



Facilino

PROGRAMACIÓN DE ARDUINO CON FACILINO

- Arduino se puede programar fácilmente gracias a la herramienta de Facilino.
- Facilino permite, mediante bloques, crear código para realizar determinadas funciones en Arduino como leer información de los sensores, imprimir texto por una pantalla LCD, etc.

ENTORNO DE PROGRAMACIÓN



¿CÓMO ENVIAR UN PROGRAMA?

- Aseguraos que la versión del microcontrolador es **“Arduino Uno”**.
- El puerto serie dependerá de la configuración de vuestro ordenador, pero al conectar y desconectar el cable USB os cambiará la lista, con lo que fácilmente lo podréis averiguar.
- Al conectar el cable USB se encenderá la luz del LED ON de forma fija.
- Al verificar un código aparecerá una ventana que nos indicará si hay algún problema. Si todo es correcto debe indicar **“Build finished”** o **“Finalizó”**.
- Al subir un código, primero verificará que esté bien, después lo enviará a Arduino y las luces de los LEDs TX y RX parpadearán. Cuando haya finalizado el programa comenzará a ejecutarse nada más finalizar.

AYUDA DE LAS INSTRUCCIONES DE FACILINO

- Facilino incorpora abundante documentación de las instrucciones. Pulsa sobre el botón derecho de una instrucción para conocer cómo utilizarla.
- Encontrarás también circuitos y códigos de ejemplo para aprender a usar la instrucción.





Bloques de Facilino

MATEMÁTICAS

- Número (entero o decimal).
- `min ()` Mínimo entre dos números.
- Operación aritmética entre dos números.
- Aleatorio entre `y` Valor aleatorio entre dos números.
- Mapea De [`-`] a [`-`] Interpolación lineal de un valor comprendido en un rango de entrada a un rango de salida.
- Senoide Amplitud Frecuencia Fase Valor medio Tiempo Onda senoidal, para valores oscilatorios alrededor de un valor medio que varían con el tiempo.

DIGITAL

- Lee el pin digital PIN `Pin digital D0` Devuelve el estado (ALTO o BAJO) de un pin digital. Selecciona un pin digital.
- Escribe en el pin digital PIN `estado` Modifica el estado de un pin digital. Estado señal digital: ALTO o BAJO.
- Escribe en PIN digital `ciclo de trabajo` Genera una señal PWM (valor entre 0 y 255). Pin digital PWM.

LÓGICA

- `=` Compara dos números (enteros o decimales).
- `y` Compara dos expresiones Booleanas.
- `no` Negación de una expresión Booleana.
- Combina dos números de 16-bits en un número de 32-bits

COMUNICACIÓN USB

- Imprime por puerto serie con salto de línea `ψ` Imprime texto o números por la consola.
- Dibuja datos por puerto serie `ψ` Grafica datos por la consola.

CONTROL DE FLUJO

- Esperar [ms] `1000` Espera un tiempo en ms.
- Inicio Ejecuta instrucciones al inicio (sólo al arrancar una sola vez) o instrucciones de forma repetitiva (durante el funcionamiento habitual).
- Repetir Ejecuta casos en función del valor numérico. Añade tantos como necesites.
- Si Ejecución periódica de tareas o ejecución alternativa de casos (añade tareas o casos según convenga).
- Si Hacer Bucles, bucles y más bucles...
- Cada
- Alternar
- Contar con desde hasta mientras Hacer

VARIABLES

- Declara variable `var2` de tipo `Texto` = Declaración de variables para almacenar información. Diferentes tipos de variables dependiendo del tipo de información a almacenar.
- Declara variable `var1` =
- Var `var1` = Escribir (establecer) el valor de una variable.
- Var `var2` Leer (obtener) el valor de una variable.

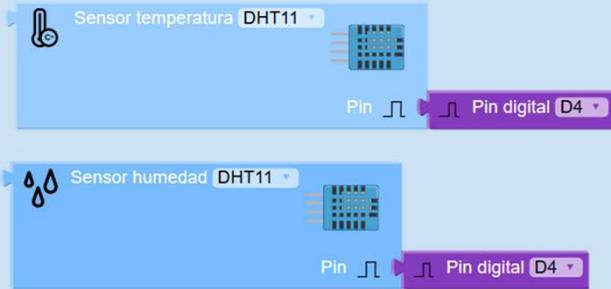
FUNCIONES Y PROCEDIMIENTOS

Organiza y reaprovecha mejor el código mediante llamadas a funciones y procedimientos.



Bloques de Facilino

TEMPERATURA Y HUMEDAD



Lee valores de temperatura y humedad con el sensor **DHT11**. La temperatura se mide en grados centígrados y la humedad es relativa (porcentaje). **El sensor está conectado al pin D4.** Debes asegurarte que entre transcurre al menos un tiempo de unos **2000ms** entre lecturas (no importa que sea de humedad o temperatura).

ZUMBADOR



Crea sonidos variados que se modifiquen en función de ciertas condiciones (p.e. la distancia del sensor de ultrasonidos). **El zumbador está conectado al pin D5.**



Crea una melodía musical. Necesitas indicar el pin del zumbador y la melodía a reproducir (en este caso es una melodía predefinida). Si seleccionas la tarea de fondo reproduce la música mientras ejecuta las siguientes instrucciones.



Puedes crear tu propia partitura añadiendo notas musicales al pentagrama. Puedes utilizar una gran combinación de notas musicales y silencios. **La partitura debe acabar con el fin de pentagrama.**

LDR



Obtén valores calibrados de luminancia con un sensor de luz LDR en luxes. **Establece valores máximos o mínimos** en luxes que esperas recibir para calcular la luz como un porcentaje. **Este sensor está conectado al pin A1.**

SERVO



Mueve un servo indicando la posición deseada en grados. **Debes esperar al menos 20ms** antes de volver a usar esta instrucción. **Conecta el servo al pin A3.**

PANTALLA LCD

Esta instrucción va siempre en el inicio



Conecta la pantalla al bus I2C (líneas SDA y SCL).



Muestra texto en una pantalla LCD de 16x2 caracteres. Puedes hacer **desplazamiento de texto** si no cabe todo en una línea.



SENSOR DE GAS



Detecta muchos tipos de gases con los sensores MQx. **Calibra el sensor** para un funcionamiento correcto y después, obtén **lecturas del gas calibradas en ppm.** **Conecta el sensor al pin A3.**



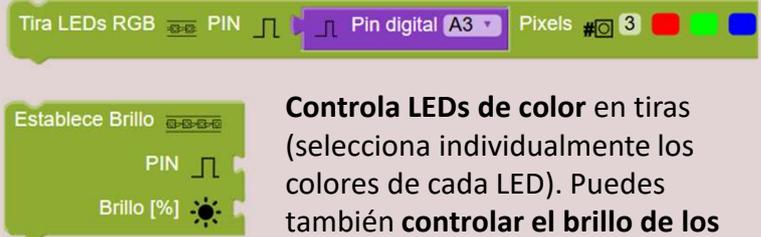
Bloques de Facilino

LEDS RGB



Seleccionar un color a mostrar en un LED RGB de 3 pines digitales. **Conecta el LED a los pines D9, D11 y D10, correspondientes a los colores Rojo (R), Verde (G) y Azul (B).**

TIRAS DE LEDS



Controla LEDs de color en tiras (selecciona individualmente los colores de cada LED). Puedes también **controlar el brillo de los LEDs**. **Conecta los LEDs al pin A3.**

RECEPTOR INFRARROJO



Recibe comandos de mandos infrarrojos. Según el botón pulsado y la marca del mando, el código puede ser diferente. Utilizan instrucciones asociadas para averiguar el código de cada botón. **El receptor infrarrojos está conectado al pin D6.**

