



Universidad Michoacana de
San Nicolás de Hidalgo



Facultad de Ingeniería Eléctrica

DISEÑO Y CONSTRUCCIÓN DE SISTEMA DE
EXPLORACIÓN Y RECONOCIMIENTO OPERADO
MEDIANTE WIFI

TESIS

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN
ELECTRÓNICA

PRESENTA:

ROBERTO ENRIQUE CARREÓN RANGEL

Asesor de tesis:

M.C. FÉLIX JIMÉNEZ PÉREZ

MORELIA MICHOCÁN, OCTUBRE 2021

Agradecimientos

A mis padres Roberto y Elia, a quienes todo se los debo, por darme siempre su apoyo incondicional en el aspecto emocional y económico, por inculcarme sus valores, pero sobre todo por sus enseñanzas y su amor. Sin su ayuda yo no estaría donde estoy ahora, gracias.

A mi hermana Yuritzi, por escucharme siempre que necesitaba conversar, por sus palabras y comprensión.

A mis abuelos y a toda mi familia, agradezco su compañía, por siempre ayudarme, por su cariño y por creer en mí.

A mis amigos con quienes he crecido mutuamente, por sus consejos y por ayudarme a llegar hasta aquí.

A mis maestros, la mayoría fueron mentores excepcionales, que me motivaron a seguir por este camino del saber y que sirvieron de escalones para este logro.

A mi profesor de la primaria Eduardo, el cual fue el primero que me motivo al estudio, que creyó en mí y con el cual aprendí demasiado.

A mi maestro de trabajo y amigo Víctor, que me proporcionó un empleo ayudándome a continuar con mis estudios, por sus enseñanzas y su apoyo.

A mi maestro y asesor Félix, sin el cual nada de esto sería posible, por su magnífica manera de enseñar, por el excelente guía que fue para mí, su gran motivación que inculca a todos sus alumnos y su apoyo incondicional, muchas gracias.

Y por último a esta gran facultad y universidad, que me abrió las puertas al conocimiento, me dejó ver un nuevo panorama y en la cual he pasado momentos que llevaré siempre presente en la mente y corazón. Gracias por seguir con la noble causa de transmitir y generar el conocimiento.

Contenido

Agradecimientos	2
Contenido	3
Resumen	7
Abstract	9
Lista de figuras	10
Lista de tablas	13
1 Capítulo 1. Introducción	15
1.1 Antecedentes	17
1.1.1 Robótica	17
1.1.2 Vehículos astro-espaciales (Rovers)	17
1.1.3 Robots de rescate	17
1.2 Descripción del problema	19
1.3 Solución propuesta	20
1.4 Objetivos	20
1.4.1 Objetivo general	20
1.4.2 Objetivos particulares	20
1.5 Justificación	21
1.6 Descripción de capítulos	21
2 Capítulo 2. Marco Teórico	23
2.1 Introducción	23
2.2 Robot móvil	23
2.2.1 Robot: investigación espacial	23
2.3 Características de un robot móvil	24
2.3.1 Morfología del robot móvil	24
2.3.2 Sensores	27
2.3.3 Navegación	30
2.3.4 Sistema de control y procesamiento	30
2.4 Comunicaciones inalámbricas	33
2.4.1 Radiofrecuencia	33
2.4.2 Comunicaciones con Protocolo estándar	36
2.5 Red de computadoras	39

2.5.1	Protocolos de redes.....	41
2.5.2	Arquitectura “cliente-servidor”	41
2.5.3	Páginas Web	44
3	Capítulo 3. Diseño y descripción de Hardware	48
3.1	Raspberry Pi	48
3.1.1	Modelos y especificaciones.....	49
3.1.2	RPI modelo 3 B	50
3.2	Sensores	52
3.2.1	Sensor de Temperatura y humedad: DHT11.....	52
3.2.2	Sensor de distancia: HC-SR04.....	54
3.2.3	Cámara	56
3.3	Chasis y sistema locomotor.....	58
3.3.1	Motores de CD	58
3.3.2	Servomotores	61
3.3.3	Módulo Dual Driver de puente completo: L298N	64
3.3.4	Optoacopladores.....	65
3.3.5	Chasis.....	66
3.4	Batería (Acumulador).....	68
3.4.1	Principio de funcionamiento	68
3.4.2	Parámetros de un acumulador.....	69
3.4.3	Batería Steren® BR-1207	69
4	Capítulo 4. Diseño y descripción de software	71
4.1	Diseño de sitio web e interfaz de usuario (Front-end).....	72
4.1.1	Estructura general de la página.....	73
4.1.2	Elementos de la página	74
4.2	Diseño y funcionamiento del servidor (Back-end)	78
4.2.1	Servidor de video <i>Motion</i>	79
4.2.2	Manejo de peticiones para envío de datos	80
4.3	Programación en JavaScript usando Node.js	80
4.3.1	Node Package Manager (NPM)	81
4.3.2	Lectura de sensores.....	83

4.3.3	Activación de actuadores mediante GPIO.....	84
5	Capítulo 5. Pruebas y resultados.....	86
5.1	Pruebas de Hardware.....	87
5.1.1	Carga y descarga de batería	87
5.1.2	Alcance de transmisión entre el robot y el usuario	87
5.1.3	Movimiento en exteriores.....	88
5.2	Pruebas de Software	90
5.2.1	Inicio de sesión	90
5.2.2	Funcionamiento de interfaz	92
5.2.3	Gráfica de variables	93
5.2.4	Servidor de video.....	94
5.2.5	Desempeño de sitio web.....	95
6	Conclusiones.....	97
6.1	Objetivos alcanzados.....	97
6.1.1	Sistema Locomotor	97
6.1.2	Manejo de sensores	97
6.1.3	Visualización del terreno.....	98
6.1.4	Interfaz de usuario	98
6.1.5	Construcción e integración.....	98
6.2	Trabajos futuros	99
	Apéndices	101
	Apéndice 1: Instalación de servidor de video Motion	101
	Apéndice 2: Instalación y ejecución de aplicación en Node.js.....	104
	Apéndice 3: Diagrama Electrónico	107
	Bibliografía	113

Resumen

En este trabajo de tesis se expone el diseño, desarrollo y construcción de un sistema de exploración basado en un robot móvil tele-operado, controlado inalámbricamente, en específico se usa la tecnología WIFI. El sistema de exploración tiene como objetivo: búsqueda, rescate y evaluación del entorno (detección de daños a estructuras) en lugares de riesgo o de acceso limitado, tales como derrumbes, zonas volcánicas o radioactivas, ductos de ventilación, cuevas, etc.

El robot se divide en dos etapas principales, la etapa de hardware y la de software. La parte de hardware se compone de un chasis y sistema locomotor, la etapa de potencia, un conjunto de sensores, una cámara y un sistema de procesamiento y control. Los sensores empleados miden las variables de temperatura, humedad y distancia; para el sistema de control se usó la computadora de placa única Raspberry Pi.

La etapa de software se compone principalmente de una aplicación web donde se implementa la interfaz de usuario, esta se desarrolla y programa mediante HTML, CSS y JavaScript. La interfaz está diseñada para el control y monitoreo del sistema robótico en tiempo real, cuenta con visualización de video, lectura de los sensores y el control de los actuadores.

Se trabaja bajo la arquitectura cliente-servidor mediante el protocolo HTTP, del lado del servidor se encuentra un servicio programado en JavaScript ejecutándose en Node.js que se encarga de servir las solicitudes del cliente, administrar el flujo de información y de manejar los sensores y actuadores.

Palabras Clave

Robot móvil, Tele-operado, Rover, Aplicación Web, IoT, WIFI, HTML, JavaScript, HighCharts, Motion, Node.js, Express, Raspberry Pi, DHT11, HC-SR04, Cámara.

Abstract

This thesis shows the design, development and construction of an exploration system based on a tele-operated mobile robot, wirelessly controlled, specifically WIFI technology is used. The exploration system aims to: search, rescue and evaluation of the environment (detection of damage to structures) in places of risk or limited access, such as landslides, volcanic or radioactive areas, ventilation ducts, caves, etc.

The robot consists of two main stages, the hardware stage and the software stage. On the hardware side, it is integrated for a chassis and locomotion system, power electronics stage, a set of sensors, a camera and a processing and control system. The sensor set measures the variables of temperature, humidity and distance; The Raspberry Pi single board computer is used for the control system.

The software stage consists mainly of a web application where the user interface is implemented, it was developed and programmed using HTML, CSS and JavaScript. The interface is designed for the control and monitoring of the robotic system in real time, it has video streaming, sensor reading and actuator control.

It works under the client-server architecture through HTTP protocol, on the server side there is a service programmed in JavaScript running on Node.js that is responsible for serving the client's requests, managing the information flow and also handling the sensors and actuators.

Keywords

Mobile robot, Tele-operated, Rover, Web app, IoT, WIFI, HTML, JavaScript, HighCharts, Motion, Node.js, Express, Raspberry Pi, DHT11, HC-SR04, Camera.

Lista de figuras

Figura 1-1 Atlas, robot humanoide [2]	15
Figura 1-2 Clasificación de robots. a) Androide [3], b) Móvil [4], c) Zoomórfico [5], d) Poli-articulado [6]	16
Figura 1-3 Diferentes vehículos astroespaciales [9]	18
Figura 1-4 FinDER V2 [11]	18
Figura 1-5 a) Rover: WiFi Remote Control Robot with WiFi Video Camera [12], b) SuperDroid HD2 Robot - WiFi Control [13]	19
Figura 1-6 Cinturón Circumpacífico [15] y regiones sísmicas en México [14]	19
Figura 2-1 Robot interior [18] (izquierda) y exterior [19] (derecha)	25
Figura 2-2 Robots terrestres. (a) Hermes (Robosoft). (b) Journey (SMU). (c) Scour (USF). [16]	26
Figura 2-3 Robot acuático (NSF) [20] y robot aéreo [21]	26
Figura 2-4 Sistema de tracción y dirección sobre todos los ejes. (Diferencial) [16]	27
Figura 2-5 Sistema de tracción y dirección en ejes independientes. (Automóvil) [16]	27
Figura 2-6 Tipos de sensores [23]	28
Figura 2-7 Sistemas embebidos: a) Smartphone [26], b) FPGA [27], c) Microcontrolador [28] y d) SBC [29]	31
Figura 2-8 Estrategias de control para robots móviles. [16]	32
Figura 2-9 Modulación ASK [34]	35
Figura 2-10 Modulación FSK [34]	35
Figura 2-11 Modulación PSK [34]	36
Figura 2-12 Modo infraestructura [37]	38
Figura 2-13 Modo 'ad hoc' [37]	39
Figura 2-14 Modelo OSI [39]	40
Figura 2-15 Arquitectura cliente-servidor [43]	42
Figura 2-16 Navegadores web [44]	44
Figura 3-1 Diagrama de bloques de Hardware	48
Figura 3-2 Raspberry Pi 2 model B (2015) [51]	49
Figura 3-3 RPI 1 Modelo A+ (izquierda), RPI 2 Modelo B (centro) y RPI 3 Modelo B (derecha) [52]	50
Figura 3-4 Diagrama de bloques (izquierda) [51] y localización de componentes en la placa (derecha) [53]	51
Figura 3-5 RPI GPIO [54]	51
Figura 3-6 RPI GPIO Software [54]	51
Figura 3-7 DHT11 [56]	52
Figura 3-8 a) Diagrama de comunicación y b) diagrama de codificación [57]	53
Figura 3-9 Trama de datos DHT11 [56]	54
Figura 3-10 Sensor ultrasónico HC-SR04 [59]	54
Figura 3-11 Funcionamiento de un sonar [60]	55
Figura 3-12 Cronograma de medición HC-SR04 [59]	55
Figura 3-13 RPI Camera Module [63]	57
Figura 3-14 Motor de CD [64]	58
Figura 3-15 Componentes del motor de CD [64]	59
Figura 3-16 Motor Maxon 278072 [65]	59
Figura 3-17 Circuito de prueba de corriente	60
Figura 3-18 Corriente de arranque del motor	61
Figura 3-19 Corriente en estado estable	61
Figura 3-20 Servomotor [67]	62
Figura 3-21 Servomotor: Clasificación común de cables por color [67]	62
Figura 3-22 Diagrama de bloques del servomotor	62

Figura 3-23 PWM para diferentes ángulos en un servomotor [67]	63
Figura 3-24 Servomotor sg90 [69]	63
Figura 3-25 Operación servomotor sg90 [69]	64
Figura 3-26 Módulo L298N [70]	64
Figura 3-27 Alimentación L298N [70]	65
Figura 3-28 Optoacoplador: diagrama general [71]	65
Figura 3-29 Sharp PC817: a) diagrama [72] y b) componente [73]	66
Figura 3-30 a) Chasis de acrílico [74] y b) chasis de aluminio [75]	67
Figura 3-31 Estructura del chasis	67
Figura 3-32 Chasis-sistema locomotor	68
Figura 3-33 a) Acopladores [76] y b) llantas usadas	68
Figura 3-34 Batería BR-1207	69
Figura 4-1 Diagrama de estructura general	71
Figura 4-2 Diagrama de flujo de información	72
Figura 4-3 Funcionamiento de CSS [80]	73
Figura 4-4 Plantilla de página web	73
Figura 4-5 Plantilla de interfaz de usuario	74
Figura 4-6 Estructura de etiquetas de elementos HTML [81]	75
Figura 4-7 Tag <iframe>	75
Figura 4-8 Ventana de video	76
Figura 4-9 Pad direccional	77
Figura 4-10 Visualización de mediciones	77
Figura 4-11 Gráfica de variables - "High Charts"	78
Figura 4-12 Visualización de video	79
Figura 4-13 AJAX: funcionamiento [85]	80
Figura 4-14 Ejemplo de servidor HTTP [86]	81
Figura 4-15 Diagrama de conexión con sensor DHT11 [88]	83
Figura 4-16 Código para lectura de sensor DHT [89]	83
Figura 4-17 Diagrama de conexión sensor HC-SR04 [90]	84
Figura 4-18 a) Código: encender led mediante PWM [90] y b) diagrama de conexión [91]	84
Figura 5-1 Diagrama Integrado	86
Figura 5-2 Adaptador de red inalámbrica USB [92]	87
Figura 5-3 Conexión: Smartphone – Interfaz de red Raspberry	88
Figura 5-4 Conexión: Router - Interfaz de red USB	88
Figura 5-5 Terreno con vegetación	88
Figura 5-6 Terreno rocoso	89
Figura 5-7 Plano inclinado	89
Figura 5-8 Página Inicial	90
Figura 5-9 Cuadro de dialogo "Inicio de sesión"	90
Figura 5-10 Usuario o contraseña sin ingresar	91
Figura 5-11 Usuario y contraseña incorrecta	91
Figura 5-12 Error límite máximo de usuarios	91
Figura 5-13 Terminal: Usuario y contraseña correctos	92
Figura 5-14 Página principal: Interfaz de usuario	92
Figura 5-15 Peticiones HTTP de control	93
Figura 5-16 Log en terminal: peticiones HTTP	93

<i>Figura 5-17 Gráfica Dinámica</i>	94
<i>Figura 5-18 Prueba: servidor de video</i>	95
<i>Figura 5-19 Resumen de pruebas a sitio web</i>	95

Lista de tablas

<i>Tabla 2-1 Clasificación de los sensores [16].....</i>	<i>29</i>
<i>Tabla 2-2 Radiofrecuencia [33].....</i>	<i>33</i>
<i>Tabla 2-3 Estándares 802.11 [36].....</i>	<i>37</i>
<i>Tabla 3-1 Especificaciones Raspberry Pi [51].....</i>	<i>49</i>
<i>Tabla 3-2 Especificaciones RPI Camera [62]</i>	<i>57</i>

1

Capítulo 1. Introducción

Los Robots son mecanismos electrónicos programables que se han desarrollado para ayudar, colaborar o incluso reemplazar a los seres humanos, de ser necesario, en actividades muy detalladas, repetitivas, complicadas, difíciles o peligrosas. Esto se logra mediante la integración de sistemas mecánicos, eléctricos, electrónicos y computacionales dentro de un mismo dispositivo programable con control automático, llamado Robot. Actualmente se considera que “un robot es una computadora con capacidad de movimiento, capaz de desarrollar múltiples tareas de manera flexible según su programación”. [1] En la Figura 1-1 se muestra un prototipo de robot bípedo desarrollado por la compañía *Boston Dynamics* como ejemplo del alcance de la robótica.



Figura 1-1 Atlas, robot humanoide [2]

Existen diferentes tipos y clases de robots, entre ellos con forma humana, de animales, de plantas o incluso de elementos arquitectónicos, pero todos se diferencian por sus capacidades y se clasifican en 4 formas: [3] Figura 1-2

- ♦ Androides: robots con forma humana. Imitan el comportamiento de las personas, su utilidad en la actualidad es solo de experimentación. La principal limitante de este modelo es la implementación del equilibrio en el desplazamiento, pues es bípedo.

- ♦ Móviles: se desplazan mediante una plataforma rodante (ruedas); estos robots aseguran el transporte de piezas de un punto a otro.
- ♦ Zoomórficos: es un sistema de locomoción imitando a los animales. La aplicación de estos robots sirve, sobre todo, para el estudio de volcanes y exploración espacial.
- ♦ Poli-articulados: mueven sus extremidades con pocos grados de libertad. Su principal utilidad es industrial, para desplazar elementos que requieren cuidados.

En esta última se puede clasificar según su morfología en: Robots angulares o antropomórficos, robots cilíndricos, robots esféricos o polares, robots tipo SCARA, robots paralelos, robots cartesianos, entre otros. [3]

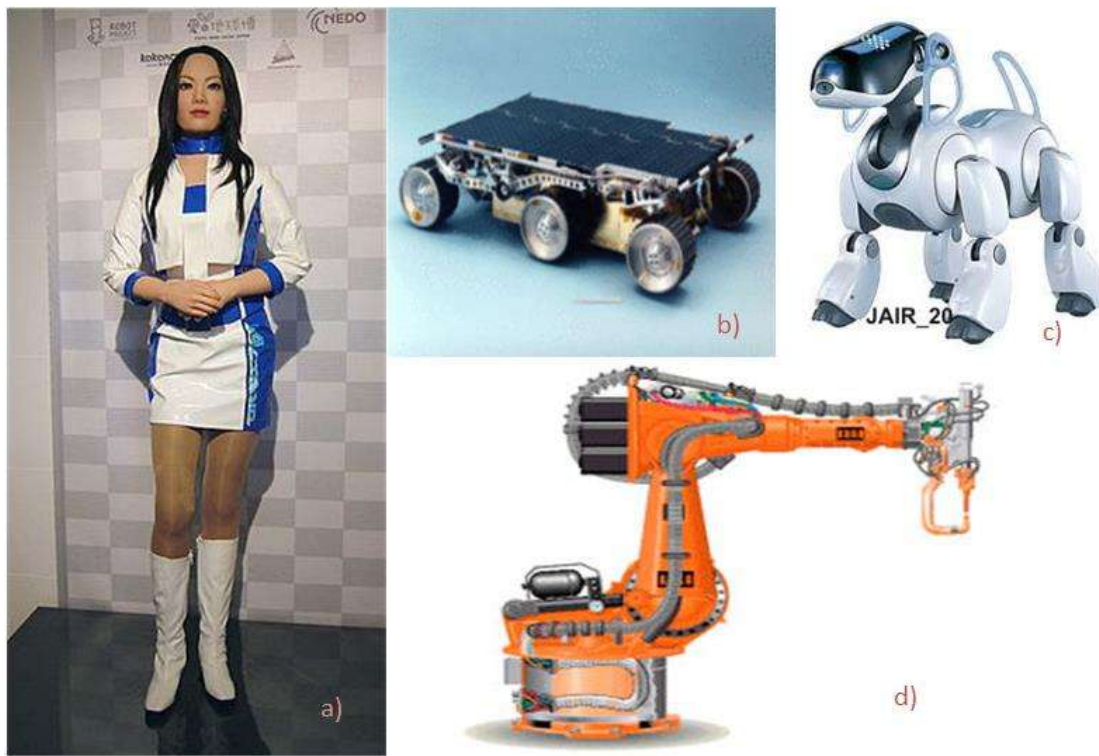


Figura 1-2 Clasificación de robots. a) Androide [3], b) Móvil [4], c) Zoomórfico [5], d) Poli-articulado [6]

Los robots móviles cuentan con gran capacidad de desplazamiento, basados en carros o plataformas y dotados de un sistema locomotor de tipo rodante. Siguen su camino por telemando o guiándose por la información recibida de su entorno a través de sus sensores. [7] Es en la categoría de robótica móvil donde se desarrolla el presente trabajo, debido a la estructura y las necesidades del proyecto.

1.1 Antecedentes

1.1.1 Robótica

En la década de 1890 el científico Nikola Tesla, inventor, entre muchos otros dispositivos, de los motores de inducción, ya construía vehículos controlados a distancia por radio. Tesla fue un visionario que escribió sobre mecanismos inteligentes tan capaces como los humanos. [8]

El término Robótica se refiere a la ciencia o arte relacionada con la inteligencia artificial (para razonar) y con la ingeniería mecánica (para realizar acciones físicas sugeridas por el razonamiento). Este término fue acuñado en 1942 por el bioquímico, escritor y divulgador científico norteamericano de origen ruso Isaac Asimov en su novela corta Runaround. [8]

Durante la década de los 60 se hace necesaria la realimentación sensorial. En 1962, H. A. Ernst publica el desarrollo de una mano mecánica controlada por computador con sensores táctiles llamada MH-1. Este modelo evolucionó adaptándole una cámara de televisión dentro del proyecto MAC. También en 1962, Tomovic y Boni desarrollan una mano con un sensor de presión para la detección del objeto que proporcionaba una señal de realimentación al motor. [8]

Durante la década de los 70, la investigación en robótica se centra en gran parte en el uso de sensores externos para su utilización en tareas de manipulación. Es también en estos años cuando se consolida definitivamente la presencia de robots en las cadenas de montaje y plantas industriales en el ámbito mundial. [8]

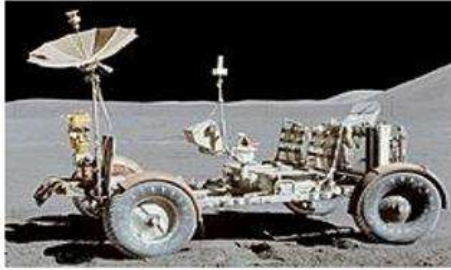
1.1.2 Vehículos astro-espaciales (Rovers)

Un Rover Planetario es un vehículo móvil de exploración espacial diseñado para recorrer la superficie de un planeta, luna, asteroide o cualquier otro cuerpo espacial. Requieren para su operación de enlaces de radio con un control remoto semi-autónomo, y con un tiempo de retardo proporcional a su distancia de la Tierra. [1] Las tortugas motorizadas diseñadas en los años cincuenta, fueron las precursoras y sirvieron de base a los estudios sobre inteligencia artificial desarrollados entre 1965 y 1973 en la Universidad de Stanford. Los vehículos operacionales actuales van del peso del microrovers de 5 kilogramos que se diseñan para exploración planetaria, a 3,000 kilogramos camiones militares, capacidad de navegación autónoma. Las actividades actuales incluyen el desarrollo de conducta autónoma, control de la conducta de microrover para ciencia y adquisición de la muestra de la superficie de la Luna y Marte. [7] Figura 1-3.

Los robots también pueden explorar planetas distantes. La sonda espacial no tripulada Galileo, de la NASA, viajó a Júpiter en 1996 y realizó tareas como la detección del contenido químico de la atmósfera joviana. [8]

1.1.3 Robots de rescate

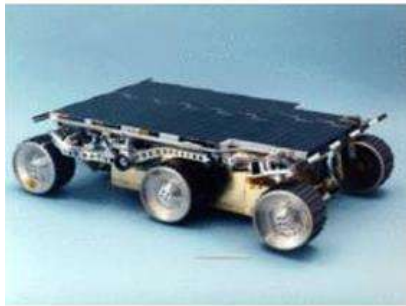
Otros vehículos incluyen robots tele-operados para la investigación de materiales peligrosos, derramamientos y desastres. Un claro ejemplo es el *FinDER V2* Figura 1-4 el cual es un robot de rescate que ha sido creado por un grupo de especialistas llamado Open Robotics Group (ORG), de la Facultad de Ingeniería de la UNAM.



Astromóvil Lunar del Apolo 15 



El astromóvil Lunokhod 



El Mars rover Sojourner 



Chang'e 3 

Figura 1-3 Diferentes vehículos astroespaciales [9]

FinDER V2 tiene un sistema usado en los videojuegos que cumple la función de los ojos y los oídos, se llama Play Station Eye; además de ser económico, detecta hasta 120 cuadros por segundos, puede ubicar de dónde proviene un sonido, calcular la profundidad a la que se encuentran los objetos y hacer mapas tridimensionales. Para localizar a posibles personas atrapadas, este rescatista mecánico utiliza una serie de sensores que identifican la temperatura de un cuerpo humano, y un sensor de dióxido de carbono (CO2) que determina si la persona está respirando. FinDER V2 se mueve por medio de una red WiFi; sin embargo, este tipo de comunicación no es suficiente si se utiliza en un edificio colapsado, por ello cuenta con un cordón umbilical, es decir, un cable para transmitir información y jalar a la máquina en caso de que dejara de funcionar. [10]

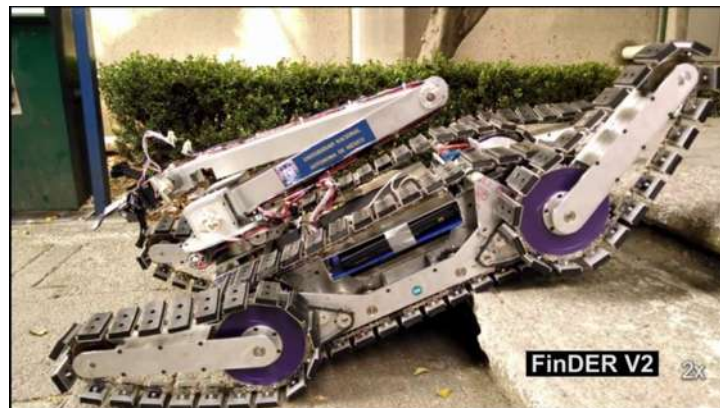


Figura 1-4 FinDER V2 [11]

En la actualidad hay diferentes opciones de robots móviles en el mercado, las características van desde un prototipo DIY (Do It Yourself), por sus siglas en inglés, hasta robots profesionales y probados mediante diferentes evaluaciones. Los costos de estos van desde los 100 dólares, hasta llegar a alcanzar los 10,000 dólares, respectivamente. Figura 1-5



Figura 1-5 a) Rover: WiFi Remote Control Robot with WiFi Video Camera [12], b) SuperDroid HD2 Robot - WiFi Control [13]

1.2 Descripción del problema

Desde la antigüedad y hasta hoy en día, en todo el planeta se han presentado desastres naturales con el paso del tiempo. La República Mexicana está situada en una de las regiones sísmicamente más activas del mundo, enclavada dentro del área conocida como el Cinturón Circumpacífico o Cinturón de Fuego donde se concentra la mayor actividad sísmica del planeta. [14]

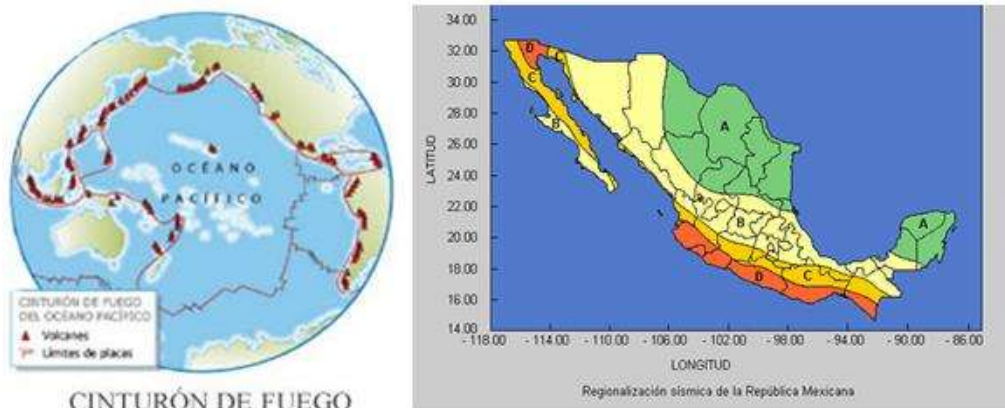


Figura 1-6 Cinturón Circumpacífico [15] y regiones sísmicas en México [14]

Por tanto, existe un riesgo latente en nuestro país, los sismos. Gracias a experiencias anteriores se tiene conocimiento de las consecuencias que conlleva un desastre de esta magnitud, es decir, los

derrumbes que pueden ocurrir en las grandes urbes y la afectación a estructuras arquitectónicas. Una vez pasado el sismo se necesita evaluar la integridad de las estructuras o más importante, desplegar equipos de búsqueda y rescate para posibles personas atrapadas. Estas acciones se deben realizar lo más rápida y eficientemente posibles para que tengan un mayor éxito.

Sin embargo, estas tareas al ser realizadas por personas se convierten en situaciones de muy alto riesgo, y tienden a ser lentas debido al cuidado que se tiene que mantener.

1.3 Solución propuesta

En la actualidad la tecnología está inmersa en nuestra vida cotidiana, desde el momento en el que contamos con un teléfono móvil o una computadora portátil para interactuar en Internet, hasta algo tan simple como la televisión o casi cualquier electrodoméstico. Debido a que esta tecnología se encuentra al alcance de nuestras manos, se debe utilizar para las necesidades de la sociedad, y es justamente lo que se realizó en este trabajo.

Se propuso diseñar un robot móvil que ayude en las labores de evaluación de daños y de rescate, al ser un vehículo no tripulado se evitan las situaciones de riesgo a las que se exponen las personas. Este robot explorador consta de un chasis y un sistema locomotor que le permite moverse a través del terreno, además cuenta con una cámara con la cual los expertos pueden evaluar los daños y sirve para la tarea de búsqueda y rescate. Aunado a esto el robot puede medir parámetros físicos del entorno como temperatura, humedad y distancia a objetos cercanos, para una mejor evaluación del terreno donde se encuentra. Todos estos datos se pueden consultar en una página web diseñada para la visualización de los parámetros y control del robot móvil. Los datos se envían inalámbricamente mediante WIFI.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un robot móvil para realizar tareas de búsqueda, rescate y evaluación de daños en desastres naturales o en exploración de lugares de riesgo como volcanes e incluso el espacio exterior.

