



**Área Departamental de Engenharia de Electrónica e
Telecomunicações e de Computadores**

Trabalho de Computação na Nuvem

Autores:	47224	André Graça
	48459	Diogo Santos
	48321	André Esteves

Relatório para a Unidade Curricular de Computação na Nuvem da
Licenciatura em Engenharia Informática e de Computadores

Professor: José Simão

26 de Maio de 2024

Resumo

Esta aplicação foi desenvolvida com o objetivo de processar imagens para identificar características em inglês, assim como as suas traduções. Esta utiliza serviços integrados da Google Cloud Platform (GCP) para armazenamento, comunicação e computação, oferecendo uma solução flexível e escalável na nuvem.

O sistema permite que os clientes enviem arquivos de imagem para detecção de características, armazenando-os no Cloud Storage. Após a identificação das características, o sistema traduz as mesmas para português. Além disso, o sistema possui capacidade de adaptação a variações de carga, permitindo aumentar ou diminuir sua capacidade de processamento de imagens simultaneamente.

As funcionalidades do sistema são disponibilizadas por meio de uma interface gRPC, que permite às aplicações cliente interagirem com o sistema de forma eficiente. Essa interface oferece operações para submissão de imagens, obtenção de resultados de detecção de características e recuperação de informações sobre as fotos processadas.

Para garantir a disponibilidade e o load balancing do sistema, são utilizadas várias réplicas do servidor gRPC, executando-se em instâncias de máquinas virtuais (VMs) de um instance group. Além disso, réplicas da aplicação Label App, responsáveis pela identificação de características nas fotos, também são executadas em instâncias de VMs. A arquitetura do sistema utiliza serviços da GCP, como o Cloud Storage, Firestore, Pub/Sub, Compute Engine, Cloud Functions, Vision API e Translation API, para armazenamento, comunicação, computação, detecção de características e tradução de palavras, respetivamente.

Neste relatório, serão apresentados os detalhes de implementação da aplicação, bem como os fluxos de operações e a sua capacidade de elasticidade. Serão discutidos os serviços da GCP utilizados, as integrações entre eles e as APIs empregadas para realizar as funcionalidades desejadas. Por fim, serão abordados aspetos relevantes da implementação e as considerações necessárias para garantir o desempenho, a escalabilidade e a disponibilidade do sistema na nuvem.

Palavras-chave: Aplicação, Processar Imagens, Google Cloud Platform (GCP), Armazenamento, Comunicação, Submissão de Imagens, Interface gRPC, Servidor, Máquina Virtual (VM), Aplicativo LandMarks, Desempenho, Escalabilidade e Disponibilidade.

Abstract

This application was developed with the objective of processing images to identify characteristics in English, as well as their translations. It utilizes integrated services from the Google Cloud Platform (GCP) for storage, communication, and computation, offering a flexible and scalable cloud solution.

The system allows clients to submit image files for feature detection, storing them in Cloud Storage. After identifying the features, the system translates them into Portuguese. Additionally, the system can adapt to load variations, allowing it to scale its image processing capacity up or down simultaneously.

The system's functionalities are made available through a gRPC interface, enabling client applications to interact with the system efficiently. This interface offers operations for image submission, obtaining feature detection results, and retrieving information about processed photos.

To ensure system availability and load balancing, multiple replicas of the gRPC server are used, running on virtual machine (VM) instances in an instance group. Additionally, replicas of the Label App, responsible for identifying features in the photos, also run on VM instances. The system architecture leverages GCP services such as Cloud Storage, Firestore, Pub/Sub, Compute Engine, Cloud Functions, Vision API, and Translator API for storage, communication, computation, feature detection, and word translation, respectively.

This report presents the details of the application's implementation, including the operational workflows and its elasticity capabilities. It discusses the GCP services used, the integrations between them, and the APIs employed to achieve the desired functionalities. Finally, it addresses relevant aspects of the implementation and the considerations necessary to ensure the system's performance, scalability, and availability in the cloud.

Keywords: Application, Image Processing, Google Cloud Platform (GCP), Storage, Communication, Image Submission, gRPC Interface, Server, Virtual Machine (VM), LandMarks App, Performance, Scalability, Availability.

Índice

Conteúdo

Não foi encontrada nenhuma entrada de índice.

