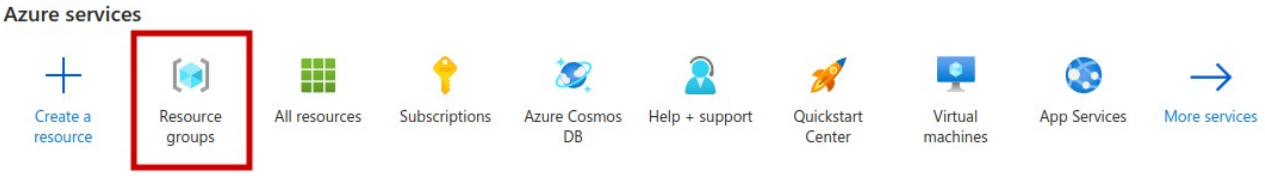
Práctica obligatoria - Elaboración de una memoria de la práctica

5.0. Creación cuenta de estudiante Azure

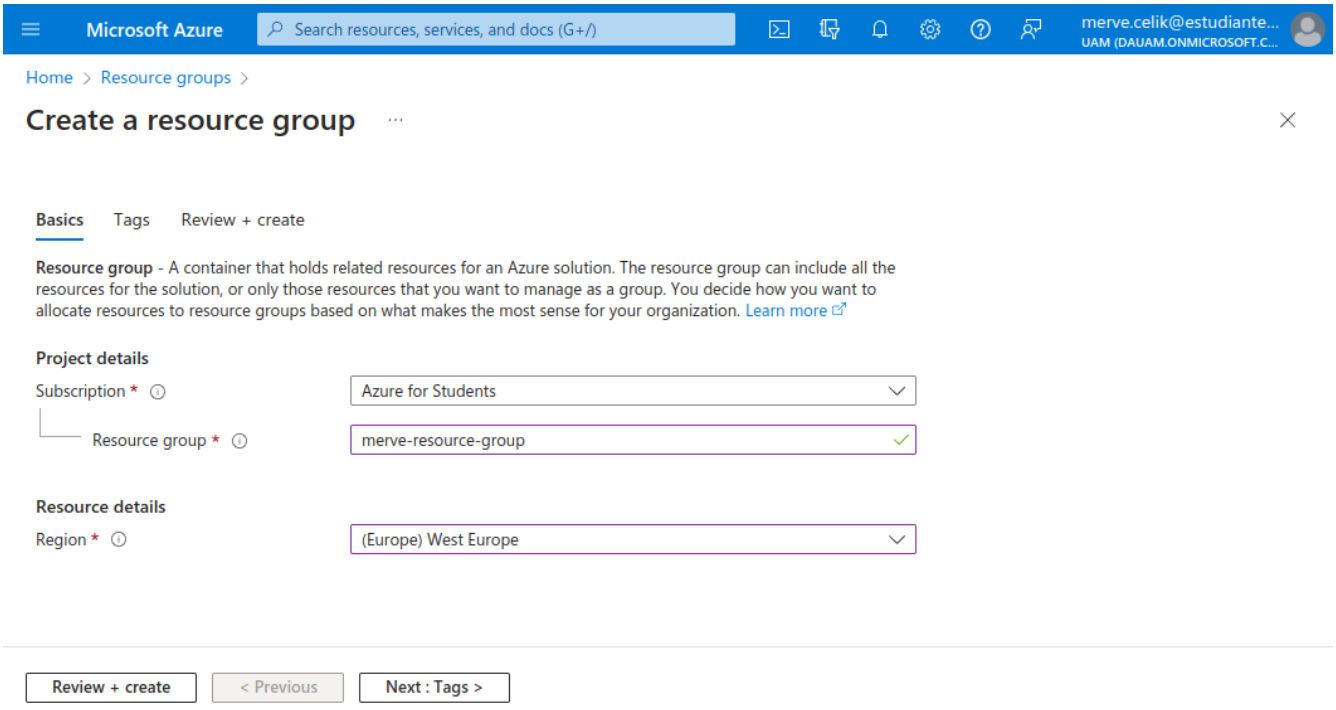
Hemos creado una cuenta de estudiante en Azure.

5.1. Creación de grupo de recursos

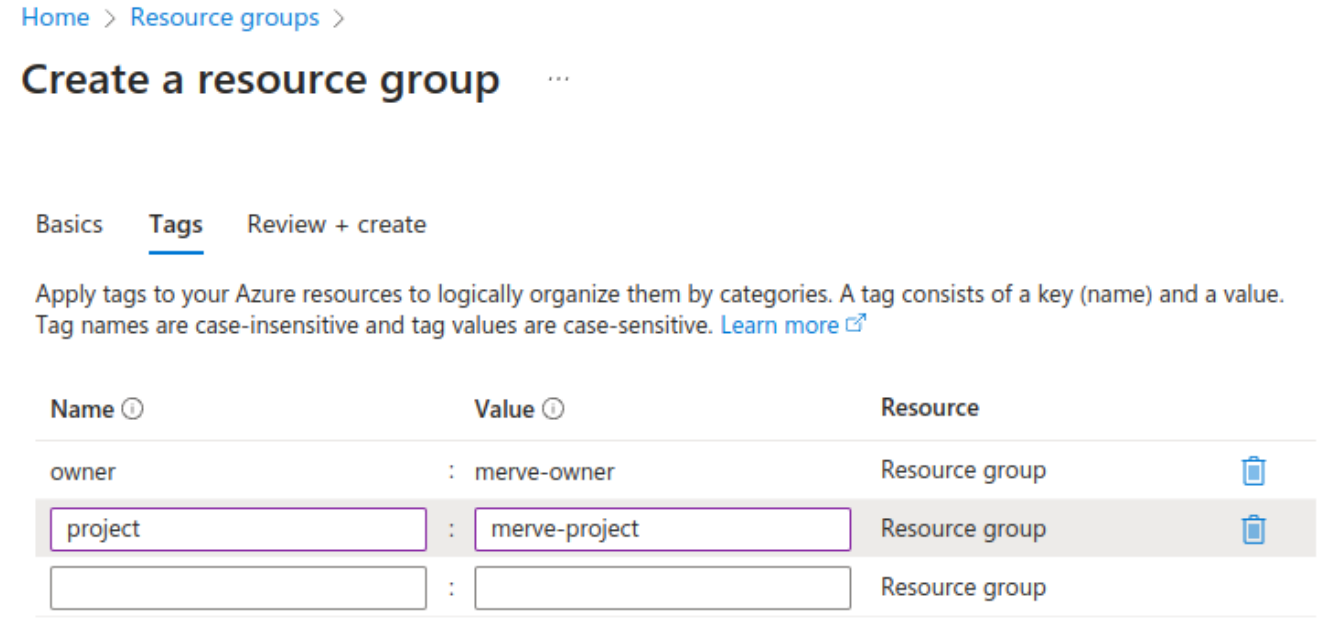
Desde la página de home pulsamos Resource Group para crear un grupo de recursos.



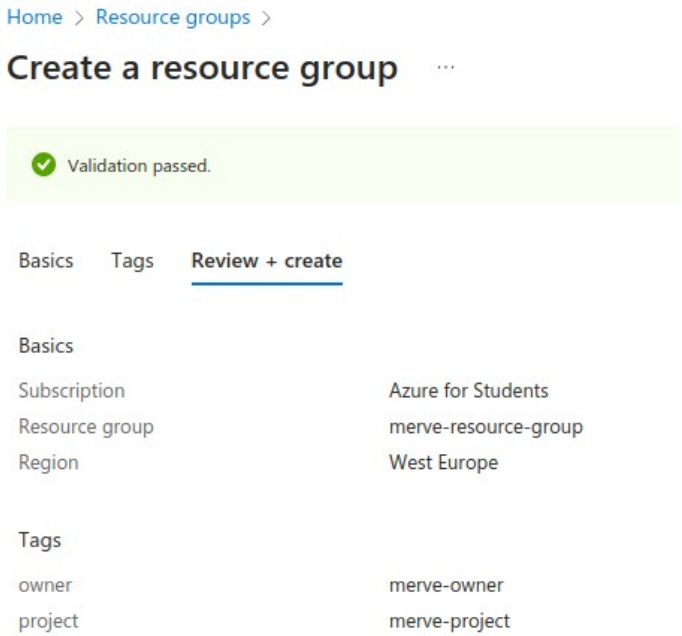
Después de hacer click en Resource Group, pulsamos “Create Resource Group”.
Rellenamos la pestaña “Basic”:
Subscription: Azure for Students
Resource group: merve-resource-group
Region: (Europe) West Europe

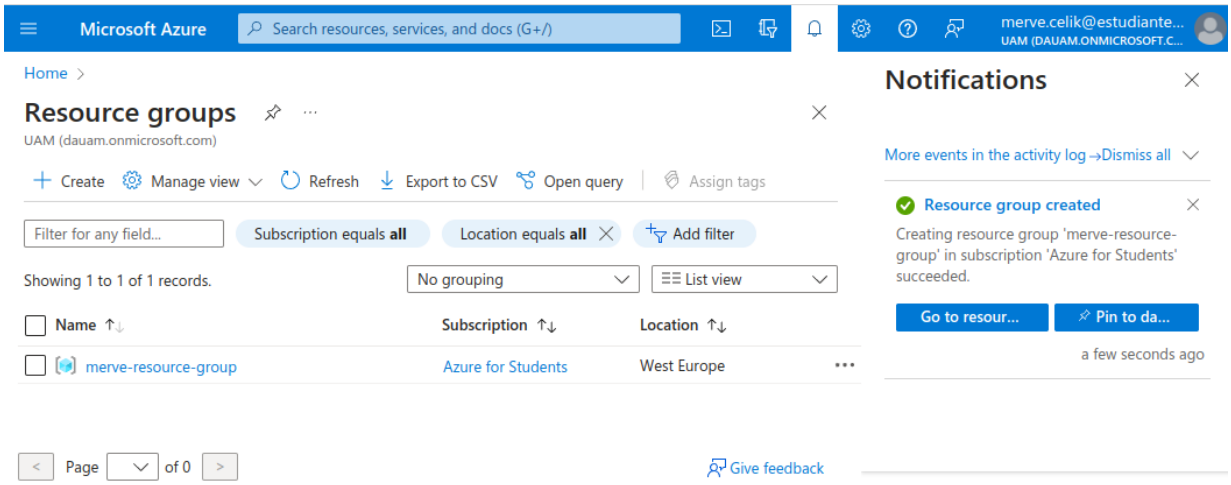


En la pestaña “Tags”:
Rellenamos 2 tags para para ayudarnos a la gestión de los servicios:
Name | Value
owner **merve-owner**
project **merve-project**



En la pestaña Review + Create
Cuando vemos Validation passed, pulsamos en Create

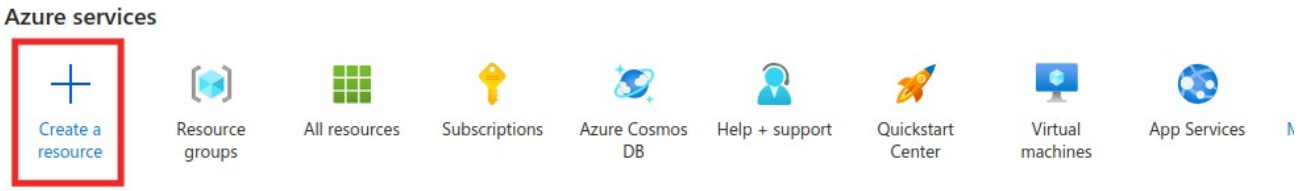




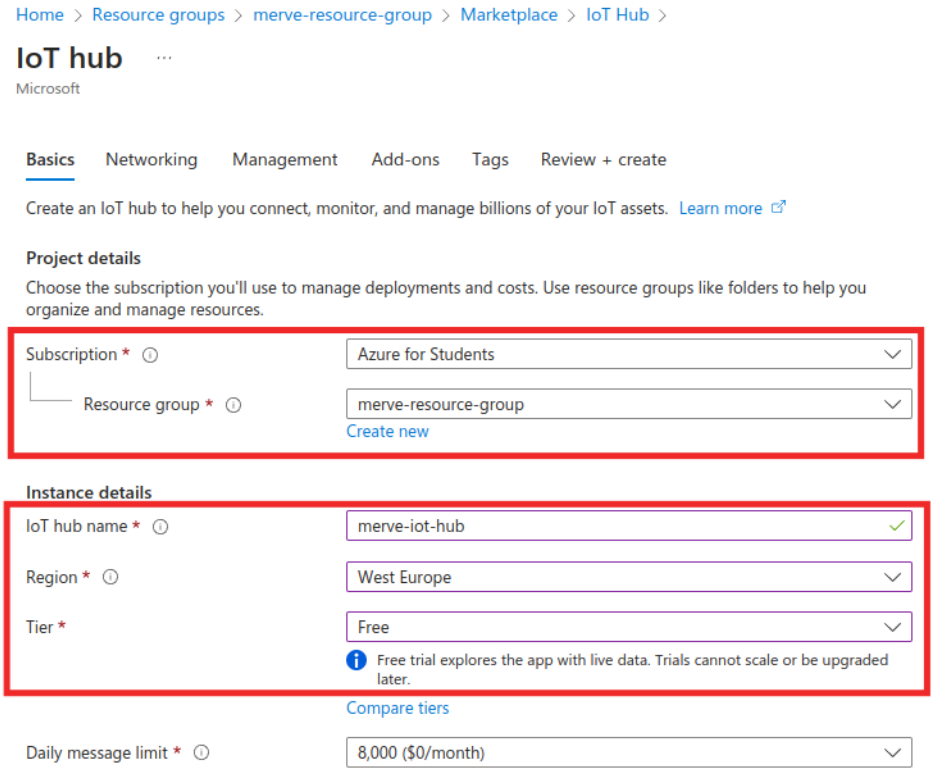
Como vemos aquí, hemos creado el resource group con éxito.

5.2. Creación de servicio de IoT Hub

Desde la página de home, pulsamos Create a Resource para crear servicios.

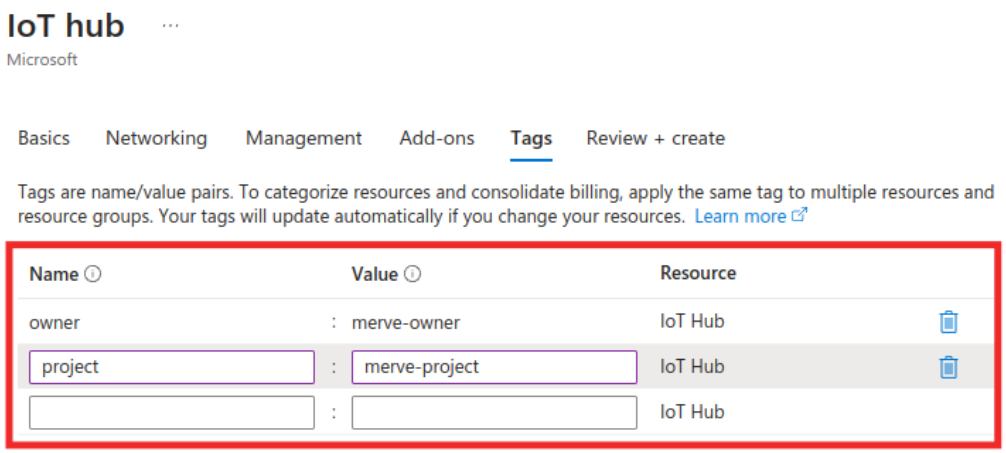


Primero, creamos el servicio de IoT Hub. Para ello, escribimos “IoT Hub” en el buscador y hacemos click en él. Después de pulsar “Create”, rellenamos la pestaña “Basic”.

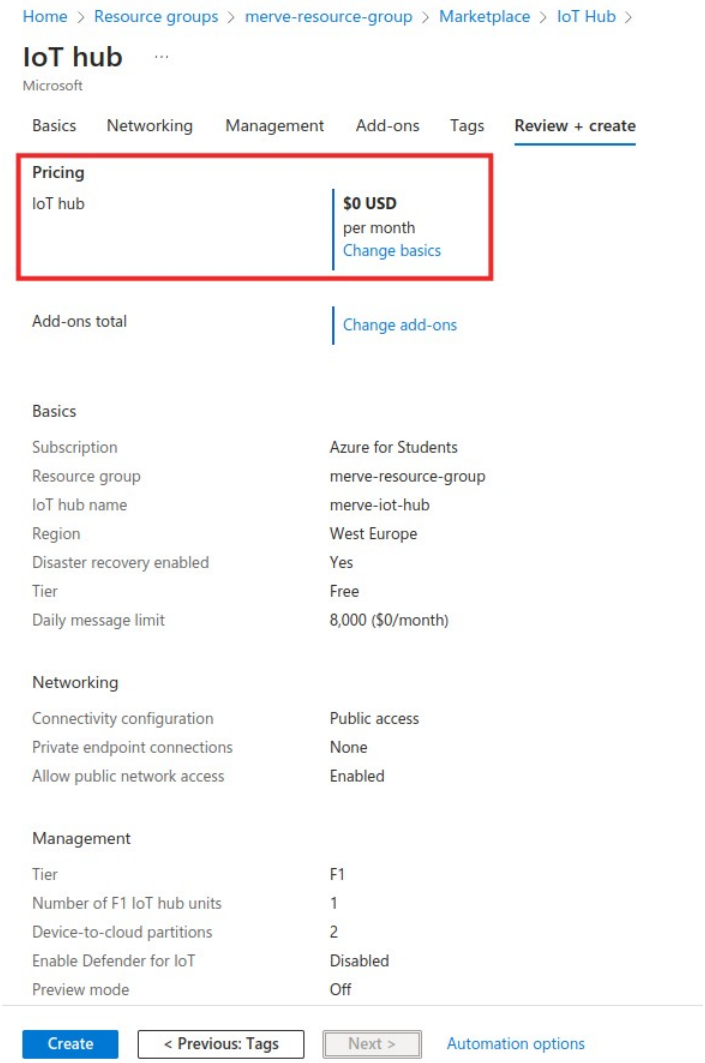


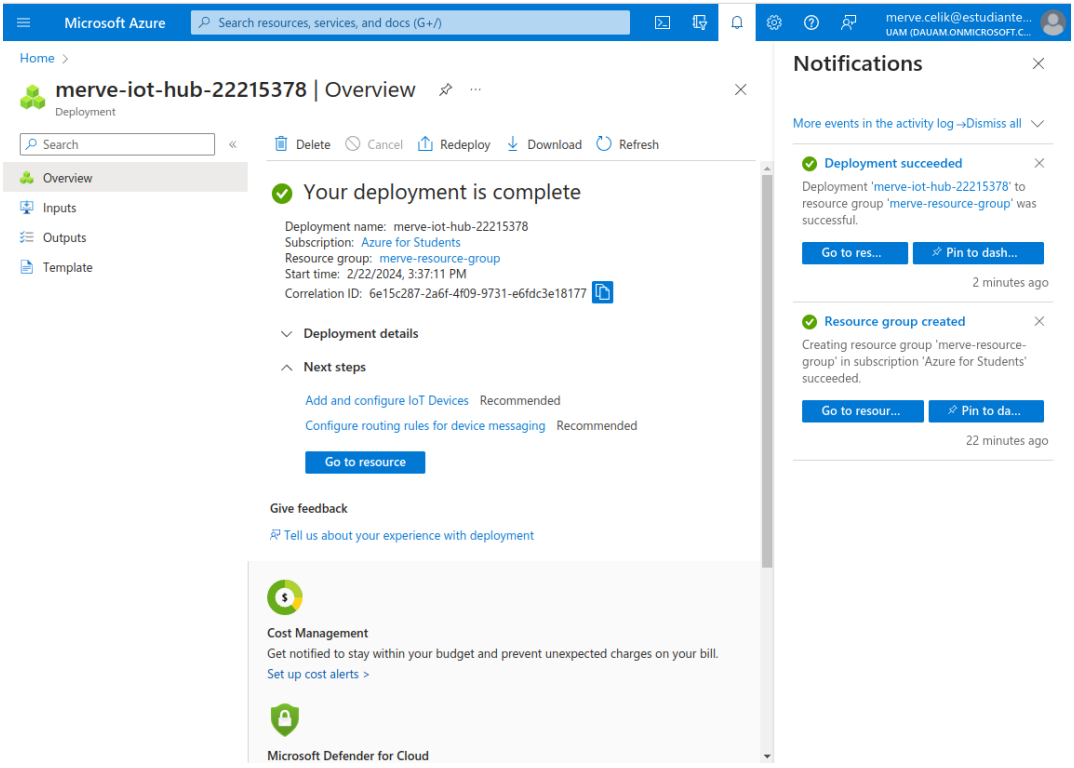
En la pestaña “Tags”, rellenamos 2 tags para ayudarnos a la gestión de los servicios como antes:

Name | Value
owner **merve-owner**
project **merve-project**



En la pestaña de Review + create visualizamos los detalles del servicio, especialmente el precio, antes de la creación. Si todo está correcto, podemos pulsar en “Create”.



