



# **Tarjeta Smart House para una habitación**

**César Andres Tejada Torres**  
**John Alejandro Barahona Pineda**

Universidad del Quindío  
Facultad de Ingeniería, Ingeniería Electrónica  
Armenia, Colombia  
2018



# Tarjeta Smart House para una habitación

**César Andres Tejada Torres**  
**John Alejandro Barahona Pineda**

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:  
**Ingeniero Electrónico**

Director:  
Ing. César Augusto Álvarez Gaspar

Línea de Investigación:  
Internet de las Cosas  
Universidad del Quindío  
Facultad de Ingeniería, Ingeniería Electrónica  
Armenia, Colombia  
2018



# Agradecimientos

Esta sección es opcional, en ella el autor agradece a las personas o instituciones que colaboraron en la realización de la tesis o trabajo de investigación. Si se incluye esta sección, deben aparecer los nombres completos, los cargos y su aporte al documento.



## Resumen

El resumen es una presentación abreviada y precisa (la NTC 1486 de 2008 recomienda revisar la norma ISO 214 de 1976). Se debe usar una extensión máxima de 12 renglones. Se recomienda que este resumen sea analítico, es decir, que sea completo, con información cuantitativa y cualitativa, generalmente incluyendo los siguientes aspectos: objetivos, diseño, lugar y circunstancias, pacientes (u objetivo del estudio), intervención, mediciones y principales resultados, y conclusiones. Al final del resumen se deben usar palabras claves tomadas del texto (mínimo 3 y máximo 7 palabras), las cuales permiten la recuperación de la información.

**Palabras clave: (máximo 10 palabras, preferiblemente seleccionadas de las listas internacionales que permitan el indizado cruzado).**

A continuación se presentan algunos ejemplos de tesauros que se pueden consultar para asignar las palabras clave, según el área temática:

**Artes:** AAT: Art y Architecture Thesaurus.

**Ciencias agropecuarias:** 1) Agrovoc: Multilingual Agricultural Thesaurus - F.A.O. y 2) GEMET: General Multilingual Environmental Thesaurus.

**Ciencias sociales y humanas:** 1) Tesauro de la UNESCO y 2) Population Multilingual Thesaurus.

**Ciencia y tecnología:** 1) Astronomy Thesaurus Index. 2) Life Sciences Thesaurus, 3) Subject Vocabulary, Chemical Abstracts Service y 4) InterWATER: Tesauro de IRC - Centro Internacional de Agua Potable y Saneamiento.

**Tecnologías y ciencias médicas:** 1) MeSH: Medical Subject Headings (National Library of Medicine's USA) y 2) DECS: Descriptores en ciencias de la Salud (Biblioteca Regional de Medicina BIREME-OPS).

**Multidisciplinarias:** 1) LEMB - Listas de Encabezamientos de Materia y 2) LCSH- Library of Congress Subject Headings.

También se pueden encontrar listas de temas y palabras claves, consultando las distintas bases de datos disponibles a través del Portal del Sistema Nacional de Bibliotecas<sup>1</sup>, en la sección Recursos bibliográficos. opción "Bases de datos".

## Abstract

Es el mismo resumen pero traducido al inglés. Se debe usar una extensión máxima de 12 renglones. Al final del Abstract se deben traducir las anteriores palabras claves tomadas del

---

<sup>1</sup>ver: [www.sinab.unal.edu.co](http://www.sinab.unal.edu.co)

texto (mínimo 3 y máximo 7 palabras), llamadas keywords. Es posible incluir el resumen en otro idioma diferente al español o al inglés, si se considera como importante dentro del tema tratado en la investigación, por ejemplo: un trabajo dedicado a problemas lingüísticos del mandarín seguramente estaría mejor con un resumen en mandarín.

**Keywords: palabras clave en inglés(máximo 10 palabras, preferiblemente seleccionadas de las listas internacionales que permitan el indizado cruzado)**



# **1 Objetivos**

## **1.1. Objetivo General**

Desarrollar una solución IoT para una habitación en un entorno de Smart House.

## **1.2. Objetivos Específicos**

- Desarrollar un prototipo de una tarjeta inalámbrica para una habitación en un entorno Smart House.
- Desarrollar una aplicación web encargada de permitir la interacción del usuario con su habitación.
- Evaluar el desempeño del sistema en una prueba beta.

