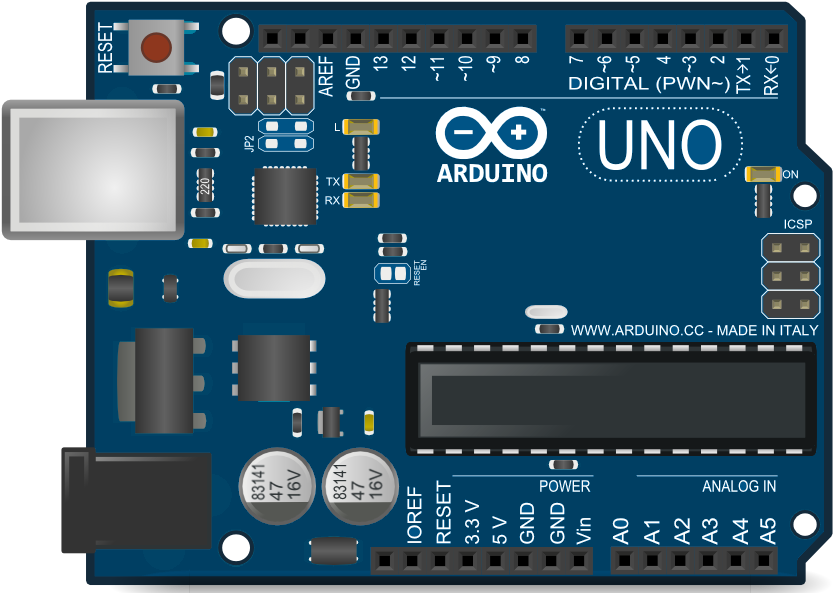
# ARDUINO



GRECIA GENESIS ROBLES PICAZO

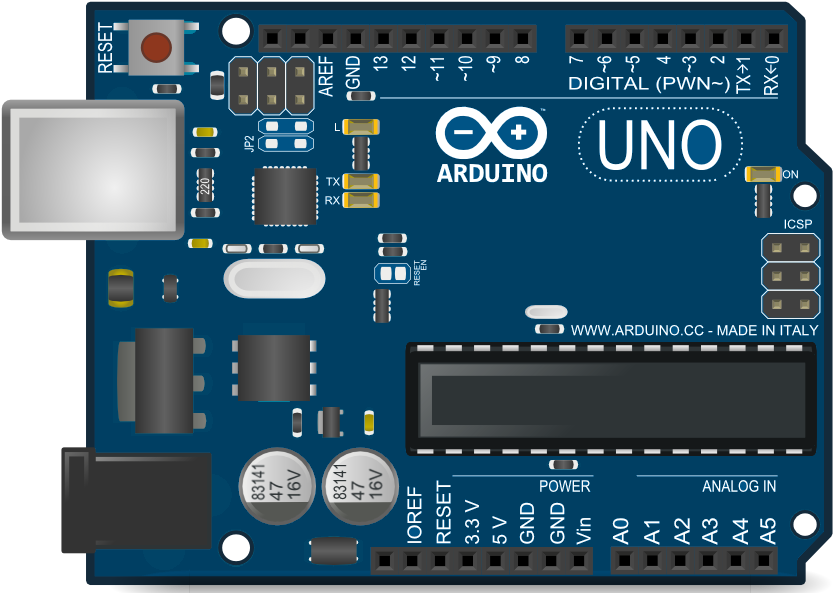
GRUPO 3 SISTEMAS

M.C LINDA PATRICIA MARTINEZ RODRIGUEZ

# PRACTICA 1 INTRODUCCION A ARDUINO Y MICROCONTROLADORES

* **Arduino**:

Arduino es una plataforma de hardware libre que permite a los usuarios crear proyectos interactivos utilizando una placa con un microcontrolador. Este microcontrolador puede ser programado para controlar una variedad de dispositivos electrónicos, como luces, motores y sensores. Arduino es popular entre aficionados y profesionales por su facilidad de uso y su amplia comunidad de soporte. Los proyectos con Arduino pueden ir desde simples experimentos hasta sistemas complejos de automatización y robótica.



