



Università degli Studi dell'Insubria  
Dipartimento di Scienze Teoriche e Applicate

---

## Laboratorio Interdisciplinare B

Specifiche di Progetto: Book Recommender  
a.a. 2023/2024

Dipartimento di Scienze Teoriche e Applicate

[loris.bozzato@uninsubria.it](mailto:loris.bozzato@uninsubria.it)

---

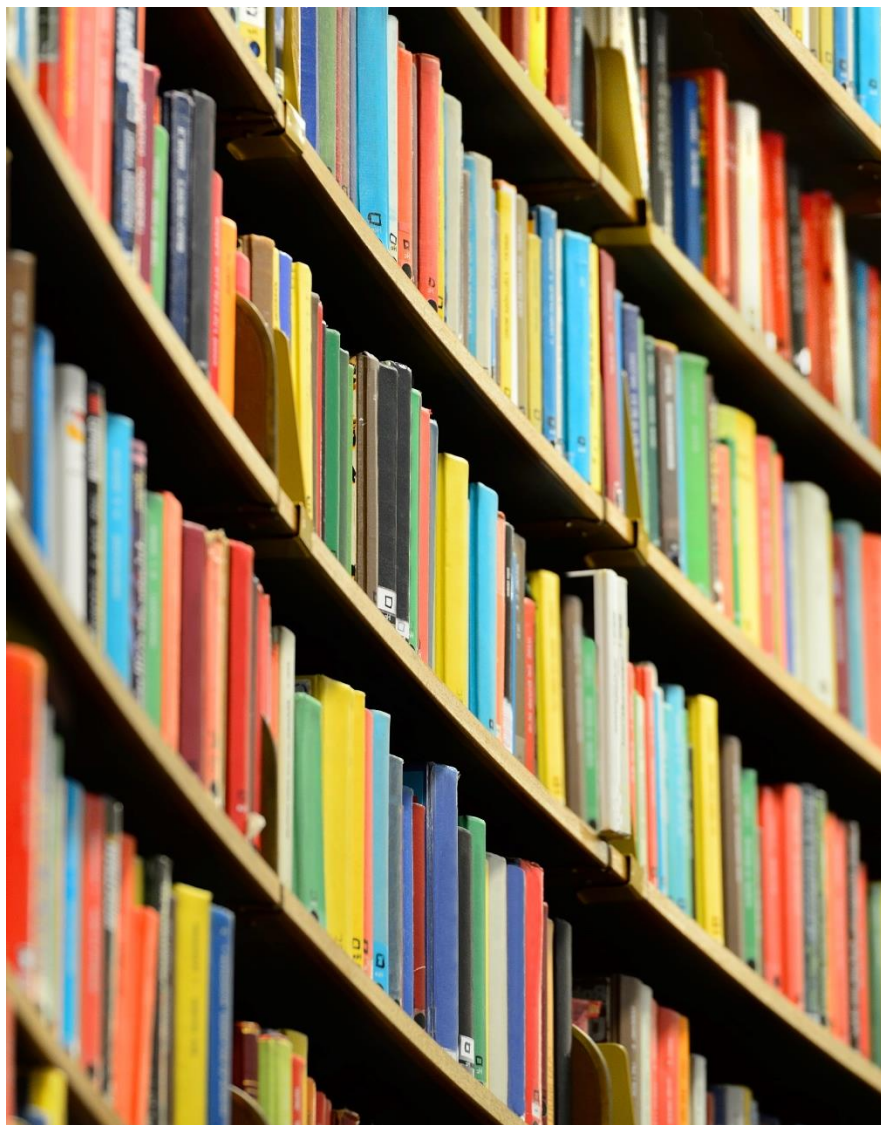


# Laboratorio Interdisciplinare - Cos'è:

---

- **Obiettivi:** mettere a fattor comune e in pratica le conoscenze acquisite nei corsi di Progettazione, Sviluppo Concorrente e Distribuito, e Basi Dati
- **Come:** progettando e sviluppando un progetto software reale

## Il Progetto – Book Recommender



- Un sistema per la valutazione e raccomandazione di libri, in grado di permettere agli utenti registrati di inserire recensioni e a tutti gli utenti di consultare le valutazioni e ricevere consigli di lettura.

