


## 7.4 Reto A04. El pulsador

En el siguiente reto vamos a utilizar el pulsador. Previamente debemos recordar que diferencia hay entre un pulsador y un interruptor. Un interruptor es un dispositivo que abre o cierra en paso de la corriente eléctrica, por ejemplo, los interruptores de la luz de nuestras casas, cada vez que los pulsamos cambiamos de estado y permanecen en él hasta ser pulsados de nuevo. Sin embargo, un

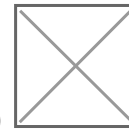
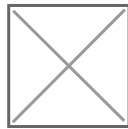
pulsador  sólo se activa mientras dure la pulsación volviendo a su estado inicial en el momento en el que se deje de pulsar.



Símbolo



Componente



Hay dos tipos de pulsadores;  los NA (normalmente abierto)  o los NC

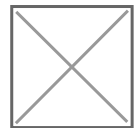
(normalmente cerrado), con lo que al pulsarlo se activará la función inversa de la que en ese momento este realizando.



La placa Imagina TDR STEAM tiene dos pulsadores de nominados SW1 y SW2 que van asociados a los pines D2 y D7 respectivamente.

Ahora vamos a realizar un programa en el cual al pulsar sobre el pulsador se encienda el led y se apague cuando lo dejemos de pulsar. En el menú de *Sensores* encontramos los dos bloques

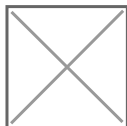
correspon



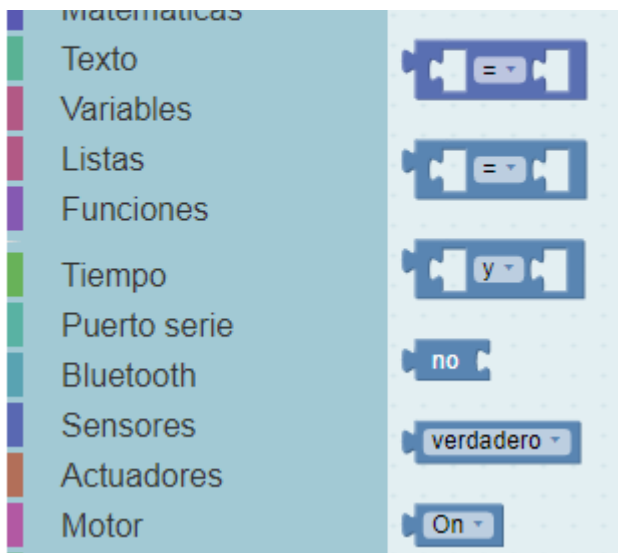
## 7.4.1 Control ON/OFF de un led con un pulsador



Para realizar este programa necesitamos conocer unas de las funciones más utilizadas en programación. Las funciones del menú *Lógica* con las funciones de condición (condicionales).



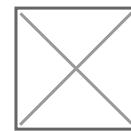
Condicionales:

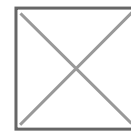


*si (condición) hacer (acciones)*

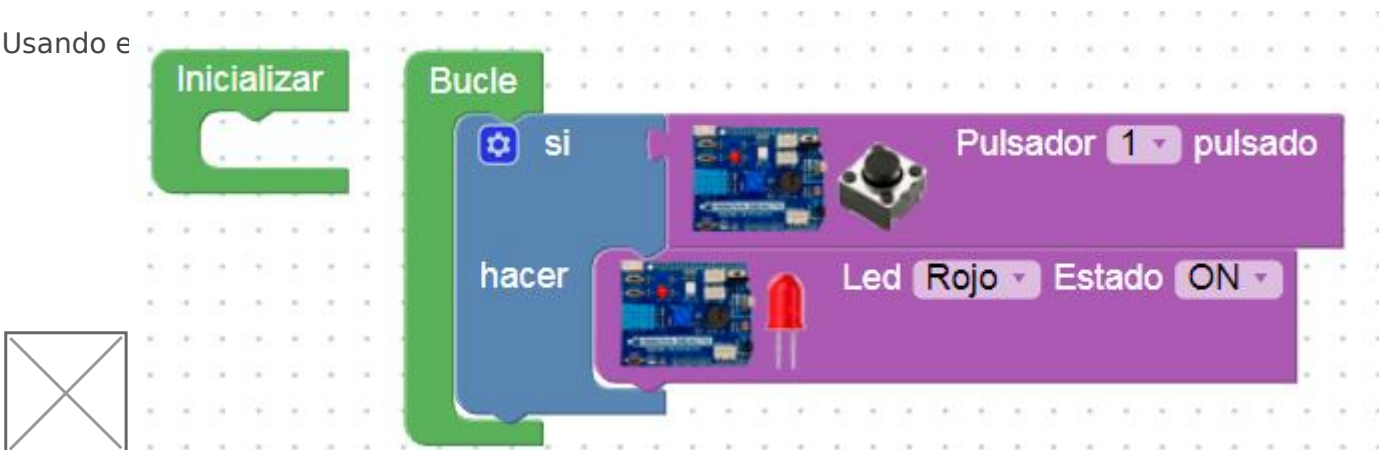
Se trata del famoso bucle *Si* (*if* en inglés) que es uno de los pilares de la programación, ya que permite evaluar estados y tomar decisiones en consecuencia.

El funcionamiento es el siguiente: si se cumple la condición incluida en su primer apartado, entonces se realiza la acción incluida en su segundo apartado. En caso contrario, no se hace nada.

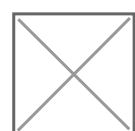
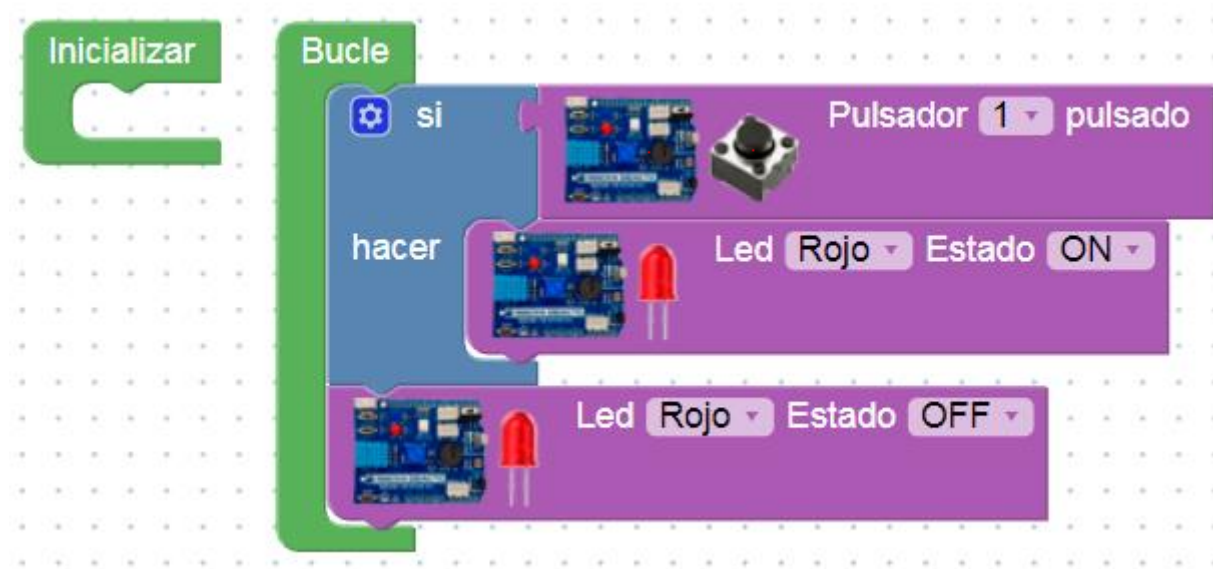


En el apartado de condiciones se pueden introducir multitud de factores:  estado de sensores (analógicos o digitales), comparaciones, igualdades, operaciones matemáticas, etc.

