



INSTITUTO TECNOLÓGICO SUPERIOR “HUAQUILLAS”

Arduino

Manual Técnico

Tecnología

- | |
|---|
| <ul style="list-style-type: none">• Tecnología superior en redes y telecomunicaciones |
|---|

Autor:

- Jorge David Herrera Sarango

Huaquillas – Ecuador

2019

ÍNDICE DE CONTENIDO

INDICE DE FIGURAS	4
INDICE DE TABLAS	5
1.INTRODUCCIÓN	6
1.1. Objetivo general	7
1.2. Objetivos específicos	7
2. CONTENIDO TÉCNICO.....	8
2.1.¿Qué es Arduino?	8
2.2. Hardware y cable USB.....	8
2.2.1. Pines de alimentación.....	8
2.2.2. Entradas / salidas digitales	9
2.2.3. Entradas analógicas	10
2.3. Primeros pasos con Arduino (Instalación “IDE”)	11
2.4. Conexión de la placa Arduino	16
2.4.1. Instalación de librerías.....	16
2.4.2. Ejecución de Arduino, Selección de Placa y Puerto serial.....	18
2.4.3. Cargar el programa a la placa.	19
2.5. Estructura básica.	20
2.5.1. Función Setup()	20
2.5.2. Funcion Loop().....	21
2.6. Controles De Flujo Y Saltos Condicionales.....	21
3. EJEMPLO PRÁCTICO	24
3.1. Materiales	24
3.2. Código.....	24
3.2.1. Explicación de las líneas de código	25
3.3. Conexión del tablero Arduino nano	25

3.4. Cargar el código a nuestro Arduino.....	26
4.RESPONSABLES.	27
5.GLOSARIO.....	29
6.REFERENCIAS.....	30

INDICE DE FIGURAS

Figura 1. Pines de alimentación de Arduino Uno.....	8
Figura 2. Entradas / salidas digitales.....	9
Figura 3. Entradas analógicas.....	10
Figura 4. Página de descarga IDE.....	11
Figura 5. Aceptación de descarga.....	11
Figura 6. Ejecución del instalador.....	12
Figura 7. Componentes a instalar.....	12
Figura 8. Confirmar la instalación.....	13
Figura 9. Proceso de instalación.....	13
Figura 10. Instalación de controladores.....	14
Figura 11. Instalación completada del IDE Arduino.....	14
Figura 12. Solicitud de permisos.....	15
Figura 13. Interfaz del IDE de Arduino.....	15
Figura 14. Conexión de Arduino a la computadora.....	16
Figura 15. Incluir librerías en Arduino.....	16
Figura 16. Librerías de Arduino.....	17
Figura 17. Añadir zip. (librerías).....	17
Figura 18. Selección de la placa.....	18
Figura 19. Selección del puerto.....	18
Figura 20. Código de programación.....	19
Figura 21. Cargar el código al tablero.....	19
Figura 22. Estructura básica de un programa en Arduino.....	20
Figura 23. descripción de la función Setup.....	20
Figura 24. Descripción de la función Loop.....	21
Figura 25. Materiales.....	24

Figura 26. Código para subir al Arduino	24
Figura 27. Setup del código.....	25
Figura 28. Void loop del código	25
Figura 29. Conexión en la placa Arduino a los componentes.....	25
Figura 30. Conexión a la computadora.....	26
Figura 31. Prueba de funcionamiento del parpadeo de la luz led.....	26

INDICE DE TABLAS

Tabla 1, Tipos de placas Arduino.....	22
--	----

1.INTRODUCCIÓN

La plataforma Arduino es una compañía de hardware libre y comunidad tecnológica, que diseña y manufactura placas de desarrollo de hardware y software compuesta respectivamente por circuitos impresos que integran un microcontrolador, y un entorno de desarrollo (IDE) en donde se programa cada placa.

Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas dirigidos a proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software son liberados bajo licencia de código abierto que permite libertad de acceso a los mismos. Por tal motivo es importante aprender a manejar tanto software como hardware, con el fin de explotar todos los beneficios que nos da esta herramienta.

Tomando en cuenta lo mencionado anteriormente se presenta un manual técnico, con el fin de que todos logren tener un correcto manejo y desenvolvimiento de la plataforma Arduino.

1.1. Objetivo general

Explicar el correcto manejo la plataforma Arduino, mediante la explicación de sus características y pines, por medio de un manual técnico el cual sirva de apoyo o guía a las personas que desean desarrollar proyectos de manera casera.

1.2. Objetivos específicos

- Investigar sobre la plataforma Arduino y explicarla de manera sencilla en el manual técnico.
- Identificar el uso correcto de la plataforma Arduino y realizar un ejercicio práctico - demostrativo.
- Realizar un manual técnico para las personas que están iniciando en el uso y manejo de la plataforma Arduino.

2. CONTENIDO TÉCNICO

2.1. ¿Qué es Arduino?

Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software (programas) para la placa. Puedes usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos con Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en tu ordenador. La placa puedes montarla tú mismo o comprarla ya lista para usar, y el software de desarrollo es abierto y lo puedes descargar gratis desde la página www.arduino.cc/en/. (Benavides, 2016)

“El Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La Fuente de alimentación se selecciona automáticamente como se muestra a continuación en la figura1” (Benavides, 2016).

2.2. Hardware y cable USB

2.2.1. Pines de alimentación



Figura 1. Pines de alimentación de Arduino Uno
Fuente: (Benavides, 2016)

De acuerdo con Benavides (2016), “Arduino se alimenta de energía mediante la conexión USB o mediante una fuente externa (recomendada de 7-12V), vamos a tener unas salidas de tensión continua debido a unos reguladores de tensión y condensadores de estabilización”.

Estos pines son:

- **VIN:** se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa.
- **5V:** fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada.
- **3.3V:** fuente de 3.3 voltios generados por el regulador interno con un consumo máximo de corriente de 50mA.
- **GND:** pines de tierra.

2.2.2. Entradas / salidas digitales



Figura 2. Entradas / salidas digitales
Fuente: (Benavides, 2016)

Según (Benavides, 2016), Como se muestra en la figura 2. Cada uno de los 14 pines digitales se puede utilizar como una entrada o salida. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas como:

- Pin 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL.
- Pin 2 y 3. Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario.
- Pin 3, 5, 6, 9, 10 y 11. PWM (modulación por ancho de pulso). Constituyen 8 bits de salida PWM con la función `analogWrite()`.
- Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines son de apoyo a la comunicación SPI.
- Pin 13. LED. Hay un LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el valor está bajo, es apagado.

