



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

**PLATAFORMA DE ANÁLISIS PARA
MEJORAR LA RESILIENCIA DE REDES
INALÁMBRICAS DE SENSORES**

FELIPE ANDRÉS SÁNCHEZ VARAS

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Profesor Supervisor:
CHRISTIAN OBERLI GRAF

Santiago de Chile, Marzo 2021

© MMXXI, FELIPE ANDRÉS SÁNCHEZ VARAS



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

PLATAFORMA DE ANÁLISIS PARA MEJORAR LA RESILIENCIA DE REDES INALÁMBRICAS DE SENSORES

FELIPE ANDRÉS SÁNCHEZ VARAS

Tesis presentada a la Comisión integrada por los profesores:

CHRISTIAN OBERLI GRAF

ANGEL ABUSLEME HOFFMAN

MAURICIO RODRIGUEZ GUZMÁN

MARCOS SEPÚLVEDA FERNÁNDEZ

Para completar las exigencias del grado de
Magíster en Ciencias de la Ingeniería

Santiago de Chile, Marzo 2021

© MMXXI, FELIPE ANDRÉS SÁNCHEZ VARAS

The true delight is in the finding out rather than in the knowing.

ISAAC ASIMOV

AGRADECIMIENTOS

Para comenzar, me gustaría agradecer a mi familia por haberme apoyado y acompañado en todo mi proceso formativo, tanto en la parte educacional como en lo emocional. Quiero destacar las distintas enseñanzas que mi hermana ha hecho durante todos estos años, junto con su motivación constante de seguir y aspirar a llegar lejos.

También quiero agradecer al profesor Christian Oberli por su apoyo, comprensión y motivación a lo largo de este trabajo de tesis, permitiéndome ser parte del grupo de investigación de LatinaUC. Sus enseñanzas en todo este proceso académico sin duda perdurarán, permitiéndome seguir creciendo para ser una mejor persona y un mejor profesional.

También agradecer a mis amigas y amigos que han estado presentes en este camino. En especial a María Ignacia y Camila por estar en los momentos que más lo he necesitado, por sus sugerencias, correcciones y apoyo continuo que sin duda han influido positivamente en mi desarrollo como mejor persona.

Debo mencionar también a todas mis mascotas: Perdy, Beijing, Dama, Kuki, Winnie, Orión y Blue, que si bien algunas ya no están presentes, su cariño me ha permitido salir adelante y han sido una parte importante a lo largo de todos estos años.

Finalmente, agradecer a profesores y compañeros de trabajo que han influenciado positivamente en mi desarrollo profesional: Angel Abusleme, Marcelo Guarini, Christian Dani Guzmán, Joaquín Venegas, Jean Paul de Villers Grandchamps, Roberto Guerrero, Matías Sepúlveda y Santiago Barros.

ÍNDICE GENERAL

| | |
|--|------|
| AGRADECIMIENTOS | IV |
| ÍNDICE GENERAL | V |
| ÍNDICE DE FIGURAS | VIII |
| ÍNDICE DE TABLAS | X |
| RESUMEN | XI |
| ABSTRACT | XII |
| 1. Introducción | 1 |
| 1.1 Contexto | 1 |
| 1.2 Línea de investigación | 2 |
| 1.3 Contexto de trabajo | 3 |
| 1.4 Problema y motivación | 3 |
| 1.5 Objetivos de la tesis | 5 |
| 1.6 Estructura del documento | 6 |
| 2. Revisión bibliográfica | 7 |
| 2.1 Resiliencia en Tecnologías de Información y Comunicación | 7 |
| 2.1.1 Resiliencia y robustez | 7 |
| 2.2 Fallas e impedimentos | 12 |
| 2.2.1 Clasificación de fallas | 13 |
| 2.3 Gestión de fallas | 16 |
| 2.3.1 Detección e identificación de fallas | 17 |
| 2.4 Conclusiones y enfoque buscado en esta tesis | 19 |
| 3. Herramientas disponibles | 20 |
| 3.1 Área de estudio | 20 |
| 3.1.1 Estaciones | 22 |

| | | |
|-------|---|----|
| 3.1.2 | SNAP: Sensor Network Administration Platform | 23 |
| 3.2 | SNAPY y necesidad de conocer el estado de la red | 24 |
| 3.2.1 | Analytics | 25 |
| 3.2.2 | Flujo detallado de Analytics | 27 |
| 3.3 | Limitaciones existentes | 29 |
| 4. | Metodología del trabajo | 31 |
| 4.1 | Solución propuesta | 31 |
| 4.1.1 | Reestructuración SNAPY e incorporación de base de datos local | 31 |
| 4.1.2 | Gestión de analíticas | 33 |
| 4.1.3 | CRUD de alertas | 34 |
| 4.1.4 | Escalabilidad del sistema | 35 |
| 5. | Análisis y resultados | 37 |
| 5.1 | Ejemplos prácticos de la aplicación de la plataforma en estaciones de alta montaña | 37 |
| 5.1.1 | Estado de carga y distancia. Caso estación Mirador Los Maquis | 37 |
| 5.1.2 | Estado de sistema de carga. Caso estación Tambor Bajo | 42 |
| 5.2 | Ejemplos prácticos de la aplicación de la plataforma en estaciones en red | 43 |
| 5.2.1 | Estado de sistema de carga. Caso estación Diego | 43 |
| 5.2.2 | Estado de estación. Caso Alto Las Vizcachas | 43 |
| 5.2.3 | Caso estación Repetidor Confluencia | 46 |
| 6. | Conclusiones, discusiones y trabajo futuro | 50 |
| 6.1 | Conclusiones de los resultados obtenidos | 50 |
| 6.2 | Aporte y discusiones | 51 |
| 6.3 | Trabajo futuro | 52 |
| | BIBLIOGRAFÍA | 55 |
| | APÉNDICES | 60 |
| A. | Diagramas de clases detallados | 61 |

| | | |
|-----|---|----|
| A.1 | Diagrama de clases de versión previa de SNAPY | 61 |
| A.2 | Diagrama de clases de versión nueva de SNAPY | 62 |

ÍNDICE DE FIGURAS

| | | |
|------|---|----|
| 2.1. | Disciplinas relacionadas con resiliencia (Sterbenz et al. (2010)). | 8 |
| 2.2. | Factores e indicadores de resiliencia en una red (Rafiei (2017)). | 10 |
| 2.3. | Clasificación de fallas. Imagen extraída de Moridi, Haghparast, Hosseinzadeh, y Jassbi (2020). | 16 |
| 2.4. | Clasificación propuesta de sistemas de gestión de fallas. Imagen obtenida de Moridi et al. (2020). | 17 |
| 3.1. | Topología Red Quebrada de Ramón. | 20 |
| 3.2. | Diagrama de una red de sensores. | 23 |
| 3.3. | Relación entre clases. (Fuente: creación propia). | 25 |
| 3.4. | Flujo de la realización de Analytics. (Fuente: creación propia). | 26 |
| 3.5. | Ejemplo de reporte generado. | 28 |
| 3.6. | Flujo del programa para generar Analytics. Desarrollo propio. | 28 |
| 4.1. | Diagrama de clases de versión nueva de SNAPY. | 32 |
| 5.1. | Estado de operación de batería en estación Mirador Los Maquis. | 38 |
| 5.2. | Estado de operación de sensor de distancia en estación Mirador Los Maquis. | 39 |
| 5.3. | Estación Mirador Los Maquis en la mantención realizada en abril de 2018. Foto cortesía LatinaUC. | 40 |
| 5.4. | Estación Mirador Los Maquis en el retiro realizado en noviembre de 2020. Foto cortesía LatinaUC. | 41 |
| 5.5. | Estado de operación de batería en estación Tambor Bajo. | 42 |
| 5.6. | Operación normal de estación Diego. | 44 |
| 5.7. | Códigos de diagnóstico en estación Diego. | 44 |
| 5.8. | Operación irregular de estación Alto Las Vizcachas. | 45 |

| | |
|---|----|
| 5.9. Códigos de diagnóstico en estación Alto Las Vizcachas. | 46 |
| 5.10. Topología de la red de la Quebrada de Ramón. Los colores más oscuros indican una mayor conexión con dicho vecino. Figura obtenida de Guerrero (2020). . . | 47 |
| 5.11. Estado irregular en estación Repetidor Confluencia. | 48 |
| 5.12. Códigos de diagnóstico en estación Repetidor Confluencia. | 48 |
| A.1. Diagrama de clases versión previa SNAPY. | 61 |
| A.2. Diagrama de clases versión nueva de SNAPY. | 62 |

ÍNDICE DE TABLAS

| | |
|---|----|
| 3.1. Estaciones y sensores de cada una de ellas. T indica temperatura y RH indica humedad relativa. | 21 |
| 4.1. Contenidos warningData. | 33 |
| 4.2. Tabla de alertas existentes. | 34 |

RESUMEN

En el Laboratorio de Tecnologías Inalámbricas UC (LatinaUC), las redes inalámbricas de sensores han sido una de las principales líneas de investigación. Estas se caracterizan por monitorear procesos de distinto tipo, siendo las aplicaciones que involucran eventos hidrometeorológicos los mayormente estudiados.

Las estaciones que contienen a los sensores se caracterizan por ubicarse en lugares remotos y en muchas ocasiones de difícil acceso, por lo que se ven expuestas a perturbaciones que pueden producir fallas y dificultar la llegada de los datos medidos. En función de esto, dar garantías de un funcionamiento resiliente ante las perturbaciones que una red pueda sufrir, es una característica fundamental que LatinaUC requiere. Para lo que es necesario poder evaluar el desempeño y proponer mejoras para los proyectos que han sido y continuarán siendo desarrollados.

LatinaUC cuenta con un software de desarrollo propio, SNAPY, el que tiene la capacidad de realizar analíticas diarias. Sin embargo, es insuficiente para realizar el procesamiento de los datos a un nivel que permita innovar, debido a una serie de limitaciones que posee, lo que dificulta la incorporación de nuevas capacidades. En este trabajo se propone una serie de mejoras para crear una plataforma de análisis que permita incrementar la resiliencia de redes inalámbricas de sensores. Se implementaron y validaron una serie de mejoras a SNAPY, permitiendo evaluar el comportamiento de una red ya desplegada en terreno, visualizando y analizando situaciones de operación normal y con fallas.

Esta plataforma tiene la capacidad de incorporar nuevas analíticas y generar nuevos conocimientos a lo ya estudiado, lo que permitirá sentar las bases para futuros estudios que sean realizados en LatinaUC, con el objetivo de lograr redes resilientes ante adversidades.

Palabras Claves: Redes de sensores inalámbricas, gestión de fallas, resiliencia, métricas, análisis.

ABSTRACT

In the Laboratorio de Tecnologías Inalámbricas UC (LatinaUC), wireless sensor networks have been one of the main research topics. These are characterized by monitoring different types of processes, in which the applications that involve hydro-metereological event have been the mostly studied.

The stations that contains the sensors are characterized by being located in remote places and often difficult to access, so they are exposed to disturbances that can cause failures and delay the arrival of the measured data. Because of this, a fundamental characteristic that LatinaUC requieres its be able to guarantee the resilience of a network in the face of disturbances. This would allow to evaluate the performance and propose improvements for the projects that have been and will continue to be developed.

LatinaUC has its own software, SNAPY, which has the capacity to perform daily analysis. However, it is insufficient to process data at a level that allows innovation due a series of limitations, which hinders the incorporation of new capabilities. In this work, a series of improvements are proposed to create an analysis platform to increase the resilience of wireless sensor networks. For this purpose, a series of improvements to SNAPY were implemented and validated, allowing to evaluate the behavior of a network already deployed in the field, visualizing and analyzing situations of normal and faulty operation.

This platform has the capacity to incorporate new analytics and generate new knowledge, which will allow laying the foundations for future studies that will be carried out in LatinaUC, with the objective of achieving resilient networks in the face of adversities.

Keywords: Wireless Sensor Networks, fault management, resiliency, metrics, analytics.

1. INTRODUCCIÓN

1.1 Contexto

En el contexto de Redes de Sensores Inalámbricas (*Wireless Sensor Networks, WSN* por sus siglas en inglés), hay numerosas aplicaciones que consisten en supervisar procesos. Por mencionar algunos ejemplos, se tiene el monitoreo de variables medioambientales y de recursos hídricos; vigilancia y seguridad; o de amenazas de origen natural, ya sea tsunamis, terremotos, erupciones volcánicas, entre otros. (Rafei (2017)).

Independientemente del uso que se le dé a las redes de sensores, es inevitable que ocurran interrupciones del servicio. Estas pueden ser causadas por fallas en los dispositivos, ataques cibernéticos, desastres naturales, entre otros. En estos casos, es necesaria la intervención humana para poder detectar fallas y realizar las acciones pertinentes para restaurar el servicio. Sin embargo, esta solución no es escalable, ya que con el crecimiento que se pronostica para la Internet de las Cosas (*Internet of things, IoT* por sus siglas en inglés), que producirá una mayor conexión entre dispositivos, muestra una necesidad que la tarea de monitorear sea automatizada o demande la menor intervención humana posible.

Existen algunas aplicaciones en que no es posible depender de un monitoreo humano para detectar e identificar situaciones que afecten el funcionamiento normal. Esto debido a que los sistemas se encuentran ubicados en áreas de difícil acceso y los datos cumplen un rol significativo a la hora de tomar decisiones sobre cómo enfrentar alguna catástrofe. Es por este motivo que existe una necesidad de poder detectar a la brevedad problemas en la red que permitan una mejor planificación para restaurar o mejorar el servicio.

