



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

**CARRERA DE ESPECIALIZACIÓN EN
SISTEMAS EMBEBIDOS**

MEMORIA DEL TRABAJO FINAL

**Sistema de notificación de alarma de
incendio**

Autor:

Ing.Daniel Marquez

Director:

Mg.Ing. Wilmer Sanz (UC)

Jurados:

Dr.Ing.Adrián Stacul (UTN)

Esp. Ing. Diego Fernández (FIUBA, DEBMEDIA)

Esp. Ing. Santiago Salamandri (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires,
entre marzo de 2020 y diciembre de 2020.*

Resumen

La presente memoria describe la implementación de un sistema de monitoreo de alarmas de incendio, con el objetivo de satisfacer la necesidad detectada por la firma Isolse SRL; brindar a sus clientes la posibilidad de conocer de forma remota el estado de su instalación de detección de incendio.

El sistema se elabora utilizando conocimientos de programación de microcontroladores e ingeniería de software, para el desarrollo de un sistema modularizado y respaldado con control de versiones. Se basa en el desarrollo de aplicaciones en sistemas operativos Linux, protocolos de comunicación, diseño de circuitos impresos, manejo de bases de datos y micro-servidor web.

Agradecimientos

Esta sección es para agradecimientos personales y es totalmente **OPCIONAL**.

Índice general

| | |
|---|-----------|
| Resumen | I |
| 1. Introducción general | 1 |
| 2. Introducción específica | 3 |
| 3. Diseño e implementación | 5 |
| 4. Ensayos y Resultados | 7 |
| 4.1. Validación de componentes | 7 |
| 4.1.1. Nrf24l01+ | 7 |
| 4.1.2. Biblioteca RF24 | 8 |
| 4.1.3. Biblioteca RF24Mesh | 10 |
| 4.2. Hardware | 11 |
| 4.2.1. Prototipo: dispositivo primario. | 11 |
| 4.2.2. Prototipo: dispositivo secundario. | 11 |
| 4.2.3. Prototipo: dispositivo secundario. | 12 |
| 4.3. Pruebas unitarias | 13 |
| 4.3.1. Estado local. | 13 |
| 4.3.2. Sistema con un dispositivo secundario. | 13 |
| 4.3.3. Sistema con dos dispositivos secundarios. | 14 |
| 4.3.4. Sistema con tres dispositivos secundarios. | 15 |
| 4.4. Prueba de sistema | 16 |
| Interfaz web | 16 |
| Aplicación Android | 16 |
| Histórico de eventos como base de datos | 17 |
| Estado de red de dispositivos en base de datos | 17 |
| Estado de red de dispositivos en base de datos | 18 |
| 5. Conclusiones | 25 |
| 5.1. Conclusiones generales | 25 |
| 5.2. Próximos pasos | 26 |

Índice de figuras

| | |
|--|----|
| 4.1. Esquema utilizado para el ensayo del Nrf24l01+. | 8 |
| 4.2. Resultados ensayo del Nrf24l01+. | 9 |
| 4.3. Esquema del ensayo de la biblioteca RF24. | 10 |
| 4.4. Resultados del ensayo de la biblioteca RF24Mesh. | 11 |
| 4.5. pendiente | 12 |
| 4.6. pendiente | 12 |
| 4.7. pendiente | 13 |
| 4.8. pendiente | 13 |
| 4.9. pendiente | 14 |
| 4.10. Resumen de GPIOs de interés para la implementación el sistema | 14 |
| 4.11. Vista 3d del diseño de la placa del dispositivo secundario. | 16 |
| 4.12. pendiente | 16 |
| 4.13. Diagrama CTM para ensayo de estado local | 18 |
| 4.14. Diagrama CTM para ensayo con un dispositivo secundario | 18 |
| 4.15. Diagrama CTM para sistema con dos dispositivos secundarios | 19 |
| 4.16. Interfaz web desarrollada para monitoreo del sistema utilizando la plataforma Node-RED | 19 |
| 4.17. Aplicación móvil desarrollado con la plataforma MIT App Inven- tor, para visualización de datos cargados en el servidor web de Fi- rebase. | 19 |
| 4.18. Ejemplo de histórico de eventos registrados en la base de dato. | 22 |
| 4.19. Ejemplo de la base de datos generada de una red de dispositivos inalámbricos | 22 |
| 4.20. Vista del archivo textitsyslog y los mensajes registrados por siste- ma de monitoreo. | 23 |

Índice de Tablas

| | |
|--|----|
| 4.1. Resultados ensayo biblioteca RF24 | 10 |
| 4.2. GPIOs NodeMCU | 15 |
| 4.3. Casos de prueba, estado local | 17 |
| 4.4. Casos de prueba, dispositivo secundario | 20 |
| 4.5. Continuación de la tabla 4.4 | 21 |

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción general

Capítulo 2

Introducción específica

Capítulo 3

Diseño e implementación

Capítulo 4

Ensayos y Resultados

Este capítulo contiene la descripción de las pruebas realizadas para la validación del sistema. La metodología para esta fase partió desde un nivel micro en módulos funcionales, hasta alcanzar el nivel de pruebas de sistema, y se finalizó con ensayos de inserción de fallas.

4.1. Validación de componentes

Como se explicó anteriormente, los criterios escogidos para la selección de los dispositivos fueron: a) el factor económico, nivel de documentación, c) disponibilidad en el mercado argentino, d) tiempo estimado de desarrollo. Sin embargo, el criterio determinante se basó en los resultados generados en los ensayos que se describen a continuación.

4.1.1. Nrf24l01+

Este ensayo fue realizado como un módulo independiente, con el objetivo de evaluar el desempeño del transceptor nrf24l01 bajo condiciones ideales y su estabilidad al utilizar la biblioteca seleccionada.