# Il Progetto – Book Recommender

- I libri sono annotati rispetto a diversi criteri di valutazione:  
*indicano la valutazione del criterio su una scala che va da 1 (minimo) a 5 (massimo)*
- Recensione testuale opzionale (max 256 caratteri liberi)
- *Il valore di Voto Finale è la calcolato come la media (arrotondata) degli altri criteri*

Criterio Valutazione	Spiegazione	Score	Note (max 256 chars)
Stile	Valutazione sullo stile di scrittura del libro	1 ..... 5	
Contenuto	Valutazione sui contenuti del libro	1 ..... 5	
Gradevolezza	Valutazione sul gradimento nella lettura del libro	1 ..... 5	
Originalità	Valutazione sulla originalità del libro	1 ..... 5	
Edizione	Valutazione sulla qualità dell'edizione del libro	1 ..... 5	
<b>Voto Finale</b>	<b>Valutazione complessiva del libro</b>	<b>1 ..... 5</b>	

- Oltre a questo, gli utenti possono associare max. 3 libri suggeriti per il libro corrente.

# Il Progetto – Book Recommender

---

A partire da un repository di libri (da costruire), l'applicazione «Book Recommender» permette:

## **Utenti registrati**

- di creare una o più «librerie» personali per raccogliere i propri libri di interesse, aggiungere annotazioni ai singoli libri secondo i criteri dati nella tabella precedente e suggerire libri correlati (selezionati dalle proprie librerie)

## **Tutti gli utenti**

- di mostrare, per ciascun libro, un prospetto riassuntivo sulle valutazioni degli utenti in forma aggregata e i libri consigliati dagli utenti

# Il Progetto – Book Recommender

---

Prima di progettare l'applicazione è necessario costruire un Database con una tabella `Libri` dove memorizzare il repository dei libri, che deve contenere almeno i seguenti campi:

- Titolo
- Autori
- Anno di pubblicazione

Campi opzionali:

- Editore
- Categoria

**NOTA:** un file *draft* da cui è possibile estrarre dati bibliografici viene fornito dal docente e reso disponibile sulla pagina e-learning dell'insegnamento (***BooksDatasetClean.csv, disponibile da:*** <https://www.kaggle.com/datasets/elvinrustam/books-dataset/data>)

# Il Progetto – Book Recommender

---

L'applicazione permette di:

1. consultare le informazioni del repository dei libri (accesso libero)
2. registrarsi all'applicazione
3. creare più «librerie» (cioè raccolte di libri) personali (solo dopo login)
4. inserire le valutazioni dei libri inseriti nelle proprie «librerie», secondo i criteri forniti (solo utenti registrati, dopo login)
5. inserire consigli di libri (max 3) per i libri contenuti nella proprie librerie (solo utenti registrati, dopo login)

# Il Progetto – Book Recommender

---

1) Per consultare le informazioni di ogni libro (non è necessario login o registrazione):

a. funzionalità di ricerca `cercaLibro()`

- ricerca per titolo (prende in input una stringa di caratteri e restituisce i libri nel cui titolo compare la stringa di caratteri)
- ricerca per autore (prende in input un autore e restituisce i titoli dei libri in cui questo compare tra gli autori)
- ricerca per autore e anno (prende in input un autore e un anno e restituisce i titoli dei libri corrispondenti ad autore e anno ricercati)

(tutte le ricerche sono non *case sensitive* e funzionano con sotto-stringhe del testo cercato)

b. funzionalità di selezione e visualizzazione `visualizzaLibro()`

- una volta trovato il libro di interesse, deve essere possibile visualizzare tutte le informazioni relative al libro selezionato
- tra le informazioni deve esserci un prospetto riassuntivo delle valutazioni associate al libro (o l'indicazione che per il libro ricercato non sono disponibili valutazioni degli utenti) e i libri consigliati per il libro corrente (o l'indicazione che non ci sono consigli dagli utenti)
- Se presenti, si potranno visualizzare tali dati in forma aggregata (es. numero di utenti per valutazione e media dei punteggi per ogni criterio; per ogni libro consigliato, quanti utenti lo hanno suggerito), oltre agli eventuali commenti lasciati dagli utenti



# Il Progetto – Book Recommender

---

2) Per registrarsi all'applicazione, tramite la funzione `registrazione()`, l'utente deve inserire:

- nome e cognome
- codice fiscale
- indirizzo di posta elettronica
- userid
- password per accedere al sistema

I dati della registrazione devono essere salvati in una tabella del DB denominata `UtentiRegistrati`

# Il Progetto – Book Recommender

---

3) Per creare le proprie «librerie» (raccolte di libri) personali, l'utente registrato:

- a. deve autenticarsi tramite `userid` e `password` fornite al momento della registrazione
- b. tramite la funzione `registraLibreria()` deve inserire:
  - Nome della libreria
  - Elenco dei libri associati alla libreria

I dati di ogni libreria per ciascun utente registrato sono memorizzati in una tabella del DB denominata `Librerie`

