

Introducción a ArduinoBlocks



4. Estructura de un proyecto

Club Robótica Granada

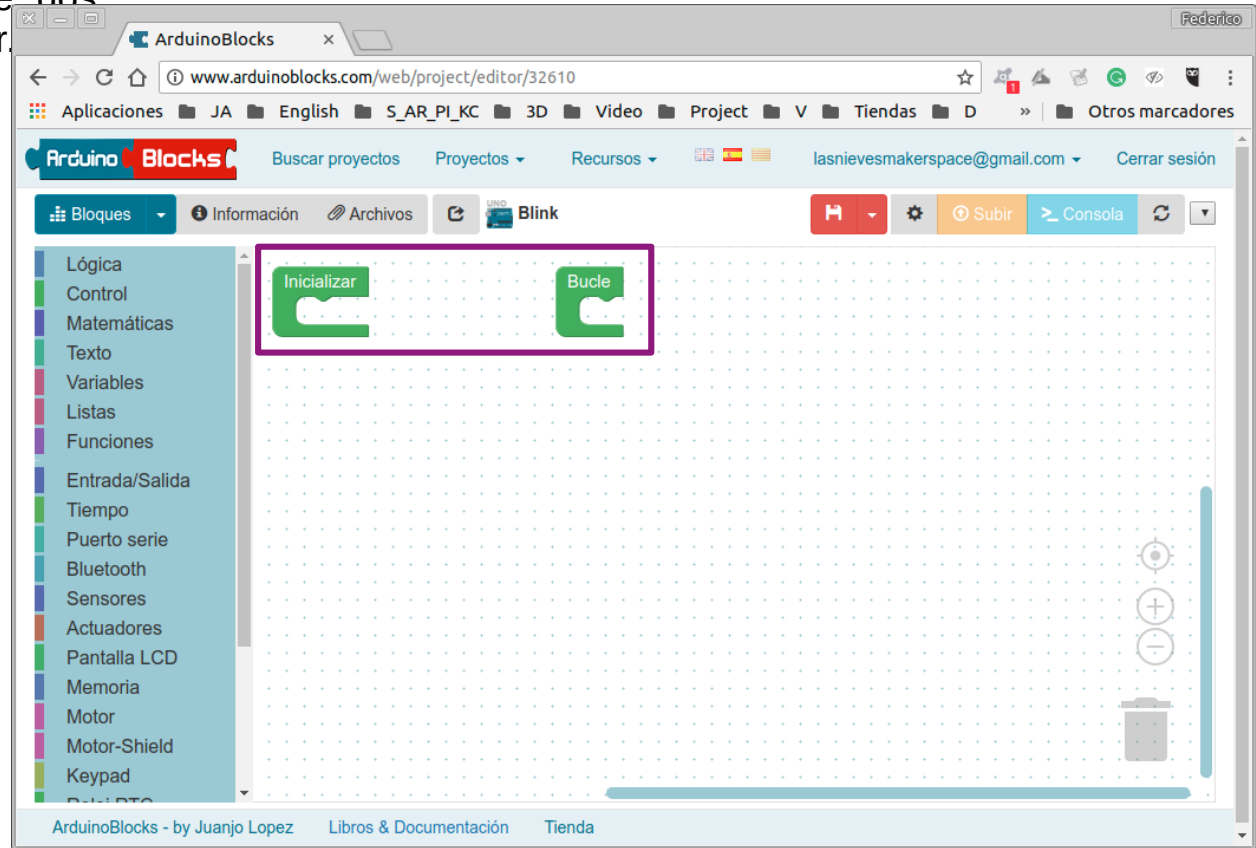
 @clubroboticagra

<https://clubroboticagranada.github.io>



Estructura de un proyecto

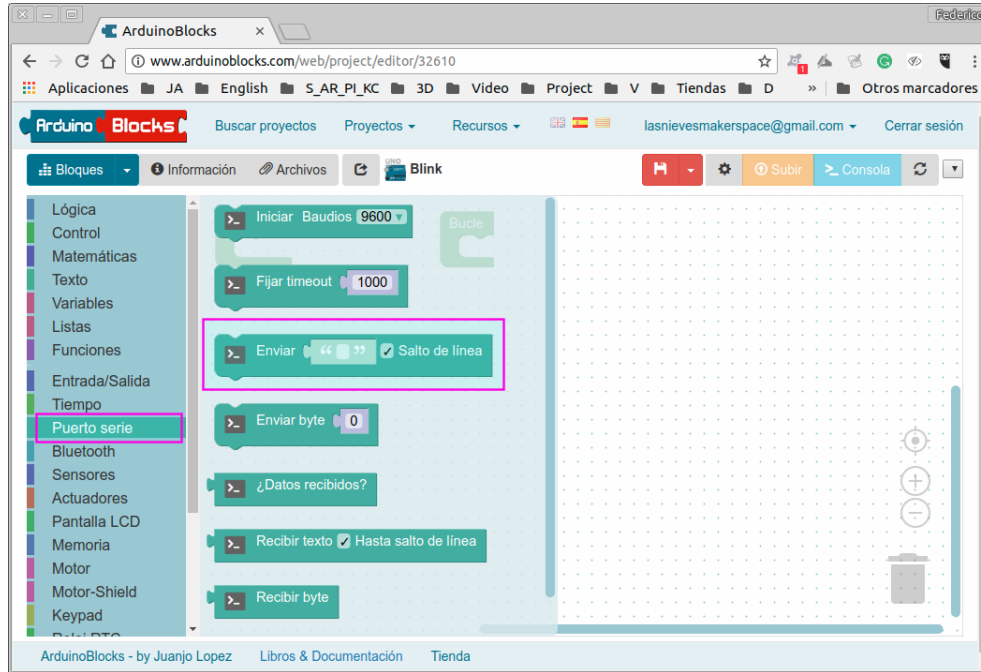
Un proyecto Arduino tiene siempre dos estructuras importantes en su interior.



