



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Diseño de una red domótica para monitoreo y alerta de emergencias en
viviendas con adultos mayores utilizando Arduino MKR WiFi 1010**

AUTOR:

Arbeláez Martínez, Carlos Alfredo

Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Zamora Cedeño, Néstor Armando

Guayaquil, Ecuador

15 de septiembre del 2021



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr.
Arbeláez Martínez, Carlos Alfredo como requerimiento para la obtención
del título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

MSc. Ing. Zamora Cedeño, Néstor Armando

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, a los 15 días del mes de septiembre del año 2021



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Arbeláez Martínez, Carlos Alfredo**

DECLARO QUE:

El trabajo de titulación “**Diseño de una red domótica para monitoreo y alerta de emergencias en viviendas con adultos mayores utilizando Arduino MKR WiFi 1010**”, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 15 días del mes de septiembre del año 2021

EL AUTOR

ARBELÁEZ MARTÍNEZ, CARLOS ALFREDO



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Arbeláez Martínez, Carlos Alfredo**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Diseño de una red domótica para monitoreo y alerta de emergencias en viviendas con adultos mayores utilizando Arduino MKR WiFi 1010”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 15 días del mes de septiembre del año 2021

EL AUTOR

ARBELÁEZ MARTÍNEZ, CARLOS ALFREDO

REPORTE DE URKUND

URKUND

Documento	Tesis Arbelaez.pdf (D111435890)
Presentado	2021-08-18 14:01 (-05:00)
Presentado por	Néstor Zamora (nestor.zamora@cu.ucsg.edu.ec)
Recibido	nestor.zamora.ucsg@analysis.orkund.com
Mensaje	Análisis Trabajo de Titulación Arbeláez Mostrar el mensaje completo 3% de estas 23 páginas, se componen de texto presente en 4 fuentes.

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA

EN TELECOMUNICACIONES TEMA: Diseño de una red

domótica para monitoreo y alerta de emergencias en viviendas con adultos mayores utilizando Arduino MKR WiFi 1010 AUTOR: Arbeláez Martínez, Carlos Alfredo

Trabajo de Titulación

previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES TUTOR: M. Sc. Zamora Cedeño, Néstor Armando Guayaquil, Ecuador 28 de agosto del 2021

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DEDICATORIA

Este trabajo está dedicado principalmente a mi madre, a la memoria de mi padre, a mi esposa y a mis tesoros Amelia y Nicolas, por quienes, a pesar de las ideas de abandonar la universidad por las diferentes dificultades que se presentaron, fueron mi motor para avanzar y nunca tirar la toalla. También está dedicado a quienes siempre confiaron en mis capacidades y por quienes no podía bajar los brazos y defraudar dicha confianza.

EL AUTOR

ARBELÁEZ MARTÍNEZ, CARLOS ALFREDO

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por las fuerzas y oportunidades con las que me ha bendecido. En segundo lugar, a mi madre y a la memoria de mi padre quienes desde pequeño fueron un ejemplo de lucha y decisión, enseñándome que no existe nada imposible de conseguir con trabajo y dedicación.

Agradezco también a todas aquellas personas que me motivaron a retomar mis estudios, porque sin ese consejo oportuno tal vez en este momento no me encontraría en estas instancias.

Y finalmente agradezco a los maestros que tuve el gusto de conocer en este camino universitario, entre ellos a mi tutor, quienes supieron orientarme académicamente y brindar su conocimiento sin reparos ni mezquindades.

EL AUTOR


ARBELÁEZ MARTÍNEZ, CARLOS ALFREDO



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. 

M. Sc. ROMERO PAZ, MANUEL DE JESÚS
DECANO

f. 

M. Sc. HERAS SÁNCHEZ, MIGUEL ARMANDO
DIRECTOR DE CARRERA

f. 

M. Sc. VALLEJO SAMANIEGO, LUIS VICENTE
OPONENTE

Índice General

Índice de Figuras	XI
Resumen	XIII
Capítulo 1: Descripción General del Trabajo de Titulación	2
1.1. Introducción.	2
1.2. Antecedentes.	2
1.3. Definición del Problema.	3
1.4. Justificación del Problema.	4
1.5. Objetivos del Problema de Investigación.	4
1.5.1. Objetivo General.	4
1.5.2. Objetivos Específicos.	4
1.6. Hipótesis.	4
1.7. Metodología de Investigación.	5
Capítulo 2: Fundamentación Teórica	6
2.1. Red de sensores inalámbricos.	6
2.1.1 Elementos que conforman una red inalámbrica de sensores	7
2.2. Sistemas domóticos.	10
2.2.1. Características de un sistema domótico robusto	11
2.2.2. Campos de aplicación.	12
2.2.3. IoT en la domótica	14
2.2.4. Arduino en la domótica	15
2.2.4.1. Antecedentes	15
2.2.4.2. Debilidades y fortalezas de Arduino frente a otros microcontroladores	16
2.2.4.3. Arduino IoT Cloud	19
2.2.4.4. Esquema general para proyectos de IoT con Arduino	20
Capítulo 3: Diseño y Análisis de Resultados	21
3.1. Diseño de la red de sensores	21
3.1.1 Diagrama de bloques de la red	21
3.1.2 Arduino MKR WiFi 1010	21
3.1.3 Esquema Arduino MKR WIFI 1010	22

3.1.4	Módulo ESP 8266 MCU.....	24
3.1.5	Sensores.....	24
3.2.	Montaje y configuración de la red.....	25
3.2.1	Conexión de los elementos.....	26
3.2.2	Configuración del entorno de desarrollo.....	27
3.2.3	Entorno local.....	27
3.2.4	Entorno cloud.....	31
3.3.	Programación de la red.....	38
3.3.1.	Punto de Acceso.....	38
3.3.2.	Gateway.....	41
3.3.3.	Nodos sensores.....	45
	Capítulo 4: Conclusiones y Recomendaciones.....	51
4.1.	Conclusiones.....	51
4.2.	Recomendaciones.....	52
	Bibliografía.....	53

Índice de Figuras

Capítulo 2

Figura 2. 1: Red Sensorial	6
Figura 2. 2: Diagrama de bloques del nodo sensor.	7
Figura 2. 3: Elementos de una res de sensores inalámbrica.	7
Figura 2. 4: Topologías comunes de redes inalámbricas.....	8
Figura 2. 5: Arquitectura de red IEEE 802.15.4.	10
Figura 2. 6: Ambiente domótico en vivienda	11
Figura 2. 7: Servicios domóticos	14
Figura 2. 8: Crecimiento de elementos inteligentes conectados a internet..	14
Figura 2. 9: Microcontroladores	17
Figura 2. 10: El universo Arduino.....	18
Figura 2. 11: Entorno de desarrollo local	18
Figura 2. 12: Entorno de desarrollo cloud	19
Figura 2. 13: Opciones de conectividad disponibles	20

Capítulo 3

Figura 3. 1: Diagrama de bloques WSN	21
Figura 3. 2: Arduino MKR WIFI 1010	22
Figura 3. 3: Esquema electrónico Arduino MKR WIFI 1010.....	22
Figura 3. 4: Esquema electrónico ATSAM21G18A-48QFN.....	23
Figura 3. 5: Esquema electrónico módulo NINA-W102-00B	23
Figura 3. 6: Diagrama de pines de Módulo ESP8266 MCU.....	24
Figura 3. 7: Módulo MQ-2	25
Figura 3. 8: Módulo MQ-2	25
Figura 3. 9: Conexión MCU con Arduino 1	26
Figura 3. 10: Conexión MCU con MQ-2 1	26
Figura 3. 11: Conexión MCU con MQ-2 3.....	27
Figura 3. 12: Ingreso al Gestor de Tarjetas	28
Figura 3. 13: Instalación módulo Arduino MKR WiFi 1010.....	28
Figura 3. 14: Arduino MKR WiFi 1010 disponible.	29
Figura 3. 15: Abrir Gestor de Librerías.....	29
Figura 3. 16: Gestor de Librerías	30
Figura 3. 17: Instalación de librería DHT sensor library	30

Figura 3. 18: Arduino Home Page.....	31
Figura 3. 19: Sign in Page	31
Figura 3. 20: Ingreso al entorno IoT Cloud	32
Figura 3. 21: Add Device Modal.....	32
Figura 3. 22: Arduino MKR WiFi 1010 Encontrado	33
Figura 3. 23: Configuración inicial del dispositivo en la plataforma cloud. ...	33
Figura 3. 24: Asignación de nombre al dispositivo.....	34
Figura 3. 25: Entorno inicial things.....	34
Figura 3. 26: Thing creada.....	35
Figura 3. 27: Renombre de thing	35
Figura 3. 28: Opción para el registro de dispositivo	36
Figura 3. 29: Modal para registro de dispositivo	36
Figura 3. 30: Dispositivo creado.....	37
Figura 3. 31: Creación de variable	38
Figura 3. 32: Configuración modo Access Point	39
Figura 3. 33: Visualización de la red generada.....	39
Figura 3. 34: Métodos de configuración.....	40
Figura 3. 35: Definición método handleRootResponse.....	40
Figura 3. 36: Definición método “sensorRequest”.....	41
Figura 3. 37: Definición métodos “setup” y “loop”	41
Figura 3. 38: Archivo de configuración “arduino_secret.h”.....	42
Figura 3. 39: Archivo de configuración “thingProperties.h”	43
Figura 3. 40: Configuración inicial de archivo principal	43
Figura 3. 41: Definición método “readSerialPort”	44
Figura 3. 42: Definición de método “setup”	44
Figura 3. 43: Definición de método “loop”	45
Figura 3. 44: Firmware nodo sensor de gas	46
Figura 3. 45: Firmware nodo sensor de temperatura	47
Figura 3. 46: Firmware nodo sensor de movimiento	48
Figura 3. 47: Dashboard de monitoreo.	48
Figura 3. 48: Alerta de evento con Blynk.	49
Figura 3. 49: Alerta de evento con Blynk 2.	50
Figura 3. 50: Alerta de evento con Blynk en smart band.	50

Resumen

La evolución de los sistemas domóticos se ha convertido en los últimos años en una gran opción para tecnificar los hogares. La toma de decisiones es uno de los principales retos de un hogar inteligente, la pronta identificación y alerta de posibles emergencias es quizá el punto de quiebre para el incremento en el uso de esta tecnología; a esto se le suma la posibilidad de una conectividad continua, que abre la puerta a la monitorización en tiempo real de todos los dispositivos vinculados. La necesidad del momento es brindar seguridad a los hogares en general, pero en mayor medida a las viviendas de adultos mayores quienes son más proclives a sufrir accidentes que se agravan por la falta de compañía en la mayoría de los casos. En este trabajo se propone una opción integral para interconectar una red de sensores con los servicios cloud de Arduino, los cuales se orientan al IoT, la cual garantiza el correcto funcionamiento y fácil escalamiento de dispositivos sensoriales en la red domótica. La idea propuesta es implementada para probar de manera práctica y validar su eficacia y utilidad.

Palabras claves: SISTEMAS, DOMÓTICA, SENSORES, ARDUINO, CLOUD, IOT

Capítulo 1: Descripción General del Trabajo de Titulación

1.1. Introducción.

Se denomina domótica al conjunto de elementos tecnológicos que conllevan a la automatización total o parcial de actividades cotidianas en un hogar común. Es desde ya, para muchos hogares un elemento fundamental y con proyección a corto tiempo de convertirse en una parte vital de las familias; y esto debido a que, su evolución a grandes escalas llega incluso a integrarse con servicios en línea. Los dispositivos que pueden intervenir en un ecosistema domótico son tan diversos como las necesidades de cada hogar, en algunos se pondrá más énfasis en los de entretenimiento, en otros se enfocarán en los de prevención de riesgos y otros en los de alerta de emergencias.

Los elementos de la red destinados a brindar seguridad son una gran herramienta para bajar la tasa de accidentes domésticos por fugas de gases, incendios o por intromisión de personas externas a la vivienda. A esta tecnología se le suma la posibilidad de consultar y tabular los datos que registran los nodos para realizar análisis o alertar si estos niveles sobrepasan el umbral denominado como seguro. Focalizando el uso de estas herramientas en los hogares con adultos mayores podemos ayudar al cuidado de estos, permitiendo una reacción oportuna; teniendo en cuenta que, la pandemia del COVID-19 ha incrementado la brecha de cercanía en las viviendas de esta población identificada como vulnerable.

1.2. Antecedentes.

A finales de la década de los 50's según los expertos fue la época que marcó el inicio de los sistemas automatizados para una vivienda, este acontecimiento se le atribuye a Joe Spira, creador del atenuador de luz el cual patentó el 15 de julio de 1959 y que posterior a este invento fundaron Lutron Electronics Company en Coopersburg, Pensilvania en 1961. Luego de esto, pasaron quince años para que se diera un gran paso no solo a la domótica, sino a toda la electrónica, con la invención del X10, un sistema de corrientes portadoras el cual era de fácil instalación, esto abrió las puertas para que se

destinara más recursos al estudio de esta tecnología en auge, con lo cual al poco tiempo aparecieron protocolos como el KNX y el LON que debido a la posibilidad que brindaban de gestionar el consumo de energía, regular la iluminación provocaron una gran sensación.(Domotizados, 2018a)

A finales de la década de los 80's y principios de los 90's con el auge de las computadoras, los edificios integraron elementos tecnológicos como el cableado estructurado el cual facilitaba la conexión de los dispositivos de la nueva era y tras una lenta pero segura introducción de la tecnología digital oficialmente se inició la revolución de los servicios en los hogares. En esta nueva era tecnológica nació un dispositivo de código abierto, que por su fácil uso y bajo costo, despuntó de entre sus competidores; este elemento se llama Arduino y a pesar de que en sus inicios fue enfocado para temas educativos, su evolución ha llegado al punto de ser considerado en muchos proyectos a nivel empresarial. La gran cantidad de módulos compatibles con Arduino han impulsado a su uso en proyectos domóticos para la administración y control de redes sensoriales con el objetivo de brindar seguridad en un determinado espacio. Normalmente los sistemas de seguridad se enfocan en grandes edificaciones, pero para la domótica este no es un aspecto a considerar, debido a que puede ser utilizado en ambos ambientes prometiendo una gran fiabilidad (Domotizados, 2018b).

En el trabajo de Antonio Olmeda Martínez del año 2018, en el cual diseña una central domótica de bajo costo, utilizando Arduino como hardware se puede evidenciar las ventajas y la facilidad de implementación de este tipo de redes. (Olmeda Martínez, 2018)

1.3. Definición del Problema.

Los adultos mayores con frecuencia permanecen solos en sus hogares, lo cual genera la necesidad de monitorear de forma remota la existencia de algún peligro o emergencia que pueda suscitarse y que podría conllevar a un riesgo para su integridad; adicionalmente, la pandemia del COVID-19 ha incrementado esta preocupación, debido a que las normas de cuidado para la prevención del contagio a este grupo vulnerable, dificulta el acercamiento

recurrente de sus familiares para constatar posibles riesgos existentes en la vivienda.

1.4. Justificación del Problema.

Los riesgos de sufrir accidentes en los domicilios deben ser tenidos en cuenta con el objetivo de evitarlos o reaccionar a tiempo ante ellos, porque, de no ser así, se podrían provocar daños a la salud, tales como: irritaciones, quemaduras, problemas respiratorios e incluso la muerte. Por esta razón el uso de tecnologías en auge como la domótica integrando IoT son herramientas que ayudan a cumplir el objetivo de una vivienda segura, y si a esto se le suma la fácil escalabilidad de un dispositivo tecnológico como lo es Arduino y sus servicios cloud, se puede obtener un sistema robusto de bajo costo y alta eficiencia.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Diseñar una red domótica mediante el uso del Arduino MKR WiFi 1010 para monitoreo en tiempo real y alerta en emergencias, mediante dispositivos utilizados en las viviendas para la seguridad de los adultos mayores.

1.5.2. Objetivos Específicos.

- Describir los fundamentos domóticos básicos y las características del internet de las cosas.
- Definir los elementos que se utilizan en la red.
- Diseñar la red de sensores inalámbricos integrada a la plataforma cloud de Arduino.
- Analizar los resultados obtenidos en las pruebas de monitoreo con la red implementada.

1.6. Hipótesis.

Con la red propuesta utilizando el Arduino MKR WiFi 1010 y los servicios cloud se permite una pronta reacción ante eventos que ponen en riesgo la integridad física de los adultos mayores, brindándoles así un hogar seguro.

1.7. Metodología de Investigación.

Se utilizó el tipo de investigación descriptiva para poder explicar y detallar el proceso de la comunicación entre los sensores y los servicios cloud de Arduino que intervienen en el monitoreo y alerta ante emergencias.

En este trabajo el uso de la metodología analítica se vio reflejado en el estudio que conllevó a la elección de los sensores utilizados puesto que por sus características de funcionamiento benefician a la seguridad de viviendas con adultos mayores.

