Taller Introducción a la Ingeniería Eléctrica
Microcontrolador Arduino y Comunicaciones

Instituto de Ingeniería Eléctrica

Taller 1: Introducción a la programación y a Arduino

13 de Agosto de 2024

Esquema de la presentación

1. Repasamos lo que quedó de deberes
2. Materiales:
   - Protoboard
   - Resistencias
   - LEDs
     - Ejercicio 1 - Clase
3. Programación:
   - Conceptos básicos:
     - Variables y tipos de datos
     - Estructura de un código
   - Diagramas de flujo
     - Ejercicio 2 - Clase
   - Operadores
     - Comparación
     - Lógicos
     - Ejercicio 3 - Clase
   - Control de flujo
     - Sentencia “if”
   - Manejo de pines digitales
     - Ejercicio 4 - Clase
4. Conocemos más materiales
   - Pulsadores e interruptores
5. Ejercicios de deberes para la próxima clase
   - Ejercicio 1
   - Ejercicio 2
   - Ejercicio 3

Esquema de la presentación

1. Repasamos lo que quedó de deberes
2. Materiales:
   - Protoboard
   - Resistencias
   - LEDs
     - Ejercicio 1 - Clase
3. Programación:
   - Conceptos básicos:
     - Variables y tipos de datos
     - Estructura de un código
   - Diagramas de flujo
     - Ejercicio 2 - Clase
   - Operadores
     - Comparación
     - Lógicos
     - Ejercicio 3 - Clase
   - Control de flujo
     - Sentencia "if"
   - Manejo de pines digitales
     - Ejercicio 4 - Clase
4. Conocemos más materiales
   - Pulsadores e interruptores
5. Ejercicios de deberes para la próxima clase
   - Ejercicio 1
   - Ejercicio 2
   - Ejercicio 3

Repaso

¿Qué tal les fue con los ejemplos y el uso del simulador?

¿Qué vieron en común entre los diferentes códigos?

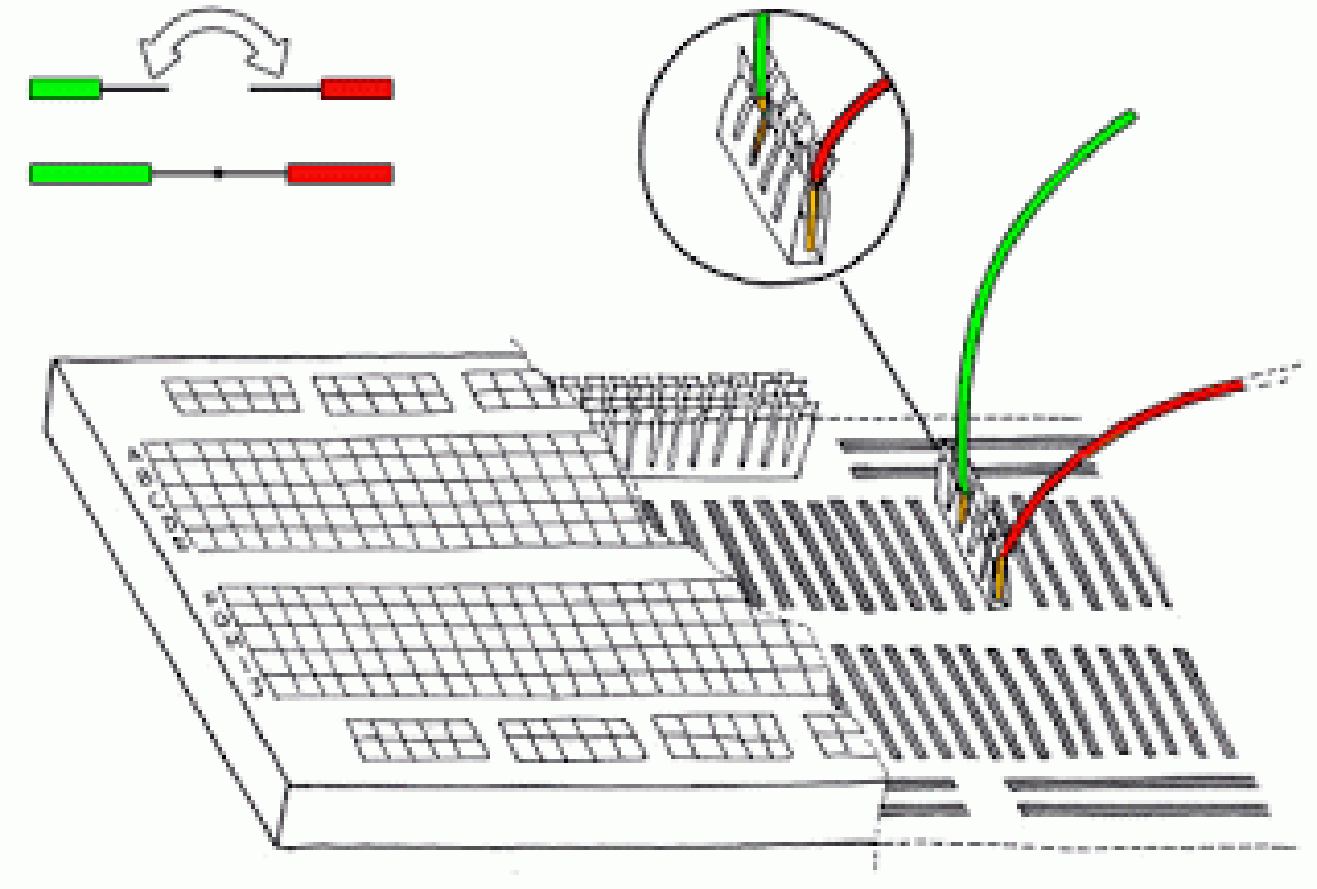
¿Pudieron realizar el último ejercicio planteado?
Hoy lo vamos a armar con materiales reales!