Creamos un nuevo proyecto al que llamaremos Blink

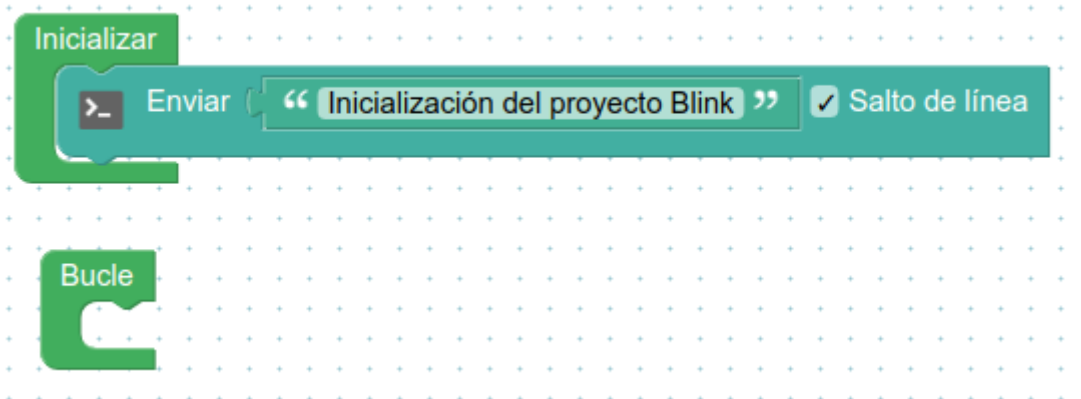
Estructura de un proyecto

Vamos a hacer que al iniciar (o hacer reset) se envíe el mensaje por el puerto serie “Inicialización del proyecto Blink”



Seleccionamos componente y lo arrastramos hasta el bloque.

Estructura de un proyecto



Situamos el componente y tecleamos el texto citado en el espacio habilitado para ello.

ArduinoBlocks connector

Es una aplicación puente entre la plataforma on-line ArduinoBlocks y el hardware Arduino.
La aplicación recibe el código generado por ArduinoBlocks, lo compila y lo sube a la placa Arduino.

- Está disponible para los principales sistemas operativos.
- En el área de descargas de arduinoblocks.com se puede obtener la última versión y más información sobre el proceso de instalación y configuración.

<http://www.arduinoblocks.com/web/site/abconnector>

ArduinoBlocks-Connector v3

La aplicación que conecta ArduinoBlocks a tu placa Arduino!



Windows Ubuntu 32 bits **Ubuntu 64 bits** Raspberry Pi MacOS

Ubuntu 64 bits

Probado en Ubuntu 64 bits (debian)



Linux

Descarga para Ubuntu 64 bits (.deb)

Método de instalación alternativo (terminal)