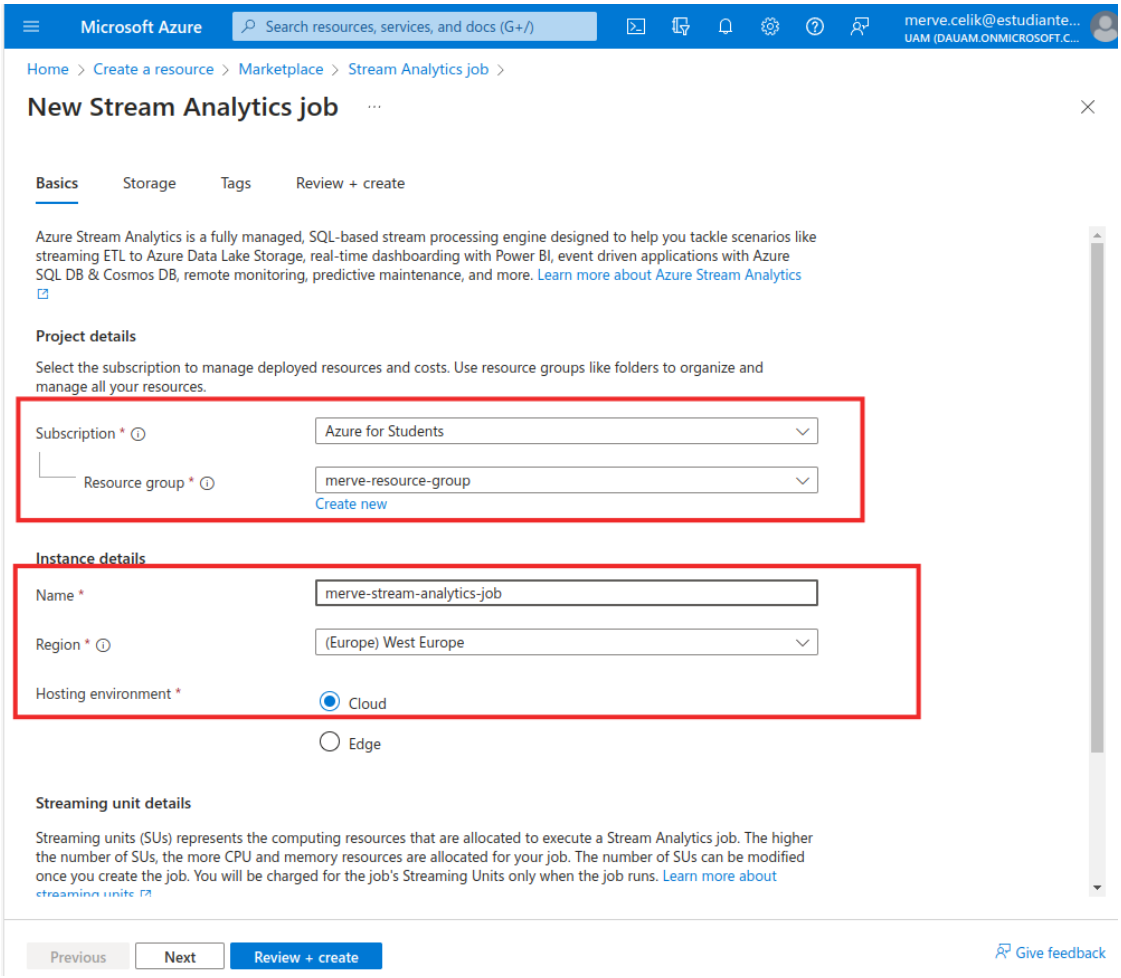


"Como vemos que “Your deployment is complete”, hemos creado el servicio IoT Hub con éxito."

5.3. Creación de servicio de Stream Analytics Job

Desde la página de home, pulsamos Create a Resource para crear servicios nuevamente.

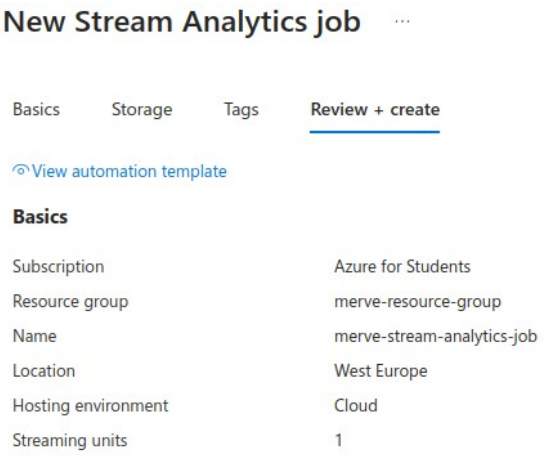
Para crear el servicio de Stream Analytics Job, escribimos “Stream Analytics Job” en el buscador y hacemos click en él. Después de pulsar “Create”, rellenamos la pestaña “Basic”.



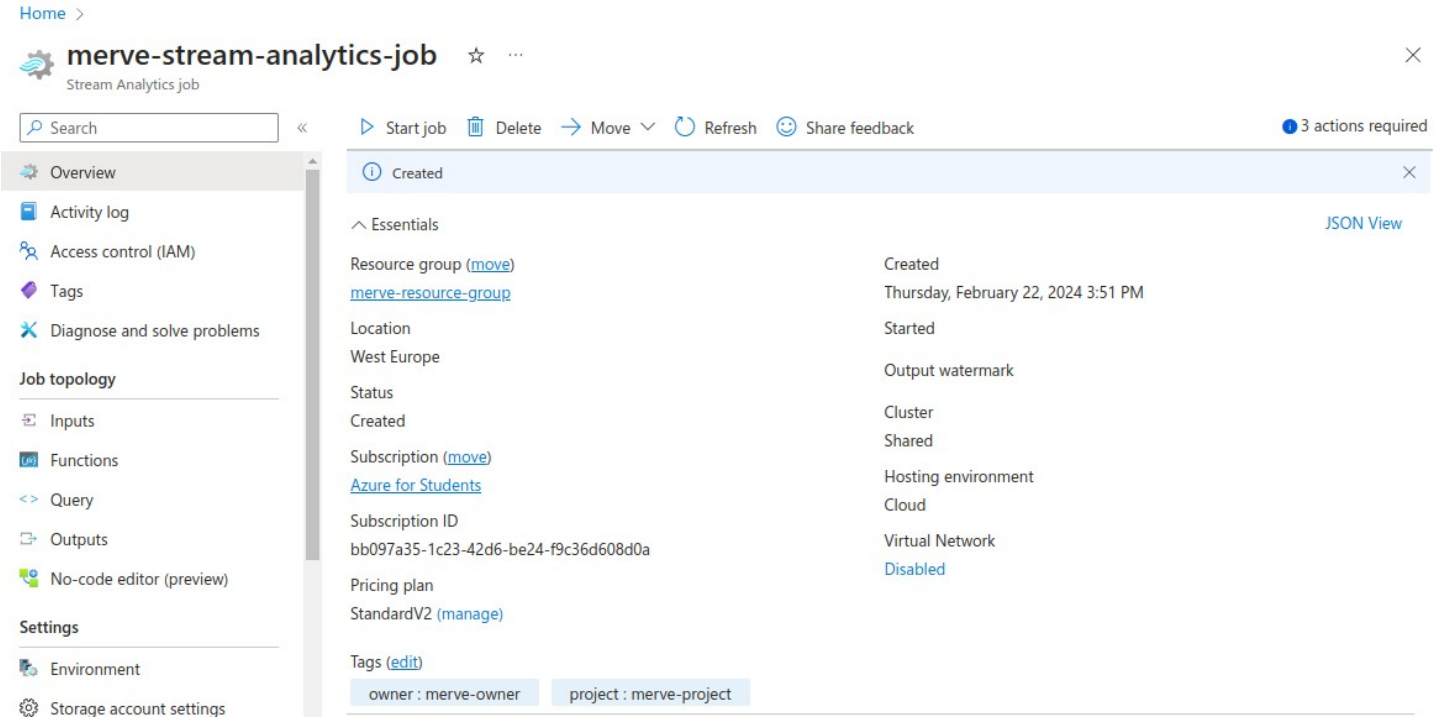
En la pestaña “Tags”, rellenamos 2 tags para ayudarnos a la gestión de los servicios como antes:

Name | Value
owner **merve-owner**
project **merve-project**

En la pestaña de Review + create también visualizamos los detalles del servicio, especialmente el precio, antes de la creación. Si todo está correcto, podemos pulsar en “Create”.



Vemos que hemos creado el servicio de Stream Analytics Job con éxito.

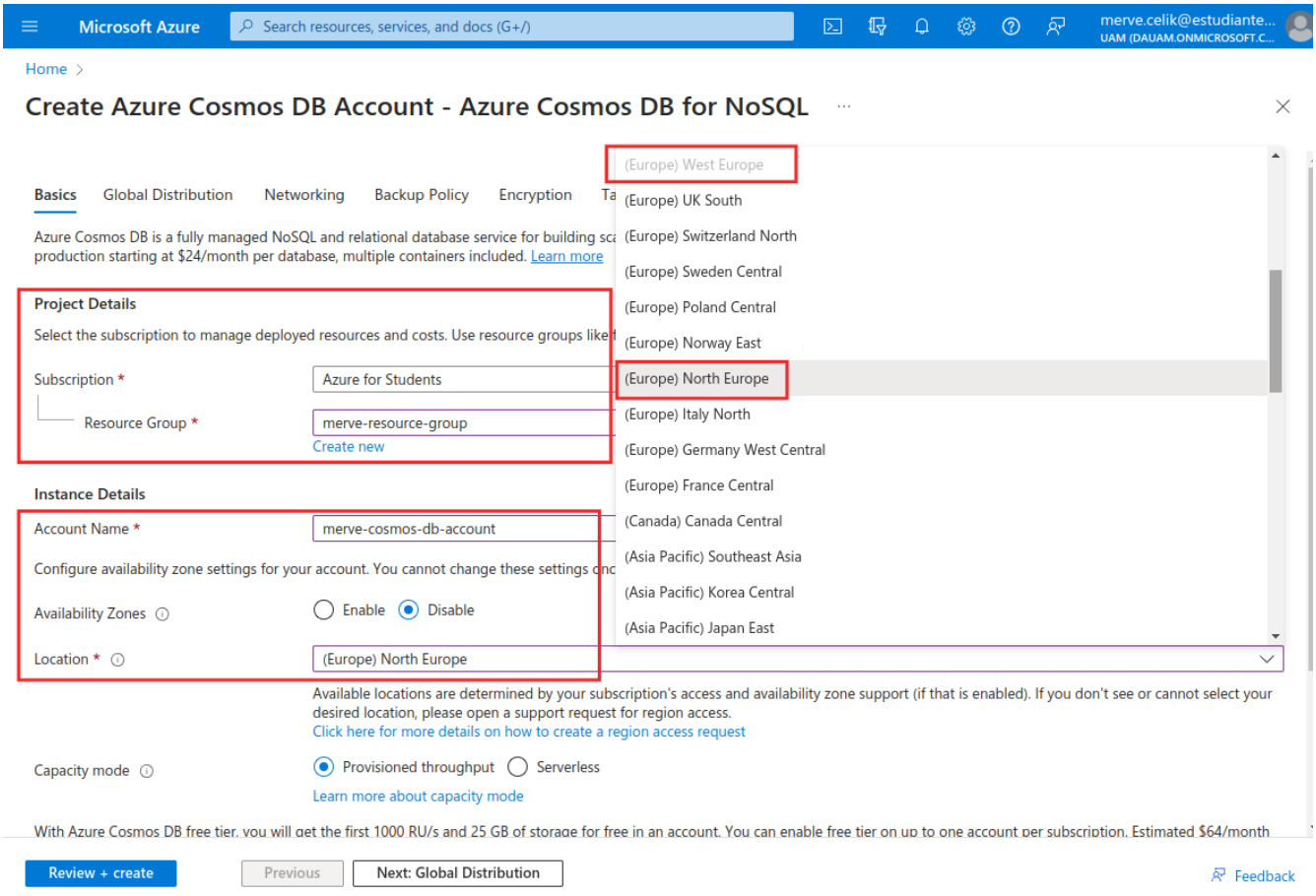


5.4. Creación de servicio de Azure Cosmos DB

Desde la página de home, pulsamos Create a Resource para crear servicios nueavamente.

Escribimos “Azure Cosmos DB” para encontrarlo y hacemos click en él. Después de pulsar “create” seleccionamos “Azure Cosmos DB for NoSQL”.

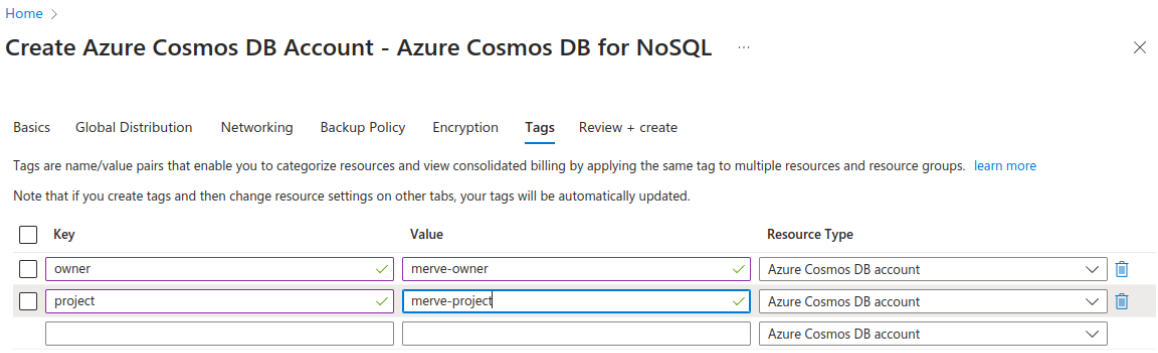
Luego, rellanamos la pestaña Basic:



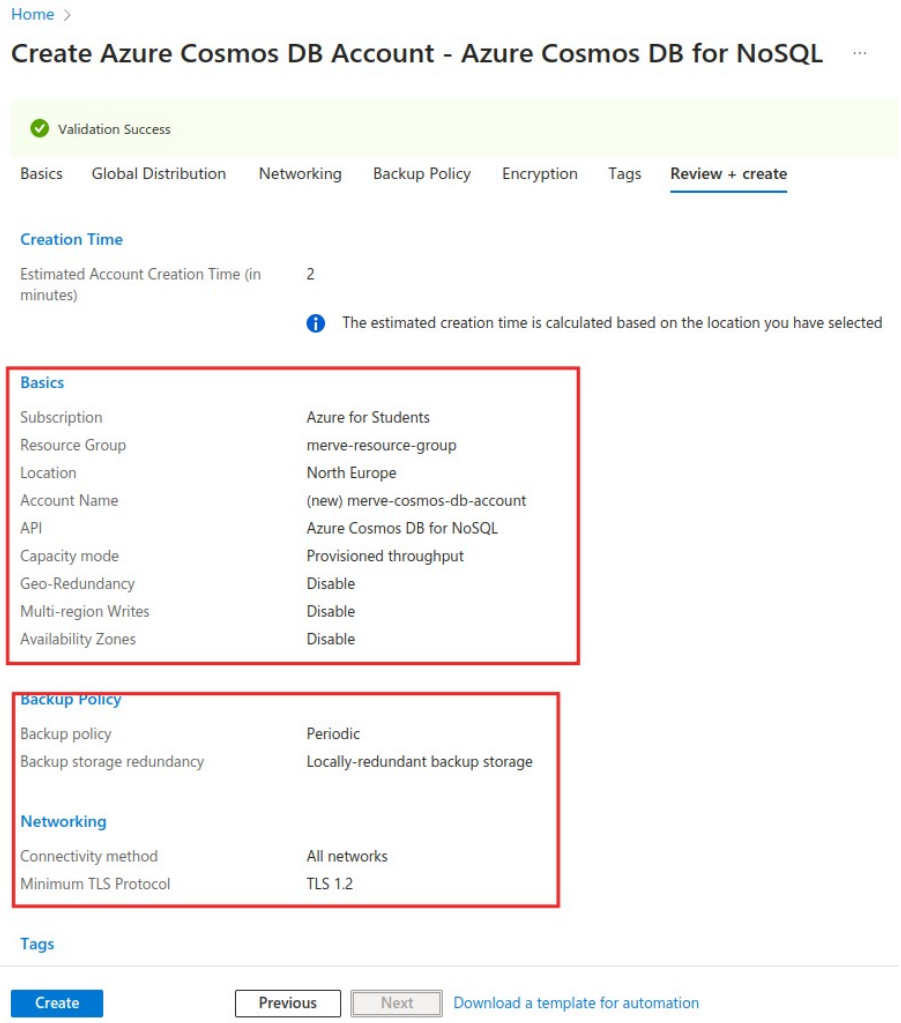
Aunque he intentado refrescar la página varias veces e incluso intenté crearla desde otros navegadores, elegí la opción de (Europe) North Europe porque la de (Europe) West Europe no aparecía.

En la pestaña “Tags”, rellenamos 2 tags para ayudarnos a la gestión de los servicios como antes:

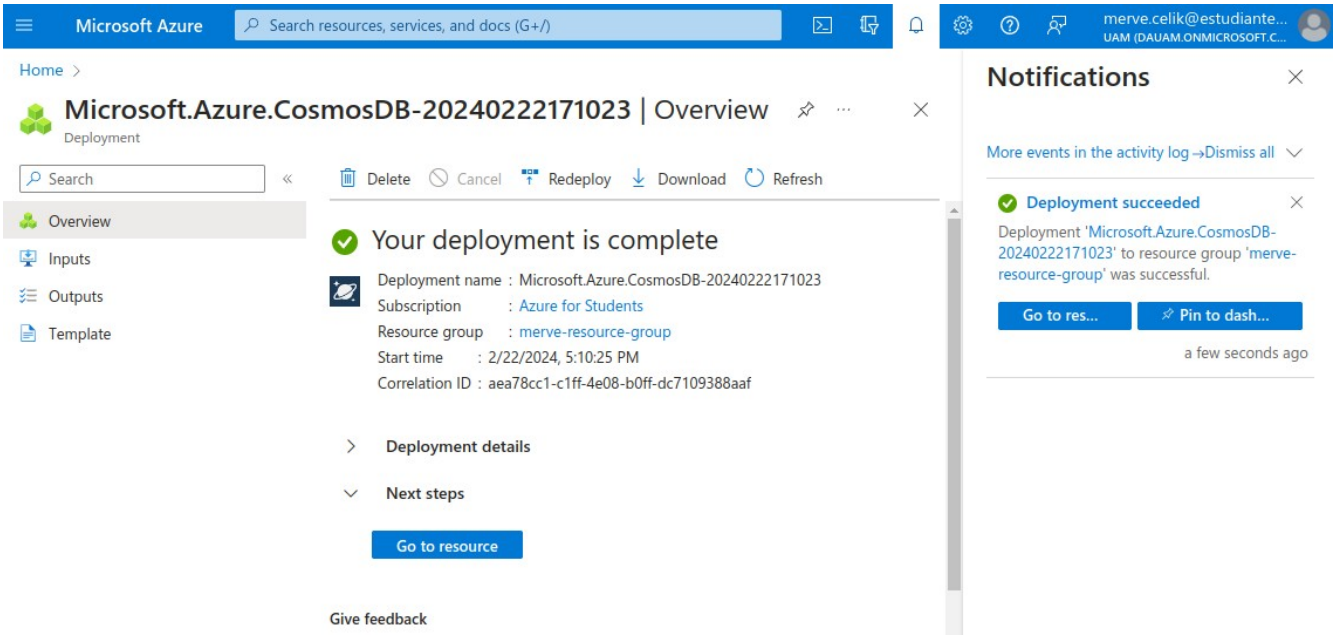
Name | Value
owner **merve-owner**
project **merve-project**



En la pestaña de Review + create también visualizamos los detalles del servicio antes de la creación. Si todo está correcto, podemos pulsar en “Create”.