# Il Progetto – Book Recommender

4) Per inserire valutazioni di libri, l'utente registrato:

- a. deve autenticarsi tramite userid e password fornite al momento della registrazione
- b. può ricercare e selezionare libri contenuti nelle proprie librerie personali
- c. può ora usare la funzione `inserisciValutazioneLibro()`

(libro e valutazioni associate dall'utente sono memorizzati nella tabella `ValutazioniLibri`)

Criterio Valutazione	Spiegazione	Score	Note (max 256 chars)
Stile	Valutazione sullo stile di scrittura del libro	1 ..... 5	
Contenuto	Valutazione sui contenuti del libro	1 ..... 5	
Gradevolezza	Valutazione sul gradimento nella lettura del libro	1 ..... 5	
Originalità	Valutazione sulla originalità del libro	1 ..... 5	
Edizione	Valutazione sulla qualità dell'edizione del libro	1 ..... 5	
<b>Voto Finale</b>	<b>Valutazione complessiva del libro</b>	<b>1 ..... 5</b>	

## Il Progetto – Book Recommender

---

5) Per inserire suggerimenti di libri, l'utente registrato:

- a. deve autenticarsi tramite `userid` e `password` fornite al momento della registrazione
- b. può ricercare e selezionare libri contenuti nelle proprie librerie personali
- c. può ora usare la funzione `inserisciSuggerimentoLibro()`

(libro e libri suggeriti associati dall'utente sono memorizzati nella tabella del DB `ConsigliLibri`)

# Il Progetto – Book Recommender

---

All'avvio l'applicazione mostra un menù iniziale dove:

Tutti possono:

- Cercare libri tramite titolo, autore, anno
- Visualizzare le valutazioni e consigli associati ad un libro selezionato
- Registrarsi all'applicazione

Gli utenti registrati possono:

- Effettuare il login all'applicazione
- Creare nuove librerie ed inserirvi libri
- Inserire valutazioni per un libro selezionato
- Inserire suggerimenti per un libro selezionato

## Il Progetto – Book Recommender

---

La **Piattaforma Book Recommender (BR)** consiste di:

1. un modulo `serverBR`, che interfacciandosi con un DBMS relazionale (PostgreSQL) fornisce servizi di back-end
2. un modulo `clientBR`, che fornisce tutti i servizi e funzionalità designate per gli utilizzatori dell'applicazione

Al lancio di `serverBR` deve essere richiesto di specificare:

1. le credenziali per accedere al `dbBR` (database di supporto all'esecuzione dei servizi della piattaforma BR)
2. l'host del DB

Una volta lanciato `serverBR`, questo dovrà rimanere in attesa di richieste di connessione da parte di client `clientBR`

L'applicazione BR dovrà essere realizzata in modo tale da supportare **l'interazione in parallelo con più utenti** connessi alla piattaforma da postazioni (client) differenti

# Il Progetto – Aspetti Chiave

---

- Strutture dati:
  - Libri
  - UtentiRegistrati
  - Librerie
  - ValutazioniLibri
  - ConsigliLibri
- Cosa e come memorizzare su DB (progettazione e realizzazione)
- Interfaccia utente (interfaccia grafica)
- Gestione della Concorrenza  
(i servizi dell'applicazione vengono erogati in parallelo a più utenti, e possono verificarsi accessi concorrenti a risorse condivise)

# Il Progetto – ToDo

---

1. Progettazione della Soluzione
2. Sviluppo della Soluzione Software
3. Documentazione di Progetto (2 documenti distinti)\*
  - Manuale Utente
  - Manuale Tecnico

\* Ulteriori informazioni su come scrivere la documentazione saranno trattate durante la prossime lezioni



# Il Progetto – ToDo

---

## 1. Progettazione della Soluzione

Le attività di analisi e progettazione devono essere adeguatamente documentate facendo uso del **linguaggio UML** per l'applicazione software e del modello **Entity-Relationship (ER)** per il database.