Esquema de la presentación

1. Repasamos lo que quedó de deberes
2. Materiales:
   - Protoboard
   - Resistencias
   - LEDs
     - Ejercicio 1 - Clase
3. Programación:
   - Conceptos básicos:
     - Variables y tipos de datos
     - Estructura de un código
   - Diagramas de flujo
     - Ejercicio 2 - Clase
   - Operadores
     - Comparación
     - Lógicos
     - Ejercicio 3 - Clase
   - Control de flujo
     - Sentencia "if"
   - Manejo de pines digitales
     - Ejercicio 4 - Clase
4. Conocemos más materiales
   - Pulsadores e interruptores
5. Ejercicios de deberes para la próxima clase
   - Ejercicio 1
   - Ejercicio 2
   - Ejercicio 3

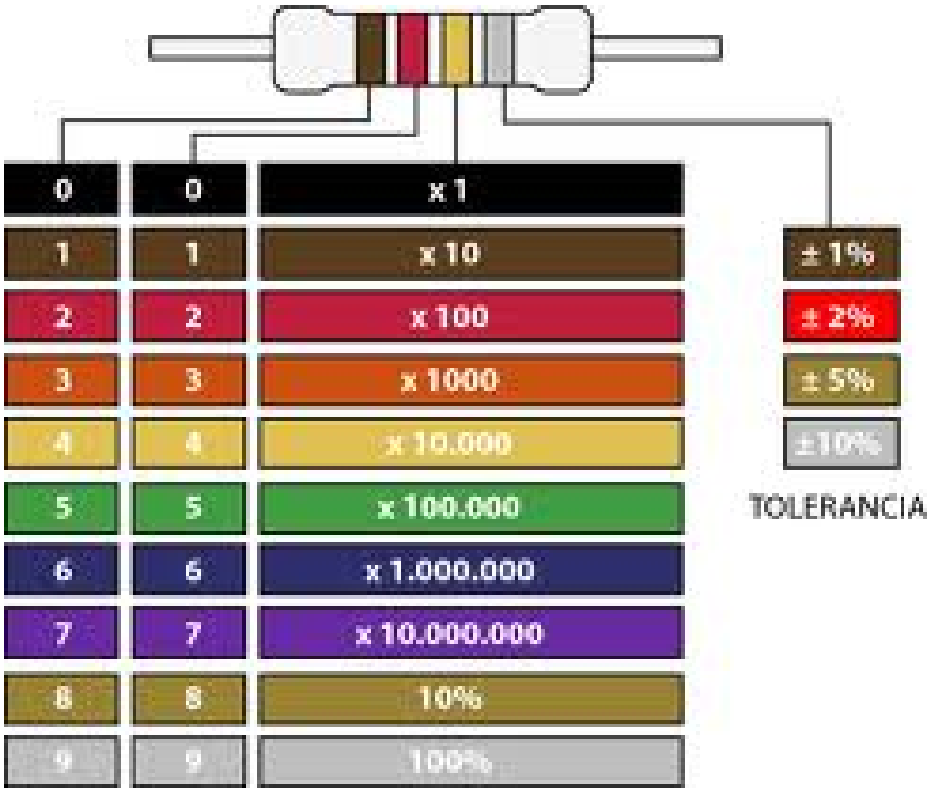
¿Cómo están conectados los bornes?

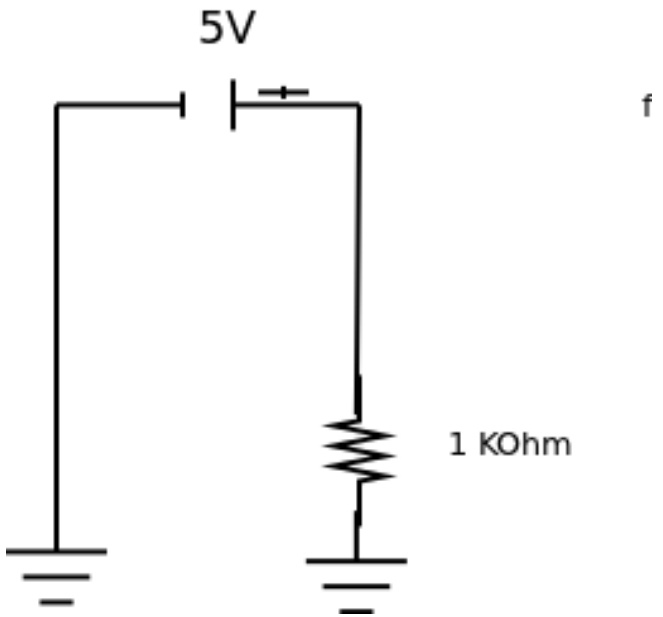
Figura: http://taller.tagabot.org/index.php/Arduino/Armada



Resistencias - Código de Colores. Corriente?

