

STEM EN TU CASA:

Bienvenido y bienvenida a hacer posible lo imposible



SAMSUNG



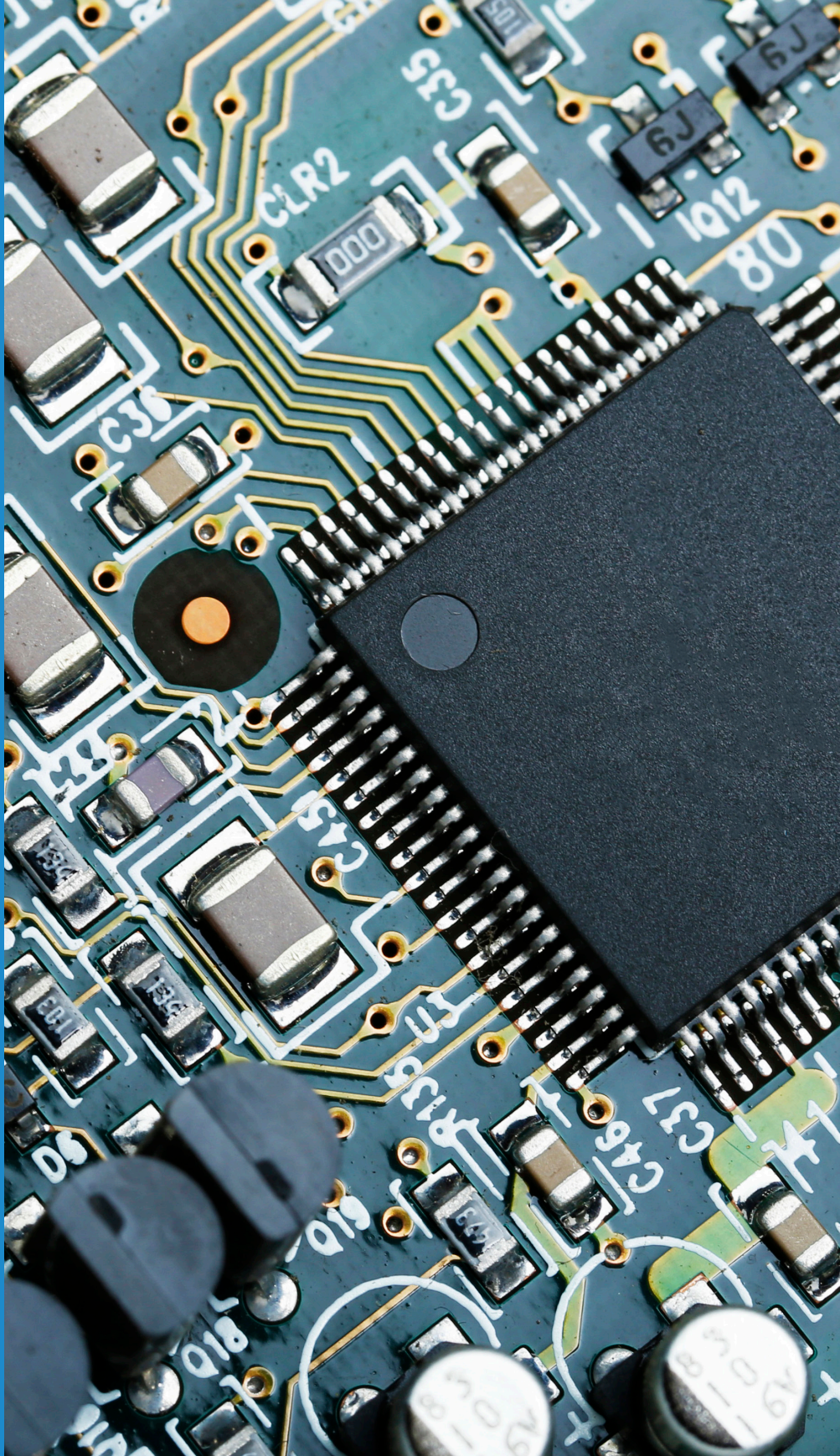
Asignatura: Educación Tecnológica

Curso: 7° y 8 básico

Edad: 12 a 14 años

ÍNDICE

INTRODUCCIÓN	4
ACTIVIDAD 1	6
Procedimiento	6
Componentes	6
ACTIVIDAD 2	7
ACTIVIDAD 3	12
ACTIVIDAD 4	15
ACTIVIDAD 5	17
CONCLUSIONES	19



OBJETIVO APRENDIZAJE MINEDUC:

TE 08 OA01: Identificar oportunidades o necesidades personales, grupales o locales que impliquen la creación de un producto tecnológico, reflexionando acerca de sus posibles aportes.

TE 08 OA 02: Diseñar y crear un producto tecnológico que atienda a la oportunidad o necesidad establecida, respetando criterios de eficiencia y sustentabilidad, y utilizando herramientas TIC en distintas etapas del proceso.

TE 08 OA03: Evaluar el producto tecnológico creado, aplicando criterios propios y técnicos, y proponer mejoras asociadas tanto a los procesos como al producto final.

El Kit con el que trabajaremos se compone de lo siguiente:



1 placa electrónica Dario S1



1 cable USB



1 tarjeta de bienvenida



1 guía de apoyo al aprendizaje.

Necesitarás además:



1 computador de escritorio, o un notebook (laptop)



Conexión a internet, a través de wifi.



Conocer el nombre de la red wifi a la que te conectarás, y su contraseña.



1 lupa.

INDICADORES DE APRENDIZAJE DE LA LECCIÓN

- Seleccionan información recogida en la investigación de oportunidades que permita diseñar un producto tecnológico respetando criterios de sustentabilidad.
- Diagraman conceptualmente por medio de herramientas gráficas un producto tecnológico que responda a la necesidad establecida, respetando criterios de sustentabilidad.
- Usan herramientas de comunicación en línea para testear las ideas conceptuales de sus diseños y los criterios de sustentabilidad propuestos.

Precauciones

- *Lee atentamente las instrucciones antes de hacer cualquier actividad.*
- *Sigue las instrucciones dadas.*
- *La placa será energizada con electricidad durante el trabajo. No debe mojarse, ni ponerse en superficies húmedas.*
- *No poner la placa sobre superficies metálicas, como carcasas de laptop, o escritorios.*
- *Al mantener conectada tu placa Dario S1 al computador, asegúrate de que no se recaliente, ni huela a plástico quemado. Si esto ocurre, debes desconectarla de inmediato, y esperar a que se enfríe.*

