

"La invención es el motor de la prosperidad. No puedo pensar en nada que acerque más a la gente al cielo que una lluvia de ideas."

Nikola Tesla

Toma de decisiones a través de un dashboard integrado en la nube para el sistema de monitoreo y prevención de incendios forestales.

Autores:

Freire Peñafiel Roberto Patricio.
Ochoa Maldonado Sebastián Alejandro.

Tutor:

Marcillo Parra, Diego Miguel, PhD.

OBJETIVOS

General

Crear el prototipo de un sistema de monitoreo y prevención de incendios forestales en una aplicación web, a través de la integración de datos conseguidos mediante sensores y drones, empleando herramientas de desarrollo de software actuales para garantizar la eficacia y seguridad de la integración, con el objetivo de mejorar la gestión y prevención de este tipo de desastres.

E S P E C Í F I C O S

1

Buscar información relevante acerca de metodologías y herramientas de desarrollo de software para el levantamiento del dashboard.

2

Fabricar un prototipo del sistema que posibilite incorporar en tiempo real la lectura de nodos de sensores y visión de una emulación de cámara de un dron.

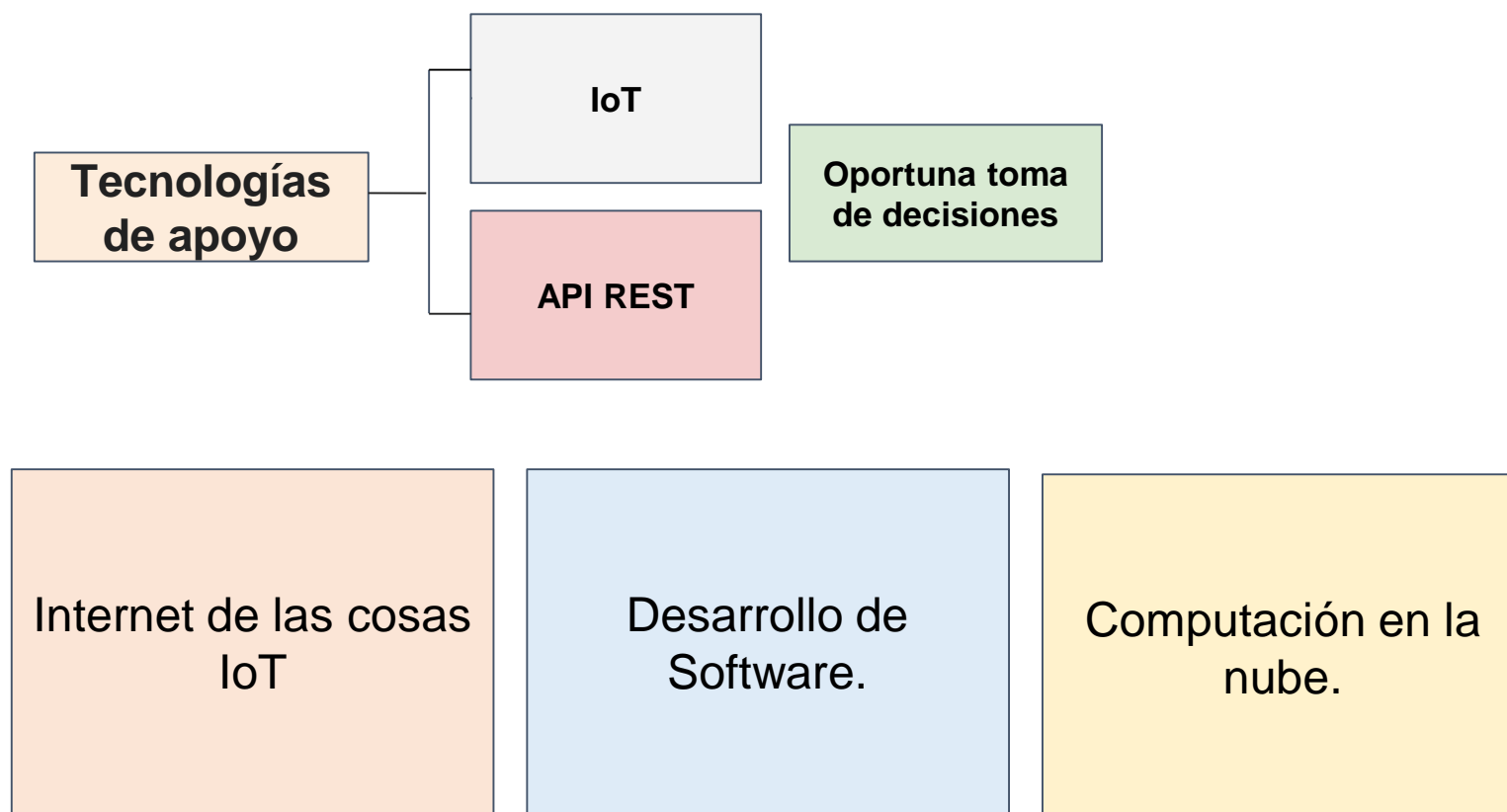
3

Realizar el despliegue e integración del prototipo en la nube con el objetivo de lograr visualizar la información obtenida de manera global, utilizando diversas plataformas de despliegue para todas las capas del sistema.

4

Valorar la correcta funcionalidad del sistema, la obtención y el registro histórico de la información.

Alcance



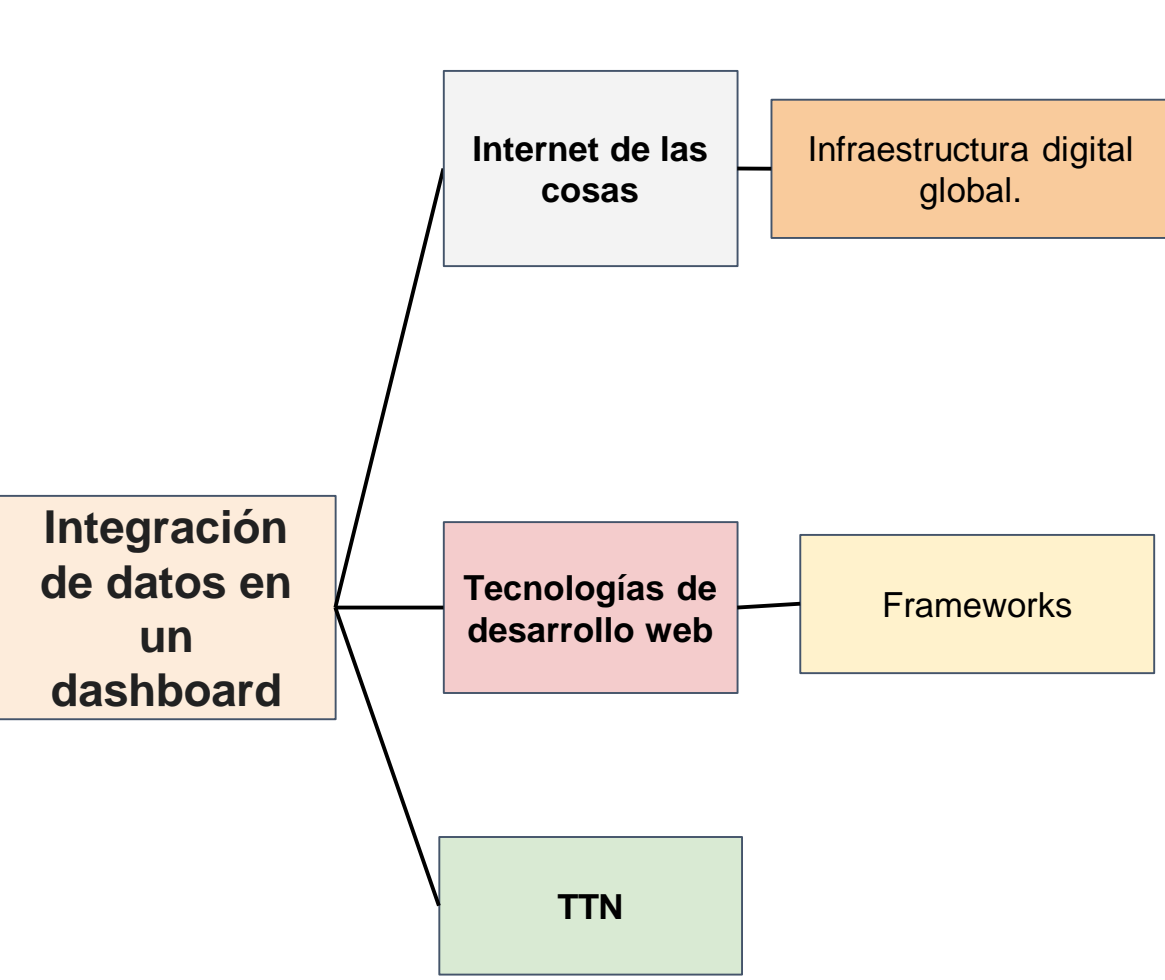
Prototipo de sistema de información para alertas tempranas de incendios forestales .

La aplicación web creada dará paso a la integración de los datos obtenidos a través de drones y sensores, los cuales a su vez serán utilizados para el desarrollo de modelos predictivos para la detección oportuna de incendios forestales.

Brindar una comunicación segura y fluida, asegurando totalmente la disponibilidad de la información actualizada para su examinación y posterior toma de decisiones oportunas.



