
T^FM Control Autogeneración

Sistema para el procesamiento y análisis de información de sistemas de autogeneración energética

**Trabajo fin de master - 8º Master de
arquitectura Big Data**

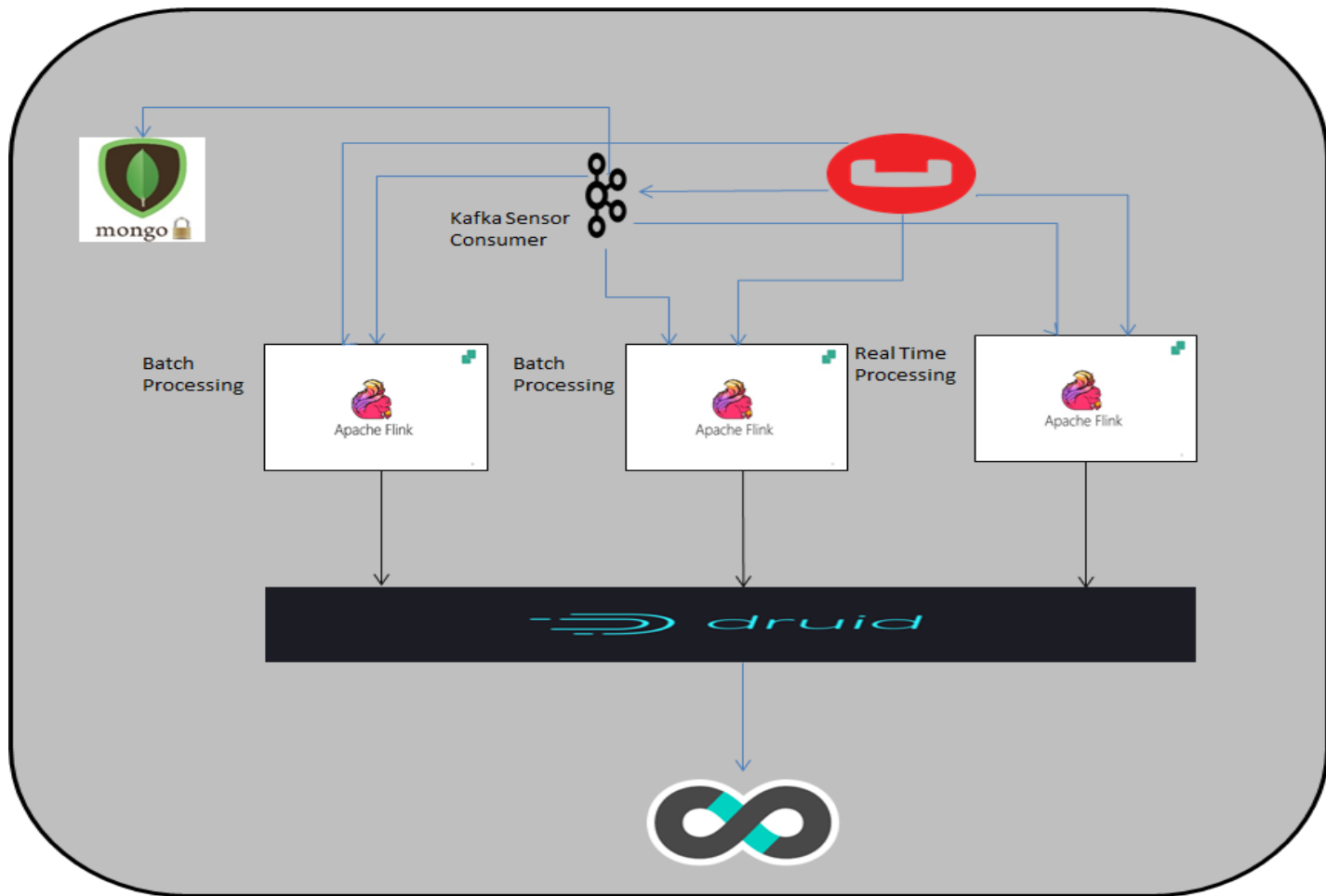
Requisitos

Se quiere disponer un sistema con el que poder:

- ☐ Recibir información de diferentes sistemas para autogeneración energética. El análisis de la información posee un componente en tiempo real y otro en modo batch. Los elementos tenidos en cuenta en el presente TFM son placas solares, sensores de luz y anemómetros.
- ☐ Ejecutar consultas de la potencia en W generada por placas solares. Esta información se generará por tramos de 1 hora. La información obtenida en tiempo real se agrupará de esta forma para obtener puntos de control que tengan un sentido práctico y sirva de comparación entre tramos y fechas.
- ☐ Ejecutar consultas sobre la información aportada por los sensores de luz. Estos sensores no proporcionan, a priori, información directa sobre la cantidad de energía que se podría producir, necesitan de un cálculo previo que da una aproximación a dicho fin. Estos sensores pueden dar información sobre el ángulo óptimo y dirección de las posibles placas a colocar en su lugar para el mejor aprovechamiento de las mismas.
- ☐ Ejecutar consultas sobre la información aportada por los sensores de luz. Estos sensores no proporcionan, a priori, información directa sobre la cantidad de energía que se podría producir, necesitan de un cálculo previo que da una aproximación a dicho fin. Estos sensores pueden dar información sobre el ángulo óptimo y dirección de las posibles placas a colocar en su lugar para el mejor aprovechamiento de las mismas.
- ☐ Consultar la información sobre el rendimiento del sistema en la entrada de la información de los diferentes topic's.
- ☐ Toda la información que se consulte referente a los tres tipos de fuentes irá enriquecida con datos de ubicación, modelo de fuente, propietario del equipo, fecha de fabricación del equipo, fecha de instalación del mismo, etc.

Arquitectura

- ❑ Para el almacenamiento y procesamiento de los datos enviados por las diferentes fuentes en tiempo real: MongoDB.
- ❑ Para el almacenamiento y procesamiento del histórico: Druid.
- ❑ Para almacenamiento información elementos e instalaciones: Couchbase.
- ❑ Motor de procesamiento en near real time: Apache Flink
- ❑ Motor de procesamiento batch: Apache Flink
- ❑ Para la visualización: Apache Superset.



Kafka - procesos

❑ StreamingSensors.py: tiene como objetivo procesar los mensajes enviados, en formato JSON por los elementos asociados al topic. La información enviada a través del topic se almacena en sin tratar MongoDB.

Además, se identificará en tiempo real el tipo de elemento asociado al mensaje del topic (placa solar, sensor de luz o anemómetro). En el caso que el elemento sea una placa o un sensor de luz, informará una tabla de agregación según una ventana temporal definida previamente (1 hora) y según el elemento (placa solar o sensor de luz).

En el caso de un anemómetro se informará una tabla de agregación cuya ventana temporal es diferente además de tener en cuenta la velocidad del viento recogida en cada instante.

Este proceso es en real time. Tanto la identificación de la tipología del elemento como la información que se almacena para las posteriores agregaciones se realiza en Couchbase.

Proceso near Real Time Flink

❑ SensorWind.py: tiene como objeto tratar la información que se graba en la tabla de agregación de anemómetro del proceso anterior. El proceso analiza en tiempo real la velocidad del viento y durante cuanto tiempo se mantiene la velocidad del mismo. En función de ese análisis, determina si procede o no generar una alerta. Si genera una alerta, enriqueceremos desde Couchbase con los datos de cliente, ubicación, timestamp y un texto asociado. En caso que se genere una alerta, la información se persistirá en Druid.

Procesos Batch Flink

❑ **SensorLight.py:** El proceso tiene como objeto tratar la información guardada en la tabla de agregación de sensor de luz una vez transcurrido el tiempo de la ventana (1 hora). La información se agregará por el identificador de cada sensor (`sensor_id`) en esa hora. Una vez agregada, cambiaremos el estado a esos registros y procederemos a calcular los diferentes datos de generación de energía (hora anterior, día anterior, mes anterior, trimestre anterior...). Una vez calculados, recuperaremos la información asociada al cliente de ese sensor y persistiremos la información en Druid en una tabla diferenciada.

