Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Cheshire Cap

¡Creemos un sombrero mágico!

Autora:
Inés Jiménez Díaz
Twitter: @httpsrim
Instagram: _https.rim_

CÍVICA PEOPLE BEYOND TECH
certinia
CGI
rti
nazaríes intelligenia
UNIT4
Financiado por la Unión Europea NextGenerationEU
incibe_
UNIVERSIDAD DE GRANADA
Si2 Soluciones
DCOOP Tierras con Alma
MONTAJES ELECTRICOS JIMENEZ
ETSIIT Escuela Técnica Superior de Ingenierías Informáticas y de Telecomunicación

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Hardware necesario para Cheshire Cap

Chesire cap

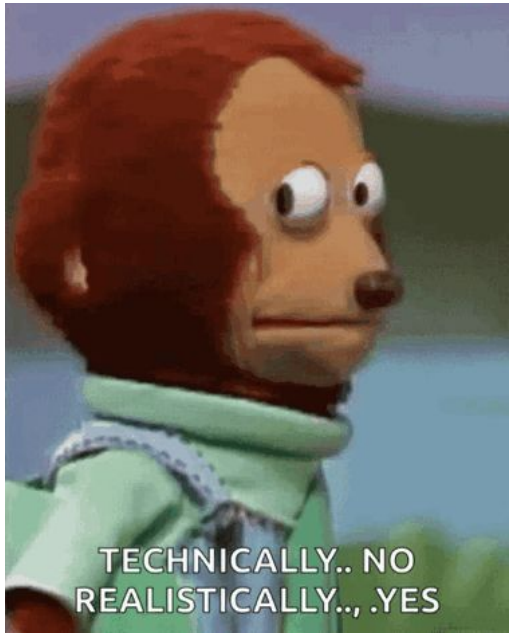
- Arduino Uno
- Módulo bluetooth HC-05
- Tira de Leds Adafruit
- Pantalla OLED
- Cables arduino
- 3 Servomotores

Antes de empezar...

¿Qué tengo que hacer para cargar una librería?

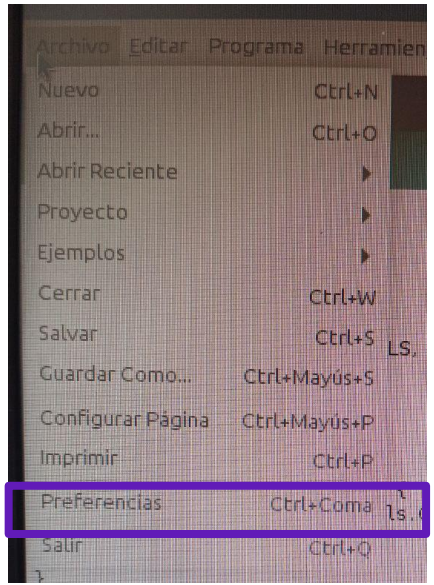
Cargar librería

- **Las librerías las vamos a tener que cargar TODOS los días en los ordenadores del campus.**
- Pero primero, antes de cargarlas, tenemos que configurar el IDE….



Cargar librería

- Para ello, entramos en el IDE de arduino, y nos vamos a preferencias.

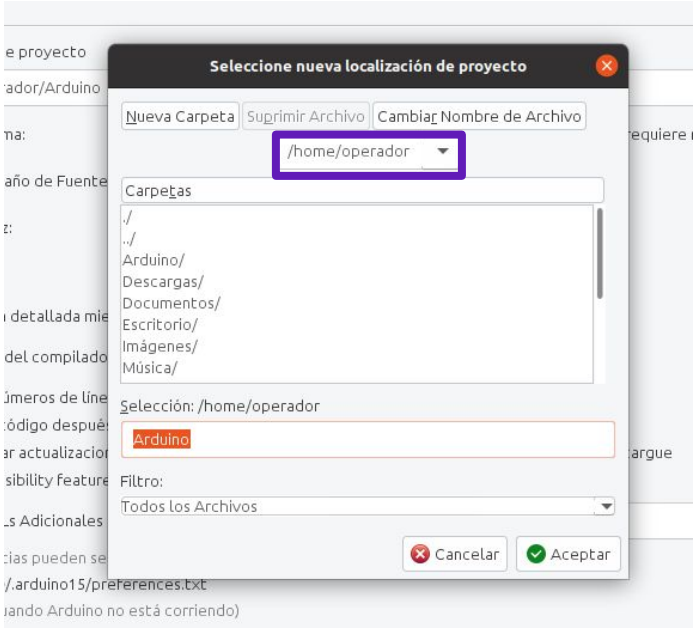


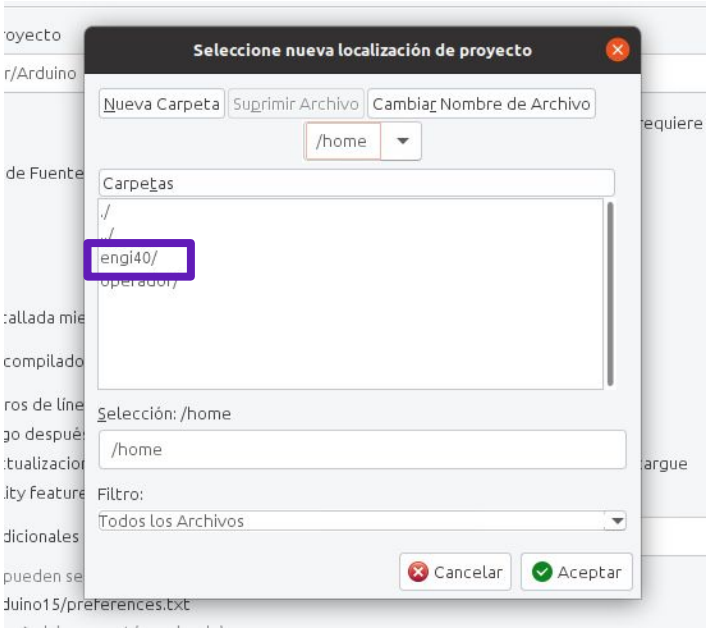
Cargar librería

- Aquí nos saldrá una pestaña como la que sale aquí, vemos que sale “Operador”.
- Esto significa que se está usando la carpeta de operador, pero nosotras queremos nuestra carpeta “engiXX”.
- Para ello…