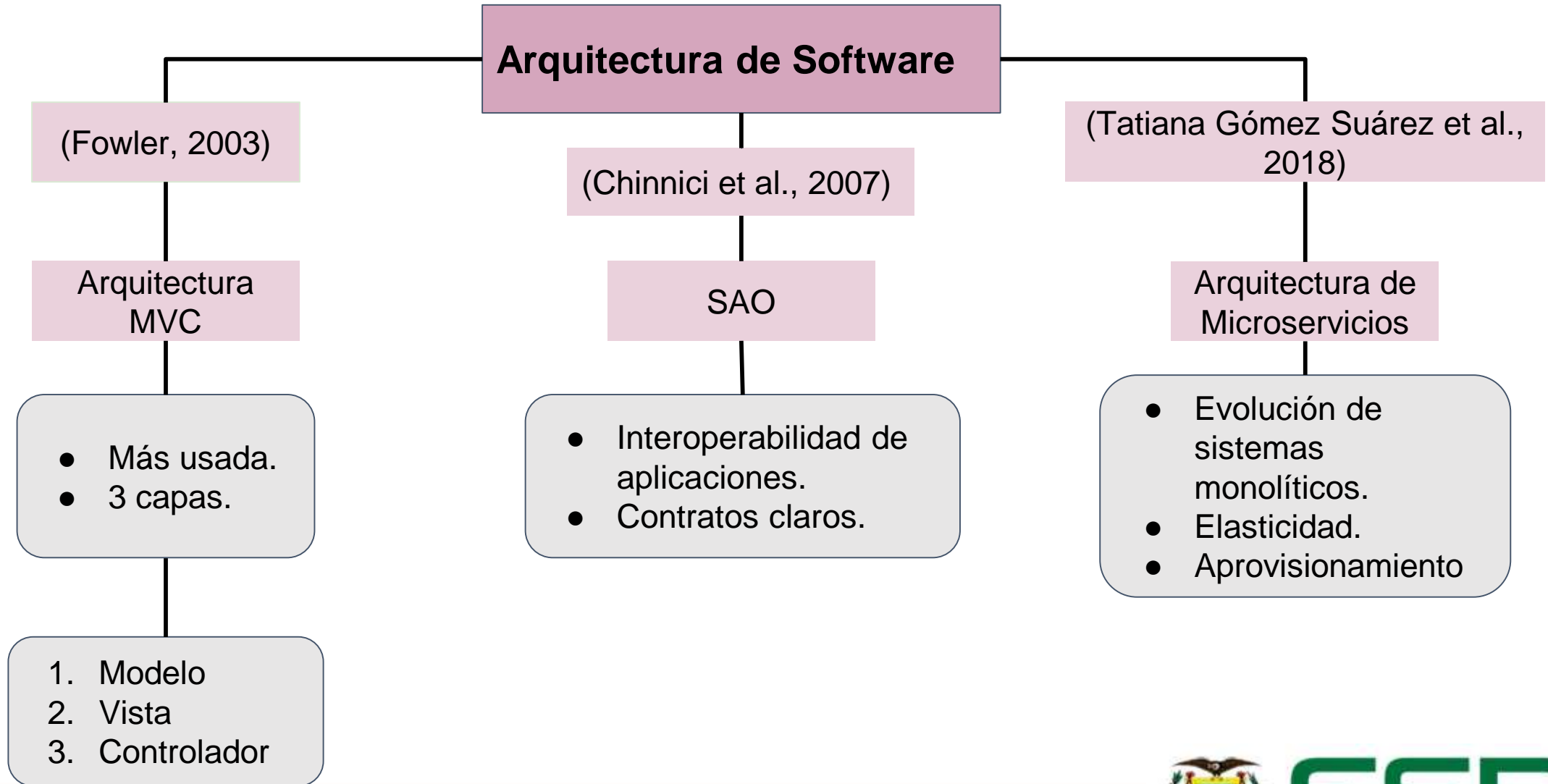
MARCO TEÓRICO



(De, 2017), Las APIs permiten la incorporación y el intercambio de datos de forma eficiente y segura, por lo que se dice que han transformado la manera en que las aplicaciones y los servicios se conectan y colaboran entre sí

Bases de datos posibilitan el acceso y el manejo competente de amplios grupos de datos ya que poseen protocolos de almacenamiento estructurado de información

MARCO TEÓRICO



Fortalezas de las arquitecturas MVC y SOA

Arquitectura MVC	Arquitectura SOA
Separación clara de responsabilidades entre el modelo, la vista y el controlador, lo que facilita el mantenimiento y la escalabilidad del código.	Promueve la reutilización de servicios a través de interfaces estandarizadas, lo que permite construir aplicaciones flexibles y adaptativas.
Proporciona una estructura organizada y modular para el desarrollo de aplicaciones, lo que facilita la colaboración entre los equipos de desarrollo.	Permite la integración de sistemas heterogéneos y la interoperabilidad entre aplicaciones de diferentes plataformas y tecnologías.
Facilita la implementación de pruebas unitarias y de integración, lo que mejora la calidad del software y agiliza el proceso de desarrollo.	Permite la evolución independiente de los servicios, lo que facilita la actualización y el mantenimiento sin afectar a otras partes del sistema.
Es especialmente adecuada para aplicaciones web y de escritorio, donde se requiere una interfaz de usuario interactiva y una gestión eficiente de la lógica de negocio.	Ofrece una arquitectura flexible y adaptable a medida que los requisitos empresariales cambian y evolucionan.
	Permite la escalabilidad horizontal, agregando o eliminando servicios según las necesidades de rendimiento y demanda del sistema.

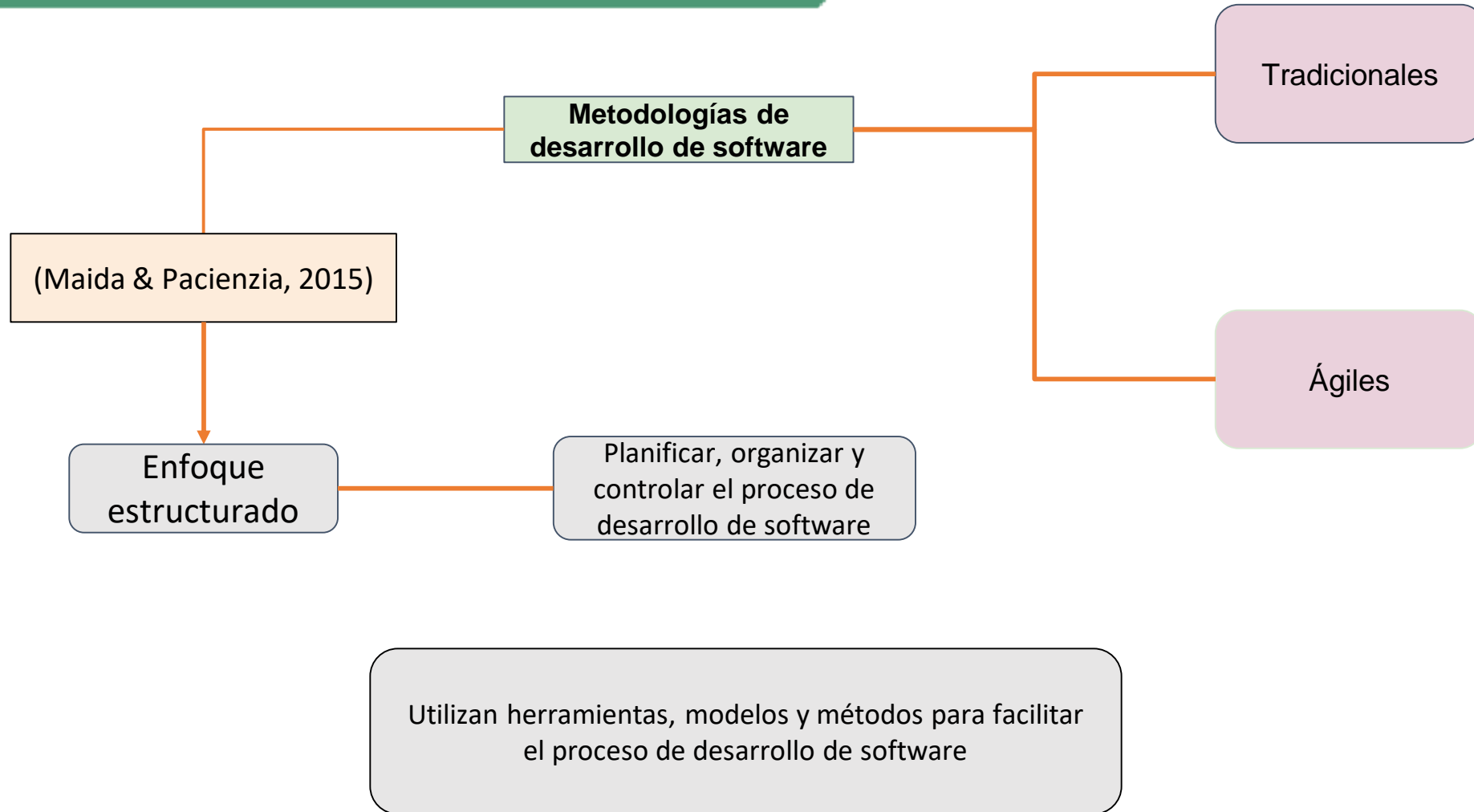


Comparativa entre la arquitectura monolítica MVC y la arquitectura de microservicios

