

# ARDUINO AVANZADO I

Pensamiento Visual - IDI1015

Alonso Canales - [aecanales@uc.cl](mailto:aecanales@uc.cl)



ESCUELA DE INGENIERÍA  
FACULTAD DE INGENIERÍA





## ¿Qué conoceremos hoy?

- Más herramientas de programación
- Sensores y actuadores varios
- Shields
- Raspberry Pi

```
        tmpFormat = 14 #Replace string by value > q
        str(key)) tempString = tempString.replace("czDataTy
        ,str(value*pow(10,14-tmpFormat)))) tempString = temp
        string + tempString + tempString.replace("czFieldID",str(key)) tempStri
        string + tempString.replace("ASCIIT STRING"))
```

## Herramientas de programación

## MÁS TIPOS DE VARIABLES

- » **Booleano:** valores verdaderos o falso.

```
bool running = false;
```

- » **Float:** número decimal.

```
float sensorCalibrate = 1.117;
```

- » **String:** una secuencia de caracteres.

```
String stringOne = "Hello String";
```

## ¿CÓMO ESCRIBIR MÁS ORDENADO?

- » Mediante **comentarios** y **funciones**.

## ¿CÓMO ESCRIBIR MÁS ORDENADO?

- » Los **comentarios** nos permiten escribir explicaciones de lo que hace nuestro código.

```
// Esto es un comentario de una línea.
```

```
/*
```

```
 Esto es un comentario de  
múltiples líneas.
```

```
 */
```

```
int funcionSuma(int x, int y) {  
    int resultado;  
  
    resultado = x + y;  
    return resultado;  
}
```

## Tipo de retorno de la función

```
int funcionSuma(int x, int y) {  
    int resultado;  
  
    resultado = x + y;  
    return resultado;  
}
```

Nombre de la función



```
int funcionSuma(int x, int y) {  
    int resultado;  
  
    resultado = x + y;  
    return resultado;  
}
```

## Parámetros de entrada

```
int funcionSuma(int x, int y) {  
    int resultado;  
  
    resultado = x + y;  
    return resultado;  
}
```

## Retorno de la función

```
int funcionSuma(int x, int y) {  
    int resultado;  
  
    resultado = x + y;  
    return resultado;  
}
```

```
void loop() {  
  
int suma = funcionSuma(2, 3);  
  
}
```

El tipo `void` sirve para funciones que no retornan

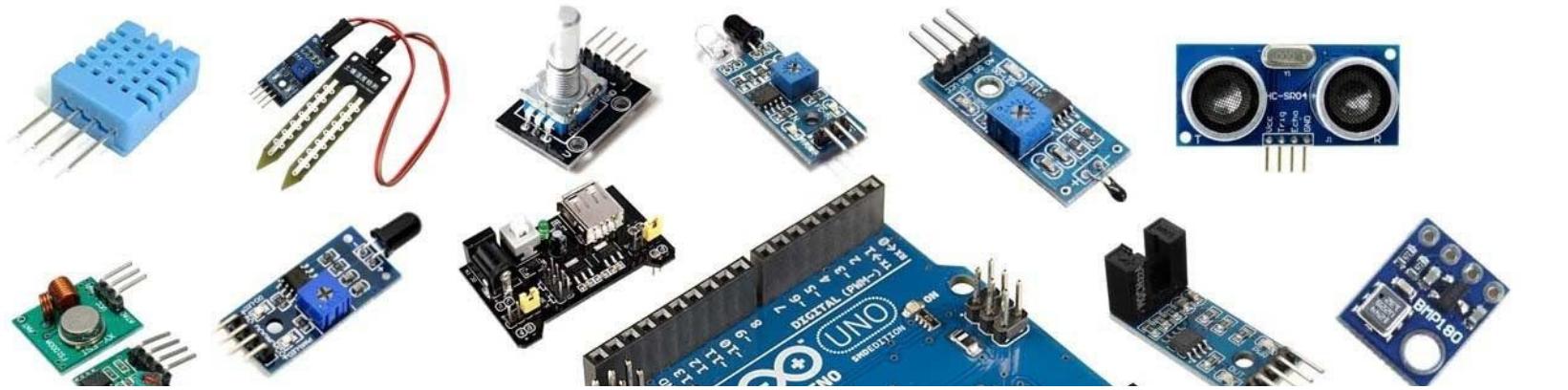
```
void girarServo(int angulo) {  
    myservo.write(angulo);  
    delay(1000);  
    myservo.write(0);  
}
```

## LIBRERÍAS

- » Funcionalidad encapsulada que nos facilita la programación de componentes.
- » Ya usaron una: la libreria Servo.
- » Se instalan mediante ZIP o Library Manager y luego se incluyen en el código mediante el comando  
`#include <Libreria.h>`

