José Antonio Vacas Martínez

# Taller didáctico con micro:bit sobre IOT

## Versión 0.99.3



Licencia CC by SA by @javacasm

José Antonio Vacas Martínez

### https://elCacharreo.com

Diciembre 2022 - Septiembre 2023

# Taller didáctico de IOT

El objetivo del taller es aprender a hacer que nuestra placa micro:bit envíe los datos de sus sensores a un servidor que los mostrará en pantalla. A partir de esos datos el servidor enviará órdenes al led RGB de nuestra micro:bit. Es lo que conocemos como **IOT**

## ¿Qué es micro:bit?



Micro:bit es una placa de desarrollo electrónico diseñada para enseñar a los estudiantes la programación y el pensamiento computacional. Está diseñada especialmente para ser fácil de usar y se puede programar utilizando una variedad de lenguajes de programación, con [bloques](https://makecode.microbit.org), Scratch, Blockly y Python.

Micro:bit fue creada por la BBC en colaboración con una serie de socios, incluyendo el Ministerio de Educación del Reino Unido y una serie de empresas tecnológicas. La placa micro:bit fue diseñada para ser utilizada en la enseñanza de la programación y el pensamiento computacional en las escuelas del Reino Unido, y se distribuyó gratuitamente a más de un millón de estudiantes en el año 2015.

Desde entonces, micro:bit se ha convertido en una placa de desarrollo electrónica muy popular en todo el mundo y ha sido utilizada en una amplia variedad de proyectos y actividades educativas en todo el mundo. Actualmente, la placa micro:bit es desarrollada y mantenida por la Fundación micro:bit, una organización sin fines de lucro que trabaja para promover el uso de la tecnología en la educación.

La placa micro:bit tiene un procesador de baja potencia, una pantalla de leds, sensores de movimiento y orientación, y puede conectarse a otros dispositivos mediante Bluetooth. También cuenta con una serie de botones y entradas de pin para conectar componentes externos.

Algunas de las cosas que se pueden hacer con micro:bit son:

* Crear proyectos de robótica y control de motores
* Desarrollar aplicaciones para la Internet de las cosas (IoT)
* Crear juegos y aplicaciones interactivas
* Realizar experimentos científicos y de ingeniería
* Aprender a programar y a pensar de manera computacional

micro:bit es una herramienta muy versátil y puede ser utilizada para una amplia variedad de proyectos y actividades educativas.

## Detalles

La placa micro:bit es una placa de desarrollo electrónica compacta y de bajo costo que se ha diseñado especialmente para enseñar a los estudiantes la programación y el pensamiento computacional. Esta placa cuenta con los siguientes componentes:

* Procesador: La placa micro:bit tiene un procesador de baja potencia que se encarga de ejecutar el código que se le envía.
* Pantalla de LED: La placa micro:bit cuenta con una pantalla de LED de 5x5 que se puede utilizar para mostrar mensajes, patrones y gráficos.
* Botones: La placa micro:bit tiene dos botones que se pueden utilizar para interactuar con el dispositivo. Además incluye un sensor táctil en su logo.
* Sensores: La placa micro:bit cuenta con un acelerómetro y un magnetómetro que se pueden utilizar para detectar el movimiento y la orientación del dispositivo. También incluye sensores de temperatura, luz y un micrófono
* Conexión Bluetooth: La placa micro:bit tiene un módulo Bluetooth que se puede utilizar para conectarse a otros dispositivos y enviar y recibir datos.
* Puertos: La placa micro:bit tiene una serie de puertos y entradas de pin que se pueden utilizar para conectar componentes externos, como sensores y motores.
* Conector de batería: La placa micro:bit tiene un conector para una batería de litio que se puede utilizar para alimentar el dispositivo.
* Conector USB: La placa micro:bit tiene un conector USB que se puede utilizar para cargar la batería y para conectar el dispositivo a una computadora para programarlo y transferir datos.
* Altavoz/buzzer capaz de reproducir sonidos.

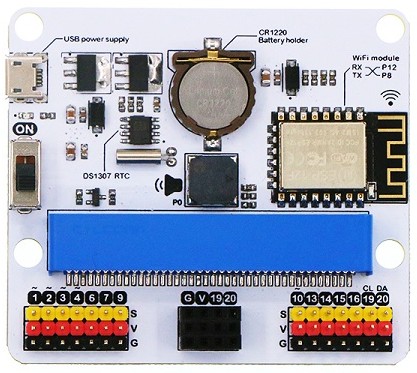
# Componentes

Vamos a utilizar estos componentes:

micro:bit



Placa IOT:bit



Pantalla OLED



Sensor de temperatura



Sensor de luz



Led RGB



3 x Cables conexión de los sensores



Cable USB

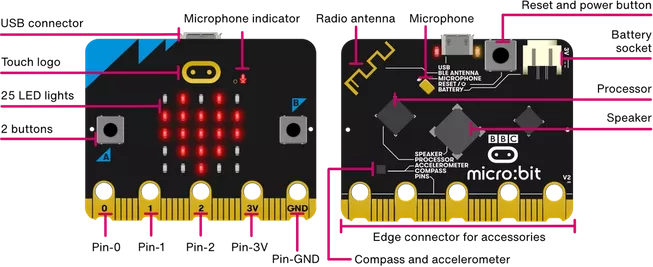


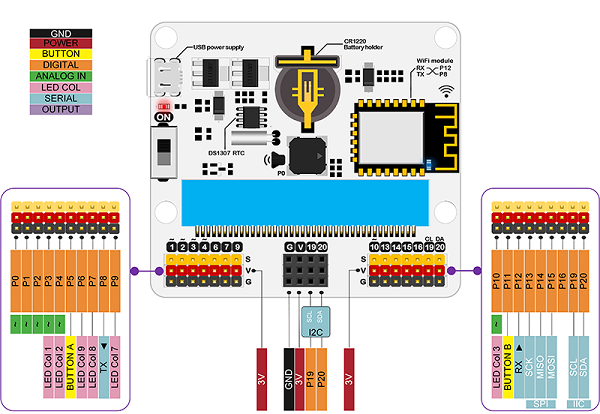
# Servidor

Raspberry Pi 4



## Más detalles





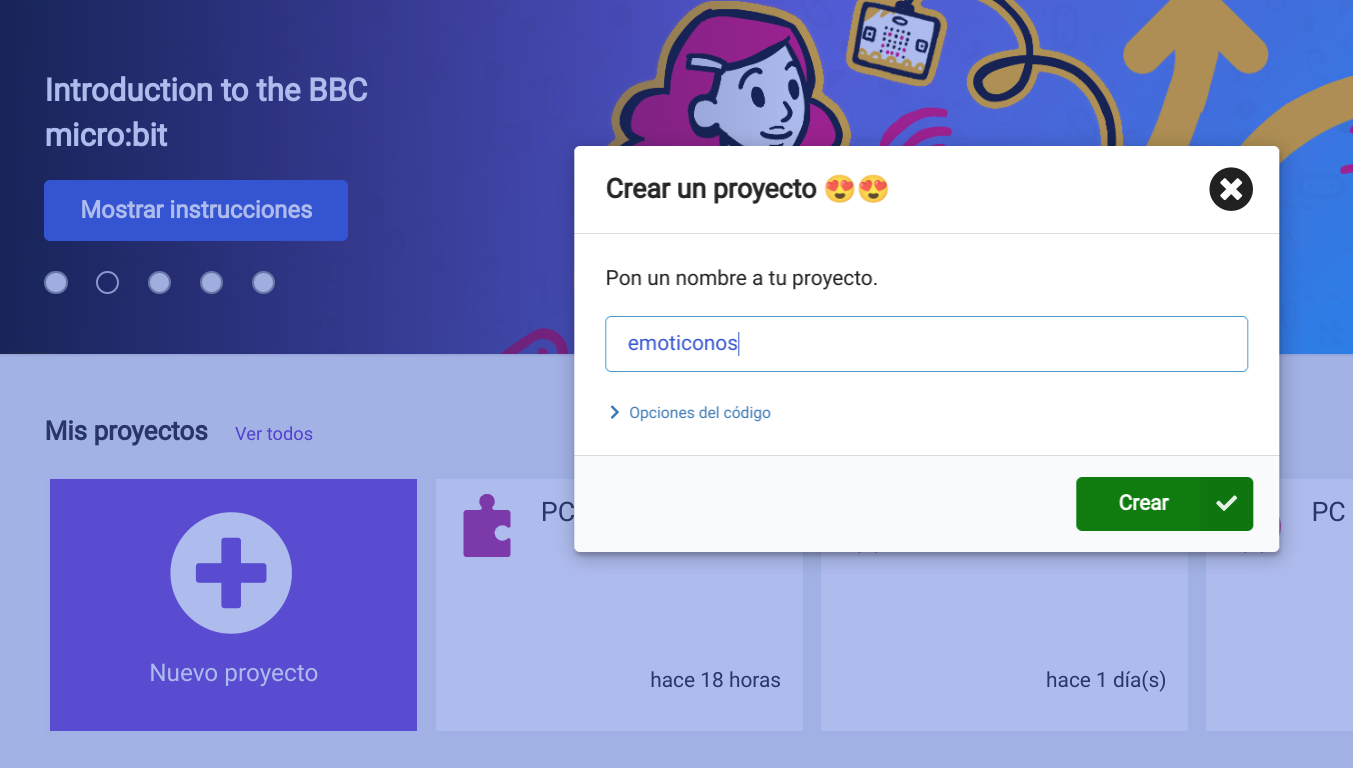
# Montaje

## Simulador micro:bit mostrando un emoticono

Necesitamos:

* micro:bit
* cable USB
* <https://makecode.microbit.org>

Desde la página de makecode pulsamos en “Nuevo proyecto” y le ponemos de nombre “emoticonos” y pulsamos el botón “Crear”



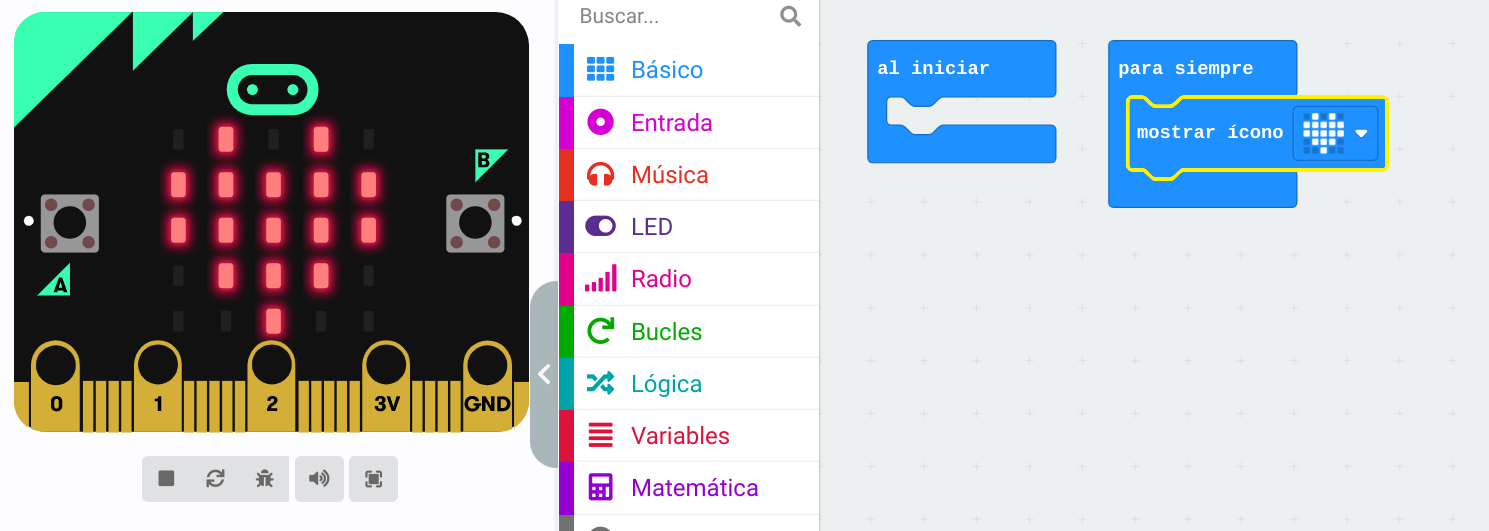
Vamos a usar el editor de bloques de Makecode



En la paleta básico encontramos bloques para mostrar dibujos en su pantalla de 5x5 leds

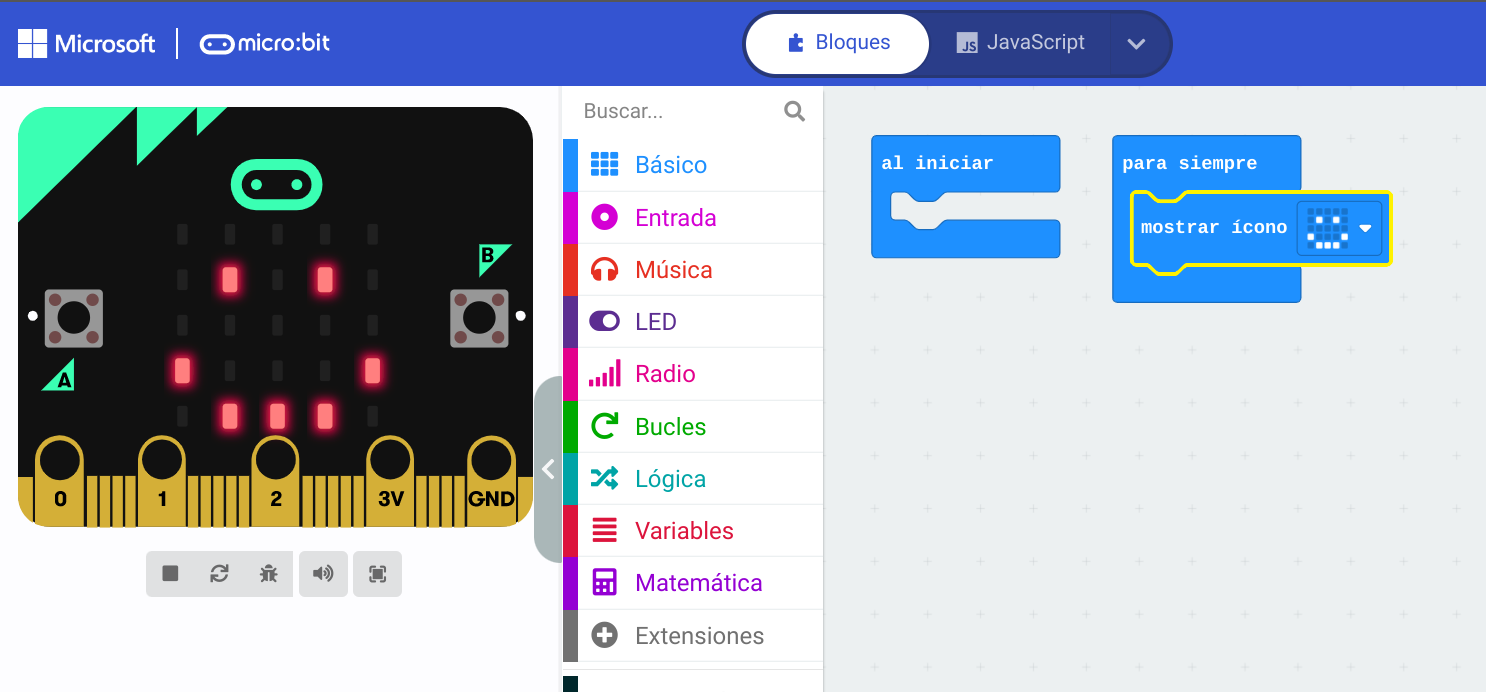


Arrastramos el bloque “mostrar icono” dentro del bloque “Para siempre” que tiene forma de “C”