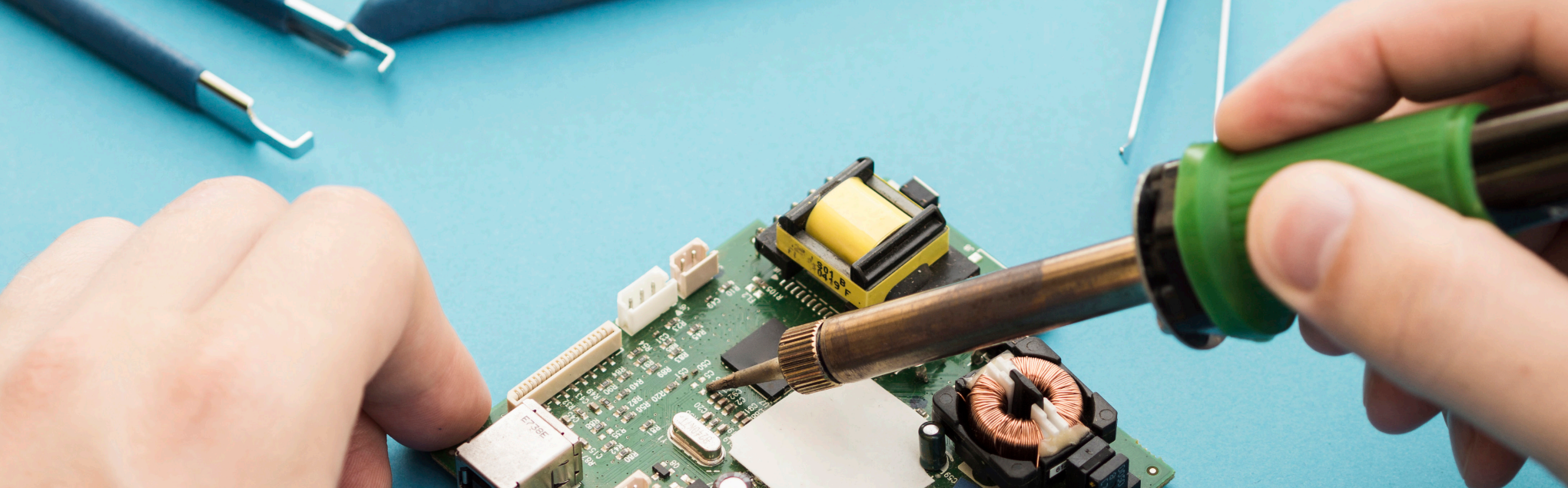


INTRODUCCIÓN

STEM es un término que conjuga la Ciencia, la Tecnología, la Ingeniería y las Matemáticas, en la creación de soluciones a necesidades y problemas de nuestro entorno, de maneras prácticas (experimentando) y colaborativas. Su objetivo es acercar la educación a las estrategias que se usan en el mundo real para inventar nuevas tecnologías, o ponerlas al servicio de la sociedad. El programa STEM EN TU CASA, al cual te invitamos a participar, pretende que detectes en tu entorno doméstico (ej. habitación, patio de tu casa) alguna necesidad o problema, al cual le puedas crear una solución usando un tipo de tecnología llamada Internet de las Cosas (de ahora en adelante lo abreviamos como IoT, derivado de la frase en inglés “Internet of Things”)

El propósito de IoT es mejorar la funcionalidad de objetos eléctricos y electrónicos, como electrodomésticos, televisores, computadores, cámaras, sensores, entre otros, permitiendo que se comuniquen entre sí a través de internet, para monitorear su funcionamiento, dar alertas, o modificar su funcionamiento (ej. apagando o encendiendo).

Si tuviéramos que dar una definición de IoT probablemente lo mejor sería decir que se trata de una red que interconecta objetos físicos valiéndose del Internet. Los objetos que se conectan se valen de sistemas incluidos (como placas con circuitos integrados, y antenas wifi) que les permiten no solo la conectividad a Internet, sino que además programar eventos específicos en función de las tareas que le sean dadas de manera remota. Ejemplos de estas aplicaciones son chaquetas de bomberos que reportan su temperatura corporal a una plataforma de monitoreo, y alertar si están en peligro de sobrecalentar su cuerpo, sistemas de medición de radiación UV, que informan cuando es necesario cubrir la piel de bloqueador, o no exponerse al sol, o ampollitas que puedes encender y apagar en tu casa desde cualquier lugar que tenga señal de internet.

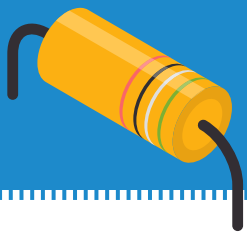


Para que inicies tu trabajo STEM, usando Internet de las Cosas, te invitamos a imaginar ideas a partir de la siguiente pregunta:

¿Qué necesidad o problema de mi entorno resolvería usando Internet de las Cosas?

Tómate un tiempo para imaginar tus ideas y responder a esta pregunta, desde lo que tú crees, no es necesario que sea una sola idea, ni que estés seguro o segura que sea la más correcta. En STEM, a estas respuestas que pueden ser posibles, le llamamos **HIPÓTESIS**. Científicos e ingenieros las formulan antes de llevar a cabo un diseño, o un experimento, con el propósito de guiar sus trabajos, y al final, comprobar si se cumplieron o no.

Para formular tu hipótesis puedes escribir tu idea, o la puedes representar de forma esquemática, mediante dibujos, flechas y explicaciones.



ACTIVIDAD 1

Reconocimiento de los componentes de la placa electrónica

PROCEDIMIENTO:

Nuestro primer paso para trabajar con IoT es aprender a dominar la programación que realizaremos sobre nuestra placa, usando el software de Arduino. Si te preguntas ¿qué es Arduino?, te contamos que es una plataforma de desarrollo tecnológico, basada en varios modelos de placas electrónicas de hardware libre, que incorpora un mini computador (o microcontrolador) re-programable y una serie de pines (conectores similares a enchufes) hembra, a los cuales puedes conectar sensores, y motores para construir, por ejemplo, un robot, o un sistema IoT. En nuestro caso, hemos desarrollado una placa especial para el que aprendas STEM en tu casa, llamada **Darío S1**, y que funciona con los softwares de la placa modelo **FireBeetle-ESP32**.

Antes de iniciar la programación, haremos un reconocimiento físico a los componentes que posee nuestra placa Darío S1:

La placa se energiza a través de la conexión de un cable con pin micro USB a la entrada USB1. Dicho cable puede estar conectado al computador, e inclusive, puede energizar estando conectado a un cargador de celular, o una batería externa para recargar celulares.

COMPONENTES:

Sensores: capturan información de variables medioambientales	Microcontrolador: Actúa guardando las programaciones, y procesando la información entregada por los sensores, para dar órdenes a los actuadores	Actuadores o efectores: componentes que responden según las órdenes entregadas por el microcontrolador
P T H : sensor de temperatura, presión atmosférica y altura geográfica. UV: Sensor de luz UV LUZ: sensor de luz visible	Espressif ESP32	RGB: Led rojo, verde y azul Sonido: bocina emisora de timbres.

Otros componentes:

On/Off	Switch encendido y apagado de la placa
RST	Tecla reinicio (reset)
Led 6	Informa flujo de información desde la placa hacia el computador.
Led 7	Informa flujo de información desde el computador hacia la placa.
F1	Fusible, impedirá que la placa se queme frente a una variación brusca de voltaje.
S1 y S2	Switch 1 y switch 2, para controlar actuadores que expandan funciones de la placa
Perilla (potenciómetro)	Permite regular de manera manual el control de algunos actuadores (sonido, RGB)
Antena WiFi	Ubicada al costado externo del microcontrolador, permite conectarse de manera inalámbrica a internet.



