

Prova d'esame di Laboratorio - Sistemi e Applicazioni Cloud

Appello 15 giugno 2020 [Tempo consegna: 2h30]

Si realizzi un'applicazione web su Google Cloud Platform per categorizzare meme con un sistema di **tag**. L'applicazione permette agli utenti di taggare dei **meme** (immagini o video) con delle **keywords** e di effettuare ricerca per tags.

Si rendano disponibili i seguenti tipi di interfacce di utilizzo:

1. API per l'inserimento e la visualizzazione dei meme e dei tag;
2. Interfaccia HTML di visualizzazione della collezione.

L'applicazione deve essere testata per il deployment su piattaforma GCP utilizzando i servizi visti a lezione:

- App Engine
- Firestore
- Endpoints
- PubSub
- Functions

API REST [Sbarramento]

L'applicazione deve esporre le seguenti funzionalità tramite opportune Web API RESTful:

1. inviando richieste **POST** allo URI `/api/v1/meme/{meme-id}` è possibile inserire il link ad un meme e i tag associati ad esso;
2. inviando richieste **GET** allo URI specificato in precedenza è possibile ottenere le informazioni riguardanti il meme indicato.

L'interfaccia di utilizzo delle API deve soddisfare **rigorosamente** il file di specifica *OpenAPI* disponibile sul sito del corso¹.

Progettazione [Voto: C]

Definire le specifiche *OpenAPI* per l'implementazione di un'ulteriore API che consenta di ottenere la lista dei meme in base al tag specificato. L'API deve ritornare i primi 5 meme che contengono il tag in base alla **distanza di Levenshtein**, permettendo quindi una ricerca "resiliente" ad eventuali errori di battitura. Per implementare l'algoritmo di Levenshtein, controllare la documentazione della funzione `distance` del modulo `python-Levenshtein`. La nuova API dovrà essere quanto più integrata con quelle esistenti ed esporre un'interfaccia simile. Si richiede inoltre di implementare l'API secondo le specifiche definite.

Web Application [Voto: B]

Realizzare un'applicazione web che permetta di visualizzare la lista dei meme inseriti nel portale organizzandoli per tag. Cliccando sul titolo del meme si accederà ad una pagina di dettaglio dove sarà possibile visualizzare il meme.

Interconnessione di servizi [Voto: A]

Realizzare un sistema per l'aggiornamento dei tag basato su *Cloud PubSub* e *Cloud Functions*. Il sistema deve prevedere la possibilità di inviare comandi tramite *PubSub* specificando:

- il meme da modificare;
- l'azione;
- il tag.

L'azione prevista può essere di due tipi diversi:

- **DELETE**: rimuove il tag;
- **INSERT**: inserisce il tag.

¹https://weblab.ing.unimore.it/sac/1920/lab/esami/mymeme__openapi.yaml

Extra [Voto: S]

Esporre un backend API realizzato mediante *Cloud Functions* che permetta di accedere alle funzionalità del punto precedente tramite richieste REST.

Note

- Il mancato soddisfacimento **rigoroso** dei requisiti specificati nel file *OpenAPI* comporta l'invalidazione diretta della prova di laboratorio.
- A tale scopo, il soddisfacimento dei requisiti è testato con un correttore automatico disponibile all'indirizzo <http://sac-backend-validation.appspot.com>.