Parte 1: Diseño del DAaaS

Definición la estrategia del DAaaS

Servicio de envíos periódicos de informes por email, con insights valiosos, para un club de karate, que permita la toma de decisiones sobre sus deportistas.

LOS OBJETIVOS DE LA PLATAFORMA:
* Facilitar la generación de informes a través de Power BI y su envío a los clientes objetivo.
* Establecer un entorno para la gestión y para el análisis de los datos de forma centralizada
* Mejorar la toma de decisiones de carácter deportivo y empresarial.

Para ello hay que determinar que personas dentro del club pueden acceder a cada tipo de información. Asegurar un acceso fácil y seguro a los informes relevantes para cada usuario tipo: Presidente del club, profesores y deportistas.

La solución debe ser escalable para adaptarse al crecimiento de datos y usuarios.

INPUT DE LOS DATOS:
* Base de datos SQL del club con los datos de los alumnos.
* Datos de ranking nacional obtenidos por un crawler en la página Karate scoring
* Resultados provinciales y regionales de la página del Federación Castellano Leonesa de Karate.

CATALOGO DE SERVICIOS:
* Definir procesos de carga desde las diferentes fuentes de los datos disponibles necesarios.
* Implementar flujos de trabajo para la limpieza y transformación de los datos, que garanticen la calidad y coherencia de los mismos para su posterior análisis.
* Datapedias: Desarrollar los recursos documentales necesarios para la comprensión de los datos.
* Identificar y ofrecer, si las necesidades del club lo requieren, otros servicios adicionales como análisis predictivos.

Arquitectura DaaaS

Hemos diseñado un sistema que aprovecha las diversas herramientas de *Google Cloud* y *Microsoft* para realizar el análisis de datos, generación de informes y visualizaciones. A continuación, detallamos los componentes clave y su funcionalidad:

- **GOOLGE CLOUD FUNCTIONS para la ejecución de la API GMAIL:** Utilizaremos Google Cloud Functions que ejecutará la API de Gmail cuando se actualice Google Storage, enviando de forma automática los correos electrónicos a cada cliente objetivo.
- **POWERBI y otras herramientas de análisis de datos:** Empleo de PowerrBI y de las herramientas necesarias para el análisis de datos y generación de visualizaciones, informes, etc, con el objetivo de conseguir una presentación efectiva de la información extraída.
- **GOOGLE CLOUD DATAPROC para ETL a través de un Google Function:** Haremos un uso racional de Google Cloud Dataproc, para realizar los trabajos de extracción, transformación y carga de forma programada. Después de cargar, limpiar y transformar los datos se generará un archivo CSV para su posterior uso en las visualizaciones de Power BI. Ejecutaremos automáticamente una Google Cloud Function cuando se actualizan los datos de entrada en Google Storage, con el objetivo de iniciar el proceso en el Dataproc, garantizando la sincronización de datos y el uso racional de este recurso.
- **GOOGLE CLOUD STORAGE para el almacenamiento:** Utilizaremos Google Cloud Storage para almacenar: datos de entrada, datos de salida y las funciones que puedan ser necesarias para el proceso, lo que nos proporcionará un almacenamiento escalable según las necesidades. Para ello se crearán Buckets independientes para los datos de entrada, salida, funciones, etc...
- **GOOGLE CLOUD FUNCTION para la actualización de la Base de Datos del Club:** Utilizaremos una Google Cloud Function programada para actualizar de forma periódica el archivo CSV de la base de datos del club, que guardará datos relevantes para su análisis, permitiendo tener siempre información actualizada para su análisis.
- **GOOGLE CLOUD FUNCTION que levante una Máquina Virtual en COMPUTE ENGINE:** Implementamos una Google Cloud Function que levante una MV en Compute Engine de Google Cloud, de forma programada ,para la ejecución de un Crawler que va a genera un CSV con información relevante de rankings desde la página **https://karatescoring.com/**.

Con esta arquitectura DAaaS se integran las herramientas necesarias, usando las soluciones que nos aporta Google Cloud y Microsoft, automatizando procesos que faciliten el análisis de datos.

