

PLATAFORMA DE GESTIÓN DE EDIFICIOS

DESCRIPCIÓN TÉCNICA

V01 (10-08-2022)

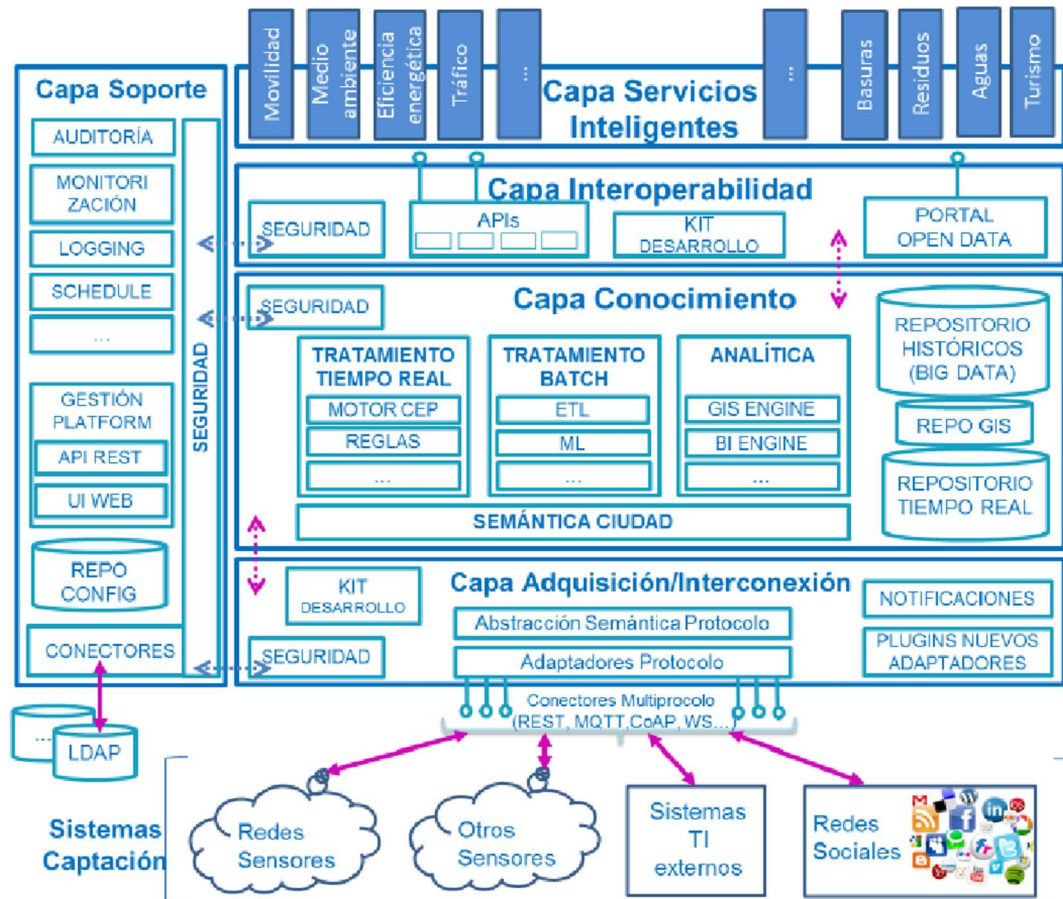
Control de versiones**Contenido**

1	Características generales de la plataforma	3
1.1	Norma de referencia	3
1.2	Arquitectura de la plataforma	4
1.3	Diagramas de bloques	5
2	Infraestructura física de la plataforma	7
2.1	Conectividad	7
2.2	Esquema de red	8
2.3	Servidores	9
3	Componentes utilizados para el desarrollo de la plataforma	10
3.1	Docker	10
3.2	Sistemas gestores de bases de datos	10
3.3	Fiware	11
3.4	Desarrollos ad hoc	17
4	Estructura de la información generada por los edificios	18
5	Características funcionales de la plataforma	25
5.1	Gestión de eventos	25
5.2	Interfaz de estado del edificio	26
5.3	Variables estáticas	27
5.4	KPIS	28
5.5	Interfaz de gráficos	31
5.6	Acceso a datos de edificios por terceras aplicaciones mediante API Rest	31
5.7	Acceso específico de sistemas Stardust	31
5.8	Interfaz de alarmas	34
5.9	Control de accesos y permisos	36

1 Características generales de la plataforma

1.1 Norma de referencia

La arquitectura de la plataforma sigue los principios recogidos en la **norma UNE 178104**. Esta norma esquematiza el concepto de plataforma de servicios inteligente así:



Los objetivos principales de una Plataforma Integral de Ciudad Inteligente son:

- > Recoger la información de la Ciudad, ciudadanos y empresas, cumpliendo los requisitos de privacidad que fueran pertinentes.
- > Distribuir la información, para que pueda ser procesada por los responsables de los diferentes servicios.
- > Analizar la información según los criterios definidos
- > Tomar decisiones devolviendo la información refinada a los sistemas encargados de ejecutar las distintas acciones.
- > Exponer datos y capacidades a desarrolladores para facilitar la creación de un ecosistema de aplicaciones sobre la plataforma, que cree un valor adicional para el ciudadano.

Estas plataformas simplifican el desarrollo de aplicaciones, reduciendo los tiempos de desarrollo y los costes de mantenimiento. Permiten la adopción de estándares de mercado y hacen los desarrollos más reusables y extensibles.

Los conceptos principales pueden trasladarse a las necesidades de la plataforma de gestión de edificios de Nasuvinsa.

1.2 Arquitectura de la plataforma

En la siguiente figura se presenta la arquitectura de la PLATAFORMA incluyendo un detalle a alto nivel de los módulos que la componen.

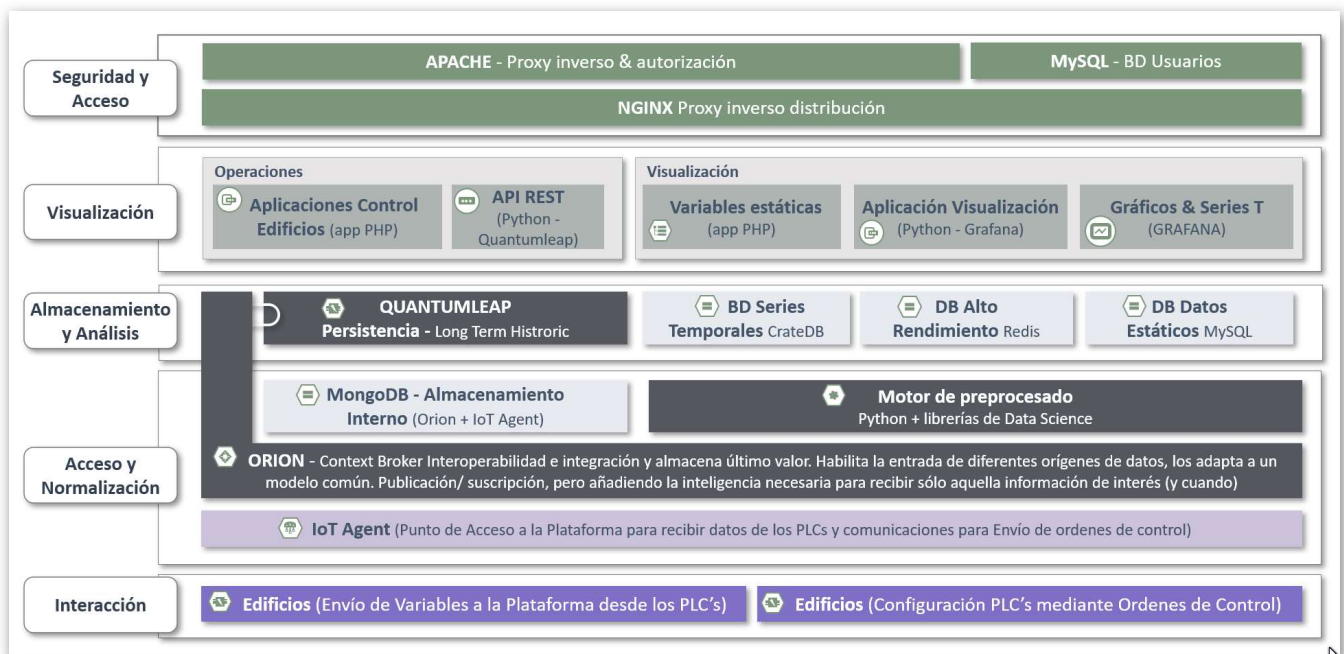
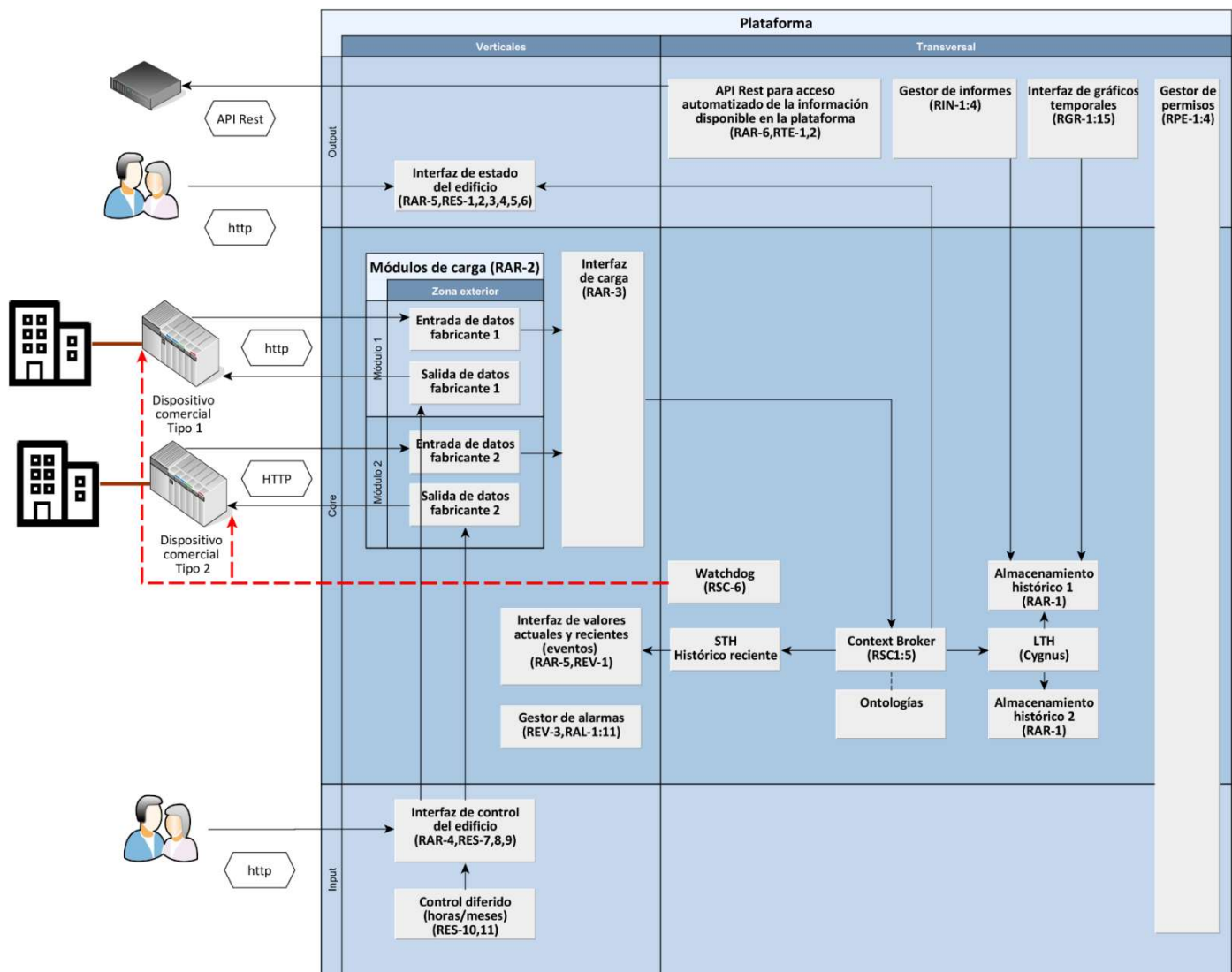


