



FIWARE

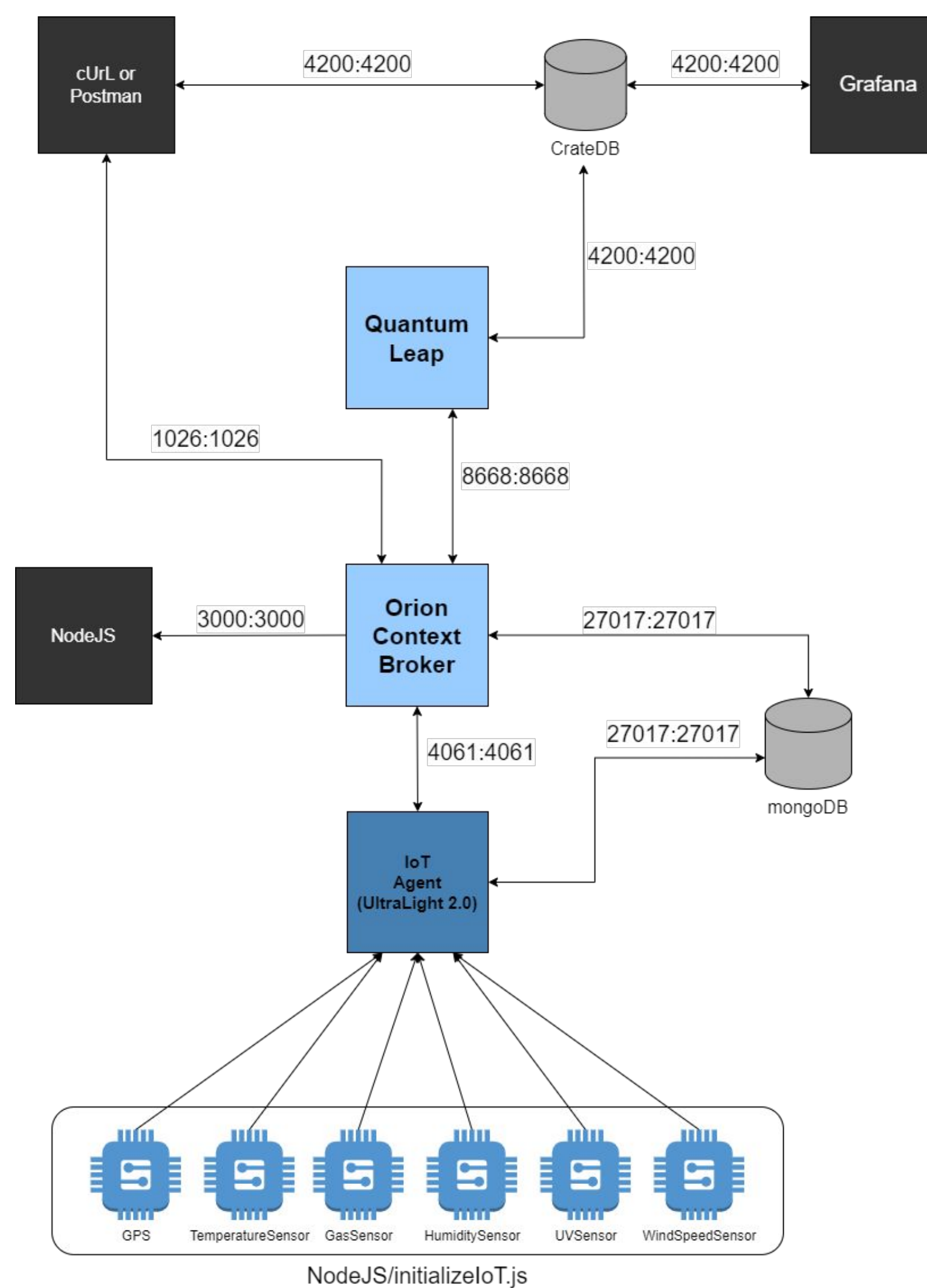
CONTROL DE INCENDIOS MEDIANTE DRONES

Ejemplo de
uso: FIWARE

Índice de componentes

Elementos que componen el sistema del caso de uso.

- 01** Sensores simulados
- 02** IoT Agent Ultralight 2.0
- 03** Orion Context Broker
- 04** MongoDB (context)
- 05** QuantumLeap
- 06** CrateDB (Time-Series Data)
- 07** Grafana



Sensores simulados

GPS

Simulación vuelo del dron

Este sensor nos permite simular lo que sería el vuelo real del dron, para lo que se calcula, a partir de unas coordenadas base, sus coordenadas intermedias, y esto se pasa como tráfico northbound simulando una lectura real de posición.

