



TU MUNDO MAKER

Arduino es tu boleto de entrada al mundo de la electrónica, la robótica y la automatización. Las placas Arduino son pequeñas computadoras que puedes programar de forma muy fácil para interactuar con el mundo que te rodea. Todo esto es posible gracias a que Arduino cuenta con una gigantesca comunidad dispuesta a compartir su experiencia y conocimiento contigo, esta comunidad son los llamados “maker”. ¿Quiénes son los Maker? Somos personas con el deseo y la pasión por crear: pensamos, diseñamos y hacemos realidad nuestros proyectos. Puedes hacer cosas increíbles con Arduino y la comunidad Maker, no hay límite para lo que puedes hacer. ¡Usando tu imaginación todo es posible!



Síguenos y forma parte de esta Comunidad MAKER que cada día crece con más personas alrededor del mundo... "Todos podemos ser un Maker".



Síguenos en nuestras redes: @tumundomaker

TABLA DE CONTENIDO

TU MUNDO MAKER	2
TABLA DE CONTENIDO	2
MIS PRIMEROS PASOS CON ARDUINO	3
¿Qué es Arduino?	3
Tipos de Arduino	3
¿Dónde comprarlo?	3
Conociendo la placa Arduino UNO	4
Cosas que puedes hacer con Arduino	4
EL ENTORNO DE DESARROLLO	5
El entorno de desarrollo (IDE) para Arduino	5
Conectando el Arduino a la PC	6
Conceptos básicos de programación en Arduino	6
METODOLOGIA MAKER PARA TUS PROYECTOS	7
“DESIGN THINKING”	7
PROYECTOS BÁSICOS CON ARDUINO	8
Parpadeo de un LED: el ‘Hola Mundo’ de Arduino	8
Semáforo	10
Dimerizando un LED (PWM)	10
Detector de Presencia	11
Alarma anti robo	11
Medidor de distancia	12
Alarma anti choque	12
Luz nocturna automática (con LED)	13

¡Todos podemos ser un Maker!

MIS PRIMEROS PASOS CON ARDUINO

¿Qué es Arduino?

Las placas Arduino son pequeñas computadoras que puedes programar de forma fácil para interactuar con el mundo que te rodea. Son tu boleto de entrada al mundo de la electrónica, la robótica y la automatización. Las placas Arduino fueron desarrolladas para democratizar la electrónica, cuando antes tenías que estudiar durante más de cinco años una carrera universitaria para poder configurar y programar un micro controlador, ahora, en cuestión de minutos, puedes hacer realidad tus propios proyectos. Las placas Arduino tienen una forma similar a la que ves en la siguiente imagen:

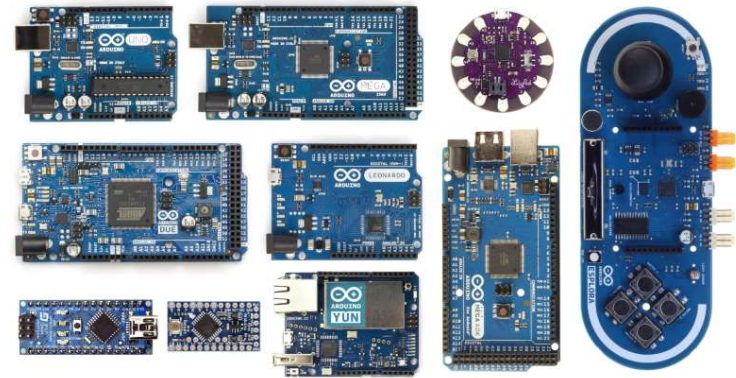


Básicamente, es una pequeña computadora integrada en una placa prefabricada a la que puedes conectar otros circuitos electrónicos, esto hace posible el intercambio de datos con el mundo exterior. El cerebro de esta placa (Arduino Uno) es un micro controlador, al cual podrás dar instrucciones que le dirán a tu Arduino qué hacer y cómo hacerlo.

La mejor parte es que Arduino es una tecnología abierta, lo que significa que no tienes que adquirir ninguna licencia para utilizarlo, esto lo convierte en una de las herramientas favoritas de los makers, y gracias a la inmensa cantidad de personas que esta comunidad tiene, podrás encontrar en internet toda una enciclopedia, tutoriales y manuales que te facilitarán el tener éxito en tus proyectos de electrónica. Además, seguramente encontrarás en tu barrio personas que conozcan y trabajen con Arduino, así podrán generar juntos nuevas ideas, nuevos proyectos y hacer crecer aún más esta comunidad.

Tipos de Arduino

Una de las cosas que puede generarte confusión al momento de entrar en el mundo de Arduino es la gran cantidad de placas y modelos que existen en el mercado, y es que hasta la fecha se han diseñado más de 60 placas reconocidas oficialmente por la comunidad Arduino. Pero que este tema no te quite el sueño. Siendo Arduino una de las primeras plataformas con micro controladores de fuente abierta que se crearon, era lógico esperarse que se desarrollaran varias versiones de ésta. En la siguiente imagen podrás ver solo algunos de los modelos existentes.



