



# Procedimientos para editar y configurar recursos educativos abiertos para un experimento físico



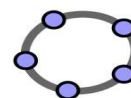
Wemos D1 R32



BLE



Phyphox Ver.1.1.11



GeoGebra

Software Dinámico de Matemáticas

<http://www.ugr.es/~anillos/geogebra/>

[martinramon.cordero.09@dgeti.sems.gob.mx](mailto:martinramon.cordero.09@dgeti.sems.gob.mx)

<https://t.me/+mVLELfqOgxQxZTEEx>

## CONTENIDO



GeoGebra  
Software Dinámico de Matemáticas

### Esp32-servidor

5. Variables del entorno.
6. Gestión de recursos y servicios.
7. Descripción del programa informático modelo.
8. Prueba y validación del programa informático modelo.

### Actividades previas

- a) Descarga de software y aplicaciones informáticas: Arduino IDE, GeoGebra, Phyphox, Telegram, Light Blue.
- b) Configuración de paquetes esp32 en Arduino IDE.
- c) Adecuación de recursos de hardware Wemos ESP32 y Shield multifuncional 9 en 1.
- d) Actualización y ejecución del firmware ESP32 - Phyphox.



GeoGebra  
Software Dinámico de Matemáticas

### Aplicación Phyphox

1. Descripción y configuración de la interface usuario.
2. Ejecución de un experimento.
3. Monitorización remota y exportación de datos.

### Procesamiento de datos con GeoGebra

9. Importación y registro de datos.
10. Presentación de datos en vista algebraica y gráfica 2D.
11. Procesamiento analítico de datos.

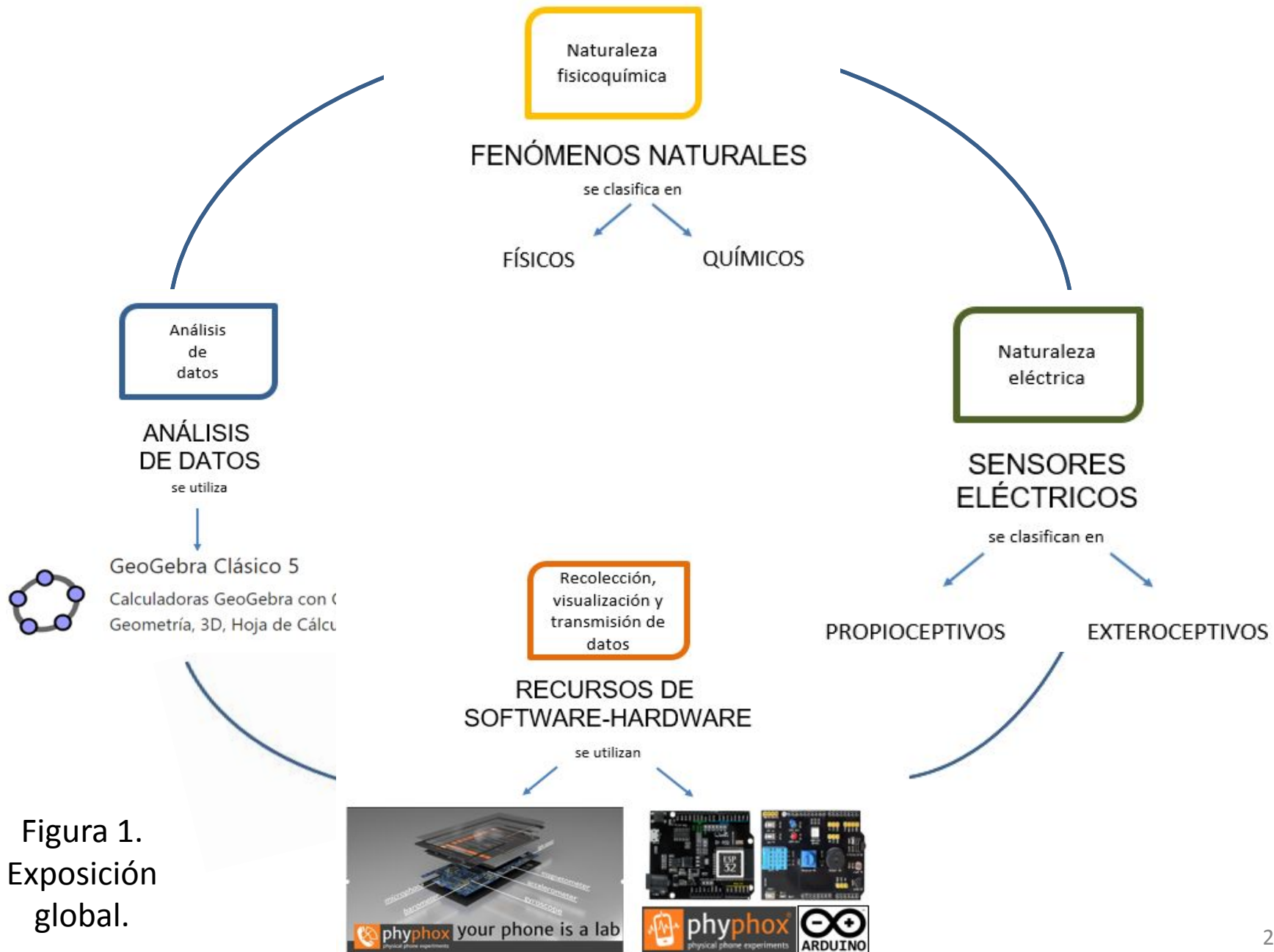


Figura 1.  
Exposición  
global.

Naturaleza  
fisicoquímica

## FENÓMENOS NATURALES

se clasifica en

FÍSICOS

QUÍMICOS

**Posición**

variación de  
velocidad lineal  
y angular

**Ópticos**

intensidad  
luminosa

**Acústicos**

amplitud y  
frecuencia

**Magnéticos**

intensidad  
de campo  
magnético

**Térmicos**

temperatura  
y humedad

**Densidad  
volumétrica**

-partes  
por millón  
-potencial  
de hidrogeno

Figura 2. Desglose por  
naturaleza fisicoquímica



[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRrE AyRYUHGx\\_PY97JQSD3jkYRuMQexQnupGx9VVQeGKW1q1 Ywfm-dde9IZjLWvj5wDUsm&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRrE AyRYUHGx_PY97JQSD3jkYRuMQexQnupGx9VVQeGKW1q1 Ywfm-dde9IZjLWvj5wDUsm&usqp=CAU)

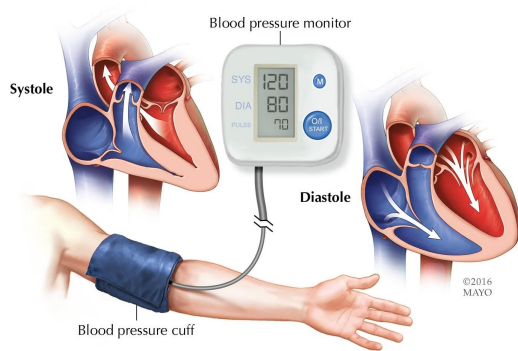
Naturaleza eléctrica

## SENSORES ELÉCTRICOS

se clasifican en

PROPIOCEPTIVOS

EXTEROCEPTIVOS



<https://www.mayoclinic.org/content/dam/media/en/images/2023/02/10/blood-pressure-measurement.jpg>



<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ80 jiiXGZpPDxXjQVwdHfTJWyxfq5-eAAJpA&usqp=CAU>



<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT-VHEVMRI6fnZ4VpejadDA9NyDW4ayOWHu8w&usqp=CAU>



<https://www.fundacionmapfre.org/educacion-divulgacion/seguirad-vial/sistemas-adass/tipossensores-de-aparcamiento/>



[https://phyphox.org/wp-content/uploads/2016/04/smartphone\\_sensors.png](https://phyphox.org/wp-content/uploads/2016/04/smartphone_sensors.png)

Desglose por naturaleza

### Figura 3. Desglose por naturaleza eléctrica.

#### Propioceptivo

1. **f. Psicol.** Percepción inconsciente de los movimientos y de la posición del cuerpo, independiente de la visión.

**Exteroceptivo**-Percepción y registro de los movimientos exteriores al objeto.



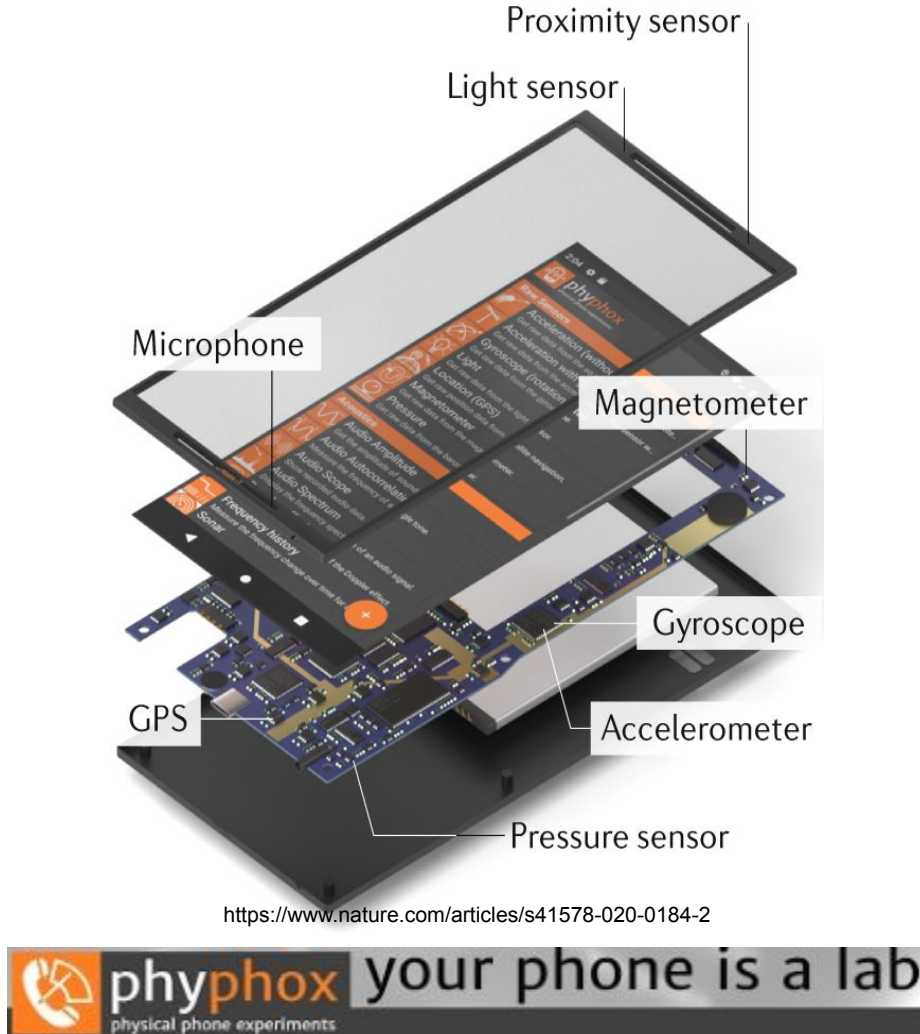


Figura.4. Aplicación móvil Phyphox..