La prueba consistió en establecer la comunicación inalámbrica entre dos nodos separados a una distancia de 40 cm. Uno de los nodos mantiene un conteo incremental e intenta transmitir la última lectura al próximo nodo -el cual se mantiene a la espera del mensaje- en caso de que transcurra un tiempo mayor o igual a 200 ms sin recibir un mensaje, se registra un evento de timeout. El resultado se obtiene a partir del porcentaje de mensajes erróneos en relación a los totales enviados, durante un periodo de tiempo de una hora.

Materiales utilizados:

- Nrf24l01 amplificador de potencia con reducción de ruido y antena.
- Nrf24l01 + antena PCB.
- Arduino Uno.
- NodeMCU v3.
- Adafruit Feather HUZZAH ESP8266.
- Raspberry Pi.

- Fuente de poder Canakit micro USB 2,5 A con filtro de ruido.
- Adaptador 120-240 VA a 5V DC USB.

La figura 4.1 muestra un esquema de la metodología empleada para realizar la primera prueba de este ensayo; la Raspberry Pi se seleccionó como el único nodo fijo durante estos ensayos, el resto de dispositivos adoptará sucesivamente el rol de nodo secundario uno a la vez. Los resultados de esta prueba son bastante desalentadores, ya que para cada dispositivo el porcentaje de mensajes exitosos apenas alcanza un 0,5 % en el mejor de los casos.

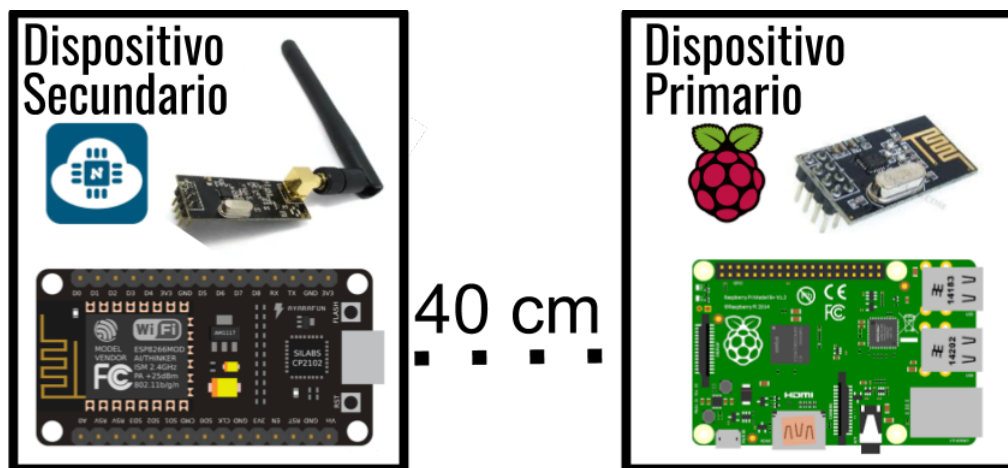


FIGURA 4.1. Esquema utilizado para el ensayo del Nrf24l01+.

Se inició una fase de búsqueda de posibles causas, que inició por una revisión de ejemplos de uso del dispositivo según la documentación de la biblioteca, así como la revisión de pruebas con resultados similares de diferentes desarrolladores y su experiencia reportada. En función de esto se identificó que un elemento clave, es proveer al dispositivo de una fuente de alimentación estable, ya que en algunos casos presentaban fallas similares causadas por fluctuaciones y ruido presente en la alimentación del dispositivo.

Una vez identificada una posible solución, se realizó nuevamente el ensayo reemplazando el adaptador por un equipo de iguales características pero de mejor calidad. Los resultados de esta segunda prueba se consideran satisfactorios y están representados en la figura 4.2, donde se puede apreciar que el menor valor corresponde a un 98 % de éxito en mensajes transmitidos.

Estos resultados evidencian lo susceptible que es el sistema a ruido en la alimentación y la necesidad de una fuente regulada, estable, con el menor ruido posible, es por esto que a partir de este punto se incluye en el diseño del sistema el adaptador del módulo transceptor nrf24l01+. Una característica a resaltar de este adaptador es que imposibilita el uso del nrf24l01+ convencional de antena en circuito impreso, es compatible únicamente con el modelo con amplificador de potencia y reducción de ruido con antena.

4.1.2. Biblioteca RF24

El objetivo de este ensayo es verificar el uso de la biblioteca RF24 para la transmisión de paquetes de información haciendo uso del transceptor nrf24l01+. La

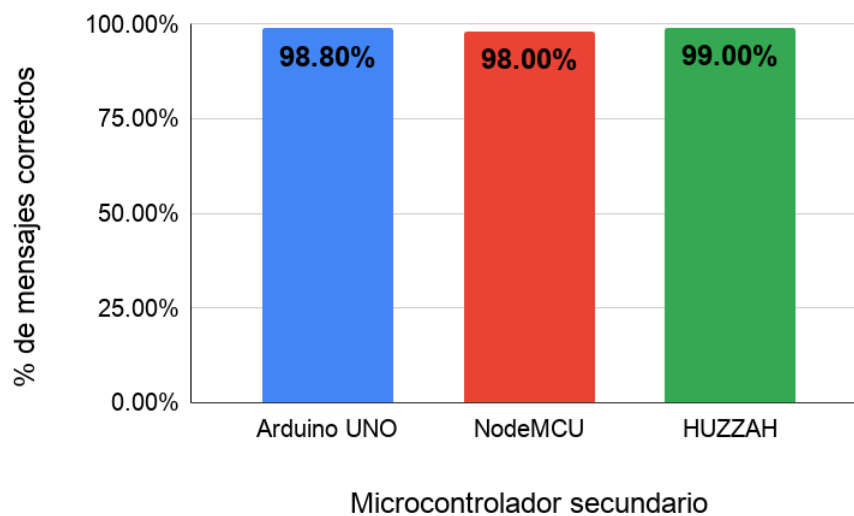


FIGURA 4.2. Resultados ensayo del Nrf24l01+.

disposición de los equipos utilizados para este ensayo consiste en:

- Utilizar la Raspberry Pi junto con un transceptor nrf24l01 + de antena interna como dispositivo primario de ubicación fija.
- Un dispositivo secundario móvil, compuesto por el Arduino UNO y nrf24l01+ de antena interna.