Esta capacidad de poder realizar acciones para volver a un estado de operación normal, luego de una situación que haya alterado su funcionamiento, es conocida como la resiliencia del sistema. Para poder lograr esto se requiere de un sistema que pueda detectar las situaciones que alteren el estado y ser lo suficientemente confiable a la hora de entregar esta información a un controlador interno o externo a la red.

1.2 Línea de investigación

Dentro de las líneas de investigación del Laboratorio de Tecnologías Inalámbricas UC (LatinaUC), está el desarrollo de redes de sensores para monitorear procesos, siendo la medición de variables hidrometeorológicas el que más se ha desarrollado.

Para lo anterior, se han desplegado diversas estaciones que portan los sensores necesarios para recopilar los datos de interés; estas estaciones tienen la característica que suelen ubicarse en lugares remotos y, en muchos casos, de difícil acceso. Por lo que el conocer su estado, de forma rápida y eficaz, es fundamental para poder validar que los datos que está entregando son correctos.

En esta situación, la visión a largo plazo de LatinaUC es poder contar con estaciones individuales y redes de sensores que puedan ser robustas y resilientes ante los distintos eventos que puedan afectar su funcionamiento. Esto involucra que se pueda conocer el estado de cada una de ellas, detectar fallas y poder realizar acciones para restaurar el servicio de forma rápida y eficaz.

Esta capacidad de conocer el estado de cada una de las estaciones tiene varias formas de estudiarse, de las que destaca un método distribuido o centralizado. En la primera se refiere a que cada una de las distintas estaciones es la encargada de hacer un propio análisis de su estado e informarlo a sus vecinos o a la nube. Por otra parte, una visión centralizada se refiere a que el análisis de su estado es realizado por una estación encargada para esta labor o por un sistema externo a la red, que utiliza los datos generados por ella para inferir su condición.

Si bien cada una de estas metodologías poseen ventajas y desventajas que dependen de su aplicación, el objetivo de este trabajo de investigación es poder desarrollar un sistema centralizado y externo a la red, que busque comprender las capacidades de identificación de estados que se pueden lograr con la información disponible. Este sistema centralizado debe ser capaz de adquirir y procesar la información necesaria para cada uno de los análisis que se realicen y generar la información de forma que sea posible procesarla a futuro.

1.3 Contexto de trabajo

Para el trabajo de investigación de esta tesis, se utilizó uno de los primeros proyectos de LatinaUC, que corresponde al desarrollo de sensores para monitoreo de variables hidrometeorológicas, cuya investigación se remonta desde el año 2010. En este proyecto el grupo de investigación de LatinaUC desarrolló el *hardware* y *software* (Campamá (2012), Barros (2013)) necesario para poder ejecutar cada una de las tareas requeridas.

Uno de los primeros despliegues en terreno comenzó en el año 2014, en que se inició la instalación y los estudios necesarios para mantener una red de sensores inalámbricas en la cuenca de la Quebrada de Ramón, ubicada en la comuna de La Reina, ciudad de Santiago, Chile, con el objetivo de tener un prototipo de un sistema de alerta temprana de aluviones.

Al estar esta red desplegada desde bastante tiempo, permite utilizar los datos que se han ido generando para poder evaluar y testear las herramientas disponibles y las propuestas, ya que corresponden a datos reales de una red. Esta ha tenido periodos de buen funcionamiento, periodos irregulares y con fallas de algunas estaciones. Además, se tienen datos de estaciones que funcionan en red y de manera *standalone*, lo que la convierte en una gran fuente de información útil para un estudio.

1.4 Problema y motivación

La red de sensores mencionada en la sección anterior, es utilizada para medir distintos tipos de variables hidrometeorológicas que son entregadas a una base de datos en la nube, en la que un sistema analiza los valores reportados para alertar de la ocurrencia de aluviones en la zona. La red se compone de 17 estaciones, de las que 3 se consideran de alta montaña, ya que se encuentran sobre 2000 m.s.n.m.¹ y funcionan de manera independiente de la red con conexión satelital. Las restantes estaciones tienen una comunicación bidireccional que permite el intercambio de datos hasta un solo acumulador, que es la

¹Metros sobre el nivel del mar.

única estación con conexión a la nube, encargada de subir los datos mediante conexión celular.

Esta red, al estar en un ambiente de difícil acceso y expuesto a condiciones climáticas adversas, es inevitable que se vea enfrentada a interrupciones del servicio, las que pueden estar causadas por distintas razones. Es por esto que si se busca contar con un sistema eficiente y resiliente ante adversidades, es necesario poder identificar cuáles son y cómo poder gestionarlas.

En la línea de trabajo de LatinaUC es fundamental conocer el estado de una red, para así poder cumplir con los objetivos para los que esta fue desarrollada. Tomando el caso de la red de la Quebrada de Ramón, el objetivo es contar con los datos para alimentar un sistema de alerta temprana, por lo que poder validar que la red es robusta y resiliente, permitiendo que los datos lleguen en los plazos esperados, es una característica necesaria (Guerrero (2020)).

Previo al desarrollo de esta tesis, LatinaUC cuenta con *SNAPY*, acrónimo que une las palabras SNAP y Python. Corresponde a un software de desarrollo propio que se ejecuta sobre Python, el que hace uso de la API (*Application Programming Interface*) que permite la conexión a la plataforma SNAP (*Sensor Network Administration Platform*, por sus siglas en inglés), que es la base de datos en la nube que aloja todos los datos medidos. *SNAPY* permite consultar y generar reportes diarios del estado de las estaciones, realizar análisis numérico y proveer un set de herramientas para estudiar los sistemas desarrollados. Sin embargo, *SNAPY* aún es limitada y poco eficiente, haciendo muy difícil su escalabilidad y dificultando el desarrollo de la visión a largo plazo de LatinaUC. Dentro de las limitaciones detectadas, se pueden encontrar las siguientes:

- Tiempo de ejecución muy lento.
- Análisis ocurren solo en la ventana de tiempo sincronizada.
- Incapacidad de unir datos sincronizados localmente.
- Incapacidad de incorporar nuevos tipos de análisis sin cambios importantes.
- Información generada no es sintetizada.

Para iniciar el camino hacia la visión a largo plazo de LatinaUC, se comenzará a sentar las bases para una plataforma de gestión de estados, atendiendo las capacidades existentes de *SNAPY*, la que funcionará de forma centralizada y externa a la red. Esta decisión se basa en que el sistema operativo de las estaciones no tiene las capacidades necesarias para realizar el procesamiento. Se buscará que la plataforma pueda tener la facultad de poder integrar nuevos tipos de análisis, tanto diarios como históricos, optimizar su ejecución, evaluar el funcionamiento de una red y sentar las bases para que el sistema sea lo suficientemente escalable.

Esta escalabilidad del sistema se fundamenta en las posibilidades que surgen de la continuidad del presente trabajo. Esto para que futuros estudios realizados en LatinaUC puedan aportar a este sistema logrando abarcar más situaciones de fallas, lo que permitiría lograr redes de sensores resilientes a las distintas adversidades que se puedan ver enfrentadas.

Estos estudios también crean el camino para que las analíticas desarrolladas puedan ser incorporadas de forma nativa, sin requerir de un procesamiento centralizado que permita conocer su estado, sino que las mismas estaciones sean las encargadas de esta acción, a medida que su capacidad de procesamiento se incremente con el tiempo. Ya sea para realizar acciones que permitan mejorar la situación, informar al servidor algún problema existente o si se encuentran en una topología de red, que pueda haber un intercambio de información entre las distintas estaciones que permita mejorar la resiliencia completa de la red en cuanto a la forma de interconexión.

1.5 Objetivos de la tesis

El objetivo general de esta tesis es mejorar la capacidad actual de *SNAPY* como un sistema integrado de gestión de estado en una red de sensores, independientemente de su aplicación, siguiendo la línea de trabajo de LatinaUC. Para lograrlo, se deben cumplir los siguientes objetivos específicos:

1. Conocer e identificar los distintos estados de fallas a los que una red de sensores puede verse expuesta.
2. Mejorar la capacidad de procesamiento actual de *SNAPY*, eliminando las limitaciones de procesamiento existentes.
3. Implementar la capacidad de incorporar nuevas analíticas a *SNAPY*, sin afectar las existentes.
4. Validar la metodología con situaciones reales en las que ocurrieron fallas.
5. Ser independiente de la aplicación en la que la red es utilizada.

1.6 Estructura del documento

El documento está estructurado como se indica a continuación:

El capítulo 2 describe los conceptos asociados a este trabajo, explorando las extensiones de resiliencia y fallas en el contexto de redes de sensores. En este se darán a conocer los aspectos relevantes que un sistema de gestión de estados debe poseer y cómo esto guía el trabajo a realizar.

En el capítulo 3 se presenta la red utilizada para este estudio y el *software* con el que LatinaUC cuenta para analizar. Además, se detallarán las limitaciones encontradas y cómo esto incide en el presente trabajo.

En el capítulo 4 se presentará el desarrollo realizado, los cambios necesarios para poder mejorar las herramientas de análisis con los que LatinaUC cuenta, y las posibilidades que esto entregará a futuro.

En el capítulo 5 se mostrarán resultados obtenidos con la herramienta implementada para mostrar casos reales en los que ocurrieron fallas y cómo estos pueden ser detectados e informados por el sistema desarrollado.

Finalmente, en el capítulo 6 se presentarán las discusiones y conclusiones de lo realizado y se profundizará en los posibles trabajos futuros que esta plataforma le permitirá a LatinaUC.

2. REVISIÓN BIBLIOGRÁFICA

En este capítulo se muestra una revisión literaria en torno a los conceptos de resiliencia y gestión de fallas que serán utilizados para guiar el trabajo en esta tesis.

2.1 Resiliencia en Tecnologías de Información y Comunicación

En el contexto de Tecnologías de Información y Comunicación (ICT, por sus siglas de *Information and Communication Technologies*) la resiliencia se ha estudiado y distintos textos dan una definición que involucran o no otros conceptos (Klau y Weiskircher (2005); Hosseini, Barker, y Ramirez-Marquez (2016)). Sin embargo, aún no hay un claro entendimiento acerca de lo que resiliencia se refiere, en algunos casos se producen confusiones con conceptos relacionados, como lo es la robustez o adaptabilidad (Bishop, Carvalho, Ford, y Mayron (2011)), tratándolos como sinónimos. Además, al existir diversas formas de medirlo hace que sea complejo poder comparar unas con otras.

Dentro de ICT, se encuentra la tecnología de Redes de Sensores Inalámbricas o WSN por sus siglas en inglés (*Wireless Sensor Network*), las que se ven afectada por perturbaciones que afectan a su servicio. Es por esto que para hablar de resiliencia en redes de sensores y la importancia que tiene en su campo, se comenzará con una revisión literaria acerca de cómo la resiliencia es abordada.

2.1.1 Resiliencia y robustez

Resiliencia viene del latín *resiliere* que literalmente significa saltar hacia atrás (R.A.E. (2014)), sin embargo la definición en el contexto de WSN e ICT, es más amplia. En Laprie (2008), es definida como la persistencia de un servicio que puede ser justificadamente confiable en la presencia de cambios.

Al incorporar otros conceptos que tienen relación con resiliencia es posible diferenciarlo como la habilidad de una red de proveer y mantener un nivel aceptable de servicio frente a desafíos de una operación normal (Sterbenz et al. (2010)).

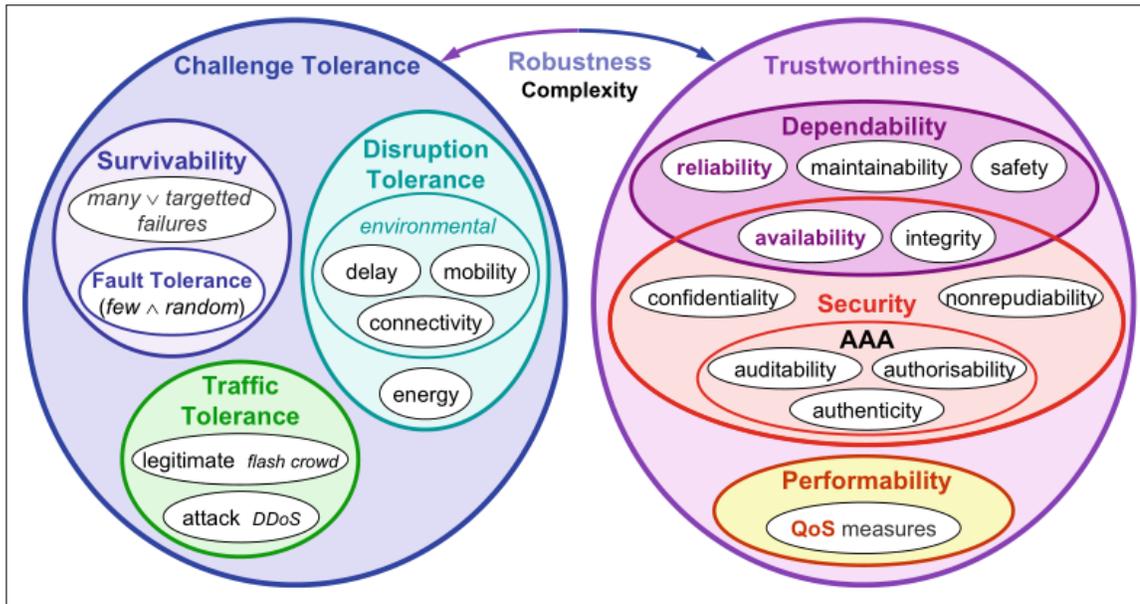


FIGURA 2.1. Disciplinas relacionadas con resiliencia (Sterbenz et al. (2010)).

En este mismo contexto, existe una gran cantidad de áreas que sirven como una base para lograr una red resiliente. Sin embargo, algunas de estas disciplinas se han ido desarrollando de forma independiente, por lo que a partir de la identificación de cada una de estas, es posible agruparlas como se ve en la Figura 2.1.