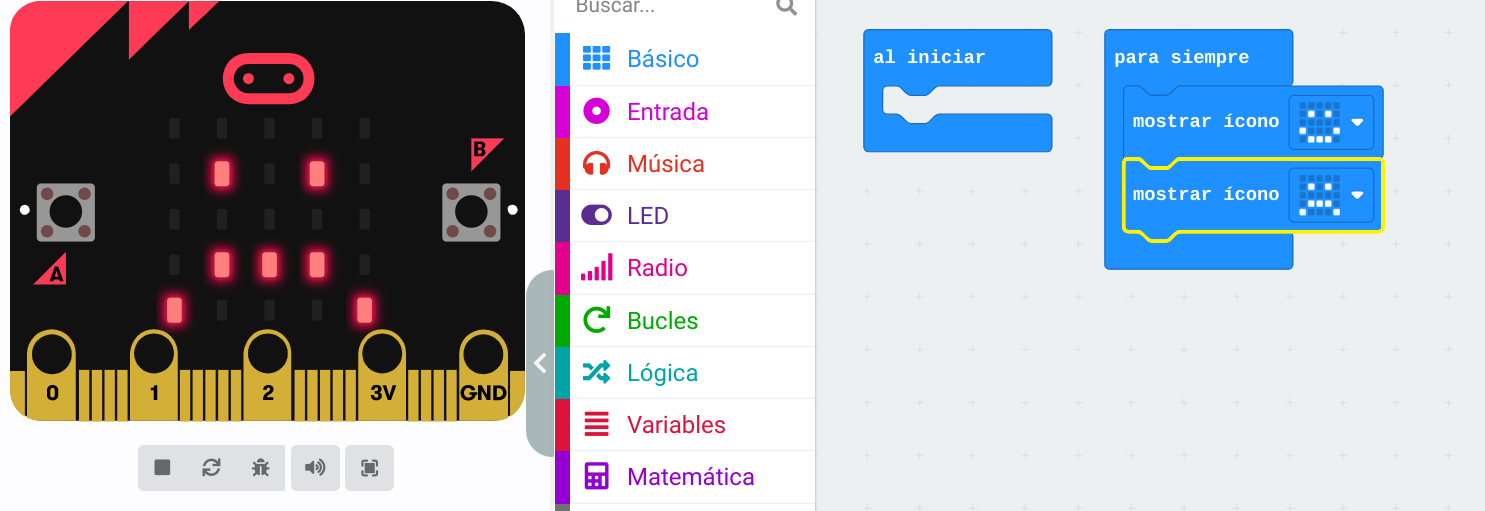


Verás que el simulador de micro:bit que se muestra a la izquierda de la página mostrará el icono.

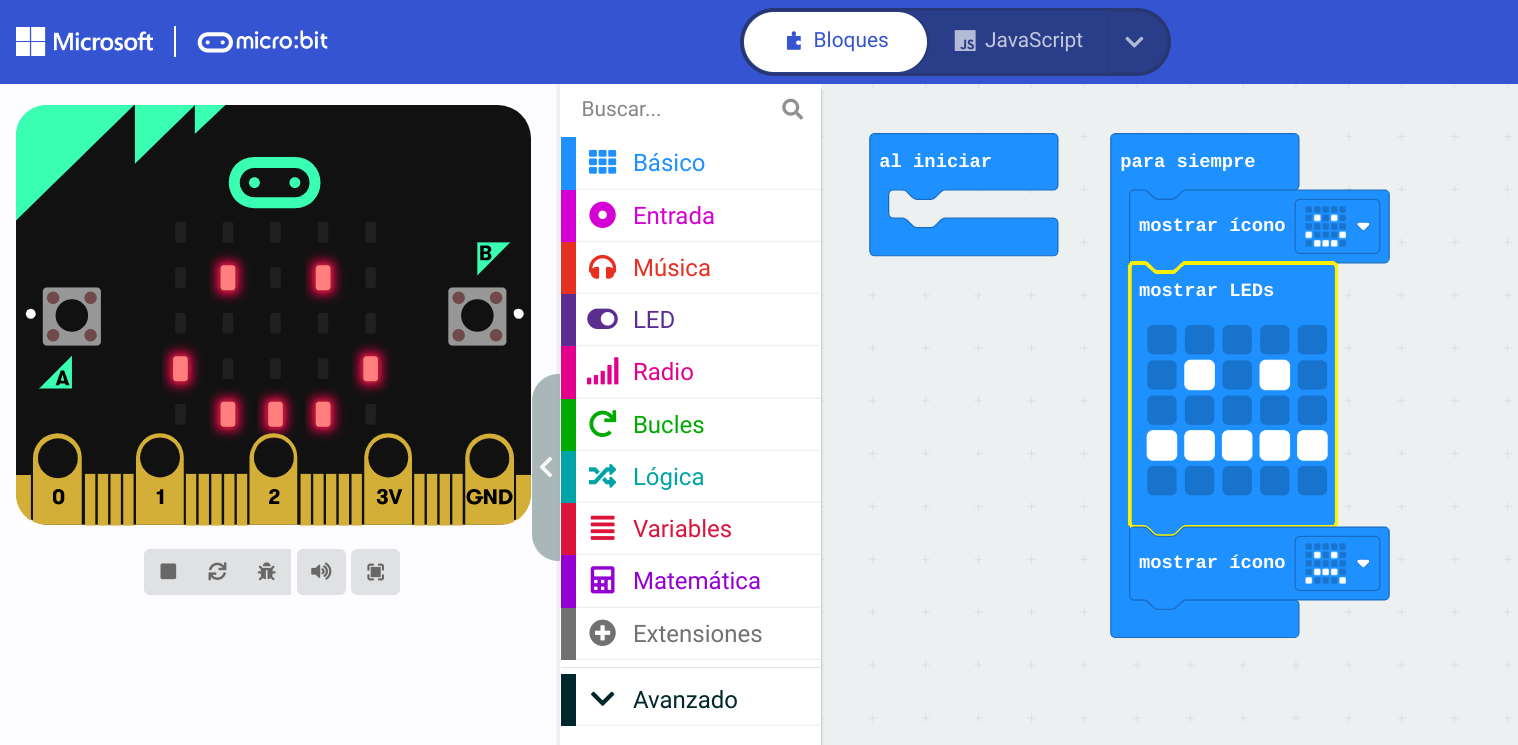
Ahora podemos seleccionar nuestro icono/dibujo preferido, y el simulador hará lo mismo



Si añades otro bloque de mostrar icono se mostrarán alternativamente los 2 dibujos en la pantalla



Incluso puedes crear tu propio dibujo usando el bloque “Mostrar leds”



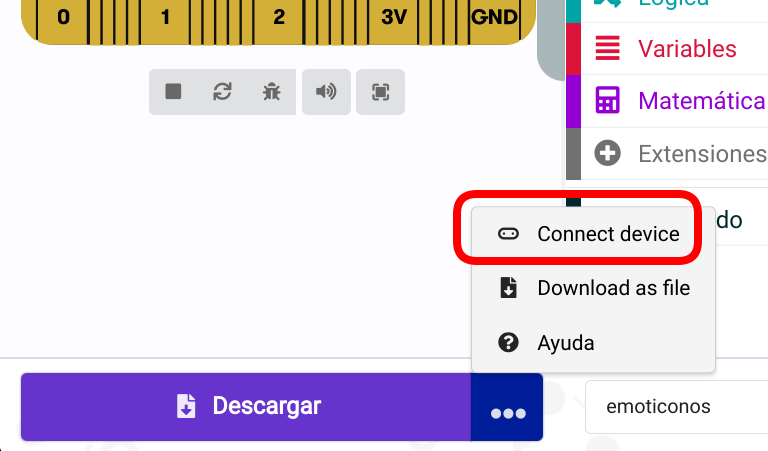
[Ejemplo](https://makecode.microbit.org/_DemH1JcjEJae)

Ahora vamos a pasar el programa a nuestra placa micro:bit

Siempre que queramos que el programa funcione en nuestra placa tenemos que pulsar el botón “Descargar”

La primera vez tenemos que hacer este proceso:

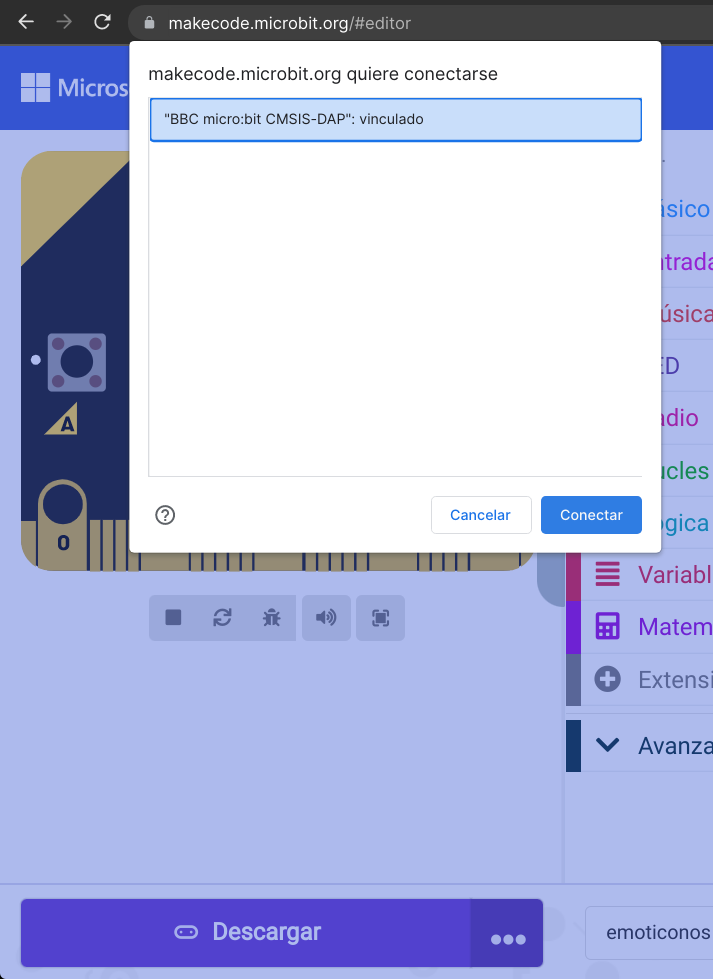
* Conectamos la micro:bit con el cable USB al ordenador
* Pulsamos en el botón de los 3 puntitos la opción



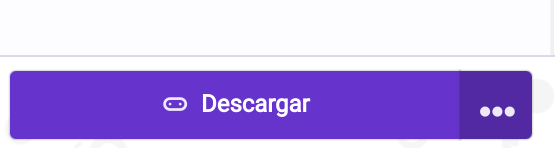
* Aparece una ventana y el nombre de nuestra placa en ella



* La seleccionamos y pulsamos “Conectar”



El icono del botón “Descargar” cambia, mostrando el logo de micro:bit, lo que quiere decir que ya está emparejada

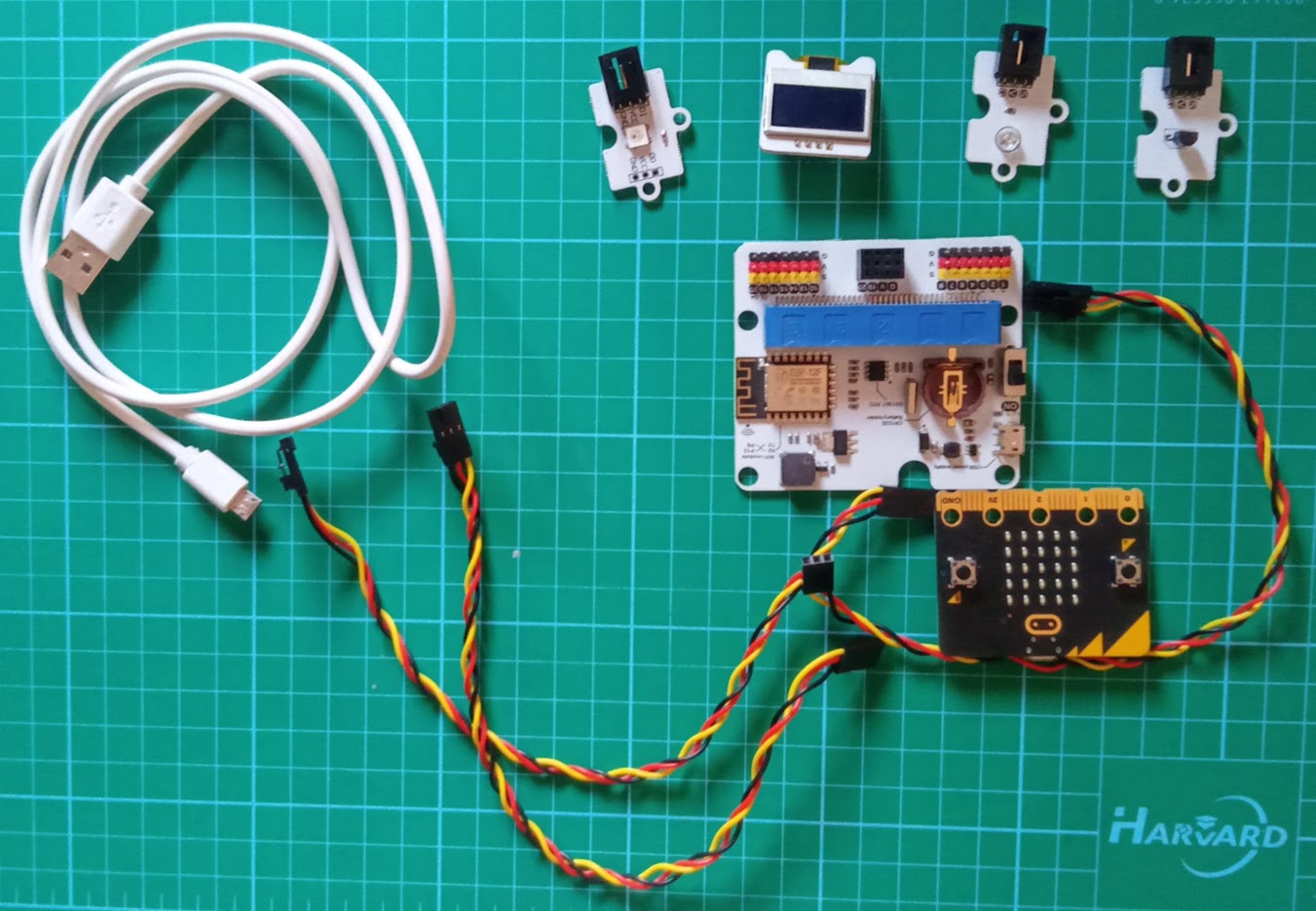


* Pulsamos el botón “Descargar” y tras unos segundos, en los que parpadea el led naranja de la parte de atrás de nuestra micro:bit, ya la tenemos programada y ejecutando nuestro programa.

