

DESARROLLO DE UN SISTEMA WEB PARA EL ANÁLISIS Y VISUALIZACIÓN DE DATOS
EN ALERTAS TEMPRANAS DE CALIDAD DEL AGUA POR TELEMETRÍA PARA
ACUEDUCTOS RURALES.



DARWIN STEVEN GOMEZ CEBALLOS

INSTITUCIÓN UNIVERSITARIA COLEGIO MAYOR DEL CAUCA
FACULTAD DE INGENIERÍA
INGENIERÍA INFORMÁTICA
POPAYÁN CAUCA
2025

DESARROLLO DE UN SISTEMA WEB PARA EL ANÁLISIS Y VISUALIZACIÓN DE
DATOS EN ALERTAS TEMPRANAS DE CALIDAD DEL AGUA POR TELEMETRÍA
PARA ACUEDUCTOS RURALES.



Proyecto de grado modalidad práctica requisitos parcial para optar por el título de
Ingeniería Informática

DARWIN STEVEN GOMEZ CEBALLOS

Asesor: MsC(c): Manuel Arturo Melo Legarda
Asesor Empresarial: MsC. Ing. Henry Eduardo Bastidas

INSTITUCIÓN UNIVERSITARIA COLEGIO MAYOR DEL CAUCA
FACULTAD DE INGENIERÍA
INGENIERÍA INFORMÁTICA
POPAYÁN CAUCA
2025

TABLA DE CONTENIDO

Introducción	1
Antecedentes	3
I. Planteamiento del problema	3
II. Justificación	5
III. Objetivos	9
Objetivo General	9
Objetivos Específicos	9
IV. Metodología	10
V. Estado del Arte	11
CAPÍTULO 1. LEVANTAMIENTO DE REQUERIMIENTOS	22
1. Reunión de planeación	23
1.1 Requisitos funcionales y no funcionales	23
1.2. Especificaciones técnicas definidas	24
1.3. Funcionalidad del producto	27
2. Casos de uso	31
3. Historias de usuario	39
3.1 Módulo Autenticación y Gestión de Usuarios	40
3.2 Módulo de monitoreo de datos en tiempo real	42
3.3 Módulo de monitoreo de alertas	43
3.4 Módulo de Almacenamiento	44
CAPÍTULO 2. CONSTRUCCIÓN DEL SISTEMA WEB PARA EL ANÁLISIS Y VISUALIZACIÓN DE DATOS EN ALERTAS TEMPRANAS DE CALIDAD DEL AGUA POR TELEMETRÍA PARA ACUEDUCTOS RURALES.	45
1. Diagrama de secuencia	46
1.1 Diagrama de secuencia Inicio de sesión	46
1.2 Diagrama de secuencia registro de usuario	46
1.3 Diagrama de secuencia Dashboard	48
1.4 Diagrama de secuencia editar usuario	48
1.5 Diagrama de secuencia Cambiar contraseña	50
1.6 Diagrama de secuencia Recuperar contraseña	50
1.7 Diagrama de secuencia Cerrar sesión	51
1.8 Diagrama de secuencia Visualizar datos en tiempo real	52

1.9	Diagrama de secuencia Lista de usuarios editar y eliminar.....	53
1.10	Diagrama de secuencia Generación de alertas	55
1.11	Diagrama de secuencia Notificaciones de alertas.....	56
1.12	Diagrama de secuencia seleccionar sensores	57
2.	Diseño de Mockups	57
2.1	Diseño de interfaz inicial	58
2.2	Diseño de la interfaz recuperar contraseña.....	58
2.3	Diseño de la interfaz de registro	59
2.4	Diseño de la interfaz principal Dashboard.....	59
2.5	Diseño de la interfaz en tiempo real:.....	60
2.6	Diseño de la interfaz Tabla de datos críticos:.....	61
2.7	Diseño de la interfaz Lista de usuario Registrados:	61
2.8	Diseño de la interfaz Editar usuario:	62
2.9	Diseño de la interfaz Cambiar Contraseña.....	62
2.10	Diseño de la interfaz Seleccionar sensores:	63
2.11	Diseño de la interfaz Alerta:.....	63
2.12	Diseño de la interfaz Cerrar sesión:	63
3.	Estructura del proyecto.....	64
3.1	Arquitectura del sistema IoT	64
3.2	Proceso de envío de datos	64
3.3	Estructura aplicación web	68
3.4	Componentes Claves del Backend	69
3.5	Tecnologías Utilizadas Backend	77
3.6	Estructura del frontend React y vite	77
3.7	Componentes Claves del frontend.....	78
CAPITULO 3. PRUEBAS DE SOFTWARE		83
1.	Metodología de evaluación.....	84
2.	Acta de confidencialidad.....	84
3.	Pre-Test y resultados	86
4.	Actividades Realizadas y Registro de Tiempos	89
5.	Post-test resultados.....	90
Conclusión		94
Recomendaciones		94

Trabajos Futuros.....	95
Referencias.....	97

TABLA DE FIGURAS

Figura 1 Diagrama De Ishikawa	4
Figura 2 Grafana	6
Figura 3 Html, Css Bootstrap	6
Figura 4 Nodejs y Express	7
Figura 5 MongoDB	7
Figura 6 GPRS.....	7
Figura 7 Comandos AT	8
Figura 8 Nodejs Socket.io	8
Figura 9 ArduinoJson	9
Figura 10 Fases de la Metodología	10
Figura 11 Data Received From Node-Red	13
Figura 12 Dashboard or Remote River Monitoring System	14
Figura 13 AquaWatch.....	14
Figura 14 HidroVu	15
Figura 15 Kisters Wiski.....	16
Figura 16 Sensores Vernier	19
Figura 17 Sensor pH Vernier.....	19
Figura 18 Sensor Turbidez Vernier	19
Figura 19 Sensor Conductividad Vernier.....	20
Figura 20 Sensor de Oxigeno Vernier	20
Figura 21 Arduino Vernier	21
Figura 22 Diagrama de Flujo de Proceso.....	27
Figura 23 Diagrama Caso de Uso.....	28
Figura 24 Diagrama de Secuencia Inicio de Sesión.....	46
Figura 25 Diagrama de Secuencia Registro de Usuario	47
Figura 26 Diagrama de Secuencia Dashboard	48
Figura 27 Diagrama de Secuencia Editar Usuario	49
Figura 28 Diagrama de Secuencia Cambiar Contraseña	50
Figura 29 Diagrama de Secuencia Recuperar Contraseña.....	51
Figura 30 Diagrama de Secuencia Cerrar Sesión.....	52
Figura 31 Diagrama de Secuencia Visualizar Datos en Tiempo Real.....	52
Figura 32 Diagrama de Secuencia Listar Usuarios	53
Figura 33 Diagrama de Secuencia Editar Usuarios.....	54
Figura 34 Diagrama de Secuencia Eliminar Usuarios	55
Figura 35 Diagrama de Secuencia Generación de Alertas	56
Figura 36 Diagrama de Secuencia Notificación de Alertas	56
Figura 37 Diagrama de Secuencia Seleccionar Sensores	57
Figura 38 Logo de Balsamiq	58
Figura 39 Interfaz Inicio de Sesión	58

Figura 40 Interfaz Recuperar Contraseña	59
Figura 41 Interfaz Registro de Usuario.....	59
Figura 42 Interfaz Principal Dashboard	60
Figura 43 Interfaz Tiempo Real.....	61
Figura 44 Interfaz Tabla Datos Críticos.....	61
Figura 45 Eliminar Lista De Usuarios Registrados	62
Figura 46 Interfaz Editar Usuario	62
Figura 47 Interfaz Seleccionar Sensores	63
Figura 48 Alerta.....	63
Figura 49 Cerrar Sesión	64
Figura 50 Proceso Envío de Datos.....	64
Figura 51 Formulas y Conversión	65
Figura 52 Formula Conversion de pH	65
Figura 53 Formula Conversion de Turbidez	66
Figura 54 Formula Conversion de Conductividad	66
Figura 55 Formula Conversion de Oxigeno.....	66
Figura 56 Serialización Json	67
Figura 57 Transmisión de Datos	67
Figura 58 Ejemplo de Envío de Datos	67
Figura 59 Arquitectura del Modelo Backend Frontend	68
Figura 60 Estructura del Backend	69
Figura 61 Conexión a La Base de Datos.....	70
Figura 62 Modelo de Usuario	71
Figura 63 Modelo de Sensores	72
Figura 64 Controlador de Usuario	73
Figura 65 Controlador de Datos de Los Sensores	74
Figura 66 Definición de Rutas	74
Figura 67 Comunicación en Tiempo Real WebSockets	75
Figura 68 Configuración y Funcionamiento del Servidor.....	76
Figura 69 Estructura del Frontend React y Vite	77
Figura 70 Configuración de Rutas y Autenticación en React	78
Figura 71 Encapsulamiento de La Aplicación	79
Figura 72 Conexión con El Servidor Mediante Socker.io	80
Figura 73 Implementación de Inicio de Sesión.....	81
Figura 74 Implementación del Dashboard.....	81
Figura 75 Implementacion del Menú y Alertas en Tiempo Real	82
Figura 76 Participante 3	85
Figura 77 Participante 2	85
Figura 78 Participante 1	85
Figura 79 Pre-Test Evidencia.....	87

Figura 80 Resultado 1 Pre-Test	88
Figura 81 Resultado 2 Pre-Test	88
Figura 82 Resultado 3 Pre-Test	89
Figura 83 Resultado 1 Post-Test.....	91
Figura 84 Resultado 2 Post-Test.....	92
Figura 85 Resultado 3 Post-Test.....	93
Figura 86 Resultado 4 Post-Test.....	93

TABLA DE TABLAS

Tabla 1 Cuadro comparativo de Tecnologías.....	5
Tabla 2 Descripción de las Funcionalidades y Capacidades del Sistema AquaWatch .	14
Tabla 3 Características Clave de HidroVu en la Gestión y Visualización de Datos	15
Tabla 4 Descripción de las Funcionalidades del Sistema WISKI para Análisis y Gestión de Datos Ambientales	17
Tabla 5 Comparativa de Soluciones de Monitoreo de Calidad del Agua.....	17
Tabla 6 Requisitos Funcionales de los Módulos del Sistema.....	24
Tabla 7 Especificaciones Técnicas para la Aplicación web del Software Calidad del Agua.....	26
Tabla 8 Tabla de Funcionalidades	27
Tabla 9 Inicio de Sesión	31
Tabla 10 Registro de Usuario.....	32
Tabla 11 Editar Perfil del Usuario.....	33
Tabla 12 Cambiar Contraseña	33
Tabla 13 Recuperar Contraseña	34
Tabla 14 Cerrar Sesión	35
Tabla 15 Visualizar Datos en Tiempo Real	35
Tabla 16 Lista de Usuarios Registrados.....	36
Tabla 17 Generación de Alertas automáticas	37
Tabla 18 Notificaciones de Alertas.....	38
Tabla 19 Seleccionar Sensores	38
Tabla 20 Historia de Usuario Iniciar Sesión	40
Tabla 21 Historia de Usuario Registro.....	40
Tabla 22 Historia de Usuario Cerrar Sesión.....	40
Tabla 23 Historia de Usuario Cambiar Contraseña.....	41
Tabla 24 Historia de Usuario Editar Usuario	41
Tabla 25 Historia de Usuario Listar Editar y Eliminar Usuarios	41
Tabla 26 Historia de Usuario Validar Datos Ingresados.....	42
Tabla 27 Historia de Usuario Visualizar Datos en Tiempo Real	42
Tabla 28 Historia de Usuario Dashboard	42
Tabla 29 Historia de Usuario Datos Críticos	43
Tabla 30 Historia de Usuario Seleccionar Sensores	43
Tabla 31 Historia de Usuarios Generación de Alertas.....	43
Tabla 32 Historia de Usuario Generación de Notificaciones	44
Tabla 33 Historia de Usuario Almacenar Datos	44
Tabla 34 Pre-Test.....	86
Tabla 35 ACTIVIDADES Y REGISTRO DE TIEMPO	89
Tabla 36 Post-Test.....	90

DEDICATORIA

Dedico este proyecto a mi querido padre, Gómez Rosero Tiberio Nativel, y a mi madre, Gloria Amparo Ceballos, y hermana Nelly Natalia Gómez, por su apoyo incondicional y por siempre creer en mí, brindándome todo lo necesario para alcanzar mis sueños. Mi más sincero reconocimiento al Ingeniero Manuel Arturo Melo Legarda, por compartir su valioso conocimiento y guiarme con paciencia y dedicación a lo largo de este proyecto. Finalmente, al Ingeniero Henry Eduardo Bastidas, por su apoyo y compromiso durante mis prácticas profesionales, siempre asegurándose de que cumpliera con mis responsabilidades.

AGRADECIMIENTO

Agradezco profundamente a mis padres por su amor, sacrificio y apoyo incondicional, y a toda mi familia por acompañarme en cada paso de este camino académico.

Mi más sincero agradecimiento al Ingeniero Manuel Arturo Melo Legarda, asesor de este proyecto, por su invaluable guía, paciencia y por compartir su conocimiento a lo largo de todo el proceso.

Agradezco también al Ingeniero Henry Eduardo Bastidas mi asesor de prácticas, por su continuo respaldo durante las prácticas profesionales.

Agradezco al Instituto Universitario Colegio Mayor del Cauca, mi alma mater, por proporcionarme no solo las herramientas y conocimientos necesarios para mi formación profesional, sino también por fomentar en mí una actitud de crecimiento continuo y aprendizaje que ha sido clave en mi desarrollo académico y personal.

Agradezco al Centro de Teleinformática y Producción Industrial CTPI-SENA, donde realicé mis prácticas, por brindarme la oportunidad de aplicar y expandir mis conocimientos en un entorno profesional, permitiéndome enfrentar desafíos reales que enriquecieron mi experiencia práctica y contribuyeron significativamente a mi desarrollo profesional.

Agradezco también a todas las personas que, de alguna u otra manera, estuvieron allí brindándome su apoyo, asesoramiento y orientación durante este proceso, lo cual fue fundamental para el desarrollo de mi trabajo.

Introducción

La creciente preocupación por la falta de agua potable en muchas partes del mundo, sumada al deterioro de las fuentes de agua, han generado la necesidad urgente de implementar herramientas simples pero efectivas que permitan monitorear la calidad del agua en tiempo real. Sin embargo, uno de los mayores retos que se enfrentan es la visualización efectiva de estos datos puesto que la calidad del agua varía constantemente, es crucial que el software de monitoreo puedan presentar la información de forma clara, accesible y en tiempo real, permitiendo una interpretación rápida y precisa. Sin una visualización adecuada, incluso los datos más detallados pueden resultar inútiles para la toma de decisiones inmediatas y la implementación de acciones correctivas en caso de detectar niveles peligrosos de contaminantes en el análisis. Además de la necesidad de monitorear la calidad del agua en tiempo real, surge la problemática de la visualización efectiva de estos datos. La calidad del agua depende de múltiples factores como: pH, turbidez, oxígeno disuelto y conductividad que al monitorearse constantemente esto hace que se generen grandes volúmenes de información. Un reto importante radica en la presentación de la información de manera sutil para usuarios de distintos niveles técnicos, desde operadores locales hasta autoridades de la salud pública. Este proyecto busca desarrollar un software de monitoreo de la calidad del agua utilizando tecnologías de Internet de las Cosas (IoT). El software empleará un dispositivo Arduino Vernier y un módulo SIM7600X HAT, equipados con sensores de marca *Vernier*, para recolectar datos clave sobre la calidad del agua.

Estos datos se transmitirán de forma inalámbrica mediante redes móviles sobre el estándar de comunicación GSM (Global System for Mobile Communications) para la comunicación de teléfonos móviles y la extensión GPRS (General Packet Radio Service) que permite la transmisión de datos de una forma más eficiente, lo cual es ideal para zonas donde las conexiones Wi-Fi son limitadas o inexistentes. El backend, desarrollado con el entorno de desarrollo de JavaScript Node.js y el marco de trabajo Framework Express.js, permitirá recibir, procesar y almacenar estos en una base de datos. Posteriormente, la información se pondrá a disposición a través de una API. Para visualizar y analizar los datos, en esto se implementará un Dashboard interactivo con Grafana, que permitirá un seguimiento en tiempo real de parámetros como el pH, la turbidez del agua entre otras características. Además, el software generará gráficos y enviará alertas automáticas si los indicadores superan los límites establecidos por las autoridades competentes.

El proyecto está diseñado para ayudar a los encargados de la gestión de recursos hídricos en zonas rurales de difícil acceso y también urbanas, donde la infraestructura tecnológica es escasa. Ya que el monitoreo de calidad del agua implica manejar datos

de múltiples variables, la visualización de estos datos se convierte en una herramienta principal. El software no solo recolecta información en tiempo real, sino que la muestra en un formato fácil de entender visualmente, permitiendo que también usuarios con conocimientos técnicos limitados puedan interpretar los resultados y así reaccionar rápidamente ante cualquier problema de calidad del agua., el enfoque se centra en crear gráficos y alertas en tiempo real que destaquen visualmente los cambios en los parámetros de calidad del agua. Esta presentación favorece una respuesta inmediata y efectiva, especialmente en áreas donde el acceso a Wi-Fi no es accesible y el software debe depender de redes móviles.

Entre los resultados previstos está la creación de un software robusto de monitoreo que no solo reúna la información al instante sobre la calidad del agua, sino que también ofrezca herramientas avanzadas de visualización. La visualización detallada de los datos históricos facilitará a los encargados identificar variaciones en la calidad del agua a lo largo del tiempo, lo que hará posible la predicción de problemas recurrentes y la planificación de medidas preventivas. Además, el software generará alertas automáticas ante cualquier desviación de los parámetros de calidad establecidos, permitiendo una respuesta rápida en situaciones de emergencia.

Antecedentes

I. Planteamiento del problema

La creciente alarma a la contaminación del agua y la falta de acceso a fuentes de agua potable en numerosas regiones del mundo genera la necesidad de desarrollar soluciones tecnológicas que permitan monitorear la calidad del agua de manera eficaz y en tiempo real. En zonas rurales o de difícil acceso, donde la infraestructura para el monitoreo constante no está disponible, el crecimiento de información oportuna sobre la calidad del agua puede poner en riesgo la salud de la población y el medio ambiente.

Las organizaciones de saneamiento enfrentan numerosos obstáculos para implementar aplicaciones web de monitoreo eficientes, especialmente en regiones rurales y de bajos recursos. (Méndez, 2020) Estas aplicaciones son esenciales para identificar y corregir irregularidades en tiempo real, sin embargo, su implementación suele verse limitada por factores como la falta de financiamiento, déficit de infraestructura adecuada, y escasez de recursos humanos capacitados para gestionar el monitoreo de manera continua. Según UNICEF y OMS, los sistemas de saneamiento dependen de políticas sólidas y financiamiento sostenible para ser eficaces, lo cual muchas veces no es posible en países con estructuras institucionales frágiles. (*Monitoring water, sanitation and hygiene*, 2024) (*Strengthening Water, Sanitation and Hygiene (WASH) Systems | UNICEF*, 2024) También la tecnología IoT ha revolucionado la capacidad para monitorear la calidad del agua en tiempo real, permitiendo recopilar y transmitir datos críticos, como pH, turbidez, y oxígeno disuelto, que son vitales para la toma de decisiones informadas en gestión hídrica. Sin embargo, muchas de estas soluciones en el mercado requieren una inversión considerable en software y en infraestructura de datos, lo que limita su viabilidad en áreas de bajos recursos o con conectividad limitada. Estas soluciones comerciales suelen demandar una infraestructura de red robusta, como el acceso constante a Wifi o a datos móviles, lo cual es una barrera en entornos rurales o aislados donde la conectividad es limitada. (Iancu et al., 2024) (*Evaluación de La Calidad Del Agua En Tiempo Real a Través de IoT*, 2024)

El reto central de este proyecto es desarrollar un software de monitoreo de la calidad del agua que sea seguro, confiable y escalable a futuro, utilizando tecnologías IoT, como sensores y módulos de comunicación móvil. Estos sensores se encargarán de recopilar datos sobre indicadores críticos de la calidad del agua, como pH (rango de 0 a 14)(*pH Sensor*, 2024), turbidez (NTU- Unidad de Turbidez Nefelométrica)(*Turbidity Sensor*, 2024), Oxígeno disuelto (0 a 20 mg/L- Miligramos por litro) (*Vernier Optical DO Probe*, 2024) y conductividad (0 a 2000 $\mu\text{S}/\text{cm}$ - microsiemens por centímetro) (*Conductivity Probe*, 2024) se encargará de transmitir a una plataforma central que los visualice en tiempo real. Esto permitirá a los encargados locales de los recursos hídricos detectar de

forma temprana problemas de calidad de agua y tomar las medidas correctivas antes de que se conviertan en amenazas significativas para la salud pública.

El diagrama de Ishikawa permite identificar las variables clave para afectan la eficacia en el monitoreo de la calidad del agua. A través de este análisis estructurado, pude detallar el problema en seis categorías: personas, software, hardware, ambiente, mediciones y datos/monitoreo.

Las variables más relevantes identificadas son aquellas relacionadas con el hardware, como los fallos en los sensores de pH, turbidez y oxígeno, que pueden dar lugar a datos inestables. Además, las variables relacionadas con datos y monitoreo son críticas, esto teniendo en cuenta que la perdida de información durante la transmisión representa un desafío importante. este proyecto busca mejorar la calidad de los datos y la consistencia del software de transmisión, buscando minimizar los errores en la captura y envío de información en tiempo real. Esto es vital para garantizar una respuesta rápida y efectiva en situaciones que amenazan la calidad del agua.

La calidad del agua es un asunto de mucha importancia para la salud pública y la sostenibilidad ambiental. Por otro lado, en muchas áreas, el monitoreo de la calidad del agua sigue siendo un desafío debido a la falta de tecnología accesible para detectar y reaccionar rápidamente a alteraciones negativas en tiempo real.

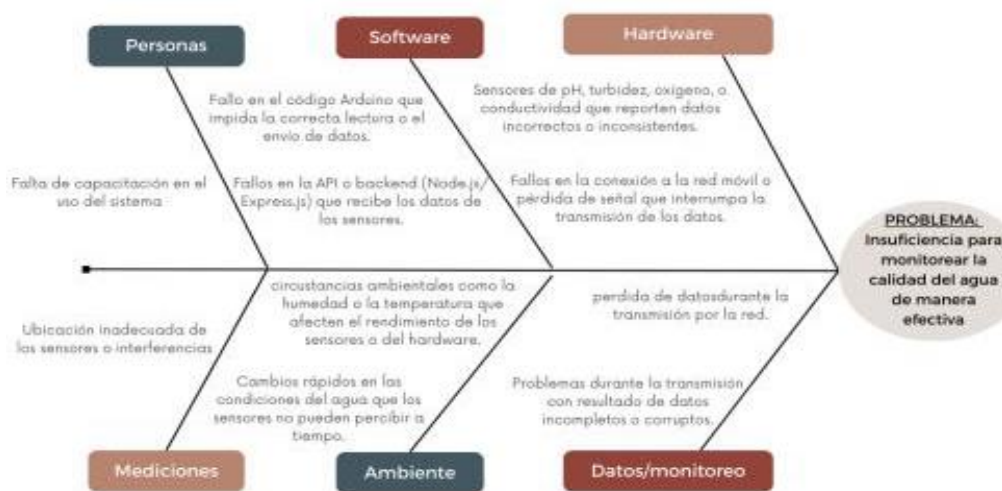


Figura 1 Diagrama De Ishikawa

Las herramientas tradicionales de monitoreo suelen ser costosas y requieren infraestructura avanzada en su instalación y uso, lo que complica su aplicación en áreas rurales o con recursos escasos. Esto limita la capacidad de los responsables para identificar problemas de calidad del agua de manera oportuna y tomar decisiones informadas. Se plantea la siguiente pregunta de proyecto:

¿De qué manera la implementación de una aplicación web para la visualización de datos y alertas tempranas mejora la gestión y la toma de decisiones en un sistema de monitorización sobre la calidad del agua?

La implementación de una aplicación web para la visualización de datos y alertas tempranas permite una gestión más eficaz de la calidad del agua al proporcionar información accesible en tiempo real y alertas oportunas, en el cual favorece en decisiones informadas y una reacción rápida antes eventuales inconvenientes que se puedan presentar.

II. Justificación

La mala calidad del agua es un factor grave para el cuidado de la salud pública y el medio ambiente, Por ende, la detección temprana de irregularidades podría ayudar a mitigar el problema de insalubridad que algunas veces no es diagnosticada a tiempo en las plantas de tratamiento. Sin embargo, en la actualidad muchas de estas plantas de saneamiento carecen de aplicaciones web de información que permitan emitir alertas tempranas de manera eficiente y accesible ante posibles anomalías. Grafana como plataforma principal para la creación de Dashboard, visualizaciones y alertas.

Tabla 1 Cuadro comparativo de Tecnologías

Tecnología	Funcionalidad	Ventajas	Desventajas
Grafana	Dashboard interactivos, alertas	Plataforma Robusta para visualización en tiempo real, alertas configurables, soporte para plugin.	Requiere configuración avanzada, curva de aprendizaje.
Tableau	Visualización y análisis	Visualizaciones avanzadas, fácil de usar.	Costoso, especialmente para los grandes equipos.
Power BI	Visualización y BI	Integración con Microsoft, buena para el análisis empresarial.	Limitado en opciones de alertas en tiempo real.
Kibana	Dashboard de datos de logs	Una buena integración con Elasticsearch, ideal para análisis de logs.	Enfocado en análisis de logs, menos flexible en gráficos.
Google Data Studio	Visualización de datos	Gratuito, fácil integración con Google Sheets	Opciones de visualización limitadas, sin alertas avanzadas.

Grafana: Seleccionada como la plataforma de visualización y alertas debido a su

capacidad para manejar datos en tiempo real y configurar alertas personalizadas, lo cual es crucial para monitorear la calidad del agua de manera eficiente, a diferencia de las demás plataformas Grafana ofrece integración directa con múltiples fuentes de datos además otra utilidad es que permite configurar alertas basadas en condiciones específicas, que permitirán notificar al usuario de forma inmediata ante cualquier anomalía.(*Grafana Dashboards | Grafana K6 Documentation, 2024*)



Figura 2 Grafana

Frontend (HTML, CSS, Bootstrap): Seleccionado por el desarrollo de la interfaz accesible y adaptable, siendo esencial para una visualización clara de interpretar para los usuarios, Borrar framework para desarrollo web que facilitara el desarrollo responsivo y acelera la implementación de diseños adaptados a diferentes dispositivos. (*HTMLandCSS - Ascent Infotech Computer Training Center Kolkata, 2021*)



Figura 3 Html, Css Bootstrap

Backend (Node.js, Express.js): Node.js y express.js permitirán un desarrollo rápido y eficiente del backend, para asegurar una comunicación fluida y en tiempo real entre la base de datos y el frontend, sabiendo que node.js es conocido por su alta capacidad de manejo en las solicitudes simultaneas, esto permite ser muy ventajoso para la transmisión de datos constantes que provienen de sensores de monitoreo. (*Understanding the Differences between JavaScript, Node.js, and Express.js, 2024*)



Figura 4 Nodejs y Express

MongoDB: Considerado debido a su capacidad para manejar grandes volúmenes de datos de una manera eficiente ya que ofrece una flexibilidad para datos no estructurados, siendo compatible con Grafana facilitando la visualización directa de los datos.(Rinke, 2021)



Figura 5 MongoDB

GPRS: Seleccionado para asegurar la transmisión de los datos en zonas rurales donde el acceso a internet puede ser limitado, permitiendo así que los sensores envíen datos a la plataforma central sin depender del uso de redes Wifi.(GPRS, 2024)



Figura 6 GPRS

Comandos AT: Seleccionado para interactuar con el módulo de comunicación SIM7600G-H 4G HAT lo cual permite controlar y configurar aspectos como la conexión GPRS. Esto asegura que la transmisión de datos sea precisa y que los sensores se mantengan conectados a la red móvil, incluso en condiciones de señal variable. («Comandos AT del módulo Bluetooth HC-05 y HC-06», 2023)

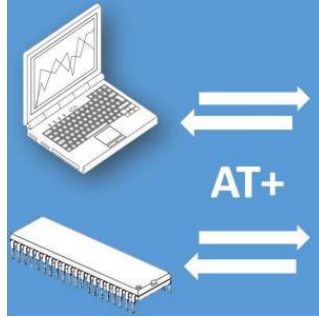


Figura 7 Comandos AT

Socket.IO: Se incluyó para habilitar la transmisión de datos en tiempo real entre el backend y el frontend. Esta tecnología permitirá una actualización constante y sin interrupciones de la información en la interfaz de usuario, es fundamental para notificar de manera inmediata ante cualquier anomalía en la calidad del agua. (Socket.IO: Implementing Real-Time Communication in Node.js | by WebClues Infotech | Sep, 2024 | JavaScript in Plain English, 2024)



Figura 8 Nodejs Socket.io

TinyGSM: Necesaria ya que es una biblioteca de Arduino diseñada para facilitar la comunicación con módulos GPRS, LTE y WiFi que utilizan interfaces de comandos AT. TinyGSM es especialmente útil para proyectos de IoT que requieren conectividad móvil. (TinyGSM/Src/TinyGsmGPS.Tpp at Máster · Vshymanskyy/TinyGSM, 2024)

PubClient: Es una biblioteca es parte de la estructura de cliente MQTT de Arduino y se utiliza para la publicación de mensajes en un bróker MQTT va de la mano con TinyGSM. (GitHub - knolleary/pubsubclient: A client library for the Arduino Ethernet Shield that provides support for MQTT., 2020)

ArduinoJson: Esta biblioteca se utiliza para serializar y de serializar datos JSON en proyectos de Arduino, facilitando la comunicación con APIs web. (Blanchon, 2014/2024)



Figura 9 ArduinoJson

Para desarrollar este proyecto de monitoreo de calidad del agua, es crucial contar con conocimientos en programación y manejo de datos, especialmente en la visualización con Grafana, que facilita la interpretación de información en tiempo real. Además, es indispensable comprender conceptos básicos de IoT y el funcionamiento de los dispositivos de sensores, porque estos dispositivos serán la fuente primaria de datos sobre los parámetros de la calidad del agua. La gestión de bases de datos también juega un rol importante, particularmente para almacenar y analizar datos de series temporales de manera eficiente y la conexión con la plataforma de visualización de datos y alertas.

