**Trabajo final**

**Marlon Gutiérrez Gallego**

**Marlon Andrés Vallejo Moreno**

**Omar García**

**Ingeniería en Sistemas, Universidad Católica Luis Amigó**

**Cloud Computing**

**Joham Álvarez Montoya**

**Medellín, Colombia**

**2024**

**Descripción del problema**

Actualmente, el constante desarrollo tecnológico en el área de realidad aumentada y realidad virtual ha creado una necesidad de datos telemétricos dado que se requiere una integración precisa con el entorno físico y virtual del usuario. Puesto que, para el correcto funcionamiento de las aplicaciones es necesario capturar y procesar en tiempo real datos precisos del usuario con su entorno, teniendo en cuenta, los objetos del espacio tridimensional en el que se encuentra el usuario, además, de la orientación y los movimientos mientras interactúa con lo que lo rodea.

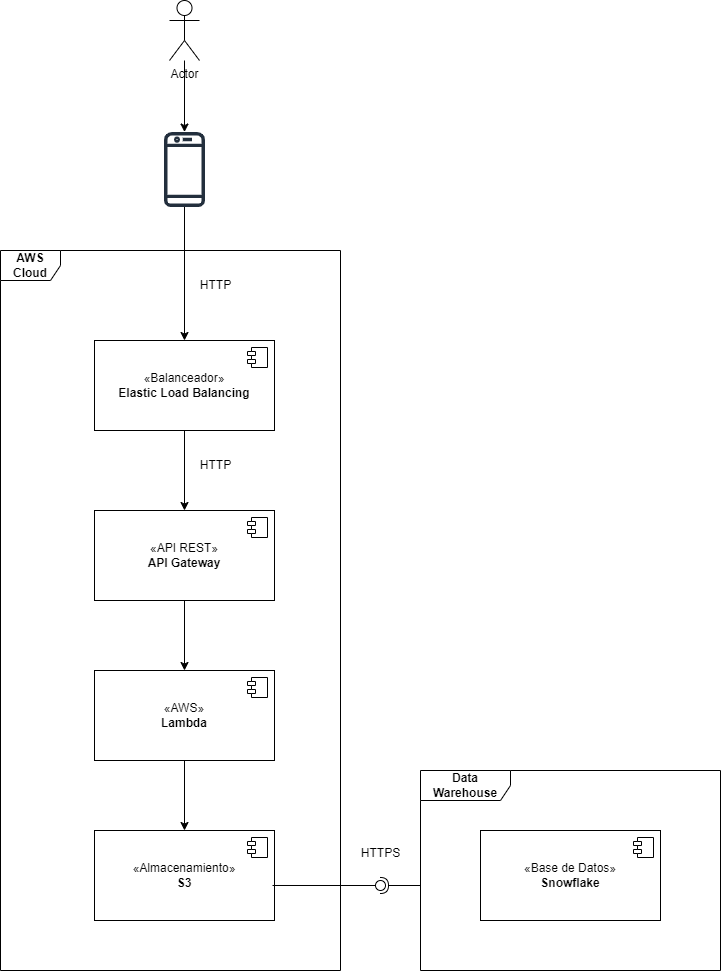
**Investigación de herramientas cloud**

En base al problema se planteó una solución escalable y económica utilizando Sensor Logger, un aplicativo móvil que registra datos telemétricos del usuario y su entorno, además, permite enviar los datos a través de peticiones HTTP desde el celular. Después, mediante una investigación se determinó que para llevar a cabo las pruebas es posible utilizar los servicios de nube gratuitos, los cuales permiten ahorrar costos y tiempo.

Los siguientes servicios de nube son gratuitos y escalan automáticamente, permitiendo llevar a cabo la solución planteada:

* **API Gateway →** Este servicio tiene como función ser la puerta de entrada entre la aplicación móvil y la nube. Este servicio se encarga de gestionar las peticiones HTTP de los datos de telemetría que envía Sensor Logger y dirigirlos a otros servicios de la nube.
* **AWS Lambda →** Este servicio recibe las peticiones de la API Gateway, y ejecuta el código que se encarga de procesar, transformar, validar y enviar los datos al bucket de almacenamiento.
* **Amazon S3 →** Este servicio almacena los datos enviados por la función Lambda de manera segura, además, maneja diferentes formatos como JSON, CSV u otro formato que sea compatible con el procesamiento posterior de los datos.
* **Snowflake →** Es una plataforma de datos en la nube que te permite consultar y analizar grandes volúmenes de datos. Además, permite la integración con otros servicios, como los de AWS.

