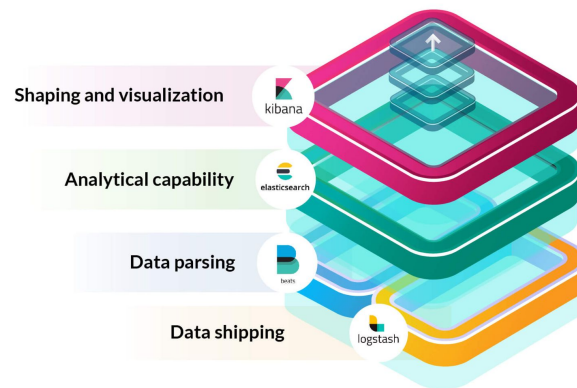


# Proyecto final - Grupo 3

## ELASTIC STACK para monitorizar sedes

### Grupo 3:

Garai Martínez de Santos  
Pablo Ruiz de Azúa  
Markel Aguirre



# Caso de uso

- Simulación de datos de sensores de una red de plantas industriales (Imaginamos que cada sede de Tecnalia tiene una planta industrial).
  - Script de python generando datos.
    - Humedad
    - Temperatura
    - Luminosidad
  - Incluir localización de las sedes
  - Enviar datos a **Elasticsearch** (con librería de python)
  - Dockerizar simulador de datos, de forma que se levante un docker y cada X tiempo genere datos.
- Captura de datos meteorológicos en las diferentes localizaciones de las plantas industriales.
  - **Logstash** con el plugin `http_poller` tirando de una API pública de meteorología.
- Visualización de la información en un **dashboard de Kibana** centralizado.
- **Monitorización** del servicio de docker **via Metricbeat**.
- Realización de consultas **mediante un RAG**.
- Levantar todo mediante **Docker Compose**.

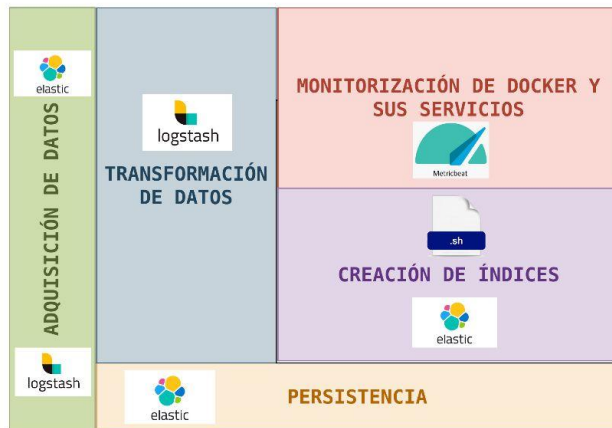
# Esquema de la arquitectura

## CAPA DATOS

### CAPA EDGE



Script generador de datos de Python



### CAPA VISUALIZACIÓN



DASHBOARD SOBRE LAS PLANTAS INDUSTRIALES DE TECNALIA Y SUS DATOS METEOROLÓGICOS



DASHBOARD CREADO POR METRICBEAT PARA MONITORIZAR DOCKER Y SUS SERVICIOS



TODOS LOS SERVICIOS LANZADOS CON DOCKER Y DOCKER-COMPOSE



**elasticsearch**



**elasticsearch**

- Script para generar datos de sensores junto a los datos de localización de las sedes.
- Dockerfile para ejecutar automáticamente el script.
- Librería elasticsearch para enviar datos del script a elasticsearch.
- requirements.txt con las librerías necesarias para la ejecución del script.



[datos\\_sedes\\_sensor\\_index](#)

● green

open

1

0

76667

24.52mb

# Script que genera datos de sensores

El script generador de datos de sensores de python obtendrá los siguientes parámetros en un rango determinado:

- humedad = 0 - 100 (%)
- temperatura = 8 - 30 (°C)
- luminosidad = 100 - 1000 (lx)
- presion\_atm = 900 - 1100 (hPa)
- fecha\_obtencion = en formato "%Y-%m-%d %H:%M:%S"



# Dificultades generadas (elasticsearch)

- Necesidad de unir los datos de los sensores junto a cada localización de planta o sede. Para que se encuentren en un mismo índice.
- Generar documentos con los datos de los sensores en cada una de las sedes o plantas a la vez. Es decir, hacer que cada cierto tiempo se generen los datos de sensores en cada una de las sedes o plantas, y no solo de una única planta.





- index.sh para la creación de los distintos índices junto a sus mappings (los cuales he modificado) y su configuración. Se ejecuta en un contenedor de docker llamado init-elastic. Una vez creados los índices, se cierra el contenedor.
- Pipelines logstash.config y logstash2.config con el plugin http\_poller para obtener documentos de la API web.
- Uso de pipelines.yml para ejecutar ambas pipelines.

<input type="checkbox"/>	<a href="#">datos_meteorologicos_actuales_sedes_index</a>	<span>● green</span>	open	1	0	1952	1.14mb
<input type="checkbox"/>	<a href="#">datos_meteorologicos_dia_siguiente_sedes_index</a>	<span>● green</span>	open	1	0	531	663.33kb

# API web utilizada



## Free Weather API

Open-Meteo is an open-source weather API and offers free access for non-commercial use. No API key required. Start using it now!

[Features](#)[Documentation](#)

<https://open-meteo.com/>



# Dificultades generadas (logstash)

- Encontrar una API gratuita que ofrezca una buena cantidad de solicitudes para obtener datos meteorológicos.
- Necesidad de implementar una URL por cada una de las sedes o plantas existentes.
- Necesidad de implementar index.sh para que no se creen los mappings de los distintos índices automáticamente.
- Ejecutar dos pipelines a la vez, lo he logrado realizar con el archivo pipelines.yml.

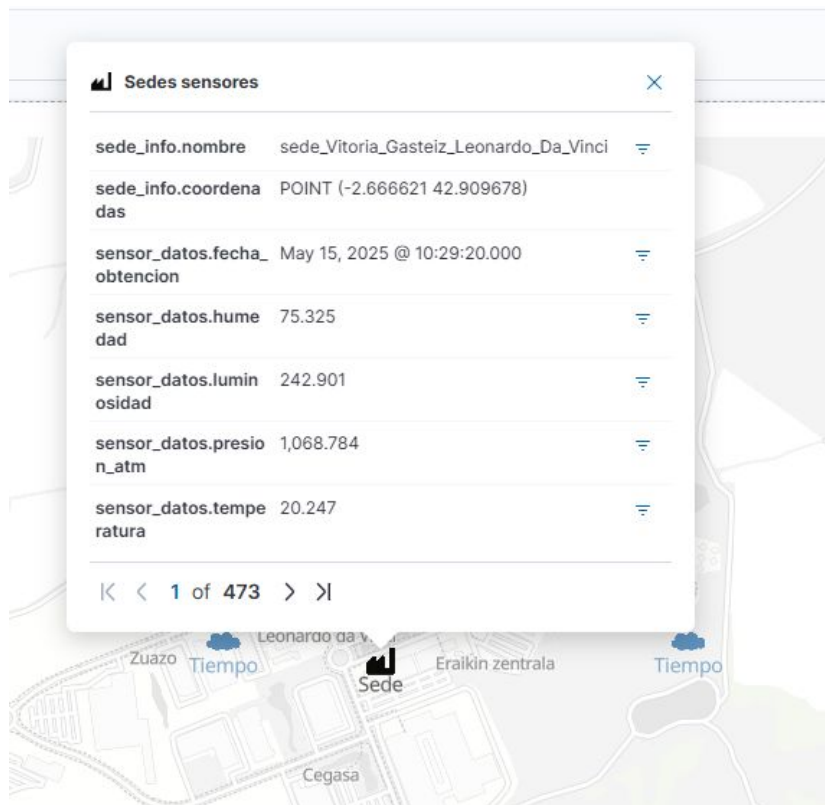


**kibana**

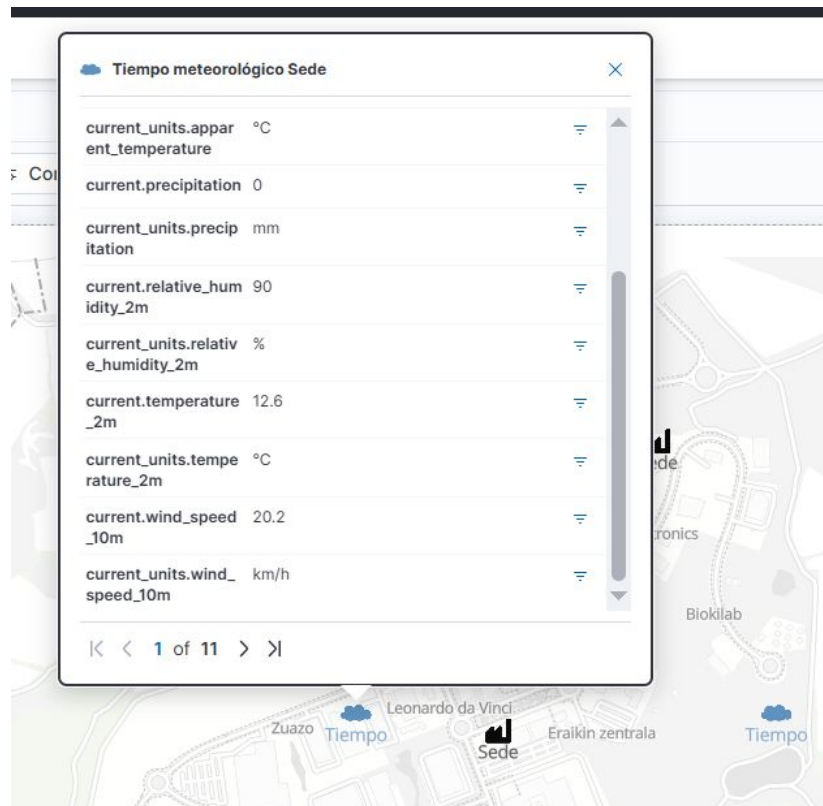
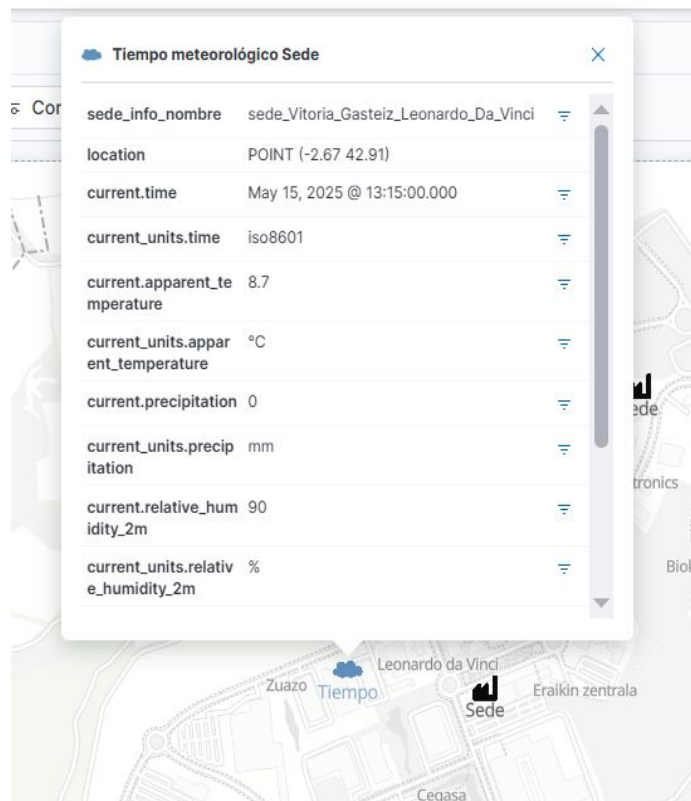


- Dashboard centralizado.
- Creación de un mapa con los datos meteorológicos y los datos de los sensores en cada localización. Seleccionando cada sede o planta se podrán ver los valores obtenidos por cada sensor junto a su fecha.
- Gráficos sobre la información obtenida por los sensores de cada sede o planta en la última hora.
- Gráficos sobre la información meteorológica actual y pasada.
- Gráficos sobre la información meteorológica del siguiente día.