¿Cuál es la solución? Muchas personas dan sus primeros pasos con el Arduino UNO, pero a medida que se avanza en el dominio de la programación y la electrónica, los requerimientos de procesamiento y número de pines de entrada y salida, aumentan de manera que el UNO ya no es suficiente para nuestros proyectos, y cuando esto ocurra será el momento de dejar tu Arduino UNO atrás para seguir avanzando en el nivel de tus proyectos.

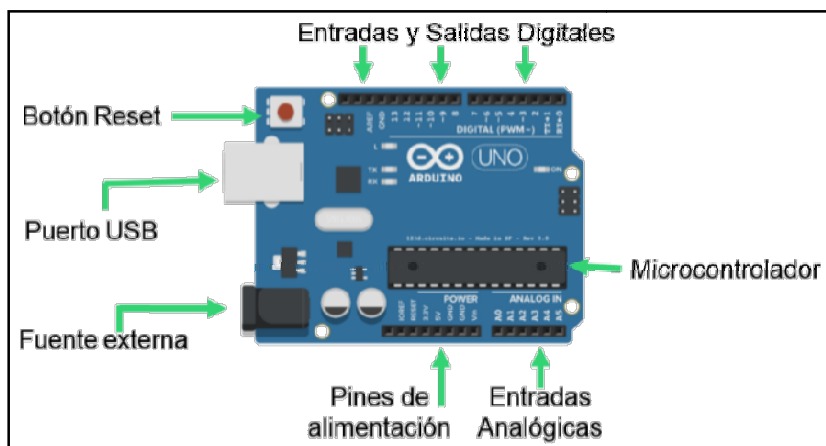
¿Dónde comprarlo?

Puedes adquirir el modelo original de Arduino desde la página web de los creadores: arduino.cc; sin embargo, por tratarse de tecnologías abiertas, puedes encontrar placas similares por un costo más bajo en casi cualquier tienda de electrónica, por internet y también con nosotros, en **Tu Mundo Maker** te ofrecemos amablemente toda la asesoría que necesitas para seleccionar el producto que más se adapte a tus necesidades. Siéntete libre de contactarnos cuando quieras. tumundomaker@gmail.com / (+57) 312 7897151



Conociendo la placa Arduino UNO

En la figura siguiente, puede ver una placa Arduino UNO etiquetada. Veamos qué hace cada parte.



Micro controlador: el micro controlador es el elemento principal de la placa Arduino, todos los componentes que contiene la placa están diseñados para brindar soporte a este micro controlador. Digamos entonces que el micro controlador es el cerebro del Arduino.

Entradas y salidas digitales: las placas Arduino UNO tienen 14 pines que sirven de entradas o salidas digitales, esto significa que estas podrán posicionarse en solo dos valores: alto (HIGH) y bajo (LOW), lo que representaría el 1 y el 0 en lenguaje binario.

Entradas analógicas: el Arduino UNO cuenta con 6 pines que sirven como entrada de señales analógicas, están etiquetados desde A0 hasta A5. Pueden leer diferentes cantidades de voltaje entre 0 y 5V. Además, si se requiere, estos también pueden reconfigurarse como pines de entrada y salida digital, sumándose a los 14 que tiene por defecto, para dar un total de 20 entradas y salidas digitales.

Pines de alimentación: internamente, la placa Arduino UNO cuenta con varios reguladores y estabilizadores de voltaje, ofreciendo salidas estables de 3.3V y de 5V, lo que es de gran utilidad ya que una gran cantidad de componentes electrónicos funcionan con estos dos voltajes.

Botón de reinicio (reset): al presionar este botón, el programa que está ejecutándose empezará de nuevo desde la primera línea de comandos, esto también se puede lograr aplicando un pequeño voltaje en el pin etiquetado como Reset, ubicado al lado de los pines de alimentación.

Puerto USB: aquí se conectará el Arduino con la PC para configurarlo y programarlo, también servirá como fuente de poder de 5V, aunque su corriente es muy limitada.

Fuente externa: aquí conectaremos una fuente de voltaje adicional cuando el consumo eléctrico de nuestro proyecto sea superior al suministrado por el cable USB, o bien, cuando no queramos utilizar el cable USB como fuente de poder.

Hay otras partes en la placa Arduino que no necesita saber por ahora. Sin embargo, si desea obtener más información al respecto, puede consultar la página web oficial de Arduino.

Cosas que puedes hacer con Arduino

Con Arduino puede enviar información al mundo exterior para realizar tareas, o puede leer información para saber qué está pasando. Lo más simple que puedes controlar con tu Arduino es una luz LED. Puedes activarlo, desactivarlo, puedes decidir cuánto tiempo está apagado y cuánto tiempo está encendido, o hasta cuántas veces parpadeará por segundo. También puedes configurarlo para que cuando se produce una situación específica (alarma), el LED se encienda o apague. A continuación veremos una lista de cosas que puedes controlar en tus primeros pasos con Arduino UNO y el Kit Starter.