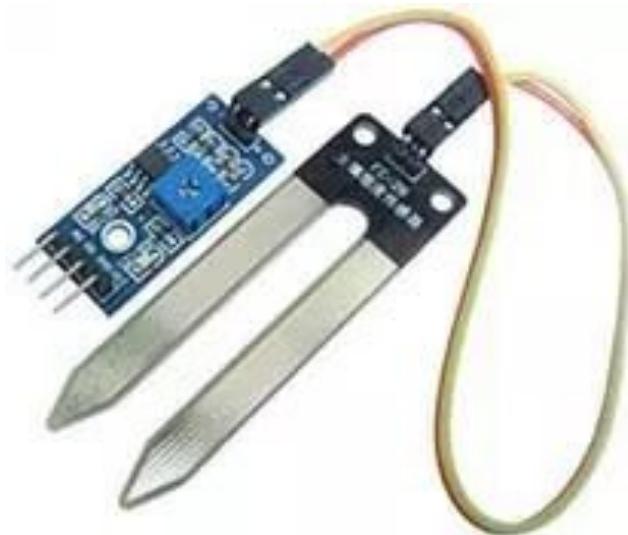
## ALGUNAS LIBRERÍAS

- » EEPROM: Para escribir a la memoria de la placa.
- » Ethernet: Para conectarse a Internet.
- » LiquidCrystal: Para controlar pantallas LCD.
- » SD: Permite leer y escribir a una tarjeta SD.
- » Y muchos más...



## Sensores y actuadores varios

## SENSOR DE HUMEDAD DE TIERRA



- » Detecta la humedad de la tierra en la cual se coloca.

# SENSOR DE SONIDO

*¡Disponible en el stock!*

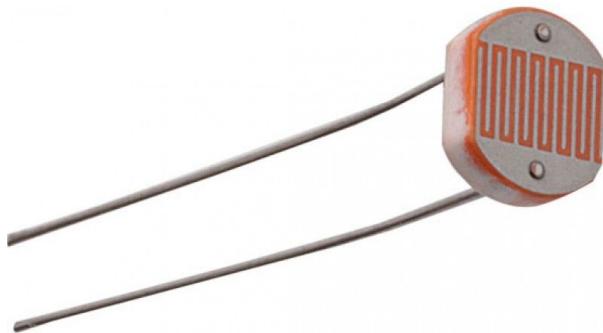


» Detecta sonido.

# FOTORESISTENCIA (LDR)

*¡Disponible en el stock!*

- » Detecta la cantidad de luz en el ambiente.

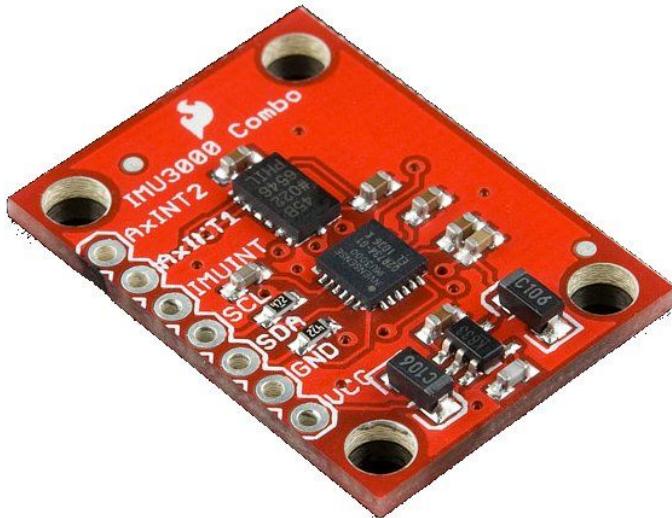


# SENSOR DE VIBRACIÓN



- » Detecta cuando se mueve el módulo.

# INERTIAL MEASUREMENT UNIT



- » Contiene accelerómetro, giroscopio y magnetómetro todo en uno.

# JOYSTICK



» El clásico de los controles de videojuegos.

# RELAY O RELÉ

*¡Disponible en el stock!*



- » Nos permite controlar artefactos eléctricos mediante un Arduino.

# DIODO LÁSER



» El clásico laser, ahora controlable por Arduino.

# NEOPIXEL



- » Anillo de LEDs RGB fáciles de controlar.

# SENSOR RFID



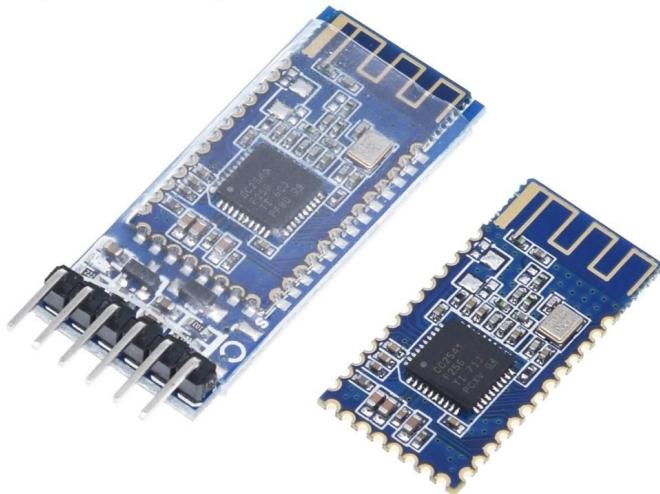
- » Identificación de anillos o tarjetas.

