

**1**

**Conociendo Arduino®**

**Objetivo**

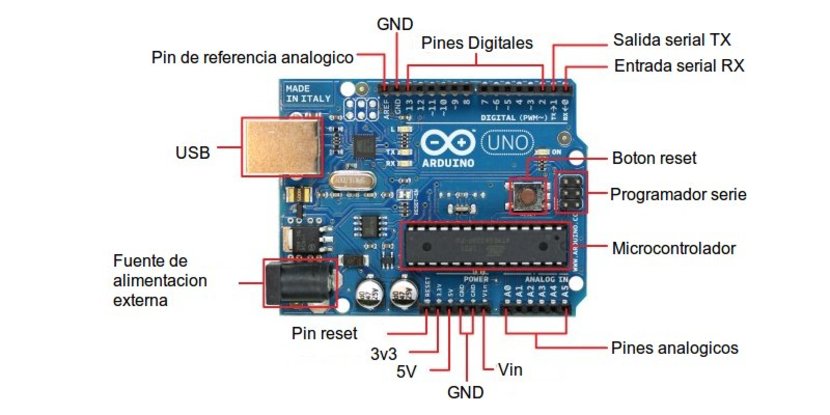
﻿Esta guía intenta ser una forma de acercarse al diseño y desarrollo de proyectos basados en Arduino para aquellas personas que nunca han trabajado con él pero que poseen el interés por iniciarse en este campo, cada día más accesible.

**¿Qué es Arduino®?**

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede “sentir“ el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en una computadora (por ejemplo con Flash, Processing, MaxMSP, etc.).Las placas se pueden ensamblar a mano o encargarlas preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

**La Placa Arduino®**



Detalle de la placa de Arduino® Uno y sus partes

Existe una amplia variedad de placas o plataformas Arduino® que se adaptan a las necesidades de diseño, función o aplicación de la misma. Puedes acceder al catálogo oficial escaneando el código QR proporcionado para tal efecto.



Catálogo

de Arduino

**Características**

﻿Arduino es una placa con un microcontrolador de la marca Atmel y con toda la circuitería de soporte, que incluye: Reguladores de tensión, un puerto USB (En los últimos modelos, aunque el original utilizaba un puerto serie) conectado a un módulo adaptador USB-Serie que permite programar el microcontrolador desde cualquier PC de manera cómoda y también hacer pruebas de comunicación con el propio chip.

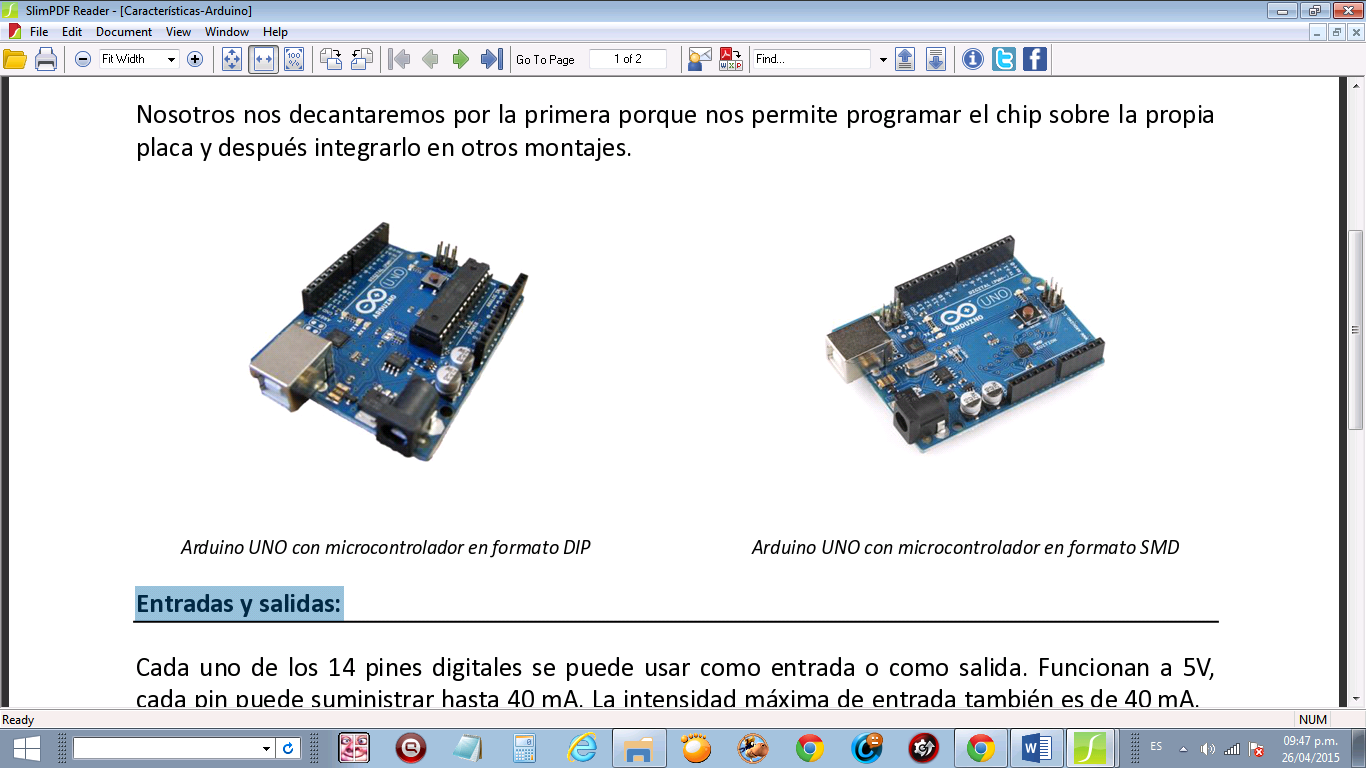
Un arduino dispone de 14 pines que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V.