Este proceso es similar al emparejamiento bluetooth de dispositivos con lo que en las placas que ya se han emparejado con un ordenador no habrá que repetirlo.

## Termómetro en la pantalla

Necesitamos:



* micro:bit
* Cable USB
* Placa IOT:bit
* Pantalla OLED
* Sensor temperatura
* Sensor luz
* LED RGB
* 3 cables de sensores

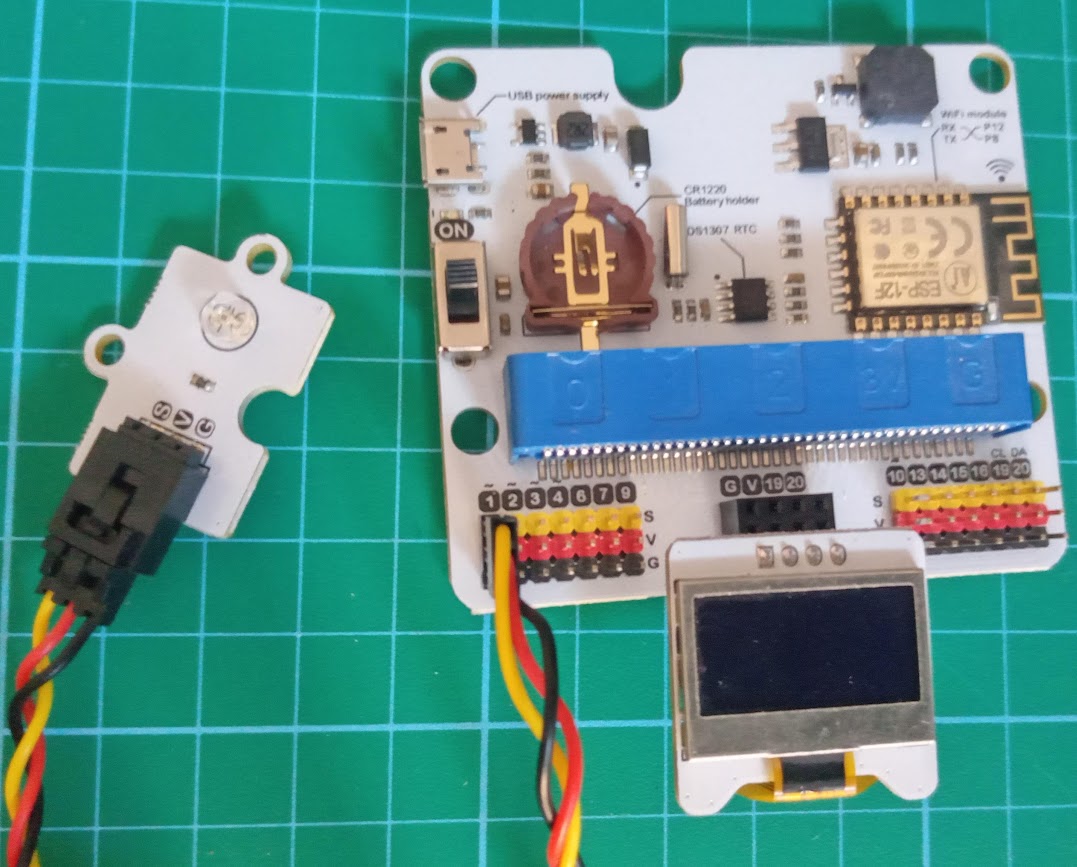
### Montaje

**¡¡TODO EL MONTAJE SE HACE CON LA PLACA DESCONECTADA!!**

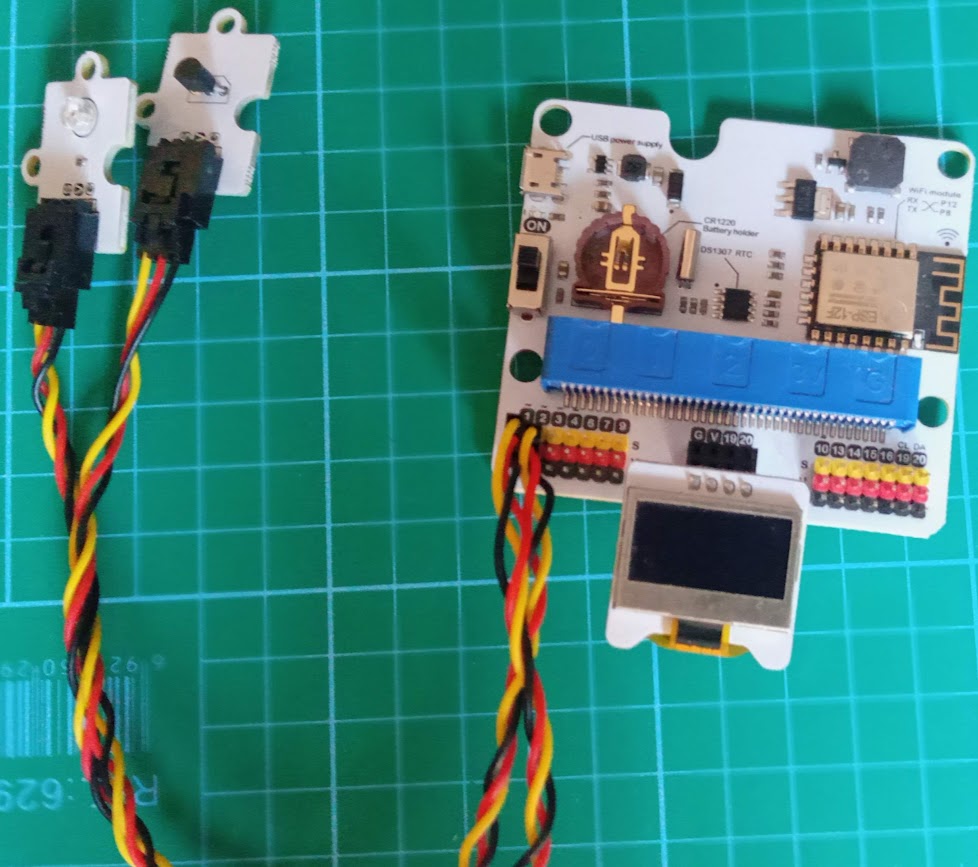
* Conectamos la pantalla



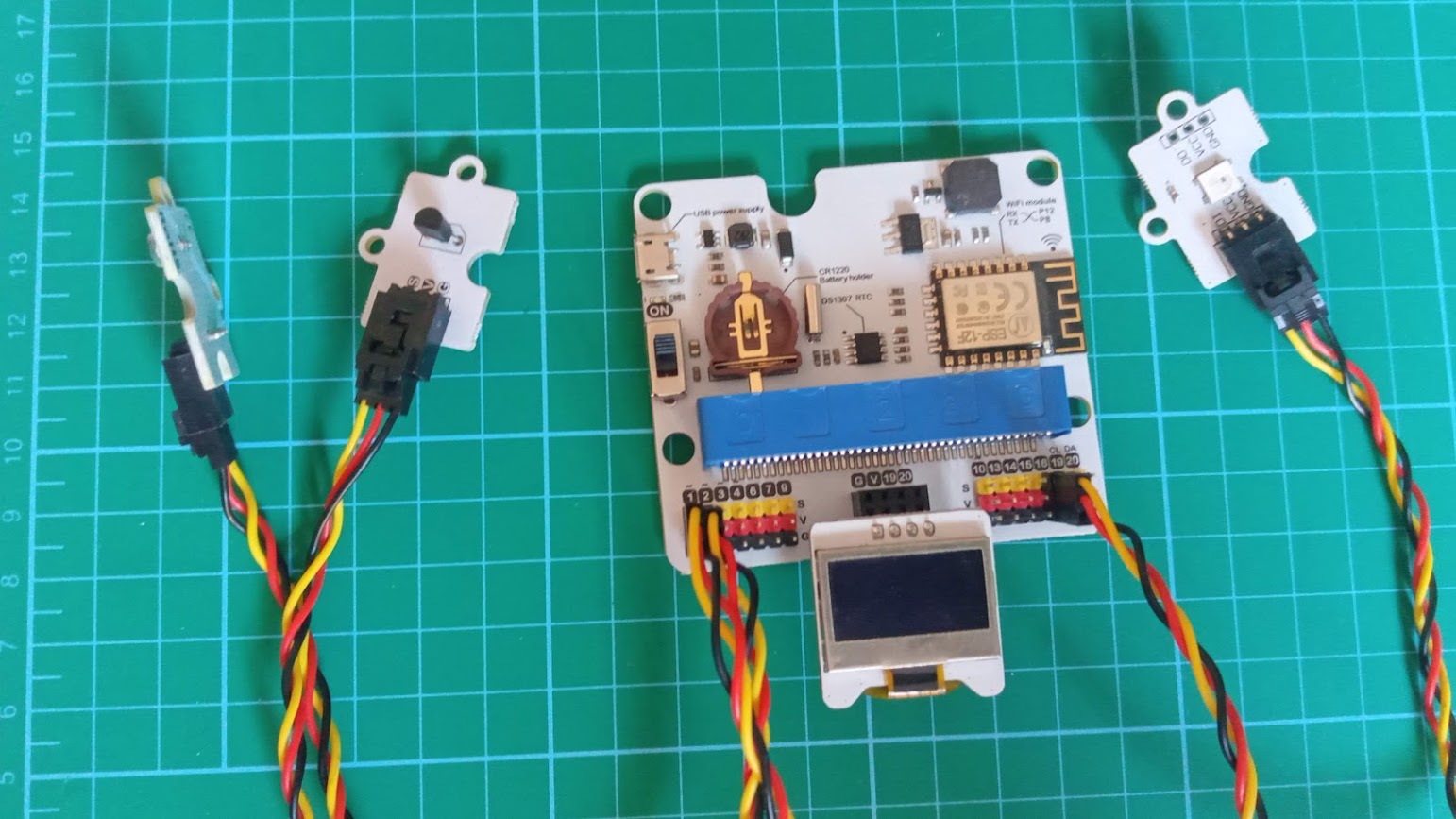
* Conectamos el sensor de luz en P1 ¡¡OJO A LOS COLORES!!
  + Cable amarillo - Pin amarillo
  + Cable rojo - Pin Rojo
  + Cable negro - Pin Negro



