



ARDUINO PARA DOCENTES: HANDS ON! ALFABETIZACIÓN CIENTÍFICA Y TECNOLÓGICA

José Luis Paternina Durán

Julián Andrés Salamanca Bernal

José De Jesús Paternina Anaya



ARDUINO PARA
DOCENTES:
HANDS ON!
ALFABETIZACIÓN
CIENTÍFICA Y
TECNOLÓGICA

ARDUINO PARA DOCENTES: HANDS ON! ALFABETIZACIÓN CIENTÍFICA Y TECNOLÓGICA

José Luis Paternina Durán

Julián Andrés Salamanca Bernal

José De Jesús Paternina Anaya



Paternina Durán, José Luis

Arduino para docentes: Hands on! alfabetización científica y tecnológica / José Luis Paternina Durán, Julián Andrés Salamanca Bernal y José de Jesús Paternina Anaya, Primera edición, Bogotá: Universidad Santo Tomás, 2026.

112 páginas.

Incluye referencias bibliográficas.

e-ISBN: 978-958-782-735-4

1. Arduino (Microcontrolador) - Enseñanza 2. Alfabetización científica 3. Alfabetización tecnológica
4. Innovaciones educativas 5. Tecnología educativa 6. Materiales de enseñanza I. Universidad Santo Tomás (Colombia).

CDD 621.39

CO-BoUST

© José Luis Paternina Durán, Julián Andrés Salamanca Bernal y José De Jesús Paternina Anaya, 2026

© Universidad Santo Tomás, 2024

Ediciones USTA

Bogotá, D. C., Colombia

Carrera 9 n.º 51-11

Teléfono: (+571) 587 8797, ext. 2991

editorial@usta.edu.co

<http://ediciones.usta.edu.co>

Corrección de estilo:

Lorena Castro Castro

Diagramación y diseño de cubierta:

Patricia Montaña D.

Hecho el depósito que establece la ley

e-ISBN: 978-958-782-735-4

Primera edición, 2026

Universidad Santo Tomás

Vigilada MinEducación

Reconocimiento personería jurídica:

Resolución 3645 del 6 de agosto de 1965,

Minjusticia

Acreditación Institucional de Alta Calidad

Multicampus: Resolución 014525 del 28 de julio

de 2022, 8 años, MinEducación

Se prohíbe la reproducción total o parcial de esta obra, por cualquier medio, sin la autorización expresa del titular de los derechos.

CONTENIDO

Prefacio	11
Referencias	15
Mi primera práctica con Arduino	17
Objetivos del capítulo y resultados de aprendizaje	17
Conociendo Arduino	17
Conceptos básicos	41
Objetivos del capítulo y resultados de aprendizaje	41
Los circuitos eléctricos	41
La carga eléctrica	42
El voltaje	42
La corriente eléctrica	43
La resistencia eléctrica	44
El concepto de nodo	46
Leyes fundamentales	48
Ley de Kirchhoff de voltaje	49
Conexiones en la protoboard	53
Montaje del semáforo	59
Sensores	67
Objetivos del capítulo y resultados de aprendizaje	67
Manejo de errores: recurso para análisis en las prácticas de laboratorio	69

Aplicación de sensores y Arduino en mi contexto	77
Objetivos del capítulo y resultados de aprendizaje	77
Sensores con Arduino	77
Sensor de distancia	78
Sensor de ruido	84
Sensores no adaptados para Arduino	87
Objetivos del capítulo y resultados de aprendizaje	87
Otros sensores	88
Sensor de gas	96
Epílogo. ¿Y ahora?	105
Referencias	107
Perfiles de los autores	109

Lista de figuras

Figura 1. Estructura del texto	13
Figura 2. Frases célebres y organización jerárquica de contenidos.	14
Figura 3. Placa electrónica Arduino uno con su cable USB.	19
Figura 4. Pantalla principal de la plataforma de desarrollo en el computador	20
Figura 5. Protoboard	
a) Protoboard y lámina metálica.	
b) Retiro de protector de adhesivo.	
c) Placa metálica unida con la protoboard.	21
Figura 6. Materiales adicionales para el experimento.	
a) LED de color amarillo, verde y rojo.	
b) Resistencias de distintos valores.	
c) Juego de jumpers hembra-hembra y macho-hembra.	22
Figura 7. Conexiones en la protoboard.	23
Figura 8. Conexiones en la placa Arduino	23
Figura 9. Montaje real del experimento uno: semáforo	24
Figura 10. Navegando en la página de Arduino.	25
Figura 11. Enlace de descarga para Windows o Linux.	25
Figura 12. Opción de descarga con donación o sin donación.	26
Figura 13. Ícono de descarga de Arduino en Google Chrome.	26
Figura 14. Términos de licencia.	27
Figura 15. Selección de los componentes a instalar.	27
Figura 16. Instalación de Arduino.	28
Figura 17. Instalación de controladores adicionales.	28
Figura 18. Instalación completa.	29
Figura 19. Logo de la plataforma de desarrollo Arduino IDE.	29
Figura 20. Pantalla principal de trabajo de Arduino IDE.	30
Figura 21. Código introducido por quien elabora el experimento.	31
Figura 22. Guardando el código elaborado.	32
Figura 23. Guardando el archivo.	33
Figura 24. Ubicación del código guardado.	33
Figura 25. Verificación del código.	34

Figura 26. Mensaje de información al tener una compilación exitosa.	35
Figura 27. Conexión a través de cable usb entre el computador y la placa electrónica de Arduino.	36
Figura 28. Selección del puerto a utilizar.	37
Figura 29. Puerto seleccionado para enviar el código.	37
Figura 30. Selección de la placa	38
Figura 31. Subiendo el código a la placa.	38
Figura 32. Semáforo funcionando	39
Figura 33. Cargas q_1 y q_2	42
Figura 34. Movimiento de la carga q_2 debido a la fuerza de repulsión entre ella y la carga q_1	43
Figura 35. Material conductor:	
a) Material conductor (gran cantidad de electrones libres, baja resistencia).	
b) Mismo material con un voltaje que cierra la trayectoria para que los electrones se muevan.	44
Figura 36. Material dieléctrico	
a) Material dieléctrico (baja cantidad de electrones libre, alta resistencia)	
b) Material dieléctrico con una fuente de voltaje que cierra la trayectoria y produce una corriente eléctrica muy pequeña.	45
Figura 37. Circuito eléctrico.	
a) Circuito eléctrico para encender el bombillo	
b) Mismos elementos, pero con el interruptor abierto. No hay circuito, ya que no existe trayectoria cerrada.	45
Figura 38. Identificación de nodos en un circuito	47
Figura 39. Circuito	
a) Circuito de ejemplo para identificar los nodos.	
b) Nodos identificados en el circuito	47
Figura 40. Circuito de ejemplo, con los nodos correctamente identificados.	48
Figura 41. Circuito de ejemplo	49
Figura 42. Circuitos con trayectorias cerradas	
a) circuito con más de una trayectoria cerrada (no es válido utilizar la ecuación 3).	
b) circuito con una trayectoria cerrada (es válido utilizar la ecuación 3)	50
Figura 43. El diodo	
a) Diodo real con la identificación de sus terminales;	
b) símbolo de circuitos para un diodo con la identificación de sus terminales	51
Figura 44. Diodo en polarización directa	52
Figura 45. Diodo en polarización inversa	52
Figura 46. El LED	52
Figura 47. Tres formas para identificar el cátodo del LED	53

Figura 48. Fuente de voltaje en la placa Arduino Uno	53
Figura 49. Identificación del primer tipo de nodo en una protoboard	54
Figura 50. Conexión entre dos resistencias	
a) Diagrama de circuitos de una conexión entre dos resistencias;	
b) Nodos identificados en la conexión.	54
Figura 51. Conexión de las resistencias de la figura 50, según los nodos de la protoboard identificados en la figura 49.	55
Figura 52. Protoboard con cuatro nodos a lo largo, resultado de unir pares de nodo de la figura 49 mediante cable azul.	55
Figura 53. Ubicación de seis nodos en la parte central de la protoboard.	56
Figura 54. Circuito que se desea colocar en un protoboard.	56
Figura 55. Nodos identificados del circuito de la figura 54.	56
Figura 56. Conexión en la protoboard del circuito de la figura 54.	57
Figura 57. Otro diagrama para representar el mismo circuito de la figura 54.	57
Figura 58. Circuito para encender un LED	58
Figura 59. Diagrama eléctrico del semáforo	59
Figura 60. Ilustración de una gráfica continua	60
Figura 61. Representación gráfica de una variable digital	61
Figura 62. Interfaz de inicio de Arduino IDE.	62
Figura 63. Código del semáforo de la sección <i>void setup ()</i> .	63
Figura 64. Código del semáforo desde la sección <i>void loop()</i> .	64
Figura 65. Ejemplo de circuito eléctrico	
a) circuito eléctrico como representación de un TV conectado a un tomacorriente.	
b) El TV y el tomacorriente representado en a),	68
Figura 66. Sensor de humedad y temperatura DHT11 para Arduino.	74
Figura 67. Sensor de ultrasonido HC-SR04	78
Figura 68. Diagrama de conexiones del sensor de ultrasonido.	79
Figura 69. Código para el funcionamiento del sensor de sonido	80
Figura 70. Selección del Monitor Serie	83
Figura 71. Monitor serie mostrando valores de distancia	84
Figura 72. Sensor KY-038	84
Figura 73. Diagrama de conexiones del experimento dos	85
Figura 74. Código para sensor de sonido	85
Figura 75. Lectura análoga del sensor KY-038 en el monitor serie de Arduino.	86
Figura 76. Sensor de temperatura LM35	88
Figura 77. Terminales del sensor LM35	88
Figura 78. Conexiones del sensor LM35 con Arduino	89

Figura 79. Código para la lectura del sensor de temperatura	90
Figura 80. Valores de temperatura mostrados en el monitor serial.	92
Figura 81. Código resultante para visualizar la gráfica en el Serial Plotter	93
Figura 82. Seleccionando el Serial Plotter de Arduino.	94
Figura 83. Serial Plotter ejecutándose.	94
Figura 84. Código de la figura 79 modificado para realizar dos gráficas simultáneas en el Serial Plotter.	95
Figura 85. Serial Plotter de Arduino visualizando las dos gráficas de la temperatura.	96
Figura 86. Sensor MQ-135	96
Figura 87. Diagrama de conexiones del sensor de gas	97
Figura 88. Código para sensor de gas	97
Figura 89. Circuito equivalente	98
Figura 90. Curvas de calibración dadas por el fabricante para cada uno de los gases que percibe el sensor.	99
Figura 91. Gráfica de CO según los datos del fabricante en la figura 90.	100
Figura 92. Lectura análoga del sensor de gas en el monitor serie de Arduino	102

Lista de tablas

Tabla 1. Unidades básicas del SI.	73
Tabla 2. Características del sensor DHT11	75
Tabla 3. Consideraciones dadas por el fabricante.	91

Prefacio

“Siempre es importante saber de dónde venimos y tener nociones hacia el lugar para dónde vamos”.

Nos encontramos en un contexto de interacción con máquinas donde la ciencia y tecnología median nuestra comunicación. Esto ha llevado a que profesionales y docentes de los distintos dominios de conocimiento sean demandados con resultados de aprendizaje integradores en ciencia y tecnología (y mucho más ahora, con la masificación del trabajo remoto) (Unesco, s. f. a).

Se ha planteado la necesidad de contar con individuos *científica y tecnológicamente alfabetizados*, es decir, personas que sean capaces de aplicar conceptos científicos y utilizar herramientas tecnológicas en la toma de decisiones para la resolución de problemas relacionados con su contexto¹ (Unesco, s. f. b).

Con base en lo anterior, diferentes autores como Cajas (2001), Fourez (1997), y Salvá y Latorre (1998) han construido un discurso en torno a la alfabetización científico y tecnológica (ACT). A partir de su trabajo se han identificado los elementos indicadores presentes en un individuo científico y tecnológicamente alfabetizado. Entre ellos, los más relevantes son:

1. Buen uso de especialistas: sabe que nadie posee la verdad absoluta, por lo cual es necesario acudir a expertos, adoptando un pensamiento crítico.
2. Buen manejo de cajas negras: profundiza en los temas de interés, tomando como “caja negra” aquellos conceptos o temáticas que no son centrales para un proyecto.
3. Buen uso de modelos simples, metáforas o comparaciones: simplifica los modelos mediante comparaciones que le permiten facilitar su comprensión.

¹ Esta concepción fue propuesta por la Unesco desde 1993 y ha fundamentado el trabajo de la alfabetización científica y tecnológica.

4. Comunicación y articulación entre saberes: fomenta un pensamiento con responsabilidad social para el uso de la tecnología, además de comunicar los resultados de sus trabajos.
5. Autonomía: toma decisiones de forma coherente para solucionar inconvenientes con el manejo de una tecnología o concepto científico.

A pesar de que la ACT contiene un discurso sólido y coherente, entre los académicos a nivel mundial la conclusión es dicente: se ha filosofado bastante en alfabetización científica y tecnológica, pero poco material se había elaborado para tal fin hasta hace 10 años (Dillon, 2009; Fourez, 1997), y todavía se sigue presentando, desde nuestra experiencia, material que no posee un diseño metodológico, no especifica una población objeto, ni mucho menos, una metodología de desarrollo y elaboración. Por ello, surge este texto como una de las primeras aproximaciones a recursos que permiten alfabetizar científica y tecnológicamente a una población en particular, con una metodología definida en el “hacer”, ¡si se prefiere *Hands On!* ¡manos a la obra!), y con el último fin de acortar la curva de aprendizaje en temas de ACT.

Este texto está dirigido a docentes de ciencias experimentales, quienes estén interesados en hacer ACT en una herramienta científico/tecnológica como Arduino, con el fin de utilizarla en su práctica profesional.

12

Cada una de las páginas de este libro ha sido cuidadosamente pensada; su estructura, metodología de escritura y consideraciones didácticas responden a un único objetivo: disminuir la curva de aprendizaje de los docentes en ciencias experimentales para el uso de herramientas científico y tecnológicas como Arduino en su contexto profesional particular.

No pretendemos con este texto entregarle un recetario de experimentos para implementar en clase, sino brindarle recursos educativos necesarios para que usted mismo pueda acudir a otras fuentes con criterio, proponer mejoras, realizar modificaciones e implementar los experimentos que necesite para enriquecer su práctica docente.

Como autores, estamos convencidos de que usted como docente desempeña un papel fundamental en la construcción de sociedad mediante la formación de sus alumnos; por lo cual decidimos escribir este recurso de ACT dirigido a usted, ya que estamos seguros de que en algún momento este conocimiento llegará a sus estudiantes.

El contenido de este texto está orientado a que usted como lector considere lo necesario para enriquecer su práctica profesional:

- ▶ Reflexiones de su quehacer como docente.

- ▶ Manejo de errores en mediciones con sensores.
- ▶ Importancia de las gráficas en montajes de laboratorio.

Contenido del texto

Este libro está estructurado en cinco capítulos, los cuales se organizaron cuidadosamente para garantizarle una vinculación afectiva positiva con el contenido de este. Le recomendamos lo lea en orden y con calma.

Estructura

Como se mencionó anteriormente, la estructura de este libro ha sido cuidadosamente pensada para que pueda tener un ritmo de lectura definido. Se han dispuesto avisos a lo largo del texto para evitar equivocaciones que normalmente se cometen; así mismo, se han incluido comentarios en la parte derecha de las páginas para hacerle énfasis en aspectos considerados relevantes. En la figura 1 se presenta la estructura del libro.

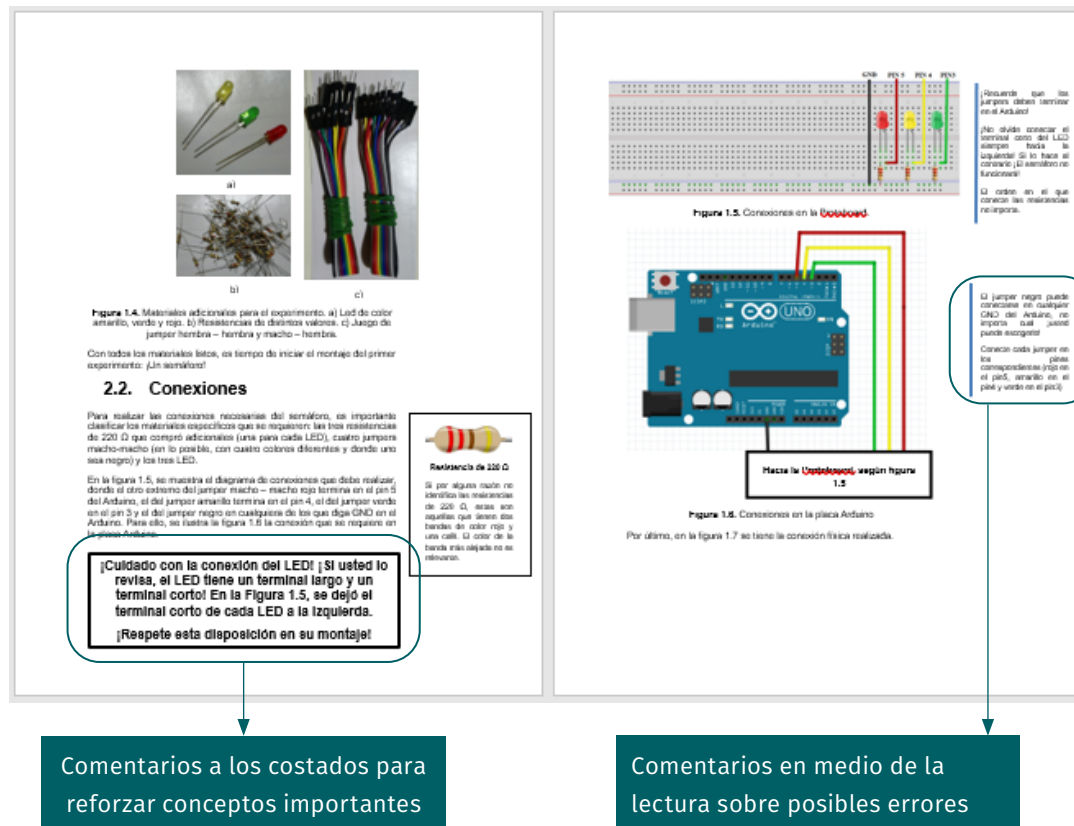


Figura 1. Estructura del texto

Fuente: elaboración propia.

Cada uno de los cinco capítulos inicia con una frase célebre. Por favor, a medida que lea el contenido de algún capítulo, reflexione sobre las implicaciones de esta frase respecto al conocimiento que está adquiriendo.

Además de las frases célebres, también se tuvo mucho cuidado con la organización jerárquica de los contenidos a través de los títulos, con el fin de que usted como lector tenga una concepción global y sistémica de la información que cada capítulo le presenta (vea la figura 2).

14

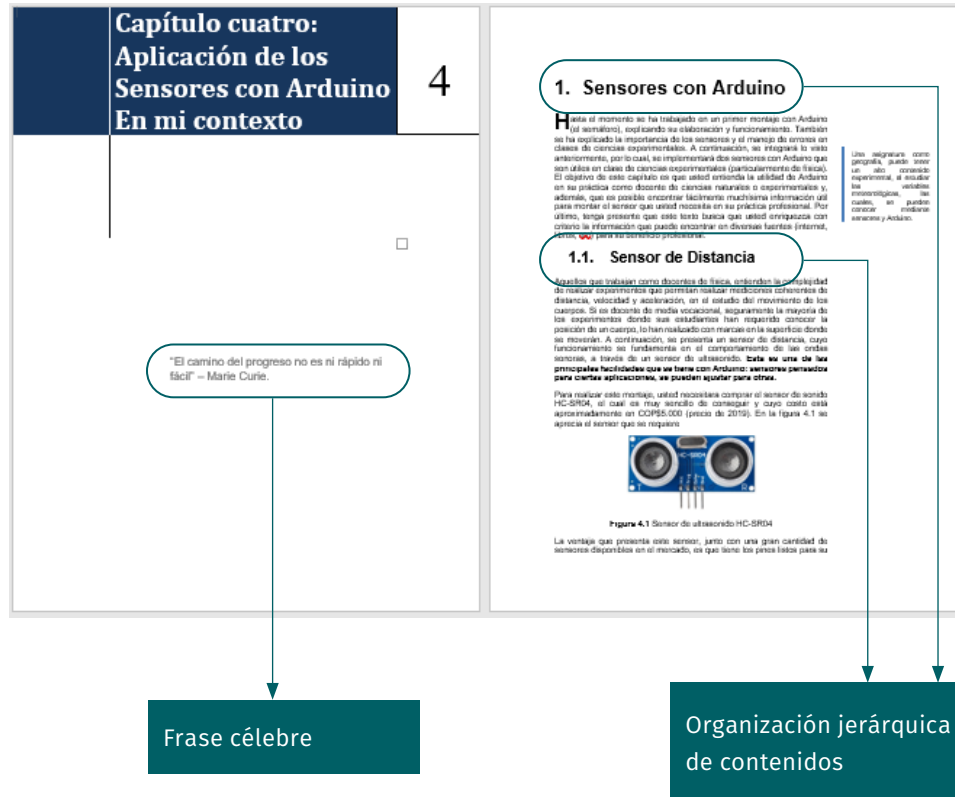


Figura 2. Frases célebres y organización jerárquica de contenidos.

Fuente: elaboración propia.

Para el lector

De acuerdo con los procesos de alfabetización científica y tecnológica que hemos desarrollado previamente, es importante que tenga en cuenta estas recomendaciones:

1. El trabajo en grupo es importante. Si le es posible iniciar este proceso de ACT con algún colega, ¡no dude en hacerlo!
2. Reflexione en las frases célebres antes de iniciar cada capítulo.

3. Aunque se ha realizado un esfuerzo grande para disminuir las posibilidades de error en los experimentos, tenga en cuenta que esta siempre estará latente. *¡Maneje su frustración!* (Zuleta, 2002).
4. Preste especial cuidado al primer capítulo, desarrolle el montaje allí propuesto con mucha calma y siguiendo las instrucciones. Del éxito de este capítulo dependerá en gran parte su trabajo con la herramienta Arduino.
5. Damos la garantía de que *todos* los experimentos propuestos funcionan. Si quiere verificarlo, diríjase al canal de YouTube del grupo FISINFOR UD y, en la sección Alfabetización Científica y Tecnológica encontrará los videos de los montajes funcionando. No olvide que el canal de YouTube es simplemente un apoyo para que tenga la certeza del funcionamiento de los montajes, *en ningún momento reemplaza este texto*.
6. Los costos aproximados de los materiales están en pesos colombianos para el año 2019-2020.

Por último, queremos contarle que este texto es el resultado de investigación de una propuesta de alfabetización en ciencia y tecnología orientada a profesores e investigadores al interior del grupo de Física e Informática de la Universidad Distrital Francisco José de Caldas. La investigación se enmarcó en la realización de cuatro seminarios de ACT, a lo largo de dos años, adscritos a la Maestría en Educación de la misma Universidad, donde más de 40 profesores, en las áreas de ciencias naturales, ingeniería y ciencias sociales, realizaron la mayoría de las experiencias que presenta este libro, supliendo así sus necesidades de medición y adquisición de datos para la realización de laboratorios a bajo costo en su quehacer docente.

Recordamos el brillo en los ojos de la profesora de biología o la euforia del profesor de sociales cuando vio el “semáforo” funcionar, y el rostro de satisfacción de la profesora de tecnología cuando contaba sobre la motivación y alegría que surgió en sus niños estudiantes en condiciones de vulnerabilidad: ¡el robot funcionó!, decían.

Referencias

- Cajas, F. (2001). Alfabetización científica y tecnológica: la transposición didáctica del conocimiento tecnológico. *Enseñanza de las Ciencias. Revista de Investigación y Experiencias Didácticas*, 19(2), 243-254. <https://doi.org/10.5565/rev/ensciencias.4001>
- Dillon, J. (2009). On scientific literacy and curriculum reform. *International Journal of Environmental and Science Education*, 4(3), 201-213. <https://doi.org/10.4324/9781315730486>

Fourez, G. (1997). Scientific and Technological Literacy as a Social Practice. *Social Studies of Science*, 27(6), 903-936. <https://doi.org/10.2307/285671>

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura [Unesco]. (s. f. a). Education: From COVID-19 school closures to recovery <https://www.unesco.org/en/covid-19/education-response>

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura [Unesco]. (s. f. b). Ethics of Science and Technology. <https://www.unesco.org/en/ethics-science-technology>

Salvá, M. y Latorre, F. A. Y. (1998). La alfabetización científica de personas adultas: un enfoque comunicativo, *Enseñanza de las Ciencias. Revista de Investigación y Experiencias Didácticas*, 16(2), 251-260. <https://doi.org/10.5565/rev/ensciencias.4129>

Zuleta, E. (2002). La participación democrática y su relación con la educación. *Polis*, (2), <http://journals.openedition.org/polis/8064>

Mi primera práctica con Arduino

“No tengo talentos especiales, pero sí soy profundamente curioso”
Albert Einstein [fin de epígrafe]

Objetivos del capítulo y resultados de aprendizaje

Objetivos

- ▶ Presentar la herramienta Arduino (tanto *hardware* como *software*) a los docentes, desde una perspectiva simple e intuitiva.
- ▶ Fomentar un vínculo emocional positivo entre el docente y la tecnología Arduino mediante un experimento sencillo.

17

Resultados de aprendizaje al finalizar el capítulo

- ▶ Reconoce el entorno de programación de Arduino y la placa electrónica mediante la manipulación de ambos elementos.
- ▶ Replica un experimento funcional con Arduino, utilizando su entorno de programación y la placa electrónica.

Conociendo Arduino

Piense por un momento que usted se encuentra frente a su computador escribiendo una carta en un editor de textos (Word, Latex, etc.). Efectivamente, al escribir una letra cualquiera, usted la visualiza en la pantalla. Por ejemplo, si oprime la tecla de la letra “a”, en la pantalla usted visualiza la misma letra. ¿Cómo ocurre esto?

Muy bien, ahora, piense en que usted necesita imprimir esa carta, para lo cual conecta una impresora al



La CPU requiere de los diferentes dispositivos periféricos (*mouse*, teclado, pantalla, etc.), para tomar decisiones con base en lo que ellos indiquen.

computador y a través de un clic con el *mouse*, ordena que su carta sea impresa. En ambos casos, todos los dispositivos del computador (teclado, impresora, *mouse*) están conectados de alguna forma (la cual no visualizamos) a la CPU (unidad central de procesamiento, por sus siglas en inglés). La función de la CPU es interpretar las distintas entradas (como los clics del *mouse*, las teclas oprimidas del teclado) y convertirlas en salidas (las palabras visualizadas en la pantalla o la impresión de la carta).

Como es de notar, los elementos externos son incapaces de trabajar por sí mismos, así como la CPU también es inoperante si no cuenta con dispositivos externos que le indiquen las tareas que debe realizar.

Un sensor es un dispositivo que capta magnitudes físicas (luz, temperatura, velocidad, etc.).

De la misma forma trabaja Arduino, el cual no es más que una CPU que requiere recibir distintas entradas (como *sensores*) para convertirlos en salidas (información útil para nosotros). Por ejemplo, si usted quiere leer la temperatura de una sustancia en una práctica de laboratorio, puede hacerlo a través de Arduino. Sin embargo, para ello, deberá conectarle un sensor de temperatura (entrada) y así, logrará visualizar un valor en la pantalla de su computador (salida). De esta forma, Arduino convierte la lectura del sensor (que nosotros no entendemos) en un valor numérico de temperatura (que sí entendemos).

18

Una pregunta que resulta interesante es: ¿cómo hace Arduino para realizar esta tarea? Muy simple, lo hace a través de dos partes: una placa electrónica y una plataforma de desarrollo.

Placa electrónica Arduino Uno

Existen en el mercado distintas placas Arduino, pero en este texto trabajaremos con la placa Arduino Uno, ya que tiene un costo bastante asequible (entre 25 000 y 30 000 COP, precio de 2019) y su funcionalidad es suficiente para el uso que se le pretende dar.

La placa Arduino Uno se visualiza en la figura 3 y allí, se observan las diferentes partes, además de su tamaño (al compararla con un esfero tradicional).



Figura 3. Placa electrónica Arduino uno con su cable usb.

Fuente: elaboración propia.

Como se evidencia en la figura 3, la placa Arduino requiere un cable usb, el cual se necesita para conectarla con el computador, donde se encuentra la plataforma de desarrollo.

En la figura 3, en la parte izquierda y derecha de la placa Arduino se encuentran unos pines de color negro, donde se conectarán los distintos sensores.

Plataforma de desarrollo Arduino

Como se mencionó anteriormente, la placa electrónica requiere una conexión al computador, donde debe estar instalada la plataforma de desarrollo de Arduino (cuya instalación se explicará en la subsección denominada *Código*, del presente capítulo). Esta plataforma de desarrollo es la que se encarga, junto con las conexiones realizadas, en la placa electrónica, de realizar tareas útiles para nosotros. En la figura 4, se observa la pantalla principal de la plataforma de desarrollo, lista para trabajar.