# PANTALLA LCD



- » Clásica forma de mostrar información
- » Vienen en distintos tamaños
- » ¡Buscar uno con adaptador I2C!

# MÓDULO BLUETOOTH

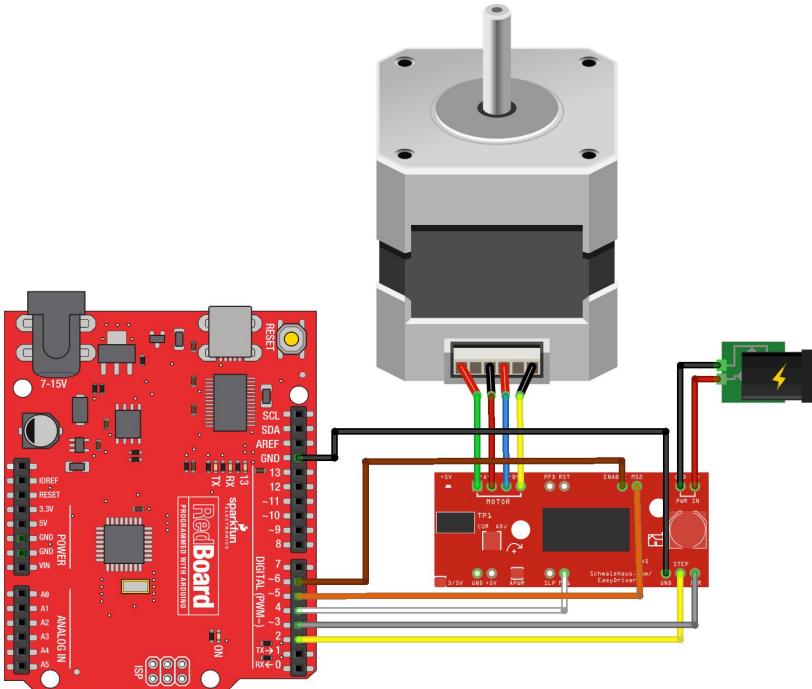


- » Permite control inalámbrico de un Arduino
- » Puede ser mediante un celular o hasta mediante un computador

# MOTOR STEPPER



- » Motor preciso para movimientos que requiere mayor torque
- » Común en las impresoras 3D
- » Dado su complejidad, necesita alimentación y un *driver* aparte



fritzing



# Shields

¿Cómo agregar funcionalidad de manera eficiente, ordenada y simple?



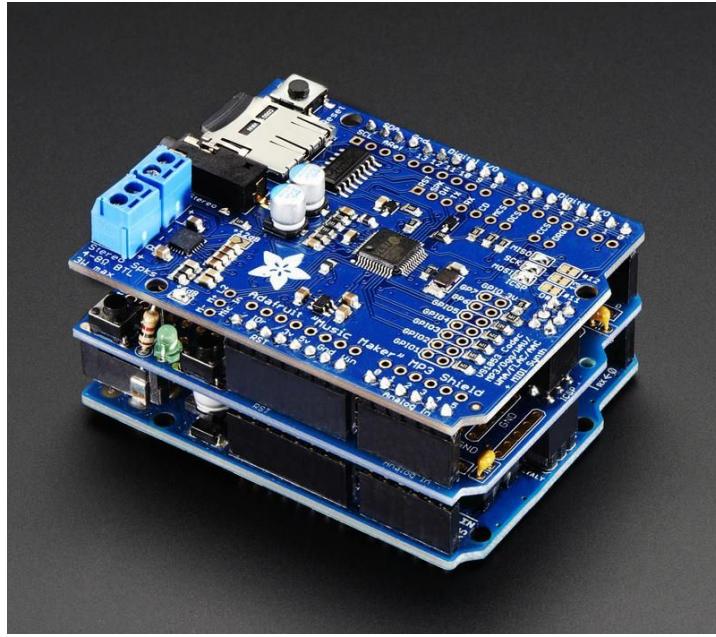
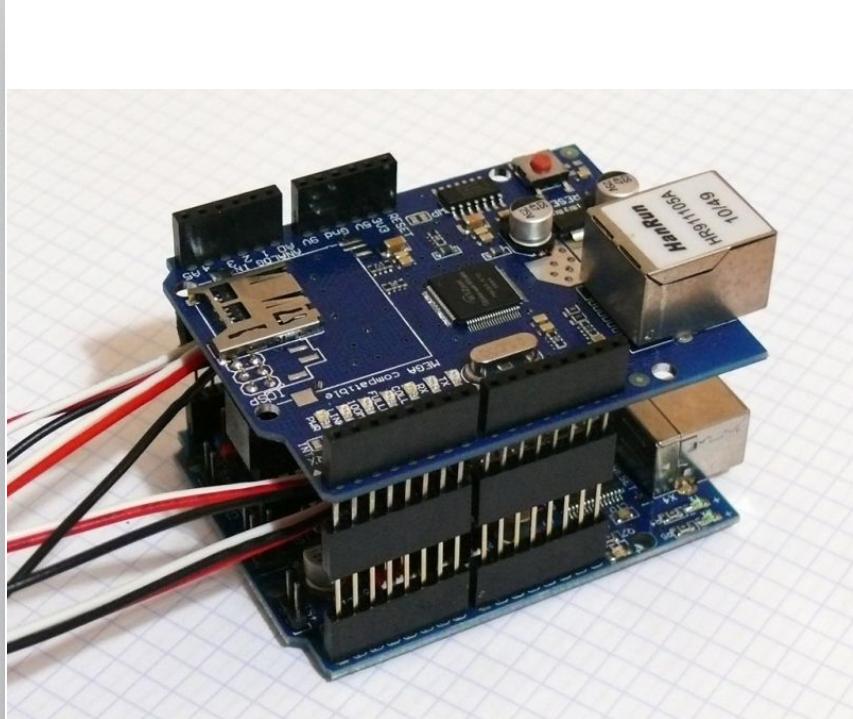