Matriz de led, o pantalla LCD

Puedes conectar una pantalla para mostrar texto o figuras con tu Arduino, la pantalla más común es la LCD de 2 filas y 16 columnas, lo que significa que puedes mostrar hasta 32 caracteres en la pantalla desde tu Arduino.



Síguenos en nuestras redes: @tumundomaker

¡Todos podemos ser un Maker!

Sensor de movimiento o presencia



El sensor PIR es por mucho el sensor más utilizado como detector de movimiento, puedes empezar tus primeros proyectos de Domótica utilizando este sensor para encender una luz cuando detecte movimiento y apagarla cuando no lo detecte.

Sensor ultrasónico



Con el sensor ultrasónico puedes medir distancia de manera muy precisa y sin hacer uso de la cinta métrica, solo conectamos el sensor ultrasónico al Arduino y tomamos las mediciones de forma digital y mucho más divertida.

Sensor de humedad y temperatura



Puedes también determinar los niveles de humedad o temperatura que hay a tu alrededor, incluso puedes empezar a armar tu primera estación meteorológica desde casa usando tu Arduino UNO

Sensor de luz



Siguiendo con la iluminación, también puedes utilizar una foto celda para que cuando los niveles de iluminación sean muy bajos (cuando esté oscuro), las luces se enciendan automáticamente.

. Motores de corriente continúa



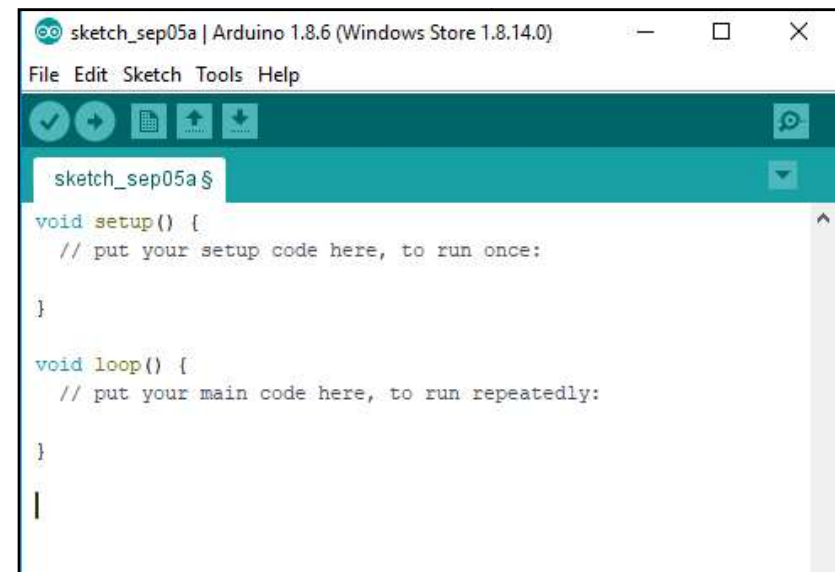
Puedes conectar pequeños motores de corriente continua para crear proyectos con movimiento, carros a control remoto, poleas automáticas, sensores de equilibrio, y otros.

EL ENTORNO DE DESARROLLO

El entorno de desarrollo (IDE) para Arduino

Arduino IDE (Integrated Development Environment) es el entorno donde desarrollarás toda la programación que le dirá al Arduino qué hacer y cuando hacerlo. Puedes hacer esto a través del puerto USB del computador y usando el Arduino IDE.

No vamos a entrar en detalles sobre cómo instalar este software, ya que el sitio web oficial de Arduino hace un gran trabajo explicando cómo hacerlo en los tres sistemas operativos: Windows, Mac y Linux. Una vez instalado, cuando abras por primera vez el IDE de Arduino verás algo similar al siguiente cuadro, aquí es donde se crea la magia: *La programación*.



Para descargar de manera gratuita el IDE de Arduino, basta con seguir el siguiente enlace:

<https://www.arduino.cc/en/Main/Software>



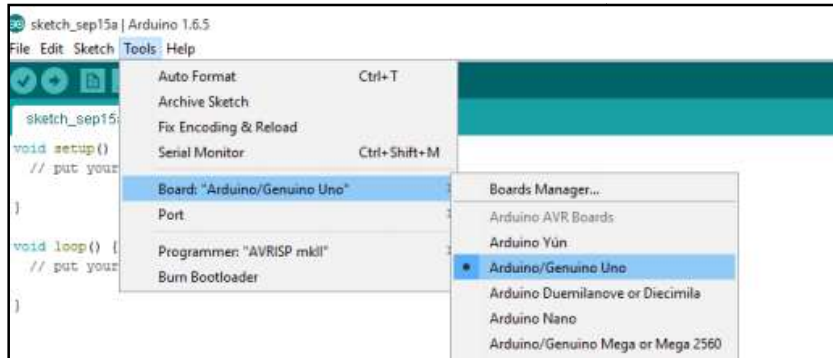
Síguenos en nuestras redes: @tumundomaker

¡Todos podemos ser un Maker!

