Immagine che contiene Carattere, testo, logo, cerchio

Il contenuto generato dall'IA potrebbe non essere corretto.

PROGETTAZIONE DI UN CLASS DIAGRAM VOLTO ALL’IMPLEMENTAZIONE DI UN HACKATHON

*Autore:*

*Corso di OBJECT ORIENTATION*

*Gabriele Letizia N86005267*

*Fabio Iannicelli N86005383*

*ANNO ACCADEMICO 2024/2025*

[***PROGETTAZIONE DEL CLASS DIAGRAM*** 2](#_Toc196313611)

[**1.1 | Introduzione** 3](#_Toc196313612)

[**1.2 | Class Diagram** 3](#_Toc196313613)

[***DOCUMENTAZIONE RELATIVA ALLE CLASSI*** 3](#_Toc196313614)

[**2.1 | Classe Hackathon** 4](#_Toc196313615)

[**2.2 | Analisi delle ridondanze** 5](#_Toc196313616)

[**2.3 | Eliminazione delle generalizzazioni** 5](#_Toc196313617)

[**2.4 | Eliminazione degli attributi Multivalore** 5](#_Toc196313618)

[**2.5 | Eliminazione degli attributi strutturati** 6](#_Toc196313619)

[**2.6 | Scelta degli identificatori primari.** 6](#_Toc196313620)

[Capitolo 3 7](#_Toc196313621)

[***DIAGRAMMI RISTRUTTURATI*** 7](#_Toc196313622)

[**3.1 | Diagramma UML (ristrutturato)** 7](#_Toc196313623)

[**3.2 | Diagramma ER (ristrutturato)** 9](#_Toc196313624)

[Capitolo 4 10](#_Toc196313625)

[***PROGETTAZIONE LOGICA*** 10](#_Toc196313626)

[**4.1 | Progettazione Logica** 10](#_Toc196313627)

[**4.2 | Mapping delle entità** 10](#_Toc196313628)

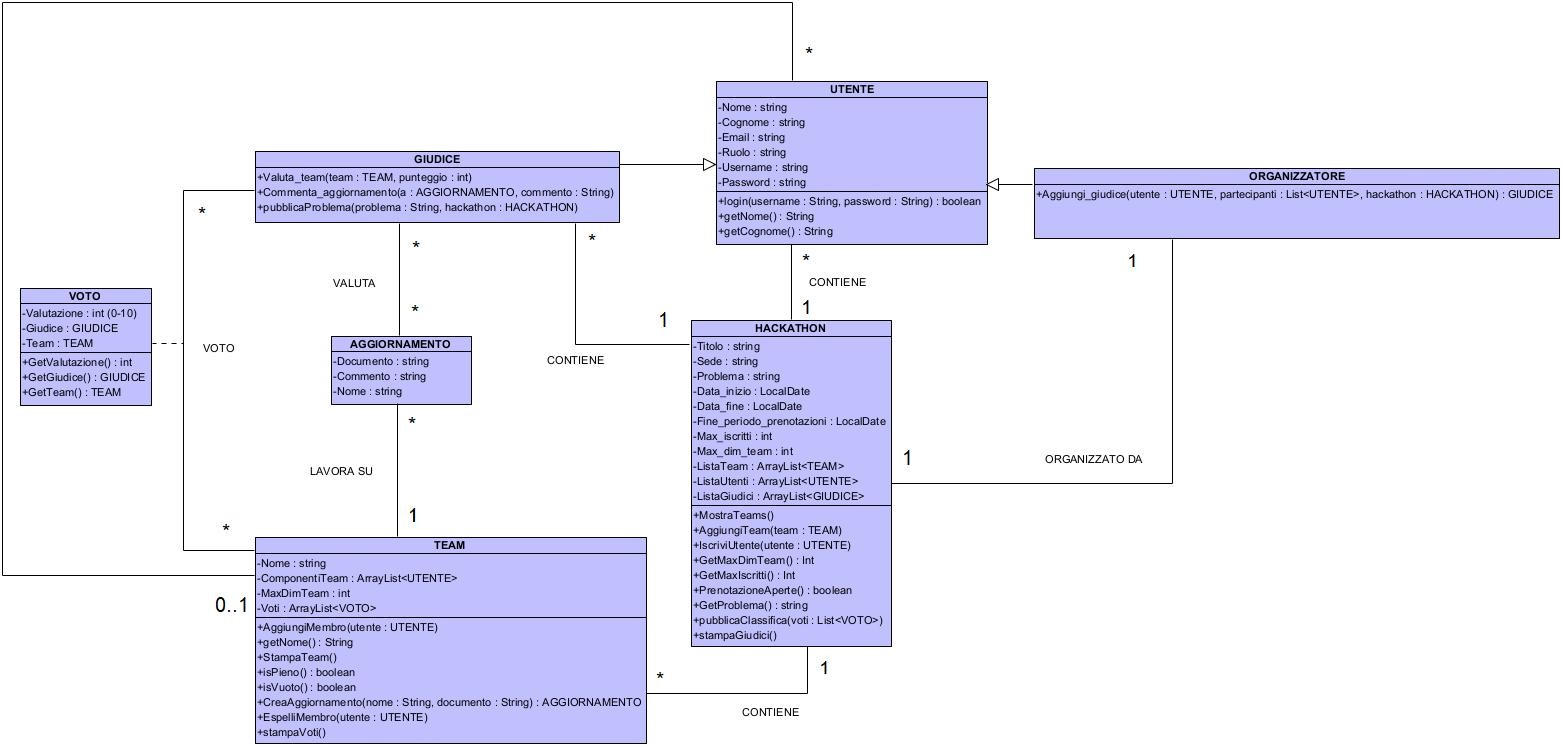
[**4.3 | Mapping delle associazioni** 11](#_Toc196313629)

