

# Alertas Ambientales A Través De Arduino En Tiempo Real En Tu Región

MIS. Marco Antonio Celis Crisostomo, MEM. Francisco Miguel Hernandez López, MIE.  
Emmanuel Vega Negrete, MER. Jorge Alberto Cardenas Magaña

## Resumen.

En este estudio, se emplea un Arduino para medir la humedad, temperatura y calcular el índice de calor en una región específica. Además, se monitorea la calidad del aire y los niveles de radiación ultravioleta (UV) del área de estudio. Los datos recolectados son transmitidos a un servidor web mediante un Shield Ethernet, que los envía a un EndPoint para su almacenamiento en una base de datos. Esta base de datos es accesible a través de un sitio web adaptativo para dispositivos móviles y web, permitiendo a los usuarios visualizar en tiempo real las condiciones climáticas registradas. Esto facilita la toma de decisiones informadas, como la práctica de deportes y la planificación de cultivos, basadas en las condiciones climáticas actuales.

**Palabras claves - Arduino, Angular, Sensores, Transmisión de Datos.**

## Abstract.

In this study, an Arduino is used to measure humidity, temperature, and calculate the heat index in a specific region. Additionally, air quality and ultraviolet (UV) radiation levels in the study area are monitored. The collected data is transmitted to a web server via an Ethernet Shield, which sends it to an Endpoint for storage in a database. This database is accessible through a responsive website for both mobile and web devices, allowing users to view real-time recorded weather conditions. This facilitates informed decision-making, such as sports activities and crop planning, based on current weather conditions.

**KeyWorks - Arduino, Angular, Sensors, Data Transmisión.**

## I. INTRODUCCIÓN

En la evolución de los sitios web a lo largo de los años, se ha puesto énfasis en dos grandes bloques principales. El primero, conocido como BackEnd [5], se encarga de proporcionar servicios mediante diversas técnicas como arquitectura Restful, APIs o microservicios [4], adaptados a las especificaciones del servidor. Este componente selecciona el lenguaje más adecuado para desarrollar estos servicios y se encarga de generar funciones que pueden ser invocadas desde el FrontEnd [5]. Esta estructura facilita el desarrollo simultáneo del sitio web, optimizando así las capacidades del servidor. En este contexto, implementamos estos servicios utilizando PHP, uno de los lenguajes más utilizados para tareas dentro del servidor, conectándose a la base de datos. La creación de microservicios se adapta según las necesidades específicas del sitio web, mejorando la presentación de datos y los procesos accesibles desde el FrontEnd, gestionados eficientemente en el BackEnd.

---

Tecnológico Nacional de México / Doctorante de Colegio de Formación Educativa TENAM / [marcoantonio.cecris@unitenam.org](mailto:marcoantonio.cecris@unitenam.org)

Tecnológico Nacional de México / Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Unidad Académica Tamazula / [francisco.hernandez@tamazula.tecmm.edu.mx](mailto:francisco.hernandez@tamazula.tecmm.edu.mx)

Tecnológico Nacional de México / Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Unidad Académica Tamazula / [emmanuel.vega@tamazula.tecmm.edu.mx](mailto:emmanuel.vega@tamazula.tecmm.edu.mx)

Tecnológico Nacional de México / Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Unidad Académica Tamazula / [jorge.cardenas@tamazula.tecmm.edu.mx](mailto:jorge.cardenas@tamazula.tecmm.edu.mx)

La aplicación de estas técnicas de servicios implica la creación de tareas específicas que garantizan un procesamiento eficiente para representar o almacenar datos, manteniendo la integridad de la seguridad y la lógica de la información en la base de datos. En nuestro caso, empleamos phpSlim [8] y phpMyAdmin como gestor de base de datos integrado en el servidor. Estos microservicios están diseñados para resolver problemas o responder a actividades asignadas, permitiendo al usuario final visualizar el sitio web de manera integral y ofreciendo una estructura flexible que permite la modificación de áreas específicas o tareas sin afectar otras partes del sistema.

Los servicios implementados pueden ser invocados desde diversas plataformas, ya sea a través de otras APIs o dispositivos como computadoras o Arduino, utilizando las rutas de conectividad y parámetros definidos en la lógica de negocios del proyecto.