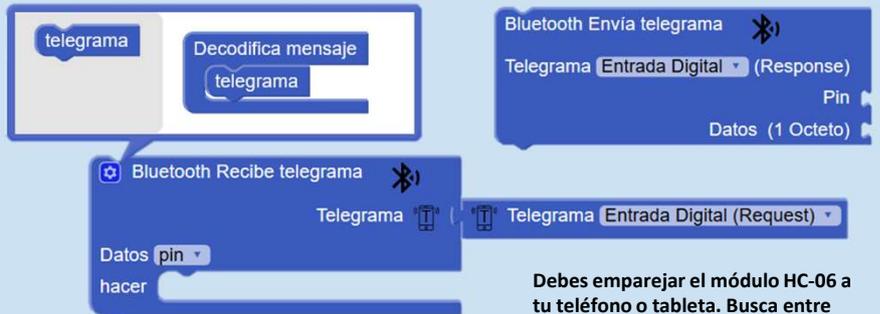
BLUETOOTH

Esta instrucción va siempre en el inicio



Define la comunicación bluetooth. Necesitas indicar los pines a los que están conectados los pines RX y TX del módulo bluetooth. **Baudios = 9600. Esta líneas van cruzadas. Conecta TX a D7 y RX a D8.**

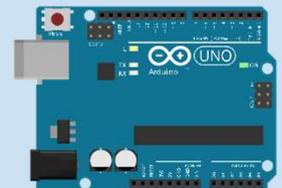
Telegramas: Entrada digital, Salida Digital, Entrada Analógica, Salida Analógica, Servo, Tono Zumbador, Melodía Zumbador, DHT, etc..



Debes emparejar el módulo HC-06 a tu teléfono o tableta. Busca entre los dispositivos bluetooth disponibles y emparéjalo con el pin 1234.



1. App envía telegrama con información.
2. Arduino recibe telegrama.
3. Si se tiene que devolver información de vuelta, Arduino envía telegrama con la información.
4. App recibe telegrama



Existen **dos tipos de telegramas:** Los que **leen información** de un sensor (*Request*) o los que **establecen valores** en dispositivos (*Command*).

Los datos de un telegrama **se almacenan en variables** que pueden ser usadas **sólo dentro del contexto del telegrama.**

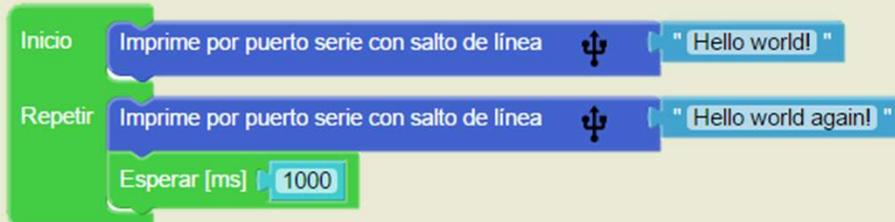
Los telegramas que solicitan información deben responder con otro telegrama con la información solicitada (véase documentación asociada para conocer el tipo de información a enviar).



Ejercicios Básicos

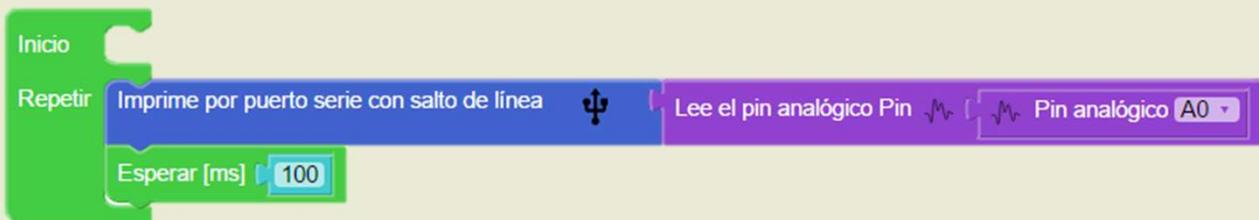
'HOLA MUNDO'

Imprime por la consola la expresión "Hello world!" al inicio (al arrancar) y después imprime la expresión "Hello world again!" cada segundo.



MOSTRAR VALORES DEL POTENCIÓMETRO

Crea un programa que envía datos de los valores medidos de la señal analógica del potenciómetro (conectado al pin A0). Los datos se envían por el puerto serie y que los puedes visualizar en el monitor serie.



GRAFICA VALORES DEL POTENCIÓMETRO

Utilizando la herramienta para graficar señales (de Facilino o Arduino IDE) muestra cómo varía el valor analógico del potenciómetro (conectado al pin A0) con el tiempo.



GRAFICA MULTIPLES VALORES

Ahora puedes usar la instrucción para graficar múltiples datos para ver cómo varían los valores analógicos del potenciómetro (conectado al pin A0) y el sensor de luz LDR (conectado al pin A1).





Ejercicios Básicos

ENCENDER UN LED AL PULSAR BOTÓN

Cuando pulsamos el pulsador conectado al pin D2 su valor es BAJO, mientras que si no está pulsado, el valor es ALTO. Enciende el LED conectado al pin D12 cuando no pulsamos el pulsador.

```
Inicio
Repetir
  Escribe en el pin digital PIN [ ] [ ] Pin digital D12 estado [ ]
  Lee el pin digital PIN [ ] [ ] Pin digital D2
```

PARPADEO DE LEDS

Haz parpadear los LEDs conectados a los pines D12 (rojo) y D13 (azul) de forma alternativa. Cuando uno está en estado ALTO, el otro debe estar en estado BAJO y viceversa. El tiempo de parpadeo de cada LED debe ser de 1Hz (una vez por segundo).

```
Inicio
Repetir
  Escribe en el pin digital PIN [ ] [ ] Pin digital D12 estado [ALTO]
  Escribe en el pin digital PIN [ ] [ ] Pin digital D13 estado [BAJO]
  Esperar [ms] [500]
  Escribe en el pin digital PIN [ ] [ ] Pin digital D12 estado [BAJO]
  Escribe en el pin digital PIN [ ] [ ] Pin digital D13 estado [ALTO]
  Esperar [ms] [500]
```

PARPADEO DE LEDS CON FRECUENCIA VARIABLE

Ahora puedes usar la instrucción para graficar múltiples datos para ver cómo varían los valores analógicos del potenciómetro (conectado al pin A0) y el sensor de luz LDR (conectado al pin A1).

```
Inicio
Repetir
  Escribe en el pin digital PIN [ ] [ ] Pin digital D12 estado [ALTO]
  Escribe en el pin digital PIN [ ] [ ] Pin digital D13 estado [BAJO]
  Esperar [ms] [Mapea] Lee el pin analógico Pin [ ] [ ] Pin analógico A0 De [ ] [0] - [ ] [1023] a [ ] [100] - [ ] [1000]
  Escribe en el pin digital PIN [ ] [ ] Pin digital D12 estado [BAJO]
  Escribe en el pin digital PIN [ ] [ ] Pin digital D13 estado [ALTO]
  Esperar [ms] [Mapea] Lee el pin analógico Pin [ ] [ ] Pin analógico A0 De [ ] [0] - [ ] [1023] a [ ] [100] - [ ] [1000]
```



Ejercicios Básicos

OPERACIONES BOOLEANAS

Implementa la operación Booleana usando dos pulsadores (conectados a los pines D2 y D3). Si pulsamos cualquiera de los botones se debe encender el LED rojo (conectado al pin D12), mientras que cuando pulsemos los dos botones se debe encender el LED azul (conectado al pin D13).



COMPARACIÓN DE UN NÚMERO

Lee el valor de la entrada analógica A0 correspondiente al potenciómetro. Divide el rango total en 4 sub-rangos de forma que los LEDs D12 y D13 muestran el número del rango en representación binaria (D12 es el bit menos significativo).





