

Gestione Esami universitari

Implementare un microservizio per la gestione degli esami universitari. Uno studente, identificato da un numero di matricola, sostiene un esame riguardante un corso (identificato da un id). Un corso è tenuto da un docente (identificato da un id univoco). A ogni corso è associato un numero chiamato Credito Formativo Universitario che ne determina la difficoltà.

Database

Entità:

- **Studente** -> matricola (PK), cognome, nome, data di nascita, data immatricolazione
- **Docente** -> id (PK), Cognome, Nome
- **Corso** -> id (PK), Nome insegnamento, CFU (Crediti Formativi Universitari), docente
- **Esame** -> matricola studente, id corso, data, voto, lode

Definire il modello dati su database seguendo le seguenti specifiche:

- Uno **studente** è identificato dal numero di matricola
- Un **corso** è tenuto da un solo **docente**
- Un **esame** lega uno **studente** e un **corso** che lo sostiene in una determinata data, con una votazione che va da 18 a 30, l'eventuale lode è un attributo aggiuntivo
- Un **esame** non può essere ripetuto
- Un **docente** può tenere più corsi

Entity Framework

Realizzare una libreria per l'accesso ai dati del modello precedentemente descritto utilizzando Entity Framework e il pattern Repository, definendo le relative relazioni tra le varie entità.

Web API

Implementare una WEB API che utilizzi la libreria precedentemente implementata e che esponga i seguenti metodi:

- Inserimento studente
- Inserimento docente
- Inserimento corso
- Registrazione esame
- Inquiry esami per studente
- Inquiry insegnamenti per docente
- Inquiry media voto per studente
- Inquiry media voto per esame
- Calcolare i CFU accumulati da uno studente

GraphQL

Implementare la funzionalità di Inquiry ordinabile e filtrabile per tutte le tabelle in GraphQL

Sicurezza

Aggiungere lo strato di sicurezza alle API implementate

Aggiungere pagina di login che chiede id docente e codice di accesso.

Ogni docente può avere più codici di accesso.

Aggiungere una nuova tabella per associare ad un utente i suoi codici di accesso.

Aggiungere funzionalità di logout.

Consentire chiamate alle API solo se il docente ha fatto il login.

Modalità

Eseguire commit e push sul repository dedicato (<https://github.com/corso-bdx/prova-finale-<NOME>>) dopo aver completato ciascun punto.