Diseño De Una Plataforma IoT Para El Registro Y Control Del Consumo De Agua Potable Suministrado A Través De Tanques De Almacenamiento Comunitarios

Andrea Fabiana Villamizar Ruiz y Anngy Nathalia Gómez Avila

Trabajo de Grado para Optar al Título de Ingeniería de Sistemas

Director
Ing. Jose Geralbert Rubiano
Especialista en Redes de Computadores

Universidad Industrial de Santander
Facultad de Ingenierías Físico-mecánicas.
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga
2021

Dedicado a

En primer lugar, a Dios por otorgarme sabiduría y paciencia en los momentos que más lo requiero, por permitirme culminar una de las metas más importantes en mi vida profesional.

A mis padres y hermanos por ser el pilar fundamental en mi vida.

A mi madre, Rusbi Ruiz por ser la mejor compañía, amiga y voz de aliento, y demostrarme que nunca es tarde para cumplir sueños.

A mi padre, Fabio Villamizar por ser mi ejemplo de superación, por enseñarme la importancia del trabajo en el cumplimiento de cada meta propuesta.

A mi tía Maritza por ser ejemplo en mi vida, por creer en mí y por el apoyo brindado.

A mis abuelitos, con quienes me hubiese encantado compartir este logro presencialmente.

A mi colega, amiga y compañera de proyecto que a pesar de todos los obstáculos que se nos presentaron fue persistente, por todos los momentos vividos incluidas risas, charlas infinitas, peleas, alegrías y tristezas. Es un gusto compartir este gran logro, fruto de un arduo trabajo y dedicación.

A Mafe, Javi, y Andrés, personas que creyeron en mí y que de una u otra forma aportaron para que esto fuera posible, gracias por cada palabra de aliento, por cada ayuda brindada y por cada momento compartido.

Andrea Fabiana Villamizar Ruiz

Dedicado a

A mis padres, Rosalbina y Neftaly por su amor y dedicación, por motivarme a seguir mis sueños apoyándome en los tiempos difíciles y en las decisiones que he tomado, dándome sus consejos y experiencias.

A mi familia que, aunque no entendían mucho cuando les hablaba sobre las situaciones complicadas que afrontaba en mi paso por la universidad siempre estuvieron para mi entendiéndome y brindándome su apoyo.

A mis amigos, los cuales conocí en la universidad, sin ustedes no hubiera sido posible disfrutar de una manera tan divertida la experiencia universitaria, gracias por sus enseñanzas, por tener la paciencia de explicarme cuando no entendía ciertos temas y acompañarme durante este camino espero que nuestra amistad perdure en el tiempo.

A mi papa, gracias por dejarme salir de la comodidad de nuestro hogar y permitir ir a la universidad en otra ciudad, aunque fuera difícil para él, aunque esto sea una dedicatoria un poco triste ya que no estas físicamente junto a mí, espero que donde estés puedas ver que logre nuestro sueño y celebres este logro.

A mi amiga y compañera de proyecto Andrea, son demasiadas las experiencias vividas juntas por las cuales agradezco eternamente, gracias por estar en los momentos difíciles que tuve, por aconsejarme de la mejor manera y estar siempre junto a mí, por la paciencia que me tuvo incluso cuando no estaba de "genio", Gracias por todo especialmente por nuestra amistad.

Anngy Nathalia Gómez Avila

Agradecimientos

A nuestro director de proyecto, el profesor José Rubiano, por confiar en nosotras. Por el tiempo dedicado, las sugerencias y aportes realizados en el desarrollo del proyecto.

A nuestros familiares, por ser el principal apoyo en momentos de debilidad, por los consejos y enseñanzas brindadas, por ser una pieza fundamental para la culminación de esta importante etapa.

A nuestra colega y gran amiga Maria Fernanda Vera por todo el tiempo compartido desde inicios del pregrado, por el apoyo incondicional y por estar dispuesta siempre a ayudar y compartir conocimiento.

A la Universidad Industrial de Santander por abrirnos sus puertas y permitirnos formarnos personal y profesionalmente.

Contenido

Pág.

Lista de figuras

Pág.

Lista de tablas

Pág.

Lista de apéndices

Pág

Resumen

Título: Diseño de una Plataforma IoT para el Registro y Control del Consumo de Agua Potable Suministrado a Través de Tanques de Almacenamiento Comunitarios[*]

Autor: Anngy Nathalia Gómez Avila, Andrea Fabiana Villamizar Ruiz[**]

Palabras Clave: Desarrollo web, Aplicación, IoT, Control.

Descripción:

El agua es un elemento fundamental e indispensable, su consumo es una necesidad básica para todos los seres vivos. La escasez de este líquido trae consigo graves consecuencias como problemas de salud y afectación en actividades económicas como la ganadería, agricultura, entre otros. En algunas zonas geográficas es complejo tener acceso constante al suministro de agua, esto se debe a la ausencia de acueducto, bajos recursos económicos, ausencia de cadena de suministros y temporadas de sequias, al plantear una solución frente a esta situación se encuentran diversos imprevistos entre los cuales resalta el cómo implementar una distribución equitativa del líquido vital.

Esta problemática ha afectado y sigue persistiendo actualmente en las comunidades, por lo tanto surge la iniciativa de este proyecto, el cual consta del diseño de una plataforma IoT conformada por un sistema de sensores tanto de medición como de identificación de acceso, los cuales proyectan en tiempo real el nivel de agua existente del suministro en una plataforma web y controlando el proceso de abastecimiento de agua al usuario teniendo como referencia el valor del mínimo vital diario de agua establecido por familia. Además, suministra informes periódicos en los cuales se registran la actividad que tuvo en ese intervalo de tiempo el tanque de almacenamiento perteneciente a la cadena de suministro del líquido vital.

---

[*] Trabajo de Grado

[**] Facultad de Ingenierías Físico-mecánicas. Escuela de ingeniería de Sistemas e informática. Director: Jose Geralbert Rubiano. Especialista en Redes de Computadoras.

Abstract

Title: Design of an IoT Platform for the Registration and Control of Drinking Water Consumption Supplied Through Community Storage Tanks[*]

Author: Anngy Nathalia Gómez Avila, Andrea Fabiana Villamizar Ruiz [**]

Key Words: Web Development,Application,IoT,Control

Description:

Water is a fundamental and indispensable element; its consumption is a basic need for living beings. The shortage of this liquid brings with it serious consequences such as health problems and effects on economic activities such as livestock, agriculture, and others. In some geographical areas it is complex to have constant access to the water supply, this is due to the absence of an aqueduct or supply chain and droughts, when proposing a solution to this situation there are various unforeseen events, among which highlights how to implement a equitable distribution of the vital fluid.

This problem that affects many people gave the initiative to this project, which consists of the design of an IoT platform made up of a system of sensors for the measurement and identification of access, projecting in real time the level of water existing in the supply in an application and supplying the user with the liquid having as a reference the daily vital minimum value established. In addition, it provides periodic reports in which the activity of the storage tank belonging to the vital liquid supply chain is recorded in that time interval.

[*] Degree Work

[**] Faculty of Physical-Mechanical, School of Systems Engineering and Informatics. Advisor: Jose Geralbeth Rubiano. Computer Network Specialist

Introducción

El desabastecimiento de agua es una problemática que diariamente afecta a comunidades alrededor del mundo, esto ha generado diversas afectaciones en la calidad de vida y salud de las personas. Debido a la falta de estos sistemas de suministros constantes de agua se hace necesario desarrollar una plataforma IoT que permita controlar y monitorear estos sistemas facilitando la llegada de estos suministros a comunidades vulnerables, ya que es una alternativa que cuenta con una mayor portabilidad y economía en comparación a los sistemas de acueductos convencionales. Además, puede ser implementado como una alternativa en situaciones climáticas en que la escasez de agua se hace presente durante largos periodos de tiempo.

Las tecnologías como IoT actualmente brindan múltiples beneficios en la automatización de procesos, gracias a la implementación de sensores y actuadores, servicios en la nube, y conexiones con servidor. Todos estos componentes interactúan y conforman lo que llamamos una plataforma IoT.

Para llevar a cabo el objetivo de este proyecto se desarrolló una plataforma la cual cuenta con una aplicación web, base de datos en la nube, conexión con servidor, sistemas de suministro en tanques inteligente implementando sensores, envió de información en tiempo real y detección de fallas en el sistemas de suministro, además de ser una plataforma sumamente completa, también es una alternativa que cuenta con protocolos de seguridad y protección de datos para los diferentes tipos de usuarios como administradores, subordinados y clientes.

Esta plataforma cuenta con tecnologías como lenguaje de programación tanto en C++ para Arduino como en JavaScript y TypeScript para la implementación de frameworks de frontend y backend como lo son Angular y NodeJs, además de servicios en la nube con Amazon Web

Services, sensores de distancias y flujos de agua. Estos junto a más tecnologías y componentes hacen parte de esta Plataforma IoT la cual como primer prototipo busca crear un punto de partida para que en un futuro esta plataforma sea implementada por grandes compañías y generen una mejora en la sociedad.

En este documento se presenta el soporte metodológico, teórico y técnico del desarrollo de los componentes y servicios que se mencionaron previamente, incluyendo las labores de administración y organización que se desempeñaron durante el desarrollo del proyecto.

1. Planteamiento y Justificación Del Problema

El suministro de agua potable es esencial para la supervivencia de los seres vivos y actividades económicas como la agricultura, la ganadería, la industria, entre otros. La escasez del agua es un problema que se origina por muchas causas entre ellas están los fenómenos naturales, los cuales generan periodos prolongados de sequías minimizando las fuentes hídricas, por otro lado, están los consumos elevados y desperdicios del vital líquido, generado por malas conexiones en las tuberías, llaves abiertas, entre otros.

Organismos como la OMS (Organización Mundial de la Salud) y UNICEF (Fondo de las Naciones Unidas para la Infancia) publican datos donde al menos 4.400 millones de personas en el mundo no cuentan con acceso a suministro de agua potable y 4.400 millones no cuentan con servicio de saneamiento (“1 de cada 3 personas en el mundo no tiene acceso a agua potable”,2019)

La escasez de agua no se presenta únicamente en regiones donde las fuentes hídricas son limitadas, como, por ejemplo, en Latinoamérica a pesar de que tiene a su disposición el 31% de

las fuentes hídricas existentes en el mundo, alrededor de 37 millones de personas no cuentan con acceso al servicio de agua potable. (“América Latina: la región con más agua, la más castigada por la sed”, 2015)

En un estudio del 4019 realizado por el Ministerio de vivienda de Colombia, evidencia que 8 de cada 100 habitantes en el país no tienen acceso a agua potable, generando alteraciones en la salud de las personas y en la sostenibilidad de la industria (Tolima, A. ,2019). Según estadísticas de la ONU (Organización de Naciones de Unidas) al menos 800 mil muertes al año en Colombia son debido a la escasez de este recurso. Además, cerca de 318 municipios en Colombia son susceptibles del desabastecimiento de agua potable. (“800 mil muertes al año por falta de agua potable”, 2019).

En tal sentido, surgen interrogantes ¿Existe actualmente mecanismos alternativos de distribución de agua en poblaciones donde no existen acueductos?, ¿Garantiza esta distribución una equidad para la población afectada? Ante esta cuestión, algunos gobiernos locales implementan un sistema periódico de suministro de agua potable mediante carrotanques, sin embargo, esta estrategia puede no ser cien por ciento efectiva, ya que depende de la disponibilidad de los vehículos e inclusive las vías de acceso. Además, al momento de suministrar el agua no hay garantía de una distribución equitativa del líquido.

Por lo expuesto anteriormente, nace la iniciativa de diseñar una plataforma IoT, para el control del suministro de agua potable a través de tanques de almacenamiento ubicados en zonas donde el acceso es escaso o nulo, este sistema permitirá mediante un dispositivo electrónico identificar al usuario en la plataforma, en la cual se tendrá programado el consumo de agua potable diario establecido (mínimo vital) que este puede tomar del sistema de suministro; de igual manera, la plataforma recibirá señales desde un tanque de almacenamiento el cual indicará al ente

encargado del suministro de agua la cantidad de líquido disponible en tiempo real, permitiéndole optimizar el sistema de suministro, tomando decisiones y planificar las frecuencias de envió de nuevos camiones a llenar el tanque.

2. Objetivos

2.1 Objetivo General

Diseñar una plataforma IoT para el registro y control del consumo de agua potable a través de tanques de almacenamiento comunitarios ubicados en poblaciones sin suministro constante del vital líquido.

2.2 Objetivos Específicos

Definir los requerimientos y variables de diseño para los sistemas que involucra la plataforma IoT.

Seleccionar los componentes de software y hardware pertinentes para el desarrollo de la plataforma IoT.

Diseñar la plataforma IoT de registro y control bajo los parámetros y funcionabilidades definidas en los objetivos anteriores.

Evaluar el desempeño del diseño y cumplimiento de los objetivos mediante la elaboración de un prototipo y maqueta a escala del sistema.

3. Marco Referencial

3.1 Marco Teórico

3.1.1 Tanques de almacenamiento comunitarios

Los tanques de almacenamiento son estructuras que tienen como función almacenar la cantidad suficiente de agua para satisfacer la demanda de una población con el objetivo de brindar un servicio eficiente en un sistema de distribución. El diseño y construcción de ellos varía según las condiciones. (Pérez, L. R. (s. f.)).

3.1.2 Internet

Internet es un sistema de redes interconectadas mediante protocolos los cuales ofrecen gran variedad de servicios y recursos, como por ejemplo el acceso a archivos a través de la web. El acceso a información y conocimiento casi ilimitados es una de las principales posibilidades que nos brinda Internet.

En la actualidad existen diversos tipos de conexiones a internet como los son las conexiones satelitales, fibra óptica, líneas eléctricas y telefónicas. Este sistema de redes ha ido revolucionando el mundo con el paso del tiempo. (N., 2020).

3.1.3 Internet de las cosas (IoT)

El término Internet de las Cosas por lo general hace referencia a una infraestructura de red inteligente en la cual la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos cotidianos que habitualmente no poseen conexión a internet .Permitiendo que estos dispositivos con una mínima intervención humana intercambien y consuman información,

no obstante no existe una definición universal que abarque todas las propiedades y ventajas que se obtiene con la implementación del término IoT.

Para su implementación se deben tener en cuenta Tres factores esenciales, estos son los dispositivos, la red y el sistema de control. los cuales deben estar conectados entres si para el correcto funcionamiento dentro del concepto de Internet de las Cosas. Los dispositivos son aquellos que recolectan y emiten información de su entorno, estos deben estar adaptados correctamente (sensores, chip, entre otros.) para llevar a cabo la comunicación con los demás factores. La red es el medio de transmisión (Wifi, Bluetooth, entre otros) de la información recibida por los dispositivos y procesada por el sistema de control. El procesamiento de los datos capturados por los dispositivos se lleva a cabo en el sistema de control en donde se pueden desplegar estos datos a través de diferentes áreas de aplicación. (Valois, M. A, 2019).

3.1.4 Hosting

El alojamiento de las páginas web garantiza que el usuario pueda tener acceso a este espacio mediante su dominio, El hosting es un servicio en línea que brinda alojamiento a las aplicaciones web en Internet, mediante el alquiler de un espacio en un servidor en el cual se pueden almacenar los archivos e información necesario que garantice el funcionamiento correcto de la página web.

El servicio que prestan el hosting depende de los servidores, los cuales son computadores físicos que funcionan todo el tiempo y certifican que el sitio web este disponibles para el usuario. Los proveedores de hosting son los responsables de mantener el servidor en funcionamiento,

proteger la información y transferir el contenido desde el servidor hasta los navegadores web. (B., G. 2021).

3.1.5 MQTT

MQTT (MQ Telemetry Transport) es un protocolo utilizado para la comunicación entre dispositivos de internet de las cosas (IoT), clasificándose por su función en un protocolo de comunicación M4M (machine-to-machine) de tipo message queue. MQTT fue creado en 1999 por el Dr. Angy Standord-Clark de IBM y Arlen Nipper de Arcom.

MQTT tiene como participantes principales en su comunicación un publicador y un suscriptor conectándose con un broker (servidor), cada mensaje enviado por este protocolo va señalado o marcado con un tópico que es el encargado de hacer llegar el mensaje al suscriptor correspondiente.

El protocolo se caracteriza por su sencillez y ligereza, características que lo hacen indispensable para las aplicaciones IoT ya que normalmente se implementan dispositivos de escasa potencia. MQTT tiene un consumo de energía baja dando capacidad para que el funcionamiento sea continuo, adicionalmente requiere de un ancho de banda mínimo. (L., 2021).

3.1.6 Broker MQTT

El broker es un elemento fundamental a la hora de emplear el protocolo de comunicación MQTT ya que es el encargado de recibir los mensajes que han sido enviados y distribuirlos entre los suscritos a los tópicos establecidos. Existe diversidad de brokers MQTT que se pueden implementar entre los que están emqx, mosquitto, mosca, MQTTnet, entre otras opciones. (L., 2019).

3.1.7 Cloud Computing

"La computación en la nube (cloud computing) es una tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local. En el modelo de nube, no hay necesidad de instalar aplicaciones localmente en computadoras" ("Cloud Computing: ¿Qué es y cuáles sus ventajas?" (s. f.).).

Dentro de las principales características de Cloud Computing están multiplataforma ya que solo con tener conexión a internet se puede acceder a la información, flexibilidad rápida brindando al usuario una experiencia dimensionada en base a la demanda, seguridad ya que el acceso a la información es independiente al dispositivo del cual se desea acceder, entre otros ("Cloud Computing: ¿Qué es y cuáles sus ventajas?" (s. f.).).

3.1.8 Servicio

Normalmente las aplicaciones están compuestas por una cantidad de servicios o componentes los cuales son reutilizables. Un servicio web es un conjunto de métodos y/o procedimientos a los cuales se puede acceder por medio de protocolos web estándar. Los servicios utilizan protocolos de transporte tales como HTTP para poder ser accesible a través de la web.

Los servicios web tienen una arquitectura establecida dentro de las cuales se encuentra la arquitectura funcional que es donde destaca el proveedor del servicio, el cliente del servicio y el registro de servicio. Por otra parte, se encuentra la arquitectura de capas de protocolos, esta arquitectura está compuesta por cada uno de los protocolos que son implementados en los servicios, estos protocolos tienen establecidas unas capas dentro de las que se encuentran el

transporte de servicios, descripción de servicios, localización de servicios y mensajería XML. (“Tema 1: Introducción a los Servicios Web”, (n.d.)).

3.1.9 Protocolos de red

Los protocolos de red son un grupo de reglas que estipulan la comunicación entre dispositivos que están conectados a una red. Estas reglas están compuestas por instrucciones las cuales permiten a los dispositivos ser identificados y poderse conectar entre sí. Además de aplicar reglas de formateo para que los mensajes viajen de manera correcta durante todo el proceso, estas reglas son determinantes para verificar si los dato son recibidos de manera correcta, han sido rechazados o existe algún tipo de problema en el proceso de transferencia de la información. (Fernández, L. 2020).

3.1.10 Servidor

Procesar solicitudes y hacer entrega de datos a las solicitudes realizadas es la principal actividad de un servidor. Un servidor puede llevar a cabo su objetivo por medio de redes locales o a través de internet, su configuración contiene variedad de propiedades dentro de las cuales destaca la memoria, espacio de almacenamiento y procesamiento solicitudes esto conlleva a que gestionen, almacenen, y procesen datos todo el tiempo obligándolos a tener sistema seguro y fiable. Existen diversos tipos de servidores como los servidores web, servidores FTP, servidores proxy o servidores de juegos online (Negocio, I. P. T, 2018).

3.1.11 Web Of Things

Es un protocolo sencillo, escalable y estandarizado. Que tiene consigo una visión extendida del paradigma de IoT, da solución a diversos problemas presentados en el desarrollo de software de dicho paradigma, como son la creación de aplicaciones interactivas que incluyan variedad de dispositivos o sensores heterogéneos y que aun así puedan trabajar a través de diferentes

tecnologías de conexión, Además, de asegurar que los servicios definidos continúen funcionando con los dispositivos y tecnologías futuras (Salgado, D. C, 2013.).

3.1.12 Editor de Código Fuente

Un editor de código fuente es la herramienta fundamental para un programador, gracias a él podemos ver y editar archivos de una aplicación ya sea agregando, eliminando o editando las funcionalidades de dicha aplicación. Dependiendo del lenguaje que se esté implementando los editores de código fuente suministran al programador una serie de ayudas para la edición del código como detección de errores, autocompletado de instrucciones, entre otros. Dentro de los editores de código fuente podemos encontrar Visual Studio Code que es el editor de código de Microsoft, es multiplataforma y de código abierto, Atom que está realizado en JavaScript, creado inicialmente por GitLab y contiene un gran número de extensiones las cuales permiten personalizar su funcionamiento, TextPad, CodeLobster, entre otros. (Editores de código. (n.d.)).

3.1.13 Lenguaje de Programación

Un elemento fundamental para los programadores es el lenguaje de programación, es un lenguaje formal con el que el programador proporcionará una serie de órdenes o instrucciones con el objetivo de darle una guía paso a paso a la máquina de tal forma que pueda cumplir la tarea propuesta. Existe un gran número de lenguajes de programación, cada uno enfocado a una especialidad que lo caracteriza, cuando hablamos de desarrollo web sobresale JavaScript, PHP, Ruby, Java, entre otros. Los lenguajes de programación llevan consigo una serie de características o componentes como son la sintaxis, semántica y pragmática. En un lenguaje de programación los caracteres forman sentencias y estas sentencias en conjunto forman instrucciones para la máquina. (Mendoza, M. L, 2020).

3.1.14 Framework

Los frameworks son esquemas de trabajas utilizados en su mayoría por programadores para el desarrollo de software. La implementación de ellos permite agilizar el proceso de programación ya que simplifica y evita la repetición de código.

Las ventajas del uso de los frameworks en el desarrollo de software son ahorro de tiempo, mayor seguridad y la definición de estándares de programación. (Qué es Framework. (s. f.).).

3.1.15 Controlador De Versiones

Los sistemas de control de versiones son herramientas de software los cuales facilitan y ayudan a los equipos de desarrollo de software a gestionar de una mejor manera los cambios en los códigos de programación a lo largo del tiempo.

Los controladores de versiones realizan un seguimiento de todas las modificaciones realizadas, de esta manera si se comete un error los desarrolladores pueden retroceder y comparar las versiones anteriores del código con esto se ayuda a resolver el error minimizando las interrupciones para todos los miembros del equipos de desarrollo, estas herramientas tienen como ventaja el registro de un historial de cambios a largo plazo de todos los archivos, también ofrecen la creación de ramas y fusiones permitiendo una mejor organización. Actualmente existen varios controladores de versiones como por ejemplo Git el cual es uno de los más usados y recomendados por diversos desarrolladores. (Atlassian. (s. f.).).

3.1.16 Frontend

Dentro del desarrollo web encontramos variedad de términos de los cuales destaca frontend ya que es uno de los principales componentes de esta estructura. Las páginas web se crean a partir de un modelo llamado cliente/servidor, el componente encargado de la interacción con el cliente

es *frontend*, su objetivo principal es el desarrollo de una interfaz gráfica en búsqueda de la satisfacción del cliente ofreciendo una interfaz amigable, llamativa e intuitiva. Para llevar a cabo un buen desarrollo de frontend es necesario realizar investigaciones relacionadas a la interacción con el cliente que lleven a la realización de un boceto o mockup, una vez aprobado se llevará a cabo el código por medio de tecnologías como HTML, CSS, JavaScript, jQuery, Ajax, BootStrap, Angular, entre otras; para posteriormente realizar pruebas y así medir la aceptación por el usuario (TicJob, R., 2019).

3.1.17 Single Page Aplication

Es una aplicación web que se ejecuta en una sola página, esto quiere decir, que se envía al navegador y la pagina no se recarga durante su uso, debido a esto el tiempo de respuesta es mucho más eficaz y fluida que en una aplicación web tradicional.

Dentro de las ventajas de Single Page Application se encuentran reducción de la complejidad al realizar actualización de versión, menor coste de servidor debido a que soporta menos carga al momento de la ejecución y existen multitud de frameworks para su desarrollo con por ejemplo Angular, React, Vue.js, entre otros (Juan D. (n.d.).) .

3.1.18 Backend

En el desarrollo web son varios los componentes que se implementan en el proceso de desarrollo, el Backend hace referencia a la parte lógica e interna del sitio, el cual se encarga de que todos los elementos funcionen de manera correcta. Este componente no se encuentra visible para el usuario y no se compone de ningún elemento gráfico, Además de garantizar funcionalidad también provee la seguridad y optimización de los recursos.

En el desarrollo del Backend se cuentan con variedad de lenguajes, sin embargo, algunos de los más usados son PHP, Python, Node.js y ASP.NET. (N.2020).

3.1.19 Base De Datos

Una base de datos conjunto de información almacenados en un sistema usualmente es controlada mediante un sistema gestor de base de datos (DBMS). Un sistema de base de datos se conoce como el conjunto conformado por el gestor, los datos y las aplicaciones asociadas.

La información almacenada ofrece la posibilidad de ser administrada, modificada, actualizada y eliminada con el objetivo de controlar y organizar de manera eficaz los datos.

Algunas ventajas de implementar bases de datos son mayor independencia de las aplicaciones que la utilizan, mayor disponibilidad facilitando el acceso, mayor seguridad brindando facilidad en la réplica de la información, menor redundancia, entre otros (“Bases de datos”. (s. f.).).

3.1.20 Servicio REST

REST es una interfaz que conecta varios sistemas basados en el protocolo HTTP, su funcionalidad es obtener y generar información y operaciones, retornado esta información en formatos como JSON. Este formato es uno de los más utilizados en la actualidad debido a que es más ligeros y legible que otro formato.

REST se apoya en HTTP, para realizar peticiones GET, POST, PUT y DELETE. Con una menor complejidad y mayor facilidad en su implementación en comparación con otras interfaces como SOAP. (Moncayo, J. M. R.2020).

3.1.21 Web Sockets

Ante la necesidad de una conexión activa bidireccional entre el servidor y el cliente, con el fin de enviar y recibir mensajes simultáneamente se crearon los WebSockets, para que esta conexión se pueda llevar a cabo es necesaria la participación de dos entidades llamadas WebSocket Server que es el backend encargado de ser quien acepta las conexiones, y WebSocket Cliente que es nuestro frontend siendo este el que establece la conexión con el servidor. En el momento en el que se establece la conexión entre estas entidades se puede iniciar el envío y recepción de mensajes. Los WebSockets fueron implementados inicialmente para mejorar en cuanto a rendimiento las aplicaciones, ya que cuantos más usuarios se conectaban a estas, se realizaban más solicitudes para una actualización de información, estas solicitudes en la mayoría de los casos eran innecesarias ya que no había información nueva para suministrar, al implementar WebSockets se encuentra un tipo de "solicitud" abierta la cual recibe datos o respuesta una vez se detecte cambio en la información relacionada con dicha solicitud. (oblancarte.2017).