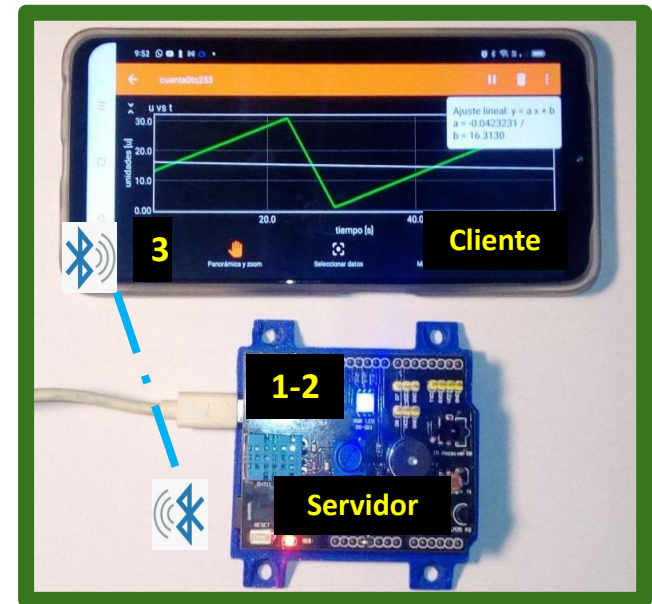
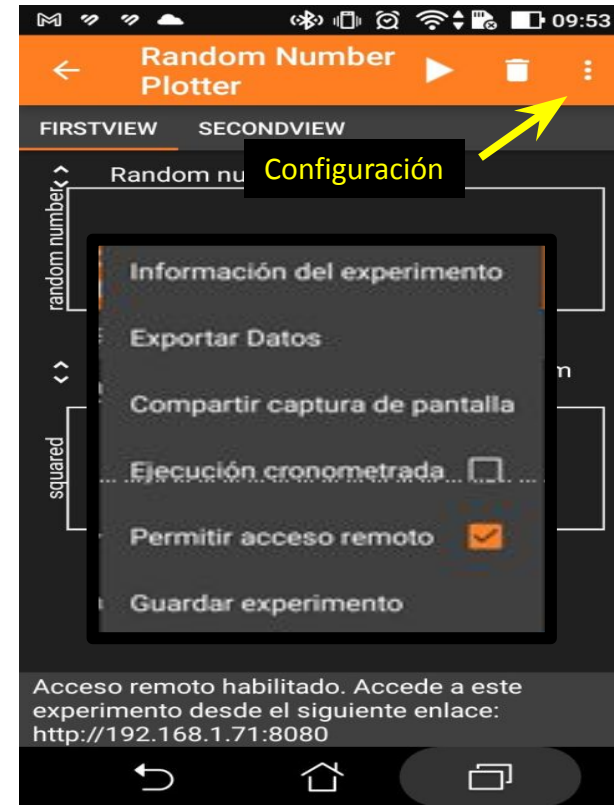


Figura.5. Ejecución de recursos de hardware-software.



Catálogo de categorías experimentos.



Configuración de experimento.

Figura 6. Procedimiento de configuración de experimento predefinido.

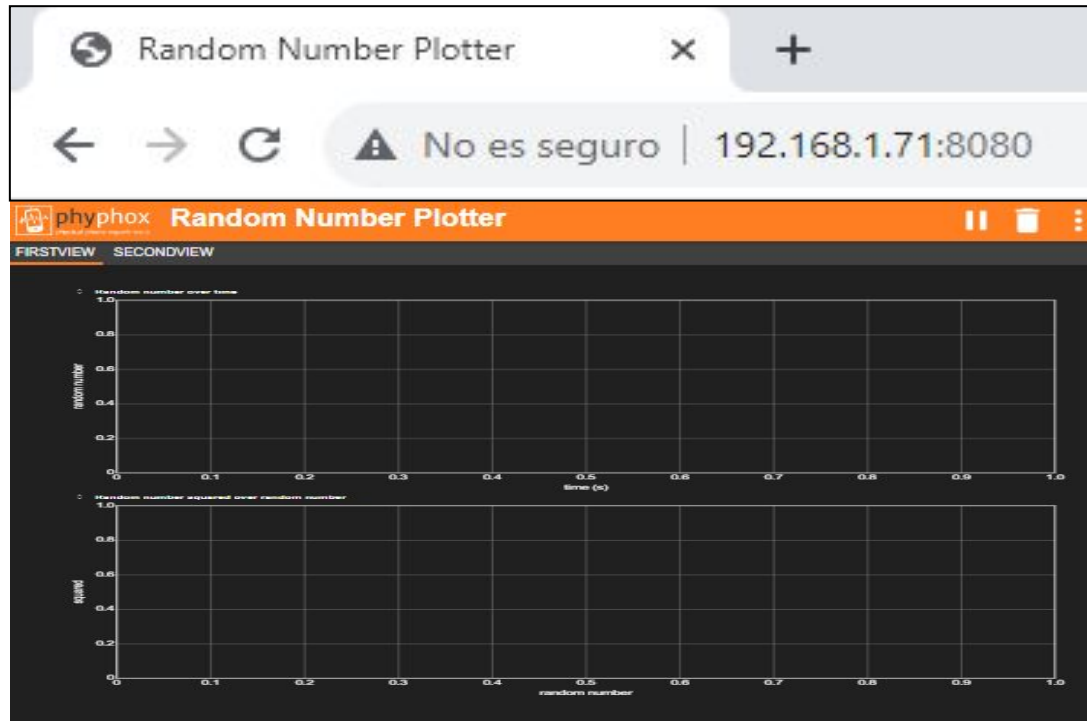


Figura 8. Pre-monitorización de experimento modo remoto.

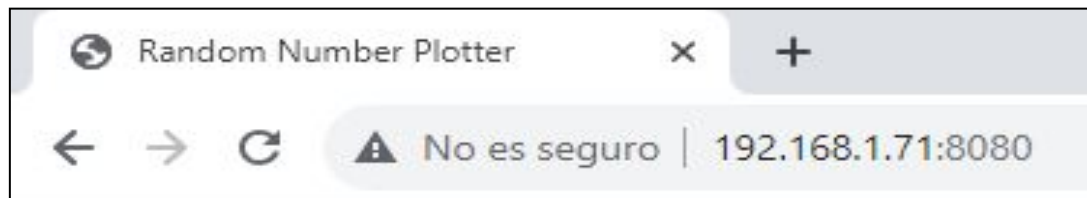
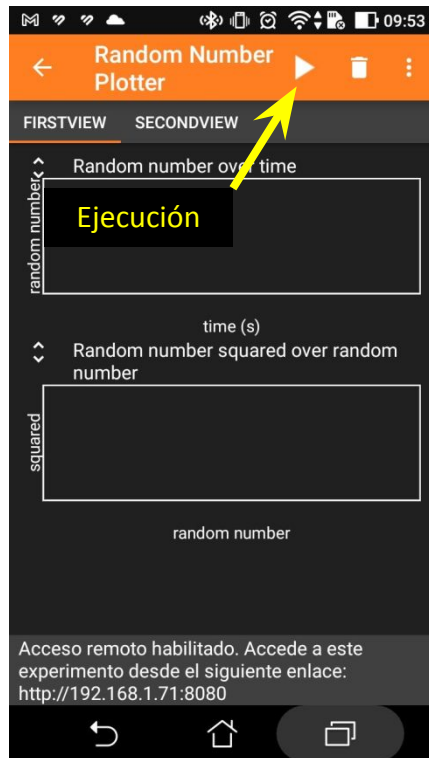
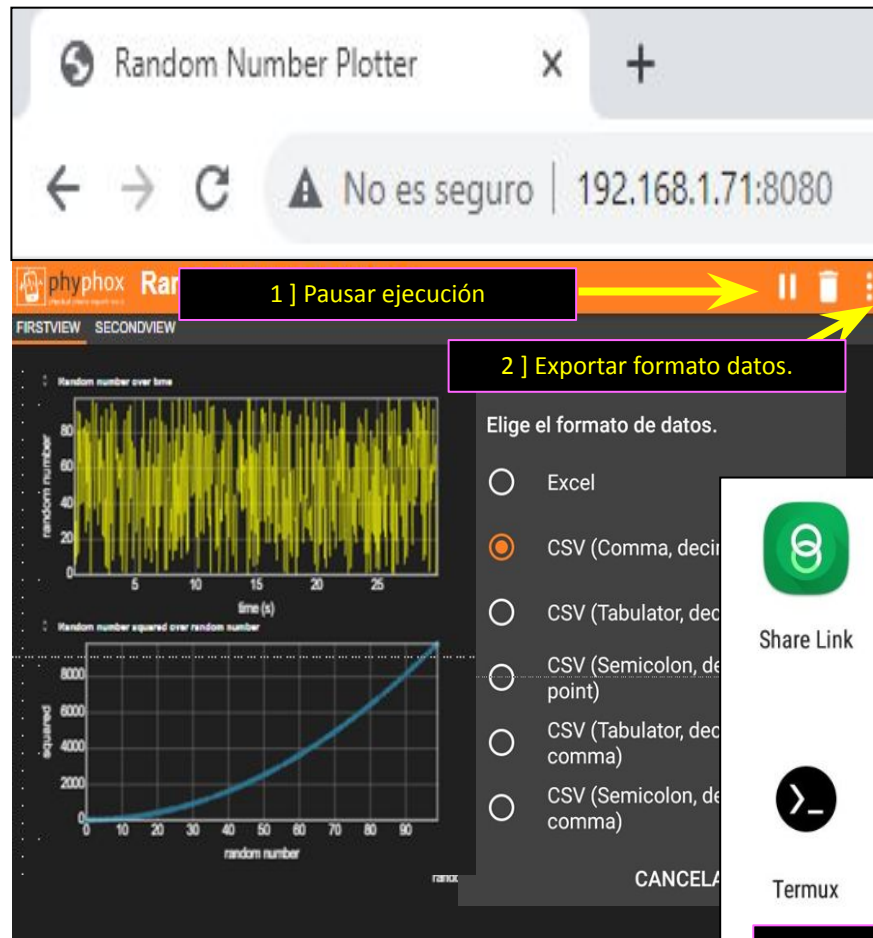


Figura 7. Red de área local  
ámbito.terminal:puerto.

Figura 7. Monitorización colectiva del experimento.



Ejecución de experimento.



Exportación de datos.

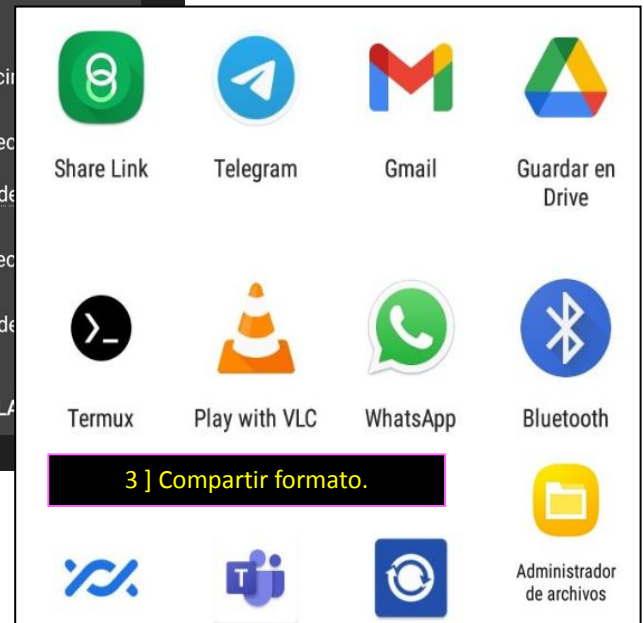
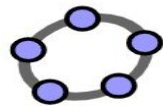


Figura 8. Procedimiento de ejecución de experimento y exportación de datos.