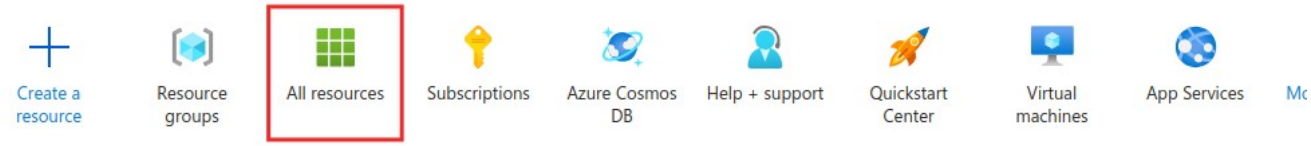


"Como vemos que “Your deployment is complete”, hemos creado el servicio Azure Cosmos DB con éxito."

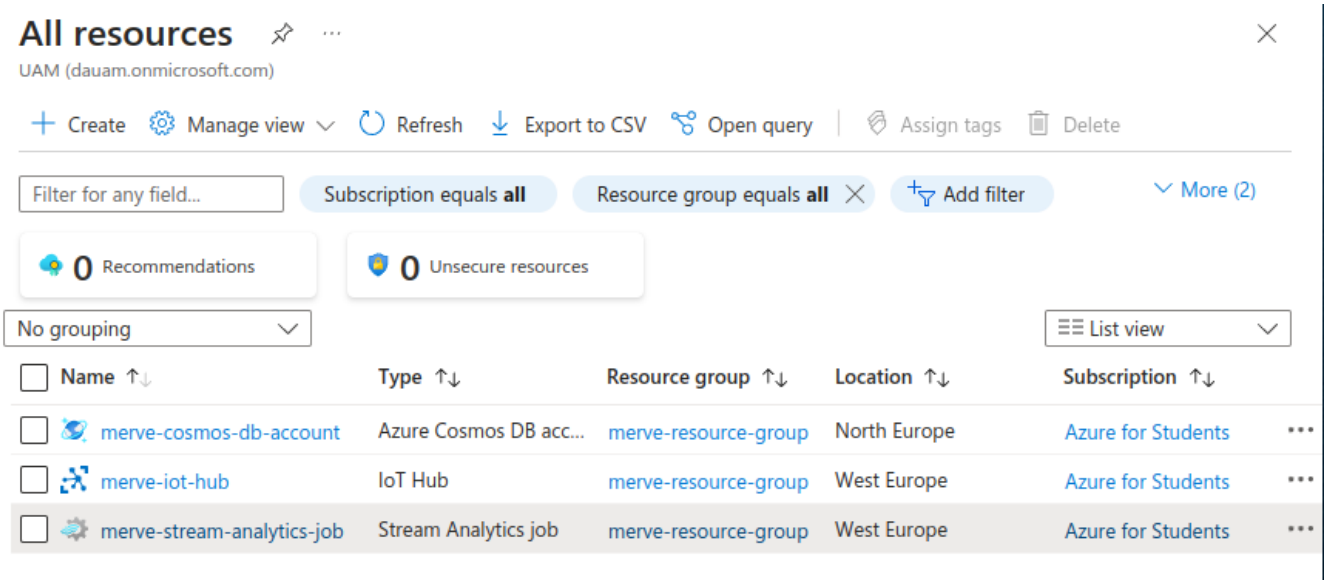


5.5. Verificación servicios creados

Desde la página de home, hacemos click en All resources para verificar si los servicios se han creado.

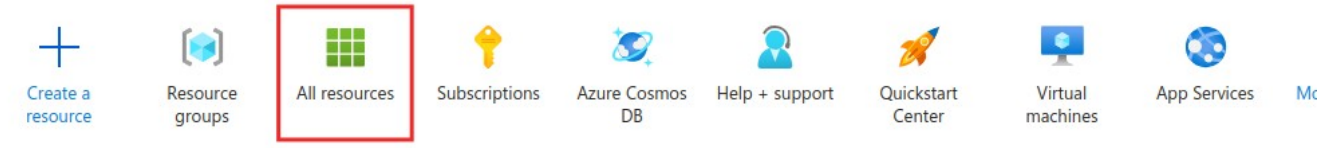


Vemos que todos los servicios se han creado correctamente.

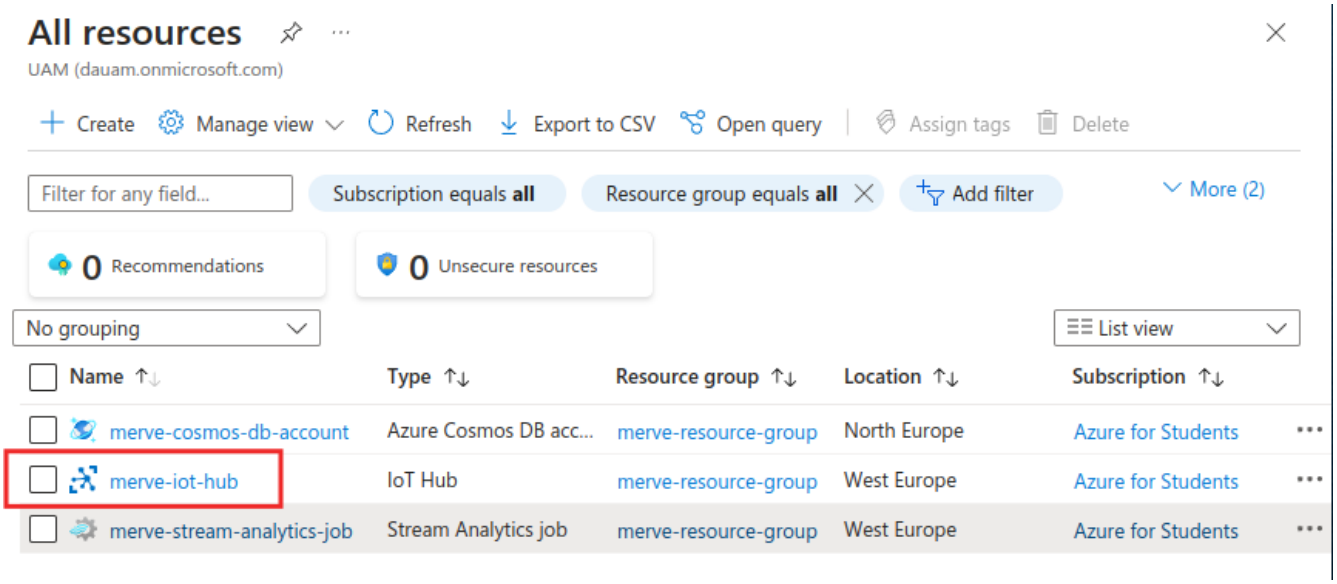


5.6. Creación Dispositivo en IoT Hub

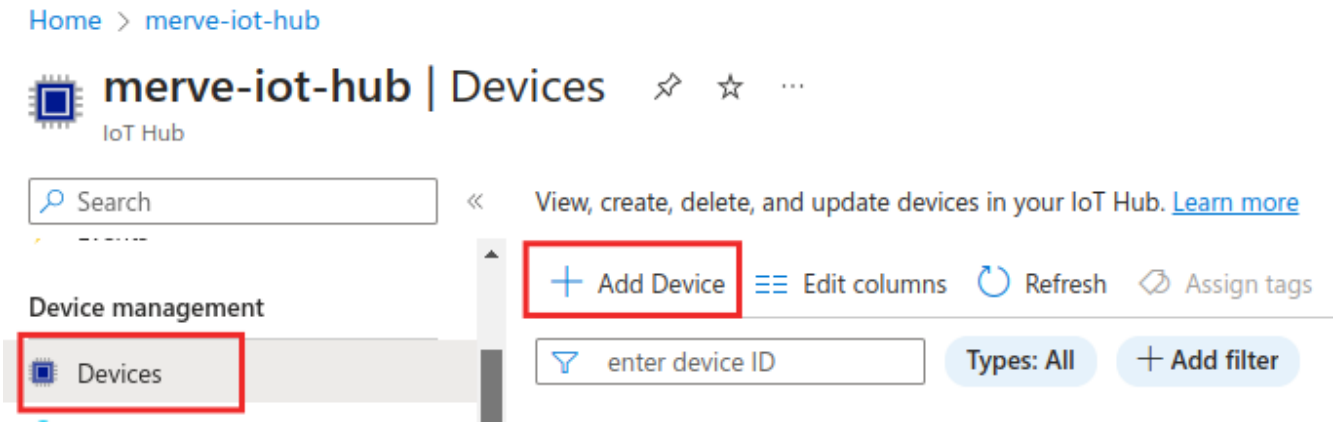
Desde la página de home, hacemos click en All resources.



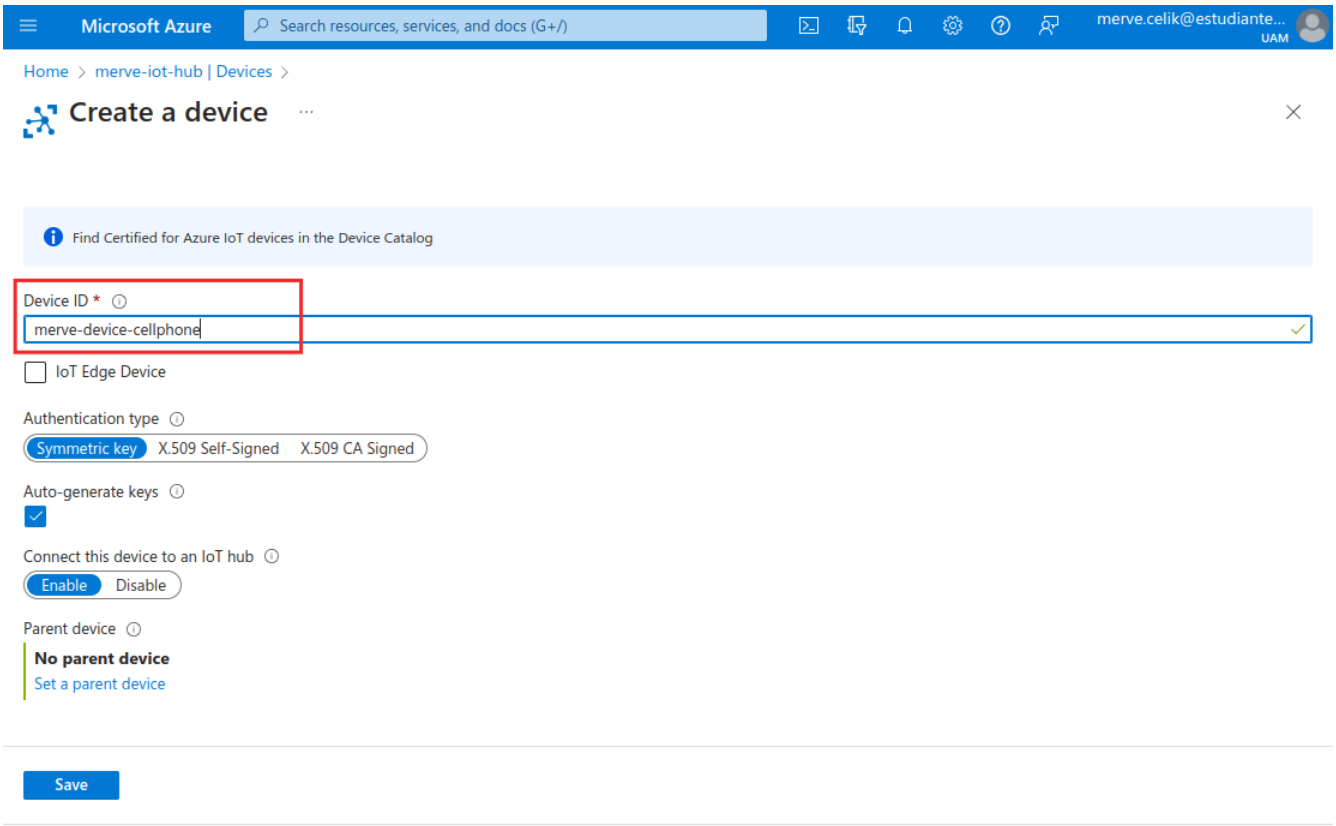
Pulsamos sobre merve-iot-hub.



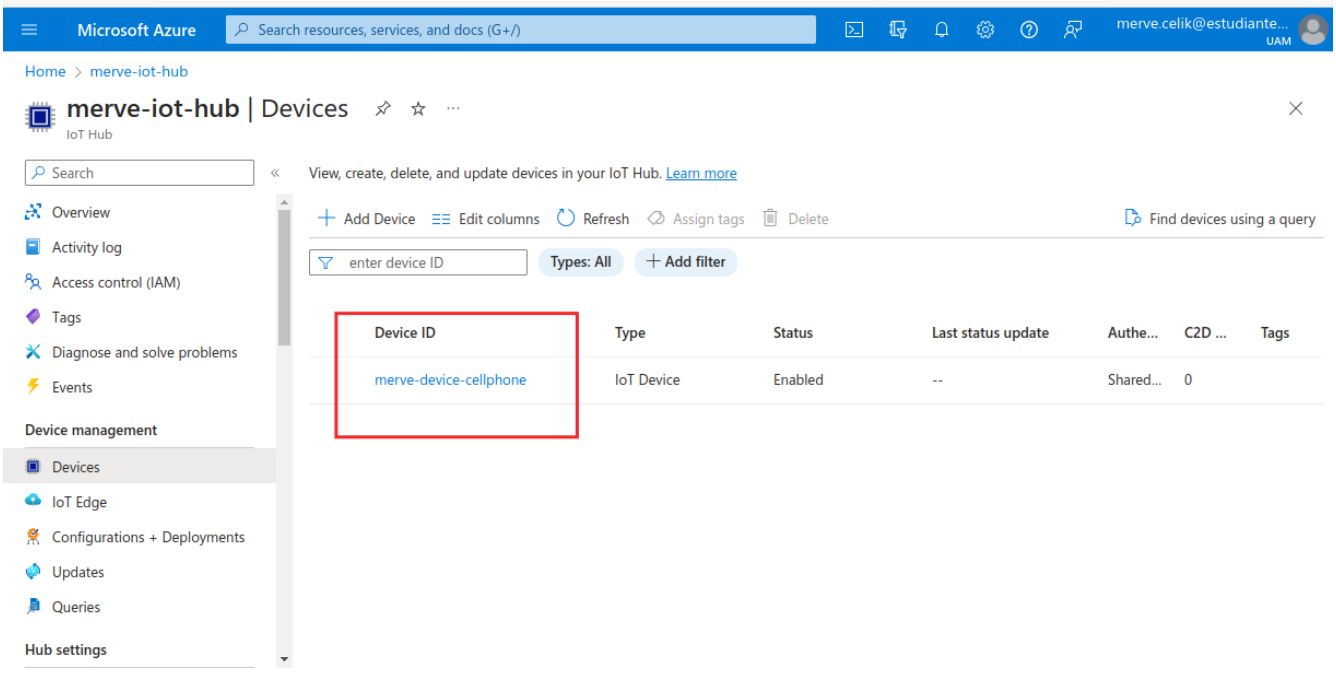
Primero, pulsamos en “Devices” y luego en “Add Device”.



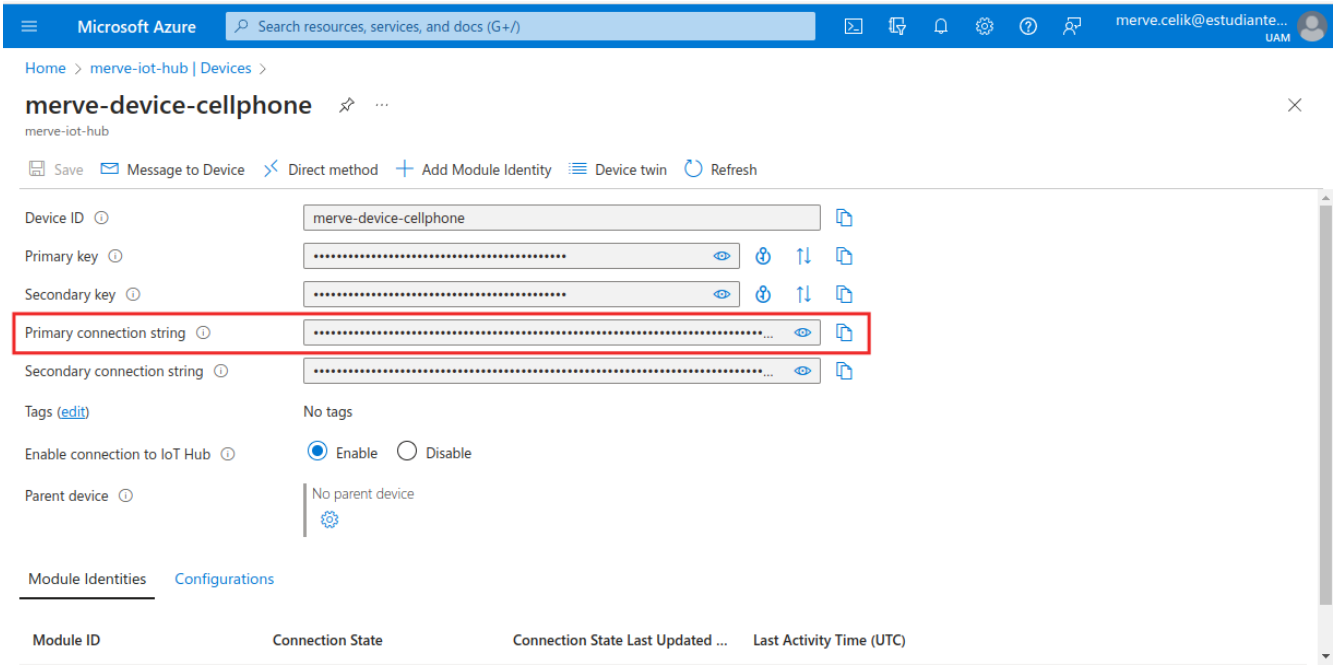
Rellenamos los campos:
Elegimos un nombre para **Device ID** personalizado: **merve-device-cellphone**
Y pulsamos “Save”.



Vemos que hemos creado el dispositivo correctamente y pulsamos sobre el nombre del dispositivo.



