

# Documentazione progetto TriviaBro

## Indice:

- Scopo del progetto:
- Realizzazione del progetto:
- Tecnologie utilizzate:

[Repository di GitHub](#)

---

## Scopo del progetto:

Lo scopo del progetto era quello di realizzare un software che implementasse l'utilizzo delle API, oppure di realizzarne una direttamente da zero.

Ho deciso quindi di realizzare un trivia game, ossia un gioco il cui scopo è rispondere correttamente al maggior numero di domande possibile. Il progetto mi ha permesso di apprendere come realizzare ed utilizzare una API.

## Realizzazione del progetto:

Il progetto si compone di tre software:

- API
- Gioco con interfaccia CLI
- Gioco con interfaccia GUI

### API:

La API è stata realizzata in python, utilizzando il framework "fastapi". Il suo scopo è quello di fornire le domande e le relative risposte ai client che le richiedono durante le partite e aggiornare al contempo uno storico dei punteggi di tutti i giocatori all'interno di un file. Sia le domande che la classifica sono memorizzate su dei file di testo codificati in formato JSON. Per avviare l'API è necessario usare il seguente comando nella directory del progetto:

```
uvicorn main:app --port 30333
```

L'API è accessibile attraverso la porta 30333 ed espone i seguenti endpoint:

- **getDomanda:** questo metodo restituisce una domanda con le relative risposte in formato json. Oltre a restituire queste informazioni viene

anche specificato il numero della domanda, che verrà poi utilizzato per verificarne la correttezza. Questo endpoint riceve come argomento una stringa contenente le domande già uscite, in modo da non inviare due volte la stessa domanda

**Request:**

```
http://localhost:30333/getDomanda?domandeGiaUscite=-1
```

**Response:**

```
{
  "domanda": "Qual e' il piu' grande stato dell'Australia?",
  "risposta1": "Nuovo Galles del Sud",
  "risposta2": "Victoria",
  "risposta3": "Queensland",
  "risposta4": "Australia Occidentale",
  "numeroDomanda": 19
}
```

- **verificaRisposta:** questo metodo verifica la correttezza di una domanda il cui numero è passato come argomento e restituisce il risultato. Oltre al numero della domanda alla richiesta viene passato come argomento anche l'indice della risposta

**Request:**

```
http://localhost:30333/verificaRisposta?numeroDomanda=0&numeroRisposta=3
```

**Response:**

```
{
  "corretta": "si"
}
```

- **getClassifica:** questo metodo restituisce un file JSON contenente la classifica. La classifica è salvata all'interno di un file testuale codificato in formato JSON presente all'interno della directory del progetto

**Request:**

```
http://localhost:30333/getClassifica
```

**Response:**

```
{
  "classifica": [
    {
      "username": "username",
      "punteggio": 10
    }
  ]
}
```

- **aggiornaClassifica:** questo metodo serve per aggiornare la classifica e accetta come argomenti il nome del giocatore e il punteggio effettuato. Il punteggio del giocatore viene aggiornato solo se questo risulta superiore ad un eventuale punteggio già registrato. Dopo aver aggiornato la classifica il metodo restituisce la leaderboard aggiornata

**Request:**

```
http://localhost:30333/aggiornaClassifica?username=simone&punteggio=1
```

**Response:**

```
{
  "classifica": [
    {
      "username": "username",
      "punteggio": 10
    }
  ]
}
```

- **reimpostaClassifica:** questo metodo cancella la classifica esistente e la restituisce all'utente in modo da confermare che l'operazione sia andata a buon fine