Como desarrollador e Ingeniero Informático, este proyecto representa un desafío único que me brinda la oportunidad de contribuir significativamente a la mejora de la salud pública y la protección del medio ambiente. Me permitirá aplicar mis conocimientos técnicos para desarrollar un software innovador y eficaz. Además, trabajar en este entorno me proporcionará una valiosa experiencia, ampliando mis habilidades en programación y en la ejecución de soluciones prácticas para problemas reales.

III. Objetivos

Objetivo General

Desarrollar una aplicación web para la visualización de alertas tempranas relacionadas con la calidad del agua, para toma rápida de decisiones empleando Grafana, en proyecto desarrollo de un prototipo para la medición y alertas tempranas de parámetros fisicoquímicos del agua por telemetría para acuerdos rurales SENA 2024

Objetivos Específicos

1. Analizar el contexto y los requerimientos del proyecto IoT, identificando los tipos de datos generados por los sensores y su relevancia para la integración en el sistema web de monitoreo
2. Construir los módulos del sistema para la visualización en tiempo real de los datos de calidad del agua generados por los sensores de pH, turbidez, concentración, saturación de oxígeno y conductividad integrando estos datos con *Grafana*.
3. Evaluar la efectividad del sistema web en la toma de decisiones, midiendo su impacto en la detección temprana de anomalías y en la gestión de alertas sobre

la calidad del agua.

IV. Metodología

Los usuarios interactúan con los diferentes objetivos propuestos en el trabajo se establece que la implementación de la metodología UP puede ayudar a proyectos que involucren la interacción entre hardware y software al permitir una planificación y diseños claros de la arquitectura del modelo, entregas incrementales de software y hardware, y un enfoque de desarrollo guiado por casos de uso que permita una comprensión clara de cómo los usuarios interactúan con el software; esto ayuda a mejorar la calidad y la eficiencia del desarrollo y reduce el riesgo de problemas de interacción entre hardware y software en el futuro.

Fase 1: Inicio

En esta fase se definirán los objetivos del proyecto, los requisitos del usuario y los criterios de éxito. También se identificarán los riesgos del proyecto y se elaborará un plan preliminar para abordarlos.

Fase 2: Elaboración

En esta fase se realizará un análisis detallado de los requisitos y se definirá la arquitectura del software. También se elaborará un plan detallado para el proyecto y se definirá un cronograma de trabajo.

Fase 3: Construcción

En esta fase se llevará a cabo la implementación del software y se realizarán pruebas unitarias para garantizar su correcto funcionamiento. También se llevarán a cabo pruebas de integración para asegurarse de que todas las partes del software funcionen correctamente juntas.

Fase 4: Transición

En esta fase se realizará la entrega del software y se llevarán a cabo las pruebas de aceptación. También se realizará una capacitación sobre el uso del software, así como su respectiva documentación de este.



Figura 10 Fases de la Metodología

V. Estado del Arte

Actualmente el manejo en el monitoreo de la calidad del agua se ha vuelto una prioridad fundamental dado a su efecto en la salud pública y su entorno ambiental por esto el desarrollo de aplicaciones web y software de observación se han vuelto cruciales para brindar una solución efectiva en el cumplimiento continuo y así realizar más eficientemente la toma de decisiones basadas en datos precisos.

El presente estado del arte se basa en la implementación de una aplicación web dedicada a la visualización de alertas tempranas sobre las variables de calidad del agua a partir de sensores vernier especializados.

El proyecto de grado abordado por Gabriela Eliza Mendoza Muñoz titulado *Desarrollo de backend para sistema de medición automática de calidad de agua, adecuado a sensores de bajo costo* el cual se basa en el Desarrollo de backend para sistema de medición automática de calidad del agua, adecuado a sensores de bajo costo en el cual implementaron una arquitectura necesaria para el procesamiento y almacenamiento de los datos tuvo los siguientes resultados (*Desarrollo de backend para sistema de medición automática de calidad de agua, adecuado a sensores de bajo costo*, 2021)

- La base de datos InfluxDB permite almacenar un gran volumen de datos
- históricos, indexados temporalmente e identificables por cliente, sector y estación.
- La API soporta hasta 100 nodos concurrentes, con las configuraciones por defecto de Apache.
- El algoritmo descompositivo entrega mejores resultados en el benchmarking comparativo, comprobando su utilidad.

Tras la finalización de este proyecto se pudo percibir unos avances significativos, al implementar una arquitectura robusta para el procesamiento y almacenamiento de los datos mostrando ser efectiva, con unos resultados muy positivos en la gestión y el análisis de la calidad del agua. En esto también la API que se desarrolló demostró un rendimiento sólido soportando hasta 100 nodos concurrentes bajo las configuraciones predeterminadas de Apache.

El algoritmo igualmente ha sido validado con éxito evidenciándoles su utilidad en la optimización del procesamiento de datos. El proyecto no solo abordó la necesidad de una solución efectiva para la monitorización continua de la calidad del agua, sino que también sienta las bases para futuras mejoras y adaptaciones como podría ser la implementación de Machine Learning. Debemos agregar que en el ámbito de la gestión de recursos hídricos y la protección ambiental como se mencionaba anteriormente, la

necesidad de monitoreo en tiempo real de parámetros como la profundidad y la velocidad del agua son igualmente esenciales y se ha convertido también en una prioridad fundamental.

La razón es que la implementación de softwares avanzados de monitoreo basados en tecnologías de Internet de las Cosas (IoT) han demostrado ser soluciones efectivas para proporcionar datos precisos y actualizados que ayuden y faciliten la toma de decisiones en la gestión de ríos y cuencas fluviales.

Un claro ejemplo relevante de esto es el proyecto realizado por Ahmad, et al. titulado *Development of Visual Dashboard for River Monitoring System* en este proyecto los autores se centraron en el desarrollo de un sistema de monitoreo remoto en tiempo real para medir la profundidad y la velocidad del agua en ríos utilizando sensores ultrasónicos y de flujo de agua, así como la integración de estos datos en una interfaz gráfica de usuario (GUI) concluyendo lo siguiente

“El sistema utiliza sensores ultrasónicos y de flujo de agua para medir la velocidad y la profundidad del río. Los datos se transmiten al software de back-end, incluyendo Arduino IDE y Node-RED, para la recolección de datos. Los datos se almacenan en la base de datos en la nube InfluxDB para su uso futuro. La plataforma Grafana puede producir una interfaz gráfica de usuario interactiva (GUI), y el administrador puede modificar el panel visual utilizando su código de programa o las características existentes. Posteriormente, se puede hacer disponible al público en general como un observador del panel visual en línea a través del Uniform Resource Locator.” (Ahmad et al., 2022)

Además de estas funcionalidades, el software puede ser expandido para monitorear parámetros adicionales de calidad del agua, como pH y turbidez, lo que lo convertiría en una herramienta valiosa para la alerta temprana. La visualización en Grafana permite que los usuarios detecten problemas a tiempo y tomen medidas preventivas rápidamente. La capacidad de personalizar alertas y analizar los datos históricos facilita la gestión efectiva de los recursos hídricos y la protección del medio ambiente

Otro trabajo relevante en investigación sobre la utilización de IoT Y así como plataformas tecnológicas que dieron resultados es el trabajo de investigación realizado por Halim Fathoni et. Al., titulado “A Monitoring System of Water Quality Tunghai Lake Using LoRaWAN” ,los responsables se centraron en el desarrollo de un sistema de monitoreo de la calidad del agua utilizando tecnologías IoT y LoRaWAN. El software propuesto permite la transmisión de datos desde sensores de pH, oxígeno disuelto, temperatura y conductividad a través de Lora, los cuales son procesados y visualizados en la

plataforma Grafana. De acuerdo con los resultados obtenidos, el software demuestra una alta efectividad para el monitoreo remoto. Los autores indican que

“Este sistema utiliza la tecnología de comunicación Lora para hacer el monitoreo más conveniente y efectivo a través de distancias largas. Se envían los datos de sensores de pH, oxígeno disuelto, conductividad y temperatura, y son visualizados en la potente plataforma Grafana, que presenta la información en un tablero visual interactivo y fácil de usar. Esto permite a los administradores obtener de inmediato toda la información y configurar alarmas para notificar en momentos de anomalías, lo que optimiza el monitoreo y reduce la pérdida de datos.” (Fathoni et al., 2020)

Además, las investigaciones recientes siguen mostrando la importancia de implementar tecnologías avanzadas para el monitoreo en tiempo real de la calidad del agua, optimizando la toma de decisiones y mejorando la respuesta ante posibles amenazas al medio ambiente.

Un estudio que también aborda este desafío es el trabajo realizado por Juan Andrés Avelino Ospina diseño e implementación de la plataforma de monitoreo para el laboratorio Kiosol, (Avelino Ospina, 2023) donde se desarrolló una plataforma IoT para el monitoreo de la calidad del agua en sistemas de energía renovable. La plataforma permite la recolección de datos en tiempo real desde sensores de bajo costo, integrando una base de datos eficiente para almacenar la información y ofreciendo visualizaciones en la plataforma Grafana. En el proyecto, se destaca en el uso de TimescaleDB y MQTT para optimizar la transmisión y almacenamiento de datos, asegurando la visualización de datos en tiempo real y la configuración de alertas ante posibles anomalías en el software.

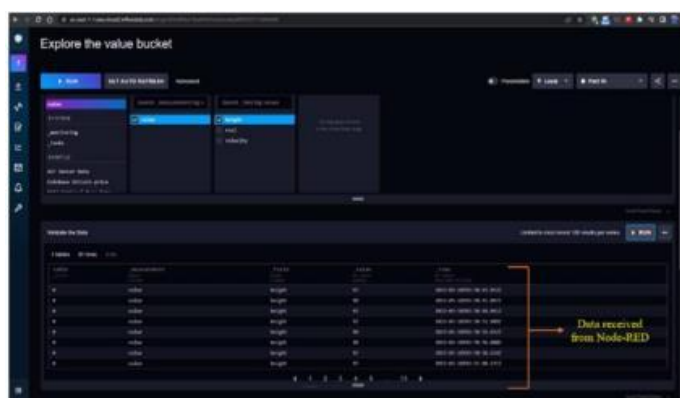


Figura 11 Data Received From Node-Red

Este enfoque de Avelino Ospina resalta la relevancia de las plataformas de visualización como Grafana para monitorear variables críticas en entornos energéticos.

Grafana es una plataforma open-source de visualización de datos interactiva

desarrollada por Grafana Labs. Permite a los usuarios ver sus datos por medio de tablas y gráficos unificados en tableros. Además, Grafana permite realizar consultas y configurar alertas sobre las métricas almacenadas. Esta herramienta facilita el análisis de datos, la identificación de tendencias y la detección de inconsistencias, reduciendo el tiempo de desarrollo al integrar con TimescaleDB, proporcionando un entorno robusto para la visualización de datos en tiempo real” (Avelino Ospina, 2023)



Figura 12 Dashboard or Remote River Monitoring System

Existen varias aplicaciones y plataformas de software que cumplen funciones similares centradas en el monitoreo de la calidad del agua, visualización de datos en tiempo real y alertas tempranas como lo son:

AquaWatch



Figura 13 AquaWatch

Ofrece un sistema de monitoreo continuo y remoto de la calidad del agua, utilizando tecnologías avanzadas como sensores y cámaras AI. Permite la recolección de datos en tiempo real de parámetros como oxígeno disuelto, turbidez, temperatura, pH se considera una solución de IoT algunas características claves son:

Tabla 2 Descripción de las Funcionalidades y Capacidades del Sistema AquaWatch

#	Funcionalidad	Descripción
1	Cámara AquaWatch AI	<ul style="list-style-type: none"> Recoge datos visuales y analiza patrones climáticos y ambientales. Detecta anomalías y genera resultados

		<p>procesables en tiempo real.</p> <ul style="list-style-type: none"> • Su capacidad para ejecutar múltiples programas simultáneamente la convierte en una herramienta poderosa.
2	Conjunto flotante de sensores	<ul style="list-style-type: none"> • Mide parámetros importantes como oxígeno disuelto, turbidez, temperatura, pH, amonio y conductividad. • Estos sensores ofrecen datos en tiempo real, lo que permite gestionar la calidad del agua de manera más efectiva, proteger ecosistemas acuáticos y mejorar la salud humana.
3	Inteligencia Visual Ambiental	<ul style="list-style-type: none"> • Utiliza la cámara AW-108 para capturar imágenes de alta resolución y procesarlas en el dispositivo. • Se entrena con modelos de aprendizaje automático para identificar problemas ambientales de manera temprana.
4	Puntuación de salud del agua	<ul style="list-style-type: none"> • Monitorea continuamente las vías fluviales y ofrece una visión completa de la calidad del agua, más allá de los análisis químicos tradicionales, evaluando su capacidad para sustentar ecosistemas saludables.
5	Gestión Predictiva del Agua	<ul style="list-style-type: none"> • AquaWatch ofrece a sectores como la agricultura, la construcción y las autoridades del agua una plataforma de monitoreo proactivo que integra datos satelitales y sensores para realizar planes a largo plazo en la protección de recursos hídricos.

HidroVu



Figura 14 HidroVu

Es una plataforma en línea para la monitorización remota de la calidad del agua. HidroVu combina elementos de IoT (Internet de las cosas), al recopilar datos a través de sensores y enviarlos a la nube, con una plataforma de Dashboard que facilita la visualización y gestión de esos datos. También se puede decir que es una plataforma de gestión de datos que proporciona una solución integral para el monitoreo remoto. Algunas de sus características son:

Tabla 3 Características Clave de HidroVu en la Gestión y Visualización de Datos

#	Funcionalidad	Descripción
1	Solución Remota	<ul style="list-style-type: none"> • HidroVu recopila datos de telemetría y los carga en la nube,

	Completa	permitiendo la visualización de estos datos a través de gráficos personalizados. Reduce la necesidad de visitas costosas al sitio gracias a la capacidad de configurar y modificar instrumentos de forma remota.
2	Visibilidad en Toda la Red	<ul style="list-style-type: none"> Administra fácilmente redes con pocos o miles de instrumentos, permitiendo entender relaciones entre parámetros y ubicaciones a través de gráficos instantáneos y una interfaz visual intuitiva.
3	Paneles de Control	<ul style="list-style-type: none"> Los paneles personalizables permiten la visualización de parámetros críticos y alarmas configuradas. Son accesibles tanto desde computadoras de escritorio como dispositivos móviles.
4	Filtros de Macros	<ul style="list-style-type: none"> Ayudan a visualizar y filtrar grandes conjuntos de datos rápidamente, permitiendo exportar solo los datos necesarios.
5	Alarmas de Múltiples Condiciones	<ul style="list-style-type: none"> Las alarmas se activan cuando los parámetros cruzan umbrales configurados por el usuario. Notificaciones por correo electrónico, SMS o llamadas automatizadas informan sobre posibles problemas en tiempo real.
6	Copia de Seguridad y Conexión a la Nube	<ul style="list-style-type: none"> El sistema mantiene un respaldo seguro de todos los datos en la nube, protegiendo los registros críticos y facilitando el acceso a ellos en todo momento.

(Servicios de Datos HydroVu | Software de Gestión de Datos de Agua | In-Situ, 2024)

KISTERS WISKI:



Figura 15 Kisters Wiski

Es una solución integral para la gestión de datos hidrológicos que ha sido ampliada para incluir información detallada sobre la calidad del agua. Se considera tanto una plataforma de gestión de datos como una solución IoT debido a su capacidad para integrar datos de sensores y dispositivos de medición en tiempo real, lo que permite la automatización y el monitoreo de redes de medición. Este sistema es especialmente útil para instituciones públicas, asociaciones del agua y centros de investigación, por lo que permite gestionar tanto los parámetros clásicos de los datos hidrológicos como los parámetros químicos y físicos relacionados con la calidad del agua. (*wiski-overview-en.pdf*, s. f.)

Tabla 4 Descripción de las Funcionalidades del Sistema WISKI para Análisis y Gestión de Datos Ambientales

#	Funcionalidad	Descripción
1	Procesamiento de datos clásicos y muestras	<ul style="list-style-type: none">WISKI puede procesar datos hidrológicos continuos (nivel del agua, velocidad de flujo, temperatura, etc.) y combinarlos con análisis de calidad del agua (oxígeno disuelto, bacterias, pH, contaminantes), permitiendo una evaluación conjunta.
2	Asignación automática de datos	<ul style="list-style-type: none">El sistema asigna automáticamente los datos químicos y físicos obtenidos en el laboratorio a las mediciones hidrológicas, permitiendo validar y correlacionar estos datos de forma eficiente, incluso cuando hay un retraso en la obtención de resultados de muestras.
3	Visión holística y gestión eficiente	<ul style="list-style-type: none">Ofrece a diferentes departamentos una plataforma de base de datos común para gestionar los datos tanto de hidrología como de calidad del agua de forma integrada, lo que facilita el análisis y la toma de decisiones.
4	Módulo de calidad del agua KiWQM	<ul style="list-style-type: none">Este módulo específico permite una visión detallada de los aspectos cuantitativos y cualitativos de los cuerpos de agua, además de calcular indicadores derivados como las cargas de contaminantes.
5	Expansión con el componente KIECO	<ul style="list-style-type: none">El software puede expandirse con datos biológicos obtenidos de estudios de campo, permitiendo una evaluación ecológica más completa.

(Environmental Data Management System | WISKI, 2024)

Comparativa de Soluciones de Monitoreo de Calidad del Agua

Tabla 5 Comparativa de Soluciones de Monitoreo de Calidad del Agua

APP	SENSORES	VISUALIZACIÓN	OBJETIVO
AQUAWATCH	Sensor de oxígeno disuelto. Sensor de turbidez. Sensor de temperatura. Sensor de pH. Sensor de amonio. Sensor de Conductividad.	Dashboard	Ofrecer una solución inteligente y fácil de usar para monitorear la calidad del agua en tiempo real. Este dispositivo emplea tecnología de sensores que mide indicadores clave de la salud del agua, como el oxígeno disuelto, la turbidez, la temperatura, el pH y la conductividad. Estos datos permiten tomar decisiones informadas sobre la gestión del agua, proteger los ecosistemas acuáticos y mejorar la salud humana.

HIDROVO	sensores infrarrojos SoliTech. sensores de turbidez y sólidos suspendidos TurbiTech. Monitor portátil 750. Sensor de turbidez TurbiTech D-ISO. Sensor de pH y ORP de WaterTech. Sonda multiparamétrica MPX4. Sensor RDO OxyTech Para 750. Sensor WaterTechFCL. Sensor WaterTech C4E.	Dashboard Filtros de macros	Proporcionar una plataforma de gestión de datos en tiempo real que facilite la supervisión remota y el análisis de la calidad del agua. A través de su interfaz web, permite a los usuarios acceder a datos de sensores desplegados en múltiples ubicaciones, almacenarlos de forma segura en la nube y configurar alarmas personalizadas. Esto reduce la necesidad de visitas de campo, mejora la toma de decisiones al proporcionar datos precisos y a tiempo, y optimiza la gestión de redes de monitoreo de calidad del agua.
KISTERS WHISKY	Sensor radar de nivel de agua. Sensor radar combinado de velocidad y nivel RG 30+C. HS40AFC/3100 Compacto. WL3100/WL3100A Transductor de presión en seco. Codificador de eje absoluto AD375MA.	Dashboard s interactivos. Gráficos personalizables.	Proporcionar una plataforma integral para la gestión, análisis y visualización de datos ambientales, especialmente centrada en la calidad del agua, hidrología, y otros recursos hídricos. WISKI permite a las organizaciones automatizar procesos complejos, validar y controlar la calidad de los datos, y generar informes y gráficos personalizados. Está diseñado para facilitar la toma de decisiones informadas en tiempo real a través de la integración de datos de diferentes sensores y fuentes.
GRAFANA	Sensor de pH, Sensor de turbidez, Sensor de oxígeno concentrado, Sensor de conductividad.	Dashboard s interactivos, Gráficos personalizables	Ofrecer una plataforma para observar y ver datos al instante, capaz de combinar diferentes sensores como pH, turbidez, oxígeno disuelto y conductividad. Este emplea gráficos interactivos que ayudan a examinar la calidad del agua, mejorando la toma de decisiones y la administración del agua mediante interfaces visuales fáciles de comprender.

A continuación, se visualizarán las respectivas herramientas que se utilizaron para dar solución al respectivo problema del proyecto, la selección de estos dispositivos por los respectivos dueños y encargados de esta actividad, cabe mencionar que estos dispositivos han sido seleccionados por la suficiencia para poder cumplir con los objetivos planteados para la captura, almacenamiento, procesamiento y visualización de datos en tiempo real; estos elementos están conformados por sensorpH, sensor de

turbidez, sensor de oxígeno disuelto, como herramientas de software, Arduino vernier Grafana, MongoDB.



Figura 16 Sensores Vernier

Sensor pH: Este sensor se utiliza para poder medir los niveles de acidez y alcalinidad del agua, esto ayuda a la proporcionalidad de datos esenciales para poder determinar la calidad de los recursos hídricos. Este medidor de pH determina la variación del potencial eléctrico entre un electrodo de referencia. (*pH Sensor User Manual, 2024a*)



Figura 17 Sensor pH Vernier

Sensor de Turbidez: Su labor es medir la turbidez del agua lo que ayuda a detectar las partículas u opacidad del agua, lo que permite detectar partículas suspendidas que afectan su transparencia. (*Turbidity Sensor User Manual, 2024a*)



Figura 18 Sensor Turbidez Vernier

Sensor de Conductividad:

Su principal función es calcular el número de oxígeno disuelto que está presente en el agua, se podría decir que cuantifica la densidad en el agua, mencionando que es un parámetro fundamental para evaluar la calidad del agua, en la salud del ecosistema acuático. (*Conductivity Probe User Manual, 2024a*)



Figura 19 Sensor Conductividad Vernier

Sensor oxígeno saturación:

Mide la saturación de oxígeno, indica la cantidad de oxígeno que está presente en el agua comparado con la cantidad máxima permitida que puede disolver a una temperatura y a una presión determinada, La saturación de oxígeno es una de las variables de la calidad del agua ,esto porque está directamente asociada con la cantidad de oxígeno disuelto en el agua , aunque más exactamente se refiere a saturación, es uno de los parámetros fundamentales para entender el estado del ecosistema acuático. (*Vernier Optical DO Probe User Manual, 2024a*)



Figura 20 Sensor de Oxígeno Vernier

Arduino vernier shield interface:

Esta placa es la encargada de conectar los sensores vernier a la placa Arduino, permitiéndole a los usuarios poder integrar diversos sensores como se mencionaba anteriormente algunos de ellos son los que hemos especificado anteriormente pg, turbidez, oxígeno, saturación entre otros más, esta se encarga de la recopilación de datos de una manera sencilla y también de forma eficiente, permitiendo la comunicación

con los sensores y con el Arduino esto para realizar las mediciones y análisis en proyecto ya sean educativos o basado en experimentos científicos , en nuestro caso está implementado para la el análisis de las variables de la calidad del agua. (*Vernier Coding Activities with Arduino®: Analog Sensors - Vernier, 2024*)



Figura 21 Arduino Vernier

Selección de dispositivos

“Ayude a los estudiantes a desarrollar confianza en sus habilidades de codificación y fomente el pensamiento crítico creativo a través de proyectos Arduino con sensores Vernier LabQuest. “
(*Vernier Arduino™ Interface Shield | Vernier Canadá, 2024*)

Beneficios:

Algunos de los beneficios de usar esta placa son los siguientes: “

- *El uso de sensores LabQuest con Arduino es una solución de aprendizaje remoto rentable.*
 - *Integrar codificación, ingeniería, matemáticas y ciencias en su plan de estudios es más fácil con la tecnología Arduino y Vernier.*
 - *Los estudiantes comprenden mejor el impacto que tiene la codificación en el mundo real.*
 - *Codificar proyectos Arduino que interactúan con sensores LabQuest ayuda a sus estudiantes a perfeccionar habilidades cruciales como la resolución de problemas y la colaboración.*
- Las interesantes actividades ayudan a sus estudiantes a aprender los conceptos básicos de la codificación de Arduino con sensores LabQuest. “*(*Arduino, 2024; Connecting Vernier Sensors to Arduino, 2023*)

CAPÍTULO 1. LEVANTAMIENTO DE REQUERIMIENTOS

1. Reunión de planeación

La reunión de planeación constituyó en el primer paso en el desarrollo de software para monitoreo de calidad del agua, en este encuentro participó la compañera de electrónica Rosa Esneda García Quintero, cabe recalcar que se llevó una integración amigable y completa en el que se discutió de los requerimientos del módulo del usuario también sobre las tecnologías que se podrían usar para dicho desarrollo, igualmente se habló de la especificación del tiempo que podría tardar el desarrollo de dicho software, así como se habló de varias integraciones para el almacenamiento y definición de la base de datos, al mismo tiempo se llegó a la definición del lenguaje de programación más adecuado que se podría usar para la implementación del desarrollo, es importante recalcar que este sistema estará integrado en un aplicativo web.

1.1 Requisitos funcionales y no funcionales

Los requisitos funcionales del aplicativo web para el software de monitoreo de calidad del agua se han definido teniendo en cuenta las necesidades específicas del proyecto y los objetivos que se han planteado.

Estos requisitos nos muestran algunas de las funcionalidades claves que se deben de cumplir en el sistema para que se pueda garantizar un correcto funcionamiento, desempeño y también la utilidad esperada en el entorno donde se debe de realizar el monitoreo de la calidad del agua en base a las variables de la calidad del agua.

Algunos de los requisitos funcionales principales que se definieron son:

1. Visualización de datos en tiempo real
2. *Inicio y control de la sesión del usuario.*
3. *Cerrar sesión.*
4. *Generación de Alertas Automáticas.*
5. *Recuperar contraseña.*
6. *Registro de usuario.*
7. *Almacenamiento de datos.*
8. *Cambiar contraseña dentro del aplicativo web.*
9. *Gestionar lista de usuarios registrados.*
10. *Modificar Datos de la Cuenta de Usuario Logueado.*
11. *Seleccionar Sensores para Recepción y escucha del socket de Datos en Tiempo Real.*
12. *Mostrar Datos Críticos que Superaron el Umbral en Tiempo Real.*
13. *Dashboard Dinámico con Indicadores de Umbral.*
14. *Generación de Notificaciones por Superación de Umbral.*

No Funcionales

En el desarrollo del sistema de monitoreo de la calidad del agua, aparte de los requisitos funcionales, también establecieron algunos los cuales estos garantizan el correcto

funcionamiento ,la usabilidad. Estos requisitos son fundamentales para poder influir en la experiencia del usuario y el desempeño del sistema Entre ellos se encuentran:

Rendimiento:

- **Tiempo de respuesta:** Se debe procesar y también mostrar en tiempo real en el que el tiempo de respuesta no puede ser mayor a 2 segundos.
- **Seguridad:**
- **Cifrado de datos:** Las contraseñas definidas por los usuarios, se deben almacenar, pero también deben ser transmitidas de manera segura utilizando algoritmos de cifrado.
- **Acceso controlado:** No se puede permitir el acceso a ninguna funcionalidad sin que el usuario este correctamente Logueado.
- **Disponibilidad del sistema:** El sistema debe estar disponible la mayor parte de tiempo

Tabla 6 Requisitos Funcionales de los Módulos del Sistema

Módulo	Requisitos funcionales
Módulo de usuario	Inicio y control de la sesión del usuario.
	Cerrar sesión
	Recuperar contraseña
	Registro de usuarios
	Cambiar contraseña dentro del aplicativo web
	Gestionar lista de usuarios registrados
	Modificar datos de la cuenta del usuario logueado
Módulo de datos en tiempo real	Visualización de datos en tiempo real
	Seleccionar sensores para recepción y escucha del socket de datos en tiempo real.
	Mostrar Datos críticos que superan el umbral en tiempo real.
Módulo de alertas	Generación de alertas automáticas
	Generación de notificaciones por superación del umbral.
Módulo de almacenamiento	Almacenamiento de datos.
Módulo del Dashboard	Dashboard dinámico con indicadores del umbral.

1.2. Especificaciones técnicas definidas

Las especificaciones técnicas del aplicativo web de visualización de datos, que serán mostrados en alertas tempranas de variables de calidad del agua se establecieron para garantizar un desarrollo eficiente y un rendimiento óptimo, de la aplicación web; en esta parte se definieron aspectos importantes que abarcan componentes claves del hardware y software al igual que la infraestructura tecnológica, las cuales son necesarias para poder cumplir con los objetivos del proyecto

Hardware

1. Sensores vernier los cuales serán usados para medir las variables de calidad del agua los cuales son: pH, Turbidez, Concentración y saturación de oxígeno y conductividad.
2. Módulo SIM7600X HAT, su función es transmitir los datos mediante GPRS en zonas de baja conectividad a redes wifi.
3. Placa Arduino vernier, esta será para la gestión y conexión de los sensores.

Servidores

1. El servidor en el cual se almacena la API es necesario para un correcto proceso de gestión en el almacenamiento en la base de datos.

Almacenamiento

1. Definición de espacio para almacenamiento, será importante para manejar datos en tiempo real y el mantenimiento de datos históricos para visualización en el Dashboard.

Software Frontend

1. Lenguajes y herramientas React, HTML, CSS, JS, los cuales serán vitales para el desarrollo de interfaces de usuario accesible y adaptativa.
2. Implementación de la aplicación de visualización gráfica de datos Grafana, importantes para la creación de un Dashboard interactivo.

Software Backend

1. El entorno de desarrollo Node.js con el framework express para la realización de la API y la comunicación con el frontend.
2. La transmisión de datos en tiempo real mediante el uso de Socket.io, esto con el fin de la actualización persistente de los datos en el Dashboard sin la exigencia de recargar la página.
3. La integración con la base de datos con MONGODB que es una base de datos NOSQL que almacenará y permitirá la consulta de los datos enviados al dispositivo IOT.

Alertas

1. El envío de notificaciones en tiempo real a través de la plataforma web, esto para mantener informado al usuario de todos los cambios pertinentes.

Seguridad

1. Fundamental el uso de JWT - JSON Web Tokens para la seguridad y protección en el acceso al sistema

Infraestructura

1. Conectividad en el uso del estándar de comunicación GSM/GPRS, un puente para una óptima transmisión de datos desde el dispositivo IOT en zonas remotas.
2. Reiteración en la transmisión de datos desde los sensores, esto con el fin de impedir la pérdida de información.
3. Configuración con el Dashboard interactivo en Grafana para visualización en tiempo real.

4. Uso de plugin librerías y APIS para una mejor integración entre datos, almacenamiento y el Dashboard.
5. Implementación de un diseño responsivo que permite el acceso desde dispositivos de diferentes tamaños computador de escritorio, tabletas y móviles.
6. Disponibilidad para los usuarios asegurando que puedan acceder a la aplicación y realizar las respectivas actividades sin interrupciones significativas.