Categoría	Arquitectura Monolítica	Arquitectura de microservicios
Código	Código fuente único para toda la aplicación.	Código base independiente para cada microservicio.
Comprensibilidad	A menudo confuso y difícil de mantener.	Mayor facilidad de lectura y mantenimiento.
Despliegue	Implementaciones complejas con ventanas de mantenimiento y paradas programadas.	Despliegue sencillo, cada microservicio se puede implementar de forma individual, con un tiempo de inactividad mínimo.
Tecnologías	Típicamente desarrollado utilizando homogeneidad de pilas tecnológicas.	Cada microservicio es desarrollado utilizando heterogeneidad de tecnologías.
Escalamiento	Requiere escalar la aplicación entera	Cada microservicio puede ser escalado de manera independiente



MARCO TEÓRICO

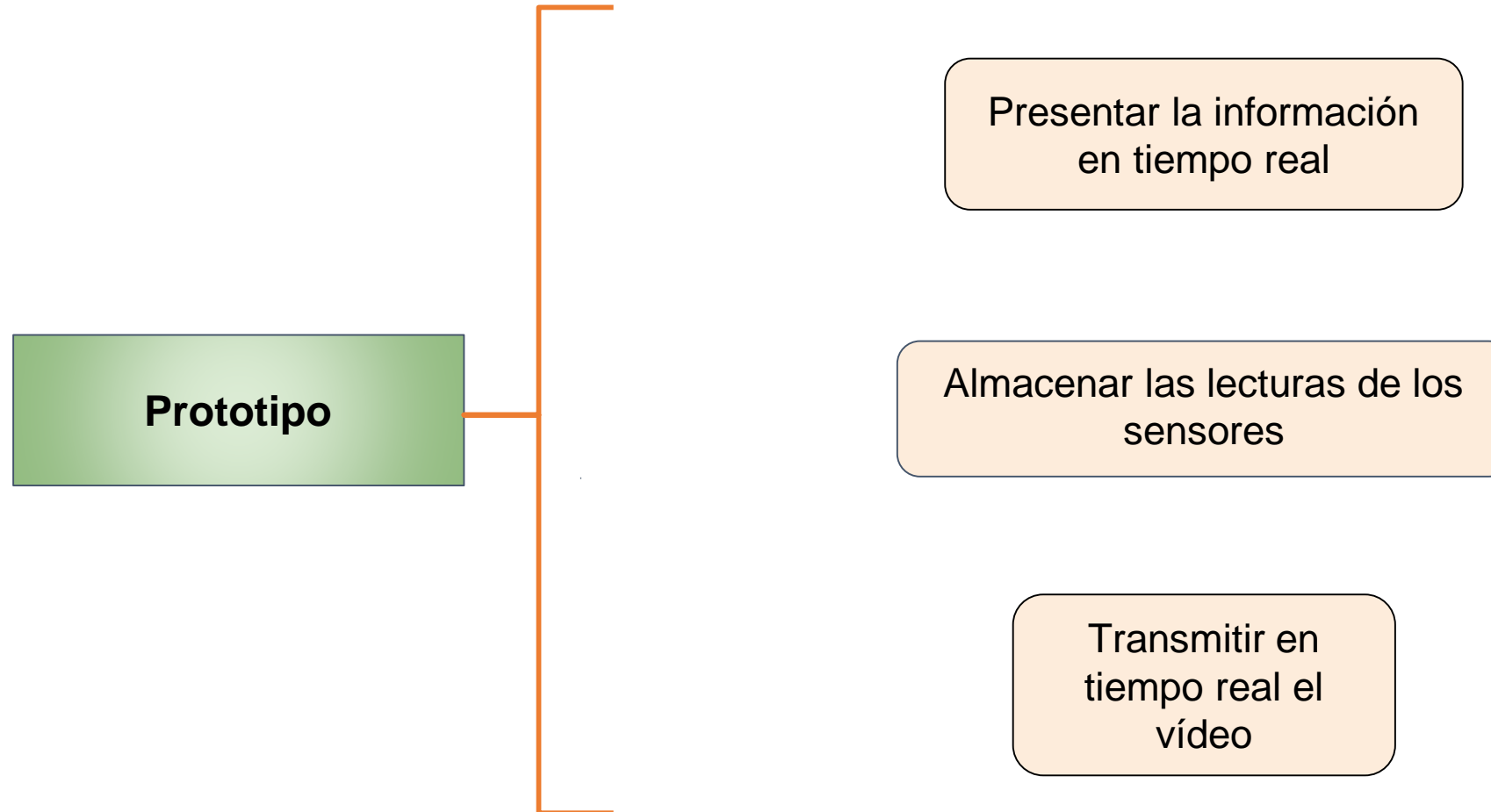


METODOLOGÍAS TRADICIONALES VS ÁGILES

Metodologías tradicionales	Metodologías ágiles
Predictivas	Adaptativas
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdivido en proyectos pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas de software constantes
Documentación extensa	Poca documentación



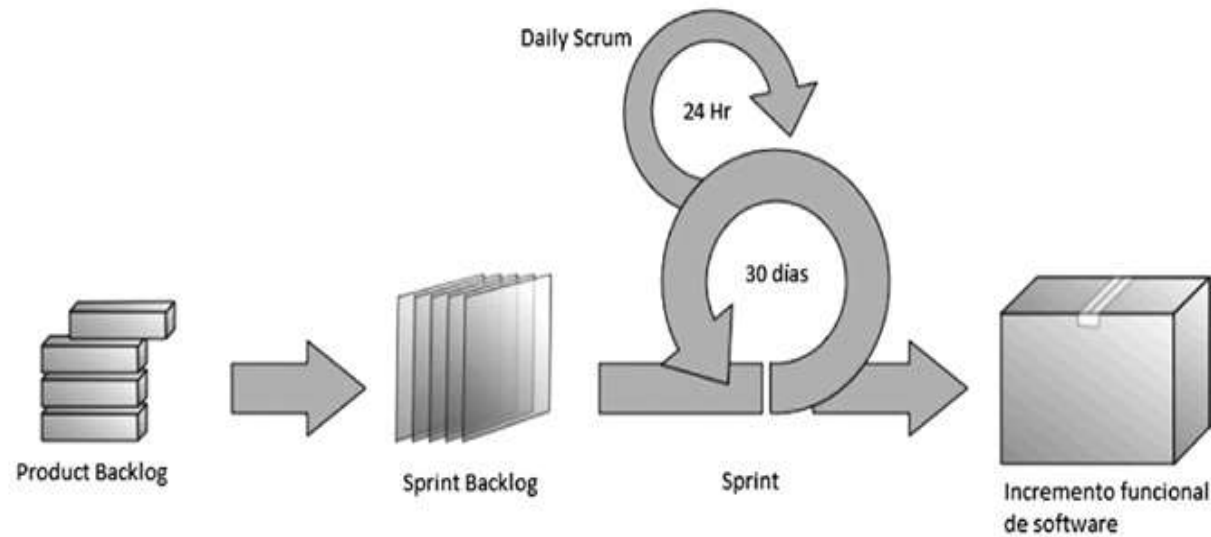
Desarrollo – Requerimientos funcionales



Arquitectura de Microservicios

- Escalabilidad
- Aprovisionamiento rápido
 - Flexibilidad
 - Elasticidad

SCRUM



Desarrollo – Selección de tecnologías



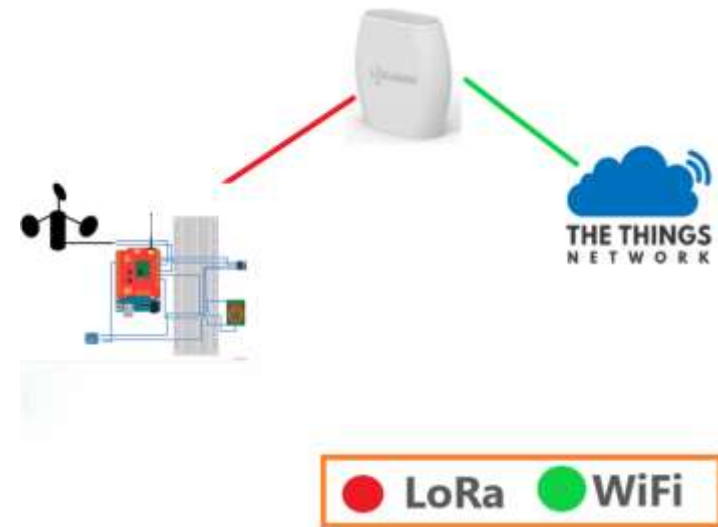
N°	Framework
1	Laravel
2	Django
3	VueJS
4	React
5	Ruby on Rails
6	Node.js
7	Angular

(Armash Aslam et al., 2015)