3.1.22 Microcontroladores

Un microcontrolador es un circuito integrado programable teniendo básicamente los componentes de un computador, es capaz de ejecutar las órdenes que son administradas y guardadas en su memoria. Un microcontrolador incluye en su interior al menos tres principales unidades funcionales de una computadora: unidad central de procesamiento o microprocesador, memoria y periféricos que son las unidades de entrada/salida.

El procesador o microprocesador incluye en sí al menos tres elementos: ALU (Unidad Aritmética y Lógica) que son los circuitos electrónicos digitales como compuertas, multiplicadores, sumadores, etc. Siendo su función principal realizar operaciones ya sean lógicas, aritméticas o miscelánea. La unidad de control que son quienes permiten la distribución de la lógica

de las señales. Los registros, que son las memorias principales de los procesadores. Por otra parte, Los periféricos son quienes permiten la interacción con los sensores o elementos del mundo exterior, entre los objetivos principales de los periféricos se encuentran habilitar o deshabilitar las salidas digitales y leer sensores analógicos.

Finalmente, la memoria es la encargada del almacenamiento de información o instrucciones suministradas al microcontrolador, la memoria está dividida en tres, La memoria FLASH que almacena el programa, la memoria RAM que se encarga de los datos o variables del programa y la memoria EEPROM la cual está creada para almacenar las configuraciones. (A.2021).

3.1.23 Sensores

Los sensores son dispositivos electrónicos capacitados para detectar acciones o estímulos externo y transformar en señales eléctricas ya sean analógicas o digitales, estas señales son detectadas y procesadas por un microcontrolador. Existe diferentes tipos de sensores según la magnitud física que se desee detectar, como por ejemplo sensores de temperatura, distancia, control de acceso, entre otros. (Medidores, A. P. D. (s. f.)).

3.1.24 Actuadores

Los actuadores son los dispositivos encargados de realizar acciones cumpliendo órdenes suministradas por el sistema de control al cual están conectados, dentro de los actuadores podemos encontrar válvulas, motores, relés, etc. El microcontrolador o controlador envía instrucciones a los actuadores de las acciones que deben realizar para cubrir una necesidad establecida, algunos actuadores tienen únicamente dos estados como abrir/cerrar, encendido/apagado, es decir, 0/1 como son barreras de acceso, motor eléctrico, iluminaciones; por otra parte, algunos actuadores

requieren de valores analógicos, es decir, señales de mando que varían entre 0-5V, 0-10V, 0-40mA y/o 4-40mA.(De control, U. I., Inmótica, S. D. o., & de control y actuadores., S. (n.d.).).

3.1.25 Controladores

Los controladores son conexiones eléctricas y electrónicas las cuales son fabricadas con el objetivo de controlar y procesar la información proveniente de los sensores, controlando el flujo de corriente eléctrica en dispositivos de uso doméstico o industrial.

Los controles eléctricos se clasifican en controles de encendido y apagado, controles de proporción de tiempo y controles de proporción de posición. (Sistemas de Control. (s. f.).).

3.2 Estado del arte

3.2.1 iWesla

El proyecto iWesla realizado en 4016, es una propuesta de emprendimiento que ganó su lugar en Madrid, España; tras su implementación logró reducir el consumo de agua hasta un 50% mejorando la eficiencia y seguridad de los suministros de agua en superficies habitables, teniendo como objetivo brindar modularidad y escalabilidad a sus clientes. Esta iniciativa para la optimización del consumo de agua y la seguridad del sistema incluye sensores que miden el flujo del agua y controlan su paso por la cadena de suministro, plataformas de recolección de información, medios de transmisión, plataformas de recepción y procesamiento, y una aplicación web de visualización de los datos procesados, la cual informa el consumo habitual a sus usuarios (Libelium. 2018).

iWesla resalta la importancia de controlar el consumo de agua implementando un sistema de sensores, debido a esto su funcionamiento y desarrollo es una guía para la realización del prototipo

3.2.2. Medición en Tiempo Real de Caudal y de Nivel en el Acueducto Municipal de Anorí

Este proyecto surge tras la necesidad de conocer en tiempo real la medición del nivel de agua presente en un tanque de almacenamiento ubicado a las afueras de Anorí, un municipio colombiano. Con el fin de evitar el desplazamiento innecesario hacia el caudal para la recolección de datos. Gracias al proyecto implementado se logró obtener la medición del tanque en tiempo real, permitiendo el control interno y proporcionando estadísticas de las diversas variables presentes en el caudal. En el montaje del proyecto se implementaron sensores de nivel de agua, entre otros, y sus respectivos gabinetes con equipos para obtener la señal del sensor; la información suministrada se almacenó en servidores virtualizados y de esta manera se logran observar los valores de las variables en tiempo real desde cualquier dispositivo con acceso a internet (Telemetrik. (s. f.).).

Uno de los objetivos del proyecto a realizar es el monitoreo del nivel del agua mediante la toma de datos en tiempo real, lo cual es un objetivo compartido con el proyecto del acueducto municipal de Anorí. La implementación de sensores es un aspecto que se asemeja a las bases tecnológicas que se quieren aplicar en el prototipo de tanques de almacenamiento comunitarios.

3.2.3. Un Cajero Automático para Sacar Agua

Este proyecto se llevó a cabo en el año 4016 en Nairobi, Kenia. Consiste en un suministro prepagado de agua potable a poblaciones que no tienen acceso a este líquido para su consumo e higiene; se implementaron principalmente sensores para el control de acceso, de esta manera las

personas tras introducir su tarjeta previamente cargada podían obtener agua potable haciendo más sencilla la obtención este servicio (African Slum Journal, 2016).

La implementación de sensores de control de acceso para la identificación del usuario es un sistema de validación que funciona como método de seguridad, evitando suplantaciones en la cadena de suministro del líquido. Este método implementado en el proyecto de cajero automático es relevante y necesario para llevar a cabo el control de acceso al suministro.

4. Desarrollo Del Proyecto

4.1 Diseño de Requerimientos

El diseño de requerimientos es uno de los pasos más importantes que se deben realizar al momento de planear un proyecto. Debido a que son el punto de partida para tener un desarrollo satisfactorio, estos requerimientos son diseñados de acuerdo con los alcances y la problemática que originó la necesidad de crear y llevar a cabo cierta propuesta, moldeando y recopilando diferentes puntos de vista.

Se clasifican en funcionales, describiendo el comportamiento del sistema de acuerdo con la funcionalidad establecida, por otro lado, tenemos los no funcionales los cuales consisten en esas restricciones y funciones que tendrán los servicios que proporcione el sistema.

Para el diseño de los requerimientos de este proyecto nos enfocaremos en la principal causa que motivo el desarrollo de esta idea, la cual consiste en el poco o nulo acceso que tienen ciertas familias al suministro de agua constante. Debido a esto, se propone un sistema que facilite el

control y monitoreo de esta cadena de suministro mediante sensores y dispositivos conectado y ubicados en los tanques comunitarios de almacenamiento.

Este sistema se implementará utilizando placas Arduino, bombas y electroválvulas para la entrada y salida de agua, sensores que midan la cantidad de agua que se le está suministrando al beneficiario. También el sistema de llenado del tanque se controlará mediante un relé el cual modificara el estado de la bombas y electroválvulas a estados como activos e inactivos dependiendo del nivel de agua que tenga el tanque, sin embargo, debemos tener en cuenta que es primordial tener un sistema de medición el cual nos comunique el nivel del tanque. En este momento, es que entra en juego ciertos sensores los cuales informarán en tipo real la medida obtenida.

Por otro lado, necesitamos validar a los beneficiarios mediante un sistema de autenticación el cual nos otorgue acceso si el identificador del usuario (tarjeta) se encuentra registrado. De esta manera, se procederá a notificarle al usuario mediante una LCD la información de su familia y la cantidad de agua que tiene disponible para retirar, procediendo con la selección de la cantidad por parte del beneficiario y la verificación de la transacción estipulando si es exitosa, o por lo contrario, es una transacción fallida ya sea por factores como lo son el nivel de agua del tanque, su estado actual o la errónea cantidad de agua seleccionada.

Aunque lo anteriormente explicado sea de gran importancia, también debemos diseñar los requerimientos para el desarrollo de un aplicativo web en el cual se pueda visualizar la información y tráfico que se presenta en el sistema de suministro, incluyendo también el acceso a usuarios y funcionalidades tales como agregar, editar, eliminar o desactivar ciertos componentes como lo son los tanques, las familias, tarjetas, entre otros. Todo esto con el objetivo de planear y desarrollar un

proyecto que otorgue una mejora mediante la automatización de los procesos, generando mayor efectividad, control y monitoreo en todo el sistema de suministro.

4.1.1 Levantamiento de Requerimientos

Los requerimientos funcionales fueron seleccionados en base al propósito del proyecto y sus objetivos establecidos como se expuso anteriormente. Estos requerimientos fueron divididos en software y hardware.

- Software: en los requerimientos de software se tuvieron en cuenta tanto el backend como el frontend del proyecto y los distintos usuarios que tendrán interacción en la aplicación web.

Tabla 1

Requerimiento funcional de autenticación y registro de usuarios

Prioridad	Alta
Identificación del Requerimiento	RF01
Nombre	Autenticación y registro de Usuarios
Descripción	La aplicación permitirá a los usuarios iniciar sesión, registrarse y modificar su información según las restricciones establecidas

Tabla 2

Requerimiento funcional usuario consulta y actualiza perfil

Prioridad	Alta
Identificación del Requerimiento	RF02
Nombre	Usuario podrá consultar y actualizar perfil.

Descripción	La aplicación permitirá al usuario consultar y actualizar la información permitida de su perfil según su rol

Tabla 3

Requerimiento funcional administrador controla componentes

Prioridad	Alta
Identificación del Requerimiento	RF03
Nombre	Administrador controlará los subordinados, clientes, tanques, familias y tarjetas.
Descripción	La aplicación permitirá al administrador registrar, consultar, activar, desactivar y eliminar familias, tarjetas y usuarios. Además, los tanques deben registrarse, consultarse, desactivarse y eliminarse por el administrador.

Tabla 4

Requerimiento funcional administrador consulta listas de información

Prioridad	Alta
Identificación del Requerimiento	RF04
Nombre	Administrador consultará las listas de familias, usuarios, tarjetas y tanques que se encuentren activos o inactivos.
Descripción	La aplicación permitirá al administrador consultar y ver las listas de las familias, tarjetas, tanques y usuarios que se encuentren registrados en el sistema. Además, de la posibilidad de filtrar estas listas de acuerdo con ciertos parámetros establecidos.

Tabla 5

Requerimiento funcional administrador visualiza dashboard principal

Prioridad	Alta
Identificación del Requerimiento	RF05
Nombre	Administrador visualizará dashboard principal
Descripción	La aplicación permitirá al administrador visualizar el dashboard principal en el cual se registra información de interés sobre los tanques, tráfico del sistema y graficas estadísticas.

Tabla 6

Requerimiento funcional administrador visualiza dashboard personalizado

Prioridad	Alta
Identificación del Requerimiento	RF06
Nombre	Administrador visualizará dashboard personalizado
Descripción	La aplicación permitirá al administrador visualizar el dashboard personalizado del tanque seleccionado en el cual se registra su información de interés, tráfico y graficas estadísticas.

Tabla 7

Requerimiento funcional administrador visualiza fallas en los tanques.

Prioridad	Alta
Identificación del Requerimiento	RF07
Nombre	Administrador visualizará las fallas de los tanques en el dashboard personalizado

Descripción	La aplicación permitirá al administrador visualizar en el dashboard personalizado del tanque la última conexión al bróker y si hubo falla en la conexión durante el día.

Tabla 8

Requerimiento funcional subordinado administra componentes

Prioridad	Alta
Identificación del Requerimiento	RF08
Nombre	Subordinado podrá registrar, consultar, desactivar y actualizar familias, tarjetas y usuarios clientes
Descripción	La aplicación permitirá al subordinado registrar, consultar, y actualizar los perfiles e información de los usuarios clientes, familias y las tarjetas.

Tabla 9

Requerimiento funcional subordinada lista información de componentes

Prioridad	Alta
Identificación del Requerimiento	RF09
Nombre	Subordinado consultará las listas de familias, usuarios, tarjetas y tanques que se encuentren activos

Descripción	La aplicación permitirá al subordinado consultar y ver las listas de las familias, tarjetas, tanques y usuarios que se encuentren en estado activo. Además, de la posibilidad de filtrar estas listas de acuerdo con ciertos parámetros establecidos.

Tabla 10

Requerimiento funcional cliente visualiza dashboard

Prioridad	Alta
Identificación del Requerimiento	RF10
Nombre	Cliente visualizará su dashboard
Descripción	La aplicación permitirá al cliente visualizar su dashboard en el cual se le presenta como la familia a la cual perteneció, mínimo vital disponible, tanque asignado y nivel de dicho tanque

- Hardware: en el levantamiento de requerimientos de este componente se analizó las distintas interacciones que podría realizar el cliente beneficiario de la cadena de suministro de agua y la información que debería suministrársele en el hardware.

Tabla 11

Requerimiento funcional cliente autentica su identidad en el sistema de suministro

Prioridad	Alta
Identificación del Requerimiento	RF11
Nombre	Cliente autenticará su identidad en el sistema de suministro
Descripción	El cliente autenticara su identidad presentando su tarjeta al lector RFID en el sistema de suministro

Tabla 12

Requerimiento funcional cliente visualiza su mínimo vital disponible

Prioridad	Alta
Identificación del Requerimiento	RF12
Nombre	Cliente visualizará su mínimo vital disponible
Descripción	El mínimo vital del cliente podrá ser visualizado en una LCD por el cliente después de su ingreso aprobado al sistema de suministro.

Tabla 13

Requerimiento funcional cliente selecciona la cantidad de agua que desea retirar

Prioridad	Alta
Identificación del Requerimiento	RF13
Nombre	Cliente seleccionará la cantidad de agua que desea retirar
Descripción	El cliente seleccionara la cantidad de agua que desea retirar de acuerdo con su mínimo vital disponible y el estado del tanque

Tabla 14

Requerimiento funcional cliente visualizar el estado del tanque

Prioridad	Alta
Identificación del Requerimiento	RF14
Nombre	Cliente visualizará si el tanque se encuentra inhabilitado al momento de realizar su retiro de agua
Descripción	El cliente visualizara en la LCD el estado en el que se encuentra el tanque y si este se encuentra fuera de servicio.

Los requerimientos no funcionales se seleccionaron basados en las necesidades de los usuarios respecto al manejo y usabilidad de la plataforma de suministro de agua y sus componentes.

Tabla 15

Requerimiento no funcional fiabilidad

Prioridad	Alta
Identificación del Requerimiento	RNF01
Nombre	Fiabilidad
Descripción	Se debe garantizar que el acceso a los servicios de visualización de la información del sistema de suministro tenga una alta disponibilidad.

Tabla 16

Requerimiento no funcional autenticación

Prioridad	Alta
Identificación del Requerimiento	RNF02
Nombre	Autenticación
Descripción	Se debe garantizar la protección sobre el acceso a los servicios y/o recursos del sistema según el rol del usuario.

Tabla 17

Requerimiento no funcional capacidad de uso

Prioridad	Alta
Identificación del Requerimiento	RNF03
Nombre	Capacidad de uso
Descripción	El sistema debe garantizar la usabilidad para todos los usuarios por lo tanto debe contar con una interfaz gráfica amigable y fácil uso

Tabla 18

Requerimiento no funcional tolerancia a fallos

Prioridad	Alta
Identificación del Requerimiento	RNF04
Nombre	Tolerancia a fallos
Descripción	El sistema debe tener una buena gestión de la tolerancia a fallos, es decir, favorecer resiliencia de la plataforma

Tabla 19

Requerimiento no funcional modularidad y reusabilidad

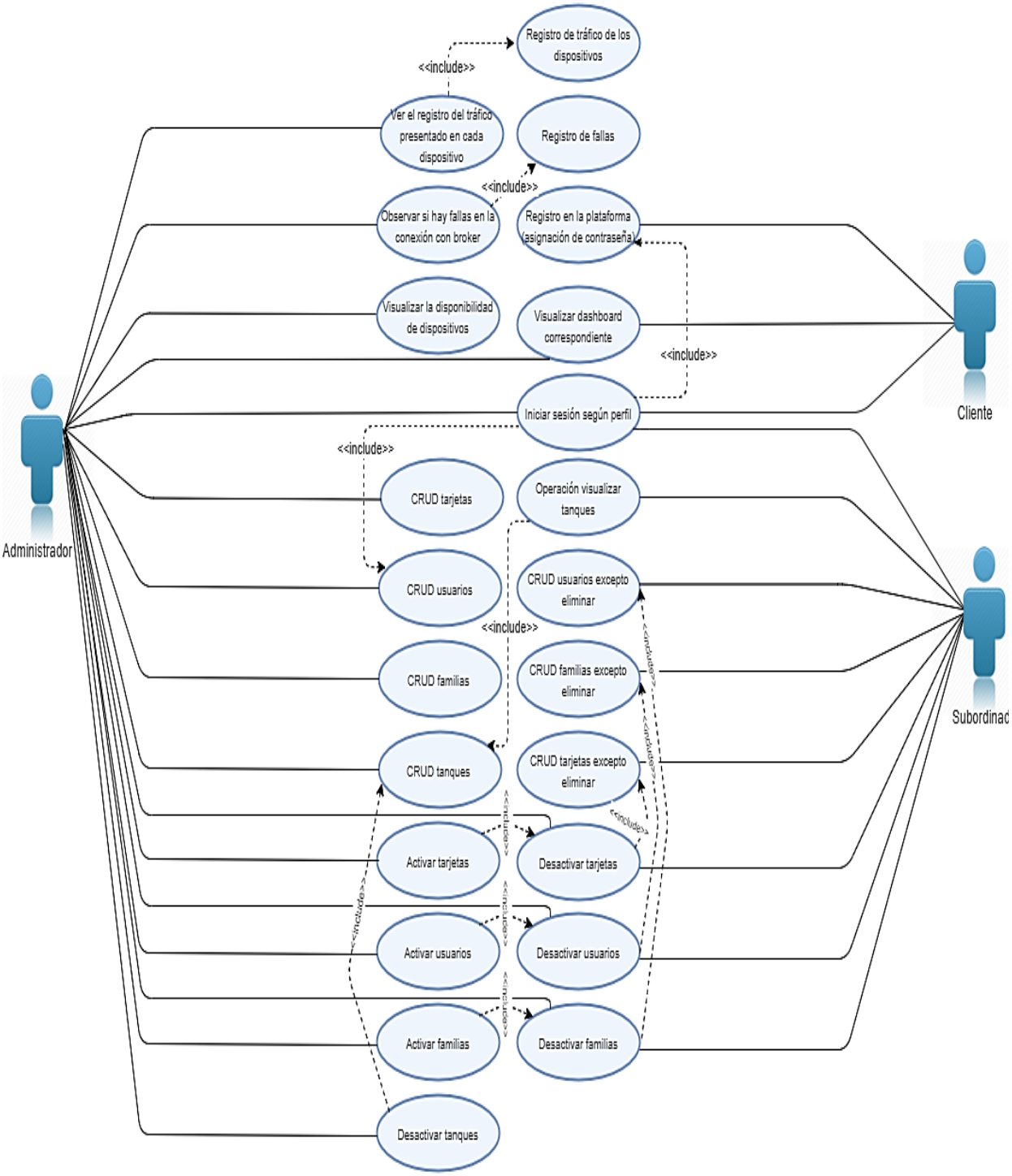
Prioridad	Alta
Identificación del Requerimiento	RNF05
Nombre	Modularidad y reusabilidad
Descripción	El sistema debe priorizar el modularidad y favorecer el reúso de componentes

4.1.2 Casos de Uso

El diagrama de casos de uso ofrece una vista general de todas las acciones que efectúa cada usuario participante del sistema. Los casos de uso que no tiene relación con ninguno de ellos, como son *"Registro de tráfico de los dispositivos"* y *"registro de fallas",* son definidos en el diagrama a causa de su importancia en este proyecto, pero cabe aclarar que estas acciones son llevadas a cabo directamente desde la comunicación constante entre bróker, dispositivos y backend (proceso que se explicará más adelante).

Figura 1.

Diagrama de casos de uso

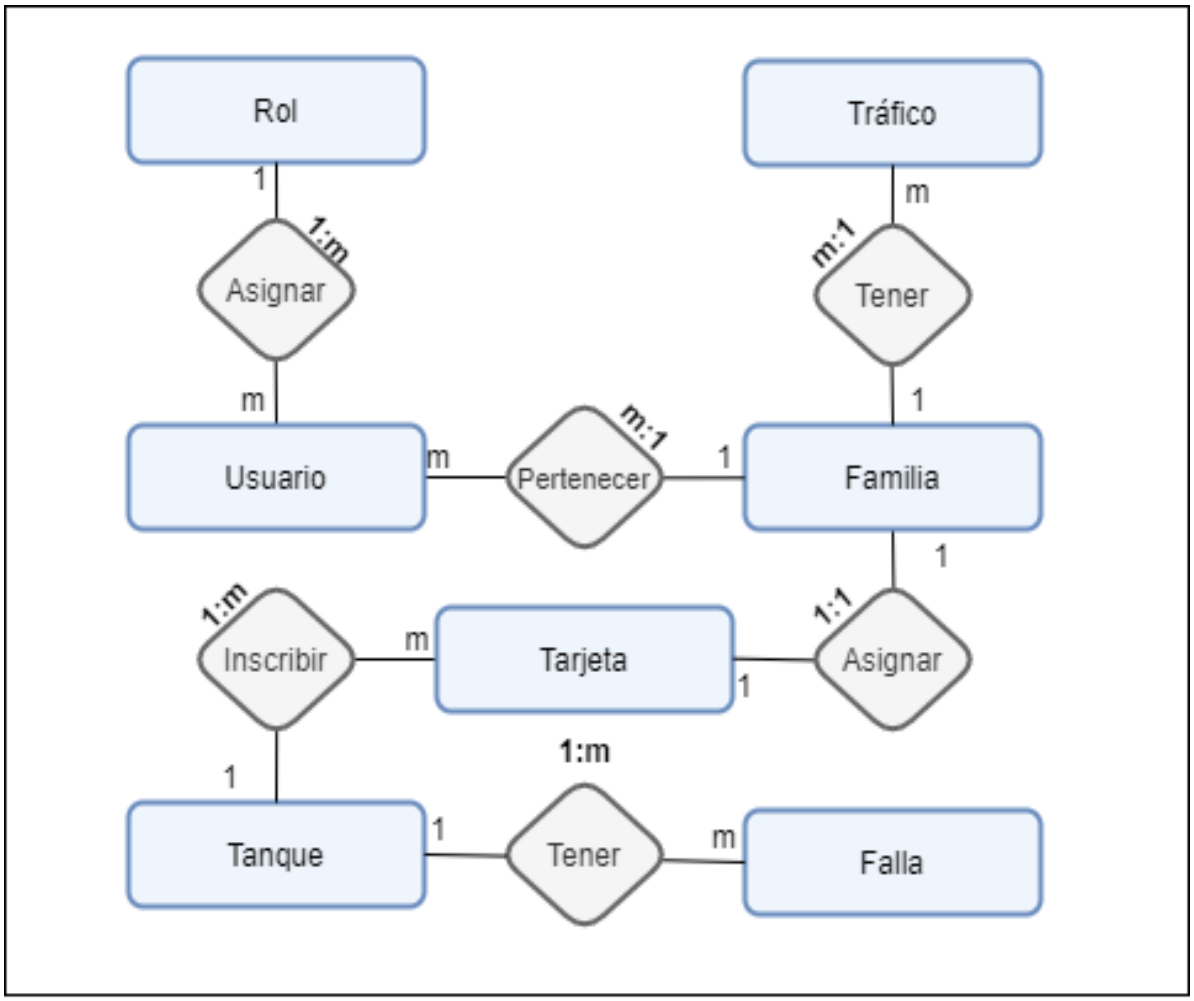

Registro de tráfico de los dispositivos
<<include>>
Ver el registro del tráfico presentado en cada dispositivo
Registro de fallas
<<include>>
Observar si hay fallas en la conexión con broker
Registro en la plataforma (asignación de contraseña)
Visualizar la disponibilidad de dispositivos
Visualizar dashboard correspondiente
<<include>>
Iniciar sesión según perfil
<<include>>
CRUD tarjetas
Operación visualizar tanques
CRUD usuarios
CRUD usuarios excepto eliminar
<<include>>
CRUD familias
CRUD familias excepto eliminar
CRUD tanques
CRUD tarjetas excepto eliminar
<<include>>
Activar tarjetas
Desactivar tarjetas
Activar usuarios
Desactivar usuarios
Activar familias
Desactivar familias
Desactivar tanques
Administrador
Cliente
Subordinad

4.1.3 Diseño de Diagrama Entidad Relación

Con el fin de entender el funcionamiento del proyecto, se utiliza diversidad de herramientas capaces de proporcionar esta facilidad, entre ellas se encuentra los diagramas de entidad-relación

Figura 2.

Diagrama entidad-relación



Como se puede observar en el esquema, el proyecto cuenta principalmente con 7 entidades que definen el funcionamiento de este. La tabla a continuación proporciona una breve explicación de la funcionalidad de cada entidad presente en el diagrama.

Tabla 20

Descripción de entidades del proyecto

ENTIDAD	DESCRIPCIÓN
USUARIO	Es la entidad que contiene toda la información de cada uno de los registrados en las plataformas sin importar el rol asignado.
ROL	Contiene el registro de los roles establecidos, con el fin de tener escalabilidad en el sistema en caso de contemplar la participación de roles adicionales.
FAMILIA	Esta entidad tiene la información necesaria de cada una de las familias beneficiadas por el suministro de agua potable
TARJETA	Contiene la información de las tarjetas que van a ser usadas como método de control de acceso al suministro de agua
TANQUE	Los registros pertenecientes a los tanques que participan del sistema de suministro se encuentran en esta entidad

FALLA	Esta entidad es la encargada de almacenar la información necesaria para llevar un control de fallas en el sistema en cuanto a la conexión con el dispositivo que envía los datos en tiempo real.
TRÁFICO	Contiene la información de cada una de las transacciones que se realizan en el sistema, es decir, cada vez que un usuario perteneciente a una familia beneficiada hace uso del suministro de agua, se almacena un registro de dicha transacción en esta entidad.