Se aplicó la metodología preexperimental realizando una evaluación previa de los resultados en un entorno controlado y supervisando una sola variable o sensor que interviene en la red para constatar la eficacia del sistema ante emergencias detectadas salvaguardando así a los adultos mayores.

Capítulo 2: Fundamentación Teórica

2.1. Red de sensores inalámbricos.

Las WSN o redes de sensores inalámbricos, por sus siglas en inglés, están compuestas por dispositivos de poco consumo energético y de bajo costo, entre sus principales características destaca la autonomía y las múltiples aplicaciones en las que estos elementos pueden intervenir. Estos dispositivos de una manera más genérica toman el nombre de nodos sensores y son capaces de comunicarse con otros elementos nodos mediante señales de radio; además de ser administrados remotamente por uno o varios controladores, los cuales son encargados de recibir las mediciones o alertas para posteriormente procesar o transmitir dicha información a través de otra red local o una global. (Eiko Yoneki, 2005)

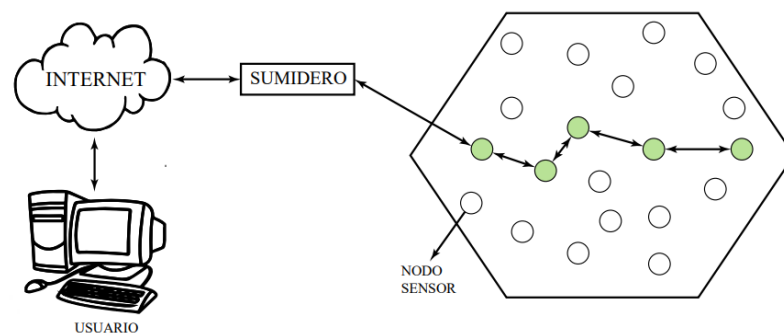


Figura 2. 1: Red Sensorial
Fuente: (Granda & Andrés, 2017)

Como muchos de los inventos de la era moderna, las redes de sensores inalámbricos fueron desarrolladas con fines militares durante la primera y segunda guerra mundial. Estos dispositivos eran utilizados por los militares para poder identificar la ubicación de sus adversarios en el área marítima. La versatilidad de estas redes de sensores hace posible que los usuarios puedan configurarla de acuerdo con sus necesidades y a las características de los nodos que la componen, estas características van desde la lectura de señales para almacenamiento y procesamiento autónomo hasta cumplir el rol de disparadores de notificaciones para los controladores de la red.

Los nodos de una red sensorial inalámbrica son dispositivos capaces de obtener información de su entorno, procesarla localmente y transmitirla a

través de enlaces inalámbricos, los elementos que conforman un nodo sensor los que se pueden observar en la siguiente imagen.

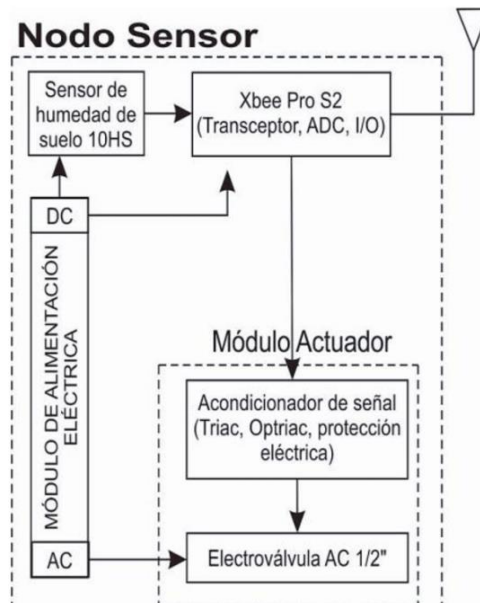


Figura 2. 2: Diagrama de bloques del nodo sensor.
Fuente: (Castro C. et al., 2016)

2.1.1 Elementos que conforman una red inalámbrica de sensores

Los nodos normalmente se encuentran conformados por placas de sensores o de elementos que adquieren datos de su entorno, además de una placa que tiene como función procesar, transmitir o recibir señales de radio la cual lleva el nombre de mota. Estos dispositivos sensoriales son capaces de comunicarse con un Gateway, que entre sus características cuenta con la posibilidad de comunicarse con ordenadores, además de otras redes incluido internet.(Fernandez Barcell, 2014)

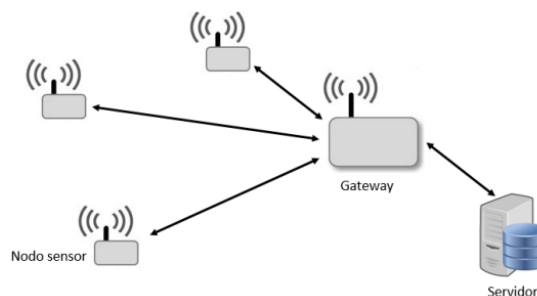


Figura 2. 3: Elementos de una res de sensores inalámbrica.
Fuente:(Castaño, 2019)

2.1.2 Sistema de adquisición de datos

Existen sensores de distintas naturalezas y tecnologías. Toman información de su entorno físico y la transforma en señales eléctricas que

interactuarán en la red. Los sensores de medida de radiación solar, luz, humedad en el suelo o aire GPS, temperatura, velocidad, sonido, etc. Son una muestra de la diversidad de parámetros que pueden ser extraídos utilizando estos dispositivos.(Fernandez Barcell, 2014)

2.1.3 Motas

Las motas son las encargadas de suministrar las características de comunicación y procesamiento al nodo sensor. Su función de procesamiento toma los datos que proporciona el sensor mediante las puertas de datos y posterior a ello envía la información hasta la estación base (“Redes de sensores inalámbricas”, 2014). Los elementos típicos que la conforman son:

- Microcontrolador
- Fuente de energía
- Transmisor/Receptor
- Elemento sensor

2.1.4 Gateway

Es el elemento encargado de brindar conectividad entre la red sensorial y una red TCP/IP. El protocolo inalámbrico utilizado para las redes WSN dependerá de las necesidades de la aplicación, pero entre los estándares disponibles se encuentran aquellos que se basan en el IEEE 802.15.4, IEEE 802.11, entre otros. La topología que se emplea depende de cómo se comuniquen los nodos con el Gateway. Entre las topologías disponibles para redes inalámbricas se puede identificar la topología, árbol o malla tal como se puede observar en la Figura 2.4.

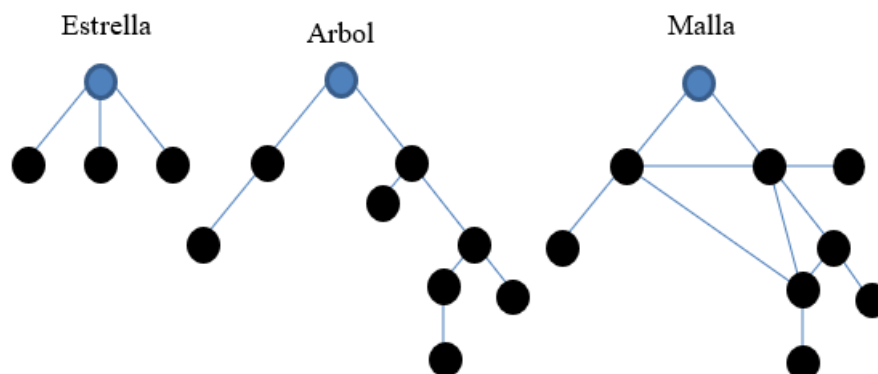


Figura 2. 4: Topologías comunes de redes inalámbricas.

Fuente:(Castaño, 2019)

2.1.5 Protocolos WSN

2.1.5.1 IEEE 802.11

Es un estándar definido para redes inalámbricas establecido por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), una institución de investigación y desarrollo ampliamente reconocida y aclamada con miembros de docenas de países, incluidos educadores y practicantes de nuevas tecnologías.(Yunquera Torres, 2005)

En sus inicios, este protocolo estaba dirigido a redes de área local inalámbricas de corto alcance (WLAN) para entornos de oficinas pequeñas y oficinas en el hogar (SOHO), pero la necesidad de realizar comunicaciones entre dispositivos de mano de alta velocidad llevó a propuestas, e incluso a implementar redes inalámbricas de mayor robustez.(Yunquera Torres, 2005)

El estándar IEEE 802.11 se divide en dos capas principales: la capa MAC (Media Access Control) y la capa física (PHY). Estos dos niveles le permiten separar la funcionalidad estándar; y, entre otras cosas, utilizar un único protocolo de datos de diferentes formas.(Yunquera Torres, 2005)

Existen diferentes variantes del protocolo IEEE 802.11 los cuales se enlistan a continuación:(Yunquera Torres, 2005)

- 802.11a
- 802.11b
- 802.11c
- 802.11d
- 802.11e
- 802.11f
- 802.11g
- 802.11h
- 802.11i
- 802.1x

2.1.5.2 IEEE 802.15.4

El estándar IEEE 802.15.4 define la capa física y el control de acceso a los medios para interfaces de bajo consumo. Desarrollado por IEEE, la

cantidad de datos por unidad de tiempo generada típicamente por dispositivos en redes de sensores. Por lo general, esto sirve para identificar transmisiones con volúmenes de datos débiles.(Alcázar, 2013)

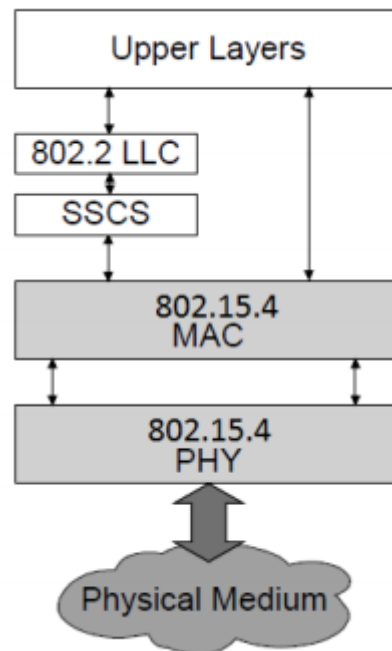


Figura 2. 5: Arquitectura de red IEEE 802.15.4.

Fuente:(Alcázar, 2013)

2.2. Sistemas domóticos.

Al hablar de un sistema domótico se refiere básicamente a una red de comunicación entre diversos elementos tecnológicos, con la finalidad de obtener información de su entorno físico con el objetivo de analizarla y realizar alguna acción.

Los elementos que pueden intervenir en este tipo de redes van desde sensores, detectores, acumuladores o captadores, hasta televisores o sistemas complejos de entretenimiento y seguridad, todos ellos administrados por un dispositivo central inteligente que se encarga de gestionar cada nodo de la red basado en configuraciones definidas con anterioridad, las cuales pueden ser generales o específicas para cada ambiente domótico.

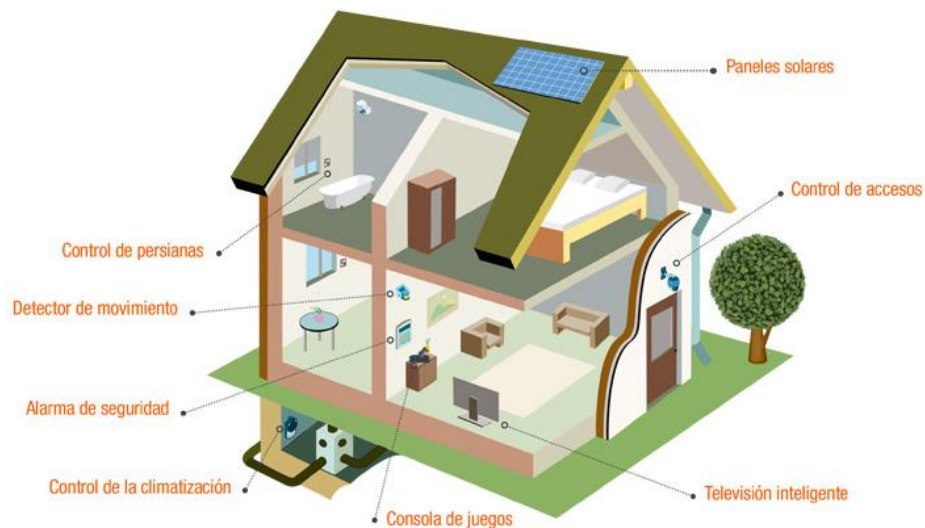


Figura 2. 6: Ambiente domótico en vivienda
Fuente: (Iberdrola, 2017)

2.2.1. Características de un sistema domótico robusto

No es ningún secreto que la implementación de sistemas domóticos en viviendas permite a las personas que la habitan gozar de una diversidad de beneficios, pero esto ocurrirá solo si el sistema cumple correctamente con todas las funciones para el cual fue diseñado. Siendo así, habría que preguntarse: ¿qué características debe tener el sistema, de manera que sea posible mejorar la vivienda y el estilo de vida de la familia que la habita, mediante la plataforma tecnológica inteligente utilizada? Por esta razón, se identifican tres características que se deben cumplir en los sistemas de automatización de hogares para ser óptimos y fiables: (All Metro Tech, 2018)

- **Infraestructura de red sólida:** Los elementos individuales de la red deben funcionar de manera óptima como la red que los contiene, si la misma no es capaz de soportar una gran cantidad de dispositivos interconectados, se producirá lentitud y afectación a los procesos automatizados.
- **Protocolos adaptados al estilo de vida de los habitantes:** Los sistemas automatizados de hogares inteligentes tienen como objetivo brindar comodidad al estilo de vida de los integrantes de la vivienda. Es por ello, que el sistema debe estar adaptado a las necesidades de cada

vivienda, puesto que un sistema generalizado podría existir huecos en los procesos necesarios y generar alertas innecesarias; por ejemplo: si la red cuenta con sensores para detectar movimientos en una habitación a ciertas horas, pero no se configura el intervalo horario, generará notificaciones constantes que podrían conllevar al colapso de la red.

- **Sistema de alimentación independiente:** El sistema domótico debe estar disponible 24/7; por lo tanto, para asegurar este requerimiento indispensable se debe considerar una fuente de alimentación independiente o por lo menos un mecanismo de backup o auxiliar, para que, ante una eventualidad en el sistema eléctrico general de la vivienda, este no detenga su funcionamiento.

2.2.2. Campos de aplicación

En la actualidad existen diversos campos de aplicación para la domótica, pero todos ellos se pueden agrupar en cinco categorías principales, las cuales se detallan a continuación:

Gestión Energética: Este campo de aplicación busca la eficiencia energética de la vivienda, o para hablar de forma más amplia, de la edificación donde está instalado el ambiente domótico. Estas acciones son posibles de realizar gracias a los nodos sensores y a los elementos reactivos para la activación o desactivación de la energía, además de la capacidad para medir los niveles energéticos utilizados para un posterior análisis.(Román, 2017)

Algunos sensores están configurados para ajustar el sistema de iluminación cuando se genera luz natural, lo cual lo hace ideal para ahorrar energía al recibir luz natural.(Zemsania, 2018)

Comunicación: Mediante este recurso nos es factible monitorear en tiempo real el estado del ambiente y no solo eso, nos hace posible el poder controlarla de manera remota, con el fin de no depender de acceso físico a la red.(Román, 2017)

Accesibilidad: Con la ayuda de este campo de aplicación, los sistemas domóticos permiten que las personas puedan acceder a lugares determinados de una manera controlada. Con esto se facilita el tránsito en los sectores del ambiente, pero no de una manera libre y desordenada, más bien con este recurso se puede conocer incluso la cantidad de personas en una habitación. Los sistemas de movimiento y biométricos juegan un papel fundamental en la explotación del mecanismo de la accesibilidad.(Román, 2017)

Confort: El confort en una edificación inteligente es quizá una de las más importantes características, además de estar intrínsecamente relacionada en todos los campos que se han revisado, y de cierta forma es el fin de la implementación de este recurso tecnológico. Desde un punto de vista aplicativo en esta categoría, la domótica se enfoca en el control de la temperatura, niveles de iluminación y control de posibles eventualidades.(Román, 2017)

Con los elementos sensoriales adecuados, es posible controlar de forma remota los termostatos en cualquier habitación. No solo puede configurar el dispositivo para establecer una temperatura específica para cada habitación, sino que también puede controlar otros factores como el aire acondicionado. Al instalar sensores que analizan la calidad del aire es posible mejorar nuestra salud. Por ejemplo, si el sensor detecta que la habitación está demasiado contaminada, se puede activar un purificador de aire o algún dispositivo que extraiga el aire impuro. (Zemsania, 2018)

Seguridad: La seguridad en la domótica se separa en varios niveles, todos conectados entre sí, y para los cuales existen estándares definidos e independientes, todo esto con el fin de proteger tanto la privacidad como la integridad de los habitantes de la estructura inteligente que la implementa. Los sensores son los encargados de monitorear las alertas suscitadas en la edificación, pero la seguridad implica el control para que esos sistemas no sean vulnerados y desactivados por agentes externos no autorizados. Los aspectos de seguridad que se deben controlar son la intrusión técnica y personal del ambiente. Los sensores proporcionan información inteligente

sobre alertas y ayudan a detectar cualquier tipo de movimiento no deseado dentro del perímetro de un edificio. Y si tiene una cámara de seguridad, puede acceder a videos y grabaciones en vivo desde su dispositivo (Zemsania, 2018).



