



Arquitectura y diseño de *software* para picosatélites orientado al monitoreo y predicción del comportamiento de incendios forestales (CAFO-2019)

Fecha de recibido: 15 de julio 2023	Fecha de aprobado: 20 de abril 2024
Reception date: July 15, 2023	Approval date: April 20, 2024
Data de recebimento: 15 de julho de 2023	Data de aprovação: 20 de abril de 2024

Ricardo Andrés Santa Quintero

<https://orcid.org/0000-0002-8399-2425>
ricardo.santaq@unilibre.edu.co

Magíster en Tecnologías Digitales Aplicadas a la Educación
 Docente e investigador – Universidad Libre, Colombia
 Rol del investigador: teórico y escritura
 Grupo de Investigación DAVINCIS
 Master's Degree in Digital Technologies Applied to Education
 Teacher and researcher – Universidad Libre, Colombia
 Role of the researcher: theorist and writer
 Davincis Research Group
 Mestrado em Tecnologias Digitais Aplicadas à Educação
 Docente e investigador – Universidad Libre, Colômbia
 Papel do investigador: teórico e escritor
 Grupo de investigação Davincis

Brayan Ricardo Moreno Espinosa

<https://orcid.org/0000-0001-7179-3763>
brayan-morenoe@unilibre.edu.co

Ingeniero de Sistemas
 Investigador – Universidad Libre, Colombia
 Rol del investigador: teórico y escritura
 Grupo de Investigación DAVINCIS
 Systems Engineer
 Researcher – Universidad Libre, Colombia
 Researcher's role: theorist and writer
 Davincis Research Group
 Engenheiro de sistemas
 Investigador – Universidad Libre, Colômbia
 O papel do investigador: teórico e escritor
 Grupo de investigação Davincis

Carlos Leonardo Adames Camargo

<https://orcid.org/0000-0002-2218-6462>
carlosl-adamesc@unilibre.edu.co

Ingeniero de Sistemas
 Investigador – Universidad Libre, Colombia
 Rol del investigador: teórico y escritura
 Grupo de Investigación DAVINCIS
 Systems Engineer
 Researcher – Universidad Libre, Colombia
 Researcher's role: theorist and writer
 Davincis Research Group
 Engenheiro de sistemas
 Investigador – Universidad Libre, Colômbia
 O papel do investigador: teórico e escritor
 Grupo de investigação Davincis

Cómo citar este artículo: Santa Quintero, R. A., Moreno Espinosa, B. R., y Adames Camargo, C. L. (2024). Arquitectura y diseño de *software* para picosatélites orientado al monitoreo y predicción del comportamiento de incendios forestales (CAFO-2019). *Ciencia y Poder Aéreo*, 19(2), 81-94. <https://doi.org/10.18667/cienciaypoderaereo.823>



Arquitectura y diseño de *software* para picosatélites orientado al monitoreo y predicción del comportamiento de incendios forestales (CAFO-2019)

Architecture and software design for picosatellites oriented towards monitoring and predicting forest fire behavior (CAFO-2019)

Arquitetura e design de *software* para picosatélites orientados ao monitoramento e previsão do comportamento de incêndios florestais (CAFO-2019)

Resumen: El proyecto se centra en el desarrollo de un sistema descentralizado para el monitoreo y la predicción de incendios forestales utilizando CanSat. El objetivo principal es facilitar la toma de decisiones oportunas en la lucha contra los incendios forestales, aprovechando las ventajas de la descentralización, la escalabilidad y la seguridad. El trabajo abarca varias etapas, desde el diseño y la construcción del CanSat, hasta el desarrollo del *software* y la definición de la arquitectura de sistema. Se utiliza el microprocesador ESP32 y el lenguaje de programación Arduino para programar el sistema, y se implementa un servidor web para visualizar los datos. El enfoque de microservicios permite una descentralización completa del sistema y una interconexión eficiente entre los nodos. Se destaca la escalabilidad independiente de la tecnología utilizada y se implementan medidas de seguridad para proteger la comunicación y los datos. Los resultados muestran que la implementación del sistema descentralizado cumple con los requisitos de disponibilidad, escalabilidad y seguridad. Se resalta la capacidad de respuesta en tiempo real y el procesamiento distribuido de datos para mejorar la toma de decisiones en la prevención y gestión de incendios forestales.

Palabras clave: arquitectura; incendios; monitoreo; picosatélites; predicción; sistemas descentralizados.

Abstract: The project focuses on the development of a decentralized system for monitoring and predicting forest fires using CanSat. The main objective is to facilitate timely decision-making in the fight against forest fires, leveraging the advantages of decentralization, scalability, and security. The work encompasses several stages, from the design and construction of the CanSat to software development and system architecture definition. The ESP32 microprocessor and Arduino programming language are used to program the system, and a web server is implemented to visualize the data. The microservices approach allows for complete decentralization of the system and efficient interconnection between nodes. The technology-independent scalability is highlighted, and security measures are implemented to protect communication and data. The results show that the implementation of the decentralized system meets the requirements for availability, scalability, and security. The real-time response capability and distributed data processing are emphasized to improve decision-making in forest fire prevention and management.

Keywords: Architecture; fires; monitoring; picosatellites; prediction; decentralized systems.

Resumo: O projeto foca no desenvolvimento de um sistema descentralizado para monitoramento e previsão de incêndios florestais utilizando CanSat. O objetivo principal é facilitar a tomada de decisões oportunas na luta contra os incêndios florestais, aproveitando as vantagens da descentralização, escalabilidade e segurança. O trabalho abrange várias etapas, desde o design e construção do CanSat até o desenvolvimento de *software* e definição da arquitetura do sistema. O microprocessador ESP32 e a linguagem de programação Arduino são usados para programar o sistema, e um servidor web é implementado para visualizar os dados. A abordagem de microsserviços permite uma descentralização completa do sistema e uma interconexão eficiente entre os nós. A escalabilidade independente da tecnologia utilizada é destacada, e são implementadas medidas de segurança para proteger a comunicação e os dados. Os resultados mostram que a implementação do sistema descentralizado atende aos requisitos de disponibilidade, escalabilidade e segurança. A capacidade de resposta em tempo real e o processamento distribuído de dados são destacados para melhorar a tomada de decisões na prevenção e gestão de incêndios florestais.

Palavras-chave: Arquitetura; incêndios; monitoramento; picosatélites; previsão; sistemas descentralizados.

Introducción

El agotamiento de los recursos naturales, la deforestación, el cambio climático y la contaminación del aire han generado una preocupación creciente debido a su grave impacto en la frecuencia y la propagación de los incendios forestales. Estos fenómenos han causado una disminución en la disponibilidad de agua y otros elementos vitales para la supervivencia de los bosques, así como un aumento en la susceptibilidad a los incendios debido a la pérdida de cobertura vegetal protectora en áreas deforestadas. Además, el cambio climático ha alterado los patrones climáticos, creando condiciones más favorables para la aparición y la intensidad de los incendios forestales. A su vez, la contaminación del aire contribuye a la acumulación de materiales inflamables y aumenta la probabilidad de que los incendios se desencadenen y se extiendan rápidamente (Instituto de Hidrología, Meteorología y Estudios Ambientales [Ideam], s. f.).