# Contenido

<b>Agradecimientos</b>	<b>v</b>
<b>Resumen</b>	<b>vii</b>
<b>1 Objetivos</b>	<b>ix</b>
1.1 Objetivo General . . . . .	ix
1.2 Objetivos Específicos . . . . .	ix
<b>Lista de símbolos</b>	<b>xii</b>
<b>2 Introducción</b>	<b>1</b>
<b>3 Marco Teórico</b>	<b>2</b>
3.1 Internet de las Cosas . . . . .	2
3.1.1 Plataforma Heroku . . . . .	3
3.1.2 Framework Laravel . . . . .	3
3.1.3 HTTP . . . . .	3
3.1.4 JSON . . . . .	4
3.2 Smart House . . . . .	5
3.3 Hardware . . . . .	5
3.3.1 ESP-WROOM-32 . . . . .	5
3.3.2 Corriente Alterna (AC) . . . . .	6
3.3.3 Corriente Directa (DC) . . . . .	6
3.3.4 Control de potencia AC por ángulo de fase . . . . .	7
3.3.5 Control de Cargas DC . . . . .	7
3.3.6 I2C . . . . .	7
3.3.7 Sensores . . . . .	7
3.4 Software . . . . .	12
3.4.1 RTOS . . . . .	12
3.4.2 ESP-IDF . . . . .	14
3.4.3 Proteus . . . . .	14
<b>4 Desarrollo e Implementación</b>	<b>16</b>
4.1 Hardware . . . . .	16
4.2 Firmware . . . . .	16

---

4.3	Software . . . . .	17
4.3.1	Servidor Heroku . . . . .	17
4.3.2	Framework Laravel . . . . .	17
<b>5</b>	<b>Resultados y Análisis</b>	<b>18</b>
<b>6</b>	<b>Conclusiones</b>	<b>19</b>
<b>7</b>	<b>Trabajos Futuros</b>	<b>20</b>
<b>8</b>	<b>Glosario</b>	<b>21</b>
	<b>Bibliografía</b>	<b>23</b>

# Lista de símbolos

## Abreviaturas

Abreviatura	Término
<i>RF</i>	Radiofrecuencia
<i>IR</i>	Infrarrojo
<i>SMS</i>	Servicio de Mensajes Cortos
<i>GSM</i>	Sistema Global para las comunicaciones Móviles
<i>SBC</i>	Computadoras de Placa Simple
<i>IoT</i>	Internet de las Cosas
<i>AC</i>	Corriente alterna
<i>DC</i>	Corriente continua

## 2 Introducción

Con el avance de las tecnologías, y el crecimiento exponencial que ha tenido el internet, es posible enlazar comunicaciones con infinidad de dispositivos en múltiples lugares del mundo, accediendo a todo tipo de información, desde series de televisión, hasta datos reales generados por procesos industriales. Para que esto sea posible, se han creado sistemas capaces de enviar datos a la red (internet), con múltiples fines, ya sea para estudios estadísticos o simple información general. Uno de los objetivos principales de IoT, es manejar esta información de manera inteligente, es decir, el proceso de tomar decisiones y acciones significativas con esta información.

IoT puede ser implementado tanto a gran escala, como a una ciudad, o a una escala más reducida, como por ejemplo una habitación, sea una cocina, una pequeña bodega de alimentos, etc., en la cual existen múltiples variables a tener en cuenta. Para el análisis de las múltiples variables presentes en una habitación, se requiere un sistema capaz de, capturar la información, tomar decisiones inteligentes sobre esta información, para posteriormente ejecutar acciones que puedan influir directa o indirectamente con dichas variables.

La gran ventaja de IoT en relación a la toma de decisiones inteligentes sobre la información, es la capacidad de darle a esa información un propósito mayor, ya que puede ser empleado para algún tipo de realimentación con fines de mejorar procesos o como se mencionó anteriormente, con propósitos generales.

## 3 Marco Teórico

### 3.1. Internet de las Cosas

La internet de las cosas es un sistema de dispositivos de computación interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos y la capacidad de transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora.

IoT ha evolucionado desde la convergencia de tecnologías inalámbricas, sistemas microelectromecánicos, microservicios e internet. La convergencia ha ayudado a derribar las paredes de silos entre la tecnología operativa y la tecnología de la información, permitiendo que los datos no estructurados generados por máquinas sean analizados para obtener información que impulse mejoras. [1]

Kevin Ashton, cofundador y director ejecutivo del Auto-ID Center de MIT, mencionó por primera vez la internet de las cosas en una presentación que hizo a Procter & Gamble en 1999. He aquí cómo Ashton explica el potencial de la internet de las cosas:

“Las computadoras de hoy –y, por lo tanto, la internet– dependen casi totalmente de los seres humanos para obtener información. Casi todos los aproximadamente 50 petabytes (un petabyte son 1.024 terabytes) de datos disponibles en internet fueron capturados y creados por seres humanos escribiendo, presionando un botón de grabación, tomando una imagen digital o escaneando un código de barras.

El problema es que la gente tiene tiempo, atención y precisión limitados, lo que significa que no son muy buenos para capturar datos sobre cosas en el mundo real. Si tuviéramos computadoras que supieran todo lo que hay que saber acerca de las cosas –utilizando datos que recopilamos sin ninguna ayuda de nosotros– podríamos rastrear y contar todo, y reducir en gran medida los desechos, las pérdidas y el costo. Sabríamos cuándo necesitamos reemplazar, reparar o recordar cosas, y si eran frescas o ya pasadas”. [2]

### 3.1.1. Plataforma Heroku

“Heroku es una plataforma en la nube basada en un sistema gestionado por contenedores, con servicios de datos integrados y un potente ecosistema, para desarrollar y ejecutar aplicaciones modernas. La experiencia de los desarrolladores de Heroku es un enfoque centrado en aplicaciones para la entrega de software, integrado con las herramientas y flujos de trabajo de desarrollador más populares de la actualidad” [3].

