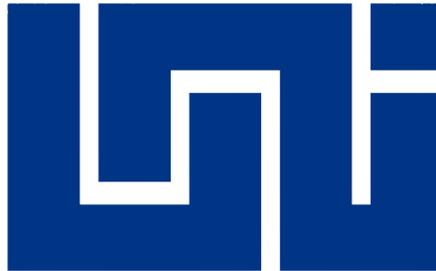


**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN
DEPARTAMENTO DE ELECTRÓNICA**



“Diseño de Prototipo de Estación de Monitoreo de hidrantes”

**TRABAJO MONOGRÁFICO PARA OPTAR EL TÍTULO DE INGENIERO
ELECTRÓNICO**

PRESENTADO POR:

Br. Ernesto de Jesús Mendoza Vallecillo

Br. Helmut Antonio Saavedra García

Tutor:

Ing. Marco Antonio Munguía Mena.

Managua - Nicaragua.

2023.

DEDICATORIA

Dedicamos este proyecto a nuestros padres, quiénes siempre nos han brindado su apoyo incondicional en nuestros proyectos, siendo este uno de los más importantes. A nuestros colegas que nos han acompañado en este camino de aprendizaje, con quienes compartimos el trayecto de ser jóvenes estudiantes a profesionales con criterio y calidad. El trabajo en equipo no siempre es fácil, pero gracias a nuestro compañerismo y dedicación, hemos logrado llevar a cabo este proyecto. Ambos nos hemos esforzado al máximo para que este trabajo refleje lo que hemos aprendido a lo largo de esta carrera y para que cumpla con los objetivos planteados. Agradecemos profundamente a nuestras familias, amigos y profesores que han estado allí para nosotros, brindándonos su ayuda y ánimo en todo momento. Gracias a ellos hemos podido superar los momentos difíciles y disfrutar de los éxitos. También agradecemos al profesor Marcos Munguía por haber desempeñado una excelente labor como nuestro tutor en esta monografía, ya que nos sirvió de guía para poder desarrollar y culminar este proyecto. Por último y no menos importante, dedicamos este proyecto a Nicaragua, esperando que sirva como un aporte al desarrollo tecnológico del país y de esta manera ofrecemos nuestro aporte para que podamos construir un mejor país.

RESUMEN

En este trabajo monográfico se presenta el Prototipo de una Estación de Monitoreo de Hidrantes para dar respuesta a la problemática actual de la falta de conocimiento sobre el estado actual de cada hidrante en la ciudad de Managua, por parte de las entidades estatales y privadas encargadas de gestionar y hacer uso de los hidrantes lo que conlleva a no tener hidrantes disponibles a la hora que los bomberos deban atender una emergencia. El proceso de diseño y desarrollo de este prototipo inició con la recopilación de información por medio de entrevistas a ENACAL, obteniéndose así, los requerimientos técnicos y funcionales del sistema a desarrollar. Dentro de estos requerimientos se pueden destacar el medir la presión de agua de los hidrantes, verificar daños físicos en el hidrante y crear una página web que muestre un mapa con la ubicación de los hidrantes y los valores de los parámetros medidos; así como alertar en caso de falla de algún hidrante. Posteriormente, se procedió a la selección de los componentes de Hardware y Software necesarios para la creación de este Prototipo de Estación de Monitoreo de Hidrantes considerando los requerimientos técnicos y funcionales. También se tomó en cuenta para esta selección factores económicos (precio) y la disponibilidad en el mercado local de los componentes a utilizar. Luego, se realizó la integración de estos componentes en el sistema para obtener una primera versión de este prototipo. Además, se desarrolló la interfaz web, por medio de la cuál, un operario en cualquier lugar con conexión a Internet puede obtener la información en tiempo real del estado de los hidrantes monitoreados y recibir alertas en caso de falla de alguno de ellos. Finalmente, en un primer momento, se realizaron pruebas a escala de laboratorio para verificar y validar el funcionamiento del sistema. Posteriormente, se diseñó un prototipo de hidrante usando dos bombas de agua, una de 1/2 HP (Caballos de fuerza) y otra de 1 HP, con las que se logró simular las variables a considerar en la estación de monitoreo y obtener el valor mínimo de presión de agua con el cuál pueden funcionar los hidrantes.

ABSTRACT

In this monographic work, the Prototype of a Hydrant Monitoring Station is presented to respond to the current problem of the lack of knowledge about the current state of each hydrant in the city of Managua, by the state and private entities in charge of manage and use of hydrants. The design and development process of this prototype began with the collection of information through interviews with ENACAL, thus obtaining the technical and functional requirements of the system to be developed. Within these requirements we can highlight measuring the water pressure of the hydrants, verifying physical damage to the hydrant and creating a web page that shows a map with the location of the hydrants and the values of the measured parameters; as well as alert in case of failure of any hydrant. Subsequently, the selection of the Hardware and Software components necessary for the creation of this Hydrant Monitoring Station Prototype was carried out considering the technical and functional requirements. Economic factors (price) and the availability in the local market of the components to be used were also taken into account for this selection. Then, the integration of these components in the system was carried out to obtain a first version of this prototype. In addition, the web interface was developed, through which an operator anywhere with an Internet connection can obtain real-time information on the status of the monitored hydrants and receive alerts in case of failure of any of them. Finally, initially, tests were carried out on a laboratory scale to verify and validate the operation of the system. Subsequently, a hydrant prototype was designed using two water pumps, one of 1/2 HP (Horsepower) and another of 1 HP, with which it was possible to simulate the variables to be considered in the monitoring station and obtain the value minimum water pressure with which the hydrants can work.

TABLA DE CONTENIDO

DEDICATORIA	2
RESUMEN	3
ABSTRACT	4
TABLA DE CONTENIDO	5
INTRODUCCIÓN	1
1. OBJETIVOS	2
1.1. OBJETIVO GENERAL	2
1.2. OBJETIVOS ESPECÍFICOS	2
2. JUSTIFICACIÓN	3
3. Requerimientos técnicos y funcionales del Sistema de monitoreo	4
3.1. Hidrantes	4
3.2. Parámetros a medir en los hidrantes	5
3.3. Requerimientos Funcionales y No Funcionales	6
3.4. Hardware	8
3.4.1. Microcontrolador	8
3.4.1.1. Arduino UNO	8
3.4.1.2. Microcontroladores PIC	11
3.4.2. ESP8266	12
3.4.3. NRFL24	13
3.4.4. Sensor de Presión de agua	14
3.4.5. Contacto Magnético	15
3.5. Software	15
3.5.1. Firebase	15
4. Medición y adquisición de datos del sistema	17
4.1. Estructura del sistema	17
4.2. Etapa de medición	18
4.2.1. Sensor de presión de agua	18
4.2.2. Contacto magnético	18
4.3. Etapa de control y transmisión	19
4.3.1. Microcontrolador Arduino UNO	19
4.3.2. IDE Arduino	20
4.4. Estructura de la Etapa de medición y transmisión de datos	21
5. Comunicación inalámbrica del sistema	22
5.1. Módulo de comunicación NRF24L01	22
5.2. Etapa de recepción	24
5.2.1. Módulo ESP8266	24
5.3. Escenario de la comunicación inalámbrica	25
6. Configuración de la base de datos	26
6.1. Base de datos firebase	26

6.2. Configuración de la base de datos	27
6.3. Estructura de la base de datos	29
7. Desarrollo del servidor web	31
7.1. Sistema web	31
7.2. Protocolo HTTP	32
7.3. Capa de presentación	32
7.3.1. HTML5	32
7.3.2. CSS	33
7.3.3. JAVASCRIPT - AJAX	33
7.3.4. Leaflet	34
7.4. Desarrollo de la aplicación Web	34
7.4.1. Visual Studio Code	34
7.4.2. Etapa de visualización de datos	35
8. Diseño del Prototipo	38
8.1. Estructura del sistema	38
8.1.1. Prototipo de hidrante	38
8.1.2. Medición de presión de agua	41
8.1.3. Detección de daños físicos al hidrante	42
8.1.4. Comunicación inalámbrica	43
8.1.5. Diseño de la interfaz web	45
8.2. Implementación de la solución	46
8.2.1. Estructura del Prototipo de Estación de Monitoreo de Hidrantes	46
8.2.2. Hidrante	47
8.2.3. Etapa de medición	47
8.2.4. Comunicación inalámbrica	49
8.2.5. Etapa de recepción	50
8.2.6. Base de datos	51
8.2.7. Interfaz web	51
9. Conclusiones	55
10. Recomendaciones	56
11. Bibliografía	57
12. Anexos	59
12.1. Anexos 1: Código de Arduino UNO	59
12.2. Anexos 2: Código de ESP8266	63
12.3. Anexos 3: Entrevista realizada a la Empresa Nicaragüense de Acueductos y Alcantarillados (ENACAL)	68
12.4. Anexos 4: Enlace al repositorio de código de página web	69
12.5. Anexos 5: Segunda entrevista realizada a la Empresa Nicaragüense de Acueductos y Alcantarillados (ENACAL)	70
12.6. Anexos 6: Entrevista realizada al Cuerpo de Bomberos	71

INTRODUCCIÓN

Los hidrantes son equipos de suministro de agua útiles en cualquier ciudad, ya que estos son usados principalmente por los bomberos para atender las distintas emergencias que se presenten. En el caso de Nicaragua la empresa que se encarga del control, instalación, reparación y mantenimiento de los hidrantes es ENACAL (Empresa Nicaragüense de Acueductos y Alcantarillados), la cual ocupa el método de las alertas ciudadanas o informes de los bomberos a la hora de identificar los hidrantes en mal estado o en desuso para luego realizar su debida reparación. (Ver Anexos 3).

El método de control de hidrantes utilizado por ENACAL es ineficiente debido a que no se cuenta con información precisa cuando se daña un hidrante, por lo general estos muchas veces se encuentran dañados o fuera de funcionamiento debido a la falta de atención que se les brinda, haciendo más difícil el servicio de atención de emergencias por parte de los bomberos, lo que provoca que se retrasen en su labor de atender las distintas emergencias, sobre todo en los incendios, ya que en estos casos el tiempo de respuesta es vital. Por lo que es necesario que esta empresa cuente con un sistema eficiente que le permita llevar un buen control de los hidrantes para que también los bomberos puedan disponer de los mismos en sus labores y no tener dificultades en este proceso.

Este trabajo monográfico propone un Prototipo de un Sistema de Monitoreo y localización de hidrantes que venga a dar respuesta a esta necesidad. El sistema consistirá de: un prototipo de hidrante conformado por dos bombas de agua para suministrar la presión de agua deseada en un hidrante, un microcontrolador Arduino que se encargará de monitorear los datos provenientes de un sensor de presión de agua para evaluar si el prototipo de hidrante se encuentra en buen funcionamiento. La localización se manejará mediante una página web y una base de datos que guardarán la información de las coordenadas de todos los hidrantes dentro de la ciudad de Managua.

En el documento se abarca el proceso de investigación, diseño y construcción de un prototipo, introduciendo al lector en cada una de las etapas que conforman el prototipo final.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Diseñar un prototipo de estación de monitoreo de hidrantes en el barrio Altagracia de la ciudad de Managua que permita medir y monitorear los niveles de presión de agua de los mismos para determinar el funcionamiento adecuado de los hidrantes.

1.2. OBJETIVOS ESPECÍFICOS

- 1) Determinar los requerimientos técnicos y funcionales del sistema de monitoreo para su buen funcionamiento.
- 2) Realizar un sistema de medición y adquisición de datos que permita la obtención y verificación de los niveles de presión de agua de los hidrantes.
- 3) Establecer un sistema de comunicación inalámbrico para el envío de los datos de presión desde los sensores hacia una base de datos en internet.
- 4) Configurar una base de datos para realizar el almacenamiento y recepción de los datos de la presión y la ubicación de los hidrantes.
- 5) Desarrollar un servidor web como medio para visualizar y consultar por medio de una página web los datos obtenidos para su posterior análisis por el personal encargado.
- 6) Diseñar un prototipo de hidrante para la simulación de las variables que se deben de considerar en la estación de monitoreo.

2. JUSTIFICACIÓN

El principal problema a considerar en este proyecto es la falta de eficiencia de los sistemas actuales utilizados por ENACAL para monitorear el estado de los hidrantes como lo son las alertas ciudadanas y los reportes de los bomberos cuando logran detectar los hidrantes en mal funcionamiento o en desuso; esto debido a que no tienen un control exacto del estado de los hidrantes ni de la disponibilidad de los mismos en las distintas zonas de Managua y tampoco de información precisa y actualizada que les permita saber qué es lo que pasa exactamente a cada hidrante para así poder repararlos debidamente o bien cambiarlos, esto a su vez perjudicando a los bomberos cuando tienen que atender distintas emergencias.

Los principales afectados debido a este problema es la población en general ya que las personas son las que terminan sufriendo las consecuencias como daños materiales o físicos debido al retraso en la acción de los bomberos provocando que a la población no se le brinde un servicio adecuado de parte de los bomberos, también este problema les afecta a los bomberos ya que su institución es la que principalmente hace uso de los hidrantes al momento de atender las emergencias que se presentan como pueden ser los incendios. Al no atender esta problemática los bomberos continuarán presentando dificultades en sus labores al no tener una apropiada planificación y gestión para la atención de los cientos de hidrantes que existen en la capital, estos continuarán en su mayoría en mal estado provocando que el número de hidrantes disponible para el uso de los bomberos sea muy reducido y esto afecte en su tiempo de atención a las emergencias, a su vez que este retraso puede causar una tragedia mayor al no poder socorrer adecuadamente las situaciones que se les presenten.

Por lo tanto el siguiente trabajo propone un diseño de un prototipo de estación de monitoreo y localización de hidrantes, que se encargará de monitorear los datos de presión de cada hidrante. La cual podrá ser visualizada a través de una página web los datos de las coordenadas, la presión y el estado de todos los hidrantes dentro de la ciudad de Managua. Adicionalmente se utilizará un sensor de contacto magnético para informar en caso de que haya robo o daño de la boquilla del hidrante que permita a ENACAL saber que hidrante está funcionando correctamente o no para que ellos puedan dar una respuesta inmediata y así mantener deseablemente el 100% de los hidrantes disponibles en caso de incendio y recordando que los incendios no avisan cuando ocurren y es deber de ENACAL mantener en su totalidad los hidrantes en buen estado.

3. Requerimientos técnicos y funcionales del Sistema de monitoreo

3.1. Hidrantes

Los hidrantes de incendios, conocidos popularmente como bocas de incendios, son aparatos conectados a una red de abastecimiento de agua destinados a suministrar agua en caso de incendio. Las bocas de incendio no tienen mangueras, sino que son llaves de paso para que bomberos o los servicios de emergencia puedan conectar sus mangueras y/o equipos de lucha contra incendios. También sirven para llenar de agua las cisternas de los camiones de bomberos. (Grupo Prointec, 2017).

Los hidrantes de incendios forman parte de la red de agua específica de protección contra incendios, por lo que están conectados directamente a dicha red, ya sea la red de uso público o la de una instalación en concreto. Debido a su naturaleza, los hidrantes suelen estar colocados en el exterior de los edificios, aunque puede haber excepciones. En cualquier caso, los hidrantes de incendios deben estar debidamente señalizados, indicando el diámetro nominal, el número de la norma y el nombre del fabricante. Cabe destacar en este punto que los hidrantes se rigen por una serie de normas que aseguran, por una parte, que el equipo utilizado para la extinción del incendio sea compatible y, por otras, que puedan asegurar una presión de agua mínima de 80 libras por pulgadas cuadradas (PSI) para poder ser utilizados (Anexos 6). Los hidrantes de incendios pueden ser de dos tipos: hidrante de columna e hidrante bajo nivel de tierra (también llamados de arqueta). Los hidrantes de columna pueden ser de columna seca y de columna húmeda:

- Los hidrantes de columna seca son los que se vacían automáticamente después de ser utilizados. De este modo, al no contener agua cuando no es necesario, no tienen riesgo de rotura por heladas. Estos hidrantes también incorporan un sistema que asegura su estanqueidad en caso de rotura por impacto.