* **Microcontrolador**:

Un microcontrolador es un chip integrado que actúa como el cerebro de un dispositivo electrónico. Incluye un procesador, memoria y puertos de entrada/salida, todo en un solo chip. Este componente permite que dispositivos como electrodomésticos, automóviles y placas Arduino realicen funciones específicas al ejecutar programas de control. Los microcontroladores son fundamentales en la creación de sistemas embebidos que automatizan tareas en una variedad de aplicaciones tecnológicas.



* Hardware:

El hardware se refiere a los componentes físicos de una computadora o sistema electrónico, tales como la placa base, el procesador, la memoria y los dispositivos periféricos como teclados y ratones. A diferencia del software, que son los programas y sistemas operativos, el hardware es tangible y se puede ver y tocar. Los componentes de hardware trabajan juntos para ejecutar y soportar las operaciones del software, permitiendo el funcionamiento de sistemas computacionales y electrónicos.



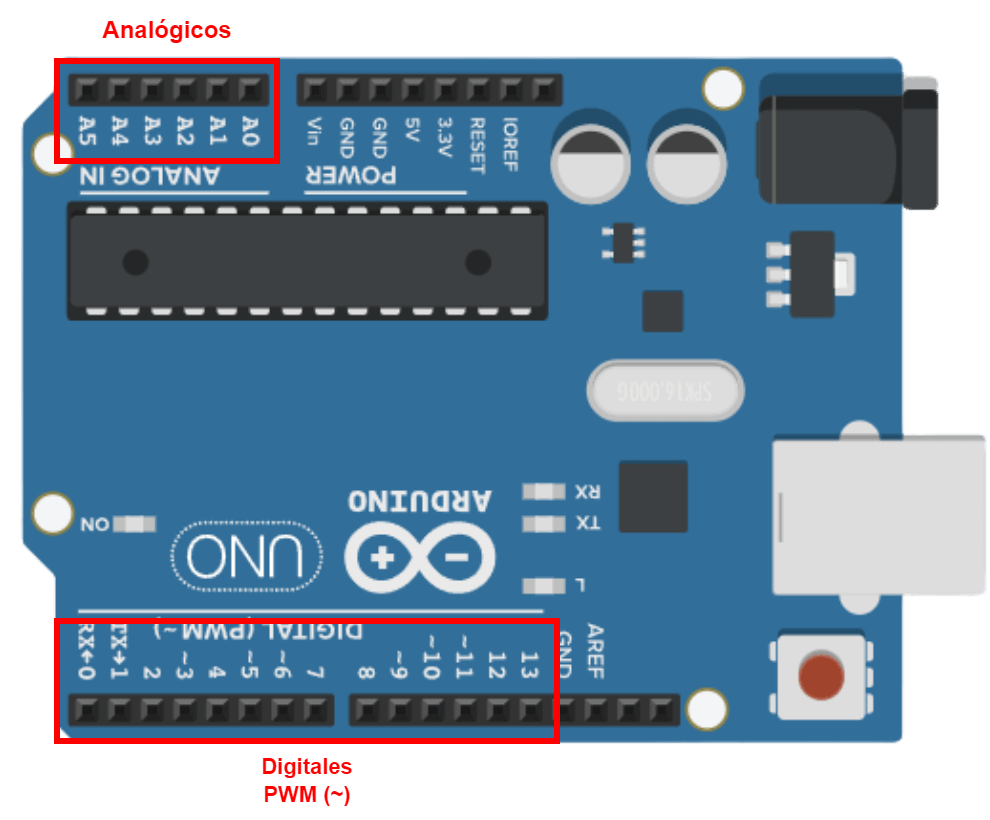
* **Software**:

El software comprende los programas y aplicaciones que permiten a una computadora o dispositivo electrónico realizar tareas específicas. Esto incluye sistemas operativos, aplicaciones de oficina, juegos y software especializado para controlar hardware. A diferencia del hardware, el software no tiene una forma física; en cambio, se compone de código que se ejecuta en el hardware para realizar funciones y procesos. El desarrollo de software implica escribir código en lenguajes de programación para crear aplicaciones que satisfagan necesidades particulares.