Cargar librería

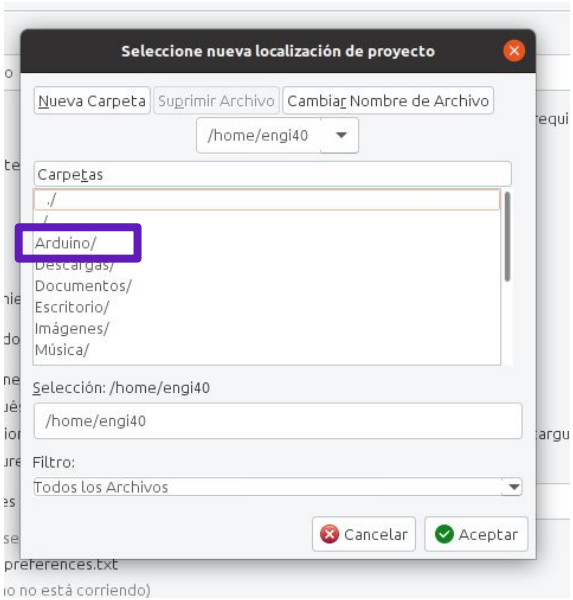
Para ello…

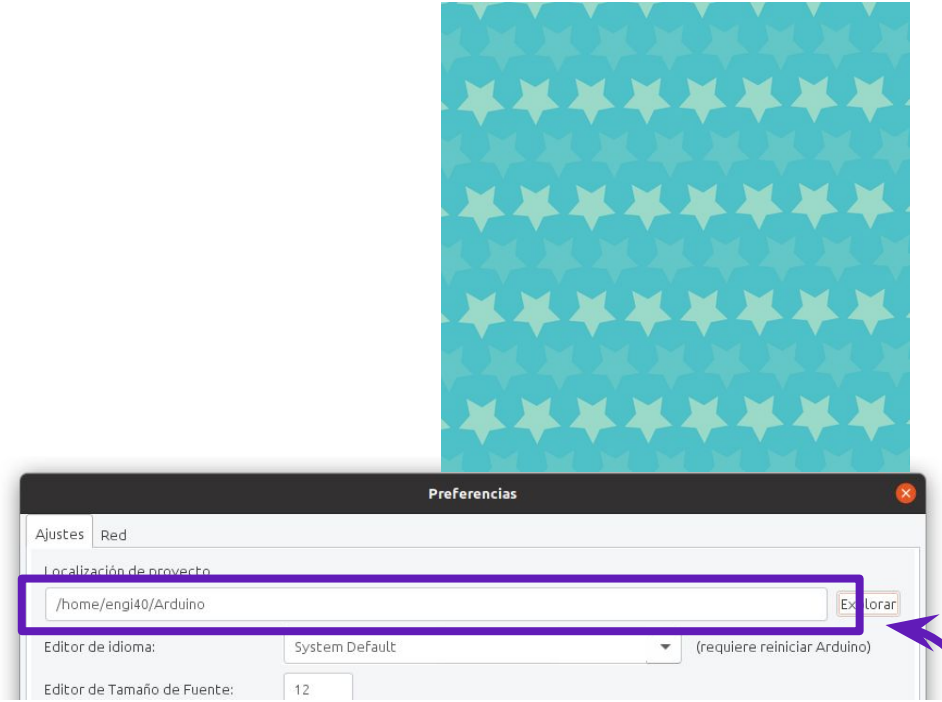




Cargar librería

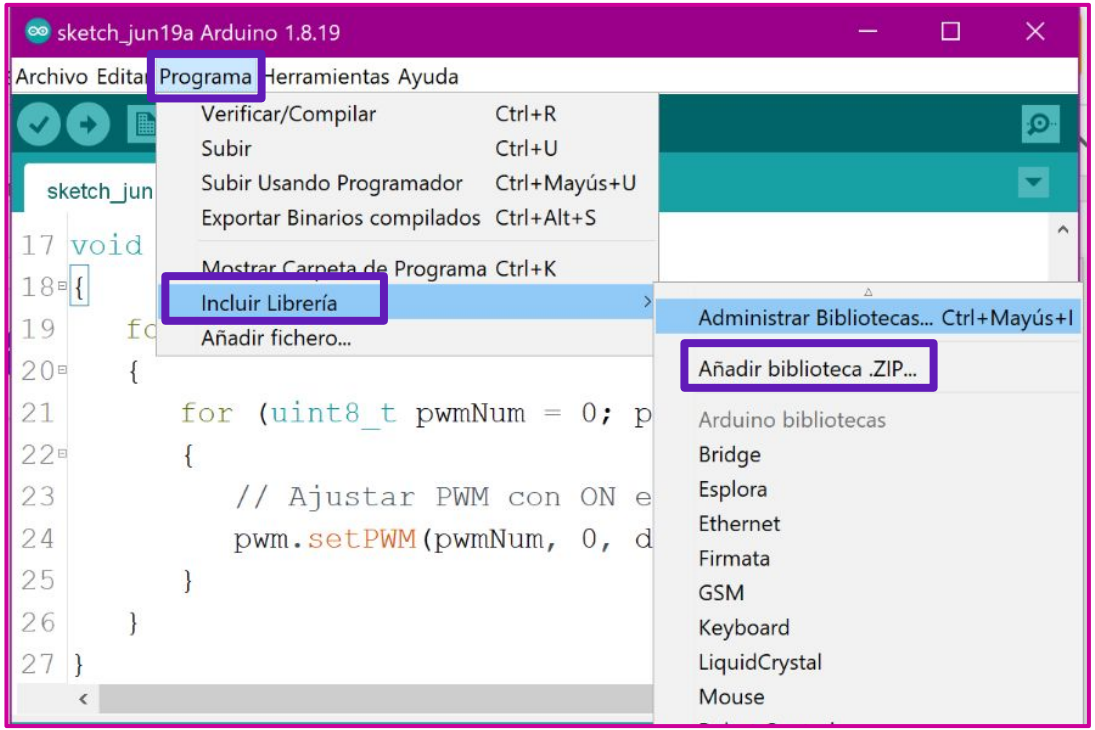
Para ello…





Así debería de quedar

Cargar librería

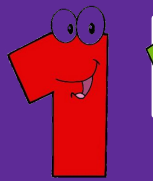


Ya después de esta configuración,

podemos añadimos las **librerías** necesarias.

Repetir este paso para cada una de las librerías.

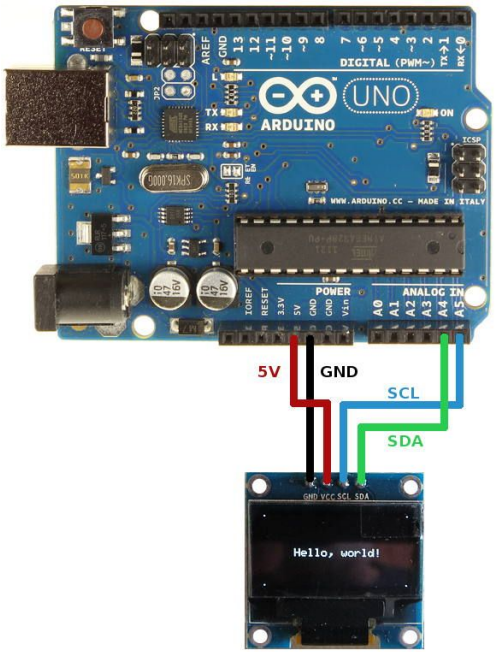
Montaje del SOMBRERO



¡Pantalla OLED!

Pantalla OLED

- Tienen 4 pines:
  - **SCA**
  - **SCL**
  - **VCC** proporciona energía a la pantalla, 3.3 V
  - **GND** Toma de tierra



Pantalla OLED

Vamos a hacer que aparezca en la pantalla un Hola Mundo:

Abre un nuevo sketch en Arduino

1. Instalar librerías: Herramientas - instalar librerías
   - Adafruit GFX
   - Adafruit SSD1306
2. **Copia y pega el código de la derecha**
3. Carga el programa

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels

#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(9600);
  // Comprbamos que la pantalla funciona correctamente
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  // Iniciamos la librería
  //se inicia con el logo de Adafruit
  display.display();
  delay(2000); // Pausa de 2 segundos

  // Limpiamos la pantalla
  display.clearDisplay();

  delay(2000);

}

void loop() {
  display.clearDisplay();

  display.setTextSize(1);  // tamaño de letra 1
  display.setTextColor(SSD1306_WHITE);// Texto en blanco
  display.setCursor(0,0);             // Empieza a escribir arriba a la izquierda
  display.println(F("Hola Mundo!"));

  display.setTextColor(SSD1306_BLACK, SSD1306_WHITE); // Fondo blanco y texto negro
  display.println("Hola Mundo!");

  display.setTextSize(2);             //Tamaño de letra 2
  display.setTextColor(SSD1306_WHITE);//Texto en blanco
  display.println("Hola Mundo!");

  display.display();
  delay(2000);
}
```

Funciones útiles


```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels

#define OLED_RESET     -1 // Reset pin # (or -1 if sharing
Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address;
0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, & Wire,
OLED_RESET);
```

Declaración de variables

```
void setup() {
  Serial.begin(9600);
  // Comprbamos que la pantalla funciona correctamente
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS )) {
    Serial.println(F("SSD1306 allocation failed" ));
    for(;;);
  }
  // Iniciamos la librería
  //se inicia con el logo de Adafruit
  display.display();
  delay(2000); // Pausa de 2 segundos

  // Limpiamos la pantalla
  display.clearDisplay();

  delay(2000);
}
```

Inicialización

Funciones útiles

- Limpiar la pantalla
- Tamaño del texto
- Color del texto
- Empezar a escribir
- Lo que escribe
- Color del fondo
- Enseña lo que pone por pantalla