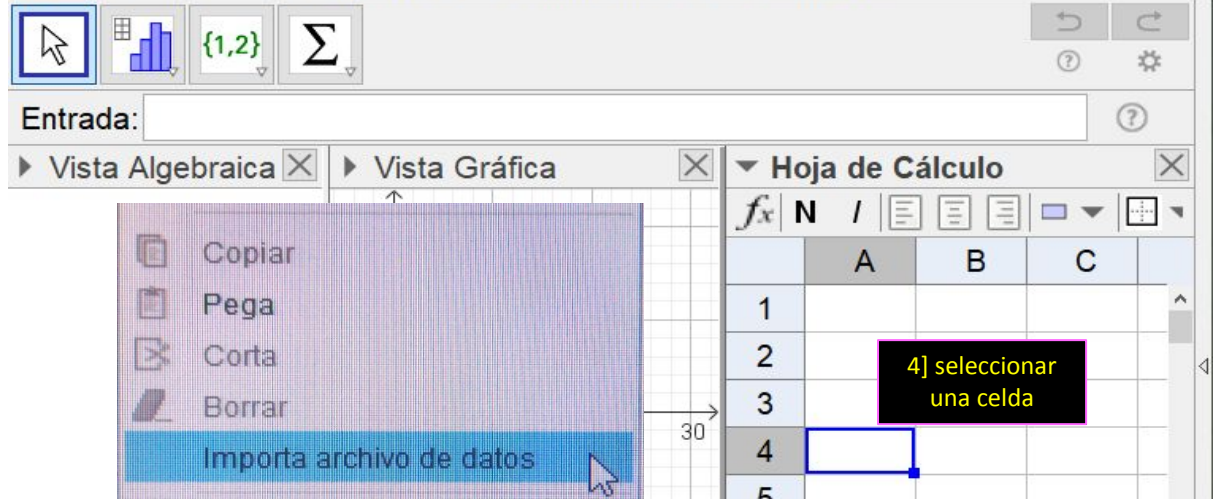




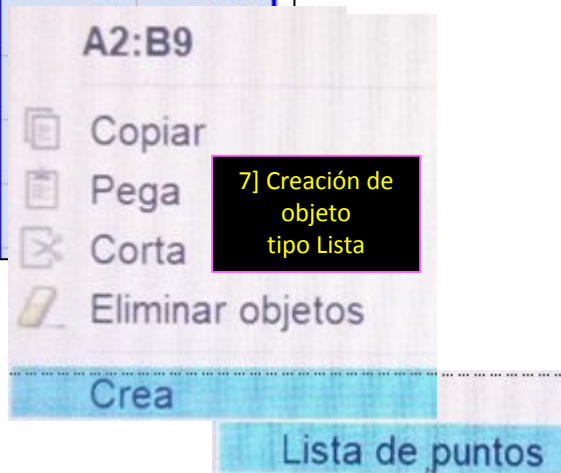
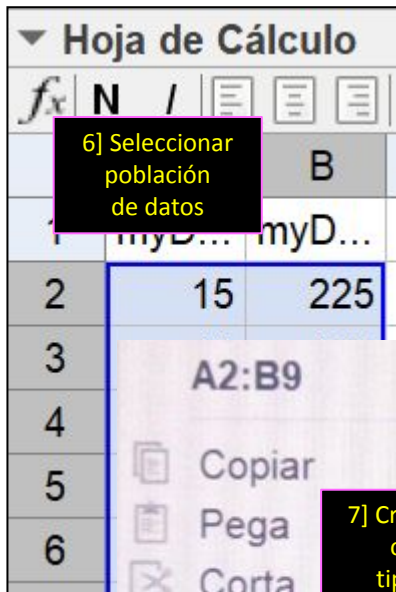
## GeoGebra Clásico 5

Calculadoras GeoGebra con Gráficas,  
Geometría, 3D, Hoja de Cálculo, CAS

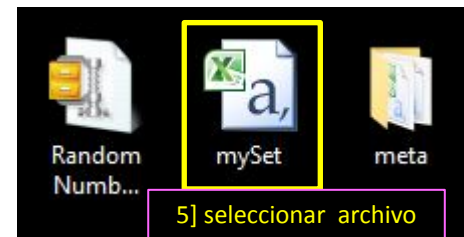
Archivo Edición Vista Opciones Herramientas Ventana Ayuda



Importar archivo CSV en GeoGebra.

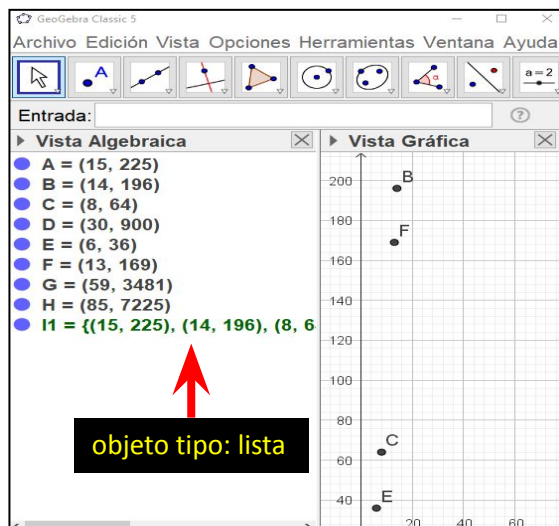


Creación de objeto lista.

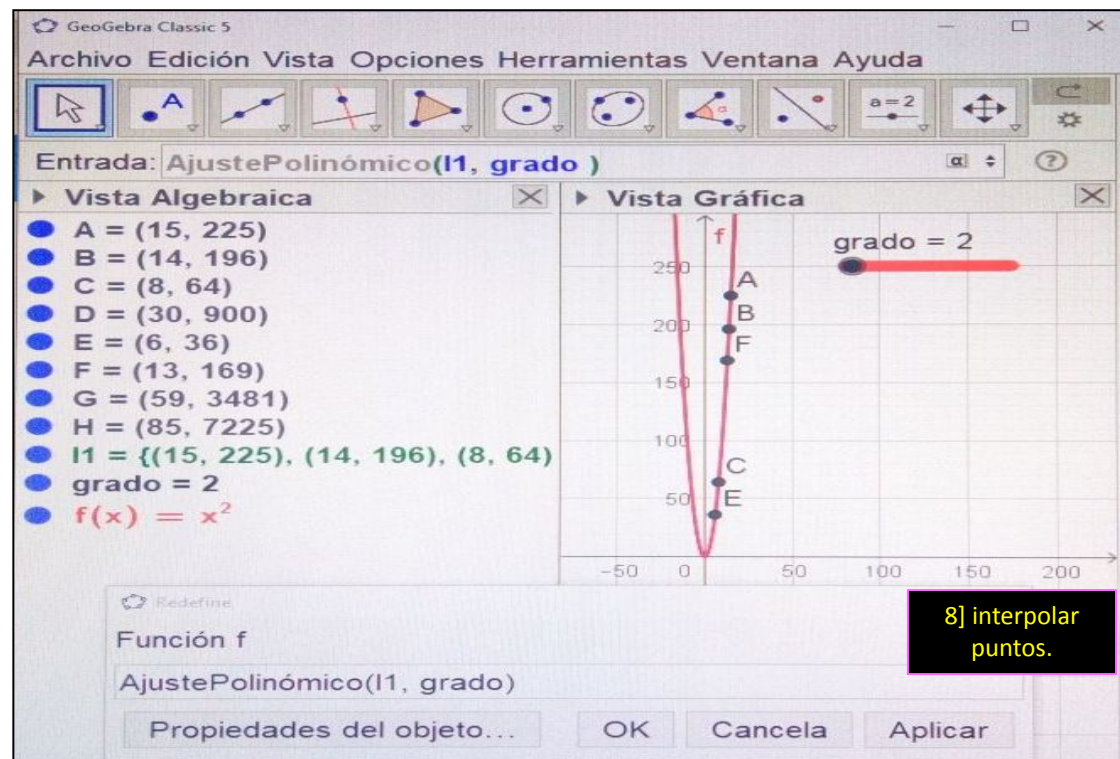


Descomprimir y seleccionar.

Figura 9. Procedimiento de análisis de datos.



Visualización semiótica de objetos.



Ajuste e interpolación de datos.

AjustePolinómico(l1, grado )

Figura 10.Procedimiento de analítica de datos.

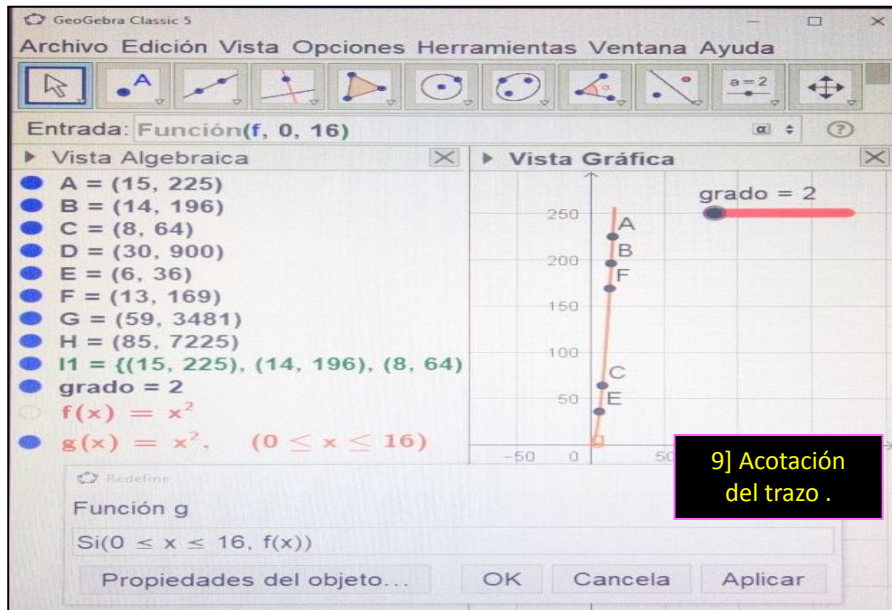


Figura 16. Acotación de función.

$\text{Función}(f, 0, 16)$

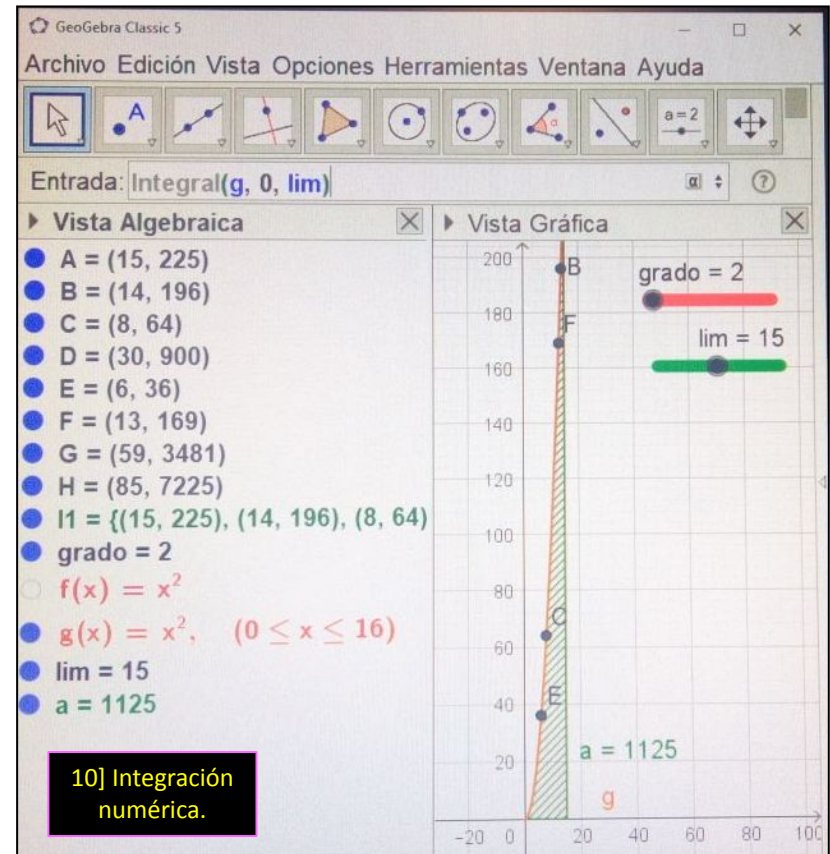


Figura 17. Área bajo la curva.

$\text{Integral}(g, 0, \text{lim})$

Figura 11. Procedimiento de interpretación de datos.