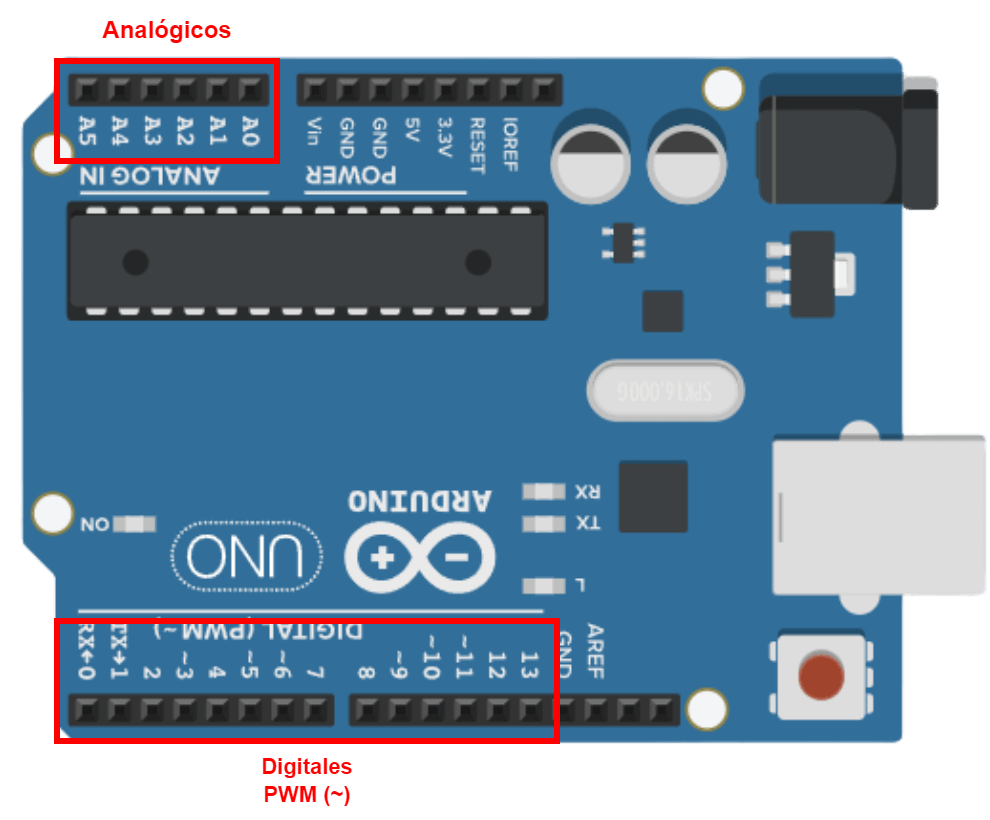
* **Pines digitales de Arduino**:

Los pines digitales en una placa Arduino son conexiones que permiten la entrada y salida de señales digitales, que son señales que solo pueden estar en uno de dos estados: encendido o apagado. Estos pines se usan para interactuar con dispositivos como LEDs, interruptores y relés. Los pines digitales se pueden configurar como entradas para leer datos o como salidas para enviar señales, facilitando el control y la comunicación con otros componentes electrónicos en proyectos de Arduino.



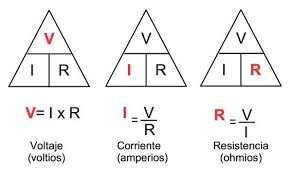
* **Pines analógicos de Arduino**:

Los pines analógicos de Arduino son conexiones que permiten medir señales analógicas, es decir, señales que varían de forma continua en lugar de tener solo dos estados. Estos pines pueden leer voltajes de entrada que cambian suavemente, como los que producen sensores de temperatura o potenciómetros. A diferencia de los pines digitales, que solo detectan estados de encendido o apagado, los pines analógicos convierten señales continuas en valores digitales que el microcontrolador puede procesar.



