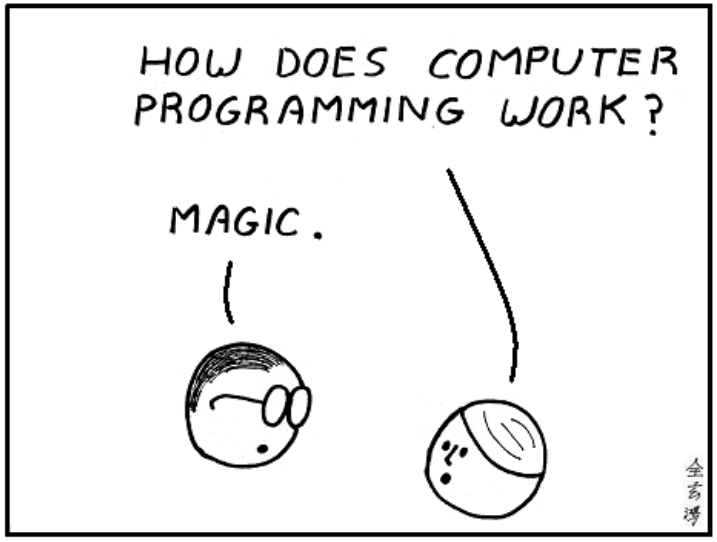
Introducción a la programación con Arduino

Para empezar…

- ¿Qué es la **programación**?
  - La acción de escribir programas informáticos
- ¿Qué es un **programa informático**?
  - Un conjunto de instrucciones que el ordenador puede entender y ejecutar
- ¿Qué es un **lenguaje de programación**?
  - Es la forma que tenemos las personas de comunicarnos con el ordenador



¿Lenguajes de programación?

- Hay muchos y de diferentes tipos:
  - Lenguaje de máquina

```
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
11001010 11001010 11110101 00101011
11001010 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
```

  - Lenguaje ensamblador

```
MODEL   Small
Stack           100h
DATASEG
Mensaje         DB 'Hola, Mundo',13,10,'$'
CODESEG
Inicio:         mov ax,@data
                mov ds,ax               ; DS ahora apunta al segmento de datos.
                mov ah,9                ; Función del DOS para impresión de cadenas.
                mov dx,OFFSET Mensaje   ; Apuntador al mensaje 'Hola Mundo'.
                int 21h
                mov ah,4Ch              ; Transferencia del control al DOS.
                int 21h
END             Inicio
```

  - Lenguaje de medio/alto nivel de abstracción

```
int a = 10;
int b = 5;
int suma = a + b;
```

El código fuente

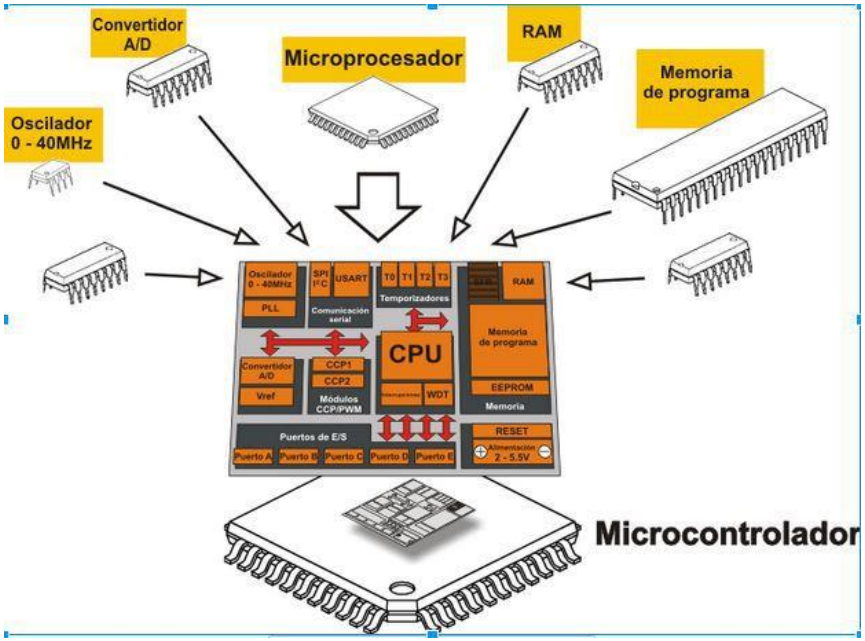
- **Código** o **código fuente**: El conjunto de órdenes que le damos al ordenador. Esas órdenes son también llamadas **instrucciones**
  - Suelen ser textos escritos siguiendo una reglas determinadas (**sintaxis**)

Algunos conceptos

- **Sintáxis**: Las reglas utilizadas por un lenguaje de programación en particular (similar a la gramática del español o del inglés)
- **Programar**: La acción de **escribir código** fuente con la intención de crear una aplicación o un programa informático
- **Entrada**: Los valores que el programa pide a los usuarios que introduzca por teclado (p.e., un nombre, un número)
- **Salida**: Los mensajes que el programa muestra en la pantalla cuando se ejecuta (p.e., el resultado de una multiplicación)
- **Consola**: Una parte de la pantalla en la que el programa muestra su **salida**

…pero, ¿qué es Arduino?

- Es un **microcontrolador**
  - Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria



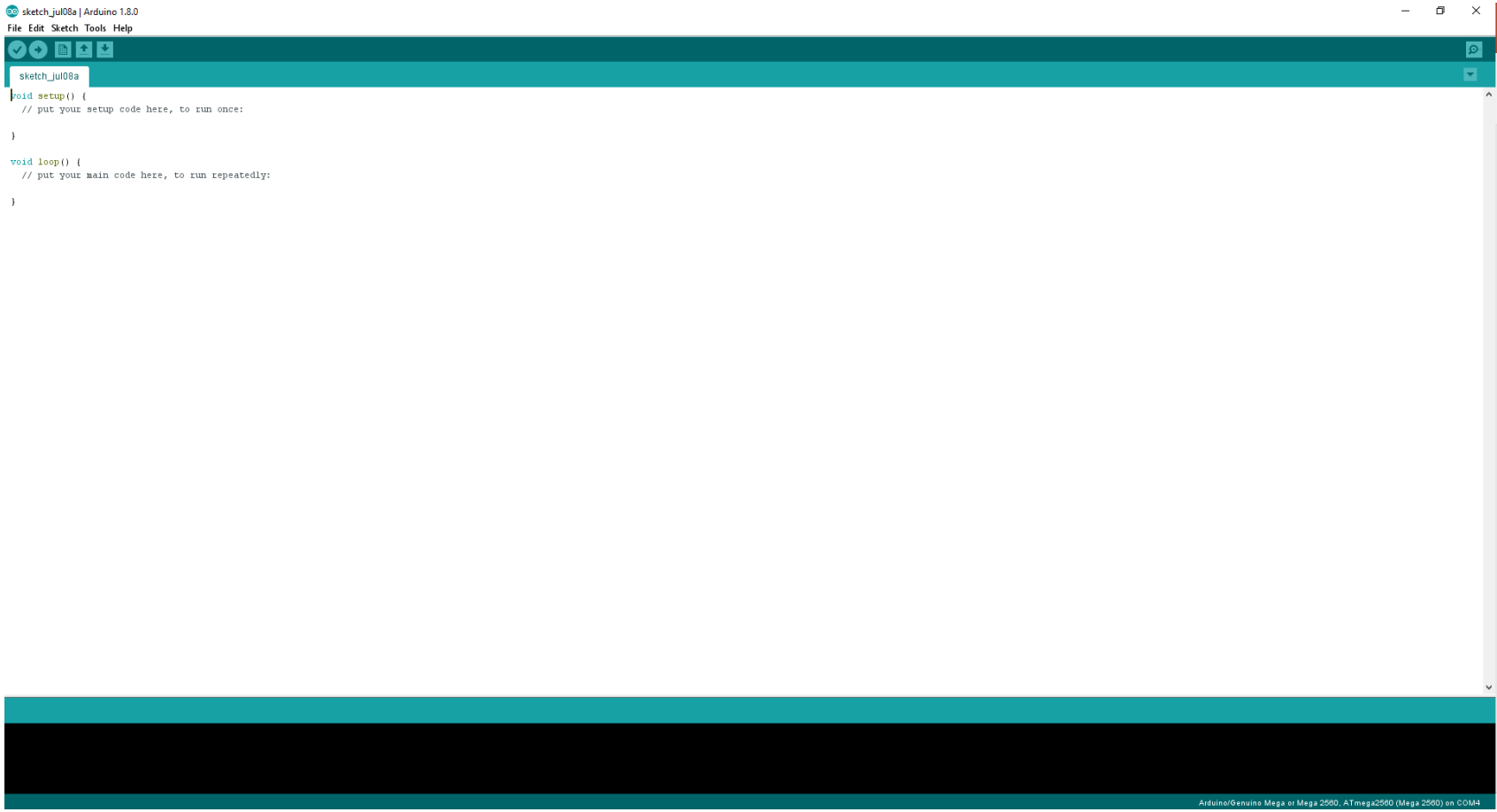
¿Para qué sirve?