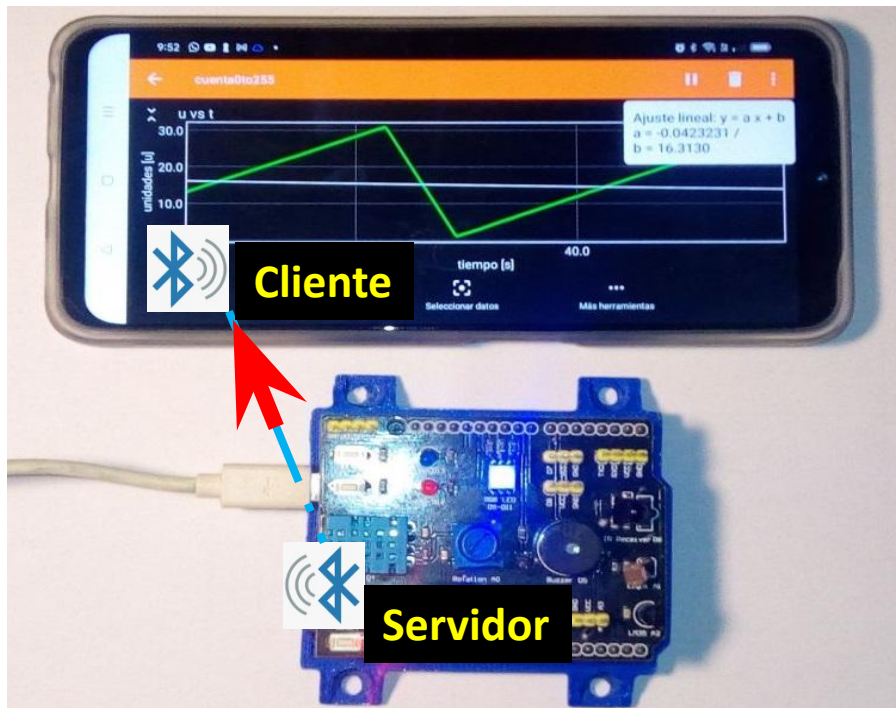


Figura.1. Ejecución de recursos de hardware-software.

Es un modelo de diseño de software, donde: el registro, pre procesamiento y transmisión de datos descriptivos de un fenómeno, son ejecutados por los recursos de hardware de un **servidor** para los recursos informáticos de monitorización y exportación de datos en un **cliente**. En este modelo cliente-servidor la conexión y comunicación de datos es Bluetooth Low Energy en modo notificación. El *remitente de una solicitud* es el cliente, dispositivo móvil, es quien inicia peticiones, espera-recibe respuestas del servidor, posee la interface gráfica de usuario y exporta datos del experimento. El *receptor de la solicitud* se le conoce como servidor, Wemos ESP32 D1 R32, registra, pre procesa y envía la datos al cliente, no guarda datos y contiene el *firmware* del experimento.



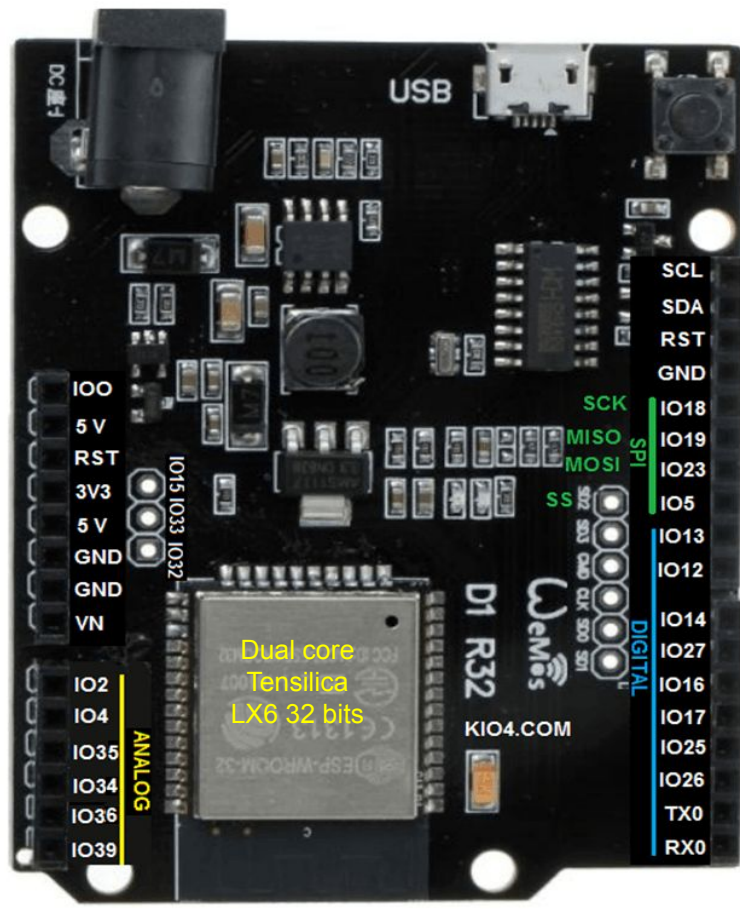
## ESP-WROOM-32

Ram: 520 KiBytes  
 Rom: 448 KiBytes  
 Fosc: 160 or 240 MHz

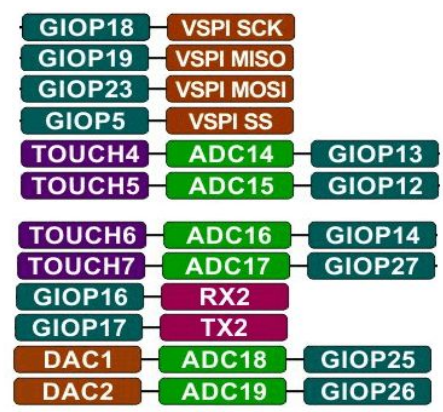
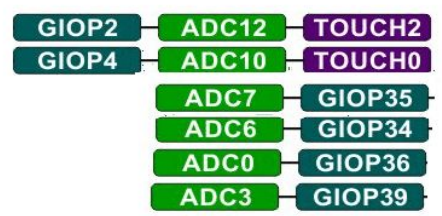
- Conectividad inalámbrica  
 Wi-Fi: 802.11 b/g/n  
 Bluetooth: v4.2 BR / EDR / LE

Vin: +7Vcd a +12Vcd  
 500mA

Vusb: +5Vcd  
 500mA  
 Usb tipo 2.0



- Interfaces periféricas  
 18 GPIO programables  
 12 Canales analógico digital



<https://www.botonroll.com/en/esp/3639-wemos-d1-r32-w-esp32-uno-r3-pinout.html>  
<https://en.wikipedia.org/wiki/ESP32>

Disponible en: <https://uelectronics.com/producto/wemos-esp32-d1-r32/>





Figura.2. Recursos materiales

## 1] ESPECIFICACIONES Y CARACTERÍSTICAS

Modelo: WeMos ESP32 D1 R32

MCU: ESP32-WROOM-32

CPU: Dual core Tensilica LX6 (32 bit)

Frecuencia de Reloj: 240MHz

ROM: 448 kB SRAM: 520kB, Memoria Flash: 4MB

Interfaz de comunicación: UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S,

IR, contador de pulsos, GPIO, Sensor Táctil Capacitivo, ADC, DAC

Seguridad: WPA, WPA2, WPA2-Enterprise, WPS

Voltaje de entrada MicroUSB: 5 V (500 mA máx.)

Voltaje de entrada VIN Power Jack: 7V a 16V DC

Consumo de corriente: Máximo 250mA, Modo ahorro: 0.15mA y Funcionamiento (sin WiFi): 20mA

Convertidor de USB a serie CH340

1]<https://uelectronics.com/producto/wemos-esp32-d1-r32/>

2] <https://uelectronics.com/producto/shield-multifuncional-9-en-1-compatible-arduino/>

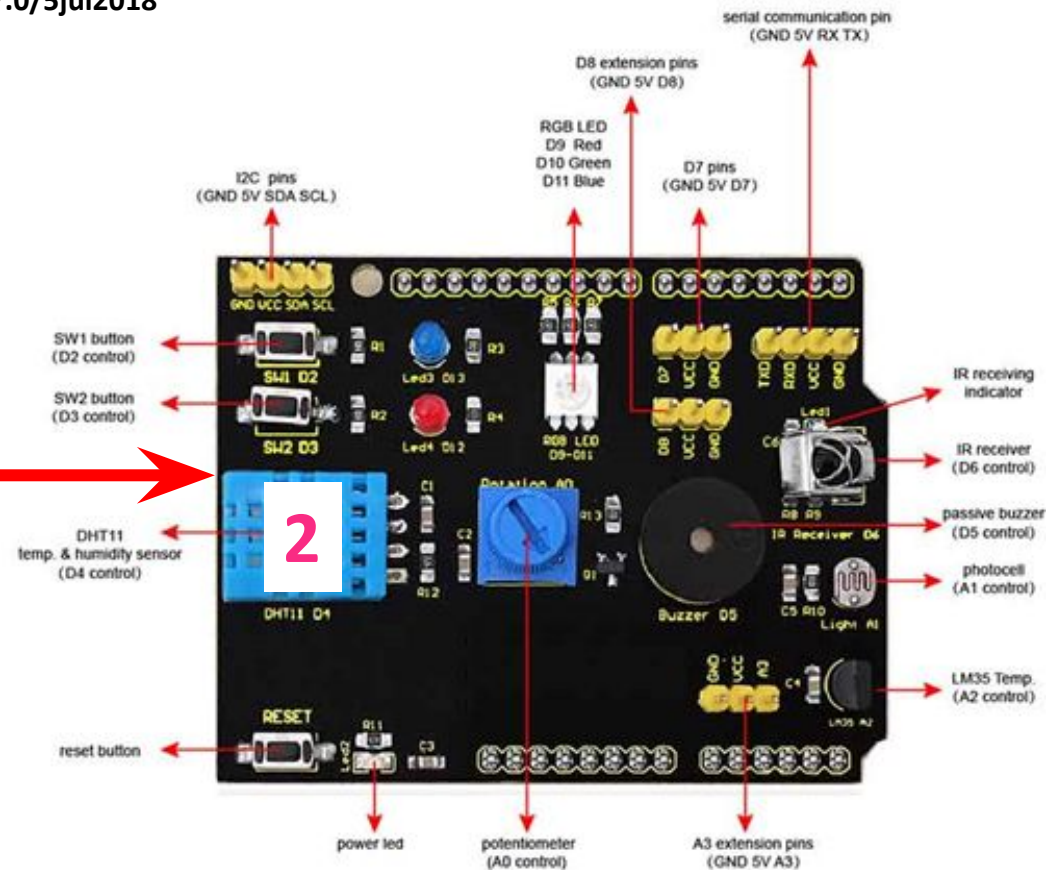


Figura. 2a. Recursos materiales  
Tarjeta de sensores.

## Recursos de hardware-software



Figura.3. Adecuación de hardware.