Figura 2. 7: Servicios domésticos
Fuente: (Román, 2017)

2.2.3. IoT en la domótica

El internet de las cosas o IoT, por sus siglas en inglés, fue el nombre con el cual se denominó a la capacidad de conexión de objetos en el mundo físico con la red de redes, y esto es posible gracias al uso de sensores.

El desarrollo de los dispositivos inteligentes conectados a internet lleva un crecimiento exponencial en los últimos años. Los estudios estimaban que en el año 2020 existirían 30 billones de estos dispositivos en uso, y la tendencia apunta que para el 2024 esta cifra se duplique, esto debido al gran abanico de oportunidades que esta tecnología ofrece, tanto en el ámbito doméstico como en el empresarial.

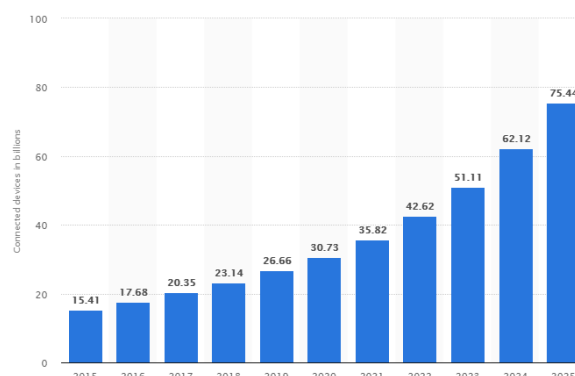


Figura 2. 8: Crecimiento de elementos inteligentes conectados a internet
Fuente: (Learte, 2018)

2.2.3.1. Uso de datos para mejorar la calidad de vida

Lo mejor de todas estas aplicaciones es que pueden extraer datos sobre su uso mediante la IoT. Gracias a ello, por ejemplo, es posible determinar cuánta luz se utiliza normalmente y de ser necesario, reducirla. Esto se aplica a todos los elementos enumerados en la sección anterior. Toda la información que se genera a diario es de gran utilidad para mejorar la calidad de vida de las personas que habitan la edificación inteligente.(Zemsania, 2018)

Otra ventaja de esta tecnología es que promueve indirectamente las ciudades inteligentes. Nuestras casas conectadas proporcionan datos que se pueden compartir con los gobiernos para analizar el consumo promedio de los residentes.(Zemsania, 2018)

Al margen de todas las ventajas expuestas, hay que tener en cuenta la importancia de planificar proyectos de domótica e IoT. A medida que integra sus aplicaciones y comienza a implementar sensores sin un objetivo específico, muchas herramientas pueden aislarse unas de otras. Esto puede hacer que sea menos conveniente y requerir más trabajo del necesario.(Zemsania, 2018)

2.2.4. Arduino en la domótica

2.2.4.1. Antecedentes

Massimo Banzi, un estudiante del Instituto IVRAE fue quien en el 2005 inventó un dispositivo al que nombró Arduino. Primero consideró hacer del Arduino un instrumento de aprendizaje para los estudiantes de informática y electrónica en el laboratorio de su instituto, debido a que conseguir una placa del microcontrolador era muy caro y se corría el riesgo de que esta no brindara la compatibilidad necesaria. Pero nunca imaginó que al poco tiempo la herramienta se convertiría en el líder mundial en tecnología "Hágalo usted mismo".(Carretero et al., 2016)

2.2.4.2. Debilidades y fortalezas de Arduino frente a otros microcontroladores

Cuando la historia vio el surgimiento de compiladores para lenguaje C en sistemas embebidos, los programadores en ASM se negaron a utilizar los nuevos lenguajes en sus microcontroladores. ¿Por qué? Probablemente debido a un nivel de optimización más bajo que en Ensamblador, o tal vez por la desconfianza en el nuevo sistema, o las presunciones en la falta de seguridad de este lenguaje, incluso pudo deberse a la negativa innata del hombre al cambio. Todas las razones anteriormente nombradas contienen parte de certezas, pero los programadores actuales no pueden discutir o negar el impacto positivo del lenguaje C para el desarrollo de la tecnología actual.(Carretero et al., 2016)

Debilidades

Al aprender a cambiar los registros internos y programar el microcontrolador utilizando C, se permite a los usuarios una mejor comprensión del hardware de los dispositivos, otorgando flexibilidad y optimización. Esto es ideal si su programa quiere identificar pequeños parámetros que por sus características pueden ser muy importantes para algunas aplicaciones. Métodos de cálculo como tiempo muerto, retraso de precisión y uso de memoria son algunos de ellos.(Carretero et al., 2016)

Al utilizar microcontroladores en lugar de tarjetas de desarrollo, le permite al programador desarrollar los conocimientos relacionados a los aspectos analógicos de la electrónica, otorgando una ventaja al momento de crear tarjetas con objetivos específicos.(Carretero et al., 2016)

Los microcontroladores proporcionan más flexibilidad en proyectos específicos. Es posible escoger de una gama mucho más amplia y superior de dispositivos que las utilizables con placas Arduino. La configuración común del dispositivo incluye operación a frecuencias más altas, soporte para temperaturas más elevadas, disponibilidad de múltiples salidas PWM y protocolo CAN, etc.(Carretero et al., 2016)



Figura 2. 9: Microcontroladores
Fuente: (Fénix, 2014)

Haciendo a un lado el Hardware, los costos juegan un papel muy importante en proyectos donde se necesita implementar software para sistemas que pueden escalar a más de 1,000 piezas, en estos casos se puede observar que comprar varias tarjetas Arduino no es tan conveniente en precio ni dimensión. Sin embargo, si es posible dentro del límite de tiempo, simplemente cargue el programa que se encuentra en la placa en un microchip puro.(Carretero et al., 2016)

Fortalezas

Posee una curva de aprendizaje bastante rápida. Permitiendo optimizar el tiempo en los proyectos, contemplando que algunos necesitan un nivel de investigación amplio. Es una herramienta altamente funcional.(Carretero et al., 2016)

Su inmensa comunidad con visión de código abierto es tal vez una de sus grandes fortalezas a lo largo de su vida. Sus premisas se basan en la pregunta “¿por qué haces lo que otros ya hicieron?” además de contar con el lema: "Solos vamos más rápido, pero juntos vamos más allá".(Carretero et al., 2016)



Figura 2. 10: El universo Arduino
Fuente:(jecrespom, 2016)

Posee un entorno de desarrollo minimalista el cual es de alto agrado en el mundo de la programación, y si este es complementado por la facilidad que da el poder escribir código y compilar en una sola herramienta, lo cual difiere con la mayoría, por no decir todas, las aplicaciones orientadas a codificar para microcontroladores. Esta última característica se ve potenciada por sus recientes servicios de editor en línea.

The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. The code is as follows:

```

Blink | Arduino 1.6.5
File Edit Sketch Tools Help
Blink
the documentation at http://www.arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}

```

Figura 2. 11: Entorno de desarrollo local
Fuente: (filehorse, 2021)

En la siguiente figura podemos observar el ambiente de desarrollo cloud, el cual adapta el mismo estilo del entorno local.

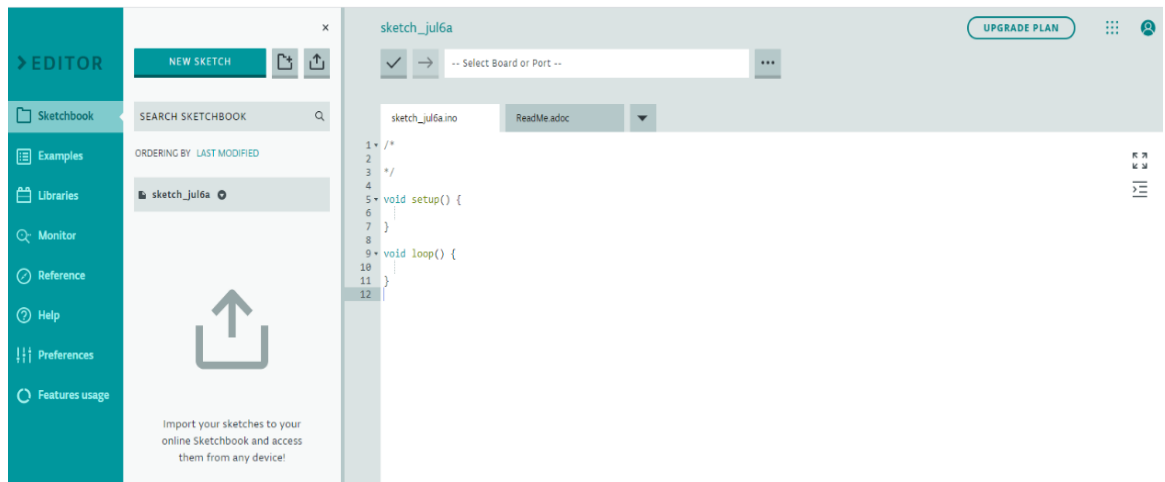


Figura 2. 12: Entorno de desarrollo cloud
Fuente: (Arduino Inc., 2021)

Sin temor a equivocación el servicio cloud de Arduino es en la actualidad su mayor fortaleza y lo que le permite ubicarse como uno de los dispositivos preferidos en los proyectos de electrónica en general, y aunque ya se lo mencionó brevemente en el párrafo anterior, es necesario ahondar en este servicio.

2.2.4.3. Arduino IoT Cloud

Al hablar de Arduino IoT Cloud se hace referencia a una plataforma que integra, la administración de conexiones, la programación de dispositivos y un gestor de aplicaciones orientadas al mundo del internet de las cosas, todo esto en un entorno de fácil uso y de amplia documentación.(Arduino Pro, 2021)

Ofrece opciones de conectividad integradas como WiFi, Ethernet, LoRa®, Sigfox, IoT móvil (LTE CatM y NBloT) y dispositivos de rápida personalización utilizando una variedad de placas de módulo Arduino, el cual contiene un microprocesador ARM de 32 bits y cifrado integrado Generate-Authentication.(Arduino Pro, 2021)

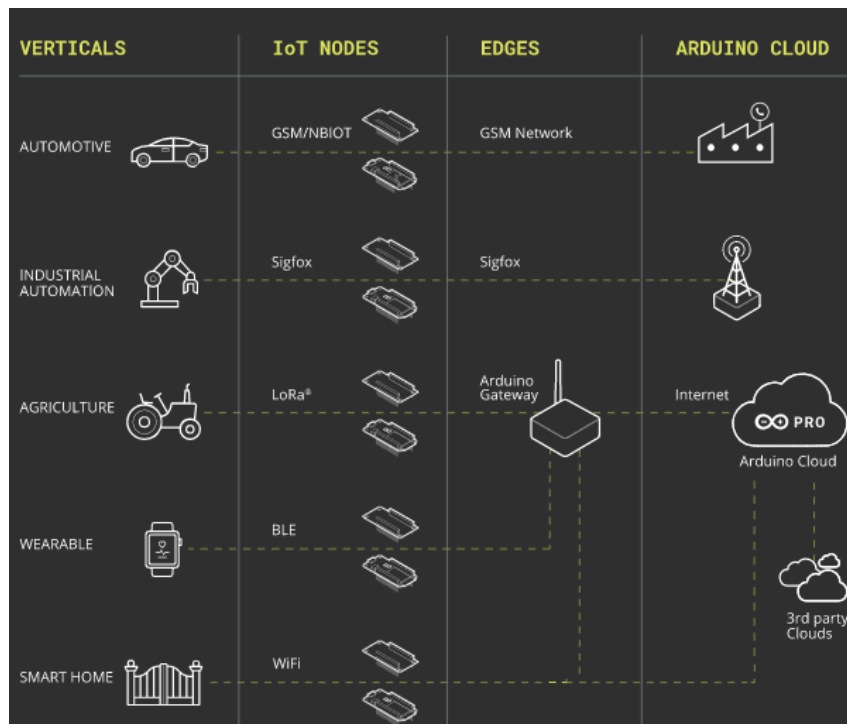


Figura 2. 13: Opciones de conectividad disponibles
Fuente: (Arduino Pro, 2021)

2.2.4.4. Esquema general para proyectos de IoT con Arduino

Antes de sumergirse en las diversas plataformas, es interesante comprender la idea global de cómo debería ser un sistema basado en arquitectura cloud. Hay tres componentes principales involucrados en este sistema: *(Proyectos IoT con Arduino, análisis de las plataformas más importantes, 2016)*

- El dispositivo conectado o del IoT.
- La plataforma en la nube.
- Los dispositivos que consumen la información en la plataforma del IoT.

Capítulo 3: Diseño y Análisis de Resultados

3.1. Diseño de la red de sensores

En una red domótica pueden intervenir un gran número de dispositivos, para monitoreo y alerta ante emergencias, los elementos utilizados en este trabajo permiten identificar eventos como: la detección de gases peligrosos, medición de altas temperaturas y monitorización de movimientos que se consideran inusuales.

3.1.1 Diagrama de bloques de la red

En la Figura 3.8 se muestra el diagrama de bloques de la red de la conexión física entre el Arduino y el MCU, que cumple las funciones de AP, además se visualiza los 3 nodos sensores que se utilizan para tomar la información que se envía a los servicios cloud de Arduino.

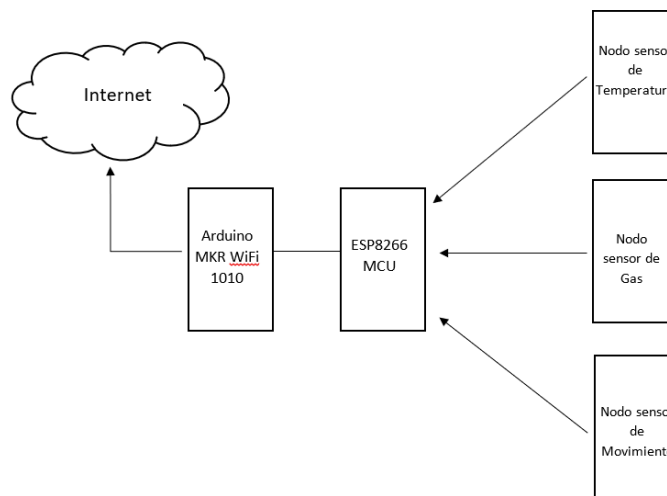


Figura 3. 1: Diagrama de bloques WSN

Fuente: Elaborado por el autor

3.1.2 Arduino MKR WiFi 1010

Este dispositivo es el punto de entrada más fácil para diseñar aplicaciones de redes-pico e IoT, ya sea que se requiera construir una red de sensores conectados a una oficina, un enrutador doméstico o si desea construir un dispositivo BLE (Bluetooth de baja energía) que envíe datos a su teléfono móvil, este elemento es una solución completa para muchos sistemas IoT básicos.

El procesador principal de la placa, al igual que otras placas de la familia Arduino MKR, es Arm Cortex M0SAMd21 de 32 bits y bajo consumo. La conectividad WiFi y Bluetooth se logra mediante el uso del módulo ublox NINA W10, un chipset también de bajo consumo que opera en el rango de 2 GHz. Además, incorpora el chip de encriptación Microchip ECC508 que garantiza una comunicación segura.



Figura 3. 2: Arduino MKR WiFi 1010
Fuente: (Arduino MKR WiFi 1010 | Arduino Official Store, 2021)

3.1.3 Esquema Arduino MKR WiFi 1010

Electrónicamente hablando el Arduino brinda un gran número de posibilidades para ser utilizado, en este proyecto no se hace uso de todas estas opciones, solamente de recursos básicos, uno de ellos es el uso de un pin analógico, mediante el cual se conecta al AP (Access Point), dado que este último se encarga de enviar las lecturas recibidas desde los nodos sensores hacia el Arduino, el cual procede a la identificación y enrutamiento hacia los servicios cloud configurados.