HOLA MUNDO

¿Cómo escribir un programa?

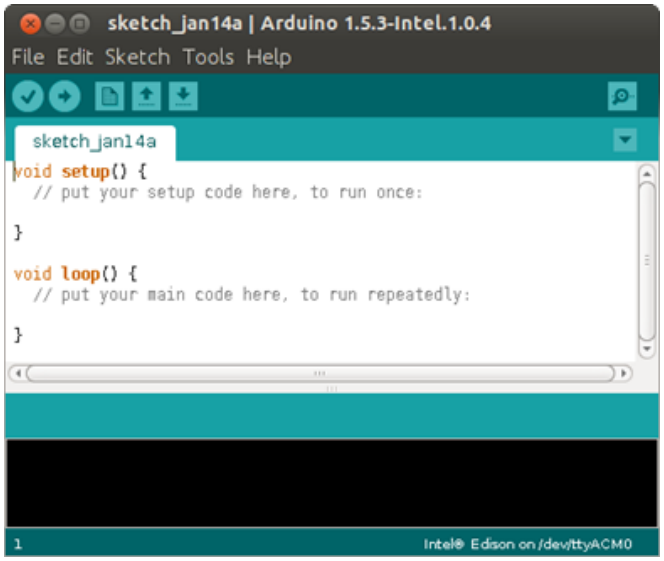
- Por ejemplo, con el **lenguaje Arduino**, sólo necesitamos
  - Un editor de texto
  - Archivos (ficheros) con extensión .ino (por ejemplo **hola.ino**)
  - …Un compilador

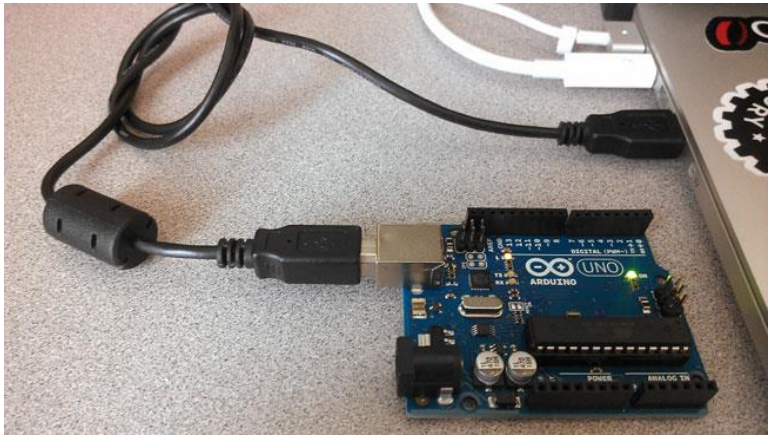
Arduino IDE


sketch_jul08a | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jul08a
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

Arduino IDE

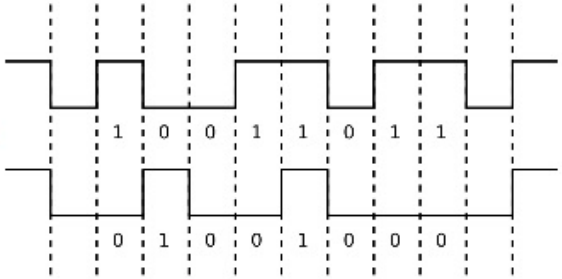
- La principal forma de comunicación entre un Arduino y el ordenador es a través del **puerto serie**





¿Qué es el puerto serie?

- **Un puerto**, en informática, es el nombre genérico que damos a los elementos que permiten la comunicación entre dos dispositivos
- Centrándonos en Arduino, son la forma principal de comunicar una placa Arduino con un ordenador

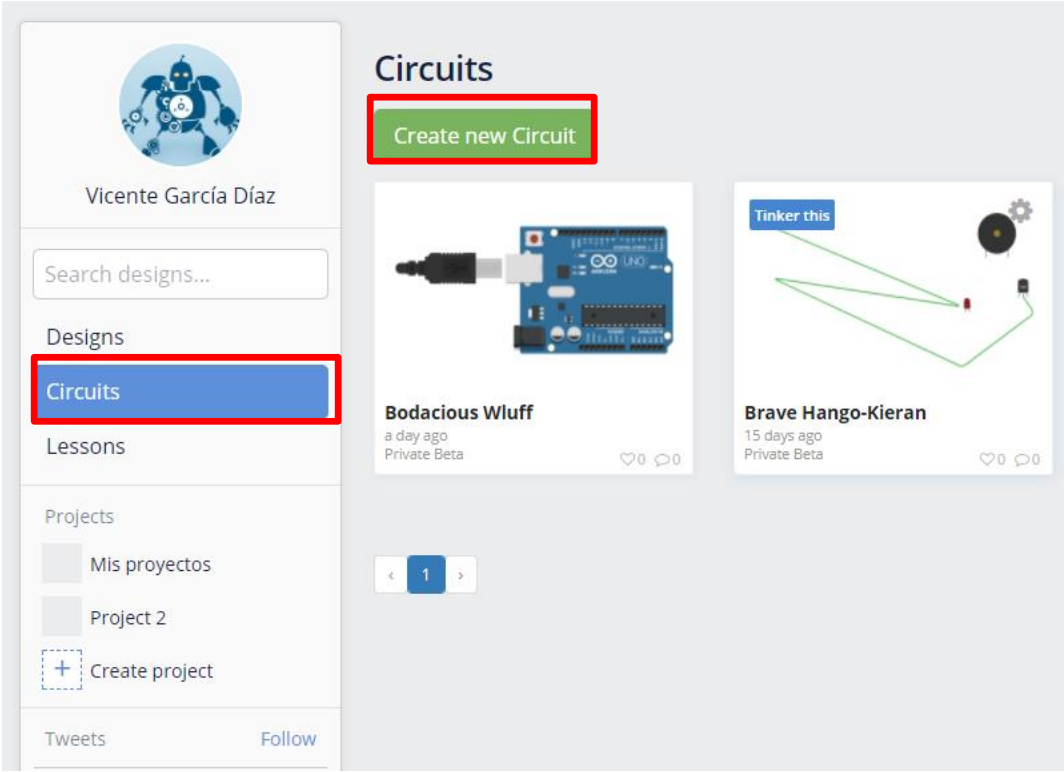


¿Y sin microcontrolador?