A continuación, se da a conocer detalle de cada una de las relaciones presentadas en el diagrama entidad-relación de este proyecto:

- Usuario – familia***:*** Un usuario puede pertenecer únicamente a una familia. A una familia pueden pertenecer muchos usuarios.

Cabe resaltar que esta relación es aplicable a los usuarios con rol ‘Cliente’ ya que los usuarios con rol ‘Administrador’ y ‘Subordinado’ llevan el control del sistema, es decir, no pueden ser beneficiarios del suministro.

- *Familia – Tráfico:* Una familia puede tener muchos registros de tráfico. Un registro de la entidad tráfico pertenece a una sola familia.
- *Familia – Tarjeta:* A una familia se le asigna exclusivamente una tarjeta. Una tarjeta es asignada únicamente a una familia.
- *Tarjeta – Tanque:* Una tarjeta es inscrita a un tanque. En un tanque puede haber inscritas muchas tarjetas.

- *Tanque – Falla:* Un tanque puede tener muchos registros de fallas. Cada registro de fallas pertenece únicamente a un tanque a la vez.

*Usuario – Rol**:*** A un usuario se le es asignado un rol. Un rol puede ser asignado a muchos usuarios.

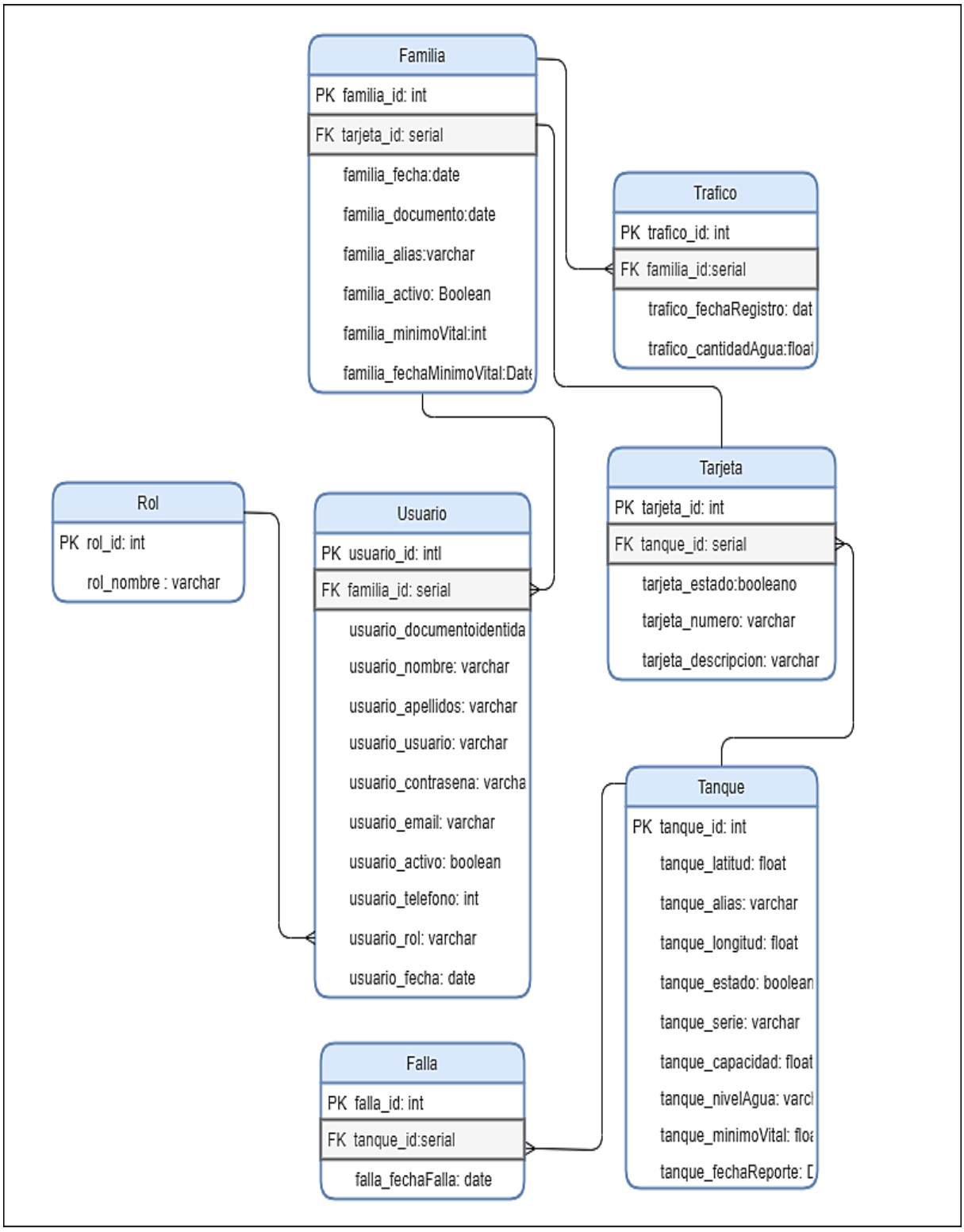
4.1.4 Diseño de la Base de Datos

Al disponer como soporte el diagrama de entidad-relación anteriormente desplegado, se lleva a cabo el diseño de la base de datos, estableciendo los atributos necesarios para cada entidad y resaltando las Foreing Key creadas dependiendo de cada relación establecida entre las entidades.

Cada uno de los atributos definidos tienen un alto nivel de importancia ya que son necesarios para ciertas transacciones que se implementan en el proyecto.

Figura 3.

Diagrama relacional



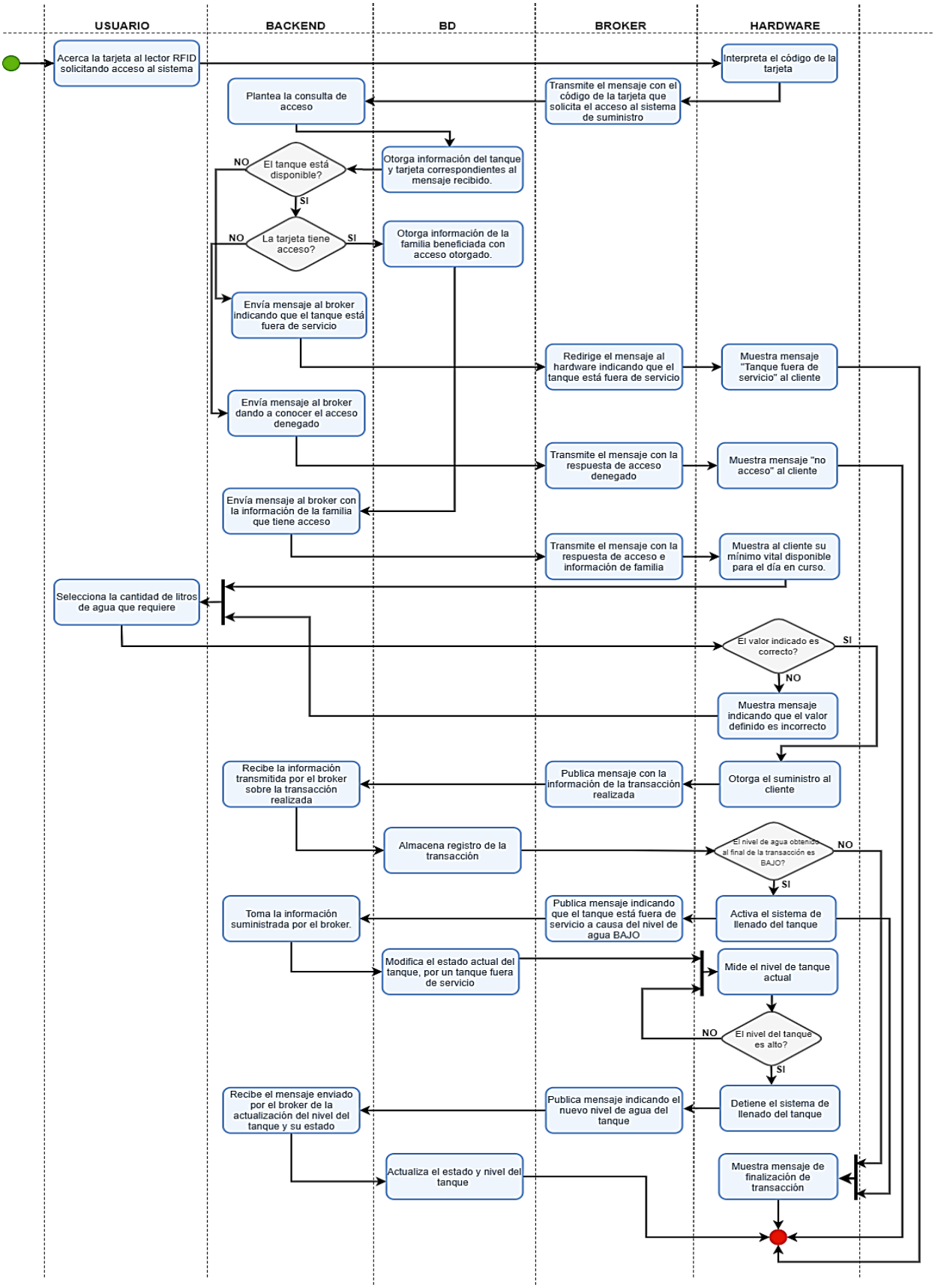
4.1.5 Diagrama de Actividades

Con el fin de dar a entender cada una de las acciones que se implementaron en este proyecto, se consideró el realizar diversos diagramas de lenguaje unificado de modelado (UML), incluyendo el diagrama de actividades. Este diagrama describe lo que sucede en el sistema. El objetivo principal de los diagramas que se muestran a continuación es dar a conocer la lógica y actividad a realizar paso a paso en el sistema planteado de una acción determinada.

En primer lugar, la Figura 4. muestra el flujo de comportamiento del sistema al instante en el que el usuario (integrante de una familia beneficiaria) solicita el suministro que requiere.

Figura 4.

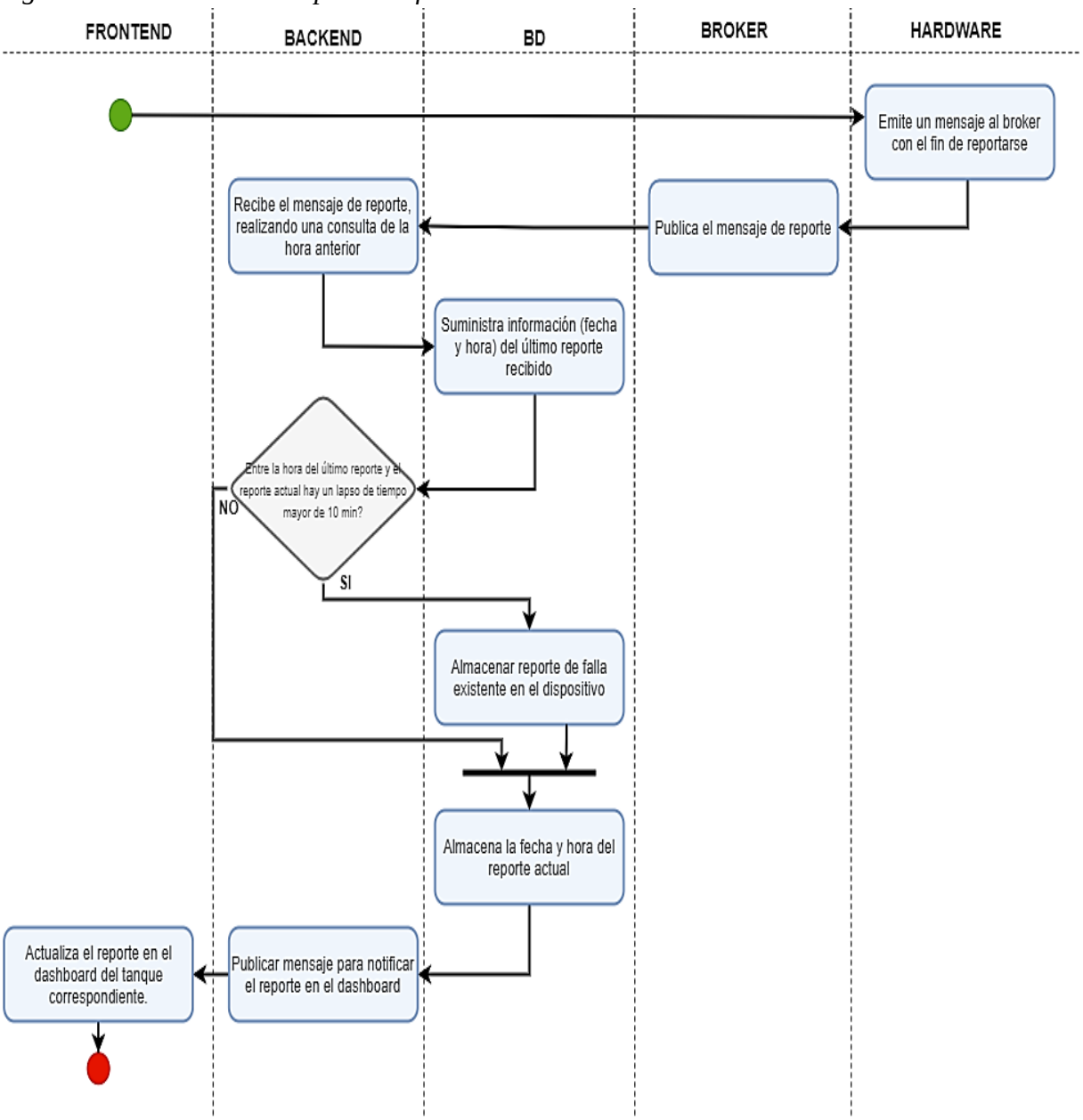
Diagrama de actividades: Usuario solicita suministro


USUARIO
BACKEND
BD
BROKER
HARDWARE
Acerca la tarjeta al lector RFID solicitando acceso al sistema
Interpreta el código de la tarjeta
Transmite el mensaje con el código de la tarjeta que solicita el acceso al sistema de suministro
Plantea la consulta de acceso
Otorga información del tanque y tarjeta correspondientes al mensaje recibido.
El tanque está disponible?
NO
SI
La tarjeta tiene acceso?
NO
SI
Otorga información de la familia beneficiada con acceso otorgado.
Envía mensaje al broker indicando que el tanque está fuera de servicio
Redirige el mensaje al hardware indicando que el tanque está fuera de servicio
Muestra mensaje "Tanque fuera de servicio" al cliente
Envía mensaje al broker dando a conocer el acceso denegado
Transmite el mensaje con la respuesta de acceso denegado
Muestra mensaje "no acceso" al cliente
Envía mensaje al broker con la información de la familia que tiene acceso
Transmite el mensaje con la respuesta de acceso e información de familia
Muestra al cliente su mínimo vital disponible para el día en curso.
Selecciona la cantidad de litros de agua que requiere
El valor indicado es correcto?
SI
NO
Muestra mensaje indicando que el valor definido es incorrecto
Recibe la información transmitida por el broker sobre la transacción realizada
Publica mensaje con la información de la transacción realizada
Otorga el suministro al cliente
Almacena registro de la transacción
El nivel de agua obtenido al final de la transacción es BAJO?
NO
SI
Toma la información suministrada por el broker.
Publica mensaje indicando que el tanque está fuera de servicio a causa del nivel de agua BAJO
Activa el sistema de llenado del tanque
Modifica el estado actual del tanque, por un tanque fuera de servicio
Mide el nivel de tanque actual
El nivel del tanque es alto?
NO
SI
Recibe el mensaje enviado por el broker de la actualización del nivel del tanque y su estado
Publica mensaje indicando el nuevo nivel de agua del tanque
Detiene el sistema de llenado del tanque
Actualiza el estado y nivel del tanque
Muestra mensaje de finalización de transacción

A continuación, se muestra el diagrama de actividades del reporte de fallas, este reporte se lleva a cabo gracias a un mensaje transmitido por el dispositivo cada cierto tiempo, en este caso cada 10 minutos, con esto hace un reporte de su conexión con el bróker ayudando a identificar si la conexión es exitosa o presenta inconvenientes.

Figura 5.

Diagrama de actividades: Reporte de fallas



4.2 Selección de Arquitectura

4.2.1 Hardware

La selección de componentes para el hardware se realizó de la mano de los requerimientos funcionales y no funcionales estipulados anteriormente en el inciso 4.1.1. De acuerdo con ellos se procedió al análisis de las diversas opciones que existen en el mercado y posteriormente a la selección, basándonos en las necesidades que debían suplirse. Por lo tanto, a continuación, podremos constatar los componentes que se seleccionaron y la tarea que se les delego en este proyecto.

4.2.1.1 Control de Acceso, dentro de las necesidades del proyecto destaca el tener un control de acceso contando con el reconocimiento de los usuarios pertenecientes a las familias beneficiadas que van a consumir el vital líquido, y de esta manera, lograr un registro de cada una de las transacciones futuras.

Al realizar la selección de dispositivo para el reconocimiento de identidad y control de acceso se contemplaron variedad de opciones dentro de las cuales destacan

- Módulo lector de huellas dactilares: El dispositivo es ideal para la tarea a realizar ya que está en capacidad de identificar al usuario de forma directa por medio del análisis de las huellas dactilares.
- Lector RFID inalámbrico: El control de acceso e identificación de usuarios en este dispositivo se realiza por medio de un llavero o tarjeta que emite una señal al momento de acercarlo al lector RFID, el lector se encarga de interpretar el mensaje e imprimir un código establecido que será consultado en la base de datos para definir el acceso de dicha tarjeta o llavero al sistema.

Cada una de las opciones que se tomaron en cuenta cumplen con el objetivo principal anteriormente expuesto que es el control de acceso de los diversos beneficiarios. Sin embargo, al considerar la situación actual, en medio de la pandemia de COVID-19 se busca un método en el cual se pueda evitar el contacto con dispositivos constantemente en caso de no ser necesario, por tal razón, el dispositivo que se implementó en el proyecto es el lector RFID.

4.2.1.2 Selección y Visualización de Información

- Pulsadores: la selección de información en este sistema de suministro es de gran importancia ya que debe ser amigable y de fácil uso para los usuarios beneficiarios. Debido a que existen factores que influyen como las diferencias de edad entre los usuarios o los conocimientos tecnológicos que tengan. Por lo tanto, se escogió para esta selección de información los pulsadores, los cuales cuentan con la versatilidad de ser de fácil uso y además a nivel de conexión al sistema también brindan facilidad y simplicidad cumpliendo con las características necesarias para ser implementados como botones de selección y confirmación de información como lo es la cantidad de agua que el beneficiario desea retirar.
- Pantalla LCD: La correcta visualización de la información es primordial, ya que otorga al usuario un mejor entendimiento y manejo del sistema de suministro, por lo tanto, la pantalla LCD de 20x4 otorga una buena imagen y tamaño. Teniendo en cuenta que la presentación de este sistema se realizara a escala de maqueta.

En esta pantalla se podrá visualizar la información que se le otorgue al usuario como lo es su mínimo vital disponible, la familia a la que pertenece, el nivel del tanque en el cual se encuentra registrado y la autorización de acceso al suministro.

4.2.1.3 Medir el Flujo de Agua, la medición del flujo de agua es imprescindible ya que los beneficiarios del abastecimiento digitarán la cantidad de líquido que requieren definida en litros, siendo necesario el control en la expulsión del suministro del sistema establecido. De esta forma, nace la necesidad de implementar un sensor de flujo de agua que permita obtener un valor puntual de la proporción de agua que se va suministrando, con el fin de otorgar únicamente la cantidad solicitada por el beneficiario.

El sensor de flujo de agua es un instrumento compuesto principalmente por una turbina que es activada en cuanto ingresa caudal de agua al dispositivo, está turbina está unida a un imán encargado de activar un sensor de efecto Hall que emite señal de pulsos, esta cantidad de pulsos varía según la rotación de la turbina, y dicha rotación depende de la tasa de flujo en la que pasa el caudal. Los pulsos detectados son convertidos al valor puntual de flujo de agua que pasa por el dispositivo, esta operación es asignada al Arduino o controlador existente en el sistema.

4.2.1.4 Medir el Nivel de Agua en los Tanques de Almacenamiento, usualmente en los tanques de agua se implementan los flotadores, los cuales trabajan mediante una boya conectada a una válvula por medio de un caño. Este sistema funciona detectando el movimiento que realiza la boya, ya que si se encuentra en posición horizontal la válvula que permite el paso del agua se cierra. En cambio, si la posición es vertical se abre la válvula dejando ingresar el líquido al tanque. Este sistema no es adecuado para los requerimientos ya que no suministra información constante del nivel de agua a medida que el líquido va disminuyendo.

Por lo tanto, el sensor ultrasónico fue seleccionado para medir el nivel de agua en tiempo real, su funcionamiento se basa en medir el tiempo que tarda en viajar una onda ultrasónica desde que sale hasta que regresa al punto de partida, con este tiempo y la velocidad del sonido se calcula

la distancia que hay entre el sensor y el objeto con el cual choco la onda. Con esta distancia se puede calibrar y establecer los niveles que tendrá el tanque y a la distancia que se debe encontrar el agua para reportar un nivel ya sea alto bajo o medio.

La fiabilidad en su medición y fácil implementación son características con la que cuenta este sensor. Además, en la industria de petróleo y aguas residuales es altamente implementado para la medición de flujos, brindando seguridad y fiabilidad en su trabajo; por lo tanto, se optó por utilizarlo para calcular el nivel de agua en los tanques de almacenamiento comunitarios , ya que de este nivel dependerá que el sistema de suministro trabaje de manera correcta y eficiente para los usuarios beneficiarios, controlando y monitoreando el sistema de entrega de agua previniendo que no se les brinde un servicio constante y eficiente.

4.2.1.5 Permitir o restringir el flujo de agua, una vez delimitada la cantidad de suministro por otorgar, se busca un mecanismo con el cual se regule la provisión del líquido según sea necesario. Dentro de las opciones se podría encontrar:

- Bomba de agua: Es un dispositivo empleado para enviar el líquido con cierta presión de un lugar a otro en el momento en el que detecta corriente.
- Electroválvula: Es un artefacto que facilita o bloquea el paso de agua. La electroválvula cuenta con una bobina solenoide que autoriza o rechaza el paso del líquido al recibir corriente alterna. Existen electroválvulas normalmente cerradas y normalmente abiertas. Las electroválvulas normalmente cerradas son aquellas que cuando no reciben corriente no permite el paso de agua, una vez detecta corriente en su sistema, permite el paso. Por otra parte, se presenta la electroválvula normalmente abierta, que a diferencia de la anteriormente expuesta permite el paso del líquido hasta el momento que recibe corriente y bloquea el acceso de agua.

Considerando las opciones mencionadas, se utilizó en el proyecto una bomba de agua para efectuar esta labor, la elección se basó principalmente en comodidad a la hora de construir la maqueta del prototipo dado que las electroválvulas consultadas en el mercado en ese entonces solicitaban presión para su correcto funcionamiento.

Al tomar como referencia el uso de una bomba de agua para ejecutar una tarea tan importante como lo es el control en la expulsión del suministro, destaca la necesidad de dominar el paso de corriente, por lo que se empleó un relé que actúa en conjunto con la bomba de agua seleccionada como un interruptor gracias a una bobina y un electroimán que permite o prohíbe el paso de corriente.

4.2.1.6 Placas Programables, en el mercado existen diversas placas programables las cuales son implementadas en el desarrollo de todo tipo de proyectos electrónicos o que cuentan con componentes electrónicos. las placas más utilizadas son las Arduino y Raspberry Pi que, aunque cuentan con similitudes en su aspecto son dos productos que fueron diseñados para distintas finalidades.

Arduino es una plataforma para la creación de código basado en software y hardware. En el mercado se pueden encontrar diversos tipos de placas, accesorios y aplicaciones compatible. Además, ofrece la plataforma IDE (Entorno de Desarrollo Integrado) el cual es un entorno de programación en el cual se pueden crear aplicaciones para las diversas placas y con diversa usabilidad.

Por otro lado, Raspberry Pi es un ordenador de tamaño pequeño, el cual está compuesto por una base que soporta distintos componentes de un ordenador como procesadores y memoria

RAM. Tiene un bajo costo, permite la conexión de dispositivos como pantallas, teclados e incluso televisores, también brinda la posibilidad de realizar conexiones a la red a través de puerto Ethernet y dependiendo del modelo incluso conexión wifi o bluetooth. Una de sus principales características es que permite programar y compilar programas.

Arduino y Raspberry Pi fueron creados con propósitos y conceptos diferentes, ya que el primero fue diseñado para brindar versatilidad y facilidades de conexiones debido a que cuenta con diversos tipos de entradas, por otro lado, la Raspberry Pi tiene más potencia de cálculo y cuenta con sistema operativo.

La selección entre estas dos opciones se fundamentó de acuerdo con la versatilidad y agilidad necesaria para el procesamiento de los programas que se desean ejecutar en estas placas, por lo tanto, se seleccionó Arduino ya que cuenta con un menor tiempo de ejecución y respuesta en comparación con la Raspberry Pi debido a que no requiere un sistema operativo para ejecutar los programas, además de que existen mayor cantidad de herramientas y plataformas compatibles con estas placas

4.2.1.7 Conexión con el Servidor, la conexión al servidor se implementa con el propósito de almacenar y consultar con mayor seguridad los datos de usuarios e información de los distintos componentes de la plataforma como lo son las tarjetas, tanques y familias registradas.

Arduino tiene diversas placas que se conectan de diferentes maneras, en este caso, se quería una conexión por wifi que fuera eficaz y que tuviera un tiempo de transmisión y recepción de información corto.

Por medio de esta conexión se realizarán consultas, almacenamientos, envió, transmisión y recepción de información de manera bidireccional, es decir, el servidor recibirá información y también enviará por medio de esta conexión.

La ESP8266 se seleccionó ya que, cuenta con conexión por wifi, tiene un bajo consumo, es fácil de utilizar, y tiene un valor económico en el mercado asequible. Además, ya se contaba con el dispositivo.

4.2.1.8 Gestión de Puertos en el Montaje del Circuito, en medio del diseño y planteamiento del proyecto se percató del número de entradas y salidas necesarias para la construcción del circuito a emplear en el prototipo de la maqueta a escala del sistema, número de entradas y salidas que no proporciona la tarjeta ESP8266, encargada del envío de información en tiempo real.