A continuación, se presenta el diagrama electrónico completo de este dispositivo:

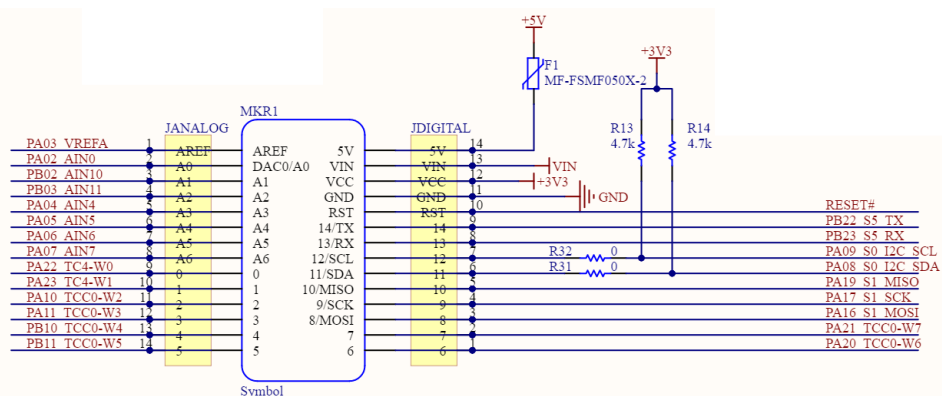


Figura 3. 3: Esquema electrónico Arduino MKR WiFi 1010
Fuente: (Arduino MKR WiFi 1010 | Arduino Official Store, 2021)

Los módulos integrados poseen su propio diagrama electrónico, y aunque no interactúan de manera individual en la red, es importante conocer su estructura, la cual se observa en la Figuras 3.3.

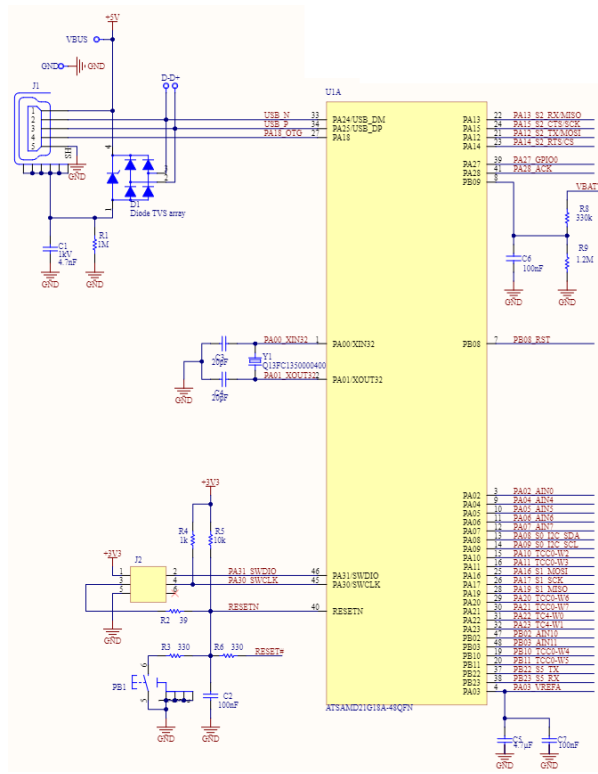


Figura 3. 4: Esquema electrónico ATSAMD21G18A-48QFN
Fuente: (Arduino MKR WiFi 1010 | Arduino Official Store, 2021)

La Figura 3.5 nos muestra el esquema electrónico del módulo Ublox NINA, el cual es el encargado de gestionar la comunicación Wireless.

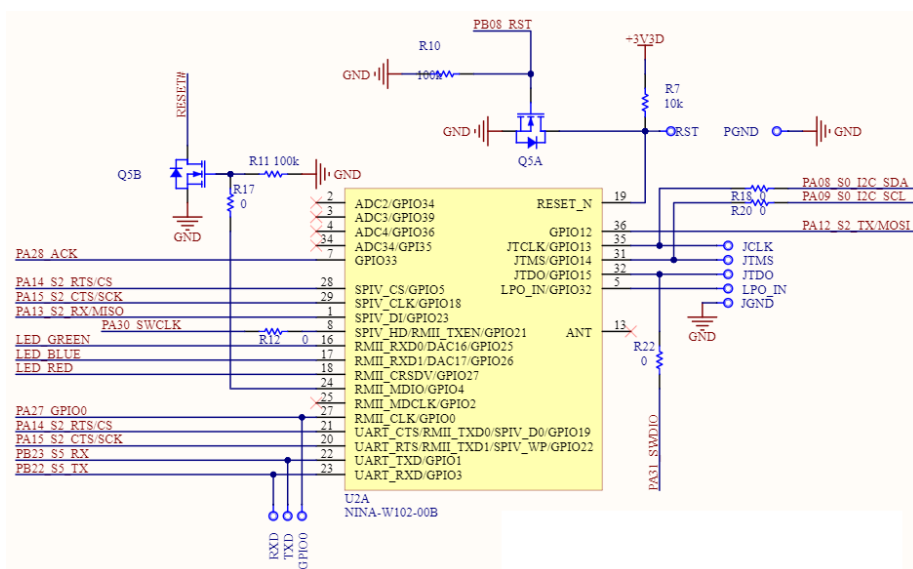


Figura 3. 5: Esquema electrónico módulo NINA-W102-00B
Fuente: (Arduino MKR WiFi 1010 | Arduino Official Store, 2021)

3.1.4 Módulo ESP 8266 MCU

El ESP8266 MCU dadas sus características y modos de funcionamiento los cuales se observaron en el capítulo anterior, sirve de puente para la comunicación entre el Arduino y la red de sensores, además de utilizarse para la transmisión de la información entre el elemento sensor con el dispositivo que realiza las funciones de AP.

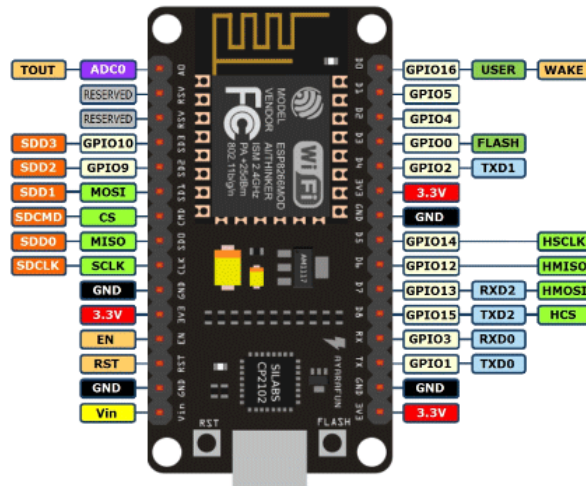


Figura 3. 6: Diagrama de pines de Módulo ESP8266 MCU
Fuente: (programarfacil, 2017)

De este módulo se utilizan 3 pines digitales y 1 analógico, los primeros sirven para notificar al módulo Arduino a qué sensor pertenece la lectura del pin analógico. Además, cabe mencionar que el MCU debe ser alimentado con 3.3V, pues de hacerlo con un voltaje mayor a este, se quema.

3.1.5 Sensores

Los dispositivos que se utilizan como Gateway o motas son capaces de interactuar con cualquier tipo de sensores, pero considerando los parámetros que se desean en este proyecto se consideraron los siguientes sensores:

MQ-2

Se utiliza este sensor para realizar la medición de la concentración de gas en el aire debido a que entre sus características destaca el ser un elemento de buena sensibilidad y perfectamente adaptable al módulo MCU, brindando una respuesta analógica fiable. Además de tener la ventaja de ser económico, lo que eleva aún más su valoración para su uso en proyectos de mediana escala.



Figura 3. 7: Módulo MQ-2
Fuente: (naylampmechatronics, 2016)

DHT11

Este módulo se utiliza para la medición de humedad en el ambiente, pero principalmente para la temperatura, gracias a su practicidad, amplia documentación y costo, es un elemento ideal para proyectos de gama media. Posee un rango de temperatura que oscila entre 0° y 50° entre lo cual solo se consideran valores enteros. Su rango de medición de humedad va del 20% al 80%. Este elemento sensor nos permite realizar una lectura cada segundo, lo cual no representa mayor inconveniente para el proyecto planteado.

Se puede encontrar 2 tipos diferentes de sensores, uno con 3 pines y otro con 4, no obstante, en este proyecto solo se utilizan 3 pines que corresponden a los pines de Vcc que opera a 3.3V, GND y Data, el cual nos sirve para la comunicación con el módulo MCU.

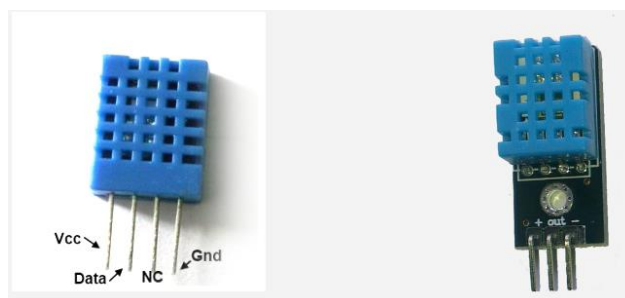


Figura 3. 8: Módulo MQ-2
Fuente: (designthemes, 2018)

3.2. Montaje y configuración de la red

Para un correcto funcionamiento de la red es necesario realizar configuraciones y conexiones específicas para los elementos que la integran.

Por ello, a continuación, describimos como se deben configurar dichos elementos, incluyendo los firmwares que se utilizan.

3.2.1 Conexión de los elementos

En la siguiente figura se puede observar la conexión entre el MCU que cumple la función de AP y el Arduino MKR WiFi 1010 que cumple las funciones de Gateway y se interconecta con los servicios cloud.

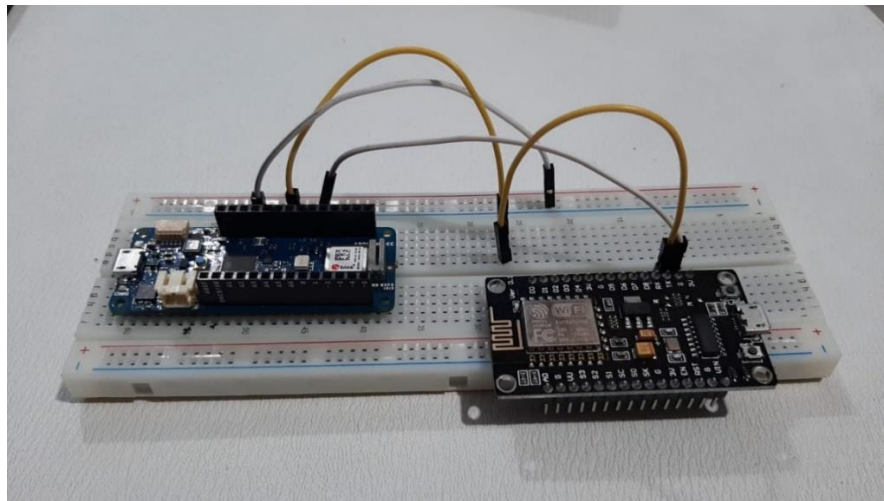


Figura 3. 9: Conexión MCU con Arduino 1
Fuente: Elaborado por el autor

En la Figura 3.10 se observa la conexión del módulo desde un ángulo superior, con lo cual es posible observar la conexión de cara pin del MQ con sus respectivos pares en el módulo MCU.

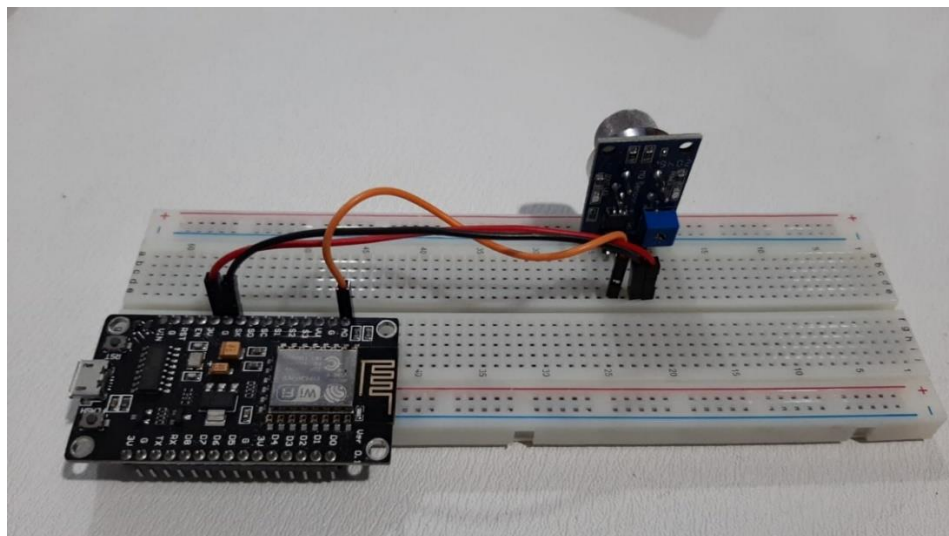


Figura 3. 10: Conexión MCU con MQ-2 1
Fuente: Elaborado por el autor

En la Figura 3.11 se observa la conexión del módulo desde un ángulo superior, con lo cual es posible observar la conexión de cara pin del MQ con sus respectivos pares en el módulo MCU.

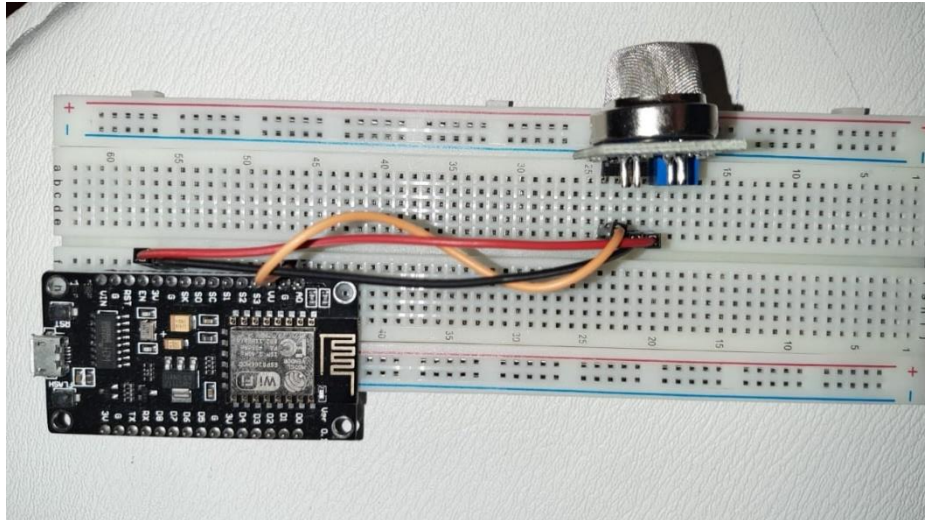


Figura 3. 11: Conexión MCU con MQ-2 3
Fuente: Elaborado por el autor

3.2.2 Configuración del entorno de desarrollo

Se utilizan 2 tipos de entornos, el local y el cloud, ambos se relacionan mediante el uso del Arduino MKR WiFi 1010.

El entorno local, es el encargado de la interacción de los elementos en la red, y mediante este se crea el firmware que se carga a los elementos encargados de obtener, procesar y transmitir la información de los sensores hasta llegar al nodo controlador, el cual tiene la responsabilidad de comunicar los datos obtenidos hacia el entorno cloud. El entorno cloud a su vez, es el responsable de recibir los datos enviados por el entorno local, con el objetivo de permitir que el usuario final pueda observarlos a manera de información en una interfaz amigable y precisa, con indicadores seleccionados y configurados en la fase de desarrollo.

3.2.3 Entorno local

Como entorno de desarrollo local se utiliza el Arduino IDE en su versión 1.8.15 debido a las ventajas que nos ofrece respecto a otros softwares de la misma rama; estas ventajas se explicaron en el capítulo 2. Al instalar el software y verificar entre las tarjetas habilitadas, se observa que no contiene

la librería necesaria para trabajar con el Arduino MKR WIFI 1010, por esta razón se detallan los pasos para agregar este modelo de placa:

1. Ingresar al “Gestor de tarjetas”, que se encuentra en el menú “Herramientas” en la opción “Placa”.

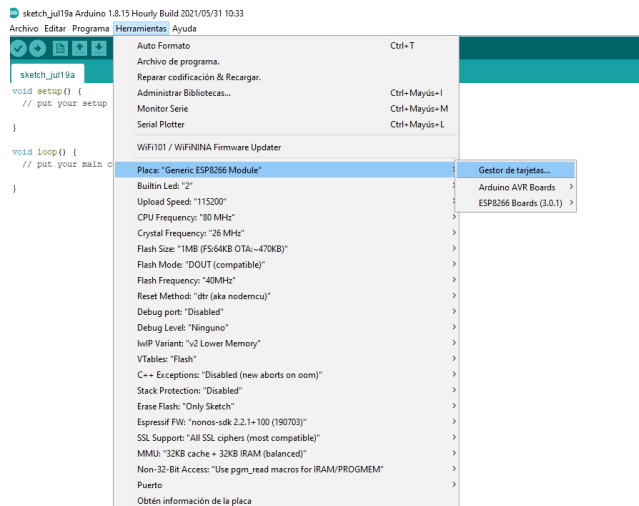


Figura 3. 12: Ingreso al Gestor de Tarjetas

Fuente: Elaborado por el autor

2. Se abre una ventana modal en la cual se buscan los complementos que se desean agregar, en nuestro caso se escribe “SAMD Board” en el buscador y se selecciona aquel que nos haga referencia al Arduino MKR WiFi 1010. Posterior a ello se presiona el botón “Instalar”. Esto se muestra en la 3.13.