Tales desafíos requieren la implementación de sistemas estratégicos que puedan monitorear y predecir el comportamiento de los incendios forestales de manera precisa y oportuna. La idea de estos sistemas es que permitan recopilar datos relevantes como la temperatura, la humedad, la velocidad del viento y la densidad de la vegetación, con el fin de evaluar el riesgo de incendios en áreas específicas. Mediante el análisis de estos datos, es posible identificar patrones, detectar incendios en etapas tempranas y predecir su propagación, lo cual permite una respuesta más eficiente y efectiva por parte de los equipos de prevención y extinción.

La implementación de picosatélites con un objetivo similar pauta las bases para la aplicación de estos dispositivos mediante la recopilación de datos climatológicos como la medición de temperatura, presión y altitud; esto, a partir del uso de tecnologías como la programación en Arduino y el desarrollo de un dispositivo que sea funcional para realizar pruebas en la ciudad de Bogotá y algunos municipios aledaños. Este es uno de los proyectos a escala real manifestados en Colombia en el año 2022 (Cagua-Colmenares y Vinche-Hernández, 2022).

La implementación de sistemas de monitoreo y predicción de incendios forestales es fundamental para proteger los ecosistemas naturales y garantizar la seguridad de las comunidades. Estos sistemas brindan una visión integral del comportamiento de los incendios, lo cual facilita la toma de decisiones informadas y la asignación adecuada de recursos para combatirlos de manera más eficiente. Al anticipar la propagación de los incendios, se pueden tomar medidas preventivas, evacuar áreas en riesgo y minimizar los daños materiales y humanos causados por estos desastres naturales.

Uno de los principales sistemas de predicción es el Sistema de Predicción de Peligro de Incendios Forestales (SPPIF), implementado en México. Desarrollado a partir del proyecto Conacyt-Conafor 2014-2-252620, es una herramienta focalizada también en la toma oportuna de decisiones para contrarrestar los incendios forestales surgidos en México. Aquel sistema, desarrollado por la Universidad Juárez del Estado de Durango, se encarga de evaluar los conglomerados de puntos de calor e incendios forestales mediante un mapeo geográfico (Secretaría de Medio Ambiente y Recursos Naturales, 20 de diciembre de 2020). La información es captada por unos satélites (Conabio) que están directamente relacionados con la herramienta, lo cual permite una actualización en tiempo casi real. Se logra por medio de un sensor AVHRR (Advanced Very High Resolution Radiometer) de los satélites NOAA (National Oceanic and Atmospheric Administration) y del sensor MODIS (Moderate Resolution Imaging Spectroradiometer) usados en los satélites Terra-1 y Aqua-1, y del sensor VIIRS (Visible and Infrared Imaging Radiometer Suite) del satélite Suomi-NPP.

Con dichos sensores, lograron fabricar una herramienta a escala real que tuviera cobertura en todo el país. Así mismo, fue la inspiración del proyecto para conceptualizar un sistema de monitoreo más focalizado y a una escala menor implementando la innovación aeroespacial.

En consecuencia, se ha identificado una oportunidad para asentar sistemas de monitoreo y predicción del comportamiento de los incendios forestales. Para ello, el grupo de investigación DAVINCI, perteneciente

a la Universidad Libre, estuvo a cargo del proyecto de investigación titulado “Perspectivas de los servicios TI e ingeniería del *software* en *blockchain*, VR y picosatélites”, en el cual se propone el uso de picosatélites en un sistema descentralizado, conocidos como CanSat. Estos dispositivos, con dimensiones específicas (66 mm de diámetro, 115 mm de altura y una masa de 300-350 gramos), están equipados con sensores climatológicos para recopilar datos sobre factores ambientales relevantes (Naylamp Mechatronics, 2016).

Además de la recopilación de datos, se considera la funcionalidad de su transmisión y su almacenamiento, ya que los factores mencionados aumentan la amenaza de los incendios forestales, y su frecuencia y gravedad. Ante esta situación, resulta crucial implementar sistemas estratégicos de monitoreo y predicción que permitan prevenir y mitigar los efectos destructivos de los incendios. Estos sistemas recopilan datos precisos sobre factores ambientales relevantes, y, mediante un análisis detallado, permiten identificar patrones, detectar incendios en etapas tempranas y predecir su propagación.

El uso de picosatélites en un sistema descentralizado, como los CanSat, ofrece una solución efectiva, dado que recopilan datos climatológicos y facilitan la transmisión y el almacenamiento de información. La instauración de estos sistemas de monitoreo y predicción es fundamental para proteger los ecosistemas naturales, garantizar la seguridad de las comunidades y permitir una respuesta más eficiente y efectiva frente a los incendios forestales. Con todo esto, se protegen los ecosistemas naturales y se garantiza la seguridad de las comunidades afectadas por los incendios forestales.

Materiales y métodos

Para el desarrollo del sistema descentralizado, se seleccionaron los siguientes componentes:

Para la fabricación de los CanSat, se emplearon los siguientes sensores:

1. **ESP32:** El módulo ESP32 es una solución de Wi-Fi/Bluetooth todo en uno, integrada y certificada que proporciona no solo la radio inalámbrica, sino también un procesador integrado con interfaces para conectarse con varios periféricos. Este microcontrolador será la base para la implementación del servidor web del CanSat junto a los demás componentes de factores ambientales (Beningo, 21 de enero de 2020)
2. **MQ135:** Sensor implementado en la detección de gases peligrosos y en controladores de la calidad del aire. Este sensor es capaz de detectar un amplio rango de gases que incluye: NH₃, NOx, alcohol, benceno, humo y CO₂ (PalcoElectronica, s. f.).
3. **MPU:** Es una unidad de medición inercial o IMU de seis grados de libertad, pues combina un acelerómetro de tres ejes y un giroscopio de tres ejes (Naylamp Mechatronics, 2016).
4. **BMP180:** Permite medir la altura respecto al nivel del mar, su funcionamiento está basado en la relación entre la presión de aire y la altitud (Naylamp Mechatronics, s. f.).
5. **NEO06MV2:** Es un módulo GPS y se utiliza para la navegación. El módulo simplemente verifica su ubicación en la tierra y proporciona datos de salida como la altura sobre el nivel del mar, la longitud y la latitud de su posición, fecha y hora GMT.

Una vez mencionados los componentes clave utilizados en los CanSat, es importante comprender cómo funcionan en conjunto para recopilar datos y realizar tareas específicas.

El módulo ESP32 actúa como el cerebro del CanSat, gestionando la comunicación con los diferentes sensores y periféricos. Este módulo recopila los datos obtenidos por sensores como el MQ135, que detecta gases peligrosos y controla la calidad del aire, y el BMP180, que mide la altura en relación con el nivel del mar. Estos datos son procesados y almacenados en la base de datos de Firebase, que proporciona la capacidad de almacenamiento y disponibilidad necesaria para el sistema.