Conectando el Arduino a la PC

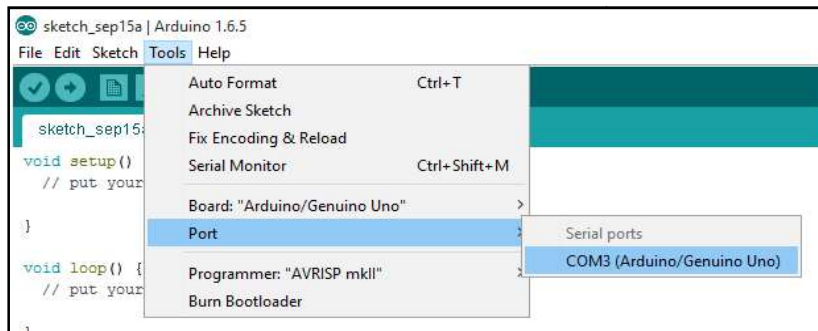
Conecta tu Arduino UNO al computador utilizando el cable USB, luego, en el Arduino IDE selecciona el modelo de la placa que estás utilizando: en nuestro caso Arduino UNO.

Tools > Board: "Arduino / Genuino Uno" > Arduino / Genuino Uno.



Luego, deberás asegurarte que el Arduino IDE tenga seleccionado el mismo puerto serie al que estás conectado tu Arduino.

Tools > Port > *Selecciona el puerto correcto, que estará marcado con el nombre de tu placa.*



Ahora si, finalmente estás listo para empezar a darle ordenes a tu Arduino.

Conceptos básicos de programación en Arduino

La programación de Arduino es la programación de un micro controlador. Esta era antes una tarea que solo los ingenieros y especialistas podían realizar, pero ahora Arduino lo ha extendido a todo el público. Arduino ha logrado democratizar la tecnología. A continuación verás unos ejemplos de programación muy básicos con los que le podrás decir a tu Arduino cómo hacer lo que tú quieras.

Configura pines para comportarse como una entrada o una salida

`pinMode(pin, INPUT);` Para configurar pin como entrada

`pinMode(pin, OUTPUT);` Para configurar pin como salida

Configurar una salida digital como un uno (1) o un cero (0)

`digitalWrite(pin, HIGH)` ; Para configurar salida como un uno (1) lógico

`digitalWrite(pin, LOW);` Para configurar salida como un cero (0) lógico

Leer el valor de un pin configurado como entrada digital

`digitalRead(Pin)` ; Devuelve el valor de la entrada digital (1 o 0)

Leer el valor de un pin configurado como entrada analógica

`analogRead(Pin);` Devuelve el valor de la entrada analógica (0 – 1023)

Imprimir un mensaje en la pantalla

`Serial.print("Todos podemos ser un Maker");`

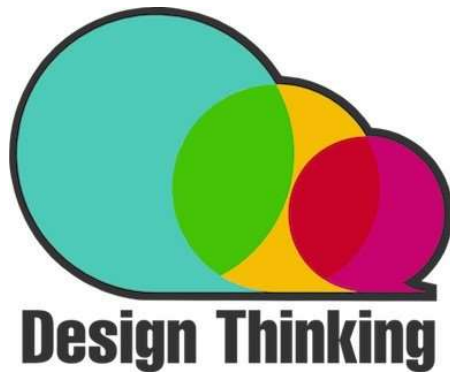
En la página oficial de Arduino podrás encontrar una lista mucho más detallada de los comandos de programación para Arduino, sigue el siguiente enlace:
<https://www.arduino.cc/reference/en/>



METODOLOGIA MAKER PARA TUS PROYECTOS

“DESIGN THINKING”

Es un método utilizado a nivel mundial para generar ideas innovadoras, su proceso se orienta en entender y dar solución a las necesidades reales de los usuarios. Proviene de la forma en la que trabajan los diseñadores de productos. De ahí su nombre, que en español significa "*Pensamiento de Diseño*": La forma en la que piensan los diseñadores. El Design Thinking se desarrolla siguiendo un proceso orientado a cumplir tres objetivos.



La generación de empatía: hay que entender los problemas, necesidades y deseos de los usuarios implicados en la solución que estamos buscando. Independientemente de qué estemos desarrollando, siempre conllevará a la interacción con personas. Satisfacerlas es la clave de un resultado exitoso.

La generación de prototipos: que defiende que toda idea debe ser validada antes de asumirse como correcta. El Design Thinking propicia la identificación de fallos, para que cuando demos con la solución deseada, éstos ya se hayan solventado.

El trabajo en equipo: que pone en valor la capacidad de los individuos de aportar singularidad. Cada persona piensa diferente y es ahí cuando toma valor esta iniciativa.

El proceso de Design Thinking se compone de cinco etapas. No es un proceso lineal. En cualquier momento podrás ir hacia atrás o hacia delante si lo consideras necesario, puedes ir saltando incluso a etapas no consecutivas.

¿Las cinco etapas del proceso?