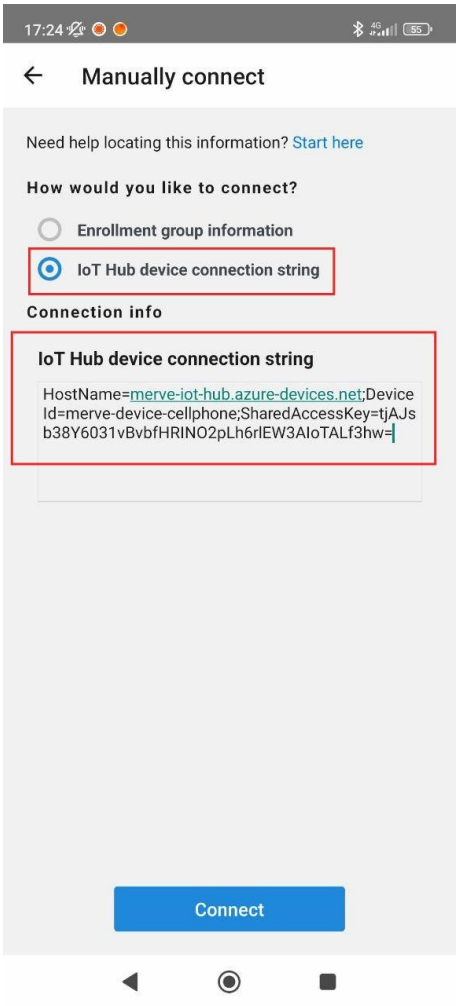
Copiamos la **Primary connection string** al clipboard.

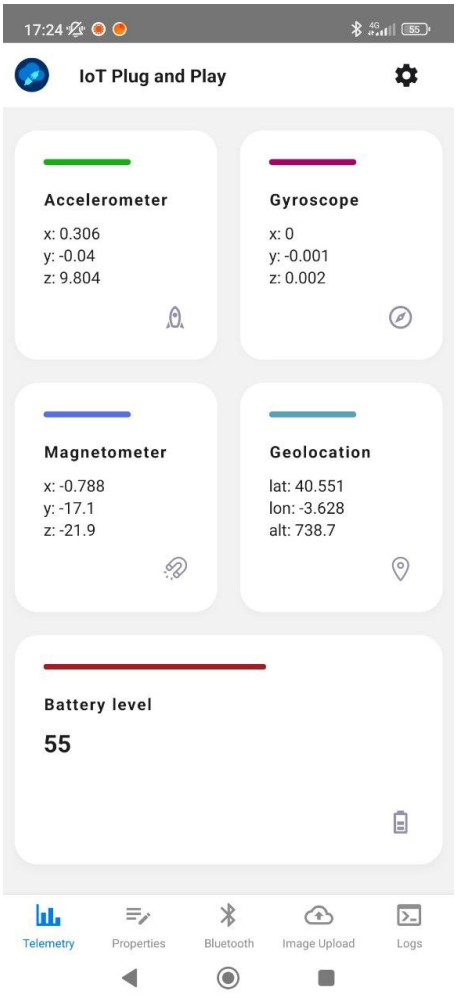


5.7. Descarga y configuración de la app IoT PnP

Descargamos una aplicación de "IoT Plug and Play" para nuestro teléfono móvil que nos permite conectarnos a la plataforma prácticamente de forma plug and play.

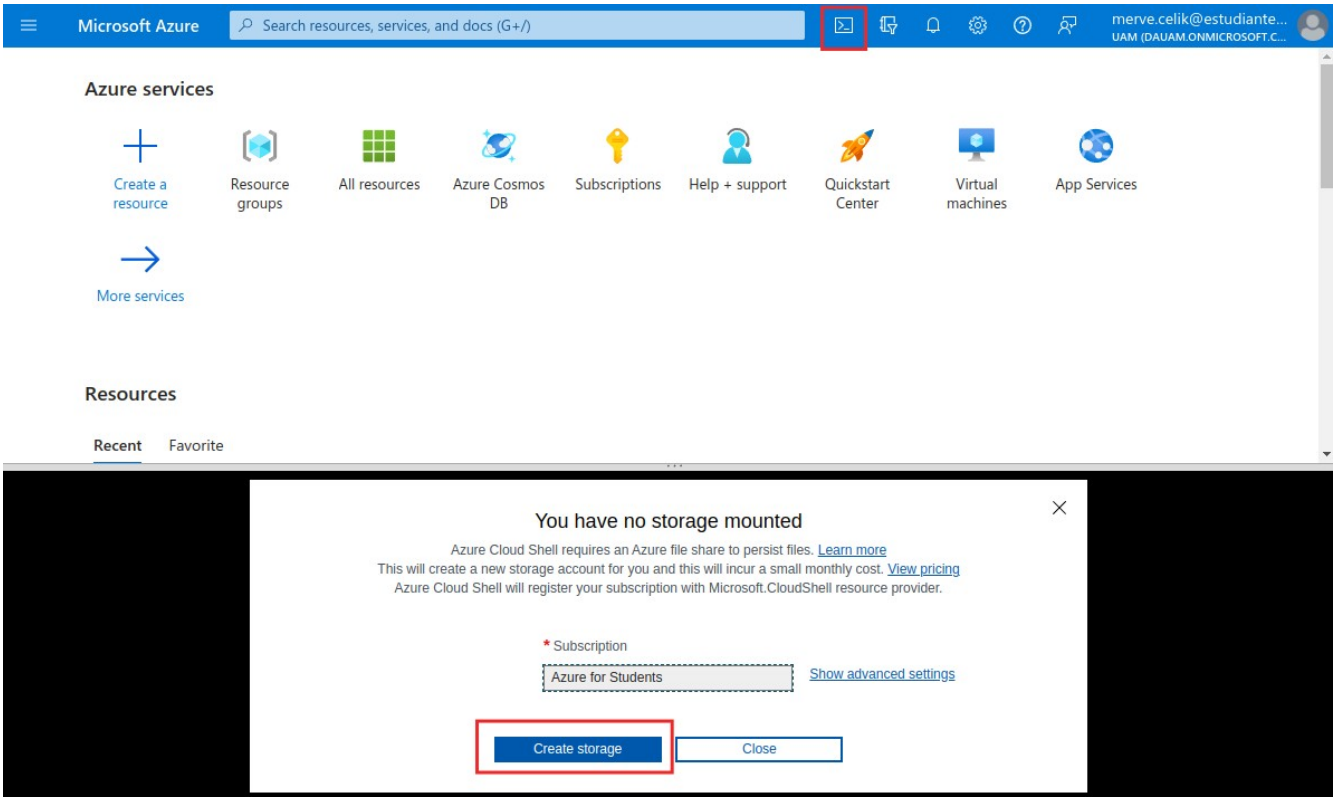
Después de pulsar sobre “Scan QR Code” y sobre “Connect manually”, pegamos la Primary connection string en el campo que se muestra al elegir “IoT Hub Device connection string”. Pulsamos sobre “Connect”. Visualizamos el dashboard con diferentes valores de sensores y otras fuentes de información del teléfono móvil.





5.8. Visualización datos simulados en Azure

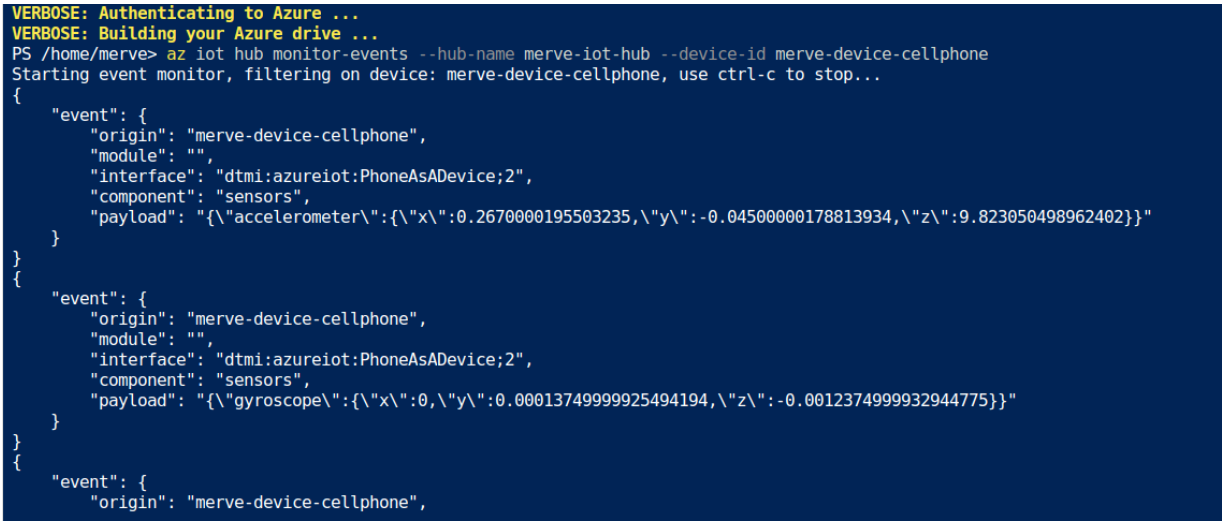
Volvemos a Azure Portal y pulsamos sobre el botón de “Cloud Shell”. Como no tenemos ningún almacenamiento montado, debemos pulsar sobre el botón “Create storage”.

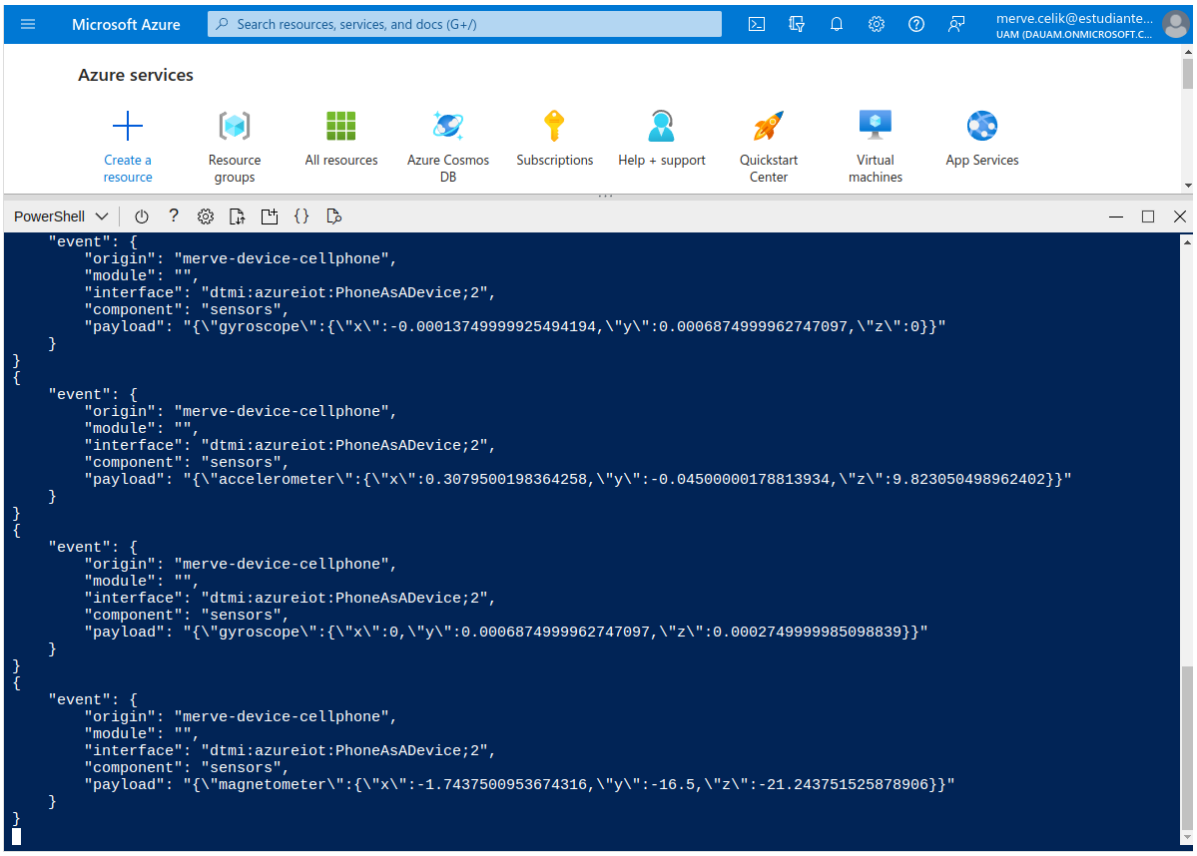


Ejecutamos el siguiente comando en la línea de comandos:

az iot hub monitor-events --hub-name merve-iot-hub --device-id merve-device-cellphone

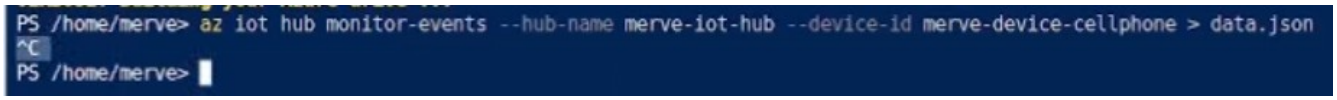
Ese comando se utiliza para monitorear los eventos del dispositivo 'merve-device-cellphone' en el IoT Hub 'merve-iot-hub'. Nos permite visualizar en tiempo real los eventos que se están generando desde ese dispositivo, ya que está conectado con el móvil usando la aplicación IoT Plug and Play. De esta manera, podemos ver el dashboard con diferentes valores de sensores y otras fuentes de información del teléfono móvil.



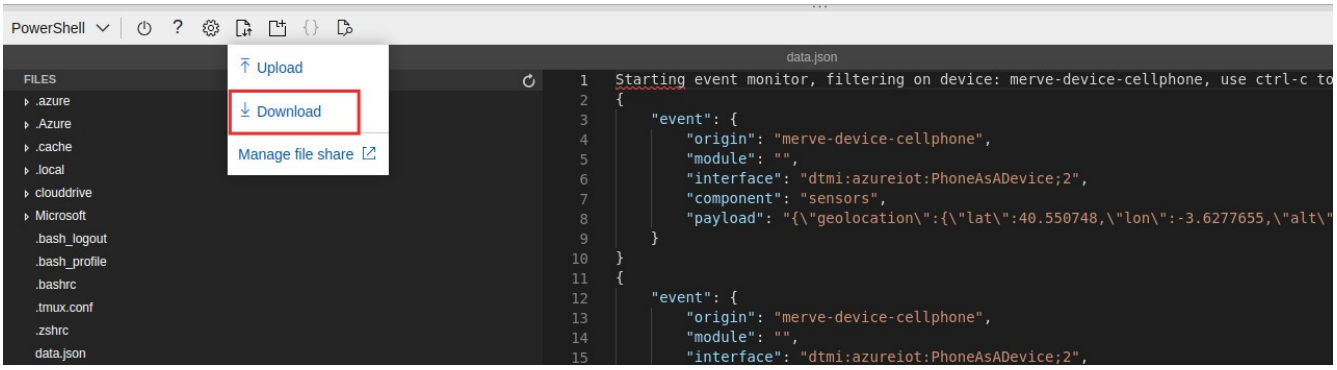


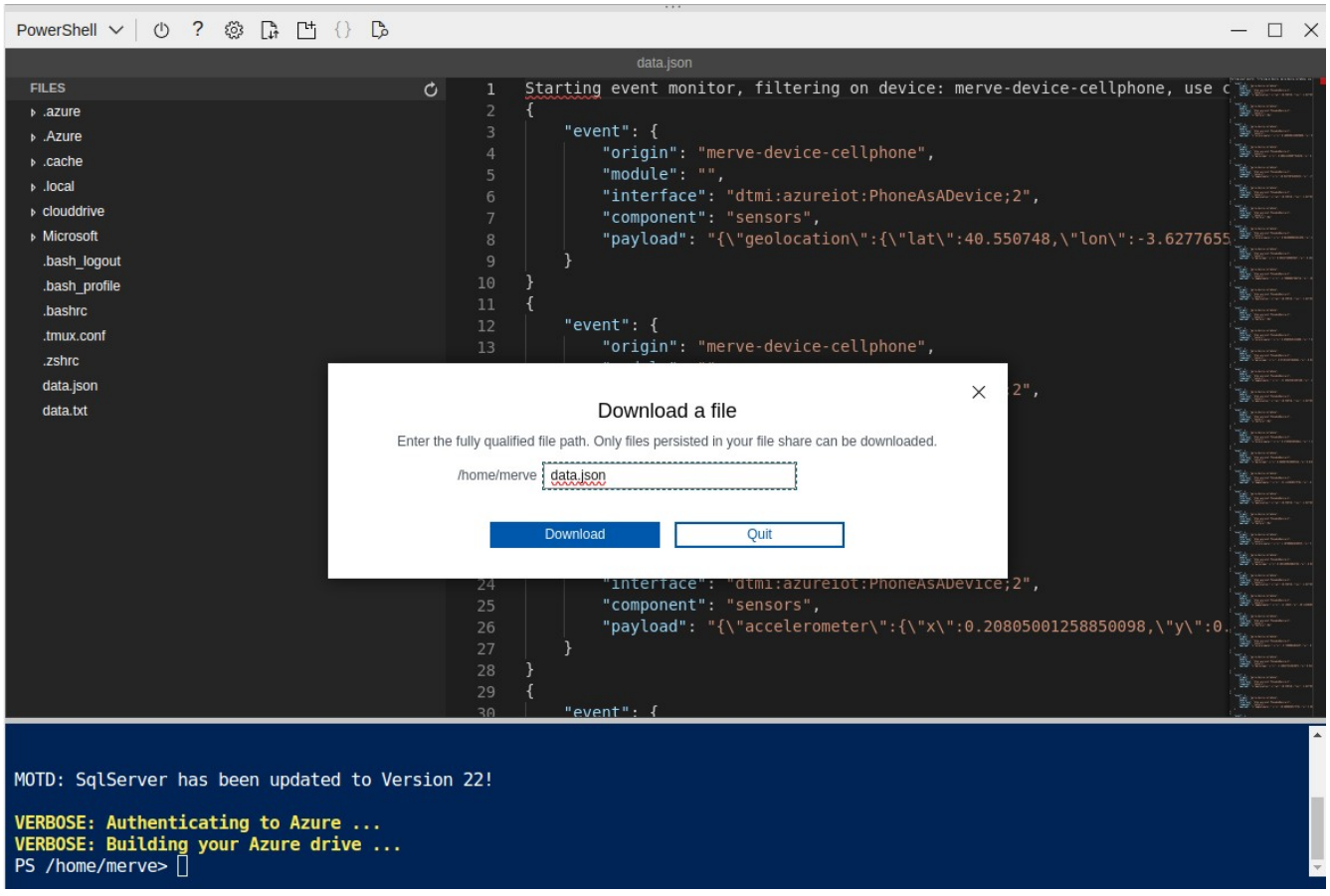
Para guardar en nuestro ordenador los datos que hemos visualizado, podemos ejecutar el siguiente comando:

az iot hub monitor-events --hub-name merve-iot-hub --device-id merve-device-cellphone > data.json



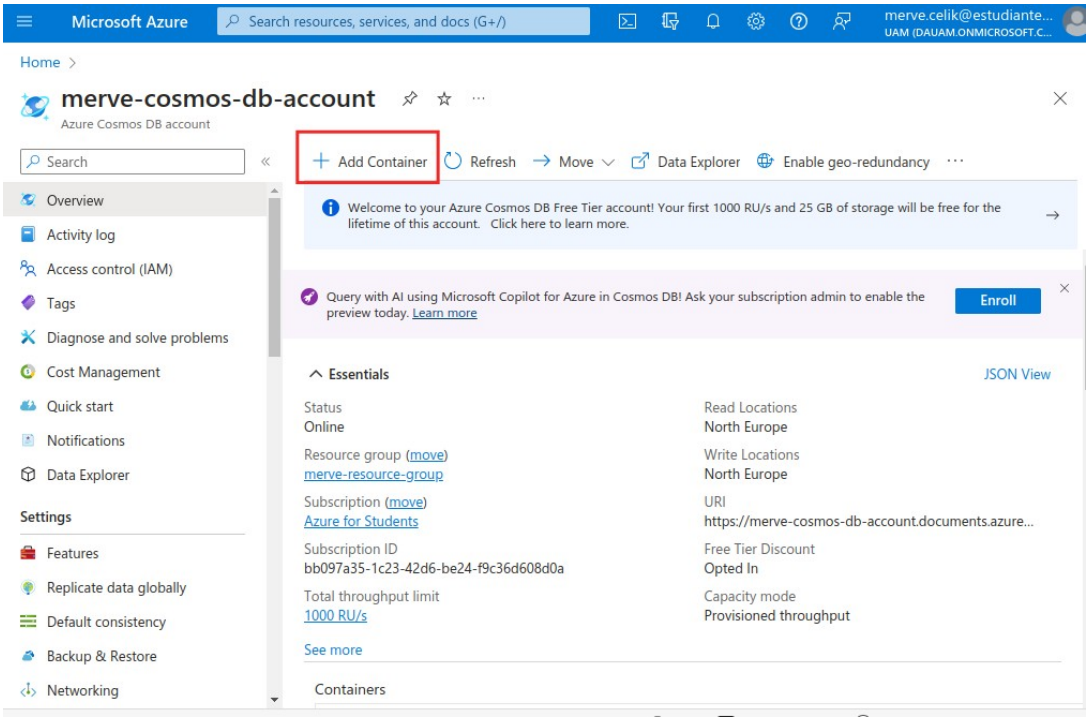
Y después de pulsar “Download”, podemos guardar los datos.

