

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de tablero de monitoreo de calidad de servicio de redes
inalámbricas wifi

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Ciencias de la Computación

Presentado por:

Josue Alexander Cobos Salvador

Andrés Xavier Vargas Vera

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto lo dedico a mis padres: Jose Javier Cobos y Katty Salvador, que han sido parte de toda mi trayectoria educativa hasta la fecha de hoy. Pilares fundamentales para la realización de cada uno de mis logros durante mi carrera.

JOSUE COBOS

AGRADECIMIENTOS

Quiero brindar mi profundo agradecimiento a Msc. Allan Avendaño por el tiempo, apoyo y guía que nos supondar durante la realización del presente proyecto.

ANDRÉS VARGAS

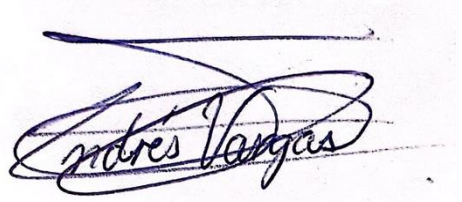
AGRADECIMIENTOS

Quiero agradecer a todas las personas que me han aportado de manera directa o indirecta en mi desarrollo como profesional, esto incluye a mis hermanos Javier Cobos y Ariel Cobos, familiares, amigos, profesores, mentores, colegas y jefes, con una mención especial para aquellas personas que ya no nos acompañan. Han sido actores cuya participación en mi vida ha sido indispensable para poder estar en estas instancias de mi carrera.

JOSUE COBOS

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Josue Cobos y Andrés Vargas damos nuestro consentimiento para que la ESPOL (Escuela Superior Politécnica del Litoral) realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

A handwritten signature in blue ink, reading "Andrés Vargas", is written over a horizontal line. The signature is stylized and cursive.

Josue Cobos Salvador

Andrés Vargas Vera

EVALUADORES

Nombre del Profesor

Msig. Ronald Criollo

PROFESOR DE LA MATERIA

Nombre del Profesor

M.Sc. Allan Avendaño

PROFESOR TUTOR

RESUMEN

La Gerencia de Tecnologías y Sistemas Informáticos (GTSI) de ESPOL administra un conjunto de redes con acceso a Internet; sin embargo, carece de un mecanismo de reporte en tiempo real de las métricas de calidad de servicio de los puntos de acceso de la red. Este proyecto tiene como objetivo diseñar e implementar una solución que le permita a GTSI visualizar en tiempo real las métricas de calidad de servicio que miden sus redes para mejorar la experiencia de usuario del servicio de Internet. Para el desarrollo del proyecto se empleó metodología SCRUM, para iterar sobre la solución inicialmente prototipada con la validación tanto del cliente como con la del tutor del proyecto. Como resultado se desarrolló una interfaz gráfica (tablero), 2 interfaces de intercambio de datos (API de tablero y Manejador de Puntos) y un componente de almacenamiento, compuesto por 3 bases de datos (Aurora, Firebase y Timestream). El Tablero reduce el desfase en el reporte y la visualización del estado real de las métricas de calidad de servicio.

Palabras Clave: Calidad de servicio, redes, WIFI, tablero.

ABSTRACT

The "Gerencia de Tecnologías y Sistemas Informáticos" (GTSI - ESPOL) manage a group of networks with Internet access; however, they have a lack of mechanism for reporting in real time all Quality-of-Service metrics. This project has as a main goal designing and implementing a solution to allow GTSI to visualize in real time all QoS variables for improving the Quality of Experience of their own network users. For the developing phase we used SCRUM as the main methodology in order to iterate over the solution with both client and tutor feedback. As result a graphic user interface ("Tablero"), 2 data exchange interfaces and a storage component composed by 3 different databases (Aurora, Firebase and Timestream). Finally, "Tablero" reduce the gap in the reporting and visualization of real state of the QoS metrics in their own networks.

Keywords: Quality of service, network, WIFI, Dashboard.

ÍNDICE GENERAL

RESUMEN.....	i
<i>ABSTRACT</i>	ii
ÍNDICE GENERAL.....	iii
ABREVIATURAS.....	v
ÍNDICE DE FIGURAS.....	vi
ÍNDICE DE TABLAS.....	vii
CAPÍTULO 1.....	1
Introducción.....	1
Descripción del problema.....	1
Justificación del problema.....	1
Objetivos.....	1
Objetivo General.....	1
Objetivos Específicos.....	1
Marco teórico.....	2
Lenguajes de modelamiento.....	2
Base de datos relacionales.....	2
Base de datos orientado a documentos.....	2
Servicios en la nube auto provisionados (Implementación de microservicios).....	3
Marco de desarrollo sin servidor (Implementación de microservicios).....	3
Marco de desarrollo web (Desarrollo de interfaz de usuario).....	3
CAPÍTULO 2.....	5
Metodología.....	5
Análisis.....	5
Requerimientos.....	5
Alcance y limitaciones de la solución.....	6
Riesgos y beneficios de la solución.....	7

Usuarios de la aplicación.....	7
Prototipado.....	7
Evaluación del prototipo.....	10
Mejoras del prototipo	11
Diseño de la Solución.....	13
Diagrama de casos de uso.....	13
Diagrama de arquitectura.....	15
CAPÍTULO 3.....	17
Resultados y análisis	17
Pruebas.....	17
Plan de implementación.....	18
Resultados.....	19
Análisis de costos.....	20
Costos de despliegue de la solución mensuales.....	20
Firebase hosting.....	20
Firebase Firestore.....	20
Amazon Timestream	20
Amazon Aurora	20
AWS IoT Core	20
Amazon API Gateway.....	21
Amazon Lambda.....	21
CAPÍTULO 4.....	22
Conclusiones Y Recomendaciones.....	22
Conclusiones	22
Recomendaciones	22
BIBLIOGRAFÍA	23
APÉNDICE	25

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral.
IOT	Internet de las cosas.
API	Interfaz de programación de aplicaciones.
GTSI	Gerencia de Tecnologías y Sistemas de Información.
PA	Punto de acceso.
HTTP	Protocolo de transferencia de hipertexto

ÍNDICE DE FIGURAS

Figura 2.1 Prototipo - Pantalla principal [autoría propia].....	8
Figura 2.2 Prototipo - Menú lateral [autoría propia]	8
Figura 2.3 Prototipo - Vista de datos históricos de métricas [autoría propia]	9
Figura 2.4 Prototipo - Vista de métricas en tiempo real [autoría propia]	10
Figura 2.5 Mejora de prototipo - Selección de métricas [autoría propia].....	11
Figura 2.6 Mejora de prototipo - Vista de métricas en tiempo real [autoría propia].....	12
Figura 2.7 Mejora de prototipo – Tema oscuro [autoría propia].....	13
Figura 2.8 Diagrama de caso de usos de API [autoría propia]	14
Figura 2.9 Diagrama de caso de usos de Interfaz de usuario [autoría propia].....	15
Figura 2.10 Diagrama de arquitectura de la solución [autoría propia]	16
Figura 3.1 Planificación de implementación [autoría propia].....	19

ÍNDICE DE TABLAS

Tabla 2.1 Requerimientos funcionales.....	5
Tabla 2.2 Requerimientos no funcionales	6
Tabla 3.1 Pruebas de aceptación de requerimientos funcionales.....	17
Tabla 3.2 Resultados de pruebas de aceptación de requerimientos funcionales.....	18

CAPÍTULO 1

Introducción

Descripción del problema

La Gerencia de Tecnología y Sistemas de Información (GTSI) de ESPOL mide la calidad de servicio de sus puntos de acceso a Internet para la mejora constante de la experiencia de usuarios de la red.

Sin embargo, para conocer el estado de la calidad de servicio se requiere de personal técnico que analice los registros del sistema y reporte manualmente el estado de la red, lo que genera un desfase en la identificación del estado real.

Justificación del problema

GTSI posee registros de estado de la red con acceso a internet que ofrece a ESPOL, sin embargo, estos datos son accesibles solo para personal técnico lo que limita la toma de decisiones oportunas para la mejora de la experiencia de usuario. Monitorizar lo que está ocurriendo con la red requiere de un profesional informático que entre al sistema, encuentre los registros objetivos y reporte el estado de la red. Este proceso de monitorización requiere ser automatizado para evitar posibles errores humanos y reducir el tiempo de ejecución.

Objetivos

Objetivo General

Desarrollar un tablero de monitoreo de calidad de servicio de redes con Internet Wifi mediante la implementación de aplicaciones web y el despliegue de servicios en la nube para la identificación del estado en tiempo real de los puntos de acceso pertenecientes a la red de GTSI - ESPOL.

Objetivos Específicos

1. Diseñar una arquitectura de flujos de datos usando herramientas de modelamiento que soporte los PA y redes para que la solución final sea escalable a otros modelos de negocio.
2. Implementar microservicios usando herramientas de computación en la nube para la administración de los datos de los puntos de accesos y redes.

3. Desarrollar una interfaz gráfica de usuario usando herramientas Web para la visualización y análisis de las métricas de calidad de servicio de una red.

Marco teórico

Lenguajes de modelamiento.

Un lenguaje de modelamiento es un conjunto de normas y reglas que permiten definir las características, alcances y limitaciones que tiene un sistema de software. Utilizar un lenguaje de modelamiento permite documentar un sistema de software, así como normalizar términos, definir módulos, herramientas y servicios que se vayan a implementar en la solución final.

UML es una técnica para la especificación de sistemas en todas sus fases que posee vistas, diagramas y símbolos [1].

Base de datos relacionales.

Una base de datos relacional (BDR) es un tipo de base de datos que estructura la información usando vínculos entre sí. Este tipo de bases de datos soportan el guardado, recuperación, actualización y borrado de los datos de la información principal de cualquier lógica de negocios. El uso de una de estas BDR nos permitirá soportar los datos de los puntos de conexión de Internet, así como los metadatos que generan.

MySQL es una BDR de código abierto con comportamiento rápido y confiable, escalable y que soporta tipos de datos compatibles con lenguajes de programación como JavaScript y Python [2].

Base de datos orientado a documentos.

