

# ANALISI REQUISITI

## 1. Obiettivo del Progetto

L'obiettivo è la realizzazione di una piattaforma web basata su architettura **Client-Server** per la gestione di un catalogo librario digitale. Il sistema permette l'archiviazione temporanea, la visualizzazione dinamica e la manipolazione dei dati attraverso protocolli di comunicazione standard (API REST).

## 2. Requisiti Funzionali (RF)

Rappresentano le capacità operative e i servizi che l'applicazione deve fornire all'utente:

- **RF1 - Popolamento Automatico:** All'avvio, il sistema deve generare autonomamente un dataset di 20 record verosimili utilizzando la libreria *Faker*, includendo campi quali Titolo, Autore, Anno e Genere.
- **RF2 - Visualizzazione Dinamica:** L'interfaccia deve presentare una griglia o lista aggiornata in tempo reale con tutti i libri presenti nel database in-memory.
- **RF3 - Inserimento Controllato:** L'utente può aggiungere nuovi volumi tramite un form. Il sistema deve garantire che il genere sia scelto tra quelli predefiniti nel backend (Narrativa, Giallo, Fantasy, ecc.) e generare un ID univoco (UUID) lato server.
- **RF4 - Filtro Ricerca:** Implementazione di una barra di ricerca che permetta il filtraggio istantaneo dei risultati lato client in base all'autore o al genere.
- **RF5 - Rimozione Puntuale:** Possibilità di eliminare un singolo record selezionato tramite il relativo identificativo.
- **RF6 - Reset Totale:** Funzionalità di "svuotamento" completa che elimina simultaneamente tutti i record presenti nella libreria.

## 3. Requisiti Non Funzionali (RNF)

Definiscono i criteri di qualità e i vincoli operativi del software:

- **RNF1 - Usabilità e Accessibilità:** L'interfaccia deve essere centrata nel browser con un layout pulito. Per garantire la leggibilità, i campi di inserimento utilizzano un contrasto elevato (testo bianco su sfondo scuro #495057).
- **RNF2 - Affidabilità:** Tutte le operazioni di eliminazione (singola o totale) devono essere precedute da una finestra di dialogo di conferma per prevenire perdite di dati accidentali.
- **RNF3 - Performance:** Il caricamento dei dati iniziali e l'applicazione dei filtri di ricerca devono avvenire in tempi minimi (<100ms) grazie alla gestione dei dati in-memory e al filtraggio lato client.
- **RNF4 - Interoperabilità:** Il backend deve supportare correttamente le policy CORS per permettere la comunicazione sicura con il frontend React.

#### 4. Vincoli Tecnici

- **Backend:** Python 3.x con framework Flask.
- **Frontend:** React.js con gestione dello stato tramite Hooks (`useState`, `useEffect`).
- **Stile:** CSS3 con utilizzo di Flexbox per la centratura e design responsive.
- **Persistenza:** Archiviazione volatile (in-memory) durante l'esecuzione del processo server.