

# Índice

1. Introducción
2. Esquema general del proyecto
3. Grupo de recursos
4. Almacenamiento
5. Azure Data Factory
  - Esquema
  - Estructura
6. Azure Data Flow
7. Databricks
  - Estructura
  - Script
8. Creacion de la BDD de forma automática
  - Esquema
  - Esquema del Data Flow
  - Procedimientos
9. Gestión de errores, alertas y métricas
  - Alerta de presupuesto
  - Azure Log Analytics
  - Logic APP orientana a notificaciones
    - Métricas de hardware
    - Métricas de Spark
10. Visualización - Grafana
  - Visualizaciones totales y temas más populares
  - Distribución de sentimientos negativos
  - Distribución por geolocalización entre Minsait y Viewnext
  - Interacciones totales de Viewnext en el último año
  - Horas de actividad de NTTDATA
  - Visualización de sentimientos de Minsait por año
  - Frecuencia de tweets por mes de Hiberus
  - Actividad de TOP 5 usuarios de Hiberus

# Introducción

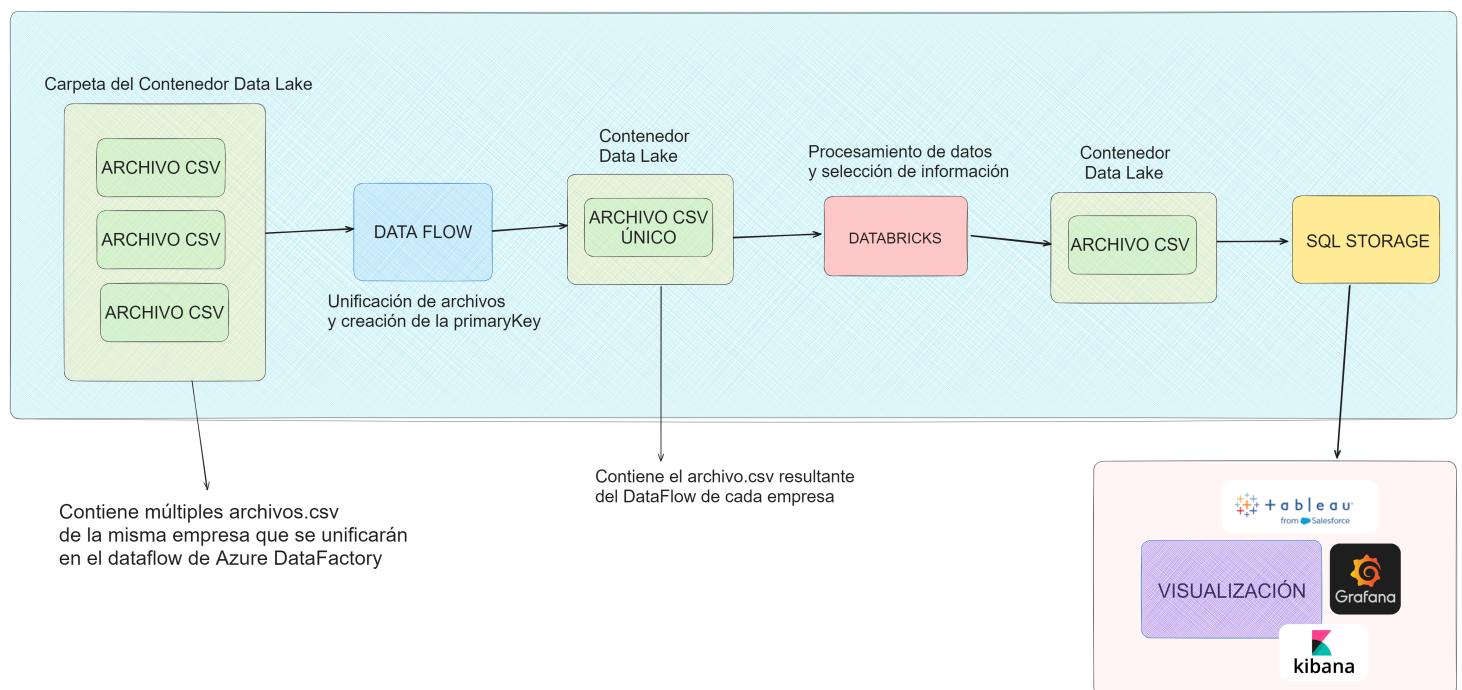
Este proyecto se ha realizado con el fin de poner en práctica los diferentes servicios que ofrece Azure. Consiste en un análisis de tweets de 4 tecnológicas españolas (Viewnext, Hiberus, Minsait y NTTDATA).

Como lenguaje de programación se ha elegido Python y también se han utilizado otras tecnologías/aplicaciones como:

- Databricks para la selección, transformación y guardado de los datos.
- Grafana para la visualización de datos.
- Azure Data Studio (gestor de BDD) para el manejo de las tablas SQL y creación de procedimientos para ejecutarlos dentro de Data Factory.

## Esquema general del proyecto

### INFRAESTRUCTURA AZURE



# Grupo de recursos

Servicio	Descripción
Cuenta de Almacenamiento Data Lake v2	Almacena la información en Azure
Databricks	Cluster de databricks para realizar ETL
Data Factory	Plataforma de orquestación de servicios y ETL
SQL	Base de datos para mantener la información en formato SQL
Logic App	Aplicación para gestión y monitorización de errores en Data Factory
Log Analytics	Servicio que guarda la información de errores, métricas y logs sobre Data Factory
Notebooks o Libros de Azure	Guardan las consultas de KQL que obtienen información de Log Analytics
Alertas personalizadas	Alertas sobre métricas, control de presupuesto y errores en Data Factory

## Almacenamiento

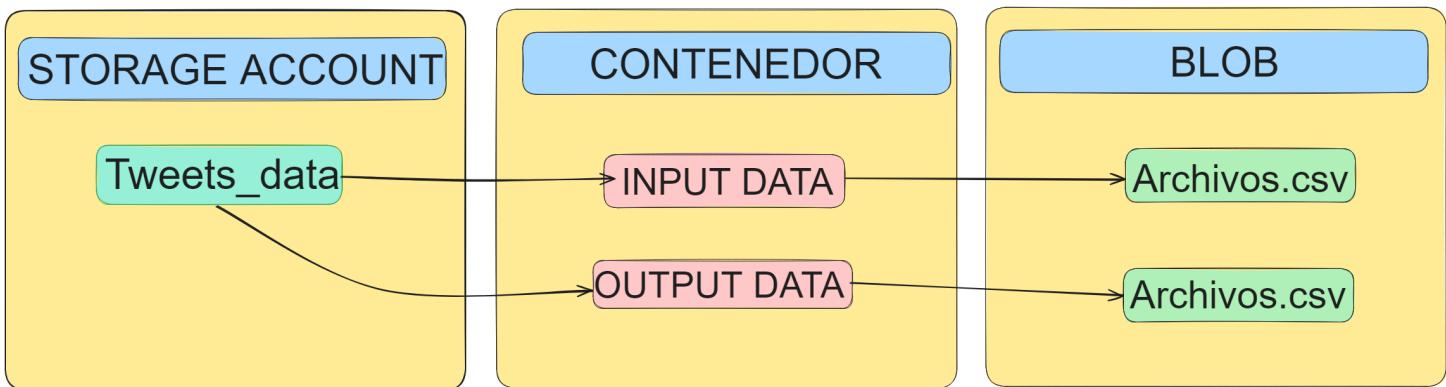
### ¿Qué es Data Lake Gen2?

Se trata de un servicio de almacenamiento optimizado donde se pueden cargar grandes volúmenes de datos para el análisis a gran escala.

Se caracteriza por no depender de un esquema de datos, por lo que da mucha flexibilidad para migrar cualquier tipo servicio/aplicación, y permite gestionar la información en contenedores blob (objeto binario grande), tablas y colas.

### Estructura

El proyecto contiene 2 contenedores y dentro sus respectivos directorios para almacenar y tener ordenada la información.