**Request:**

```
http://localhost:30333/reimpostaClassifica
```

**Response:**

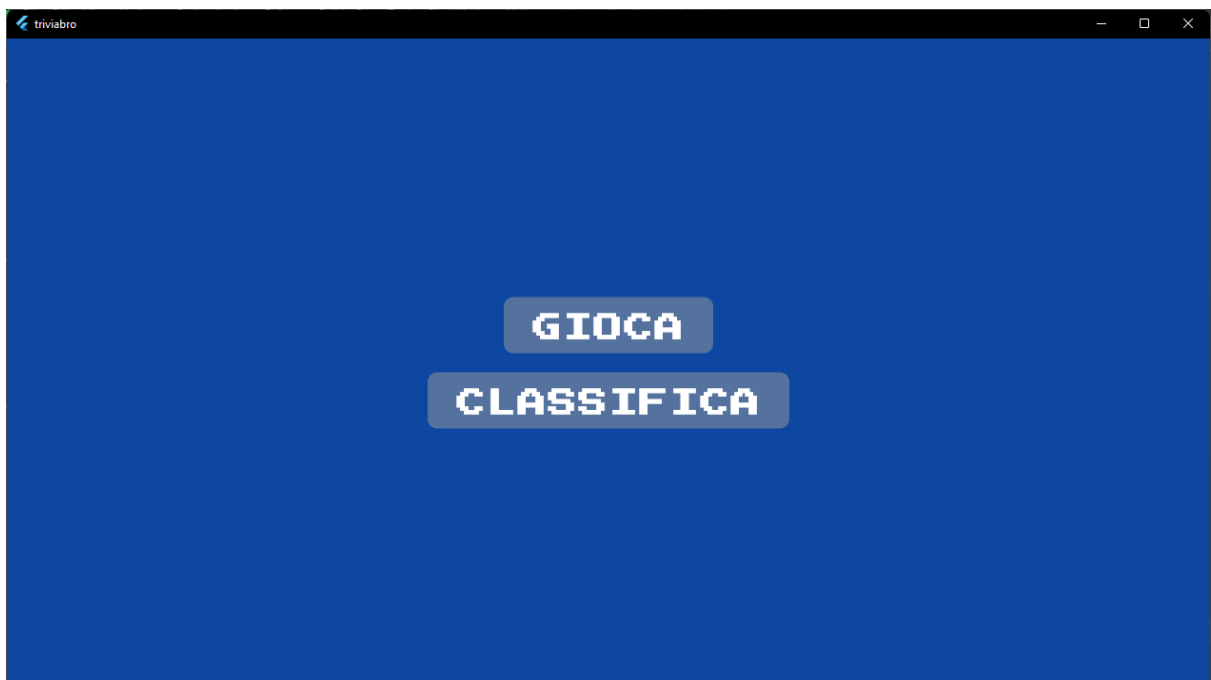
```
{  
  "classifica": []  
}
```

**GIOCO CLI:**

Questo software è un semplice script da riga di comando realizzato in python da cui si può giocare una partita a TriviaBro e visualizzare la classifica. In allegato alla consegna è presente il codice del gioco.

**GIOCO GUI:**

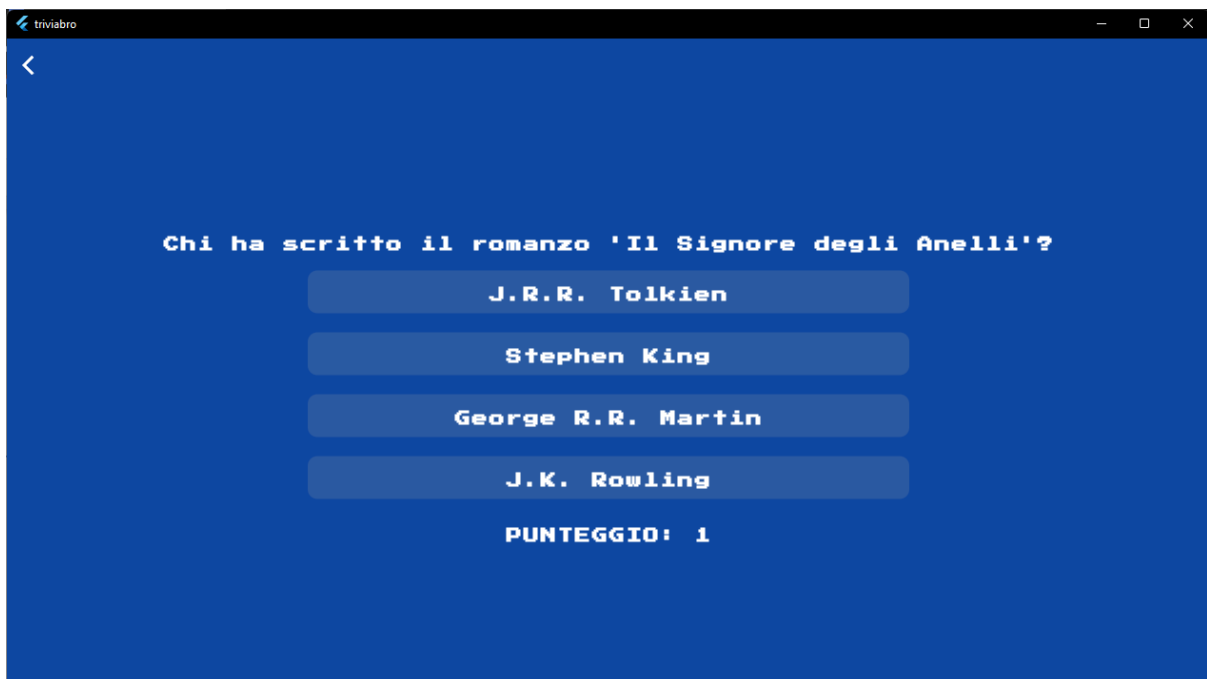
Questo software è un applicativo desktop realizzato in Flutter, da cui è possibile giocare e consultare la classifica. Di seguito gli screenshot del gioco:





The screenshot shows a web browser window with the title 'triviabro'. The page has a blue background and a white back arrow in the top left corner. The main content is a leaderboard titled 'CLASSIFICA' in white, bold, uppercase letters. Below the title is a table with two columns: 'Username' and 'Punteggio'. The table lists three players: 'Giocatore 1' with a score of 10, 'Giocatore 2' with a score of 7, and 'Giocatore 3' with a score of 2. The text is in a monospace font.

Username	Punteggio
1. Giocatore 1	10
2. Giocatore 2	7
3. Giocatore 3	2



The screenshot shows a web browser window with the title 'triviabro'. The page has a blue background and a white back arrow in the top left corner. The main content is a trivia question in white, monospace font: 'Chi ha scritto il romanzo 'Il Signore degli Anelli'?'. Below the question are four answer options, each in a light blue rounded rectangle: 'J.R.R. Tolkien', 'Stephen King', 'George R.R. Martin', and 'J.K. Rowling'. At the bottom, the score is displayed in white, monospace font: 'PUNTEGGIO: 1'.

Chi ha scritto il romanzo 'Il Signore degli Anelli'?

J.R.R. Tolkien

Stephen King

George R.R. Martin

J.K. Rowling

PUNTEGGIO: 1



## Tecnologie utilizzate:

Per il testing dell'API che ho realizzato in questo progetto ho utilizzato Postman. Grazie all'utilizzo di un workspace personalizzato sono stato in grado di realizzare dei template di richieste da utilizzare per semplificare il lavoro. Al posto di digitare ogni volta le richieste a mano ho infatti creato una serie di richieste già fatte che hanno velocizzato notevolmente il lavoro