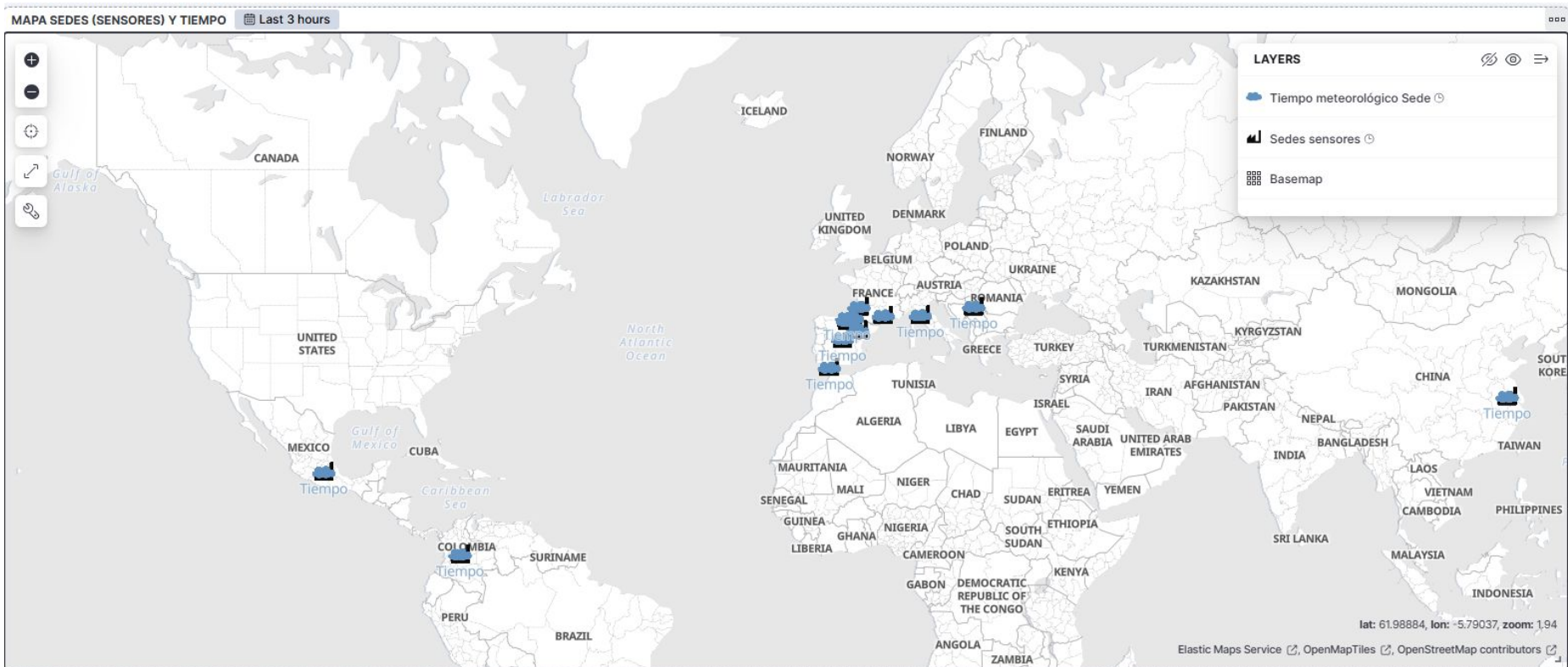
# Datos sensores por sede o planta en el mapa



# Datos meteorológicos por sede o planta en el mapa



# Datos por sede o planta en el mapa

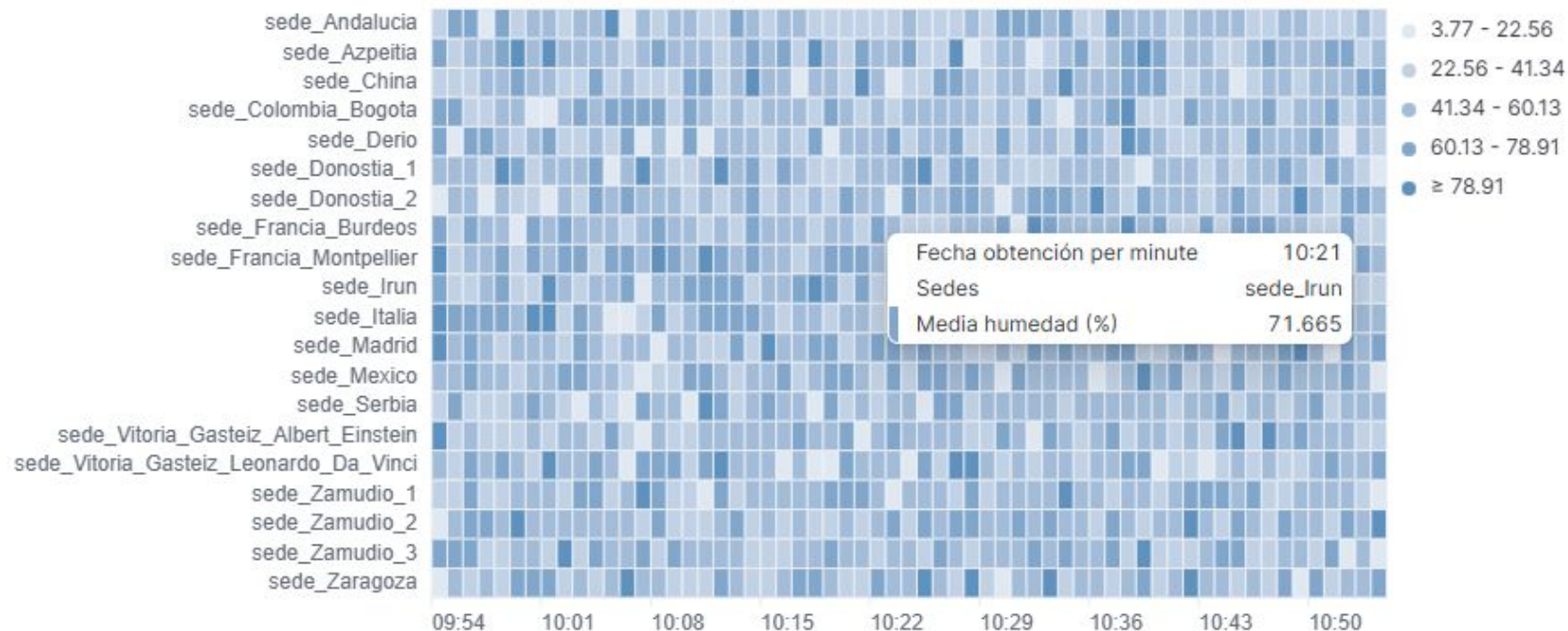


# Gráficos de datos de los sensores

MEDIA DE HUMEDAD DE LAS SEDES (sensores)