Ejercicios Básicos

SUMA DOS NÚMEROS

Implementa una función que toma como argumentos de entradas dos números y devuelve la suma de los mismos. Interpreta dos números enteros recibidos por el monitor serie y llama a la función para realizar la suma. Muestra el resultado de la suma por el monitor serie.

```
func. (con retorno) sum int x, int y
ejecutar
  devuelve Entero
  Var x
  +
  Var y
```

VARIABLES LOCALES Y GLOBALES

Implementa un programa que imprima el valor de una variable local al 'Inicio' e imprima otra variable local (con el mismo nombre) en la sección 'Repetir'.

Luego, crea un programa que declare una variable global en el 'Inicio' y use esa variable en la sección de 'Repetir'.

```
Inicio
  Declara variable i = 34
  Imprime por puerto serie con salto de línea
  Crea texto con " This is a local variable: i="
  Var i

Repetir
  Declara variable i = 23
  Imprime por puerto serie con salto de línea
  Crea texto con " This is another local variable: i="
  Var i
```

```
Inicio
  Declara variable GLOBAL i = 34
  Imprime por puerto serie con salto de línea
  Crea texto con " This is a global variable: i="
  Var i

Repetir
  Imprime por puerto serie con salto de línea
  Crea texto con " We can use it here too! i="
  Var i
```



Ejercicios Básicos

ONDA SENOIDAL

Crea una onda senoidal con una frecuencia de 1Hz, una amplitud de $\sqrt{2}$, usando la instrucción 'senoidal'. Dibuja la señal usando el Serial Plotter de Arduino.

```
Inicio
  Declara variable GLOBAL time0 de tipo Entero largo = Tiempo desde el arranque (us)
Repetir
  Declara variable time de tipo Entero largo = Tiempo desde el arranque (us) -> Var time0
  Dibuja datos por puerto serie Senoide Amplitud 1.41 Frecuencia 1 Fase 0 Valor medio 0 Tiempo Var time
  Esperar [ms] 20
```

UNIR TEXTO

Usa la instrucción 'Crea texto con' para unir texto con números para imprimir el valor del potenciómetro conectado al pin A0 con un mensaje justo antes del valor, por ejemplo: 'The value of the potentiometer is '.

```
Inicio
Repetir
  Imprime por puerto serie con salto de línea
  Crea texto con "The value of the potentiometer is "
  Lee el pin analógico Pin Pin analógico A0
  Esperar [ms] 100
```

MÍNIMO Y MÁXIMO

Usando las medidas obtenidas del potenciómetro (conectado al pin A0), calcular los valores máximo y mínimo de los mismos (los valores estarán escalados para estar entre 0 y 100). Al iniciar, enciende el LED rojo (conectado al pin D12) y realiza los cálculos para obtener el máximo y mínimo histórico hasta que se pulse el pulsador conectado al pin D2. Una vez pulsado, apagad el LED rojo y mostrar por consola los valores del máximo y mínimo de todas las medidas registradas.

```
Inicio
  Declara variable min_val = 100
  Declara variable max_val = 0
  Escribe en el pin digital PIN Pin digital D12 estado ALTO
  mientras
    Lee el pin digital PIN Pin digital D2 = ALTO
  Hacer
    Declara variable val = Mapea Lee el pin analógico Pin Pin analógico A0 Valor entre [0-100]
    Var min_val = min ( Var min_val , Var val )
    Var max_val = max ( Var max_val , Var val )
  Escribe en el pin digital PIN Pin digital D12 estado BAJO
  Imprime por puerto serie con salto de línea Crea texto con " Min: "
  Var min_val
  " Max: "
  Var max_val
Repetir
```



Ejercicios con DHT11

MEDIR TEMPERATURA Y HUMEDAD

Leer la temperatura y humedad con el sensor DHT11. Entre lecturas consecutivas, ya sea de temperatura o humedad, debemos esperar unos 2000ms para poder realizar correctamente la medición. Utiliza la instrucción 'Cada' para controlar el tiempo que tardas entre las mediciones de la misma variable.

```
Inicio
Repetir
  Cada
    Tiempo transcurrido (ms) 5000
    hacer
      Declara variable temperature = Sensor temperatura DHT11
      Esperar [ms] 2000
      Declara variable humidy = Sensor humedad DHT11
      Esperar [ms] 2000
      Imprime por puerto serie con salto de línea
      Crea texto con
        " Temperature (Celsius): "
        Var temperature
        " Humidity (%): "
        Var humidy
```

MEDIR TEMPERATURA Y HUMEDAD SIN ESPERAS

En ocasiones, **es conveniente no introducir esperas**, ya que esto puede afectar a otras actividades, como por ejemplo leer información del bluetooth (**mientras esperas, no se pueden procesar los telegramas o hacer otras tareas**). Una forma sencilla de solventar esto, es usar una instrucción de alternar casos. Se han omitido las instrucciones de mostrar el resultado por pantalla.

```
Inicio
  Declara variable GLOBAL temperature = 0
  Declara variable GLOBAL humidity = 0
Repetir
  Cada
    Tiempo transcurrido (ms) 5000
    hacer
      Alternar
        Caso 1
          hacer
            Var temperature = Sensor temperatura DHT11
        Caso 2
          hacer
            Var humidity = Sensor humedad DHT11
```



Ejercicios con LDR

MEDICIÓN DE LUZ

Imprime los valores de las mediciones sin procesar de un sensor LDR conectado al pin A1 y también los valores calibrados de luminancia (en luxes). Apunta los valores de la luminancia (aproximados) que lees cuando tapas o no el sensor con la mano.

```
Inicio
Repetir
  Imprime por puerto serie con salto de línea "Raw: "
  Leer LDR PIN Pin analógico A1
  "Luminance: "
  Luminancia (lux) PIN Pin analógico A1
Esperar [ms] 100
```

CONTROL DE ILUMINACIÓN CON UN LED

Ahora, con los valores previamente anotados, establece los valores máximo y mínimo de luminancia de forma que podamos devolver la luminancia como un porcentaje entre estos rangos de luz. Si el valor medido está por debajo del 30%, enciende el LED rojo (conectado al pin D12). Si el valor medido está por encima del 70%, apaga el LED rojo.

```
Inicio
  Establece Luz Máxima 500
  Establece Luz Mínima 100
Repetir
  Declara variable raw = Leer LDR PIN Pin analógico A1
  Declara variable calibrated = Luminancia (%) PIN Pin analógico A1
  Imprime por puerto serie con salto de línea "Raw: "
  Crea texto con "Percent: "
  Var raw
  Var calibrated
  si Var calibrated <= 30
  Hacer Escribe en el pin digital PIN Pin digital D12 estado ALTO
  en cambio, si Var calibrated >= 70
  Hacer Escribe en el pin digital PIN Pin digital D12 estado BAJO
Esperar [ms] 100
```



Ejercicios con LEDs de Colores

ALTERNAR COLORES

Usando el LED RGB, alternar los colores Rojo, Verde y Azul cada 1000ms.



VARIAR LA INTENSIDAD DE LUZ DE UN LED

Regula la intensidad de luz del LED rojo, de forma que varía de 0 a intensidad máxima (255) en un bucle.