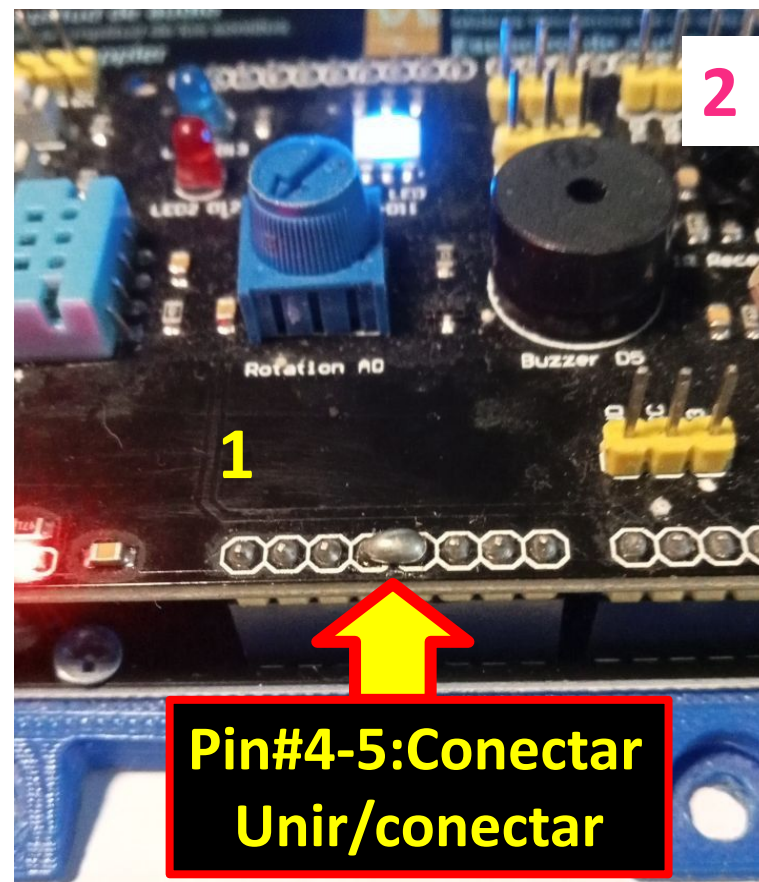


Figura.3a. Adecuación de hardware.

Polarización a 3.3Volts para la tarjeta shield-multifuncional-9-en-1-compatible-arduino. No funciona el sensor de temperatura LM35. Su rango de polarización es a partir de +4Volts. Información Apartado 6 <https://www.ti.com/document-viewer/LM35/datasheet/specifications#SNIS1597135>

El Afirmente ofrece la Obra "tal y como es" y no hace representaciones o garantías de ningún tipo concernientes a la Obra.

← → ↻ <https://phyphox.org/editor/>

**Main** Input Output Views Analysis Export → XML Download/Transfer

**Experiment information**

*Information about this experiment like its title, icon and description.*

Title ? cuenta0to255 0 Translations

Category ? Demostración 0 Translations

Icon ? cnt cnt

Color ? 44D62C

Description ? Contador cíclico ascendente a partir de 1 hasta 255.  
1de2]cuenta0to255.phyphox  
2de2]cuenta0to255.ino  
Verifica: Id,name,UUId,CharId  
0 Translations

Label of new Link ? Add a web link

Default locale ? es

Figura 4. Configuración del experimento desde editor.

## Main

<b>Title</b>	cuenta0to255
<b>Category</b>	Demostración
<b>Icon</b>	cnt
<b>Color</b>	00ff00
<b>Description</b>	Contador cíclico ascendente a partir de 1 hasta 255.
<b>Default locale</b>	es

Tabla 1. Atributos informativos del experimento.

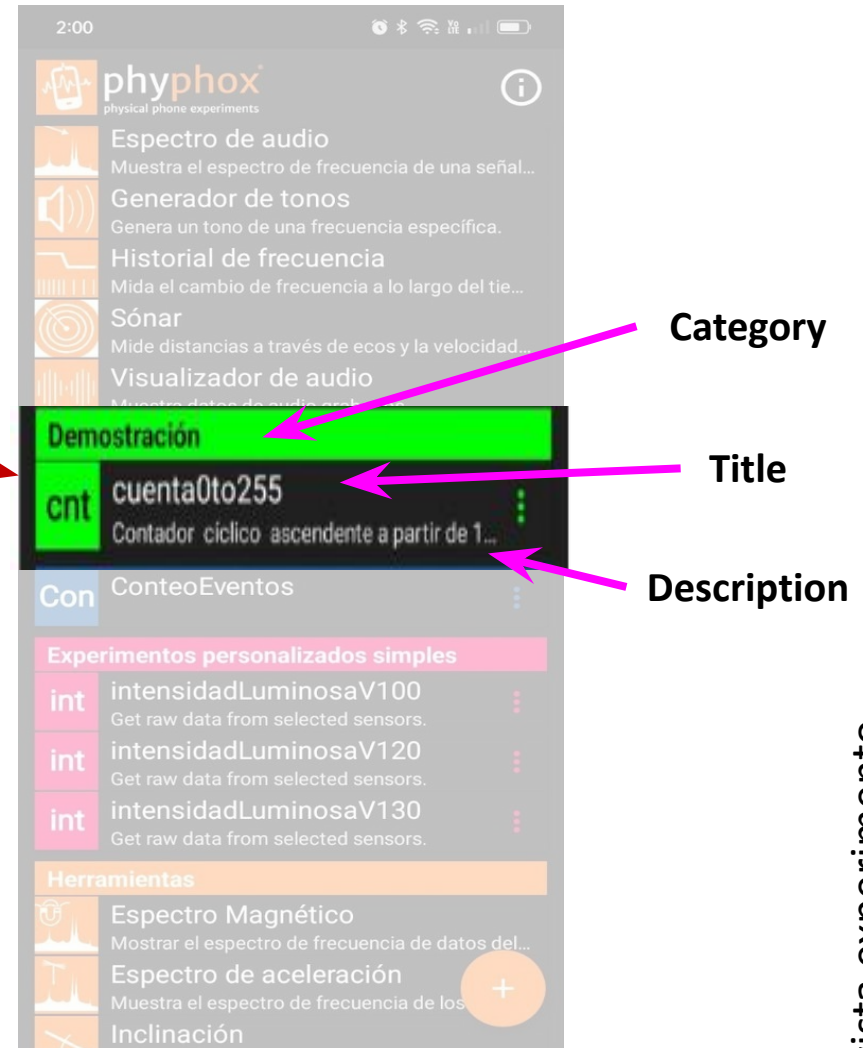
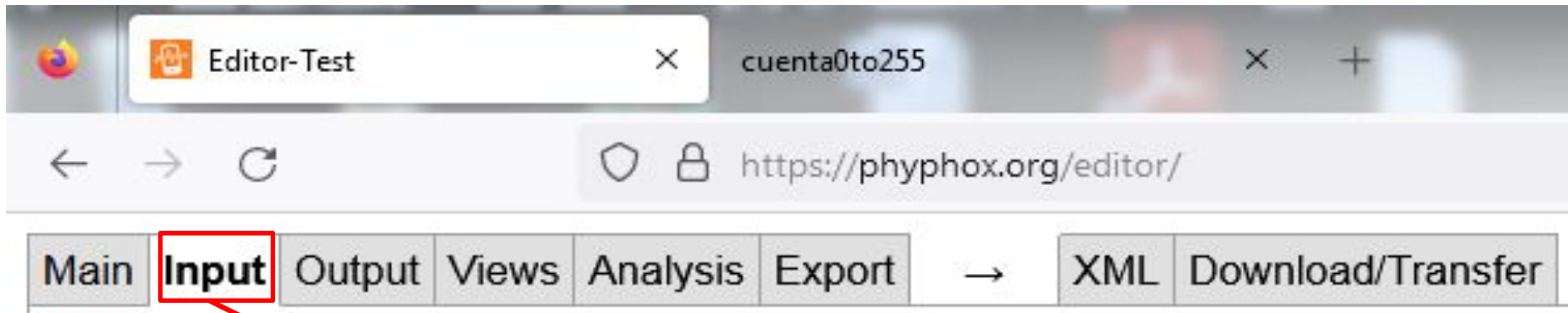


Figura 5. AppPhyphox-vista del experimento.





Category

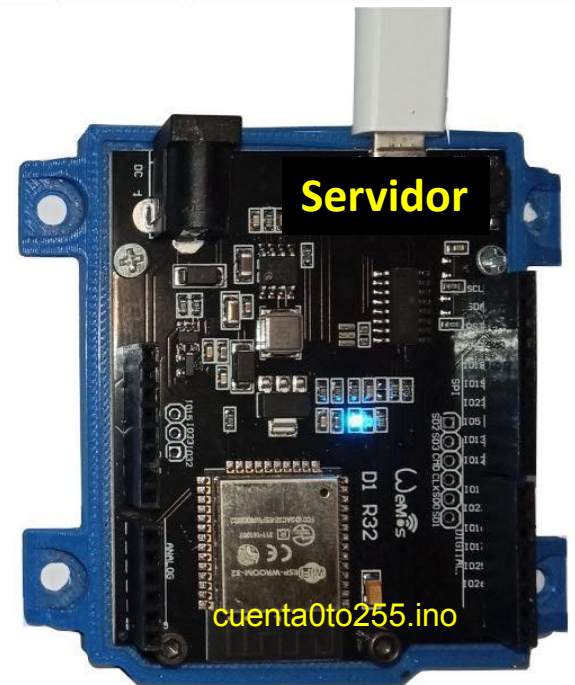
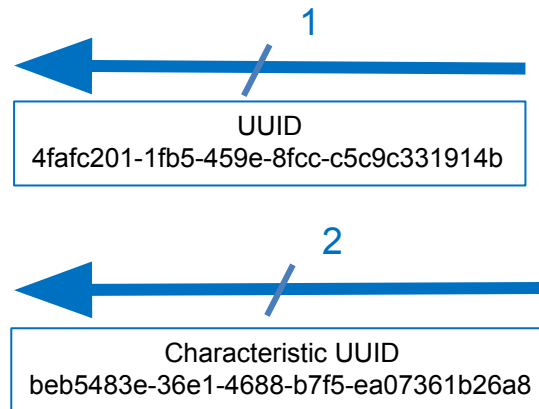


Figura 6. Protocolo de servicio bluetooth ble  
Cliente-Servidor.

Seleccionar dispositivo emisor de datos

**New input**

Add this module

New input type

- bluetooth ▼
- audio
- sensor
- location
- bluetooth

1 Selección de fuente de datos

**Input modules**

bluetooth

Identification ? ESP32

Device Name ? Esp32SoC

Service UUID ? 4fafc201-1fb5-459e-8fcc-c5

Mode ? Notification ▼

Rate (Hz) ? 1

Subscribe on start? ? ☐

Config

add Empty

Output

Figura 6. Definición de capa física.

#### Input modules

##### Bluetooth

Input

Identification ESP32

Device Name Esp32SoC

Service UUID 4fafc201-1fb5-459e-8fcc-c5c9c331914b

Mode Notification

Tabla 2. Configuración de control de acceso.



add Empty

Output

add Empty 2 añadir 2 módulos emisores de datos.

output

characteristic ?	beb5483e-36e1-4688-b7f5-
Conversion Function ?	singleByte
offset ?	
length ?	
separator ?	
label ?	
index ?	
extra ?	

#### Input modules

#### Output

Characteristic beb5483e-36e1-4688-b7f5-ea07361b26a8  
conversion function singleByte

#### Output

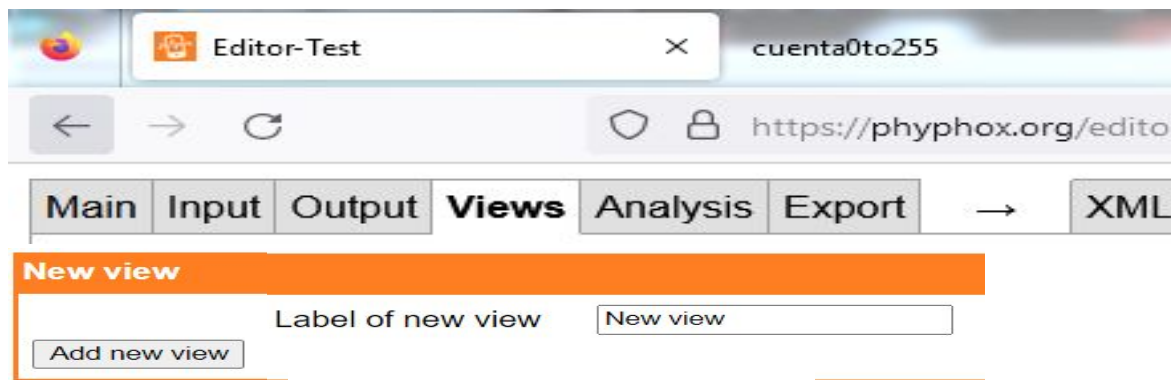
Characteristic beb5483e-36e1-4688-b7f5-ea07361b26a8  
Conversion string  
Extra time

output

characteristic ?	beb5483e-36e1-4688-b7f5-
Conversion Function ?	string
offset ?	
length ?	
separator ?	
label ?	
index ?	
extra ?	time

Tabla 2a. Configuración de control de acceso.