2.2.3. Entradas analógicas

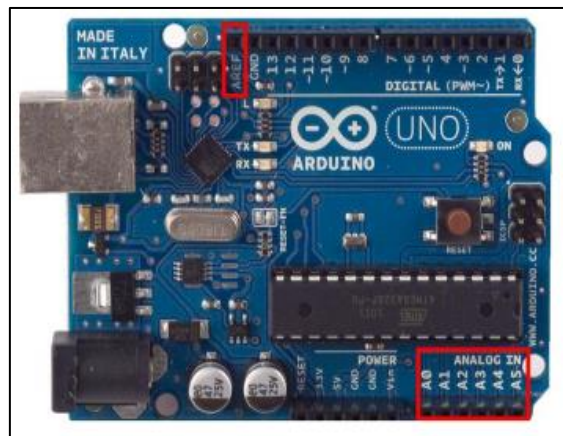


Figura 3. Entradas analógicas
Fuente: (Benavides, 2016)

Según (Benavides, 2016), El Arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, cada una de las cuales ofrecen 10 bits de resolución (es decir, 1024 estados). Por defecto, tenemos una tensión de 5V, pero podemos cambiar este rango utilizando el pin de AREF y utilizando la función `analogReference()`, donde le introducimos una señal externa de continua que la utilizara como referencia tal como se muestra en la figura 3.

2.3. Primeros pasos con Arduino (Instalación “IDE”)

- Se debe descargar el instalador desde la página oficial: www.arduino.cc/en/. Para Windows, tal como se muestra en la figura 4 y 5.



Figura 4. Página de descarga IDE
Elaborado por: Los autores



Figura 5. Aceptación de descarga
Elaborado por: Los autores

- Ejecutamos el instalador y se nos presentara una ventana donde le daremos en permitir modificaciones, posteriormente se nos presentara otra ventana donde le daremos en I Agree, tal como se muestra en la figura 6

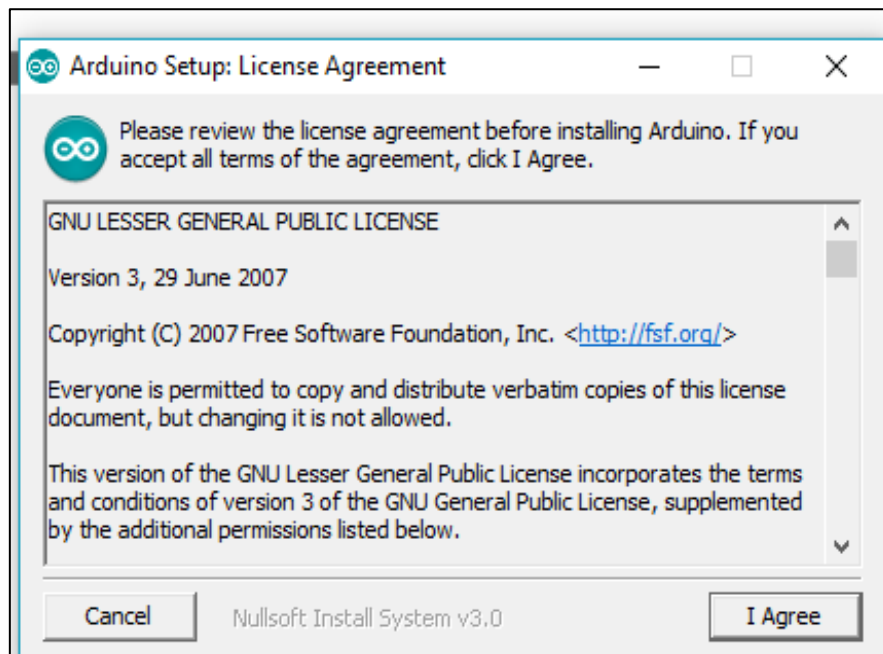


Figura 6. Ejecución del instalador
Elaborado por: Los autores

- De acuerdo a la figura 7. A continuación, le damos en next.

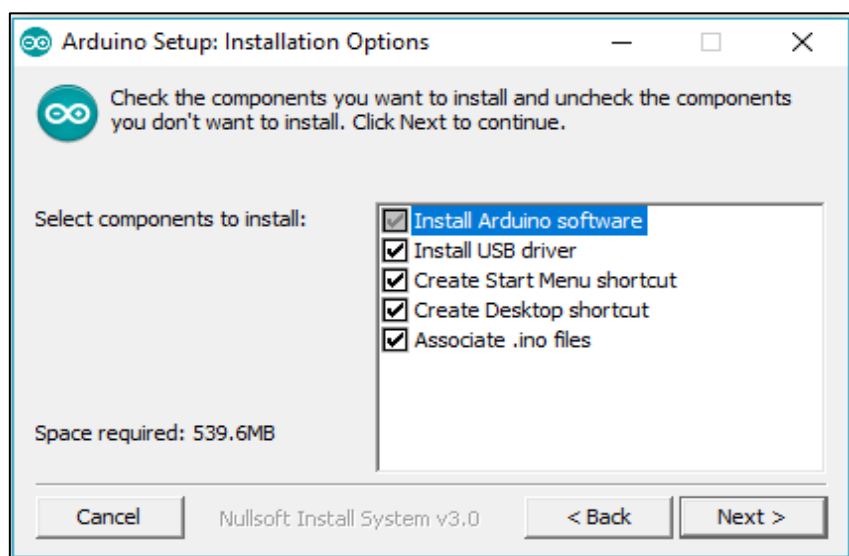


Figura 7. Componentes a instalar
Elaborado por: Los autores

- Y procedemos a darle a Install e inmediatamente empezará a cargar tal como se puede apreciar en la figura 8 y 9.