Una vez descubierta esta necesidad se indaga para llegar a una solución, que en este caso es la implementación de un modelo de protocolo de comunicación Maestro-esclavo. Para llevar a cabo dicha propuesta, se aplicó esta comunicación entre el ESP8266 y un Arduino MEGA el cual nos proporciona una gran cantidad de entradas y salidas analógicas y digitales supliendo la necesidad de estas.

4.2.1.9 Sistema de Llenado de los Tanques de Almacenamiento, el sistema de llenado de los tanques de almacenamiento se plantearon con un estado por defecto inactivo el cual cambiara de acuerdo con el nivel de agua que tiene el tanque, para que este sistema se active este nivel deberá ser bajo.

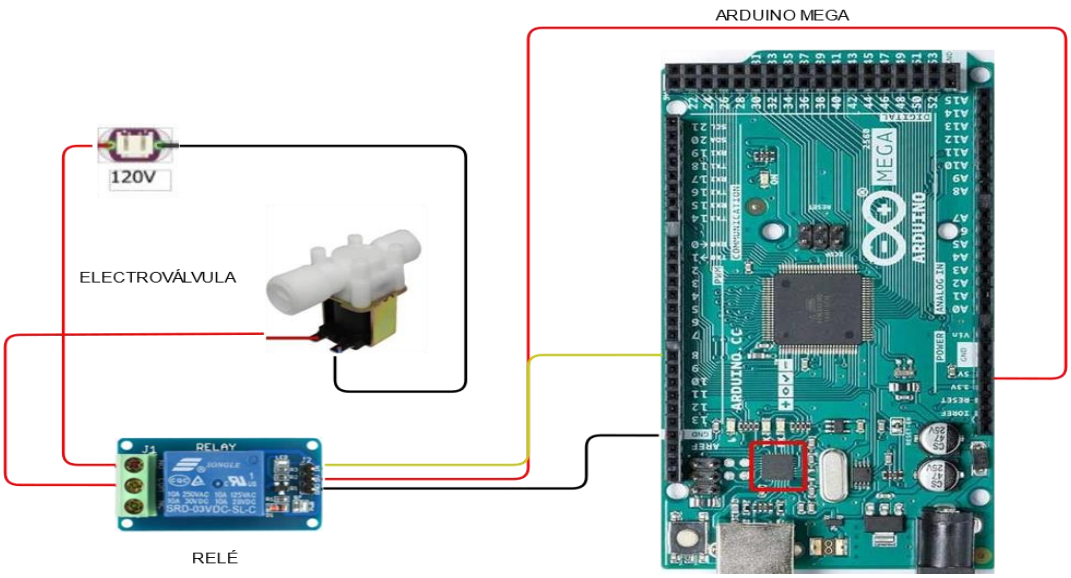
Para llevar a cabo lo planteado anteriormente, se seleccionó una electroválvula solenoide cuyo fin es controlar el flujo del líquido mediante el suministro de corriente eléctrica. Además, se incorporará un relé a este sistema de llenado.

El relé es un interruptor eléctrico, el cual permite el paso de la corriente eléctrica cuando se encuentra cerrada e interrumpe la corriente cuando está abierta, esto se acciona de forma eléctrica y no manualmente; la función que tendrá este interruptor es controlar el paso de corriente hacia la electroválvula de acuerdo a si el tanque necesita que se realice la activación del sistema de llenado.

La conexión entre estos dos componentes como lo son la electroválvula y el relé se puede observar en el diagrama del sistema.

Figura 6.

Diagrama de conexión del sistema de llenado de los tanques de almacenamiento



4.2.2 Software

La selección de componentes para el hardware se realizó de la mano de los requerimientos funcionales y no funcionales estipulados anteriormente en el inciso 4.1.1, se realizó un análisis de las diferentes opciones tecnológica que existen en el mercada de servicio cloud, lenguajes de programación y frameworks para de esta manera analizar las ventajas que prestan cada una de estas opciones e implementar en el proyecto la que mejor se adapte.

4.2.2.1¿Por Qué Amazon Web Service?, al seleccionar una opción favorable en cuanto al entorno en el cual se despliega nuestro sistema de Internet de las cosas, es decir, el servidor a utilizar. Se tiene en cuenta las tareas principales desempeñadas por este, dentro de las que destacan:

- Hosting: Este permite alojar la página o aplicaciones web, facilitando el acceso a los usuarios desde cualquier dispositivo. De esta manera, los usuarios pueden interactuar con la aplicación cómodamente supliendo sus necesidades.
- Broker: Es un sistema encargado de recibir los datos otorgados por los diversos sensores o dispositivos participantes del montaje, toma estos mensajes y los retransmite a las aplicaciones que anteriormente han sido suscritas para su recepción. Este componente tiene una relación directa con las aplicaciones y bases de datos en ciertos casos. Adicionalmente, es el responsable de hacer llegar a los dispositivos hardware ordenes que se emiten desde una aplicación.
- Base de datos: La base de datos es la encargada de almacenar la información y comportamiento que generan tanto dispositivos como las aplicaciones.
- Aplicación (Frontend, backend): Son los encargados de dar funcionalidad y lógica

a la aplicación en conjunto. El backend emite todos los procedimientos complejos y simples a implementar, así como el tratado de datos. El frontend se encarga de la interacción con el cliente, ofreciendo una interfaz amigable y productiva.

Una vez se tiene clara cada tarea a desempeñar, se considera la implementación del sistema desde cero, con el fin de tener un contacto directo con cada paso en medio de la creación de un proyecto IoT.

El VPS finalmente seleccionado es un servicio EC2 ofrecido por Amazon Web Service, la elección se llevó a cabo al resaltar ciertas características dentro de las cuales destacan el costo acorde con el servicio prestado, mayor tolerancia de errores, mejor disponibilidad, libre elección en cuanto al sistema operativo y leguaje de programación, base de datos, permitiendo así el cargue y despliegue de diversos servicios y software requeridos.

4.2.2.2 Base de Datos, en la actualidad las bases de datos se dividen en relaciónales (SQL) y no relacionales (NoSQL), para seleccionar el tipo de bases de datos que queremos aplicar en el proyecto debemos analizar las ventajas y desventajas que ofrecen cada una de las opciones.

Las bases de datos relacionales (SQL) cuentan con un mayor soporte técnico debido a su antigüedad, los datos deben cumplir requisitos de integridad, utilizan identificadores los cuales ayudan a una mejor organización de la información en partes pequeñas, también cuentan con una mayor capacidad de almacenamiento y son menos vulnerables ante fallas. Además de que cuentan con una estructura establecida; si esta estructura es modificada, podría perjudicar todo el sistema ya que las tablas se encuentran relacionadas.

Por otra parte, tenemos las bases de datos no relacionales (NoSQL) las cuales cuentan con facilidad en su desarrollo, no poseen identificadores que sirvan de relación entre conjuntos de

datos, sin embargo, también existe una menor seguridad al ejecutar consultas a este tipo de base de datos

Para este proyecto se seleccionó las bases de datos relacional, ya que de acuerdo con la estructura del proyecto la cual cuenta con usuarios, familias, tarjetas, tanques y tráfico, es sumamente importante contar con una estructuración sólida, con identificadores en cada tabla que establezcan las distintas relaciones entre los componentes y además poder obtener un soporte amplio del sistema. Todo lo expuesto anteriormente son las principales características de las bases de datos SQL y por lo tanto esta fue nuestra mejor opción.

4.2.2.3 Backend, el desarrollo backend para un proyecto es de gran importancia, esto se debe a que es la parte responsable de un buen rendimiento, es el encargado de interactuar con los datos almacenados en la base, verificar sesiones de usuarios, otorgar soluciones a un gran número de solicitudes realizadas por el frontend, entre otras asignaciones importantes. Aunque para el cliente estos procesos no tienen gran visibilidad, son quieres proporcionan los datos para que se logre organizar las vistas de la mejor manera posible y así, tener como resultado una buena experiencia de usuario.

En cuanto a la elección de tecnología cabe resaltar la importancia actual en el mercado de NodeJs y SpringBoot, por esta razón, son las tecnologías que se estudiarán a continuación para lograr una buena elección.

NodeJs cuenta con diversas herramientas que le impulsan a ser una de las mejores opciones en cuanto a maneras de sobrellevar todo el procesamiento del backend. Dentro de las ventajas resalta la aceptación en variedad de servidores entre los cuales se encuentran Mac OS, Unix, Microsoft Windows. Su rendimiento es otro factor a favor, NodeJs permite desarrollar proyectos

con muy alto rendimiento, gran calidad, disminuyendo el margen de errores técnicos. Adicionalmente, es idóneo para manejar aplicaciones que presenten alto tráfico tanto de usuarios como de eventos.

Express.js es una de las herramientas destacables de NodeJs, es una infraestructura web, que además de ser rápida y flexible, permite ordenar su código de la mejor manera, habiendo un mejor rendimiento en sus aplicaciones. Esta infraestructura trae consigo un conjunto de características para implementar en distintas aplicaciones web y móviles. Por otra parte, pero no menos importante cabe resaltar el nivel de utilidad y rendimiento que se obtiene gracias a la implementación de HTTP y middleware.

Spring es un framework que presenta variedad de ventajas dentro de las cuales se destaca principalmente la facilidad en el momento de usar muchas dependencias, o su lenguaje base que es Java. Sin embargo, se logró determinar ciertas desventajas como son la alta utilización de memoria en algunos casos y la repetición de código.

Dependiendo del enfoque aplicado al proyecto, podemos decidir cuál tecnología es la adecuada para utilizar, sin embargo, para este proyecto se seleccionó NodeJs y express.js para el manejo del backend en base a todas las ventajas anteriormente nombradas que este proporciona.

4.2.2.4 Frontend, el desarrollo del frontend de la aplicación web implica una selección de lenguaje de programación, frameworks o librerías que se quieran implementar, en este proceso se tuvieron en cuenta las tecnologías que en la actualidad están generando un auge en su implementación para el desarrollo de aplicaciones. Este análisis da como resultado un framework y una librería los cuales actualmente están muy bien posicionados tanto en las empresas como en el desarrollo de aplicaciones los cuales son Angular y React.

Para realizar una selección entre estos dos se tuvieron en cuenta las distintas ventajas y desventajas que ofrecen cada uno de ellos de esta manera se puede realizar una selección óptima, eficaz y acorde con los requerimientos del proyecto.

Angular es un framework de JavaScript para desarrollo web y móvil, el cual se encuentra basado en TypeScript. Mayormente es implementado en el desarrollo de aplicaciones web de página única (páginas que se recargan al instante) o de varias páginas. El uso de este framework generalmente no requiere de la instalación de librerías adicionales para su desarrollo, ya que su paquete de instalación es completo.

Este framework cuenta con una mayor complejidad de aprendizaje, debido a que tiene mayor grado de conceptualización que otros frameworks o librerías que se encuentran en el mercado tecnológico, sin embargo, esto no fue limitante ya que tiene una documentación y soporte técnico muy completo. Además, cuenta con material de diseños preconstruidos lo cual facilita el diseño de las interfaces de usuario. Las ventajas que ofrece este framework para el desarrollo de aplicaciones web son:

- Estructura modular: este framework implementa módulos los cuales ayudan a una mejor organización de la aplicación, separándolos en componentes. De esta manera, permite que todos los elementos sean ejecutados eficaz y rápidamente
- Estructura basada en componentes: brinda la posibilidad al programador de reutilizar estos componentes para construir cualquier aplicación e incluso implementar componentes ya establecidos por Angular
- Facilidad en el mantenimiento de software: el mantenimiento y mejora de los componentes de manera sencilla se debe a la estructura de componentes que manejan los proyectos de Angular.

React es una librería de JavaScript que se utiliza para el desarrollo de aplicaciones web de página única o de varias páginas. Las aplicaciones desarrolladas con esta librería deben instalar librerías adicionales como como Redux, React Router, entre otros para su implementación. Funciones como la validación de formulario también requieren de la instalación de librerías; cuenta con ciertas ventajas como lo son:

- Estructura basada en componentes: brinda la posibilidad al programador de reutilizar estos componentes para construir cualquier aplicación e incluso implementar componentes ya establecidos por Angular
- React native: facilita el desarrollo de aplicaciones móvil implementando el código JavaScript de la aplicación web

Angular es el framework en el cual se desarrollará el frontend del proyecto ya que cuenta con muchas ventajas, soporte técnico, además de ser muy completo ya que no se necesitan en su gran mayoría de la instalación de librerías adicionales a las que ya vienen estipuladas al momento de instalar angular.

La arquitectura en la cual se basa las aplicaciones desarrolladas con este framework consiste en módulos, componentes, rutas, servicios decoradores personalizados o establecidos como los pipes entre otros.

Los módulos declaran un contexto de compilación para los componentes, estos pueden asociar componentes que tienen código relacionado, para formar unidades funcionales. En estos módulos se pueden importar funcionalidades de otros módulos o exportar sus funcionalidades. También toda aplicación generada con angular cuenta con un módulo raíz llamado *App Module* el cual provee el mecanismo de arranque de la aplicación.

Los componentes hacen parte de toda aplicación de angular ya que por lo menos tiene uno, cada componente define una clase que contiene información y lógica, la cual se vincula a un archivo HTML el cual es el que refleja los componentes de las aplicaciones como vistas de usuario

Servicios e inyección de dependencias, toda la lógica e información que no se encuentra asociada de manera directa a una vista y que se requiere utilizar en diversos componentes pueden ser escritas en un servicio, los cuales son exportados como clases. Para la implementación de estos servicios se debe realizar la inyección de dependencia en el componente en el cual se desea implementar dicho servicio

La arquitectura expuesta anteriormente junto a las diversas ventajas que ofrece este framework, además, de que ya se contaban con conocimientos previos, fundamentan su elección para la implementación y desarrollo del frontend de la aplicación web del proyecto.

4.2.2.5 MQTT, para Internet de las cosas es necesaria la implementación de un protocolo de comunicación eficiente y con un liviano modo de transmitir datos. MQTT cumple con estos requerimientos, es un protocolo creado por IBM predilecto para IoT. La telemetría, aplicaciones de chat, comunicación entre aplicaciones y servicios de notificación son algunas de las formas en las que se emplea este protocolo.

El bróker MQTT es el encargado del registro de los suscriptores a cada tópico establecido, administrando la gestión de estos. Se destaca como característica principal un desacoplamiento tridimensional, esto quiere decir que existe un *desacople en espacio,* porque quienes publican y quienes se suscriben no se conocen entre sí necesariamente, *desacople en tiempo* ya que estos mensajes no se entregan en tiempo real obligatoriamente, *y desacople en sincronización*, así pues,

no hay bloqueos en los mensajes y se pueden implementar diversidad de beneficios de este protocolo sin enviar o recibir mensajes.

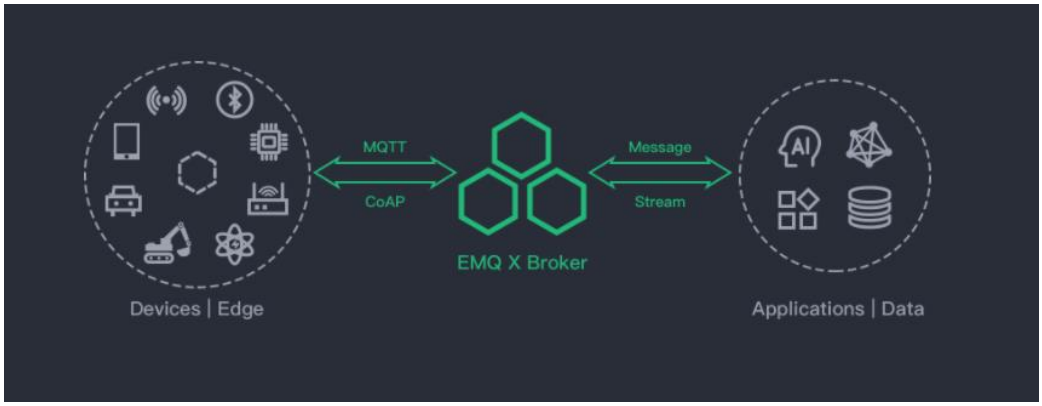
Tópico es una de las definiciones más importantes de este protocolo, es una “etiqueta” que permite clasificar los mensajes a emitir/recibir, facilitando la comunicación del sistema.

Dentro de las acciones principales del MQTT se encuentran

- CONNECT, es puesto en práctica para la conexión al protocolo de comunicación, se debe conectar quien publica los mensajes y quien desea suscribirse para recibirlos.
- SUBSCRIBE, con esta acción se lleva a cabo la suscripción a un tópico, es decir, se requieren y reciben únicamente los mensajes enviados con el tópico específico al que se realiza la suscripción.
- UNSUBSCRIBE, se utiliza para cancelar la suscripción a un tópico determinado, una vez se emplee esta acción, no se recibirán mensajes etiquetados con dichos tópicos.
- PUBLISH, es la acción que se emplea para publicar un mensaje, se establece tanto el tópico como el mensaje a enviar.

Figura 7.

Conexión EMQ X.



Tomada de https://www.emqx.io/

Para lograr la implementación del servidor al broket MQTT sostiene variedad de opciones, dentro de las que destacan mosquitto que es una comunidad grande, proporcionando suficiente documentación al respecto. Por otra parte, mosca es otra opción llamativa para iniciar con el mundo de Internet de las cosas y protocolos de comunicación. Finalmente, se encuentra EMQ, que es altamente escalable, es el intermediario de mensajes MQTT distribuido de código abierto más utilizado, y entre sus fundadores hay personal de importantes empresas como son Huawei, IBM y Amazon; adicionalmente, tiene sistema gestor de clustering otorgándole mayor eficiencia, es masivamente escalable y tiene alta disposición por lo cual fue seleccionado para este proyecto.

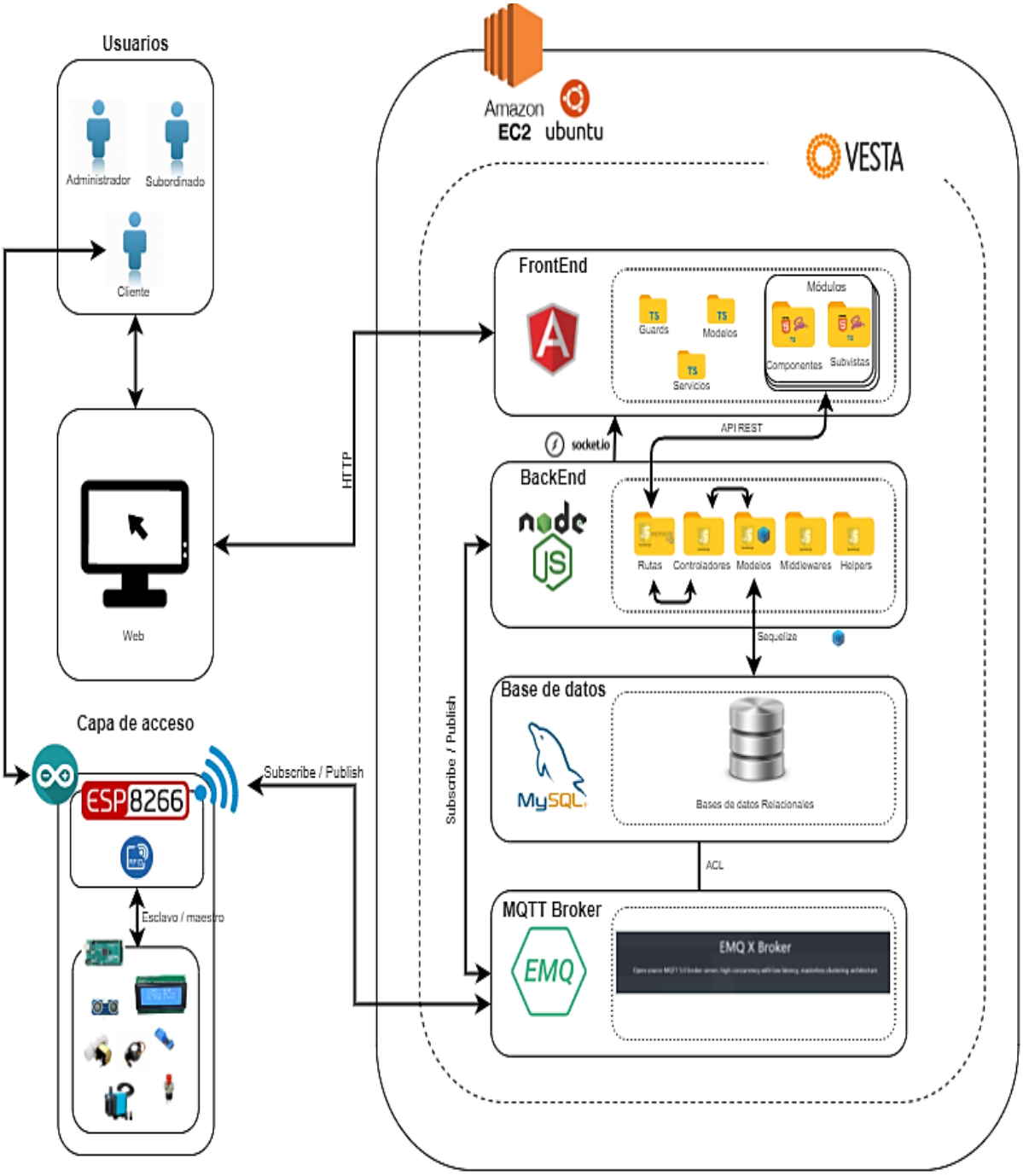
4.2.3 Esquema de Arquitectura Seleccionada

Teniendo en cuenta cada una de las tecnologías anteriormente seleccionadas, tanto hardware como software se plantea el siguiente esquema de arquitectura del proyecto, partiendo principalmente de los usuarios que interactúan en él.

En el diagrama se logra observar la comunicación entre capas resaltando las tecnologías a usar, así como la estructura del Backend y Frontend, base de datos y bróker desplegado en el panel de control web hosting Vesta. Todo esto parte de un servidor virtual privado otorgado por el servicio EC2 de Amazon Web Service.

Figura 8.

Arquitectura del prototipo



4.3 Desarrollo del Prototipo

Una vez seleccionada cada uno de los componentes y arquitectura a implementar, se procedió al desarrollo del prototipo. En el transcurso de esta etapa se fue aplicando diversidad de pruebas al sistema con el fin de garantizar un buen desarrollo de este.

4.3.1 Creación del Entorno en Amazon Web Service

Para iniciar la creación del entorno fue necesario crear una cuenta en el panel de Amazon Web Service seleccionando la zona más cercana disponible que en este caso fue *Sao Pablo, South America*. Una vez establecida la cuenta, se accede al servicio EC2, capa que permite crear el VPS necesario para el proyecto.

Inicialmente se creó y se ejecutó una instancia, de esta manera se procedió a instalar el sistema operativo a ejecutar que en este proyecto fue una máquina *Ubuntu Server 18.0.4 LTS.* **Figura 9**.

Grupo de seguridad establecido para el proyecto

Reglas de entrada (26) — Editar

Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional
HTTP	TCP	80	0.0.0.0/0	HTTP
HTTP	TCP	80	::/0	HTTP
TCP personalizado	TCP	8083	181.129.184.236/32	vesta
TCP personalizado	TCP	18083	0.0.0.0/0	emqx
TCP personalizado	TCP	18083	::/0	emqx
TCP personalizado	TCP	8883	0.0.0.0/0	EMQX
TCP personalizado	TCP	8883	::/0	EMQX
TCP personalizado	TCP	8090	0.0.0.0/0	emquex
TCP personalizado	TCP	8090	::/0	emquex
TCP personalizado	TCP	1883	0.0.0.0/0	EMQX
TCP personalizado	TCP	1883	::/0	EMQX
HTTPS	TCP	443	0.0.0.0/0	HTTPS
HTTPS	TCP	443	::/0	HTTPS
TCP personalizado	TCP	8093	0.0.0.0/0	EMQX
TCP personalizado	TCP	8093	::/0	EMQX

Seguido a esto, se creó el grupo de seguridad en donde se definió cada puerto a utilizar, restringiendo el acceso únicamente a nuestra IP en algunos puertos para una mayor seguridad.

Figura 10.

Grupo de seguridad establecido para el proyecto

TCP personalizado	TCP	12000 - 12100	0.0.0.0/0	FTP PASIVO
SSH	TCP	22	0.0.0.0/0	SSH
SSH	TCP	22	::/0	SSH
TCP personalizado	TCP	21	0.0.0.0/0	FTP
TCP personalizado	TCP	21	::/0	FTP
TCP personalizado	TCP	8094	0.0.0.0/0	emqx
TCP personalizado	TCP	8094	::/0	emqx
MYSQL/Aurora	TCP	3306	0.0.0.0/0	bd
MYSQL/Aurora	TCP	3306	::/0	bd
TCP personalizado	TCP	3000	0.0.0.0/0	htt
TCP personalizado	TCP	3000	::/0	http

A continuación, se procedió a reservar y asociar una IP fija con la instancia creada, con el propósito de hacer la asignación al VPS, si se reinicia el servidor la IP no se restaura y sigue siendo la misma.

Con relación a Vesta como panel web hosting, se realizó la instalación y configuración por medio de Putty, descargado anteriormente ya que la conexión al servidor se llevó a cabo en un equipo con sistema operativo Windows.

Se debe seleccionar un dominio ya que otorga distintivo a la plataforma y facilita el acceso de los usuarios a la misma. El dominio elegido para el proyecto es *www.proyectoiot.gq* el cual se obtuvo mediante la página freenom.com con una prueba gratuita por cierto periodo de tiempo. Una vez asociado el dominio a la IP y asignado el puerto respectivo para el panel de Vesta se procede a hacer los ajustes necesarios para el servidor FTP con el objeto de tener disponible el protocolo para el subir o transferir los archivos al hosting y así quede a disposición de los usuarios.

En cuanto a la creación de la base de datos en el hosting y el tener a disposición está información, se dirigió hacía el panel de Vesta ubicado en el puesto 8083 de nuestro dominio http://proyectoiot.gq:8083 y se creó la base de datos relacional.

4.3.2 Planteamiento de tópicos

En vista de las funcionalidades definidas, se plantearon ciertos tópicos que suplen las necesidades del sistema.

Tabla 21

Estructura de tópicos establecidos

#	Emisor	Tópico	Mensaje
1	ESP8266	tanque_serie/**access_query**	tarjeta_numero
2	NodeJS	tanque_serie/**command**	tanqueout noacceso alias/minimoVital/tanque_capacidad/tanque_nivelAgua/familia_id/tanque_id
3	ESP8266	tanque_serie/**datosAgua**	minimoVital/trafico_cantidadAgua/tanque_nivelAgua/familia_id/tanque_id
4	ESP8266	tanque_serie/**nivelTanque**	tanque_nivelAgua
5	ESP 8266	tanque_serie/**reporte**	activo

A continuación, se otorga una descripción con el fin de contextualizar los escenarios en los que se implementa cada tópico expuesto en la tabla anterior.