Especificar técnicas definidas para la aplicación móvil del software de variables de calidad del agua

Tabla 7 Especificaciones Técnicas para la Aplicación web del Software Calidad del Agua

Especificar técnicas definidas para la aplicación web del software de variables de calidad del agua	
Lenguaje de programación	Se ha escogido el lenguaje de programación JavaScript para el desarrollo de la aplicación web.
Base de datos	Se escoge MongoDB como base de datos ya que es NOSQL.
Dispositivo	Inicialmente se decidió que el software sería un sistema web pero adaptable a diferentes tamaños de dispositivos siendo responsive.
Seguridad	Se decidió que la aplicación web debe tener una seguridad de autenticación jwt, esto con el fin de conservar la seguridad del sistema.
Fiabilidad	La aplicación web del sistema está diseñado para ofrecer una experiencia accesible y confiable para la toma rápida de decisiones
Disponibilidad	La aplicación web del sistema mantendrá una alta disponibilidad para el recibimiento de los datos y para los usuarios, asegurando que puedan acceder a la aplicación y realizar las respectivas actividades

Algo fundamental en el desarrollo de la aplicación web del software definir estas especificaciones técnicas, ya que establece los requisitos detallados que el sistema debe cumplir; estas especificaciones proporcionan una guía clara no solo para el equipo de desarrollo sino también para las partes interesadas garantizando que todos los involucrados tengan una comprensión común de cómo debe funcionar y que debe incluir el producto final al momento del funcionamiento.

Se busca establecer un modelo que sea sostenible, pero a su vez escalable, para que pueda ser implementado en diferentes comunidades que permitan una gestión más segura y preparativa de la calidad del agua.

1.3. Funcionalidad del producto

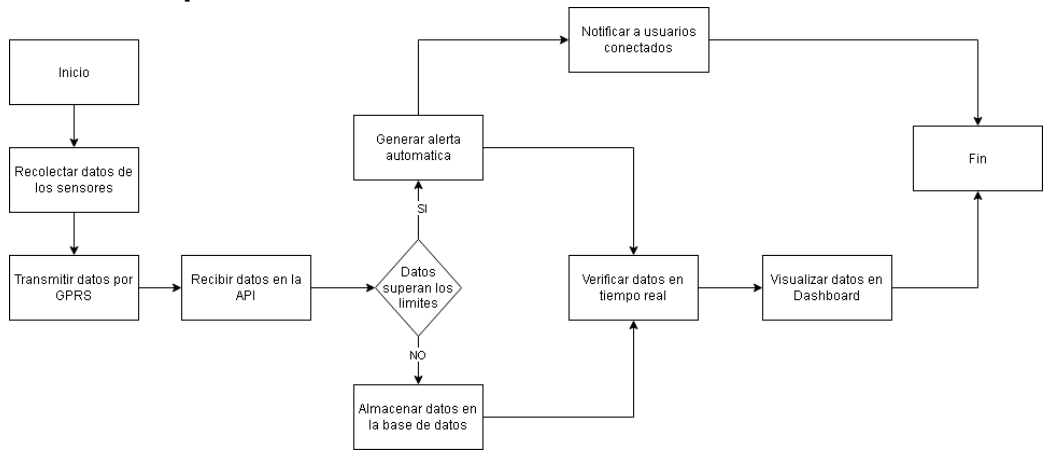


Figura 22 Diagrama de Flujo de Proceso

Cómo es posible evidencia en el diagrama de flujo que explica las interacciones principales de la funcionalidad del aplicativo web de monitoreo de calidad del agua que describe de una manera continua como se recolectan, transmiten y se procesan los datos provenientes del sistema IoT en base a cada sensor, cada fase muestra las interacciones que son claves en cómo estos contribuyen al flujo general.

El flujo aquí comienza con la recolección de los datos como pH, turbidez, oxígeno de concentración entre otros, mediante sensores IoT especializados, estos datos son transmitidos a través del estándar de datos GPRS a la respectiva API del sistema donde estos se encargan de procesarlos y almacenarlos en una base de datos centralizada para el almacenamiento continuo de los datos que están llegando; luego estos datos son transmitidos para generar gráficos interactivos en el Dashboard de Grafana pero también, para la evaluación que si los datos que llegan superan los valores establecido por las autoridades competentes de sanidad del agua, entonces se generan alertas en tiempo real y se le notifican al usuario para una rápida toma de decisión.

Cabe mencionar que este flujo refleja la interacción del usuario quien es el encargado de poder visualizar los datos en tiempo real y recibir las notificaciones de alerta.

Tabla 8 Tabla de Funcionalidades

TABLA DE FUNCIONALIDADES		
Funcionalidad	Descripción	Tecnología utilizada
Recibimiento de los datos	Detención y transmisión de datos en tiempo real desde el hardware IoT con los sensores hasta el backend del aplicativo web.	Sensores Vernier como pH, turbidez, oxígeno de saturación, etc. Modulo SIM7600X 4G HAT

Almacenamiento de datos	Almacenar los datos, en una base de datos para el posterior análisis y su visualización para la toma de decisiones en base que vayan llegando.	MongoDB
Visualización en Dashboard Grafana.	Mostrar los respectivos gráficos y la actualización de los datos en tiempo real en base a un tiempo estimado así como en tiempo real.	Grafana
Generación de alertas	Notificar al usuario con alertas en caso de que los datos superen el respectivo umbral y se muestre anomalías en los parámetros críticos.	JavaScript Socket.IO
Panel de control (Dashboard)	Interacción de la interfaz para la visualización de datos en tiempo real así como los datos con anomalías en respectiva tabla	HTML, CSS JavaScript
Operaciones fuera de línea	Permite guardar los datos en una pila y sincronizar datos automáticamente cuando la conexión se restablezca.	Arduino Modulo SIM7600X 4G HAT



Figura 23 Diagrama Caso de Uso

En este diagrama de casos de uso que ilustra las interacciones principales, y a continuación se detalla cada uno de los elementos presentes en el diagrama:

Actores

1. **Usuario:** Representa al individuo que interactúa con el sistema, llevando a cabo acciones como iniciar sesión, registrar una cuenta, y la visualización de datos y alertas en tiempo real.
2. **Iniciar Sesión:** Esta opción le permite al usuario que pueda ingresar

correctamente al sistema, la validación de credenciales está incluida dentro de inicio de sesión y registro de usuario, ya que siempre es necesario poder verificar que las credenciales ingresadas por el usuario en ambos procesos sean correctas.

Dentro del inicio de sesión se encuentra las opciones para el usuario entre ellas encontramos:

3. **Recuperar Contraseña:** Un subproceso que se extiende en la misma vista de iniciar sesión siendo una su opción del mismo y es porque permite a los usuarios recuperar la contraseña en caso de olvido, para que el proceso de recuperar la contraseña el aplicativo web le solicita al usuario el correo electrónico registrado con anterioridad, el cual se le envía un enlace único de verificación y una vez aceptado, le permite ingresar nuevamente una nueva contraseña, la petición del correo es con el fin de garantizar que solo el usuario dueño de la cuenta pueda poder recuperar el acceso a la misma.
4. **Gestionar Usuario:** Este es otro subproceso que forma parte del sistema tras *realizar el inicio de sesión* y se incluye porque permite administrar tanto la información del usuario que se encuentra dentro del sistema, es decir logueado, como la de los usuarios registrados dentro del aplicativo web.

Relaciones de Inclusión («include»)

- a) Ver lista de usuarios: Permite visualizar todos los usuarios registrados en el aplicativo web.
 - b) Editar usuarios: Opción para modificar la información de cualquier usuario listado.
 - c) Eliminar usuarios: Como el nombre indica, facilita eliminar usuarios del aplicativo web.
 - d) Editar usuario logueado: Si un usuario desea modificar su información como nombre edad y correo, mientras tenga la sesión iniciada, es importante esta funcionalidad para que este proceso sea posible.
 - e) Cambiar contraseña: Permite al usuario actualizar su contraseña dentro del sistema. Este subproceso es fundamental para mantener actualizados los datos de los usuarios y gestionar el acceso de manera eficiente.
-
5. **Seleccionar Sensores:** El sistema permite al o los usuarios seleccionar sensores, este es uno de los puntos más importantes dentro del proceso, ya que este significa que puede elegir de qué sensor desea recibir los datos en tiempo real para el posterior monitoreo y visualización. Esta opción incluye algunas funciones que son claves
 6. **Visualizar datos en tiempo real:** Componente donde el usuario muestra los

valores actualizados de los sensores en directo, permitiendo un monitoreo constante.

7. **Visualizar tabla de datos críticos:** Este permite organizar la información de las alertas, es decir los datos que se recibieron y superaron el respectivo umbral establecido en una respectiva tabla junto con su fecha y hora esto permite un mejor análisis y una comparación de los datos.
8. **Visualizar Dashboard:** El subproceso de visualización de datos dentro de seleccionar sensores esto le va a permitir al usuario a monitorear las variables de la calidad del agua en tiempo real, información proveniente de los sensores. Esto es muy importante para garantizar que se esté llevando a cabo una correcta transferencia de datos y así detectar posibles comportamientos adecuados o si se presenta alguna anomalía en la respectiva muestra: para poder facilitar la identificación de los estados de los datos se implementa un sistema de colores:
 - a) **Verde:** Estado normal los datos están dentro del rango permitido.
 - b) **Amarillo:** Estado de alerta posibles datos con irregularidades que requieren atención.
 - c) **Rojo:** Estado crítico datos recibidos que se salieron del rango permitido presentando anomalías y que requiere atención inmediata.

Este mecanismo ayuda a la rápida toma de decisiones, lo que permite asegurar un control eficiente del sistema IoT.

9. **Cerrar Sesión:** Estos casos de uso dependen de la autenticación del usuario, lo que implica que antes de realizar la visualización y alerta de datos, el usuario debe haber iniciado sesión en el sistema.
10. **Registrar Usuario:** Esta acción es la que le permite al usuario poder salir del sistema de forma segura. El usuario al cerrar sesión, se eliminan las credenciales de acceso almacenadas temporalmente, esto lo que permite es evitar que otra persona pueda usar la cuenta sin plena autorización.

Relaciones de Inclusión («include»)

11. **Validar Credenciales:** Se incluye porque es necesario verificar que el correo no esté registrado previamente y que la contraseña cumpla con los requisitos de seguridad así como que los campos no estén vacíos.
12. **Recuperar Contraseña:** Esta función ayuda ya que si el usuario por alguna razón olvida su contraseña, antes de volver a registrarse va a poder recuperar la

contraseña, la cual está totalmente relacionada con el inicio de sesión esto permite que los usuarios vuelvan a tener acceso a su cuenta sin tener que comprometer la seguridad del aplicativo web.

13. **Ver Notificaciones de alerta:** Permitir al usuario a visualizar las alertas las cuales son generadas en tiempo real por el aplicativo web lo cual está relacionado con datos críticos que superaron el respectivo umbral. Este caso de uso incluye algunas acciones las cuales son:

14. **Generar Alerta:** Es un subproceso incluido en "Ver notificaciones de alerta". Para que las notificaciones sean visibles, primero deben ser generadas por el aplicativo web con base en parámetros críticos detectados por los sensores.

15. **Limpiar notificaciones:** Es un subproceso que permite al usuario gestionar las notificaciones existentes, eliminando aquellas que ya no son relevantes o que han sido atendidas.

Ver Notificaciones: Es un subproceso incluido en Ver notificaciones de alerta, permite al usuario a visualizar las notificaciones de las alertas generadas las cuales se reciben en tiempo real, al momento que el aplicativo web detecta una anomalía en los datos se genera la alerta de guardan pa poderlas ver en la parte de notificaciones.

2. Casos de uso

Los casos de uso proporcionan una descripción detallada del flujo de un sistema desde la perspectiva del usuario, lo que permite visualizar cómo interactúan los usuarios con el software para cumplir sus objetivos. Además de facilitar la comprensión del flujo principal los casos de uso también identifican posibles subflujos alternativos y excepciones, ayudando a anticipar situaciones como errores del sistema o comportamientos inesperados, Esto no solo contribuye a un diseño más robusto m sino que también garantiza que el sistema responda adecuadamente a falencias o cualquier situación que pueda surgir durante su operación. Para poder entender un poco mejor el funcionamiento del módulo del usuario que se desarrollan en la aplicación web del software se elaboraron los siguientes casos de uso:

CASO DE USO: INICIO DE SESIÓN:

Tabla 9 Inicio de Sesión

Caso de uso: Inicio de sesión
Objetivo: Permite al usuario ingresar a la aplicación.
Actores: Usuario
Precondiciones: El usuario debe estar registrado en el sistema

<p>Secuencia normal:</p> <p>Este caso de uso comienza cuando el usuario requiere ingresar al aplicativo web e ingresar a la interfaz de inicio de sesión.</p> <p>El sistema muestra un formulario donde el usuario debe ingresar su correo y contraseña.</p> <p>El usuario ingresa su correo y contraseña en los campos correspondientes.</p> <p>El usuario hace clic en el botón "Iniciar sesión" para enviar sus credenciales al sistema.</p> <p>El sistema antes de permitir el ingreso valida que los campos no estén vacíos, que el correo esté en un formato correcto y que las credenciales estén almacenadas en la base de datos.</p> <p>Si las credenciales de correo y contraseña son correctas el sistema muestra una ventana con un mensaje de ingreso exitoso.</p> <p>El sistema redirige a la interfaz Dashboard de la aplicación web.</p> <p>Fin del caso de uso.</p>
<p>Sub-flujo:</p> <p>S-1: Usuario no registrado:</p> <p>Si el usuario quiere ingresar con un correo que no existe en la base de datos se le muestra una ventana con un mensaje de usuario no registrado.</p> <p>Si la contraseña que el usuario ingrese no es la correcta, se le muestra una ventana con un mensaje de correo y/o contraseña incorrecta.</p>
<p>Excepciones:</p> <p>E-1: Campos vacíos</p> <p>Indicar completar los campos requeridos al usuario.</p> <p>Se regresa al paso 2 de secuencia normal.</p> <p>E-2: Credenciales incorrectas</p> <p>El sistema muestra un mensaje de error indicando que los datos que ingresó en el formulario no son los correctos. Subflujo 2</p> <p>Se regresa al paso 2 de secuencia normal.</p>

CASO DE USO: REGISTRO DE USUARIO

Tabla 10 Registro de Usuario

Caso de uso: Registro de cuenta
Objetivo: Permitir al usuario registrarse en la aplicación
Actores: Usuario
<p>Precondiciones:</p> <p>El usuario debe ingresar a la interfaz de registro de usuario desde la opción Registrar cuenta</p>
<p>Secuencia normal:</p> <p>Este caso de uso comienza cuando el usuario requiere registrarse y da en la opción registrar cuenta en la interfaz de inicio de sesión.</p> <p>El usuario completa el formulario de registro ingresando sus datos (nombre, edad, correo, contraseña)</p> <p>El usuario da clic en el botón Registrar cuenta.</p> <p>El sistema valida que los datos sean correctos.</p> <p>Que el correo ingresado sea un formato permitido.</p> <p>Que todos los campos estén completos.</p> <p>Que el correo no esté registrado en la base de datos.</p> <p>El sistema registra al usuario en la base de datos y envía un mensaje de confirmación en pantalla</p> <p>El sistema redirige al usuario a la ventana de inicio de sesión</p> <p>Fin de caso de uso.</p>
Sub-flujo:
<p>Excepciones:</p> <p>E-1: Completar todos los campos.</p> <ol style="list-style-type: none"> 1. El sistema le indica al usuario que campos faltan por completar. 2. Se regresa al paso 2 de secuencia normal. <p>E-2: El correo ya ha sido registrado.</p> <ol style="list-style-type: none"> 1. El sistema muestra una ventana con un mensaje que indica que el correo ya se encuentra registrado

2. Se regresa al paso 2 de secuencia normal.
E-3: <u>Error de registro.</u>
1. El sistema muestra un mensaje de error mostrando al usuario que hubo inconvenientes con su registro.
2. Se regresa al paso 2 de secuencia normal.

CASO DE USO: EDITAR PERFIL DEL USUARIO

Tabla 11 Editar Perfil del Usuario

Caso de uso: Editar perfil de usuario
Objetivo: Permitir al usuario actualizar información de su cuenta información como: correo, nombre, edad.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión
Secuencia normal: Este caso de uso comienza cuando el usuario requiere actualizar la información de su cuenta. El usuario ingresa a la opción Editar usuario. El sistema le muestra la interfaz Editar usuario con los datos de su cuenta correo nombre, edad. El usuario modifica la información de su cuenta. El usuario da clic en la opción Guardar Cambios. El sistema valida que la nueva información modificada por el usuario esté correcta. El correo este en un formato correcto. Que todos los campos estén completos. Que el correo no este registrado en la base de datos. Si todo es correcto, el sistema le muestra una ventana para confirmar si realmente quiere modificar sus datos. El usuario confirma la modificación de los datos dando clic en Sí. El sistema le muestra un mensaje de Usuario editado correctamente. El sistema redirige al usuario a la interfaz de Dashboard. Fin del caso de uso.
Sub-flujo: S-1: <u>Ventana de confirmación:</u> El Usuario selecciona una de las opciones de confirmación de editar usuario. Si: Continúa con el paso 9 de secuencia normal. No: El sistema cancela la actualización de información muestra un mensaje y permite al usuario corregir la información.
Excepciones: E-1: <u>Completar todos los campos.</u> 1. Indicar completar los campos requeridos al usuario 2. Se regresa al paso 3 de la secuencia normal. E-2: <u>Error para actualizar los datos.</u> 1. El sistema le permite al usuario corregir la información e internarla nuevamente. 2. Se regresa al paso 4 de secuencia normal.

CASO DE USO: CAMBIAR CONTRASEÑA DENTRO DEL SISTEMA

Tabla 12 Cambiar Contraseña

Caso de uso: Cambiar contraseña del usuario desde el sistema
Objetivo: Permitir que el usuario cambie su contraseña dentro del sistema, sin tener que recuperarla desde la opción Recuperar contraseña.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión El usuario debe recordar su contraseña actual

<p>Secuencia normal:</p> <p>Este caso de uso empieza cuando el usuario ingresa a la opción cambiar contraseña dentro del sistema.</p> <p>El sistema muestra un formulario con dos campos</p> <p>Contraseña Actual</p> <p>Contraseña nueva</p> <p>El usuario ingresa su contraseña actual y su nueva contraseña. Sub-flujo 1 en caso de error.</p> <p>Si todo es correcto el sistema actualiza la nueva contraseña en la base de datos.</p> <p>El sistema muestra un mensaje de contraseña cambiada correctamente.</p> <p>El sistema redirige al usuario a la interfaz de Dashboard.</p> <p>Fin del caso de uso.</p>
<p>Sub-flujo:</p> <p>S-1: Contraseña actual incorrecta;</p> <p>Verifica que la contraseña actual sea la correcta registrada en la base de datos.</p> <p>Si la contraseña es incorrecta, el sistema muestra un mensaje de error La contraseña actual es incorrecta.</p> <p>El sistema permite al usuario corregir.</p> <p>El sistema regresa al paso 2 de secuencia normal.</p>
<p>Excepciones:</p> <p>E-1: <u>Completar todos los campos.</u></p> <ol style="list-style-type: none"> 1. Indicar completar los campos requeridos al usuario en caso de que estén vacíos. 2. Se regresa al paso 2 de secuencia normal. <p>E-2: <u>Error para actualizar la contraseña.</u></p> <ol style="list-style-type: none"> 1. Si por alguna razón el sistema no puede actualizar la contraseña por un problema interno, muestra un mensaje al usuario.

CASO DE USO: RECUPERAR CONTRASEÑA

Tabla 13 Recuperar Contraseña

Caso de uso: Recuperar contraseña del usuario
Objetivo: Permitir al usuario recuperar la contraseña en caso de olvido.
Actores: Usuario
<p>Precondiciones:</p> <p>El usuario debe tener una cuenta registrada en el sistema.</p> <p>El sistema debe estar activo y conectado a la base de datos.</p>
<p>Secuencia normal:</p> <p>Este caso de uso comienza cuando el usuario selecciona la opción "Recuperar contraseña" en la página de inicio de sesión.</p> <p>El sistema solicita al usuario ingresar su correo electrónico asociado a la cuenta.</p> <p>El usuario ingresa su correo electrónico y confirma la acción.</p> <p>El sistema valida que el correo electrónico ingresado está registrado en la base de datos. Sub-flujo 1 si no es correcto.</p> <p>El sistema envía un enlace de recuperación al correo electrónico ingresado.</p> <p>El usuario abre el correo y hace clic en el enlace proporcionado.</p> <p>El sistema redirige al usuario a una página para establecer una nueva contraseña.</p> <p>El usuario ingresa y confirma su nueva contraseña.</p> <p>El sistema valida que la nueva contraseña cumpla con los requisitos establecidos (longitud mínima, caracteres especiales, etc.).</p> <p>El sistema actualiza la contraseña del usuario en la base de datos y muestra un mensaje de confirmación.</p> <p>Fin del caso de uso.</p>
<p>Sub-flujo:</p> <p>S-1: <u>Validar correo electrónico:</u></p> <p>Si el correo ingresado no está registrado en la base de datos, el sistema muestra un mensaje de error.</p>

<p>El flujo regresa al paso 2 de secuencia normal</p> <p>S-2: <u>Verificar requisitos de contraseña:</u> Si la nueva contraseña no cumple con los requisitos, el sistema muestra un mensaje indicando las reglas. Se regresa al paso 8 del flujo principal.</p>
<p>Excepciones:</p> <p>E-1: <u>Error de conexión al servidor de correo:</u> Si el sistema no puede enviar el enlace de recuperación, muestra un mensaje de error.</p> <p>E-2: <u>Enlace de recuperación expirado:</u> Si el usuario intenta usar un enlace de recuperación expirado, el sistema muestra un mensaje de error. El usuario puede reenviar el enlace. Se regresa al paso 2 de secuencia normal.</p> <p>E-3: <u>Error al actualizar la contraseña:</u> Si el sistema no puede actualizar la contraseña en la base de datos, muestra un mensaje de error: "No se pudo actualizar la contraseña." Se permite al usuario intentarlo nuevamente.</p>

CASO DE USO: CERRAR SESIÓN

Tabla 14 Cerrar Sesión

Caso de uso: Cerrar sesión
Objetivo: Permitir al usuario cerrar la sesión de forma segura del usuario logueado.
Actores: Usuario
<p>Precondiciones: El usuario debe haber iniciado sesión previamente. El sistema debe estar activo.</p>
<p>Secuencia normal: Este caso de uso comienza cuando el usuario selecciona la opción "Cerrar sesión" en la interfaz de la aplicación. El sistema muestra un mensaje de confirmación para asegurarse de que el usuario desea cerrar sesión. El usuario confirma la acción. El sistema elimina la sesión activa del usuario y borra cualquier token o cookie de autenticación. El sistema redirige al usuario a la página de inicio de sesión. Fin del caso de uso.</p>
<p>Sub-flujo: S-1: <u>Confirmar cierre de sesión:</u> El sistema muestra una ventana de confirmación Si el usuario selecciona no, el sistema cancela el proceso y regresa a la interfaz principal. Si el usuario selecciona si, continúa con el paso 4 del flujo principal.</p>
<p>Excepciones: E-1: <u>Fallo en la desconexión del servidor:</u> Si el sistema muestra un fallo de desconexión, automáticamente cierra la sesión y elimina el token o cookie que esté en el sistema.</p>

CASO DE USO: VISUALIZACIÓN DATOS EN TIEMPO REAL

Tabla 15 Visualizar Datos en Tiempo Real

Caso de uso: Visualizar datos en tiempo real.
Objetivo: Permitir al usuario visualizar datos actualizados en tiempo real, provenientes de sensores conectados al sistema.
Actores: Usuario
<p>Precondiciones: El usuario debe haber iniciado sesión previamente.</p>

El sistema debe estar activo.
Los sensores deben estar conectados y enviando datos correctamente.
Secuencia normal: Este caso de uso comienza cuando el usuario accede a la opción Tiempo Real del menú en el aplicativo web. El sistema establece una conexión con los sensores para recibir los datos. El sistema está a la escucha de los datos provenientes de los sensores. Los valores actuales de los parámetros, como pH, turbidez y oxígeno disuelto, se muestran en tiempo real en la interfaz del usuario. Los datos se actualizan automáticamente en base a cómo van llegando desde los sensores. Fin del caso de uso.
Sub-flujo:
Excepciones: E-1: <u>Fallo en la conexión con los sensores:</u> Si la conexión falla y no permite el envío de los datos se muestra en la consola el error del socket. Si la reconexión falla, se informa al usuario y se finaliza el caso de uso.

CASO DE USO: LISTA DE USUARIOS REGISTRADOS

Tabla 16 Lista de Usuarios Registrados

Caso de uso: Lista de usuarios registrados
Objetivo: Permitir al usuario visualizar una lista completa de los usuarios registrados en el sistema.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión previamente en el sistema. El sistema debe estar activo y conectado a la base de datos.
Secuencia normal: Este caso de uso comienza cuando el usuario selecciona la opción "Lista Usuarios" en el menú principal. El sistema se conecta a la base de datos para recuperar la lista de usuarios registrados. El sistema muestra la lista en la interfaz, incluyendo información básica de cada usuario como (nombre, correo, edad, acciones) Fin del caso de uso.
Sub-flujo:
Excepciones: E-1 <u>Error conexión con base de datos:</u> Si el sistema no puede acceder a la base de datos, muestra un mensaje de error indicando la falla.

Caso de uso: Editar lista de usuarios registrados
Objetivo: Permitir al usuario editar en una lista completa de los usuarios registrados en el sistema.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión previamente en el sistema. El sistema debe estar activo y conectado a la base de datos. El sistema debe contener usuarios registrados en la base de datos.
Secuencia normal: El usuario selecciona la opción "Lista de Usuarios" en el menú principal. El sistema recupera la lista de usuarios registrados de la base de datos en una interfaz. El usuario selecciona un usuario de la lista y hace clic en la opción "Editar" . El sistema muestra un formulario con la información actual del usuario seleccionado. El usuario modifica los campos deseados (nombre, edad, correo) y confirma los cambios. El sistema valida los datos ingresados.

El sistema actualiza la información del usuario en la base de datos y muestra un mensaje de confirmación. Fin del caso de uso.
Sub-flujo:
Excepciones: E-1 <u>Error cargar lista de usuarios:</u> Si algún campo está vacío o tiene un formato incorrecto, el sistema muestra un mensaje de error. El flujo regresa al paso 5 para que el usuario corrija los errores. E-2 <u>Error al guardar los cambios:</u> Si el sistema no puede actualizar los datos se muestra un mensaje de error.
Caso de uso: Eliminar lista de usuarios registrados
Objetivo: Permitir al usuario eliminar en una lista completa de los usuarios registrados en el sistema.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión previamente en el sistema. El sistema debe estar activo y conectado a la base de datos. El sistema debe contener usuarios registrados en la base de datos.
Secuencia normal: El usuario selecciona la opción " Lista de Usuarios " en el menú principal. El sistema recupera la lista de usuarios registrados de la base de datos en una interfaz. El usuario selecciona un usuario de la lista y hace clic en la opción " Eliminar ". El sistema muestra un formulario para confirmar la eliminación del usuario. El sistema muestra un mensaje de usuario eliminado correctamente. El sistema redirecciona al usuario a la lista de usuarios. El sistema actualiza la lista de los usuarios en la base de datos y muestra la lista de usuarios actualizada. Fin del caso de uso.
Sub-flujo:
Excepciones: E-1 <u>Error eliminar al usuario:</u> Si el sistema no permite eliminar el usuario seleccionado muestra un mensaje de error. El flujo regresa al paso 2 de secuencia normal . E-2 <u>Error eliminar usuario logueado:</u> Si el sistema detecta que el usuario elimino el usuario que esta iniciado sesión, elimina el token o cookie y direcciona al usuario a la interfaz de inicio de sesión.

CASO DE USO: GENERACIÓN DE ALERTAS AUTOMATICAS

Tabla 17 Generación de Alertas automáticas

Caso de uso: Generación de alertas automáticas
Objetivo: Permitir que el sistema genere alertas automáticamente cuando los datos recibidos desde los sensores IoT superen un umbral configurado por el usuario.
Actores: Sistema Genera las alertas automáticamente. Usuario quien recibe las alertas.
Precondiciones: El usuario debe haber iniciado sesión previamente en el sistema. El sistema debe estar activo y conectado en tiempo real a los sensores.
Secuencia normal: El sistema está a la escucha de los datos en tiempo real provenientes de los sensores IoT. Cuando llega un nuevo dato, el sistema verifica si supera el umbral de los sensores seleccionados por el usuario. Si el dato supera el umbral, el sistema genera una alerta automática.

El sistema envía la alerta al módulo de notificaciones para que sea registrada y visualizada por el usuario. Fin del caso de uso.
Sub-flujo:
Excepciones: E-1: <u>No hay conexión con los sensores:</u> Si el sistema no está recibiendo datos de los sensores, se muestra un mensaje de error en la consola y se intenta reconectar. Si la reconexión falla, se notifica al usuario sobre el problema de conexión.

CASO DE USO: NOTIFICACIONES DE ALERTAS

Tabla 18 Notificaciones de Alertas

Caso de uso: Notificaciones de alertas
Objetivo: Permitir que el usuario reciba notificaciones cada vez que se genere una alerta y pueda gestionar su visualización.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión previamente. El sistema debe estar generando alertas automáticas cuando los valores de los sensores superen los umbrales configurados.
Secuencia normal: Una alerta automática es generada por el sistema debido a un valor fuera del umbral. La alerta se registra en el módulo de notificaciones. El contador de notificaciones se autoincrementa. El usuario hace clic en el icono de notificaciones para visualizar los datos generados. El sistema muestra una ventana emergente con la lista de notificaciones. El usuario puede elegir entre: a. Cerrar la ventana de notificaciones. b. Limpiar las notificaciones, eliminándolas del historial. c. Buscar una notificación específica. Fin del caso de uso.
Sub-flujo:
Excepciones: E-1: No hay alertas generadas Si el usuario accede a la ventana de notificaciones y no hay alertas registradas, el sistema muestra un mensaje indicando que no hay notificaciones.