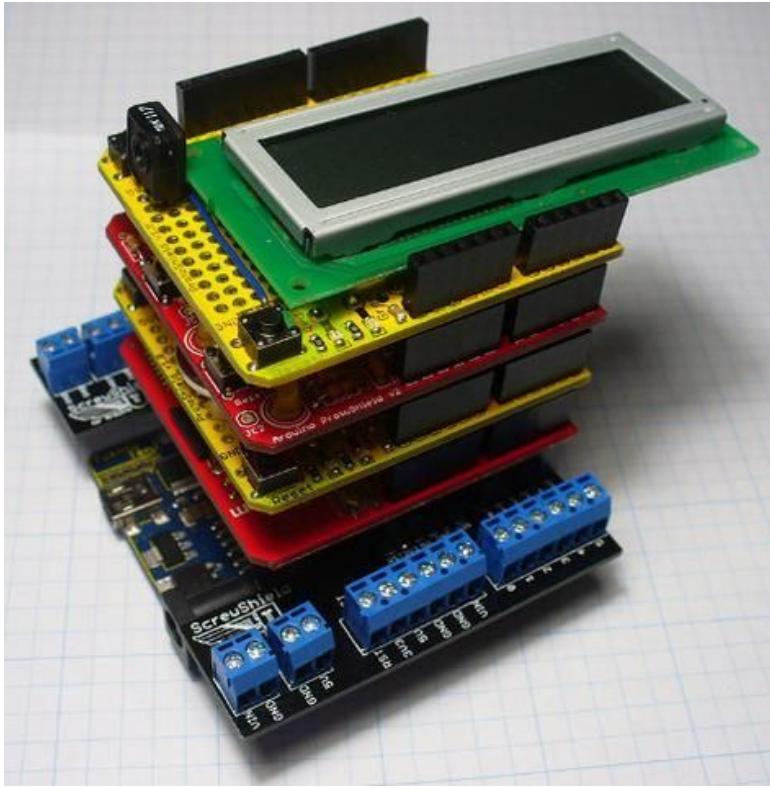


## SOBRE SHIELDS



- » Provee la misma funcionalidad de manera más rápida y fácil de instalar.
- » Ojo que sí ocupan ciertos pines, por lo que deben revisar la documentación o en [shieldlist.org](http://shieldlist.org).