Estos conceptos abordan distintos aspectos de sistemas relacionados con ICT, la definición de los principales se presenta a continuación:

- **Tolerancia a desafíos**, el primer set de términos tiene relación con la tolerancia a cambios que ocurren en el sistema para mantener el servicio buscado, en particular se tiene:
 - **Tolerancia a fallas**, disciplina en que la tolerancia se basa en la redundancia, con el objetivo de evitar fallas que no estén correlacionadas. Es necesaria, pero no suficiente para ofrecer resiliencia.
 - **Supervivencia**, definida como la capacidad del sistema de lograr el objetivo de la comunicación, a pesar de posibles fallas de largo alcance, pudiendo incorporar diversidad en el sistema.

- **Tolerancia a la interrupción**, referido a que el sistema es capaz de tolerar problemas de comunicación entre sus distintos componentes.
- **Confiabilidad**, definido como la capacidad de un sistema de comportarse según lo esperado, basándose en:
 - **Fiabilidad**, cuantifica la dependencia que se puede poner el sistema para mantener una continuidad del servicio. Definido como la probabilidad de que el sistema se mantenga operable por un periodo específico de tiempo.
 - **Rendimiento**, cuantifica aquellas variables que permiten medir la calidad del servicio (*QoS, Quality of Service*).
 - **Seguridad**, referido a todo lo relacionado con que el acceso a la información sea solo para los dispositivos autorizados.
- **Robustez y complejidad**, dos conceptos que permiten unir la tolerancia a desafíos y la confiabilidad. Se relacionan en que el primero es la capacidad de mantener una operatividad del sistema ante perturbaciones y, por otro lado, la complejidad se refiere a la forma que una gran cantidad de sistemas pueden interactuar entre sí.

Otro estudio importante por mencionar es el de Rafiei (2017), quien hace una revisión extensa de resiliencia con un foco primordial en WSN. En este se considera como aquello que permite enfrentar distintas situaciones que afectan el normal funcionamiento de una red. A partir de esto elabora la conexión entre distintos factores que involucran la resiliencia de una red. Estas conexiones es posible visualizarlas en la Figura 2.2.

Incorporando el término robustez, en Kott y Abdelzaher (2014) es entendida como la habilidad de un sistema de tolerar eventos internos o externos sin la degradación de su desempeño. Además, la diferencia de resiliencia en que esta es la habilidad de un sistema de recuperar o regenerar su desempeño después de un evento no esperado, que ha causado impacto en el funcionamiento normal del sistema.

En Huang, Martínez, Sendra, y López (2015), es entendida como la habilidad de una red de resistir fallas, pero no necesariamente implica la habilidad de recuperarse de ella.



FIGURA 2.2. Factores e indicadores de resiliencia en una red (Rafiei (2017)).

En este trabajo también definen resiliencia como la habilidad de proveer y mantener un nivel aceptable de servicio frente a fallas, respecto de su operación normal. De esta forma, la red debe proveer y mantenerse operativa bajo condiciones adversas, como así proveer una rápida recuperación del mismo.

Para poder entregar un valor de resiliencia, en este estudio se introduce una medida que utiliza la topología de la red y cómo los distintos nodos que la conforman se conectan entre sí. Se dice que la red estará k -conectada si es que al menos todos los nodos tienen k o más conexiones.

En Pappas, Verma, Swami, y Ko (2008), se menciona un modelo de resiliencia basado en la probabilidades, específicamente en el sistema circulatorio. De aquí definen una métrica $i(n) \in [0, 1]$ que indica la fracción de nodos que dejan de comunicarse cuando falla el nodo n de la red. Para esto, la red completa tiene su medida de resiliencia dada por $R = \sum p(n)(1 - i(n))/N$, en donde $p(n)$ es la probabilidad de que el nodo siga funcionando a pesar de tener una falla en curso y N es la cantidad de nodos que tiene la red.

Luego de la revisión en lo que respecta a resiliencia y sin tener una definición precisa de ella, un aspecto importante a considerar en estudio y así dar garantías, es comprender cuáles son las causas que llevan a una falta de resiliencia. A partir de este enfoque, es posible notar que las causas pueden caer en dos categorías:

1. Si todos los elementos que forman parte de la cadena sensor a nube operan de la manera asignada, pero la eficiencia de la comunicación es insuficiente. Esto indicaría que existen problemas en el protocolo de comunicación utilizado, que no está funcionando de forma correcta.
2. En que uno o más elementos de la cadena sensor a nube están presentando fallas o algún impedimento que no permite que opere de la forma esperada. Esto podría ser causado por fallas de *hardware* o *software* y plantea la pregunta de cómo es posible realizar una detección, identificación y gestión de fallas que afectan a una red.

En esta tesis de magíster se analizará la segunda categoría, sentando las bases necesarias para poder dar garantías de resiliencia mediante una plataforma de análisis que lo permita. Para esto, primero es importante conocer cómo son las fallas que una red puede presentar y cómo estas pueden ser categorizadas, para lo que se realizará una revisión en torno a los conceptos de falla y sus derivados.

2.2 Fallas e impedimentos

El concepto de falla en redes de sensores inalámbricas ha sido estudiado desde diversos ámbitos. Tomando como punto de partida lo dicho en King y Nittel (2010), se menciona que una falla puede ser considerada como el desvío de las variable en un modelo predictivo ya conocido, al que se le asignan límites que indican que los valores están dentro de un rango esperado.

En Yuvaraja y Sabrigiriraj (2015), la falla es analizada en dos escenarios distintos: como una falla de nodo y una de red. En la primera, se considera que la falla es identificada por nodos vecinos, ya sea que el nodo informa de su mal funcionamiento o que los nodos vecinos analizan los datos que llegan. Por otra parte, las fallas de red en las que la topología completa de la red se ve afectada.

Complementando a las anteriores, en Selmic, Phoha, y Serwadda (2016), se desarrolla el concepto de *Fault-Aware*, en el que se es consciente de la existencia de las fallas, pudiendo detectarlas, aislarlas e idealmente mitigarlas. Para esto se plantea un sistema de control de fallas en las que se busca incrementar la robustez y resiliencia del sistema, pero que requiere de una mayor cantidad de información para asegurar su funcionamiento.

El concepto de impedimento se analiza en Roy, Misra, Kar, y Mondal (2017), donde se introduce el término *dumb node*, definido como aquel nodo que tiene una conexión inestable y afecta el desempeño de la red. La formulación matemática de un nodo con estas características se ve en la fórmula 2.1. En esta, Ψ_d indica que un nodo es considerado como *dumb node* o no, $r_c(t_i)$ es la distancia requerida para comunicación en un instante de tiempo t_i , d_{min} es la distancia mínima hacia un nodo vecino activo y R es la distancia máxima de comunicación.

$$\Psi_d = \begin{cases} 1, & \{0 < d_{min} \leq r_c(t_i) \leq R\} \\ & \wedge \{0 \leq r_c(t_i) < d_{min} \leq R\} \\ & \forall t_i \forall t_j t_i \neq t_j \\ 0, & \text{en otro caso.} \end{cases} \quad (2.1)$$

En la fórmula anterior se ve que un nodo tiene este comportamiento debido a que entra y sale del rango de comunicación con un nodo vecino de manera constante, posiblemente causado por condiciones atmosféricas. Para mantener la conexión y poder entregar los datos, se introduce un protocolo de ruteo que permite utilizar un enrutamiento alternativo y así llegar al destino.

2.2.1 Clasificación de fallas

Conociendo una definición inicial de falla e impedimento, se plantea la necesidad de identificar y poder clasificarlas, para así conocer de qué tipo son y con esto tomar las acciones necesarias. En este sentido en Selmic et al. (2016), se plantea una clasificación basada en niveles en los que podría ocurrir una falla, los que se presentan a continuación:

- **Nivel Componente**, que se enfoca a problemas de sensores que pueden ocurrir a nivel de sensores o fallas en la radio, en que esta última podría producir un problema de conectividad.
- **Nivel Nodo**, en el que podría involucrar una falla de conectividad de datos o de adquisición de los mismos. La detección de los mismos se produce ya sea en el mismo nodo, o en una colaboración de los nodos vecinos que son capaces de identificar el problema.
- **Nivel Red**, el que se enfoca en fallas que podrían ocurrir en la topología de la red, en la que si no se tiene un enrutamiento eficiente, se podrían producir problemas de congestión.

- **Nivel Sistema**, en el que se necesita información de todos los nodos para determinar fallas, por lo que se requiere de un sistema capaz de conocer y poder utilizar toda la información disponible.

Por otra parte, en Panda y Khilar (2015), la clasificación se considera dependiendo de la gravedad de la misma, partiendo de que una falla en un nodo ocurre porque no está comportándose cómo se debe, por lo que se separa en fallas leves y graves. Las primeras se refieren a un tipo de falla en que el nodo es capaz de mantener una comunicación, pero realiza un envío erróneo de datos; las últimas se refieren a aquellas en donde el nodo falla en comunicarse con el resto de la red.

En Zhang et al. (2018), las fallas son consideradas como un cambio inesperado o un funcionamiento erróneo del sistema, por lo que se puede considerar como una desviación del comportamiento respecto a un modelo esperado. A diferencia de Panda y Khilar (2015) las fallas son clasificadas respecto de sus características temporales, separándolas en tres categorías:

- **Fallas permanentes:** También conocidas como *hard fault*, debido a que involucra un corte en la transmisión, puede producirse por fallas de *hardware* y/o *software*.
- **Fallas temporales:** Son aquellas fallas que se presentan por un periodo, pero luego desaparecen o el servicio es recuperado.
- **Fallas intermitentes:** A diferencia de la anterior, es una falla cuya ocurrencia y mejoría es constante. Un ejemplo de esto puede ser un *impedimento* de un *dumb node*, como los mencionados en Roy et al. (2017).

En Moridi et al. (2020), se muestra un análisis más global, incorporando un nuevo concepto para la característica temporal de fallas y extendiendo la clasificación en función de los componentes afectados y área involucrada.

Respecto a los componentes, se consideran 4 subáreas en las que una falla se puede clasificar:

- **Falla de nodo:** Considerado el componente central de una red de sensores, las fallas pueden estar relacionadas a *hardware* o *software*.
- **Fallas de red:** Referido a los problemas de conexión que podrían ocurrir, estos podrían desencadenar un envío incorrecto de mensajes o retardos inaceptables. El caso crítico es un nodo que no es alcanzable, lo que hace que esa información sea perdida.
- **Falla de estación base:** Debido a que se encuentra en el nivel más alto de una red, una falla en este nodo puede producir una pérdida de datos de la red completa, debido a que el destino deja de estar presente.
- **Aplicación:** Referido a fallas que ocurren en donde la información es procesada, por lo que un mal manejo de esta o una falla en la información, podría producir una ineficiencia en el sistema completo.

También es posible clasificar las fallas que se presentan en las redes de sensores, en función del área que se ve afectada. Se consideran dos clasificaciones:

- **Local:** cuando solo 1 o más nodos se ven afectados por una falla.
- **Global:** aquellas en las que la falla de un solo nodo desencadena una de gran extensión. Esto, debido a que el nodo que falla puede ser la estación base o un nodo del tipo *cluster head*, que es aquel que tiene comunicación con un grupo de nodos y es el único que permite la conexión hacia la red.

Esta clasificación es posible verla en la Figura 2.3:

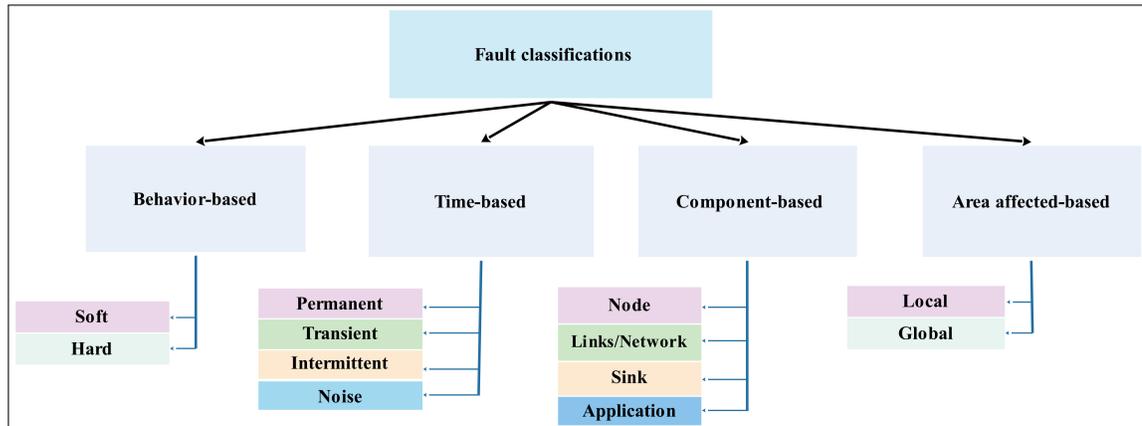


FIGURA 2.3. Clasificación de fallas. Imagen extraída de Moridi et al. (2020).

2.3 Gestión de fallas

La gestión de fallas se refiere a cómo un sistema es capaz de reaccionar y enfrentar la ocurrencia de las mismas. En este caso, la estructuración de gestión de fallas de Moridi et al. (2020) se separa en 3 grandes categorías:

- **Detección de fallas**, etapa que puede ser activa o pasiva, en la que se identifica la ocurrencia de una falla dependiendo de los síntomas que se tengan.
- **Identificación de fallas**, etapa en la que en función de los síntomas recopilados se procede a identificarla y clasificarla en las categorías mencionadas en la sección anterior.
- **Recuperación de fallas**, etapa en la que, dependiendo si es posible, se procede a recuperarse de la falla, cambiando conexiones o reiniciando los módulos, de ser necesario.