```
void loop() {
  display.clearDisplay();

  display.setTextSize(1);  // tamaño de letra 1
  display.setTextColor(SSD1306_WHITE);// Texto en
blanco
  display.setCursor(0,0);             // Empieza a
escribir arriba a la izquierda
  display.println(F("Hola Mundo!"));

  display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
// Fondo blanco y texto negro
  display.println("Hola Mundo!");

  display.setTextSize(2);             //Tamaño de
letra 2
  display.setTextColor(SSD1306_WHITE);//Texto en
blanco
  display.println("Hola Mundo!");

  display.display();
  delay(2000);
}
```

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

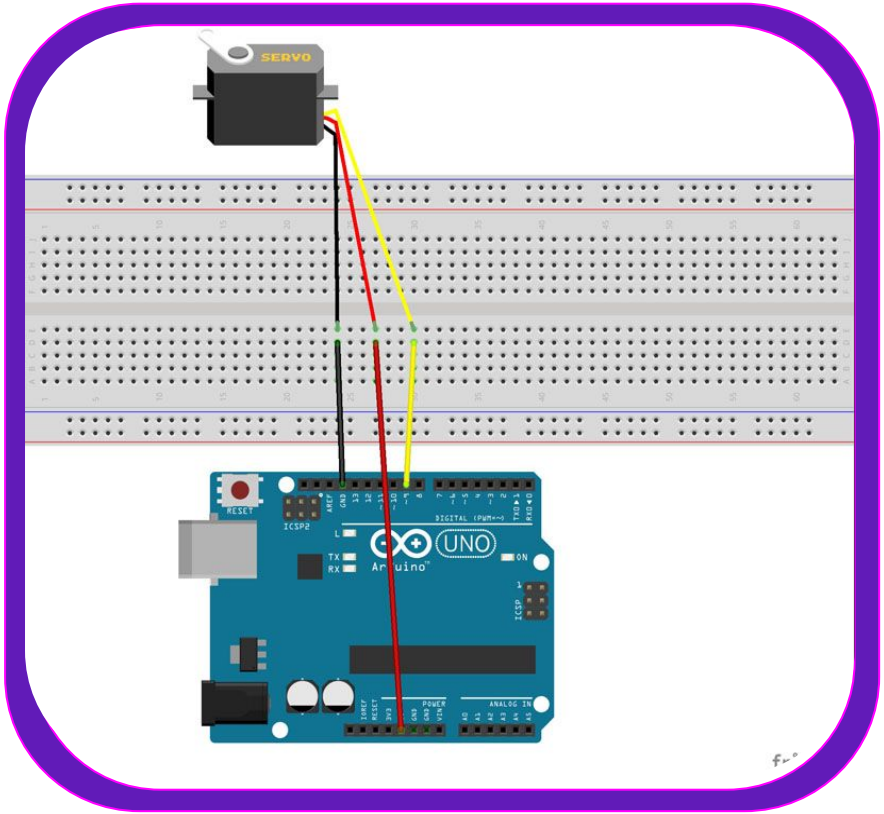
Montaje del SOMBRERO

Servomotores

Servomotores

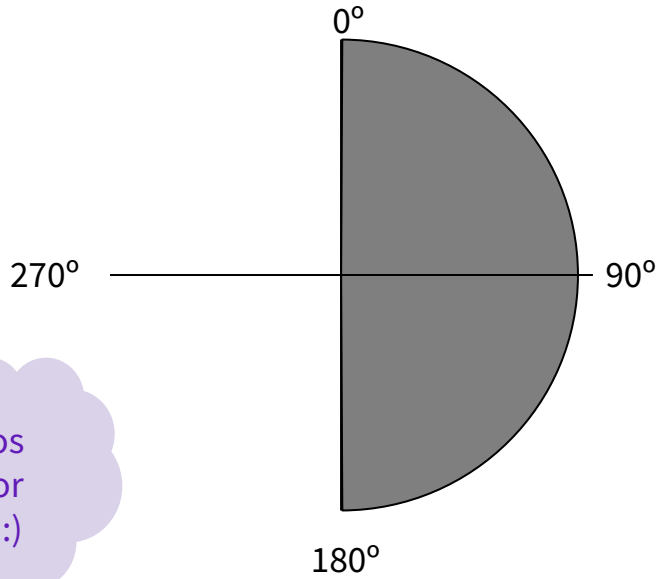
- **Servo motor 180º.**
  - Tienen 3 pines:
    - **DIN** pin de entrada **PWM,** tienen un **~**
    - **+5V** proporciona energía al servo. **VCC**
    - **GND** Toma de tierra.**GND**

¡Ojo! no te equivoques al conectarlo que puedes quemar el motor.



ServoMotor

Los servomotores tienen un rango de movimiento limitado, que es de 0 a 180 grados.



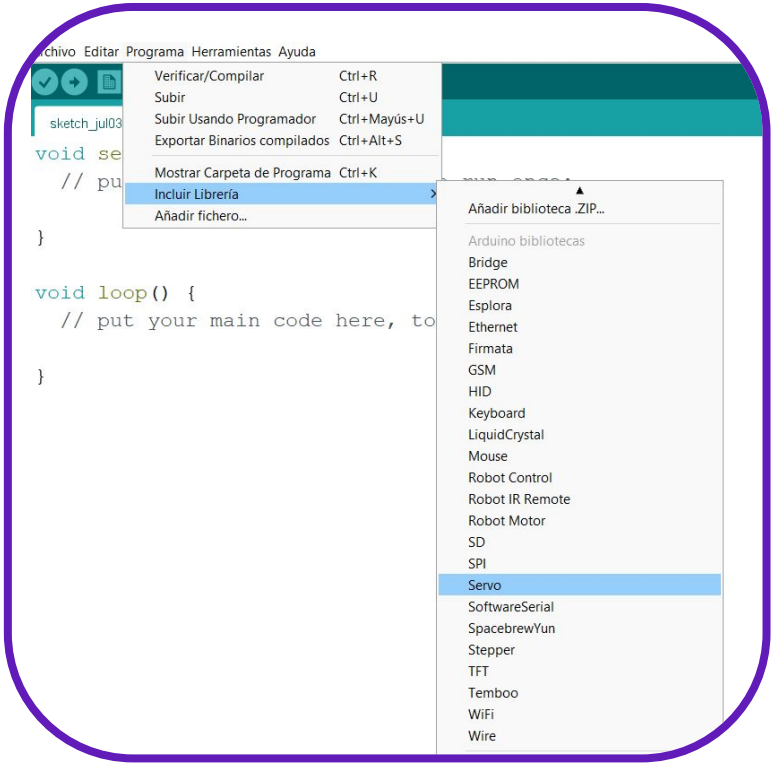
Os dejo los grados por si dudáis :)

SERVOMOTOR

2. ¡Comencemos a programar!

Para este servomotor hace falta incluir una **librería**, para ello pulsamos en:

1. Programa
2. Incluir librería
3. pinchamos en: Servo



ServoMotor

Y para usar el servomotor hay que asociar el objeto **servo** al pin del servo:

servo §

```
#include <Servo.h>

Servo servoMotor; // Crea un objeto Servo

int servoPin = 8; // Pin al que está conectado el servo

void setup() {
  servoMotor.attach(servoPin); // Asocia el objeto Servo al pin del servo
}
```

Servomotores

Este es un pequeño ejemplo de cómo funciona:

Los bucles for son para que gire los grados que queramos.

Dentro del bucle, servo1.write(pos) es para que cambie el servo a ese grado.

NOTA 1: Aquí gira para un sentido

NOTA 2: Aquí gira para el otro sentido.

```
for (pos = 0; pos <= 50; pos += 1) {
  servo1.write(pos);
  delay(15);
}
delay(1000);
for (pos = 50; pos >= 0; pos -= 1) {
  servo1.write(pos);
  delay(15);
}
```

Servomotores

Utilizamos el servo - Vamos a hacer que gire 90 º cada segundo:

- **servoMotor.write(0);**
  - El servomotor se pone a 0 grados
- **delay(1000);**
  - Esperamos 1 segundo

servo §

```
#include <Servo.h>

Servo servoMotor; // Crea un objeto Servo

int servoPin = 8; // Pin al que está conectado el servo

void setup() {
  servoMotor.attach(servoPin); // Asocia el objeto Servo al pin del servo
}
```

escribimos el punto 0 en el objeto servo
esperamos 1 seg

. . . vosotras . . .

(Hasta 180)