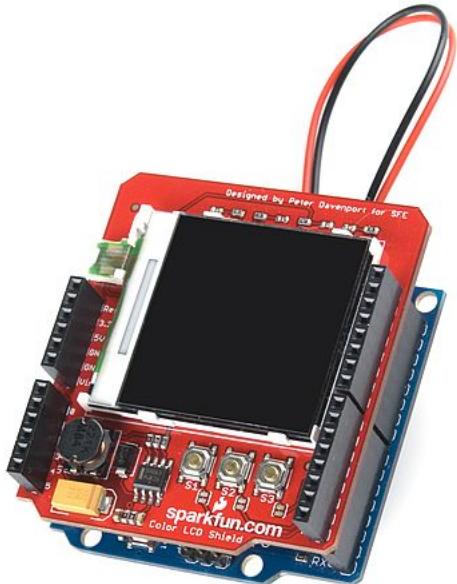


# JOYSTICK SHIELD



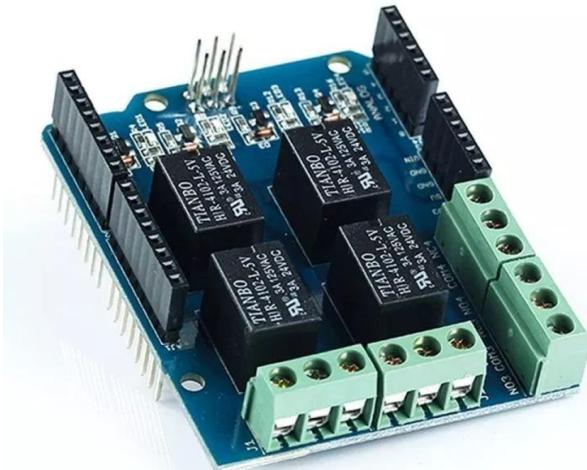
- » Haz un control tradicional de juego de manera rápida y fácil.

# COLOR LCD SHIELD



- » Agrega una pantalla de 128x128 pixeles a tu Arduino.

# RELAY SHIELD



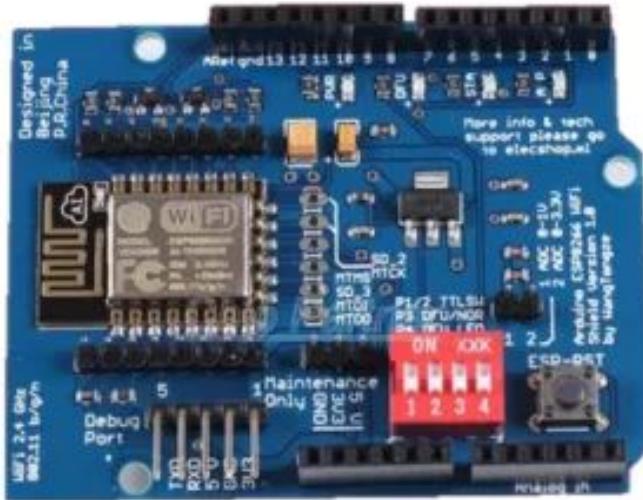
- » Incluye 4 relés y dónde conectar los cables.

# PROTOSHIELD



- » Una protoboard integrada para llevar a todas partes.

# WIFI SHIELD



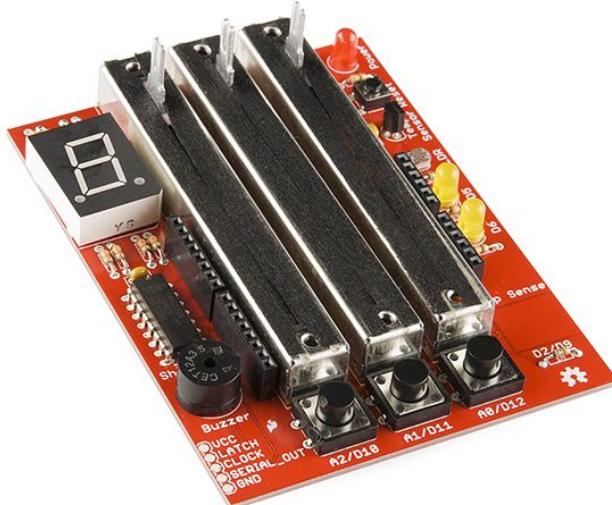
- » Conecta tu Arduino a Internet mediante WiFi.