El objetivo de este ensayo fue establecer el alcance de los dispositivos bajo las siguientes condiciones en común:

- Presencia de redes WiFi de diferente intensidad.
- Equipos electrónicos de bajo consumo.

Las condiciones antes mencionadas se mantuvieron a lo largo del ensayo y adicional a esto, se procedió a evaluar la respuesta ante situaciones con los siguientes tipos de obstrucción:

- Obstrucción nula.
- Obstrucción total.
 - Puerta de vidrio
 - Puerta metálica.

El propósito de la prueba es medir la máxima distancia que puede existir entre los nodos primario y secundario, manteniendo una comunicación inalámbrica estable con una tasa de éxito de al menos 90 % de mensajes transmitidos. En la figura 4.3 se puede apreciar la metodología utilizada para el ensayo, inicialmente se parte de una distancia se confirmaba que la comunicación entre los dispositivos era correcta y se procedía mover el dispositivo secundario a una nueva ubicación y repetir nuevamente el proceso de verificación, así hasta alcanzar la máxima distancia a la que era posible establecer la comunicación entre los dispositivos.

Los resultados se muestran en la tabla 4.1, donde se puede apreciar la susceptibilidad del sistema ante la obstrucción de objetos sólidos.

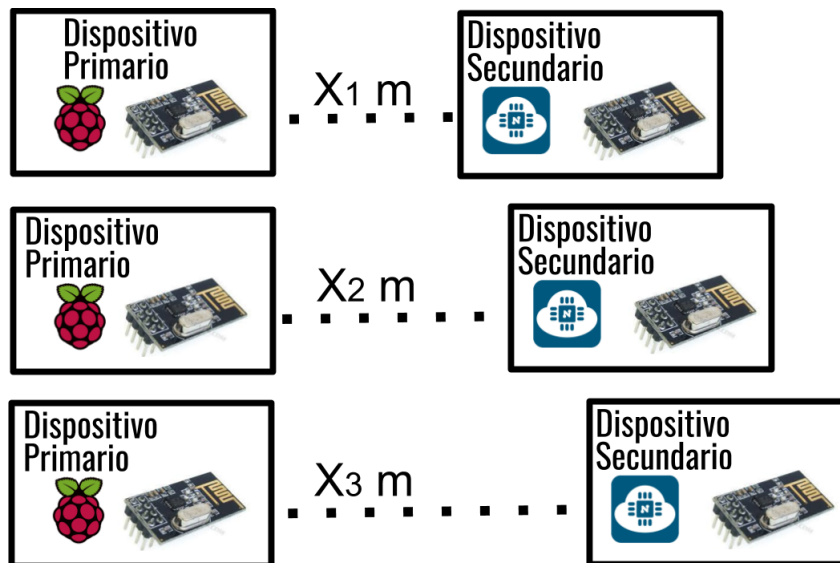


FIGURA 4.3. Esquema del ensayo de la biblioteca RF24.

TABLA 4.1. Resultados de alcance logrado para el ensayo de la biblioteca RF24

| | Sin obstáculos | Obstrucción total | |
|-------------|----------------|-------------------|-----------------|
| | | Ventanal | Puerta metálica |
| Alcance (m) | 8,5 | 8 | 0 |

4.1.3. Biblioteca RF24Mesh

El ensayo anterior establece los valores aproximados de alcance en diferentes condiciones, para este ensayo se requiere validar el funcionamiento de la biblioteca RF24Mesh. El objetivo del ensayo se mantiene, registrar la máxima distancia que se puede alcanzar al separar los dispositivos, manteniendo una comunicación estable. Las condiciones de la prueba son similares al ensayo anterior, consiste en dos nodos en ubicaciones fijas separados una distancia de 7 m, pero esta vez se añade un segundo dispositivo secundario que se utilizará para medir el alcance en esta nueva configuración.

Los resultados del ensayo se pueden observar en la figura 4.4], se obtuvo una distancia máxima de 33 m, al comparar con los resultados del ensayo nrf24l01, se aprecia un incremento en el alcance del sistema, pero esto es debido a las características del dispositivo secundario 2, ya que cuenta con un sistema de amplificación de potencia.

Adicional a la medición del alcance en la nueva configuración de dispositivos inalámbricos, se validan las siguientes funcionalidades de la biblioteca RF24Mesh:

- Gestión automática de dispositivos conectados: el sistema fue capaz de incluir de manera automática al dispositivo secundario 2 sin necesidad de configuraciones adicionales.
- Retransmisión de mensajes: El sistema cuenta con la posibilidad de retransmitir los mensajes entre los nodos hasta alcanzar el nodo destino, esto se

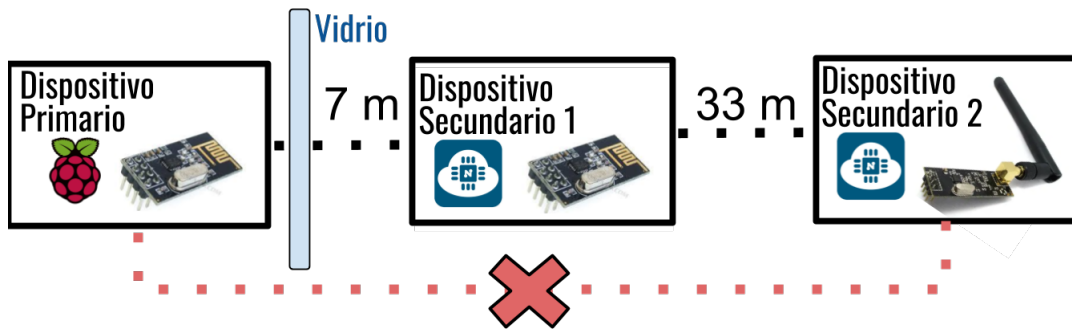


FIGURA 4.4. Resultados del ensayo de la biblioteca RF24Mesh.

comprueba ya que no fue posible establecer la comunicación de forma directa entre el dispositivo secundario 2 y el dispositivo primario, pero al existir el dispositivo secundario 1 en la red, este se encarga de recibir el mensaje del dispositivo secundario 2 y transmitirlo al dispositivo primario.