```
}
```

Montaje del SOMBRERO

¡Hágase la luz!

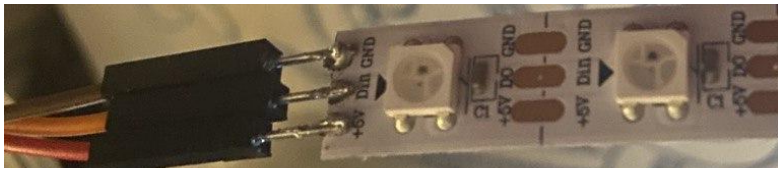
¡HÁGASE LA LUZ!

- **Leds RGB direccionables.**
  - Tienen 4 pines:
    - **DIN** pin de entrada
    - **DO** pin de salida
    - **+5V** proporciona energía a los leds
    - **GND** Toma de tierra

Antes de cortar los leds fijaros que los **DIN** están conectados a los **DO** recuerdalo cuando vayas a incluirlos en el proyecto final al conectarlos.

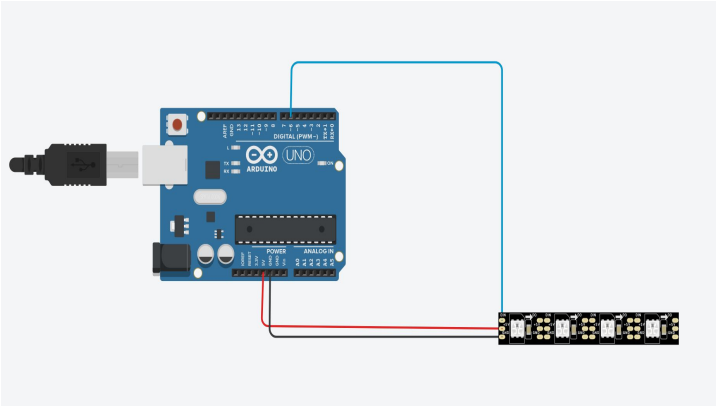
¡HÁGASE LA LUZ!

- **Leds RGB direccionables.**
- Se llaman direccionables porque las señales se transmiten en una dirección a través de los leds.
- **¡Fíjate cómo están conectados!**



¡HÁGASE LA LUZ!

- **Leds RGB direccionables.**
- Se llaman direccionables porque las señales se transmiten en una dirección a través de los leds. **Del Arduino al primer led, al segundo, al tercero… hasta el último**



1. Incluir en el IDE la librería Adafruit_NeoPixel

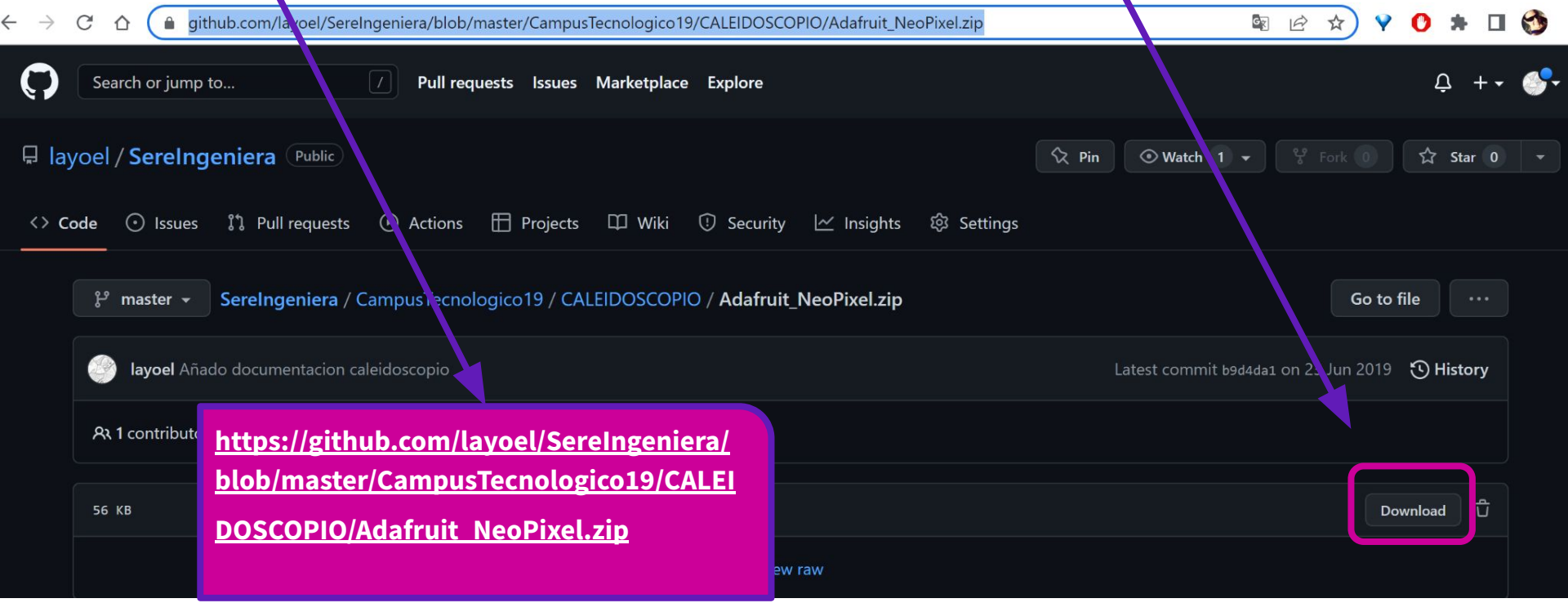
- Hay **dos formas de incluir librerías** a la biblioteca de nuestro arduino.
  - Si tenemos el archivo comprimido con extensión .zip
  - Buscarlas en el repositorio de Arduino

- Para la tira de led tenemos el archivo comprimido con el nombre **Adafruit_NeoPixel.zip**
- La **descargamos** y la incluimos de la 1º forma.

¡HÁGASE LA LUZ!

1. Incluir en el IDE la librería Adafruit_NeoPixel

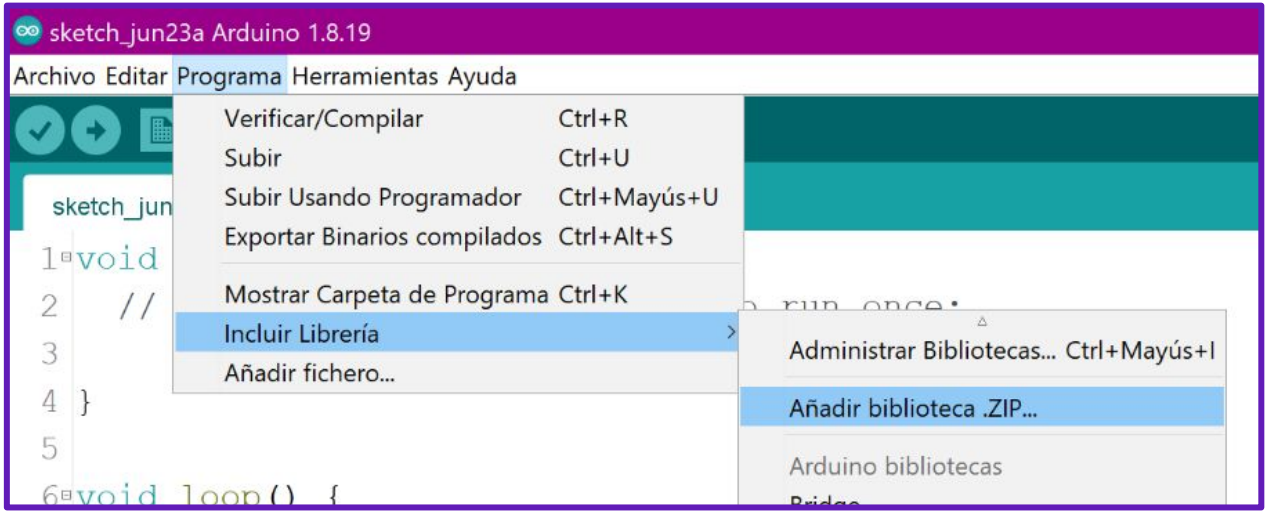
Vamos al enlace y descargamos la librería haciendo clic en download



https://github.com/layoel/SereIngeniera/blob/master/CampusTecnologico19/CALEIDOSCOPIO/Adafruit_NeoPixel.zip

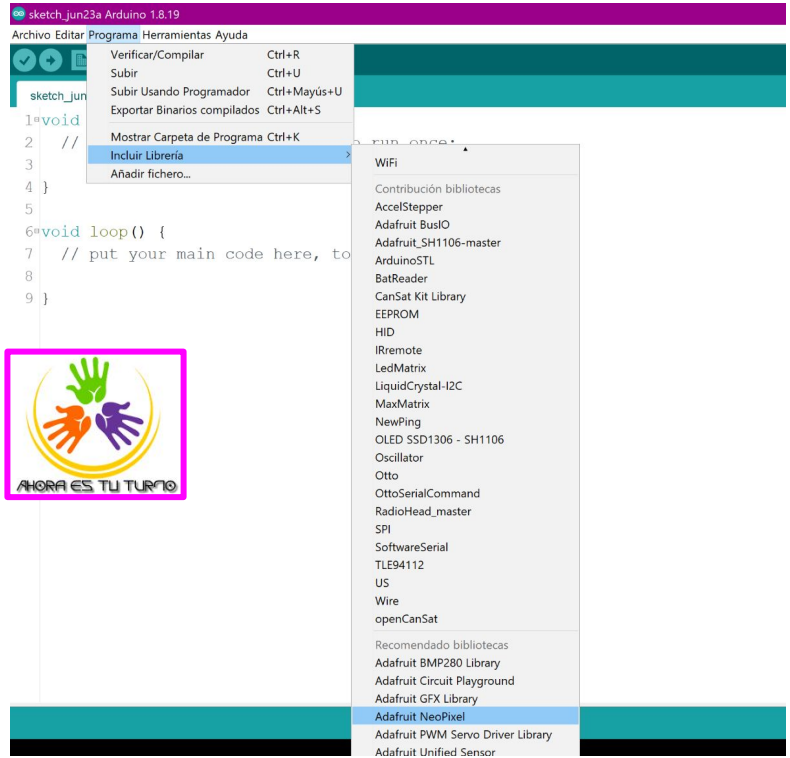
¡HÁGASE LA LUZ!

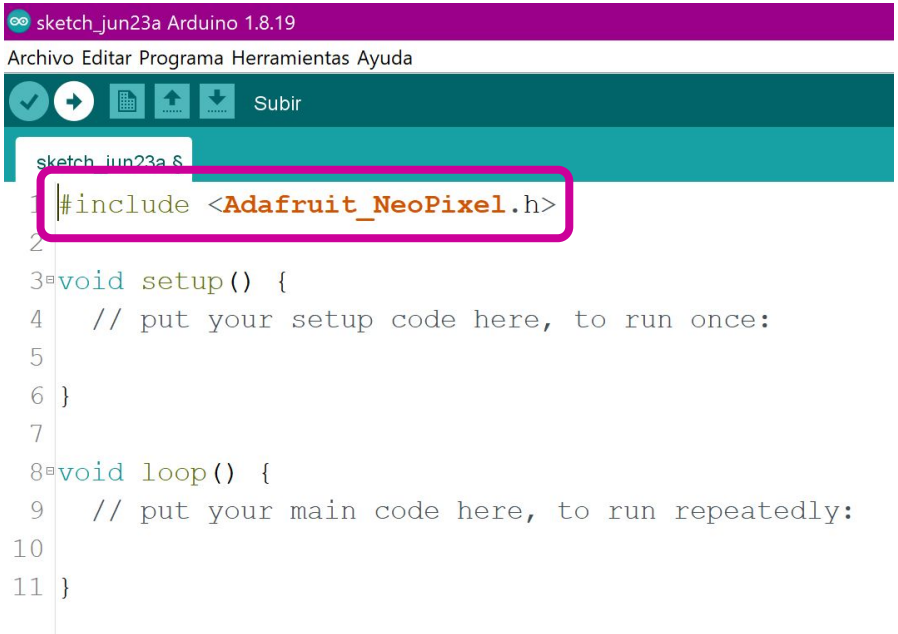
1. Incluir en el IDE la librería Adafruit_NeoPixel



¡HÁGASE LA LUZ!

2. Para poder usar la librería tenemos que añadirla en el sckech.





3. Declaramos un objeto con la cadena de leds que vamos a emplear.

- Para ello utilizaremos la siguiente **función de la librería NeoPixel** que acabamos de incluir.
  - **Adafruit_NeoPixel pixels(**NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800**);**
- Los parámetros que se pasan a la función son:
  - **NUMPIXELS** que será una variable donde pondremos el número de leds que queremos controlar
  - **PIN** que será el pin de arduino donde hemos conectado el **IN** de la tira de led
  - **NEO_GRB + NEO_KHZ800** este parámetro indica que usaremos leds de colores y la controladora que utilizan.
- Vale pero, ¿qué tenemos que hacer y cómo?

¡HÁGASE LA LUZ!

3. Declaramos un objeto con la cadena de leds que vamos a emplear.

- Aquí tenéis **un ejemplo para controlar 3 leds**, vuestra tarea es adaptarlo para poder usar los 8 leds.

Esto hay que hacerlo en el scketch donde vamos a crear el piano

sketch_jun23a Arduino 1.8.19

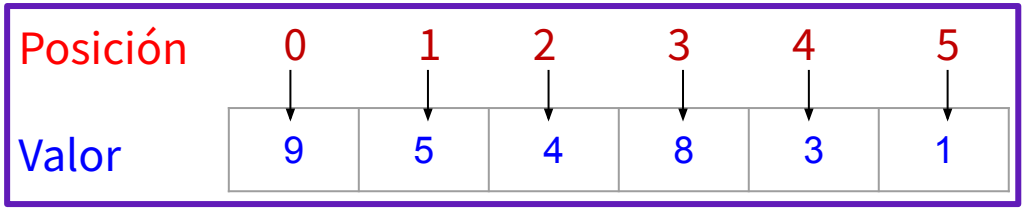
Archivo Editar Programa Herramientas Ayuda

sketch_jun23a §