Existen diversas formas de abordar la gestión de fallas, tanto en Moridi et al. (2020) como en Zhang et al. (2018) se aborda una clasificación en función de los agentes involucrados y el lugar en que se mantiene y procesa la información. De aquí se tienen 4 principales categorías, cuya visualización gráfica se observa en la Figura 2.4.

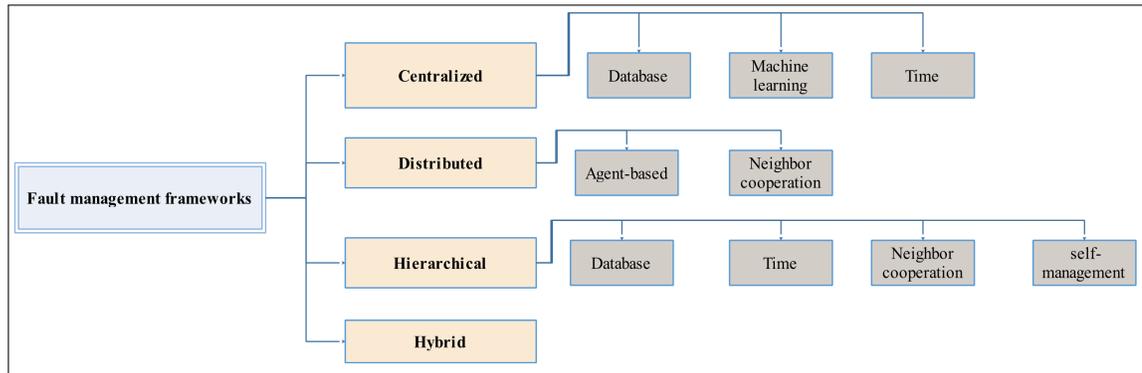


FIGURA 2.4. Clasificación propuesta de sistemas de gestión de fallas. Imagen obtenida de Moridi et al. (2020).

- **Centralizado:** En la que el procesamiento se concentra en el nodo base o en el servidor que recibe todos los datos de la red.
- **Distribuido:** En que el procesamiento es realizado en cada uno de los nodos o realizando una cooperación entre nodos vecinos.
- **Jerárquico:** En que el procesamiento es realizado por el nodo base, utilizando un sistema centralizado y distribuido.
- **Híbrido:** Mecanismo que una en distinto grado las separaciones anteriores, realizando el procedimiento no solo en un nodo principal.

Cada uno de estos métodos comparten lo mencionado al principio de esta sección. Para poder analizar fallas el proceso requiere de, al menos, detección e identificación de las fallas. Para comprender en más detalle esta necesidad a continuación se abordan estos conceptos.

2.3.1 Detección e identificación de fallas

Es conocido que los nodos se ven enfrentados a distintos tipos de perturbaciones, por lo que los sistemas deben estar condicionados a ser capaces de diferenciar la ocurrencia de un estado normal en contraste con uno anormal. Esto es lo que se conoce como detección e identificación de fallas.

Existen distintas metodologías acerca de cómo detectar fallas, estas se diferencian en el tipo de fallas que abordan y cómo el proceso es realizado, debido a que la organización puede ser centralizada, distribuida y autogestionada. Algunos de los trabajos que realizan los algoritmos de detección se presentan a continuación.

Una implementación que hace uso de redes neuronales es mencionada en Barron, Moustapha, y Selmic (2008), en esta se van utilizando las mediciones obtenidas por el nodo y por sus vecinos de forma de determinar si un valor que ha sido medido es correcto o no, identificando los límites en los cuales se debe mover. Utiliza un acercamiento distribuido.

Hay algunas fallas que no pueden ser identificadas directamente o la ocurrencia de algunos tipos de fallas no es completamente conocida. Sin embargo, en Miao et al. (2013) se introduce el concepto de *Diagnóstico Agnóstico*, referido a que la detección de fallas es realizada sin conocer la falla que está ocurriendo, para lo que se utilizan mapas de correlación de los datos obtenidos.

En Panda y Khilar (2015), se ve la detección de fallas mediante prueba de hipótesis en el mismo nodo, por lo que se considera un tipo de auto-gestión. En este se toma en cuenta la probabilidad de ocurrencia de una falla y a partir de esto se estima si un dato está o no dentro de un rango esperado, siendo capaz de autodetectar la ocurrencia de una falla.

En Yu, Hu, Li, Deng, y Yang (2014), se hace uso de SVM (por las siglas en inglés de *Support Vector Machines*) y una reducción de los datos a estudiar mediante un procesamiento centralizado. Al igual que en Zidi, Moulahi, y Alaya (2018), en donde se utiliza un clasificador SVM para identificar data correcta o errónea, focalizándose principalmente en detectar datos que escapan de rangos esperados o de comportamientos típicos.

En Alshammari y Ahmed (2017), se realiza un procesamiento híbrido en que los nodos son agrupados en *Clusters*, en cada uno de estos el procesamiento es llevado por un nodo cabeza, conocido como *Cluster Head*, que compara los datos entre los vecinos y es el encargado de comunicar al acumulador o a otros *Cluster Head* los nodos con problemas.

2.4 Conclusiones y enfoque buscado en esta tesis

Luego de la revisión bibliográfica sobre resiliencia y fallas, es posible notar que ambos conceptos son abordados desde puntos de vista muy amplios y diversos, lo que dificulta o imposibilita lograr un método generalizable para una variedad de fallas. Además, algunos de los métodos elaborados son teóricos y no han sido validados con datos reales de estaciones de monitoreo.

Junto con lo anterior, es posible notar que para poder hablar de resiliencia de una red y poder obtener una validación, es necesario tener claridad de los tipos de fallas a los que una red se verá expuesta. Lo que trae consigo la necesidad de gestión y esto a su vez involucra la detección e identificación de las mismas.

En lo que respecta a este trabajo, y en particular a LatinaUC, se busca la generación de una plataforma que permita realizar la gestión necesaria de los estados a los que una red se puede ver expuesta, desde un punto de vista centralizado. Esto para poder sentar las bases requeridas para poder comprender e identificar las fallas a las que las redes desarrolladas por LatinaUC podrían verse enfrentadas.

El objetivo a largo plazo es lograr la resiliencia de redes. Utilizando la información adquirida en la revisión bibliográfica, se comenzará con una plataforma centralizada, que permita generar los conocimientos y herramientas necesarias para analizar y estudiar los datos obtenidos. Con esta información en futuros estudios, y a medida que las capacidades de las estaciones continúen aumentando con el tiempo, el procesamiento podría ser realizado de forma híbrida o distribuida en cada una de ellas.

3. HERRAMIENTAS DISPONIBLES

En este capítulo se mostrarán las herramientas disponibles y una descripción del área de estudio sobre el que se trabajó en esta tesis. Se comenzará con una descripción de la red de la Quebrada de Ramón y la forma en que sus datos son accedidos, para finalizar con las características y limitaciones de las herramientas de análisis con las que LatinaUC disponía, previo al desarrollo del presente trabajo.

3.1 Área de estudio

El despliegue de la red de la Quebrada de Ramón comenzó en 2014, con el objetivo de tener una serie de estaciones de bajo costo que se situaran a lo largo de toda la zona (DICTUC (2017)). En la figura 3.1 se puede ver la ubicación de las estaciones.

La red cubre un área de 39,5km² con elevaciones desde 600 a cerca de 3000 m.s.n.m. Consiste en 17 estaciones, 13 de ellas conforman una red de sensores inalámbricas en



FIGURA 3.1. Topología Red Quebrada de Ramón.

TABLA 3.1. Estaciones y sensores de cada una de ellas. T indica temperatura y RH indica humedad relativa.

| Nombre estación | Altura (m.s.n.m.) | Coordenadas (Latitud, Longitud) | T & RH Aire | T & RH Suelo | Precipitaciones | Nivel de nieve | Nivel de agua | Estado |
|------------------------|-------------------|------------------------------------|-------------|--------------|-----------------|----------------|---------------|----------|
| Acumulador QR | 878 | (-33.434210,-70.516147) | | | | | | Retirada |
| Reforestación | 954 | (-33.431608,-70.511716) | • | • | • | | | Retirada |
| Diego (Acumulador) | 1007 | (-33.433980,-70.501510) | • | | | | | |
| Puente Los Peumos | 1046 | (-33.437140,-70.492180) | • | | | | • | |
| QR04 | 1162 | (-33.439116,-70.502333) | • | | | | | |
| Confluencia | 1178 | (-33.440100,-70.481600) | | | | | • | |
| QR05 | 1240 | (-33.440150,-70.501230) | • | • | | | | |
| Repetidor Confluencia | 1345 | (-33.442060,-70.485330) | • | • | | | | |
| Repetidor Los Peumos 2 | 1363 | (-33.432980,-70.491370) | • | • | | | | |
| QR07 | 1487 | (-33.448056,-70.495278) | • | • | | | | |
| Repetidor Apoquindo | 1596 | (-33.443850,-70.474400) | • | • | • | | | |
| QR08 | 1712 | (-33.450000,-70.487770) | • | | | | | Retirada |
| Alto Las Vizcachas | 1768 | (-33.426350,-70.474400) | • | • | | | | |
| Provincia Bajo | 2143 | (-33.434540,-70.451180) | • | • | • | • | | |
| Tambor Bajo | 2654 | (-33.447300,-70.435110) | • | • | • | • | | |
| Mirador Los Maquis | 2972 | (-33.463290,-70.433410) | • | • | • | • | | |

las que solo la estación “Diego” tiene conexión directa a la nube mediante enlace celular, actuando como el nodo acumulador. De las 4 restantes, la estación “Recinto Aguas Andinas” tiene conexión celular funcionando de forma independiente. Por otra parte las estaciones “Provincia Bajo”, “Tambor Bajo” y “Mirador los Maquis” son consideradas de alta montaña, debido al difícil acceso y que solo cuentan con conexión satelital.

Cabe mencionar que al momento de este estudio, una serie de estaciones habían sido retiradas del lugar con anterioridad, por lo que no se consideraron para el análisis. El detalle de cada una de estas, se encuentra en la tabla 3.1.

3.1.1 Estaciones

Cada una de las estaciones fue diseñada y fabricada en la etapa previa al despliegue en terreno por LatinaUC. El procesamiento y control de los procesos es realizado en un microcontrolador MSP430 de Texas Instruments. Aquellas estaciones que se encuentran en red se comunican utilizando el modulo de radiofrecuencia Digi XBee-Pro S1 802.15.4, que funciona en la banda de 2.4 GHz. La estación acumuladora posee el módulo Cinterion BGS2 wireless M2M Module, para enviar los datos a la nube. Por otra parte, las estaciones satelitales utilizan la red Iridium para este propósito.

El sistema operativo que las estaciones utilizan es denominado LatinOS (Campamá (2012)), propiedad de LatinaUC. Este es un desarrollo basado en Sensorscope, que a su vez está basado en otro sistema operativo para redes de sensores inalámbricas, TinyOS (Barrenetxea et al. (2008); Ingelrest et al. (2010)). El diseño del desarrollo del protocolo de comunicaciones implementado en LatinOS es abordado en Barros (2013), al que posterior a la fecha de publicación, se le han realizado una serie de modificaciones para mejorar la inicialización y sincronización de la red, conexión con vecinos, mejoras en la administración energética, entre otros.

Además, de los sensores indicados en la tabla 3.1, las estaciones envían información acerca del nivel de batería, número de *hops*¹ al acumulador (si es que se encuentra en red), y una serie de datos de diagnóstico producidos por LatinOS que informan del estado de los sensores, reinicios, entre otros.

Además, en la última actualización del sistema operativo se agregó la capacidad de guardar la “Tabla de vecinos” cada 5 minutos (Guerrero (2020)), la que indica la calidad de los enlaces con las que cada estación se puede comunicar. Sin embargo, esta información se encuentra solo disponible en la tarjeta SD incorporada en las estaciones y no es accesible a través de SNAP (*Sensor Network Administration Platform*). Es por esto que, para efectos de esta tesis, estos datos no se encuentran incorporados al sistema.

¹El número de *hops* se refiere a la cantidad de estaciones por las que un dato debe atravesar para poder llegar a la estación de destino

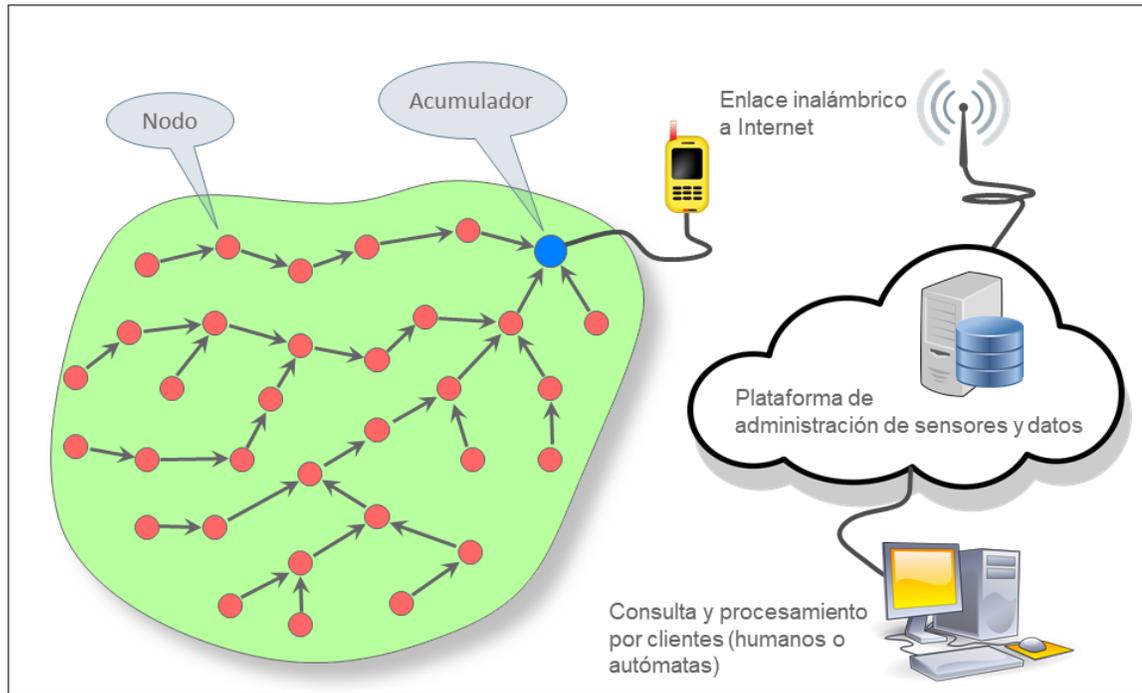


FIGURA 3.2. Diagrama de una red de sensores.