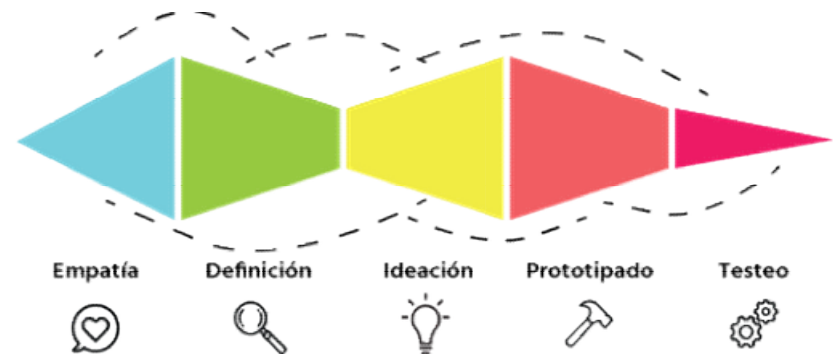
EMPATIZA: El proceso de Design Thinking comienza con una profunda comprensión de las necesidades de los usuarios implicados en la solución que estemos desarrollando, y también de su entorno. Debemos ser capaces de ponernos en la piel de dichas personas para ser capaces de generar soluciones consecuentes con sus realidades.

DEFINE: Durante la etapa de Definición, debemos filtrar la información recopilada durante la fase de Empatía y quedarnos con lo que realmente aporta valor y nos lleva al alcance de nuevas perspectivas interesantes. Identificaremos problemas cuyas soluciones serán clave para la obtención de un resultado innovador.

IDEA: La etapa de Ideación tiene como objetivo la generación de un sinfín de opciones. No debemos quedarnos con la primera idea que se nos ocurra. En esta fase, las actividades favorecen el pensamiento expansivo y debemos eliminar los juicios de valor. A veces, las ideas más estrambóticas son las que generan soluciones visionarias.

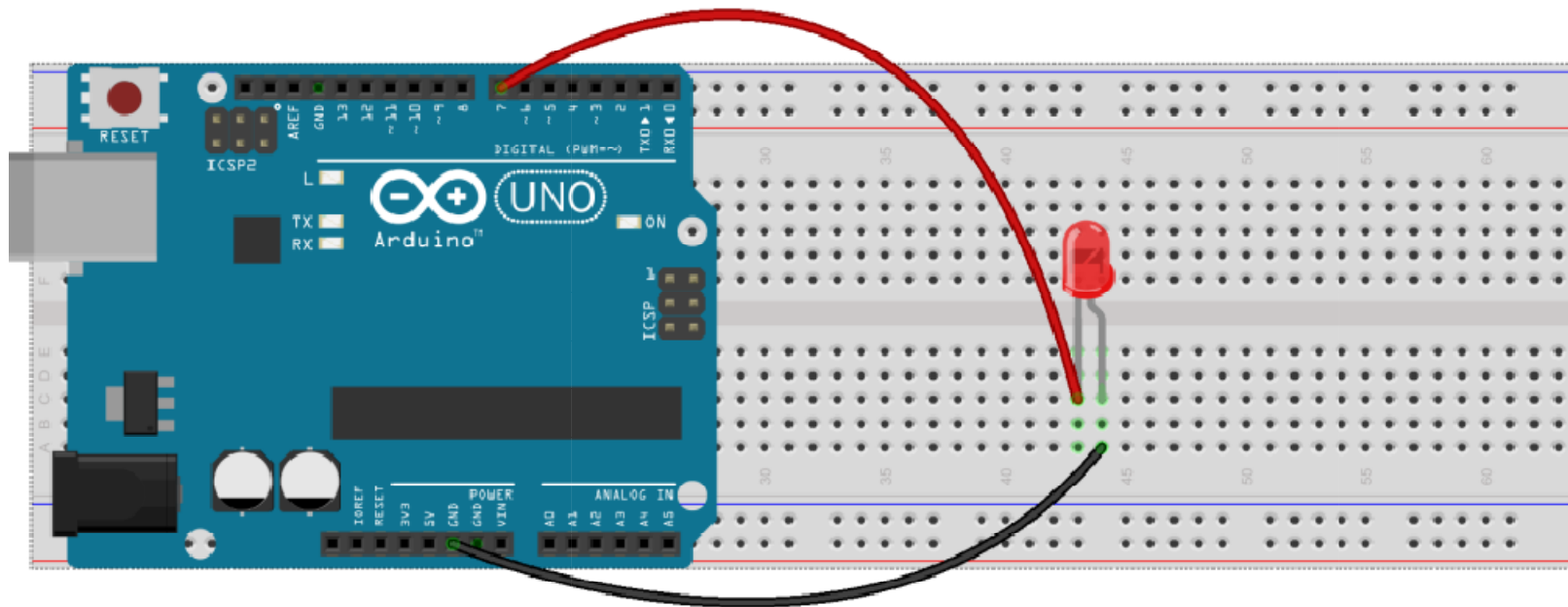
PROTOTIPA: En la etapa de Prototipado volvemos las ideas realidad. Construir prototipos hace las ideas palpables y nos ayuda a visualizar las posibles soluciones, poniendo de manifiesto elementos que debemos mejorar o refinar antes de llegar al resultado final.