# Azure Data Factory

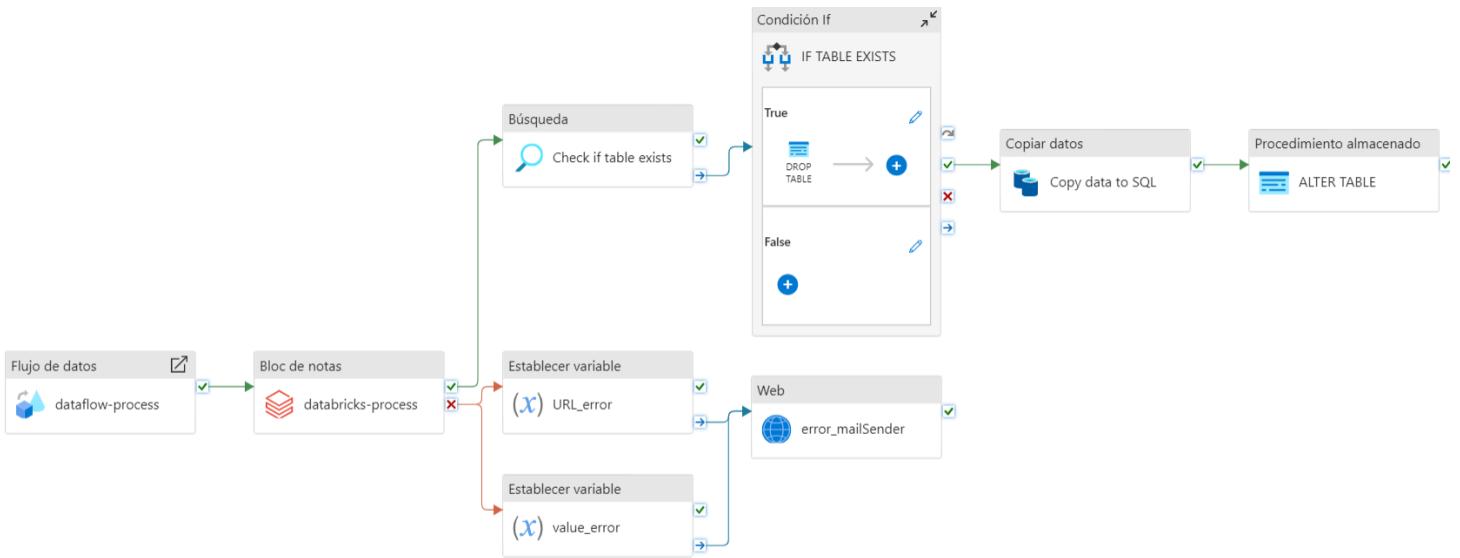
## ¿Qué es Data Factory?

Azure Data Factory permite conectar los diferentes servicios de Azure entre sí para realizar la extracción, transformación y carga (ETL).

En este caso, Data Factory actúa como la raíz de infraestructura. Se compone de una única pipeline parametrizada que realiza todo el proceso. También integra una gestión de errores y alertas dedicadas.

## Esquema de Data Factory





## Estructura del Data Factory

Consiste en un `forEach` que recibe por parámetros los archivos a procesar que pueden ser 1, 2, 3 o N. Después cada elemento del for (archivo.csv), va avanzando por cada actividad interna hasta completarse.

Se ha elegido usar parámetros en lugar de variables para, que en caso de tener 30 archivos, puedes elegir procesar uno o varios de ellos. Esto también simplifica la estructura del Data Factory y otorga flexibilidad.

A continuación se explicarán las actividades internas por separado.

## Parámetros

Los parámetros en Azure funcionan por niveles o capas. Puedes definir parámetros a distintos niveles:

- A nivel de Pipeline
- A nivel de DataFlow
- A nivel de DataSet
- etc..

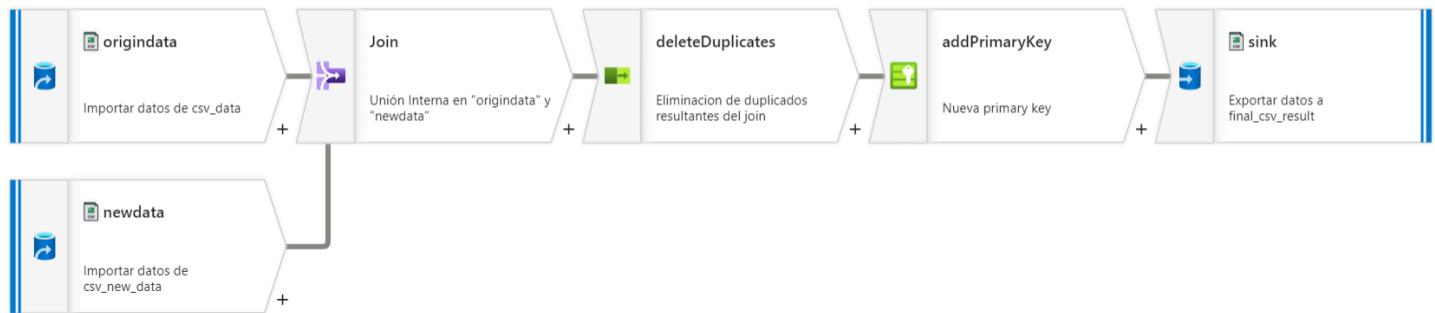
Dentro del proyecto, los parámetros a nivel de Pipeline se propagan a las capas inferiores haciendo que por ejemplo, el DataFlow reciba el elemento del for y procese un archivo diferente en cada iteración o que el archivo resultante se guarde con el nombre de la empresa en lugar de 'part-001-fragmentid-14445'.

En este caso, existe un único parámetro con los nombres del archivo.csv que se propaga al resto de actividades y los emails a los que se van a enviar los detalles en caso de que se produzca un error.

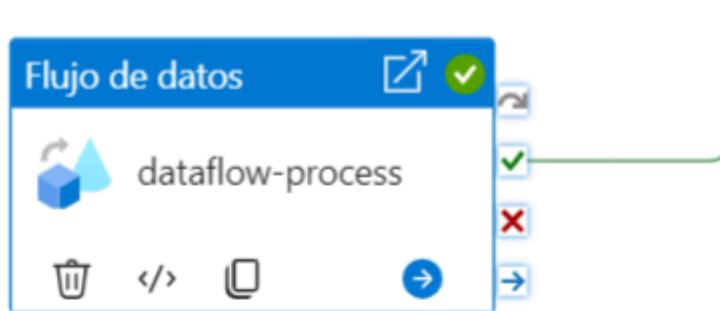
<input type="button"/> Nuevo	<input type="button"/> Eliminar	
Nombre	Tipo	Valor predeterminado
<input type="text"/> file_name_origin	<input type="button"/> Array	<input type="text"/> ["Hiberus.csv","Viewnext.csv"]
<input type="text"/> email	<input type="button"/> String	<input type="text"/> dmitry.e@vgmail..com

## Azure DataFlow

Es la primera actividad que va a ejecutar el ForEach. Su función es procesar los diferentes archivos.csv y unirlos en un único resultado. Además, se agrega una columna adicional que se usará como Primary Key en cada tabla de la BDD.



## Configuración de parámetros del DataFlow



Acción Parámetros Propiedades de usuario

dataflow\_processss ▼ Abierto + Nuevo

∨ origindata parámetros ⓘ

Nombre	Valor
file_name	@item()

∨ newdata parámetros ⓘ

Nombre	Valor
file_name	@concat('tema',item())

\* ⓘ AutoResolveIntegrationRuntime ▼

Conjuntos de datos 5

- CSV 3
  - csv\_data
  - csv\_new\_data
  - final\_csv\_result
- SQL 2
  - ds\_csv\_to\_sql
  - ds\_sql

**El parámetro que envía la información al Dataset origindata/csv\_data** carga el archivo.csv del DataFlow que es el elemento en curso del forEach como: "Hiberus.csv".

**El parámetro que envía la información al dataset de newdata**, concatena "tema" al nombre del fichero para identificar correctamente el archivo adicional. Esta aproximación también permite evitar crear más parámetros a nivel de pipeline con nombres csv explícitos.

El resultado se carga en los datasets y se unen mediante las actividades internas del dataflow (join, union etc...) en un único archivo que se guarda en un contenedor del Data Lake.

## Databricks

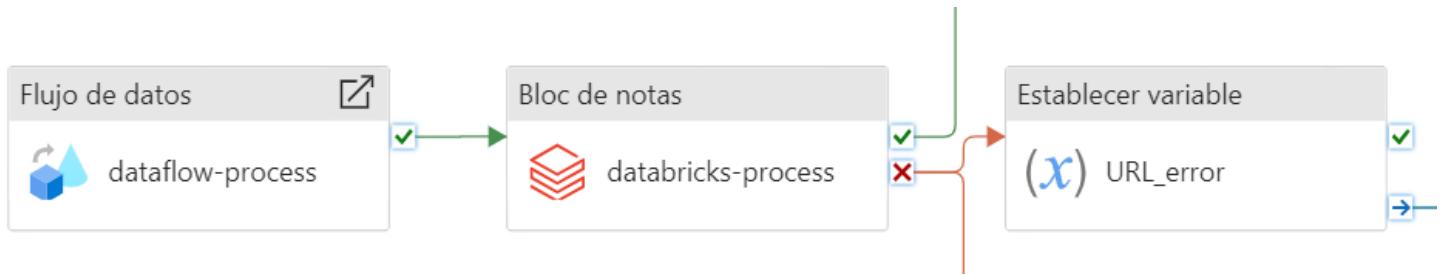
### ¿Qué es Databricks?

Es una plataforma analítica basada en Apache Spark que ofrece una solución ETL completa y permite tratar los datos en forma de batch (por lotes), streaming o via machine learning.

En este caso, se ha utilizado un notebook de Databricks para realizar el **análisis de sentimientos** de cada tweet a través de un modelo LLM.

### Estructura

Se utilizó un Token SAS para conectarse al servicio de Azure Databricks. Después, dentro de la pipeline Databricks procesa los archivos resultantes del Data Flow.



El nombre de cada archivo se obtiene de forma dinámica, mediante parámetros que se indican tanto en Azure como en el Notebook.

Después se realiza un Script en Python para:

- Realizar un análisis de los tweets con un modelo LLM de HuggingFace.
- Seleccionar, tratar y organizar la información.
- Guardar la información en un directorio temporal para renombrarla y moverla al contenedor de Azure correspondiente.