4.2. Hardware

Partiendo de los requerimientos del sistema, se realizó la fabricación de placas prototipo, para la verificación de los diseños de los módulos que componen el del sistema. A continuación se presentan los resultados obtenidos del diseño y fabricación de hardware.

4.2.1. Prototipo: dispositivo primario.

4.2.2. Prototipo: dispositivo secundario.

El prototipo se basa en los esquemáticos de las figuras 4.9,4.6,4.7, cada uno de estos diseños fue probado mediante simulaciones realizadas, en la plataforma proteus. En la figura 4.8 se presenta el prototipo realizado, a pesar de no ser un dispositivo secundario completo, su elaboración fue muy enriquecedora para el diseño final, influyó en una reducción importante de costos de producción tanto económicos como de tiempo invertido.

inicialmente el objetivo de la fabricación del prototipo era permitir realizar pruebas reales del software durante toda la etapa de desarrollo del trabajo, pero adicional a los aportes de *testing* de software, la elaboración proporcionó la información necesaria para detectar puntos de falla desapercibidos hasta el momento.

A simple vista se puede observar en la figura 4.10 que el NodeMCU cuenta con GPIOs suficientes para la implementación del dispositivo secundario, ya que el número de recursos disponibles supera el número de recursos requeridos. El diseño descrito en el capítulo 3 ejecuta un protocolo de reset automático, como parte de un mecanismo de recuperación ante fallas en la comunicación inalámbrica, eléctricamente el disparador de este mecanismo es un estado digital alto en el pin RST, lo que implica, que es de vital importancia que el pin RST se mantenga en estado digital bajo en todo momento, hasta ser requerido por el algoritmo.



FIGURA 4.5. pendiente



FIGURA 4.6. pendiente

Durante el encendido del NodeMCU, se ejecuta una secuencia de arranque que altera el estado de diferentes GPIOs de forma aleatoria, la tabla 4.2 hace referencia a esta secuencia de arranque y muestra un listado de los pines afectados durante el arranque del chip. Al cruzar la información de la figura 4.10, con los pines que cumplen la condición de estabilidad durante el arranque según la tabla 4.2 podemos inferir que la aseveración realizada anteriormente no es correcta, el número de recursos solicitados es mayor al número de recursos disponibles, lo que hace imposible la implementación del diseño haciendo uso exclusivo de entradas y salidas digitales.

La solución a este problema radica en aprovechar otra característica del NodeMCU, la medición del voltaje se realizará a través del convertidor analógico digital incluido en el microcontrolador, lo que permite compensar el déficit de GPIOs y mantener el diseño intacto, en concreto se utilizará para el monitoreo del estado del contacto seco de falla.

4.2.3. Prototipo: dispositivo secundario.

Al recopilar la información obtenida de la fabricación de los prototipos, el siguiente paso consistió en consolidar todas estas consideraciones en el esquemático del sistema, para dar inicio a la fase de elaboración de dispositivos para la fabricación, las figuras 4.11 e 4.12 presentan el circuito impreso generado en vista 3D y finalmente la placa en su estado final.



FIGURA 4.7. pendiente



FIGURA 4.8. pendiente

4.3. Pruebas unitarias

4.3.1. Estado local.

La tabla 4.3 corresponde a los casos de prueba obtenidos como resultado de aplicar la metodología CTM, apoyándonos en el diagrama de la figura 4.13 como punto de partida.

Al comparar los resultados obtenidos con los resultados esperados, se concluye que el dispositivo primario funciona correctamente, por lo que a partir de la modificación de los contactos de alarma o falla, somos capaces de establecer correctamente el estado del sistema de monitoreo.

4.3.2. Sistema con un dispositivo secundario.

Utilizando la misma metodología del ensayo número uno, se generan la figura 4.14 y la tabla 4.4 y 4.5, que describen el detalle del ensayo. Se confirma que todos los casos de prueba son ejecutados de forma satisfactoria, por lo que hasta este punto se puede concluir, que el dispositivo primario es capaz de comunicarse de forma inalámbrica con un dispositivo secundario, y además puede definir el estado correspondiente del sistema de detección, a partir de su estado propio y el estado del dispositivo secundario.



FIGURA 4.9. pendiente

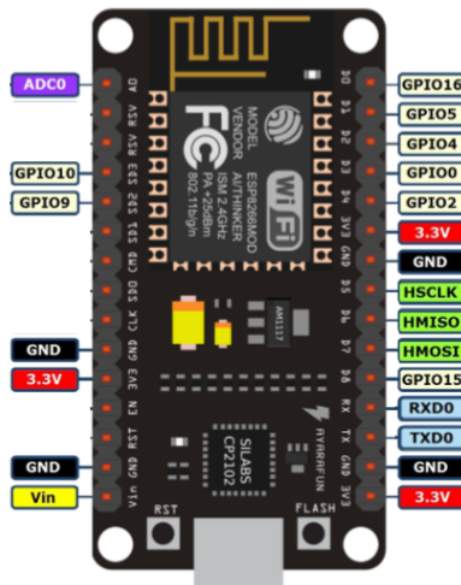


FIGURA 4.10. Resumen de GPIOs de interés para la implementación el sistema

4.3.3. Sistema con dos dispositivos secundarios.

Se mantiene la estrategia para el diseño de casos de prueba y se obtiene como resultado la figura 4.15, pero a diferencia del ensayo anterior, para este ensayo se incluye un nodo que permanecerá en estado de alarma durante todos los casos de prueba. Los resultados en esta ocasión son negativos, ya que ninguno de los casos de prueba es cumplido.