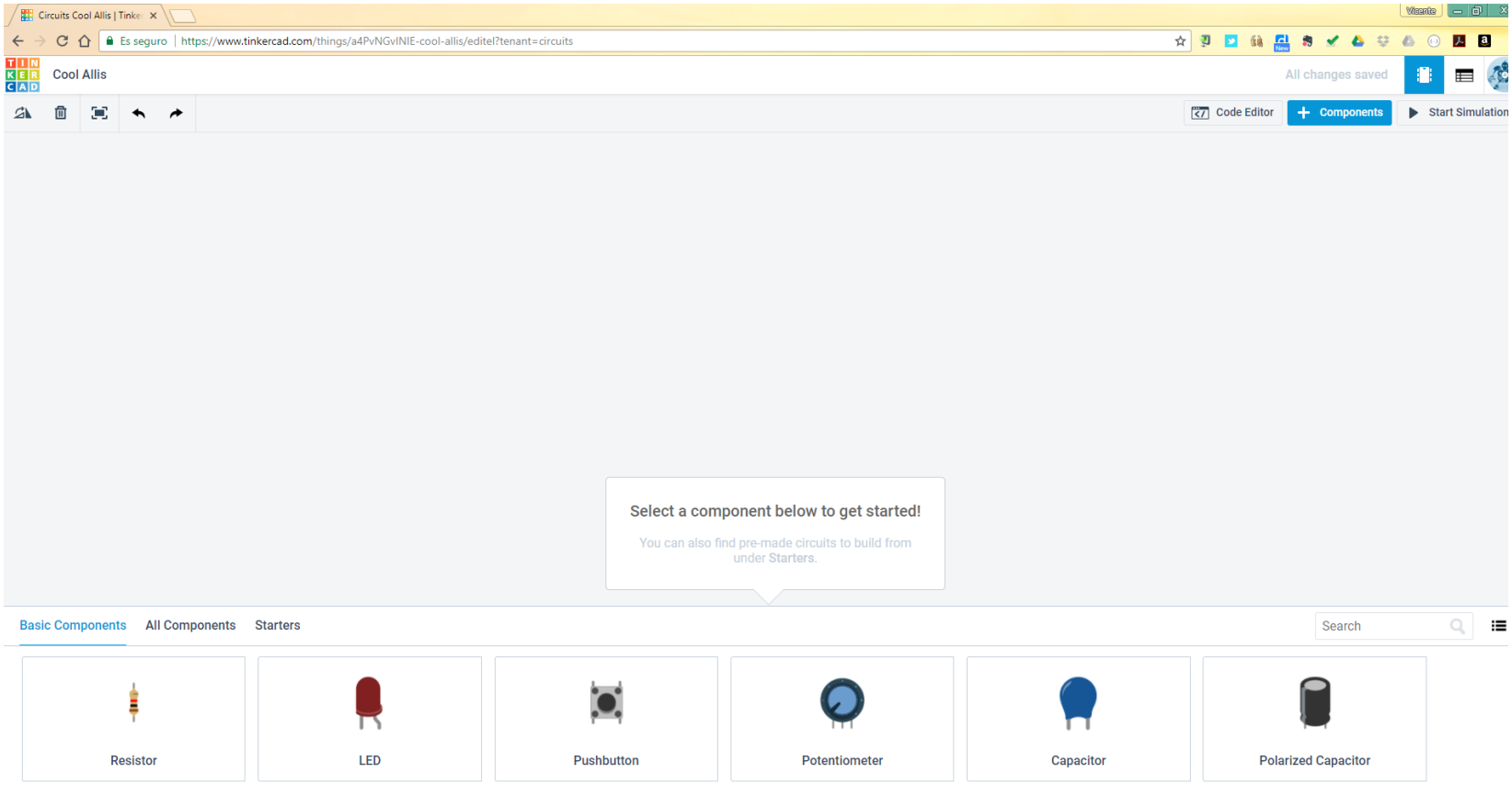
- Necesitaremos usar un simulador
- Hay muchos como por ejemplo el integrado en Tinkercad https://www.tinkercad.com/

Diseño de circuitos electrónicos

- Completamente integrado en el software

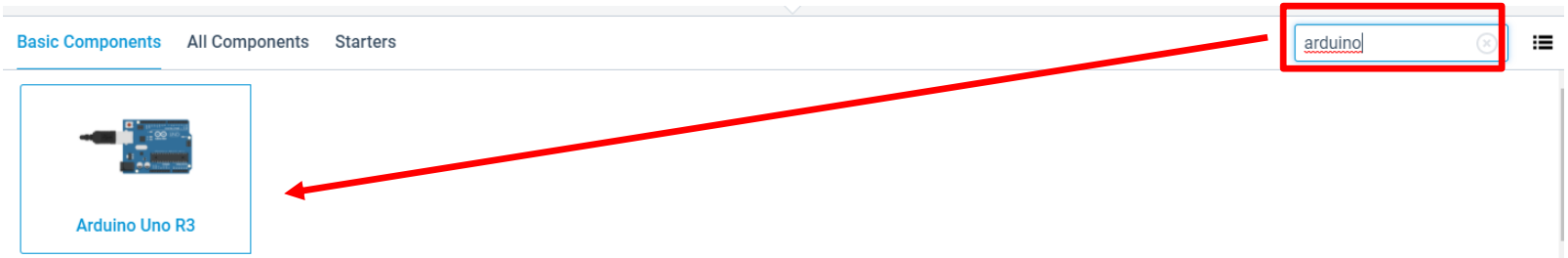


Panel de trabajo



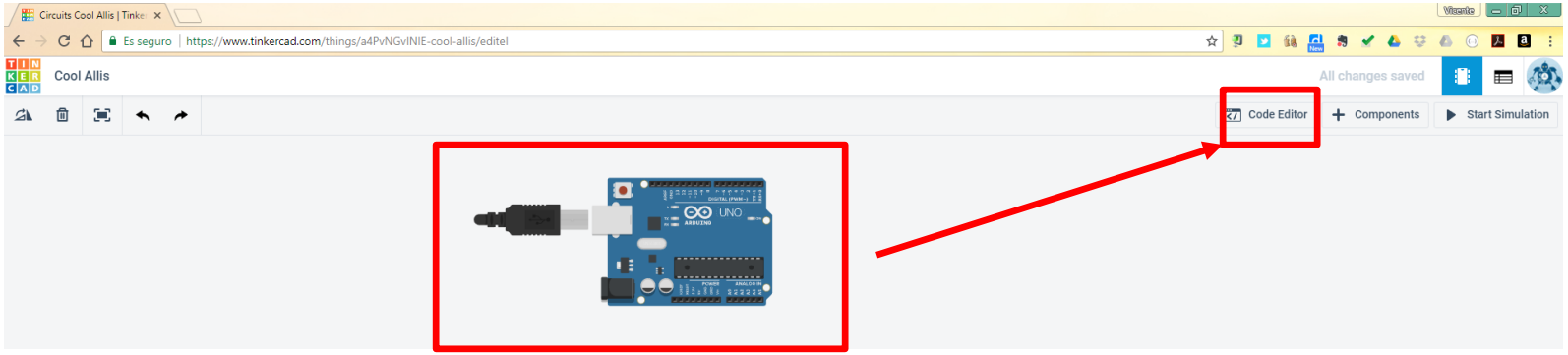
Inserción de un Arduino en el circuito

- Buscamos Arduino para trabajar con un Arduino Uno R3



Editor de código

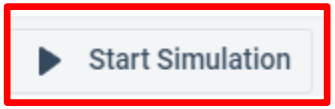
- Con el Arduino ya podemos comenzar a programar



Ventana de trabajo

Empezamos

- Código inicial




```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Vamos a simplificar las cosas

- Este es un programa básico
- …realmente no hace nada