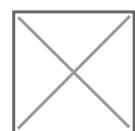
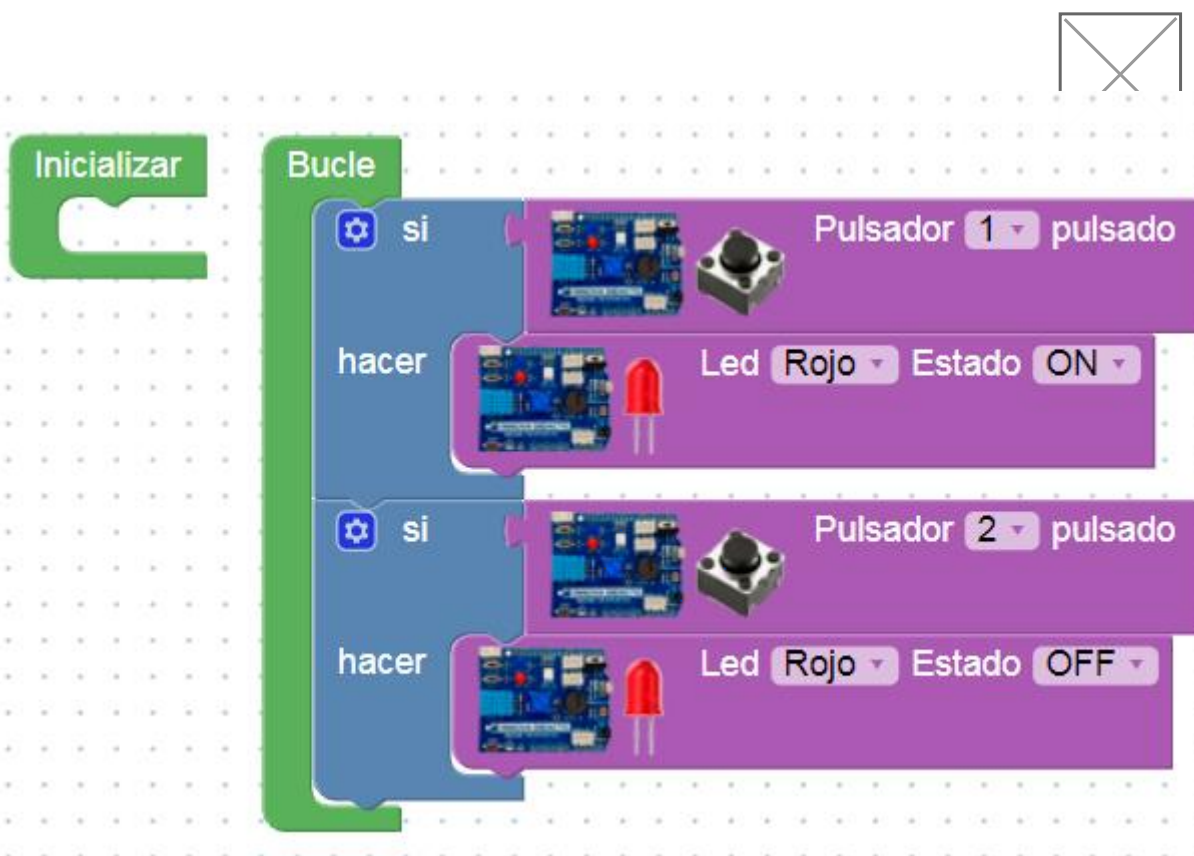
Usando e



No se apaga el led azul nunca, esto no es lógico, en ningún momento del programa decimos que el led tenga que estar en la posición OFF. En el siguiente programa conseguiremos que el led rojo solamente esté encendido cuando apreteemos el pulsador 1.



SW2 se a

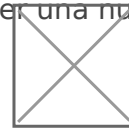


Actividad de ampliación: prueba ahora de apaga los dos leds con el pulsador SW1 y encenderlos con SW2.