DAaaS Operating Model Design and Rollout

A continuación detallamos el flujo de operaciones para el despliegue:

- **Carga archivos de Excel en Google Storage:** Un administrador será el responsable de cargar los archivos con los resultados de las competiciones en formato .XLS en Google Storage. Estos archivos son proporcionados por las federaciones autonómicas y la federación nacional a los clubes en este formato.
- **Ejecución mensual de Google Function para la actualización de rankings:** Mensualmente se ejecutará una Google Function que levantará una MV en Compute Engine, donde se ejecutará un Crawler que actualizará el archivo CSV de los rankings, manteniendo así siempre esta información actualizada.
- **Ejecución mensual de Google Function para la actualización de base de datos del club:** Mensualmente se ejecutará una Google Function que conectará con la base de datos SQL del club para actualizar el archivo CSV de los datos relevantes de los alumnos, para mantener esta formación actualizada.
- **Ejecución de Dataproc a través de una Google Function:** Una vez actualizados los datos anteriores se ejecutará a través de una Google Function programada, que levantará un entorno en Dataproc con los jobs necesarios para procesar dichos datos. Como resultado final se generará un nuevo archivo CSV que se almacenará en Google Storage.
- **Análisis de datos a través de PowerBI y otras herramientas si fuera necesario:** Un analista de datos utilizará PowerBI para las visualizaciones, y otras herramientas que se consideren útiles, para la presentación de insights de calidad al cliente. PowerBI actualizará de forma programada el CSV previo.
- **Carga de informes en Google Storage:** Los informes generados en el punto previo serán cargados por el analista de datos en Google Storage para su posterior distribución y almacenamiento.
- **Envío de correos electrónicos a los clientes finales a través a API Gmail:** Una vez cargados los archivos finales con los informes, se ejecutará una Google Function para conectar la API de Gmail y enviar los correos electrónicos necesarios a los clientes, asegurando una comunicación efectiva y de calidad.

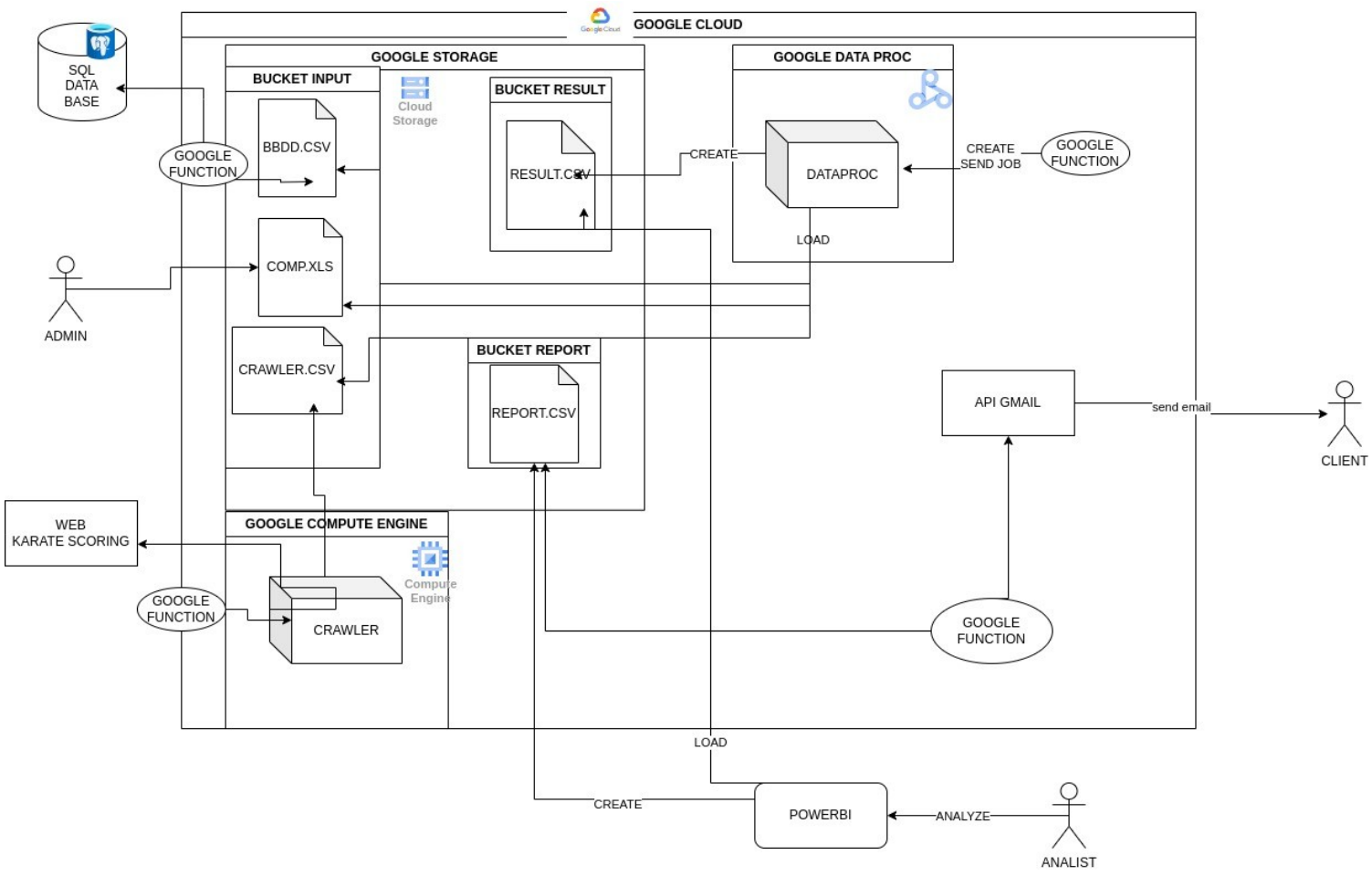
Con este modelo operativo estructurado que combina la automatización de tareas y la supervisión de otras, pretendemos garantizar la coherencia y puntualidad en la entrega de la información clave, optimizando el proceso de análisis de datos.