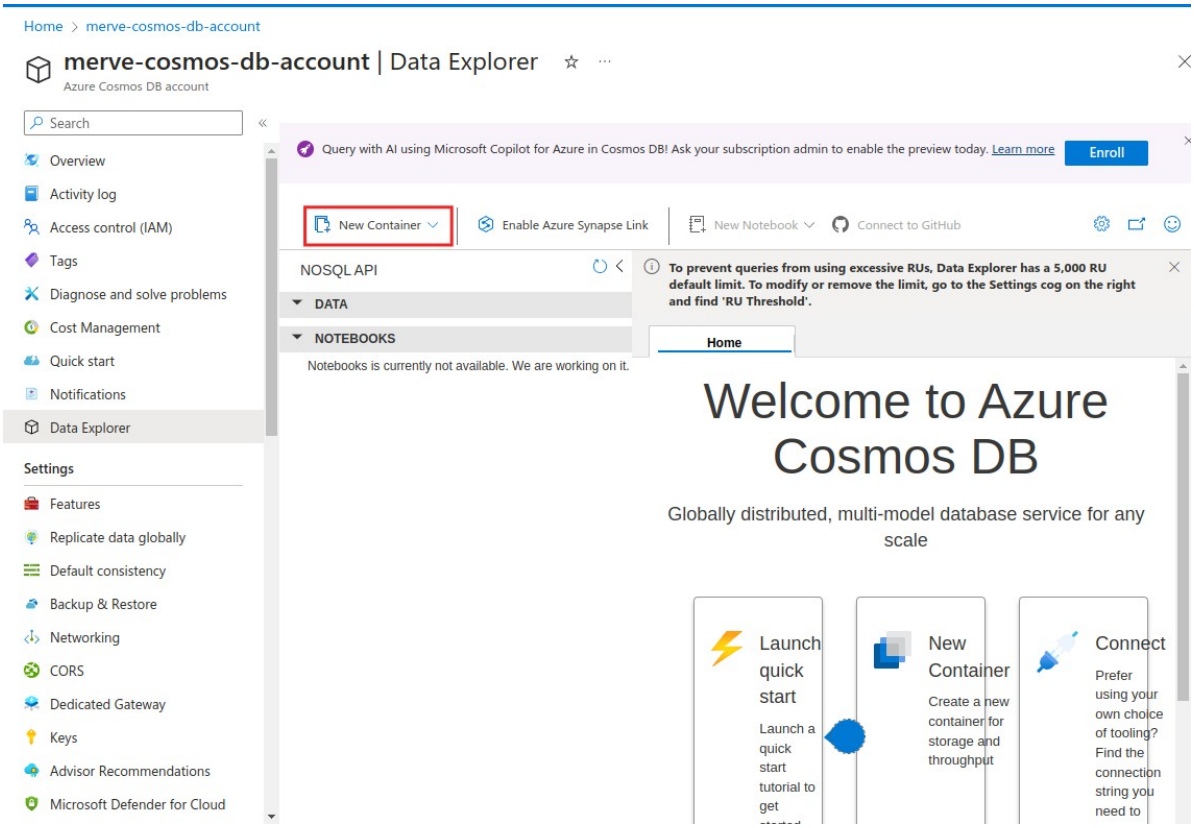




5.9. Creación de contenedor en CosmosDB

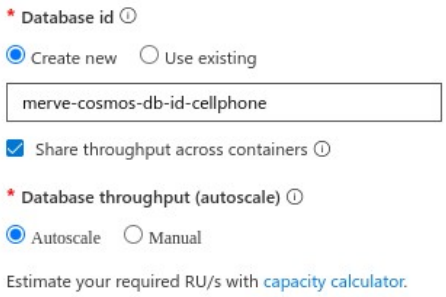
Volvemos al servicio ya desplegado de Cosmos DB
Creamos un Contenedor para albergar los datos, por lo que pulsamos en “+ Add Container” y luego “New Container”

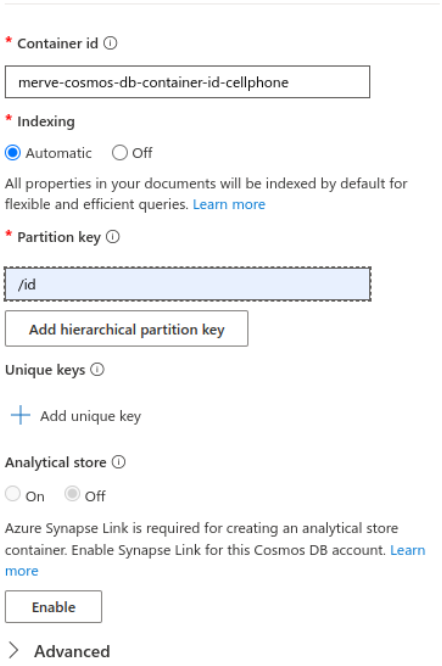




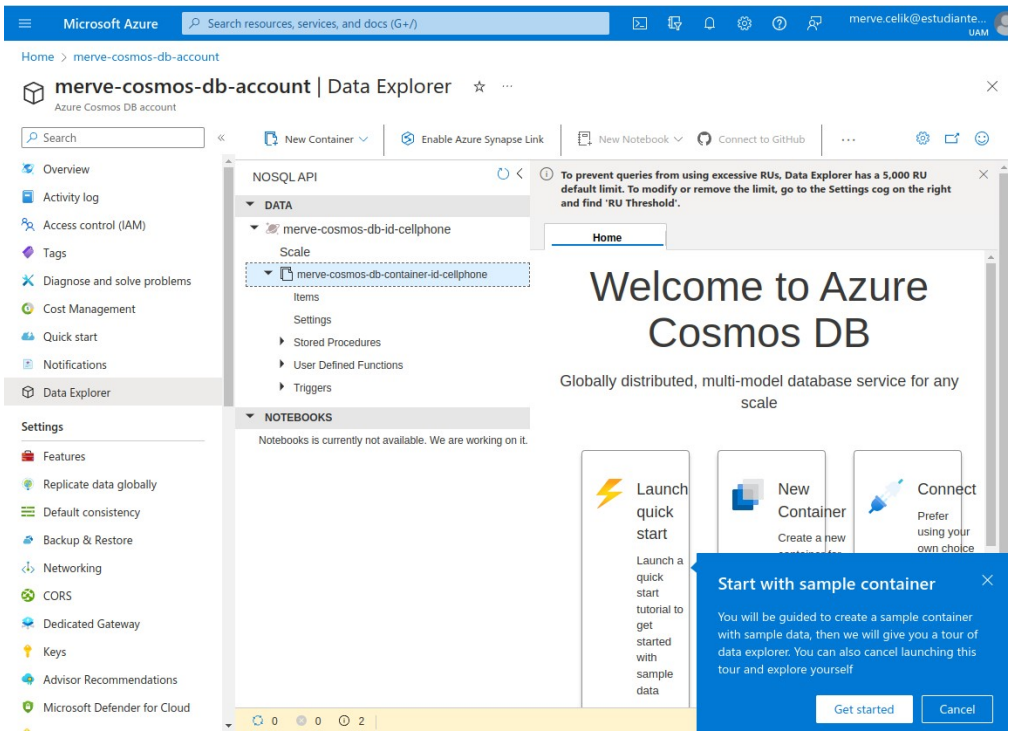
Rellenamos los campos:
Database id: merve-cosmos-db-id-cellphone
Container id: merve-cosmos-db-container-id-cellphone
Partition: /id

El resto de valores los dejamos por defecto
Hacemos scroll hacia abajo en la barra lateral y pulsamos sobre Ok



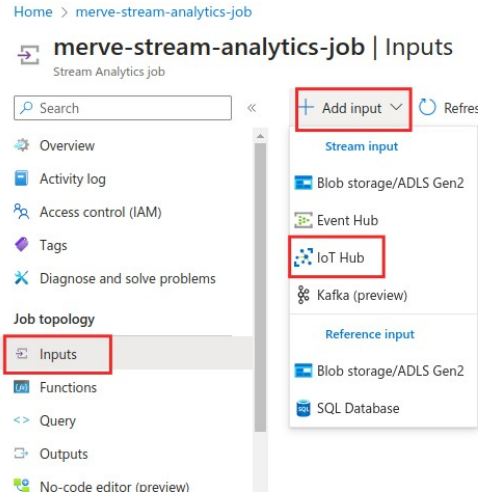


Creamos Contenedor correctamente.



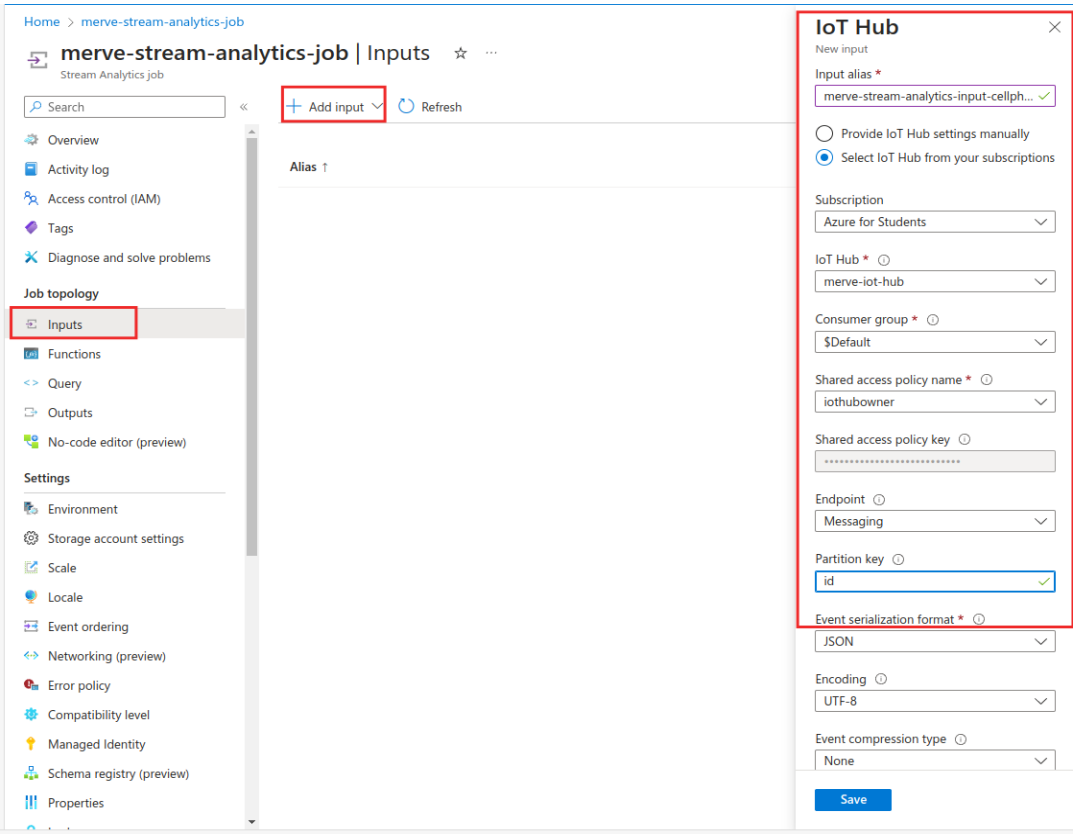
5.10. Entradas y Salidas en Stream Analytics Job

Volvemos al servicio ya desplegado de Stream Analytics Job
Creamos una entrada de datos, por lo que pulsamos en “Inputs”, luego sobre “+Add input” y en el desplegable, en “IoT Hub”:



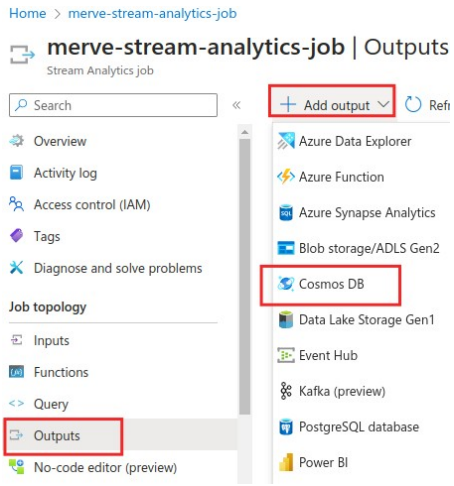
Rellenamos los campos:

Input alias: merve-stream-analytics-input-cellphone
Subscription: Azure for Students
Shared Access policy name: iothubowner
Partition key: id
El resto de valores los dejamos por defecto y pulsamos en “Save”

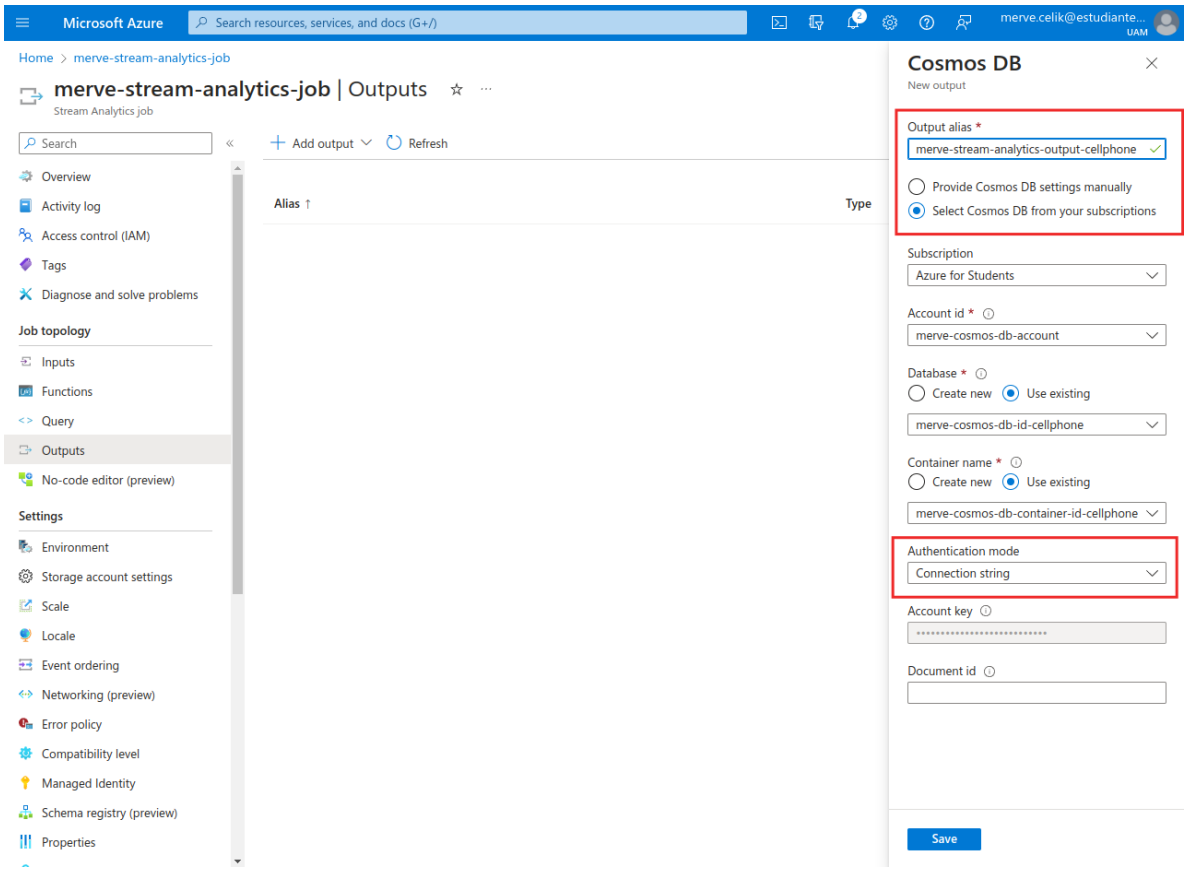


Ahora creamos una salida de datos, por lo que pulsamos en “Outputs” en la venta principal del servicio de Stream Analytics Jobs.