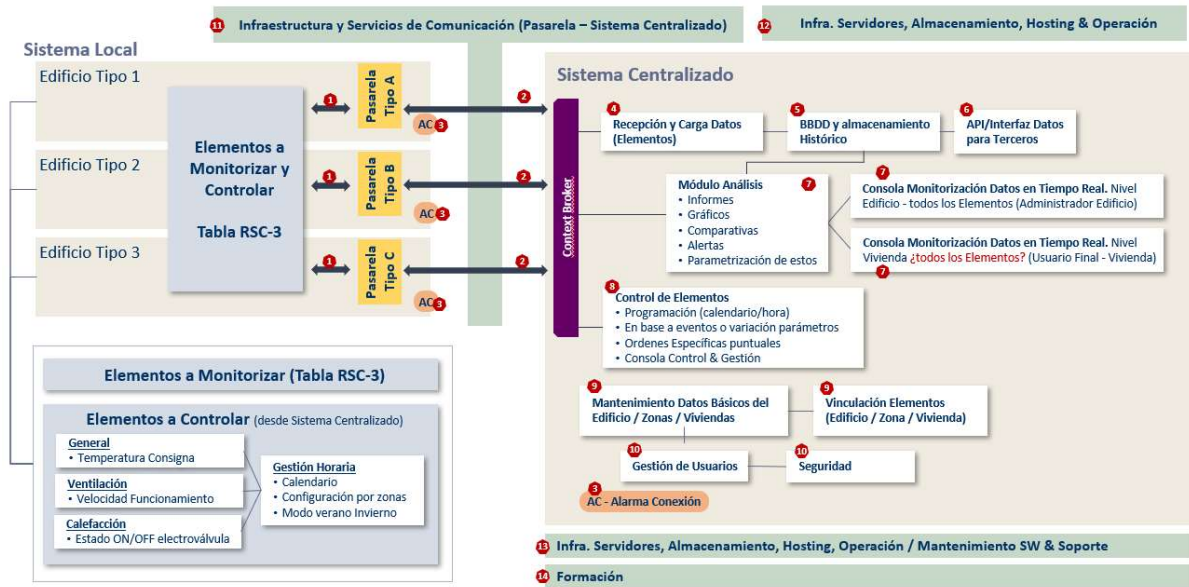
Ilustración 2. Esquema de bloques de la plataforma de servicios inteligentes

1.3 Diagramas de bloques

El objetivo de la plataforma es, por una parte, **capturar información de estado de los elementos de edificios de viviendas** y por otra parte controlar manualmente y los parámetros de funcionamiento de sus automatismos. Durante el proyecto se integrará un total de 25 edificios desde los que se recabará información en tiempo real cuantificable en aproximadamente 600 variables por edificio cada 5 minutos. El sistema dispone de un sistema de alertas programable que permite notificar a los responsables del edificio o actuar sobre parámetros de control en tiempo real. Además, se dispone de información histórica para su análisis estadístico e información de indicadores clave o KPI.

El diagrama funcional de la plataforma es el siguiente:





2 Infraestructura física de la plataforma

La arquitectura física de la plataforma se compone de los elementos principales indicados en el diagrama a continuación

2.1 Conectividad

Conectividad interna

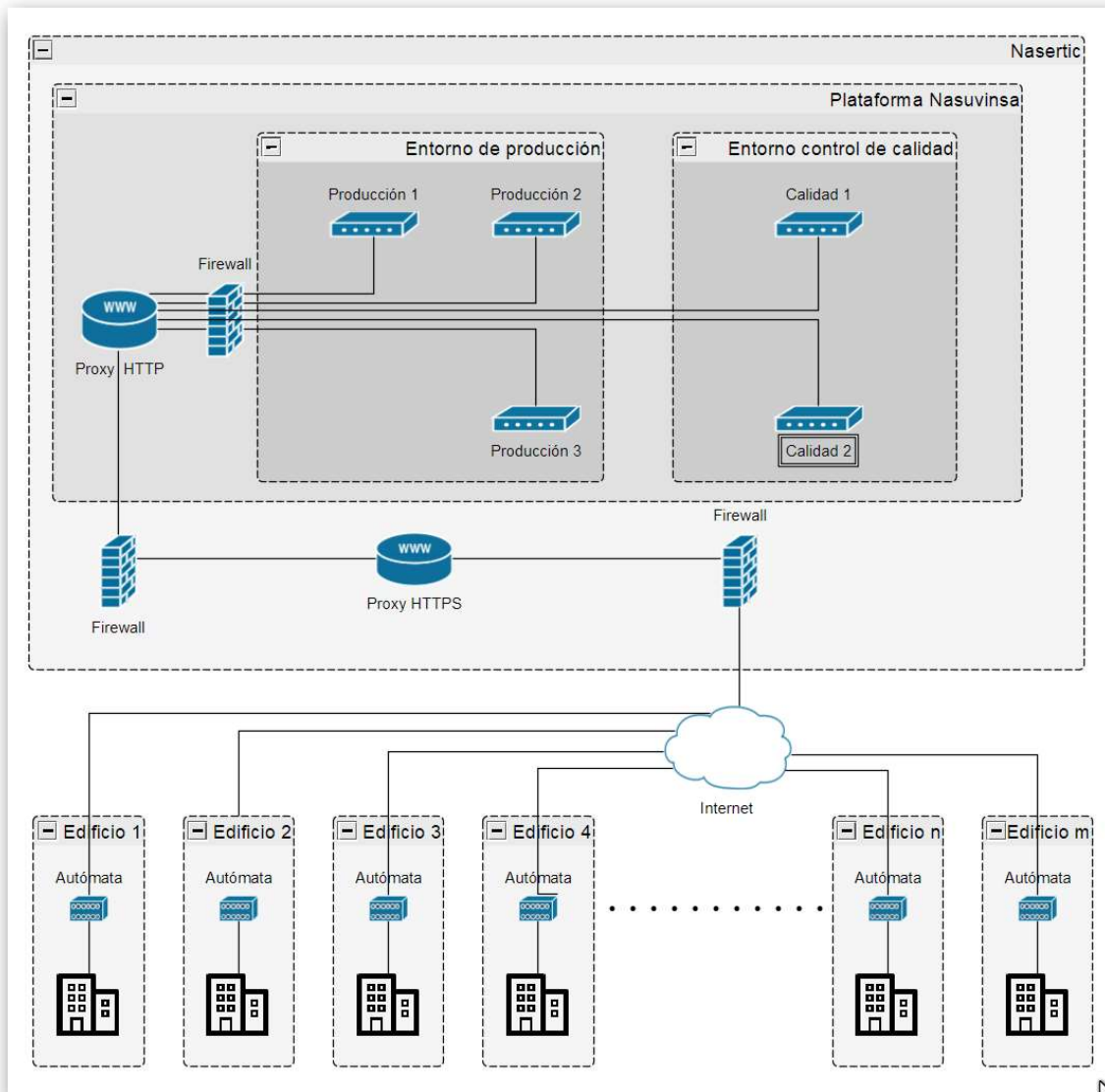
Los servidores que alojan los elementos constructivos de la plataforma se encuentran en una VLAN aislada del resto de redes en Nasertic. El objetivo de esta topología es aislar todo el sistema de otros sistemas por motivos de seguridad y rendimiento. Para gestionar la plataforma es necesario acceder a la VLAN mediante VPN a un equipo puente con SO Windows desde el que se accede a los servidores de la plataforma mediante protocolo SSH.

Conectividad externa

La conexión principal desde los edificios a la plataforma se realiza [mediante acceso HTTPS al API Rest correspondiente al edificio](#). El punto de acceso será un proxy HTTPS gestionado por Nasertic. Éste redirige la conexión a [un segundo proxy HTTP gestionado por la plataforma Nasuvinsa](#), que reenvía la conexión entrante al servicio solicitado.