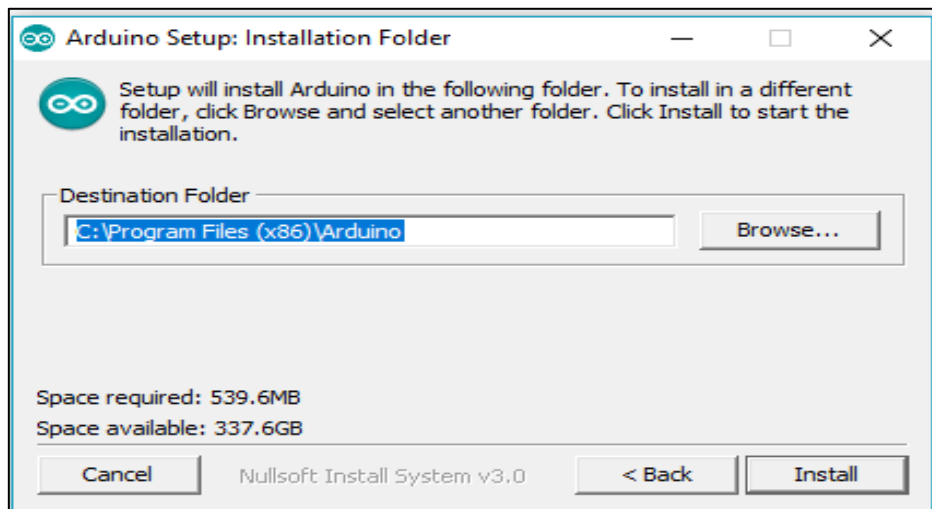


Figura 8. Confirmar la instalación
Elaborado por: Los autores

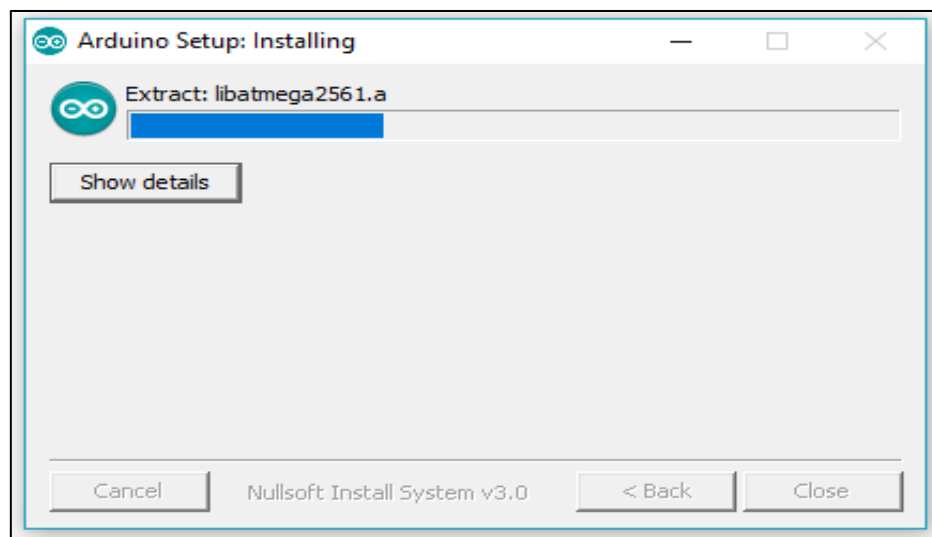


Figura 9. Proceso de instalación
Elaborado por: Los autores

- Tal como se muestra en la figura 10. En transcurso de que cargue se nos desplegara una ventana que no pide permiso para instalar los controladores y le damos en instalar.

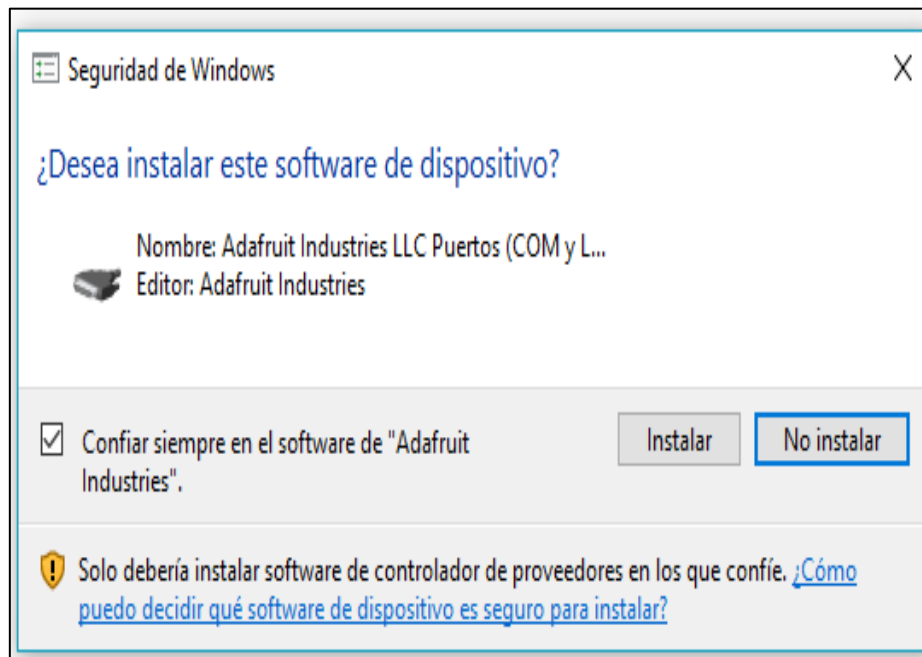


Figura 10. Instalación de controladores

Elaborado por: Los autores

- Al culminar le daremos en close e inmediatamente se nos levantara otra ventana donde le daremos en permitir el acceso tal como se muestra en la figura 11 y 12.