```
#include <Adafruit_NeoPixel.h>

#define PIN        2
#define NUMPIXELS 3

using namespace std;

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

4. Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.

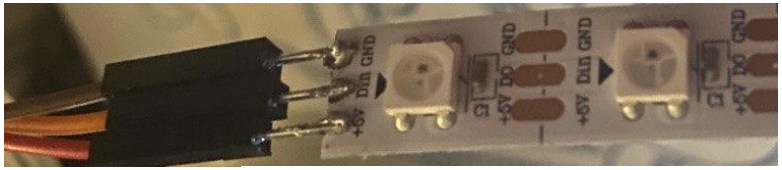
- Antes de empezar a usar las funciones de la librería necesitamos saber cómo acceder a las posiciones de cada led.
- ¿Sabes que es un **vector**?



- El vector tiene **posiciones**
- La primera posición **siempre empieza en 0**
- **Cada posición** tiene **un valor** (en la imagen, la posición 0 tiene valor 9, la posición 4 tiene valor 3…)
- El tamaño máximo del vector es el número de posiciones que tiene. En este ejemplo el vector tiene **6 posiciones que van del 0 al 5**.
- Los led se encienden en el mismo orden que el vector del 0 al 8 en nuestro caso.

¡HÁGASE LA LUZ!

4. **Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.**

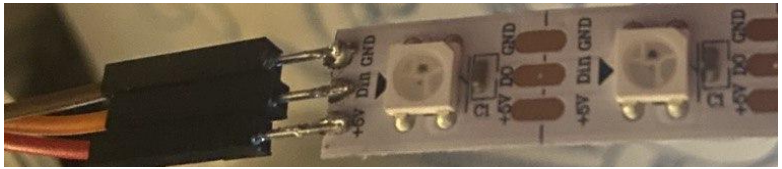


- ¿Cómo encedemos un led?

¡HÁGASE LA LUZ!

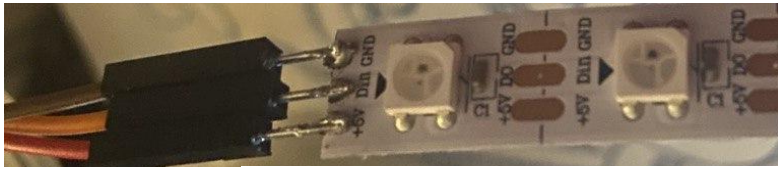
- **Peero antes de seguir...Debemos de soldar la tira de leds**

- Así debería de quedar la soldadura!!!



¡HÁGASE LA LUZ!

4. **Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.**



- ¿Cómo encedemos un led?

- **¡Usaremos la librería Neopixel!**

¡HÁGASE LA LUZ!

- Pero…

CUIDADO

CON

QUEMARSE!!

4. Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.

- Utilizaremos estas funciones de la librería Neopixel.
  - **pixels.begin();** esta función inicializa los leds.
  - **pixels.clear();** esta función borra la configuración guardada en los leds.
  - **pixels.Color(R, G, B);** esta función es para seleccionar el color.
    - **R** es el parámetro donde se pondrá el valor del color **rojo** (toma valores de 0 a 255)
    - **G** es el parámetro donde se pondrá el valor del color **verde** (toma valores de 0 a 255)
    - **B** es el parámetro donde se pondrá el valor del color **azul** (toma valores de 0 a 255)
  - **pixels.setPixelColor(led, color);**
    - **led** es el número de led que queremos que se encienda del **color** que pongamos.
    - **color** es el color que se va a poner y se sustituye por: **pixels.Color(R, G, B);**

¡HÁGASE LA LUZ!

4. Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.