# ***PROGETTAZIONE DEL CLASS DIAGRAM***

## **1.1 | Introduzione**

Un hackathon, ovvero una "maratona di hacking", è un evento durante il quale, vari team di partecipanti si sfidano per progettare e implementare nuove soluzioni basate su una certa tecnologia o mirate a un certo ambito applicativo.

## **1.2 | Class Diagram**



# ***DOCUMENTAZIONE RELATIVA ALLE CLASSI***

## **2.1 | Classe Hackathon**

Nell’implementazione della classe “Hackathon” , sono stati inseriti vari attributi per descrivere ogni singola edizione.

Ogni Hackathon ha un titolo identificativo con relativa sede, ha una data di inizio e di fine, con una data massima di iscrizione (solitamente due giorni prima della data di inizio).

Inoltre, avremo un massimo numero di iscritti, e una massima dimensione per ogni team, sono state poi inserite delle liste dove verranno salvati tutti i partecipanti, tutti i team e tutti i giudici.

Metodi della classe:

* **MostraTeams()** : stampa a video la lista dei team presenti in quell’edizione dell’Hackathon.
* **AggiungiTeam(team : TEAM)** : passiamo a questo metodo un team che verrà aggiunto alla lista dei team presenti all’Hackathon.
* **IscriviUtente(utente : UTENTE)** : passiamo a questo metodo un utente che verrà aggiunto alla lista degli utenti iscritti all’Hackathon.
* **Int GetMaxDimTeam()** : restituisce la dimensione massima per i team.
* **Int GetMaxIscritti()** : restituisce il numero massimo di iscrizioni possibili all’Hackathon.
* **Boolean PrenotazioniAperte()** : restituisce TRUE se non è stato superato il termine massimo di iscrizione all’Hackathon, FALSE altrimenti.
* **String GetProblema()** : restituisce la traccia del problema, assegnato dai giudici, che i team dovranno risolvere.
* **PubblicaClassifica(voti : List<VOTO>)** : stampa a video i team con la relativa media di voti.
* **StampaGiudici()** : stampa a video la lista relativa ai giudici assegnati all’Hackathon.

## **2.2 | Analisi delle ridondanze**

Una ridondanza in uno schema concettuale corrisponde alla presenza di un dato che può essere derivato da altri dati.