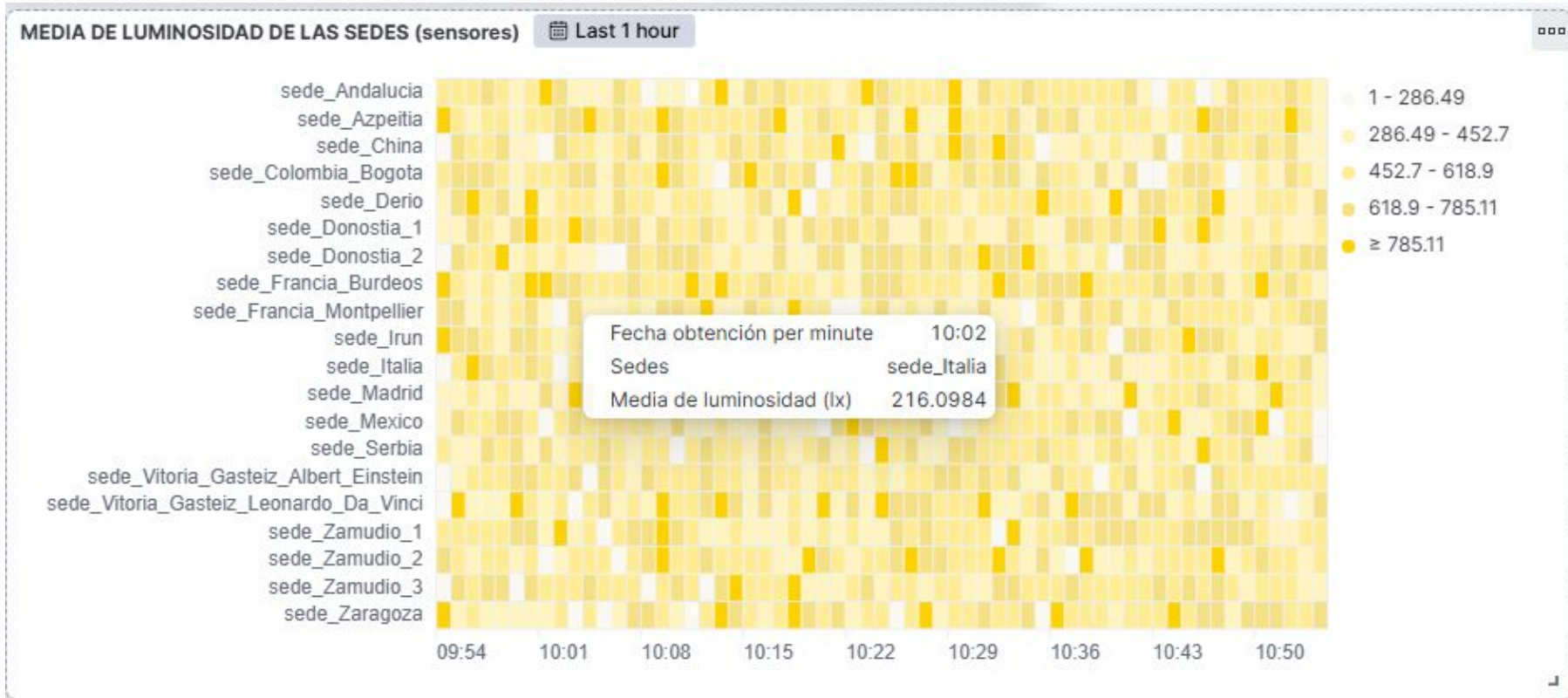
📅 Last 1 hour

☰





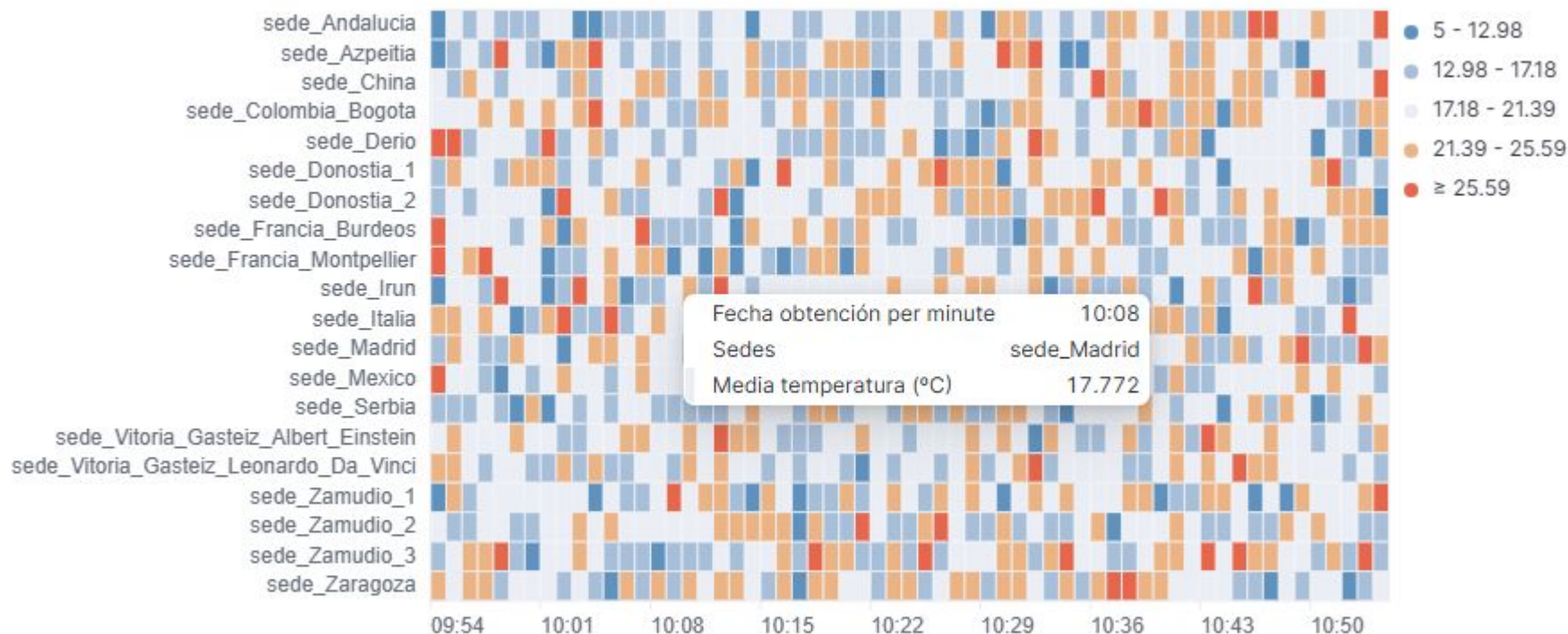
# Gráficos de datos de los sensores



# Gráficos de datos de los sensores

MEDIA DE TEMPERATURA DE LAS SEDES (sensores)

Last 1 hour



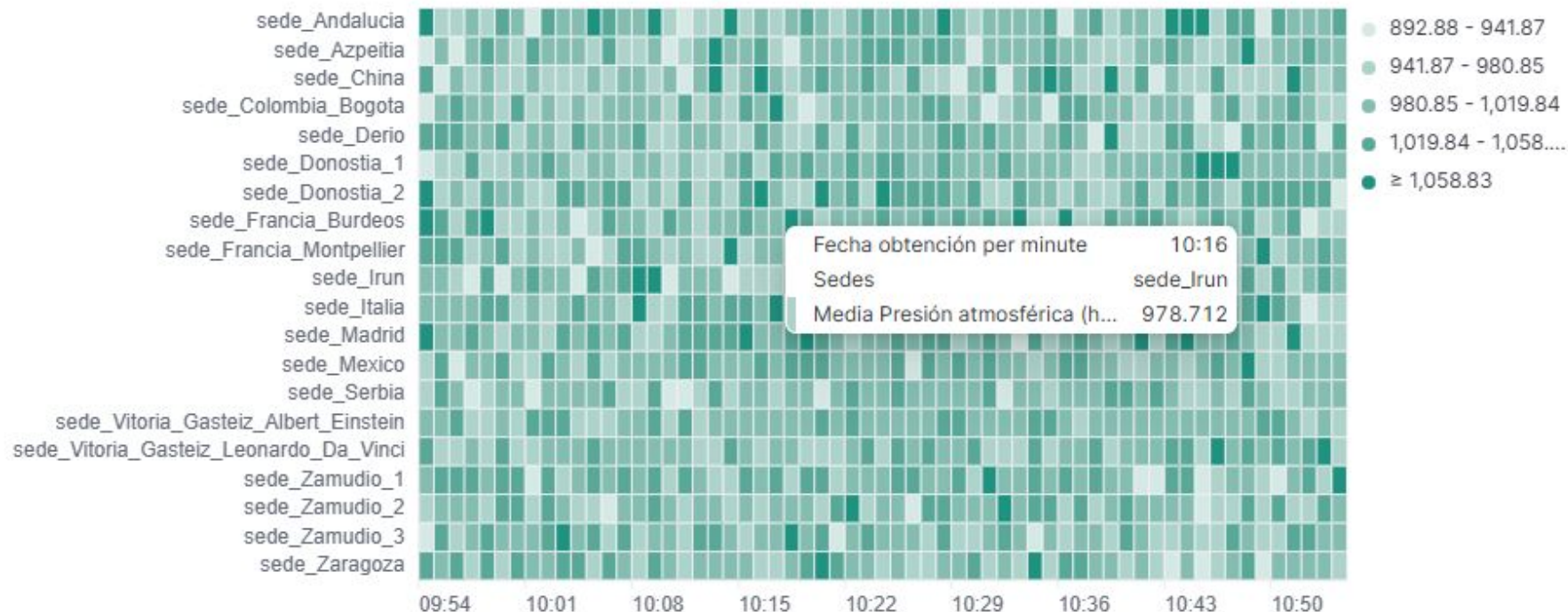


# Gráficos de datos de los sensores

MEDIA DE LA PRESIÓN ATMOSFÉRICA DE LAS SEDES (sensores)