3.1.2 SNAP: Sensor Network Administration Platform

Cada dato medido por las estaciones es subido a la plataforma SNAP y es respaldado en la tarjeta SD. Para que esta información llegue a la nube, si es que el dato fue medido en una estación de alta montaña, hará uso de la red satelital Iridium. Por otra parte, si los datos fueron adquiridos dentro de la red, serán enviados al acumulador mediante los mecanismos indicados por el protocolo de ruteo de LatinOS (Barros (2013)), para ser subidos a la nube mediante conexión a red celular. Un diagrama de la red es mostrado en la figura 3.2.

Cada paquete de dato subido a la plataforma SNAP es tratado de forma indistinta y la información que cada uno contiene es:

- Un número ID, que identifica a la estación que midió este valor.
- Un número ID, que identifica a la variable asociada.
- El valor medido como un número entero de 16 bits.
- La hora de medición como valor UTC.

Estos valores son complementados en la plataforma con un identificador único para cada paquete, denominado ID del dato. Antes de guardarlos en la base de datos, la plataforma SNAP revisa que el valor medido esté dentro de rangos válidos para dicho sensor, ya que en caso contrario el dato es enviado a cuarentena. Para acceder a la información contenida SNAP está equipado con una API SOAP que permite la comunicación computador a nube. Dentro de las funciones disponibles, se encuentran las siguientes:

- `getSites`. Esta función permite obtener información de los grupos, sitios que se encuentran dentro de estos grupos y variables disponibles por cada sitio.
- `getData`. Esta función permite obtener datos filtrados por sitio, variable, grupo y rango de fechas. Cada uno de estos campos es opcional, por lo que para tener conocimiento de los distintos ID, es necesario utilizarla en conjunto con la función `getSites`.

La API contiene otras funcionalidades para el control de usuarios y contratos, además de poder enviar datos manualmente a cuarentena, sin embargo para el desarrollo de este trabajo no fueron considerados. Para acceder a la información, cada usuario debe contar con una API *key* que restringe el acceso a la plataforma y administran los permisos que dicho usuario posee.

3.2 SNAPY y necesidad de conocer el estado de la red

Esta herramienta tiene las clases necesarias para ejecutar programas de análisis. La relación entre las distintas clases que lo componen se puede observar en la figura 3.3. Cabe señalar que para una mejor comprensión se muestra una versión simplificada del diagrama de clases, si es que se desea observar con mayor detalle este se puede observar en el anexo A.1.

Cada una de las clases se describen a continuación:

- **`soapsites`**. Clase encargada de la comunicación con la API para hacer uso de la función `getsites`.

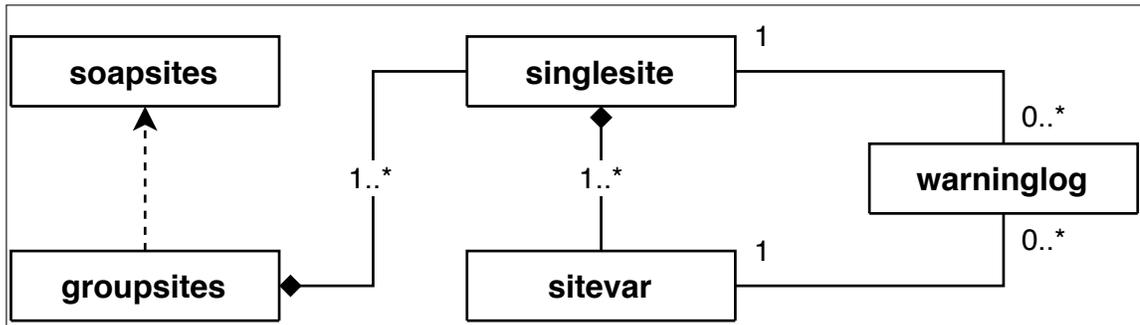


FIGURA 3.3. Relación entre clases. (Fuente: creación propia).

- **groupsites**. Clase que crea un objeto que modela a un grupo desplegado. Contiene información de los sitios y variables. Cabe mencionar que un grupo puede contener solo 1 o varios sitios.
- **singlesite**. Clase que crea un objeto que modela a un sitio desplegado. Contiene información del sitio y de las variables asociadas a él. Cabe mencionar que puede tener solo 1 o varias variables, además de tener o no información de su estado.
- **sitevar**. Clase que crea un objeto que modela a una variable en un sitio en específico. Contiene información de la variable y todos los datos en un rango de tiempo determinado. Permite ejecutar distintos análisis relacionados con la variable y guarda la información del estado de esta variable.
- **warninglog**. Clase que aloja la información del estado del sitio. Este estado es denominado *warning*, el que tiene asignado un `msg_id` único. Cada uno de estos mensajes tiene asociado un `DEFCON_LEVEL` que indica la gravedad de la situación, estos van desde nivel 0 que indica una operación normal y esperada, a un nivel 4 que indica el estado de desconexión del sitio.

3.2.1 Analytics

Actualmente, para realizar un análisis y conocer el estado de cada uno de los sensores y las estaciones, se ejecuta una serie de instrucciones que en LatinaUC son denominados Analytics. Estas llevan a cabo un análisis diario, en que se revisan los datos receptionados

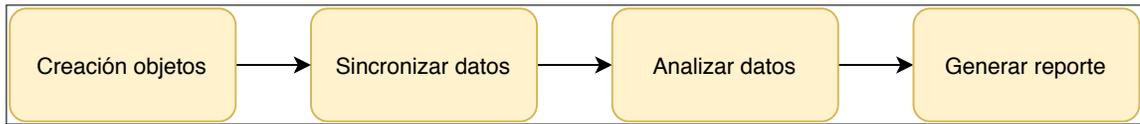


FIGURA 3.4. Flujo de la realización de Analytics. (Fuente: creación propia).

en las últimas 24 hrs., con el objetivo de generar un reporte que es enviado vía correo a los encargados de monitorear el estado de las estaciones.

La metodología para obtener un reporte del estado de un grupo, es separado en 4 fases principales, tal como se muestra en la figura 3.4.

El detalle de cada una se explica a continuación:

a) Creación de objetos

En esta fase de inicialización se crean los objetos necesarios para generar el reporte. En esta fase se realiza una conexión con SNAP para obtener la información de los grupos, sitios y variables, existe la opción de almacenar esta información en un archivo de extensión **pkl** para que en un futuro análisis no sea necesario volver a descargar la información. Además, para cada archivo se calcula su altura utilizando la API de Google Maps (*APIs de geolocalización — Google Maps Platform — Google Cloud (2020)*).

b) Sincronización de datos

El este paso se sincronizan los datos de cada una de las variables para realizar un futuro análisis. Para esto se itera sobre cada grupo de interés, luego sobre cada sitio, para, finalmente, realizar llamados individuales a la API de SNAP para cada variable. Con esto se crea un arreglo de datos, ordenados por la fecha de medición. Al igual que en el paso anterior, está la posibilidad de guardar en archivo de extensión **pkl** (archivos de tipo Pickle) para que en un futuro análisis no sea necesario volver a descargar los datos, si es que se desea volver a analizar este rango.

c) Analizar datos

Para cada variable se realiza un procesamiento para determinar su estado, a este se pueden agregar acciones específicas que dependen del tipo de variable, como lo es el caso de nivel de batería. Los siguientes son los tipos de análisis que se consideraron al momento de la realización de la tesis:

- Revisar la regularidad de los tiempos de muestreo.
- Analizar el retraso en los datos bajo un umbral definido.
- Obtener máximos, mínimos y promedio de los datos.
- Analizar que los rangos de batería se encuentren dentro de lo esperado.
- Identificar la ocurrencia de mensajes de diagnóstico, propios del sistema operativo LatinOS.

Luego de este análisis se crea un *warning* asociado al resultado. Este es guardado en la clase `warninglog` para su posterior uso en el reporte. Cada uno de estos tiene asociado un valor `DEFCON_LEVEL`, que es usado para la clasificación.

d) Generación de reporte

Posterior al análisis realizado, se genera un reporte ordenado por sitio y por nivel de *Defcon* que entrega el resultado de la estación. Este es un texto plano que tiene información estructurada para ser leída por una persona, un ejemplo de un reporte para la red de la Quebrada de Ramón se puede ver en la figura 3.5. Además, este reporte puede ser enviado vía correo a los responsables del monitoreo de dichas estaciones.

3.2.2 Flujo detallado de Analytics

En la figura 3.6 es posible observar con más detalle el flujo del programa que fue caracterizado en la figura 3.4. En este se observa el proceso para guardar los datos de cada una de las variables y cómo se realiza el análisis que fue explicado anteriormente.

analysis_2020-11-24-023902-UTC_report-for-group_QR.txt

REPORT BY DEFCON FOR GROUP 'QR'

| DEFCON | Site | Variable | Time | Warning |
|--------|---------------------------|--|---------------------|--|
| 4 | 10 Reforestacion | | 2020-11-24 02:38:56 | Site is offline. |
| 4 | 12 Estanque | | 2020-11-24 02:38:56 | Site is offline. |
| 4 | 15 QR 06 | | 2020-11-24 02:38:57 | Site is offline. |
| 4 | 17 QR 08 | | 2020-11-24 02:38:57 | Site is offline. |
| 4 | 18 Acumulador QR | | 2020-11-24 02:38:57 | Site is offline. |
| 4 | 24 Recinto Aguas Andinas | | 2020-11-24 02:38:57 | Site is offline. |
| 4 | 25 Repetidor Los Peumos | | 2020-11-24 02:38:57 | Site is offline. |
| 4 | 27 Confluencia | | 2020-11-24 02:38:58 | Site is offline. |
| 4 | 34 Tambor Bajo | | 2020-11-24 02:38:59 | Site is offline. |
| 4 | 36 Provincia Bajo | | 2020-11-24 02:38:59 | Site is offline. |
| 3 | 13 QR 04 | 44 Hops to sink | 2020-11-24 02:38:57 | Variable is offline (no data within the analysed period). |
| 3 | 13 QR 04 | 46 RSSI | 2020-11-24 02:38:57 | Variable is offline (no data within the analysed period). |
| 3 | 13 QR 04 | 45 Neighbour link quality | 2020-11-24 02:38:57 | Variable is offline (no data within the analysed period). |
| 3 | 13 QR 04 | 43 Neighbour | 2020-11-24 02:38:57 | Variable is offline (no data within the analysed period). |
| 3 | 35 Mirador Los Maquis | 3 Distance | 2020-11-24 02:38:59 | Variable is offline (no data within the analysed period). |
| 3 | 35 Mirador Los Maquis | 39 Snow depth | 2020-11-24 02:38:59 | Variable is offline (no data within the analysed period). |
| 2 | 14 QR 05 | 23 LatinOS diagnostic code | 2020-11-24 02:38:57 | Decagon 5TH sensor at 30 cm not responding (1 repetitions of LatinOS diagnostic code 21). |
| 2 | 28 Puente Los Peumos | 4 Battery voltage | 2020-11-24 02:38:58 | Battery reached critical low charge (3.58 V) during analysed period. |
| 2 | 35 Mirador Los Maquis | 23 LatinOS diagnostic code | 2020-11-24 02:38:59 | Maximally measurement error (144 repetitions of LatinOS diagnostic code 3). |
| 2 | 47 Repetidor Los Peumos 2 | 23 LatinOS diagnostic code | 2020-11-24 02:38:59 | Decagon 5TH sensor at 30 cm not responding (2 repetitions of LatinOS diagnostic code 21). |
| 1 | 13 QR 04 | 2 Relative air humidity | 2020-11-24 02:38:56 | Detected irregular sampling times (max gap = 20.0 minutes, min gap = 10.0 minutes, max/min = 2.0). |
| 1 | 13 QR 04 | 4 Battery voltage | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 20.0 minutes, min gap = 10.0 minutes, max/min = 2.0). |
| 1 | 14 QR 05 | 12 Apparent dielectric permittivity at 30 cm depth | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 20.0 minutes, min gap = 10.0 minutes, max/min = 2.0). |
| 1 | 16 QR 07 | 2 Relative air humidity | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 40.0 minutes, min gap = 10.0 minutes, max/min = 4.0). |
| 1 | 16 QR 07 | 4 Battery voltage | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 40.0 minutes, min gap = 10.0 minutes, max/min = 4.0). |
| 1 | 16 QR 07 | 1 Air temperature | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 40.0 minutes, min gap = 10.0 minutes, max/min = 4.0). |
| 1 | 16 QR 07 | 12 Apparent dielectric permittivity at 30 cm depth | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 30.0 minutes, min gap = 10.0 minutes, max/min = 3.0). |
| 1 | 16 QR 07 | 15 Soil temperature at 30 cm depth | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 30.0 minutes, min gap = 10.0 minutes, max/min = 3.0). |
| 1 | 26 Repetidor Confluencia | 12 Apparent dielectric permittivity at 30 cm depth | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 30.0 minutes, min gap = 10.0 minutes, max/min = 3.0). |
| 1 | 26 Repetidor Confluencia | 1 Air temperature | 2020-11-24 02:38:57 | Detected irregular sampling times (max gap = 30.0 minutes, min gap = 10.0 minutes, max/min = 3.0). |

FIGURA 3.5. Ejemplo de reporte generado.