CASO DE USO: SELECCIONAR SENSORES

Tabla 19 Seleccionar Sensores

Caso de uso: Seleccionar sensores
Objetivo: Permitir al usuario elegir qué sensores desea recibir datos en tiempo real y generar alertas cuando los valores superen los umbrales configurados.
Actores: Usuario
Precondiciones: El usuario debe haber iniciado sesión previamente. El sistema debe estar activo y recibiendo datos en tiempo real desde los sensores.
Secuencia normal: El usuario accede a la opción "Seleccionar Sensores" en la interfaz del sistema.

<p>El sistema muestra una lista de los sensores disponibles para monitoreo (Ej: pH, turbidez, oxígeno disuelto, conductividad, etc.).</p> <p>El usuario selecciona los sensores de los cuales desea recibir datos y alertas.</p> <p>El usuario confirma la selección presionando el botón "Guardar".</p> <p>El sistema registra la selección del usuario y actualiza la configuración de monitoreo.</p> <p>A partir de este momento, el sistema sólo genera alertas y notificaciones para los sensores seleccionados.</p> <p>Fin del caso de uso.</p>
<p>Sub-flujo:</p> <p>S-1: Modificar la selección de sensores</p> <p>El usuario accede nuevamente a la opción "Seleccionar Sensores".</p> <p>El usuario cambia los sensores.</p> <p>El usuario confirma los cambios presionando "Guardar".</p> <p>El sistema actualiza la configuración y aplica los nuevos sensores seleccionados.</p>
<p>Excepciones:</p> <p>E-1: No hay sensores disponibles</p> <p>Si el sistema no detecta sensores conectados, muestra un mensaje: "No hay sensores".</p> <p>El usuario solo podrá seleccionar sensores una vez que se establezca la conexión.</p> <p>E-2: Usuario no selecciona ningún sensor</p> <p>Si el usuario intenta guardar sin haber seleccionado sensores, el sistema muestra un mensaje de advertencia: "Debe seleccionar al menos un sensor".</p> <p>Se permite al usuario regresar a la lista y seleccionar los sensores deseados.</p>

3. Historias de usuario

Las historias de usuario es una técnica clave en la ingeniería de requerimientos, que es utilizada en metodologías ágiles, especialmente como lo es SCRUM y Kanban esto para poder definir y así tener una estructuración más clara de los requisitos o funcionalidades que se deben implementar en el producto software.

Las historias de usuario son una herramienta que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesarios. Forman parte de la fórmula de captura de funcionalidades definida en 2001 por Ron Jeffries de las tres Cs:

- I. **Card:** cada historia de usuario se reduce hasta hacerla fácil de memorizar y de sintetizar en una tarjeta o post-it. La tarjeta sirve como recordatorio y promesa de una conversación posterior.
- II. **Conversation:** el equipo de desarrollo y el propietario del producto añaden criterios de aceptación a cada historia poco antes de su implementación. Los cambios son bienvenidos en agilidad, por lo que no tiene sentido profundizar en estos detalles antes. La situación puede variar mucho desde el momento en el que se sintetiza la funcionalidad en la tarjeta hasta que se implementa.
- III. **Confirmation:** el propietario del producto o usuario de negocio confirma que el equipo de desarrollo ha entendido y recogido correctamente sus requisitos revisando los criterios de aceptación. A veces se pueden presentar transformados en escenarios de pruebas. (Alexander Menzinsk & Juan Palacio, Miguel Ángel Sobrino, 2022)

Teniendo en cuenta lo anterior, para la implementación de historias de usuario para el desarrollo de las funcionalidades del módulo de usuario, esto fue fundamental para lograr los respectivos avances continuos, las historias definidas fueron las siguientes.

3.1 Módulo Autenticación y Gestión de Usuarios

3.1.1 Historia de usuario Iniciar sesión

Tabla 20 Historia de Usuario Iniciar Sesión

Historia de usuario	Fecha
Historia de usuario No: 001: Iniciar sesión	
Nombre: Iniciar sesión	Prioridad: Alta
Descripción: Como usuario, quiero poder iniciar sesión con mi correo y contraseña, para así acceder a la plataforma y poder visualizar mi información y hacer uso de las funciones.	
Estimación:	
Criterios de aceptación: El usuario tiene que poder ingresar el correo y contraseña válidos. Si las credenciales ingresadas son incorrectas, se debe mostrar un mensaje de error. Si las credenciales ingresadas son correctas se muestra un mensaje de éxito y se accede a la interfaz de Dashboard.	

3.1.2 Historia de usuario registro de usuario

Tabla 21 Historia de Usuario Registro

Historia de usuario	Fecha
Historia de usuario No: 002: Registro de usuario	
Nombre: Registro de usuario	Prioridad: Alta
Descripción: Como usuario nuevo quiero poder registrarme en el sistema con los datos para poder iniciar sesión y acceder a las funciones del sistema.	
Estimación:	
Criterios de aceptación: El sistema debe solicitarle al usuario algunos datos como nombre, edad, correo, contraseña Si el correo ya se encuentra registrado en la base de datos, el sistema debe mostrar un mensaje de error. Si los datos son correctos, el sistema debe permitir crear la cuenta, permitir el inicio y dirigir al usuario a la interfaz de inicio de sesión.	

3.1.3 Historia de usuario cerrar sesión

Tabla 22 Historia de Usuario Cerrar Sesión

Historia de usuario	Fecha
Historia de usuario No: 003: Cerrar sesión	
Nombre: Cerrar sesión	Prioridad: Media
Descripción: Como usuario quiero poder cerrar sesión de la cuenta de forma segura eliminando cualquier token o cookie almacenado.	
Estimación:	
Criterios de aceptación: Debe haber una opción visible de "Cerrar Sesión". El usuario debe confirmar la acción antes de cerrar sesión.	

El sistema debe eliminar la sesión y redirigir al interfaz de inicio de sesión.
Después de cerrar sesión, el usuario no puede acceder a las funciones de la aplicación sin volver a iniciar sesión.

3.1.4 Historia de usuario cambio de contraseña dentro del sistema

Tabla 23 Historia de Usuario Cambiar Contraseña

Historia de usuario	Fecha
Historia de usuario No: 004: Cambio de contraseña dentro del sistema	
Nombre: Cambio de contraseña dentro del sistema.	Prioridad: Alta
Descripción: Como usuario completamente autenticado quiero poder cambiar mi contraseña desde el sistema sin tener ir a la opción de recuperar contraseña esto para mejorar la seguridad de mi cuenta.	
Estimación:	
Criterios de aceptación: El usuario debe ingresar su correo contraseña Actual para verificar antes de cambiar a la nueva contraseña. Si la confirmación de la nueva contraseña no coincide entonces, se le debe mostrar un mensaje de error. El usuario debe confirmar el cambio de contraseña. Si la contraseña se cambia correctamente, se debe mostrar un mensaje de confirmación en el cambio de contraseña.	

3.1.5 Historia de usuario Editar usuario

Tabla 24 Historia de Usuario Editar Usuario

Historia de usuario	Fecha
Historia de usuario No: 005: Editar usuario	
Nombre: Editar usuario	Prioridad: Alta
Descripción: Como usuario quiero poder editar la información de mi cuenta, como lo es el nombre, edad y correo de forma segura para mantener mi perfil actualizado.	
Estimación:	
Criterios de aceptación: El usuario debe ingresar a la opción modificar usuario y podrá modificar la información de la cuenta. Si las modificaciones que el usuario haga no son correctas en el formato del correo o campos vacíos, se debe mostrar un error. Si las modificaciones son correctas, se debe mostrar un mensaje de modificación exitosa de los datos. El usuario será dirigido a la interfaz de Dashboard. El usuario podrá ver su perfil completamente actualizado con la nueva información.	

3.1.6 Historia de usuario Listar editar y eliminar usuarios registrados

Tabla 25 Historia de Usuario Listar Editar y Eliminar Usuarios

Historia de usuario	Fecha
Historia de usuario No: 006: Listar editar y eliminar usuarios registrados.	
Nombre: Listar editar y eliminar usuarios registrados.	Prioridad: Alta
Descripción: Como usuario quiero poder ver en una tabla a todos los usuarios registrados en el sistema y que me permita realizar las respectivas acciones como eliminar y editar usuarios.	

Estimación:
Criterios de aceptación: El usuario debe poder ver la lista de todos los usuarios registrados y poder editarlos o eliminarlos. Si el usuario selecciona la opción eliminar, se debe mostrar una ventana para permitir la confirmación de la eliminación de esa cuenta. Si el usuario selecciona la opción de editar, se debe mostrar los datos de ese usuario y poder editarlos. Si las modificaciones son correctas se debe mostrar un mensaje de éxito. Si las modificaciones no son correctas se debe mostrar un mensaje de error. El será dirigido a la tabla de la lista de todos los usuarios registrados con los datos actualizados con la nueva información.

3.1.7 Historia de usuario validar datos ingresados

Tabla 26 Historia de Usuario Validar Datos Ingresados

Historia de usuario	Fecha
Historia de usuario No: 007: Validar datos ingresados	
Nombre: Validar datos ingresados	Prioridad: Alta
Descripción: Como usuario quiero que el sistema valide los datos que introduzca para asegurar que la información es correcta y no hay campos requeridos vacíos, para poder efectuar las acciones del usuario especificadas anteriormente.	
Estimación:	
Criterios de aceptación: El sistema debe validar los datos introducidos por el usuario. El usuario debe mostrar un mensaje de error si los datos no son válidos o hay campos requeridos que no pueden estar vacíos.	

3.2 Módulo de monitoreo de datos en tiempo real

3.2.1 Historia de usuario visualizar datos en tiempo real

Tabla 27 Historia de Usuario Visualizar Datos en Tiempo Real

Historia de usuario	Fecha
Historia de usuario No: 008: Visualizar datos en tiempo real	
Nombre: Visualizar datos en tiempo real	Prioridad: Alta
Descripción: Como usuario quiero poder visualizar datos en tiempo en base a como vayan siendo enviados desde el dispositivo IoT.	
Estimación:	
Criterios de aceptación: Los datos de los sensores (pH, turbidez, oxígeno disuelto, entre otros) deben actualizarse en tiempo real sin tener que estar recargando la página. Los datos deben ser claros y fáciles de entender y ser adaptados a diferentes dispositivos.	

3.2.2 Historia de usuario visualizar datos en Dashboard

Tabla 28 Historia de Usuario Dashboard

Historia de usuario	Fecha
Historia de usuario No: 009: Visualizar datos en Dashboard	
Nombre: Visualizar datos en Dashboard	Prioridad: Alta

Descripción: Como usuario, quiero poder visualizar los datos de los sensores en tiempo real en un Dashboard interactivo creado en Grafana, para identificar de forma rápida los cambios en los parámetros de calidad del agua para una rápida toma de decisiones.
Estimación:
Criterios de aceptación: Los datos de los sensores (pH, turbidez, oxígeno disuelto, entre otros) deben actualizarse en tiempo real sin tener que estar recargando la página. Los datos deben ser claros y fáciles de entender y ser adaptados a diferentes dispositivos. El Dashboard debe cambiar en base a los rangos de los valores (Verde, amarillo, rojo) El usuario puede filtrar los tiempos de los datos a (5 minutos, 10 minutos, 1 hora etc.). Los datos deben agregarse automáticamente a cada gráfica el cual corresponde cada dato.

3.2.3 Historia de usuario: Visualizar datos en una tabla de datos críticos

Tabla 29 Historia de Usuario Datos Críticos

Historia de usuario	Fecha
Historia de usuario No: 010: Visualizar datos en una tabla de datos críticos	
Nombre: Visualizar datos en una tabla de datos críticos.	Prioridad: Alta
Descripción: Como usuario, quiero poder visualizar los datos de los sensores en tiempo real en una tabla para identificar y confirmar de forma rápida los cambios en los parámetros de calidad del agua.	
Estimación:	
Criterios de aceptación: Los datos de los sensores (pH, turbidez, oxígeno disuelto, fecha entre otros) deben agregarse a la tabla automáticamente sin necesidad de recargar la página. Los datos deben ser fáciles de entender y adaptables a diferentes dispositivos.	

3.2.4 Historia de usuario seleccionar sensores a monitorear

Tabla 30 Historia de Usuario Seleccionar Sensores

Historia de usuario	Fecha
Historia de usuario No: 011: seleccionar sensores a monitorear	
Nombre: Seleccionar sensores a monitorear	Prioridad: Alta
Descripción: Como usuario quiero poder seleccionar los sensores de los cuales quiero recibir los valores, esto con el fin de poder recibir las alertas y notificaciones solo de los sensores que me interesen.	
Estimación:	
Criterios de aceptación: El sistema debe mostrar los sensores disponibles. El usuario debe poder seleccionar los sensores. Si el usuario no selecciona ningún sensor, el sistema no debe mostrar datos en ningún momento.	

3.3 Módulo de monitoreo de alertas

3.3.1 Historia de usuario: Generación de alertas automáticas

Tabla 31 Historia de Usuarios Generación de Alertas

Historia de usuario	Fecha
Historia de usuario No: 012: Generación de alertas automáticas	

Nombre: Generación de alertas automáticas	Prioridad: Alta
Descripción: Como usuario quiero recibir alertas automáticamente cuando los valores de los sensores superen el umbral para una reacción rápida ante situaciones críticas en las variables de la calidad del agua.	
Estimación:	
Criterios de aceptación: Las alertas solo deben generarse cuando un valor supere el umbral establecido. Las alertas deben mostrarse en cualquier interfaz del sistema en tiempo real. Si no hay sensores seleccionados, el sistema no debe poder generar alertas.	

3.3.2 Historia de usuario: Generación de notificaciones

Tabla 32 Historia de Usuario Generación de Notificaciones

Historia de usuario	Fecha
Historia de usuario No: 013:	
Nombre: Generación de alertas automáticas	Prioridad: Alta
Descripción: Como usuario quiero recibir alertas automáticamente cuando los valores de los sensores superen el umbral para una reacción rápida ante situaciones críticas en las variables de la calidad del agua.	
Estimación:	
Criterios de aceptación: Las alertas solo deben generarse cuando un valor supere el umbral establecido. Las alertas deben mostrarse en cualquier interfaz del sistema en tiempo real. Si no hay sensores seleccionados, el sistema no debe poder generar alertas.	

3.4 Módulo de Almacenamiento

3.4.1 Historia de usuario: almacenar datos recibidos

Tabla 33 Historia de Usuario Almacenar Datos

Historia de usuario	Fecha
Historia de usuario No: 014: Almacenar datos recibidos	
Nombre: Almacenar datos recibidos	Prioridad: Media
Descripción: Quiero poder almacenar los datos críticos de los sensores para poder visualizarlos.	
Estimación:	
Criterios de aceptación: Todos los datos recibidos que son críticos se deben almacenar en la base de datos.	

CAPÍTULO 2. CONSTRUCCIÓN DEL SISTEMA WEB PARA EL ANÁLISIS Y VISUALIZACIÓN DE DATOS EN ALERTAS TEMPRANAS DE CALIDAD DEL AGUA POR TELEMETRÍA PARA ACUEDUCTOS RURALES.

1. Diagrama de secuencia

Los diagramas de secuencia son un tipo de diagrama UML (Unified modeling Language) que permite capturar el orden de las interacciones en los diagramas de secuencia se hace uso de diferentes elementos que nos permiten entender el flujo de los diagramas.

Los principales elementos del diagrama de secuencia

“El diagrama de secuencias es un esquema conceptual que permite representar el comportamiento de un sistema, para lo cual emplea la especificación de los objetos que se encuentran en un escenario y la secuencia de mensajes intercambiados entre ellos, con el fin de llevar a cabo una transacción del sistema.”

1.1 Diagrama de secuencia Inicio de sesión

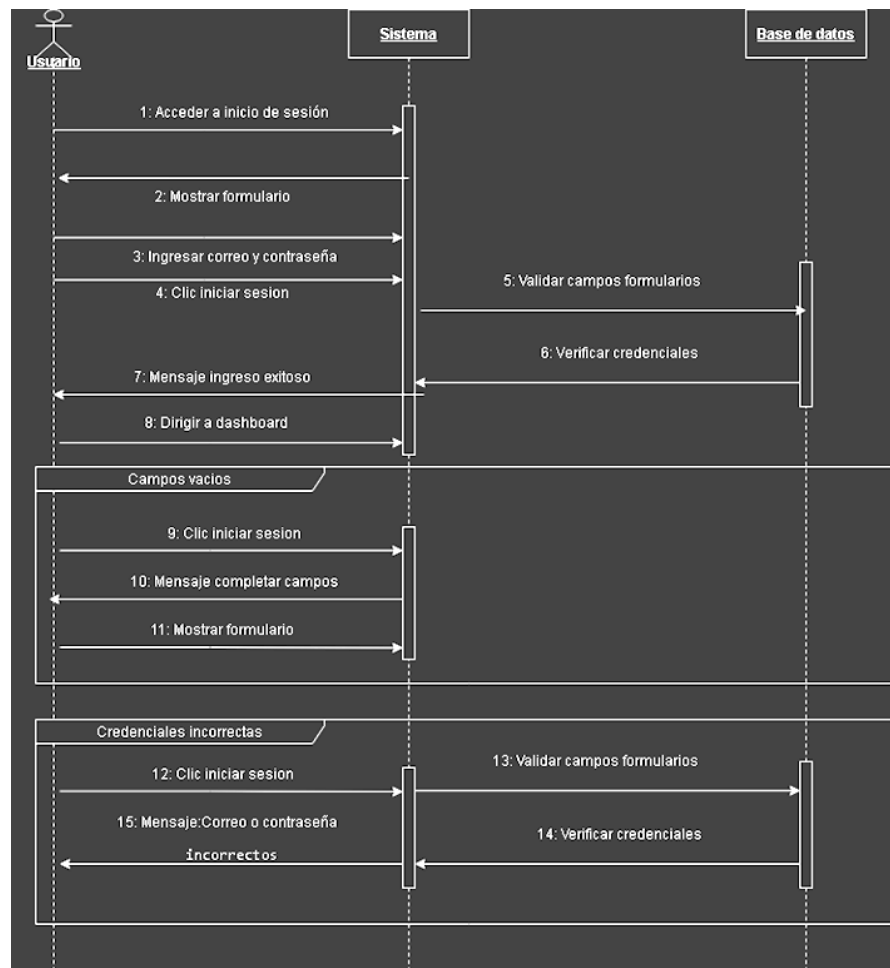


Figura 24 Diagrama de Secuencia Inicio de Sesión

1.2 Diagrama de secuencia registro de usuario

Este diagrama de secuencia describe el flujo para que el usuario se registre en el sistema. Detalla cómo el usuario ingresa sus datos, cómo el sistema valida y registra la información en la base de datos, y cómo se redirige al usuario a la página de inicio de

sesión. También contempla escenarios de error, como campos incompletos o correos ya registrados, proporcionando retroalimentación al usuario.

Este diagrama de secuencia ilustra diferentes escenarios cada paso mencionado en el diagrama de flujo resalta el mensaje y la respuesta en función de si las credenciales son correctas, o si son incorrectas. Esto ayuda a comprender mejor el comportamiento del sistema en el escenario de autenticación.

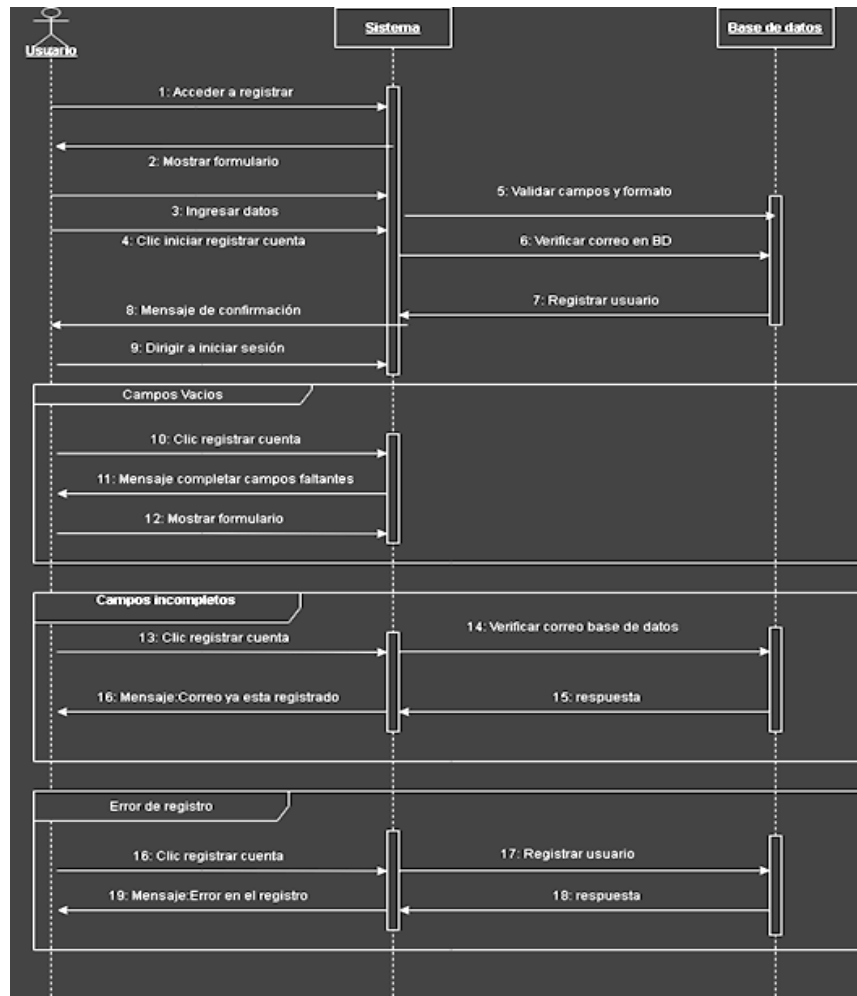


Figura 25 Diagrama de Secuencia Registro de Usuario

1.3 Diagrama de secuencia Dashboard

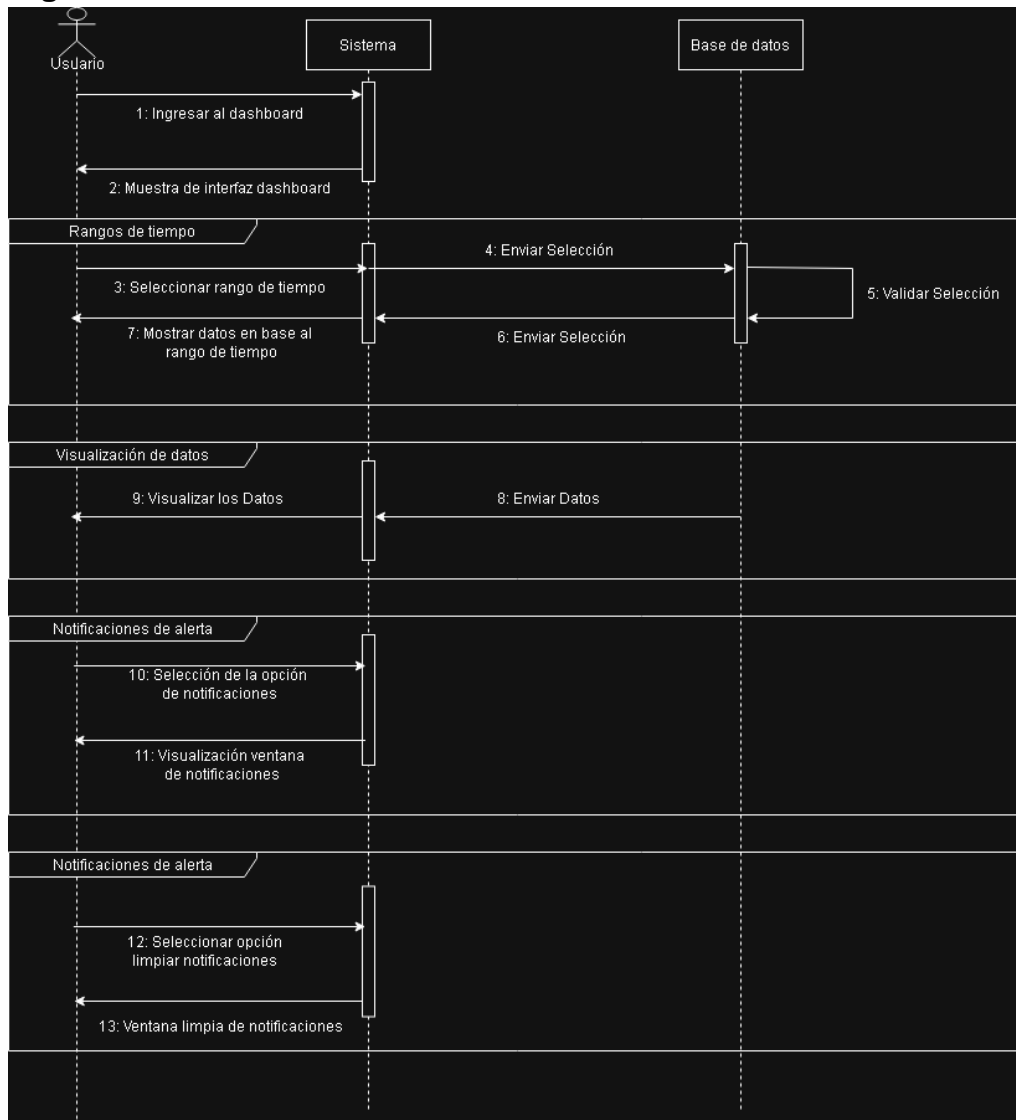


Figura 26 Diagrama de Secuencia Dashboard

1.4 Diagrama de secuencia editar usuario

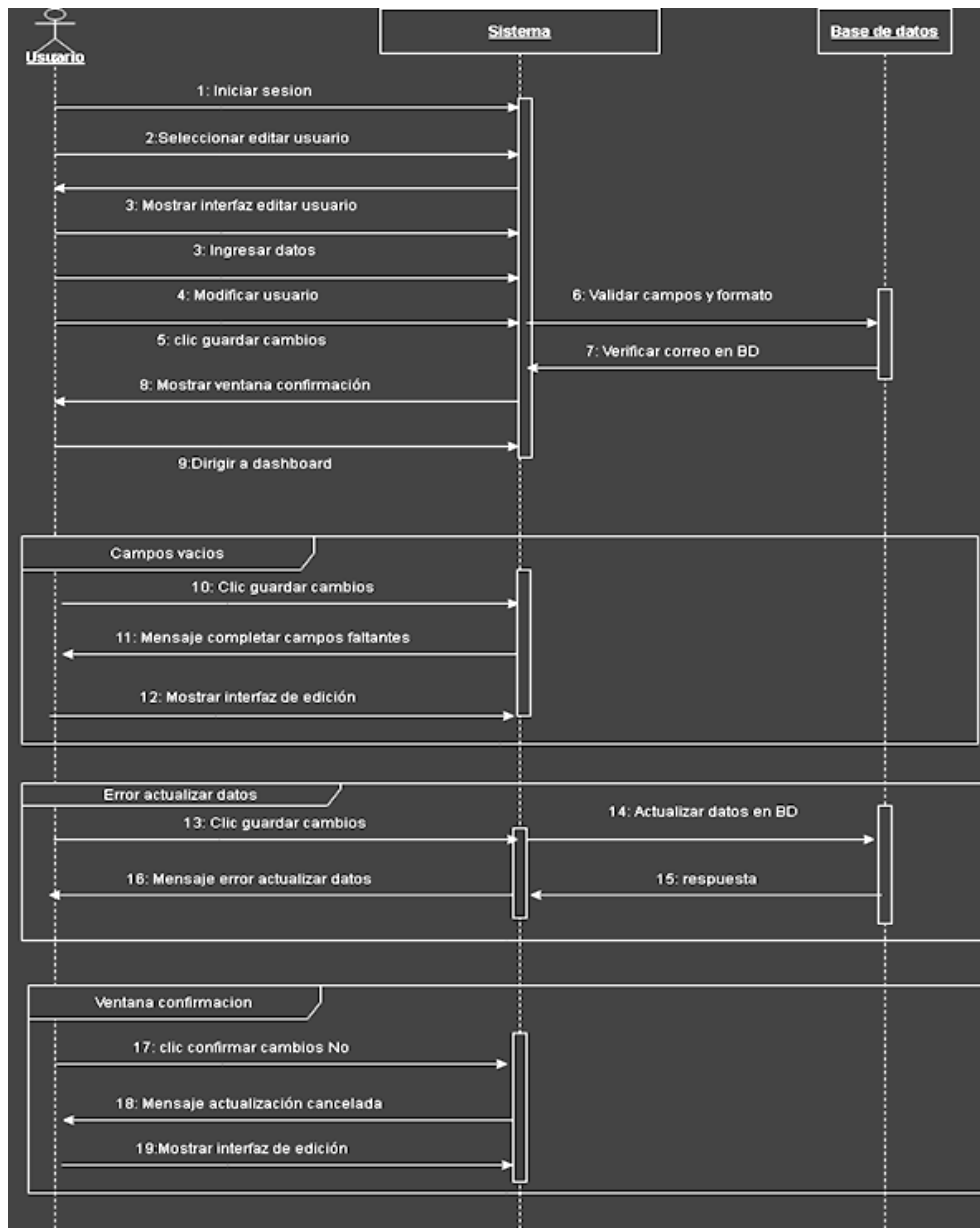


Figura 27 Diagrama de Secuencia Editar Usuario

1.5 Diagrama de secuencia Cambiar contraseña

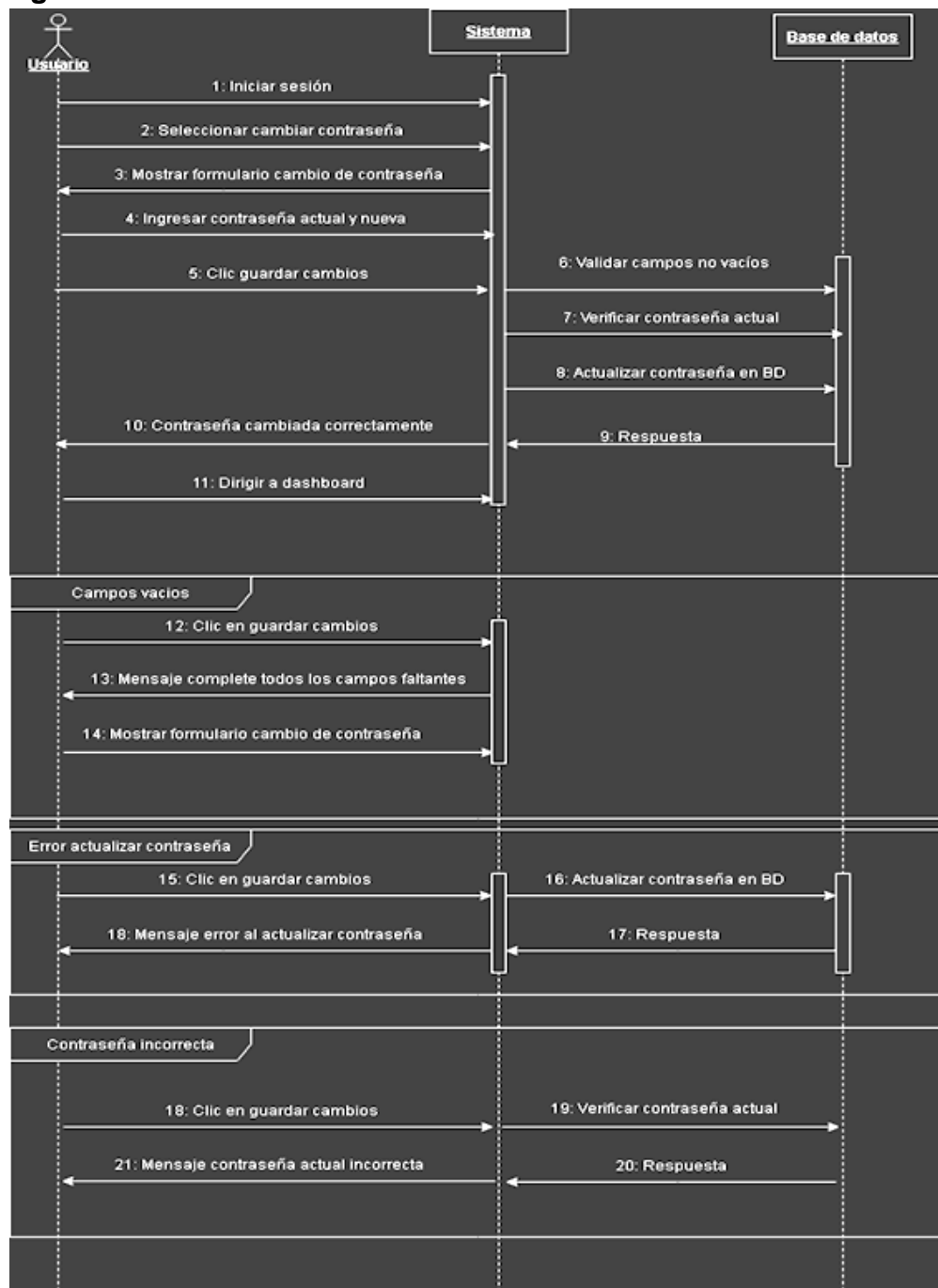


Figura 28 Diagrama de Secuencia Cambiar Contraseña

1.6 Diagrama de secuencia Recuperar contraseña

La secuencia ilustrada a continuación describe el flujo para que el usuario recupere su contraseña en caso de olvido, de forma precisa ilustra cómo el usuario solicita la recuperación, cómo el sistema valida el correo electrónico, envía un enlace de recuperación, y cómo el usuario establece una nueva contraseña, igualmente contempla

escenarios de error, como fallos en la conexión con el servidor de correo o enlaces expirados, proporcionando retroalimentación al usuario.