# Script

8

```
> esquema = tipos.StructType([ ...  
])  
  
#Indicamos el archivo a procesar  
file_location = f'abfss://rawdata@tweetsfiles.dfs.core.windows.net/processed_dataflow_csv/{filename}'  
raw_tweets = spark.read.schema(esquema).format("csv").option("delimiter", "*").option("header", "true").load(file_location).distinct()  
  
# Wrapper del LLM  
sentiment_classifier = pipeline(  
    model="lxyuan/distilbert-base-multilingual-cased-sentiments-student",  
    top_k=1  
)  
  
# Definimos la UDF para obtener el sentimiento => return esperado: 'positivo, 0.4254297'  
def apply_sentiment(text):  
    texto = str(text)  
    sentiment = sentiment_classifier(texto)  
    score_sentiment = sentiment[0][0]['score']  
    label_sentiment = sentiment[0][0]['label']  
  
    if label_sentiment == 'positive' and score_sentiment <= 0.61:  
        label_sentiment = 'neutral'  
    return f'{label_sentiment}, {score_sentiment}'  
  
sentiment_udf = udf(apply_sentiment)  
  
# Definimos la UDF para sacar modificar los valores nulos  
def apply_clear_null(text):  
    if text is None:  
        text = "No especificado"  
    return text  
clear_null_udf = udf(apply_clear_null)  
  
tweets = (raw_tweets  
    .select("primaryID", "user_id", "username", "tweet_id", "created_at", "retweet_count", "likes", "view_count", "hashtags", "text", "location",  
    "longitud")  
    # Sacamos la URL y la eliminamos del texto principal. Ademas, normalizamos el texto eliminando las tildes  
    .withColumn("URL", fn.regexp_extract("text", r'(http)\S+', 0))  
    .withColumn("text", fn.regexp_replace(fn.translate("text", "áéíóú", "aeiou"), r'(http)\S+', ""))  
    # Creamos la columna de sentimientos y de ahí, sacamos 2 columnas [score y el valor]  
    .withColumn("Sentimientos", sentiment_udf('text'))  
    .withColumn("Score", fn.substring(fn.split("Sentimientos", r',\s')[1], 0, 4))  
    .withColumn("Sentimientos", fn.split("Sentimientos", r',\s')[0])  
    # Normalizamos los valores nulos  
    .withColumn("location", clear_null_udf("location"))  
    .withColumn("hashtags", clear_null_udf("hashtags"))  
)
```

## Guardar los datos en el DataLake

10

```
#ruta = f'abfss://outputdata@tweetsfiles.dfs.core.windows.net/{routename}'  
ruta = f'abfss://outputdata@tweetsfiles.dfs.core.windows.net/lib/{filename}'  
ruta_temporal = f'abfss://outputdata@tweetsfiles.dfs.core.windows.net/tmp'  
tweets.repartition(1).write.format("CSV").option("header", "true").option("delimiter", "*").mode("overwrite").save(ruta_temporal)  
  
contenedor = dbutils.fs.ls(ruta_temporal)  
for fichero in contenedor:  
    if '.csv' in fichero.name:  
        file = fichero.name  
  
dbutils.fs.cp(f'{ruta_temporal}/{file}', ruta)  
dbutils.fs.rm(ruta_temporal, True)
```

## Creación de la BDD de forma automática

### Esquema general del proceso



Azure Data Factory permite copiar datos con la actividad de `COPY DATA`. En este caso, recuperaremos el archivo.csv resultante del Databricks de nuestro contenedor de Data Lake para copiar y guardar los datos en una BDD SQL.

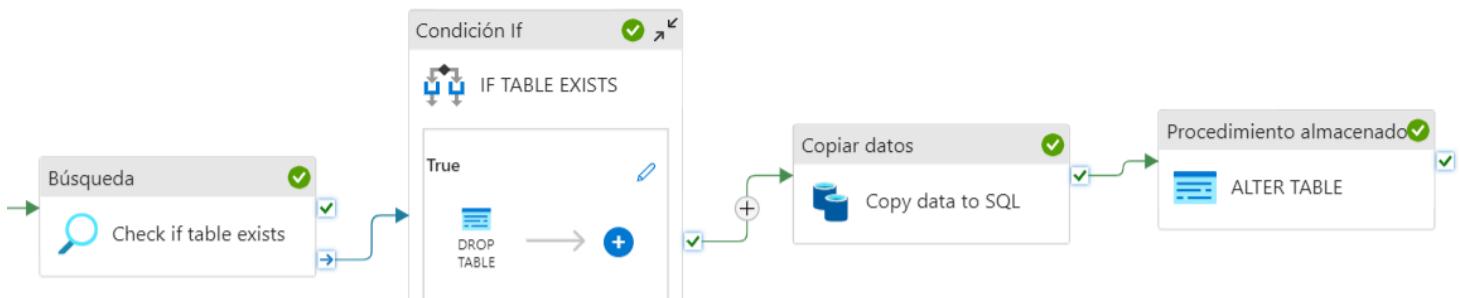
Asimismo, esta actividad es capaz de generar una, o múltiples tablas de forma automática con una estructura personalizada. Es decir, mediante `COPY DATA` puedes automatizar todo el proceso de creación del esquema e inserción de datos en la BDD.

Para ello, necesitas configurar el flujo de Data Factory, en mi caso he incluido:

- Una actividad de `COPY DATA` para que haga la tabla con sus tipos de datos.
- Una actividad de `PROCEDIMIENTO` para borrar los datos en caso de existir.

- Una actividad de PROCEDIMIENTO para modificar la tabla y definir una PRIMARY KEY .
- Parametrizar las actividades para que por cada elemento del for se cree una nueva tabla con el nombre correspondiente.

## Esquema del DataFlow



Podemos ligar la actividad de BÚSQUEDA con la BDD para realizar una consulta automática. Si esta consulta nos devuelve un resultado, significa que existen datos en la tabla y se procederá a ejecutar el PROCEDIMIENTO DE DROP TABLE .

Después enlazamos el flujo con la actividad de COPY DATA para generar la tabla con el nombre de la empresa en curso y sus respectivos datos. Por último, desencadenamos los PROCEDIMIENTOS creados en la BDD para personalizar la tabla resultante.

General    Origen    **Receptor**    Asignación    Configuración    Propiedades de usuario

---

Conjunto de datos de receptor \*  Abierto  [Más información](#)

**Propiedades del conjunto de datos**

Nombre	Valor
table_name	@split(item(), ':')[0]

Comportamiento de escritura  Insertar  Actualizar/insertar (upsert)  Procedimiento almacenado

Bloqueo de tabla de inserción masiva  Yes  No

Opción de tabla  Ninguno  Crear tabla automáticamente

## Resultado y código de los Procedimientos

twitterSQL

Tables

- > dbo.Hiberus
- > dbo.Minsait
- > dbo.Nttdata
- > dbo.Viewnext
- > Dropped Ledger Tables
- > Viasarc

New Column ▲ Move Up ▼ Move Down

ve	Name	Type	Primary Key	Allow Nulls	Default Value	Remove
=	primaryID	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="Delete"/>
=	user_id	bigint	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="button" value="Delete"/>
=	username	nvarchar(MAX)	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="button" value="Delete"/>
=	tweet_id	bigint	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="button" value="Delete"/>

El parámetro que usan los procedimientos de la base de datos es **el elemento** del forEach en curso.

```
CREATE PROCEDURE [dbo].[ModificarTabla]
    @NombreTabla NVARCHAR(100)
AS
BEGIN
    EXEC('ALTER TABLE ' + @NombreTabla + ' ALTER COLUMN primaryID INT NOT NULL;');
    EXEC('ALTER TABLE ' + @NombreTabla + ' ADD PRIMARY KEY (primaryID);');
    EXEC('DELETE FROM ' + @NombreTabla + ' WHERE user_id IS NULL;');
END;
GO
```

```
▶ Run □ Cancel ⚙ Disconnect ⓘ Change | Database: twitterSQL
1 CREATE PROCEDURE [dbo].[BorrarTabla]
2 @NombreTabla NVARCHAR(100)
3 AS
4 IF OBJECT_ID(@NombreTabla, 'U') IS NOT NULL
5 BEGIN
6     EXEC ('DROP TABLE ' + @NombreTabla);
7 END;
8 GO
```

El 'u' en la función OBJECT\_ID es un parámetro que especifica el tipo de objeto que se está buscando en la base de datos.

- 'U': Se refiere a una tabla de usuario.
- 'V': Se refiere a una vista.
- 'P': Se refiere a un procedimiento almacenado.
- 'FN': Se refiere a una función definida por el usuario (escalar).
- 'IF': Se refiere a una función definida por el usuario (tabla en línea).

**Si se encuentra una tabla con ese nombre, la función devuelve su identificador único de objeto (object\_id); de lo contrario, devuelve NULL .**

## Gestión de errores, alertas y métricas

Azure tiene múltiples formas para gestionar los logs, métricas y errores. En esencia, es posible configurar una Alerta para cada servicio o grupo de recursos en particular.