❑ **Panel.py:** El proceso tiene como objeto tratar la información guardada en la tabla de agregación de panel solar una vez transcurrido el tiempo de la ventana (1 hora). La información se agregará por el identificador de cada panel (`panel_id`) en esa hora. Una vez agregada, cambiaremos el estado a esos registros y procederemos a calcular los diferentes datos de generación de energía (hora anterior, día anterior, mes anterior, trimestre anterior...). Una vez calculados, recuperaremos la información asociada al cliente de ese panel y persistiremos la información en Druid en una tabla diferenciada.

Procesos Alta Couchbase

- ❑ AddNewInstall.py: tiene como objeto añadir información a las tablas de Couchbase sobre nuevos elementos instalados con la información necesaria para los procesos en near real time o batch.
- ❑ AddNewElement.py: tiene como objeto añadir información a las tablas de Couchbase sobre nuevos tipos de sensores, placas o anemómetros.
- ❑ AddNewRela.py: tiene como objeto añadir información a las tabla de Couchbase sobre la tipología de cada uno de los elementos (sensores, placas o anemómetros).

Kafka - procesos

Proceso	Periodicidad
StreamingSensors.py	Real Time

Procesos Real Time Flink

Proceso	Periodicidad
SensorWind.py	Real Time

Procesos Batch Flink

Proceso	Periodicidad
SensorLight.py	Batch
Panel.py	Batch

Procesos Alta Couchbase

Proceso	Periodicidad
AddNewInstall.py	Batch
AddNewElement.py	Batch
AddNewRela.py	Batch

Modelo de datos - Batch (DRUID) 1/3

panel_report	
campo	descripción
client_id	código cliente
panel_id	identificación placa
address	dirección cliente
postal code	código postal
name	nombre del cliente
apell_1	apellido del cliente
apell_2	apellido del cliente
power_gen	energía generada última hora
timestamp	timestamp de generación dato
diff_last_hour	delta de generación de la última hora a la que está tratando en ese momento
diff_last_day	delta de generación de la misma hora del día anterior
diff_last_month	delta de generación de la misma hora del mes anterior
diff_last_quar	delta de generación de la misma hora un trimestre antes
diff_last_six	delta de generación de la misma hora un semestre antes
diff_last_year	delta de generación de la misma hora de un año antes
acum_day	acumulado del día
acum_month	acumulado del mes
acum_quarter	acumulado del trimestre
acum_semestre	acumulado del semestre
acum_year	acumulado del año

Modelo de datos - Batch (DRUID) 2/3

sensor_report	
campo	descripción
client_id	código cliente
sensor_id	identificación placa
address	dirección cliente
postal code	código postal
name	nombre del cliente
apell_1	apellido del cliente
apell_2	apellido del cliente
power_gen	energía generada última hora
timestamp	timestamp de generación dato
diff_last_hour	delta de generación de la última hora a la que está tratando en ese momento
diff_last_day	delta de generación de la misma hora del día anterior
diff_last_month	delta de generación de la misma hora del mes anterior
diff_last_quar	delta de generación de la misma hora un trimestre antes
diff_last_six	delta de generación de la misma hora un semestre antes
diff_last_year	delta de generación de la misma hora de un año antes
acum_day	acumulado del día
acum_month	acumulado del mes
acum_quarter	acumulado del trimestre
acum_semestre	acumulado del semestre
acum_year	acumulado del año

Modelo de datos - Batch (DRUID) 3/3

panel_alarm		sensor_alarm	
campo	descripción	campo	descripción
client_id	código cliente	client_id	código cliente
panel_id	identificación placa	light_id	identificación placa
address	dirección cliente	address	dirección cliente
postal code	código postal	postal code	código postal
name	nombre del cliente	name	nombre del cliente
apell_1	apellido del cliente	apell_1	apellido del cliente
apell_2	apellido del cliente	apell_2	apellido del cliente
power_low	diferencia de generación	power_low	diferencia de generación
text_alarm	texto asociado a la alarma	text_alarm	texto asociado a la alarma
timestamp		timestamp	