## 7.4.2 Control ON/OFF de un led con un pulsador II

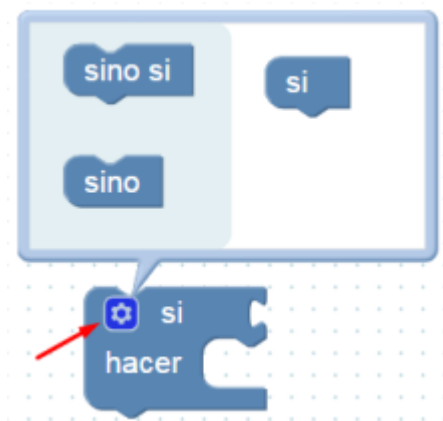


Necesitar dos pulsadores para tener que controlar un solo Led no parece la mejor opción, necesitamos introducir un sino. Si usamos un único pulsador debemos poner una nueva condición:

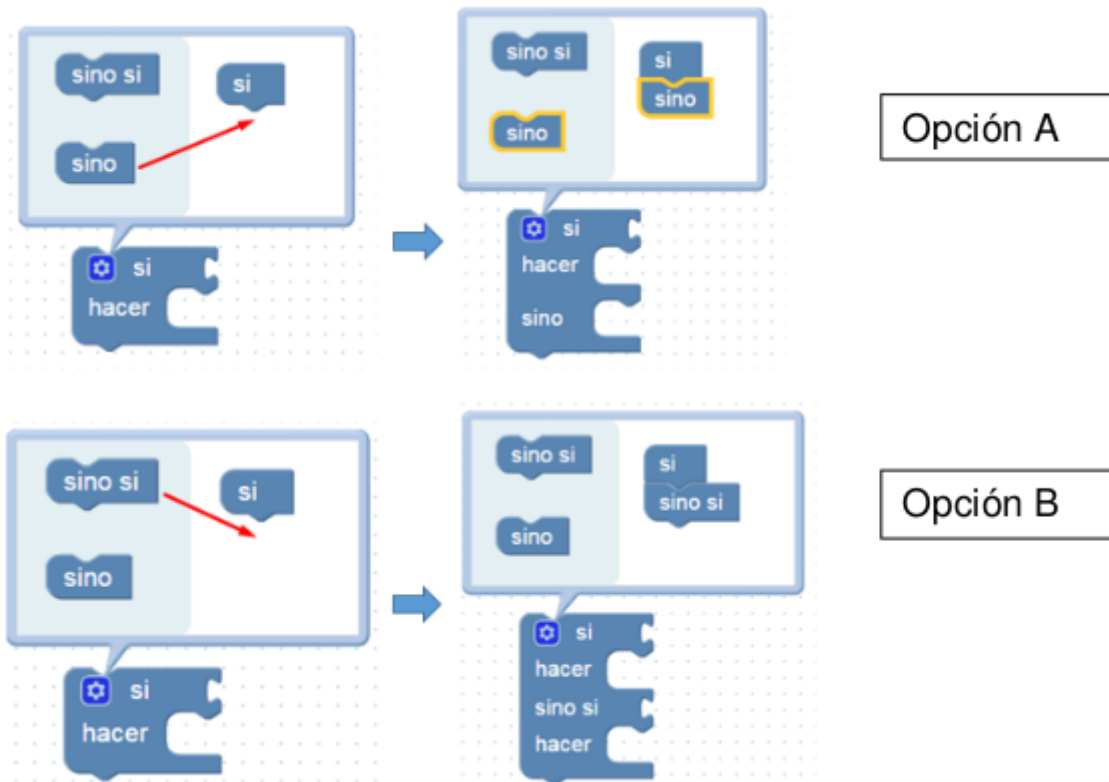


un *sino* para cambiar el estado del led a OFF. Para ello debemos ampliar el condicional.

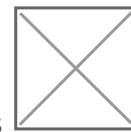
Pulsando sobre el símbolo del engranaje (señalado con la flecha roja en la imagen) nos aparece un cuadro con funciones con las que podemos ampliar el condicional *Si*.