También dispone de entradas y salidas analógicas. Mediante las entradas analógicas podemos obtener datos de sensores en forma de variaciones continuas de un voltaje. Las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM.

Arduino UNO es la última versión de la placa, existen dos variantes, la Arduino UNO convencional y la Arduino UNO SMD. La única diferencia entre ambas es el tipo de microcontrolador que montan:

* ﻿La primera es un microcontrolador Atmega en formato DIP.
* Y la segunda dispone de un microcontrolador en formato SMD.

Nosotros nos decantaremos por la primera porque nos permite programar el chip sobre la propia placa y después integrarlo en otros montajes.

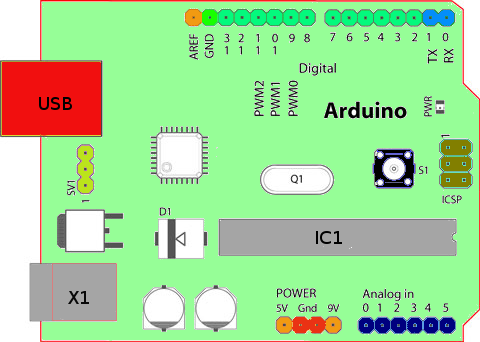


Arduino Uno con microcontrolador en formato SMD

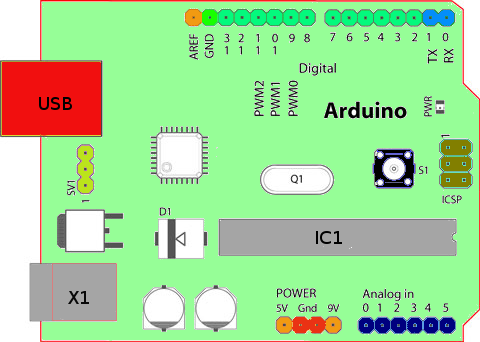
Arduino Uno con microcontrolador en formato DIP

﻿

**Pines Digitales y Analógicos**

Cada uno de los 14 pines digitales se puede usar como entrada o como salida. Funcionan a 5V, cada pin puede suministrar hasta 40 mA. La intensidad máxima de entrada también es de 40 mA.

Cada uno de los pines digitales dispone de una resistencia de pull-up interna de entre 20KΩ y 50 KΩ que está desconectada, salvo que nosotros indiquemos lo contrario.

Arduino también dispone de 6 pines de entrada analógicos que trasladan las señales a un conversor analógico/digital de 10 bits.

**Pines especiales de entrada y salida**

* **RX y TX:** Se usan para transmisiones serie de señales TTL.
* **Interrupciones externas:** Los pines 2 y 3 están configurados para generar una interrupción en el atmega. Las interrupciones pueden dispararse cuando se encuentra un valor bajo en estas entradas y con flancos de subida o bajada de la entrada.
* **PWM:** Arduino dispone de 6 salidas destinadas a la generación de señales PWM de hasta 8 bits.
* **SPI:** Los pines 10, 11, 12 y 13 pueden utilizarse para llevar a cabo comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.
* **I2C:** Permite establecer comunicaciones a través de un bus I2C. El bus I2C es un producto de Phillips para interconexión de sistemas embebidos. Actualmente se puede encontrar una gran diversidad de dispositivos que utilizan esta interfaz, desde pantallas LCD, memorias EEPROM, sensores...

**Alimentación de Arduino**

﻿Puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un pequeño transformador o, por ejemplo una pila de 9V. Los límites están entre los 6 y los 12 V. Como única restricción hay que saber que si la placa se alimenta con menos de 7V, la salida del regulador de tensión a 5V puede dar menos que este voltaje y si sobrepasamos los 12V, probablemente dañaremos la placa.