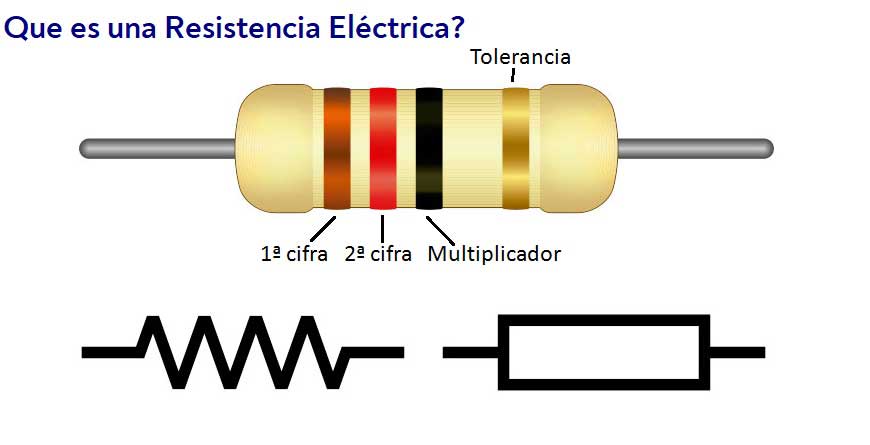
* **Ley de Ohm**:

La Ley de Ohm es una fórmula fundamental en electrónica que describe la relación entre el voltaje, la corriente y la resistencia en un circuito eléctrico. Según esta ley, el voltaje (V) es igual a la corriente (I) multiplicada por la resistencia (R), expresado como V=I×RV = I \times RV=I×R. Esta relación es crucial para diseñar y analizar circuitos eléctricos, ya que permite calcular cómo cambiarán el voltaje o la corriente al ajustar la resistencia en un circuito.



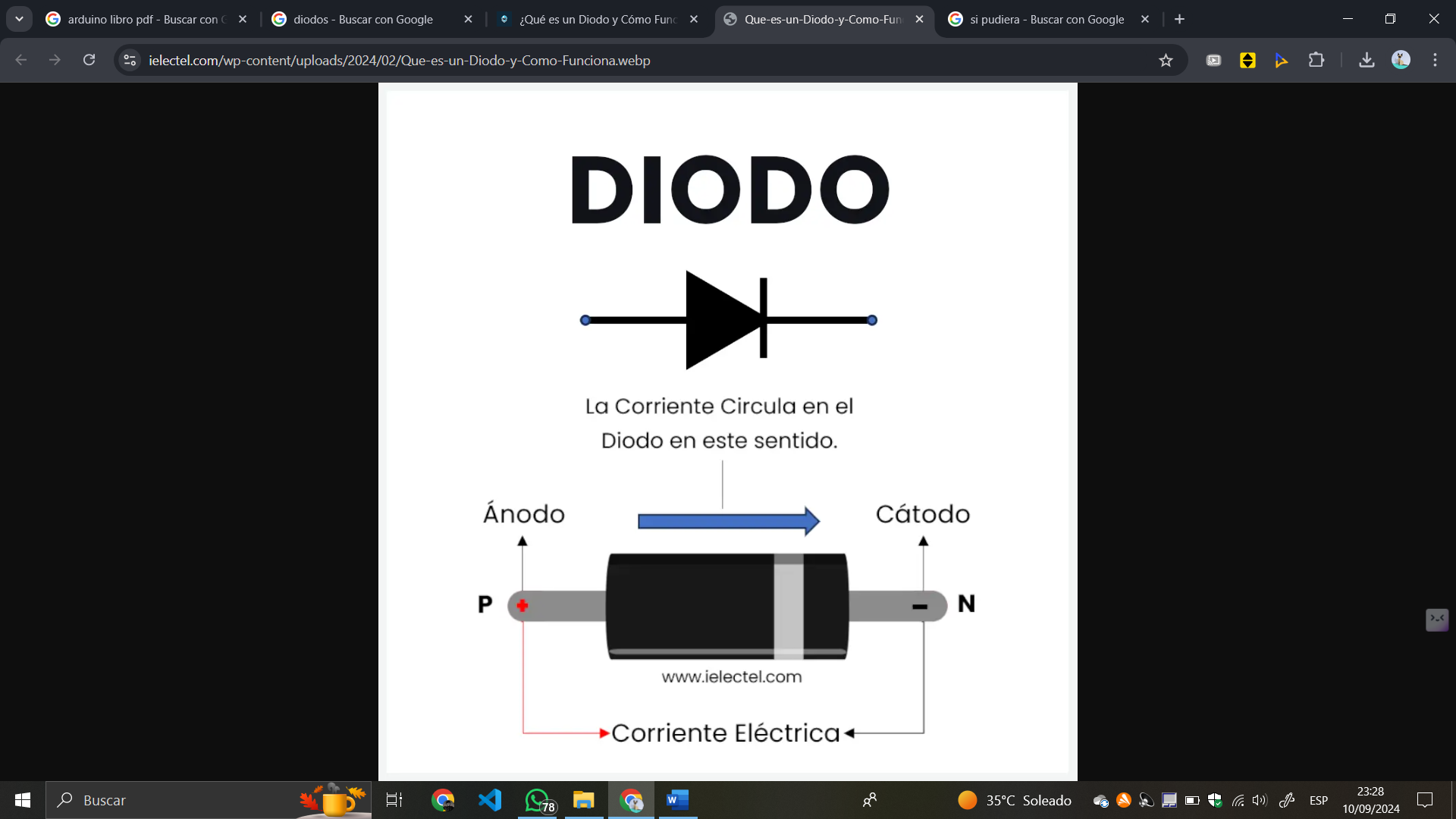
* **Resistencias**:

Las resistencias son componentes electrónicos que se utilizan para limitar el flujo de corriente en un circuito. Se colocan en serie con otros componentes para protegerlos de la sobrecarga de corriente, controlar el voltaje en diferentes partes del circuito y dividir la tensión en varias partes. Las resistencias vienen en diferentes valores y tamaños, y su valor se mide en ohmios. Al elegir una resistencia adecuada, se puede ajustar el comportamiento del circuito para que funcione correctamente.



* **Diodos**:

Los diodos son componentes electrónicos que permiten que la corriente fluya en una sola dirección mientras bloquean el flujo en la dirección opuesta. Esto es útil para convertir corriente alterna (AC) en corriente continua (DC), proteger circuitos de voltajes inversos y detectar señales. Los diodos tienen una característica llamada "polaridad" que determina en qué dirección permiten el flujo de corriente. Su aplicación incluye rectificación en fuentes de alimentación y protección de circuitos.



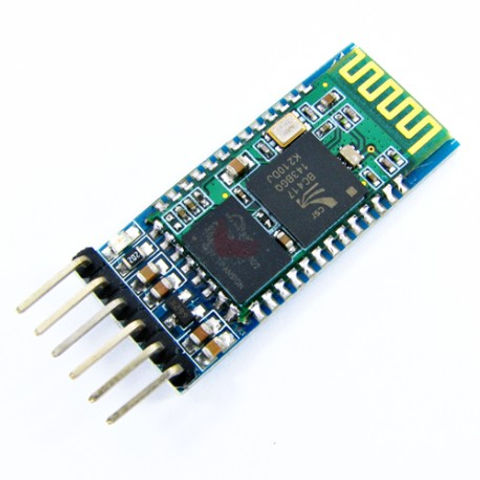
* **Módulo WiFi de Arduino**:

Los módulos WiFi para Arduino permiten que la placa Arduino se conecte a redes inalámbricas, proporcionando acceso a internet. Esto facilita la creación de proyectos que pueden enviar y recibir datos a través de la web, como aplicaciones de monitoreo remoto o sistemas de control basados en la nube. Los módulos WiFi más comunes incluyen el **ESP8266** y el **ESP32**, que son populares por su capacidad de integrar conectividad WiFi en proyectos de Arduino de manera sencilla y económica.