# BLUETOOTH SHIELD

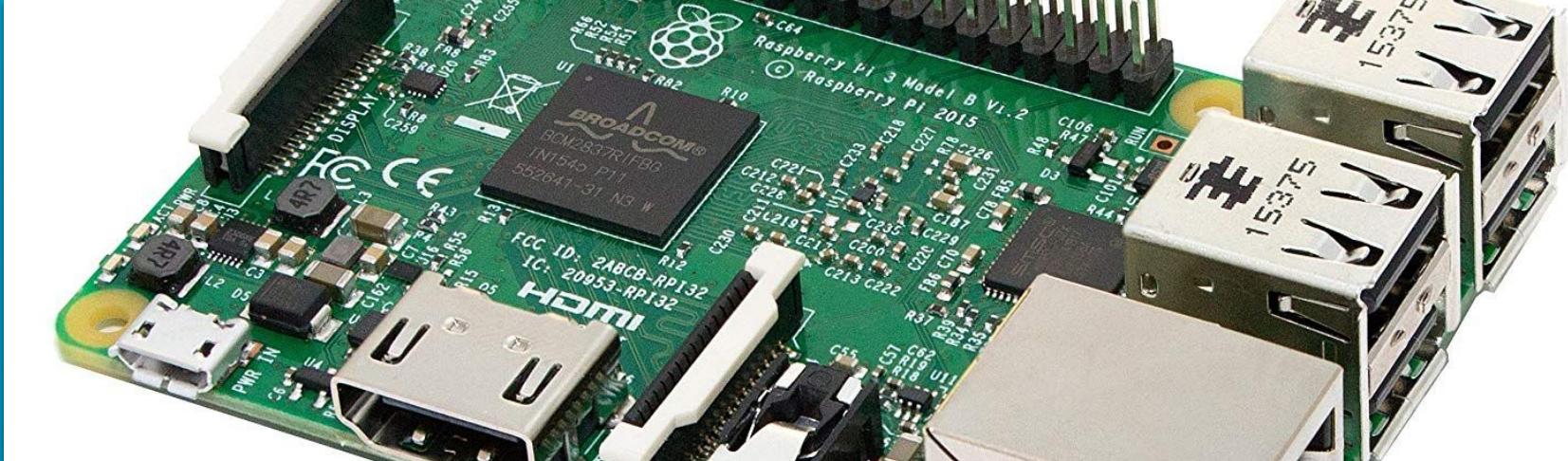


» Conecta tu Arduino a dispositivos Bluetooth.

# “DANGER” SHIELD



» Sensores varios.