```
//Este es un programa básico

void setup() {
}

void loop() {
}
```

Comunicación con el puerto serie

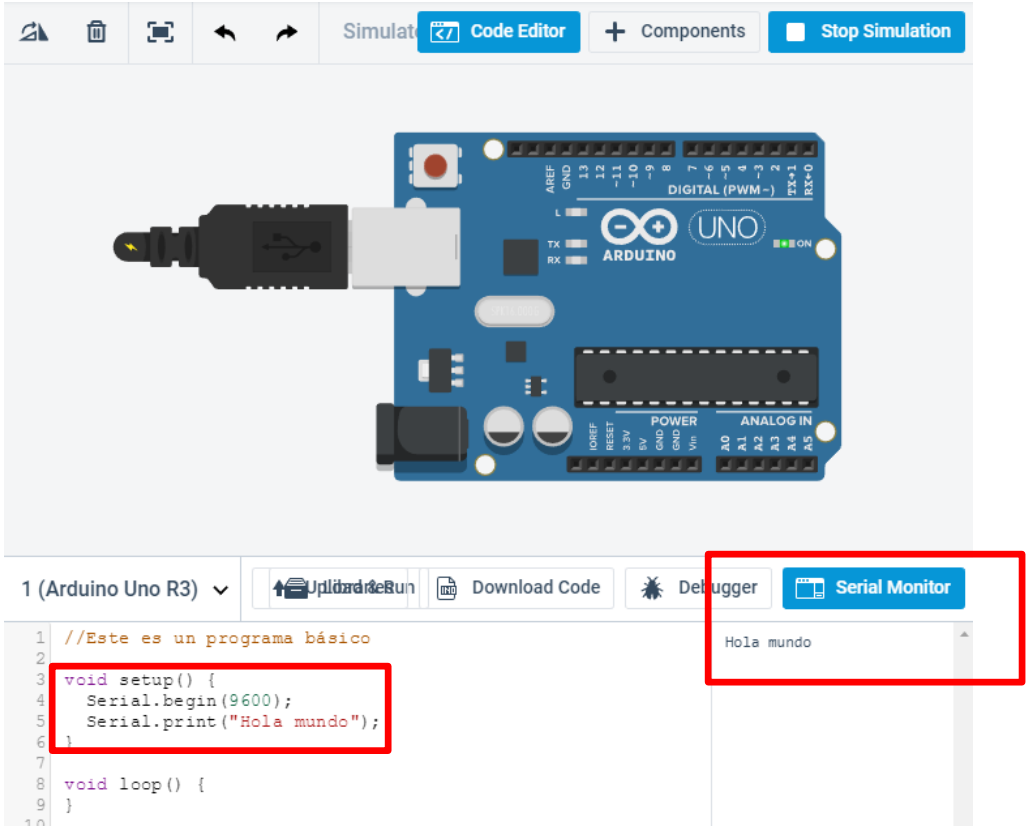
- Antes de nada tenemos que abrir la comunicación con
  - `Serial.begin(VELOCIDAD)`
  - VELOCIDAD puede ser por ejemplo 9600 (valor típico)

- Después podemos enviar datos desde Arduino con
  - `Serial.print(DATO)` – Sin salto de línea
  - `Serial.println(DATO)` – Con salto de línea

- Son ejemplos de lo que llamaremos **funciones**
- Todas las instrucciones se terminan en **;**

Nuestro primer código

- Ya podemos ver nuestro **Hola Mundo** en el puerto serie



¿Cómo podemos detener el código?

- Existe una función `delay(MILISEGUNDOS)` que podremos utilizar para detener el código un número determinado de milisegundos

Delay - reto

¿Puedes hacerlo?

- ¿Podrías hacer que el código muestre una letra "A" cada 2 segundos, separadas una en cada línea?

ARCHIVO RetoDelay.ino

VARIABLES Y TIPOS DE DATOS

Tipos básicos de datos

- Los programas manejan **información**
- La información básica que manejaremos será de diferentes tipos:

Tipo de dato	Tipo de dato en Arduino	Ejemplo
Número enteros	`int`	5
Números reales	`float`	1.5f
Caracteres	`char`	'Z'
Boolean	`bool`	0
Cadena de texto	`String`	"casa"

- Una variable es un nombre a través del cual podemos almacenar y manipular valores de un determinado tipo (operador =)
  - `[TIPO_DE_DATO] NOMBRE_VARIABLE;`
  - `[TIPO_DE_DATO] NOMBRE_VARIABLE = VALOR;`

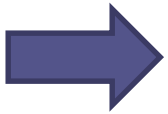
Tipos básicos de datos - ejemplo

- Un ejemplo básico que inserta en el puerto serie los valores

```
int entero = 10;
float real = 10.3;
char caracter = 'A';
bool logico = true;
String cadena = "Mi casa";

void setup() {
  Serial.begin(9600);
  Serial.println(entero);
  Serial.println(real);
  Serial.println(caracter);
  Serial.println(logico);
  Serial.println(cadena);
}

void loop() {
}
```



```
10
10.30
A
1
Mi casa
```

Detalles sobre la impresión

- `Serial.print` y `Serial.println` admiten un segundo valor (parámetro) en algunos casos
  - Con número **enteros**
    - **BIN, OCT, DEC, HEX**
  - Con número **reales**
    - El número de decimales que se quiere utilizar

- Además, podemos introducir algunos caracteres especiales:
  - **/n** → una nueva línea
  - **/t** → un tabulador

Números - reto

¿Puedes hacerlo?

- Partiendo del siguiente código

```
int entero = 100;
float real = 12.329876;
```

- ¿Cómo lo completarías para mostrar el valor del número entero en binario?
- ¿Cómo lo completarías para mostrar sólo 3 decimales del número real?

ARCHIVO RetoNumeros.ino

Obtención de valores (entrada)

- Seguiremos trabajando con `Serial`

- `Serial.available()` devuelve el número de bytes disponibles para ser leídos del puerto serie
- `while(Serial.available() == 0)` mientras no haya nada que disponible…

- `Serial.parseInt()` devuelve el primer número entero disponible en el puerto serie

Obtención de valores - reto

¿Puedes hacerlo?

1. Realiza un programa que espere para leer dos números y los imprima por pantalla

Obtención de valores – reto (II)

¿Puedes hacerlo?

1. Realiza un programa que pida un número, lo muestre por pantalla, espere 4,5 segundos y pida otro número – y así indefinidamente

ARCHIVO RetoObtencionValores2.ino

Operadores aritméticos

- Ya conocemos el operador básico de asignación

Descripción	Uso en Arduino	Ejemplo
Suma	+	-5 + 3
Resta	-	-5 - 3
Multiplicación	*	-5 * 3
Cociente división (entero)	/	-5 / 3
Resto división (entero)	%	-5 % 3

- Los operadores aritméticos sirven para realizar operaciones matemáticas básicas

Operadores aritméticos (II)

```
int numero1;
int numero2;
int suma;
int resta;
int mult;
int cociente;
int resto;

void setup() {
  Serial.begin(9600);

  Serial.println("Introduce dos numeros");
  while(Serial.available() == 0);
  numero1 = Serial.parseInt();
  numero2 = Serial.parseInt();

  suma = numero1 + numero2;
  resta = numero1 - numero2;
  mult = numero1 * numero2;
  cociente = numero1 / numero2;
  resto = numero1 % numero2;

  Serial.println("La suma es:" + String(suma));
  Serial.println("La resta es: " + String(resta));
  Serial.println("La multiplicacion es: " + String(mult));
  Serial.println("El cociente es: " + String(cociente));
  Serial.println("El resto es: " + String(resto));
}

void loop() {
}
```


ARCHIVO OperadoresAritmeticos.ino

Operadores aritméticos - reto

¿Puedes hacerlo?

1. Realiza exactamente lo mismo que en el ejemplo anterior pero con 10 líneas menos de código

ARCHIVO RetoOperadoresAritmeticos.ino

Elementos matemáticos

- `min()`
  - Calcula el mínimo de dos números
- `max()`
  - Calcula el máximo de dos números
- `abs()`
  - Calcula el valor absoluto de un número
- `pow()`
  - Calcula el valor de número elevado a una potencia
- `sqrt()`
  - Calcula la raíz cuadrada de un número

Elementos matemáticos - reto

¿Puedes hacerlo?

- Utiliza las fórmulas matemáticas para mostrar:
  - El mínimo entre 100 y 12.329876
  - El máximo entre 100 y 12.329876
  - El valor absoluto de -12.329876
  - El resultado de elevar 100 al cuadrado
  - La raíz cuadrada de 12.329876

ARCHIVO RetoElementosMatematicos.ino

Operadores relacionales

Descripción	Uso en Arduino	Ejemplo
Mayor	>	`5 > 3.3`
Mayor o igual	>=	`5 >= 3.3`
Menor	<	`5 < 3.3`
Menor o igual	<=	`5 <= 3.3`
Igual	==	`5 == 3.3`
Diferente	!=	`5 != 3.3`

- Los operadores relacionales comparan dos operandos y devuelven verdadero - `true` (`1`) o falso - `false` (`0`)

Operadores relacionales II

```
int numero1;
int numero2;
String resultado;

void setup() {
  Serial.begin(9600);
  Serial.println("Introduce dos numeros");
  while (Serial.available() == 0);
  numero1 = Serial.parseInt();
  numero2 = Serial.parseInt();

  Serial.print("Operador == ==>"); Serial.println(numero1 == numero2);
  Serial.print("Operador != ==>"); Serial.println(numero1 != numero2);
  Serial.print("Operador > ==>"); Serial.println(numero1 > numero2);
  Serial.print("Operador < ==>"); Serial.println(numero1 < numero2);
  Serial.print("Operador >= ==>"); Serial.println(numero1 >= numero2);
  Serial.print("Operador <= ==>"); Serial.println(numero1 <= numero2);
}

void loop() {
}
```


ARCHIVO OperadoresRelacionales.ino

Operadores relacionales - reto

¿Puedes hacerlo?

1. Ejecuta el ejemplo anterior e introduce 5 y 5.4
    1. ¿Qué sucede?
    2. ¿Podrías evitarlo cambiando algo en el código?

ARCHIVO RetoOperadoresRelacionales.ino

Conversiones de tipos

- En ocasiones tenemos que realizar conversiones explícitas que nos permiten convertir un tipo de dato en otro