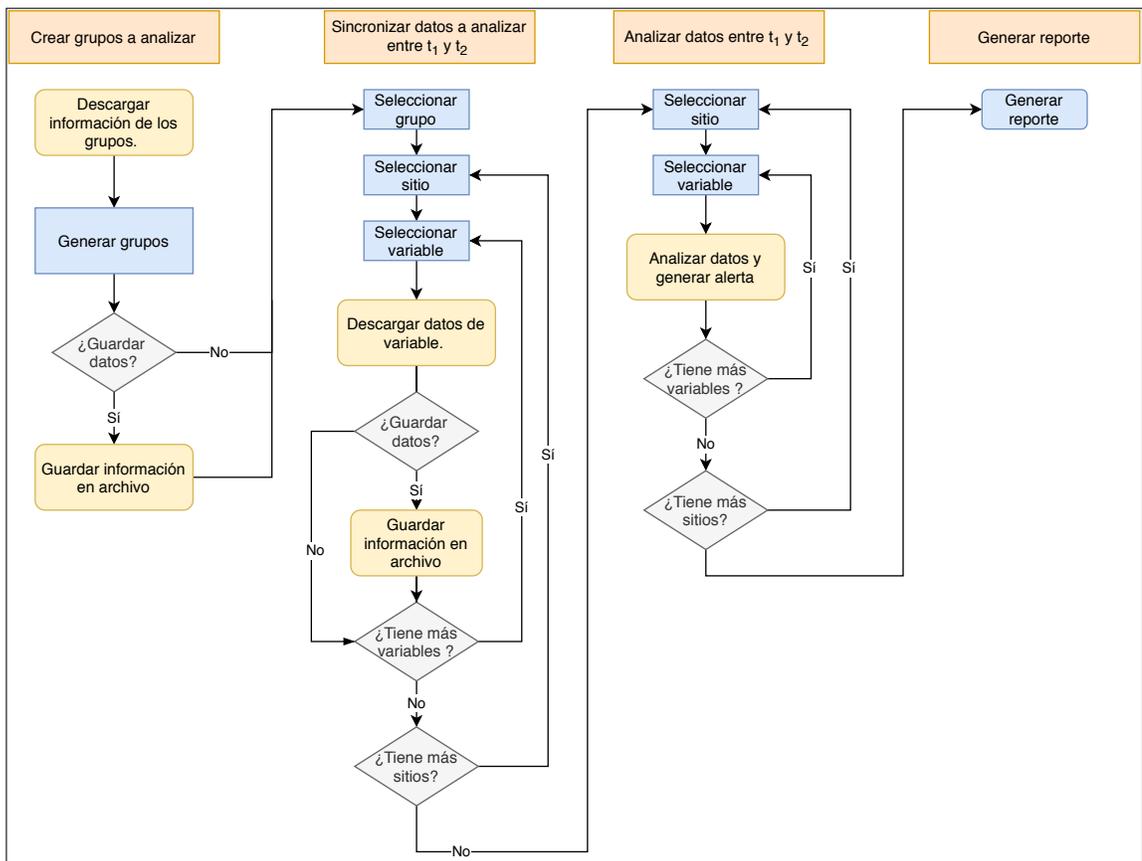


FIGURA 3.6. Flujo del programa para generar Analytics. Desarrollo propio.

3.3 Limitaciones existentes

La herramienta mostrada en la sección anterior permite realizar análisis y generar reportes para tiempos especificados. Sin embargo, posee una serie de limitaciones que impiden su crecimiento a largo plazo, lo que dificulta la capacidad de incorporar nuevos análisis. A continuación se explican las limitaciones detectadas.

a) Tiempo de ejecución muy lento

Uno de los principales problemas detectados, es que el tiempo requerido para realizar un análisis es muy alto. Este es causado porque los llamados a la API son realizados desde la clase `sitevar`, lo que se observa en el cuadro “Descargar datos de variable”, mostrado en la segunda etapa del flujo de la figura 3.6. La causa principal es la gran cantidad de solicitudes a la API de SNAP que se requieren para obtener los datos, debido a que se ingresa en un procedimiento cíclico por cada variable, estación y grupo.

b) Análisis ocurren solo en la ventana de tiempo sincronizada

Tal como se observa en la figura 3.6 los análisis están condicionados a la ventana de tiempo sincronizada, sea o no sea guardada localmente. Esto limita la incorporación de estudios que requieran de ventanas de tiempo distintas o, por ejemplo, de ventanas de tiempo móviles.

Un ejemplo de un análisis que requiere de una ventana de tiempo móvil (o sobrelapada), es determinar con una mayor precisión el momento en que una estación comienza a tener un retraso mayor. Una forma de realizarlo, sería ejecutando un análisis de las últimas 24 hrs., pero repitiendo el análisis a intervalos menores que 24 hrs. (por ejemplo, cada 20 min.), a diferencia de como se realiza actualmente, según lo indicado en la subsección 3.2.1.

c) Incapacidad de unir datos sincronizados localmente

Sumado a lo anterior, existe una incapacidad para unir los datos sincronizados localmente, debido a que los datos son guardados en formato **pkl** y este no permite una rápida

unión de su contenido. Tomando el mismo ejemplo de lo anterior, al realizar análisis con ventanas de tiempo sobrelapadas, lo correcto es sincronizar únicamente los datos faltantes y ejecutar el análisis entre el tiempo deseado, pero esto no es posible.

d) No es posible analizar una ventana de tiempo más pequeña

Este problema surge porque el tiempo sincronizado condiciona al tiempo de análisis. Solo es posible realizar análisis en toda la ventana de tiempo sincronizada, no pudiendo escoger una sección de los datos incorporados.

e) Incapacidad de incorporar nuevos tipos de análisis

Actualmente, si se desea crear un nuevo tipo de análisis, hay que realizar cambios importantes en código o utilizar un programa externo, ya que SNAPY no está preparado para recibir nuevas funciones sin requerir un cambio profundo. Esto limita la capacidad de su uso a medida que se despliegan nuevos grupos de estaciones o se desee utilizar los datos de forma eficiente.

f) Información generada por los Analytics no es sintetizada

Este problema recae en que los Analytics producen información cada 24 hrs, generando un reporte en texto plano para que una persona los pueda leer, lo que produce que el modo de almacenamiento no permita que una máquina los pueda utilizar, a menos que se realice un desarrollo de procesamiento del lenguaje natural, lo que escapa a las intenciones de LatinaUC.

Lo anterior provoca que no sea posible tener una visión global de la situación de un sensor o una estación, impidiendo reutilizar la información ya obtenida. Esta capacidad recae en la retención que logre una persona o en trabajar manualmente los distintos reportes y sintetizar cada vez que se requiera. Esto produce un problema de escalabilidad, ya que al aumentar la cantidad de estaciones y variables, el problema se torna excesivamente complejo de abordar.

4. METODOLOGÍA DEL TRABAJO

En este capítulo se presentará el desarrollo de la plataforma propuesta, comenzando con la justificación de la forma de abordar los problemas encontrados, para concluir con una explicación de los posibles usos que esto traerá a LatinaUC.

4.1 Solución propuesta

Tal como se mencionó en la sección anterior, el sistema con que el que LatinaUC contaba al momento de la realización de esta tesis tenía una serie de limitaciones e inconvenientes que producían dificultades para aspirar al objetivo a largo plazo.

Para poder guiar el camino a este objetivo se comenzó con la realización de una plataforma de gestión de estado de estaciones, construyéndose a partir de lo ya desarrollado en LatinaUC. Para esto se implementaron una serie de cambios que buscan mejorar las capacidades existentes.

4.1.1 Reestructuración SNAPY e incorporación de base de datos local

Para perfeccionar la forma de solicitar los datos a la plataforma SNAP, se realizó una reestructuración del código de SNAPY. Esta reestructuración solucionó los inconvenientes que se producían al solo poder actualizar los datos de una variable, lo que causa la lentitud en la ejecución de los códigos. Junto con esto se le añadió al sistema la capacidad de trabajar con una base de datos local, la que permite guardar los datos para su procesamiento y posterior uso.

La base de datos creada está realizada bajo MongoDB, que corresponde a un tipo de base de datos no relacional, en donde cada dato es guardado como documentos dentro de colecciones (*The most popular database for modern apps* (s.f.)). El uso de este tipo de base de datos ha aumentado en el último tiempo, principalmente en aplicaciones que generan datos heterogéneos, como lo que sucede en sistemas que utilizan redes de sensores (Kang, Park, Rhee, y Lee (2016)). Diversos estudios validan el uso de bases de datos

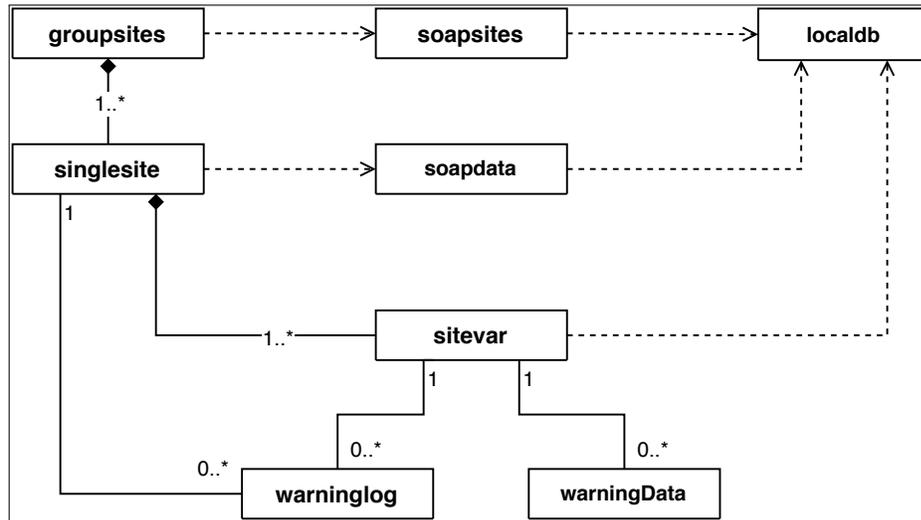


FIGURA 4.1. Diagrama de clases de versión nueva de SNAPY.

no relacionales en aplicaciones de IoT (Rautmare y Bhalerao (2016); Atat et al. (2018); Eyada, Saber, El Genidy, y Amer (2020)).

La elección de este tipo de base de datos se justifica debido a la alta versatilidad y escalabilidad horizontal que posee, además de no depender de un esquema predefinido como sucede en base de datos relacionales. Esto permite que el sistema se pueda ir actualizando y evolucionando a futuro y poder tener de forma unificada la información de los datos generados por análisis, sin tener que necesariamente cambiar la estructura de la base de datos.

En cuanto a la reestructuración del código de la nueva versión de SNAPY se presenta un diagrama de clases reducido en la Figura 4.1, esto es, sin incluir los atributos y funciones para focalizarse en las conexiones entre las clases; una visualización completa puede encontrarse en el Anexo A.2. En este es posible observar que se incorporó la clase `soapData`, que se encarga de comunicarse con SNAP; además de que distintas clases tienen acceso a la base de datos local a través de `localsdb`. Esto permite tener un procesamiento unificado que se encarga de gestionar las distintas solicitudes que se realicen a la base de datos.

TABLA 4.1. Contenidos `warningData`.

| | |
|----------------------------|---|
| <code>msgID</code> | identificador único de cada tipo de alerta |
| <code>siteID</code> | ID del sitio al que pertenece la alerta |
| <code>varID</code> | ID de la variable a la que pertenece la alerta |
| <code>dtFrom</code> | fecha de origen de la alerta |
| <code>dtTo</code> | fecha de término de la alerta |
| <code>analyticsDate</code> | fecha en la que se generó la alerta |
| <code>info</code> | información extra generada por la última alerta |

Además, se incorporó la clase `warningData`, que se explicará en la siguiente subsección.

4.1.2 Gestión de analíticas

Esta nueva incorporación viene a solucionar la incapacidad de gestionar analíticas generadas en intervalos de tiempo y momentos distintos. En la versión anterior, no era posible reutilizar los resultados obtenidos para hacer nuevos análisis, como por ejemplo conocer la tendencias de estados o conocer la duración que una alerta se ha mantenido en una estación.

En esta línea de desarrollo, junto con la reestructuración del sistema de SNAPY, se incorporó la capacidad de llevar registro de los estados de los distintos sensores y estaciones que se generan, mediante los distintos análisis que se llevan a cabo. Este registro es guardado en la base de datos, solo que en una colección dedicada para almacenar esta información.

Para realizar esta gestión, se incorporó la clase `warningData`, la que aloja la información necesaria para crear una alerta. Esta incluye los parámetros indicados en la tabla 4.1.