UV

Simulación

Permite la identificación rápida de la ubicación exacta del incendio, ya que las llamas emiten radiación UV que puede ser detectada incluso a través de la humareda.

TEMPERATURE

Simulación temperatura

En el contexto de este caso de uso, un sistema para control de incendios mediante drones, este valor nos permite generar alertas en base a la peligrosidad del posible incendio.

HUMIDITY

Simulación

La baja humedad en el aire y en la vegetación aumenta el riesgo de incendios forestales. Los datos de humedad ayudan a evaluar la sequedad del entorno y la probabilidad de propagación del fuego.

GAS

Simulación

La detección temprana de gases como el monóxido de carbono, el dióxido de carbono u otros productos químicos inflamables nos permitiría actuar rápidamente.

WIND SPEED

Simulación velocidad del aire

El viento puede afectar significativamente la propagación del fuego. Conocer la velocidad y dirección del viento ayuda a prever cómo se propagará el incendio y a planificar estrategias para combatirlo.

IoT Agent for the Ultralight 2.0 protocol

Agente IoT desplegado en docker que se encarga de comunicar los dispositivos IoT con el Orion Context Broker.



QUÉ ES

Ultralight 2.0 es un protocolo ligero basado en texto dirigido a dispositivos con restricciones y comunicaciones donde el ancho de banda y la memoria del dispositivo pueden ser recursos limitados.



TRANSPORTE

Este agente proporciona diferentes protocolos de transporte, como HTTP, MQTT y AMQP. En nuestro caso de uso el protocolo de transporte utilizado es HTTP.

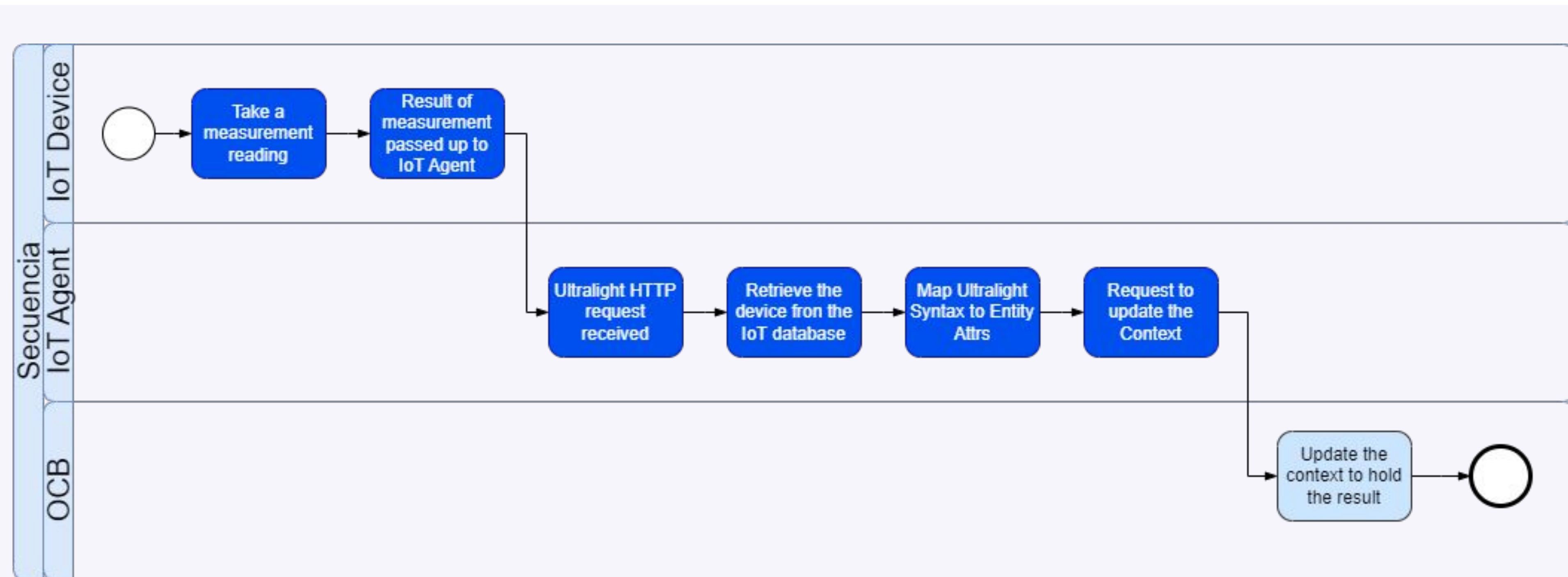


MENSAJES

En primer lugar este IoT agent debe tener registrado el dispositivo del que va a recibir la lectura. Una vez generado este entity (que tendrá una equivalencia en el OCB), el intercambio de información se realiza con comandos:

t | 10 | s | true | b | 78.8

Northbound Traffic (Measurements)



Orion Context Broker

Implementación Publish/Subscribe Context Broker GE con interfaz NGSI.