```
float numero_decimal = 60.5;
int numero_entero = (int)numero_decimal;
//el ASCII 60 se corresponde con el caracter <
char caracter = (char)numero_entero;
int numero_entero2 = (int)caracter;

void setup() {
  Serial.begin(9600);
  Serial.println(numero_decimal);
  Serial.println(numero_entero);
  Serial.println(caracter);
  Serial.println(numero_entero2);
}

void loop() {
}
```

ARCHIVO Conversiones.ino

Conversiones - reto

¿Puedes hacerlo?

1. Crea un pequeño código en el que dividas dos números enteros (9 y 3) y muestra el resultado

2. Crea un pequeño código en el que dividas dos números enteros (10 y 3) y muestra el resultado

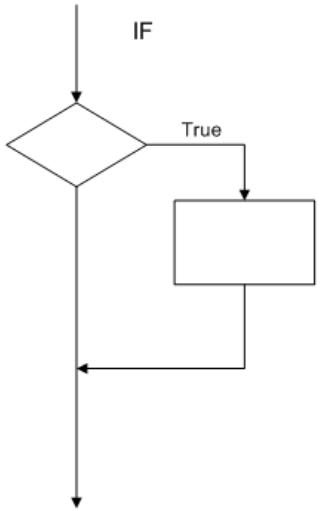
3. ¿Son correctos los resultados? ¿Cómo arreglarías lo que no está bien?

ESTRUCTURAS DE CONTROL CONDICIONAL

if

- Se utilizan las instrucciones para determinar el camino del código:
  - **if**…
  - if…else
  - if…else…if…
- La forma más básica es simplemente con un `if`

```
if (condicion) {
     //se ejecutan instrucciones aquí
     //si se cumple la condición
}
```



Condiciones

- Las **condiciones** son expresiones que pueden ser `true` o `false`
- Los **operadores relacionales** siempre devolverán `true` o `false`

Descripción	Uso en Arduino	Ejemplo
Mayor	`>`	`5 > 3.3`
Mayor o igual	`>=`	`5 >= 3.3`
Menor	`<`	`5 < 3.3`
Menor o igual	`<=`	`5 <= 3.3`
Igual	`==`	`5 == 3.3`
Diferente	`!=`	`5 != 3.3`

if - ejemplo

- Programa que muestra un mensaje si la edad del usuario es mayor que 18, diciéndole que es mayor edad

```
int edad = 20;
String mensaje;

void setup() {
  Serial.begin(9600);
  Serial.println("Inicio del programa");
  if (edad > 18) {
    mensaje = "Eres mayor de edad. Tienes ";
    mensaje += edad;
    mensaje += " años";
    Serial.println(mensaje);
  }
  Serial.println("Fin del programa");
}

void loop() {
}
```

Equivalente a:

```
Serial.println("Eres mayor de edad. Tienes " + String(edad) + " años");
```


ARCHIVO CondicionalIf.ino

if - reto

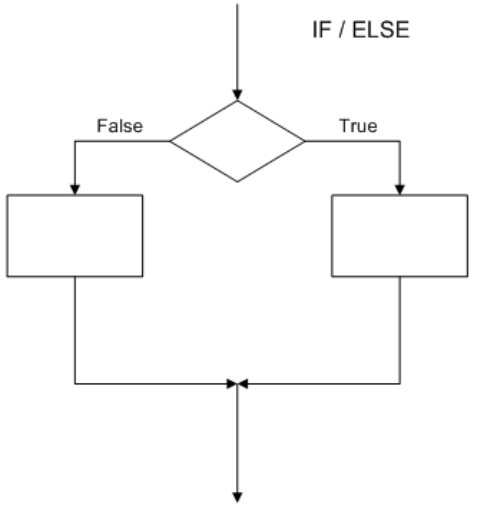
¿Puedes hacerlo?

1. Pide dos edades al usuario (dos números)
2. Escribe las dos edades que el usuario ha introducido
3. Si la primera edad introducida es menor o igual que la segunda, entonces se deberá mostrar un mensaje en la pantalla indicándolo

if else

- Se utilizan las instrucciones:
  - `if…`
  - **`if…else`**
  - `if…else…if…`
- Podemos determinar qué hacer si una condición no se cumple

```
if (condicion) {
    //hacemos algo
}
else {
    //hacemos otra cosa
}
```



if else - ejemplo

- Programa que muestra un mensaje si la edad del usuario es mayor o igual que 18, diciéndole que es mayor edad
- Si el usuario no es mayor de edad, entonces mostrará un mensaje diferente

```
int edad = 10;

void setup() {
  Serial.begin(9600);
  Serial.println("Inicio del programa");

  if (edad >= 18) {
    Serial.println("Eres mayor de edad");
  }
  else {
    Serial.println("Aun no eres mayor de edad");
  }

  Serial.println("Fin del programa");
}

void loop() {
}
```


ARCHIVO CodicionalIfElse.ino

if else - reto

¿Puedes hacerlo?

1. Modifica el programa anterior para pedir la edad al usuario

2. Si el usuario es menor de edad, el programa deberá indicarle cuántos años le faltan para ser mayor de edad

ARCHIVO RetoCondicionalIfElse.ino

Operadores lógicos

- Las **condiciones** son expresiones que pueden ser `true` o `false`
- Se pueden combinar **operadores relacionales** utilizando **operadores lógicos**

Descripción	Uso en Arduino	Ejemplo
And	`&&`	`((5 > 3.3) && (true))`
Or	`||`	`((5 == 3.3) || (4 < 10))`
Not	`!`	`!(10 == 13)`

- Dependiendo de los valores de los operandos, el resultado sería:

a	b	!a	a && b	a || b
false	**false**	true	false	false
false	**true**	true	false	true
true	**false**	false	false	true
true	**true**	false	true	true

if else – reto II

¿Puedes hacerlo?

1. Pide un número al usuario

2. Si el número es múltiplo de 10 o de 6 entonces se deberá mostrar el mensaje "**El numero introducido es multiplo de 10 o de 6**" al usuario

3. Si el número no es múltiplo de 10 o de 6 entonces se deberá mostrar el mensaje "**El numero introducido NO es multiplo de 10 o de 6**" al usuario

if else – reto III

¿Puedes hacerlo?

1. Cambia el programa anterior para que;
2. Si el número es múltiplo de 10 y de 6 entonces se deberá mostrar el mensaje "**El numero introducido es multiplo de 10 y de 6**" al usuario
3. Si el número no es múltiplo de 10 y de 6 entonces se deberá mostrar el mensaje "**El numero introducido NO es multiplo de 10 y de 6**" al usuario

ARCHIVO RetoCondicionalIfElse3.ino

if else if

- Se utilizan las instrucciones:
  - if…
  - if…else
  - **if…else…if…**
- Podemos utilizar varias condiciones con diferentes caminos

```
if (condicion) {
    //hacemos algo
}
else if (condicion2) {
    //o hacemos otra cosa
}
else if (condicion3) {
    //o puede que alguna otra cosa
}
else {
    //si no se cumple ninguna de las condiciones previas, hacemos lo que ponemos aquí
}
```

if else if - reto

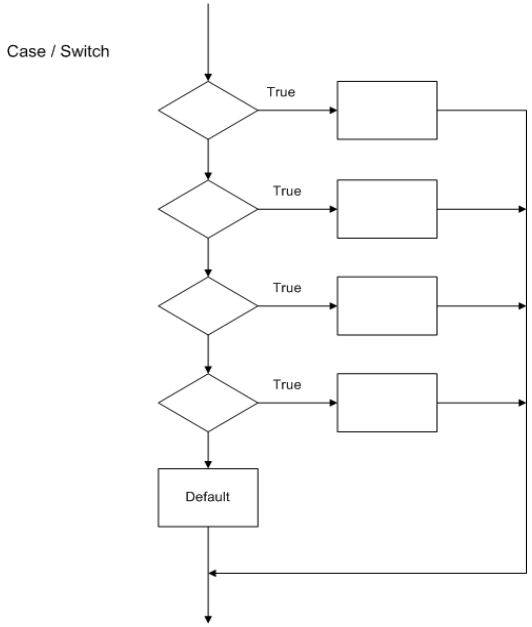
¿Puedes hacerlo?

1. Queremos un programa que pida al usuario un día de la semana
    1. Si el usuario introduce 1 => se refiere al lunes
    2. Si el usuario introduce 2 => se refiere al martes
    3. Si el usuario introduce 3 => se refiere al miércoles
    4. Si el usuario introduce 4 => se refiere al jueves
    5. Si el usuario introduce 5 => se refiere al viernes
    6. Si el usuario introduce 6 => se refiere al sábado
    7. Si el usuario introduce 7 => se refiere al domingo
2. Si es lunes o martes, el programa debe mostrar un mensaje "Aún estamos comenzando la semana"
3. Si es miércoles o jueves, el programa debe mostrar un mensaje "Ya estamos a mitad de semana"
4. Si es viernes, el programa debe mostrar un mensaje "Estamos a punto de terminar la semana"
5. Si es cualquier otro día el programa debe mostrar un mensaje "Ya es fin de semana"

ARCHIVO RetoCondicionalIfElseIf.ino

switch

- Cuando tenemos muchos `if else if` conviene cambiar la estructura para dejar el código más claro:
- La solución es utilizar `switch`




