

DESAFIO TECNICO

SERVIDOR PARA SENSORES DE TEMPERATURA

Se plantea la realización de un servidor para el almacenamiento y consulta de mediciones de sensores de temperatura.

Las consultas a la base de datos deben realizarse a través de APIs HTTP, por lo que se decide realizar de igual manera la carga de la telemetría, siempre teniendo en cuenta las condiciones establecidas en los diferentes puntos.

Para la realización del servidor HTTP se decide utilizar el lenguaje JavaScript con "NODEJS", utilizando visual studio code como entorno de desarrollo.

A su vez se utiliza la librería "express" que permite un desarrollo más ágil del servidor.

Los datos serán almacenados en una base de datos NoSQL. MongoDB es una herramienta de costo cero para el tamaño del desarrollo que se está implementando, muy fácil para trabajar con JavaScript, y que permite tipos de consulta que serán útiles en cuanto a las características especificadas.

SERVICIO DE TELEMETRIA - CARGA

Los sensores de temperatura deberán realizar la carga mediante APIs HTTP.

El servidor lo correremos localmente, accediendo desde <http://127.0.0.1:3333/TEMPERATURAS>

Los dispositivos realizaran una petición del tipo POST para cargar los valores de temperaturas medidos. Suponemos que el sensor tiene la programación como para leer los códigos de respuesta HTTP y así saber que no ha habido pérdida de información.

Los datos se envían en el body de la consulta, en formato json. Otra posibilidad seria enviarlos en un query string.

Por su parte el servidor, una vez que hay un método POST valido, corrobora la ID en la base de datos, si esta existe verifica que el alta del sensor haya sido realizada con mas de 1 minuto de antigüedad. De no ser así responde con un código de error y informa cuanto tiempo falta para comenzar a almacenar.

Las consultas devuelven un código de error y un mensaje de error en formato json.

Por ultimo se crea un índice TTL, que elimina el valor almacenado luego de transcurrido el tiempo especificado, en este caso 30 días.

POST:

url: <http://127.0.0.1:3333/TEMPERATURAS>

body:

```
{
  "Id": "<Id del sensor>",
  "Temp": "<Temperatura medida>"
}
```

Respuesta:

```
Code: 200 - La temperatura fue almacenada correctamente
      400 - Bad request
      401 - Dispositivo no encontrado
      405 - Falta dato de temperatura
      444 - Falta dato de Id
      446 - El sensor no esta en la base de datos
      447 - No paso el minuto necesario para poder comenzar a almacenar
```

SERVICIO DE TELEMETRIA - CONSULTA

La consulta de los datos de telemetría se realizará a través del método GET.

GET:

url: <http://127.0.0.1:3333/TEMPERATURAS>

body:

```
{
  "Id": "<Id del sensor>",
  "Tinicial": <Timestamp>,
  "Tfinal": <Timestamp>,
  "Opcion": <Numero>
}
```

Respuesta:

```
Code: 200 - OK
      400 - Bad request
      401 - Dispositivo no encontrado
      405 - Falta dato de temperatura
      444 - Falta dato de Id
      446 - El sensor no está en la base de datos
      447 - No paso el minuto necesario para poder comenzar a almacenar
      448 - Código de opción invalido
      449 - No hay datos de mediciones
```

“Tinicial” – “Tfinal”: Rango de tiempo en que se hará la búsqueda. Si no se proporcionan estos datos, no se filtrará por fecha.

“Opcion”: 1 - Promedio
 2 - MAXIMO
 3 - mínimo

Ejemplo de body:

```
{
  "Id": "AAA123",
  "Tinicial": 1659308870922,
  "Tfinal": 1659308870925,
  "Opcion": 1
}
```

SERVICIO DE CONFIGURACION

En el servicio de configuración, ingresaremos nuevos sensores con el método POST, consultaremos por un sensor con el método GET y borraremos sensores con el método DELETE.

Lo primero que hace el servidor es validar los caracteres de la Id, ya que solo están permitidas las letras mayúsculas y los números.

GET:

url: <http://127.0.0.1:3333/DISPOSITIVOS>

body:

```
{
  "Id": "<Id del sensor>"
}
```

Respuesta:

```
Code: 200 - OK
      400 - Bad request - Consulta invalida
      401 - Id invalido. Carácter no permitido
      402 - Sensor no encontrado
```

La respuesta será el Id del sensor y la fecha de alta del mismo.

POST:

url: <http://127.0.0.1:3333/DISPOSITIVOS>

body:

```
{
  "Id": "<Id del sensor>"
}
```

Form-data: file: <file csv>

Respuesta:

```
Code: 200 - OK
      400 - Bad request - Consulta invalida - Error lectura csv
      401 - Id invalido. Carácter no permitido
      444 - Sensor_Alta_Err0 - El sensor ya existe
```

Se puede cargar un sensor solo, a través del body de la request, o se puede cargar un archivo csv con las ids de los sensores, para darle de alta a un grupo.

DELETE:

url: <http://127.0.0.1:3333/DISPOSITIVOS>

body:

```
{  
  "Id": "<Id del sensor>"  
}
```

Form-data: file: <file csv>

Respuesta:

```
Code: 200 - OK  
400 - Bad request - Consulta invalida - Error lectura csv  
401 - Id invalido. Carácter no permitido  
444 - Sensor_Alta_Err0 - El sensor ya existe
```

Se puede borrar un sensor solo, a través del body de la request, o se puede cargar un archivo csv con las ids de los sensores, para darle de alta a un grupo.