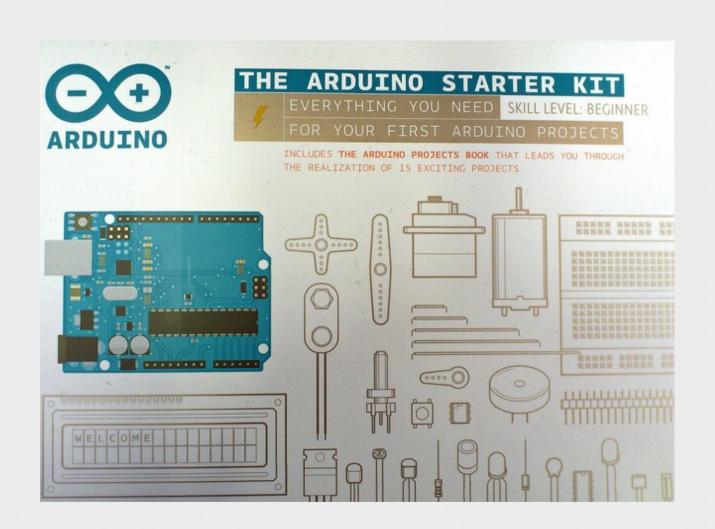
CODEWEEK

MIS PRIMEROS PASOS CON ARDUINO

¿Que sabes de Arduino?

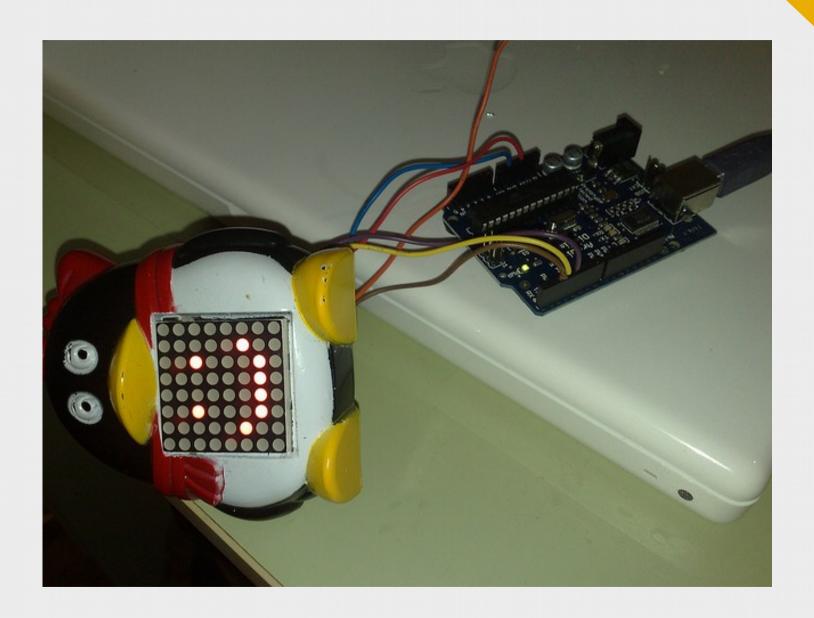
- ¿Que es Arduino?
- ¿Para que sirve?
- ¿Que es la robótica?
- ¿Para que se usa?