TESTEA: Durante la fase de Testeo, probaremos nuestros prototipos con los usuarios implicados en la solución que estemos desarrollando. Esta fase es crucial, y nos ayudará a identificar mejoras significativas, fallos a resolver, posibles carencias. Durante esta fase evolucionaremos nuestra idea hasta convertirla en la solución que estábamos buscando.



PROYECTOS BÁSICOS CON ARDUINO

Parpadeo de un LED: el 'Hola Mundo' de Arduino



Código del programa (IDE)



```
Arduino IDE: Blink | Arduino 1.8.6 (Windows Store 1.8.14.0)
File Edit Sketch Tools Help

Blink $

/*
  TU MUNDO MAKER
  TODOS PODEMOS SER UN MAKER
*/

// PROYECTO 1. ENCENDER UN LED

char LED = 7; // Nombramos "LED" al pin digital 7

void setup() {
  pinMode(LED, OUTPUT); // Configuramos el pin del LED como salida
}

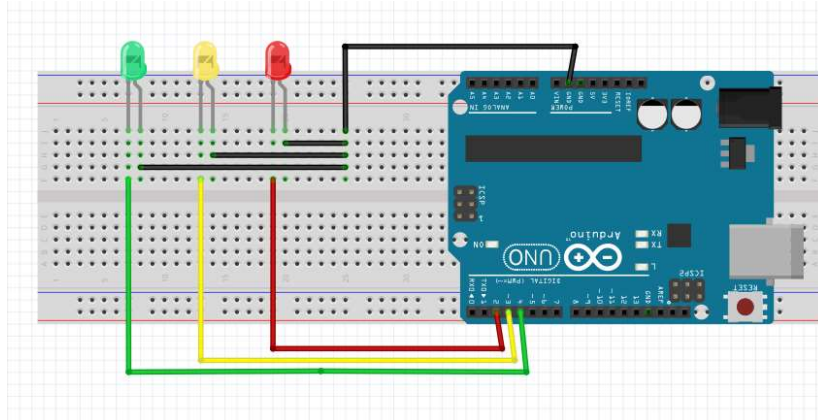
void loop() {
  digitalWrite(LED, HIGH); // Encendemos el LED
  delay(500); // Esperamos medio segundo
  digitalWrite(LED, LOW); // Apagamos el LED
  delay(500); // Esperamos medio segundo
}
```



Mis primeros pasos con Arduino

Guía básica para ser un Maker

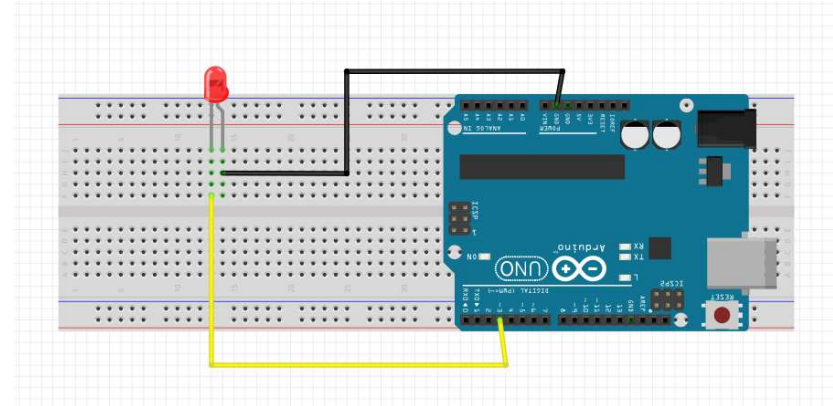
Semáforo



Código fuente (IDE)

```
void setup() {  
  pinMode(2, OUTPUT); //luz verde  
  pinMode(3, OUTPUT); //luz amarilla  
  pinMode(4, OUTPUT); //luz roja  
}  
  
void loop() {  
  digitalWrite(2, HIGH); // Encendemos luz verde  
  digitalWrite(3, LOW);  
  digitalWrite(4, LOW);  
  delay(5000);  
  digitalWrite(2, LOW); // Encendemos luz amarilla  
  digitalWrite(3, HIGH);  
  digitalWrite(4, LOW);  
  delay(1000);  
  digitalWrite(2, LOW); // Encendemos luz roja  
  digitalWrite(3, LOW);  
  digitalWrite(4, HIGH);  
  delay(5000);  
}
```

Dimerizando un LED (PWM)



Código fuente (IDE)