Como segundo bloque, se encuentra la creación de sitios web basados en estándares o modelos de desarrollo específicos. Estos modelos dependen del lenguaje elegido durante la creación y ejecución del sitio web. Uno de los patrones más adoptados es el modelo MVC (Modelo-Vista-Controlador) [3], ampliamente utilizado bajo el lenguaje de programación Angular [1]. Este modelo proporciona ventajas significativas en términos de depuración, generación y desarrollo paralelo, dependiendo del equipo de desarrolladores involucrado en el proyecto.

Una característica crucial al implementar un sitio web es su capacidad para ejecutarse según las especificaciones particulares del usuario final [2], utilizando los componentes necesarios y definiendo las características únicas de cada uno. Posteriormente, estos componentes pueden ser invocados por otros dentro del mismo sitio web durante su ejecución y creación.

Para maximizar estas ventajas, hemos optado por Angular versión 16 debido a su enfoque en la modularidad y reutilización de componentes, lo que facilita la gestión eficiente del código y proporciona mayor flexibilidad en la implementación del sitio web.

El desarrollo FrontEnd se integra con la maquetación de MaterializeCSS [6], asegurando que el sitio web sea accesible desde dispositivos como computadoras o teléfonos móviles, adaptándose así a las preferencias y capacidades del usuario final.

Además, la implementación del circuito encargado del censo y envío de datos al servidor se realizó mediante el microcontrolador Arduino, junto con la placa Shield y sensores de humedad, temperatura, gases y rayos ultravioleta (UV), cumpliendo con los objetivos específicos de este proyecto.

## II. METODOLOGÍA

Para la implementación y desarrollo del proyecto, tomamos en cuenta las características que mejor se adaptarían a las funciones de cada uno de los integrantes del equipo de desarrollo. A continuación, se definieron los siguientes aspectos a seguir:

### A. Explorando la problemática a desarrollar

En nuestra región la agricultura forma parte del proceso económico, donde utilizamos diversos métodos, desde riego manual hasta sistemas automáticos e incluso invernaderos. Estas diversas técnicas producen diferentes tipos de cultivos: de temporada, de sistemas automáticos o protegidos. Teniendo esto en cuenta, hemos ideado un sistema web conectado a una antena que detecta los cambios ambientales. Los datos recopilados por esta antena se pondrán a disposición de las empresas o habitantes para que puedan estar más informados sobre su entorno, ayudándoles a la toma de decisiones con respecto a riegos o practicas de deportes al aire libre. En la siguiente imagen, denominada Imagen 1, mostramos los resultados del proceso de implementación.

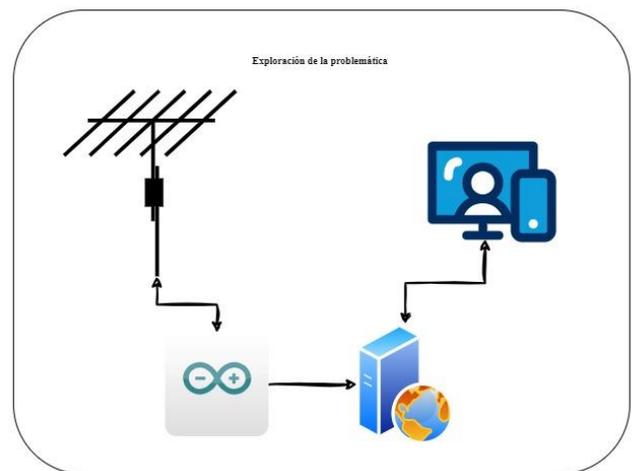


Imagen 1. Diagrama de la propuesta  
Fuente: Elaboración propia (2024)

### B. Planificación de actividades:

En esta etapa, se han establecido las características del proyecto, resaltando tanto las actividades grupales como las individuales. También se ha definido la calendarización para la entrega de avances del proyecto, además de especificar las herramientas y lenguajes de programación para las distintas partes del proyecto.

*Tabla 1 Listado de actividades generales por el equipo de desarrollo*

Sprint y Estatus	Nombre de la actividad	Encargado
AATRT01	Creación del esquema de la base de datos para almacenar información de residencias.	PR01
	Implementación de las relaciones entre las entidades de la base de datos.	PR01
	Diseño de lógica de negocio para la recuperación de datos.	PR02
	Pruebas de funcionalidad por parte del tester y generación de reporte de retroalimentación.	PR01 PR02 TES01
	Reunión de retroalimentación del Sprint terminado.	PM01
AATRT02	Instalación del servidor Web de manera local para pruebas internas.	PR01
	Instalación del lenguaje de programación del frameworks phpSlim dentro del servidor	PR02
	Codificación de las actividades generadas para cada una de los EndPoints	PR02
	Pruebas de funcionalidad por parte del tester y generación de reporte de retroalimentación.	PR01 PR02 TES01
	Reunión de retroalimentación del Sprint terminado.	PM01
AATRT03	Instalación de node e instalación del lenguaje de programación angular	PR02
	Codificación del sitio web de las diferentes funcionalidades.	PR02
	Codificación de la comunicación entre FrondEnd y BackEnd el proyecto.	PR02
	Pruebas de funcionalidad por parte del tester y generación de reporte de	PR01 PR02 TES01