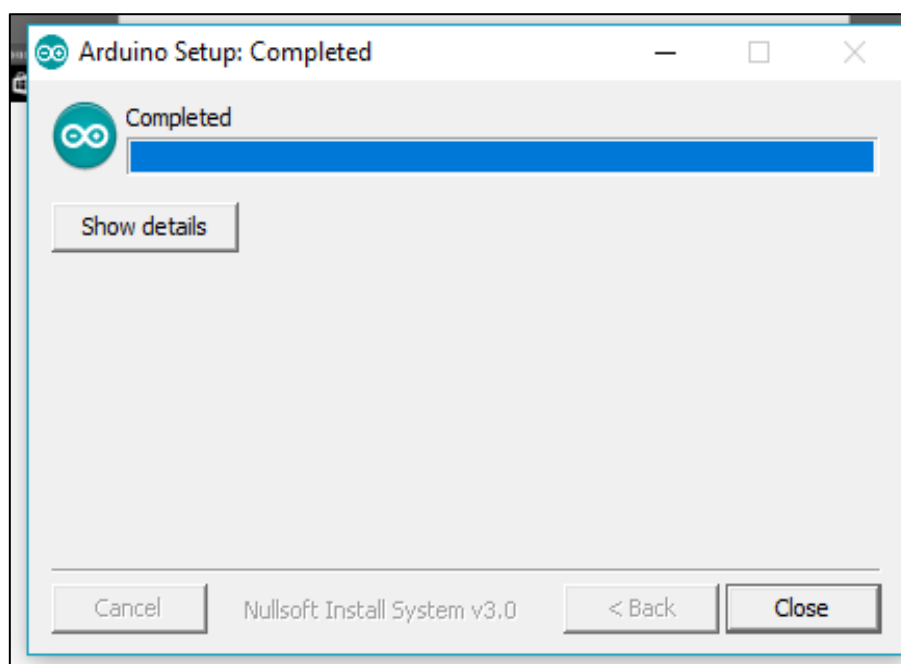


Figura 11. Instalación completada del IDE Arduino

Elaborado por: Los autores

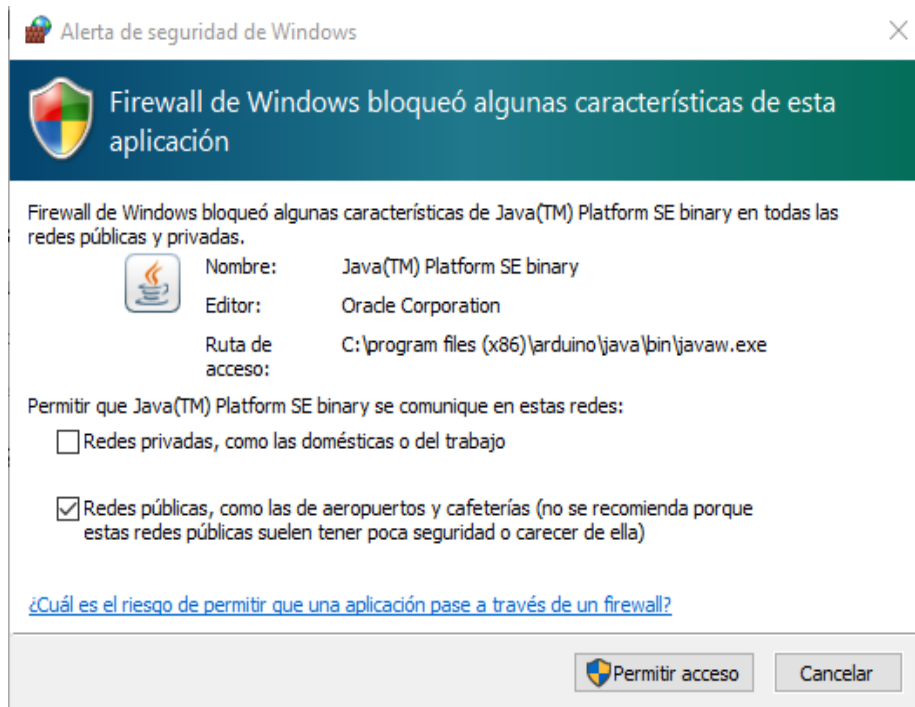


Figura 12. Solicitud de permisos
Elaborado por: Los autores

- De acuerdo a la figura 13. ya tendríamos instalado nuestro software de Arduino

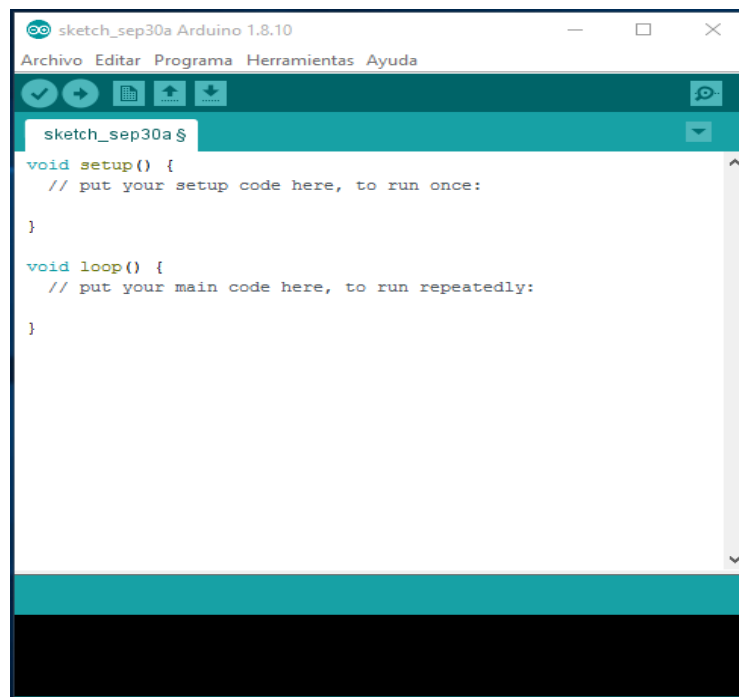


Figura 13. Interfaz del IDE de Arduino
Elaborado por: Los autores

2.4. Conexión de la placa Arduino

Conectamos la placa Arduino al ordenador usando el cable USB, una vez conectada el led de la placa PWR (led de alimentación) deberá permanecer encendido a partir de ahora tal como se muestra en la figura 14.

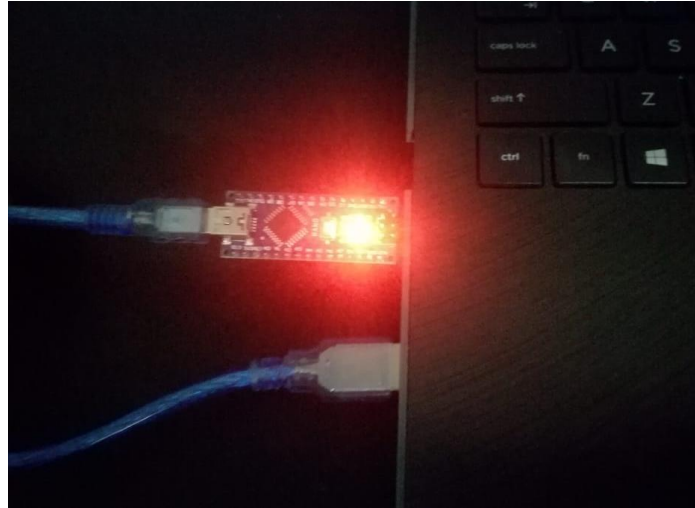


Figura 14. Conexión de Arduino a la computadora
Elaborado por: Los autores

2.4.1. Instalación de librerías

- Al conectar el Arduino, Windows automáticamente deberá de inicializar la instalación de los drivers.
- En caso de necesitar librerías específicas nos podemos dirigir a programa, incluir librería y en administrar biblioteca posteriormente se nos presenta una ventana tal como se puede ver en la figura 15. donde podemos encontrar todas las librerías y una barra donde podemos buscar alguna librería en específico.