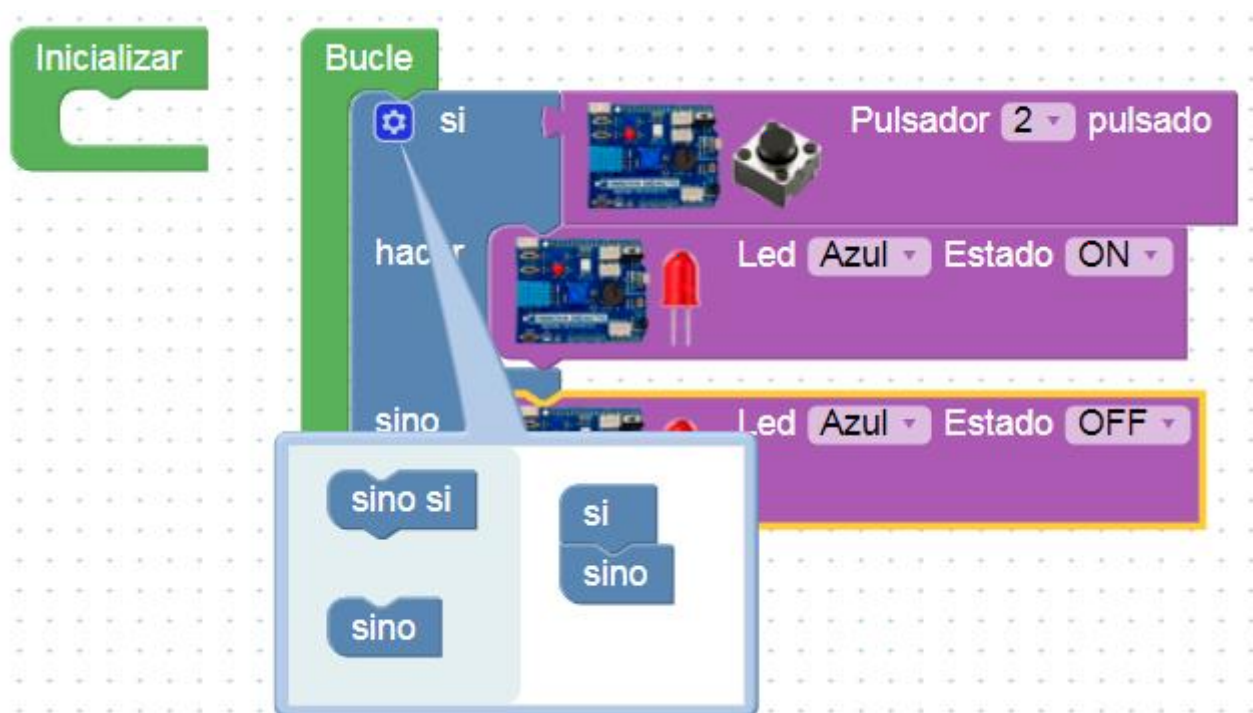
Hay dos opciones que se consiguen arrastrando los bloques como se aprecia en las siguientes imágenes:



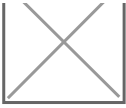
Podemos ir encadenando varias estructuras (opción B).



Veremos ejemplos del uso de estas variantes a lo largo de diferentes programas en este documento. Pero, continuando con este reto, vamos a realizar un programa que al pulsar el pulsador 2 se encenderá el led azul y *sino*, se apagará.







El código Morse es un sistema de representación de letras y números mediante señales emitidas de forma intermitente. Estas señales pueden ser luminosas como con nuestro Led o también podrían ser acústicas utilizando el zumbador.



Este es el código Morse:

<b>A</b> • —	<b>U</b> • • —
<b>B</b> — • • •	<b>V</b> • • • —
<b>C</b> — • — •	<b>W</b> • — —
<b>D</b> — • •	<b>X</b> — • • —
<b>E</b> •	<b>Y</b> — • — —
<b>F</b> • • — •	<b>Z</b> — — • •
<b>G</b> — — •	
<b>H</b> • • • •	
<b>I</b> • •	
<b>J</b> • — — —	
<b>K</b> — • —	<b>1</b> • — — — —
<b>L</b> • — • •	<b>2</b> • • — — —
<b>M</b> — —	<b>3</b> • • • — —
<b>N</b> — •	<b>4</b> • • • • —
<b>O</b> — — —	<b>5</b> • • • • •
<b>P</b> • — — •	<b>6</b> — • • • •
<b>Q</b> — — • —	<b>7</b> — — • • •
<b>R</b> • — •	<b>8</b> — — — • •
<b>S</b> • • •	<b>9</b> — — — — •
<b>T</b> —	<b>0</b> — — — — —

Actividad de ampliación: haz una máquina de código Morse que envíe un mensaje de ayuda (SOS).