* Conectamos el sensor de temperatura en P2

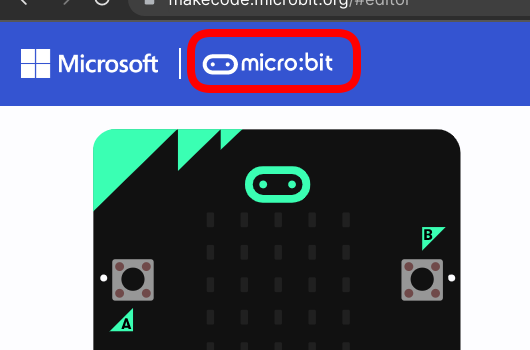


* Conectamos el led al pin P16



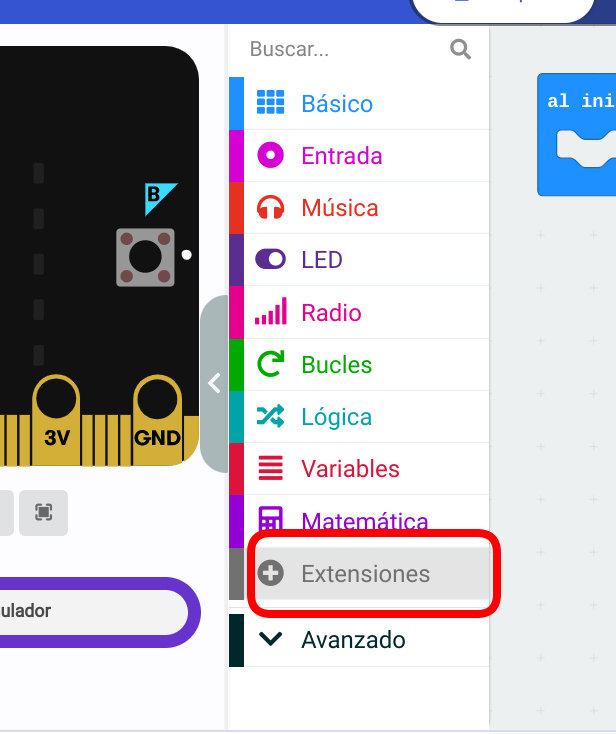
### Programación

* Creamos un nuevo proyecto pulsado en el logo de microbit

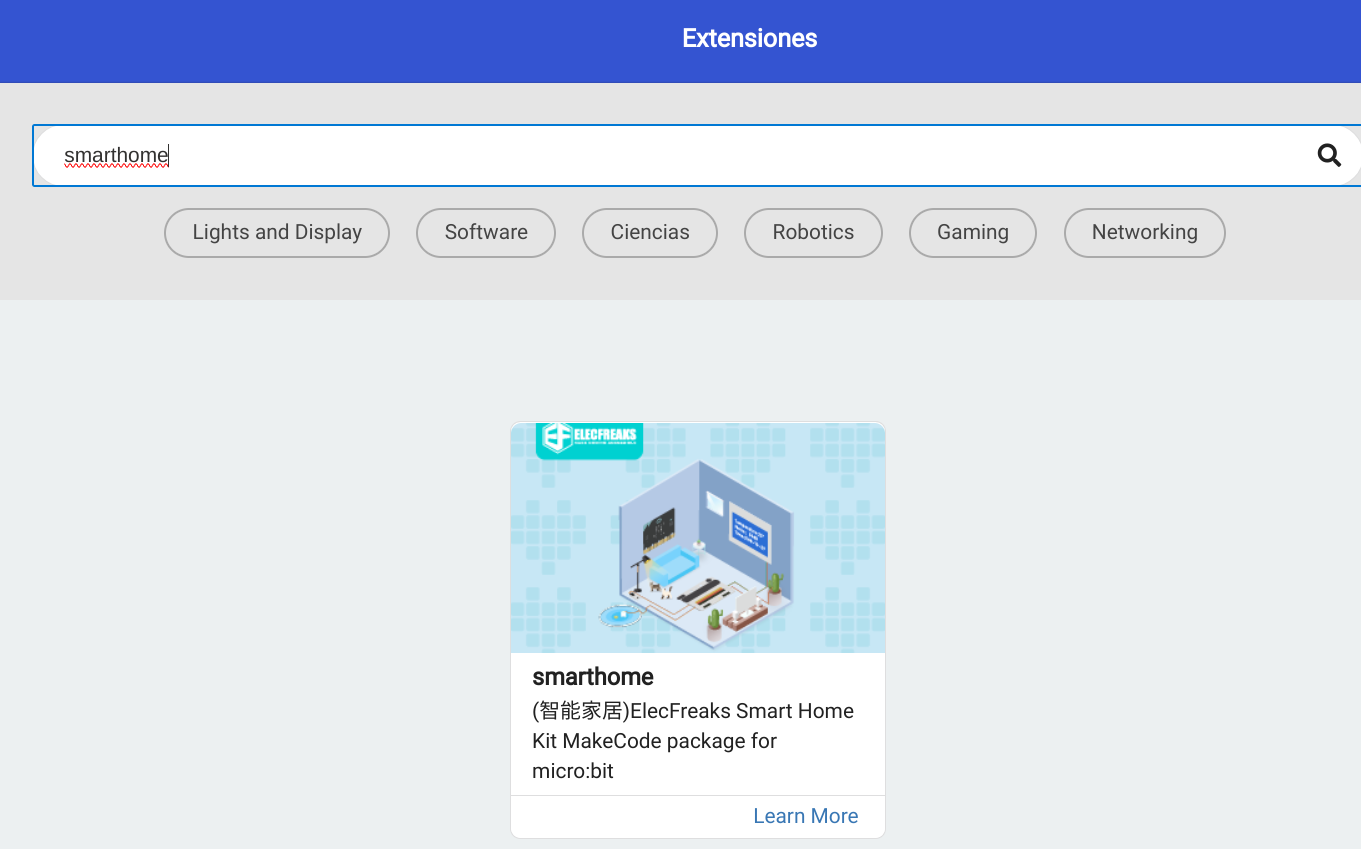


Para usar los sensores vamos a necesitar nuevos bloques y por eso vamos a incluir la extensión de nuestros sensores y componentes.

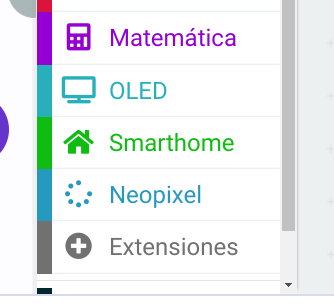
Pulsamos en “Extensiones”



Y buscamos la extensión “Smarthome”

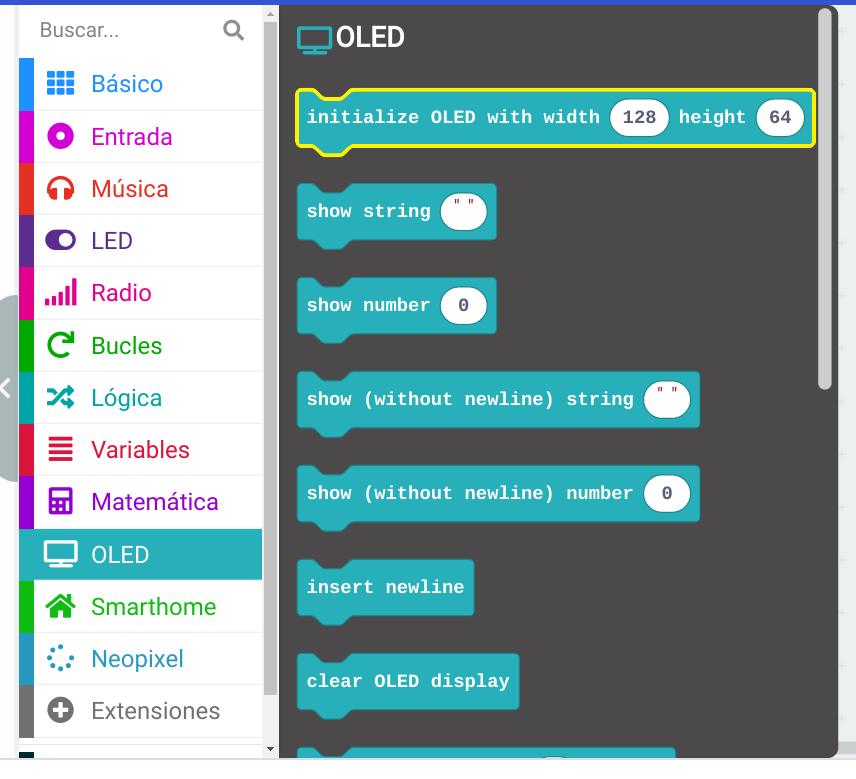


Tras pulsar sobre ella aparecerán nuevas paletas de bloques

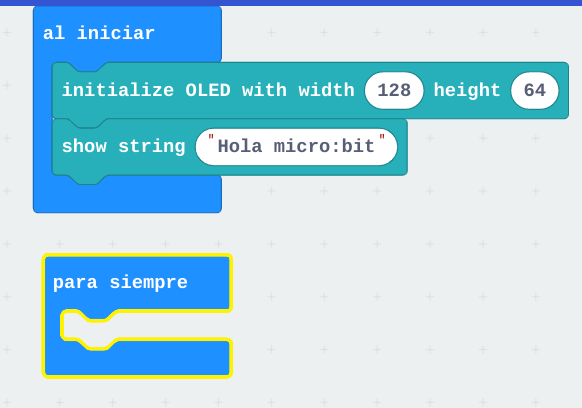


Vamos a mostrar un texto en la pantalla

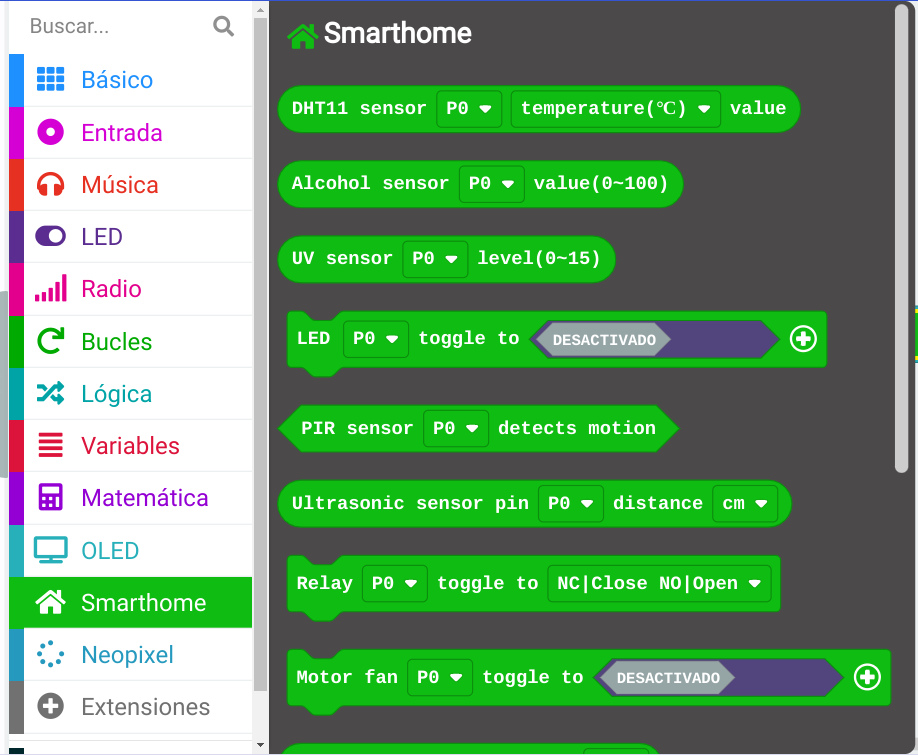
* Abrimos la paleta “OLED”



* Añadimos los bloques de “Initialize OLED ….” y “show string…” ¡¡Hay que aprender inglés!!
* Completaremos con un saludo



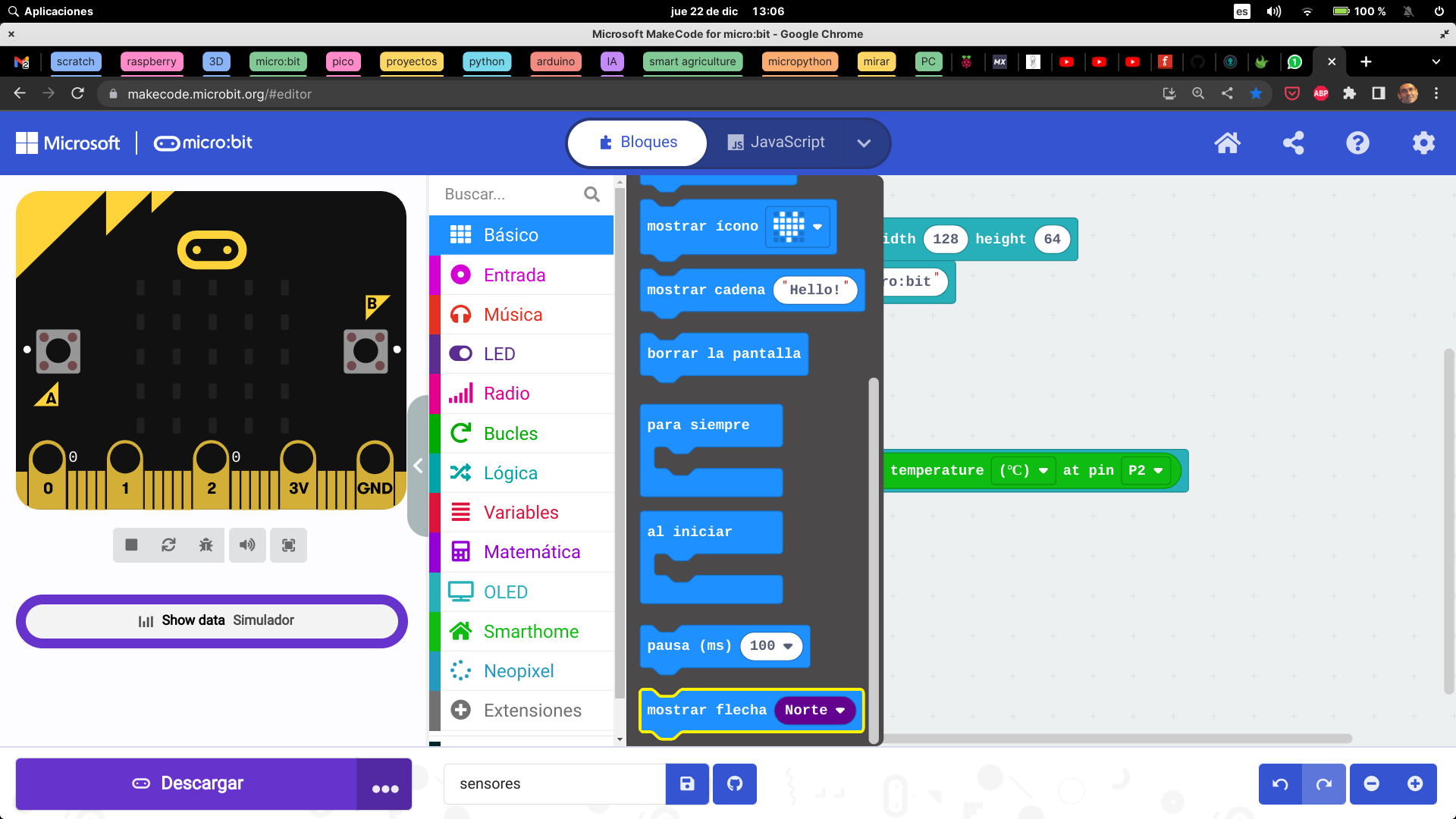
* Reprogramamos nuestra micro:bit y veremos el saludo en pantalla
* Añadimos el bloque de borrado de la pantalla “clear OLED display” y el de “show number” de la paleta OLED
* Ahora abrimos la paleta “Smarthome” y añadimos el bloque “value of temperature…” dentro del anterior “show number”