1.4.2 Objetivos particulares

- Construir un sistema locomotor simple para el desplazamiento del robot.
- Integrar una serie de sensores para la medición de variables físicas.
- Integrar una cámara para la visualización del terreno.
- Diseñar una página web para la visualización de parámetros y control del robot en tiempo real.
- Construir e integrar todas las etapas del robot móvil para su correcto funcionamiento.

1.5 Justificación

Como se mencionó anteriormente, debido a que nuestro país se encuentra en una ubicación de riesgo, específicamente en fenómenos sísmológicos, se necesitan equipos de evaluación de daños y de rescate, considerando la seguridad de los equipos y minimizando las situaciones de riesgo a las que se someten.

Por esta razón se decidió desarrollar un prototipo de robot explorador para desempeñar esas tareas, y como objetivo principal salvaguardar la vida y la integridad de las personas.

Con este dispositivo se espera disminuir las situaciones de riesgo en los desastres naturales, además de aumentar las probabilidades de realizar una búsqueda y rescatar exitosamente a la(s) persona(s).

1.6 Descripción de capítulos

Los capítulos que se desarrollan en el presente trabajo se pueden resumir como se muestra a continuación:

Capítulo 1. Se plantea la problemática y la propuesta de solución, además de una introducción que engloba los antecedentes y la justificación del trabajo.

Capítulo 2. Marco teórico, donde se describen todos los componentes y temas relacionados con el proyecto para tener un contexto del trabajo realizado.

Capítulo 3. Se detalla la descripción de la parte física del proyecto (Hardware), es decir, su diseño y construcción tanto mecánica como electrónicamente.

Capítulo 4. En este capítulo se describe la parte lógica y de programación del proyecto (Software), así como el diseño de la página web e interfaz de usuario, la estructura y los elementos de los cuales estará conformada.

Capítulo 5. En este apartado se exponen las pruebas realizadas al prototipo y se resaltan los resultados obtenidos de este.

Capítulo 6. Las conclusiones están descritas en el último capítulo como parte de la culminación del proyecto.

2

Capítulo 2. Marco Teórico

2.1 Introducción

En este capítulo se presenta el contexto teórico de los temas abordados en el trabajo, es decir, las tecnologías involucradas con la robótica y la descripción del robot móvil, las tecnologías inalámbricas y los conceptos de redes de computadoras y protocolo de internet, así como los lenguajes usados para el diseño y programación de sitios web. Con el objetivo que el lector obtenga todas las herramientas para la mejor comprensión del proyecto.

2.2 Robot móvil

Los robots móviles son dispositivos de transporte automático, es decir, una plataforma mecánica dotada de un sistema de locomoción capaz de navegar a través de un determinado ambiente de trabajo, dotado de cierto nivel de autonomía para su desplazamiento portando cargas. Sus aplicaciones pueden ser muy variadas y siempre están relacionadas con tareas que normalmente son riesgosas o nocivas para la salud humana, en áreas como la agricultura, en el transporte de cargas peligrosas o en tareas de exploración solitarias o cooperativas junto a otros vehículos no tripulados. Ejemplos clásicos son el traslado y acopio de materiales, las tareas de mantenimiento en reactores nucleares, la manipulación de materiales explosivos, la exploración subterránea, etc. [16]

Como aplicación especial y campo posible de investigación cabe destacar que los robots teleoperados son empleados en situaciones que implican peligro para los seres humanos; ellos han sido usados en el seguimiento de la limpieza de incidentes como los ocurridos en las plantas nucleares de Chernóbil y Three Mile Island. [17]

2.2.1 Robot: investigación espacial

La misión Pathfinder, que viajó a Marte en 1997, llevó el que se puede considerar el robot móvil autónomo de mayor éxito en la historia. El robot Sojourner concentró la atención de millones de personas en su exploración del planeta rojo. Ejecutó fielmente los comandos enviados desde

controladores en el laboratorio de Propulsión a Cohete en California; también podía tomar acciones por sí mismo, manteniéndose alerta de obstáculos y abismos. El equipo fue el resultado del desarrollo de varios prototipos tempranos, y fue programado de acuerdo a los principios del control por comportamientos. [17]

2.3 Características de un robot móvil

Las propiedades representativas de los robots son la versatilidad y la auto-adaptabilidad. La primera se entiende como la potencialidad estructural de ejecutar tareas diversas y/o ejecutar una misma tarea de forma diversa, lo cual implica una estructura mecánica de geometría variable. La auto-adaptabilidad significa que un robot debe, por sí solo, alcanzar su objetivo a pesar de las perturbaciones imprevistas del entorno a lo largo de la ejecución de su tarea; esto supone una consciencia del entorno otorgada por la posesión de sentidos artificiales. [17]

Los robots móviles operando en grandes ambientes no estructurados deben enfrentarse con significativas incertidumbres en la posición e identificación de objetos. En compensación por tener que enfrentarse con más incertidumbres del entorno, no se espera que un robot móvil siga trayectorias o alcance su destino final con el mismo nivel de precisión que se espera de un manipulador industrial (en el orden de las centésimas de milímetro). [16]

Las prioridades para los robots móviles están firmemente orientadas en las áreas de sensado y raciocinio.

- ♦ Percepción: El robot móvil debe ser capaz de determinar la relación con su entorno de trabajo, mediante el sistema sensorial a bordo. La capacidad de percepción del robot móvil se traduce en la síntesis de toda la información provista por los sensores, con el objeto de generar mapas globales y locales del entorno de acuerdo a los diversos niveles de control.
- ♦ Razonamiento: El robot móvil debe ser capaz de decidir qué acciones son requeridas en cada momento, según el estado del robot y el de su entorno, para alcanzar su(s) objetivo(s). La capacidad de razonamiento del robot móvil se traduce en la planificación de trayectorias globales seguras y en la habilidad para modificarlas en presencia de obstáculos inesperados (control local de trayectoria) para permitirle, al robot, el cumplimiento de los objetivos encomendados.

El grado de autonomía depende en gran medida de la facultad del robot para abstraer el entorno y convertir la información obtenida en órdenes, de tal modo que, aplicadas sobre los actuadores del sistema de locomoción, garantice la realización eficaz de su tarea. [16]

Un robot operacional está constituido por cuatro sistemas relacionados entre sí: mecánico, sensorial, control y alimentación. [17]

2.3.1 Morfología del robot móvil

La mayoría de los robots móviles poseen características particulares que los hacen aptos para una determinada tarea. Es la tarea misma la que determina en una primera etapa las particularidades

estructurales del robot que van desde el tipo de rueda, el sistema de tracción y dirección y la forma física del robot. [16]

Según el área de trabajo el entorno del robot puede ser interior o exterior, esta es la característica que fija mayores restricciones sobre el robot móvil. Ejemplos clásicos de entornos interiores y exteriores son los grandes edificios públicos y los parques o bosques, respectivamente. Véase Figura 2-1 [16]



Figura 2-1 Robot interior [18] (izquierda) y exterior [19] (derecha)

2.3.1.1 Sistema locomotor

El sistema de locomoción es una de las primeras características de un robot que está condicionada por su entorno. De acuerdo a las características del mismo el robot puede ser: [16]

- ◆ Terrestre (Figura 2-2)
 - patas
 - ruedas
 - cadenas
- ◆ Acuático (Figura 2-3)
 - flotante
 - submarino
- ◆ Aéreo (Figura 2-3)

Además, el sistema de locomoción de un robot móvil está determinado en gran medida por su peso, dado que éste afecta la movilidad y define la potencia de los motores necesaria para el desplazamiento. A su vez, el peso se encuentra determinado por las siguientes partes: chasis, motores, baterías y sistema electrónico. [17]



Figura 2-2 Robots terrestres. (a) Hermes (Robosoft). (b) Journey (SMU). (c) Scour (USF). [16]



Figura 2-3 Robot acuático (NSF) [20] y robot aéreo [21]

Dentro de los sistemas de locomoción terrestre los principales sistemas de movilidad se dividen en cuatro grupos, a saber:

- ◆ Diferencial: nombre técnico dado al sistema oruga utilizado por los tanques militares; mediante sistemas de engranajes y dos motores se logran desplazamientos lentos y giros de 360°. (Figura 2-4)
- ◆ Automóvil: movilidad análoga a la empleada por los carros, utilizándose un motor de tracción y otro de dirección. (Figura 2-5)
- ◆ Triciclo: es el más empleado por experimentadores en robótica dada la facilidad de su diseño; utiliza dos motores para la movilidad y dirección, y la estabilidad se realiza utilizando ruedas locas (castres).
- ◆ Omnidireccional: es un sistema de alta tecnología que utiliza un motor para cada una de las extremidades que produzcan movilidad; requiere un alto nivel de integración de los sistemas.

En cuanto a los sistemas de engranajes utilizados para la transmisión de fuerza en los robots debe decirse que se encuentran altamente desarrollados trenes de engranajes rectos llamados moto-reductores. De estos dispositivos es necesario conocer su relación de transmisión, eficiencia y tamaño, principalmente.

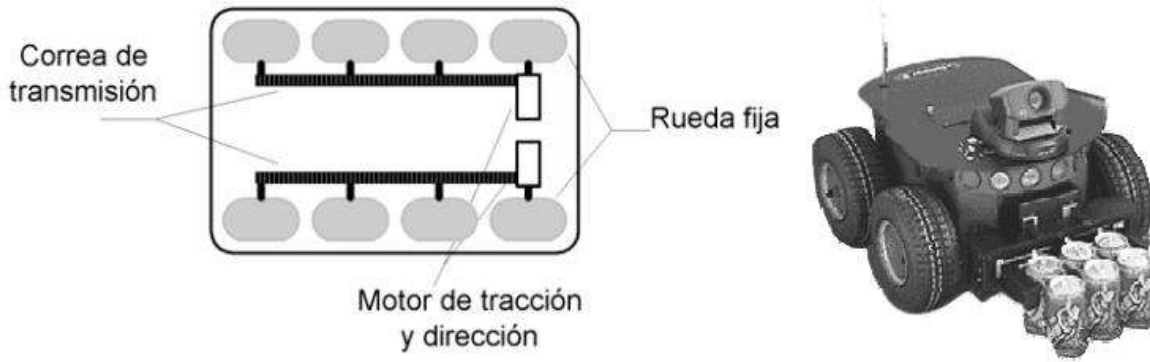


Figura 2-4 Sistema de tracción y dirección sobre todos los ejes. (Diferencial) [16]

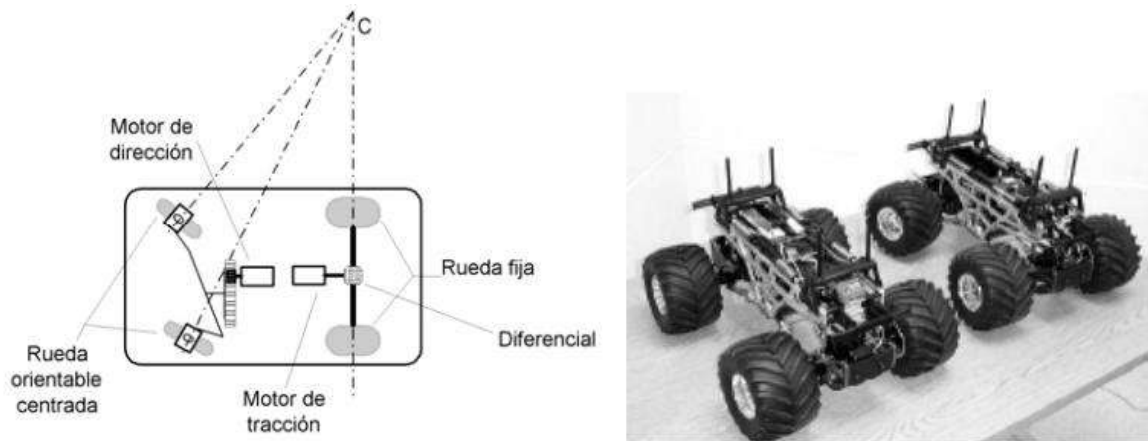


Figura 2-5 Sistema de tracción y dirección en ejes independientes. (Automóvil) [16]

2.3.2 Sensores

Un sensor es un dispositivo capaz de variar una propiedad ante magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas con un transductor en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: intensidad luminosa, temperatura, distancia, aceleración, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica, etc. [22] Algunos ejemplos de sensores se muestran en la Figura 2-6.



Figura 2-6 Tipos de sensores [23]

Para el estudio de los sensores en robótica se los ha clasificado de diversas formas, empleando diversos criterios. [16] Véase Tabla 2-1

Los sensores pueden clasificarse utilizando como criterio la interacción sensor-objeto, atendiendo a que la información pueda ser obtenida por contacto directo entre el entorno u objeto y la parte sensorial del robot, o sin mediar contacto físico alguno. De esta manera los sensores podrán clasificarse de contacto y de no contacto. Otra clasificación, se basa en la necesidad de obtener y procesar informaciones complejas a través de sensores muy desarrollados. Se podría clasificar a los sensores en elementales y complejos. Una última clasificación está basada en el hecho de si los sensores emiten o únicamente captan energía para hacer la medición y se clasifican en dos grupos:

Sensores activos: Son aquellos que emiten algún tipo de energía al medio (por ejemplo: luz o ultrasonidos) y miden su interacción con el medio. Éstos proporcionan directamente medidas de distancias al entorno. Entre los sistemas más empleados se encuentran los sonares (sensores de ultrasonido) y los escáneres (sensores láser).

Sensores pasivos: Se limitan a captar la energía existente en el medio. Entre los sistemas más empleados se encuentran las cámaras de vídeo y los sensores infrarrojos. [16]

Tabla 2-1 Clasificación de los sensores [16]

Sensores en robótica			
Según el medio relativo al robot	Según el tipo de interacción robot-objeto	Según el tipo de información	Según el principio de funcionamiento
Propioceptivos	Contacto	Elementales	Carga eléctrica
Exteroceptivos	No contacto	Complejos	Radiación luminosa
			Resistencia
			Inductancia
			Capacitancia
			Radiación térmica

Los sensores propioceptivos son sensores que miden variables internas del robot, estos sensores han sido cronológicamente los primeros sensores utilizados, encontrándose actualmente en la mayoría de los robots. Suministran la información del estado interno del robot: posición, dirección, sentido, velocidad. En contraposición los sensores exteroceptivos determinan el estado del robot en función de la percepción del entorno. [16]

El sistema de percepción de un robot móvil permite que este sea capaz de hacer frente a situaciones cambiantes del entorno, lo que exige la utilización de un sistema sensorial que suministre la información del medio. Por ello, un robot móvil rara vez va equipado con un único sensor para realizar todas sus tareas, sino que la práctica más habitual consiste en combinar dentro del sistema sensorial varios sensores que en mayor o menor medida se complementan. [16]

2.3.2.1 Características de un sensor

- ◆ Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- ◆ Precisión: es el error de medida máximo esperado.
- ◆ Linealidad o correlación lineal.
- ◆ Sensibilidad de un sensor: mínima magnitud en la señal de entrada requerida para producir una determinada magnitud en la señal de salida.
- ◆ Resolución: mínima variación de la magnitud de entrada que puede detectarse a la salida.
- ◆ Rapidez de respuesta: puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- ◆ Derivas: son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como

la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.

- ♦ Repetitividad: error esperado al repetir varias veces la misma medida.

La resolución de un sensor es el menor cambio en la magnitud de entrada que se aprecia en la magnitud de salida. Sin embargo, la precisión es el máximo error esperado en la medida. [22]

2.3.3 Navegación

Se define navegación como la metodología que permite guiar el curso de un robot móvil a través de un entorno con obstáculos. Existen diversos esquemas, pero todos ellos poseen en común el afán por llevar el vehículo a su destino de forma segura. [24]

Todos los robots móviles se han caracterizado por tener algún sistema para evitar obstáculos (generador local de trayectorias), desde los primitivos algoritmos que detectaban un obstáculo y detenían al robot a corta distancia de éste a fin de evitar una colisión, hasta llegar a los algoritmos más sofisticados que le permiten al robot rodear al obstáculo para poder llegar al punto destino. [16]

Hay muchos tipos de navegación del robot de donde se pueden distinguir:

Manual a distancia o tele-operado

Un robot manualmente tele-operado está totalmente bajo el control de un conductor con una palanca de mando o de otro dispositivo de control. El dispositivo puede ser conectado directamente en el robot, puede ser un joystick inalámbrico, o puede ser un accesorio para un equipo inalámbrico u otro controlador. Un robot de tele-operado se suele utilizar para mantener al operador fuera de peligro.

Robot de manera autónoma al azar

Los robots autónomos con el movimiento al azar, básicamente, rebotan en las paredes, si esas paredes se detectan.

Robot autónomo guiado

Un robot autónomo guiado conoce al menos alguna información acerca de dónde está y cómo llegar a diferentes metas y puntos de interés o en el camino. “Localización” o el conocimiento de su ubicación actual, se calcula por uno o más medios, el uso de sensores codificadores de estos motores, la visión, visión estereoscópica, láser y sistemas de posicionamiento global. Los sistemas de posicionamiento a menudo utilizan triangulación, la posición relativa y / o localización Monte-Carlo/Markov para determinar la localización y orientación de la plataforma, de la que puede planificar una ruta a su siguiente punto de referencia u objetivo. [25]

2.3.4 Sistema de control y procesamiento

Es el sistema principal, si se considera que en él se procesa toda la información del robot. Actualmente realiza las siguientes funciones: control de motores, manejo de información del sistema sensorial, algoritmos de navegación y algoritmos de procesamiento de información. [17]

Existen diferentes sistemas embebidos que pueden realizar estas tareas (Véase Figura 2-7), tales como:

- ♦ FPGA's.
- ♦ Microcontroladores.
- ♦ DSP's.
- ♦ Smartphone.
- ♦ SBC (Single Board Computer).

Cada uno de estos dispositivos tiene ventajas, desventajas y depende en gran medida de las características del proyecto para seleccionar el más adecuado. Comúnmente se suele utilizar una combinación de algunos de estos dispositivos para así obtener las ventajas de cada uno.

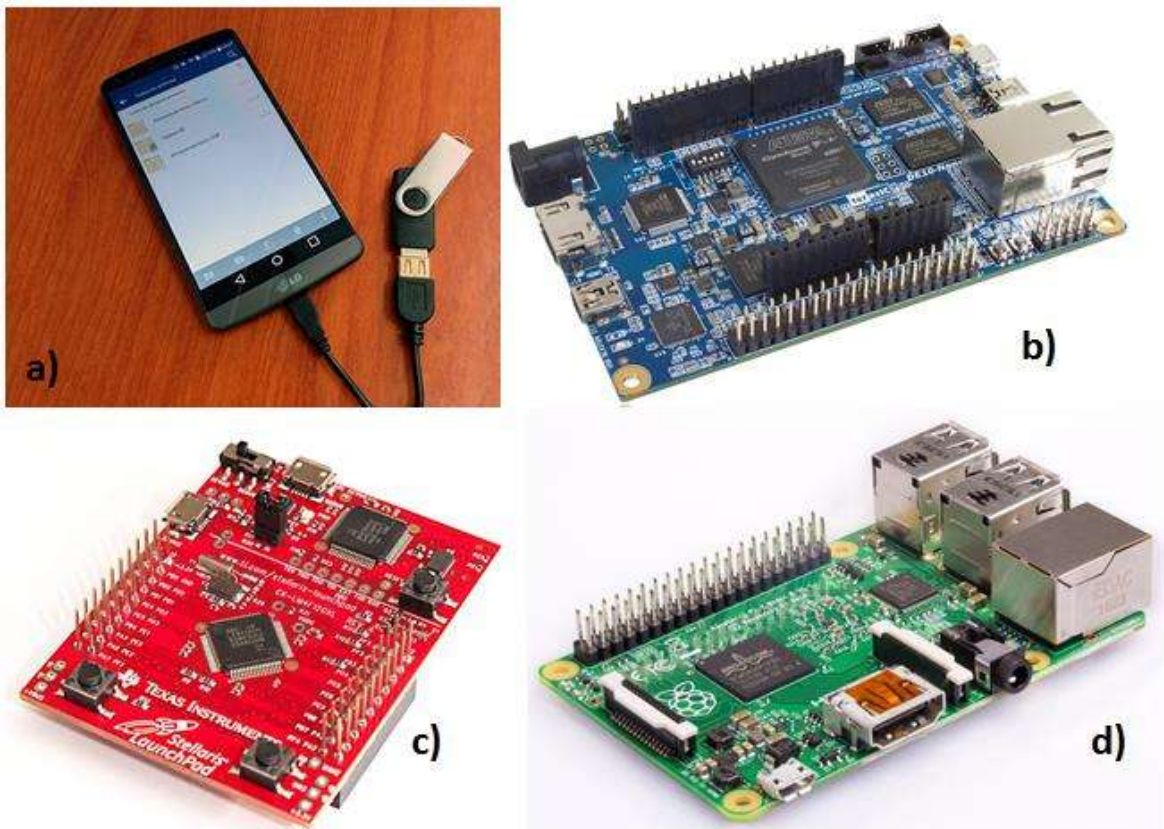


Figura 2-7 Sistemas embebidos: a) Smartphone [26], b) FPGA [27], c) Microcontrolador [28] y d) SBC [29]

Para llevar a cabo cada una de estas funciones se ha requerido desarrollar niveles cognitivos que incorporen un procesamiento de información altamente relacionado y transportable a dispositivos electrónicos. Los principales conceptos utilizados en la robótica son: lógica difusa, redes neuronales artificiales, inteligencia artificial, etc. [17]

2.3.4.1 Lógica de control

La implementación de un sistema robótico está caracterizada, además de su estructura física, por la lógica de control y la lógica de aplicación que desarrolla. La lógica de control es la responsable de determinar el flujo de ejecución, decidir qué tareas han de desarrollarse en cada instante y

según la reacción del sistema y del entorno que está siendo intervenido, cuáles deberán realizarse en un futuro, de mantener el sistema bajo unos parámetros de seguridad razonables y, en general, controlar la actividad del robot para alcanzar la misión global del sistema. La lógica de aplicación implementa las habilidades del sistema, la funcionalidad, la forma en la que el robot va a desplazarse, a realizar un movimiento, a extraer una conclusión sobre lo que sus sensores captan del entorno o comúnmente conocido como implementar la inteligencia del robot. Debido a la complejidad de un sistema robótico, la separación entre ambas lógicas muchas veces no está clara y por ello a lo largo del tiempo han ido surgiendo una serie de enfoques y corrientes que permiten caracterizar el sistema de control de un robot desde diversos puntos de vista, tales como el reactivo, el deliberativo o el basado en comportamientos. [30] Véase Figura 2-8

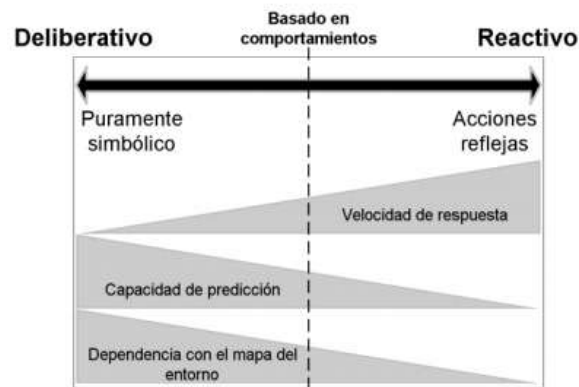


Figura 2-8 Estrategias de control para robots móviles. [16]

Las estrategias de control deliberativo se basan en una estrategia puramente simbólica, esto significa que la semejanza entre el entorno y su modelo en el robot debe ser precisa para que el comportamiento del robot sea el deseado. Por ejemplo, un robot móvil preparado para navegar en un entorno estático probablemente no sabrá qué hacer cuando se encuentre en un pasillo por el que circulan personas; esta dependencia con el entorno limita su rango de aplicación. Sin embargo, las estrategias deliberativas incluyen un análisis de estabilidad que permite garantizar, a priori, bajo qué condiciones del entorno el robot móvil cumplirá con sus objetivos. Esto conduce a sistemas de control y de procesamiento de la información complejos que consumen un importante costo computacional que, además, restringe su velocidad de respuesta. [16]

Por el contrario, las estrategias de control reactivo se basan en un esquema de acciones reflejas, esto significa que el entorno se percibe como un estímulo (distancia a los objetos, nivel de luz, temperatura, etc.) que genera una acción de control en función de la intensidad del mismo. Esta independencia con el entorno, junto con funciones de control simples del tipo proporcional, es lo que potencia a este tipo de estrategias por su elevada velocidad de respuesta y su bajo costo computacional. Sin embargo, la superposición de acciones reflejas termina por generar en muchos casos un comportamiento emergente no deseado, por ejemplo, un robot móvil diseñado para buscar fuentes de luz, pero que en presencia de penumbra gira 180° y escapa, podrá tener un comportamiento no deseado si entre la fuente de luz y el robot existe un obstáculo que proyecta su sombra sobre la trayectoria del robot. En la mayoría de los casos los comportamientos

emergentes no son tan obvios y solo se perciben cuando el sistema de control es implantado en el robot. [16]

En el medio quedan las estrategias de control basado en comportamientos que tratan de rescatar las ventajas del control deliberativo y del reactivo incorporando estrategias de control híbrido para garantizar la estabilidad de múltiples controladores simples operando en paralelo junto con técnicas de aprendizaje para mejorar el desempeño del robot para lograr independizarlo del modelo del entorno. [16]

2.4 Comunicaciones inalámbricas

La comunicación inalámbrica o comunicación a distancia, es aquella capaz de enviar una cantidad de datos de un punto a otro (emisor y receptor) sin la necesidad de un agente o un hardware que conecte ambos puntos físicamente. Principalmente el componente que logra el funcionamiento de dicha comunicación es la programación (software). El hardware que se utiliza para el correcto funcionamiento son antenas, alambres conductores de energía a distancia, etc. [31]

2.4.1 Radiofrecuencia

La radiocomunicación es una forma de telecomunicación que se realiza a través de ondas de radio u ondas hertzianas, la que a su vez está caracterizada por el movimiento de los campos eléctricos y campos magnéticos. La comunicación vía radio se realiza a través del espectro radioeléctrico cuyas propiedades son diversas dependiendo de sus bandas de frecuencia. Así tenemos bandas conocidas como baja frecuencia, media frecuencia, alta frecuencia, muy alta frecuencia, ultra alta frecuencia, etc. En cada una de ellas, el comportamiento de las ondas es diferente.

Aunque se emplea la palabra radio, las transmisiones de televisión, radio, radar y telefonía móvil están incluidos en esta clase de emisiones de radiofrecuencia. [32]

La radiofrecuencia se puede dividir en bandas del espectro electromagnético según su frecuencia. Véase Tabla 2-2.

Tabla 2-2 Radiofrecuencia [33]

Nombre de frecuencia	Frecuencia en inglés	Banda UIT	Frecuencias	Longitud de onda
Frecuencia extremadamente baja	<i>Extremely Low Frequency (ELF)</i>	1	3-30 Hz	100.000 – 10.000 km
Súper baja frecuencia	<i>Super Low Frequency (SLF)</i>	2	30-300 Hz	10.000 – 1.000 km
Ultra baja frecuencia	<i>Ultra Low Frequency (ULF)</i>	3	300 – 3.000 Hz	1.000 – 100 km

Muy baja frecuencia	<i>Very Low Frequency (VLF)</i>	4	3 – 30 kHz	100 – 10 km
Baja frecuencia	<i>Low Frequency (LF)</i>	5	30 – 300 kHz	10 – 1 km
Media frecuencia	<i>Medium Frequency (MF)</i>	6	300 – 3.000 kHz	1 km – 100 m
Alta Frecuencia	<i>High Frequency (HF)</i>	7	3 – 30 MHz	100 m – 10 m
Muy alta frecuencia	<i>Very High Frequency (VHF)</i>	8	30-300 MHz	10 – 1 m
Ultra alta frecuencia	<i>Ultra High Frequency (UHF)</i>	9	300-3.000 MHz	1 m – 100 mm
Súper alta frecuencia	<i>Super High Frequency (SHF)</i>	10	3-30 GHz	100 – 10 mm
Frecuencia extremadamente alta	<i>Extremely High Frequency (EHF)</i>	11	30-300 GHz	10 – 1 mm

2.4.1.1 Transmisión y recepción

Una onda de radio se origina cuando una partícula cargada (por ejemplo, un electrón) se excita a una frecuencia situada en la zona de radiofrecuencia (RF) del espectro electromagnético. Cuando la onda de radio actúa sobre un conductor eléctrico (la antena), induce en él un movimiento de carga eléctrica (corriente eléctrica) que puede ser transformado en señales de audio u otro tipo de señales portadoras de información.

El emisor tiene como función producir una onda portadora, cuyas características son modificadas en función de las señales (audio o vídeo) a transmitir. Propaga la onda portadora así modulada. El receptor capta la onda y la «demodula» para hacer llegar al espectador auditor tan solo la señal transmitida. [32]

2.4.1.2 Modulación Digital

Las formas básicas de modulación digital son ASK, FSK, PSK.