Tópico #1. Es el enviado en el momento en el que un posible beneficiario solicita acceso al sistema por medio de la tarjeta RFID, el sensor toma el valor de la tarjeta y este es el mensaje enviado por el tópico establecido.

Tópico #2. Este tópico tiene definido tres tipos de mensajes según cual sea la situación que se presente. Es enviado por el backend a los dispositivos.

En primer lugar, está el mensaje ‘tanqueout’ que se envía en caso de que el tanque al que se quiere acceder este fuera de servicio a causa del nivel de agua que dispone. Seguido a este, se encuentra el mensaje ‘noacceso’ enviado en el momento en que la tarjeta que solicita acceso no tiene permiso para solicitar el suministro en ese tanque específico. Por último, se encuentra un mensaje con información completa de la familia que solicita el canje de suministro.

Tópico #3. El tópico es enviado por el ESP8266 tan pronto se termina el proceso de suministro de agua. Una vez la transacción es exitosa, el dispositivo envía la información necesaria para registrar en la base de datos el tráfico.

Tópico#4. Este tópico envía el nivel del tanque actual. Es enviado en momentos estratégicos con el fin de mantener al tanto al backend del nivel de agua del tanque actual.

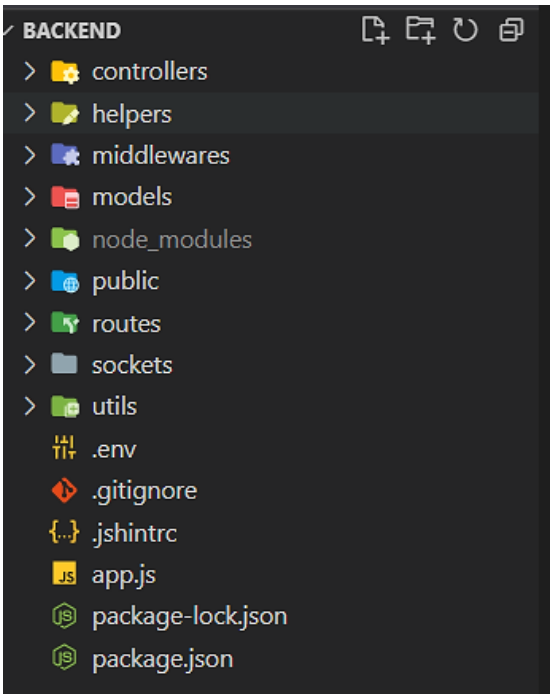
Tópico#5. Tópico implementado cada 10 minutos, ya que es el tiempo que se estableció para pasar el reporte de actividad y conexión del tanque.

4.3.3 Creación del proyecto NodeJs

En la creación del proyecto de backend con NodeJS, se tuvo en cuenta una de las estructuras más utilizadas en el desarrollo, MVC(Modelo Vista Controlador), esta es una arquitectura que busca separar el código según la responsabilidad otorgada a cada uno de los archivos. En primer lugar, el modelo es quien se encarga de gestionar toda la información que lo requiera. Por otra parte, la vista que es lo que el usuario puede ver, es decir, con lo que el interactua, en este caso la vista es la respuesta JSON que se envía al proyecto de Angular para ser mostrado. Finalmente, se define el controlador que es el archivo en donde se modifica o adapta los datos de tal manera que sea interpretado tanto por el modelo, como por la vista.

Figura 11.

Proyecto backend



Adicional a esto, se crearon ciertas carpetas que participan activamente en todo el proyecto, como son:

Helpers, En esta carpeta se encuentra algunas consultas complejas a la base de datos que se emplearon para la visualización del dashboard, validaciones necesarias, generación del JWT(Json Web Token). Está enfocada en funcionalidades que se utilizan en diversos archivos del proyecto con el fin de evitar la redundancia en código.

Middlewares, Se pueden encontrar validaciones de campos, validaciones de JWT, y validaciones de roles. En esta carpeta se encuentran validaciones debido a que se implementa tan pronto entra la solicitud al proyecto.

Public, Es donde se almacena el proyecto de angular ya generado para producción.

Routes, En esta carpeta estan definidos todos los endpoints establecidos, creados y funcionales.

Sockets, Encargada de la conexión entre el frontend y backend por medio de sockets.

Utils Se encuentra la conexión a la base de datos y la conexión con el mqtt.

app.js Archivo encargado de la inicialización

Testing, teniendo en cuenta la importancia del buen funcionamiento de cada uno de los endpoints generados, se llevo a cabo el proceso de testing de los mismos con ayuda de postman, en dicha herramienta se realizaron las pruebas necesarios para cada uno de los casos a presentas en los diferentes endpoints desarrollados, adicionalmente, se documento y publicó toda la información. A continuación, se muestra una tabla con los links correspondientes a cada colección creada en postman donde se encuentra la documentación completa de cada endpoint creado.

Tabla 22

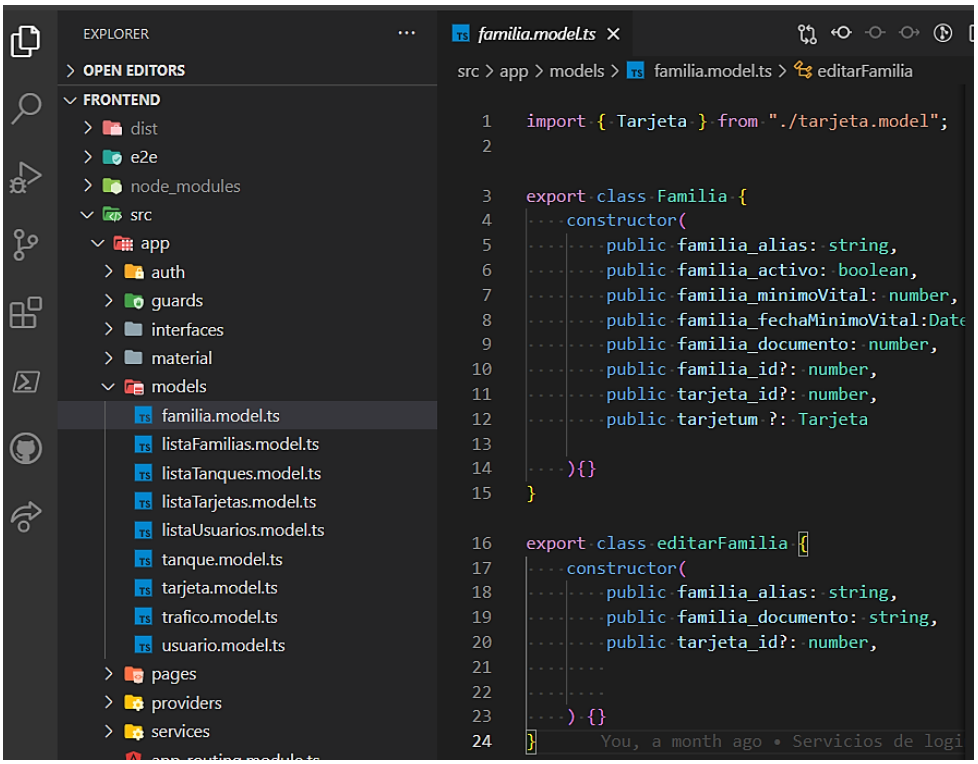
Documentación de Endpoints en Postman

Colección	**Url de documentación**
Autenticación	https://documenter.getpostman.com/view/14887192/TzRXARZJ
Usuarios	https://documenter.getpostman.com/view/14887192/TzRXARdb
Familias	https://documenter.getpostman.com/view/14652065/TzRa64HG
Tarjetas	https://documenter.getpostman.com/view/14887192/TzRXARdd
Tanques	https://documenter.getpostman.com/view/14887192/TzRXARdf
Dashboard	https://documenter.getpostman.com/view/14887192/TzkyLzRg

4.3.3.1 Modelos en el Proyecto de NodeJs, En el proyecto se empleo sequelize como ORM. En la carpeta *models* se realizó la definición de cada una de las entidades creadas y expuestas en el item 6.1.4 donde se llevo a cabo el diseño de la base de datos. En la siguiente imagen se logra visualizar la forma en que se modeló la tabla familias, de igual manera se establecieron cada uno de los modelos definidos en la base de datos.

Figura 12.

Modelo familia NodeJs



4.3.3.2 Endpoints Generados, dentro de los Endpoints realizados para la funcionalidad del proyecto, cabe resaltar la implementación de CRUD en la mayoría de entidades existentes. Se aplicó CRUD (Create, Read, Update, Delete) para familias, usuarios, tarjetas, tanques. Por otra parte, se desarrollaron endpoints para el despliegue de información en las graficas

respectivas del dashboard general del usuario con rol administrador, el dashboard general del usuario con rol cliente y dashboard de un tanque especifico, del cual solo tiene acceso el usuario con rol administrador. Adicionalmente y no menos importante, los endpoint implementados para el inicio de sesión, registro por parte de los usuarios con rol cliente y finalmente el de cerrar sesión.

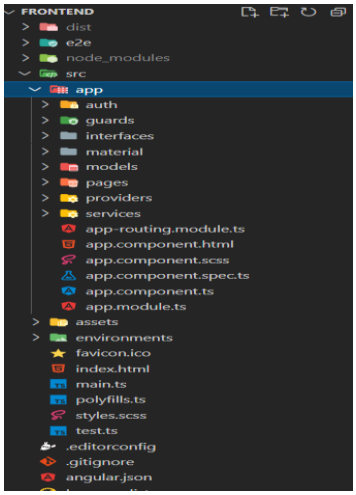
4.3.3.3 JWT (JSON Web Token), la implementación de JSON Web Token en el proyecto fue necesaria para el mejoramiento de seguridad en la página web, ya que con la generación de token de acceso se permitió propagar la identidad de quien ingresa al sistema de una mejor manera facilitando la implementación de los roles establecidos.

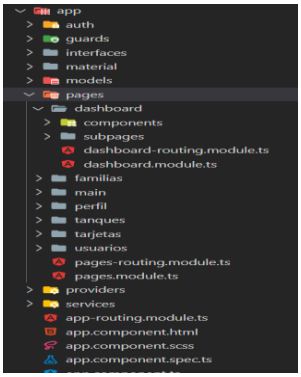
4.3.4 Creación del Proyecto Angular

El proyecto Frontend fue modularizado, buscando tener como resultado un proyecto funcional, eficaz y escalable. Asimismo, se aplicó Lazy Load en el enrutamiento del proyecto, procurando disminuir el tiempo de inicio al momento de carga en cada página y sus respectivos componentes. En cuanto a la maquetación, se logró obtener una página en la que destaca la usabilidad, implementando la librería de componentes web Angular Material.

Figura 13.

Proyecto frontend





4.3.4.1 Módulos Establecidos, los módulos establecidos en el proyecto fueron creados resaltando las entidades planteadas a la hora de diseñar el sistema de registro y control del consumo de agua potable, adicionando a estos el módulo empleado para la autenticación e ingreso al sistema y el módulo implementado para la visualización de informes por medio de gráficas estratégicas donde se dan a conocer los datos más relevantes del sistema en funcionamiento.

4.3.4.2 Guards, frente a la necesidad de controlar el acceso de ciertos usuarios a vistas o componentes determinados, se implementaron guards. Estos componentes permiten o bloquean el acceso bajo condiciones previamente establecidas. Los guards implementados en el proyecto son de tipo CanActivate, ya que principalmente se requiere el control de permisos o accesos a ciertos usuarios dependiendo del rol que cumplen en el sistema garantizando el inicio de sesión.

La página web no tiene información para presentar al público no registrado, es decir, un usuario que no ha sido previamente registrado en la base de datos únicamente podrá observar la página de inicio de sesión y registro.

Los administradores tienen visibilidad de un dashboard general en el que se logra observar un informe, acompañado de gráficas que facilitan su entendimiento, esta información presentada se basa en la actividad reciente de los tanques en conjunto que hacen parte del sistema de suministro de agua potable. Adicionalmente, tienen a su disposición información detallada de cada uno de los tanques de forma individual. Los administradores pueden realizar el respectivo CRUD de todas las entidades pertenecientes al sistema, es decir, usuarios, familias, tarjetas y tanques.

Por otra parte, los usuarios registrados con rol subordinados no tienen acceso a los dashboard anteriormente nombrados, sin embargo, se basan principalmente en controlar los registros de las familias beneficiadas por este sistema, teniendo acceso al CRUD con excepción de

eliminar, ya que en el momento en el que un beneficiario procede a eliminar ya sea una familia, usuario de rol cliente o tarjeta se lleva a cabo la desactivación del mismo.

En cuanto al usuario con rol cliente, al iniciar sesión en la plataforma tiene acceso a un dashboard personal, donde encuentra información acerca del tanque al que pertenece, la cantidad de agua que tiene a su disposición y las últimas transacciones realizadas por su familia.

4.3.4.3 Servicios, los servicios creados con el fin de consumir los Endpoints definidos en el proyecto backend, se ponen en funcionamiento una vez sean llamados en el archivo .ts de cada vista o componente implementado. En la carpeta de servicios se logra ubicar un archivo enfocado en las acciones de cada una de las entidades creadas (usuarios, familias, tarjetas, tanques) adicionando el servicio de autenticación y dashboard.

4.3.5 Creación de la Base de Datos

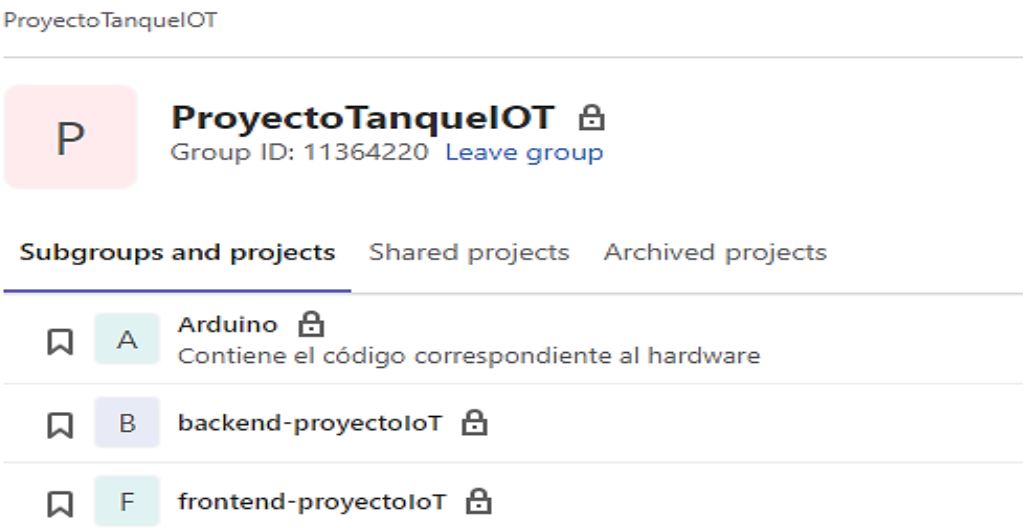
Basados en la estructura planteada previamente en el inciso 4.1.4 donde se puso en marcha el diseño de la base de datos según las exigencias y necesidades identificadas para la elaboración del proyecto, se implementó el modelo allí definido. El administrados o gestor de bases de datos que se utilizó fue HeidiSQL, este administrador tiene la capacidad de conectarse con bases de datos MySQL, MSSQL, PostgreSQL y demás bases de datos relacionales. Para conectarnos con la base de datos creada en la nube, únicamente se dispuso a completar el formulario de inicio presentado en el software de administración en el que se solicitaba las credenciales de la base a conectar. Una vez dentro, se procedió a la creación de ciertas tablas faltantes, ya que anteriormente se realizó el mapeo con sequelize en el repositorio de backend con NodeJS.

4.3.6 Creación del Repositorio en el Sistema de Control de Versiones GitLab

Como consecuencia del trabajo en equipo practicado al dar cumplimiento al presente proyecto, surge la necesidad de implementar un sistema de control de versiones con el fin de facilitar la labor de equipo, otorgando eficiencia, compatibilidad y confianza a la hora de trabajar simultáneamente en dicho proyecto. Se implementó un grupo en GitLab llamado "ProyectoTanqueIOT", en su interior se encuentran tres repositorios, uno correspondiente al backend del proyecto, un segundo repositorio enfocado al frontend y finalmente un último repositorio con los archivos tanto del maestro como el esclavo del hardware.

Figura 14.

Grupo implementado en GitLab

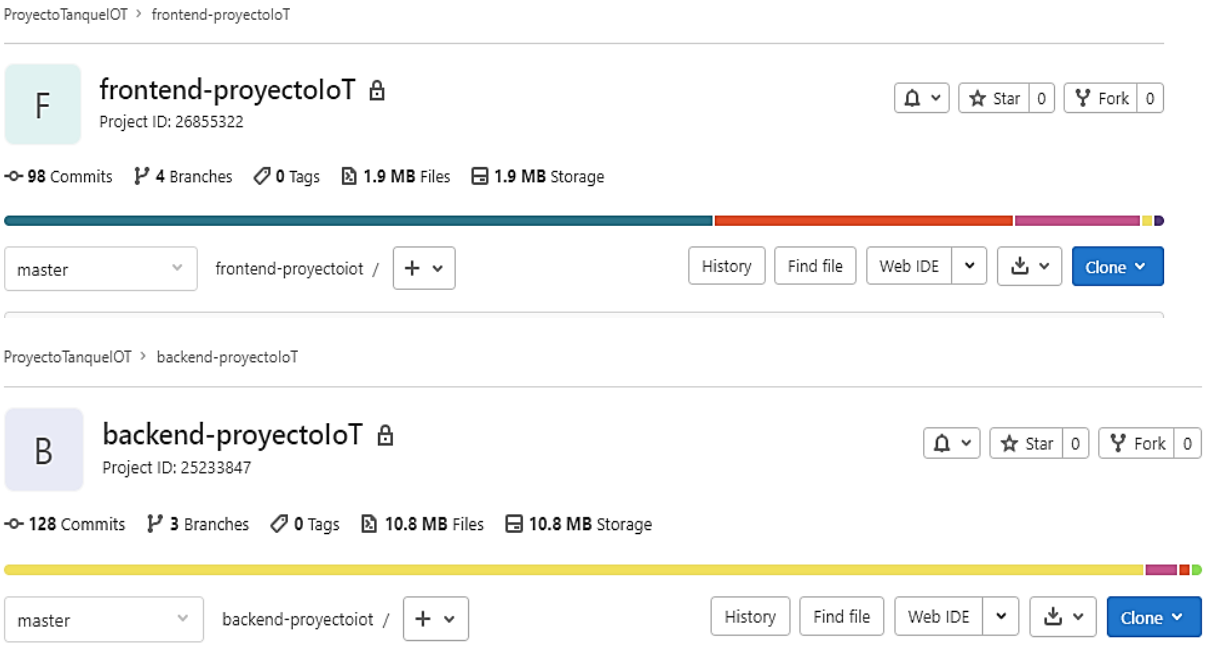


A medida en que se avanzó en el proyecto, se logró identificar diversas versiones del mismo. En total se implementaron 128 commits y 3 ramas en el repositorio del backend. Así

mismo, en el repositorio enfocado en el frontend se obtuvieron en total 98 commits y 4 ramas definidas dependientes a la demanda del proyecto.

Figura 15.

Commits y ramas implementadas en el proyecto



4.3.7 Conexión Wifi del ESP8266

La conexión Wifi del ESP8266 es el primer paso que se implementó durante la elaboración del montaje de los dispositivos, la conexión se hace buscando el acceso del sistema a una red establecida donde se mantiene una comunicación continua entre dispositivos y cloud del proyecto (backend y base de datos en la nube) con el fin de hacer cumplir el objetivo propuesto.

Figura 16.

Conexión de ESP8266 a red WiFi

```
/*
   ========================================================================
   ========================================================================
            CONEXIÓN WIFI
   ========================================================================
   ========================================================================
*/
void setup_wifi();
void callback(char* topic, byte* payload, unsigned int length);
   ========================================================================
   ========================================================================
                            CONEXIÓN WIFI
   ========================================================================
   ========================================================================
*/
void setup_wifi() {
  delay(10);
  // Nos conectamos a nuestra red Wifi
  Serial.println();
  Serial.println("Conectando a ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("Conectado a red WiFi!");
  Serial.println("Dirección IP: ");
  Serial.println(WiFi.localIP());
}
```

4.3.8 Conexión MQTT y EMQX

Para la implementación de EMQX se abrieron los puertos necesarios como son 1883(protocolo MQTT) , 8883(protocolo MQTT/SSL), 8093(protocolo MQTT/ websocket), 8094(protocolo protegido MQTT/ websocket), 8090(HTTP API), 18083(Interface Dashboard) en el cortafuego de VESTA, estos puertos se agregan adicionalmente en el grupo de seguridad de AWS. Seguido a esto, se descarga EMQX por medio de la terminal, la versión obtenida para la implementación del proyecto fue la V3.0.1. Se descomprimió el archivo obtenido y con el

comando *emqx start* se da inicio al protocolo, para confirmar que la instalación es correcta nos dirigimos al dashboard generado, el cual está ubicado en el puerto 18083 de nuestro dominio

Se encontró la necesidad de cambiar ciertos puertos que vienen configurados por defecto debido a que ya estaban empleados por las instalaciones anteriormente realizadas. El puerto 8083 se actualiza por el 8093 y se adiciona el puerto 8094 estableciéndose como listener.ws.external. Las configuraciones anteriormente expuestas se llevan a cabo en el archivo emqx.conf.

En cuanto a la mejora de seguridad, se llevó un control de acceso para quienes pueden enviar tópicos a nuestro servidor, empleando ACL con MySql para cumplir este objetivo. Principalmente se crearon dos tablas en la base de datos para llevar a cabo esta configuración de privacidad, creando un registro en la tabla mqtt_user, los usuarios registrados en esta tabla son quienes tendrán acceso a la suscripción y publicación de mensajes.

Figura 17.

ACL MySql EMQX

Authentication / Superuser Table

```
CREATE TABLE `mqtt_user` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(100) DEFAULT NULL,
  `password` varchar(100) DEFAULT NULL,
  `salt` varchar(35) DEFAULT NULL,
  `is_superuser` (1) DEFAULT 0,
  `created` datetime DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `mqtt_username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Sample data:

```
-- Client information
INSERT INTO `mqtt_user` ( `username`, `password`, `salt`, `is_superuser`)
VALUES
    ('emqx', 'efa1f375d76194fa51a3556a97e641e61685f914d446979da50a551a4333ffd7', NULL, 0);
```

ACL rule table

```
CREATE TABLE `mqtt_acl` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `allow` int(1) DEFAULT 1 COMMENT '0: deny, 1: allow',
  `ipaddr` varchar(60) DEFAULT NULL COMMENT 'IpAddress',
  `username` varchar(100) DEFAULT NULL COMMENT 'Username',
  `clientid` varchar(100) DEFAULT NULL COMMENT 'ClientId',
  `access` int(2) NOT NULL COMMENT '1: subscribe, 2: publish, 3: pubsub',
  `topic` varchar(100) NOT NULL DEFAULT '' COMMENT 'Topic Filter',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Para culminar la configuración necesaria del ACL con MySql, se dirigió al dashboard del dominio(puerto 18083) y se realizó la definición de datos requeridos para la conexión con ACL MySql.

La conexión MQTT desde el arduino y desde el proyecto backend con NodeJS, tiene gran importancia ya que la correcta funcionalidad del proyecto depende de esta.

En primer lugar, se implementó la conexión entre el arduino y el servidor MQTT, la configuración que se puso en práctica consta principalmente de la asignación de valores a las variables necesarias para conectarse, la creación de una función que tenga como tarea principal el

detectar si la conexión es estable, en caso de no serlo la función debe volver a efectuar la conexión. Finalmente, en el setup se hace la inicialización del proceso.

Figura 18.

Conexión mqtt en ESP 8266

```
/*
  ============================================================================
  ============================================================================
          ESTABLECE VALORES DE CONEXIÓN MQTT Y
          VALOR SERIAL DEL DISPOSITIVO
  ============================================================================
  ============================================================================
*/
const String serial_number = "123456789";
const char* ssid = "VILLAMIZAR";
const char* password = "[redacted]";
const char *mqtt_server = "proyectoiot.gq";
const int mqtt_port = 1883;
const char *mqtt_user = "[redacted]";
const char *mqtt_pass = "[redacted]";
```

```
/*
  ====================================
  ====================================
          CLIENTE CONECTADO MQTT
  ====================================
  ====================================
*/

if (!client.connected()) {
  reconnect();
}
client.loop();
```

```
  ============================================================
        FUNCIÓN ENCARGADA DE HACER RECONEXIÓN MQTT EN CASO
                    DE QUE SE CAIGA
  ============================================================
  ============================================================
*/
void reconnect() {
  while (!client.connected()) {
    Serial.print("Intentando conexión Mqtt...");
    // Creamos un cliente ID
    String clientId = "esp8266";
    clientId += String(random(0xffff), HEX);
    // Intentamos conectar
    if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {
      Serial.println("Conectado!");
      char topic[40];
      String topic_aux = serial_number + "/command";
      topic_aux.toCharArray(topic, 40);
      client.subscribe(topic);

    } else {
      Serial.print("falló :( con error -> ");
      Serial.print(client.state());
      Serial.println(" Intentamos de nuevo en 5 segundos");
      delay(5000);
    }
  }
}
```

```
void setup() {
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
```

En el proyecto de Node, se ejecuta la conexión por primera vez en un documento independiente dentro de la carpeta utils. Adicionalmente, existe un archivo controller donde se encuentra la mayoria de funcionalidades implementadas.

Figura 19.