Además de la recopilación de datos, el CanSat utiliza el sensor MPU (que mide el movimiento y la orientación) para obtener información precisa sobre su posición en el espacio. Esto se combina con el módulo receptor NEO06MV2, que permite la geolocalización del CanSat utilizando señales de satélites GPS.

El CanSat, equipado con estos componentes, se convierte en un nodo independiente capaz de recopilar datos ambientales, medir la calidad del aire, determinar la posición geográfica y transmitir toda esta información a través de su servidor web implementado en el módulo ESP32. Los CanSat pueden interactuar entre sí, compartiendo datos y actuando como canales de comunicación para el almacenamiento de información.

En conjunto, dichos componentes permiten la recopilación de datos ambientales relevantes, la geolocalización y la comunicación eficiente entre los nodos CanSat. Esta integración de servicios independientes en un sistema descentralizado ofrece una visión más completa y precisa del comportamiento de los incendios forestales, facilitando la toma de decisiones informadas y la respuesta efectiva ante situaciones de riesgo (Jiménez, 15 de enero de 2022).

La elección de SCRUM se basa en su enfoque de obtención de resultados a corto y mediano plazo. Se sigue un enfoque de desarrollo incremental en bloques temporales cortos y fijos. Las fases del proyecto incluyen la planificación (Product Backlog), donde se establecen las tareas prioritarias; la ejecución (Sprint), donde se determina el intervalo de tiempo para el desarrollo y la implementación; y el control (Burn Down), donde se mide el progreso del proyecto (Raeburn, 25 de agosto de 2022).

Para el desarrollo del proyecto de monitoreo y predicción de incendios forestales utilizando picosatélites en un sistema descentralizado, se puede aplicar la metodología ágil SCRUM planteada de la siguiente forma:

- Planificación del proyecto: Se crea el Product Backlog, que contiene las funcionalidades y los requisitos del sistema, como la recolección de datos ambientales, la comunicación entre los picosatélites y el almacenamiento de información.

- Sprint: Se establecen Sprints de corta duración; por ejemplo, de dos semanas. Durante cada Sprint, se seleccionan las funcionalidades prioritarias del Product Backlog y se establece un objetivo específico para el Sprint.
- Reuniones diarias de seguimiento: Se llevan a cabo reuniones diarias de corta duración para que el equipo revise el progreso, discuta los obstáculos y planifique las tareas diarias. Esto permite mantener la colaboración y la transparencia dentro del equipo.
- Desarrollo iterativo: Durante cada Sprint, el equipo desarrolla las funcionalidades seleccionadas, como la utilización de los sensores y la configuración de la comunicación entre los picosatélites. Al finalizar cada Sprint, se debe tener una funcionalidad completa y lista para su revisión.
- Revisión del Sprint: Al finalizar cada Sprint, se realiza una reunión de revisión del Sprint, en la que se muestra el trabajo completado y se recopila el *feedback* del cliente y los *stakeholders*. Esto permite evaluar el progreso y realizar ajustes si es necesario.
- Control y mejora continua: Se utiliza un Burn Down Chart para visualizar el progreso del proyecto y garantizar que se están cumpliendo los objetivos establecidos (Rocha-Garrido, 15 de febrero de 2022). Además, se realiza una retrospectiva al final de cada Sprint para identificar áreas de mejora y planificar acciones correctivas.

En el contexto de este proyecto, la metodología SCRUM permite un enfoque iterativo e incremental en el desarrollo del sistema descentralizado. El equipo trabaja en ciclos cortos para entregar incrementos funcionales que incluyen la recopilación de datos ambientales, la comunicación entre los picosatélites y el almacenamiento de información. Esta metodología fomenta la flexibilidad, la adaptabilidad y la entrega continua de valor, lo cual es fundamental en un proyecto que busca desarrollar un sistema eficiente y escalable para el monitoreo y la predicción de incendios forestales.

Resultados

Etapa 1. Diseño y construcción del picosatélite

Durante la primera etapa del proyecto, se llevó a cabo el diseño y la construcción del picosatélite, centrado especialmente en el desarrollo lógico del circuito. Esta etapa fue de suma importancia, ya que se estableció la conexión adecuada de los componentes que se utilizarán en la fabricación del CanSat, garantizando así su funcionamiento óptimo. En este contexto, es importante resaltar el papel fundamental desempeñado por el ESP32, un microprocesador versátil y potente que actúa como la base central del CanSat. El ESP32 asume múltiples funciones clave en el proyecto, entre las que se destacan dos de ellas en particular (Beningo, 21 de enero de 2020).

En primer lugar, el ESP32 se encargará de alojar un servidor web que permitirá la comunicación y el intercambio de datos con otros dispositivos o sistemas externos. Esta capacidad de comunicación es esencial para el éxito del CanSat, ya que facilita la transferencia de información relevante y posibilita su integración en redes de monitoreo más amplias.

En segundo lugar, el ESP32 asume la responsabilidad de administrar los sensores ambientales. Estos sensores son fundamentales para el monitoreo de incendios forestales, ya que recolectan los datos necesarios para su detección y seguimiento. Gracias a la capacidad de procesamiento del ESP32, se puede realizar un análisis en tiempo real de los datos recopilados, lo cual permite tomar medidas rápidas y efectivas en caso de emergencia.

La elección del ESP32 como microprocesador central del CanSat es estratégica, debido a sus características sobresalientes. Además de su versatilidad y potencia, el ESP32 ofrece una gran capacidad de procesamiento, lo que le permite manejar múltiples tareas de manera eficiente. Asimismo, cuenta con una conectividad sólida que facilita la interacción con otros dispositivos y sistemas externos (Beningo, 21 de enero de 2020).

Al utilizar el ESP32 como componente central, se logra una integración eficiente de los sensores y un control centralizado para el funcionamiento del sistema. Esto garantiza un desempeño óptimo del CanSat y abre la posibilidad de futuras mejoras y ampliaciones en función de las necesidades específicas del proyecto.

En resumen, la primera etapa del proyecto se enfocó en el diseño y la construcción del picosatélite, con énfasis en el desarrollo lógico del circuito. El ESP32 desempeña un papel fundamental como microprocesador central, al encargarse de funciones clave como la comunicación, el intercambio de datos y la administración de los sensores ambientales. Su elección estratégica se basa en su capacidad de procesamiento, conectividad y flexibilidad, que permite una integración eficiente y un control centralizado del sistema.

En la segunda parte de la primera etapa del proyecto, se realiza la construcción física del CanSat utilizando los componentes previamente mencionados. En esta etapa, se lleva a cabo la integración de una *proto-board*, la cual desempeña un papel fundamental al servir como base para la conexión del circuito (ver Figura 1).