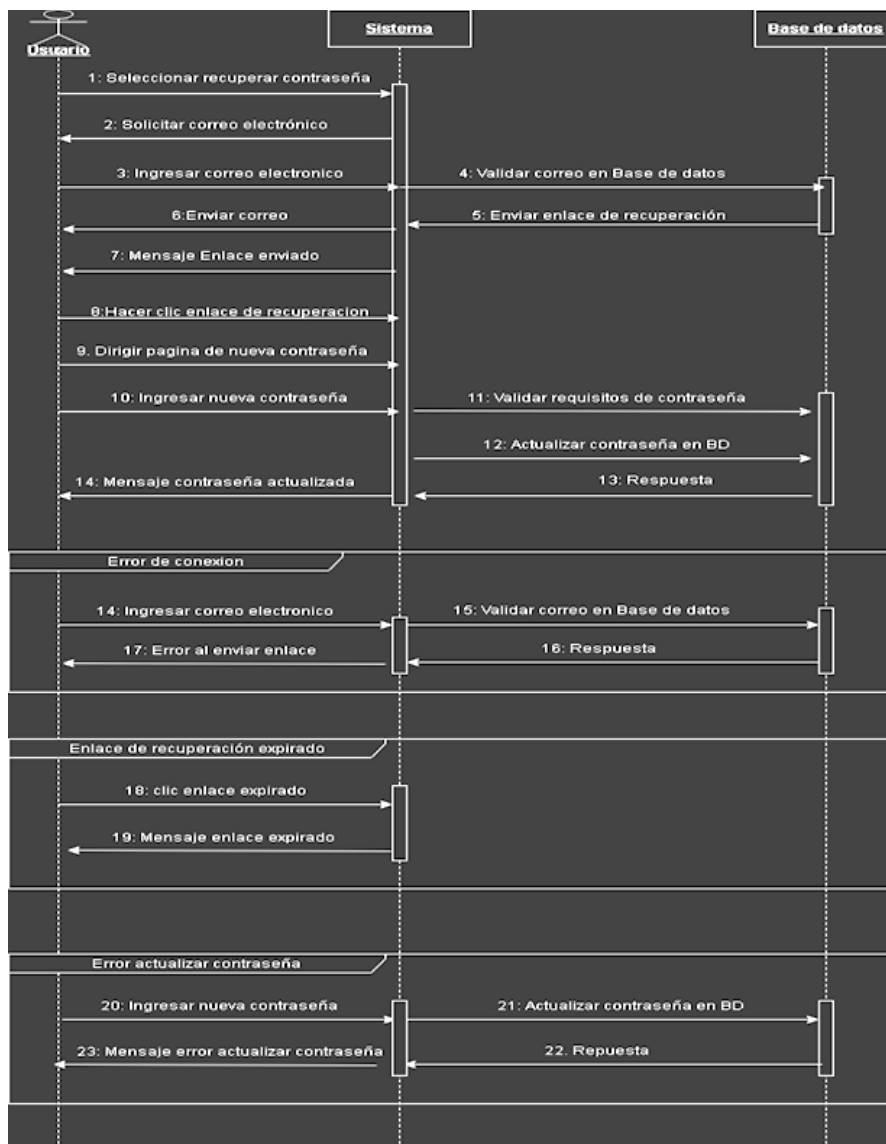


Figura 29 Diagrama de Secuencia Recuperar Contraseña

1.7 Diagrama de secuencia Cerrar sesión

El flujo de la siguiente ilustración, o mejor dicho del diagrama que se encuentra encima de este enunciado, muestra de forma gráfica el proceso para que el usuario cierre su sesión de forma segura en el sistema, detalla cómo el usuario solicita cerrar sesión, cómo el sistema elimina la sesión activa y redirige al usuario a la página de inicio de sesión a su vez contempla el escenario de fallo en la desconexión del servidor, proporcionando retroalimentación al usuario.

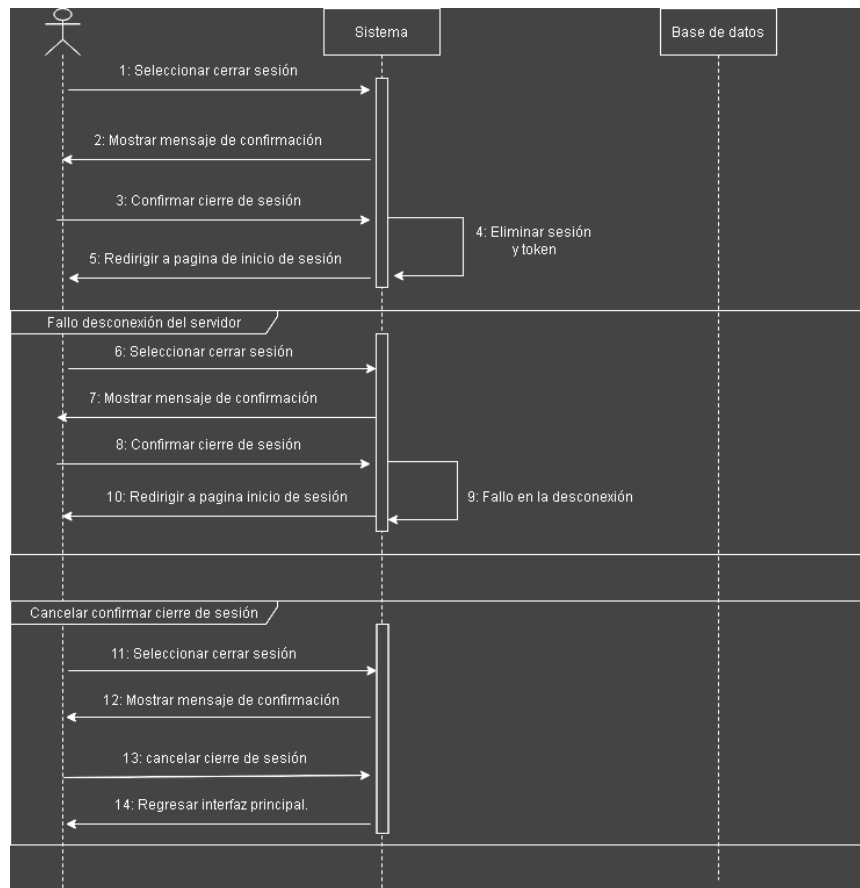


Figura 30 Diagrama de Secuencia Cerrar Sesión

1.8 Diagrama de secuencia Visualizar datos en tiempo real

El diagrama de flujo mostrado en la parte superior es para que el usuario visualice datos en tiempo real provenientes de los sensores conectados al sistema, punto por punto detalla cómo el sistema establece una conexión con los sensores, recibe y muestra los datos en la interfaz del usuario a su vez contempla el escenario de fallo en la conexión con los sensores, proporcionando retroalimentación al usuario.

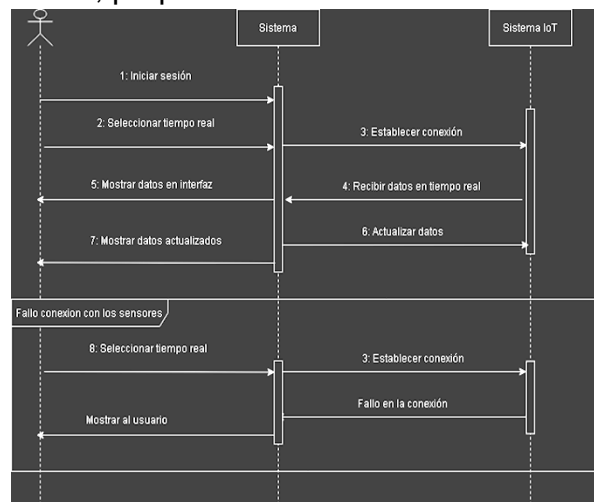


Figura 31 Diagrama de Secuencia Visualizar Datos en Tiempo Real

1.9 Diagrama de secuencia Lista de usuarios editar y eliminar

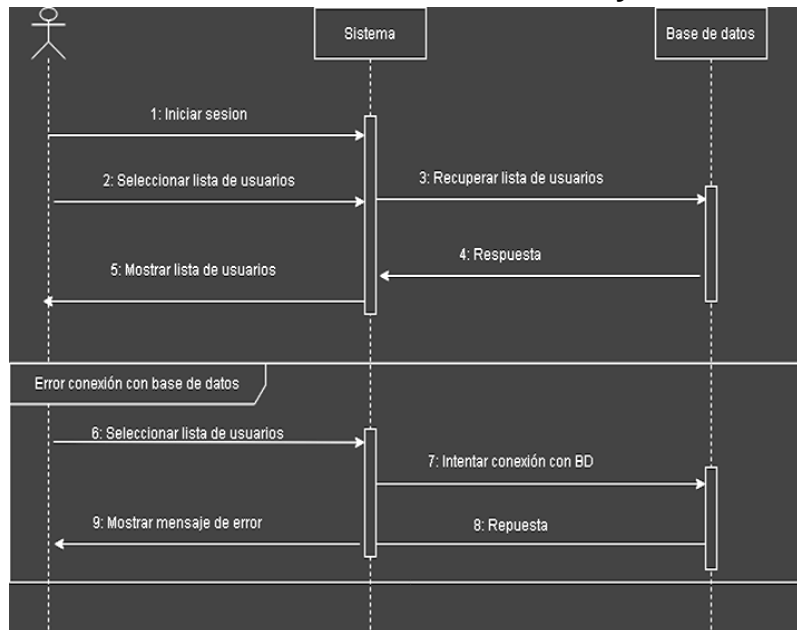


Figura 32 Diagrama de Secuencia Listar Usuarios

El diagrama de secuencia describe el flujo para que el usuario visualice una lista completa de los usuarios registrados en el sistema, mostrando de manera precisa cómo el sistema recupera la lista de usuarios de la base de datos y la muestra en la interfaz del usuario, también contempla el escenario de error en la conexión con la base de datos, proporcionando retroalimentación al usuario.

Mientras que el 2do diagrama de secuencia describe el flujo para que el usuario edite la información de los usuarios registrados en el sistema, un proceso claro y conciso que va paso a paso ilustrando cómo el usuario selecciona un usuario de la lista, modifica sus datos, y cómo el sistema valida y actualiza esta información en la base de datos. También contempla escenarios de error, como campos vacíos o fallos al guardar los cambios, proporcionando retroalimentación al usuario.

Por último, el tercer diagrama de secuencia describe el flujo para que el usuario elimine usuarios registrados en el sistema. Detalla cómo el usuario selecciona un usuario de la lista, confirma la eliminación, y cómo el sistema elimina al usuario de la base de datos. También contempla escenarios de error, como fallos al eliminar o intentar eliminar al usuario logueado, proporcionando retroalimentación al usuario.

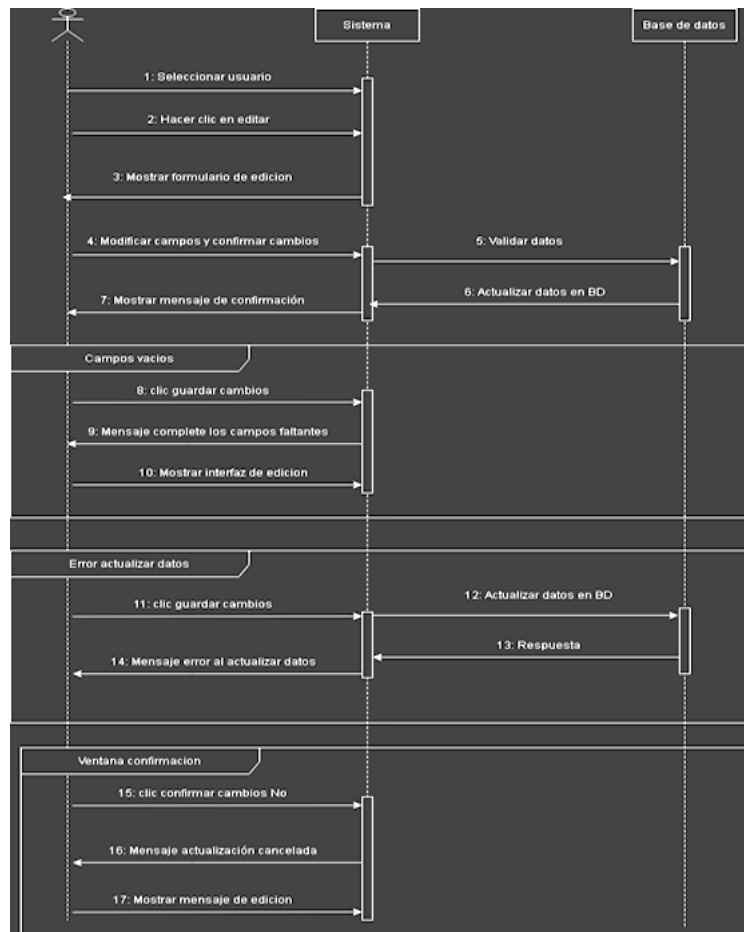


Figura 33 Diagrama de Secuencia Editar Usuarios

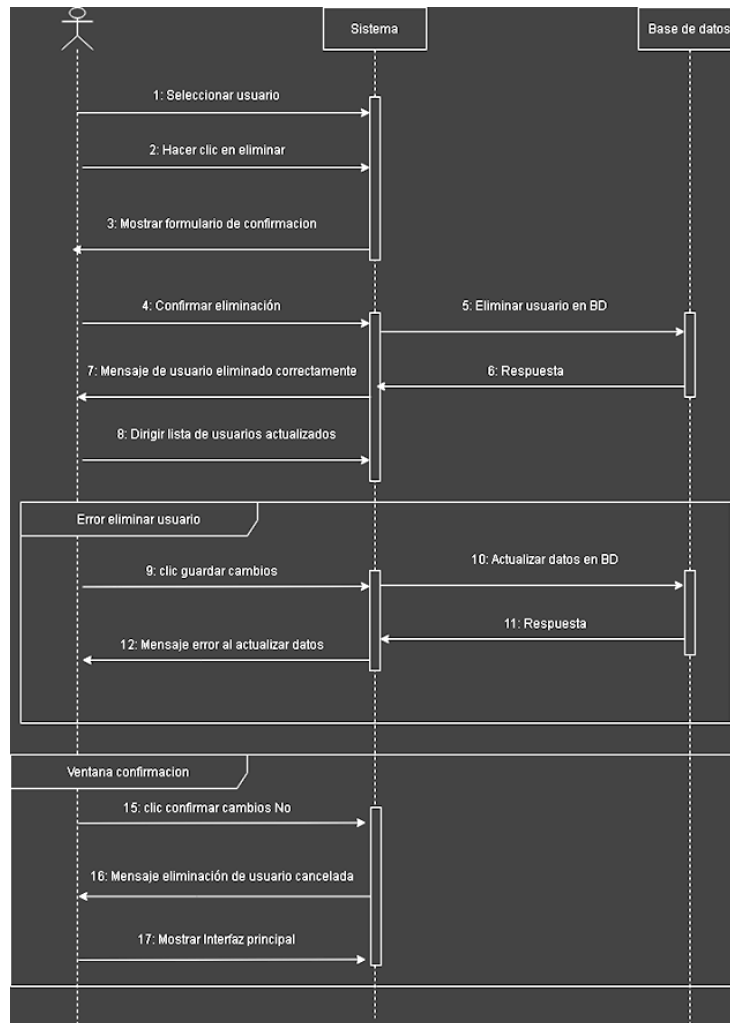


Figura 34 Diagrama de Secuencia Eliminar Usuarios

1.10 Diagrama de secuencia Generación de alertas

El diagrama evidenciado ilustra de manera clara la secuencia describe el flujo para que el sistema genera alertas automáticas cuando los datos recibidos de los sensores IoT superen los umbrales configurados, ilustra de manera detallada cómo el sistema verifica los datos, genera alertas, y las envía al módulo de notificaciones para que sean registradas y visualizadas por el usuario y al mismo tiempo contempla el escenario de fallo en la conexión con los sensores, proporcionando retroalimentación al usuario.

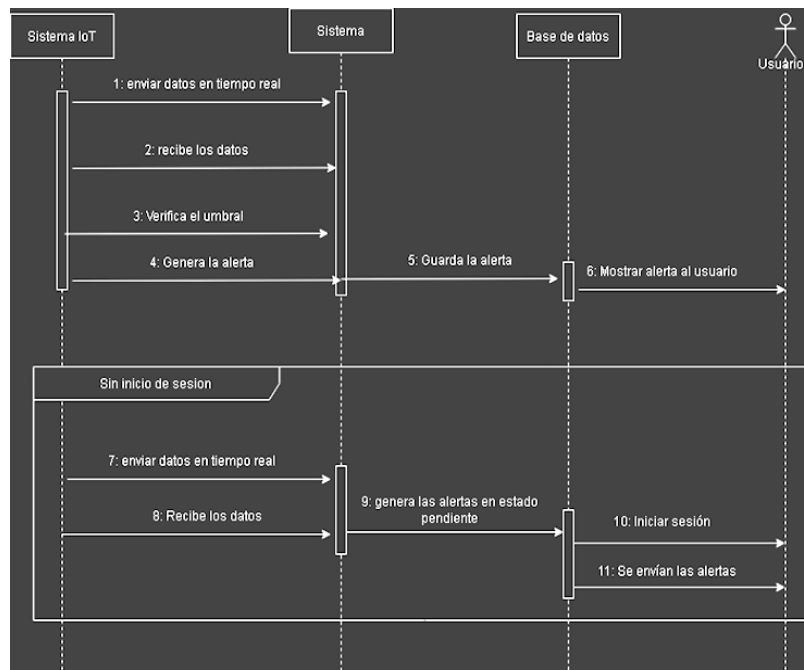


Figura 35 Diagrama de Secuencia Generación de Alertas

1.11 Diagrama de secuencia Notificaciones de alertas

La secuencia ilustrada describe el flujo para que el usuario reciba y gestione notificaciones de alertas generadas por el sistema, cómo el sistema registra las alertas, incrementa el contador de notificaciones, y cómo el usuario visualiza y gestiona estas notificaciones, y contempla el escenario en el que no hay alertas generadas, proporcionando retroalimentación al usuario.

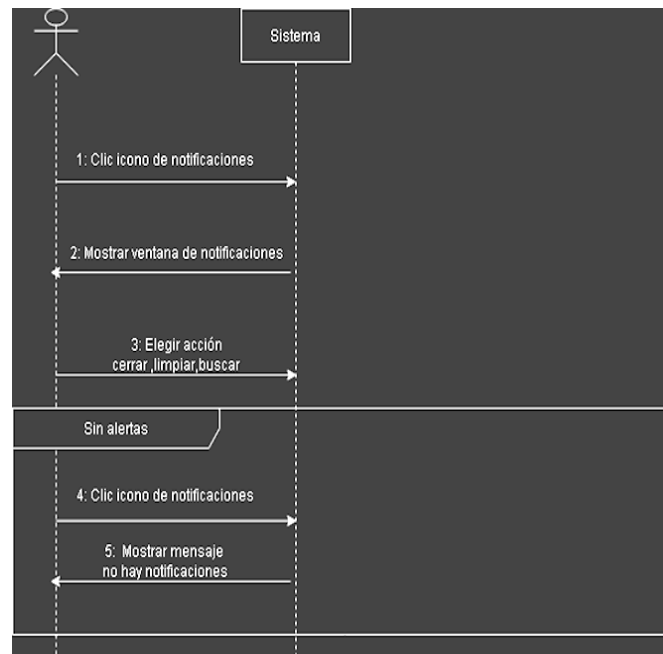


Figura 36 Diagrama de Secuencia Notificación de Alertas

1.12 Diagrama de secuencia seleccionar sensores

La secuencia de la imagen adjunta - es decir, la que se encuentra encima de este enunciado - describe el flujo para que el usuario seleccione los sensores que desea monitorear en tiempo real y recibir alertas cuando los valores superen los umbrales configurados. Detalla cómo el usuario accede a la lista de sensores, realiza la selección, y cómo el sistema registra y aplica esta configuración. También contempla escenarios de error, como la falta de sensores disponibles o la omisión de selección, proporcionando retroalimentación al usuario en cada caso.

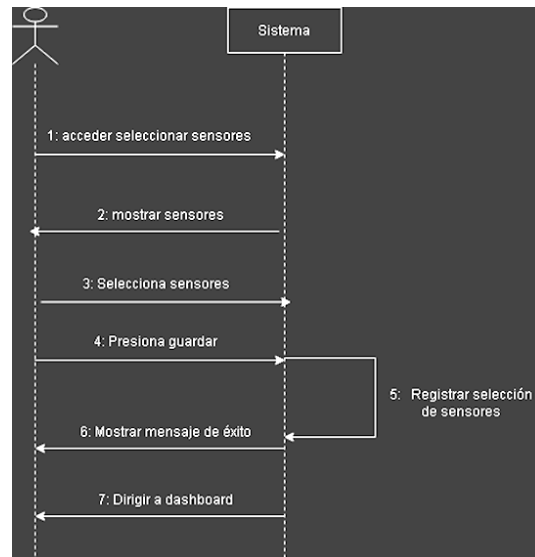


Figura 37 Diagrama de Secuencia Seleccionar Sensores

2. Diseño de Mockups

Esta etapa es crucial, ya que proporciona una representación clara de la interfaz de usuario y ayuda a identificar posibles mejoras o ajustes en el diseño y la experiencia del usuario. Los mockups actúan como una guía visual para el equipo de desarrollo y aseguran que el equipo tenga una comprensión común del aspecto y funcionalidad que se espera de la aplicación.

Para este proyecto, se realizaron los mockups de los diferentes módulos del aplicativo web, basándose en los requerimientos previamente definidos. Los mockups fueron diseñados en Balsamiq. (*Coming soon: Balsamiq Wireframes - Balsamiq Company News*, 2024)



Figura 38 Logo de Balsamiq

Enlace: (<https://balsamiq.cloud/saf74oo/pna5b4u>)

Una herramienta de diseño colaborativa basada en la nube que permite crear interfaces de usuario. Los principales mockups realizados para la aplicación son:

2.1 Diseño de interfaz inicial

Esta es la primera pantalla que verá el usuario al abrir la aplicación. En ella se ofrecen opciones como el inicio de sesión mediante correo y contraseña, la opción de registrarse y la opción de recuperar contraseña.

The mockup shows a light gray background with a white rounded rectangle in the center. At the top of the rectangle is the title 'Inicio de Sesión'. Below the title are two input fields: the first is labeled 'Correo' and the second is labeled 'Contraseña'. Below these fields is a green button with the text 'Iniciar Sesión'. At the bottom of the white rectangle, there are two lines of text: '¿No tienes una cuenta? [Regístrate aquí](#)' and '¿Recuperar Contraseña? [Recuperar aquí](#)'.

Figura 39 Interfaz Inicio de Sesión

2.2 Diseño de la interfaz recuperar contraseña

Esta interfaz le va a permitir al usuario recuperar su contraseña de su cuenta en caso de olvido, para poder tener acceso nuevamente en el aplicativo web esto usando la verificación del correo electrónico y envió de un enlace para el restablecimiento.

Figura 40 Interfaz Recuperar Contraseña

2.3 Diseño de la interfaz de registro

Esta pantalla permite al usuario registrarse en el aplicativo web, Esta cuenta con varios campos básicos pero esenciales que deben ser completados para llevar a cabo el proceso de registro correctamente y hacer uso de la web.

inicia sesión aquí'."/>

Figura 41 Interfaz Registro de Usuario

2.4 Diseño de la interfaz principal Dashboard

Esta es la pantalla principal que verá el usuario una vez que inicie sesión en el aplicativo web correctamente como es una web donde el punto primordial es dar prioridad a las variables de la calidad del agua en esta se mostrará el Dashboard interactivo hecho con Grafana, el cual permite al usuario poder entender de una manera más detallada el comportamiento de los datos, además cuenta una barra de tiempo donde podrá seleccionar qué cantidad de datos desea filtrar esto en base en el tiempo de 5 minutos para recibir los datos en tiempo real , 10 minutos , 1 hora , 3 horas entre otros más para ver el comportamiento de los datos en un rango de tiempo especificado, también se podrá ver el icono de las notificaciones que se irán incrementando en base a como las alertas se van generando al dar clic se mostrará una ventana en donde se mostrarán las

notificaciones son sus valores teniendo la opción de cerrar la ventana de notificaciones o limpiar la lista de ellas.

También con una opción para poder desplegar el menú principal desde el cual podrá tener acceso a las demás funcionalidades de la aplicación web. Cabe recalcar que el Dashboard se implementa en la parte superior unos campos en donde los datos van a ir cambiándose en tiempo real en base a cómo van llegando también al hacer clic en alguno de ellos se desplegará un Dashboard donde las variables están todas juntas para poder filtrar y entender un poco más su comportamiento de la variable seleccionada.

En el Dashboard se implementó colores personalizado en base a los datos para poder determinar el rango de ellos los cuales son:

- **Verde:** Los datos que están dentro del rango establecido.
- **Amarillo:** Los datos que están dentro del rango, pero en estado de advertencia.
- **Rojo:** Los datos que presentan anomalías y se debe tomar medidas al respecto.

Mas en la parte Inferior de la interfaz se podrá ver las gráficas correspondientes a cada variable de la calidad del agua de las cuales se están obteniendo los datos.

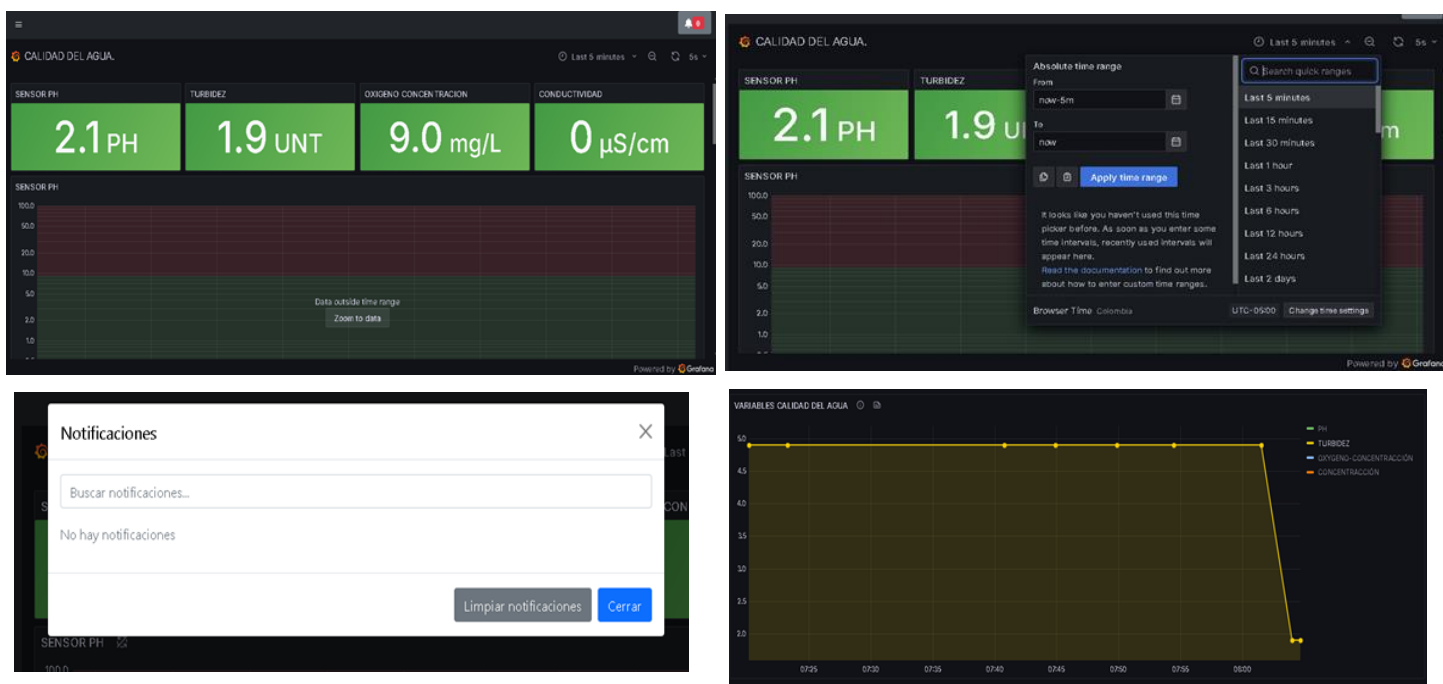


Figura 42 Interfaz Principal Dashboard

2.5 Diseño de la interfaz en tiempo real:

Esta ventana es donde el usuario en caso de no visualizar el Dashboard y quisiera ver más con exactitud o de una forma más segura y actualizada. A diferencia del Dashboard.