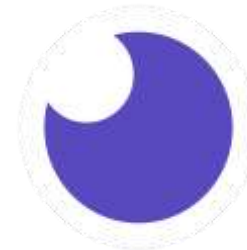
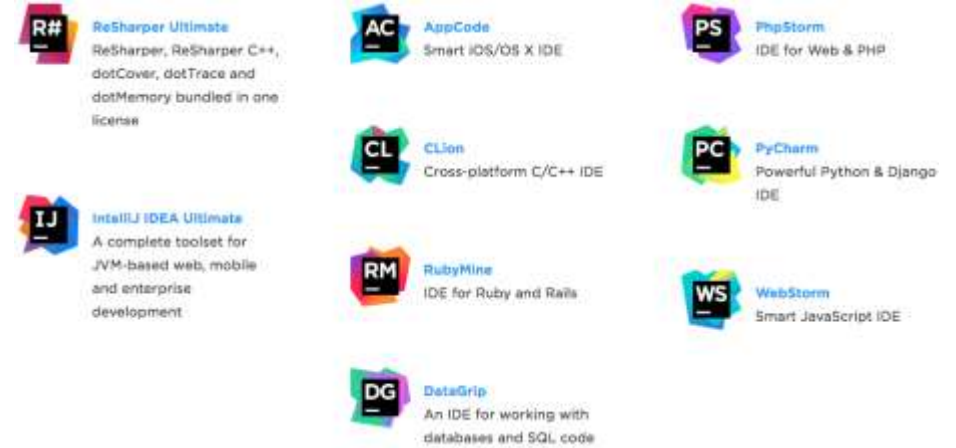
Desarrollo – Herramientas de hardware

- Nodo o módulo de recolección de información
- Indoor Gateway TTN.
- Arduino UNO
- ESP32-CAM

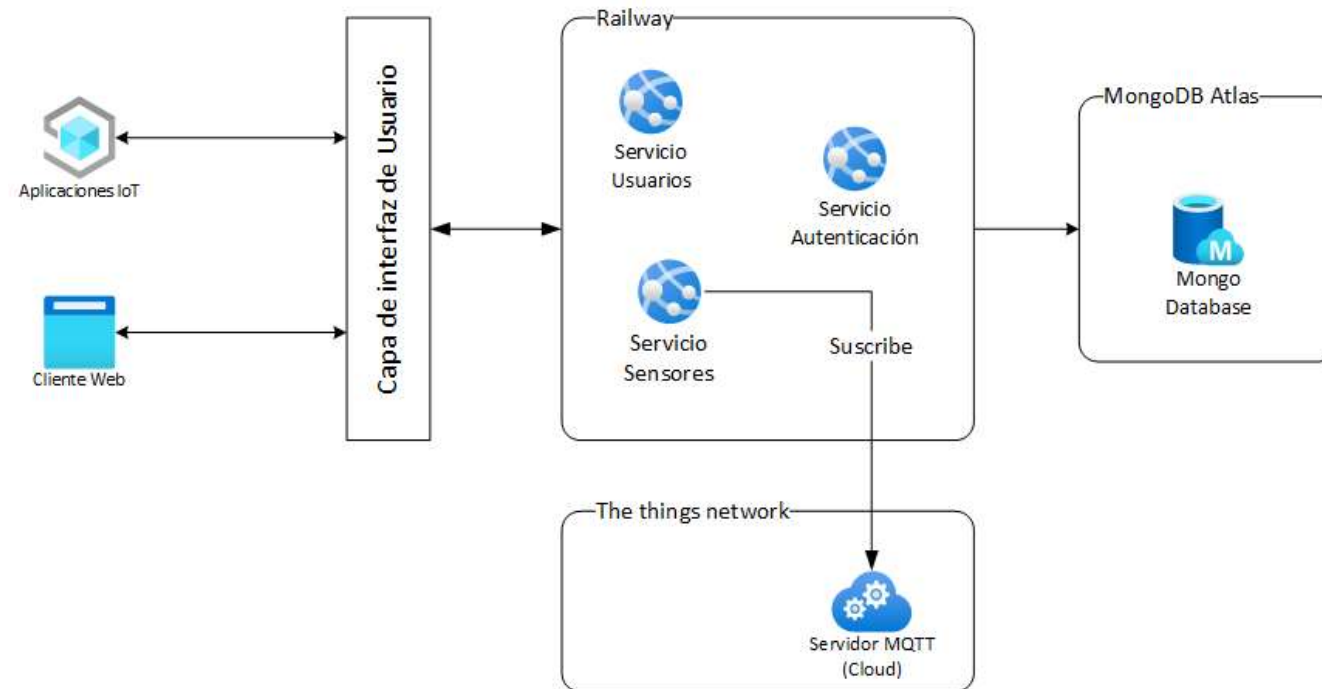


Desarrollo – Herramientas de software

- Software de programación
- Software de control de proceso de desarrollo
- Software de administración de base de datos
- Software de aplicación



Arquitectura propuesta



Punto de vista de capas

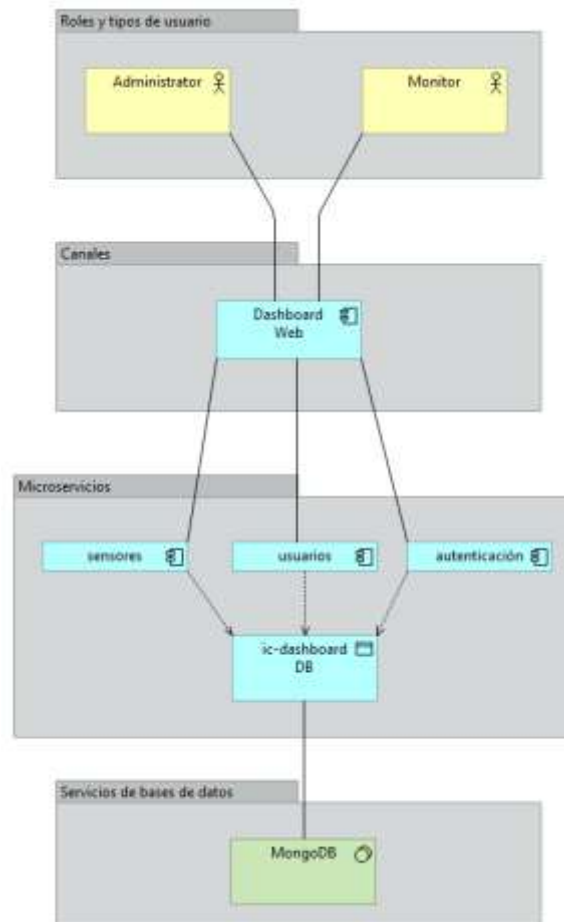


Diagrama de contexto

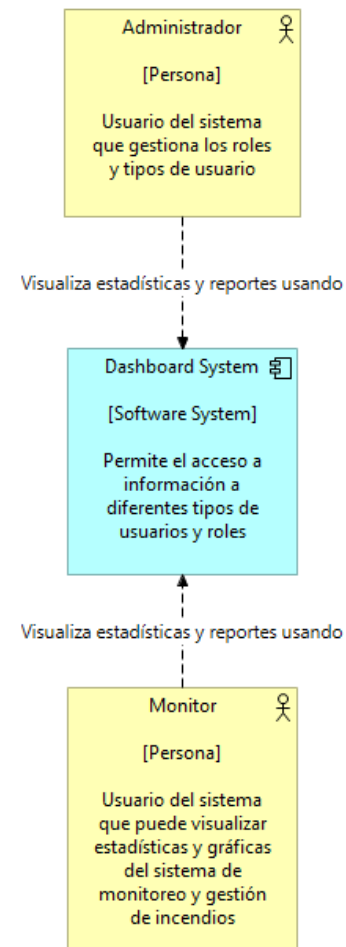
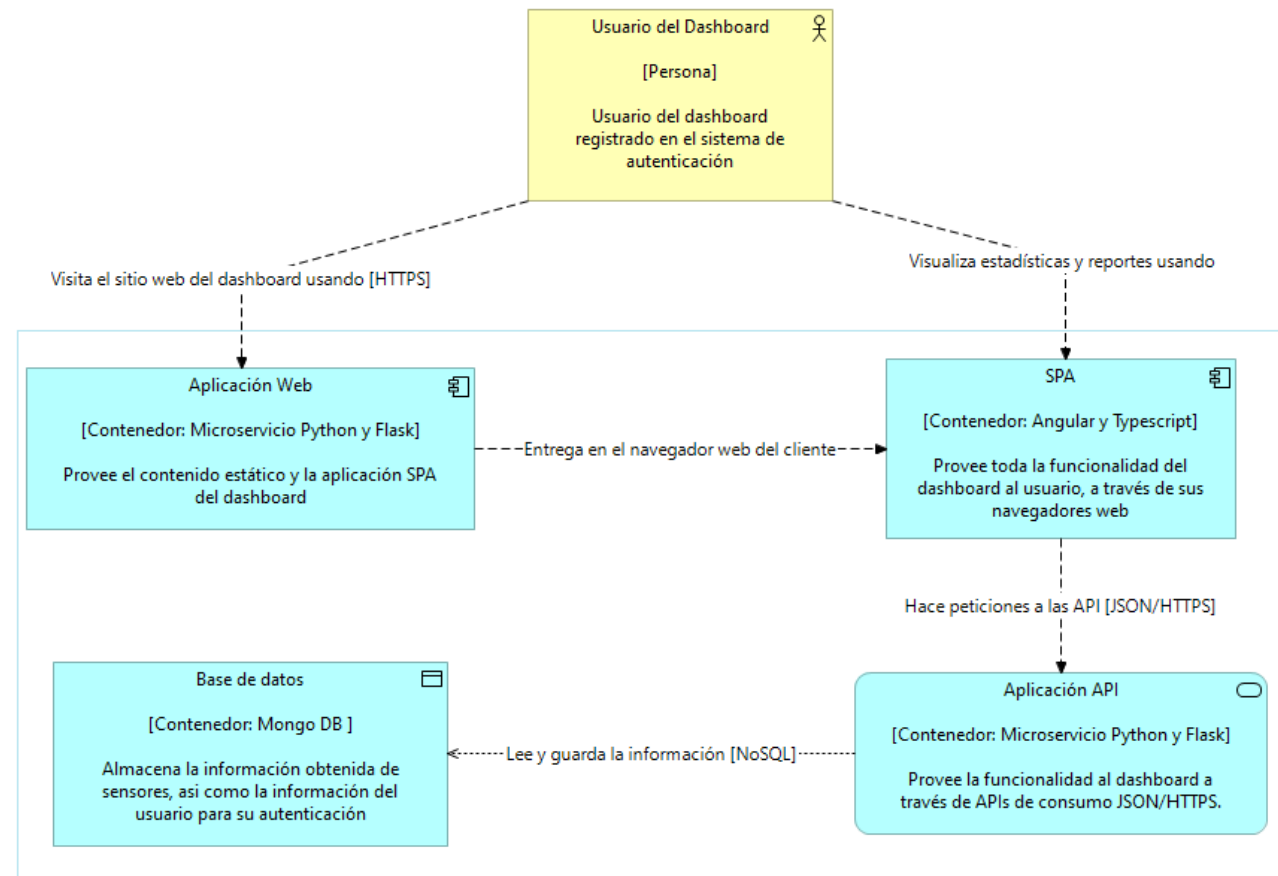