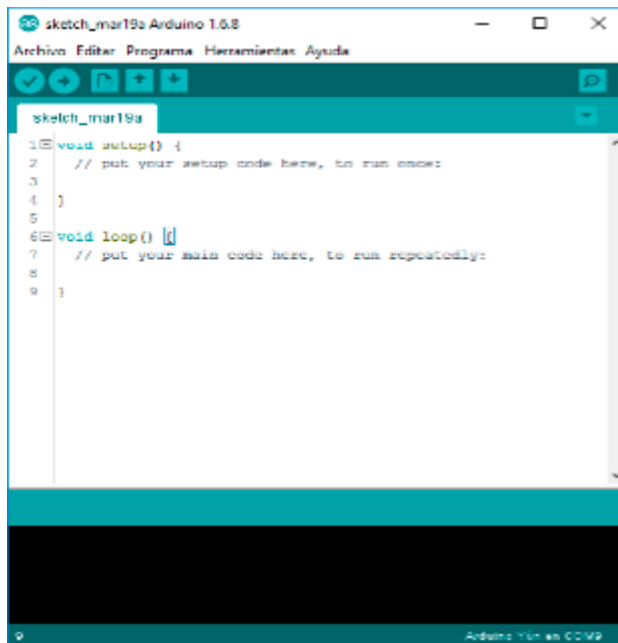


Figura 4. Pantalla principal de la plataforma de desarrollo en el computador

Fuente: elaboración propia.

Mi primer experimento con Arduino

Con lo visto hasta el momento, solo interesa que usted tenga claro que Arduino tiene dos partes igual de importantes: la placa electrónica y la plataforma de desarrollo. A continuación, se realizará un experimento, un primer montaje con Arduino para visualizar lo que es capaz de hacer. Por favor, no se preocupe aún por entender el funcionamiento, ese no es el objetivo por ahora. Solo lea cuidadosamente las instrucciones que se van a presentar a continuación y esté muy atento a los errores comunes que se le indicarán como comentarios en la parte derecha de la página. ¡Ánimo y mucho éxito en su aventura!

Estos costos son aproximados. Los implementos se consiguen en zonas donde haya alta variedad de almacenes electrónicos. En el caso bogotano, se encuentran en la carrera Novena con calle Diecinueve.

Materiales

Todos los materiales que usted requiere para este experimento se consiguen en cualquier tienda de elementos electrónicos. A continuación, se listarán los materiales que necesitará y el costo aproximado en pesos colombianos (año 2019):

- Placa Arduino Uno (28 000 cop)
- Protoboard (12 000 cop) – figura 5.
- Juego de LED: rojo, amarillo y verde (1000 cop) – figura 6a.

- Juego de resistencias de distintos valores (*y aparte, tres resistencias de 220Ω*) (1500 COP) – figura 6b.
- Juego de jumpers macho-hembra y otro juego macho-macho (6000 COP por juego) – figura 6c.

De esta forma, el costo estimado de estos materiales será de 54 500 COP, lo cual es bastante económico, teniendo en cuenta que está incluida la placa Arduino y otros materiales que podrá reutilizar en otros experimentos más adelante (al igual que la placa).

Como seguramente usted no ha manejado antes una protoboard, a continuación, se le mostrará un ajuste que es aconsejable realizar.

En la figura 5a, se tiene una protoboard recién sacada de su caja. Como se observa, esta viene con dos elementos (la protoboard como tal y una lámina metálica). En la parte de abajo de la protoboard, se encuentra un adhesivo con su respectivo protector, el cual se retira (figura 5b) para colocar en su lugar la lámina metálica (figura 5c).

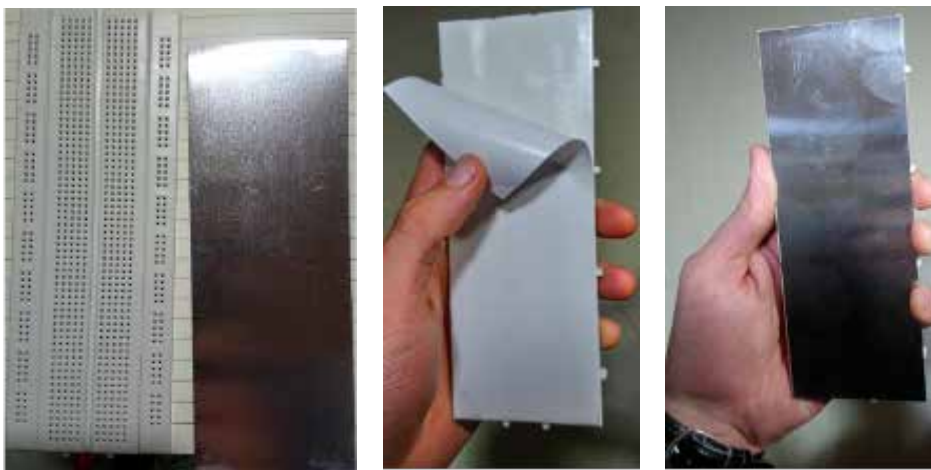
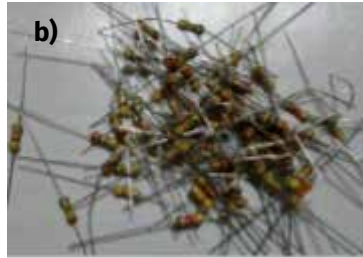
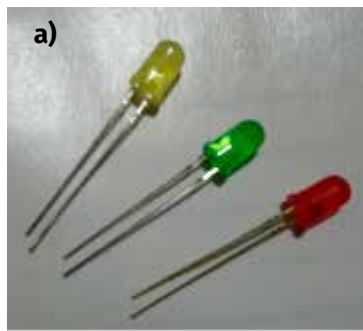


Figura 5. Protoboard

- a)** Protoboard y lámina metálica. **b)** Retiro de protector de adhesivo.
c) Placa metálica unida con la protoboard



Su protoboard puede no traer esta lámina y ser más pequeña. No se preocupe, este ajuste que se realiza es para lograr mayor firmeza en las conexiones, pero funcionará igualmente si no la coloca (porque no quiere hacerlo o porque su protoboard no la trae).

Figura 6. Materiales adicionales para el experimento.

- a)** LED de color amarillo, verde y rojo. **b)** Resistencias de distintos valores.
c) Juego de jumpers hembra-hembra y macho-hembra.

Fuente: elaboración propia.

22

Con todos los materiales listos, es tiempo de iniciar el montaje del primer experimento: ¡un semáforo!

Conexiones

Para realizar las conexiones necesarias del semáforo, es importante clasificar los materiales específicos que se requieren: las tres resistencias de 220Ω que compró adicionales (una para cada LED), cuatro jumpers macho-macho (en lo posible, con cuatro colores diferentes y donde uno sea negro) y los tres LED.

En la figura 7, se muestra el diagrama de conexiones que debe realizar, donde el otro extremo del jumper macho-macho rojo termina en el pin 5 del Arduino, el del jumper amarillo termina en el pin 4, el del jumper verde en el pin 3 y el del jumper negro en cualquiera de los que diga GND en el Arduino. Para ello, se ilustra en la figura 8 la conexión que se requiere en la placa Arduino.



Resistencia de 220Ω

Si por alguna razón no identifica las resistencias de 220Ω , estas son aquellas que tienen dos bandas de color rojo y una café. El color de la banda más alejada no es relevante.

¡Cuidado con la conexión del LED! ¡Si usted lo revisa, el led tiene un terminal largo y un terminal corto! En la figura 7, se dejó el terminal corto de cada LED a la izquierda.
¡Respete esta disposición en su montaje!

¡Recuerde que los jumpers deben terminar en el Arduino!
¡No olvide conectar el terminal corto del LED siempre hacia la izquierda! Si lo hace al contrario, ¡el semáforo no funcionará!
El orden en el que conecte las resistencias no importa.

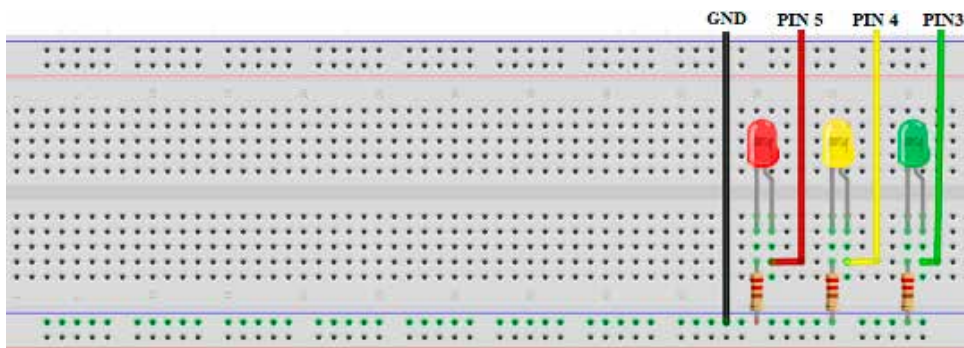
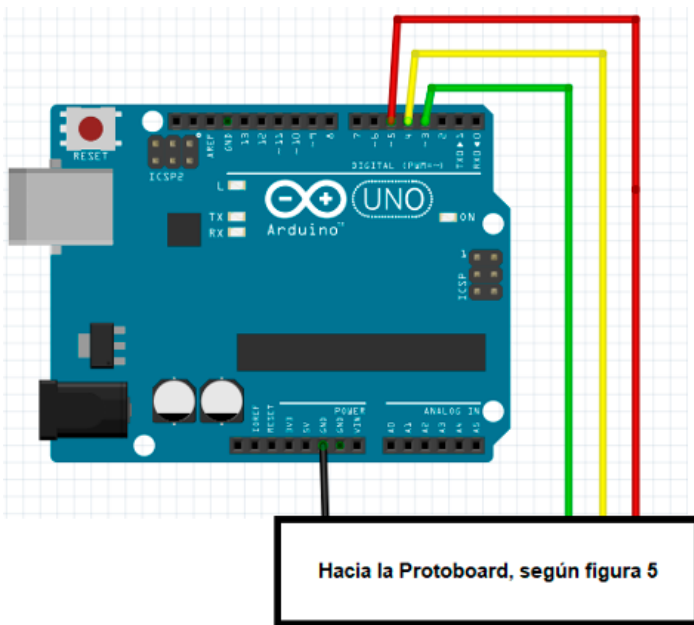


Figura 7. Conexiones en la protoboard.

Fuente: elaboración propia.

23

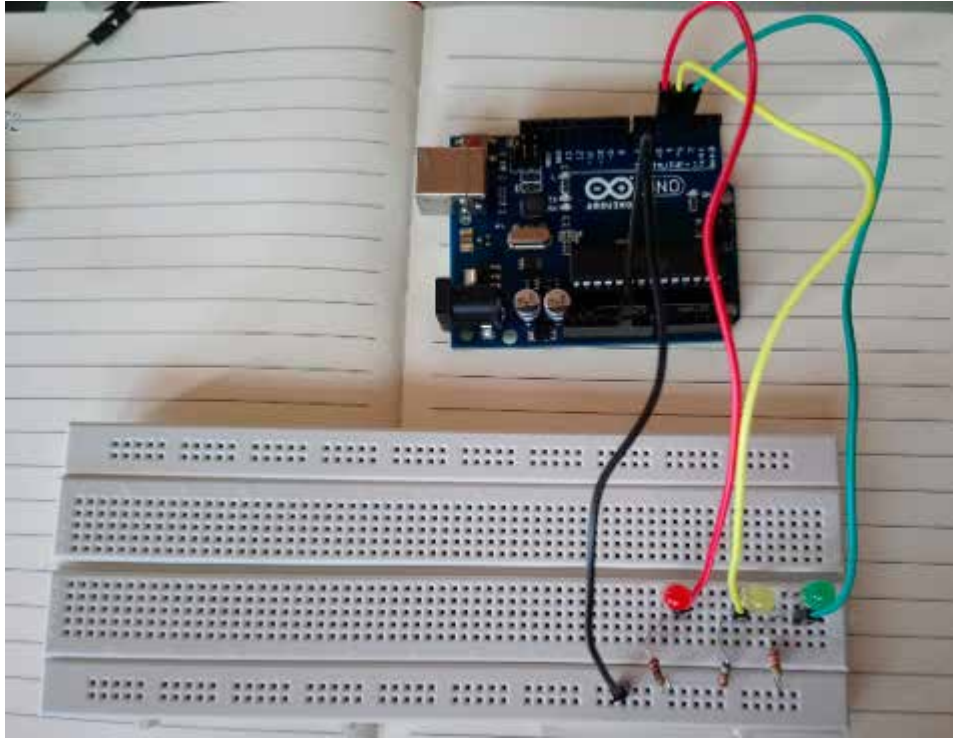


El jumper negro puede conectarse en cualquier GND del Arduino, no importa cuál, ¡usted puede escogerlo!
Conecte cada jumper en los pines correspondientes (rojo en el pin 5, amarillo en el pin 4 y verde en el pin 3)

Figura 8. Conexiones en la placa Arduino

Fuente: elaboración propia.

Por último, en la figura 9 se observa la conexión física realizada.



24

Figura 9. Montaje real del experimento uno: semáforo

Fuente: elaboración propia.

Note que las conexiones realizadas en la figura 9, son las mismas que aparecen en la figura 7 y 8. Por lo anterior, se recomienda siga estos esquemas.

Código

Como se ha mencionado, Arduino requiere además de las conexiones físicas con la placa electrónica, un trabajo adicional en la plataforma de desarrollo. Allí, se debe escribir un código, el cual se encargará de hacer funcionar los experimentos (en este caso, el semáforo). Sin embargo, no se asuste, si usted no es bueno con la programación, ¡acá se le ayudará a superar esos temores! Siempre recuerde que el éxito de este proceso de alfabetización (y cualquier otro) está en el manejo adecuado de sus temores y, sobre todo, de la frustración.

Lo primero que se debe realizar es instalar la plataforma de desarrollo. Esta se puede descargar directamente de la página de Arduino (www.arduino.cc).

No se considere menos por tener dificultades con la programación. Muchas más personas de las que usted cree, y de carreras especializadas (como de ingenierías) tienen problemas con este aspecto. Este es un buen momento para iniciarse en este fascinante mundo. ¡Aproveche la oportunidad!

arduino.cc). Estando allí, ubique la opción *Software* que se encuentra en la parte superior de la página web, tal como se ilustra en la figura 10. Al colocar el cursor del *mouse*, sobre la opción *software* se desplegará un menú con dos opciones: *online tools* y *downloads*, seleccione esta última.

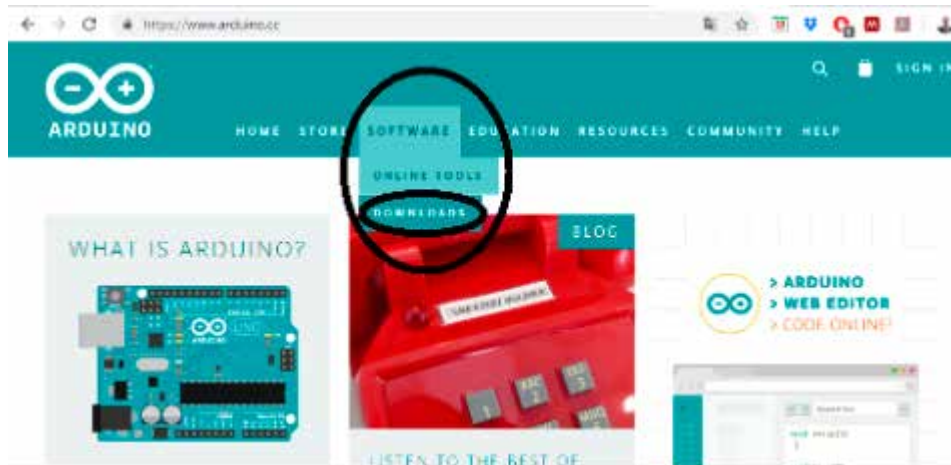


Figura 10. Navegando en la página de Arduino.

Fuente: elaboración propia.

Al dar clic sobre la opción *Downloads* cargará una nueva página (figura 11). En esta, baje hasta encontrar la sección *Download the Arduino ide* y, a la derecha encontrará las opciones de descarga. Seleccione la de su preferencia.

25



Figura 11. Enlace de descarga para Windows o Linux.

Fuente: elaboración propia.

Se recomienda seleccionar la opción *Windows Installer, for Windows XP and up*. Después de seleccionar la opción deseada, se cargará una página como la que se

visualiza en la figura 12. Allí, Arduino solicita una donación voluntaria (tenga en cuenta que el entorno de desarrollo Arduino es libre; es decir, puede ser utilizado de forma gratuita), si usted no quiere realizar ninguna donación, seleccione con un clic la opción *Just download* en la parte de abajo (ver figura 12).

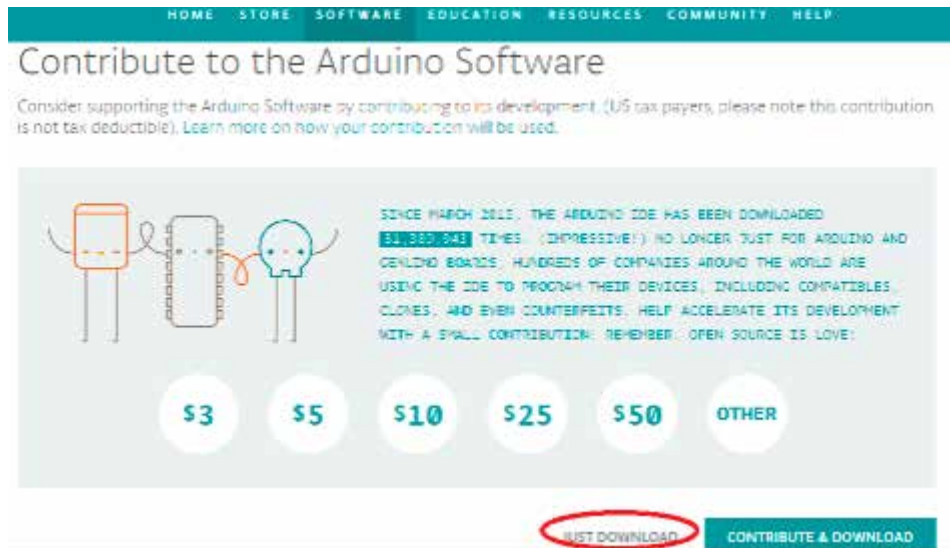


Figura 12. Opción de descarga con donación o sin donación.

Fuente: elaboración propia.

26

Al darle clic a la opción *Just Download* se inicia la descarga del instalador de la plataforma de desarrollo (Conocida como Arduino IDE). Cuando se finalice la descarga, inicie la instalación (dándole doble clic al ícono de descarga que se observa en la figura 13).

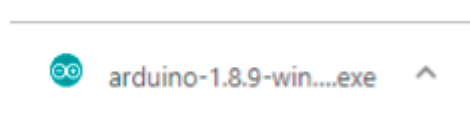


Figura 13. Ícono de descarga de Arduino en Google Chrome.

Fuente: elaboración propia.

Al iniciar la instalación, deberá aceptar las condiciones de licencia con la opción *I Agree* (figura 14).

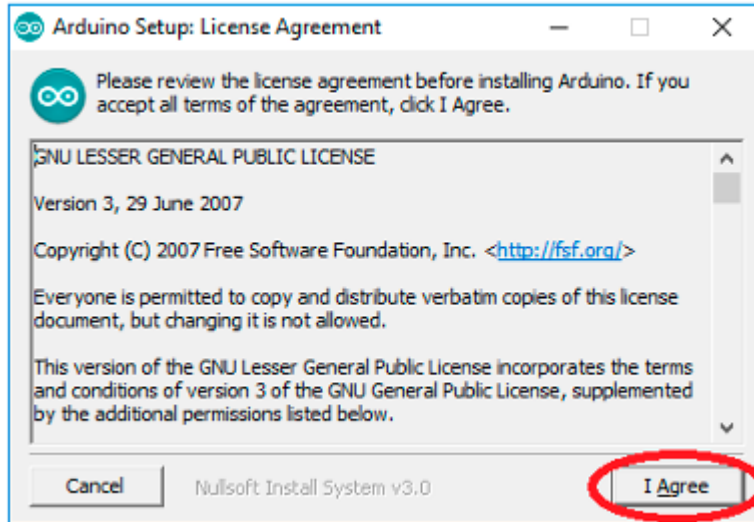


Figura 14. Términos de licencia.

Fuente: elaboración propia.

Después, el asistente de instalación le solicitará que seleccione los componentes a instalar (deje los que están seleccionados de forma predeterminada) y dé clic en *Next* (figura 15).

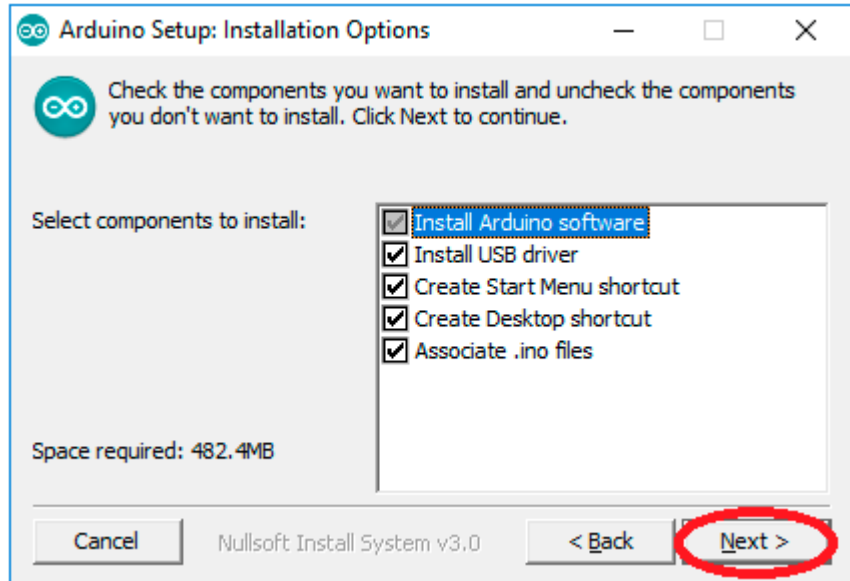


Figura 15. Selección de los componentes a instalar.

Fuente: elaboración propia.

Por último, aparecerá la última ventana (figura 16), donde se selecciona el lugar de instalación (dejar el que aparece predeterminado). Una vez allí, debe seleccionar la opción *Install*.

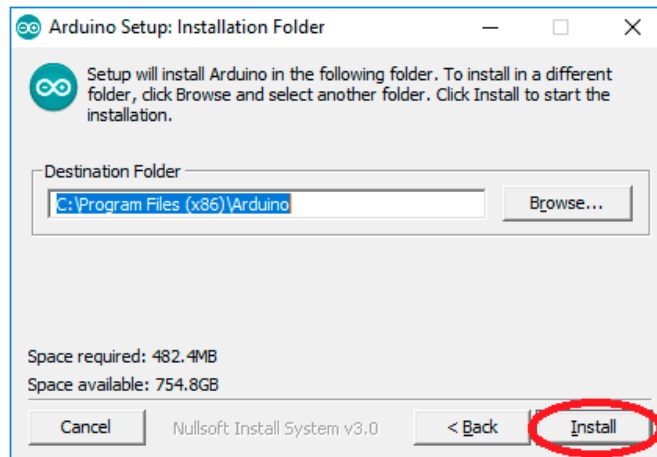


Figura 16. Instalación de Arduino.

Fuente: elaboración propia.

28

Después de seleccionar la opción *Install*, se iniciará toda la instalación del Arduino IDE. En la mayoría de las ocasiones, el computador ya cuenta con todos los controladores listos para lograr la conexión entre la plataforma de desarrollo y la placa electrónica. Sin embargo, en otros momentos, requiere instalarlos también. El instalador del Arduino IDE se encarga de detectar si requiere instalar dichos controladores, por lo que abrirá ventanas como la que se observa en la figura 17, en medio de la instalación. Seleccione la opción *Instalar* a todas estas solicitudes adicionales de instalación (por lo general, son tres las solicitudes que aparecerán).

Si su computador ha sido recientemente formateado, lo más seguro es que al momento de instalar Arduino IDE, requiera instalar otros controladores. En la instalación de la plataforma de desarrollo, ella le ayuda en el proceso.

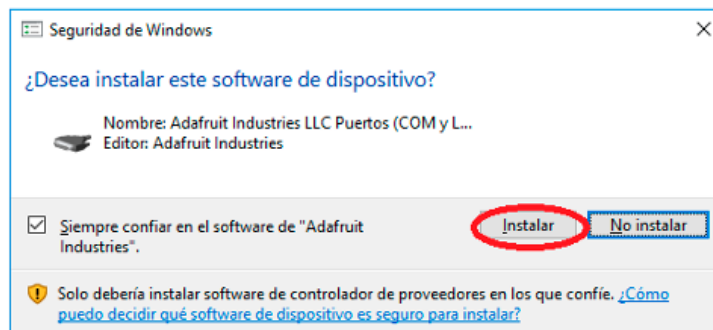


Figura 17. Instalación de controladores adicionales.

Fuente: elaboración propia.

Después de seleccionar la instalación de los controladores adicionales (si no los tiene) finalizará toda la instalación del Arduino IDE. Espere a que finalice este proceso (el cual demora unos minutos) y seleccione la opción *Close* (figura 18). Después de esto, la ventana de instalación se cerrará.

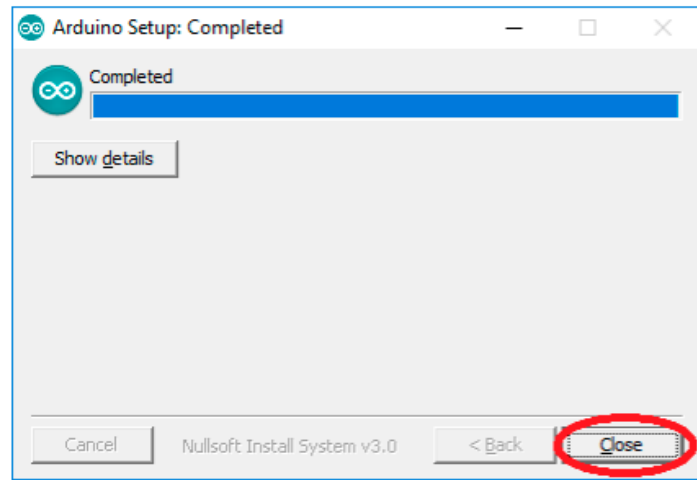


Figura 18. Instalación completa.

Fuente: elaboración propia.

La plataforma de desarrollo no abrirá automáticamente, pero si usted se dirige hacia el escritorio, encontrará el ícono correspondiente (ver figura 19)

29

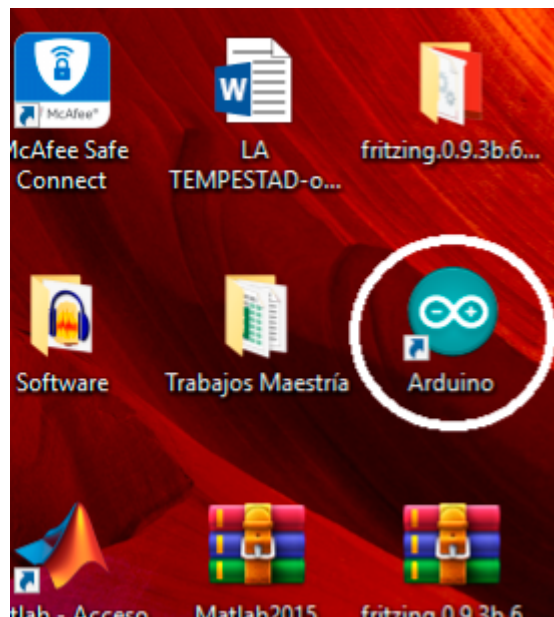


Figura 19. Logo de la plataforma de desarrollo Arduino IDE.

Fuente: elaboración propia. Inicie la aplicación dando doble clic sobre el ícono de la figura 19, al hacerlo, aparecerá la pantalla principal, lista para trabajar (figura 20).

No borre nada del código que aparece al momento de iniciar la aplicación. Por favor, tenga en cuenta que los corchetes cumplen un papel muy importante. ¡No borre ninguno!

Todo el código que vamos a añadir, estará entre los corchetes { }, no entre los paréntesis. Una parte irá entre el primer juego de corchetes y la otra entre el par de corchetes faltante. Más adelante será más claro.