Modelo de datos - Real Time (DRUID) 1/1

wind_alarm	
campo	descripción
client_id	código cliente
wind_id	identificación placa
address	dirección cliente
postal code	código postal
name	nombre del cliente
apell_1	apellido del cliente
apell_2	apellido del cliente
wind_speed	velocidad del viento
text_alarm	texto asociado a la alarma
timestamp	

Modelo de datos - Real Time (MongoDB) 1/1

topic		topic_alarm	
campo	descripción	campo	descripción
topic_id	nombre del topic de kafka	topic_id	nombre del topic de kafka
timestamp	timestamp	timestamp	timestamp
msg_value	datos del mensaje del topic	msg_value	datos del mensaje del topic
		text_alarm	texto asociado a la alarma

Modelo de datos - Real Time (Couchbase) 1/2

agregation_panel	
campo	descripción
system_id	id del panel
power_cap	potencia generada
window_time	hora de generación
state	indicador de si se ha tratado o no

agregate_sensor_light	
campo	descripción
system_id	id del sensor
power_cap	potencia generada
window_time	hora de generación
state	indicador de si se ha tratado o no

Modelo de datos - Real Time (Couchbase) 2/2

agregate_sensor_wind	
campo	descripción
system_id	id del sensor
wind_speed	velocidad recogida
event_timestamp	timestamp del evento

system_rel	
campo	descripción
system_id	id del elemento instalado
system_type	tipo de elemento (panel, sensor, anemómetro)
system_model	modelo del elemento instalado
init_date	fecha de inicio de la relación
end_date	fecha fin de la relación

Modelo de datos - Enriquecimiento (Couchbase) 1/2

client		panel		sensor_light		sensor_wind	
campo	descripción	campo	descripción	campo	descripción	campo	descripción
client_id	código cliente	model	modelo panel	model	modelo panel	model	modelo panel
address	dirección cliente	manufacture_date	fecha fabricación	manufacture_date	fecha fabricación	manufacture_date	fecha fabricación
postal code	código postal	install_date	fecha instalación	install_date	fecha instalación	install_date	fecha instalación
client_signature	identificación cliente	power_cap	potencia del panel	power_cap	potencia del panel	variable_data	información adicional
open_date	fecha alta	panel_price	precio de compra	variable_data	información adicional	sensor_price	precio de compra
end_date	fecha baja	panel_sold	precio de venta	sensor_price	precio de compra	live_cicle	ciclo de vida útil
client_type	tipo de cliente	live_cicle	ciclo de vida útil	live_cicle	ciclo de vida útil	end_date	fecha de baja
name	nombre del cliente	variable_data	información adicional	end_date	fecha de baja		
apell_1	apellido del cliente	end_date	fecha de baja				
apell_2	apellido del cliente						

Modelo de datos - Enriquecimiento

(Couchbase) 2/2

clients_rel	
campo	descripción
client_id	id cliente
panel_id	id panel
light_id	id sensor
wind_id	id anemómetro
variable_data	información adicional
init_panel_date	fecha de alta relación cliente-panel
end_panel_date	fecha de baja relación cliente-panel
init_sensor_date	fecha de alta relación cliente-sensor
end_sensor_date	fecha de baja relación cliente-sensor
init_wind_date	fecha de alta relación cliente-wind
end_wind_date	fecha de baja relación cliente-wind

Superset

Historico empresas ☆

Ingresos totales

456M

Filtro compañías

Since

or

Until

or

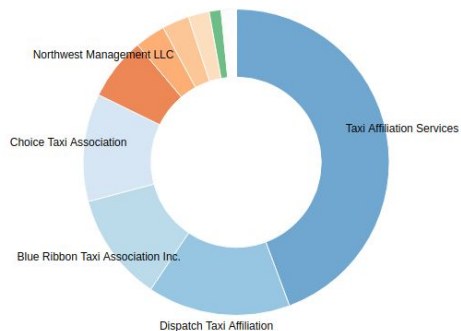
Time Grain

month

company

Select [company]