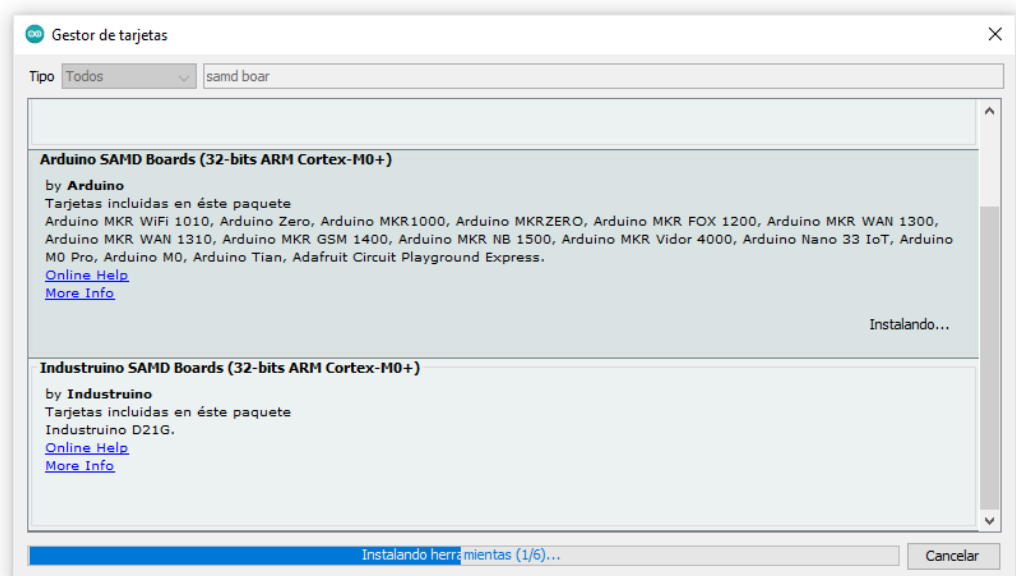


Figura 3. 13: Instalación módulo Arduino MKR WiFi 1010

Fuente: Elaborado por el autor

Cuando el proceso de instalación esté finalizado se observa que se habilita la tarjeta para su uso.

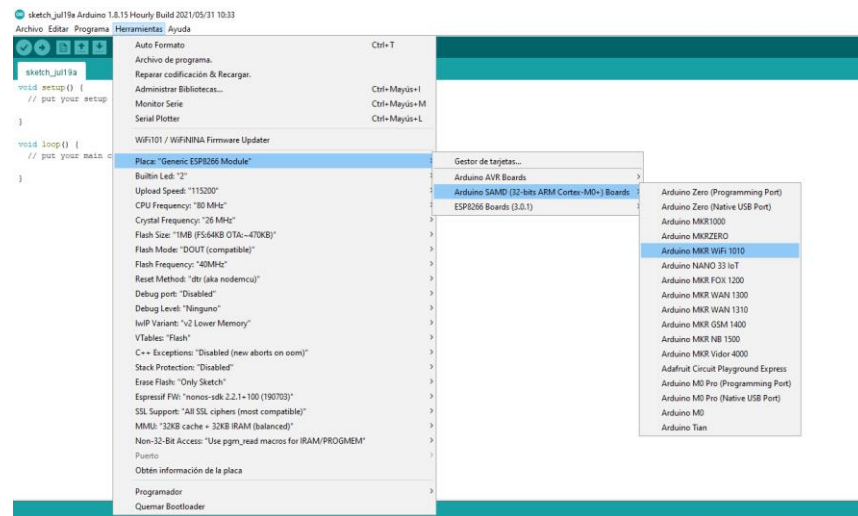


Figura 3. 14: Arduino MKR WiFi 1010 disponible.
Fuente: Elaborado por el autor

Para el correcto uso de los recursos de conectividad, es necesario agregar la librería **WiFiNINA** debido a su compatibilidad con el modelo de la ublox del Arduino que se utiliza. Para ello es necesario realizar el siguiente proceso:

1. Abrir el administrador de biblioteca, que se encuentra en el menú "Programa", en la opción incluir librería.

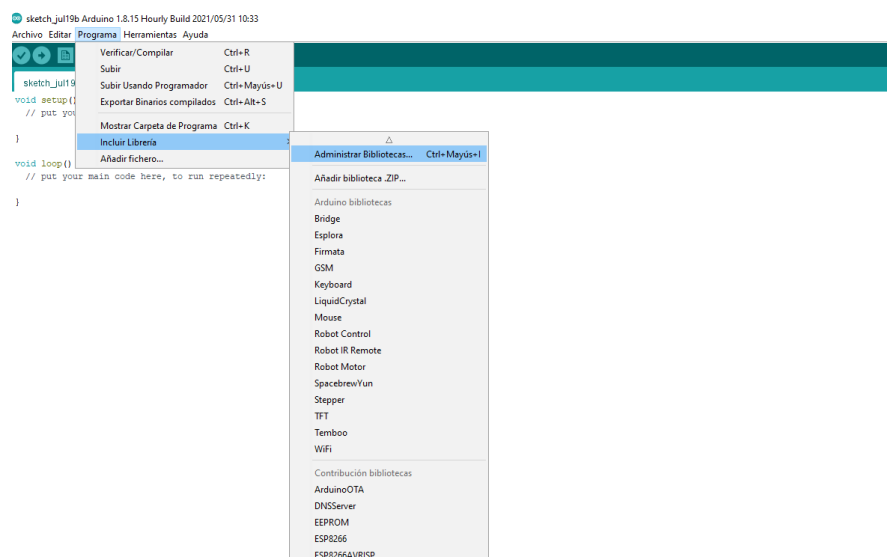


Figura 3. 15: Abrir Gestor de Librerías
Fuente: Elaborado por el autor

2. Se encuentra la librería utilizando el buscador que brinda el gestor, y una vez encontrada, se procede con la instalación de esta.

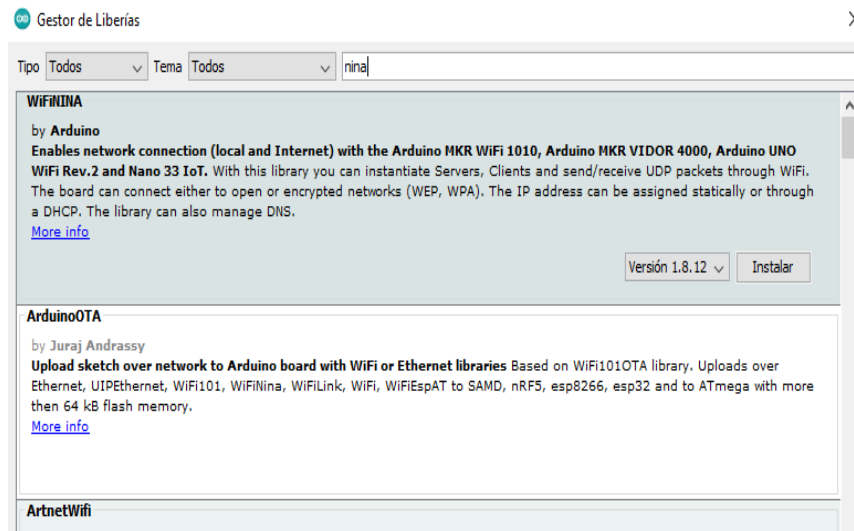


Figura 3. 16: Gestor de Librerías
Fuente: Elaborado por el autor

Una vez que se agregaron las librerías necesarias para el correcto funcionamiento de la tarjeta Arduino, es necesario incorporar las librerías que se utilizan para la toma de datos desde los sensores, las mismas son detalladas a continuación:

DHT sensor library, es la librería encargada de la administración, del módulo de temperatura y humedad del ambiente, esta librería facilita la interacción de la mota con este sensor. Para una mejor referencia, en la siguiente figura observa la librería desde la pantalla del gestor.

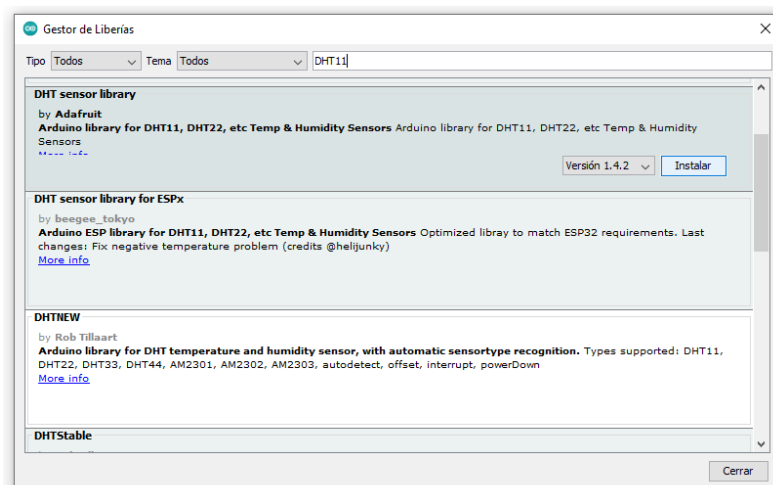


Figura 3. 17: Instalación de librería DHT sensor library
Fuente: Elaborado por el autor

3.2.4 Entorno cloud

En nuestro navegador preferido se ingresa a la ruta <https://www.arduino.cc/>, la cual corresponde a la página principal de Arduino.

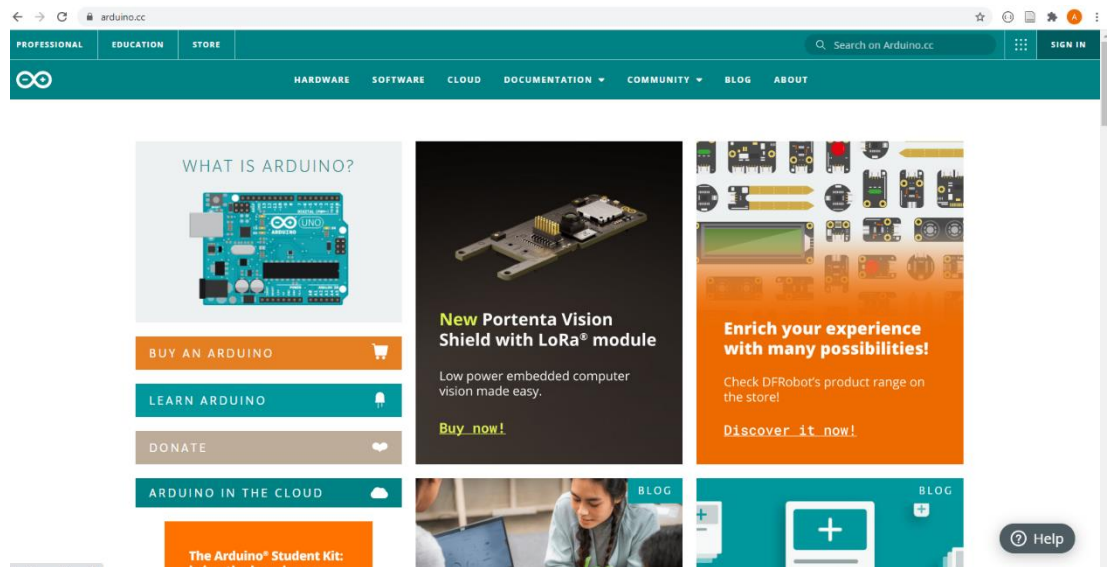


Figura 3. 18: Arduino Home Page
Fuente: Elaborado por el autor

Al darle clic en la opción “Sign in” que se encuentra en la parte superior derecha, la página direcciona al formulario para el ingreso a la plataforma cloud, es en este lugar donde se debe ingresar el usuario y la clave personal.

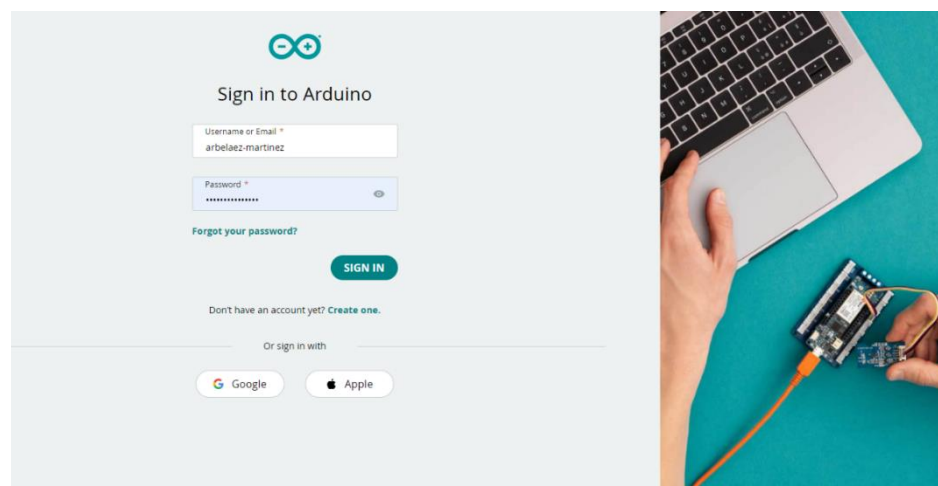


Figura 3. 19: Sign in Page
Fuente: Elaborado por el autor

Una vez se ingresen las credenciales y estas sean validadas, se accede a la página de inicio de nuestra cuenta, para desde allí, ingresar al

entorno IoT. Para ello se hace clic sobre el ícono punteado que se encuentra en la parte superior derecha, posterior a ello se elige la opción “IoT Cloud”.

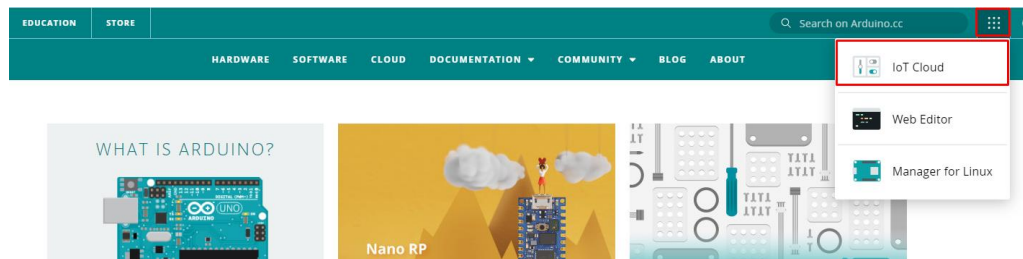


Figura 3. 20: Ingreso al entorno IoT Cloud
Fuente: Elaborado por el autor

Al realizar las opciones antes mencionadas finalmente se puede visualizar el entorno en el que se realizan todas las configuraciones necesarias para la integración de la red con los servicios cloud. Para esto, es necesario registrar el dispositivo en la plataforma. Esto es útil también si se desea hacer uso del IDE Online.

En el proceso de instalación es necesario especificar el modelo y asignarle un nombre; cuando este proceso está en marcha se requiere que el dispositivo se encuentre conectado a la misma computadora en la que se está realizando la configuración. Este proceso se detalla a continuación:

De entre las opciones disponibles en nuestro entorno IoT Cloud, se escoge la opción “Devices” y posterior a esto se debe hacer clic en el botón “ADD DEVICE”, con lo cual se mostrará un modal, mismo que es posible observar en la siguiente figura.

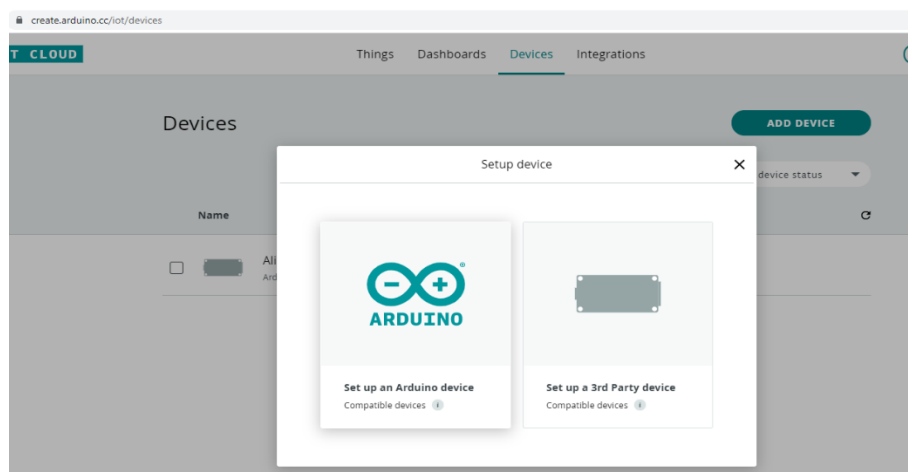


Figura 3. 21: Add Device Modal
Fuente: Elaborado por el autor

Si el dispositivo se encuentra conectado a la computadora en la que se está realizando el proceso de registro, aparecerá un mensaje como el que se observa en la Figura 3.22. Este mensaje nos denota que la búsqueda finalizó con éxito y que el proceso de configuración empieza.