Ejercicios con LEDs de Colores

COLOR ALEATORIO

Genera un color aleatorio usando el LED RGB cada segundo utilizando combinaciones de colores en los canales RGB que generen una señal PWM (en cada canal) que varíe entre 0 y 255.

```
Inicio
Repetir
  Escribe en PIN digital [Pin PWM D9] ciclo de trabajo [Aleatorio entre 0 y 255]
  Escribe en PIN digital [Pin PWM D11] ciclo de trabajo [Aleatorio entre 0 y 255]
  Escribe en PIN digital [Pin PWM D10] ciclo de trabajo [Aleatorio entre 0 y 255]
  Esperar [ms] [1000]
```

TRANSICIÓN DEL ROJO AL AMARILLO

Implementa una función que realice una transición de color, de un color a otro. Por simplicidad, se realizará la transición en dos canales, rojo y verde, para cambiar del rojo (R=250,G=0) al amarillo (R=255,G=255) y luego, del amarillo al rojo. La función realiza un bucle con 100 iteraciones en las que cambia progresivamente los colores, desde los valores iniciales en los canales rojo y verde hasta los valores finales, con un retardo de 10ms en cada iteración.

```
Inicio
Repetir
  RG_color_fade
  Red1 255
  Green1 0
  Red2 255
  Green2 255
  time 10
  RG_color_fade
  Red1 255
  Green1 255
  Red2 255
  Green2 0
  time 10

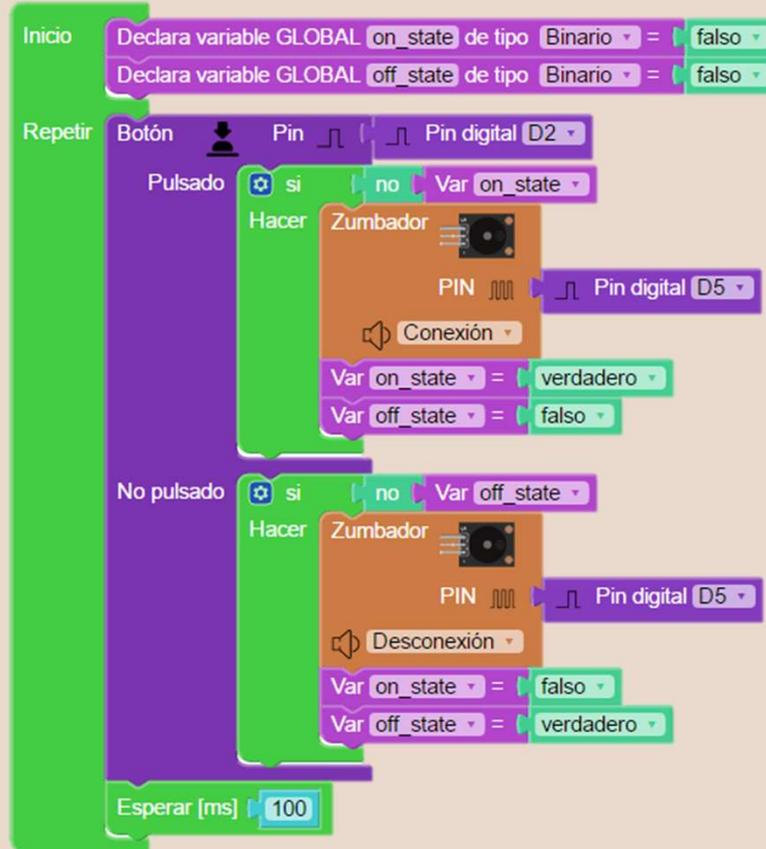
func. (sin retorno) RG_color_fade byte Red1, byte Green1, byte Red2, byte Green2, ..
ejecutar
  Declara variable deltaR de tipo Decimal = A número Decimal [Var Red2 - Var Red1] + 100
  Declara variable deltaG de tipo Decimal = A número Decimal [Var Green2 - Var Green1] + 100
  Declara variable i = 0
  Contar con Var i desde 1 hasta 100
  Hacer
    Declara variable Red de tipo Octeto = A número Octeto [Var deltaR * Var i] + Var Red1
    Declara variable Green de tipo Octeto = A número Octeto [Var deltaG * Var i] + Var Green1
    Escribe en PIN digital [Pin PWM D9] ciclo de trabajo [Var Red]
    Escribe en PIN digital [Pin PWM D11] ciclo de trabajo [Var Green]
    Esperar [ms] [Var time]
```



Ejercicios con Zumbador de Sonido

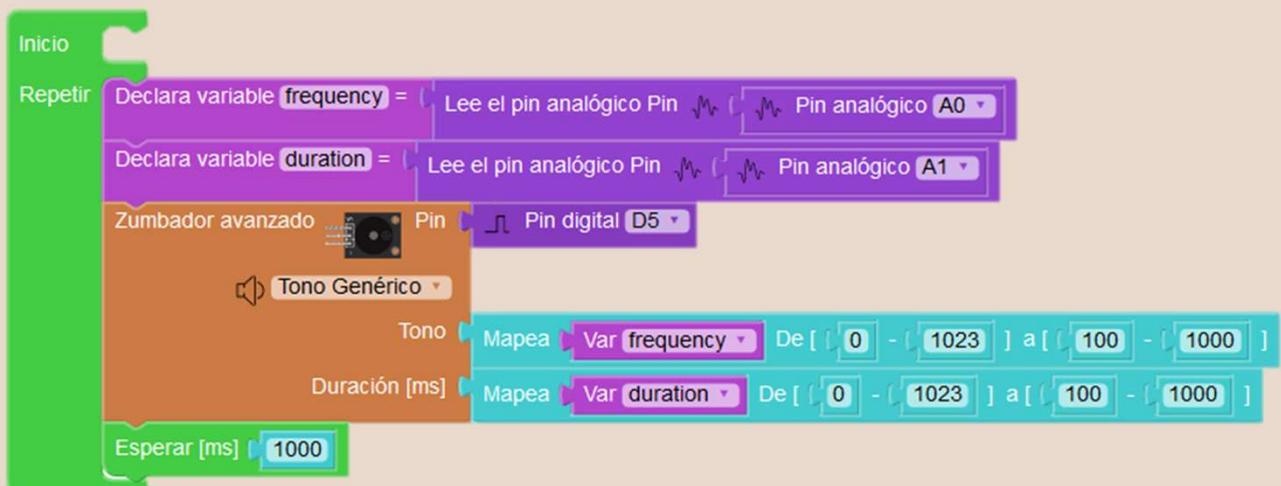
GENERANDO SONIDOS PREDEFINIDOS

Genera un sonido de conexión cuando se presione el botón del interruptor D2 y un sonido de desconexión cuando se suelte.



GENERANDO PITIDOS CON FRECUENCIA Y DURACIÓN VARIABLE

Genera un pitido cuya frecuencia de vibración y duración dependa de la posición del potenciómetro y la cantidad de luz recibida. El pitido puede variar entre 100Hz y 1000Hz con una duración que puede oscilar entre 100ms y 1000ms. Espera 1000ms después de cada pitido.

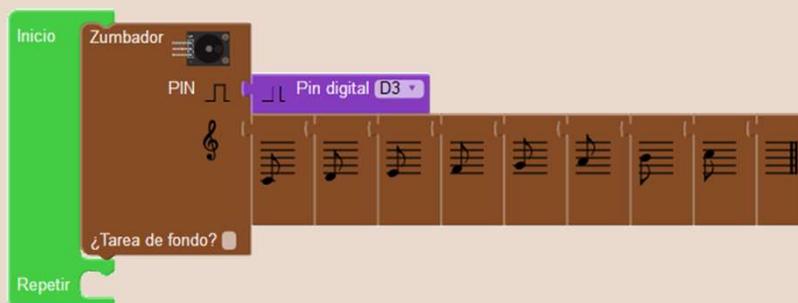




Ejercicios con Zumbador de Sonido

GENERANDO MELODÍAS

Para generar sonidos, con una melodía personalizada, podemos crear un pentagrama con las notas musicales a reproducir. En este ejemplo, se genera una escala musical al inicio del programa (si ponemos el bloque en el bucle principal las notas musicales no pararán de reproducirse, mientras que al ponerlas en el bloque de inicio, sólo sonarán una vez).

