

Índice

1. Hoja de control	2
2. Introducción	3
3. Acceso al API	3
4. Extraer identificadores	5
4.1 Lista de instalaciones	5
4.2 Lista de controles de una instalación	6
4.3 Obtención de la lista de variables	8
5. Histórico de datos de una o varias variables	10
5.1 Petición personalizada de parámetros	10
5.2 Petición desde una gráfica creada	14
6. Consulta de variables de categoría Alarma	16
6.1 Consultar el total de alarmas activas	17
6.2 Consultar alarmas activas en una instalación	18
6.3 Consultar alarmas activas en un control	20
7. Proceso de consulta de datos	21

1. Hoja de control

Proyecto	Kiconex		
Documento	Manual uso API V4.00		
Nombre del Fichero	MAN - Uso API V4.00.pdf		
Versión	V. 4.00	Fecha versión	03/11/2023
Aprobado por	A. Luque	Fecha aprobación	03/11/2023

REGISTRO DE CAMBIOS

Versión	Causa del cambio	Responsable del cambio	Área	Fecha del cambio
V. 1.00	Redacción informe	G. Sánchez	kiconex	27/04/2021
V 2.00	Cambio consulta múltiple status a GET	A. Cobos	kiconex	05/04/2022
V 3.00	Ampliación con todo tipo de peticiones	CC. Sánchez	kiconex	02/10/2023
V 4.00	Cambios por las modificaciones de api en plataforma	J.C.Ferreira A. Cobos	Kiconex	03/11/2023

2. Introducción

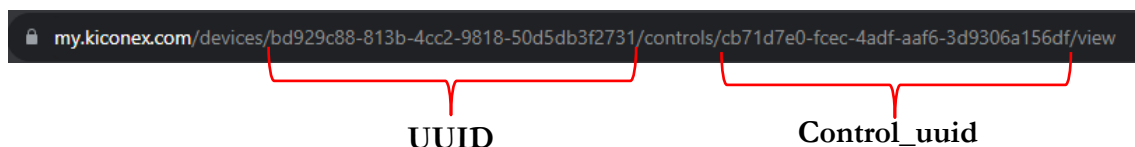
El objetivo de esta guía es facilitar al usuario el uso y consumo de la API implementada en la plataforma *app.kiconex.com*.

Para poder utilizar esta funcionalidad, es necesario contar con un usuario de la plataforma. Las peticiones que podrá realizar el usuario se verán siempre limitadas por el paquete contratado en cada instalación.

Todas las consultas realizadas a través de esta herramienta siguen la misma metodología. Por tanto, y a fin de facilitar la lectura de esta guía/manual, solo se escribirá una vez la ruta raíz, y se obviará a lo largo de todo el documento:

- <https://my.kiconex.com/api/v1>

Cada elemento de la plataforma se caracteriza mediante un identificador único. Las peticiones se realizan encadenando estos identificadores. Por ejemplo, para consultar una variable en concreto, habrá que indicar el identificador de la instalación y del control que la aloja, así como el de la variable. Los identificadores de la instalación y control se pueden obtener fácilmente desde la URL del navegador:



El identificador de cada variable se puede obtener mediante una petición (ver apartado 4).

3. Acceso al API

Para poder hacer consultas, el primer paso será conseguir el token de acceso. Para ello, se deberá realizar la siguiente petición:

- **Método:** POST
- **URL:** /login

En el cuerpo del mensaje, se deberá enviar el usuario y la contraseña en formato json. De esta manera, la consulta quedaría como:

- Petición

POST	https://app.kiconex.com/api/v1/login
------	--------------------------------------

- Cuerpo:

Params	Authorization	Headers (8)	Body ●	Pre-request Script	Tests	Settings
☐ none	☐ form-data	☐ x-www-form-urlencoded	☒ raw	☐ binary	☐ GraphQL	☒ JSON ▼
<pre>1 { 2 "user_name": "test.user", 3 "user_password": "test.123456@" 4 }</pre>						

- Respuesta:

```
"token": "P1dwnm18[REDACTED]",
```

Una vez obtenido el token, ya podrán realizarse consultas como si se estuviese navegando por la plataforma. Este token solo es necesario renovarlo si se activa el “tiempo de vida” en la configuración del perfil de usuario (es decir, que el usuario haga “logout” automático cada X tiempo de inactividad).

Para realizar las sucesivas peticiones, habrá que incluir en la cabecera de la petición este token, de la siguiente manera:

- **key:** Authorization
- **value:** Basic token

Ejemplo:

☒ Authorization	Basic [REDACTED]
---	------------------

4. Extraer identificadores

4.1 Lista de instalaciones

Es posible conocer los identificadores de cada instalación (UUID) a través de la siguiente petición API:

- **Método:** GET
- **URL:** /devices

Ejemplo:

- Petición:

GET	▼	https://app.kiconex.com/api/v1/devices
-----	---	--

- Cabecera:

GET

https://app.kiconex.com/api/v1/devices/bd929c88-813b-4cc2-9818-50d5db3f2731

Params

Authorization

Headers (7)