	retroalimentación.	
	Reunión de retroalimentación del Sprint terminado.	PM01
AATRT04	Diseñar circuito con un diseñador digital	PR01
	Implementar circuito con los diferentes sensores.	PR01
	Diseñar CAD para la antena así como impresión en 3D de la pieza final.	PR01
	Codificación de las diferentes funcionalidades que contendrá el dispositivo.	PR01
	Pruebas de funcionalidad por parte del tester y generación de reporte de retroalimentación.	PR01 PR02 TES01
	Reunión de retroalimentación del Sprint terminado.	PM01
AATRT05	Realización de pruebas de funcionamiento del sistema en un entorno de producción.	PR01 PR02 TES01 PM01
	Ajustes finales basados en retroalimentación de pruebas y usuarios.	PR01 PR02 TES01 PM01
	Despliegue del sistema de cuida tu entorno, cuida tu salud: alertas ambientales en tiempo real en tu región	PR01 PR02 TES01 PM01

*Fuente: elaboración propia (2024)*

En la Tabla 1 se detallan las actividades llevadas a cabo para implementar la funcionalidad del proyecto, con un equipo de desarrollo conformado por 2 programadores, 1 tester y el encargado del proyecto. Estos miembros ejecutarán las actividades de acuerdo con el plan establecido.

### C. Diseño de casos de uso

Derivado de las actividades generadas, se definieron funcionalidades para los casos de uso de la futura aplicación. A continuación, se presenta el proceso principal del proyecto, en el cual se muestra cómo interactuarán los usuarios dentro del sistema, definiendo las funcionalidades a realizar dentro de la aplicación web:

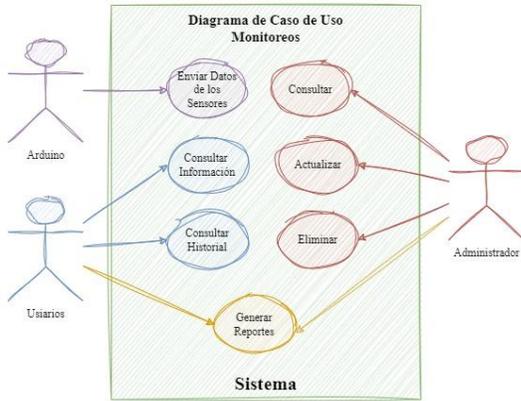


Imagen 2. Diagrama de caso de uso de Monitoreos  
Fuente: Elaboración propia (2024)

Después de realizar los casos de uso, se define cuál será la secuencia que se implementará de acuerdo con las necesidades del proyecto:

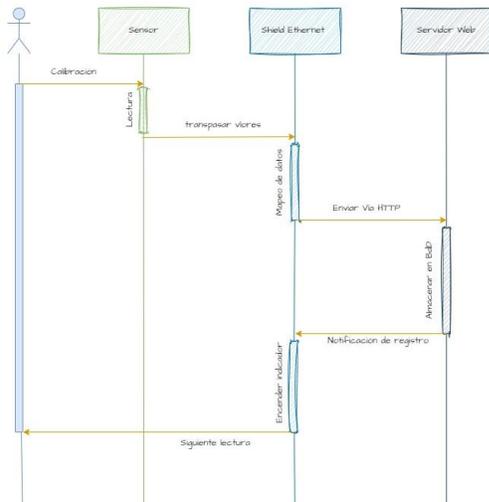


Imagen 3. Diagrama de secuencia del caso monitoreo  
Fuente: Elaboración propia (2024)

En este diagrama se define la secuencia que debe seguir la solicitud de envío de datos al servidor.

D. Diseño de Base de datos:

Se diseñó en base a las características definidas dentro del proyecto, las cuales cubren en su totalidad lo que se busca implementar [7].