Last 1 hour

☰

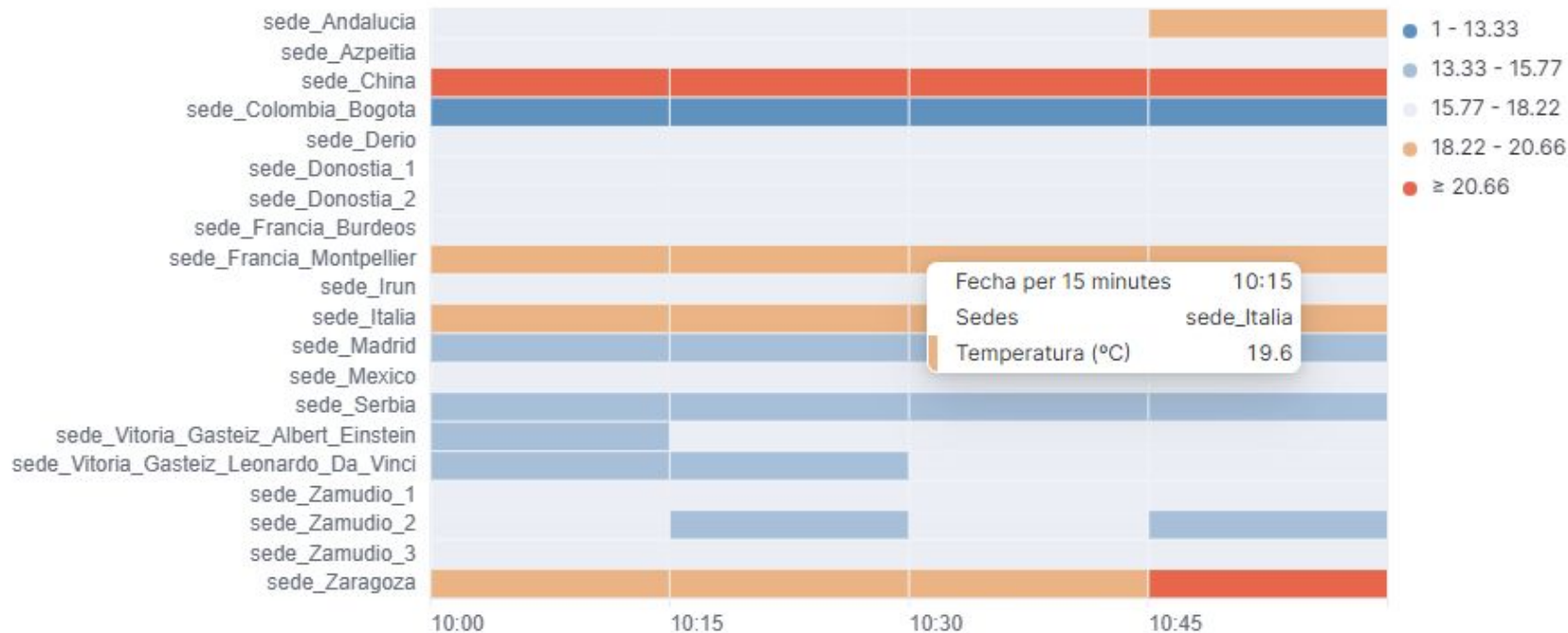


# Gráficos de datos meteorológicos por localización

## TEMPERATURA POR CADA SEDE (DATOS METEOROLÓGICOS)

📅 Last 1 hour rounded to the hour

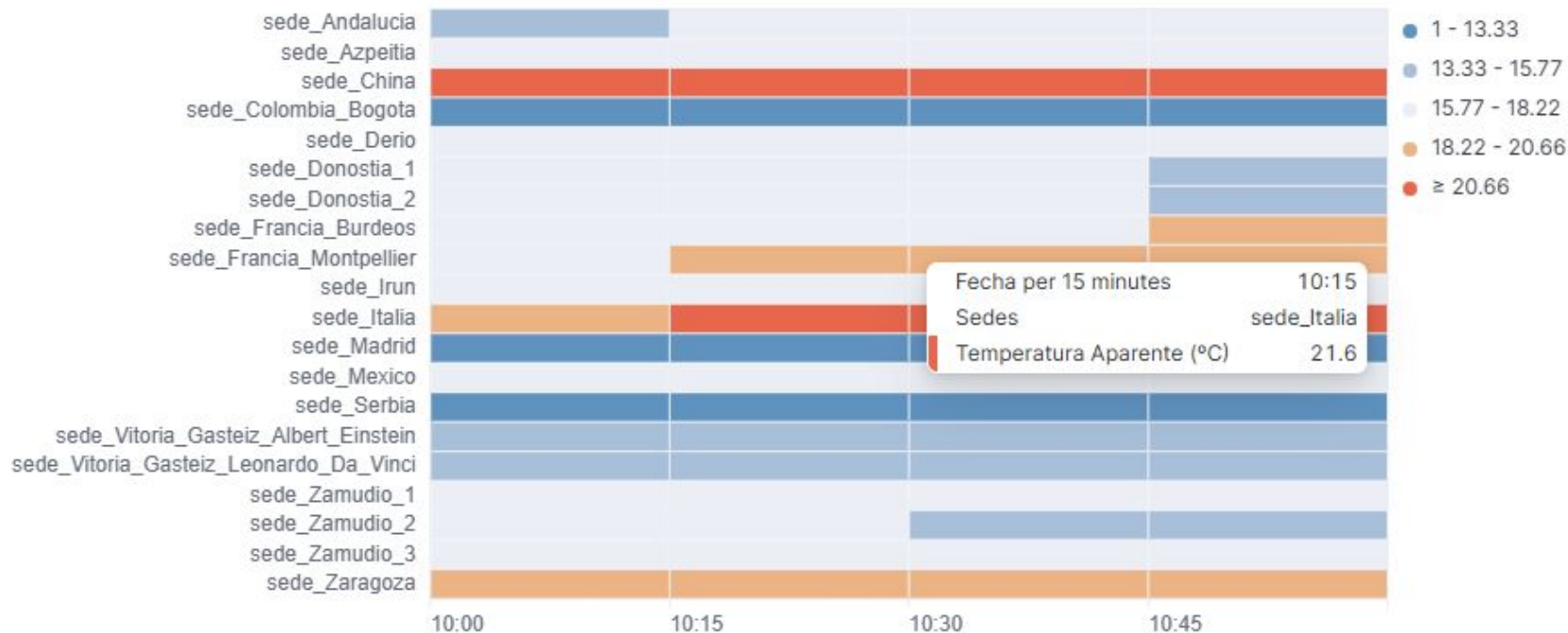
☰



# Gráficos de datos meteorológicos por localización

TEMPERATURA APARENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

Last 1 hour rounded to the hour

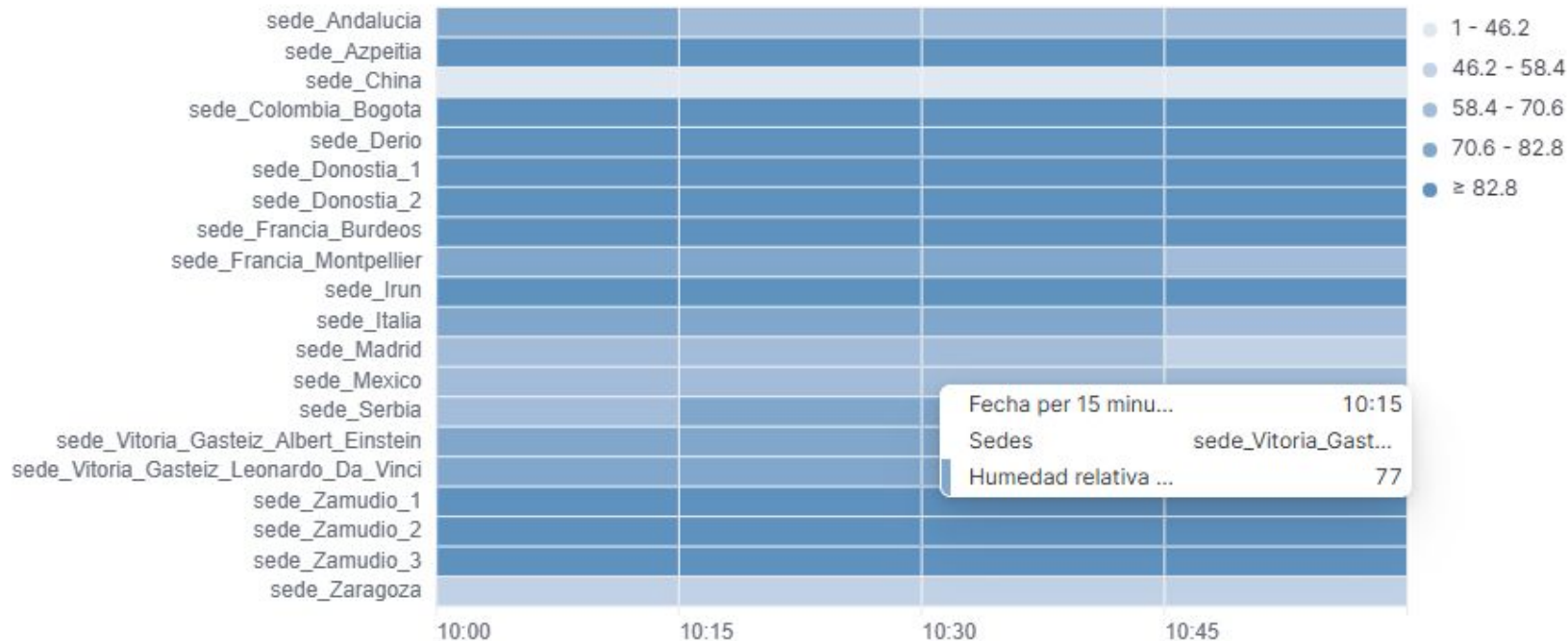


# Gráficos de datos meteorológicos por localización

HUMEDAD RELATIVA POR CADA SEDE (DATOS METEOROLÓGICOS)

Last 1 hour rounded to the hour

☰



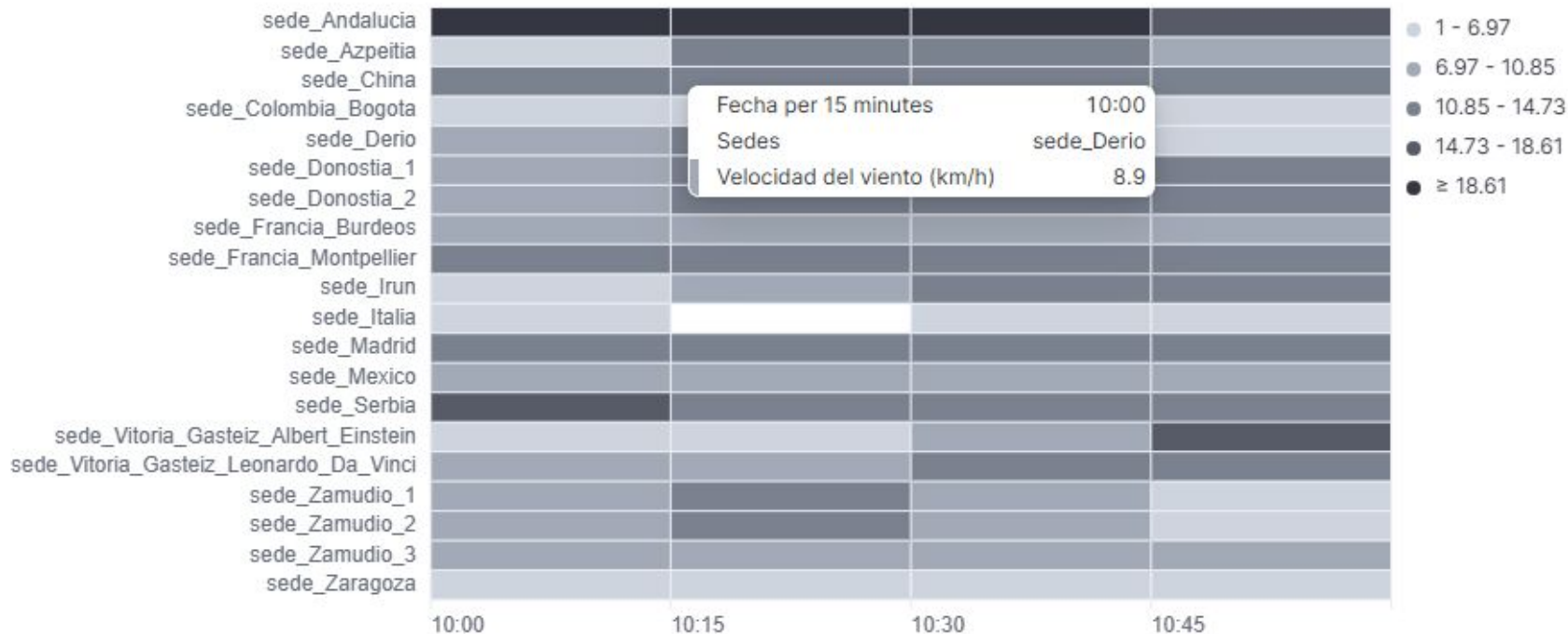
└

# Gráficos de datos meteorológicos por localización

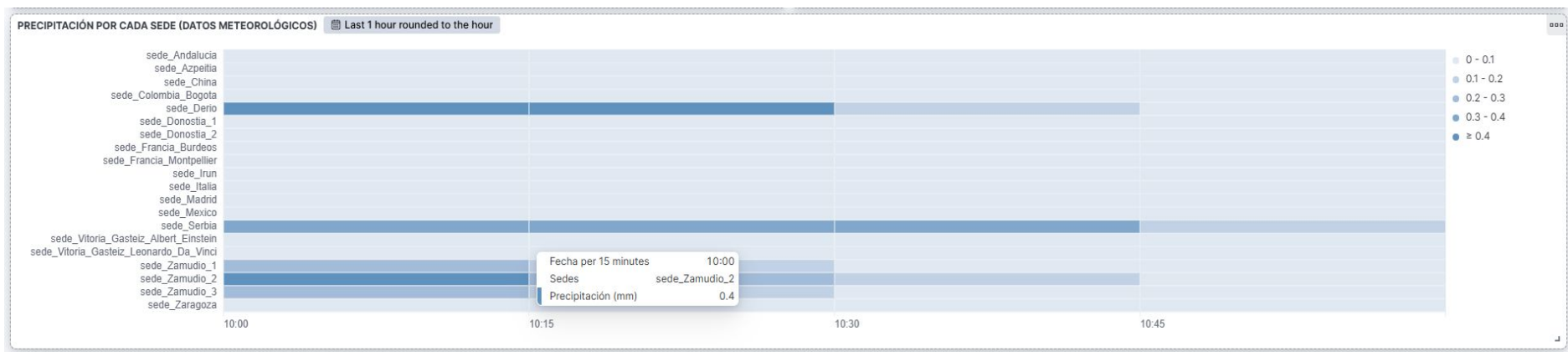
## VELOCIDAD DEL VIENTO POR CADA SEDE (DATOS METEOROLÓGICOS)

