



# **Universidad Nacional de la Patagonia San Juan Bosco**

## **Facultad de Ingeniería**

### **Tesina de grado:**

Desarrollo y construcción de un sistema autónomo robótico  
administrado por una aplicación web para exploración

#### **Alumnos:**

**Mansilla Fernando Damián**

**Schlapp Agustín Pablo**

#### **Tutor:**

**Lic. Defossé Nahuel**

**Trelew**  
**Año 2017**

## Índice

Capítulo 1 - Introducción .....	5
1.1 Objetivo general.....	5
1.1.1 Objetivos específicos.....	5
1.1.2 Metodología.....	5
1.2 Motivación .....	6
1.3 Desarrollos Propuestos .....	7
1.4 Resultados Esperados .....	7
Referencias Bibliográficas .....	<b>¡Error! Marcador no definido.</b>
Capítulo 2 - La robótica .....	8
2.1 ¿Qué es la robótica? .....	8
2.2 Estructura física de los robots .....	9
2.2.1 Poliarticulados.....	9
2.2.2 Móviles.....	10
2.2.3 Androides.....	10
2.2.4 Zoomórficos .....	10
2.2.5 Híbridos.....	11
2.3 Distintas tecnologías para la robótica educativa .....	11
2.4 Microcontroladores y computadora de placa reducida (SBC) .....	11
2.5. Comunicación entre distintas arquitecturas de cómputo .....	13
2.6 ¿Qué es un SAR (Sistema Autónomo Robótico)? .....	14
2.7 La robótica en la educación .....	14
Capítulo 3 – Arduino .....	16
3.1 ¿Qué es Arduino?.....	16
3.2 ¿Qué es Processing? .....	16
3.3 ¿Qué es Wiring? .....	17
3.4 Entonces Arduino es.....	17
3.4 Hardware Arduino.....	17
3.5 Distintas plataformas para Arduino .....	18
3.6 Universo Arduino .....	18
3.6.1 Placas .....	18
3.6.2 Placas de expansión (shields) .....	18
3.6.3 Kits .....	18
3.6.4 Accesorios .....	19
3.6.5 Impresoras 3d .....	19
3.7 Diferencias entre distintas placas de la familia Arduino .....	19
3.8 ¿Por qué usar Arduino?.....	20

3.8.1 La comunidad formada a su alrededor y la generación de habilidades compartidas .....	20
3.8.2 La sencillez del lenguaje de programación .....	20
3.8.3 Es hardware de bajo costo .....	20
3.8.4 Incorporación de Arduino en las escuelas .....	20
3.9 Actuadores y sensores .....	21
3.10 Actuadores en el SAR .....	21
3.11 Sensores en el SAR .....	22
3.12 Módulos o shields en el SAR .....	23
Bibliografía.....	<b>¡Error! Marcador no definido.</b>
Capítulo 4 – Raspberry Pi .....	24
4.1 ¿Qué es Raspberry Pi? .....	24
4.2 Especificaciones técnicas de las distintas versiones.....	24
4.3 Sistemas Operativos compatibles .....	25
4.4 Accesorios para Raspberry Pi .....	25
4.5 ¿Por qué elegir Raspberry Pi? .....	27
Capítulo 5 - Aplicaciones Móviles .....	29
5.1 ¿Qué son las aplicaciones móviles? .....	29
5.2 Las App's .....	29
5.2.1 Las webApps .....	29
5.2.2 Ventajas de las Web-App: .....	30
5.2.3 Desventajas de las Web-Apps .....	30
5.3 ¿Cuáles son los distintos sistemas operativos para dispositivos móviles? .....	30
5.4 Android .....	31
5.5 Aplicaciones móviles multiplataforma .....	32
5.5.1 Diferencias entre aplicaciones y web móviles.....	32
5.5.2 Aplicaciones Nativas.....	32
5.5.3 Aplicaciones Web .....	33
5.5.4 Aplicaciones Híbridas .....	34
5.5.5 Aplicación híbrida: Web App .....	34
5.5.6 Aplicación híbrida: app interpretada.....	34
5.6 Investigación en Desarrollo de Apps y WebApps .....	35
5.6.1 Android Studio .....	35
5.6.2 App inventor .....	35
5.6.3 Intel XDK.....	36
5.6.4 Ionic .....	36
5.6.5 Cordova .....	36
5.6.6 Meteor .....	36

5.6.7 Meteor y Cordova .....	36
Capitulo 6 – Stack MEAN .....	38
Capítulo 7 – Librería Johnny-five y el protocolo Firmata .....	41
7.1 ¿Qué es Johnny-five? .....	41
7.2 Instalación .....	41
7.3 Arduino Firmata .....	42
7.4 Instalación Firmata .....	42

# Capítulo 1 - Introducción

## 1.1 Objetivo general

Se pretende desarrollar un prototipo de un Sistema Autónomo Robótico (SAR), gestionado por un software definido como agente inteligente (que responda al modelo basado en objetivos<sup>1</sup>) para la exploración y análisis del medio ambiente.

### 1.1.1 Objetivos específicos

- Desarrollar una aplicación web multiplataforma que mediante comunicación inalámbrica permita el control del Robot móvil.
- Ensamblar un robot móvil integrando las plataformas Arduino y Raspberry Pi con diversos módulos y software.
- Investigar protocolos existentes y evaluar la necesidad de diseño de protocolos de comunicación para el control y procesamiento de datos entre el microcontrolador y la aplicación.
- Ensamblar físicamente e integrar a nivel de software los distintos componentes (sensores y actuadores) al SAR.
- Extender la aplicación para interactuar con la información que brinda el SAR de los sensores.

### 1.1.2 Metodología

El SAR se creará mediante las plataformas Arduino y Raspberry Pi. El robot poseerá motores como actuadores para desplazarse sobre la superficie a explorar y diversos sensores que permitan tomar muestras del ambiente explorado. Todos estos componentes se ensamblarán sobre distintas piezas estructurales para conformar el robot móvil o RM.

El RM estará en un estado receptivo, donde se le otorga el control a una aplicación web, la cual contará con una interfaz de usuario que facilitará la comunicación con el SAR. La aplicación permitirá manipular el desplazamiento del RM sobre la superficie y obtener las muestras del ambiente según se soliciten, en otras palabras, la lectura de los sensores.

---

<sup>1</sup> Agente basado en objetivos: “Almacena información del estado del mundo, así como del conjunto de objetivos que intenta alcanzar, y que es capaz de seleccionar la acción que eventualmente lo guiará hacia la consecución de sus objetivos” [Inteligencia Artificial un enfoque moderno. Person. Stuart Russell, Peter Norving 2da Ed. Pág. 57]

La comunicación entre el SAR y la aplicación se realizará por medio de señales inalámbricas de radiofrecuencia. Se mantendrá una arquitectura de diseño denominada cliente/servidor, donde el cliente es el dispositivo que ejecuta la aplicación y el servidor es el SAR.

## 1.2 Motivación

Las nuevas tendencias de hardware como microcontroladores, Smartphones y nuevos dispositivos programables, requieren contar con un nuevo esquema de diseño donde se puedan integrar las distintas tecnologías relacionadas (robótica, redes, plataformas móviles, etc.) en un área de conocimiento específica, para lograr una integración de saberes y disminuir la curva de aprendizaje de personas que se introducen en estas temáticas.

Para esto se necesita incursionar en el investigación y desarrollo en los ámbitos de la computación, control, mecánica y electrónica. Los cuales dieron paso a la robótica como técnica que combina diversas disciplinas, logrando un alto impacto en la sociedad en diversos ámbitos.

En la actualidad es muy popular la utilización de teléfonos móviles inteligentes (*smartphones*). De estos dispositivos, un segmento mayoritario se basa en el sistema operativo Android, presentado por Google en el 2007.

Android está basado en Linux y utiliza Java como lenguaje de desarrollo de aplicaciones. Por otro lado, Arduino, introducido en el año 2005, es una plataforma de hardware libre para electrónica orientado a la computación física (Physical Computing).

Arduino aprovecha ciertas características de C++ para permitir el desarrollo de pequeños programas o sketches con conocimientos básicos de programación y electrónica. Esta simplicidad, sumado al bajo coste de las placas ha otorgado a la plataforma una gran popularidad.

Tanto Java como C++ han sido lenguajes utilizados en las actividades de laboratorio de varias cátedras de la Licenciatura por lo cual consiste en una motivación para llevar a cabo esta tesina.

Los nuevos avances en interoperabilidad de las distintas plataformas de las áreas de robótica y programación tanto en hardware como software, brindan un excelente recurso en materia de educación de nivel medio y superior permitiendo agilidad en el desarrollo de proyectos educativos con escaso conocimiento en dichas áreas. Es por ello que se necesita un estándar o prototipo de dónde partir, que se encuentre testeado con una biblioteca de funciones inmersas en el mismo y una arquitectura

moldeable a distintas temáticas. Este prototipo base es el denominado SAR que se quiere desarrollar. En síntesis, el objetivo del SAR es crear un instrumento didáctico para la comprensión e incentivación de los alumnos en las distintas áreas mencionadas (robótica e informática).

### 1.3 Desarrollos Propuestos

- Diseño y desarrollo del software necesario para el funcionamiento del SAR.
- Ensamblado de un prototipo hardware basado en Arduino y Raspberry Pi, integrado por distintos módulos compatibles con dichas plataformas.
- Diseño y desarrollo de una aplicación web que permita controlar el RM cuya interfaz integre la visualización de valores recolectados por los sensores integrados al SAR y generación de estadísticas a partir de estos datos.
- Selección de un medio de comunicación inalámbrica (Radiofrecuencia) que permita la interrelación entre la aplicación móvil y el SAR.

### 1.4 Resultados Esperados

Al finalizar la tesina esperamos haber construido el robot móvil a partir de la integración de las diversas plataformas previamente mencionadas, conformando el denominado SAR.

Se espera aportar conocimiento significativo para futuros proyectos que requieran la utilización de protocolos de comunicación inalámbricos entre aplicaciones móviles y microcontroladores.

Tanto el desarrollo del software como el hardware serán liberados para contribuir a un mejor proceso de enseñanza de la informática y robótica en principio en el nivel medio.

Un resultado esperable es que el SAR en su conjunto sea fácilmente extensible y por lo tanto se prevé que otros continúen la evolución del producto y sea utilizado como base para nuevos proyectos relacionados con la robótica y aplicaciones móviles.

Otro resultado esperado es que los anexos referentes a la utilización de módulos sean de utilidad para la enseñanza de electrónica en nivel medio.

## Capítulo 2 - La robótica

### 2.1 ¿Qué es la robótica?

A lo largo de la historia el ser humano ha sentido fascinación por las máquinas que puedan imitar las figuras y movimientos de seres animados. El poder desarrollar sistemas electromecánicos que simulen o realicen actividades típicas de seres vivos, ofrece la sensación de tener un propósito propio, lo cual fue un motivador para su estudio.

A este tipo de maquinaria se la denomina Robot. Según la RIA (Robotic Industries Association):

*“Un robot es un manipulador funcional reprogramable, capaz de mover material, piezas, herramientas o dispositivos especializados mediante movimientos variables programados, con el fin de realizar tareas diversas.”*

Una de las grandes diferencias entre los robots y el resto de las máquinas es la versatilidad que adquieren los mismos al poder variar su propósito modificando su programación. Todas las tareas que realizan los robots están basadas en la manipulación de su entorno.

Se le considera robótica a la ciencia y técnica encargada del diseño, construcción y aplicación de robots. Esta ciencia involucra diversas disciplinas tales como la mecatrónica, electrónica, mecánica, e informática, entre otras.

Actualmente la robótica ha ido evolucionando rápidamente, dando lugar a innovaciones tecnológicas destacadas para la historia de la humanidad, logrando un alto impacto socio-económico. Hoy en día, la robótica no es solo utilizada en los ámbitos industriales o militares, sino que podemos ver a robots en variadas áreas como por ejemplo en la medicina o en la educación.

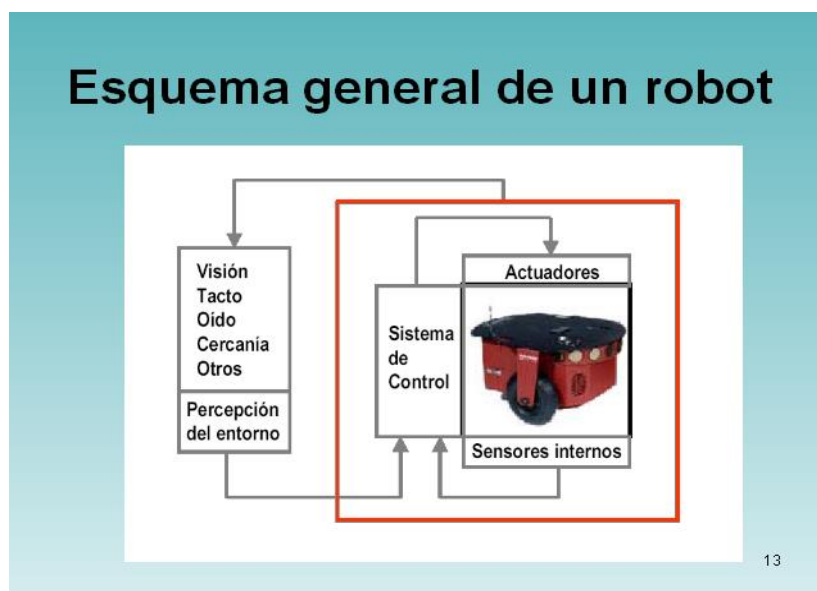


Ilustración 1 - Esquema básico de un robot



La robótica está constituida por tres grandes temas como lo son; la *percepción*, la *planificación* y la *manipulación*. En conjunto permiten el desarrollo de robots con un gran índice de autonomía, logrando acciones básicas que realiza un ser humano al ejecutar ciertas tareas. Cuando una persona ha decidido que tiene una necesidad, los primeros que hace es estudiar su entorno con alguno de sus cinco sentidos (*percepción*); luego toma la decisión de realizar acciones con determinados movimientos (*planificación*) para que, finalmente, las ejecute de modo secuencial (*manipulación*).

Las funciones a realizar por cada uno de los temas podrían ser las siguientes:

Percepción:

- Sensores
- Tratamiento de información
- Procesamiento de información

Planificación:

- Trayectorias
- Tareas
- Planificación de tareas
- Toma de decisiones

Manipulación:

- Mecánica
- Actuadores
- Sistema de control
- Sistema de programación

## 2.2 Estructura física de los robots

La estructura es definida por el tipo de configuración general de las distintas piezas que conforman el Robot. Es difícil establecer una clasificación coherente de los mismos que resista un análisis crítico y riguroso. La subdivisión de los Robots, con base en su arquitectura, se podría hacer dentro de alguno de los siguientes grupos: poliarticulados, móviles, andróides, zoomórficos e híbridos.

### 2.2.1 Poliarticulados

Se les denomina robots poliarticulados a aquellos que en su mayoría son sedentarios o de desplazamientos muy limitados y tanto su forma como configuración pudiera ser muy diversa. En este grupo entrarían aquellos robots estructurados para mover sus componentes terminales (Ej.: sus actuadores) en un espacio determinado de trabajo con una simetría específica. Ejemplos, podrían ser los robots industriales, cartesianos y/o manipuladores.



Ilustración 2 - Ejemplo de robot poliarticulado

### 2.2.2 Móviles

Estos robots se caracterizan, primordialmente, por su capacidad de desplazamiento. Su forma, por lo general, se basa en diseños típicos de vehículos como los automóviles. Su objetivo prioritario suele ser recorrer un determinado camino guiándose por la información de su entorno, obtenida a través de sus sensores. Pueden ser dotados de un cierto nivel de inteligencia (gracias a su programación) e incluso sortear obstáculos.



*Ilustración 3 - Ejemplo de robot móvil*

### 2.2.3 Androides



*Ilustración 4 - Androide Asimo de Honda*

Se les llama androide a los robots que intentan simular y/o reproducir la forma y comportamiento cinemático de seres vivos. Todavía no cuentan con alguna aplicación práctica específica, sino más que, para el estudio y la experimentación.