Los casos de prueba fallan debido a que el sistema es incapaz de responder de forma correcta en un tiempo menor a diez segundos, analizando el algoritmo de gestión de comunicación inalámbrica en el dispositivo primario y compararlo con el algoritmo de transmisión de paquetes, se descubrió que el dispositivo secundario está configurado para la transmisión de información cada 15 ms y el dispositivo primario disponía de 10 ms como ventana de tiempo para hacer el análisis del sistema, lo que genera un comportamiento inestable en el sistema, al no poder asegurar que el sistema reciba la información de todos los nodos que conforman la red en una ventana tan corta de tiempo. Un problema adicional que se presentó por la lectura inadecuada de los mensajes recibidos por el transceptor, fue la acumulación de mensajes, lo que ocasiona retrasos en la respuesta del sistema, que se agravaba a medida que el sistema se mantuvo en funcionamiento.

Al identificar la causa del problema, se planteó una modificación de los tiempos

TABLA 4.2. Descripción de pines durante arranque de NodeMCU

| Etiqueta | GPIO | Input | Output | Secuencia de arranque |
|----------|------|--------------------|--------------|--|
| D0 | 16 | Sin problema | Sin problema | Alto al arranque |
| D1 | 5 | Sin problema | Sin problema | - |
| D2 | 4 | Sin problema | Sin problema | - |
| D3 | 0 | Conexión pull up | Sin problema | El arranque falla si se encuentra en estado bajo |
| D4 | 2 | Conexión pull up | OK | Alto al arranque |
| D5 | 14 | SPI | SPI | - |
| D6 | 12 | SPI | SPI | - |
| D7 | 13 | SPI | SPI | - |
| D8 | 15 | Conexión pull down | OK | El arranque falla si se encuentra en estado alto |
| RX | 3 | UART | UART | Alto al arranque |
| TX | 1 | UART | UART | Alto al arranque |
| A0 | ADC0 | Entrada analogica | - | - |

configurados para la comunicación inalámbrica en ambos dispositivos, de manera que se evite la desincronización del sistema. Los nuevos parámetros establecidos para la comunicación, son 200 ms para el dispositivo primario y 30 ms para el dispositivo secundario, esta nueva configuración permitió cumplir correctamente los casos de prueba.

Un punto de mejora que resulta de la realización de este ensayo, es la posibilidad de incluir una interfaz que permita facilitar la inclusión de dispositivos al sistema, ya que la metodología actual de trabajo, hace imposible al usuario incluir un nodo sin conocer el funcionamiento del código y su estructura.

4.3.4. Sistema con tres dispositivos secundarios.

Basados en la información recopilada de los ensayos previos, se propone elaborar un nuevo ensayo con un nodo adicional, pero esta vez con la intención de validar si la configuración de tiempo seleccionada es adecuada. El ensayo incluye de forma similar al ensayo número tres un nodo con un estado de sistema estático, con la diferencia de que en este caso el nuevo nodo permanecerá en estado normal, durante todos los casos de prueba, con el propósito de poder validar la velocidad de respuesta del sistema. El ensayo logró confirmar que el sistema funciona correctamente con los parámetros fijados en el ensayo con dos dispositivos secundarios.

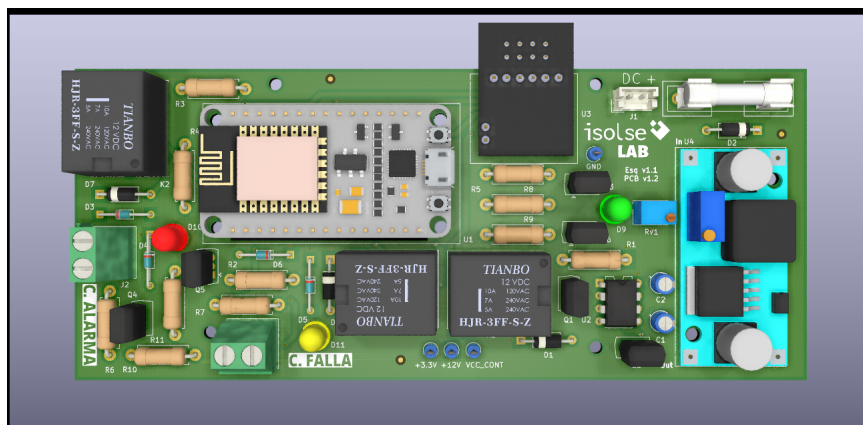


FIGURA 4.11. Vista 3d del diseño de la placa del dispositivo secundario.



FIGURA 4.12. pendiente

4.4. Prueba de sistema

Los resultados obtenidos hasta este punto manifestaron errores y segmentos de código que debían ser atendidos, por lo que la finalidad de este ensayo es evaluar el impacto de los cambios realizados. Los casos de prueba de este sistema serán los descritos en las tablas 4.4 y 4.5, correspondientes al ensayo con un dispositivo secundario, con la diferencia de que en esta ocasión, se incluyen las siguientes características al sistema:

Interfaz web

Interfaz web para la visualización del estado actual del sistema, desarrollado en la plataforma Node-RED, con la limitación de que debemos encontrarnos conectados a la misma red que el dispositivo primario. Un ejemplo de la misma puede observarse en la la figura 4.16.

Aplicación Android

Aplicación móvil para dispositivos Android, conectada a la plataforma de Firebase para el monitoreo del sistema desde cualquier ubicación con conexión a internet, en la 4.17 se muestra una captura de pantalla de la interfaz.