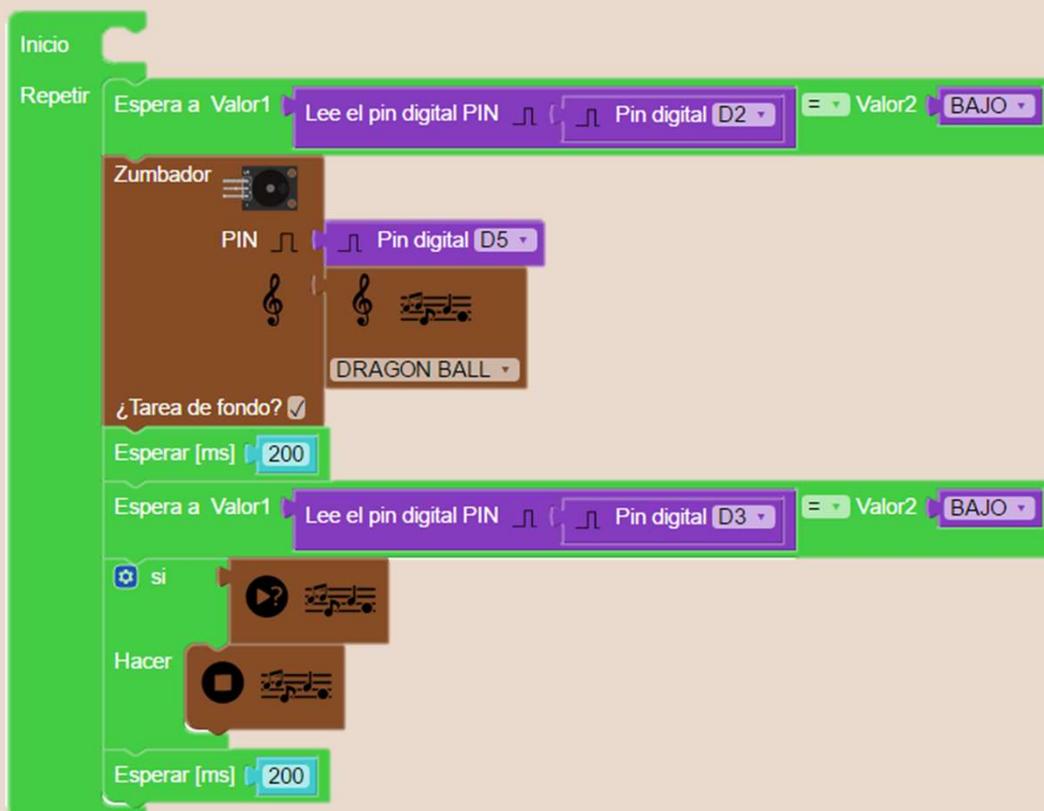


Prueba a crear tu propia melodía. Aquí te proporcionamos un ejemplo.



GENERANDO MELODÍAS EN UNA TAREA DE FONDO

Facilino permite crear melodías que se ejecutan en una tarea de fondo. Esto te permite realizar otras tareas mientras se reproduce el sonido. Ojo que al activar la casilla de tarea de fondo, se trabaja con interrupciones y algunas librerías/bloques son incompatibles con esta opción.





Ejercicios con Pantalla LCD

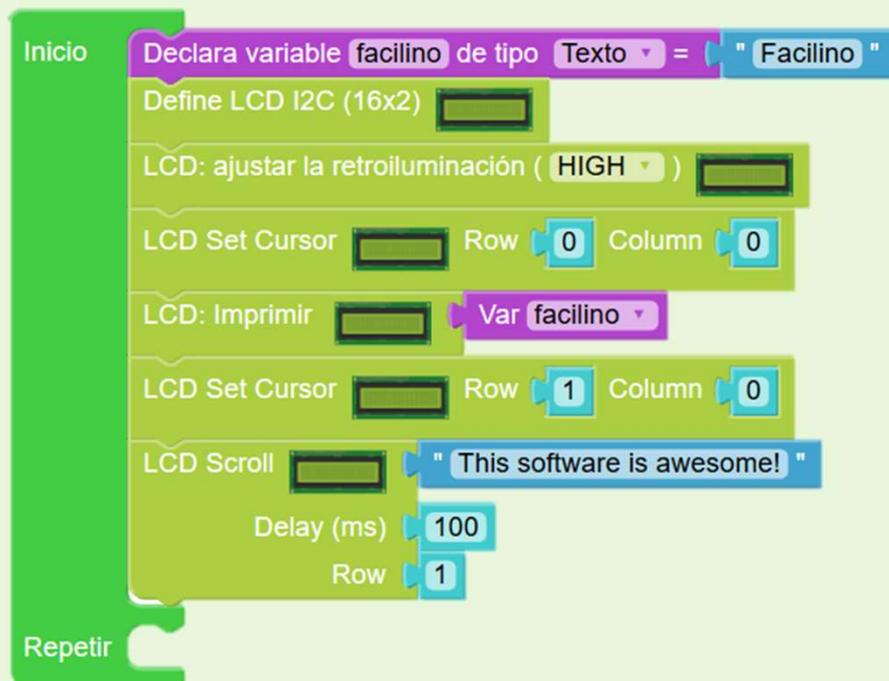
MOSTRAR TEXTO ESTÁTICO CENTRADO

Mostrar la expresión "This is a very long text" en la pantalla LCD. Como la expresión completa no cabe en una sola línea, escribiremos "This is a very" en la primera línea (fila 0) y "long text" en la segunda línea (row 1). Si queremos centrar el texto, podemos posicionar el inicio del texto en la columna según su longitud, tal como se muestra:



MOSTRAR TEXTO DESPLAZADO

En ocasiones, el mensaje que queremos mostrar es demasiado largo, con lo que podemos desplazar el texto (crear una animación que desplazará el texto hacia la izquierda). En esta ocasión, mostramos el texto "Facilino" en la primera línea (fila 0) y la expresión "This software is awesome!" en la segunda línea (fila 1) con un texto desplazado cada 100ms.

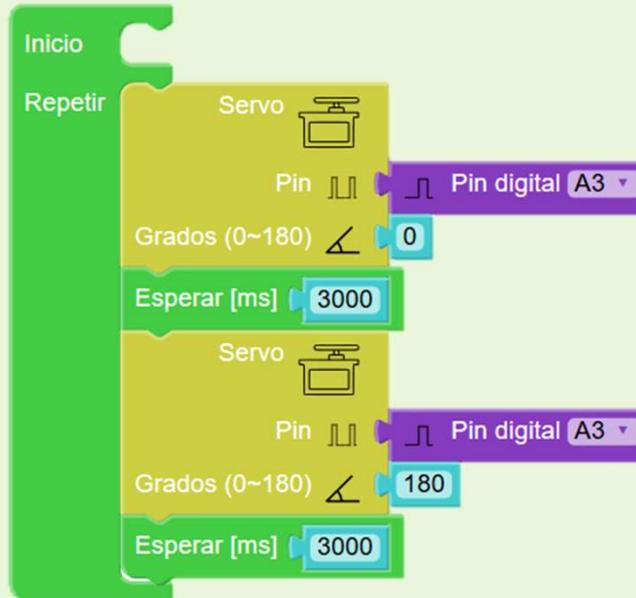




Ejercicios con Servo

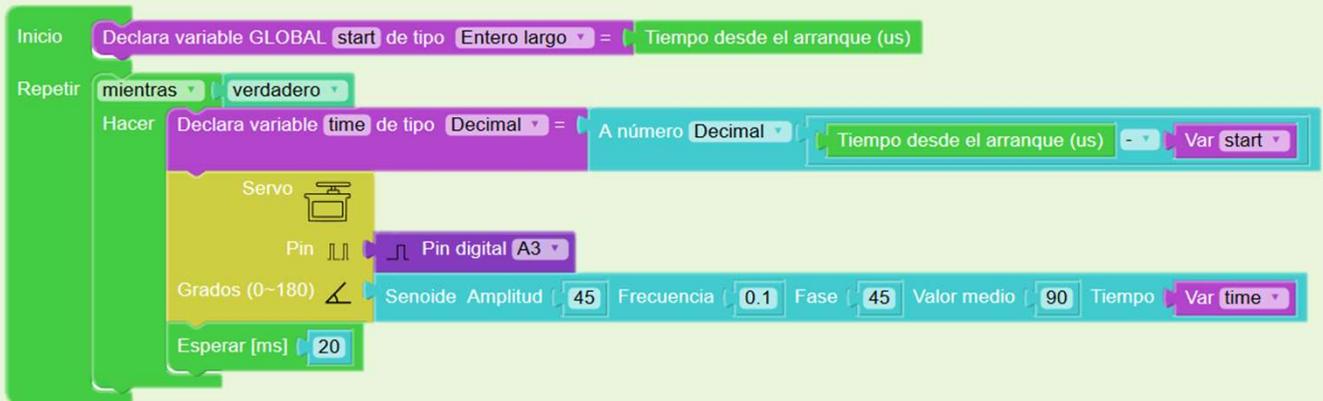
MOVIENDO UN SERVO

Mueve un servo a sus posiciones 0° y 180° cada 3000 ms. Puedes colocar una hoja agitada por el servo para producir un flujo de aire similar a un abanico.