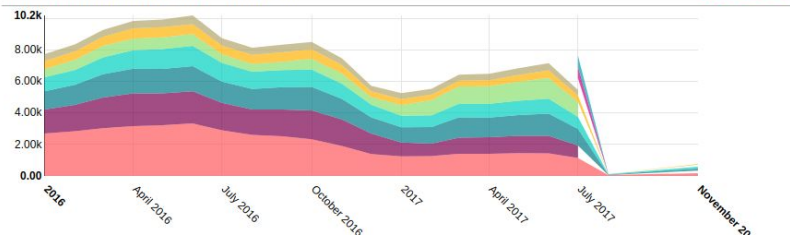
Distribución de los ingresos por empresas



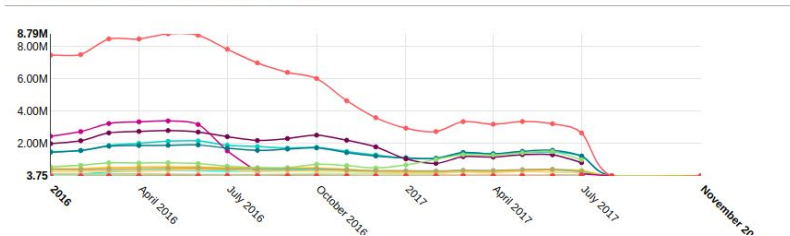
Empresas por ingresos

company	sum_trip_totals
Taxi Affiliation Services	106M
Dispatch Taxi Affiliation	36.1M
Blue Ribbon Taxi Association Inc.	30.3M
Choice Taxi Association	29.2M
Chicago Elite Cab Corp. (Chicago Carriag	20.1M
Northwest Management LLC	16.0M
KOAM Taxi Association	7.66M
Top Cab Affiliation	7.20M
Chicago Medallion Leasing INC	5.44M
Suburban Dispatch LLC	5.01M
Chicago Medallion Management	3.00M
T.A.S. - Payment Only	1.15M
Blue Cab Co	210k
Dispatch Taxi Affiliation (credit hold)	163k
0118 - 42111 Godfrey S.Awr	141k
5129 - 87128	134k
6743 - 78771 Luhak Corp	134k
Star North Management LLC	119k

Media mensual de ingresos por empresa

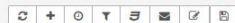


Evolución ingresos mensuales por empresa

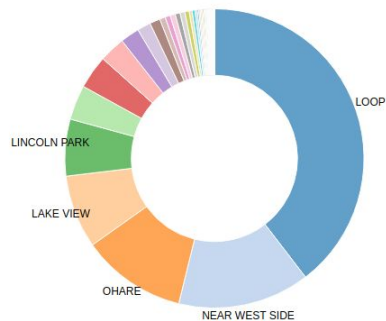


Superset

Histórico áreas ☆



Distribución de los viajes por área de inicio



Filtro - áreas

Since Until

Time Grain

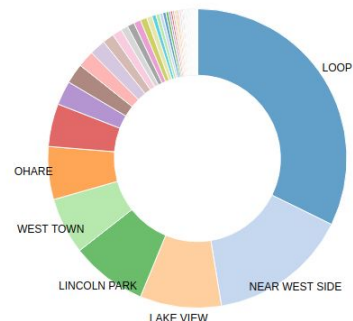
dropoff_community_area_name

Select [dropoff_community_area_name]

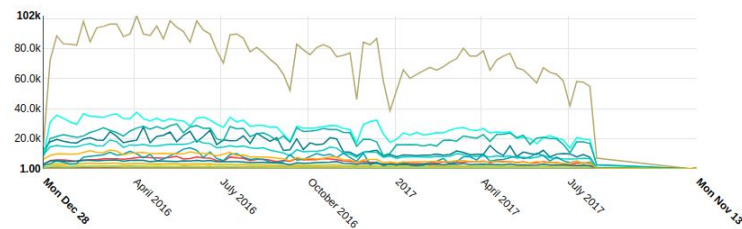
Número total de viajes

27.6M

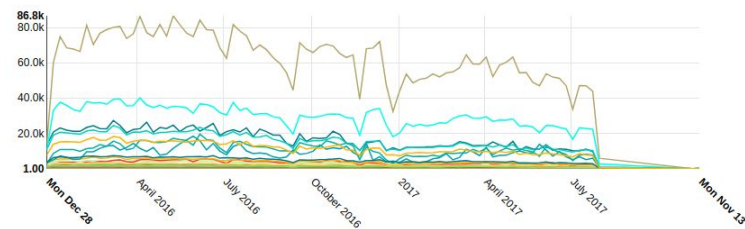
Distribución de los viajes por área de fin