### 3.1.2. Framework Laravel

“Laravel es un framework de aplicaciones web con una sintaxis expresiva y elegante. Laravel intenta aliviar el dolor del desarrollo al facilitar las tareas comunes que se utilizan en la mayoría de los proyectos web, como la autenticación, el enrutamiento, las sesiones y el almacenamiento en caché.

Laravel tiene como objetivo hacer que el proceso de desarrollo sea agradable para el desarrollador sin sacrificar la funcionalidad de la aplicación. Con este fin, se ha intentado combinar lo mejor de lo que hemos visto en otros marcos web, incluidos los marcos implementados en otros lenguajes, como Ruby on Rails, ASP.NET MVC y Sinatra.

Laravel es accesible, pero potente, y proporciona potentes herramientas necesarias para aplicaciones grandes y robustas. Una magnífica inversión de contenedores de control, un sistema de migración expresivo y un soporte de prueba de unidades estrechamente integrado le brindan las herramientas que necesita para construir cualquier aplicación” [4].

### 3.1.3. HTTP

El protocolo de transferencia de hipertexto (HTTP) es un protocolo de la capa de aplicación en el modelo OSI, se usa para transmitir documentos hipermedia, como HTML. Fue diseñado para la comunicación entre navegadores web y servidores web, pero también se puede usar para otros propósitos. HTTP sigue un modelo clásico de cliente-servidor, con un cliente que abre una conexión para realizar una petición, luego esperando la respuesta. HTTP es un protocolo sin estado, significa que el servidor no conserva ningún dato entre dos solicitudes. Para realizar las peticiones este protocolo cuenta con diferentes métodos [5].

**Metodos de solicitud HTTP:** HTTP define un conjunto de métodos de solicitudes para indicar la acción deseada que se realizará para un recurso determinado. Aunque pueden ser sustantivos, estos métodos algunas veces se denominan verbos HTTP. Cada uno de ellos implementa una semántica diferente, pero algunas características comunes son compartidas

por un grupo de ellos [6].

- GET: este método solicita una representación del recurso especificado. Las peticiones que lo usan, solo deben regresar datos.
- HEAD: este método solicita una respuesta igual a una petición GET, pero sin el cuerpo de la respuesta.
- POST: se usa para enviar una entidad al recurso especificado, causando a menudo un cambio en el estado o efectos secundarios en el servidor.
- PUT: reemplaza todas las representaciones actuales del recurso destino con la petición de carga útil.
- DELETE: esta petición elimina el recurso especificado.
- CONNECT: establece un túnel para el servidor identificado por el recurso de destino.
- OPTIONS: se usa para describir las opciones de la comunicación para el recurso destino.
- TRACE: realiza una prueba de mensaje loop-back a lo largo de la ruta del recurso de destino.
- PATCH: se usa para realizar modificaciones parciales a un recurso.

### 3.1.4. JSON

“JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos” [7].

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.



Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras [7].

## 3.2. Smart House

El concepto de Smart House implica tres características básicas. En primer lugar, el monitoreo a través de redes de sensores para obtener información sobre la casa y sus residentes. En segundo lugar, los mecanismos que controlan el uso de la comunicación entre dispositivos para permitir la automatización y el acceso remoto. Por último, las interfaces de usuario, como los teléfonos inteligentes y las computadoras que permiten a los usuarios especificar las preferencias, así como presentar información a las personas acerca de estas preferencias.

Smart House es un entorno que tiene sistemas sofisticados a través de los cuales se pueden controlar algunas de las cosas de la casa, como luces, puertas, ventanas, además puede racionalizar el consumo de energía, entre otras funciones mediante el uso de sensores. Básicamente, uno de los beneficios más importantes del uso de la tecnología en las casas, es la prestación de servicios a las personas.[8]

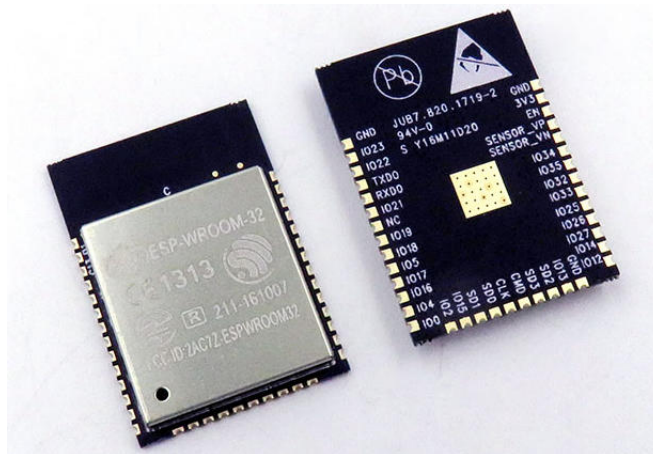
## 3.3. Hardware

### 3.3.1. ESP-WROOM-32

Es un potente módulo MCU Wi-Fi + BT + BLE que se dirige a una amplia variedad de aplicaciones, desde redes de sensores de baja potencia hasta las tareas más exigentes, como codificación de voz, transmisión de música y decodificación de MP3, además de su reducido tamaño, como se observa en la figura **3-1**.