- Los hidrantes de columna húmeda son una alternativa más eficiente y eficaz a los hidrantes de columna seca, ya que disponen de todo el mecanismo en la superficie sin las complicaciones que suponen los hidrantes de columna seca. Uno de estos inconvenientes tiene que ver con la reparación, ya que en los hidrantes de columna húmeda no hay que picar el suelo para descubrir el mecanismo, mientras que en los de columna seca sí.

Los hidrantes bajo nivel de tierra o de arqueta son aquellos que permanecen totalmente enterrados, de forma que no dan lugar problemas de espacio, no tienen riesgo de rotura por impacto y están más protegidos de las heladas. Sin embargo, dispone de menores prestaciones de caudal respecto a los hidrantes de columna. Además, su uso requiere más tiempo, ya que hay que localizarlo primero y quitar la tapa después para poder colocar la manguera. En cualquier caso, ofrecen una solución eficaz para núcleos urbanos muy poblados en los que puede haber problemas de espacio.

3.2. Parámetros a medir en los hidrantes

Para que los hidrantes desempeñen un funcionamiento correcto se necesitan evaluar ciertos parámetros medibles como lo son la presión de agua y el flujo de la misma para que no haya inconvenientes a la hora de atender una emergencia.

Presión de agua: La presión es una magnitud física que mide la proyección de la fuerza en dirección perpendicular por unidad de superficie, y sirve para caracterizar cómo se aplica una determinada fuerza resultante sobre una línea. Cuando se sumerge un cuerpo en un fluido como el agua, el fluido ejerce una fuerza perpendicular a la superficie del cuerpo en cada punto de su superficie, esto es debido al choque de las moléculas del fluido con las paredes del cuerpo (y también del recipiente). (Sánchez, 2006).

En el Sistema Internacional de Unidades la presión se mide en una unidad derivada que se denomina pascal (Pa), que es equivalente a una fuerza total de un newton (N) actuando uniformemente sobre un área de un metro cuadrado (m^2). En el sistema anglosajón la presión se mide en libra por pulgada cuadrada (pound per square inch o PSI), que es equivalente a una fuerza total de una libra actuando sobre un área de una pulgada cuadrada.

En este trabajo se trató principalmente con la presión de fluidos, en Física, un fluido es una sustancia que se deforma continuamente (fluye) bajo la aplicación de una tensión tangencial, por muy pequeña que sea; específicamente la presión del agua; que en este caso se trata de la fuerza que ejerce el agua en el área interna del hidrante, la cual como se mencionó anteriormente deberá tener un valor mínimo de 80 PSI.

3.3. Requerimientos Funcionales y No Funcionales

Un **sistema** es un conjunto de elementos interrelacionados entre sí para lograr un mismo objetivo. Los componentes de sistema son:

- **Entradas:** Datos, información, insumos que ingresan al sistema.
- **Procesos:** Cambios que se producen a las entradas para generar salidas, resultados del sistema.
- **Salidas:** Resultados de los procesos realizados en el sistema.

Un requerimiento es una característica que debe incluirse en un sistema para satisfacer alguna necesidad en su implementación y puede consistir ya sea en una forma de captar o procesar datos, producir información, controlar una actividad o dar apoyo a una tarea.

Dentro de los requerimientos principales de un sistema podemos encontrar los requerimientos funcionales y los requerimientos no funcionales.

Los requerimientos funcionales describen las funciones que un sistema debe de realizar, pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Entre los requerimientos funcionales de este proyecto podemos encontrar:

- Almacenar los datos del sensor de presión y contacto magnético en la base de datos No SQL.
- Almacenar información de la ubicación de los hidrantes mediante el uso de coordenadas.
- Visualizar la ubicación de un hidrante en un mapa incrustado en la página web.
- Mostrar la información del estado de un hidrante en tiempo real.

Y en los requerimientos no funcionales se describen las características generales y descripciones del sistema, es decir aquello que no está relacionado con su funcionamiento y operatividad del sistema. En este proyecto se tiene como ejemplo:

- Contar con una interfaz gráfica de usuario (WEB) amigable e intuitiva.
- El sistema debe ser responsivo.
- El sistema tendrá su dominio y su certificado SSL.

La combinación de los requerimientos funcionales y no funcionales es esencial para desarrollar un sistema completo y efectivo en el monitoreo de hidrantes. En este proyecto se hizo un análisis exhaustivo de las necesidades y expectativas de las entidades reguladoras y de los bomberos, para asegurar que se cumplen todas las funciones esenciales del sistema, y se incluyen criterios estéticos que mejoren la

usabilidad del sistema. Para ello, se realizaron entrevistas y consultas con ENACAL y los bomberos, y se documentó en un informe de requerimientos (Anexos 3), y se definieron las funcionalidades necesarias para el sistema, y se especificaron los criterios de calidad estética y de usabilidad. A continuación en la **tabla 1** se describen en detalle los requerimientos del sistema de monitoreo de hidrantes.

Tabla 1: Requerimientos funcionales y no funcionales del prototipo de estación de monitoreo de hidrantes.

Requerimientos	Parámetros
Requerimientos funcionales	<ul style="list-style-type: none"> ● Medir la variable de presión de agua para evaluar si es la adecuada para un hidrante. ● Monitorear el estado físico de un hidrante y así determinar que se encuentre en óptimas condiciones. ● Diseñar una página web donde se puedan visualizar los datos de cada hidrante. ● Almacenar todos los datos medidos en una base de datos.
Requerimientos de interfaz web	<ul style="list-style-type: none"> ● Diseñar una página web que contenga un mapa incrustado en el cuál se pueda visualizar la ubicación de todos los hidrantes. ● Mostrar la información del estado de un hidrante en tiempo real. ● Almacenar información de la ubicación de los hidrantes mediante el uso de coordenadas geográficas.

3.4. Hardware

Los componentes de Hardware considerados a utilizar para desarrollar el Prototipo de Estación de Monitoreo de Hidrantes son los siguientes:

3.4.1. Microcontrolador

El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida. También incluye un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM). Funciona como una mini PC. Su función es la de automatizar procesos y procesar información. (*Microcontrolador - Qué Es Y Para Que Sirve - HeTPro-Tutoriales*, n.d.)

El microcontrolador es un circuito integrado de muy alta escala de integración que contiene las partes funcionales de un computador: (Cucho et al., 2007)

- CPU (Central Processor Unit o Unidad de Procesamiento Central)
- Memorias volátiles (RAM), para datos
- Memorias no volátiles(ROM, PROM, EPROM) para escribir el programa
- Líneas de entrada y salida para comunicarse con el mundo exterior.
- Algunos periféricos (comunicación serial, temporizador, convertidor A/D, etc)

Existe una amplia variedad de microcontroladores en el mercado, por lo que, parte del trabajo de diseñar este prototipo de estación de monitoreo de hidrantes, fue decidir el microcontrolador a utilizar en base a ciertos criterios como pueden ser económicos, disponibilidad, documentación e información disponible acerca del microcontrolador a elegir. Las opciones consideradas se detallan a continuación:

3.4.1.1. Arduino UNO

El Arduino, como se puede ver en la figura 1.1, es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales se escriben con el lenguaje de programación utilizado en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que se conecten se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos.

El tipo de periféricos que se pueden utilizar para enviar datos al microcontrolador depende en gran medida del uso que se esté pensando dar. Pueden ser cámaras para obtener imágenes, teclados para introducir datos, o diferentes tipos de sensores.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.



Figura 1.1. Arduino UNO

En la **tabla 2** podemos ver las ventajas y desventajas a considerar en el uso del microcontrolador Arduino UNO.

Tabla 2: Ventajas y desventajas del microcontrolador Arduino UNO

Ventajas	Desventajas
<ul style="list-style-type: none">● Simplicidad en el proceso de trabajar con microcontroladores.● Tienen bajos costos, por lo que son más accesibles que otras plataformas de microcontroladores.● Multi-Plataforma. El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.● Tiene un entorno de programación simple y directo.● Software ampliable y de código abierto.● Hardware ampliable y de Código abierto.● Debido a su popularidad se puede encontrar mucha información disponible en la web así como documentación.	<ul style="list-style-type: none">● Dado que la programación no se realiza en ensamblador, el precio a pagar por el uso de las librerías es un retraso en la ejecución de las instrucciones.● El hecho de que la plataforma venga ya ensamblada le quita flexibilidad a los proyectos, así por ejemplo se obligaría a usar un espacio y forma acorde con el PCB del arduino.

3.4.1.2. Microcontroladores PIC

Los microcontroladores PIC (conocidos simplemente como PIC), a como se puede ver en la figura 1.2, son circuitos integrados que pueden ser programados para ejecutar una secuencia de instrucciones automáticamente. El programa se escribe en un computador en un ambiente de programación adecuado (por ejemplo en lenguaje C) y luego se transfiere como código binario a la memoria de programa del microcontrolador para su ejecución en el momento de encendido del dispositivo. Los microcontroladores PIC son fabricados por la empresa Microchip y constan básicamente de una CPU, espacios de memoria para el programa y los datos, y

varios periféricos para la interacción con el mundo externo, todo esto encapsulado dentro de un solo chip de silicio. (*Introducción a La Programación De Microcontroladores PIC*, n.d.)

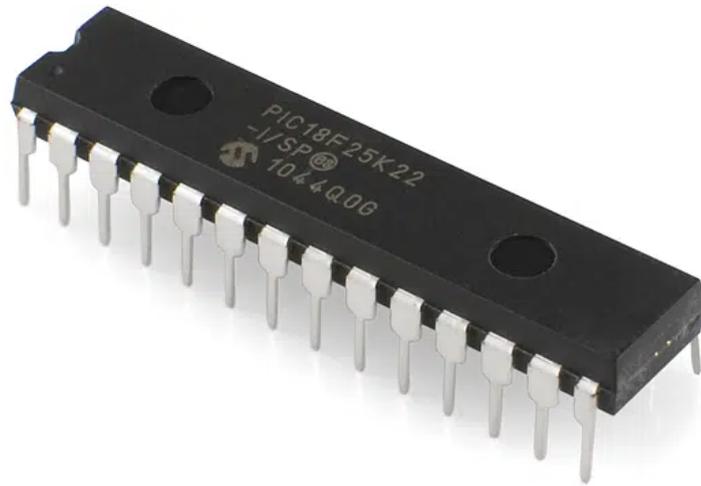


Figura 1.2. Microcontrolador PIC

Existe una amplia variedad de microcontroladores PIC en el mercado y esto permite poder elegir entre diversas características que uno no puede tener pero el otro sí, como cantidad de puertos, cantidad de entradas/salidas, conversión analógico a digital, tamaño de memoria o espacio físico para así tener una mejor elección de acuerdo a nuestras necesidades.

En la **tabla 3** a continuación podemos ver las ventajas y desventajas en el uso de estos microcontroladores.

Tabla 3: Ventajas y Desventajas de los microcontroladores PIC

Ventajas	Desventajas
<ul style="list-style-type: none">• Los microcontroladores PIC son muy fáciles de grabar, ya que solo necesitamos una computadora o una notebook.• En el mercado existen varios softwares que nos ayudan a programar un microcontrolador de este tipo, como por ejemplo el PICC, o el MPLAB.• Existe una gran diversidad de microcontroladores PIC en el mercado de Microchip y ésta también es una gran ventaja, ya que podemos elegir entre diversas características que uno no tiene pero otro si.	<ul style="list-style-type: none">• Una de las desventajas de un microcontrolador PIC es que se necesitan llamar a muchas instrucciones para realizar una tarea en particular.• Otra de las desventajas pero no tan significativas o importantes es que los PIC no son tan baratos como uno los puede esperar.• Tanto el lenguaje de programación como los software de programación de los microcontroladores PIC, suelen ser un poco complejos.

Tomando en cuenta los dos tipos de microcontroladores considerados para este proyecto, se decidió utilizar el microcontrolador Arduino UNO, debido a su fácil disponibilidad en el mercado, también debido a que cuenta con más documentación e información que los microcontroladores PIC y mayor facilidad de conexión de distintos componentes y módulos, sobre todo por tener software y hardware libre.

3.4.2. ESP8266

A como se puede apreciar en la figura 1.3, el ESP8266 es un chip de bajo costo WI-FI con un stack TCP/IP completo y un microcontrolador, fabricado por Espressif, una empresa ubicada en Shanghái, China.

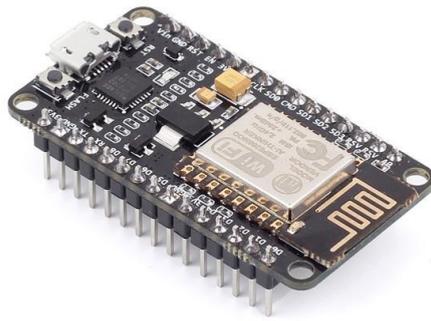


Figura 1.3. Placa de desarrollo ESP8266

Características

- RAM de instrucción de 64 KB, RAM de datos de 96 KB
- Capacidad de memoria externa flash QSPI - 512 KB a 4 MB* (puede soportar hasta 16 MB)
- IEEE 802.11 b/g/n Wi-Fi
 - Soporte de autenticación WEP y WPA/WPA2
- 16 pines GPIO (Entradas/Salidas de propósito general)
- SPI, I²C,
- Interfaz I²S con DMA (comparte pines con GPIO)
- Pines dedicados a UART, más una UART únicamente para transmisión que puede habilitarse a través del pin GPIO2
- 1 conversor ADC de 10-bit

Este módulo se utilizó en el sistema como receptor de los datos transmitidos desde la placa Arduino y los sensores usados y a la vez es el encargado de procesar estos datos hacia la base de datos del servidor.

3.4.3. NRFL24

En la figura 1.4 se observa el módulo transceptor NRF24L01. Utiliza la banda de 2,4 GHz y puede operar con velocidades de transmisión de 250 kbps hasta 2 Mbps. Si se usa en espacios abiertos y con menor velocidad de transmisión, su alcance puede llegar hasta los 100 metros. Para mayores distancias, hasta 1000 metros, existen módulos provistos con una antena externa en lugar de una antena trazada sobre la misma placa. Este módulo fue utilizado para permitir la comunicación vía radiofrecuencia (RF) entre el Arduino Uno y los sensores hacia el módulo ESP8266.



Figura 1.4. Módulo transceptor NRF24L01

3.4.4. Sensor de Presión de agua

Este sensor que se puede ver en la figura 1.5 mide la presión que hay dentro de distintas sustancias líquidas, de tal manera que el valor a medir será el de la presión que ejerce el líquido sobre el sensor. Puede ser usado tanto en agua, aceite y combustible y soporta una presión de hasta 200 PSI. Está hecho de acero inoxidable para garantizar que no sufra ningún daño debido a que se sumergirá en el líquido para realizar las mediciones esperadas.

En este proyecto se ocupó para medir la presión del agua dentro de los hidrantes para verificar que se cumple en todo momento con el valor adecuado y así poder informar si un hidrante está funcionando correctamente. Al mismo tiempo los valores extraídos de este sensor son procesados por medio de la placa Arduino Uno para así ser evaluados dentro del sistema.