Una base de datos orientada a documentos (DBOO) es un tipo de base de datos que soporta datos estructurados o semi estructurados que no necesariamente guarden relación entre entidades. Los formatos soportados son XML y JSON [3]. El uso de esta DBOO nos permitirá manipular la información no estructurada que se genere en la comunicación de nuestros servicios como por ejemplo los datos en tiempo real de los puntos de conexión de internet.

Firestore tiene un servicio de base de datos en tiempo real que permite almacenar los datos en documentos, hacer requerimientos y sincronizar datos entre

aplicaciones de tiempo real, adicionalmente tiene kits de desarrollo de software para lenguajes como JavaScript y Python [4].

Servicios en la nube auto provisionados (Implementación de microservicios).

Son servicios que ofrecen infraestructura para el despliegue, mantenimiento e integración continua de aplicaciones y plataformas. La idea principal del uso de estos servicios es poder enfocarse en la lógica de negocio dejando al proveedor el manejo y el aprovisionamiento de recursos como la memoria, poder computacional y tamaño de disco.

AWS proporciona servicios atractivos como ECS (Elastic Container Service) que administra aplicaciones que se ejecutan en contenedores y que a la vez proporciona alta confiabilidad al proveer escalamiento automático [5].

Marco de desarrollo sin servidor (Implementación de microservicios).

Es una forma de desarrollar software que va más allá de los servicios de auto aprovisionamiento ya que son piezas de código que son pensadas para ser ejecutadas bajo demanda y la idea es que el pago por estos recursos sean exclusivamente los consumidos, esquivando la estrategia de los servidores con alta disponibilidad y confiabilidad. Estos servicios serán de utilidad para darle alta confiabilidad a los servicios de actualización en tiempo real, ya que a primera vista se tendrá alta demanda en la recolección de los datos producidos por los puntos de conexión.

AWS lambda es consumo de CPU bajo demanda, que permite cargar código como zip o en contenedor donde se puede configurar un evento de ejecución ya sea por agendamiento o a través de un endpoint, además es compatible con lenguajes de programación como Javascript y Python [6].

Marco de desarrollo web (Desarrollo de interfaz de usuario).

Es una manera de programar en la web que posee directivas y librerías que agilizan el desarrollo de aplicaciones. Esta abstracción tiene sus ventajas como la encapsulación de código y la facilidad de reutilizar módulos que le ofrecen al programador sin embargo también tiene sus desventajas como la pérdida de

rendimiento de aplicaciones teniendo como referencia el tiempo de ejecución de una aplicación escrita en el lenguaje nativo.

ReactJS es una librería de desarrollo web para construir aplicaciones altamente reactivas, cuenta con un ambiente amplio teniendo librerías de interfaces gráficas, módulos de mapas entre otros [7].

CAPÍTULO 2

Metodología

Análisis

Requerimientos

Requerimientos funcionales

Se realizó una entrevista con el cliente para poder obtener información acerca de las principales características que debe tener la solución, las se pueden encontrar en la Tabla 2.1

No.	Descripción
RF-1	La aplicación debe mostrar un mapa interactivo para visualizar la ubicación de los puntos de acceso cuando el zoom es pequeño
RF-2	La aplicación debe mostrar en un mapa interactivo agrupaciones o cúmulos de puntos de acceso cuando el zoom es bastante grande
RF-3	La aplicación debe permitir seleccionar a manera de filtro la métrica a mostrar de los puntos de acceso
RF-4	La aplicación debe mostrar en el mapa interactivo un gráfico de dona las métricas de los cúmulos o puntos de acceso
RF-5	La aplicación debe permitir hacer zoom a los puntos de acceso
RF-6	La aplicación debe permitir realizar consultas históricas de las métricas de cada punto de dato
RF-7	La aplicación debe permitir actualizar puntos de acceso
RF-8	La aplicación debe permitir visualizar los datos en tiempo real

Tabla 2.1 Requerimientos funcionales

Requerimientos no funcionales

Luego de la entrevista con el cliente se pudo obtener también los criterios que debe tener la operación del servicio que brindará la solución. Los que están descritos en Tabla 2.2.

No.	Descripción	Característica
RNF-1	Los usuarios deben poder contestar preguntas de negocios con mínimo 5 acciones dentro de la aplicación.	Usabilidad

RNF-2	Los datos medidos por estampas de tiempos deben ser únicos.	Integridad de los datos
RNF-3	Los datos de los usuarios deben ser encriptados durante la comunicación entre APIs.	Seguridad
RNF-4	La aplicación debe permanecer activa las 24 horas del día, permitiendo una disponibilidad mínima de 99%.	Disponibilidad
RNF-5	La aplicación debe mantener un soporte para todos los navegadores actuales.	Portabilidad
RNF-6	La aplicación debe ser separada por módulos para un mejor mantenimiento del código.	Mantenibilidad
RNF-7	Debe existir documentación de los módulos de la aplicación.	Mantenibilidad
RNF-8	La aplicación debe mantener tiempos de respuestas de máximo 2 segundos.	Desempeño
RNF-9	La aplicación debe mantener soportar alta transaccionalidad	Desempeño

Tabla 2.2 Requerimientos no funcionales

Alcance y limitaciones de la solución

El presente proyecto contempla el desarrollo y la entrega del producto mínimo viable de la aplicación de monitoreo de calidad de internet de redes Wifi, cumpliendo con todos los requerimientos funcionales y no funcionales estipulados en la Tabla 2.1 y Tabla 2.2. La aplicación final solo comprenderá el monitoreo de las redes de GTSI - ESPOL.

Riesgos y beneficios de la solución

Los beneficios que ofrece la aplicación se ven reflejados en el tiempo que los administradores de las redes pueden llegar a obtener reportes y visualizaciones en tiempo real de cada uno de los puntos de acceso de las redes de GTSI.

En cuanto a los riesgos que pueden existir, están los fallos en los servidores de AWS a pesar de contar con una disponibilidad de 99.99%. La aplicación al utilizar los servicios de AWS está ligada fuertemente al rendimiento y disponibilidad que ofrecen los servicios usados.

Usuarios de la aplicación

Los principales usuarios de la aplicación son los administradores de las redes de GTSI - ESPOL. Quienes utilizarían la aplicación para monitorear los puntos de acceso para conocer el estado de cada uno de ellos y la calidad de servicio que estos ofrecen a los usuarios.

La plataforma no solo adopta el modelo de GTSI, sino que también se puede adaptar a otros modelos de negocio como puede ser el monitoreo de calidad de oxígeno en piscinas camaroneras.

Prototipado

Para la realización del prototipo se utilizó la herramienta Figma¹. Se tuvieron en cuenta los requerimientos funcionales del cliente para poder diseñar las siguientes pantallas.

En la Figura 2.1 se muestra la pantalla principal del tablero, en donde se pueden observar gráficos de dona, donde cada porción indica el porcentaje de puntos de acceso que califican en cada rango definido. El número central indica la cantidad de puntos de acceso que se encuentran agrupados en dicha área. En la parte inferior derecha se observa la leyenda que indica el significado de cada color para cada rango definido en los gráficos.

¹ Figma, <https://www.figma.com>

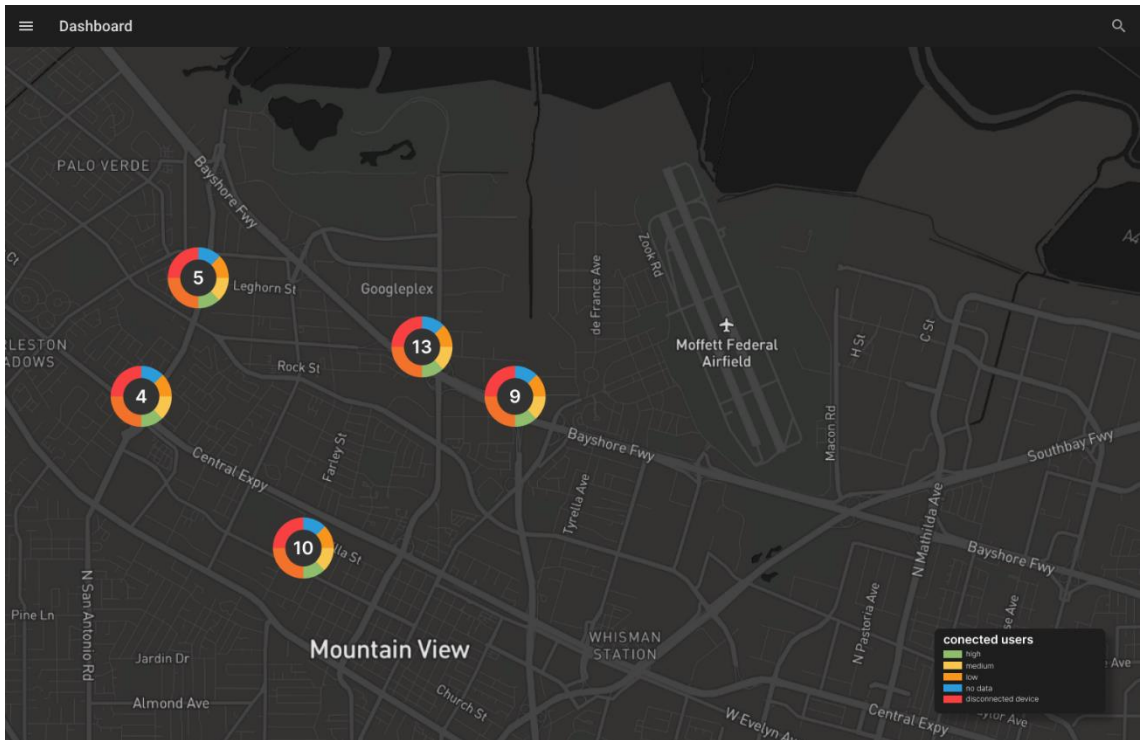


Figura 2.1 Prototipo - Pantalla principal [autoría propia]

En la Figura 2.2 se muestra el menú lateral en donde el usuario puede escoger el filtro que desea aplicar para cambiar la métrica de visualización.