Dicho servicio podrá ser un IoT agent cuando el remitente de la comunicación sea un autómata de edificio o una aplicación web cuando el usuario sea un administrador de aplicación.

2.2 Esquema de red



Edificios con su correspondiente autómata que se comunica con la plataforma a través de internet mediante protocolo HTTPS y API Rest proporcionado por Fiware IoTAgent. Adicionalmente, los edificios reciben instrucciones de control de la plataforma para configurar los valores operativos del edificio mediante un desarrollo adhoc programado con lenguaje PHP.

Infraestructura alojada en Nasertic. Como se ha indicado, la conexión desde los edificios a la plataforma se realiza mediante acceso HTTPS al API Rest específico para cada edificio. El punto de acceso es el proxy https gestionado por Nasertic que redirige la conexión al proxy HTTP inverso gestionado por la plataforma Nasuvinsa. Este segundo proxy, que es parte integrante de la plataforma, redirige la conexión entrante al servicio final solicitado, que podrá ser:

1. Un lot agent cuando el remitente de la comunicación sea un autómata de edificio o

2. Una aplicación web cuando el acceso sea de un usuario o un administrador de la aplicación.
3. Un servicio API Rest específico cuando el acceso sea para un consumo automatizado de datos desde otras plataformas o aplicaciones

2.3 Servidores

La plataforma se encuentra instalada en 6 servidores con sistema operativo Rocky Linux 8 con las siguientes características:

Entorno	Función	Servidores
Calidad	Docker CrateDB Bases de datos auxiliares Mongodb	1
Calidad	Docker Context broker + QuantumLeap IoTagent Grafana Entorno python + django Api rest + STH	1
Producción	Proxy inverso apache Servidor de autenticación y control de accesos	1
Producción	Docker CrateDB Bases de datos auxiliares Aplicaciones Python Ad Hoc	1
Producción	Docker Grafana Proxy inverso nginx Mongodb Context broker QuantumLeap Aplicaciones PHP Ad hoc	1
Producción	Docker Procesos auxiliares, copias de seguridad	1

Gran parte de los servicios y subsistemas se despliegan mediante tecnología Docker pero otra parte considerable se configura mediante paquetes disponibles en la distribución Rocky Linux 8

3 Componentes utilizados para el desarrollo de la plataforma

3.1 Docker

La parte principal de los componentes se despliega mediante **contenedores Docker**. Esta modalidad optimiza el uso de recursos hardware, facilita el despliegue de infraestructura lógica y permite disponer de una estructura de múltiples elementos similares en un solo servidor, al contrario que mediante instalación de dichos elementos (por ejemplo, Orion) directamente en sistema operativo.

Por otra parte, la instalación mediante contenedores Docker es la modalidad propuesta por gran parte de los fabricantes de componentes individuales.

Debido a la relativa sencillez de la topología propuesta no se recomienda el uso de orquestadores para la gestión del ciclo de vida de los contenedores, por lo que se utiliza una configuración basada en **Docker-compose**.

Componente	Enlace	Licencia
Docker	https://www.docker.com/	Los componentes de código abierto de Docker generalmente se distribuyen bajo la licencia Apache 2.0 https://www.docker.com/legal/components-licenses https://www.apache.org/licenses/LICENSE-2.0

3.2 Sistemas gestores de bases de datos

La plataforma usa varios sistemas gestores de bases de datos: Mariadb, MongoDB y Cratedb

Cratedb para almacenamiento histórico de **datos generados por los edificios**. Es el SGBD principal de la plataforma.

Mariadb para almacenamiento de **información estática**, de configuración y de datos para desarrollos adhoc

Mongodb para almacenamiento de datos necesarios para el funcionamiento de componentes, principalmente **Orion e lotAgent**.

Se puede acceder a la documentación de cada uno de los sistemas en sus correspondientes sitios web:

Componente	Enlace	Licencia
Mariadb	https://mariadb.com/	The MariaDB server is available under the terms of the GNU General Public License, version 2.

		https://mariadb.com/kb/en/library/mariadb-license/ http://www.gnu.org/licenses/old-licenses/gpl-2.0.html
Mongodb	https://www.mongodb.com/	Server Side Public License https://www.mongodb.com/licensing/server-side-public-license
Cratedb	https://crate.io	Crate is licensed under the Apache License, Version 2.0. https://crate.io/legal/license/

3.3 Fiware



FIWARE es un marco seleccionado de componentes de plataforma de código abierto que se pueden ensamblar entre sí y con otros componentes de plataforma de terceros para acelerar el desarrollo de soluciones inteligentes.

FIWARE es una plataforma abierta que integra las tecnologías clave para el desarrollo de aplicaciones IoT permitiendo la creación de un ecosistema sostenible de innovación abierto.

Mediante el uso de Fiware es posible abordar soluciones a retos como la conexión con sensores y dispositivos, el almacenamiento, acceso, procesado, publicación y análisis tanto de contenidos multimedia como de datos a gran escala (BigData), la cocreación de aplicaciones y contenidos, el desarrollo de interfaces de usuario avanzadas con capacidades 3D y de realidad aumentada.

Las especificaciones de la plataforma FIWARE son públicas y libres de royalties.

La mayor parte de los componentes utilizados son implementaciones de componentes que cumplen con las especificaciones FIWARE, lo que asegura una convergencia efectiva y la creación de un ecosistema de Innovación y Emprendimiento en torno a ella.

Se puede obtener más información del ecosistema Fiware en la página web

<https://www.fiware.org/> y de los componentes individuales que forman el catálogo de elementos Fiware para desarrolladores en <https://www.fiware.org/catalogue/>

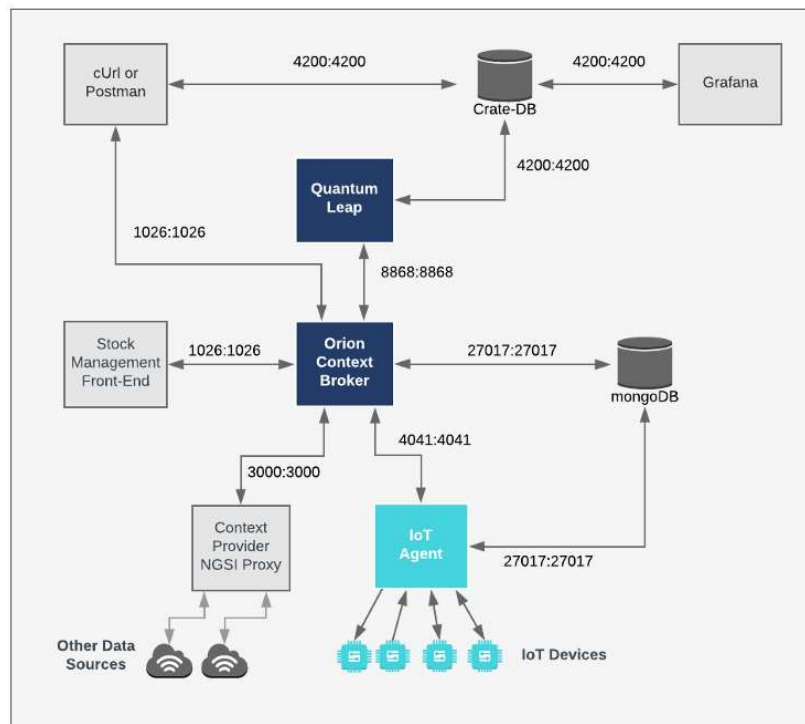
Se indica a continuación la relación de componentes principales de fiware o adicionales utilizados en el desarrollo de la plataforma:

Persistencia de datos

QuantumLeap es un servicio REST para almacenar, consultar y recuperar datos espacio-temporales NGSI v2.

QuantumLeap convierte los datos semiestructurados de NGSI en formato tabular y los almacena en una base de datos de series de tiempo, asociando cada registro de base de datos con un índice de tiempo y, si está presente en los datos de NGSI, información de geoposición.

Los clientes REST pueden recuperar entidades NGSI filtrando conjuntos de entidades a través de rangos de tiempo y operadores geoespaciales. Desde el punto de vista del cliente, estas consultas se definen en entidades NGSI en lugar de tablas de base de datos. Sin embargo, la funcionalidad de consulta disponible a través de la interfaz REST es bastante básica y las consultas más complejas generalmente requieren la utilización directa de la base de datos.



Componente	Enlace	Licencia
Quantumleap	https://github.com/smartsdk/ngsi-timeseries-api	MIT License

		https://github.com/smartsdk/ngsi-timeseries-api/blob/master/LICENSE
Cratedb	https://crate.io	Crate is licensed under the Apache License, Version 2.0. https://crate.io/legal/license/

Interfaz de estado del edificio

Toda la información de estado de los edificios se ha implementado con instancias individuales para cada edificio de Grafana

Grafana es una herramienta de visualización de datos en series temporales distribuida bajo licencia open source habitualmente usada en monitorización de infraestructura y sensores, automatización, condiciones ambientales y control de procesos. Ofrece una interfaz web para la creación de componentes gráficos basadas en series temporales utilizando distintas bases de datos con sus respectivos lenguajes y sentencias.

Componente	Enlace	Licencia
Grafana	https://grafana.com/	AGPLv3 LICENSE https://www.gnu.org/licenses/agpl-3.0.en.html
Cratedb	https://crate.io	Crate is licensed under the Apache License, Version 2.0. https://crate.io/legal/license/

API Rest para accesos automatizados

Permite el acceso a datos de terceros, por ejemplo, la plataforma Stardust del Ayuntamiento de Pamplona

Componente	Enlace	Licencia
Desarrollo mediante Python y desarrollo AdHoc.	https://www.python.org/	
QuantumLeap	https://github.com/orchestracities/ngsi-timeseries-api	MIT License

Interfaz de gráficos temporales

El acceso a información temporal se ha desarrollado mediante Grafana.