📅 Last 1 hour rounded to the hour

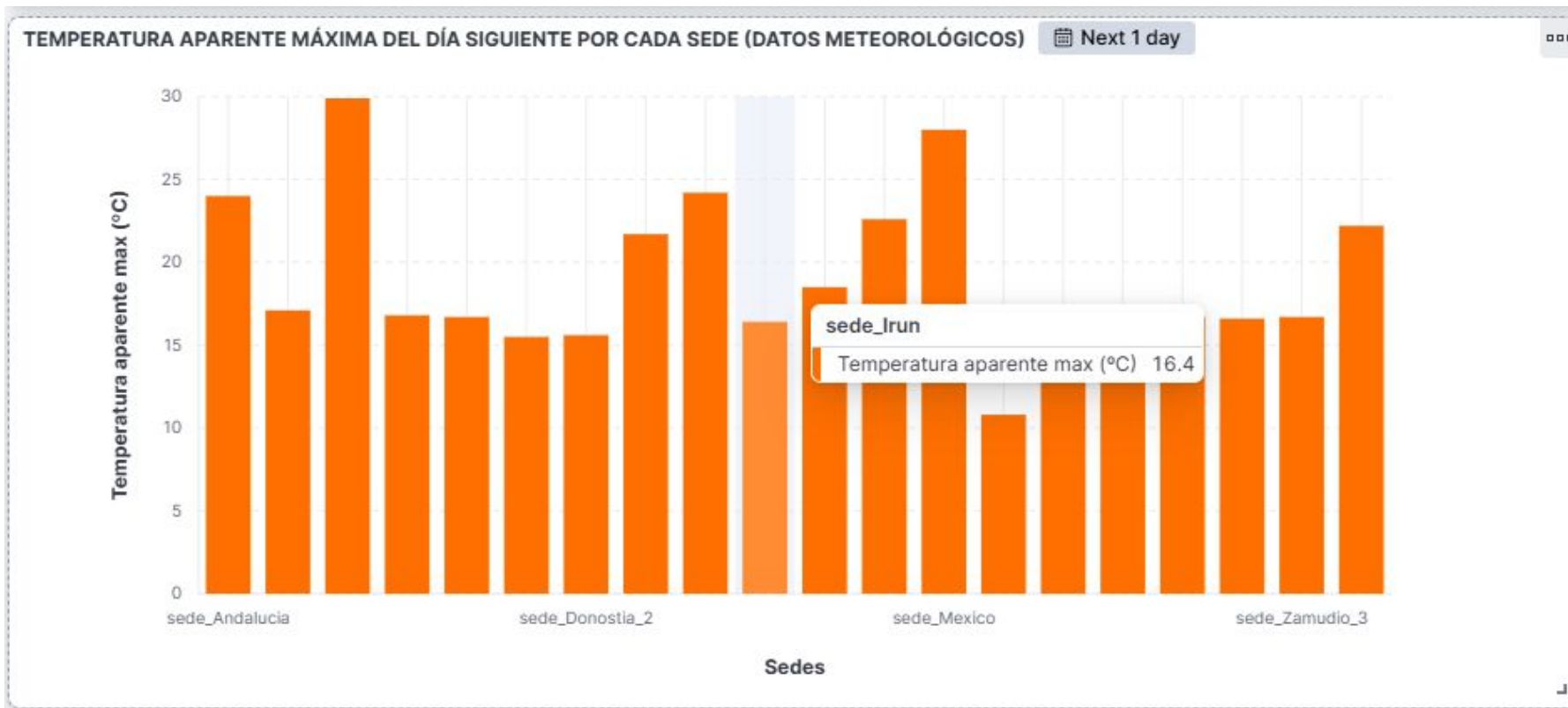
☰



# Gráficos de datos meteorológicos por localización



# Gráficos de datos meteorológicos por localización

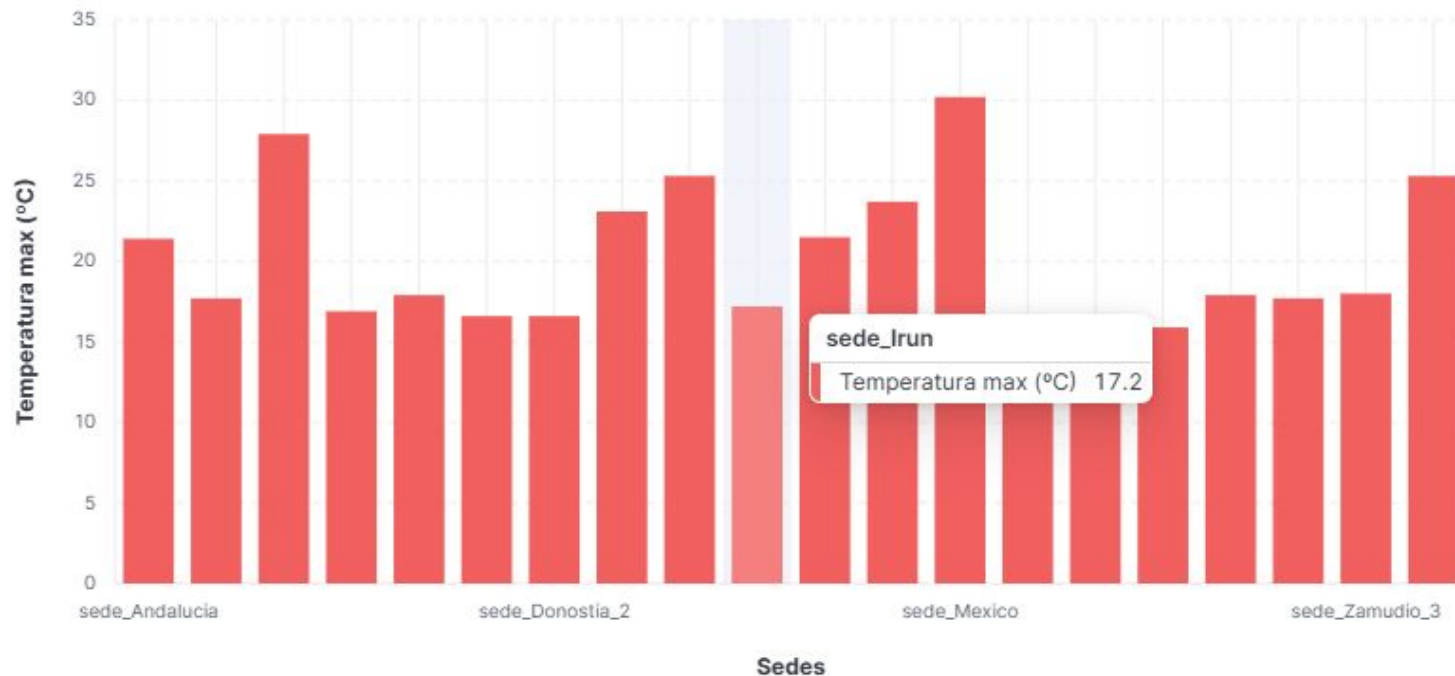




# Gráficos de datos meteorológicos por localización

TEMPERATURA MÁXIMA DEL DÍA SIGUIENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

Next 1 day



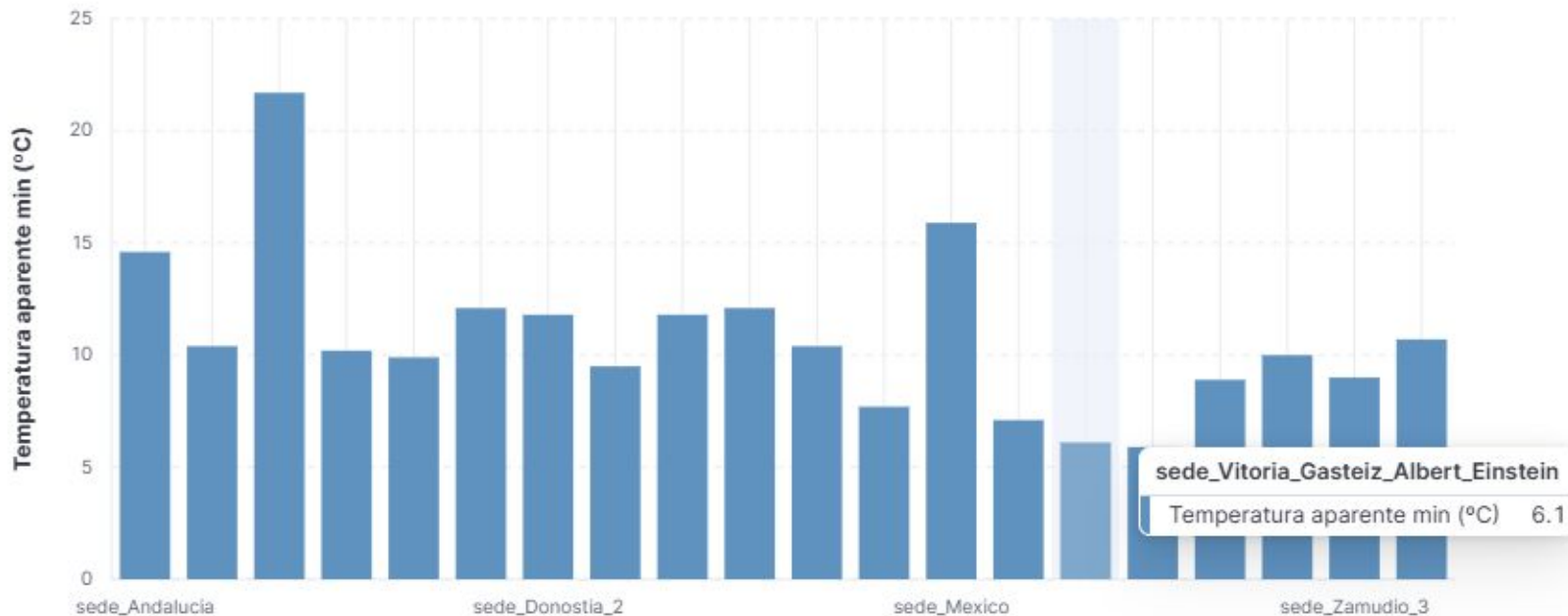


# Gráficos de datos meteorológicos por localización

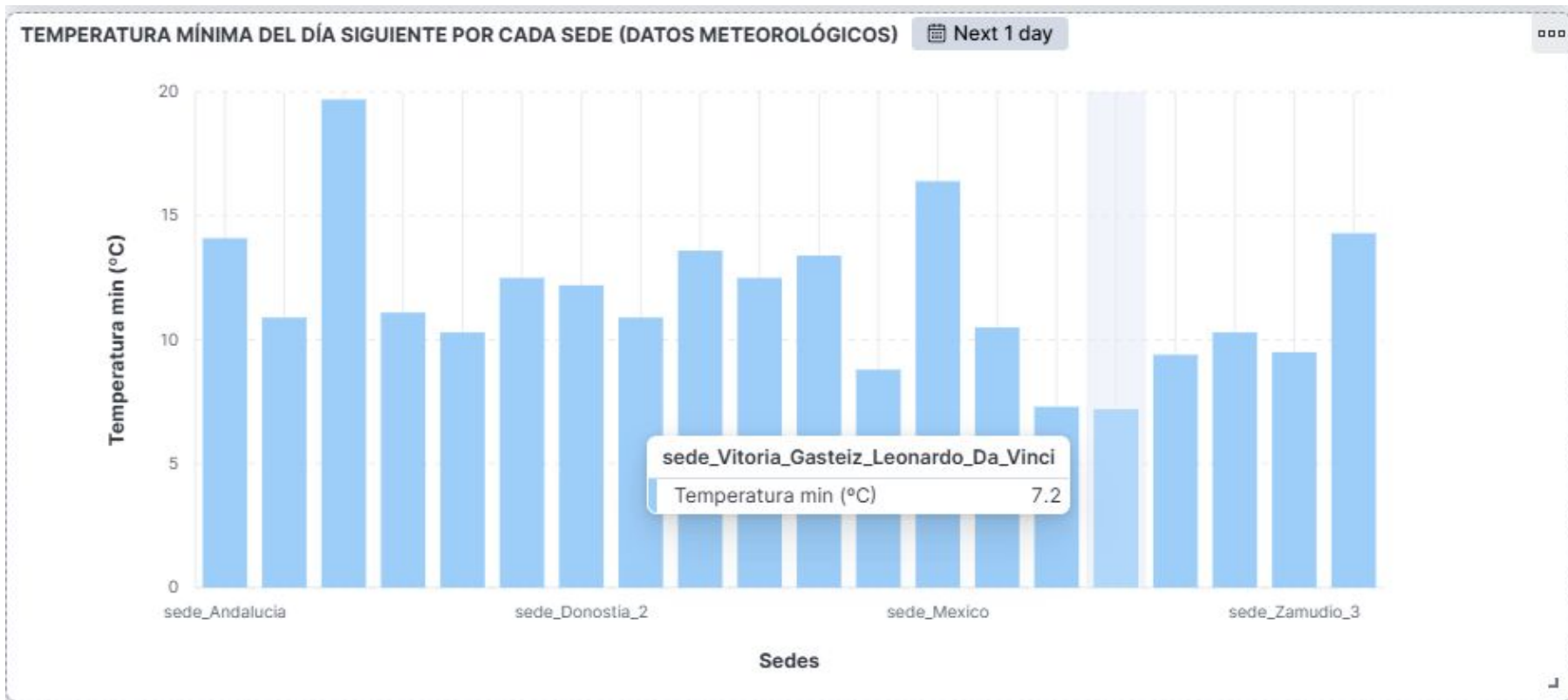
TEMPERATURA APARENTE MÍNIMA DEL DÍA SIGUIENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