Para este proyecto **se ha configurado Data Factory** para volcar todas las métricas y logs en **Azure Log Analytics** y después se han creado alertas personalizadas mediante el lenguaje kusto (KQL).

## Alerta de presupuesto

Podemos definir alertas para que una vez superado un cierto umbral se nos avise via mail o SMS.

1. Creamos un presupuesto en el apartado de Cost Management / presupuesto.
2. Configuramos la alerta en la misma interfaz basándonos en el presupuesto especificado.

✓ Crear un presupuesto    ✓ **Establecer alertas**

Configure condiciones de alerta y envíe notificaciones por correo electrónico en función de sus gastos.

### \* Condiciones de alerta

Tipo	Porcentaje del presupuesto	Importe	
Real	50	45	
Real	20	18	
Seleccionar tipo	Escriba el porcentaje.	-	

### \* Destinatarios de la alerta (correo electrónico)

Destinatarios de la alerta (correo electrónico)

example@email.com

Se recomienda agregar `azure-noreply@microsoft.com` a la lista de permitidos de correo electrónico para asegurarse de que los correos de alerta no vayan a la carpeta de correo no deseado.

### Preferencia de idioma

Seleccione el idioma en el que quiera recibir el correo electrónico de alerta para todos los destinatarios indicados anteriormente. El valor predeterminado es el idioma asociado a su inscripción.

Idiomas \*    Predeterminado

# El presupuesto Quiebra ha superado el umbral y debe revisarlo

El importe total evaluado para su presupuesto de Azure (Quiebra) es ahora \$108.48. Este importe supera el umbral de \$45.00 que especificó. Debe revisar este presupuesto en Azure Portal y analizar el costo y el uso.

[Revisar el presupuesto >](#)

---

Nombre del presupuesto: Quiebra

Fecha de inicio del presupuesto: miércoles, 1 de mayo de 2024

Tipo de presupuesto: Cost

Umbral del presupuesto: \$90.00

Valor evaluado: \$108.48

Umbral de notificación para la alerta: \$45.00

Hora de generación de la alerta: miércoles, 15 de mayo de 2024 12:23:00 UTC

---

## Azure Log Analytics

Este servicio nos permite unificar todas las métricas/logs de nuestros servicios y se usa para editar y ejecutar consultas de registro. Además, lo interesante de este sistema centralizado es que puedes hacer “notebooks” aka Libros de cualquier consulta y se pueden guardar para su posterior reutilización.

De forma que, una vez hecha una consulta no hay necesidad de volverla a escribir. Esto permite crear filtros de logs, alertas, métricas totalmente a tu gusto y mantener todas las consultas documentadas en un solo lugar.

## Alertas personalizadas con KQL

1. Creamos el servicio de Log Analytics.

2. Activamos el volcado de datos en el servicio de Data Factory y seleccionamos los datos que queramos obtener.

Inicio > **factoria-twitter**

## factoria-twitter | Configuración de diagnóstico

Factoría de datos (V2)

Buscar Actualizar Comentarios

Información general Registro de actividad Control de acceso (IAM) Etiquetas Diagnosticar y solucionar problemas

Configuración Redes Identidades administradas Propiedades Bloqueos

Introducción Inicio rápido

Supervisión Alertas Métricas

Configuración de diagnóstico

Registros

La configuración de diagnóstico se usa para configurar la exportación de métricas a destinos independientes. [Más información sobre la configuración de diagnóstico](#)

Configuración de diagnóstico

Nombre	Cuenta de almacén
No se ha definido ninguna configuración de diagnóstico.	

+ Agregar configuración de diagnóstico

Haga clic en "Aregar configuración de diagnóstico" anteriormente para ver las configuraciones existentes.

- Pipeline activity runs log
- Pipeline runs log
- Trigger runs log
- Sandbox Pipeline runs log
- Sandbox Activity runs log
- SSIS package event messages
- SSIS package executable statistics
- SSIS package event message context
- SSIS package execution component phases
- SSIS package execution data statistics
- SSIS integration runtime logs
- Airflow task execution logs
- Airflow worker logs
- Airflow dag processing logs
- Airflow scheduler logs
- Airflow web logs
- AllMetrics

**Log Analytics** generará diferentes tablas con la información seleccionada con las cuales podemos interactuar tal como si de SQL se tratase:

Registros

Configuración

- Tablas
- Agentes
- Uso y costos estimados
- Exportación de datos
- Aislamiento de red
- Cuentas de almacenamiento vinculadas
- Propiedades
- Bloqueos

Clásico

- Administración de agentes antiguos
- Conector de registro de

Se muestran 20 resultados.

Nombre de la tabla ↑↓	Tipo ↑↓	Plan ↑↓
ADFActivityRun	Tabla de Azure	Analytics
ADFPipelineRun	Tabla de Azure	Analytics
ADFSandboxActivityRun	Tabla de Azure	Analytics
ADFSandboxPipelineRun	Tabla de Azure	Analytics
ADFSISIntegrationRuntimeLogs	Tabla de Azure	Analytics
ADFSISPackageEventMessageContext	Tabla de Azure	Analytics
ADFSISPackageEventMessages	Tabla de Azure	Analytics
ADFSISPackageExecutableStatistics	Tabla de Azure	Analytics
ADFSISPackageExecutionComponentPhase	Tabla de Azure	Analytics
ADFSISPackageExecutionDataStatistics	Tabla de Azure	Analytics
ADFTtriggerRun	Tabla de Azure	Analytics
Alert	Tabla de Azure	Analytics

### 3. Realizamos consultas personalizadas en KQL para integrarlas posteriormente con las Alertas

**TweetLogs | Libros | statusFilter**

Área de trabajo de Log Analytics

Buscar

Libros Edición finalizada

System Center Resumen del área de trabajo (en desuso)

Service Map (en desuso) Máquinas virtuales (en desuso)

Configuraciones de ámbito (en desuso)

Supervisión Conclusiones Alertas

2 Editando el elemento consulta: consulta - 0

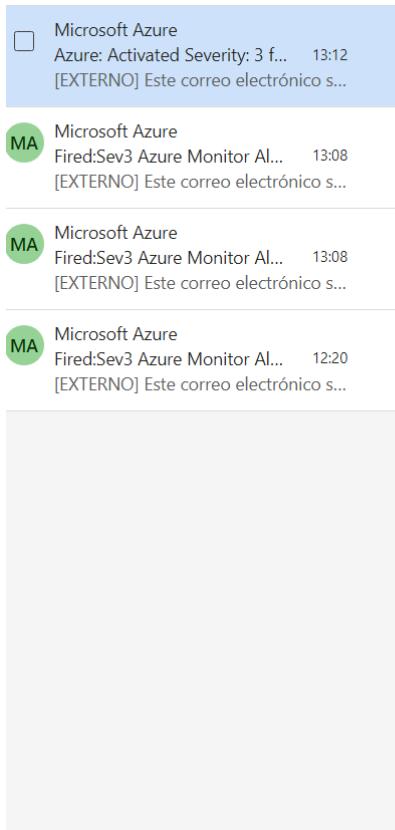
Ejecutar consulta Muestras Consulta ⓘ (cambiar) Intervalo de tiempo Últimas 30 dí... Visualización ⓘ Tar...

Consulta de Área de trabajo de Log Analytics Registros

```
ADFActivityRun
| project TimeGenerated, ActivityName, Status, ErrorCode, ErrorMessage, Output
| where Status !in("Succeeded", "Queued", "InProgress")
```

TimeGenerated	↑↓ ActivityName	↑↓ Status	↑↓ ErrorCode	↑↓ ErrorMessage	↑↓
16/5/2024, 9:49:02,121	databricks-process	Failed	3204	Databricks execution failed with error state: Terminated, e...	
16/5/2024, 9:50:20,119	Procesamiento de datos	Cancelled			
16/5/2024, 9:50:20,392	databricks-process	Cancelled			
16/5/2024, 9:58:17,063	databricks-process	Failed	3204	Databricks execution failed with error state: Terminated, e...	
16/5/2024, 10:00:40,125	databricks-process	Failed	3204	Databricks execution failed with error state: Terminated, e...	
16/5/2024, 10:01:39,417	Procesamiento de datos	Cancelled			
16/5/2024, 13:02:18,041	Procesamiento de datos	Cancelled			
16/5/2024, 13:02:18,298	databricks-process	Cancelled			
15/5/2024, 17:53:52,245	databricks-process	TimedOut	ActionTimedOut	Activity timed out	

4. Finalmente, configuraremos la alerta para que nos envíe un mail en caso de cumplir las condiciones establecidas



Your Azure Monitor alert was triggered

Azure monitor alert rule failedAlertSender was triggered for factoria-twitter at May 16, 2024 11:11 UTC.