Diagrama de contenedores



Flask - microframework

Diseño minimalista
y modular

Flexibilidad de
incorporar
características

Capacidad de
mantener código
limpio y legible



Microservicios

dashboard-sensors

● ic-ws-sensors-production.up.railway.app

dashboard-users

● ic-ws-users.up.railway.app

dashboard-auth

● ic-ws-auth-production.up.railway.app



Angular - Framework

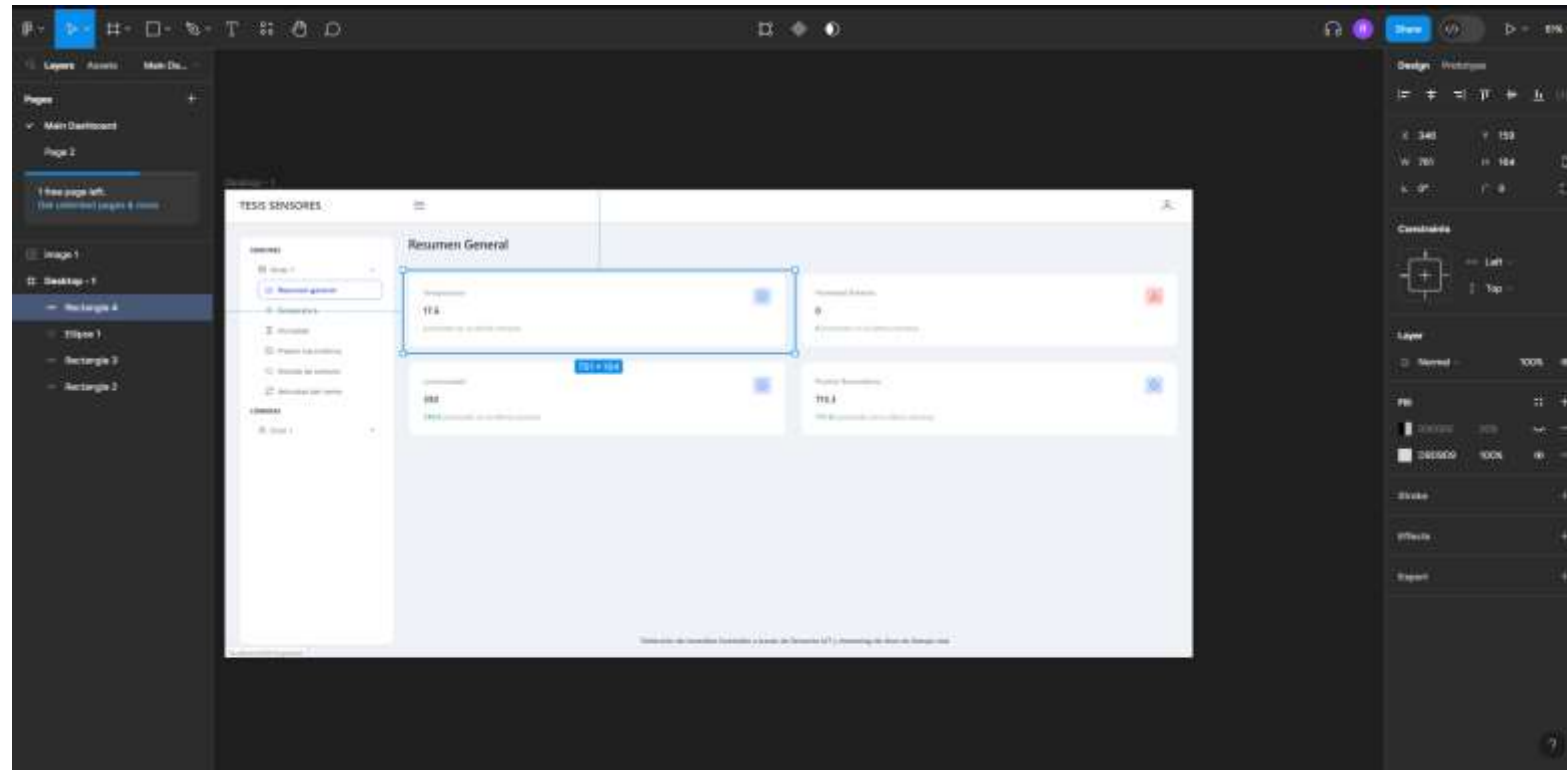
Componentes
reutilizables

Gestión eficiente de
dependencias

Manipulación
flexible del DOM



Diseño de la estructura del dashboard y sus componentes



Estructura de visualización

SENSORES

 Nodo 1 

 Resumen general

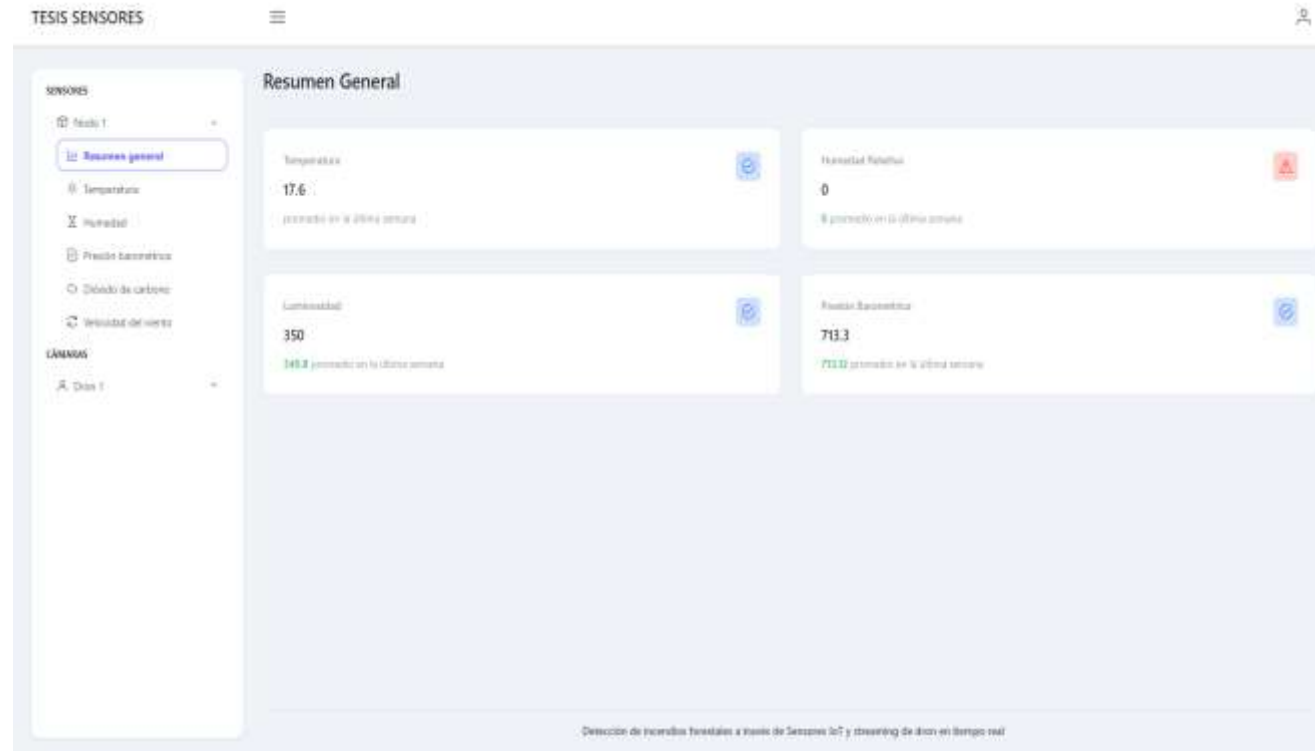
 Temperatura

 **Humedad**

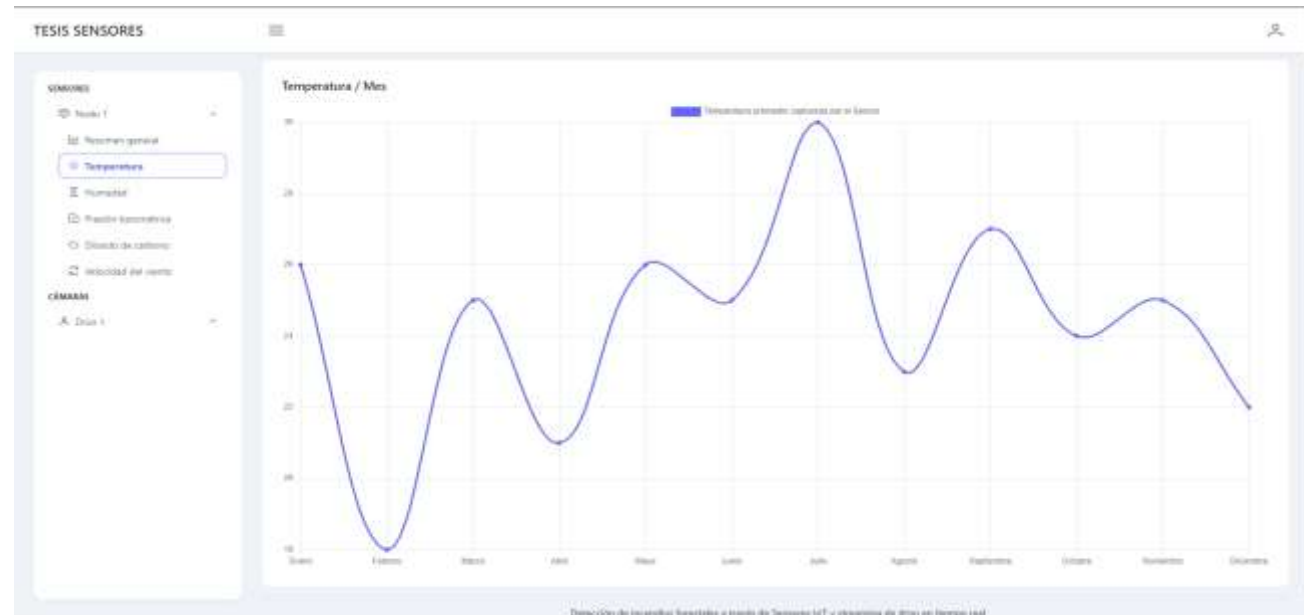
Presión barométrica



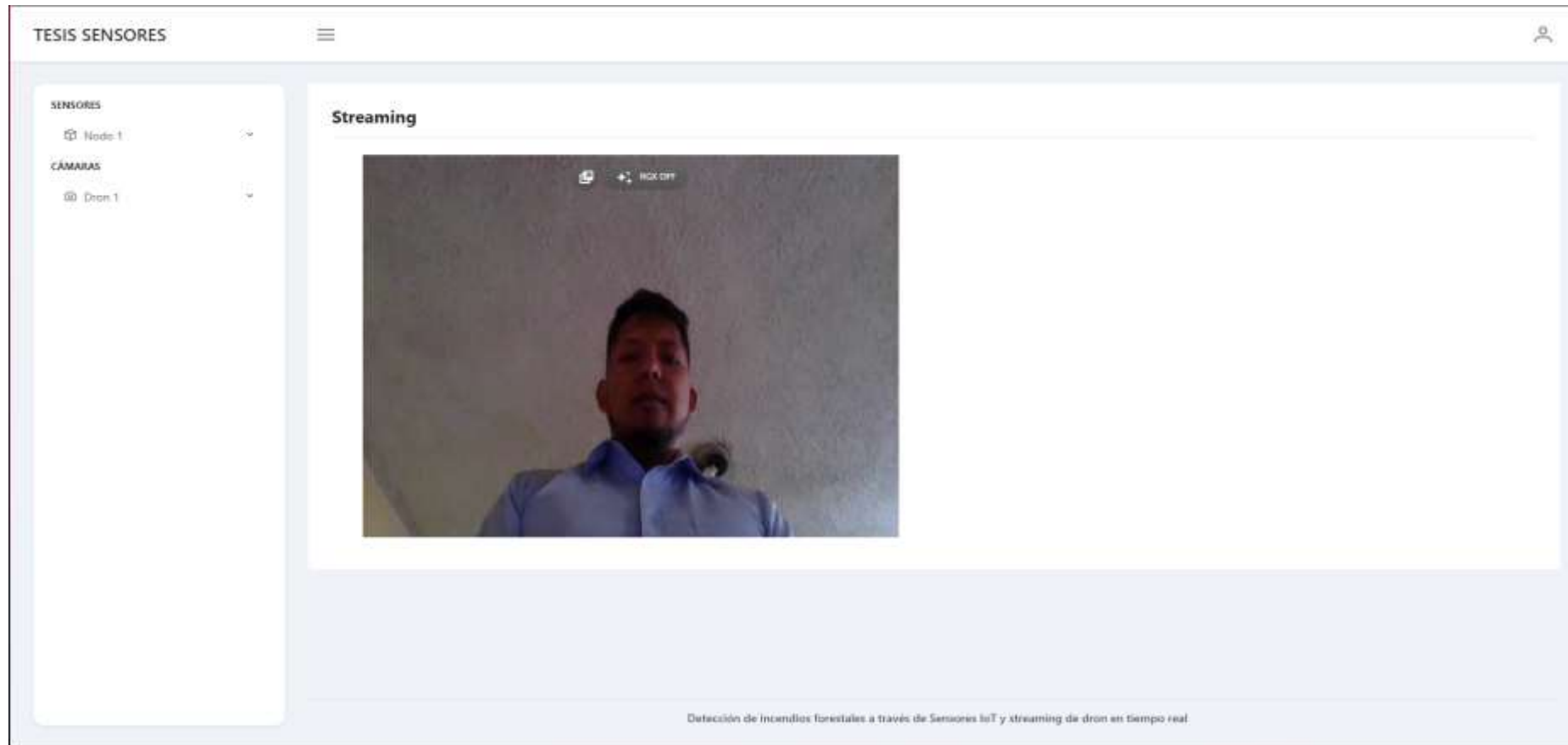
Módulo General