Desarrollo de la plataforma DAaaS. (ligera descripción del desarrollo)

Crawler para descargar los datos de ranking nacional senior masculino y femenino de kata y kumite.

https://colab.research.google.com/drive/12_KeV4YOnhDpJBXpnsVL-aA-GqVkdepJ?usp=sharing

DIAGRAMA



Parte 2 [opcional]

Crear un scraper en Google Collaboratory a partir de un API o de un crawler con scrapy, que descargue los datos a un archivo de formato estructurado

Adjunto enlace de un crawler para extraer datos de la página las categorías senior https://karatescoring.com/RankingRFEK/ tanto en kata como kumite. Hemos obviado el resto de rankings y categorías para un mejor manejo en la práctica y no aumentar los tiempos de ejecución, ya que son muchas las categorías y rankings los que están disponibles.

https://colab.research.google.com/drive/12_KeV4YOnhDpJBXpnsVL-aA-GqVkdepJ

Parte 3 [opcional]

Utilizar un proveedor de Cloud para montar un cluster de al menos 3 contenedores configurados correctamente o hacerlo en el cluster local

Configuración de un clúster en Google Cloud con Debian 11 utilizando Compute Engine:

- **Configuración del clúster:**

  Levantamos un clúster en Google Cloud en COMPUTER ENGINE, con un nodo master y 2 workers. Ubicamos nuestro clúster dentro de una región Europe (ejemp: Europe North1).

- **Elección del sistema operativo:**

  Optamos por utilizar Debian 11 como sistema operativo para todos los nodos del clúster.

- **Configuración de recursos:**

  Ajustamos los recursos necesarios de CPU y Memoria según las necesidades de cada proyecto, Para este proyecto decidimos usar instancias de tipo E2 Standard de 8G de memoria RAM y 500G de capacidad en un disco duro standard para todos los nodos.

- **Programación de borrado automático:**

  Implementamos una programación de borrado automático para evitar posibles costos excesivos.

Parte 4 [opcional]

Subir los archivos extraídos durante la parte 2 al cluster de Hadoop e insertarlos en el HDFS. Indicar pasos necesarios para realizar esto, dependiendo de la opción elegida en el Sprint 3.

Realizar la tarea de procesamiento de datos sobre los datos extraídos utilizando WordCount.

1. **Creación de reglas de Firewall:** Crear una regla de Firewall para abrir los puertos 8088 para acceder al YARN, y el puerto 9870 para acceder a HDFS. Poner los rangos de IP que pueden acceder en nuestro caso la IP de nuestro dispositivo y la IP pública del nodo maestro.

2. **Comprobación de conexión:** Podemos comprobar que accedemos correctamente al clúster del apartado 3 de l práctica accediendo desde la IP pública de nuestro nodo master y el puerto del HDFS y del YARN: *34.88.84.20:9870 34.88.84.20:8088*

3. **Carga del archivo a Google Storage:** Cargamos nuestro archivo karate.csv a Google Storage dentro de un bucket específico.

4. **Implementación del Job que ejecute wordcount:** Creamos un Job para el clúster dentro de la misma zona, con el archivo file:////usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar. Definimos los argumentos: wordcount, la dirección del archivo en el bucket y la dirección donde queremos guardar el archivo resultante (output/results si queremos guardarlo en hdfs o la dirección del Bucket de destino gs://bucketarquitectura13/karate/results/karate_result0224)

Parte 5 [opcional]

Utilizar HIVE/Elastic/Kafka/Mongo para insertar los datos extraídos durante el Sprint 2 y realizar operaciones con los mismos.

