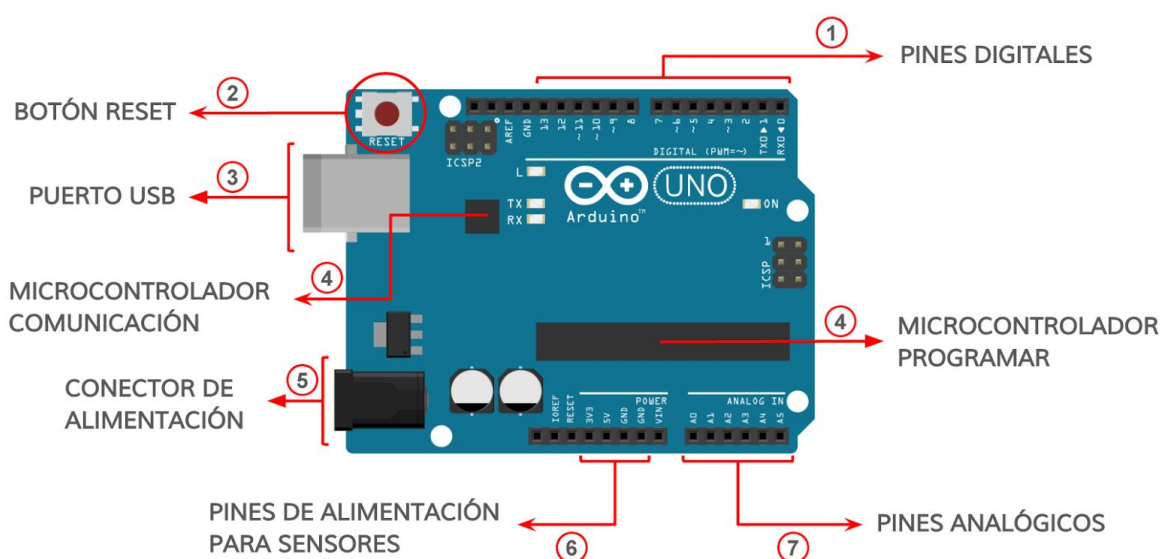


# Las 7 claves de Arduino UNO

La mejor placa para iniciarse en este mundo es el Arduino UNO. Aunque podríamos empezar con cualquier otra, prácticamente todas tienen las mismas funcionalidades básicas, Arduino UNO nos ofrece una relación prestaciones-precio muy buena y por lo tanto, es la mejor opción para iniciarse en este mundo de la electrónica y la programación.

Lo primero que debemos conocer es ¿qué nos ofrece esta placa a nivel de hardware? Yo te voy a dar las **7 claves que debes conocer** antes de empezar a crear tus proyectos.



## #1. Pines digitales

Los pines digitales son una de las ventanas con el mundo exterior. Arduino cuenta con 14 pines digitales a los que podemos conectar sensores, LEDs, servomotores y demás componentes que funcionen en un entorno digital.

Luego hablaremos del mundo analógico, pero es importante saber diferenciar estos dos mundos, el digital y el analógico.

Una señal digital solo puede tener dos estados o 0 o 1. También se representa como LOW (Bajo) o HIGH (Alto), o como 0V o 5V (V significa Voltios, la magnitud que mide el voltaje).

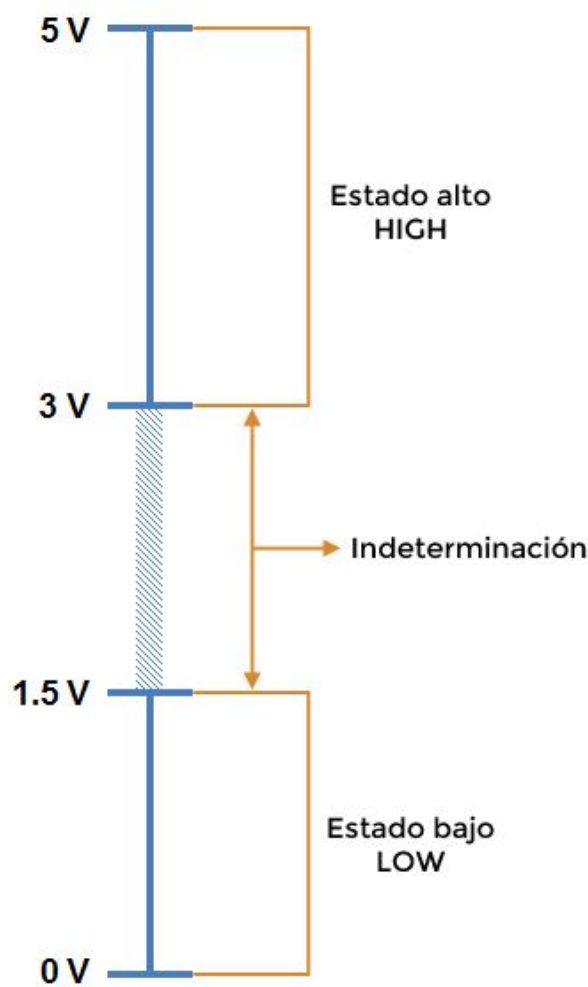
Por lo tanto, cuando le decimos a Arduino que ponga un pin digital en estado alto (HIGH), este suministra 5V por la salida que hayamos indicado.