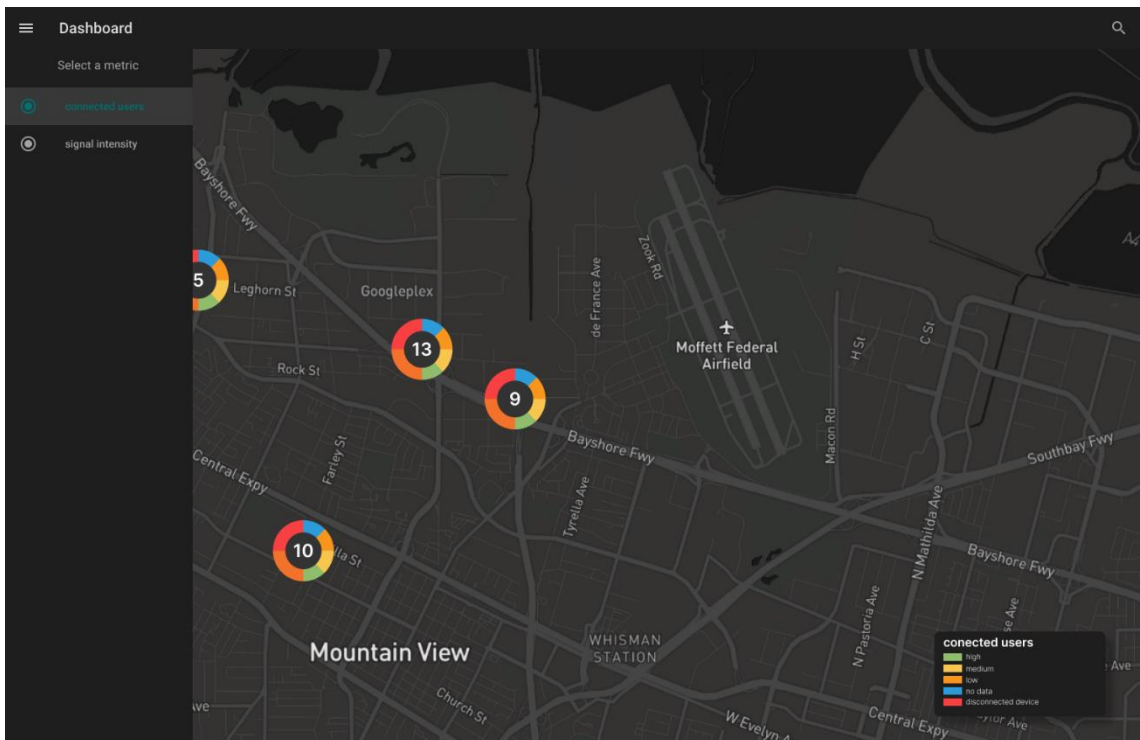


Figura 2.2 Prototipo - Menú lateral [autoría propia]

En la Figura 2.3 se muestra la sección de datos históricos de cada PA, el usuario puede filtrar por rangos de fecha personalizados o filtros estáticos como: hoy, ayer, última semana y último mes.

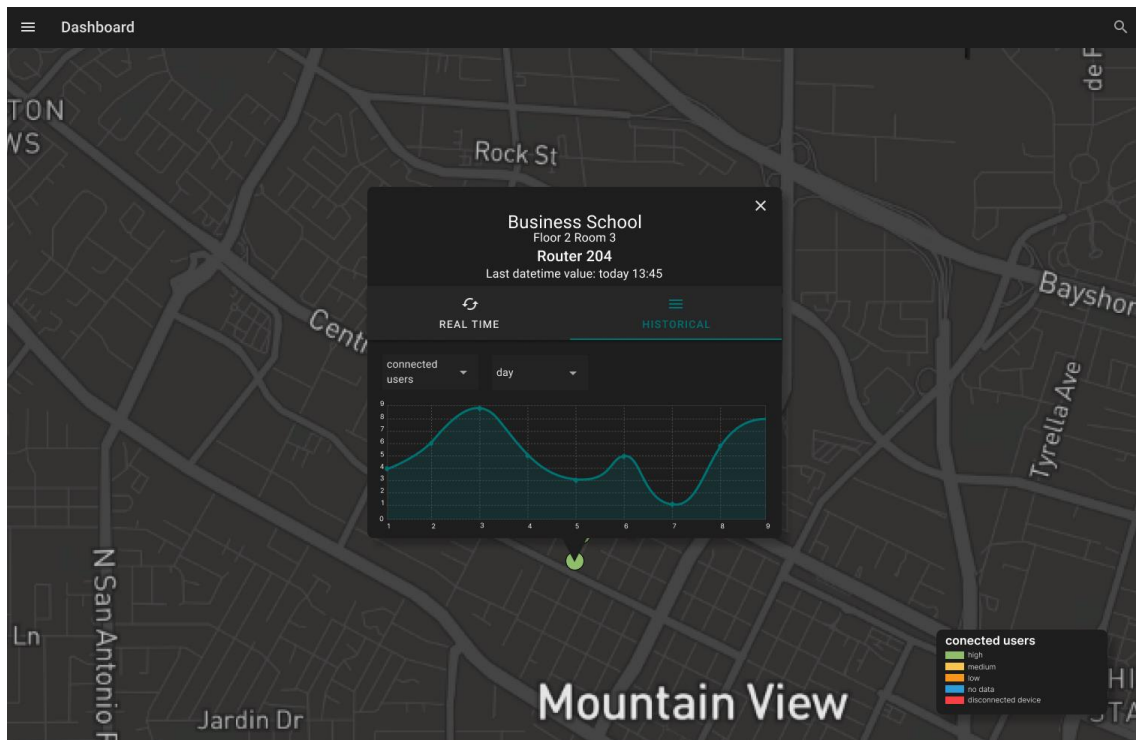


Figura 2.3 Prototipo - Vista de datos históricos de métricas [autoría propia]

En la Figura 2.4 se puede ver el detalle de cada PA con la última lectura de cada métrica en tiempo real.

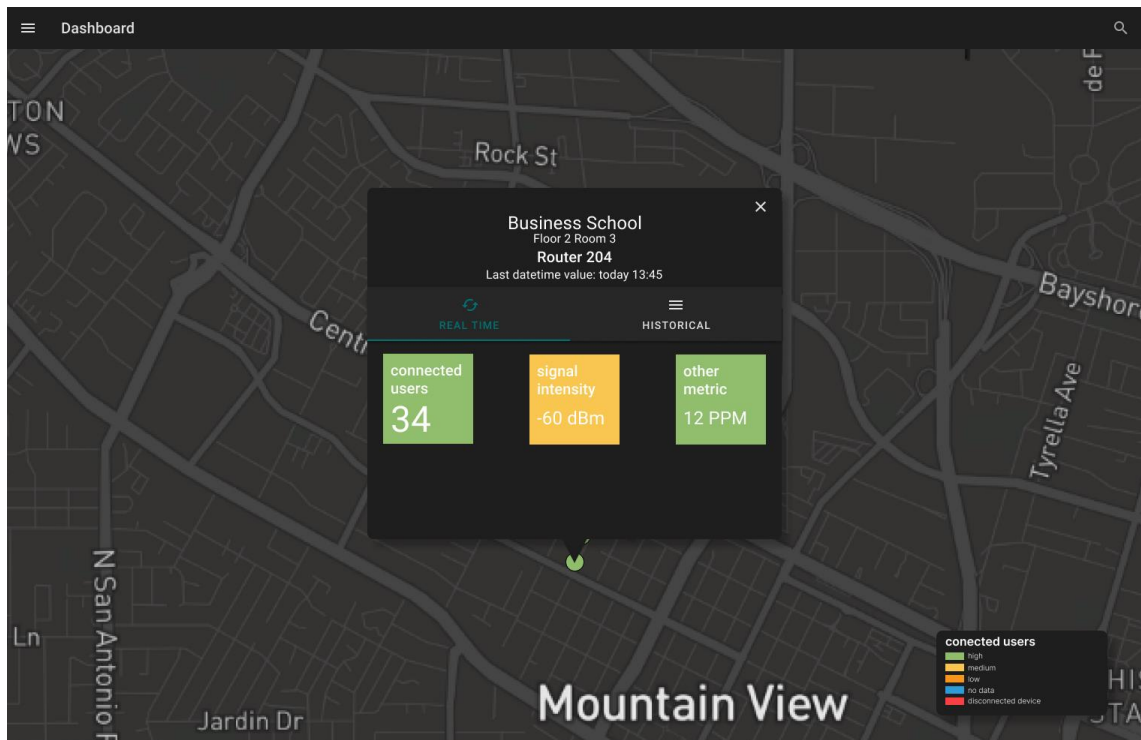


Figura 2.4 Prototipo - Vista de métricas en tiempo real [autoría propia]

Evaluación del prototipo

Se validó el prototipo con el cliente del cual se recibió retroalimentación de ciertas pantallas. Los puntos más importantes de la retroalimentación son los descritos a continuación:

- Mejorar el uso de los colores, debido a que se esperaba que estos hagan alusión a ESPOL.
- Mantener consistencia en el idioma utilizado, en este caso se recomendó utilizar español.
- Hacer uso de la ubicación en la cual va a ser usada la aplicación, es decir, reubicar los puntos en las cercanías de ESPOL.
- En la tabla de caracterización del atributo se recomendó hacer uso de rangos de valores, para que el usuario tenga una mejor retroalimentación del significado de dicha caracterización.

Además, se realizó una evaluación del prototipo con personal profesional en el campo de IoT con experiencia en el desarrollo de tableros similares al mostrado en la aplicación. De esta revisión se logró obtener críticas constructivas referentes a los gráficos usados para mostrar las métricas. Las recomendaciones fueron:

- El uso de un gráfico de dona no aportaba más información, en su lugar causaba ruido, por ende, se recomendó mantener un cúmulo describiendo solo el número de puntos agrupados
- Se recomendó mantener consistencia en el espaciado de las cartas mostradas en la sección de tiempo real
- Finalmente se procedió a realizar los cambios sugeridos obtenidos en las distintas revisiones.

