

# Sistemi e Applicazioni Cloud - Laboratorio

*Appello 14 giugno 2023* [Tempo consegna: 2h 30m]

Si realizzi un'applicazione su Google Cloud Platform per la gestione di tabelle di routing.

Si rendano disponibili i seguenti tipi di interfacce di utilizzo:

1. API per gestire le regole di routing
2. Pagina Web per visualizzare la tabella di routing e per testare l'instradamento

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

- App Engine
- Firestore

## Parte 1 - Backend API REST

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

1. inviando richieste **POST** allo URI `/api/v1/routing/{id}` è possibile inserire una nuova regola di routing
2. inviando richieste **GET** allo URI `/api/v1/routing/{id}` è possibile leggere una regola
3. inviando richieste **PUT** allo URI `/api/v1/routing/{id}` è possibile modificare una regola
4. inviando richieste **DELETE** allo URI `/api/v1/routing/{id}` è possibile cancellare una regola
5. inviando richieste **GET** allo URI `/api/v1/routing/` è possibile ottenere la lista delle regole nella tabella di routing (in ordine decrescente di lunghezza del prefisso))
6. inviando richieste **POST** allo URI `/api/v1/routing/` è possibile verificare il routing verso un indirizzo IP
7. inviando richieste **POST** allo URI `/api/v1/clean` è possibile azzerare la tabella di routing

L'interfaccia di utilizzo delle API deve soddisfare **rigorosamente** il file di specifica *OpenAPI* disponibile insieme alla presente specifica

Per testare la funzionalità dell'API è possibile far riferimento al seguente URL:

<https://sacvalidator.appspot.com/>

## Parte 2 - Web Application

Realizzare una pagina Web che mostra:

- il contenuto della tabella di routing
- un form per inserire un indirizzo IP

Se viene inviato il form, viene mostrata di nuovo la pagina, ma si evidenzia con il colore rosso la regola di routing che viene usata per il routing dell'indirizzo specificato. In caso di errori nei campi del form, viene invece mostrato un messaggio di errore

## Note

- La validazione dell'API legata all'URL `/api/v1/routing/{id}` e `/api/v1/clean` è necessaria per avere la sufficienza. In mancanza di questo il compito non sarà corretto.
- La API legata all'URL `/api/v1/routing/` viene validata automaticamente e la sua implementazione corretta contribuisce al punteggio.
- In caso di validazione positiva il voto sarà basato su quante parti dell'esame sono state portate a termine e sulla qualità del codice e delle soluzioni proposte.

## Dettagli implementativi

Per l'implementazione si ricorda che:

- si considera solo il caso di routing di indirizzi IPv4
- le regole di routing sono strutturate per avere i seguenti campi: *netID*, *netmask* (espressa in formato CIDR, anche detto *prefix length*), *next-hop router* (anche indicato come *gateway*), *device*
- la netmask e il netID devono essere in accordo, ovvero tutti i bit della parte hostID dell'indirizzo devono essere a 0. Per esempio `192.168.1.0/23` non va bene perché il bit 24 (che è parte dell'hostID vale 1)
- Per il routing le regole vanno controllate a partire da quelle con il prefisso più lungo
- La regola per il default gateway è indicata come `0.0.0.0/0`
- Per la manipolazione degli indirizzi IP in modo semplice (incluso il matching di un indirizzo IP con un netID) si può fare riferimento alla libreria `ipaddress` inclusa in Python.