Figura 1.5. Sensor de Presión de agua

3.4.5. Contacto Magnético

Un sensor magnético o contacto magnético como el que se puede ver en la figura 1.6, es un elemento que consta de una cápsula de vidrio conteniendo un par de contactos metálicos en su interior y un par de terminales que permiten acceder a conectar dichos contactos. Estos contactos normalmente están eléctricamente aislados el uno del otro. Cuando un campo magnético de la magnitud adecuada se acerca, estos contactos se cierran. Este sensor fue utilizado para informar del robo o daño de las boquillas de los hidrantes, cuando la boquilla se separa de la estructura del hidrante este indicará una alerta de un posible robo o daño.



Figura 1.6. Contacto Magnético

3.5. Software

3.5.1. Firebase

Firebase de Google, a como se puede ver su logo en la figura 1.7, es una plataforma en la nube para el desarrollo de aplicaciones web y móviles. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo. Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Una de las herramientas más destacadas y esenciales de Firebase son las bases de datos en tiempo real. Estas se alojan en la nube, son No SQL y almacenan los datos

como JSON. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados aunque el usuario no realice ninguna acción.

Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos nuevos en el disco. Aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizan una vez restablecida la conexión.

Mediante Firebase se diseñó la base de datos del sistema que maneja los distintos datos utilizados como lo son los valores extraídos de los sensores y las coordenadas de las ubicaciones de los hidrantes.



Figura 1.7. Base de datos Firebase

4. Medición y adquisición de datos del sistema

4.1. Estructura del sistema

La estación de monitoreo de hidrantes está compuesta según el diagrama mostrado en la figura 2.1 que será explicado a continuación:

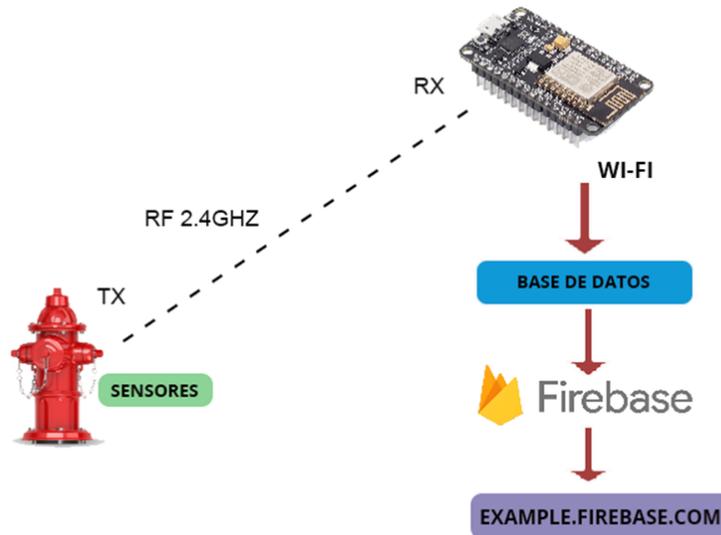


Figura 2.1. Representación del sistema

El proyecto fue distribuido en tres áreas importantes, las cuales son la medición y transmisión de los parámetros o variables medidas, la recepción de los parámetros medidos y la visualización de los datos obtenidos, estos tres campos son los que permiten tener un control total sobre los hidrantes monitoreados y permitir el óptimo funcionamiento de una estación de monitoreo de hidrantes. En la etapa de transmisión se miden los parámetros de interés para luego ser enviados a la etapa de recepción que se encargará de recibir estos datos en una base de datos y finalmente estos se puedan visualizar en una página web para la debida interpretación de parte de las personas encargadas del monitoreo.

4.2. Etapa de medición

4.2.1. Sensor de presión de agua

Para monitorear la presión de agua se utilizó un sensor de presión de agua como el que se puede ver en la figura 2.2, el cual en su conexión electrónica consiste de tres cables, un cable de alimentación positiva (Vdc), un cable de tierra (GND) y un cable de datos que es el encargado de enviar las lecturas del sensor hacia el controlador. Estos cables son conectados a la placa Arduino UNO.



Figura 2.2. Sensor de presión

4.2.2. Contacto magnético

El contacto magnético que se puede apreciar en la figura 2.3 se conecta al Arduino UNO para enviar datos booleanos para indicar los estados de cierre y apertura de este mismo. Este contacto fue utilizado para detectar robo o daños físicos en los hidrantes, ya que al estar el contacto cerrado tendrá un valor de 1 ó True y al estar el contacto abierto tendrá un valor de 0 ó False.



Figura 2.3. Contacto magnético

4.3. Etapa de control y transmisión

4.3.1. Microcontrolador Arduino UNO

El microcontrolador utilizado en este proyecto fue la placa Arduino UNO (ver figura 2.4), la cual es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc. La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos.

En esta placa Arduino UNO fueron conectados el sensor de presión y el contacto magnético, los cuales los datos obtenidos de estos fueron procesados por el Arduino UNO, a su vez también fue conectado el módulo de comunicación NRF24L01 para gestionar los datos que han sido enviados por medio de la etapa de comunicación.



Figura 2.4. Arduino UNO

4.3.2. IDE Arduino

Un IDE (Entorno de Desarrollo Integrado por sus siglas en inglés) es una aplicación de software la cual tiene la función de ser un editor de código fuente para que así se pueda desarrollar código de manera eficiente tanto para software como para hardware. En este proyecto se utilizó el IDE de Arduino para desarrollar el código que llevan tanto la placa Arduino UNO como el módulo ESP8266.

A como se puede apreciar en la figura 2.5, a continuación se muestra parte del código que tendrá el Arduino UNO para controlar tanto al sensor de presión de agua como el contacto magnético y el módulo de comunicación, el código completo se podrá ver en **Anexos 1**.

```
transmisor Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
transmisor
1 /* Comunicación inalámbrica con NRF24L01 Arduino
2 Ejemplo 1 - Programa para el Transmisor
3 Librería:
4 TMRh20/RF24 https://github.com/tmrh20/RF24/ */
5
6 /* Debemos incluir la librería SPI básica y la
7 librería RF24 que ya hemos instalado */
8 #include <SPI.h>
9 #include <RF24.h>
10 // #include <BH1750.h> // incluye librería BH1750
11 #include <Wire.h> // incluye librería para bus I2C
12
13 /* Crear un objeto RF24 al que llamaremos "radio".
14 Los dos argumentos que se utilizan en la función
15 son los números de pin que vamos a usar para CS y CE. */
16 RF24 radio(7, 8); // CE, CSN
17
18 //BH1750 sensor: // crea objeto con nombre sensor
19
20 /* Creamos una matriz de bytes que represente la
21 dirección con la que se reconoceran entre si los
22 dos módulos. Podemos cambiar el valor de esta
23 dirección a cualquier conjunto de 5 letras y/o
24 números. Esta dirección permite elegir con que
25 receptor vamos a comunicarnos. En nuestro caso,
26 usaremos la misma dirección tanto en el receptor
27 como en el transmisor. */
28 const byte identificación[6] = "00001";
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
```

- **Microcontrolador Arduino UNO:** Utilizado como una de las unidades centrales del sistema, el cuál tiene conectado los demás componentes como lo son el sensor de presión de agua, el contacto magnético y el módulo NRF24L01. Este microcontrolador fue usado en cada hidrante a monitorear y recibe los datos provenientes del sensor de presión de agua y del contacto magnético para luego transmitirlos por medio del módulo NRF24L01.
- **Sensor de presión de agua:** Este sensor estará en contacto con el agua de los hidrantes, ya que es el encargado de medir la presión de agua de los mismos para que luego el Arduino UNO reciba estos datos y los procese adecuadamente en el resto del sistema.
- **Contacto magnético:** Este dispositivo envía una señal binaria al Arduino UNO para que luego esta sea interpretada como una alarma en caso que el imán se aleje del contacto magnético, ya que en esta situación dicho fenómeno se puede interpretar como el robo de la boquilla del hidrante o daños físicos al mismo.
- **Módulo NRF24L01:** Este módulo realizó la comunicación entre la etapa de transmisión y recepción del sistema. Está configurado en modo de transmisión en la programación del Arduino UNO para que pueda enviar los datos de los sensores hacia la etapa de recepción.

5. Comunicación inalámbrica del sistema

5.1. Módulo de comunicación NRF24L01

El módulo NRF24L01 mostrado en la figura 3.1, es un módulo de comunicación por radiofrecuencia (RF), el cual consta de 8 pines, GND es el pin de tierra, VCC es el pin de alimentación del dispositivo que funciona con un voltaje entre 1.9V y 3.9V, CE (Chip Enable) es un pin activo-alto encargado de la transmisión y recepción de datos según su configuración, CSN (Chip Select Not) es una clavija activa-BAJA y normalmente se mantiene ALTA. Cuando esta clavija se mantiene baja, el NRF24L01 comienza a escuchar en su puerto SPI los datos y los procesa en consecuencia. SCK (Serial Clock) acepta los pulsos de reloj proporcionados por el maestro del bus SPI. MOSI (Master Out Slave In) es la entrada SPI del NRF24L01.

MISO (Master In Slave Out) es la salida SPI del NRF24L01. Por último IRQ es un pin de interrupción que puede alertar al maestro cuando hay nuevos datos disponibles para procesar. En este proyecto este módulo es el que establece la comunicación entre el Arduino UNO y el módulo ESP8266, se ocuparon dos módulos NRF24L01 debido a que un módulo fue conectado al Arduino UNO para transmitir los datos provenientes de los sensores, y el otro módulo fue conectado al ESP8266 para recibir los datos transmitidos.

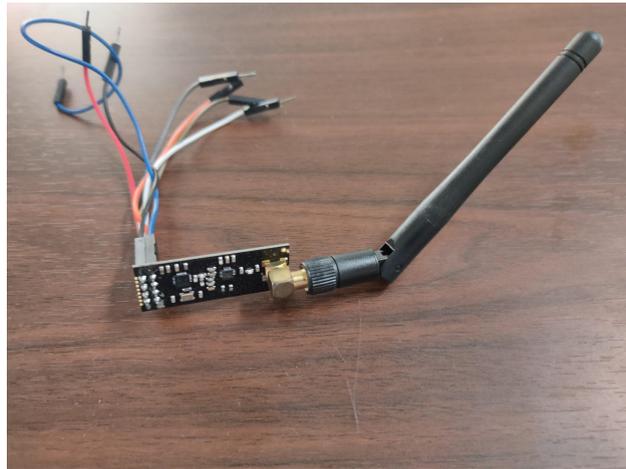


Figura 3.1. Módulo NRF24L01

Dentro de las características principales de este módulo se pueden destacar las siguientes:

- **Frecuencia de Operación:** Opera en la banda de frecuencia de 2.4 GHz, lo que es comúnmente conocido como la banda ISM (Industrial, Scientific, and Medical) utilizada para dispositivos inalámbricos.
- **Potencia de Transmisión:** La potencia de transmisión del NRF24L01 es ajustable, generalmente en pasos de 0.5 dBm. Puede variar desde aproximadamente -18 dBm hasta 0 dBm, dependiendo de la versión específica del módulo. La potencia de transmisión se puede ajustar y la sensibilidad de recepción es alta, lo que permite comunicaciones en distancias moderadas.
- **Sensibilidad de Recepción:** La sensibilidad de recepción, que indica la mínima potencia de señal que puede detectar el módulo, es típicamente en el rango de -85 dBm a -90 dBm.
- **Protocolo de Comunicación:** Utiliza un protocolo propio de comunicación basado en la técnica de modulación GFSK (Gaussian Frequency-Shift Keying).

- **Ancho de Banda:** Tiene varios ajustes de ancho de banda, generalmente de 250 kbps, 1 Mbps y 2 Mbps, que afectan la velocidad de transmisión y la tolerancia a interferencias.
- **Consumo de Energía:** El NRF24L01 es conocido por su eficiencia energética, con modos de bajo consumo de energía para prolongar la vida útil de la batería en sistemas alimentados por energía.
- **Comunicación Multicanal:** Ofrece múltiples canales de comunicación para evitar interferencias con otros dispositivos en la misma banda.

Dentro de las aplicaciones en las que es utilizado este módulo podemos destacar las siguientes:

- Sistemas de seguridad y alarmas
- Comunicación inalámbrica de datos
- Automatización de hogares
- Telemetría
- Automotores
- Equipo deportivo inteligente
- Sensores industriales
- Juguetes

5.2. Etapa de recepción

5.2.1. Módulo ESP8266

El módulo WiFi ESP8266 (ver figura 3.2) es un autocontenedor SOC, con pila integrada protocolo TCP/IP que puede dar acceso a cualquier microcontrolador a su red WiFi. El ESP8266 es capaz de acoger ya sea una aplicación o la descarga de todas las funciones de red Wi-Fi desde otro procesador de aplicaciones. Cada módulo ESP8266 viene pre-programado con un conjunto de firmware de comando AT, es decir, sólo tiene que conectar esto a su dispositivo Arduino y obtener aproximadamente la misma cantidad de WiFi como ofrece WiFi Shield.

Este módulo tiene una potente capacidad suficiente a bordo de procesamiento y almacenamiento que le permite integrarse con los sensores y dispositivos específicos de la aplicación a través de sus GPIO con un desarrollo mínimo por

adelantado y la carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite la circuitería externa mínima, incluido el módulo front-end, está diseñado para ocupar la misma área de PCB.

En este proyecto el módulo ESP8266 tiene conectado directamente el módulo NRF24L01, ya que se encargará de recibir los datos transmitidos por el Arduino UNO y una vez recibidos se procederá a almacenarlos en una base de datos. Ver en **Anexos 2** el código utilizado para este módulo.



Figura 3.2. Módulo ESP8266

5.3. Escenario de la comunicación inalámbrica

El escenario a considerar donde se realizó la comunicación inalámbrica fue en el barrio Altagracia de la ciudad de Managua, considerando un sector en específico en el cuál el monitoreo se realizará en dos hidrantes específicos indicados por dos punteros de color azul a como se puede ver en la figura 3.3.

También se puede observar un recuadro rojo, este indica la ubicación de la estación de monitoreo, esta última se encuentra ubicada a una distancia entre 100 y 200 metros de cada hidrante, siendo posible la comunicación inalámbrica a estas distancias sobre todo usando radiofrecuencia de 2.4GHz que es con la que opera el módulo de comunicación.