Mejoras del prototipo

Luego realizar las respectivas validaciones del prototipo se procedió a ejecutar las mejoras. Como se muestra en la Figura 2.5 se procedió a utilizar colores relacionados a la institución para la cual va a ser realizada la aplicación. Además, se procedió a utilizar etiquetas en español y a reemplazar los gráficos de donas por simples cúmulo de puntos, para reducir la carga visual del usuario

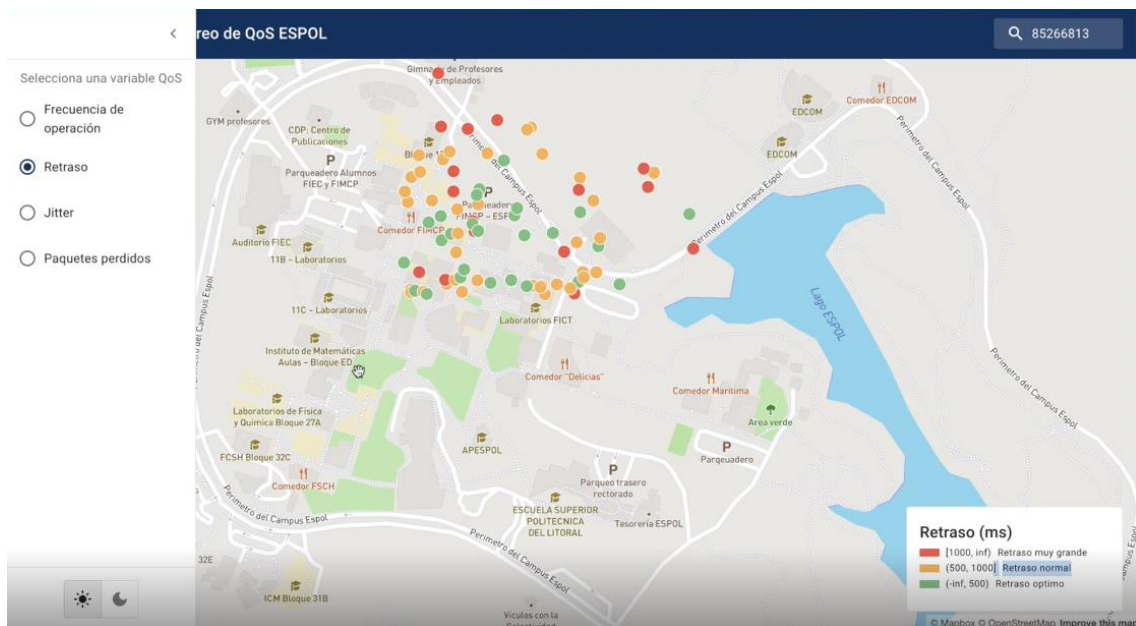


Figura 2.5 Mejora de prototipo - Selección de métricas [autoría propia]

Otra de las mejoras realizadas al prototipo es el uso de los colores referentes a los rangos de cada métrica para poder indicar si dicha métrica se encuentra dentro de los rangos normales de funcionamiento tal como se puede observar en Figura 2.6.

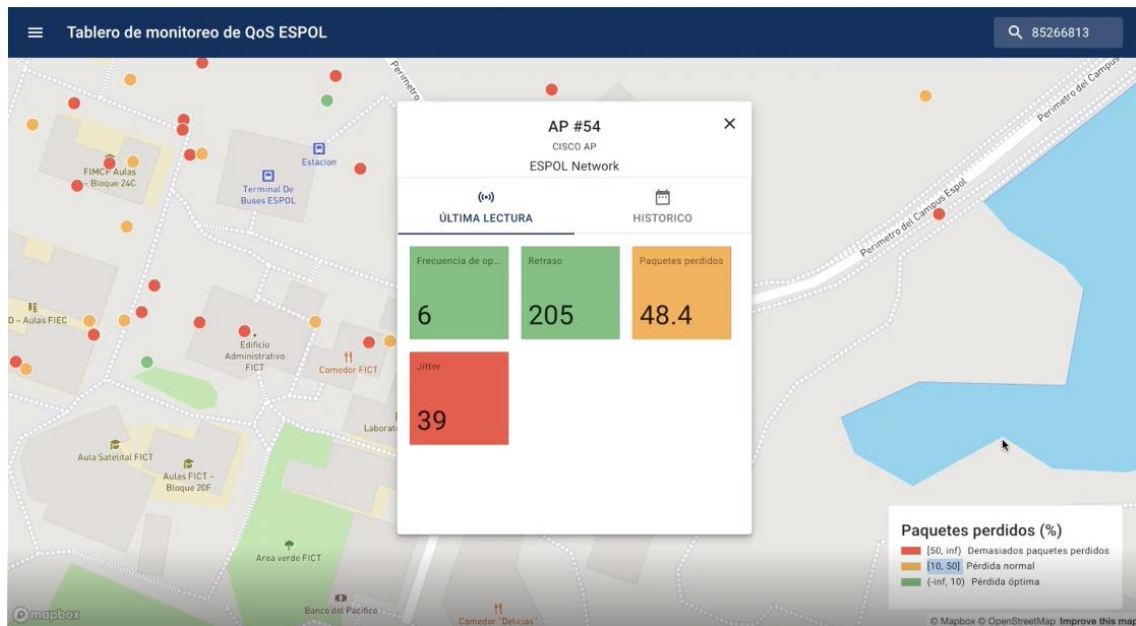


Figura 2.6 Mejora de prototipo - Vista de métricas en tiempo real [autoría propia]

Finalmente se procedió a agregar la opción de cambio de tonalidades de oscuro a claro cómo se puede visualizar en la Figura 2.7, para que el usuario decida entre estos dos temas de manera personal. Este cambio de tonalidad brindará la mejor gama de colores dependiendo de la hora del día, con la finalidad de poder

monitorizar las redes por largos periodos de tiempo evitando en lo posible la fatiga visual por alto brillo o contraste de la interfaz gráfica.

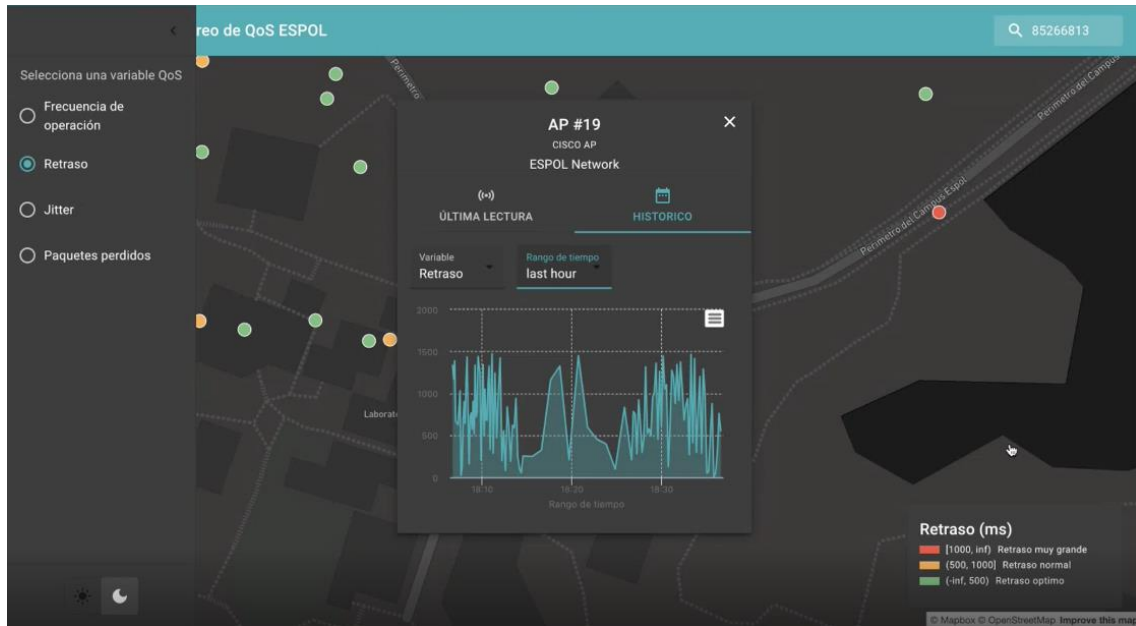


Figura 2.7 Mejora de prototipo – Tema oscuro [autoría propia]

Diseño de la Solución

Diagrama de casos de uso

En la Figura 2.8 se muestra el diagrama de caso de usos de los componentes de interfaz de comunicación de datos “API de tablero” y “Manejador de puntos”. En donde se describen alguna de las principales acciones que el usuario de esta interfaz debe ser capaz de realizar.

1. Recuperar y editar datos de los puntos de acceso.
2. Recuperar y editar datos de las redes.
3. Recuperar y editar datos de las métricas
4. Escribir y recuperar las lecturas realizadas por los puntos de acceso.

