

# Esercizio Java: Quiz sui Pokemon

---

Realizzate un'applicazione Java che metta alla prova le conoscenze degli utenti tramite un quiz sui Pokemon.

## Home page

L'applicazione dispone di una vista home che permette di scegliere tra la modalità allenamento e quiz, eventuali altre funzionalità, per esempio 'scegli livello difficoltà', ricorda le risposte, giochi, ...

## Vista Quiz

L'applicazione dispone di una vista quiz. Ogni aggiornamento della pagina presenterà un nuovo quiz, ad esempio trovare la tipologia del pokemon, e fornirà tre opzioni di risposta, di cui solo una corretta. (a seconda del livello si possono proporre più scelte)

## Descrizione del Quiz

Il quiz conterrà una serie di domande riguardanti vari aspetti dei Pokemon, come la tipologia, i punti, l'evoluzione, l'immagine del pokemon, ecc.

## Vista allenamento

Questa vista mostra i dati uno per uno per favorire la memorizzazione: del tipo, dell'immagine, etc,

## Implementazione per progetto testuale

### (TUI per console)

1. Definire una classe **Domanda** che rappresenti una singola domanda del quiz. Ogni domanda avrà un testo, un'opzione corretta e due opzioni errate.
2. Creare una classe **Quiz** che gestisca la logica del quiz. Questa classe dovrebbe contenere un elenco di domande e gestire il caricamento delle domande da un file di dati o da una sorgente dati.
3. Implementare una classe **QuizRunner** che gestisca l'interfaccia utente. Questa classe presenterà le domande agli utenti e raccoglierà le loro risposte, fornendo feedback sull'accuratezza delle risposte.
4. Utilizzare un file di dati per memorizzare le domande del quiz. Il file potrebbe essere formattato in modo che ogni riga rappresenti una domanda con il testo della domanda e le tre opzioni di risposta, separate da un delimitatore.

## Funzionalità Aggiuntive

Per arricchire l'esperienza del quiz, potete considerare l'implementazione delle seguenti funzionalità:

- Aggiungere un timer per ogni domanda, dando agli utenti un limite di tempo per rispondere.
- Tenere traccia del punteggio dell'utente e visualizzare il punteggio finale alla fine del quiz.
- Consentire agli utenti di scegliere tra diversi livelli di difficoltà del quiz.
- Aggiungere domande bonus o domande a risposta multipla.
- Implementare una funzionalità di salvataggio dei punteggi migliori per tener traccia delle prestazioni degli utenti nel tempo.

- Implementare un gioco del memory con le immagini disponibili
- Implementare un'architettura a livelli, tiers: entity, repos, services, controllers.
- Implementare le viste con approccio REST e/o server-side MVC.
- Implementare l'app seguendo le best practice di programmazione orientata agli oggetti e utilizzando classi, metodi e attributi appropriati per garantire la chiarezza e la manutenibilità del codice.

Per semplificare il lavoro, si possono utilizzare i framework ai vari livelli.