Conexión MQTT en NodeJs

```
var mqtt = require('mqtt');
require('dotenv').config();
const mqttController = require('../controllers/mqttController');
const TanqueModel = require('../models/tanqueModel');

//CREDENCIALES MQTTP
var options = {
    port: process.env.mqttPort,
    host: process.env.mqttHost,
    clientId: 'acces_control_server_' + Math.round(Math.random() * (0- 10000) * -1) ,
    username: process.env.mqttUsername,
    password: process.env.mqttPassword,
    keepalive: 60,
    reconnectPeriod: 1000,
    protocolId: 'MQIsdp',
    protocolVersion: 3,
    clean: true,
    encoding: 'utf8'
  };

  var client = mqtt.connect(process.env.mqtthostT, options);

//SE REALIZA LA CONEXION
client.on('connect', function () {
  console.log("Conexión  MQTT Exitosa!");
  // client.subscribe('123456789/access_query', function (err) {
  client.subscribe('+/access_query', function (err) {
  });
  client.subscribe('+/datosagua', function (err) {
  });
  client.subscribe('+/nivelTanque', function (err) {
  });
  client.subscribe('+/reporte', function (err) {
  });
})
```

4.3.9 Control de Acceso con RFID en ESP8266

Dentro de las necesidades destacadas en este proyecto se encuentra el control y monitoreo del suministro de agua potable, dando cabida a priorizar la implementación del control de acceso, por esta razón, es uno de los primeros montajes que se llevaron a cabo en el proyecto. En la siguiente imagen se puede observar la conexión que se implementó entre el ESP8266-RC522 y el lector RFID. De igual manera, se puede contemplar los puertos utilizados.

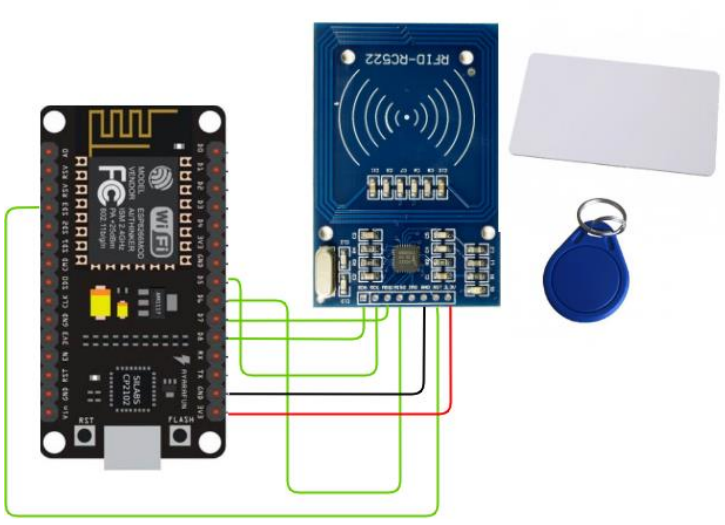
Tabla 23

Puertos del control de acceso RFID

RFID	ESP8266
SDA	D8
SCK	D5
MOSI	D7
MISO	D6
GND	GND
RST	S3
3.3V	3V

Figura 20.

Control de acceso con RFID en ESP8266



4.3.10 Suministro de Agua en ESP8266

En el desarrollo del sistema de selección que realiza el usuario en la transacción de retiro de agua se implementó una lógica la cual está compuesta por tres botones, lo cuales son de confirmación, aumento de cantidad y disminución del valor a retirar, placa Arduino ESP8266, control de acceso RFID y pantalla LCD.

El valor de agua a sacar se convirtió en segundos ya que esta lógica no contaba con los sistemas de suministros de agua, por lo tanto, los litros que el usuario elija sacar se manejaron como segundos y se visualizaron mediante un led, el cual duraba encendido los segundos que se habían seleccionados en la transacción

En el funcionamiento de esta lógica el usuario inicia la autenticación por medio de su tarjeta asignada, se envía a NodeJs el tópico 1 *tanque_serie/acces_query* de la tabla de tópicos del inciso 4.3.2, con el objetivo de verificas si la tarjeta se encuentra registrada, activa y el tanque está disponible, si esta verificación es correcta se procede a consultar la información necesaria para continuar con la transacción. Se estructura el tópico 2 *tanque_serie/command* con la información pertinente y es enviado por NodeJs a la placa ESP8266.Despues de recibir el tópico, el usuario visualiza en la LCD el alias de su familia en caso de tener acceso y que el tanque se encuentre disponible, además, se le presenta el mínimo vital que tiene disponible por lo tanto el usuario procede a seleccionar un valor de agua a retirar mayor a 0 y menor o igual a su suministro disponible. Después de establecer la cantidad, el LED prende durante el tiempo estipulado simulando los litros de agua que fueron elegidos, por último se finaliza la transacción y la ESP8266 envía el tópico 3 *tanque_serie/datosagua* el cual contiene el mínimo vital de la familia actualizado, el tráfico que se realizó ,la cantidad de agua que se suministró, el nivel de agua actualizado del tanque y los ID tanto del usuario que realizo la transacción como de la familia a la cual se encuentra asignado, esto con el objetivo de almacenar en la base de datos del sistema la información de la transacción realizada.

4.3.11 Implementación de Maestro-Esclavo

La configuración de las dos placas Arduino que fueron seleccionadas para el prototipo, se realizó implementando un protocolo de comunicación I2C.El cual funciona con una arquitectura de maestro-esclavos; por un lado, tenemos el maestro el cual es el que coordina la comunicación, también se tienen esclavos los cuales son esos dispositivos que están a la espera de que un maestro establezca comunicación con ellos.

para implementar esta comunicación se necesitan tres pines los cuales son SDA, SCL y GND. La implementación de este protocolo de comunicación se utiliza en su mayoría cuando no hay suficientes de entradas y no basta con el uso de una sola placa Arduino.

Este protocolo se implementó en el proyecto estableciendo al Arduino ESP8266 como maestro y al Arduino MEGA 2560 como esclavo, se realizó el montaje respectivo teniendo en cuenta los pines establecidos por cada placa para I2C, prosiguiendo con la ejecución de pruebas las cuales consistían en enviar mensajes en el sentido maestro-esclavo y esclavo maestro. Esto debido a que según nuestros requerimientos las dos placas deben tener la capacidad de enviar y recibir información de manera correcta.

Con la implementación de la comunicación maestro esclavo entre las dos placas de Arduino cambiaron las conexiones iniciales que se tenían tanto de la RFID como de la lógica de selección de agua y retiro de suministro.

La RFID mantuvo su conexión en la placa ESP8266. Por otra parte, la lógica del suministro de agua se trasladó a la placa MEGA 2560 ya que en esta placa se tienen más entradas y una mayor versatilidad al momento de realizar las conexiones de los botones de selección y la pantalla LCD

4.3.12 Planteamiento de Estructura de Hardware

Una vez se tuvo conocimiento acerca de cada uno de los dispositivos que participan en el circuito objeto del proyecto, y se complementó esta información con la conexión y comunicación I2C entre el Arduino Mega como esclavo y el ESP8266 como maestro, se procedió a definir la estructura final. Con este fin se estudiaron las entradas y salidas, analógicas y digitales disponibles tanto en el Arduino MEGA como en la ESP8266.

La distribución de dispositivos entre estas dos placas quedó de la siguiente manera:

En ESP8266 se implementó:

- el control de acceso (Lector RFID).

A su vez, en el Arduino MEGA se establecieron:

- Selección del dato principal (pulsadores y pantalla LCD)
- Medición de flujo de agua (Sensor de flujo de agua)
- La detección del flujo de agua (Sensor de flujo de agua)
- El paso del flujo de agua (Bomba de agua y relé)
- Llenado de tanque de almacenamiento de agua (electroválvula y relé)

4.3.13 Sistema de Salida de Agua

En el sistema de suministro de agua del inciso 4.3.13 simulamos que los litros seleccionados por el usuario se mostraran como el tiempo que permanecía un LED encendido, ya en esta parte, se incluyó la salida de agua implementando un sensor de flujo, el cual mide la cantidad de agua que sale del tanque por medio de la bomba. También se incorporó el control de este sistema mediante un relé, esto controla que el sistema está activo cuando el usuario selecciona el valor de agua a sacar y se desactiva al momento en el que sensor haya medido la cantidad de agua selecciona.

Este sistema está conformado por componentes y sensores los cuales se tuvieron que implementar de la siguiente manera:

- Sensor de flujo de agua: este sensor tuvo que ser calibrado, debido a que esto otorga una mayor precisión en sus medidas. Para este proceso se tuvo que calcular una constante K de conversión, esta constante transforma la medida que por defecto este sensor tiene, la cual es frecuencia de pulsos a medidas de flujo de agua.

  Esta constante varía dependiendo del modelo de sensor que se utilice y se calcula con la siguiente fórmula

$$K = \frac{n°Pulsos}{Volumen \,.\, 60}$$

Donde K es la constante de conversión, n°Pulsos es la cantidad de pulsos que midió el sensor, de acuerdo con el volumen de líquido suministrado. Para el sensor que se utilizó, este cálculo nos arrojó como resultado que nuestro factor de conversión K tiene un valor de 108.69. Teniendo este factor calculado, se realizaron pruebas en las cuales se le enviaba cierta cantidad de agua y comprobamos si efectivamente el sensor estaba midiendo de manera correcta la cantidad de líquido suministrado.

- Bomba y relé: estos dos componentes son los que bombean y controlan el sistema. El relé tiene como tarea desactivar la bomba cuando el sensor de flujo haya medido la cantidad de agua que selecciono el usuario, de lo contrario la mantendrá activa hasta que se otorgue el agua solicitada.

Este sistema de salida de agua se activa después de que el usuario seleccione la cantidad de agua que desea retirar y como se dijo anteriormente se desactiva cuando el suministro haya sido otorgado. Con este montaje se elimina del suministro de agua el uso del LED y se adapta para que

efectivamente se entregue el agua que el usuario quiera retirar al momento de realizar la transacción.

4.3.14 Medición de Nivel de Agua en los Tanques de Almacenamiento

Para el desarrollo de esta lógica se establecieron los 3 niveles que podría tener el tanque los cuales son Bajo, Medio y Alto, estos niveles se calculan mediante los niveles superiores e inferiores calculados.

La definición de estos niveles se realizó estableciendo como límite superior el 30% de la capacidad del tanque y el límite inferior es el 95% de la capacidad del tanque, la cual se obtiene al momento de que NodeJs retorna el tópico 2 del inciso 4.3.2, ya que este contiene la información acerca de la capacidad que contiene este tanque, sin embargo, a esta capacidad se le resto la altura de la bomba ya que esta debe estar sumergida todo el tiempo en agua para su correcto funcionamiento. Teniendo los límites establecidos se configuran los niveles de la siguiente manera:

- Nivel Bajo: el tanque se encuentra en nivel bajo si la distancia medida por el sensor ultrasónico es mayor al límite inferior.
- Nivel Alto: si el límite superior es mayor que la distancia medida por el sensor, entonces el tanque tiene un nivel Alto.
- Nivel Medio: este nivel se establece siempre y cuando la distancia medida sea mayor o igual al límite superior y menor o igual al límite inferior.

La medición constante del nivel de agua del tanque es sumamente importante porque determina si es necesario llenar el tanque y si el usuario efectivamente puede retirar agua, además, si el tanque notifica un nivel Bajo al momento de que el usuario está realizando la transacción, se procede cancelar la transacción y enviar el tópico 4 del inciso 4.3.2 a NodeJs el cual recibe el nivel de agua y si este nivel efectivamente es Bajo desactiva el tanque hasta que sea llenado.

Para comprobar la efectividad de este sistema de medición se realizaron diversas pruebas en las cuales se medía la distancia y se constataba que el tanque efectivamente se encontraba en el nivel de agua que daba como resultado el programa.

Esta lógica de medición de los niveles se implementó en la placa de Arduino MEGA 2560 uniéndose con las lógicas del suministro y salida de agua desarrollados anteriormente.

4.3.15 Llenado de Tanque de Almacenamiento

El llenado de los tanques de almacenamiento se desarrolló mediante la implementación de una electroválvula y un relé. La activación de esta función de llenado se da en el momento en que el usuario autenticado previamente selecciona y retira el agua, en ese momento se llama a la función que mide el nivel de agua del tanque, si este nivel es Bajo, efectivamente se procede a que la placa ESP8266 envié el tópico 4 *tanque_serie/nivelTanque* del inciso 4.3.2 a NodeJs el cual desactiva el tanque, actualizando el nivel y el estado en la base de datos. Por otra parte, se procede a llamar la función de llenado la cual mediante el relé activa la electroválvula permitiendo la entrada de agua al tanque, durante este proceso se está midiendo constantemente el nivel del agua y en el momento en que la medición dé como resultado que el tanque se encuentra en nivel Alto, se desactiva la electroválvula, el ESP8266 envía el tópico 4 *tanque_serie/nivelTanque* del inciso 4.3.2 con el objetivo de que NodeJs active el tanque y actualice la información del estado incluyendo el nivel del tanque en la base de datos. Cuando el tanque se encuentre ya con nivel Alto se termina el llamado de esta función.

El proceso que se realiza al momento de llamar la función de llenado se puede visualizar mediante un diagrama de actividades el cual se encuentra en el inciso 4.1.5 específicamente en la figura 4.

4.3.16 Reporte de Fallas en el Sistema

El reporte de fallas en la conexión entre el hardware y el servidor se implemento Utilizando en el programa de Arduino del maestro (ESP8266) una función la cual envía cada 10 minutos el tópico 5 *tanque_serie/reporte* del inciso 4.3.2 a NodeJs, el cual recibe el tópico y genera la fecha y hora exacta en la que recibió el tópico a partir de este momento pueden suceder dos cosas las cuales son:

- Reporte Falla: este reporte se establece mediante la diferencia entre la última fecha de actualización que tiene el tanque en la base de datos y la fecha que genero NodeJs al momento de recibir el, si esta diferencia es mayor a 10 minutos entonces se realiza el reporte de la falla, almacenando la información como la fecha y hora en la que se reportó el fallo en la tabla de Fallas y modificando la fecha y hora de la última actualización del tanque por la fecha y hora en la que se recibió el ultimo tópico.
- Reporte de Actualización: este reporte se realiza si la diferencia entre la fecha de la última actualización del tanque y la fecha que genero NodeJs al momento de recibir el tópico es menor a 10 minutos, entonces solo se realiza la modificación en la base de datos de la fecha de la última actualización del tanque por la nueva fecha generada al reportar tópico.

5. Prototipo Final

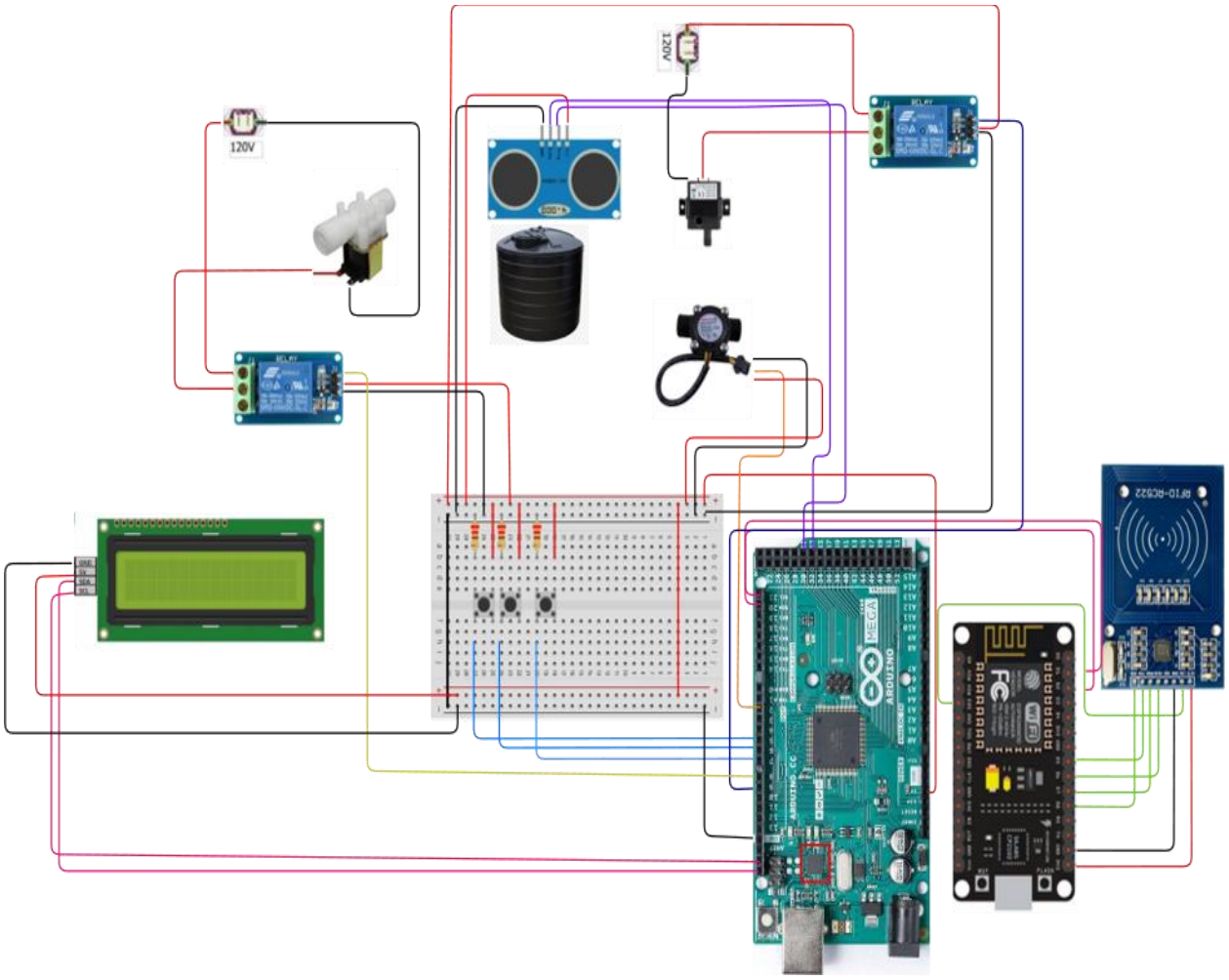
5.1 Esquema de Montaje del Hardware

En este esquema se puede ver y analizar los distintos componentes y dispositivos electrónicos que jugaron un papel importante en la realización de este sistema de suministro. Las conexiones que se realizaron se pueden apreciar en su totalidad incluyendo la interacción entre las distintas partes. Como se puede observar, contamos con los distintos dispositivos y sensores que se han expuesto con anterioridad como lo son la RFID, LCD, Arduino, sensores de caudal y proximidad, entre otros.

Este esquema es muy importante ya que ayuda a apreciar una vista general y completa de todo lo que se tuvo en cuenta para la realización de esta plataforma y más específicamente su hardware.

Figura 21.

Esquema de conexiones del hardware



5.2 Maqueta a escala del sistema

Como fin principal del proyecto se contempló el mostrar el funcionamiento del sistema realizado por medio de una maqueta a escala del sistema. A continuación, se puede observar el resultado de la maqueta implementada.

Figura 22.

Circuito de maqueta

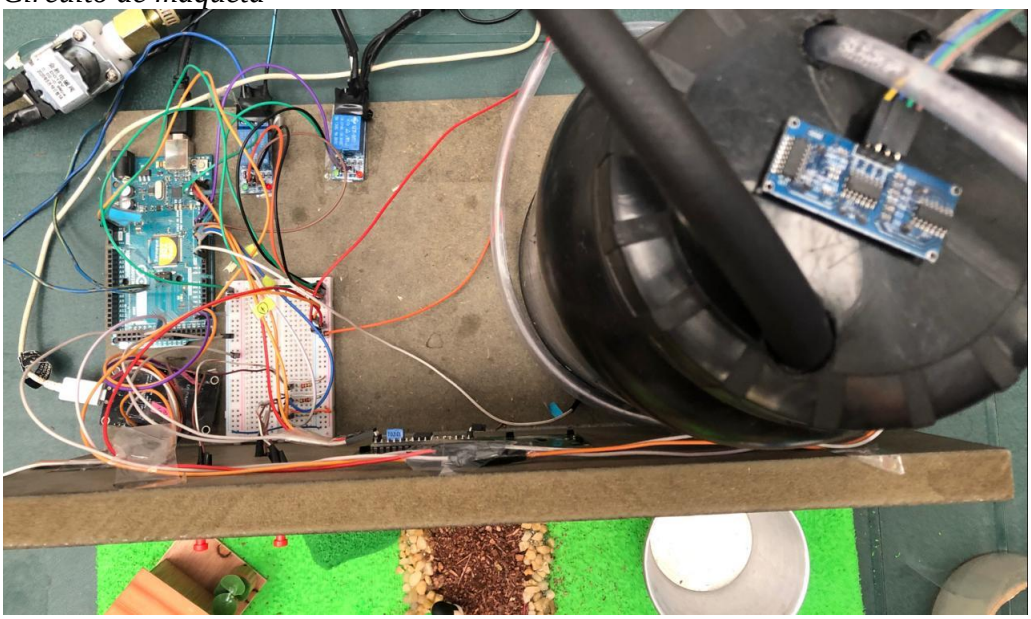


Figura 23.

Maqueta finalizada

5.3 Vista de usuario de la aplicación web

Parte importante del proyecto es la gestión y monitoreo del comportamiento de cada tanque registrado en el sistema de abastecimiento de agua.

En seguida se muestra la página para iniciar sesión y registro por parte de los usuarios de tipo cliente.

Figura 24.

Vista inicio de sesión

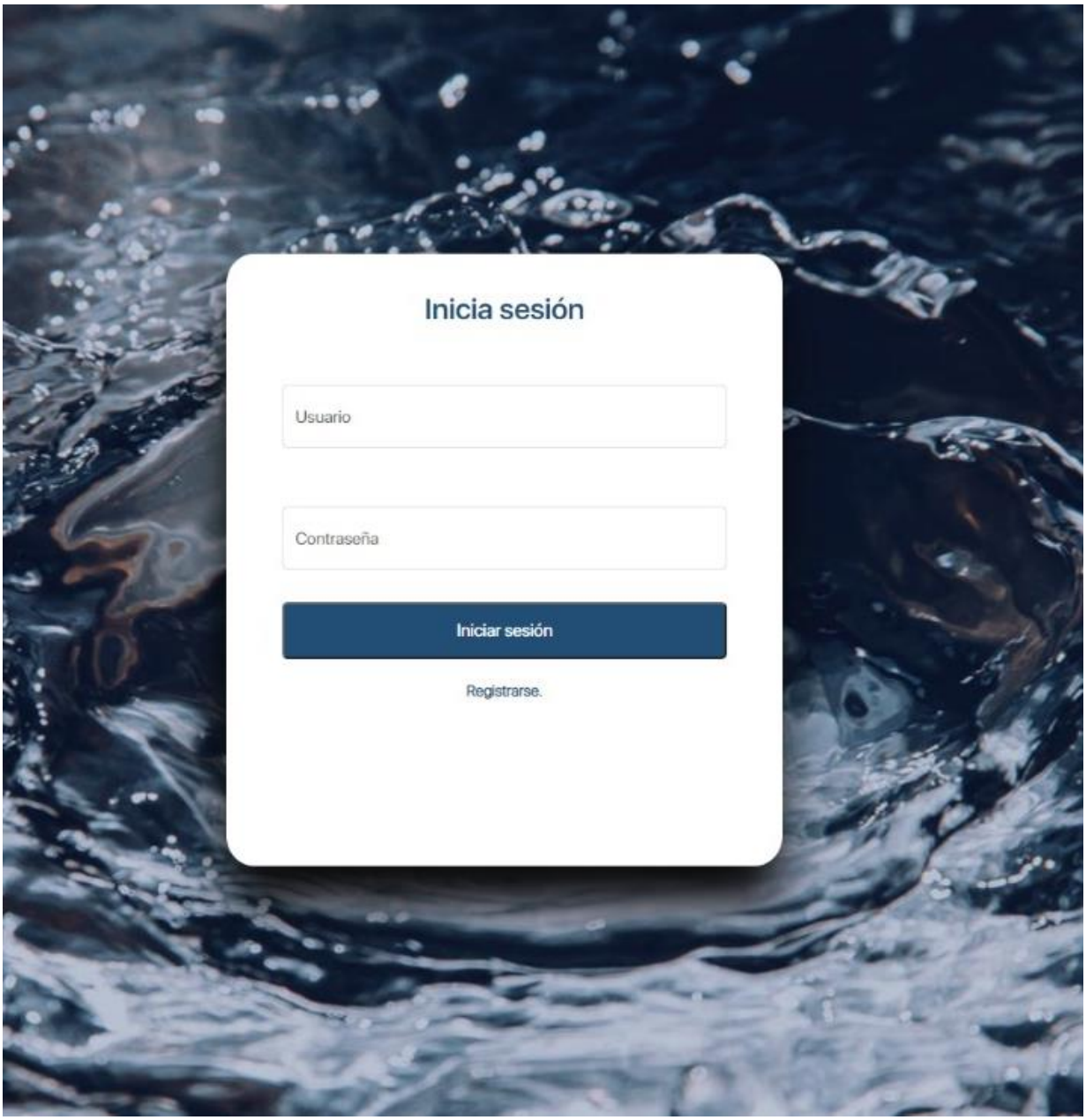
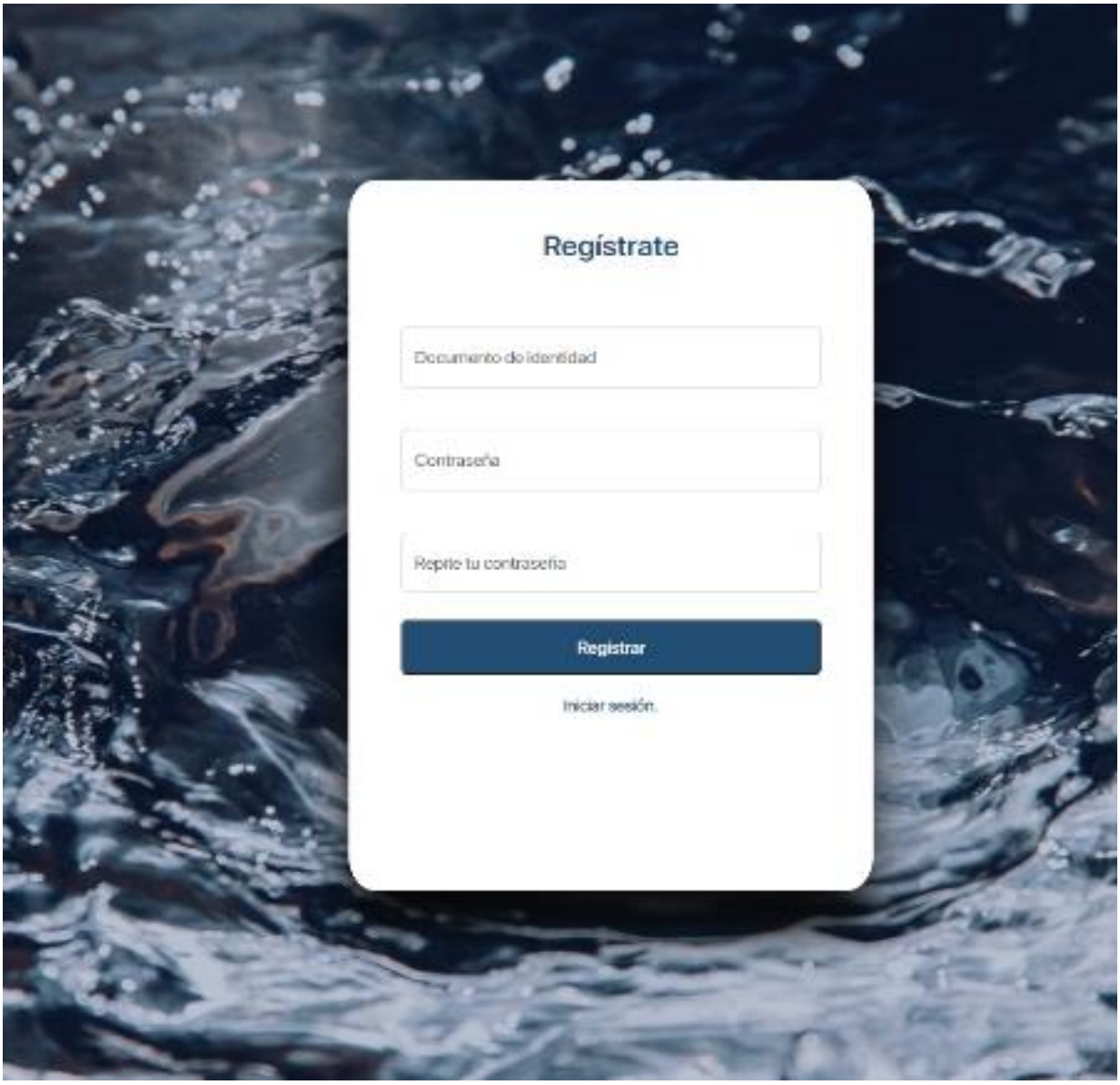

Inicia sesión
Usuario
Contraseña
Iniciar sesión
Registrarse.