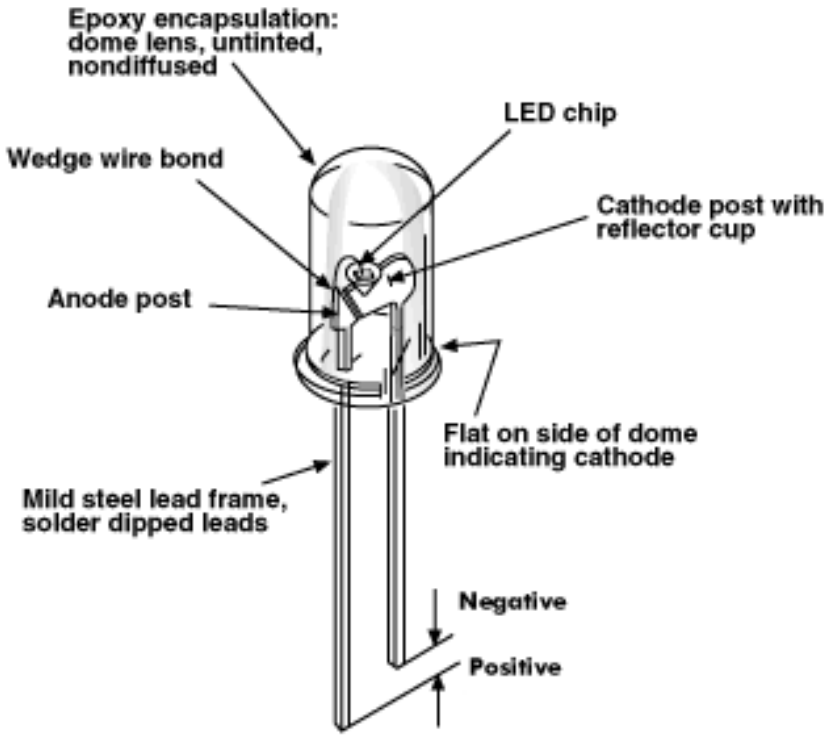
Código colores: tecnorik.blogspot.com

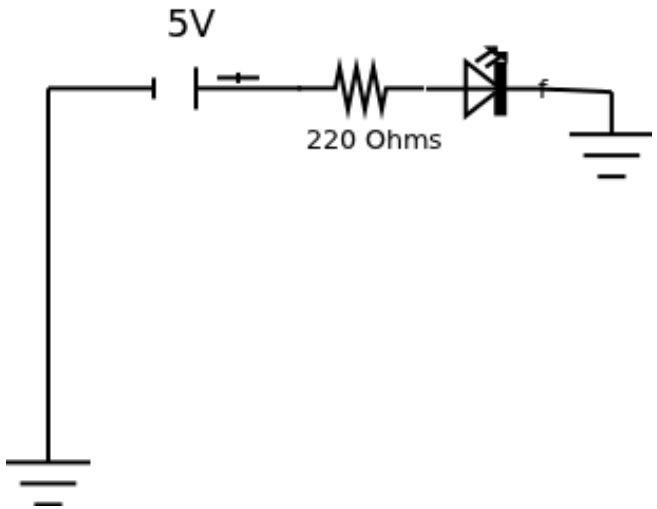




LEDs. Corriente?