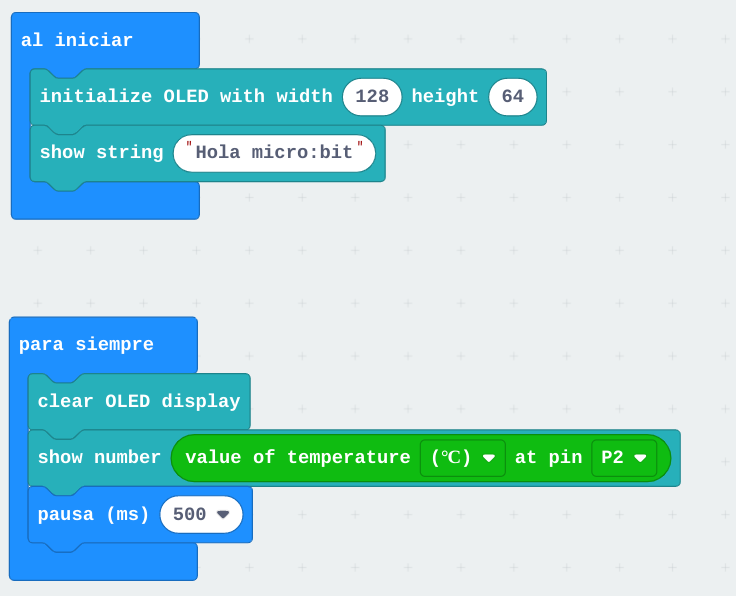
* Seleccionamos P2 que es donde hemos conectado nuestro sensor



* La micro:bit funciona muy rápido. Por eso vamos a añadir un pequeño retardo para que nos dé tiempo a leer el valor de la temperatura. Abrimos la paleta “Básico” y cojemos el bloque “pausa”

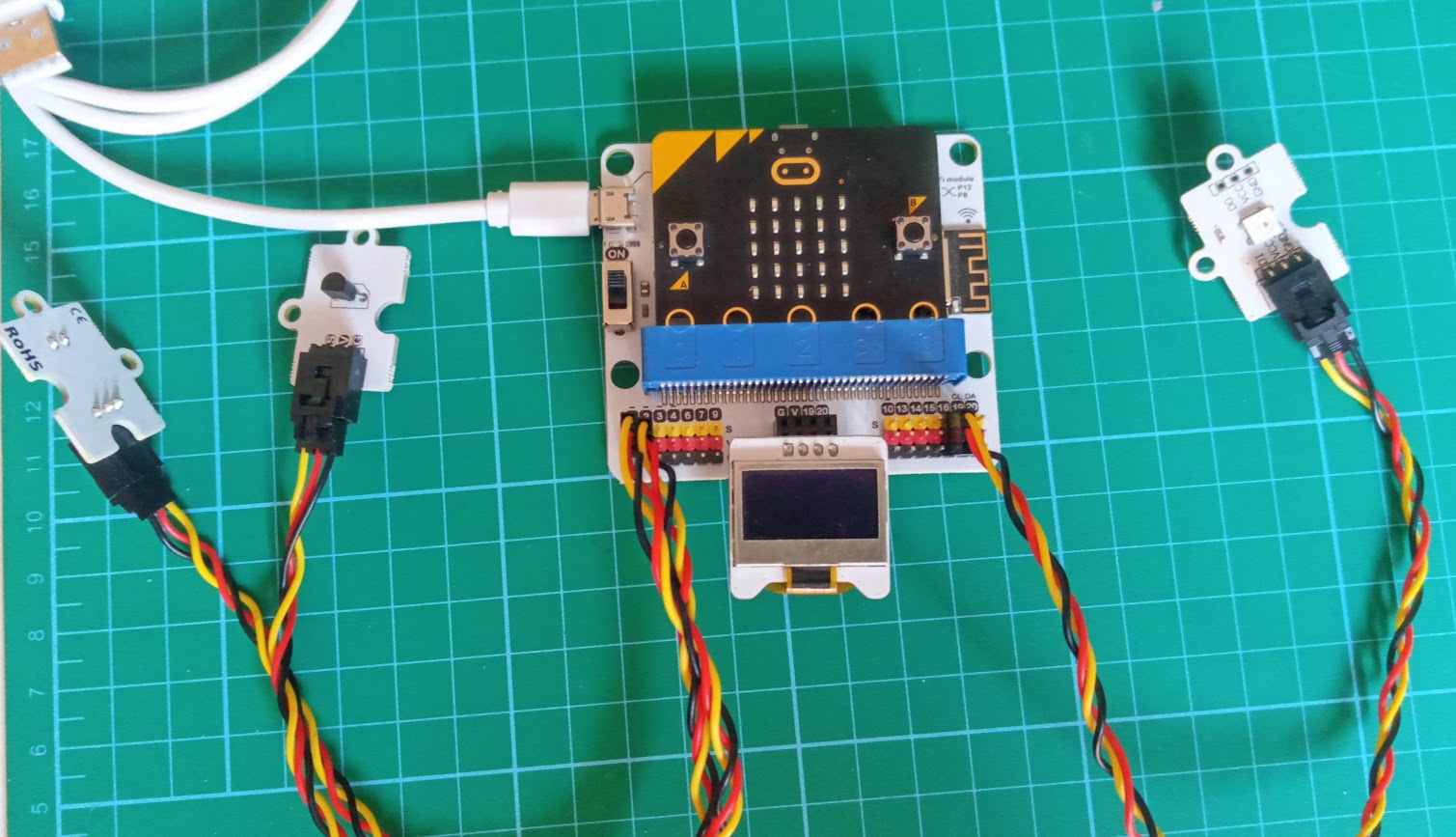


Nuestro programa queda así:



Programamos nuestra placa y veremos los valores en pantalla

Para que mida mejor cambiamos el cable USB para que alimente la placa IOT:bit

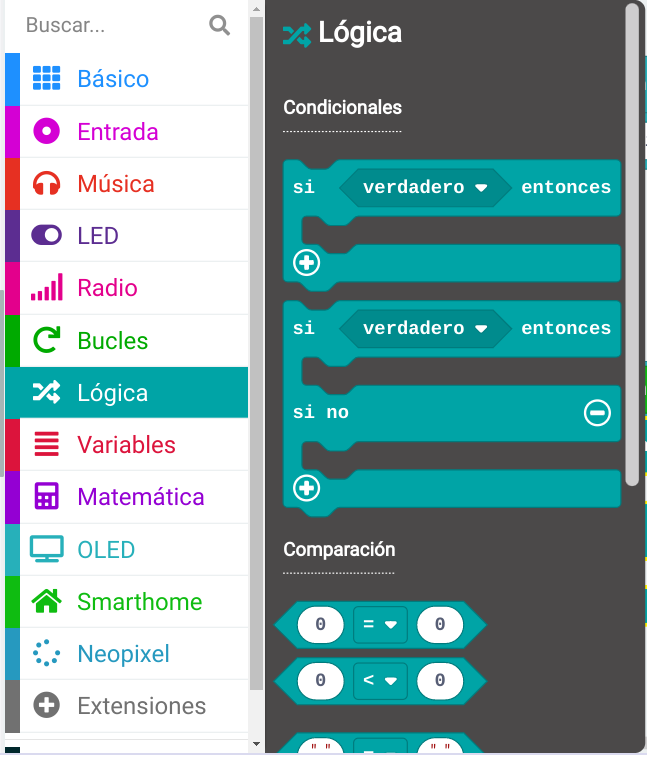


[Programa: termómetro](https://makecode.microbit.org/_aXR3UD7g4f17)

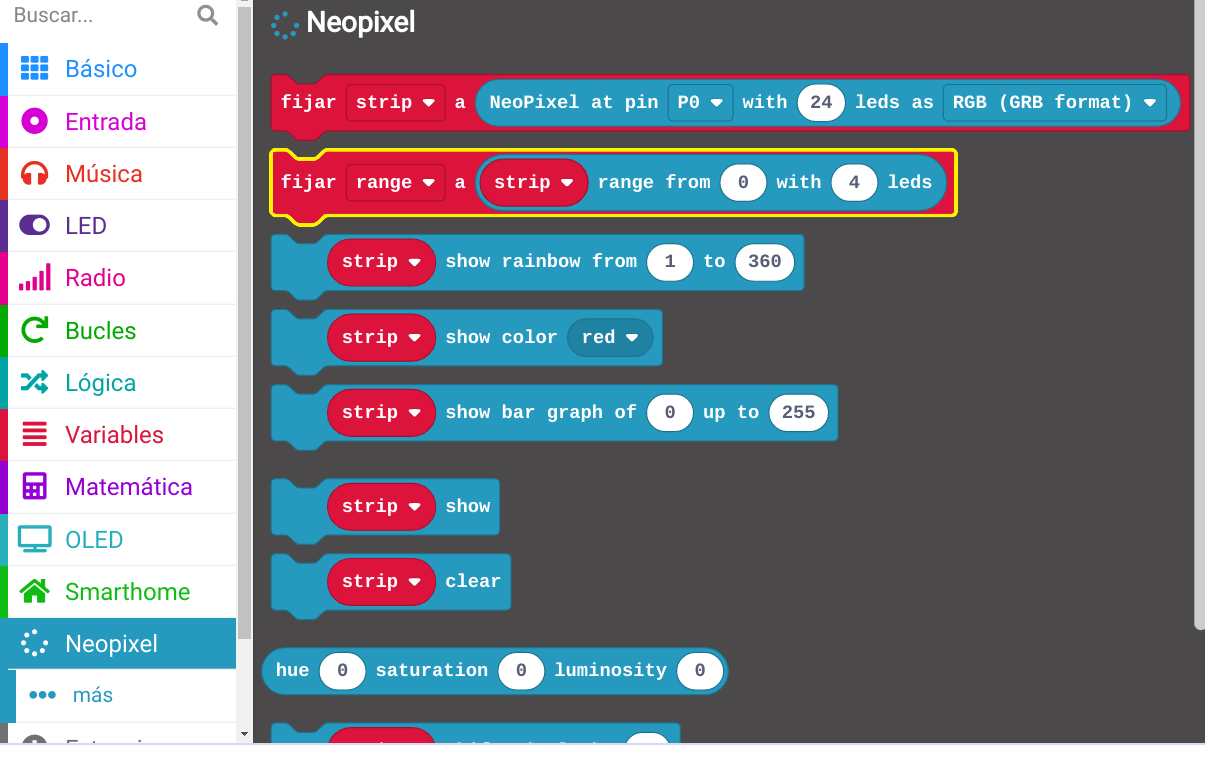
### Proyecto Termostado

Si queremos convertir nuestro termómetro en un termostato:

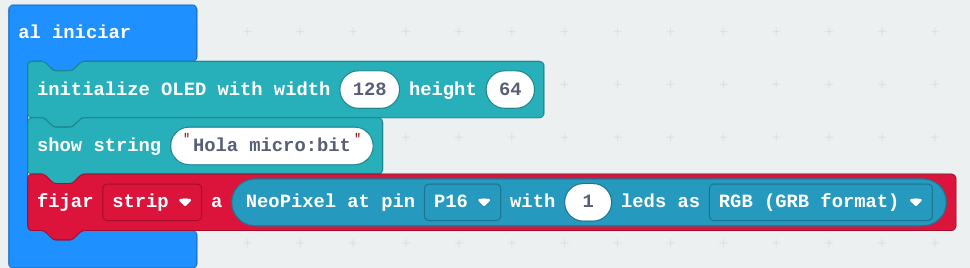
* Vamos a comparar la temperatura con un valor dado:
  + Si es superior encendemos el led rojo
  + Si es menor lo encendemos verde
* Abrimos los componentes de la paleta lógica y cojemos “Si verdadero…si no” y el operador “<” que pondremos sobre el hexágono “verdadero”



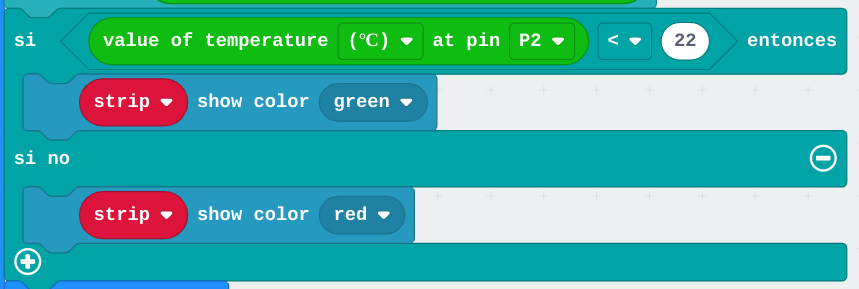
* Ponemos otro bloque de “value of temperature …” (Recuerda cambiar el pin a P2) y asignamos una temperatura umbral
* Abrimos la paleta neopixel para controlar nuestro led RGB ¿qué significa RGB?



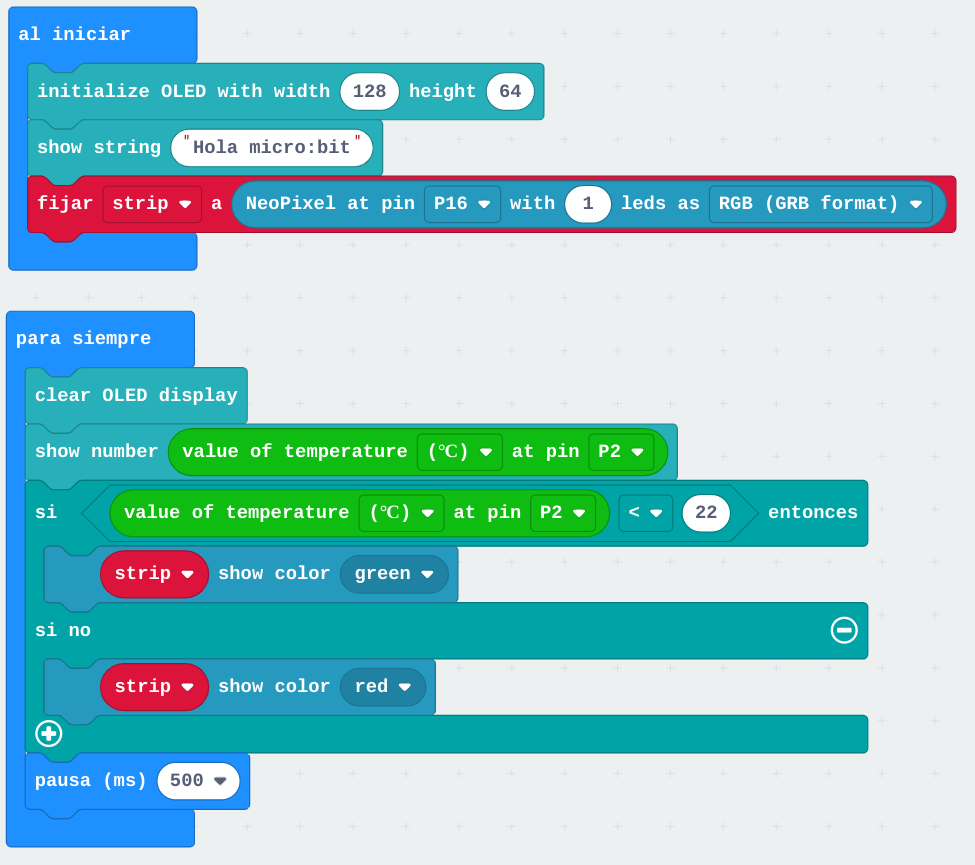
* Añadimos un bloque de configuración del neopixel en la parte de configuración en “al iniciar”