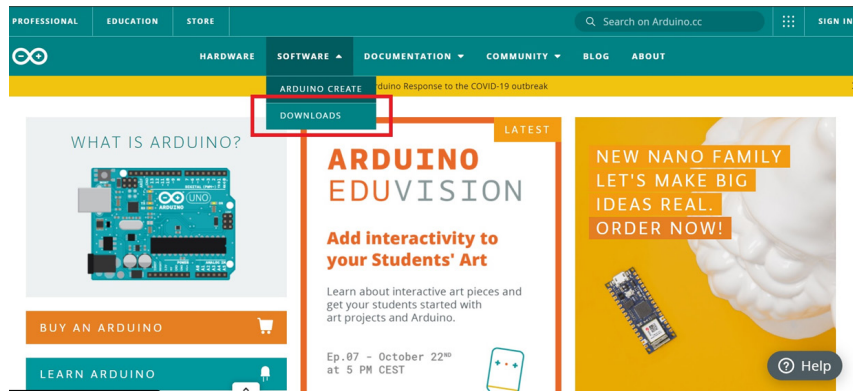
ACTIVIDAD 2

Instalación de software para programar placa electrónica

Para comenzar la programación de nuestra placa Darío s1, realizaremos cada uno de los siguientes pasos de manera ordenada.

1 En tu navegador de internet ingresa la siguiente URL:
<https://www.arduino.cc/>

2 Anda al link “software”, y en el submenú, selecciona la opción “downloads” (descargas)



3 Si es necesario baja un poco hasta *Download the Arduino IDE* (Descarga el IDE de Arduino)

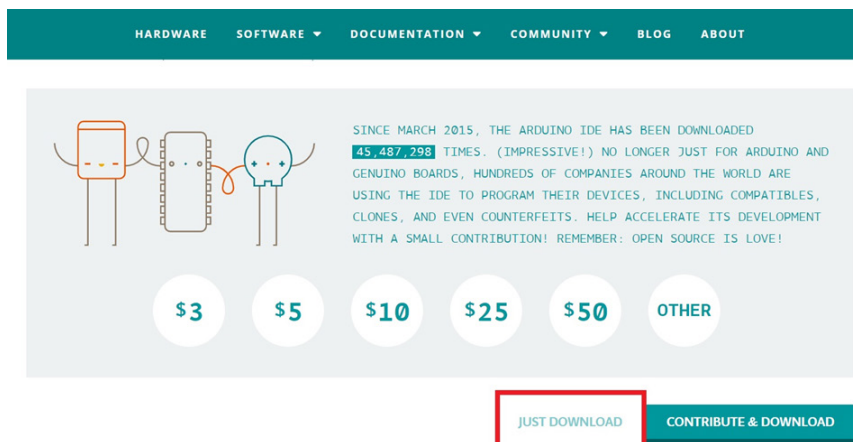
4 Ahora debes buscar la opción que corresponde al sistema operativo de tu computador (windows/, 8 o 10, Mac OS X, o Linux) y debes de seleccionar.

IMPORTANTE: Este software no funciona con tablets o celulares (sistema operativo Android), ni tampoco I pads o I phones.



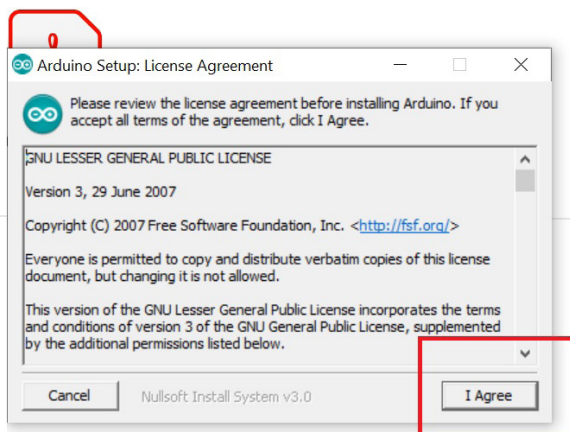
5 En la siguiente pantalla, dale clic en “just download” (sólo descargar)”

6 Te preguntará en qué carpeta vas a descargar el programa. Te sugerimos guardarlo en la carpeta "documentos".



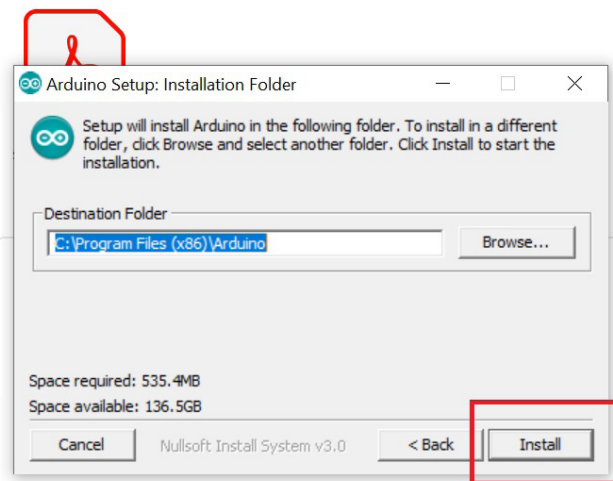
7 Luego abre el archivo que se descargó en tu pc (archivo arduino terminado en .exe)