Indicar los pasos y las decisiones de diseño respecto a cómo organizar los datos.

El siguiente enfoque nos permite almacenar datos flexibles en MongoDB, ya que podemos agregar información adicional a los competidores posteriormente sin restricciones de esquema.

1. **Provisionar una Máquina Virtual en Compute Engine:**

   Utilizamos Google Cloud para levantar una MV en Compute Engine.

   Abrimos el terminal (SSH) de la MV y ejecutamos los siguientes comandos para instalar y configurar MongoDB:

```
sudo apt update
sudo apt install -y mongodb
sudo sed -i 's/^bind_ip = .*/bind_ip = 0.0.0.0/' /etc/mongodb.conf &&
sudo systemctl restart mongodb
sudo systemctl status mongodb
```

2. **Configuración de las reglas de Firewall:**

   Creamos una nueva regla de Firewall para el puerto 27017 de MongoDB.

   Especificamos las IP a las que queramos permitir el acceso como IP pública de nuestra máquina, la IP de la MV y la IP de Google Colab.

3. I**nstalación y conexión Mongo Compass:**

   Descargamos e instalamos MongoDB Compass, interfaz gráfica para MongoDB.

   Conectamos a MongoDB utilizando la siguiente URL:

   mongodb://IP_Publica_MV:27017/?directConnection=true&tls=false

4. **Creación de una nueva Base de Datos y las Colecciones necesarias:**

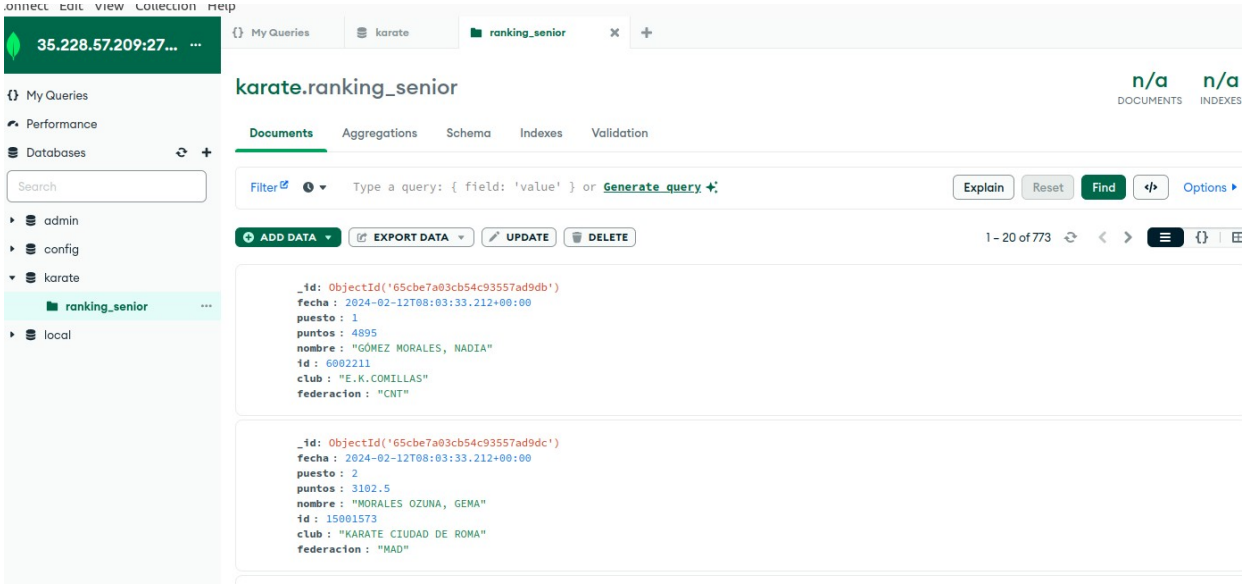
   Creamos una nueva Base de Datos llamada *“karate”*.

Dentro de la base de datos, creamos una nueva colección *“ranking_senior”.*

5. **Importación de Datos desde CSV:**

   Abrimos el archivo CSV previamente para editar las cabeceras y delimitar los strings con dobles comillas.

   Importamos los datos editados a la colección *“ranking_senior”.*



6. **Comprobación de conexión a través de Google Colab:**

   Creamos un notebook de Google colab con los scripts necesarios apra la comprobación de la conexión y realizar algunas queries:

https://colab.research.google.com/drive/1IMnxrNdfybYdbTO-02_dO8Q_T7BS06TU?usp=sharing