30

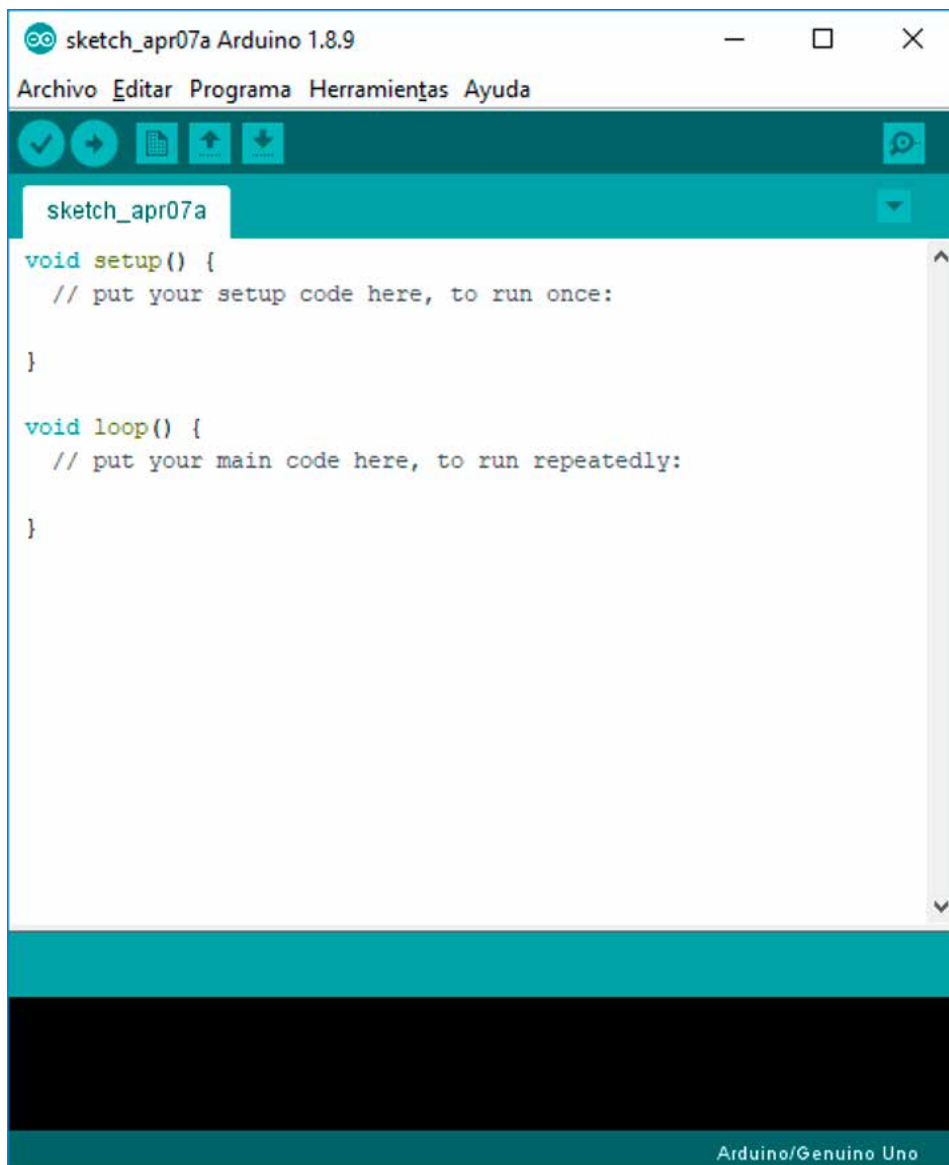


Figura 20. Pantalla principal de trabajo de Arduino IDE.

Fuente: elaboración propia.

A continuación, vamos a introducir el código que permitirá el funcionamiento del semáforo. Allí coloque las líneas de código que aparecen encerradas en la figura 21 (no borre nada del código que aparece predeterminado al momento de iniciar la aplicación).

¡CUIDADO!

Tenga en cuenta estas consideraciones para evitar errores comunes:

La palabra *pinMode* debe tener todas las letras en minúscula *menos la M que debe ser mayúscula*.

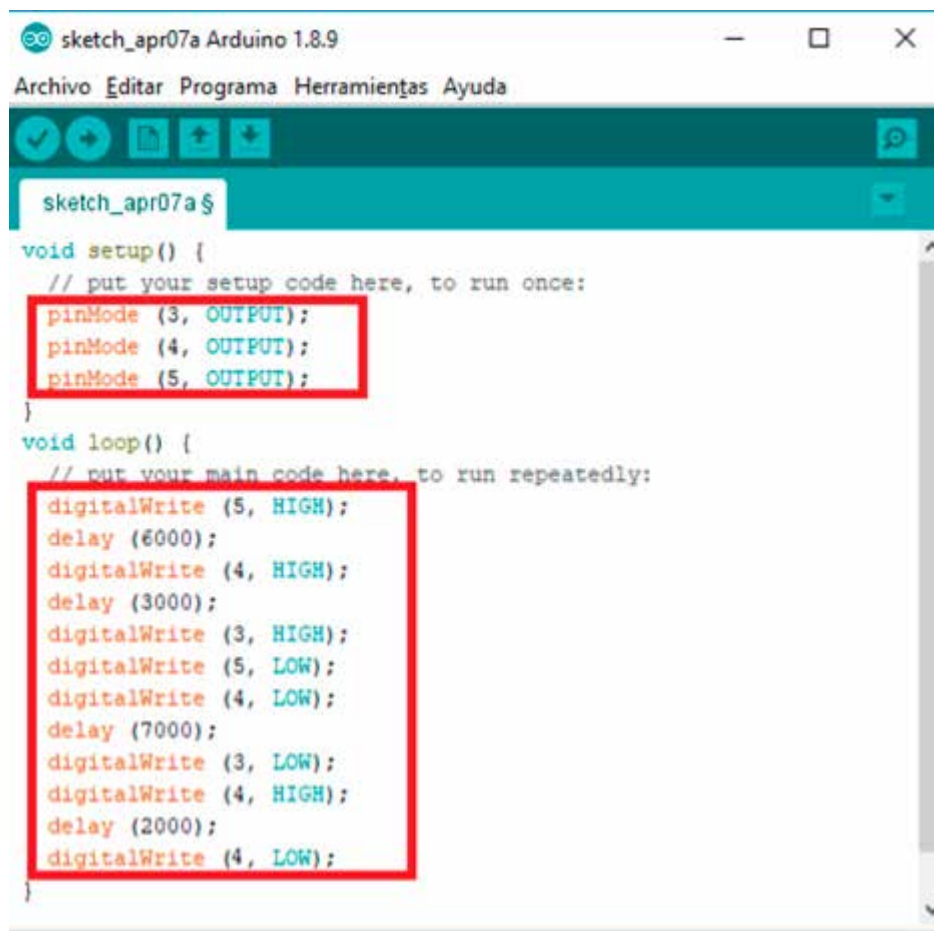
La palabra *OUTPUT* debe estar *toda en mayúscula*.

La palabra *digitalWrite* debe tener todas las letras en minúscula *menos la W que debe ser mayúscula*.

Las palabras *HIGH* y *LOW* deben estar *todas en mayúscula*.

Note que todas las palabras en cada línea de código *inician en minúscula*.

No olvide el punto y coma (;) al final de cada línea de código que introduzca.



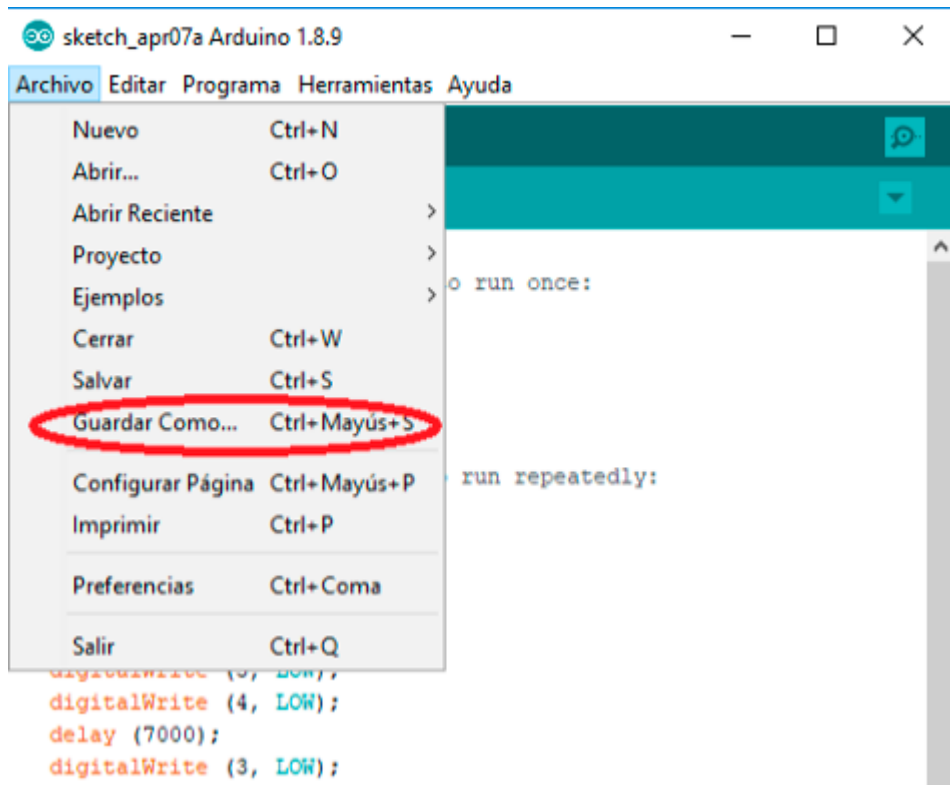
```
sketch_apr07a Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
sketch_apr07a $
void setup() {
  // put your setup code here, to run once:
  pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (5, OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite (5, HIGH);
  delay (6000);
  digitalWrite (4, HIGH);
  delay (3000);
  digitalWrite (3, HIGH);
  digitalWrite (5, LOW);
  digitalWrite (4, LOW);
  delay (7000);
  digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  delay (2000);
  digitalWrite (4, LOW);
}
```

Figura 21. Código introducido por quien elabora el experimento.

Fuente: elaboración propia.

Note que las líneas de código que aparecían en la figura 20, se mantienen en la figura 21. Lo que se hizo fue adicionar nuevas líneas de código (sea muy cuidadoso de ubicarlo donde se ilustra en la figura 21 y sin omitir nada).

A continuación, se debe guardar el código, para ello, en la barra de herramientas, seleccione el menú *Archivo* y la opción *Guardar como*, tal como se observa en la figura 22.



32

Figura 22. Guardando el código elaborado.

Fuente: elaboración propia.

Introduzca el nombre que se desee y seleccione la opción *Guardar* (figura 22).

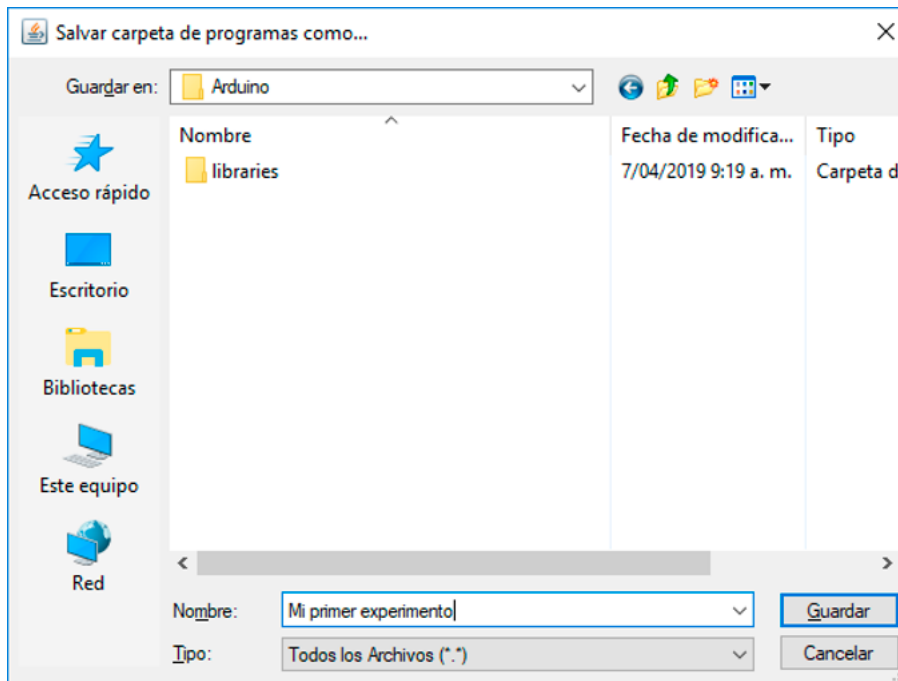


Figura 23. Guardando el archivo.

Fuente: elaboración propia.

Por lo general, la carpeta donde se guardó el archivo con el código, se llama Arduino y está ubicada en la carpeta *Mis documentos*. Dentro de esa carpeta, se crea otra con el nombre asignado al momento de guardar. Allí se encuentra el código disponible para ser modificado cuando se desee (figura 23).

33



Figura 24. Ubicación del código guardado.

Fuente: elaboración propia.

Después de guardar el código, usted debe verificarlo; para ello, seleccione la opción *Verificar* ubicada en la parte superior izquierda de la aplicación, cuyo símbolo se ilustra en la figura 25, encerrado en el recuadro rojo.

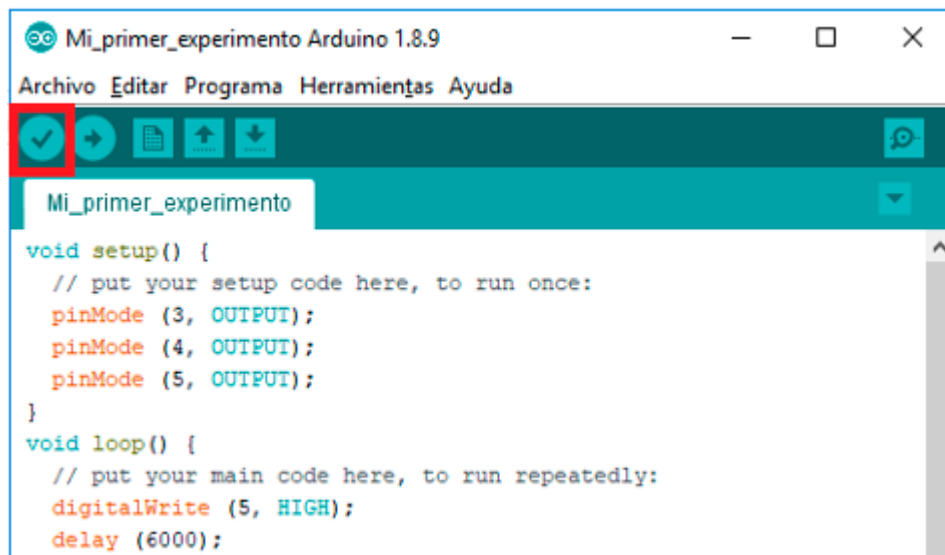


Figura 25. Verificación del código.

Fuente: elaboración propia.

Al darle clic sobre esta opción, se debe esperar unos segundos a que la aplicación verifique el código realizado. Si no se presentan errores después de la verificación, en la parte inferior de la pantalla, aparecerá lo que se observa en la figura 26.

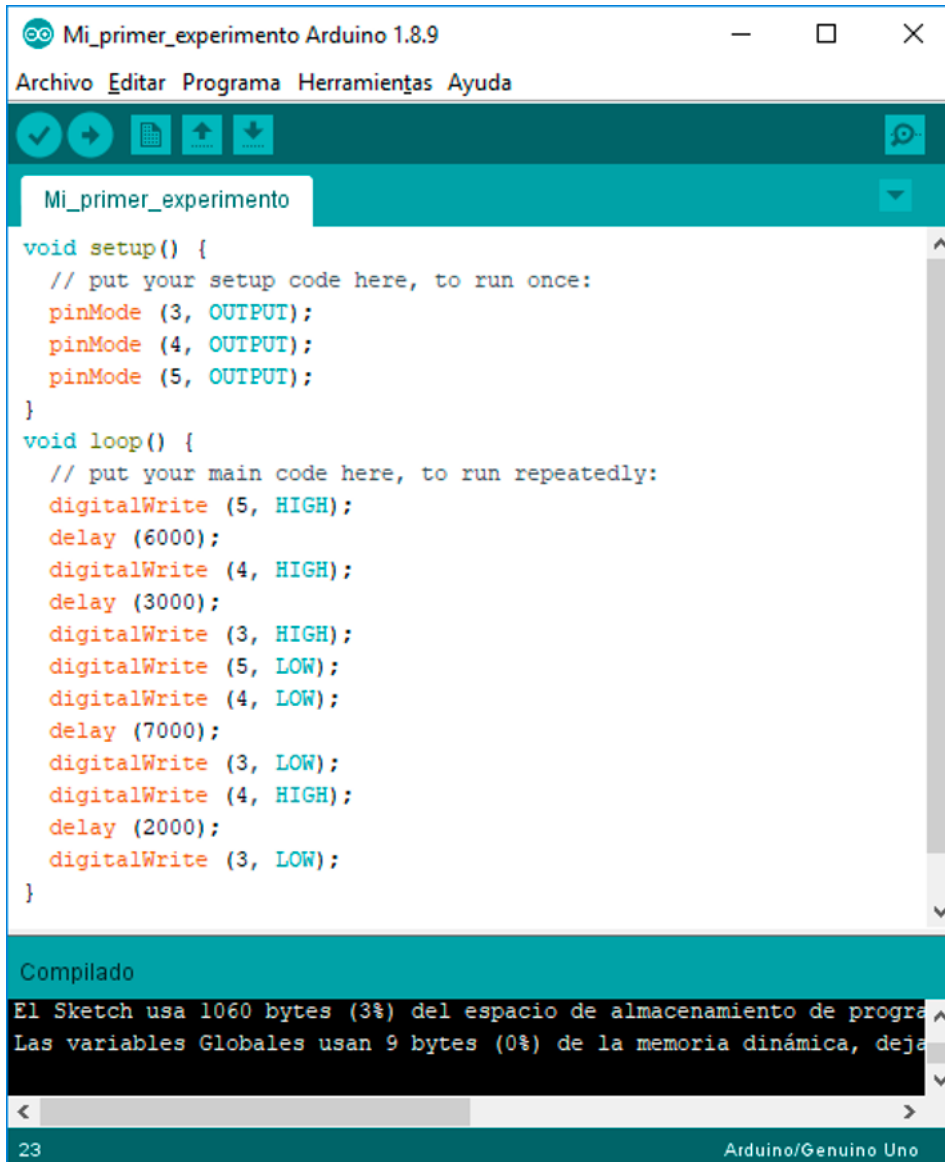


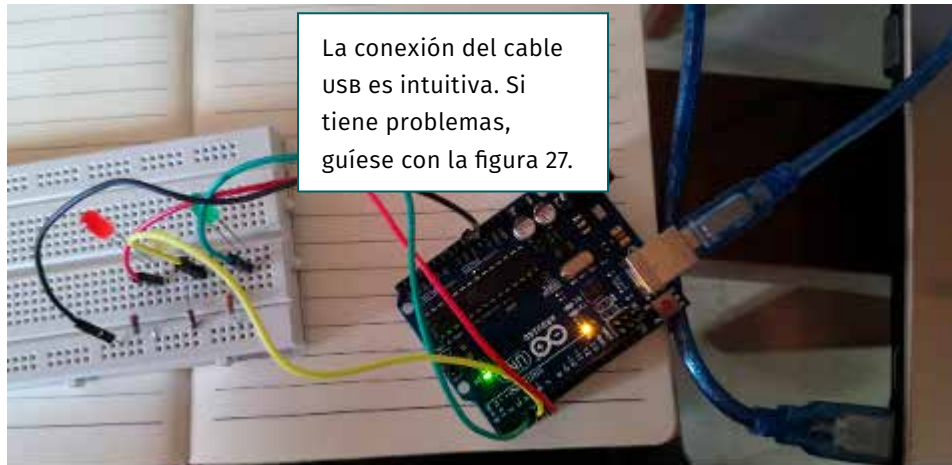
Figura 26. Mensaje de información al tener una compilación exitosa.

Fuente: elaboración propia.

En esta sección aparecerán los errores que se puedan presentar al momento de verificar el código (compilarlo) o de subirlo a la placa

Muy bien. En este momento, se tiene la placa electrónica con las conexiones realizadas y el código en la plataforma de desarrollo. Sin embargo, para que el semáforo funcione, se debe realizar la conexión entre ambas partes (placa y plataforma Arduino IDE). Para ello, utilice el cable USB que vino con su placa Arduino y conéctela a cualquier puerto de su computador y a la placa Arduino. Esta conexión se observa en la figura 27.

Si aparecen problemas, ¡no se preocupe! Seguramente cometió algunos de los errores comunes que se mencionan en los comentarios a la derecha de la página. ¡No se frustre! Vuélvalo a intentar y maneje su frustración. ¡Usted puede!



36

Figura 27. Conexión a través de cable USB entre el computador y la placa electrónica de Arduino.

Fuente: elaboración propia.

Al analizar la figura 27, se logra percibir que, al conectar la placa con el computador, varias indicaciones luminosas se encienden. En verde y sin parpadear, se enciende la indicación de *on*, además de que parpadea la indicación de *tx* y cerca de ella una luz naranja permanece encendida.

Después de esto, se requiere enviar el código de la aplicación del computador a la placa. Para ello, vaya a la aplicación Arduino IDE y en la barra de herramientas ubique la opción *Herramientas*. Allí, seleccione la opción puerto y dé clic en la opción que le aparece (ver figura 28).

Si aparece más de un puerto y no sabe cuál seleccionar, haga lo siguiente:
Antes de conectar la placa Arduino, vaya a panel de control.
En panel de control, busque la opción *Hardware y Sonido*. Luego, en la sección *Dispositivos e impresoras* aparece la opción *Administrador de dispositivos*, al conectar la placa Arduino, aparece una nueva sección llamada *Puertos COM y LPT*, selecciónela y aparecerá el puerto USB al cual está conectado su placa Arduino.

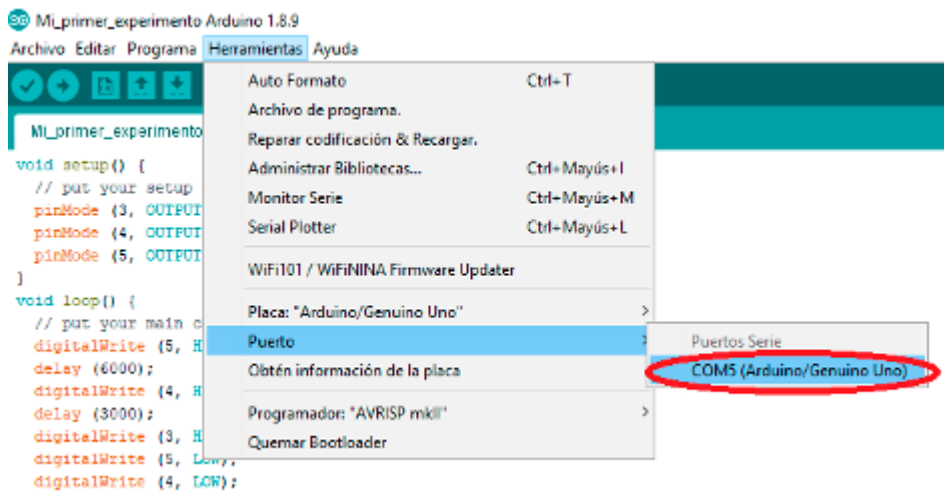


Figura 28. Selección del puerto a utilizar.

Fuente: elaboración propia.

Al darle clic, note que aparece una marca de verificación junto al puerto seleccionado (ver figura 29).

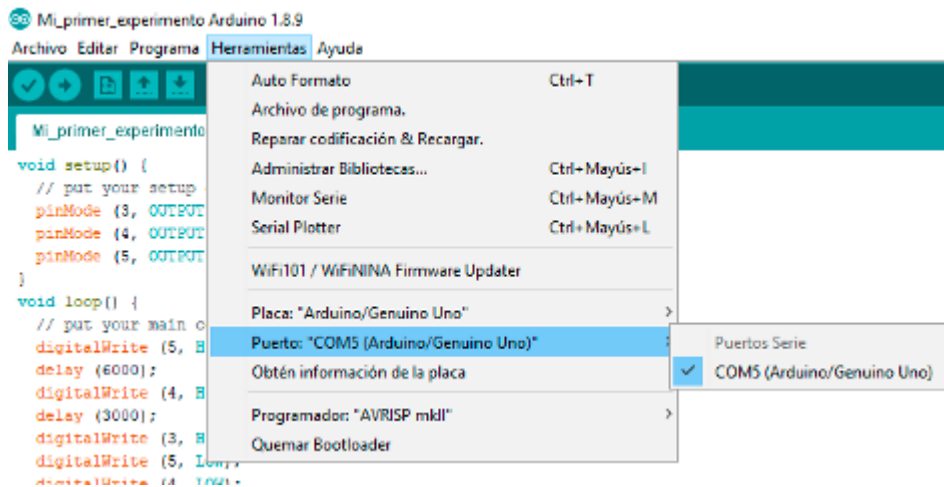


Figura 29. Puerto seleccionado para enviar el código.

Fuente: elaboración propia.

Después de esto, verifique que la placa seleccionada en el programa para trabajar sea la placa Arduino Uno. Para ello, vaya nuevamente a la barra de herramientas y, encima de la opción *Puerto*, utilizada anteriormente, seleccione la opción *Placa*. Allí aparecen todas las placas electrónicas que se pueden trabajar con la aplicación Arduino IDE, como nosotros estamos trabajando con la placa Arduino Uno seleccione la opción (Arduino/Genuino Uno) como aparece en la figura 30.

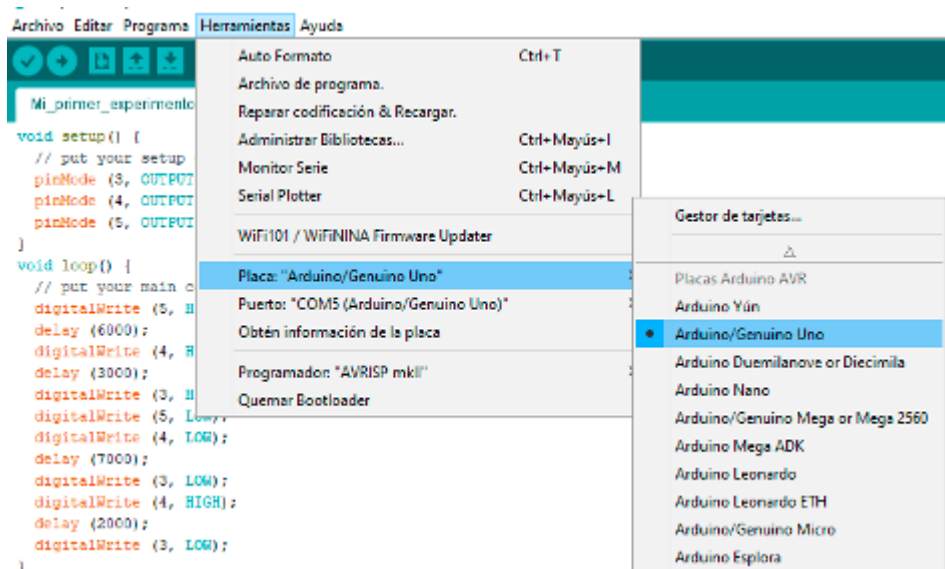


Figura 30. Selección de la placa

Fuente: elaboración propia.

Después de realizar este procedimiento, se procede a enviar el código de la plataforma de desarrollo en el computador, a la placa electrónica. Para ello, en la parte superior izquierda, al lado de la opción *Verificar* utilizada anteriormente, está la opción *Subir*, encerrada en el recuadro de color rojo en la figura 31.

38

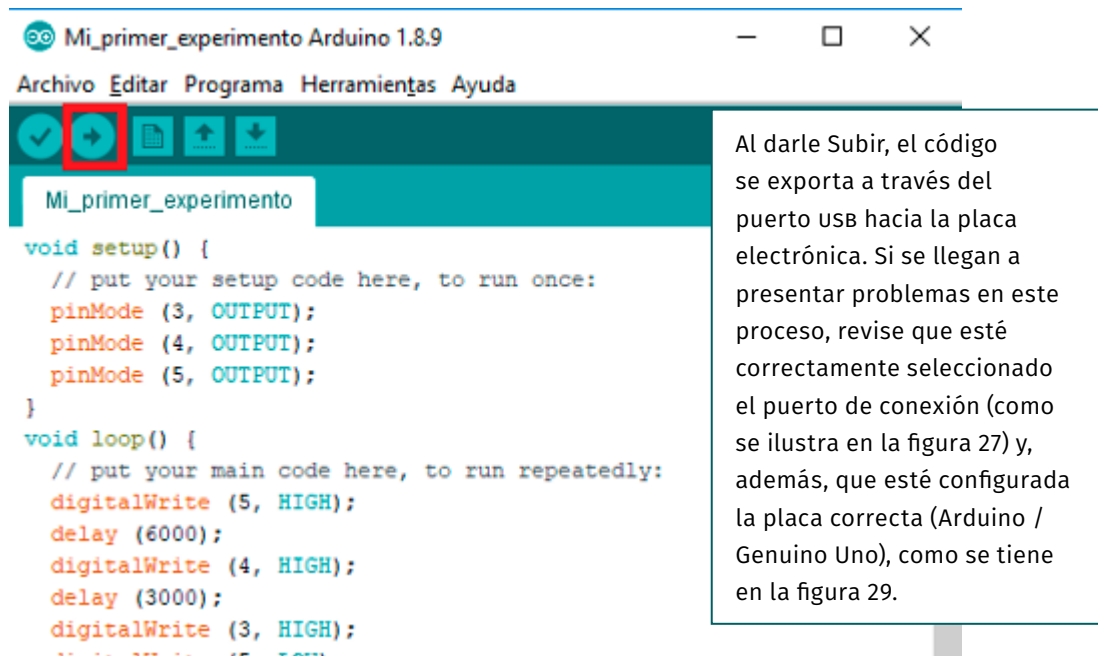
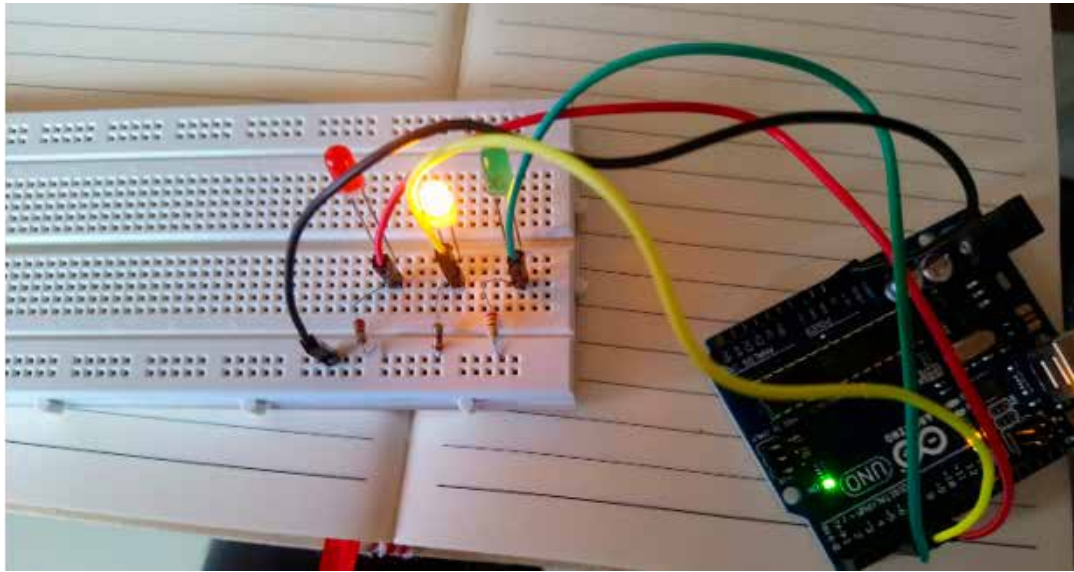


Figura 31. Subiendo el código a la placa.