Descripción Led: electroschematics.com





Ejercicio 1 - Clase

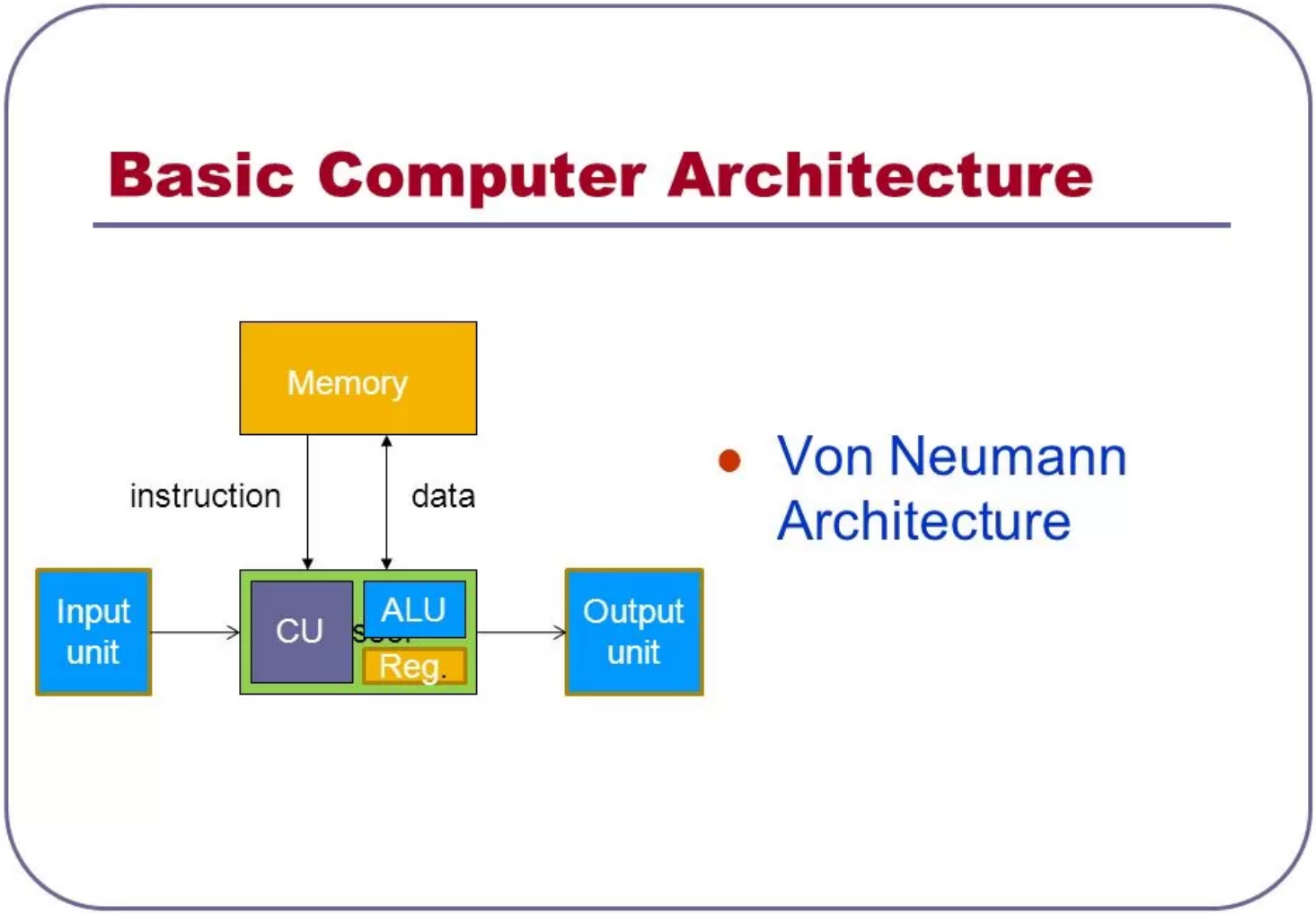
Trabajando con materiales reales

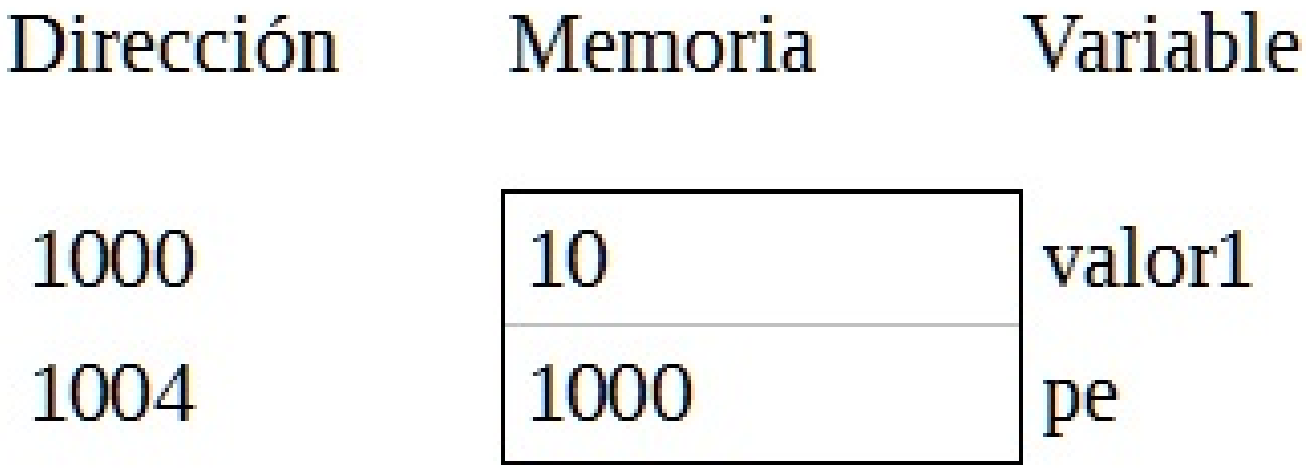
Armar con materiales reales el último ejercicio de deberes que hicieron funcionar en el simulador.
El programa lo van a cargar a través de IDE de Arduino (repasar pasos en presentación del Taller 0).

Esquema de la presentación

1. Repasamos lo que quedó de deberes
2. Materiales:
   - Protoboard
   - Resistencias
   - LEDs
     - Ejercicio 1 - Clase
3. Programación:
   - Conceptos básicos:
     - Variables y tipos de datos
     - Estructura de un código
   - Diagramas de flujo
     - Ejercicio 2 - Clase
   - Operadores
     - Comparación
     - Lógicos
     - Ejercicio 3 - Clase
   - Control de flujo
     - Sentencia "if"
   - Manejo de pines digitales
     - Ejercicio 4 - Clase
4. Conocemos más materiales
   - Pulsadores e interruptores
5. Ejercicios de deberes para la próxima clase
   - Ejercicio 1
   - Ejercicio 2
   - Ejercicio 3

Qué es una variable??


Basic Computer Architecture
Memory
instruction
data
Input unit
CU
ALU
Reg.
Output unit
Von Neumann Architecture


Dirección
Memoria
Variable
1000
10
valor1
1004
1000
pe

Variables

Definición, declaración e inicialización

```
int nombreVariable1 = 0;

float nombreVariable2 = 1.24;

int nombreArray[] = {valor0, valor1, valor2, ...};

int nombreArray2[5];
```

Variables

Tipos de Datos

- byte: Enteros (1 byte): 0 a 255

```
byte minutos = 56;
```

- int: Enteros (2 bytes): -32.768 a 32767

```
int estudiantesFING = 7569;
```

- long: Enteros (4 bytes)

```
long estudiantesUDELAR = 135757;
```

- float, double: Números en punto flotante (4 bytes)

```
float temperatura = 37.6;
```

- boolean: Verdadero (**TRUE**) o Falso (**FALSE**)

```
boolean pulsado = true;
```