Módulo Sensores



Módulo Drone



Resultados

Conectividad del sistema de lectura de sensores y almacenamiento en la nube

```
Temperatura: 22.38 *C , Presion: 756.49 mb
Packet queued
LMIC.freq:868100000
Receive data:
142020: 10
EV_TXCOMPLETE (includes waiting for RX windows)
##### NO.2 #####
The temperautre and humidity :
[nan°C,nan%]
Temperatura=nanHumedad=nanAirQua=339 PPM
Temperatura: 22.35 *C , Presion: 756.46 mb
Packet queued
LMIC.freq:868300000
Receive data:
907955: 10
EV_TXCOMPLETE (includes waiting for RX windows)
```



Resultados

Streaming en tiempo real

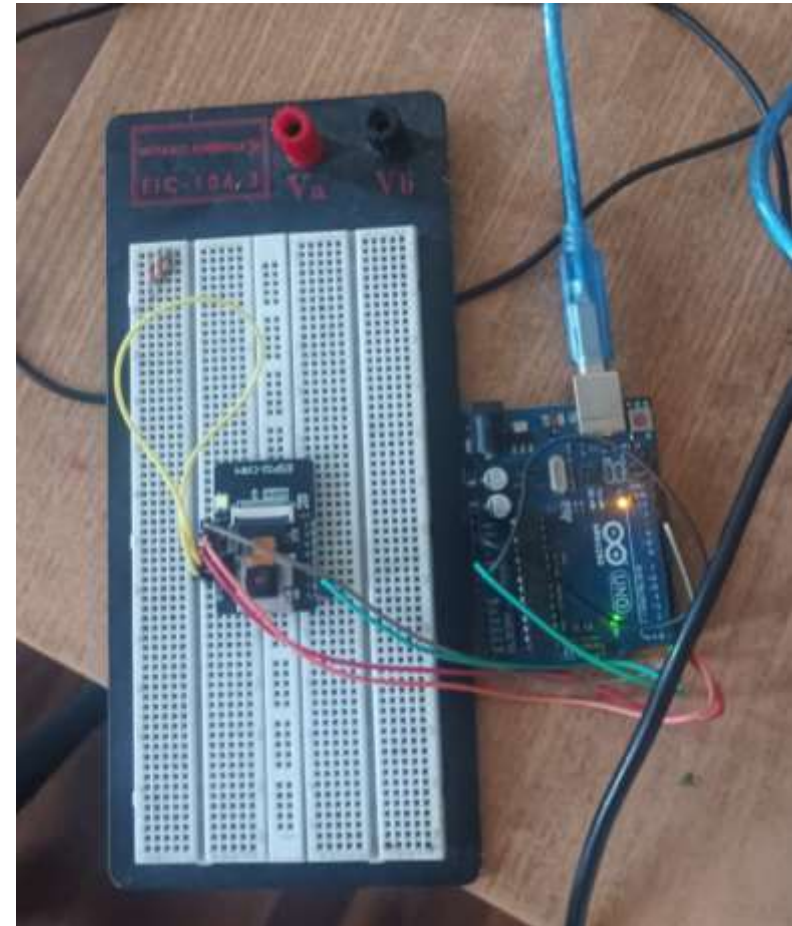
Para el evaluar el streaming en tiempo real se procedió ejecutar la simulación del dron con el Arduino y el ESP32-CAM y se accedió a la dirección IP provista mediante el componente insertado en el front-end