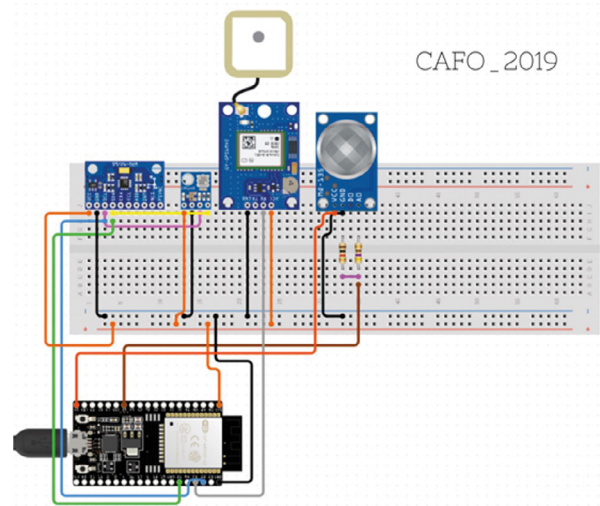


Figura 1. Diseño lógico elaborado en la herramienta Circuit.io
Fuente: elaboración propia.

La *proto-board*, también conocida como placa de pruebas, es un dispositivo que permite la conexión y la interconexión de los componentes electrónicos de manera temporal. Se caracteriza por proporcionar una

superficie con una matriz de orificios en la que se pueden insertar los componentes y realizar las conexiones necesarias para el funcionamiento del CanSat (330ohms, 2 de marzo de 2016).

En la fase de prototipado, la utilización de una *protoboard* en la construcción del CanSat ofrece numerosas ventajas. En primer lugar, facilita la tarea de ensamblar los componentes de manera ordenada, lo cual resulta fundamental para garantizar la correcta conexión entre ellos. Al contar con una superficie organizada y con los orificios dispuestos estratégicamente, se logra una disposición estructurada que facilita el proceso de ensamblaje y minimiza los posibles errores de conexión.

Además, la *protoboard* brinda una mayor flexibilidad en el proceso de construcción del CanSat. Al ser una solución temporal, permite realizar modificaciones y ajustes en el circuito de forma más sencilla, sin la necesidad de soldar los componentes de manera permanente. Esto resulta especialmente útil durante la etapa de pruebas y validación del sistema, ya que se pueden realizar cambios y mejoras de forma ágil y rápida.

Es importante tener en cuenta que, aunque la *protoboard* es una solución temporal para la construcción del prototipo del CanSat, en etapas posteriores del proyecto se podría considerar la fabricación de una placa de circuito impreso (PCB) personalizada. La PCB es una placa que se diseñaría específicamente para el CanSat, con las conexiones y los componentes soldados de forma permanente (Shamkhalichenar *et al.*, 2020). Esto proporcionaría una mayor eficiencia y robustez al diseño, pues elimina las posibles conexiones inestables y reduce el tamaño del sistema.

En resumen, en la segunda parte de la primera etapa del proyecto se lleva a cabo la construcción física del CanSat, como se puede observar en la Figura 2, y se destaca la integración de una *protoboard* como base para la conexión del circuito.

La *protoboard* facilita el ensamblaje ordenado de los componentes y asegura las conexiones correctas, además de ofrecer flexibilidad durante el proceso de pruebas y ajustes. En etapas posteriores, se podría considerar la fabricación de una PCB personalizada para mejorar la eficiencia y la robustez del diseño.

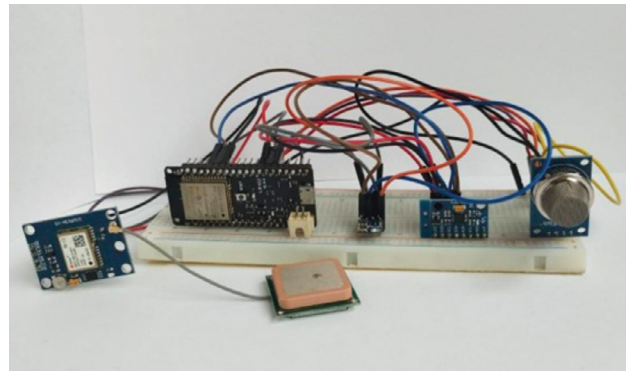


Figura 2. Construcción del circuito
Fuente: elaboración propia.

Etapa 2. Desarrollo del *software*, análisis de patrones arquitectónicos y arquitectura de sistema

En la segunda etapa del proyecto, se hace énfasis en la implementación del *software* necesario para el funcionamiento del CanSat, así como en el análisis y el diseño de los patrones arquitectónicos y la arquitectura de sistema.

En primer lugar, se lleva a cabo el desarrollo del *software* que permitirá el control y la gestión del CanSat. Esto implica la programación de algoritmos y la creación de interfaces de usuario que faciliten la interacción con el sistema. Además, se incluye la configuración de la comunicación con otros dispositivos o sistemas externos, lo que posibilita la transmisión y recepción de datos relevantes.

Durante esta etapa, es fundamental realizar un análisis detallado de los patrones arquitectónicos apropiados para el proyecto. Los patrones arquitectónicos son soluciones probadas y comprobadas que ayudan a estructurar y organizar el *software* de manera eficiente. Se estudian diferentes patrones, como el modelo-vista-controlador (MVC), el modelo de capas y otros, con el objetivo de seleccionar el más adecuado para el CanSat.

Además, se realiza el diseño de la arquitectura de sistema del proyecto. Esto implica definir las estructuras, los componentes y las relaciones que conformarán el sistema en su conjunto. Se analizan los requisitos funcionales y no funcionales, así como las restricciones

y las necesidades específicas del proyecto, para establecer una arquitectura que garantice un rendimiento óptimo y una escalabilidad adecuada.

Durante el desarrollo del *software*, el equipo se enfoca en garantizar la calidad del código y la funcionalidad del sistema. Se realizan pruebas exhaustivas para validar el correcto funcionamiento de todas las operaciones, y para identificar y corregir posibles errores o fallos. También, se presta especial atención a la seguridad del sistema, implementando medidas de protección y asegurando la integridad de los datos.

Se inicia el desarrollo del *software* utilizando el lenguaje de programación Arduino. Esta elección se basa en su compatibilidad con el microprocesador ESP32 y su amplia comunidad de desarrolladores que brindan soporte y recursos adicionales. En esta etapa, se procede a construir el código necesario para programar el ESP32 y permitir su interacción con los sensores conectados. El objetivo principal es coordinar la adquisición de datos provenientes de los sensores y hacer que estos sean visibles a través de una interfaz web.