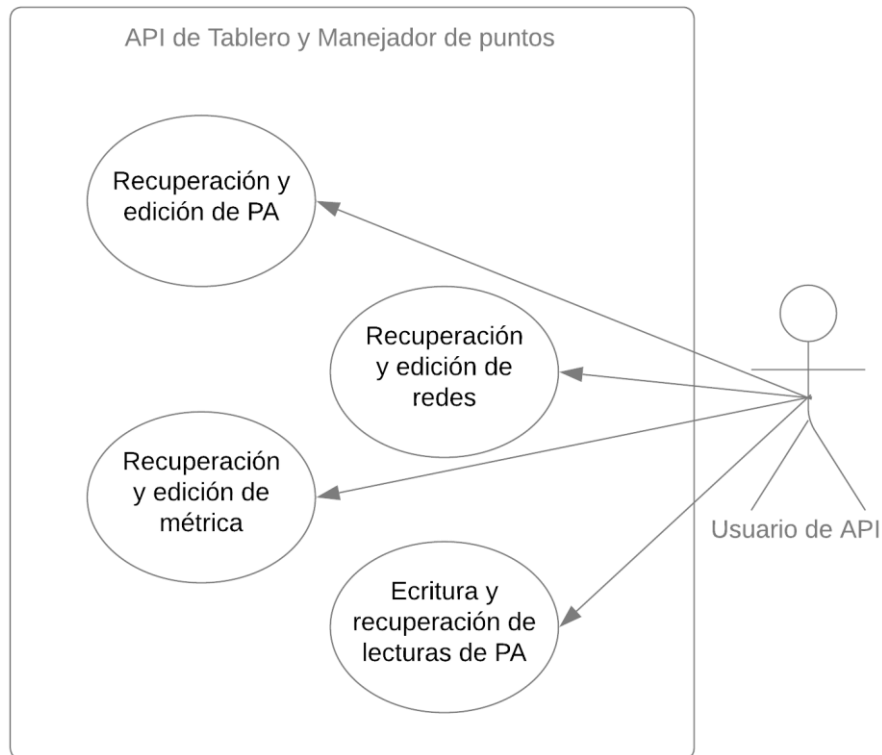


Figura 2.8 Diagrama de caso de usos de API [autoría propia]

En cuanto a los usuarios del Tablero, en la Figura 2.9 se muestran las acciones que los usuarios deber ser capaces de realizar en este componente gráfico. Entre dichas acciones tenemos:

1. Visualizar el tablero, en el cual se debe mostrar un mapa interactivo.
2. Visualizar puntos de acceso, estos nodos deben ser mostrados en la posición geográfica real dentro del mapa interactivo del tablero
3. Visualizar las métricas soportadas por la lógica del negocio, en este caso son frecuencia de operación, retraso, jitter y paquetes perdidos.
4. Visualizar las lecturas de los PAs en tiempo real.
5. Visualizar las lecturas de los PAs por rangos de tiempo.

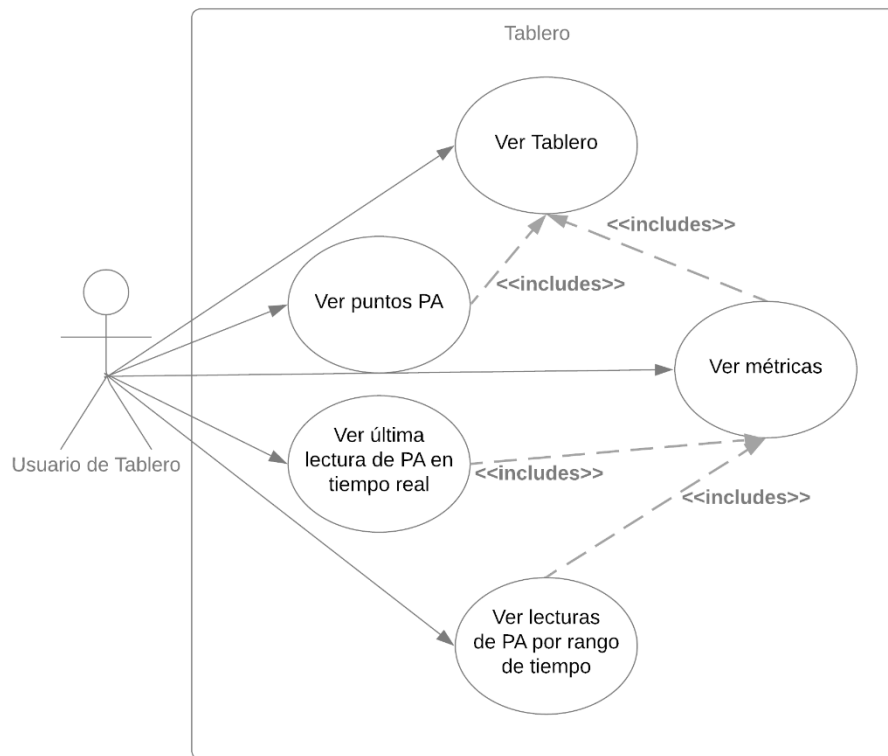


Figura 2.9 Diagrama de caso de usos de Interfaz de usuario [autoría propia]

Diagrama de arquitectura

Para la elegir los componentes utilizados dentro del diagrama de arquitectura se optó por la utilización de servicios en la nube de Amazon Web Services² y Google Cloud³.

En la Figura 2.10 se puede observar que el servicio de hospedaje de la aplicación web utilizada será Firebase hosting⁴. La aplicación web se comunicará con el API de Tablero para obtener información de las diferentes entidades que estarán almacenadas en Amazon Aurora⁵ (base de datos relacional).

El API de Tablero está conformado por Amazon API Gateway⁶ y Amazon Lambda⁷, los cuales son servicios que en conjunto permiten la implementación de un API REST de alta disponibilidad.

Otro de los componentes descritos es el API de manejador de puntos, el cual es la interfaz de comunicación de datos que utilizarán los desarrolladores para poder

² AWS, <https://aws.amazon.com/>

³ Google Cloud, <https://cloud.google.com/>

⁴ Firebase hosting, <https://firebase.google.com/products/hosting>

⁵ Amazon Aurora, <https://aws.amazon.com/es/rds/aurora/>

⁶ Amazon API Gateway, <https://aws.amazon.com/es/api-gateway/>

⁷ Amazon Lambda, <https://aws.amazon.com/es/lambda/>

realizar el registro de las lecturas obtenidas de los PAs. Este componente está formado 2 subcomponentes API de puntos y Computador de puntos.

El API de puntos recibe una solicitud HTTP para registrar el valor de la lectura de un PA y enviarlo a IoT Core a través del kit de desarrollo de Amazon. AWS IoT Core⁸ es un componente de colas de mensajes de alta disponibilidad es utilizado para registrar el valor de la lectura en el componente de almacenamiento.

El componente de almacenamiento está formado por 3 bases de datos: una instancia de Aurora (base de datos relacional), una instancia de Amazon Timestream⁹ (base de datos orientada a estampas de tiempo) y una colección de Firestore¹⁰ (base de datos orientada a documentos). Aurora nos provee de organización y la relación entre PA, métrica y las lecturas de las métricas; Timestream da soporte para la visualización de lecturas de las métricas a lo largo del tiempo; por último, Firestore guarda la última lectura de cada PA dando soporte para la visualización de métricas en tiempo real.

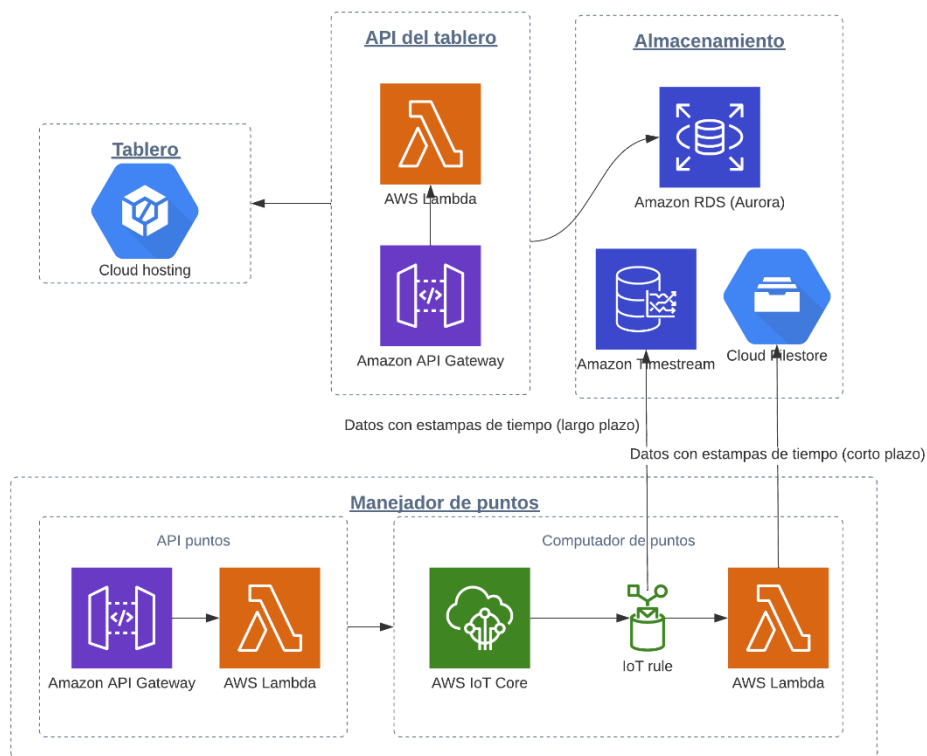


Figura 2.10 Diagrama de arquitectura de la solución [autoría propia]

⁸ AWS IoT Core, <https://aws.amazon.com/es/iot-core/>

⁹ Amazon Timestream, <https://aws.amazon.com/es/timestream/>

¹⁰ Firebase Firestore, <https://firebase.google.com/products/firestore>

CAPÍTULO 3

Resultados y análisis

Pruebas

En la Tabla 3.1 se describen los requerimientos funcionales, los criterios de aceptación usados para evaluarlos y el dispositivo por el cual se deben realizar las pruebas.

Id de prueba	Id de requerimiento	Criterio de aceptación	Dispositivo
P-1	RF-1	Se deberá renderizar en pantalla sin retrasos al menos 80 puntos de acceso	Computadora o teléfono inteligente
P-2	RF-3	Se deberá escoger entre al menos 4 métricas	Computadora o teléfono inteligente
P-3	RF-6	Se deberá consultar datos históricos de una métrica de un nodo de al menos 1 mes o 100 valores registrados	Computadora
P-4	RF-12	Se deberán obtener actualizaciones en tiempo real de un punto de acceso durante al menos 15 minutos de manera ininterrumpida.	Computadora

Tabla 3.1 Pruebas de aceptación de requerimientos funcionales

En la Tabla 3.2 se describen los resultados de las pruebas planteadas en la Tabla 3.1

Id de prueba	Resultados obtenidos	Detalle del resultado
--------------	----------------------	-----------------------

P-1	Criterio de aceptación aprobado	Se pueden renderizar en pantalla al menos 80 puntos de acceso.
P-2	Criterio de aceptación aprobado	Cada PA puede tener al menos 4 métricas asociadas.
P-3	Criterio de aceptación aprobado	Se puede consultar al menos 100 lecturas usando el filtro de rango de tiempo.
P-4	Criterio de aceptación aprobado	Se puede visualizar actualizaciones en tiempo real durante al menos 15 minutos.