Resultados

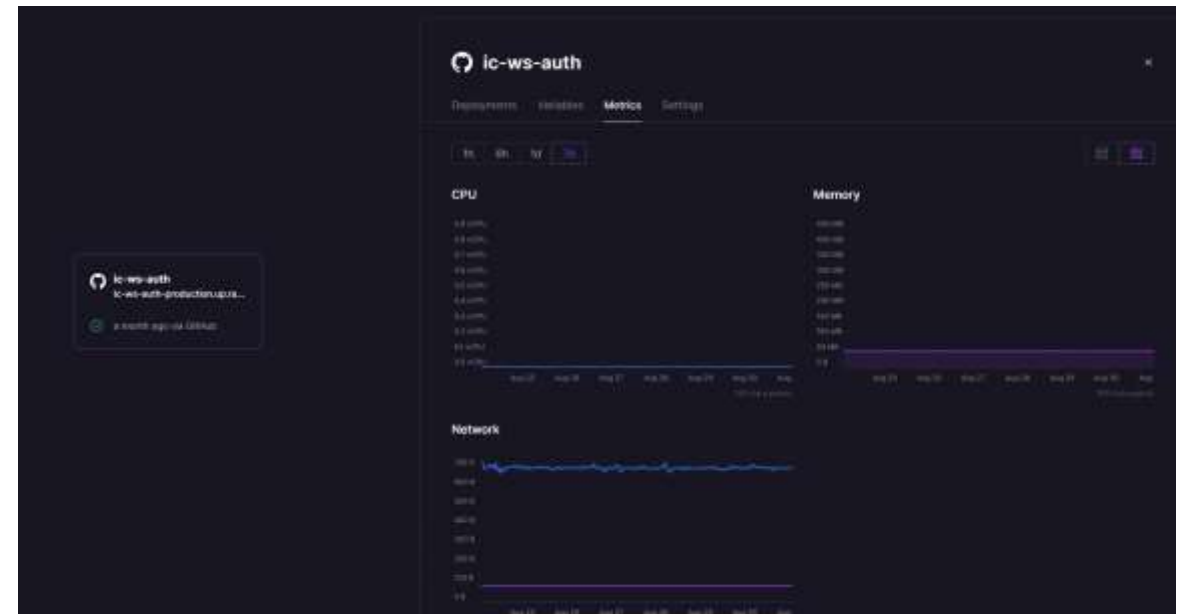
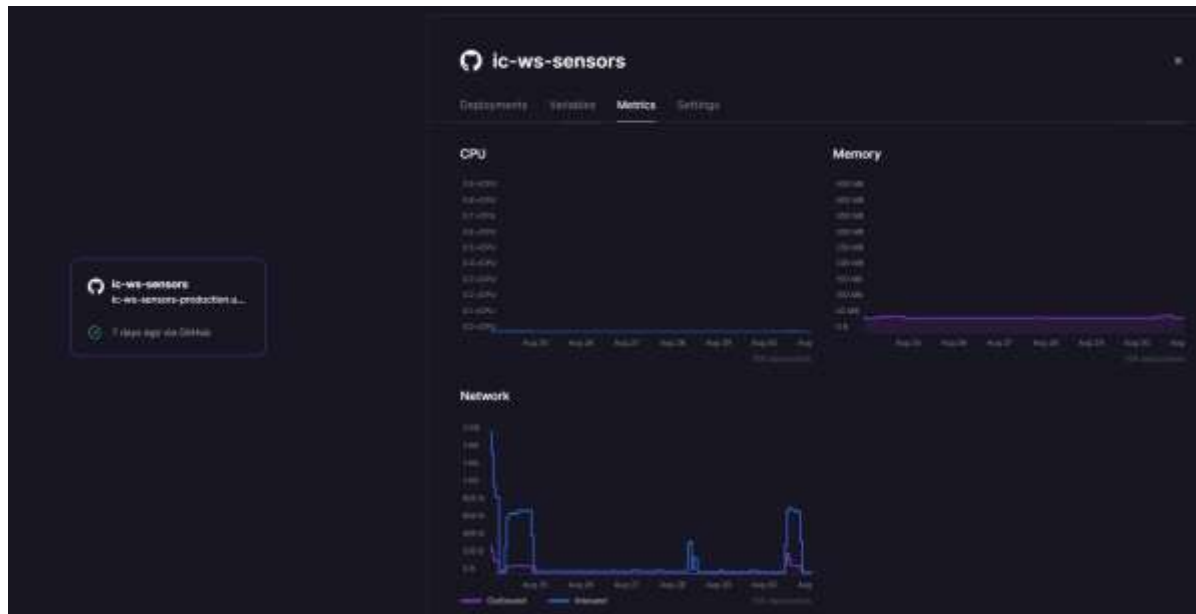
Streaming en tiempo real

```
CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
94  #endif
95  }
96
97  #if defined(CAMERA_MODEL_ESP_EYE)
98  pinMode(13, INPUT_PULLUP);
99  pinMode(14, INPUT_PULLUP);
100 #endif
101
102 // camera init
103 esp_err_t err = esp_camera_init(&config);
104 if (err != ESP_OK) {
105     Serial.printf("Camera init failed with error 0x%x", err);
106     return;
107 }
108
109 sensor_t * s = esp_camera_sensor_get();
110 // initial sensors are flipped vertically and colors are a bit saturated
111 if (s->id.PID == OV3660_PID) {
112     s->set_vflip(s, 1); // flip it back
113     s->set_brightness(s, 1); // up the brightness just a bit
114     s->set_saturation(s, -2); // lower the saturation
115 }
116 // drop down frame size for higher initial frame rate
117 if (config.pixel_format == PIXFORMAT_JPEG){
118     s->set_framesize(s, FRAMESIZE_QVGA);
119 }
120
121 #if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
122     s->set_vflip(s, 1);
123     s->set_hmirror(s, 1);
124 #endif
125
126 #if defined(CAMERA_MODEL_ESP32S3_EYE)
127     s->set_vflip(s, 1);
128 #endif
```



