

Trabalho final

Data limite para entrega do trabalho: 2 de junho de 2023

Para além do desenvolvimento realizado que cada grupo realizará autonomamente fora das aulas, haverá aulas específicas, predominantemente nas aulas de 3 horas, para apoio à realização do trabalho.

Componentes a entregar:

- 1) Ficheiro ZIP com as componentes desenvolvidas, incluindo ficheiros README com informações sobre configurações, pressupostos de execução, teste ou outros.
- 2) Documento em formato PDF com descrição da solução: Diagramas de arquitetura, contratos (ex: *protobuf* entre cliente e servidor gRPC) e pressupostos entre as partes envolvidas, formatos de dados e mensagens envolvidos nas interações, bem como os aspectos relevantes da implementação e eventuais pontos de falha.

Objetivos: Saber planear e realizar um sistema para submissão e execução de tarefas de computação na nuvem, com capacidade de adaptação a variações de carga (elasticidade), utilizando de forma integrada serviços da Google Cloud Platform para armazenamento, comunicação e computação: Cloud Storage, Firestore, Pub/Sub, Compute Engine, Cloud Functions, Vision API e Static Maps API.

1. Introdução

Desenvolva um sistema, designado *CN2223TF*, com o objetivo de processar imagens (fotos) para verificar a existência de monumentos ou locais famosos (*landmarks*). Após a identificação do local, o sistema obtém uma imagem correspondente a um mapa da zona do monumento identificado. O sistema deve ainda ter elasticidade, aumentando ou diminuindo a sua capacidade de processamento de imagens em simultâneo.

As funcionalidades do sistema estão disponíveis para as aplicações cliente através de uma interface gRPC com as seguintes operações:

- Submissão de um ficheiro imagem (foto) para deteção de monumentos. Esta operação recebe o conteúdo de um ficheiro em *stream* de blocos, guardando o mesmo como um *blob* no serviço Cloud Storage. No final, a operação retorna um identificador do pedido (por exemplo, uma composição única entre o nome do *bucket* e do *blob*) que será usado posteriormente para obter o resultado da submissão;
- A partir de um identificador retornado na chamada à operação anterior deve ser possível obter:
 - a lista de resultados que inclui nomes de *landmarks* identificados na imagem, a localização geográfica (latitude e longitude em formato de graus decimais¹) e o respetivo nível de certeza associado à identificação (valor entre 0 e 1);

¹ https://pt.wikipedia.org/wiki/Graus_decimais

- a imagem com o mapa estático de uma das localizações identificadas;
- Obter os nomes de todas as fotos onde houve identificação de monumento com um grau de certeza acima de t (por exemplo, acima de 0,6) e o respetivo nome do local identificado.

Todas as operações de submissão e posterior consulta são disponibilizadas através de um servidor gRPC, o qual funciona como a fachada do sistema, isto é, a aplicação cliente não usa nem sabe que o servidor usa serviços da plataforma GCP. Para aumentar a disponibilidade e balanceamento de carga do sistema devem existir várias réplicas do servidor gRPC, cada uma a executar-se numa VM de um *instance group*. A arquitetura do sistema *CN2223TF* usa os seguintes serviços GCP e a API Static Maps da Google:

- O serviço Cloud Storage armazena as imagens a processar e os mapas estáticos;
- O serviço Firestore guarda a informação relevante sobre o processamento das imagens, nomeadamente o identificador do pedido, identificação dos *blobs* armazenados no *Storage*, os nomes e as localizações dos locais identificados nas imagens, e outros que achar convenientes;
- O serviço Pub/Sub é usado para troca desacoplada de mensagens entre os componentes do sistema;
- O serviço Compute Engine é usado para alojar as várias máquinas virtuais onde se executam as réplicas do servidor gRPC e as várias máquinas virtuais onde se executam as réplicas da aplicação (*Landmarks App*) para identificação de monumentos em fotos;
- O serviço Vision API para detetar monumentos (*landmarks*) nas imagens;
- API Static Maps para obter imagens de mapas numa determinada latitude e longitude.

As diferentes interações entre os componentes do sistema são apresentadas na Figura 1.

