Proyecto # 2

Cristian Camilo Lopera Galvis

María Camila Quintero

Docente: Alexander López Parrado

Asignatura: IOT

Maestría en Ingeniería – Analítica de Datos
Facultad de Ingeniería
Universidad del Quindío
2023

Second stage (cloud computing): it is required you add windowed queries to the database and new REST API resources to allow the following:

- Request all sensor data between two dates.
- Request all sensor data between two times in a given date.
- Avoid scan-type queries.

Solución

Dadas las Lambdas creadas en AWS con Python, se crearon 6 recursos, de los cuales uno de ellos es la solución al trabajo planteado para la segunda entrega.

A continuación, se presenta el enlace general a la API REST en AWS:

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha

Como se había mencionado se tienen 6 recursos GET para obtener data como se evidencia en la siguiente figura.

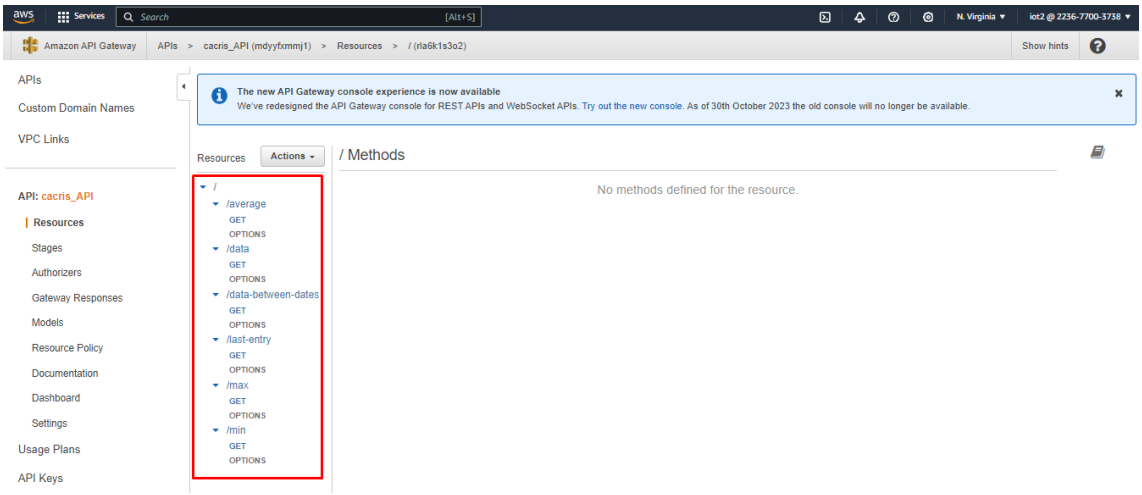

Figure 1 Resources

Inicialmente se creó el recurso /data para obtener toda la información de la base de datos DynamoDB y luego se crearon los demás, los cuales se van a presentar de la siguiente manera:

Data:

Esta es la URL para obtener toda la información que se tiene en la base de datos, https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/data

En la siguiente imagen se puede evidenciar el uso de este request y el response de toda la información almacenada en la tabla e DynamoDB.

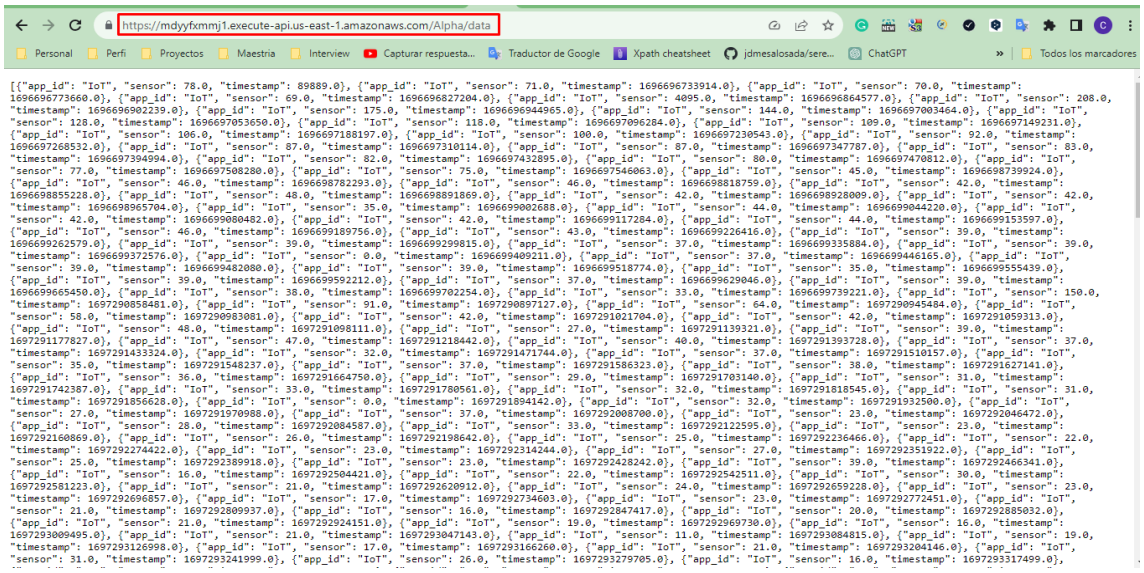

Figure 2. Data request

Last-entry:

Se creo el siguiente request para obtener el ultimo valor que se almaceno en la tabla de DynamoDB, por medio de la siguiente URL se puede obtener este dato.

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/last-entry

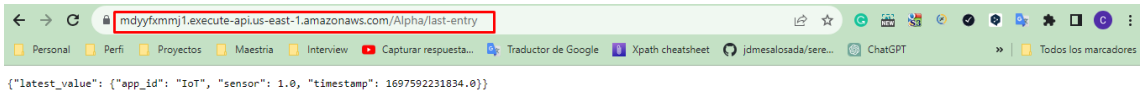

Figure 3. Last-entry request

Average:

Se creo el siguiente request para obtener el valor promedio que se almaceno en la tabla de DynamoDB, por medio de la siguiente URL se puede obtener este dato.

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/average

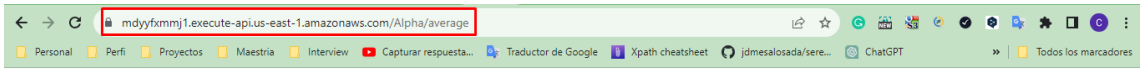

Figure 4. Average request

Min:

Se creo el siguiente request para obtener el mínimo valor que se almaceno en la tabla de DynamoDB, por medio de la siguiente URL se puede obtener este dato.

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/min

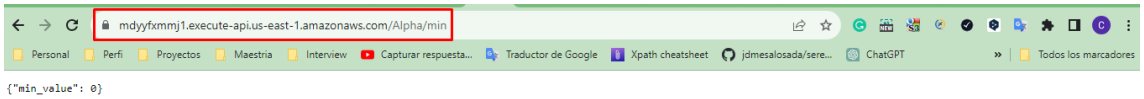

Figure 5. Min request

Max:

Se creo el siguiente request para obtener el máximo valor que se almaceno en la tabla de DynamoDB, por medio de la siguiente URL se puede obtener este dato.

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/max

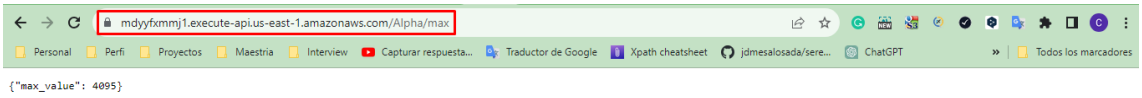

Figure 6. Max request

data-between-dates:

Por último, se creó el siguiente request;

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/data-between-dates?start_date=2023-10-15T01:31:00&end_date=2023-10-21T00:00:00

Para darle solución a los dos requerimientos solicitados.

Este recibe dos parámetros que son start_date y end_date, estas fechas se reciben por medio de "%Y-%m-%dT%H:%M:%S", como se puede ver se puede ingresar el tiempo, por ende con un mismo recurso se puede dar solución a ambos requerimientos.

En la siguiente figura se puede evidenciar los datos obtenidos mediante el request con una fecha inicial "start_date=2023-10-10T00:31:00" y una fecha final "end_date=2023-10-18T10:00:00", como se solicitó en el primer requerimiento.

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/data-between-dates?start_date=2023-10-10T00:31:00&end_date=2023-10-18T10:00:00