ASK (Amplitude Shift Keying)

Las ventajas de este tipo de modulación son el sencillo diseño (menor coste) y el bajo consumo, la desventaja es la fragilidad en presencia de interferencias por ruido eléctrico, que pueden provocar errores en los datos recibidos. [34] Véase Figura 2-9

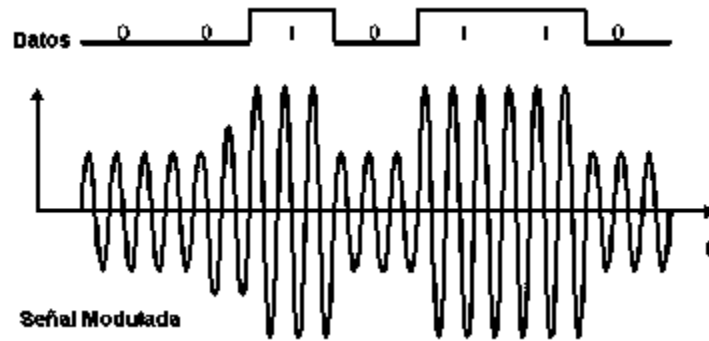


Figura 2-9 Modulación ASK [34]

FSK (Frequency Shift Keying)

La modulación por desplazamiento de frecuencia (FSK), véase Figura 2-10, con un 0 digital se transmite una portadora a una frecuencia y con un 1 digital se transmite la portadora a otra frecuencia distinta, con la misma amplitud. La ventaja de este tipo de modulación es la mayor robustez ante la presencia de interferencias. La desventaja es la complejidad del sistema (mayor coste) y el consumo que permanece siempre presente durante la transmisión. [34]

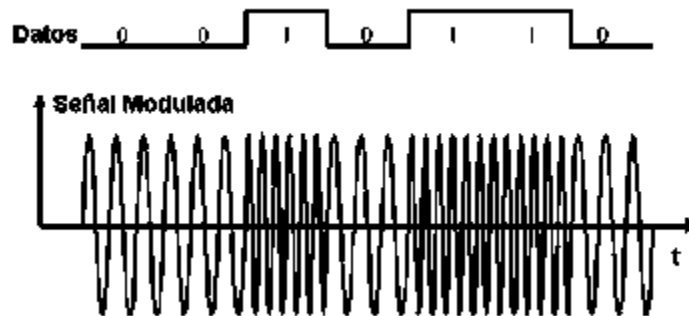


Figura 2-10 Modulación FSK [34]

PSK (Phase Shift Keying)

Se codifican los valores binarios como cambios de fase de la señal portadora. La modulación PSK (véase Figura 2-11) consiste en cada estado de modulación está dado por la fase que lleva la señal respecto de la original. Técnicamente utilizando el concepto de modulación PSK, es posible aumentar la velocidad de transmisión a pesar de los límites impuestos por el canal de comunicación. [34]

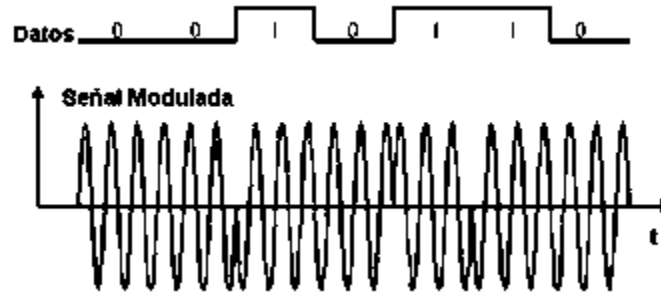


Figura 2-11 Modulación PSK [34]

2.4.2 Comunicaciones con Protocolo estándar

2.4.2.1 Bluetooth (802.15)

Bluetooth es un sistema de comunicación inalámbrico que aparece asociado a las Redes de Área Personal Inalámbricas en inglés WPAN (Wireless Personal Area Network). Esta nueva especificación está establecida para el enlace entre dispositivos de voz y datos a corto alcance, de forma fácil y simple. Bluetooth opera en una banda no licenciada ISM (Industrial Scientific Medical) de 2.4- 2.5GHz permitiendo la transmisión de voz y datos, de forma rápida y segura con un rango de hasta 10 metros con 1 miliwatt o 100 metros si se usa un amplificador con 100 miliwatts. [34]

Dentro de una aplicación típica de Bluetooth se puede encontrar los siguientes elementos:

- ♦ Master: es el dispositivo Bluetooth que establece e inicializa la conexión, la secuencia de control “hopping” y la temporización de los demás dispositivos colocados en lo que se llama una red “Piconet”.
- ♦ Slave: es el dispositivo habilitado en una Piconet. Una red Piconet tiene un máximo de 7 esclavos.
- ♦ Piconet: una red de hasta 8 dispositivos conectados (1 maestro+ 7 esclavos). Scatternet: red formada por diferentes redes Piconet.

La arquitectura bluetooth se organiza en "piconets", formadas por dos o más dispositivos compartiendo un canal; uno de los terminales actúa como maestro de la “piconet”, mientras que el resto actúan como esclavos. [34]

2.4.2.2 Wifi (IEEE 802.11)

Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con WIFI (tales como computadoras personales, teléfonos, televisores, videoconsolas, reproductores de música...) pueden conectarse entre sí o a internet a través de un punto de acceso de red inalámbrica. Wi-Fi es una marca de la Alianza Wi-Fi, la organización comercial que adopta, prueba y certifica que los equipos cumplen con los estándares 802.11 relacionados con redes inalámbricas de área local. [35] Tabla 2-3

Tabla 2-3 Estándares 802.11 [36]

Estándares	Velocidad (Teórica)	Velocidad (Práctica)	Frecuencia	Año de implementación
802.11	2Mb/s	1Mb/s	2.4GHz	1997
802.11 b	11Mb/s	6Mb/s	2.4GHz	1999
802.11 g	54Mb/s	22Mb/s	2.4GHz	2003
802.11 n	600Mb/s	100Mb/s	2.4 y 5 GHz	2009
802.11 ac	6.93Gb/s	100Mb/s	5 GHz	2013

La norma IEEE 802.11 fue diseñada para sustituir el equivalente a las capas físicas y MAC de la norma 802.3 (Ethernet). Esto quiere decir que en lo único que se diferencia una red WIFI de una red Ethernet es en cómo se transmiten las tramas o paquetes de datos; el resto es idéntico. Por tanto, una red local inalámbrica 802.11 es completamente compatible con todos los servicios de las redes locales (LAN) de cable 802.3 (Ethernet). [35]

2.4.2.2.1 Dispositivos

Existen varios dispositivos WIFI, los cuales se pueden dividir en dos grupos: dispositivos de distribución o de red, entre los que destacan los enrutadores, puntos de acceso, repetidores; y dispositivos terminales que en general son las tarjetas receptoras para conectar a la computadora personal, ya sean internas (tarjetas PCI) o bien USB. [35]

Dispositivos de distribución o de red:

- ♦ Los puntos de acceso son dispositivos que generan un conjunto de servicios que se pueden definir como una red WIFI a la que se pueden conectar otros dispositivos. Los puntos de acceso permiten conectar dispositivos de forma inalámbrica a una red existente. Pueden agregarse más puntos de acceso a una red para generar redes de cobertura más amplia, o conectar antenas más grandes que amplifiquen la señal.
- ♦ Los repetidores inalámbricos son equipos que se utilizan para extender la cobertura de una red inalámbrica. Se conectan a una red existente que tiene señal más débil y crean una señal más fuerte a la que se pueden conectar los equipos dentro de su alcance.
- ♦ Los enrutadores inalámbricos son dispositivos especialmente diseñados para redes pequeñas (hogar o pequeña oficina). Estos dispositivos incluyen un enrutador (encargado de interconectar redes; por ejemplo, nuestra red del hogar con Internet), un punto de acceso y generalmente un conmutador que permite conectar algunos equipos vía cable (Ethernet y USB). Su tarea es tomar la conexión a Internet y brindar a través de ella acceso a todos los equipos que conectemos, sea por cable o en forma inalámbrica. [35]

Los dispositivos terminales abarcan dos tipos mayoritarios: tarjetas PCI y tarjetas USB:

- ♦ Las tarjetas PCI para WIFI se agregan (o vienen de fábrica) en las computadoras de escritorio. Hoy en día están perdiendo terreno debido a las tarjetas USB. Dentro de este grupo también pueden agregarse las tarjetas MiniPCI que vienen integradas en casi cualquier computador portátil disponible hoy en el mercado.

- ♦ Las tarjetas USB para WIFI son el tipo de tarjeta más común que existe en las tiendas y más sencillo de conectar a un ordenador personal, ya sea de sobremesa o portátil, disponiendo de todas las ventajas de la tecnología USB. Hoy en día puede encontrarse incluso tarjetas USB con el estándar 802.11n (Wireless-N) que es el último estándar liberado para redes inalámbricas.
- ♦ También existen impresoras, cámaras Web y otros periféricos que funcionan con la tecnología WIFI, permitiendo un ahorro de cableado en las instalaciones de redes y especialmente gran movilidad de equipo. [35]

2.4.2.2.2 Modos de conexiones

Existen dos tipos de conexiones Wi-Fi: el modo "infraestructura" y el modo "ad hoc".

- ♦ El modo infraestructura es la conexión que se efectúa entre un equipo o dispositivo y un punto de acceso inalámbrico (AP) ya sea un router o un punto público. [36] Figura 2-12 Modo infraestructura Figura 2-12

La configuración formada por el punto de acceso y las estaciones ubicadas dentro del área de cobertura se llama conjunto de servicio básico o BSS. Estos forman una célula. Cada BSS se identifica a través de un BSSID (identificador de BSS) que es un identificador de 6 bytes (48 bits). En el modo infraestructura el BSSID corresponde al punto de acceso de la dirección MAC. [37]

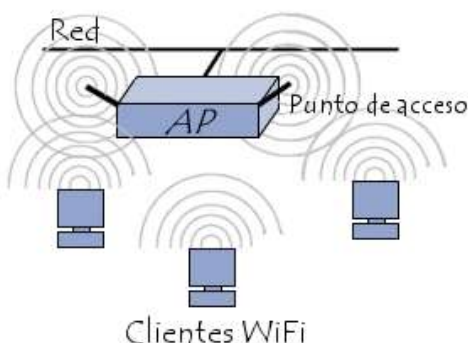


Figura 2-12 Modo infraestructura [37]

- ♦ El segundo, el modo ad-hoc es la conexión que se establece entre dos equipos o dispositivos de forma independiente. Figura 2-13. Esta conexión solo permite algunos metros de alcance. [36] La configuración que forman las estaciones se llama conjunto de servicio básico independiente o IBSS. A diferencia del modo infraestructura, el modo ad hoc no tiene un sistema de distribución que pueda enviar tramas de datos desde una estación a la otra. Entonces, por definición, un IBSS es una red inalámbrica restringida. [37]

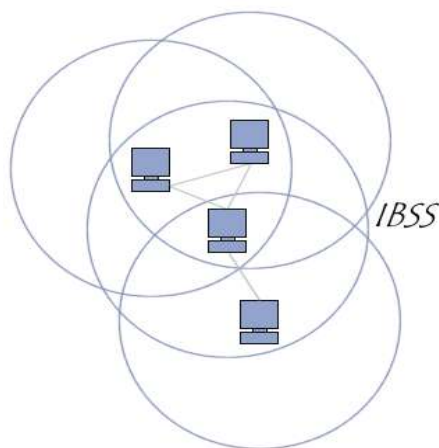


Figura 2-13 Modo 'ad hoc' [37]

2.5 Red de computadoras

Una red de computadoras, también llamada red de ordenadores o red informática, es un conjunto de equipos (computadoras y/o dispositivos) conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten Información (archivos), recursos (CD-ROM, impresoras), servicios (acceso a internet, E-mail, chat, juegos). Una red de comunicaciones es un conjunto de medios técnicos que permiten la comunicación a distancia entre equipos autónomos (no jerárquica -master/slave-). Normalmente se trata de transmitir datos, audio y vídeo por ondas electromagnéticas a través de diversos medios de transmisión (aire, vacío, cable de cobre, Cable de fibra óptica). [38]

Para simplificar la comunicación entre programas (aplicaciones) de distintos equipos, se definió el modelo OSI por la ISO, el cual especifica 7 distintas capas de abstracción. Con ello, cada capa desarrolla una función específica con un alcance definido. Figura 2-14 [38]

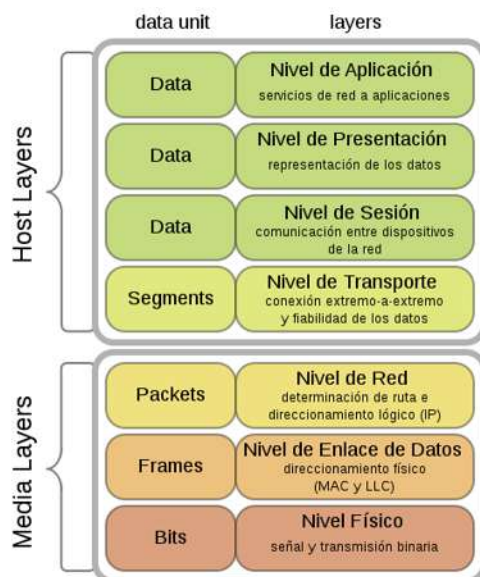


Figura 2-14 Modelo OSI [39]

Clasificación:

Las redes pueden clasificarse mediante distintos criterios [38]:

- ♦ Por alcance:
 - Red de área personal (PAN).
 - Red de área local (LAN).
 - Red de área de campus (CAN).
 - Red de área metropolitana (MAN).
 - Red de área amplia (WAN).
 - Red de área simple (SPL).
 - Red de área de almacenamiento (SAN).
- ♦ Por Método de la Conexión:
 - Medios guiados: Cable coaxial, cable de par trenzado, Fibra óptica y otros tipos de cables.
 - Medios no guiados: radio, infrarrojos, microondas, láser y otras redes inalámbricas.
- ♦ Por relación funcional:
 - Cliente-servidor.
 - Igual-a-igual (P2P).
- ♦ Por topología de red:
 - Red en bus.
 - Red en estrella.
 - Red en anillo (o doble anillo).
 - Red en malla (o totalmente conexa).
 - Red en árbol.
 - Red mixta (cualquier combinación de las anteriores).

- ♦ Por la Direccionalidad de los datos (tipos de transmisión):
 - Simplex (unidireccionales): un Equipo Terminal de Datos transmite y otro recibe. (p. ej. Streaming).
 - Half-Duplex (bidireccionales): sólo un equipo transmite a la vez. También se llama Semi-Duplex (p. ej. una comunicación por equipos de radio, si los equipos no son full dúplex, uno no podría transmitir (hablar) si la otra persona está también transmitiendo (hablando) porque su equipo estaría recibiendo (escuchando) en ese momento).
 - Full-Duplex (bidireccionales): ambos pueden transmitir y recibir a la vez una misma información. (p. ej. videoconferencia). [38]

2.5.1 Protocolos de redes

El Protocolo de red o también protocolo de Comunicación es el conjunto de reglas que especifican el intercambio de Datos u órdenes durante la Comunicación entre las entidades que forman parte de una red. [38]

Estándares de redes:

- ♦ IEEE 802.3, estándar para Ethernet.
- ♦ IEEE 802.11, estándar para Wi-Fi.
- ♦ IEEE 802.15, estándar para Bluetooth.

Para la disciplina científica y la ingeniería que estudia las redes de ordenadores, una red de ordenadores es el conjunto de ordenadores conectados junto con un sistema de telecomunicaciones con el fin de comunicarse y compartir recursos e información. Expertos en la materia de discusión del establecimiento de una red dicen que si dos ordenadores están conectados entre sí en forma de medio de comunicaciones constituyen una red. Sin embargo, unos afirman que una red se constituye de tres ordenadores conectados o más. [38]

2.5.2 Arquitectura “cliente-servidor”

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. [40]

2.5.2.1 HTTP

El Protocolo de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol o HTTP) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web, definiendo la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. [41]

HTTP define una serie predefinida de métodos de petición (algunas veces referido como "verbos") que pueden utilizarse. Cada método indica la acción que desea que se efectúe sobre el recurso identificado. Lo que este recurso representa depende de la aplicación del servidor. Por ejemplo, el recurso puede corresponderse con un archivo que reside en el servidor. [41]

Los métodos más usados son “GET” y “POST”, el primero pide una representación del recurso especificado en el URL y el segundo envía los datos para que sean procesados por el recurso identificado. Por seguridad el método “GET” no debería ser usado para aplicaciones que causen efectos ya que transmite información agregando parámetros a la URL. [41]

2.5.2.2 Servidores

Un servidor es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, incluso en computadoras dedicadas a las cuales se les conoce individualmente como «el servidor». En la mayoría de los casos una misma computadora puede proveer múltiples servicios y tener varios servidores en funcionamiento. La ventaja de montar un servidor en computadoras dedicadas es la seguridad. Por esta razón la mayoría de los servidores son procesos diseñados de forma que puedan funcionar en computadoras de propósito específico. [42]

Los servidores operan a través de una arquitectura cliente-servidor que se puede apreciar en la Figura 2-15. Los servidores son programas de computadora en ejecución que atienden las peticiones de otros programas, los clientes. Por tanto, el servidor realiza otras tareas para beneficio de los clientes. Ofrece a los clientes la posibilidad de compartir datos, información y recursos de hardware y software. Los clientes usualmente se conectan al servidor a través de la red, pero también pueden acceder a él a través de la computadora donde está funcionando. En el contexto de redes Internet Protocol (IP), un servidor es un programa que opera como oyente de un socket. [42]

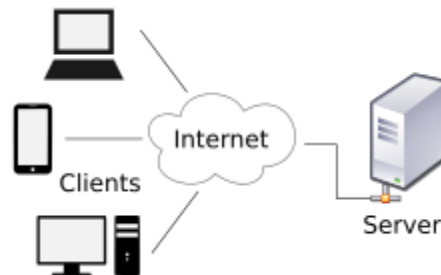


Figura 2-15 Arquitectura cliente-servidor [43]

Algunos tipos de servidores [38]:

- ◆ Servidor de archivos: almacena varios tipos de archivo y los distribuye a otros clientes en la red.
- ◆ Servidor de correo: almacena, envía, recibe, enruta y realiza otras operaciones relacionadas con e-mail para los clientes de la red.
- ◆ Servidor de fax: almacena, envía, recibe, enruta y realiza otras funciones necesarias para la transmisión, la recepción y la distribución apropiadas de los Fax.
- ◆ Servidor proxy: realiza un cierto tipo de funciones a nombre de otros clientes en la red para aumentar el funcionamiento de ciertas operaciones (p. ej., prefetching y depositar documentos u otros datos que se soliciten muy frecuentemente). También sirve como sistema de seguridad; esto es, tiene un Firewall (cortafuegos). Permite administrar el

acceso a internet en una red de computadoras permitiendo o negando el acceso a diferentes sitios web.

- ◆ Servidor web: almacena Documentos HTML, imágenes, Archivos de texto, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.
- ◆ Servidor de reserva: tiene el Software de reserva de la red instalado y tiene cantidades grandes de almacenamiento de la red en discos duros u otras formas del almacenamiento (cinta) disponibles para que se utilice con el fin de asegurarse de que la pérdida de un servidor principal no afecte a la red. Esta técnica también es denominada clustering.
- ◆ Servidor DNS: Este tipo de servidores resuelven nombres de dominio sin necesidad de conocer su dirección IP. [38]

2.5.2.3 Navegador web (cliente)

Navegador o explorador web. Es un programa o Software, que permite visualizar páginas web a través de Internet o en el propio ordenador, además posibilita acceder a otros recursos de información alojados también en Servidores Web, como pueden ser videos, imágenes, audio y archivos XML.

El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos. Véase Figura 2-16 [44]

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web). Los documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen. El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a la Internet, se llama navegación, de donde se origina el nombre navegador. [44]

¿Cómo funcionan los navegadores?

Los navegadores se comunican con los servidores web por medio del protocolo de transferencia de hipertexto (HTTP) para acceder a las direcciones de Internet (URLs) a través de los motores de búsqueda. La mayoría de los exploradores web admiten otros protocolos de red como HTTPS (la versión segura de HTTP), Gopher, y FTP, así como los lenguajes de marcado o estándares HTML y XHTML de los documentos web. Los navegadores además interactúan con complementos o aplicaciones (Plug-ins) para admitir archivos Flash y programas en Java (Java applets). [44]



Figura 2-16 Navegadores web [44]

2.5.3 Páginas Web

Una página web, o página electrónica, página digital, o ciber página es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes y muchas otras cosas, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar acceso a otras páginas web mediante enlaces de hipertexto. Frecuentemente también incluyen otros recursos como pueden ser hojas de estilo en cascada, guiones (scripts), imágenes digitales, entre otros. [45]

Las páginas web pueden estar almacenadas en un equipo local o en un servidor web remoto. El servidor web puede restringir el acceso únicamente a redes privadas, por ejemplo, en una intranet corporativa, o puede publicar las páginas en la World Wide Web. El acceso a las páginas web es realizado mediante una transferencia desde servidores, utilizando el protocolo de transferencia de hipertexto (HTTP). [45]

2.5.3.1 HTML

El HTML (Hyper Text Markup Language) es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, vídeo, etc., el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores (programas que permiten visualizar las páginas web). [46]

Utiliza marcas para describir la forma en la que deberían aparecer el texto y los gráficos en un Navegador web que, a su vez, están preparados para leer esas marcas y mostrar la información en un formato estándar. Una marca es el elemento básico del código que asigna el formato a la página e indica al navegador cómo mostrar determinados elementos. Las marcas no aparecen cuando se muestran las Páginas Web, pero son una parte fundamental de la creación HTML. Estos símbolos son esenciales, pues le indicarán al navegador que se trata de una instrucción, no de texto que debe aparecer en la pantalla. [46]

2.5.3.2 CSS

Hojas de estilo en cascada (en inglés Cascading Style Sheets), CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. [47]

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- ◆ Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ◆ Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- ◆ Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- ◆ El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

2.5.3.3 JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. [48]

JavaScript surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un "lenguaje de scripting" del lado del cliente, sin embargo, hoy JavaScript es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 han provocado que el uso de JavaScript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel. Pero en los últimos años JavaScript se está convirtiendo también en el lenguaje "integrador". Lo encontramos en muchos ámbitos, ya no solo en Internet y la Web, también es nativo en sistemas operativos para ordenadores y dispositivos, del lado del servidor y del cliente. [49]

A JavaScript se le denomina "del lado del cliente" porque donde se ejecuta es en el navegador (cliente web), en contraposición a lenguajes como PHP que se ejecutan del "lado del servidor". En

el lado que nos ocupa con JavaScript, el cliente, es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con todos los navegadores modernos se ha convertido en un estándar como lenguaje de programación del lado del cliente. [49]

El uso más común de JavaScript es escribir funciones embebidas o incluidas en páginas HTML y que interactúan con el Document Object Model (DOM o Modelo de Objetos del Documento) de la página. Algunos ejemplos sencillos de este uso son: [48]

- ◆ Cargar nuevo contenido para la página o enviar datos al servidor a través de AJAX sin necesidad de recargar la página (por ejemplo, una red social puede permitir al usuario enviar actualizaciones de estado sin salir de la página).
- ◆ Animación de los elementos de página, hacerlos desaparecer, cambiar su tamaño, moverlos, etc.
- ◆ Contenido interactivo, por ejemplo, juegos y reproducción de audio y vídeo.
- ◆ Validación de los valores de entrada de un formulario web para asegurarse de que son aceptables antes de ser enviado al servidor.
- ◆ Transmisión de información sobre los hábitos de lectura de los usuarios y las actividades de navegación a varios sitios web. Las páginas Web con frecuencia lo hacen para hacer análisis web, seguimiento de anuncios, la personalización o para otros fines. [48]

JavaScript en el lado servidor

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. [50] Es capaz de servir para el desarrollo web, igual que es también capaz de servir para muchas otras cosas que no tienen nada que ver. Por ese motivo, aunque a veces se denomina a Node.js como el JavaScript del lado del servidor. [49]

Node.js es un lenguaje que tiene unas características diferentes a otros lenguajes, como su característica más peculiar encontramos lo que se llama programación "asíncrona" que no es más que la capacidad de realizar acciones que lleven un tiempo para la ejecución sin necesidad de mantener procesos en estado de espera. Esto provoca que su ejecución sea bastante ligera y eficiente. A pesar que sirve para muchas cosas, Node.js se ha convertido en un lenguaje muy usado por los desarrolladores web. Muchas herramientas del día a día de los desarrolladores están programadas con Node.js como los gestores de paquetes "npm" o "bower". [49]

3

Capítulo 3. Diseño y descripción de Hardware

En este capítulo se describen las características del hardware empleado, tales como los sistemas electrónicos y mecánicos.

Como se mencionó en la sección 2.3.4, se necesita un sistema de control y procesamiento que se encargue de los motores, realizar la lectura de sensores, algoritmos de procesamiento de información, transmisión de datos, etc. De acuerdo a la Figura 3-1. De entre los posibles dispositivos capaces de realizar esta labor, se seleccionó el computador de placa única (SBC) debido a sus especificaciones, este dispositivo es el que mejor se adapta al proyecto descrito. Y específicamente se seleccionó la Raspberry Pi 3 Modelo B, ya que esta placa está diseñada para desarrolladores y cuenta con un software flexible y “open source”.

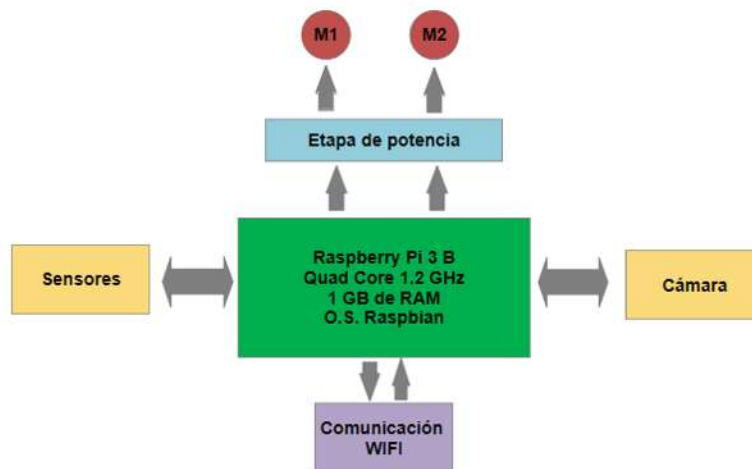


Figura 3-1 Diagrama de bloques de Hardware

3.1 Raspberry Pi

Raspberry Pi es un computador de placa única reducida o de placa simple (SBC) de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi. El software es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite usar otros sistemas operativos, incluido una versión de Windows 10. En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet, 40

pinos GPIO y un conector para cámara. Ninguna de sus ediciones incluye memoria para el sistema operativo, siendo esta en su primera versión una tarjeta SD y en ediciones posteriores una tarjeta MicroSD. [51] En la Figura 3-2 se muestra una versión de la Raspberry Pi.



Figura 3-2 Raspberry Pi 2 model B (2015) [51]

3.1.1 Modelos y especificaciones

Dependiendo de la versión del computador se tienen diferentes especificaciones o características, las cuales se listan en la Tabla 3-1. Véase Figura 3-3.

Tabla 3-1 Especificaciones Raspberry Pi [51]

	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				RISC de 64 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 de 2012		1 GB (compartidos con la GPU)	
Puertos USB 2.0:	1	2 (vía hub USB integrado)	4		
Entradas de vídeo:	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RPF				
Salidas de vídeo:	Conector RCA (PAL y NTSC)	HDMI (rev1.3 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 (RJ-45) via hub USB		10/100 Ethernet (RJ-45) vía hub USB, Wifi 802.11n, Bluetooth 4.1
Periféricos de bajo nivel:	8 x GPIO, SPI, I ² C, UART			17 x GPIO y un bus HAT ID
Consumo energético:	500 mA, (2.5 W)	700 mA, (3.5 W)	600 mA, (3.0 W)	800 mA, (4.0 W)
Fuente de alimentación:	5 V vía Micro USB o GPIO header			
Dimensiones:	85.60mm × 53.98mm (3.370 × 2.125 inch)			
Sistemas operativos soportados	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM. RISC OS			



Figura 3-3 RPI 1 Modelo A+ (izquierda), RPI 2 Modelo B (centro) y RPI 3 Modelo B (derecha) [52]

3.1.2 RPI modelo 3 B

Sacada a la luz en el año 2016, renueva procesador, una vez más de la compañía Broadcom, una vez más un Quad-Core, pero pasa de 900MHz a 1.20GHz. Mantiene la RAM en 1GB. Su mayor novedad fue la inclusión de Wi-Fi (802.11n Wireless lan) y Bluetooth (4.1 Low Energy) sin necesidad de adaptadores. [51] Véase Figura 3-4

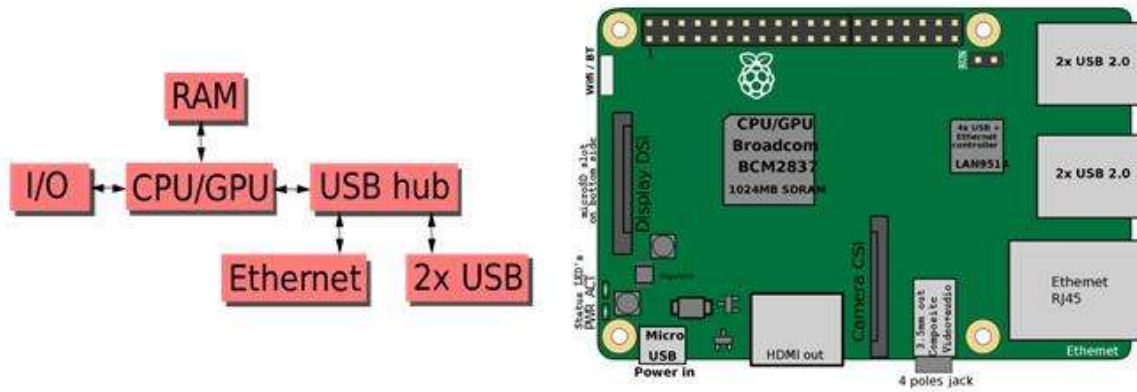


Figura 3-4 Diagrama de bloques (izquierda) [51] y localización de componentes en la placa (derecha) [53]

3.1.2.1 GPIO

Una característica poderosa de la Raspberry Pi es la fila de pines GPIO (entrada / salida de propósito general) a lo largo del borde superior de la placa. Un conector GPIO de 40 pines se encuentra en todas las placas Raspberry Pi actuales (no en Pi Zero y Pi Zero W) como se muestra en la Figura 3-5. Antes del Pi 1 Modelo B + (2014), los tableros comprendían un conector más corto de 26 pines. [54]

Los pines de propósito general GPIO, por sus siglas en Inglés (General Purpose I/O), permiten controlar e interactuar con el mundo real. Cualquiera de los pines GPIO puede ser designado (mediante software) como entrada o salida y usado para un amplio rango de propósitos.