MOVIMIENTO ONDULAR CON UN SERVO

Genera una onda senoidal con un servo para que la posición varíe entre 45° y 135° (esto suele ser útil para los patrones de caminar en algunos robots o para generar un flujo de aire para disminuir la temperatura con una hoja fijada al servo emulando un abanico).





Ejercicios con Sensor de Gas

ALARMA BÁSICA DE GAS

Implementa una alarma que se dispare si el valor de la lectura del sensor de gas supera una determinada cantidad y que se apague si los valores de gas detectados vuelven a la normalidad (por debajo de un valor). Aquí hemos utilizado 120 y 200 como los valores umbrales, pero podrían ser muy diferentes en tu caso, con lo que debes comprobarlo realizando algunas mediciones previas.

```
Inicio
Repetir
  Declara variable gas = Lee el pin analógico Pin (Pin analógico A3)
  si Var gas <= 120
    Hacer Zumbador No tone PIN (Pin digital D5)
  en cambio, si Var gas >= 200
    Hacer Zumbador PIN (Pin digital D5)
    Modo 1
  Esperar [ms] 200
```

CALIBRACIÓN DEL SENSOR DE GAS

Para poder utilizar el sensor de gas en condiciones optimas, debes calibrarlo para el entorno en el que trabaja. Déjalo unos 20-30 minutos encendido y ejecuta el siguiente código, pulsa el botón SW1 y anota el valor mostrado en consola.

```
Inicio
  Imprime por puerto serie con salto de línea (\"Waiting for pre-heat up... Press the button when...\")
  mientras Botón Pin (Pin digital D2) = ALTO
    Hacer
      Imprime por puerto serie con salto de línea (\"Calibrating...\")
      Esperar [ms] 1000
      Declara variable R0 de tipo Decimal = Calibra R0 (Sensor MQ2, Pin analógico A3)
      Imprime por puerto serie con salto de línea (\"Sensor calibrated. R0= \" + Var R0 + \" kOhms \")
  Repetir
```

ALCOHOLÍMETRO

Con el valor de resistencia previamente anotado, ahora puedes realizar medidas calibradas de las ppm de alcohol en el aire.

```
Inicio
  Declara variable GLOBAL R0 de tipo Decimal = 1.76
  Repetir
    Imprime por puerto serie con salto de línea Leer Gas (ppm) (Sensor MQ2, Gas Alcohol, Pin analógico A3, R0 Var R0)
  Esperar [ms] 1000
```



Completa el Programa!

RECEPCIÓN DE TELEGRAMAS

El siguiente código muestra cómo gestionar la recepción de telegramas. Por favor, sigue las instrucciones del propio código para completar el código. Los telegramas del tipo 'Request' requieren enviar un telegrama de vuelta del mismo tipo, pero sólo para el sensor solicitado, con lo que debe comprobarse que el pin es el correcto. La variable 'pin' NO debe usarse en las instrucciones de lectura o escritura, como por ejemplo los LEDs. Esta variable debe usarse para comparar que su valor coincide con el pin esperado (véase la siguiente página para más aclaración).

```
Inicio
  Define Bluetooth
    Tasa de baudios 9600
    RX Pin digital D7
    TX Pin digital D8

  Declara variable GLOBAL temp de tipo Entero largo = 0
  Declara variable GLOBAL humid de tipo Entero largo = 0
  Declara variable GLOBAL pot = 0
  Declara variable GLOBAL LDR = 0
  Declara variable GLOBAL SW1 de tipo Binario = falso
  Declara variable GLOBAL SW2 de tipo Binario = falso

Repetir
  // Lee temperatura y humedad cada 10s
  // sin retardos. Almacena las medidas en 'temp'
  // y 'humid para usarlas después
  // Lee el potenciómetro y guárdalo en 'pot'
  // Lee la LDR y guárdala en 'LDR'
  // Lee los pulsadores y almacénalos en 'SW1'
  // y 'SW2'

  Bluetooth Recibe telegrama
    Telegrama Telegrama Entrada Digital (Request)

  Datos pin
  hacer
    // Envía un telegrama con el estado del pulsador
    // comprueba el pin y manda el del pulsador
    // correspondiente (sólo ese).
    Telegrama Telegrama Salida Digital (Command)

  Datos pin
  hacer
    // Establece el estado del LED correspondiente
    Telegrama Telegrama Entrada Analógica (Request)

  Datos pin
  hacer
    // Envía un telegrama con la lectura analógica
    // Sólo la que corresponda según el pin
    Telegrama Telegrama Salida Analógica (Command)

  Datos pin
  hacer
    // Establece el valor del PWM correspondiente
    Telegrama Telegrama DHT (Request)

  Datos pin
  hacer
    // Envía un telegrama con la información
    // Combina la 'humid' y 'temp' en un 'Entero largo'
    Telegrama Telegrama Tono Zumbador (Command)

  Datos pin
  hacer
    // Establece la frecuencia y duración correspondie...
```



Ejercicios con Bluetooth

AL RECIBIR UN TELEGRAMA DEL TIPO 'COMMAND'...

```
Bluetooth Recibe telegrama
  Telegrama Servo (Command)
  Datos pin
  hacer
    Servo
      Pin 0
      Var pin
      (0-180)
      Var value
```

```
Bluetooth Recibe telegrama
  Telegrama Servo (Command)
  Datos pin
  hacer
    si Var pin = Pin digital A3
      Hacer
        Servo
          Pin Pin digital A3
          Grados (0-180)
          Var value
```

El ejemplo de código de la izquierda no funcionará, porque al inicio del programa, hará que el servo se conecte a lo que vale pin (0 al inicio) y por tanto no funcionará como se espera. El ejemplo de la derecha, comprueba que el valor del pin coincide con el esperado y usa la instrucción del movimiento del servo junto con la instrucción de pin al que está conectado (tal y como se espera).

AL RECIBIR UN TELEGRAMA DEL TIPO 'REQUEST'...

```
Bluetooth Recibe telegrama
  Telegrama Entrada Analógica (Request)
  Datos pin
  hacer
    Bluetooth Envía telegrama
      Telegrama Entrada Digital (Response)
      Pin Var pin
      Datos (1 Octeto) A número Octeto
      Lee el pin digital PIN Var pin
```

```
Bluetooth Recibe telegrama
  Telegrama Entrada Digital (Request)
  Datos pin
  hacer
    si Var pin = Pin digital D2
      Hacer
        Bluetooth Envía telegrama
          Telegrama Entrada Digital (Response)
          Pin Pin digital D2
          Datos (1 Octeto) A número Octeto
          Lee el pin digital PIN Pin digital D2
```

De forma similar, en este ejemplo, el código de arriba no funciona como se espera, porque se está realizando la lectura del pin con la variable 'pin', en vez de la instrucción con el pin digital 'D2'.