Evolución de número de viajes por área de inicio



Evolución número de viajes por área de fin



Real time ☆

Filtro - real time

Since

2 days ago

Until

Time Granularity

1 minute

company

Select [company]

pickup community area name

Select [pickup_community_area name]

dropoff community area name

Select [dropoff_community_area_name]

Evolución del número de viajes

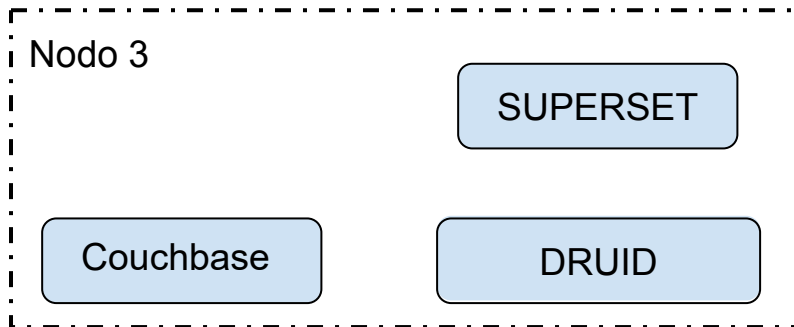
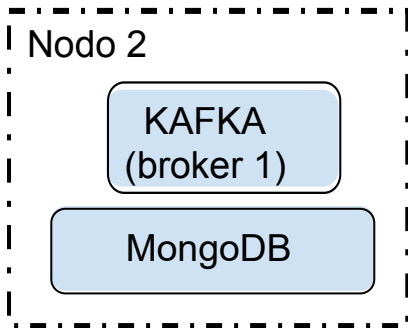
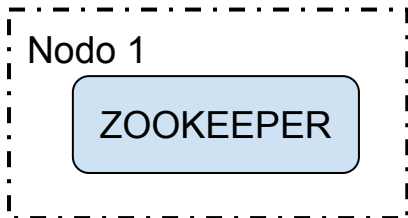


Número total de viajes

1.22k

Áreas por número de viajes

A word cloud of Chicago neighborhood names. The words are arranged in a horizontal, somewhat circular pattern. The largest words are 'NEAR WEST SIDE' in dark purple and 'LOOP' in large red letters. Other prominent words include 'OHARE' in teal, 'LAKE VIEW' in teal, 'WEST TOWN' in yellow, and 'ROGERS PARK' in teal. Smaller words include 'HUMBOLDT PARK', 'GRAND BOULEVARD', 'BELMONT CRAGIN', 'JEFFERSON PARK', 'GREATER GRAND CROSSING', 'ARMOUR SQUARE', 'LOGAN SQUARE', 'PORTAGE PARK', 'MORGAN PARK', 'UPTOWN', 'LOWER WEST SIDE', 'DOUGLAS', 'LINCOLN SQUARE', 'NORWOOD PARK', 'IRVING PARK', 'NEAR SOUTH SIDE', 'ENGLEWOOD', 'SOUTH LAWNDALE', 'NORTH CENTER', 'WEST RIDGE', 'AVONDALE', 'BRIDGEPORT', 'WOODLAWN', 'ALBANY PARK', 'LINCOLN PARK', 'EDgewater', 'HYDE PARK', and 'GARFIELD RIDGE'. The colors of the words vary, including shades of purple, teal, yellow, red, and green.



Problemas / Soluciones

- ❑ Para paliar el tráfico de red entre Kafka y MongoDB ubicaremos en el mismo nodo ambos elementos.
- ❑ Para paliar el tráfico de red entre Druid y Superset ubicaremos en el mismo nodo ambos elementos.

DEMO