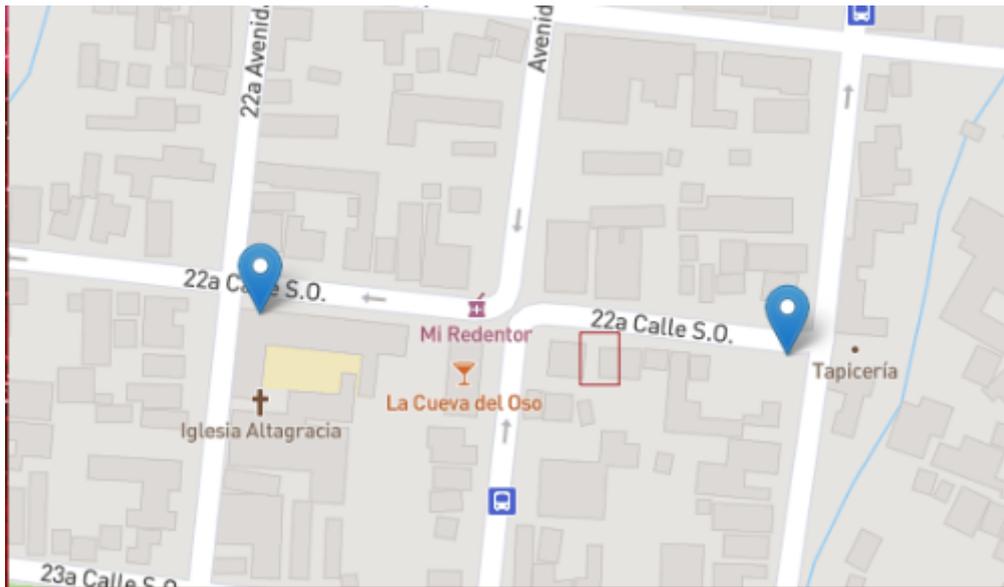


Figura 3.3. Escenario de comunicación inalámbrica

6. Configuración de la base de datos

6.1. Base de datos firebase

Firestore es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014. Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

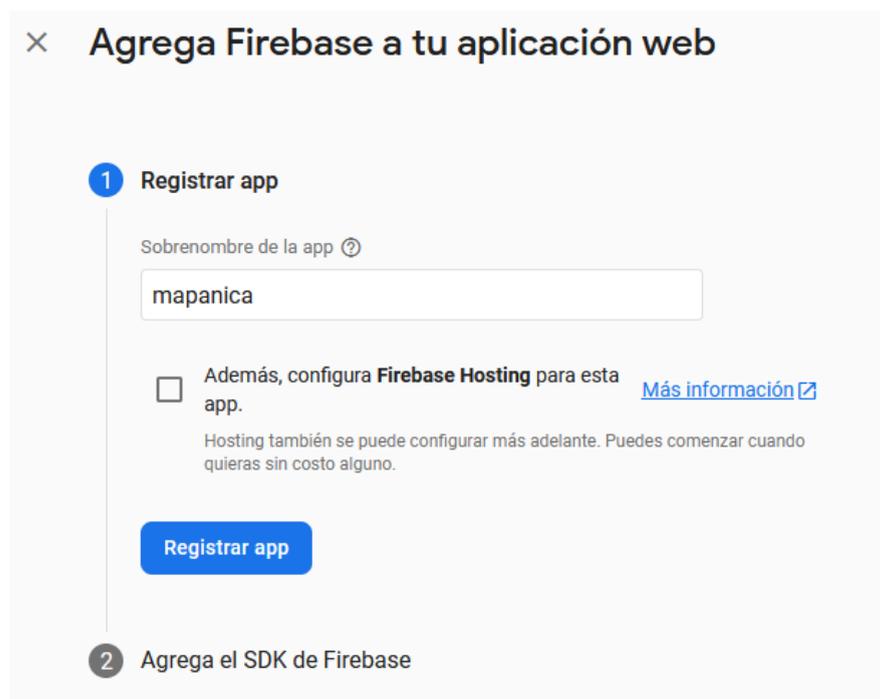
Los desarrolladores tendrán una serie de ventajas al usar esta plataforma:

- Sincronizar fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Usa un conjunto de herramientas multiplataforma: se integra fácilmente tanto en plataformas web como en aplicaciones móviles.
- Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

Firestore es la base de datos utilizada en este proyecto, en esta plataforma se guardan todos los datos provenientes de los hidrantes como pueden ser la presión de agua, la localización y si hay una alarma activada mediante el contacto magnético.

6.2. Configuración de la base de datos

Primeramente se creó una aplicación o proyecto en el cual también nos da la opción de alojarlo en hosting de ellos. Con esto nos estamos registrando en la plataforma de Firebase como se puede apreciar en la figura 4.1.



The screenshot shows a dialog box titled "Agrega Firebase a tu aplicación web" with a close button (X) in the top left corner. The dialog is divided into two steps:

- 1 Registrar app**: This step contains a text input field labeled "Sobrenombre de la app" with a help icon (question mark) to its right. The input field contains the text "mapanica". Below the input field is a checkbox that is currently unchecked. To the right of the checkbox is the text "Además, configura **Firestore** para esta app." followed by a blue link "Más información" with an external link icon. Below this text is a smaller line of text: "Firestore también se puede configurar más adelante. Puedes comenzar cuando quieras sin costo alguno." At the bottom of this step is a blue button labeled "Registrar app".
- 2 Agrega el SDK de Firebase**: This step is partially visible at the bottom of the dialog.

Figura 4.1. Registro de la aplicación

Luego que se logró registrar la aplicación el siguiente paso es crear un SDK (Un kit de desarrollo de software) de firebase con lo cual vamos a utilizar mediante un script, esto lo debemos de colocar en los archivos de html para hacer uso de nuestra base de datos. Lo que contiene este SDK son ciertas configuraciones para poder hacer uso de la base de datos. Como se muestra en la figura 4.2.



Figura 4.2. Añadir el SDK de firebase

Además del SDK hay que decirle a nuestra base de datos en que servidor se va ubicar en este caso se asignó el servidor de Estados Unidos ya que es el más cercano con respecto a la aplicación o solución que se quiere brindar. Como se muestra en la figura 4.3.

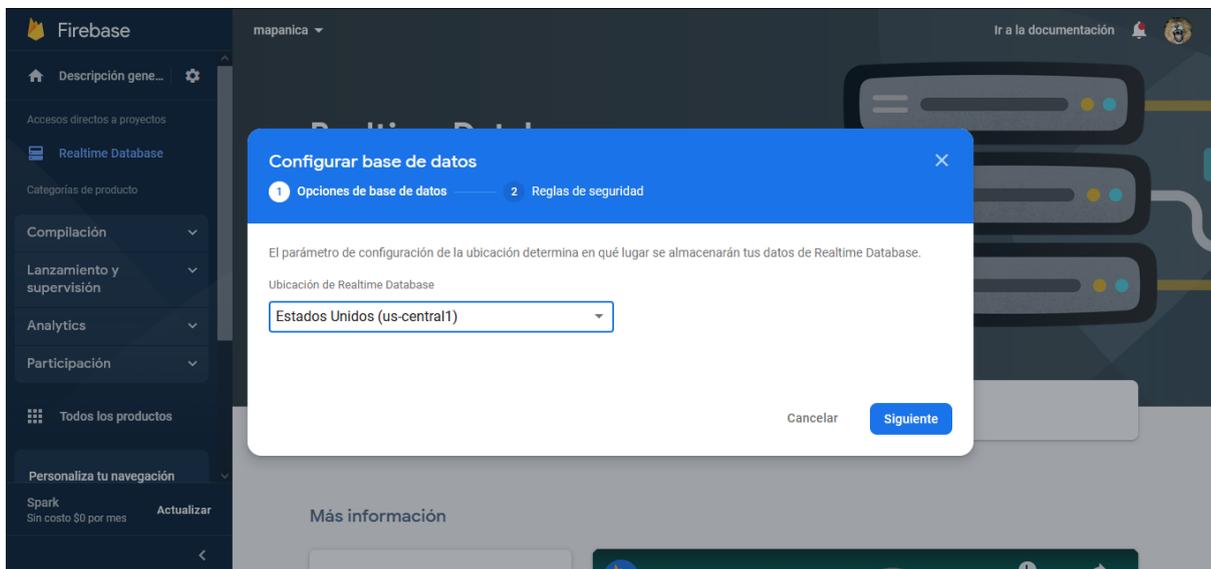


Figura 4.3. Configuración de la base de datos

Luego que se configuró la base de datos, establecemos comunicación con el dispositivo o hardware, en este caso es el ESP8266 que nos sirve como enlace entre los datos que queremos captar y la base de datos. Una vez que se establece la comunicación, se procedió a enviarle los datos que deseamos almacenar en la base de datos como lo son los datos de presión y verificación de la alarma y

mediante un script en el ESP8266 para enviarlos a la base de datos a través de una conexión a Internet como se puede visualizar en la figura 4.4.

Es importante tener en cuenta que el ESP8266 es solo un módulo de conectividad y no tiene la capacidad suficiente de procesar o almacenar datos por sí mismo, por lo que debe trabajar junto con una aplicación o dispositivo externo y una base de datos para llevar a cabo estas tareas.

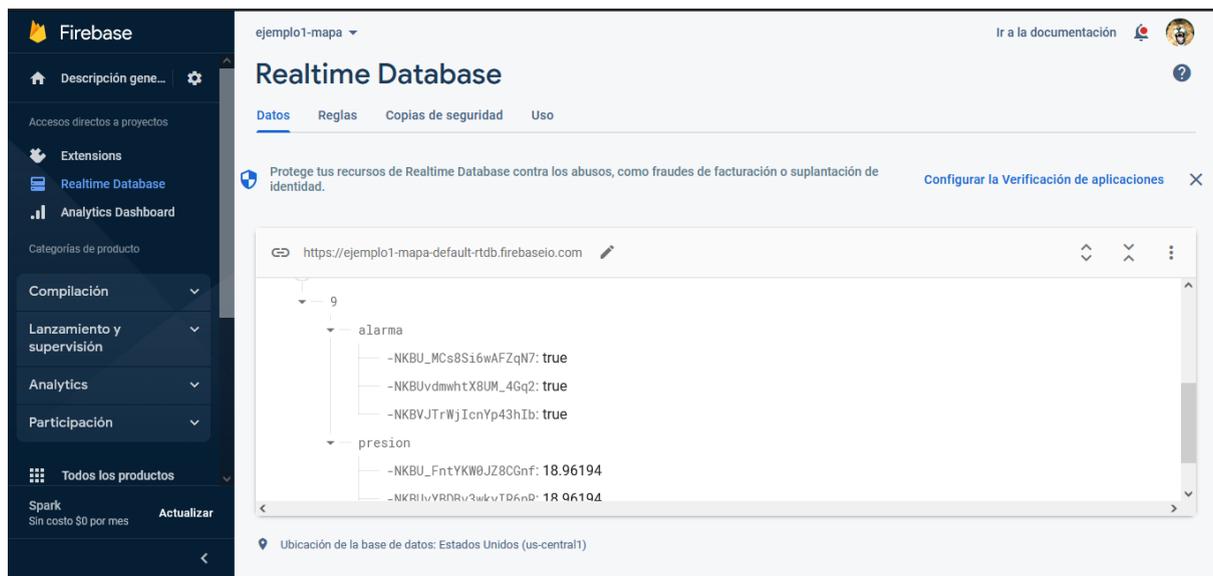


Figura 4.4. Base de datos con los datos recolectados

6.3. Estructura de la base de datos

La estructura de Firebase es mediante una jerarquía de nodos, en la cuál tenemos nodos padres que dan lugar a nodos hijos y todos estos nodos pueden almacenar datos en forma de objetos JSON. La base de datos puede conceptualizarse como un árbol JSON alojado en la nube. A diferencia de una base de datos de SQL, no hay tablas ni registros. Cuando le agregas datos al árbol JSON, estos se convierten en un nodo de la estructura JSON existente con una clave asociada. Puedes entregar tus propias claves, como ID de usuario o nombres semánticos, o también puedes obtenerlas mediante el método `push()`.

La estructura de la base de datos usada en este proyecto en formato JSON de manera general se puede expresar de la siguiente manera:

```

{
  "Hidrantes": {
    "Código de hidrante": {
      "Valores de alarma": {
        "Clave": "Valor"
      },
      "Valores de presión": {
        "Clave": "Valor"
      }
    }
  }
}

```

En esta estructura podemos apreciar que tenemos un nodo principal llamado "Hidrantes", el cuál tendrá como nodos hijos el nodo "Código de hidrante", donde este código podrá ser de tipo numérico o alfanumérico para que se pueda tener una numeración adecuada de los hidrantes y así saber qué hidrantes se están monitoreando. Luego este nodo tendrá como nodos hijos los "Valores de alarma" y "Valores de presión" que estos a su vez contendrán nodos hijos formados por Claves y Valores, donde la Clave será un código alfanumérico dado por defecto por Firebase a la hora de recibir datos mediante la instrucción push(), como en el caso de este proyecto, por último el Valor será el dato entregado que haya sido medido por el contacto magnético en el caso de los valores de alarma o por el sensor de presión en el caso de los valores de presión, además en el caso de la alarma los valores serán de tipo booleano "True" o "False" y valores numéricos decimales en el caso de la presión de agua.

Esta estructura se puede apreciar con más claridad en la figura 4.5:



Figura 4.5. Estructura de la base de datos

7.2. Protocolo HTTP

La comunicación entre el cliente y servidor mostrada en el diagrama de la figura 5.3 se da mediante el protocolo HTTP, este protocolo de comunicación es el método más utilizado para el intercambio de información en el mundo web (world wide web) y es la manera en la que se transfieren las páginas Web entre servidores y clientes. (Mozilla, 2022)

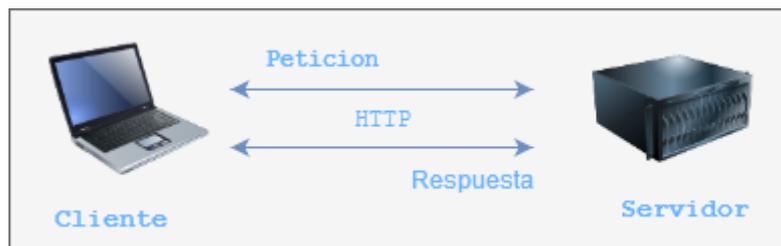


Figura 5.3. Esquema general de una aplicación web

7.3. Capa de presentación

En este proyecto se hizo uso de los lenguajes: HTML, CSS y JavaScript, para la estructura del contenido del sistema de monitoreo, es decir la forma en la cual será presentada la información a los usuarios del sistema. A continuación se describe cada uno de ellos.

7.3.1. HTML5

Es la quinta revisión importante del lenguaje básico de la **WWW** (world wide web) y **HTML** (HyperText Markup Language). Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. (Wide Web Consortium (W3C), n.d.)

En el contexto del sistema de monitoreo de hidrantes, HTML fue utilizado para presentar los datos obtenidos por el prototipo de manera clara y estructurada en una página web. Por ejemplo, se pueden utilizar para visualizar los datos de presión y ubicación de cada hidrante de manera fácilmente comprensible.

7.3.2. CSS

CSS (Cascading Style Sheets) es un lenguaje de diseño usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación o estilo, aunque algunos programadores optan por introducir el estilo en la cabecera de la página. (Mozilla, 2021)

En el contexto del sistema de monitoreo de hidrantes, CSS se utilizó para dar un aspecto atractivo y coherente a la página web que presenta los datos del prototipo. Se pueden utilizar reglas CSS para especificar el tipo de letra y el color del texto para los títulos y párrafos, así como para definir el diseño del fondo y la disposición de los elementos en la página.

7.3.3. JAVASCRIPT - AJAX

JavaScript es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente. (Mozilla, 2022)

AJAX (Asynchronous JavaScript And XML) es una técnica de desarrollo web para crear aplicaciones interactivas, conocidas como RIA (Rich Internet Application). El objetivo de Ajax es mantener una comunicación asíncrona entre la aplicación presentada al cliente y el servidor, esto significa que en cualquier momento puede realizar peticiones al servidor, sin que este se encuentre en sincronía con el cliente (Mozilla, 2022).

JavaScript es el lenguaje en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM) .

La función principal de este lenguaje es presentar una página web dinámica, donde los cambios o modificaciones realizadas se vean reflejadas de inmediato en el navegador como el buscador de la página además de incorporar un mapa para ver las ubicaciones de los hidrantes y sus datos mediante el uso de una librería llamada leaflet.js.

7.3.4. Leaflet

Leaflet es una biblioteca de código abierto de JavaScript que se utiliza para crear mapas interactivos para dispositivos móviles. Con un tamaño de aproximadamente 42 KB de JS, es ligero y fácil de implementar en cualquier proyecto. La librería ha sido diseñada teniendo en cuenta la simplicidad, el rendimiento y la facilidad de uso, lo que significa que es fácil de usar tanto para desarrolladores principiantes como para aquellos con experiencia. Además, Leaflet es compatible con todas las principales plataformas móviles y de escritorio, por lo que puede utilizarse en una amplia variedad de proyectos.

Leaflet también se puede ampliar mediante la utilización de complementos, lo que significa que puede adaptarse a las necesidades específicas de cada proyecto. La API de Leaflet es hermosa y fácil de usar, con una documentación completa que guía al desarrollador a través de cada paso del proceso. Además, el código fuente de Leaflet es legible y simple, lo que lo hace atractivo para aquellos que quieran contribuir al proyecto.