Imagen 4 Diseño de la base de datos en plataforma <https://dbdiagram.io/>  
Fuente: Elaboración propia (2024)

El diseño de la base de datos nos brinda la posibilidad de tener más de una antena que se pueda consultar o anexar al proyecto.

E. Modelado del Sitio Web

El lenguaje seleccionado es Angular en su versión 16, combinado con el CSS Materialize-CSS, para crear un sitio web adaptable a diversos tamaños de pantalla, desde dispositivos móviles hasta pantallas Smart. Esta elección garantiza una experiencia de usuario consistente y atractiva en diferentes dispositivos. Además, Angular utiliza la metodología MVC (Modelo-Vista-Controllador) [3], lo que facilita el trabajo colaborativo y permite una codificación más ágil. Esta metodología mejora la eficiencia en la detección y corrección de errores de codificación, lo que resulta en un desarrollo más rápido del proyecto.

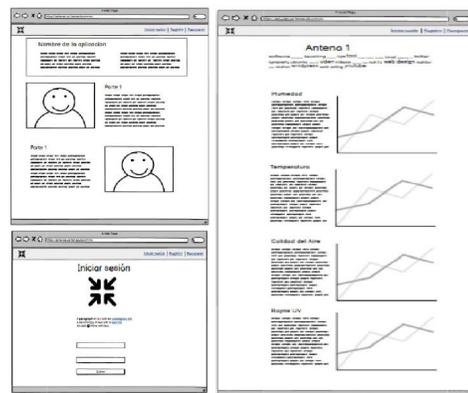


Imagen 5 Diseño del sitio Web con la herramienta Balsamiq.  
Fuente: Elaboración propia (2024)

El diseño del proyecto se llevó a cabo utilizando la plataforma Balsamiq[11]. Este diseño nos permite distribuir tareas y asignar actividades de forma paralela, de acuerdo con el diseño propuesto y las características del CSS. Para la implementación, se utilizó MaterializeCSS[12], el cual se integra con el framework Angular.

#### F. Integración de Arduino

La implementación y configuración de los sensores de humedad, temperatura, gases y radiación ultravioleta (UV) se llevó a cabo considerando las especificaciones de cada componente. Se programaron las características de envío de datos utilizando el protocolo JSON, que se implementó en la invocación POST hacia el servidor web. De esta manera, se aseguró la transmisión eficiente y precisa de la información capturada por los sensores al servidor, permitiendo su posterior procesamiento y almacenamiento para su análisis o visualización. A continuación representamos el diseño del circuito realizado dentro de la herramienta fritzing[13] la cual brinda el panorama general del prototipo interno de la antena.

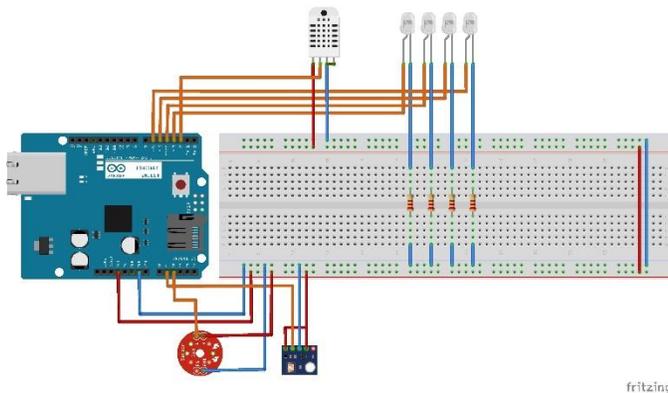


Imagen 6 Diseño del circuito en la herramienta fritzing.

Fuente: Elaboración propia (2024)

Para la implementación del sistema, se han seleccionado sensores y modelos de Arduino específicos por su precisión y durabilidad. Entre ellos se encuentran un sensor para monitorear la humedad del suelo, otro para medir temperatura y humedad ambiental, uno más para la radiación (UV). Estos dispositivos, junto con una plataforma de Arduino, fueron elegidos por su capacidad de adaptarse a diversas aplicaciones, su facilidad de programación y el amplio soporte documental disponible, lo cual es fundamental para gestionar múltiples sensores y

llevar a cabo operaciones de procesamiento de datos en tiempo real.