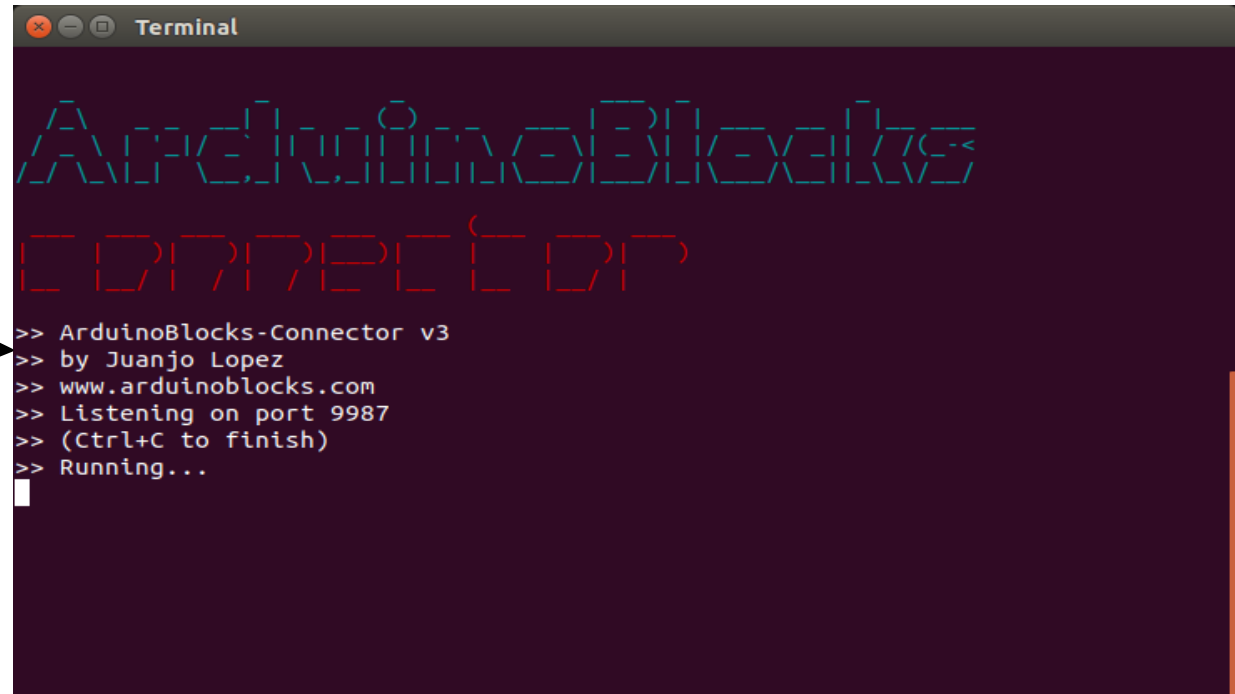
```
wget "http://www.arduinoblocks.com/abconnector/abconnector_3_x64.deb"
sudo dpkg -i abconnector_3_x64.deb
```

Método alternativo de ejecución (terminal)

```
/opt/abconnector/abconnector
```

ArduinoBlocks connector

Para poder grabar nuestro proyecto en la placa Arduino necesitamos activar Connector.

A terminal window titled "Terminal" with a dark background. It displays the ASCII art logo for "ArduinoBlocks" in green and "CONNECTOR" in red. Below the logo, it shows the version "v3" and the author "by Juanjo Lopez". It also indicates it is listening on port 9987 and is ready to run. A cursor is visible at the end of the "Running..." line.

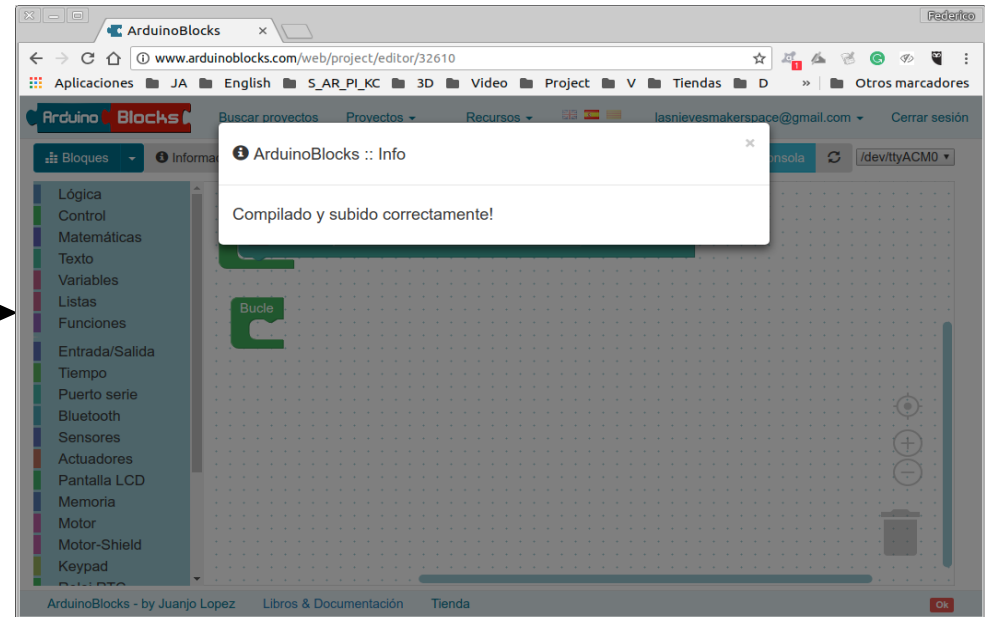
```
Terminal

  ArduinoBlocks
CONNECTOR

>> ArduinoBlocks-Connector v3
>> by Juanjo Lopez
>> www.arduinoBLOCKS.com
>> Listening on port 9987
>> (Ctrl+C to finish)
>> Running...
█
```