Fuente: elaboración propia.

Al darle clic a esta opción, se debe esperar unos segundos. ¿Cómo sabe usted que el código se subió correctamente? Mire su semáforo: ¡está funcionando!

En la figura 32 se observa el semáforo funcionando. Sin embargo, para que usted tenga las garantías de que el código y la conexión funcionan, los puede visualizar en el canal de YouTube del grupo FISINFOR de la Universidad Distrital, en la sección: Alfabetización de Arduino para docentes.



39

Figura 32. Semáforo funcionando

Fuente: elaboración propia.

Es normal que el LED verde irradie menos luz que los otros dos. Si percibe un funcionamiento intermitente (como si no hubiese la suficiente energía) verifique sus conexiones en la protoboard (introduzca bien los terminales de cada elemento en los huecos correspondientes).

Si su montaje no funcionó, vuelva a intentarlo revisando con calma cada detalle y los errores comunes que se mencionan en los comentarios. ¡No se rinda!

Conceptos básicos

“La alegría de ver y entender es el más perfecto don de la naturaleza”
Albert Einstein

Objetivos del capítulo y resultados de aprendizaje

Objetivos

- ▶ Describir los conceptos básicos relacionados con los circuitos eléctricos y la programación para el uso de Arduino.
- ▶ Explicar el funcionamiento del experimento inicial con base en los conceptos relacionados con los circuitos eléctricos y con la programación.

41

Resultados de aprendizaje al finalizar el capítulo

- ▶ Identifica los conceptos básicos de circuitos eléctricos y programación aplicados a experimentos con Arduino.
- ▶ Explica el funcionamiento del experimento inicial con fundamento conceptual y científico.
- ▶ Reconoce los elementos eléctricos del experimento inicial y su relación con los diagramas circuitales.

Los circuitos eléctricos

En el capítulo uno se implementó un primer experimento con Arduino: el semáforo. Sin embargo, no se profundizó sobre su funcionamiento, ya que dicha explicación se realizará en el presente capítulo. Para ello, debe entender que cualquier experimento o montaje que realice a través de Arduino tiene dos partes: la primera relacionada con la placa electrónica y sus conexiones (mediante circuitos eléctricos) y la segunda con códigos de programación. En esta sección se darán los conocimientos básicos para entender el funcionamiento de la primera parte: circuitos eléctricos.

Un *círculo eléctrico* se puede entender como la interconexión de distintos elementos eléctricos (como fuentes de voltaje, resistencias, cables, LED, etc.) *que conforman una trayectoria cerrada a través de la cual circula una corriente eléctrica*.

Calma, no se preocupe, seguramente la definición anterior le dejó más dudas que certezas por la presencia de conceptos que no ha estudiado antes, pero que a continuación se desglosarán con detalle.

La carga eléctrica

En primera instancia, todos estos fenómenos eléctricos tienen su fundamento en una propiedad de la materia que se manifiesta a través de fuerzas de atracción o repulsión: *la carga eléctrica*.

El hecho de tener fuerzas de atracción o de repulsión, sugiere la presencia de por lo menos dos tipos de carga eléctrica. Seguramente recordará las convenciones establecidas en clases de física eléctrica de la secundaria: cargas de igual signo se repelen y de signos contrarios se atraen. De esta forma, se ha establecido que en la naturaleza se encuentran dos tipos de carga: las de signo positivo y las de signo negativo.

El voltaje

42

Ahora suponga que tiene una carga eléctrica q_1 de signo positivo y le acerca otra carga eléctrica q_2 del mismo signo, colocándola en el punto P_0 como se ilustra en la figura 33.



Figura 33. Cargas q_1 y q_2

Fuente: elaboración propia.

Según la convención definida, se presentará una fuerza de repulsión entre ambas cargas, por lo cual la carga q_1 se moverá hacia la izquierda y la carga q_2 hará lo mismo, pero hacia la derecha manifestándose así, la fuerza de repulsión mencionada. Por ejemplo, analizando únicamente el movimiento de la carga q_2 , se puede concluir que esta se movió del punto P_0 al punto P_1 como se observa en la figura 34.

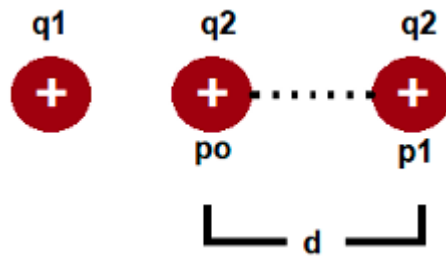


Figura 34. Movimiento de la carga q_2 debido a la fuerza de repulsión entre ella y la carga q_1

Fuente: elaboración propia.

Es importante aclarar que la carga q_1 también se mueve debido a la fuerza de repulsión, pero por simplicidad en el análisis, solo se estudia el movimiento de la carga q_2 .

Con esta claridad, se puede definir el primer concepto clave de circuitos, el voltaje.

Voltaje: energía que se requiere para mover una carga eléctrica de un punto a otro.

La energía está relacionada con la capacidad que tiene una fuerza de producir un movimiento. En el ejemplo mostrado, esa fuerza es la fuerza eléctrica de repulsión (el análisis es similar si la fuerza es de atracción).

De esta forma, la cantidad física que se necesita para mover cargas eléctricas de un punto a otro se conoce como voltaje (el cual lleva de forma implícita en su definición el concepto de fuerza eléctrica). Su unidad de medida en el sistema internacional de unidades son los voltios [V].

43

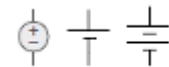
Los elementos más conocidos que entregan voltaje a los circuitos son las fuentes de voltaje, las cuales pueden ser de distintos tipos: baterías, un tomacorriente, etc. En el recuadro de la derecha, se observan los diagramas típicos de una fuente de voltaje, donde es importante definir el terminal positivo y el negativo.

En muchas ocasiones, el terminal negativo suele reemplazarle por la tierra o GND. ¡Téngalo en cuenta!

La corriente eléctrica

Del movimiento de cargas, surge otro concepto fundamental de circuitos: *la corriente eléctrica*.

Corriente eléctrica: velocidad de movimiento de la carga respecto al tiempo. Medida en amperios [A]. Solo hay corriente eléctrica en trayectorias cerradas.



Tres símbolos para representar una misma fuente de voltaje. En los tres casos, todo lo que se conecte arriba estará con polaridad positiva e igualmente, lo conectado con la terminal de abajo tendrá polaridad negativa.



Continuando con el ejemplo de las dos cargas de la figura 33, el movimiento analizado de la carga q_2 se realiza a una velocidad en específico (la posición cambia de P_0 a P_1 en un tiempo determinado), por lo cual, surge una corriente eléctrica (siempre y cuando este movimiento se realice en una trayectoria cerrada).

La resistencia eléctrica

La resistencia eléctrica es otro elemento de circuitos muy utilizado y cuya característica principal es que se opone al paso de la corriente.

Resistencia eléctrica: elemento de circuitos que se opone al paso de la corriente. Se mide en ohm $[\Omega]$.

Una de las características más importantes de la resistencia eléctrica es que es un elemento no polarizado. Es decir, no importa si el terminal positivo de la fuente de voltaje se conecta a la primera terminal de la resistencia, ya que funcionará igual si se conecta primero el terminal negativo. (observe el recuadro de la derecha)

44

Del concepto de resistencia surge la clasificación de dos tipos de materiales: los conductores y los dieléctricos (o aislantes).

Un material conductor tiene una gran cantidad de electrones (cargas eléctricas negativas) libres. De esta forma, al aplicar un voltaje específico mediante el cual se constituye una trayectoria cerrada, esos electrones libres se moverán a una velocidad determinada, produciendo una corriente eléctrica (ver figura 35).

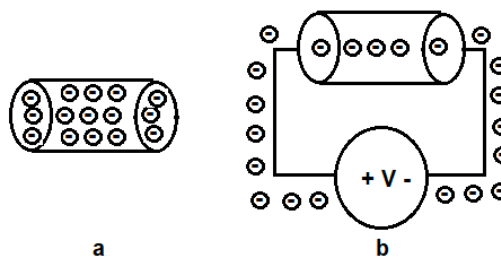


Figura 35. Material conductor **a)** Material conductor (gran cantidad de electrones libres, baja resistencia); **b)** Mismo material con un voltaje que cierra la trayectoria para que los electrones se muevan.

Fuente: elaboración propia.

Como es evidente en la figura 35, al tener una cantidad considerable de electrones libres, y al construir una trayectoria cerrada con una fuente de voltaje (la cual “empuja a los electrones”) se produce una corriente eléctrica (movimiento de electrones a una velocidad determinada).

Por su parte, un material dieléctrico (o aislante) es aquel que tiene una poca cantidad de electrones libres, por lo cual, al aplicar el mismo voltaje de la figura 35, se producirán corrientes eléctricas mucho más pequeñas (menos electrones moviéndose). Este comportamiento se muestra en la figura 36.

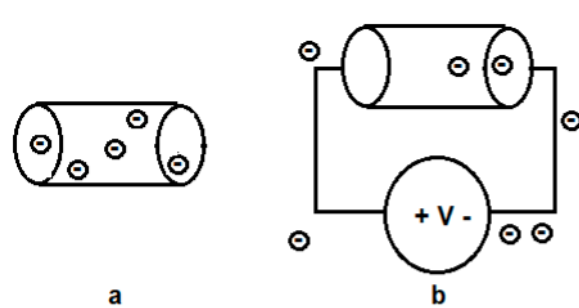


Figura 36. Material dieléctrico

- a)** Material dieléctrico (baja cantidad de electrones libre, alta resistencia);
- b)** Material dieléctrico con una fuente de voltaje que cierra la trayectoria y produce una corriente eléctrica muy pequeña.

Fuente: elaboración propia.

Para entender mucho más estos conceptos y la relación entre los mismos, suponga la siguiente situación:

En el tablero eléctrico se encuentran los *brakers* o interruptores de una casa.

Se tiene un bombillo el cual requiere encenderse. Para ello, se dispone de una fuente de voltaje (por ejemplo, el tablero eléctrico de una casa). Para que el bombillo emita luz, se requiere un flujo de corriente eléctrica a través de este, lo cual se logra conectándolo a través de cables con el tablero eléctrico (la fuente de voltaje). La conexión sería aproximadamente la que se ilustra en la figura 37.

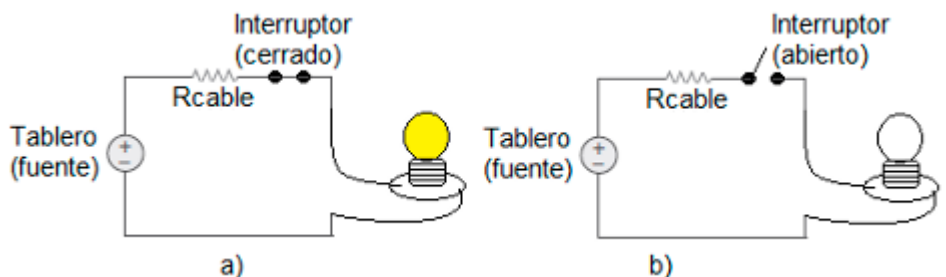


Figura 37. Circuito eléctrico. **a)** Circuito eléctrico para encender el bombillo **b)** Mismos elementos, pero con el interruptor abierto. No hay circuito, ya que no existe trayectoria cerrada.

Para que haya corriente eléctrica se requiere un voltaje que la produzca y una trayectoria cerrada que le permita circular

En este caso, la resistencia del cable es muy pequeña (ya que es un elemento conductor) y sirve para unir la fuente (el tablero) con la carga (el bombillo). De esta forma, la fuente de voltaje “mueve” los electrones libres y produce una corriente eléctrica que, al pasar por el bombillo, produce un flujo luminoso. Note que, al existir la trayectoria cerrada y la fuente de voltaje, existe la corriente eléctrica. Si se tiene la fuente de voltaje, pero no una trayectoria cerrada, la corriente eléctrica no puede fluir y, por lo tanto, no hay brillo en el bombillo.

Si el cable no fuese conductor su resistencia sería muy alta y, por lo tanto, el flujo de corriente con la misma fuente de voltaje conectada sería mucho menor, teniendo menos brillo en el bombillo

El concepto de nodo

Desde la experiencia profesional en la docencia de circuitos y con las sesiones de alfabetización en ciencia y tecnología con docentes (donde se ha tenido la posibilidad de mediar como tutor), se ha observado la existencia de un concepto fundamental, el cual simplifica muchas cosas y ayuda a disminuir la posibilidad de cometer errores en las conexiones eléctricas: el concepto de *nodo*.

Le recomendamos que no solo interiorice el concepto de nodo, sino el de voltaje, corriente y resistencia (vistas anteriormente), así como los conceptos de diodo y LED (que se explicarán más adelante).

Un nodo es un punto de conexión de dos o más elementos de un circuito

Note especial atención en lo sencillo del concepto. Queremos hacerle la recomendación de que lo interiorice y lo tenga muy claro, ya se dará cuenta de la utilidad!

Por ejemplo, en la figura 38 se tiene un circuito donde se identifican los nodos, con ayuda de la definición anterior.

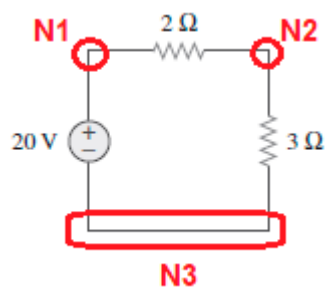


Figura 38. Identificación de nodos en un circuito

Fuente: elaboración propia.

En el nodo 1 se conecta el terminal positivo de la fuente con una terminal de la resistencia de $2\ \Omega$. En el nodo 2 se conecta la otra terminal de la resistencia de $2\ \Omega$ con una de las terminales de la resistencia de $3\ \Omega$. Por último, en el nodo 3 se conecta la otra terminal de la resistencia de $3\ \Omega$ con el terminal negativo de la fuente de voltaje.

Sin embargo, hay circuitos como el de la figura 39a, donde la identificación de los nodos no es tan sencilla y se pueden tener equivocaciones (*identificar más nodos de los que hay en realidad*). Para ello, tenga en cuenta el siguiente consejo

Cuando dos o más nodos se encuentren unidos por un cable sin resistencias ni fuentes, se tratará del mismo nodo

47

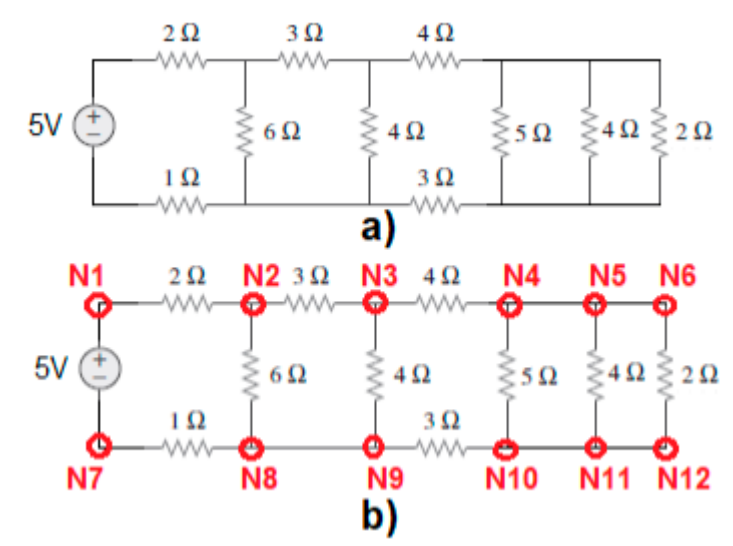


Figura 39. Circuito **a)** Circuito de ejemplo para identificar los nodos.

b) Nodos identificados en el circuito

Fuente: elaboración propia.

Después de identificar los posibles nodos (con ayuda de la definición propuesta) conviene revisar que no se hayan colocado nodos adicionales a los que realmente existen. Para ello, note que en la figura 39b el nodo 8 y el nodo 9 están unidos sin tener en medio resistencias o fuentes, por lo cual será el mismo nodo. A su vez, ocurre lo mismo entre los nodos 10, 11 y 12, los cuales serán el mismo nodo (al no tener en medio de ellos elementos de circuitos). Por último, el nodo 4, nodo 5 y nodo 6 son el mismo nodo, ya que no hay resistencias ni fuentes entre ellos. De tal forma, el número de nodos reales del circuito estarán en la figura 40.

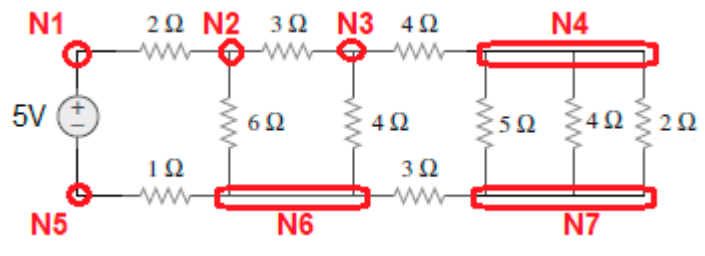


Figura 40. Circuito de ejemplo, con los nodos correctamente identificados.
Fuente: elaboración propia.

Leyes fundamentales

48

El análisis de una gran variedad de circuitos es posible mediante la ley de Ohm y las leyes de Kirchoff (junto con la claridad de los conceptos expuestos anteriormente).

Ley de Ohm

La ley de Ohm afirma que si usted conoce el valor de una resistencia y la corriente que pasa por ella, podrá conocer el voltaje en la resistencia en cuestión a través de la ecuación 1:

$$V = I * R \quad (1)$$

Donde V será el voltaje en la resistencia, I la corriente en la resistencia y R el valor de la resistencia analizada. Lógicamente, es posible determinar la corriente en una resistencia, si se conoce el voltaje que se le aplica y el valor de la resistencia.

Ejemplo: a una resistencia de 220Ω se le aplica un voltaje de 5 V. ¿Cuánto es la corriente que circula por la resistencia?

Despejando la corriente de la ecuación 1, se tendrá que

$$I = \frac{V}{R} = \frac{5V}{220\Omega} = 0.02272A \quad (2)$$

Ley de Kirchhoff de voltaje

Aunque la ley de Ohm es útil, es insuficiente para circuitos más complejos. Por ejemplo, analice el circuito de la figura 41.

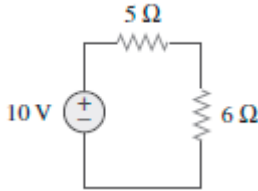


Figura 41. Circuito de ejemplo

Fuente: elaboración propia.

Para analizar un circuito como el de la figura 41, es útil entender la ley de la conservación de la energía: “La energía no se crea ni se destruye, solo se transforma”.

Realmente el voltaje no se consume, se utiliza esta analogía para que quede clara la ley de Kirchhoff de voltaje.

De esta forma, el voltaje producido por la fuente debe ser “consumido” por los receptores (las resistencias). Esto quiere decir que el voltaje de una fuente debe ser igual a la suma de los voltajes en las resistencias de una trayectoria cerrada, lo cual es el enunciado de la ley de Kirchhoff de voltaje.

49

Para que quede más claro, en la figura 40, el voltaje de 10 V de la fuente, debe ser igual a la suma de los voltajes en las resistencias.

¿Cómo se calcula entonces el voltaje en las resistencias? Evidentemente, se realiza a través de la ley de Ohm (ecuación 1), pero, para ello, se requiere conocer la corriente que circula por las resistencias.

Para un circuito como el de la figura 41 (y únicamente con esas conexiones), la corriente se puede calcular con la ecuación 3.

$$I = \frac{\text{Voltaje de la fuente}}{\text{sumatoria de las resistencias}} \quad (3)$$

$$I = \frac{10V}{5\Omega + 6\Omega} = \frac{10V}{11\Omega} = 0.9090A \quad (4)$$

Ya con este valor de corriente, es posible calcular el voltaje en cada resistencia a través de la ley de Ohm (ecuación 1).

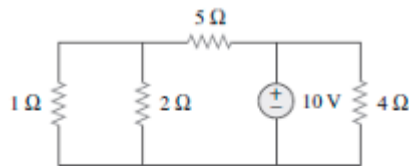
$$V_{R5\Omega} = (0.9090A)(5\Omega) = 4.5454V \quad (5)$$

$$V_{R6\Omega} = (0.9090A)(6\Omega) = 5.454V \quad (6)$$

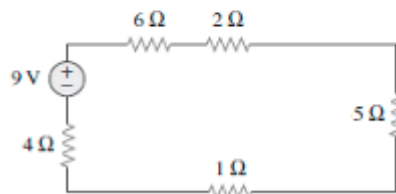
Note que la ley de Kirchhoff de voltaje se cumple, ya que la sumatoria del voltaje en las resistencias es igual al voltaje de la fuente.

Cuando en un circuito solo hay una trayectoria cerrada, se puede utilizar la ecuación 3 para calcular la corriente. Esa corriente circulará por todos los elementos del circuito.

En la figura 42a se observa un circuito donde *no* es válida la ecuación 3, y en la figura 42b otro circuito, donde *sí* es válida dicha ecuación.



a)



b)

Determine la corriente que pasa por el circuito de la figura 41b y los voltajes de cada elemento.

Respuestas:

$I=0,5A$; $Vr6=3V$; $Vr2=1V$;
 $Vr5=2,5V$; $Vr1=0,5V$; $Vr4=2V$.

50

Figura 42. Circuitos con trayectorias cerradas

a) circuito con más de una trayectoria cerrada (no es válido utilizar la ecuación 3).

b) circuito con una trayectoria cerrada (es válido utilizar la ecuación 3)

Fuente: elaboración propia.

El diodo y el LED

Además de los materiales conductores y dieléctricos, existe otro tipo de material bastante particular, el cual ha desempeñado un papel muy importante en el desarrollo de las nuevas tecnologías electrónicas: los materiales semiconductores.

Un material semiconductor en palabras simples es aquel que bajo ciertas circunstancias se comporta como aislante, pero bajo otras se comporta como conductor. Los dos elementos semiconductores que se analizarán son el diodo y el LED.

El diodo

El diodo es un elemento semiconductor el cual permite el flujo de la corriente en un solo sentido. Esto dependerá de la polaridad de la fuente conectada al diodo. Por esta razón, el diodo *sí se considera un elemento polarizado*.

En la figura 43a se tiene un diodo real y en la figura 43b el símbolo de circuitos para un diodo.

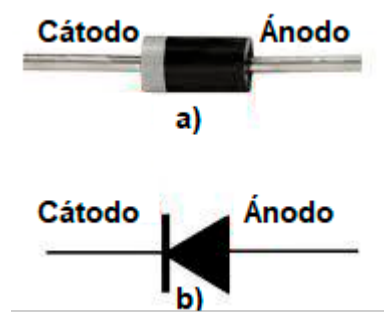


Figura 43. El diodo

a) Diodo real con la identificación de sus terminales; **b)** símbolo de circuitos para un diodo con la identificación de sus terminales

Fuente: elaboración propia.

Ya que el funcionamiento del diodo depende de la forma en la que se conecte, es de especial relevancia aprender a identificar cada una de sus terminales. En el diodo real, el terminal conocido como *cátodo* es el que está identificado con la marca gris. En el símbolo para circuitos del diodo, el cátodo está identificado con una línea. La otra terminal se conoce como *ánodo*.

Después de identificar los terminales de un diodo, se explicarán sus dos formas de polarización: polarización directa y polarización inversa.

- ▶ **Polarización directa:** se presenta cuando el terminal positivo de la fuente de voltaje se conecta al *ánodo* del diodo y el terminal negativo o la tierra (GND) al *cátodo*. De esta forma, el diodo se comportará como un interruptor cerrado y permitirá el flujo de corriente (ver figura 44).
- ▶ **Polarización inversa:** se presenta cuando el terminal positivo de la fuente de voltaje se conecta al *cátodo* del diodo y el terminal negativo o la tierra (GND) al *ánodo*. De esta forma, el diodo se comportará como un interruptor abierto y no permitirá el flujo de corriente (ver figura 45).

El diodo es similar a un interruptor. Funciona como interruptor cerrado en polarización directa y como interruptor abierto en polarización inversa. Recuerde que, para que haya flujo de corriente, se requiere una fuente de voltaje y *una trayectoria cerrada*.

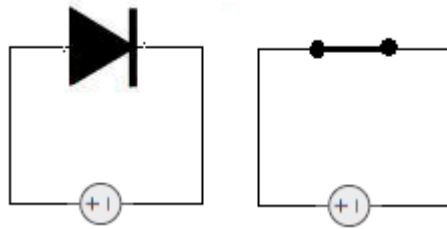


Figura 44. Diodo en polarización directa

Fuente: elaboración propia.

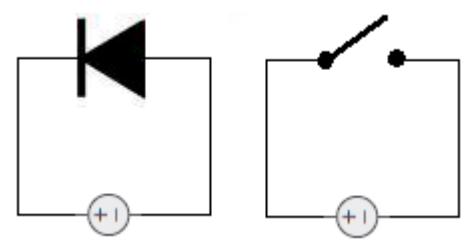


Figura 45. Diodo en polarización inversa

Fuente: elaboración propia.

El LED

52

El LED (*light emitting diode* por sus siglas en inglés) es un diodo, por lo cual también tendrá dos posibles estados de funcionamiento (polarización directa y polarización inversa). La diferencia sustancial es que, al conectarse en polarización directa, emitirá luz. En la figura 46a se identifica un LED real y en la figura 46b el símbolo de circuitos para un LED.

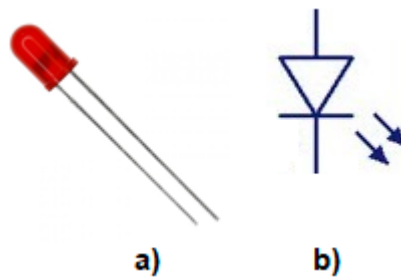


Figura 46. El LED

a) LED real; b) símbolo de un LED para circuitos.

Hay varias formas para identificar el ánodo y cátodo de un LED. Si el LED es nuevo, tendrá una terminal más larga que la otra. La terminal corta será el cátodo y la terminal larga, el ánodo (figura 47). Si por alguna razón las terminales tienen el mismo largo, observe cuidadosamente dentro del cristal. Notará al interior un elemento grueso y otro delgado. El elemento grueso identificará el cátodo del LED y el delgado,

el ánodo (figura 47b). Si tampoco es posible visualizar al interior del LED, mírelo desde arriba y notará que uno de los costados del cristal parece tener un corte; el terminal que se encuentra en ese lado es el cátodo (figura 47c).

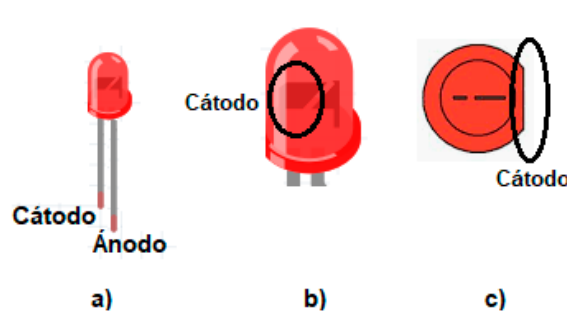


Figura 47. Tres formas para identificar el cátodo del LED

Fuente: elaboración propia.

Conexiones en la protoboard

A continuación, se explicarán las conexiones en una protoboard. Para ello, tenga en cuenta las siguientes consideraciones:

1. La fuente de voltaje será el Arduino. Es decir, de la placa electrónica Arduino Uno saldrá la terminal positiva (cuyo valor siempre será 5 V) y la terminal negativa de la fuente (que siempre será el pin GND). En la figura 48 se observa (encerrado en rojo) de donde es posible sacar los +5 V y la GND del Arduino. Usted puede utilizar cualquiera de las dos GND del Arduino (*pero solo una*).

