



MONITORIZACIÓN DE LÍNEAS INDUSTRIALES

Abel Sebastian Exposito



Principales Objetivos

Conseguir visibilidad de las máquinas y las líneas

Controlar el consumo energético

Análisis de forma automática, rapido y eficaz

Detección sencilla de anomalías

MODELO DE ALTO NIVEL

API INYECCION DE DATOS

Datos extraídos
por nuestro
equipo de
extracción

AWS

- Usaremos AWS gestión datos
- Automatismos con Lambda
- RDS creación de BD
- Podremos gestionar todo mediante python

BASE DE DATOS

- Creamos una base de datos que se actualizara en tiempo real
- Sacaremos de aqui los datos tanto para grafana como para el modelo
- Usaremos PostgreSQL

MODELO

- Se ejecutaria en python
- Modelo de forecasting entrenado semanalmente
- Resultados listos para la comparación con los datos en grafana

GRAFANA

- Aquí tendremos nuestro panel principal
- Diferentes Dashboards con información útil para cliente
- Leeremos con consultas directamente de nuestra base de datos de AWS
- Tendríamos también los datos a futuro para comparar de nuestro modelo



AWS

Usaremos AWS por infraestructura ya creada, integración nativa, disponibilidad

Mediante python primera extracción de la API y recopilar los datos históricos para llenar la BD

Usaremos la herramienta lambda de AWS para automatismos

Recogida de datos en tiempo real, actualización BD, etc

Crearemos la BD con la herramienta RDS

Python para automatizar y controlar las herramientas mediante boto3

El coste de AWS en lambda de 1 millón de ejecuciones es alrededor de 9 euros mensuales

BD en RDS serán como máximo de 140 euros mensuales



Ejemplos de código extracción y BD

Ejemplo de extracción completa en código

https://github.com/AbelSebas/Portfolio/blob/main/RedElectricaES/02.extraction_upload.ipynb

Ejemplo de creación BD con RDS directamente con python

https://github.com/AbelSebas/Portfolio/blob/main/RedElectricaES/04.bbdd_creation.ipynb



Base de Datos

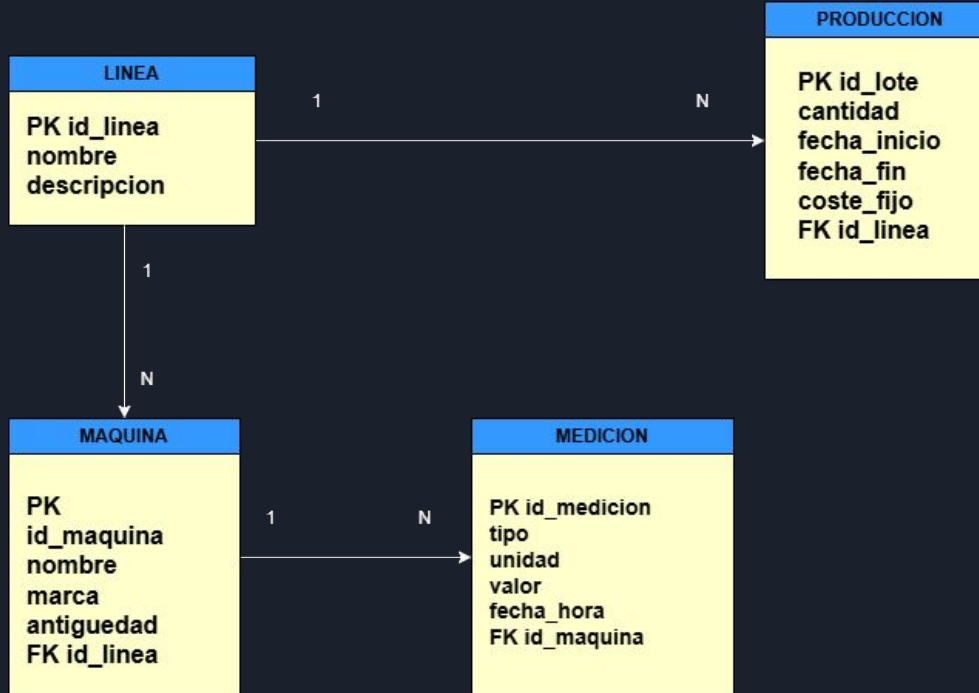
Crearemos una base de datos inicial con los datos recopilados

