

Ejercicio 1 – Introduccion a Ubidots

Tiempo estimado: 90 min

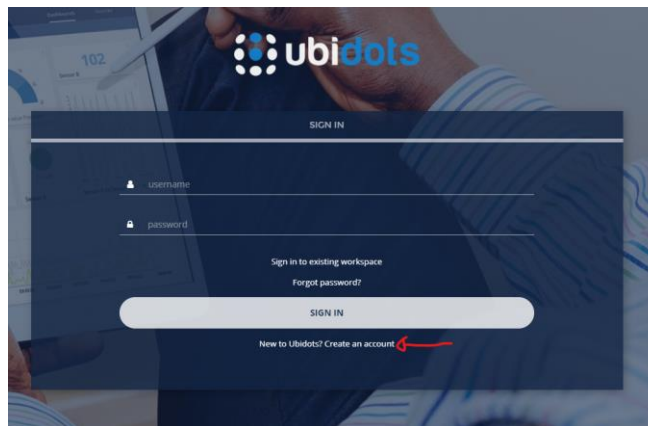
En esta practica realizara los pasos para crear un tablero que permita la representacion grafica de los valores obtenidos por un dispositivo IoT.

Para efectos de este ejercicio simularemos un dispositivo IoT instalado en una habitacion y que implementa la siguiente funcionalidad:

- Variables de Lectura
 - Temperatura de la habitacion
 - Porcentaje de Humedad relativa en la habitacion
- Variables de Interaccion
 - Encendido/Apagado de luz interior
 - Encendido/Apagado de luz exterior

Para completar esta practica realice los siguientes pasos:

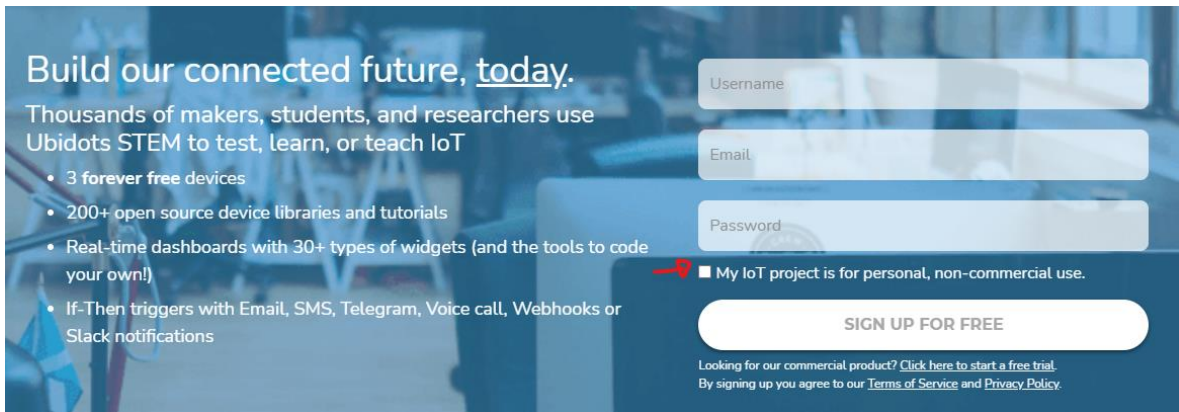
Paso 1: Ingrese a <https://stem.ubidots.com/accounts/signin/> y cree una cuenta



Paso 2: Seleccione la opcion: “Para uso educativo o personal”



Paso 3: Complete los datos del formulario y marque la casilla que indica que su proyecto no es para uso comercial.



Build our connected future, today.

Thousands of makers, students, and researchers use Ubidots STEM to test, learn, or teach IoT

- 3 forever free devices
- 200+ open source device libraries and tutorials
- Real-time dashboards with 30+ types of widgets (and the tools to code your own!)
- If-Then triggers with Email, SMS, Telegram, Voice call, Webhooks or Slack notifications

Username

Email

Password

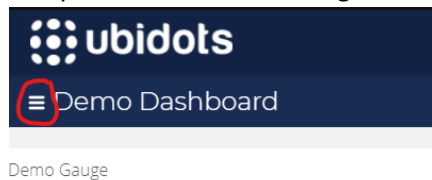
☒ My IoT project is for personal, non-commercial use.

SIGN UP FOR FREE

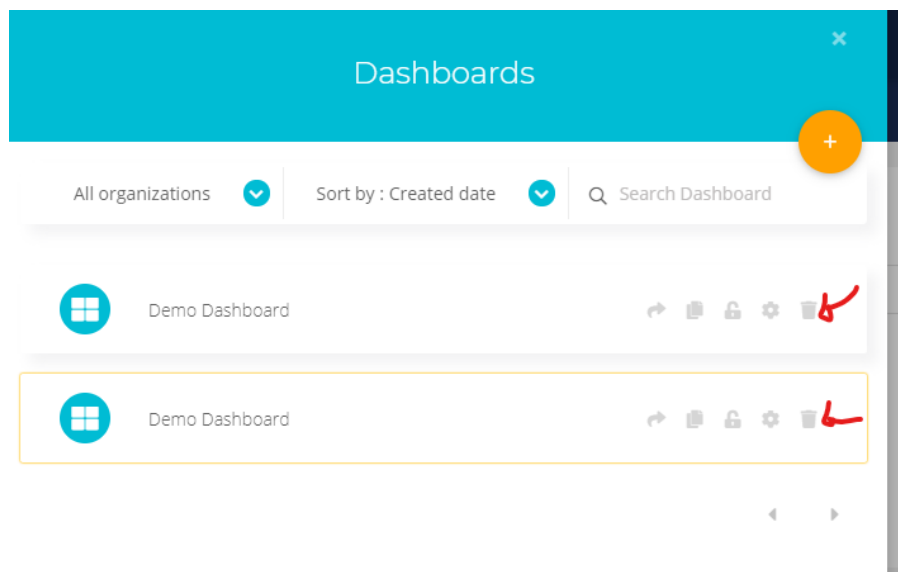
Looking for our commercial product? [Click here to start a free trial](#).
By signing up you agree to our [Terms of Service](#) and [Privacy Policy](#).

Paso 4: Limpie los Dashboards de muestra que crea el tutorial de bienvenida, realice las siguientes tareas:

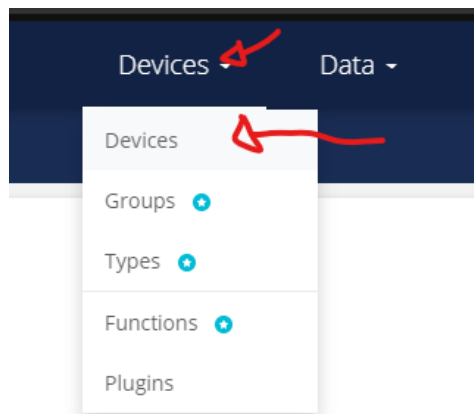
1. Ubique el menu de hamburguesa en la parte superior izquierda



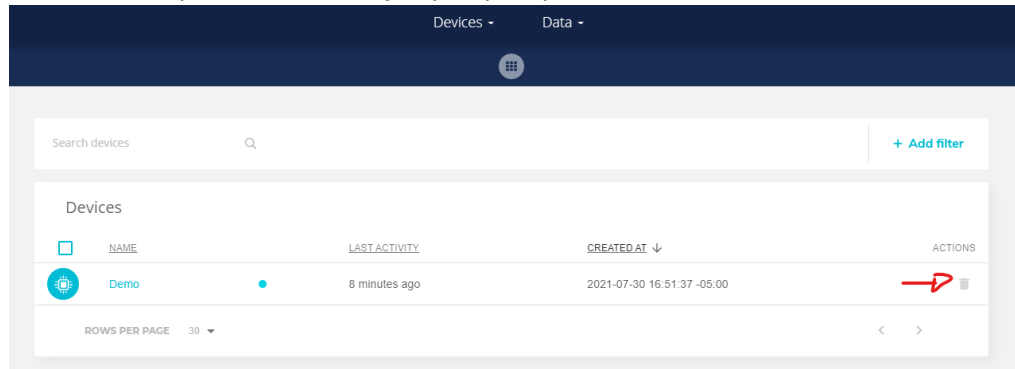
2. Al pulsar menu aparecen el o los registros correspondientes al proyecto de ejemplo, eliminelos



3. Limpie los dispositivos IoT de ejemplo, ubique el menu "Devices" y seleccione la opcion "Devices"

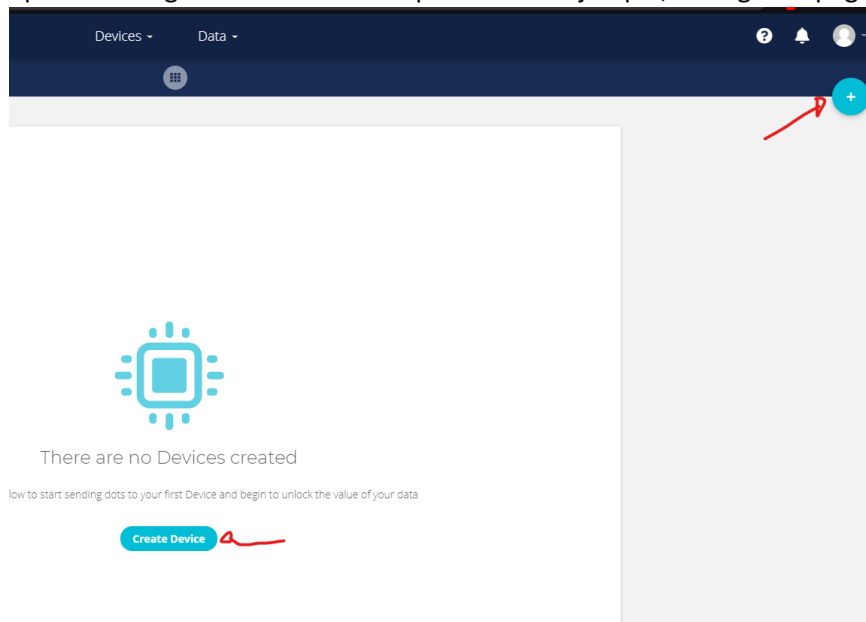


4. Elimine los dispositivos IoT de ejemplo que aparezcan



Paso 5: Cree el dispositivo IoT del ejercicio, realice las siguientes tareas:

1. Ubique cualquiera de los dos botones para agregar un dispositivo y presionelo (si no aparecen luego de eliminar los dispositivos de ejemplo, recargue la pagina)



2. Del menu “Add New Device”, seleccione la opcion: “Blank Device”

Add New Device

Create a new blank device shell, or automatically create new devices using the below libraries and documentation. The first time a dot is sent to Ubidots, a new Device is automatically created. ✕ CLOSE

CONNECTIVITY

Ethernet

Cellular

LoRaWAN

LTE-M

NB-IoT

Sigfox

WiFi

HARDWARE TYPE

Chips & Modules

Dev Kits

Gateways

Production Ready

INTEGRATION TYPE

Library

Plug-n-Play

Tutorial

Blank Device

Adafruit

Alorium Technology

Ambient Weather

Amplified Engineering

- Ingrese el valor: “hab-101” para el nombre del dispositivo y su etiqueta y pulse el boton verde para guardar, verifique que se ha agregado a la tabla de dispositivos.