Rule ID	/subscriptions/[REDACTED]/resourceGroups/Tweets/providers/Microsoft.Insights/metricAlerts/failedAlertSender <a href="#">View Rule &gt;</a>
Resource ID	/subscriptions/[REDACTED]/resourceGroups/Tweets/providers/Microsoft.DataFactory/factories/factoria-twitter <a href="#">View Resource &gt;</a>

**Alert Activated Because:**

Metric name	PipelineCancelledRuns
Metric namespace	factories/factoria-twitter
Dimensions	ResourceId = /SUBSCRIPTIONS/[REDACTED]/RESOURCEGROUPS/TWEETS/PROVIDERS/MICROSOFT.DATAFACTORY/FACTORIES/FACTORIA-TWITTER
Time Aggregation	Total

## Logic Apps orientadas a notificaciones

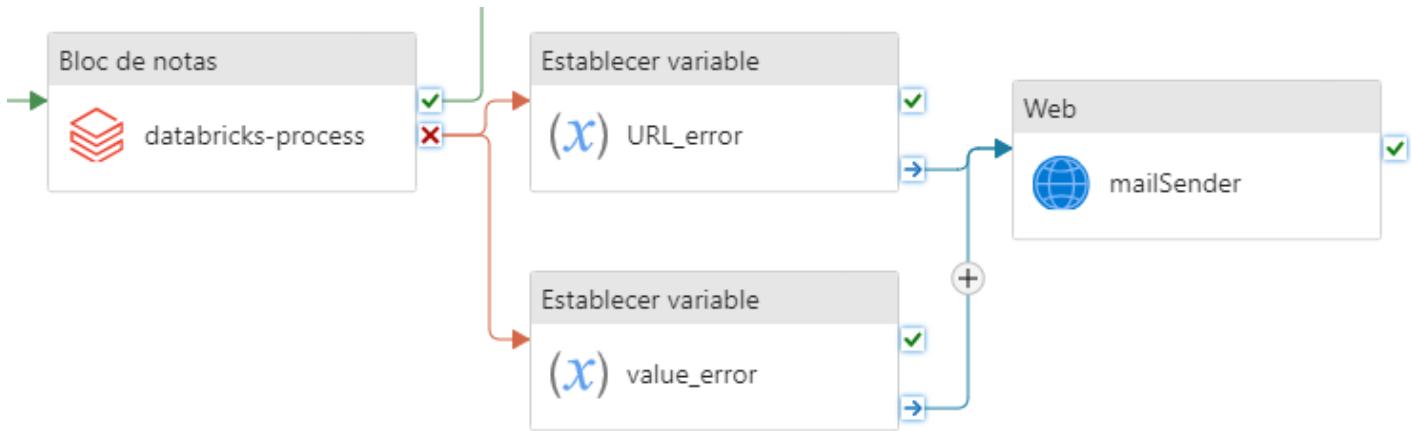
Es posible configurar y crear pequeñas aplicaciones que notifiquen al destinatario y personalizar esta información para tus necesidades y caso particular.

Por ejemplo, si se tiene constancia de que un servicio es propenso al fallo, podríamos configurar una alerta al correo electrónico para que nos muestre que ha sucedido.

## Alerta de error personalizada con Logic APP

A continuación, se muestra un ejemplo para el servicio de Databricks en un DataFactory.

1. Configuramos el pipeline para que, en caso de error se pueda aislar, parsear y transmitir por HTTP a la logic app.



2. La logic app recibe el JSON de la actividad y procede a realizar la acción pre-configurada. El esquema JSON es totalmente personalizable



3. El destinatario recibe el email donde se especifica:

- El nombre de la pipeline, factoría y archivo que dio el error.
- El error en “bruto” de Databricks.
- La URL del debugger de Databricks.



outlook\_7EB8FA65FB6CCB4E@outlook.com en nombre de outlook\_7EB8FA65FB6CCB4E@outlook.com<dimacontacto94@gmail.com>  
Para: DMITRY POLDENEV KUZNETSOV

···  
· · · · ·  
· Mar 14/05/2024 14:

**[EXTERNO]** Este correo electrónico se envió desde fuera de la organización. No haga clic en enlaces ni abra archivos adjuntos a menos que reconozca al remitente y sepá que el contenido es seguro

Ha ocurrido un error en main\_process. El archivo viewnext.csv a fallado en procesarse demasiadas veces.

**Más información:** Databricks execution failed with error state: Terminated, error message: Workload failed, see run output for details. For more details please check the run page [here](https://adb-852903750915076.16.azuredatabricks.net/?o=852903750915076#job/350034229515367/run/1007835841641668).

**Fallo:** PythonException: An exception was thrown from the Python worker. Please see the stack trace below. 'NameError: name 'texto' is not defined', from root.ipynotebook16270command-2542129566031020-3615426011, line 37. Full traceback below: Traceback (most recent call last): File root.ipynotebook16270command-2542129566031020-3615426011, line 37, in apply\_sentiment NameError: name 'texto' is not defined

**La URL de Databricks requiere autenticación, pero te permite acceder a un análisis del problema muy completo que incluye diferentes métricas:**