### 2.2.4 Zoomórficos

Los Robots zoomórficos, se caracterizan principalmente por sus sistemas de locomoción que tienen como objetivo imitar a los diversos seres vivos. A pesar de la disparidad morfológica de sus posibles sistemas de locomoción se suelen distinguir entre dos categorías principales: caminadores y no caminadores. El grupo de los no caminadores está muy poco evolucionado. Los Robots zoomórficos caminadores multípedos son muy numerosos y están siendo objeto de experimentos en diversos laboratorios con vistas al desarrollo posterior de verdaderos vehículos terrestres, pilotados o autónomos, capaces de evolucionar en superficies muy accidentadas. Las aplicaciones de estos Robots apuntan a su utilización en el campo de la exploración espacial y en el estudio de los volcanes.



*Ilustración 5 - Robot Zoomórfico caminador*

### 2.2.5 Híbridos

Los robots híbridos se les considera a aquellos a los cuales es difícil clasificar dentro de las mencionadas anteriormente o bien es la combinación de algunas de ellas.



*Ilustración 6 - Robot móvil-poliarticulado*

## 2.3 Distintas tecnologías para la robótica educativa

Sin duda alguna, en los últimos años, las arquitecturas más destacadas para la enseñanza y desarrollo de robótica a nivel educativo han sido las plataformas **Arduino** y **Raspberry Pi**. Debido a su accesible costo y variedades de versiones a disponibilidad, estas tecnologías son utilizadas en las diversas disciplinas relacionadas con la robótica educativa. En el caso de Arduino, presenta una notable ventaja dentro de este ámbito dado que la compañía que los fabrica (del homónimo Arduino) de hardware libre la cual ofrece una amplia variedad de modelos para usos múltiples (se brindará más detalle sobre esta tecnología en el siguiente capítulo). Por otro lado, Raspberry Pi es un computador reducido creado con el objetivo de la enseñanza de la informática, cuenta con notables capacidades de procesamiento en relación a su bajo costo.

La gran ventaja de estas arquitecturas con respecto a las que se mencionan a continuación, es su gran soporte y compatibilidad, dada la amplia comunidad que las utiliza.

Como se dijo, existen otras tecnologías para el desarrollo de la robótica tales como; la plataforma **Intel galileo**, similar a Raspberry Pi pero desarrollada por Intel, es también un computador reducido certificado por Arduino que integra la arquitectura Intel X86; **BeagleBone**, es una placa computadora de hardware libre diseñada como plataforma de evaluación y de prototipos para ingenieros profesionales; **Nanode**, es una placa de microcontrolador de código abierto, similar a Arduino, pero con capacidad de conectividad a internet con el objetivo de la experimentación en IoT (Internet de las cosas).

## 2.4 Microcontroladores y computadora de placa reducida (SBC)

Un **microcontrolador** es un circuito integrado programable, por lo general en una PCB (placa de circuito impreso), con la capacidad de ejecutar órdenes cargadas en su memoria. Su velocidad de procesamiento es limitada, dado que su objetivo es el de funcionar como controlador. Sus usos son comunes en electrodomésticos, que requieren de un control sencillo.

Puede ser muy común pensar que un microcontrolador es igual a un microprocesador, pero esto no es así, de hecho difieren en muchos aspectos. La principal diferencia es su funcionalidad, dado que, para utilizar un

microprocesador en alguna aplicación real, se debe conectar con diversos componentes tales como memorias o buses de transmisión de datos.

Aunque el microprocesador se considera una máquina de computación poderosa, no está preparado para la comunicación con los dispositivos periféricos que se le conectan. Para que el microprocesador se comunique con algún periférico, se deben utilizar los circuitos especiales. Así era en el principio y esta práctica sigue vigente en la actualidad.

Por otro lado, al microcontrolador se diseña de tal manera que tenga todos los componentes integrados en el mismo chip. No necesita de otros componentes especializados para su aplicación, porque todos los circuitos necesarios, que de otra manera correspondan a los periféricos, ya se encuentran incorporados. Así se ahorra tiempo y espacio necesario para construir un dispositivo.

Es por estas razones que han tenido grandes repercusiones para el desarrollo de la robótica.

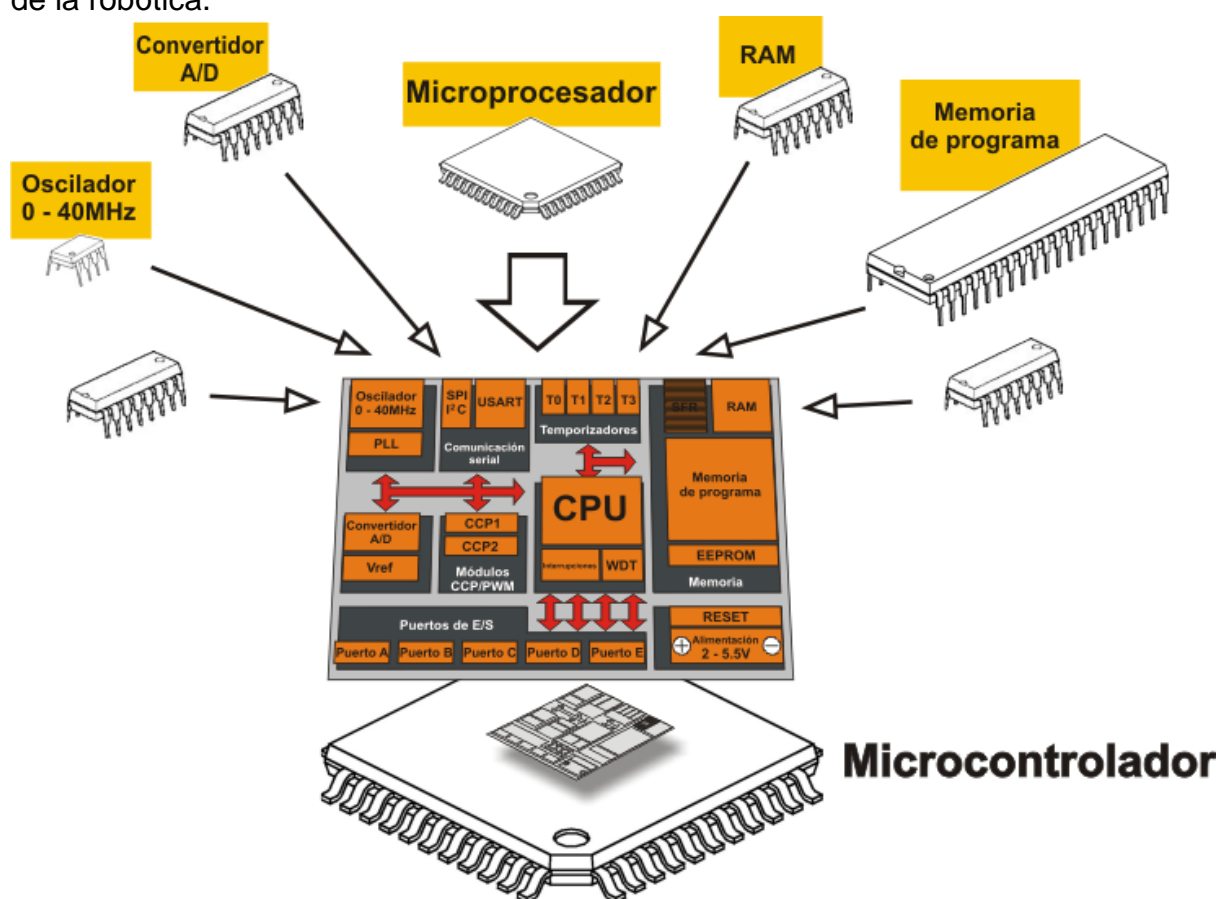


Ilustración 7 - Arquitectura de un microcontrolador

Una **computadora de placa reducida** (SBC, *Single Board Computer*), en cambio, es una computadora completa que integra todos los componentes necesarios, que definen a la misma, en un solo circuito (la placa madre o *motherboard*) con la particularidad de que la misma es de un tamaño mucho más reducido que el de una computadora común y corriente como las PCs. Ejemplos típicos de este tipo de computadoras son tanto las arquitecturas Arduino como Raspberry Pi.

En el caso de Arduino, dentro de su placa se integra un microcontrolador para el procesamiento de sus órdenes programadas, en cambio, Raspberry Pi

integra un microprocesador con capacidades de ejecutar un sistema operativo con interfaz gráfica.

## 2.5. Comunicación entre distintas arquitecturas de cómputo

Existen diversos medios de comunicación entre las PCs y las computadoras de placa reducida o de dispositivos de cómputo entre sí, a continuación, se listan algunos de ellos:

- Serial: La comunicación serie o serial es una interfaz de comunicación de datos digitales que nos permite establecer transferencia de información entre varios dispositivos. Un puerto es el nombre genérico con que denominamos a las interfaces, físicas o virtuales, que permite esta comunicación entre dispositivos. Un puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión). Las placas Arduino actuales cuenta con un puerto USB para realizar este tipo de comunicación y es su principal interfaz para conectarlos a una PC donde cargar la secuencia de órdenes que luego ejecutará.
- Inalámbricas: Los medios de comunicación inalámbricos, para computadoras, han evolucionado de forma exponencial desde su aparición. Su gran ventaja, como su nombre lo dice, es que no necesitan de un medio de propagación físico (como los cables) para la transmisión de los datos, sino que, para el envío de los mismo se utiliza la modulación de ondas electromagnéticas a través del espacio. Existen diversos tipos, con grandes diferencias en cuanto a velocidades y rangos de alcance. En cuanto para la robótica podemos encontrar dispositivos que nos permitan conectar computadoras de placas reducidas con diversos computadores por medio de:
  - *Radiofrecuencia*: Existen módulos compatibles con Arduino, como el módulo de radiofrecuencia RF 433Mhz, que nos permiten conectar dos plaquetas de este tipo entre sí de forma inalámbrica a través de radiofrecuencia.
  - *Infrarrojo*: Las redes de luz infrarroja están limitadas por el espacio, se utilizan por lo general en dispositivos que se encuentran en un mismo espacio físico como un cuarto o un piso. Como su nombre lo indica se utilizan leds de luz infrarroja para la transmisión y recepción de datos.
  - *Bluetooth*: Es una especificación industrial que permite crear redes inalámbricas de área personal (WPAN), mediante un enlace de radiofrecuencia que trabaja en la banda ISM (Industrial Scientific and Medical) de 2.4 GHz posibilitando la transmisión de voz y datos.
  - *Wifi*: Este mecanismo de comunicación inalámbrica es el más popular entre computadoras de hoy en día. A su vez, es una marca de la Alianza Wi-fi la cual certifica que los dispositivos



cumplan con los estándares IEEE 802.11 vigentes relacionados a redes inalámbricas de área local.

## 2.6 ¿Qué es un SAR (Sistema Autónomo Robótico)?

Se le considera SAR o sistema autónomo robótico a aquellos robots que presentan cierto grado de autonomía (Inteligencia artificial). Tienen la capacidad de testear su entorno (por medio de sensores) para decidir qué acciones realizar (por medio de actuadores). Por ende, podemos decir que, son sistemas dinámicos que consisten en un controlador electrónico acoplado a un cuerpo mecánico.

En el desarrollo propuesto por esta tesina, se diseñó y armó un sistema autónomo robótico móvil que posee su grado de inteligencia, pero a su vez, permite ser manipulado desde una aplicación web.

## 2.7 La robótica en la educación

En educación pueden diferenciarse dos tipos de uso de la programación y la robótica como apoyo en la clase: por un lado, la robótica y la programación educacional, que consiste en un conjunto de elementos físicos o de programación que motivan a los estudiantes a construir, programar, razonar de manera lógica y crear nuevas interfaces o dispositivos; por otro, la programación y la robótica como elemento social, por ejemplo a modo de juego o gamificación, de forma que sistemas autónomos o semiautónomos interactúan con humanos u otros agentes físicos o software en roles como entrenador, compañero, dispositivo tangible o registro de información.

El desarrollo de actividades educativas basadas en robots o en programación pueden incrementar el compromiso por el aprendizaje en otras áreas como literatura o historia a través del juego y la motivación. Aún más, su uso puede mejorar el desarrollo ético, emocional y social en base al impacto que, por ejemplo, un robot con atribuciones sociales puede causar en los niños y las niñas.

Otro beneficio prometedor, es su potencial educativo para niños y niñas con necesidades especiales tanto en las áreas cognitivas como psicosociales. La escalabilidad de las propuestas educativas basadas en robots, y su enorme potencial motivador, lo hacen especialmente útil en programas de refuerzo y de educación especial.

Una de las grandes controversias en estas áreas, es sobre los materiales que deben utilizarse en el aula. Algunos investigadores afirman que los dispositivos tangibles aumentan el nivel de inmersión porque los estudiantes están manipulando las cosas en un mundo real. Sin embargo, podemos encontrar otros estudios que entienden que los dispositivos no tangibles, como los elementos de programación, atraen más y evitan limitaciones a causa de la necesidad de un cuerpo físico en el espacio real. Por tanto, lo que parece

lógico es un enfoque híbrido entre robótica y programación, donde una fusión entre lo físico y lo virtual proporciona más flexibilidad a los docentes y a los estudiantes.

La robótica y la programación en conjunto introducen a sus interesados en una dimensión sensorial, dado que brindan una experiencia de aprendizaje particular respecto a otras áreas, porque la potencia computacional se localiza no solo en una pantalla, sino también, en objetos tangibles, que comparten con nosotros un espacio físico y la posibilidad de ser afectados por nuestro entorno. Aprender a través de la robótica aumenta el compromiso de los niños en actividades basadas en la manipulación, el desarrollo de habilidades motoras, la coordinación ojo-mano y una forma de entender las ideas abstractas. Además, las actividades basadas en robots proporcionan un contexto apropiado para el comportamiento cooperativo y el trabajo en equipo.

En Argentina, existen distintos centros de estudios relacionados con la robótica educativa, uno de los más renombrados es RoboGroup. Esta es una empresa nacional dedicada al diseño, fabricación y capacitación en robótica, que, según la misma, su objetivo es insertar la robótica como sistema interdisciplinario de aprendizaje en las entidades educativas de todos los niveles de nuestro país. Anualmente organiza campeonatos de robots para alumnos de colegios primarios y secundarios llamados Roboliga.

## Capítulo 3 – Arduino

### 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. Es decir, que promete ser una forma sencilla de realizar proyectos interactivos para cualquier persona.

Arduino se basa en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación **Processing**. Es un dispositivo que conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital.



Ilustración 8 - Logo Arduino

### 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.



Ilustración 9 - Logo de Processing

Uno de los objetivos declarados de Processing es el de actuar como herramienta para que artistas, diseñadores visuales y miembros de otras comunidades ajenos al lenguaje de la programación, aprendieran las bases de la misma a través de una muestra gráfica instantánea y visual de la información.

El lenguaje de Processing se basa en Java, aunque hace uso de una sintaxis simplificada y de un modelo de programación de gráficos.

Processing dio lugar a otro proyecto, Wiring, que utiliza el IDE de Processing con una versión simplificada del lenguaje

C++ como modo de enseñar a artistas cómo programar microcontroladores. En la actualidad hay dos Proyectos independientes de hardware, Wiring y Arduino, utilizando el entorno de Wiring y su lenguaje. El entorno Fritzing de software es otro del mismo tipo, que ayuda a los diseñadores y artistas a documentar sus prototipos interactivos y dar paso en la creación de prototipos físicos al producto real.



### 3.3 ¿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.

### 3.4 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 multidisciplinarios.
- ✓ 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.

### 3.4 Hardware Arduino

El hardware 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.