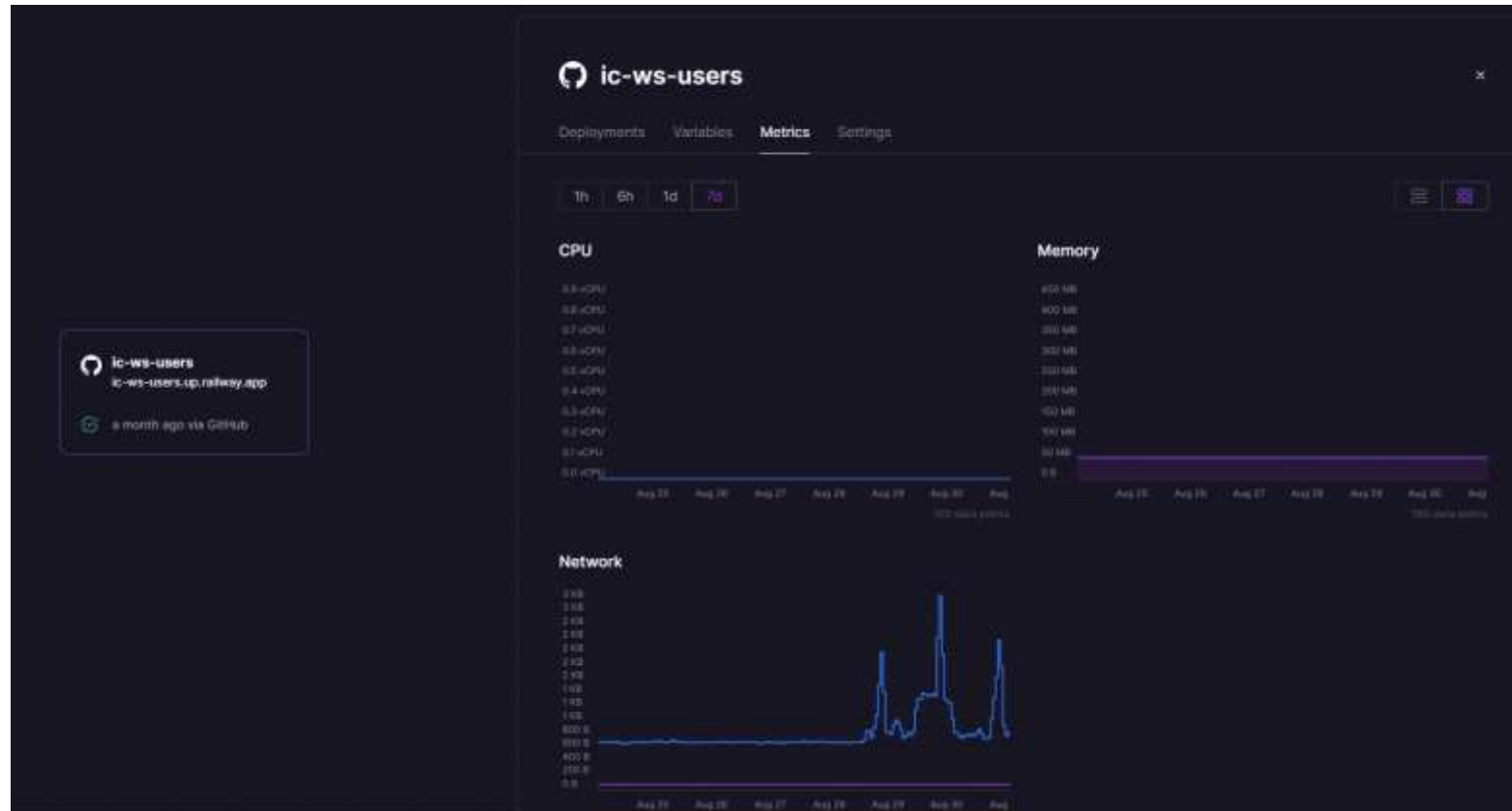
Resultados

Despliegue en la nube del prototipo



Resultados

Despliegue en la nube del prototipo



Conclusiones

La investigación sobre metodologías y herramientas de desarrollo de software permitió identificar las mejores prácticas y enfoques más actuales para la construcción del dashboard

La integración exitosa de la lectura de nodos de sensores y la transmisión en tiempo real de la emulación de cámara de un dron dentro de una aplicación web demostró la viabilidad técnica de la propuesta.

La evaluación de la funcionalidad del sistema y la obtención de información, así como el registro histórico, arrojó resultados alentadores. Se comprobó que el prototipo era capaz de capturar y transmitir datos de manera eficiente, brindando información valiosa para la toma de decisiones

La etapa de despliegue e implementación en la nube marcó un paso fundamental hacia la globalización de la información. La capacidad de acceder a la información del sistema desde diferentes plataformas y ubicaciones se convirtió en un activo esencial para la gestión y prevención de incendios forestales.



Trabajos Futuros

Permitir la visualización y gestión de información de toda una red de nodos en lugar de limitarse a un solo nodo. Esta ampliación permitiría una vista más completa y holística del sistema de monitoreo y prevención de incendios forestales, lo que facilitaría la identificación de patrones y tendencias en el comportamiento de la red. Para lograr esto, será necesario implementar técnicas de agregación y visualización de datos que permitan la representación eficiente de grandes volúmenes de información provenientes de múltiples nodos.

Considerar la implementación de técnicas de análisis de big data para procesar y analizar grandes volúmenes de datos generados por el sistema de monitoreo. El uso de tecnologías y herramientas de big data, como Apache Hadoop y Apache Spark, permitiría identificar patrones y tendencias significativas en los datos recopilados, lo que proporcionaría información valiosa para mejorar la eficiencia y la efectividad de las operaciones de prevención de incendios forestales.

La dockerización de los componentes del sistema brindaría ventajas en términos de portabilidad y gestión de contenedores. Al encapsular cada componente en contenedores Docker, se garantiza una mayor independencia y aislamiento, lo que facilitaría el despliegue en diferentes entornos y la gestión de versiones.

Para lograr una mayor escalabilidad, disponibilidad y rendimiento del sistema, se sugiere considerar el despliegue en nubes más profesionales, como Azure u otras plataformas de nube reconocidas. Estas nubes ofrecen servicios avanzados que facilitarían la gestión de recursos, el escalado automático y la alta disponibilidad del sistema. Además, la implementación de prácticas de DevOps, como integración continua (CI/CD), permitiría una entrega de software más ágil y eficiente.



REFERENCIAS

- API Documentation Tool | Postman. (s/f). <https://www.postman.com/api-documentation-tool/>
- Armash Aslam, F., Nabeel Mohammed Jummal Musab Mohd Munir Murade Aaraf Gulamgaus, H., & Lokhande Assistant Professor, P. S. (2015). Efficient Way Of Web Development Using Python And Flask. En International Journal of Advanced Research in Computer Science (Vol. 6, Número 2). www.ijarcs.info
- Bogotá, M. F. C., Lizcano, V. N., & Quintero, R. A. S. (2023). DISEÑO DE UN PICOSATÉLITE PARA EL MONITOREO Y PREDICCIÓN DEL COMPORTAMIENTO DE INCENDIOS FORESTALES. Revista Ingeniería, Matemáticas y Ciencias de la Información. <https://api.semanticscholar.org/CorpusID:258230257>
- Cadavid, A. N., Martínez, J. D. F., & Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. Prospectiva, 11(2), 30–39. redalyc.org/pdf/4962/496250736004.pdf
- Chinnici, R., Moreau, J.-J., Ryman, A., & Weerawarana, S. (2007). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation. <https://www.w3.org/TR/2007/REC-wsdl20-20070626/>
- De, B. (2017). API Management. API Management, 15–28. https://doi.org/10.1007/978-1-4842-1305-6_2
- Beynon P. (n.d.). Sistemas de bases de datos. Retrieved August 14, 2023, from https://books.google.com.ec/books?hl=es&lr=&id=XjbeDwAAQBAJ&oi=fnd&pg=PR5&dq=Bases+de+datos&ots=DIEYxIJVLO&sig=u8sW20F7hNWW_bC5f3FPjmOa2nk&redir_esc=y#v=onepage&q=Bases%20de%20datos&f=false
- del Busto, Ing. H. G., & Enríquez, Ing. O. Y. (2012). BASES DE DATOS NoSQL. Telemática, 11(3), 21–33. <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/74>
- Ozierańska, A., Skomra, A., Kuchta, D., & Rola, P. (2016). The critical factors of Scrum implementation in IT project– the case study. Journal of Economics and Management, 25, 79–96. <https://doi.org/10.22367/JEM.2016.25.06>
- Espinosa-Hurtado, R. (2021). Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web. CEDAMAZ, 11(2), 133–141. <https://doi.org/10.54753/cedamaz.v11i2.1182>
- Fowler, M. (2003). Patterns_of_Enterprise_Application_Archi. Addison-Wesley, Illustrated, 553.
- Grady, B. (2007). Object-oriented Analysis and Design with Applications. Addison-Wesley, 3, 691.

REFERENCIAS

- Google. (2023). Angular - Introduction to the Angular docs. Introduction to the Angular Docs. <https://angular.io/docs>
- Guerrero, H., & Carolina, K. (2019). Sistema monitoreo y alerta temprana de incendios forestales, estructurales y cobertura vegetal en la ciudad de Santa Marta. <https://api.semanticscholar.org/CorpusID:192048191>
- Ibarra, S. G. P., Quispe, J. R., Mullicundo, F. F., & Lamas, D. A. (2021). Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd. XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja), August 2021, 963–968. <http://sedici.unlp.edu.ar/handle/10915/120476>
- Maida, E. G., & Pacienza, J. (2015). Metodologías de desarrollo de software.
- MongoDB Documentation. (s/f). <https://www.mongodb.com/docs/>
- Myatt, G. J., & Johnson, W. P. (2009). Making sense of data II: A practical guide to data visualization, advanced data mining methods, and applications (Vol. 2). John Wiley & Sons.
- Orellana, C., Patricio, R., Tigse, R., Patricio, F., Parra, M., & Miguel, D. (2021). Prototipo de sistema de información para alertas tempranas de incendios forestales.
- Pantoja, L., & Pardo, C. (2016). Evaluando la Facilidad de Aprendizaje de Frameworks mvc en el Desarrollo de Aplicaciones Web. Publicaciones e Investigación, 10, 129–142. <https://doi.org/10.22490/25394088.1592>
- Pressman, R. S. (2010). Ingeniería del software : un enfoque práctico. McGraw-Hill.
- Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud computing: An overview. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5931 LNCS, 626–631. https://doi.org/10.1007/978-3-642-10665-1_63/COVER
- Railway Docs. (s/f). <https://docs.railway.app/>
- Suwaid, M. M., Habaebi, M. H., & Khan, S. (2019). Embedded LoRaWAN for Agricultural Sensing Applications. ICETAS 2019 - 2019 6th IEEE International Conference on Engineering, Technologies and Applied Sciences. <https://doi.org/10.1109/ICETAS48360.2019.9117346>

REFERENCIAS

- Talwana, J. C., & Hua, H. J. (2017). Smart World of Internet of Things (IoT) and Its Security Concerns. Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCOM-Smart Data 2016, 240–245. <https://doi.org/10.1109/ITHINGS-GREENCOM-CPSCOM-SMARTDATA.2016.64>
- Tatiana Gómez Suárez, K., Anaya, R., & Cano, A. F. (2018). Un acercamiento a los microservicios. <https://martinfowler.com/mi>
- VISUALIZACIÓN DE DATOS - Campus2B Bilbao. (s/f). Recuperado el 15 de agosto de 2023, de <https://www.campus2b.com/evento/visualizacion-de-datos/>
- Patricia, J., Gamboa, Z., Alexandra, C., & Arreaga, L. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. Evolution of the Methodologies and Models used in Software Development. INNOVA Research Journal, 3(10), 20–33.