CONSULTAR CONTEXT DATA

El OCB almacena la información de contexto actualizada, no persistente. Esta información de contexto consiste en entidades (por ejemplo, un dron), y sus atributos (por ejemplo la velocidad máxima o los sensores ique tiene instalados).



ACTUALIZAR CONTEXT DATA

Por ejemplo, enviar actualizaciones de la temperatura. Esto enlaza con el IoT Agent descrito anteriormente, pero también se puede realizar de forma manual.

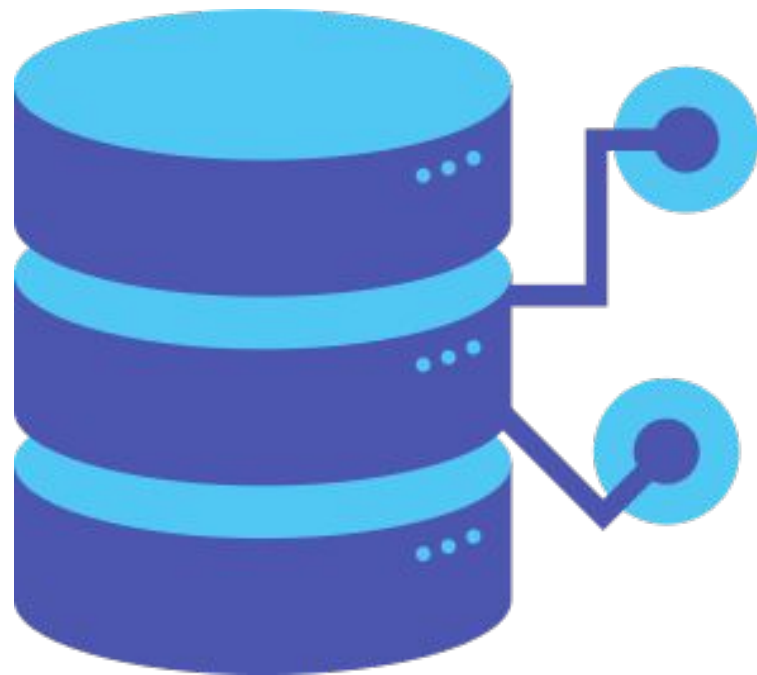


SUSCRIBIRSE A CAMBIOS EN EL CONTEXT DATA

Por ejemplo, enviar actualizaciones de temperatura a diferentes endpoints desde los que se esté escuchando, donde estos endpoints pueden ser otros generic enablers habilitados o endpoints de una aplicación propia.

MongoDB

Context DataBase



OCB

Esta base de datos es utilizada por el Orion Context broker para guardar información de las entidades, suscripciones y registros, así como de los atributos de las entidades y sus valores actualizados, si procede.

IOTA

Esta base de datos es utilizada también por los IoT Agents para guardar información de los dispositivos, como pueden ser URLs y Keys.

QuantumLeap

QuantumLeap es un generic enabler de Fiware utilizado para la persistencia y consulta de bases de datos de series temporales (crateDB). Esto nos sirve para persistir información procedente de las lecturas de los sensores y, posteriormente, realizar las acciones que queramos con ellas, como mostrarla en dashboards.



CrateDB persisting time-series data

CrateDB es un DBMS SQL distribuido diseñado para su uso con IoT. Es capaz de ingerir un gran número de puntos de datos por segundo y puede consultarse en tiempo real. La base de datos está diseñada para la ejecución de consultas complejas, como datos geoespaciales y series temporales. La recuperación de estos datos de contexto histórico permite crear gráficos y cuadros de mando que muestran las tendencias a lo largo del tiempo.



Grafana

Este elemento no es un generic enabler de Fiware, pero es la herramienta que se utilizará para interaccionar con los datos alojados en la CrateDB. Esta herramienta opensource nos permite generar dashboards, así como alertas y análisis de información para detectar patrones y/o tendencias de históricos.



cURL or Postman

Se utiliza esta herramienta para múltiples acciones en cuanto a los datos de conexto.

CHECKS DE FUNCIONAMIENTO

Se utiliza esta herramienta para realizar llamadas a los distintos generic enablers desplegados en docker que nos permitan saber si están operando correctamente.