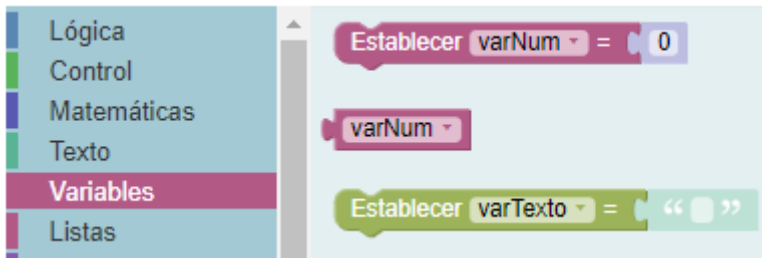
## 7.4.3 Control ON/OFF de un led con un pulsador III



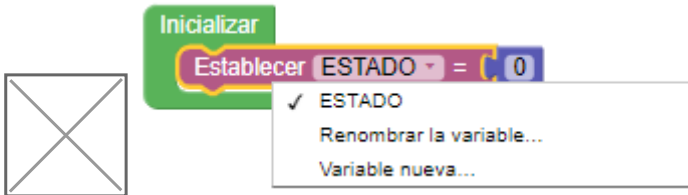
En este reto vamos a hacer que el Led Azul se quede en estado encendido o apagado pulsando una sola vez el pulsador SW1.

En el programa que vamos a hacer a continuación necesitamos asegurarnos que cada vez que accionamos el pulsador sea detectado como una sola señal, ya que vamos a realizar un contador.

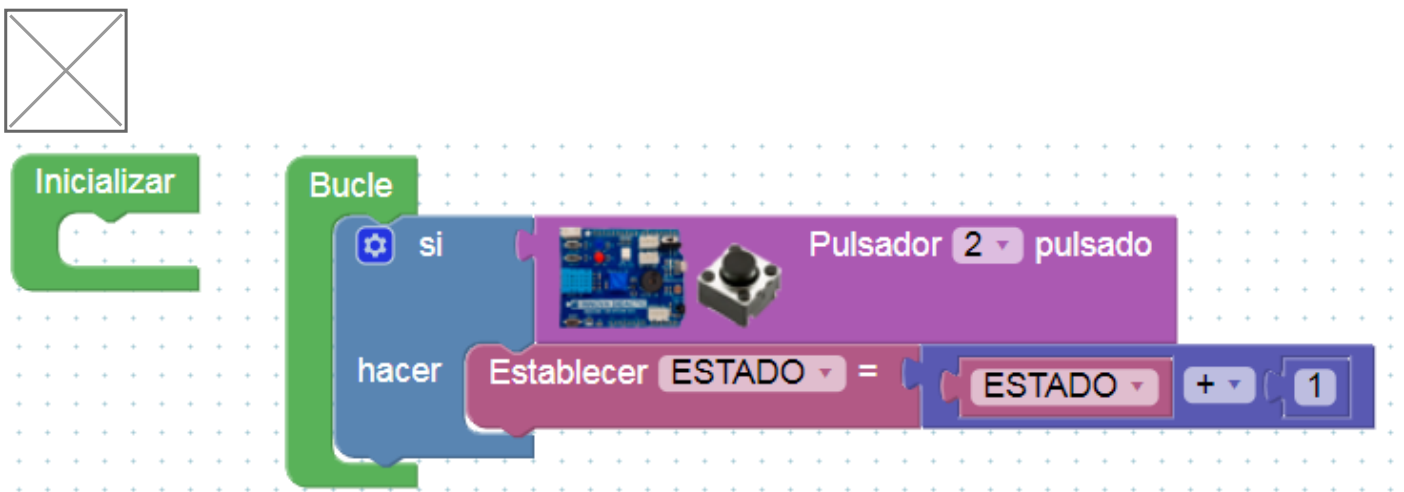
Para hacer el contador vamos a utilizar una variable que la llamaremos *Estado*.