<input type="checkbox"/>	NAME	LAST ACTIVITY	CREATED AT ↓	ACTIONS
	hab-101	No last activity	2021-07-30 17:17:09 -05:00	

ROWS PER PAGE 30

- Pulse sobre el nombre del dispositivo (letras en azul) para configurar las variables que controlará y vaya al siguiente paso.

Paso 6: Configure las variables para su dispositivo, realice las siguientes tareas:

- Una vez dentro del panel de configuracion del dispositivo pulse en la opcion: “Add Varibale” y luego seleccione la opcion: “Raw”

hab-101

Description

Change description

API Label ⓘ

hab-101

ID ⓘ

61047a650310a001d4442e64

Token

.....

Tags

Add new tag

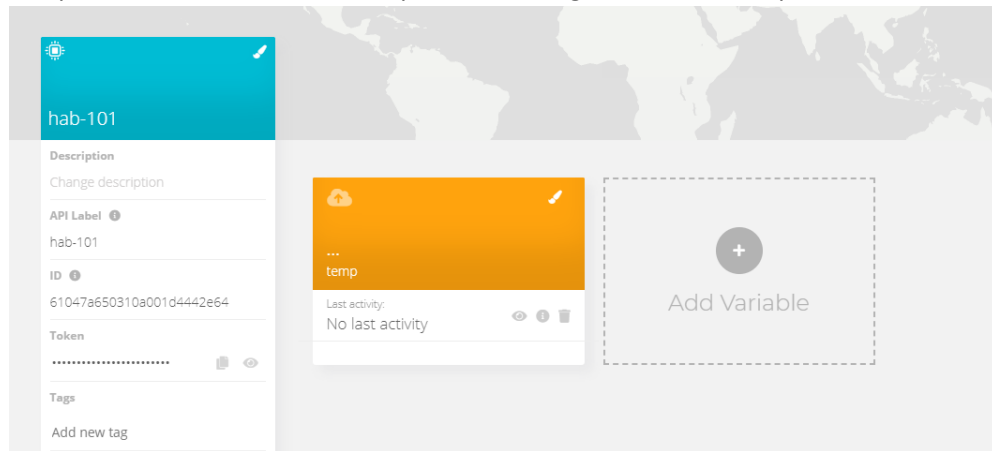
Last activity

Add Variable

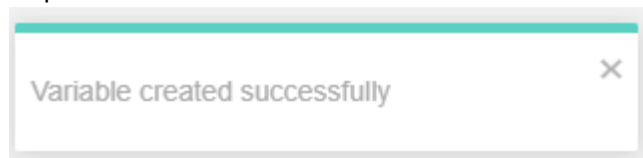
Raw

Synthetic

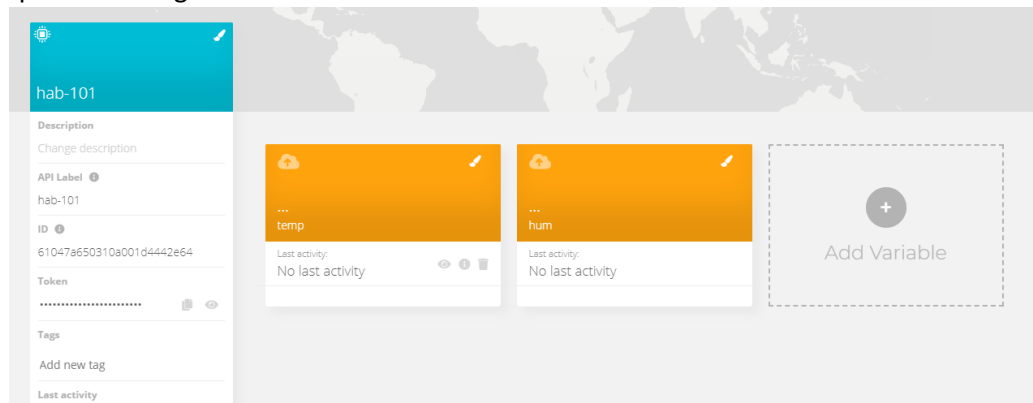
2. Asigne el nombre: “temp” a la primera variable, esta representara el valor de la temperatura de la habitacion, el panel de configuracion debera quedar asi:



3. Asegurese de vizualizar el mensaje de confirmacion que indica que la variable ha sido creada, aparece en la parte inferior izquierda cada vez que crea una variable para el dispositivo:



4. Repita el paso 2 para crear la variable que representa la humedad relativa de la habitacion, a esta la denominaremos: “hum”, el panel de configuracion del dispositivo queda de la siguiente manera:

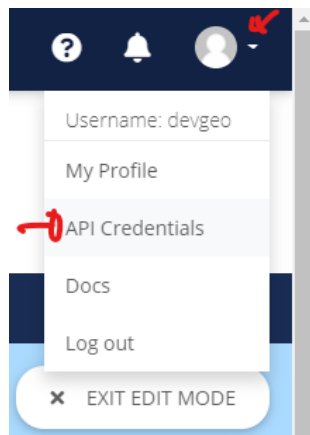


Paso 7: Pruebe la conexión con el API de ubidots y verifique que el dispositivo creado recibe las variables correctamente, realice las siguientes tareas:

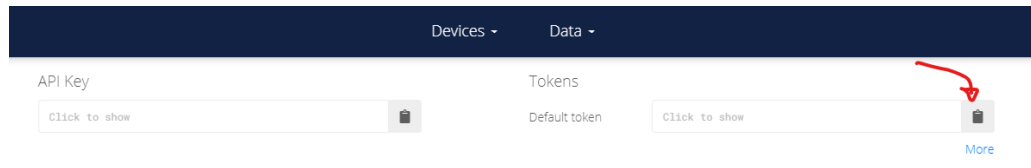
1. Para probar la conexión con el API de ubidots necesitaremos usar Postman, un cliente de peticiones HTTP REST que nos ayudara a verificar que nuestro dispositivo virtual creado en ubidots esta recibiendo correctamente las variables, en caso de no tenerlo instalado puede desacargarlo de la siguiente liga:

<https://www.postman.com/downloads/>

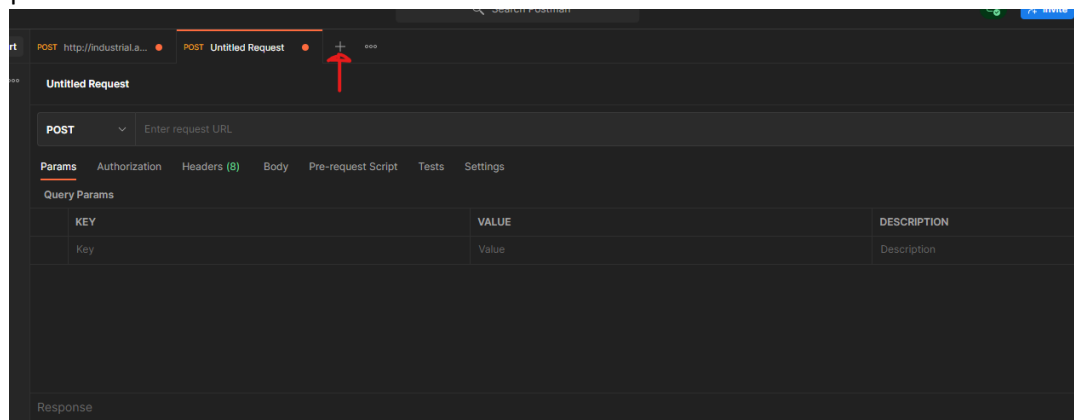
2. Para conectarte con el API de ubidots, necesitaras un token de acceso, para obtenerlo ve a la parte superior derecha de la pantalla, donde aparece tu avatar, pulsa en la lista desplegable y selecciona la opcion: “API Credentials”