En el núcleo de este módulo está el chip ESP32-D0WDQ6. El chip integrado está diseñado para ser escalable y adaptable. Hay dos núcleos de CPU que se pueden controlar individualmente, y la frecuencia del reloj es ajustable de 80 MHz a 240 MHz. El usuario también puede apagar la CPU y utilizar el coprocesador de baja potencia para monitorear constantemente los periféricos en busca de cambios o cruces de umbrales. ESP32 integra un amplio conjunto de periféricos, que van desde sensores táctiles capacitivos, sensores Hall, interfaz de tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

La integración de Bluetooth, Bluetooth LE y Wi-Fi garantiza que se pueda orientar una am-

**Figura 3-1:** ESP WROOM 32[10]

plia gama de aplicaciones, el uso de Wi-Fi permite un gran alcance físico y conexión directa a Internet a través de Wi-Fi, mientras usa Bluetooth, le permite al usuario conectarse convenientemente al teléfono o transmitir balizas de baja energía para su detección. La corriente de reposo del chip ESP32 es inferior a 5 uA, lo que lo hace adecuado para aplicaciones de electrónica con batería y portátiles. ESP32 admite una velocidad de datos de hasta 150 Mbps y una potencia de salida de 20.5 dBm en la antena para garantizar el rango físico más amplio.

El sistema operativo elegido para ESP32 es freeRTOS con LwIP; TLS 1.2 con aceleración de hardware está integrado también. También se admite la actualización segura (cifrada) a través del aire (OTA), de modo que los desarrolladores puedan actualizar continuamente sus productos incluso después de su lanzamiento.[9]

### 3.3.2. Corriente Alterna (AC)

“Es un tipo de corriente eléctrica, en la que la dirección del flujo de electrones va y viene a intervalos regulares o en ciclos. La corriente que fluye por las líneas eléctricas y la electricidad disponible normalmente en las casas procedente de los enchufes de la pared es corriente alterna. La corriente estándar utilizada en los EE.UU. es de 60 ciclos por segundo (es decir, una frecuencia de 60 Hz); en Europa y en la mayor parte del mundo es de 50 ciclos por segundo (es decir, una frecuencia de 50 Hz.)”. [11]

### 3.3.3. Corriente Directa (DC)

“Es la corriente eléctrica que fluye de forma constante en una dirección, como la que fluye en una linterna o en cualquier otro aparato con baterías es corriente continua.

Una de las ventajas de la corriente alterna es su relativamente económico cambio de voltaje. Además, la pérdida inevitable de energía al transportar la corriente a largas distancias es mucho menor que con la corriente continua”. [11]

#### **3.3.4. Control de potencia AC por ángulo de fase**

“Los SCR y los TRIAC, permiten aplicar una técnica muy conveniente y eficaz para controlar el voltaje promedio y por lo tanto la potencia aplicada a una carga, cambiando el ángulo de fase con el cual la fuente de voltaje se aplica a ésta. Esta técnica de control de voltaje es muy usada en las aplicaciones de regulación de motores, iluminación y temperatura, por ser el voltaje la variable principal en estos tres procesos”. [12]

“Para entender como se controla el ángulo de fase, por medio de un TRIAC conectado en serie con la carga, se puede asumir que el TRIAC se comporta idealmente como un interruptor controlador por la corriente de compuerta  $I_g$  que se cierra o se abre ante su presencia o ausencia. Observando la figura **3-2** puede verse el control de una onda seno de tensión con un período con un período de 360 grados; en la parte (a) de la figura se muestra la tensión a través del TRIAC, mientras que en la parte (b) se ve la tensión sobre la carga; allí puede verse que el TRIAC se comporta como un circuito abierto durante los primeros 45 grados de cada semiciclo, y todo el voltaje cae en sus terminales eliminando el flujo de corriente por la carga. La porción del semiciclo durante la cual se presenta esta situación se conoce como ángulo de disparo” [12].

“Una vez el TRIAC es disparado a través de su terminal de compuerta (G), éste se engancha y se comporta como un interruptor cerrado, permitiendo que todo el voltaje se aplique a la carga durante los 135 grados restantes de cada semiciclo. La porción del semiciclo durante la cual el TRIAC conduce se denomina ángulo de conducción” [12].

