

**INSTITUTO SUPERIOR de ENGENHARIA de LISBOA**  
Licenciatura em Engenharia Informática e Multimédia

2.º Semestre Letivo 2020/2021

**Computação na Nuvem Trabalho Final**



**Grupo nº: 14**

**Alunos:**

Miguel Silvestre nº 45101

Pedro Henriquesº 45415

Duarte Domingues nº 45140

## Índice

Objetivos .....	4
Introdução.....	4
Desenvolvimento .....	5
Instance groups.....	5
gRPC .....	5
Contrato .....	6
Pub/Sub.....	7
Carregamento de imagens na Google Cloud Storage .....	8
Detecção de características (labels) das imagens .....	9
Tradução das categorias .....	10
Organização dos dados na Google Firestore .....	11
Pedidos de informação sobre os ficheiros submetidos .....	12
Conclusões .....	13
Referências.....	14

## Índice de ilustrações

Figura 1 - Contrato .....	6
Figura 2 -Subscrição Pull.....	7
Figura 3 - Armazenamento de ficheiros no bucket.....	8
Figura 4 - Armazenamento de informação dum ficheiro na firestore .....	8
Figura 5 - Tópico topicworkers.....	9
Figura 6 – Arquitetura do processo de deteção de labels .....	9
Figura 7 - Arquitetura do processo de tradução.....	10
Figura 8 - Armazenamento da informação associada a uma imagem na firestore .....	11
Figura 9 - Arquitetura do processo de pedido de informação sobre os ficheiros .....	12

## Objetivos

Este trabalho prático tem como objetivos saber planejar e realizar um sistema para submissão e execução de tarefas de computação na nuvem, com requisitos de elasticidade, utilizando de forma integrada serviços da Google Cloud Platform para armazenamento, comunicação e computação, nomeadamente, Cloud Storage, Firestore, Pub/Sub, Compute Engine e Cloud Functions.

## Introdução

Neste trabalho prático irá ser desenvolvido um sistema de deteção de características, como por exemplo, *tree*, *street*, *night*, *fish* em ficheiros de imagem (PNG, JPC, etc) e traduzir estas características de inglês para outras línguas como o português.

As funcionalidades do sistema estão disponíveis para aplicações cliente através de uma interface gRPC. Em gRPC uma aplicação cliente consegue chamar diretamente um método numa aplicação server numa máquina diferente como se fosse um objeto local, gRPC facilita a criação de aplicações e serviços distribuídos.

Este trabalho prático vai usufruir de múltiplos serviços GCP (Google Cloud Platform), respetivamente, google Cloud Storage, Firestore, Pub/Sub, Compute Engine e o Cloud Functions.

- Google Cloud Storage – Serviço para armazenar e aceder objetos na Google Cloud em buckets (contentores de dados). No armazenamento na *cloud* os ficheiros são guardados como objetos imutáveis do tipo BLOB (*Binary Large Object*). A partir deste serviço vão ser armazenadas as diferentes imagens a ser processadas.
- Google Firestore – Serviço para armazenar dados NoSQL como coleções de documentos. No Firestore uma base dados é uma coleção de documentos, em que cada coleção dispõe dum conjunto de documentos, tendo cada documento um conjunto de campos diferentes de vários tipos, como por exemplo, inteiros, booleanos e arrays. Este serviço vai ser utilizado para guardar informação relacionada com o processamento de um ficheiro como por exemplo, nome do ficheiro, identificador, características detetadas nas imagens e as suas respetivas traduções.
- Google Pub Sub – Serviço real time de troca de mensagens que permite enviar e receber mensagens entre aplicações independentes. As mensagens são publicas em tópicos, tópicos representam categorias de informação para organizar mensagens. Este serviço vai ser utilizado para trocar mensagens entre as diferentes componentes do projeto.
- Google Compute Engine – Serviço de computação seguro e personalizável que permite criar e executar máquinas virtuais na infraestrutura do Google. Este serviço foi utilizado para alojar máquinas virtuais onde se executam os servidores de gRPC e as aplicações de deteção de características de imagens.
- Google Cloud Function – Serviço utilizado para implementar uma função de tradução de texto. Este serviço permite desenvolver uma lógica entre diferentes serviços GCP através da deteção e reposta a eventos.

## Desenvolvimento

### Instance groups

A partir do serviço Compute Engine vão ser criadas várias instâncias de máquinas virtuais do servidor gRPC e das aplicações de deteção e tradução de características de imagens. De forma que estes dois conjuntos de VM possam ter um número diferente de instâncias, os servidores gRPC e os *workers* são alocados em Instance Groups diferentes.