Con esta información es posible guardar la alerta generada por un proceso de análisis. Por su parte, cada `msgID` tiene asociado un mensaje fijo y un nivel de alerta denominado `DEFCON_LEVEL`, este tiene un valor discreto entre 0 a 4, en donde 0 indica funcionamiento normal y 4 indica desconexión total de la estación. Además, se incluye el parámetro `info`, que tiene información extra generada por la última alerta de este objeto, la que

TABLA 4.2. Tabla de alertas existentes.

| MSG ID | DEFCON | Mensaje |
|--------|--------|---|
| 1 | 1 | Variable's data is not fresh. Refresh before analysing. |
| 2 | 3 | Variable is offline. |
| 3 | 1 | Irregular sampling period. |
| 4 | 2 | Unknown exit status. Can't analyse. |
| 5 | 1 | Unable to analyse: no method available. |
| 6 | 2 | Battery reached critical low charge. |
| 7 | 0 | Battery reached full charge. |
| 8 | 1 | Record has duplicate samples. |
| 9 | 0 | Variable looks ok. |
| 10 | 1 | Site is overdue for a new post. |
| 11 | 4 | Site is offline. |
| 12 | 0 | No diagnostic codes were reported during the analysed period. |
| 13 | 2 | Site reported diagnostic codes. |
| 14 | 2 | Runtime exception. |
| 15 | 2 | sitevar.minRow(): Data set is all NaN. |
| 16 | 2 | sitevar.maxRow(): Data set is all NaN. |

puede incluir tiempos de muestreo, ID de un dato en particular, tiempo desde el último dato recibido, entre otros. La lista de alertas actualmente presentes se indica en la tabla 4.2.

4.1.3 CRUD de alertas

Para poder utilizar y analizar los datos de alertas generados se implementó un procedimiento, el que permite realizar operaciones del tipo **CRUD** sobre la base de datos que se ha generado, es decir:

Create : permite la creación de alertas con la información indicada en la subsección anterior. Cada alerta tiene un identificador único generado por MongoDB.

Read : permite leer la información de las alertas que se encuentran alojadas en la base de datos. Se incluye la capacidad de filtrado por cada una de las variables de interés, dependiendo de la aplicación.

Update: permite actualizar la información de las alertas, fusionando aquellas que ocurren en periodos de tiempo sobrelapados y que corresponden al mismo sitio,

variable y mensaje. Esto da la posibilidad de conocer los periodos en los que una alerta se ha mantenido activa en una variable para alguna estación en particular.

Delete : permite eliminar alertas cuya información ya no es relevante para algún estudio.

Con estas operaciones es posible tener un registro e ir actualizando la información de las alertas, a medida que se realicen nuevos análisis cuyos resultados estén relacionados con los ya existentes. Esta capacidad permite analizar sobre ventanas de tiempo sobrelapadas, logrando contar con un registro individualizado por cada estación, variable y tipo de alerta, junto con la duración que ésta ha estado presente.

Un ejemplo de uso en donde ocurrirá que las ventanas se deben sobrelapar, es cuando se desea conocer con mayor precisión el estado de una estación. Actualmente, los análisis se ejecutan en ventanas de 24 hrs. y solo una vez al día, de esta forma el siguiente análisis ocurrirá 24 hrs. después y así sucesivamente. Esto produce que el horario inicial de una alerta sea inmediatamente el término de una alerta anterior y no hay forma clara de obtener alertas que se fusionen a medida que entregan la misma información.

Sin embargo, con la incorporación de este nuevo sistema, es posible realizar los mismos análisis de las últimas 24 hrs., pero repitiéndolos cada 1 hr. u otra ventana de tiempo a elección. Ya que el resultado obtenido irá actualizando la fecha de inicio y término de las alertas al fusionarlas o creará una nueva, si es que la alerta entrega un mensaje distinto. Esto da una mayor flexibilidad a la hora de tener un registro de la información.

4.1.4 Escalabilidad del sistema

La nueva estructura del programa, en conjunto con la incorporación de la base de datos y la gestión de alertas, permite tener un código escalable. Esto debido a que el sistema está estructurado para recibir nuevos tipos de alertas (que sigan el formato entregado) o la incorporación de una forma distinta de generar reportes, que utilicen la información que las alertas fusionadas entreguen. Esta capacidad sigue en la línea de investigación de LatinaUC, ya que a medida que se desarrollan nuevos proyectos y se incorporan nuevas

estaciones con diversos sensores, es importante poder contar con un sistema que pueda ser eficaz a la hora de gestionar e informar el estado de las estaciones.

5. ANÁLISIS Y RESULTADOS

Este capítulo muestra los resultados que se pueden lograr con el uso de la plataforma desarrollada, para esto se utilizaron los datos de las estaciones de la Quebrada de Ramón, en el periodo que va desde el 1 de enero de 2018 hasta el 31 de diciembre de 2019. La decisión de este periodo se basa en que hay conocimiento de que algunas de las estaciones sufrieron fallas que alteraron su comportamiento. En algunos casos ocurrieron antes de este periodo y las estaciones fueron puestas en línea nuevamente, y otras ocurrieron durante estas fechas.

5.1 Ejemplos prácticos de la aplicación de la plataforma en estaciones de alta montaña

En esta sección, se mostrarán ejemplos prácticos del uso de la plataforma para analizar los estados históricos de las estaciones, con datos reales obtenidos de aquellas que se encontraban desplegadas en la alta montaña de la Quebrada de Ramón. Solo se consideraron los datos enviados por las estaciones a SNAP y no aquellos que son guardados en las respectivas tarjetas SD.

5.1.1 Estado de carga y distancia. Caso estación Mirador Los Maquis

Como antecedentes la estación Mirador Los Maquis, se encuentra ubicada a 2972 m.s.n.m. siendo la estación que se ubica a una mayor altura. Por su localización, esta estación está equipada con un sensor de distancia, para poder medir altura de nieve, y con pluviómetro, para medir agua caída. Esto debido a que en la mayor cantidad de tiempo, solo recibe nieve, ya que la isoterma cero promedio de la zona central es de 2200 m.s.n.m. (Garreaud (2013)), pero hay momentos en que la isoterma cero sube de esta altura y la estación puede recibir agua.

Esta estación se encontraba *offline* desde finales de 2017, por lo que, en una campaña de mantenimiento de abril de 2018, se volvió a poner en línea. Mediante la plataforma realizada, se analizó la situación de esta estación en todo el año 2018 y 2019, realizando

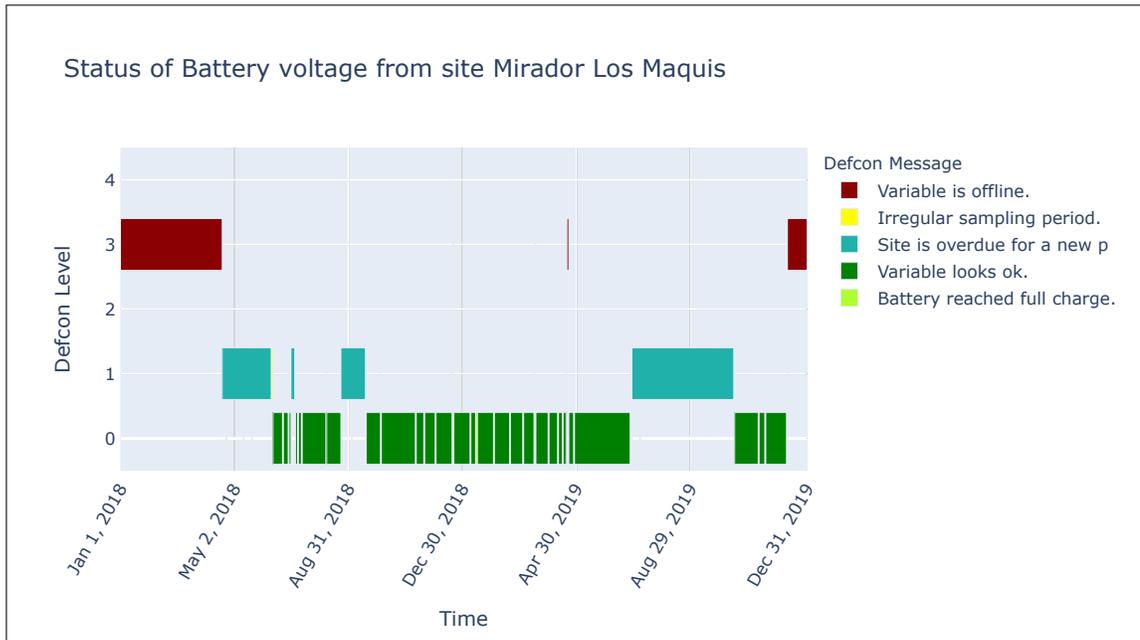


FIGURA 5.1. Estado de operación de batería en estación Mirador Los Maquis.

analíticas cada 24 hrs., las que fueron guardadas y fusionadas, siguiendo lo explicado en la sección anterior. Al graficar la información generada del análisis de batería se obtiene la Figura 5.1.

En este gráfico, el eje vertical representa el DEFCON_LEVEL del estado detectado y en el eje horizontal la fecha en la que esa alerta estuvo activa. En la Figura 5.1 se puede observar que a principios del año 2018 la estación se encontraba desconectada y que en abril de ese año comenzó a volver a estar en línea, lo que se ve representado en el primer bloque celeste, que si bien indica que algunos datos llegaron atrasados, la estación estaba transmitiendo.

Es posible observar que la batería se consideraba en un estado normal durante gran parte del año 2018 y 2019, solo que hasta finales de ese año, la estación nuevamente volvió a reportarse como desconectada. Si esta información es contrastada con las alertas del sensor de distancia en este mismo periodo se puede observar una diferencia en los resultados. El gráfico correspondiente es el de la Figura 5.2.

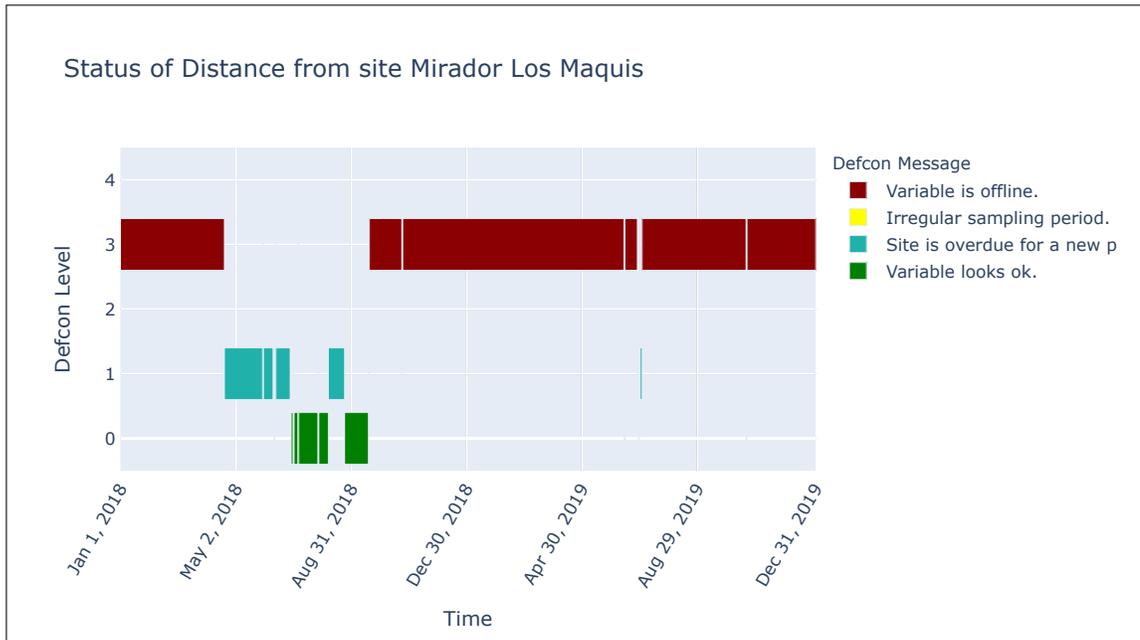


FIGURA 5.2. Estado de operación de sensor de distancia en estación Mirador Los Maquis.

Aquí es posible observar que la estación estaba entregando información de distancia durante el periodo invernal de 2018, que cubre los meses de mayo a agosto. Pero, aproximadamente en septiembre del mismo año, la estación dejó de entregar información de su sensor de distancia, a excepción de algunos breves periodos durante el año 2019. Esto puede indicar que la estación estaba teniendo un problema en sus mediciones, que impedía que el sensor de distancia realizara una medición correcta.

Con solo esta información, se puede llegar a la hipótesis que la estación sufrió un problema estructural, posiblemente causado por las condiciones ambientales a las que se encuentra, como lo pueden ser la acumulación excesiva de nieve o los fuertes vientos de la zona, afectando principalmente al sensor de distancia. Esta conclusión pudo ser confirmada debido a que el día 30 de noviembre de 2020 se visitó la estación para su retiro y se pudo comprobar que tenía daños estructurales. Esto se puede observar en la Figura 5.3 que muestra la estación instalada luego de la mantención realizada y en la Figura 5.4 que muestra el estado de la estación al momento del retiro.



FIGURA 5.3. Estación Mirador Los Maquis en la mantención realizada en abril de 2018. Foto cortesía LatinaUC.



FIGURA 5.4. Estación Mirador Los Maquis en el retiro realizado en noviembre de 2020. Foto cortesía LatinaUC.

Al ver en detalle la imagen de la Figura 5.4, se observa que el panel solar se encontraba apuntando hacia el cielo, lo que permitía que la estación pudiera seguir cargándose. Sin embargo, el sensor de distancia se encontraba apuntando hacia el suelo y por las características indicadas por el fabricante (*MB7574 SCXL-MaxSonar-WRST7 (2020)*), este sensor no mide a distancias menores a 50 cm, lo que justifica lo observado en los gráficos realizados. En este caso, la falla cae en la categoría de falla de componente, según lo indicado en la subsección 2.2.1, debido a que algunos componentes siguen funcionando a pesar de que otros están presentando problemas.