* Añadimos un bloque “show color red” y otro “show color green” dentro del bloque “Si…si no”

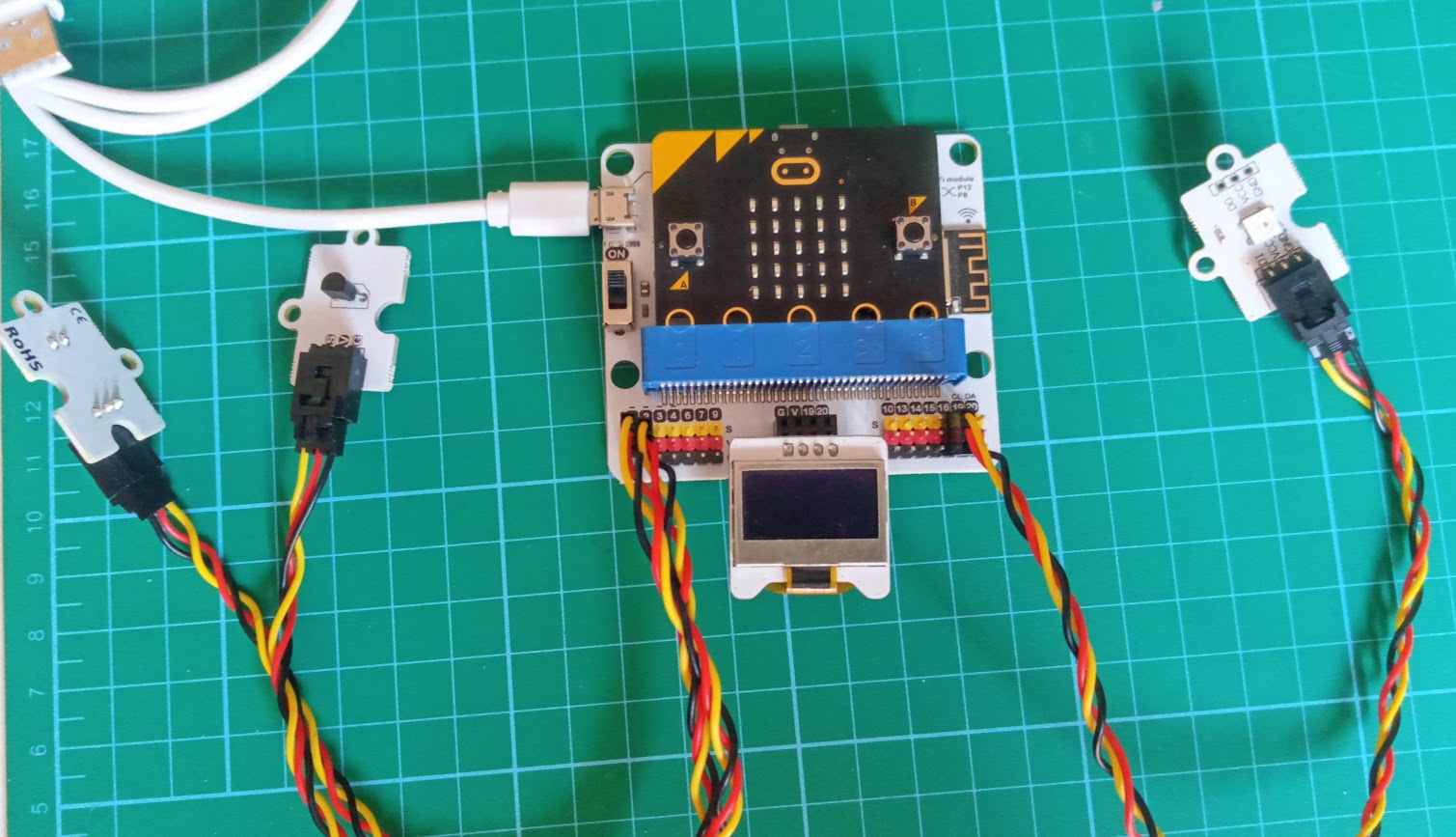


El programa completo queda así:



Reprogramamos nuestra micro:bit y cuando calentemos el sensor de temperatura el led cambiará de color

Para que funcione bien cambiamos el cable USB para que alimente la placa IOT:bit

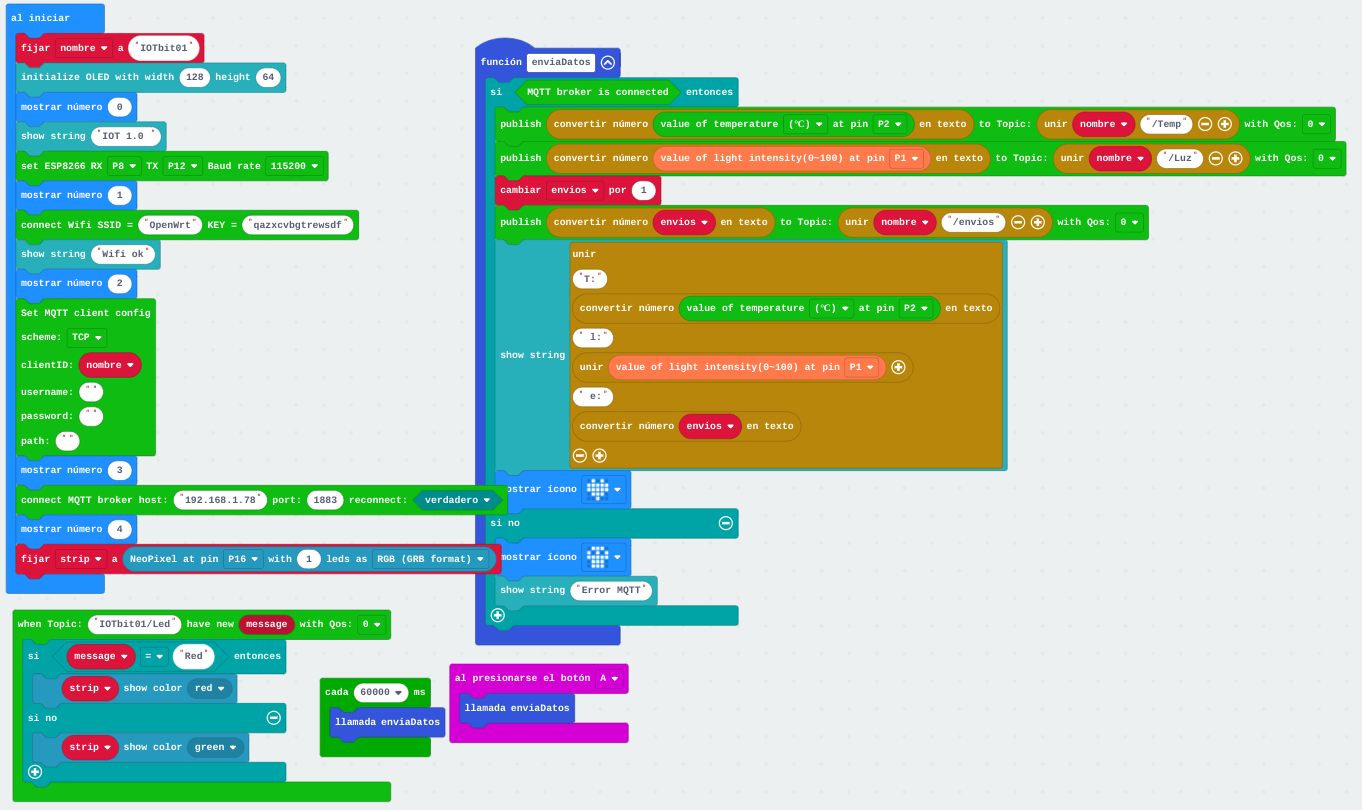


[Proyecto: Termostato](https://makecode.microbit.org/_HXfaVfRHT73b)

# Programa IOT

Con el montaje actual vamos a usar un programa ya más elaborado que hace muchas cosas:

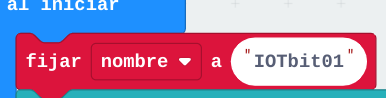
* Conecta a internet (wifi)
* Conecta con el servidor (Raspberry)
* Mide la temperatura y la luz
* Envía los datos al servidor
* Espera órdenes del servidor
* Si el servidor recibe una temperatura menor que 22 le ordena que ponga el led verde y si es mayor rojo



Este trasiego de datos y órdenes es lo que llamamos IOT

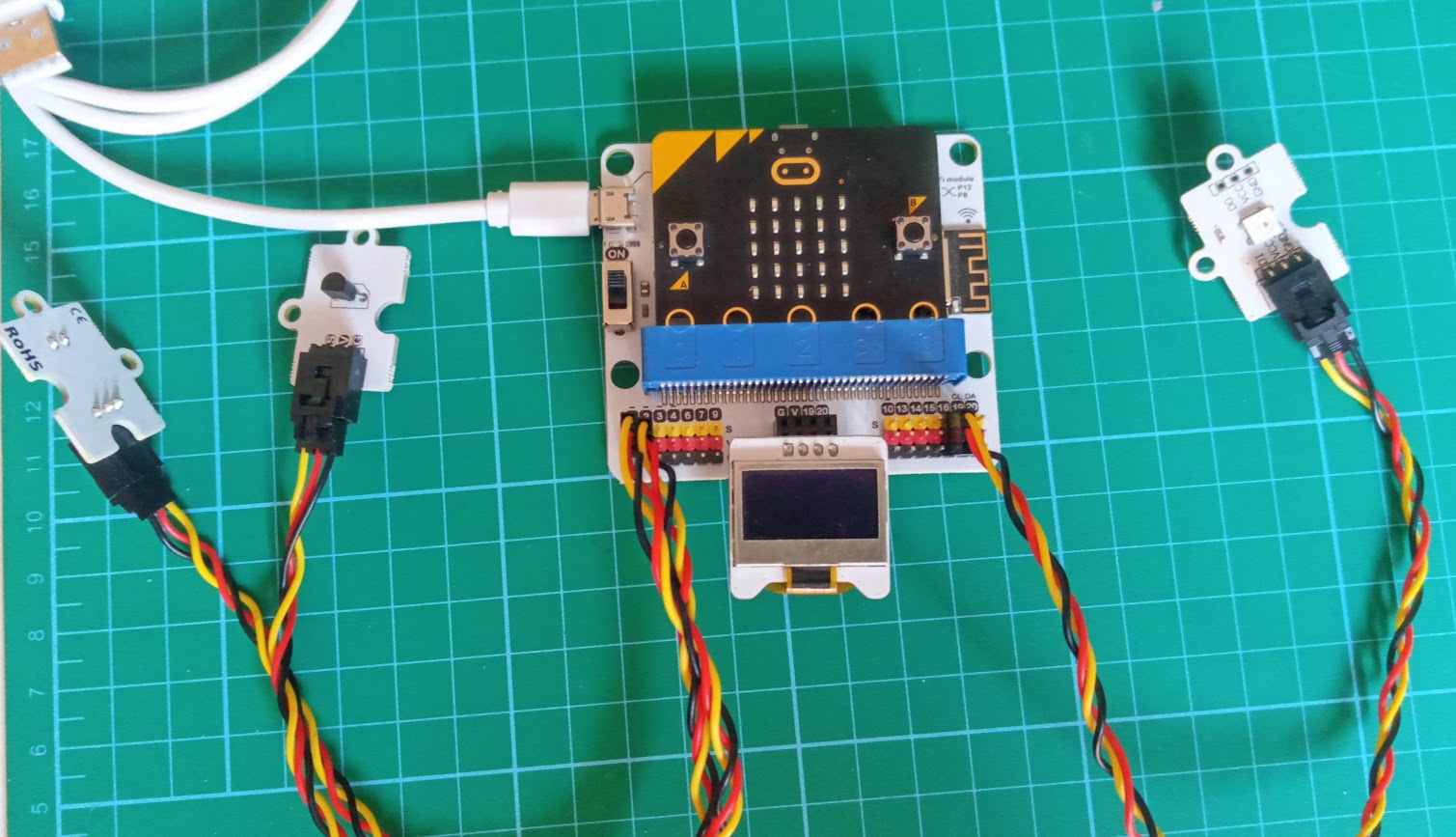
[Programa IOT](https://makecode.microbit.org/_63RcvY9U55YD)

* Cambia el nombre de tu dispositivo con tu número



Programa tu placa (recuerda que para programarla necesitas conectarle el USB a la micro:bit)

Para que funcione bien cambiamos el cable USB para que alimente la placa IOT:bit



# Raspberry Pi



Raspberry Pi ordenador en una sola placa, de pequeño tamaño y bajo costo, que se ha diseñado para ser utilizada en proyectos de enseñanza de la informática y la programación. Aunque es pequeña, Raspberry Pi cuenta con un procesador potente y puede ser utilizada para muchas de las mismas tareas que una computadora de escritorio o portátil tradicional.

Raspberry Pi se ha utilizado ampliamente en proyectos de robótica y para crear aplicaciones para la Internet de las cosas (IoT). También se ha utilizado para construir sistemas embebidos y media centers, y se ha utilizado en proyectos de ensamblaje de computadoras y en la enseñanza de la programación.

Raspberry Pi es una placa de computadora muy versátil y puede ser utilizada en una amplia variedad de proyectos y aplicaciones. Es especialmente popular entre los aficionados a la tecnología y los educadores debido a su bajo costo y su capacidad para ser utilizada en proyectos de enseñanza.

## Software

100% Open Source

* Broker de conexiones (MQTT): mosquitto
* Servidor gráfico: Grafana
* Base de datos: InfluxDB
* Programación de flujos: NodeRed

## Detalles técnicos

La Raspberry Pi 4 es la última versión de la placa de computadora de bajo costo de Raspberry Pi. Algunas de sus características técnicas incluyen:

Procesador: La Raspberry Pi 4 viene en tres versiones diferentes, con procesadores de cuatro núcleos Cortex-A72 a 1,5 GHz, Cortex-A72 a 1,8 GHz o Cortex-A72 a 2,0 GHz.