App Inventor 2

CONFIGURACIÓN

Iniciar sesión en App Inventor 2 (requiere una cuenta de Gmail)

<http://ai2.appinventor.mit.edu>

VISTA 'DISEÑADOR' DE APP INVENTOR 2



Componentes:

- **Visibles:** Botones, etiquetas, imágenes, lienzos, distribución (horizontal, vertical, etc...), deslizadores, etc.
- **No Visibles:** Reloj, acelerómetro, cliente bluetooth, bases de datos, etc.

Extensiones: componentes no visibles que mejoran las funcionalidades básicas de AI2. Extensión de los ficheros *aix*.

Visor: Dispone de una interfaz de usuario que permite 'previsualizar' el aspecto de la App. Es el lugar donde colcaremos los componentes (visibles y no visibles). Los no visibles aparecerán como una lista de componentes justo debajo de la interfaz de usuario.

Jerarquía de componentes: Lista con los componentes de la ventana. Si un componente está dentro de otro, como por ejemplo al usar componentes de distribución, éstos aparecerán con una jerarquía (unos dentro de otros).

Propiedades de componente: Para el componente seleccionado, aparecen un conjunto de propiedades que pueden ser modificadas en tiempo de diseñador.

Lista de medios: conjunto de ficheros de medios (imágenes, audio, vídeo) que pueden ser usados en la aplicación.

PROGRAMACIÓN POR BLOQUES DE APP INVENTOR 2

App Inventor2 usa bloques para poder programar el código de la App (vista de 'Bloques'). Observaréis un conjunto de bloques que van desde los que están integrados, es decir, que todas las Apps puede hacer uso de esos bloques para realizar las funcionalidades más básicas como por ejemplo bucles, condicionales, variables, etc... Además, encontrarás una lista con los componentes que tiene la App. Los componentes pueden tener tres tipos de instrucciones:

- **Eventos** (color ocre) que se ejecutan cuando sucede algún fenómeno o evento, como por ejemplo un clic.
- **Métodos** (color morado) que realizan una función determinada.
- **Propiedades** (color verde) que permiten establecer o obtener el valor de una propiedad del componente.



Componentes Principales de AI2

INTERFAZ DE USUARIO

Texto para Botón1

Botón: permite pulsar sobre él y **detectar cuando hacemos clic, al presionar o al soltar**. Modifica las propiedades del botón para que tenga una aspecto diferente (texto, imagen, tamaño, etc.). **Propiedades del botón:** Ancho, alto, imagen, texto, visible...

cuando Botón1 .Clic
ejecutar

cuando Botón1 .ClicLargo
ejecutar

cuando Botón1 .Presionar
ejecutar

cuando Botón1 .Soltar
ejecutar



Imagen: Permite mostrar una *imagen estática*, es decir, no interactuamos con la imagen. **Propiedades de la imagen:** Ancho, alto, imagen, visible...



Texto para Etiqueta1

Etiqueta: Permite **mostrar texto e información** de determinados valores para informar al usuario. **Propiedades de la imagen:** Ancho, alto, texto, visible...

Texto para SelectorDeLista1

Selector de lista: Como un botón, pero que permite mostrar una lista para seleccionar un elemento. Incorpora un evento de '**AntesDeSelección**' y otro de '**DespuésDeSelección**'. **Lo utilizaremos para mostrar la lista de dispositivos bluetooth disponibles.**

cuando SelectorDeLista1 .AntesDeSelección
ejecutar

cuando SelectorDeLista1 .DespuésDeSelección
ejecutar

poner SelectorDeLista1 . Elementos como

SelectorDeLista1 . Selección



Deslizador: Permite modificar la barra de deslizamiento con el dedo entre un valor máximo y mínimo.

cuando Deslizador1 .PosiciónCambiada

posiciónDelPulgar
ejecutar

tomar posiciónDelPulgar

poner posiciónDelPulgar a

Deslizador1 . PosiciónDelPulgar



Texto para Switch1



Conmutador: Permite conmutar un interruptor a **encendido o apagado**.

cuando Switch1 .Cambiado
ejecutar

poner Switch1 . On como

Switch1 . On

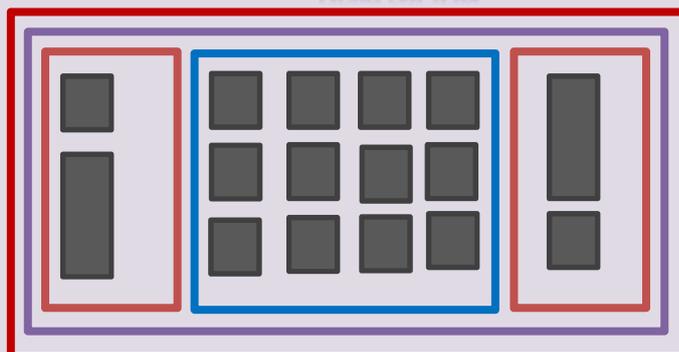
DISPOSICIÓN

Pantalla

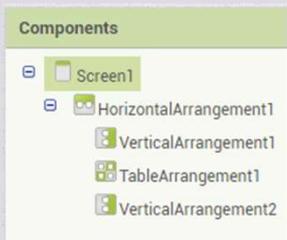
Contenedor horizontal

Contenedor vertical

Contenedor tabular



Los componentes dentro de un contenedor son "hijos" del contenedor jerárquicamente superior





Componentes Principales de AI2

SENSORES DEL DISPOSITIVO



Reloj

Reloj: Permite ejecutar instrucciones cada cierto intervalo de tiempo en milisegundos.

```
when Clock1 .Timer
do
```

IntervaloDelTemporizador

BLUETOOTH



ClienteBluetooth

Cliente bluetooth: Permite conectarse al módulo bluetooth HC-05, HC-06, SPP-C o a cualquier dispositivo bluetooth (clásico). Normalmente utilizaremos este componente para enviar o recibir datos por bluetooth, si bien, aquí usaremos una extensión que se encargará de la gestión de los telegramas.

```
llamar ClienteBluetooth1 .Conectar
dirección
```

```
ClienteBluetooth1 . DireccionesYNombres
```

```
ClienteBluetooth1 . Disponible
```

```
llamar ClienteBluetooth1 .Desconectar
```

```
ClienteBluetooth1 . Conectado
```

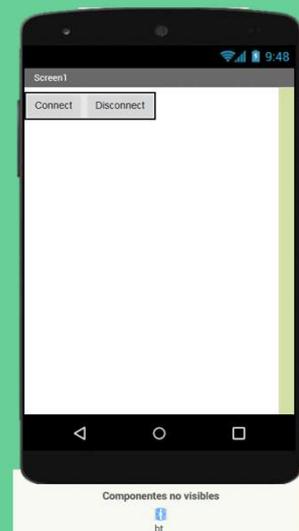
EJEMPLO DE CONEXIÓN/DESCONEXIÓN BLUETOOTH

Necesitamos un componente del tipo Selector de Lista, que renombramos 'connect', un botón con nombre 'disconnect' y el cliente bluetooth con nombre 'bt'. Trata de conseguir el aspecto mostrado en la pantalla utilizando un contenedor con disposición horizontal.

```
cuando connect .AntesDeSelección
ejecutar
  si
  entonces poner connect . Elementos como bt . DireccionesYNombres
```

```
cuando connect .DespuésDeSelección
ejecutar
  si
  entonces llamar bt .Conectar
  dirección connect . Selección
```

```
cuando disconnect .Clic
ejecutar llamar bt .Desconectar
```





Extensión Facilino para AI2

DESCARGA DE LA EXTENSIÓN <https://roboticafacil.es/descargas/>

Descarga la extensión (aix) de Robótica Fácil e importa la extensión en tu proyecto de App Inventor 2 (en la paleta de componentes, al final del todo).