8 Y acepta la licencia (I agree)

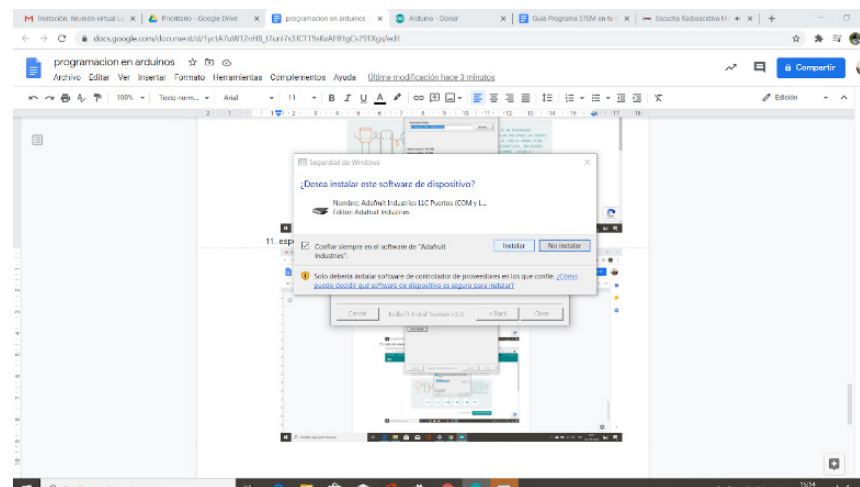


9 Debes dar clic en next o siguiente

10 Ahora dale clic en *install* (instalar), y espera mientras se instala el programa.

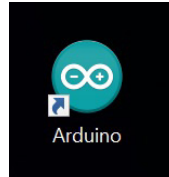


11 Una vez finalizado, vuelve a dar clic en instalar

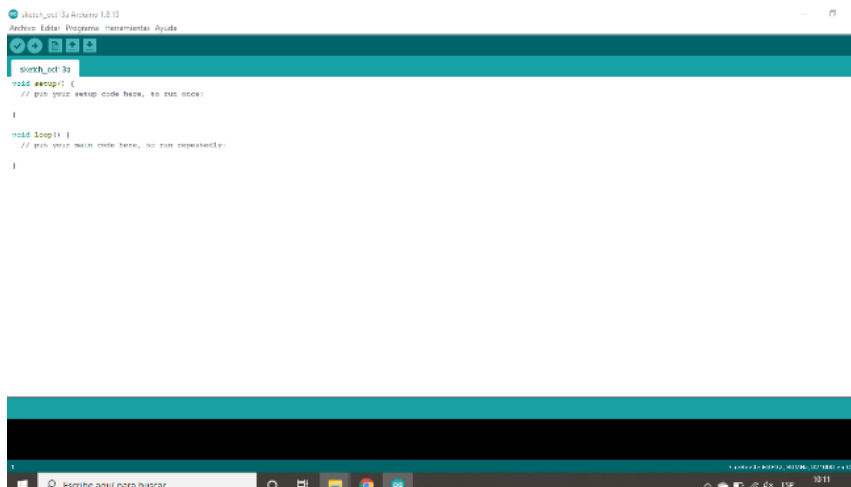


12 Finalmente dale clic en cerrar o *close*

13 En el escritorio de tu computador se debió instalar un acceso directo del programa Arduino.



14 Al abrir el programa, debe verse así:



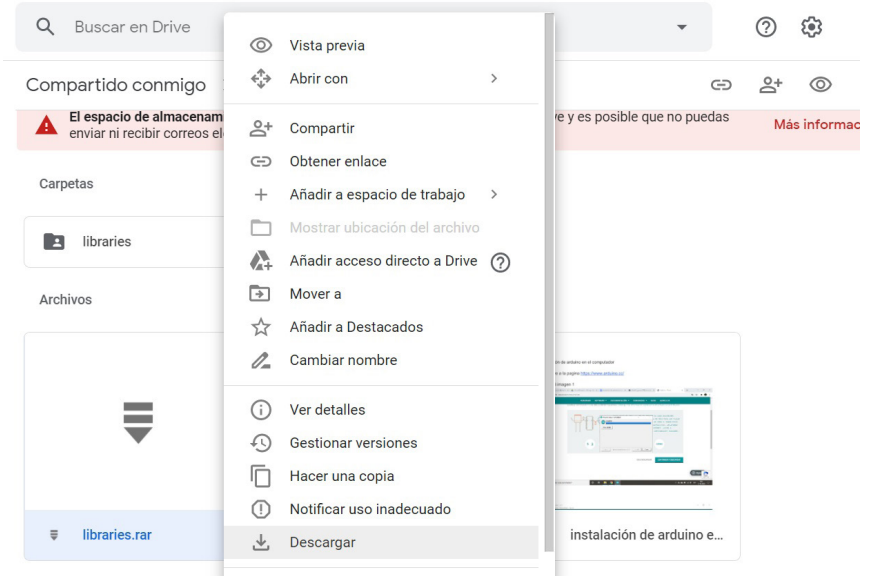
Con estos pasos ejecutados correctamente, has terminado la instalación del software que te permitirá insertar operaciones en tu placa Darío S1.

Ahora, continuaremos con la descarga en tu computador de las bibliotecas (libraries) y las programaciones, que te permitirán controlar cada uno de los sensores de tu placa, sus comunicaciones y los rangos de respuestas que pueden realizar.

15 En tu navegador de internet, inserta la siguiente URL:
https://drive.google.com/drive/u/1/folders/1MCBRrQBdeZ5qW_iOpPWv2Zy9fCRV0Bpf

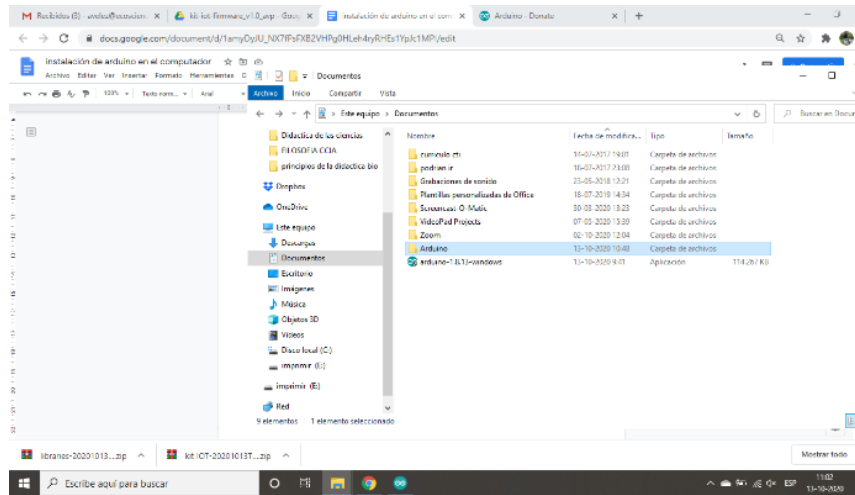
Si tienes el link disponible en un correo electrónico, o en tu plataforma classroom, bastará que le hagas click.

16 Descarga los dos archivos (kit-iot-firmware.rar y libraries.rar), parentando clic derecho del mouse sobre ellas, y poniendo "descargar". Debes guardarlas en la carpeta documentos, subcarpeta Arduino.

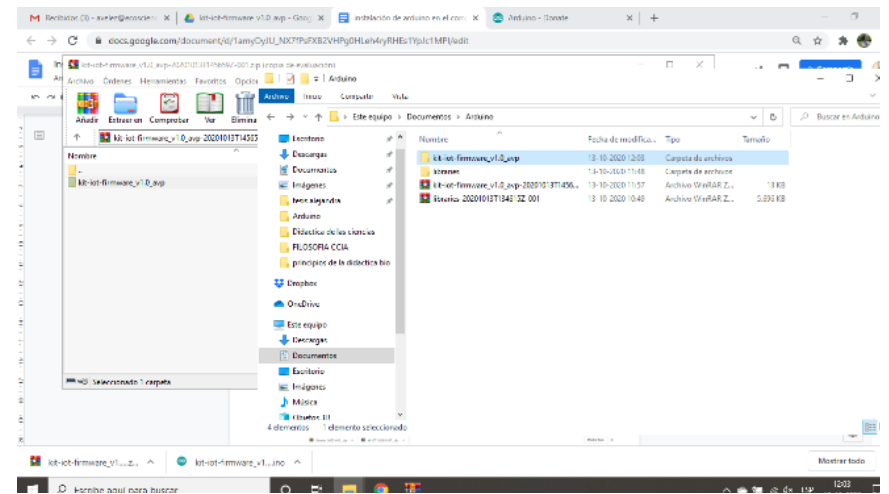
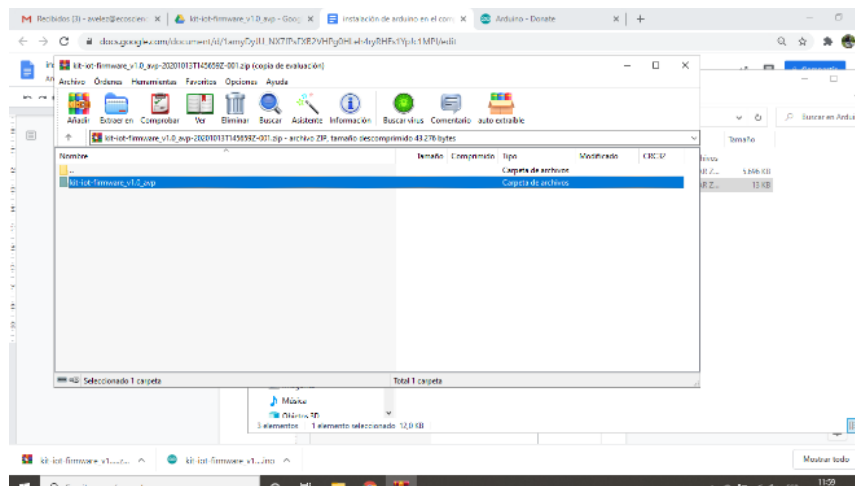