La función principal de Leaflet en el sistema de monitoreo es permitir visualizar un mapa, lo que nos permite ver la ubicación de los hidrantes de manera clara y precisa. Esto puede ser especialmente útil en situaciones de emergencia, cuando es necesario localizar rápidamente los hidrantes más cercanos. En resumen, Leaflet es una herramienta valiosa para cualquier proyecto que requiera la visualización de mapas interactivos en dispositivos móviles o web.

7.4. Desarrollo de la aplicación Web

Una de las herramientas de las cuales se usaron es Visual Studio Code.

7.4.1. Visual Studio Code

El desarrollo de los scripts (HTML, CSS y JavaScript) se realizó con visual studio code, este se muestra en la figura 5.4. Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET). (Microsoft, n.d.).

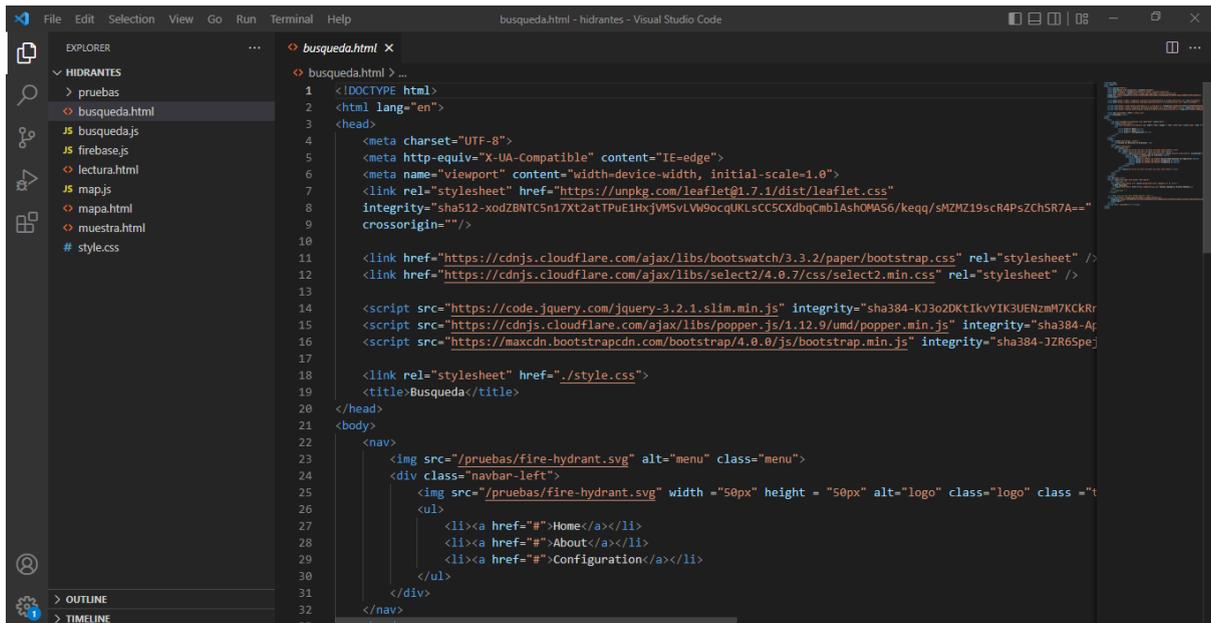


Figura 5.4. Editor de texto Visual Studio Code

7.4.2. Etapa de visualización de datos

La capa de presentación es una de las capas de un sistema de monitoreo, y se encarga de mostrar la información y los resultados obtenidos por el sistema de una manera clara y fácilmente comprensible para el usuario. En este caso, la presentación de la ubicación del hidrante en un mapa y los datos relacionados con él puede ser un ejemplo de cómo la capa de presentación puede ayudar a visualizar y comprender los resultados del sistema de monitoreo.

Es importante tener en cuenta que la capa de presentación trabaja en conjunto con otras capas del sistema, como la capa de aplicación y la capa de datos, para proporcionar una experiencia de usuario completa y coherente. La capa de aplicación se encarga de llevar a cabo tareas específicas, como el procesamiento de datos y la toma de decisiones, mientras que la capa de datos se encarga de almacenar y gestionar los datos del sistema.

A continuación en la figura 5.5 se muestra la capa de presentación de la página web utilizada en el sistema de monitoreo.



Sistema de Monitoreo de Hidrantes

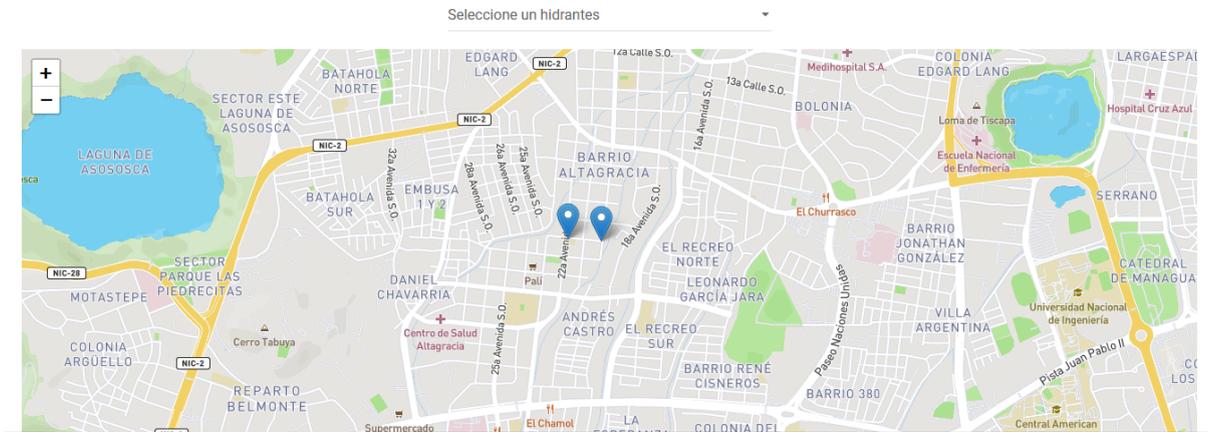


Figura 5.5. Interfaz del sistema de monitoreo de Hidrantes

Además, al hacer clic en un punto del mapa que representa a un hidrante, es posible acceder a más información detallada sobre el mismo (ver figura 5.6), como la alarma que indica si el hidrante tiene una tapa y la presión del hidrante. Esta información puede ser útil para evaluar el estado del hidrante y determinar si es necesario tomar medidas para solucionar cualquier problema o para mantener el hidrante en buenas condiciones. (El código utilizado para la página web se puede ver en Anexos 4).

Además de la información mencionada, también es posible que se muestren otros datos relacionados con el hidrante, como la fecha de instalación, la fecha de última inspección y el historial de mantenimiento. Todos estos datos pueden ser útiles para llevar un seguimiento del estado y el rendimiento del hidrante y tomar decisiones informadas sobre cómo gestionarlo.



Sistema de Monitoreo de Hidrantes

Seleccione un hidrantes



Figura 5.6. Muestra de datos del hidrante en tiempo real

8. Diseño del Prototipo

8.1. Estructura del sistema

8.1.1. Prototipo de hidrante

Para este trabajo se decidió recrear un hidrante y simular su funcionamiento, esto debido a que modificar un hidrante requiere de una aprobación por parte de ENACAL (ver anexos 5), la cuál debe estar debidamente fundamentada y someterse a evaluación, lo que hace esto un procedimiento que requiere bastante tiempo, por lo que en este trabajo se optó por crear un prototipo de un hidrante usando distintos equipos y materiales como una bomba de agua, tubo PVC, reductor de $\frac{1}{2}$ pulgada a $\frac{1}{8}$ de pulgada, pieza T de PVC y llaves de pase de agua.

En este diseño se utilizó una bomba de agua modelo Truper de 1/2 HP (Caballo de fuerza) y otra bomba de agua modelo Truper de 1 HP, ambas conectadas en serie, las cuales simularán el suministro de agua con la presión de agua que poseen los hidrantes, las bombas de agua utilizadas para este trabajo se pueden apreciar en la figura 6.1 y 6.2:



Figura 6.1. Bomba de agua de $\frac{1}{2}$ HP



Figura 6.2. Bomba de agua de 1 HP

Se realizó una conexión en serie entre ambas bombas de agua ya que una sola bomba de esta no suministraba el valor mínimo de presión de agua deseada, por lo que al colocar en serie una segunda bomba de agua esta amplificaba la presión proporcionada por la primera bomba.

El suministro de agua en los hidrantes no siempre es constante, esto debido a que en Managua suelen ocurrir cortes del servicio de agua potable sin previo aviso, entonces esto provoca que algunos hidrantes en un determinado momento no tengan agua disponible, para simular este fenómeno se utilizó una llave o válvula de pase para simular el corte del suministro de agua, esto se puede apreciar en la figura 6.3.



Figura 6.3. Llave de pase de agua

Para que los bomberos puedan sacar agua de un hidrante ellos tienen que abrir con sus equipos una válvula que permite el paso de agua hacia el hidrantes y así luego poder conectar sus mangueras, para simular este mecanismo a nivel de prototipo se utilizó una llave de agua para que simule la salida de agua del hidrante. (Ver figura 6.4).



Figura 6.4. Llave de agua

Finalmente, para diseñar el prototipo del hidrante se utilizaron los materiales anteriormente mencionados siguiendo el siguiente procedimiento:

- 1) Se conectó el suministro de agua domiciliar hacia la entrada de la primera bomba de agua.

- 2) Luego en la salida de la bomba de agua se introdujo un pequeño segmento de tubo PVC por el cuál pasaría el agua saliente de la primera bomba.
- 3) La salida de la primera bomba fue conectada con la entrada de la segunda bomba mediante el tubo PVC.
- 4) A la salida de la segunda bomba se utilizó otro segmento de tubo PVC y una pieza T de PVC para posteriormente colocarle un reductor de $\frac{1}{2}$ a $\frac{1}{8}$ de pulgada en la parte superior y poder introducir el sensor de presión de agua.
- 5) Finalmente se unió el resto del tubo PVC por la salida restante y se conectó la llave de agua por la cuál sale el agua.

El resultado final del prototipo del hidrante se puede apreciar en la figura 6.5 a continuación:



Figura 6.5. Prototipo de hidrante

8.1.2. Medición de presión de agua

Para monitorear la presión del agua se utilizó un sensor de presión de agua, el cual consiste de tres cables, un cable de alimentación positiva (Vdc) (color rojo), un cable de tierra (GND) (color negro) y un cable de datos (color verde) que es el encargado de enviar las lecturas del sensor hacia el Arduino UNO. En el Arduino UNO se utilizó el pin de 5V, el cuál provee una alimentación de 5 voltios al sensor, el pin GND que sirve como tierra y el pin analógico A0, que es el que recibirá los datos a través del cable de datos del sensor. La conexión del sensor se puede apreciar en el esquema de la figura 6.6.

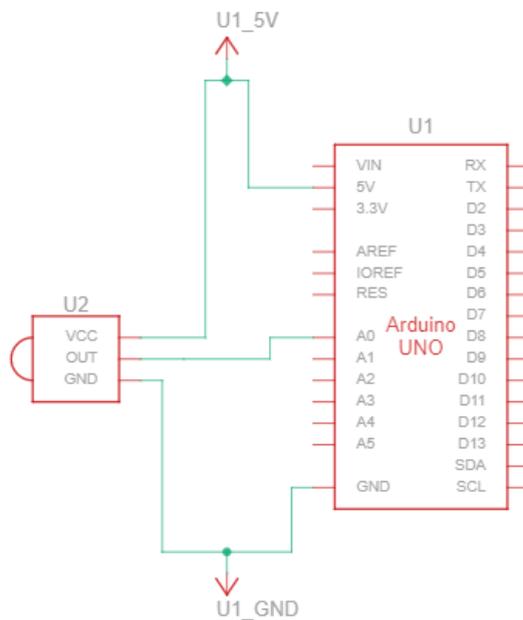


Figura 6.6. Conexión de sensor de presión de agua

Debido a que el sensor de presión de agua es de tipo analógico, el Arduino UNO no puede leer adecuadamente los datos que le suministre el sensor, por lo que es necesario utilizar una ecuación en el código del Arduino para que los datos de la presión de agua se puedan mostrar correctamente.

8.1.3. Detección de daños físicos al hidrante

Para detectar cualquier daño físico o robo de alguna pieza se utilizó un contacto magnético, el cual va conectado en la parte superior del balde, colocado tanto en el extremo superior del balde como en la tapa del mismo, esto para simular cualquier robo o daño físico que se pueda ocasionar a un hidrante.

El contacto magnético consiste de dos piezas, las cuales una de ellas es un imán que se encarga de accionar el contacto magnético según se encuentre cerca o lejos del contacto, luego se tiene el propio contacto magnético, el cuál es el componente principal de este dispositivo, este tiene dos cables que van conectados uno de ellos al pin digital 2 del Arduino UNO, donde enviará una señal binaria de 1 ó 0 y el otro cable va conectado al pin GND del mismo microcontrolador. Esto lo podemos apreciar en la figura 6.7.

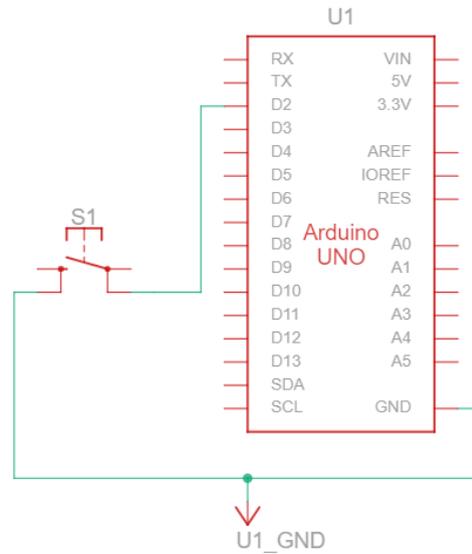


Figura 6.7. Conexión del contacto magnético.

Este contacto magnético está configurado por defecto como Normalmente Cerrado (NC), es decir que siempre que se encuentre cerca del imán este abrirá el circuito interno y emitirá un valor de 0 hacia el Arduino UNO, lo que indica que la alarma estará en estado LOW refiriéndose a que se encuentra desactivada, por otro lado cuando se aleje del imán el contacto cerrará el circuito interno y emitirá un valor de 1 o HIGH, indicando en este caso que la alarma ha sido activada, por lo que existe la posibilidad que el hidrante haya sufrido algún tipo de daño físico.

8.1.4. Comunicación inalámbrica

Para la comunicación del sistema se ocuparon dos módulos NRFL24, de los cuáles uno de ellos se ubicó en la etapa de transmisión del sistema, es decir, conectado al Arduino UNO, la conexión se realizó de acuerdo a la figura 6.8.