MÉTODOS Y EVENTOS PRINCIPALES DE LA EXTENSIÓN

Tipo	Nombre	Propiedades	Descripción
Principal	FacilinoBluetooth	<ul style="list-style-type: none">ClienteBluetoothIntervalorDelTemperozador	Seleccionar el componente del cliente bluetooth de tu App en la propiedad 'ClienteBluetooth'. El intervalo determina el tiempo en el que se comprueba si se ha recibido información de los sensores.
Sensor Analógico	AnalogRead	<ul style="list-style-type: none">AnalogPinFacilinoDevice	Pin analógico al que está conectado.
Sensor Digital	DigitalRead	<ul style="list-style-type: none">PinFacilinoDevice	Pin digital al que está conectado físicamente el sensor.
Actuador Digital	DigitalWrite	<ul style="list-style-type: none">PinFacilinoDevice	Pin digital al que está conectado físicamente el actuador.
Servo (de posición)	Servo	<ul style="list-style-type: none">PinFacilinoDevice	Pin digital al que está conectado físicamente el sensor.
Temperatura y humedad	DHT	<ul style="list-style-type: none">PinFacilinoDeviceHighHumidityThresholdLowHumidityThresholdHighTemperatureThresholdLowTemperatureThreshold	Pin digital al que está conectado físicamente el actuador. Además, dispone de umbrales de temperatura y humedad para generar avisos.
Señal PWM	AnalogWrite	<ul style="list-style-type: none">PinFacilinoDevice	Pin PWM al que está conectado físicamente el actuador.
Zumbador de sonido	Buzzer	<ul style="list-style-type: none">PinFacilinoDevice	Pin PWM al que está conectado físicamente el zumbador.
Servo (de rotación continua)	ServoCont	<ul style="list-style-type: none">PinFacilinoDevice	Pin digital al que está conectado físicamente el actuador.

Selecciona el componente 'FacilinoBluetooth' en la propiedad 'FacilinoDevice' del resto de componentes.

Pines (Arduino Uno):

- Los pines digitales D0-D13 corresponden a los números 0-13.
- Los pines PWM son: 3, 5, 6, 9, 10, y 11.
- Los pines analógicos/digitales A0-A5 corresponden a los números 14-19.



Extensión Facilino para AI2

SENSORES

llamar Pot .Request

Request: Envía una solicitud de lectura del sensor (no bloqueante). Tras recibir la información, se ejecutará el evento 'Received'.

llamar Pot .Update

Update: Envía una solicitud de lectura del sensor y espera a recibir la información (bloqueante).

cuando Pot .Received

value

ejecutar

Received: Evento que se llama tras recibir una lectura de un sensor. La variable 'value' contiene la información de lectura.

cuando SW1 .Cambiado

value

ejecutar

cuando DHT1 .HighHumidity

ejecutar

Otros eventos: Algunos sensores, disponen de otros eventos que avisan de cambios de la señal medida, como por ejemplo la temperatura elevada en los sensores DHT o el cambio de estado de una entrada digital.

ACTUADORES

call Red_LED .Toggle

Digital Write: Tienen dos posibles métodos. Puedes conmutar el valor actual del LED o establecer un valor a verdadero o falso. En la figura se muestra cómo puede establecerse este valor con la propiedad 'On' de un conmutador.

call Red_LED .Set

value

Switch2 . On

call RGB_Red_LED .Set

value

Red_slider . ThumbPosition

call Green_RGB_LED .Set

value

Green_slider . ThumbPosition

call Blue_RGB_LED .Set

value

Blue_slider . ThumbPosition

Analog Write: Tiene un único método para establecer el valor de la señal PWM. Acepta un número entre 0 y 255. Aquí se muestra cómo cambiar el color de un LED RGB con tres deslizadores.

call Buzzer1 .Melody

melody

make a list
440
125
493
250

Buzzer: Tiene dos métodos. El primero permite enviar una melodía (lista de frecuencias y duraciones) a ser reproducida. El segundo método acepta una única frecuencia y una duración.

call Buzzer1 .Tone

frequency

freq_slider . ThumbPosition

duration

duration_slider . ThumbPosition

call Servo1 .Move

angle

servo_slider . ThumbPosition

Servo: Tiene un único método que permite mover el servo al ángulo indicado.



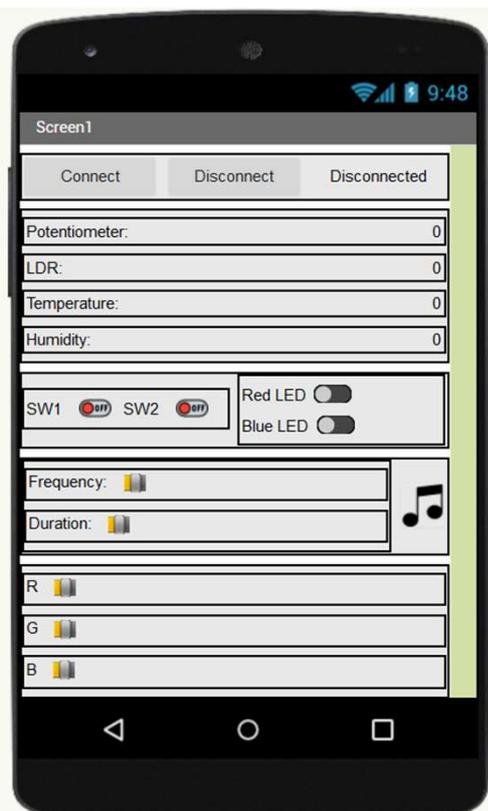
Completa la App!!

INTERFAZ APP

Hemos creado una interfaz con la App que tienes que trabajar (accede al código QR para descargar el fichero 'aia'). La interfaz incluye los componentes principales según se indica en la tabla. Completa la App para comunicarte con tu dispositivo!

https://roboticafacil.es/facilino/ai2/demos/Multisensor_ArduinoUno_vacio.zip

Nombre	Tipo	Descripción
connect	Selector de lista	Pulsa para mostrar la lista de dispositivos bluetooth disponibles y conéctate a la placa multisensor.
Disconnect	Botón	Pulsa para desconectar.
PotVal	Etiqueta	Muestra el valor del potenciómetro.
LDRVal	Etiqueta	Muestra el valor de la LDR.
tempVal	Etiqueta	Muestra el valor de temperatura.
humidVal	Etiqueta	Muestra el valor de la humedad.
SW1_img	Imagen	Cambia la imagen según el estado del pulsador SW1 de la placa multisensor.
SW2_img	Imagen	Cambia la imagen según el estado de pulsador SW2 de la placa multisensor.
Switch1	Conmutador	Pulsa para cambiar el estado del LED rojo.
Switch2	Conmutador	Pulsa para cambiar el estado del LED azul.
freq_slider	Deslizador	Desliza para cambiar la frecuencia del sonido del zumbador.
duration_slider	Deslizador	Desliza para cambiar la duración del sonido del zumbador.
Red_slider	Deslizador	Desliza para cambiar el componente rojo del LED RGB.
Green_slider	Deslizador	Desliza para cambiar el componente verde del LED RGB.
Blue_slider	Deslizador	Desliza para cambiar el componente azul del LED RGB.
playSound	Botón	Pulsa para reproducir la nota.



CÓDIGO QR



RECUERDA EMPAREJARTE CON EL DISPOSITIVO! PIN: 1234



Soluciones



Si estás aquí es porque no has sido capaz de finalizar todas las actividades propuestas, lo cual lamentamos mucho.

Aún así, queremos proporcionarte las soluciones para el control remoto del robot, pero no te lo vamos a poner tan fácil. Debes decodificar la siguiente URL (las letras han sido modificadas):

`sggkh://ilylgrxzuzxro.vh/uzxrorml/zr2/wvnlh/Nfogrvmhli_ZiwfrmlFml.ark`

Pista

Sólo las letras de la a-z han sido codificadas (los números y los símbolos especiales como . : / _ no se han modificado. Se mantienen las mayúsculas y minúsculas.