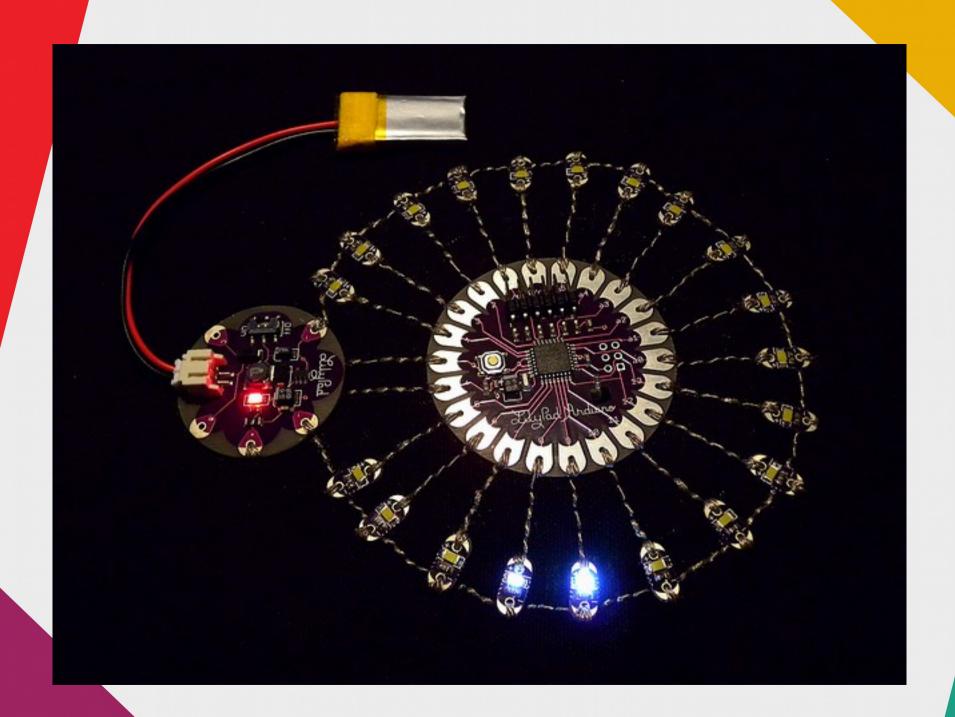


¿Que podemos hacer con Arduino?





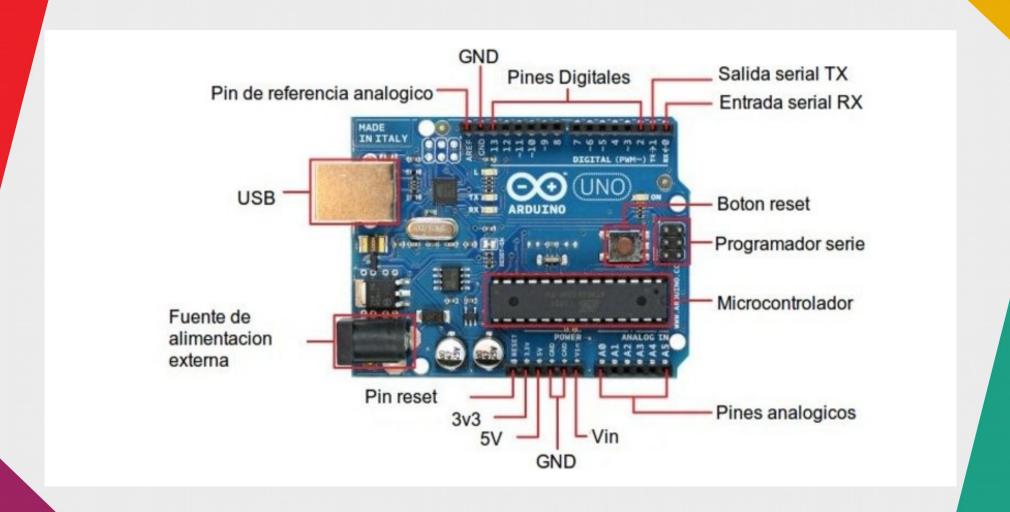




Algunos datos

- Maximo Banzi 2005
- Un bar le da nombre
- Color azul (No Placas Convencionales)
- Lider en DIY (Do It Yourself)
- Plug and Play
- Precio objetivo 30\$
- 100% free source