Figura 6a. Definición de capa física.



3

Selección de fuente de datos

New view

Label of this view: New view 0 Translations

graph

Label	u vs t	0 Translations
Aspect ratio	2.5	
Style	Lines	
Line Width	3	
Color	ffff00	
Partial Update	<input type="checkbox"/>	
History	1	
x axis label	tiempo [s]	0 Translations
y axis label	unidades [u]	0 Translations
x Precision	3	
y Precision	3	
z Precision	3	
Min x value	0	
Max x value	350	
Min y value	0	
Max y value	270	

#### Graph

label=u vs t  
 aspectRatio=2.5  
 style=lines  
 lineWidth=3  
 color=ffff00  
 labelX=tiempo [s]  
 labelY=unidades [u]  
 xPrecision=3  
 yPrecision=3  
 zPrecision=3

Tabla 4. Información

Elemento: graph.

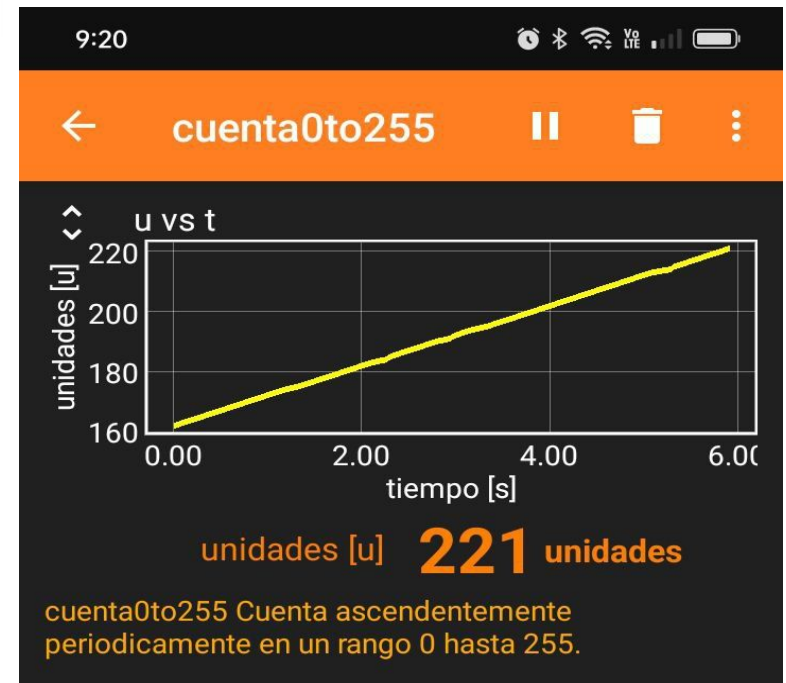
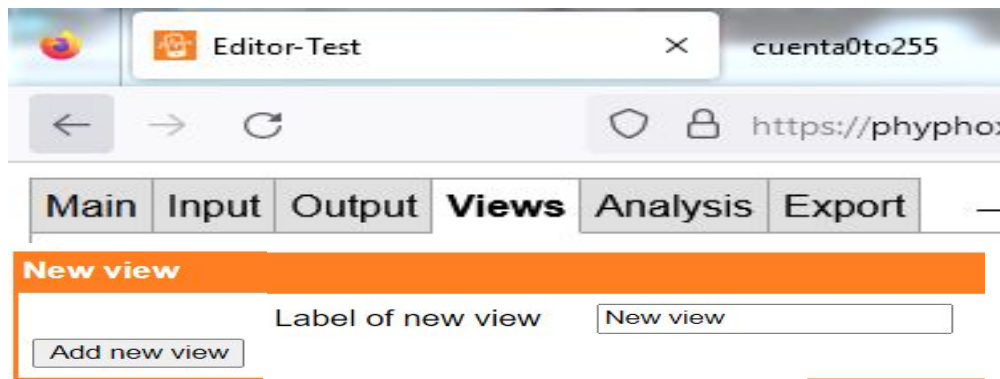


Figura 7. Configuración y vista gráfica.

Configuración elementos vista



3 Selección elementos de vista.

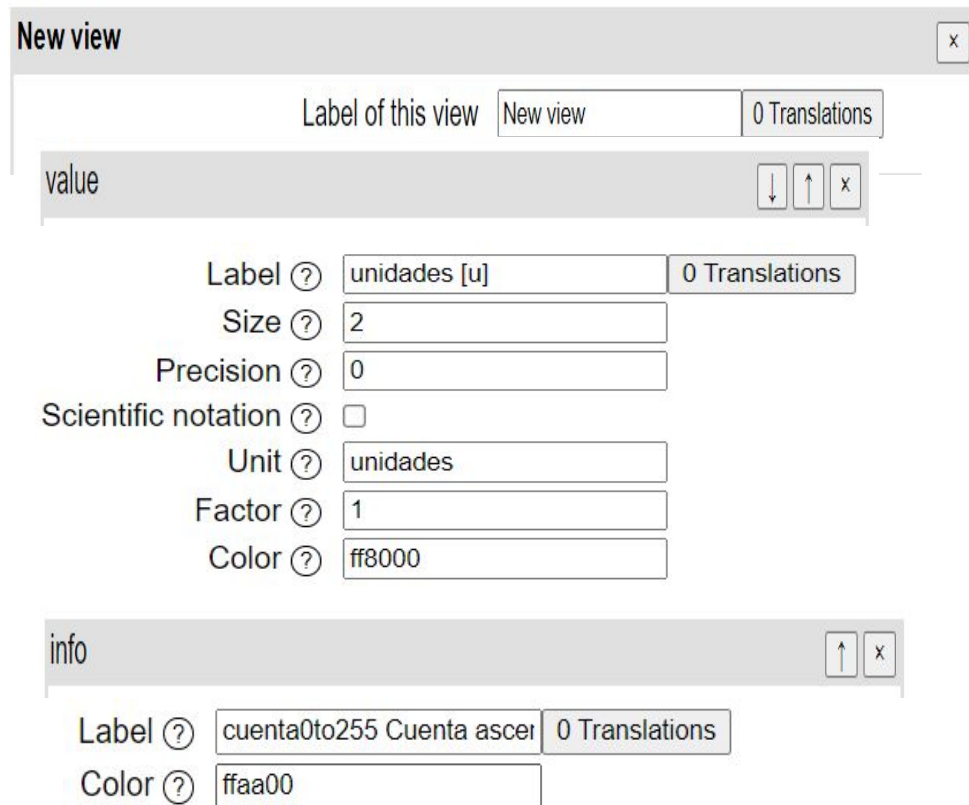
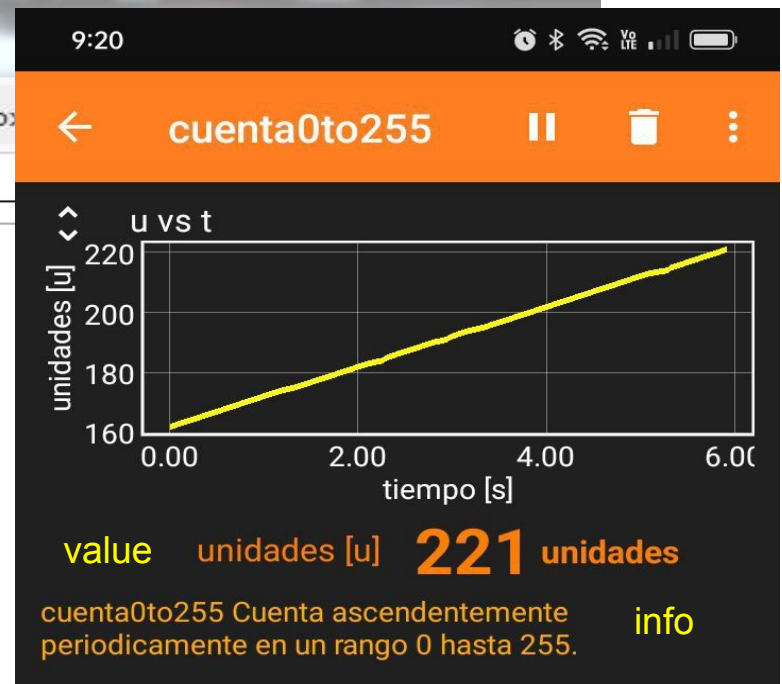


Figura 7a. Configuración y vista gráfica.



#### Value

label=unidades [u]  
Size=2  
Unit=unidades  
lineWidth=3  
Factor=1  
Color=ff8000

#### Info

Label=cuenta0to255 Cuenta ascendente periodicamente en un rango 0 hasta 255.  
Color=ffaa00

Tabla 4. Información elemento: value,info.

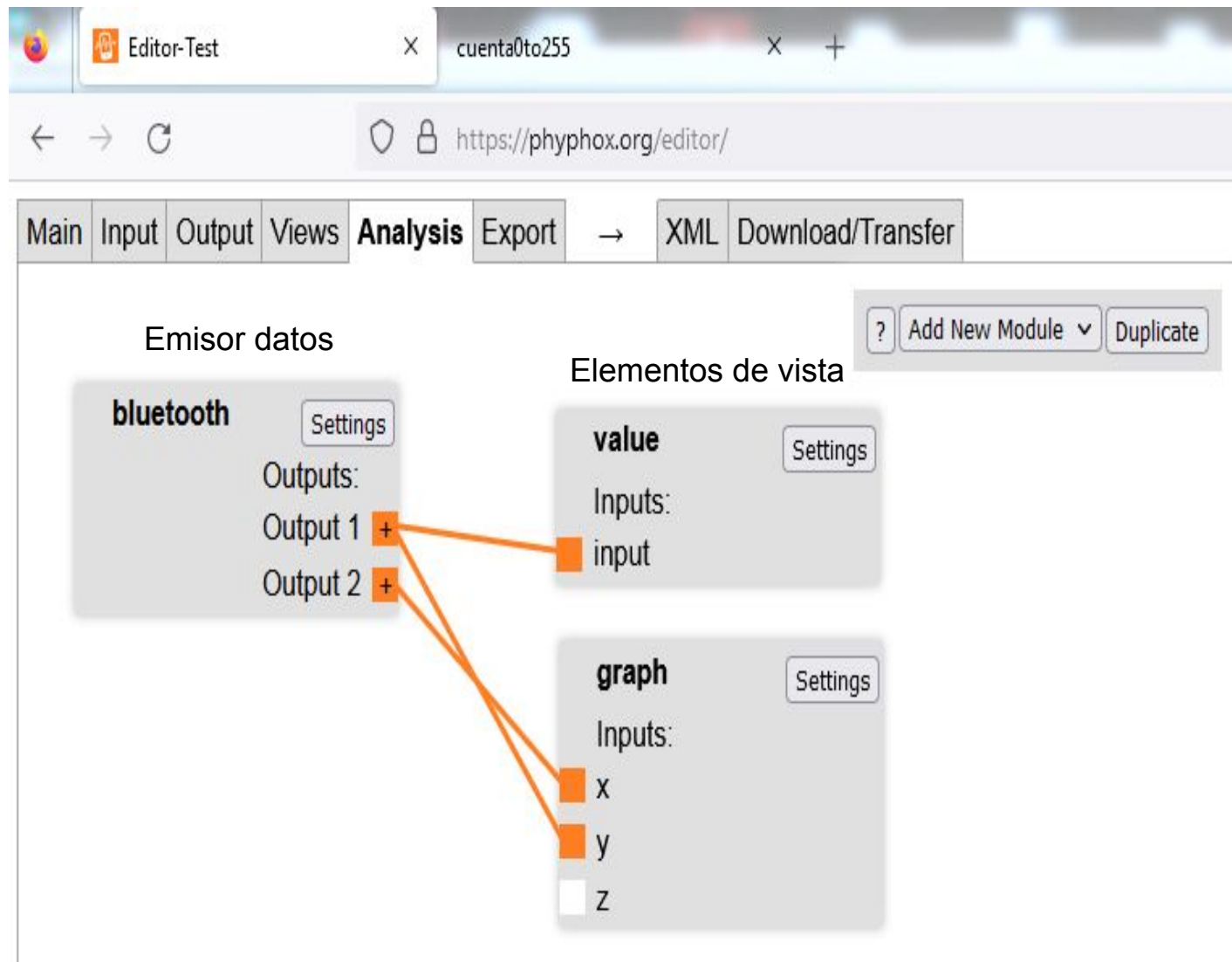


Figura 8. Relación emisor - elementos de vista.