TABLA 4.3. Casos de prueba del ensayo de sistema con un dispositivo primario

| Caso de prueba | Parámetro | Valor | Comentario | Resultado esperado |
|----------------|--------------------|---------|---|---------------------------------------|
| 1 | Contacto de alarma | Cerrado | Sistema monitoreado en estado de alarma y falla, al menos una alarma y una falla están presentes en el sistema. | Led: rojo-amarillo |
| | Contacto de falla | Cerrado | | Estado RF: ok Estado: alarma-falla |
| 2 | Contacto de alarma | Abierto | Sistema monitoreado en estado normal, sin fallas ni alarmas presentes. | Led: verde |
| | Contacto de falla | Abierto | | Estado RF: ok Estado: normal |
| 3 | Contacto de alarma | Cerrado | Sistema monitoreado en estado de alarma, al menos una alarma está presente en el sistema. | Led: rojo |
| | Contacto de falla | Abierto | | Estado RF: ok Estado: alarma |
| 4 | Contacto de alarma | Abierto | Sistema monitoreado en estado de falla, al menos una falla está presente en el sistema. | Led: amarillo |
| | Contacto de falla | Cerrado | | Estado RF: ok Estado: falla |

Histórico de eventos como base de datos

Registro de eventos en bases de datos relacionales, esta funcionalidad permite al sistema registrar de forma ordenada los eventos y la posibilidad de analizar el histórico de eventos del sistema con el uso del lenguaje SQL. La figura 4.18 corresponde a un ejemplo de la base de datos generada.

Estado de red de dispositivos en base de datos

Registro de nodos conectados al sistema en bases de datos relacionales, similar a la base de datos de registro de eventos, se incluye una base de datos con el registro de los nodos conectados, su estado actual y si se encuentran comunicándose correctamente con el dispositivo primario o no, se puede ver un ejemplo de esta base de datos en la figura 4.19.

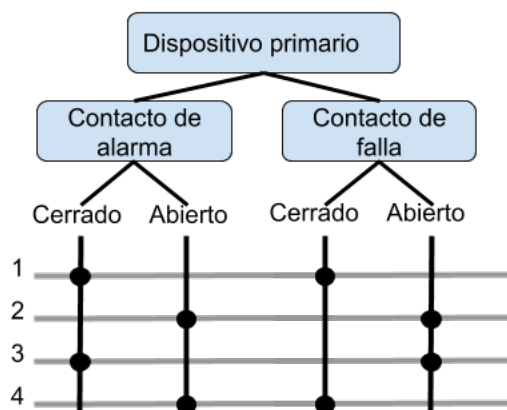


FIGURA 4.13. Diagrama CTM para ensayo de estado local

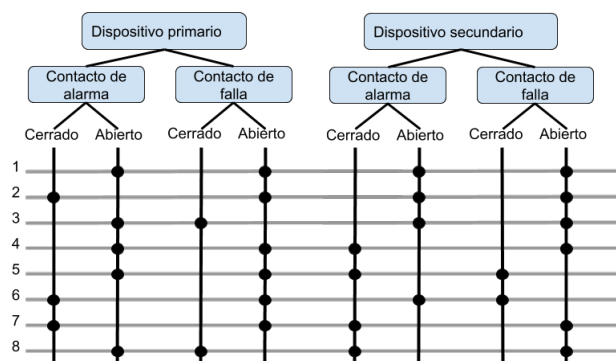


FIGURA 4.14. Diagrama CTM para ensayo con un dispositivo secundario

Estado de red de dispositivos en base de datos

textitLogging gestionado por la biblioteca textitsyslog, se hace uso del recurso del archivo textitsyslog para facilitar el análisis en caso de fallas, registro de tareas de mantenimiento del sistema y uso como herramienta de desarrollo para futuros proyectos. Una vista del textitsyslog del sistema en funcionamiento se puede apreciar en la figura 4.20.

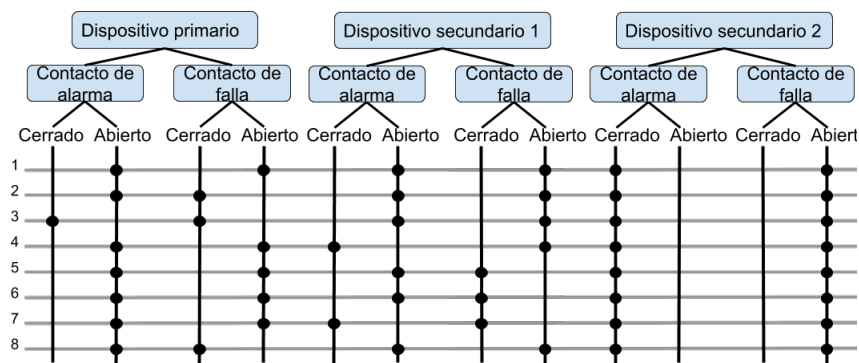


FIGURA 4.15. Diagrama CTM para sistema con dos dispositivos secundarios

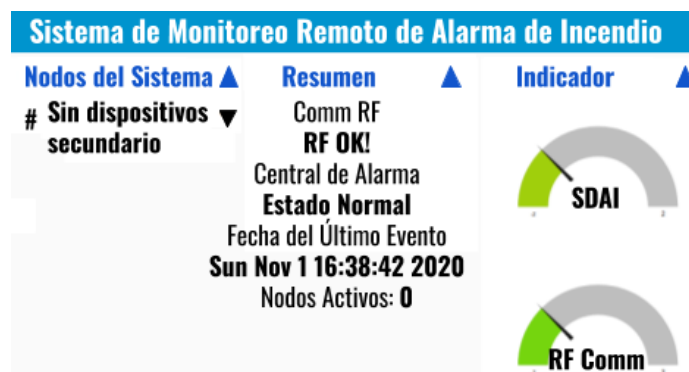


FIGURA 4.16. Interfaz web desarrollada para monitoreo del sistema utilizando la plataforma Node-RED



FIGURA 4.17. Aplicación móvil desarrollado con la plataforma MIT App Inventor, para visualización de datos cargados en el servidor web de Firebase.