Actualizar

Spark

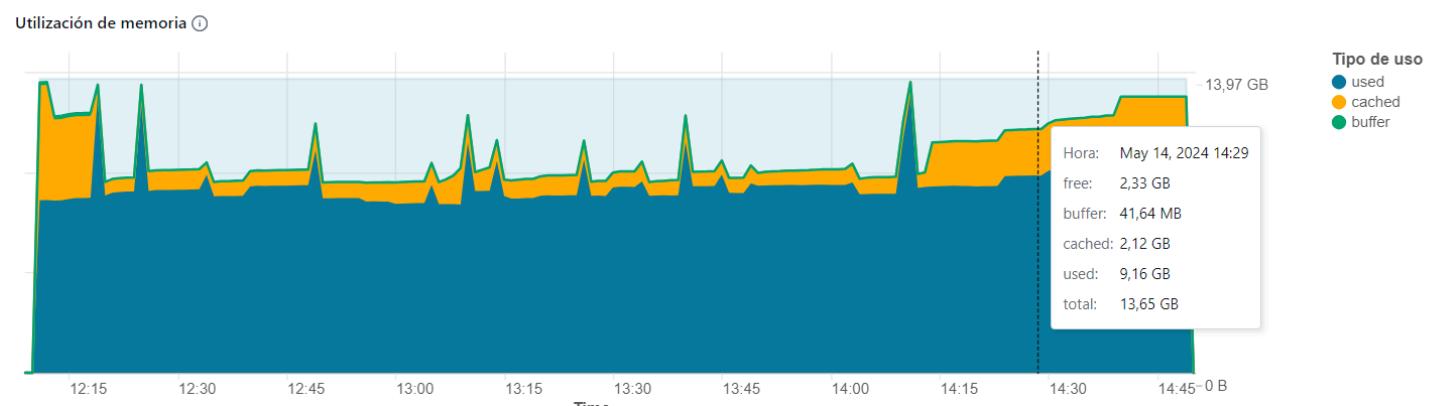
areas activas ⓘ

Hardware

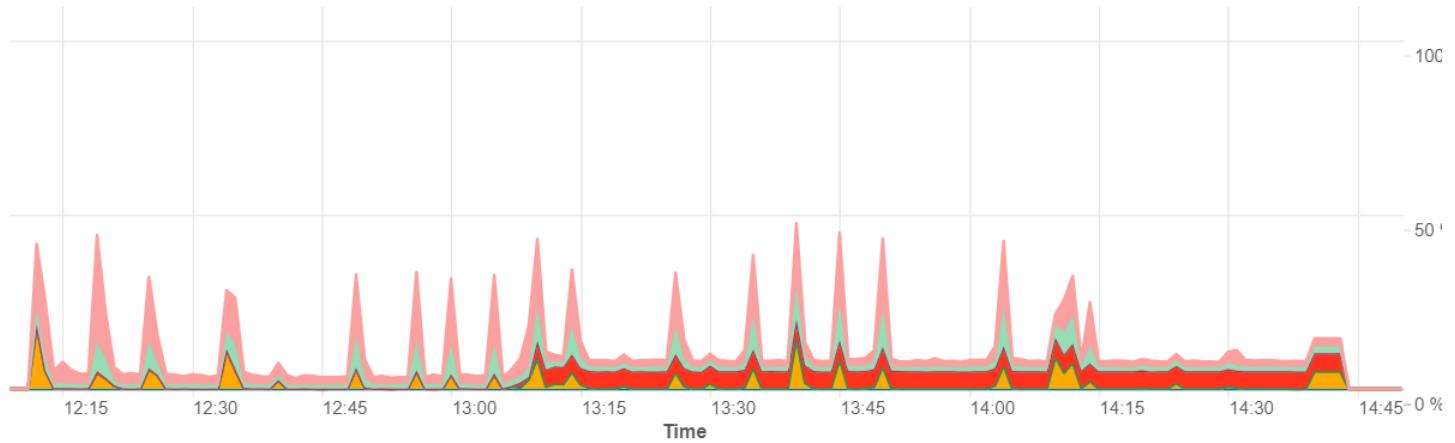
✓ Spark

GPU

## Métricas del hardware

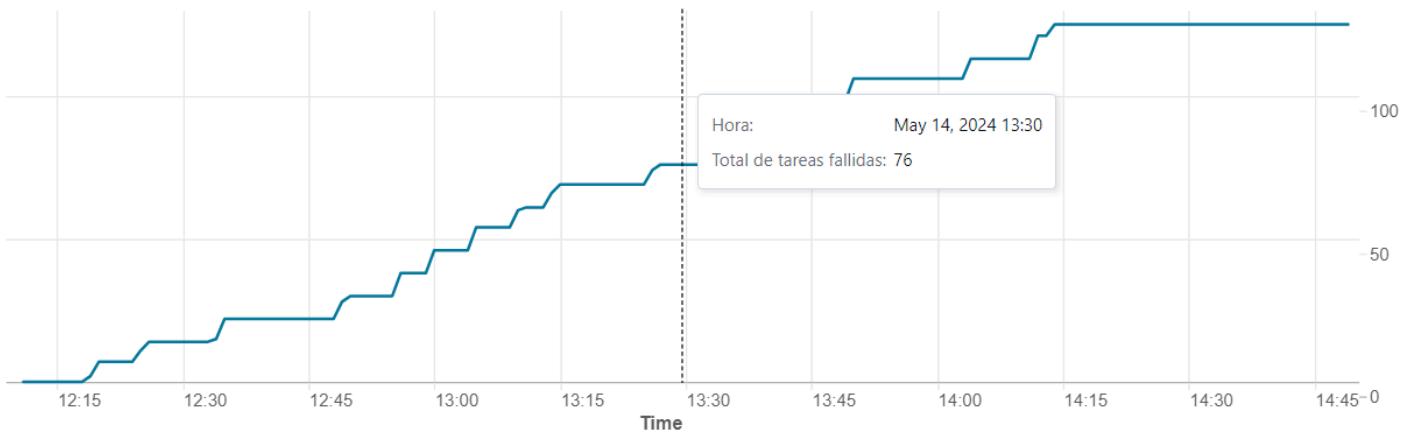


### Utilización de la CPU ⓘ

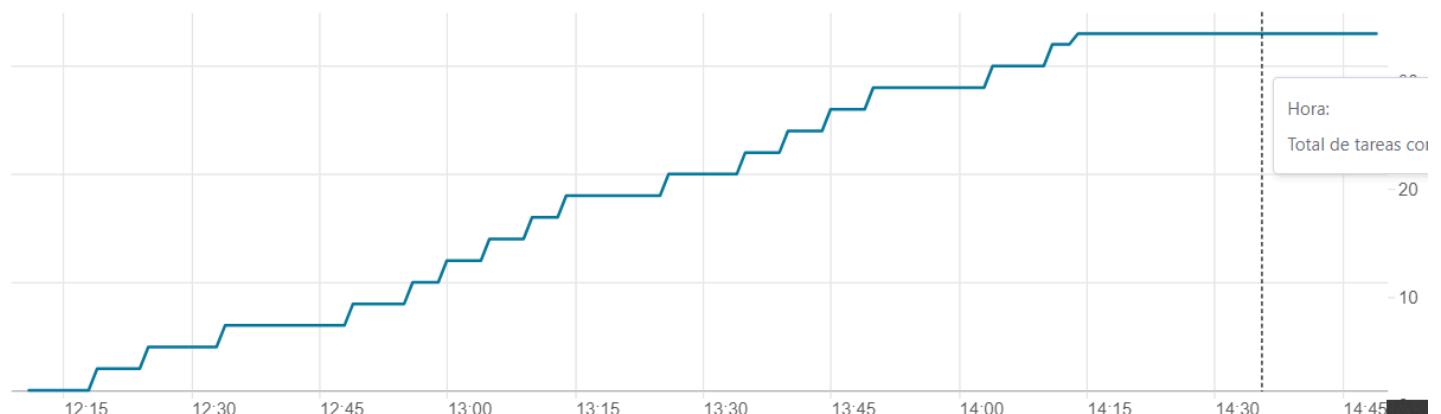


## Métricas de Spark

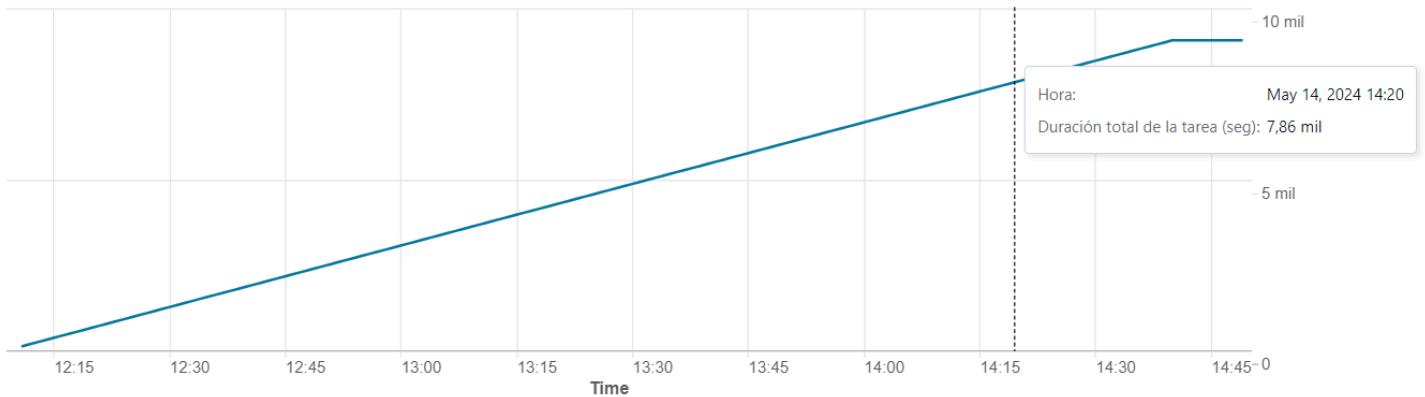
### Total de tareas fallidas ⓘ



### Total de tareas completadas ⓘ



Duración total de la tarea (segundos) ⓘ



# Visualización - Grafana

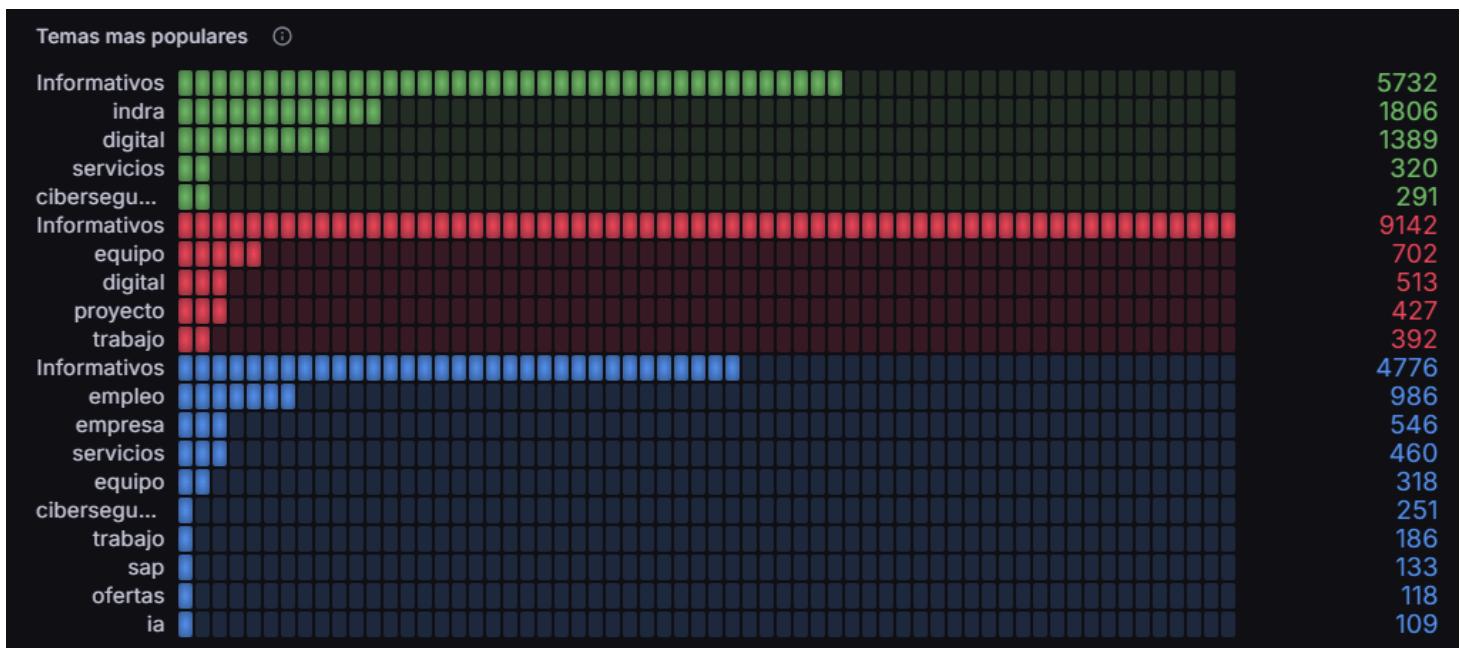
En la carpeta `PDF` es posible ver todos los gráficos.