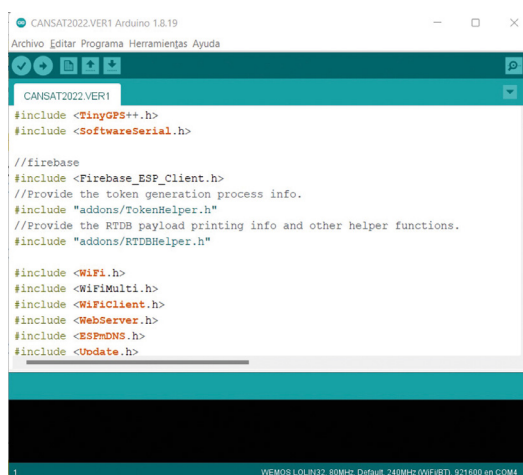
El código comienza por configurar las interfaces de comunicación apropiadas para establecer la conexión con los sensores. Esto puede incluir el uso de protocolos como I2C, SPI o UART, dependiendo de los sensores utilizados. A continuación, se implementan las instrucciones necesarias para leer los valores de los sensores y almacenarlos en variables.

Además, se consideran aspectos como el muestreo de datos, la frecuencia de lectura de los sensores y la gestión de eventos para garantizar la captura precisa y oportuna de los datos ambientales relevantes. Se aplican técnicas de programación eficientes para optimizar el rendimiento del sistema y asegurar una respuesta rápida a los cambios en las condiciones ambientales.

La implementación del servidor web es otro componente importante en el desarrollo del *software*. Se utilizan librerías específicas de Arduino para crear un servidor web que permita visualizar los datos recopilados a través de una interfaz amigable. Esto implica establecer las rutas y páginas web necesarias para mostrar los valores de los sensores y permitir su acceso desde dispositivos externos. A medida que se avanza en el desarrollo del *software*, se realizan pruebas

minuciosas para validar la funcionalidad y el rendimiento del código. Se verifican los valores leídos de los sensores, se comprueba la estabilidad del servidor web y se evalúa la respuesta del sistema ante diferentes situaciones (Quisilema-Carrera, 2022).

Es importante mencionar que el uso del lenguaje de programación Arduino facilita el desarrollo del *software*, ya que proporciona una sintaxis sencilla y una amplia gama de bibliotecas y ejemplos disponibles (Fernández, 23 de septiembre de 2022). Esto agiliza la implementación de las funcionalidades necesarias y permite un desarrollo más eficiente.



```

CANSAT2022.VER1
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

//firebase
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

#include <WiFi.h>
#include <WiFiMulti.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <Update.h>

```

Figura 3. Estructura lógica en Arduino

Fuente: elaboración propia.

Durante esta etapa, se realiza el establecimiento de un servidor web en el ESP32 a fin de crear una interfaz intuitiva y accesible para visualizar los datos recopilados. Con tal objetivo, se lleva a cabo la configuración de rutas y puntos finales que permitirán la correcta comunicación entre el servidor y el navegador. Además, se define el formato y el diseño de la página web, asegurándose de que los elementos de la interfaz sean visualmente atractivos y faciliten la comprensión de los datos en tiempo real.

Así mismo, la estructuración de un servidor web es primordial para brindar una experiencia interactiva al usuario. Una vez que el servidor está en funcionamiento y el código del ESP32 ha sido programado para adquirir y almacenar los datos provenientes de los

sensores, es posible acceder a la interfaz web desde cualquier navegador compatible. A través de esta interfaz, se presenta de manera clara y comprensible la información recopilada por los sensores, lo que permite una fácil interpretación y análisis de los datos.

Es importante destacar que la interfaz web se desarrolla de manera cuidadosa, considerando aspectos de usabilidad y experiencia del usuario. Se busca que la visualización de los datos sea intuitiva y que la navegación por la interfaz sea fluida y amigable. Además, se pueden incorporar gráficos, tablas y otros elementos visuales que ayuden a representar de forma efectiva los datos recopilados.

Como bien se tiene claro, la instauración exitosa del servidor web y la interfaz de visualización de datos constituyen un hito importante en el proyecto. A través de esta etapa, se logra proporcionar a los usuarios una herramienta eficaz para monitorear y analizar la información relevante sobre incendios forestales recopilada por los picosatélites. Esto contribuye a la creación de un sistema robusto y funcional que cumple con los objetivos establecidos, brindando información valiosa para la toma de decisiones y la prevención de desastres.

En la Figura 4, durante el proceso de análisis de patrones arquitectónicos, se llevó a cabo un riguroso estudio de diferentes enfoques para determinar la mejor estructura del sistema. Se exploraron varias alternativas, considerando aspectos como la escalabilidad, la arquitectura, la modularidad, la mantenibilidad y otros requisitos específicos del proyecto.

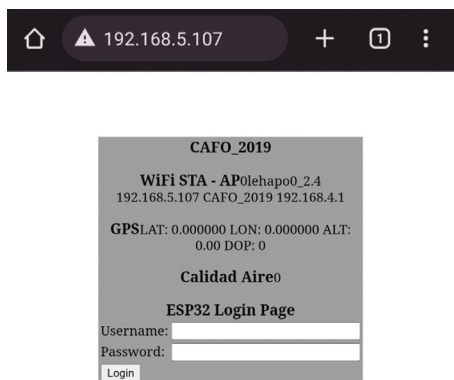


Figura 4. Presentación de la interfaz del servidor web
Fuente: elaboración propia.

Entre los patrones arquitectónicos considerados se encontraban la arquitectura de tres capas, la arquitectura cliente-servidor, la arquitectura orientada a servicios y el patrón de microservicios, entre otros. Cada uno de estos patrones fue evaluado en términos de su aplicabilidad, sus beneficios y sus limitaciones para el contexto del proyecto.

Después de un análisis detallado, se determinó que el patrón de microservicios era el más adecuado para el desarrollo del sistema. Sin embargo, antes de tomar esta decisión, se presentó un modelo inicial de arquitectura en capas como ejemplo, que sirvió como punto de partida para comprender las interacciones entre los diferentes componentes del sistema.

Este modelo inicial, basado en una arquitectura en n capas, proporcionó una visión general de cómo los distintos módulos y componentes se relacionarían entre sí. Se establecieron las capas de presentación, la lógica de negocio y el acceso a datos, definiendo las responsabilidades y las interacciones entre ellas. Este enfoque de múltiples capas fue considerado como un punto de partida para el diseño de la arquitectura final del sistema, brindando una estructura inicial que se podía adaptar y evolucionar según las necesidades y los requisitos específicos del proyecto.

En la Figura 5, se presenta la estructura y las características del *software* en los niveles 3 y 4 del sistema. Estos niveles son fundamentales para el funcionamiento y la distribución del sistema, y engloban aspectos clave como la adquisición de datos, el procesamiento de la información y el almacenamiento de acuerdo con los criterios y requisitos establecidos durante la segunda etapa del desarrollo del *software*.

En el nivel 4, se enfoca en la adquisición de datos, es decir, en la recolección y captura de información relevante proveniente de los sensores ubicados en los picosatélites. Este proceso se lleva a cabo de manera coordinada y eficiente, garantizando la recopilación precisa y confiable de los datos ambientales necesarios para el monitoreo y la predicción de incendios forestales. Se establecen mecanismos para la comunicación y sincronización entre los picosatélites (SAJR, CAFO METEO) y el servidor central, asegurando la transmisión segura y oportuna de los datos adquiridos.