Tabla 3.2 Resultados de pruebas de aceptación de requerimientos funcionales

Plan de implementación

Para la realización del proyecto se mantuvo la planificación mostrada en la Figura 3.1, que comprendía dos meses y medio. Durante el primer mes se procedió al desarrollo de la aplicación en ambiente local, para una posterior revisión con el cliente y tutor.

Luego con la retroalimentación obtenida se realizaron cambios en la aplicación para proceder a desplegar todos los componentes en ambiente de desarrollo y realizar una segunda revisión con el cliente y tutor del proyecto.

Finalmente se logró desplegar todos los componentes de la aplicación en ambiente de producción para realizar las pruebas correspondientes detalladas en la Tabla 3.1.

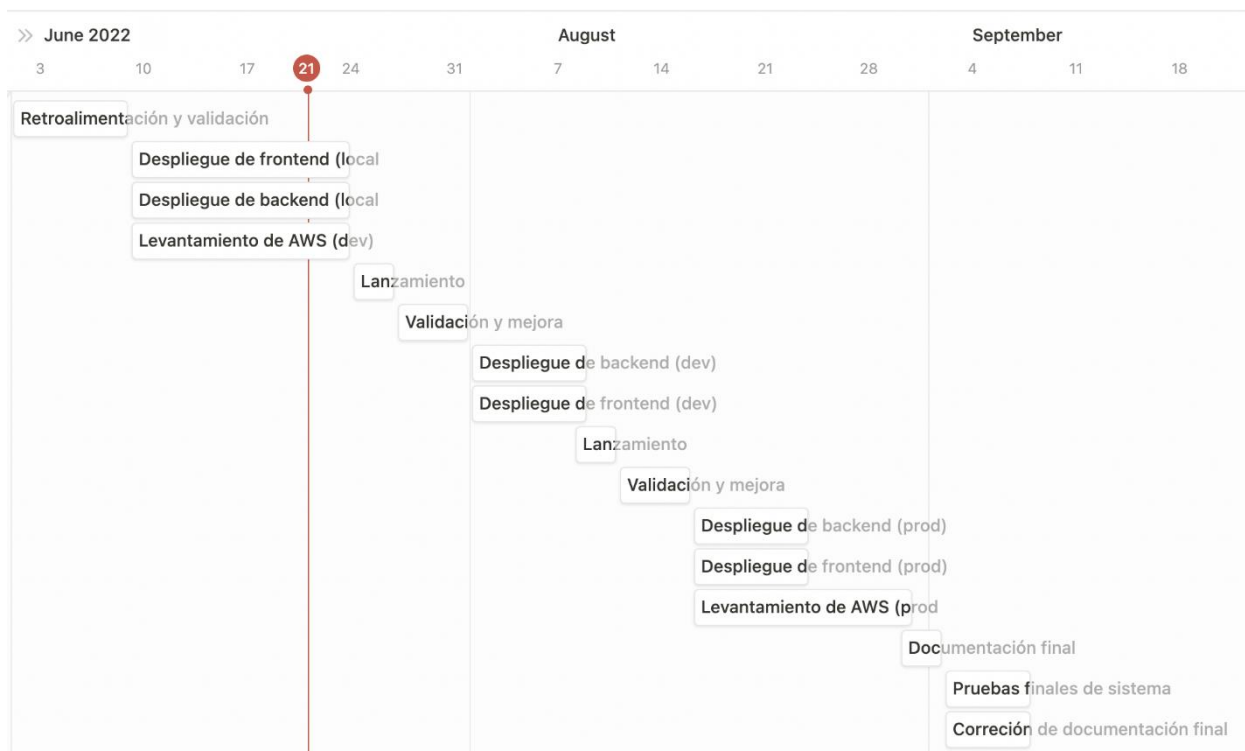


Figura 3.1 Planificación de implementación [autoría propia]

Resultados

Los 4 componentes de la plataforma fueron desplegados en servicios de Google Cloud y AWS. Estos servicios son auto aprovisionados ofreciendo el beneficio de la auto escalabilidad lo que se traduce en una disponibilidad del 99.99% anual. La implementación de Aurora con Timestream nos permitió relacionar las entidades de la plataforma optimizando el desempeño del manejo de grandes cantidades de datos. La implementación de Timestream con Firestore nos permitió ofrecer datos en tiempo real y datos históricos.

Uno de los principales beneficios de la solución es que, a pesar de ser capaz de soportar la lógica del cliente actual, también es posible adaptarse a otros modelos de negocio en los cuales se requiera del monitoreo de puntos de acceso. Un claro ejemplo puede ser el monitoreo de sensores de temperatura dentro de almacenes frigoríficos.

Análisis de costos

Costos de despliegue de la solución mensuales

Firestore hosting

El servicio de despliegue y alojamiento de la aplicación web utilizado es Firestore hosting el cual dentro de la capa gratuita ofrece un almacenamiento de hasta 10GB y con transferencia de datos de hasta 360MB por día.

Considerando un uso de 100 MB de transferencia diaria, se puede decir que el despliegue y alojamiento de la aplicación tiene un costo mensual aproximado de \$0.

Firestore

El servicio de base de datos en tiempo real utilizado es Firestore el cual ofrece dentro de la capa gratuita hasta 50.000 lecturas diarias y hasta 20.000 escrituras diarias. Considerando 20.000 escritura y 20.000 lecturas diarias, se puede decir que el costo mensual aproximado es de \$0.

Amazon Timestream

El servicio de base de datos de series de tiempo es Amazon Timestream el cual ofrece un almacenamiento a largo plazo y uno en memoria para los datos más recientes.

Para las escrituras se tiene un estimado de 20.000 escrituras diarias con un precio de \$ 1,45 por GB de datos registrados. Si asumimos que cada escritura tiene un tamaño de 200KB, los datos registrados al final de mes serían 120 GB lo que se traduce en un costo mensual aproximado de \$174.

Amazon Aurora

El servicio de base de datos relacional utilizada es AWS Aurora. Este servicio tiene un costo de \$0,06 la unidad de hora en soporte en ejecución, considerando 720 horas de uso al mes, se traduce en costo fijo de alrededor de \$43.20 al mes. El mismo servicio tiene un costo de \$0.26 por unidad de hora de ejecución de una instancia "large", teniendo en cuenta 100 horas de uso proyectadas, se traduce en aproximadamente \$26 (este valor puede variar según la cantidad de usuarios en la plataforma). Costo aproximado mensual \$69.20.

AWS IoT Core

Este servicio de IoT Core tiene una capa gratuita de hasta 500.000, lo que para el alcance de este proyecto (20.000 mensajes) se traduce en un costo aproximado mensual de \$0.

Amazon API Gateway

El servicio de ejecución de API remota utilizada es AWS API Gateway, el cual ofrece un costo de \$3.50 en los primeros 333 millones de requerimientos por mes. El número de requerimientos estimados de la aplicación es de 1 millón, razón por lo cual se espera que el costo aproximado mensual sea de \$0.

Amazon Lambda

El servicio de ejecución de funciones remotas utilizada es AWS Lambda. Dicho servicio ofrece un costo de \$0.20 por cada millón de requerimientos, teniendo en cuenta el millón de requerimientos, podemos decir que el costo aproximado mensual es de \$0.20.

Considerando el uso de los servicios anteriormente descritos, el valor aproximado total mensual asciende a \$243.4. Siendo este un valor referencial debido a que este aproximado considera la participación de un solo usuario en la plataforma.

CAPÍTULO 4

Conclusiones Y Recomendaciones

Conclusiones

- La arquitectura de flujo de datos permite dar soporte al modelo de datos GTSI – ESPOL, y de igual forma tiene la capacidad de adaptarse a otros modelos de negocio como puede ser el monitoreo de calidad de agua en piscinas camaroneras.
- La implementación de microservicios auto aprovisionados provee de escalamiento automático, lo que nos permite soportar alta transaccionalidad.
- El Tablero nos permite reducir el desfase que existía entre lo reportado manualmente y el estado real red. El Tablero cuenta con visualización en tiempo real de las métricas de calidad de servicio lo que reduce el desfase previamente mencionado.

Recomendaciones

- Ampliar el alcance de la plataforma añadiendo la característica de notificaciones por mensaje de texto o correo electrónico para monitorizar de manera pasiva el valor de las variables de calidad de servicio en caso de encontrarse un valor anómalo.
- Ampliar el alcance de la plataforma añadiendo la característica de reportes exportables.
- Implementar herramientas de integración y despliegue para automatizar la entrega de las actualizaciones.

BIBLIOGRAFÍA

- [1] Espol, «ESPOL apoya la transformación digital de Pymes en Ecuador, con lanzamiento de herramienta gratuita que mide la madurez digital,» 19 Octubre 2020. [En línea]. Available: <https://www.espol.edu.ec/noticias/espola-apoya-la-transformacion-digital-de-pymes-en-ecuador-con-lanzamiento-de-herramienta>.
- [2] Telefónica, «La digitalización de las empresas repercute hasta un 25% en su productividad,» 7 Marzo 2022. [En línea]. Available: <https://www.telefonica.com/es/sala-comunicacion/blog/la-digitalizacion-de-las-empresas-repercute-hasta-un-25-en-su-productividad/>.
- [3] Jorge Guerrón Eras, «Ecuador y su primera Ley Orgánica de Protección de Datos Personales,» 16 Junio 2021. [En línea]. Available: <https://dpd.aec.es/ecuador-y-su-primera-ley-organica-de-proteccion-de-datos-personales/>.
- [4] Universidad Andina Simón Bolívar, «Protección de datos,» 2020. [En línea]. Available: <https://www.uasb.edu.ec/ciberderechos/proteccion-de-datos/>.
- [5] L. E. Álvarez, «Paradigmas de la protección de datos personales en Ecuador. Análisis del proyecto de Ley Orgánica de Protección a los Derechos a la Intimidad y Privacidad sobre los Datos Personales,» *Revista de Derecho*, pp. 43-61, 2017.
- [6] M. R. Aguilar Martínez, D. P. Gordillo Cevallos, J. A. Paredes López y G. P. León Burgos, «La protección de datos personales en Ecuador,» *Estudios del Desarrollo Social: Cuba y América Latina*, p. 46, 2022.
- [7] V. Crissien Rueda, «Diseño de un aplicativo web para el análisis de la madurez digital de la Pymes. Laboratorio Digital,» 14 Junio 2021. [En línea]. Available: <https://repository.javeriana.edu.co/handle/10554/54114>.
- [8] ESPOL, «Fomentar la digitalización de las MiPymes en Ecuador es el objetivo de la ESPOL y los ministerios de Telecomunicaciones y de Producción,» 24 Enero 2022. [En línea]. Available: <https://www.espol.edu.ec/es/noticias/fomentar-la-digitalizacion-de-las-mipymes-en-ecuador-es-el-objetivo-de-la-espola-y-los>.
- [9] J. C. C. Gómez, «Protección de Datos Personales,» 2005. [En línea]. Available: <http://201.147.98.14/camara/content/download/193820/464897/file/datos%20personales.pdf>.