## 3.5 Distintas plataformas para Arduino

ENTRY LEVEL	<div> <div>UNO</div> <div>LEONARDO</div> <div>101</div> <div>ESPLORA</div> <div>MICRO</div> <div>NANO</div> <div>MINI</div> <div>MKR2UNO ADAPTER</div> </div> <div> <div>STARTER KIT</div> <div>LCD SCREEN</div> </div>
ENHANCED FEATURES	<div> <div>MEGA</div> <div>ZERO</div> <div>DUE</div> <div>MEGA ADK</div> <div>MO</div> <div>MO PRO</div> <div>MKR ZERO</div> <div>MOTOR SHIELD</div> </div> <div> <div>USB HOST SHIELD</div> <div>PROTO SHIELD</div> <div>MKR PROTO SHIELD</div> <div>4 RELAYS SHIELD</div> </div> <div> <div>MEGA PROTO SHIELD</div> <div>MKR RELAY PROTO SHIELD</div> <div>ISP</div> <div>USB2SERIAL MICRO</div> </div> <div> <div>USB2SERIAL CONVERTER</div> </div>
INTERNET OF THINGS	<div> <div>YÜN</div> <div>ETHERNET</div> <div>TIAN</div> <div>INDUSTRIAL 101</div> <div>LEONARDO ETH</div> <div>MKR FOX 1200</div> </div> <div> <div>MKR WAN 1300</div> <div>MKR CSM 1400</div> <div>MKR1000</div> <div>YÜN MINI</div> <div>YÜN SHIELD</div> <div>WIRELESS SD SHIELD</div> </div> <div> <div>WIRELESS PROTO SHIELD</div> <div>ETHERNET SHIELD V2</div> <div>GSM SHIELD V2</div> <div>MKR IoT BUNDLE</div> </div>
EDUCATION	<div> <div>CTC101</div> </div>
WEARABLE	<div> <div>GEMMA</div> <div>LILYPAD ARDUINO USB</div> <div>LILYPAD ARDUINO MAIN BOARD</div> <div>LILYPAD ARDUINO SIMPLE</div> </div> <div> <div>LILYPAD ARDUINO SIMPLE SNAP</div> </div>
3D PRINTING	<div> <div>MATERIA 101</div> </div>

BOARDS

MODULES

SHIELDS

KITS

ACCESSORIES

COMING NEXT

## 3.6 Universo Arduino

### 3.6.1 Placas

Arduino Galileo, Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yún, Arduino Tre (En Desarrollo), Arduino Zero, Arduino Micro, Arduino Esplora, Arduino Mega ADK, Arduino Ethernet, Arduino Mega 2560, Arduino Robot, Arduino Mini, Arduino Nano, LilyPad Arduino Simple, LilyPad Arduino SimpleSnap, LilyPad Arduino, LilyPad Arduino USB, Arduino Pro Mini, Arduino Fio, Arduino Pro, Arduino MKR1000/Genuino MKR1000, Arduino MICRO/Genuino MICRO, Arduino 101/Genuino 101, Arduino Gemma.

### 3.6.2 Placas de expansión (shields)

Arduino GSM Shield, Arduino Ethernet Shield, Arduino WiFi Shield, Arduino Wireless SD Shield, Arduino USB Host Shield, Arduino Motor Shield, Arduino Wireless Proto Shield, Arduino Proto Shield.

### 3.6.3 Kits

The Arduino Starter Kit, Arduino Materia 101.

### 3.6.4 Accesorios

TFT LCD Screen, USB/Serial Light Adapter, Arduino ISP, Mini USB/Serial Adapter.

### 3.6.5 Impresoras 3d

Arduino 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 AVR más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños, aunque también nos encontramos con microcontroladores CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR son plataformas diferentes, pero gracias al IDE de Arduino los programas se compilan y luego se ejecutan sin cambios en cualquiera de las plataformas.

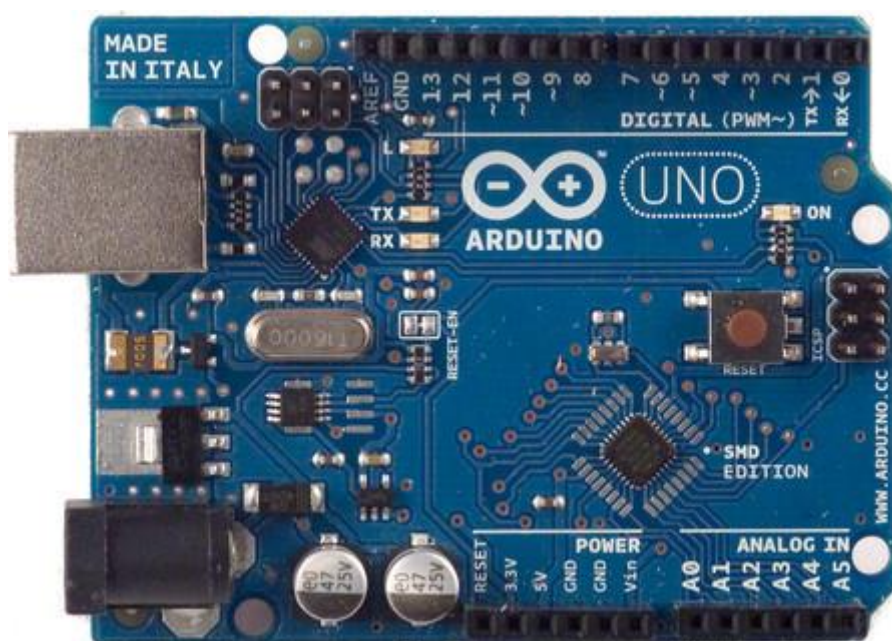


Ilustración 10 - Arduino Uno

## 3.7 Diferencias entre distintas placas de la familia Arduino

La diferencia entre los distintos Arduino la encontraremos por un lado en la tensión utilizada en las placas. Generalmente las microcontroladoras con CortexM3 tienen un voltaje de 3,3 voltios, mientras que la mayor parte de las placas con AVR utilizan una tensión de 5 voltios. Esto de todas formas no es un factor decisivo en la adquisición de una placa dado que existen conmutadores de tensión como actuadores/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).

### 3.8 ¿Por qué usar Arduino?

#### 3.8.1 La comunidad formada a su alrededor y la generación de habilidades compartidas

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.

#### 3.8.2 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.

#### 3.8.3 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.

#### 3.8.4 Incorporación de Arduino en las escuelas

La posibilidad que permite articular el modo de funcionamiento propio de la comunidad Arduino con la dinámica de las escuelas a través de la conformación de una red de trabajo colaborativo, utilizando los medios multimediales para abordar sobre cursos y capacitaciones.

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 (**3r: reducir, reciclar, reutilizar**).

### 3.9 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.

Este conjunto de 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).

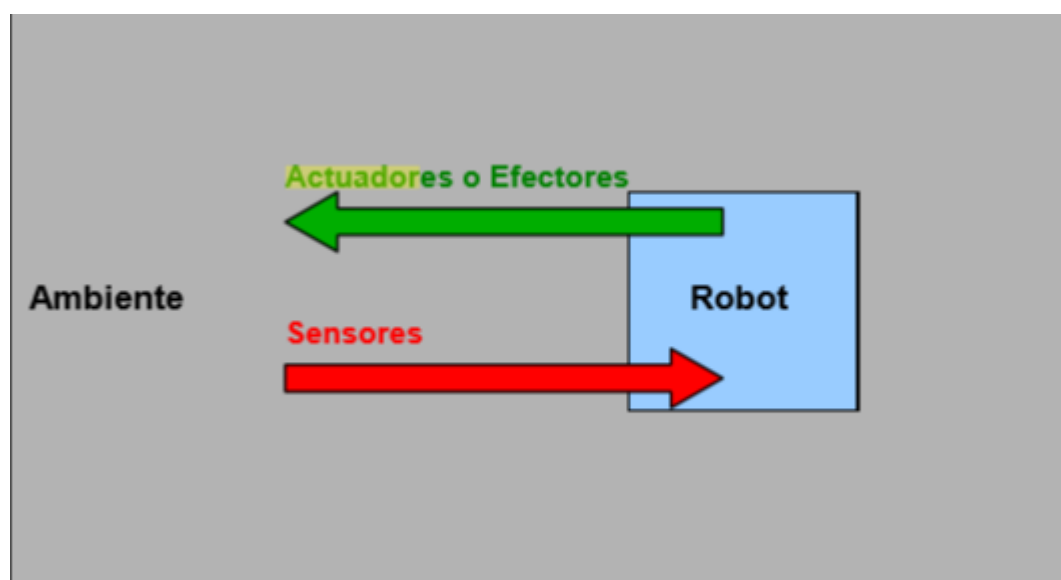


Ilustración 11- Representación actuadores y sensores

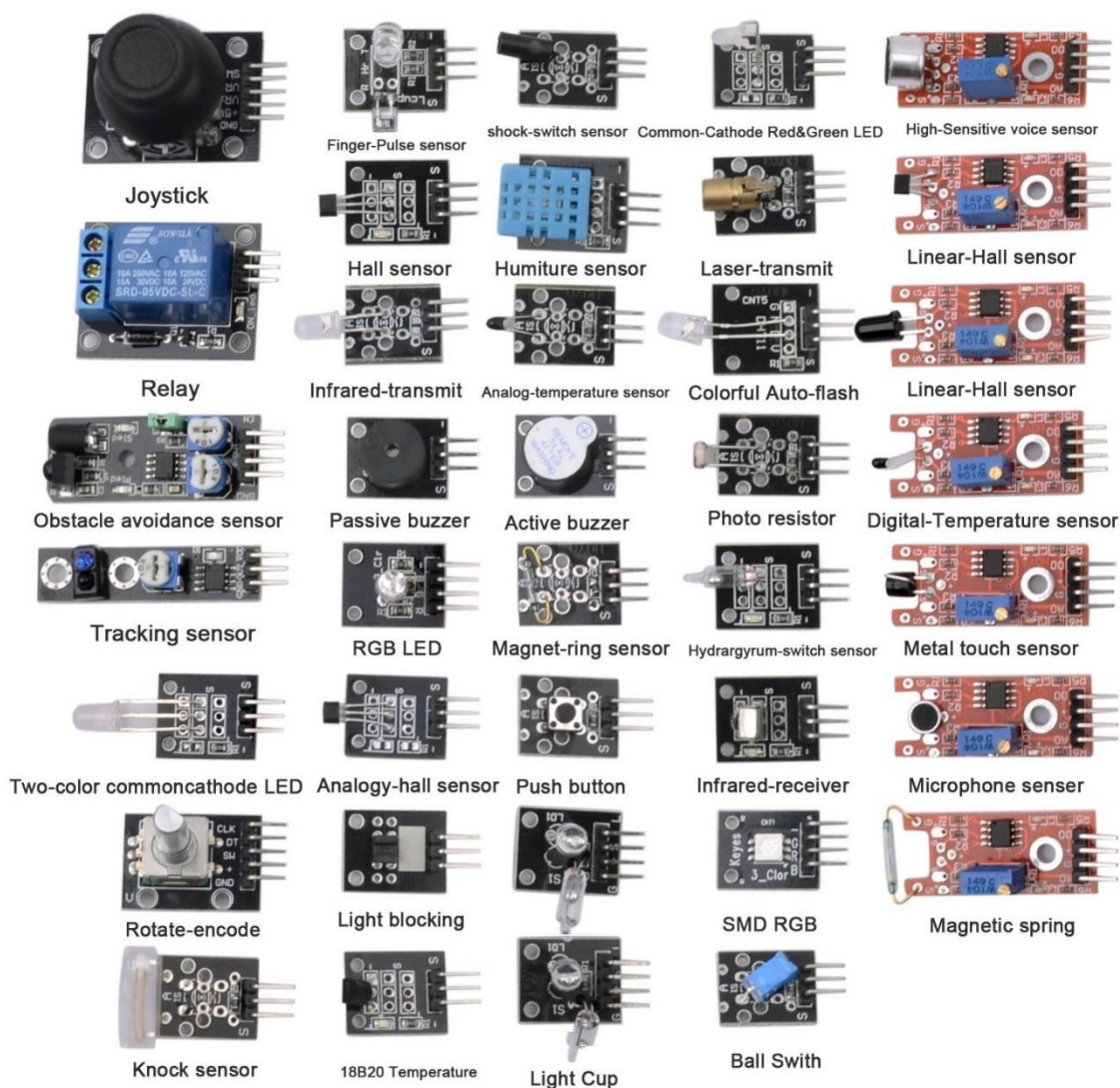
### 3.10 Actuadores en el SAR

Una de las ventajas que dio el empuje industrial alrededor de tecnologías como Arduino fue la creación de actuadores compatibles con estas placas.

Precisamente en el SAR (Sistema Autónomo Robótico) se utilizarán

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





*Ilustración 12- Actuadores y Sensores (Mundo Arduino)*

### 3.11 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
- Temperatura KY-001
  - Incorporado para analizar la temperatura del ambiente
- Sensor de presencia de gases MQ-7
  - Detección de monóxido de carbono

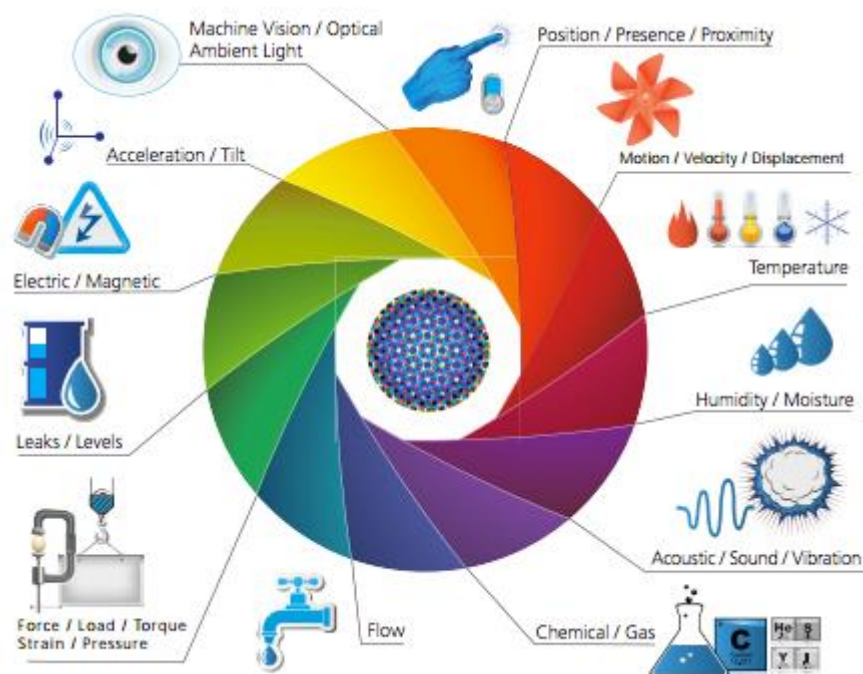


Ilustración 13- Representación de sensores

### 3.12 Módulos o shields en el SAR

El SAR usa

- 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

\*El uso de estos módulos queda en forma tentativa, dado que existen también en la Raspberry y su uso puede ser complementario.

\*\* La implementación y el esquema se encuentran en los anexos a este documento.

## Capítulo 4 – Raspberry Pi

### 4.1 ¿Qué es Raspberry Pi?

**Raspberry Pi** es un computador de placa reducida (SBC) desarrollado en Reino Unido por la Fundación Raspberry Pi. Su lanzamiento fue el 29 de febrero del 2012 con el *Raspberry Pi 1 Modelo A*. Su costo es relativamente bajo en relación a sus especificaciones técnicas, dado que su objetivo primordial es el de estimular la enseñanza de la informática en las escuelas.