Body

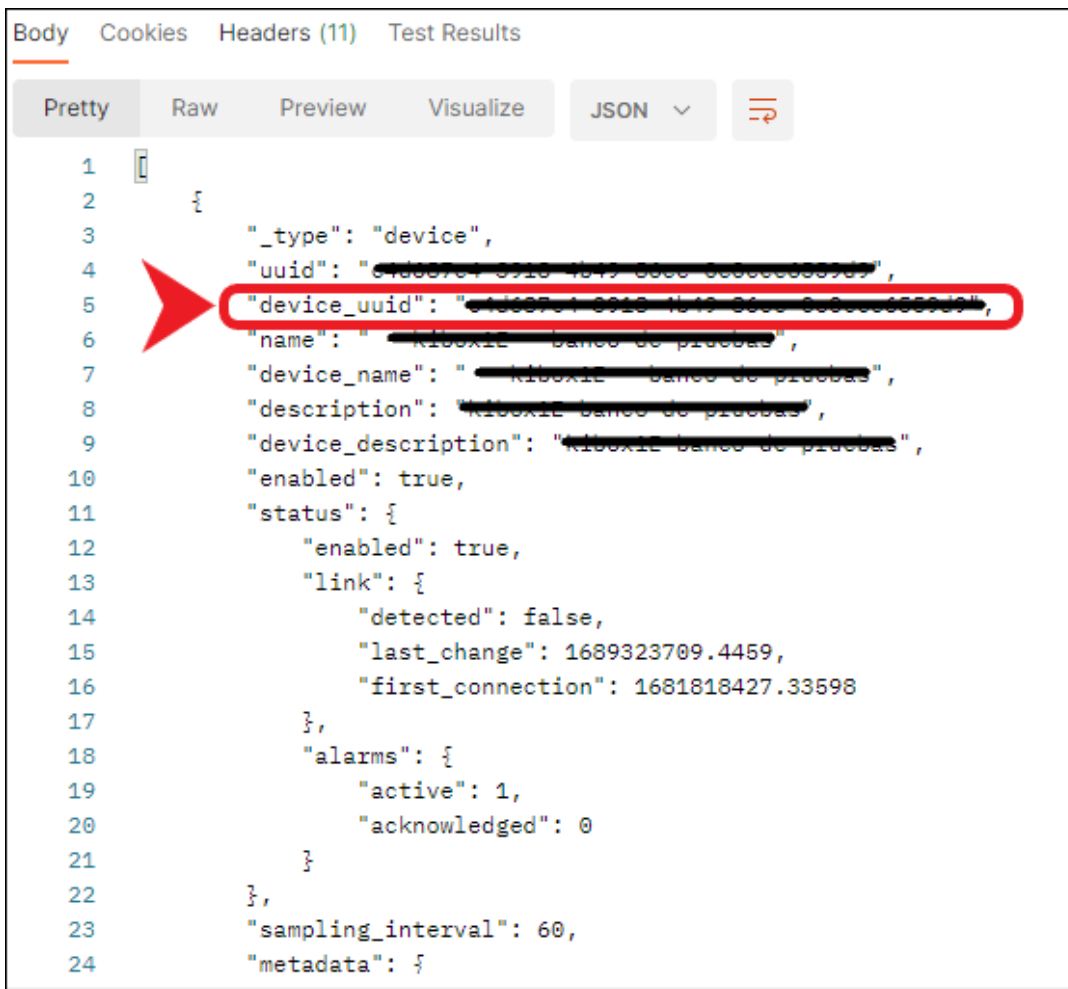
Pre-request Script

Tests

Settings

☒	Authorization	 Basic
☒	Postman-Token	 <calculated when request is sent>
☒	Host	 <calculated when request is sent>
☒	User-Agent	 PostmanRuntime/7.33.0
☒	Accept	 */*
☒	Accept-Encoding	 gzip, deflate, br
☒	Connection	 keep-alive
	Key	Value

- Respuesta:



```
1  {
2
3    "_type": "device",
4    "uuid": "c4d00704-0910-4b49-8000-000000000000",
5    "device_uuid": "c4d00704-0910-4b49-8000-000000000000",
6    "name": "kibox12 - banco de pruebas",
7    "device_name": "kibox12 - banco de pruebas",
8    "description": "kibox12 - banco de pruebas",
9    "device_description": "kibox12 - banco de pruebas",
10   "enabled": true,
11   "status": {
12     "enabled": true,
13     "link": {
14       "detected": false,
15       "last_change": 1689323709.4459,
16       "first_connection": 1681818427.33598
17     },
18     "alarms": {
19       "active": 1,
20       "acknowledged": 0
21     }
22   },
23   "sampling_interval": 60,
24   "metadata": {
```

4.2 Lista de controles de una instalación

De cada instalación es posible exportar la lista de controles existentes para conocer el UUID de cada uno, a través de la siguiente petición API:

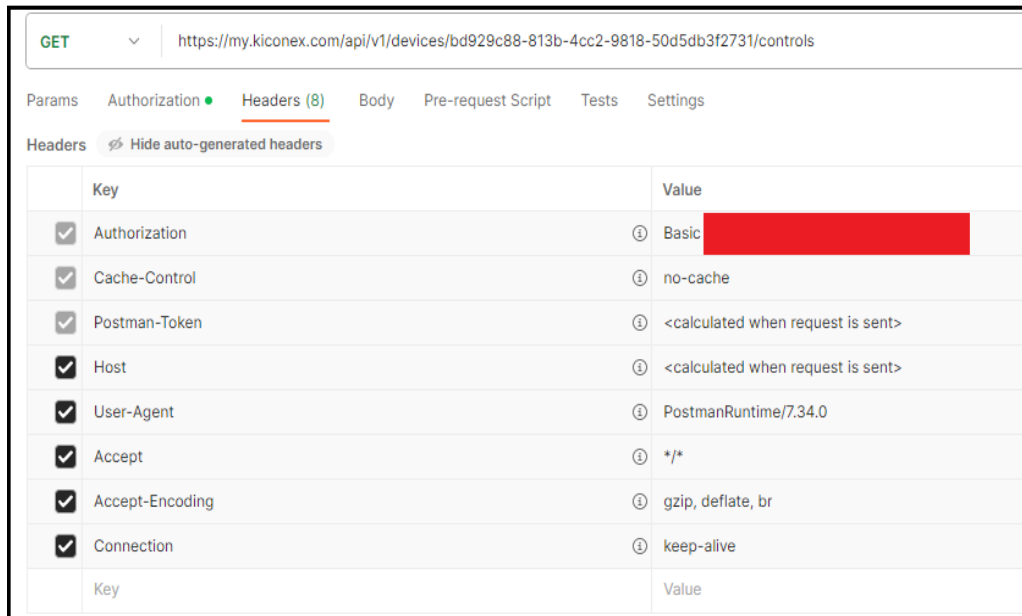
- **Método:** GET
- **URL:** /devices/*uuid*/controls
- **UUID:** hace referencia al identificador de la instalación. Obtenido de la barra de navegación o de la petición API del punto 4.1

Ejemplo:

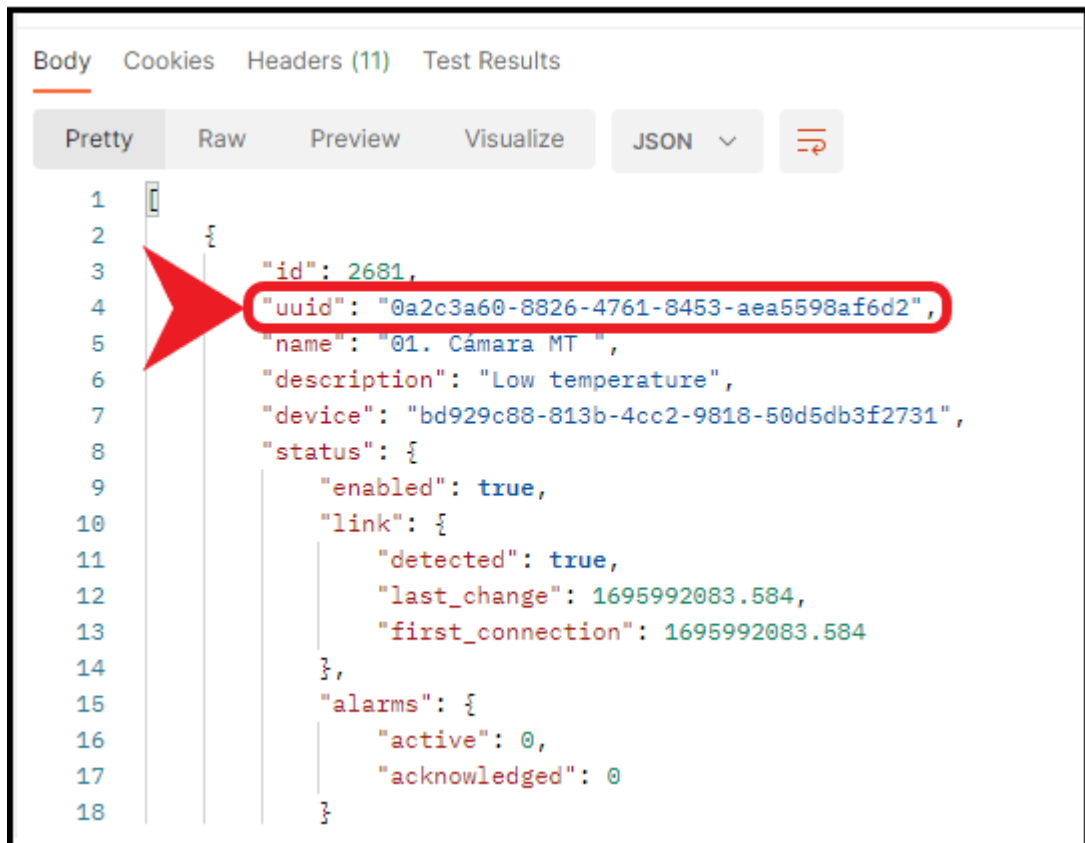
- Petición:



- Cabecera:



- Respuesta:



4.3 Obtención de la lista de variables

Como se indicaba al principio, cada variable tiene asociada un identificador único, que sirve para consultar su información (nombre, valor instantáneo, unidad, histórico, etc.). Este identificador nunca cambia dentro de un control, por lo que solo suele ser necesario realizar la consulta una vez.

Para obtener el identificador de las variables de un control, se debe realizar la siguiente petición:

- **Método:** GET
- **URL:** /devices/*uuid*/controls/*control_uuid*/parameters
- **Uuid:** hace referencia al identificador de la instalación. Obtenido de la barra de navegación o de la petición API del punto 4.1
- **Control_uuid:** hace referencia al uuid del control del que se va a extraer la variable, obtenido de la barra de navegación o de la petición API del punto 4.2

Ejemplo:

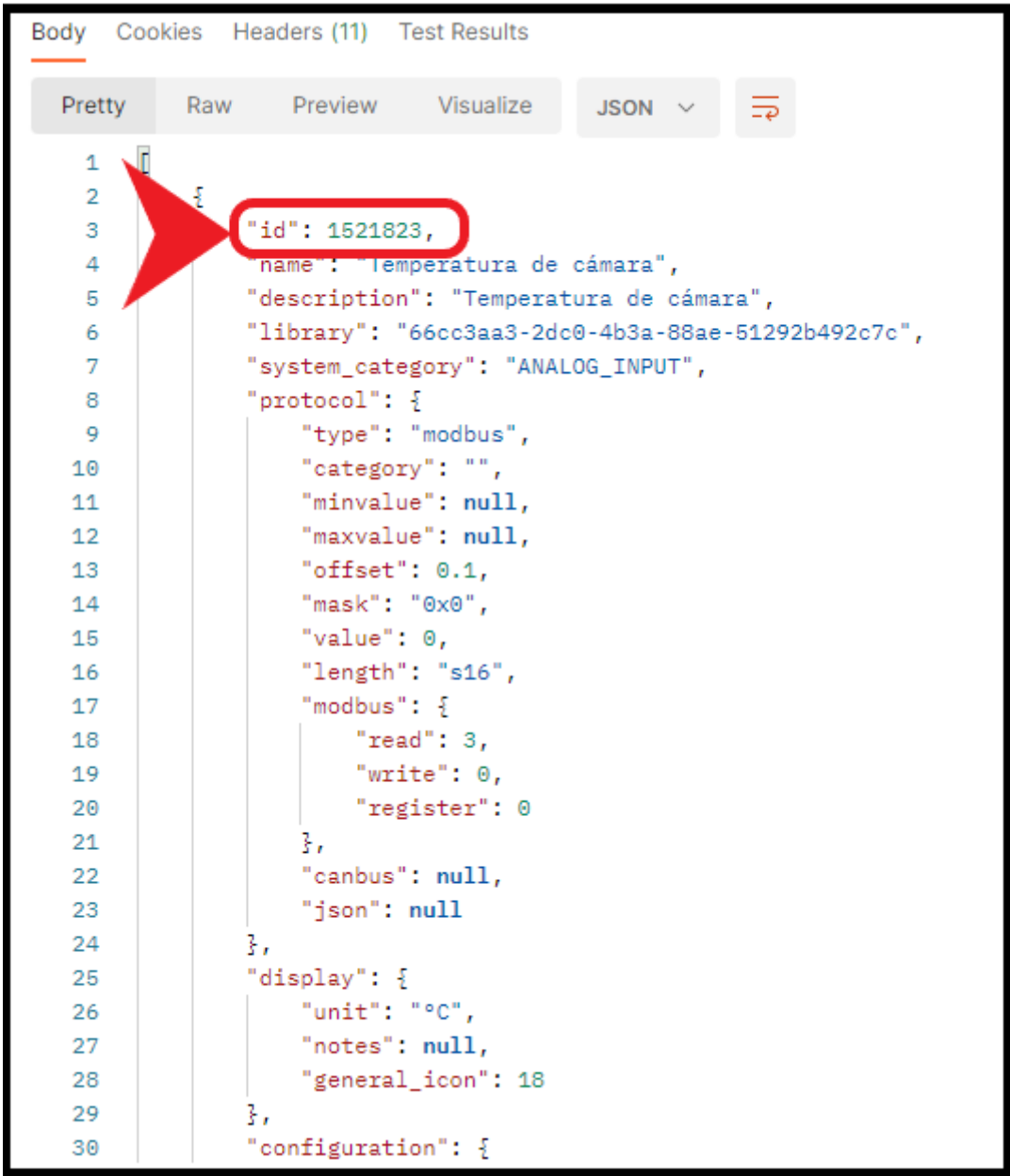
- Petición:



- Cabecera:

Params	Authorization	Headers (7)	Body	Pre-request Script	Tests	Settings
Headers Hide auto-generated headers						
		KEY	VALUE			
☒		Postman-Token	<calculated when request is sent>			
☒		Host	<calculated when request is sent>			
☒		User-Agent	PostmanRuntime/7.26.8			
☒		Accept	*/*			
☒		Accept-Encoding	gzip, deflate, br			
☒		Connection	keep-alive			
☒		Authorization	Basic P1dwr			
		Key	Value			

- Respuesta:



The screenshot shows the 'Body' tab of a web browser's developer tools. The JSON response is displayed in a 'Pretty' format. A red arrow points to the 'id' field, which is circled in red. The JSON structure is as follows:

```
1 {
2   "id": 1521823,
3   "name": "temperatura de cámara",
4   "description": "Temperatura de cámara",
5   "library": "66cc3aa3-2dc0-4b3a-88ae-51292b492c7c",
6   "system_category": "ANALOG_INPUT",
7   "protocol": {
8     "type": "modbus",
9     "category": "",
10    "minvalue": null,
11    "maxvalue": null,
12    "offset": 0.1,
13    "mask": "0x0",
14    "value": 0,
15    "length": "s16",
16    "modbus": {
17      "read": 3,
18      "write": 0,
19      "register": 0
20    },
21    "canbus": null,
22    "json": null
23  },
24  "display": {
25    "unit": "°C",
26    "notes": null,
27    "general_icon": 18
28  },
29  "configuration": {
30
```

5. Histórico de datos de una o varias variables

Para esta función disponemos de dos posibles peticiones: una sobre una gráfica creada en la **plataforma mykiconex** y otra en la que le indicamos en la propia petición los parámetros que deseamos consultar.

5.1 Petición personalizada de parámetros

La petición sería:

- **Método:** POST
- **URL:** /charts

Además de esto, se deben de introducir los parámetros de la consulta en el *body* de la petición siguiendo el siguiente formato:

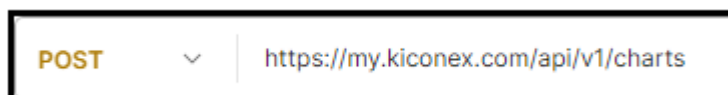
```

{
  "parameters": {
    "format": "raw",
    "from": "hh:mm_YYYYMMDD", //Fecha de inicio en formato UTC.
    "until": "hh:mm_YYYYMMDD ", // Fecha de fin en formato UTC.
    "interval": "60s", // Intervalo de tiempo la unidad s => segun
dos, m => minutos, h => horas.
    "sections": [
      { // Primera section.
        "series": [ // Diferentes series de la primera section
          { // Primera serie.
            "control": {
              "uuid": "cba97cd6-54d9-482e-b951-788e69b5a8b0"
            },
            "device": {
              "uuid": "9c80ef49-870e-4232-b481-f2b09c7d1c13"
            },
            "parameter":{
              "id": 4360637
            },
            "type": "digital" // Tipo de parametro digital o
analog, OPCIONAL
          }
        /*
        Si queremos más de una serie(variable), de la siguiente
        forma separadas por coma.
          {
            Segunda serie
          },
          {
            Tercera serie.
          }
        */
      ]
    }
    /*
    Sí queremos diferentes section, de la sigueinte forma
    separadas por coma. */
  ]
}

```

De esta manera, la petición quedaría como:

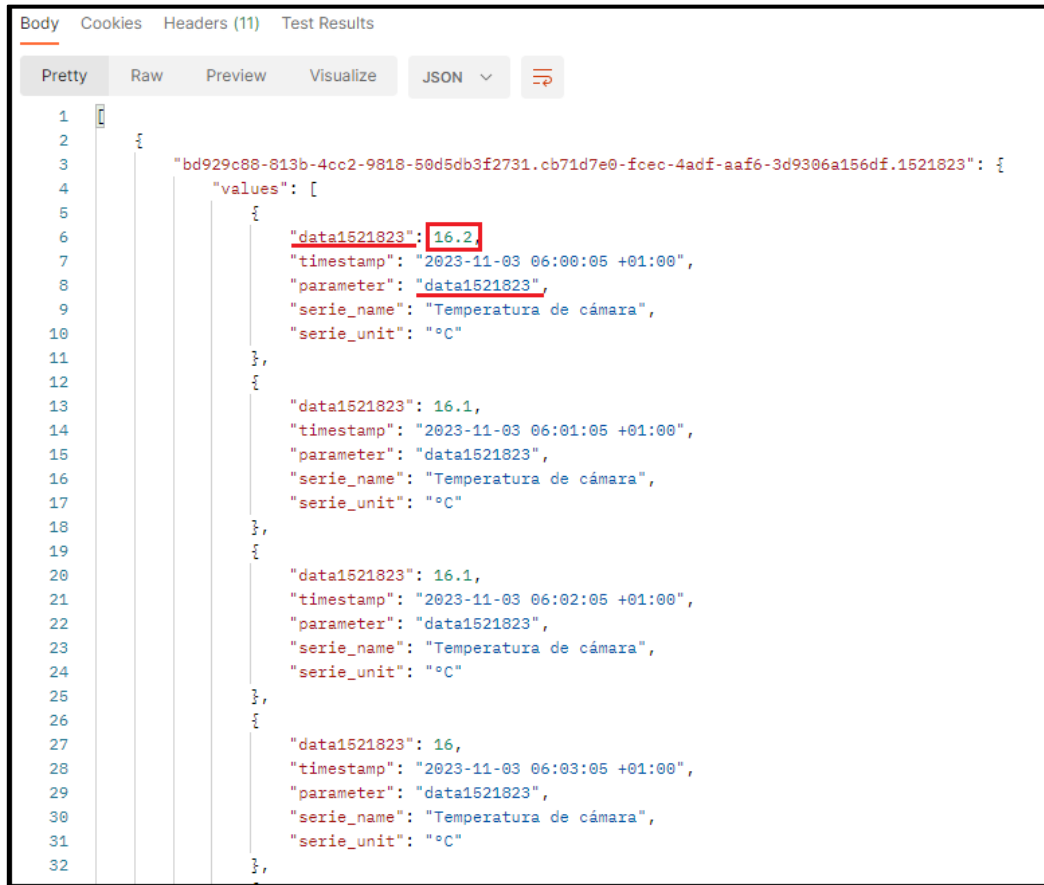
- Petición:



- Body:

```
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   "parameters": {
3     "format": "raw",
4     "from": "05:00_20231011", // Fecha de inicio.
5     "interval": "60s", // Intervalo de tiempo la unidad s=> segundos, m=> minutos, h=> horas.
6     "until": "06:00_20231011", // Fecha de fin.
7     "sections": [
8       // Primera section.
9       {
10        "series": [ // Diferentes series de la primera section
11          // Primera serie.
12          {
13            "control": {
14              "uuid": "cba97cd6-54d9-482e-b951-788e69b5a8b0" // control uuid del parametro.
15            },
16            "device": {
17              "uuid": "9c80ef49-870e-4232-b481-f2b09c7d1c13" // device uuid del parametro.
18            },
19            "parameter": {
20              "id": "4360637" // id de parametro
21            },
22            "type": "digital" // Tipo de parametro digital o analog, OPCIONAL
23          }
24        ]
25      }
26      // Si queremos mas de una serie, de la siguiente forma separadas por coma.
27      {
28        // Segunda serie.
29      },
30      {
31        // Tercera serie.
32      }
33    ]
34  }
35  // Si queremos diferentes section, de la siguiente forma separadas por coma.
36  {
37    "series": [
38      // Diferentes series de la primera section
39    ]
40  }
```

- Respuesta:



```
1 {
2   "bd929c88-813b-4cc2-9818-50d5db3f2731.cb71d7e0-fcec-4adf-aaf6-3d9306a156df.1521823": {
3     "values": [
4       {
5         "data1521823": 16.2,
6         "timestamp": "2023-11-03 06:00:05 +01:00",
7         "parameter": "data1521823",
8         "serie_name": "Temperatura de cámara",
9         "serie_unit": "°C"
10      },
11      {
12        "data1521823": 16.1,
13        "timestamp": "2023-11-03 06:01:05 +01:00",
14        "parameter": "data1521823",
15        "serie_name": "Temperatura de cámara",
16        "serie_unit": "°C"
17      },
18      {
19        "data1521823": 16.1,
20        "timestamp": "2023-11-03 06:02:05 +01:00",
21        "parameter": "data1521823",
22        "serie_name": "Temperatura de cámara",
23        "serie_unit": "°C"
24      },
25      {
26        "data1521823": 16,
27        "timestamp": "2023-11-03 06:03:05 +01:00",
28        "parameter": "data1521823",
29        "serie_name": "Temperatura de cámara",
30        "serie_unit": "°C"
31      }
32    ]
33  }
```

En la imagen podemos observar como recibimos un bloque *values* con tantos objetos de datos como se hayan pedido. Esta clave viene dentro de otra clave “*device_uuid.control_uuid.parameter_id*”, si la petición se hace sobre varios parámetros tendremos tantas claves como parámetros formadas igual y con un campo *values* en su interior.

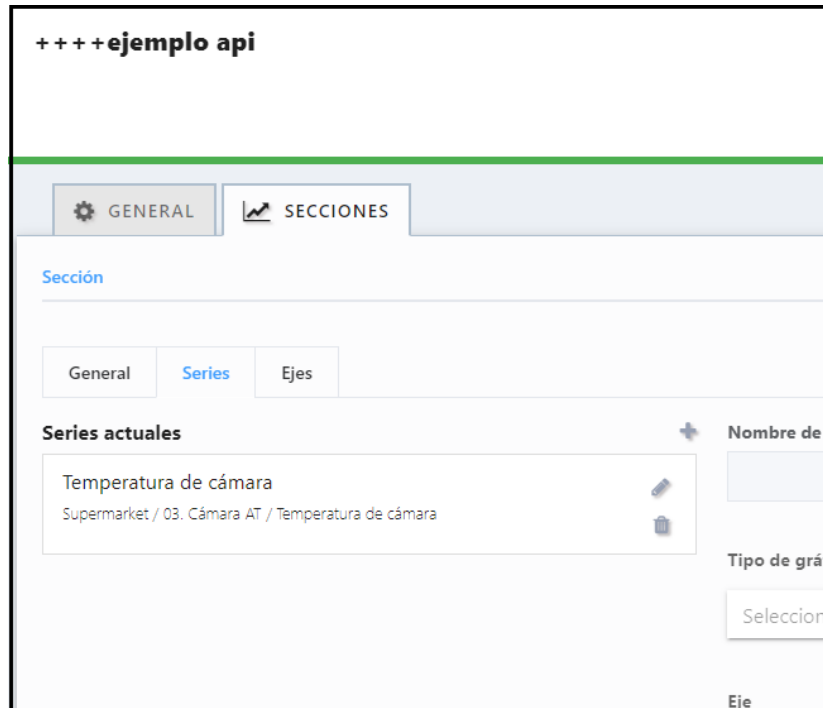
Para acceder al valor debemos de recoger el campo “*data+parameter_id*”

NOTA IMPORTANTE: A mayor cantidad de datos, más lenta será la consulta, y mayor será el tamaño del archivo obtenido. Lo óptimo es realizar una única petición con todos los parámetros deseados en el rango de tiempo a consultar.

Las fechas recibidas respetarán la zona horaria que el usuario configure en su perfil de la plataforma mykiconex.

5.2 Petición desde una gráfica creada

En este tipo de petición partimos de una gráfica creada en la plataforma mykiconex.



La petición sería:

- **Método:** GET
- **URL:** /devices/uuid/charts/chart_uuid
- **UUID:** hace referencia al identificador de la instalación. Obtenido de la barra de navegación o de la petición API que se realiza en el punto 4.1
- **Chart_uuid:** hace referencia al uuid de la gráfica de la que se van a extraer los datos. Obtenido de la barra de navegación.

Además de esto, se deben introducir unos parámetros de entrada, los cuales servirán para seleccionar el intervalo de recogida de datos, y el tiempo del muestreo:

- **from:** Fecha del primer dato a extraer en formato UTC. Su formato es HH:MM_YYYYMMDD
- **until:** Fecha del último dato a extraer en formato UTC. Su formato es HH:MM_YYYYMMDD

- **interval:** Lapso entre un dato y el siguiente. Este intervalo no puede ser menor que el establecido en la vista del propio control. Su formato es:
 - **NúmeroUnidad**, siendo *Unidad* la base temporal, y *Número* el intervalo en dicha unidad de tiempo. Las unidades serían:
 - **s** - Si se quiere establecer en segundos
 - **m** - Si se quiere establecer en minutos
 - **h** - Si se quiere establecer en horas
- **format:** Formato en el que recibiremos los datos. En API siempre usaremos *raw*.