53

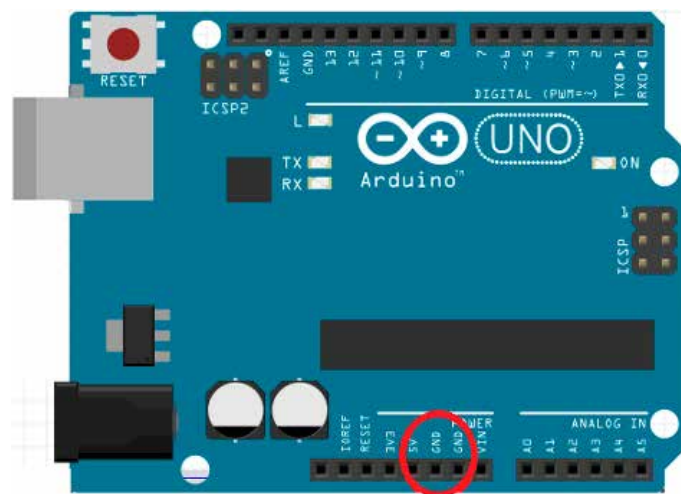


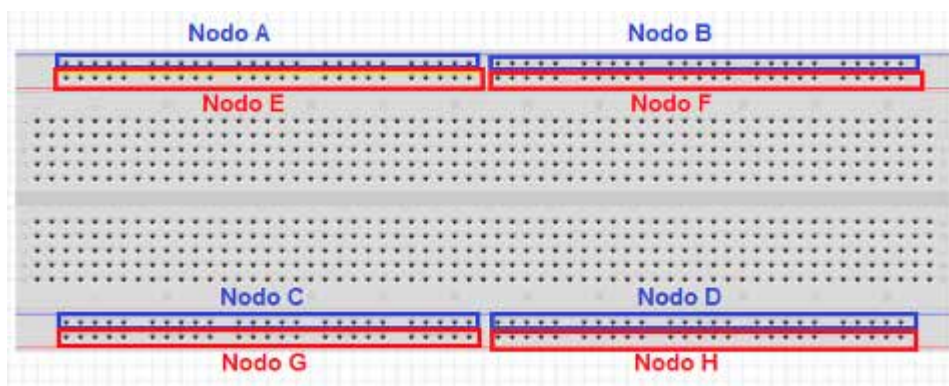
Figura 48. Fuente de voltaje en la placa Arduino Uno

Fuente: elaboración propia.

Además de ello, en ocasiones podrá evitar el uso de los 5 V, ya que los pines digitales 2 al 13 de la parte superior del Arduino en la figura 48, llevan 5 V predeterminados. Es decir, cada vez que usted utilice alguno de esos pines, estará trabajando con +5 V. La GND sí la debe tomar aparte, ya sea de uno de los pines que se encuentran en la parte de abajo del Arduino y encerrados de color rojo o, la GND de la parte superior.

2. Para realizar correctamente las conexiones en la protoboard, debe tener muy claro el concepto de nodo.

En una protoboard se tienen dos tipos de nodos: el primero se distribuye a lo largo de la protoboard y el segundo, a lo ancho de la misma. En la figura 49 se presenta la identificación de cuatro nodos distintos en una misma protoboard.



54

Figura 49. Identificación del primer tipo de nodo en una protoboard

Fuente: elaboración propia.

Como es claro en la figura 49, en la parte superior y en la parte inferior de la protoboard se tienen nodos a lo largo. Si usted conecta el terminal de un elemento en cualquier punto de la región nombrada como nodo A, ese terminal estará en el nodo A. De esa forma, si usted quiere conectar otro elemento en ese mismo nodo, deberá ubicar una de sus terminales en cualquier otro punto de la misma región llamada nodo A. Por ejemplo, si desea conectar dos resistencias como aparece en la figura 5, su conexión en los nodos superiores o inferiores podría realizarse como aparece en la figura 51.

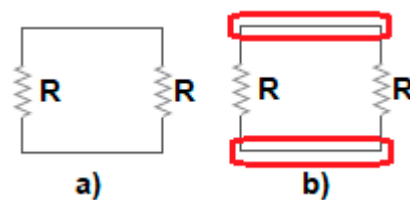


Figura 50. Conexión entre dos resistencias

- a) Diagrama de circuitos de una conexión entre dos resistencias;
- b) Nodos identificados en la conexión.

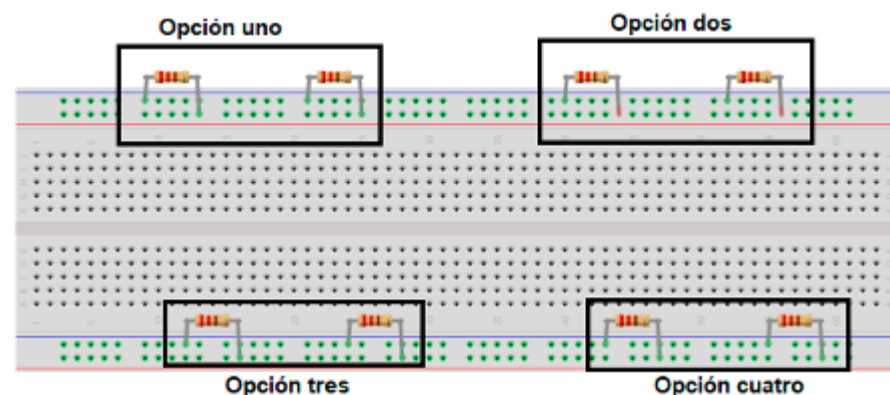


Figura 51. Conexión de las resistencias de la figura 50, según los nodos de la protoboard identificados en la figura 49.

A su vez, los nodos A-B, C-D, E-F y G-H pueden unirse entre sí a través de un cable, dejando cuatro nodos que abarcan todo el largo de la protoboard (figura 52).

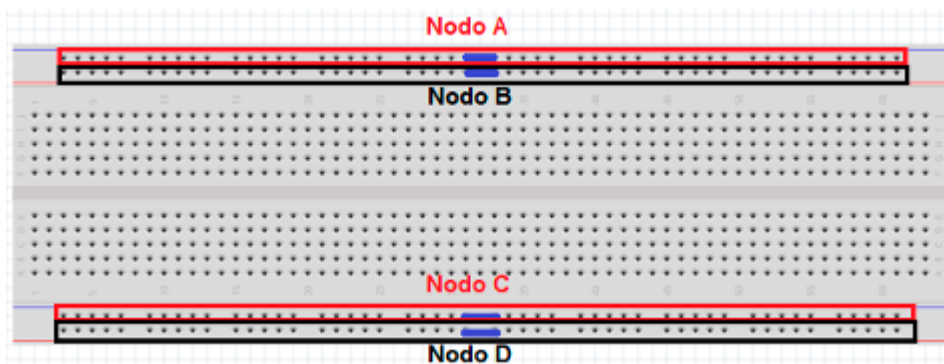


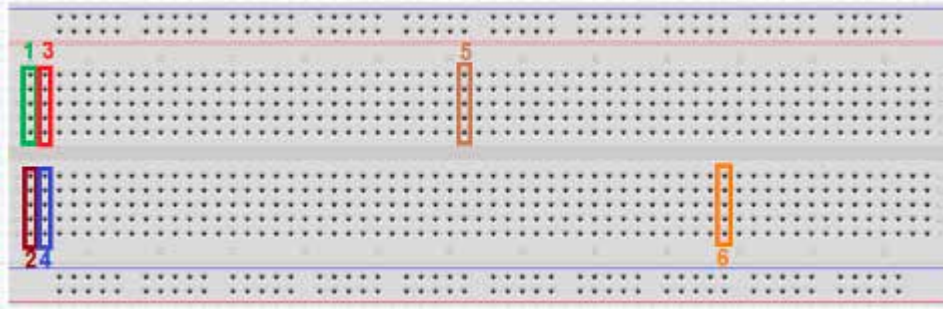
Figura 52. Protoboard con cuatro nodos a lo largo, resultado de unir pares de nodo de la figura 49 mediante cable azul.

Fuente: elaboración propia.

La forma más sencilla de trabajar con estos nodos es definir el par de arriba o el par de abajo como los nodos de la fuente de voltaje. Por ejemplo, tomar el nodo A o el nodo C como +5 V (también conocido como +VCC) conectando en uno de ellos la terminal positiva de la fuente de voltaje. Y a su vez, tomar el nodo B o el nodo D como la terminal negativa (o GND) de la fuente de voltaje.

Otra forma de llamar al terminal positivo de una fuente de voltaje es VCC.

Después de identificar el primer tipo de nodo presente en una protoboard, se muestra en la figura 53 el segundo tipo de nodo, que se ubica a lo ancho de la misma y se encuentra en el centro.



En la figura 53 solo se identifican 6 nodos tipo dos de los 126 existentes.

Figura 53. Ubicación de seis nodos en la parte central de la protoboard.
Fuente: elaboración propia.

Como se da cuenta, usted puede trabajar en cualquier parte de la protoboard, siempre y cuando tenga clara la disposición de los nodos.

3. Debe entender los diagramas eléctricos, los cuales pueden ser de dos tipos.

A continuación, se muestra un diagrama eléctrico que se desea colocar en la protoboard (figura 54).

56

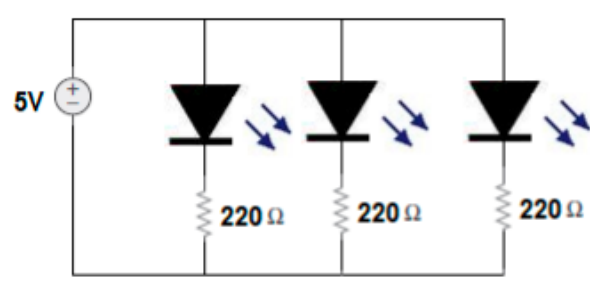


Figura 54. Circuito que se desea colocar en un protoboard.
Fuente: elaboración propia.

En la figura 55, se presentan los nodos identificados de la figura 54.

Recuerde que un *nodo* es el punto de conexión de dos o más elementos de un circuito.

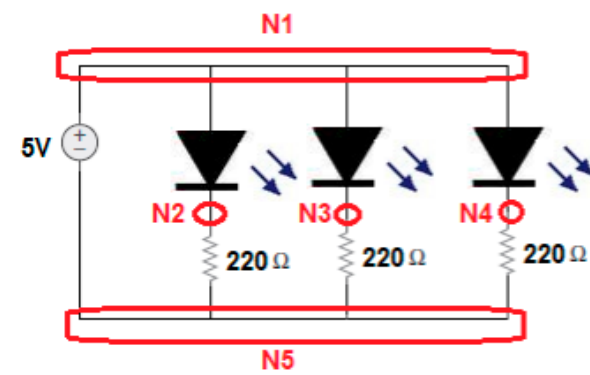


Figura 55. Nodos identificados del circuito de la figura 54.
Fuente: elaboración propia.

La conexión en la protoboard de dicho circuito se ilustra en la figura 56.

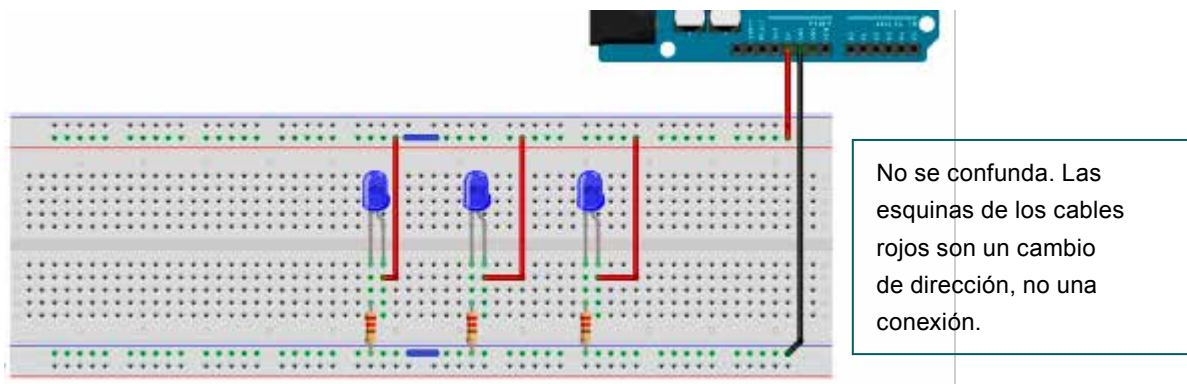


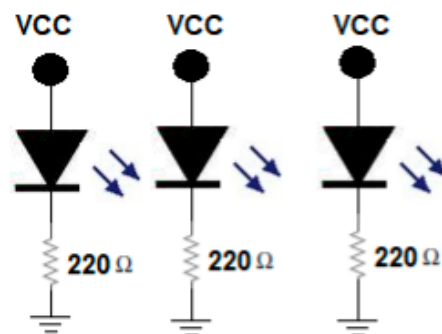
Figura 56. Conexión en la protoboard del circuito de la figura 54.

Fuente: elaboración propia.

Note que en la figura 56 se tienen unidos dos de los nodos a lo largo de la parte superior, formando uno solo y dos de los inferiores, formando uno solo también (a través de un cable azul en la mitad). El cable negro es el negativo (o la GND) de la fuente y va conectado al primer nodo de la parte inferior. Todo lo que se conecte a un punto de ese nodo, estará conectado con el negativo de la fuente (según el plano, una de las terminales de cada resistencia, debe ir a este nodo). El cable rojo que sale del Arduino es +5 V y se conecta al segundo nodo de la parte superior de la protoboard. Ya que está unido mediante un cable en la mitad, todo lo que se conecte en un punto de esa hilera, estará conectado al terminal positivo de la fuente de 5 V. Después, se ubican los terminales de cada LED según el plano, ubicando el cátodo en el mismo nodo de la resistencia y uniendo mediante el cable rojo el ánodo de cada LED con el nodo de la terminal positiva de la fuente de voltaje.

57

El mismo diagrama eléctrico de la figura 54 se puede redibujar como aparece en la figura 57.



Abajo se ilustran dos símbolos de la tierra (GND, negativo de la fuente de voltaje o Ground).

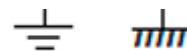


Figura 57. Otro diagrama para representar el mismo circuito de la figura 54.

Fuente: elaboración propia.

En la figura 57 se tiene el mismo circuito de la figura 54, por lo cual su implementación en protoboard sería la misma. Recuerde que VCC hace referencia al terminal positivo de la fuente de voltaje y que en el Arduino es el pin de 5 V. También tenga presente que el terminal negativo es la GND del Arduino o la tierra (*Ground* o GND). Note que todas las resistencias están conectadas a la tierra (negativo de la fuente) y que los ánodos de cada LED se encuentran conectados con el positivo de la fuente (VCC), por lo cual se tendría el mismo circuito de la figura 52.

4. Debe proteger el LED con una resistencia limitadora.

Note que siempre que se colocó un LED, se le conectaba en uno de sus extremos una resistencia. El valor de esa resistencia se asume en $220\ \Omega$ y el motivo de su presencia en la conexión es que el voltaje de funcionamiento de un LED depende de su color y está aproximadamente en 3.6 V. Si usted conecta directamente los 5 V y la GND al LED, este se dañará, ya que no está diseñado para soportar un voltaje de ese valor. Por ello, se le conecta una resistencia, de tal forma que la conexión resultante sería la de la figura 58.

Recuerde que este texto busca que usted tome decisiones con criterio. Por ello, es suficiente con que entienda por qué debe colocar una resistencia conectada con un LED. El cálculo del valor de resistencia no es necesario.

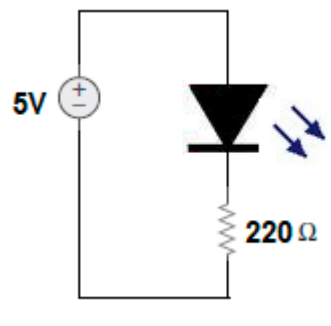


Figura 58. Circuito para encender un LED

Como el LED está en polarización directa, se va a comportar como un interruptor, permitirá el paso de la corriente y se iluminará. El LED necesitará un voltaje particular (por ejemplo de 3.6 V) y por la ley de Kirchoff de voltaje debe colocarse una resistencia que “consume” los 1.4 V restantes. Para cada color de LED se tiene un voltaje en particular y una corriente máxima que soporta, por lo cual para cada color debería colocarse una resistencia distinta que permita obtener el brillo máximo de cada LED. Sin embargo, con una resistencia de $220\ \Omega$ se obtiene un brillo óptimo para la mayoría de los colores.

Solo a manera de ejemplo, se realizará el cálculo de resistencia para un LED de color verde.

El LED verde requiere para funcionar un voltaje de 3 V y una corriente de 0.02 A. Al aplicar la ley de Kirchoff de voltaje en el circuito de la figura 58 (suponiendo que aún no se conoce el valor de la resistencia), se tendría la siguiente ecuación:

$$5V = 3V + V_{resistencia} \quad (7)$$

Donde el voltaje de la resistencia sería de 2 V.

La corriente que debe circular por el LED debe ser de 0.02 A y en dicho circuito, la corriente será la misma en todos los elementos, por lo tanto, la corriente en la resistencia también será de 0.02 A. Al aplicar la ley de Ohm y despejar la resistencia de la ecuación 1, se tendrá que el valor en ohm de la resistencia limitadora debe ser de 150 Ω .

Montaje del semáforo

El diagrama eléctrico del semáforo montado en el capítulo uno, se muestra en la figura 59.

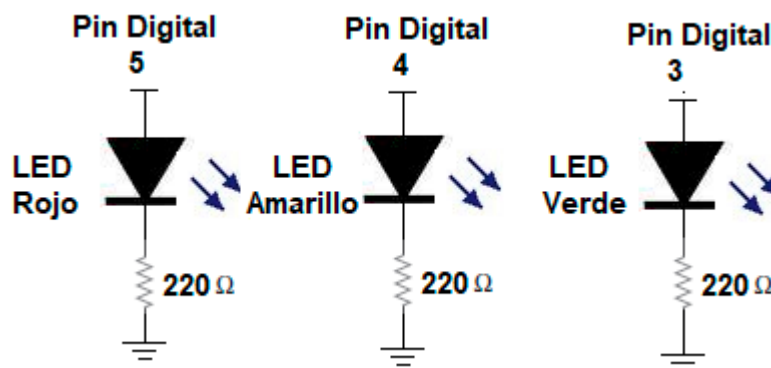


Figura 59. Diagrama eléctrico del semáforo

Fuente: elaboración propia.

Note que el diagrama del circuito es exactamente el mismo de la figura 57, con la diferencia que el ánodo de cada LED no va a +5 V del Arduino sino a tres pines digitales distintos. Recuerde lo que se mencionó con anterioridad: *los pines digitales tienen por defecto +5 V cada uno*. De esta forma, cada LED quedaría energizado con +5 V también. Así, el montaje real de este diagrama sería tal como se ilustró en el capítulo 1 (figuras 7 y 8).

Cabe recordar las advertencias sobre las conexiones que se hicieron en el capítulo uno. Por ejemplo, si el LED se conectaba de forma incorrecta (invertido a como indica el diagrama), siempre estará en polarización inversa, por lo cual nunca emitiría luz. Si

usted no conectaba el GND como lo muestra el diagrama, el circuito no cerraría y no habría flujo de corriente por lo cual tampoco funcionaría el LED.

También se hacían recomendaciones respecto a la ubicación de las conexiones en la protoboard, para evitar las divisiones con las que viene por defecto. Por favor, analice las conexiones de la figura 7 del primer capítulo y verifique que coinciden con el diagrama de la figura 59.

Ya con la conexión establecida, los tres LED quedarán funcionando de forma ininterrumpida. Lo que se debe hacer a continuación es la configuración mediante el código en la plataforma de desarrollo de Arduino IDE para definir tiempos de encendido, apagado y de espera.

Desarrollando el código

El código desarrollado en Arduino IDE cumple un papel fundamental, ya que es el “cerebro” detrás de las conexiones electrónicas realizadas. Para continuar con él, se deben considerar dos tipos de variables que se manejan a través del código, y cuyo tratamiento es distinto: las variables analógicas y las digitales.

Una variable análoga es aquella cuya representación se hace de forma continua (sin interrupciones en el tiempo), por lo cual puede tomar cualquier valor. En su defecto, una variable digital es aquella que *solo puede tomar dos estados*. Para ilustrarlo con más claridad, piense en el siguiente ejemplo: se realizan mediciones de temperatura en diez horas diferentes, a partir de las cuales se obtuvieron los siguientes valores: 9° C, 15° C, 13° C, 13.5° C, 12° C, 11.7° C, 8° C, 10° C, 12° C, 16° C.

Si usted grafica dichos valores de temperatura en función del tiempo, obtendría la gráfica que se ilustra en la figura 60.

60

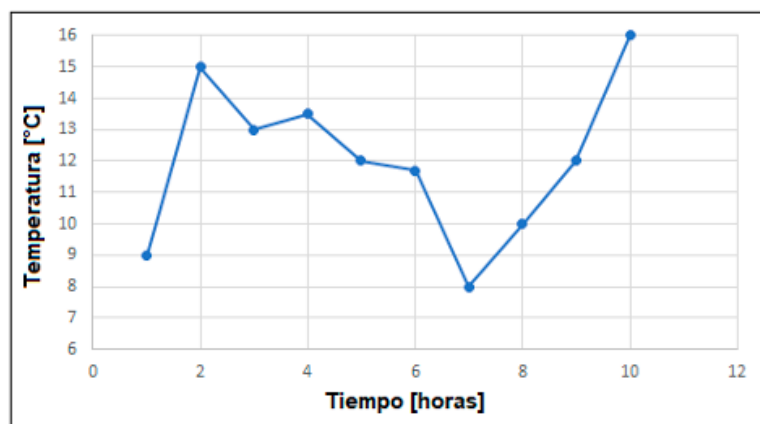


Figura 60. Ilustración de una gráfica continua

Fuente: elaboración propia.

Como se observa claramente en la figura 60, la temperatura tiene valores puntuales en cada hora estudiada, pero no hay interrupciones en puntos intermedios, es decir, también habrá valores de temperatura entre cada hora estudiada. Este es un ejemplo de una variable análoga y su comportamiento gráfico (continuo o sin interrupciones).

Ahora, analice la misma situación, con las mismas mediciones y en las mismas horas. Sin embargo, el interés ahora es proteger un alimento cuya temperatura debe estar siempre en 12°C , permitiendo oscilaciones entre $+2^{\circ}\text{C}$ y -2°C . Es decir, la temperatura a la cual el alimento puede conservarse es entre 10°C y 14°C (dos grados por encima y dos grados por debajo de la temperatura deseada). Con esa información, usted desea saber en cuáles horas se superó la temperatura mínima o máxima. Para ello, la gráfica toma la forma de la figura 61.

A diferencia de las variables análogas, no hay valores intermedios entre los tiempos medidos. Es decir, se tiene un estado en las horas en cuestión, pero ninguno entre horas.

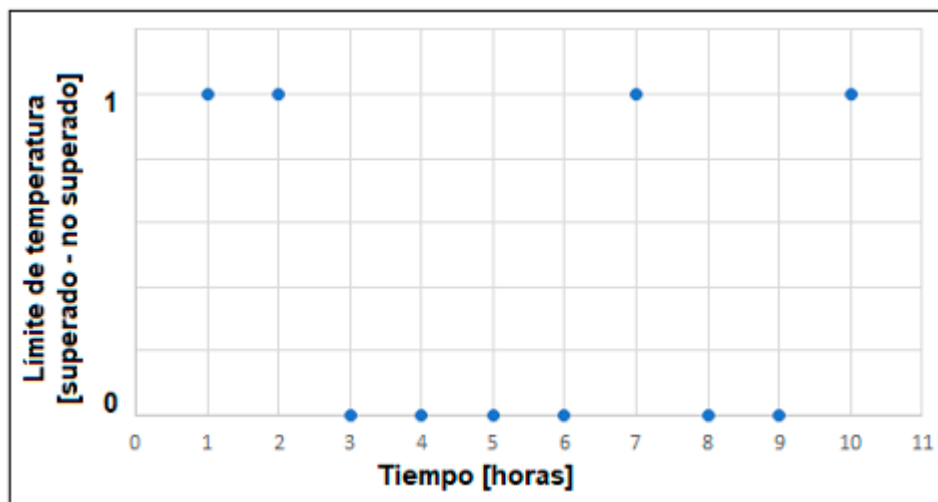


Figura 61. Representación gráfica de una variable digital

Fuente: elaboración propia.

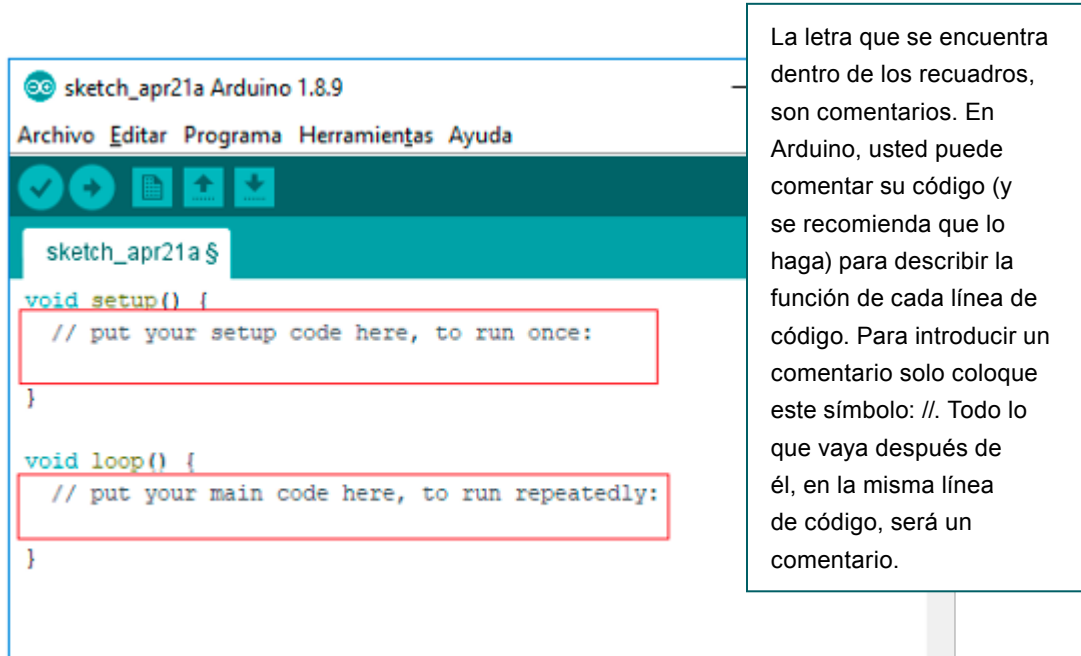
Note que, a diferencia de la figura 60, en cada hora, la variable analizada (límite de temperatura) solo puede tomar dos posibles valores: superado o no superado. Por lo tanto, la gráfica deja de ser continua. Cuando una variable analizada solo puede tener dos estados, se conoce como digital. En el caso de la figura 61, el valor 1 del eje y representa que el límite de temperatura inferior o superior ha sido superado, y el 0 representa que el valor de temperatura en dicha hora se mantiene dentro de los límites mínimos y máximos.

Partes del código

En la página principal de la aplicación de Arduino, vamos a realizar la explicación del código (también conocido como Sketch). Lo primero que se debe considerar es que,

para el funcionamiento correcto del código, debe guardarse en una carpeta que tenga el mismo nombre del *sketch*. Esto se hace de forma automática al momento de guardar el archivo como se mostró en el capítulo 1 (figura 22 y 23).

En la figura 62, se tienen las dos secciones que aparecen por defecto al abrir un nuevo archivo con la aplicación de Arduino IDE.



La letra que se encuentra dentro de los recuadros, son comentarios. En Arduino, usted puede comentar su código (y se recomienda que lo haga) para describir la función de cada línea de código. Para introducir un comentario solo coloque este símbolo: //. Todo lo que vaya después de él, en la misma línea de código, será un comentario.

62

Figura 62. Interfaz de inicio de Arduino IDE.

Fuente: elaboración propia.

Cada uno de los títulos de estas dos secciones finaliza con un juego de paréntesis (). No escriba nada en ellos. El código debe ir incluido entre los corchetes {} de cada sección.

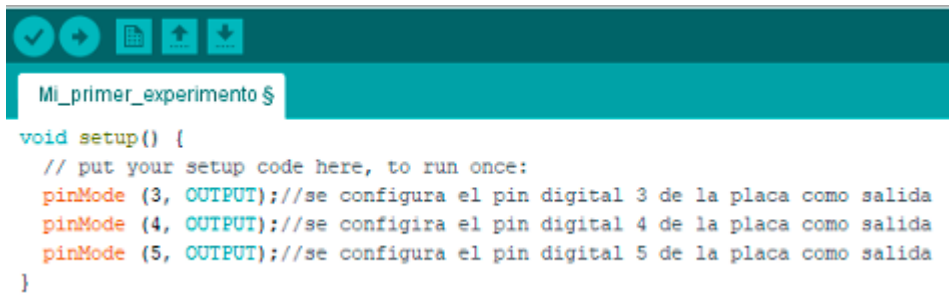
Las dos partes que aparecen por defecto y se deben mantener durante el desarrollo del código es el *void setup ()* y el *void loop ()*.

Note que, después del *void setup ()* aparece un corchete {, que se cierra unas líneas abajo con el símbolo }. Estos dos corchetes encierran la sección (primer recuadro rojo) donde irá el código cuya ejecución se debe realizar una única vez. Por ejemplo, definir una salida, configurar los pines de entrada. Esta configuración dependerá del proyecto. Lo mismo ocurre con el *void loop ()*, con la diferencia de que el código que se utilizará en el segundo recuadro rojo será el que se ejecute de forma

cíclica y sin detenerse. En otras palabras, en la sección *void loop* deberá ir el código *pensante* o aquel que dice *qué hacer* con las variables (entradas y salidas) configuradas en el *void setup*.

Explicación del código implementado

En la figura 63, se tiene el código colocado en la sección *void setup ()* para el semáforo.



```
Mi_primer_experimento $  
void setup() {  
  // put your setup code here, to run once:  
  pinMode (3, OUTPUT); //se configura el pin digital 3 de la placa como salida  
  pinMode (4, OUTPUT); //se configura el pin digital 4 de la placa como salida  
  pinMode (5, OUTPUT); //se configura el pin digital 5 de la placa como salida  
}
```

Figura 63. Código del semáforo de la sección *void setup ()*.

Fuente: elaboración propia.