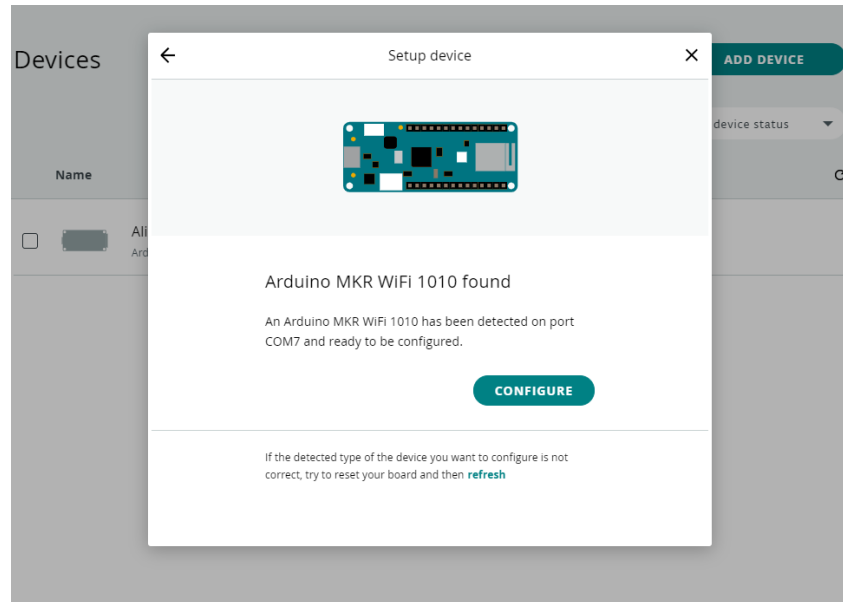


Figura 3. 22: Arduino MKR WiFi 1010 Encontrado
Fuente: Elaborado por el autor

Posterior a darle clic en el botón “CONFIGURE” el proceso toma aproximadamente 5 minutos, durante los cuales es posible observar el progreso de este.

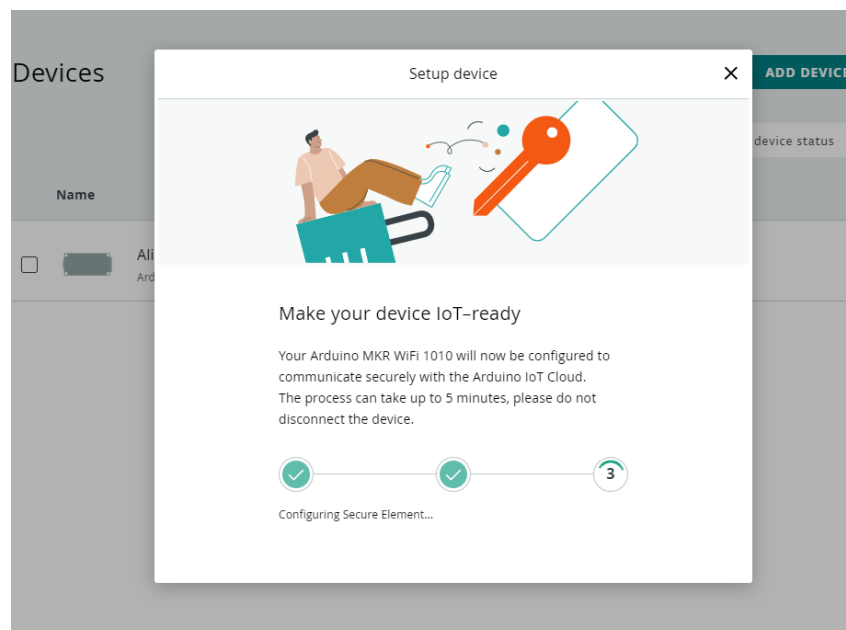


Figura 3. 23: Configuración inicial del dispositivo en la plataforma cloud.
Fuente: Elaborado por el autor

Cuando el proceso de configuración inicial finalice, se muestra la opción para asignar un nombre a nuestro dispositivo.

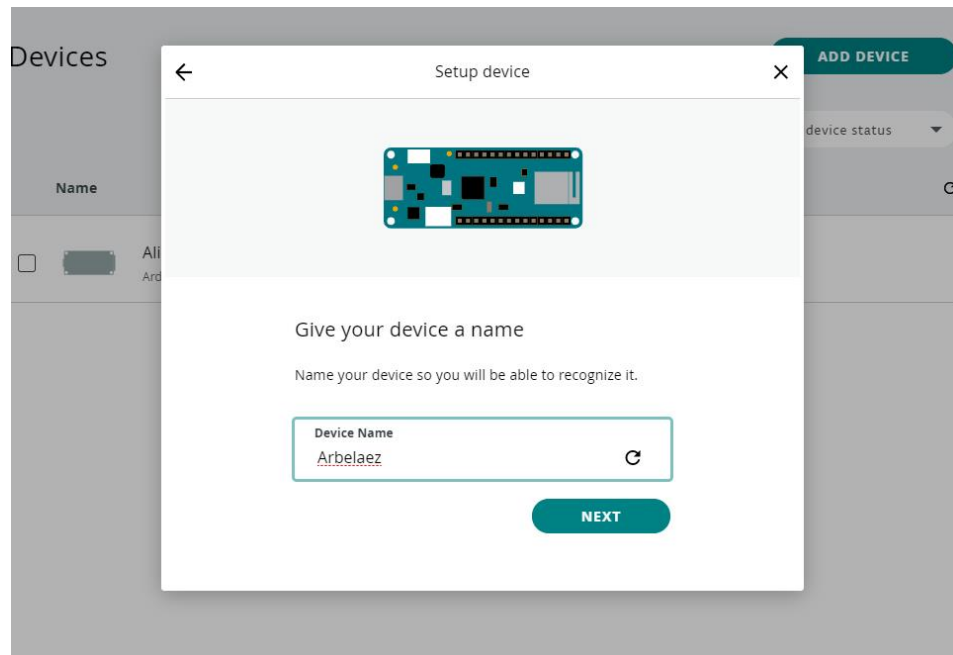


Figura 3. 24: Asignación de nombre al dispositivo
Fuente: Elaborado por el autor

Al finalizar estos pasos, el dispositivo se encuentra configurado y listo para ser utilizado en el proyecto.

Posterior a la configuración del dispositivo se crea la “thing”. Este objeto se lo puede considerar como el eje principal del proyecto en el entorno cloud; puesto que, es en él que se relacionan las variables con los dispositivos y en donde se puede hacer uso del editor Online para el proyecto. En la Figura 3.25 se puede observar el entorno inicial de las “things”;

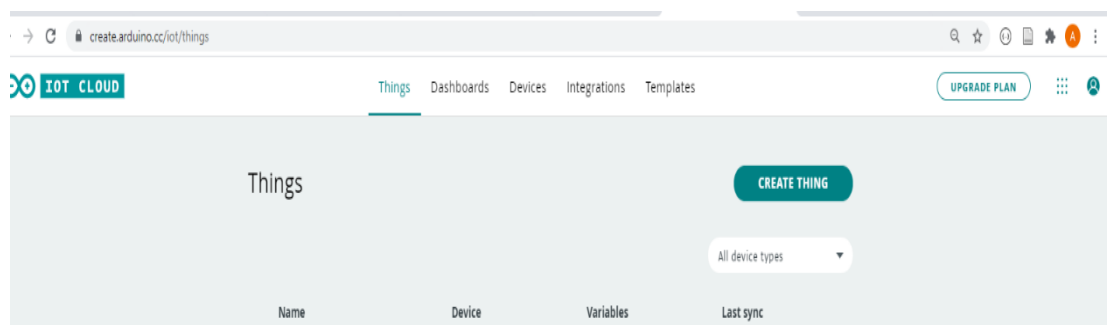


Figura 3. 25: Entorno inicial things
Fuente: Elaborado por el autor

Para poder crear la thing del proyecto, se procede a dar clic en el botón “CREATE THING” que se encuentra en la parte central derecha de la página.

Una vez realizado este paso, se crea la “thing” y se ingresa a la misma, para realizar las configuraciones requeridas.

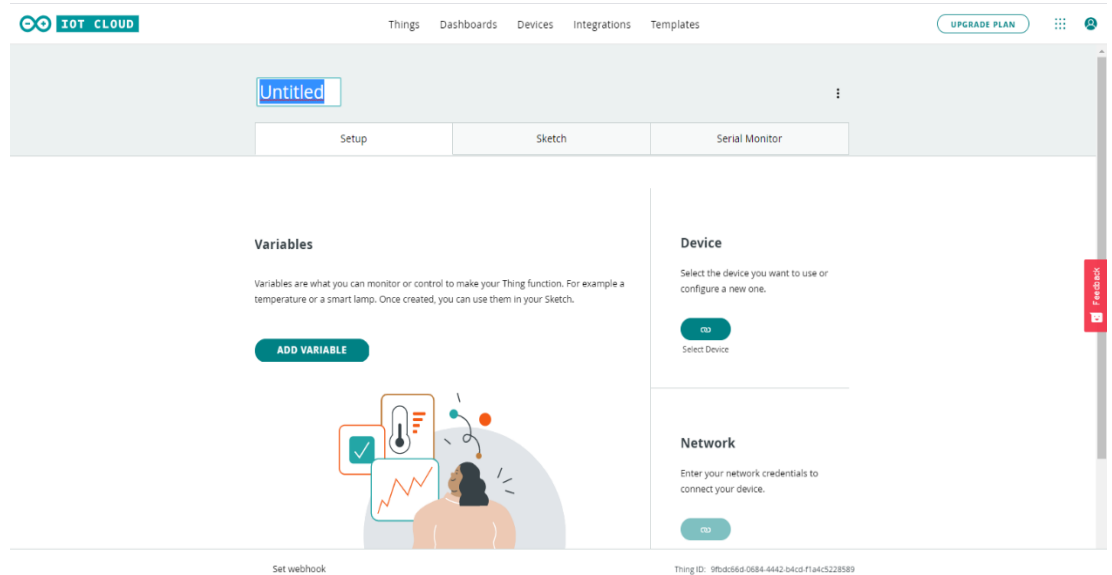


Figura 3. 26: Thing creada
Fuente: Elaborado por el autor

Para poder cambiar el nombre, basta con colocarse en la palabra “Untitled” que se encuentra en la parte superior izquierda y escribir el nombre deseado, para este proyecto se coloca “Tesis Thing”. Esto se puede apreciar en la Figura 3.27 respectivamente.

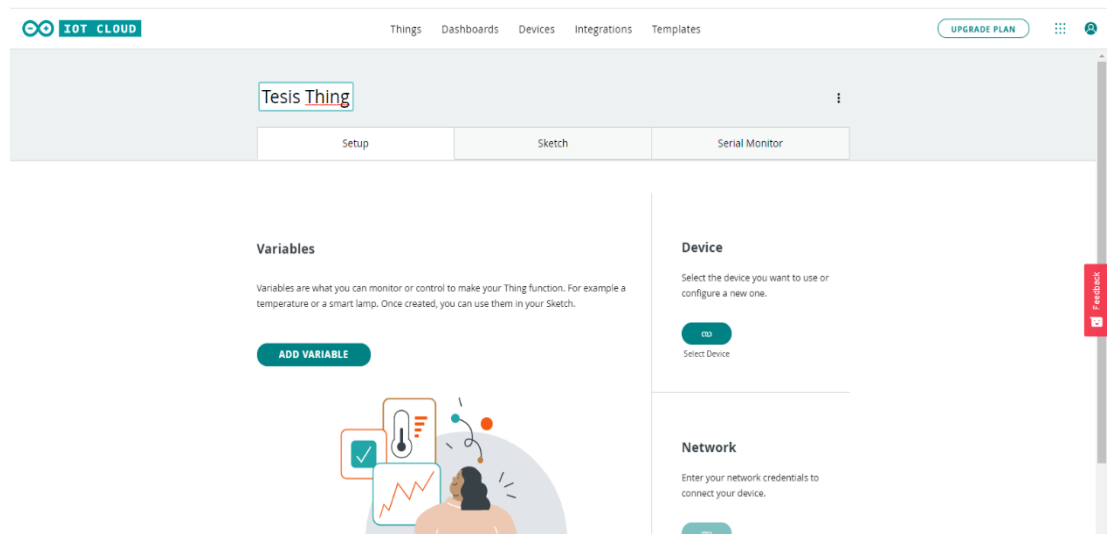


Figura 3. 27: Renombre de thing
Fuente: Elaborado por el autor

En la parte derecha se muestra la opción para vincular la “thing” con un dispositivo, esto se realiza, dando clic en la opción “Select Device”.

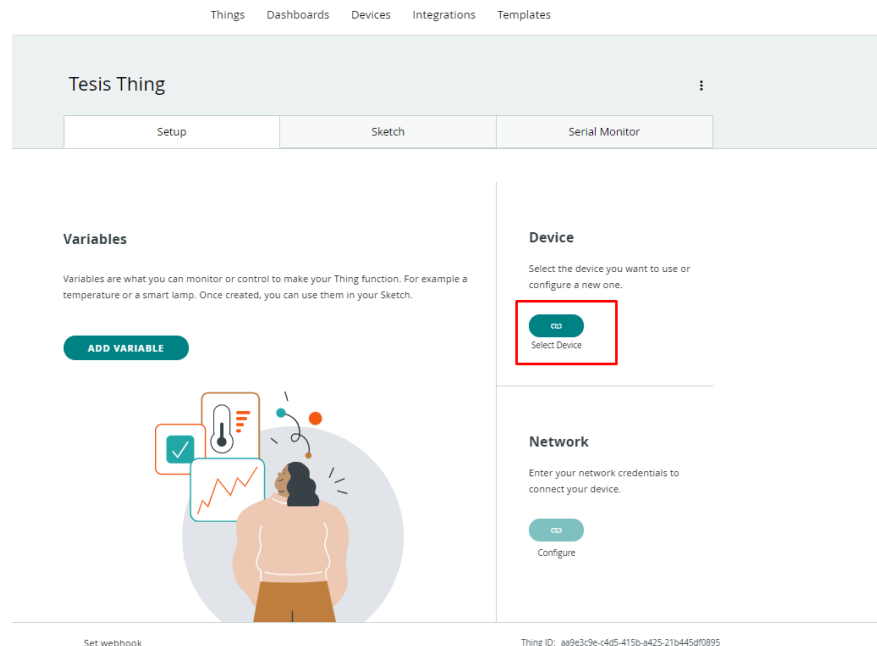


Figura 3. 28: Opción para el registro de dispositivo
Fuente: Elaborado por el autor

Posterior a ello, se abre un modal en el que aparece el dispositivo agregado anteriormente, el cual se selecciona para que se vincule a nuestro objeto.

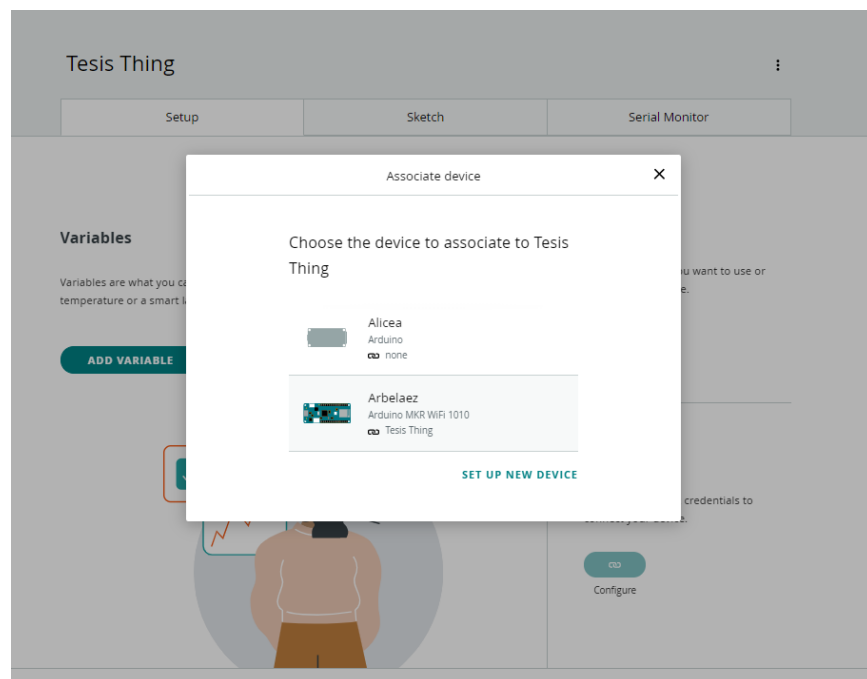


Figura 3. 29: Modal para registro de dispositivo
Fuente: Elaborado por el autor

Luego de finalizar el registro del dispositivo, el mismo se puede observar en la parte derecha de la pantalla.