17 Te darás cuenta que se descargan como archivos comprimidos .rar.

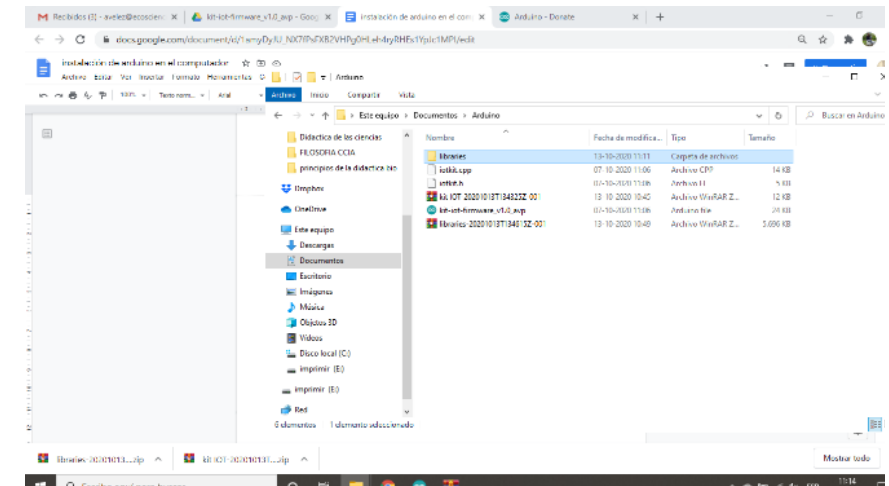
18 Ahora, anda a la carpeta documentos, y abre la carpeta **Arduino**. Notarás que ahí se encuentran los archivos .rar

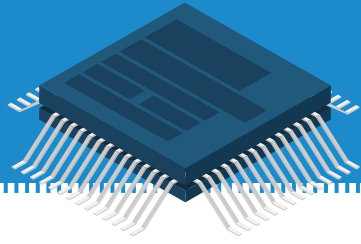


19 Abramos la primera **kit-iot-firmware_v1.0.2**. Copiamos y pegamos esta carpeta, dentro de la carpeta **Arduino**.



20 Realiza lo mismo con el archivo **libraries.rar**. Cuando lo abras, vuelve a abrir la carpeta **libraries** en su interior, selecciona todas las subcarpetas que tiene, ahora copialas con clic derecho del mouse, y luego las pegas dentro de la subcarpeta **libraries**, que se encuentra en la carpeta **Arduino**.





ACTIVIDAD 3

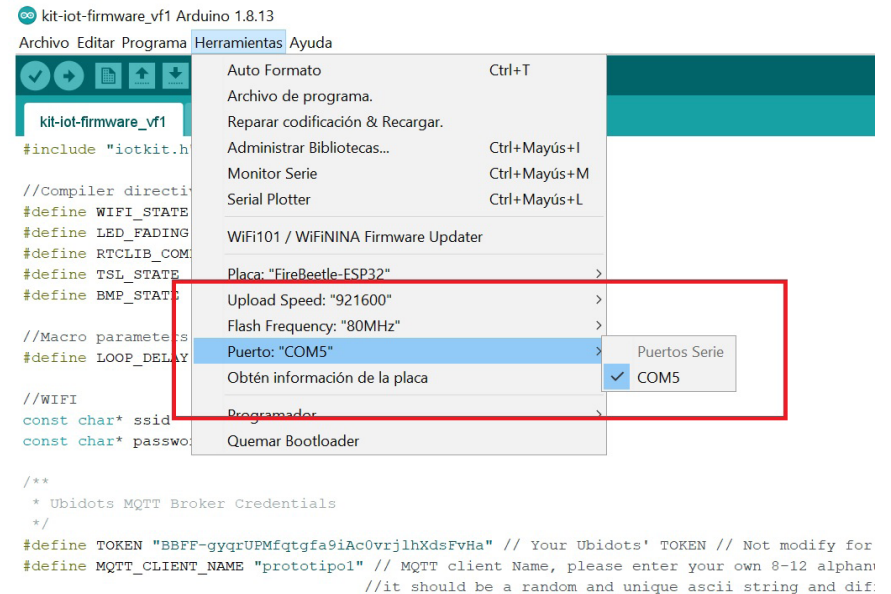
Habilitando la programación de nuestra placa electrónica

1 Saca tu placa Darío S1 de la caja, cuidando de no ponerla en una superficie metálica, o que esté mojada. Conecta el extremo micro USB del cable en la placa, y el extremo USB en el computador, como se ve en la imagen. Asegúrate que tu placa esté encendida, y que no se sobrecaliente, que lo notarás si emana un olor a plástico quemado. Si ocurriera esto, desconéctala rápidamente.

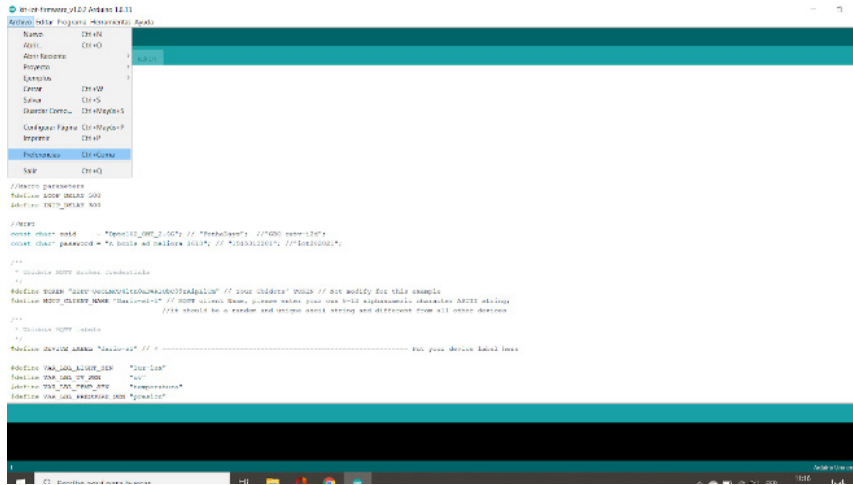


2 Verifica que no estés usando otro puerto USB en uso en tu computador, (por ejemplo, un pendrive puesto). En el programa Arduino selecciona: *herramientas*, y luego puerto. Ahí te podría figurar, por ejemplo COM3, u otro COM seguido de un número. Selecciona el que te haya aparecido.