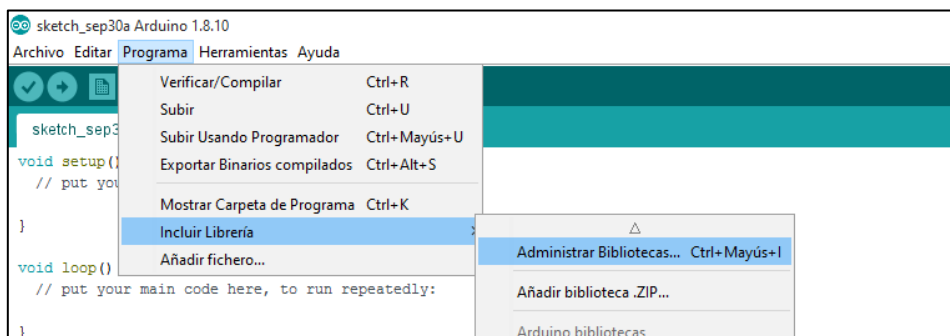


Figura 15. Incluir librerías en Arduino
Elaborado por: Los autores

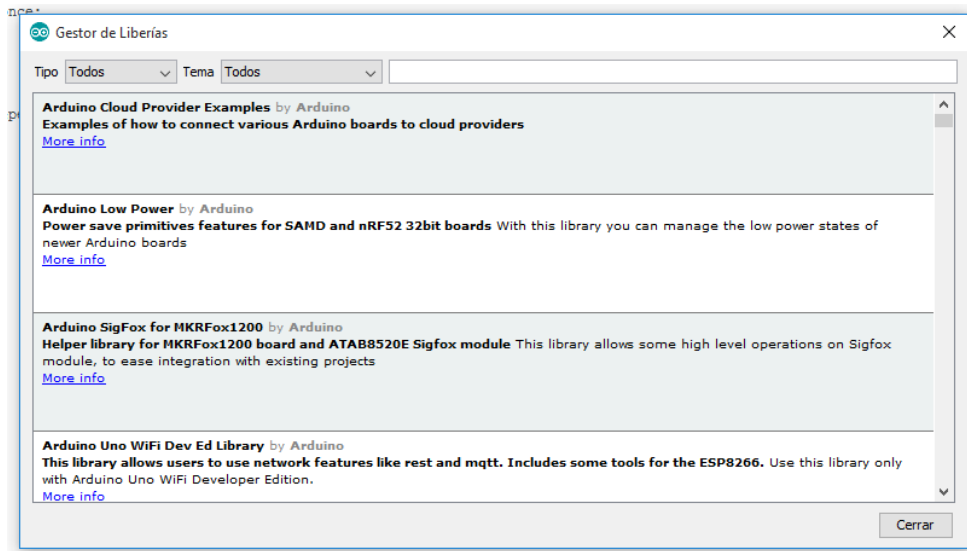


Figura 16. Librerías de Arduino
Elaborado por: Los autores

- Según la figura 16. De ser el caso y contar con drivers que se descarguen desde la web en forma de archivo comprimido lo podemos incluir a nuestra librería de Arduino dirigiéndonos a programa, incluir librería y en añadir biblioteca Zip.

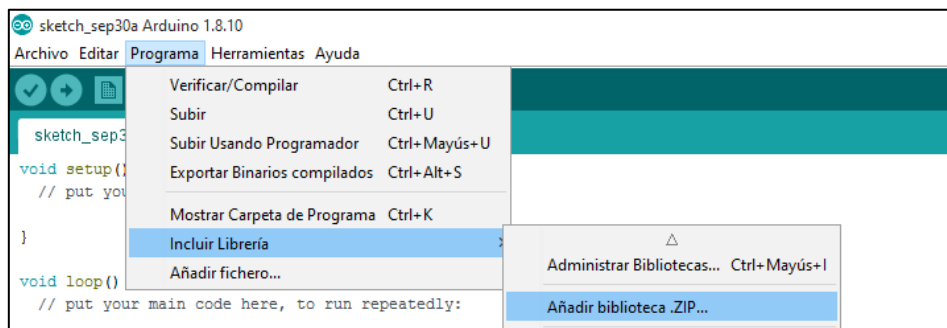


Figura 17. Añadir zip. (librerías)
Elaborado por: Los autores

2.4.2. Ejecución de Arduino, Selección de Placa y Puerto serial.

Una vez abierta la aplicación nos vamos a herramientas ► placa ► “seleccionamos la placa a usar”, tal como se muestra en la figura 18.

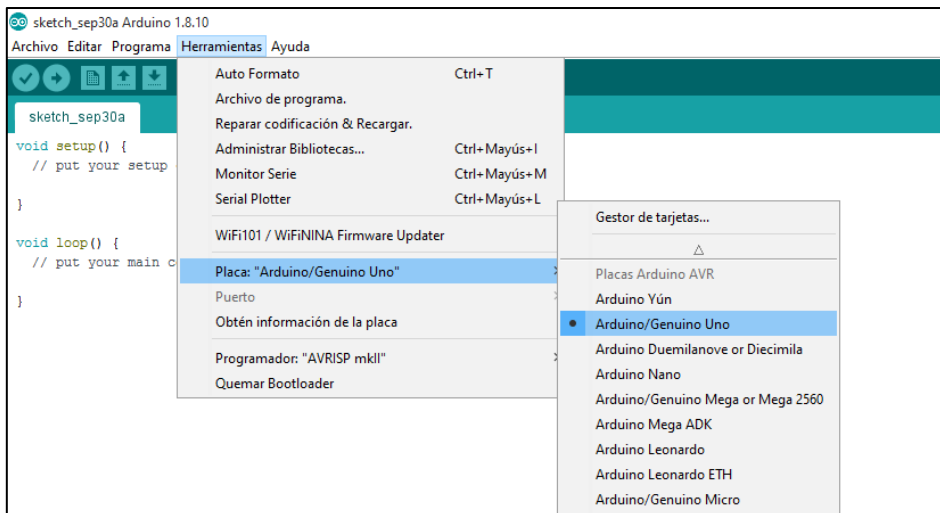


Figura 18. Selección de la placa
Elaborado por: Los autores

- Según la figura 19. Una vez seleccionado el modelo de nuestra placa tendremos que seleccionar el dispositivo serie de la placa:

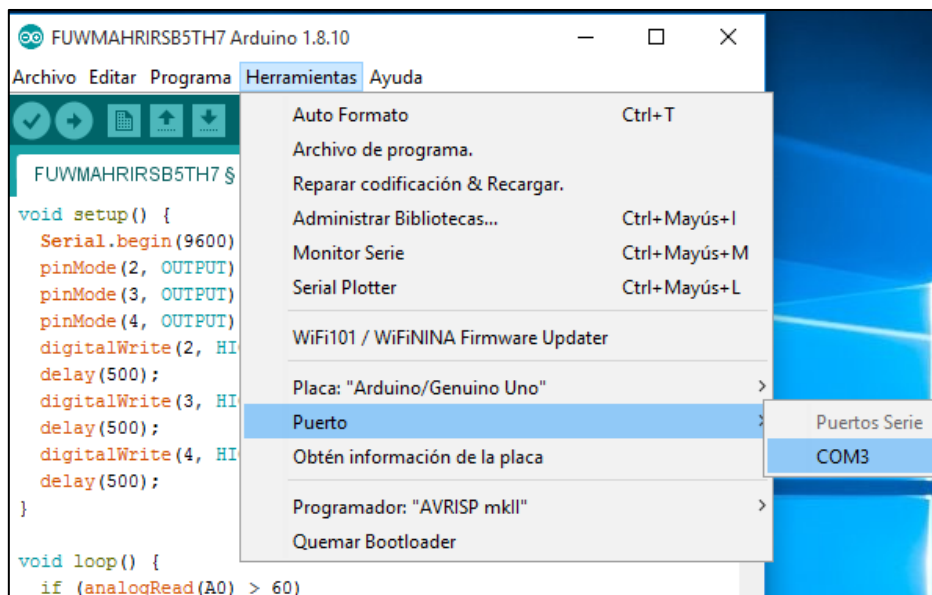


Figura 19. Selección del puerto
Elaborado por: Los autores

- Una vez que tenemos configurada nuestra placa Arduino al ordenador, vamos a agregar nuestro código tal como se muestra en la figura 20.



```
FUWMAHRIRSB5TH7 $
void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(3, HIGH);
  delay(500);
  digitalWrite(4, HIGH);
  delay(500);
}

void loop() {
  if (analogRead(A0) > 60)
  {
    digitalWrite(4, HIGH);
  }
  else
  {
    digitalWrite(4, LOW);
  }
}
```

Compilado

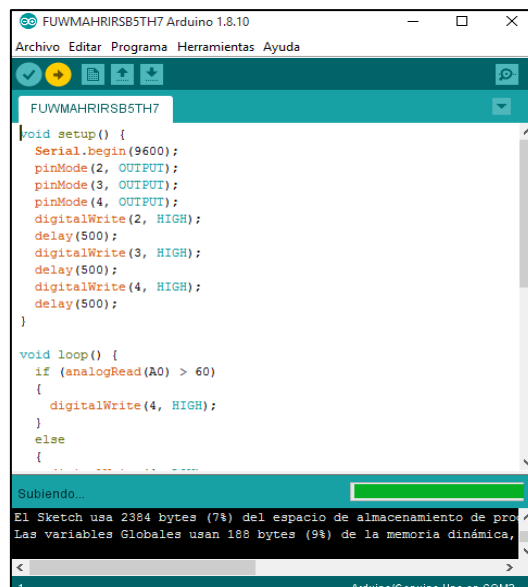
El Sketch usa 2384 bytes (7%) del espacio de almacenamiento de pro
Las variables Globales usan 188 bytes (9%) de la memoria dinámica,

Arduino/Genuino Uno

Figura 20. Código de programación
Elaborado por: Los autores

2.4.3. Cargar el programa a la placa.

Una vez que tenemos desarrollado el programa completo para cargarlo en el Arduino solo tenemos que presionar la flecha en dirección a la derecha que es de subir:



```
FUWMAHRIRSB5TH7
void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(3, HIGH);
  delay(500);
  digitalWrite(4, HIGH);
  delay(500);
}

void loop() {
  if (analogRead(A0) > 60)
  {
    digitalWrite(4, HIGH);
  }
  else
  {
    digitalWrite(4, LOW);
  }
}
```

Subiendo...

El Sketch usa 2384 bytes (7%) del espacio de almacenamiento de pro
Las variables Globales usan 188 bytes (9%) de la memoria dinámica,

Arduino/Genuino Uno en COM3

Figura 21. Cargar el código al tablero
Elaborado por: Los autores

2.5. Estructura básica.

La estructura de todo programa de Arduino cuenta con dos subrutinas elementales denominadas **void setup()** y **void loop()**, en ellas se encierran bloques de código donde se hacen declaraciones e instrucciones lógicas.

La subrutina **setup()** es la encargada de recoger la configuración y parametrizaciones estáticas del programa mientras que **loop()** es la que contienen las sentencias lógicas del programa. (Leonardo, 2017)

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Figura 22. Estructura básica de un programa en Arduino
Elaborado por: Los autores

2.5.1. Función Setup()

Según (Leonardo, 2017), Se invoca una sola vez al comienzo del programa, en ella se setean el funcionamiento de los pines, si serán de entrada o salida y que pines se utilizarán en el proyecto y muchas más cuestiones.

```
void setup() {  
  // set the digital pin as output:  
  pinMode(ledPin, OUTPUT);  
}
```

Figura 23. descripción de la función Setup
Elaborado por: Los autores

2.5.2. Funcion Loop()

Según (Veloso, 2016), Luego de ser invocada la función **setup()** se ejecuta la función **loop()**, en forma cíclica conteniendo toda la lógica del programa en ejecución, todo las acciones deben ser programadas dentro de esta función.

