



**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), Label App, Desempenho, Escalabilidade e Disponibilidade.



# Índice

1. Introdução .....	6
2. Estrutura da aplicação .....	6
3. Cliente .....	7
3.1. IP Lookup .....	7
4. Servidor .....	8
4.1. Google Cloud Storage.....	8
4.2. Google Firebase.....	8
4.3. Google Pub/Sub .....	8
5. Label App.....	9

## Lista de Figuras

Figura 1 – Esquema da estrutura do trabalho .....	6
Figura 2 - Contrato Protobuf do serviço funcional .....	10
Figura 3 - Contrato Protobuf do serviço de gestão.....	11



# 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 da Label 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

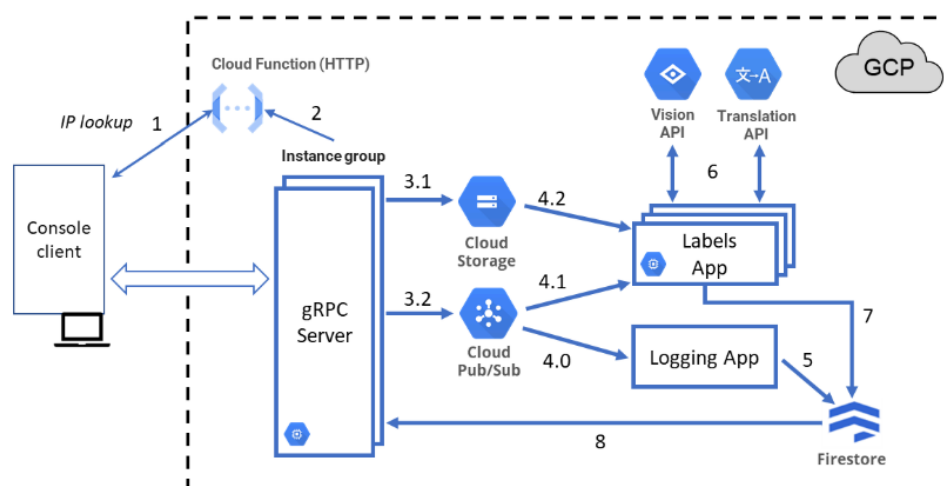


Figura 1 - Esquema da estrutura do trabalho



### 3. 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.

#### 3.1. 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. O cliente ao receber esta lista pode escolher a qual dos IPs se vai conectar.

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. Se o utilizador inserir 0, uma exceção é levantada para não permitir a um utilizador de deixar o serviço sem servidores.

**Resize Label App Instances:** Permite mudar o número de instâncias de máquinas virtuais que correm a aplicação de processamento de imagens. Se o utilizador inserir 0, não existirá nenhuma instância para processar imagens.

## 4. 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.

### 4.1. Google Cloud Storage

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

### 4.2. 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.

### 4.3. Google Pub/Sub

O Google Pub/Sub é onde acontece a troca desacoplada de mensagens entre o servidor, a Label e Logging App. No mesmo tópico onde as mensagens são publicadas pode haver várias subscrições, onde estas são retiradas.

## 5. Label App

A aplicação Label App 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.

### 5.1. 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 das características nas imagens fornecidas. Garantir a detecção correta e precisa das características é essencial para o funcionamento adequado da aplicação.

### 5.2. Translation API

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

### 5.3. 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.

## 6. Logging App

Esta aplicação é corrida apenas numa VM e foca-se simplesmente em tirar mensagens publicadas no tópico “pending-labels”, a partir da subscrição “logs-sub”, cumprindo assim a Fan-Out Pattern.

## 7. Contratos Protobuf

Os contratos 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. O contrato protobuf tem a seguinte descrição:

```
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;
}
```

Figura 2 - Contrato Protobuf do serviço funcional

Para enviar imagens de tamanho superior, enviamos estas em stream por chunks. Cada chunk tem 64KB da imagem. O servidor monta um array a partir da stream de bytes recebidos e reencaminha-o para o Google Cloud Storage para armazenamento.

- Sistema de Gestão: Permite aumentar ou diminuir o número de instâncias de máquinas virtuais a servir o nosso servidor ou aplicação de deteção de imagens. O contrato protobuf tem a seguinte descrição:

```
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;  
}
```

Figura 3 - Contrato Protobuf do serviço de gestão