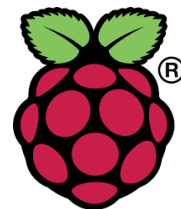


Ilustración 14 - Logo oficial de Raspberry Pi

### 4.2 Especificaciones técnicas de las distintas versiones

	Raspberry Pi 1 Modelo A	Raspberry Pi 1 Modelo B	Raspberry Pi 1 Modelo B+	Raspberry Pi 2 Modelo B	Raspberry Pi 3 Modelo B
SoC	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB)			Broadcom BCM2836 (CPU + GPU + DSP + SDRAM + Puerto USB)	Broadcom BCM2837 (CPU + GPU + DSP + SDRAM + Puerto USB)
CPU	ARM 1176JZF-S a 700 MHz (familia ARM11)			900 MHz quad-core ARM Cortex A7	1.2GHz 64-bit quad-core ARMv8
Juego de instrucciones	RISC de 32 bits				
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC				
Memoria SDRAM	256 MiB compartidos con la GPU	512 MiB compartidos con la GPU, desde el 15 de octubre del 2012		1 GB compartidos con la GPU	
Puertos USB 2.0	1	2	4		
Entradas de vídeo	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la Fundación Raspberry Pi				
Salidas de vídeo	Conector RCA (PAL y NTSC), HDMI (rev 1.3 y 1.4), interfaz DSI para panel LCD				
Salidas de audio	Conector de 3.5 mm, HDMI				
Almacenamiento integrado	SD, MMC, ranura para SDIO		MicroSD		
Conectividad de red	Ninguna	10/100 Ethernet (RJ45) via hub USB			10/100 Ethernet (RJ45) vía hub USB, Wifi 802.11n, Bluetooth 4.1
Periféricos de	8 x GPIO, SPI, I <sup>2</sup> C, UART			17 x GPIO y un bus HAT ID	



<b>bajo nivel</b>				
<b>Consumo energético</b>	500 mA (2.5 W)	700 mA (3.5 W)	600 mA (3.0 W)	800 mA (4.0 W)
<b>Fuente de alimentación</b>	5 V vía Micro USB o GPIO header			
<b>Dimensiones</b>	85.60mm x 53.98mm			
<b>SO soportados</b>	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM.  RISC OS			

### 4.3 Sistemas Operativos compatibles

Los computadores Raspberry Pi utilizan en su mayoría sistemas operativos basados en GNU/Linux compatibles con el mismo, alguno de ellos son los siguientes:

- Arch Linux
- Android
- Debian Whezzy
- Ubuntu Mate
- Google Chromium OS
- Raspbian

Este último (Raspbian), es una distribución derivada del sistema operativo Debian, la cual fue modificada y optimizada para el hardware de Raspberry Pi. Es la distribución por defecto recomendada por la Fundación Raspberry Pi para utilizarse en dicho computador.

Por otro lado, también existe una versión de Windows 10 desarrollada específicamente para sistemas embebidos, denominada Windows CE, compatible con esta plataforma.

Los sistemas operativos del tipo RISC son también compatibles con Raspberry Pi.

### 4.4 Accesorios para Raspberry Pi

La placa **Raspberry Pi** necesita de ciertos accesorios para poder ponerla en funcionamiento, como una fuente de alimentación de al menos 1000mah, un cable HDMI, una tarjeta de memoria SD/microSD con el Sistema Operativo y un adaptador WIFI o un cable RJ45 para poder conectarla en red. Además, por estética o por protección existen variadas cajas o carcasas para su resguardo.

Algunos de los accesorios más comunes compatibles para esta plataforma son los siguientes:

- Cámara para Raspberry Pi V2: Es una cámara de alta definición (HD) que se puede conectar a cualquier modelo de Raspberry para la captura de imágenes o videos en HD. Esta cámara posee un sensor de imagen IMX219PQ de Sony el cual ofrece imágenes de video de alta velocidad y alta sensibilidad, además de un enfoque fijo de hasta 8 megapíxeles.



Ilustración 15 -  
Cámara Raspberry  
Pi V2

- Pantalla táctil LCD para Raspberry Pi de 7": Es la pantalla táctil oficial de esta plataforma. Se trata de una pantalla táctil LCD capacitiva de varios toques (hasta 10 toques con los dedos), con un display de 7 pulgadas, una resolución de 800x480 con 60 fotogramas por segundo (fps) y color RGB de 24 bits. Se conecta a través de una placa adaptadora que se ocupa de la conversión de potencia y señal. Sólo se requieren dos conexiones a la Pi; la de energía a través del puerto GPIO del Pi y un cable de cinta que se conecta al puerto DSI presente en todo modelo de Raspberry Pi.



Ilustración 16 - Pantalla táctil de  
Raspberry Pi

- Adafruit Prototyping Pi Plate Kit: Se trata de una placa que se empalma en la parte superior de las Raspberry Pi, en la cual se pueden soldar componentes en su área de GPIO y además cuenta en su centro con un área de protoboard.



Ilustración 17 - Adafruit  
Prototyping Pi

- Western digital Pidrive: Es un disco rígido exclusivo para esta plataforma, de una capacidad de 314 GB, creado por la marca homónima. Cuenta con una interfaz de conexión USB para comunicarse con la Raspberry Pi.



Ilustración 18 - Pidrive

- Pi TFT: Es una pequeña pantalla táctil de 2.8 pulgadas del tipo resistiva, que se empalma en la parte superior del Raspberry. Su resolución es de 320x240 y color de 16 bits. Se le pueden soldar 4 botones de forma opcional para su manipulación.



Ilustración 19 - Pi TFT

## 4.5 ¿Por qué elegir Raspberry Pi?

Al igual que lo que se mencionó en el capítulo 3 con respecto a Arduino, la plataforma Raspberry Pi presenta una serie de ventajas, con respecto a otras arquitecturas similares, que se describen a continuación:

- **Comunidad:** Existe una vasta comunidad en variadas partes del mundo que trabaja, da soporte y utiliza esta plataforma para diversos proyectos, que dado esto, se expande con él tiempo. A su vez, como se mostró en el apartado anterior, se cuenta con una serie de accesorios que facilitan su uso.
- **Bajo costo:** Como se mencionó con anterioridad esta SBC, como lo es la Raspberry Pi, se puede conseguir a un bajo costo teniendo en cuenta las prestaciones que posee. Dado que su fin es más bien educativo.
- **Prestaciones:** Después de un análisis en relación a variados proyectos educativos que han utilizado esta plataforma, se puede deducir que cumple con el objetivo para el cual fue desarrollado alla por el año 2012.
- **Interfaces y GPIO:** Cuenta con una variedad de interfaces para la conexión de distintos periféricos (HDMI, USB, Ethernet, Wifi, bluetooth) y a su vez, los modelos más actuales (la versión 3), vienen con 40 pines del tipo GPIO, lo que lo convierte en un SBC muy versátil en cuanto a su utilidad. En las siguientes imágenes se pueden apreciar las distintas interfaces y GPIOs de esta plataforma.

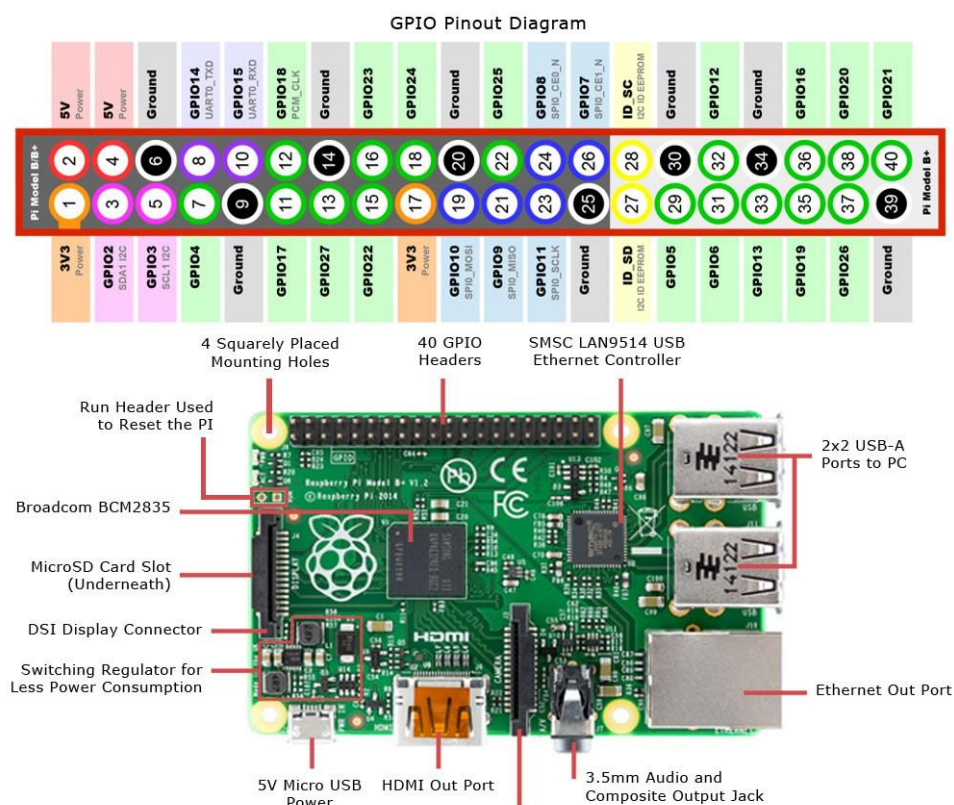


Ilustración 20 - Raspberry Pi y sus GPIO

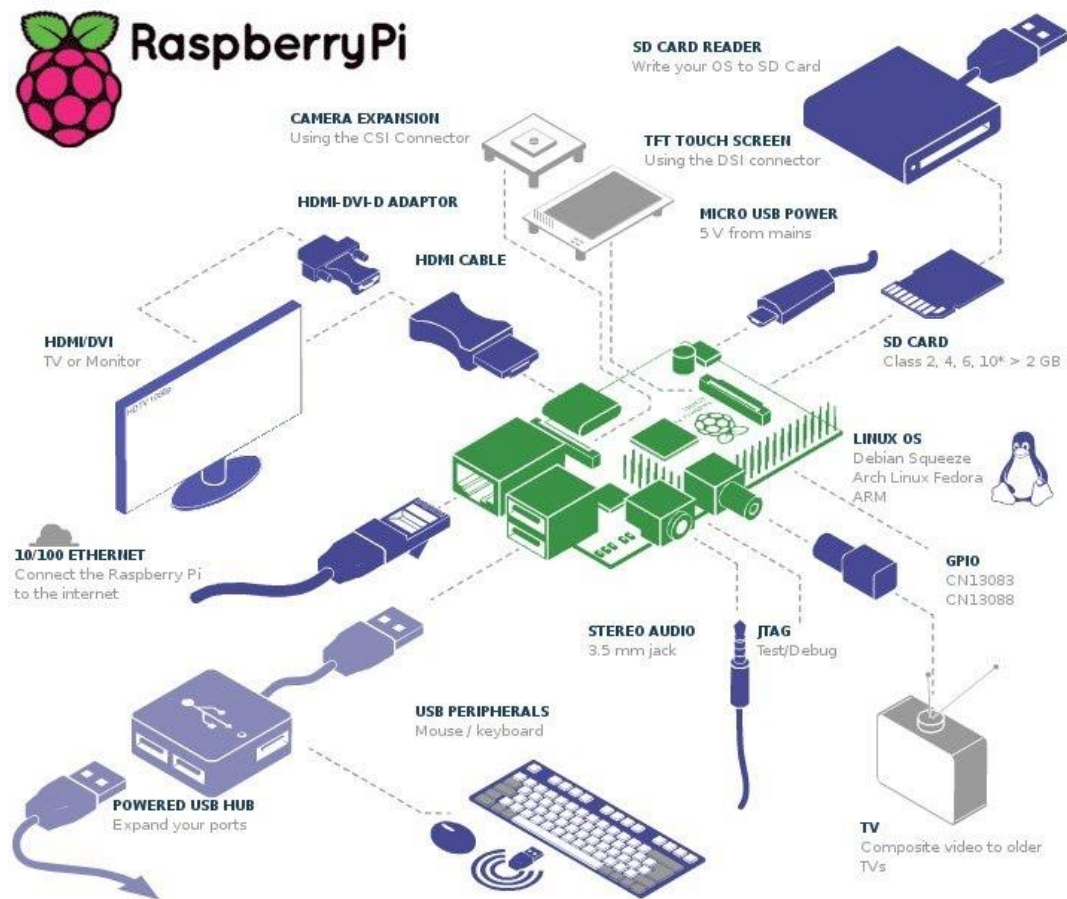


Ilustración 21 - Interfaces de Raspberry Pi

## Capítulo 5 - Aplicaciones Móviles

### 5.1 ¿Qué son las aplicaciones móviles?

“Una aplicación móvil es un programa que se puede descargar y/o acceder directamente desde un aparato móvil”

### 5.2 Las App's

“Una aplicación móvil o app es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo — profesional, de ocio, educativas, de acceso a servicios, etc—, facilitando las gestiones o actividades a desarrollar”



Ilustración 22 - Aplicaciones móviles

Al ser aplicaciones residentes en los dispositivos

están escritas en algún lenguaje de programación compilado, y su funcionamiento y recursos se encaminan a aportar una serie de ventajas tales como:

- Un acceso más rápido y sencillo a la información necesaria sin necesidad de los datos de autenticación en cada acceso.
- Un almacenamiento de datos personales que, a priori, es de una manera segura.
- Una gran versatilidad en cuanto a su utilización o aplicación práctica.
- La atribución de funcionalidades específicas.
- Mejorar la capacidad de conectividad y disponibilidad de servicios y productos (usuario-usuario, usuario-proveedor de servicios).

#### 5.2.1 Las webApps

Una web app no es más que una versión de página web optimizada y adaptable, a cualquier dispositivo móvil independientemente del sistema operativo que utilice. Esta optimización es posible gracias al lenguaje HTML5, combinado con hojas de estilo CSS3, que permiten dicha adaptabilidad, denominada “Responsive Web Design”.



Ilustración 23 - App nativa vs Web App



### 5.2.2 Ventajas de las Web-App:

- ✓ No ocupa espacio de memoria en los dispositivos (no es una app).
- ✓ No requiere actualizaciones ya que al ser una página web siempre se accede a la última versión.
- ✓ Menor consumo de recursos del dispositivo y mejor rendimiento del mismo al no ser una aplicación nativa.
- ✓ El proyecto de implementación de una Web-App es más económico que el de una App.

### 5.2.3 Desventajas de las Web-Apps

- ✓ No permite la promoción y distribución a través de los markets (Playstore, Nokia store, App Store, Windows Phone Apps)
- ✓ Requiere de una conexión entre el cliente y el servidor (por ejemplo por internet o una LAN).
- ✓ Menor usabilidad, al no poder aprovecharse de los recursos del propio dispositivo (geolocalización, notificaciones “push”).
- ✓ Carece de un icono de lanzamiento específico. Es necesario un navegador



*Ilustración 24 – WebApps – Diseño multipropósito*

## 5.3 ¿Cuáles son los distintos sistemas operativos para dispositivos móviles?

Un sistema operativo móvil o SO móvil es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él. Al igual que los PCs que utilizan Windows, Linux o Mac OS, los dispositivos móviles tienen sus sistemas operativos como Android, IOS o Windows Phone, entre otros.

A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de

mercado de sistemas operativos móviles en el primer trimestre de 2016 era el siguiente:

- ✓ Android 84,1 %
- ✓ iOS 14,8 %
- ✓ Windows Phone 0,7 %
- ✓ BlackBerry OS 0,2 %
- ✓ Otros 0,2 %

Android tiene la mayor cuota, desde enero 2011, con más de la mitad del mercado, experimentó un creciente aumento y en solo dos años (2009 a comienzos de 2011) ha pasado a ser el SO móvil más utilizado.

Es por esto, que en principio se pensó desarrollar la App que se comunica con el SAR y por ende al RM, en Android.

## 5.4 Android

Es sin duda el líder del mercado móvil en sistemas operativos, está basado en Linux, diseñado originalmente para cámaras fotográficas profesionales, luego fue vendido a Google y modificado para ser utilizado en dispositivos móviles como los teléfonos inteligentes y luego en tablets. Actualmente se encuentra en desarrollo para usarse en netbooks y PCs; además de la creación de la Open Handset Alliance,

compuesto por 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para celulares, esto le ha ayudado mucho a Google a masificar el S.O, hasta el punto de ser usado por empresas como HTC, LG, Samsung, Motorola entre otros.



Ilustración 25 - Arquitectura de Android



Ilustración 26 - Logo de Android

Las aplicaciones para Android se escriben y desarrollan en Java aunque con unas APIs propias, por lo que las aplicaciones escritas en Java para PC y demás plataformas ya existentes no son compatibles con este sistema.