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```




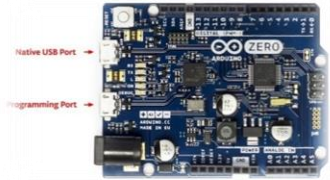


Figura 24. Descripción de la función Loop
Elaborado por: Los autores


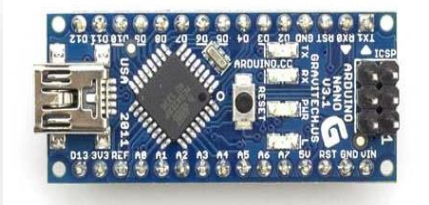


2.6. Controles De Flujo Y Saltos Condicionales

Según (Benavides, 2016), Como todo lenguaje de programación, existen saltos condicionales que se pueden utilizar según sea el objetivo que se necesita cumplir, no es el objetivo de este artículo comentar para que sirve cada uno, asumimos que el lector tiene un grado de conocimientos mínimos de lenguajes de programación. Al igual que en C o C++ tenemos disponibles para nuestro uso los siguientes saltos condicionales o controles de flujo.

- If/Else
- For
- While
- Do While

Tabla 1, Tipos de placas Arduino

Tipos de Arduino	Gráfico
Arduino UNO	 A blue Arduino Uno board with a USB Type-B port, a DC power jack, and a 5-pin header.
Arduino MKR1000	 Two small, blue Arduino MKR1000 boards, one slightly behind the other, showing their compact design.
Arduino/Genuino 101	 A green Arduino/Genuino 101 board featuring an Intel Atom processor, a USB Type-C port, and a micro-USB port.
Arduino Zero	 A blue Arduino Zero board with two red arrows pointing to the 'Native USB Port' and the 'Programming Port'.
Arduino Yun	 A blue Arduino Yun board with a white Ethernet shield and a USB Type-B port.
Arduino Leonardo	 A blue Arduino Leonardo R3 board with a USB Type-B port and a DC power jack.

Arduino Due	 The image shows an Arduino Due microcontroller board. It is a blue printed circuit board with a large AT91SAM3X8E microcontroller chip in the center. The board features a USB Type-C port on the left side, a DC power jack, and a 5-pin header. The text 'ARDUINO DUE' is printed on the board.
Arduino Nano	 The image shows an Arduino Nano microcontroller board. It is a small, blue printed circuit board with a USB Type-B port on the left side. The board features a DC power jack, a reset button, and a 5-pin header. The text 'ARDUINO NANO' and 'BRUNNTECH.COM' are visible on the board.
Arduino Pro	 The image shows an Arduino Pro microcontroller board. It is a small, blue printed circuit board with a USB Type-B port on the left side. The board features a DC power jack, a reset button, and a 5-pin header. The text 'ARDUINO Pro' and 'sparkfun.com' are visible on the board.
Arduino Mega	 The image shows an Arduino Mega microcontroller board. It is a large, blue printed circuit board with a USB Type-B port on the left side. The board features a DC power jack, a reset button, and a 5-pin header. The text 'ARDUINO MEGA' and 'ATMEGA2560' are visible on the board.

Elaborado por: Los autores

3. EJEMPLO PRÁCTICO

En este apartado se presentará y describirá el ‘parpadeo de un diodo led para poner en práctica todo lo descrito anteriormente.

3.1. Materiales

- 1 Arduino nano
- 1 diodo led
- 1 resistencia 10 ohm
- 1 proto board
- Cables de conexión

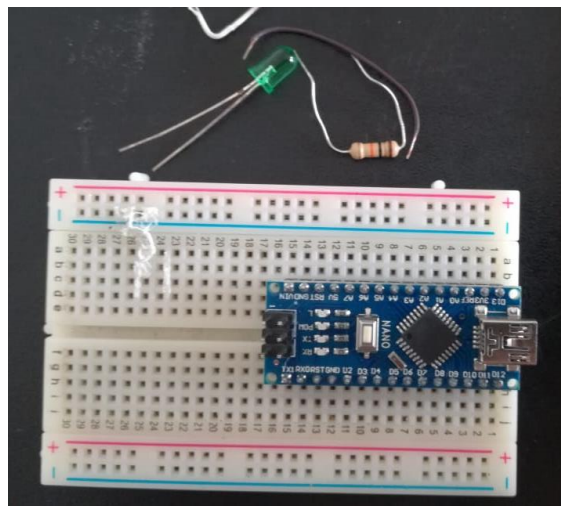


Figura 25. Materiales
Elaborado por: Los autores

3.2. Código

```
Blink Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

Blink $

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(12, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(12, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(12, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

Figura 26. Código para subir al Arduino
Elaborado por: Los autores

3.2.1. Explicación de las líneas de código

En la primera parte de **setup()** se configura la ejecución del programa, se puede seleccionar el pin con que vayamos a trabajar.

```
// La función de configuración se ejecuta una vez
cuando presiona restablecer o enciende la placa

void setup () {
```

Figura 27. Setup del código
Elaborado por: Los autores

De acuerdo con la figura 28. En la línea de código tenemos el inicio de un ciclo que es void loop() en donde tenemos digitalWrite high el cual es el encargado de enviar una señal digital de pulso alto al pin que nosotros lo seleccionemos, así mismo tenemos digitalWrite (low) el cual envía un pulso bajo de señal digital y finalmente tenemos delay el cual es el intervalo de tiempo que se va a realizar esta acción.

```
void loop () {

digitalWrite(12, HIGH); // encienda el LED (HIGH es el nivel de voltaje)