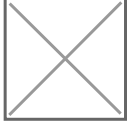


En *Inicializar* establecemos la variable a 0 y la llamamos *ESTADO*. Esta variable la utilizaremos como indicador de cambio de estado del pulsador.



El contador consiste en ir sumando una unidad a la variable *ESTADO* cada vez que se dé al pulsador. Cada vez que pulsemos el pulsador, la variable tendrá su valor anterior más 1. Por ejemplo: si al inicio del programa la variable *ESTADO* tiene un valor de 0, al dar al pulsador su nuevo valor será igual a  $0+1=1$ ; al volver a dar al pulsador su valor será  $1+1=2$ ; al volver a pulsar  $2+1=3$ , etc.



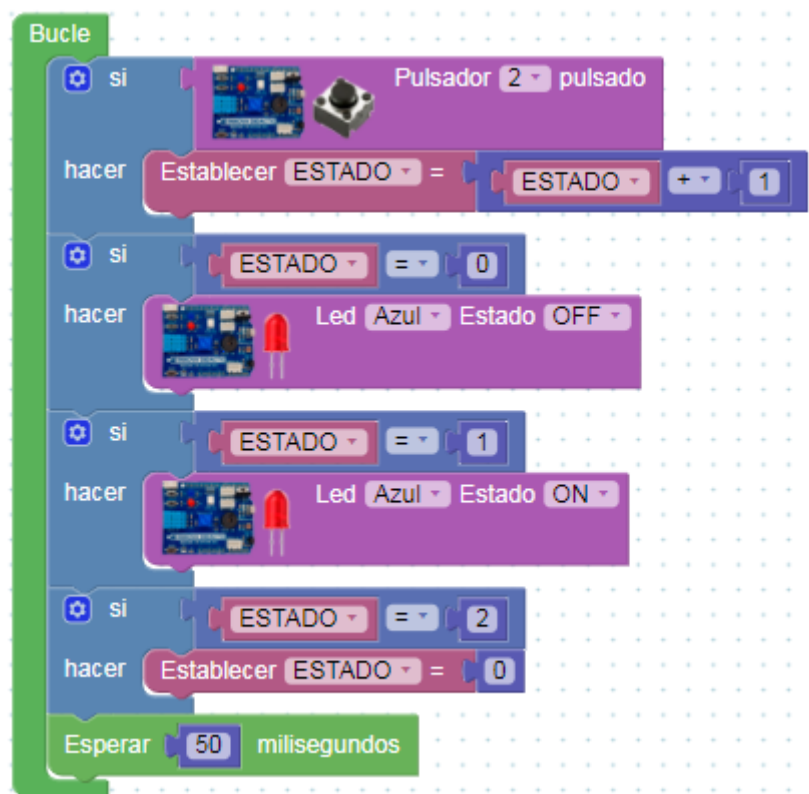
Este programa irá incrementando el valor indefinidamente cada vez que  apretemos el pulsador. Pero vamos a hacer que se reinicie la variable cada vez que llegue al valor 2, es decir, que vuelva a poner el valor de la variable a 0. De esa manera, cuando *Estado* tiene un valor de 0 el Led estará apagado, cuando *ESTADO* tenga un valor de 1 el Led estará encendido y cuando se vuelva a dar al pulsador el valor de *ESTADO* será 2 y se le mandará volver a un valor de 0 por lo que el Led estará apagado nuevamente. Este sería el programa:





Inicializar

Establecer ESTADO = 0



Actividad de ampliación: intenta hacer el mismo programa, pero más simplificado (utilizando menos instrucciones).

Revision #1

Created 1 June 2022 11:00:09 by Equipo CATEDU

Updated 1 June 2022 11:08:57 by Equipo CATEDU