- [1 ZAP, «zapoxy,» [En línea]. Available: <https://www.zaproxy.org/getting-started/>.
0] [Último acceso: 01 07 2022].
- [1 W. Zeller y E. W. Felten, «Cross-Site Request Forgeries: Exploitation and
1] Prevention,» de *Cross-Site Request Forgeries: Exploitation and Prevention*,
Princeton, 2008.
- [1 National Vulnerability Database, «National Vulnerability Database,» 1 8 2021. [En
2] línea]. Available: <https://nvd.nist.gov/vuln/detail/CVE-2015-9251>. [Último acceso: 1 7
2022].
- [1 M. West y J. Medley, «web.dev,» 19 06 2020. [En línea]. Available:
3] <https://web.dev/csp/>. [Último acceso: 1 7 2022].
- [1 C. Shuptrine, «SameSite Cookie Attribute: What It Is And Why It Matters,» 08 06
4] 2020. [En línea]. Available: <https://www.kevel.com/blog/chrome-samesite/#:~:text=What%20is%20the%20SameSite%20attribute,a%20cookie%20to%20be%20accessed..> [Último acceso: 01 07 2022].
- [1 MDN Contributors, «Cache-Control,» Resources for Developers,, 30 05 2022. [En
5] línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Headers/Cache-Control>. [Último acceso: 1 07 2022].

APÉNDICE

MANUAL DE USUARIO

INTRODUCCIÓN

Propósito y alcance

Este documento tiene la intención de explicar las funcionalidades básicas del Tablero para un usuario no técnico. Se revisarán las características de visualización y sus funcionalidades.

Glosario

- Atributo: Característica cuantificable de un nodo
- Nodo: Punto de acceso.
- Red: Conjunto arbitrario de nodos (definidos por el usuario).
- Tablero: Interfaz gráfica de analítica mediante la cual el usuario extrae información de los puntos de acceso.
- Cúmulo: Representación gráfica de una red.
- IU: Interfaz gráfica.

Capacidades del sistema

Descripción general

La aplicación comprende un tablero de control, el cual permite al usuario obtener información relevante de los puntos de acceso de las redes de ESPOL en tiempo real y a través de un mapa interactivo en donde se puede visualizar la ubicación geográfica de cada punto de acceso. Dicha información comprende atributos de QoS (Quality of Service) como: frecuencia de operación, retraso, jitter y paquetes perdidos.

Descripción de funciones del sistema

Visualizar todos los atributos disponibles en el tablero.

Pasos para la navegación

- Abrir el tablero principal
- Hacer clic en el botón menú de la parte superior izquierda.
- Seleccionar un atributo diferente al predeterminado.

Detalle de la vista

Una vez se haya seleccionado el nuevo atributo se cerrará el menú y se mostrará el tablero con los nodos desplegados, pero con los valores actualizados correspondiente al nuevo atributo seleccionado. También actualizará la leyenda que caracteriza los valores y se mostraran los nodos pintados de acuerdo con el rango mostrado por dicha caracterización.

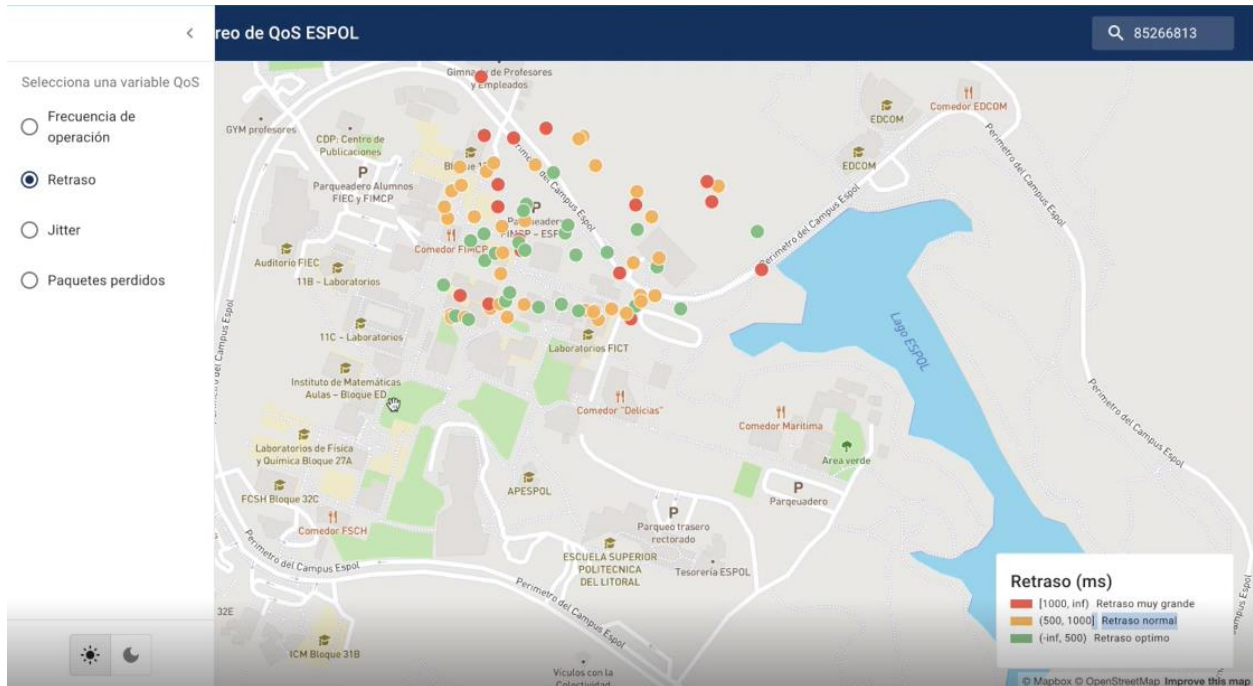


Figura A.1 Detalle de vista [autoría propia]

Descripción de funciones del sistema

Visualizar todos los nodos en el mapa

Pasos para la navegación

- Abrir el tablero principal
- Hacer clic en el cúmulo de nodos que se presenta en la pantalla
- Seguir haciendo clic en el cumulo de nodos que se desea visualizar de manera completa en la pantalla

Detalle de la vista

Se visualizarán todos los nodos en la pantalla pintados de acuerdo con la caracterización del atributo por defecto seleccionado.

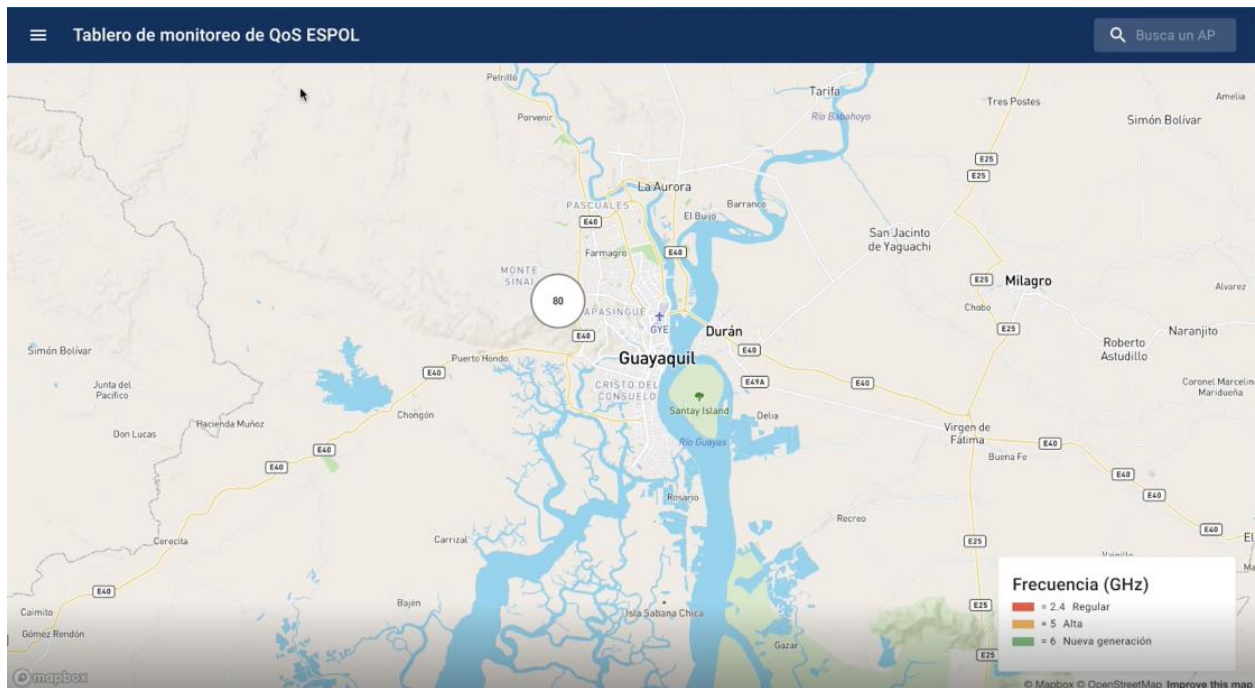


Figura A.2 Detalle de vista [autoría propia]