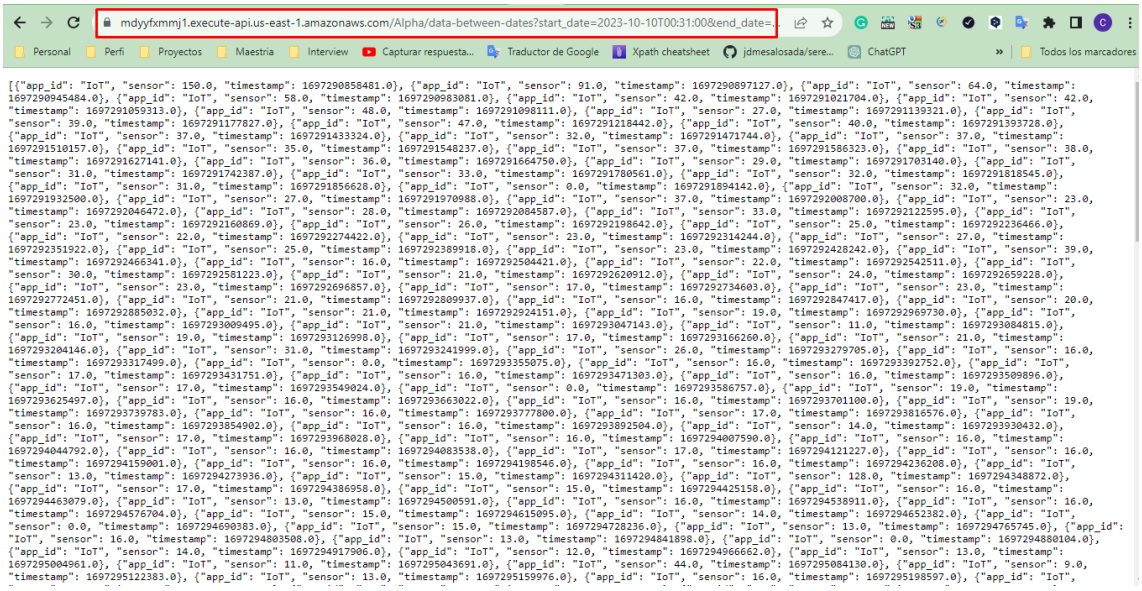

Figure 7. Data-between-dates request

Para darle solución al segundo requerimiento se ingreso el siguiente request

https://mdyyfxmmj1.execute-api.us-east-1.amazonaws.com/Alpha/data-between-dates?start_date=2023-10-18T00:31:00&end_date=2023-10-18T10:00:00

el cual tiene la misma fecha “2023-10-18T00:31:00” pero con diferente tiempo, la inicial es “start_date=2023-10-18T00:31:00” y la final “end_date=2023-10-18T10:00:00”, la cual obtiene como response la data que se puede observar en la siguiente figura.

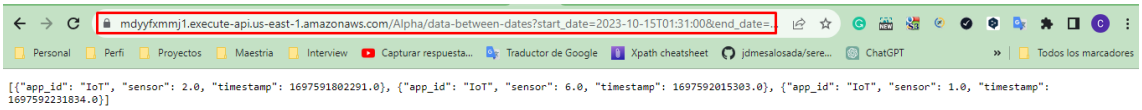

Figure 8. Data-between-dates times request

De manera general este request se compone de un queryStringParameters para obtener las dos fechas por parámetro, dado que la fecha que se ingresa esta sobre este formato “%Y-%m-%dT%H:%M:%S", procedemos a realizar una modificación con datetime.datetime.strptime para analizar una cadena de fecha y hora y convertirla en un objeto datetime, en caso tal de que se ingresó una fecha incorrecta, se ingresa a una exception como se puede evidenciar en la siguiente figura.

```
elif operation == '/data-between-dates':
    query_parameters = event.get('queryStringParameters', {})
    start_date_str = query_parameters.get('start_date')
    end_date_str = query_parameters.get('end_date')

    try:
        start_date = datetime.datetime.strptime(start_date_str, "%Y-%m-%dT%H:%M:%S")
        end_date = datetime.datetime.strptime(end_date_str, "%Y-%m-%dT%H:%M:%S")
    except ValueError:
        return {
            'statusCode': 400,
            'headers': {
                'Access-Control-Allow-Origin': '*'
            },
            'body': json.dumps('Invalid date format')
        }
```

Figure 9. datetime.datetime.strptime

Luego se procede a convertir esta fecha a timestamp y luego la ejecución para obtener la data entre las fechas especificadas como se puede evidenciar en la siguiente figura.

```
        start_timestamp = int(start_date.timestamp() * 1000)
        end_timestamp = int(end_date.timestamp() * 1000)

        if start_timestamp > end_timestamp:
            end_timestamp = start_timestamp + 1

        response = table.query(
            KeyConditionExpression = boto3.dynamodb.conditions.Key('app_id').eq('IoT') &
            boto3.dynamodb.conditions.Key('timestamp').between(start_timestamp, end_timestamp)
            )

        items = response.get('Items', [])
        return {
            'statusCode': 200,
            'headers': {
                'Access-Control-Allow-Origin': '*'
            },
            'body': json.dumps(items, cls=JSONEncoder)
        }
    else:
```

Figure 10. timestamp and response

NOTA:

El request le hacen faltan excepciones de tipo solo agregar o el start_date o el end_date

Se agrega la URL del repositorio de GitHub para analizar el código de cada una de las Lambdas:

https://github.com/cristianlopera24/IoT