3. Copia el Default Token de tu cuenta y guardalo en un lugar seguro, lo ocuparemos mas adelante.



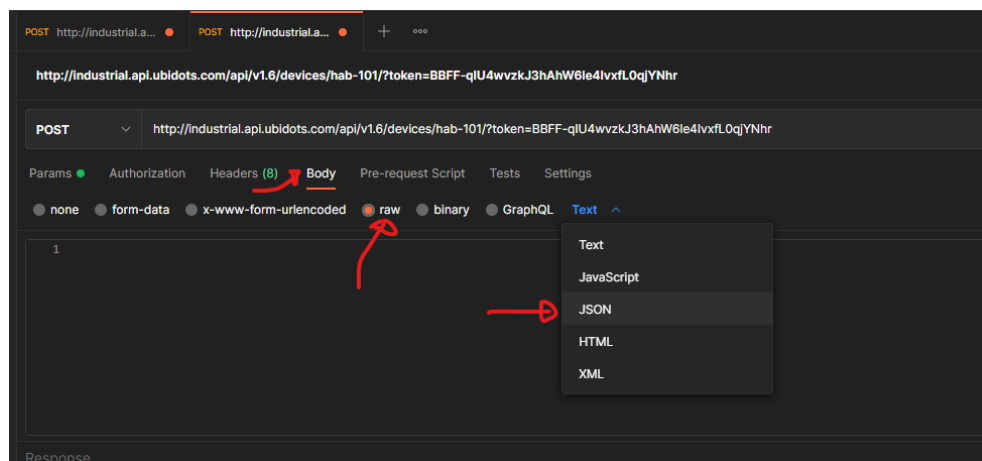
4. En postman da click en el icono de “mas”, sobe el area de trabajo, para crear una nueva peticion HTTP:



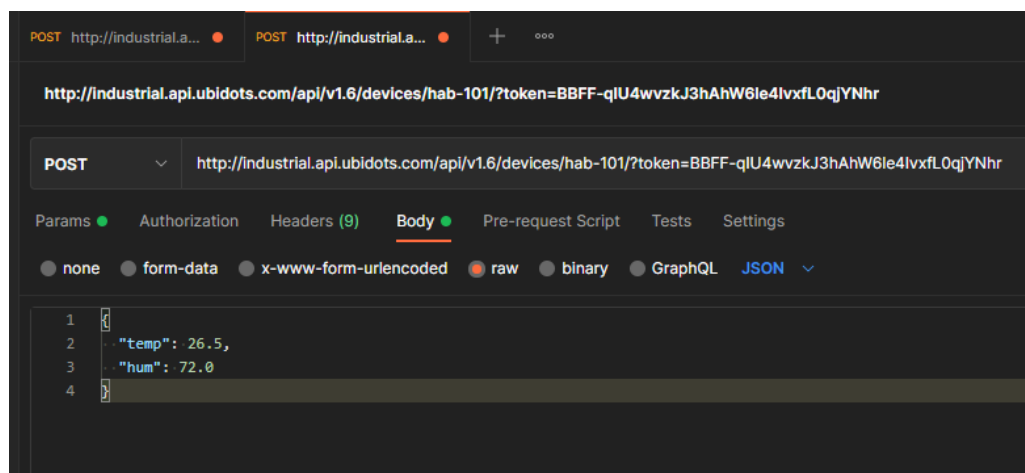
5. Copia el siguiente url, que es el endpoint del API de ubidots para conectar con el dispositivo: <http://industrial.api.ubidots.com/api/v1.6/devices/>, seguido de eso tendras que agregar el nombre del dispositivo y **el Default Token asociado a tu cuenta**, la url de tu endpoint deberia verse con algo parecido a esto:

<http://industrial.api.ubidots.com/api/v1.6/devices/hab-101/?token=BBFF-glU4wvzkJ3hAhWPQ54lvxfL0qjYNhr>

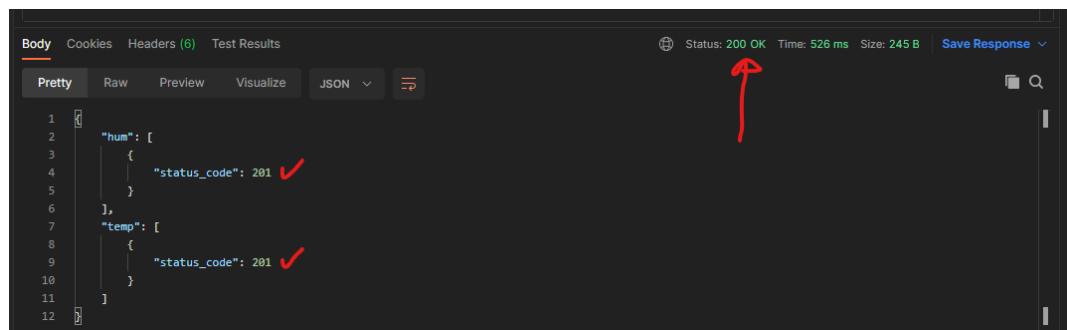
6. Como el dispositivo maneja dos variables, tenemos que crear una data en formato JSON con los valores de esas variables, para enviarlas al API de ubidots, para eso dentro de la pestaña que creaste en postman, selecciona la opcion: “Body”, a continuacion la opcion: “raw” y finalmente la opcion: “JSON”, con esto podremos crear la data de prueba

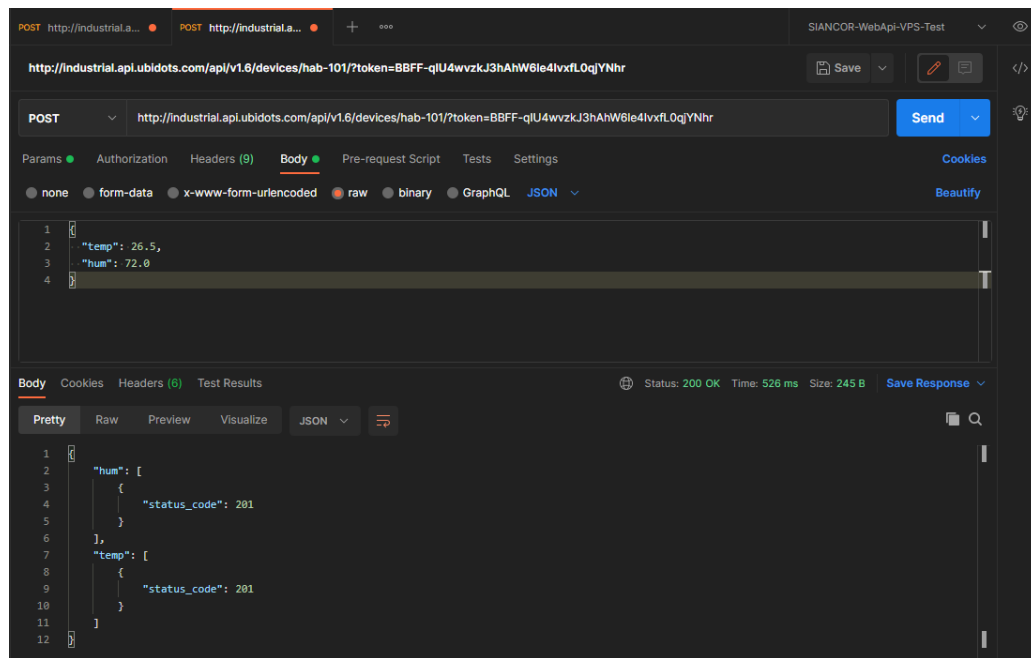


7. A continuacion estableceremos los valores de la temperatura y la humedad relativa, asignaremos 26.5 °c para temperatura y una humedad relativa del 72%, estos valores transformados a JSON deberian verse de la siguiente manera en postman:

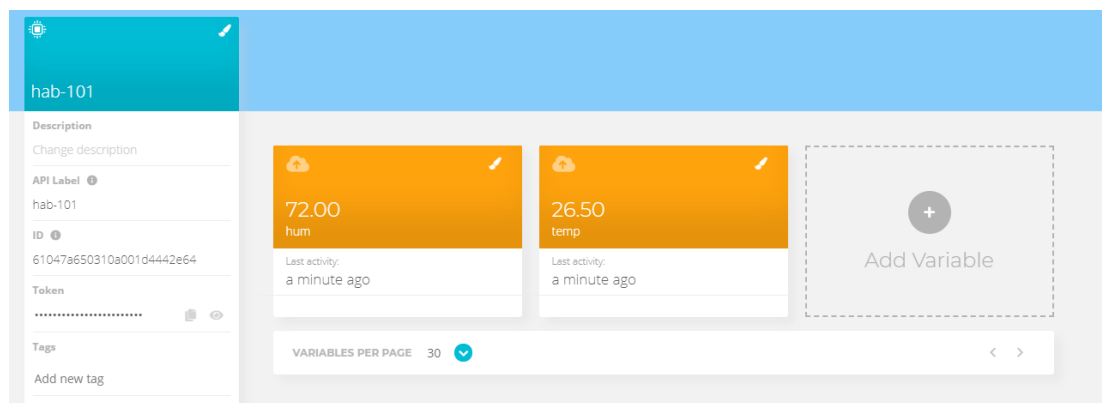


8. Presiona el boton send del postman para enviar la informacion al api de ubidots, deberias obtener un codigo de respuesta 200 OK del servidor y un payload del API indicando que cada valor de la variable fue creado correctamente en ubidots:

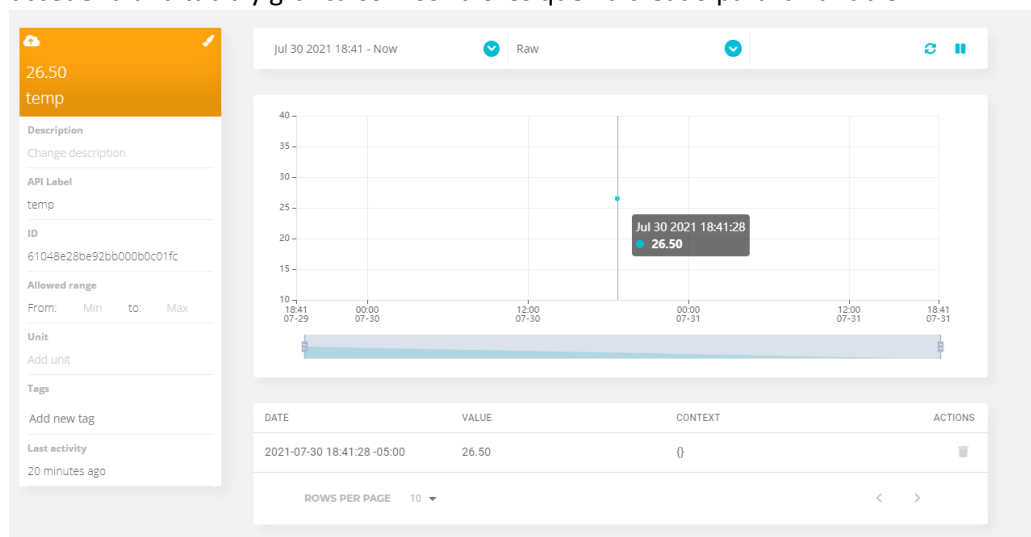




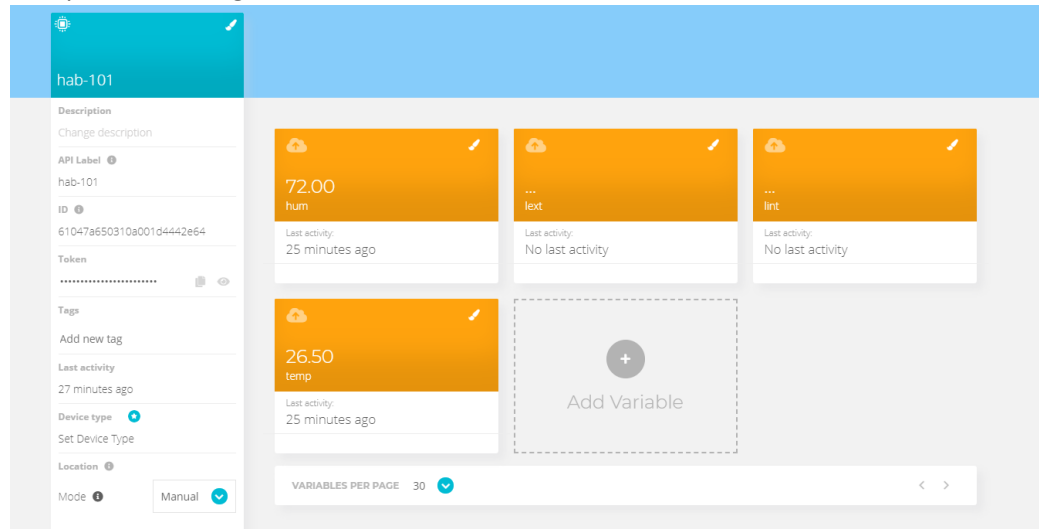
- En ubidots actualiza la pagina del panel de control de variables del dispositivo, ahora deberias poder ver los valores que has enviado mediante postman



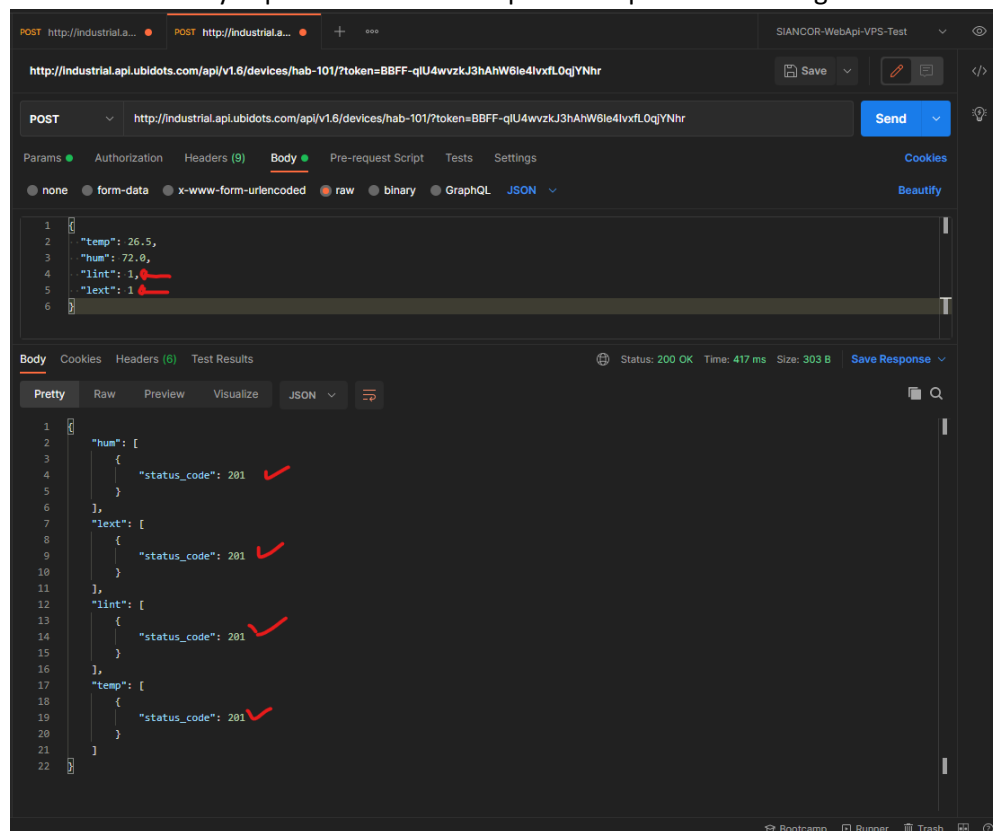
- Si da click sobre el encabezado de cualquiera de las tarjetas (la parte naranja) podra acceder a una tabla y grafica con los valores que ha creado para la variable



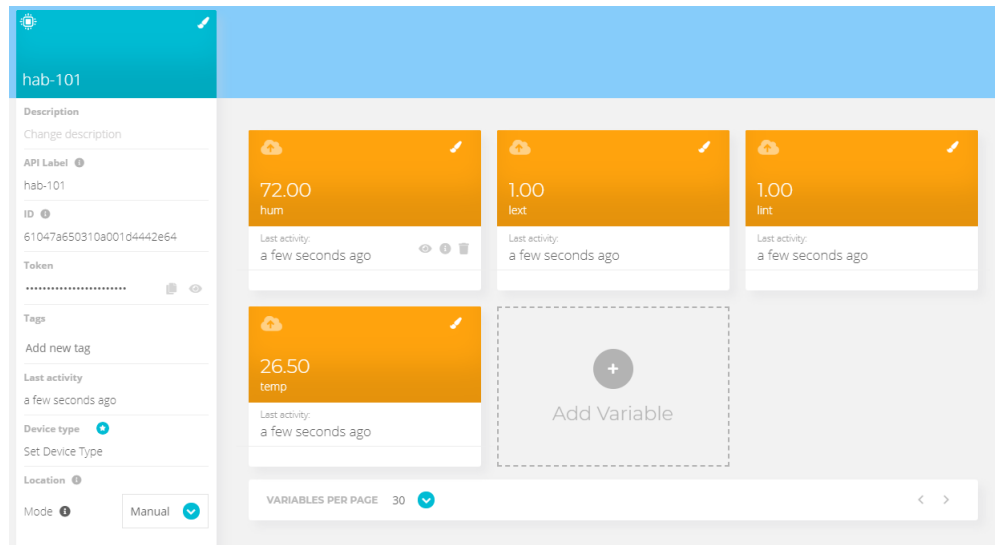
11. Hasta aquí hemos configurado las variables del dispositivo (temperatura y humedad), sin embargo el dispositivo tambien debe ser capaz de interactuar con las luces de la habitacion (las interiores y las exteriores), para ello configuraremos dos variables mas "lext" para luces del exterior y "lint" para luces del interior.
12. Repita el paso 6.2 para agregar las variables, el panel de control de las variables deberia quedar de la siguiente manera



13. Modificaremos la data del request de postman incorporando los valores de las nuevas variables, para tales efectos asumiremos que los valores de las luces seran binarios, donde 0 corresponde a la accion de apagar las luces y 1 a la accion de encenderlas, la data modificada y la prueba de conexion postman quedaria de la siguiente manera:

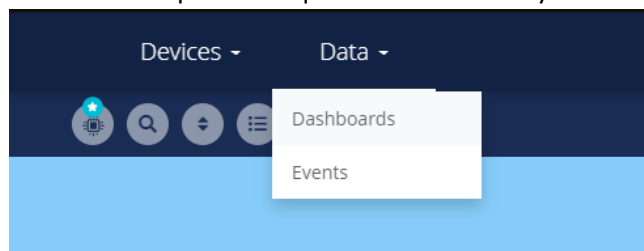


14. Actualice el panel de variables de ubidots, debería ver los cambios reflejados en las nuevas variables:

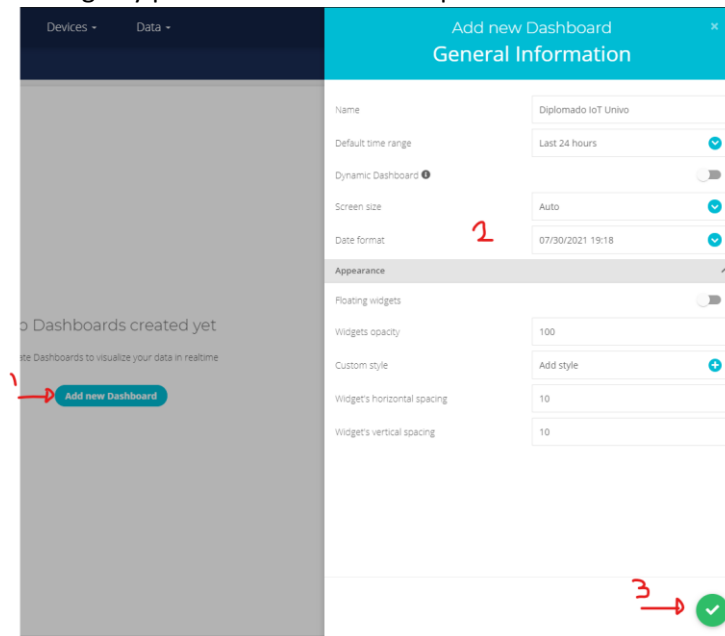


Paso 8: Ahora ya que tenemos nuestro dispositivo configurado y conectado con el API, crearemos un Dashboard para poder representar de manera grafica la informacion, complete las siguientes tareas:

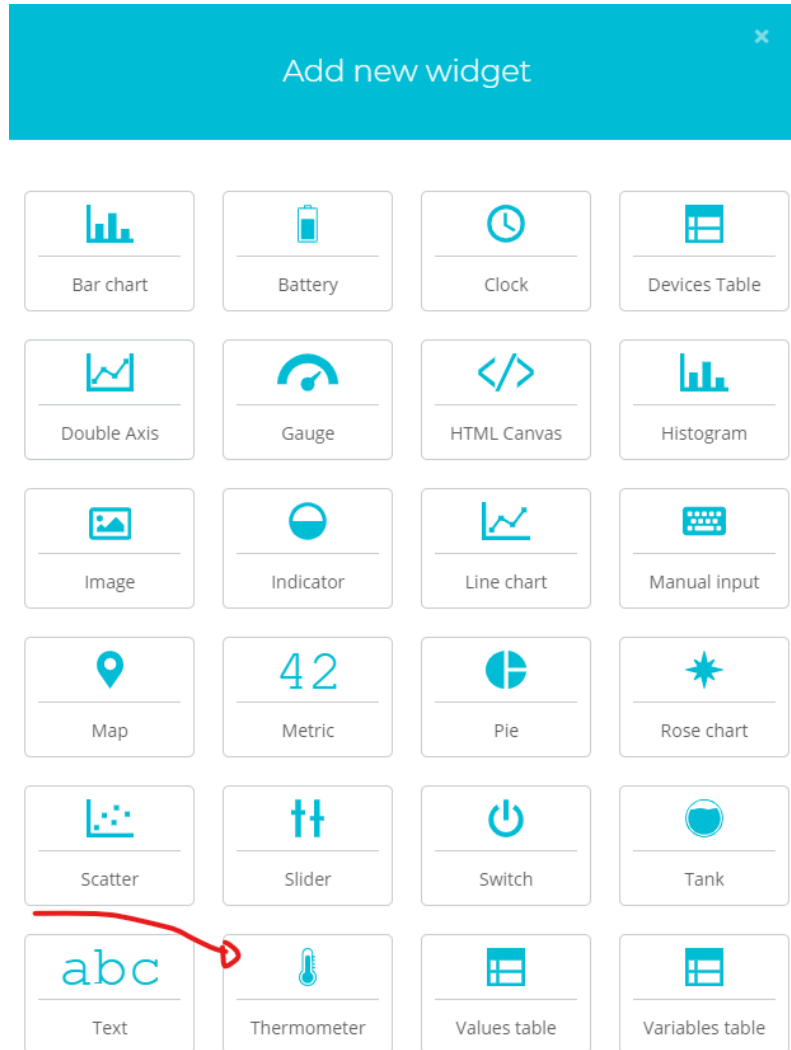
1. En el menu superior ubique el menu “Data” y seleccione la opcion “Dashboards”



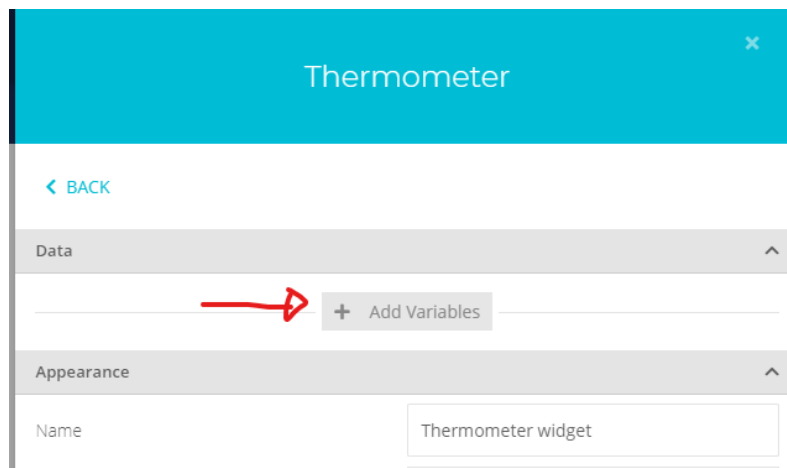
2. Seleccione la opcion: “Add new Dashboard”, establezca los valores correspondientes a la imagen y pulse en el boton verde para crearlo



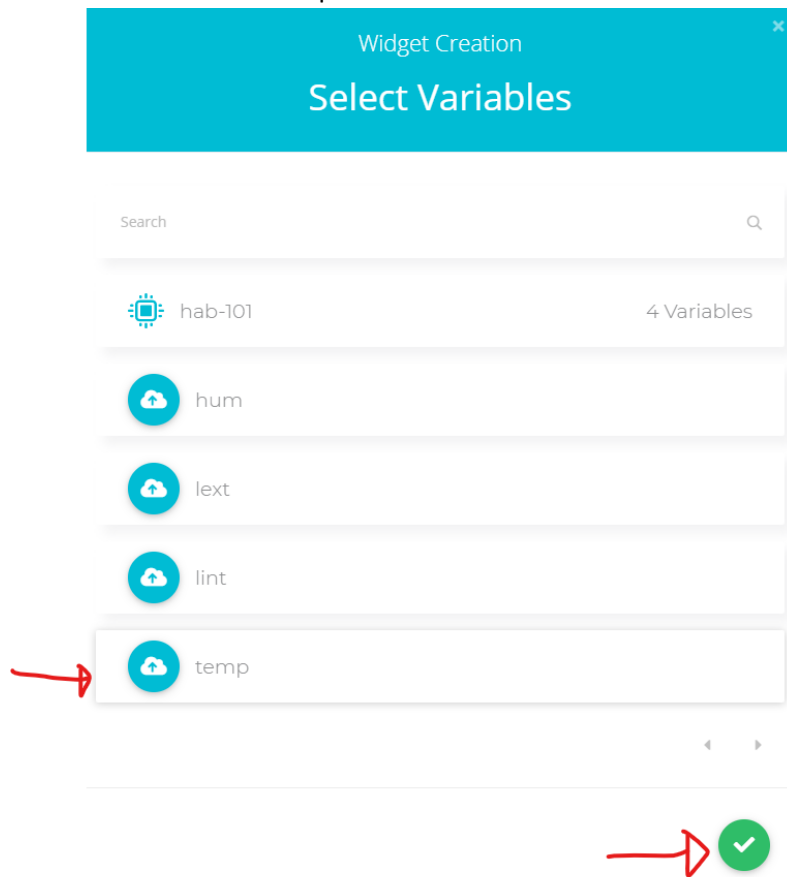
3. Una vez creado el Dashboard agregaremos algunos widgets para representar la información del dispositivo, comenzaremos con la variable de temperatura, de click en opción: “Add new Widget” y seleccione el widget: “Thermometer”



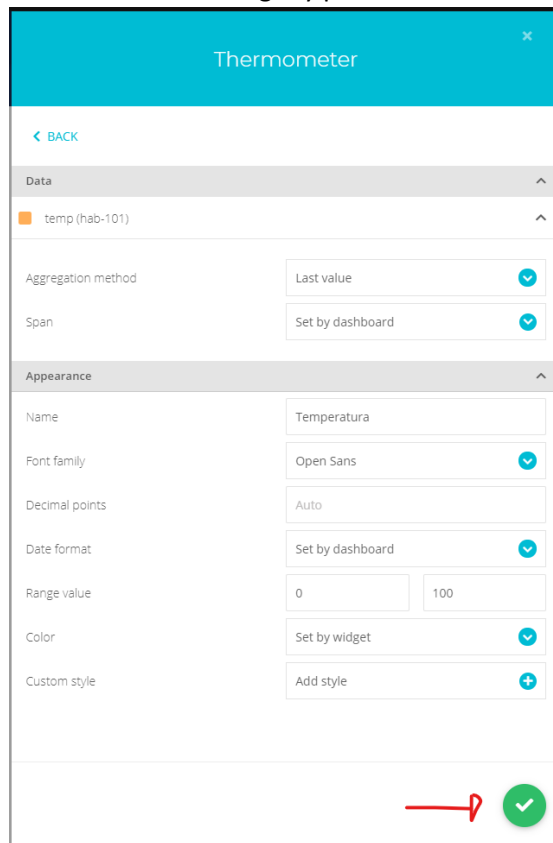
4. Configure la variable asociada al widget, bajo el menu “Data”, presione la opción: “Add Variables”



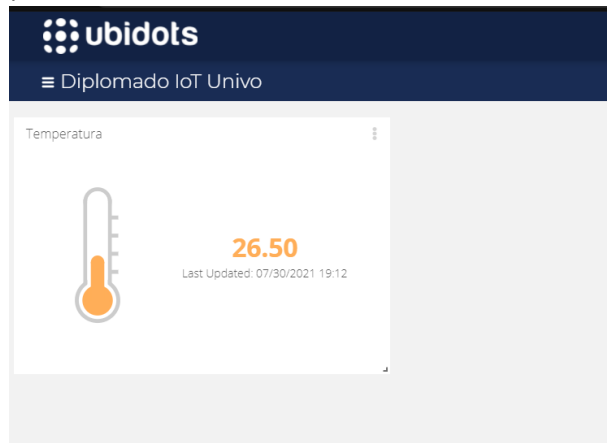
5. Seleccione el dispositivo creado para este ejercicio (hab-101) y a continuacion, seleccione la variable temp



6. De un nombre al widget y presione el boton verde



7. Notara que en el Dashboard el widget ha sido agregado y toma el ultimo valor ingresado para la variable seleccionada:



8. Ahora agregaremos un widget que nos permita representar de manera grafica una serie de valores para la temperatura, ubique el boton con el simbolo de + en la parte superior derecha de la pantalla y presionelo, seleccione el widget: "Line chart", repita el paso 5 para seleccionar la variable de temperatura del dispositivo y asegurese de que los valores de su widget correspondan con los de las siguientes imágenes:

The screenshot shows the configuration interface for a line chart widget. It is divided into two main sections: "Data" and "Appearance".