Luego pulsamos sobre “+Add output” y en el desplegable, en “Cosmos DB”:



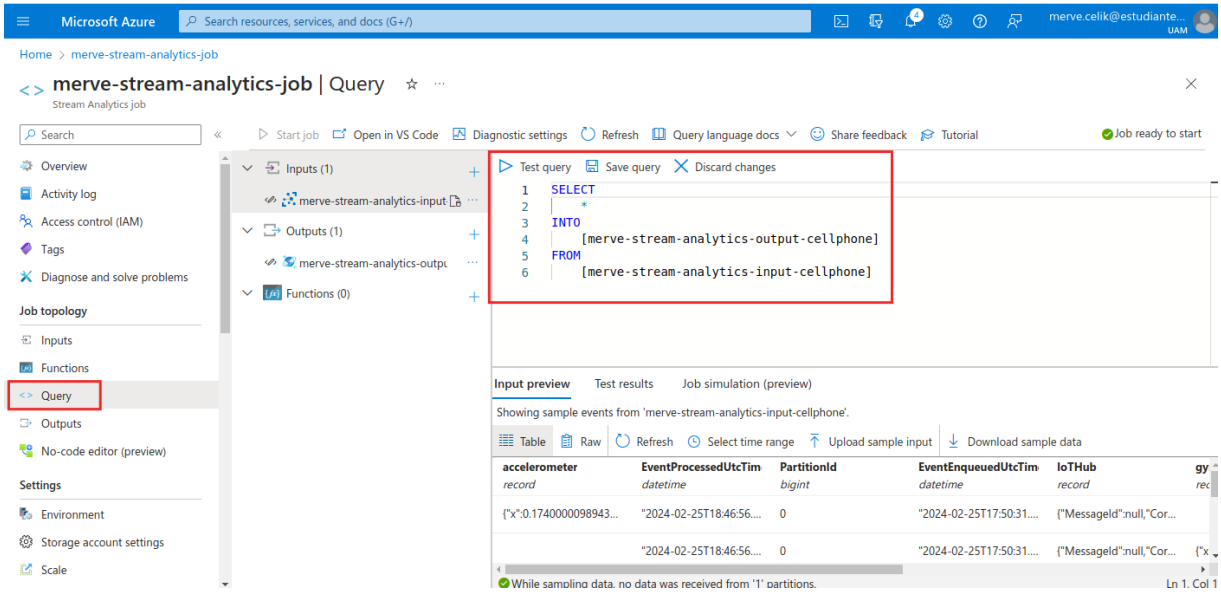
Rellenamos los campos:
Output alias: merve-stream-analytics-output-cellphone
Authentication mode: Connection string
El resto de valores los dejamos por defecto y pulsamos en Save



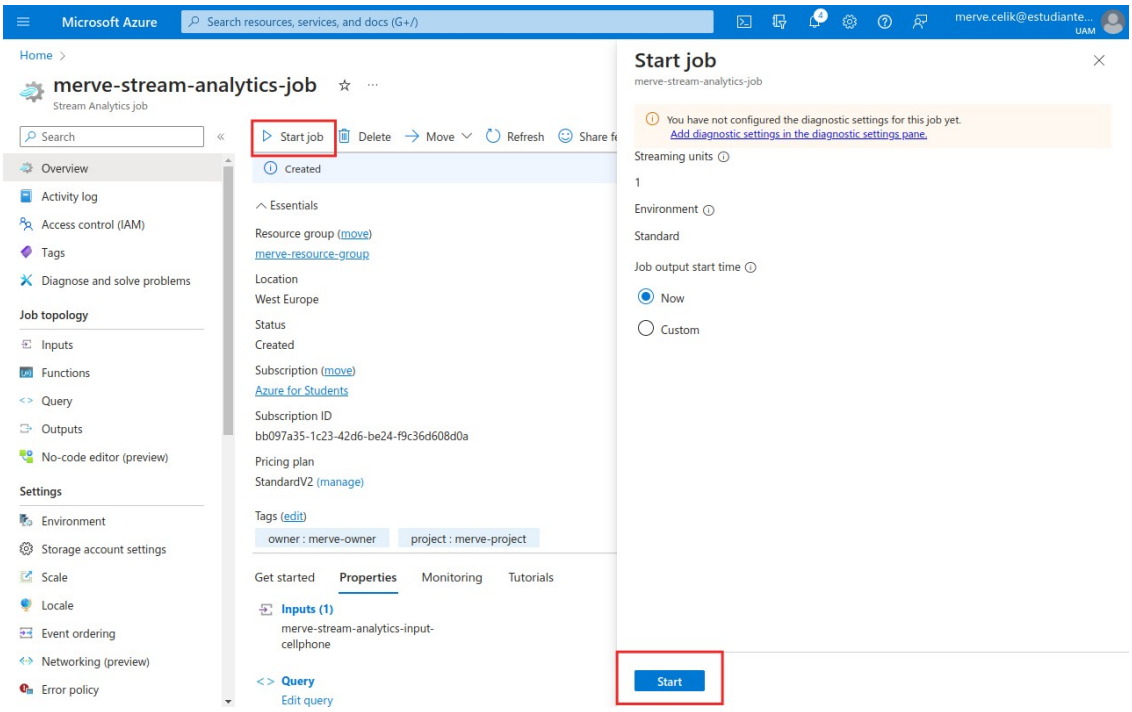
5.11. Consulta en Stream Analytics Job

En la barra lateral del servicio Stream Analytics Job, donde gestionamos las entradas y salidas, seleccionamos “Query”. Luego, ajustamos la consulta, reemplazando los campos

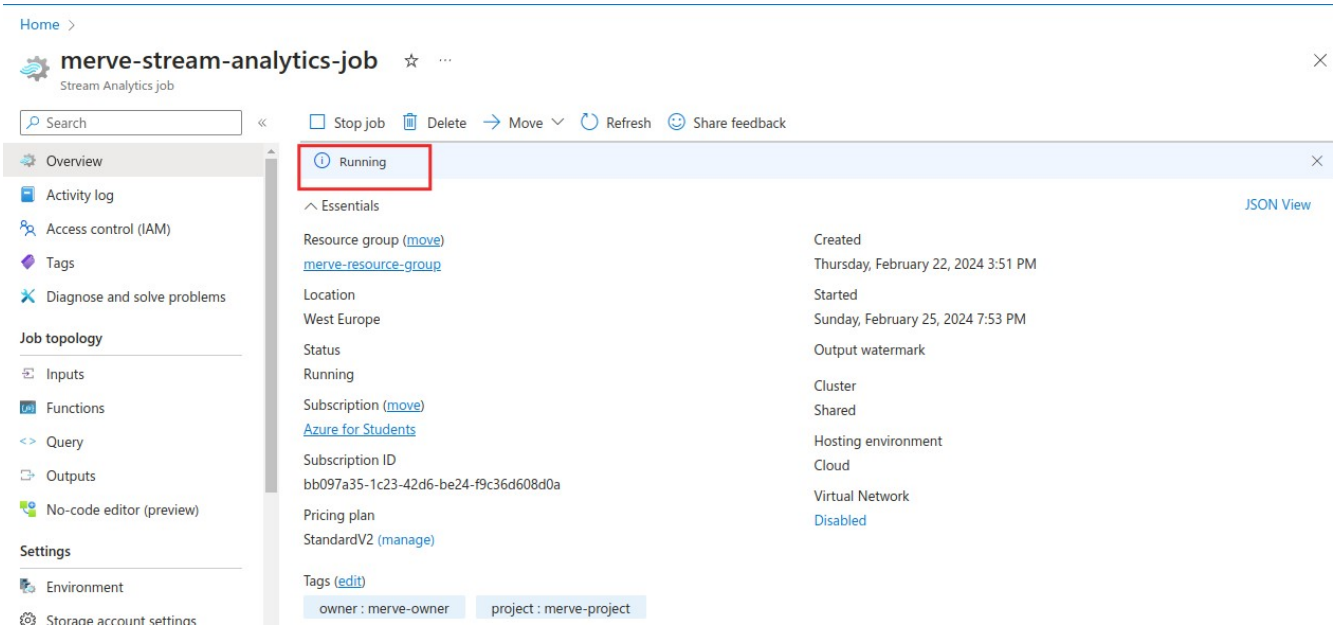
MyOutputAlias: merve-stream-analytics-output-cellphone
MyInputAlias: merve-stream-analytics-input-cellphone
Finalmente, guardamos la consulta (Save Query).



Vamos a la ventana de Overview dentro de Stream Analytics Jobs, en la barra lateral. Iniciamos el Job, por lo que pulsamos en “Start”.

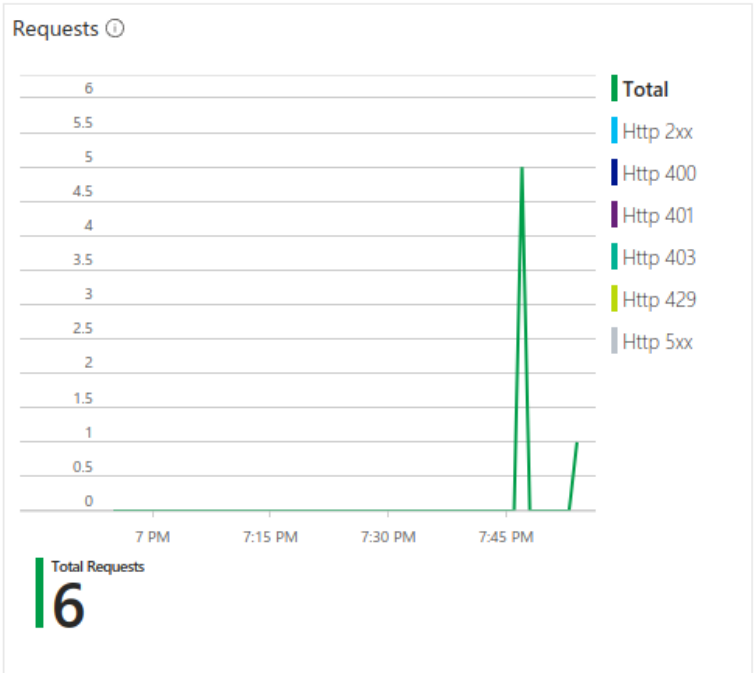


Esperamos a que Job se inicie y observamos su estado hasta que cambie a “Running".

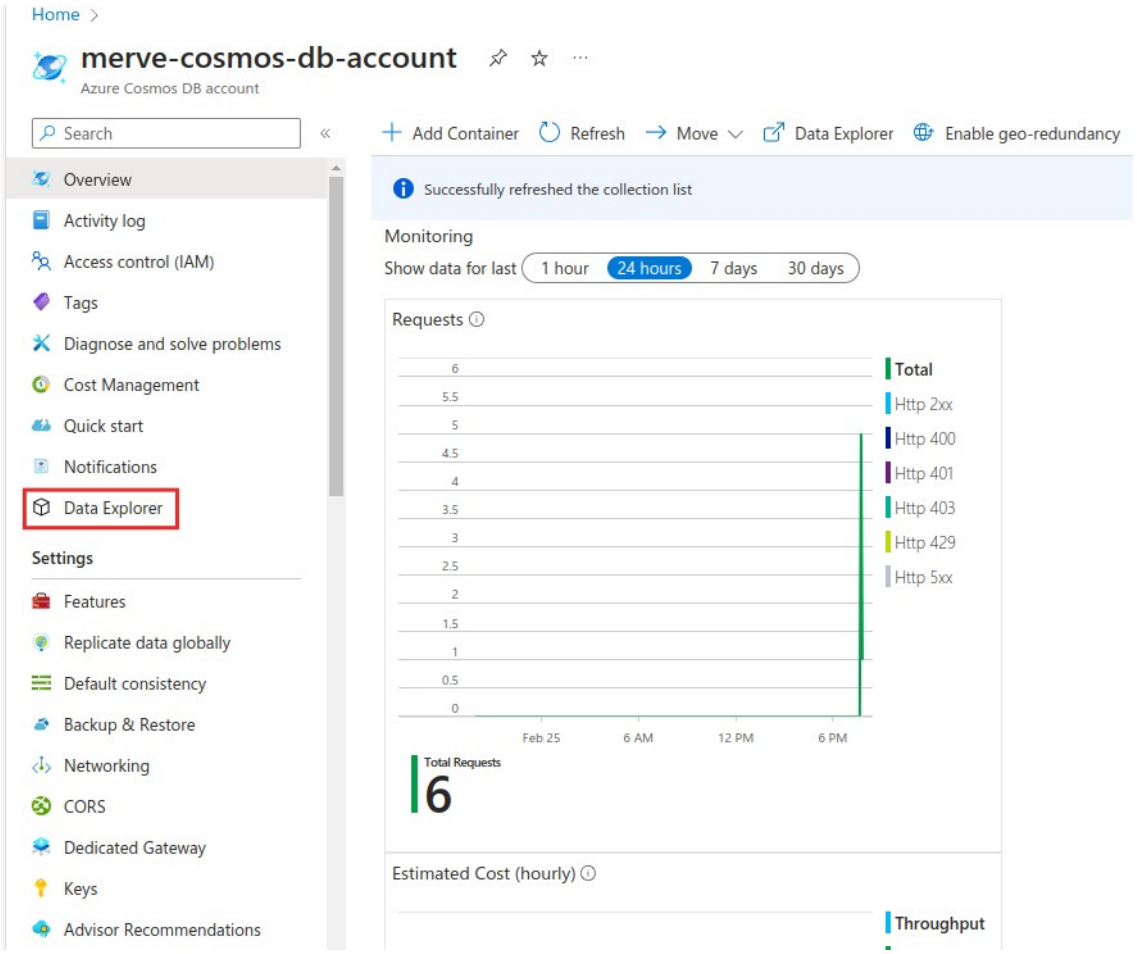


5.12. Exploración de los datos

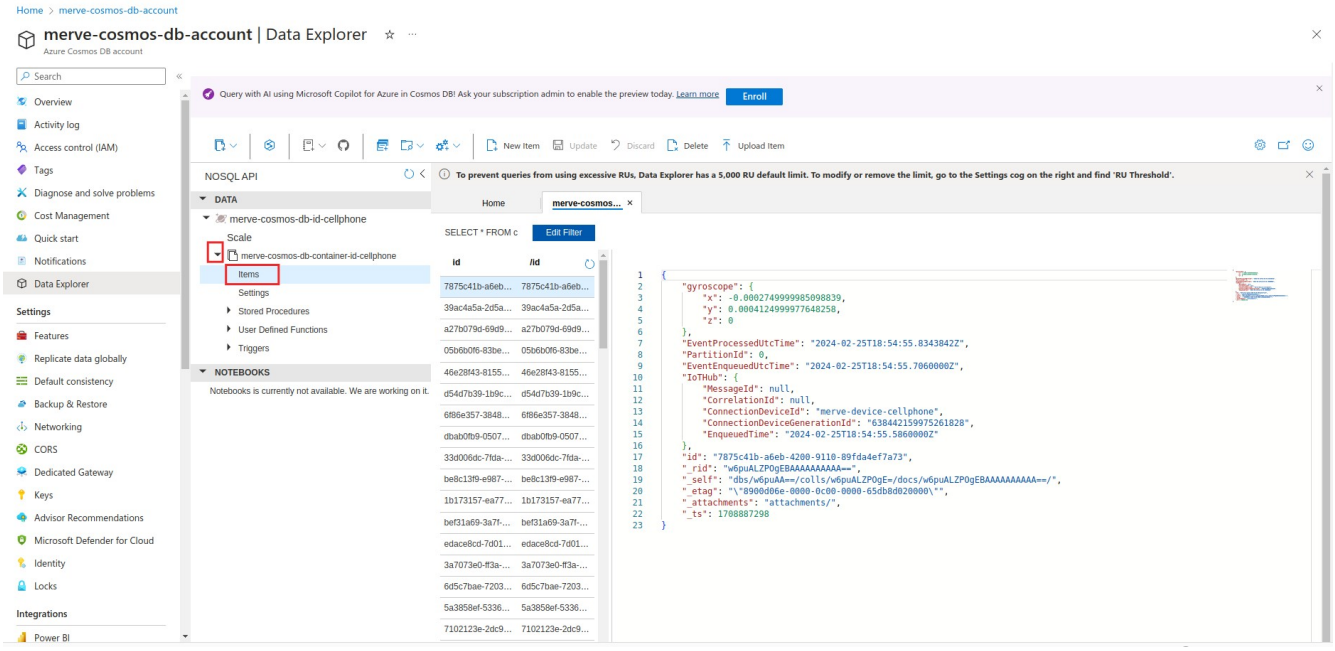
Volvemos al servicio de Cosmos DB.
Podemos ver que ya entran datos en el dashboard de Monitorización



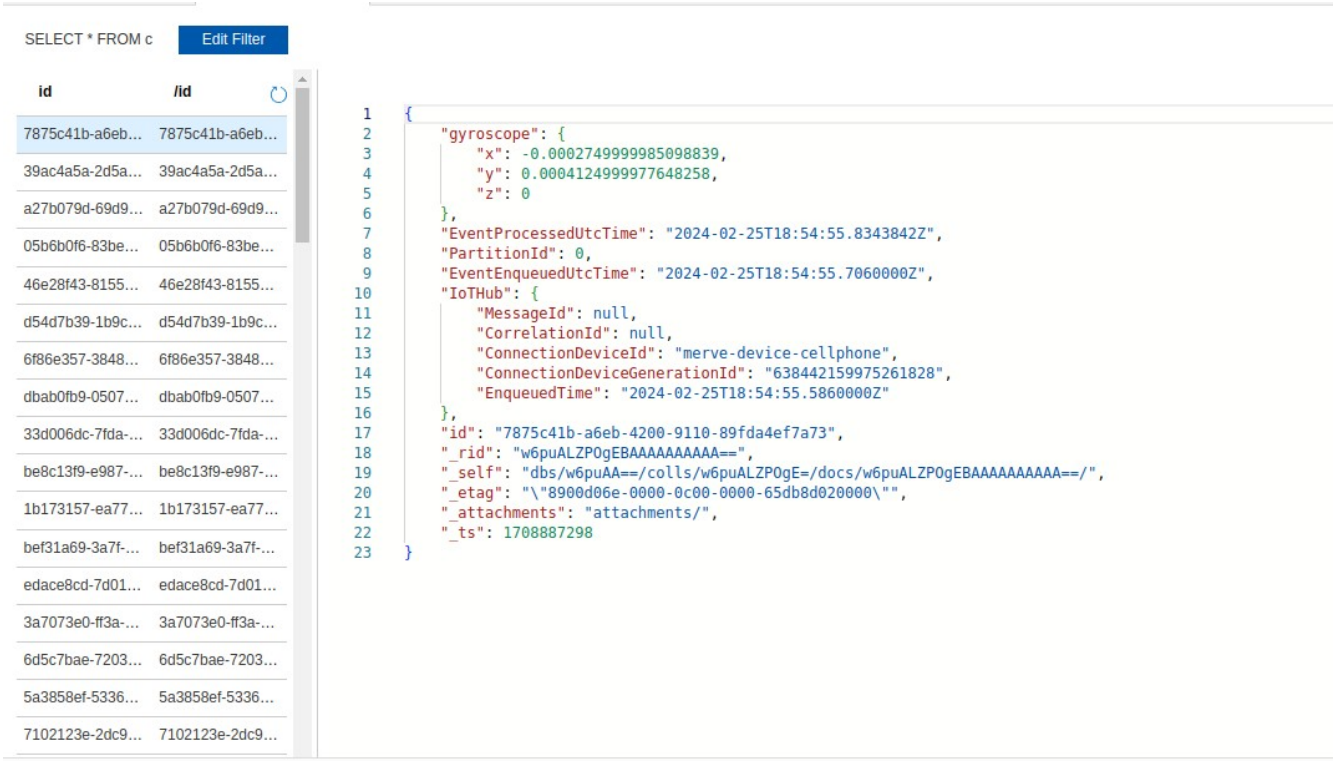
Pulsamos sobre Data Explorer:

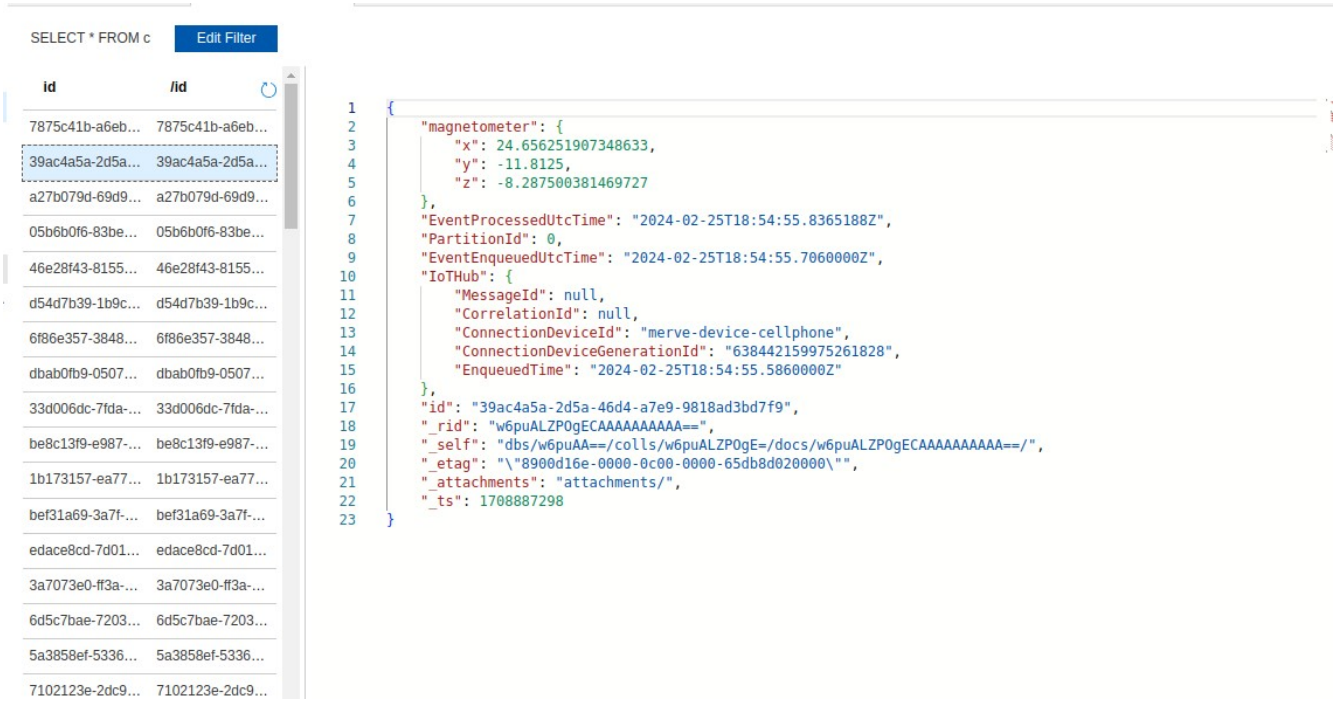


Pulsamos sobre la flecha junto al contenedor previamente creado y luego sobre “Items”.