Figura 3-5 RPI GPIO [54]

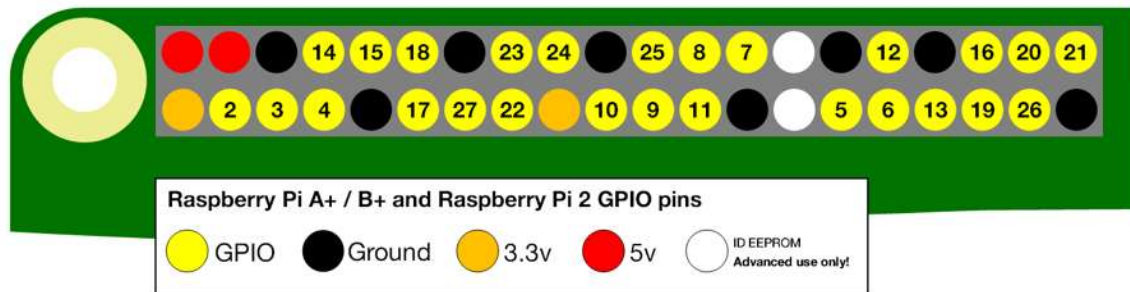


Figura 3-6 RPI GPIO Software [54]

Dos pines de 5V y dos pines de 3.3V están disponibles en la placa, como un número de pines de tierra (0V), los cuales no son configurables. Los pines restantes son todos de 3.3V de uso general, es decir, las salidas son puestas en 3.3V y las entradas son tolerantes a 3.3V. Véase Figura 3-6

Además de los dispositivos simples de entrada y salida, los pines GPIO se pueden utilizar con una variedad de funciones alternativas, algunos están disponibles en todos los pines, otros en pines específicos. [54]

GPIO Periféricos especiales:

- ◆ PWM (Pulse Width Modulation).
- ◆ SPI.
- ◆ I2C.
- ◆ Serial.

3.2 Sensores

El sistema de percepción del robot móvil, es decir, los sensores, es un componente indispensable para el funcionamiento, sin ellos el robot no podría conocer su ubicación ni los obstáculos cercanos a él. En esta sección se describen cada uno de los sensores que se utilizan en el proyecto.

3.2.1 Sensor de Temperatura y humedad: DHT11

Existe una amplia gama de sensores en la actualidad que pueden hacer mediciones de temperatura y humedad, sin embargo, cabe destacar que el usado en este proyecto fue el Sensor DHT11 debido a que es digital nos presenta una gran ventaja frente a los sensores analógicos como en LM35 que son propensos al ruido por esa misma característica.

El DHT11 presume de ser un sensor con alta fiabilidad y estabilidad debido a su señal digital calibrada, aunque es básico y un tanto lento respecto a la frecuencia de muestreo. El sensor DHT11 que se muestra en la Figura 3-7 está compuesto de dos partes, un sensor capacitivo de humedad y un termistor. También hay dentro un chip básico que hace la conversión de las señales de analógico a digital y responde con una señal digital de las dos mediciones. [55]

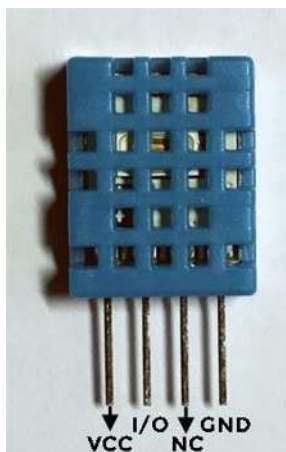


Figura 3-7 DHT11 [56]

Los pines de la versión sin PCB del DHT11 son (vista frontal de izquierda a derecha):

- VCC: alimentación.

- I/O: transmisión de datos.
- NC: no conecta, pin al aire.
- GND: conexión a tierra.

Características

- ♦ Bajo costo.
- ♦ Alimentación en un rango de 3 a 5 Volts.
- ♦ Consumo de 2.5 mA.
- ♦ Máxima frecuencia de muestreo de 1 Hz.
- ♦ Temperatura:
 - Rango de medición de 0 °C a 50 °C.
 - Precisión a 25 °C de ± 2 °C.
 - Resolución de 1 °C (8 bit).
- ♦ Humedad:
 - Rango de medición del 20% RH a 90% RH.
 - Precisión entre 0 °C y 50 °C de $\pm 5\%$ RH.
 - Resolución de 1% RH.
- ♦ Señal de salida digital [55] [56].

Protocolo de comunicación

El DHT11 no utiliza una interfaz serial estándar como I2C, SPI o 1Wire (es similar a este último). En cambio, requiere su propio protocolo para comunicarse a través de un solo hilo. Afortunadamente el protocolo de comunicación del DHT11 es simple. [57] Véase Figura 3-8

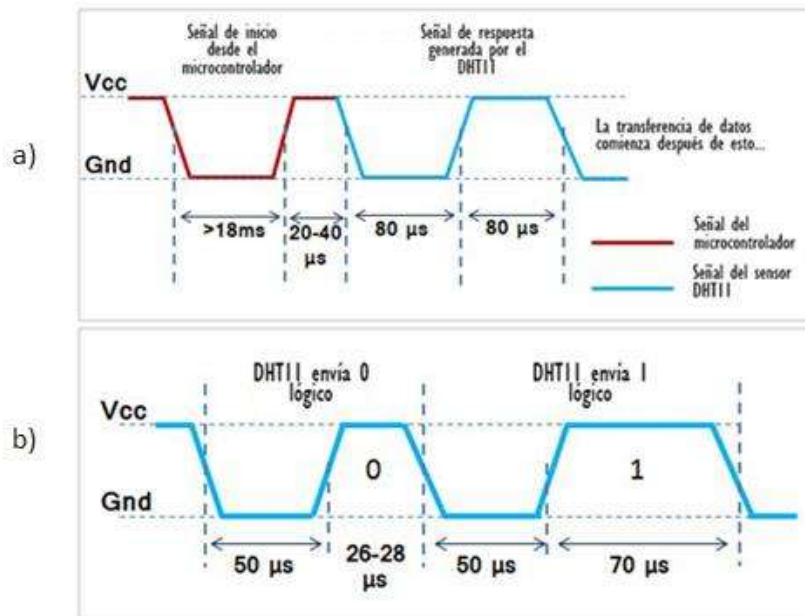


Figura 3-8 a) Diagrama de comunicación y b) diagrama de codificación [57]

La trama de datos es de 40 bits correspondiente a la información de humedad y temperatura del DHT11 como se muestra en Figura 3-9. El primer grupo de 8-bit es la parte entera de la humedad y el segundo grupo la parte decimal. Lo mismo ocurre con el tercer y cuarto grupo, la parte entera de la temperatura y la parte decimal. Por último, los bits de paridad para confirmar que no hay datos con errores. Estos bits de paridad aseguran que la información es correcta, sumando los 4 primeros grupos de 8-bit. [56]

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1001</u>
8 bits humedad	8 bits humedad	8 bits temperatura	8 bits temperatura	bits de paridad

Figura 3-9 Trama de datos DHT11 [56]

3.2.2 Sensor de distancia: HC-SR04

El sensor HC-SR04 es una excelente opción como sensor de distancia ultrasónico. Su relación de costo/beneficio lo hace óptimo para un gran abanico de aplicaciones. El uso de este módulo es bastante sencillo debido a que toda la electrónica de control, transmisión y recepción se encuentra contenida en PCB. El usuario debe enviar un pulso de disparo y medir el tiempo en alto del pulso de respuesta. Se requieren 4 señales para completar la interfaz con el módulo de sensor HC-SR04 como se muestra en la Figura 3-10. [58]

Características

- ◆ Alimentación de 5 volts.
- ◆ Interfaz sencilla: 4 señales Vcc, Trigger, Echo, GND.
- ◆ Rango de medición: 2 cm a 400 cm.
- ◆ Corriente de alimentación: 15 mA.
- ◆ Frecuencia del pulso: 40 KHz.
- ◆ Apertura del pulso ultrasónico: 15°.
- ◆ Señal de disparo: 10µs.
- ◆ Dimensiones del módulo: 45x20x15 mm.

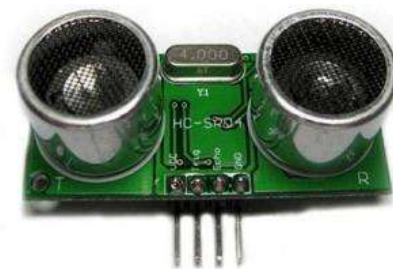


Figura 3-10 Sensor ultrasónico HC-SR04 [59]

Funcionamiento

Se conocen como ultrasonidos las ondas acústicas superiores a la máxima frecuencia auditiva perceptible por el ser humano. Nominalmente esto incluye cualquier frecuencia sobre 20 kHz, se

trabaja a una frecuencia de 40 KHz. La mayoría de los sensores de ultrasonido de bajo coste se basan en utilizar simplemente el tiempo de llegada (Time of Arrival, ToA) del pulso de ultrasonidos para calcular distancias, en caso de que el emisor y el receptor conozcan cuando se ha emitido el pulso de ultrasonidos, como por ejemplo en el caso de un sonar, ilustrado en la Figura 3-11. En este caso, para calcular la distancia es muy sencillo, sólo se ha de tener en cuenta que la señal rebota en el blanco y vuelve, con lo que recorre dos veces esta separación: [60] Véase (3-1).

$$d = \frac{V_s * t}{2} \quad (3-1)$$

Donde V_s es la velocidad del sonido y t es el tiempo que tarda la onda en llegar hasta el blanco, rebotar y volver de nuevo emisor. [60]

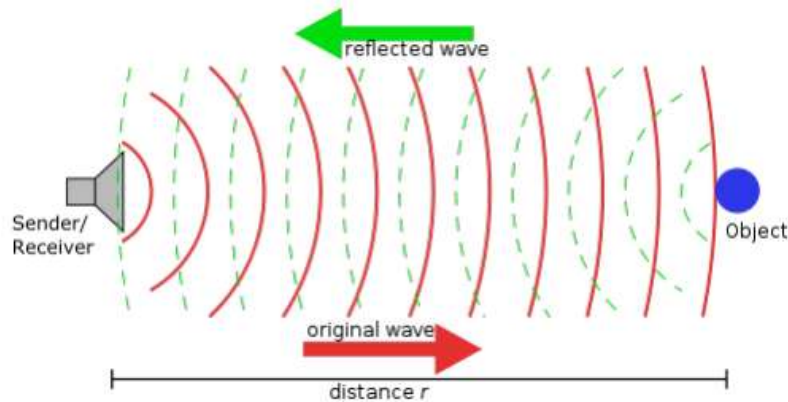


Figura 3-11 Funcionamiento de un sonar [60]

Este sensor requiere dos conductores, adicionales a la alimentación. Por uno de ellos se le envía el pulso de disparo (trigger) y por el otro recibimos el pulso de eco (echo) cuya amplitud es directamente proporcional a la distancia a la que ha sido detectado el obstáculo interpuesto. Según el siguiente diagrama de tiempo:

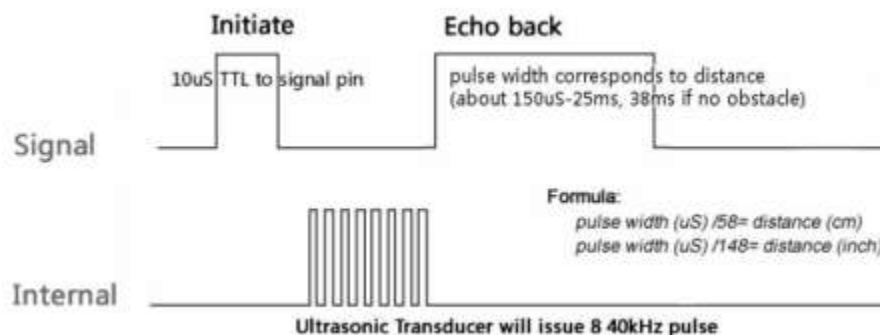


Figura 3-12 Cronograma de medición HC-SR04 [59]

Un breve impulso ultrasónico se transmite en el momento 0, reflejado por un objeto. El sensor recibe esta señal y la convierte en una señal eléctrica. El próximo pulso puede transmitirse cuando

el eco se ha desvanecido. Este período de tiempo se llama período de ciclo. El período de ciclo recomendado debe ser no menos de 50ms. La distancia medida es proporcional al ancho del pulso de eco y se puede calcular con la ecuación (3-2). Si no hay obstáculo detectado, el pin de salida dará una señal de nivel alto de 38 ms. [59]

$$distancia\ (cm) = \frac{ancho\ de\ pulso(\mu s)}{58} \quad (3-2)$$

3.2.3 Cámara

3.2.3.1 Definición

El sensor de imagen es el elemento de una cámara electrónica, tanto de vídeo como de fotografía estática, que detecta y captura la información que compone la imagen. Esto se logra al convertir la atenuación de las ondas de luz (cuando estas atraviesan o son reflejadas por cuerpos) en señales eléctricas. Las ondas capturadas por el sensor pueden ser luz u otro tipo de radiación electromagnética. Se trata de un chip formado por millones de componentes sensibles a la luz (fotodiodos o fototransistores) que al ser expuestos capturan las ondas proyectadas de un objetivo, que compone la imagen. Los sensores de imagen son utilizados en diversos dispositivos, tales como cámaras digitales, equipos médicos y equipos de visión nocturna. [61]

Cada uno de los elementos fotosensibles del sensor se denomina pixel o píxel, palabra proveniente del acrónimo inglés picture element. El número de píxeles del sensor se suele medir en millones de píxeles (o megapíxeles, Mpx). De forma general se puede decir que mayores números indican la posibilidad de imprimir (o visualizar) fotos a tamaños más grandes con pérdidas de calidad menores (mayor resolución de imagen). [61]

Otro factor importante con respecto al sensor es el tamaño y forma del mismo. Un sensor que contenga un número relativamente pequeño de píxeles debería tener una gran área por píxel; y viceversa: un sensor de menor tamaño con el mismo número de píxeles tendrá una reducida área por píxel. Los píxeles de mayor tamaño tienden a generar una mejor calidad de imagen y una mayor sensibilidad. [61]

3.2.3.2 RPI Camera Module

El módulo de la cámara RPI es un producto oficial de Raspberry Pi Foundation. El modelo original de 5 megapíxeles fue liberado en 2013 (Véase Figura 3-13), y un módulo V2 de 8 megapíxeles fue liberado en 2016. [62]



Figura 3-13 RPI Camera Module [63]

Especificaciones

En la Tabla 3-2 se muestran las características más relevantes de este módulo.

Tabla 3-2 Especificaciones RPI Camera [62]

	Módulo de cámara V1
Precio Neto	\$25
Tamaño	25 × 24 × 9 mm
Peso	3g
Resolución de imagen	5 Megapíxeles
Modos de video	1080p30, 720p60 and 640 × 480p60/90
Integración con Linux	Disponible V4L2 driver
API programada en C	OpenMAX IL y otras disponibles
Sensor	OmniVision OV5647
Resolución de sensor	2592 × 1944 píxeles
Área de imagen del sensor	3.76 × 2.74 mm
Tamaño del píxel	1.4 μm × 1.4 μm
Tamaño óptico	1/4"
Full-frame SLR lens equivalent	35 mm
Relación señal/ruido (S/N)	36 dB

Rango dinámico	67 dB @ 8x gain
Sensibilidad	680 mV/lux-sec
Corriente oscura	16 mV/sec @ 60 C
Enfoque fijo	1 m hasta infinito
Longitud focal	3.60 mm +/- 0.01
Campo horizontal de vista	53.50 +/- 0.13 grados
Campo vertical de vista	41.41 +/- 0.11 grados
Relación Focal (F-Stop)	2.9

3.3 Chasis y sistema locomotor

En este apartado se describen los componentes mecánicos que se emplearon en el proyecto actual, es decir, el chasis o estructura principal y los motores.

3.3.1 Motores de CD

El motor eléctrico es una máquina que se encarga de convertir la energía eléctrica en energía mecánica a través de la acción de los campos magnéticos producidos por sus bobinas. Los motores de corriente continua tienen un par de arranque alto comparado con los de corriente alterna, también es más fácil de controlar su velocidad, por tal motivo son eficaces en aplicaciones donde se requiera un control de velocidad. Son usados para ascensores, trenes, tranvías, automóviles eléctricos y todas aquellas aplicaciones en las que se requiere un control de velocidad constante. [64] Véase Figura 3-14



Figura 3-14 Motor de CD [64]

Un motor común de corriente directa o continua se compone de las siguientes partes o piezas:

- ◆ Carcaza.

- ♦ Rotor.
- ♦ Colector o conmutador.
- ♦ Escobillas.

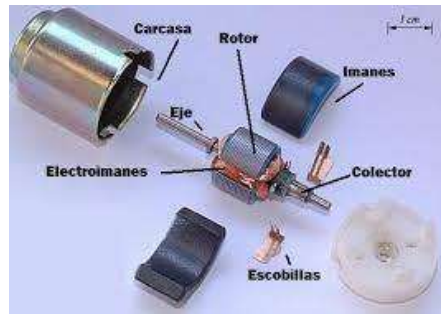


Figura 3-15 Componentes del motor de CD [64]

En el proyecto diseñado se emplearon dos motores de CD de la marca Maxon modelo 278072, para llevar a cabo la locomoción y debido al acoplo de engranes que tiene instalado nos entrega un alto par a la salida.

3.3.1.1 Motor Maxon 278072

Características:

- ♦ Velocidad de salida del eje sin carga a 6V DC: 25 RPM.
- ♦ Corriente sin carga a 6V DC: 120mA.
- ♦ Engranaje planetario incluido.
- ♦ Resistencia en terminal: 2.7 ohm.
- ♦ Encoder HEDS 5540 A11 equipado.
- ♦ Frente del engranaje con 4 orificios de 25 mm de diámetro para un fácil montaje.
- ♦ Equipado con 2 terminales para la conexión de alimentación del motor y 5 pines de conexión del encoder.
- ♦ Puede ser operado en cualquier dirección, polaridad de reversa simple.
- ♦ Peso: 396 gr. Véase Figura 3-16.



Figura 3-16 Motor Maxon 278072 [65]

3.3.1.2 Pruebas de arranque

En el instante del arranque, al estar parado el motor su velocidad es nula, la fuerza contraelectromotriz que es proporcional a la velocidad también es nula. Esto provoca que toda la tensión de alimentación energice en el devanado del inducido, en el instante del arranque la intensidad que recorre el motor es muy elevada, alcanzando valores de hasta diez veces la intensidad nominal en régimen de funcionamiento estable y más aún para motores de gran potencia. [66]

Como se menciona, en el arranque se presentan corrientes de hasta 10 veces la corriente nominal, se deben hacer pruebas en el arranque y en estado estable para conocer la corriente que consume y de esta manera seleccionar un dispositivo adecuado que maneje la potencia del motor.

La prueba consiste en arrancar el motor con carga y medir la corriente, debido a que se realiza en un pequeño instante se necesita la ayuda del osciloscopio para poder ver la forma de onda de la corriente y de una resistencia en serie de 1 ohm, en la cual se medirá la caída de tensión provocada por la corriente, el diagrama se aprecia en la Figura 3-17.

$$i = \frac{V}{R} \quad \text{si } R = 1 \rightarrow i = V \quad (3-3)$$

Como se observa en (3-3), la corriente es igual al voltaje siempre y cuando la resistencia sea 1, por tanto, al medir con el osciloscopio el voltaje en R, se está midiendo indirectamente la corriente.

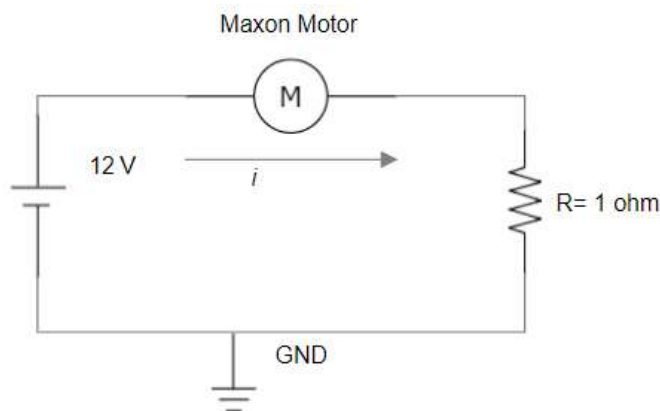


Figura 3-17 Circuito de prueba de corriente

Prueba en el arranque con carga:

Se puede observar que la corriente tiene un pico de 2.24 Amperios de amplitud, sin embargo, solo está presente por un periodo de 5 a 10 ms donde empieza a decaer por debajo de 2 Amperios. Véase Figura 3-18. En total el periodo transitorio dura aproximadamente 90 ms para llegar a un valor en estado estable de 250 mA.

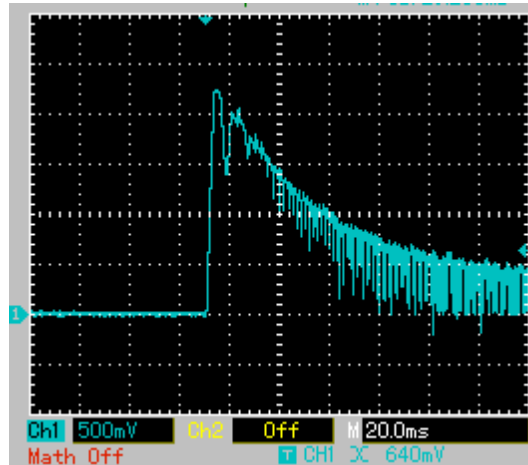


Figura 3-18 Corriente de arranque del motor

Prueba en estado estable (carga/sin carga):

La medición muestra que la corriente puede alcanzar 550 mA con carga y sin ella se encuentra aproximadamente en 250 mA. La Figura 3-19 representa la forma de onda de la corriente, se puede distinguir la curva cuando se libera la carga del motor.

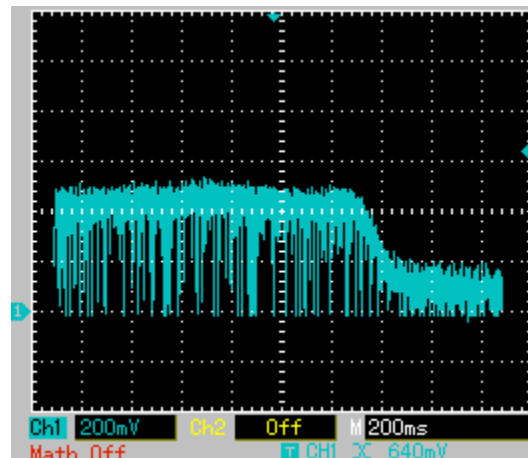


Figura 3-19 Corriente en estado estable

3.3.2 Servomotores

Un servomotor (o servo) es un tipo especial de motor con características especiales de control de posición. Al hablar de un servomotor se hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos. [67]

Son dispositivos capaces de llevar el motor a posiciones angulares específicas al enviar una señal codificada. Mientras esta señal exista en la entrada, el motor mantendrá la posición angular del engranaje. Si esta señal cambia, la posición cambia, y si desaparece, el motor deja de mantener la posición. Dentro de un servomotor hay un motor de corriente continua, una caja reductora, un potenciómetro y electrónica de control, como se aprecia en la Figura 3-20. [68]

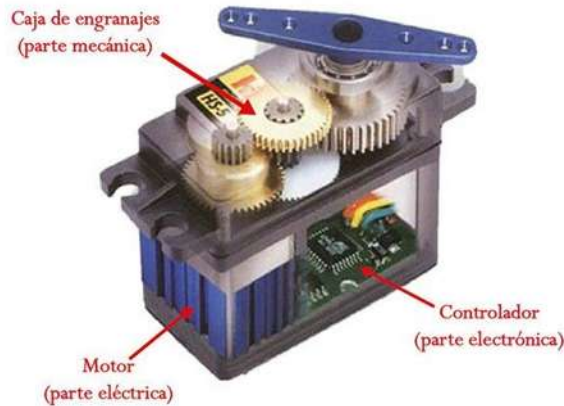


Figura 3-20 Servomotor [67]

Los servomotores de modelismo operan a voltajes bajos en corriente directa, típicamente entre 4 y 6 voltios, son el tipo más común de servomotor. Permiten una rotación de 180 grados, por lo cual son incapaces de completar una vuelta completa.

3.3.2.1 Funcionamiento

Los servomotores poseen tres cables, a diferencia de los motores comunes que sólo tienen dos. Estos tres cables casi siempre tienen los mismos colores, por lo que son fácilmente reconocibles. La señal de control es rectangular y se modula por ancho de pulso (PWM), normalmente se necesita un circuito de control para generarla.

Voltaje positivo	Tierra (ground)	Señal de control

Figura 3-21 Servomotor: Clasificación común de cables por color [67]

El diagrama de bloque del servomotor representa de forma visual el servomotor como un sistema. De acuerdo a la Figura 3-22 el circuito electrónico es responsable de recibir la señal PWM y traducirla en movimiento del Motor DC. El eje del motor DC está acoplado a un potenciómetro, el cual permite formar un divisor de voltaje. El voltaje en la salida del divisor varía en función de la posición del eje del motor DC. El potenciómetro permite que el circuito de control electrónico pueda retroalimentar con la posición del motor en un momento dado. Esto, en Teoría de Control se conoce como un sistema de lazo cerrado. [67]

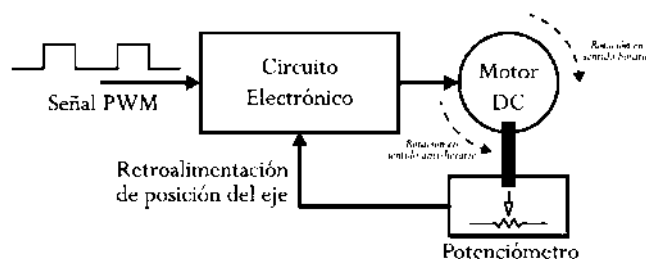


Figura 3-22 Diagrama de bloques del servomotor

Las señales que se observan en la Figura 3-23 son las que permiten que el eje del motor se coloque en determinada posición. Estas señales deben repetirse en el tiempo para que el motor mantenga una posición fija.

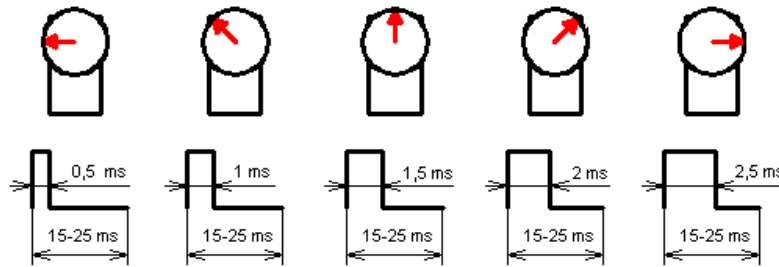


Figura 3-23 PWM para diferentes ángulos en un servomotor [67]

3.3.2.2 Servomotor sg90

Es un micro servo ligero de gran calidad y diminutas dimensiones, con alta potencia de salida. Puede rotar aproximadamente 180 grados. Los cables en el conector están distribuidos de la siguiente forma: Rojo = Alimentación (+), Café = Alimentación (-) o tierra, Naranja = Señal PWM. [69] Véase Figura 3-24



Figura 3-24 Servomotor sg90 [69]

Tomando en cuenta las prestaciones, se seleccionó este servomotor debido a que únicamente se requiere mover la base que soporta la cámara y el sensor de distancia. Este movimiento es de manera horizontal, es decir, el peso recae en el eje del servomotor y esto reduce la potencia mecánica necesaria.

Especificaciones

- Peso: 9g
- Dimensiones: 22.2 x 11.8 x 31 mm.
- Torque: 1.8 kgf-cm
- Velocidad de operación: 0.1seg/60 grados
- Voltaje de operación: 4.8 V (~5V)
- Rango de temperatura: 0 – 55 °C

Operación

De acuerdo a la Figura 3-25, el periodo de la señal de control debe ser de 20 milisegundos, es decir, se debe generar una señal PWM a una frecuencia de 50 Hz. Para obtener ángulos desde 0° hasta 180° se varía el ciclo de trabajo desde 1ms hasta 2 ms, siendo el ciclo de trabajo de 1.5 ms para obtener un ángulo de 90°.

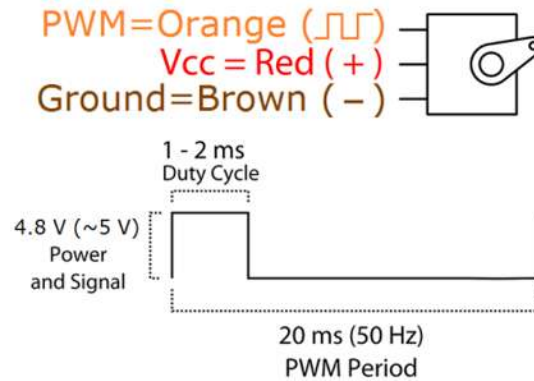


Figura 3-25 Operación servomotor sg90 [69]

3.3.3 Módulo Dual Driver de puente completo: L298N

Este módulo basado en el chip L298N permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios de corriente continua, y en modo pulsante con un tiempo en alto de 10 ms y un ciclo de trabajo de 80% soporta una corriente de hasta 2.5 amperios. Por estas características este driver es adecuado para el manejo de los motores aun con el pico de corriente de 2.24 Amperios.

El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N como se muestra en la Figura 3-26. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B). La salida A está conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4. Los pines de habilitación son ENA y ENB respectivamente. [70]

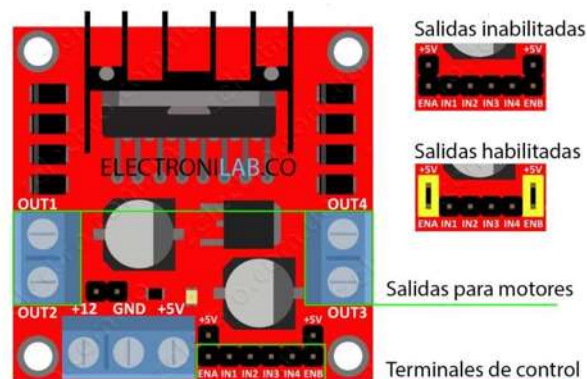


Figura 3-26 Módulo L298N [70]

En la parte inferior se encuentran los pines de control del módulo, marcados como IN1, IN2, IN3 e IN4.