Mediante la herramienta lambda y RDS manejados mediante python la actualizaremos en tiempo real cada 1 o 2 minutos

Base de datos sencilla pero efectiva para trabajar con muchos datos de forma eficiente

En un principio usaremos postgresSQL

Diagrama de la Base de Datos





Modelo

datos de nuestra BD semanalmente para DataFrame

Libreria Pandas,sklearn y numpy de python para Análisis de datos,limpieza y preparación del modelo

DataFrame para entrenar modelo de forecast de consumo

Usaremos modelo XGboost,más aceptado para forecast

Con esto podremos comparar valores reales vs valor futuro

Comprobación de posibles desviaciones y su motivo



Ejemplo de código para modelo

Ejemplo de forecasting de red eléctrica española con XGboost

https://github.com/AbelSebas/Portfolio/blob/main/RedElectricaES/05.model_training_funciones.ipynb



Grafana

Desde Grafana ejecutaremos las consultas de SQL

Se conectará a la BD creada con RDS

Consultas con dashboards interesantes para el cliente con kpi que ayuden a tomar decisiones

También tendrá datos a futuro del modelo para comparar

Como extra si el cliente quiere algo con ia podríamos levantar una pequeña FASTAPI con un modelo de tratamiento de texto que traduce consultas sencillas de lenguaje coloquial a SQL para mas monitorización

https://github.com/AbelSebas/Portfolio/blob/main/RedElectricaES/Text_to_SQL_HF.ipynb



CONSULTAS SQL

Mediciones medias de cada lote

monitorizamos datos, valores anómalos en producción, comparamos consumo usando la media, posibilidad a futuro de filtrar por fecha

SELECT

p.id_lote, AVG(med.valor) AS valor_medio, med.tipo, med.unidad

FROM

Produccion p

JOIN Linea l ON p.id_linea = l.id_linea

JOIN Maquina m ON l.id_linea = m.id_maquina

JOIN Medicion med ON m.id_maquina = med.id_maquina

Where med.fecha_hora < p.fecha_fin AND med.fecha_hora > p.fecha_inicio

GROUP BY p.id_lote, med.tipo, med.unidad;



Consumo de cada línea

consulta simple para saber consumo total de la línea, si es necesario se podría modificar para tiempos concretos

SELECT

l.id_linea, SUM(med.valor) AS valor_total, med.unidad

FROM

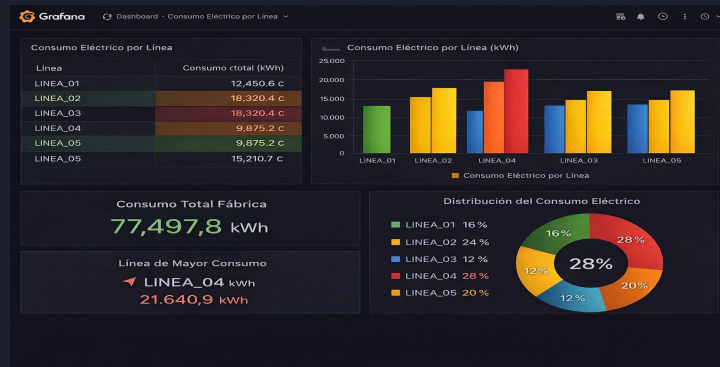
Linea l

JOIN Maquina m ON l.id_linea = m.id_linea

JOIN Medicion med ON m.id_maquina = med.id_maquina

Where med.tipo="Electricidad"

GROUP BY l.id_linea, med.unidad;



Datos de Cada Máquina

Valor máximo medio y mínimo de cada máquina de cada línea así ,util para comparar como trabajan y monitorizar

SELECT

l.id_linea,m.id_maquina,MAX(med.valor) AS valor_maximo,

MIN(med.valor) AS valor_minimo,AVG(med.valor) AS valor_medio,

med.tipo,med.unidad

FROM

Linea l

JOIN Maquina m ON l.id_linea = m.id_linea

JOIN Medicion med ON m.id_maquina = med.id_maquina

GROUP BY l.id_linea,m.id_maquina,med.tipo,med.unidad;