De esta manera, la petición quedaría como:

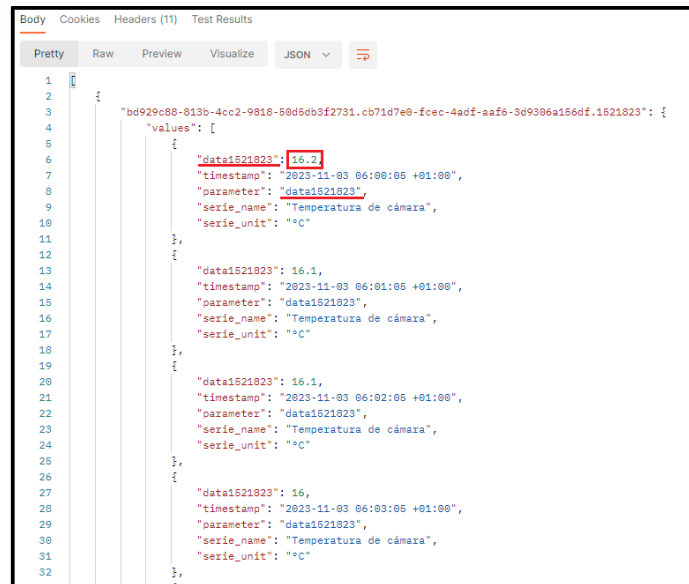
- Petición:



- Parámetros:

Params	Authorization	Headers (8)	Body	Pre-request Script	Tests	Settings
Query Params						
	Key		Value			
☒	interval		60s			
☒	from		05:00_20231103			
☒	until		06:00_20231103			
☒	format		raw			

- Respuesta:



```
1 {
2   {
3     "bd929c88-813b-4cc2-9818-50d5b3f2731.cb71d7e0-fcec-4adf-aaf6-3d9306a156df.1521823": {
4       "values": [
5         {
6           "data1521823": 16.2,
7           "timestamp": "2023-11-03 06:00:05 +01:00",
8           "parameter": "data1521823",
9           "serie_name": "Temperatura de cámara",
10          "serie_unit": "°C"
11        },
12        {
13          "data1521823": 16.1,
14          "timestamp": "2023-11-03 06:01:05 +01:00",
15          "parameter": "data1521823",
16          "serie_name": "Temperatura de cámara",
17          "serie_unit": "°C"
18        },
19        {
20          "data1521823": 16.1,
21          "timestamp": "2023-11-03 06:02:05 +01:00",
22          "parameter": "data1521823",
23          "serie_name": "Temperatura de cámara",
24          "serie_unit": "°C"
25        },
26        {
27          "data1521823": 16,
28          "timestamp": "2023-11-03 06:03:05 +01:00",
29          "parameter": "data1521823",
30          "serie_name": "Temperatura de cámara",
31          "serie_unit": "°C"
32        }
33      ]
34    }
35  }
```

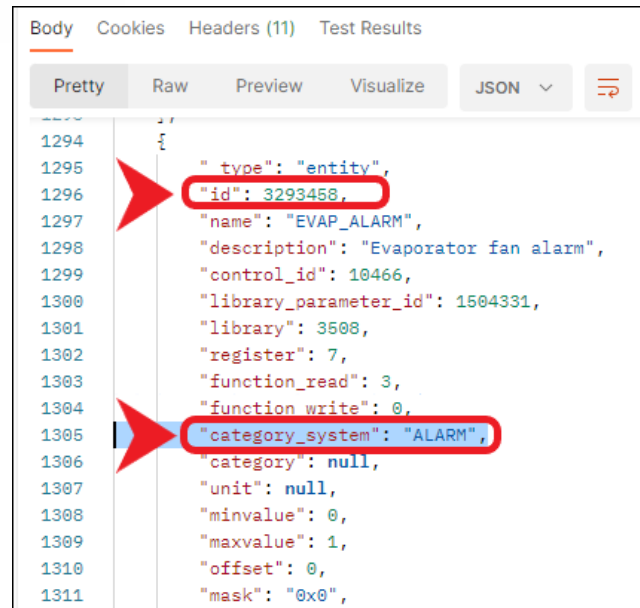
La respuesta sigue el mismo formato que el indicado en el punto 5.1.

NOTA IMPORTANTE: A mayor cantidad de datos, más lenta será la consulta, y mayor será el tamaño del archivo obtenido. Lo óptimo es realizar una única petición con todos los parámetros deseados en el rango de tiempo a consultar.

Las fechas recibidas respetarán la zona horaria que el usuario configure en su perfil de la plataforma mykiconex.

6. Consulta de variables de categoría Alarma

En la consulta que se realiza en el punto 4.1, se indentifica la categoría de cada variable a través del campo “*category_system*”. A través de dicho campo, se pueden identificar las variables que son de tipo alarma, ejemplo:



A las variables de tipo alarma, se le pueden realizar una serie de consultas que se detallan a continuación:

6.1 Consultar el total de alarmas activas

Con una sola consulta, se puede extraer una lista de alarmas activas en cualquiera de las instalaciones a las que se tiene acceso.

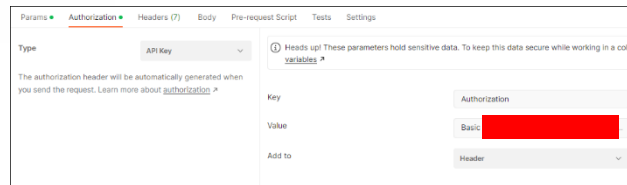
- **Método:** GET
- **URL:** /devices/alarms?state=1,2
- **State:** indica qué estado de alarmas se desea consultar:
 - 0=paradas
 - 1=activas
 - 2=silenciadas

De esta manera, la petición quedaría como:

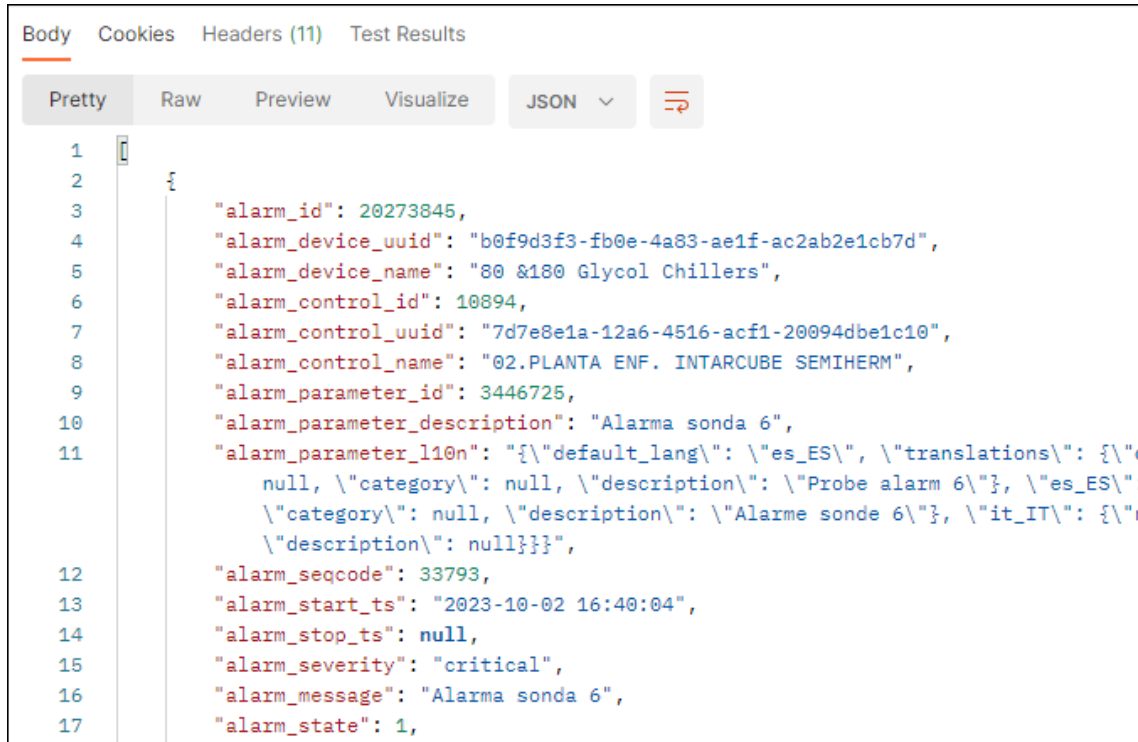
- Petición:



- Cabeceras:



- Respuesta, que indica ID del control afectado, uuid de la instalación a la que pertenece y más datos de cada alarma:



6.2 Consultar alarmas activas en una instalación

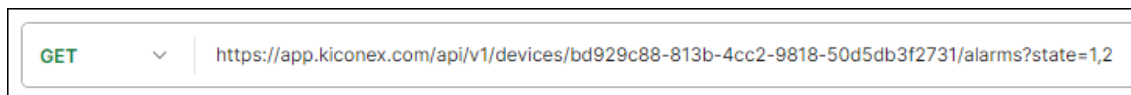
Pueden consultarse las alarmas activas en una instalación, añadiendo lo siguiente a la petición del punto **8.1**.

- **Método:** GET
- **URL:** /devices/*uuid*/alarms?state=1,2
- **State:** indica qué estado de alarmas se desea consultar:
 - 0=paradas
 - 1=activas
 - 2=silenciadas

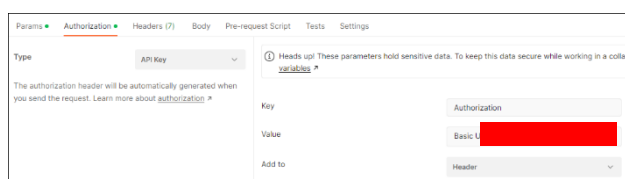
- **UUID:** hace referencia al identificador de la instalación. Obtenido de la barra de navegación o de la petición API que se realiza en el punto **4.1**

De esta manera, la petición quedaría como:

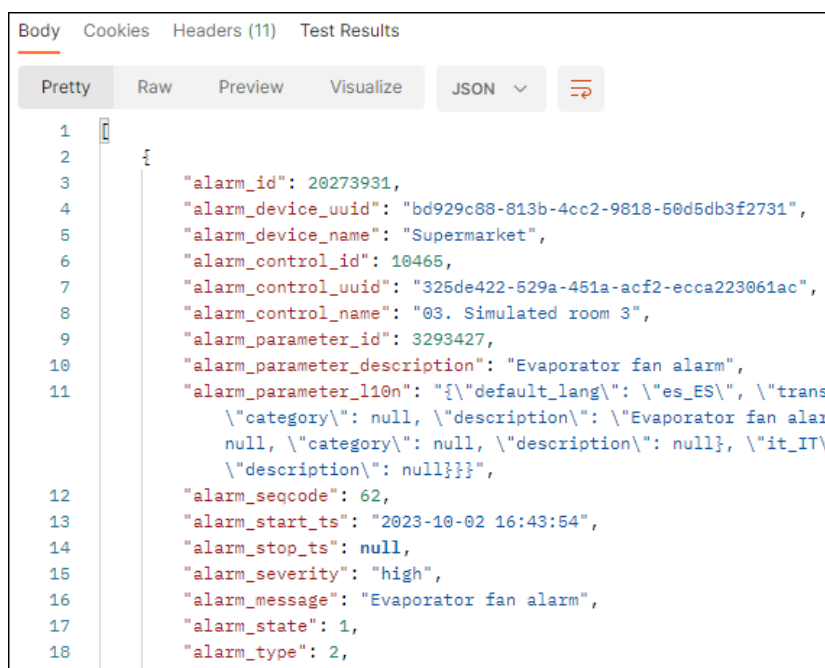
- Petición:



- Cabeceras:



- Respuesta, que indica ID del control afectado, uuid de la instalación a la que pertenece y más datos de cada alarma:



6.3 Consultar alarmas activas en un control

Se puede profundizar más en la consultar de alarmas y obtener aquellas que afectan a un control en concreto. Pueden consultarse las alarmas activas en una instalación, añadiendo lo siguiente a la petición del punto **8.2**.