- En este ejemplo se usan 6 leds.
  - **Prueba el código en tu scketch**
    - ¿Qué leds se encienden?
    - ¿En qué color?
  - **Adapta el código para los 4 leds**
  - **Ve haciendo pruebas para encender uno u otro led o varios a la vez y probar los diferentes colores.**


```
#include <Adafruit_NeoPixel.h>

#define PIN        2
#define NUMPIXELS 3

using namespace std;

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int tablero[]={0,0,0
               0,0,0};

void setup() {
  // put your setup code here, to run once:
  pixels.begin();

}

void loop() {
  // put your main code here, to run repeatedly:
  pixels.clear();
  pixels.setPixelColor(0, pixels.Color(30, 0, 0));
  pixels.setPixelColor(1, pixels.Color(0, 30, 0));
  pixels.setPixelColor(2, pixels.Color(0, 0, 30));
  pixels.setPixelColor(3, pixels.Color(30, 30, 30));
}
```

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Para poder hacer esto, vamos a aprender a escribir condicionales.
  - **Condicional simple**
    - **Si** es cierto… **entonces…**
  - **Condicional doble**
    - **Si** es cierto… **entonces**…. **sino**… **entonces**…
  - **Condicional compuesto**
    - **Si** esto **o** esto es cierto… **entonces…**
    - **Si** esto **y** esto es cierto… **entonces…**

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Para hacer condicionales **necesitamos operadores**

que pueden ser:

- Matemáticos:
  - **+, -, *, /, %.**
- Relacionales:
  - **==, !=, <, >, >=, <=**
- Lógicos
  - **&&, and, ||, or, !**

P	!P
True	False
False	True

P	Q	P&&Q	P||Q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

5. Encender de uno en uno los led pares en rojo y led impares en verde

Podemos usar un condicional simple usando los operadores == (igual) **y !=** (distinto)

```
If (condición){
    sentencias;
}
```

== ➡ ¿Son iguales?
!= ➡ ¿Son distintos?

```
8  Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
9
10
11
12
13 void setup() {
14   // put your setup code here, to run once:
15   pixels.begin();
16
17 }
18
19 void loop() {
20
21   int numLed = 2; //El led 2
22   /*Calcula par o impar. si el resto es 0 es par sino es impar*/
23   int calculo = numLed % 2;
24
25   if(calculo == 0){
26     pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27     }
28
29   if(calculo != 0){
30     pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
31     }
```

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Veamos que hace el código:
  - Creo una **variable numLed** y le asigno el valor **2. (línea 21)**
  - Creo una **variable calculo** a la que le asigno el **resto de dividir numLed entre 2** (para saber si es par o impar) **línea 23.**
  - Si el valor de calculo es igual a 0
    - entonces enciende el led numLed de color rojo.
  - Si el valor de calculo es distinto de 0
    - entonces enciende el led numLed de color verde.

```
8  Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
9
10
11
12
13 void setup() {
14   // put your setup code here, to run once:
15   pixels.begin();
16
17 }
18
19 void loop() {
20
21   int numLed = 2; //El led 2
22  /*Calcula par o impar. si el resto es 0 es par sino es impar*/
23   int calculo = numLed % 2;
24
25   if(calculo == 0){
26     pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27     }
28
29   if(calculo != 0){
30     pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
31     }
```

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Veamos que hace el código:
  - Creo una **variable numLed** y le asigno el valor **2. (línea 21)**
  - Creo una **variable calculo** a la que le asigno el **resto de dividir numLed entre 2** (para saber si es par o impar) **línea 23.**
  - Si el valor de calculo es igual a 0
    - entonces enciende el led numLed de color rojo.
  - Si el valor de calculo es distinto de 0
    - entonces enciende el led numLed de color verde.

```
8  Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
9
10
11
12
13 void setup() {
14   // put your setup code here, to run once:
15   pixels.begin();
16
17 }
18
19 void loop() {
20
21   int numLed = 2; //El led 2
22  /*Calcula par o impar. si el resto es 0 es par sino es impar*/
23   int calculo = numLed % 2;
24
25   if(calculo == 0){
26     pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27     }
28
29   if(calculo != 0){
30     pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
31     }
```

5. Encender de uno en uno los led pares en rojo y led impares en verde

- ¿Y si queremos que haga las comprobaciones de manera automática para todos los leds?

- **¡Usaremos bucles!**

Bucles o ciclos

Bucles

```
Loop(){
  Lo que se va a repetir siempre
}

for (inicio; fin; incremento){
  Lo que se va a repetir desde inicio a fin
}

while(condición){
lo que quiero que haga mientras se cumpla la condición
}
```

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Bucles:
- Loop(){
  Lo que se va a repetir siempre
  }
- for (inicio; fin; incremento){
  Lo que se va a repetir desde inicio a fin
  }
- while(*condición*){
  lo que quiero que haga mientras se cumpla
  la condición
  }

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Bucles:
- **Loop(){**
  Lo que se va a repetir siempre
  }

```
18
19 void loop() {
20
21   int numLed = 2; //El led 2
22  /*Calcula par o impar. si el resto es 0 es par sino es impar*/
23   int calculo = numLed % 2;
24
25   if(calculo == 0){
26     pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27   }else{
28       pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
29         }
```

¡HÁGASE LA LUZ!

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Bucles:
- **for (inicio; fin; incremento){**
  Lo que se va a repetir desde inicio a fin
  }

¡HÁGASE LA LUZ!

5. Encender de uno en uno los led pares en rojo y led impares en verde

- Bucles:
- **while(*condición*){**
  lo que quiero que haga mientras
  se cumpla la condición
  }

6. Aprendemos el condicional simple y el condicional compuesto

Condicional doble

```
If (condición ){
    sentencia1;
}else{
    sentencia2;
}
```

- Este condicional traducido a lenguaje significa :
- si se cumple condicion
  - entonces ejecuta sentencia1
- sino se cumple condición
  - entonces ejecuta sentencia2

6. Aprendemos el condicional simple y el condicional compuesto

- Veamos el ejemplo de par o impar, usando un condicional doble.
- Sería:
  - Si el cálculo es igual a cero,
    - entonces enciende el led2 de color rojo.
  - Sino,
    - entonces enciende el led 2 de color verde.

```
#define NUMPIXELS 3

using namespace std;

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);



void setup() {
  // put your setup code here, to run once:
  pixels.begin();

}

void loop() {

  int numLed = 2; //El led 2
 /*Calcula par o impar. si el resto es 0 es par sino es impar*/
  int calculo = numLed % 2;

  if(calculo == 0){
    pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
  }else{
      pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
        }
```

6. Aprendemos el condicional simple y el condicional compuesto

- Veamos el ejemplo de par o impar, usando un condicional doble.
- Sería:
  - Si el cálculo es igual a cero,
    - entonces enciende el led2 de color rojo.
  - Sino,
    - entonces enciende el led 2 de color verde.

```
#define NUMPIXELS 3

using namespace std;

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);



void setup() {
  // put your setup code here, to run once:
  pixels.begin();

}

void loop() {

  int numLed = 2; //El led 2
 /*Calcula par o impar. si el resto es 0 es par sino es impar*/
  int calculo = numLed % 2;

 if(calculo == 0){
   pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
 }else{
    pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
       }
```

5. Encender de uno en uno los led pares en rojo y led impares en verde

Ahora que ya sabes encender los leds, edita el código para que, se encienda de uno en uno los led pero que se apague el anterior.

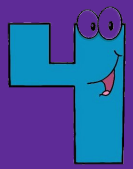
Te dejo el algoritmo. Piensa como se programa aplicando lo aprendido antes.


```
Para i=0 mientras i sea menor que el número de pixel incrementa i en 1
        función que pone el color al pixel i
        si i es distinto de 0
                función que pone el color al pixel i anterior y poner el color 0 que es apagado
        aplicar la configuración a los pixel
        esperar medio segundo
```

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

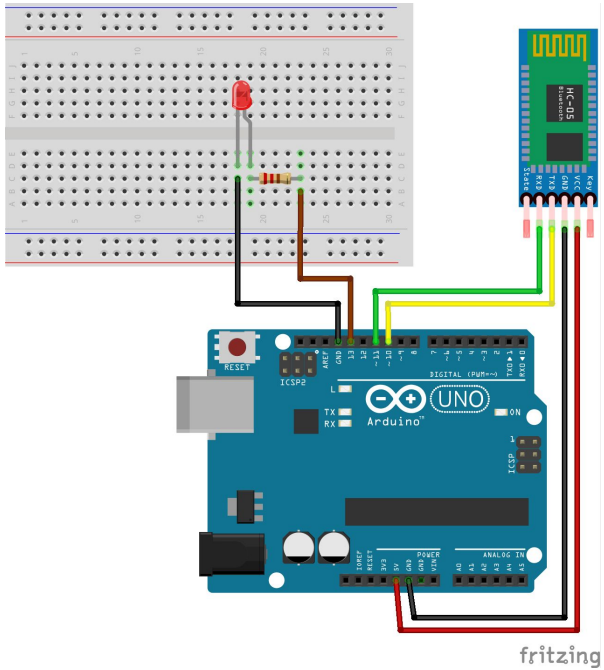
Montaje del SOMBRERO



¡Bluetooth!