- char: Un solo caracter

```
char operacion = '+';
```

- String: Listas de caracteres

```
String mensajeBienvenida = "Ingrese sus
    credenciales para continuar";
```

Variables

Definición, declaración e inicialización

Tipo	Tamaño	Rango
boolean	1 byte	0 a 1
char	1 byte	-128 a 127
unsigned char	1 byte	0 a 255
int	2 bytes	-32.768 a 32.767
unsigned int	2 bytes	0 a 65535
word	2 bytes	0 a 65535
long	4 bytes	-2.147.483.648 a 2.147483.647
unsigned long	4 bytes	0 a 4.294.967.295
float	4 bytes	3,4028235E-38 a 3,4028235E+38
double	4 bytes	3,4028235E-38 a 3,4028235E+38

Estructura básica de un código

```
declaraciOn e inicializaciOn de constantes;

declaraciOn e inicializaciOn de variables;

void setup()
{
        inicializaciOn de pines
        inicializaciOn de la comunicaciOn serial
        lo que se ejecuta una sola vez
}

void loop()
{
        lo que se va a hacer todo el tiempo
}
```

Estructura avanzada de un código

```
declaraciOn e inicializaciOn de constantes;

declaraciOn e inicializaciOn de variables;

encabezado de funciones propias implementadas

void setup()
{
        seteo de pines
        inicializaciOn de la comunicaciOn serial
        lo que se ejecuta una sola vez
}

void loop()
{
        lo que se va a hacer todo el tiempo
}

encabezado e implementaciOn de funciones propias
```

¡No olvidar “;” al final de cada sentencia!

Comentarios

Herramienta fundamental al escribir un código, ya que permite tener presente siempre qué hace cada línea o bloque, permite que otra persona lo pueda entender, etc.

```
/* Este es un
bloque de
comentarios
*/

// En cambio, este es un comentario de linea
```

Diagramas de flujo

¿Cómo programar?

Es una representación gráfica (bloques) de un algoritmo (¿Qué es un algoritmo?):

- Inicio/Fin

- Dirección de flujo

- Procesos/Tareas

- Decisión

Se quiere abrir una puerta siempre que se pulse un botón. Luego, la puerta debe cerrarse automáticamente cinco segundos después.

¿Diagrama de flujo?

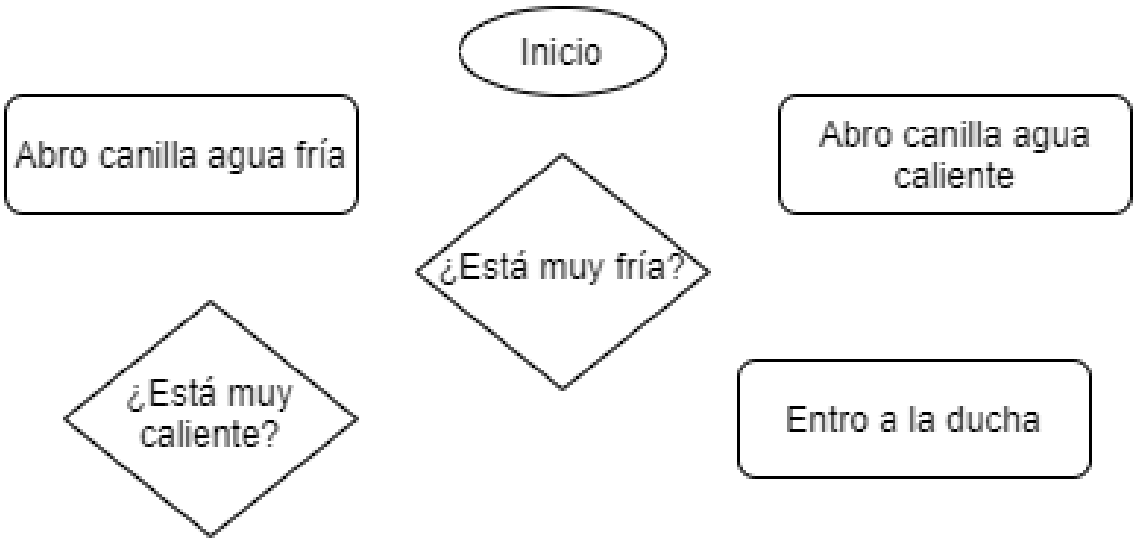
Diagrama de flujo

Ejemplo

Otra forma (en funcionamiento continuo):

Objetivo: Muy simplificadamente representar la situación de "regular el agua de la ducha" previo a ingresar a bañarse.

Tratar de usar una y solo una vez las siguientes formas.



Operadores de comparación

- `A == B` : A igual a B
- `A != B` : A distinto de B
- `A < B` : A menor que B
- `A <= B` : A menor o igual a B
- `A > B` : A mayor que B
- `A >= B` : A mayor o igual a B

