

ALARMA CONTRA INCENDIOS

Quintana Fuentes Jose Daniel
Diseño Microelectronico Digital
Universidad del Magdalena
Santa Marta, Magdalena
 josequintanadf@unimagdalena.edu.co

Resumen—En este laboratorio se realizó una alarma contra incendios que se basaba en la implementación de esta para que cuando se detecte que la temperatura exceda los 315° grados Celsius a la altura de los ojos y envíe una alerta a la cuenta de e-mail del propietario inscrita.

Palabras Clave: incendio, alerta, base de datos.

Abstract—In this laboratory, a fire alarm was made that was based on the implementation of this so that when it is detected that the temperature exceeds 315 degrees Celsius at eye level and I send an alert to the owner's email account enrolled.

Keywords: *fire, alert, database.*

OBJETIVOS:

- Desarrollar un detector de temperatura ideal para ser utilizado en espacios cerrados usando herramientas como plataformas de almacenamiento de datos y dispositivos electrónicos.

I. Modelo Teorico

Temperatura de incendios

El calor es más perjudicial que las llamas. La temperatura ambiente en un incendio puede ser de 100 grados Fahrenheit, es decir, 38 grados Celsius al nivel del piso y llegar a 600 grados Fahrenheit, es decir, 315 grados Celsius al nivel de los ojos [1].

Sensor DHT22



Figura 1. Sensor DHT22 Tomado de [2]

El sensor DHT22 admite monitorear temperatura y humedad relativa de manera precisa y sencilla a un bajo costo. La salida suministrada es digital, no se requiere utilizar entradas analógicas [2].

BASES DE DATOS

Relacionales: Es una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas.

- **MySQL:**

MySQL es un sistema de comisión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y fundado en el lenguaje de consulta estructurado (SQL). MySQL trabaja prácticamente en todas las plataformas, conteniendo Linux, UNIX y Windows. Aunque puede utilizarse en una amplia gama de aplicaciones, MySQL se asocia frecuentemente con las aplicaciones web y la publicación en línea [4].

- **MariaDB:**

MariaDB es un sistema de comisión de bases de datos que está muy relacionado con MySQL, ya que fue creado por uno de los desarrolladores, Michael “Monty” Widenius. Su desarrollo tiene como objetivo mantener el software de gestión de base de datos en un modelo de software libre [5].

- PostgreSQL

En este sistema, las consultas relacionales se basan en SQL. Dos detalles para destacar de PostgreSQL es que posee data types (tipos de datos) avanzados y admite ejecutar optimizaciones de rendimiento modernas, que son tipologías que por lo general solo se ven en sistemas de bases de datos comerciales, como por ejemplo SQL Server de Microsoft u Oracle de la compañía homónima [6].

No relacionales: Poseen esquemas flexibles y son fácilmente escalables.

- **MongoDB:**

MongoDB (del inglés humongous, "enorme") es un sistema de base de datos NoSQL encaminado a documentos de código abierto y escrito en C++, que en lugar de almacenar los datos en tablas lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico [7].

- **DynamoDB**

Amazon DynamoDB es una base de datos NoSQL de clave-valor sin servidor y totalmente administrada que está trazada para ejecutar aplicaciones de alto rendimiento a cualquier escala. DynamoDB ofrece seguridad integrada, copias de seguridad continuas, replicación computarizada en varias regiones, acumulación de caché en memoria y herramientas de importación y exportación de datos [8].

- Cassandra

Apache Cassandra es un sistema de gestión de bases de datos (DBMS) de código abierto para bases de datos muy grandes, pero arregladas. Gracias a la buena escalabilidad, estas bases de datos se pueden distribuir a diferentes clústeres, por lo que Cassandra no se encuentra incorporada a un único servidor [9].

Dash:

Escrito sobre Flask, Plotly.js y React.js, Dash es ideal para crear aplicaciones de visualización de datos con interfaces de usuario altamente personalizadas en Python puro. Las aplicaciones de Dash se representan en el navegador web. Puede implementar sus aplicaciones en servidores y luego compartirlas a través de URL.

ChartJS:

Gráficos de JavaScript simples pero flexibles para diseñadores y desarrolladores.