```
switch (variable) {
case valor1:
   //hacemos algo
   break;
case valor2:
   //hacemos otra cosa
   break;
case valor3:
   //hacemos otra cosa diferente
   break;
default:
   //si no se cumple ninguna otra opción, por defecto se hace esto
   break;
}
```

switch - reto

¿Puedes hacerlo?

1. Modifica el programa anterior para convertir el `if else if` en una estructura `switch`

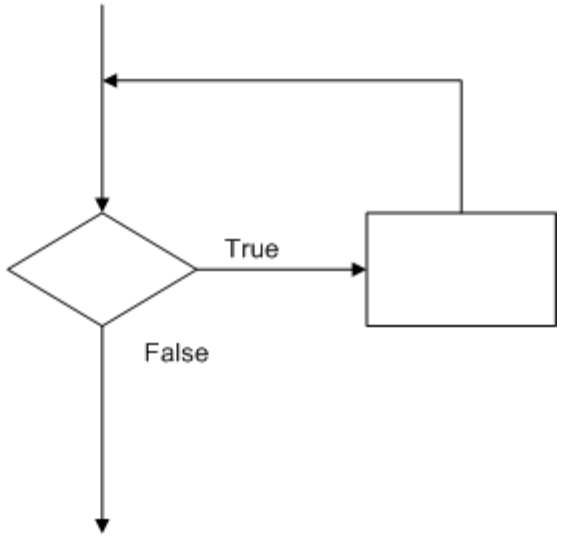
ARCHIVO RetoCondicionalIfElseIfSwitch.ino

ESTRUCTURAS DE CONTROL REPETITIVAS

while

- Con `while` se repite un bloque de instrucciones mientras se cumpla una condición determinada
- Es decir, mientras sea `true`

```
while (condicion) {
   //se ejecutan instrucciones aquí
   //si se cumple la condición
}
```



while - reto

¿Puedes hacerlo?

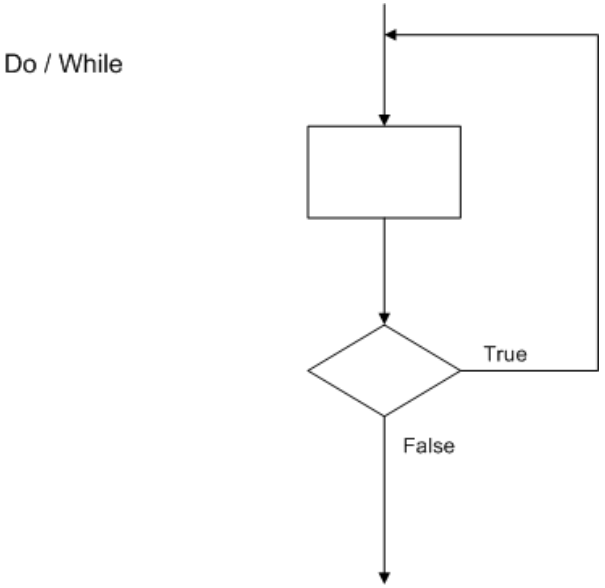
1. Crea un programa en el que el ordenador genere un número aleatorio entre 1 y 10
    1. `random(INICIO_INCLUSIVO, FIN_NOINCLUSIVO)`
2. El programa debe pedir un número al usuario del 1 al 10
3. Mientras el usuario no acierte cual es el número aleatorio:
    1. Se debe mostrar un mensaje de fallo al usuario
    2. Se debe volver a pedir un número al usuario
4. Si el usuario acierta el número aleatorio:
    1. Se debe mostrar un mensaje felicitando al usuario
    2. Se debe finalizar el programa

ARCHIVO RetoWhile.ino

do while

- Con `do while` se repite un bloque de instrucciones mientras se cumpla una condición determinada
- Es decir, mientras sea `true`
- La novedad es que al menos se ejecutan una vez

```
do {
   //se ejecutan instrucciones aquí
   //se siguen repitiendo si se cumple la
siguiente condición
} while (condicion);
```



do while - reto

¿Puedes hacerlo?

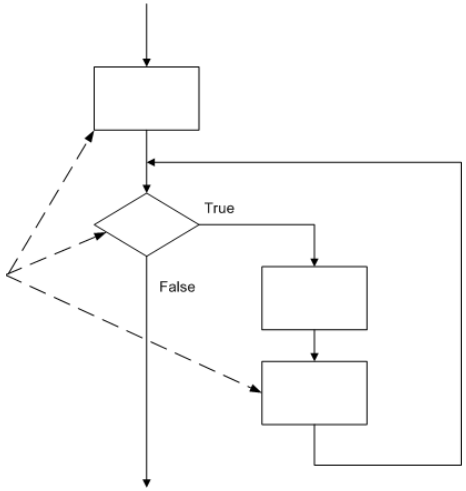
1. Modifica el programa anterior para convertir el `while` en un `do while`

ARCHIVO RetoDoWhile.ino

for

- Con `for` también se repite un bloque de instrucciones mientras se cumpla una condición determinada
- Es decir, mientras sea `true`
- La novedad es al mismo tiempo se puede iniciar y modificar una variable que puede servir para controlar la ejecución

```
for (inicializacion; condicion; modifacionVariable) {
   //se ejecutan instrucciones aquí
   //mientras la condición sea verdadera
}
```



for - ejemplo

- Programa que pide al usuario 5 números y los suma

```
int numero;
int suma;

void setup() {
  Serial.begin(9600);

  for (int i=0; i<5; i++) {
    Serial.println("Introduce un numero");
    while (Serial.available() == 0);
    numero = Serial.parseInt();
    suma = suma + numero;
    //suma += numero;
  }

  Serial.println(suma);
}

void loop() {
}
```


ARCHIVO For.ino

for - reto

¿Puedes hacerlo?

1. Modifica el programa anterior para convertir el `for` en un `while`

ARCHIVO RetoForWhile.ino

Constantes

- Hasta ahora todas las variables que habíamos utilizado podían cambiar de valor
- Para hacer el código más limpio, conviene indicar cuándo una variable nunca va a cambiar su valor
- En esos casos las **variables** pasan a llamarse **constantes**

```
const float NUMERO_PI = 3.14159;
const String CIUDAD = "Oviedo";
const char LETRA = 'A';
const int EDAD = 30;

void setup() {
  Serial.begin(9600);
  Serial.println(NUMERO_PI);
  Serial.println(CIUDAD);
  Serial.println(LETRA);
  Serial.println(EDAD);

}

void loop() {
}
```

ARCHIVO Constantes.ini

Canción "99 bottles de milk" - reto

¿Puedes hacerlo?

1. Crea un programa, con la menor cantidad de instrucciones posible, que muestre la canción en la pantalla del ordenador

99 bottles of milk on the wall, 99 bottles of milk.
Take one down and pass it around, 98 bottles of milk on the wall.

98 bottles of milk on the wall, 98 bottles of milk.
Take one down and pass it around, 97 bottles of milk on the wall.

97 bottles of milk on the wall, 97 bottles of milk.
Take one down and pass it around, 96 bottles of milk on the wall.

…

3 bottles of milk on the wall, 3 bottles of milk.
Take one down and pass it around, 2 bottles of milk on the wall.

2 bottles of milk on the wall, 2 bottles of milk.
Take one down and pass it around, 1 bottle of milk on the wall.

1 bottle of milk on the wall, 1 bottle of milk.
Take one down and pass it around, no more bottles of milk on the wall.

No more bottles of milk on the wall, no more bottles of milk.
Go to the store and buy some more, 99 bottles of milk on the wall.

ARCHIVO RetoCancion.ini

TRABAJO CON CADENAS DE TEXTO

Cadenas de texto

- Existe un tipo de dato denominado `String` para trabajar con cadenas

```
const String aficion = "informatica";
String nombre;
String mensaje;

void setup() {
  Serial.begin(9600);
  Serial.println("Introduce tu nombre");
  while (Serial.available() == 0);
  nombre = Serial.readString();

  mensaje = "Tu nombre es ";
  mensaje += nombre;
  mensaje += " y te gusta la ";
  mensaje += aficion;