TABLA 4.4. Casos de prueba del ensayo de sistema con un dispositivo primario

| Caso de prueba | Parámetro | Valor | Comentario | Resultado esperado |
|----------------|-------------------------|---------|--|--------------------|
| 1 | D.P. Contacto de alarma | Abierto | Ambos sistemas en estado normal, sin fallas ni alarmas presentes. | Led: verde |
| | D.P. Contacto de falla | Abierto | | Estado RF: ok |
| | D.S. Contacto de alarma | Abierto | | Estado: normal |
| | D.S. Contacto de falla | Abierto | | |
| 2 | D.P. Contacto de alarma | Cerrado | Dispositivo primario: sistema en estado de alarma, al menos una alarma presente en el sistema. | Led: rojo |
| | D.P. Contacto de falla | Abierto | | Estado RF: ok |
| | D.S. Contacto de alarma | Abierto | | Estado: alarma |
| | D.S. Contacto de falla | Abierto | Dispositivo secundario: sistema en estado normal, sin fallas ni alarmas presentes. | |
| 3 | D.P. Contacto de alarma | Abierto | Dispositivo primario: sistema en estado de falla, al menos una falla presente en el sistema. | Led: amarillo |
| | D.P. Contacto de falla | Cerrado | | Estado RF: ok |
| | D.S. Contacto de alarma | Abierto | | Estado: falla |
| | D.S. Contacto de falla | Abierto | Dispositivo secundario: sistema en estado normal, sin fallas ni alarmas presentes. | |
| 4 | D.P. Contacto de alarma | Abierto | Dispositivo primario: sistema en estado normal, sin fallas ni alarmas presentes. | Led: rojo |
| | D.P. Contacto de falla | Abierto | | Estado RF: ok |
| | D.S. Contacto de alarma | Cerrado | | Estado: alarma |
| | D.S. Contacto de falla | Abierto | Dispositivo secundario: sistema en estado de alarma, al menos una alarma presente en el sistema. | |

TABLA 4.5. Casos de prueba del ensayo de sistema con un dispositivo primario

| Caso de prueba | Parámetro | Valor | Comentario | Resultado esperado |
|----------------|-------------------------|---------|---|-------------------------------------|
| 5 | D.P. Contacto de alarma | Abierto | Dispositivo primario: sistema en estado normal, sin fallas ni alarmas presentes. | Led: rojo-amarillo Estado RF: ok |
| | D.P. Contacto de falla | Abierto | | |
| | D.S. Contacto de alarma | Cerrado | Dispositivo secundario: sistema en estado de alarma y falla, al menos una alarma y una falla están presentes en el sistema. | Estado: alarma-falla |
| | D.S. Contacto de falla | Cerrado | | |
| 6 | D.P. Contacto de alarma | Cerrado | Dispositivo primario: sistema en estado de alarma, al menos una alarma presente en el sistema. | Led: rojo-amarillo Estado RF: ok |
| | D.P. Contacto de falla | Abierto | | |
| | D.S. Contacto de alarma | Abierto | Dispositivo secundario: sistema en estado de falla, al menos una falla presente en el sistema. | Estado: alarma-falla |
| | D.S. Contacto de falla | Cerrado | | |
| 7 | D.P. Contacto de alarma | Cerrado | Dispositivo primario: sistema en estado de alarma, al menos una alarma presente en el sistema. | Led: rojo Estado RF: ok |
| | D.P. Contacto de falla | Abierto | | |
| | D.S. Contacto de alarma | Cerrado | Dispositivo secundario: sistema en estado de alarma, al menos una alarma presente en el sistema. | Estado: alarma |
| | D.S. Contacto de falla | Abierto | | |
| 8 | D.P. Contacto de alarma | Abierto | Dispositivo primario: sistema en estado de falla, al menos una falla presente en el sistema. | Led: rojo-amarillo Estado RF: ok |
| | D.P. Contacto de falla | Cerrado | | |
| | D.S. Contacto de alarma | Cerrado | Dispositivo secundario: sistema en estado de alarma. | Estado: alarma-falla |
| | D.S. Contacto de falla | Abierto | | |

Table: EVENTS

| ID | Date | State | RF_State | Nodes |
|--------|---------------|------------|----------|-----------|
| Filter | Filter | Filter | Filter | Filter |
| 1 1 | Sun Nov 1 ... | ALARM_FAIL | COMM OK! | ACTIVES N |
| 2 2 | Sun Nov 1 ... | FAIL | COMM OK! | ACTIVES N |
| 3 3 | Sun Nov 1 ... | ALARM_FAIL | COMM OK! | ACTIVES N |
| 4 4 | Sun Nov 1 ... | FAIL | COMM OK! | ACTIVES N |
| 5 5 | Sun Nov 1 ... | ALARM_FAIL | COMM OK! | ACTIVES N |

1 - 5 of 5

Go to: 1

FIGURA 4.18. Ejemplo de histórico de eventos registrados en la base de dato.

Table: RF_NODES

| ROW | ID | CODE | STATUS |
|--------|--------|--------|--------|
| Filter | Filter | Filter | Filter |
| 1 1 | 5 | 654 | 1 |
| 2 2 | 37 | 987 | 1 |