#### **3.3.5. Control de Cargas DC**

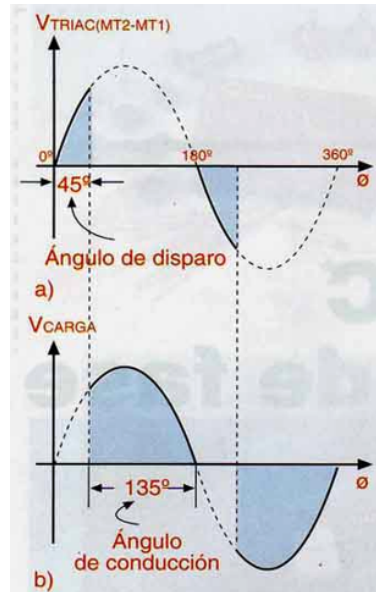
#### **3.3.6. I2C**

#### **3.3.7. Sensores**

##### **Módulo GY-30**

“Sensor GY-30 BH1750FVI. Es un sensor digital de intensidad de luz ambiente, tiene un conversor ADC de 16bits interno y comunicación por I2C como se observa en la figura **3-3**. Esta es una versión mejorada del típico sensor de luz a base de un LDR, el cual simplemente entrega un valor analógico. Compatible con Arduino, PIC, etc.

**Figura 3-2:** Representación gráfica del ángulo de disparo y de conducción del TRIAC y de la carga [12].



El módulo BH1750 es un sensor de luz, que a diferencia del LDR es digital y nos entrega valores de medición en Lux ( lumen /m<sup>2</sup> ) que es una unidad de medida estándar para el nivel de iluminación (iluminancia). Tiene alta precisión y un rango ente 1 – 65535 lx el cual es configurable.

La interfaz de comunicación es I2C pudiéndolo implementar en la mayoría de micro controladores, el módulo aparte de los pines de alimentación y pines I2C tiene un pin para establecer la dirección”.[13]

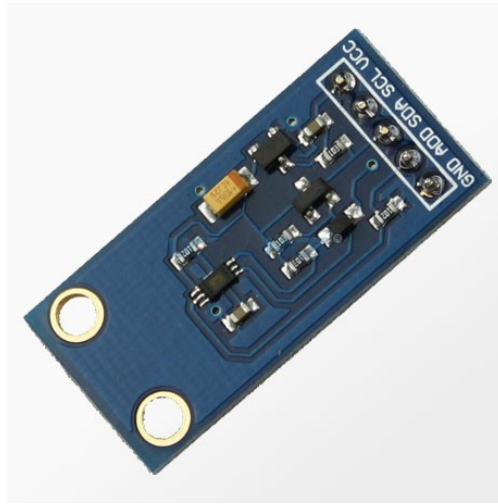
### Temperatura y Humedad DHT11

“El DHT11 es un sensor de temperatura y humedad digital de bajo costo. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica). Es bastante simple de usar, pero requiere sincronización cuidadosa para tomar datos. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán mínimo cada 2 segundos”. [14]

### Módulo sensor de calidad de aire MQ-135

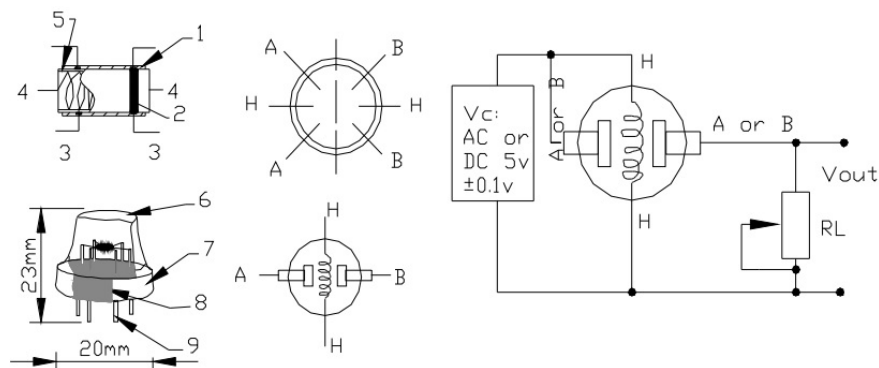
La serie MQ de sensores de gas son sensores analógicos por lo que son fáciles de implementar con cualquier microcontrolador que posea un conversor analógico digital (ADC) adecuado. Estos sensores son electroquímicos y varán su resistencia con la exposición a determinados

**Figura 3-3:** Modulo GY-30 [13]



**Figura 3-4:** Sensor de temperatura y humedad DHT11 [14]



**Figura 3-5:** Estructura del sensor MQ [15]**Figura 3-6:** Modulo sensor de calidad de Aire MQ-135 [15]

gases, internamente poseen un calentador que se encarga de aumentar la temperatura interna para que el sensor pueda reaccionar con los gases provocando un cambio de valor en la resistencia, su estructura interna se puede observar en la figura 3-5.[15]

El Sensor Calidad Aire MQ135 se utilizan en equipos de control de calidad del aire para edificios y oficinas, son adecuados para la detección de  $\text{NH}_3$ ,  $\text{NO}_x$ , alcohol, benceno, humo,  $\text{CO}_2$ , etc. Además de que estos sensores vienen en módulos como se observa en la figura 3-6, lo que facilita el uso de estos, simplemente se debe conectar al microcontrolador sin necesidad de hacer algún circuito de acople. [15]

Este sensor es sensible en similar proporción a los gases mencionados, con lo que se puede determinar si el aire está limpio o si existe presencia de algún gas nocivo.