Es sencillo, los pines digitales solo permiten tener 5V o 0V, o todo o nada. Hay placas que funcionan con otro voltaje de operación, 3,3V. En este caso sería de 3,3V a 0V.

Esto nos puede servir para alimentar un LED, activar un relé o simplemente para transmitir información binaria, unos y ceros.

Pero como no vivimos en un mundo ideal, en muchas ocasiones no tenemos 0V o 5V exactamente. Lo normal es que tengamos un rango de valores por el cual una señal digital sea un estado bajo o estado alto.

En la siguiente imagen puedes ver los rangos para determinar el estado de una entrada digital.



Estados lógicos para el microcontrolador  
ATMega328

Por lo tanto, no hace falta tener 5V para tener un estado alto o HIGH. A partir de 3V se considera este estado. Lo mismo ocurre con el estado bajo o LOW. A partir de 1,5V se considera estado bajo.

## #2. Botón reset

Todas las placas de Arduino tienen un botón reset. Este botón permite reiniciar el programa o sketch que tenemos cargado en la placa.

Inicia la secuencia de ejecución (lo veremos más adelante) y comienza ejecutando las importaciones de librerías externas, declaración de constantes y variables, ejecución función `setup()` y por último la función `loop()`.



El botón reset no elimina el programa que hemos cargado ni altera el código. Solo resetea la placa para empezar a ejecutar el sketch desde el principio.

## #3. Puerto USB

De alguna manera tenemos que comunicarnos con la placa bueno, más concreto con el microcontrolador. Esto lo hacemos a través del puerto serie.

Esta forma de comunicarse no es exclusiva de Arduino, los ratones y teclados utilizan el mismo protocolo para mandar órdenes al ordenador.

En el caso de Arduino (también en componentes como ratones y teclados) tenemos una interfaz USB por comodidad. Esto es debido a que es más común encontrar este tipo de conexiones en los ordenadores actuales. Pero al final, la comunicación es serie.

¿Para qué se utiliza el puerto USB en Arduino?

Principalmente tiene tres tareas. Por un lado nos permite cargar los programas a la memoria del microcontrolador, y por otro lado permite que Arduino envíe información al ordenador.

Esta información la mostramos a través del monitor serie y gracias a ella, podemos ver los datos de temperatura de un sensor, el valor de una entrada analógica o digital y cualquier tipo de información susceptible de ser mostrada.

Además, permite que alimentemos la placa a través de este puerto. Cuando la conectamos al ordenador, automáticamente estamos suministrando energía a la placa para que funcione correctamente.

## #4. Microcontroladores

Como ya he comentado anteriormente, la placa de Arduino no es sólo un microcontrolador. En ella encontramos más componentes que hacen sencillo trabajar con estos dispositivos. En concreto, Arduino UNO tiene dos microcontroladores.

El primer microcontrolador se encuentra situado al lado del botón reset. Se encarga de gestionar la comunicación a través del puerto serie (puerto USB). Esa es su tarea específica.

El otro microcontrolador es el cerebro de la placa. Aquí es donde vamos a cargar los programas o sketch con la lógica necesaria para que haga lo que nosotros queremos.

Arduino trabaja principalmente con una única marca, ATMEL, aunque también podemos encontrar algún tipo de Arduino que tiene microcontroladores INTEL. En concreto, el Arduino UNO viene integrado con un ATmega328P. En la siguiente tabla te muestro la características más importantes de este microcontrolador.