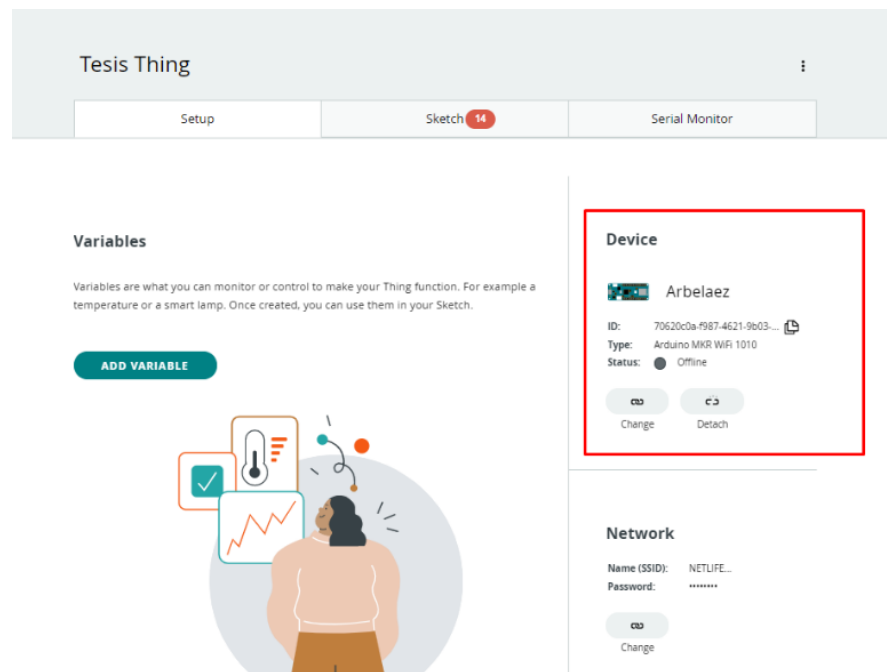


Figura 3. 30: Dispositivo creado
Fuente: Elaborado por el autor

Una vez que la “thing” se generó el siguiente paso es crear las variables que se utilizan en el proyecto. Estas variables, en este caso, corresponden a los valores entregados por los sensores y que posteriormente se los utiliza en los indicadores del dashboard de monitoreo.

La primera variable que se agrega corresponde a la temperatura. Esto se hace dando clic en el botón “ADD VARIABLE”. Al hacer esto se abre un modal en el cual se ingresa la información necesaria para la creación de la variable, los parámetros utilizados se pueden observar en la Figura 3.31.

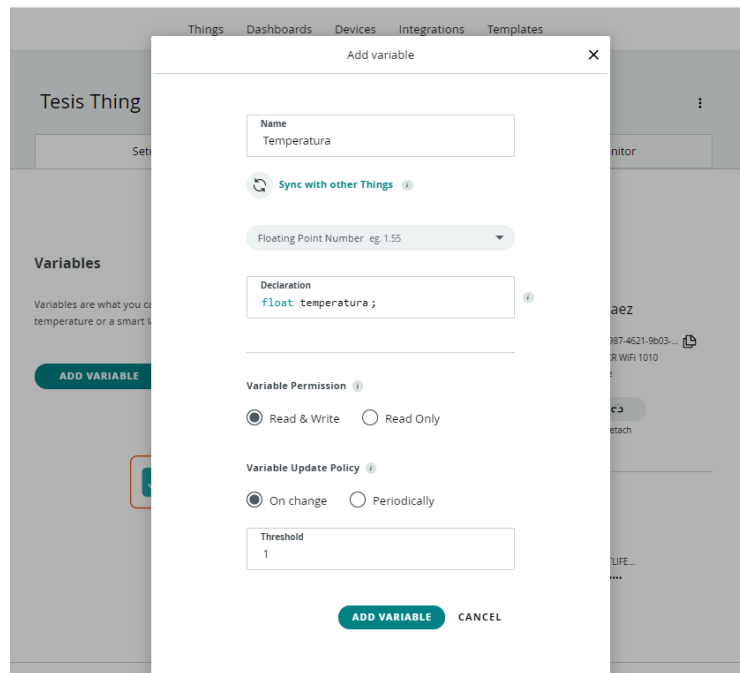


Figura 3. 31: Creación de variable
Fuente: Elaborado por el autor

3.3. Programación de la red.

Para el correcto funcionamiento de los elementos de la red, es necesario generar el firmware del punto de acceso, Gateway y nodos sensores.

3.3.1. Punto de Acceso

El elemento que realiza las funciones de Access Point es el módulo MCU en modo AP, esto con el fin de crear una red de área local inalámbrica que permita la conexión de las motas, las cuales se encargan de transmitir la lectura del sensor y cuyo funcionamiento y programación se verá más adelante. Al nodo AP corresponde la recepción de la señal para una posterior comunicación con el Arduino y que este pueda transmitir al servicio cloud.

Posteriormente en el método *“configureAccesspoint”* se inicializa el modo AP con los parámetros antes señalados. El mismo que es utilizado en el método *“setup”*, el cual se verá más adelante.

En la Figura 3.32 se observa, la declaración de las variables que corresponden al nombre y contraseña de la subred creada por el AP. Cabe mencionar que por practicidad se define una ip estática a la cual se conectan todas las motas con la finalidad de fijar el punto de acceso al cual se

transmiten los datos de cada sensor. El puerto seleccionado para acceder a este AP es el puerto 80.

```
const char *ssid = "My-SSID";  
const char *password = "@Abc112358";  
  
ESP8266WebServer server(80);  
  
IPAddress ip(192, 168, 1, 200);  
IPAddress gateway(192, 168, 1, 1);  
IPAddress subnet(255, 255, 255, 0);
```

Figura 3. 32: Configuración modo Access Point
Fuente: Elaborado por el autor

Luego de realizar estos pasos se procede a cargar el programa en el MCU. Al terminar de cargar el firmware en el dispositivo, es posible utilizar nuestro teléfono celular para comprobar la disponibilidad de la red. Esto se puede observar en la Figura 3.33.



Figura 3. 33: Visualización de la red generada
Fuente: Elaborado por el autor

El firmware hace uso de 2 métodos principales, uno de configuración y otro de inicialización. El primero inicializa la comunicación serial, el cual se

utiliza para la comunicación con el Arduino, el loop se encarga de esperar hasta que la comunicación se encuentre habilitada.

El segundo método se encarga de configurar el MCU con los parámetros de ip, Gateway y subnet, definidos anteriormente, además de asignarle nombre y clave a nuestra red. Para la comunicación con el AP se habilitan 2 endpoints, ambos a la dirección root, pero con diferentes métodos HTTP. La definición de estos métodos se realiza más adelante.

```
void initSerialPort() {  
  Serial.begin(9600);  
  while (!Serial);  
}  
  
void configurateAccesspoint() {  
  WiFi.softAPConfig(ip, gateway, subnet);  
  WiFi.softAP(ssid, password);  
  
  server.on("/", HTTP_GET, handleRootResponse);  
  server.on("/", HTTP_POST, sensorRequest);  
  server.begin();  
}
```

Figura 3. 34: Métodos de configuración

Fuente: Elaborado por el autor

El método “*handleRootResponse*” genera una respuesta HTTP para confirmar que el AP se encuentra escuchando los request de los clientes conectados. Esta definición se visualiza en la Figura 3.35.

```
void handleRootResponse() {  
  server.send(200, "text/html", "<h1>You are connected</h1>");  
}
```

Figura 3. 35: Definición método *handleRootResponse*

Fuente: Elaborado por el autor

En el metodo “*sensorRequest*” se verifica que la consulta realizada por un cliente traiga los parámetros necesarios, como lo son el “id” que identifica el nodo sensor y el “value”, que es el valor entregado por el sensor. Si estos parámetros no existen se responde un código HTTP 404 y se sale del proceso, pero si los parámetros recibidos son correctos se envía por comunicación serial la trama del identificador del nodo y el valor recibido separados por un “;”. Las líneas de código pueden ser observadas en la Figura 3.36.

```

void sensorRequest() {

    if( !server.hasArg("id") || !server.hasArg("value")
        || server.arg("id") == NULL || server.arg("value") == NULL) {
        server.send(400, "text/plain", "400: Invalid Request");
        return;
    }

    Serial.print(server.arg("id")+ ";" + server.arg("value"));
    delay(1200);

    server.send(200, "text/html", "ok");
}

```

Figura 3. 36: Definición método “sensorRequest”
Fuente: Elaborado por el autor

En la Figura 3.37 se puede observar los métodos “setup” y “loop”, el primero de ellos se ejecuta automáticamente y por una sola vez al energizar el dispositivo, y en este se puede visualizar cómo se llaman a los otros métodos previamente configurados. De la misma manera, el método “loop” muestra la búsqueda constante del servido por peticiones HTTP.

```

void setup() {
    initSerialPort();
    configurateAccesspoint();
}

void loop() {
    server.handleClient();
}

```

Figura 3. 37: Definición métodos “setup” y “loop”
Fuente: Elaborado por el autor

3.3.2. Gateway

Como se vio en el capítulo anterior el Gateway es el encargado de la conectividad de la WSN con una red externa, por lo tanto, es el responsable de la comunicación con los servicios cloud.

El dispositivo que desarrolla esta función es el Arduino MKR WiFi 1010, el código necesario para este elemento de la red se lo describe a continuación:

Para llevar un orden adecuado se separan los archivos de configuración respecto a la lógica del mecanismo de comunicación entre el Arduino y la “Thing” configurada en el entorno cloud.

El primero de los archivos es el “arduino_secret.h” en el que se coloca la información de la red local, a la cual se conecta, para la comunicación vía internet. Esto se puede apreciar en la Figura 3.38.

```
#include <Arduino_ConnectionHandler.h>

/* MKR1000, MKR WiFi 1010 */
#if defined(BOARD_HAS_WIFI)
  #define SECRET_SSID "My-ssid"
  #define SECRET_PASS "My-pass123"
#endif
```

Figura 3. 38: Archivo de configuración “arduino_secret.h”
Fuente: Elaborado por el autor

El segundo archivo de comunicación corresponde a la configuración de la “Thing”, en él se definen las mismas variables que se configuraron en el ambiente cloud. Se debe tener mucho cuidado al momento de colocar los nombres, pues deben ser exactamente iguales a los de la plataforma “Arduino IoT”, de no ser así, los datos no se transmitirán aun cuando en tiempo de compilación el código no genere errores.

El archivo “thingProperties.h” empieza importando las librerías para la conexión con el entorno cloud, las mismas que fueron agregadas en la sección de configuración del entorno local. Como paso 2, se reconoce la tarjeta con la cual se trabaja, posterior a ello se identifica la “Thing” y la “Board” configurada en la plataforma Arduino. Luego de realizar estos pasos se procede a la declaración de las variables mencionadas anteriormente y en el método “initProperties”, que posteriormente será llamado en el método “setup”, se realizan las inicializaciones de estas variables. Como última parte, se procede a iniciar la conexión del dispositivo con el servicio de la nube. Este archivo se observa en la Figura 3.39.

```

#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

#if defined(BOARD_HAS_WIFI)
#else
  #error "Arduino MKR WiFi 1010 not found"
#endif

#define THING_ID "38d766cd-86e4-4f37-b722-4664e84748cb"
#define BOARD_ID "70620c0a-f987-4621-9b03-ba6d04e099b8"

float temperatura;
float movimiento;
float gas;

void initProperties() {

  ArduinoCloud.setThingId(THING_ID);
#if defined(BOARD_HAS_WIFI) || defined(BOARD_HAS_GSM) || defined(BOARD_HAS_NB)
  ArduinoCloud.addProperty(temperatura, Permission::Read).publishOnChange(10);
  ArduinoCloud.addProperty(movimiento, Permission::Read).publishOnChange(10);
  ArduinoCloud.addProperty(gas, Permission::Read).publishOnChange(10);
#endif
}

#if defined(BOARD_HAS_WIFI)
  WiFiConnectionHandler ArduinoIoTPreferredConnection(SECRET_SSID, SECRET_PASS);
#endif

```

Figura 3. 39: Archivo de configuración “thingProperties.h”
Fuente: Elaborado por el autor

En el archivo principal se procede inicialmente a importar los 2 archivos de configuración que se crearon; y, a definir las variables que se utilizan para la lectura de los datos recibidos por el MCU. Esto se observa en la Figura 3.40.

```

#include "arduino_secrets.h"
#include "thingProperties.h"

char c;
String valueSensor;
char idSensor;

```

Figura 3. 40: Configuración inicial de archivo principal
Fuente: Elaborado por el autor

En la Figura 3.41 se observa la definición del método “readSerialPort”, él se encarga de leer la información recibida desde el módulo MCU mediante el puerto serial. Según se definió la trama de transmisión, la primera parte hacía referencia al sensor y la segunda al valor, y dependiendo del id se envía a la variable configurada en el entorno cloud.

```

void readSerialPort() {
  Serial.println(Serial1.available());
  if(Serial1.available() > 0) {
    Serial.println("Recibido: ");

    int index = 0;
    valueSensor = "";
    while(Serial1.available() > 0) {

      c = (char) Serial1.read();

      if(index == 0)
        idSensor = c;

      if(index > 1 && c != 'c')
        valueSensor += c;

      index++;
    }

    if(idSensor == "t")
      temperatura = valueSensor.toFloat();
    else if(idSensor == "g")
      gas = valueSensor.toFloat();
    else if(idSensor == "m")
      movimiento = valueSensor.toFloat();
  }
}

```

Figura 3. 41: Definición método “readSerialPort”
Fuente: Elaborado por el autor

En el método “setup” el cual se observa en la Figura 3.42, primero se aprecia la inicialización del puerto serial para la comunicación con el MCU, luego la ejecución del método “initProperties” que se encarga de inicializar las variables del IoT, después la configuración para los mensajes que se ejecutan internamente por la librería de Arduino; y, posterior a ello la inicialización del proceso de debug para los recursos cloud.

```

void setup() {

  Serial.begin(115200);
  Serial1.begin(9600);

  while(!Serial);

  initProperties();

  /* Initialize Arduino IoT Cloud library */
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  setDebugMessageLevel(DBG_INFO);
  ArduinoCloud.printDebugInfo();
}

```

Figura 3. 42: Definición de método “setup”
Fuente: Elaborado por el autor

Finalmente, en el método “loop” se ejecutan los procesos antes definidos para leer la información del puerto serial y transmitir a los servicios en la nube. A todo esto, se le aplica un retardo de 1s entre cada iteración.

```
void loop() {  
    ArduinoCloud.update();  
    readSerialPort();  
    delay(1000);  
}
```

Figura 3. 43: Definición de método “loop”
Fuente: Elaborado por el autor

3.3.3. Nodos sensores

El funcionamiento de los nodos sensores es tal vez la parte más simple de la red, pero no por esto la menos importante, pues de ellos se adquiere la información para que tenga sentido todo. A continuación, se revisa el código del firmware de cada uno de ellos.

Sensor de Gas

En la Figura 3.44 se observa la inicialización del MCU en modo cliente, conectada a la red del AP que se configuró antes, en esta porción de código se observa la razón por la que se definió una ip fija, pues de no ser así, cada vez que se inicializa el AP, esta cambia. En el método “loop” se visualiza cómo se lee la data del puerto analógico, para posteriormente ser transmitida por HTTP hasta el servidor.

```

#include <ESP8266WiFi.h>

#define SERVER_IP "192.168.1.200"

WiFiClient client;

char ssid[] = "My-SSID";
char pass[] = "@Abcl12358";
int mq2 = A0;
int data = 0;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
}

void loop() {
  if(client.connect(SERVER_IP, 80)) {

    data = analogRead(mq2);

    String url = "/update?id=g&value=";
    url += String(data);

    client.print(String("POST ") + url + " HTTP/1.1\r\n" + "Host: " + SERVER_IP + "\r\n" + "Connection: keep-alive\r\n\r\n");
  }

  delay(1000);
}

```

Figura 3. 44: Firmware nodo sensor de gas
Fuente: Elaborado por el autor

Sensor de Temperatura

Para poder tomar la temperatura utilizando el sensor DHT11 es necesario importar las librerías de este elemento, la misma que ya se la tiene incorporada en nuestra biblioteca. En la Figura 3.43, se puede observar que la configuración de conexión hacia el AP no varía, y de todo el código que se aprecia, solo cambia la inicialización de la variable que se encarga de capturar la información, mediante el uso del pin digital 2 (DHTPIN) y el tipo de sensor que se está utilizando (DHTTYPE).