Los valores de salida de este sensor no son valores absolutos, simplemente proporciona una salida analógica que se debe monitorear y ser comparada con los valores típicos proporcio-

**Figura 3-7:** Sensor de Lluvia [17]

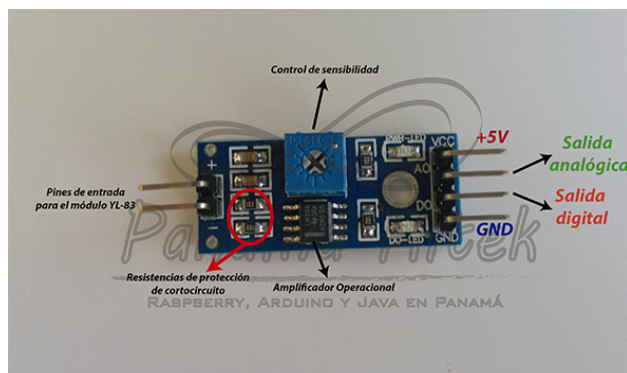
nados en la hoja de datos como se menciona anteriormente.[16]

### Sensores de Estado

Los sensores de estado describen si la variable esta en alto (1) o en bajo (0), para estos sensores se tienen diferentes variables tipicamente, como el estado de una puerta o una ventana (abierta o cerrada), la lluvia, el movimiento.

**Módulo detector de lluvia:** Es un módulo relativamente simple que consiste en una serie de pistas conductoras organizadas de forma paralela e impresas sobre una placa de baquelita como se observa en la figura 3-7. La separación entre sus pistas es muy pequeña, con el fin de crear un corto circuito cada vez que las pistas se mojan, ya que es un circuito abierto y el agua hace que se cree un camino de baja resistencia entre las pistas que tienen diferente potencial ( $V_{cc}$ -GND). La corriente que fluye a través de estas pistas se ve limitada por resistencias de 10K en cada conductor, lo que impide que el corto circuito que se genera cuando se moja la placa vaya a estropear el micro controlador.[17]

El circuito de control es el que posee las resistencias limitadoras de corriente y es el encargado de alimentar el módulo de la figura 3-7. Como se observa en la figura 3-8 posee un amplificador operacional, específicamente el circuito integrado LM392. Este es el encargado de amplificar el pequeño diferencial de voltaje que se genera cuando una gota de agua cae sobre las pistas del módulo. Aquí es donde se genera la señal de salida que puede ser del tipo analógica o digital. La señal digital oscilará entre los valores HIGH y LOW dependiendo de si hay agua o no sobre las pistas de la placa.



**Figura 3-8:** Módulo de sensor de lluviar [17]

La salida analógica entregará un nivel de voltaje que variará dependiendo de la cantidad de agua que haya sobre el módulo.[17]

**Módulo PIR HC-SR501:** La función de los sensores PIR es detectar movimiento, normalmente se busca detectar el movimiento de una persona dentro del rango del sensor. Son baratos, pequeños, de bajo consumo y fáciles de utilizar, además no se desgastan. Normalmente se encuentran en electrodomésticos y gadgets para la oficina o el hogar. Son conocidos como PIR, “Sensores Infrarrojos” o “Sensores de movimiento”.

Este módulo contiene un sensor Piroelectrico, el cual puede detectar niveles de radiación infrarroja como se observa en la figura 3-9. Este sensor de movimiento está dividido en 2 mitades, la razón para esto es que se busca la diferencia en el movimiento y no el promedio. Las dos mitades están unidas de modo que se cancelan una a otra. Entonces, si una mitad recibe más o menos radiación IR, la salida cambiará a Alto o Bajo. [18]

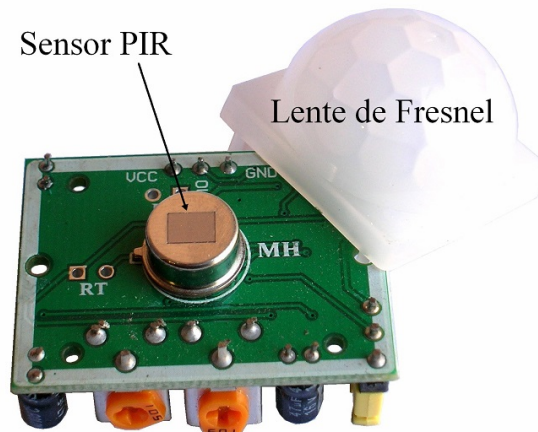
El módulo PIR modelo HC-SR501, es pequeño y de bajo costo como se observa en la figura , incorporando la tecnología más reciente en sensores infrarrojos pasivos para detectar movimiento. La emisión infrarroja se da en personas por su temperatura comporal y en animales (mamíferos), ya que emiten una radiación similar a los humanos. [19]

## 3.4. Software

### 3.4.1. RTOS

Los sistemas operativos en tiempo real, tienen como parámetro clave al tiempo, ya que en gran variedad de situaciones, como por ejemplo, un proceso industrial, se requiere recolectar múltiples datos, los cuales son usados para el control de múltiples procesos, los cuales deben ser ejecutados en determinados instantes, de no ser así, podría causar desde la mala ejecución



**Figura 3-9:** Sensor HCSR501 [19]

de una tarea, hasta un accidente según la delicadeza del proceso.