Alimentación

Este módulo se puede alimentar de 2 maneras gracias al regulador integrado LM7805. Véase Figura 3-27

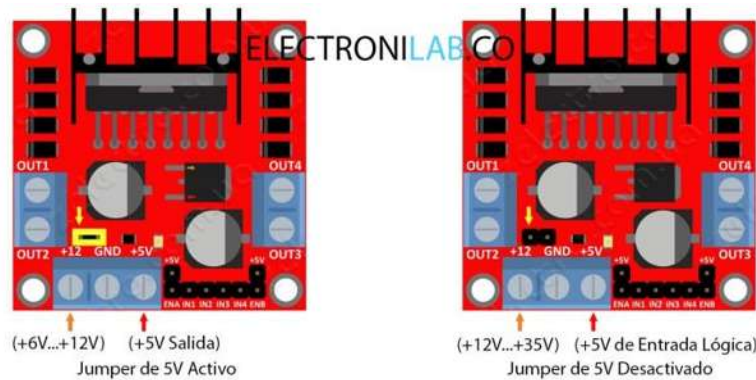


Figura 3-27 Alimentación L298N [70]

Cuando el jumper de selección de 5V se encuentra activo, el módulo permite una alimentación de entre 6V a 12V DC. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Cuando el jumper de selección de 5V se encuentra inactivo, el módulo permite una alimentación de entre 12V a 35V DC. Como el regulador no está funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte lógica del L298N. [70]

3.3.4 Optoacopladores

En la sección anterior se explicó la etapa de potencia que controla los motores usados en el robot móvil, sin embargo, para proteger la etapa de control se necesita un aislamiento entre los componentes. Una forma de lograrlo es usando optoacopladores entre las señales de control de la Raspberry Pi y las entradas del módulo L298N, de esta forma si existe alguna perturbación en el voltaje o corriente de la etapa de potencia, no afecta nuestra etapa de control.

Un optoacoplador también llamado optoaislador, es un circuito electrónico que funciona como un interruptor aislado ópticamente. Es decir, permite una conexión eléctricamente aislada entre dos circuitos que operan a distintos voltajes. Está construido por un led y un circuito de control activado por luz infrarroja. Entre otras cosas, una de las ventajas principales de los optoacopladores es su aislación eléctrica entre la carga y la electrónica de control. La única conexión entre ambos elementos es la luz del led que activa al foto-transistor. La Figura 3-28 muestra un diagrama general para un optoacoplador con salida a foto-transistor. [71]

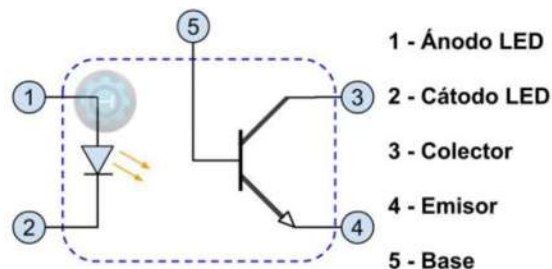


Figura 3-28 Optoacoplador: diagrama general [71]

Un optoacoplador está diseñado con dos elementos principales. El primero es un LED infrarrojo, este dispositivo activa “remotamente” al opto-transistor. El segundo elemento es el dispositivo electrónico de control. Dependiendo del tipo, este puede ser un opto-transistor, un opto-TRIAC, un transistor Darlington, SCR o una compuerta digital. Finalmente, el propósito del led es activar al elemento de control. [71]

Las aplicaciones de optoacopladores incluyen el activar cargas que puedan inducir ruido eléctrico al sistema de control. Cuando una carga inductiva, como un motor, se activa y desactiva produce perturbaciones eléctricas en la alimentación del sistema. Los optoacopladores se usan para aislar a estas perturbaciones.

3.3.4.1 Optoacoplador PC817

Este dispositivo contiene un LED infrarrojo acoplado ópticamente a un fototransistor y viene empaquetado en un DIP de 4 pines. Véase Figura 3-29

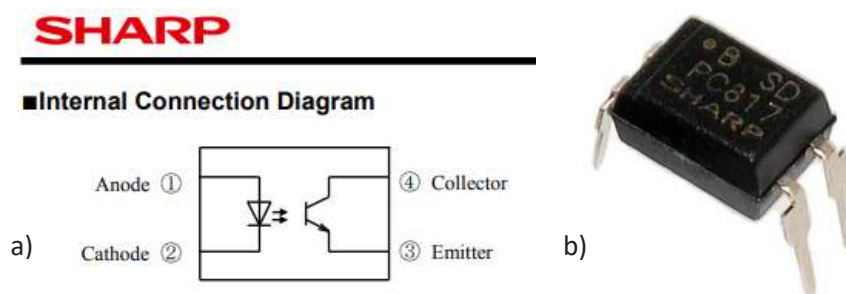


Figura 3-29 Sharp PC817: a) diagrama [72] y b) componente [73]

Características:

- Voltaje de aislamiento alto entre la entrada y salida (5kV rms)
- Voltaje de colector-emisor alto (V_{ce0} : 80V)
- Relación de transferencia de corriente (CTR: min 50% a $I_f=5mA$, $V_{ce0}=5V$)
- 1 canal de aislamiento

3.3.5 Chasis

La base del robot es su cuerpo. Este cuerpo puede ser de varias formas, colores y materiales, pero siempre debe ser capaz de soportar y mover el resto de los componentes del robot. Este chasis puede ser móvil, puede ser modular, con niveles, de madera, de acero, etc. Véase Figura 3-30. Las características de los materiales y diseños pueden influir en el funcionamiento del mismo. [68]

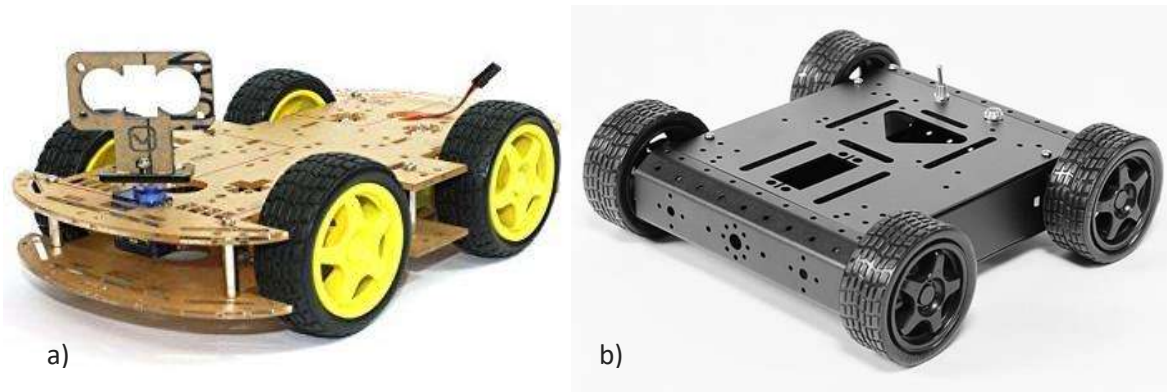


Figura 3-30 a) Chasis de acrílico [74] y b) chasis de aluminio [75]

3.3.5.1 Diseño y construcción

Para el desarrollo del chasis se necesita de un material resistente debido a las condiciones donde se desempeñará el robot, además para implementar un proyecto ecológico se hizo uso de material reciclado, se optó por usar metal galvanizado, que comúnmente se emplea en la construcción y es fácil encontrar sobrantes en las empresas constructoras.

Para construir el marco se utilizaron dos tramos de 26 cm y 2 tramos de 45 cm de largo y ambos con 7 cm de alto, los cuales se atornillaron de tal manera que formaran un rectángulo, como se muestra en la Figura 3-31. Aunado a esto, se colocó una lámina de triplay de la misma medida que sirve de base para los componentes.



Figura 3-31 Estructura del chasis

Una vez asegurada la estructura, se fijaron los motores en la parte trasera y se colocaron las llantas tanto en los motores como en la parte delantera, proceso mostrado en la Figura 3-32.



Figura 3-32 Chasis-sistema locomotor

Las llantas son recicladas de un juguete a control remoto y se acoplaron al motor con ayuda de adaptadores de bronce, Figura 3-33.



Figura 3-33 a) Acopladores [76] y b) llantas usadas

3.4 Batería (Acumulador)

Un componente esencial de cualquier robot móvil es la fuente de poder, encargada de proveer la energía a todos los sistemas. En este caso se usa una batería, debido a que el robot utiliza tecnología inalámbrica.

Una batería eléctrica o acumulador eléctrico, es un dispositivo que consiste en una o más celdas electroquímicas que pueden convertir la energía química almacenada en corriente eléctrica. Cada celda consta de un electrodo positivo, o ánodo, un electrodo negativo, o cátodo, y electrolitos que permiten que los iones se muevan entre los electrodos, permitiendo que la corriente fluya fuera de la batería para llevar a cabo su función, alimentar un circuito eléctrico. [77]

3.4.1 Principio de funcionamiento

El principio de funcionamiento de un acumulador está basado esencialmente en un proceso químico reversible llamado reducción-oxidación (también conocida como redox), en el que uno de

los componentes se oxida (pierde electrones) y el otro se reduce (gana electrones); es decir, un proceso cuyos componentes no resulten consumidos ni se pierdan, sino que meramente cambian su estado de oxidación y, que a su vez pueden retornar a su estado original en las circunstancias adecuadas. Estas circunstancias son, en el caso de las baterías y pilas recargables, el cierre del circuito externo durante el proceso de descarga y la aplicación de una corriente externa durante la carga. [77]

3.4.2 Parámetros de un acumulador

3.4.2.1 Tensión

La tensión o diferencia de potencial (en voltios) es el primer parámetro a considerar, pues es el que suele determinar si el acumulador es adecuado para el uso al cual se le destina. Viene fijado por el potencial de reducción del par redox utilizado; suele estar entre 1 V y 4 V por elemento.

3.4.2.2 Intensidad de corriente

Es la carga eléctrica neta (medida en culombios) transferida a través de una sección transversal de un conductor por unidad de tiempo. Suele expresarse en amperios (A).

3.4.2.3 Capacidad

La capacidad de carga o capacidad del acumulador es la carga que puede almacenar el elemento. Se mide en amperios-hora (Ah) y es el segundo parámetro a considerar. En las baterías de baja capacidad de carga, suele expresarse en miliamperios-hora (mAh).

Una capacidad de carga de 1 amperio-hora significa que la batería puede suministrar una intensidad de corriente de 1 A durante 1 hora antes de agotarse. Entre una batería o pila de 1200 mAh y otra de 2400 mAh, la segunda durará más tiempo (el doble) porque tiene más carga eléctrica almacenada. [77]

3.4.3 Batería Steren® BR-1207



Figura 3-34 Batería BR-1207

Para esta aplicación se usó una batería recargable de ácido-plomo modelo BR-1207 con un voltaje nominal de 12 V (Figura 3-34). Este tipo de baterías son comúnmente usadas en motonetas o UPS® Backups debido a su gran capacidad de energía. Por este motivo se eligió para el proyecto principalmente y considerando que los motores trabajan a esa diferencia de potencial.

4

Capítulo 4. Diseño y descripción de software

En este capítulo se describe el diseño del software que es responsable de realizar las actividades desempeñadas por el robot, desde la visualización de información en el sitio web y la comunicación a través de la red hasta el manejo de los actuadores y la lectura de los sensores mediante JavaScript. Véase Figura 4-1

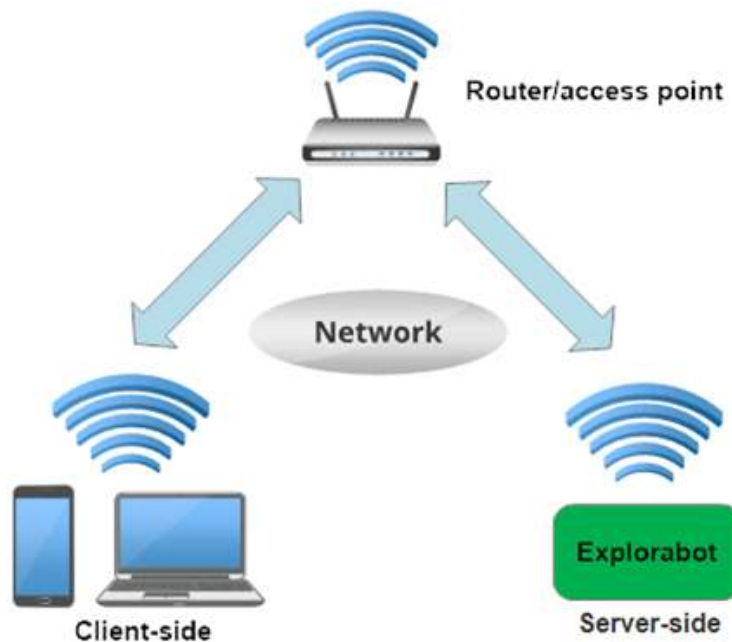


Figura 4-1 Diagrama de estructura general

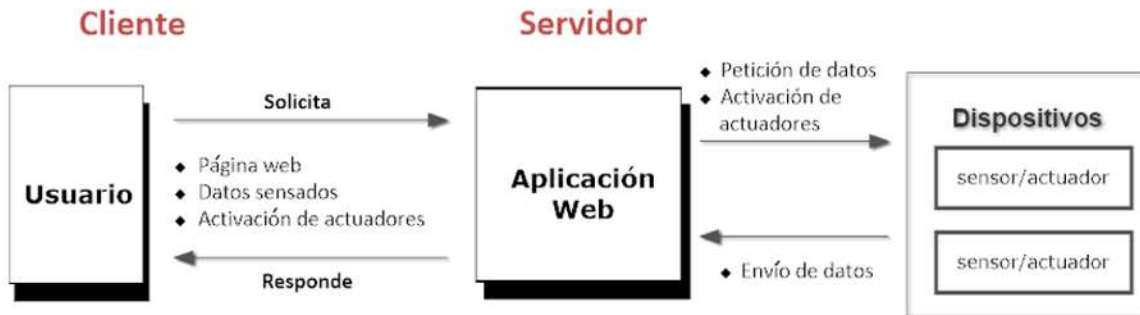


Figura 4-2 Diagrama de flujo de información

De acuerdo a la Figura 4-2, el usuario (cliente) realiza una petición HTTP mediante el navegador que recibe y procesa la aplicación web (servidor). El servidor crea una respuesta y la envía de regreso al navegador, comúnmente se envía la página inicial si la petición hace referencia a la ruta raíz (/).

En diseño de software, el front-end es la parte del software que interactúa con los usuarios y el back-end es la parte que procesa la entrada desde el front-end. La idea general es que el front-end sea el responsable de recolectar los datos de entrada del usuario, y los transforme de acuerdo a las especificaciones que demanda el back-end para procesarlos, regresando generalmente una respuesta que el front-end recibe y expone al usuario de una forma entendible para este. [78]

Se comienza describiendo la parte de front-end (lado cliente), para después dar paso a la parte operativa del sistema (lado servidor).

4.1 Diseño de sitio web e interfaz de usuario (Front-end)

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. [79] En este caso, la interfaz de usuario fue implementada en un sitio web y es presentada mediante un navegador, esto implica en términos generales utilizar HTML, CSS y JavaScript.

HTML + CSS + JavaScript

Un documento suele ser un archivo de texto estructurado usando un Lenguaje de marcado (HTML es el más común, pero hay otros como SVG o XML). [80] El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<, >, /). HTML también puede describir, hasta un cierto punto, la apariencia de un documento. [81] Por ejemplo:

```
<h2>Contenido</h2>
```

Establece el *contenido* como un encabezado de segundo nivel, el cual se muestra en el navegador de una manera similar al título.

El marcado estructural describe el propósito del texto, aunque no define cómo se ve el elemento, debido a esto la mayoría de los navegadores web han estandarizado el formato de los elementos. Puede aplicarse un formato específico al texto por medio de *hojas de estilo en cascada* o CSS por sus siglas en ingles. [81]

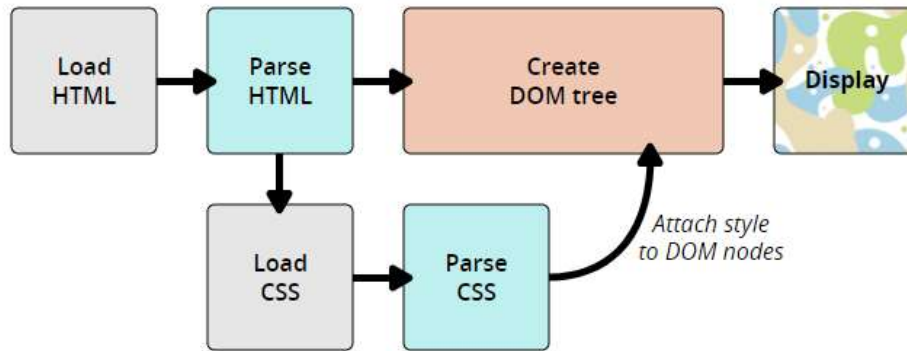


Figura 4-3 Funcionamiento de CSS [80]

En la Figura 4-3 se muestra el proceso que realiza el navegador para desplegar el documento descrito por los archivos HTML y CSS.

Primero el navegador convierte HTML y CSS en un DOM (Document Object Model). Este DOM representa el documento en la memoria del ordenador. Combinando el contenido del documento con su estilo. Y después el navegador muestra el contenido del DOM. [80]

Como ya se mencionó, el navegador nos devuelve un DOM que describe el contenido del documento como un conjunto de objetos y es justo de esta manera que JavaScript puede actuar sobre ellos. El DOM permite un acceso a la estructura de una página HTML mediante el mapeo de los elementos de esta página en un árbol de nodos. Afortunadamente, JavaScript permite acceder a cada uno de los elementos de la página utilizando tan sólo algunos métodos y propiedades. [82] Estas funciones hacen posible los efectos sobre páginas web, como crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color, etc. Por otro lado, JavaScript también permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos) con lo que se puede crear páginas interactivas con programas como calculadoras, agendas, hojas de cálculo, etc. [49]

4.1.1 Estructura general de la página

Frecuentemente, los sitios web muestran el contenido en múltiples columnas (como una revista) véase Figura 4-4, de esta manera cada elemento está posicionado de acuerdo al diseño empleado, y se obtiene una página más organizada y limpia.

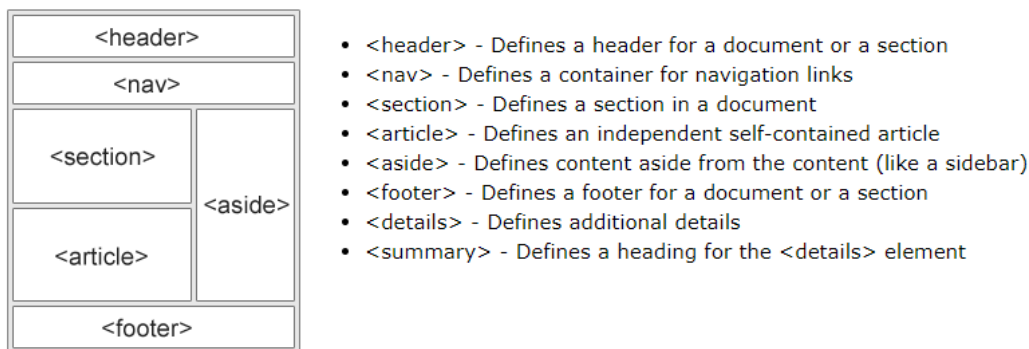


Figura 4-4 Plantilla de página web

Existen diferentes técnicas para la creación de plantillas multicolumnas, no obstante, las más utilizadas se basan en CSS, que como ya se mencionó es un lenguaje usado para diseñar la forma de presentar la página, y son:

- ◆ CSS float property
- ◆ CSS flexbox
- ◆ CSS Framework

En este trabajo se creó la plantilla mediante *CSS float property*, como su nombre lo dice, nos permite posicionar elementos mediante la propiedad de flotar, es decir, se posicionan relativamente al documento.

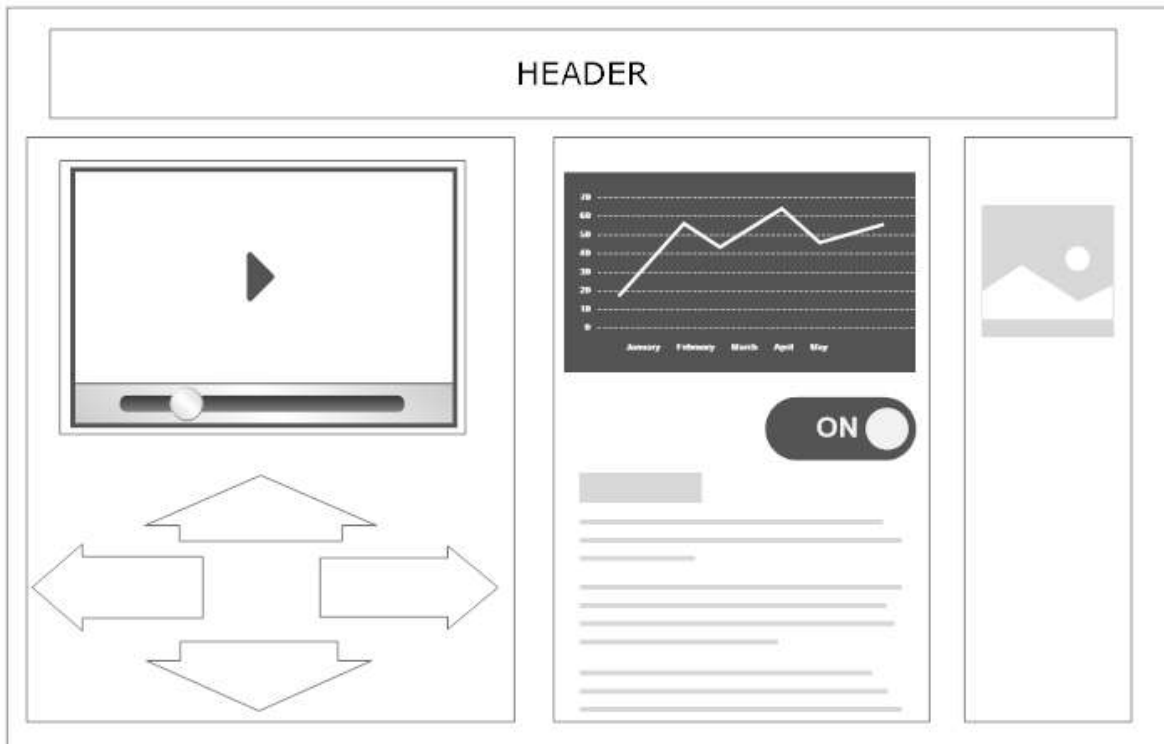


Figura 4-5 Plantilla de interfaz de usuario

En la Figura 4-5 se muestra la estructura de la interfaz de usuario, como se aprecia cuenta con una ventana de video, donde se visualiza la transmisión en vivo, un pad direccional con el que se controla el movimiento del robot y una sección para visualizar los parámetros de los sensores. Estos objetos serán explicados con mayor detalle posteriormente.

4.1.2 Elementos de la página

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo, `<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>`). Algunos elementos, tales como `<br>`, no tienen contenido ni llevan una etiqueta de cierre. Véase Figura 4-6

El marcado estructural describe el propósito del texto. Por ejemplo, `<h2>Golf</h2>` establece «Golf» como un encabezamiento de segundo nivel. [81]

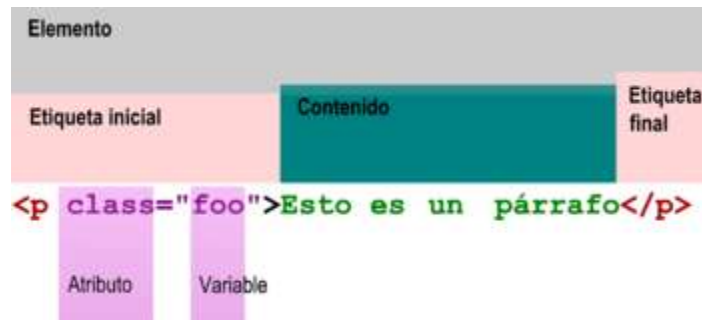


Figura 4-6 Estructura de etiquetas de elementos HTML [81]

En su mayoría, los atributos de un elemento son pares nombre-valor, separados por un signo de igual «=» y escritos en la etiqueta de inicio de un elemento, después del nombre del elemento. [81] Por ejemplo:

```

```

`<img>`: imagen. Requiere del atributo `src`, que indica la ruta en la que se encuentra la imagen.

Los atributos de los elementos son un requisito importante para el funcionamiento de la página, debido a que una vez procesados por el navegador, pasan a ser propiedades del propio objeto (elemento) y además permiten identificarlos para después poder manejarlos mediante CSS o JavaScript.

4.1.2.1 Ventana de video

Para lograr la visualización del video en vivo, se hizo uso del elemento `<iframe>`, el cual permite mostrar una página web, como un bloque, dentro de otra página web. Es decir, se puede direccionar a un documento HTML o se puede direccionar a una dirección URL para ser mostrada.

Los atributos más relevantes del elemento son `src`, `width` y `height`, que definen la fuente de la página web, el ancho y el alto del bloque en pixeles, respectivamente. Véase Figura 4-7

```
<iframe src="demo_iframe.htm" height="200" width="300"></iframe>
```

Figura 4-7 Tag `<iframe>`

De esta manera se creó un elemento `<iframe>`, con el cual podemos obtener el video independientemente de la página principal, y se enlazo con la dirección web donde se encuentra la transmisión del video, véase Figura 4-8. Los atributos del ancho y alto fueron fijados por el ancho y alto del video transmitido.



Figura 4-8 Ventana de video

De acuerdo a la Figura 4-8, en la parte inferior se cuenta con un “slider” el cual consta de un botón móvil que se desliza en una barra con un rango configurado. Este slider se usa para establecer un ángulo de visión de la cámara en el eje horizontal, como se mencionó en la sección 3.3.2.2, mediante el movimiento del servomotor.

La transmisión de video es realizada por un servidor que se ejecuta paralela e independientemente del servidor principal, es por esta razón que podemos enlazar el elemento `<iframe>` con la dirección del servidor de video.

4.1.2.2 D-Pad para el movimiento del robot

Elemento Input (entrada): los elementos HTML `<input>`, especifican un campo de entrada donde el usuario puede introducir datos. Un campo de entrada puede variar de muchas formas, dependiendo del tipo de atributo que se le asigne. [83]

Para la construcción del D-Pad (Pad de cuatro direcciones) se utilizaron elementos input del tipo “button” los cuales definen gráficamente un botón. Véase Figura 4-9. Como se mencionó en la sección 4.1, una vez procesada la página web, este elemento se convierte en un objeto y esto permite programar acciones con JavaScript. Estas acciones son disparadas por eventos, y estos a su vez son producidos por el objeto, por ejemplo, cuando el usuario presiona un botón con el mouse se genera el evento “onclick”. [84]



Figura 4-9 Pad direccional

Existen diversos eventos generados por los objetos, en nuestro caso para controlar el robot, se necesita realizar una acción cuando el usuario presione un botón (activar los motores) y otra cuando se deje de presionar el botón (detener los motores), por esta razón se emplearon los eventos “onmousedown” y “onmouseup” respectivamente.

La acción realizada por el evento cuando se presiona el botón, es llamar a una función que genera una petición HTTP al servidor y mediante la dirección URL se envía la información acerca de cuál botón ha sido presionado. Una vez liberado el botón se genera otra acción que llama a la misma función, pero cambiando la información que se envía en el URL. Estas peticiones las procesa el servidor web y tras analizarlas realiza una determinada acción que depende de la información recibida.

4.1.2.3 Presentación de variables

Para realizar la presentación de los valores medidos por los sensores, se necesita de una página dinámica que muestre los cambios sin necesidad de recargar la página, esto se logra, al igual que en el caso del D-Pad, mediante peticiones HTTP y AJAX.

Para obtener las mediciones constantemente, se programó una rutina en JavaScript, la cual mediante un ciclo realiza las peticiones para la temperatura y humedad cada 3 segundos y una vez por segundo para la proximidad de los objetos, para un mayor control estas peticiones son controladas por un botón del tipo “on-off”. Cuando el botón se encuentra en estado “on” se generan las peticiones que son manejadas por el servidor y una vez procesadas genera una respuesta y la envía de vuelta al navegador.

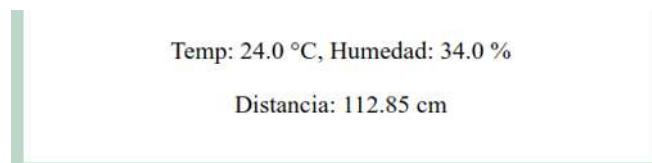


Figura 4-10 Visualización de mediciones

Una vez que el navegador recibe la información de los sensores, la muestra en forma de texto mediante el elemento `<p>` párrafo, como se observa en la Figura 4-10. Para que la interfaz sea más atractiva al usuario, se agregaron dos gráficas dinámicas, una muestra la evolución de la temperatura y humedad y la segunda la proximidad de los objetos.

Esta gráfica se logra mediante el uso de la biblioteca “High Charts” que permite crear gráficos de una manera sencilla, pero con una visualización interactiva y dinámica. Véase Figura 4-11

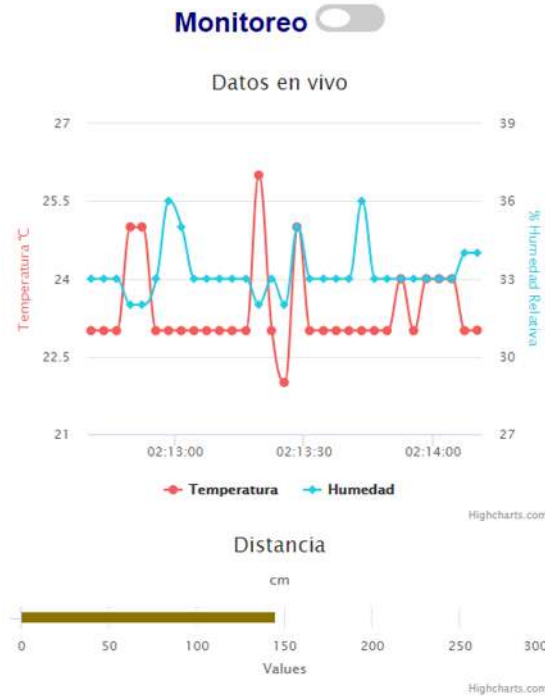


Figura 4-11 Gráfica de variables - "High Charts"

4.2 Diseño y funcionamiento del servidor (Back-end)

En la sección 4.1 se explicó el concepto de "Front-end", que es la interfaz con el usuario, pero esta no tendría sentido sin una parte operativa que realice todas las funciones. Esta sección del software se denomina "Back-end" y se realiza en el servidor.

Entre los lenguajes más populares de Back-end se encuentran Ruby, Python, SQL, PHP y Java. En el presente proyecto se desarrolló el servidor en lenguaje JavaScript ejecutándose sobre Node.js, posteriormente abordaremos Node.js para una mayor descripción.

La función del servidor comúnmente es acceder a información que se solicita, adecuarla y devolverla al usuario final, sin embargo, en términos generales el servidor realiza alguna acción dependiendo de la petición del usuario. A través de este funcionamiento se programa un servidor HTTP, que recibe las peticiones del navegador por un puerto específico, es decir, el servidor es una aplicación que se ejecuta en un computador y que está escuchando en un puerto las peticiones del usuario.

Estas peticiones realizadas por el navegador, tienen 2 elementos principales: la dirección URL a la que se quiere acceder y el método que se usa para la petición. El servidor puede contar con un servicio de direccionamiento, la dirección URL es la que especifica la ruta a la cual se quiere acceder, y dependiendo de esta, el servidor formula una acción en respuesta. El segundo parámetro, el método, es uno de los métodos admitidos por el protocolo HTTP, en este caso se usó el método "GET" debido a su simple procesamiento.

Por ejemplo, para las peticiones generadas por los botones del Pad direccional, se usó el URL "move" y para identificar que botón había sido presionado, se agregó un parámetro al URL:

“move?m=up” para el caso del botón superior. Esto significa que en el servidor debe existir una ruta llamada “/move” que manejará esa petición, además se debe analizar el URL entrante para obtener el valor del botón presionado, una vez hecho esto, el servidor debe realizar la acción adecuada, y esto se logra mediante la estructura “Switch-case” usando como parámetro el valor recibido.

Además del servicio de direccionamiento, el servidor diseñado puede servir archivos estáticos guardados localmente, tales como archivos JavaScript, HTML, CSS, imágenes, documentos, etc. Estos archivos son de gran ayuda para la estética del sitio web, por ejemplo, el archivo CSS o imágenes que pueden cambiar la vista de la página.

4.2.1 Servidor de video *Motion*

Una de las principales características del proyecto, es la visualización en tiempo real del video que transmite el robot móvil, puesto que es tele-operado, sin este recurso sería difícil el control de movimiento del mismo. Para realizar esta transmisión se hizo uso del programa “Motion” el cual permite convertir una variedad de cámaras conectadas a nuestro computador, en servidores de video y aunque como su nombre lo dice, es usado principalmente para detectar movimiento, se puede desactivar esta función y dejarlo simplemente como un software que emula una cámara ip.

Este software permite personalizar en gran medida aspectos acerca de la transmisión, como las dimensiones del video, la tasa de captura (FPS), la tasa de transmisión (FPS), el puerto por el cual se transmite, etc. Todo esto se especifica en un archivo de configuración del programa y además tiene la ventaja de poder ejecutarse al arrancar el computador automáticamente.

Una vez instalado y configurado (Véase Apéndice 1: Instalación de servidor de video Motion), basta con iniciar el servicio para comenzar la difusión del video en la red y consultar la dirección ip del dispositivo especificando el puerto que se definió en el archivo de configuración. Véase Figura 4-12



Figura 4-12 Visualización de video

4.2.2 Manejo de peticiones para envío de datos

En esta sección se describe con mayor detalle el mecanismo de comunicación entre el usuario y el robot móvil (cliente-servidor) mediante el uso de AJAX (Asynchronous JavaScript And XML).

AJAX solo usa una combinación de:

- El objeto XMLHttpRequest incluido en el navegador (Para solicitar datos del servidor web)
- JavaScript y HTML DOM (Para mostrar o usar los datos)

AJAX permite a las páginas web ser actualizadas asincrónicamente mediante el intercambio de datos con el servidor web en un segundo plano. Esto significa que es posible actualizar partes de una página sin la necesidad de recargar la página completa. [85] En la Figura 4-13 se muestra el proceso de comunicación entre el cliente y servidor mediante AJAX.

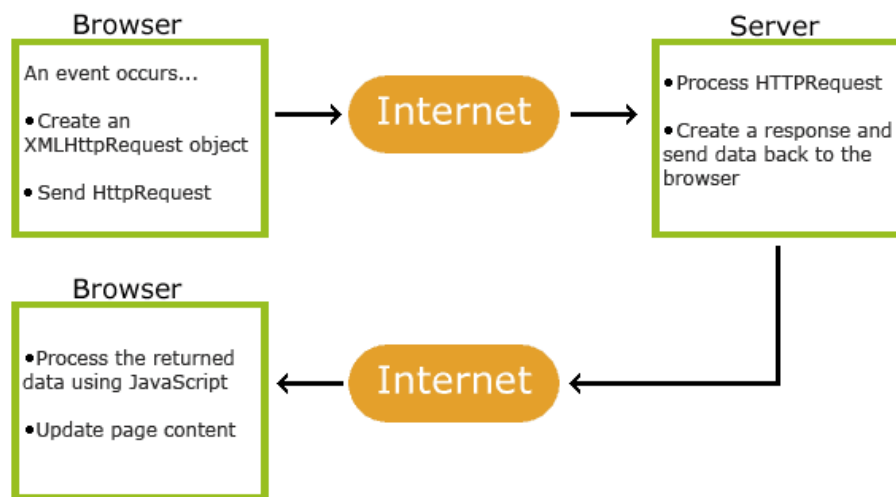


Figura 4-13 AJAX: funcionamiento [85]

Este esquema de comunicación es implementado para el intercambio de información entre el Front-end y el Back-end. En la Figura 4-13 se muestra como medio de comunicación internet, sin embargo, no es necesario tener acceso a este, porque se puede realizar la comunicación mediante una red de área local inalámbrica (WIFI).

4.3 Programación en JavaScript usando Node.js

Dentro de este proyecto se realiza la programación para el manejo y monitoreo de las variables del robot mediante JavaScript ejecutándose gracias a Node.js. Como se mencionó en el segundo capítulo en la sección 2.5.3.3, JavaScript es un lenguaje de programación interpretado orientado a objetos, su uso principal es del lado del cliente (client-side) implementado en un navegador web permitiendo sitios web dinámicos, sin embargo, también existe una forma de JavaScript en el lado del servidor (server-side). Este permite que todo lo que está detrás de un sitio web funcione correctamente. Y aunque Node.js es capaz de servir para el desarrollo web del lado servidor no se limita solo a esa función, debido a que es un entorno de ejecución de propósito general.

Esto se logra con mayor facilidad gracias a los paquetes o bibliotecas disponibles, los cuales tienen un amplio campo de aplicación. Un ejemplo es Express, infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.

Un ejemplo básico “Hello World” de un servidor HTTP en Node.js se puede realizar haciendo uso del paquete “express”:

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Figura 4-14 Ejemplo de servidor HTTP [86]

La aplicación inicia un servidor y escucha las conexiones en el puerto 3000. La aplicación responde con “Hello World!” para las solicitudes en la URL raíz (/) o a la ruta raíz. Para cada vía de acceso no implementada, responderá con un error 404 Not Found. [86]

El código se guarda con extensión “js” por ejemplo “app.js” y para ejecutar la aplicación se usa el siguiente comando:

```
$ > node app.js
```

A continuación, se introduce la siguiente dirección `http://localhost:3000/` en un navegador para ver la salida.

Como se aprecia, Node.js facilita realizar aplicaciones, en colaboración con los paquetes instalables. Estos paquetes están disponibles desde un ecosistema de Node.js llamado NPM, por sus siglas en inglés (Node Package Manager), que a continuación se describe.

4.3.1 Node Package Manager (NPM)

El ecosistema de paquetes de Node.js, NPM, es el ecosistema más grande de bibliotecas de código abierto en el mundo. [50] Este consiste de 3 distintos componentes:

- ◆ El sitio Web: Usado para descubrir paquetes, configurar perfiles etc.
- ◆ La interfaz de línea de comandos (CLI): Esta se ejecuta desde una terminal y es como la mayoría de los desarrolladores interactúan con NPM.
- ◆ El registro: La base de datos pública de JavaScript y la meta-información que lo rodea.

4.3.1.1 *Interfaz de línea de comandos*

El componente que nosotros usamos es la interfaz de línea de comandos, desde ahí se puede iniciar un proyecto npm:

Desde la terminal se ejecuta el siguiente comando que nos lanza un asistente para configurar nuestro proyecto:

```
$ > npm init
```

Al finalizar, el asistente crea un archivo llamado “package.json”, este es el archivo de configuración de un proyecto node.js

Dentro de este archivo se definen y manejan características como:

- ◆ Nombre de tu proyecto.
- ◆ Versión.
- ◆ Dependencias.
- ◆ Repositorio.
- ◆ Autores.
- ◆ Licencia.

A través de este archivo, se puede garantizar la integridad del proyecto. Es decir, podemos asegurar que quienes tengan una copia del mismo, podrán acceder a las mismas propiedades y sincronizar entre múltiples partes cada vez que decidan hacer un cambio. De cierta forma, podemos considerar este package.json como un manifiesto de nuestro proyecto.

4.3.1.2 *Paquetes NPM*

Existen diferentes opciones en la instalación de paquetes [87]:

```
$ > npm install [nombre del paquete]
```

En esta opción se hace una instalación única, es decir, se instalará en la carpeta donde se encuentra la terminal de comandos y no se registra en el archivo package.json

```
$ > npm install [nombre del paquete] --save
```

Se instalan en la carpeta donde se encuentra la terminal y se registran en el archivo package.json como dependencias del proyecto, es decir, que el proyecto requiere estos paquetes para funcionar.

```
$ > npm install [nombre del paquete] --global
```

Se instalan localmente en el ordenador independientemente de donde se encuentre la terminal de comandos, esto significa que estarán disponibles en todas las carpetas del ordenador, NO se registran en el archivo package.json

El archivo package.json es de gran utilidad, cuando un proyecto cuenta con este archivo se pueden instalar todas las dependencias de proyecto y desarrollo que tenga registradas en el mismo ejecutando el siguiente comando.

```
$ > npm install
```

4.3.2 Lectura de sensores

En la sección anterior se describió la herramienta que nos proporciona Node.js para la descarga de paquetes, NPM, con este se instalaron paquetes que nos permiten la lectura y escritura de los pines GPIO de la raspberry.

La primera biblioteca que se instaló fue “node-dht-sensor”, la cual permite la comunicación con los sensores DHT, en este caso específicamente con el DHT11 como se aprecia en la Figura 4-15, y mediante una función regresa las mediciones para enviarlas al usuario final.

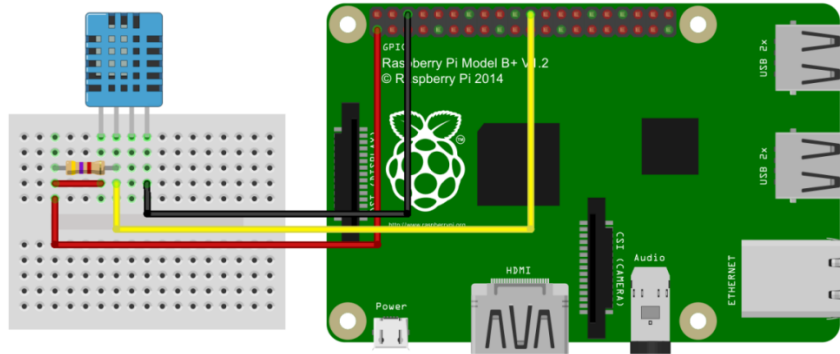


Figura 4-15 Diagrama de conexión con sensor DHT11 [88]

La Figura 4-16 muestra el código de la función empleada para la lectura del sensor DHT22, en nuestro caso solo modificamos el parámetro “22” por un “11” para especificar el tipo de sensor usado.

```
var sensor = require('node-dht-sensor');

sensor.read(22, 4, function(err, temperature, humidity) {
  if (!err) {
    console.log('temp: ' + temperature.toFixed(1) + '°C, ' +
      'humidity: ' + humidity.toFixed(1) + '%');
  }
});
```

Figura 4-16 Código para lectura de sensor DHT [89]

Otro paquete que se requiere es “pigpio”, este permite controlar cualquier pin GPIO, ya sea de entrada o salida, además tiene funciones para PWM o interrupciones en los pines. Para funcionar hace uso de la biblioteca pigpio escrita en C la cual está preinstalada por Raspbian en la raspberry.

Al manejar los pines gpio se puede implementar la medición del sensor HC-SR04, primeramente, generando un pulso de 10 microsegundos en el pin “trigger” y después midiendo el tiempo que está en alto el pin “echo” el cual es proporcional al tiempo que tarda en regresar la señal ultrasónica. Véase Figura 4-17

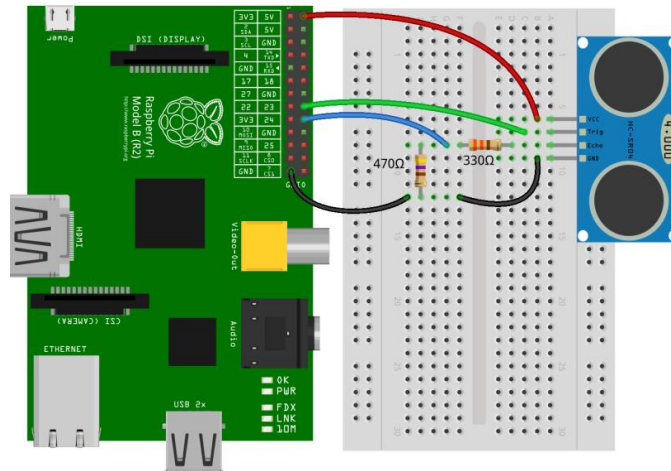


Figura 4-17 Diagrama de conexión sensor HC-SR04 [90]

Para obtener las mediciones constantemente se programa una rutina para que cada periodo de tiempo se ejecute y devuelva los valores medidos. Con esto permitimos que el servidor no se bloquee y ejecute acciones mediante eventos.

4.3.3 Activación de actuadores mediante GPIO

Como se mencionó en la sección 4.3.2, la biblioteca “pigpio” también permite escribir los pines de la raspberry, es decir que tengan dirección de salida, es por esta característica que se usó para la activación de los actuadores (motores).

Ejemplo:

```

a) var Gpio = require('pigpio').Gpio,
    led = new Gpio(17, {mode: Gpio.OUTPUT}),
    dutyCycle = 0;

    setInterval(function () {
        led.pwmWrite(dutyCycle);

        dutyCycle += 5;
        if (dutyCycle > 255) {
            dutyCycle = 0;
        }
    }, 20);
    
```

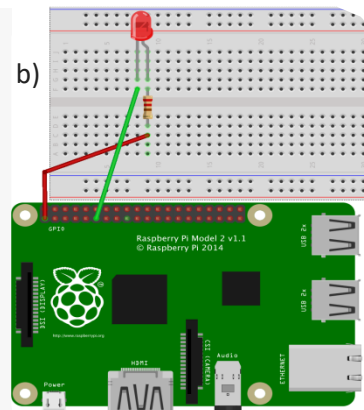


Figura 4-18 a) Código: encender led mediante PWM [90] y b) diagrama de conexión [91]

En la Figura 4-18 se muestra el código empleado para controlar la intensidad de un led aumentando gradualmente su ancho de pulso (PWM) y el diagrama de conexión realizado para su funcionamiento, esto usando el paquete antes mencionado.

Como se puede ver, una de las ventajas de usar Node.js, es la facilidad de programación que nos ofrece gracias a JavaScript.

5

Capítulo 5. Pruebas y resultados

En los capítulos anteriores se describió el hardware y el software del sistema robótico, sin embargo, se hizo de manera independiente y sin comunicación entre ambos. En este apartado se presenta el sistema ya integrado (Figura 5-1), así como las pruebas aplicadas tanto al hardware como al software y sus respectivos resultados.

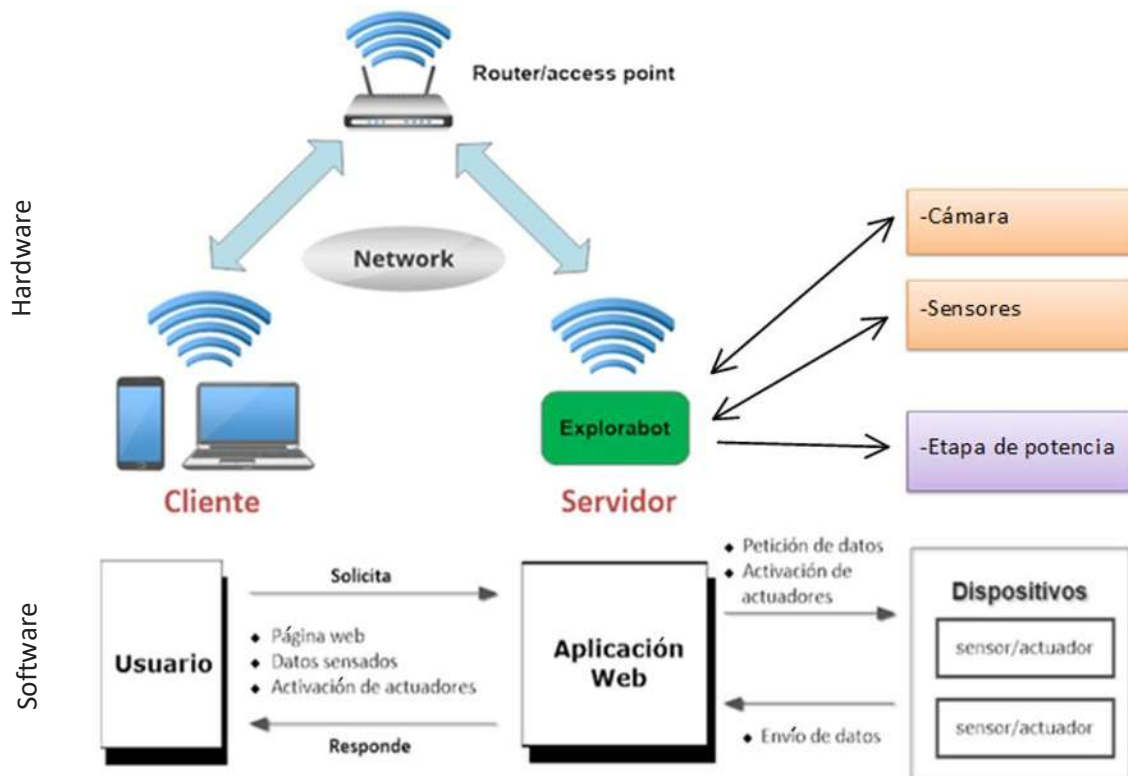


Figura 5-1 Diagrama Integrado

5.1 Pruebas de Hardware

5.1.1 Carga y descarga de batería

De acuerdo a la sección 3.4.3, en este proyecto se usa una batería recargable de plomo-ácido modelo BR-1207 con un voltaje nominal de 12 V y un voltaje en uso de 13.5 V. La cantidad de energía que puede otorgar es de 7 Amperios por 1 hora (7Ah). El periodo de operación del vehículo móvil depende de la carga de la batería y de la cantidad de trabajo que se exija; el tiempo es de 2 horas aproximadamente con uso continuo y una carga completa. El tiempo necesario para obtener una carga completa con la batería descargada es de aproximadamente 5 horas mediante el uso de un APC® Backup modelo BE550G, sin embargo, si se usa una fuente externa se puede alcanzar en un rango de 2 a 3 horas.

5.1.2 Alcance de transmisión entre el robot y el usuario

La comunicación entre el robot y el usuario se realiza mediante el estándar IEEE 802.11, mejor conocido como WIFI, por lo tanto, el alcance queda limitado a esta tecnología.

Se debe considerar que la comunicación entre el usuario final y el robot móvil se realiza a través de un router, como se aprecia en la Figura 5-1, es este caso el router es de la marca Arcadyan modelo “VRV9519KWAC23”, que cuenta con 2 antenas 5dBi.

Solo se considera la distancia máxima del robot al router, ya que se supone el usuario final estará dentro del alcance de este último.

Para la realización de esta prueba se emplearon dos distintas conexiones, una a través del chip Wifi Built-in de la Raspberry y otra con un adaptador de red de la marca Encore, modelo “N150” con una antena 5dBi (Figura 5-2), con la finalidad de comparar cuál de las interfaces tiene un mayor alcance.



Figura 5-2 Adaptador de red inalámbrica USB [92]

La primera conexión se realizó usando un teléfono inteligente Samsung modelo J700M como punto de acceso y la interfaz de red inalámbrica de la Raspberry. En este caso se alcanzó una distancia máxima de 30 metros antes de perder la conexión. Figura 5-3

La segunda se realizó usando el router Arcadyan como punto de acceso y la interfaz de red inalámbrica USB Encore (Véase Figura 5-4). A diferencia de la primera conexión, se logró una distancia máxima de 60 metros. Ambas pruebas fueron realizadas en un área libre de obstáculos.



Figura 5-3 Conexión: Smartphone – Interfaz de red Raspberry



Figura 5-4 Conexión: Router - Interfaz de red USB

5.1.3 Movimiento en exteriores

Se realizaron pruebas en distintos tipos de terreno, para evaluar la capacidad de movilidad del sistema locomotor, debido a que el objetivo del robot móvil es ingresar a áreas de difícil acceso.

5.1.3.1 Terreno con vegetación



Figura 5-5 Terreno con vegetación

En este terreno el vehículo no tiene dificultad alguna para moverse, puede girar completamente y solo presenta retrasos cuando el terreno es demasiado abultado o deforme. Figura 5-5

5.1.3.2 Terreno rocoso



Figura 5-6 Terreno rocoso

La movilidad del robot en este tipo de terreno es la más limitada, si se hace una comparación entre el terreno con vegetación o el asfalto. Debido a las rocas, las ruedas quedan atrapadas perdiendo el control del movimiento (Figura 5-6), si se quiere una mejor movilidad se necesitaría tener tracción en las cuatro ruedas o un tipo de locomoción oruga.

5.1.3.3 Plano inclinado



Figura 5-7 Plano inclinado

Otro obstáculo es el plano inclinado, en este caso se instaló con un ángulo de inclinación de 30 grados aproximadamente (Véase Figura 5-7), al hacer la prueba el robot la superó adecuadamente, comprobándose la capacidad de los motores.

5.2 Pruebas de Software

En la Figura 5-8, se muestra la página inicial al ingresar a la dirección IP y puerto del robot, en este caso se usó el puerto 3000. Dentro de ésta se presentan 2 botones, los cuales permiten iniciar sesión para el manejo del robot o simplemente ver la transmisión del video en vivo.

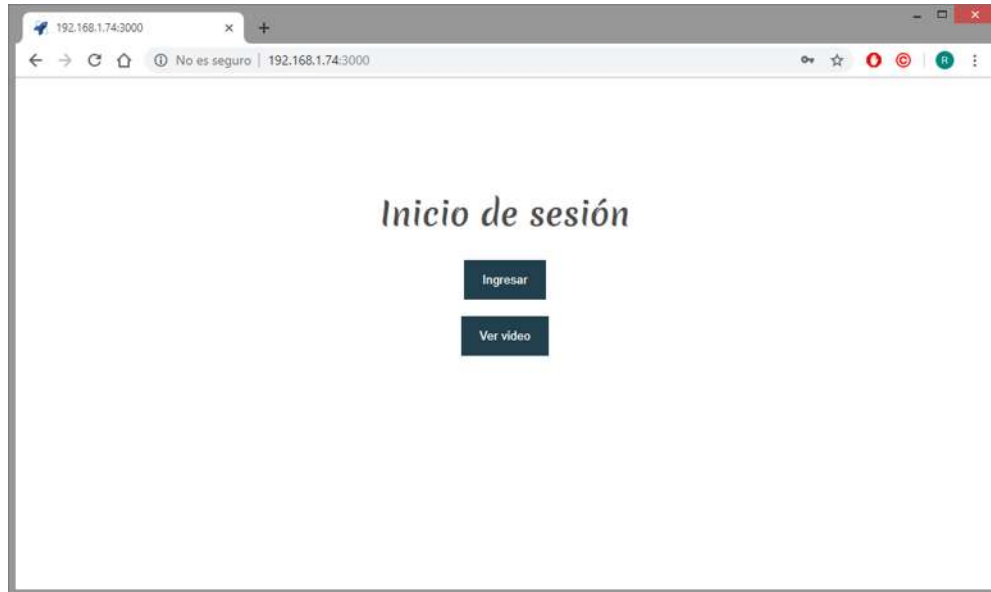


Figura 5-8 Página Inicial

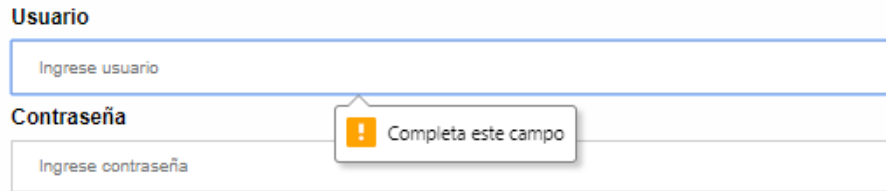
5.2.1 Inicio de sesión

Si se da clic en el botón “Ingresar” se despliega un cuadro de dialogo donde se pide el usuario y contraseña del sistema. Véase Figura 5-9.



Figura 5-9 Cuadro de dialogo "Inicio de sesión"

Si se omite el usuario o la contraseña se muestra una alerta y no permite avanzar. Figura 5-10.



Usuario

Ingrese usuario

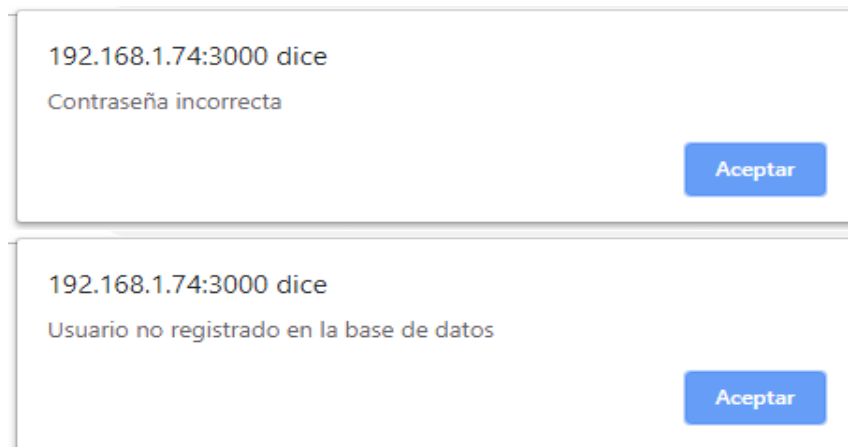
Contraseña

Ingrese contraseña

Completa este campo

Figura 5-10 Usuario o contraseña sin ingresar

Si se ingresa un usuario no registrado o la contraseña es incorrecta, el servidor responde con una alerta y se redirecciona a la página inicial como se observa en la Figura 5-11:



192.168.1.74:3000 dice
Contraseña incorrecta

Aceptar

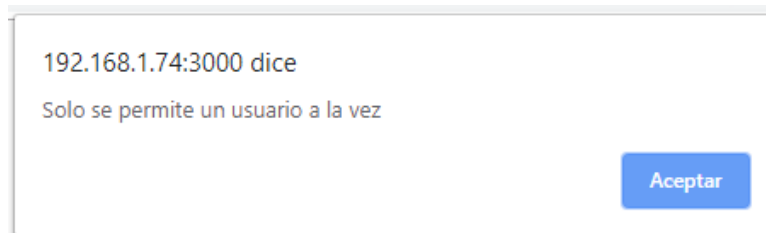
192.168.1.74:3000 dice
Usuario no registrado en la base de datos

Aceptar

Figura 5-11 Usuario y contraseña incorrecta

El inicio de sesión se implementó para evitar el manejo del robot por usuarios no autorizados y de esta forma solo existe un único usuario ingresado para su correcto funcionamiento.

Si alguien trata de ingresar al sistema por segunda ocasión, el servidor le niega el acceso (Figura 5-12) y se le redirecciona a la página inicial:

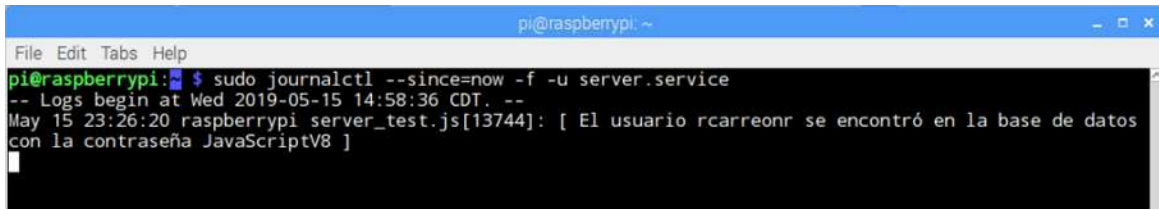


192.168.1.74:3000 dice
Solo se permite un usuario a la vez

Aceptar

Figura 5-12 Error límite máximo de usuarios

Cuando se ingresa el usuario y contraseña, el servidor compara los campos para verificar que coincida con el registrado y así permitir o no el acceso a la interfaz de usuario. Como se aprecia en la Figura 5-13, el log generado por el servicio muestra cuando se ingresa el usuario correctamente.