1 - 2 of 2

Go to: 1

FIGURA 4.19. Ejemplo de la base de datos generada de una red de dispositivos inalámbricos .

```
pi@raspberrypi: ~
File Edit Tabs Help
Nov 1 17:17:30 raspberrypi Fire_Alarm_Monitor[1154]: Fire Monitor System: Timeout Warning
Nov 1 17:17:34 raspberrypi Fire_Alarm_Monitor[1154]: Fire Monitor System: Timeout Warning
Nov 1 17:18:33 raspberrypi Fire_Alarm_Monitor[1154]: Fail Status
Nov 1 17:18:34 raspberrypi Fire_Alarm_Monitor[1154]: Fail Status
Nov 1 17:18:39 raspberrypi Fire_Alarm_Monitor[1154]: Fire Monitor System: Timeout Warning
Nov 1 17:18:42 raspberrypi Fire_Alarm_Monitor[1154]: Fail Status
Nov 1 17:18:42 raspberrypi Fire_Alarm_Monitor[1154]: Fail Status
Nov 1 17:20:41 raspberrypi Fire_Alarm_Monitor[1154]: RF Reset
Nov 1 17:20:47 raspberrypi Fire_Alarm_Monitor[1781]: Fire Monitor System: ON
Nov 1 17:20:51 raspberrypi Fire_Alarm_Monitor[1781]: Alarm-Fail Status
Nov 1 17:20:52 raspberrypi Fire_Alarm_Monitor[1781]: Alarm-Fail Status
Nov 1 17:22:30 raspberrypi Fire_Alarm_Monitor[1781]: Fail Status
Nov 1 17:22:36 raspberrypi Fire_Alarm_Monitor[1781]: Alarm-Fail Status
Nov 1 17:22:37 raspberrypi Fire_Alarm_Monitor[1781]: Alarm-Fail Status
Nov 1 17:22:41 raspberrypi Fire_Alarm_Monitor[1781]: Fail Status
Nov 1 17:22:42 raspberrypi Fire_Alarm_Monitor[1781]: Alarm-Fail Status
Nov 1 17:22:43 raspberrypi Fire_Alarm_Monitor[1781]: Alarm-Fail Status
Nov 1 17:30:26 raspberrypi Fire_Alarm_Monitor[1781]: Fire Monitor System: Timeout Warning
```

FIGURA 4.20. Vista del archivo textitsyslog y los mensajes registrados por sistema de monitoreo.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

El trabajo finaliza con el desarrollo de un sistema de monitoreo remoto, compatible con sistemas de detección que funcionen con contactos secos programables, además cuenta con la capacidad de poder establecer comunicación inalámbrica con diferentes dispositivos, en una estructura de red de nodos repetidores. Cada nodo ejecuta un protocolo de recuperación ante pérdidas de comunicación, que les permite restaurarse de forma automática ante diferentes eventos de falla, evitando en la gran mayoría de ocasiones el traslado de personal técnico al lugar de la instalación.

Este sistema fue puesto a prueba en obra para validar su correcto funcionamiento, se utilizaron dispositivos con acceso web y se comprobó que el usuario pudiese visualizar el estado actual de su sistema de seguridad de forma remota.

Se resaltan las prácticas que fueron clave para cada etapa del proyecto:

- Planificación: aunque la planificación no se ejecutó según lo planteado originalmente, se considera una herramienta indispensable, ya que permite monitorear el progreso de las diferentes actividades que integran el proyecto, así como identificar retrasos y/o tareas que requieren mayor dedicación, lo cual permite tomar las acciones necesarias en el tiempo adecuado para cumplir con los objetivos planteados.
- Diseño: utilizar un patrón de diseño, combinado con una estrategia de desarrollo de software modularizado, permitió generar una estructura de código enfocado principalmente en escalabilidad. Este diseño evolucionó en una metodología de desarrollo, fundamentado en la elaboración de pequeños bloques funcionales probados con elementos de *testing*. Lo cual facilitó la inclusión de los conocimientos adquiridos a lo largo de la carrera de forma organizada.
- Software: a partir de las experiencias previas con sistemas operativos, fue posible realizar dos dispositivos funcionales, basados en sistemas operativos diferentes. El dispositivo primario aprovecha la gestión de un sistema operativo de propósito general para establecer una interfaz web bidireccional, asimismo el dispositivo secundario utiliza un sistema operativo cooperativo, que brinda compatibilidad con diferentes microcontroladores y a la vez escalabilidad al proyecto.

- **Hardware:** La fabricación de un dispositivo prototipo, complementa la metodología de desarrollo de software explicada anteriormente, generando una plataforma de prueba que permitió la elaboración de un sistema de pruebas continuas a medida que el sistema se iba concretando. Un aporte invaluable para el trabajo fueron los conocimientos adquiridos en el curso de diseño de circuitos impresos, donde se realizaron cambios al diseño original, que se ven reflejados en mejoras en las características de robustez, manufacturabilidad, detección de fallas y la generación de documentación adecuada para el mantenimiento y evolución del proyecto.

5.2. Próximos pasos

El sistema actual cumple con los requerimientos planteados inicialmente. Sin embargo, durante su desarrollo surgieron puntos de mejora y trabajos a desarrollar, entre los cuales se resaltan los siguientes:

- Aprovechar la conectividad WiFi del dispositivo secundario, para establecer una vía alterna para la carga del estado del sistema en el servidor web, en caso de fallas en el dispositivo primario.
- Desarrollo de una interfaz web personalizada, que complemente los avances realizados en la plataforma Node-RED.
- Diseño de un tercer dispositivo que funcione como nodo repetidor, que prescinda de funcionalidades de adquisición de información y se enfoque en extender el alcance de la comunicación por radiofrecuencia.
- Implementación de un servidor que regule el acceso de los usuarios, que además permita visualizar la información desde cualquier punto con conectividad a internet.
- Elaboración de aplicaciones Android e IOS con la capacidad de generar notificaciones, ante el arribo de nuevos eventos.

El sistema actual se basa en el monitoreo de contactos secos programables, con el objetivo de mantener compatibilidad con diferentes marcas comerciales. A la vez existen segmentos del mercado con requerimientos más exigentes. Algunos posibles proyectos son:

- Hacer uso del puerto de conexión de impresoras de algunas marcas comerciales de centrales de alarma de incendio, para obtener un mayor detalle de los eventos que ocurren en la instalación.
- Diseñar una interfaz gráfica que se adecue a cada obra, facilitando al usuario la documentación de la instalación, tal como información de los equipos instalados, instrucciones ante diferentes eventos, planos con el estado de real de los dispositivos, datos históricos, próximos mantenimientos, etc.

Un aporte significativo que se espera poder alcanzar con esta serie de proyectos, consiste en establecer un sistema con la robustez y confiabilidad suficiente, para instaurar un vínculo con personal capacitado. Lo cual puede favorecer a generar una respuesta temprana ante situaciones de incendio, disminuyendo los daños ocasionados y salvaguardando de manera eficaz la vida de las personas en situaciones de emergencia.