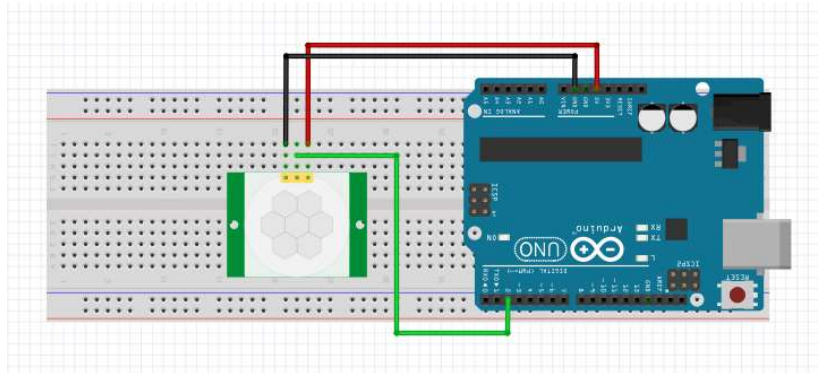
```
void setup() {  
  pinMode(3, OUTPUT); //salida del led en el pin 3  
}  
  
void loop() {  
  analogWrite(3, 10); //dimeriado a 10/255 (5%)  
  delay(100);  
  analogWrite(3, 25); //dimeriado a 25/255 (10%)  
  delay(100);  
  analogWrite(3, 50); //dimeriado a 50/255 (20%)  
  delay(100);  
  analogWrite(3, 100); //dimeriado a 100/255 (40%)  
  delay(100);  
  analogWrite(3, 150); //dimeriado a 150/255 (60%)  
  delay(100);  
  analogWrite(3, 200); //dimeriado a 200/255 (80%)  
  delay(100);  
  analogWrite(3, 255); //dimeriado a 255/255 (100%)  
  delay(100);  
}
```



Síguenos en nuestras redes: @tumundomaker

¡Todos podemos ser un Maker!

Detector de Presencia



Código fuente (IDE)

```
int ledPin = 13;           // Pin de salida del led
int inputPin = 2;          // Pin de entrada del sensor

int PIR_sensor = 0;        // Variable que marca el estado del sensor

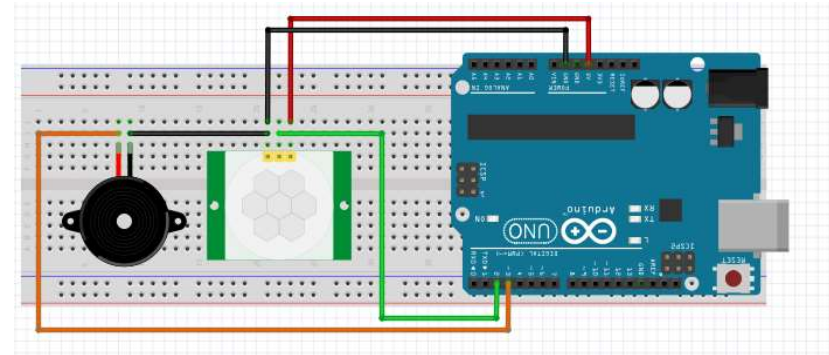
void setup() {
  pinMode(ledPin, OUTPUT); // Declaramos pin 13 como pin salida del led
  pinMode(inputPin, INPUT); // Declaramos pin 2 como pin entrada del sensor
}

void loop(){

  PIR_sensor = digitalRead(inputPin); // Tomamos la lectura del sensor

  if (PIR_sensor == HIGH) {          // Si el valor es uno (1)
    digitalWrite(ledPin, HIGH);      // Encender el LED
  }
  else {                             // Sino, entonces el valor es cero (0)
    digitalWrite(ledPin, LOW);       // Apagar el LED
  }
}
```

Alarma anti robo



Código fuente (IDE)

```
int ledPin = 13;           // Pin de salida del led
int inputPin = 2;          // Pin de entrada del sensor
int alarmaPin = 3;         // Pin de entrada de alarma

int PIR_sensor = 0;        // Variable que marca el estado del sensor

void setup() {
  pinMode(ledPin, OUTPUT); // Declaramos pin 13 como pin salida del led
  pinMode(alarmaPin, OUTPUT); // Declaramos pin 3 como pin salida del buzzer
  pinMode(inputPin, INPUT); // Declaramos pin 2 como pin entrada del sensor
}

void loop(){

  PIR_sensor = digitalRead(inputPin); // Tomamos la lectura del sensor

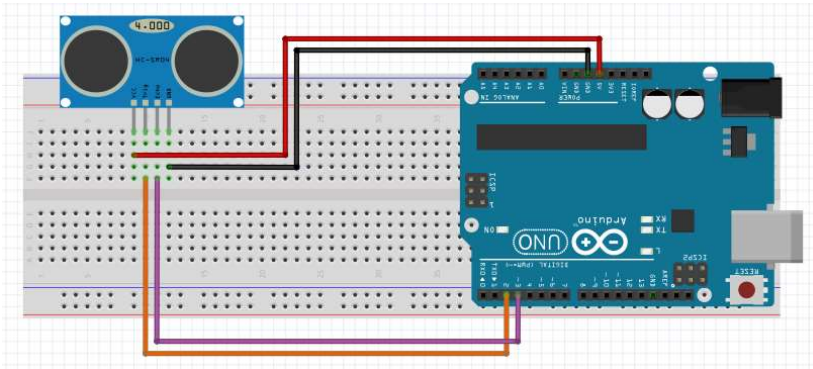
  if (PIR_sensor == HIGH) {          // Si el valor es uno (1)
    digitalWrite(ledPin, HIGH);      // Encender el LED y la alarma
    digitalWrite(alarmaPin, HIGH);
  }
  else {                             // Sino, el valor es cero (0)
    digitalWrite(ledPin, LOW);       // Apagar el LED y la alarma
    digitalWrite(alarmaPin, LOW);
  }
}
```