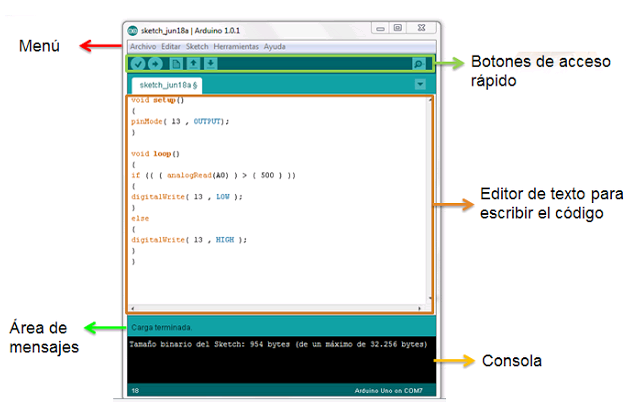
La alimentación puede conectarse mediante un conector de 2,1mm con el positivo en el centro o directamente a los pines Vin y GND marcados sobre la placa.

Hay que tener en cuenta que podemos medir el voltaje presente en el jack directamente desde Vin. En el caso de que el Arduino esté siendo alimentado mediante el cable USB, ese voltaje no podrá monitorizarse desde aquí.

**IDE Arduino®**

Arduino también incluye un entorno interactivo de desarrollo (IDE) que permite programar fácilmente la tarjeta. El IDE (Integrated Development Environment) se basa en Processing y Wiring (open source ).

El Lenguaje de programación Arduino se basa en C/C++, y se simplifica con el uso de la biblioteca Arduino.



Captura de pantalla del IDE de Arduino

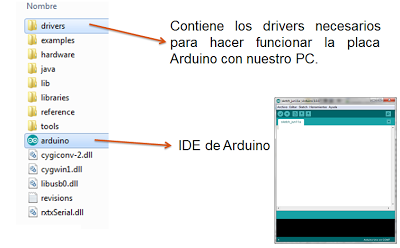
**Instalando el IDE de Arduino**



Descargar IDE

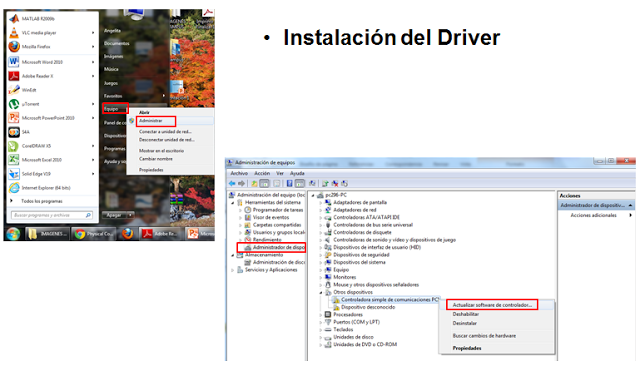
Descargar la IDE de Arduino de la página oficial según el sistema operativo.

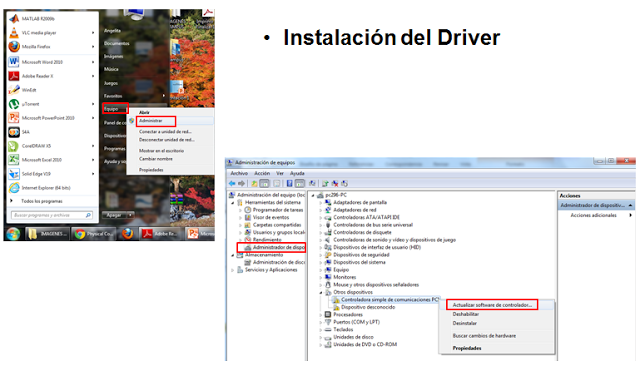
**Instalación de Arduino en Windows**



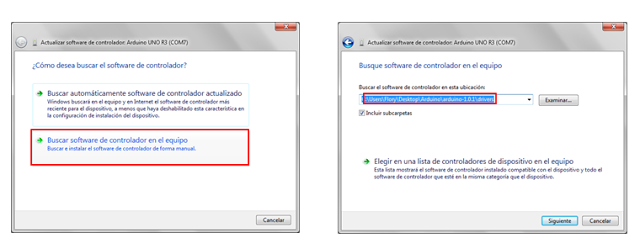
**Instalación del Driver.**

Vamos a administrar dispositivos (Inicio-clic derecho en equipo - administrar- administrar dispositivos) buscamos en otros dispositivos los elementos desconocidos y con clic derecho seleccionamos actualizar controlador.





Ahora seleccionamos buscar software de controlador en el equipo y buscamos la ubicación de carpeta Drivers.



**Configuración del IDE de Arduino**

Lo primero que se debe realizar es seleccionar el tipo de tarjeta que se está trabajando y el puerto en el que esta conectada la tarjeta:

