# Capítulo 3 – Arduino

Flata intro

## **3.1 ¿Qué es Arduino?**

Arduino es una plataforma y compañía, del mismo nombre, de electrónica "open-source" o de código abierto cuyos principios son contar con software y hardware fáciles de usar y que cualquiera pueda fabricar y mejorrar. Es decir, se propone como una plataforma sencilla de aprender para realizar proyectos interactivos para público no necesariamente con conocmientos técnicos.

Ilustración 8 - Logo Arduino

Arduino se trata de una SBC con entradas y salidas, analógicas y digitales, la cual es programda bajo un entorno de desarrollo, inspirado en el entorno de programación **Processing**. En la imagen (Ilustración 8 - Logo Arduino) se puede ver el logo oficial de la compañía.

## **3.2 ¿Qué es Processing?**

**Processing**es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. En la imagen (Ilustración 9 - Logo de Processing) se puede apreciar su logo.

Uno de los objetivos expresos de Processing es el de actuar como herramienta para que artistas, diseñadores visuales y miembros de otras comunidades ajenos a la programación, aprendan las bases de la misma a través de una realimentación gráfica inmediata y visual de los resultados obtenidos de su experiencia de programación.

Ilustración 9 - Logo de Processing

El lenguaje de Processing se basa en Java, aunque hace uso de una sintaxis simplificada y de un una biblioteca sencilla para generación de gráficos.

## FALTA EJEMPLO DE PROCESSING PARA CONECTAR CONCEPTOS CON WIRING

## Wiring

Wiring es un framework basado en Processing, enfocado en la programación de microcontroladores en vez de programación gráfica. Proveé una librería de C/C++ la cual simplifica operaciones comunes como el manejo de entrada/salida. Los programas de Wiring están escritos en C/C++, pese a que sus usuarios sólo necesiten definir dos funciones para hacer un programa ejecutable:

setup() – una función ejecutada sólo una vez en el arranque de la placa, la cual puede ser usada para definir los ajustes iniciales de un entorno.

loop() – una función llamada repetidamente hasta que la placa es apagada.

Como podemos apreciar en la siguiente ilustración (Ilustración 10 - Wiring IDE) hacer un blink a un led es muy sencillo dado la abstracción que nos otorga la librería. Un blink es un parpadeo de un led conectado a la placa. Se lo considera el “hola mundo” de Arduino.

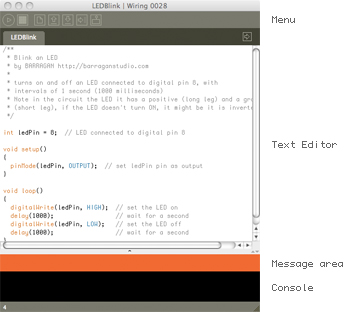


Ilustración 10 - Wiring IDE

Por otro lado, trabajando con C++ sin librerías quedaría como en la siguiente figura (Ilustración 11 - C++ Blink ejemplo) haciendo el mismo blink, programando directamente con la biblioteca AVR (control de los puertos de entrada y salida) y en C.[[1]](#endnote-1)



Ilustración 11 - C++ Blink ejemplo

## **3.3** **Fritzing**

El entorno Fritzing ayuda a los diseñadores y artistas a documentar sus prototipos interactivos y dar paso en la creación de prototipos físicos al producto real. Como podemos apreciar en la siguiente ilustración (Ilustración 12 - Entorno Fritzing), permite arrastrar componentes y generar un sketch.