**Diseño del Proyecto**

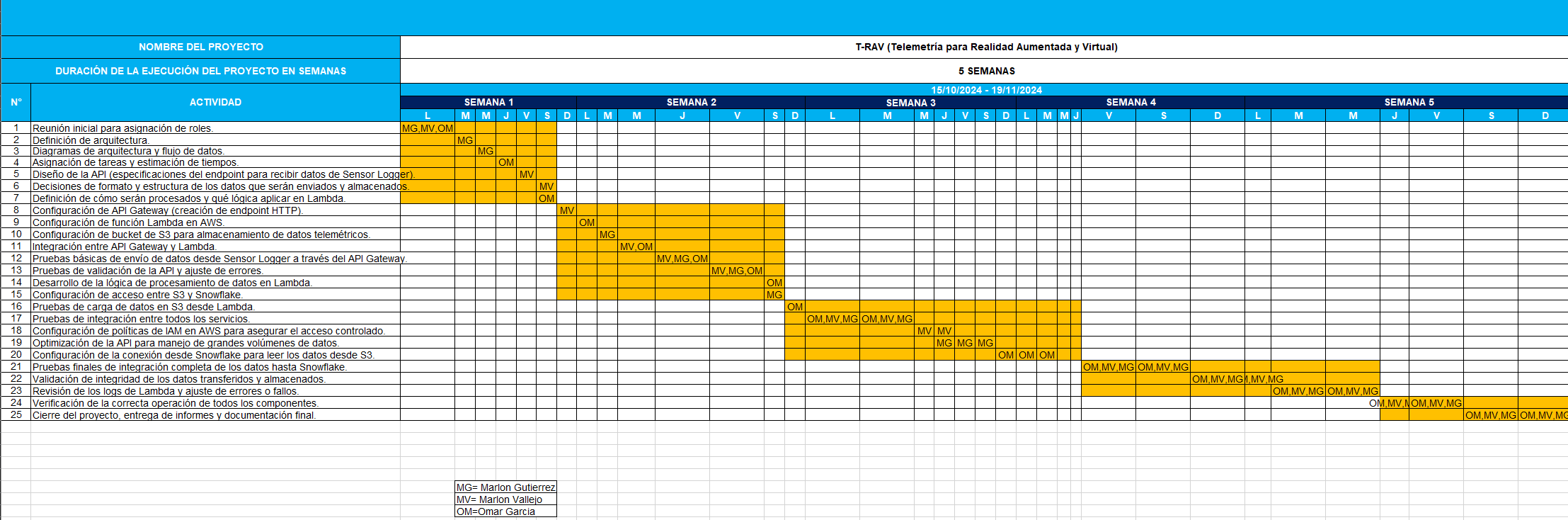


**Flujo de trabajo**

El flujo sería el siguiente, el Sensor Logger envía los datos telemétricos recolectados al API Gateway mediante solicitudes HTTP. El API Gateway recibe las solicitudes y las redirige a una función Lambda. La función Lambda procesa los datos de telemetría, validándolos y transformándolos si es necesario. Una vez procesados, la función Lambda almacena los datos en un bucket de Amazon S3. Y finalmente, Snowflake consume los datos desde S3 para realizar análisis y visualizaciones determinadas.

**Cronograma**

[T-RAV](Cronograma%20T-RAV.xlsx)



**Presupuesto**

* **API Gateway →** **Límite gratuito:** 1 millón de solicitudes al API por mes, además, de 750.000 minutos de conexión al mes durante un máximo de doce meses.
* **AWS Lambda → Límite gratuito:** 1 millón de solicitudes gratuitas al mes y hasta 3,2 millones de segundos de tiempo de informática por mes.
* **Amazon S3 → Límite gratuito**: 5 GB de almacenamiento estándar, 20 mil solicitudes GET, y 2 mil solicitudes PUT por 12 meses.
* **Snowflake → Límite gratuito:** Snowflake ofrece una prueba gratuita por tiempo limitado con un crédito inicial de $400 dólares.

Ahora, la estrategia a implementar para evitar sobrepasar el límite gratuito sería la siguiente:

* **Monitoreo constante**: Utiliza las herramientas de monitoreo de AWS como CloudWatch para supervisar el uso de todos los servicios en tiempo real. Configura alarmas para que te notifiquen cuando el consumo se acerque al límite gratuito.
* **Optimización de recursos:** Si es posible, agrupa o comprime los datos antes de enviarlos para reducir el número de solicitudes y el almacenamiento utilizado. Además, optimiza el código en las funciones Lambda y las consultas de Snowflake para minimizar el tiempo de ejecución y el uso de recursos.
* **Políticas de almacenamiento y ciclo de vida:** Implementa políticas de ciclo de vida en S3 para que los datos sean eliminados después de un cierto periodo de tiempo.
* **Ajustar frecuencias de envío de datos:** Si los datos de telemetría no necesitan ser enviados en tiempo real, puedes reducir la frecuencia de las solicitudes a la API.