* Nell’entità “Collana” è presente l’attributo “Numero libri” questo è un attributo derivabile e quindi possiamo eliminarlo per evitare ridondanze, poiché possiamo calcolarlo contando il numero di libri associati a quella collana.
* Nell’entità “Collana” sono presenti gli attributi “Data pubblicazione primo libro” e “Data pubblicazione ultimo libro”, anche questi possono essere omessi poiché possono essere ricavati andando a verificare, la data più lontana di pubblicazione per trovare la data del primo libro pubblicato, e quella più recente per l’ultimo.
* Nell’entità “Conferenza” è presente l’attributo “Anno”, quest’ultimo lo possiamo eliminare perché è ricavabile e non svolgere un ruolo fondamentale, può essere ricavato da “Data inizio” e da “Data fine”

## **2.3 | Eliminazione delle generalizzazioni**

In questa fase ci poniamo come obiettivo quello di eliminare le generalizzazioni, come:

* Nel class diagram abbiamo l’entità “Pubblicazione” che ha come specializzazioni le entità “Libro” e “Articolo”, ciò che possiamo fare è accorpare gli attributi del padre nelle entità figlie, poiché non ci sono operazioni che fanno distinzione tra entità padre ed entità figlie, questo accorpamento dal padre nelle figlie è possibile solo se la generalizzazione è totale, e ciò significa che ogni istanza nella superclasse deve essere membro di qualche sottoclasse nella specializzazione.
* L’entità “Articolo” ha due specializzazioni che sono “Articolo per rivista” e “Articolo per conferenza”, questa generalizzazione va tolta, possiamo accorpare le entità figlie nel padre, mantenendo la relazione con “Rivista” e “Conferenza” andando a modificare le cardinalità già presenti.

## **2.4 | Eliminazione degli attributi Multivalore**

In questa fase andremo ad eliminare gli attributi Multivalore. L’unico attributi multivalore presente ancora nel nostro schema è “Autori”, poiché l’attributo “Numero libri” è stato rimosso poiché era una ridondanza. Quindi andremo a definire “Autori” come un entità in relazione con “Libro” e “Articolo”, è stato scelto questo approccio per far si che, cercando un autore, si potesse sapere quali libri fossero in relazione con quell’autore.

## **2.5 | Eliminazione degli attributi strutturati**

Nel nostro schema ristrutturato non potranno poi essere presenti attributi strutturati, come ad esempio: “Editore”, “Responsabile”, “Struttura organizzatrice” e “Luogo”, anche “Autori” era un attributo strutturato, ma nel punto precedente lo abbiamo già definito in un'altra entità. Consideriamo ogni singolo attributo e vediamo cosa converrebbe fare:

* Per “Luogo” inserisco tutti i suoi attributi direttamente nell’entità, quindi “Città” e “Stato” saranno due attributi direttamente presenti nell’entità “Conferenza”
* Anche per “Responsabile” faremo la stessa cosa fatta per “Luogo”, a discapito che, per cercare se una persona è responsabile di una rivista o conferenza, dovremmo andare a controllare singolarmente ogni rivista/conferenza.
* Per “Struttura organizzatrice” invece vogliamo che, cercando una certa struttura, si possa sapere quali conferenze ha organizzato quella struttura. Per fare ciò “Struttura organizzatrice” la andremo a definire come un’entità, aggiungendo poi una relazione tra “Struttura organizzatrice” e “Conferenza”
* Anche per “Editore” faremo la stessa cosa fatta per “Struttura organizzatrice”, così che si possa cercare a quali pubblicazioni corrispondano a quell’editore.

## **2.6 | Scelta degli identificatori primari.**

In questa fase andremo a scegliere quali attributi considerare degli identificatori primari, e se necessario li andremo ad aggiungere, consideriamo ogni entità prese singolarmente:

* Per “Libro” è stato scelto di utilizzare come identificatore primario l’attributo “ISBN”, è una sequenza numerica di 13 cifre usata convenzionalmente in tutto il mondo per la classificazione dei libri;
* Per “Articolo” è stato scelto di utilizzare come identificatore l’attributo “DOI”, è un codice alfanumerico assegnato a un oggetto digitale, come nel nostro schema, un articolo scientifico.
* Per “Rivista” ” è stato scelto di utilizzare come identificatore l’attributo “ISSN”, è una sequenza numerica internazionale che identifica le pubblicazioni in serie.
* Per “Collana” è stato scelto di inserire un attributo “ID\_collana”, che rappresenta il codice associato ad una collana.
* Per “Autore” è stato scelto di inserire un attributo “ID\_autore”, che rappresenta il codice associato ad un autore.
* Per “Editore” è stato scelto di inserire un attributo “ID\_editore”, che rappresenta il codice associato ad un editore.
* Per “Struttura organizzatrice” è stato scelto di inserire un attributo “ID\_struttura”, che rappresenta il codice associato ad una struttura.
* Per “Conferenza” è stato scelto di utilizzare una identificatore primario composto da più attributi, questi sono “Nome”, “Data inizio” e “Data fine”.