## 5.5 Aplicaciones móviles multiplataforma

### 5.5.1 Diferencias entre aplicaciones y web móviles

Las aplicaciones comparten la pantalla del teléfono con las webs móviles, pero mientras las primeras tienen que ser descargadas e instaladas antes de usar, a una web puede accederse simplemente usando Internet y un navegador compatible con la aplicación web. Pero, no todas pueden verse correctamente desde una pantalla generalmente más pequeña que la de un ordenador de escritorio.

Las que se adaptan especialmente a un dispositivo móvil se llaman “web responsiva” y son ejemplo del diseño líquido (adaptativo), ya que se puede pensar en ellas como un contenido que toma la forma del contenedor, mostrando la información según sea necesario.

### 5.5.2 Aplicaciones Nativas

Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un sistema diferente, por lo que si se desea que una app esté disponible en todas las plataformas se deberán de crear varias apps con el lenguaje del sistema operativo seleccionado.

- ✓ Las apps para iOS se desarrollan con lenguaje Objective-C
- ✓ Las apps para Android se desarrollan con lenguaje Java, sobre API's
- ✓ Las apps en Windows Phone se desarrollan en .Net

Cuando hablamos de desarrollo móvil casi siempre nos estamos refiriendo a aplicaciones nativas. La principal ventaja con respecto a los otros dos tipos, es la posibilidad de acceder a todas las características del hardware del móvil: cámara, GPS, agenda, dispositivos de almacenamiento y la falta de necesidad de estar conectado a internet para que funcione. Por otro lado, existe la promoción de las Apps por medio de las tiendas de aplicaciones (app store). Las aplicaciones nativas presentan un mayor rendimiento y aprovechamiento del hardware de cada smartphone o dispositivo móvil.



Ventajas	Inconvenientes
<ul style="list-style-type: none"> <li>• Acceso completo al dispositivo</li> <li>• Mejor experiencia del usuario</li> <li>• Visibilidad en APP Store</li> <li>• Envío de notificaciones o “avisos” a los usuarios</li> <li>• La actualización de la app es constante</li> </ul>	<ul style="list-style-type: none"> <li>• Diferentes habilidades / idiomas / herramientas para cada plataforma de destino</li> <li>• Tienden a ser más caras de desarrollar</li> <li>• El código del cliente no es reutilizable entre las diferentes plataformas</li> </ul>

Ilustración 27 - Cuadro comparativo nativas

### 5.5.3 Aplicaciones Web

Una aplicación web o webapp es la desarrollada con lenguajes muy conocidos por los programadores, como pueden ser HTML, Javascript y CSS.

La principal ventaja con respecto a la nativa es la posibilidad de programar independientemente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL.

La gran diferencia con una aplicación nativa es que no necesita instalación por lo que no pueden estar visibles en app store y la promoción y comercialización debe realizarse de forma independiente. De todas formas, se puede crear un acceso directo o link para acceder a la aplicación en el dispositivo.

Las apps web móviles son siempre una buena opción si nuestro objetivo es adaptar la web a formato móvil.

Ventajas	Inconvenientes
<ul style="list-style-type: none"> <li>• El mismo código base reutilizable en múltiples plataformas</li> <li>• Proceso de desarrollo más sencillo y económico</li> <li>• No necesitan ninguna aprobación externa para publicarse (a diferencia de las nativas para estar visibles en app store)</li> <li>• El usuario siempre dispone de la última versión</li> <li>• Pueden reutilizarse sitios “responsive” ya diseñados</li> </ul>	<ul style="list-style-type: none"> <li>• Requiere de conexión a internet</li> <li>• Acceso muy limitado a los elementos y características del hardware del dispositivo</li> <li>• La experiencia del usuario (navegación, interacción..) y el tiempo de respuesta es menor que en una app nativa</li> <li>• Requiere de mayor esfuerzo en promoción y visibilidad</li> </ul>

Ilustración 28 - Cuadro comparativo - Aplicaciones Web

### 5.5.4 Aplicaciones Híbridas

Una aplicación híbrida es una combinación de las dos anteriores. Las apps híbridas se desarrollan con lenguajes propios de las webapps, es decir, HTML, Javascript y CSS por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que es posible agrupar los códigos y distribuirla en app store.

PhoneGap es uno de los frameworks más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas. Otro ejemplo de herramienta para desarrollar apps híbridas es Cordova. También existe la opción de servirse de cross-platforms frameworks, para desarrollar una web app.

### 5.5.5 Aplicación híbrida: Web App

Cosiste en diseñar la aplicación como si fuera una web app, y será el propio navegador el que la ejecute. Sin embargo, ésta es precisamente su gran virtud, pero también su principal inconveniente. Nos encontramos con que el código es válido para todos los dispositivos, pero la sensación del usuario y la apariencia no es tan buena como en la app nativa. Phonegap es el framework más popular en esta tecnología.

### 5.5.6 Aplicación híbrida: app interpretada

La aplicación interpretada significa que la aplicación es programada y luego cada terminal la traduce a su propio lenguaje de programación. Facilita el desarrollo de aplicaciones y reduce el esfuerzo considerablemente. Aunque el resultado no es idéntico a la nativa, la apariencia es bastante buena, y en muchas ocasiones puede ser la solución al problema del desarrollo de aplicaciones multiplataforma.

Ventajas	Inconvenientes
<ul style="list-style-type: none"> <li>• Es posible distribuirla en las tiendas de iOS y Android.</li> <li>• Instalación nativa pero construida con JavaScript, HTML y CSS</li> <li>• El mismo código base para múltiples plataformas</li> <li>• Acceso a parte del hardware del dispositivo</li> </ul>	<ul style="list-style-type: none"> <li>• Experiencia del usuario más propia de la aplicación web que de la app nativa</li> <li>• Diseño visual no siempre relacionado con el sistema operativo en el que se muestre</li> </ul>

*Ilustración 29 - Comparativa aplicaciones híbridas*

## 5.6 Investigación en Desarrollo de Apps y WebApps

Al momento de seleccionar el tipo de App o webApp se investigó en las distintas tecnologías para seleccionar la adecuada



*Ilustración 30 - Herramientas para desarrollo de apps*

### 5.6.1 Android Studio

Es el IDE oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA . Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de apps para Android. Además, posibilita el uso de emuladores con distintos niveles y hardware específico para testear las app creadas.

### 5.6.2 App inventor

Es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

Otra gran cualidad es la posibilidad de insertarlo en la educación dado su programación por medio de bloques gráficos, que resultan ser muy intuitivos en aquellas personas que se introducen.

Unas primeras apps para el SAR fueron realizadas bajo esta plataforma, concluyendo que tiene una curva de aprendizaje corta.

### 5.6.3 Intel XDK

Es un kit de desarrollo creado por Intel para crear aplicaciones nativas para los teléfonos celulares y las tabletas que utilizan tecnologías de la web como HTML5, CSS y Javascript. Las aplicaciones están compiladas vía on-line en la plataforma Cordova para hacer aplicaciones cross-platform, pensando en crear apps para IOT. Posee un emulador, App preview mediante QR, Drag and Drop y soporte de plantillas. Además es compatible y da soporte para Android, IOS, Windows Phone y entre otros.

### 5.6.4 Ionic

Es una herramienta (framework), gratuita y open source, para el desarrollo de aplicaciones híbridas, inicialmente pensado para móviles y tablets, basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS, permitiendo una mayor facilidad de desarrollo de aplicaciones, con una menor inversión económica y la creación de aplicaciones con una misma base de código.

### 5.6.5 Cordova

Apache Cordova es un entorno de desarrollo de aplicaciones móviles, originalmente creado por Nitobi y comprado por Adobe. Más tarde fue liberado como Apache Cordova. Permite, construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5, y Javascript. Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa (porque toda la representación gráfica se realiza vía vistas de Web en vez del framework nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo).

### 5.6.6 Meteor

Es una plataforma para crear aplicaciones web en tiempo real construida sobre Node.js. Meteor se localiza entre la base de datos de la aplicación y su interfaz de usuario y se encarga que las dos partes estén sincronizadas.

Como Meteor usa Node.js, se utiliza JavaScript en el cliente y en el servidor. Y más aún, Meteor es capaz de compartir código entre ambos entornos.

### 5.6.7 Meteor y Cordova

Meteor se integra con Cordova, por ende es posible tomar una aplicación existente y ejecutarla en un dispositivo iOS o Android de forma sencilla. Una

aplicación de Cordova es una aplicación web escrita usando HTML, CSS y JavaScript como de costumbre, pero se ejecuta en una vista web incrustada en una aplicación nativa en lugar de en un navegador móvil independiente. Un beneficio importante de empaquetar su aplicación web como una aplicación de Cordova es que todos sus activos se incluyen con la aplicación. Esto asegura que su aplicación se cargue más rápido que una aplicación web que se ejecuta en un servidor remoto, lo que puede hacer una gran diferencia para los usuarios con conexiones lentas. Otra característica de la integración de Cordova en Meteor es la compatibilidad con hot code push, que le permite actualizar su aplicación en los dispositivos de los usuarios sin pasar por el proceso habitual de revisión de la tienda de aplicaciones. Cordova también abre el acceso a ciertas características nativas a través de una arquitectura de complemento. Los complementos permiten utilizar funciones que normalmente no están disponibles para aplicaciones web, como acceder a la cámara del dispositivo o al sistema de archivos local, interactuar con lectores de código de barras o NFC. Debido a que una aplicación de Cordova es una aplicación web, significa que utiliza elementos web estándar para crear su interfaz de usuario en lugar de confiar en componentes de interfaz de usuario nativos específicos de la plataforma.

## Capítulo 6 – Stack MEAN

### 6.1 ¿Qué es MEAN?

“Es un framework o conjunto de subsistemas de software para el desarrollo de aplicaciones, y páginas web dinámicas, que están basadas, cada una de estas en el popular lenguaje de programación conocido como JavaScript. Gracias a esta característica el conjunto se integra exitosamente en una plataforma auto-suficiente”.

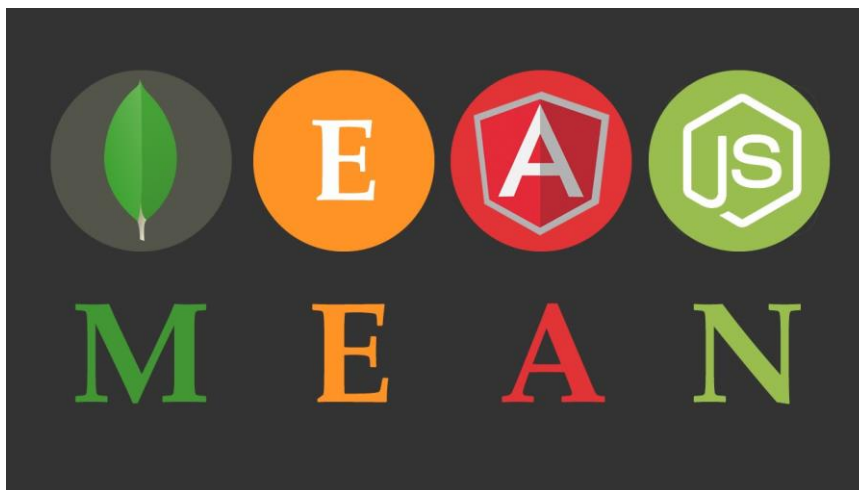


Ilustración 31 - Acrónimo MEAN

### 6.2 ¿Cómo se compone MEAN?

#### 6.2.1 Mongo

Es un sistema de base de datos NoSQL, que almacena los datos en estructuras o “documentos”, los cuales están definidos con la notación JSON (Notación simple de objeto tipo JavaScript), lo que permite una rápida manipulación y transferencia de los datos.

#### 6.2.2 Express

Es un módulo de NodeJS, ofrece los métodos suficientes en JavaScript, para poder manejar las solicitudes o peticiones que se hacen por medio de los métodos del protocolo. También ofrece un sistema simple de enrutamiento (Routing), que dentro del mean stack es aprovechado en el back-end o en el lado del servidor.

#### 6.2.3 Angular

Es un framework para aplicaciones web de TypeScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en

navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

#### 6.2.4 Node

Es la plataforma encargada del funcionamiento del servidor, y trabaja totalmente con JavaScript.

Node.js utiliza por debajo el motor de JavaScript de Google, denominado V8, y provee de una arquitectura orientada a eventos (como la de los navegadores) así como una serie de APIs no-bloqueantes (asíncronas) que le proporcionan un rendimiento y una escalabilidad muy elevadas.



Ilustración 312 - Logo del motor V8

Se puede utilizar para crear cualquier tipo de lógica de aplicación, pero dado que incorpora un módulo para poder actuar como un servidor web, es especialmente popular para crear aplicaciones web. Actualmente lo emplean para sus aplicaciones multitud de empresas de todos los ámbitos, pero especialmente de Internet.

### 6.3 Otros complementos

#### 6.3.1 Twitter Bootstrap

El más popular framework que integra HTML, CSS, y JS para el desarrollo de proyectos adaptables a diferentes medios de reproducción (Responsive).

#### 6.3.2 Compodoc

Documentador JavaScript estático.

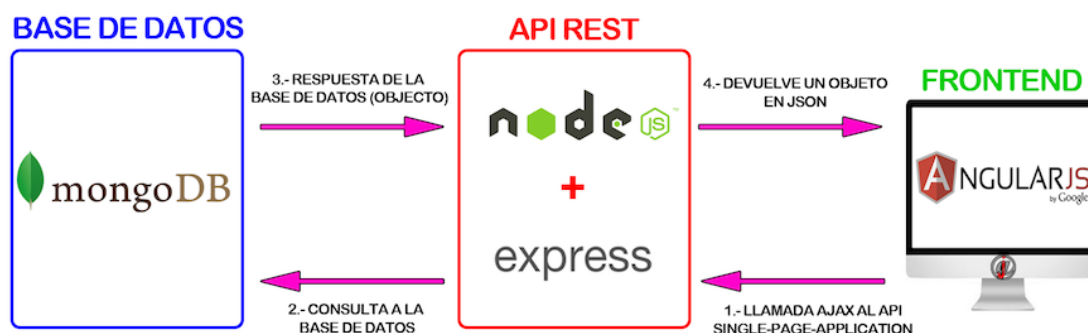


Ilustración 33 - Arquitectura de interacción MEAN

#### 6.3.3 Json

Es también a su vez una palabra formada por iniciales de palabras, y significa: JavaScript Simple Object Notation.



Como su propio nombre indica JSON (pronunciado "yeison") permite representar objetos (en realidad estructuras complejas) en forma de código JavaScript que luego podemos evaluar.

Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función eval(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

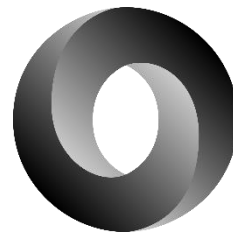


Ilustración 324 - Logo de JSON

“JSON es el pegamento de todas las capas. Es el formato en el que se transfieren los datos entre todos los niveles de la aplicación: navegador, servidor web y servidor de datos”

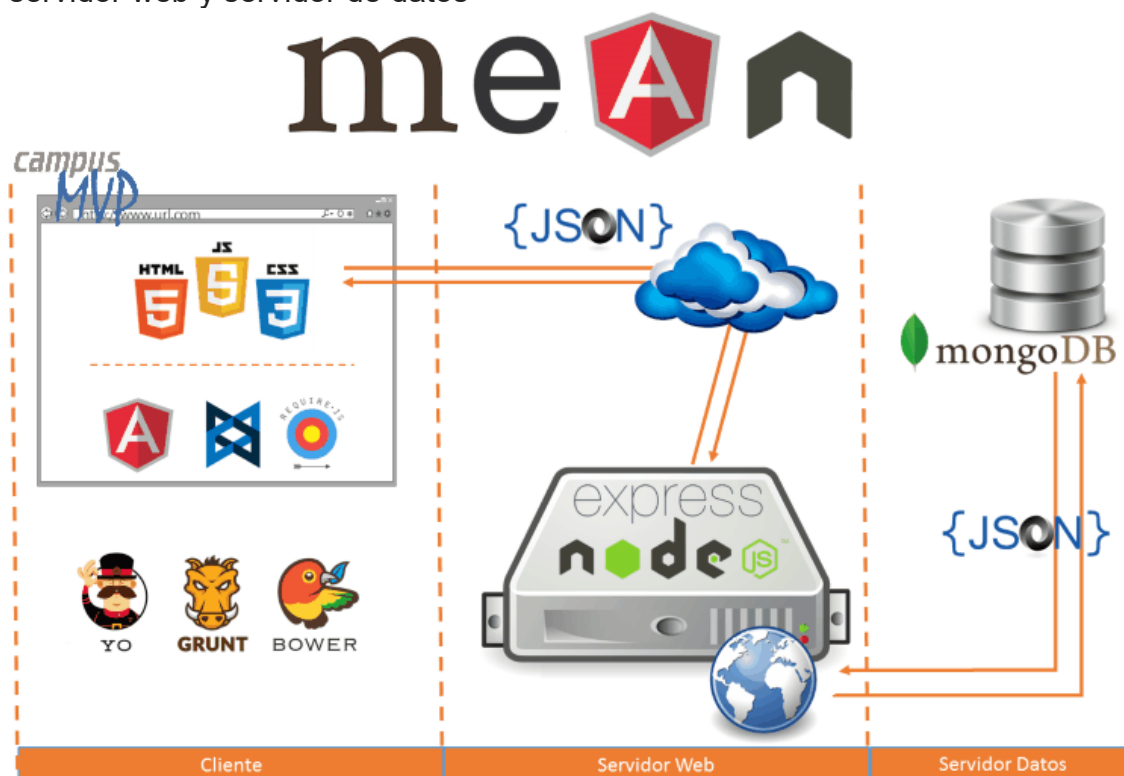


Ilustración 35 - Json pegamento de tecnologías



## Capítulo 7 – Librería Johnny-five y el protocolo Firmata

### 7.1 ¿Qué es Johnny-five?

Johnny-five, más conocido en la jerga informática como J5, es un framework de programación robótica basado en javascript lanzado por la compañía de tecnología web Bocoup en el 2012, es de código abierto, por lo cual cuenta con una gran variedad de

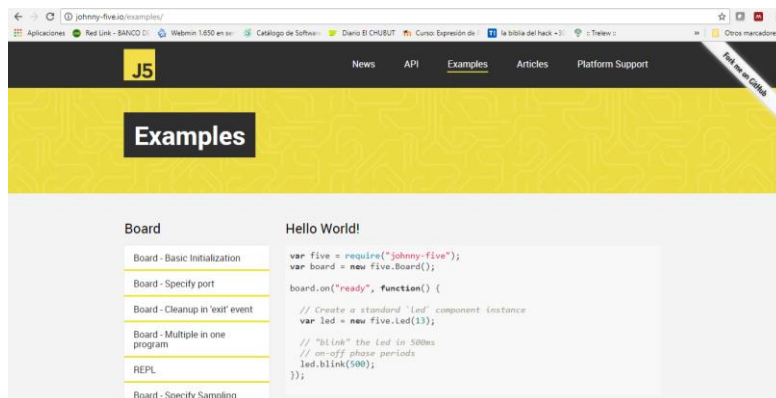


Ilustración 36 - Página oficial de Johnny-Five

desarrolladores, ingenieros y colaboradores que están constantemente perfeccionando y agregando características nuevas a esta herramienta.

Es compatible y da soporte a la mayoría de los microcontroladores y SBC más populares como, los utilizados en el desarrollo de esta tesina, Arduino y Raspberry Pi (ambos en todas sus versiones). Además de placas como BeagleBone, SparkFun, ChipKit, Intel-Galileo, entre otras.

Posee una librería muy completa, con compatibilidad a una vasta cantidad de sensores y actuadores, y ejemplos útiles de usos e implementación de los mismos. Dichos ejemplos se encuentran disponibles en su página oficial con el esquema de conexión de los componentes a las distintas plataformas compatibles.

### 7.2 Instalación

Para poder utilizar Johnny-Five, se debe contar con el Framework node.js (visto en el capítulo 6) instalado en el Sistema Operativo del computador.

La instalación de esta librería es sencilla, dado que la misma se puede agregar al paquete de librerías que se obtienen al crear un proyecto de node.js. Para ello se debe ejecutar el siguiente comando, en una consola, dentro del directorio del proyecto:

*npm install Johnny-five*

Como se puede apreciar, se utiliza el comando npm que no es más que el gestor de módulos y aplicaciones de node.js

## 7.3 Arduino Firmata

Firmata es un protocolo genérico utilizado para la comunicación con microcontroladores desde software instalado en una computadora. Este protocolo se puede implementar en cualquier arquitectura de microcontroladores, así como en cualquier paquete de software.

El objetivo de firmata es permitir controlar completamente un microcontrolador, por ejemplo Arduino, desde un programa instalado en una computadora, sin escribir código de Arduino.

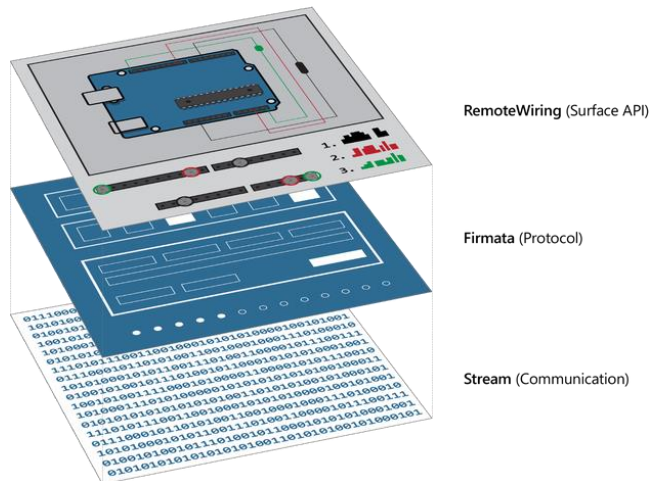


Ilustración 37 - Remote Wiring de Windows

### Ventajas:

- Nuestro programa no está limitado por la memoria RAM y Flash de Arduino
- El software de control se puede programar en cualquier lenguaje, no solo C++, siempre que tenga soporte para firmata. Por ejemplo: Firmata: Processing, Visual Basic, Perl, C#, PHP, Java, **JavaScript**, Ruby y por su puesto Python.

### Desventajas:

- Programas más restringidos, es posible que algunas operaciones complejas no sea posible hacerlas. Por ejemplo, el uso de interrupciones.
- El Arduino siempre debe estar conectado al computador. Debe tener comunicación para poder ejecutar acciones.

## 7.4 Instalación Firmata

Para instalar Firmata se necesita tener conectado el chip Arduino a la computadora a través de un puerto USB. Luego de ello se debe seleccionar desde el Sketch de Arduino el código Firmata, dentro de su respectiva librería, según los dispositivos que se requieran comunicar con el mismo. Para ello se debe ir a Archivo → Ejemplos → Firmata.

En nuestro caso se utilizaron dos códigos Firmata:

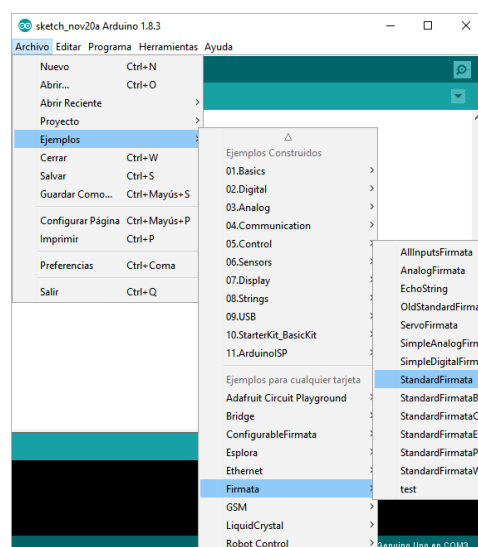


Ilustración 38 - Sketch de Arduino

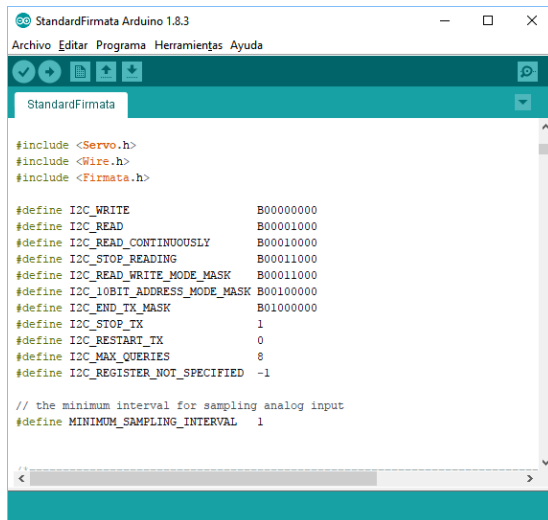


Ilustración 39 - Código StandardFirmata

**ConfigurableFirmata:** Esta versión personalizada del protocolo, es la utilizada para captar la temperatura mediante el sensor para dicho fin conectado al Arduino Nano. Permite separar las características del protocolo en clases individuales, haciendo más sencillo mezclar las características estándar del protocolo con otras personalizadas.

**StandardFirmata:** Es, como su nombre lo indica, el estándar del protocolo que permite la comunicación con la mayoría de los componentes compatibles con Arduino. En nuestro caso, es el utilizado para cargarlo dentro del Arduino Mega para manipular la mayoría de sensores y actuadores del SAR.

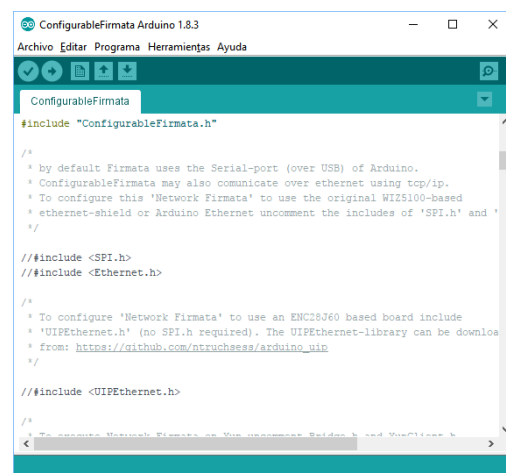


Ilustración 40 - Código ConfigurableFirmata

## Capítulo 8 - Python en Raspberry Pi

### 8.1 ¿Qué es Python?

Se le llama Python a un lenguaje de programación multiparadigma y multiplataforma desarrollado en el año 1991 por Guido Van Rossum. A partir de marzo del 2001 la Python Software Foundation es la encargada de administrar este lenguaje. Su misión es la de fomentar el desarrollo de la comunidad Python.



Ilustración 41 - Logo de Python

Este lenguaje de programación posee las siguientes características principales:

- Es de código abierto.
- Es multiparadigma, permite los estilos de programación orientado a objetos, imperativo y funcional.
- Permite otros paradigmas, a parte de los mencionados, con el uso de extensiones.
- Es interpretado.
- Utiliza tipado dinámico.
- Extensible.

### 8.2 Python con Raspberry Pi

A Python se lo considera un lenguaje de programación fácil de aprender, además de ser muy popular y potente, es un lenguaje serio, usado en distintos ámbitos profesionales. Como se identificó en el apartado anterior, es un lenguaje de código abierto y multiplataforma, por lo que se puede utilizar en cualquier sistema con total libertad e incluso con fines comerciales si se quisiese.

Dicho esto, y por otros motivos, es que la Raspberry Pi Foundation lo ha seleccionado como el lenguaje de programación “estándar” para su plataforma Raspberry Pi. Lo que no significa que la misma se deba programar solo en Python. Raspberry Pi es una computadora que corre un sistema operativo, lo cual le da la facultad de correr diversos programas y por ende variados lenguajes.



Ilustración 42 - Menú de Raspbian

Python es un lenguaje interpretado, que quiere decir que un “intérprete” va leyendo las instrucciones y ejecutándolas en tiempo real. Un intérprete es un software encargado de convertir nuestras sentencias a código máquina. Puede instalarse cualquiera de los muchos intérpretes que existen para tu máquina, ya sea la Raspberry Pi o un ordenador personal.

El sistema operativo Raspbian, desarrollado específicamente para la Raspberry Pi, viene con las versiones de Python 2 y 3 pre- instaladas, con sus respectivos intérpretes.

### 8.3 Raspberry PI – GPIO

La Raspberry Pi 3 cuenta con una cantidad de 40 pines del tipo GPIO (*General Purpose Input/Output, Entrada/Salida de propósito general*)

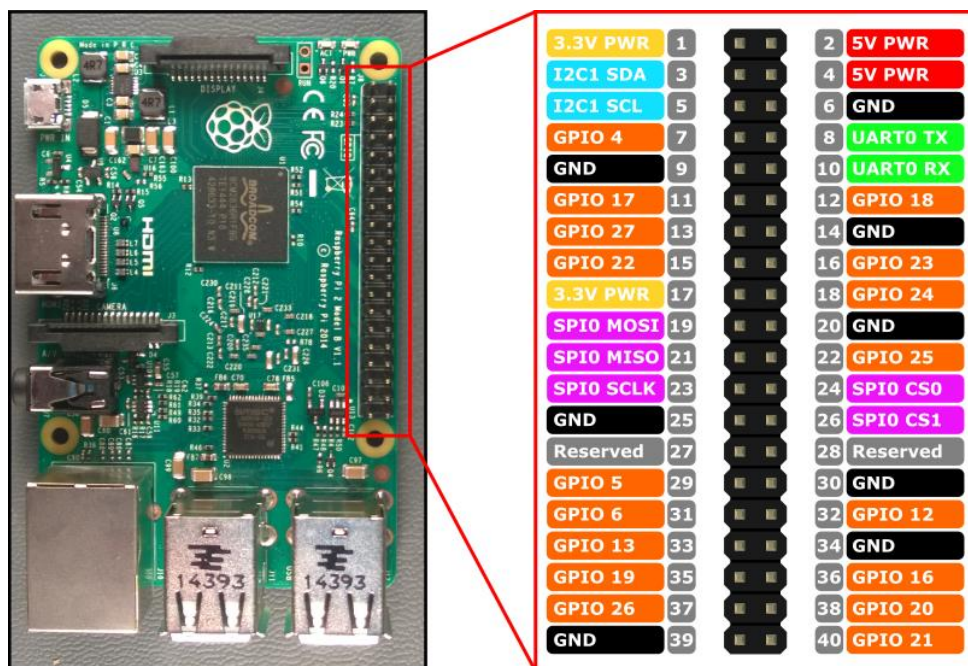


Ilustración 43 - Raspberry Pi 3 y sus GPIOs

Con los cuales, al igual que en la plataforma Arduino, se pueden conectar y manipular un número considerable de actuadores y sensores. Cabe aclarar que todos estos pines son del tipo digital, por lo que, este computador no cuenta con pines del tipo analógico. Para la lectura de sensores que devuelvan valores analógicos se debe utilizar un convertidor externo o un Arduino.

Para manipular estos pines, existe una librería nombrada como RPi.GPIO que nos permite, mediante Python, configurarlos rápidamente.

Como se puede apreciar en la ilustración anterior podemos distinguir entre dos tipos de identificación de estos pines, según su orden físico en la placa o según su posición correspondiente a su conexión física a su CPU (en Raspberry Pi, Broadcom), estas formas de identificación se llaman BCM y BOARD respectivamente.

El número que identifica al pin en el modo BCM es el que esta después de la palabra GPIO (ver diagrama anterior), el índice para identificar los pines en modo BOARD son los que están dentro del círculo. Es importante mencionar

que la identificación de los pines en modo BCM cambio entre la revisión 1 y la revisión 2 del hardware.