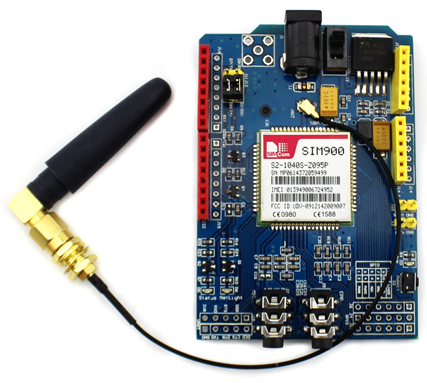
* **Módulo Bluetooth de Arduino**:

Los módulos Bluetooth para Arduino permiten la comunicación inalámbrica con otros dispositivos Bluetooth, como teléfonos inteligentes, tablets o computadoras. Esto es útil para crear aplicaciones que requieran control remoto o intercambio de datos entre el Arduino y otros dispositivos. Ejemplos de módulos Bluetooth incluyen el HC-05 y el HC-06, que son ampliamente utilizados por su facilidad de integración y configuración en proyectos de Arduino.



* **Módulo SIM/GSM de Arduino**:

Los módulos SIM/GSM para Arduino proporcionan la capacidad de comunicación a través de redes móviles, permitiendo que el Arduino envíe y reciba mensajes de texto (SMS) o datos. Esto es ideal para proyectos que necesitan operar en áreas sin acceso a internet WiFi o para aplicaciones de seguimiento y notificación a través de redes móviles. Los módulos SIM/GSM, como el SIM800 y el SIM900, permiten la creación de sistemas de comunicación robustos y versátiles en proyectos de Arduino.



* **IDE de Arduino**:

El entorno de desarrollo integrado (IDE) de Arduino es el software que se utiliza para escribir, compilar y cargar código en las placas Arduino. Ofrece un editor de texto para escribir programas, una consola para ver mensajes de error y un compilador que convierte el código en instrucciones que la placa puede ejecutar. El IDE facilita la programación de Arduino al proporcionar una interfaz simple y herramientas específicas para trabajar con esta plataforma, y es esencial para desarrollar proyectos con Arduino.



* **Lenguaje de programación Arduino**:

El lenguaje de programación utilizado para Arduino es una versión simplificada del lenguaje C/C++. Está diseñado para ser fácil de aprender y usar, con funciones y estructuras específicas para trabajar con la placa y sus componentes. Los programas escritos en este lenguaje se conocen como "sketches" y se cargan en la placa Arduino para controlar el hardware conectado. La simplicidad del lenguaje permite a los principiantes y a los expertos desarrollar proyectos rápidamente y con menos complicaciones.



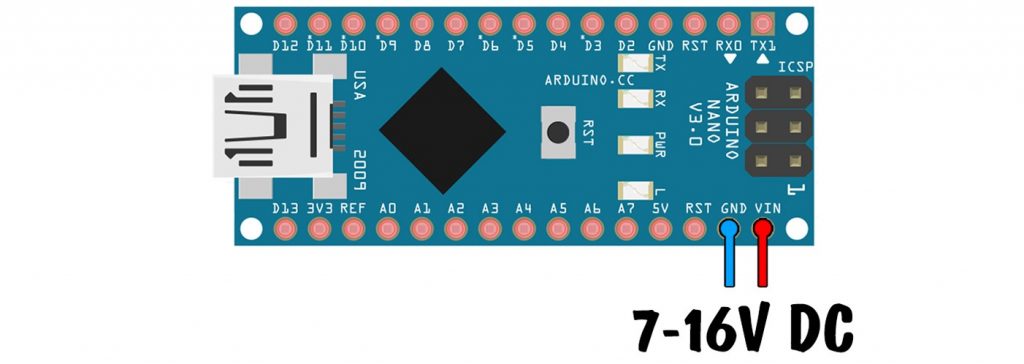
* **Voltaje**:

El voltaje es la medida de la diferencia de potencial eléctrico entre dos puntos en un circuito. Se mide en voltios (V) y es lo que impulsa la corriente a través de los componentes del circuito. El voltaje determina cuánta energía se transferirá a los componentes y afecta el funcionamiento de dispositivos electrónicos. En un circuito eléctrico, el voltaje es fundamental para el flujo de corriente y el funcionamiento adecuado de los dispositivos conectados.