# Capitolo 3

# ***DIAGRAMMI RISTRUTTURATI***

## **3.1 | Diagramma UML (ristrutturato)**

Immagine che contiene testo, diagramma, Piano, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

## **3.2 | Diagramma ER (ristrutturato)**

# Capitolo 4

# ***PROGETTAZIONE LOGICA***

## **4.1 | Progettazione Logica**

In questa fase andremo a tradurre il nostro schema concettuale, in un modello logico che è lo step successivo. I passaggi fondamentali per il mapping sono:

* Codificare le entità
* Individuare la chiavi primarie
* Codificare le associazioni

## **4.2 | Mapping delle entità**

**LIBRO**(ISBN, Titolo, Data di pubblicazione, Formato, Sala/libreria di presentazione, Lingua, Categoria)

**AUTORI**(Nome, Cognome, ID\_autore)

**EDITORE**(Nome, Sede, ID\_editore)

**COLLANA**(Nome, ID\_collana, Descrizione)

**ARTICOLO**(DOI, Titolo, Data di pubblicazione, Formato)

**RIVISTA**(ISSN, Anno di pubblicazione, Nome, Argomento, Nome\_responsabile, Cognome\_responsabile)

**CONFERENZA**(Nome, Data inizio, Data fine, Nome\_responsabile, Cognome\_responsabile, Città, Stato)

**STRUTTURA ORGANIZZATRICE**(Nome, Sede, ID\_struttura)

## **4.3 | Mapping delle associazioni**

**SCRITTURA**(ISBN, ID\_autore)

**SCRITTURA. ISBN 🡪 LIBRO. ISBN**

**SCRITTURA. ID\_autore 🡪 AUTORI. ID\_autore**

**PUBBLICAZIONE**(ISBN, ID\_editore)

**PUBBLICAZIONE. ISBN 🡪 LIBRO. ISBN**

**PUBBLICAZIONE. ID\_editore 🡪 EDITORE. ID\_editore**

**APPARTENENZA**(ISBN, ID\_collana)

**APPARTENENZA. ISBN 🡪 LIBRO. ISBN**

**APPARTENENZA. ID\_collana 🡪 COLLANA. ID\_collana**

**SCRITTURA**(DOI, ID\_autore)

**SCRITTURA. DOI 🡪 ARTICOLO. DOI**

**SCRITTURA. ID\_autore 🡪 AUTORE. ID\_autore**

**PUBBLICAZIONE**(DOI, ID\_editore)

**PUBBLICAZIONE. DOI 🡪 ARTICOLO. DOI**

**PUBBLICAZIONE. ID\_editore 🡪 EDITORE. ID\_editore**

**PUBBLICATO**(DOI, ISSN)

**PUBBLICATO. DOI 🡪 ARTICOLO. DOI**

**PUBBLICATO. ISSN 🡪 RIVISTA. ISSN**

**PRESENTATO**(DOI, Nome, Data inizio, Data fine)

**PRESENTATO. DOI 🡪 ARTICOLO. DOI**

**PRESENTATO. Nome 🡪 CONFERENZA. Nome**

**PRESENTATO. Data inizio 🡪 CONFERENZA. Data inizio**

**PRESENTATO. Data fine 🡪 CONFERENZA. Data fine**

**ORGANIZZA**(ID\_struttura, Nome, Data inizio, Data fine)

**ORGANIZZA. ID\_struttura 🡪 STRUTTURA ORGANIZZATRICE. ID\_struttura**

**ORGANIZZA. Nome 🡪 CONFERENZA. Nome**

**ORGANIZZA. Data inizio 🡪 CONFERENZA. Data inizio**

**ORGANIZZA. Data fine 🡪 CONFERENZA. Data fine**