## Capítulo 9 - Análisis y selección de tecnologías para desarrollo del SAR

### 9.1 Primer análisis

A partir del análisis de las distintas tecnologías hardware en microcontroladores, microprocesadores actuales y teniendo en cuenta las ventajas y desventajas, tanto en factores como rendimientos, tiempos de respuesta, consumo energético, portabilidad; examinada con la experiencia de la utilización de la familia de Arduino (Arduino Uno, Arduino Mega y Arduino Nano), PC (Ubuntu Mate), Raspberry Pi 3 Modelo B, como así también los distintos módulos arduino-compatibles como la cámara OV7670, GPS, ESP8266(Wifi), Bluetooth, entre otros; y articulando con el software dentro del desarrollo de aplicaciones móviles, permitió analizar Android con sus respectivos entornos de desarrollo como: Android Studio, Intel XDK, Ionic, App Inventor, Cordova, etc; y dada la necesidad de que el SAR debiera ser portable para cualquier dispositivo que posea una conexión Wifi, es que se decide realizar una aplicación web, seleccionando el stack MEAN.

Estas investigaciones nos permiten concluir en la siguiente selección final de tecnologías hardware y software para la creación del SAR.

### 9.2 Selección tecnologías hardware

#### 9.2.1 ¿Por qué Arduino?

Como se abordó en el capítulo 5 (Arduino), siendo una arquitectura hardware pensada para hobbistas, diseñadores y personas no relacionadas con la electrónica ni la programación a bajo nivel, Arduino permite una curva de aprendizaje más corta y la facilidad de conexión de los distintos componentes la hace muy atractiva, para encarar distintos proyectos con diversos niveles de complejidad.

Las placas Arduino, fueron pensadas mayormente para un uso del control de sensores y actuadores utilizando un microcontrolador, esto genera una mejor transición desde la electrónica discreta a la electrónica programable.

Las placas utilizadas fueron la Arduino UNO, donde se elaboraron distintos prototipos con protoboard simples, desde la manipulación de actuadores con motores hasta la toma de datos de distintos sensores como temperatura, humedad, obstáculos, entre otros. El problema encontrado en la placa Arduino UNO era la poca disponibilidad de pines E/S para la cantidad de sensores/actuadores y módulos que se requerían conectar, es por ello que se prefirió ampliar la cantidad de pines optando por la placa Arduino Mega. Esta última, otorgaba mayor cantidad de pines, pero no expandía la cantidad de memoria y procesamiento, generando dificultad a la hora de programar, sin contar con tantas interrupciones hardware y generando un pool constante en su bucle principal (loop). Estos problemas surgieron a la hora de conectar el

módulo de la cámara OV7670 y el módulo wifi ESP8266, los cuales requerían una alta cantidad de pines y nivel de cómputo.

Dado lo limitado en cuanto a poder de procesamiento y memoria, es por ello que se necesitaba otra plataforma que haga uso de las placas Arduino, resultando ser la Raspberry Pi.

### ¿Por qué Raspberry?

En el capítulo 4 se mencionó y analizó el computador de placa reducida (SBC) Raspberry Pi, en el cual se pudo apreciar las características fundamentales del mismo. Esta plataforma se diseñó, primordialmente, con fines didácticos para la educación secundaria, es por esto, que su costo es relativamente bajo. Al contar, la misma, con todas las capacidades básicas de una computadora portátil de hoy en día con su respectivo microprocesador (bastante potente para nuestras necesidades), memorias y puertos físicos (como el USB, HDMI, microSD, entre otros); y la posibilidad de instalar un sistema operativo totalmente funcional y con interfaz gráfica (en este caso Raspbian), es que se seleccionó como centro de administración y control del SAR.

Además, cuenta con pines GPIO para las conexión y manipulación de distintos módulos (como actuadores y sensores), aunque como se justifica en el apartado siguiente, es Arduino Mega y Arduino Nano quienes se encargan de estas funcionalidades, exceptuando la conexión y procesamiento de imágenes dadas por la cámara v2 exclusiva de Raspberry Pi y las comunicaciones inalámbricas proporcionadas por los módulos wifi y bluetooth integrados a este computador.

## Comparativa entre Arduino Mega, Arduino Nano y Raspberry Pi 3 Model b

Factor	Arduino Mega	Arduino Nano	Raspberry Pi3 Model B
Microcontrolador/ Microprocesador	ATmega 1280 - 16Mhz 8bits	ATmega328 - 16Mhz 8bits	Quad Core 1.2GHz Broadcom BCM2837 64bit
Tensión	5v	5v	5v
Memoria	128 KB (Bootloader 4KB)	32 KB (Bootloader 2KB)	1 GB
Digital I/O	54, 15 PWM	22, 6 PWM	40 GPIO
Analog I/O	16	8	
Interfaces	USB x 1(energía)	USB x 1 (energía)	USB x 4, HDMI, CSI, DSI, MicroSD, WLAN y BLE, microUSB (Energía)

Dada la comparativa entre las tecnologías, se decide utilizarlas como articulación entre la computación física (Arduinos) con manejo de sensores y actuadores; y por otro lado el procesamiento y comunicación proporcionado por la Raspberry.

¿Cuáles son los beneficios de esta tupla? se podría utilizar únicamente Raspberry para la elaboración del SAR, pero existen numerosos beneficios que proporciona las placas Arduino frente a la Raspberry y son:

- Menor costo de adquisición del producto en caso de fallos energéticos.
- Mayor flexibilidad y facilidad en la conexión con distintos componentes electrónicos.
- Mayor compatibilidad, con los módulos arduino-compatible y la familia adafruit
- Buen tiempo de respuesta de I/O.
- Alta confiabilidad en la lectura de sensores y en los valores de manipulación de actuadores.

Varios de estos beneficios se deben a que Arduino no posee un sistema operativo, sino un único programa que se ejecuta indefinidamente (LOOP) sin necesidad de correr algún software auxiliar que lo dispare o ejecutando como servicio; logrando concentrar su poder de procesamiento en el único programa definido. La ejecución de servicio genera retrasos propios de sistemas operativos.

## Cámara V2 de Raspberry Pi

La cámara V2 de Raspberry, es una cámara exclusiva de esta plataforma la cual se conecta al puerto CSI de cualquier modelo de este computador (desde la Raspberry Pi 1 hasta el modelo actual, ósea, la Raspberry Pi 3), lo cual permite obviar la conexión pin a pin y abstraernos de la comunicación y procesamiento de la cámara. Como se comentó en el capítulo 4, es una cámara de alta definición de 8 megapíxeles, suficiente para el objetivo que se pretende con el desarrollo del SAR y saltando las problemáticas que se nos presentaron a la hora de probar la cámara OV7670 con Arduino; como el poder de procesamiento de imágenes y transmisión de las mismas (inalámbricamente) hacia otro dispositivo tal como una PC o un dispositivo móvil (en nuestro caso smartphones).

## Módulos de Arduino

Dentro de los módulos, sensores y actuadores de Arduino que se probaron y/o se utilizan, se encuentran:

### Utilizados en el sar:

- El módulo GPS, será utilizado para determinar la ubicación geográfica del SAR (Geolocalización)
- Sensor de temperatura KY-001(-55° a +125°)

- Sensor ultrasonido HC-SR04 para determinar presencia de objetos a determinadas distancia y tratar de evitar el impacto con los mismos
- Motores CC para la movilidad del SAR dentro del ambiente

Probados:

- El módulo wifi esp8266 y el módulo Bluetooth HC-05, no se utilizarán debido a que la Raspberry Pi3 Model B, cuenta con cada uno de ellos.
- El módulo Acelerómetro MMA7361.
- Servomotor sg90.
- Sensor de evasión de obstáculos KY032.
- Sensor de golpe KY-031.
- Sensor de llamas KY-026.

## Capítulo 10 – Ensamblado del robot móvil

Para el desarrollo de esta tesina se procedió a armar un prototipo del SAR mediante un robot móvil, el cual cuenta con una variedad de actuadores y sensores que le permiten interactuar con el entorno que lo rodea.

### Componentes

**Una Raspberry Pi 3 model B:** Componente principal del SAR, es el servidor del mismo, encargado de almacenar la aplicación web y recibir las peticiones de los clientes para luego mandar las ordenes a las placas Arduino. Cuenta con una tarje microSD donde almacena el sistema operativo Raspbian el cual se ejecuta al encenderla y permite correr la aplicación desarrollada.



Ilustración 4433 – Raspberry Pi 3



Ilustración 45 - Arduino Mega

**Un Arduino Mega:** Es el principal controlador del SAR, en él se conectan todos los sensores y actuadores (a excepción del sensor de temperatura). Funciona como intermediario entre la Raspberry y el resto de los componentes, dado que recibe todas las ordenes de ejecución de la misma. En su memoria, se encuentra almacenada una versión del protocolo Firmata nombrada como StandarFirmata (dado por la librería Firmata de Arduino, **Anexo X**) necesaria para establecer la comunicación con los comandos enviados desde Javascript por la aplicación web.

**Un Arduino Nano:** Esta versión de Arduino es la que

se encarga de capturar la temperatura obtenida por el sensor DS18B20. Se debió optar por el uso de otro Arduino, dado que para la captura de temperaturas y el envío de los datos a la Raspberry mediante javascript se necesita una versión particular del protocolo Firmata, nombrada como ConfigurableFirmata (**Anexo X1**).

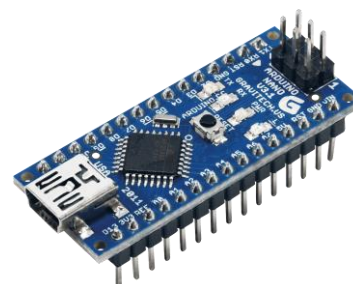


Ilustración 46 - Arduino Nano

**Cuatro motores DC (corriente continua de 3v a 6v) con caja reductora:** Estos motores, en conjunto con cuatro ruedas de plástico cubiertas con una goma cada una, son los que permiten darle la movilidad al SAR.



Ilustración 47 -  
Motores CC



Ilustración 4834 - Sensor de ultrasonido

### Tres sensores ultrasónicos

**HC-SR04:** Los sensores ultrasónicos, se utilizan para determinar la presencia de algún objeto a una distancia menor a 20 centímetros, tanto al frente del SAR como en sus laterales. Al identificar un objeto a una distancia menor a la mencionada, se bloquea el avance del robot en la dirección en donde se encuentre dicho objeto.

**Dos portas pilas AA x4 con sus respectivas pilas recargables:** Utilizados para alimentar de corriente eléctrica a los 4 motores.



Ilustración 49 - Portapilas

**Dos puentes H L298N:** Son los intermediarios entre el Arduino Mega y los motores, cada uno de ellos se encarga de la manipulación de dos motores.



Ilustración 5035 -  
Módulo Puente H

**Una mini Protoboard:** Utilizada como extensión de pines, más que nada para los pines GND y 5v de la placa Arduino Mega.



Ilustración 5136 - Mini-  
protoboard



Ilustración 52 -  
Sensor de  
Temperatura

**Un sensor de temperatura DS18B20 montado sobre una placa KY-001:** Este módulo es el encargado de sensor la temperatura, se encuentra conectado al Arduino Nano.

Ilustración 53- MQ7 CO

**Un sensor de monóxido de carbono MQ-7:** El sensor de monóxido, conectado al Arduino Mega, detecta la ausencia o presencia de dicho gas.





**Un GPS GY-GPS6MV2:** Con este módulo de GPS obtenemos toda la información necesaria con respecto a la Geolocalización del SAR (latitud, longitud, punto cardinal, velocidad, orientación, fecha y hora).

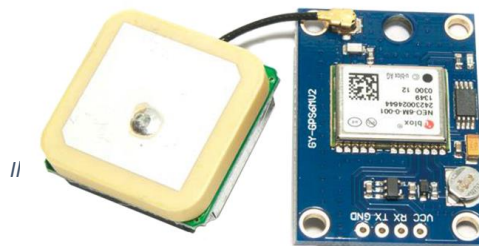


Ilustración 54 - GPS



Ilustración 55 - Cámara V2

**Cámara de Raspberry Pi V2:** Esta cámara, exclusiva de Raspberry, es la utilizada para captar con señal de video en tiempo real (mediante el software motion) el entorno que rodea al SAR.

**PowerBank Malibu de 20Ah con panel solar:** Funciona como batería del SAR, provee de corriente eléctrica a la Raspberry y por ende a los arduinos conectados a ella.



Ilustración 56 - Panel Solar Power Bank

## Estructura

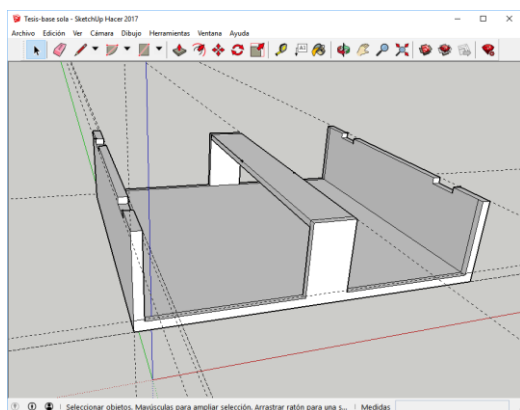


Ilustración 57 - Diseño estructura SketchUp

Para el armado de la estructura se procedió a diseñar en 3D las distintas piezas por medio del entorno de diseño gráfico SketchUp 2017. Se tomaron medidas de los distintos componentes y en base a ellas decidimos dividir el gabinete del SAR en cuatro niveles.

Una vez armados los modelos de los distintos niveles, fuimos imprimiendo los mismos mediante una impresora 3D.

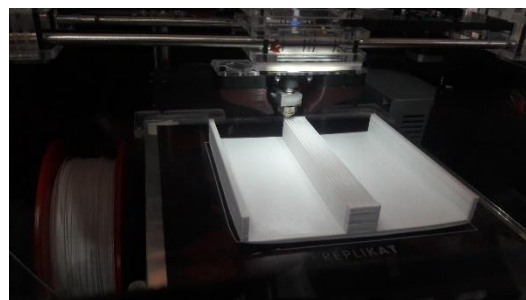


Ilustración 58 – Impresión 3D

**Nivel 1:** El primer nivel es en donde se instalaron los motores, con distintas piezas estructurales metálicas diseñadas exclusivamente para dicha función, además se encuentran los dos puentes H

L298N conectados a cada par de motores respectivamente. Cada motor cuenta con su rueda de plástico.

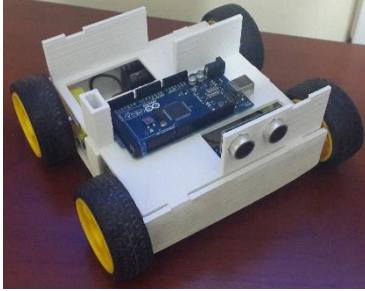


Ilustración 59 - Nivel 2 descubierto

**Nivel 2:** En este nivel se adhirió con tornillos el Arduino Mega y la mini protoboard cada uno con sus respectivas conexiones, además de los porta pilas utilizados para la alimentación de los motores. En su frente se colocó uno de los sensores ultrasónicos HC-SR04 que verifica la presencia de objetos en la parte delantera del SAR.

**Nivel 3:** En el tercer nivel se encuentra la Raspberry Pi y el Arduino nano, en conjunto con una gran variedad de sensores, tales como, 2 sensores HC-SR04, uno en cada uno de los laterales para verificar objetos en dichos lugares. El sensor de monóxido de carbono MQ-7. La cámara de Raspberry en el frente del SAR.

**Nivel 4:** El nivel superior es el que se equipa con la batería portátil solar y que se conecta directamente a la Raspberry, además se encuentra a la vista el GPS y el sensor de temperatura DS18B20. Este último conectado al Arduino Nano.

