

Programación Microcontrolador Arduino



C3.1 Arduino UNO salida digital

Arduino IDE y Arduino UNO, utilizando un diodo LED y una resistencia



Instrucciones

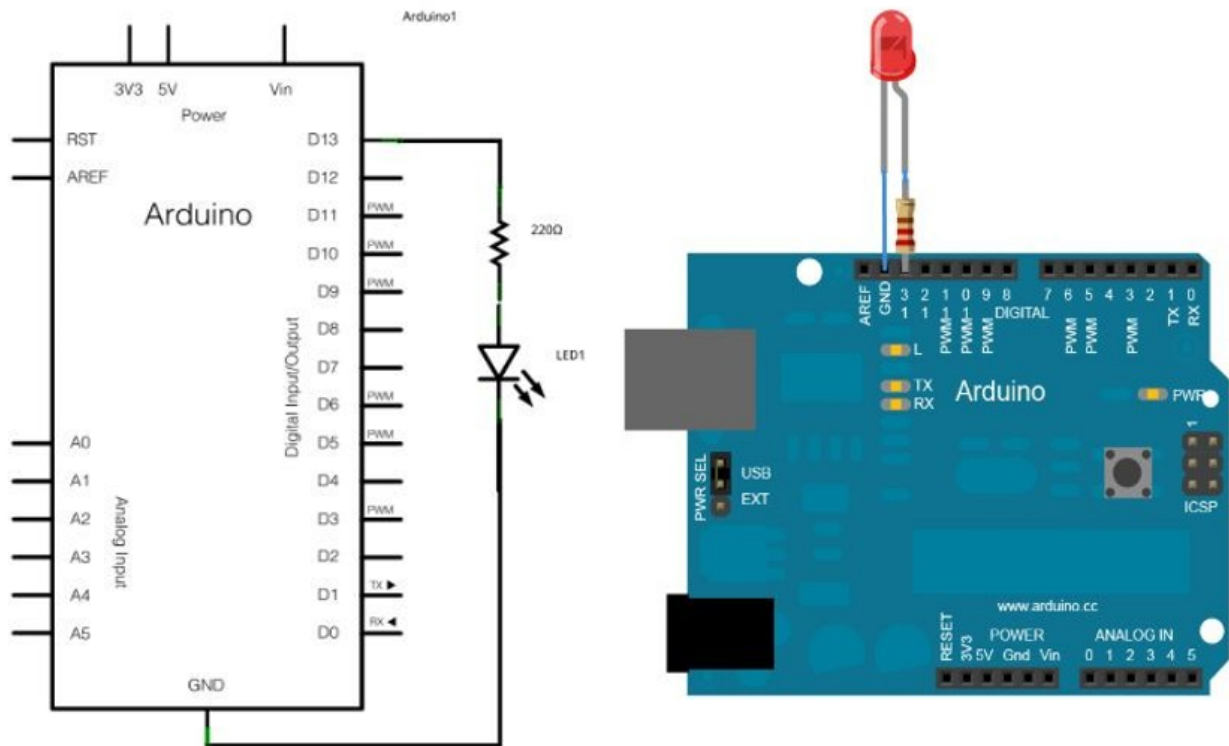
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado desarrollo.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo **Enlace a mi GitHub**
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo .md se debe exportar un archivo .pdf con la nomenclatura **C3.1_NombreAlumno_Equipo.pdf**, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma **oficial** aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
| readme.md
| | blog
| | | C3.1_TituloActividad.md
| | | C3.2_TituloActividad.md
| | | C3.3_TituloActividad.md
| | img
| | docs
| | | A3.1_TituloActividad.md
| | | A3.2_TituloActividad.md
```



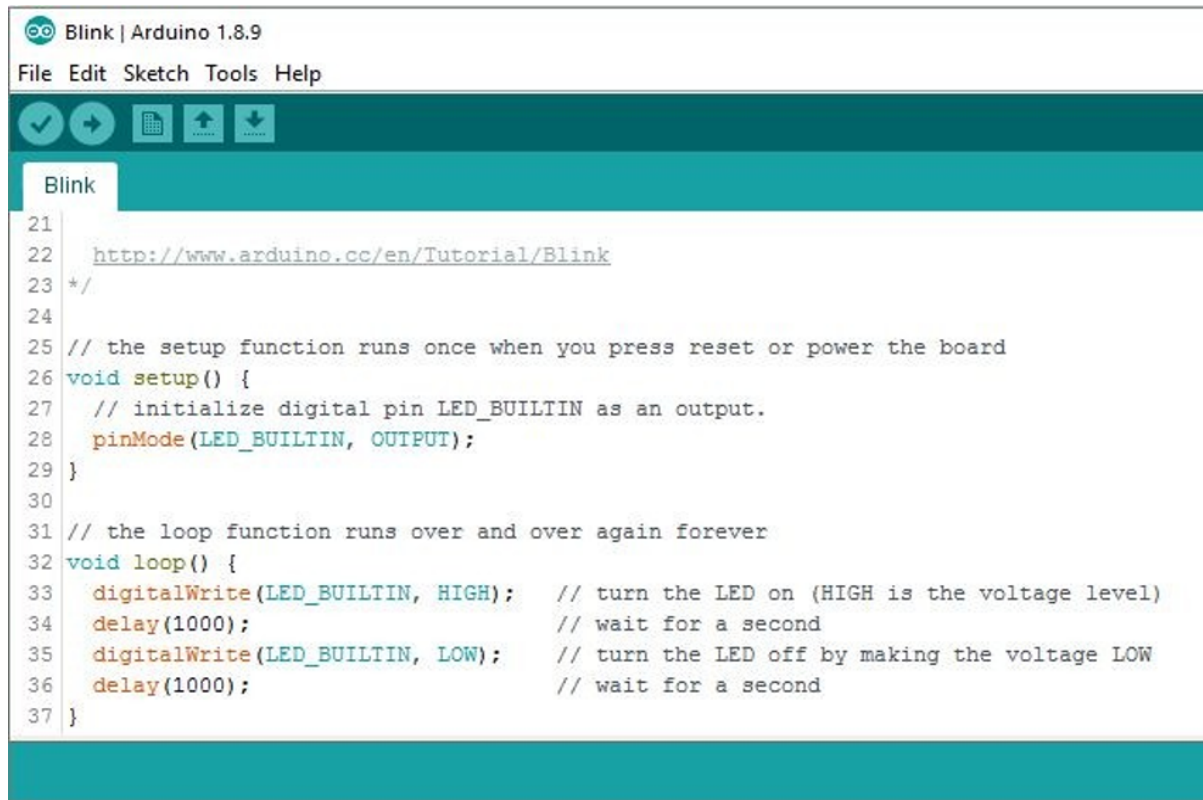
Desarrollo

1. Basado en el siguiente circuito, y utilizando uno de los simuladores propuestos, ensamblar lo que observa.



2. Analice y escriba el programa que se muestra a continuación.

- Como se ve el SETUP, se pone solo una vez al comenzar, ya que es el que se encarga de inicializar los modos de trabajo de los pins.
 - Luego vemos que dentro de los corchetes se pone el bloque de instrucciones, donde se ve la instrucción de PINMODE, que pide el pin y tipo, esta instrucción es para configurar el modo de trabajo.
 - Después se usa la función LOOP, se encarga de ejecutar en forma cíclica las instrucciones.
 - Dentro de esta función encontramos las instrucciones de DIGITALWRITE, que indica el led que se usara, y si es alto o bajo.
 - Finalmente el uso de DELAY detiene la ejecución del programa.

The image shows the Arduino IDE interface. At the top, the title bar says "Blink | Arduino 1.8.9". Below it is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Under the "Sketch" menu, there are icons for a checkmark, a right arrow, a document, an upload arrow, and a download arrow. The main text area shows the "Blink" sketch. It starts with a comment on line 21: "http://www.arduino.cc/en/Tutorial/Blink". Line 22 is a multi-line comment "*/". Line 24 is another comment: "// the setup function runs once when you press reset or power the board". Line 26 is the start of the "void setup() {" function. Line 27 is a comment: "// initialize digital pin LED_BUILTIN as an output.". Line 28 is the code "pinMode(LED_BUILTIN, OUTPUT);". Line 29 is the closing brace "}". Line 30 is a comment: "// the loop function runs over and over again forever". Line 32 is the start of the "void loop() {" function. Line 33 is the code "digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)". Line 34 is the code "delay(1000); // wait for a second". Line 35 is the code "digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW". Line 36 is the code "delay(1000); // wait for a second". Line 37 is the closing brace "}".

```
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
```

3. Explique el resultado observado.

- En cuanto corre la simulación el led se enciende y se apaga cada segundo, ya que se detiene cada 1000 ms.

4. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido.

The screenshot shows the Tinkercad web interface. At the top, the version 'c3.1' is displayed on the left, and a status message 'Se han guardado tus cambios' (Your changes have been saved) is on the right. Below the top bar is a toolbar with icons for undo, redo, delete, and view. The main workspace contains an Arduino Uno board connected to a breadboard. A red LED is connected to digital pin 13, and a 10k resistor is connected between pin 13 and ground. The code editor on the right contains the following code:

```

1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8
9   digitalWrite(13, HIGH);
10  delay(1000);
11  digitalWrite(13, LOW);
12  delay(1000);
13 }

```

At the bottom, there is a 'Monitor en serie' (Serial Monitor) button and a partially visible serial monitor window showing a scale from 0 to 2.