Componente	Enlace	Licencia
Grafana	https://grafana.com	grafana está licenciado bajo la Licencia Apache 2.0 https://github.com/grafana/grafana/blob/master/LICENSE
Cratedb	https://crate.io	Crate is licensed under the Apache License, Version 2.0. https://crate.io/legal/license/

Módulos de carga

Para la entrada de datos desde los edificios hasta la plataforma: Desarrollo mediante Fiware IoT Agent + Node.js utilizando el protocolo Ultralight 2.0 utilizado para la carga de datos desde los PLCs de edificios.

IoT agent emplea mongodb como backend para almacenamiento de datos y configuración

“Un agente de IoT es un componente que permite a un grupo de dispositivos enviar sus datos (y ser administrados) desde context broker mediante sus propios protocolos nativos. Los agentes de IoT también deben ser capaces de tratar los aspectos de seguridad de la plataforma FIWARE (autenticación y autorización del canal).

Orion Context Broker utiliza exclusivamente solicitudes NGSI para todas sus interacciones. Cada agente de IoT proporciona una interfaz NGSI de puerto norte que se utiliza para las interacciones con el agente de contexto y todas las interacciones debajo de este puerto se producen mediante el protocolo de comunicación definido para los dispositivos IoT.

Esto aporta una interfaz estándar a todas las interacciones de IoT en el nivel de administración de información de contexto. Cada grupo de dispositivos IoT puede usar sus propios protocolos propietarios y mecanismos de transporte dispares internamente, mientras que el agente de IoT asociado ofrece una interfaz común para manejar esta complejidad.”¹

IoT agent Ultralight 2.0 utiliza los siguientes componentes:

Componente	Enlace	Licencia
------------	--------	----------

¹ Información extraída de <https://github.com/FIWARE/tutorials.IoT-Agent>

IoTAgent UL 2.0	https://fiware-iotagent-ul.readthedocs.io/en/latest/	The IoT Agent for JSON is licensed under Affero General Public License (GPL) version 3.
Mongodb	https://www.mongodb.com/	Server Side Public License https://www.mongodb.com/licensing/server-side-public-license
Node.js	https://nodejs.org/es/	https://github.com/nodejs/node/blob/master/LICENSE
Eclipse ² Mosquitto™ An open source MQTT broker	http://mosquitto.org/	This project is dual licensed under the Eclipse Public License 1.0 and the Eclipse Distribution License 1.0 as described in the epl-v10 and edl-v10 files. https://github.com/eclipse/mosquitto/blob/master/LICENSE.txt
RabbitMQ, open source message broker	https://www.rabbitmq.com/	MOZILLA PUBLIC LICENSE Version 1.1 https://www.rabbitmq.com/mpl.html

Salida de datos

Desarrollo mediante Python para la exportación mediante API Rest

Gestor de permisos y seguridad

Fiware Keyrock, Fiware Wilma y Auth Force

Componente	Enlace	Licencia
Proxy inverso Apache	https://httpd.apache.org/	Server Side Public License https://www.mongodb.com/licensing/server-side-public-license
Mariadb	https://mariadb.com/	The MariaDB server is available under the terms of the GNU General Public License, version 2. https://mariadb.com/kb/en/library/mariadb-license/ http://www.gnu.org/licenses/old-licenses/gpl-2.0.html

² Mosquitto y RabbitMq no se utilizan directamente por la plataforma, su instalación es requisito para la instalación de IoTagent

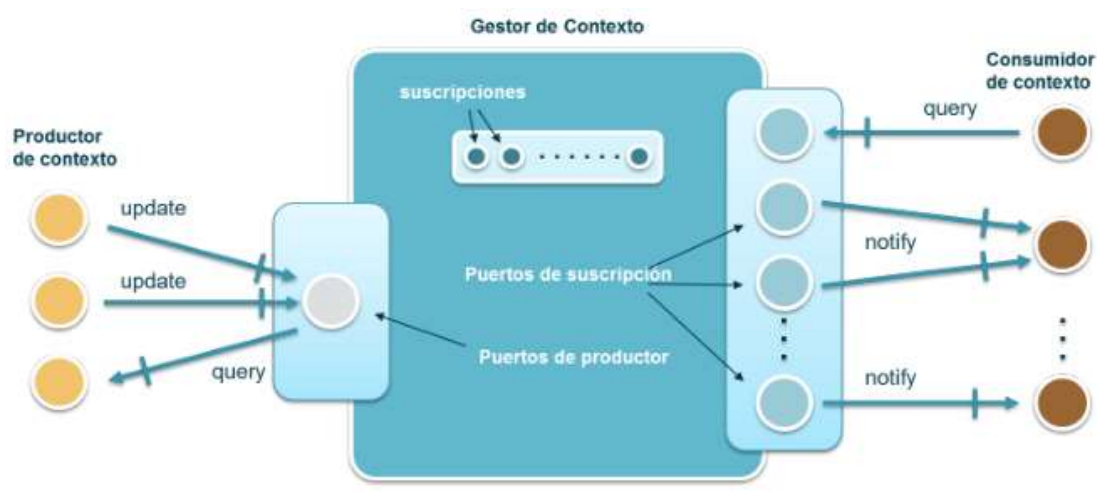
--	--	--

Interfaz de carga, Interfaz de valores actuales y recientes

Desarrollo mediante Python

Context broker

El Gestor de contextos es la pieza clave dentro del entramado de la plataforma FIWARE en la que se basa la plataforma. Este componente juega un papel estratégico en la capa de interoperabilidad e integración, habilitando la entrada de los diferentes orígenes de datos y adaptándolos a un modelo común.



Por otra parte, Orion proporciona toda la información de contexto³ de los dispositivos gestionados.

Componente	Enlace	Licencia
Orion	https://github.com/telefonicaid/fiware-orion/	GNU Affero General Public License v3.0 https://github.com/telefonicaid/fiware-orion/blob/master/LICENSE
Mongodb	https://www.mongodb.com/	Server Side Public License https://www.mongodb.com/licensing/server-side-public-license

Ontologías

³ La información de contexto es el conjunto de datos relevantes de un elemento controlado con sus valores actuales. La documentación completa de Orion se encuentra en <https://github.com/Fiware/context.Orion>

Diseño y definición ad hoc de la ontología edificio compatibilizando necesidades de gestión de edificio, interfaz de carga mediante IoT Agent UltraLight 2.0 y Orion context broker.

LTH (Histórico de largo plazo)

La información histórica de series temporales se almacena mediante el sistema gestor de bases de datos Cratedb. La conexión Orion-Cratedb se implementa mediante el componente Fiware Quantumleap.

El acceso a información temporal se desarrollará mediante Grafana.

Componente	Enlace	Licencia
Cratedb	https://crate.io	Crate is licensed under the Apache License, Version 2.0. https://crate.io/legal/license/
Quantumleap	https://github.com/smartsdk/ngsi-timeseries-api	MIT License https://github.com/smartsdk/ngsi-timeseries-api/blob/master/LICENSE
Grafana	https://grafana.com/	grafana está licenciado bajo la Licencia Apache 2.0 https://github.com/grafana/grafana/blob/master/LICENSE

3.4 Desarrollos ad hoc

Interfaz de control del edificio

Desarrollo ad hoc mediante PHP con conexiones API Rest a PLCs en los edificios

Interfaz de control de variables estáticas

Desarrollo ad hoc mediante PHP con almacenamiento en CrateDb

Sistema de KPIs

Desarrollo ad hoc mediante Python y librerías de análisis de datos Pandas con almacenamiento en CrateDb y visualización con Grafana

4 Estructura de la información generada por los edificios

La información a gestionar se resume a modo de ejemplo en la siguiente tabla. Se debe tener en cuenta que la tabla de datos puede variar de edificio a edificio, tanto en número de variables, como en tipología de variables como en nomenclatura.

Cada edificio tiene asociada una tabla de variables. Cada edificio genera una media de 600 datos cada 5 minutos.

Aunque algunos elementos son comunes, la plataforma almacena y gestiona la información en bases de datos específicas para cada edificio.

	VARIABLE A REGISTRAR	UNIDADES	VARIABLE A CONTROLAR	FREC
SALA PRODUCCIÓN				
GENERAL	Temperatura exterior	°C		5 min
	CO ₂ Exterior	ppm		5 min
	Temperatura entrada agua	°C		24h
	Caudal entrada agua	m ³ m ³ /h		60 min
MEDICIÓN CONSUMOS ENERGÍA (nº Según esquema principio sala)	Térmica Primario 1	kWh		5min
	Térmica Primario 2	kWh		5min
	Térmica Primario 3	kWh		5min
	Térmica Bomba de calor 1	kWh		5min
	Térmica Bomba de calor 2	kWh		5min
	Térmica Bomba de calor 3	kWh		5min
	Térmica Secundario 1	kWh		5min
	Térmica Secundario 2	kWh		5min
	Térmica Secundario 3	kWh		5min
	Térmica Primario ACS	kWh		5min
	Térmica Secundario ACS	kWh		5min
	Térmica Distribución (circuito 1)	kWh		5min
	Térmica Distribución (circuito 2)	kWh		5min
	Térmica Distribución (circuito 3)	kWh		5min
	Térmica Distribución ACS	kWh		5min
	Eléctrica sala calderas	kWh		5min
	Eléctrica sistema de apoyo 1	kWh		5min
	Eléctrica sistema de apoyo 2	kWh		5min
	Eléctrica sistema de apoyo 3	kWh		5min
	Eléctrica Bomba de calor 1	kWh		5min
	Eléctrica Bomba de calor 2	kWh		5min
	Eléctrica Bomba de calor 3	kWh		5min
	Eléctrica sistema ventilación viviendas	kWh		60min
	Eléctrica garajes	kWh		5min
	Eléctrica sistemas generales (portal)	kWh		60min
	Eléctrica TIC (RITI)	kWh		60min
CIRCUITO PRIMARIO 1	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m ³ m ³ /h		5 min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
CIRCUITO PRIMARIO 2	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m ³ m ³ /h		5 min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
CIRCUITO PRIMARIO 3	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m ³ m ³ /h		5 min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio

	VARIABLE A REGISTRAR	UNIDADES	VARIABLE A CONTROLAR	FREC
BOMBA DE CALOR 1	Estado	Frio/calor		Cambio
	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5min
	Temperatura acumulador	°C		5min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
BOMBA DE CALOR 2	Estado	Frio/calor		Cambio
	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Temperatura acumulador	°C		5min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
BOMBA DE CALOR 3	Estado	Frio/calor		Cambio
	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Temperatura acumulador	°C		5min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
CIRCUITO SECUNDARIO 1	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
CIRCUITO SECUNDARIO 2	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
CIRCUITO SECUNDARIO 3	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Temperatura depósito inercia	°C		5min
	Estado bomba distribución	ON / OFF		Cambio
CIRCUITO PRIMARIO ACS	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Estado bomba	ON / OFF		Cambio
CIRCUITO SECUNDARIO ACS	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5min
	Estado bomba	ON / OFF		Cambio
CIRCUITO 1 DISTRIBUCIÓN	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Estado bomba	ON / OFF		Cambio
CIRCUITO 2 DISTRIBUCIÓN	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min
	Estado bomba	ON / OFF		Cambio
CIRCUITO 3 DISTRIBUCIÓN	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Caudal	m³ m³/h		5 min

	VARIABLE A REGISTRAR	UNIDADES	VARIABLE A CONTROLAR	FREC
CIRCUITO DISTRIBUCIÓN ACS	Estado bomba	ON / OFF		Cambio
	Temperatura almacenamiento depósito ACS	°C		5min
	Temperatura impulsión	°C		5min
	Temperatura retorno	°C		5min
	Estado bomba recirculación	ON / OFF		Cambio
GESTION HORARIO	Modo verano/invierno	ON / OFF	X	
SILO	CO	ppm		
	Centralita de CO alarma (sala calderas)	Estado		Cambio
	Centralita de incendios alarma (sala calderas)	Estado		Cambio

PATINILLOS / VIVIENDAS (nº viviendas)				FREC
VENTILACIÓN VIVIENDAS (Nº VIVIENDAS)	Estado ON/OFF/AUT del recuperador	ON / OFF/AUT	X	Cambio
	Velocidad de funcionamiento	V1-V2-V3		Cambio
	Humedad relativa en vivienda	% v1 0 - 62 v2- 62 - 70 v3 > 70	X	5min
	Sonda CO ₂ vivienda	ppm v1 - 0 - 1000 v2- 1000 - 1.500 v3 > 1.500	X	5min
	Temperatura ambiental vivienda	°C		5min
	Temperatura admisión recuperador	°C		5min
	Temperatura expulsión recuperador	°C		5min
	Temperatura impulsión vivienda recuperador	°C		5min
	Temperatura extracción vivienda recuperador	°C		5min
	Caudal	m³ m³/h		5min
	Alarma ensuciamiento filtros	ON / OFF		Cambio
	Estado ON/OFF by pass	ON / OFF		Cambio
	Estado ON/OFF bat. postratamiento	ON / OFF		Cambio
ACS (Nº VIVIENDAS)	Caudal	m³ m³/h		60min
	Temperatura entrada	°C		5min
CALEFACCIÓN (Nº VIVIENDAS)	Temperatura ida	°C		60min
	Temperatura retorno	°C		60min
	Térmica Calefacción vivienda	kWh		60min
	Caudal	m³ m³/h		60min
	Estado ON/OFF/AUT electroválvula	ON/OFF/AUT	X	Cambio
	Temperatura mínima	°C	X	Cambio
	Temperatura máxima	°C	X	Cambio
	Energía de servicio	ON / OFF		Cambio

La carga de datos desde el PLC de un edificio se realiza mediante un API Rest específico que permite enviar información de cada elemento controlado a partir de una petición HTTPS GET que posteriormente se almacenará en base de datos.

Cada edificio dispone de su propio mapa de datos, a continuación mostramos un ejemplo:

fiware-servicepath	CADENA_DE_TEXTO
fiware-service	general

Transporte	HTTP		
Interfaz	API Rest		
Protocolo	UltraLight 2.0 sobre HTTP https://fiware-iotagent-ul.readthedocs.io/en/latest/usermanual/index.html		
Forma general	http://(host):7896/iot/d?k={apikey}&i={i}&d={d} {valor} * falta autenticación		
Ejemplo 1	http://(host):7896/iot/d?k=zaqwsx9876&i=prod-energia&d=elec_sala 38		
Ejemplo 2	http://(host):7896/iot/d?k=zaqwsx9876&i=1a-general&d=temp_min 19		
em42	Parámetros		
	i	d	Descripción
	Sala de producción		
	general	nombre	
	general	descr	
	general	direccion	
	general	num_viv	
	general	sup_total	
	general	sup_util	
	prod-general	temp_ext	TEMPERATURA EXTERIOR (TEMP_EXT)
	prod-general	co2_ext	CO2 EXTERIOR (CO2_EXT)
	prod-general	temp_red	TEMPERATURA ENTRADA AGUA FRIA (TEMP_RED)
	prod-general	caudal_red	CAUDAL ENTRADA AGUA FRIA (CAUDAL_RED)
	prod-energia	elec_sala	Electrica sala calderas
	prod-energia	elec_apoyo	Electrica sistemas apoyo
	prod-energia	term_sala	Termica sala calderas
	prod-energia	term_apoyo	Termica sistema apoyo
	prod-energia	term_acs	Termica acs
	prod-energia	term_prim	Termica primario
	prod-energia	term_distr	Termica distribucion
	prod-energia	elec_vent	Electrica sistema ventilacion viviendas
	prod-energia	elec_garaje	Electrica garajes
	prod-energia	elec_comun	Electrica sistemas generales
	prod-energia	elec_tics	Electrica rti
	prod-circuito_printemp	temp_imp	TEMPERATURA IMPULSION (TEMP_IMP)
	prod-circuito_printemp	temp_ret	TEMPERATURA RETORNO (TEMP_RET)
	prod-circuito_printemp	est_bomba_ret	ESTADO BOMBA RETORNO (EST_BOMBA_RET)
	prod-circuito_disttemp	temp_imp	TEMPERATURA IMPULSION (TEMP_IMP)
	prod-circuito_disttemp	temp_ret	TEMPERATURA RETORNO (TEMP_RET)
	prod-circuito_disttemp	est_bomba_ret	ESTADO BOMBA RETORNO (EST_BOMBA_RET)
	prod-acs	temp_deposito	TEMPERATURA ALMACENAMIENTO (TEMP_DEPOSITO)
	prod-acs	temp_imp	TEMPERATURA IMPULSION (TEMP_IMP)
	prod-acs	temp_ret	TEMPERATURA RETORNO (TEMP_RET)
	prod-acs	caudal	CAUDAL (CAUDAL)
	prod-horario	conf_horario	CALENDARIO CONFIGURACION DIARIO/SEMANAL (CONF_HORARIO)
	prod-horario	conf_zonas	CONFIGURACION POR ZONAS (CONF_ZONAS)
	prod-horario	modo	MODOS VERANO/INVIERNO (MODOS)
	prod-silo	co	CO (CO)
	prod-silo	alarma_co	CENTRALITA CO (ALARMA_CO)
	prod-silo	alarma_incendio	CENTRALITA INCENDIOS (ALARMA_INCENDIO)
	Viviendas (en los ejemplos, vivienda 1a)		
	GENERAL (GENERAL)		
	1a-general	pe	POBREZA ENERGETICA (PE)
	1a-general	temp_min	TEMPERATURA CONSIGNA MINIMA (TEMP_MIN)
	1a-general	temp_max	TEMPERATURA CONSIGNA MAXIMA (TEMP_MAX)
	VENTILACION (VENT)		
	1a-vent	est	ESTADO DE EQUIPO (EST)
	1a-vent	vel	VELOCIDAD DE FUNCIONAMIENTO (VEL)
	1a-vent	hum	HUMEDAD EN VIVIENDA (HUM)
	1a-vent	co2	CO2 EN VIVIENDA (CO2)
	1a-vent	temp_viv	TEMPERATURA EN VIVIENDA (TEMP_VIV)
	1a-vent	temp_adm	TEMPERATURA ADMISION RECUPERADOR (TEMP_ADM)
	1a-vent	temp_exp	TEMPERATURA EXPULSION RECUPERADOR (TEMP_EXP)
	1a-vent	temp_imp	TEMPERATURA IMPULSION RECUPERADOR (TEMP_IMP)
	1a-vent	temp_ext	TEMPERATURA EXTRACCION RECUPERADOR (TEMP_EXT)
	1a-vent	caudal	CAUDAL AIRE RECUPERADOR (CAUDAL)
	1a-vent	alarm_filtro	ALARMA FILTROS SUCIOS (ALARM_FILTRO)
	1a-vent	est_bateria_post	ESTADO BATERIA POST-CALENTAMIENTO (EST_BATERIA_POST)
	ACS (ACS)		
	1a-acs	caudal	CAUDAL (CAUDAL)
	1a-acs	e_acs	ENERGIA ACS (E_ACS)
	1a-acs	temp	TEMPERATURA ENTRADA (TEMP)
	CALEFACCION (CAL)		
	1a-cal	temp_imp	TEMPERATURA IMPULSION (TEMP_IMP)
	1a-cal	temp_ret	TEMPERATURA RETORNO (TEMP_RET)
	1a-cal	e	ENERGIA CALEFACCION (E)
	1a-cal	caudal	CAUDAL (CAUDAL)
	1a-cal	est_ev	ESTADO ELECTROVALVULA (EST_EV)

El método de acceso a la plataforma por parte de los autómatas de edificio se resume en la tabla a continuación.

Transporte	HTTPS
Interfaz	API Rest
Protocolo	UltraLight 2.0 sobre HTTP(S) https://fiware-iotagent-ul.readthedocs.io/en/latest/usermanual/index.html
Forma general	https://{{host}}:7896/iot/d?k={{apikey}}&i={{i}}&d={{d}} {{valor}} * falta autenticación
Ejemplo 1	https://{{host}}:7896/iot/d?k=-----9876&i=prod-energia&d=elec_sala 38
Ejemplo 2	https://{{host}}:7896/iot/d?k=-----9876&i=1a-general&d=temp_min 19

A partir de las variables indicadas en rojo en la URL de acceso general:

<https://{{host}}:7896/iot/d?k={{apikey}}&i={{i}}&d={{d}}|{{valor}}>

Se muestra a continuación un ejemplo de tabla de parámetros que podría corresponder a un edificio concreto.