Familia Arduino







Arduino Mega 2560



Arduino FIO



Arduino PRO



Arduino Mega ADK

Familia Arduino



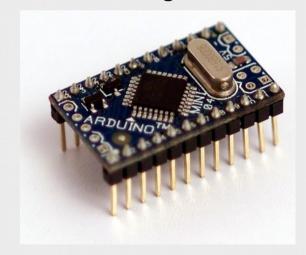
Arduino Ethernet



Arduino Nano



Arduino Mega 2560



Arduino Mega ADK

Conceptos Básicos de Electrónica

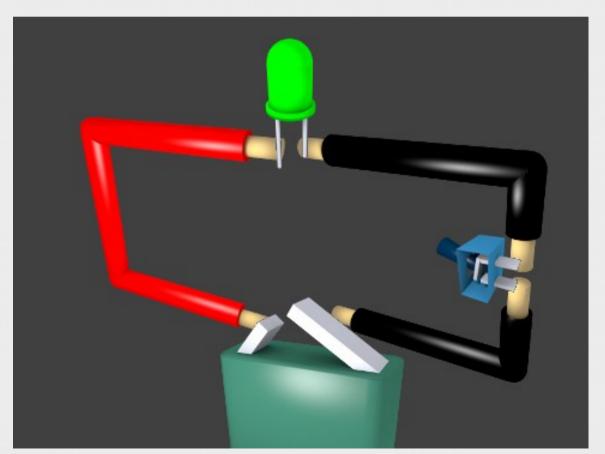
Alimentación y tierra

Tipos de señales

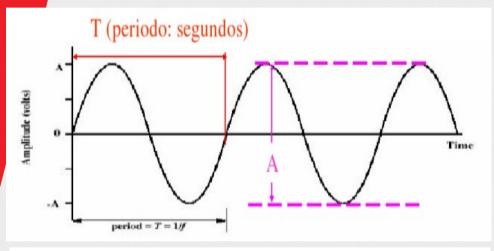
Circuitos Básicos

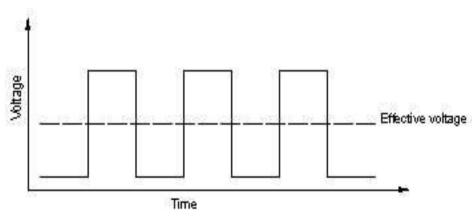
Alimentación y Tierra

- Los circuitos tienen dos conexiones (+) y (-)
- Ningún circuito puede tener una única conexión
- Los circuitos tiene que estar alimentados para funcionar



Tipos de señales





Señales analógicas:

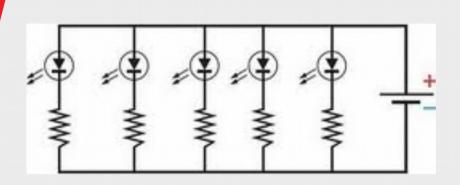
 Pueden tener cualquier valor dentro de un rango.

En Arduino entre 0 y 5 v

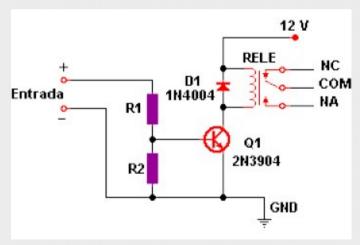
Señales digitales:
 Tienen dos estados:

High (H) entre 4,5 y 5,3 v Low (L) entre -0,3 y 0,5v

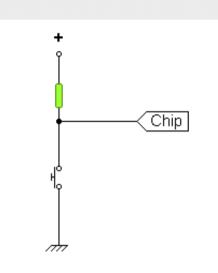
Circuitos básicos



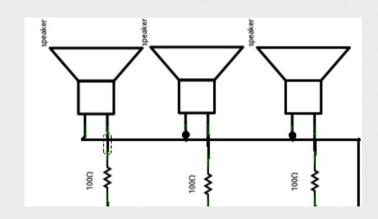
Conexión de un LED



Conexión de un relé



Conexión de un botón



Conexión de un altavoz

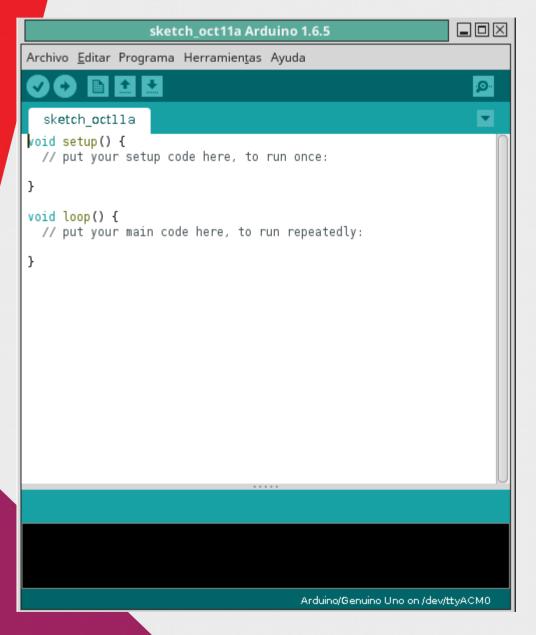
TOMA DE CONTACTO

- Instalación del Software
- Entorno de desarrollo
- Configuración del puerto Usb
- Compilación

Instalación del software

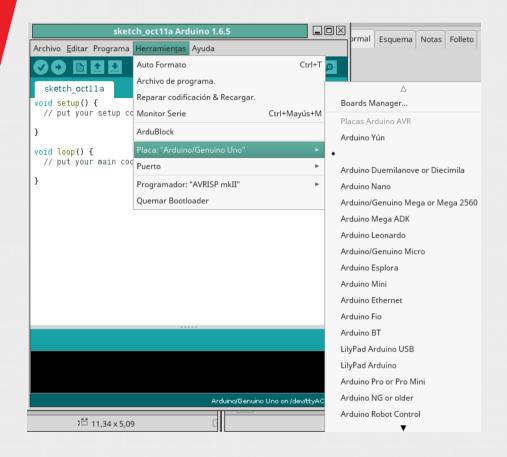
- Descargamos la última versión de: http://www.arduino.cc/es
- Descargar e instalar Java
- El software consta de dos partes_
 - Entorno de desarrollo
 - Puerto virtual FTDI (Serie)

Entorno de desarrollo



- Apariencia muy sencilla
- 4 Secciones básicas.
- En Archivos tenemos: Proyectos de demostración.
- En Programa tenemos: Opciones de compilación.
- En Herramientas tenemos:
 - Configuración de la plataforma
 - Configuración del puerto

Entorno de desarrollo



- Seleccionar tarjeta correcta:
- Tools >> Arduino Uno



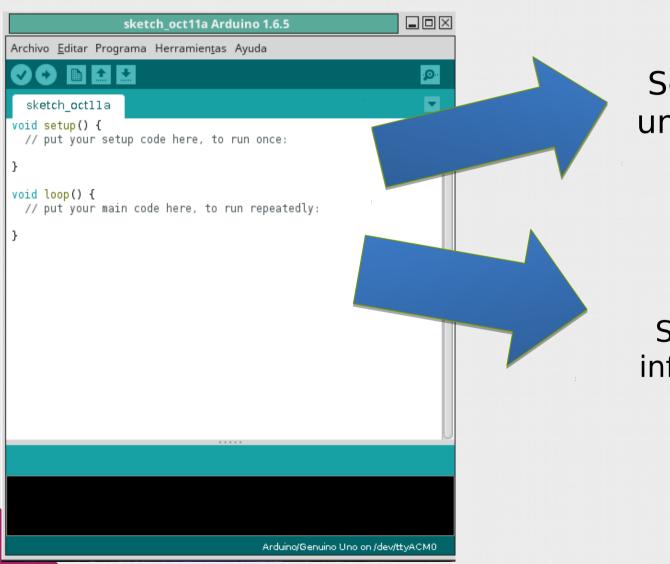
Seleccionar puerto serie:

- (Virtual)
- Depende del sistema operativo.
 - Serial port >> tty.xxx
 - Serial port >> COMx

Si todo está correcto el led I/O 13 parpadea

BLOQUES BASICOS INSTRUCCIONES BASICAS ESTRUCTURAS DE CONTROL PRIMER PROGRAMA

Bloques básicos de código



Se ejecuta una sola vez

Se ejecuta infinitamente

Instrucciones básicas



pinMode(nº Pin, Configuración) nº Pin: entre 0 y 13. Configuración: INPUT o OUTPUT

digitalWrite(nº Pin, Estado)
 nº Pin: entre 0 y 13.
 Estado: HIGH o LOW

■ digitalRead(nº Pin) nº Pin: entre 0 y 13.

analogRead(nº Pin); nº Pin: entre 0 y 5.

