti in programma.

### 1.2.2 Interfaccia Utente

L'interfaccia utente di Symposium sarà realizzata utilizzando JavaFX, fornendo un'esperienza utente intuitiva e piacevole. Gli organizzatori possono accedere alla piattaforma per registrarsi e visualizzare i dettagli delle conferenze già presenti nel database. Ogni utente avrà la possibilità di gestire le proprie conferenze, specificare il programma delle sessioni, specificare la nomina dei comitati e altro ancora.

In conclusione, Symposium è un sistema informativo embrionale per la gestione di conferenze scientifiche che offre funzionalità complete e una piattaforma intuitiva per organizzatori.

## Capitolo 2

# Progettazione del software

### 2.1 Analisi delle entità

Le entità che possono essere individuate nel problema sono elencate all'interno della Tabella 2.1. In Figura 2.1 è presente un Class Diagram del modello di dominio.

Entità	Descrizione
<b>Conferenza</b>	Per le conferenze delle quali si vuole poter gestire le informazioni. Di ogni conferenza si conservano il <i>nome</i> , l' <i>inizio</i> e la <i>fine</i> e una <i>descrizione</i> .
<b>Ente</b>	Per gli enti che organizzano le conferenze scientifiche. Di ogni ente si conserva il <i>nome</i> e la <i>sigla</i> .
<b>Sponsor</b>	Per gli sponsor che coprono le spese della conferenza. Di ogni sponsor si conserva il <i>nome</i> .
<b>Comitato</b>	Per i gruppi di organizzatori che si occupano della gestione della conferenza. Si distinguono in comitati <i>scientifici</i> e <i>locali</i> .
<b>Organizzatore</b>	Per i membri dei comitati. Di ogni organizzatore si riportano <i>titolo</i> , <i>nome</i> , <i>cognome</i> , <i>email</i> ed <i>istituzione di afferenza</i> .
<b>Sede</b>	Per descrivere il luogo dove si tengono le varie conferenze. Di ogni sede si conservano il <i>nome</i> , l' <i>indirizzo</i> e la <i>città</i> .
<b>Sala</b>	Per tenere traccia dell'ubicazione delle varie sessioni. Di ogni sala si conserva il <i>nome della sala</i> e la sua <i>capacità</i> .
<b>Sessione</b>	Per rappresentare le sessioni di una conferenza. Per ogni sessione si riporta il <i>titolo</i> , un <i>coordinatore</i> , data e orario d' <i>inizio</i> e di <i>fine</i> .
<b>Programma</b>	Per il programma di ciascuna sessione. Ogni programma specifica la presenza di un <i>keynote speaker</i> , ovvero un partecipante di rilievo.
<b>Intervento</b>	Per i vari interventi di una sessione. Per ogni intervento si conserva un <i>abstract</i> e l' <i>orario</i> dello stesso.
<b>Speaker</b>	Per descrivere chi effettua un intervento.
<b>Partecipante</b>	Per i partecipanti delle varie sessioni. Ogni partecipante ha gli stessi attributi degli organizzatori.
<b>Intervallo</b>	Per descrivere i vari intervalli presenti all'interno di una sessione. Questi possono essere di due tipologie: <i>coffee break</i> oppure dei <i>pranzi</i> . Per ogni intervallo si riporta l' <i>orario</i> .
<b>Evento sociale</b>	Per i vari eventi sociali previsti all'interno di una sessione. Questi possono essere di varia natura. Come per gli intervalli se ne riporta l' <i>orario</i> .
<b>Utente</b>	Per i vari utenti che creano le conferenze all'interno di un applicativo.