Os Instance Groups foram criados sem *autoscale*, este tipo de Instance Group tem a vantagem de através da consola podemos aumentar ou diminuir o número de instâncias e de reduzir a quantidade de “stress” inferida no sistema necessitando de menos utilização do CPU.

### gRPC

O gRPC baseia-se na ideia da definição dum serviço, especificando os métodos que podem ser chamados remotamente com os seus parâmetros e tipos de retorno. No lado do servidor, o servidor implementa uma interface e corre um servidor gRPC para lidar com chamadas do lado do cliente. Do lado do cliente, o cliente tem *stubs* que fornecem os mesmos métodos que o servidor.

O gRPC utiliza protocolos Buffers, um mecanismo de serialização de dados. Para trabalhar com Protocol Buffers é necessário definir a estrutura dos dados que vão ser serializados, para isto utiliza-se um proto file. Proto file é um ficheiro de texto com uma extensão do tipo (*.proto*).

Informação nos protocolos Buffers é estruturada por mensagens. Nos ficheiros proto são definidos serviços com parâmetros de métodos RPC e tipos de retorno especificados como mensagens.

## Contrato

O contrato utilizado define um serviço chamado “tpService” com um método “sendFileBlocks”. Este método é utilizado para chamar funções remotas que serão capazes de guardar os ficheiros na Google Cloud Storage.

O contrato tem também uma mensagem chamada “Content”, a mensagem “Content” armazena as informações relativas ao blob que vai ser enviado do cliente para o servidor, o cliente divide o ficheiro em blocos de bytes (*chunks*), enviando estes blocos em *stream* para o servidor. Desta forma reduzimos a largura de banda necessária para enviar uma mensagem, em vez de enviar o ficheiro inteiro numa vez, o ficheiro é enviado em *chunks*, portanto se ocorrer um erro no envio do ficheiro não é necessário repetir a transmissão de tantos bytes.

A mensagem “StoredContentID” é a resposta do servidor depois do ficheiro ser armazenado com sucesso, e corresponde ao identificador atribuído ao ficheiro da imagem na Google Cloud Storage.

O contrato foi elaborado de forma a ser o mais simples possível, a mensagem StoredContentID poderia ter campos adicionais, como por exemplo, as *labels* das imagens, porém, não é necessário pois os campos são obtidos no futuro com mensagens Pub/Sub.

```
syntax = "proto3";

// each class is defined in each own file, inside a common package
option java_multiple_files = true;
option java_package = "tpFinalService";

package tpFinalService; // package do proto

service tpService {
  rpc sendFileBlocks(stream Content) returns(StoredContentID);
}

message Void {}

message Content {
  bytes fileBlockBytes = 1; // bytes type is an array of bytes
  string filename = 2;
  string contentType = 3;
}

message StoredContentID {
  string identifier = 1;
}
```

Figura 1 - Contrato

## Pub/Sub

O serviço Pub/Sub foi utilizado neste trabalho prático para trocar mensagens entre diferentes componentes do sistema. Enviando mensagens pelo Pub/Sub em vez de RPC síncronos é possível assegurar desacoplamento entre produtores e consumidores das mensagens, com recurso a filas de mensagens, tornando o sistema mais robusto e simplificando o processo de desenvolvimento.

O Pub/Sub tem os seguintes conceitos principais:

- Tópico – Recurso do Google Cloud Platform definido por um nome para qual editores enviam mensagens.
- Mensagem – Composta por um campo de dados arbitrário, as mensagens são enviadas pelos editores para um tópico para serem posteriormente entregues aos subscritores.
- Subscrição – Recurso definido por nome, que representa a intenção de um assinante receber mensagens de um tópico específico. A subscrição tem associada a si uma fila de mensagens que se encontram retidas até serem consumidas.

O publisher cria e envia mensagens para um determinado tópico, os aplicativos interessados em receber estas mensagens criam uma subscrição para este tópico.

O modo da subscrição pode estar em push ou pull, sendo este o seguinte funcionamento:

- Subscrição Pull

1. O subscritor realiza um request de pull, pedindo as mensagens ao Pub/Sub.
2. O Pub/sub responde ao pedido pull com a mensagem.
3. Subcritor envia uma mensagem de acknowledge para avisar o Pub/Sub da receção e processamento da mensagem.