Para garantizar la precisión de los datos obtenidos, se han implementado varios métodos de validación. Esto incluye una calibración inicial de los sensores en entornos controlados, así como pruebas de campo para comparar las lecturas con registros de aplicaciones WEB. Además, se han desarrollado algoritmos específicos para las características específicas del proyecto. La fase inicial de prueba, que involucró la retroalimentación de empresas y residentes locales, fue crucial para ajustar los parámetros de los sensores y mejorar la fiabilidad del sistema. Estos procedimientos aseguran que los datos recolectados sean precisos y confiables.

### III. RESULTADOS

#### A. Comparación y Análisis con Estudios Similares

Este estudio analiza el uso eficiente de PHP y phpSlim en aplicaciones web, destacando cómo phpSlim mejora la implementación de EndPoints y la integración de bases de datos, abordando áreas que otros estudios no cubren en profundidad. Además, aunque la utilización de Postman para pruebas de APIs es una práctica comúnmente documentada, este estudio ofrece una perspectiva adicional al detallar cómo la configuración específica y el uso de JSON facilitan la verificación de funcionalidades entre el cliente y el servidor. En cuanto a la integración de Arduino con servidores web, se presenta una contribución única al abordar la prevención de valores duplicados, una optimización crucial para la eficiencia en sistemas IoT. En el diseño de interfaces web, se destaca la implementación de visualización en tiempo real y acceso a historiales, ofreciendo funcionalidades avanzadas no exploradas en estudios anteriores. Sin embargo, se identifican limitaciones técnicas y de usabilidad, como la dependencia del hardware de Arduino y los desafíos de compatibilidad de herramientas.

#### B. Codificación del BackEnd

Basado en la etapa anterior, el servidor de pruebas y la ubicación de la base de datos utilizarán PHP como lenguaje de programación, reconocido por su versatilidad. Se empleará un microframework llamado phpSlim, específicamente diseñado para PHP, para implementar los EndPoints del proyecto. Este framework será responsable de la codificación y

# 6

ejecución de cada EndPoint, asegurando así una implementación eficiente y coherente dentro del sistema.

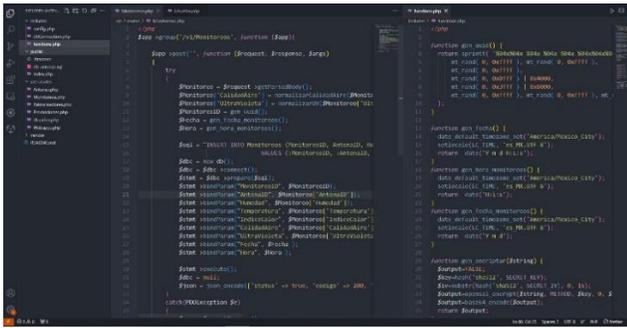


Imagen 7 Codificación en el IDE Visual Studio Code. Fuente: Elaboración propia (2024)

En esta imagen se muestra un fragmento de código donde se llevó a cabo el desarrollo para implementar la funcionalidad del proyecto. El código se escribió en el IDE Visual Studio[10].

Para la codificación de los microservicios, se diseñó la base de datos que será responsable de almacenar y verificar la integridad de los datos de acuerdo con el diseño propuesto para nuestro proyecto.

### C. Pruebas y depuración

Para comprobar la funcionalidad, se utilizó la herramienta Postman. Este IDE permite generar invocaciones a una API y contiene tanto revisión de rutas como una serie de configuraciones para realizar solicitudes al microservicio del servidor, ya sea de manera local o remota. La configuración puede incluir métodos como GET, POST, PUT o DELETE, dentro de los cuales se especifica cómo se transmitirá la solicitud. En nuestro caso, se utilizó JSON para el modelo de request y response del microservicio. Con esta herramienta, podemos verificar la funcionalidad y los requisitos que deben cumplirse entre el cliente y el servidor para realizar la solicitud correspondiente del microservicio solicitado por el cliente.

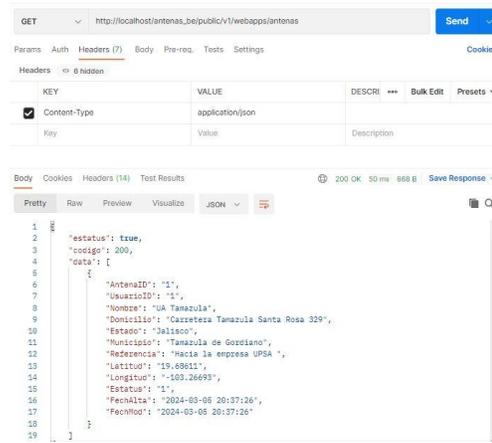


Imagen 8 Pruebas realizadas por el desarrollador del BackEnd Fuente: Elaboración propia (2024)

Esta prueba de conectividad se realizó con la herramienta Postman[9], la cual nos permite indicar la ruta que tendrá dentro de nuestro servidor, ya sea local o global. Además, permite especificar el método por el cual se realiza la consulta de información, así como el formato con el que se transmite la información.