Esto lo que hace es garantizar una visualización inmediata y exacta de los datos que van llegando desde el hardware IoT correspondientes a las variables de la calidad del agua.



Figura 43 Interfaz Tiempo Real

2.6 Diseño de la interfaz Tabla de datos críticos:

En esta sección se visualizarán los datos críticos que se actualizarán en tiempo real sin tener que recargar la página, estos datos son los que salieron del umbral establecido visualizando datos de las variables y la fecha y hora exacta en la que se generó cada alerta. Esto lo que hace es proporcionar un registro más detallado y sigiloso además seguro de los eventos importantes que se deben considerar tomar prioridad. Esta tabla está diseñada para mostrar información de una manera ordenada y cuenta con un sistema de paginación, lo que lo hace accesible para una navegación fácil a través de grandes volúmenes de datos, asegurando que el usuario pueda acceder a toda la información sin ningún inconveniente, para ir a esta interfaz podemos encontrar la opción “Datos críticos” en el menú principal.

PH	Turbidez	Oxígeno	Conductividad	Fecha
5.1	4.9	0	0	8/2/2025, 8:01:33 a.m.
5.1	4.9	0	0	8/2/2025, 7:54:27 a.m.
5.1	4.9	0	0	8/2/2025, 7:49:51 a.m.
5.1	4.9	0	0	8/2/2025, 7:44:53 a.m.
5.1	4.9	0	0	8/2/2025, 7:40:45 a.m.

Figura 44 Interfaz Tabla Datos Críticos

2.7 Diseño de la interfaz Lista de usuario Registrados:

En esta interfaz se visualiza una lista completa de todos los usuarios registrados en la aplicación web. Como es un sistema personal los usuarios podrán gestionar los perfiles de manera eficiente. Los usuarios también tienen la posibilidad de editar la información de cualquier usuario registrado, permitiendo modificar datos como nombre, edad, correo electrónico. Además, la interfaz ofrece la opción de eliminar usuarios de la plataforma, lo que garantiza un control completo sobre la gestión de la base de datos de usuarios.

Para mayor comodidad, la información está organizada de forma clara y accesible, mediante una tabla con los datos de cada usuario.

La interfaz está diseñada para asegurar un manejo seguro y eficiente de la información, brindando a los Usuarios todas las herramientas necesarias para gestionar el acceso y las interacciones dentro de la plataforma.

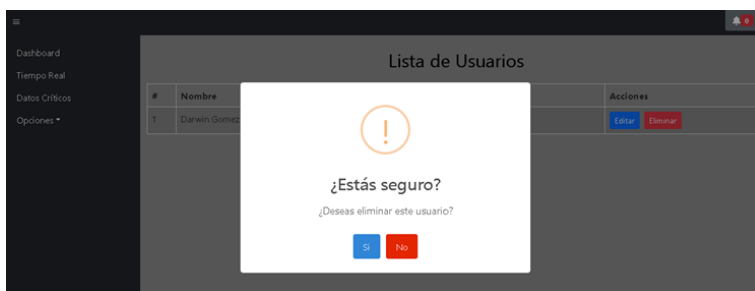


Figura 45 Eliminar Lista De Usuarios Registrados

2.8 Diseño de la interfaz Editar usuario:

La ventana "Editar Usuario" permite a los usuarios autenticados editar la información de la cuenta dentro de la plataforma de una forma segura y fácil. Como se trata de un sistema personalizado, cada usuario también podrá editar la propia información del perfil. La interfaz está diseñada con un enfoque en la usabilidad y la seguridad, garantizando que los datos sean actualizados de forma eficiente, con las validaciones necesarias para evitar errores o accesos no autorizados.



Figura 46 Interfaz Editar Usuario

2.9 Diseño de la interfaz Cambiar Contraseña

La Ventana "**Cambiar Contraseña**" se implementó para proporcionar a los usuarios autenticados un proceso seguro y eficiente para poder cambiar sus credenciales de acceso. Su diseño sigue los principios de seguridad, en el poder garantizar que el proceso de cambio de la contraseña se haga de manera sutil y protegida contra la inseguridad. Esta opción se encuentra ubicada dentro del menú principal, en la sección "**Opciones**", permitiendo una entrada rápida sin necesidad de navegar por múltiples

pantallas. Esto mejora la experiencia del usuario al simplificar el proceso de actualización de contraseña, sin comprometer la seguridad del sistema.

2.10 Diseño de la interfaz Seleccionar sensores:

La interfaz "Seleccionar Sensores" permite a los usuarios elegir qué parámetros desean monitorear en tiempo real dentro del sistema. Está diseñada con un enfoque en usabilidad, eficiencia y personalización, garantizando que cada usuario pueda habilitar o deshabilitar sensores según sus necesidades. Este módulo optimiza la visualización y el procesamiento de datos al permitir que solo los sensores seleccionados envíen información, evitando la sobrecarga de datos innecesarios en la interfaz.

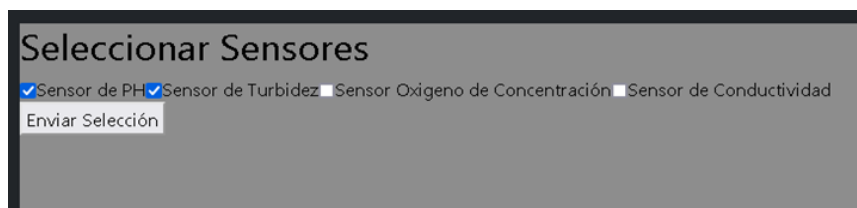


Figura 47 Interfaz Seleccionar Sensores

2.11 Diseño de la interfaz Alerta:

La ventana de alerta es generada por el aplicativo web, esto hace referencia a que cuando se detecta que de los valores enviados desde el dispositivo IoT superaron el umbral establecido, automáticamente usando socket se genera una alerta para el usuario, donde se va a visualizar los valores de los sensores que el usuario seleccionó, en caso de que los valores no superen el umbral, solo se guardará los valores en la base de datos para visualizarlos en el Dashboard muy externo a la alerta, también mencionar que no se mostrará una alerta porque ningún valor recibido ha superado el umbral.

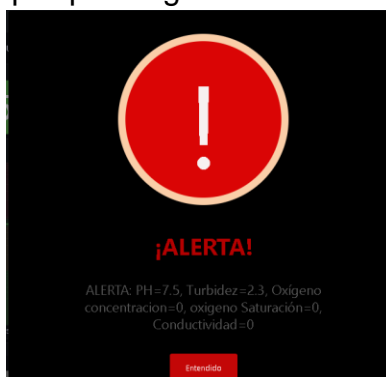


Figura 48 Alerta

2.12 Diseño de la interfaz Cerrar sesión:

Esta sesión es para que el usuario pueda cerrar sesión del aplicativo web de forma segura, eliminando cualquier token o cookie generado durante el inicio de sesión, esto le permite al usuario asegurar la cuenta, y no permitir el ingreso de personas no autorizadas

también al momento de cerrar sesión y navegar a alguna interfaz del aplicativo web, se dirigirá a la interfaz de iniciar sesión, sin permitir el ingreso sin antes haber creado una cuenta, o haber iniciar sesión correctamente.

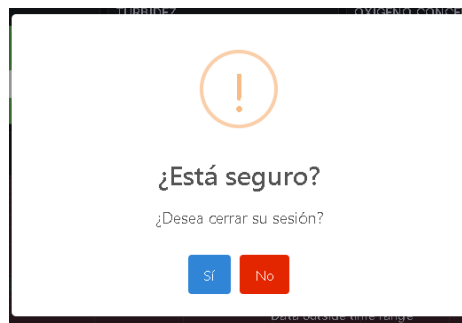


Figura 49 Cerrar Sesión

3. Estructura del proyecto

3.1 Arquitectura del sistema IoT

El sistema cuenta con una red de sensores IoT estos son los encargados de medir la calidad del agua en tiempo real. Estos sensores se encargan de recopilar los datos que son:

- **pH:** Evalúa la acidez o alcalinidad del agua.
- **Turbidez:** Indica la cantidad de partículas suspendidas en el agua.
- **Oxígeno de concentración:** Determina la cantidad de oxígeno disuelto, fundamental para la vida acuática.
- **Conductividad:** Mide la capacidad del agua para conducir electricidad, indicando la presencia de sales disueltas.

Los datos de estos sensores son leídos por un microcontrolador **Arduino Vernier**, el cual se encarga de procesarlos con el módulo SIM7600X 4G HAT. Se envían al servidor, cabe mencionar que la comunicación con el respectivo servidor se usa el protocolo HTTP, enviando los datos por medio de comandos AT enviando en un formato JSON para su posterior almacenamiento y visualización.

3.2 Proceso de envío de datos

3.2.1 Captura de datos: Los sensores generan Lectura analógica

```
void loop() {  
  // Leer el valor del sensor  
  float localphvalue = analogRead(A2) * (5.0 / 1023.0);  
  float localTurbValue = analogRead(A0) * (5.0 / 1023.0);  
  float oxygenVoltage = analogRead(A2) * (5.0 / 1023.0);  
  int localConductivityValue = analogRead(A0);  
}
```

Figura 50 Proceso Envío de Datos

3.2.2 Fórmulas y Conversión: Aquí se aplican fórmulas para la conversión del voltaje en datos representativos.

```
// Convertir valores de otros sensores
globalPHValue=ConvertirAph(localphvalue);
globalTurbValue = convertirATurbidez(localTurbValue);
globalOxygenConcentration = convertirAConcentracion(oxygenVoltage);
globalConductivity = calcularConductividad(localConductivityValue);
```

Figura 51 Formulas y Conversión

Conversión de pH: La siguiente imagen conversión de PH muestra la función en C++, esta lo que hace es convertir el voltaje del sensor pH Vernier en un valor de pH en el que se utiliza la ecuación oficial de vernier proporcionada en su librería. (*pH Sensor User Manual*, 2024b)

$$\text{pH} = -3.78 \times \text{voltaje} + 13.52$$

Este cálculo es fundamental ya que el sensor entrega una señal analógica en voltios, que debe ser transformada para conseguir la medición de pH.

```
//PH
float ConvertirAph(float localphvalue)
{
    float voltage = localphvalue;
    Serial.println("voltaje ph");
    Serial.println(voltage);
    float ph=-3.78*voltage+13.52;
    return ph ;
}
```

Figura 52 Formula Conversion de pH

Conversión Turbidez: La siguiente imagen Conversion de Turbidez una función que se encarga de transformar el voltaje detectado por el sensor, en una unidad estándar de turbidez (NTU) se aplica la fórmula de conversión oficial por Vernier. .

(*Turbidity Sensor User Manual*, 2024b)

$$\text{turbidez} = (150.00 \times \text{voltaje}) + (-80.00)$$

Donde el valor es normalizado y se escala dentro de un intervalo de 0 a 5 donde se emplea la función map() , se aplica una restricción para que el valor no exceda el límite para una mejor medición

```

float convertirATurbidez(float localTurbidezvalue)
{
    float voltage = localTurbidezvalue;
    //fórmula calcular la turbidez en NTU
    float turbidez = (150.00 * voltage + (-80.00));
    // turbidez en el rango de 0 a 5.0
    float turbidezEscalada = map(turbidez * 100, 0, 100000, 0, 500) / 100.0;
    //valor dentro del rango [0, 5.0]
    if (turbidezEscalada < 0) turbidezEscalada = 0.0;
    if (turbidezEscalada > 5.0) turbidezEscalada = 5.0;
    return turbidezEscalada;
}

```

Figura 53 Formula Conversion de Turbidez

Conversión de conductividad: La siguiente imagen Conversion de Conductividad es una función que se encarga de transformar un valor analógico del sensor un valor eléctrico de Conductividad, se hace la Conversion de la señal usando la fórmula de vernier.(*Conductivity Probe User Manual*, 2024b)

$$\text{Conductividad} = 960.00 * \text{voltage}$$

Esta medición es obligatoria para poder conseguir las mediciones en unidades de conductividad en el cual se basa en la cantidad de iones disueltos.

```

//Conductividad
float calcularConductividad(float localConductivityValue)
{
    // Convertir el valor analógico a voltaje
    float voltage=localConductivityValue * (5.0 / 1023.0);
    Serial.println(voltage);
    float conductividad=960.00*voltage;
    return conductividad;
}

```

Figura 54 Formula Conversion de Conductividad

Conversión Oxígeno de concentración: La siguiente función transforma el voltaje del sensor de oxígeno de concentración la formula por vernier es la siguiente.

$$\text{oxigeno} = 4.44 * \text{voltaje} + (-0.44)$$

Esta relación facilita la Conversion en una medición estandarizada la cual es importante para valorar la calidad del agua. (*Vernier Optical DO Probe User Manual*, 2024b)

```

//OXIGENO
float convertirAConcentracion(float oxygenVoltage) {
    float voltage=oxygenVoltage;
    float oxigeno=4.44*voltage+(-0.44);
    return oxigeno;
}

```

Figura 55 Formula Conversion de Oxigeno

3.2.3 Serialización JSON: Los datos se estructuran en un objeto JSON antes de ser enviados al servidor, esto es para redondear el valor y ajustarlo a un decimal de precisión lo que estandariza los valores y mejora la legibilidad de los datos enviados.

```
doc["sensorPH"] = round(globalPHValue * 10) / 10.0;
doc["sensorTurbidez"] = round(globalTurbValue * 10) / 10.0;
doc["sensorOxygenConcentration"] = round(globalOxygenConcentration * 10) / 10.0;
doc["sensorConductividad"] = round(globalConductivity * 10) / 10.0;

// Serializar JSON y enviar al servidor
String jsonString;
serializeJson(doc, jsonString);
```

Figura 56 Serialización Json

3.2.4 Transmisión de datos: Se envía los datos al servidor a través de una petición HTTP POST en el que se emplea comandos AT, esta serie de comandos establece la conexión, ajusta y transmite el objeto JSON con los datos de los sensores.

```
sendCommand("AT+HTTPINIT");
delay(2000);
sendCommand("AT+HTTPPARA=\"URL\", \"https://d3nm9bfn-3100.use2.devttunnels.ms/api/1.0/sensores_vista\");
delay(2000);
sendCommand("AT+HTTPPARA=\"CONTENT\", \"application/json\");
delay(1000);
sendCommand("AT+HTTPDATA=" + String(jsonString.length()) + ",10000");
delay(2000);
// Mostrar JSON en el monitor serial
Serial.println(jsonString);
delay(2000);
sendCommand("AT+HTTPACTION=1");
delay(2000);
sendCommand("AT+HTTPREAD=0,100");
sendCommand("AT+HTTPTERM");
delay(2000);
```

Figura 57 Transmisión de Datos

3.2.5 Ejemplo de datos enviados:

Cada lectura de en base a los sensores se crea en una estructura JSON. Estos valores se actualizan constantemente y se reflejan en tiempo real en la aplicación web.

```
1
2 {
3   "sensorPH":10.9,
4   "sensorTurbidez":44,
5   "sensorOxygenConcentration":22,
6   "sensorOxygenSaturacion":11,
7   "sensorConductividad":33
8 }
```

Figura 58 Ejemplo de Envío de Datos

3.3 Estructura aplicación web

El aplicativo web está desarrollado bajo una arquitectura basada en **Cliente-Servidor**, en el cual se utilizan tecnologías modernas para poder respaldar flexibilidad, modularidad y eficiencia. Esto se compone de dos principales capas:

Backend: Esta capa está implementado con Node.js y Express, en el cual se sigue el patrón Modelo. Controlador y ruta. Este se encarga de gestionar la lógica del negocio, las operaciones sobre la base de datos y la autenticación mediante JWT.

Frontend: Esta capa está desarrollada con React y Vite, en el que se proporciona una interfaz dinámica e interactiva para los usuarios. Esto consume los datos del backend a través de una API REST.

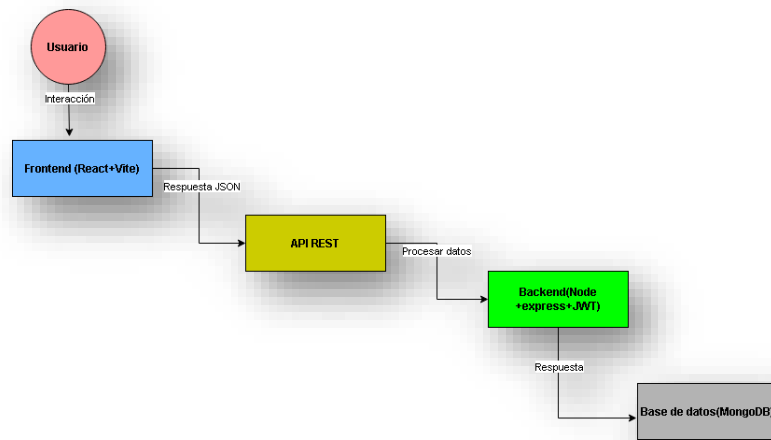


Figura 59 Arquitectura del Modelo Backend Frontend

Backend Servidor Node.js y Express: La estructura del backend sigue el patrón para organizar la lógica de la aplicación de manera modular.

La organización de carpetas y archivos dentro del backend es la siguiente:

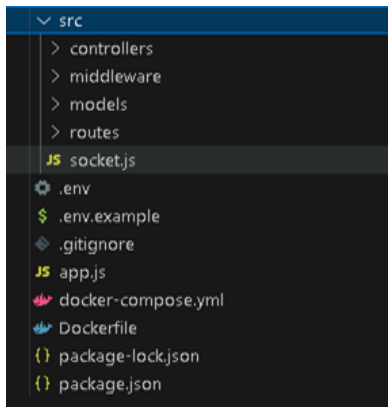


Figura 60 Estructura del Backend

- **Controllers:** Controladores con la lógica del aplicativo web.
- **middleware:** para validaciones y seguridad.
- **Models:** Modelos de los datos en MongoDB.
- **Routes:** Definición de las rutas de las APIS.
- **Socket.js:** La configuración de WebSockets con Socket.IO.
- **app.js:** Punto de entrada del aplicativo web.
- **.env:** Son las variables de entorno.
- **.env.example:** Es la plantilla de las variables de entorno.
- **.gitignore:** Archivos ignorados en git.
- **docker-compose.yml:** Configuración de Docker.
- **Dockerfile:** Instrucciones para el contenedor.
- **package.json:** Dependencias y configuración del backend.
- **package-lock.json:** Registro de versiones de dependencias.

3.4 Componentes Claves del Backend

A continuación, se presentan los principales módulos del backend y su funcionalidad dentro del sistema.

3.4.1 Conexión a la base de datos

El sistema utiliza MongoDB como base de datos NoSQL, lo que permite almacenar y gestionar datos de manera flexible y escalable. A diferencia de bases de datos relacionales, MongoDB utiliza documentos en formato JSON, lo que facilita la integración con Node.js y Express.

```
mongo.js x
Proyecto_Sena_2024_Calidad_Del_Agua > config > mongo.js > ...
Darwin Gomez, 3 weeks ago | 2 authors (Darwin Gomez and one other)
1 //Módulo para gestionar la conexión a la base de datos MongoDB
2 const mongoose = require('mongoose');
3
4 //Implementación de función asíncrona para la conexión a la base de datos
5 //En la variable de entorno esta URLBASEDEDATOS donde especifica la base de datos si es local o en la nube
6 //En caso de no encontrar la URLBASEDEDATOS se muestra un mensaje de error
7 //Se implementa un try catch para la conexión a la base de datos Darwin Gomez, 3 weeks ago • Comentar
8 // Si la conexión falla se intenta reconectar después de 5 segundos.
9 const conexionabasededatos=async()->
10 {
11   const URLBASEDEDATOS = process.env.URLBASEDEDATOS;
12   if (!URLBASEDEDATOS) {
13     console.error("La url de la base de datos no se encuentra ");
14     process.exit(1);
15   }
16   try
17   {
18     await mongoose.connect(URLBASEDEDATOS);
19     console.log("Conexión Correcta a la base de datos");
20   } catch (error)
21   {
22     console.error("Problemas de conexión en la base de datos", error);
23     setTimeout(conexionabasededatos, 5000);
24   }
25 };
26
27 //En todas encuentras este código el cual nos ayuda a exportar las funciones definidas en cada archivo
28 module.exports = {conexionabasededatos};
```

Figura 61 Conexión a La Base de Datos

3.4.2 Modelo de usuarios

El modelo de usuarios almacena la información de los usuarios registrados en el sistema, incluyendo su nombre, edad, correo electrónico y contraseña.

correo: Campo único para evitar registros duplicados.

Password: Almacenado de manera segura para autenticación.


```

usuarioModel.js X
Proyecto_Sena_2024_Calidad_Del_Agua > src > models > usuarioModel.js > ...
Darwin Gomez, 3 weeks ago | 1 author (Darwin Gomez)
1 //Modelo de datos para los usuarios en la base de datos MongoDB.
2 //Define la estructura y restricciones de los documentos de usuario.
3 const mongoose=require('mongoose')
4 //Esquema de usuario con las propiedades y validaciones necesarias.
5 const EsquemaUsuario=new mongoose.Schema({
6   //El nombre es obligatorio
7   nombre:{
8     type:String,
9     required: true,
10  },
11   //La edad es obligatoria
12   edad:{
13     type:Number,
14     required: true,
15  },
16   //El correo debe ser único
17   //Es un campo obligatorio
18   //Se almacena en minúsculas
19   correo:{
20     type:String,
21     unique: true,
22     required: true,
23     lowercase: true,
24  },
25   //La contraseña debe tener al menos 10 caracteres
26   //Es un campo obligatorio
27   password:{
28     type:String,
29     minlength: 10,
30     required: true,
31  }
32 })

```

Figura 62 Modelo de Usuario

3.4.1 Modelo de sensores aplicativo web

Cada sensor registrado en el sistema almacena sus mediciones en tiempo real, incluyendo sensorpH, sensor Turbidez, sensorOxygenConcentration, sensorOxygenSaturacion, sensor Conductividad.

```

sensoresvistaModel.js
Proyecto_Sena_2024_Calidad_Del_Agua > src > models > sensoresvistaModel.js > VistaSensores > sensorConductividad
Darwin Gomez, 3 weeks ago | 1 author (Darwin Gomez)
1 //Modelo de datos para la vista de sensores en la base de datos MongoDB.
2 //Define la estructura de los datos capturados por los sensores.
3 const mongoose = require("mongoose");
4 //Esquema de VistaSensores que almacena los datos de los sensores en la base de datos.
5 const VistaSensores = new mongoose.Schema(
6   {
7     // Valor del pH del agua
8     sensorPH: {
9       type: Number,
10    },
11    // Medida de la turbidez del agua
12    sensorTurbidez: {
13      type: Number,
14    },
15    // Concentración de oxígeno en el agua
16    sensorOxygenConcentration: {
17      type: Number,
18    },
19    // Nivel de saturación de oxígeno
20    sensorOxygenSaturacion: {
21      type: Number,
22    },
23    // Conductividad eléctrica del agua
24    sensorConductividad: {
25      type: Number
26    },
27  },
28  {
29    // Agrega automáticamente createdAt y updatedAt
30    timestamps: true,
31    // Desactiva el campo __v de versión de documentos
32    versionKey: false,
33  }
34);
35 //Exporta el modelo de vistaSensores basado en el esquema.
36 module.exports = mongoose.model("vistaSensores", VistaSensores);

```

Figura 63 Modelo de Sensores

- **sensorpH:** valor del pH del agua.
- **sensorTurbidez:** valor de turbidez del agua.
- **sensorOxygenoConcentration:** Valor de oxígeno de concentración del agua.
- **SensorOxygenoSaturacion:** Valor del oxígeno de saturación del agua.
- **SensorConductividad:** Valor de conductividad del agua.
- **createdAT:** fecha se almacena automáticamente al registrar una cadena de datos.

3.4.1 Controlador de usuario

El controlador es responsable de la lógica del negocio relacionada con la autenticación de usuarios y la gestión de credenciales. A continuación, se muestra un fragmento del controlador de autenticación:

```

const loginUsuario = async (req, res) => {
  try {
    // Se extraen el correo y la contraseña del cuerpo de la solicitud
    const { correo, password } = req.body;
    // Se normaliza el correo electrónico a minúsculas
    const correoNormalizado = correo.toLowerCase();
    // Se busca al usuario en la base de datos utilizando el correo electrónico normalizado
    const usuario = await usuariomodelo.findOne({ correo: correoNormalizado });
    // Si el usuario no existe se responde con un mensaje de error
    if (!usuario) {
      // Se responde con un código de estado 404 y un mensaje
      return res.status(404).json({ msg: "Usuario y/o contraseña incorrectos" });
    }
    // Se compara la contraseña ingresada con la contraseña almacenada en la base de datos
    const isMatch = await bcrypt.compare(password, usuario.password);
    if (!isMatch) {
      // Se responde con un código de estado 400 y un mensaje
      return res.status(400).json({ msg: "Usuario y/o contraseña incorrectos" });
    }
    // Se genera un token JWT con el ID del usuario y una expiración de 1 hora
    const token = jwt.sign(
      { usuarioId: usuario._id },
      process.env.JWT_SECRETO,
      { expiresIn: '1h' }
    );
    // Se responde con un mensaje de bienvenida el token de autenticación y los datos básicos del usuario
    res.json({
      token,
      usuario: { _id: usuario._id, nombre: usuario.nombre, correo: usuario.correo },
      msg: "Bienvenido"
    });
  } catch (e) {
    // Si ocurre un error, se maneja con la función 'httpError', que envía la respuesta correspondiente
    httpError(res, e);
  }
};

```

Figura 64 Controlador de Usuario

Este fragmento demuestra cómo se autentica a un usuario y se genera un token JWT para gestionar sesiones de manera segura.

3.4.1 Controlador de datos de los sensores

El controlador de sensores es un componente fundamental en el backend del sistema, ya que se encarga de recibir, procesar y enviar datos de sensores en tiempo real. Además, gestiona la generación de notificaciones automáticas en caso de detectar valores críticos, asegurando una respuesta rápida ante posibles anomalías.

```

const crearDatosSensor = async (req, res) => {
  try {
    // Se extraen los valores de los sensores desde el cuerpo de la solicitud.
    const { sensorPH, sensorTurbidez, sensorOxygenConcentration, sensorOxygenSaturation, sensorConductividad } = req.body;
    // Se obtiene la instancia de Socket.IO para la comunicación en tiempo real.
    const io = obtenerIo();
    // Los sensores están enviando datos, como el usuario aun no ha seleccionado que sensores desea ver o si aun no ha iniciado sesion
    // entonces los datos llegan a Map especificado anteriormente en el archivo socket.js
    // Si no hay usuarios conectados, se almacenan los datos como "pendientes" en la base de datos.
    if (obtenerSensoresSeleccionados().size === 0) {
      console.log("No hay usuarios conectados.");
      // Se almacena toda la información del sensor.
      // Se guarda la fecha en que se recibe la notificación.
      // Se marca como pendiente para procesarla después.
      await NotificacionModelo.create({
        datosSensor: req.body,
        fecha: new Date(),
        estado: "pendiente",
      });
      console.log("Notificación guardada en MongoDB su estado es pendiente por ahora.");
      return res.send({ message: "se almacenaron los datos como pendiente..." });
    }
    // Si hay usuarios conectados, se envían los datos en tiempo real a cada cliente.
    for (const [socketId, sensores] of obtenerSensoresSeleccionados().entries()) {
      // Se filtran los datos según los sensores seleccionados por cada usuario.
      const datosFiltrados = {
        sensorPH: sensores.includes("sensorPH") ? sensorPH : 0,
        sensorTurbidez: sensores.includes("sensorTurbidez") ? sensorTurbidez : 0,
        sensorConductividad: sensores.includes("sensorConductividad") ? sensorConductividad : 0,
        sensorOxygenConcentration: sensores.includes("sensorOxygenConcentration") ? sensorOxygenConcentration : 0,
        sensorOxygenSaturation: sensores.includes("sensorOxygenSaturation") ? sensorOxygenSaturation : 0,
      };
      const io = obtenerIo();
      // Se obtiene nuevamente la instancia de Socket.IO para enviar los datos a un cliente específico.
      io.to(socketId).emit("nuevodataosensor", datosFiltrados);
      // Se verifica si alguno de los sensores ha superado los umbrales críticos establecidos definidos en la tabla.
    }
  }
};

```

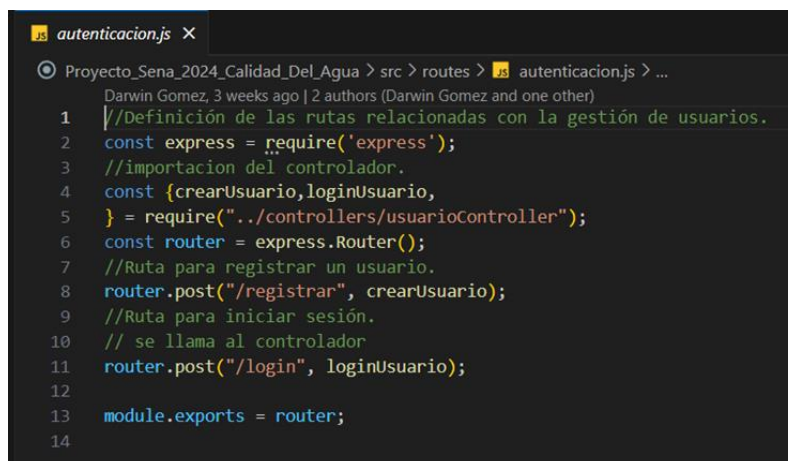
Figura 65 Controlador de Datos de Los Sensores

- **Recepción de datos** desde dispositivos IoT.
- **Verificación de sensores seleccionados** por cada usuario.
- **Almacenamiento de datos en MongoDB** en estado "pendiente" hasta ser procesados.
- **Generación de alertas automáticas** si los valores superan los umbrales establecidos.
- **Envío de datos en tiempo real con Socket.IO** a los clientes conectados.