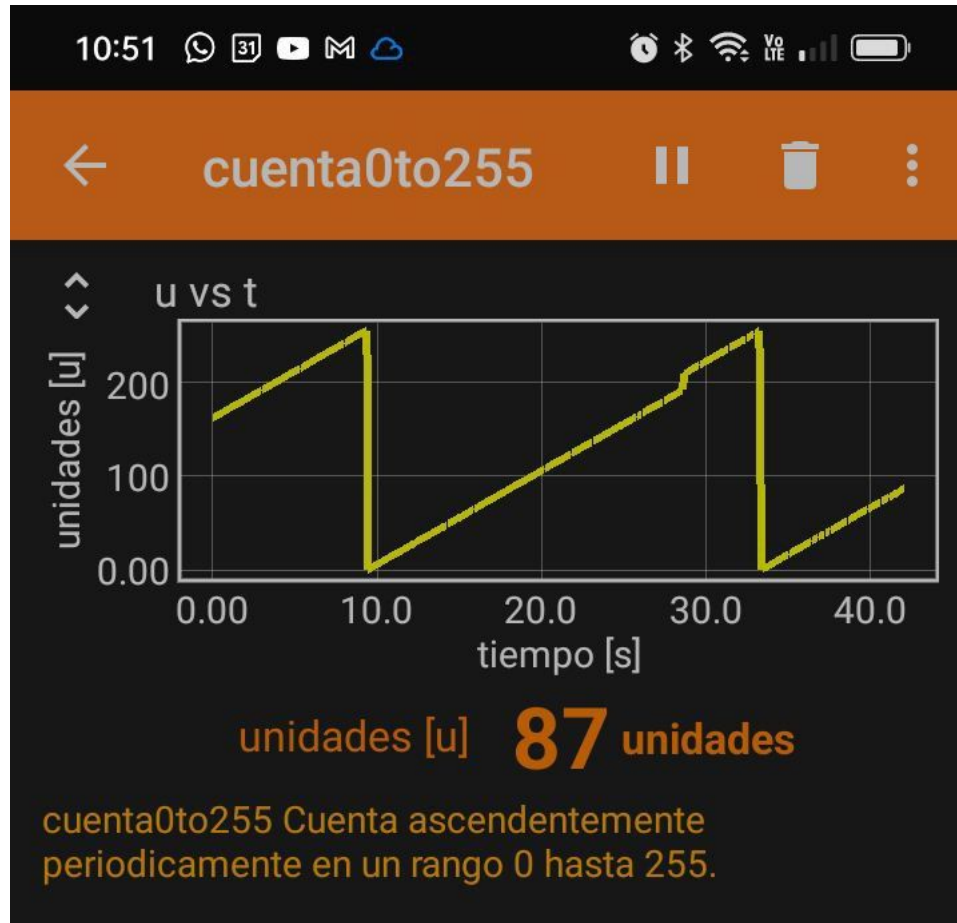


Figura 7a. Relación interface-rotulo interface usuario.



# Download/Transfer

## Download options

Online QR code (simple distribution) ?

Generate online QR-Code

Offline QR code (bulky, but no internet connection needed) ?

Generate offline QR-Codes

Download phyphox file (save your work to open in editor later) ?

Download phyphox-file

Download zip package (distribute as compact file) ?

Download package



Figura 9. Código 2d. Descarga de experimento.

```
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>
```

## Liga de descarga **BLE\_notifyCount.ino**

[https://drive.google.com/file/d/1EM42ksA-9uKr886d8gjkP\\_yYh22qJAV/view?usp=sharing](https://drive.google.com/file/d/1EM42ksA-9uKr886d8gjkP_yYh22qJAV/view?usp=sharing)

```
BLEServer* pServer = NULL;
BLECharacteristic* pCharacteristic = NULL;
bool deviceConnected = false;
bool oldDeviceConnected = false;
uint8_t value = 0;
```

```
// See the following for generating UUIDs:
// https://www.uuidgenerator.net/
```

Identificador único universal (UUID) es un número de 32 bytes globalmente único que se utiliza para identificar un servicio y tipo de datos en un perfil de atributo genérico.

```
#define SERVICE_UUID      "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```

```
class MyServerCallbacks: public BLEServerCallbacks {
    void onConnect(BLEServer* pServer) {
        deviceConnected = true;
    };

    void onDisconnect(BLEServer* pServer) {
        deviceConnected = false;
    }
};
```

Es un vector de caracteres que representa a un dato del dispositivo periférico y como acceder a este.

Figura 12. Variables de entorno

```

void setup() {
    Serial.begin(115200);

    // Create the BLE Device
    BLEDevice::init("ESP32");

    // Create the BLE Server
    pServer = BLEDevice::createServer();
    pServer->setCallbacks(new MyServerCallbacks());

    // Create the BLE Service
    BLEService *pService = pServer->createService(SERVICE_UUID);

    // Create a BLE Characteristic
    pCharacteristic = pService->createCharacteristic(
        CHARACTERISTIC_UUID,
        BLECharacteristic::PROPERTY_READ |
        BLECharacteristic::PROPERTY_WRITE |
        BLECharacteristic::PROPERTY_NOTIFY |
        BLECharacteristic::PROPERTY_INDICATE
    );

    // https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.descriptor.gatt.client_characteristic_configuration.xml
    // Create a BLE Descriptor
    pCharacteristic->addDescriptor(new BLE2902());

    // Start the service
    pService->start();

    // Start advertising
    BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
    pAdvertising->addServiceUUID(SERVICE_UUID);
    pAdvertising->setScanResponse(false);
    pAdvertising->setMinPreferred(0x0); // set value to 0x00 to not advertise this parameter
    BLEDevice::startAdvertising();
    Serial.println("Waiting a client connection to notify...");
}

```

UUID 0x2902 descriptor de la característica del cliente.

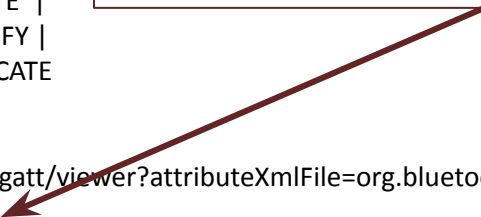


Figura 13. Configuración de recursos.