Data Section:

- temp (hab-101)**: The selected variable.
- Aggregation method**: Last value (selected with a blue checkmark).
- Span**: Set by dashboard (selected with a blue checkmark).
- Type**: Line (selected with a blue checkmark).
- Y-Axis**: Default Y-axis (selected with a blue checkmark).

Appearance Section:

- Name**: Grafica de Temperatura
- Decimal points**: Auto
- Show legend**: Toggle switch is turned off.
- Date format**: Jul 30 2021 19:53 (selected with a blue checkmark).
- Display X-Axis data zoom**: Toggle switch is turned on.
- X-axis label**: None
- Custom style**: Add style (selected with a blue plus icon).

Default Y-axis

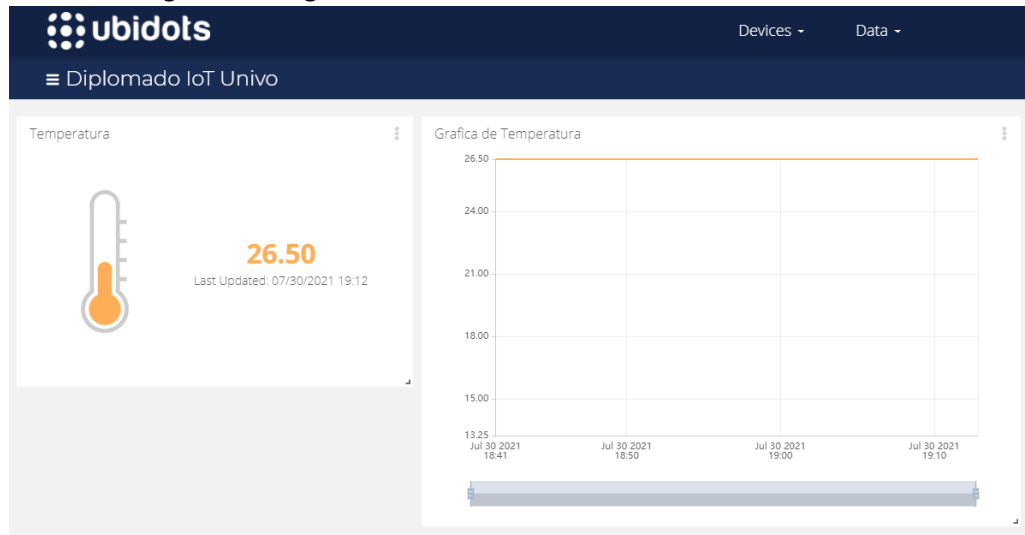
Y-axis name

Position

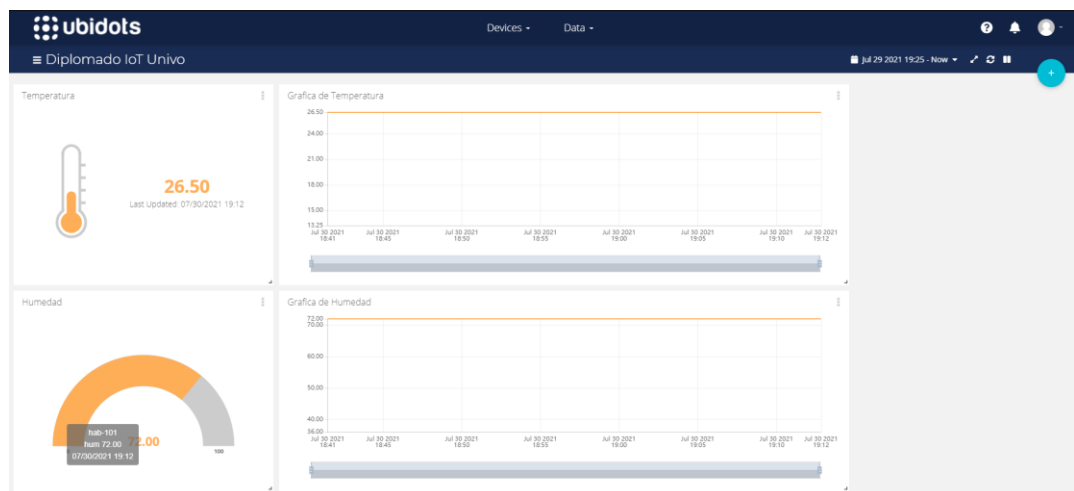
Y-axis range

Use SI prefix ☐

9. Notara que la grafica toma el ultimo valor de temperatura, el Dashboard, deberia verse similar a la siguiente imagen:



10. Repita los pasos anteriores para agregar la representacion grafica de la varibale de humedad, seleccione el widget: "Gauge" para representar el valor actual de la humedad y el widget: "Line chart" para los valores historicos, ordene el Dashboard como estime conveniente, al final deberia tener un panel similar al siguiente:



11. Ahora agregue un widget para representar el estado de las luces de la habitacion, vaya al boton con signo de + en la parte superior derecha y seleccin del menu la opcion:

“Switch”, ingrese los valores que aparecen en la siguiente imagen y presione el boton verde para guardar

Switch

[< BACK](#)

Data

l1nt (hab-101)

Minimum Value

0

Maximum Value

1

Off message

Luces Apagadas

On message

Luces Encendidas

+ Add Variables

Appearance

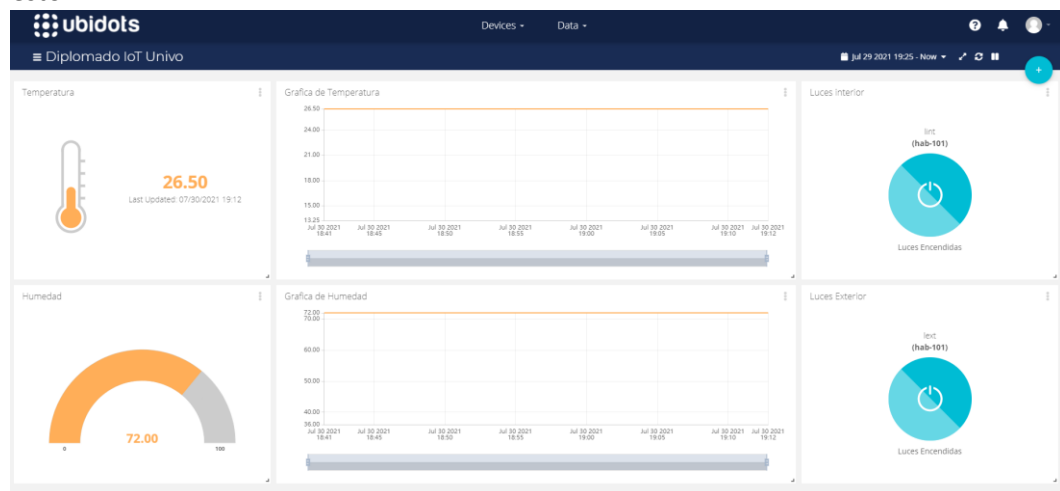
Name

Luces interior

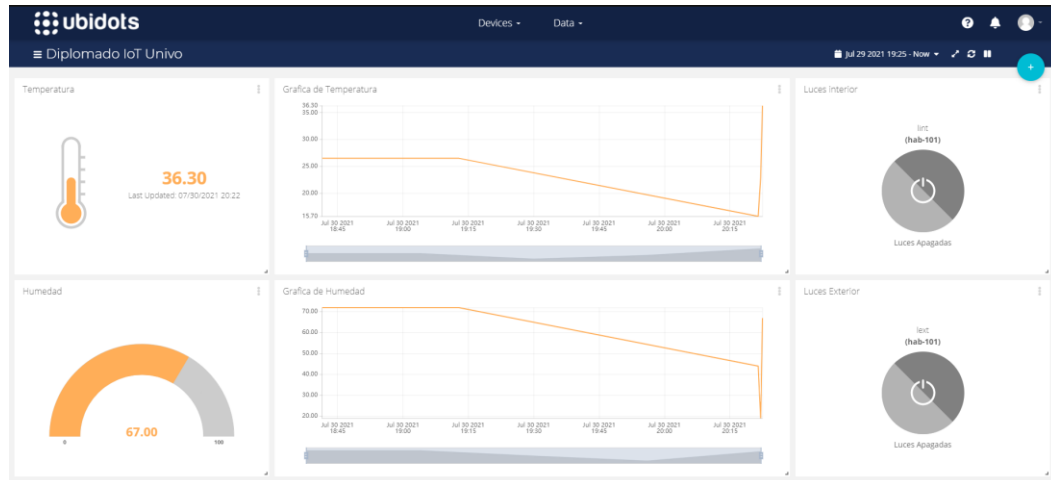
Custom style

Add style

12. Repita el paso anterior para agregar el control de las luces exteriores y cambie los valores de las etiquetas como corresponda, al final el Dashboard debería lucir similar a esto:



13. Haga dos o tres peticiones desde postman con diferentes valores para cada variable y asegurese de que los cambios se vean reflejados en el Dashboard, despues de las pruebas el Dashboard deberia verse parecido a esto:



14. Hasta este punto su Dashboard esta listo para representar datos de un dispositivo virtual (hab-101), es hora de crear un emulador de software que alimente de datos al dispositivo virtual para que la informacion se refleje en linea en el tablero, eso lo realizaremos en los siguientes pasos.