Next 1 day

☰



# Gráficos de datos meteorológicos por localización

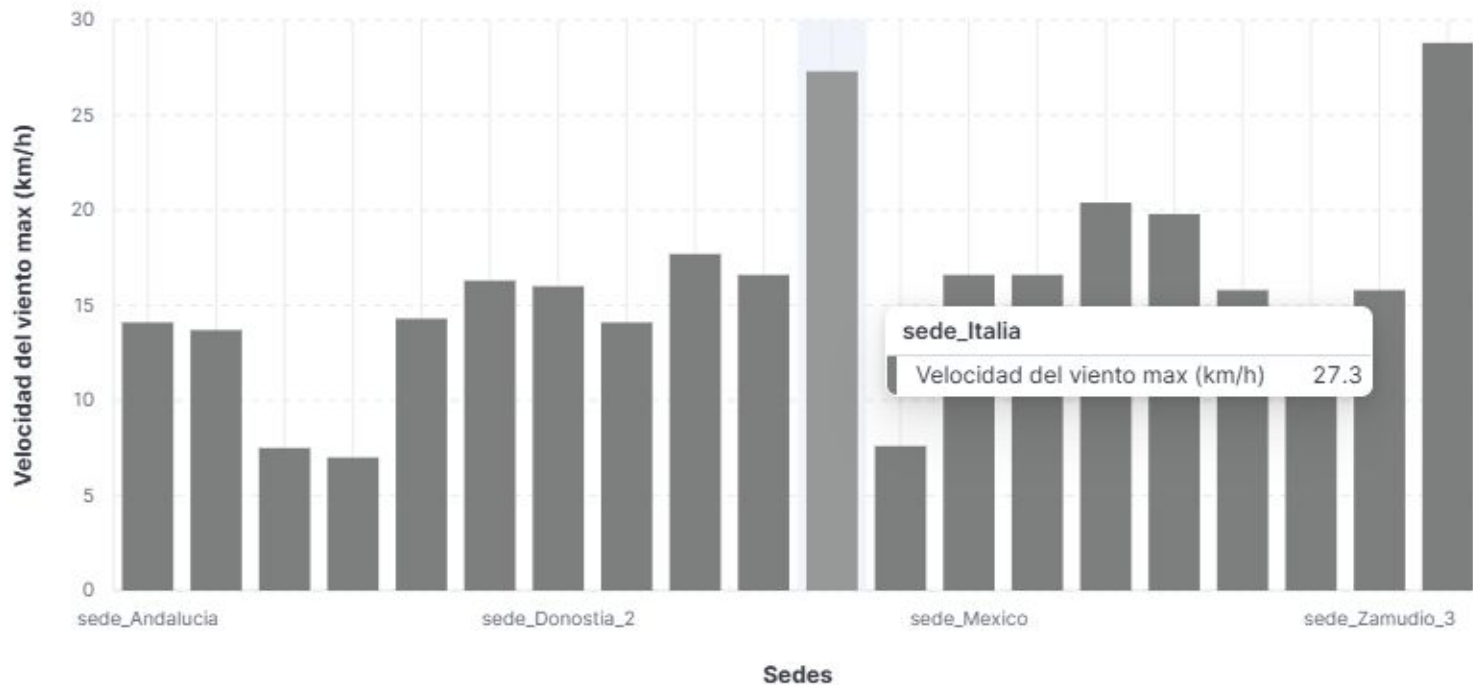


# Gráficos de datos meteorológicos por localización

VELOCIDAD DEL VIENTO MÁXIMA DEL DÍA SIGUIENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

Next 1 day

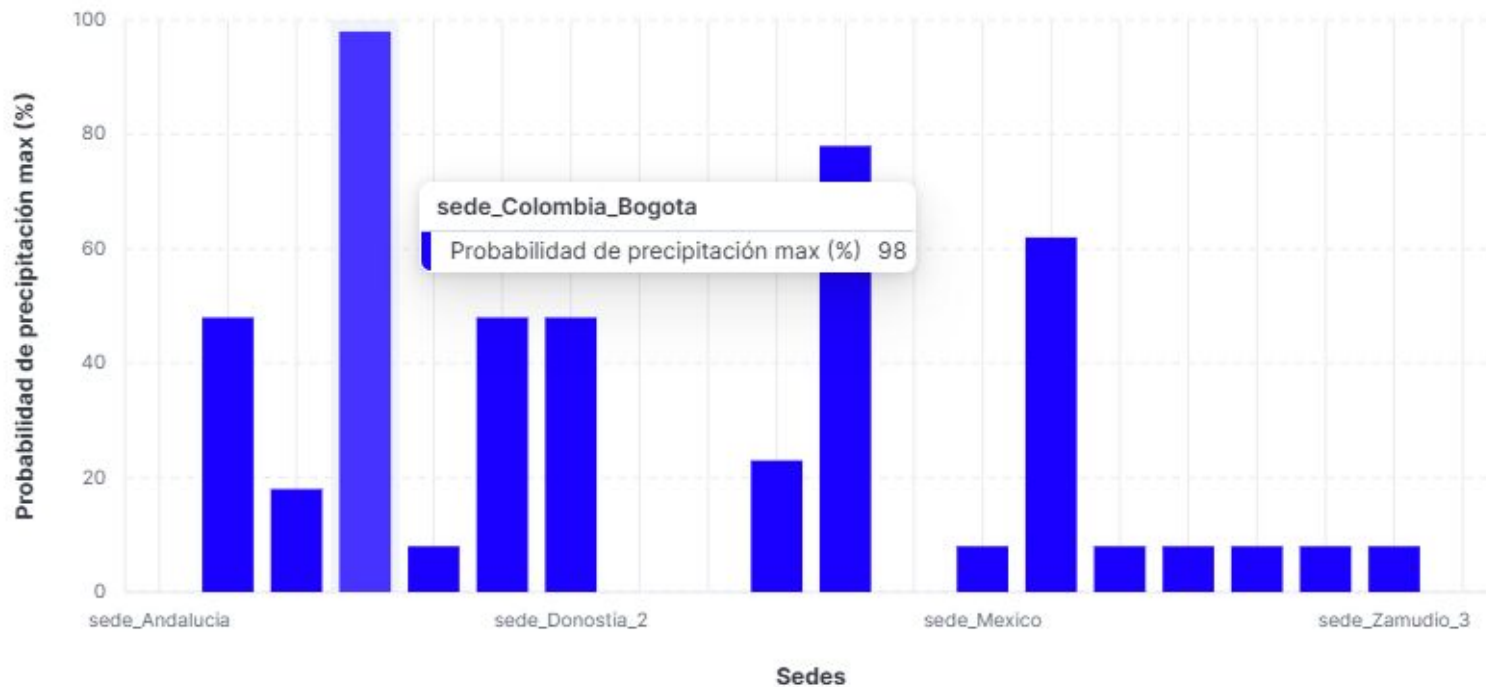
☰



# Gráficos de datos meteorológicos por localización

PROBABILIDAD DE PRECIPITACIÓN MÁXIMA DEL DÍA SIGUIENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

Next 1 day



# Desplegables para filtrar datos

|| Nombre sede (Datos Sensores) Any

MAPA SEDES (SENSORES) Y TIEMPO

Search: Starts with...

20 options

sede_Colombia_Bogota	288
<u>sede_Derio</u>	288
sede_Donostia_1	288
sede_Donostia_2	288
sede_Francia_Burdeos	288
sede_Francia_Montpellier	288
sede_Irlanda	288

Include Exclude

UNITED STATES North Atlantic

# Desplegables para filtrar datos

|| Nombre sede (Datos Meteorológicos)

Any

Q Starts with...

20 options

Exists

sede\_Andalucia7

sede\_Azpeitia7

sede\_Colombia\_Bogota7


sede\_Derio7

sede\_Donostia\_17

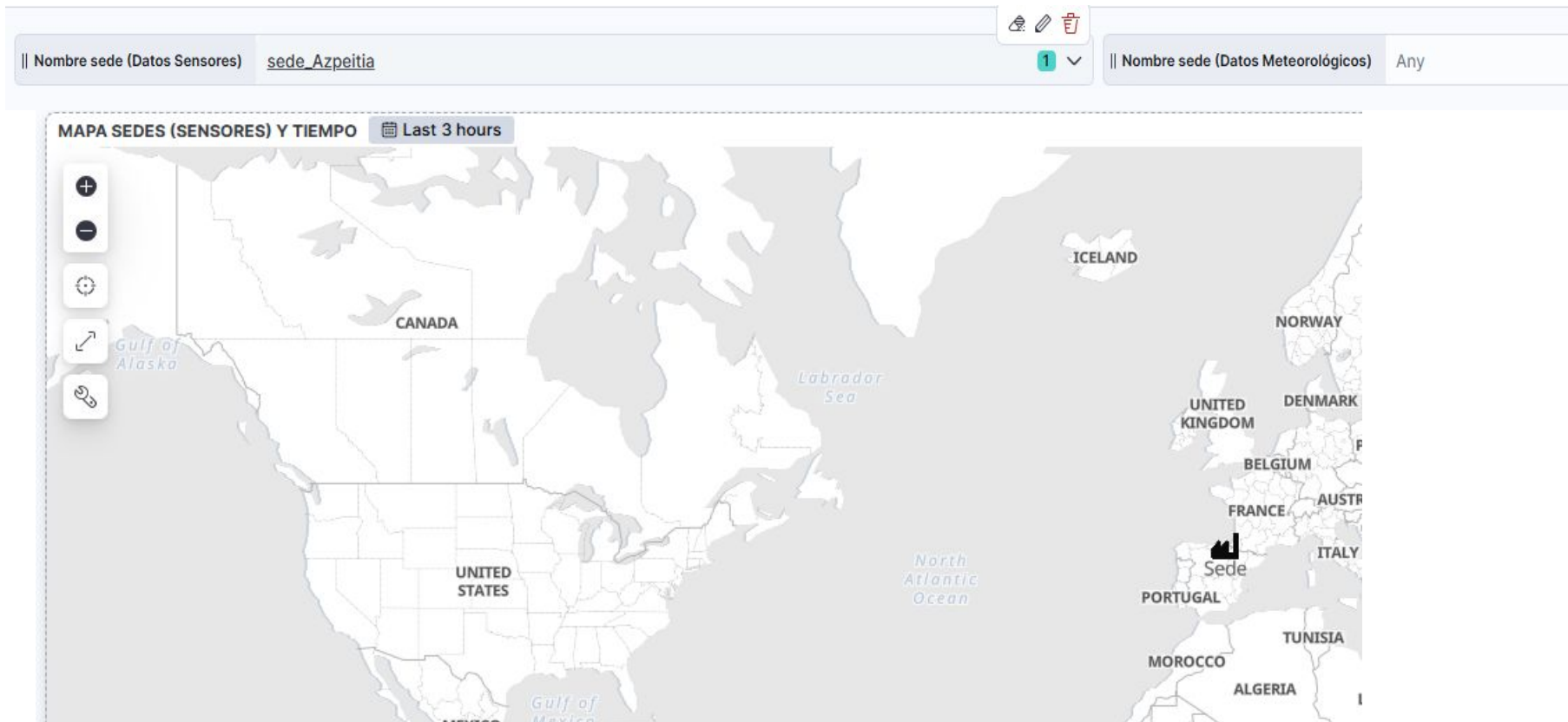
sede\_Donostia\_27

Include

Exclude



# Desplegables para filtrar datos

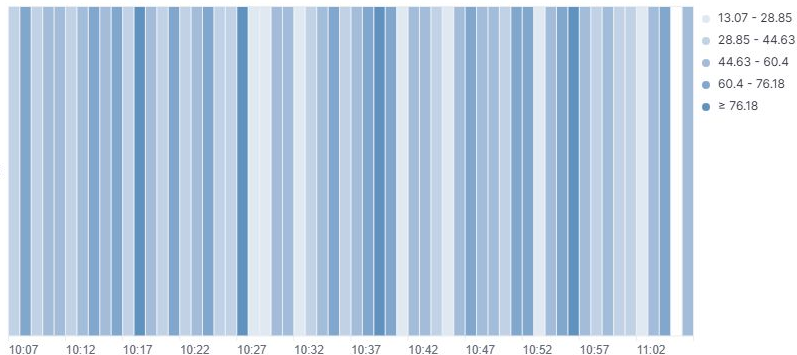


# Desplegables para filtrar datos

MEDIA DE HUMEDAD DE LAS SEDES (sensores)

Last 1 hour

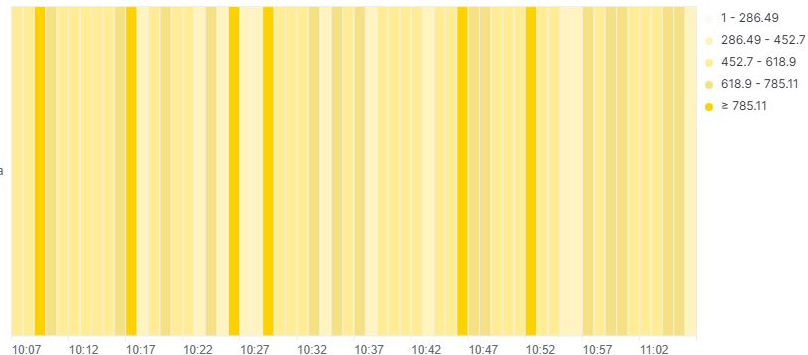
sede\_Azpeitia



MEDIA DE LUMINOSIDAD DE LAS SEDES (sensores)