  Serial.println(mensaje);
}

void loop() {
}
```


ARCHIVO Cadenas.ini

Cadenas de texto (II)

- Se pueden realizar muchas operaciones con las cadenas invocando diferentes tipos de "funciones"
- En breve sabremos más sobre las funciones

```
const String gijon = "Gijon";
const String oviedo = "Oviedo";
const String mieres = "Mieres";

void setup() {
  Serial.begin(9600);

  Serial.print("Tamaño de la cadena \"gijon\": ");
  Serial.println(gijon.length());


  Serial.print("El primer caracter de la cadena \"oviedo\": ");
  Serial.println(oviedo.charAt(0));

  Serial.print("Una subcadena [2, 5) de la cadena \"mieres\": ");
  Serial.println(mieres.substring(2, 5));

  String juntas = gijon + ", " + oviedo + " y " + mieres;
  Serial.print("Todas las ciudades juntas: ");
  Serial.println(juntas);

  Serial.print("Gijon y Oviedo son iguales: ");
  Serial.println(gijon.equals(oviedo));
  Serial.print("Gijon y Gijon son iguales: ");
  Serial.println(gijon.equals("Gijon"));
}

void loop() {
}
```


ARCHIVO CadenasOperaciones.ini

Operaciones - reto

¿Puedes hacerlo?

1. Crea una cadena de texto (por ejemplo `String gijon = "Gijón";`)

2. Crea un bucle que itere por cada uno de los caracteres de la cadena (habrá que tener en cuenta su tamaño), y que vaya imprimiendo caracter a caracter hasta formar la palabra completa

ARCHIVO RetoCadenasOperaciones.ini

FUNCIONES

¿Qué es una función?

- Es un grupo de instrucciones que realizan una tarea específica
- Se les da un nombre para que se les pueda "llamar" desde otra parte del código
- Una vez que llamamos a la función y ésta se termina, el código sigue en el mismo punto donde había terminado
- **¿Cómo se crean las funciones?**

```
void calculo() {
  Serial.println("Hola!");
  Serial.println("...voy a hacer un calculo");
}
```

¿Cómo se utiliza una función?

- Simplemente se utiliza **su nombre** las veces que queramos

```
void calculo() {
  Serial.println("Hola!");
  Serial.println("...voy a hacer un calculo");
}

void setup() {
  Serial.begin(9600);
  calculo();
  calculo();
  calculo();
  calculo();
}

void loop() {
}
```


ARCHIVO Operacion.ini

Vamos a mejorar esto

- Podemos pasarle valores de cualquier tipo a las funciones

```
void calculo(int numero) {
  Serial.print("Hola!");
  Serial.print("...voy a hacer un calculo con el numero ");
  Serial.println(numero);
}

void setup() {
  Serial.begin(9600);
  calculo(1);
  calculo(2);
  calculo(3);
  calculo(4);
}

void loop() {
}
```

Ya tenemos una operación

- Podemos pasarle valores de cualquier tipo a las funciones

```
void calculo(int numero, int numero2) {
  Serial.print("La suma de los numeros es: ");
  int suma = numero + numero2;
  Serial.println(suma);
}

void setup() {
  Serial.begin(9600);
  calculo(1, 3);
  calculo(3, 2);
  calculo(5, 10);
  calculo(4, 340);
}

void loop() {
}
```


ARCHIVO Operacion3.ini

Pero mejor aún…

- Podemos devolver valores para que se hagan con ellos lo que se desee

```
int resultado3;
int resultado4;

int calculo(int numero, int numero2) {
  int suma = numero + numero2;
  return suma;
}

void setup() {
  Serial.begin(9600);
  Serial.println(calculo(1, 3));
  Serial.println(calculo(3, 2));
  resultado3 = calculo(5, 10);
  resultado4 = calculo(4, 340);
}

void loop() {
}
```

Ejemplo de función

```
float kmh;

float velocidad(float km, float horas) {
  float resultado = 0;
  resultado = km / horas;
  return resultado;
}

void setup() {
  Serial.begin(9600);

  kmh = velocidad(100, 2.5);
  Serial.print("La velocidad del coche es " + String(kmh) + " KH/H");
}

void loop() {
}
```


ARCHIVO Coche.ini

Coche - reto

¿Puedes hacerlo?

1. Reduce el número de instrucciones de la función `velocidad()` a únicamente una instrucción

2. Calcula también la velocidad de un coche que recorre 1 km en 0.02 horas

ARCHIVO RetoCoche.ini

Coche – otro reto

¿Puedes hacerlo?

1. Crea un nueva función llamada `aceleracion()` que, sabiendo la velocidad final y la velocidad inicial del coche en metros por segundo, junto con el tiempo en segundos que ha tardado en llegar a dicha velocidad final, calcule la aceleración del coche

2. Calcula la aceleración y muestra los resultados para este caso:
   1. Vf = 46,1 m/s
   2. Vi = 18,5 m/s
   3. T = 2,47 s

3. Calcula la aceleración y muestra los resultados para este caso:
   1. Vf = 0 m/s
   2. Vi = 22,4 m/s
   3. T = 2,55 s

ARCHIVO RetoCoche2.ini

Coche – otro reto más

¿Puedes hacerlo?

1. Crea un nueva función llamada `modelo()` que únicamente devuelva el modelo del coche como una cadena de texto (por ejemplo, "Seat Leon TDI")

2. Llama a la función y muestra el resultado

ARCHIVO RetoCoche3.ini

Operaciones - reto

¿Puedes hacerlo?

1. Realiza un programa que muestre un menú con dos opciones como las siguientes:

```
1- Elevar un número a otro
2- Multiplicación de dos números y división por el tercero de ellos
Por favor, elige una opción
```

2. Si el usuario elige la opción uno, deberá introducir dos valores:
3. La aplicación deberá utilizar una función para devolver el resultado de elevar un número al otro (realizar la potencia)

```
Introduce un número
2
Introduce otro número
5
2^5=32
```

4. Si el usuario introduce la opción dos, deberá introducir tres números:
5. La aplicación deberá utilizar una función para devolver el resultado de multiplicar el primer número por el segundo y dividir el resultado por el tercero

```
Introduce un número
2
Introduce otro número
3
Introduce otro número más
4
2*3/4=1,500000
```

6. Si el usuario no elige ni la opción 1 ni la opción 2, un mensaje de error deberá aparecer

```
Opción no válida
```


ARCHIVO RetoMenuOperaciones.ini

IMPORTAR CÓDIGO

¿Para qué importar código?

- Muchas veces tenemos que utilizar código de otras personas
- …o nuestro pero creado en otro momento

- Es una de las claves de un buen desarrollador

- Ese código se importa en Arduino a través de unas líneas de código llamadas **cabeceras**

- Por ejemplo `#include <Coche.h>`
- **¿Qué código relacionado esperas que tenga algo llamado "Coche"?**

Cabeceras de código

- ¿Qué es en el ejemplo anterior **coche.h**?
- Es un fichero escrito en C++, abriéndolo podremos encontrarnos cosas como las siguientes
- **Nombre** del tipo
- **Funciones** que podemos utilizar libremente (como las que ya hemos definido)
- **Variables** con información relevante sobre el tipo (como las que ya hemos definido)

```
#ifndef Coche_h
#define Coche_h

#include <Arduino.h>
#include <stdio.h>
#include <string.h>

class Coche
{
  public:
    Coche();
    void moverArriba(int unidades);
    void moverAbajo(int unidades);
    void moverDerecha(int unidades);
    void moverIzquierda(int unidades);
    String ver();
    int x;
    int y;
    int gasolina;
};

#endif
```

¿Cómo se utiliza el código importado?

- Como cualquier otro código
- Únicamente hay que estar seguro de que el código ha sido añadido en Arduino previamente

```
#include <Coche.h>

Coche coche1;

void setup() {
  Serial.begin(9600);

  Serial.println("----------------------");
  coche1.moverArriba(1);
  Serial.println("Gasolina: "+String(coche1.gasolina));
  Serial.println(coche1.ver());