Figura 25.

Vista registro



A continuación, se pueden observar las vistas resultantes del usuario con rol administrador para el dashboard de los tanques registrados en conjunto y el dashboard de un tanque específico.

Figura 26.

Dashboard de tanque especifico

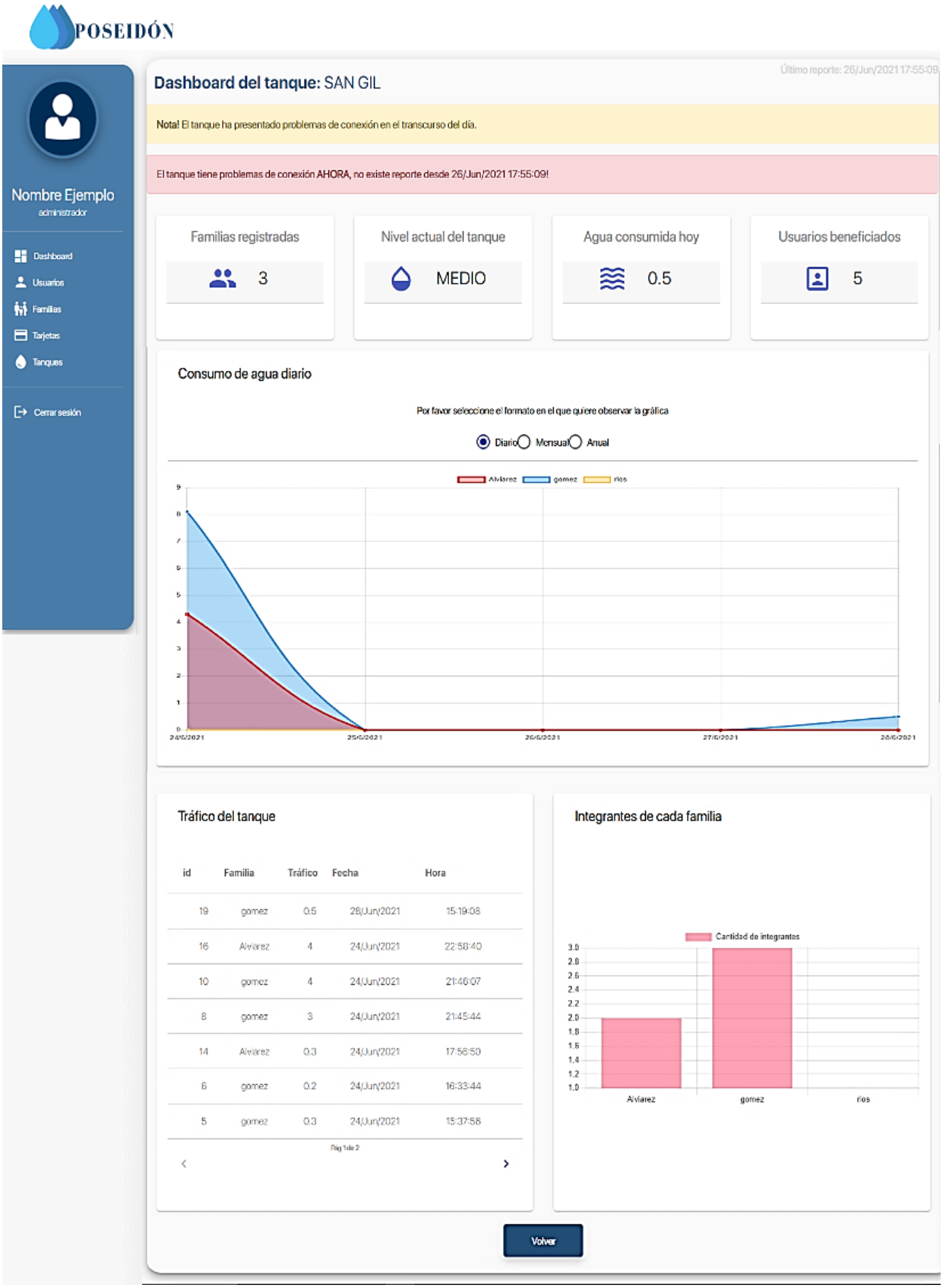
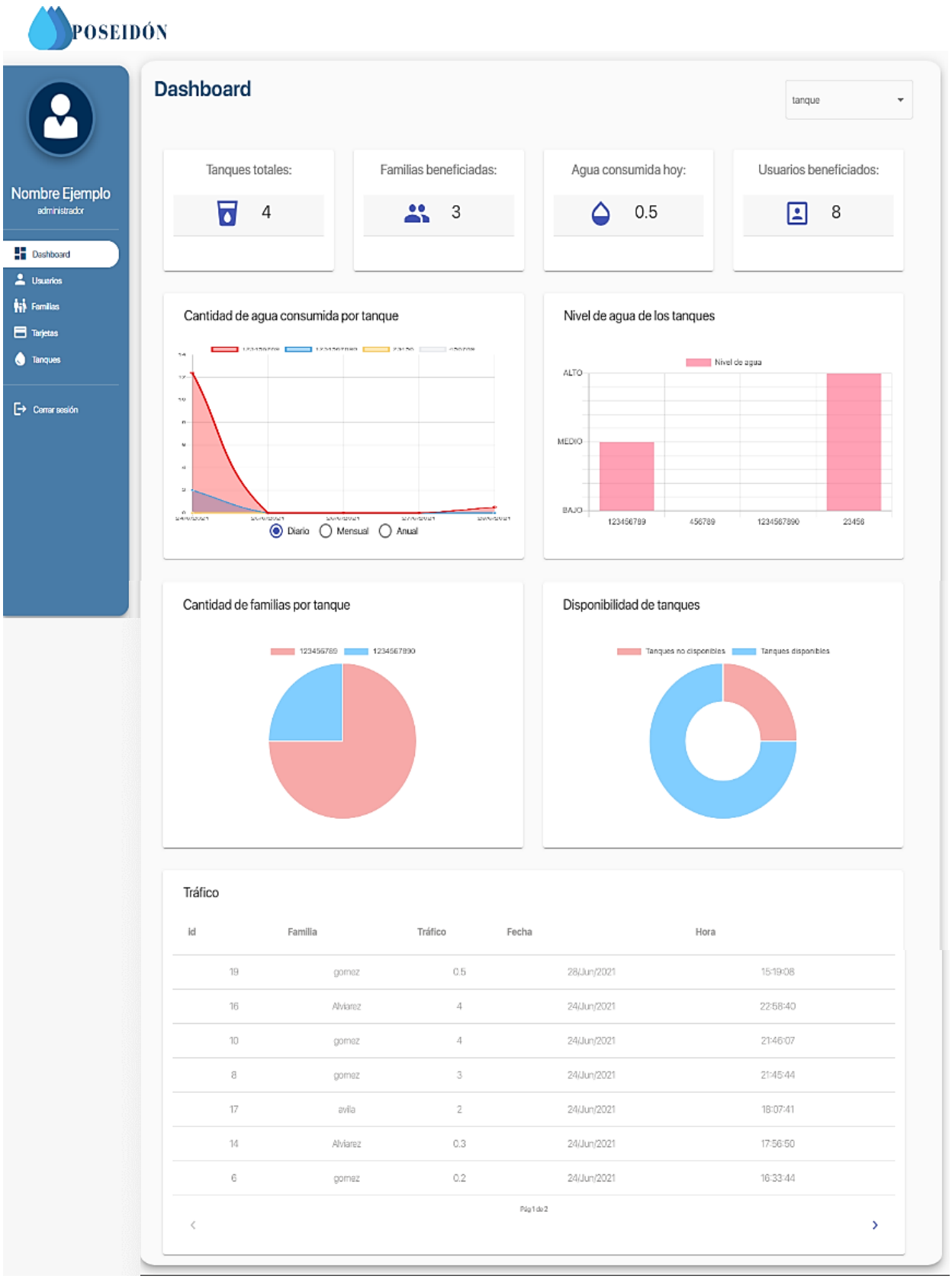


Figura 27.

Dashboard principal



Adicionalmente, el administrador puede observar la información de usuarios, familias, tarjetas y tanques registrados en la base de datos, agregar, editar y eliminarlos. Los registros desactivados (eliminados por los subordinados) los puede ver resaltados, y decidir si se eliminan definitivamente de la base de datos, o si se reactivan.

Figura 28.

Lista de usuarios

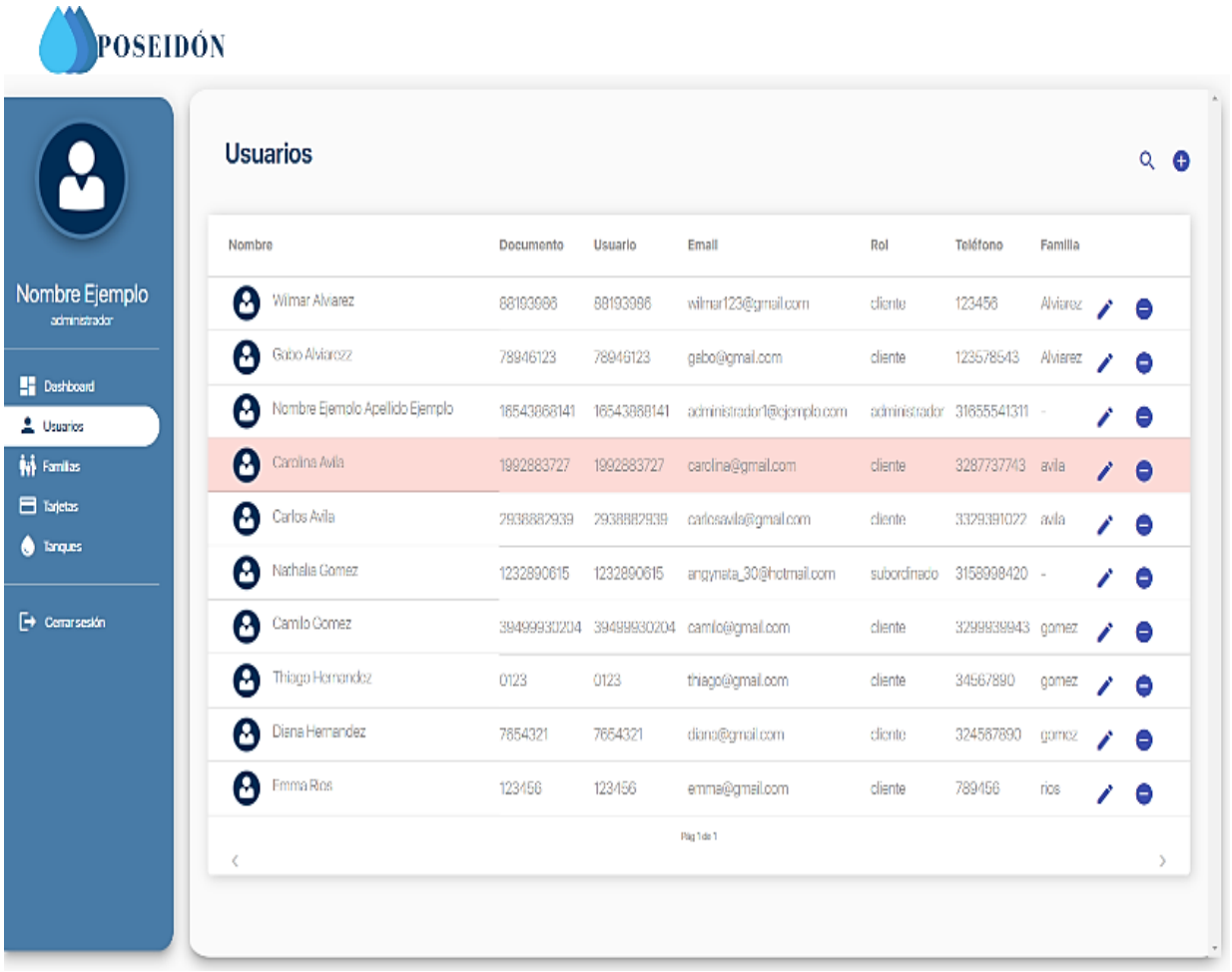
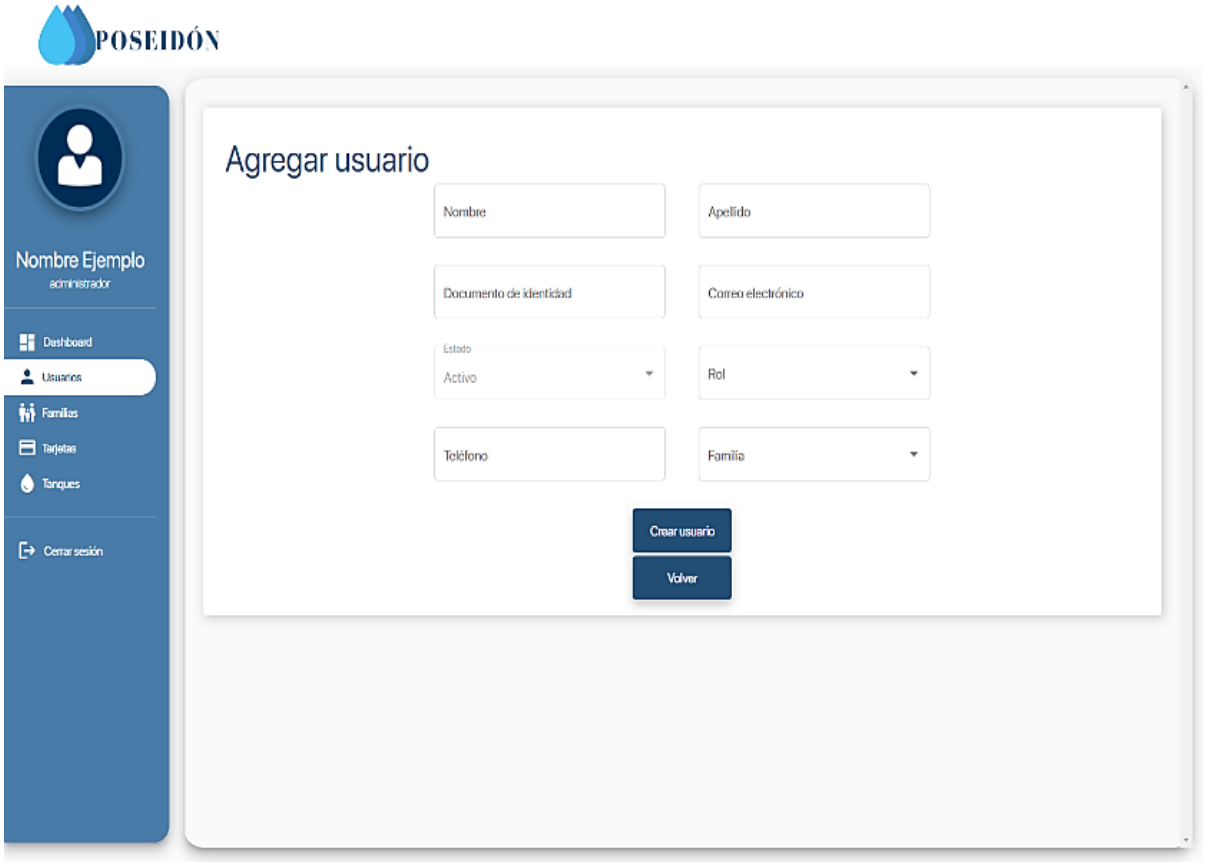


Figura 29.

Agregar Usuario



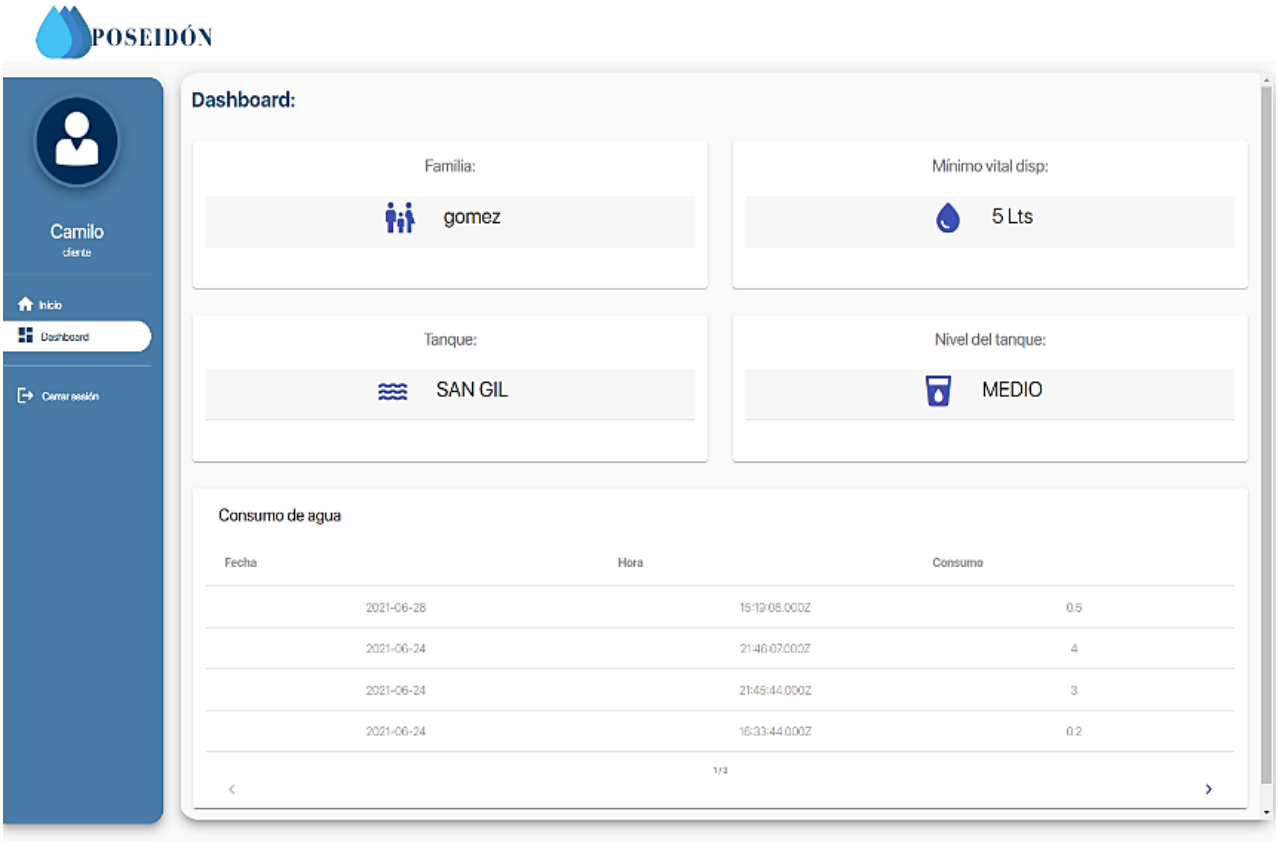
El usuario con rol cliente, es decir, el beneficiario del suministro de agua puede acceder a la página para ver en tiempo real el nivel de agua del tanque al que está registrado, puede observar en la plataforma información adicional como son las últimas transacciones realizadas por el u otros integrantes de su familia, el mínimo vital que tiene a su disposición y el alias de su familia.

Figura 30.

Inicio Usuario

Figura 31.

Dashboard Usuario



Finalmente, el usuario con rol subordinado no tiene a su disposición un dashboard que le ofrezca informes acerca de los tanques ya que su función es la gestión de los beneficiarios del sistema, es decir, el subordinado puede editar, crear, o desactivar usuarios con rol cliente, familias. Las tarjetas puede observarlas, asignarlas al crear o editar una familia y marcarlas como extraviadas. En cuanto a los tanques, únicamente puede verlos listados.

Figura 32.

Vista listar tarjetas para subordinados

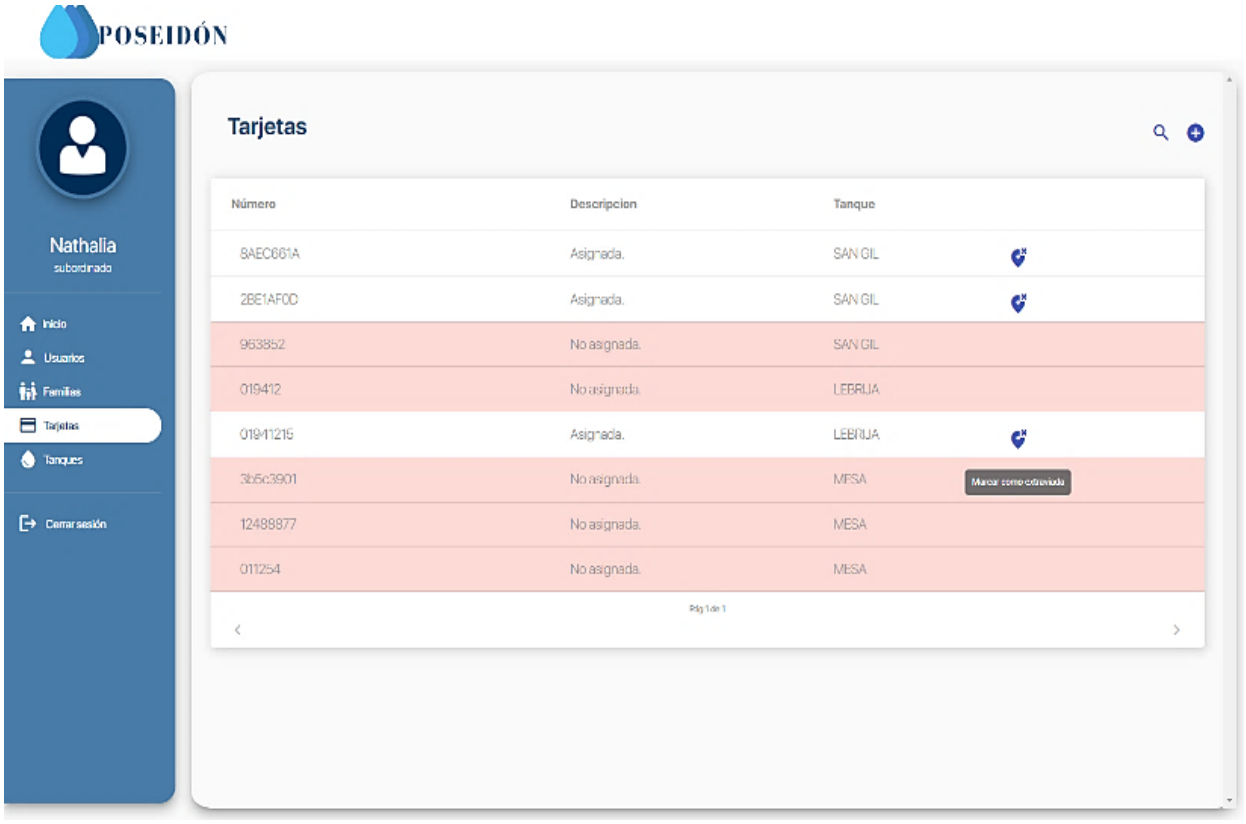
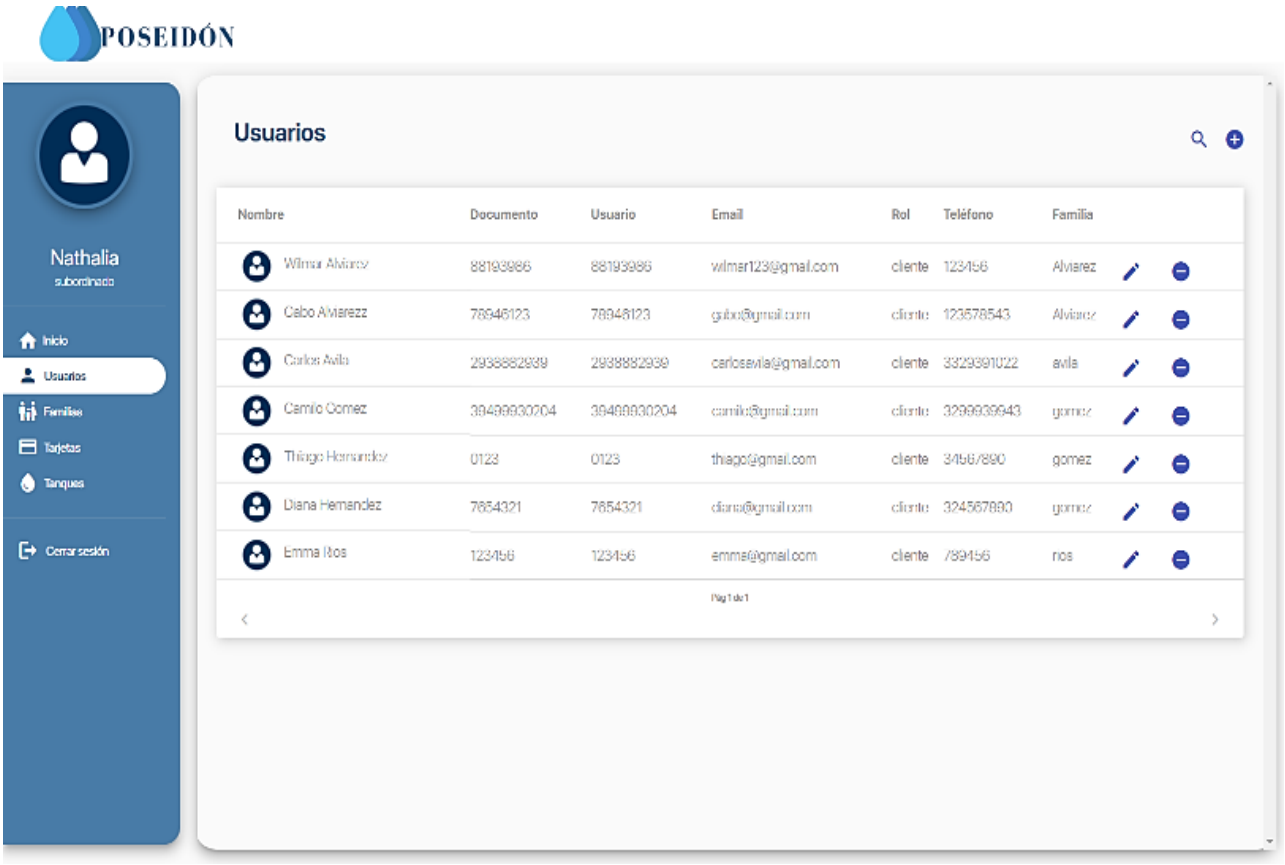


Figura 33.