Parámetros		
i	d	Descripción
Sala de producción		
general	nombre	
general	descr	
general	direccion	
general	num_viv	
general	sup_total	
general	sup_util	
prod-general	temp_ext	TEMPERATURA EXTERIOR {TEMP_EXT}
prod-general	co2_ext	CO2 EXTERIOR {CO2_EXT}
prod-general	temp_red	TEMPERATURA ENTRADA AGUA FRIA {TEMP_RED}
prod-general	caudal_red	CAUDAL ENTRADA AGUA FRIA {CAUDAL_RED}
prod-energia	elec_sala	Electrica sala calderas
prod-energia	elec_apoyo	Electrica sistemas apoyo
prod-energia	term_sala	Termica sala calderas
prod-energia	term_apoyo	Termica sistema apoyo
prod-energia	term_acs	Termica acs
prod-energia	term_prim	Termica primario
prod-energia	term_distr	Termica distribucion
prod-energia	elec_vent	Electrica sistema ventilacion viviendas
prod-energia	elec_garaje	Electrica garajes
prod-energia	elec_comun	Electrica sistemas generales
prod-energia	elec_tics	Electrica riti
prod-circuito_prim	temp_imp	TEMPERATURA IMPULSION {TEMP_IMP}
prod-circuito_prim	temp_ret	TEMPERATURA RETORNO {TEMP_RET}
prod-circuito_prim	est_bomba_ret	ESTADO BOMBA RETORNO {EST_BOMBA_RET}
prod-circuito_distr	temp_imp	TEMPERATURA IMPULSION {TEMP_IMP}
prod-circuito_distr	temp_ret	TEMPERATURA RETORNO {TEMP_RET}

prod-circuito_distr	est_bomba_ret	ESTADO BOMBA RETORNO {EST_BOMBA_RET}
prod-ac	temp_deposito	TEMPERATURA ALMACENAMIENTO {TEMP_DEPOSITO}
prod-ac	temp_imp	TEMPERATURA IMPULSION {TEMP_IMP}
prod-ac	temp_ret	TEMPERATURA RETORNO {TEMP_RET}
prod-ac	caudal	CAUDAL {CAUDAL}
prod-horario	conf_horario	CALENDARIO CONFIGURACION DIARIO/SEMANAL {CONF_HORARIO}
prod-horario	conf_zonas	CONFIGURACION POR ZONAS {CONF_ZONAS}
prod-horario	modo	MODO VERANO/INVIERNO {MODO}
prod-silo	co	CO {CO}
prod-silo	alarma_co	CENTRALITA CO {ALARMA_CO}
prod-silo	alarma_incendio	CENTRALITA INCENDIOS {ALARMA_INCENDIO}
Viviendas (en los ejemplos, vivienda 1): a		
GENERAL		
1a-general	pe	POBREZA ENERGETICA {PE}
1a-general	temp_min	TEMPERATURA CONSIGNA MINIMA {TEMP_MIN}
1a-general	temp_max	TEMPERATURA CONSIGNA MAXIMA {TEMP_MAX}
VENTILACION		
1a-vent	est	ESTADO DEL EQUIPO {EST}
1a-vent	vel	VELOCIDAD DE FUNCIONAMIENTO {VEL}
1a-vent	hum	HUMEDAD EN VIVIENDA {HUM}
1a-vent	co2	CO2 EN VIVIENDA {CO2}
1a-vent	temp_viv	TEMPERATURA EN VIVIENDA {TEMP_VIV}
1a-vent	temp_adm	TEMPERATURA ADMISION RECUPERADOR {TEMP_ADM}
1a-vent	temp_exp	TEMPERATURA EXPULSION RECUPERADOR {TEMP_EXP}
1a-vent	temp_imp	TEMPERATURA IMPULSION RECUPERADOR {TEMP_IMP}
1a-vent	temp_ext	TEMPERATURA EXTRACCION RECUPERADOR {TEMP_EXT}
1a-vent	caudal	CAUDAL AIRE RECUPERADOR {CAUDAL}
1a-vent	alarm_filtro	ALARMA FILTROS SUCIOS {ALARM_FILTRO}
1a-vent	est_bateria_post	ESTADO BATERIA POST-CALENTAMIENTO {EST_BATERIA_POST}
ACS		
1a-ac	caudal	CAUDAL {CAUDAL}
1a-ac	e_acs	ENERGIA ACS {E_ACS}
1a-ac	temp	TEMPERATURA ENTRADA {TEMP}
CALEFACCION		
1a-cal	temp_imp	TEMPERATURA IMPULSION {TEMP_IMP}
1a-cal	temp_ret	TEMPERATURA RETORNO {TEMP_RET}
1a-cal	e	ENERGÍA CALEFACCION {E}
1a-cal	caudal	CAUDAL {CAUDAL}
1a-cal	est_ev	ESTADO ELECTROVALVULA {EST_EV}

5 Características funcionales de la plataforma

5.1 Gestión de eventos

En este apartado se definen y explican todos los desarrollos correspondientes a la “Configuración del Interfaz de Eventos” de la Plataforma Nasuvinsa.

Los módulos utilizados en para la configuración del interfaz de eventos son: IoTAgent, Orion+MongoDB, Mosquitto, Rabbitmq, Quantumleap, CrateDB y Redis.

Se definen como “eventos a recibir de un edificio” como aquellas variables que se envían desde el edificio en cuestión hasta la plataforma. Estos eventos son definidos y dados de alta mediante un fichero que lanza una orden de definición.

El fichero de definición se compone de dos partes diferenciadas.

Definición del entorno:

En esta sección se define el entorno de trabajo que equivale a un edificio concreto. Así pues, para cada nuevo edificio, se generará un nuevo entorno.

Cada fiware-service define un entorno de trabajo que en el caso de la plataforma se asociará a cada edificio por separado. De manera que todas las variables de todos los edificios quedan totalmente diferenciadas por este identificador único.

Definición de los eventos:

Las variables que cada edificio genera se definen también en este fichero. Se asocian internamente a cada entorno, de forma que cada edificio pueda tener variables idénticas y variables diferentes. Lo habitual será que las variables de zonas comunes, sala de calderas, etc... sean idénticas a todos los edificios, y las variables de viviendas puedan coincidir o no en función de su identificador y de los elementos de campo.

Puede obtenerse información detallada del proceso de definición de entornos y eventos en la documentación de Fiware IoTAgent UltraLight 2.0: <https://fiware-zone.readthedocs.io/es/latest/iot-agent.html>

Algunas de las variables de cada edificio tienen un umbral asociado dentro del cual, el estado de la variable se considera normal, y fuera del cual, el estado de la variable pasa a un estado de “alarma”. Según el tipo de variable, un valor concreto puede considerarse alarma. Dada la naturaleza de la plataforma, se ha optado por tratar las alarmas a modo de suscripciones al context broker, de forma que será éste quien conozca los umbrales correctos de cada variable, y en el caso de que dichas variables se salgan de su umbral, generen un trigger y registren dicho evento y valor.

Para simplificar la carga de estos umbrales de alarma, se ha definido una estructura en un fichero .CSV que será parseado por un script en Python que se encargará de refrescar las suscripciones de alarma del context broker.

Los módulos implicados en este apartado son IoTAgent, Orion+Mongo, QuantumLeap y Cratedb.

Se dispone de un entorno de desarrollo en lenguaje Python que permite automatizar la creación de alarmas a partir de un fichero CSV con rangos de valores aceptables.

Dado el listado completo de variables, cada edificio tendrá unas u otras variables a utilizar. El proceso para configurar qué variables se dan de alta para cada edificio se genera mediante una hoja de cálculo Excel que tras ser parseada genera ficheros de texto intermedios que permiten generar semiautomáticamente los scripts que mediante conexión API Rest configuran los eventos del edificio.

Como se ha avanzado con anterioridad, se utiliza la propiedad del sistema *fiware-service* como identificador único para cada edificio. Se trata de un entorno de trabajo diferenciado para cada edificio. En otras topologías se suele denominar galaxia o universo para designar a un subconjunto muy extenso dentro de una aplicación, que *no puede mezclarse salvo por un gestor global externo* como sería en este caso la Plataforma Nasuvinsa.

En otras palabras, es una forma lógica de separar virtualmente todas las variables de cada uno de los edificios de la plataforma. De modo que están totalmente aislados unos de otros. Y sólo la plataforma podrá solicitar datos de cada uno de ellos y realizar el tratamiento que sea preciso.

Esta definición se realiza en primera instancia en *IoTAgent* que es quien *recibe los datos del exterior, los organiza y estructura, y envía a Orion para que sean almacenados*. La definición del entorno ya se ha documentado en el apartado REV-1. A continuación se documenta cómo consultar los entornos existentes y sus datos asociados. Todo queda recogido en el fichero *entornos.rest*. Al lanzar la consulta (en el ejemplo, al edificio *entremutilvas_42*) se devuelven los parámetros asociados.

De este modo *Las variables pueden cambiar de un edificio a otro y por lo tanto cada uno de los edificios tendrá un tabla de diferente*.

La variabilidad de los edificios es muy alta, de modo que es previsible que ocurran casos como, por ejemplo, un edificio con múltiples salas de caldera, múltiples portales, diferentes tipos de agrupaciones, etc. De modo que sólo habrá que adaptar las nuevas nomenclaturas de variables, identificadores únicos, y generar los nuevos archivos de configuración para el caso concreto.

5.2 Interfaz de estado del edificio

Este interfaz tiene como función principal mostrar los parámetros de un edificio y modificar sus variables de funcionamiento. Es la principal aplicación con la que el operador se comunicará con el sistema.