JavaScript

Es un lenguaje del lado del cliente que se ejecuta completamente en el navegador web. Para invocarlo, debe estar entre las etiquetas HTML `<script>` y `</script>`.

SERVICIOS EN LA NUBE

- Amazon Web Services

AWS cuenta con un conjunto de servicios y de tipologías incluidas en ellos que supera la de cualquier otro proveedor de la nube, ofreciendo desde tecnologías de infraestructura como cómputo, acopio y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas. Esto hace que trasladar las aplicaciones existentes a la nube sea más rápido, fácil y rentable y permite crear casi cualquier cosa que se pueda imaginar [10].

- Google Cloud

Es manejada para crear ciertos tipos de medios a través de la tecnología almacenada en la nube y permite por ejemplo destacar la rapidez y la escalabilidad de su infraestructura en las aplicaciones del buscador. [11]

- Microsoft Azure

Azure es una nube pública de pago por uso que te condesciende compilar, implementar y administrar rápidamente aplicaciones en una red global de datacenters (centros de datos) de Microsoft [12].

- Alibaba Cloud

Alibaba Cloud, también conocida como AliCloud, brinda una gama de servicios de infraestructura, plataforma y aplicaciones en muchas áreas. Dichas ofertas circunscriben servicios de almacenamiento, red, base de datos, análisis, seguridad y plataforma de aplicaciones [13].

- Heroku

Heroku es una solución de Plataforma como Servicio (PaaS) fundada en la nube para que el cliente solo se preocupe de desplegar su aplicación mientras Heroku se encarga de la infraestructura que hay detrás [14].

- DigitalOcean

Digital Ocean es un proveedor de hospedaje en la nube y que cuenta con opciones superiores a métodos tradicionales. Considerada como una de las 3 mejores del giro a nivel mundial, su principal encantador consiste

en la escalabilidad, lo que te permite contar con más recursos a medida que tu presencia digital se fortalece [15].

Servicio AMAZON ELASTIC COMPUTE CLOUD (AMAZON EC2)



Figura 2. AMAZON ELASTIC COMPUTE CLOUD Tomado de [3]

Amazon Elastic Compute Cloud (Amazon EC2) suministra capacidad de computación escalable en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como precise, establecer la seguridad y las redes, y administrar el almacenamiento [3].

MicroPython

Es una consumación software del lenguaje de programación Python 3, escrita en C, y que está optimizada para poder ejecutarse en un microcontrolador. MicroPython es un compilador completo del lenguaje Python y un motor e intérprete en tiempo de ejecución, que funciona en el hardware del microcontrolador [16]. Entre sus características se encuentran:

- Optimizado para ejecutarse en microcontroladores.
- Posee características avanzadas, entre las cuales se encuentran:
 - Prompt interactivo.
 - List comprehension.
 - Manejo de excepciones.
- MicroPython es compacto, puede ser alojado y ejecutado en sólo 256k memoria de programa y 16k de RAM.

Protocolo MQTT

La comunicación MQTT utiliza el protocolo SSL para resguardar los datos confidenciales que transmiten los dispositivos IoT. Puede efectuar la identidad, la autenticación y la autorización entre los clientes y el agente mediante certificados SSL y contraseñas [17].

II. DESARROLLO DE CONTENIDOS

Para el desarrollo de este proyecto se decidió realizar una alarma contra incendio, haciendo uso de la ESP32 y el sensor de temperatura y humedad DHT22, también el uso de una bocina o

luces de advertencia a 110AC y módulo relay.
Se tomarán datos de la temperatura mediante el sensor DHT22 Y enviados a la ESP32, esta se encarga de almacenar los datos de los valores de la temperatura medidas y posteriormente se visualizarán en tiempo real de manera gráfica en una página web. Al captar un valor de temperatura elevada sobre los valores normales, este activara un led como alarma y mandara un mensaje al correo del propietario. Está alarma estará conectada a la red 110AC de manera interrumpida por un módulo relay que al ser activado permitirá el paso de una línea activando el led.

A continuación, podremos observar el código para la *ESP32* y para el *Servidor* respectivamente.