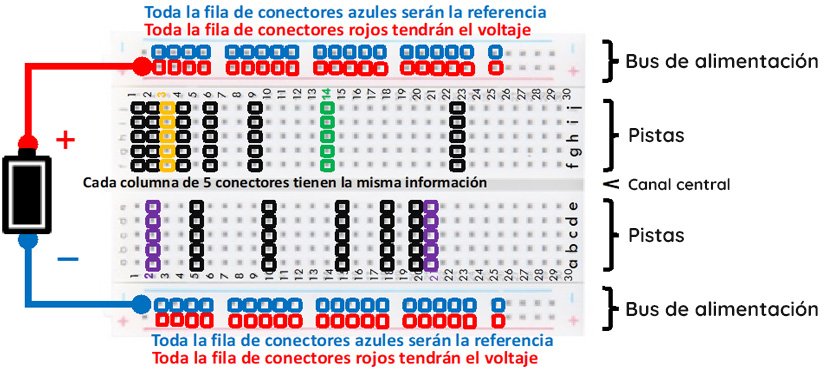
* **Voltaje que soporta Arduino**:

Las placas Arduino están diseñadas para soportar un rango específico de voltajes en su pin de alimentación, generalmente entre 7 y 12 voltios. Esto permite que la placa opere de manera segura y eficiente al convertir el voltaje de entrada a los niveles necesarios para el microcontrolador y otros componentes. El voltaje de operación del microcontrolador de Arduino es típicamente de 5V o 3.3V, y el diseño de la placa asegura que no se excedan estos límites para evitar daños.



* **Protoboard**:

Una protoboard, también conocida como placa de pruebas, es una herramienta utilizada para construir prototipos de circuitos electrónicos sin necesidad de soldar los componentes. Consiste en una matriz de agujeros conectados en filas y columnas, lo que permite insertar componentes y cables para construir circuitos de manera temporal. La protoboard es útil para experimentar y realizar pruebas antes de realizar un diseño final soldado en una placa de circuito impreso (PCB).



# Bibliografía

**Bibliografía:**

1. *Arduino Cookbook* por Michael Margolis. Más en [Arduino.cc](https://www.arduino.cc/).
2. *Microcontroladores: Diseño y programación* por José Luis Rivas. Información básica en [Microchip](https://www.microchip.com/).
3. *Hardware: The Definitive Guide* por Andrew S. Tanenbaum. Artículos explicativos en Techopedia.
4. *Software Engineering: A Practitioner's Approach* por Roger S. Pressman. Introducción en HowStuffWorks.
5. Documentación oficial de Arduino: Arduino Digital Pins y Arduino Analog Pins.
6. *Principios de Electricidad y Electrónica* por Charles Alexander y Matthew Sadiku. Explicación en Electronics Tutorials.
7. *Electronic Devices and Circuit Theory* por Robert L. Boylestad y Louis Nashelsky. Más detalles en Electronics Tutorials.
8. Tutoriales en el sitio de Arduino y en plataformas como [SparkFun](https://www.sparkfun.com/).
9. Documentación oficial en Arduino IDE.
10. *Arduino Programming in 24 Hours* por Richard Blum. Más en Arduino Reference.
11. *Electricity and Magnetism* por Edward Purcell. Explicación en Electronics Tutorials.
12. Documentación oficial en Arduino Specifications.
13. *Make: Electronics: Learning by Discovery* por Charles Platt. Más en All About Circuits.