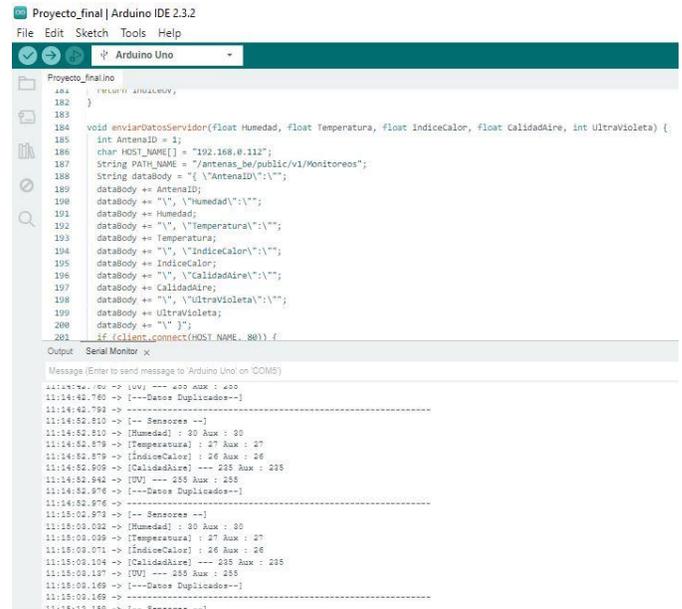


Imagen 9 Comprobando la conectividad del Arduino con el servidor WEB. Fuente: Elaboración propia (2024)

El desarrollador del circuito también comprueba que la comunicación no presente problemas al momento de enviar o recibir respuestas por parte del servidor.

### E. Sitio WEB del usuario final

La siguiente imagen muestra la navegación dentro del sitio web, donde se incluyen diferentes enlaces que permitirán al usuario final acceder a la información que desee visualizar.

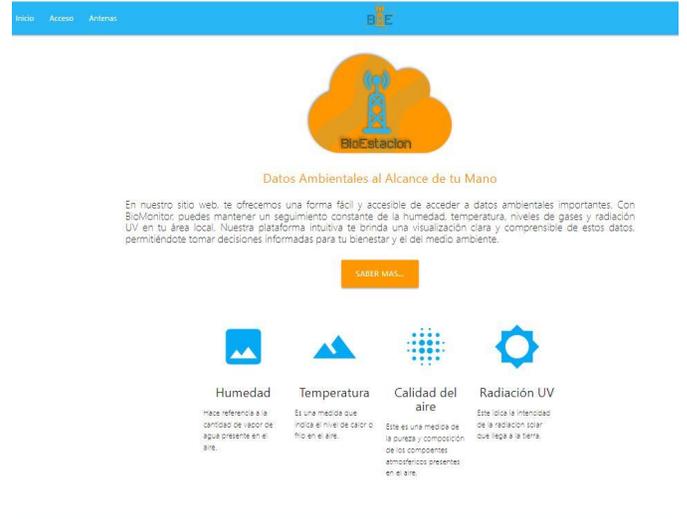


Imagen 12 Diseño final del sitio Web. Fuente: Elaboración propia (2024)

Se implementaron funciones de navegación para el sitio web, entre las cuales se encuentran Inicio, Acceso y Antenas. La pantalla actual muestra la información del sitio web, detallando para qué fue creado y qué actividades podrás realizar en él.



Imagen 13 Diseño final para página de acceso. Fuente: Elaboración propia (2024)

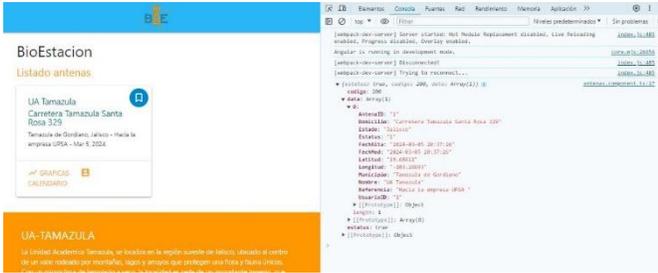


Imagen 10 Pruebas realizados por parte del desarrollador FrondEnd. Fuente: Elaboración propia (2024)

Por otro lado, el desarrollador FrontEnd puede crear la solicitud de acuerdo con las características con las que se creó el microservicio, con el fin de obtener la misma respuesta y lograr la persistencia de datos dentro del sitio web.