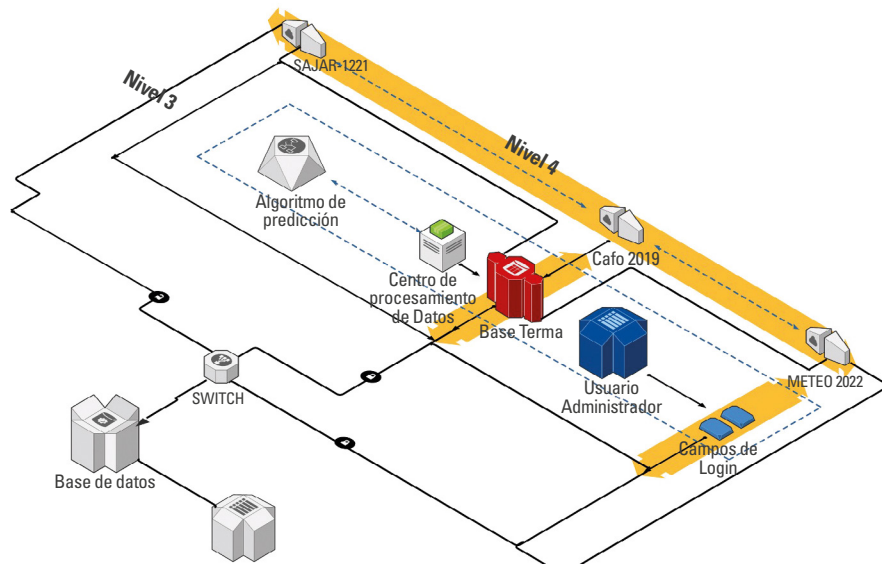


Figura 5. Diagrama de componentes de la arquitectura inicial
Fuente: elaboración propia.

En el nivel 3, se abordan el procesamiento y el almacenamiento de los datos recolectados. Aquí, se aplican algoritmos y técnicas específicas para analizar y transformar la información adquirida en concimientos relevantes. Este procesamiento puede incluir tareas como la detección de patrones, la generación de modelos predictivos y la identificación de situaciones anómalas. Además, se establecen mecanismos para el almacenamiento seguro y eficiente de los datos procesados, asegurando su disponibilidad y preservación a largo plazo.

Es importante destacar que la estructuración y caracterización del *software* en los niveles 3 y 4 del sistema se lleva a cabo en estrecha relación con los criterios y requisitos establecidos durante la segunda etapa del desarrollo del *software*. Estos criterios y requisitos son el resultado de un análisis exhaustivo de las necesidades del proyecto, considerando aspectos como la escalabilidad, la confiabilidad, la seguridad y el rendimiento del sistema, tal como se ve en la Figura 6.

En los niveles 4 y 5, se establece la capa de procesos o de negocio, donde se distribuye y se caracteriza

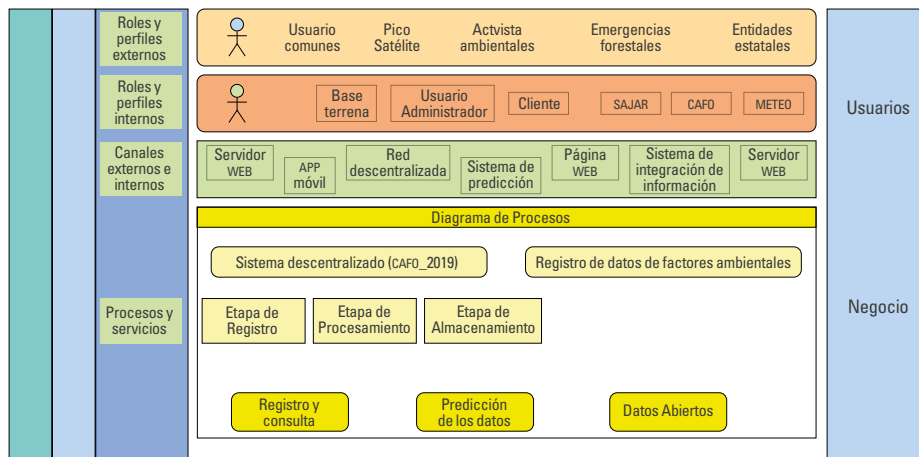


Figura 6. Estructura de diagrama de microservicios
Fuente: elaboración propia.

el *software*. Se pone un énfasis particular en el funcionamiento del sistema, incluyendo la toma de datos, el procesamiento de estos y su respectivo almacenamiento, todo ello basado en los criterios y requerimientos establecidos durante el desarrollo del *software* en la segunda etapa.

En esta capa, se define cómo se capturan los datos necesarios para el monitoreo de incendios forestales, ya sea a través de los sensores incorporados en los CanSat o mediante otras fuentes de información relevante. Luego, se realiza el procesamiento de estos datos, aplicando algoritmos y técnicas pertinentes para obtener información valiosa sobre el comportamiento de los incendios. Además, se establecen los criterios y

requisitos específicos para el almacenamiento de los datos procesados. Esto puede incluir consideraciones de seguridad, integridad y escalabilidad, asegurando que la información esté disponible y sea confiable para su posterior análisis y toma de decisiones.

Sin embargo, tras un análisis más profundo y considerando las ventajas inherentes al patrón de microservicios, se decidió adoptar este enfoque arquitectónico en lugar del modelo de múltiples capas inicialmente propuesto. El patrón de microservicios demostró ser más adecuado para el sistema, pues brinda una mayor flexibilidad, escalabilidad y modularidad, y permite el desarrollo, el despliegue y la gestión independiente de cada microservicio, como lo denota la Figura 7.