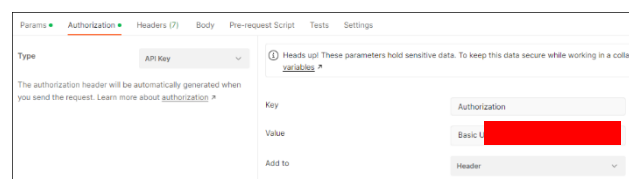
- **Método:** GET
- **URL:** /devices/*uuid*/controls/*uuid_control*?children=alarms&state=1,2
- **State:** indica qué estado de alarmas se desea consultar:
 - **0=paradas**
 - **1=activas**
 - **2=silenciadas**
- **UUID:** hace referencia al identificador de la instalación. Obtenido de la barra de navegación o de la petición API que se realiza en el punto **4.1**
- **Uuid_control:** hace referencia al identificador del control. Obtenido de la barra de navegación o de la petición API que se realiza en el punto **4.2**

De esta manera, la petición quedaría como:

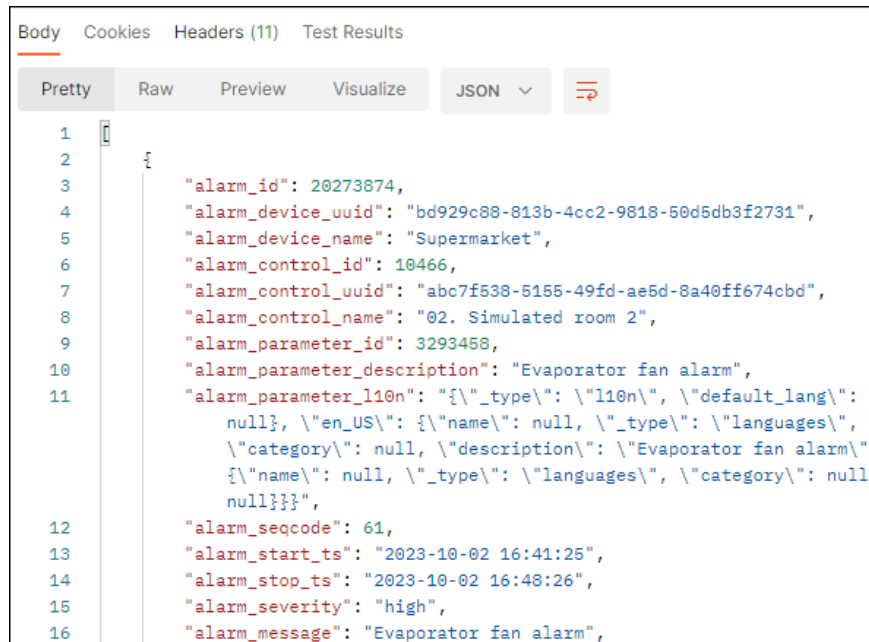
- Petición:



- Cabeceras:



- Respuesta, que indica UUID del control afectado, uuid de la instalación a la que pertenece y más datos de cada alarma:



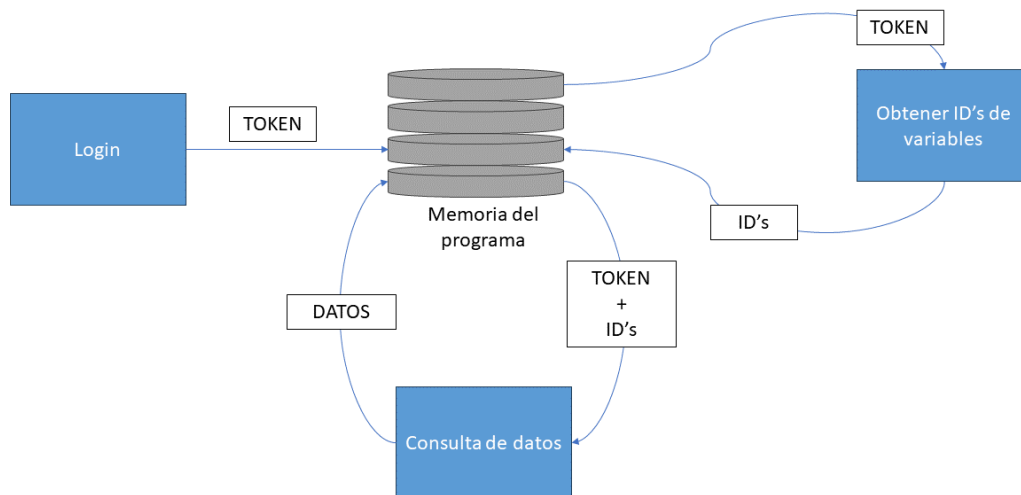
```
Body Cookies Headers (11) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "alarm_id": 20273874,
3   "alarm_device_uuid": "bd929c88-813b-4cc2-9818-50d5db3f2731",
4   "alarm_device_name": "Supermarket",
5   "alarm_control_id": 10466,
6   "alarm_control_uuid": "abc7f538-5155-49fd-ae5d-8a40ff674cbd",
7   "alarm_control_name": "02. Simulated room 2",
8   "alarm_parameter_id": 3293458,
9   "alarm_parameter_description": "Evaporator fan alarm",
10  "alarm_parameter_l10n": {"_type": "l10n", "default_lang": null, "en_US": {"name": null, "_type": "languages", "category": null, "description": "Evaporator fan alarm"}, {"name": null, "_type": "languages", "category": null, "description": null}},
11  "alarm_seqcode": 61,
12  "alarm_start_ts": "2023-10-02 16:41:25",
13  "alarm_stop_ts": "2023-10-02 16:48:26",
14  "alarm_severity": "high",
15  "alarm_message": "Evaporator fan alarm",
16 }
```

7. Proceso de consulta de datos

En los capítulos anteriores, se describía el procedimiento para realizar diferentes tipos de consultas:

- Realizar un inicio de sesión
- Obtener los ID's que interesan: ID de la instalación (UUID), del control (UUID) y de sus variables.
- Consultar históricos de datos

En el siguiente esquema se resumen el proceso que deben seguir las consultas realizadas:



En primer lugar, se realiza un programa que se ejecute una sola vez o con una periodicidad alta, por ejemplo 24 horas:

1. Se realiza el inicio de sesión descrito en el punto **3**, usando un usuario y una contraseña. El resultado del inicio de sesión es un token que se debe guardar para realizar sucesivas consultas.
2. Se realizan consultas, descritas en el punto **4**, usando el token generado anteriormente, para obtener los ID's de las variables que se desea consultar.

En segundo lugar, se elabora una rutina que cada cierto tiempo consulte el histórico de datos de cada variable, como se describe en el punto **5**. En esta consulta es necesario el token, el UUID de la instalación, el UUID del control y el ID de las variables cuyos datos se quieren conocer. La periodicidad de esta consulta depende mucho de las necesidades y objetivos de la aplicación.