Operadores lógicos

- `A && B` : A **AND** B
- `A || B` : A **OR** B
- `!A` : **NOT** A

Ejercicio 3 - Clase

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;

  d = (a || b);

  d = (b && c);

  d = (a != c) && b;

  d = ( a == !(b && c) ) || ( b == ( !a && c ) );
```

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true}); \Rightarrow \text{true}$
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true}); \Rightarrow \text{true}$
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true}; \Rightarrow \text{true}$
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true})); \Rightarrow \text{true}$

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true};$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true}));$ $\Rightarrow$ true

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $d = (\underbrace{a}_{false} || \underbrace{b}_{true});$ $\Rightarrow$ true
- $d = (\underbrace{b}_{true} \&\& \underbrace{c}_{true});$ $\Rightarrow$ true
- $d = (\underbrace{a}_{false} != \underbrace{c}_{true}) \&\& \underbrace{b}_{true};$ $\Rightarrow$ true
- $d = (\underbrace{a}_{false} == !(\underbrace{b}_{true} \&\& \underbrace{c}_{true})) || (\underbrace{b}_{true} == (!\underbrace{a}_{false} \&\& \underbrace{c}_{true}));$ $\Rightarrow$ true

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true};$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true}));$ $\Rightarrow$ true

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true};$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true}));$ $\Rightarrow$ true

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true};$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true}));$ $\Rightarrow$ true

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true};$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true}));$ $\Rightarrow$ true

Ejercicio 3 - Clase (solución)

Exprese el valor asignado a la variable **d** para cada una de las operaciones. Para cada operación tomar los valores de las variables tal cual están en la incialización.

```
boolean a=false;
boolean b=true;
boolean c=true;
boolean d;
```

- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} || \underbrace{\mathrm{b}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true});$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} != \underbrace{\mathrm{c}}_{true}) \&\& \underbrace{\mathrm{b}}_{true};$ $\Rightarrow$ true
- $\mathrm{d} = (\underbrace{\mathrm{a}}_{false} == !(\underbrace{\mathrm{b}}_{true} \&\& \underbrace{\mathrm{c}}_{true})) || (\underbrace{\mathrm{b}}_{true} == (!\underbrace{\mathrm{a}}_{false} \&\& \underbrace{\mathrm{c}}_{true}));$ $\Rightarrow$ true

Control de flujo

Sentencia “if”

```
if (condicion) {
  hacer algo;
}
```

Control de flujo

Sentencia "if"

```
if (condicion) {
  hacer algo;
}

if (condicion) {
  hacer algo;
} else {
  hacer otra cosa;
}
```

Control de flujo

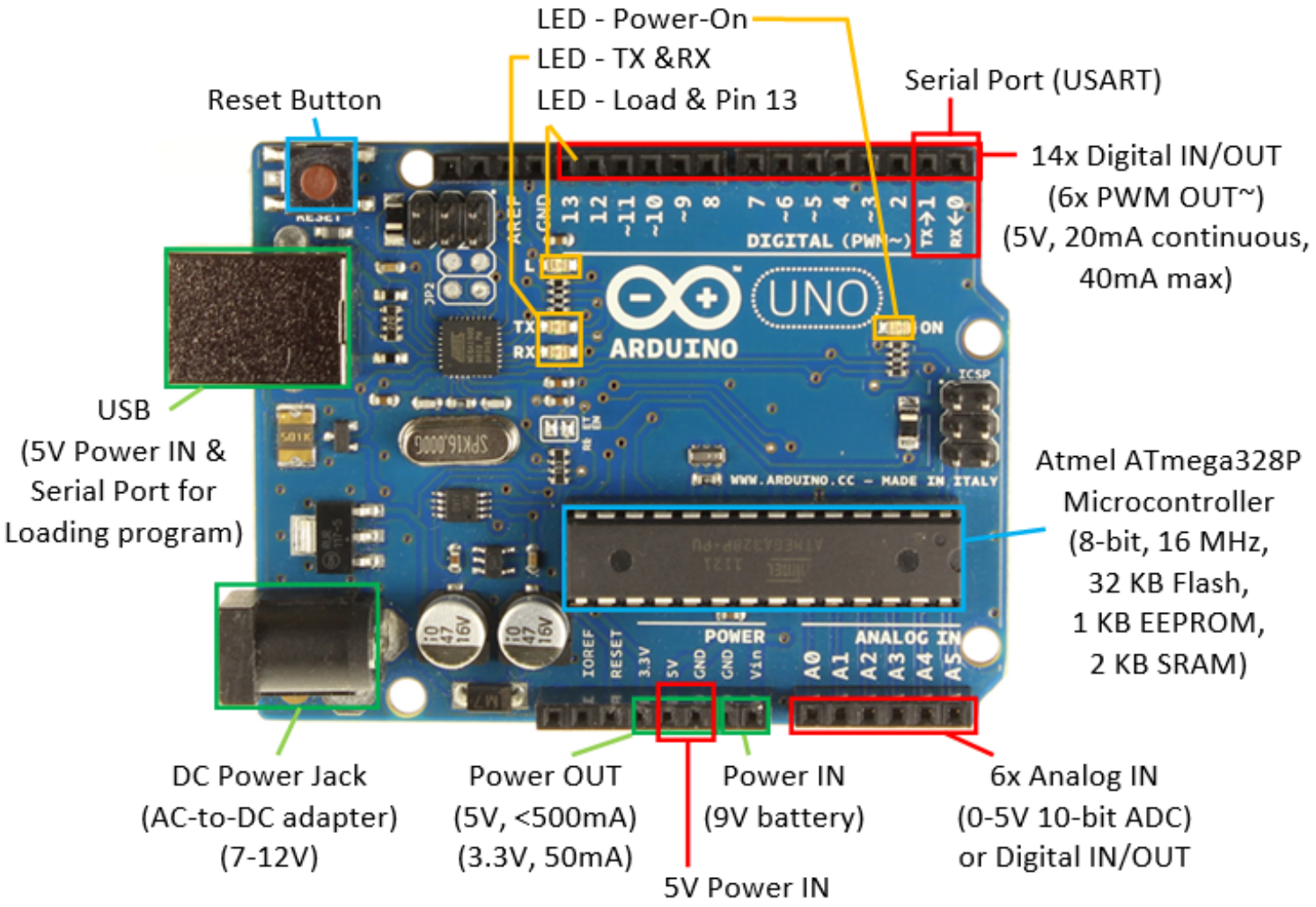
Sentencia "if"

```
if (condicion) {
  hacer algo;
}

if (condicion) {
  hacer algo;
} else {
  hacer otra cosa;
}

 // Ejemplo:
 if (a == 0){
   b=1;
 }
 else {
   b=0;
 }
```

Manejo de pines
DIGITALES



Manejo de pines
DIGITALES

Los pines digitales (pin 0 a 13) pueden tomar **sólo 2 valores**: 0V (LOW) o 5V (HIGH)[1].