delay(1000);           // Espera un segundo

digitalWrite(12, LOW); // apague el LED haciendo el voltaje LOW
```

Figura 28. Void loop del código
Elaborado por: Los autores

3.3. Conexión del tablero Arduino nano

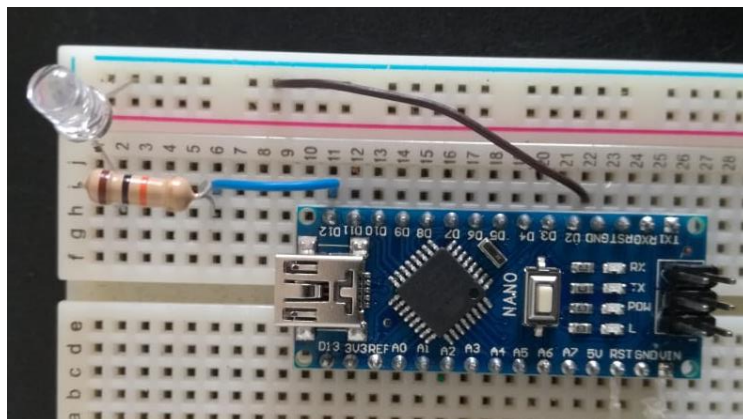


Figura 29. Conexión en la placa Arduino a los componentes
Elaborado por: Los autores

- Conectamos un cable desde el pin seleccionado a la resistencia.
- Se conecta la luz led a un extremo de la resistencia y e otro a tierra con las polaridades correctas
- Con un cable conectamos tierra de nuestro tablero Arduino (GND) a tierra de nuestro proto board tal como se muestra en la figura 29.

3.4. Cargar el código a nuestro Arduino

Primero conectamos nuestro Arduino a la computadora tal como se muestra en la figura 30.

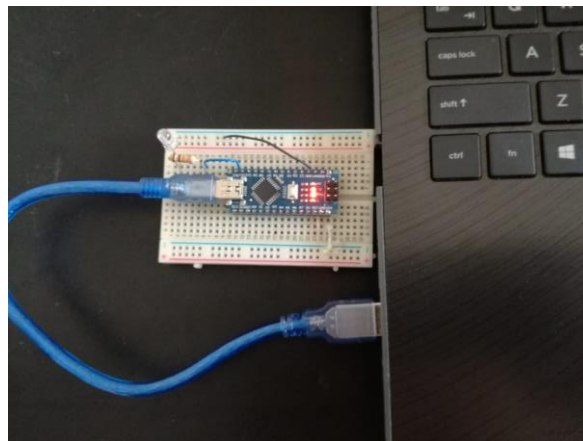


Figura 30. Conexión a la computadora
Elaborado por: Los autores

Una vez listo procedemos a cargar nuestro código y ya estaría completo tal como se aprecia en la figura 31.

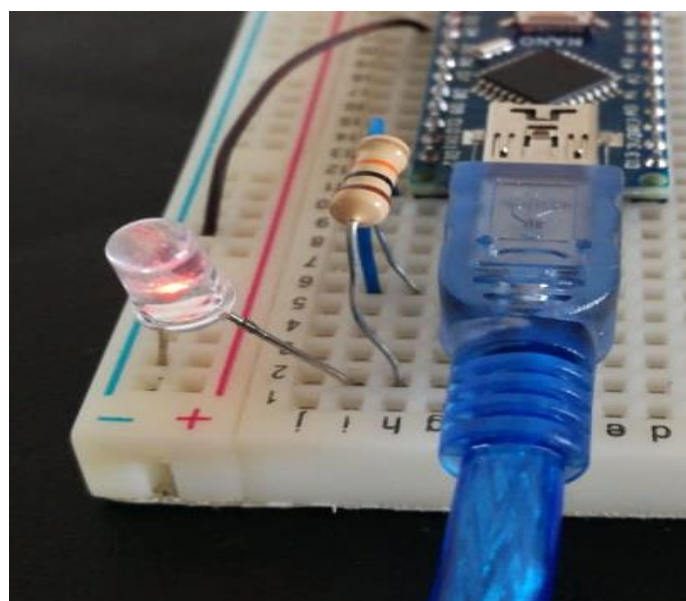


Figura 31. Prueba de funcionamiento del parpadeo de la luz led
Elaborado por: Los autores

4.RESPONSABLES.

- **Chanataxi Vega Carlos Antonio**

Perfil:

- Estudiante actual en la carrera de redes y telecomunicaciones en el Instituto Tecnológico Superior Huaquillas

Cursos y seminarios:

- Foro de comunicaciones en la tecnología 5G por la Universidad Nacional de Loja
- Curso del internet de las cosas (cisco) por la ESPOL

Experiencia profesional:

- Empresa proveedora de internet ZCELL comunicaciones
- Perteneciente al club de robótica del Instituto tecnológico Superior Huaquillas

- **Sarango Cárdenas Ángel Javier**

Perfil:

- Estudiante actual en la carrera de redes y telecomunicaciones en el Instituto Tecnológico Superior Huaquillas

Cursos y seminarios:

- Foro de comunicaciones en la tecnología 5G por la Universidad Nacional de Loja
- Curso del internet de las cosas (cisco) por la ESPOL

Experiencia profesional:

- Empresa proveedora de internet LINKPC

- **Ing. Jorge David Herrera Sarango**

Perfil:

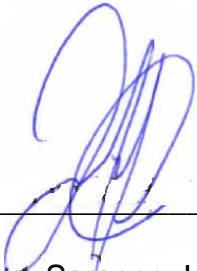
- Ingeniero en Electrónica y Telecomunicaciones.
- Magister en tecnologías de la información con mención en seguridad de redes y comunicaciones.

Cargos Ocupados:

- Responsable de proyectos / EDILOJA CIA.LDTA.
- Soporte técnico / EDILOJA CIA.LDTA.
- Técnico operador / EDILOJA CIA.LDTA.
- Asistente/instalador auxiliar de informática y telecomunicaciones / INTEC.
- Proyectista de diseños en Fibra Óptica / CNT.

Cátedra y conferencias:

- Docente del Instituto Superior Tecnológico “Huaquillas” impartiendo las materias de: teoría de redes informáticas, fundamento de redes y telecomunicaciones, electrónica básica, arquitectura de redes, telecomunicaciones.
- Coordinador de investigación del Instituto Superior Tecnológico “Huaquillas”.

Responsables:

Ing. Herrera Sarango Jorge David

Revisado y aprobado por:

Ing. Paquita Cuadros

5.GLOSARIO.

- **GND:** (Ground) es la toma de tierra.
- **Input/Output:** Constantes usadas con la función `pinMode()` para definir el modo de un pin digital como INPUT u OUTPUT.
- **IDE:** Integrated Development Environment. "Entorno de desarrollo integrado".
- **RX:** RX (recepción).
- **TX:** TX (transmitir).
- **Void:** Variable sin tipo definido

6.REFERENCIAS.

Astudillo, G. (10 de junio de 2018). *wikihow*. Obtenido de <https://es.wikihow.com/instalar-VirtualBox>

Benavides, A. (19 de diciembre de 2016). *Arduino tecnologia para todos* . Obtenido de <https://arduinodhtics.weebly.com/tipos-de-arduino.html>

Bolagay, J. (27 de agosto de 2017). *aprendiendo arduino* . Obtenido de <https://aprendiendoarduino.wordpress.com/2017/06/18/ide-arduino-y-configuracion/>

Leonardo, S. (26 de junio de 2017). *MCI electronics*. Obtenido de <http://arduino.cl/como-instalar-arduino-en-windows/>

Veloso, C. (13 de mayo de 2016). *E tools*. Obtenido de <https://www.electrontools.com/Home/WP/2016/05/13/programacion-arduino-uno/>