RITM0001 – Task Service  
Specifiche Tecniche

### Architettura del Servizio

Il servizio deve essere sviluppato seguendo un’architettura RESTful, garantendo la separazione tra la logica di business e la gestione della persistenza. L’interazione con il database deve essere astratta mediante un repository per facilitare la manutenzione e l’estensibilità del sistema.

### Struttura dell’Applicazione

L’applicazione sarà suddivisa nei seguenti livelli:

* Controller: Gestisce le richieste HTTP e interagisce con il livello di servizio.
* Service: Contiene la logica di business e gestisce l’interazione con il repository.
* Repository: Fornisce l’accesso ai dati memorizzati nel database.
* Model/Entity: Definisce la struttura dei dati delle attività.

### Modello Dati Task

Il modello dati per le attività deve includere i seguenti campi:

* ID (identificativo univoco)
* Titolo (descrizione breve dell’attività)
* Descrizione (dettagli aggiuntivi sull’attività)
* Stato (es. TODO, IN\_PROGRESS, COMPLETED)
* Data di creazione (timestamp della creazione dell’attività)
* Data di aggiornamento (timestamp dell’ultima modifica dell’attività)

### Persistenza dei Dati

* Il database relazionale memorizzerà le attività con l’uso di un ORM per la gestione delle operazioni CRUD.
* L’accesso ai dati deve essere ottimizzato con query efficienti per garantire performance elevate.
* Deve essere previsto un sistema di caching opzionale per ridurre il numero di query dirette al database

### Validazione e Sicurezza

* Validazione Input: Il sistema deve verificare che i dati forniti rispettino i requisiti minimi richiesti prima di essere processati.
* Gestione Errori: Devono essere previsti messaggi di errore chiari con codici di stato HTTP appropriati.
* Autenticazione e Autorizzazione: L’accesso alle API può essere protetto mediante autenticazione JWT per garantire la sicurezza dei dati.
* Logging e Monitoraggio: Il servizio deve includere un sistema di logging per tracciare le operazioni e facilitare il debugging.