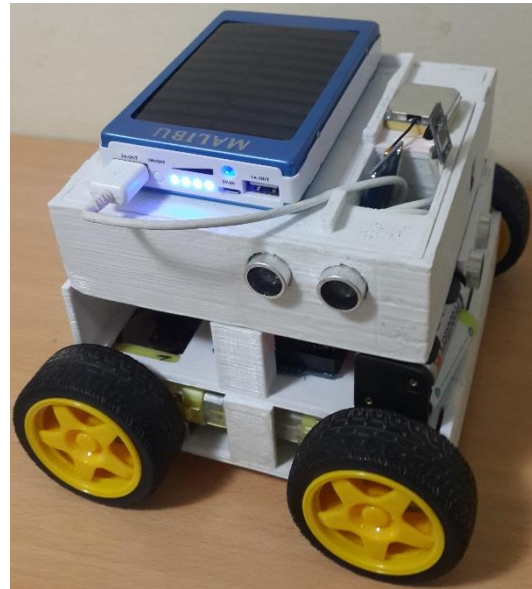


Ilustración 60 - RM Vista Lateral

**Falta agregar esquemas fritzing**

## Capítulo 11 – Desarrollo del SAR

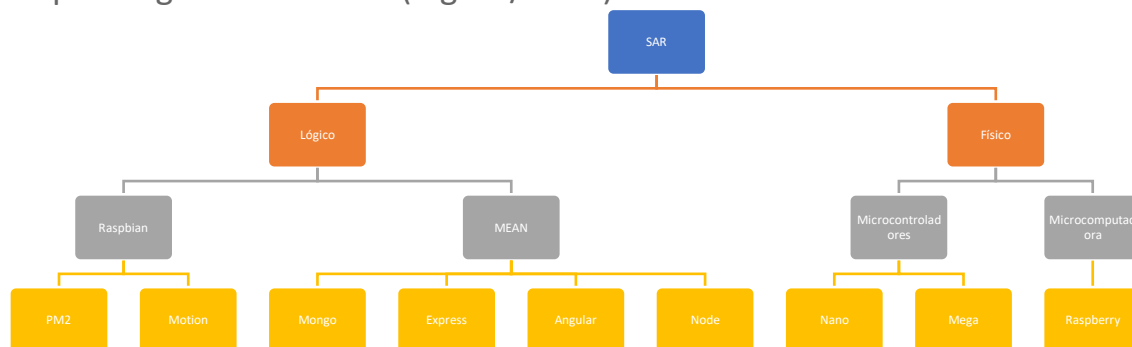
El desarrollo del SAR se descompone en varios niveles de capas. Por un lado, existen dos esquemas muy diferenciados el lógico y el físico.

El esquema físico se compone de los dispositivos electrónicos que controlan los actuadores, efectores y sensores. Además de los microcontroladores (Arduino Mega, Arduino Nano) y la microcomputadora Raspberry Pi 3.

El esquema lógico se compone del sistema operativo Raspbian y una aplicación web desarrollada bajo la arquitectura cliente/servidor respetando el conjunto de herramientas MEAN. Además contiene un administrador de servicios para Node denominado PM2, y un controlador de cámaras de video conectadas al Sistema Operativo llamado Motion.

Este esquema a su vez se encuentra organizado en dos unidades funcionales llamadas Back-end y Front-end.

Esquema general del SAR (lógico /físico)



*Ilustración 6138 – Esquema general del SAR*

### Desarrollo de la aplicación (front-end)

El front-end se encuentra desarrollado en Angular 4+, contando con los siguientes esquemas:

(Pegar esquemas del Compodoc)

Al conectarnos al servidor, desde un cliente (Browser) nos descarga la aplicación embebida, y la comunicación con el servidor sucede a través de Json.

## Desarrollo del servidor (back-end)

El back-end, construido en Node y Express, almacena los estáticos del front-end y los suministra al conectarse un cliente. Los directorios del proyecto son:  
COLOCAR DIRECTORIOS

Express filtra con las rutas **AGREGAR RUTAS** resolviendo y respondiendo las distintas consultas realizadas por la App en Angular.



*Ilustración 63 - Logo PM2*

Node es administrado por PM2 (administrador de procesos para JavaScript) el cual inicia el servidor, automáticamente al arrancar Raspbian, controlando y monitoreándolo.

**MOSTRAR PM2 EN FUNCIONAMIENTO**

## Esquema de la arquitectura lógica

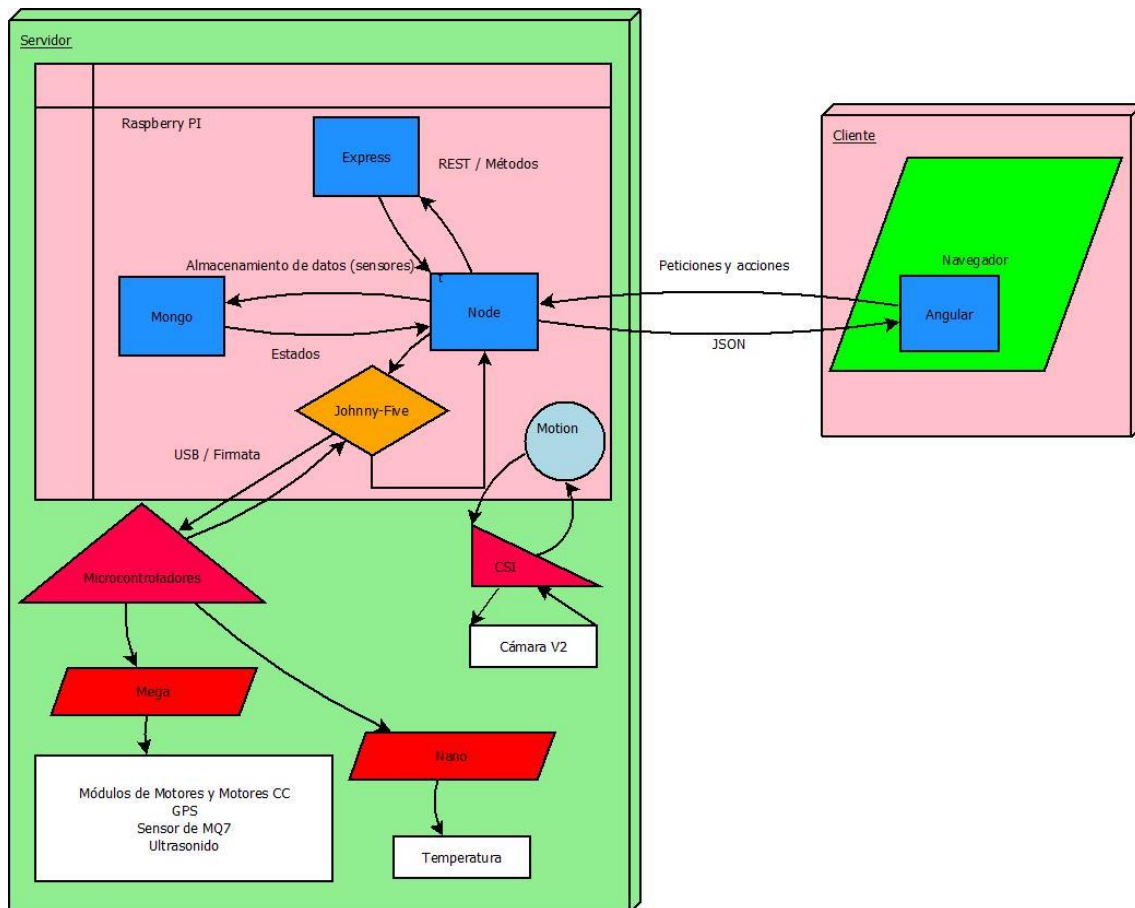


Ilustración 64 - Arquitectura lógica del SAR

## Capturas de pantalla de la aplicación

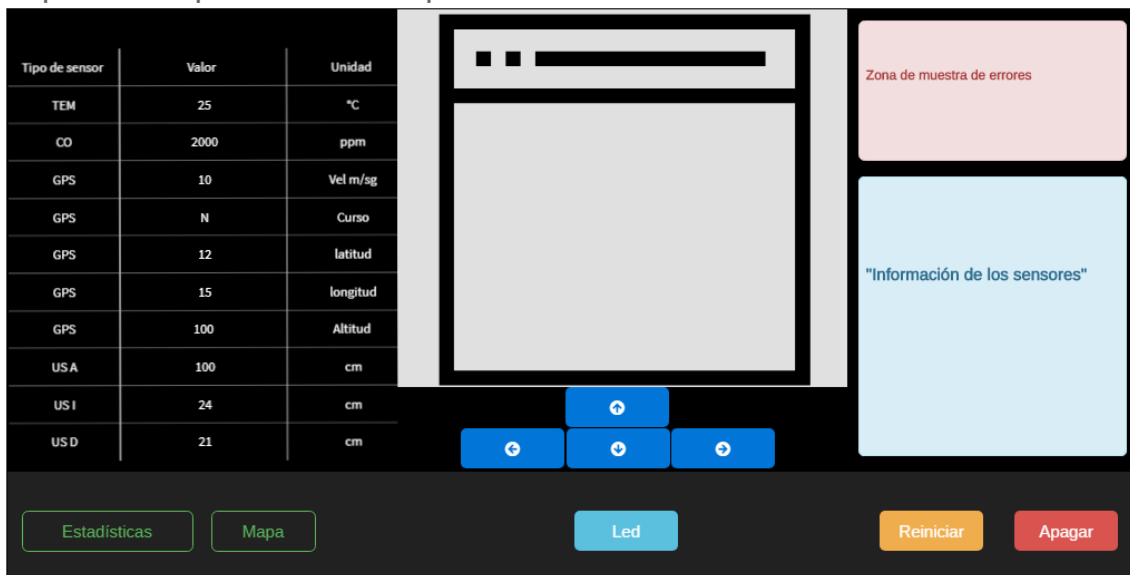


Ilustración 65 - Aplicación Web



Funcionamiento de la App

**FALTA FINALIZARLA**

Como se construyen las estadísticas

**FALTA DETERMINAR LAS MUESTRAS**

Parámetros de configuración

Dentro de esta sección agregar MOTION, PM2, Ad-hoc

**FALTA FINALIZARLA**

## Referencias bibliográficas

### Capítulo 1

Arduino (s.f.). En *Wikipedia*. Recuperado el 4 de marzo del 2017 de <https://es.wikipedia.org/wiki/Arduino>

Robótica (s.f.). En *Wikipedia*. Recuperado el 4 de marzo del 2017 de <https://es.wikipedia.org/wiki/Rob%C3%B3tica>

Aplicación móvil (s.f.). En *Wikipedia*. Recuperado el 4 de marzo del 2017 de [https://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_m%C3%B3vil](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil)

Enrique, C. (2016). Aprendiendo Arduino. Obtenido de <https://aprendiendoarduino.wordpress.com>

Inteligencia Artificial un enfoque moderno. Person. Stuart Russell, Peter Norving 2da Ed.

### Capítulo 2

<https://es.wikipedia.org/wiki/Rob%C3%B3tica>

<https://es.wikipedia.org/wiki/Robot>

<http://www.robotgroup.com.ar/>

<http://www.maestrosdelweb.com/mejores-plataformas-de-hardware-para-proyectos-diy/>

*Avances en robótica y visión por computador / coordinador, José André Somolinos Sánchez - Cuenca: Ediciones de la Universidad de Castilla - La mancha, 2002*

*Robótica - Manipuladores y robots móviles - Aníbal Ollero Baturone, 2001*

<http://www.educacotic.es/blog/robotica-educativa-con-arduino-en-el-aula-de-es-eso-incubegg-kubo-e-izar-qalaktik-mertxe-i-badiola>

<http://blogs.upm.es/observatoriogate/2017/02/01/arduino-en-la-programacion-y-robotica-educativa/>

[https://www.elconfidencial.com/tecnologia/2015-11-01/raspberry-pi-arduino-como-sacarle-partido-a-los-mini-ordenadores-low-cost\\_1076718/](https://www.elconfidencial.com/tecnologia/2015-11-01/raspberry-pi-arduino-como-sacarle-partido-a-los-mini-ordenadores-low-cost_1076718/)

[http://www.eldiario.es/turing/BBC\\_micro-bit-utilizaran-escolares-Reino-Unido\\_0\\_411209780.html](http://www.eldiario.es/turing/BBC_micro-bit-utilizaran-escolares-Reino-Unido_0_411209780.html)

<https://www.linuxadictos.com/raspberry-pi-moodle-una-plataforma-barata-e-learning.html>

<http://misionesonline.net/2017/11/10/llegaron-los-kit-didacticos-trabajar-la-ensenanza-robotica-las-escuelas-misioneras/>

Agregar en Beagle Bone

<https://www.digikey.com/es/articles/techzone/2013/sep/beaglebone-black-brings-arduino-style-connectivity-simplicity-to-embedded-linux>

### Capítulo 3

<https://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino>

<https://aprendiendoarduino.wordpress.com/2015/03/22/que-es-el-hardware-libre/>

<https://es.wikipedia.org/wiki/Arduino>  
<https://es.wikipedia.org/wiki/Processing>  
<https://es.wikipedia.org/w/index.php?title=Wiring&oldid=98682099>  
 30 proyectos con Arduino – Simon Monk – Editorial Estribor  
<https://es.wikipedia.org/wiki/Actuador>  
<https://es.wikipedia.org/wiki/Sensor>

## Capítulo 4

[https://es.wikipedia.org/wiki/Raspberry\\_Pi](https://es.wikipedia.org/wiki/Raspberry_Pi)  
<https://www.raspberrypi.org/>  
<http://www.prometec.net/indice-raspberry-pi/>  
<https://www.raspberrystore.com/accessories-raspberry-pi.php>  
<https://raspberryparatorpes.net/>  
<https://rasberry-pi.xyz/>

## Capítulo 5

<https://www.consumidor.ftc.gov/articulos/s0018-aplicaciones-moviles-que-son-y-como-funcionan>  
[https://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_m%C3%B3vil](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil)  
[https://es.wikipedia.org/wiki/Sistema\\_operativo\\_m%C3%B3vil](https://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil)  
<http://appdesignbook.com/es/contenidos/las-aplicaciones/>  
<https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>  
 melquisedec cruz gtz  
<https://www.yeeply.com/blog/apps-en-cualquier-dispositivo-desarrollo-de-aplicaciones-multiplataforma/>  
<https://developer.android.com/studio/intro/index.html?hl=es-419>  
[https://es.wikipedia.org/wiki/App\\_Inventor](https://es.wikipedia.org/wiki/App_Inventor)  
<https://software.intel.com/es-es/intel-xdk>  
[https://es.wikipedia.org/wiki/Intel\\_XDK](https://es.wikipedia.org/wiki/Intel_XDK)  
<http://www.phonegapspain.com/que-es-y-como-empezar-con-ionic-framework/>  
<https://www.desarrolloweb.com/articulos/que-es-ionic2.html>  
[https://es.wikipedia.org/wiki/Apache\\_Cordova](https://es.wikipedia.org/wiki/Apache_Cordova)  
<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>  
<http://es.discovermeteor.com/chapters/introduction/>  
[http://docs.meteor.com/?\\_ga=2.115577542.131333140.1496787062-695334936.1496349438#/full/](http://docs.meteor.com/?_ga=2.115577542.131333140.1496787062-695334936.1496349438#/full/)  
<https://guide.meteor.com/mobile.html>  
<https://developer.android.com/studio/index.html?hl=es-419>  
<https://guide.meteor.com/mobile.html#introduction>

## Capítulo 6

<https://es.wikipedia.org/wiki/MEAN>  
<https://www.campusmvp.es/recursos/post/Que-es-el-stack-MEAN-y-como-escoger-el-mejor-para-ti.aspx>  
<https://es.wikipedia.org/wiki/JSON>  
[https://es.wikipedia.org/wiki/Angular\\_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))

<https://www.cloudexperto.com/academia/angular2-framework/curso-de-angular-2/>

## Capítulo 7

<http://johnny-five.io/>

<https://github.com/rwaldron/johnny-five>

<https://aprendiendoarduino.wordpress.com/2016/03/06/firmata/>

<http://untitled.es/arduino-nodejs-johnny-five/>

## Capítulo 8

<https://es.wikipedia.org/wiki/Python>

<https://aprendiendoarduino.wordpress.com/2016/03/06/firmata/>

[http://elinux.org/RPi\\_BCM2835\\_GPIOs](http://elinux.org/RPi_BCM2835_GPIOs)

<https://www.raspberrypi.org/documentation/usage/python/>

## Capítulo 11

<http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>

[https://en.wikipedia.org/wiki/PM2\\_\(software\)](https://en.wikipedia.org/wiki/PM2_(software))