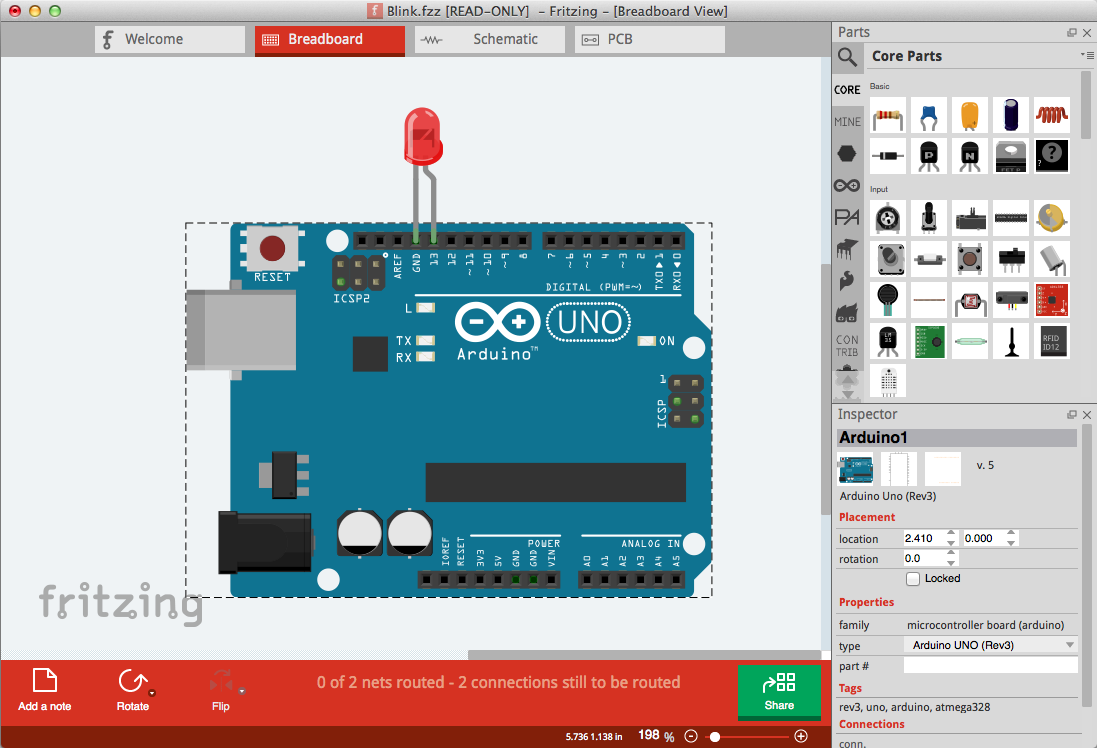


Ilustración 12 - Entorno Fritzing

## **3.4 ¿Qué es Wiring?**

Wiring es una plataforma de prototipado electrónico de fuente abierta compuesta de un lenguaje de programación, un entorno de desarrollo integrado (IDE), y un microcontrolador.

Esta plataforma permite escribir software para controlar dispositivos conectados a la tarjeta electrónica para crear toda clase de objetos interactivos, espacios o experiencias físicas que sienten y responden al mundo físico.

Este proceso se llama *sketching* con hardware; se explora una gran cantidad de ideas de forma muy rápida, se seleccionan las más interesantes, se afinan y producen prototipos en un proceso iterativo.

## **3.5 Entonces Arduino es…**

* Una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.
* Una plataforma de hardware abierto que facilita la programación de un microcontrolador. Los microcontroladores nos rodean en nuestra vida diaria, usan los sensores para escuchar el mundo físico y los actuadores para interactuar con el mismo. Los microcontroladores leen sobre los sensores y escriben sobre los actuadores.

## **3.6 Plataforma Arduino**

La plataforma consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida los cuales pueden conectarse a placas de expansión (*shields*), que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.

Los puertos serie que cuentan las distintas plataformas Arduino presentan puertos de serie como UART. La UART (universally asynchronous receiver/transmitter) es una unidad que incorporan ciertos procesadores, encargada de realizar la conversión de los datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada.

Por otro lado, también opera en nivel TTL (transistor-transistor logic). Esto significa que la comunicación se realiza mediante variaciones en la señal entre 0V y Vcc (donde Vcc suele ser 3.3V o 5V). Por el contrario, otros sistemas de transmisión emplean variaciones de voltaje de -Vcc a +Vcc (por ejemplo, los puertos RS-232 típicamente varían entre -13V a 13V).

Como podemos observar en la siguiente ilustración (Ilustración 13 - Ejemplo serie), se realiza una comunicación serie a (9600 bps) imprimiendo un contador. La zona marcada con rojo, es un botón que al presionarlo nos permite acceder a la terminal y ver el flujo serie seteando el clock correspondiente.

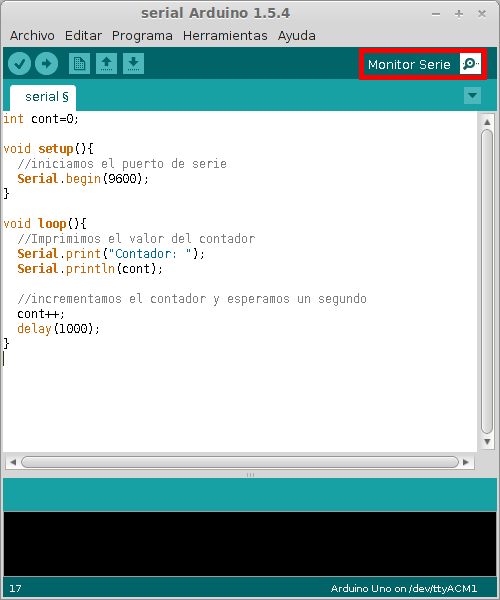


Ilustración 13 - Ejemplo serie

## **3.7 Distintas plataformas para Arduino**

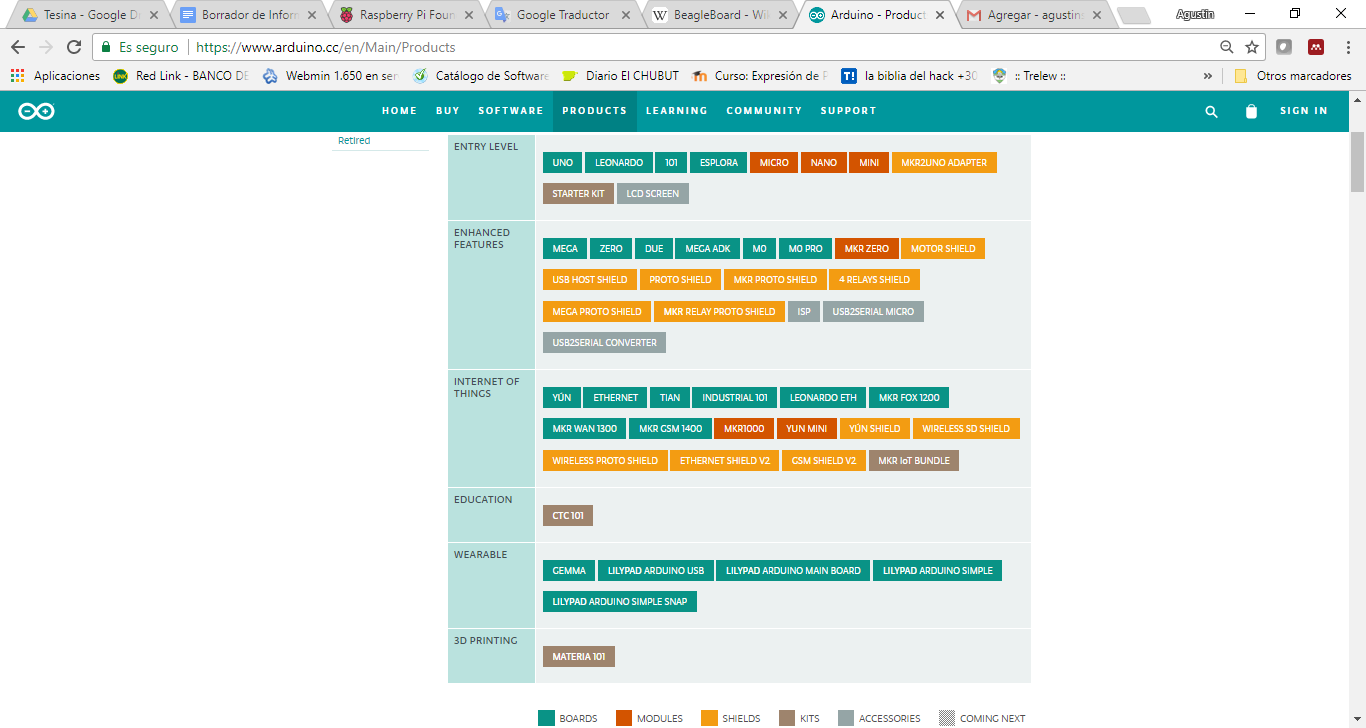


Ilustración 14 - Niveles de entrada a la plataforma Arduino

Existe una gran variedad de productos Arduino, la compañía los cataloga, como se puede aprecias en la imagen anterior (Ilustración 14 - Niveles de entrada a la plataforma Arduino), en distintos niveles según su utilidad[[2]](#endnote-2):

* Nivel de entrada: Son los más sencillos de utilizar, ideales para comenzar con la plataforma Arduino y realizar proyectos sencillos.
* Características mejoradas: Estas plataformas poseen características superiores, con respecto a las del nivel de entrada, están pensadas para proyectos más avanzados o de respuesta más rápida.
* Internet de las cosas: Estas placas vienen incorporadas con componentes que permitan realizar trabajos relacionados con la IoT (Internet de las cosas)[[3]](#footnote-1) mediante la incorporación de hardware de conectividad.
* Educación: En este caso, Arduino, ofrece un kit con herramientas y más de 25 proyectos, orientados a la educación, para realizar con sus plataformas.
* Usables: Estas plataformas están pensadas para “agregarle algo de electrónica” a prendas de vestir.
* Impresión 3D: Arduino ofrece una impresora 3D nombrada como Materia 101.

El hardware Arduino más sencillo consiste en una placa con un microcontrolador y una serie de puertos de entrada y salida. Los microcontroladores de 8 bits de AVR más utilizados en estas placas son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste, aunque también se dispone de microcontroladores ARM, cómo el caso del CortexM3 de 32 bits. A pesar de que ARM y AVR son plataformas diferentes, al utilizar la IDE de Arduino, los programas se compilan y luego se ejecutan sin cambios en cualquiera de las plataformas. En la imagen (Ilustración 15 - Arduino Uno) se visualiza la distribución física de puertos y componentes de la versión Arduino Uno R3.

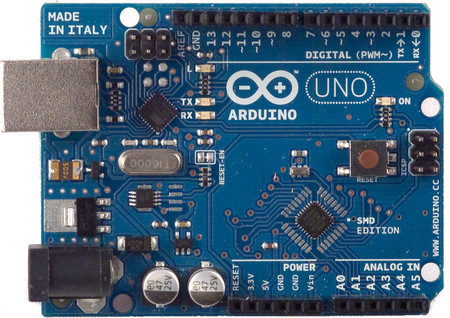


Ilustración 15 - Arduino Uno

## **3.8 Diferencias entre distintas placas de la familia Arduino**

Una primera diferenciación entre los distintos modelos de Arduino la encontraremos en el voltaje o tensión de alimentación de las placas. Las basadas en CortexM3 operan con un voltaje de 3,3 voltios, mientras que la mayor parte de las placas basadas en AVR utilizan una tensión de 5 voltios. Esto de todas formas no es un factor decisivo en la elección de una placa, dado que existen conmutadores de tensión en muchos actuadores y sensores compatibles.

Los usos posibles que se le pueden dar a un Arduino, en forma general son:

* Aquellos en los que el Arduino es utilizado como microcontrolador, tiene un programa descargado desde un ordenador y funciona de forma independiente de éste, y controla y alimenta determinados dispositivos y toma decisiones de acuerdo al programa descargado e interactúa con el mundo físico gracias a sensores y actuadores.
* La placa Arduino hace de interfaz entre un ordenador (como podría ser una Raspberry Pi) u otro dispositivo, que ejecuta una determinada tarea, para traducir dicha tarea en el mundo físico a una acción (actuadores).

## **3.9 ¿Por qué usar Arduino?**

### 3.9.1 La comunidad

La comunidad Arduino se desarrolla y enriquece a partir del trabajo con la placa, de la experimentación, de la producción de conocimiento en torno a ella, y estas habilidades se comparten dentro de la comunidad, pudiendo cualquier persona tener acceso a ellas.

La página principal de Arduino da soporte por medio de secciones y dedica un blog histórico con novedades y proyectos que se encuentran en desarrollo.

Además, se crean sitios como Arduino playground (<http://playground.arduino.cc/>) que es una wiki donde todos los usuarios de Arduino pueden contribuir. Es el lugar donde publicar y compartir código, diagramas de circuitos, guías, manuales, cursos. Es la base de datos de conocimiento de la comunidad de Arduino. Este sitio a su vez tiene soporte a distintos lenguajes como el español (<https://playground.arduino.cc/Es/Es>).

Por otro lado, existe Arduino Hub, un lugar donde se comparten los proyectos, dando los distintos pasos para reproducirlo.

El manifiesto de la comunidad Arduino dice (traducción al español):

“Apoyar al ecosistema de hardware y software de open source Arduino, haciendo que los productos electrónicos sean abiertos y participativos.

Servir como un evangelista para Arduino, expandir el ecosistema de código abierto a estudiantes, fabricantes, desarrolladores, diseñadores, ingenieros y empresas dentro de sus comunidades locales.

Construir una red global de comunidades que diseñen y codifiquen proyectos, intercambien ideas, organicen actividades de colaboración y dicten cursos oficiales de Arduino, independientemente de su edad, sexo, idioma y capacidad técnica”[[4]](#footnote-2)

La siguiente imagen (Ilustración 16 - Logotipo comunidad open-source de Arduino) muestra el logotipo oficial de la comunidad open-source de Arduino.



Ilustración 16 - Logotipo comunidad open-source de Arduino

### 3.9.2 La sencillez del lenguaje de programación

Programar la placa es muy sencillo y accesible, y la ayuda por parte de la comunidad lo hace aún más fácil. Como se mostró en las ilustraciones (Ilustraciones 10 y 11), podemos apreciar librerías con un alto nivel de abstracción en cuanto al acceso a bajo nivel.

La definición de puertos con la sentencia a alto nivel, seteando los puertos del 1 al 7 en un Arduino UNO como salida sería:

PinMode(1,OUTPUT) ;

PinMode(2,OUTPUT) ;

…;

PinMode(7,OUTPUT) ;

en contraposición con:

DDRD = B11111110; [[5]](#footnote-3) // sets Arduino pins 1 to 7 as outputs, pin 0 as input

### 3.9.3 Es hardware de bajo costo

Lo único que “vale” en la placa son sus componentes, ya que no debemos pagar el costo de la licencia de su creador, por el hecho de ser hardware libre.

## **3.10 Incorporación de Arduino en las escuelas**

Gracias a la plataforma Arduino y su comunidad, se permite una vinculación con la dinámica de las escuelas a través de la creación de una red de trabajo colaborativo. Esto conlleva a:

* La utilización de medios multimediales, para conformar cursos y capacitaciones.
* Creación de proyectos articulares entre distintos espacios curriculares
* Uso de diversas herramientas que son Arduino-compatibles y gratuitas.
* Costos relativamente bajos en la adquisición de las distintas plataformas.

Todos estos puntos anteriores favorecen la introducción en el aula y clubes.

Por otro parte, la sencillez del lenguaje de programación de la placa, permite la rápida utilización por alumnos y docentes, no necesariamente del ámbito de la informática y la electrónica. Esto contribuye a la construcción colectiva del conocimiento, promoviendo la interdisciplinariedad escolar, donde docentes de distintas áreas articulan para crear proyectos.

Utilizando clubes de ciencia o proyectos específicos permite que el trabajo se apoye sobre un modelo pedagógico de aprendizaje en proceso, donde el sujeto que aprende es participante activo de ese proceso, desde la concepción de la idea hasta el producto final, incorporando conocimientos técnicos específicos.

Este tipo de actividades educativas hacen que la tecnología y su uso se pongan al servicio de la creatividad, el juego, la experimentación y la invención, con la posibilidad de ser adaptado al contexto en el que se inserta. Además, proporcionar la recuperación de la tecnología obsoleta existente en ellas como se describe en la siguiente sección.

### 3.10.1 Las tres erres

Las tres erres (reducir, reutilizar, reciclar) es una regla para cuidar el medio ambiente, específicamente para reducir el volumen de residuos o basura generada.

Cuando hablamos de reducir lo que estamos diciendo es que se debe tratar de simplificar el consumo de los productos directos.

Al decir **reutilizar**, nos estamos refiriendo a poder volver a utilizar los objetos y darles la mayor utilidad posible antes de que llegue el momento de desecharlos.

Por otro lado, **reciclar** consiste en el proceso de someter los materiales a una transformación en el cual se puedan volver a utilizar.

Esta definición se pretende aplicar en las escuelas haciendo un proceso de clasificación, selección y desoldando componentes electrónicos de placas en desuso y materiales que se han desechado en las instituciones o en hogares de los alumnos.

## **3.11 Actuadores y sensores**

Un **actuador** es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de una acción con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control, como por ejemplo un LED.

Un **sensor** es un objeto capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, pH, etc.

En conjunto, los sensores y actuadores, permiten la creación de distintos tipos de artefactos, que posibilitan comunicarse con el ambiente que los rodea, modificándolo (actuadores) o recibir estímulos (sensores).

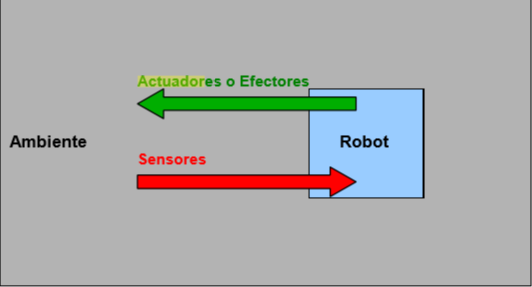


Ilustración 17- Representación actuadores y sensores

En esta imagen (Ilustración 17- Representación actuadores y sensores) se representa la toma de datos un robot puede capturar del ambiente que lo rodea mediante sus diversos sensores, y a su vez como podría interactuar con el mismo mediante actuadores.

## **3.12 Actuadores en el SAR**

La electrónica industrial ha generado estandarización en el campo de los sensores y actuadores, muchos de estos últimos con buen soporte en Arduino. Precisamente en el SAR se utilizarán:

* Motores de corriente continua
  + Para el desplazamiento del robot móvil
* LED
  + Para indicar estados del RM

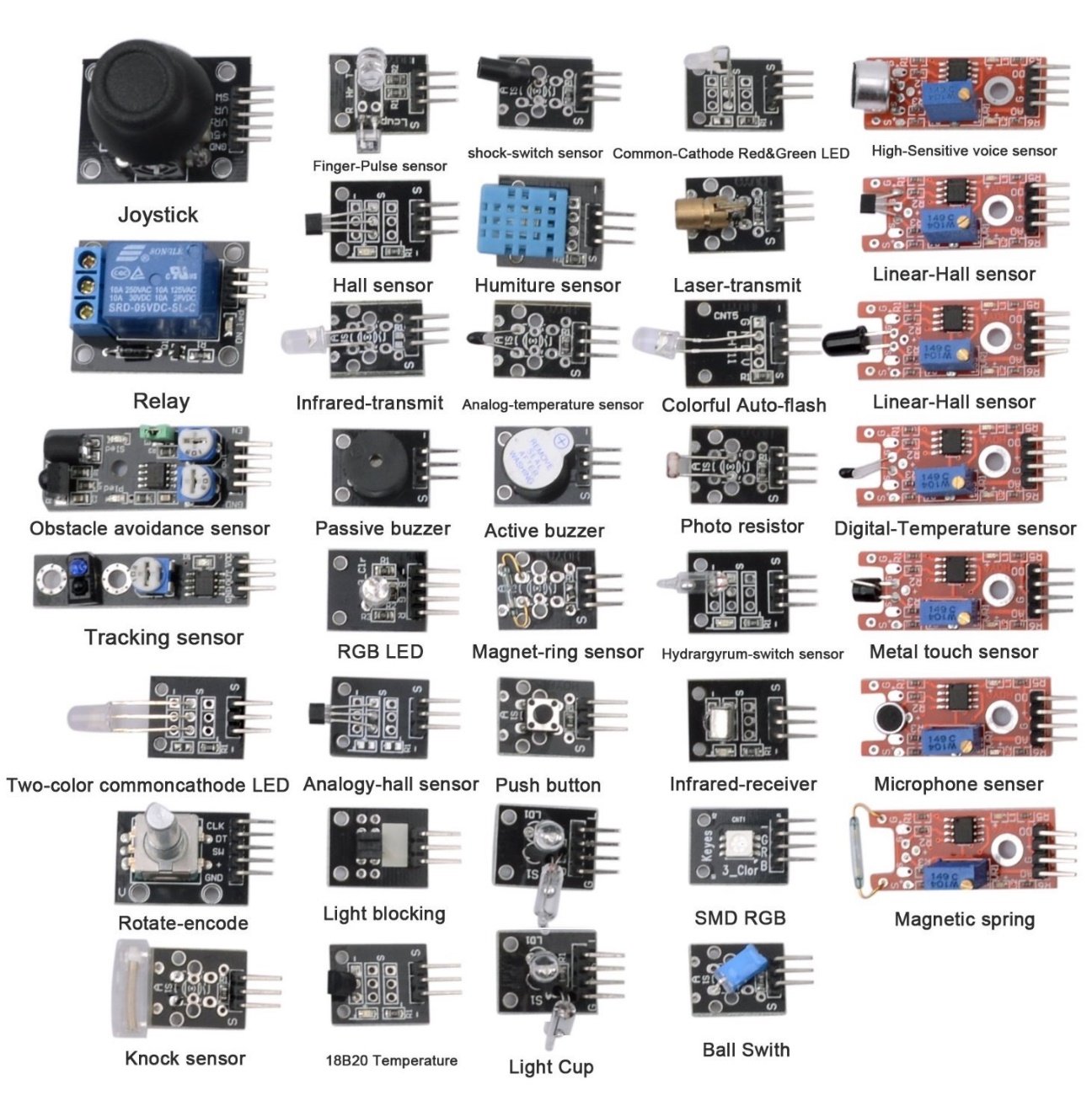


Ilustración 18 - Actuadores y sensores compatibles con Arduino

## **3.13 Sensores en el SAR**

El SAR utiliza los siguientes sensores:

* Sensor ultrasónico HC-SR04
  + Para detectar objetos, y distancia entre el RM y elementos del ambiente
* Sensor de Temperatura KY-001
  + Incorporado para analizar la temperatura del ambiente
* Sensor de presencia de gases MQ-7
  + Detección de monóxido de carbono

Algunos de los sensores y actuadores se pueden apreciar en la ilustración anterior (Ilustración 18 - Actuadores y sensores compatibles con Arduino).

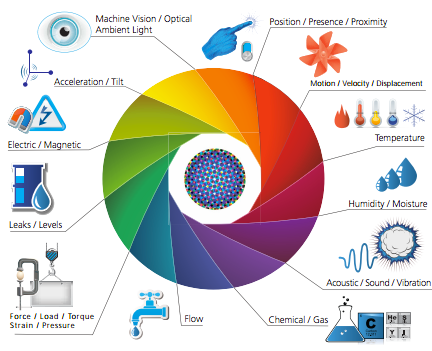


Ilustración 19- Representación de sensores

En esta imagen (Ilustración 19- Representación de sensores) se pueden apreciar los distintos factores de un entorno que pueden ser evaluadores con sensores mencionados anteriormente.

## **3.14 Módulos o *shields* en el SAR**

El SAR utiliza[[6]](#footnote-4):

* MotorShield L298
  + Para administración del puente H y gestión de los motores de CC
* Módulo bluetooth HC-05
  + Para la comunicación con dispositivos compatibles (móviles y/o computadoras)
  + Envío de órdenes
* Módulo GPS NEO-6
  + Para la geolocalización del RM
* Módulo ESP8266
  + Conectividad y transferencia de datos vía WIFI
  + Activación del Ad-hoc

A lo largo del desarrollo de la tesina se fueron implementando diversos casos de pruebas sobre los sensores, actuadores y módulos especificados en esta sección. Las pruebas se encuentran anexas en este documento.

1. http://comoprogramarpic.blogspot.com.ar/2012/06/programando-un-atmel-mi-primer-programa.html [↑](#endnote-ref-1)
2. <https://www.arduino.cc/en/Main/Products> [↑](#endnote-ref-2)
3. Internet de las cosas: “Concepto que se refiere a la interconexión de objetos cotidianos con Internet. “ <https://es.wikipedia.org/wiki/Internet_de_las_cosas> [↑](#footnote-ref-1)
4. <https://www.arduino.cc/en/aug/> [↑](#footnote-ref-2)
5. <https://www.arduino.cc/en/Reference/PortManipulation> [↑](#footnote-ref-3)
6. El uso de algunos de estos módulos queda en forma tentativa, dado que existen también en la Raspberry y su uso puede ser complementario. [↑](#footnote-ref-4)