ESP32:

En la ESP32 encontraremos dos códigos, primero el “main” donde se ejecuta el código principal y la librería de “umqttsimple” necesaria para el cliente de MQTT.

- main.py:

Primero es necesario conectar la esp32 vía internet:

```
8  SSID = "Jose2001"
9  PASSWORD = "aligente"
10
11
12 def do_connect(SSID, PASSWORD):
13     wlan = network.WLAN(network.STA_IF)
14     wlan.active(True)
15
16     if not wlan.isconnected():
17         print("connecting to network...")
18         wlan.connect(SSID, PASSWORD)
19
20     while not wlan.isconnected():
21         pass
22     print("network config:", wlan.ifconfig())
```

Luego nos conectamos con nuestro servidor por medio de MQTT:

```
25 def run():
26     do_connect(SSID, PASSWORD)
27
28     SERVER = "15.228.254.5"
29     client = MQTTClient("test", SERVER)
30
31     topic = "test"
32
33     client.connect()
```

Finalmente leemos los datos del sensor:

```
35 while True:
36     variables = {
37         "fecha": str(time.localtime()),
38         "numero1": random.random()*10,
39         "numero2": random.random(),
40     }
41     payload = ujson.dumps(variables)
42     print(payload)
43     client.publish(topic, payload)
44     time.sleep(1)
45
46     client.disconnect()
```

SERVIDOR:

- PRIMERA PARTE: INSTANCIA EC2 – AWS

El servicio de nube seleccionará es AWS porque este nos permite tener 12 meses gratis después de la fecha inicial, además ofrece diversos servicios de los cuales entre estos esta la

instancia de EC2, la cual será donde se alojará el servidor de Linux.

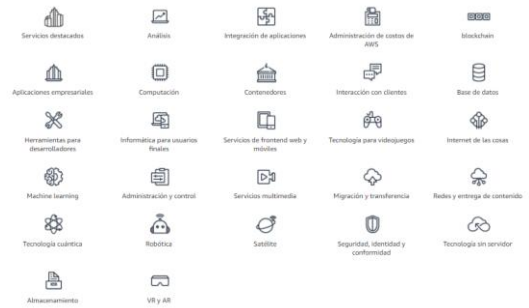


Figura 3. Servicios de AWS.

Se elige el apartado de computación porque la instancia EC2 se encuentra ahí:



Figura 4. Computacion en AWS.

Se selecciona la instancia Amazon Elastic Compute Cloud (Amazon EC2). Esto conduce al panel principal de EC2 que nos deja hacer diferentes funciones y la instancia.

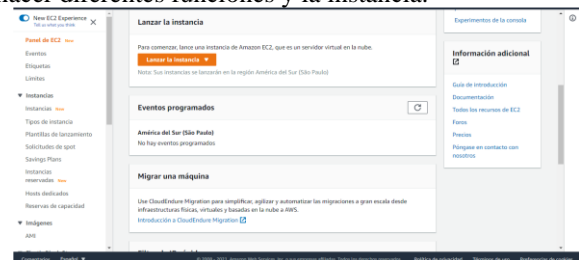


Figura 5. Panel inicial de EC2.

Se elige una imagen de AMI para mayor facilidad se usará un en tomado “Amazon Linux 2 de AMI” de 64bits (x86).

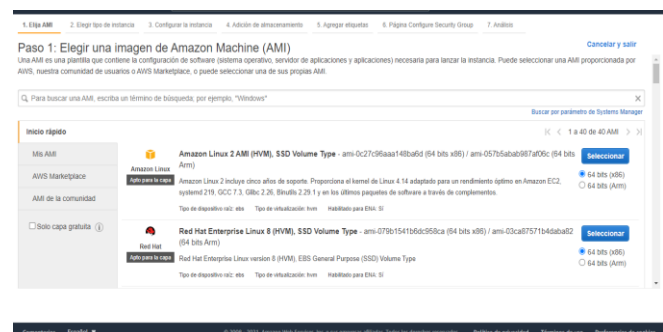


Figura 6. Elegir AMI.

A continuación, se elige el tipo de instancia, se utilizo la familia t2.micro el cual es apto para la capa gratuita que se posee.