  Serial.println("------------------------------");
  coche1.moverArriba(1);
  Serial.println("Gasolina: "+String(coche1.gasolina));
  Serial.println(coche1.ver());
}

void loop() {

}
```

ARCHIVO ImportarCoche.ini

MÁS RETOS

Factorial de un número

¿Puedes hacerlo?

- La fórmula matemática de los números factoriales es:

$$n! = \begin{cases} 1 & \text{si, } n = 0 \\ (n-1)! \times n & \text{si, } n > 0 \end{cases}$$

1. Crea una "función" que admita como entrada un número mayor o igual que 0 y que retorne su factorial
2. Pruébalo para calcular el factorial de 5

ARCHIVO RetoFactorial.ini

Factorial de un número (II)

¿Puedes hacerlo?

- La fórmula matemática de los números factoriales es:

$$n! = \begin{cases} 1 & \text{si, } n = 0 \\ (n-1)! \times n & \text{si, } n > 0 \end{cases}$$

1. Modifica el programa anterior para que el usuario introduzca por teclado el número para el que desea calcular el factorial
2. Si el usuario introduce un número negativo, el programa seguirá insistiendo hasta que el número introducido por el usuario sea mayor o igual que 0

ARCHIVO RetoFactorial2.ini

Factorial de un número (III)

¿Puedes hacerlo?

1. Haz que todo el código anterior se repita cada segundo

ARCHIVO RetoFactorial3.ini

Cuadrado

¿Puedes hacerlo?

1. Crea una programa que le pida al usuario el tamaño de un cuadrado y que lo dibuje en pantalla con números
2. Un ejemplo para un tamaño de 5 sería el siguiente:

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```


ARCHIVO RetoCuadrado.ini

Cuadrado II

¿Puedes hacerlo?

1. Crea una programa que le pida al usuario el tamaño de un cuadrado y que lo dibuje en pantalla con números
2. Un ejemplo para un tamaño de 5 sería el siguiente:

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```


ARCHIVO RetoCuadrado2.ini

Factores de un número

¿Puedes hacerlo?

- El matemáticas, un factor de un número entero `n` es otro número entero menor que `n` que lo divide exactamente. Es decir, el resto de la división de `n` por el factor tiene que ser 0

1. Crea una "función" que admita como entrada un número mayor que 0 y que imprima todos sus factores
2. El usuario introducirá por teclado el número para el que desea calcular sus factores
3. Si el usuario introduce un número negativo, el programa seguirá insistiendo hasta que el número introducido por el usuario sea mayor que 0

ARCHIVO RetoFactores.ini

Fechas

¿Puedes hacerlo?

1. Pídele al usuario tres números:
    1. Día del mes
    2. Mes
    3. Año
2. Y escribe en pantalla una cadena de texto con la fecha, por ejemplo "**La fecha que has introducido es: 10 de mayo de 2001**"

Triángulo

¿Puedes hacerlo?

1. Crea una programa que le pida al usuario el tamaño de un triángulo y que lo dibuje en pantalla con números
2. Un ejemplo para un tamaño de 5 sería el siguiente:

```
    1
  1 2 3
1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9
```


ARCHIVO RetoTriangulo.ini

Más…

¿Puedes hacerlo?

1. Pedir dos números y decir si son iguales o no
2. Pedir un número e indicar si es positivo o negativo
3. Pedir dos números y decir cuál es mayor
4. Pedir dos números y decir cuál es mayor o si son iguales
5. Pedir dos números y mostrarlos ordenados de mayor a menor
6. Pedir dos números y decir si uno es múltiplo de otro
7. Pedir tres números y mostrarlos ordenados de mayor a menor
8. Pedir los coeficientes de una ecuación de segundo grado y mostrar las soluciones reales. Si no existen habrá que indicarlo
9. Pedir el radio de un círculo y calcular su área
10. Pedir el radio de una circunferencia y calcular su longitud

Más…(II)

¿Puedes hacerlo?

1. Pedir un número entre 0 y 9999 y decir cuántas cifras tiene
2. Pedir un número entre 0 y 9999 y mostrarlo con las cifras al revés
3. Pedir un número entre 0 y 999 y decir si es capicúa
4. Pedir una nota de 0 a 10 y mostrarla de la forma: Insuficiente, Suficiente, Bien, …
5. Pedir el día, mes y año de una fecha e indicar si la fecha es correcta. Suponemos que todos los meses tienen 30 días
6. Pedir el día, mes y año de una fecha e indicar si la fecha es correcta. Sin años bisiestos
7. Pedir el día, mes y año de una fecha correcta y mostrar la fecha del día siguiente. Suponemos que todos los meses tienen 30 días
8. Pedir el día, mes y año de una fecha correcta y mostrar la fecha del día siguiente. Sin años bisiestos
9. Pedir dos fechas y mostrar el número de días que hay de diferencia. Suponemos que todos los meses tienen 30 días

Más…(III)

¿Puedes hacerlo?

1. Pedir una nota numérica entre 0 y 10 y mostrar dicha nota de la forma: cero, uno, dos, tres, …
2. Pedir un número de 0 a 99 y mostrarlo escrito. Por ejemplo, para 56 mostrar: cincuenta y seis
3. Leer un número y mostrar su cuadrado, repetir el proceso hasta que se introduzca un número negativo
4. Leer un número e indicar si es positivo o negativo. El proceso se repetirá hasta que se introduzca un 0
5. Leer números hasta que se introduzca un 0. Para cada uno indicar si es par o impar
6. Pedir números hasta que se teclee uno negativo, y mostrar cuántos números se han introducido
7. Realizar un juego para adivinar un número. Para ello pedir un número N, y luego ir pidiendo números indicando "mayor" o "menor" según sea mayor o menor con respecto a N. El proceso termina cuando el usuario acierta

Más…(IV)

¿Puedes hacerlo?

1. Pedir números hasta que se teclee un 0, mostrar la suma de todos los números introducidos
2. Pedir números hasta que se introduzca uno negativo, y calcular la media
3. Pedir un número N, y mostrar todos los números del 1 al N
4. Escribir todos los números del 100 al 0 de 7 en 7
5. Pedir 15 números y escribir la suma total
6. Diseñar un programa que muestre el producto de los 10 primeros números impares
7. Pedir 10 números. Mostrar la media de los números positivos, la media de los números negativos y la cantidad de ceros
8. Pedir 10 sueldos. Mostrar su suma y cuantos hay mayores de 1000€

Más…(V)

¿Puedes hacerlo?

1. Dadas las edades y alturas de 5 alumnos, mostrar la edad y la estatura media, la cantidad de alumnos mayores de 18 años, y la cantidad de alumnos que miden más de 1.75
2. Pide un número (que debe estar entre 0 y 10) y mostrar la tabla de multiplicar de dicho número
3. Una empresa que se dedica a la venta de desinfectantes necesita un programa para gestionar las facturas. En cada factura figura: el código del artículo, la cantidad vendida en litros y el precio por litro
4. Se pide de 5 facturas introducidas: Facturación total, cantidad en litros vendidos del artículo 1 y cuantas facturas se emitieron de más de 600 €
5. Dadas 6 notas, escribir la cantidad de alumnos aprobados, condicionados (=4) y suspensos
6. Pedir un número N, introducir N sueldos, y mostrar el sueldo máximo
7. Pedir 10 números, y mostrar al final si se ha introducido alguno negativo

Más…(VI)

¿Puedes hacerlo?

1. Pedir 5 números e indicar si alguno es múltiplo de 3
2. Diseña una aplicación que muestre las tablas de multiplicar del 1 al 10
3. Necesitamos mostrar un contador con 5 dígitos (X-X-X-X-X), que muestre los números del 0-0-0-0-0 al 9-9-9-9-9, con la particularidad que cada vez que aparezca un 3 lo sustituya por una E
4. Realizar un programa que nos pida un número n, y nos diga cuantos números hay entre 1 y n que son primos
5. Realizar una función, a la que se le pase como parámetro un número N, y muestre por pantalla N veces, el mensaje: "Módulo ejecutándose"
6. Diseñar una función que tenga como parámetros dos números, y que calcule el máximo
7. Diseñar una función que tenga como parámetros tres números, y que calcule el máximo
8. Función a la que se le pasan dos enteros y muestra todos los números comprendidos entre ellos, inclusive

Más…(VII)

¿Puedes hacerlo?

1. Función que muestra en pantalla el doble del valor que se le pasa como parámetro
2. Realizar una función que calcule (muestre en pantalla) el área o el volumen de un cilindro, según se especifique. Para distinguir un caso de otro se le pasará el carácter 'a' (para área) o 'v' (para el volumen). Además hemos de pasarle a la función el radio y la altura
3. Escribir una función que calcule el máximo común divisor de dos números
4. Escribir una función que calcule el mínimo común múltiplo de dos números