Instrucciones básicas



- Serial.begin(Velocidad)
 Velocidad: típicamente 9600.
- Serial.println(Variable)
 - Variable: Dato a mandar
- Serial. Read()
- Delay(ms)
 - ms: Tiempo de espera en milisegundos.
- Todas las funciones en: http://arduino.cc/es/Reference/Home ePage

Programación: funciones básicas

Funciones E/S digitales:

pinMode(pin, modo);

digitalWrite(pin, estado);

Val=digitalRead(pin);

Funciones E/S analógicas:

analogWrite(pin, valor);

Val=analogRead(pin);

val=map(val,ini,fin,ini2,fin2);

Funciones puerto serie:

Val= Serial.read();

Serial.print(a)/Serial.println(a);

Serial.write(a);

Serial.begin(baudios);

Val=Serial.available();

Función espera:

delay(ms)

Inicialización

Activación de una salida

Lectura de una entrada

Salida en PWM

Entrada de PWM

Mapear valor

Recepción serie

Imprimir en pantalla

Envío de un dato

Inicialización serie

Número de bytes en puerto

Retraso en milisegundos



NUESTRO PRIMER PROGRAMA



ENCENDER PRIMER LED

```
Blink Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
                                                                  ø.
  Blink §
void setup() {
  pinMode(13, OUTPUT);
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
                                        Arduino/Genuino Uno on /dev/ttyACM0
```

Shields Arduino



Motores













GPS

Familia Arduino



Display 7 segmentos



Display LCD Alfanum.



Display LCD Matricial







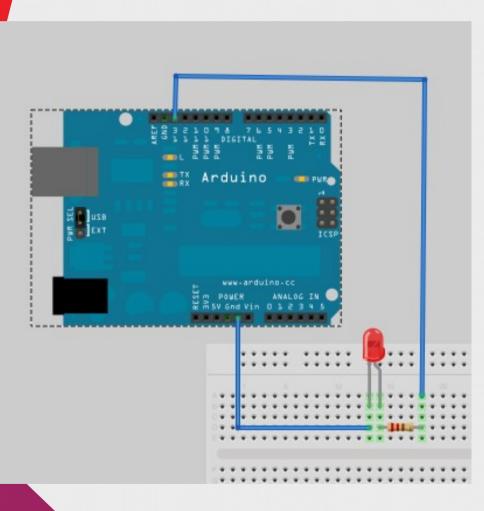


¿Para que sirve Fritzing?



- Software de apoyo para el diseño de circuitos con Arduino
- Dispone de componentes comunes:
 - Resistencias,
 - Leds,
 - Alimentaciones,
 - Motores,
 - Sensores,
 - **-** ...