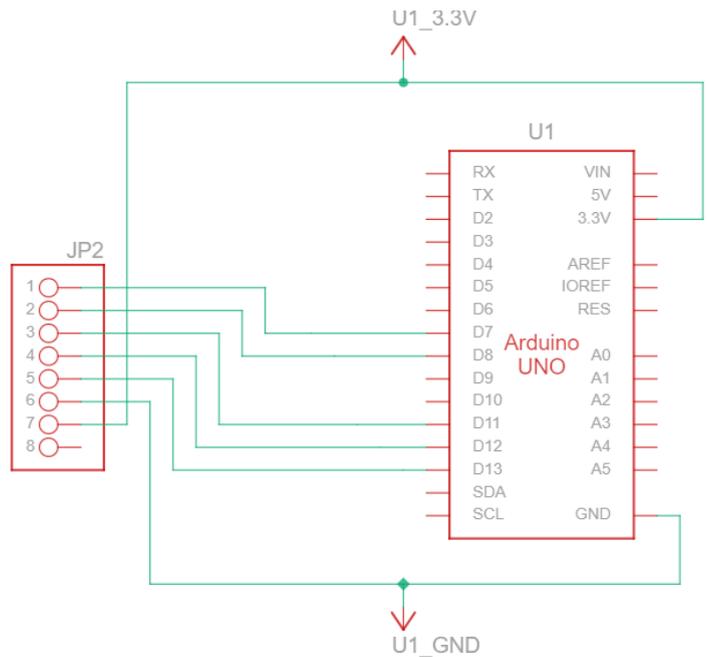


Figura 6.8. Conexión del NRF24 al Arduino UNO

A como se puede apreciar en el esquema, el pin 7 (Vcc) es el pin de alimentación del módulo, el cuál va conectado al pin de voltaje de 3.3V que proporciona el microcontrolador Arduino UNO, el pin 6 es GND, el cuál va conectado al GND del Arduino, luego los pines 1 y 2 (CE y CSN) van conectados a los pines 7 y 8 del Arduino UNO ya que estos se encargaran de configurar el módulo en modo de transmisión. También tenemos los pines MOSI(3), MISO(4) y SCK(5), los cuáles van conectados a los pines 11, 12 y 13 del Arduino UNO respectivamente.

El otro módulo se ubicó en la etapa de recepción del sistema, es decir, conectado al módulo ESP8266, la conexión se puede apreciar en el siguiente diagrama mostrado en la figura 6.9:

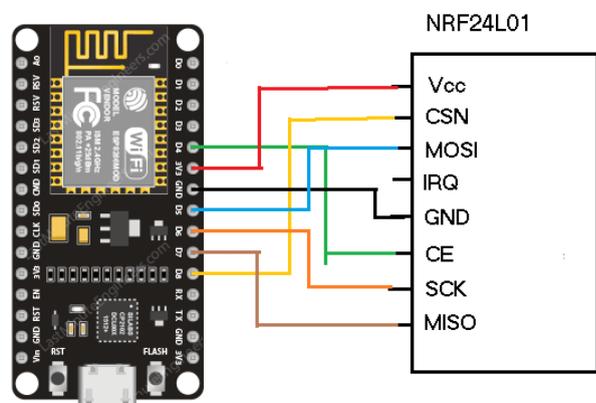


Figura 6.9. Conexión del NRF24 al ESP8266

A como se puede apreciar en la figura 6.9, el pin Vcc del módulo va conectado al pin 3v3 del ESP8266, ya que este es el pin que alimentará al módulo, de manera similar los pines GND se encuentran conectados entre sí en ambos módulos. Luego, los pines CE y CSN van conectados a los pines D4 y D8 respectivamente, ya que estos se encargaran de configurar el módulo en modo de recepción. Por último los pines MOSI, MISO y SCK van conectados a los pines D5, D7 y D6 respectivamente.

8.1.5. Diseño de la interfaz web

La interfaz web diseñada para este proyecto consiste de tres componentes principales, estos son: la base de datos, el servidor web y la página web (ver figura 6.10).

Estos tres componentes se relacionan a como se puede visualizar en el siguiente diagrama de bloques:

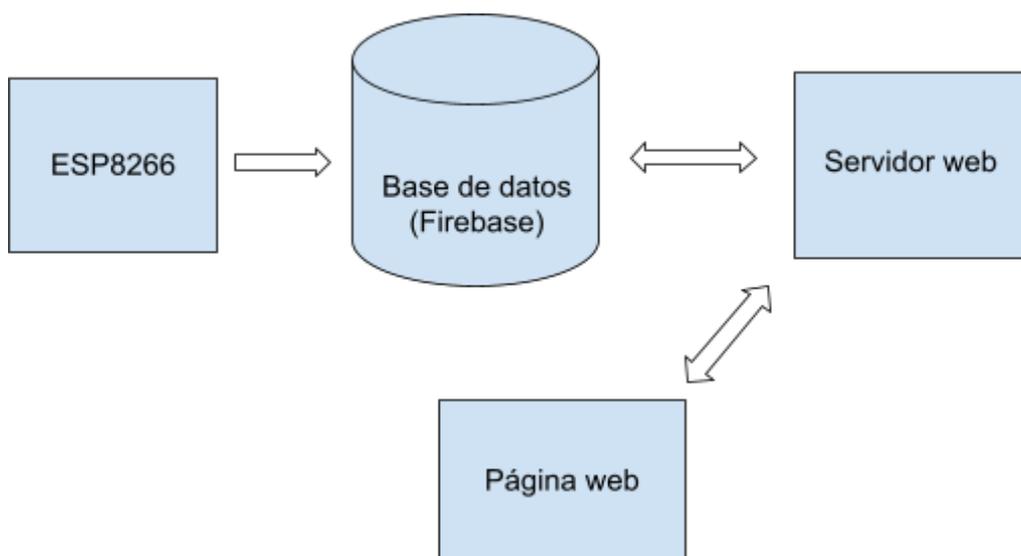


Figura 6.10. Diagrama de bloques de aplicación web

En Firestore se almacenarán todos los datos medidos provenientes de las lecturas de los sensores, a través de la comunicación inalámbrica el Arduino UNO transmitirá los datos medidos hacia el módulo ESP8266 y este último a su vez enviará estos datos hacia Firestore, luego los datos almacenados en Firestore serán procesados por el servidor web y finalmente la página web se encargará de mostrar los datos por medio de un mapa incrustado en esta misma.

8.2. Implementación de la solución

Al finalizar las pruebas del prototipo del sistema de monitoreo se realizó un presupuesto del costo de la implementación del prototipo. En la tabla 4 se muestra el listado de los materiales:

Tabla 4. Presupuesto de la implementación del prototipo de estación de monitoreo.

Presupuesto del diseño de prototipo de estacion de monitoreo de hidrantes			
Componentes para la contruccion del prototipos			
Cantidad	Descripcion	Precio Unitario	Precio total
1	Arduino UNO R3	\$20,00	\$20,00
1	ESP8266	\$10,00	\$10,00
2	Modulo de Transmision y Recepcion NRFL24	\$8,00	\$16,00
1	Transductor de presión de 200 PSI con conector – 1/8 pulgadas	\$30	\$30
1	Cables Kit for Arduino(30 unidades)	\$3,00	\$3,00
1	Contacto magnetico	\$10,00	\$10,00
1	9V 1.5A AC/DC Adaptador de Energia	\$10,00	\$10,00
1	Power Bank	\$20,00	\$20,00
Subtotal		\$111,00	\$119,00
Mano de obra y gasto por servidor			
100	Programacion en Arduino (horas)	\$10,00	\$1.000,00
200	Programacion Web (horas)	\$10,00	\$2.000,00
50	Prueba del circuito en protoboard	\$10,00	\$500,00
12	Pago mensual del dominio y Hosting	\$30,00	\$360,00
Subtotal		\$60,00	\$3.860,00
Total		C\$147.223,00	\$3.979,00

Los precios de los materiales y equipo pueden variar de acuerdo al proveedor por lo que se muestra un aproximado del costo máximo de cada unos de los materiales y la mano de obra para la elaboración del prototipo de monitoreo de hidrantes.

8.2.1. Estructura del Prototipo de Estación de Monitoreo de Hidrantes

La estructura de este prototipo se realizó de acuerdo al siguiente diagrama de bloques mostrado en la figura 6.11:

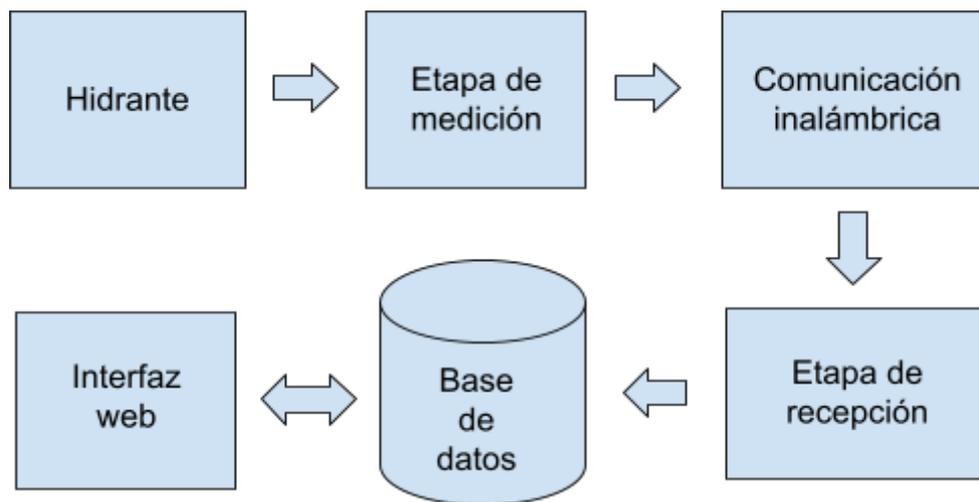


Figura 6.11. Diagrama de estación de monitoreo de hidrantes

8.2.2. Hidrante

El prototipo de hidrante utilizado es el que fue mostrado en la figura 6.5, donde las bombas de agua simularán el suministro de agua de la red de agua potable y el segmento de tubo PVC es donde se estará simulando el funcionamiento del hidrante y donde se estarán realizando las mediciones de la presión del agua.

8.2.3. Etapa de medición

En esta etapa a como se puede apreciar en la figura 6.12 se encuentran el microcontrolador Arduino UNO con todos los sensores a utilizar para realizar las debidas mediciones en el prototipo de hidrante, el sensor de presión de agua se encuentra instalado en medio del tubo PVC, que es por donde circula el agua y el contacto magnético se encuentra ubicado en la tapa del hidrante para simular los daños físicos que se puedan ocasionar al mismo.



Figura 6.12. Conexión de la etapa de medición

Las mediciones realizadas se pueden apreciar en el monitor serie del IDE de Arduino, a continuación en las figuras 6.13 y 6.14 se pueden apreciar algunas mediciones realizadas:

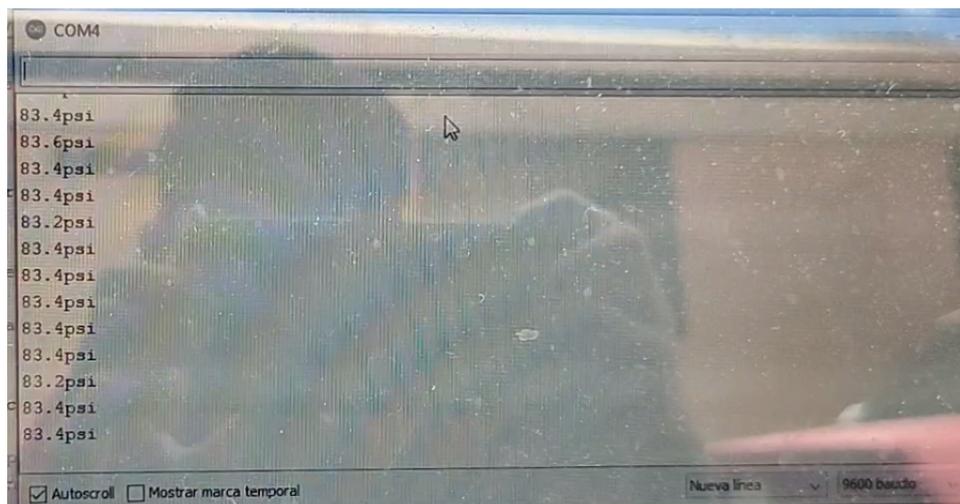


Figura 6.13. Muestra de mediciones realizadas

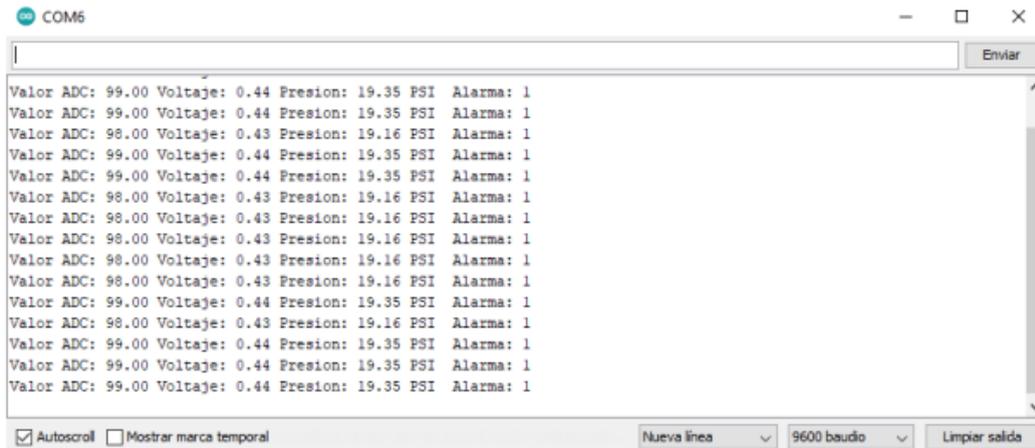


Figura 6.14. Muestra de mediciones realizadas

8.2.4. Comunicación inalámbrica

Para poder transmitir los datos medidos hacia una interfaz web, fue necesario establecer algún tipo de comunicación entre el microcontrolador Arduino UNO y el módulo ESP8266 (ver figura 6.15), en este proyecto se decidió utilizar el módulo NRF24L01, el cuál permite comunicación vía RF(RadioFrecuencia) entre dos dispositivos colocando un módulo de estos en cada uno de los componentes que deseamos comunicar entre sí y configurando uno de ellos en modo transmisión y el otro en modo de recepción.

En el Arduino UNO se configuró este módulo en modo de transmisión, ya que desde aquí es donde provienen los datos medidos en el hidrante, esto se realiza mediante el IDE de Arduino utilizando la sentencia “*objetorf24.openWritingPipe(dirección);*”.

Finalmente en el ESP8266 se configuró este módulo en modo de recepción, ya que éste se encarga de recibir los datos obtenidos en la etapa de medición, para realizar esto se utilizó en el IDE de Arduino la sentencia “*objetorf24.openReadingPipe(dirección);*”.

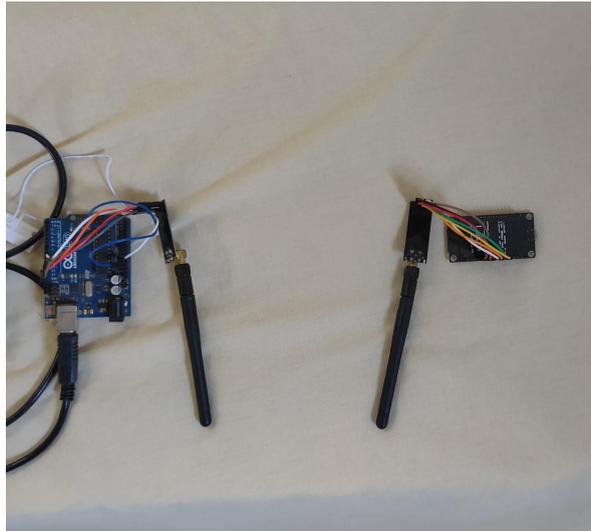


Figura 6.15. Componentes de la comunicación inalámbrica

8.2.5. Etapa de recepción

El componente principal de esta etapa es el módulo ESP8266, debido a que éste es el que se encarga de recibir los datos provenientes de la etapa de recepción y a su vez este módulo por sus características es capaz de comunicarse vía WiFi con distintas aplicaciones web, en este caso se utiliza para establecer comunicación con la base de datos Firebase.