Valores anómalos de cada máquina

registro todo los fallos de cada maquina históricamente,util para que máquinas fallan más,mostrar datos completos para posibles cambios o reparaciones

SELECT

m.id_maquina,m.antiguedad,m.nombre,m.marca,med.valor,med.tipo,med.unidad,COUNT(*) AS fallos

FROM

maquina m

JOIN Medicion med ON m.id_maquina = med.id_maquina

WHERE

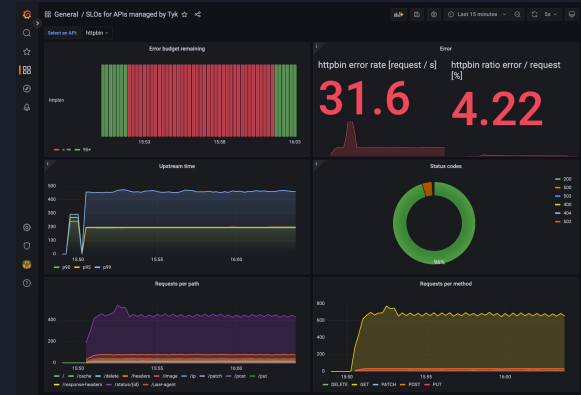
(med.tipo="Electricidad" AND med.valor=anomalo_electrico)

OR (med.tipo="Temperatura" AND med.valor=anomalo_temperatura)

GROUP BY

m.id_maquina,m.antiguedad,m.nombre,m.marca,med.valor,med.tipo,med.unidad

Order By fallos DESC



Último valor de cada medición

Esencial para monitorizar cada máquina de cada línea en tiempo real, control completo de la fábrica

SELECT

l.id_linea, m.id_maquina, med.tipo, med.unidad, med.valor

FROM

linea l

JOIN Maquina m ON l.id_linea = m.id_linea

JOIN Medicion med ON m.id_maquina = med.id_maquina

WHERE med.fecha_hora=(

SELECT MAX(Medicion.fecha_hora)

FROM Medicion

WHERE Medicion.id_maquina=m.id_maquina

)



Coste total de cada lote y consumo

suma del coste fijo y gasto eléctrico total, indica coste total ,útil para decisiones de mercado

SELECT

p.id_lote,SUM(med.valor) AS consumo _total,((SUM(med.valor)*0.191) + coste_fijo) AS coste_total,med.unidad

FROM

Produccion p

JOIN Linea l ON p.id_linea = l.id_linea

JOIN Maquina m ON l.id_linea = m.id_linea

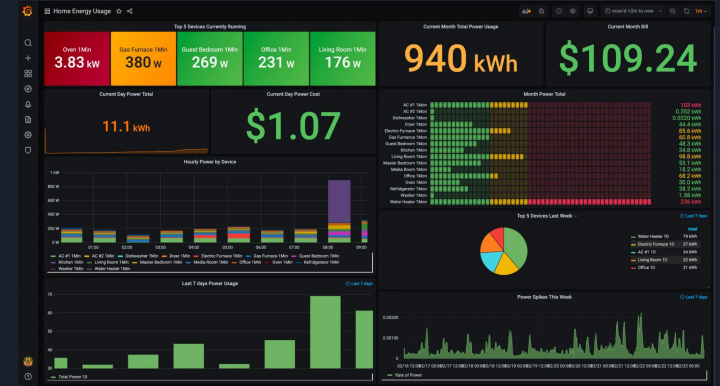
JOIN Medicion med ON m.id_maquina = med.id_maquina

Where

(med.fecha_hora<p.fecha_fin AND med.fecha_hora > p.fecha_inicio)

AND (med.tipo="Electricidad")

GROUP BY p.id_lote,med.unidad;



Ejemplo de cuadro de mando grafana

Prototipo de cuadro completo con todas las consultas anteriores






Conclusiones

Dato de consultas SQL orientados negocio centrados en consumo y anomalías

Arquitectura escalable de forma sencilla

Herramientas de desarrollo conocidas y accesibles



Hasta aqui mi presentacion,gracias al equipo de desarrollo por su atencion

Ahora es turno de Dudas y Preguntas