Vista listar usuarios "clientes" para subordinados



5.4 Pruebas a la Plataforma IoT

La plataforma en este punto ya se encontraba desarrollada, por lo tanto, se continuo con la realización de pruebas las cuales tuvieron como objetivo verificar que todos los componentes como la aplicación web, conexión con el servidor, base de datos y sistema de suministro, se encontraron

funcionando de manera correcta, de esta manera se analizaron los resultados verificando si se presentaron errores en el funcionamiento, Las pruebas realizadas fueron:

5.4.1 Pruebas de Verificación

Estos tipos de pruebas son unas de las más implementadas, las cuales se realizan aplicando técnicas para detectar errores. Se desarrollaron con el objetivo de verificar que los datos obligatorios no quedaran vacíos, los tipos de datos establecidos si estuvieran correctamente protegidos frente a errores en la digitación y almacenamiento, también que los tipos de usuarios fueran autenticados de manera correcta y sus permisos en la aplicación web estuvieran implementados de manera correcta.

5.4.2 Pruebas de Integración

Una vez desarrollados los servicios, se implementaron estas pruebas para verificar que todos funcionaran de manera correcta, validando que las consultas a la base datos se hayan realizada de manera correcta, también se analizó que se estuvieran visualizando los servicios en la aplicación web de la manera establecida.5.4.3 Pruebas de Integración

Con el objetivo de comprobar que la plataforma IoT funcionara de manera correcta se realizaron diversas pruebas, simulando tanto transacciones que están aprobadas como las que no. De esta manera se constató que efectivamente el sistema de suministro se desarrolló de manera correcta, también se verifico que la aplicación web se estuviera visualizando de la forma correcta el tráfico realizado en los tanques de almacenamiento. Por lo tanto, con los resultados en las pruebas de integración y las otras pruebas anteriormente realizadas se obtuvieron ciertas fallas las cuales fueron solucionadas, dejando de esta manera la plataforma IoT y todos sus componentes listos como un primer prototipo y con un funcionamiento óptimo para su presentación.

5.4.3 Pruebas Evaluativas del Prototipo

Para evaluar el prototipo se plantearon los distintos casos que pueden presentarse al momento de usarlo y como responde tanto la plataforma web como el sistema de suministro a nivel de maqueta en sus todas sus etapas

Etapa de Autenticación de usuarios en el sistema de suministro de agua, en esta etapa la persona se autentica acercando su tarjeta asignada al lector RFID, internamente el ESP8266 envía el tópico 1 *tanque_serie/acces_query* del inciso 4.3.2 planteamiento de tópicos a NodeJs, con el fin de consultar si esta tarjeta está registrada y cuenta con acceso al tanque correspondiente. En el siguiente paso NodeJs envía como respuesta el tópico 2 *tanque_serie/command* del inciso 4.3.2 planteamiento de tópicos, esta información varía según los tres escenarios posibles, los cuales son:

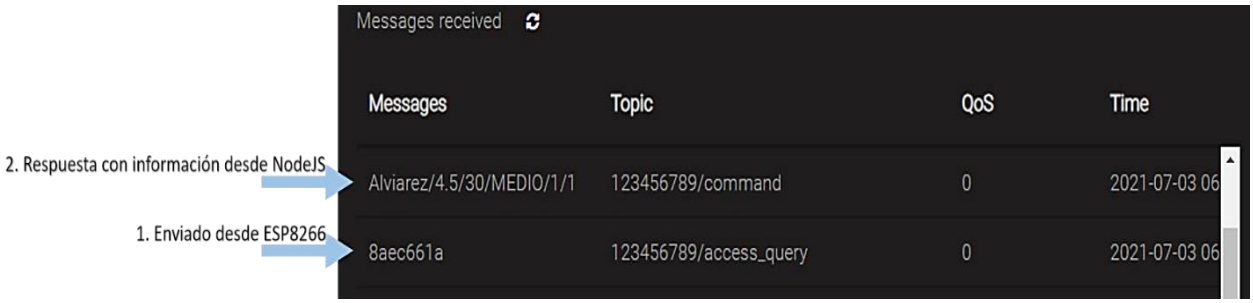
- Usuario es autenticado y el tanque está disponible (Nivel de agua diferente a bajo): en este caso en el tópico se envía información como el alias de su familia, capacidad del tanque, mínimo vital disponible, entre otros. Estos datos son necesarios para continuar con la transacción en sus siguientes etapas.

Figura 34.

Información del beneficiario autenticado

Figura 35.

Vista del dashboard de EMQ X con tópicos involucrados en el escenario expuesto



- Usuario es autenticado y el tanque no está disponible (Nivel de agua bajo): cuando la consulta realizada por NodeJs a la base de datos da como resultado que el tanque no se encuentra disponible debido a su nivel de agua actual, el mensaje recibido por el ESP8266 es tanqueout, mostrándole al usuario por medio de la pantalla LCD el mensaje “Fuera de servicio”, activando el sistema de llenado del tanque el cual se ejecuta hasta que el nivel del tanque sea alto. Cuando esto sucede el ESP8266 enviar el tópico 4 *tanque_serie/nivelTanque* del inciso 4.3.2 planteamiento de tópicos a NodeJs, el cual procede a actualizar el estado del tanque activándolo y modificando su nivel de agua.

Figura 36.

Vista del dashboard de EMQ X, tópicos involucrados cuando el tanque está fuera de servicio

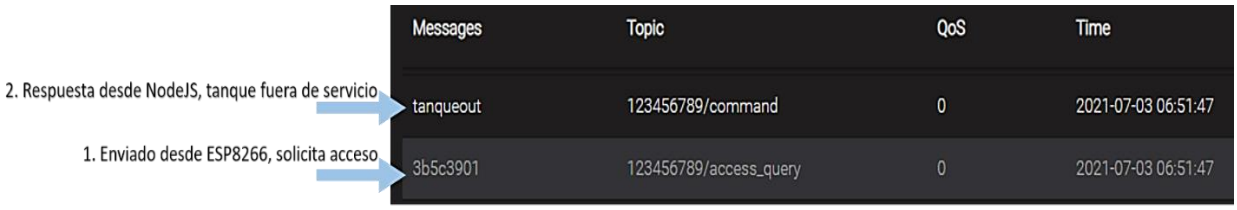


Figura 37.

Respuesta de la maqueta en caso tal de que el tanque se encuentre fuera de

Figura 38.

Reporte en la página web de tanque fuera de servicio

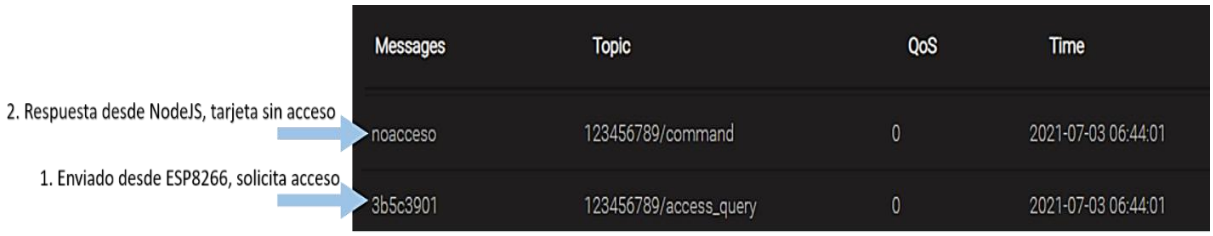
- Usuario no es autenticado: en este escenario se obtiene como resultado en el tópico recibido por el ESP8266 que efectivamente la tarjeta no está registrada en el tanque de almacenamiento, notificándole al usuario en la pantalla LCD el mensaje “sin acceso”, por lo tanto, la transacción es cancelada.

Figura 40.

Respuesta de la maqueta en caso de autenticación fallida

Figura 39.

Vista del dashboard EMQ X. Tópicos involucrados en autenticación fallida



Etapa de selección de cantidad de agua que desea retirar el usuario del sistema de suministro, una vez el usuario se encuentra autenticado correctamente, se muestra en la pantalla LCD su cantidad de agua disponible y se procede a seleccionar la cantidad de suministro en litros. En este proceso de selección pueden presentarse tres escenarios posibles:

- Usuario selecciona valor correspondiente, es decir, que este valor sea un numero positivo y sea menor o igual a su suministro disponible: cuando el valor seleccionado es correcto se muestra en la pantalla LCD el mensaje “otorgando suministro” y se activa el sistema de salida de agua, este sistema se apaga cuando se verifica que efectivamente se retiró el agua seleccionada y se procede a enviar el tópico 3 *tanque_serie/datosagua* del inciso 4.3.2 planteamiento de tópicos a NodeJs, el cual almacena la información del tráfico y actualiza los datos tanto del tanque como de la familia correspondiente terminando la transacción

Figura 41.

Suministro otorgado

Figura 42.

Vista del dashboard EMQ X. Tópicos resultantes de transacción exitosa

Figura 43.

Otorgando suministro

- Usuario perteneciente a familia beneficiaría solicita suministro de agua mayor a su mínimo vital disponible: si el valor seleccionado es superior a su líquido vital disponible se muestra en la pantalla LCD “Supera el mínimo vital” y se solicita que seleccione nuevamente el valor de agua a retirar.

Figura 44.

Respuesta de la maqueta en caso tal de seleccionar un número de litros mayor al disponible

- Usuario perteneciente a familia beneficiaría solicita suministro de agua seleccionando un valor negativo: si el valor seleccionado es un numero negativo se muestra en la pantalla LCD el mensaje “Sin números negativos” y se solicita que seleccione nuevamente el valor de agua a retirar.

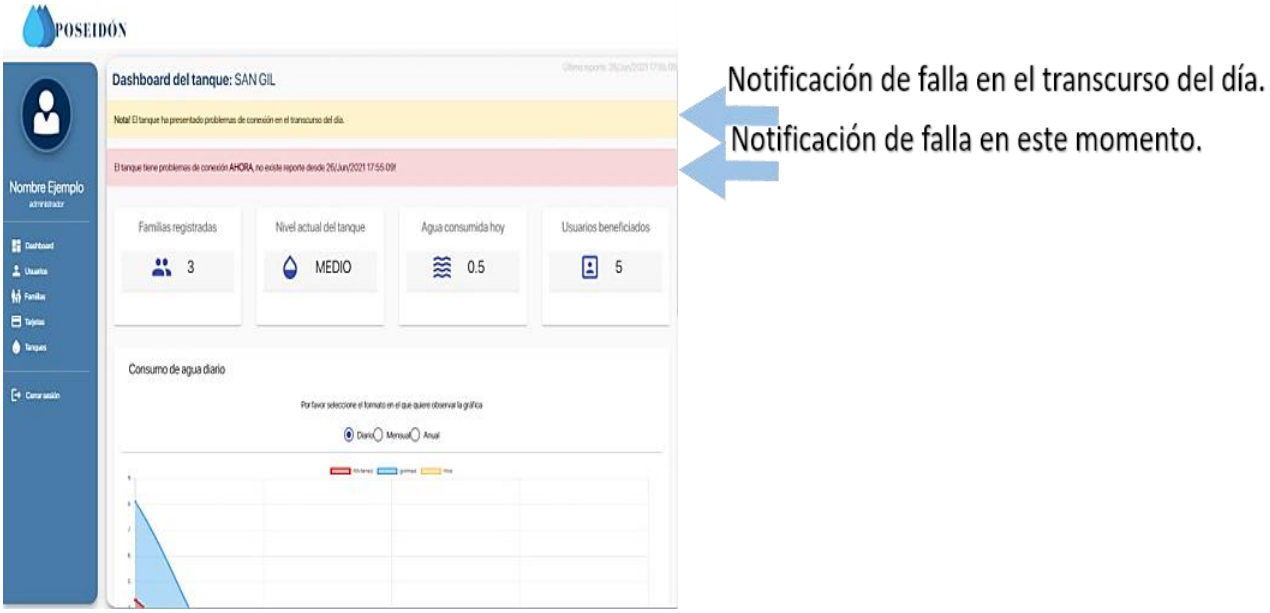
Figura 45.

Respuesta de la maqueta en caso tal de seleccionar un numero negativo

Por otra parte, se probó que efectivamente se estuviera realizando el reporte de fallas de conexión al Broker o wifi en la plataforma web desconectando la red de forma manual. Este reporte se puede visualizar en el dashboard del administrador, como se puede ver en la siguiente imagen.

Figura 46.

Dashboard en la página web, reporte de fallas.



6. Conclusiones

Este proyecto se desarrolló con el objetivo de diseñar una plataforma alternativa para el control del consumo de agua potable en tanques de almacenamiento comunitarios, ya que actualmente en las poblaciones donde no cuentan con acueducto los sistemas de suministros no alcanzan a ser equitativos. El proyecto aquí presentado logró cubrir las necesidades planteadas al principio de este, ya que, a través de sus distintas fases implementadas se llegó a un prototipo funcional para tal fin.

En cuanto a la formación tecnológica y levantamiento de requerimientos planteados al inicio del trabajo, se realizó capacitación e investigación de conceptos tecnológicos que eran necesarios para realizar el levantamiento de requerimientos de la plataforma y su desarrollo. Esta etapa fue muy importante ya que dio claridad sobre los alcances y delimitaciones que iba a tener la plataforma.

Para el análisis y selección de arquitectura la metodología utilizada permitió analizar un espectro amplio de posibilidades determinantes para desarrollar la plataforma IoT, así como seleccionar las más acorde a las necesidades. A lo largo del desarrollo del prototipo se comprobó la utilidad del análisis anteriormente expuesto, facilitó la interconexión y el correcto desarrollo de la plataforma.

Ya en la etapa de diseño y desarrollo del proyecto, el establecer una estructura y organización nos ayudó al momento de ir desarrollando los distintos componentes del prototipo, también hizo más sencillo enlazarlos. En esta fase se fue verificando los requerimientos establecidos mediante pruebas con el objetivo de identificar fallas y proporcionar soluciones de

manera efectiva. Esta etapa junto a la de validación de requerimientos fueron ejecutadas de manera repetitiva ya que constantemente se realizaron pruebas en estas dos fases y retroalimentación de las fallas que fueron surgiendo.

Para la validación de requerimientos del prototipo, se realizaron las pruebas que se expusieron en el inciso 4.5.4 tanto en la plataforma web, como en la conexión al servidor y el sistema de suministro. Con la ejecución de estas pruebas se validaron que los requerimientos fueron cumplidos y que la plataforma si cumple con el objetivo de ser una alternativa que controle y monitoree el suministro de agua de manera equitativa y efectiva a la sociedad. En la validación de requerimientos surgió la problemática de que en algunos casos la conexión por medio wifi no brinda la estabilidad necesaria, por lo tanto, se sugiere que en trabajos futuros se plantee un esquema de reglas offline y otra alternativa de conexión como por ejemplo fibra óptica para un mejor desempeño del sistema.

A fin de presentar el prototipo final, se desarrolló la maqueta a nivel escala del sistema, la cual funcionó como medio de realización de pruebas y corrección de errores del sistema de suministro permitiendo visualizar y comprobar los distintos escenarios con su respectivo funcionamiento. Además, se realizó la respectiva documentación sobre todos los procesos e investigación tecnológica que se consultó para obtener como resultado este primer prototipo de plataforma IoT de monitoreo y control de suministro de agua en tanques de almacenamiento comunitarios.

Teniendo en cuenta lo anteriormente mencionado, hemos podido llegar a un prototipo funcional de un sistema de control de suministro de agua potable para conectar los tanques de almacenamiento comunitario, el cual permite el control y monitoreo del sistema de suministro mediante una aplicación web en la cual se puede visualizar información acerca de los diferentes

usuario y componentes que hacen parte de la plataforma IoT. Dando respuesta a la problemática planteada que originó este trabajo.

7. Trabajo Futuro

- Actualización en nuevas tecnologías para la aplicación web como el desarrollo de un aplicativo móvil.
- Dimensionar los sensores y actuadores acorde a los tanques en los cuales se desee implementar el sistema de suministro de agua
- Implementar un módulo que brinde soporte técnico a los tanques de almacenamiento y los sensores que se encuentran en el sistema de suministro.
- Aplicación de reglas offline en caso de que la conexión con el broker falle.
- Implementar otro método de conexión diferente al wifi, el cual brinde mayor estabilidad.
- Implementar una prueba piloto del sistema.
- Plantear una estrategia de pruebas de desempeño al software.

Referencias Bibliográficas

.UNICEF (2019a, junio 18). 1 de cada 3 personas en el mundo no tiene acceso a agua potable. https://www.unicef.org/es/comunicados-prensa/1-de-cada-3-personas-en-el-mundo-no-tiene-acceso-a-agua-potable#:%7E:text=Alrededor%20de%202.200%20millones%20de,para%20el%20lavado%20de%20manos

El Nuevo Siglo. (2021, 23 agosto). 800 mil muertes al año por falta de agua potable. https://www.elnuevosiglo.com.co/articulos/03-2019-800-mil-muertes-al-ano-por-falta-de-agua-potable

Abril, A. (s. f.). LAS 10 VENTAJAS PRINCIPALES DE USAR REACT.JS. SistemasGeniales.com Desarrollo de software y páginas web Colombia. Recuperado 9 de febrero de 2021, de http://sistemasgeniales.com/software/las-10-ventajas-principales-de-usar-react-js/

African Slum Journal. (2016, 16 agosto). Un cajero automático para sacar agua. EL PAÍS. https://elpais.com/elpais/2016/08/15/planeta_futuro/1471272030_740041.html#:%7E:text=Un%20sistema%20de%20tarjeta%20prepago,historia%20de%20African%20Slum%20Journal&text=Cuando%20tienen%20necesidad%20de%20agua,grifo%20para%20llenar%20sus%20recipientes.

América Latina: la región con más agua, la más castigada por la sed. (2015, 13 mayo). World Bank. https://www.bancomundial.org/es/news/feature/2015/05/13/america-latina-la-region-con-mas-agua-la-mas-castigada-por-la-

sed#:%7E:text=%22%20Aproximadamente%2037%20millones%20de%20personas,%2C%20Ecuador%2C%20Per%C3%BA%20y%20Bolivia.

Andrés, R. (2014, 16 diciembre). ¿Qué es y para qué sirve el dominio de tu página web? ComputerHoy. https://computerhoy.com/noticias/internet/que-es-que-sirve-dominio-tu-pagina-web-22007

Atlassian. (s. f.). Qué es el control de versiones | Atlassian Git Tutorial. Recuperado 18 de abril de 2021, de https://www.atlassian.com/es/git/tutorials/what-is-version-control

B., G. (2021, 16 febrero). ¿Qué es un hosting y cómo funciona? Hosting web para principiantes. Tutoriales Hostinger. https://www.hostinger.co/tutoriales/que-es-un-hosting#:%7E:text=Hosting%20VPS%20(Servidor%20privado%20virtual,Hosting%20con%20servidor%20dedicado

Cloud Computing: ¿Qué es y cuáles sus ventajas? (s. f.). Salesforce.com. Consultado 11 de agosto de 2020, de https://www.salesforce.com/mx/cloud-computing/

De control, U. I., Inmótica, S. D. o., & de control y actuadores., S. (n.d.). Sensores, actuadores y elementos del sistema de control. Recuperado April 19, 2021, Micronica.es website: http://www.micronica.es/files/pdfs/SIHD/SIHD_Sens_Actu_EC.pdf

Editores de código. (2020). Desarrollo Web. https://desarrolloweb.com/colecciones/editores-codigo

Fernández, L. (2020, 21 marzo). Protocolos de redes: la guía completa con todos los protocolos básicos. RedesZone. https://www.redeszone.net/tutoriales/internet/protocolos-basicos-redes/

Fernández, Y. (2018, 28 agosto). Arduino y Raspberry Pi: qué son y cuáles son sus diferencias. Xataka. https://www.xataka.com/basics/arduino-raspberry-pi-que-cuales-sus-diferencias

Juanda. (n.d.). Arquitectura de un SPA · Desarrollo de aplicaciones web. Recuperado 11 de agosto de 2020, from Gitbooks.io, https://juanda.gitbooks.io/webapps/content/spa/arquitectura_de_un_spa.html

L. (2021b, April 15). Controlar GPIO y PWM del ESP8266/ESP32 desde interface Web con Vue y Websockets. Luis Llamas. Recuperado 11 de agosto de 2020, https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/

L. (2019, December 3). Principales broker MQTT Open Source para proyectos IoT. Luis Llamas. Recuperado 10 de julio de 2020, https://www.luisllamas.es/principales-broker-mqtt-open-source-para-proyectos-iot/

López, J. (2019, 22 octubre). Qué es el ESP8266 y qué ventajas aporta a la plataforma Arduino. HardZone. Recuperado 17 de julio de 2020, https://hardzone.es/reportajes/tema/esp8266-2n2222-arduino/

Medidores, A. P. D. (s. f.). Sensores. PCE. Recuperado 10 de agosto de 2020, https://www.pce-iberica.es/instrumentos-de-medida/sistemas/sensores.htm

Mendoza, M. L. (2020, August 10). Qué es un lenguaje de programación. OpenWebinars.net. Recuperado 10 de Julio de 2020, https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/

Moncayo, J. M. R. (2020, 25 junio). ¿Qué es REST? Conoce su potencia. OpenWebinars.net. Recuperado 8 de agosto de 2020, https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/

Negocio, I. P. T. (2018, June 21). ¿Qué es un servidor y para qué sirve? Informática para tu negocio. Recuperado 10 de agosto de 2020, https://www.informaticaparatunegocio.com/blog/que-es-un-servidor-y-para-que-sirve/

N. (2020, 21 octubre). Internet. Concepto. https://concepto.de/internet/

N. (2020, 30 septiembre). ¿Qué es Back-End, Front-End y Back Office y por qué es importante para tu web? Agencia Inbound Marketing Madrid. Recuperado 18 de agosto de 2020, https://nestrategia.com/desarrollo-web-back-end-front-end/#:%7E:text=En%20otras%20palabras%2C%20el%20Back,la%20comunicaci%C3%B3n%20con%20el%20servidor.

O. (2017, 20 febrero). Introducción a los WebSocket. Oscar Blancarte - Software Architecture. https://www.oscarblancarteblog.com/2017/02/20/introduccion-a-lo-websocket/

A. (2021, 7 marzo). Microcontrolador - qué es y para qué sirve. Recuperado 10 de agosto de 2020, HETPRO/TUTORIALES. https://hetpro-store.com/TUTORIALES/microcontrolador/

Pérez, L. R. (s. f.). Tanque de almacenamiento. sswm.info. Recuperado 10 de agosto de 2020, de https://sswm.info/es/gass-perspective-es/tecnologias-de-agua-y-saneamiento/tecnologias-de-abastecimiento-de-agua/tanque-de-almacenamiento.

Qué es Framework. (s. f.). Arimetrics. Recuperado 17 de abril de 2021, de https://www.arimetrics.com/glosario-digital/framework

Salgado, D. C. (2013). servicio web sobre internet de las cosas o web of things. gradiant. Recuperado 10 de agosto de 2020, de https://www.gradiant.org/noticia/servicios-web-sobre-internet-de-las-cosas-o-web-of-things-2/

Sistemas de Control. (s. f.). Definición.xyz. Recuperado 18 de abril de 2021, de https://www.tecnologiatecnica.com.ar/sistemadecontrol/index%20sistemasdecontrol_archivos/Page381.html

Tolima, A. (2019, 14 julio). Ocho de cada cien habitantes de Colombia no tienen agua potable. Alerta Tolima. Recuperado 10 de agosto de 2020, https://www.alertatolima.com/noticias/politica/ocho-de-cada-cien-habitantes-de-colombia-no-tienen-agua-potable#:%7E:text=Ocho%20de%20cada%20cien%20habitantes%20de%20Colombia%20no%20tienen%20agua%20potable,Alerta&text=La%20existencia%20de%20familias%20sin,la%20cobertura%20no%20es%20total

Telemetrik. (s. f.). Medición en tiempo real de caudal y de nivel en el Acueducto Municipal de Anorí. Recuperado 12 de agosto de 2020, de https://telemetrik.co/project/aplicacion-medicion-tiempo-real-caudal-nivel-acueducto-municipal-anori/.

TicJob, R. (2019, March 4). Programador Frontend: ¿Qué tareas lleva a cabo? Zona Ticjob - MCPRO. https://www.muycomputerpro.com/zona-ticjob/que-es-un-programador-frontend/

Valois, M. A. (2019, 12 agosto). Qué es internet de las cosas y cómo funciona. Blog HostGator México. Recuperado 10 de agosto de 2020, https://www.hostgator.mx/blog/internet-de-las-cosas/.

Ventajas de Angular para crear aplicaciones web - Bloguero Pro. (s. f.). Ventajas de Angular. Recuperado 9 de febrero de 2021, de https://blogueropro.com/blog/ventajas-de-angular-para-crear-aplicaciones-web

Apéndices

Apéndice A.

Esquema de la metodología implementada en el desarrollo del proyecto