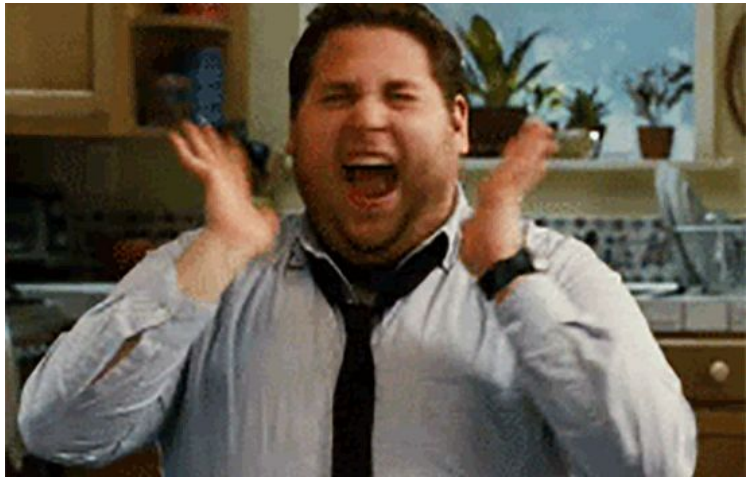
Conexiones

- **RXD** con 11
- **TXD** con 10
- **GND** con GND
- **VCC** con 5V



¿CÓMO LO VAMOS A HACER?

- Vamos a hacerlo con un módulo bluetooth y nuestro arduino uno!
- Además, vamos a usar una aplicación en nuestro móvil para conectarnos!

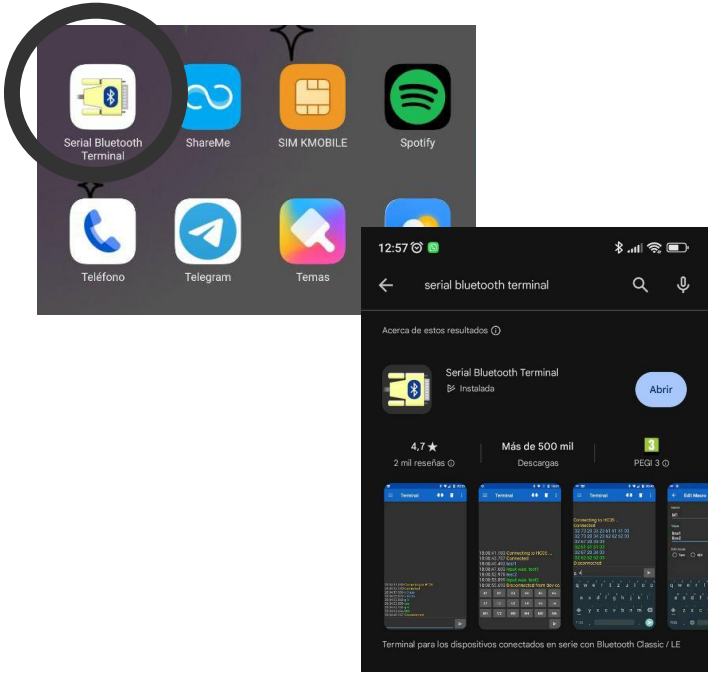


Oye...¿pero qué aplicación?

- Yo recomiendo:

  Serial Bluetooth Terminal.

AVISO: Por favor, solo funciona el módulo bluetooth con dispositivos **ANDROID**



¡FUNCIONES QUE VAMOS A USAR!

begin() → Inicializar

```
bluetoothSerial.begin(9600);
```

available() → Ver si está conectado

```
bluetoothSerial.available()
```

read() → leer lo que le pasemos por el móvil

```
bluetoothSerial.read();
```

¿Cómo lo haremos? En el void setup

Lo que vamos a hacer aquí es inicializar el serial, a 115200 baudios. Además, inicializamos el bluetooth a 9600 baudios.

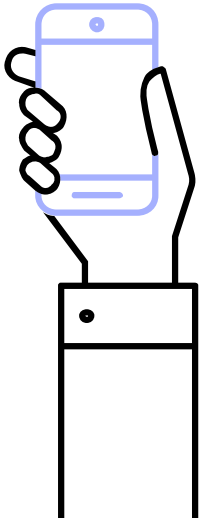
Hay que tener en cuenta de que en la terminal(la lupita) del arduino tenemos que poner los baudios a ese mismo valor para que todo vaya bien

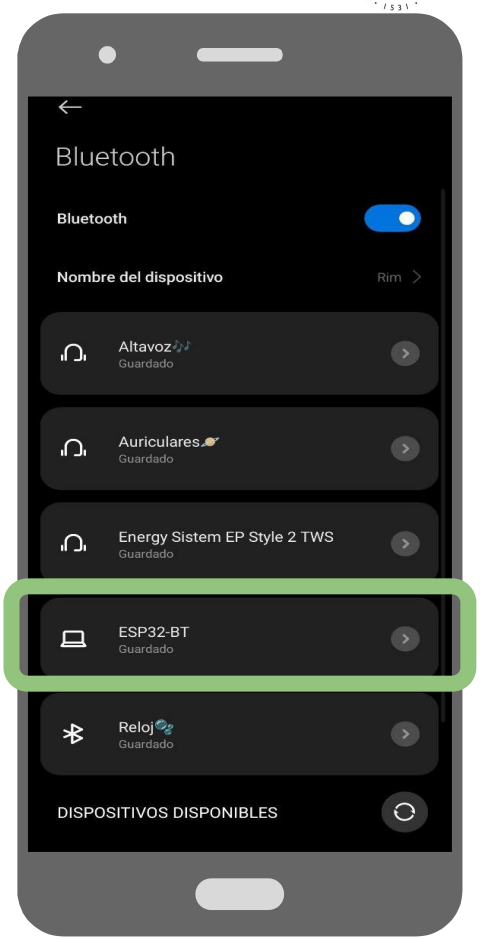
¿Cómo lo haremos? En el void loop

Primero vamos a mirar a ver si nuestro bluetooth está disponible.
En el caso de que esté:

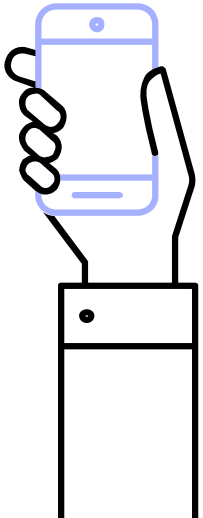
- hacemos lo que nosotras qeuramos dentro de ella, por ejemplo, podemos mandar unas letras o mensajes e imprimirlo por la terminal del arduino.

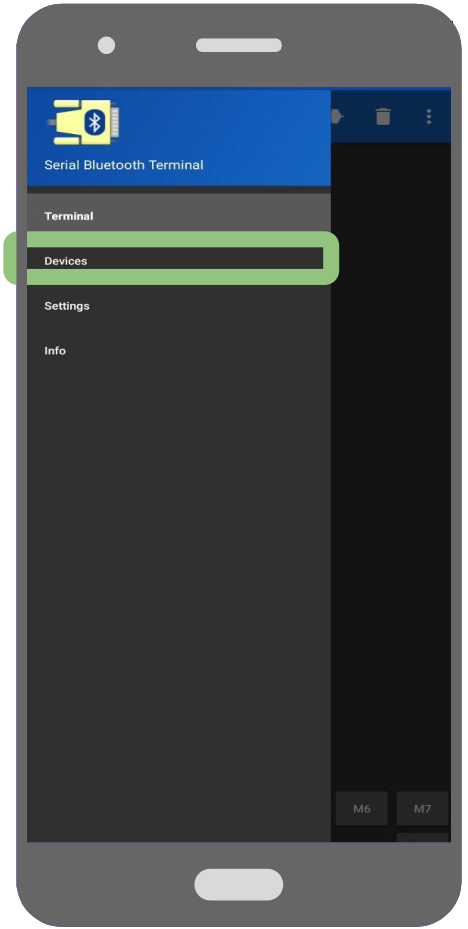
¿CÓMO ME CONECTO AL MOVIL?



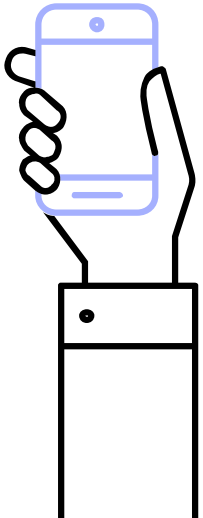


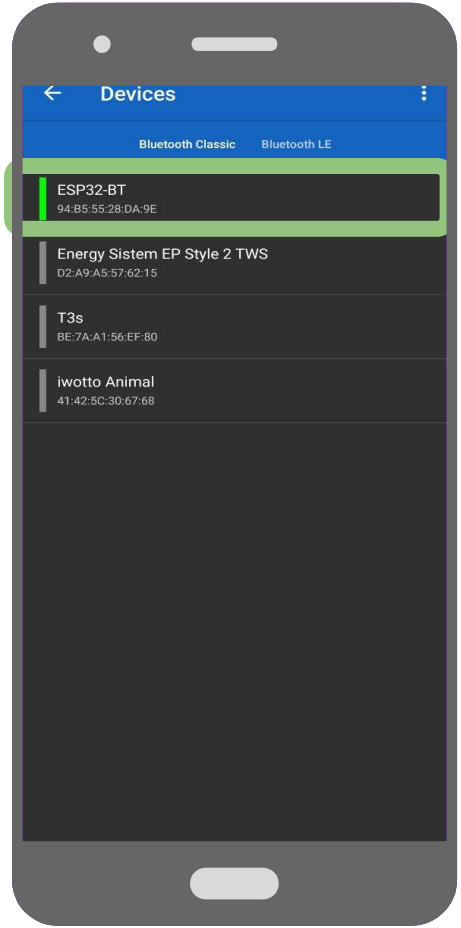
¿CÓMO ME CONECTO AL MOVIL?