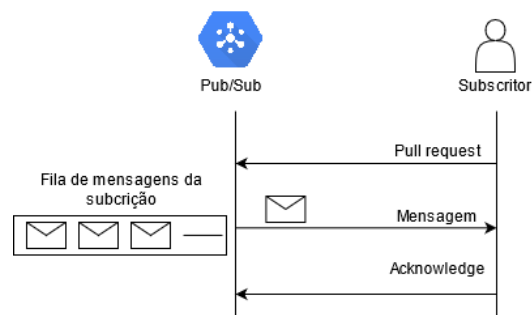


Figura 2 -Subscrição Pull

- Subscrição Push

- A subscrição envia mensagens push com uma solicitação HTTPs para um endpoint definido na subscrição para um pre-configurado.
- Endpoint confirma que a mensagem foi corretamente enviada com uma confirmação HTTP.

## Carregamento de imagens na Google Cloud Storage

Foi utilizado como repositório de imagens um *bucket* (`bucket_cn_tp_final`) usado Google Cloud Storage. Para efetuar o carregamento de uma imagem para este *bucket*, o cliente tem que fornecer o *path* para a imagem. Após estes passos serem completados e o pedido efetuado, a imagem é carregada para o *bucket* em formato de *blob*, o nome do *bucket* tem que ser único, pois os *buckets* têm um espaço de nomes universal.




<input type="checkbox"/>	Nome	Tamanho	Tipo
<input type="checkbox"/>	 DEC-VT180.jpg	32,9 KB	image/jpeg
<input type="checkbox"/>	 chuto-bola.jpg	36,2 KB	image/jpeg
<input type="checkbox"/>	 data-center-computer.jpeg	92,5 KB	image/jpeg

Figura 3 - Armazenamento de ficheiros no bucket

Informação sobre o processamento do ficheiro é guardada na Google Firestore, nomeadamente o identificador do ficheiro, nome do ficheiro, tipo do conteúdo e o *bucket* onde foi guardado o ficheiro.

isel_cn_tp-final\$DEC-VT180.jpg
<a href="#">+ INICIAR COLEÇÃO</a>
<a href="#">+ ADICIONAR CAMPO</a>
<code>contentType: "image/jpeg"</code>
<code>filename: "DEC-VT180.jpg"</code>
<code>identifier: "isel_cn_tp-final\$DEC-VT180.jpg"</code>

Figura 4 - Armazenamento de informação dum ficheiro na firestore

De seguida é enviada para um tópico Pub/Sub uma mensagem com o identificador do ficheiro, nome do ficheiro tipo de conteúdo e bucket onde foi guardado o ficheiro, para esta imagem ser processada e serem obtidas as características (labels) associadas à imagem e o identificador do pedido.



## Deteção de características (labels) das imagens

O tópico que foi chamado “topicworkers” tem associada uma subscrição cujo é consumida por vários *workers*. Um *worker* de análise de imagem recebe o nome do ficheiro duma imagem a partir do Pub/Sub e a partir deste nome obtém o conteúdo desta mesma imagem a partir do Google Cloud Storage.

De seguida à obtenção do conteúdo da imagem é realizada a deteção de características sobre a imagem. Para este fim utilizamos a Vision API, este serviço permite também obter para cada característica uma pontuação de confiança classificada entre 0 e 1.

Após a obtenção das diferentes labels da imagem é necessário enviar uma mensagem para um tópico Pub/Sub com as *labels* que se pretende traduzir e os valores de certeza associados e o identificador do pedido.

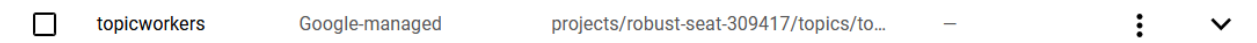


Figura 5 - Tópico topicworkers

Na figura abaixo pode-se observar o formato das diferentes mensagens Pub/Sub que vão ser enviadas do servidor gRPC para o vision worker.

```
===== MENSAGENS ENVIADAS ENTRE AS DIFERENTES PARTES DO PROJETO =====  
  
===== Server to PUB/SUB(Vision Worker) (depois da imagem ser armazenada na Cloud)  
  
Target topic = topicworkers  
Message(Id:<Random long> Data:<Img hash value>)  
Attributes : {targetBucket=<bucket onde foi guardada a imagem>,  
              identifier=<Img hash value>,  
              filename=<filename na máquina de destino>  
              contentType=image/jpeg}
```

O processo de deteção de *labels* das imagens segue a seguinte arquitetura.