El estado del edificio se muestra en tiempo real mediante Grafana. El sistema de modificación de variables de control es una aplicación desarrollada con lenguaje PHP que presenta un formulario con los valores actuales del autómata y permite modificarlos mediante comunicación API Rest.

Puesto que la herramienta base que muestra la interfaz de usuario es Grafana, para cualquier modificación o ampliación se debe recurrir a la documentación de desarrollo disponible en su página web.⁴

⁴ Se puede acceder a la documentación de desarrollo de Grafana desde el enlace <https://grafana.com/docs/>

5.3 Variables estáticas

Cada edificio cuenta con una serie de variables estáticas (aproximadamente 50) que se establecen manualmente. Se almacenan del mismo modo que las variables dinámicas (generadas por el edificio) mediante un desarrollo ad hoc en lenguaje PHP.

La carga inicial se realiza generando un script para automatización de conexiones API Rest con la ayuda de una hoja de cálculo. La secuencia de comandos API Rest resultante se debe enviar a la plataforma.

Tras la configuración de variables estáticas mediante el procedimiento descrito anteriormente, es posible acceder a la visualización y modificación de valores mediante la aplicación web desarrollada en php indicada anteriormente.

El interfaz de usuario es el mostrado en la imagen

Variable	Valor	Unidad	Check	Acción
Superficie total edificio	4600	m²	☐	R
Rendimiento teórico sistema calefacción_2	60	%	☐	R
Rendimiento teórico sistema ACS_1	85	%	☐	R
Grados día		°C	☐	R
Superficie de referencia energética		m²	☐	R
Pérdidas distribución ACS		kWh	☐	R
Temperatura vivienda límite superior invierno		°C	☐	R
Humedad vivienda límite inferior verano		%	☐	R
Concentración vivienda límite superior CO₂		ppm	☐	R
Coefficiente de paso electricidad a CO₂		kgCO₂/kWh	☐	R
Coefficiente de paso electricidad a energía primaria		kWhEP/kWhEF	☐	R
Estación climática		verano/invierno	☐	R
Fecha última recarga silo			☐	R
número de viviendas		uds.	☐	R
Rendimiento teórico sistema calefacción_3		%	☐	R
Rendimiento teórico sistema ACS_2		%	☐	R
Superficie opaca		m²	☐	R
Pérdidas distribución calefacción		kWh	☐	R
Temperatura vivienda límite superior verano		°C	☐	R
Temperatura vivienda límite inferior invierno		°C	☐	R
Humedad vivienda límite superior verano		%	☐	R
Humedad vivienda límite inferior invierno		%	☐	R
Temperatura límite sobrecalentamiento		°C	☐	R
Coefficiente de paso gas natural a CO₂		kgCO₂/kWh	☐	R
Coefficiente de paso gas natural a energía primaria		kWhEP/kWhEF	☐	R
Recarga de astilla	500	m³	☐	R

Variable	Valor
Superficie total edificio OK	4600 m² ☐ m R

Para modificar el valor de una variable es necesario colocar el valor deseado en el campo 'Valor', marcar el checkbox 'm' y pulsar el botón 'R':

Redimiento teórico sistema ACS_1	70 % ☒ m R
----------------------------------	---

5.4 KPIS

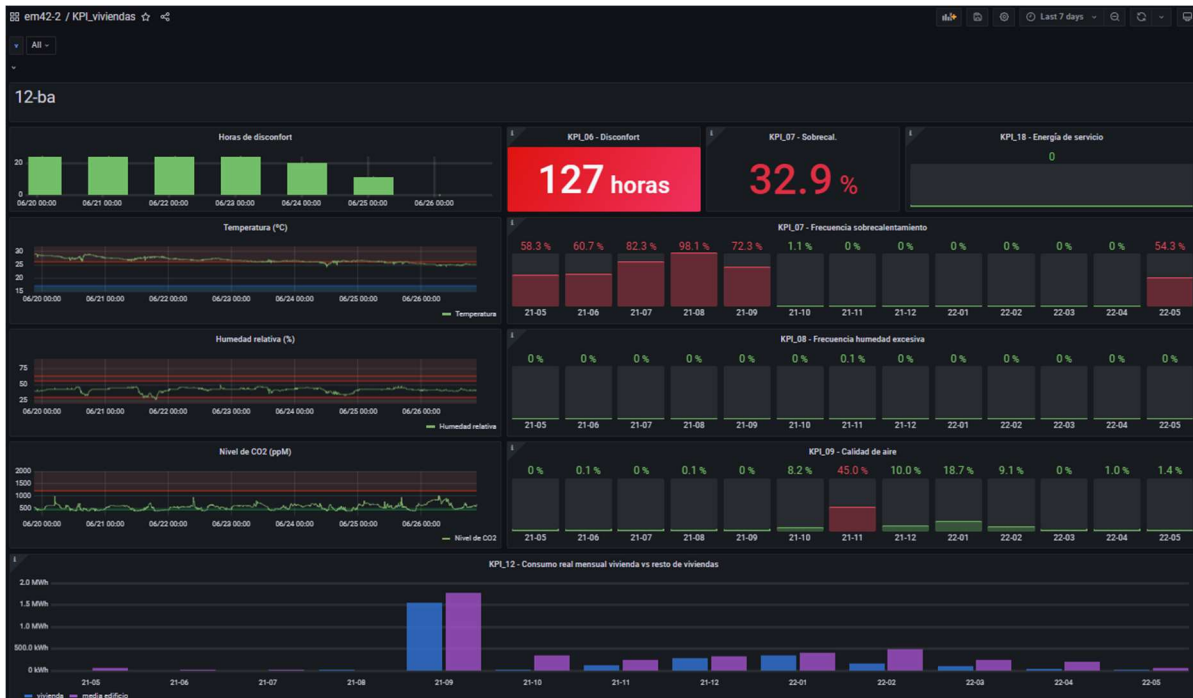
La plataforma cuenta con la sección KPIS en la que se puede consultar una serie de indicadores agrupados en dos niveles: KPIS de edificio y KPIS de vivienda:

Tabla 1 Listado de KPIS

nom.	KPI EDIFICIO	Unidad
KPI 1	COP real Bomba Calor	%
KPI 2	Pérdidas por distribución	%
KPI 3	Autonomía del silo	Días
KPI 4	Generación renovable	%
KPI 10	Consumo medio mensual	kWh/año
KPI 11	Consumo medio mensual vs serie anterior	kWh/año
KPI 19 1	Reducción anual energía consumida	kWh
KPI 19 2	Reducción económica anual	€
KPI 20 1	Reducción anual emisiones CO ₂	%
KPI 20 2	Reducción anual emisiones CO ₃	tCO ₂

nom.	KPI VIVIENDAS	Unidad
KPI 6	Disconfort en vivienda	h
KPI 7	Frecuencia sobrecalentamiento	%
KPI 8	Frecuencia humedad excesivamente alta	%
KPI 9	Calidad aire (% frecuencia fuera del rango)	%
KPI 12	Consumo medio mensual vs otras viviendas del edificio	kWh/año
KPI 18	Energía de servicio	kWh





Los scripts ejecutables desarrollados con lenguaje Python se ejecutan diariamente mediante programación cron. Los resultados de la ejecución de cada script se almacenan en tablas específicas preprocesadas en la base de datos principal de cada edificio y se pueden visualizar mediante los correspondientes dashboards de Grafana.

5.5 Interfaz de gráficos

El interfaz de gráficos tiene como objetivo mostrar la evolución temporal de las variables de tal manera que nos permitan visualizar el comportamiento de un edificio, comparar dos meses de un mismo edificio o comparar los datos de varios edificios

La interfaz de gráficos está desarrollada con Grafana y el almacén de datos es el SGBD principal CrateDb. La información de cada edificio se almacena en una única tabla de la base de datos específica para ese edificio.

5.6 Acceso a datos de edificios por terceras aplicaciones mediante API Rest

La plataforma **Nasuvinsa Smart Building** permite ofrecer información histórica sobre consumo energético de los edificios gestionados mediante el [API Rest descrito en este documento](#). Una vez definido y acotado el conjunto de series temporales expuestos mediante API Rest con las facilidades de parametrización que el propio API Rest proporciona, es posible interrogar al sistema para obtener información conforme a los métodos descritos a continuación.

Definición estructural de los componentes que forman parte del sistema de acceso de terceras partes.

La plataforma implementa una separación lógica de la información generada por los edificios controlados que son objeto de exportación mediante consulta al API Rest.

Dicho subconjunto de información se almacena en una base de datos de [series temporales gestionada por CrateDB](#)⁵. El acceso a la información se realiza mediante el componente Fiware Quantumleap⁶.

La información obtenida tras realizar las consultas mediante el API Rest⁷ propuesto por Quantumleap se ajusta al estándar Fiware NGSIv2. Dicho estándar se ha denominado NGSI-TSDB (NGSSI Time Series Data Base) y los detalles de su especificación se encuentran en el sitio web de Quantumleap⁸.

Como en el resto de componentes de la plataforma, la selección del edificio sobre el que se realizala consulta de datos históricos se hace mediante la cabecera HTTP fiware-service.

Cada edificio dispone de su identificador único que deberá ser asignado a la cabecera fiware-service. La tabla de identificadores únicos se proporciona en la actualización de este documento convenientemente, así como la relación de variables expuestas mediante API Rest.

5.7 Acceso específico de sistemas Stardust

La plataforma **Nasuvinsa Smart Building** permite ofrecer información histórica sobre consumo energético de los edificios gestionados mediante el API Rest específico definido en el documento

⁵ Puede ver información adicional de CrateDB en el sitio web <https://crate.io/>

⁶ Puede ver información adicional de Quantumleap en el sitio web <https://quantumleap.readthedocs.io/en/latest/>

⁷ En los anexos de este documento se detalla y ejemplifica el formato de los accesos a la información ofrecida por la plataforma a terceras partes

⁸ Puede acceder a la especificación completa de NGSI-TSDB en el sitio web <https://app.swaggerhub.com/apis/smartsdk/ngsi-tsdb/0.7>

“API de acceso a terceras partes”. Adicionalmente al acceso remoto completo, existe la posibilidad de acceder a información resumida de ciertos KPIs. En este manual se describe el método de acceso al API Rest de KPIs de la Plataforma Nasuvinsa.