Para establecer la comunicación entre el ESP8266 y Firebase se utilizaron las siguientes líneas de código en el IDE de Arduino:

Primero se debe conectar el módulo a una red WiFi utilizando las siguientes sentencias:

```
const char *ssid = "Nombre_red"; // nombre de la red
const char *password = "Contraseña";
```

Luego se debe acceder a la base de datos de Firebase mediante sus credenciales como lo son el Host y la clave de autenticación, en las siguientes líneas se muestra la sintaxis para realizar esto:

```
const char *FIREBASE_HOST = "https://base-de-datos.firebaseio.com/";

const char *FIREBASE_AUTH = "Clave de base de datos";
```

Por último para enviar los datos hacia la base de datos, se ocupa la sentencia “*Firebase.pushInt(datos);*”.

8.2.6. Base de datos

Para este proyecto se ocupó la base de datos de Firebase, la cuál es una base de datos no relacional. En esta etapa se hará la función de recibir los datos enviados desde el módulo ESP8266 y recopilar todos esos datos y luego hacer uso de estos en la interfaz web.

Firebase muestra los datos en forma de nodos, teniendo como nodos principales el identificador de cada hidrante y dentro de este nodo se encuentran dos nodos hijos, los cuáles representan la presión de agua y la señal de alarma, finalmente dentro de cada uno de estos últimos nodos se almacenarán los valores de las mediciones realizadas.

8.2.7. Interfaz web

Este proyecto cuenta con una interfaz web que hará la función tanto de servidor para procesar los datos recibidos así como un medio para la visualización de todos los datos de los hidrantes (ver figura 6.16).

La página web consiste en tres secciones importantes, el inicio o página de presentación, la sección del sistema de monitoreo y una guía paso a paso en la cuál se explica cómo hacer uso de la página.

En la sección de inicio simplemente se muestra información importante acerca de la página como lo son el propósito de ésta, cómo surgió la idea y la importancia de esta página en cuánto a la implementación de un sistema de monitoreo de hidrantes, además de algunos elementos decorativos como imágenes. En la siguiente figura podemos apreciar la sección de inicio de la página web:



Figura 6.16. Sección de inicio de página web

Luego tenemos la sección del sistema de monitoreo mostrada en la figura 6.17, siendo esta la más importante, ya que aquí es donde podemos encontrar el mapa de la ciudad de Managua donde se encuentran señalados mediante marcadores las ubicaciones específicas de los hidrantes, al hacer clic en un marcador podemos apreciar la información de dicho hidrante.

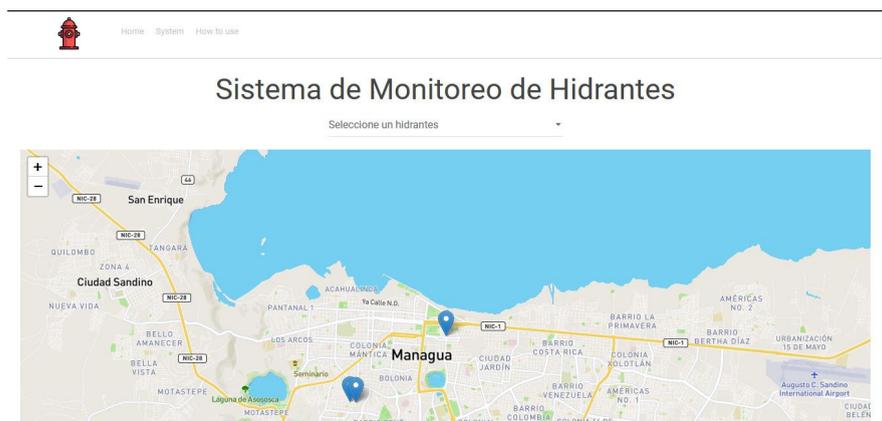


Figura 6.17. Mapa del sistema de monitoreo

En esta sección también podremos observar las distintas alertas que emita la página web en cuanto a la activación de la alarma por medio del contacto magnético de un hidrante, también en caso que la presión de agua sea menor a 80 PSI emitirá una alerta de baja presión de agua, en caso que la presión sea 0 PSI o cercana a cero entonces se emitirá una alerta de falta de abastecimiento de agua del hidrante, esto lo podemos observar en las figuras 6.18, 6.19 y 6.20:

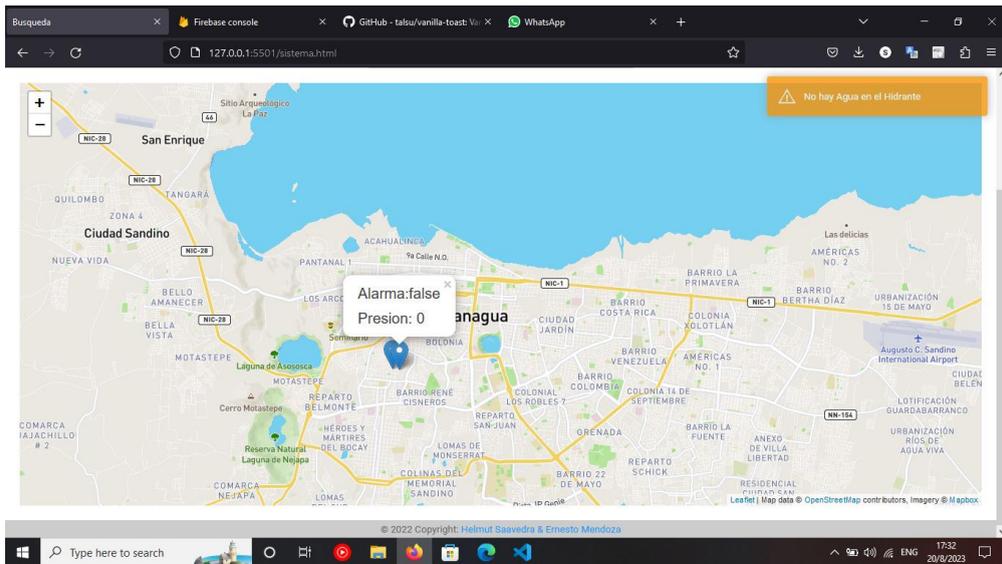


Figura 6.18. Alerta de falta de agua en el hidrante

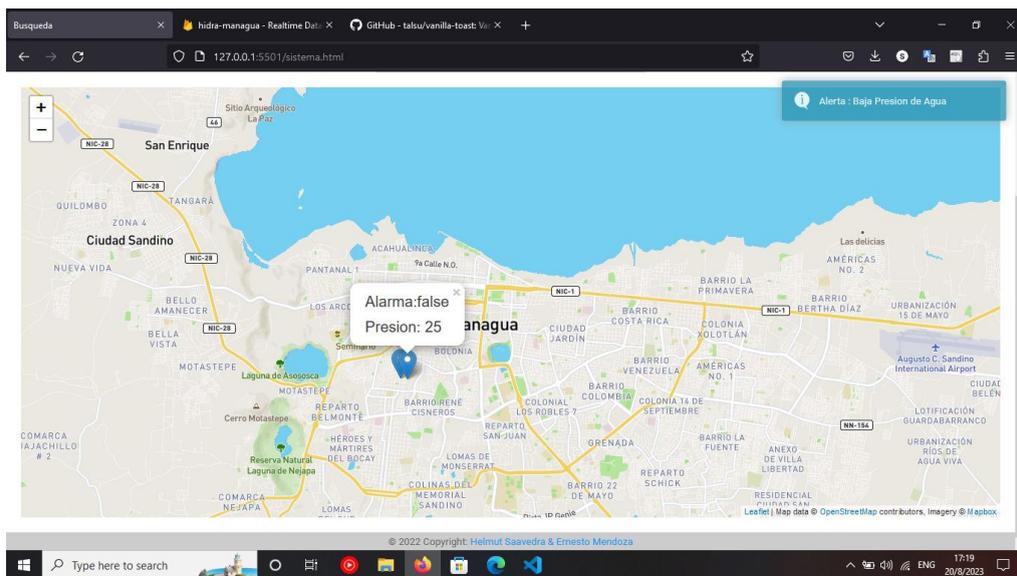


Figura 6.19. Alerta de baja presión de agua

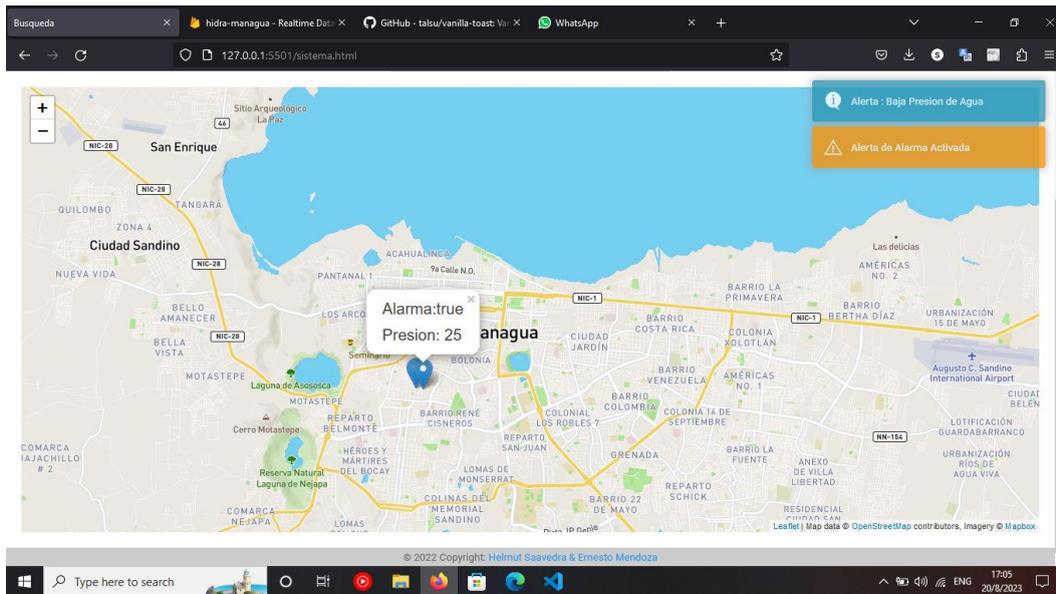


Figura 6.20. Alerta de alarma activada

Por último encontramos la sección de la guía paso a paso (ver figura 6.21), la cuál simplemente es un tutorial sobre cómo usar esta página web, ya sea para visualizar la información de un hidrante, buscar un hidrante en específico, registrar un hidrante y entre otras funciones.

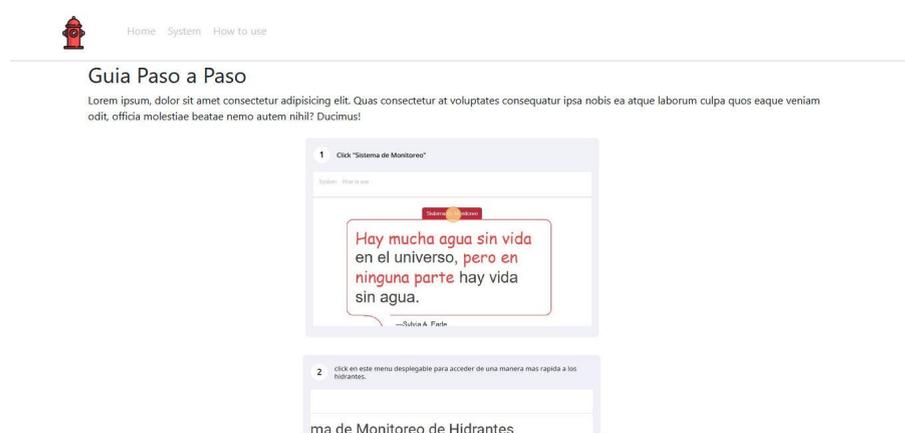


Figura 6.21. Guía paso a paso

9. Conclusiones

- En el presente trabajo monográfico, se desarrolló un Prototipo de Estación de Monitoreo de Hidrantes. Primeramente, se lograron determinar los requerimientos técnicos y funcionales de la estación de monitoreo, mediante entrevistas a personal de la Empresa Nicaragüense de Acueductos y Alcantarillados (ENACAL) y los bomberos, identificándose entre otros requerimientos, la relevancia de la medición de la presión del agua como factor determinante de la funcionabilidad de los hidrantes.
- La estación de monitoreo propuesta, cuenta con dos módulos o partes principales: una de medición y transmisión y la otra que es recepción y transmisión a la internet. El primer módulo consiste en un sistema de medición y adquisición de datos, basado en sensores de presión de agua y contactores, que permiten el registro y monitoreo de la variable presión, así como la activación de una alarma que indica que el funcionamiento del hidrante no es el apropiado. Así mismo, este módulo cuenta con un sistema de comunicación inalámbrico, basado en radiofrecuencia de 2.4 GHz que se comunica con el segundo módulo.
- Este segundo módulo, además de servir como receptor de los datos generados por el módulo de medición y adquisición, cuenta con un módulo WiFi que opera en la banda de los 2.4 GHz, que permite que toda la información sea enviada a una base de datos, encargada de organizar y almacenar todos los datos recibidos y luego estos son entregados a un sistema web, que sirve para la visualización y monitoreo en tiempo real de los datos generados en cada hidrante con el primer módulo.
- Finalmente, la estación fue evaluada mediante pruebas de comunicación, desplegadas en hidrantes que se encuentran en el barrio Altagracia de Managua. Al no poder contar con los permisos para desplegar la estación en los hidrantes, se diseñó un prototipo de hidrante con dos bombas de agua modelo Truper para simular la presión de agua mínima en un hidrante, logrando así, obtener mediciones de presión equivalentes a las que deben mantener los hidrantes. Así mismo, al detener la bomba, se activó la alerta que indica que el hidrante no está funcionando correctamente.

10. Recomendaciones

Además de la información mencionada, también se recomienda que se muestran otros datos relacionados con el hidrante, como la fecha de instalación, la fecha de última inspección y el historial de mantenimiento. Todos estos datos pueden ser útiles para llevar un seguimiento del estado y el rendimiento del hidrante y tomar decisiones informadas sobre cómo gestionarlo. También se pueden tomar en cuenta las siguientes recomendaciones adicionales:

- Establecer protocolos de mantenimiento y monitoreo periódico para garantizar la continuidad y eficiencia del sistema.
- Brindar capacitación al personal encargado del uso y mantenimiento del sistema.
- En caso de monitorear un hidrante real, se deben calibrar las variables de los sensores en función de las mediciones realizadas a un hidrante real, para así tener mayor precisión de los valores obtenidos en un hidrante.
- Si se desea realizar una implementación del sistema de monitoreo a gran escala se recomienda usar un módulo de comunicación 3G para abarcar mayores distancias y asegurar la debida protección de todos los componentes que serán instalados en el hidrante.

11. Bibliografía

Agafonkin, V. (n.d.). *Leaflet*. Leaflet - a JavaScript library for interactive maps.

Retrieved September 15, 2022, from <https://leafletjs.com/>

Hidrantes de incendios. Qué son y qué tipos de hidrantes de incendios hay. (2017).

Grupo Prointex. Retrieved January 4, 2023, from

<https://www.grupopointex.com/hidrantes-de-incendios/>

Sánchez, J. A. (2006). *Instrumentación y control avanzado de procesos.*

[https://books.google.es/books?id=3NkfbokoggcC&pg=PA27&dq=presi%C3%](https://books.google.es/books?id=3NkfbokoggcC&pg=PA27&dq=presi%C3%B3n+relativa&hl=es&sa=X&ved=0ahUKEwja8eGD28vZAhULnRQKHV4DjgQ6AEIJzAA#v=onepage&q=presi%C3%B3n%20relativa&f=false)

[B3n+relativa&hl=es&sa=X&ved=0ahUKEwja8eGD28vZAhULnRQKHV4Djg](https://books.google.es/books?id=3NkfbokoggcC&pg=PA27&dq=presi%C3%B3n+relativa&hl=es&sa=X&ved=0ahUKEwja8eGD28vZAhULnRQKHV4DjgQ6AEIJzAA#v=onepage&q=presi%C3%B3n%20relativa&f=false)

[Q6AEIJzAA#v=onepage&q=presi%C3%B3n%20relativa&f=false](https://books.google.es/books?id=3NkfbokoggcC&pg=PA27&dq=presi%C3%B3n+relativa&hl=es&sa=X&ved=0ahUKEwja8eGD28vZAhULnRQKHV4DjgQ6AEIJzAA#v=onepage&q=presi%C3%B3n%20relativa&f=false)

Microsoft. (n.d.). *Documentation for Visual Studio Code*. Visual Studio Code.