Last 1 hour

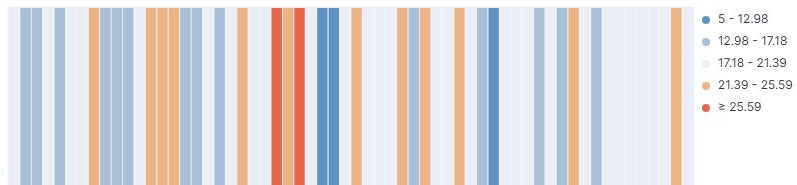
sede\_Azpeitia



MEDIA DE TEMPERATURA DE LAS SEDES (sensores)

Last 1 hour

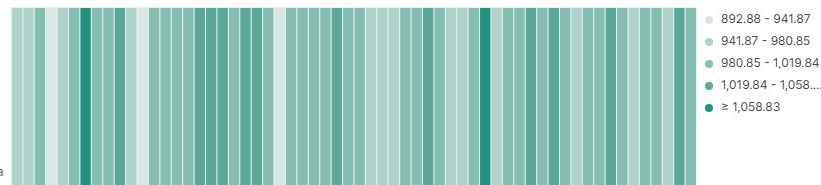
sede\_Azpeitia



MEDIA DE LA PRESIÓN ATMOSFÉRICA DE LAS SEDES (sensores)

Last 1 hour

sede\_Azpeitia

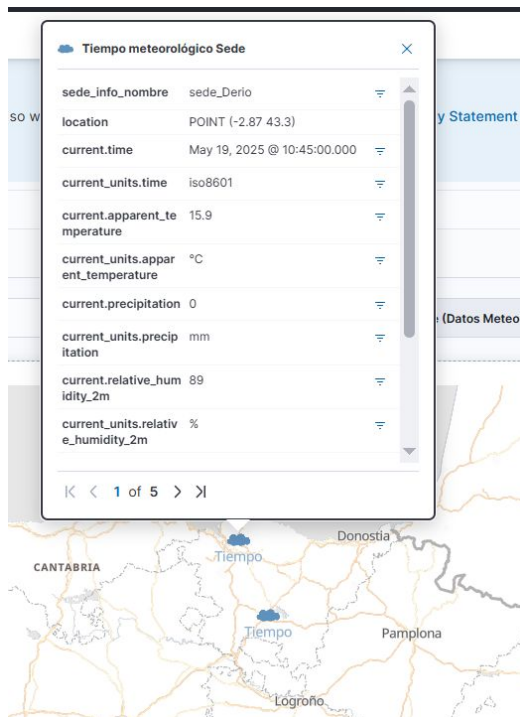




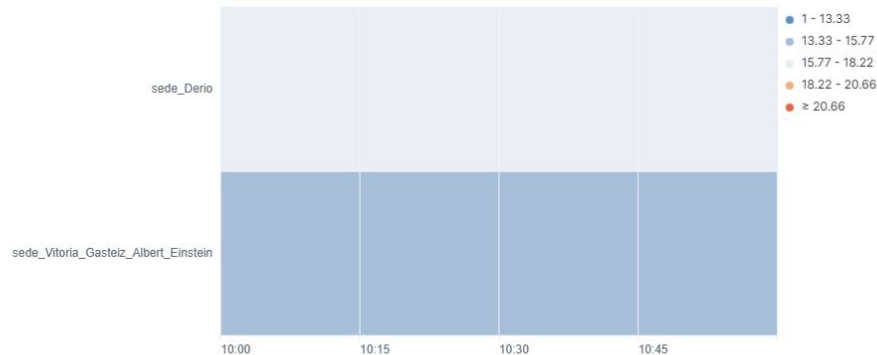
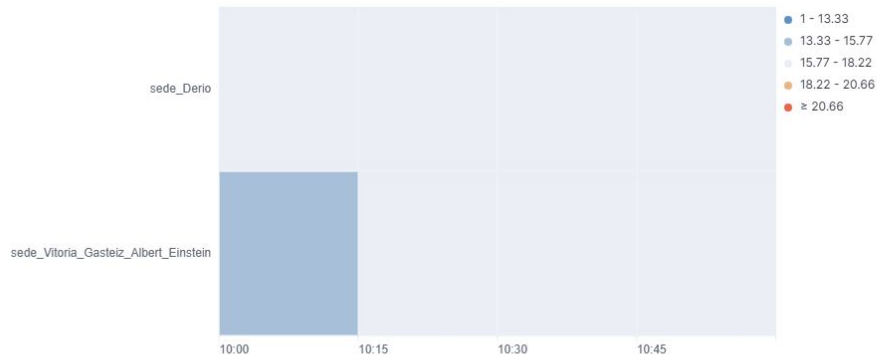
# Desplegables para filtrar datos

|| Nombre sede (Datos Meteorológicos)

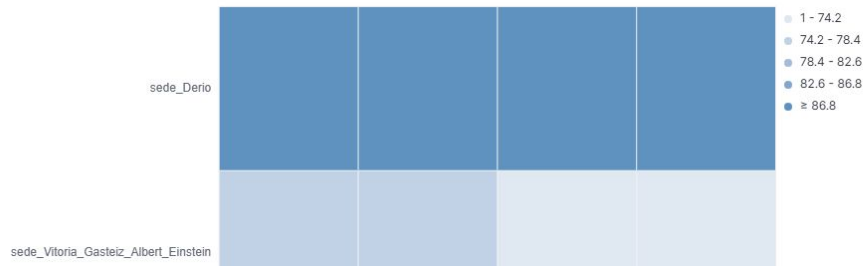
sede\_Derio, sede\_Vitoria\_Gasteiz\_Albert\_Einstein



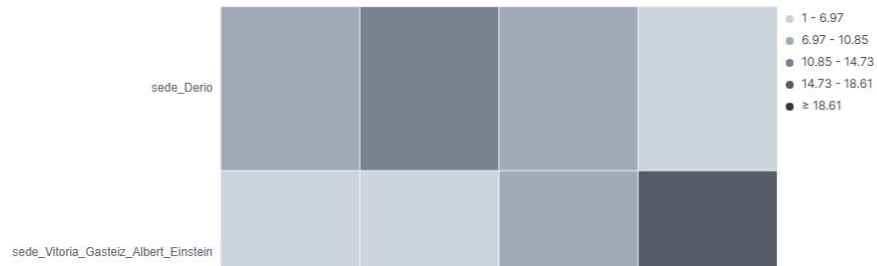
# Desplegables para filtrar datos



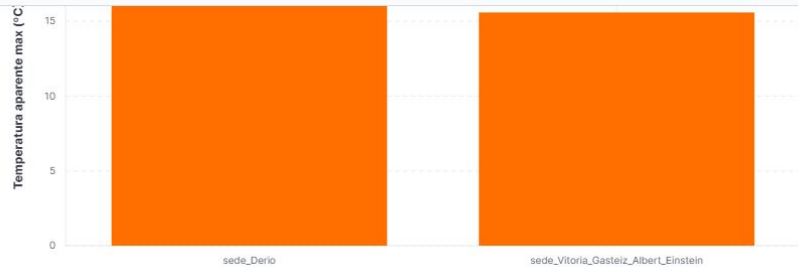
HUMEDAD RELATIVA POR CADA SEDE (DATOS METEOROLÓGICOS) Last 1 hour rounded to the hour



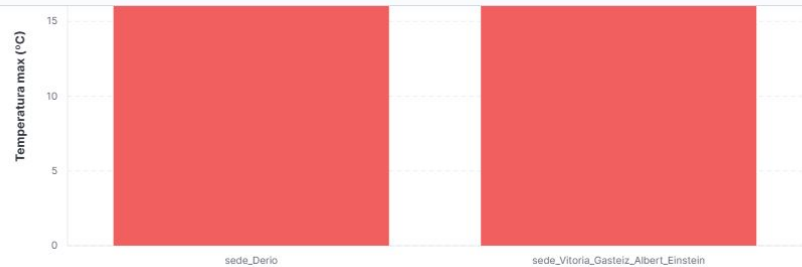
VELOCIDAD DEL VIENTO POR CADA SEDE (DATOS METEOROLÓGICOS) Last 1 hour rounded to the hour



# Desplegables para filtrar datos



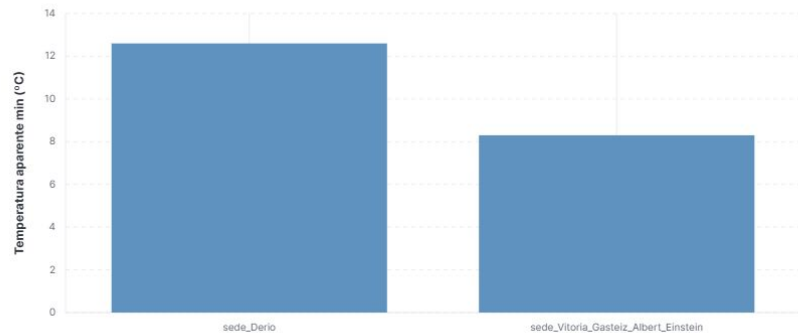
Sedes



Sedes

TEMPERATURA APARENTE MÍNIMA DEL DÍA SIGUIENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

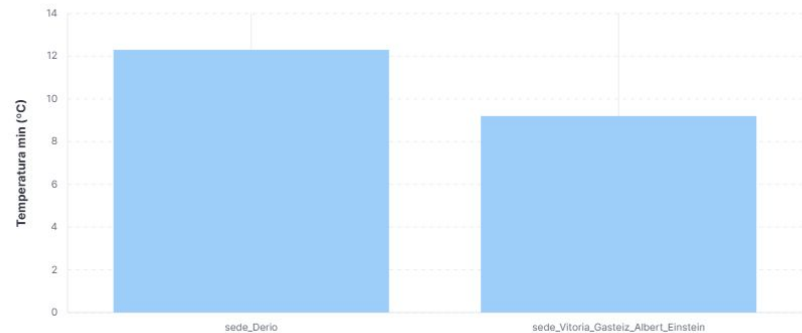
Next 1 day



Sedes

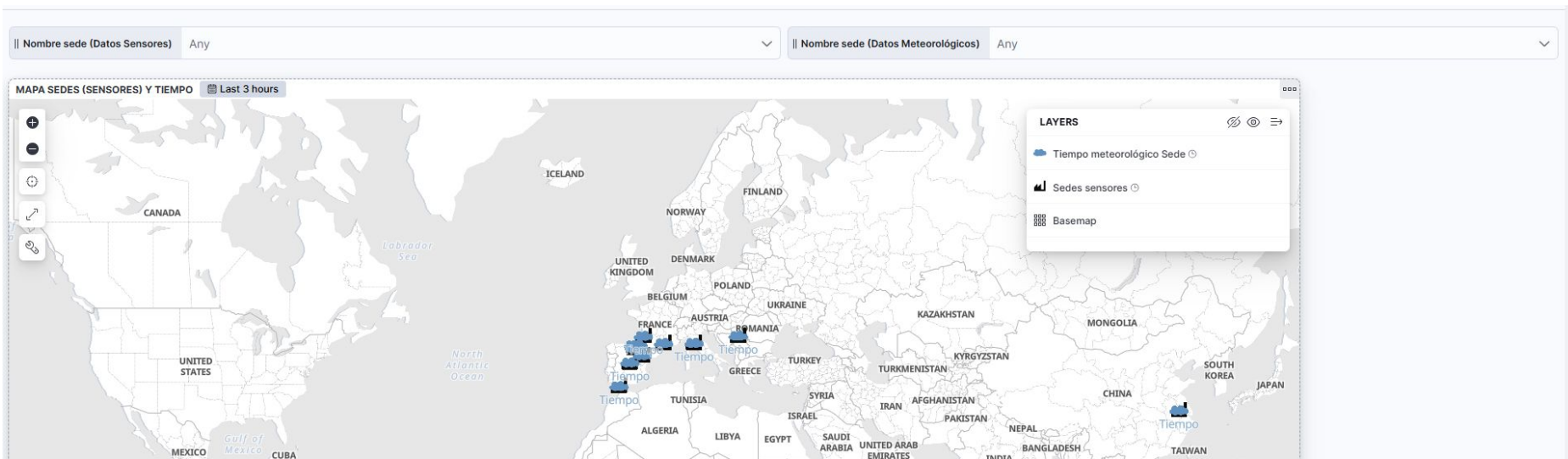
TEMPERATURA MÍNIMA DEL DÍA SIGUIENTE POR CADA SEDE (DATOS METEOROLÓGICOS)

Next 1 day



Sedes

# Como ver toda la información con los desplegables



# Dificultades generadas (kibana)

- Necesidad de tener un mapping adecuado en cada índice para visualizar los datos.
- Cambio de datos de tipo float a geo\_point para poder visualizar los datos en el mapa que ofrece Kibana. Para ello he tenido que añadir un campo llamado location mediante logstash.
- Creación de los gráficos para ver los datos meteorológicos correctamente. Para ello he tenido que obtener de la API los datos actuales y del siguiente día.
- La API responde a la solicitud de las coordenadas específicas con los de una zona cercana.