De esta manera, habrás conectado tu placa Darío S1 al programa Arduino de tu computador.

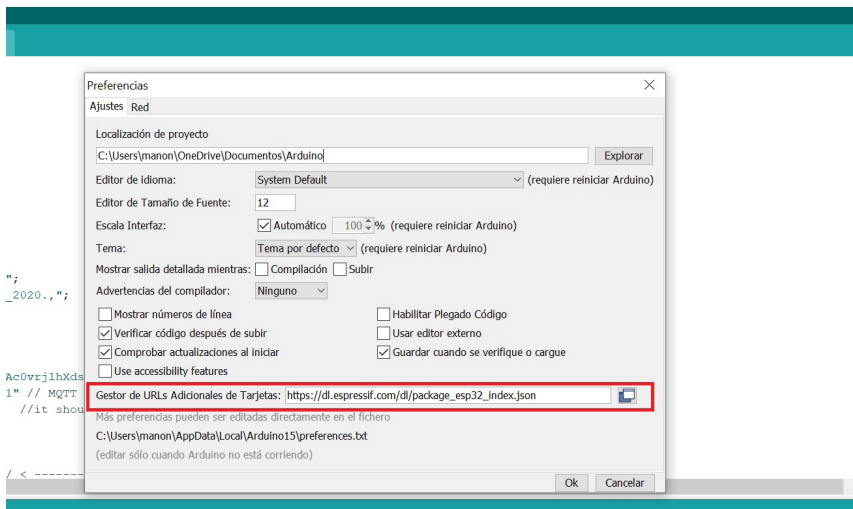


3 Ahora, anda a archivo, y luego *preferencias*

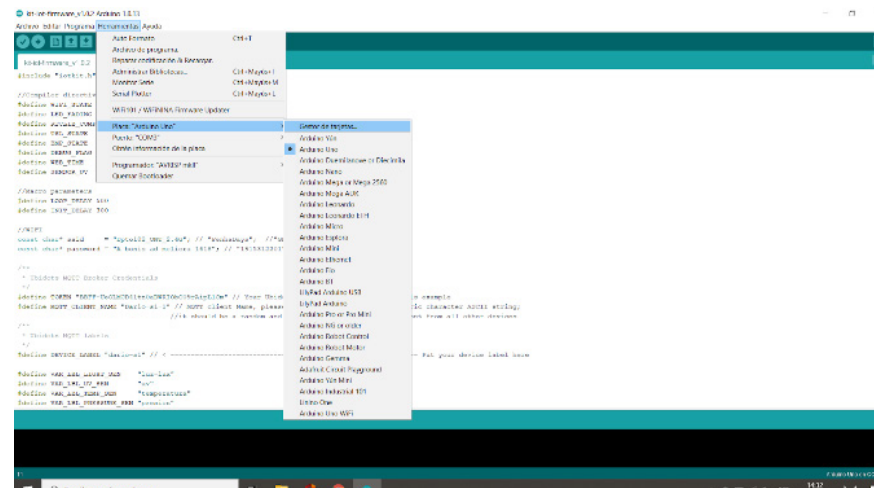


4 Dale clic a *preferencias* y en la ventana que dice “gestor de URLs adicionales de tarjetas” escribe la siguiente URL https://dl.espressif.com/dl/package_esp32_index.json

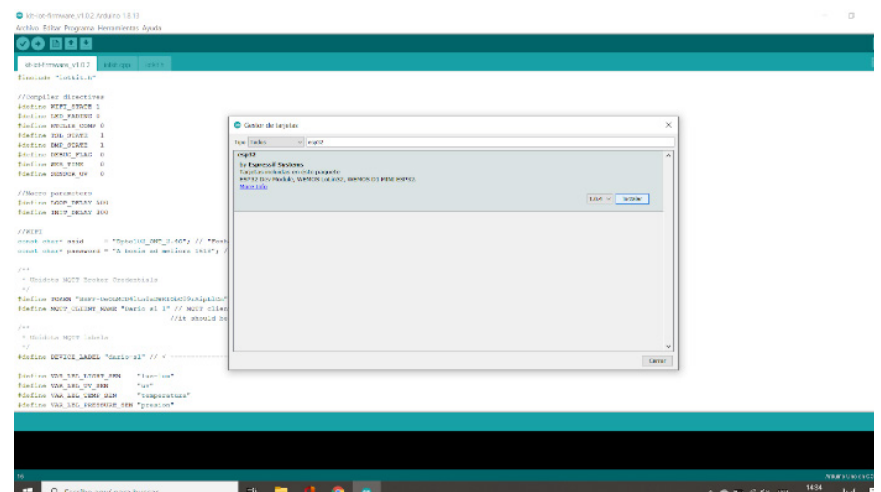
Luego, dale clic en OK.



5 Anda a *herramientas*, luego a *placa* y después a *gestor de tarjetas*.



6 Dale clic en *gestor de tarjetas* y escribe *esp32*, y ahora dale clic en *instalar*.



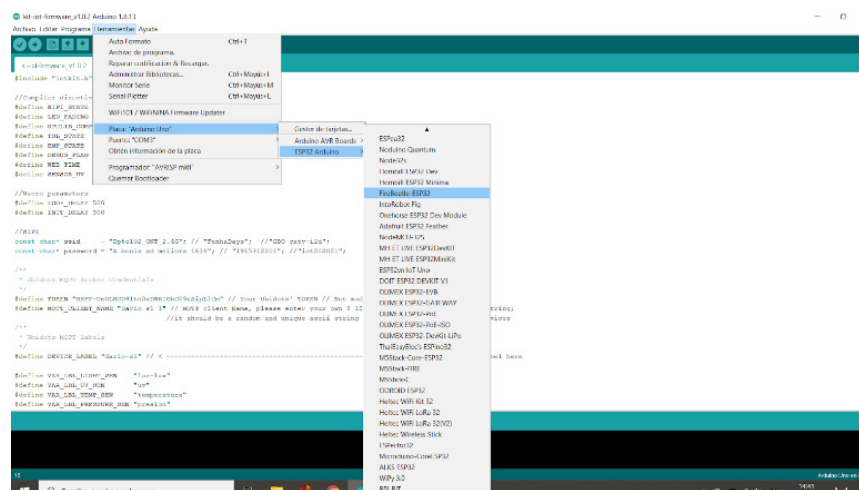
7

Espera mientras se instalan los gestores de la placa. Cuando finalice la instalación dale cerrar.

8

Ahora nuevamente ve a *herramientas*, luego a *placa*, después a *ESP32arduino*, y finalmente busca **FireBeetleESP32**. Cuando la encuentres, le damos clic.

Con este paso le hemos indicado al programa Arduino, que nuestro trabajo lo haremos con ese modelo de placa (compatible con Darío S1).



Con estos pasos correctamente realizados, has habilitado tu placa Darío S1 para ser programada desde tu computador. Sé que son varios pasos, pero si lo comparas, no es tan distinto a lo que hacemos cuando instalamos una nueva impresora.

Ahora, te invito a conectar tu placa a internet, y a recolectar la información de sus sensores.