KPIs accesibles mediante API

Aunque la Plataforma Nasuvinsa genera un número amplio de KPIs, solo un subconjunto es accesible mediante API. Concretamente, los KPIs 19-1, 20-1 y 20-2 para los edificios EM42, AR34 y RI62 generan información accesible mediante API. Se describe a continuación la información generada por estos tres KPIs.

KPI 19-1. Reducción anual de energía consumida.

Comparación del consumo real del edificio por meses frente al consumo teórico según criterios CTE-2019

Unidad de medida: kWh

KPI 20-1. Porcentaje de reducción anual emisiones CO2

Diferencia entre las emisiones de CO2 y las emisiones que se producirían si el edificio siguiera criterios CTE-2019 y solo se usara gas natural.

Unidad de medida: %

KPI 20-2. Reducción anual emisiones CO2

Diferencia entre las emisiones de CO2 y las emisiones que se producirían si el edificio siguiera criterios CTE-2019 y solo se usara gas natural.

Unidad de medida: tCO2

Acceso a KPIs mediante API Rest

El formato de las solicitudes de acceso a la información de KPIs generada por la Plataforma Nasuvinsa es el siguiente:

Transporte: HTTPS

Método: GET

Autenticación: HTTP Basic Auth⁹

URL: [https://plataforma.nasuvinsa.es/kpi/\[EDIFICIO\]/\[KPI\].json](https://plataforma.nasuvinsa.es/kpi/[EDIFICIO]/[KPI].json)

- El sistema de acceso a esta información se ha desarrollado como parte del sistema de KPIs: programación ad hoc mediante lenguaje Python que genera varios ficheros de texto con formato Json y son servidos por Nginx.

Por ejemplo, el resultado es un fichero como el siguiente:

⁹ Al contrario que para otros puntos de acceso en la Plataforma Nasuvinsa, el subsistema de acceso a KPIs mediante API solo requiere la cabecera HTTP ‘Authorization’, por lo que se puede acceder a la información desde un navegador web estándar disponiendo de credenciales de acceso válidas.

```
▼ columns:
  0: "edificio_id"
  1: "kpi_id"
  2: "fecha"
  3: "v_01"
```

```
▼ data:
  ▼ 0:
    0: "em42"
    1: "KPI_19"
    2: "2020-07"
    3: 28.3977732992
```

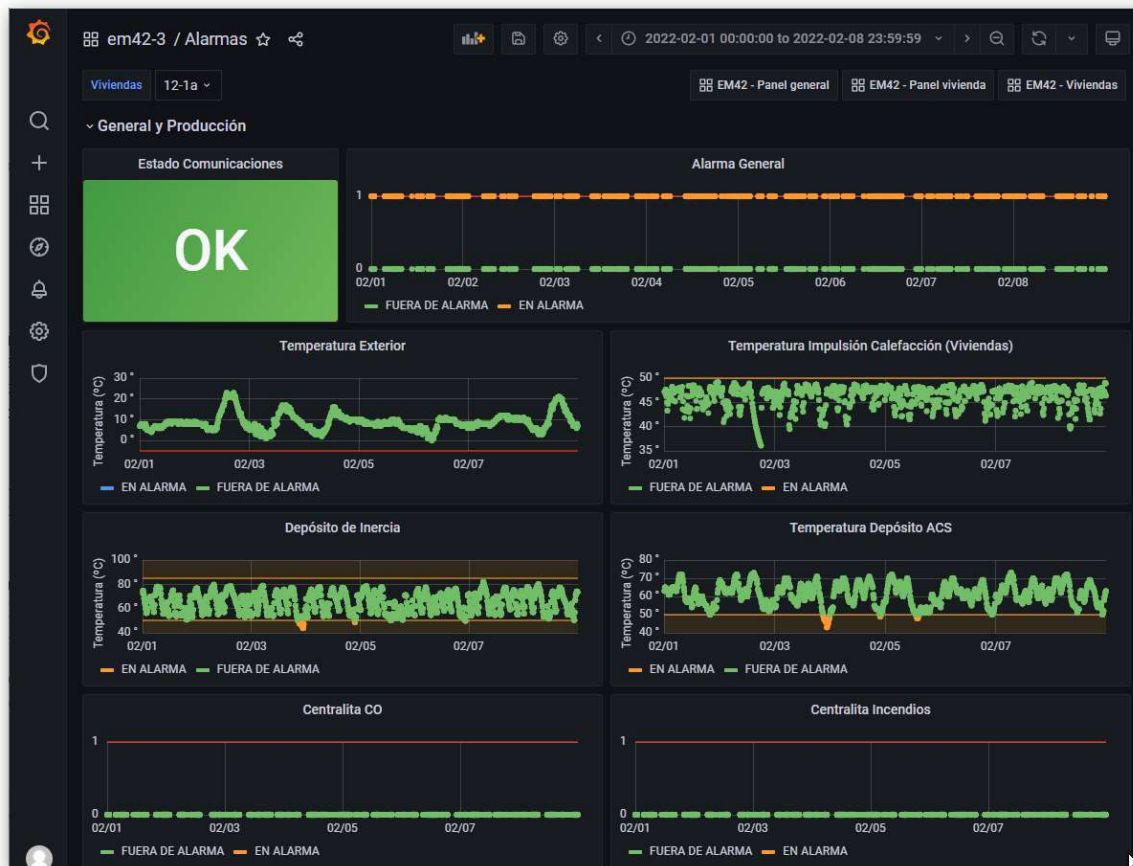
```
  ▼ 1:
    0: "em42"
    1: "KPI_19"
    2: "2020-08"
    3: 25.6258112066
```

```
  ▼ 2:
    0: "em42"
```

5.8 Interfaz de alarmas

Este interfaz tiene como objetivo mostrar los eventos configurados como alarma que ha recibido el sistema. Su función principal es dar al operador una herramienta para conocer de forma rápida y clara el estado de los edificios.

En Grafana, se define una serie de paneles generales del edificio y de las viviendas donde se muestran los eventos recibidos cuyo valor se encuentre fuera de límites aceptables.



Los valores que se consideran alarma se configuran directamente en el dashboard de Grafana. No es necesaria una configuración externa

Adicionalmente, en la sección de acceso general a edificios, se muestra alguna de las alarmas principales en un cuadro resumen que tiene como objetivo avisar sobre cualquier incidencia que requiera de una actuación urgente por parte de los gestores de los edificios:

Nasuvinsa

PLATAFORMA NASUVINSA

PLATAFORMA INTELIGENTE PARA LA GESTIÓN DE VIVIENDA PÚBLICA.

Nasuvinsa
Navarra de Suelo y Vivienda, S.A.

Edificio	General	Centralita CO	Centralita Incendios	Comunicación	Disconfort	Viviendas con disconfort
Entremutillas 42	OFF	OFF	OFF	OFF	7	12-1a, 12-2d, 12-3b, 12-...
Ardoi 34					4	16_3a, 16_3b, 18_ba, 18...
Ripagaina 62	OFF	OFF	OFF	OFF	0	
Barañain 29					4	15_2a, 15_4a, 16_1b, 16...
Rotxapea 20					12	6_1a, 6_1b, 6_2a, 6_2b, ...
Compañía 04						

Alarmas general

Panel de alarmas general con información de estado crítico para todos los edificios

Acceder

ALMS

5.9 Control de accesos y permisos

El filtro de protección de accesos a los componentes de la plataforma se ha diseñado mediante la integración de un proxy HTTP inverso **Apache httpd** colocado entre el proxy inverso general de Nasertic y los servidores funcionales de la Plataforma Nasuvinsa.

El sistema de gestión de permisos de la **PLATAFORMA Nasuvinsa Smart Building** se fundamenta en la autenticación **Basic Auth** proporcionada por el proxy inverso Apache que se encuentra entre el proxy exterior de Nasertic y la subred formada por los servidores que alojan la plataforma.

En el subsistema de gestión de permisos se ha implementado un esquema de pertenencia a grupos de usuarios a los que se asigna una capacidad de acceso de varios niveles.

Se define un total de 5 niveles de acceso o roles para limitar los accesos a ciertas secciones de la plataforma por dos motivos:

1. Limitación de funcionalidades
2. Limitación por edificio

En la tabla a continuación se detalla las posibilidades de acceso a los usuarios que pertenecen a cada perfil de usuario

PLATAFORMA SGWE								
PERMISOS PLATAFORMA NSV	ACCESO	PROMOCIONES	GENERAL	PANEL VIVIENDA	VIVIENDAS	HISTÓRICO SALA	HISTÓRICO VIVIENDAS	ALARMAS
	EDIFICIO	ALARMAS GENERAL						
USUARIO NIVEL 0	TODAS	TODAS						
USUARIO NIVEL 1	TODAS	TODAS						
USUARIO NIVEL 2	GRUPO	TODAS						
USUARIO NIVEL 3	GRUPO	TODAS						
USUARIO NIVEL 4	GRUPO	TODAS						
PERMISOS PLATAFORMA NSV	DISCONFORT	CONTROLES	VARIABLES	VARIABLES	SINÓPTICO	KPI EDIFICIO	KPI VIVIENDA	CONSUMOS ENERGÉTICOS
			SILO	ESTÁTICAS				
USUARIO NIVEL 0								
USUARIO NIVEL 1								
USUARIO NIVEL 2								
USUARIO NIVEL 3								
USUARIO NIVEL 4								

Adicionalmente, se debe configurar Grafana incorporando los grupos y usuarios adaptados a los niveles de acceso de cada edificio gestionado.

La creación de grupos y asociación a dashboards según perfil se realiza inicialmente en la etapa de integración del edificio, pero la creación de usuario e inclusión en perfil se debe realizar manualmente tras la creación del usuario descrita en el apartado 'Nivel de acceso general' anterior. La configuración de usuario se efectúa a nivel de edificio.