Luego de las inicializaciones correspondientes solo basta con capturar la información mediante el método “readTemperature()”, que se lo utiliza en el método “loop”, posterior a ello se procede a enviar la información con la definición de trama que se había definido anteriormente.

```

#include <ESP8266WiFi.h>
#include <DHT.h>

#define SERVER_IP "192.168.1.200"
#define DHTPIN 2
#define DHTTYPE DHT11

WiFiClient client;
DHT dht(DHTPIN, DHTTYPE);

char ssid[] = "My-SSID";
char pass[] = "@Abcl12358";
float temp = 0;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  dht.begin();
}

void loop() {
  if(client.connect(SERVER_IP, 80)) {

    temp = dht.readTemperature();

    String url = "/update?id=tvale=";
    url += String(temp);

    client.print(String("POST ") + url + " HTTP/1.1\r\n" + "Host: " + SERVER_IP + "\r\n" + "Connection: keep-alive\r\n\r\n");
  }

  delay(1000);
}

```

Figura 3. 45: Firmware nodo sensor de temperatura
Fuente: Elaborado por el autor

Sensor de Movimiento

El sensor de movimiento o también conocido como PIR, es quizá el más sencillo de programar su firmware, pues basta con identificar un valor alto en el pin configurado, en este caso el pin 2 (INPIN), que se define de tipo "INPUT" en el método "setup", y posterior a ello, se lee dicho pin, haciendo uso del método "digitalRead". El código de este firmware se puede observar en la Figura 3.46.

```

#include <ESP8266WiFi.h>

#define SERVER_IP "192.168.1.200"
#define INPIN 2

WiFiClient client;

char ssid[] = "My-SSID";
char pass[] = "@Abc112358";
int pir = 0;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass);

  pinMode(INPIN, INPUT);
}

void loop() {
  if(client.connect(SERVER_IP, 80)) {

    pir = digitalRead(INPIN);

    String url = "/update?id=m&value=";
    url += String(pir);

    client.print(String("POST ") + url + " HTTP/1.1\r\n" + "Host: " + SERVER_IP + "\r\n" + "Connection: keep-alive\r\n\r\n");
  }

  delay(1000);
}

```

Figura 3. 46: Firmware nodo sensor de movimiento
Fuente: Elaborado por el autor

3.4. Visualización de resultados

En la Figura 3.47 se muestra el dashboard de monitoreo, y los elementos “Gauge”, que son los que muestran el nivel de temperatura y de gas emitido por los sensores; otro elemento que se visualiza es un “switch”, que sirve para activar manualmente la alerta del sensor de movimiento, que tiene como rango de ejecución automático, el horario de 22:00 a 6:00. El último elemento que se utiliza es el led; este se usa para alertar cuando el nivel detectado es igual o superior al punto crítico.

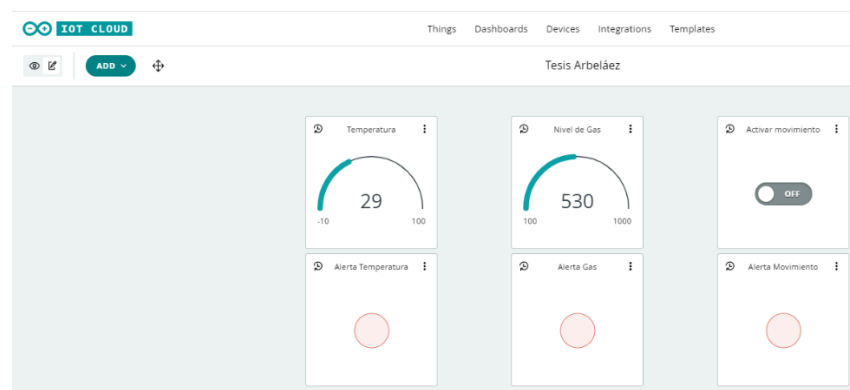


Figura 3. 47: Dashboard de monitoreo.
Fuente: Elaborado por el autor

Los valores críticos que se utilizan en el sistema se pueden ver en la siguiente tabla.

Nombre Sensor	Valor crítico
Temperatura	40
Gas	600
Movimiento	1

Tabla 3. 1: Valores críticos.

Fuente: Elaborado por el autor

El sistema es apto para monitorear las incidencias determinadas por el firmware del elemento Gateway, permitiendo visualizar en cualquier momento los niveles que emiten estos sensores; pero esto no es suficiente en cuanto a seguridad, por lo tanto, es posible utilizar aplicativos móviles que hacen uso del concepto IoT para notificar alguna incidencia. Blynk es uno de ellos, mediante una configuración fácil nos permite vincular nuestros parámetros a lo que ellos denominan “notificators” con la finalidad de enviarnos un mensaje al teléfono inteligente o a cualquier otro dispositivo Smart que se conecte a este, como por ejemplo los relojes o pulseras inteligentes.

Las notificaciones se pueden apreciar en las siguientes imagenes



Figura 3. 48: Alerta de evento con Blynk.

Fuente: Elaborado por el autor

Al abrir el listado de notificaciones se puede apreciar con mayor claridad la notificación recibida, y el mensaje de esta.

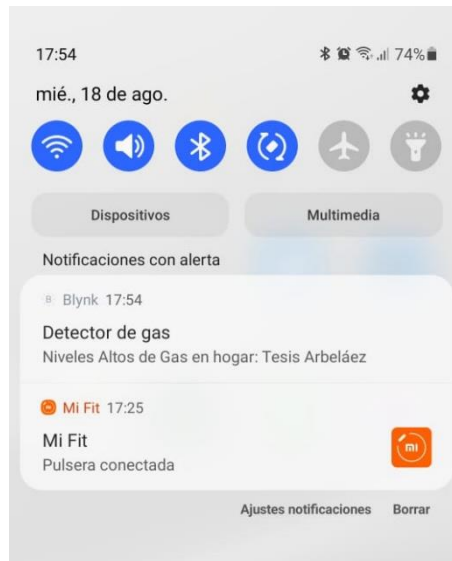


Figura 3. 49: Alerta de evento con Blynk 2.
Fuente: Elaborado por el autor

Si el usuario cuenta con un reloj inteligente o una Smart band vinculada al teléfono celular con el que monitorea la aplicación, también le llegará la notificación.



Figura 3. 50: Alerta de evento con Blynk en smart band.
Fuente: Elaborado por el autor

Capítulo 4: Conclusiones y Recomendaciones

4.1. Conclusiones

- Este diseño está principalmente elaborado para hogares de adultos mayores, pues son más susceptibles a sufrir accidentes por aspectos propios de su edad, entre ellos tal vez el más común es dejar abiertas las hornillas de la cocina y exponerse a gases tóxicos.
- Con el uso del Arduino MKR WiFi 1010 se logró una conexión estable con los recursos cloud, debido a que una de las características principales de este modelo de placa es el uso en proyectos IoT.
- En el presente trabajo se logró diseñar una red, cuyo acceso es controlado utilizando credenciales de usuario, protegiendo de esta forma, la integridad de los datos analizados y posteriormente transmitidos a los dashboards de monitoreo.
- En el diseño de la red fueron considerados elementos inalámbricos, con lo cual se evita el uso de dispositivos conectados de forma física, permitiendo distribuir los sensores en donde se consideren mayores posibilidades de incidencia.
- El uso del sensor de gas otorga a la vivienda del adulto mayor un recurso de seguridad, al evitar la asfixia por inhalación accidental en sus horas de descanso.
- Con el uso del sensor de temperatura incorporado en este diseño se puede identificar si los niveles térmicos no son aptos para la salud del adulto mayor y aplicar correctivos a corto plazo.
- Debido a que no siempre un usuario podría estar monitoreando las señales tomadas por los sensores, se habilitaron alertas que llegarán

a todos los dispositivos móviles que se encuentren aptos para recibirlas.

- Con el uso de blynk, se consiguió la alerta en tiempo real y con ello notificar a los usuarios conectados.

4.2. Recomendaciones

- Se recomienda el uso de Blynk en los proyectos de mayor escala, pues brinda una mayor versatilidad para en el envío de notificaciones.
- Se recomienda que se incremente el número de sensores para tener una mayor cobertura de prevención, incorporando un extractor de aire cuando el sistema detecte gas en el ambiente.
- Se recomienda monitorear la humedad, la cual es una propiedad que ya lo realiza el sensor DHT que se utiliza para la medición de temperatura.

Bibliografía

- Alcázar, V. A. (2013). *Estudio de las prestaciones de IEEE 802.15.4e*. 66.
- All Metro Tech. (2018). All Metro Tech. <https://allmetrotech.com/blog/3-characteristics-of-the-best-smart-home-automation-systems>
- Arduino Inc. (2021). *Arduino Editor*. <https://create.arduino.cc/editor/alfredo-arbelaez/00467b6e-1eb1-458c-8df7-a61a5d478afc>
- Arduino MKR WiFi 1010 | Arduino Official Store. (2021). <https://store.arduino.cc/usa/mkr-wifi-1010>
- Arduino Pro. (2021). *Arduino Pro*. <https://www.arduino.cc/pro/overview>
- Carretero, O. S., Lladró, R. C., & de Julio. (2016). *Casa Domótica con Arduino*. 182.
- Castaño, D. F. M. (2019). *DESARROLLO DE UNA RED INALÁMBRICA DE BIOSENSORES PARA LA DETECCIÓN DE CONTAMINANTES TÓXICOS EN AGUAS*. 61.
- Castro C., N. D., Chamorro F., L. E., & Viteri M., C. A. (2016). Una red de sensores inalámbricos para la automatización y control del riego localizado. *Revista de Ciencias Agrícolas*, 33(2), 106. <https://doi.org/10.22267/rcia.163302.57>
- designthemes. (2018). *Sensores de temperatura DHT11 | Tienda y Tutoriales Arduino*. <https://www.prometec.net/sensores-dht11/>
- Domotizados. (2018a, enero 24). Así se originó la domótica y esta es su evolución. *Domotizados.co*. <https://domotizados.co/asi-se-origino-la-domotica-y-esta-es-su-evolucion/>

Domotizados. (2018b, enero 30). La importancia de tener un sistema domótico de seguridad en el hogar. *Domotizados.co*. <https://domotizados.co/la-importancia-de-tener-un-sistema-domotico-de-seguridad-en-el-hogar/>

Eiko Yoneki, J. B. (2005). *A survey of Wireless Sensor Network technologies: Research trends and middleware's role* [Technical]. University of Cambridge.

https://d1wqtxts1xzle7.cloudfront.net/42844186/Survey_of_Wireless_Sensor_Networks_UCAM-CL-TR-646-with-cover-page-v2.pdf?Expires=1624427265&Signature=bgrMRymzloww0ZbgRPrle3LgfNFSds9FWhcBQ716lZel71PnTIT3i4i6JWmrneHA8NKG27zCQ3rRdO2TCw0kIE7UO8GR0ne9416T9CTsrHVvsQRQKRErNZ4ei9q2is1WaT73VEt265WDqz3zcYLZyoeA-I7M1rSEAmDWSpz7HXAGFIkJVBiJjNuerH-7Ql0HgZsrQRNTwT1z-g32gAvjPEkPNLwjxMBWZR9NsZQUz2hhu8wfOxKOpCsGpbviHSR~56YLNKAnhLGdy2WRKMmvG-t0QN63ynNr6UTMvkXPhLAXh~vXA7UibYm1ZHdvjPF9xAQN1R9s6BrUgXaU4P~OKw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Fénix, N. (2014). *Arduino vs PIC: La gran batalla | Nextia Fenix*. <https://www.nextiafenix.com/arduino-vs-pic-la-gran-batalla/>

Fernandez Barcell, M. (2014). Redes de sensores inalámbricas. *Wireless Sensor Network*, 20.

filehorse. (2021). *Arduino 1.8.15 Descargar para Windows / Imágenes / FileHorse.com*. <https://www.filehorse.com/es/descargar-arduino/imagenes/>

- Granda, B., & Andrés, B. (2017). *Diseño de una red de sensores inalámbricos para el monitoreo del tránsito vehicular y la contaminación CO2 dentro de un sector urbano* [Tesis]. 44444444444444. <https://dspace.ups.edu.ec/bitstream/123456789/14153/1/UPS-CT006969.pdf>
- Iberdrola. (2017). *La casa domótica: ¿presente o futuro?* Iberdrola. <https://www.iberdrola.com/innovacion/domotica>
- jecrespom. (2016). *Aplicaciones Industriales. Aprendiendo Arduino*. <https://aprendiendoarduino.wordpress.com/tag/aplicaciones-industriales/>
- Learte, P. J. (2018, julio 16). Introducción al IoT, Internet of Things. *Blog de Hiberus Tecnología*. <https://www.hiberus.com/crecemos-contigo/introduccion-al-iot-internet-of-things/>
- naylampmechatronics. (2016). *Tutorial sensores de gas MQ2, MQ3, MQ7 y MQ135*. Naylamp Mechatronics - Perú. https://naylampmechatronics.com/blog/42_tutorial-sensores-de-gas-mq2-mq3-mq7-y-mq135.html
- Olmeda Martínez, A. (2018). *Diseño de una central domótica low-cost, basado en arduino para viviendas particulares y pequeñas empresas* [TAZ, Universidad de Zaragoza]. <http://zaguan.unizar.es/record/78617/files/TAZ-TFG-2018-4467.pdf>
- programarfacil. (2017). *Cómo programar NodeMCU con IDE Arduino, tutorial paso a paso*. <https://programarfacil.com/esp8266/como-programar-nodemcu-ide-arduino/>

Proyectos IoT con Arduino, análisis de las plataformas más importantes.

(2016). <https://programarfacil.com/podcast/proyectos-iot-con-arduino/>

Román, E. F. (2017). *DKONXMOTIZACIÓN DE UN EDIFICIO Autor: David*

González Cañadas. 187.

Yunquera Torres, J. (2005). *Diseño de una red Wi-Fi para la E.S.I. [TAZ].*

<http://bibing.us.es/proyectos/abreproy/11138/>

Zemsania. (2018, noviembre 16). Domótica e IoT para hacer que tu casa sea

inteligente—Zemsania. *Zemsania Global Group.*

<https://zemsaniaglobalgroup.com/domotica-e-iot-hogar/>



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Arbeláez Martínez, Carlos Alfredo** con C.C: # 092034257-3 autor del Trabajo de Titulación: **Diseño de una red domótica para monitoreo y alerta de emergencias en viviendas con adultos mayores utilizando Arduino MKR WiFi 1010** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 15 de Septiembre del 2021

f. _____

Nombre: Arbeláez Martínez, Carlos Alfredo

C.C: 092034257-3

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Diseño de una red domótica para monitoreo y alerta de emergencias en viviendas con adultos mayores utilizando Arduino MKR WiFi 1010		
AUTOR(ES)	ARBELÁEZ MARTÍNEZ, CARLOS ALFREDO		
REVISOR(ES)/TUTOR(ES)	M. Sc. EDWIN F. PALACIOS MELÉNDEZ		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	15 de Septiembre del 2021	No. DE PÁGINAS:	68
ÁREAS TEMÁTICAS:	Sistemas Embebidos y Comunicaciones Inalámbricas		
PALABRAS CLAVES/ KEYWORDS:	SISTEMAS, DOMÓTICA, SENSORES, ARDUINO, CLOUD, IOT		
RESUMEN/ABSTRACT (150-250 palabras):			
<p>La evolución de los sistemas domóticos se ha convertido en los últimos años en una gran opción para tecnificar los hogares. La toma de decisiones es uno de los principales retos de un hogar inteligente, la pronta identificación y alerta de posibles emergencias es quizá el punto de quiebre para el incremento en el uso de esta tecnología; a esto se le suma la posibilidad de una conectividad continua, que abre la puerta a la monitorización en tiempo real de todos los dispositivos vinculados. La necesidad del momento es brindar seguridad a los hogares en general, pero en mayor medida a las viviendas de adultos mayores quienes son más proclives a sufrir accidentes que se agravan por la falta de compañía en la mayoría de los casos. En este trabajo se propone una opción integral para interconectar una red de sensores con los servicios cloud de Arduino, los cuales se orientan al IoT, la cual garantiza el correcto funcionamiento y fácil escalamiento de dispositivos sensoriales en la red domótica. La idea propuesta es implementada para probar de manera práctica y validar su eficacia y utilidad.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593995931075	E-mail: alfredoarbelaez26@gmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez, Edwin Fernando		
	Teléfono: +593-9-67608298		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			