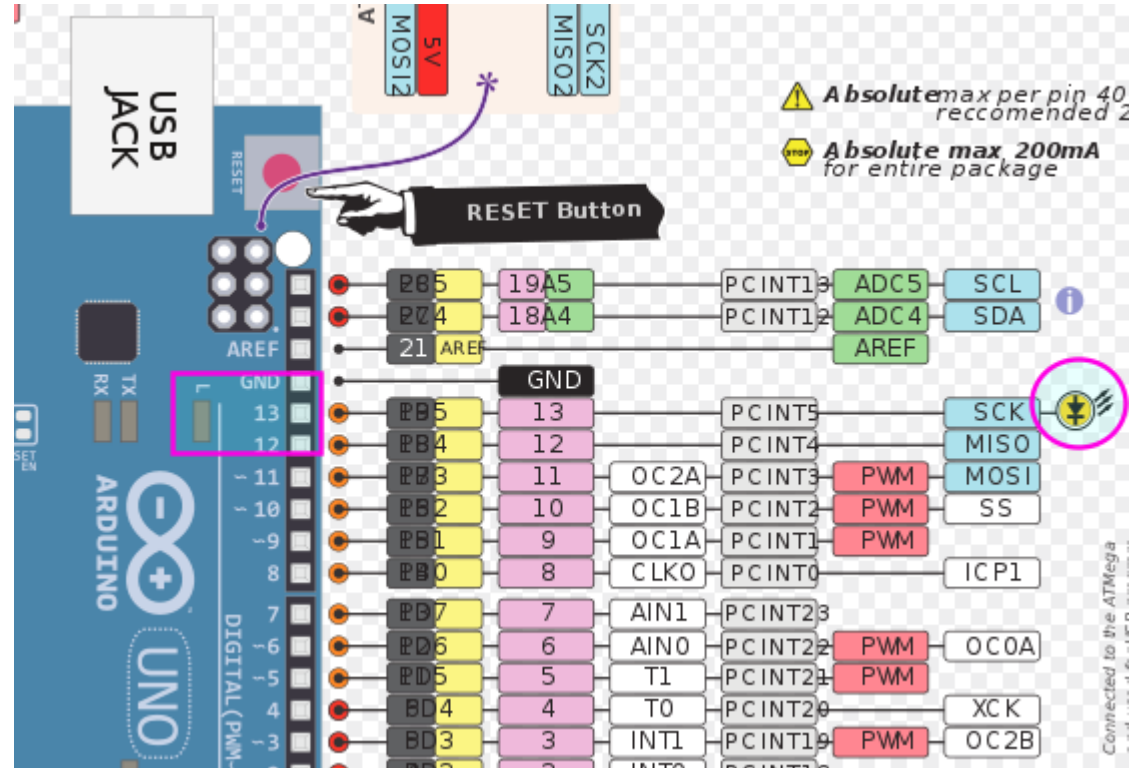
ArduinoBlocks connector

Ya podemos grabar el firmware directamente en nuestra placa a través de Connector.



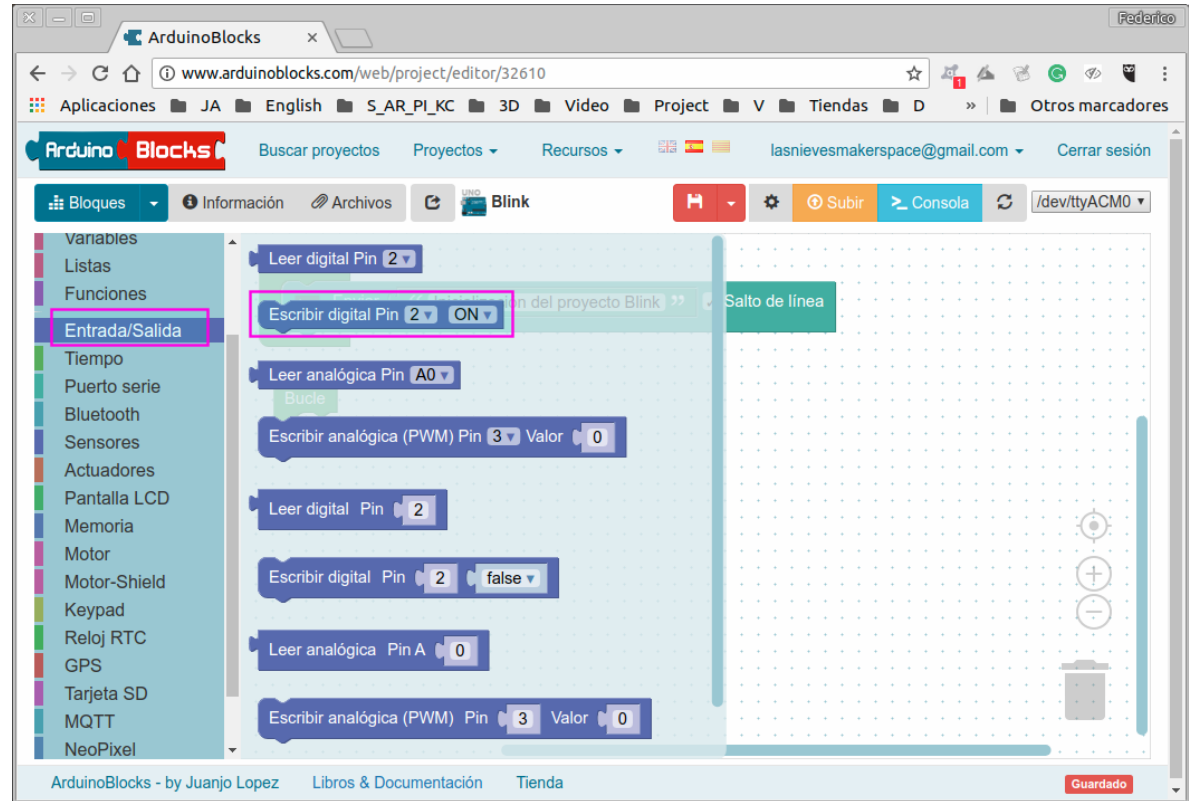
Blink

Consultando el pinout de la placa Arduino UNO vemos que en el pin 13 está conectado el LED L. Vamos a hacerlo parpadear.