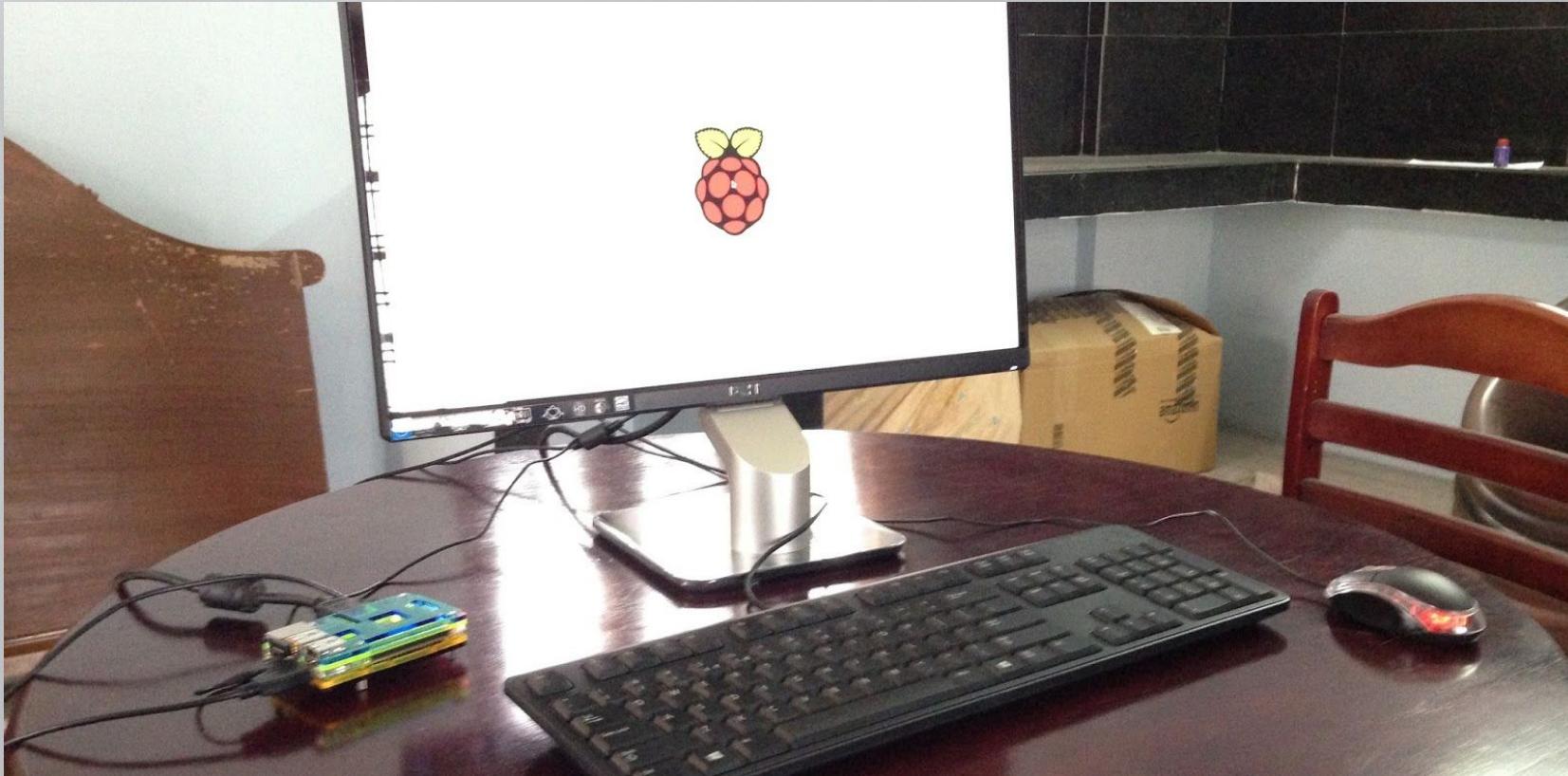
# Raspberry Pi

“

*¿Es lo mismo que un Arduino?*



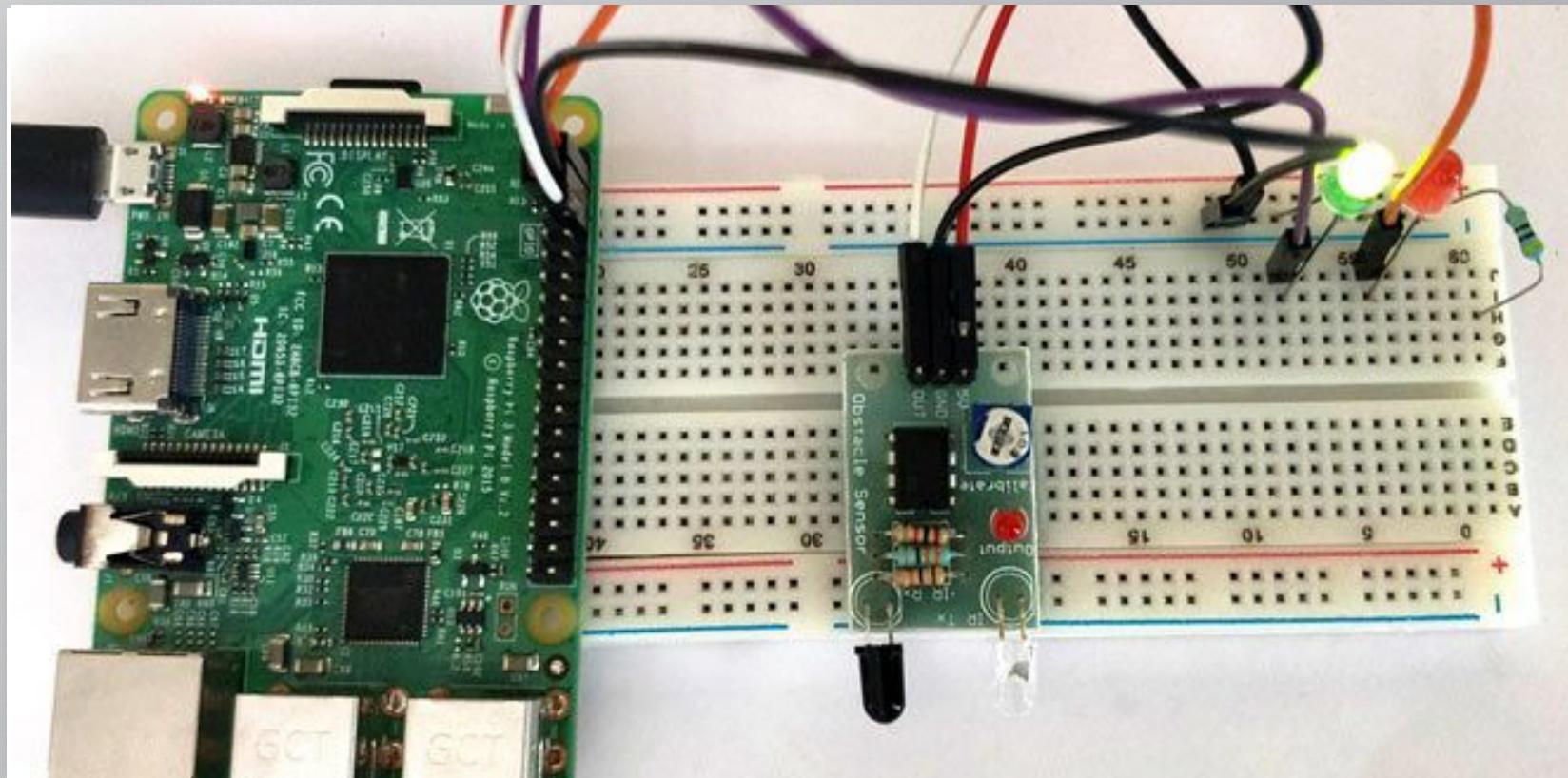
El Raspberry Pi es un mini-computador.







```
>>> for x in range(10): setblock(x, 0, 0, DIRT)
...>>>
```



“

*Podemos llegar a replicar mucho de  
lo que un Arduino puede hacer  
usando un Raspberry Pi.*