Memoria: La Raspberry Pi 4 viene en tres versiones diferentes, con 1 GB, 2 GB o 4 GB de RAM LPDDR4.

Almacenamiento: La Raspberry Pi 4 viene con un slot para tarjeta microSD para almacenamiento masivo.

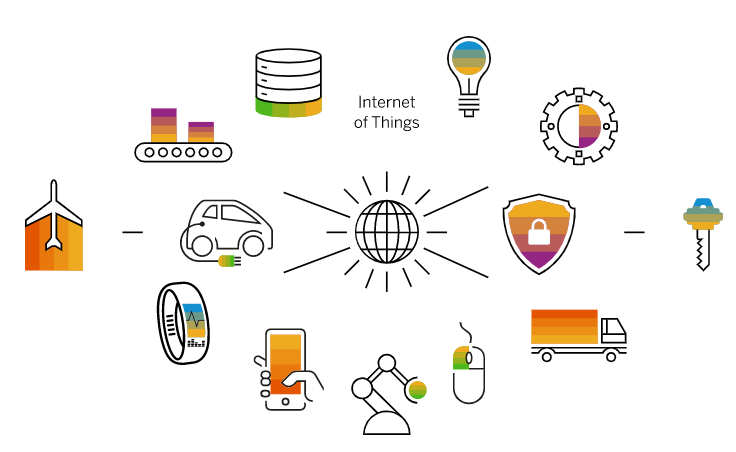
Conectividad: La Raspberry Pi 4 viene con conectividad Ethernet Gigabit y Wi-Fi 5 (IEEE 802.11ac). También tiene Bluetooth 5.0.

Puertos: La Raspberry Pi 4 tiene dos puertos HDMI de alta definición, cuatro puertos USB 3.0, un puerto Ethernet Gigabit y un puerto de carga USB-C para alimentación. También tiene una salida de audio de 3,5 mm y una salida de védeo compuesto.

Dimensiones: La Raspberry Pi 4 mide 85 x 56 x 17 mm.

Sistema operativo: La Raspberry Pi 4 es compatible con varios sistemas operativos, incluyendo Raspbian (un sistema operativo basado en Debian especialmente diseñado para Raspberry Pi), así como otras distribuciones de Linux y sistemas operativos basados en Unix. También es compatible con Windows 10 IoT.

# ¿Qué es IOT?



La Internet de las cosas (IoT, por sus siglas en inglés) se refiere a la conectividad de dispositivos cotidianos a Internet, permitiendo que estos dispositivos puedan enviar y recibir datos y puedan ser controlados a través de Internet. Estos dispositivos pueden incluir desde electrodomésticos comunes como termostatos inteligentes y aspersores de jardín hasta dispositivos industriales como sensores de monitoreo y equipos de producción.

La IoT tiene el potencial de mejorar la eficiencia y la productividad al permitir que los dispositivos se comuniquen y se controlen automáticamente sin la intervención humana. También puede proporcionar una mayor cantidad de datos y análisis para tomar decisiones informadas y mejorar la toma de decisiones en una amplia variedad de aplicaciones.

A medida que más dispositivos se conectan a Internet, la IoT está creciendo rápidamente y está teniendo un impacto cada vez mayor en la forma en que vivimos y trabajamos.

# Big Data, IOT & IA

Los dispositivos IOT generan enormes cantidades de datos. Estos flujos de datos han de ser convertidos en información entendible y aprovechable

El IoT se despliega en forma de millones de cámaras, grabadoras, ordenadores, móviles, radares, sensores, drones, códigos, termómetros, higrómetros, etc., que toman registros de lo que sucede y los envían a los centros de procesamiento. Sensores y dispositivos IoT de recopilación de datos son instalados en fábricas para generar una gran cantidad de datos no estructurados que se recopilan en el sistema de Big Data. Este último depende en gran medida de sus 3Vs principales: volumen, velocidad y variedad, otros las amplían a 5Vs, añadiendo la veracidad de los datos y el valor que aportan al negocio.

En el sistema de Big Data, que es una base de datos distribuida compartida, se almacenan los distintos tipos de datos que residen en una especie de lago de datos formado por archivos de Big Data.

Se analizan los macro datos de IoT almacenados mediante herramientas analíticas, que generan informes y gráficos de resultados.

Podemos ver Big IoT Data Analytics como una caja de herramientas de la que sacaremos una herramienta u otra en función del tipo de información y conocimiento que queramos adquirir de los datos. Muchas de estas herramientas son algoritmos tradicionales, o mejoras o adaptaciones de los mismos, con principios estadísticos y algebraicos muy similares.

El Big Data está generado por humanos. Los dos se retroalimentan a través de métricas adicionales, configuraciones y personalizaciones.

## Aplicación

* Personalización de la experiencia de usuario: muy en boga en el mundo del marketing y el contenido digital, se basa en plantear a cada individuo un recorrido distinto en función de su comportamiento previo.
* Smart Cities: reorganizar el tráfico, tomar medidas inmediatas para atajar la contaminación ambiental, reducir el despilfarro de agua, planificar mejor la política de urbanismo… Todo esto es posible en las ciudades que han sensorizado sus equipamientos y que sacan partido a los datos.
* Mantenimiento predictivo de la maquinaria: un enfoque Big Data es capaz de monitorizar el desgaste una máquina y detectar las señales que, recogidas a través del IoT, anticipan fallos inminentes en el sistema.
* Medicina preventiva: la adopción de los wearables abre un gran abanico de posibilidades para el fomento de hábitos saludables y el diagnóstico precoz de patologías latentes.
* Dispositivos “ponibles” o wearables: cualquier objeto o ropa, como relojes o gafas, en las que se incluyan sensores para mejorar sus funcionalidades.
* Dispositivos cuantificadores de la actividad de la persona: cualquier aparato diseñado para ser utilizado por aquellos que quieran almacenar o monitorizar datos sobre sus hábitos o estilo de vida.
* Dispositivos para domótica: cualquier aparato que permita controlar o alterar de manera remota por internet algún objeto, o que contenga sensores para detectar el movimiento, o permitir sistemas de identificación u otras medidas de seguridad en un espacio cerrado.
* Dispositivos industriales: cualquier aparato que permita convertir variables físicas (como temperatura, presión, humedad etc.) en señales eléctricas o de otro tipo. Son los sensores o actuadores industriales:
  + Una batería formada por miles de células de batería
  + Estudio de averías en flotas de vehículos
  + Google reduce 40% su gasto en refrigeración de sus servidores
  + Analizar los datos de los wereables para estimar el riesgo de estrés por calor que pueden sufrir los trabajadores de las fábricas

# AIOT

AIoT: la inteligencia artificial de las cosas

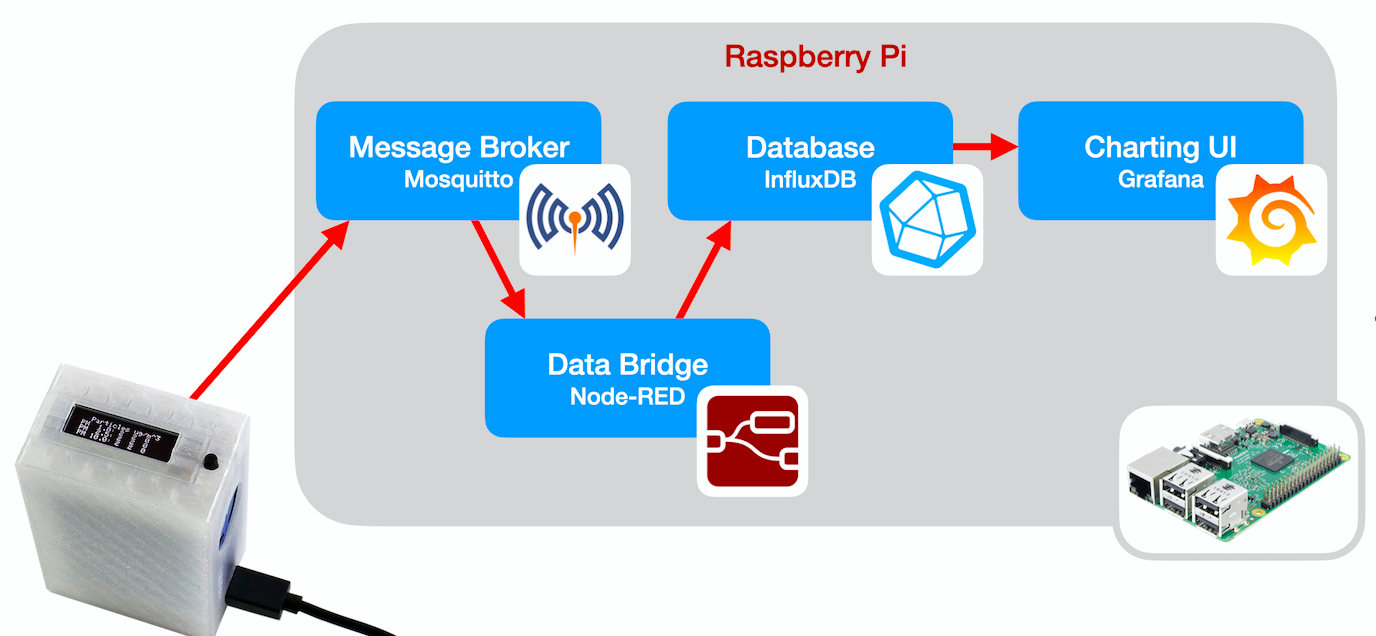
Son sistemas inteligentes y conectados que son capaces de tomar decisiones por sí mismos, evaluar los resultados de estas decisiones y mejorar con el tiempo.

El análisis basado en el aprendizaje automático permite identificar patrones automáticamente y detectar anomalías en los datos que generan los sensores y los dispositivos inteligentes.

1. Por un lado podríamos seguir hablando de ese cerebro como un sistema centralizado que procesa todos los impulsos y toma decisiones. En este caso nos estaríamos refiriendo a un sistema en la nube que recibe de forma centralizada toda la telemetría y actúa. Sería la IA en la nube (Inteligencia Artificial en la Nube).
2. Por otro lado, también debemos hablar de una parte muy importante del sistema nervioso: los reflejos. Los reflejos son decisiones autónomas que el sistema nervioso toma sin necesidad de enviar toda la información al procesador central (el cerebro). Estas decisiones se toman en la periferia, cerca de la fuente donde se originaron los datos. Es lo que se denomina Edge AI (Inteligencia Artificial en la Periferia).

# Arquitectura IOT

Para que la raspberry pi actúe como un servidor de IOT, necesitamos que estén instalados y funcionando al menos 4 servicios, que reciben, formatean, almacenan y muestran los datos



* Mosquitto - servidor (Broker) MQTT al que se conectan los dispositivos remotos usando el protocolo MQTT. Es el servidor al que nos conectamos, en el programa pondremos los datos de conexión (wifi), IP y puerto del servidor. También necesitamos especificar el protocolo MQTT y por eso usamos bloques para este tipo de conexión
* Node-RED - servicio que convierte los datos recibidos desde MQTT en un formato adecuado para procesarlos, convirtiendo los datos en valores decimales y añadiéndoles fecha y hora de recepción. Una vez convertidos los inserta en la base de datos. Necesitamos establecer un
* InfluxDB - base de datos donde se guardan los datos recibidos, preparándolos para que se puedan procesar, filtrar, generar acumulados, promedios…
* Grafana - sistema de visualización de datos que nos permite filtrar, agregar y mostrar gráficamente

## Programa IOT micro:bit

[Programa IOT micro:bit https://makecode.microbit.org/\_63RcvY9U55YD](https://makecode.microbit.org/_63RcvY9U55YD)

## Mosquitto

Configuración Wifi para conectividad: SSID y contraseña Configuración broker MQTT: IP y puerto

Cada micro:bit tiene un nombre distinto para poder distinguir los datos de cada uno. Los datos que se envía van etiquetados con ese nombre

Veremos en la pantalla OLED los datos que se envían

Si estamos habituados a trabajar con la raspberry Pi, podemos comprobar que los datos que llegan al servidor con el comando

mosquitto\_sub -h 192.168.1.100 -t "#" -d

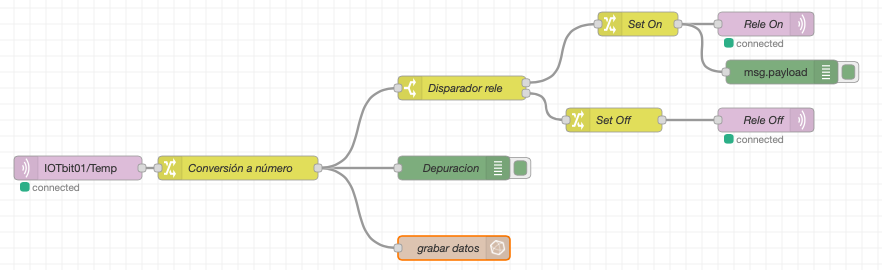
y veremos los datos que llegan

Client mosqsub|18752-raspi4 received PUBLISH (d0, q0, r0, m0, 'IOTbit01/Temp', ... (2 bytes))  
24  
Client mosqsub|18752-raspi4 received PUBLISH (d0, q0, r0, m0, 'IOTbit01/rele', ... (2 bytes))  
On  
Client mosqsub|18752-raspi4 received PUBLISH (d0, q0, r0, m0, 'IOTbit01/Luz', ... (1 bytes))  
8  
Client mosqsub|18752-raspi4 received PUBLISH (d0, q0, r0, m0, 'IOTbit01/envios', ... (2 bytes))

## Node-Red

Flujos para transformar los datos que envía la micro:bit en datos numéricos, etiquetándolos con fecha y hora

http://raspiIP:1880



Cuando se recibe un dato, se filtra por el nombre, se convierte a número y por un lado se guarda en la base de datos y por otro se comprueba si la temperatura es mayor o menor de 22 grados se envía una orden para hacer que el led se ilumine en rojo o en verde

Por cada placa micro:bit, con un nombre distinto tenemos que crear una copia de esta regla con la etiqueta adecuada según el nombre de la placa tanto en la recepción de datos como en el guardado de datos en la base de datos.