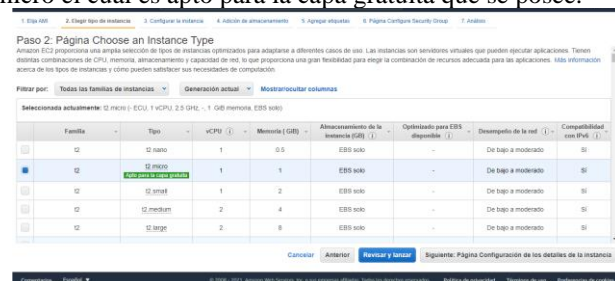


Figura 7. Seleccionar instancia.

Ahora, se configura el grupo de seguridad y se añade un nuevo

tipo ademas de SSH llamado HTTP .

Figura 8. Configurar grupo de seguridad.

Se pasa a generar la clave del par de claves el cual permitira conectarse a la instancia de forma segura.

Figura 9. Generar clave para conexión ssh.

Finalmente, se termina el proceso de configuracion de la instancia en EC2

Figura 10. Fin de la configuración para el lanzamiento.

En la siguiente imagen se puede visualizar nuestra instancia en ejecución

Figura 11. Instancia en ejecución.

En el apartado de conectar en la instancia se puede ver las maneras de algunas informaciones generales de la instancia y como conectarnos a ellas por medio del cliente SSH

Figura 12. Información general sobre la instancia.

De esta manera se puede conectar a la instancia por medio del cliente SSH el cual AWS nos brinda una guía de como realizar.

Figura 13. Conexión ssh.

Una vez siguiendo los pasos anteriores se puede entrar a nuestra instancia siguiendo los pasos de SSH

Figura 14. Ingresar con ssh.

- **SEGUNDA PARTE: CONFIGURAR APACHE CON WSGI:**

Para ayudarnos con la configuración de un sitio web vamos a apoyarnos con el servidor de HTTP de Apache

1. **Crear carpeta:** /var/www/Prueba

- Crear archivo: prueba.wsgi
- Crear archivo: prueba.py

2. **Crear archivo .conf:** en la ruta /etc/httpd/conf.d, ejemplo:

- /etc/httpd/conf.d/prueba.conf

3. **Editar el archivo:**

/etc/httpd/conf.modules.d/10-wsgi.conf

Cambiar: LoadModule wsgi_module modules/mod_wsgi.so

Por: LoadModule wsgi_module

“usr/local/lib64/python3.7/site-packages/mod_wsgi/server/mod_wsgi-py37.cpython-37m-x86_64-linux-gnu.so”

- **Nota:** La ruta del módulo wsgi varía dependiendo de la versión de python que se tenga instalada.

4. **Reiniciar Apache:** sudo systemctl restart httpd

- **TERCERA PARTE: CREACIÓN DE BASE DE DATOS**

Se procede a crear las bases de datos necesarias:

1. Tabla de datos de las gráficas en tiempo real (fecha de adquisición, primer dato, segundo dato)

CREATE DATABASE monitoreo ;

USE monitoreo ;

CREATE TABLE datos_t tiempo_real (
i d INT NOT NULL AUTO_INCREMENT,
fecha_adquisicion VARCHAR(255) ,
numero1 FLOAT,

numero2 FLOAT,
PRIMARY KEY (i d)) ENGINE MyISAM;

2. Tabla de registro de los usuarios (email)

USE monitoreo ;

```
CREATE TABLE datos_form (
  i d INT NOT NULL AUTO_INCREMENT,
  email VARCHAR(255) ,
  PRIMARY KEY ( i d ) ) ENGINE MyISAM;
```

- CUARTA PARTE: CONFIGURAR AMI PARA USAR MQTT



III.CONCLUSIONES

Finalmente, se pudo realizar la alarma contra incendios, para llegar a esto, nos apoyamos inicialmente del servicio brindado por AWS por medio de E2. Luego de esto, se realizó el código en microphyton en el cual se describía el comportamiento base del sistema configurando una temperatura adecuada para poder realizar la prueba del sistema en situación ausente de incendio, dado que lo anterior se dio con éxito se comprobó el funcionamiento correcto de este y el límite de temperatura se dejó a disposición de la modificación del usuario.