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ sudo journalctl --since=now -f -u server.service
-- Logs begin at Wed 2019-05-15 14:58:36 CDT. --
May 15 23:26:20 raspberrypi server_test.js[13744]: [ El usuario rcarreonr se encontró en la base de datos con la contraseña JavaScriptV8 ]

```

Figura 5-13 Terminal: Usuario y contraseña correctos

Una vez que se inició sesión exitosamente, el servidor permite el acceso a la página principal que contiene la interfaz de usuario: Figura 5-14

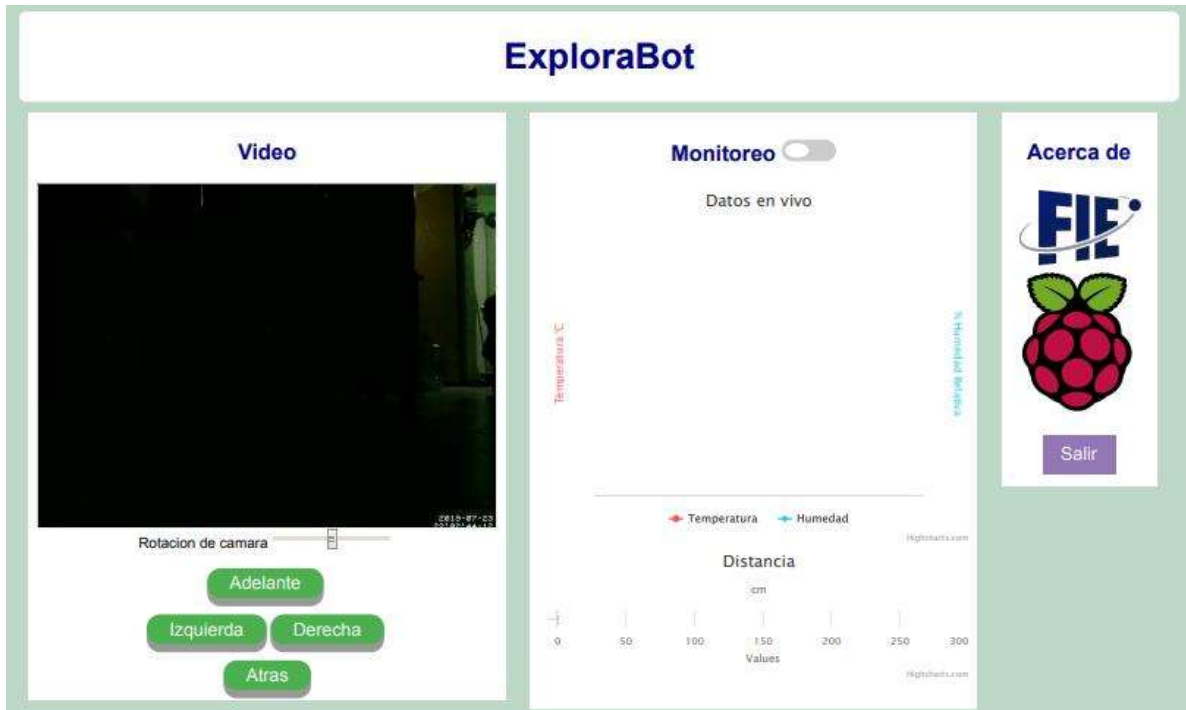


Figura 5-14 Página principal: Interfaz de usuario

5.2.2 Funcionamiento de interfaz

Como se explicó en la sección 4.1.2.2, al presionar los botones dedicados al movimiento del robot se envía una petición HTTP al servidor y este realiza la acción adecuada.

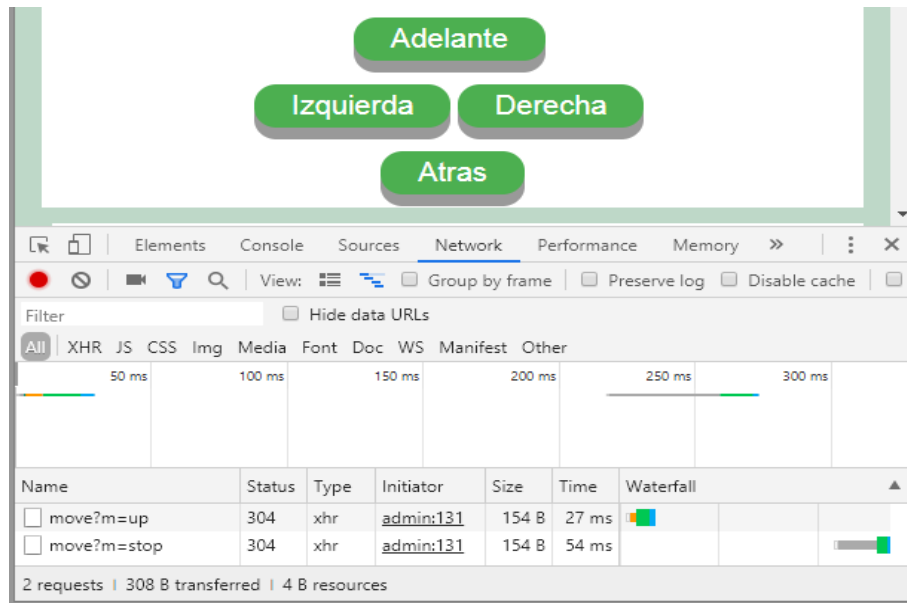


Figura 5-15 Peticiones HTTP de control

En la Figura 5-15 se muestra la petición enviada por el navegador cuando se presiona el botón “Adelante”, se nota que al soltar este botón vuelve a enviar la petición, pero con la información “stop”. Además, que el tiempo que tarda la petición en ser enviada, procesada por el servidor y en ser respondida, se maneja con magnitudes en el orden de los milisegundos.

Mediante una captura de terminal, se puede observar cómo se reciben las peticiones en el lado servidor, en este caso, las mismas que se enviaron desde el navegador. Figura 5-16

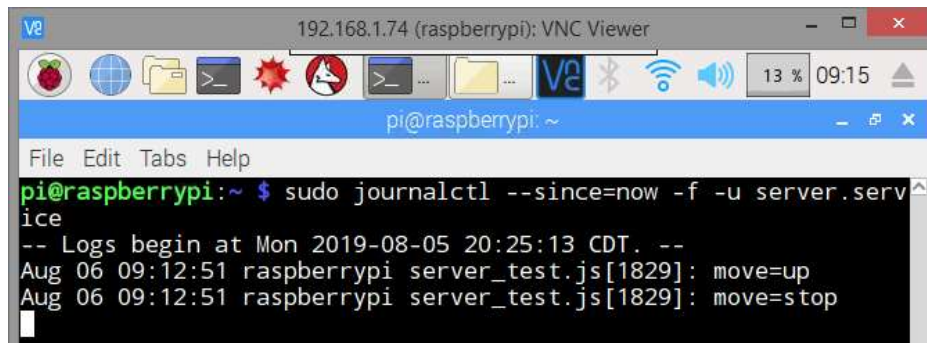


Figura 5-16 Log en terminal: peticiones HTTP

Este mecanismo es usado igualmente para los datos de los sensores y para toda comunicación entre cliente y servidor.

5.2.3 Gráfica de variables

Como se mencionó en la sección 4.1.2.3, para crear una interfaz más intuitiva e interactiva hacia el usuario se agregó una gráfica dinámica con los valores medidos de temperatura y humedad. Figura 5-17

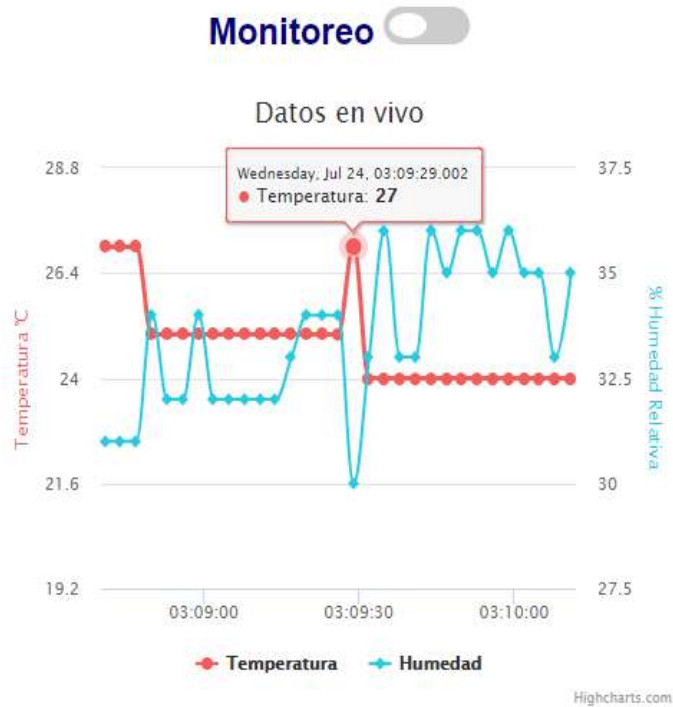


Figura 5-17 Gráfica Dinámica

Se observan dos ejes verticales con su respectiva graduación, además, se tienen diferentes colores para cada tipo de variable y cada punto en la gráfica representa una medición donde se muestra su magnitud y el instante en el que fue registrada. Al aplicar calor al sensor se comprobó que las mediciones se actualizan con el tiempo, tanto la temperatura como la humedad registraron cambios.

5.2.4 Servidor de video

Uno de los puntos a comprobar es la capacidad del servidor de video, debido a que varios usuarios pueden observar la transmisión en tiempo real, en necesario probar el número máximo de conexiones simultáneas, sin ocasionar retrasos en la transmisión o disminuir el desempeño.

Se hicieron conexiones desde teléfonos móviles usando el navegador Google Chrome®, con este tipo de dispositivos se lograron 5 conexiones estables y sin retrasos. A su vez, se agregó otra conexión desde una laptop, pero en este caso se utilizaron dos navegadores simultáneamente, Chrome® y Firefox®. En total, se realizaron 7 conexiones simultáneas con la transmisión de video en vivo en una red local y no se registraron retrasos, interrupciones o disminución de la fluidez. Véase Figura 5-18.



Figura 5-18 Prueba: servidor de video

5.2.5 Desempeño de sitio web

El navegador Google Chrome® mediante las herramientas de desarrollador ofrece una opción para aplicar una serie de pruebas o auditorías a un sitio web y de esta manera, medir aspectos de desempeño, accesibilidad, mejores prácticas, etc. Una vez realizadas las pruebas se presenta un resumen donde se muestran los diferentes aspectos medidos en un rango de 0 a 100. Además, en las áreas de menor calificación se muestran los principales errores o puntos de mejora.

En la Figura 5-19 se presentan los resultados obtenidos por el sitio web donde se implementa la interfaz de usuario. Como se observa el desempeño obtuvo una alta puntuación (Performance), así como las buenas prácticas (Best Practices). La accesibilidad tuvo un nivel intermedio, esta se define como la capacidad de que personas con problemas de vista, audición o algún problema locomotor puedan interactuar fácilmente con la página. Y la puntuación más baja se obtuvo en el aspecto de aplicación web progresiva (Progressive Web App), donde se consideran características que le permitan al sitio web aumentar su funcionalidad dependiendo el dispositivo en el que ejecuta.

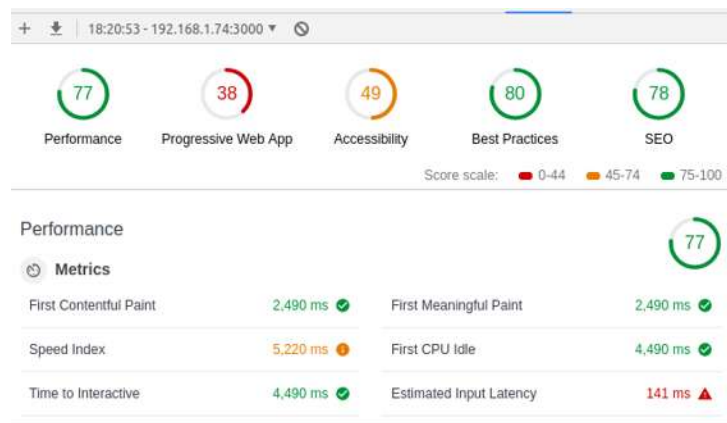


Figura 5-19 Resumen de pruebas a sitio web



Conclusiones

En este trabajo de tesis se llevó a cabo el diseño y construcción de un sistema robótico móvil para la exploración en lugares de riesgo para el ser humano. En este último capítulo se describen las conclusiones a las que se llegó tras el desarrollo y los resultados obtenidos.

Como conclusión general se puede mencionar que el desarrollar este proyecto necesitó de muchas áreas de la ingeniería electrónica e ingeniería en computación, debido a que la interfaz se implementa en la línea de programación web, el sistema de exploración está en el área de la robótica móvil, también se incluye el campo de electrónica analógica y digital y por último para integrar el proyecto se hace mediante desarrollo de software y arquitectura de redes computacionales, como lo es la arquitectura cliente-servidor. El trabajar con todas estas áreas de la ingeniería fue un reto, en especial las que se relacionan con las ciencias computacionales y que no se tenía una gran experiencia con ellas, pero a su vez representó una gran oportunidad para poner a prueba y demostrar las habilidades que se adquieren en la licenciatura. Al finalizar, se obtuvo un gran aprendizaje y satisfacción del trabajo realizado.

En la sección 1.4.2 se describen los objetivos particulares de este proyecto, a continuación, se expondrán las conclusiones relacionadas a estos objetivos y hasta qué punto se cumplieron.

6.1 Objetivos alcanzados

6.1.1 Sistema Locomotor

Se construyó un sistema locomotor diferencial simple para el desplazamiento del robot, el cual está compuesto por 2 motores que se acoplan a ruedas de goma. Este tipo de locomoción es sencilla de implementar y presenta muchas ventajas, entre ellas, un fácil control de los motores, permite girar en cualquier sentido sobre su propio eje, configuración mecánica sencilla, etc. Después de las pruebas realizadas en la sección 5.1.3 se concluyó que siendo este un prototipo la locomoción es buena, si se quiere llegar a un producto comercial se necesita que el sistema locomotor y mecánico sea más complejo según las necesidades de la aplicación.

6.1.2 Manejo de sensores

Para el monitoreo del terreno se integró una serie de sensores que ayudan al control del vehículo, permitiendo al operador tener una referencia del entorno en el que se desempeña,

específicamente se midieron las variables de temperatura ambiente, humedad y distancia de objetos. Con estos parámetros se puede orientar al operador y advertir de situaciones de riesgo, como un aumento súbito en la temperatura o una distancia crítica hacia un obstáculo.

Al trabajar en la integración de los sensores con el sistema de control principal (Raspberry) se concluyó que la actividad debe realizarse cuidadosamente y con ciertas consideraciones, si bien los sensores pueden ser analógicos o digitales, para ambos casos se debe acondicionar la señal para los niveles de voltaje adecuados del sistema de control, elegir la frecuencia adecuada con la que se muestrean las variables y además para el caso de sensores digitales se debe tener en cuenta el tipo de dato que obtendremos de esa medición y el protocolo usado para un correcto manejo de la información.

6.1.3 Visualización del terreno

Al sistema se añadió una cámara y un servidor de video para la visualización del terreno donde opera el robot, esta imagen junto con los datos recopilados por los sensores da como resultado un modelo del entorno. Como se mencionó en la sección 4.2.1, el servidor de video fue implementado mediante un software dedicado llamado “Motion”, la ventaja al seleccionar este tipo de servidor es que está programado para tener una excelente eficiencia, gran libertad de configuraciones y su integración no presenta una gran dificultad, de esta manera se ahorró una gran cantidad de tiempo y código a desarrollar para esa aplicación. Además, debe recordarse que el objetivo principal del proyecto es la integración de todas las etapas en un prototipo funcional.

No obstante, si la aplicación requiriera el manejo de las imágenes capturadas, se podría trabajar en el desarrollo de un servidor según las necesidades, en la sección 6.2 se aborda con mayor detalle este tema.

6.1.4 Interfaz de usuario

Una de los elementos de mayor importancia en este proyecto es la interfaz de usuario, el robot es controlado a distancia (tele-operado), por tanto, debe ser fácil su manejo y la visualización de parámetros. Esta interfaz se diseñó e implementó en una página web, la cual puede ser accedida desde cualquier navegador sin importar la plataforma en la que se ejecute; esta es una de las razones por la cual se decidió realizar una aplicación web.

Implementar la interfaz en una aplicación web tiene la ventaja de tener acceso a todas las tecnologías y herramientas de diseño web, como bibliotecas y frameworks. Como desventaja presenta que, al ser ejecutada en diferentes dispositivos (móviles o escritorio), puede tener algunas características incompatibles dependiendo del navegador usado y su versión. Teniendo en cuenta los pros y contras, como conclusión es preferible el uso de una aplicación web si se quiere tener un mayor alcance y accesibilidad hacia los usuarios.

6.1.5 Construcción e integración

Con la finalidad de obtener un prototipo de bajo costo se procuró que las piezas usadas para el ensamble del robot fueran de material reciclado o económico, además el diseño es simple y se puede replicar de una manera sencilla. Una vez finalizado el ensamble mecánico se montó la etapa

de potencia, así como la de control y se procedió a realizar pruebas de movilidad. Acto continuo se fijaron los sensores y la cámara al chasis y se probaron en conjunto.

Se lograron integrar todas las etapas del sistema robótico para un correcto funcionamiento, así como el adecuado manejo de datos que fluye del usuario final al sistema y viceversa. Y a pesar de que el software no se ejecuta en un sistema operativo en tiempo real (RTOS) se concluye que, para la aplicación, es más que suficiente el tiempo de respuesta que se obtiene. Esto se logra gracias a la arquitectura cliente-servidor, el protocolo ligero HTTP y un servidor orientado a eventos no bloqueante.

6.2 Trabajos futuros

Como continuación de este trabajo de tesis, existen diversas líneas de investigación y desarrollo que quedan abiertas y en las que es posible continuar trabajando. A continuación, se presentan algunos trabajos futuros que pueden desarrollarse como resultado de esta investigación o que, por exceder el alcance de esta tesis, no han podido ser tratados con la suficiente profundidad.

Como se hace mención en la sección 6.1.3, una mejora que se puede implementar es procesar las imágenes que están siendo capturadas en tiempo real, mediante el procesamiento digital de imágenes, se pueden aplicar ciertos filtros los cuales serían de gran ayuda para detectar y evaluar daños en estructuras. Otra aplicación en esta línea de desarrollo podría ser el detectar rostros mediante estas mismas herramientas, una vez implementada esta característica de reconocimiento facilitaría la tarea de búsqueda y rescate de personas. Para alcanzar este objetivo se puede hacer uso del lenguaje de programación “Python” en conjunto con la biblioteca especializada para el procesamiento de imágenes llamada “OpenCV”. Python es un lenguaje de alto nivel de propósito general que se ha popularizado por su uso en el desarrollo web y en el ámbito científico, como consecuencia de que puede ser extendido (vinculado) a C/C++. Por otro lado, la biblioteca OpenCV ofrece una gran variedad de funciones especializadas para trabajar con matrices (imágenes), está escrita en C/C++, gracias a esto el procesamiento se realiza con una gran eficiencia computacional y rapidez.

Otra línea de desarrollo en potencia es el sistema locomotor, donde existe gran área de oportunidad, esto como consecuencia directa de que el presente proyecto es un prototipo y se implementó un sistema mecánico simple que cumplió con las funciones de movilidad fijadas. No obstante, para que el prototipo pueda desempeñarse en todos los terrenos para los que fue pensado, es necesario que cuente con un sistema más robusto que le permita evadir y superar obstáculos propios del entorno, por ejemplo: escaleras, desniveles abruptos, arena, etc. Una solución eficaz es reemplazar el sistema locomotor por uno tipo oruga, de esta forma se obtiene una mayor superficie de tracción y por lo tanto un mejor agarre.

Para finalizar, otra característica que mejoraría considerablemente la funcionalidad del prototipo es el añadir un brazo manipulador el cual cuente con alguna herramienta dedicada según la aplicación. De esta forma el brazo podría ayudar en la eliminación de obstáculos, en la recuperación de objetos, incluso en las propias líneas de búsqueda y rescate de personas, las aplicaciones aumentarían en gran proporción.

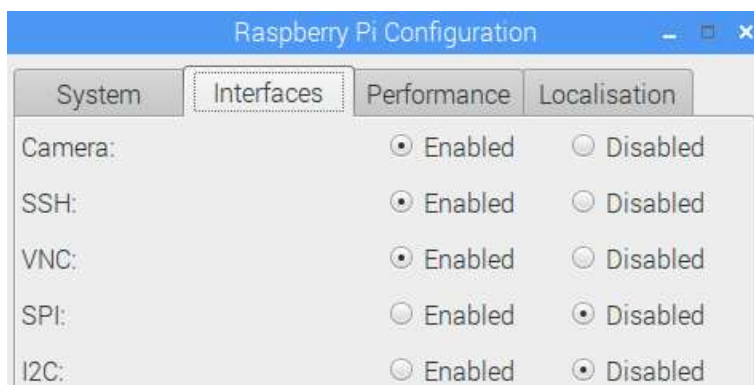
Apéndices

Apéndice 1: Instalación de servidor de video Motion

Para poder hacer uso de la transmisión de video en vivo, se tiene que efectuar la instalación del software “Motion” y su correcta configuración. A continuación, se presentan los pasos a seguir para llevar a cabo esta etapa, en particular este procedimiento fue probado en Raspberry Pi® 3 B con el sistema operativo Raspbian GNU/Linux 8.0 (jessie).

1. Configurar la interfaz en Raspberry ®

Como primer paso se necesita habilitar la interfaz de la cámara en los ajustes de la Raspberry®, desde el escritorio se abre el menú principal y se da clic en “Preferences -> Raspberry Pi Configuration -> Interfaces”, después solo se tiene que cambiar el estado de la cámara a “Enabled” como se observa en la siguiente imagen.



2. Instalar “Motion”

Es necesario verificar que la Raspberry® este actualizada, mediante los siguientes comandos ejecutados en una terminal.

```
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get upgrade
```

A continuación, se necesita ejecutar los siguientes comandos que instalaran “Motion” y activaran el controlador oficial.

```
pi@raspberrypi:~ $ sudo apt-get install motion && sudo modprobe bcm2835-v4l2
```

Para terminar con este proceso, también se necesita configurar la Raspberry® para que cada vez que se reinicie el sistema se active el controlador. Para esto se modifica el archivo “modules” en el directorio “/etc” con el comando “`sudo nano /etc/modules`” y se verifica que al final de este se encuentre el nombre del controlador, si no es así, se debe agregar y guardar como se muestra en la siguiente imagen:

```

GNU nano 2.2.6      File: /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be$
# at boot time, one per line. Lines beginning with "#" are igno$
i2c-dev
bcm2835-v4l2 ←
[ Read 7 lines ]
^G Get Hel ^O WriteOu ^R Read Fi ^Y Prev Pa ^K Cut Tex ^C Cur Pos
^X Exit    ^J Justify ^W Where I ^V Next Pa ^U UnCut T ^T To Spell

```

Debido a que se requiere que el servidor de video este ejecutándose continuamente se necesita configurar como un proceso en segundo plano (daemon) y que se inicie automáticamente. Para eso ejecutamos en terminal “`sudo nano /etc/default/motion`” y se abrirá el archivo como se observa en la imagen inferior, en este modificamos la línea “`start_motion_daemon`” y le asignamos el valor “`yes`”, con esto terminaríamos la instalación del servidor de video.

```

GNU nano 2.2.6      File: /etc/default/motion
# set to 'yes' to enable the motion daemon
start_motion_daemon=yes

```

3. Configurar Motion

Primeramente, se hace un respaldo del archivo de configuración por si se comete algún error poder regresar a la configuración inicial, esto se logra mediante el siguiente comando:

```
$ > sudo cp /etc/motion/motion.conf /etc/motion/motion.conf.bak
```

Después se necesita modificar el archivo de configuración, para abrirlo se necesita ejecutar el siguiente comando:

```
$ > sudo nano /etc/motion/motion.conf
```

En este archivo se encuentra organizada todas las configuraciones disponibles y una pequeña descripción de cada una, para este caso en particular se necesita modificar las siguientes características:

```

daemon on
stream localhost off
output_pictures off
ffmpeg_output_movies off

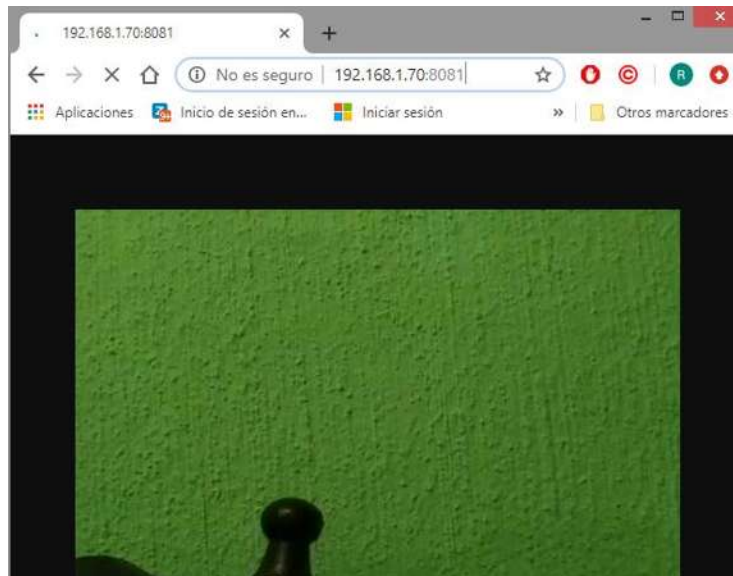
```

```
framerate 15  
width 512  
height 384  
stream_port 8081
```

Básicamente estas configuraciones permiten realizar el streaming, además de definir los parámetros del video, como ancho, alto y los cuadros por segundo. Una vez realizados los cambios, se guarda el archivo y se reinicia el servicio para que se tomen en cuenta, esto se realiza con el siguiente comando:

```
$ > sudo service motion restart
```

Y finalmente se puede observar el video-streaming desde un navegador, simplemente definiendo la dirección IP de la Raspberry® y el puerto especificado, en este caso es el 8081.



Si existieran dudas al respecto, un tutorial más completo puede ser encontrado en el siguiente link: <https://hackernoon.com/spy-your-pet-with-a-raspberry-pi-camera-server-e71bb74f79ea>

Apéndice 2: Instalación y ejecución de aplicación en Node.js

En este apartado se muestran los pasos a seguir para poder ejecutar la aplicación web, desde la instalación del software Node.js hasta la ejecución del servicio que maneja todas las peticiones, procesos y datos del sistema de exploración. Similarmente al Apéndice 1: Instalación de servidor de video Motion, este procedimiento fue probado en Raspberry Pi® 3 B con el sistema operativo Raspbian GNU/Linux 8.0 (jessie).

1. Instalación de Node.js

El primer paso a realizar es abrir una terminal y actualizar la Raspberry® como se menciona en la sección Instalar “Motion”. Una vez actualizada se necesita habilitar el repositorio donde están las últimas versiones de Node.js, para esto se ejecuta el siguiente comando en la terminal:

```
$ > curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -
```

Una vez que el repositorio está habilitado, se instala Node.js y NPM con el siguiente comando:

```
$ > sudo apt-get install -y nodejs
```

Para verificar que se ha instalado correctamente y su versión, se ejecuta:

```
$ > node -v
```

Si no hubo error con alguno de estos pasos se habrá realizado la correcta instalación y se podrá empezar a trabajar con Node.js

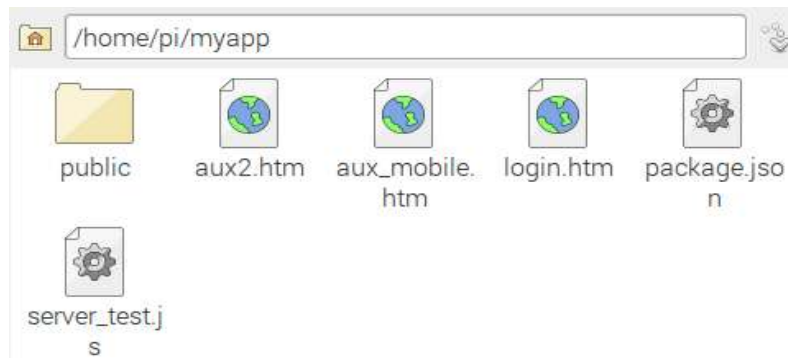
2. Instalación de la aplicación

Para que la aplicación se pueda ejecutar correctamente, es necesario tener los archivos fuente y estáticos que la conforman. Estos archivos se encuentran en el CD-ROM en “/código_fuente/”, a continuación, se explica cómo se organizan e instalan.

En primer lugar, se necesita crear un nuevo directorio donde se alojarán los archivos de la aplicación, en la terminal se ejecuta el siguiente comando, donde “myapp” es el nombre del directorio:

```
$ > mkdir myapp && cd myapp
```

Una vez dentro del directorio, se copian los archivos de la aplicación para que queden organizados de la siguiente manera:



Los archivos *aux2.htm* y *aux_mobile.htm* son las páginas principales donde se desarrolló la interfaz de usuario, uno es la versión de escritorio y el otro la versión móvil optimizada para dispositivos touchscreen. El archivo *login.htm* es la página que se muestra cuando nos conectamos al sistema de exploración y es donde se inicia sesión para tener el control del robot. Se puede decir que los archivos antes mencionados conforman la etapa de Front-end en conjunto con la carpeta */public* donde se guardan los archivos estáticos que ayudan a la vista de la aplicación (imágenes, archivos de estilo, bibliotecas).

La etapa de Back-end está compuesta por los archivos *server_test.js* y *package.json*, en el primero se definen todas las rutinas y funciones que conforman al servidor web y al sistema robótico, es el archivo principal y es el servicio que se ejecuta sobre Node.js. El segundo es el archivo que define todas las dependencias que necesita nuestro servicio para poder ejecutarse, contiene los datos de nuestro proyecto y los módulos necesarios para poder ejecutarlo correctamente, es de suma importancia ya que sin este archivo no se podría instalar la aplicación.

El siguiente paso es instalar las dependencias definidas en el archivo *package.json*, ejecutando el siguiente comando:

```
$ > npm install
```

Una vez terminado el proceso, se generará la carpeta *node_modules* donde se instalaron los módulos y el archivo *package-lock.json* que prácticamente es un manifiesto extendido del proyecto, como se observa en la siguiente figura, con esto se finalizaría la instalación de la aplicación.



Ejecución de la aplicación

Para iniciar la aplicación se necesita ejecutar el siguiente comando:

```
$ > sudo node server_test.js
```

Se antepone la palabra sudo debido a que se necesita permiso de administrador para tener acceso a los pines de entrada/salida y a los temporizadores propios de la Raspberry®, además es necesario localizarse en el directorio donde está instalada la aplicación para poder ejecutarla.

Ahora el servicio está corriendo y se puede acceder a ella desde el navegador ingresando la dirección IP de la Raspberry® en el puerto 3000.

Apéndice 3: Diagrama Electrónico

En la figura inferior se muestra el diagrama de conexiones completo del proyecto, sin embargo, se excluyó la etapa de opto-acoplamiento para mejorar la claridad de las conexiones entre los dispositivos y los pines de entrada y salida.

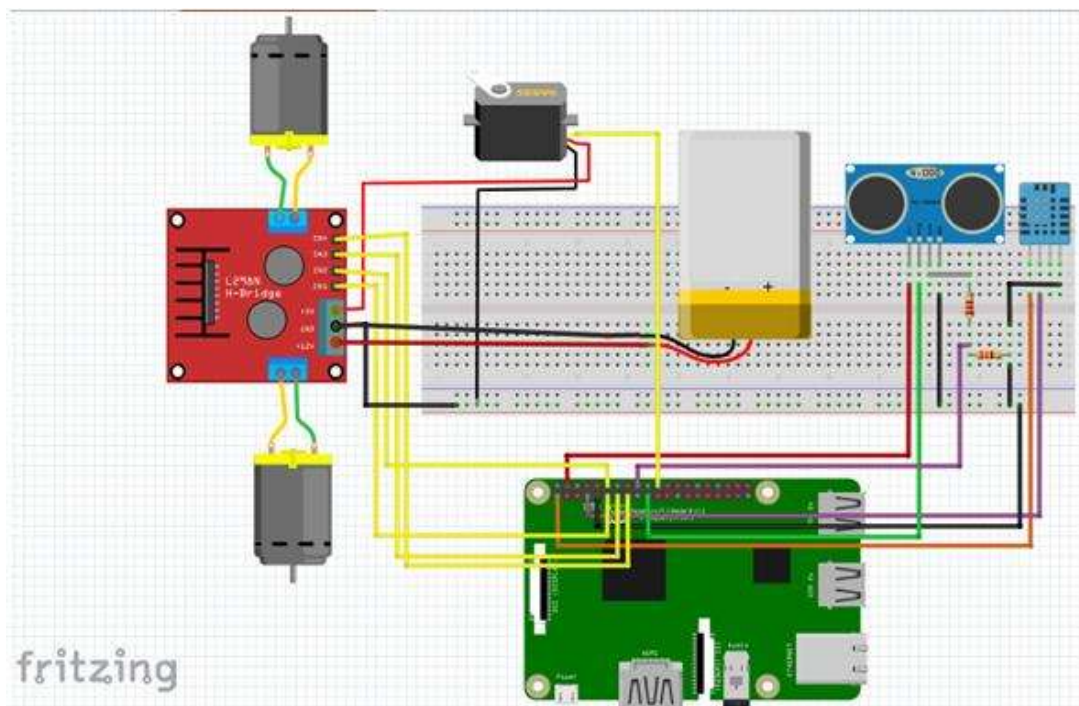
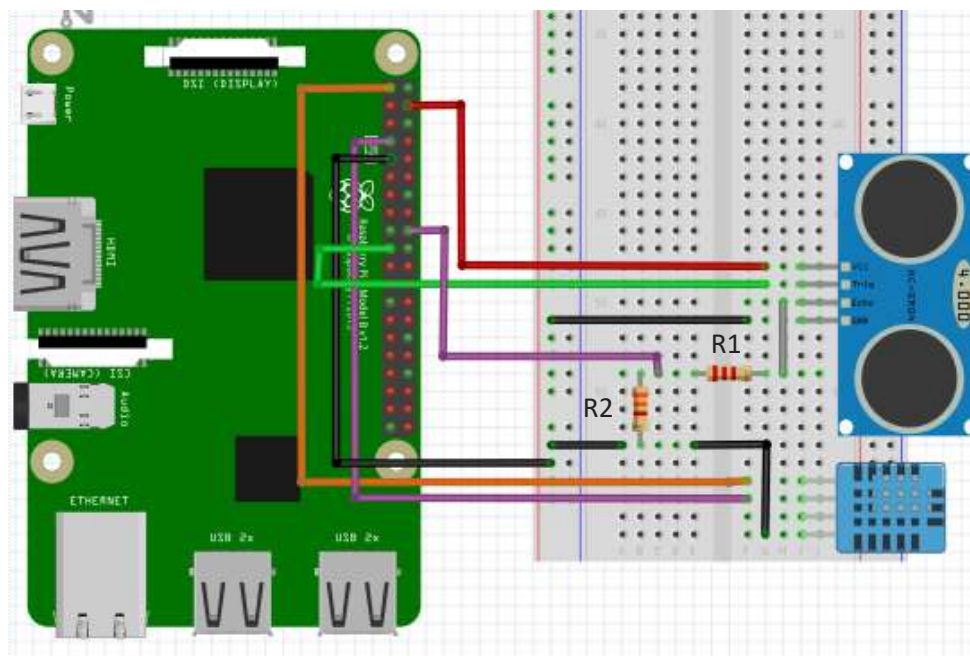


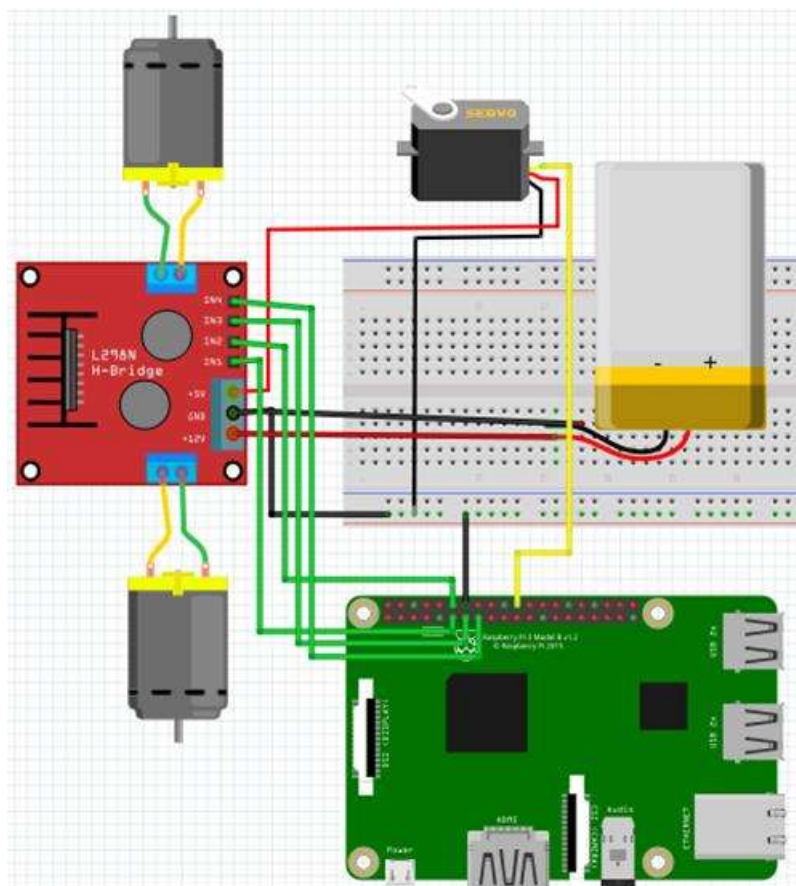
Diagrama de sensores:



En la imagen anterior se puede apreciar el diagrama de conexiones específico para el conjunto de sensores, cabe señalar que el sensor ultrasónico trabaja con un nivel de voltaje de 5V y la Raspberry® acepta solo el nivel de 3.3V, por esta razón se necesita un divisor de voltaje y de esta forma adecuar la señal eléctrica, un par de resistencias que pueden lograr este resultado son: $R1=1K\Omega$ y $R2=2K\Omega$ o cualquier otra combinación siempre y cuando $R2$ sea aproximadamente el doble de $R1$. A continuación se muestra la relación entre los pines de la Raspberry® y de los dispositivos para reducir los errores a la hora de cablear.

Raspberry®	DHT11
3v3	VDD
GPI04	DATA
GND	GND
	HC-SR04
5V	VCC
GPI010	TRIG
GPI024	ECHO (DIVISOR DE VOLTAJE)
GND	GND

Diagrama de actuadores:



La imagen superior muestra el diagrama específico para la etapa de actuadores, se omite la etapa de opto-acopladores para un diagrama más simple y claro.

En la siguiente tabla se muestran la relación entre pines entre las Raspberry® y los actuadores:

Raspberry®	L298n
GPIO17	IN1
GPIO18	IN2
GPIO27	IN3
GPIO22	IN4
GND	GND
	Servomotor
GPIO25	PULSE
GND	GND
5V (L298n-ver diagrama)	VCC

Cabe mencionar que además de los diagramas, se hace una recomendación a que se analice la hoja de datos de cada dispositivo usado antes de realizar la integración de los componentes y de esta manera tener mejor panorama del hardware implementado.

Apéndice 4: Detalles de construcción

A continuación, se muestra un listado de los componentes y materiales utilizados en el desarrollo y construcción del sistema robótico junto con un aproximado del costo.

Lista de componentes electrónicos:

- ◆ Raspberry Pi 3 Modelo B (microSD) - Precio: \$1000 pesos.
- ◆ Raspberry Pi Camera Module - Precio: \$1200 pesos.
- ◆ Sensor de temperatura y humedad DHT11 - Precio: \$60 pesos.
- ◆ Sensor de distancia HC-SR04 - Precio: \$50 pesos.
- ◆ Optoacoplador PC817 (5 piezas) - Precio: \$50 pesos.
- ◆ Módulo Dual Driver L298N - Precio: \$70 pesos.
- ◆ Motores de CD Maxon 278072 (2 piezas) - Precio: \$3000 pesos.
- ◆ Servomotor SG90 - Precio: \$70 pesos.
- ◆ Batería Steren BR-1207 - Precio: \$500 pesos.

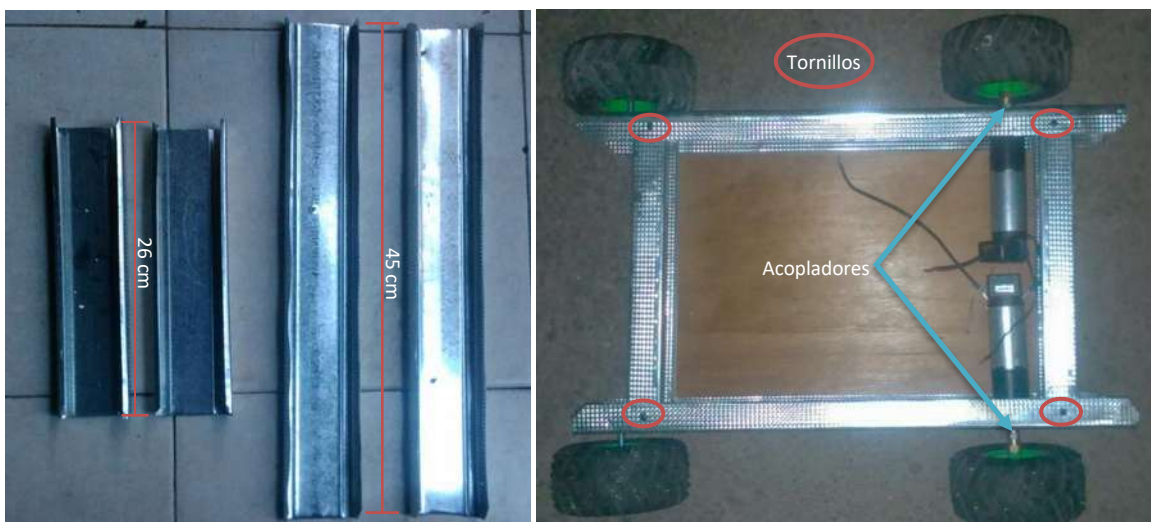
Lista de materiales utilizados en el chasis:

- ◆ Perfil de metal galvanizado (Tablaroca) - Precio: N/D reciclado.
- ◆ Hoja de triplay o plástico - Precio: N/D reciclado.
- ◆ Pieza de plástico pequeña (soporte cámara) - Precio: N/D reciclado.
- ◆ Acopladores de bronce (2 piezas) - Precio: \$150 pesos.
- ◆ Tornillos - Precio: \$20 pesos.

El costo aproximado del vehículo es de \$6370 pesos.

Métricas del chasis

Sección inferior:



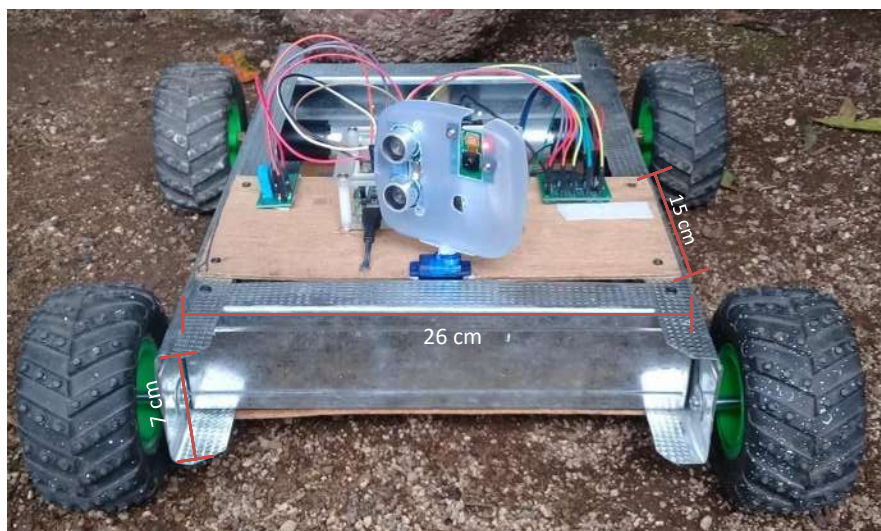
Se debe atornillar el perfil galvanizado de manera que se forme un rectángulo, como se muestra en la imagen superior, después se puede fijar la hoja de triplay en la parte inferior, sirviendo de soporte para los componentes como la batería y la etapa de potencia.

El siguiente paso es fijar los motores de corriente directa en la parte trasera del chasis, teniendo especial cuidado en la simetría y lo más cercano posible al plano inferior.

Por último, se debe instalar las llantas de goma con los acopladores al eje del motor. Para las llantas delanteras se pueden usar tornillos en lugar del eje del motor.



Sección superior:



En la sección superior también es necesario instalar una lámina de triplay que sirve de base para la Raspberry, la cámara y los sensores. Una vez ensamblados todos los componentes, tanto mecánicos como electrónicos, el vehículo alcanza un peso de 3.7 Kg. El resultado final se muestra en la imagen de arriba.

Bibliografía

- [1] «Hacia el espacio,» marzo 2018. [En línea]. Available: <http://haciaelespacio.wixsite.com/haciaelespacio007/robots-espaciales>. [Último acceso: 6 mayo 2018].
- [2] «www.muyinteresante.es,» 2018. [En línea]. Available: <https://www.muyinteresante.es/tag/robots>. [Último acceso: 12 mayo 2018].
- [3] 2.-2. types, «es.wikipedia.org,» 2018. [En línea]. Available: <https://es.wikipedia.org/wiki/Robot>. [Último acceso: 12 mayo 2018].
- [4] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/Rover>. [Último acceso: 13 mayo 2018].
- [5] «sites.google.com,» [En línea]. Available: <https://sites.google.com/a/unitecnica.net/dmhenao/articulos/zoomorficos>. [Último acceso: 13 mayo 2018].
- [6] «robotroxs.blogspot.mx,» [En línea]. Available: <http://robotroxs.blogspot.mx/2015/07/robots-poliarticulados.html>. [Último acceso: 13 mayo 2018].
- [7] «HISTORIA DEL ARTE DE LA ROBOTICA: MOVILES O VEHICULOS ROBOT,» 2009. [En línea]. Available: <http://robotik-jjlg.blogspot.mx/2009/04/moviles-o-vehiculos-robot.html>. [Último acceso: 6 mayo 2018].
- [8] Monografias.com, «www.monografias.com,» [En línea]. Available: <http://www.monografias.com/trabajos94/definicion-y-antecedentes-robots/definicion-y-antecedentes-robots.shtml>. [Último acceso: 12 mayo 2018].
- [9] «masterastronomia.blogspot.mx,» marzo 2018. [En línea]. Available: <http://masterastronomia.blogspot.mx/2014/03/astromoviles-vehiculos-de-exploracion.html>. [Último acceso: 14 mayo 2018].
- [10] «www.dailytrend.mx,» 2015. [En línea]. Available: <http://www.dailytrend.mx/tecnologia/robots-rescatistas-unam#pageview-1>. [Último acceso: 14 mayo 2018].
- [11] «www.youtube.com,» 12 marzo 2017. [En línea]. Available: https://www.youtube.com/watch?v=574bVz7Z_Qc. [Último acceso: 14 mayo 2018].
- [12] «store.linksprite.com,» [En línea]. Available: <http://store.linksprite.com/rover-wifi-remote-control-robot-with-wifi-video-camera/>. [Último acceso: 14 mayo 2018].

-
- [13] C. Robot, «www.robotshop.com,» [En línea]. Available: <https://www.robotshop.com/en/superdroid-hd2-robot-wifi-control-camera.html>. [Último acceso: 14 mayo 2018].
- [14] «Sismología de México,» 2017. [En línea]. Available: <https://www.sgm.gob.mx/Web/MuseoVirtual/Riesgos-geologicos/Sismologia-de-Mexico.html>. [Último acceso: 6 mayo 2018].
- [15] «cienciamasciencia.blogspot.com,» [En línea]. Available: <http://cienciamasciencia.blogspot.com/2010/05/el-cinturon-de-fuego-o-circumpacifico.html>. [Último acceso: 16 julio 2018].
- [16] «www.aadeca.org,» [En línea]. Available: http://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.pdf. [Último acceso: 21 mayo 2018].
- [17] «www.researchgate.net,» 13 enero 2015. [En línea]. Available: https://www.researchgate.net/publication/267798579_ROBOTS_MOVILES_TEORIA_APLICACIONES_Y_EXPERIENCIAS. [Último acceso: 25 mayo 2018].
- [18] J. Andruskevich, «www.pinterest.cl,» [En línea]. Available: <https://www.pinterest.cl/pin/401946335477461701/>. [Último acceso: 22 mayo 2018].
- [19] «www.uma.es,» [En línea]. Available: <https://www.uma.es/robotics-and-mechatronics/info/78317/proyectos/>. [Último acceso: 22 mayo 2018].
- [20] «karinamedina77.blogspot.mx,» marzo 2018. [En línea]. Available: <http://karinamedina77.blogspot.mx/2012/06/robot-subacuatico.html>. [Último acceso: 25 mayo 2018].
- [21] «nexciencia.exactas.uba.ar,» 30 septiembre 2016. [En línea]. Available: <http://nexciencia.exactas.uba.ar/drones-robots-aereos-robotica-cuadricopteros-pablo-de-cristoforis-matias-nitsche-ricardo-carelli-diego-pereyra-medivac>. [Último acceso: 25 mayo 2018].
- [22] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/Sensor>. [Último acceso: 25 mayo 2018].
- [23] «robots-argentina.com.ar,» [En línea]. Available: http://robots-argentina.com.ar/Sensores_general.htm. [Último acceso: 25 mayo 2018].
- [24] «webpersonal.uma.es,» [En línea]. Available: <http://webpersonal.uma.es/~VFMM/PDF/cap2.pdf>. [Último acceso: 25 mayo 2018].
- [25] «oberobot.wordpress.com,» [En línea]. Available: <https://oberobot.wordpress.com/2012/04/23/robots-moviles/>. [Último acceso: 21 mayo 2018].

- 2018].
- [26] «[www.proandroid.com,](https://www.proandroid.com/)» [En línea]. Available: <https://www.proandroid.com/que-es-cable-otg-moviles-compatibles/>. [Último acceso: 28 mayo 2018].
 - [27] D. Krambeck, «[www.allaboutcircuits.com,](https://www.allaboutcircuits.com/)» 18 febrero 2018. [En línea]. Available: <https://www.allaboutcircuits.com/news/top-three-FPGA-development-boards-new-designers/>. [Último acceso: 28 mayo 2018].
 - [28] «[electronilab.co,](https://electronilab.co/)» [En línea]. Available: <https://electronilab.co/tienda/ti-stellaris-lm4f120-launchpad/>. [Último acceso: 28 mayo 2018].
 - [29] «[www.raspberrypi.org,](https://www.raspberrypi.org/)» febrero 2015. [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Último acceso: 28 mayo 2018].
 - [30] «[rua.ua.es,](https://rua.ua.es/)» [En línea]. Available: <https://rua.ua.es/dspace/bitstream/10045/13610/1/sistemas%20de%20control%20para%20robots.pdf>. [Último acceso: 25 mayo 2018].
 - [31] «[es.wikipedia.org,](https://es.wikipedia.org/)» [En línea]. Available: https://es.wikipedia.org/wiki/Comunicaci%C3%B3n_inal%C3%A1brica. [Último acceso: 28 mayo 2018].
 - [32] R. F., «[es.wikipedia.org,](https://es.wikipedia.org/)» [En línea]. Available: <https://es.wikipedia.org/wiki/Radiocomunicaci%C3%B3n>. [Último acceso: 28 mayo 2018].
 - [33] «[es.wikipedia.org,](https://es.wikipedia.org/)» [En línea]. Available: <https://es.wikipedia.org/wiki/Radiofrecuencia>. [Último acceso: 28 mayo 2018].
 - [34] «[www.jcee.upc.edu,](http://www.jcee.upc.edu/)» [En línea]. Available: http://www.jcee.upc.edu/JCEE2004/pdf/EstadoActual_WirelessRF_SILICA.pdf. [Último acceso: 28 mayo 2018].
 - [35] «[es.wikipedia.org,](https://es.wikipedia.org/)» [En línea]. Available: <https://es.wikipedia.org/wiki/Wifi>. [Último acceso: 28 mayo 2018].
 - [36] N. Carrodegas, «[norfipc.com,](https://norfipc.com/)» [En línea]. Available: <https://norfipc.com/redes/tipos-redes-estandares-wi-fi-diferencias.php>. [Último acceso: 28 mayo 2018].
 - [37] «[es.ccm.net,](https://es.ccm.net/)» [En línea]. Available: <https://es.ccm.net/contents/791-como-implementar-una-red-wifi-estandar-802-11>. [Último acceso: 28 mayo 2018].
 - [38] «[www.ecured.cu,](https://www.ecured.cu/)» [En línea]. Available: https://www.ecured.cu/Red_de_computadoras. [Último acceso: 3 junio 2018].
 - [39] «[es.wikipedia.org,](https://es.wikipedia.org/)» [En línea]. Available: https://es.wikipedia.org/wiki/Modelo_OSI. [Último

acceso: 4 junio 2018].

- [40] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/Cliente-servidor>. [Último acceso: 29 junio 2018].
- [41] «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto. [Último acceso: 15 julio 2018].
- [42] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/Servidor>. [Último acceso: 3 junio 2018].
- [43] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/Cliente-servidor>. [Último acceso: 3 junio 2018].
- [44] «www.ecured.cu,» [En línea]. Available: https://www.ecured.cu/Navegador_web. [Último acceso: 4 junio 2018].
- [45] «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/P%C3%A1gina_web. [Último acceso: 4 junio 2018].
- [46] «www.ecured.cu,» [En línea]. Available: <https://www.ecured.cu/HTML>. [Último acceso: 4 junio 2018].
- [47] «www.ecured.cu,» [En línea]. Available: <https://www.ecured.cu/CSS>. [Último acceso: 4 junio 2018].
- [48] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>. [Último acceso: 4 junio 2018].
- [49] Desarrolloweb.com, «desarrolloweb.com,» [En línea]. Available: <https://desarrolloweb.com/javascript/>. [Último acceso: 4 junio 2018].
- [50] F. d. Node.js, «nodejs.org,» [En línea]. Available: <https://nodejs.org/es/>. [Último acceso: 4 junio 2018].
- [51] «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Raspberry_Pi. [Último acceso: 7 junio 2018].
- [52] «www.raspberrypi.org,» [En línea]. Available: <https://www.raspberrypi.org/products/>. [Último acceso: 7 junio 2018].
- [53] «commons.wikimedia.org,» julio 2016. [En línea]. Available: https://commons.wikimedia.org/wiki/File:RaspberryPi_3B.svg.
- [54] «www.raspberrypi.org,» [En línea]. Available: <https://www.raspberrypi.org/documentation/usage/gpio/README.md>. [Último acceso: 7

junio 2018].

- [55] «cdn-learn.adafruit.com,» [En línea]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf>. [Último acceso: 8 junio 2018].
- [56] «programarfacil.com,» [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>. [Último acceso: 8 junio 2018].
- [57] J. Ruben, «www.geekfactory.mx,» [En línea]. Available: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/dht11-con-arduino/>. [Último acceso: 8 junio 2018].
- [58] E. Themes, «www.geekfactory.mx,» [En línea]. Available: <https://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-de-distancia-ultrasonico/>. [Último acceso: 10 junio 2018].
- [59] «biblioteca.geekfactory.mx,» [En línea]. Available: http://biblioteca.geekfactory.mx/Sensor_HC-SR04/DS_HC-SR04.pdf. [Último acceso: 10 junio 2018].
- [60] «upcommons.upc.edu,» [En línea]. Available: <https://upcommons.upc.edu/bitstream/handle/2099.1/9440/Proyecto-final.pdf?sequence=1&isAllowed=y>. [Último acceso: 10 junio 2018].
- [61] «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Sensor_de_imagen. [Último acceso: 10 junio 2018].
- [62] «www.raspberrypi.org,» [En línea]. Available: <https://www.raspberrypi.org/documentation/hardware/camera/README.md>. [Último acceso: 10 junio 2018].
- [63] «cdtecnologia.net,» [En línea]. Available: <http://cdtecnologia.net/raspberry-pi/39-camara-de-video-para-raspberry-pi.html>. [Último acceso: 15 junio 2018].
- [64] «www.ingmecafenix.com,» [En línea]. Available: <http://www.ingmecafenix.com/electricidad-industrial/motor-corriente-continua/>. [Último acceso: 15 junio 2018].
- [65] «www.ebay.com,» [En línea]. Available: <https://www.ebay.com/itm/Large-Maxon-Motor-w-Gearhead-HEDS-Encoder-32mm-D-x-124mm-L-25-RPM-6-VDC-/282505105182>. [Último acceso: 16 junio 2018].
- [66] «e-ducative.catedu.es,» [En línea]. Available: http://e-ducative.catedu.es/44700165/aula/archivos/repositorio/4750/4933/html/45_arranque_de_los_motores_de_cc.html. [Último acceso: 22 junio 2018].

- [67] «panamahitek.com,» [En línea]. Available: <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>. [Último acceso: 14 julio 2018].
- [68] «upcommons.upc.edu,» [En línea]. Available: <https://upcommons.upc.edu/bitstream/handle/2117/97322/PFC%20David%20Ortiz%20Mart%C3%ADnez.pdf?sequence=1&isAllowed=y>. [Último acceso: 16 junio 2018].
- [69] «akizukidenshi.com,» [En línea]. Available: <http://akizukidenshi.com/download/ds/towerpro/SG90.pdf>. [Último acceso: 14 julio 2018].
- [70] «electronilab.co,» [En línea]. Available: <https://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>. [Último acceso: 17 junio 2018].
- [71] «hetpro-store.com,» [En línea]. Available: <https://hetpro-store.com/TUTORIALES/optoacoplador/>. [Último acceso: 14 julio 2018].
- [72] «www.sharp-world.com,» [En línea]. Available: http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/pc817xnnsz_e.pdf. [Último acceso: 14 julio 2018].
- [73] «www.upokoron.com,» [En línea]. Available: <https://www.upokoron.com/product/optocoupler-pc817/>. [Último acceso: 14 julio 2018].
- [74] «www.amazon.in,» [En línea]. Available: <https://www.amazon.in/VEEROBOT-Robot-Chassis-Sensor-Holder/dp/B01ANE9DXW>. [Último acceso: 16 junio 2018].
- [75] «hobbyking.com,» [En línea]. Available: https://hobbyking.com/es_es/aluminum-4wd-robot-chassis-black-kit.html?__store=es_es. [Último acceso: 16 junio 2018].
- [76] «glowroad.com,» [En línea]. Available: <https://glowroad.com/generic-2pcs-4mm-brass-hex-shaft-coupling-coupler-rc-gear-motor-connector-robot-diy/az/B072L1MLNZ>. [Último acceso: 22 junio 2018].
- [77] «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Bater%C3%ADa_el%C3%A9ctrica. [Último acceso: 05 Agosto 2019].
- [78] «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Front-end_y_back-end. [Último acceso: 29 junio 2018].
- [79] «www.ecured.cu,» [En línea]. Available: https://www.ecured.cu/Interfaz_de_usuario. [Último acceso: 29 junio 2018].
- [80] «developer.mozilla.org,» [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/CSS/Introduction_to_CSS/Como_funciona_CS

- S. [Último acceso: 30 junio 2018].
- [81] «es.wikipedia.org,» [En línea]. Available: <https://es.wikipedia.org/wiki/HTML>. [Último acceso: 30 junio 2018].
- [82] «www.maestrosdelweb.com,» 2007. [En línea]. Available: <http://www.maestrosdelweb.com/dom/>. [Último acceso: 30 junio 2018].
- [83] A. Html, «www.w3schools.com,» [En línea]. Available: https://www.w3schools.com/tags/tag_input.asp. [Último acceso: 4 julio 2018].
- [84] R. Data., «www.w3schools.com,» [En línea]. Available: https://www.w3schools.com/js/js_events.asp. [Último acceso: 4 julio 2018].
- [85] «www.w3schools.com,» [En línea]. Available: https://www.w3schools.com/js/js_ajax_intro.asp. [Último acceso: 6 julio 2018].
- [86] «expressjs.com,» [En línea]. Available: <http://expressjs.com/es/starter/hello-world.html>. [Último acceso: 23 junio 2018].
- [87] «ed.team,» [En línea]. Available: <https://ed.team/blog/introduccion-npm>. [Último acceso: 23 junio 2018].
- [88] «raspberrypifromscratch.blogspot.com,» [En línea]. Available: <http://raspberrypifromscratch.blogspot.com/2015/11/temperaturehumidity-notifications-with.html>. [Último acceso: 6 julio 2018].
- [89] «www.npmjs.com,» [En línea]. Available: <https://www.npmjs.com/package/node-dht-sensor>. [Último acceso: 17 julio 2018].
- [90] «www.npmjs.com,» [En línea]. Available: <https://www.npmjs.com/package/pigpio>. [Último acceso: 6 julio 2018].
- [91] «incredibits.io,» [En línea]. Available: <https://incredibits.io/project/getting-started-with-raspberry-pi/raspberry-pi-blinking-led>. [Último acceso: 17 julio 2018].
- [92] «tecnit.com.ec,» [En línea]. Available: <https://tecnit.com.ec/producto/adaptador-usb-wireless-n150-encore-wu8188cus5-antena-5dbi/>. [Último acceso: 23 julio 2019].
- [93] «www.ingmecafenix.com,» [En línea]. Available: <http://www.ingmecafenix.com/electricidad-industrial/servomotor/>. [Último acceso: 14 julio 2018].
- [94] «www.educachip.com,» [En línea]. Available: <http://www.educachip.com/optoacoplador-que-es-y-como-utilizarlo/>. [Último acceso: 14 julio 2018].