Paso 9: Cree un emulador que simule el comportamiento del dispositivo IoT del ejercicio, para esto complete las siguientes tareas:

1. Crea una carpeta de trabajo y abrela con VS Code
2. Dentro de la carpeta de trabajo crea el archivo index.html con el siguiente codigo:

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Emulador IoT - UNIVO </h1>
  <h3>Lectura de Variables</h3>
  <button id="iniciar">Inicar</button>
  <button id="detener">Detener</button>

  <br><br>
  <h3>Acciones de la Habitacion</h3>
  <button id="luzInt">On/Off - Luces Interiores</button>
  <button id="luzExt">On/Off - Luces Exteriores</button>

  <script src="scripts.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
</body>
</html>
```

3. Dentro de la carpeta de trabajo crea el archivo scripts.js con el siguiente codigo:


```

//Constantes de acceso a elementos del documento HTML
const iniciar = document.getElementById('iniciar');
const detener = document.getElementById('detener');
const luzInt = document.getElementById('luzInt');
const luzExt = document.getElementById('luzExt');

//Constante para url API ubidots CAMBIE LOS DATOS POR SU TOKEN PERSONAL
const apiURL =
'http://industrial.api.ubidots.com/api/v1.6/devices/hab-101/?token=BBFF-qIU4wvzkJ3hAhW6le4Ivxfl0qjYNhr';
const tasaRequest = 2500;

//Variables para manipulacion del emulador
var data;
var edoLuzInt = 0;
var edoLuzExt = 0;

//Manejador de Evento click del boton que inicia el emulador
iniciar.addEventListener('click', ()=>{
  console.log('::: INICA EMULACION :::');
  data = setInterval(generarDatos, tasaRequest);
});

```

```

//funcion para generar datos de prueba aleatorios para lectura de variables del emulador
function generarDatos(){
  let d = new Date();
  let t = d.toLocaleTimeString();

  let temperatura = parseFloat((Math.random() * (36 - 22) + 22).toFixed(2));
  let humedad = parseFloat((Math.random() * (90 - 65) + 65).toFixed(2));

  //uso axios para enviar datos al api de ubidots
  axios(
    {
      method: 'POST',
      url: apiURL,
      data: {
        temp: temperatura,
        hum: humedad
      }
    }
  ).then(res => {
    console.log(t, ' - ', 'temp: ', temperatura,
      '; humedad: ', humedad ,res.statusText);
  }).catch(errors => {
    console.log('REQUEST DE VARIABLES NO PROCESADO');
  });
}

```

```

//Evento para encender/apagar luces interiores
luzInt.addEventListener('click', ()=>{
  axios(
    {
      method: 'POST',
      url: apiUrl,
      data: {
        lint: switchLuz('int')
      }
    }
  ).then(res => {
    console.log('LUCES INTERIORES : ', edoLuzInt);
  }).catch(errors => {
    console.log('REQUEST LUCES INTERIORES NO PROCESADO');
  });
});

//Evento para encender/apagar luces exteriores
luzExt.addEventListener('click', ()=>{
  axios(
    {
      method: 'POST',
      url: apiUrl,
      data: {
        lext: switchLuz('ext')
      }
    }
  ).then(res => {
    console.log('LUCES EXTERIORES : ', edoLuzExt);
  }).catch(errors => {
    console.log('REQUEST LUCES EXTERIORES NO PROCESADO');
  });
});
});

```

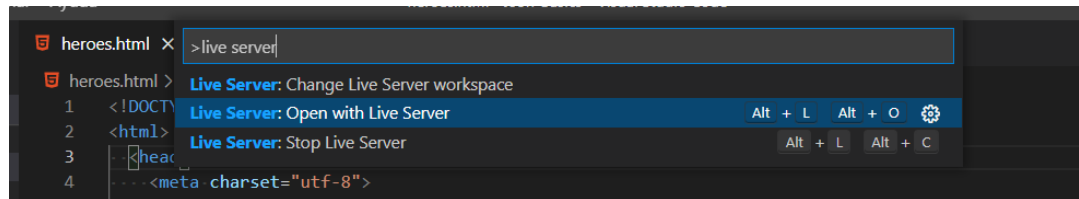
```

// Funcion que simula el cambio de estado de las luces
function switchLuz(tipo){
  if (tipo === 'int') {
    return edoLuzInt === 0 ? edoLuzInt = 1 : edoLuzInt = 0;
  } else {
    return edoLuzExt === 0 ? edoLuzExt = 1 : edoLuzExt = 0;
  }
}