Podemos ver los mensajes enviados por el dispositivo pulsando sobre cualquier ID por el que hemos segmentado nuestra base de datos NoSQL,





SELECT * FROM c Edit Filter

id	/id
7875c41b-a6eb...	7875c41b-a6eb...
39ac4a5a-2d5a...	39ac4a5a-2d5a...
a27b079d-69d9...	a27b079d-69d9...
05b6b0f6-83be...	05b6b0f6-83be...
46e28f43-8155...	46e28f43-8155...
d54d7b39-1b9c...	d54d7b39-1b9c...
6f86e357-3848...	6f86e357-3848...
dbab0fb9-0507...	dbab0fb9-0507...
33d006dc-7fda-...	33d006dc-7fda-...
be8c13f9-e987-...	be8c13f9-e987-...
1b173157-ea77...	1b173157-ea77...
bef31a69-3a7f-...	bef31a69-3a7f-...
edace8cd-7d01...	edace8cd-7d01...
3a7073e0-ff3a-...	3a7073e0-ff3a-...
6d5c7bae-7203...	6d5c7bae-7203...
5a3858ef-5336...	5a3858ef-5336...
7102123e-2dc9...	7102123e-2dc9...


```
{
    "accelerometer": {
        "x": 0.2749500274658203,
        "y": 0.019050000235438347,
        "z": 9.829950332641602
    },
    "EventProcessedUtcTime": "2024-02-25T18:54:55.7723447Z",
    "PartitionId": 0,
    "EventEnqueuedUtcTime": "2024-02-25T18:54:55.5810000Z",
    "IoTHub": {
        "MessageId": null,
        "CorrelationId": null,
        "ConnectionDeviceId": "merve-device-cellphone",
        "ConnectionDeviceGenerationId": "638442159975261828",
        "EnqueuedTime": "2024-02-25T18:54:55.4900000Z"
    },
    "id": "a27b079d-69d9-4cbe-bcdc-a9d3ca456532",
    "_rid": "w6puALZPOgEDAAAAAAAAAA==",
    "_self": "dbs/w6puAA==/colls/w6puALZPOgE=/docs/w6puALZPOgEDAAAAAAAAAA==/",
    "_etag": "\"8900d26e-0000-0c00-0000-65db8d020000\"",
    "_attachments": "attachments/",
    "_ts": 1708887298
}
```


SELECT * FROM c Edit Filter

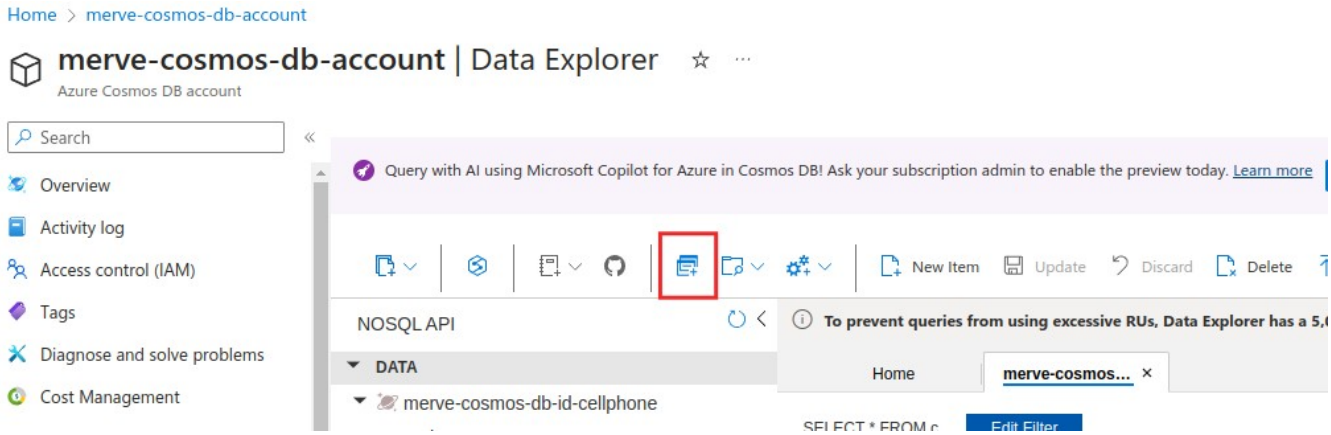
id	/id
7875c41b-a6eb...	7875c41b-a6eb...
39ac4a5a-2d5a...	39ac4a5a-2d5a...
a27b079d-69d9...	a27b079d-69d9...
05b6b0f6-83be...	05b6b0f6-83be...
46e28f43-8155...	46e28f43-8155...
d54d7b39-1b9c...	d54d7b39-1b9c...
6f86e357-3848...	6f86e357-3848...
dbab0fb9-0507...	dbab0fb9-0507...
33d006dc-7fda-...	33d006dc-7fda-...
be8c13f9-e987-...	be8c13f9-e987-...
1b173157-ea77...	1b173157-ea77...
bef31a69-3a7f-...	bef31a69-3a7f-...
edace8cd-7d01...	edace8cd-7d01...
3a7073e0-ff3a-...	3a7073e0-ff3a-...
6d5c7bae-7203...	6d5c7bae-7203...
5a3858ef-5336...	5a3858ef-5336...
7102123e-2dc9...	7102123e-2dc9...


```
{
    "battery": 49,
    "EventProcessedUtcTime": "2024-02-25T18:54:57.5448169Z",
    "PartitionId": 0,
    "EventEnqueuedUtcTime": "2024-02-25T18:54:57.4710000Z",
    "IoTHub": {
        "MessageId": null,
        "CorrelationId": null,
        "ConnectionDeviceId": "merve-device-cellphone",
        "ConnectionDeviceGenerationId": "638442159975261828",
        "EnqueuedTime": "2024-02-25T18:54:57.3850000Z"
    },
    "id": "05b6b0f6-83be-4b47-84d2-01874bd61294",
    "_rid": "w6puALZPOgEEAAAAAAAAAA==",
    "_self": "dbs/w6puAA==/colls/w6puALZPOgE=/docs/w6puALZPOgEEAAAAAAAAAA==/",
    "_etag": "\"8900d36e-0000-0c00-0000-65db8d030000\"",
    "_attachments": "attachments/",
    "_ts": 1708887299
}
```

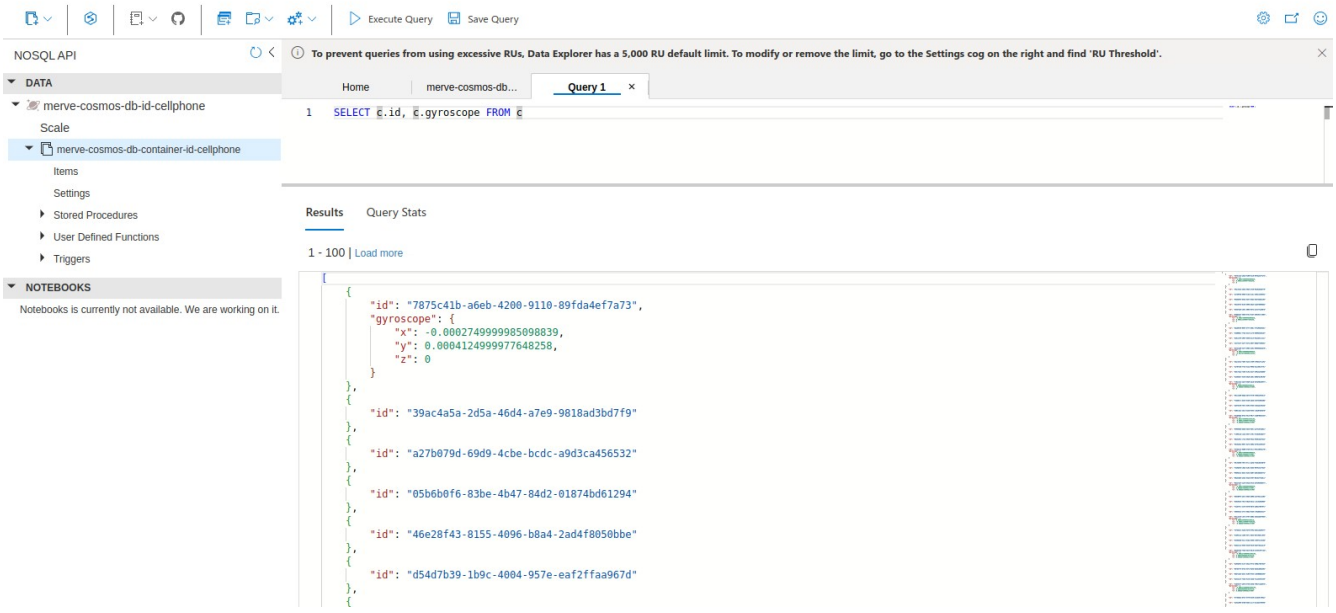
5.13. Refinando la exploración de los datos

Seguimos en el servicio de Cosmos DB, en Data Explorer:

Hacemos click en el icono “New SQL Query”

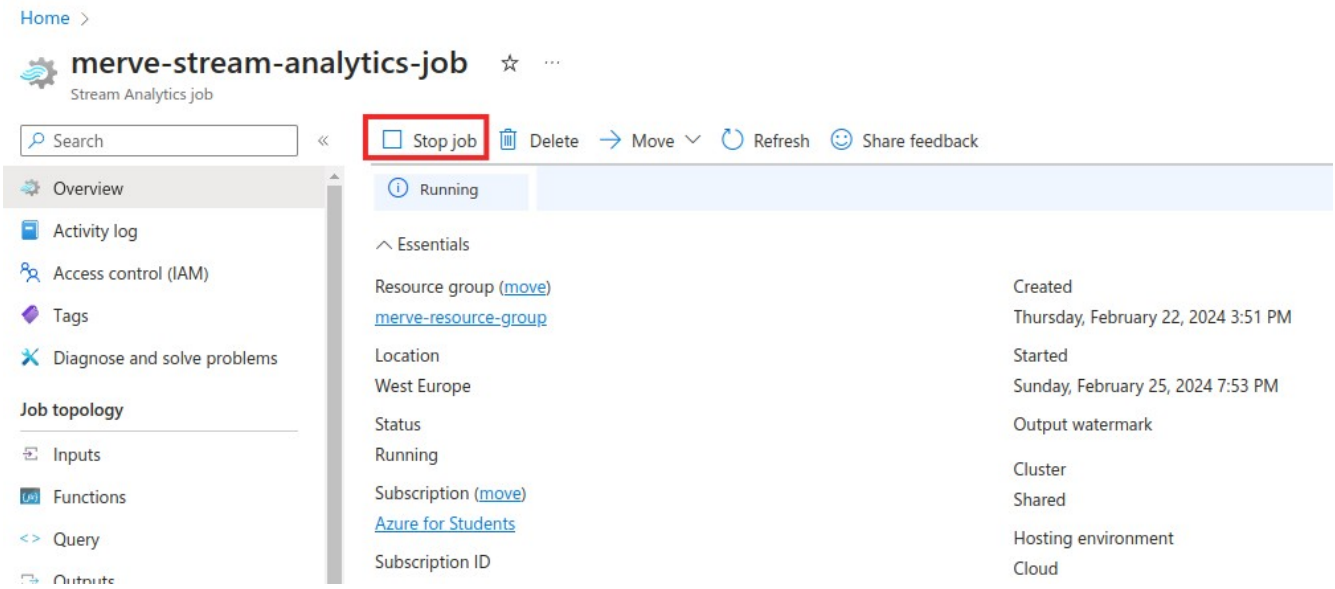


Introducimos la siguiente consulta: **SELECT c.id, c.gyroscope FROM c**
Hacemos click en “Execute Query”.
Si todo ha ido bien, veremos solo los datos de ID y giroscopio cuando se reciba el mensaje.



5.15. Detener app de datos y el Job de Stream Analytics Jobs

Para finalizar, pulsamos sobre Stop para detener la ejecución del Job en la página de Overview de Stream Analytics Jobs.



También cerramos la aplicación para detener la ejecución.

Práctica opcional – Análisis Ciencia de Datos

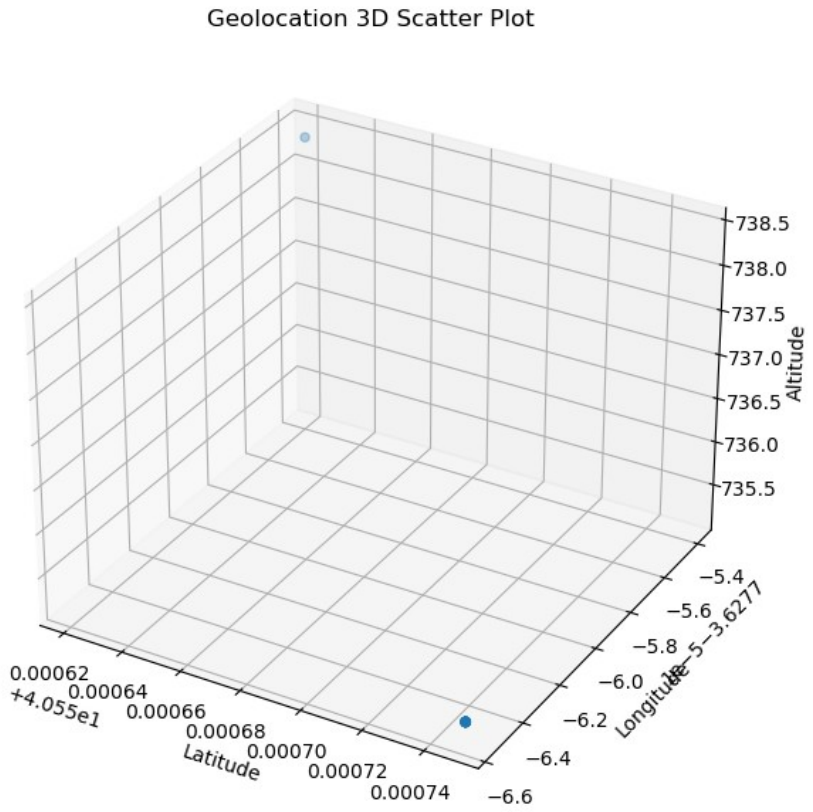
¿Qué podemos hacer con esos datos?

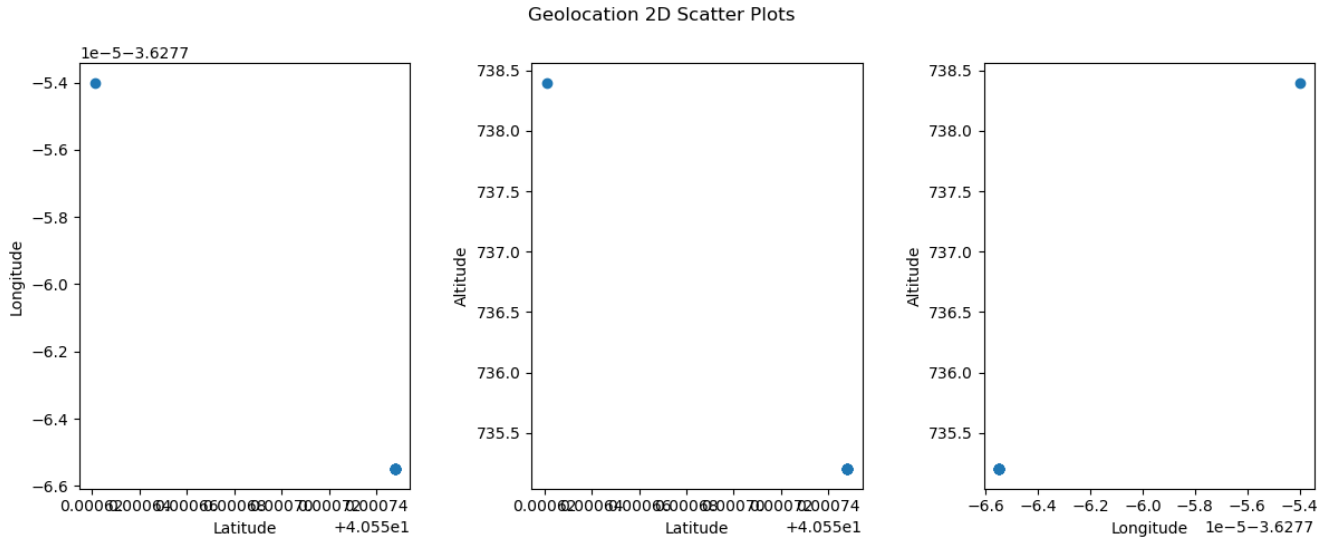
- Es posible acceder a los valores individuales de cada sensor.
- Se pueden calcular medidas estadísticas como el mínimo, el máximo, la media, la mediana o la desviación estándar para cada sensor.
- Se pueden utilizar visualizaciones, como gráficos y diagramas creados con librerías como matplotlib o seaborn, para presentar los datos de los sensores de forma informativa.

¿Representaciones gráficas?

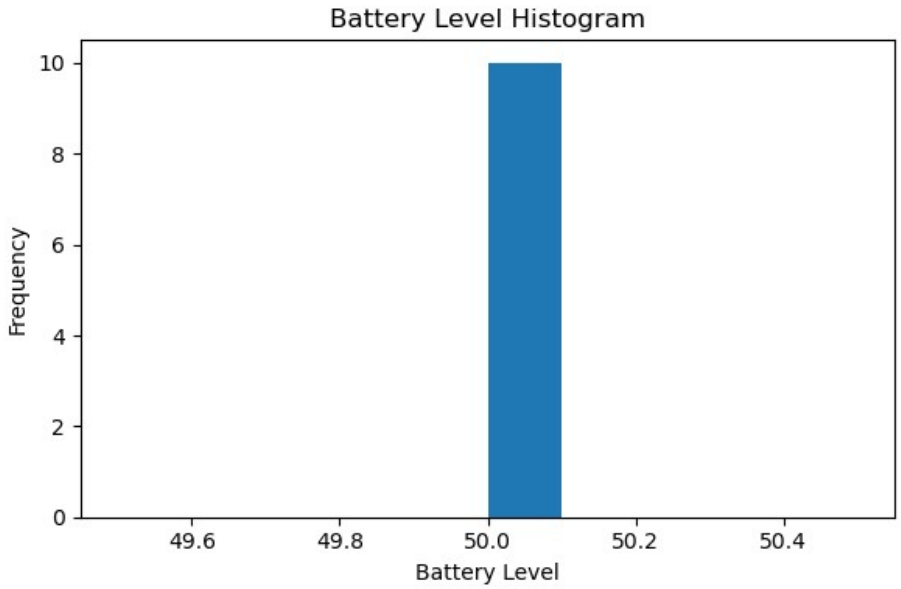
Adjunto algunos gráficos aquí.
También adjunto a la práctica el archivo de Python con el que he trabajado.

Generamos gráficos de scatter para visualizar los datos de geolocalización en dos y tres dimensiones.