```

void loop()
{
  // notify changed value
  if (deviceConnected) {
    pCharacteristic->setValue((uint8_t*)&value, 4);
    pCharacteristic->notify();
    value++;
    Serial.println(value);
    delay(1000);
  }
  // disconnecting
  if (!deviceConnected && oldDeviceConnected) {
    delay(500); // give the bluetooth stack the chance to get things ready
    pServer->startAdvertising(); // restart advertising
    Serial.println("start advertising");
    oldDeviceConnected = deviceConnected;
  }
  // connecting
  if (deviceConnected && !oldDeviceConnected) {
    // do stuff here on connecting
    oldDeviceConnected = deviceConnected;
  }
}

```

conteo ascendente periódico de números enteros  
en un rango de 0 hasta 255; transmite el estado del conteo

Figura 14. Configuración de recursos.

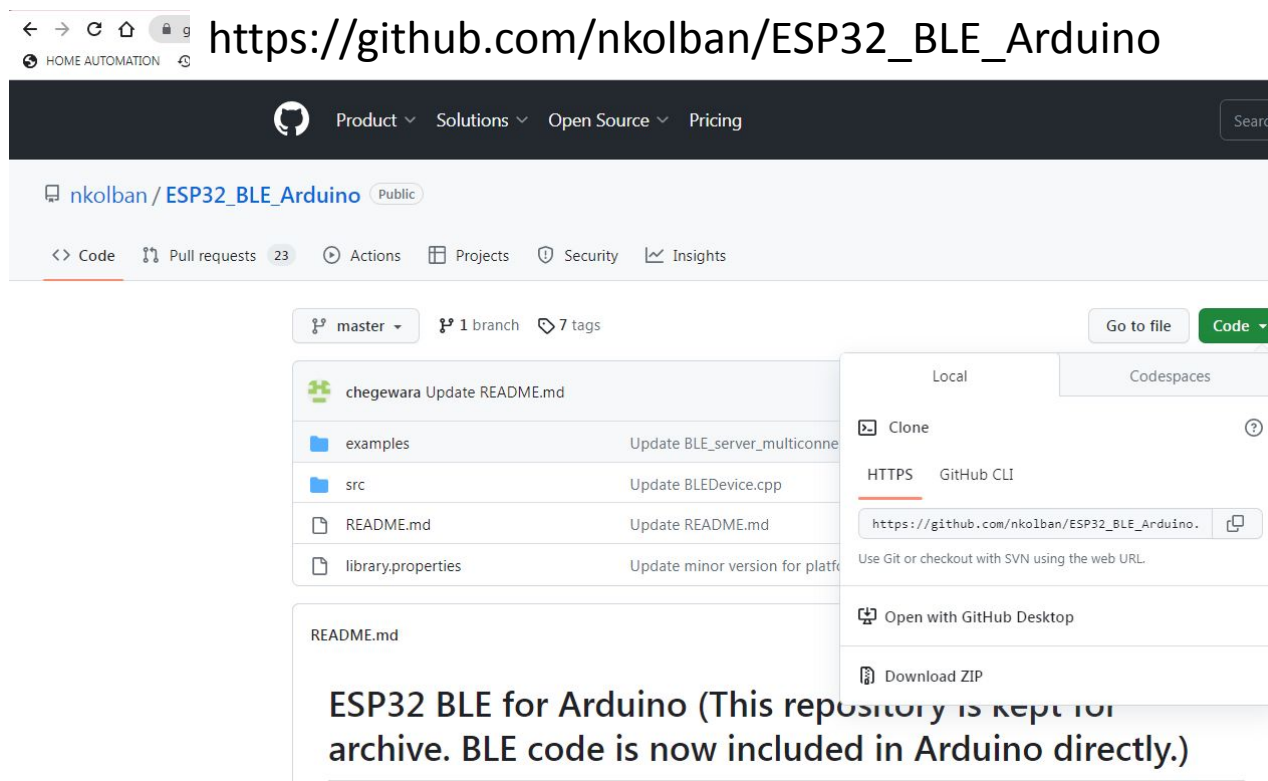


Figura 15. Instalación de la biblioteca ESP32 BLE Arduino.

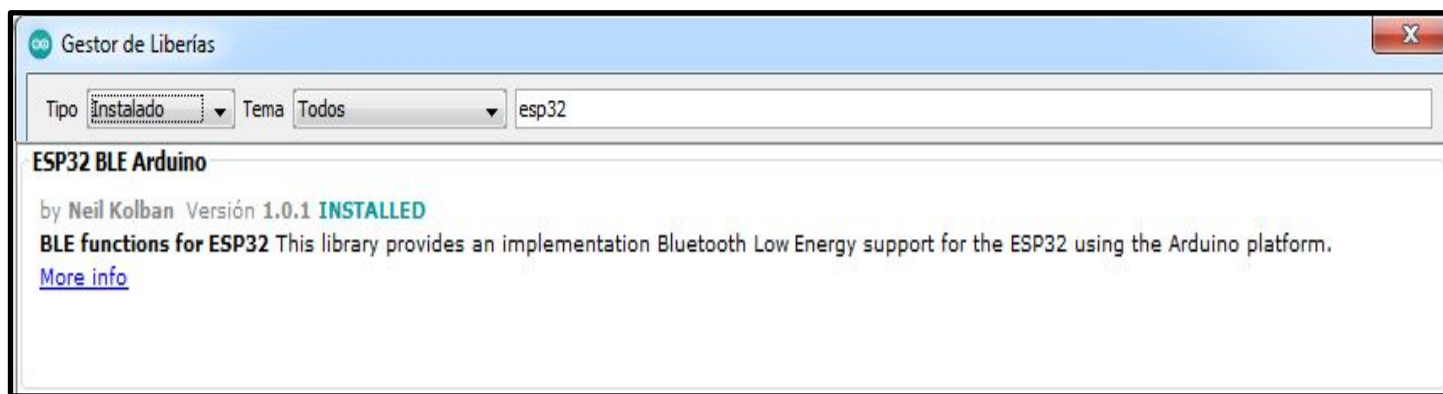


Figura 15a. Instalación de la biblioteca ESP32 BLE Arduino gestor librerías.



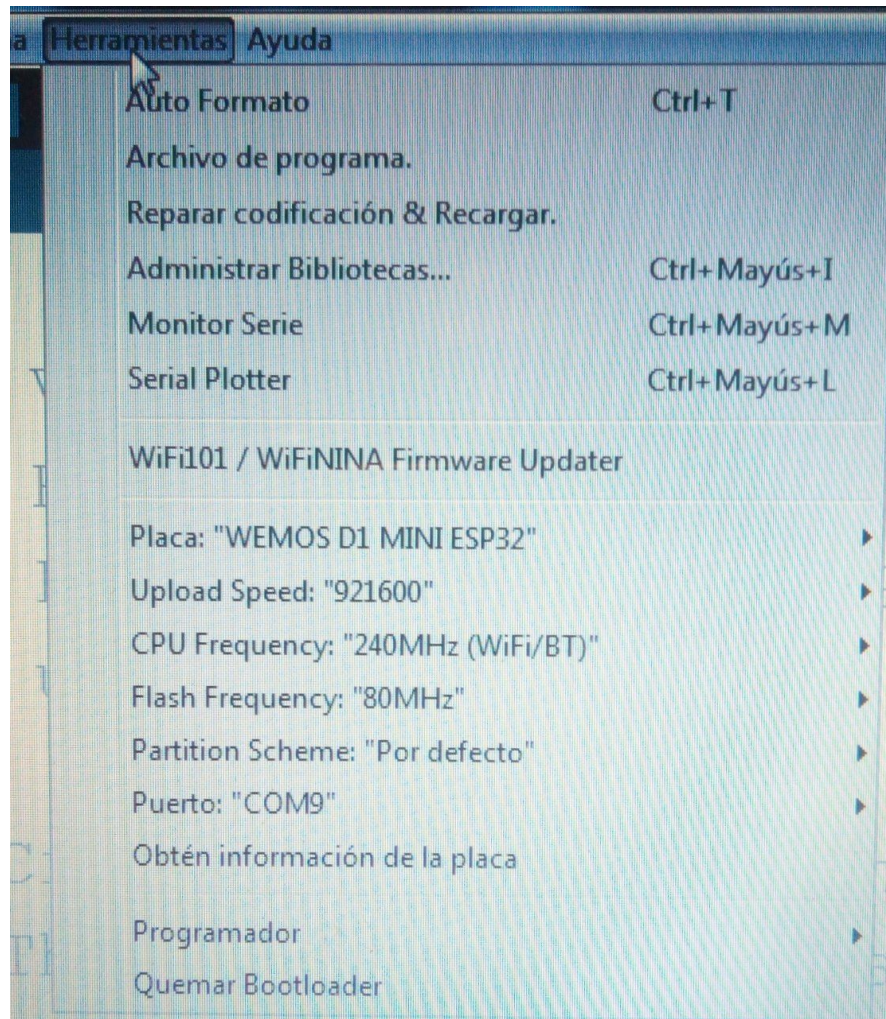


Figura 15. Prueba y validación del sistema integrado tipo Arduino: WeMos ESP32 D1 R32

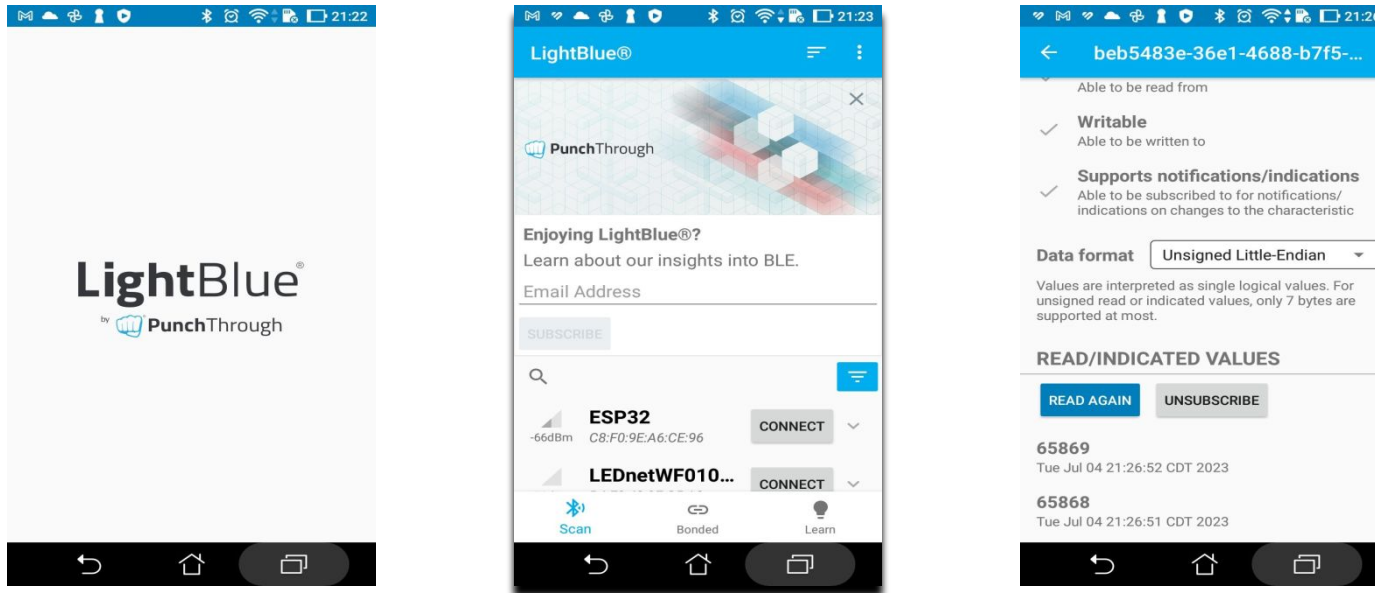
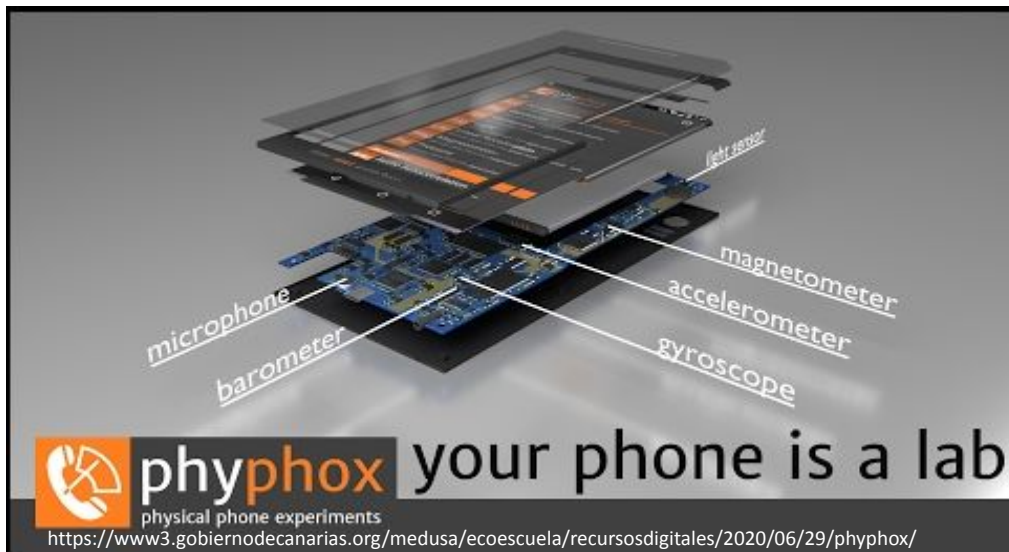


Figura 15. Prueba y validación de la herramienta de monitorización.

**Ligthblue** es una aplicación compatible con dispositivos móviles (teléfonos, tabletas) compatibles con el sistema operativo Android, hace uso de protocolo de comunicaciones Bluetooth Low Energy para conexiones punto a punto en distancias menores a 10m; esta aplicación es una herramienta de soporte de lectura, escritura, notificación de datos UTF-8 y monitorización de identificadores de dispositivo (UUID, CHARACTERISTIC\_UUID, MAC, DEVICE\_NAME)

Disponible en: <https://play.google.com/store/search?q=ligthblue&c=apps&hl=es-419>



### Sensores embebidos compatibles:

- acelerómetro.
- magnetómetro.
- giroscopio.
- presión atmosférica
- micrófono.
- proximidad.
- intensidad luminosa.
- GPS.

**Phyphox** es una aplicación compatible con el sistema operativo iOS y Android, es capaz de ejecutar una colección de experimentos predefinidos, propios y de software de código abierto con los sensores electrónicos embebidos compatibles y recursos en un teléfono móvil y, o tableta electrónica, puede exportar los datos adquiridos en formatos .xlsx, csv y tsv, así como controlar de forma remota un experimento a través de una interface web, desde cualquier equipo de cómputo y móvil en la misma red empleando un navegador web.

**Phyphox** es una aplicación desarrollada y actualizada por la Universidad técnica de Aquisgrán, Alemania; defina sus propios experimentos de física seleccionando entradas de sensores, definiendo pasos de análisis y creando vistas como una interfaz, utilizando nuestro editor web <http://phyphox.org/editor>.

Disponible en:

<https://play.google.com/store/search?q=pyhphox&c=apps>

UUID	Del inglés <i>universal unique identifier</i> , es un número de 128-bit usado para identificar una única entidad. Es un identificador único de longitud 128 bits Representado por 32 caracteres hexadecimales separados en campos 4fafc201-1fb5-459e-8fcc-c5c9c331914b Time_low- time_mid-time_hi_and_version-clock_seq_hi_and_res-node	<a href="https://coderolls.com/uuid-in-java/">https://coderolls.com/uuid-in-java/</a>

## 1] ESPECIFICACIONES Y CARACTERÍSTICAS

Modelo: WeMos ESP32 D1 R32

MCU: ESP32-WROOM-32

CPU: Dual core Tensilica LX6 (32 bit)

Frecuencia de Reloj: 240MHz

ROM: 448 kB SRAM: 520kB, Memoria Flash: 4MB

Interfaz de comunicación: UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S,

IR, contador de pulsos, GPIO, Sensor Táctil Capacitivo, ADC, DAC

Seguridad: WPA, WPA2, WPA2-Enterprise, WPS

Voltaje de entrada MicroUSB: 5 V (500 mA máx.)

Voltaje de entrada VIN Power Jack: 7V a 16V DC

Consumo de corriente: Máximo 250mA, Modo ahorro: 0.15mA y Funcionamiento (sin WiFi): 20mA

Convertidor de USB a serie CH340

1]<https://uelectronics.com/producto/wemos-esp32-d1-r32/>