Si los valores superan un umbral, se guarda la alerta en MongoDB y se notifica en tiempo real a los usuarios conectados.

3.4.1 Definición de rutas

Las rutas permiten que el frontend pueda comunicarse con el backend mediante endpoints RESTful. A continuación, se muestra la implementación de una ruta de autenticación:



```
autenticacion.js x
Proyecto_Sena_2024_Calidad_Del_Agua > src > routes > autenticacion.js > ...
Darwin Gomez, 3 weeks ago | 2 authors (Darwin Gomez and one other)
1 //Definición de las rutas relacionadas con la gestión de usuarios.
2 const express = require('express');
3 //importación del controlador.
4 const {crearUsuario,loginUsuario,
5 } = require("../controllers/usuarioController");
6 const router = express.Router();
7 //Ruta para registrar un usuario.
8 router.post("/registrar", crearUsuario);
9 //Ruta para iniciar sesión.
10 // se llama al controlador
11 router.post("/login", loginUsuario);
12
13 module.exports = router;
14
```

Figura 66 Definición de Rutas

Este fragmento expone un **endpoints** /login para que los usuarios puedan autenticarse en el sistema.

3.4.1 Comunicación en Tiempo Real con WebSockets

El sistema utiliza WebSockets para la transmisión de datos en tiempo real, permitiendo que los usuarios seleccionen qué sensores desean monitorear y reciban datos dinámicamente. Para garantizar la seguridad de la conexión, se implementa un middleware de autenticación con JWT (JSON Web Token) antes de permitir la comunicación entre el cliente y el servidor.

Middleware de Autenticación de WebSockets

Antes de establecer una conexión, el servidor verifica que el usuario envíe un **token JWT válido**. Si el token no es válido o ha expirado, la conexión se rechaza.

Manejo de Conexión de Usuarios

Una vez autenticado, el servidor gestiona la conexión del usuario y le permite seleccionar los sensores que desea monitorear.

```
//Middleware de autenticación de WebSocket.
//Verifica el token JWT antes de permitir la conexión.
io.use((socket, next) => {
  const token = socket.handshake.auth.token;
  console.log("Token recibido en el socket backend:");
  if (!token) {
    return next(new Error("Token no proporcionado."));
  }
  try {
    const user = jwt.verify(token, process.env.JWT_SECRET);
    socket.user = user;
    next();
  } catch (error) {
    console.log("Error durante la autenticación:", error.message);
    if (error.name === "TokenExpiredError") {
      return next(new Error("El token ha expirado"));
    } else {
      return next(new Error("El token proporcionado es inválido o ha expirado."));
    }
  }
});

io.on("connection", (socket) => {
  console.log("Usuario conectado:");
  socket.on("seleccionarSensores", (sensoresSeleccionados) => {
    //Evento para seleccionar sensores a monitorear.
    sensoresSeleccionadosPorCliente.set(socket.id, sensoresSeleccionados);
    //poder almacenar los sensores elegidos por cada cliente en un 'Map'
    //a la espera de que el cliente seleccione los sensores que desea monitorear y enviarselos
    obtenerNotificacionesPendientes(socket, sensoresSeleccionados);
  });
  //Este evento desconecta al usuario del socket es decir desconexión del usuario.
  //Elimina los sensores seleccionados cuando el cliente se desconecta.
  socket.on("disconnect", () => {
    console.log("Usuario desconectado:");
    sensoresSeleccionadosPorCliente.delete(socket.id);
  });
});
});
```

Figura 67 Comunicación en Tiempo Real WebSockets

3.4.1 Configuración y funcionamiento del servidor

El archivo app.js es el punto de entrada del backend de la aplicación. Se encarga de la configuración del servidor, la autenticación de usuarios, la conexión a la base de datos y la gestión de rutas.

Configuración y Dependencias

Para garantizar el correcto funcionamiento del servidor, se cargan las variables de entorno y se importan las librerías necesarias:

- **Express:** Framework para manejar las solicitudes HTTP.
- **HTTP:** Módulo para la creación del servidor.
- **CORS:** Permite que la API sea accesible desde diferentes orígenes.
- **Mongoose:** Maneja la conexión con la base de datos MongoDB.
- **Socket.IO:** Permite la comunicación en tiempo real.
- **JWT:** Middleware para la autenticación de usuarios.

Creación del Servidor y Configuración de Rutas

Se define el puerto en el cual se ejecuta la aplicación y se configuran las rutas principales de la API. Además, se habilita la **carpeta pública** donde se aloja el frontend. Se implementa un middleware que **protege el acceso** a la interfaz principal mediante **verificación de tokens JWT**.

Conexión con la Base de Datos

La aplicación se conecta a MongoDB mediante la función `conexionabasededatos()`, asegurando que el servidor solo inicie cuando la base de datos esté disponible.

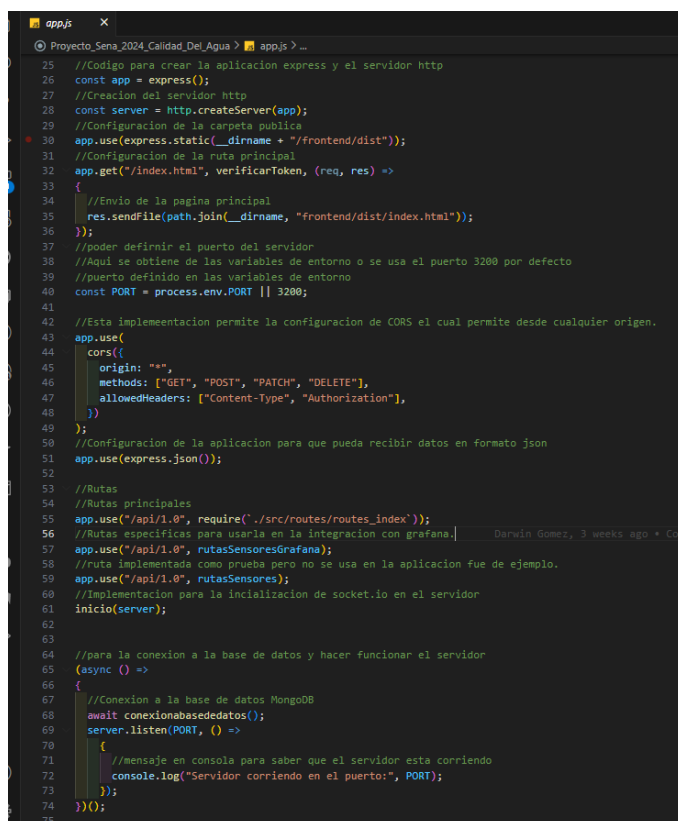
Integración con WebSockets

Para la comunicación en tiempo real, se configura **Socket.IO**, permitiendo que los usuarios reciban datos de sensores en tiempo real según sus preferencias.

Inicio del Servidor

Finalmente, el servidor se ejecuta en el puerto definido y muestra un mensaje en consola indicando que está en funcionamiento.

Esta configuración garantiza que la aplicación sea **modular, escalable y segura**, permitiendo la correcta interacción entre el backend, el frontend y los dispositivos IoT conectados.



```
25 //Codigo para crear la aplicacion express y el servidor http
26 const app = express();
27 //Creacion del servidor http
28 const server = http.createServer(app);
29 //Configuracion de la carpeta publica
30 app.use(express.static(__dirname + "/frontend/dist"));
31 //Configuracion de la ruta principal
32 app.get("/index.html", verificarToken, (req, res) =>
33 {
34 //Envio de la pagina principal
35 res.sendFile(path.join(__dirname, "frontend/dist/index.html"));
36 });
37 //poder definir el puerto del servidor
38 //Aqui se obtiene de las variables de entorno o se usa el puerto 3200 por defecto
39 //puerto definido en las variables de entorno
40 const PORT = process.env.PORT || 3200;
41
42 //Esta implementacion permite la configuracion de CORS el cual permite desde cualquier origen.
43 app.use(
44 cors({
45   origin: "*",
46   methods: ["GET", "POST", "PATCH", "DELETE"],
47   allowedHeaders: ["Content-Type", "Authorization"],
48 })
49 );
50 //Configuracion de la aplicacion para que pueda recibir datos en formato json
51 app.use(express.json());
52
53 //Rutas
54 //Rutas principales
55 app.use("/api/1.0", require("./src/routes/routes_index"));
56 //Rutas especificas para usarla en la integracion con grafana.
57 app.use("/api/1.0", rutasSensoresGrafana);
58 //ruta implementada como prueba pero no se usa en la aplicacion fue de ejemplo.
59 app.use("/api/1.0", rutasSensores);
60 //Implementacion para la inicializacion de socket.io en el servidor
61 inicio(server);
62
63
64 //para la conexion a la base de datos y hacer funcionar el servidor
65 (async () =>
66 {
67 //Conexion a la base de datos MongoDB
68 await conexionabasededatos();
69 server.listen(PORT, () =>
70 {
71 //mensaje en consola para saber que el servidor esta corriendo
72 console.log("Servidor corriendo en el puerto:", PORT);
73 });
74 })();
75
```

Figura 68 Configuración y Funcionamiento del Servidor

3.5 Tecnologías Utilizadas Backend

El backend utiliza las siguientes tecnologías para garantizar seguridad, escalabilidad y eficiencia:

- **Node.js y Express:** Framework ligero y rápido para la construcción de APIs RESTful.
- **MongoDB y Mongoose:** Base de datos NoSQL flexible y escalable.
- **JWT (JSON Web Token):** Implementación de autenticación segura.
- **Socket.IO:** Comunicación en tiempo real para notificaciones y datos dinámicos.
- **Grafana:** Visualización de datos de sensores en tiempo real.
- **Docker:** Contenedorización para facilitar despliegue en entornos de producción.

3.6 Estructura del frontend React y vite

El frontend de la aplicación está desarrollado con React y Vite, lo que permite una interfaz rápida y dinámica. Su principal función es consumir la API del backend, manejar la autenticación de usuarios y mostrar datos en tiempo real.

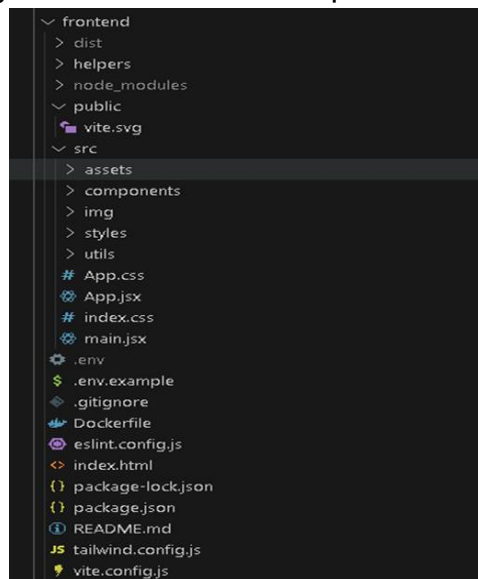


Figura 69 Estructura del Frontend React y Vite

- **dist:** Archivos compilados para producción.
- **Node_modules:** Dependencias de node.js
- **Public:** archivo publico icono.
- **Src:** código fuente principal.
- **Assets:** Recursos como fuentes imágenes.
- **Components:** Componente de las interfaces.
- **Img:** Almacenamiento imágenes de la aplicación.

- **Styles:** Archivo de estilos CSS.
- **Utils:** Funciones y utilidades reutilizables.
- **App.jsx:** Componente principal de la aplicación.
- **Main.jsx:** Punto de entrada de React.
- **App.css:** Estilos globales de la aplicación.
- **Index.css:** Estilos principales.
- **.env:** Variables de entorno.
- **Index.html:** Archivo HTML base.
- **Package.json:** Configuración y dependencias del proyecto.
- **Tailwind.config.js:** Configuración de Tailwind CSS.
- **Vite.config.js:** Configuración de vite.

3.7 Componentes Claves del frontend

3.7.1 Configuración de rutas y autenticación en el frontend

Aquí se definen las rutas principales, el frontend de la aplicación utiliza React Router para gestionar la navegación entre las diferentes vistas de la aplicación. Se implementa un sistema de rutas públicas y protegidas, asegurando que solo los usuarios autenticados puedan acceder a ciertas funcionalidades.

Para restringir el acceso a determinadas rutas, se implementa un **componente de ruta protegida (RutaProtegida)**, que valida si el usuario tiene un **token JWT** almacenado en **LocalStorage**.

Funcionamiento:

1. Si el token es válido, se permite el acceso.
2. Si no hay un token, el usuario es **redirigido automáticamente** a la página de inicio de sesión.

```
// es un objeto que contiene los elementos que envuelven un componente.
function RutaProtegida({ children })
{
  // Se obtiene el token de autenticación del almacenamiento local
  const autenticacionValida = localStorage.getItem('token');
  // Si el usuario está autenticado, se muestra el contenido, de lo contrario, se redirige a la página de inicio de sesión
  return autenticacionValida ? children : <Navigate to="/" />;
}
```

Figura 70 Configuración de Rutas y Autenticación en React

Toda la aplicación está encapsulada dentro del **componente ProveedorDeNotificaciones**, lo que permite gestionar **alertas en tiempo real** cuando ocurren eventos relevantes como datos críticos de sensores.


```

// Se encapsula la aplicación dentro de ProveedorDeNotificaciones para manejar alertas y n
<ProveedorDeNotificaciones>
  <Router>
    <Routes>
      /* Ruta pública para la pantalla de inicio de sesión */
      <Route path="/" element=<login /> />
      /* Ruta pública para la página de registro */
      <Route path="/registrar" element=<Registro /> />
      /* Ruta pública para la restauración de contraseña con un token dinámico */
      <Route path="/restablecercontrasena/:token" element=<Restablecercontrasena /> />
      /* Rutas protegidas: Solo accesibles si el usuario está autenticado */
      <Route
        path="/"
        element={
          <RutaProtegida>
            <MiComponente />
          </RutaProtegida>
        }
      />
      /* Subrutas protegidas dentro del menú principal, menu principal es padre , y los demás
      <Route path="micomponente" element=<MiComponentes /> />
      <Route path="dashboard" element=<Dashboard /> />
      <Route path="datoscriticos" element=<TablaDatosCriticos /> />
      <Route path="/usuario" element=<ListaUsuarios /> />
      <Route path="/editarusuario/:id" element=<EditarUsuario /> />
      <Route path="/perfil/:id" element=<MostrarUsuario /> />
      <Route path="seleccionarsensores" element=<Seleccionarsensores /> />
      <Route path="/cambiarcontrasena" element=<Cambiarcontrasena /> />
    </Routes>
  </Router>
</ProveedorDeNotificaciones>

```

Figura 71 Encapsulamiento de La Aplicación

3.7.1 Gestión de la conexión con el servidor mediante Socket.IO

La aplicación utiliza Socket.IO para la comunicación en tiempo real con el servidor, asegurando que los datos se actualicen dinámicamente sin necesidad de recargar la página. Para ello, se implementa un módulo que gestiona la conexión, autenticación y reconexión automática en caso de fallos.

Antes de establecer la conexión, el sistema verifica que el usuario tenga un token de autenticación válido. Si el token no está presente o ha expirado, el usuario es redirigido automáticamente a la página de inicio de sesión. Esta validación garantiza que solo usuarios autenticados puedan interactuar con los datos en tiempo real.

El proceso de conexión con el servidor se realiza mediante la función **conectarSocket()**, la cual inicializa la comunicación y configura parámetros esenciales como la reconexión automática en caso de interrupciones, el uso exclusivo del protocolo **WebSockets** y la implementación de un mecanismo de ping para mantener activa la conexión. Además, se registran eventos que permiten detectar la conexión **connect**, manejar errores **connect_error** y capturar desconexiones **disconnect**.

En caso de que la conexión falle debido a un token expirado, el sistema elimina la sesión del usuario y lo notifica con un mensaje de alerta, evitando así el acceso no autorizado. Para evitar la creación de múltiples instancias innecesarias del socket, el sistema verifica si una conexión ya está activa antes de establecer una nueva. También se proporciona una función para obtener la instancia del socket en cualquier momento sin necesidad de reconectar, así como un método para desconectarlo manualmente cuando sea necesario.

```

Proyecto_Sena_2024_Calidad_Del_Agua > frontend > src > utils > socket.js > conectarSocket
const API_URL = import.meta.env.VITE_API_URL;
import { io } from "socket.io-client";
import { verificarToken } from "../autenticacion";
let socket;
export const conectarSocket = () => {
  const token = verificarToken();
  if (!token) {
    console.error("No se puede conectar al socket sin un token.");
    return null;
  }
  if (socket && socket.connected) {
    console.log("Socket ya conectado");
    return socket;
  }
  socket = io(`${API_URL}`, {
    auth: { token },
    transports: ["websocket"],
    reconnection: true,
    reconnectionAttempts: Infinity,
    reconnectionDelay: 1000,
    pingTimeout: 60000,
    pingInterval: 25000,
  });
  socket.on("connect", () => {
    console.log("Socket conectado con ID:", socket.id);
  });
  socket.on("connect_error", (err) => {
    console.error("Error al conectar:", err.message);
    if (err.message === "El token ha expirado") {

```

Figura 72 Conexión con El Servidor Mediante Socker.io

3.7.1 Implementación de inicio de sesión

El módulo de autenticación en React permite la validación segura de credenciales, gestionando el inicio de sesión mediante una solicitud al backend. Si las credenciales son correctas, se almacena el token **JWT** en **LocalStorage**, se establece conexión con WebSockets y se redirige al usuario al Dashboard.

En caso de error, se notifican las fallas mediante **SweetAlert2**. Además, se integra una función de recuperación de contraseña que envía un enlace de restablecimiento al correo del usuario. El diseño modularizado con CSS optimiza la presentación del formulario, asegurando una interfaz clara y funcional. Esta implementación garantiza una autenticación eficiente y segura, optimizando la experiencia del usuario.

```

const IniciarSesion = () => {
  const [correo, establecerCorreo] = useState('');
  const [contrasena, establecerContrasena] = useState('');
  const [mensaje, establecerMensaje] = useState('');
  const navegar = useNavigate();

  const manejarEnvio = async (evento) => {
    evento.preventDefault();
    const datosFormulario = {
      correo,
      password: contrasena,
    };
    try {
      const respuesta = await fetch(`${API_URL}/api/1.0/autenticacion/login`, {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(datosFormulario),
      });
      const datos = await respuesta.json();
      if (datos.token) {
        localStorage.setItem('token', datos.token);
        localStorage.setItem('usuarioId', datos.usuario_id);
        Swal.fire({
          title: '¡Éxito!',
          text: 'Inicio de sesión exitoso',
          icon: 'success',
          timer: 1500,
          showConfirmButton: false,
        }).then(() => {
          conectarSocket(datos.token);
          navegar('/dashboard');
        });
      } else {
        // ...
      }
    }
  };
};

```

Figura 73 Implementación de Inicio de Sesión

3.7.1 implementación del Dashboard

El Dashboard permite la visualización en tiempo real de la calidad del agua mediante un iframe embebido que carga un panel de control alojado externamente. Este componente asegura una integración fluida con el sistema de monitoreo sin necesidad de recargar la aplicación.

El diseño está optimizado con CSS, asegurando una presentación clara y sin bordes. Además, se mejora la accesibilidad al proporcionar una descripción del iframe. Esta implementación garantiza una visualización eficiente y en tiempo real de los datos críticos del sistema.

```

Proyecto_Sena_2024_Calidad_Del_Agua > frontend > src > components > dashboard.jsx > ...
1 // Se importa la hoja de estilos específica para el Dashboard donde se especifica el tamaño del iframe
2 import "../styles/dashboard.css";
3 // Componente funcional Dashboard que muestra un iframe embebido con un panel de control
4 // para visualizar la calidad del agua en tiempo real.
5 const Dashboard = () => {
6   return(
7     //Contenedor del iframe con una clase CSS para aplicar estilos.
8     <div className="iframe-container">
9       /* Se inserta un iframe que carga el dashboard público de calidad del agua URL DEL DASHBOARD */
10      <iframe className="dashboard" src="https://d3nm9bfn-3808.use2.devtunnels.ms/public-dashboards/3773fddc87da4958be27bb54c77d8596"
11        // Descripción accesible del iframe para mejorar la accesibilidad
12        title="Dashboard de calidad del agua"
13        // Se elimina el borde del iframe para una apariencia más limpia
14        frameBorder="0">
15      </iframe>
16    </div>
17  );
18 };
19 // Se exporta el componente Dashboard para ser utilizado en otras partes de la aplicación.
20 export default Dashboard;

```

Figura 74 Implementación del Dashboard

3.7.1 Implementación del menú y alertas en tiempo real

El componente MiComponente gestiona la interfaz principal de la aplicación, proporcionando un menú de navegación dinámico, conexión en tiempo real con sensores y un sistema de notificaciones automáticas.

Este módulo permite a los usuarios acceder a diferentes secciones del sistema mediante un menú lateral interactivo, el cual puede expandirse o colapsarse según la preferencia del usuario. La navegación se gestiona con React Router, asegurando una experiencia fluida y sin recargas innecesarias.

El componente también se conecta al servidor a través de WebSockets, recibiendo datos de sensores en tiempo real y actualizando automáticamente los valores de monitoreo. Además, gestiona alertas de eventos críticos, utilizando SweetAlert2 para mostrar notificaciones visuales cuando se detectan anomalías en los sensores.

Adicionalmente, el sistema de notificaciones incluye un modal interactivo que permite a los usuarios visualizar, buscar y limpiar las alertas recibidas, asegurando un manejo eficiente de la información.

```
yecto_Sena_2024_Calidad_Del_Agua > frontend > src > components > menuprincipal.jsx > MiComponente
4  const MiComponente = () => {
9  }, [searchQuery, notifications]);
10
11  useEffect(() => {
12    const token = verificarToken();
13    if (!token) {
14      console.error("No hay token válido. Redirigiendo al login...");
15      navigate("/");
16      return;
17    }
18    let socket = obtenerSocket();
19    if (!socket) {
20      console.warn("El socket no está inicializado. Inicializando ahora...");
21      socket = conectarSocket(token);
22    }
23
24    socket.on("nuevodatoasensor", (datos) => {
25      setSensorData({
26        sensorPH: datos.sensorPH,
27        sensorTurbidez: datos.sensorTurbidez,
28        sensorOxygenConcentration: datos.sensorOxygenConcentration,
29        sensorOxygenSaturacion: datos.sensorOxygenSaturacion,
30        sensorConductividad: datos.sensorConductividad,
31      });
32    });
33    socket.on("notificacionAlerta", (data) => {
34      addNotification(data.message);
35      Swal.fire({
36        title: "¡Alerta!",
37        text: data.message,
38        icon: "warning",
39        confirmButtonText: "Entendido",
40        backdrop: true,
41        allowOutsideClick: false,
42        customClass: {
43          popup: "alerta-grande",
44        },
45      });
46    });
47  });
48  });
```

Figura 75 Implementacion del Menú y Alertas en Tiempo Real

Enlace de prueba de software:

<https://grafana-app-123-735f992a7113.herokuapp.com/>

CAPITULO 3. PRUEBAS DE SOFTWARE

1. Metodología de evaluación

Para validar la usabilidad y efectividad del aplicativo web, se implementó una fase de pruebas con usuarios. Esta evaluación incluyó un pre-test, en el que los participantes realizaron una serie de tareas antes de interactuar con el sistema, seguido de una fase de interacción con la plataforma y, posteriormente, un post-test para medir mejoras en desempeño y comprensión.

Se seleccionaron tres participantes bajo un acuerdo de confidencialidad, garantizando que los datos obtenidos fueran utilizados únicamente con fines de análisis del sistema. Durante la prueba, se registraron tiempos de ejecución de cada actividad para evaluar la eficiencia de la interfaz y la facilidad de uso.

2. Acta de confidencialidad

Antes de iniciar las pruebas, cada participante firmó un **acta de confidencialidad**, asegurando su consentimiento para la recopilación y análisis de datos. A continuación, se presenta el documento firmado por los participantes.

Acta de confidencialidad

Yo _____, identificado con C.c _____, Autorizo a Darwin Steven Gomez Ceballos con C.c xxxxxxxxxxxxxx, para que realice pruebas de usabilidad que serán aplicadas en la institución universitaria colegio mayor del cauca al aplicativo web de variables de calidad del agua, por medio de este documento; permito la captación de imagen voz y testimonio durante dichas pruebas. Como persona entiendo y acepto que los datos captados durante la realización de esta práctica de usabilidad podrán ser utilizados únicamente con fines de análisis para el presente estudio el cual tiene una finalidad exclusivamente académica. Acepto que dicha información no será comercializada ni utilizada con fines comerciales o de lucro. En caso de que desee revocar este consentimiento en cualquier momento, me comprometo notificarlo por escrito, Asimismo, Acepto que toda la información recolectada durante las pruebas de usabilidad es propiedad de la Institución Universitaria Colegio Mayor Del Cauca.



Figura 78 Participante 1

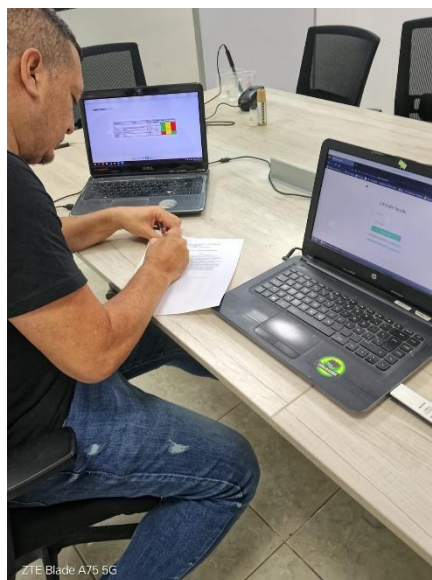


Figura 77 Participante 2



Figura 76 Participante 3

Descripción del proceso de pruebas

Le damos una calida bienvenida y agradecemos su participación en nuestra evaluación de usabilidad de la aplicación web de variables de calidad del agua. Su contribución es fundamental para mejorar la calidad de la aplicación y brindar una experiencia más satisfactoria.

El objetivo de esta evaluación es identificar posibles dificultades en el uso del aplicativo web y recopilar información valiosa para optimizar su funcionalidad queremos enfatizar que la evaluación se centra en la aplicación y no en su desempeño personal como usuario. La evaluación constara de tres etapas:

1. En la primera etapa, se le solicitara completar un breve cuestionario que abordara aspectos relacionados con su experiencia previa y sus hábitos de uso. Esta información nos ayudara a comprender mejor su perspectiva y contexto como usuario.
2. En la segunda etapa, se le presentará una serie de tareas que deberá realizar utilizando la aplicación. El objetivo es evaluar la usabilidad de la aplicación.
3. En la tercera etapa se le solicitara completar un cuestionario final donde podrá brindarnos su opinión generan sobre la experiencia de uso y compartir cualquier sugerencia o comentario adicional que considere relevante.

Si tiene alguna pregunta o duda durante el proceso de evaluación, nuestro equipo de evaluadores estará disponible para brindarle asistencia y aclarar cualquier inquietud que pueda surgir.

3. Pre-Test y resultados

El pre-test consistió en una serie de preguntas relacionadas con la temática del sistema y la facilidad percibida para realizar las tareas. Los participantes respondieron una serie de preguntas para evaluar su conocimiento previo y percepción sobre la plataforma. Estas preguntas fueron diseñadas para medir su nivel de familiaridad con el sistema y establecer un punto de comparación con los resultados del Post-Test.

Tabla 34 Pre-Test

Preguntas	Participante 1	Participante 2	Participante 3
Sabe usted que son los parámetros de calidad del agua como el PH, la turbidez la concentración de oxígeno y la conductividad.?	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Conoce algunas herramientas que permita monitorear en tiempo real las variables de la calidad del agua.?	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No
Has utilizado antes un Dashboard interactivo, como Grafana, para la visualización de datos.?	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Que perspectiva tienes al usar un aplicativo web de monitoreo de parámetros de la calidad del agua.?	Respuesta: calidad en la medición de los parámetros dispuestos conectividad y cumplimiento de restricciones.	Respuesta: Que permita monitorear y controlar las diferentes variables que se presentan para tener una buena calidad en el agua.	Respuesta: que los Dashboard sean precisos para permitir tomar decisiones y actuar a tiempo, es importante las alarmas.
Que tan importante consideras la visualización de datos en tiempo real para la toma de decisiones.?. califica con una (X) en una escala de 1 al 5 onde 1 es 'poco importante' y 5 es 'muy importante'.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input checked="" type="checkbox"/> 5	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input checked="" type="checkbox"/> 5	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input checked="" type="checkbox"/> 5
Que entiendes por "alerta temprana" en el contexto de calidad del agua.?. Especifique su respuesta:	Respuesta: por medio de datos específicos dar a conocer a partir de alertas tempranas la calidad del agua, para evitar posibles problemas hacia el futuro por su consumo.	Respuesta: una alerta que se genera para prevenir que la calidad del agua se vea afectada y llegado el caso poder dar solución.	Respuesta: Alerta al permitir ser proactivo y no correctivo al momento de identificar una variable fuera de sus parámetros normales.

Que tan importante crees que te sentirías al usar un aplicativo web para realizar tareas técnicas.? Califica con una(X) en una escala de 1 al 5 donde 1 es 'poco cómodo' y 5 es 'muy cómodo'	<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> </div>	<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> </div>	<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> </div>
Que características esperas que tenga un sistema de monitoreo eficiente.? Especifique su respuesta:	Respuesta: calidad, conectividad, análisis de datos, eficiencia, efectividad y pruebas.	Respuesta: Una interfaz de usuario fácil de usar y que permita mostrar y monitorear fácilmente las diferentes variables para mantener la calidad del agua.	Respuesta: oportunidad y claridad en la toma de la muestra y la generación de las alertas.

A continuación, se presentan capturas de las pruebas realizadas por los participantes durante el Pre-Test, donde se observa la interacción inicial con el sistema y el registro de sus respuestas.