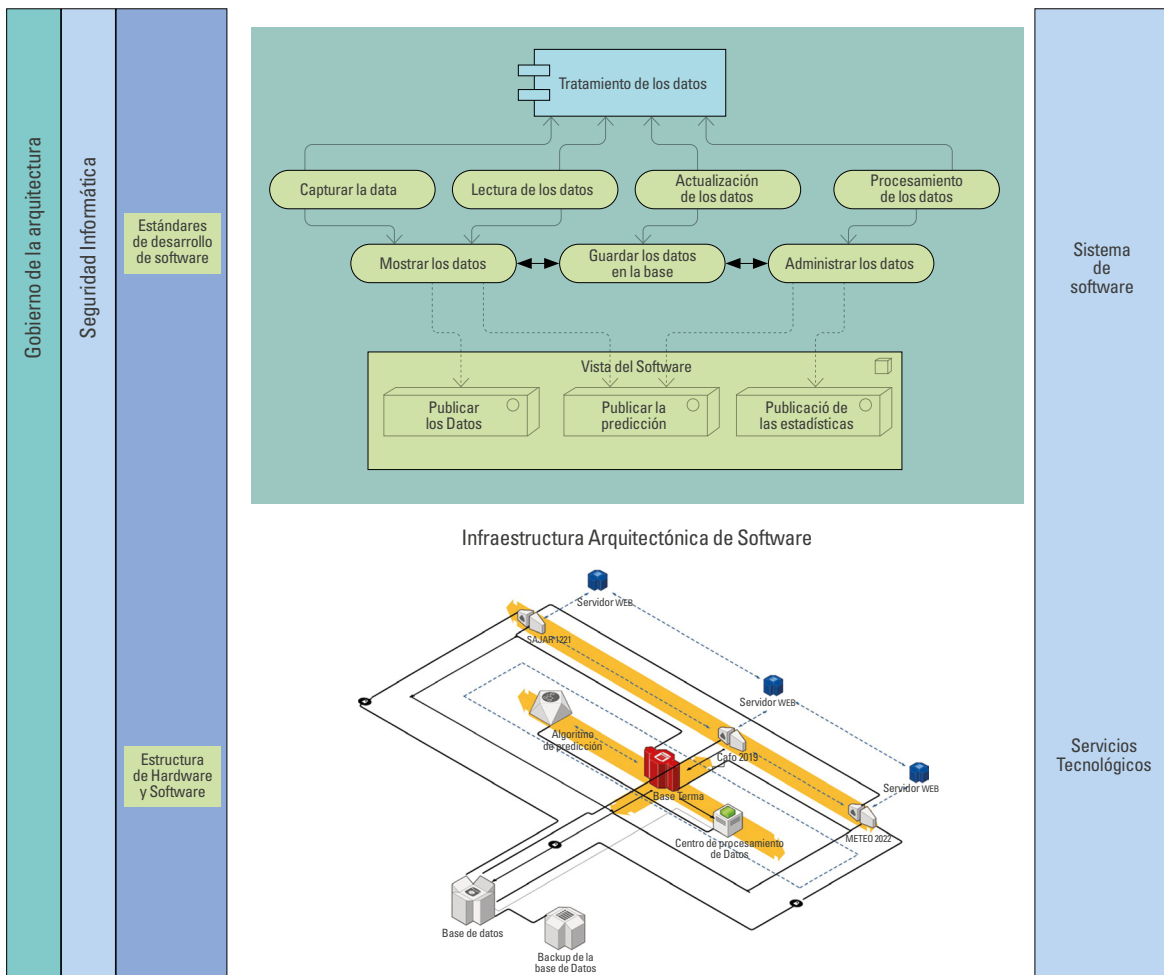


Figura 7. Diagrama de arquitectura final
Fuente: elaboración propia.

En conclusión, se realizó un estudio intensivo de diferentes patrones arquitectónicos, incluyendo un modelo inicial de arquitectura en capas. Sin embargo, tras evaluar las ventajas y desventajas de cada opción, se optó por utilizar el patrón de microservicios debido a su idoneidad para cumplir con los requisitos de escalabilidad, modularidad y mantenibilidad del proyecto. Este enfoque permitió una mayor flexibilidad en el diseño y desarrollo del sistema, brindando una arquitectura robusta y adaptada a las necesidades específicas del monitoreo y la predicción de incendios forestales mediante picosatélites en un sistema descentralizado.

Implementación del patrón arquitectónico

En el marco de este proyecto, se ha realizado un exhaustivo análisis para identificar el patrón arquitectónico más adecuado, y se ha determinado que el enfoque de microservicios es el más apropiado para alcanzar los objetivos planteados. Este patrón arquitectónico permite una descentralización completa del sistema, donde cada nodo desempeña un papel específico y se comunica eficientemente con otros nodos.

El uso de microservicios en los CanSat implica que cada uno de ellos cumple funciones clave y establece conexiones con otros nodos de forma efectiva. Estas conexiones no solo se limitan a la transferencia de datos, sino que también facilitan la interacción con otros componentes esenciales, como la base terrena y la base de datos.

La adopción de esta arquitectura promueve una mayor interconexión entre los nodos, lo cual contribuye a satisfacer los atributos de calidad establecidos para el proyecto, como la seguridad, la escalabilidad y la disponibilidad. Los microservicios permiten un diseño modular y flexible, donde cada componente tiene la capacidad de escalar de forma independiente según las necesidades específicas, garantizando así la disponibilidad y el rendimiento óptimo del sistema.

Además, la arquitectura de microservicios facilita la implementación de medidas de seguridad en cada uno de los servicios. Esto asegura que la comunicación y los datos se mantengan protegidos en todo momento, algo crucial para la integridad y confidencialidad de la información recopilada y transmitida por los CanSats. Por ende, se muestra en la Figura 8 cómo sería la implementación del *software* con la arquitectura de microservicios.

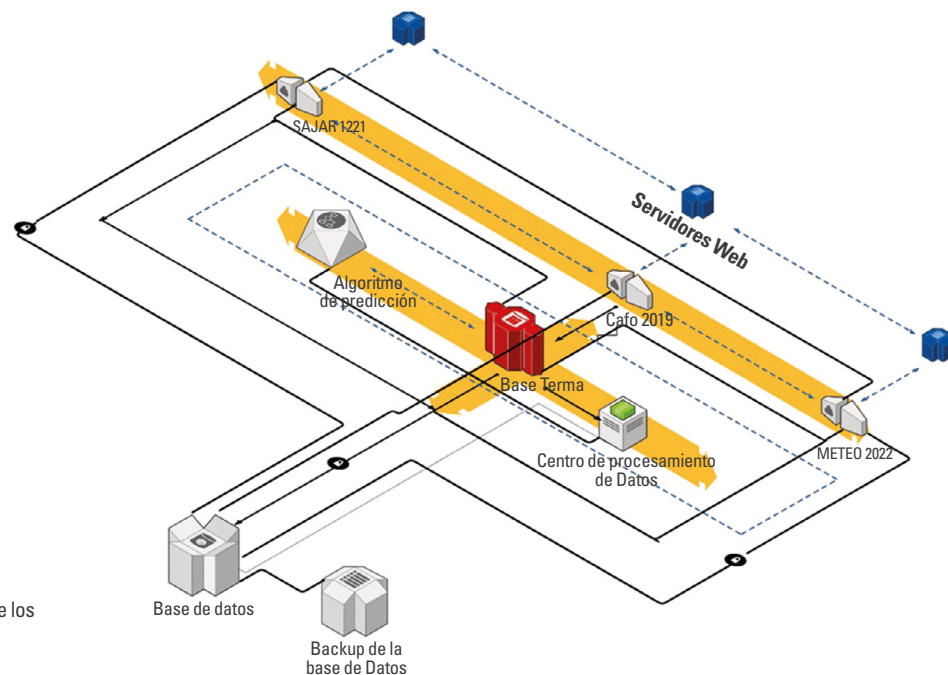


Figura 8. Diseño arquitectónico de los componentes de microservicios

Fuente: elaboración propia.

Discusión

Los sistemas descentralizados brindan una oportunidad única para esta investigación, ya que presentan fortalezas notables en términos de seguridad, escalabilidad y disponibilidad. Tales características demuestran su capacidad de implementación en diversos contextos, ya que permiten administrar de manera eficiente las dependencias y establecer conexiones entre los nodos para lograr un flujo de datos óptimo en tiempo real, especialmente en situaciones críticas como los incendios forestales.