Microcontrolador	ATmega328P
Voltaje de operación	5V
Alimentación externa (Voltaje)	De 6V a 20V
Memoria Flash (Donde se cargan los programas)	32 KB
SRAM (Para almacenar valores variables, info volátil)	2 KB
EEPROM (info no volátil, permanente)	1 KB



Cuando hablamos de componentes electrónicos, es muy importante ver la hoja de características técnicas. Nos da mucha información sobre el componente. Si quieres saber más sobre el microcontrolador ATmega328P puedes ver su hoja de características técnicas o datasheet [aquí](#).

## #5. Conector de alimentación

Esta sería la segunda opción para alimentar el Arduino UNO. Ya hemos visto que lo podemos hacer a través del puerto USB. Este conector nos permite suministrar energía con unas pilas o batería.

Puede resultar muy útil en muchos proyectos, cuando queramos que nuestro Arduino UNO funcione de manera autónoma.



Ojo con alimentar con más voltaje que lo que nos indica la hoja de características técnicas. Esto puede dañar la placa. Lo recomendable es entre 7V y 12V.

## #6. Pines de alimentación para sensores

Dentro de todos los pines que tenemos a nuestra disposición, tenemos una serie de ellos que nos permiten alimentar componentes externos.

La placa de Arduino UNO tiene un pin que suministra 5V y otro de 3,3V. Además disponemos de dos pines de tierra marcados como GND (viene de ground, tierra en inglés).

Estos pines pueden ser utilizados para alimentar sensores u otros componentes como servomotores a nuestro Arduino.



Hay que tener precaución a la hora de alimentar componentes externos sobre todo cuando estamos trabajando con pilas o baterías. Puede que la corriente no sea suficiente. Lo mejor es comprobar con un multímetro.

## #6. Pines analógicos

Ya hemos vistos los pines digitales, que solo pueden tener dos estados o alto o bajo. Con los pines analógicos vamos a poder medir un rango de voltaje pero con limitaciones. Lo primero que tenemos que saber es que estos pines solo pueden medir entre 0V y 5V.

En un mundo ideal, cualquier voltaje entre 0V y 5V podría ser medido por un pin analógico. La realidad es otra.

Entre 0 y 5 existen infinitos valores. Imaginate el 3,15141672657858. Este es un valor dentro del rango pero que es imposible medirlo con un pin analógico.

Esto es debido a la resolución. Como máximo podemos medir 10-bits. Por lo tanto vamos a tener 10-bits que nos dirán que voltaje tenemos a la entrada del pin analógico.

El 0 en binario con 10 bits es 0000000000 (diez ceros). El número más grande que podemos representar con 10-bits es el 1111111111 (diez unos) que equivaldría a 1023 en decimal.

Así que, dentro del programa o sketch que hemos cargado a Arduino, cuando leemos el pin analógico nos dará un valor de 0 a 1023. Además nos lo dará en decimal. Como las entradas analógicas sólo pueden medir de 0V a 5V, entonces el 0 corresponderá a los 0V y el 1023 corresponderá a los 5V.

Pero ¿qué pasa si nos da un valor de 257? Resulta sencillo aplicar una simple regla de 3 para obtener los Voltios que corresponden a esa medida.

$$\begin{aligned} \text{Si } 5V &\Rightarrow 1023 \\ \text{Entonces } nV &\Rightarrow 257 \end{aligned}$$

$$5 \times 257 = n \times 1023 \Rightarrow n = \frac{5 \times 257}{1023} = 1,2561094V$$

Aproximadamente podemos decir que se pueden medir cambios de voltaje de 5 mV (4,88 mV).



Las unidades más comunes para medir los voltajes son los Voltios (V) y los Milivoltios (mV). 1 Voltio equivale a 1.000 Milivoltios. Para pasar de Voltios a Milivoltios se multiplica por 1.000 y para pasar de Milivoltios a Voltios se divide por 1.000

## Conclusiones

Si quieres sacar todo el rendimiento a un Arduino UNO, debes entender cómo funciona y cuáles son sus partes fundamentales.

A nivel de hardware ya lo hemos visto, son 7 claves con las que podrás sacar el máximo provecho a tus proyectos.