Para procesos con nula tolerancia a fallos, se conoce como un sistema en tiempo real duro, muchos de estos sistemas se encuentran en el control de procesos industriales, en aeronáutica, en la milicia y en áreas de aplicación similares. el caso contrario, cuando se tiene cierta tolerancia a que muy ocasionalmente existan fallos, se conoce como sistema en tiempo real suave, los sistemas de audio digital o de multimedia están en esta categoría. Los teléfonos digitales también son ejemplos de sistema en tiempo real suave. [20]

“Como en los sistemas en tiempo real es crucial cumplir con tiempos predeterminados para realizar una acción, algunas veces el sistema operativo es simplemente una biblioteca enlazada con los programas de aplicación, en donde todo está acoplado en forma estrecha y no hay protección entre cada una de las partes del sistema. Un ejemplo de este tipo de sistema en tiempo real es freeRTOS. Las categorías de sistemas para computadoras de bolsillo, sistemas integrados y sistemas en tiempo real se traslapan en forma considerable. Casi todos ellos tienen por lo menos ciertos aspectos de tiempo real suave. Los sistemas integrados y de tiempo real sólo ejecutan software que colocan los diseñadores del sistema; los usuarios no pueden agregar su propio software, lo cual facilita la protección.

Los sistemas de computadoras de bolsillo y los sistemas integrados están diseñados para los consumidores, mientras que los sistemas en tiempo real son más adecuados para el uso industrial. Sin embargo, tienen ciertas características en común”. [20]

### 3.4.2. ESP-IDF

ESP-IDF es el entorno de desarrollo oficial para el ESP32 desarrollado por Espressif System, el cual mediante una serie de comandos específicos escritos en la terminal (para el caso de linux), permite realizar una configuración del ESP32 en cuanto a su funcionamiento, es decir, permite encender o apagar características como el WiFi, el Bluetooth o realizar particiones de memoria, además de esto, se puede cargar el código por el puerto USB al ESP32, al igual que se puede visualizar la información generada por el ESP32 por este mismo puerto. Este entorno se encuentra construido con diferentes características y APIs, algunas de ellas se mencionan a continuación. [21]

**FreeRTOS:** este framework está desarrollado sobre este sistema operativo de tiempo real, tomando de este diferentes funcionalidades como las tareas y las colas.

**Wi-Fi:** el módulo ESP-WROOM-32 posee la capacidad en el hardware para la conexión a una red Wi-Fi, y este software proporciona las librerías y las funcionalidades para realizar dicha conexión y enviar o recibir datos.

**Particiones:** el MCU contiene una memoria flash, la cual por medio de este framework se le pueden realizar particiones de memoria, dedicadas a guardar el programa, y también para guardar ciertos archivos. Estas se configuran por medio de un fichero csv.

**Consola:** el framework provee una librería para realizar la programación de una consola, para realizar diferentes tareas dentro del sistema, las cuales se pueden programar.

**OTA:** este entorno permite además manejar la memoria, de tal forma que la aplicación ya compilada se pueda enviar desde un servidor hacia el dispositivo, para que este se actualice.

**HTTP Request:** para realizar diferentes peticiones HTTP como las descritas anteriormente, el entorno de desarrollo cuenta con la librería LWIP para realizarlas.

**Timers:**

### 3.4.3. Proteus

Proteus combina facilidad de uso con características de gran alcance para ayudar a diseñar, probar y el diseño de PCB profesionales. Con casi 800 variantes de microcontroladores listos para la simulación directamente desde el esquemático, además de contar con uno de los paquetes de diseño de PCB profesionales más intuitivos en el mercado y un autoruteo de clase

---

mundial incluido como estándar. [22]

Además es una herramienta muy completa y potente de simulación de circuitos y diseño de PCBs. Dentro de la simulación de circuitos admite componentes pasivos, digitales, analógicos y componentes más complejos como LCDs y motores. Por tanto, se puede hacer casi cualquier cosa, el programa está pensado para que una vez se tenga el circuito diseñado se pueda pasar a una PCB [23]. Adicionalmente los dispositivos que no se encuentren se pueden agregar, ya sea de la pagina oficial o construyendolos dentro de la herramienta.

## 4 Desarrollo e Implementación

### 4.1. Hardware

El hardware se diseña por medio del software Proteus ...

### 4.2. Firmware

El firmware se desarrolla sobre el framework o SDK oficial de Espressif Systems, ESP-IDF el cual posee una documentación ...

Sobre el firmware se desarrollan los siguientes temas:

#### **Consola**

Se usa para hacer pequeñas pruebas por medio de la construcción de comandos ligados a funciones ...

#### **HTTP Request**

Para realizar las diferentes peticiones de HTTP se utiliza la librería lwip ...

#### **Tareas**

Las tareas son funciones propias de los sistemas operativos en tiempo real ...