1 Si te fijas en la programación pre establecida en Arduino tenemos un espacio para incorporar nuestra red y nuestra clave wifi.

```
ki-iot-firmware_v1.0 Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

//iot-firmware_v1.0
#include "iotkit.h"

//Compiler directives
#define WIFI_STATE 1
#define LED_FADING 0
#define RTCLIB_COMP 0
#define TSL_STATE 1
#define BMP_STATE 1

//Macro parameters
#define LOOP_DELAY 500

//Wifi
const char* ssid = "micasa";
const char* password = "America3085_2020.";

//MQTT
#define MQTT_CLIENT_NAME "prototipo1"

//MQTT Broker Credentials
#define TOKEN "BBFF-gyqrUPMfqtgfa9iAc0vrjhhXdsFvHa"
#define MQTT_CLIENT_NAME "prototipo1"

//MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//It should be a random and unique ascii string and different from all other devices
```

2 Para modificarla y adaptarla a tu casa debemos hacer lo siguiente:

Si tu red, por ejemplo, se llama micasa, esa es la que debes incluir en este espacio marcado en amarillo:

```
const char* ssid = "micasa";
```

Debe quedar entre las comillas. No debes borrar nada más de esa línea, atento o atenta con eso.

```
ki-iot-firmware_v1.0 iotkit.cpp iotkit.h

#include "iotkit.h"

//Compiler directives
#define WIFI_STATE 1
#define LED_FADING 0
#define RTCLIB_COMP 0
#define TSL_STATE 1
#define BMP_STATE 1

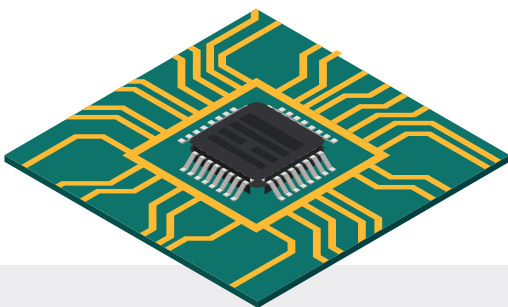
//Macro parameters
#define LOOP_DELAY 500

//Wifi
const char* ssid = "micasa";
const char* password = "America3085_2020.";

//MQTT
#define MQTT_CLIENT_NAME "prototipo1"

//MQTT Broker Credentials
#define TOKEN "BBFF-gyqrUPMfqtgfa9iAc0vrjhhXdsFvHa"
#define MQTT_CLIENT_NAME "prototipo1"

//MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//It should be a random and unique ascii string and different from all other devices
```



3

Ahora debemos incluir la contraseña de nuestra red wifi, de la misma manera que lo hicimos anteriormente, entre las comillas. Por ejemplo, si tu contraseña del wifi es: 123456789, entonces debes escribirla así:

`const char* password = "123456789";`

```
kit-iot-firmware_v1.0 $  iotkit.cpp  iotkit.h
#include "iotkit.h"

//Compiler directives
#define WIFI_STATE 1
#define LED_FADING 0
#define RTCLIB_COMP 0
#define TSL_STATE 1
#define BMP_STATE 1

//Macro parameters
#define LOOP_DELAY 500

//WIFI
const char* ssid = "micasa";
const char* password = "123456789";

/**
 * Ubidots MQTT Broker Credentials
 */
#define TOKEN "BBFF-gyqrUPMfgtqfa9iAcOvrj1hXdsFvHa" // Your Ubidots' TOKEN // Not modify for this example
#define MQTT_CLIENT_NAME "prototipol" // MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//it should be a random and unique ascii string and different from all other devices

/**
 * Ubidots MQTT labels
 */
#define DEVICE_LABEL "prototipol" // < ----- Put your device label here
```

4

A continuación, debemos incorporar el nombre de nuestra placa en los dos espacios. El nombre de tu placa lo encuentras en su caja y además en un adhesivo detrás de la placa. Se puede llamar, por ejemplo, puertooctay1

Si la placa se llama puertooctay1 escribir:

`#define MQTT_CLIENT_NAME "puertooctay1" // MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;`

`//it should be a random and unique ascii string and different from all other devices`

```
kit-iot-firmware_v1.0, app Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda

iotkit.cpp iotkit.h

//WIFI
const char* ssid = "VTR-8955785"; // "FenhaDays"; // "GB0 catv-12d";
const char* password = "5jhyphXtnocx"; // "1919312201"; // "iot202021";

/**
 * Ubidots MQTT Broker Credentials
 */
#define TOKEN "BBFF-UeOLMCD41tn0aWRI0bC09rAipLlCm" // Your Ubidots' TOKEN // Not modify for this example
#define MQTT_CLIENT_NAME "puertooctay1" // MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//it should be a random and unique ascii string and different from all other devices

/**
 * Ubidots MQTT labels
 */
#define DEVICE_LABEL "prototipol" // < ----- Put your device label here

#define VAR_LBL_LIGHT_SEN "luz-lux"
#define VAR_LBL_UV_SEN "uv"
#define VAR_LBL_TEMP_SEN "temperatura"
#define VAR_LBL_PRESURE_SEN "presion"
#define VAR_LBL_PRESURE_SEN "presion"
#define VAR_LBL_ALTURA_SEN "altura"
#define VAR_LBL_POT_SEN "perilla"
#define VAR_LBL_SWITCH1 "switch1"
#define VAR_LBL_SWITCH2 "switch2"

Librería inválida encontrada en C:\Users\aleve\OneDrive\Documentos\Arduino\libraries\U8glib-master: No encontrado archivos de encabezado (.h) en
```

5

Deberás poner también el nombre de tu placa en la siguiente línea:

`#define DEVICE_LABEL "puertooctay1" // < ----- Put your device label here`

```
kit-iot-firmware_v1.0, app Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda

iotkit.cpp iotkit.h

//WIFI
const char* ssid = "VTR-8955785"; // "FenhaDays"; // "GB0 catv-12d";
const char* password = "5jhyphXtnocx"; // "1919312201"; // "iot202021";

/**
 * Ubidots MQTT Broker Credentials
 */
#define TOKEN "BBFF-UeOLMCD41tn0aWRI0bC09rAipLlCm" // Your Ubidots' TOKEN // Not modify for this example
#define MQTT_CLIENT_NAME "puertooctay1" // MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//it should be a random and unique ascii string and different from all other devices

/**
 * Ubidots MQTT labels
 */
#define DEVICE_LABEL "puertooctay1" // < ----- Put your device label here

#define VAR_LBL_LIGHT_SEN "luz-lux"
#define VAR_LBL_UV_SEN "uv"
#define VAR_LBL_TEMP_SEN "temperatura"
#define VAR_LBL_PRESURE_SEN "presion"
#define VAR_LBL_PRESURE_SEN "presion"
#define VAR_LBL_ALTURA_SEN "altura"
#define VAR_LBL_POT_SEN "perilla"
#define VAR_LBL_SWITCH1 "switch1"
#define VAR_LBL_SWITCH2 "switch2"

Librería inválida encontrada en C:\Users\aleve\OneDrive\Documentos\Arduino\libraries\U8glib-master: No encontrado archivos de encabezado (.h) en
```

Finalmente, en el menú superior selecciona "archivo", y luego "salvar"



A continuación, seguirás los pasos que te permitirán recoger las mediciones de algunas variables medioambientales que registran los sensores incluidos en la placa Dario S1.