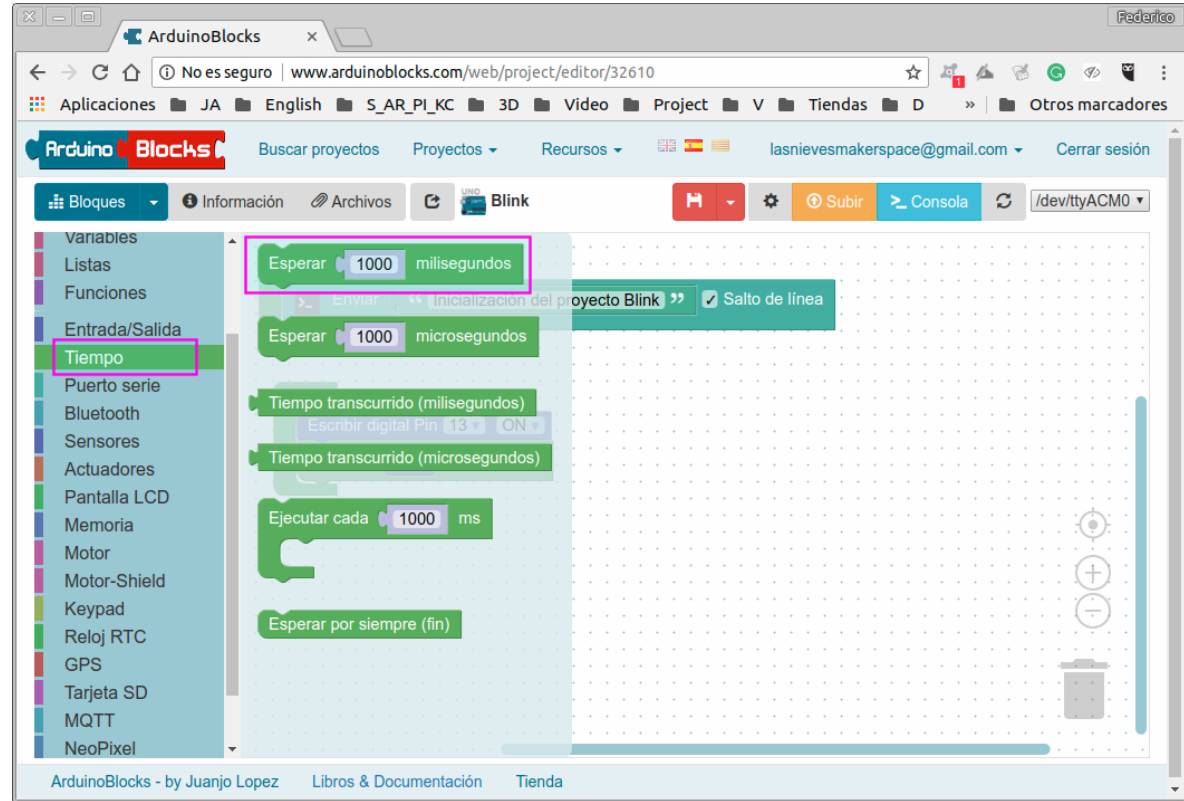
Blink

Seleccionamos componentes y arrastramos.