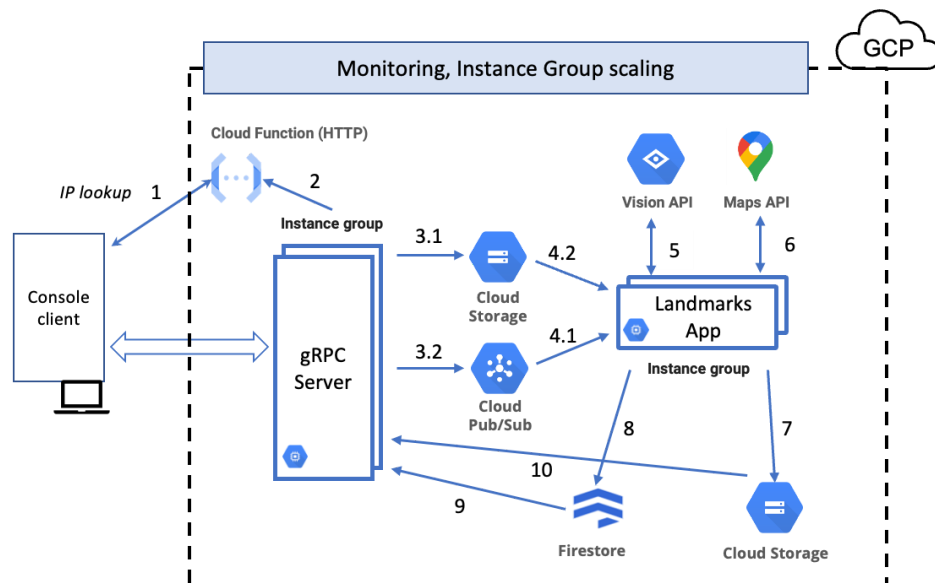


Figura 1: Componentes do *CN2223TF* e respetivas interações

2. Fluxo de operações

Tendo em conta os números de sequência de ações, apresentados na Figura 1, a lista seguinte descreve cada uma das funcionalidades:

- O serviço *Lookup Function*, usado pela aplicação cliente (1 e 2) para obtenção dos endereços IP dos servidores gRPC, deve ser desenvolvido como uma *Cloud Function* que obtém (2) os endereços IP das VM que fazem parte do *instance group*. A aplicação cliente escolhe um IP aleatoriamente da lista retornada pela função e, em caso de falha de ligação ao servidor gRPC através do IP escolhido, tenta outro IP ou repete o processo de *lookup* para atualizar a lista dos IP e estabelecer uma nova ligação;
- Após a submissão de uma imagem, a mesma é guardada no Cloud Storage (3.1) e é retornado ao cliente gRPC um identificador único para posteriormente ser possível realizar as interrogações. De seguida, é enviado para um tópico Pub/Sub (3.2) uma mensagem que contém o identificador do pedido, o nome do *bucket* e do *blob* para processamento de deteção de *landmarks*;
- Associado ao tópico referido anteriormente existe uma subscrição partilhada por vários *workers* (*work-queue pattern*). Um *worker* (*Landmarks App*) de análise de imagem recebe, em cada mensagem, o nome do *bucket* e do *blob* da imagem a processar (4.1) que permite obter uma referência global (URI *gs://*) do Cloud Storage (4.2), interagindo depois com o serviço Vision API (5) para identificação de locais e com o serviço de mapas (6) para obtenção de mapas estáticos;
- Após o processamento da foto, são guardados (7) no Cloud Storage os mapas dos locais identificados e guardada no Firestore (8) a informação relevante do pedido e do resultado da análise;
- A aplicação cliente, a qualquer momento, usando o descritor do pedido, pede ao servidor gRPC informações sobre as imagens submetidas, tal como descrito na Secção 1. Para retornar essa informação o servidor gRPC consulta o Firestore (9) e/ou o Storage (10).

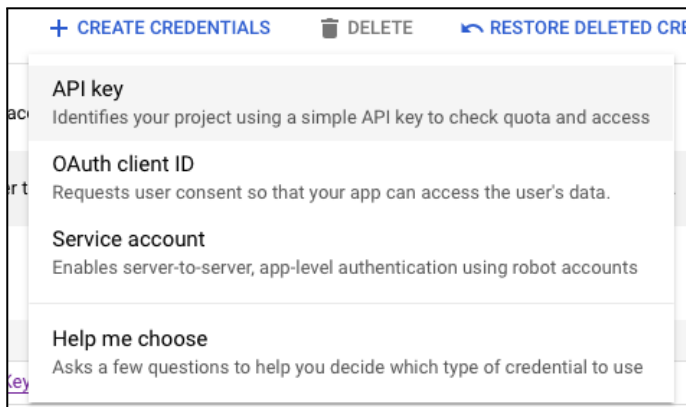
3. Elasticidade

Para que o sistema tenha maior disponibilidade e garantia de elasticidade devem ser considerados os seguintes requisitos:

- Deve existir um *instance group* com réplicas do servidor gRPC cuja dimensão pode aumentar (máximo 3) ou diminuir (mínimo 1);
- Deve existir um *instance group* com réplicas da aplicação *Landmarks App* (*worker*) para deteção de características (máximo 2) ou diminuir (mínimo 0);
- A dimensão dos *instance groups* pode ser gerida através da interface web do GCP ou através de aplicação específica para o efeito a correr no computador local.

4. Aspetos de implementação:

- A API de visão faz identificação de monumentos e locais de interesse em imagens, retornando, entre outras informações, uma *string* (nome do local), um indicador do nível de certeza da identificação (entre 0 e 1) e as coordenadas geográficas do local em formato de graus decimais. Mais informação e exemplos em Java podem ser consultados aqui: <https://cloud.google.com/vision/docs/detecting-landmarks>;
- A API de mapas estáticos está disponível através do URL <https://maps.googleapis.com/maps/api/staticmap> e parâmetros indicados na respectiva *query string*: <https://developers.google.com/maps/documentation/maps-static/start#map-parameter>. Dos diferentes parâmetros destaca-se o *center*, onde é indicado o local central do mapa (ex: center=38.756881,-9.116445), o *zoom*, o qual indica o nível de detalhe do mapa (ex: zoom=15) e o *size*, o qual indica a dimensão em pixels do mapa retornado (ex: size=600x300). Para utilizar a API é também necessária uma chave (API key) obtida através da consola web do GCP em <https://console.cloud.google.com/apis/credentials>. O parâmetro *key* indica a chave na chamada à API.



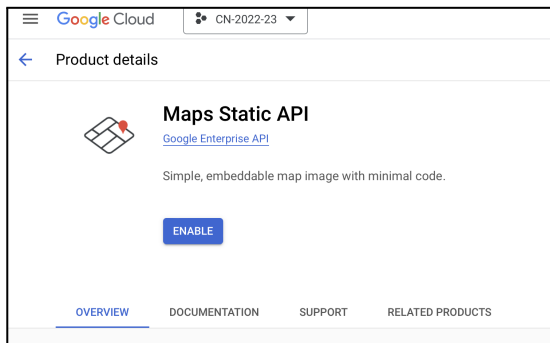
Com base nos parâmetros usados no exemplo anterior, a seguinte chamada com uma chave válida retornaria a imagem indicada:

<https://maps.googleapis.com/maps/api/staticmap?center=38.756881,-9.116445&zoom=15&size=600x300&key=<valor da API key>>



Antes de usar a API tem de se garantir que a mesma está ativa através deste *endpoint* no contexto do seu projeto:

<https://console.cloud.google.com/apis/library/static-maps-backend.googleapis.com>



Click em “Enable” devendo passar a apresentar o texto “Manage”.

Mais informações sobre os parâmetros e exemplos da API de mapas estáticos podem ser consultados aqui:

<https://developers.google.com/maps/documentation/maps-static/overview>

5. Critérios de avaliação do trabalho:

- ❖ 30% - Qualidade do relatório, que permita a um leitor entender claramente a arquitetura e as decisões de interação entre as partes, evitando apresentar código, exceto se o mesmo ajudar a explicar detalhes relevantes. O relatório deve indicar os pressupostos assumidos, indicando eventuais comparações com outras decisões possíveis. Deve constar no relatório qual a(s) parte(s) onde cada elemento do grupo de alunos teve mais ou menos responsabilidade.
- ❖ 60% - Operacionalidade, simplicidade e flexibilidade das soluções, nomeadamente na configuração e utilização da solução;
 - Nesta avaliação será ponderado o resultado da apresentação e demonstração da funcionalidade da solução a toda a turma nas aulas da última semana de aulas. Para tal, será posteriormente estabelecido para cada grupo um calendário de apresentação, bem como um guião dos aspetos principais a demonstrar.
- ❖ 10% - Participação individual de cada elemento do grupo durante as aulas afetas à realização do trabalho, bem como na apresentação do trabalho à turma.

José Simão, Luís Assunção, Fernanda Passos