### D. Circuito de Arduino

La integración de los componentes en la placa Arduino permite censar los valores y enviarlos al servidor. Es importante señalar que los valores duplicados no se envían al servidor, ya que esos datos ya existen en el sistema.

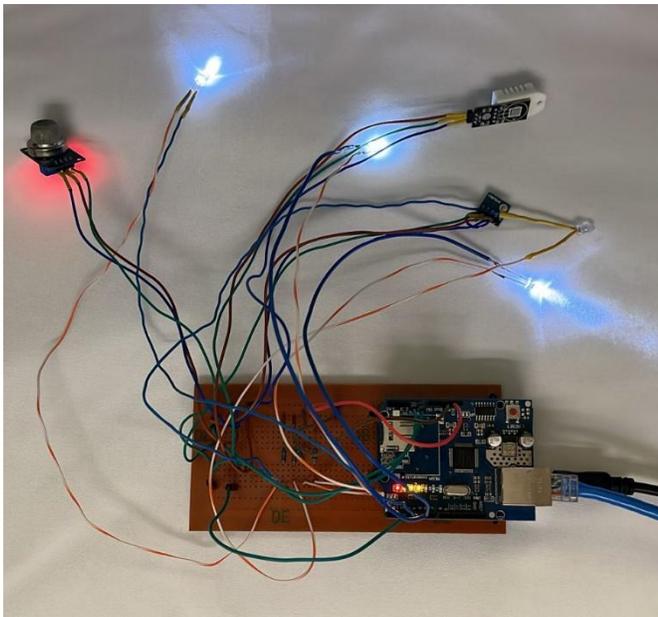


Imagen 11 Prototipo final de la antena. Fuente: Elaboración propia (2024)

En esta imagen, podemos visualizar el resultado final del circuito, así como la forma de conexión implementada en el prototipo de la antena.

En esta página, los usuarios administradores podrán acceder para realizar actividades generales dentro de la plataforma, como agregar más antenas y revisar posibles errores dentro del sitio para garantizar que la información almacenada en el sistema se mantenga sin alteraciones.



Imagen 14 Diseño final de la página antenas.  
Fuente: Elaboración propia (2024)

En esta página se presentan las antenas actualmente registradas y en funcionamiento. Ofrece dos acciones posibles: visualizar gráficas de la información en tiempo real y acceder al calendario para revisar datos almacenados en días anteriores de dicha antena.

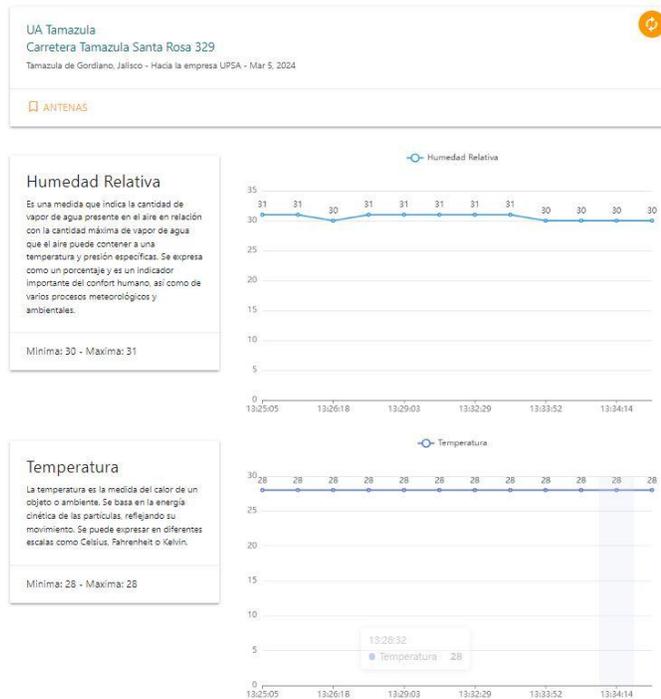


Imagen 15 Graficas de los valores almacenados.  
Fuente: Elaboración propia (2024)

En esta página, los datos almacenados en el servidor se grafican según la fecha en que fueron enviados por parte del microcontrolador Arduino. Puedes analizar las gráficas pasando el cursor sobre ellas para ver los datos almacenados.