## ¿Qué es Grafana?

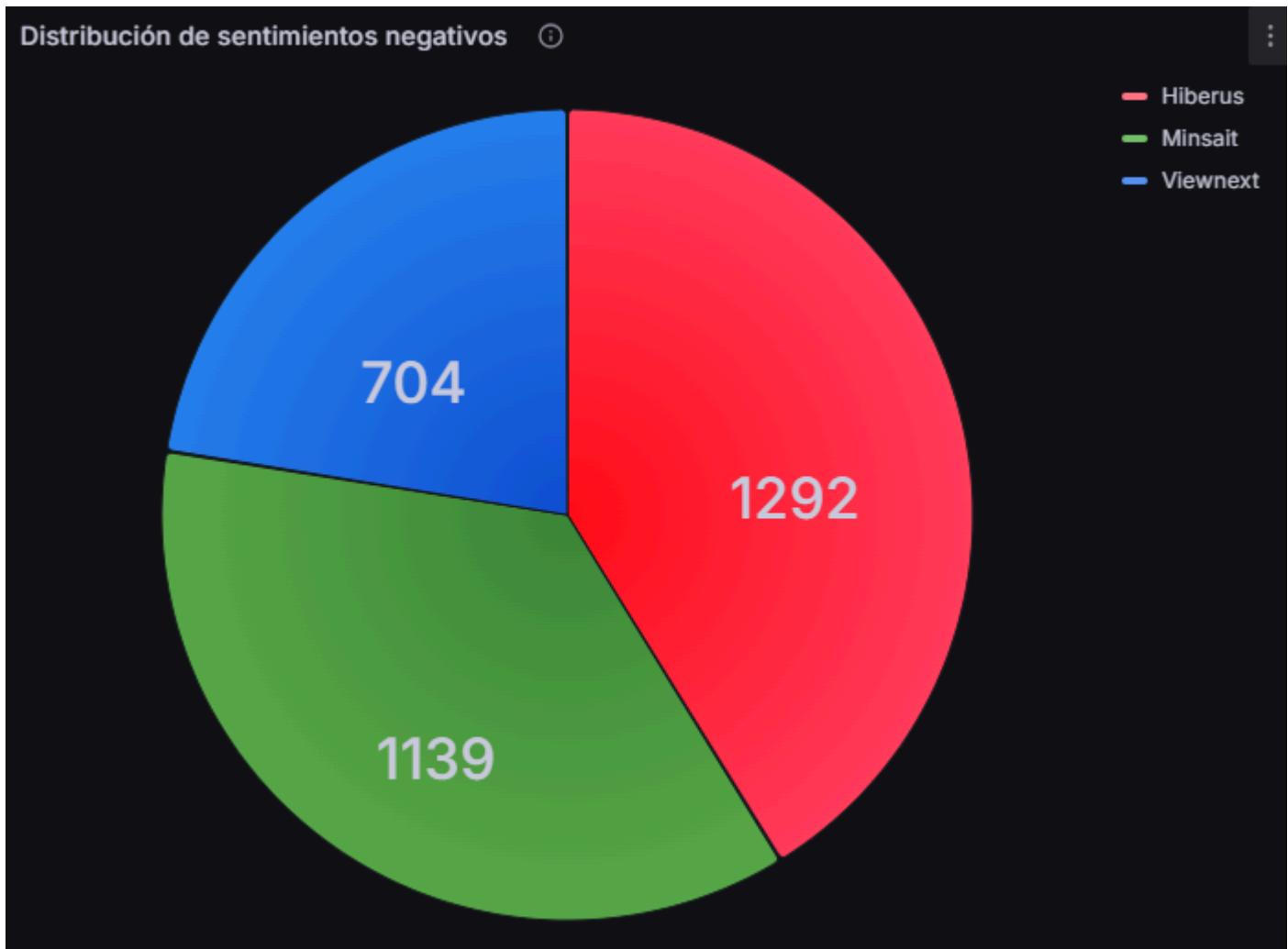
Grafana es una plataforma interactiva y dinámica de código abierto. Permite almacenar, visualizar, analizar y comprender métricas de rendimiento/datos de una forma clara y sencilla.