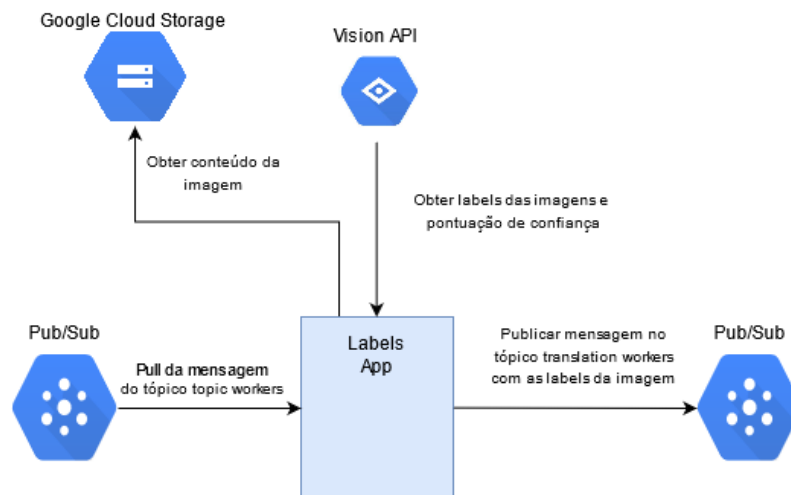


Figura 6 – Arquitetura do processo de deteção de labels

## Tradução das categorias

No processo de tradução das categorias (*labels*) das imagens utilizou-se uma API de tradução, chamada Google Cloud Translation. Esta API permite traduzir texto dinamicamente entre centenas de pares de linguagem. A partir da Google Translation é possível detetar a linguagem do texto automaticamente, mas preferiu-se fornecer a linguagem fonte de forma a obter melhores resultados, sendo que a tradução é sempre feita de inglês para outra linguagem.

Uma Cloud Function com a função de tradução recebe os pedidos Pub/Sub com as labels das imagens, de seguida a partir do serviço de tradução é realizada a tradução das diferentes labels.

Na figura abaixo pode-se observar o formato das diferentes mensagens Pub/Sub que vão ser enviadas do servidor gRPC para a Cloud Function (Translation Worker).

```
===== Server (Vision Worker) to Cloud Function (Translation Worker) (depois de serem obtidos os labels)
Target topic = labelTranslationworkers
Message(Id:<Random long> Data:<Img hash value>)
Attributes :    {label-<language>=[<Score do label>]} // 0 ou mais elementos na lista
```

Os resultados obtidos para as traduções são guardados na Firestore na coleção associada ao documento da imagem.

O processo de tradução de *labels* segue a seguinte arquitetura:

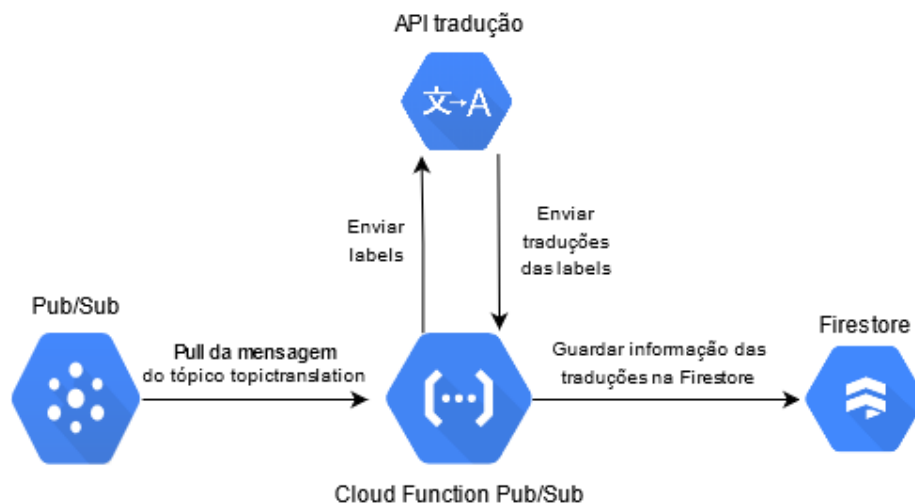


Figura 7 - Arquitetura do processo de tradução

## Organização dos dados na Google Firestore

Dentro da Firestore, cada imagem que é guardada corresponde a um documento com os meta-dados armazenados em dicionários chave-valor.

As *labels* traduzidas são guardadas numa lista e introduzidas num campo. Inicialmente criámos coleções individuais para as traduções de forma a organizar de forma concisa traduções de diferentes línguas. Porém, dificultava a pesquisa por valores implicando fazer o download de todos os documentos para uma máquina e pesquisar um a um pela tradução respetiva, isto não seria muito escalável pois iria aumentar significativamente a dificuldade computacional da pesquisa por *queries* a estes campos.