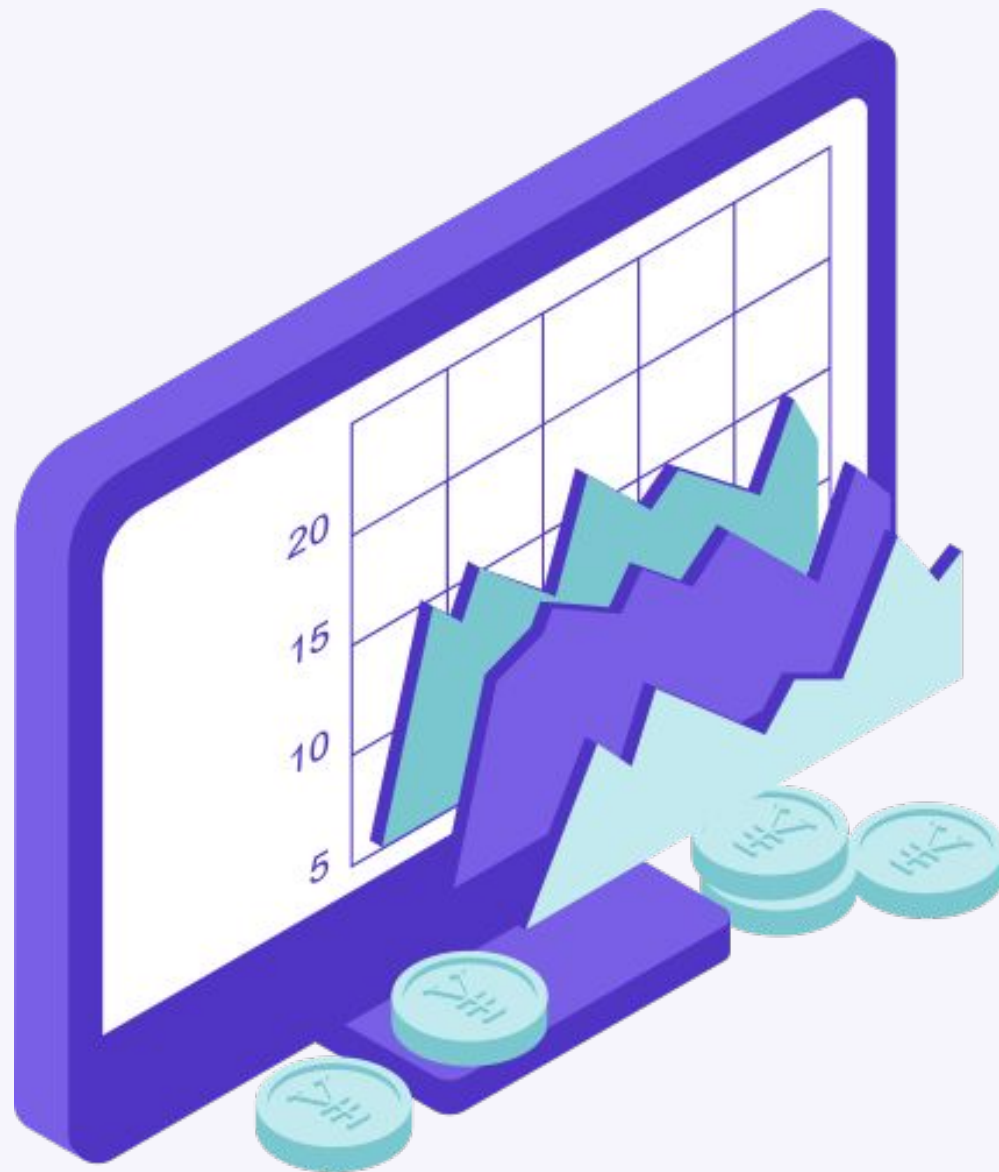
MODIFICACIÓN DE ESTRUCTURAS DE DATOS

De esta forma nos ahorramos tener que generar un frontend (siempre que no sea necesario) para realizar las inserciones y modificaciones tanto de las entidades de contexto como de sus atributos, agilizando enormemente estas acciones.

CONSULTAS Y COMPROBACIONES

De nuevo, una de sus ventajas más notorias es la posibilidad de prescindir de código en cuanto a la gestión de datos manejados por los generic enablers, ya que estos contienen una API preparada para poder realizar una gestión completa con comandos cURL.

Posibles mejoras futuras



- 01 MQTT (mosquito)
- 02 Hacer más robusto el sistema completo
- 03 Introducir diferentes IoT Agents
- 04 uso en escenario real
- 05 Generar suscripciones a QL más elaboradas en cuanto a Time-Series se refiere.

NodeJS APP DEMO

A continuación una demostración de uso y funcionamiento del sistema desarrollado