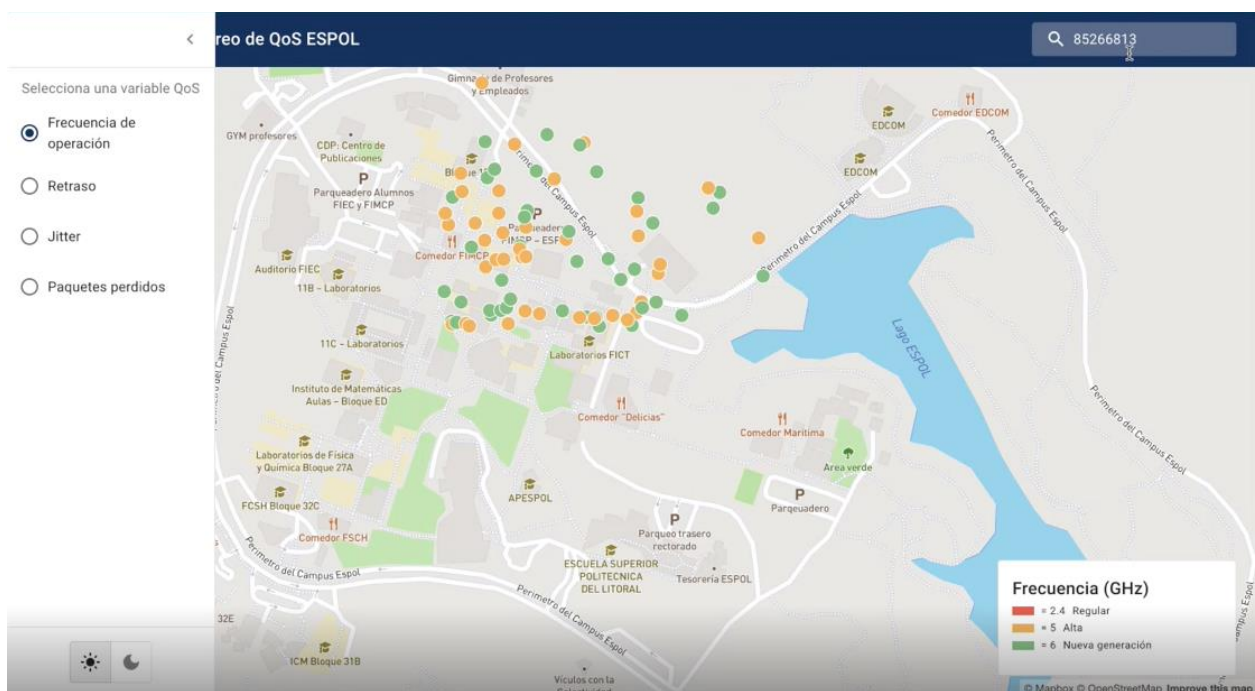


Figura A. 3 Detalle de vista [autoría propia]

Descripción de funciones del sistema

Visualizar información en tiempo real de un nodo en específico

Pasos para la navegación

- Dar clic sobre uno de los nodos visibles en el mapa o escribir el identificador del nodo en el buscador
- Se abrirá una ventana emergente y se debe hacer clic en la pestaña de última lectura.

Detalle de la vista

Se visualizará una ventana emergente con información del nodo seleccionado, y cartas pertenecientes a cada atributo con el valor de la última lectura captada de dicho nodo. Dicha carta se pintará de acuerdo con la caracterización de cada atributo.

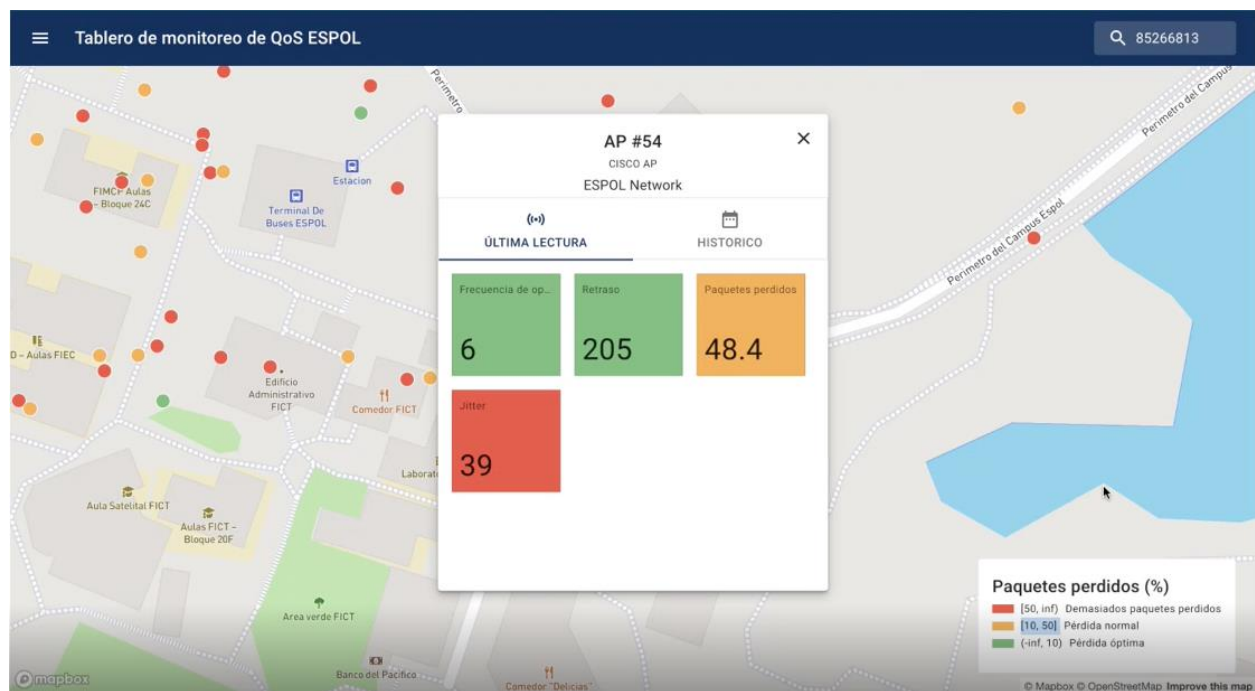


Figura A.4 Detalle de vista [autoría propia]

Descripción de funciones del sistema

Visualizar información histórica de un nodo en específico

Pasos para la navegación

- Dar clic sobre uno de los nodos visibles en el mapa o escribir el identificador del nodo en el buscador
- Se abrirá un modal y se debe hacer clic en la pestaña de histórico.
- Seleccionar el atributo que se desea visualizar

- Seleccionar el rango de tiempo del cual se desean observar lecturas.

Detalle de la vista

Se visualizará un modal con información del nodo seleccionado en conjunto con dos cajas de selección para poder seleccionar el atributo seleccionado y el rango de tiempo del cual se desea observar lecturas.

Debajo de las cajas de selección aparecerá una gráfica de líneas en donde el eje x es el tiempo y el eje y es el valor leído. La gráfica se va actualizando en tiempo real a medida que se vayan realizando lecturas

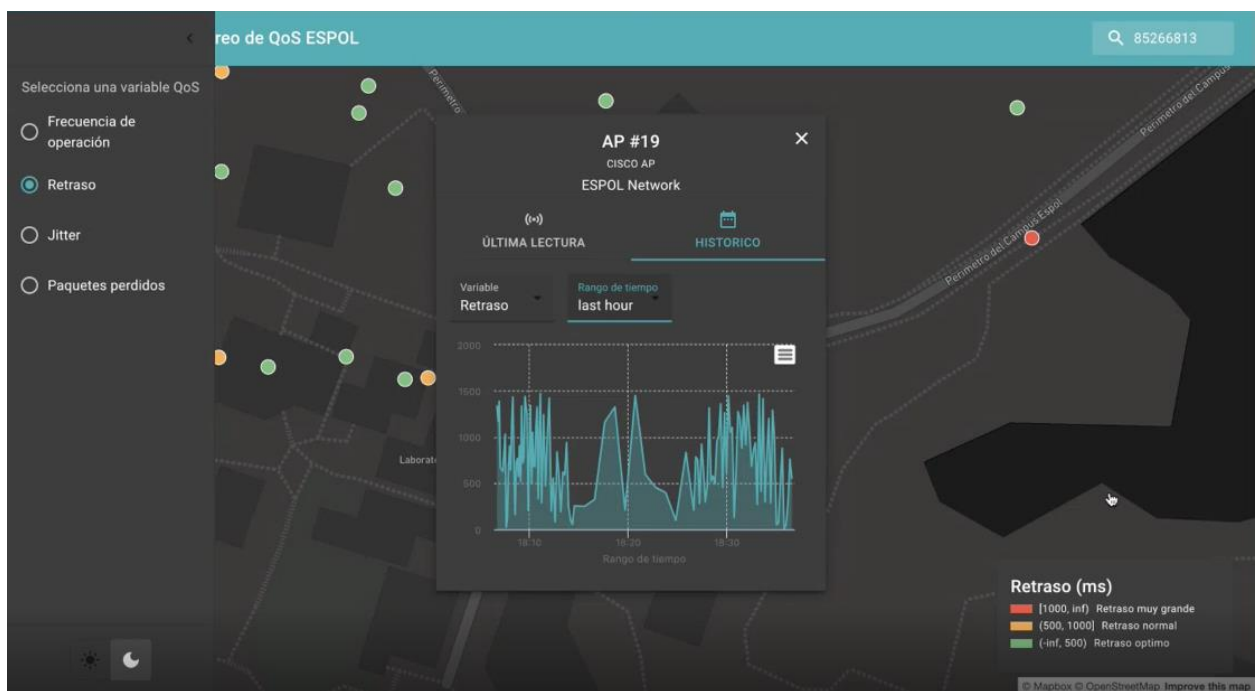


Figura A.5 Detalle de vista [autoría propia]