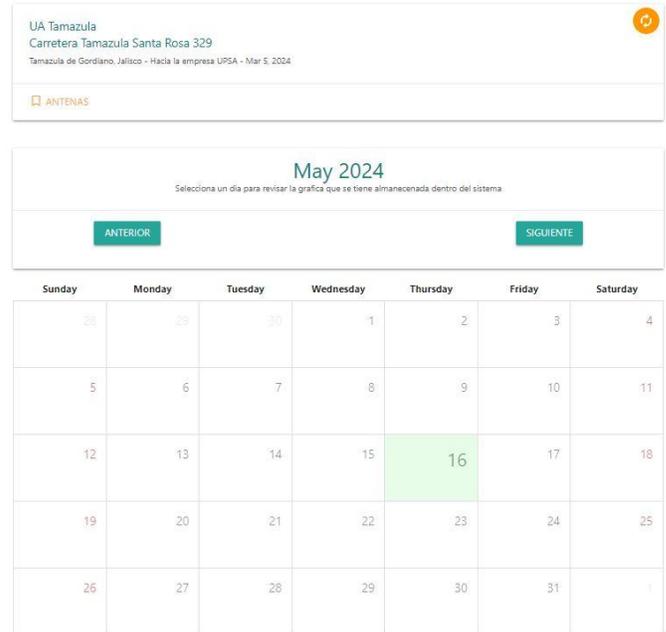


Imagen 16 Calendario del mes para revisar días anteriores.  
Fuente: Elaboración propia (2024)

En esta página se ofrece la posibilidad de visualizar datos almacenados en fechas anteriores

#### IV. CONCLUSIONES

El desarrollo de esta plataforma permite visualizar variables como la temperatura, la humedad, la calidad del aire y los rayos ultravioleta (UV), lo cual ofrece una ventaja significativa en la toma de decisiones. Esto es útil tanto para planificar actividades deportivas al aire libre como para realizar análisis en sistemas de riego implementados en cultivos, mejorando así la programación y el suministro de agua.

La integración de diferentes herramientas y su comunicación efectiva ha sido exitosa, promoviendo la implementación de estos sistemas en la comunidad. Esto beneficia tanto a las personas que practican deportes como a los agricultores de la región, optimizando sus actividades y recursos.

## V. REFERENCES

- [1]. Solis, C. (2015). Manual del Guerrero: AngularJS. Recuperado de [https://www.academia.edu/39532608/Manual\\_del\\_Guerrero\\_AngularJS](https://www.academia.edu/39532608/Manual_del_Guerrero_AngularJS).
- [2]. Ollivier, S., & Pierre-Alexandre, G. U. R. Y. (2016). AngularJS: Desarrolle hoy las aplicaciones web de mañana. Ediciones ENI.
- [3]. González, Y. D., & Romero, Y. F. (2012). Patrón Modelo-Vista-Controlador. Telemática, 11(1), 47-57.
- [4]. Martínez, D. R., Aranda, P. J. V., & Bosch, V. T. (2018). Microservicios Un enfoque integrado. Ra-Ma Editorial.
- [5]. Montávez Sánchez, M. (2023). Arquitectura de programación web: backend.
- [6]. Ocampo, J. P. F., Pazmiño, E. A. A., Meneses, L. J. U., & Esteban, J. A. (2018). Aplicación web con notificaciones mediante el uso del framework de diseño materializecss. Mikarimin. Revista Científica Multidisciplinaria, 4(4), 177-186.
- [7]. A Free Database Designer for Developers and Analysts. (s. f.). <https://dbdiagram.io/d/652420ecffb5169f05563e2>
- [8]. Slim Framework. (2024, 3 marzo). Slim Framework. <https://www.slimframework.com/>
- [9]. Postman API Platform. (s. f.). <https://www.postman.com/>
- [10]. Visual Studio Code - Code editing. Redefined. (2021, 3 noviembre). <https://code.visualstudio.com/>
- [11]. Balsamiq: Fast, focused wireframing for teams and individuals | Balsamiq. (s. f.). <https://balsamiq.com/>
- [12]. Documentation - materialize. (s. f.). <https://materializecss.com/>
- [13]. Fritzing. (2024, 15, marzo). Fritzing IDE. <https://fritzing.org/>