- 1 Para esto dale clic a la flecha ubicada en la parte superior izquierda (te indica la leyenda *Subir*). Haz clic, espera, y cuando diga **subido** esto quiere decir que tu placa está lista, y podrá reportar los datos de los sensores. Es posible que se encienda una luz, y haya una secuencia de sonido.



```
kit-iot-firmware_vf1 Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda
Subir
kit-iot-firmware_vf1 $
#include "iotkit.h"

//Compiler directives
#define WIFI_STATE 1
#define LED_FADING 0
#define RTCLIB_COMP 0
#define TSL_STATE 1
#define BMP_STATE 1

//Macro parameters
#define LOOP_DELAY 500

//WIFI
const char* ssid = "NODO_2,4Ghz";
const char* password = "America3085_2020.,";

/**
 * Ubidots MQTT Broker Credentials
 */
#define TOKEN "BBFF-gyqUfMfgtga9iAc0vrjlhXdsFvHa" // Your Ubidots' TOKEN // Not modify for this example
#define MQTT_CLIENT_NAME "prototipo1" // MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//it should be a random and unique ascii string and different from all other devices

/**
 * Ubidots MQTT labels
 */
```

- 3 En el programa Arduino, anda a la lupa que se encuentra en la parte superior derecha (*Monitor serie*), y dale clic.



```
kit-iot-firmware_vf1 Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda
Monitor Serie
#include "iotkit.h"

//Compiler directives
#define WIFI_STATE 1
#define LED_FADING 0
#define RTCLIB_COMP 0
#define TSL_STATE 1
#define BMP_STATE 1

//Macro parameters
#define LOOP_DELAY 500

//WIFI
const char* ssid = "NODO_2,4Ghz";
const char* password = "America3085_2020.,";

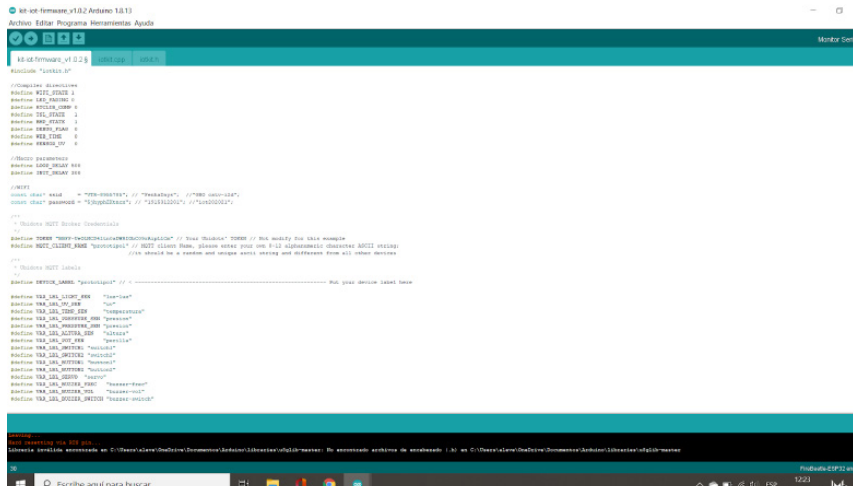
/**
 * Ubidots MQTT Broker Credentials
 */
#define TOKEN "BBFF-gyqUfMfgtga9iAc0vrjlhXdsFvHa" // Your Ubidots' TOKEN // Not modify for this example
#define MQTT_CLIENT_NAME "prototipo1" // MQTT client Name, please enter your own 8-12 alphanumeric character ASCII string;
//it should be a random and unique ascii string and different from all other devices

/**
 * Ubidots MQTT labels
 */
#define DEVICE_LABEL "prototipo1" // < ----- Put your device label here
```

- 2 Ahora, veamos como nuestros sensores de luz, temperatura, y demás comienzan a darnos sus reportes.

4

Se debe abrir una nueva ventana. Si te das cuenta, nos está reportando los valores que cada uno de los sensores está tomando.



¡Felicitaciones! si lograste realizar este procedimiento diste tus primeros pasos como programador de IoT

Te proponemos ir modificando las variables que mide tu placa y registrar sus valores de la siguiente manera:

Variable	Valor inicial	Modificación sugerida	Valor posterior a modificación
Temperatura		Poner secador de pelo por 40 segundo, apuntando la placa a 30 cm de distancia	
Presión atmosférica		Poner secador de pelo por 40 segundo, apuntando la placa a 30 cm de distancia	
Luminosidad		Poner luz de linterna, apuntando a sensor de luz	
UV		Poner placa a plena luz del sol, ojalá a mediodía	

Recuerda que la placa actualiza constantemente los valores desde la ventana de Monitor serie.



Desafío:

Si desconectas la placa del computador, y la energizas con un cargador de celular

¿sigue arrojando lecturas en el Monitor serie?

Repite la experiencia de modificación de variables anterior, sacando la placa al patio de tu casa y registrando los valores que te da para las siguientes variables:

Variable	Valores dentro de la casa	Valores en el patio
Temperatura		
Presión atmosférica		
Luminosidad		
UV		

Como verás, vuestra placa registra las condiciones medioambientales en cada lugar donde la ubicas, siendo lo más importante que esté conectada a internet. Esta información es llevada al microcontrolador, el cual reporta a través de la antena wifi dichos datos hasta nuestro computador, que también debe estar conectado a la misma red.

¿Qué acciones podemos pedirle que realicen a los actuadores de nuestra placa?

Te dejamos invitado a la siguiente clase on line, donde aprenderás a controlar desde internet el funcionamiento de tu placa Darío S1, y podrás materializar tu diseño para solucionar una necesidad o problemática de tu entorno, usando internet de las cosas.

CONCLUSIONES

Ahora y después de realizada la instalación de tu placa Darío s1, es momento de retomar la hipótesis formulada a la pregunta: ¿Qué necesidad o problema de mi entorno resolvería usando Internet de las Cosas? y pensar ¿se tuvo que ajustar tu hipótesis inicial en relación con lo que descubriste con las actividades? Explícalo en las siguientes líneas.

Es posible que tu respuesta final haya cambiado... pero no te preocupes, de eso se trata el trabajo en STEM, de plantear ideas, diseñar soluciones comprobarlas por medio de la experimentación, tener en cuenta los resultados, y muy importante también compartir tus ideas con otras personas, para nutrirse de nuevos puntos de vista.



Somos una fundación sin fines de lucro que **buscamos construir puentes entre la ciencia y la sociedad**. Llevamos educación científica a niños, niñas y jóvenes; impulsamos el diálogo entre expertos, expertas, ciudadanos y ciudadanas a través de espacios de exposición y conversación; potenciamos el impacto social de la ciencia a través del financiamiento de iniciativas particulares.

Súmate a la comunidad Ecoscience y conoce más de nuestro trabajo. Síguenos o suscríbete a nuestras redes sociales. Nos puedes encontrar como @ecoscienceorg.



www.ecoscience.org