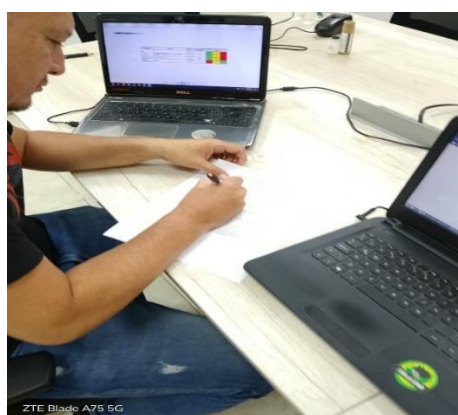


Figura 79 Pre-Test Evidencia

La primera pregunta del Pre-Test evaluó si los participantes estaban familiarizados con los parámetros de calidad del agua, como el pH, la turbidez, la concentración de oxígeno y la conductividad.

- Resultado: 100% de los participantes respondieron afirmativamente, indicando que conocen estos parámetros.

Estos resultados reflejan que todos los participantes tienen un conocimiento previo sobre los parámetros de calidad del agua, lo que puede facilitar su comprensión al interactuar con el sistema. Sin embargo, aunque están familiarizados con estos conceptos, esto no implica que tengan experiencia en su monitoreo o análisis en tiempo real.

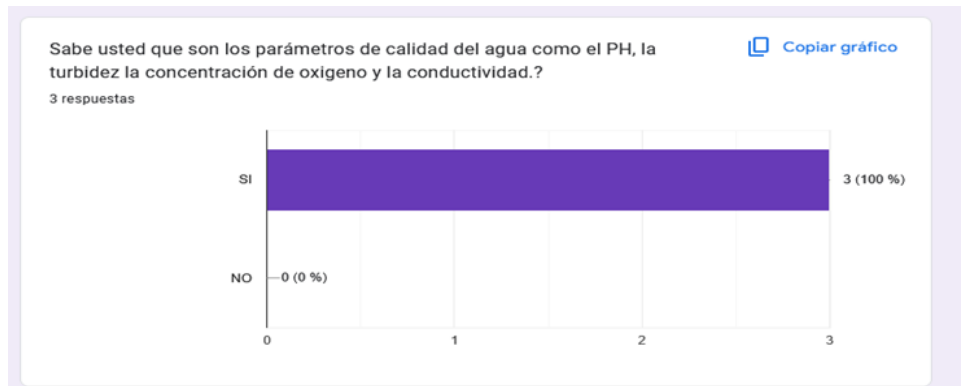


Figura 80 Resultado 1 Pre-Test

La segunda pregunta del Pre-Test evaluó si los participantes conocían herramientas que permitan monitorear en tiempo real las variables de calidad del agua.

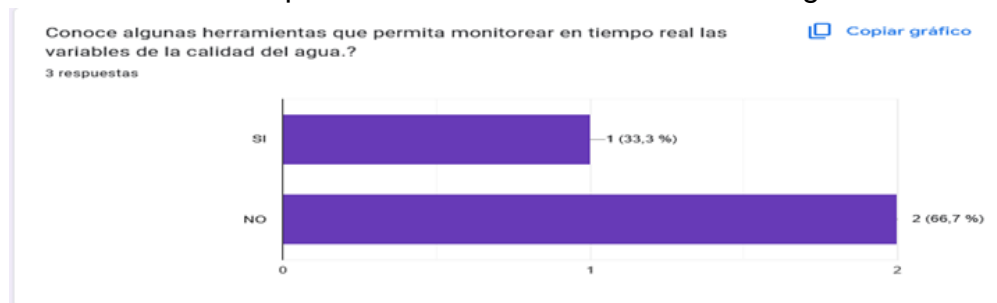


Figura 81 Resultado 2 Pre-Test

- Resultado: 66.7% de los participantes (2 de 3) respondieron "No", indicando que no conocen herramientas de este tipo.
33.3% (1 de 3) respondieron "Sí", lo que sugiere que solo una persona tiene noción de estas herramientas.

Estos resultados evidencian que la mayoría de los participantes no están familiarizados con tecnologías de monitoreo en tiempo real, lo que sugiere que el sistema debe proporcionar una interfaz intuitiva y explicaciones claras sobre su funcionamiento.

Otra pregunta buscó identificar las expectativas de los participantes sobre las funcionalidades clave que debería tener un sistema de monitoreo de calidad del agua.

Los participantes destacaron tres aspectos fundamentales para un sistema eficiente:

1. Precisión y oportunidad en la toma de muestras y generación de alertas.
 - Se enfatiza la importancia de que el sistema notifique oportunamente cuando los valores de calidad del agua exceden los límites aceptables.
2. Interfaz de usuario intuitiva y fácil de usar.

- Se espera que el sistema permita visualizar y monitorear de manera sencilla las variables de calidad del agua, sin requerir conocimientos avanzados.
3. Calidad, conectividad, análisis de datos y eficiencia.
- Los participantes consideran que el sistema debe garantizar confiabilidad en los datos, conectividad estable y herramientas analíticas para la toma de decisiones.

Que características esperas que tenga un sistema de monitoreo eficiente.? Especifique su respuesta:

3 respuestas

oportunidad y claridad en la toma de la muestra y la generación de las alertas.

Una interfaz de usuario fácil de usar y que permita mostrar y monitorear fácilmente las diferentes variables para mantener la calidad del agua.

calidad,conectividad,análisis de datos,eficiencia ,efectividad y pruebas.

Figura 82 Resultado 3 Pre-Test

4. Actividades Realizadas y Registro de Tiempos

Durante la prueba, cada participante realizó tareas específicas dentro del sistema. Se midió el tiempo de ejecución y se comparó el desempeño entre participantes.

Tabla 35 Actividades y Registro de Tiempo

Actividad	Participante 1	Participante 2	Participante 3	Promedio
Registrar usuario	35.36 s	59.59 s	41.60 s	45.51 s
Iniciar sesión	18.06 s	44.10 s	20.38 s	27.51 s
Seleccionar sensores	30.02 s	35.59 s	41.47 s	35.69 s
Visualizar datos en tiempo real	19.00 s	20.24 s	10.35 s	16.53 s
Verificar alerta generada	15.05 s	10.00 s	20.02 s	15.02 s
Consultar notificaciones	38.10 s	28.24 s	29.06 s	31.80 s
Revisar datos críticos	15.25 s	25.00 s	10.01 s	16.75 s
Editar perfil de usuario	1 min 16 s	41.28 s	56.66 s	58.31 s
Cambiar intervalo de tiempo en Dashboard	1 min 02 s	47.83 s	45.53 s	51.79 s
Filtrar datos en el Dashboard	30.02 s	30.00 s	1 min	40.01 s
Cerrar sesión	02.06 s	08.34 s	11.23 s	7.21 s

Los tiempos obtenidos muestran que la mayoría de las tareas se completaron en **menos de un minuto**, lo que sugiere que el sistema es **intuitivo y fácil de navegar**. Sin embargo, se identificaron algunas actividades que tardaron más tiempo, lo que puede indicar la necesidad de mejoras en la interfaz o en la orientación del usuario.

Registro de usuario: Fue la tarea más variable en tiempos, con un rango de **35 a 59 segundos**, lo que indica que algunos usuarios pueden requerir mayor claridad en el formulario.

Perfil de usuario: Fue la tarea más lenta, con un promedio de **58.31 segundos**, lo que podría sugerir mejoras en la navegación de esta función.

Filtrado de datos en el Dashboard: Presentó una diferencia significativa en el tiempo de ejecución del tercer participante, lo que puede indicar que algunos usuarios necesitan más orientación sobre esta función.

5. Post-test resultados

Después de la prueba, los participantes respondieron nuevamente preguntas relacionadas con su experiencia, facilidad de uso y mejoras percibidas en comparación con el pre-test.

Tabla 36 Post-Test

Preguntas	Participante 1	Participante 2	Participante 3
¿Qué tan fácil te resultó aprender a usar el aplicativo web? en una escala del 1 al 5, donde 1 significa 'muy difícil' y 5 significa 'muy fácil'	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
El Dashboard interactivo te facilitó la identificación rápida de cambios en los parámetros de calidad del agua?"	si no parcialmente	si no parcialmente	si no parcialmente
¿Los datos en tiempo real se actualizaron correctamente en el Dashboard?	si no parcialmente	si no parcialmente	si no parcialmente
¿Crees que las alertas fueron claras y oportunas?	si no parcialmente	si no parcialmente	si no parcialmente
¿Qué tan útil te resultó el uso de colores y gráficos para interpretar las alertas y resultados? en una escala del 1 al 5. Siendo 1 nada útil y 5 muy útil.	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
¿Qué aspectos del sistema considerarías mejorar?	Respuesta: Experiencia de usuario	Respuesta: El tamaño de la alerta.	Respuesta: Temas de usabilidad o guía del proceso, no dar por sentado de que la persona que realiza la prueba tiene o no conocimiento.

¿Qué parte del sistema te resultó más útil y por qué?	Respuesta: Análisis de datos efectivo y bajo niveles de tiempo.	Respuesta: El poder mirar en tiempo real La generación de la alerta.	Respuesta: El Dashboard ya que me muestra en tiempo real las alertas y toma de muestra.
¿Qué funcionalidad agregarías para mejorar el sistema?	Respuesta: Recomendaciones según resultados generados	Respuesta: Al momento de generar la alerta, adicionalmente se pudiera generar un sonido.	Respuesta: Reportes y descarga de los reportes que puede ser en formato PDF.

En esta pregunta del Post-Test, se evaluó qué tan fácil les resultó a los participantes aprender a usar la plataforma, utilizando una escala del 1 al 5 (donde 1 es "muy difícil" y 5 es "muy fácil").

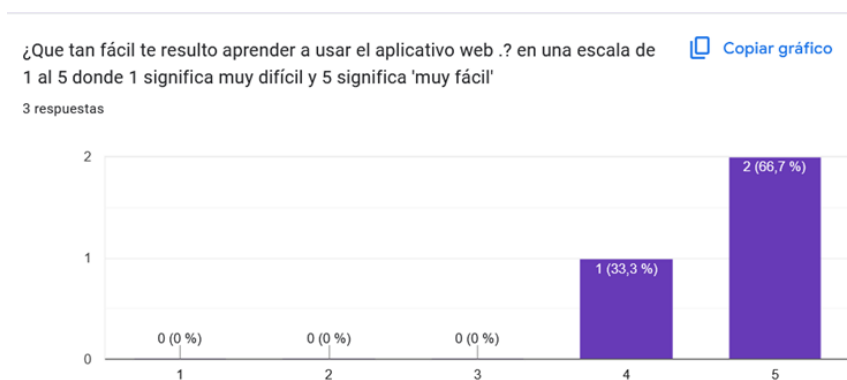


Figura 83 Resultado 1 Post-Test

- Resultado: 66.7% (2 de 3) calificaron con 5, indicando que el uso del sistema fue muy fácil de aprender.

33.3% (1 de 3) calificó con 4, lo que también refleja una experiencia de aprendizaje positiva.

0% de los participantes indicaron dificultades significativas en el uso del sistema.

Los resultados reflejan que la plataforma es intuitiva y fácil de aprender, incluso para nuevos usuarios. La interfaz y las funcionalidades parecen estar diseñadas de manera accesible, lo que permite que los usuarios se familiaricen rápidamente con el sistema.

Esta pregunta del Post-Test evaluó si las alertas generadas por el sistema fueron percibidas como claras y oportunas por los participantes.

- Resultado: 100% de los participantes (3 de 3) respondieron "Sí", indicando que las alertas fueron efectivas y se generaron en el momento adecuado.

0% indicó que las alertas fueron "parcialmente claras" o "no claras", lo que refleja una alta precisión en la funcionalidad de alertas del sistema.

Estos resultados confirman que el sistema de alertas cumple con su propósito, notificando a los usuarios de manera efectiva y en el momento correcto cuando se detectan valores críticos en los parámetros de calidad del agua. La claridad de las alertas permite que los usuarios puedan tomar decisiones informadas sin confusión ni retrasos.

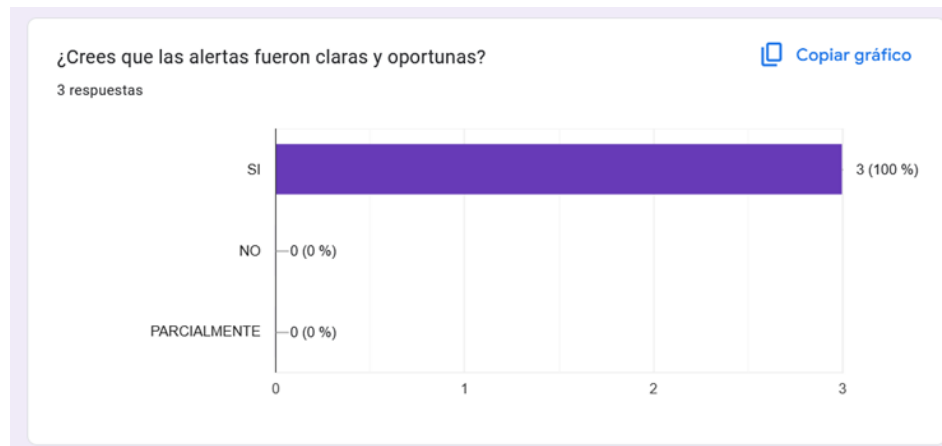


Figura 84 Resultado 2 Post-Test

Esta pregunta buscó identificar oportunidades de mejora en la plataforma según la experiencia de los participantes.

Las respuestas reflejan tres áreas principales de mejora:

1. Tamaño de la alerta

- Se sugiere ajustar el tamaño de las notificaciones para que sean más visibles y llamativas, mejorando la percepción del usuario sobre eventos críticos.

2. Usabilidad y guías del proceso

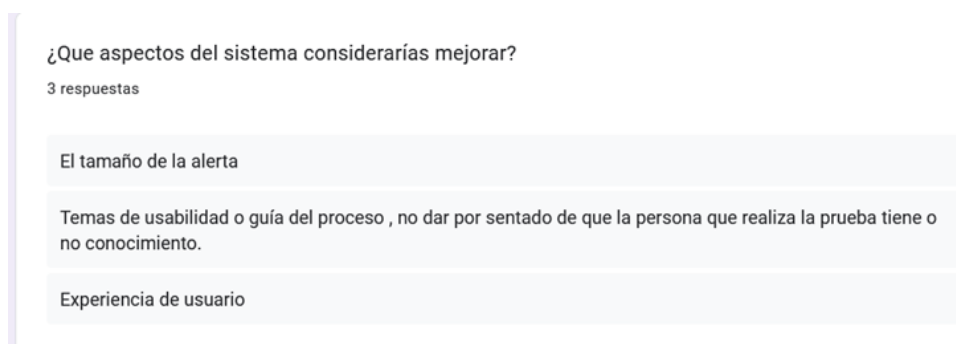
- Se recomienda incluir una **guía interactiva o tutorial** dentro del sistema para ayudar a los nuevos usuarios a comprender mejor el flujo de trabajo. Esto evitaría asumir que el usuario ya tiene conocimientos previos.

3. Experiencia de usuario (UX)

- Se destaca la necesidad de mejorar la **interfaz y navegación**, optimizando el diseño para una interacción más intuitiva y fluida.

A pesar de que el sistema fue bien recibido en términos generales, los participantes identificaron mejoras relacionadas con visibilidad, accesibilidad y experiencia de usuario.

Esto indica que, aunque la plataforma es funcional, pequeños ajustes en diseño y orientación podrían mejorar aún más su usabilidad.



¿Que aspectos del sistema considerarías mejorar?

3 respuestas

El tamaño de la alerta
Temas de usabilidad o guía del proceso , no dar por sentado de que la persona que realiza la prueba tiene o no conocimiento.
Experiencia de usuario

Figura 85 Resultado 3 Post-Test

Esta pregunta permitió a los participantes proponer mejoras funcionales que aumentarían la utilidad del sistema, las respuestas sugieren tres mejoras clave:

1. **Alerta sonora para notificaciones**

- Se recomienda agregar **un sonido al generar una alerta**, lo que facilitaría una respuesta inmediata a eventos críticos.

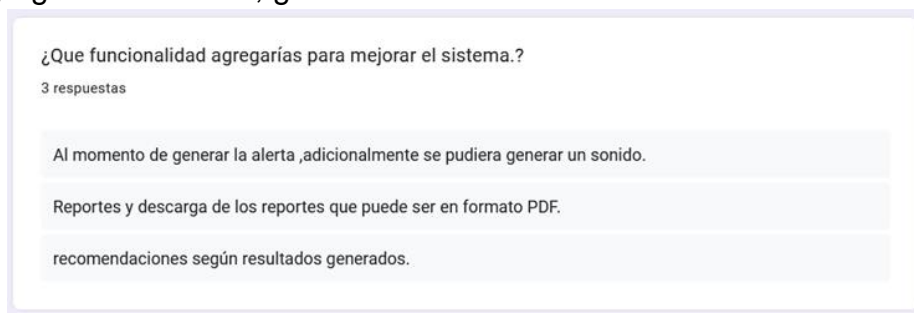
2. **Descarga de reportes en formato PDF**

- 3. Se sugiere incluir la opción de **generar y descargar reportes en PDF**, lo que permitiría almacenar y compartir información de manera más formal y accesible.

Recomendaciones basadas en los datos generados

- Se plantea la posibilidad de que el sistema **genere sugerencias automáticas** según los resultados, ayudando a los usuarios a interpretar los datos y tomar decisiones.

Las funcionalidades propuestas buscan mejorar la experiencia del usuario y la gestión de la información. La incorporación de alertas sonoras haría que las notificaciones sean más efectivas, mientras que la generación de reportes facilitaría el análisis y respaldo de los datos. Además, incluir recomendaciones basadas en los resultados proporcionaría un valor agregado al sistema, guiando al usuario en la toma de decisiones.



¿Que funcionalidad agregarías para mejorar el sistema.?

3 respuestas

Al momento de generar la alerta ,adicionalmente se pudiera generar un sonido.
Reportes y descarga de los reportes que puede ser en formato PDF.
recomendaciones según resultados generados.

Figura 86 Resultado 4 Post-Test

Conclusión

- La creación del sistema web para el estudio y visualización de datos en alertas precoces de calidad del agua ha probado ser una solución eficaz para el seguimiento en tiempo real de parámetros vitales como el pH, la turbidez, el oxígeno disuelto y la conductividad. La adopción de tecnologías como tableros interactivos, advertencias automáticas y almacenamiento de datos ha facilitado la comprensión de la información, fomentando una toma de decisiones más rápida y basada en datos exactos.
- Los hallazgos del pre-test y post-test mostraron un notable avance en la percepción y utilización del sistema de los participantes. Aunque inicialmente había incertidumbres respecto al uso de herramientas digitales para monitoreo en tiempo real, tras la prueba práctica, los usuarios expresaron una experiencia favorable, destacando la sencillez de manejo y la relevancia de la visualización de datos en tiempo real. No obstante, las pruebas también facilitaron la detección de oportunidades de mejora, tales como modificaciones en la experiencia del usuario, incorporación de alertas sonoras y alternativas sofisticadas para la creación de informes en formatos de fácil acceso como PDF.
- Adicionalmente, este sistema no solo satisface una demanda urgente en la supervisión de la calidad del agua en acueductos rurales, sino que también ofrece la oportunidad de extender su uso a otros contextos, como plantas de tratamiento de agua, redes de distribución y sistemas industriales que necesiten un control exacto de la calidad del recurso hídrico.
- Pese a los progresos alcanzados, resulta crucial seguir con optimizaciones y validaciones futuras en situaciones reales. Además, sería provechoso investigar la integración con aparatos móviles para facilitar el acceso a distancia y alertas en tiempo real a los encargados del seguimiento.
- Para finalizar, este proyecto describe un avance importante hacia la digitalización de la inspección de la calidad del agua, impulsando la transparencia, eficacia y cercanía en la administración de este recurso vital. Su autoridad podría superar el ámbito local, funcionando como relato para iniciativas de cuidado ambiental en diversos sectores.

Recomendaciones

En el marco del desarrollo del sistema web para el seguimiento de la calidad del agua en tiempo real, se sugiere implementar una serie de medidas que faciliten la mejora y

optimización de futuros proyectos de este tipo. Primero, es crucial llevar a cabo un análisis de las necesidades globales, o sea, una valoración de las funcionalidades extra que podrían incorporarse al sistema. Esto no solo facilitará la identificación de las necesidades actuales de los usuarios finales, sino que también permitirá anticipar futuras ampliaciones, como la inclusión de análisis predictivos, adaptación de alertas y compatibilidad con aparatos móviles. Una organización exhaustiva desde el comienzo asegurará que el sistema pueda ajustarse a nuevas demandas y potenciar su rendimiento a lo largo del tiempo.

Además, se aconseja considerar la utilización de una base de datos relacional en vez de MongoDB, lo que facilitará una mejor organización de los datos y mejoraría las consultas para el almacenamiento de información histórica. Un modelo de datos claramente establecido ayudaría a incrementar la eficacia del sistema, previniendo dificultades de escalabilidad y garantizando la integridad de los datos obtenidos al supervisar los parámetros de calidad del agua. Si se pretende conservar MongoDB, sería aconsejable mejorar la indexación y administración de los datos en tiempo real para asegurar un desempeño óptimo.

Finalmente, se recomienda valorar la incorporación de tecnologías emergentes que optimicen la experiencia del usuario y el rendimiento del sistema. Incorporar alertas de sonido, informes en PDF y una interfaz más sencilla simplificará la interacción con la plataforma. Igualmente, la implementación de una versión móvil facilitaría a los usuarios el acceso a la información en cualquier instante y recibir alertas en tiempo real. La ejecución de ensayos constantes con los usuarios finales asegurará que el sistema continúe progresando y se ajuste a las demandas variables, garantizando su eficacia y beneficio a largo plazo.

Trabajos Futuros

Para asegurar el progreso y perfeccionamiento constante del sistema web de seguimiento de calidad del agua en tiempo real, es esencial continuar expandiendo su funcionalidad y mejorar su desempeño. Esto garantizará que la plataforma no solo permanezca activa, sino que también pueda ajustarse a nuevas demandas y retos en la administración del recurso hídrico.

Dentro de las mejoras a tener en cuenta en futuras versiones del sistema, se propone la implementación de un modelo de predicción fundamentado en inteligencia artificial, que facilite prever irregularidades en los parámetros de calidad del agua antes de que lleguen a niveles críticos. Esto simplificará el proceso de tomar decisiones preventivas y mejoraría la utilización de los recursos. Además, se sugiere la incorporación de módulos para el análisis histórico de datos, lo que posibilitará a los usuarios observar tendencias

y hacer comparaciones entre distintos lapsos de tiempo, potenciando la capacidad de reacción frente a variaciones en las condiciones del agua.

Otra tarea futura implica la creación de una aplicación móvil adicional, que facilite el acceso a los datos de seguimiento en tiempo real desde cualquier sitio y obtener alertas inmediatas a través de notificaciones Push. Esto incrementará la accesibilidad y la habilidad para reaccionar frente a sucesos críticos. Además, se recomienda la mejora del sistema de informes, incluyendo la posibilidad de exportar datos en formatos PDF y CSV, lo que simplifica la documentación y el estudio de los datos.

Se aconseja seguir validando el sistema mediante pruebas en situaciones reales, con la implicación de usuarios de diversos sectores. Esto facilitará la recolección de comentarios esenciales para continuar con la modificación de la interfaz, optimizar la experiencia del usuario y asegurar que la plataforma alcance su objetivo de proporcionar un seguimiento fiable y eficaz de la calidad del agua. La aplicación de estas mejoras potenciará el efecto del sistema y lo establecerá como una herramienta sólida y flexible ante necesidades futuras.

Referencias

- Ahmad, Y. A., Gunawan, T. S., Munir, M. H. M. R. A., Yusoff, N. Md., Kartiwi, M., & Mansor, H. (2022). *2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 267-272.
<https://doi.org/10.1109/ICSIMA55652.2022.9928973>
- Alexander Menzinsk, G. L., & Juan Palacio, Miguel Ángel Sobrino, R. Á. y V. Rivas. (2022). *Historias de Usuario Ingeniería de Requisitos Ágil*. 3.01, 62.
- Arduino. (2024). Vernier. <https://www.vernier.com/engineering/arduino/>
- Avelino Ospina, J. A. (2023). *Diseño e implementación de la plataforma de monitoreo para el laboratorio Kiosol*. <http://hdl.handle.net/1992/64045>
- Blanchon, B. (2024). *Bblanchon/ArduinoJson* [C++]. <https://github.com/bblanchon/ArduinoJson>
(Obra original publicada en 2014)
- Comandos AT del módulo Bluetooth HC-05 y HC-06. (2023, marzo 14). *MICROCHIPOTLE*.
<https://microchipotle.com/comandos-at-del-modulo-bluetooth-hc-05-y-hc-06/>
- Coming soon: *Balsamiq Wireframes*—*Balsamiq Company News*. (2024). [Balsamiq Wireframes]. Coming soon: Balsamiq Wireframes.
<https://balsamiq.com/company/news/balsamiq-wireframes/>
- Conductivity Probe*. (2024). Vernier. <https://www.vernier.com/product/conductivity-probe/>
- Conductivity Probe User Manual*. (2024a). Vernier. <https://www.vernier.com/manuals/con-bta/?srsltid=AfmBOortnZxQy0gvOga6NMtg4JzwMAoOLdaTcXGNEN8kWdeinBqPnvAr>
- Conductivity Probe User Manual*. (2024b). [Conductivity Probe User Manual]. Vernier.
https://www.vernier.com/manuals/con-bta/?srsltid=AfmBOooC6pYNLbbdOjkga7Nllv-g8WWQe-2FR6ywXXjvWUY7P_HzOsGw
- Connecting Vernier Sensors to Arduino*. (2023). Vernier.
<https://www.vernier.com/engineering/arduino/arduino-online-guide/connecting-vernier-sensors-to-arduino/>

Desarrollo de backend para sistema de medición automática de calidad de agua, adecuado a sensores de bajo costo. (2021). [Tesis, universidad de chile].

<https://repositorio.uchile.cl/handle/2250/181235>

Environmental data management system | WISKI. (2024, septiembre 21). Kisters.

<https://www.kisters.net/wiski/>

Fathoni, H., Miao, H.-Y., Chen, C.-Y., & Yang, C.-T. (2020). A Monitoring System of Water Quality Tunghai Lake Using LoRaWAN. *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*, 281-283. <https://doi.org/10.1109/ICPAI51961.2020.00060>

GitHub—Knolleary/pubsubclient: A client library for the Arduino Ethernet Shield that provides support for MQTT. (2020). <https://github.com/knolleary/pubsubclient>

GPRS. (2024). <https://www.linkedin.com/pulse/gprs-sajeni-g>

Grafana dashboards | Grafana k6 documentation. (2024). Grafana Labs.

<https://grafana.com/docs/k6/latest/results-output/grafana-dashboards/>

HTMLandCSS - Ascent Infotech Computer Training Center Kolkata. (2021, mayo 25).

<https://ascentcomputer.co.in/htmlandcss/>

Méndez, J. O. M. (2020). LOS RETOS DEL ACCESO A AGUA POTABLE Y SANEAMIENTO BÁSICO DE LAS ZONAS RURALES EN COLOMBIA. *Revista de Ingeniería*, 49, Article 49. <https://doi.org/10.16924/revinge.49.5>

pH Sensor. (2024). Vernier. <https://www.vernier.com/product/ph-sensor/>

pH Sensor User Manual. (2024a). https://www.vernier.com/manuals/ph-bta/?srsltid=AfmBOorUXzbd7_9oE9h4wvu7MRK3NRSX0FNon9e29Z5yqsgMxH3KneT

pH Sensor User Manual. (2024b). pH Sensor User Manual.

[https://www.vernier.com/manuals/ph-bta/?srsltid=AfmBOorfYYkJ3-](https://www.vernier.com/manuals/ph-bta/?srsltid=AfmBOorfYYkJ3-pbdf5tl81p5vtv6wRL5uo4YNm6zSa86Q41lpZ8Oszq)

[pbdf5tl81p5vtv6wRL5uo4YNm6zSa86Q41lpZ8Oszq](https://www.vernier.com/manuals/ph-bta/?srsltid=AfmBOorfYYkJ3-pbdf5tl81p5vtv6wRL5uo4YNm6zSa86Q41lpZ8Oszq)

Rinke, A. (2021, abril 1). What is MongoDB? - OPC-Router.de explains it to you exactly. *OPC Router - The Communication Middleware.* <https://www.opc-router.com/what-is-mongodb/>

Servicios de Datos HydroVu | Software de Gestión de Datos de Agua | In-Situ. (2024, septiembre 21). <https://in-situ.com/es/hydrovu-data-services>

Socket.IO: Implementing Real-Time Communication in Node.js | by WebClues Infotech | Sep, 2024 | JavaScript in Plain English. (2024). <https://javascript.plainenglish.io/socket-io-implementing-real-time-communication-in-node-js-a7869d76e4af>

TinyGSM/src/TinyGsmGPS.tpp at master · vshymanskyy/TinyGSM. (2024). GitHub. <https://github.com/vshymanskyy/TinyGSM/blob/master/src/TinyGsmGPS.tpp>

Turbidity Sensor. (2024). Vernier. <https://www.vernier.com/product/turbidity-sensor/>

Turbidity Sensor User Manual. (2024a). <https://www.vernier.com/manuals/trb-bta/?srsltid=AfmBOopY2FNOHsqHClqRbJh9RMQA6zoUCUVm4q6hTyNsjNttczj9uGzl>

Turbidity Sensor User Manual. (2024b). Turbidity Sensor User Manual. https://www.vernier.com/manuals/trb-bta/?srsltid=AfmBOorlBapwj8Z_m2i01KDrJCDUXAtKVUB-f2fb2j6dY4Dj5ixwS1pJ

Understanding the differences between JavaScript, Node.js, and Express.js. (2024, junio 3). DEV Community. <https://dev.to/richardshaju/understanding-the-differences-between-javascript-nodejs-and-expressjs-5cb9>

Vernier Arduino™ Interface Shield | Vernier Canada. (2024). <https://www.verniercanada.ca/product/interfaces/vernier-arduino-interface-shield/>

Vernier Coding Activities with Arduino®: Analog Sensors— Vernier. (2024). <https://www.vernier.com/product/vernier-coding-activities-with-arduino-analog-sensors/>

Vernier Optical DO Probe. (2024). Vernier. <https://www.vernier.com/product/vernier-optical-do-probe/>

Vernier Optical DO Probe User Manual. (2024a). <https://www.vernier.com/manuals/odo-bta/>

Vernier Optical DO Probe User Manual. (2024b). <https://www.vernier.com/manuals/odo-bta/>

Wiski-overview-en.pdf. (s. f.). Recuperado 14 de octubre de 2024, de <https://www.kisters.net/app/uploads/2023/06/wiski-overview-en.pdf>