#### **Timers**

Son utilizados principalmente para el control de cargas AC ...

#### **I2C**

El protocolo I2C se activa por medio de la instalación del driver en algún par de pines ...

#### **PWM**

Se usan para el control de cargas DC ...

## **GPIO**

El ESP-WROOM-32 posee diferentes GPIO, los cuales se usan para leer o escribir señales digitales a los sensores o ...

## **Interrupciones**

Las interrupciones se usan para no gastar recursos en un monitoreo constante de las entradas, solo cuando exista un cambio de nivel en la entrada el dispositivo desencadena una serie de instrucciones ...

## **4.3. Software**

### **4.3.1. Servidor Heroku**

### **4.3.2. Framework Laravel**

## **5 Resultados y Análisis**

## **6 Conclusiones**

## **7 Trabajos Futuros**



## 8 Glosario

**AC (Corriente alterna):** corriente eléctrica variable en la que las cargas eléctricas (electrones) cambian el sentido del movimiento a través de un conductor de manera periódica.

**DC (Corriente continua):** corriente de intensidad constante en la que el movimiento de las cargas eléctricas (electrones) siempre es en el mismo sentido.

**Internet del todo (IoE):** es un concepto que extiende el énfasis de la internet de las cosas (IoT) en las comunicaciones de máquina a máquina para describir un sistema más complejo que también abarca personas y procesos.[24]

**Internet de las Cosas (IoT):** parte fundamental del internet del todo (IdT), el cual se refiere principalmente a la interacción máquina-máquina, en incluso interacción máquina-persona.

**Software:** conjunto de programas y rutinas que permiten a un sistema realizar determinadas tareas.

**Hardware:** partes físicas que componen un sistema electrónico, como por ejemplo los componentes de un circuito electrónico.

**Firmware:** programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo, es decir, software que maneja físicamente al hardware.

**Radiofrecuencia:** es la porción del espectro electromagnético (frecuencias) que es empleado en la radiocomunicación.

**Infrarrojo:** se refiere a la radiación electromagnética con longitud de onda mayor (menor energía) a la de la luz visible por el ser humano.

**PIC:** familia de microcontroladores tipo RISC (Computador con Conjunto de Instrucciones Reducidas). [25]

**Zigbee:** es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal. [26]

**HTML5:** siendo la última versión de HTML, contiene elementos, atributos y comportamientos nuevos, además de un conjunto más amplio de tecnologías que proporciona mayor diversidad y alcance a los sitios Web.

**PHP (Preprocesador de hipertexto):** es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

**SQL (lenguaje de consulta estructurado):** es un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla.

**VisualBasic .NET:** es un lenguaje de programación orientado a objetos que cuenta con los beneficios que le brinda .NET Framework, el modelo de programación diseñado para simplificar la programación de aplicaciones en un entorno sumamente distribuido como lo es Internet.[27]

# Bibliografía

- [1] TechTarget. Internet de las cosas (iot).
- [2] Kevin Ashton. That 'internet of things' thing. 2009.
- [3] Heroku Inc. The heroku platform, 2018.
- [4] Laravel. Laravel philosophy, 2013.
- [5] MDN web docs Mozilla. Http, 2018.
- [6] MDN web docs mozilla. Http request methods.
- [7] JSON.org. Introducción a json.
- [8] Aadel Howedi and Ali Jwaid. Design and implementation prototype of a smart house system at low cost and multi-functional. 2016.
- [9] Espressif Systems. Esp-wroom-32 datasheet, 2018.
- [10] Electrodragon. Esp-wroom-32.
- [11] Greenfacts. Corriente alterna y corriente continua.
- [12] CEKIT. *Curso práctico de Electrónica Industrial y Automatización*. 2001.
- [13] AMGkits. Sensor de intensidad óptica (sensor de luz) gy-30 bh1750fvi.
- [14] ElectroniLab. Sensor de temperatura y humedad dht11.
- [15] Naylamp Mechatronics. Tutorial sensores de gas mq2, mq3, mq7 y mq135.
- [16] Geekbot Electronics. Mq-135 módulo sensor de calidad del aire.
- [17] Antony Garcia Gonzales. Módulo yl-83: Un detector de lluvia, 2014.
- [18] Naylamp Mechatronics. Módulo pir hc-sr501.
- [19] Punto Flotante S.A. Pruebas al módulo sensor pasivo infrarrojo pir hc-sr501. sensor de movimiento (body motion).

- [20] Andrew S. Tanenbaum. *Sistemas Operativos Modernos*. 3ra edition, 2009.
- [21] Espressif Systems. Esp-idf programming guide.
- [22] Labcenter Electronics Ltd. Proteus design suite.
- [23] Geeky Theory. ¿qué es proteus?
- [24] Margaret Rouse. Internet de todo (ioe), 2017.
- [25] Wikipedia. Microcontrolador pic, 2017.
- [26] Zigbee. What is zigbee?, 2017.
- [27] Universidad Nacional Autonoma de Mexico. Lenguaje de programación visual basic .net, 2009.