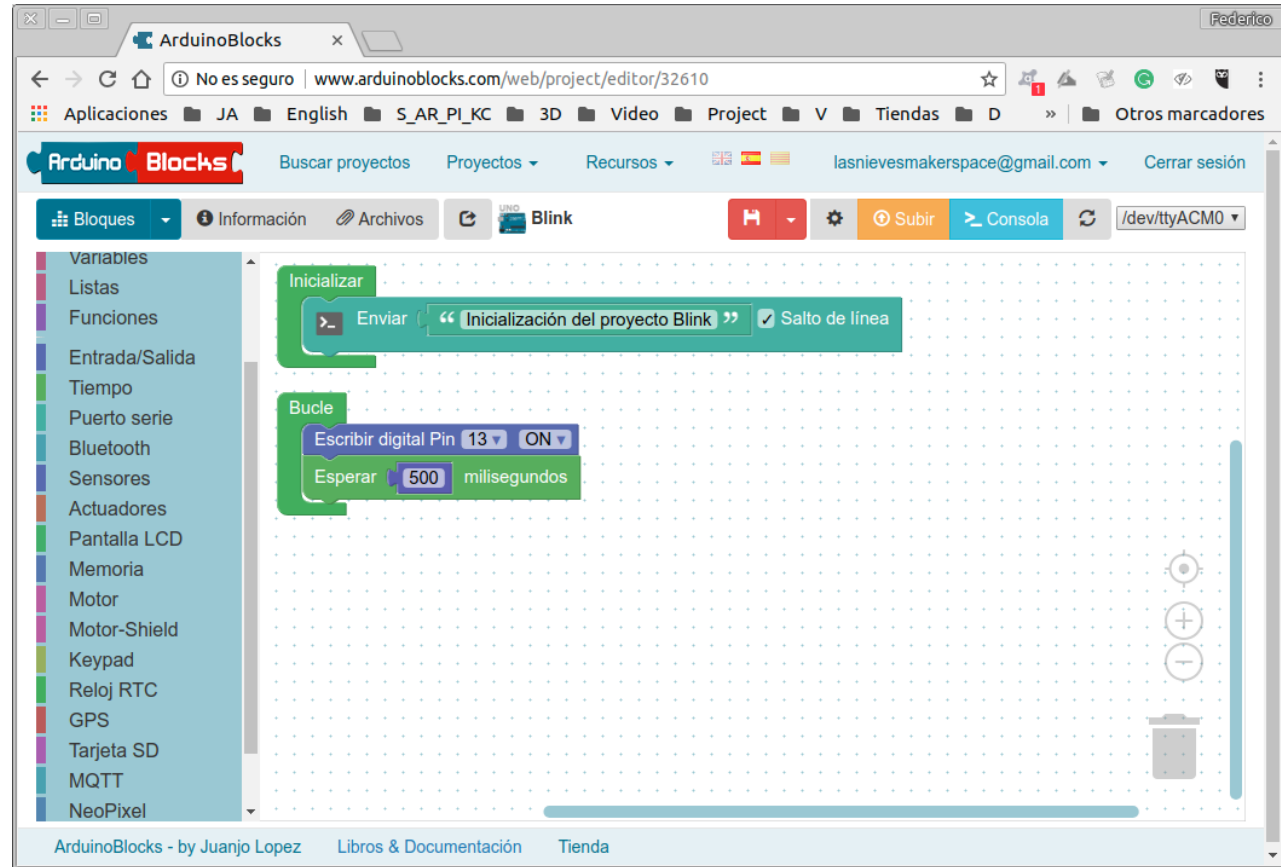
Blink

Seleccionamos componentes y arrastramos.



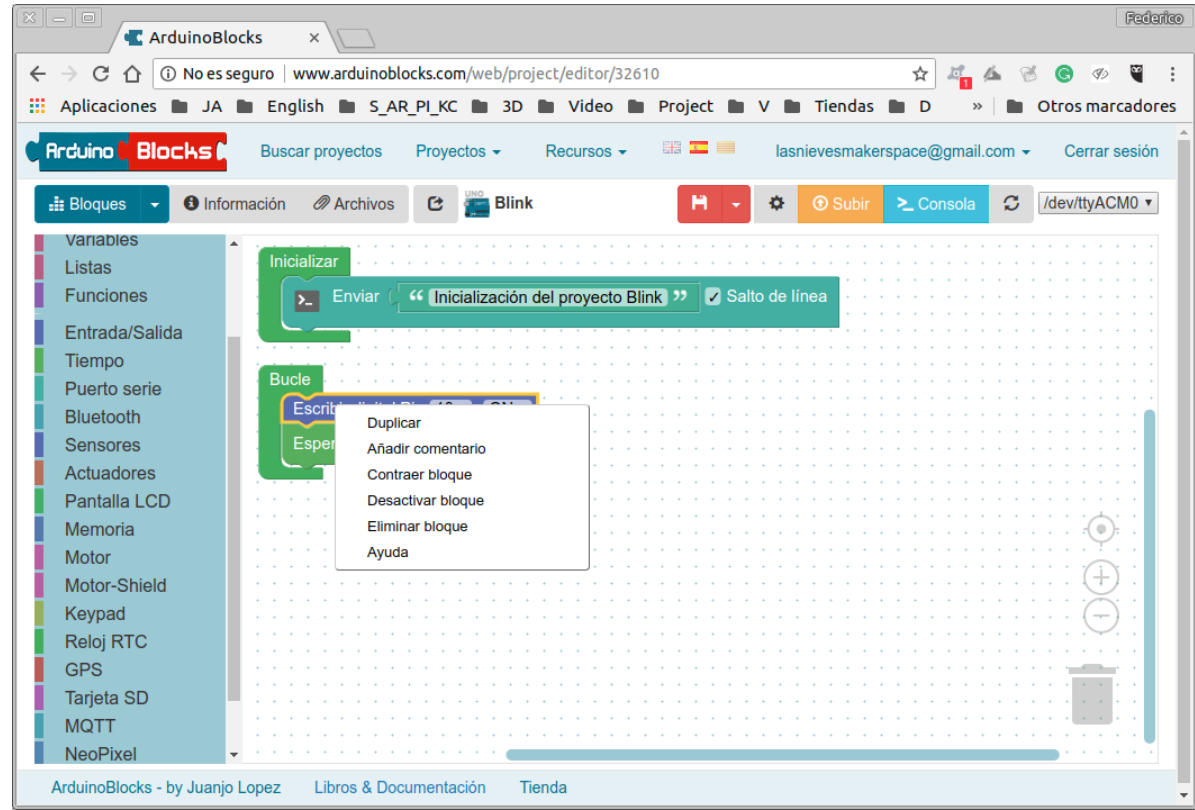
Blink

Configuramos los componentes.