Podemos copiar todo el flujo con Ctrl+C y hacer una copia con Ctrl+V

En el nuevo cambiamos la etiqueta del paquete qu se recibe y en el nodo de grabación de datos

Hacemos “Deploy”

## InfluxDB

Datos guardados en formato de series temporales para facilitar su recuperación

## Grafana

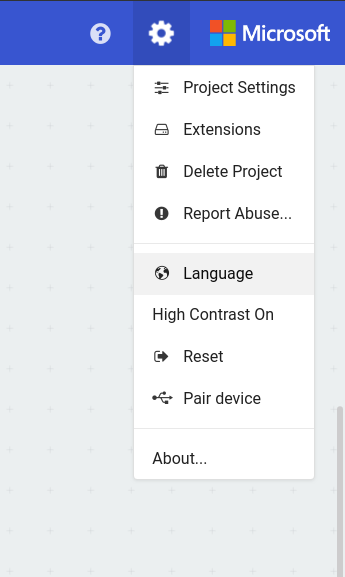
Creamos cada gráfica que queremos mostrar. Filtramos los datos según lo que queramos mostrar. Para ello tenemos que haber recibido algunos datos.

http://raspiIP:3000 con usuario/contraseña admin/ParqueCiencias55

[Vídeo resumen](https://youtu.be/yV0yYC0gze0) [](https://youtu.be/yV0yYC0gze0)

# Preguntas frecuentes

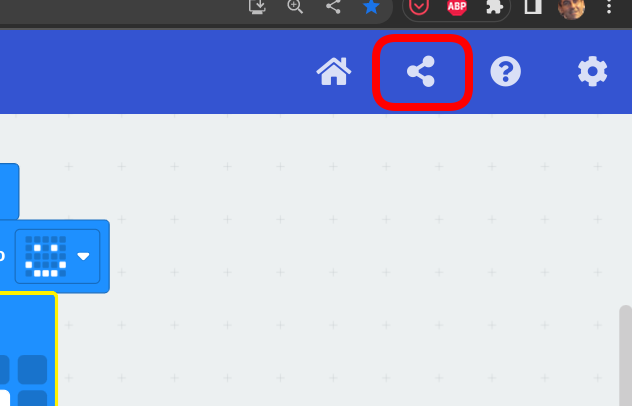
* Si queremos cambiar el idioma del entorno



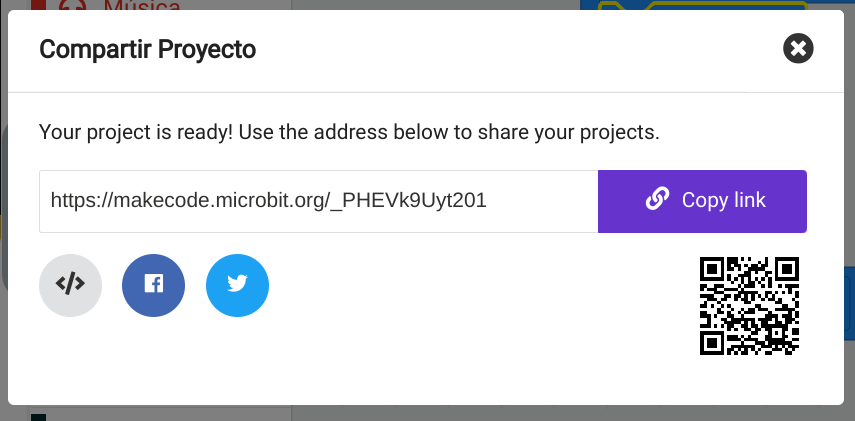
* Podemos cambiar el nombre del programa editándolo en la caja de abajo.



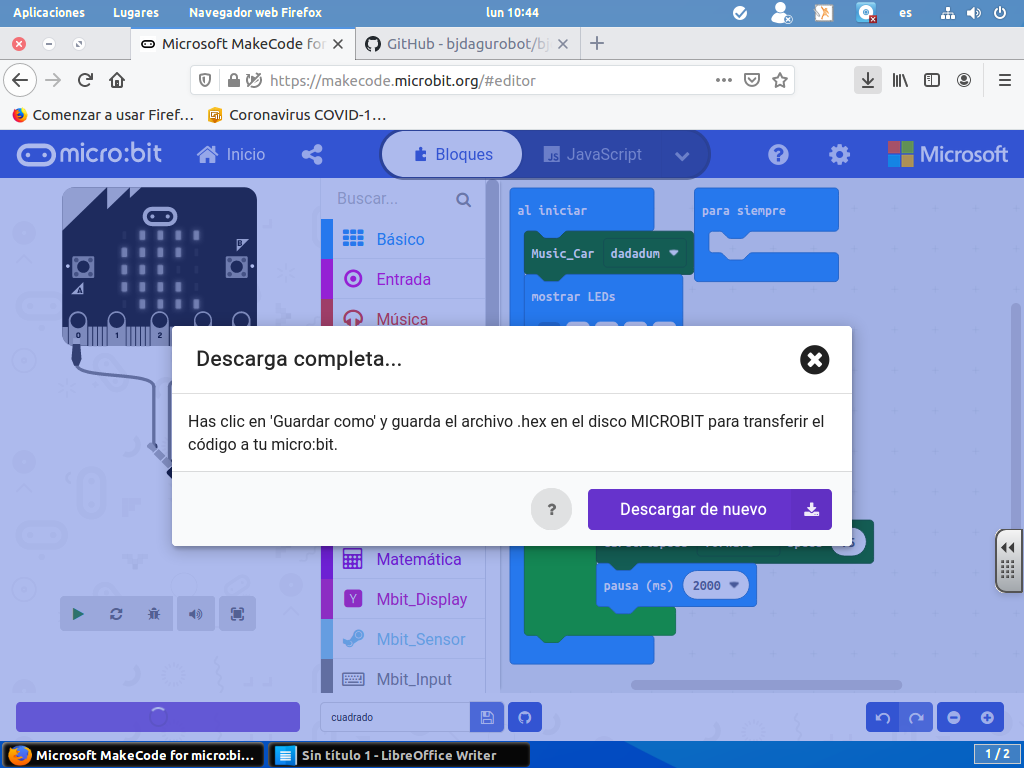
* Si quiero compartir mi programa o me lo quiero “llevar” puedes pulsa el icono compartir



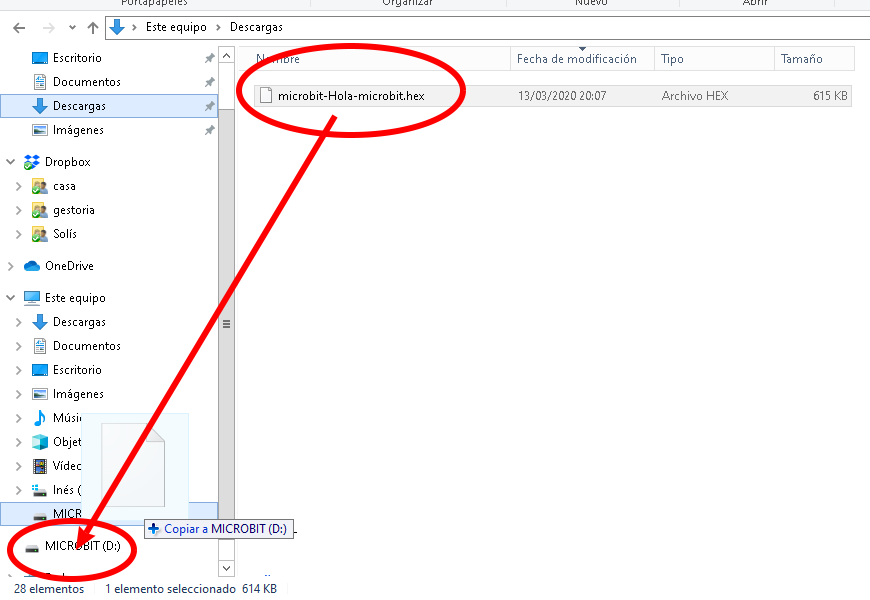
Y en la ventana que aparece podemos copiar el enlace



* Si al intentar programar la micro:bit no conecta:
  + Revisa el cable USB. ¿es un cable de datos? ¿no será un cable de carga?
  + El cable tiene que estar conectado a la micro:bit
  + Desconecta y Conecta
* Si no funciona el emparejamiento, al conecta la micro:bit veremos que aparece en el ordenador como si hubiéramos conectado un USB llamado “Microbit”
  + Descargamos el fichero del programa



\* Copiamos el fichero al "pen drive microbit"



# Instalación

Raspberry Pi OS 64

usuario/contraseña de la Raspberry Pi pi/ParqueCiencias55

## ssh y VNC

Activamos el acceso remoto con ssh y VNC

## Actualización

sudo apt update  
sudo apt full-upgrade  
sudo apt autoclean  
sudo apt autoremove

Cambiamos el idioma y la configuración de teclado

## Software

### MQTT broker - mosquitto

sudo apt install mosquitto mosquitto-clients

Editamos la configuracion **/etc/mosquitto/mosquitto.conf**

port 1883  
  
allow\_anonymous true

(http://www.steves-internet-guide.com/mossquitto-conf-file/)

#### test

mosquitto\_sub -t "#" -v  
  
# (desde otra maquina)  
  
mosquitto\_pub -h raspiIP -t '/test' -m 'hola'

Cambiando raspiIP por la IP de nuestra Raspberry Pi

# micro:bit

Emparejamos micro:bit

[Programa IOT](#programa-iot)https://makecode.microbit.org/\_6FhDDqfjR7Xr)

Luz - 1  
TMP - 2  
LED - 16

Cada uno tiene que cambiar su número en el nombre de su microbit

## node-red

instalacion

bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

Activamos como servicio

sudo systemctl enable nodered.service

Arrancamos

sudo systemctl start nodered.service

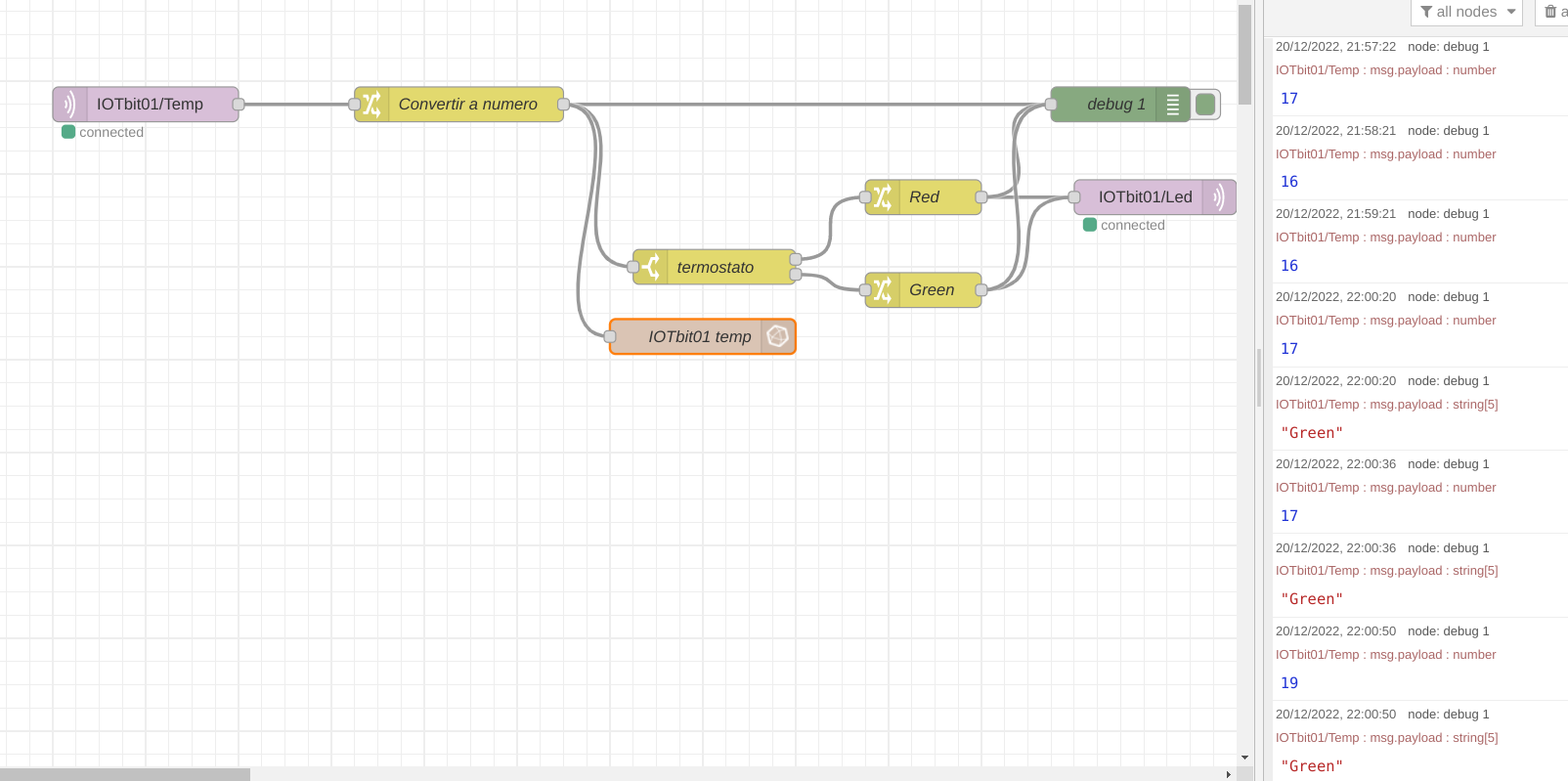
Accedemos desde un navegador

http://raspiIP:1880

con usuario/contraseña pi/ParqueCiencias55

# Flujo

* Damos de alta el servidor MQTT
* Añadimos la paleta de influxDB
* Damos de alta el servidor InfluxDB



## influxDB

sudo apt install influxdb influxdb-clients  
influxdb  
  
>> create database IOTPC

# Grafana

## instalacion grafana

curl https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /usr/share/keyrings/grafana-archive-keyrings.gpg >/dev/null  
  
echo "deb [signed-by=/usr/share/keyrings/grafana-archive-keyrings.gpg] https://apt.grafana.com stable main" | sudo tee /etc/apt/sources.list.d/grafana.list  
  
sudo apt update   
  
sudo apt install grafana  
  
## admin/ParqueCiencias55  
  
sudo systemctl enable grafana-server.service  
  
sudo systemctl start grafana-server.service

Podemos acceder con usuario/contraseña admin/ParqueCiencias55

http://raspiIP:3000

(Cambiando raspiIP por la IP de la Raspberry Pi)

## Configuración de vistas

* add influxdb como Datasource
* Add dashboard y select measurement

sudo /bin/systemctl enable grafana-server

sudo /bin/systemctl start grafana-server