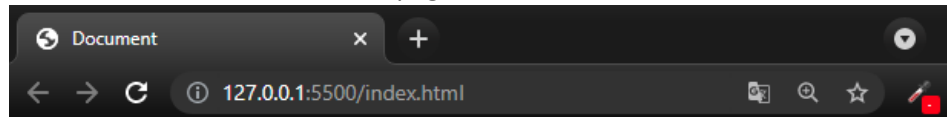
//Manejador de eventos para detener el emulador
detener.addEventListener('click', ()=>{
  clearInterval(data);
  console.log(":::: EMULACION DETENIDA ::::");
});

```

- Una vez creados los archivos abra la extension Active la extension live server para monitorear los cambios en la pagina web, dentro de vs code, presione las teclas: ctrl+shift+p, esto abra una barra de busqueda, teclee: "live server" y seleccione la opcion: "abrir con live server", esto abra una navegador web con la informacion contenida en "index.html"



- Debera obtener una salida de la pagina como esta:



Emulador IoT - UNIVO

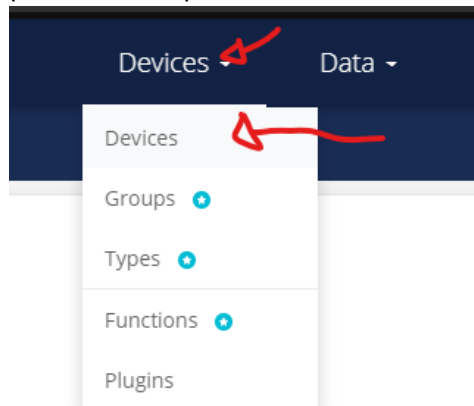
Lectura de Variables

Inicar Detener

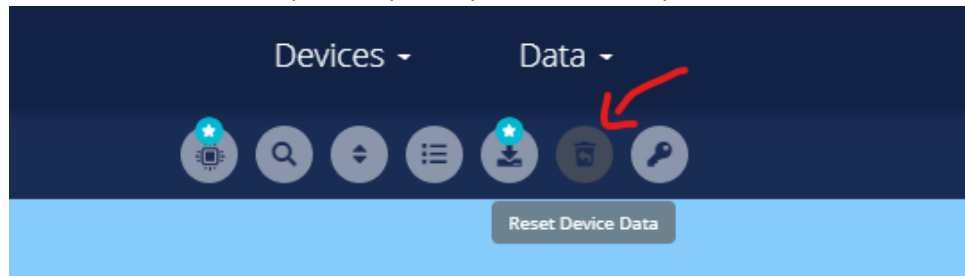
Acciones de la Habitacion

On/Off - Luces Interiores On/Off - Luces Exteriores

- Para realizar la prueba de enlace entre el emulador, el dispositivo virtual (hab-101) y el dashboard, limpie los datos de prueba ingresados anteriormente, par esto vaya a la pantalla de dispositivos:



7. Pulse sobre el dispositivo hab-101, una vez dentro del panel de variables ubique los botones de menu de la parte superior y seleccione la opcion: "Reset Device Data"



8. Escriba en la ventana emergente el nombre del dispositivo virtual y presione "Delete"

hab-101

Delete Device Data

Are you sure you want to remove all the data of this Device?

This action will erase all the values stored in the selected time span in every variable of this Device. This action can't be undone.

1969-12-31 18:00:00

2021-07-30 23:59:59

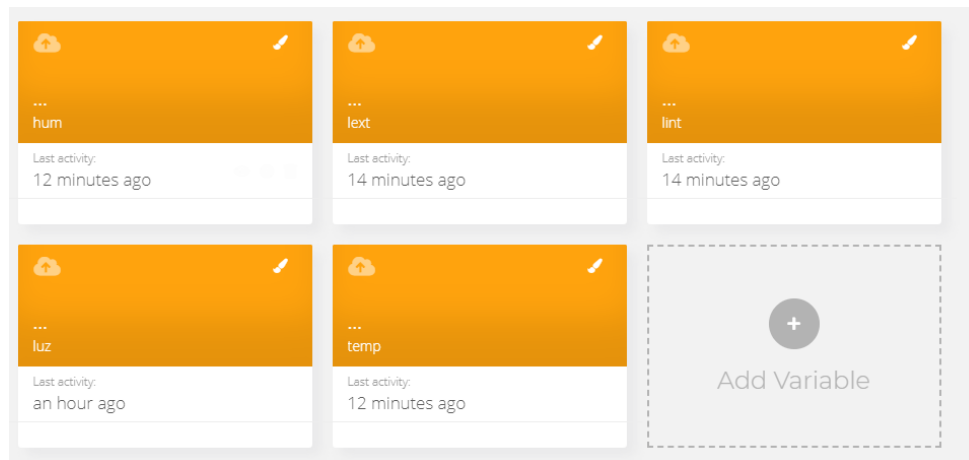
Please type in the name of the Device to continue:

hab-101

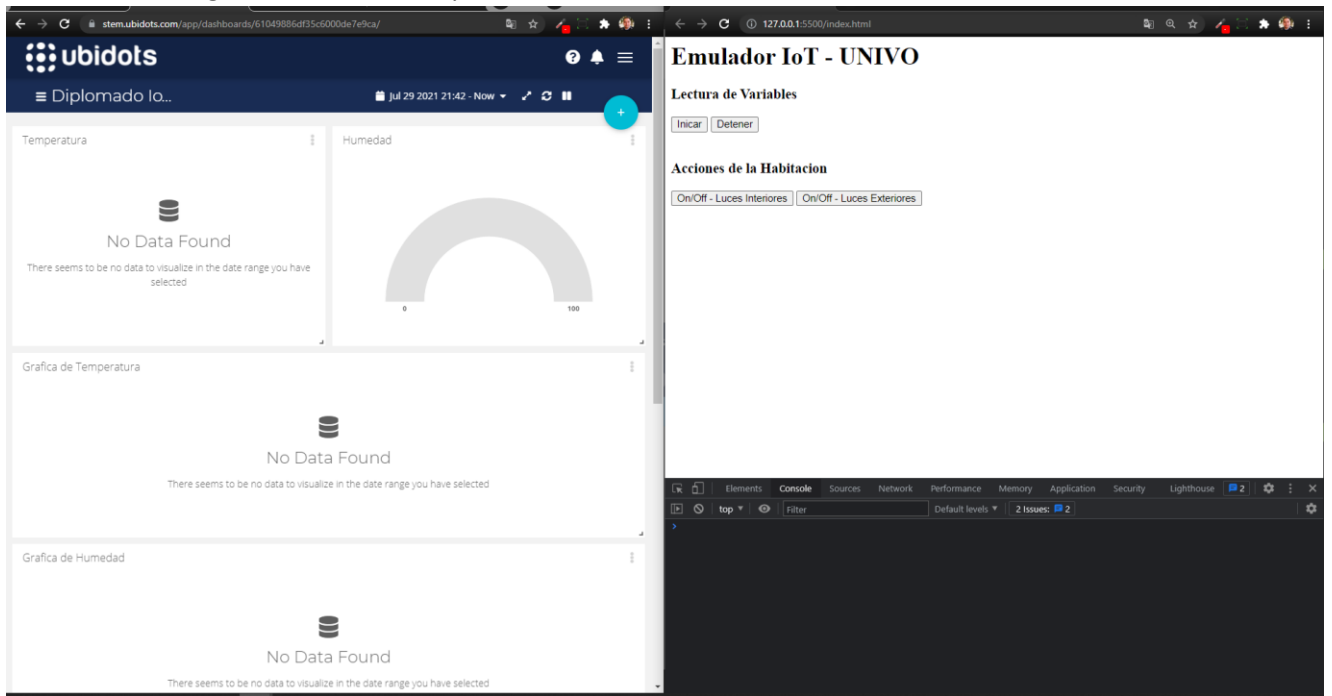
Cancel

Delete

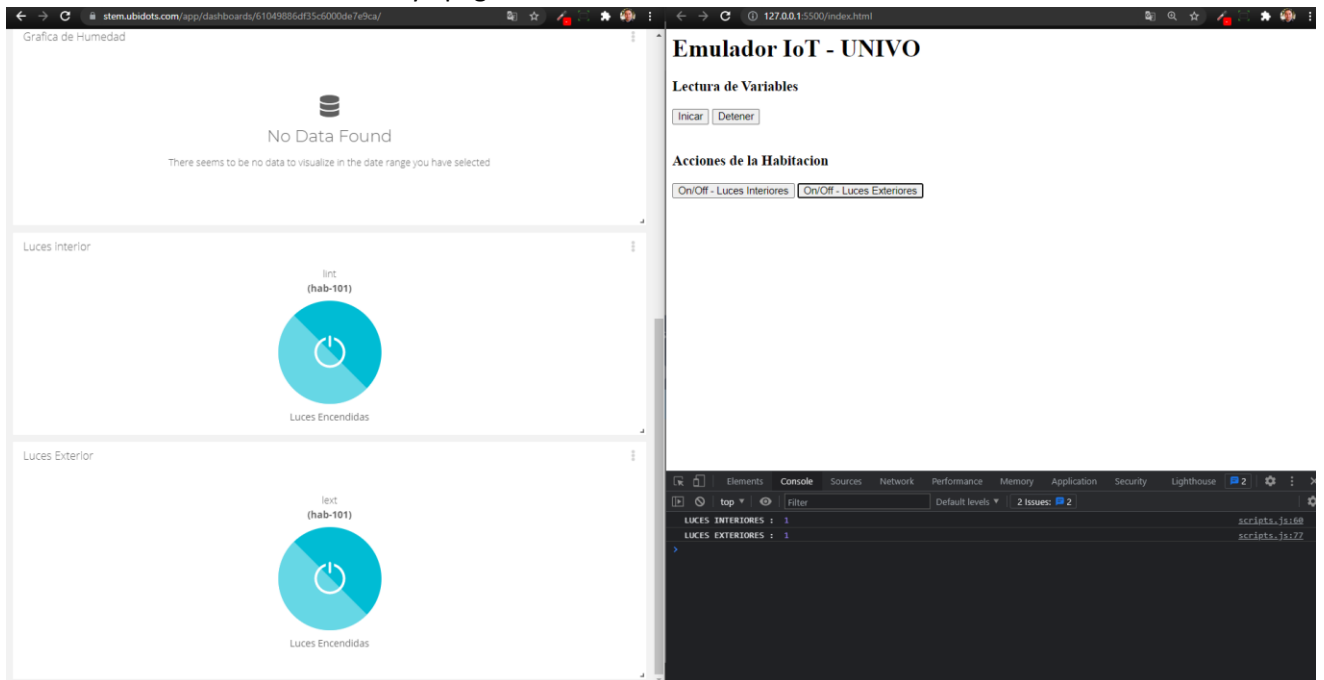
9. Actualice la pagina y los valores de las variables aparecen en blanco



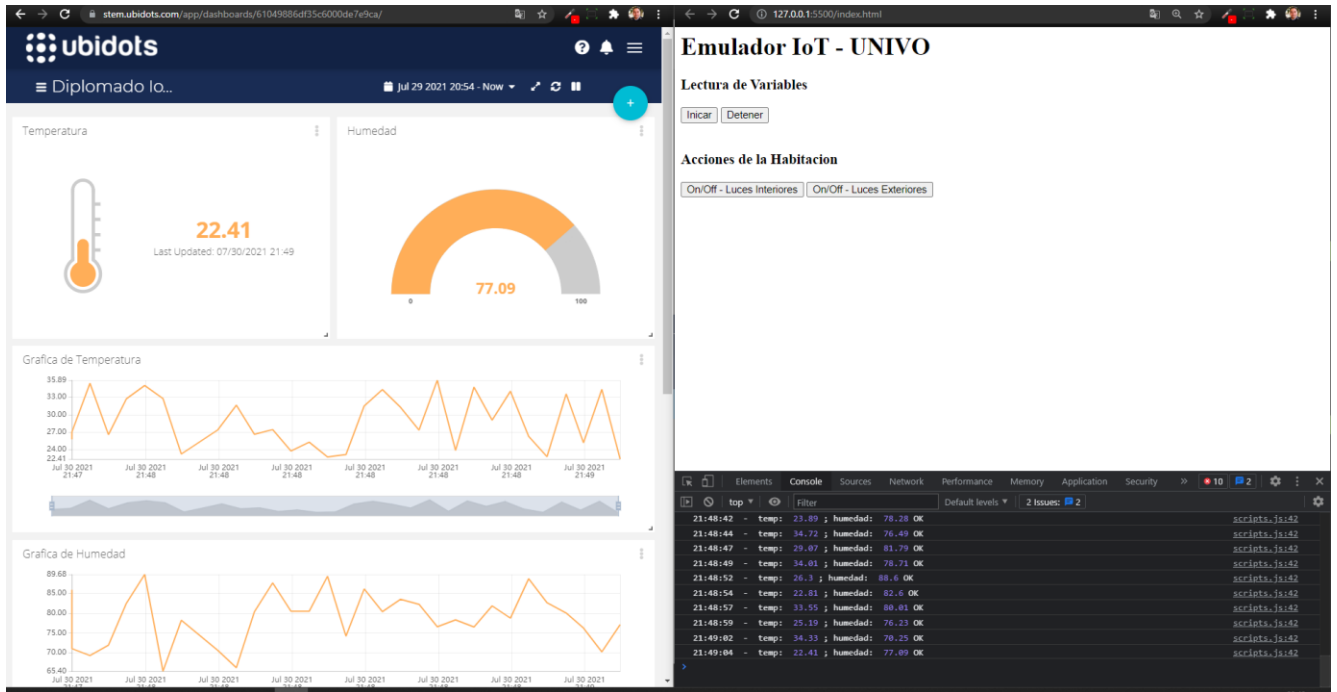
10. Regrese al Dashboard y ajústelo tal manera que pueda ver las dos ventanas del navegador al mismo tiempo



11. Pruebe con encender y apagar las luces de la habitación



12. Ahora su Dashboard esta conectado con el dispositivo virtual que recibe datos del emulador en tiempo real



Evidencia documental de la practica:

Suba un archivo pdf con las capturas de pantalla de los siguientes elementos:

1. Panel de variables del dispositivo con las variables configuradas para el ejercicio
2. Pantalla de pruebas en postman con todas las variables OK
3. Dashboard del ejercicio
4. Pantalla del Emulador corriendo en live server
5. Pantalla combinada del Dashboard trabajando en conjunto con el emulador

Ejercicio 2 – Crea tu propio Dashboard

Tiempo estimado: 60 min

Tomando como base los conocimientos adquiridos en la ejercicio 1 realiza tu propio Dashboard para una app IoT, puedes tomar como base algun ejemplo que se realaciones con tu entorno laborar, realizar una busqueda en la red o simplemente inventar uno.

El dispositivo virtual y el emulador, **deberan manejar al menos 4 variables de lectura y al menos 2 variables de interaccion.**

Cada variable de lectura debera tener un widget para visualizar el valor actual y un widget para la representacion del historico de valores de la variable (grafica).

Como evidencia documental de esta actividad deberas generar un archivo PDF con los siguientes puntos:

- Descripcion del problema que resuelve o campo de aplicación del Dashboard
- Listado de variables de lectura y variables de interaccion
- Captura de pantalla de configuracion del dispositivo virtual en ubidots
- Captura de pantalla de las pruebas de comunicación desde postman
- Captura de pantalla del codigo en javascript del emulador del dispositivo fisico
- Captura de pantalla del trabajo conjunto entre el emulador y el dashboard