Raiz	final_cn_g14	isel_cn_tp-final\$data-center-computer.jpeg
+ INICIAR COLEÇÃO	+ ADICIONAR DOCUMENTO	+ INICIAR COLEÇÃO
<div><div>final_cn_g14</div><div>lab6_cn_g14</div></div>	<div><div>fzJ7aBJRblUSfs1SeNMM</div><div>isel_cn_tp-final\$DEC-VT180.jpg</div><div>isel_cn_tp-final\$data-center-computer.jpeg</div></div>	<div>+ ADICIONAR CAMPO</div> <div>contentType: "image/jpeg"</div> <div>filename: "data-center-computer.jpeg"</div> <div>identifier: "isel_cn_tp-final\$data-center-computer.jpeg"</div> <div>labels-pt<ul style="list-style-type: none"><li>0: "prédio"</li><li>1: "engenharia"</li><li>2: "computador portátil"</li><li>3: "computador pessoal"</li><li>4: "eletricidade"</li><li>5: "azul elétrico"</li><li>6: "computador"</li><li>7: "aparelho eletrónico"</li><li>8: "máquina"</li><li>9: "redes de computadores"</li></ul>targetBucket: "isel_cn_tp-final"</div>

Figura 8 - Armazenamento da informação associada a uma imagem na firestore

## Pedidos de informação sobre os ficheiros submetidos

As aplicações cliente têm a funcionalidade de poderem pedir informação ao servidor gRPC sobre os diversos ficheiros submetidos, nomeadamente as *labels* e as suas respetivas informações. De forma ao servidor gRPC ter acesso a essa informação é necessário consultar a informação associada às diferentes imagens na Firestore.

É necessário obter a coleção que contém os documentos com meta-dados dos blobs previamente enviados e depois utilizar *queries* da Google Firestore para obter a informação pretendida.

Foram criadas as seguintes operações de pesquisa.

1. Listagem de documentos com os meta-dados das imagens numa coleção.
2. Obter os campos todos associados a um documento.
3. Obter um array com as labels dum documento.
4. Ordenar os documentos segundo um critério para obter documentos semelhantes.
5. Obter uma lista de documentos que contêm determinado valor.

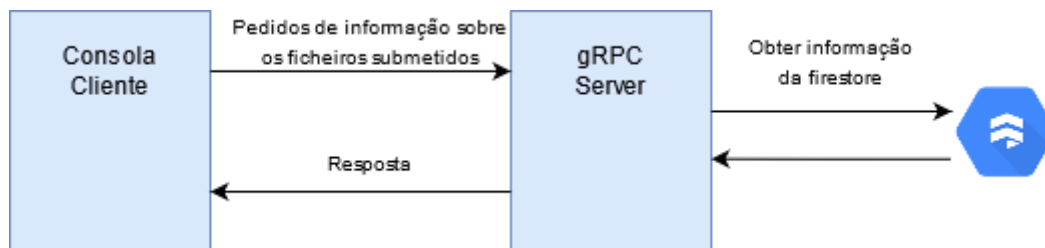


Figura 9 - Arquitetura do processo de pedido de informação sobre os ficheiros

## Conclusões

A realização do trabalho prático permitiu obter e consolidar conhecimento sobre a utilização de múltiplos serviços da Google Cloud Platform.

Neste trabalho prático foi planeado e realizado um sistema de submissão de tarefas, com elasticidade conferida pelos serviços da GCP.

Neste trabalho prático foram alcançados os resultados obtidos no que consta à categorização de imagens e traduções.

Neste trabalho prático lidamos com diversos problemas no processo de desenvolvimento, como por exemplo, problemas com permissões e configuração de scripts de start-up devido a falta de experiência no sistema operativo Linux e em plataformas na Cloud. No entanto os resultados obtidos foram os esperados, tendo seguido a arquitetura fornecida pelos professores.

## Referências

- [1] Slides de Computação na Nuvem Semestre de Verão 2020 – 2021, da autoria dos professores, José Simão e Luís Assunção.
- [2] Google Cloud Storage, Compute Engine and Insufficient Permission error - <https://stackoverflow.com/questions/47530481/google-cloud-storage-compute-engine-and-insufficient-permission-error>
- [3] Google Cloud Pub/Sub - <https://cloud.google.com/pubsub/docs/>
- [4] Introduction to gRPC - <https://grpc.io/docs/what-is-grpc/introduction/>
- [5] Cloud Firestore - <https://firebase.google.com/docs/firestore>