**Tabella 2.1:** *Entità del problema*



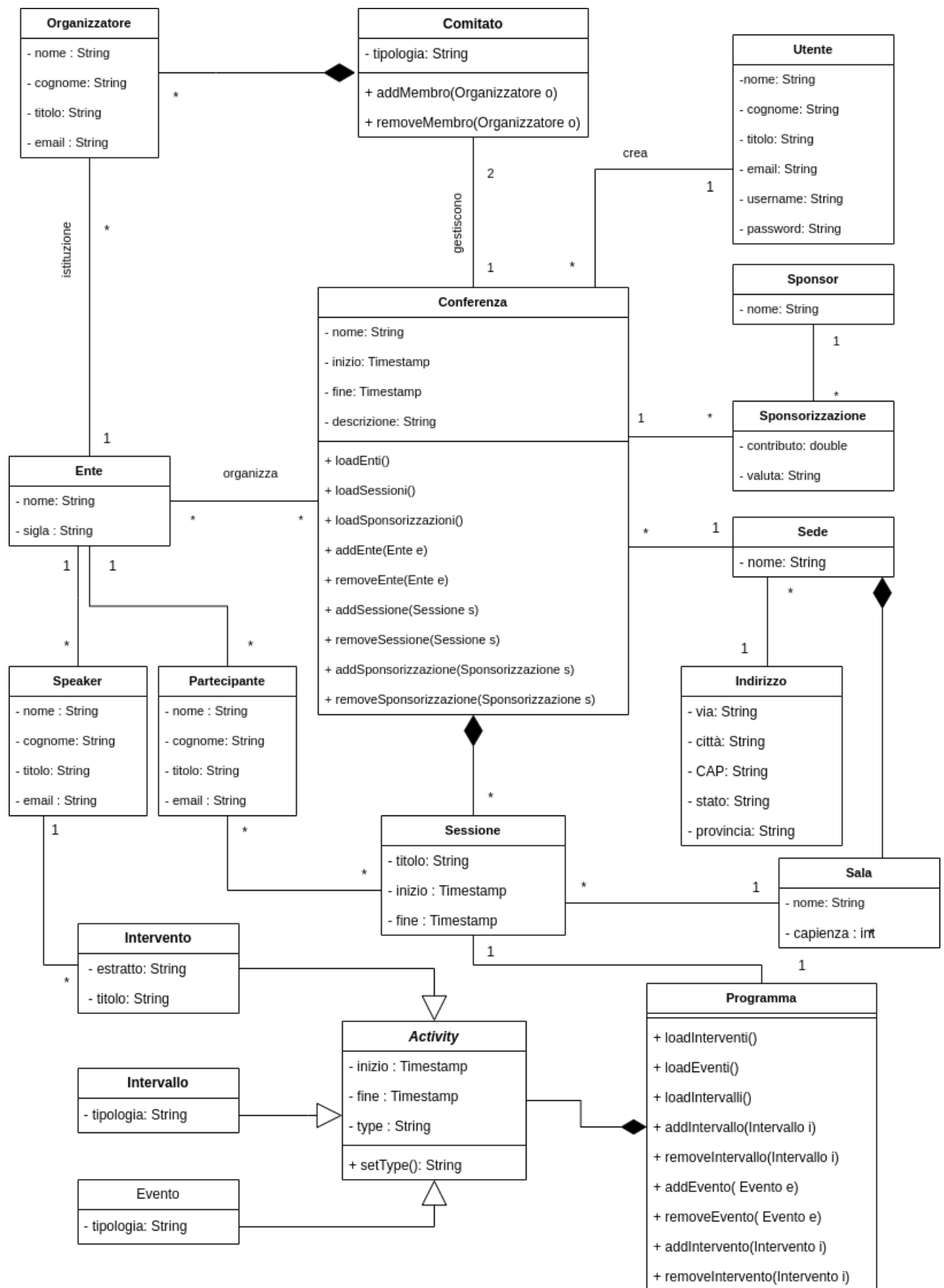


Figura 2.1: Class Diagram del modello di dominio

# Capitolo 3

## CRC Cards

### 3.1 Suddivisione delle classi

Nell'organizzazione del codice sorgente del nostro progetto abbiamo applicato la seguente suddivisione in packages per le classi, catalogandole seguendo l'euristica Model-View-Controller:

1. **Model**

- (a) **DAO**
- (b) **DbConfig**
- (c) **Entities**
- (d) **Utilities**

2. **View**

- (a) **FXML**
- (b) **CSS**

3. **Controller**

4. **Exceptions**

All'interno del package **View** sono presenti tutti i file **.fxml** che descrivono le interfacce grafiche dell'applicazione e i file **.css** utilizzati per la loro personalizzazione, mentre all'interno del package **Controller** sono presenti le classi che implementano i controller delle interfacce grafiche.

All'interno del package **Model** sono presenti vari packages che implementano il modello del nostro progetto come presentato nel Diagramma 2.1. Nel sub-package **DAO** sono state inserite tutte le classi utilizzate per implementare il pattern DAO, mentre nel sub-package **DbConfig** è presente la classe per la configurazione del database. Infine, nel sub-package **Utilities** sono presenti le classi che ci sono state necessarie per la gestione di molteplici istanze delle classi di dominio.

### 3.2 Il package Model

#### 3.2.1 Il subpackage DAO

#### 3.2.2 Il subpackage Entities

#### 3.2.3 Il subpackage Utilities

### 3.3 Il package Controller

<b>Nome della classe</b>	<b>Superclasse</b>
<b>Sottoclassi</b> Eventuali classi derivate	
<b>Responsabilità</b> Responsabilità degli oggetti della classe: di quali informazioni sono depositari, quali azioni eseguono	
<b>Collaborazioni</b> Relazioni con altre classi	

## Capitolo 4

# Sequence Diagram