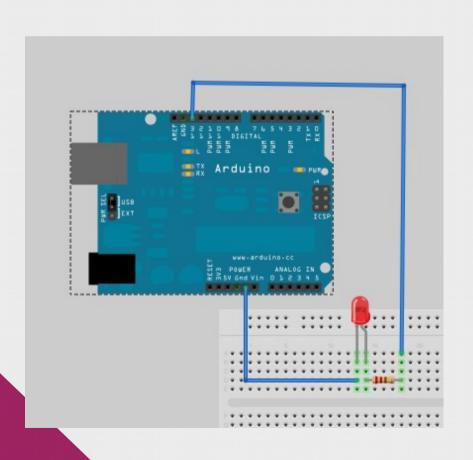
Diseño con protoboard

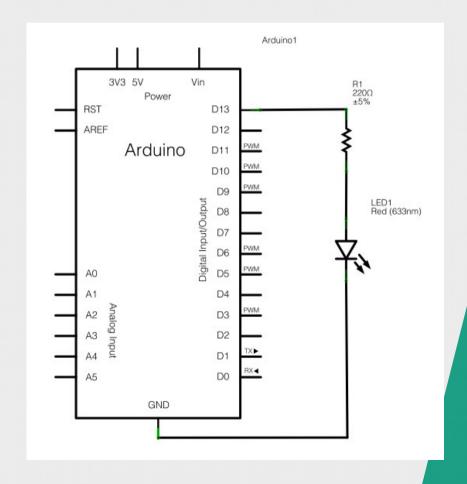


- Dispone de una amplia librería de proyectos
- Se pueden añadir componentes nuevos.
- Representa gráficamente las conexiones en la protoboard.
 - Los montajes son fácilmente entendibles

Diseño con esquemático

Todos los cambios del diseño con protoboard se importan automáticamente y viceversa.





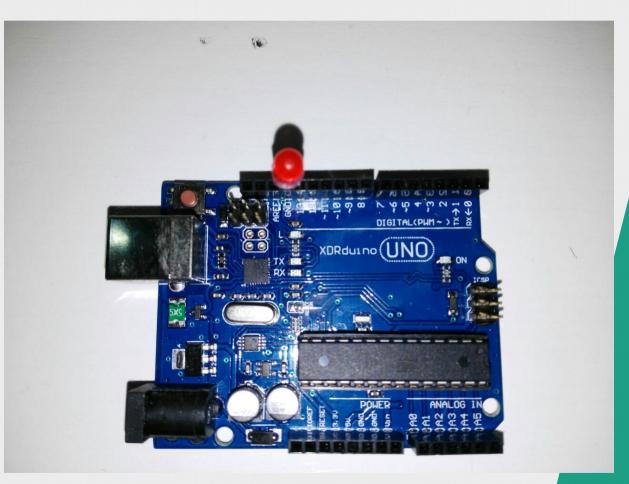
Segundo Ejercicio



Montaje y encendido de primer led

Necesitamos:

- Led
- Placa Arduino



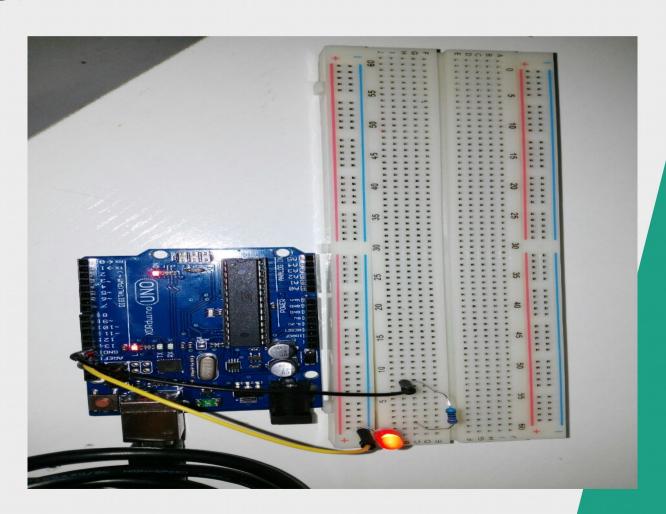
Tercer Ejercicio



Montaje y encendido de primer led

Necesitamos:

- Led
- Protoboard
- Resistencia
- Cables
- Placa Arduino





Semáforo

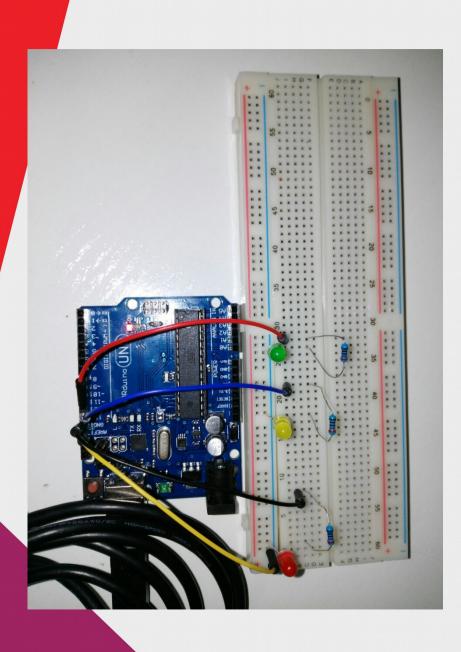
- Led (Amarillo, Rojo, Verde)
- Protoboard
- Resistencias
- Cables
- Placa Arduino



SOLUCIÓN



Solución



```
void setup() {
 pinMode(13, OUTPUT);
 pinMode(12, OUTPUT);
 pinMode(11, OUTPUT);
void loop() {
 digitalWrite(13, HIGH);
 delay(1000);
 digitalWrite(13, LOW);
 delay(1000);
 digitalWrite(12, HIGH);
 delay(1000);
 digitalWrite(12, LOW);
 delay(1000);
 digitalWrite(11, HIGH);
 delay(1000);
 digitalWrite(11, LOW);
 delay(1000);
```

Semáforo Viene y Va

- Led (Amarillo, Rojo, Verde)
- Protoboard
- Resistencias
- Cables
- Placa Arduino



```
if (condicion) {
....
} else {
....
}
```

Ejecuta el bloque separado por {} siempre y cuando se cumpla la condición, acompañada a esta estructura existe else (sinó) que es para el caso contrario a la condición, sin tener que evaluar la condición negada.

for (inicial;condición;repetición) { }

Se ejecuta la sentencia inicial, y si se cumple la condición se ejecuta el código en el interior de la llaves, una vez que se termina se ejecuta la repetición y se vuelve a comparar la condición, mientras la condición sea cierta el bloque encerrado entre llaves se seguirá ejecutando.

while (condición){

Mientras se cumpla la condición entre paréntesis, se ejecuta el código en el interior de {} de manera infinita hasta.

```
switch (varieble numérica) {
case num1:
break;
case num2:
break;
default:
```

```
switch ($i)
  case 0:
     echo "i es igual a 0";
  case 1:
     echo "i es igual a 1";
  case 2:
     echo "i es igual a 2";
```

La sentencia switch es un tanto compleja y probabilísticamente poco usada ya que no puede usarse con objetos y las comparaciones son primitivas, en algunos trozos o segmentos de código es crucial para optimizar la velocidad de nuestro programa, de acuerdo al contenido de la variable numérica se ejecuta la sentencia case que coincida, cada bloque de código debe terminar con break, ya que en caso contrario se seguirá ejecutando la siguiente cláusula case, existe una opción default que es para cuando ninguno de los casos resulta ser el correcto.

```
do{
...
}while(condición);
```

El bloque dentro de la palabra clave do, se ejecutará al menos una vez y si la condición de while se cumple, se repetirá mientras sea cierta.

Arrays

Una matriz es un conjunto de variables que se accede con un número de índice

```
//Ejercicio con un array y con un contador para recorrer el array//
int leds[3]={11,12,13};
int tiempo=100;
int led=0:
void setup()
for(led=0;led <= 3;led++)
 { pinMode (leds[led],OUTPUT); }
void loop()
{ for(led=0;led <= 3;led++)
  digitalWrite(leds[led],HIGH);
  delay(tiempo);
  digitalWrite(leds[led],LOW);
  delay(tiempo);
for(int led=3;led>=0;led--)
{ digitalWrite(leds[led],HIGH);
     delay(tiempo);
     digitalWrite(leds[led],LOW);
     delay(tiempo);
```

```
int tiempo=10;
int led;
void setup()
 for (led=8;led<=12;led++)
  pinMode(led,OUTPUT);
void loop()
 for (led=8;led<=12;led++)
 digitalWrite(led,HIGH);
 delay(tiempo);
 digitalWrite(led,LOW);
 delay(tiempo);
```