È richiesto di progettare l'applicazione avvalendosi dove possibile dell'uso di **design patterns**, e di realizzare l'applicazione con un'opportuna **interfaccia grafica**, usando il linguaggio Java e gli strumenti utili

Progettare e realizzare un database utilizzando **PostgreSQL** per la sua implementazione (<http://www.postgresql.org>) e **JDBC** per l'accesso alla base di dati da programma Java (<http://jdbc.postgresql.org>)

# Il Progetto – ToDo

---

## 2. Sviluppo della Soluzione – Database

- Si ristruttururi, se necessario, secondo le metodologie di progettazione i requisiti descritti. Si scelgano le metodologie per la costruzione dello schema ER, motivando le scelte fatte.
- Si definisca lo schema concettuale ER per il database, evidenziando le entità e le associazioni di interesse, nonché i vincoli di cardinalità e di identificazione, motivando le scelte effettuate. Altri eventuali vincoli devono essere espressi in linguaggio naturale.
- Si effettui la ristrutturazione dello schema ER motivando le scelte effettuate.
- E' richiesto di produrre un documento di analisi dei requisiti ristrutturato e documentazione associata allo schema ER (ristrutturato e non), con eventuale specifica di vincoli in linguaggio naturale.
- Si effettui la traduzione dello schema ER ristrutturato in un equivalente schema relazionale. E' richiesto di produrre la documentazione associata allo schema relazionale derivato dallo schema concettuale.
- Si realizzi il database utilizzando PostgreSQL, e SQL per la definizione dei dati, l'implementazione dei vincoli identificati, e la manipolazione dei dati, secondo le operazioni previste dall'applicazione.
- Documentare gli script SQL necessari alla creazione della base di dati e dei vincoli definiti sui dati e le query SQL a supporto dei servizi erogati da Interfacce di Programmazione.

# Il Progetto – Codice Sorgente

---

1. Il progetto deve essere sviluppato in linguaggio **Java** (versione recente, non oltre la 17) e deve essere multiplatforma
2. Il codice deve essere opportunamente commentato in formato **javadoc**
3. Il package `bookrecommender` deve essere definito e deve contenere le relative classi (ulteriori package sono ammessi)
4. Il `main` per l'esecuzione dell'applicazione deve essere contenuto nella classe di nome `BookRecommender` del package `bookrecommender`
5. L'intestazione di tutti i file `*.java` deve contenere *nome, cognome, num. matricola, sede (VA o CO)* degli autori del progetto

# Il Progetto – ToDo

---

## 3. Documentazione di Progetto

- (Seguirà lezione ad hoc su come scrivere un corretto documento utente e manuale tecnico)

# Il Progetto – Consegna

---

Il progetto deve essere consegnato **come link a una directory OneDrive dell'Università o Google Drive del project manager** e al link si dovrà trovare una dir compressa denominata con ***cognome\_matricola*** del **project manager del team**. La cartella compressa dovrà contenere:

- un file dal nome **autori.txt** contenente cognome, nome, numero di matricola e sede (VA o CO) di ogni membro del team
- la directory **doc** contenente il manuale utente, il manuale tecnico in formato .pdf, e tutti gli artefatti (diagrammi ER, UML) prodotti
- la directory **src** contenente il codice sorgente del progetto
- file di build Apache Ant (<http://ant.apache.org/>) o Maven (<https://maven.apache.org/>) per compilare il progetto, lanciare il server e i client, creare il database, creare la documentazione javadoc, etc.
- eventuali librerie necessarie alla compilazione e/o all'esecuzione
- file README con indicazioni precise sull'installazione e sulla compilazione, specificando i comandi Ant/Maven da utilizzare, ed indicazioni di particolari librerie, usate in modo non standard.

## Il Progetto – Consegna

---

3. La consegna viene fatta tramite invio link all'indirizzo [loris.bozzato@uninsubria.it](mailto:loris.bozzato@uninsubria.it) con oggetto email: **Progetto Lab B**
4. **Responsabile della consegna sarà il project manager tramite email istituzionale di ateneo**
5. Le date di consegna verranno comunicate di volta in volta (indicativamente una decina di giorni prima della data di appello d'esame)

# Il Progetto – Dubbi e domande

---

Per ogni dubbio sulle specifiche che non sia chiarito durante la lezione di presentazione del progetto, è possibile scrivere sul forum della pagina del corso e le varie domande saranno indirizzate durante la prossima attività laboratoriale.

Il **forum** si chiama "**Specifiche progetto lab B**"

# Il Progetto – Valutazione

---

- In fase di discussione orale verrà:
  - richiesto allo studente di saper argomentare in modo opportuno le scelte progettuali, algoritmiche, e implementative adottate
  - verificata l'effettiva padronanza delle tecniche utilizzate attraverso una serie di domande
- La valutazione terrà conto dei seguenti fattori:
  - l'aderenza del sistema realizzato ai requisiti proposti
  - la qualità dei documenti di analisi e progettazione prodotti sia per la realizzazione del software che per il database (correttezza sintattica, semantica, completezza e leggibilità, minimalità dello schema logico)
  - le scelte algoritmiche e di progettazione effettuate (design pattern)
  - la qualità del codice sorgente prodotto (funzionalità, correttezza, facilità d'uso).