Lista de Figuras

1. Introdução

Esta aplicação foi desenvolvida com o objetivo de processar imagens para identificar monumentos ou locais famosos. Utiliza serviços da Google Cloud Platform (GCP) para armazenamento, comunicação e computação, proporcionando uma solução escalável na nuvem. Os clientes podem enviar imagens para detecção de monumentos, armazenando-as no Cloud Storage. O sistema é capaz de se adaptar a variações de carga, aumentando ou diminuindo a capacidade de processamento de imagens conforme necessário. As funcionalidades são disponibilizadas através de uma interface gRPC, permitindo aos clientes interagir com o sistema de forma eficiente. Réplicas do servidor gRPC e do aplicativo Landmarks App são executadas em máquinas virtuais para garantir disponibilidade e load balancing. O sistema utiliza serviços como Cloud Storage, Firestore, Pub/Sub, Compute Engine, Cloud Functions, Vision API e Static Maps API da GCP.

2. Estrutura da aplicação

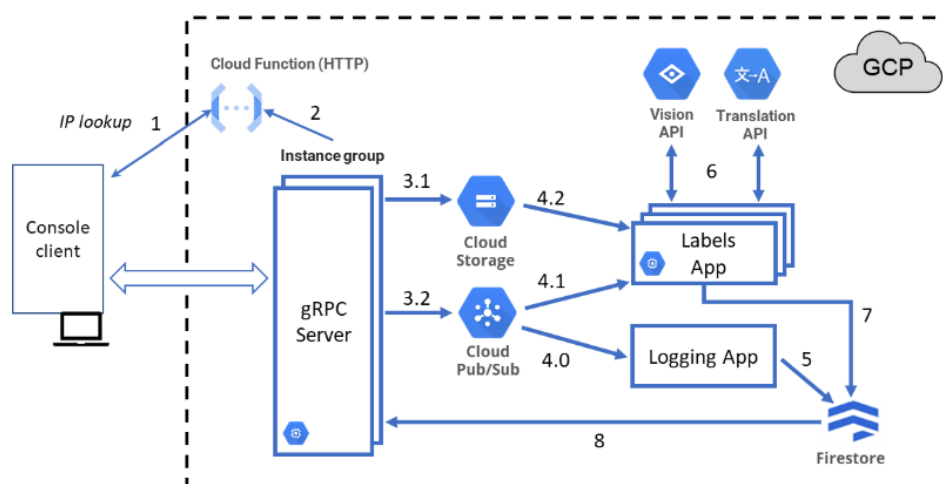


Figura 1 – Esquema da estrutura do trabalho

2.1 Cliente

O componente "client" é responsável por interagir com os usuários ou aplicações cliente, fornecendo uma interface intuitiva para submissão de imagens, obtenção de resultados e informações sobre as fotos processadas. Ele utiliza a interface gRPC para comunicar com o servidor.

2.1.1 IP Lookup

De modo a ter um IP estático para o qual o cliente pode contactar e usar o serviço, criámos uma Cloud Function que nos retorna, em formato JSON, a lista de IPS pertencente ao projeto e à Instance Group passado na query.

As opções disponíveis no menu incluem:

Ping Server: Permite ao utilizador obter a latência da sua ligação do servidor.

Submit Image: Permite ao utilizador enviar uma imagem para o serviço. O utilizador é solicitado a fornecer o caminho da imagem, e a imagem é lida e enviada para o servidor em blocos usando o StreamObserver do gRPC. O servidor retorna um identificador, constituído pelo bucket e nome do blob onde a imagem foi guardada no Google Cloud Storage.

Get Image Info: Recupera informações da imagem do servidor com base no identificador fornecido. O utilizador é solicitado a fornecer o identificador, e as informações da imagem correspondente são obtidas usando o StreamObserver do gRPC.

Get Image Names With Label: Recupera uma lista de nomes de imagens que contêm a característica especificada pelo utilizador. Os nomes das imagens correspondentes são recebidos usando o StreamObserver do gRPC.

Resize Server Instances: Permite mudar o número de instâncias de máquinas virtuais que correm o servidor.

Resize Label App Instances: Permite mudar o número de instâncias de máquinas virtuais que correm a aplicação de processamento de imagens.

2.2 Servidor

O componente "servidor" é a parte central do sistema, que recebe as solicitações do cliente através da interface gRPC e coordena o processamento das imagens. Ele interage com os serviços da Google Cloud Platform, como Cloud Storage, Firestore, Pub/Sub, Compute Engine, Cloud Functions, Vision API e Translator API, para armazenar as imagens, identificar características, armazenar informações relevantes. O servidor também lida com a distribuição de tarefas para as réplicas do aplicativo Label App e retorna os resultados ao cliente.