Mis primeros pasos con Arduino

Guía básica para ser un Maker

Medidor de distancia



Código fuente (IDE)

```
#include <Ultrasonic.h>           // Incluimos la librería del sensor
Ultrasonic ultrasonic(2,3);       // Configuramos sensor (Trigger,Echo)

int distancia = 0;                // Declaramos la variable que almacena distancia

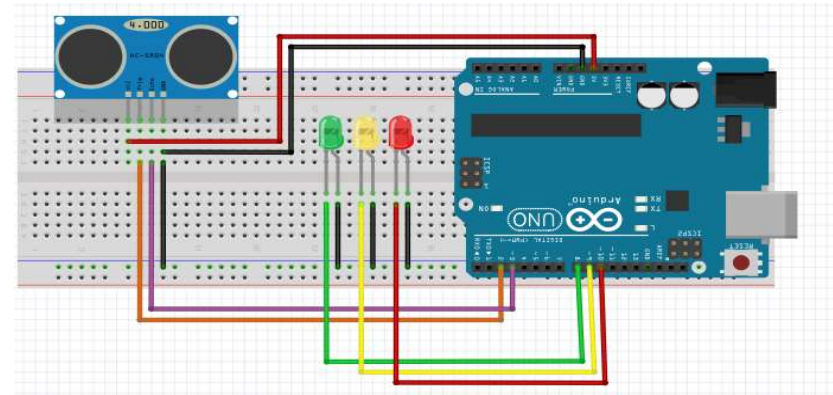
void setup() {
  Serial.begin(9600);             // Iniciamos comunicación serial
}

void loop(){

  distancia = ultrasonic.Ranging(CM); // Tomamos la lectura del sensor

  Serial.print(distancia);         // Mostramos la medición en centímetros
  Serial.println(" cm" );
  delay(100);
}
```

Alarma anti choque



Código fuente (IDE)

```
#include <Ultrasonic.h>           // Incluimos la librería del sensor
Ultrasonic ultrasonic(2,3);       // Configuramos sensor (Trigger,Echo)

int distancia = 0;                // Declaramos la variable que almacena distancia

void setup() {
  pinMode(8, OUTPUT);             // luz verde
  pinMode(9, OUTPUT);             // luz amarilla
  pinMode(10, OUTPUT);            // luz roja
}

void loop(){
  distancia = ultrasonic.Ranging(CM); // Tomamos la lectura del sensor

  if (distancia < 5) {            // Si la distancia es menor a 5cm, encendemos luz rojo
    digitalWrite(10, HIGH);
  } else if (distancia < 10) {    // Si la distancia es menor a 10cm, encendemos amarilla
    digitalWrite(9, HIGH);
  } else if (distancia < 20) {    // Si la distancia es menor a 20cm, encendemos luz verde
    digitalWrite(8, HIGH);
  }
  delay(100);
}
```



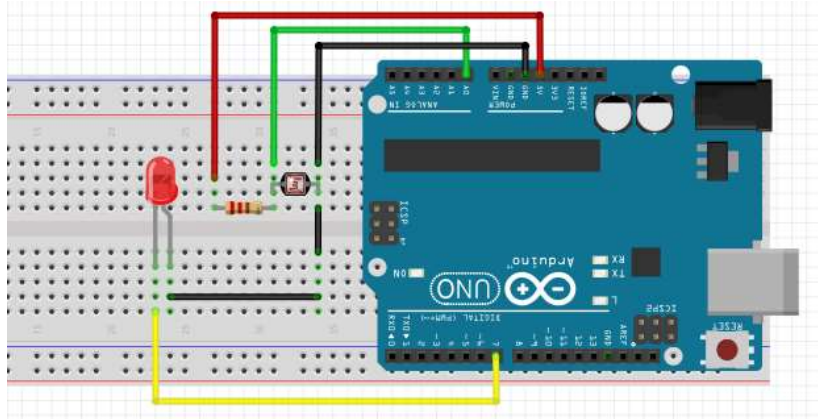
Síguenos en nuestras redes: @tumundomaker

¡Todos podemos ser un Maker!

Mis primeros pasos con Arduino

Guía básica para ser un Maker

Luz nocturna automática (con LED)



Código fuente (IDE)

```
int LEDPin = 7;
int LDRPin = A0;
int luz = 100;

void setup() {
  pinMode(LEDPin, OUTPUT);
  pinMode(LDRPin, INPUT);
}

void loop() {
  int input = analogRead(LDRPin);
  if (input > luz) {
    digitalWrite(LEDPin, HIGH);
  }
  else {
    digitalWrite(LEDPin, LOW);
  }
}
```



Síguenos en nuestras redes: @tumundomaker

¡Todos podemos ser un Maker!