```
int pinEntrada = 10;
int pinSalida = 13;
int llave = 0;
void setup(){
  pinMode(pinEntrada, INPUT);
  pinMode(pinSalida, OUTPUT);
}

void loop(){
  llave = digitalRead(pinEntrada);//lee el valor del pin 10
  if(llave == LOW){
        digitalWrite(pinSalida, HIGH);//pone el pin 13 en 5V
  }else{
        digitalWrite(pinSalida, LOW); //pone el pin 13 en 0V
  }
}
```

[1] Excepto los pines con PWM

Ejercicio 4 - Clase

Utilizando el simulador o los materiales reales y partiendo del proyecto del Ejercicio de deberes del Taller 0 (*el de prender y apagar un led*):

Se pide:
- Escribir un programa que prenda y apague el LED cada un segundo si la entrada 5 está a 5V y la entrada 6 está a 5V. Y que lo mantenga prendido permanentemente en cualquier otro caso.
- Probar el comportamiento en los distintos casos para confirmar el funcionamiento deseado.

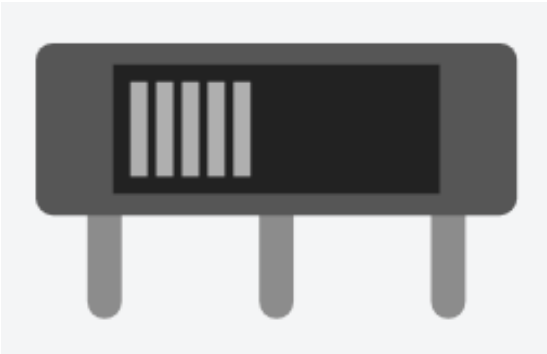
Nota: para poder fijar las tensiones a los pines 5 y 6 utilizar cables (desde los pines de 5V y GND presentes en la placa).

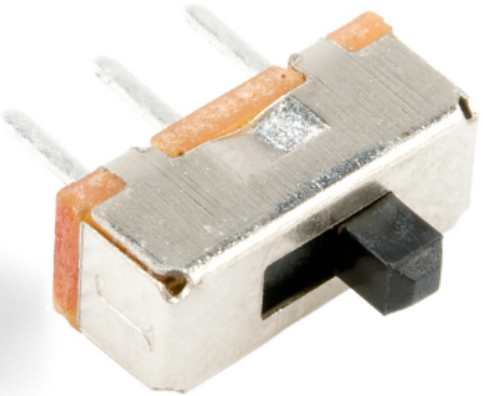
Esquema de la presentación

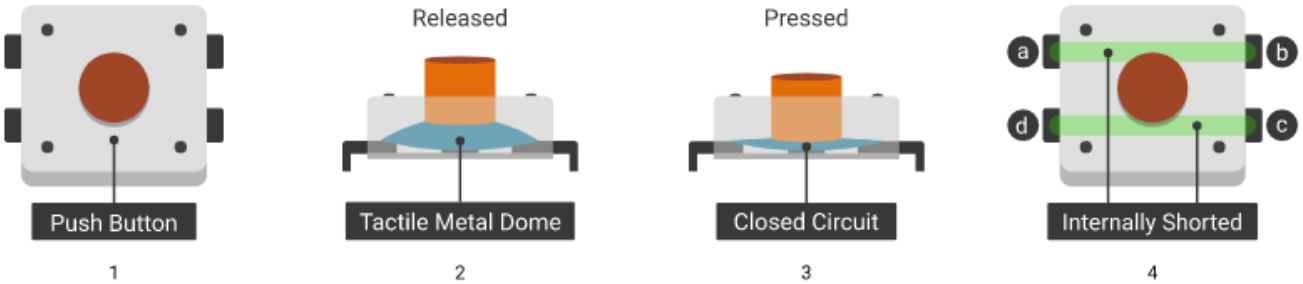
1. Repasamos lo que quedó de deberes
2. Materiales:
   - Protoboard
   - Resistencias
   - LEDs
     - Ejercicio 1 - Clase
3. Programación:
   - Conceptos básicos:
     - Variables y tipos de datos
     - Estructura de un código
   - Diagramas de flujo
     - Ejercicio 2 - Clase
   - Operadores
     - Comparación
     - Lógicos
     - Ejercicio 3 - Clase
   - Control de flujo
     - Sentencia "if"
   - Manejo de pines digitales
     - Ejercicio 4 - Clase
4. Conocemos más materiales
   - Pulsadores e interruptores
5. Ejercicios de deberes para la próxima clase
   - Ejercicio 1
   - Ejercicio 2
   - Ejercicio 3

Materiales

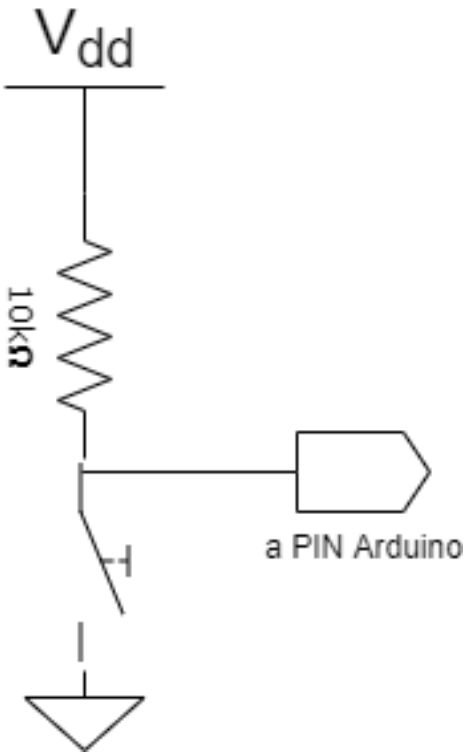
Pulsadores e interruptores

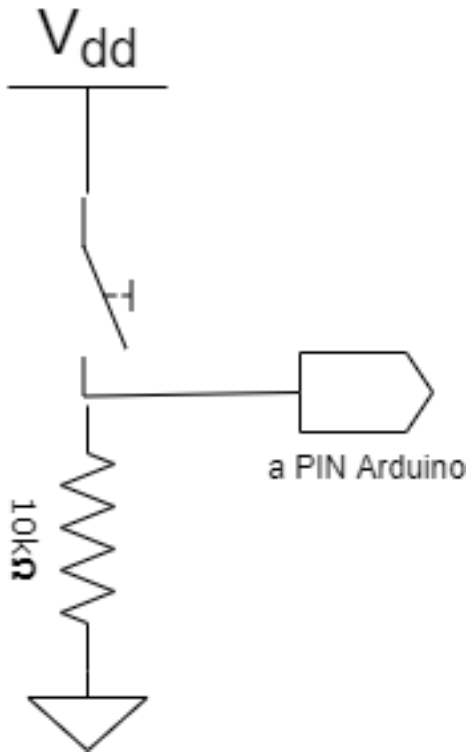






http://unciarobotics.com/robotics/





	A = PULL_UP	B = PULL_DOWN
CERRADO	Leo un '0'	Leo un '1'
ABIERTO	Leo un '1'	Leo un '0'

Esquema de la presentación

1. Repasamos lo que quedó de deberes
2. Materiales:
   - Protoboard
   - Resistencias
   - LEDs
     - Ejercicio 1 - Clase
3. Programación:
   - Conceptos básicos:
     - Variables y tipos de datos
     - Estructura de un código
   - Diagramas de flujo
     - Ejercicio 2 - Clase
   - Operadores
     - Comparación
     - Lógicos
     - Ejercicio 3 - Clase
   - Control de flujo
     - Sentencia "if"
   - Manejo de pines digitales