## Visualizaciones totales y temas más populares

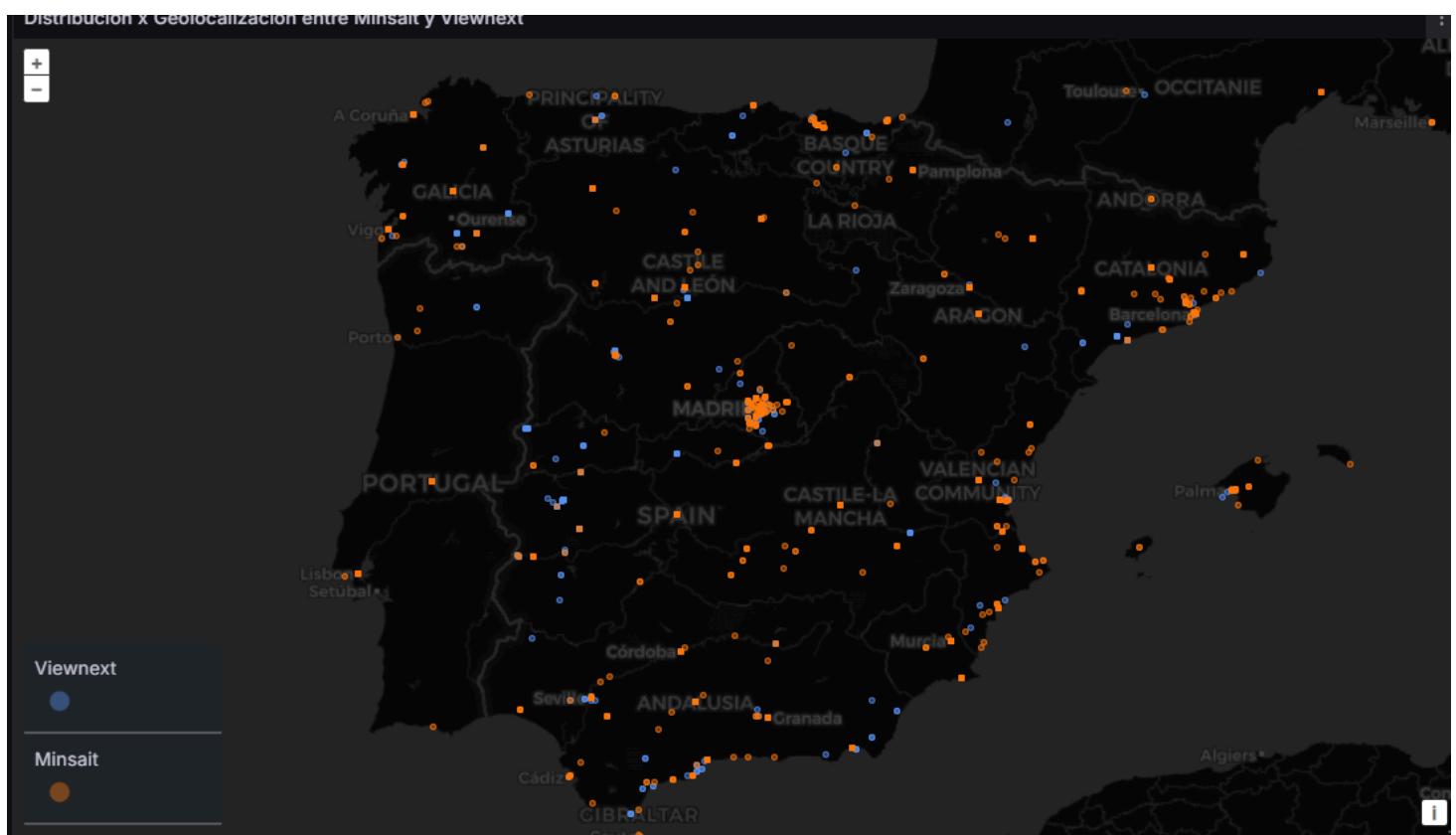




## Distribución de sentimientos negativos

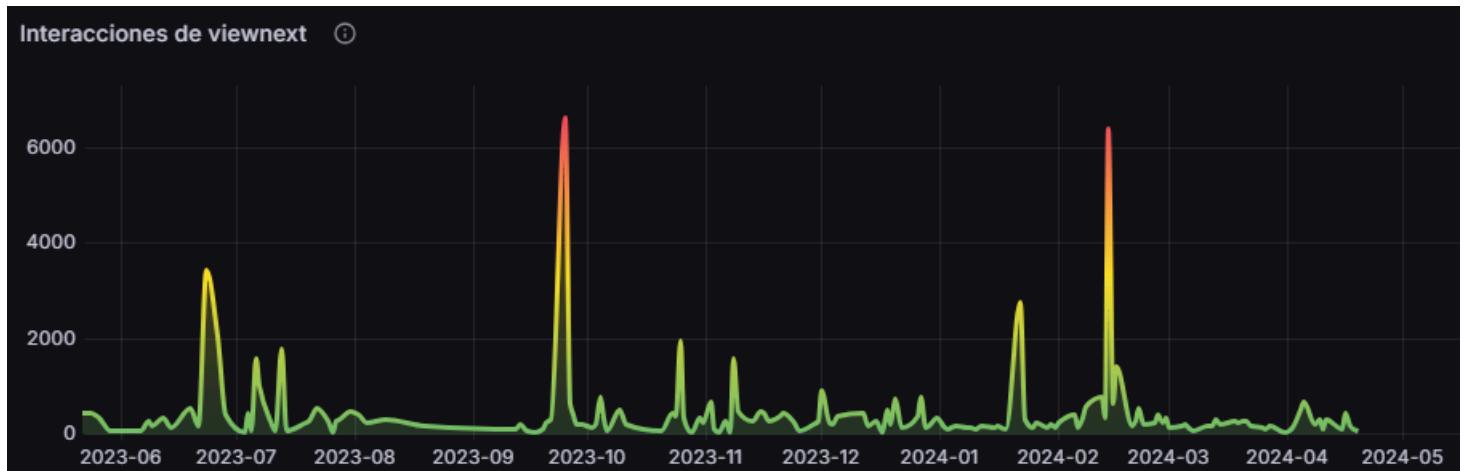


## Distribución por geolocalización entre Minsait y Viewnext

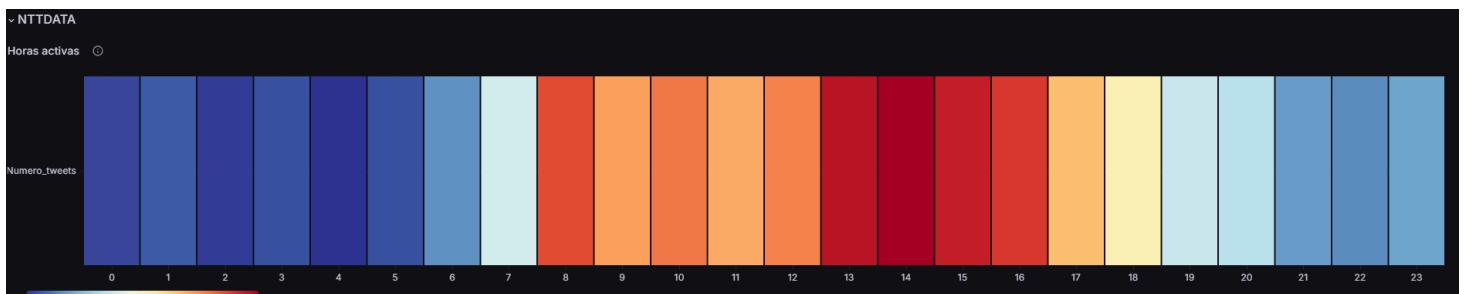


El GEOMAP indica desde dónde se han enviado los tweets en España.

## Interacciones totales (likes, retweets y visualizaciones) de Viewnext en el último año

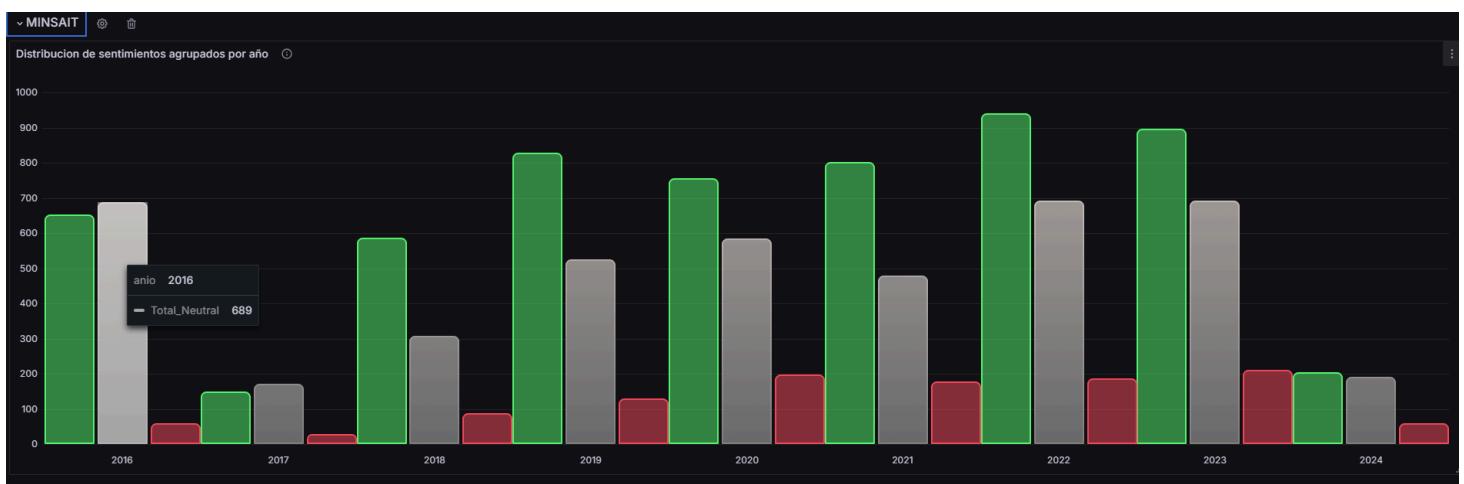


## Horas de actividad donde más aparece NTTDATA



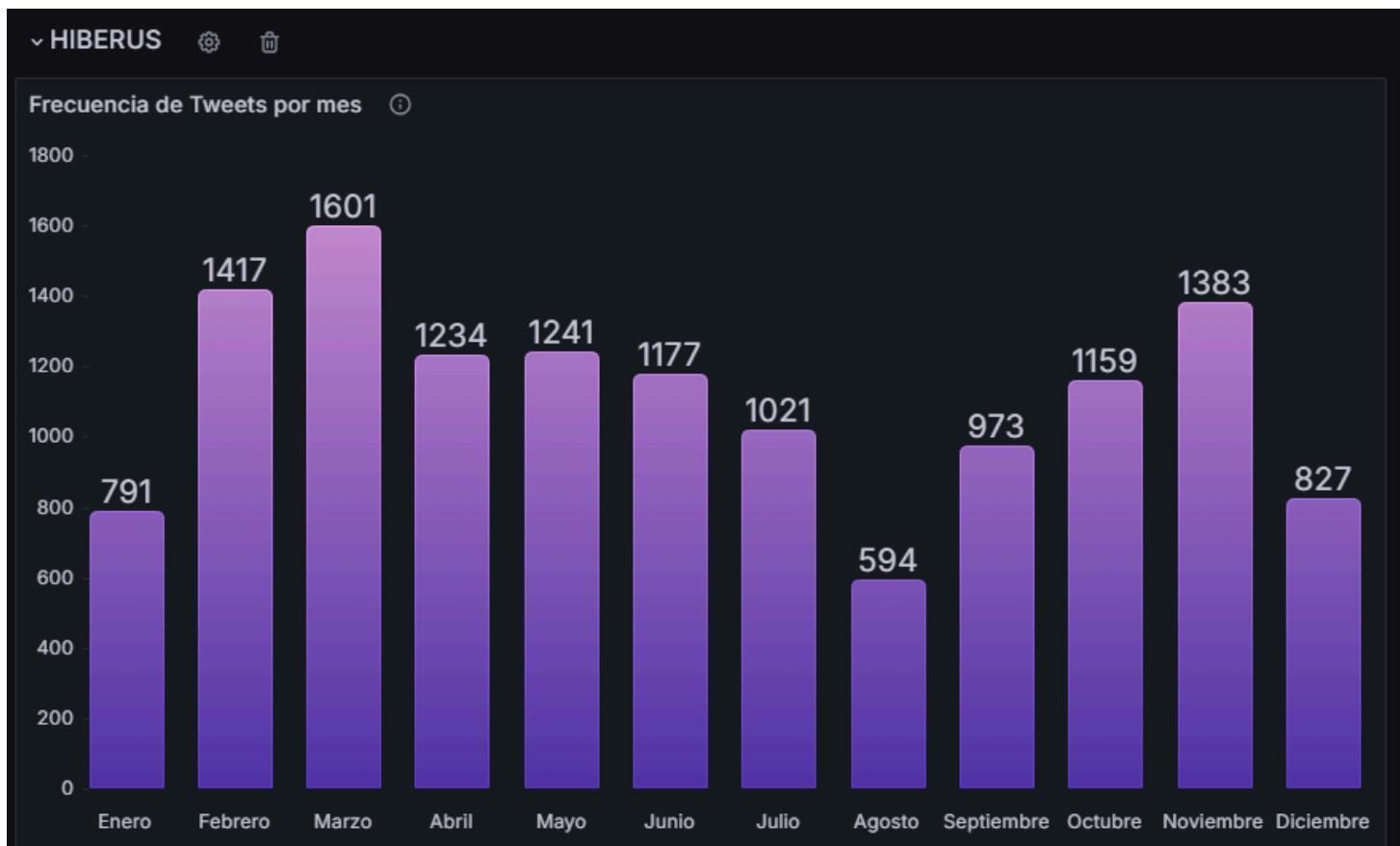
En este mapa de calor, entre las 13:00 y las 15:00 horas sería el rango donde han twiteado sobre la empresa.

## Visualización de sentimientos de Minsait por año



- Color verde: Tweets con tendencia positiva.
- Color gris: Tweets neutrales (informativos).
- Color rojo: Tweets con tono negativo.

## Frecuencia de Tweets por mes de Hiberus



## Actividad de TOP 5 usuarios de Hiberus