Referencias

- QUINTA PARTE: RECIBIR LOS DATOS

En este apartado además de recibir los datos por MQTT del sensor se realiza la función de advertencia de un posible incendio por medio del correo electrónico que anteriormente en la página había registrado.

```
def email():
    message = '#Advertencia! Posible incendio!#'
    subject='Advertencia!'
    message = 'Subject: {}\\n\\n{}'.format(subject, message)
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(config('NAME_MAIL'), config('PASSWORD_MAIL'))
    server.sendmail('Alerta.Temprana.Incendio@gmail.com', 'josequintanad@unimagdalena.edu.co', message)
    server.quit()
    print("Correo enviado exitosamente!")
```

En la misma función donde recibimos el Payload con los datos del sensor realizamos una verificación del valor de la temperatura.

```
def on_message(client, userdata, msg):
    valores_json = str(msg.payload, 'utf-8')
    valores = json.loads(valores_json)
    cargarDB(valores)
    print(valores)
    if valores['temp'] > 27:
        email()
```

Prototipo Final:



- [1] D. y. emergencias, «Incendios en el hogar,» [En línea]. Available: <https://www.ready.gov/es/incendios-en-el-hogar>.
- [2] NAYLAMP, «SENSOR DE TEMPERATURA Y HUMEDAD RELATIVA DHT22 (AM2302),» [En línea]. Available: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>.
- [3] aws, «¿Qué es Amazon EC2?,» [En línea]. Available: https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html.
- [4] ComputerWeekly.es, «MySQL,» [En línea]. Available: <https://www.computerweekly.com/es/definicion/MySQL>.
- [5] hostingplus, «Qué es MariaDB y cuáles son sus características,» [En línea]. Available: <https://www.hostingplus.com.co/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/>.
- [6] infranetworking, «Servidor PostgreSQL,» [En línea]. Available: <https://blog.infranetworking.com/servidor-postgresql/>.
- [7] OpenWebinars, «QueEsMongoDB,» [En línea]. Available: <https://openwebinars.net/blog/que->

es-mongodb/.

- [8] aws, «Amazon DynamoDB,» [En línea]. Available:
https://aws.amazon.com/es/dynamodb/?trk=74365dcd-ec2d-40b2-a201-8d654d09350d&sc_channel=ps&s_kwcid=AL!4422!3!590500029985!e!!g!!amazon%20dynamodb&ef_id=Cj0KCQiA14WdBhD8ARIsANa007j-RiPkStwPWbBN1M0l9AsQCZSyfGpd6ON-TCgY9Lo38H_TbrJlh4saAgL6EALw_wcB:Gs&s_kwci.
- [9] D. G. OINOS, «Apache Cassandra: gestión distribuida de grandes bases de datos,» [En línea]. Available:
<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/apache-cassandra/>.
- [10] aws, «aws,» [En línea]. Available:
<https://aws.amazon.com/es/what-is-aws/>.
- [11] wikipedia, «Google Cloud,» [En línea]. Available:
https://es.wikipedia.org/wiki/Google_Cloud.
- [12] S. i. Tecon, «Blog. Simplificanco la Tecnología.,» [En línea]. Available:
<https://www.tecon.es/que-es-microsoft-azure-como-funciona/>.
- [13] ComputerWeekly.es, «Una introducción a Alibaba Cloud para empresas occidentales,» [En línea]. Available:
<https://www.computerweekly.com/es/consejo/Una-introduccion-a-Alibaba-Cloud-para-empresas-occidentales>.
- [14] S. nts, «¿Qué es Heroku? ¿Para qué sirve? ¿Con Salesforce cómo se integra?,» [En línea]. Available: <https://www.nts-solutions.com/blog/heroku-que-es.html>.
- [15] rockcontent, «Digital Ocean: qué es, cómo usar, ventajas y desventajas,» [En línea]. Available:
<https://rockcontent.com/es/blog/digital-ocean/>.
- [16] udc, «Microphyton,» [En línea]. Available:
<https://www.uco.es/atdfiware/programacion/34-micropython>.
- [17] aws, «¿Qué es MQTT?,» [En línea]. Available:
<https://aws.amazon.com/es/what-is/mqtt/>.