     - Ejercicio 4 - Clase
4. Conocemos más materiales
   - Pulsadores e interruptores
5. Ejercicios de deberes para la próxima clase
   - Ejercicio 1
   - Ejercicio 2
   - Ejercicio 3

Ejercicio 1 - Deberes

Utilizando el simulador y partiendo del proyecto del Ejercicio 4 de clase.

Sin modificar el código, se pide:

- Utilizar y conectar interruptores o pulsadores para poder modificar en la misma simulación el estado de las entradas 5 y 6.
- Una vez que se observa el comportamiento adecuado, replicar las conexiones con materiales reales, cargar el código a la placa y comprobar qué el funcionamiento es el esperado.

Nota: en caso de utilizar interruptores, solo se necesitará utilizar la pata del medio y la de uno de los costados, para usar los tipos de conexiones vistos.

Ejercicio 2 - Deberes

A partir del Ejercicio 1 de clase, haciendo uso de materiales reales y del IDE de Arduino.

Se pide:
- Modificar el circuito y la programación para que 3 leds se vayan prendiendo y apagando secuencialmente.
Es decir, se prende el primer led por un segundo, luego se apaga y en ese instante se prende el siguiente por otro segundo. El segundo se apaga y se prende el último, cuando éste se apaga se vuelve a prender el primero y se repite el ciclo.

Nota: se debe utilizar protoboard.

Ejercicio 3 - Deberes

A partir del último ejercicio de la secuencia de 3 leds (Ejercicio 2 de deberes).

Se pide:
- Modificar la programación e incorporar pines de entrada y pulsadores (ver Ejercicio 4 de clase) para lograr qué si el valor de la entrada 5 está a $0\,V$ se recorran los leds de derecha a izquierda. Y si la entrada 5 está a $5\,V$ se recorran de izquierda a derecha.

Resumen para la próxima clase:

1. Si no se terminaron los ejercicios para hacer en este taller, terminarlos.
2. Tener funcionando los ejercicios 1, 2 y 3 de deberes para compartir en la siguiente clase. Por dudas utilizar el *Foro de consultas*. Estaremos disponibles para clase de consulta si así lo solicitan durante la semana.
3. Queda disponible un cuestionario sobre esta clase, que deberá ser completado en el sitio EVA. Lo deberá hacer cada estudiante individualmente!!
4. Se recomienda continuar con la lectura de la documentación sugerida en la sección de *Introducción* en el sitio de EVA. En particular, seguir leyendo sobre control de flujo (sentencias como "*for*", "*while*" y "*switch-case*"), leer sobre pines digitales con PWM, entradas analógicas y funciones.