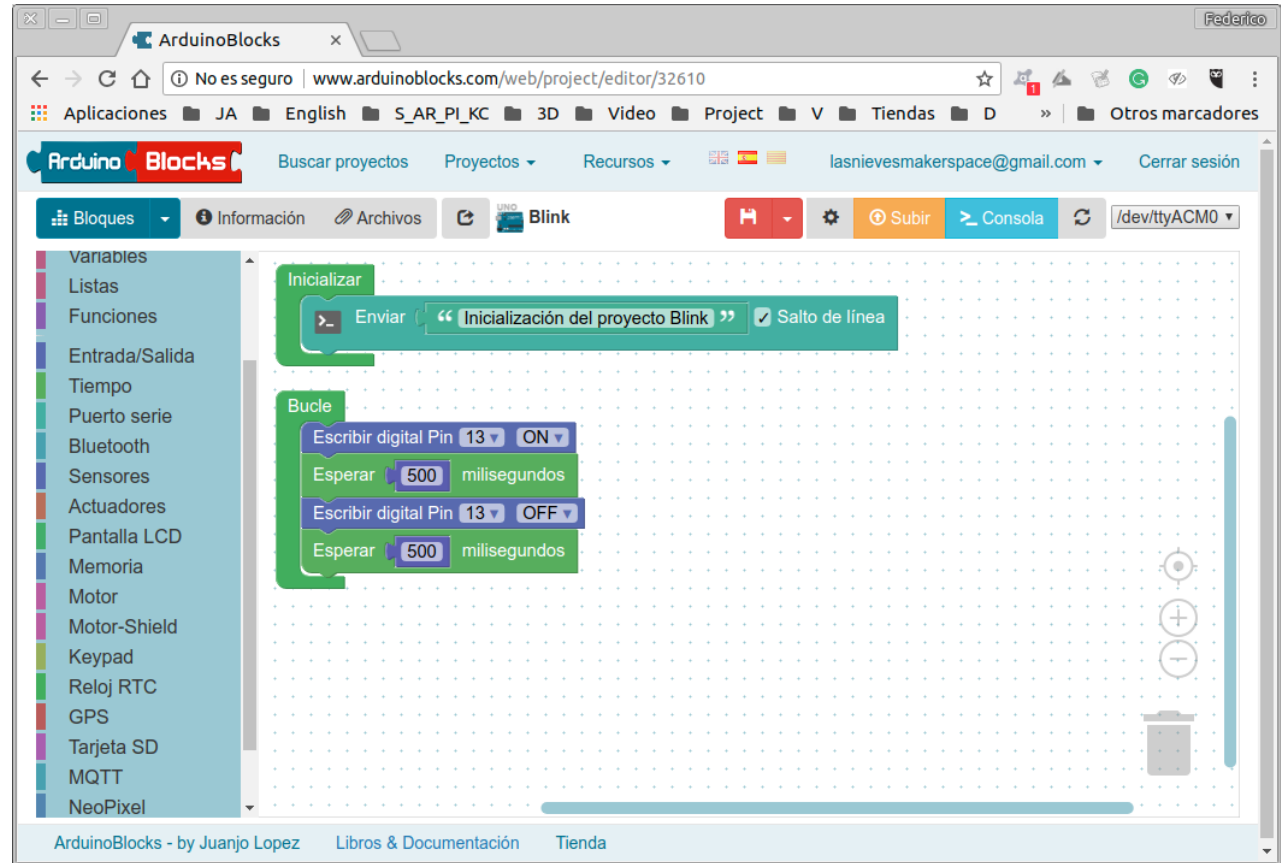
Blink

Duplicamos y configuramos los componentes.