A divisão do projeto em partes bem definidas facilita o desenvolvimento, a manutenção e a compreensão do projeto. Cada componente possui responsabilidades específicas possibilita uma solução robusta e escalável na nuvem.

2.2.1 Google Cloud Storage

As imagens enviadas pela cliente são guardadas no Google Storage, as imagens são Blob que são guardados num Bucket específico.

2.2.2 Google Firebase

As informações da imagem, como por exemplo a localização geográfica, nome do monumento e referencia do mapa estático, estas informações são armazenadas no Google Firebase numa determinada Collection em Documents, posteriormente utilizada pelo server.

2.2.3 Google Pub/Sub

O Google Pub/Sub é onde estabelece ligação de servidor com Landmarks, o servidor envia uma mensagem que contem informação da localização da imagem no Google Storage. O landmarks recebe essa mesma mensagem e processa-o.

Label App

A aplicação Landmarks tem como objetivo detectar e processar informações sobre características em imagens. Ela utiliza várias APIs do Google, incluindo a Google Vision API, Google Translator API, Google Storage e Google Firebase.

2.2.4 Google Vision API

A Google Vision API é usada para detetar pontos de referência em imagens. O problema principal relacionado a essa API é realizar a detecção precisa dos landmarks nas imagens fornecidas. Garantir a detecção correta e precisa dos landmarks é essencial para o funcionamento adequado da aplicação.

2.2.5 Translation API

A Translation API é usada para traduzir cada característica de inglês para português para ser armazenada na Label da imagem correspondente.

2.2.6 Firestore API

A API do Firestore é usada para armazenar as imagens e as suas características traduzidas detetadas, incluindo os dados sobre o id do pedido, a data quando foi processada e o nome do ficheiro para cada imagem. É fundamental que os dados sejam armazenados e recuperados de forma confiável e eficiente, garantindo a consistência das informações para os usuários da aplicação.

2.3 Contratos Protobuf

Os contrato da interface gRPC, estabelecem as operações disponíveis e os parâmetros necessários para cada operação. Estes contratos asseguram a compatibilidade e interpretação dos dados e promovem uma comunicação consistente entre o cliente e o servidor, garantindo que ambos estejam em conformidade com o contrato estabelecido.

Para dividir as acesso aos sistemas de forma separada, separámos os seguintes 2 módulos diferentes:

- Sistema Funcional: Permite submissão de imagens para caracterização (com tradução) das mesmas e posterior pesquisa.

```
service FunctionalService {
    // ping server for testing service availability
    rpc IsAlive(google.protobuf.Empty) returns (google.protobuf.Empty);

    // uploads an Image by sending blocks of bytes to be labeled
    rpc SubmitImageForLabeling(stream ImageChunkRequest) returns
    (RequestId);

    // obtains an Image details by the given identifier
    rpc GetLabeledImageByRequestId(RequestId) returns
    (LabeledImageResponse);

    // obtains names of files between 2 dates and with a label
    rpc GetFileNamesWithLabel(FileNamesWithLabelRequest) returns
    (FileNamesWithLabelResponse);
}

message ImageChunkRequest {
    bytes chunk_data = 1;
    string file_name = 2;
    string content_type = 3;
}

message RequestId {string Id = 1;}

message LabelResponse {
    string value = 1;
    string translation = 2;
}

message LabeledImageResponse {
    RequestId request_id = 1;
    repeated LabelResponse labels = 2;
    string process_date = 3;
}

message FileNamesWithLabelRequest {
```

```

    string start_date = 1;
    string end_date = 2;
    string label = 3;
}

message FileNamesWithLabelResponse {
    repeated string file_names = 1;
}

```

- Sistema de Gestão: Permite aumentar ou diminuir o numero de instâncias de máquinas virtuais a servir o nosso servidor ou aplicação de deteção de imagens.

```

service ManagementService {
    // ping server for testing service availability
    rpc IsAlive(google.protobuf.Empty) returns (google.protobuf.Empty);

    // increase the number of instances of the gRPC server
    rpc ChangeGRPCServerInstances(InstanceCount) returns
(google.protobuf.Empty);

    // increase the number of instances of the image processing server
    rpc ChangeImageProcessingInstances(InstanceCount) returns
(google.protobuf.Empty);
}

message InstanceCount {
    int32 count = 1;
}

```