Al analizar el sistema descentralizado y su implementación utilizando los CanSats, y con base en el enfoque de microservicios, se evidencia una escalabilidad independiente de la tecnología utilizada. Esto significa que el sistema puede adaptarse y crecer sin inconvenientes, sin importar los cambios o las actualizaciones en los componentes tecnológicos. Además, la transmisión de datos a través de la conexión entre nodos garantiza la integridad de la información, lo cual refuerza la idoneidad de este enfoque para el proyecto en cuestión.

La implementación de un sistema descentralizado basado en microservicios implica una planificación cuidadosa de la estructura y la comunicación entre los nodos. La distribución de las tareas y la asignación de roles específicos a cada nodo permiten una coordinación eficiente y una interacción fluida entre los diferentes componentes del sistema. Esta arquitectura modular también facilita la implementación de medidas de seguridad en cada uno de los servicios, lo cual contribuye a proteger la comunicación y los datos en todo momento.

Es importante destacar que el enfoque descentralizado y basado en microservicios no solo ofrece beneficios técnicos, sino que también se alinea con los objetivos del proyecto en términos de monitoreo y predicción de incendios forestales. La capacidad de respuesta en tiempo real y la capacidad de procesamiento distribuido permiten una mayor eficacia en la recolección y el análisis de datos, lo que a su vez

mejora la toma de decisiones y las acciones preventivas frente a los incendios forestales.

Conclusiones

El presente proyecto ha logrado cumplir con el objetivo de orientar la toma de decisiones oportuna para comenzar la lucha contra los incendios forestales mediante la usabilidad de los picosatélites. Esto ha permitido establecer un sistema descentralizado que cumple con los requisitos fundamentales de disponibilidad, escalabilidad y seguridad, abordando de manera efectiva la problemática de los incendios forestales.

La implementación de un sistema descentralizado ha demostrado ser beneficiosa en varios aspectos. En primer lugar, está la distribución de los nodos y la descentralización de las tareas, que han mejorado el manejo de recursos del sistema al evitar puntos únicos de falla y permitir una mayor redundancia. En segundo lugar, la escalabilidad del sistema se ha logrado de manera independiente de la tecnología utilizada, lo que garantiza que el sistema pueda crecer y adaptarse a medida que se requiera sin limitaciones significativas.

En cuanto a la disponibilidad, la arquitectura descentralizada ha proporcionado un entorno propicio para implementar medidas de comunicación en cada uno de los nodos y servicios. Esto ha contribuido a salvaguardar la integridad de la comunicación y los datos, lo cual resulta esencial en un proyecto que involucra la recopilación y el procesamiento de información sensible relacionada con incendios forestales.

La adopción de un enfoque arquitectónico descentralizado ha sido fundamental para el éxito de este proyecto. La combinación de disponibilidad, escalabilidad y seguridad ha permitido abordar eficazmente los desafíos asociados con los incendios forestales y proporcionar una plataforma robusta para la toma de decisiones basada en datos precisos y actualizados. Este proyecto sienta las bases para futuras investigaciones y aplicaciones en el campo del monitoreo y la

predicción de incendios forestales, brindando una solución innovadora y efectiva para proteger nuestros recursos naturales.

Referencias

- 330ohms. (2016, marzo 2). ¿Qué es una protoboard? [en línea]. <https://tinyurl.com/2sf3td7t>
- APD. (2024, abril 9). *Cómo aplicar la metodología SCRUM* y qué es el método SCRUM [en línea]. <https://tinyurl.com/3d5na2by>
- Beningo, J. (2020, enero 21). *Cómo seleccionar y usar el módulo ESP32 con Wi-Fi/Bluetooth adecuado para una aplicación de IoT industrial*. DigiKey [en línea]. <https://tinyurl.com/mtw5ta2p>
- Buitrago, P., Camacho-Briñez, R., Tejada, J. C., Marmolejo, P. y Jaramillo, Ó. A. (2019). Diseño de un CanSat para medición de variables ambientales. *Publicaciones e Investigación*, 13(5). <https://doi.org/10.22490/25394088.3468>
- Cagua-Colmenares, K. T. y Vinche-Hernández, A. T. (2022). *Construcción de un CanSat para la medición de temperatura, presión atmosférica y altitud* (trabajo de grado, Fundación Universitaria Los Libertadores). Repositorio institucional. <https://tinyurl.com/yc4hbpwn>
- Fernández, Y. (2022, septiembre 23). *Qué es Arduino, cómo funciona y qué puedes hacer con uno* [en línea]. <https://tinyurl.com/mus3zusu>
- Instituto de Hidrología, Meteorología y Estudios Ambientales (Ideam). (s. f.). *Zonificación del riesgo a incendios* [en línea]. <https://tinyurl.com/mt5dxprs>
- Jiménez, J. (2022, enero 15). *En qué consiste una red descentralizada y qué ventajas tiene*. Redes Zone [en línea]. <https://tinyurl.com/2uc6c7nu>
- Leantec.es (2015, abril 25). *Tutorial Arduino: Módulo GPS GPS6MV2* [en línea]. <https://tinyurl.com/mseeahmm>
- Naylamp Mechatronics. (s. f.). *Sensor de presión BMP180* [en línea]. <https://tinyurl.com/tmbewd6f>
- Naylamp Mechatronics. (2016). *Tutorial MPU6050, acelerómetro y giroscopio* [en línea]. <https://tinyurl.com/3wx77rux>
- PalcoElectronica. (s. f.). *Módulos sensores MQ* [en línea]. <https://tinyurl.com/2e4uuex5>
- Quisilema-Carrera, E. W. (2022). *Metodología para pruebas dinámicas de la protección de distancia en dispositivos electrónicos inteligentes, aplicación al sistema nacional de transmisión* (tesis de maestría, Escuela Politécnica Nacional). Repositorio institucional EPN. <https://tinyurl.com/yfmcvm5c>
- Raeburn, A. (2022, agosto 25). *Qué es product backlog y guía para hacer uno con ejemplo*. Asana [en línea]. <https://tinyurl.com/dp528zux>
- Rocha-Garrido, G. (2022, febrero 15). *Burndown Chart (SCRUM): qué es, cómo hacer uno y ejemplos* [en línea]. <https://tinyurl.com/mrx7mwwa>
- Secretaría de Medio Ambiente y Recursos Naturales. (2020, diciembre 20). *Manual de usuario del Sistema de Predicción de Peligro de Incendios Forestales (SPPIF) de México*. <https://tinyurl.com/2n54kebj>
- Shamkhalichenar, H. Bueche, C. J. y Choi, J.-W. (2020). *Printed Circuit Board (PCB) Technology for Electrochemical Sensors and Sensing Platforms*. *Biosensors*, 10(11). <https://doi.org/10.3390/bios10110159>