# Technical Architecture

## Descrizione

Il backend di **TaskFlow** è progettato per gestire in modo efficace la creazione e il monitoraggio di progetti, sviluppi e documenti. Segue un’architettura modulare, scalabile e orientata ai servizi, facilitando l’integrazione con altri sistemi e garantendo una gestione chiara dei dati.

## Struttura del Backend

L’applicazione è suddivisa in più livelli logici per garantire una chiara separazione delle responsabilità:

* **Gestione Progetti** → Permette agli utenti di creare, modificare e monitorare i progetti, centralizzando tutte le informazioni rilevanti.
* **Gestione Sviluppi** → Organizza le attività (task) all’interno dei progetti, facilitando la collaborazione tra i membri del team.
* **Gestione Documentale** → Consente il caricamento, l’archiviazione e l’analisi dei documenti per estrarre informazioni utili.
* **Generazione di Metadati** → Automatizza l’estrazione di dati strutturati dai documenti caricati.
* **Analisi Repository GitHub** → Esamina i repository esistenti per facilitare la generazione automatica di codice basata sulla documentazione e sugli asset presenti.
* **Commit Automatico su GitHub** → Permette di integrare le modifiche direttamente nei repository, migliorando la gestione del versioning.

## Dipendenze e Tecnologie Utilizzate

L’applicazione si basa su un ecosistema tecnologico moderno, garantendo efficienza e flessibilità:

* **Spring Boot 3.1.2** → Framework principale per lo sviluppo dell’applicativo.
* **Spring Boot Starter Web** → Per l’esposizione delle API REST.
* **Spring Boot Starter Data JPA** → Per l’integrazione con il database e la gestione dei dati.
* **Jakarta Validation API** → Per garantire l’integrità e la correttezza dei dati ricevuti.
* **H2 Database (runtime/test)** → Per testare l’applicazione in ambienti di sviluppo senza configurare un database esterno.
* **PostgreSQL (database di produzione)** → Per la memorizzazione stabile e scalabile dei dati.
* **Spring Boot Starter Test** → Per garantire la qualità del software attraverso test automatizzati.

## Principali Caratteristiche dell’Architettura

**Modularità** → Ogni componente è separato, garantendo un’elevata manutenibilità.  
**Scalabilità** → Supporta l’integrazione di nuove funzionalità senza impatti significativi sulle esistenti.  
**Sicurezza** → Implementa meccanismi di validazione e autenticazione per proteggere i dati.  
**Interoperabilità** → Le API REST permettono di comunicare facilmente con sistemi esterni come GitHub.