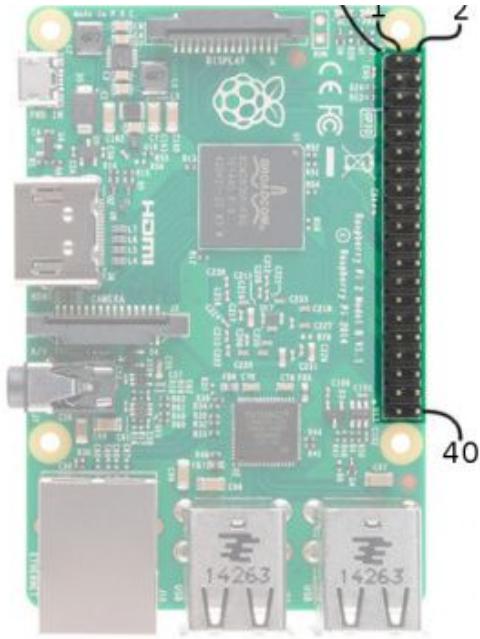
“

*Podemos llegar a replicar mucho de  
lo que un Arduino puede hacer  
usando un Raspberry Pi.*

*¿...para qué usamos Arduino entonces?*

## RASPBERRY PI VS ARDUINO

- » Una placa Arduino es mucho más barata.
- » Es mucho más fácil usar un Arduino que una Raspberry Pi.

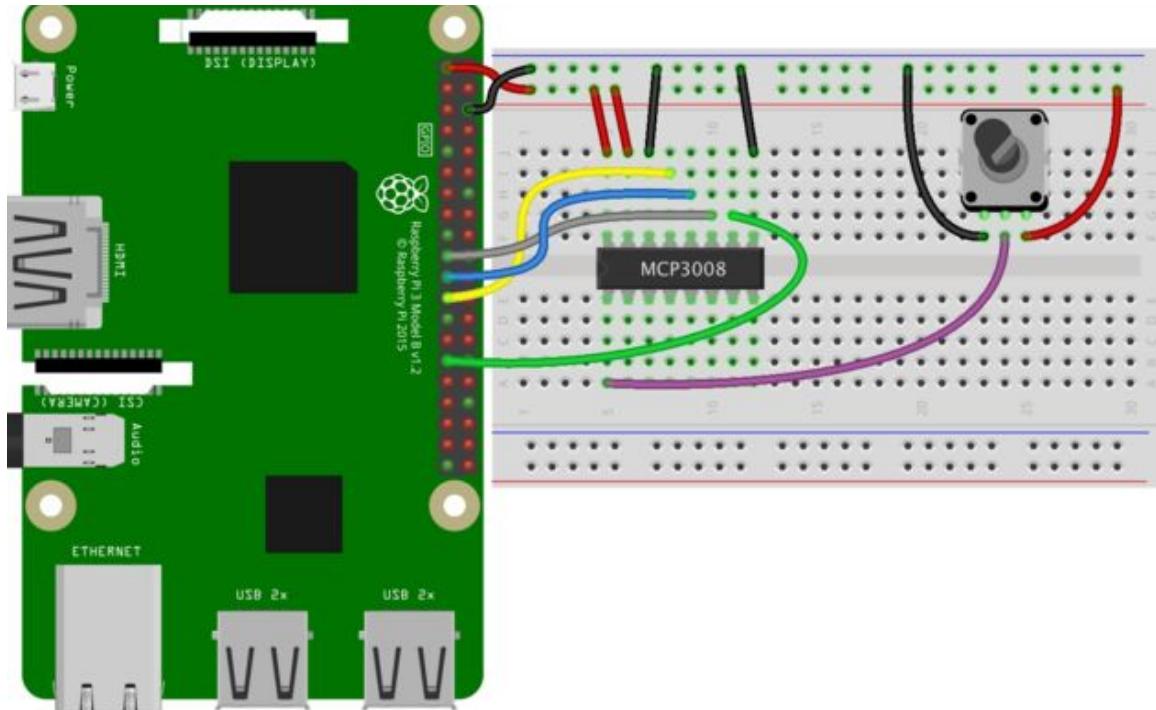


Raspberry Pi2 GPIO Header			
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>C</sup> )	DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>C</sup> )	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>C</sup> ID EEPROM)	(I <sup>C</sup> ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev 1  
26/01/2014

<http://www.element14.com>

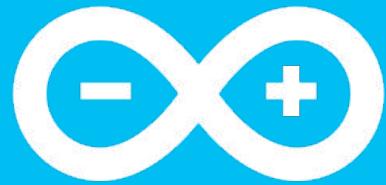




fritzing

## ¿CUANDO USAR UNA RASPBERRY PI?

- » Proyecto puesto de forma permanente en el cual quieren ahorrarse el costo de un computador.
- » Proyecto que donde no basta el procesador del Arduino para realizar las operaciones y es difícil conectarse a un computador.



# ARDUINO AVANZADO I

Pensamiento Visual - IDI1015

Alonso Canales

aecanales@uc.cl