En esta sección, se definen los pines digitales 3, 4 y 5 de la placa Arduino como salidas. Para ello, la sintaxis que debe utilizar es la palabra *pinMode* (sin espacios y con todo en minúscula, menos la letra M que deberá estar mayúscula). Después, debe ir dentro de paréntesis el número del pin que desea configurar, inmediatamente una coma y, por último, la palabra *OUTPUT* en mayúscula. *Cada línea de código debe finalizar con un punto y coma. ¡No lo olvide!*

Una forma de notar que no hay errores en el código escrito es que la palabra *pinMode* aparece de color naranja y la palabra *OUTPUT*, de azul claro.

No importa el pin que usted escoja, lo único que debe tener en cuenta es que el pin utilizado en la placa debe coincidir con el nombrado en esta sección del código.

Por último, también puede configurar entradas, para ello, en lugar de la palabra *OUTPUT*, debe colocar la palabra *INPUT*. Para el caso del semáforo, los pines deben ser digitales y todos configurados como salidas (es decir, desde el código se le indicará qué hacer al pin correspondiente).

En la figura 64, aparece el código de la sección *void loop ()* para el semáforo.

Quando se requiera colocar pines analógicos, coloque el nombre de uno de ellos, en lugar de los números 3, 4 o 5. Los nombres de los pines analógicos son: A0, A1, A2, A3, A4 y A5. Además, usted puede emplear otros pines digitales de la placa, en lugar de los utilizados.

```

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(5, HIGH); //se enciende el LED rojo
  delay (6000); //se esperan 6 segundos
  digitalWrite (4, HIGH); //se enciende el LED amarillo
  delay (3000); //se esperan 3 segundos
  digitalWrite (3, HIGH); //se enciende el LED verde
  digitalWrite (5, LOW); //se apaga el LED rojo
  digitalWrite (4, LOW); //se apaga el LED amarillo
  delay (7000); //se esperan 7 segundos
  digitalWrite (3, LOW); //se apaga el LED verde
  digitalWrite (4, HIGH); //se enciende el LED amarillo
  delay (2000); //se esperan 2 segundos
  digitalWrite (4, LOW); //se apaga el LED amarillo
}

```

Figura 64. Código del semáforo desde la sección `void loop()`.

Fuente: elaboración propia.

En esta sección se tienen dos comandos particulares: `digitalWrite` y `delay`.

Lo que se realizará con el `digitalWrite` es configurar un voltaje de 5 V sobre el pin correspondiente si se encuentra en HIGH, o de 0 V si se encuentra en LOW. De esta forma, la primera línea de código en la figura 64

64

Recuerde que está trabajando con pines digitales. Por lo explicado anteriormente, este tipo de pines solo puede tener dos estados. En el caso de Arduino, su estado será HIGH O LOW.

“`digitalWrite(5, HIGH); //se enciende el LED rojo`” coloca 5 V sobre el pin digital 5 de la placa Arduino. Así, todo lo que se conecte a este pin, estará con 5 V. Si usted revisa su montaje, el pin 5 está conectado al LED rojo, por lo cual al encontrarse en estado de HIGH estará encendido.

Para que el comando `digitalWrite` funcione, debe estar todo en minúscula, menos la letra W, que debe ir en mayúscula. Después debe ir un juego de paréntesis donde irá el número del pin (debe ser uno configurado en el `void setup`) y, separado por una coma, el estado HIGH o LOW del mismo.

No olvide el punto y coma para finalizar cada línea de código. Si los omite, ¡Arduino mostrará errores en el código!

En la segunda línea, se utiliza el segundo comando empleado en el código: la instrucción `delay`, la cual permite realizar una espera en la ejecución del código. Para que quede mucho más claro, ubíquese en la línea anterior. En dicha línea se define el estado de HIGH en el LED rojo. Después, aparece el `delay` y, en la siguiente se enciende el LED amarillo. Si el `delay` no estuviese

en el código, se encendería inmediatamente el LED amarillo (al igual que el rojo). Sin embargo, eso no ocurre en un semáforo real, donde la luz roja se mantiene encendida

y, después de un tiempo, se enciende la luz amarilla (siguiendo encendida la roja). Esa espera, se consigue a través del *delay*. Lo único que se debe hacer es colocar la palabra en minúscula y, dentro de paréntesis, el tiempo de espera que se requiere *en milisegundos*. Por lo tanto, el tiempo de espera para encender el LED amarillo mientras permanece encendido el LED rojo es de seis mil milisegundos, lo cual equivale a 6 segundos (según la figura 64).

El funcionamiento del código será: encender el LED rojo, esperar seis segundos y encender el LED amarillo (manteniendo encendido el rojo). Esperar tres segundos, encender el LED verde y apagar el LED rojo y amarillo. Esperar siete segundos, apagar el LED verde y encender el LED amarillo, esperar dos segundos y apagar el LED amarillo.

Note que el código termina en apagar el LED amarillo. ¿Por qué sigue funcionando el semáforo ininterrumpidamente? Muy sencillo, recuerde que el código de la sección *void loop* se repite indefinidamente. Además, el código se ejecuta línea por línea, de esta forma, al llegar al final, la ejecución se devuelve a la primera línea para ejecutarse nuevamente.

Sensores

“¡Oh, qué maravilla! ¡Cuán bella es la humanidad!”
William Shakespeare. *La Tempestad*, Acto v. [fin de epígrafe]

Objetivos del capítulo y resultados de aprendizaje

Objetivos

- ▶ Describir los conceptos básicos relacionados con la instrumentación y la teoría del error.
- ▶ Analizar la teoría del error en el contexto de prácticas de laboratorio, de tal forma que se fortalezcan los procesos formativos.

67

Resultados de aprendizaje al finalizar el capítulo

- ▶ Reconoce los conceptos relacionados con la teoría del error y su utilidad en las prácticas de laboratorio.
- ▶ Distingue las principales características relacionadas al error en un sensor básico.

Magnitudes físicas

La física, la química y la biología son ciencias experimentales mediante las cuales se busca descubrir generalidades sobre la estructura y el comportamiento del Universo y la naturaleza. Para lograr interpretaciones útiles de la realidad se utilizan los *sistemas*, los cuales no son más que la interacción de un conjunto de objetos, que por lo general se pueden representar mediante un modelo matemático. Para realizar la interpretación de los sistemas, se estudian *únicamente* las propiedades relevantes de los objetos que están relacionadas con el fenómeno que se va a analizar. Por ejemplo, si se desean estudiar las propiedades eléctricas de un motor, es importante tener claros conceptos de electromagnetismo (campo magnético, ley de Faraday), sin embargo, si

Por lo general, un sistema se puede especificar de acuerdo con la disciplina que lo analice: sistema físico, sistema biológico o sistema químico.

Un sistema es un conjunto de objetos relacionados entre sí. Se puede representar por un modelo matemático

el objetivo es analizar el movimiento realizado por el mismo motor y su trayectoria circular, los conceptos para tener en cuenta no son del electromagnetismo, sino de la cinemática (posición, velocidad y aceleración).

En el entorno podemos encontrar diferentes ejemplos de sistemas, como una estrella, un haz luminoso, una mezcla, el aparato digestivo, entre muchos más. *Para cada una de las ciencias experimentales y sus respectivas ramas, se tienen sistemas que permiten estudiar los fenómenos que se buscan explicar.*

En la figura 65a se observa un circuito eléctrico, mediante el cual se puede representar una instalación eléctrica. Por ejemplo, una resistencia puede representar un televisor (solo como ejemplo representativo y simplificado) conectado a un tomacorriente (fuente de voltaje) como en la figura 65b.



68

Figura 65. Ejemplo de circuito eléctrico

a) circuito eléctrico como representación de un TV conectado a un tomacorriente.

b) El TV y el tomacorriente representado en a),

Fuente: elaboración propia.

Como se mencionó anteriormente, la física, la química y la biología son ciencias experimentales y, para realizar los experimentos, analizarlos, plantear hipótesis y llegar a resultados (aplicar el método científico), se requiere hacer mediciones.

La principal característica de las magnitudes físicas es que estas son cuantificables.

A través de las mediciones es posible realizar comparaciones entre las diferentes variables que intervienen en el sistema estudiado con el experimento. Una *magnitud física* es la propiedad de un cuerpo o fenómeno natural que puede ser medida, es decir, se puede cuantificar.

Por ejemplo, la masa, la longitud o el tiempo son algunas magnitudes físicas. Por el contrario, otros fenómenos como el sabor o el amor no son magnitudes físicas y, por ende, no se pueden cuantificar.

Mediciones

Como se ha mencionado, en la experimentación y, por lo tanto, en el desarrollo de la ciencia, es fundamental realizar mediciones. Sin embargo, una pregunta que se puede plantear es: ¿Qué es medir?

Una *medición* es la comparación de una cantidad con un estándar de referencia. Por ejemplo, al decir que el “huso horario” en Colombia es de -5 horas (GMT), se está realizando una comparación entre la hora real (medida) de Colombia con una hora de referencia, que en el caso de la geografía es el tiempo del meridiano de Greenwich o GMT, por sus siglas en inglés. De esta forma, los estándares definen una unidad de la cantidad estudiada.

Una medición es una comparación entre una cantidad y un estándar.

Otro ejemplo es la estatura de una persona. Si se dice que Juan mide 1.85 metros, eso implica que la estatura de Juan es 1.85 veces más larga que una vara de metro. En este caso, la comparación se realiza entre la estatura (medición) de Juan con una referencia (metro).

Un estándar define una unidad de la cantidad estudiada.

Instrumentos de medición

Las mediciones se realizan a través de instrumentos de medición. Para garantizar que la medida de un instrumento sea confiable, estos deben encontrarse calibrados de acuerdo con un instrumento referencia.

Por ejemplo, la medida de un día en la Tierra (en horas, minutos y segundos) se realiza mediante un reloj atómico. Este reloj se “calibra” de acuerdo con el movimiento de rotación del planeta.

Existen instrumentos de medición más sofisticados que otros. Esto depende de la magnitud física que se vaya a medir. Por ejemplo, el desplazamiento se puede medir con una regla y el tiempo con un cronómetro. Pero otra variable física como la velocidad requiere un instrumento más sofisticado para su medición: el velocímetro. Sin embargo, como la velocidad es la relación entre el desplazamiento y el tiempo, en lugar de utilizar un velocímetro se pueden utilizar las medidas de la regla y el cronómetro (realizando la operación matemática para obtener el valor de la variable de interés).



Un cronómetro es un instrumento de medición, mediante el cual se puede medir el tiempo.

Manejo de errores: recurso para análisis en las prácticas de laboratorio

En las prácticas de laboratorio, es imposible evitar la incertidumbre (o error) en las mediciones. El error en una medición se puede producir por:

- Factores humanos: la apreciación particular de una persona.

El objetivo de esta sección es enriquecer el desarrollo de su labor como docente de ciencias experimentales. Se ha logrado evidenciar que pocos profesores transmiten a sus estudiantes las nociones de errores en las mediciones. ¡Es tiempo de mejorar la práctica docente!

- ▶ Factores del medio: un experimento no se puede repetir dos veces de manera exacta.
- ▶ Factores de calibración: el instrumento de medición se encuentra mal calibrado.

Por ejemplo, si al utilizar un amperímetro (instrumento para medir la corriente eléctrica), se obtiene una medida de 30 A, a esta medida se le debe adicionar el porcentaje de error que indique el fabricante del instrumento. Por ejemplo, si el fabricante del amperímetro advierte que su instrumento introduce un error a la medición de 0.05 A, la medición debe expresarse como 30 ± 0.05 A.

Exactitud de una medida

La exactitud define que tan cerca se encuentra de la referencia una medición realizada.

La exactitud de una medida se define como la proximidad de una medida al valor de referencia o conocido. Por ejemplo, el nivel de voltaje residencial en Colombia es de 120 V, este valor será la referencia. Al utilizar un voltímetro (instrumento para medir el voltaje), la medida de este instrumento será mucho más exacta, cuanto más se aproxime a los 120 V.

70

Del concepto de exactitud, surgen los conceptos de *error absoluto* y *error relativo*.

- ▶ **Error absoluto:** valor absoluto (cantidad mayor a cero) de la diferencia entre el valor obtenido en una medición y el valor que se toma como referencia.

Matemáticamente, el error absoluto se define en la ecuación 8:

$$\text{Error absoluto} = |\text{Valor medido} - \text{Valor de referencia}| \quad (8)$$

- ▶ **Error relativo:** cociente entre el valor absoluto y la referencia.

Matemáticamente, el error relativo se define en la ecuación 9:

$$\text{Error relativo} = \frac{|\text{Valor medido} - \text{Valor de referencia}|}{\text{Valor de referencia}} \quad (9)$$

Precisión de una medida

La precisión de una medida se define como la dispersión de los valores obtenidos de las mediciones *repetidas* de una magnitud. Entre más precisa es la medida, las mediciones son menos dispersas. En otras palabras, cuando un instrumento realiza la

medición de una magnitud en *repetidas ocasiones y bajo las mismas circunstancias*, la medida debería ser la misma, lo cual indica una precisión en el instrumento.

Para ilustrar mejor el concepto de precisión, se tienen dos voltímetros: voltímetro A y voltímetro B. Con cada uno de los voltímetros se realizan cinco mediciones bajo las mismas circunstancias como se observa a continuación:

- ▶ Voltímetro A: 120 V - 120 V - 121.05 V - 120.9 V - 120 V.
- ▶ Voltímetro B: 122 V - 120.9 V - 119.08 V - 120 V - 118.5 V.

Como se puede observar, en tres de las cinco mediciones, el voltímetro A midió 120 V, en contraparte con el voltímetro B, que nunca repitió un valor de medición. En este ejemplo, el voltímetro A es más preciso que el voltímetro B.

Procedimiento para mediciones en un laboratorio

En cualquier disciplina donde se requieran prácticas de laboratorio es importante verificar los errores de las mediciones realizadas en un experimento. Para ello, se propone el siguiente procedimiento:


1. Tomar varias medidas de la(s) variable(s) de interés.
2. Definir la referencia (por lo general, se realiza a través de cálculos teóricos).
3. Calcular el valor promedio de las mediciones con ayuda de la ecuación 10 (promedio aritmético):


$$\bar{x} = \frac{x_1 + x_2 + x_3 + x_4 + \dots + x_n}{n} \quad (10)$$

Para cualquier número de mediciones, solo se debe tener en cuenta que el denominador de la ecuación 10 es el número de mediciones que se realiza; y el numerador es la sumatoria de las mediciones realizadas.

4. Se calcula la desviación de los datos tomados en las mediciones respecto al promedio con la ecuación 11.

$$DM = \frac{|x_1 - \bar{x}| + |x_2 - \bar{x}| + |x_3 - \bar{x}| + \dots + |x_n - \bar{x}|}{n} \quad (11)$$

a) 

b) 

Tiro al blanco. Entre más cerca estén los tiros de dar en el blanco (centro) los tiros son exactos. Por otro lado, sin importar que tan cerca estén del blanco (siempre y cuando todos los disparos estén cerca entre sí), se dice que los tiros fueron precisos.

En a) se tiene una precisión baja pero una alta exactitud.

En b) se tiene una exactitud baja pero una alta precisión.

5. Finalmente, se expresará la medición de la variable de interés como

$$\text{Variable} = \bar{x} \pm DM \quad (12)$$

6. Se calcula el error relativo con ayuda de la ecuación 9 o con la ecuación 13.

$$\text{Error relativo} = \frac{DM}{\bar{x}} \quad (13)$$

Ejemplo:

Se mide cinco veces el voltaje en un tomacorriente con un voltímetro graduado en voltios. Los resultados obtenidos son: 120.1 V; 120.4 V; 119.9 V; 120.0 V; 119.8 V.

Solución:

Con la ecuación 10 se determina el valor promedio de las mediciones:

$$\bar{x} = \frac{120.1 + 120.4 + 119.9 + 120.0 + 119.8}{5} = 120.04 \text{ V} \quad (14)$$

72

Se calcula la desviación media con la ecuación 11

$$DM = \frac{|120.1 - 120.04| + |120.4 - 120.04| + |119.9 - 120.04| + |120.0 - 120.04| + |119.8 - 120.04|}{5} \quad (15)$$

$$DM = 0.168 \quad (16)$$

El voltaje finalmente es:

$$V = 120.04 \pm 0.168 \text{ V} \quad (17)$$

Por último, el error relativo es

$$\text{Error relativo} = \frac{DM}{\bar{x}} = 0.0014 = 0.14\% \quad (18)$$

El SI fue creado por la Conferencia General sobre Pesos y Medidas y adoptado por casi todos los países industriales del mundo a partir de 1960.

Sistema internacional de unidades

Los científicos y los ingenieros requieren de mediciones exactas y confiables, las cuales solo es posible lograr a través de unidades inmutables, que permitan ser duplicadas en cualquier parte.

El sistema de unidades empleado de forma estandarizada a nivel mundial es el *Sistema Internacional (SI)*.

Las unidades básicas del SI aparecen en la tabla 1.

Tabla 1. Unidades básicas del SI.

CANTIDAD	NOMBRE UNIDAD	SÍMBOLO
Longitud	Metro	m
Masa	Kilogramo	Kg
Tiempo	Segundo	s
Corriente eléctrica	Ampere	A
Temperatura	Kelvin	K
Cantidad de sustancia	Mol	Mol
Intensidad luminosa	Candela	Cd

Fuente: elaboración propia.

Si bien en la tabla 1 se encuentran las unidades básicas del SI, a continuación, se examinarán en detalle las tres unidades fundamentales: longitud, masa y tiempo.

Longitud

73

La longitud es un concepto de medida para definir la *distancia* de objetos geométricos. En el Sistema Internacional, la unidad de longitud es el metro, cuyo símbolo se observa en la tabla 1. Durante muchos años, la definición del metro no era precisa, ya que se consideraba como la distancia existente entre dos marcas de una barra de platino (metro de referencia o metro patrón). El problema de precisión radicaba en que cualquier material presenta dilataciones o contracciones por efecto de la temperatura. Por esta razón, la longitud se define como *la distancia que recorre la luz en el vacío en un valor de tiempo (este valor de tiempo se definió en 1982)*.

Un año luz es la distancia que recorre la luz en un año. Es una unidad fundamental en estudios relacionados con el espacio y el Universo.

La unidad básica es el metro (m) y en ocasiones se utilizan múltiplos (como el kilómetro) o submúltiplos (centímetro), según se requiera. Una unidad particular que se utiliza para la longitud es el *año luz*.

La unidad fundamental del SI para la masa es el kilogramo y no el gramo. Sin embargo, los prefijos sí se utilizan en función del gramo.

Masa

La masa es una magnitud que representa la cantidad de materia de un cuerpo. La unidad que utiliza el SI para la medición de la masa es el *kilogramo* (kg). También se utilizan múltiplos o submúltiplos del gramo

según se requiera. Por ejemplo, los miligramos son un submúltiplo utilizado para definir la cantidad de medicamentos que requiere alguna persona.

Tiempo

El tiempo es la magnitud física que permite medir la duración de acontecimientos bajo estudio. La unidad en el SI del tiempo es el segundo. Se pueden utilizar múltiplos o submúltiplos de este según se requiera, como el milisegundo (muy utilizado en la ingeniería eléctrica), la hora o los minutos (estos se definen en función de segundos).

Sensores

Un sensor es un dispositivo que varía una de sus propiedades internas ante la presencia de magnitudes físicas. En otras palabras, indica la presencia y la magnitud de dicha variable a la cual es sensible. Por lo general, el sensor transforma esa propiedad interna en una variable eléctrica. De esta forma, la equivalencia entre el parámetro eléctrico y la propiedad modificable del sensor muestra el valor de la magnitud física de interés.

Arduino tiene una ventaja y es el hecho de contar con una gran variedad de sensores adaptados para funcionar con la placa electrónica. Por ejemplo, en la figura 66 se tiene un sensor de humedad y temperatura

74



Figura 66. Sensor de humedad y temperatura DHT11 para Arduino.

Fuente: elaboración propia.

El sensor de la figura 66 modifica unas propiedades internas ante la presencia de humedad o temperatura, y las convierte en parámetros eléctricos. A través de la placa electrónica y del código, se interpreta la relación entre dichos parámetros eléctricos y la humedad/temperatura del entorno. De esta forma, un valor de corriente o voltaje determinado será proporcional a un valor de temperatura y de humedad.

Se considera pertinente que para cada sensor se conozca el error que, introduce en las medidas, así como su rango de medición.

Por ejemplo, un sensor como el de la figura 66 tiene las siguientes características que se listan en la tabla 2.

Tabla 2. Características del sensor DHT11

	ALIMENTACIÓN	ENTR 3.5 Y 5 V
	Señal de salida	Digital
Temperatura	Rango de medida	De 0 °C a 50 °C
	Precisión	De 2 °C a 25 °C
Humedad	Rango de medida	De 20 % RH a 90 RH
	Precisión	De 5 % RH entre 0 °C y 50 °C

Fuente: elaboración propia.

Esta información surge de las hojas de especificaciones (datasheet en inglés) que tienen los sensores y se encuentran fácilmente por internet.

La tabla 2 tiene utilidad al momento de desarrollar proyectos pedagógicos o investigativos, ya que dicha información permite identificar los rangos en los cuales el sensor puede ser utilizado y el error que introduce a las mediciones correspondientes.

Uno de los objetivos fundamentales de las ciencias naturales es lograr la explicación del entorno físico, químico y biológico mediante las competencias de sus respectivas disciplinas. Es evidente que estas explicaciones son mucho más claras a través de las prácticas de laboratorio. Sin embargo, no es un secreto que en la mayoría de las instituciones hay falencias para el desarrollo de una gran variedad de prácticas por la falta de instrumentación. Aunque usted no lo crea, diferentes estudios han evidenciado que la infraestructura de los laboratorios de ciencias naturales, repercute en la calidad educativa (véase, por ejemplo, Redacción Educación, 2016). Por ello, este texto es un esfuerzo para que las falencias de infraestructura que la gran mayoría de docentes en ciencias experimentales tienen en las instituciones donde laboran, puedan ser en cierta medida subsanadas, para enriquecer su práctica profesional. Por ejemplo, una gran cantidad de sensores con utilidad en clases de física, química y biología pueden ser implementados mediante Arduino.

Es muy importante como docentes reflexionar sobre la propia práctica para encontrar estrategias de mejora. Arduino es una herramienta útil que usted puede implementar en su práctica para mejorar los ejercicios en el laboratorio.

El uso de Arduino y sus sensores en las prácticas de laboratorio permite superar deficiencias de infraestructura que le impiden fortalecer el contenido de su respectiva cátedra (física, química, biología)

Aplicación de sensores y Arduino en mi contexto

“El camino del progreso no es ni rápido ni fácil”
Marie Curie

Objetivos del capítulo y resultados de aprendizaje

Objetivos

- ▶ Conocer algunos sensores utilizados con Arduino para la medición de variables físicas en la práctica docente.
- ▶ Estimar la calibración de un sensor con base en la información obtenida en su hoja de especificaciones.

77

Resultados de aprendizaje al finalizar el capítulo

Gestiona un mismo instrumento para la medición de diferentes variables físicas.

Modela la variable medida por un instrumento y su equivalente eléctrico para la calibración correspondiente.

Sensores con Arduino

Hasta el momento se ha trabajado en un primer montaje con Arduino (el semáforo), y se explicó su elaboración y su funcionamiento. También se ha explicado la importancia de los sensores y el manejo de errores en clases de ciencias experimentales. A continuación, se integrará lo visto anteriormente, por lo cual se implementarán dos sensores con Arduino que son útiles en clase de ciencias experimentales (particularmente de física). El objetivo de este capítulo es que usted entienda la utilidad de Arduino en su práctica como docente de ciencias naturales o experimentales y, además, que

Una asignatura como geografía puede tener un alto contenido experimental (al estudiar las variables meteorológicas, las cuales se pueden conocer mediante sensores y Arduino).

es posible encontrar fácilmente muchísima información útil para montar el sensor que usted necesita en su práctica profesional. Por último, tenga presente que este texto busca que usted enriquezca con criterio la información que puede encontrar en diversas fuentes (internet, libros, etc.) para su beneficio profesional.

Sensor de distancia

Aquellos que trabajan como docentes de física, entienden la complejidad de realizar experimentos que permitan realizar mediciones coherentes de distancia, velocidad y aceleración, en el estudio del movimiento de los cuerpos. Si es docente de media vocacional, seguramente la mayoría de los experimentos donde sus estudiantes han requerido conocer la posición de un cuerpo, los ha realizado con marcas en la superficie donde se moverán. A continuación, se presenta un sensor de distancia, cuyo funcionamiento se fundamenta en el comportamiento de las ondas sonoras, a través de un sensor de ultrasonido. *Esta es una de las principales facilidades que se tiene con Arduino: sensores pensados para ciertas aplicaciones se pueden ajustar para otras.*

Para realizar este montaje, usted necesitará comprar el sensor de sonido HC-SR04, el cual es muy sencillo de conseguir y cuyo costo es de aproximadamente en 5000 COP (precio de 2019). En la figura 67 se aprecia el sensor que se requiere.

78



Figura 67. Sensor de ultrasonido HC-SR04

Fuente: elaboración propia.

La ventaja que presenta este sensor, junto con una gran cantidad de sensores disponibles en el mercado, es que tiene los pines listos para su conexión con la placa Arduino. En este caso, la conexión de los pines en la figura 67, de izquierda a derecha son: VCC, Trigger, Echo, GND.

Tenga muy presente que, para cada sensor, usted deberá cerrar el circuito, por lo cual debe energizarlo con +VCC (por lo general 5 V) y utilizar el pin GND (análogo al negativo de la fuente de voltaje).

El funcionamiento del sensor se fundamenta en el principio de las ondas de ultrasonido: envía un disparo ultrasónico, el cual es detectado por el mismo sensor. De aquí, es posible determinar el tiempo que tarda la onda en ser emitida y recibida. Con

dicha información, se calcula la distancia que recorrió la onda en ir o volver, ya que la velocidad del sonido es constante y equivale a 343m/s (en el aire y a 20° C). De esta forma, el montaje que se debe realizar se aprecia en la figura 68.

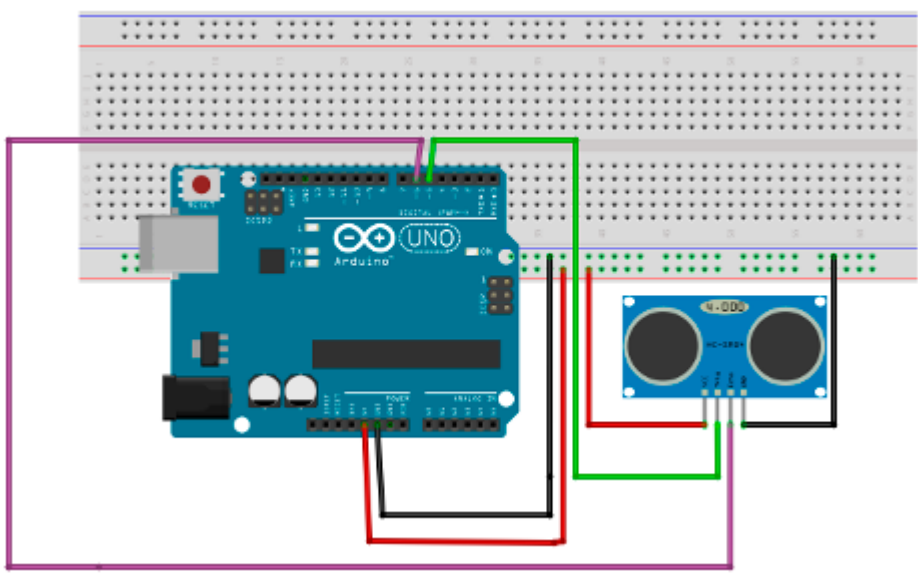


Figura 68. Diagrama de conexiones del sensor de ultrasonido.

Fuente: elaboración propia.

Para las conexiones tenga en cuenta las siguientes consideraciones:

- Uno de los nodos inferiores se polarizó con +5 V de Arduino y será el correspondiente vcc. El otro nodo inferior, se polarizó con GND. De esta manera, se puede sacar provecho de estos nodos, para conectar los pines correspondientes del sensor en “Trigger” al pin digital 5 de la placa Arduino y el pin llamado “Echo” al pin digital 6.

El código para el funcionamiento del sensor se ilustra en la figura 69.

Sensor_de_Distancia\$

```
1 const int trig=5;//se define la variable trigger en el pin 6
2 const int echo=6;//se define la variable echo en el pin 12

3 void setup() {
4 Serial.begin(9600);//se inicia la comunicación con el puerto serial
5 pinMode(trig,OUTPUT);//se define el pin5 como salida
6 pinMode(echo, INPUT);//se define el pin6 como entrada

7 }

8 void loop() {
9 long tiempo;//se define la variable tiempo (tipo long)
10 long distancia;//se define la variable distancia (tipo long)
11 digitalWrite(trig, LOW);//el trigger debe estar inicialmente apagado
12 delayMicroseconds(2);//se esperan 2 microsegundos para garantizar que el trigger esté apagado
13 digitalWrite(trig,HIGH);//Se activa un pulso de salida
14 delayMicroseconds(10);//se mantiene el pulso activado durante 10us
15 digitalWrite(trig,LOW);//se apaga el pulso
16 digitalWrite(echo, HIGH);//se activa el pin que recibe el eco de la onda sonora
17 tiempo=pulseIn(echo,HIGH);//se determina el tiempo de duración del eco
18 distancia=(tiempo/2/29.1);//se determina la distancia
19 Serial.println(String(distancia)+"cm");//se imprime la distancia en pantalla
20 delay(500);//se esperan 0,5segundos para volver a empezar
21}
```

Figura 69. Código para el funcionamiento del sensor de sonido

Fuente: elaboración propia.