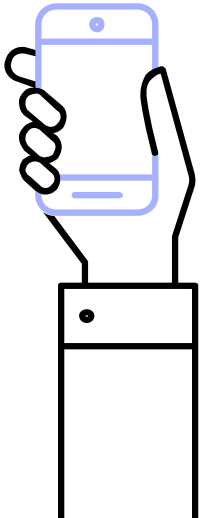


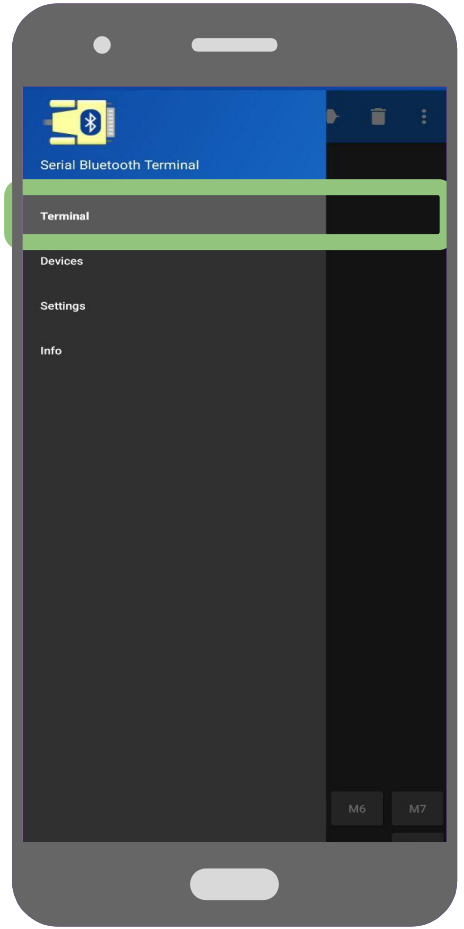
¿CÓMO ME CONECTO AL MOVIL?



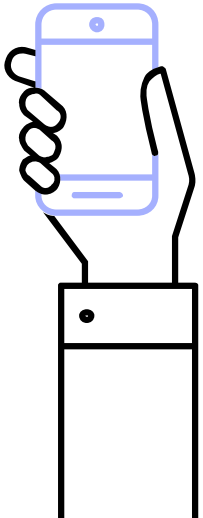


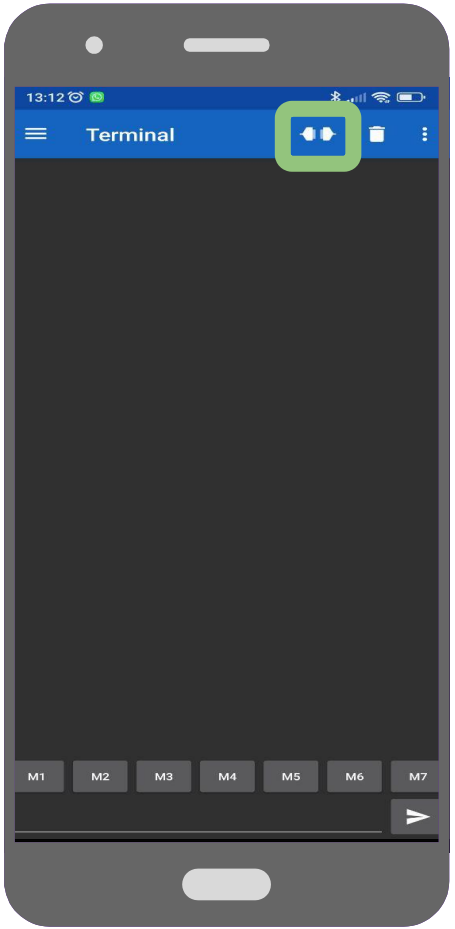
¿CÓMO ME CONECTO AL MOVIL?



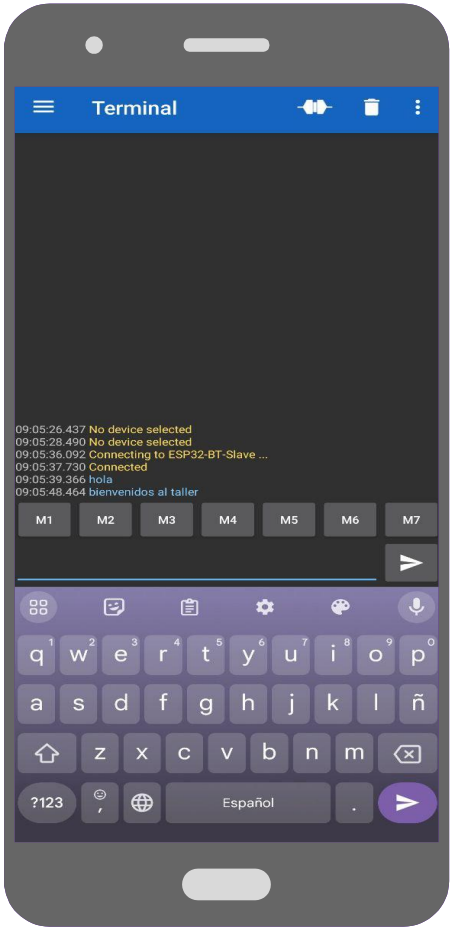


¿CÓMO ME CONECTO AL MOVIL?





¿CÓMO ME CONECTO AL MOVIL?



¿Como hago mi primer programa?

Una vez que tenemos la estructura que está explicada diapositivas arriba, primero vamos a intentar que se mande mensajes del móvil al ordenador. Para ello, voy a dejar una pequeña guía(*un pseudocódigo*).

¿Como hago mi primer programa?

```
void setup(){
        inicialización de Serial y del
        módulo bluetooth
}
void loop(){
si el bluetooth está disponible{
                tenemos una variable
                caracter, que es el resultado
                de lo que se lee por bluetooth.
                Eso se imprime por pantalla
        }
}
```

Oye…¿pero como lo inicializo?

Para iniciarlo, lo hacemos con la siguiente línea del código:

SoftwareSerial <nombreBluetooth>(10,11);

donde <nombreBluetooth> es el nombre que le pongáis para identificar en el código el módulo.

```
#include <SoftwareSerial.h>

SoftwareSerial bluetoothSerial(10, 11);
```

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Montaje del SOMBRERO

¡Programación!

Recordemos...

Condicionales if e if else

```
If (condición ){
    sentencias;
}else if(condición){
    sentencias;
}else{

    sentencias;
}
```

Atención!!! Estructura nueva!

switch

```
switch(variable ){
    case variable1:
        sentencias;
        break;
    ...
    case variableN:
        sentencias;
        break;

}
```

ATENCIÓN!!! ESTRUCTURA NUEVA!

Ejemplo switch

Nota: data_received es una variable tipo **char**.

char = carácter, una letra, número o signo de nuestro teclado!

```
switch(data_received){
  case '0':
    display.clearDisplay();
    display.display();
    break;
```

¿Qué vamos a hacer con esta estructura?

Así vamos a crear distintas acciones, dependiendo de qué carácter le mandemos a arduino…

Por ejemplo:

- ★ Si envío 1, que se enciendan los leds en rojo
- ★ Si envío a, que se mueva el servo 1.

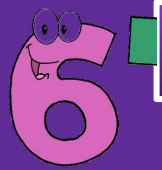
¿Qué vamos a hacer con esta estructura?

De ahí, con los componentes que hemos aprendido a programar antes, tenéis rienda suelta a vuestra imaginación para hacer las **funciones que queráis**!!

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Montaje del SOMBRERO



¡Decoración!

¿CÓMO VAMOS A HACER LA ESTRUCTURA?

Mi sombrero está hecho a base de **cartón**, **tela**, **goma eva** y silicona caliente. Pero vostras podeís utilizar los componentes que queráis!

¿Cómo vamos a hacer la estructura?

Aquí dejo algunas referencias por si queréis hacer un sombrero.
Pero… sois libres de hacer la parte física/decoración como os guste más!