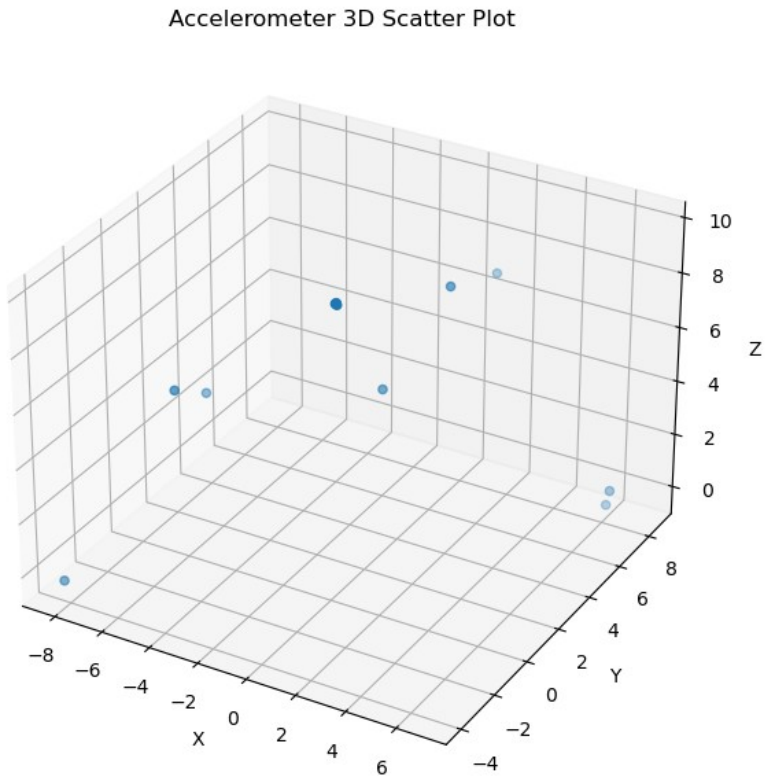


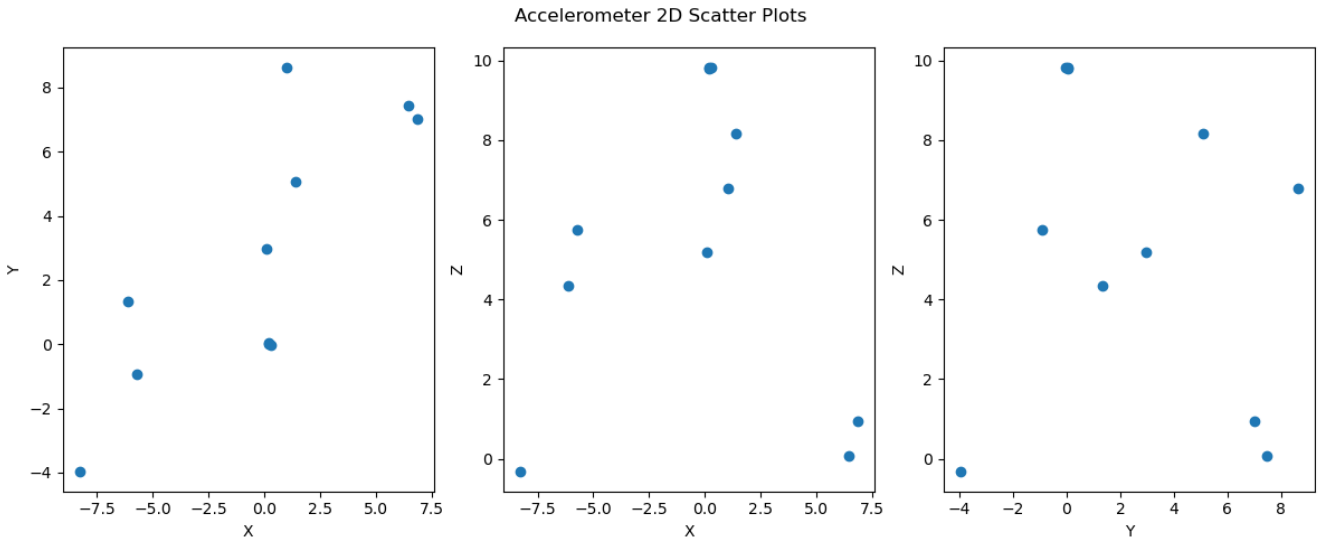


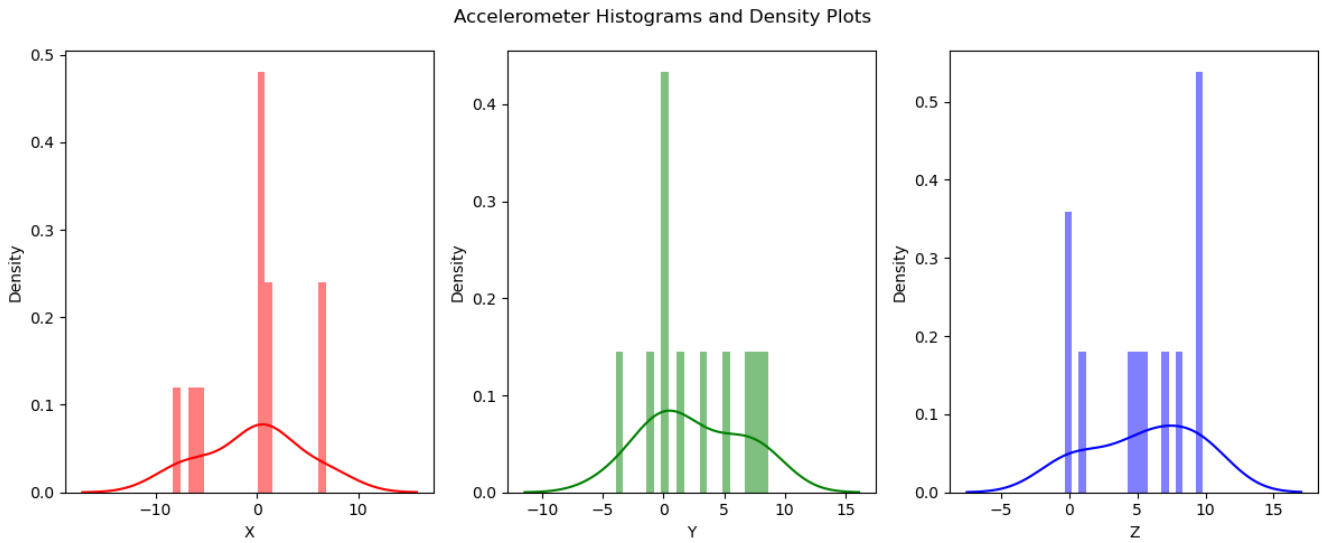
Generamos un gráfico de histograma para visualizar los cambios en el nivel de la batería a partir de los datos de la batería.



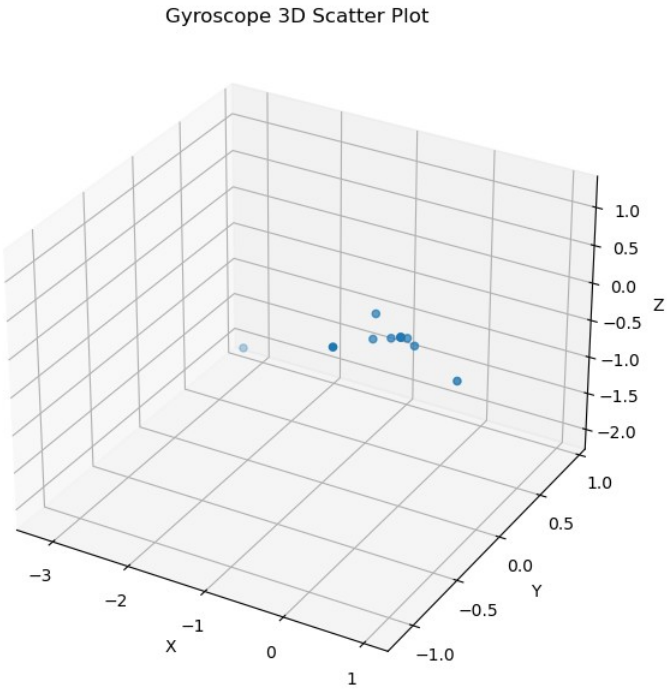
Generamos gráficos de scatter en dos y tres dimensiones para visualizar los cambios de aceleración, gráficos de histogramas y densidad para visualizar la distribución de las lecturas del acelerómetro a lo largo de cada eje (x, y, z) a partir de los datos del accelerometer.

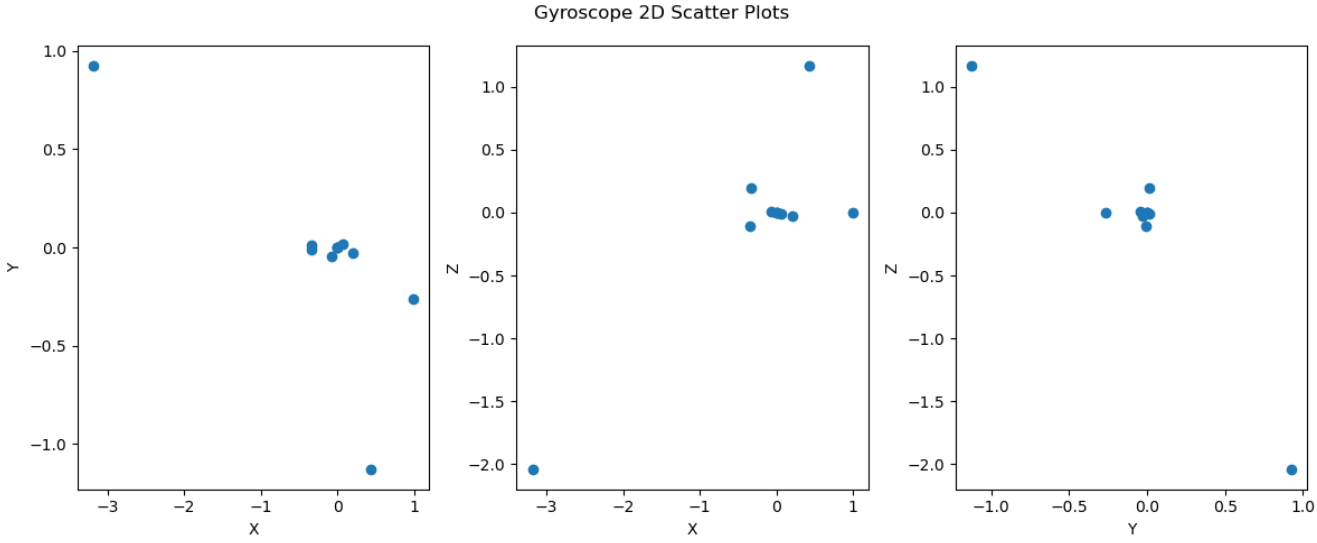


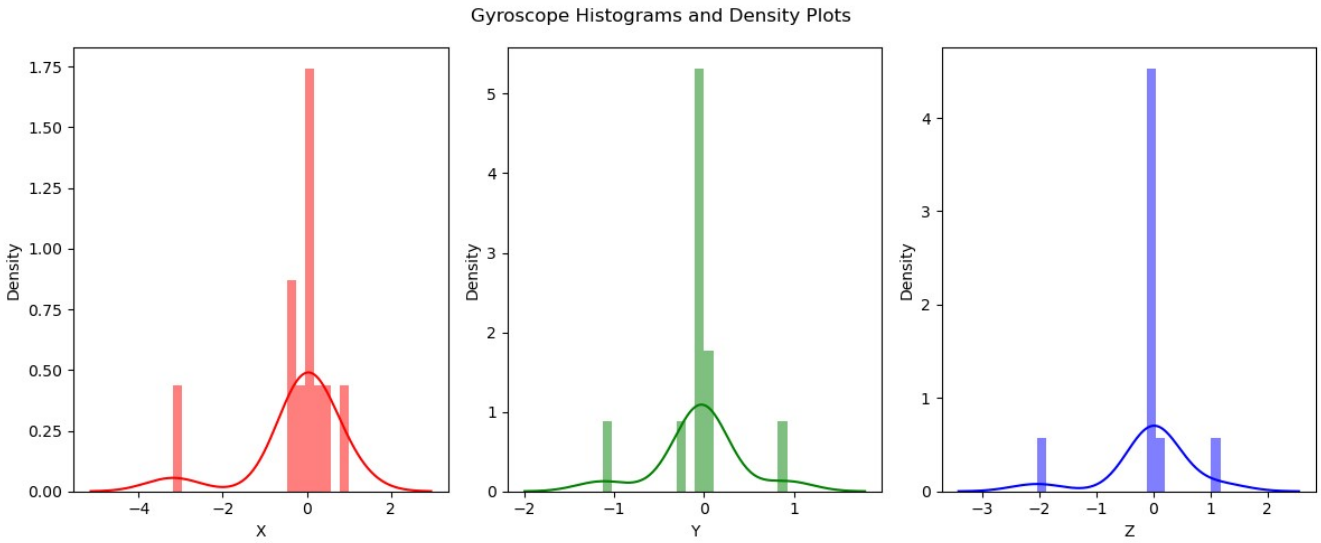

Accelerometer 2D Scatter Plots
X
Y
Z
X
Y
Z
Y


Accelerometer Histograms and Density Plots
Density
X
Density
Y
Density
Z

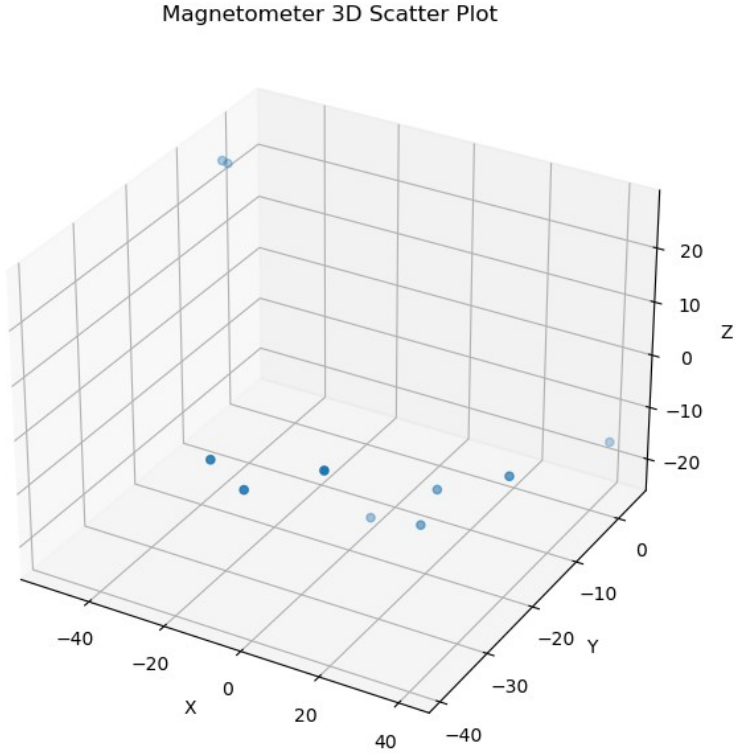
Generamos gráficos de scatter en dos y tres dimensiones para visualizar los cambios de orientación, gráficos de histogramas y densidad para visualizar la distribución de las lecturas del giroscopio a lo largo de cada eje (x, y, z) a partir de los datos del gyroscope.

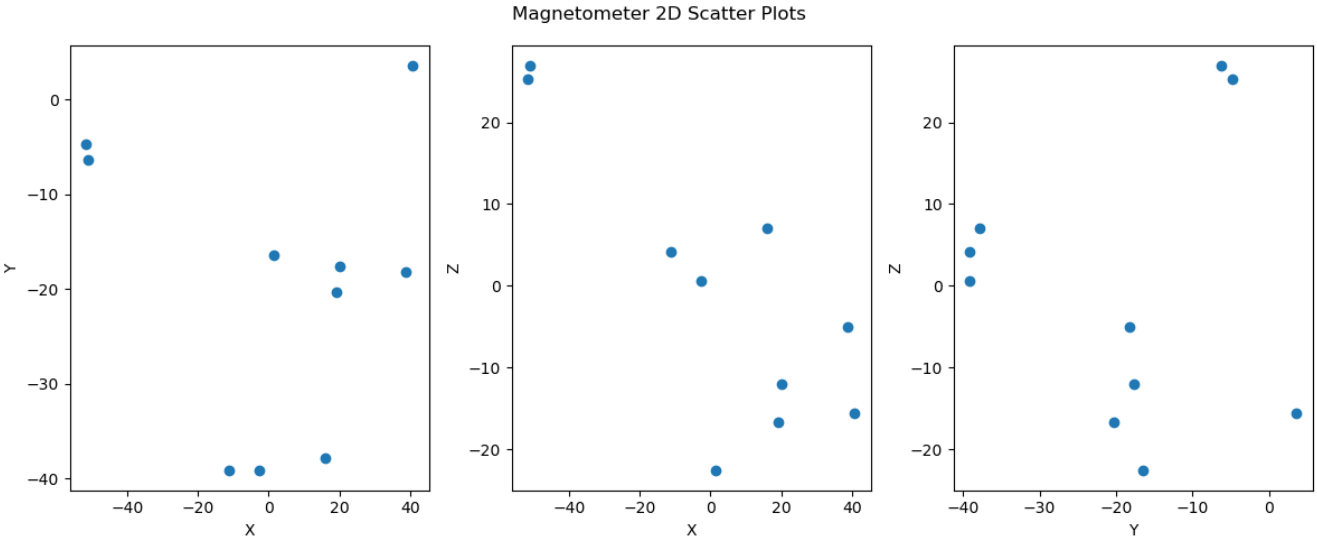


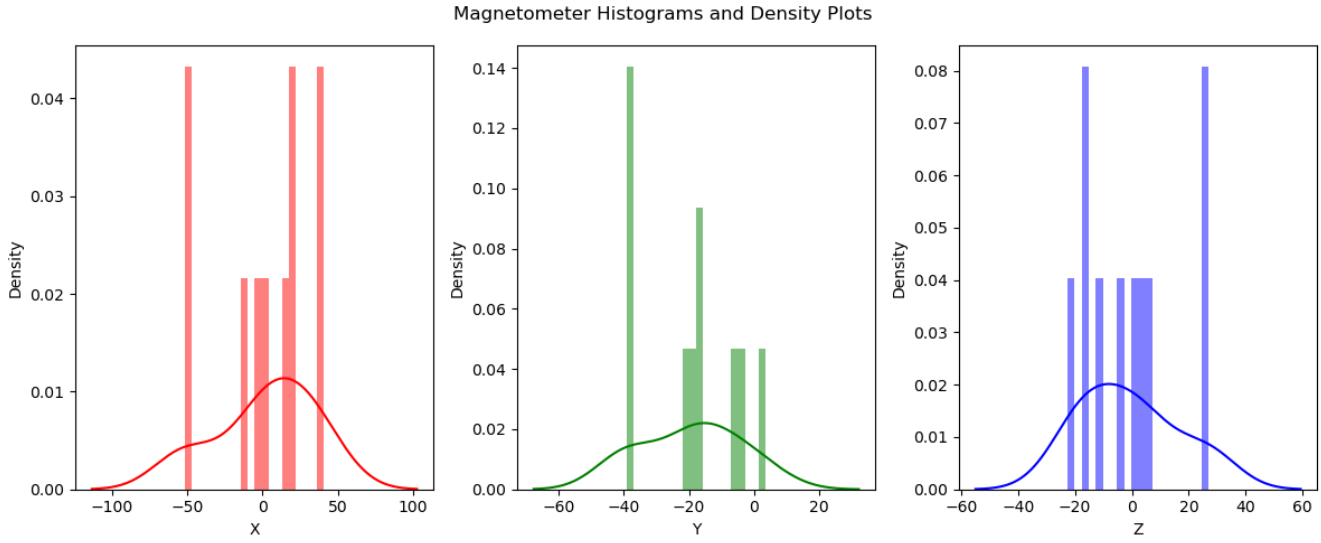




Generamos gráficos de scatter en dos y tres dimensiones para visualizar los cambios en la intensidad del campo magnético, gráficos de histogramas y densidad para visualizar la distribución de las lecturas del magnetómetro a lo largo de cada eje (x, y, z). a partir de los datos del magnetometer.




Magnetometer 2D Scatter Plots
X
Y
Z
Y


Magnetometer Histograms and Density Plots
Density
X
Y
Z