80

Si bien en la figura 69 se explica el código desarrollado a través de los comentarios, a continuación, se realizará una explicación mucho más detallada línea por línea (note la numeración en negrita que se colocó a la izquierda, esto se añadió para facilitar la explicación)

Línea uno `const int trig=5;` y **línea dos** `const int echo=6;`

Antes del `void setup` se definieron dos variables con el comando `const int` (todo en minúscula y con espacio entre las dos palabras). Esto renombró al pin 5 de Arduino como `trig` y al pin 6 de Arduino como `echo`. La razón de hacer esto es para evitar confusiones en el código más adelante.

Recuerde que, dentro de los paréntesis después de la línea de `void setup` no se incluye nada. Tampoco olvide que la única línea que no finaliza con ; es la de `void loop` y `void setup`.

Línea tres `void setup() {` y **línea cuatro** `Serial.begin(9600);`

Se inicia el `void setup` colocando el código entre los corchetes. El comando `serial.begin` de la línea cuatro indica que se iniciará una comunicación con el puerto serial (puerto USB) donde se conecta el Arduino. El valor 9600 es la velocidad en baudios (número de unidades de señal por segundo) a la cual se realizará la comunicación de información.

Línea cinco `pinMode (trig, OUTPUT);` **línea seis** `pinMode (echo, INPUT);` **y línea siete** `}`

En la línea cinco se define la variable *trig* como salida y en la línea seis, la variable *echo* como entrada. Esto se hace de la misma manera como se definieron las variables de salida en el semáforo. Por último, en la línea siete se cierra la sección de *void setup*, con el corchete que se abrió en la línea tres.

Línea ocho `void loop() {` **línea nueve** `long tiempo;` **y línea diez** `long distancia;`

Se inicia la sección de *void loop* en la línea ocho, guardando los mismos cuidados de la sección *void setup* (colocar el código entre los corchetes, no colocar nada entre los paréntesis que finalizan la línea *void loop* y evitar colocar ; al finalizar esa línea. Por su parte, en la línea nueve y diez, se definen dos nuevas variables (tiempo y distancia), las cuales serán de tipo long (un tipo de datos de Arduino que trabaja números enteros).

Línea once `digitalWrite (trig, LOW);`

En esta línea, se imprime un primer estado de apagado en el pin *trigger* del sensor, el cual se llamó *trig* en el código. Esto se realiza de la misma forma como se daba un estado de High o de Low en el semáforo. Esto se hace, para verificar que se inicie con el *trigger* apagado.

Línea doce `delayMicroseconds (2);`

Después de apagar el trigger, se esperan dos microsegundos. Para ello, se utiliza la función *delayMicroseconds*, el cual funciona de la misma forma que la función *delay* utilizada en el código del semáforo, con la única diferencia que ahora. La espera no se realizará en milisegundos, sino en microsegundos. Esta espera se hace para garantizar la verificación de la línea once.

El *delay* se realiza en microsegundos, ya que la lectura que hace Arduino en la línea diecisiete la hace en microsegundos.

Línea trece `digitalWrite (trig, HIGH);`

Se cambia el estado del *trigger*, colocándolo en High. Aquí, el sensor enviará un disparo a través de una onda ultrasónica.

Línea catorce `delayMicroseconds (10);`

Se esperan diez microsegundos, el cual es el tiempo recomendado por el fabricante para que el disparo funcione correctamente. Esta información se consigue en la hoja de especificaciones.

Recuerde que las hojas de especificaciones son de gran utilidad y se encuentran fácilmente por internet.

Línea quince `digitalWrite(trig,LOW);`

Se apaga el disparo del *trigger*.

Línea dieciséis `digitalWrite(echo, HIGH);`

Se enciende el pin *echo* del sensor para lograr captar la señal del *trigger* de regreso. Recuerde que el principio de funcionamiento es que se envía una onda ultrasónica, la cual debe chocar con un obstáculo ubicado a una distancia determinada para reflejarse. Con el pin *echo*, se captará la onda reflejada.

Línea diecisiete `tiempo=pulseIn(echo,HIGH);`

En esta línea se mide el tiempo que tardó el sensor en captar la onda ultrasónica reflejada. Para ello, se utiliza la función *pulseIn* (se escribe todo en minúscula, excepto la letra I). Esta función leerá el tiempo que tardará el pin *echo* (equivalente al pin 6 de la placa electrónica) en pasar de encendido a apagado. Tenga en cuenta que, después de encender el pin *echo*, este se apagará al recibir la onda ultrasónica reflejada. La función *pulseIn* realiza la lectura de tiempo en *microsegundos*, razón por la cual el *delay* también se realizó en microsegundos.

Línea dieciocho `distancia=((tiempo/2/29.1));`

82

Para medir la distancia, se utilizó un poco de física. Al ser una onda sonora, se tendrá una velocidad constante, por lo cual la expresión de distancia se expresa como el producto de velocidad (en m/s) por tiempo (en s) como se tiene en la ecuación 19. La velocidad del sonido es constante y equivale a 343 m/s en el aire (este valor depende de la temperatura).

$$x = vt \quad (19)$$

Por lo tanto, para recorrer un metro, el sonido demorará

$$t = \frac{1m}{343m/s} = 0.0029s \quad (20)$$

Sin embargo, el pulso se está midiendo en μs , por lo cual se requiere un ajuste que permita leer más fácil los resultados. Para ello, se sabe que el sonido recorre en 0.0029 s un metro (ecuación 20), pero convertido a $\mu s/cm$ queda

$$0.0029 \frac{s}{m} * \frac{1000000\mu s}{s} * \frac{1 m}{100cm} = 29\mu s/cm \quad (21)$$

Note que los $29\mu s/cm$ es el inverso de la velocidad, por lo cual la ecuación 19 quedaría como

$$x = \frac{t}{29\mu s/cm} \quad (22)$$

¡El resultado de la ecuación 22 debe dividirse entre dos! Recuerde que el tiempo de interés es la mitad del medido, porque solo interesa el tiempo que tardó la onda en ir hasta el obstáculo donde se quiere medir la distancia y la medición del *echo* en el sensor es el doble.

Línea diecinueve `Serial.println(String(distancia)+"cm");`

Con esta línea de código, se imprime el resultado calculado de distancia en el monitor serial de Arduino. El *monitor serial* es una herramienta con la cual cuenta Arduino, que permite imprimir en la pantalla del computador los resultados obtenidos.

Para aprovechar las ventajas del monitor serial, debe utilizar la función `Serial.println()`. Dentro del paréntesis, se debe colocar la variable que desea visualizar en el monitor serial, que en nuestro caso es la distancia. Sin embargo, como puede observar, primero se coloca la palabra *String* en mayúscula y después sí se coloca la variable de interés dentro de paréntesis. Esto se hace con el fin de enviar los datos de distancia como texto al puerto USB y visualizarlos en pantalla. Por último, en esta línea se finaliza con un + y *cm*. Mediante esto, se indica que el texto de distancia enviado debe ir con el rótulo de *cm*. Ya con esto, solo falta abrir el monitor serial.

83

Es muy sencillo visualizar los resultados en pantalla a través del monitor serial: diríjase a la barra de herramientas y en la opción *Herramientas* seleccione la opción *monitor serie*, como se ilustra en la figura 66.

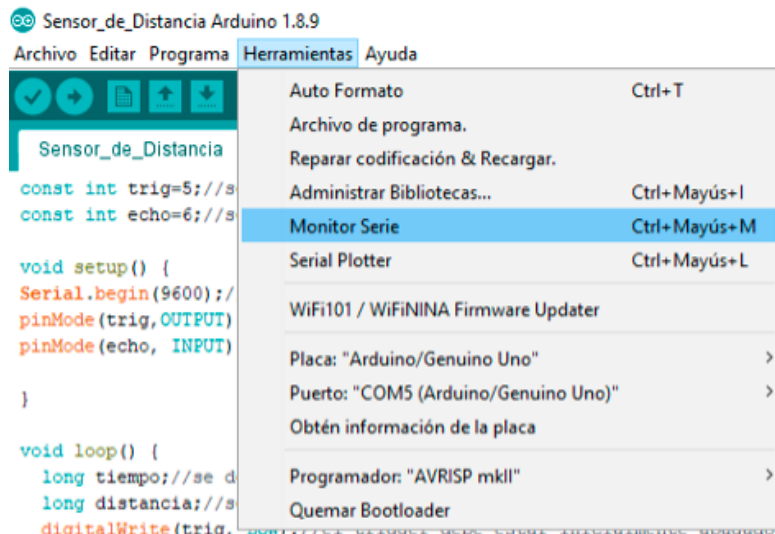


Figura 70. Selección del Monitor Serie

Fuente: elaboración propia.

Después de compilar y subir el código a la placa, habilite el monitor serie como se observa en la figura 70 y aparecerá una pantalla como la que se ve en la figura 71.

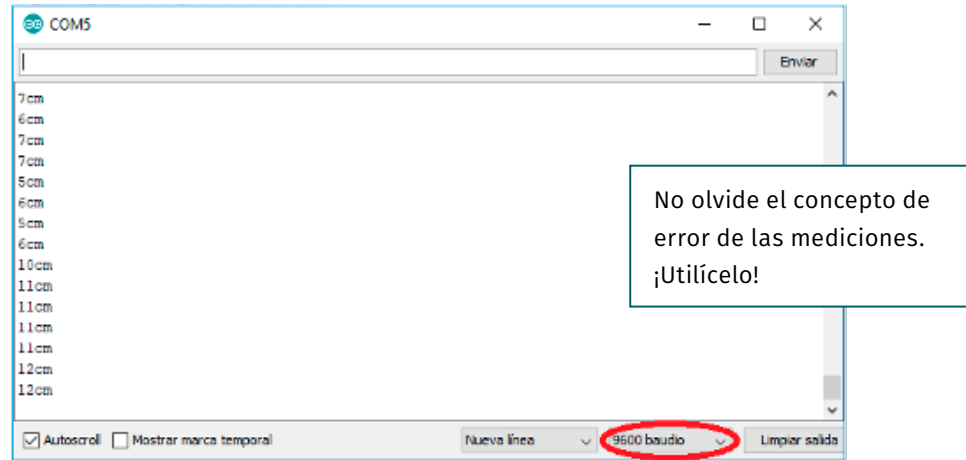


Figura 71. Monitor serie mostrando valores de distancia

Fuente: elaboración propia.

Algo muy importante que vale la pena aclarar es que la velocidad de 9600 baudios definida en el código también debe configurarse en el monitor serial, para ello hay que seleccionarla como aparece en el círculo rojo de la figura 71. Como se logra apreciar, el monitor serial de Arduino permite visualizar los valores de distancia en cm.

84

Línea veinte `delay(500);` **y línea veintiuno** `}`

La línea veinte de código define una espera de 0.5 segundos para realizar las mediciones de distancia y la última cierra el corchete abierto en la línea ocho del `void loop`.

Sensor de ruido

El siguiente sensor es uno de ruido, el cual mostrará la forma mediante la cual es posible leer datos análogos de un sensor. El sensor que se requiere para este montaje es el KY-038 (figura 72).

A0 será la salida análoga del sensor, G la tierra, vcc los +5 V y D0 la salida digital.

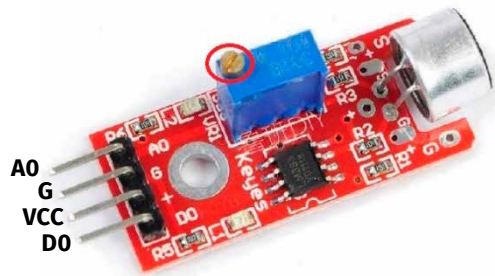
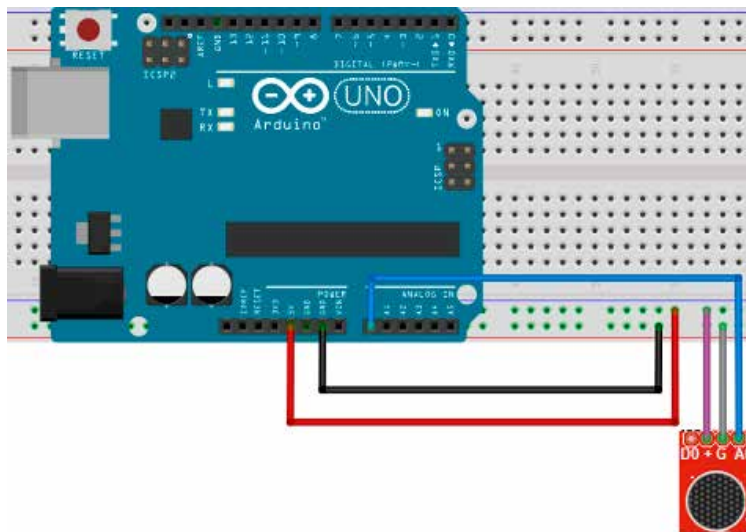


Figura 72. Sensor KY-038

Fuente: elaboración propia.

El sensor KY-038 es un sensor de sonido, el cual tiene salida digital y análoga. A continuación, se presentará un montaje y un código para la salida digital. En diversas fuentes, se encuentra una gran cantidad de tutoriales para implementar la salida digital. Este sensor es relativamente económico (6000 COP). Para este sensor es posible ajustar la sensibilidad, moviendo con un destornillador el graduador encerrado en el círculo rojo de la figura 72.

El montaje a realizar se ilustra en la figura 73. En rojo, se tiene los +5 V del Arduino hasta uno de los nodos de la protoboard, donde se conectará el terminal + del sensor. El jumper negro será la GND y se conectará en otro nodo de la protoboard, donde irá el terminal G del sensor. Por último, el terminal Ao del sensor se conecta al pin Ao del Arduino. El terminal digital del sensor se deja sin conexión.



85

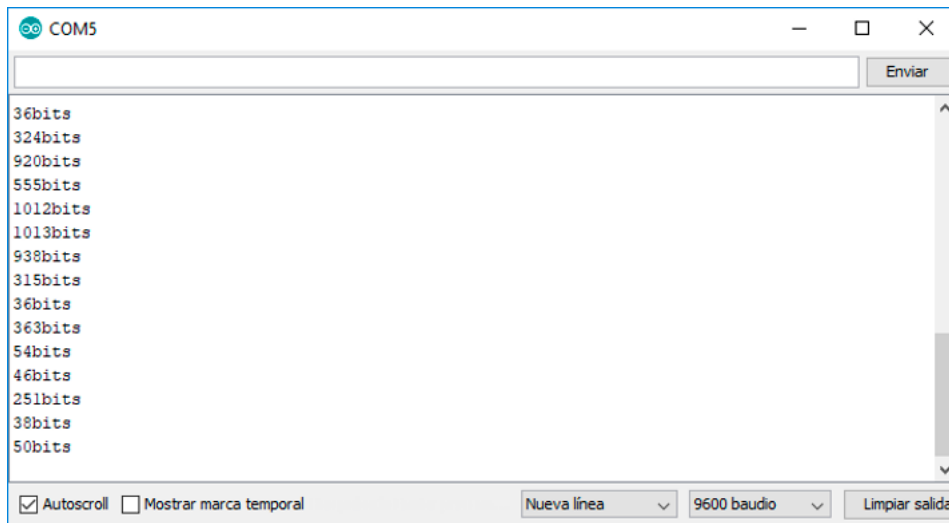
Figura 73. Diagrama de conexiones del experimento dos
Fuente: elaboración propia.

El código utilizado es sencillo y se observa en la figura 74.

```
Sensor_de_Sonido $  
  
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  int ruido=analogRead(A0);  
  Serial.println(String(ruido)+"bit");  
  delay (200);  
}
```

Figura 74. Código para sensor de sonido
Fuente: elaboración propia.

Como es claro en la figura 74, el código para la lectura del sensor es bastante sencillo y lo único que se requiere es: iniciar la comunicación con puerto serial en el *void setup*, definir la variable de tipo entero *ruido* con la función `analogRead(A0)`; Esta función permite leer los valores análogos del pin seleccionado (cuyo nombre se coloca dentro de paréntesis). Si se hubiese utilizado otro de los seis pines analógicos de la placa, se debía colocar el nombre correspondiente en el código. Esta lectura está entre 0 y 1023 bits. Este valor de lectura se puede ajustar con la sensibilidad y aparece en el monitor serie como se ilustra en la figura 75.



86

Figura 75. Lectura análoga del sensor KY-038 en el monitor serie de Arduino.

Fuente: elaboración propia.

Este sensor puede ser utilizado para detectar sonido o como un indicador de superación de niveles de ruido.

Lo ideal hubiese sido que la lectura en pantalla mostrase el nivel de ruido en decibeles (dB), sin embargo, para ello se requiere una conversión que se explicará en el siguiente capítulo y para la cual se requieren unos datos específicos que contiene la hoja de especificaciones (*datasheet*) de la mayoría de los sensores, pero que no están disponibles en el caso del KY-038.

Sensores no adaptados para Arduino

“La ciencia no es sino una perversión de sí misma, a menos que tenga como objetivo final el mejoramiento de la humanidad”
Nikola Tesla

Objetivos del capítulo y resultados de aprendizaje

Objetivos

- ▶ Desarrollar experimentos con Arduino y algunos sensores adicionales para su utilización en el laboratorio.
- ▶ Explicar la construcción de gráficas que permiten la visualización de parámetros obtenidos mediante sensores y Arduino.
- ▶ Definir mediante un ejemplo el proceso de caracterización de una curva de calibración para sensores.

87

Resultados de aprendizaje al finalizar el capítulo

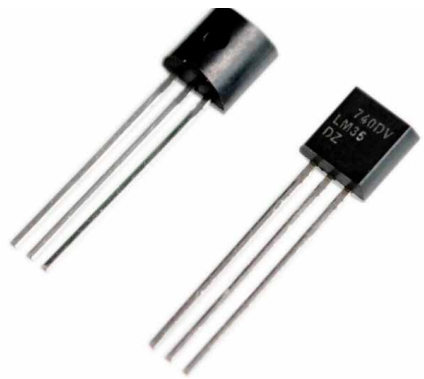
- ▶ Incorpora análisis gráfico al integrar sensores con Arduino en su práctica profesional.
- ▶ Interpreta gráficos desarrollados con Arduino para la visualización de parámetros obtenidos mediante sensores.
- ▶ Entiende la curva de calibración de un instrumento para la interpretación de resultados de medida en su qué hacer profesional.

Otros sensores

En este capítulo se abordará otro tipo de sensor que por lo general no tiene los pines marcados para trabajar con Arduino. La labor adicional será identificar estos pines a través de la revisión de la hoja de especificaciones del fabricante. Además de ello, también se profundizará en procedimientos que permitan unas lecturas mucho más comprensibles de los sensores de acuerdo con las equivalencias que define cada fabricante. Para esto, se trabajará con el sensor de temperatura LM35 y el sensor de gas MQ-135.

Sensor de temperatura

Para realizar este montaje, usted necesitara comprar el sensor de temperatura LM35, el cual es muy sencillo de conseguir y cuyo costo está aproximadamente de 4000 COP (precio de 2019). En la figura 76 se aprecia el sensor que se requiere.



88

Figura 76. Sensor de temperatura LM35

Fuente: Hojas de datos de acceso libre.

Como se evidencia en la figura 76, a diferencia de los sensores utilizados en el capítulo anterior, no se tiene una identificación de los terminales, por lo cual se debe tener especial cuidado al momento de conectarlo. Para ello, es necesario acudir a la hoja de especificaciones del fabricante (basta con buscar en Google “Datasheet LM35”). Allí, se indica la forma de identificar los tres terminales del sensor, lo cual se muestra en la figura 77.

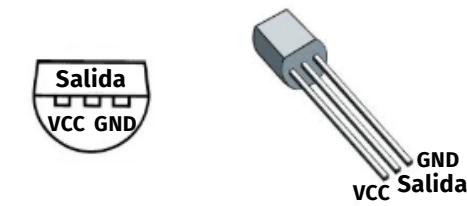


Figura 77. Terminales del sensor LM35

Fuente: hojas de datos de acceso libre.

De esta forma, lo único que tendrá que hacer es observar el sensor por la cara plana y:

1. Conectar los 5 V del Arduino en la terminal izquierda del sensor.
2. Conectar el pin Ao del Arduino a la terminal del medio del sensor.
3. Unir el GND del Arduino con la terminal derecha del sensor.

También puede identificar los terminales de este sensor mirándolo desde el extremo de los terminales y con la cara plana hacia arriba.

Estas conexiones se muestran en la figura 78.

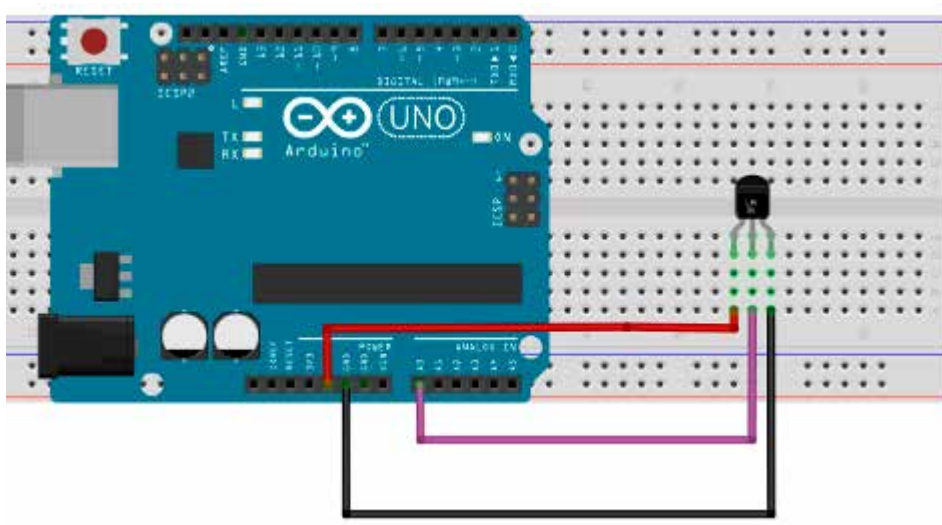


Figura 78. Conexiones del sensor LM35 con Arduino

Fuente: elaboración propia.

En la figura 78 se observa que el jumper rojo une los +5 V del Arduino con el terminal izquierdo del sensor. El terminal del medio se une mediante el jumper morado con el pin A0 del Arduino y, el jumper negro une la GND de la placa con el terminal derecho del sensor.

Recuerde identificar los terminales del sensor mirándolo por el lado de la cara plana.

El código propuesto se muestra en la figura 79.

```

Sensor_de_temperatura Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
Sensor_de_temperatura $
1 int sensor=A0;//se define el pin A0 como sensor
2 float lectura;//se define la variable lectura
3 float temperatura;//se define la variable temperatura
4 float temperaturaK;//se define la variable T en kelvin
5 void setup() { //se inicia el void setup
6 Serial.begin(9600);//se inicia la comunicación serial
7 }//se cierra la sección void setup
8 void loop() { //se inicia el voidloop
9 lectura=analogRead(sensor);//se realiza la lectura del pin A0
10 temperatura=(lectura*5*100)/1024;//se determina la temperatura
11 temperaturaK=temperatura+273;//se determina en Kelvin
12 Serial.println("Temperatura °C y K");//Se imprime en pantalla
13 Serial.println(String (temperatura)+"°C");//se imprime la T
14 Serial.println(String(temperaturaK)+"K");//se imprime la T en K
15 delay (5000);//se esperan cinco segundos por medición
16 }//se cierra la sección void loop

```

Figura 79. Código para la lectura del sensor de temperatura

Fuente: elaboración propia

90

Línea uno `int sensor=A0;`**Línea dos** `float lectura;`**Línea tres** `float temperatura;`
Línea cuatro `float temperaturaK;`

La palabra float indica que el tipo de dato de las variables lectura, temperatura y temperaturaK serán decimales.

En la línea uno se define la variable sensor y esta será tomada del pin A0 de la placa electrónica. De la línea dos a la línea cuatro, se definen las variables que se utilizarán para los cálculos en el código.

Línea cinco `void setup() {` **Línea seis** `Serial.begin(9600);` **y línea siete** `}`

En la línea cinco se inicia el *void setup* y se cierra en la línea siete. Recuerde que no es necesario escribir el punto y coma en estas líneas (donde inicia y finaliza el *void setup*). En la línea seis, se inicia la comunicación con el puerto serial a una velocidad de 9600 baudios.

Línea ocho `void loop() {`

Al estar conectado el A0 del Arduino con la salida del sensor, se está censando la salida de este último.

Se inicia la sección *void loop*.

Línea nueve `lectura=analogRead(sensor);`

En la línea nueve se realiza la lectura del pin A0 del Arduino, el cual se nombró como A0 en el código.

Línea diez `temperatura=(lectura*5*100)/1024;`

La lectura análoga que realiza Arduino toma valores entre 0 y 1023 bits, teniendo un total de 1024 valores posibles. Por ello, debe convertirse ese valor a un valor de temperatura que sea entendible para cualquier persona que lo visualice en pantalla.

Para ello, tenga en cuenta la información de la tabla 3, la cual se tomó del *datasheet* del fabricante.

Tabla 3. Consideraciones dadas por el fabricante.

Sensor LM35
Rango de operación
Entr -55 °C y 150 °C
Equivalencia
1 °C equivale a 10 mV
Entrada
Desde 4 V hasta 30 V
Error
0.02

Fuente: elaboración propia.

El error de la tabla es garantizado por el fabricante para temperaturas entre 5 °C y 25 °C

91

Para realizar la conversión a valores comprensibles, lo primero que se debe realizar es calcular el equivalente de los bits censados por el Arduino, pero en voltaje. Para esto, se realiza una regla de tres simple donde se sabe que, 1024 bits (el máximo de bits) equivaldrá a los cinco voltios (5 V)

$$\text{Voltaje} = \frac{\text{lectura} * 5V}{1024} \quad (23)$$

Después, se calcula el equivalente del voltaje en temperatura de acuerdo con la ecuación 24 y la tabla 3.

$$\text{Temperatura} = \frac{\text{lectura} * 5V}{1024} * \frac{1}{10mV} \quad (24)$$

Lo cual equivale a

$$\text{Temperatura} = \frac{\text{lectura} * 5V * 100}{1024} \quad (25)$$

La ecuación 25 es la que se tiene en la línea diez del código.

Línea once `temperaturaK=temperatura-273;`

Con el cálculo de temperatura en grados Celsius, se realiza el cálculo de temperatura en grados Kelvin.

Recuerde que la función `Serial.println()` solo tiene en mayúscula la letra S. Recuerde que la palabra `String` convierte las variables numéricas en texto para imprimir en pantalla.

Línea doce `Serial.println("Temperatura °C y K");` **Línea trece** `Serial.println(String (temperatura)+"°C");` **y Línea catorce** `Serial.println(String(temperaturaK)+"K");`

En estas líneas se imprime en el monitor serial las lecturas de los cálculos realizados. En la línea doce se imprime un título que precederá cada medición: temperatura °C y K. En las dos líneas siguientes, se muestra la temperatura en cada unidad correspondiente.

Línea quince

En esta línea se realiza una espera mediante la función `delay` para que las mediciones se realicen cada cinco segundos.

`delay (5000);`

Línea dieciséis }

92

Por último, se cierra con corchete el `void loop` que se abrió en la línea ocho.

No olvide verificar y subir el código. Sin este paso, el sensor no funcionará.

Después de verificar y subir el código, se abre el monitor serial de Arduino para visualizar las mediciones de temperatura del sensor, como se ilustra en la figura 80.

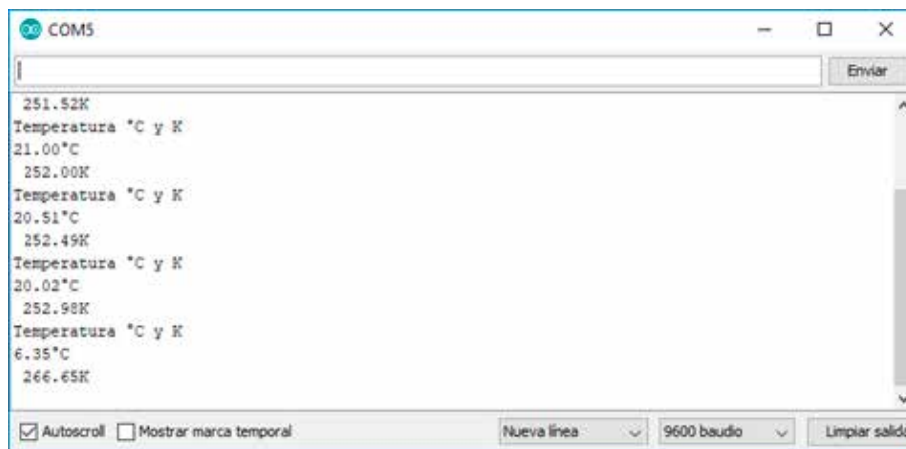


Figura 80. Valores de temperatura mostrados en el monitor serial.

Fuente: elaboración propia.

Serial Plotter

Es evidente que la construcción de gráficas es una parte fundamental del desarrollo de prácticas de laboratorio. Con Arduino, esto también es posible a través de la herramienta Serial Plotter. Sin embargo, para utilizarla, debe modificar unas líneas del código utilizado en la figura 79, específicamente las líneas 12, 13 y 14. Inicialmente, se va a graficar únicamente la temperatura en grados centígrados. Para ello, elimine la línea doce y catorce del código mostrado en la figura 79 y modifique la línea trece como se explica a continuación.

El ejemplo se realizó para el sensor de temperatura, pero es posible realizarlo para cualquier sensor.