# Monitorización del servicio de docker vía metricbeat

Se ha implementado metricbeat, con objetivo de monitorizar Docker y cada servicio.

Para ello, se ha tenido que modificar el Docker-compose añadiendo el servicio de metricbeat y se ha creado un archivo de configuración propio de metricbeat llamado metricbeat.yml .



# Índices generados sobre los servicios que ofrece Docker

<input type="checkbox"/> Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
<input type="checkbox"/> <a href="#">.monitoring-es-7-2024.02.01</a>	<span>● green</span>	open	1	0	11540	7.22mb	
<input type="checkbox"/> <a href="#">.monitoring-es-7-2024.02.05</a>	<span>● green</span>	open	1	0	5054	4.09mb	
<input type="checkbox"/> <a href="#">.monitoring-kibana-7-2024.02.01</a>	<span>● green</span>	open	1	0	2058	1.53mb	
<input type="checkbox"/> <a href="#">.monitoring-kibana-7-2024.02.05</a>	<span>● green</span>	open	1	0	702	1.1mb	
<input type="checkbox"/> <a href="#">.monitoring-logstash-7-2024.02.01</a>	<span>● green</span>	open	1	0	13237	1.34mb	
<input type="checkbox"/> <a href="#">.monitoring-logstash-7-2024.02.05</a>	<span>● green</span>	open	1	0	4544	1.84mb	

Y el índice propio de metricbeat (en este caso obtiene información sobre el sistema y Docker):

<input type="checkbox"/> <a href="#">.ds-metricbeat-8.11.1-2024.02.08-000001</a>	<span>● yellow</span>	open	1	1	241250	292.21mb	<a href="#">metricbeat-8.11.1</a>
--	-----------------------	------	---	---	--------	----------	-----------------------------------

# Dashboard sobre docker generado mediante Metricbeat

## Docker containers [Metricbeat Docker] ECS

Export

Name	CPU usage (%)	DiskIO	Mem (%)	Mem RSS	Number of Contai...
metricbeat	8.1%	0B	0.7%	0B	2
script_container	14.4%	0B	0.2%	0B	2
log	80.1%	0B	5.7%	0B	2
es01	115.4%	0B	61.5%	0B	2
kib01	197.4%	0B	4.7%	0B	2
	415.4%	0B	72.8%	0B	10

## Number of Containers [Metricbeat Docker] ECS

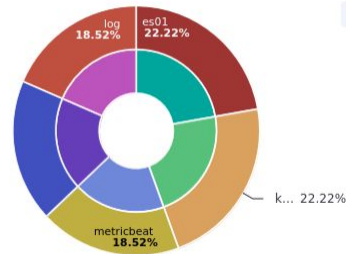
5 0 1  
Running Paused Stopped

## Docker containers per host [Metricbeat Docker] ECS



- 6981dbb59b27
- ee17fc6373bb

## Docker images and names [Metricbeat Docker] ECS

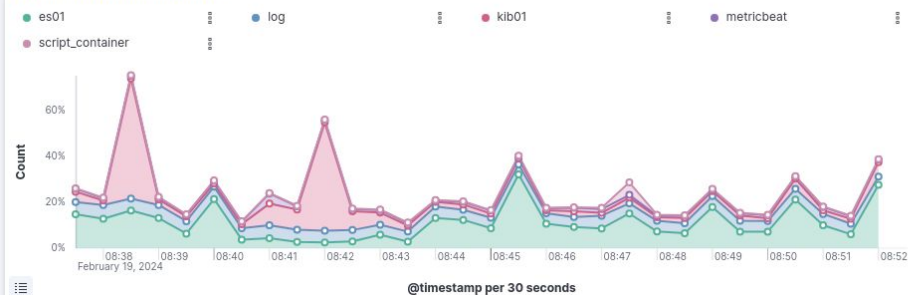


- docker.elastic.co/elasticsearch/elasticsearch:8.11.1
- docker.elastic.co/kibana/kibana:8.11.1

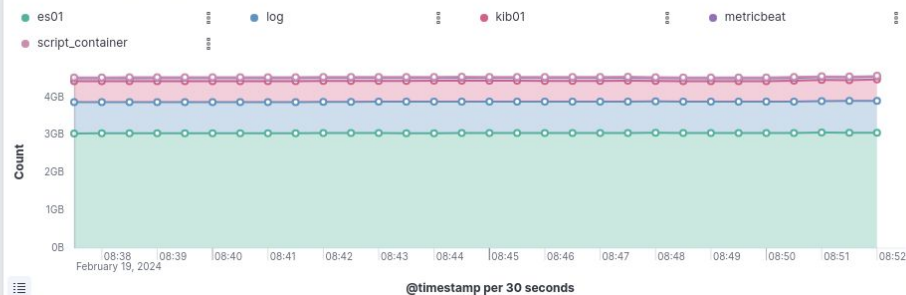


# Dashboard sobre docker generado mediante Metricbeat

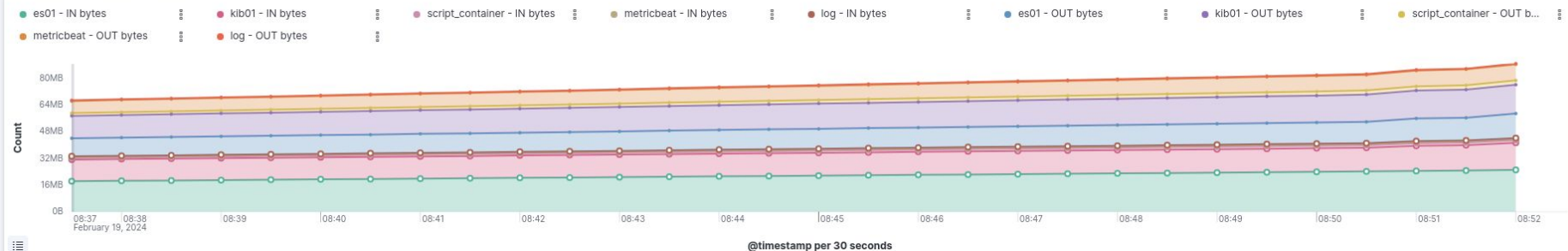
CPU usage [Metricbeat Docker] ECS



Memory usage [Metricbeat Docker] ECS

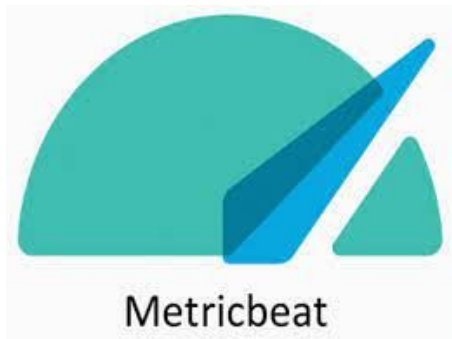


Network IO [Metricbeat Docker] ECS



# Dificultades generadas (metricbeat)

- Observar que índices han sido creados por metricbeat. (Estaban ocultos).
- Necesidad de implementarlo con la versión utilizada.(La manera de implementar metricbeat cambia bastante de una versión a otra)
- Fallo de conexión entre metricbeat y kibana, metricbeat se iniciaba antes.



# RAG

2 maneras de poder hacerlo:

Fragmentar la información en bases de datos vectoriales



Se construye un contexto con los fragmentos que más cerca estén vectorialmente de la pregunta.

Enfocado a grandes cantidades de información

Meter toda la información en el contexto directamente

Si el modelo es lo suficientemente potente, es lo mejor

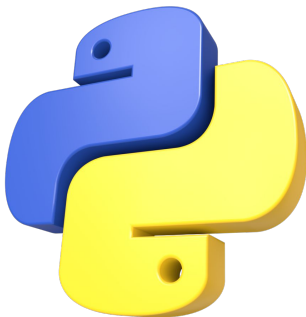
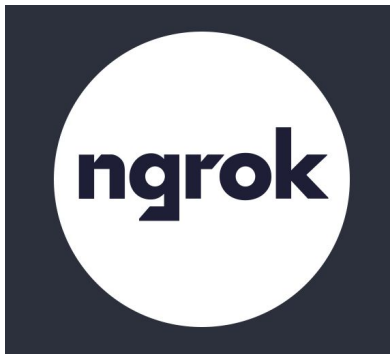
**RAG**

Google Colaboratory



# RAG

Otras tecnologías usadas para el RAG:



# Nuestro RAG

## Consultas a Sedes de TecNALIA

Tu pregunta:

Ej: ¿Qué sede tuvo mayor temperatura ayer?

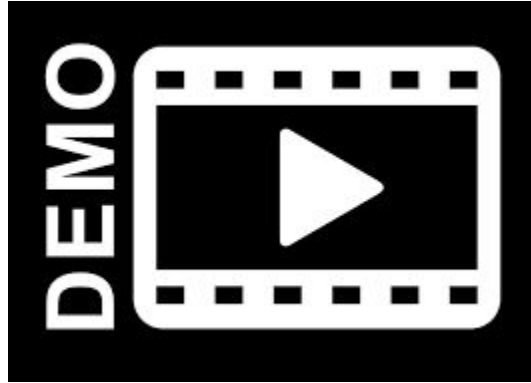
Preguntar

Respuesta:

Aquí aparecerá la respuesta...

# Vídeo demo

[https://drive.google.com/file/d/1icqqNH\\_iWhYqPXzp7VjDMsB0S8HyiEgi/view?usp=sharing](https://drive.google.com/file/d/1icqqNH_iWhYqPXzp7VjDMsB0S8HyiEgi/view?usp=sharing)



# Posibles Vías de Mejora

- Añadir autenticación y control de acceso a Elasticsearch y Kibana, y crear un cluster.
- Incluir detección de anomalías en tiempo real mediante Machine Learning (ej. River o modelos de ML de Elastic).
- Simulación más compleja basada en comportamiento real de sensores industriales.





# Alternativas Posibles

- Sustitución de Logstash por Filebeat o Mage AI.
- Uso de otras herramientas de visualización como Grafana con Elasticsearch como fuente.
- Implementación de servicios en la nube con Elastic Cloud o AWS Elasticsearch Service.