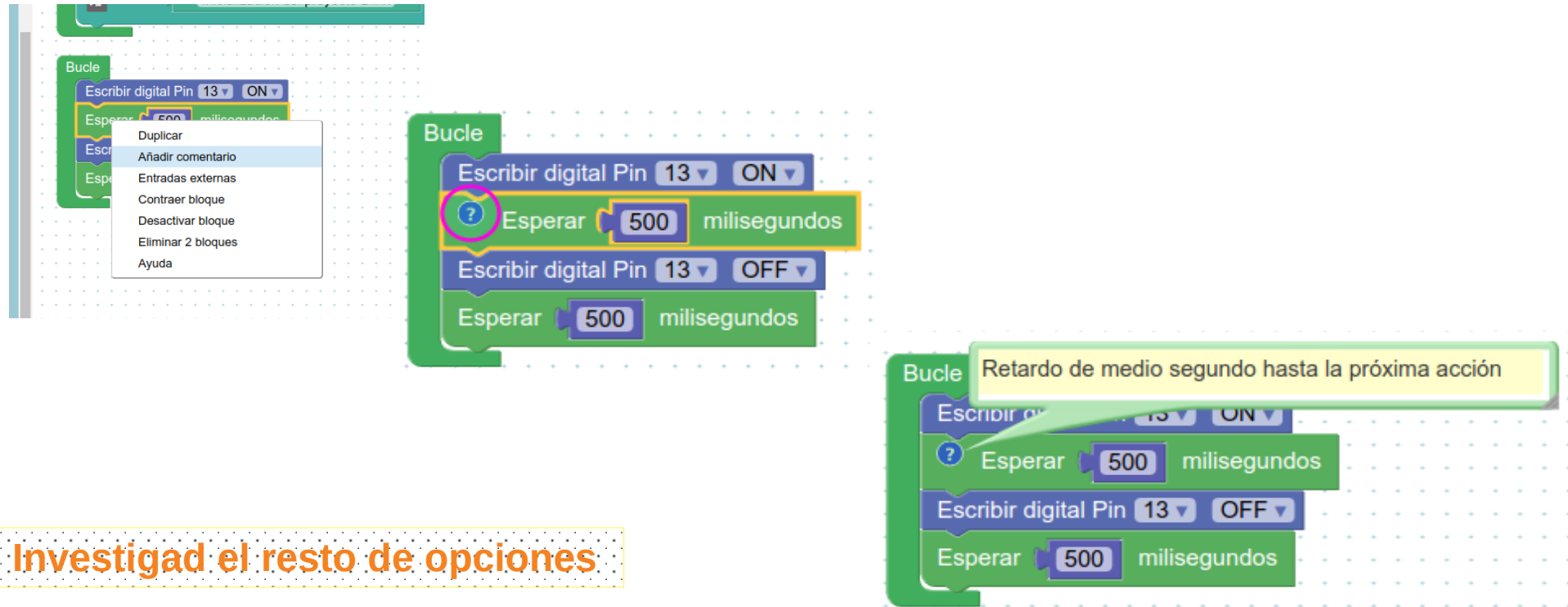
Blink

Aspecto final del ejemplo blink



Opciones botón secundario sobre bloque

Para añadir comentarios seleccionamos la opción, en el bloque aparece una interrogación, clicamos sobre ella y escribimos el comentario.



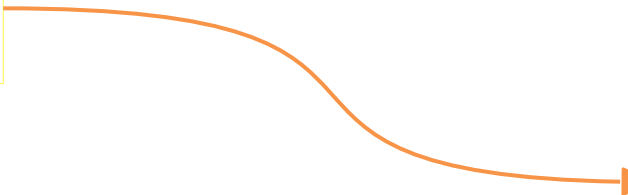
The image shows a sequence of steps in the Arduino IDE to add a comment to a block within a loop:

- Left Panel:** A 'Bucle' (Loop) block containing three sub-blocks: 'Escribir digital Pin 13 ON', 'Esperar 500 milisegundos', and 'Escribir digital Pin 13 OFF'. A context menu is open over the 'Esperar' block, with 'Añadir comentario' (Add comment) selected.
- Middle Panel:** The 'Esperar' block now has a question mark icon in its top-left corner, indicating that a comment can be added.
- Right Panel:** A callout box points to the question mark icon on the 'Esperar' block, containing the text: 'Retardo de medio segundo hasta la próxima acción'.

Investigad el resto de opciones

Análisis del código

Veamos el código por bloques y linea a línea



Código Arduino


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

  pinMode(13, OUTPUT);
  Serial.println(String("Inicialización del proyecto Blink"));
}
```

```
void loop()
{
  digitalWrite(13, HIGH);
  // Retardo de medio segundo hasta la próxima acción
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

Análisis del código

Se ejecuta una vez



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

  pinMode(13, OUTPUT);
  Serial.println(String("Inicialización del proyecto Blink"));
}
```

Inicializamos la consola con una velocidad de transmisión de 9600 baudios:

Serial.begin(9600);

Establecemos el pin 13 como salida:

pinMode(13, OUTPUT);

Mandamos el mensaje entrecomillado a la consola:

Serial.println(String("Inicialización del proyecto Blink"));

Análisis del código

Se ejecuta de forma indefinida



```
}  
  
void loop()  
{  
  
  digitalWrite(13, HIGH);  
  // Retardo de medio segundo hasta la próxima acción  
  delay(500);  
  digitalWrite(13, LOW);  
  delay(500);  
}
```

Establecemos un nivel alto en el pin 13 mediante escritura digital:

digitalWrite(13, HIGH);

Este es el comentario que introducimos en el primer bloque delay:

// Retardo de medio segundo hasta la próxima acción

Retardo de medio segundo:

Delay(500);

Establecemos un nivel bajo en el pin 13 mediante escritura digital:

digitalWrite(13, LOW);

Retardo de medio segundo:

delay(500);

Presentación disponible en:

<https://github.com/fgcoca/Chats/tree/master/21-11-19-Arduinoblocks>



Y ahora
EduBásica!!!