Línea trece: esta línea originalmente era

`Serial.println(String (temperatura)+"°C");` pero recuerde que esto se hizo para lograr visualizar en pantalla como texto la temperatura calculada, lo cual es posible mediante la palabra *String*. Sin embargo, para graficar en el Serial Plotter, la temperatura debe volver a ser un valor numérico. Por ello, se debe evitar convertirlo en tipo texto con la palabra *String*, por lo que ahora esta línea debe quedar de la siguiente manera:

```
Serial.println((temperatura));
```

De esta forma, el código queda como se ilustra en la figura 81, donde también se observa que el *delay* se redujo a 0.2 s, para que la construcción de la gráfica fuese más rápida.

93

```
Sensor_de_temperatura $
1 int sensor=A0;//se define el pin A0 como sensor
2 float lectura;//se define la variable lectura
3 float temperatura;//se define la variable temperatura
4 float temperaturaK;//se define la variable T en kelvin
5 void setup() { //se inicia el void setup
6 Serial.begin(9600);//se inicia la comunicación serial
7 }//se cierra la sección void setup
8 void loop() { //se inicia el voidloop
9 lectura=analogRead(sensor);//se realiza la lectura del pin A0
10 temperatura=(lectura*5*100)/1024;//se determina la temperatura
11 temperaturaK=temperatura+273;//se determina en Kelvin
12 Serial.println((temperatura));//se imprime la T
13 delay (200);//se esperan cinco segundos por medición
14 }//se cierra la sección void loop
```

Figura 81. Código resultante para visualizar la gráfica en el Serial Plotter

Fuente: elaboración propia.

Para acceder al Serial Plotter, acceda a herramientas en el menú de Arduino y seleccione la opción Serial Plotter como se tiene en la figura 82.

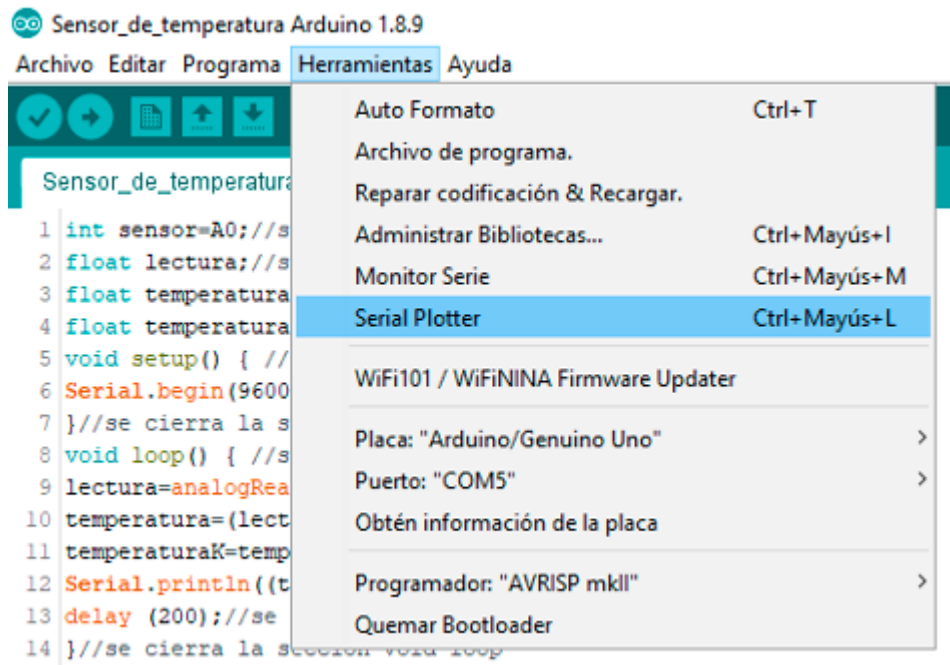


Figura 82. Seleccionando el Serial Plotter de Arduino.

Fuente: elaboración propia.

94

Al seleccionar la opción, se abrirá una nueva ventana donde se irá construyendo la gráfica de temperatura. No olvide definir la misma velocidad de comunicación de 9600 en el círculo rojo de la figura 83.



Figura 83. Serial Plotter ejecutándose.

Fuente: elaboración propia.

En la figura 83 se está graficando la temperatura censada por el LM35 en el tiempo. El eje horizontal está en segundos.

Sin embargo, usted también puede graficar dos o más variables simultáneas en el Serial Plotter. Para ello, vuelva al código original expuesto en la figura 79. Al igual que en la primera gráfica, debe realizar modificaciones a las líneas de código 12, 13 y 14.

Línea doce: elimínela.

Línea trece `Serial.println(String (temperatura)+"°C");` modifíquela para que quede de la forma `Serial.print (temperatura);`

Línea catorce `Serial.println(String(temperaturaK)+"K");` modifíquela para que quede de la forma `Serial.println (temperaturaK);`

Por último, añada una nueva línea entre las dos modificadas anteriormente, la cual debe ser `Serial.print(",");`.

Se quitó la conversión a texto y, además, en la primera variable se omitieron las letras ln del Serial.println al igual que en la línea de código añadida

De esta forma, el código resultante para graficar las dos temperaturas será el de la figura 84.



```
Sensor_de_temperatura $
1 int sensor=A0;//se define el pin A0 como sensor
2 float lectura;//se define la variable lectura
3 float temperatura;//se define la variable temperatura
4 float temperaturaK;//se define la variable T en kelvin
5 void setup() { //se inicia el void setup
6 Serial.begin(9600);//se inicia la comunicación serial
7 }//se cierra la sección void setup
8 void loop() { //se inicia el voidloop
9 lectura=analogRead(sensor);//se realiza la lectura del pin A0
10 temperatura=(lectura*5*100)/1024;//se determina la temperatura
11 temperaturaK=temperatura+273;//se determina en Kelvin
12 Serial.print (temperatura);//se imprime la T
13 Serial.print(",");
14 Serial.println (temperaturaK);
15 delay (200);//se esperan cinco segundos por medición
16 }//se cierra la sección void loop
```

Figura 84. Código de la figura 79 modificado para realizar dos gráficas simultáneas en el Serial Plotter.

Fuente: elaboración propia.

Por último, las dos gráficas se visualizan en la figura 85. De color rojo se tiene la temperatura en grados Kelvin y de color azul en grados Celsius.

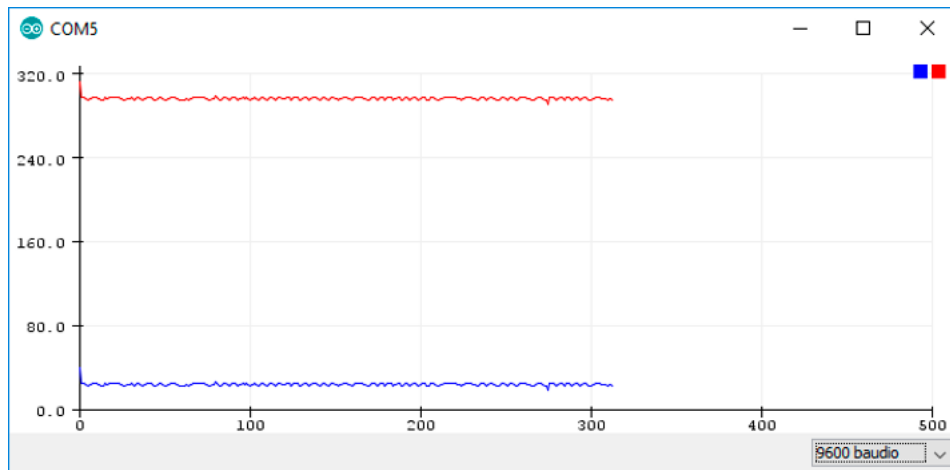


Figura 85. Serial Plotter de Arduino visualizando las dos gráficas de la temperatura.

Fuente: elaboración propia.

Sensor de gas

96

El último sensor que se propone es un sensor de gas. Este tipo de sensor fue utilizado por unos docentes de química y biología en uno de los seminarios de alfabetización científica y tecnológica de la maestría en Educación de la Universidad Distrital. La razón de utilizar un sensor como estos era solucionar unas problemáticas propias de las instituciones donde laboraban los docentes mencionados. El sensor que se requiere para este montaje es el MQ-135 (figura 86).

Recuerde que alguien alfabetizado en ciencia y tecnología utiliza herramientas como Arduino para solucionar problemáticas propias de su contexto.

Este sensor también cuenta con un ajuste de sensibilidad (encerrado de color rojo).



Figura 86. Sensor MQ-135

Fuente: elaboración propia.

El MQ-135 es un sensor de gas perteneciente a una familia de sensores electroquímicos, los cuales varían su resistencia interna ante la presencia de gases

en cierta concentración. Este sensor en particular detecta amoníaco, óxidos de nitrógeno, alcohol, benceno, humo, dióxido de carbono y monóxido de carbono.

El montaje a realizar se ilustra en la figura 87. En rojo, se tiene los +5 V del Arduino unidos con el terminal vcc del sensor. El jumper negro será la GND y se conectará al terminal GND del sensor. Por último, el terminal Ao del MQ135 se conecta al pin Ao del Arduino a través del jumper verde. El terminal digital del sensor se deja sin conexión.

Si usted desea, puede utilizar la salida digital y emplear el sensor como detector de algún tipo de gas en particular. Lo invitamos a que consulte otras fuentes para ello.

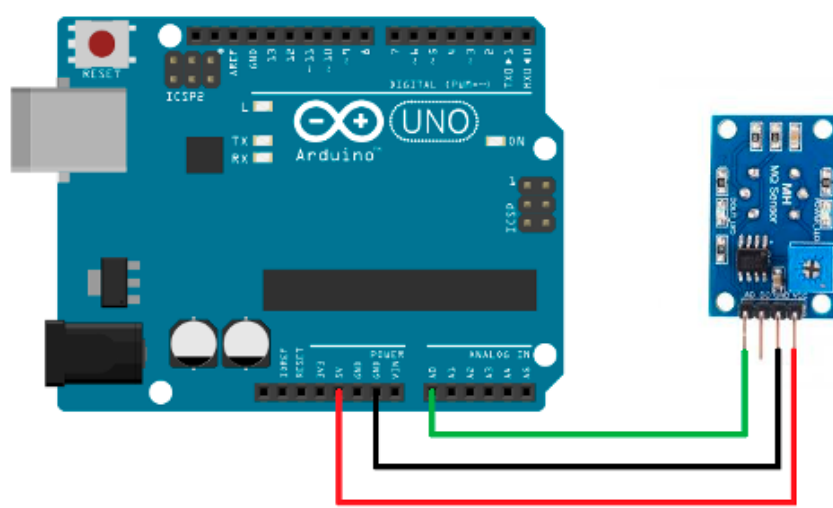


Figura 87. Diagrama de conexiones del sensor de gas

Fuente: elaboración propia.

El código utilizado se visualiza en la figura 88.

```
Sensor_de_gas $
1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() {
5   int MQ=analogRead(A0);
6   float vleido=MQ*5/1024;
7   float Rs= ((5*1000)/vleido)-1000;
8   double CO=521.88*pow((Rs/455.46), -2.294);
9   Serial.println("Concentración de CO en ppm");
10  Serial.println(String (CO)+"ppm");
11  delay(5000);
12 }
```

Figura 88. Código para sensor de gas

Fuente: elaboración propia.

Línea uno `void setup() { Línea dos Serial.begin(9600); y Línea tres }`

Se inicia el *void setup*, mediante la comunicación con el puerto serial a una velocidad de 9600 y se cierra la sección del *void setup* con el corchete de la tercera línea.

Línea cuatro `void loop() {`

Se inicia la sección del *void loop*, colocando el código pertinente entre el corchete abierto en esta línea y cerrado en la doceava.

Línea cinco `int MQ=analogRead(A0);`

Se nombra una variable de tipo entero (leerá valores enteros entre 0 y 1023 ya que es una entrada análoga) llamada MQ y corresponderá a lo leído por el Arduino en el pin A0 de la placa.

Línea seis `float vleido=MQ*5/1024;`

Se hace la transformación de dichos bits a voltaje, al igual como se hizo con el sensor de temperatura. En este caso, recuerde que cada bit corresponde a un valor de concentración de gas y, por lo tanto, a un valor de voltaje; siendo 0 bits una concentración mínima (voltaje de 0 V), y 1023 bits una concentración máxima, con voltaje de 5 V. De esta forma, entre 0 V y 5 V se tendrá 1024 posibles valores. Con la ecuación de esta línea, se hace la equivalencia correspondiente y se define en decimal (al ser tipo *float*).

Línea siete `float Rs= ((5*1000)/vleido)-1000;`

En esta línea se determina el valor de la resistencia equivalente del sensor. Para ello, tenga en cuenta que, en este caso particular, se requiere determinar su curva de calibración. Esto se hace de la siguiente forma:

1. El voltaje que Arduino entrega a la entrada del sensor es de 5 V. Se requiere determinar el voltaje que habrá a la salida (en RL), para lo cual se tiene el circuito equivalente de la figura 89.

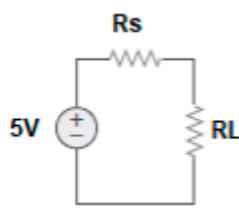


Figura 89. Circuito equivalente

Fuente: elaboración propia.

En este circuito, el sensor se representa como la resistencia R_s y el circuito se cierra con una resistencia R_L cuyo valor promedio para este tipo de sensores es de $1000\ \Omega$. Con esta información, es posible determinar la corriente que pasa en este circuito (ecuación 26).

$$I = \frac{5V}{R_s + 1000\Omega} \quad (26)$$

Luego, con ayuda de la ley de Ohm, es posible determinar el voltaje en R_L , el cual es el voltaje que censará Arduino en el pin Ao. Este voltaje se llama voltaje leído y se calcula con ayuda de la ecuación 27:

$$V_{leído} = \frac{5V * 1000\Omega}{R_s + 1000\Omega} \quad (27)$$

2. Determinar la curva de calibración para el gas a analizar.

Esto se realiza con la información del *datasheet* del fabricante del sensor, donde se encuentra la gráfica de la figura 90.

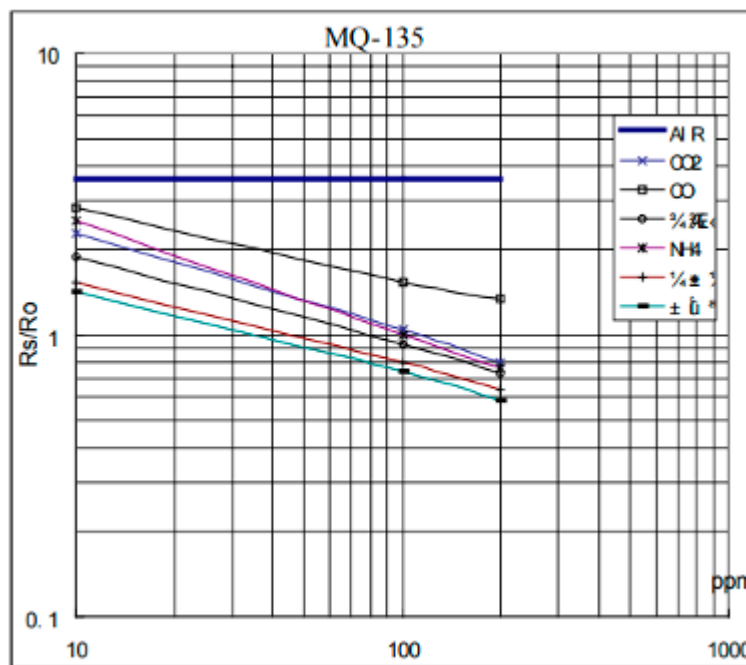


Figura 90. Curvas de calibración dadas por el fabricante para cada uno de los gases que percibe el sensor.

Fuente: ficha técnica de acceso libre.

En este caso, observe que no se tiene una gráfica lineal (por la escala de los ejes) y tampoco la información puntual de relación entre voltaje y gas (como sí se tenía la

temperatura y su equivalencia de voltaje del sensor LM35). Debido a esto, se debe identificar la mayor cantidad de puntos de la gráfica relacionada del gas al cual se desea determinar su concentración en ppm. Para este ejemplo, se utilizó la curva de monóxido de carbono (CO). Al identificar diez puntos de dicha gráfica, se logró construir en Excel la gráfica de la figura 91.

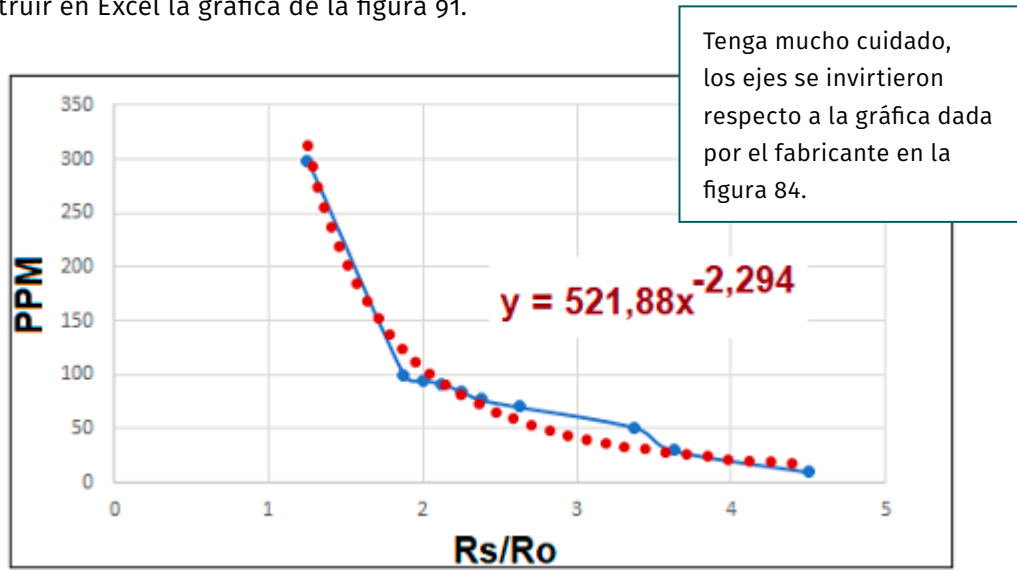


Figura 91. Gráfica de CO según los datos del fabricante en la figura 90.

Fuente: elaboración propia.

100

La gráfica de color azul fueron los valores estimados que se tomaron de la figura 91, para la curva de CO. Después, se realizó una regresión, agregando una línea de tendencia de tipo potencial, la cual se visualiza de color rojo en la figura 91. Note que también se obtuvo la ecuación con ayuda de Excel.

El eje y se llama concentración de CO en ppm y el eje x es el cociente R_s/R_o , por lo cual, la ecuación del gráfico se puede expresar como

$$CO = 521.88 \left(\frac{R_s}{R_o} \right)^{-2,294} \quad (28)$$

En la ecuación 28, R_o es una constante y R_s es la resistencia del sensor, la cual dependerá del valor de concentración del gas en cuestión. Por lo tanto, para conocer la concentración en ppm de monóxido de carbono, se requiere saber el valor de la resistencia del sensor y R_o .

3. Cálculo de R_s y R_o

Se despeja R_s de la ecuación 27, obteniendo la ecuación 29.

$$R_s = \frac{5000}{V_{leído}} - 1000 \quad (29)$$

Como es claro en la ecuación 29, R_s depende del voltaje leído en el terminal A0 del sensor y dicho valor equivale a una concentración determinada del gas. Sin embargo, para determinar los valores de concentración se requiere utilizar la ecuación 28, para la cual se necesita conocer el valor constante R_0 . Para ello, analice la figura 89. Note que, por el comportamiento exponencial, para altas concentraciones de CO el cociente R_s/R_0 presentará pocas variaciones y será muy cercano a 1. Con esta consideración, para una concentración de 321.57 ppm (último punto de la gráfica), el valor R_s/R_0 equivale a 1.235, como se expresa en la ecuación 30.

Este valor de R_s/R_0 se obtiene de la gráfica

$$\frac{R_s}{R_0} = 1,235 \quad (30)$$

De esta ecuación, es posible despejar R_0 con ayuda de la ecuación 31

$$R_0 = \frac{R_s}{1,235} \quad (31)$$

Sin embargo, para determinar el valor de R_0 se requiere conocer el valor de R_s (el cual depende del voltaje leído a la salida del sensor). De esta forma, solo se necesita conocer un valor de R_s para calcular la constante R_0 .

La forma de conocer un valor aproximado de R_s es recordando que, la concentración será equivalente a un valor de voltaje, siendo 0 V la concentración mínima del sensor y 5 V la concentración máxima. Considerando que los cálculos se hicieron para una concentración alta, se necesita un valor de voltaje entre 0 V y 5 V equivalente a concentraciones altas de gas. Experimentalmente se ha logrado evidenciar que voltajes mayores a 3 V representan una concentración alta de CO. Con ello, se asume un voltaje de 3.2 V como voltaje leído y se reemplaza en la ecuación 27, obteniendo un valor de R_s igual a 562.5 Ω . Este valor se reemplaza en la ecuación 31, obteniendo un valor de R_0 igual a 455.46 Ω .

Recuerde que R_s es la resistencia del sensor y cuyo valor dependerá de la concentración de gas (CO) que se tenga.

De esta forma, la ecuación 28 queda expresada como la ecuación 32

$$CO = 521,88 \left(\frac{R_s}{455,46} \right)^{-2,294} \quad (32)$$

Línea ocho `double CO=521.88*pow((Rs/455.46), -2.294);`

La ecuación 32 es la que permitirá calcular los valores de concentración de CO. Por lo tanto, antes de calcular el valor de concentración, se requiere conocer el valor de Rs para el voltaje que cense el Arduino en la salida del sensor (línea siete del código).

Para colocar la ecuación 32 dentro del código de Arduino se utiliza la función *pow()*, la cual se emplea para expresar potencias indicando la base y el exponente de la misma. Dentro del paréntesis de esta función se coloca primero la base y después, separado por una coma, el valor del exponente.

Línea nueve `Serial.println("Concentración de CO en ppm");` y **Línea diez** `Serial.println(String (CO)+"ppm");`

Finalmente, se imprime en el monitor serial un encabezado que dirá concentración de CO en ppm y en el renglón siguiente la lectura que se haga del sensor.

Línea once `delay(5000);` y **doce** `}`

En la figura 92 se visualiza la concentración de CO en ppm en el monitor serial de Arduino.

El fabricante recomienda esperar 20 s para empezar a censar (ya que es el tiempo estimado que tarda el sensor en calentarse). Además, especifica el rango de funcionamiento en partes por millón que detecta. Le recomendamos siempre buscar la información del *datasheet*.

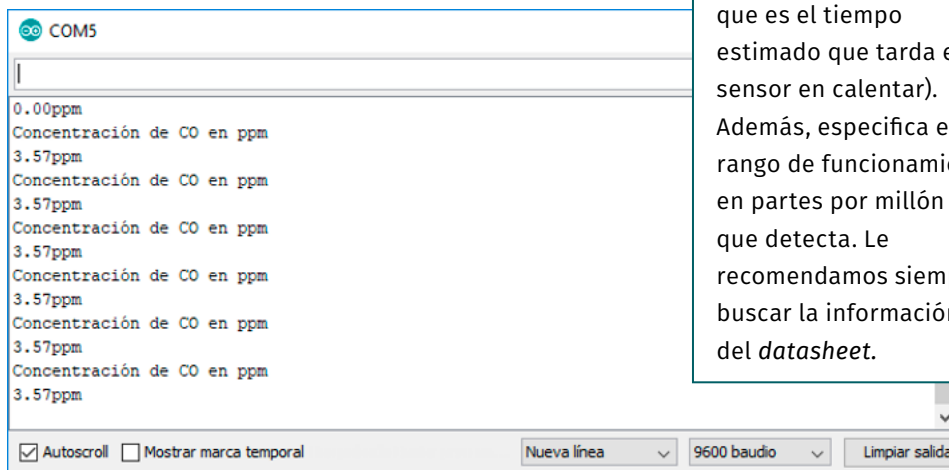


Figura 92. Lectura análoga del sensor de gas en el monitor serie de Arduino

Fuente: elaboración propia

Por favor, tenga en cuenta los errores que se introducen en cada medición. En el trabajo con sus estudiantes, no olvide reflexionar sobre la información de las hojas de especificaciones y los rangos de operación de cada sensor. Es una invitación a enriquecer su quehacer como docente.

Epílogo. ¿Y ahora?

“Fracasa una vez, fracasa mejor”

Nos complace saber que ha finalizado la lectura del libro. Esperamos que haya sido una experiencia de aprendizaje continuo y, sobre todo, que su intelecto se haya visto estimulado con cada una de las páginas que leyó. Ahora, le queremos dar luces respecto a qué hacer con el conocimiento adquirido.

En primera instancia, le queremos invitar a fortalecer su ejercicio reflexivo, con el objetivo de que el proceso de enseñanza-aprendizaje con sus estudiantes se vea beneficiado.

Contemple la alfabetización científico y tecnológica y el modelo didáctico de Mallart en sus cursos de Ciencias. Evalúe siempre que, en la sociedad actual, es necesario que todos los escenarios se vean permeados con herramientas TIC (incluido los formativos), y que es fundamental para ello el desarrollo práctico.

Tenga presente que este ejercicio de laboratorio debe ser pensado también. Para ello, *considere que el aspecto emocional es fundamental* al momento de interactuar con nuevas tecnologías. En otras palabras, *es de vital importancia garantizar el funcionamiento de los experimentos que como docente proponga a sus estudiantes, con el fin de evitar el rechazo de ellos ante la nueva tecnología.*

Ahora, nosotros como autores del texto somos los principales interesados en que este trabajo sea simplemente una puerta para que ingrese a un mundo de laboratorios mucho más profesionales en su ejercicio docente. Con base en esto, seguramente querrá utilizar otros sensores, captar información de otros procesos (acidez de sustancia, por ejemplo) que estén directamente relacionados con su área de conocimiento, e implementar sistemas de medición mucho más robustos (¿qué tal una estación meteorológica?).

Por esto, le queremos recalcar que a partir de ahora usted tiene todas las bases para buscar información por su cuenta y evaluar la pertinencia de experimentos similares a los que usted desea implementar, pero que han sido desarrollados por

otros (recortando de esta forma camino para usted). ¡Recuerde que acudir a fuentes confiables de información también es una habilidad de alguien alfabetizado en ciencia y tecnología!

Por último, anímese a proponer mejoras en aquellos experimentos que ya existen y tenga muy presente esto: ¡siempre estamos sobre hombros de gigantes! Y ¡del error, salimos mejor!

¡Muchos éxitos!

José Luis, Julián y José de Jesús

Referencias

Redacción Educación (2016, 29 de octubre). Mejor infraestructura, menos tasa de repetición. *El Espectador*. <https://www.elespectador.com/educacion/mejor-infraestructura-menos-tasa-de-repeticion-article-662939/>

Perfiles de los autores

José Luis Paternina Durán

Ingeniero eléctrico, magíster en Educación y candidato a doctor en Ingeniería Eléctrica. Cuenta con más de 8 años de experiencia en procesos investigativos y académicos. Entre sus líneas de interés, se encuentran las microrredes eléctricas, optimización matemática, eficiencia energética, máquinas eléctricas, formación de profesores, educación en ingeniería y vocación de carreras STEM. Actualmente es decano académico de la Facultad de Ingenierías TIC en la sede principal de la Universidad Santo Tomás.

Correos:

josepaterninad@gmail.com
dec.ingteleco@usta.edu.co

109

Julián Andrés Salamanca Bernal

Licenciado en Física, magíster en Física y doctor en Física Aplicada. Cuenta con una amplia trayectoria como investigador y profesor a nivel tanto nacional como internacional. Dentro de sus áreas de trabajo, están la alfabetización científica y tecnológica, la física experimental de partículas, la ciencia de datos, ciencias de la computación, ciencias atmosféricas y fuentes alternas de energía. Actualmente es profesor titular de la Facultad de Ciencias y Educación en la Universidad Distrital Francisco José de Caldas.

Correo: jasalamanca@udistrital.edu.co

José de Jesús Paternina Anaya

Ingeniero electrónico, especialista en Ingeniería de Software y magíster en Ciencias de la Información y las Comunicaciones. Es un referente académico de la ingeniería

electrónica en Colombia, con más de 30 años de experiencia como profesor universitario. Así mismo, ha liderado iniciativas corporativas basadas en soluciones de ingeniería para el sector productivo. Dentro de sus líneas de interés, se encuentran la educación en ingeniería, procesamiento de señales, electrónica de potencia y sistemas digitales. Actualmente es profesor asociado de la Facultad de Ingeniería en la Universidad Distrital Francisco José de Caldas.

Correo: jjpaterninaa@udistrital.edu.co

Esta obra se editó en Ediciones
USTA. Tipografías de la familia Fira
Sans y Share Tech.
2026

Modular

¿Cómo transformar el aula en un laboratorio de innovación tecnológica? Arduino para docentes: Hands on! Alfabetización científica y tecnológica es una guía diseñada para empoderar a los educadores en la integración de herramientas electrónicas de bajo costo y alto impacto. Los autores proponen superar el temor a la nueva tecnología, brindando las bases necesarias para implementar sistemas de medición, uso de sensores y recolección de datos en tiempo real. Más que un manual técnico, esta obra es una invitación a la alfabetización científica: fomenta el aprendizaje basado en la experimentación y el desarrollo de proyectos robustos, desde la química hasta la meteorología. Una publicación de Ediciones USTA imprescindible para los docentes que buscan revolucionar su enseñanza y construir conocimiento científico codo a codo con sus estudiantes. ¡Del error, salimos mejor!