Retrieved September 15, 2022, from <https://code.visualstudio.com/docs>

Mozilla. (2021, July 7). *CSS | MDN*. MDN Web Docs. Retrieved September 15, 2022,

from <https://developer.mozilla.org/es/docs/Web/CSS>

Mozilla. (2022, August 4). *AJAX - Guía de Desarrollo Web | MDN*. MDN Web Docs.

Retrieved September 15, 2022, from

<https://developer.mozilla.org/es/docs/Web/Guide/AJAX>

Mozilla. (2022, August 4). *JavaScript | MDN*. MDN Web Docs. Retrieved September

15, 2022, from <https://developer.mozilla.org/es/docs/Web/JavaScript>

Mozilla. (2022, August 13). *Generalidades del protocolo HTTP - HTTP | MDN*. MDN Web Docs. Retrieved September 15, 2022, from <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

Wide Web Consortium (W3C). (n.d.). *HTML Standard*. HTML Standard. Retrieved September 15, 2022, from <https://html.spec.whatwg.org/>

12. Anexos

12.1. Anexos 1: Código de Arduino UNO

```
/* Debemos incluir la libreria SPI basica y la
libreria RF24 que ya hemos instalado */

#include <SPI.h>

#include <RF24.h>

#include <Wire.h> // incluye libreria para bus I2C

/* Crear un objeto RF24 al que llamaremos "radio".
Los dos argumentos que se utilizan en la funcion
son los numeros de pin que vamos a usar para CS y CE. */
RF24 radio(7, 8); // CE, CSN

const byte identificacion[6] = "00001";

int contacto = 2;

bool alarma;

const int entrada_presion = A0; //Seleccionar el pin analógico
const int presion_cero = 96.00; //Lectura analógica del ADC a 0 psi
const int presion_maxima = 921.6; //Lectura analógica del ADC a 200 psi
const int presion_maxPSI = 200; //Valor de presión máximo del sensor utilizado
const int baudRate = 9600; //Baud Rate del monitor serie
```

```
const int Delay = 2000; //Delay en milisegundos

float valor_presion = 0; //Valor de presión medido por el sensor

void setup() {

    radio.begin();

    /* Mediante la funcion radio.openWritingPipe(),
    establecer la direccion del receptor al que
    enviaremos los datos. En este ejemplo, la cadena
    de 5 caracteres que hemos configurado. */
    radio.openWritingPipe(identificacion);

    /* Configuramos el nivel del amplificador de potencia,
    usando la funcion radio.setPALevel(). Mientras
    estemos a este nivel de prueba, es decir prototipos
    sobre la mesa de trabajo, lo establecemos al minimo,
    ya que los modulos estaran cerca el uno del otro.
    Si se utiliza un nivel mas alto, se recomienda
    utilizar capacitores de filtro entre las entradas
    GND y 3,3 V de los mdulos para que su voltaje se
    mantenga más estable durante el funcionamiento. */
    radio.setPALevel(RF24_PA_MIN);
```

```

pinMode(contacto, INPUT_PULLUP);

/* La función radio.stopListening() cumple la de establecer al módulo como
transmisor. */

radio.stopListening();

Wire.begin(); // inicializa bus I2C

// sensor.begin(); // inicializa sensor con valores por defecto

Serial.begin(9600); // inicializa monitor serie a 9600 bps

}

void loop() {

    valor_presion = analogRead(entrada_presion); //Leer el valor de presión con el pin
analógico

    valor_presion =
((valor_presion-presion_cero)*presion_maxPSI)/(presion_maxima-presion_cero);
//Ecuación para convertir lectura analógica a PSI

    Serial.print(valor_presion, 1); //Imprime el valor de la linea anterior al monitor serie

    Serial.println("PSI");

    //Contacto magnético normalmente cerrado

    /*Cuando el imán está cerca el contacto pasa a

    * estado LOW (circuito abierto)

    * y cuando se aleja el imán pasa a

    * estado HIGH (circuito cerrado)

    */

```

```
if(digitalRead(contacto) == LOW)
{

    alarma = false;

    Serial.print(" Alarma: ");

    Serial.println(alarma);

    radio.write(&alarma, sizeof(alarma));

}

if(digitalRead(contacto) == HIGH)
{

    alarma = true;

    Serial.print(" Alarma: ");

    Serial.println(alarma);

    radio.write(&alarma, sizeof(alarma));

}

radio.write(&valor_presion, sizeof(valor_presion));

delay(Delay);

}
```

12.2. Anexos 2: Código de ESP8266

```
/* Debemos incluir la libreria SPI basica y la
libreria RF24 que ya hemos instalado */

#include <SPI.h>

#include <RF24.h>

#include <Wire.h> // incluye libreria para bus I2C

#include <ESP8266WiFi.h>

#include "FirebaseESP8266.h"

// Credenciales wifi

const char *ssid = "Nombre de red"; // nombre de la red wifi

const char *password = "*****"; // contraseña

const char *FIREBASE_HOST = "URL de la base de datos";

//const char *FIREBASE_AUTH = "*****"; //Clave secreta de la base de datos

/* Crear un objeto RF24 al que llamaremos
"radio". Los dos argumentos que se utilizan
en la funcion son los numeros de pin que
vamos a usar para CS y CE. */

RF24 radio(D4, D8); // CE, CSN
```

```
/* Creamos una matriz de bytes que represente la
direccion con la que se reconoceran entre si los
dos modulos. Podemos cambiar el valor de esta
direccion a cualquier conjunto de 5 letras y/o
numeros. Esta direccion permite elegir con que
receptor vamos a comunicarnos. En nuestro caso,
usaremos la misma direccion tanto en el receptor
como en el transmisor. */
```

```
const byte identificacion[6] = "00001";
```

```
FirebaseData firebaseData;
```

```
bool iterar = true;
```

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    radio.begin();
```

```
/* En el receptor establecemos la misma direccion
que en el transmisor usando la funcion
radio.setReadingPipe() y asi queda habilitada
```

```
la comunicacion entre los dos modulos. */  
radio.openReadingPipe(0, identificacion);
```

```
/* Configuramos el nivel del amplificador de potencia,  
usando la funcion radio.setPALevel(). Mientras  
estemos a este nivel de prueba, es decir prototipos  
sobre la mesa de trabajo, lo establecemos al minimo,  
ya que los modulos estaran cerca el uno del otro.
```

```
Si se utiliza un nivel mas alto, se recomienda  
utilizar capacitores de filtro entre las entradas  
GND y 3,3 V de los mdulos para que su voltaje se  
mantenga mas estable durante el funcionamiento. */
```

```
radio.setPALevel(RF24_PA_MIN);
```

```
/* La funcion radio.startListening() establece al  
modulo como receptor. */
```

```
radio.startListening();
```

```
Wire.begin(); // inicializa bus I2C
```

```
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED)
```

```
{
```

```
    Serial.print(".");
    delay(250);
}

Serial.print("\nConectado al Wi-Fi");
Serial.println();

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

}

void loop() {

    String nodo = "Hidrantes";

    //while (iterar)
    //{
        if (radio.available()) {
            /*char texto[32] = "";
            */
            while (iterar)
            {

                float presion;

                /* Creamos una matriz de caracteres donde recibir el
```

```
mensaje */
```

```
bool alarma;
```

```
radio.read(&presion, sizeof(presion)); // Esperamos que llegue algo por RF desde el módulo
```

```
/* El símbolo "&" delante del nombre de la variable establece un indicador que señala a la variable que contiene los datos que queremos enviar y, con el segundo argumento, establecemos la cantidad de bytes que se van a usar de esa variable. En este caso, la función sizeof() nos devolverá la cantidad total de los bytes de la cadena "texto". */
```

```
Serial.print("Presion: "); // muestra texto
```

```
Serial.print(presion);
```

```
Serial.println(" PSI"); // muestra texto y salto de línea
```

```
delay(1000);
```

```
Firebase.pushInt(firebaseData, nodo + "/9/presion", presion);
```

```
delay(150);
```

```
//Firebase.pushInt(firebaseData, nodo + "/temperatura", 31);
```

```
//delay(150);
```

```
//Firebase.pushInt(firebaseData, nodo + "/temperatura", 30);  
  
//delay(150);  
  
//Firebase.pushInt(firebaseData, nodo + "/Humedad", 67);  
  
radio.read(&alarma, sizeof(alarma));  
  
Firebase.pushInt(firebaseData, nodo + "/9/alarma", alarma);  
delay(30000);  
  
//iterar = false;  
  
//Firebase.end(firebaseData);  
}  
}  
  
//iterar = false;  
  
//Firebase.end(firebaseData);  
  
}
```

12.3. Anexos 3: Entrevista realizada a la Empresa Nicaragüense de Acueductos y Alcantarillados (ENACAL)

Se realizó una entrevista personal al Supervisor General de Operaciones de Enacal, Ing. Moisés Tercero, el día 14 de julio del año 2021 a las 9:00 am, la cual se centró principalmente en tres preguntas claves que son presentadas a continuación junto a sus respectivas respuestas por parte del entrevistado.

- 1) ¿Tiene ENACAL algún sistema de monitoreo digital o de otro tipo que realice el monitoreo de los hidrantes en la ciudad de Managua?

“Actualmente no se dispone en ENACAL de ningún sistema para monitorear hidrantes en Managua”

- 2) ¿Cómo monitorea ENACAL el estado de los hidrantes?

“No se realiza ningún tipo de monitoreo como tal, solo se reciben reportes como alertas ciudadanas o de los bomberos cuando encuentran un hidrante en mal estado o fuera de funcionamiento”

- 3) Al realizar el mantenimiento o reparación de algún hidrante, ¿Se guían por las alertas ciudadanas o hacen una calendarización anual para luego enviar una cuadrilla?

“ENACAL no realiza ningún plan de calendarización anual para efectuar la reparación de hidrantes, solamente las alertas ciudadanas o los reportes de los bomberos son los que permiten a ENACAL saber sobre el estado de los hidrantes, por lo que al final no hay un control total del estado de los hidrantes y sumado a que la información de ubicaciones de hidrantes que se dispone no se encuentra debidamente actualizada”

Una vez finalizada la entrevista se llegó a la conclusión de que hay un problema que amerita solucionarse, por lo que el sistema planteado en este trabajo ofrece una solución adecuada a la problemática presente en ENACAL en cuanto a tener un control efectivo de los hidrantes en la ciudad de Managua.

12.4. Anexos 4: Enlace al repositorio de código de página web

<https://github.com/suricata01/mapaHidrante>

12.5. Anexos 5: Segunda entrevista realizada a la Empresa Nicaragüense de Acueductos y Alcantarillados (ENACAL)

Se realizó una entrevista personal a la subordinada del Supervisor General de Operaciones de ENACAL, Ing. María Jocabed Gutiérrez García, el día 6 de marzo del año 2023 a las 2:00 pm, de la cuál se habló principalmente del tema de los permisos y gestiones necesarias a la hora de hacer modificaciones en un hidrante en la ciudad de Managua. También se brindaron algunos detalles adicionales respecto a las preguntas realizadas en la entrevista anterior. A continuación se muestran las preguntas realizadas junto a sus respectivas respuestas por parte de la entrevistada.

- 1) ¿Qué procedimiento o gestión se debe seguir con ENACAL si deseamos modificar un hidrante, es decir, añadirle algún dispositivo electrónico como un sensor?

“En ese caso se debe enviar una carta de parte de la universidad en la cuál quede descrito de manera bien fundamentada la modificación y el trabajo que se va a realizar y la necesidad de hacerlo, luego ENACAL analizará el caso y dará su debida aprobación o rechazo. Esto también dependerá del planteamiento del tema monográfico, en este caso el tema planteado es un prototipo y no una implementación, por lo que ENACAL no consideraría necesario hacer una modificación a algún hidrante.”

- 2) ¿Tiene ENACAL algún sistema de monitoreo digital o de otro tipo que realice el monitoreo de los hidrantes en la ciudad de Managua?

“En ENACAL no se cuenta con ningún tipo de sistema de monitoreo de hidrantes, esto debido a que implementar algo así en la ciudad de Managua sería muy costoso y demandaría mucho personal con el que no se cuenta actualmente.”

- 3) ¿Cómo monitorea ENACAL el estado de los hidrantes?

“No se realiza ningún tipo de monitoreo, solamente se puede saber del estado de los hidrantes mediante reportes de los ciudadanos o bien de entidades como los bomberos o la Alcaldía de Managua.”

- 4) ¿ENACAL realiza algún plan de mantenimiento preventivo a los hidrantes en la ciudad de Managua?

“No se realiza ningún plan de mantenimiento preventivo, solamente correctivo que es cuando se reporta un hidrante en mal estado y se trata de reparar lo más pronto posible, esto se debe a que ENACAL dedica la mayor parte de su labor a la distribución del agua potable, por lo que los hidrantes quedan en segundo plano respecto a planes de mantenimiento.”

- 5) ¿ENACAL tiene algún registro de los hidrantes en la ciudad de Managua, ya sea un mapa físico o digital?

“Se tiene un catastro en el cuál se encuentran nombrados todos los hidrantes y en el sector en que se ubican, también se tiene un mapa digital pero éste posiblemente no se encuentre actualizado, además es de uso interno de la empresa y solamente se puede mostrar a una tercera parte realizando una solicitud formal.”

Una vez finalizada la entrevista se tomó en cuenta esta información para un mejor planteamiento de la problemática de este trabajo.

12.6. Anexos 6: Entrevista realizada al Cuerpo de Bomberos

Se realizó una entrevista personal al capitán Carlos Artola del Cuerpo de Bomberos del barrio Altagracia el día 12/07/2023 a las 3:00pm, de la cuál se habló acerca de aspectos generales de los hidrantes y cómo es que los bomberos hacen uso de ellos. A continuación se muestran las preguntas realizadas junto a sus respectivas respuestas por parte del entrevistado.

- 1) ¿Cuál es la presión de agua mínima que debe de tener un hidrante para que pueda ser utilizado por los bomberos sin ningún inconveniente?

“La presión mínima que debe tener un hidrante para ser usado debe ser en un rango entre 80 y 100 PSI.”

- 2) ¿De qué manera los bomberos hacen uso de los hidrantes?

“Los hidrantes son utilizados principalmente para el abastecimiento de agua de los camiones cisterna, pero en caso que no tuvieran agua los camiones entonces se conectan directamente las mangueras a los hidrantes para poder actuar rápidamente.”

3) ¿Cuántos hidrantes se encuentran en buen funcionamiento en el barrio Altagracia?

“Aproximadamente el 10% de los hidrantes son los que se pueden utilizar en el barrio Altagracia.”

4) ¿Cuál es el principal problema que influye en el daño físico de los hidrantes?

“Que se roben las tapas de las boquillas donde se conectan las mangueras y estas boquillas queden abolladas, lo que ocasiona que las mangueras no se puedan conectar correctamente y también que roben o dañen la tuerca de operación que permite accionar la válvula de los hidrantes.”

Una vez finalizada la entrevista se tomaron estos datos para sustentar el problema planteado en este trabajo y poder dar una solución más acertada a esta problemática.