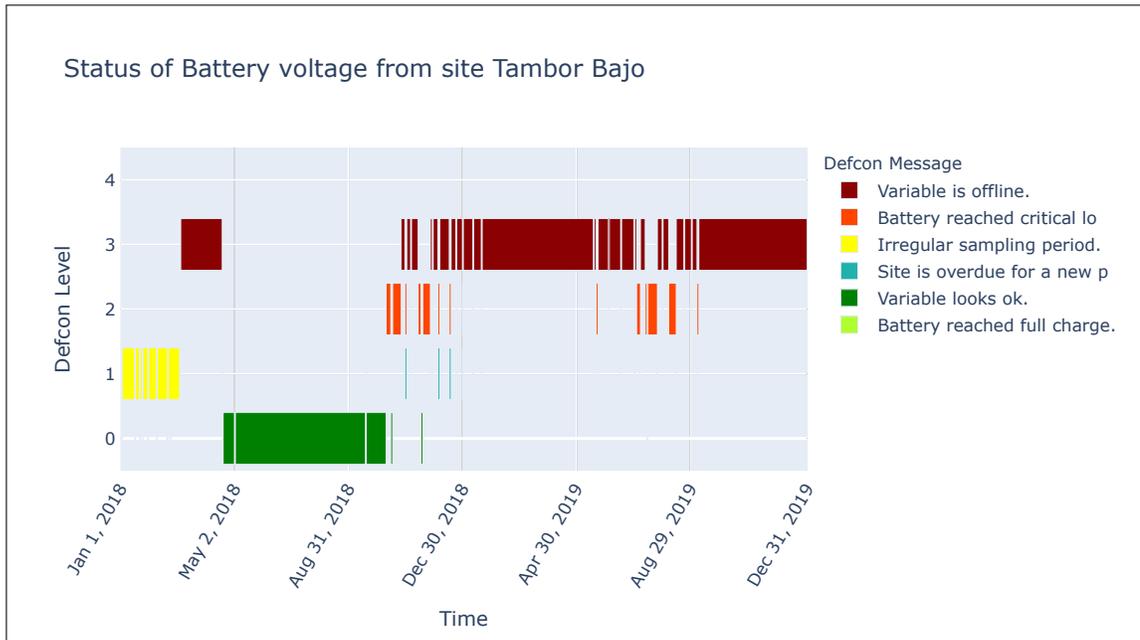


FIGURA 5.5. Estado de operación de batería en estación Tambor Bajo.

5.1.2 Estado de sistema de carga. Caso estación Tambor Bajo

Como antecedente la estación Tambor Bajo, se encuentra a una altura de 2654 m.s.n.m. siendo la segunda estación ubicada a mayor altura. Esta estación cuenta con los mismos sensores que la estación Mirador Los Maquis y previo al 2018 se había detectado un problema con el sensor de temperatura y humedad del aire, lo que había producido que la estación estuviese desconectada durante marzo de 2018. Este sensor fue reemplazado en el plan de mantenimiento realizado en abril de 2018, solucionando ese problema.

En la Figura 5.5 se puede observar un caso distinto en que los análisis de la batería reportaron distintos escenarios. En un comienzo es posible observar la situación descrita en el párrafo anterior, seguida por un periodo de funcionamiento normal hasta aproximadamente septiembre del año 2018. En este punto se puede ver que la estación comenzó a tener una baja de su batería y, además, periodos de desconexión total.

Estos cambios se mantuvieron durante gran parte de 2019, en que la estación no volvió a una situación operativa normal. Con la información disponible es posible inferir que la

estación tenía un problema en su modulo de carga, ya que no era capaz de llegar a niveles estables, por lo que otros sensores no podían funcionar de forma normal. Este tipo de falla cae en la categoría de falla de nodo, según lo indicado en la subsección 2.2.1, debido a que al fallar la alimentación la estación completa no puede funcionar.

5.2 Ejemplos prácticos de la aplicación de la plataforma en estaciones en red

En esta sección se mostrarán ejemplos prácticos del uso de la plataforma para analizar el estado de estaciones que se encuentran conectadas en red.

5.2.1 Estado de sistema de carga. Caso estación Diego

La estación Diego se ubica a 1007 m.s.n.m., siendo la estación que se encuentra a menor altura que aún está en terreno. Esta estación tiene la característica de servir como el nodo acumulador de red, ya que los datos adquiridos por las estaciones llegan a esta para poder ser subidos a la nube.

Esta estación, como se indica en la Tabla 3.1, solo posee un sensor de temperatura y humedad del aire. Por lo que para conocer el estado de operación, se realizó un análisis sobre su batería en el periodo indicado al comienzo. Los resultados se pueden observar en la Figura 5.6.

En este, a diferencia de como ocurría en otras estaciones, es posible notar que la variable se mantuvo en un funcionamiento normal durante todo 2018 y 2019. Si bien hay periodos de muestreo irregular, estos se justifican debido a que en este periodo la estación fue visitada en varias ocasiones para realizar una descarga de sus datos y se actualizó en *firmware*, lo que produce reinicios del sistema operativo que alteran levemente el funcionamiento, tal como se muestra en la Figura 5.7. Sin embargo, para efectos prácticos, se considera como un ejemplo de una estación que ha tenido un buen comportamiento.

5.2.2 Estado de estación. Caso Alto Las Vizcachas

La estación Alto Las Vizcachas se encuentra 1768 m.s.n.m., emplazada en la ladera del cerro Alto Las Vizcachas en la Quebrada de Ramón. Esta estación posee un sensor

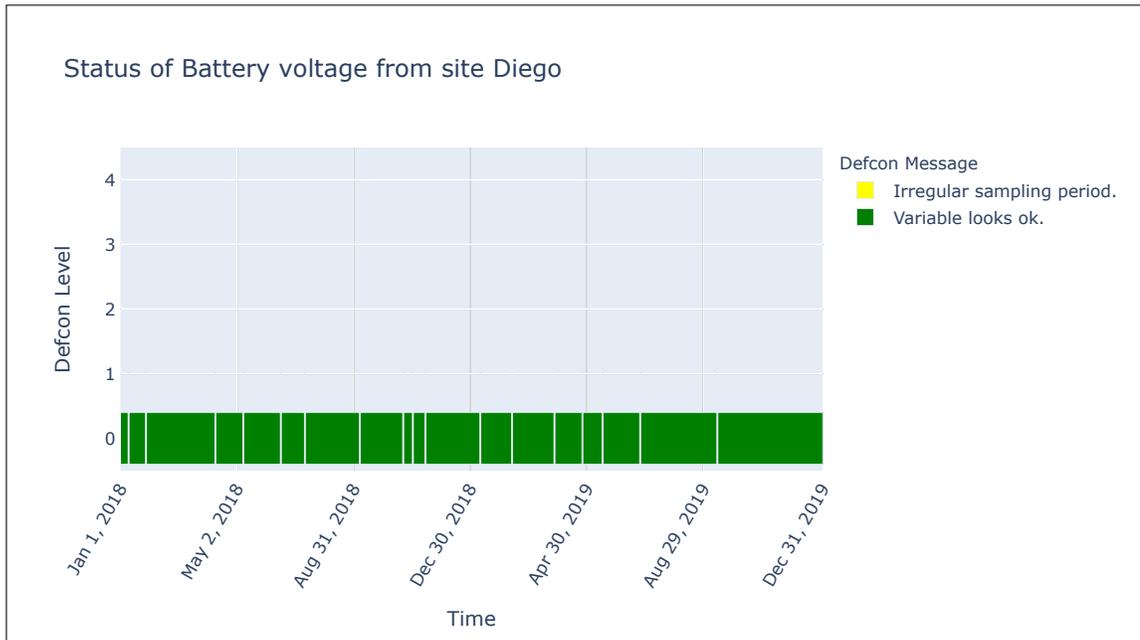


FIGURA 5.6. Operación normal de estación Diego.

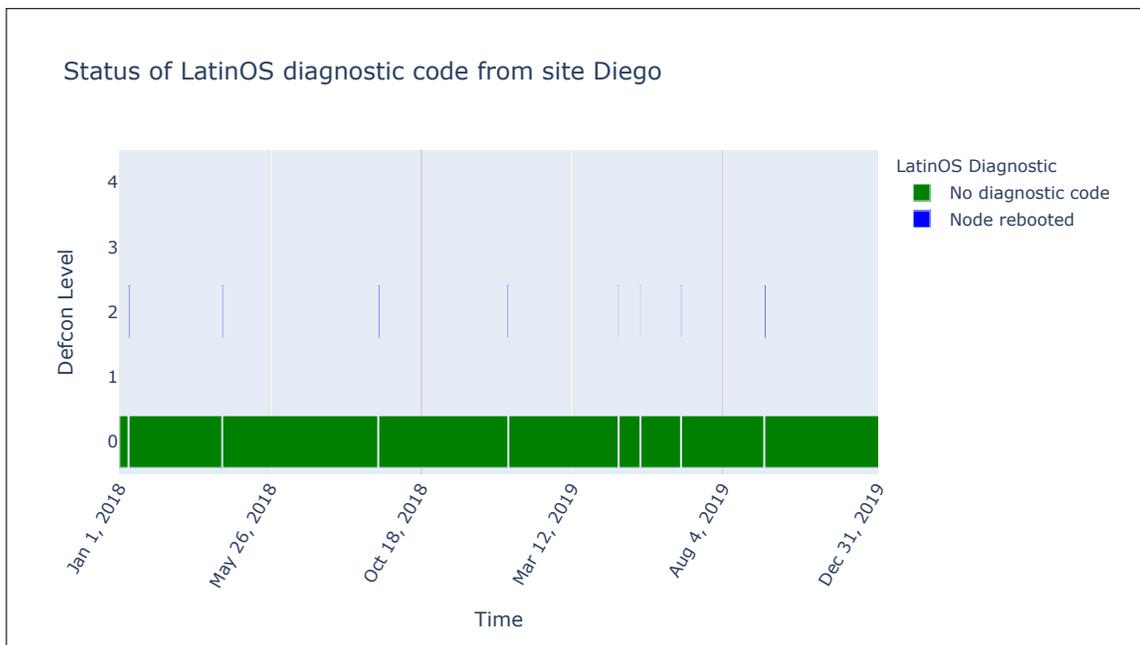


FIGURA 5.7. Códigos de diagnóstico en estación Diego.

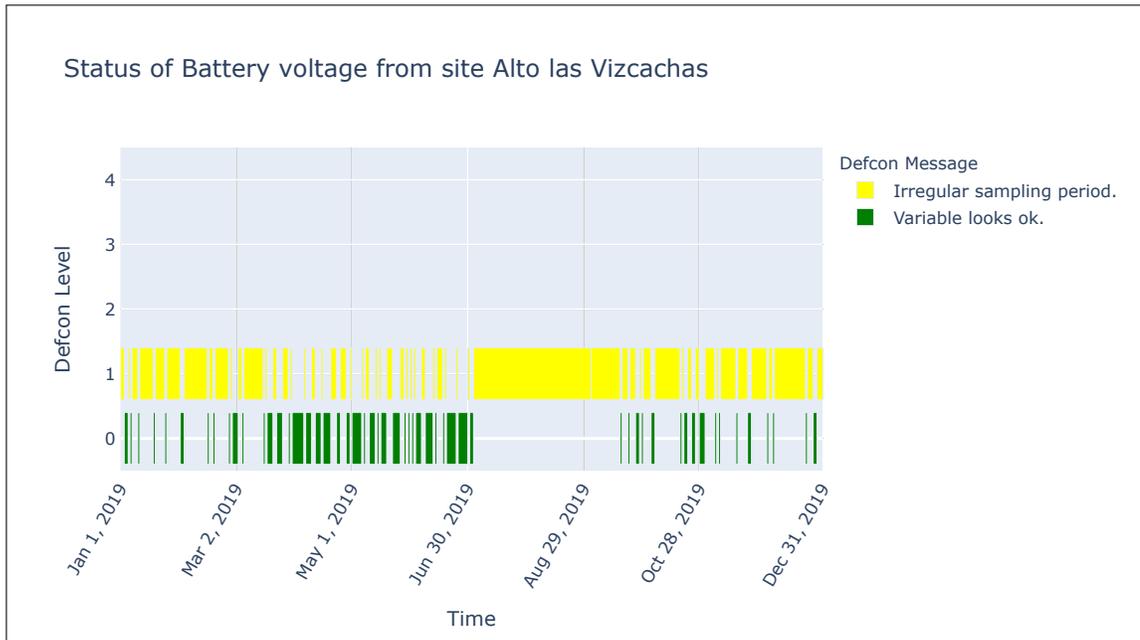


FIGURA 5.8. Operación irregular de estación Alto Las Vizcachas.

de humedad de suelo junto con un sensor que mide temperatura y humedad del aire. Para evaluar su comportamiento se analizaron los datos durante el año 2019. Los resultados se pueden ver en la Figura 5.8.

Como se puede observar esta estación presenta una gran cantidad de alertas que indican que está ocurriendo un muestreo irregular. A diferencia de lo que sucedía con la estación Diego, esta no había sido visitada regularmente para que actualizaciones del *firmware* justifiquen esta situación. Para poder determinar la causa, se puede contrastar esta información con la ocurrencia de códigos de diagnóstico mostrado en la Figura 5.9. Aquí es posible observar una alta ocurrencia de problemas del sensor de humedad y temperatura de suelo, el que es un Decagon 5TM (*Decagon 5TM VWC + Temp (2014)*).

Esta situación puede producir problemas en el sistema operativo de la estación, ya que al encontrarse con un sensor con fallas intermitentes, otros procesos, necesarios para lograr el envío de los datos medidos, pueden verse dañados por esta situación. Esta situación podría producir una falla a nivel de red causando problemas de congestión, como fue mencionado en la subsección 2.2.1.

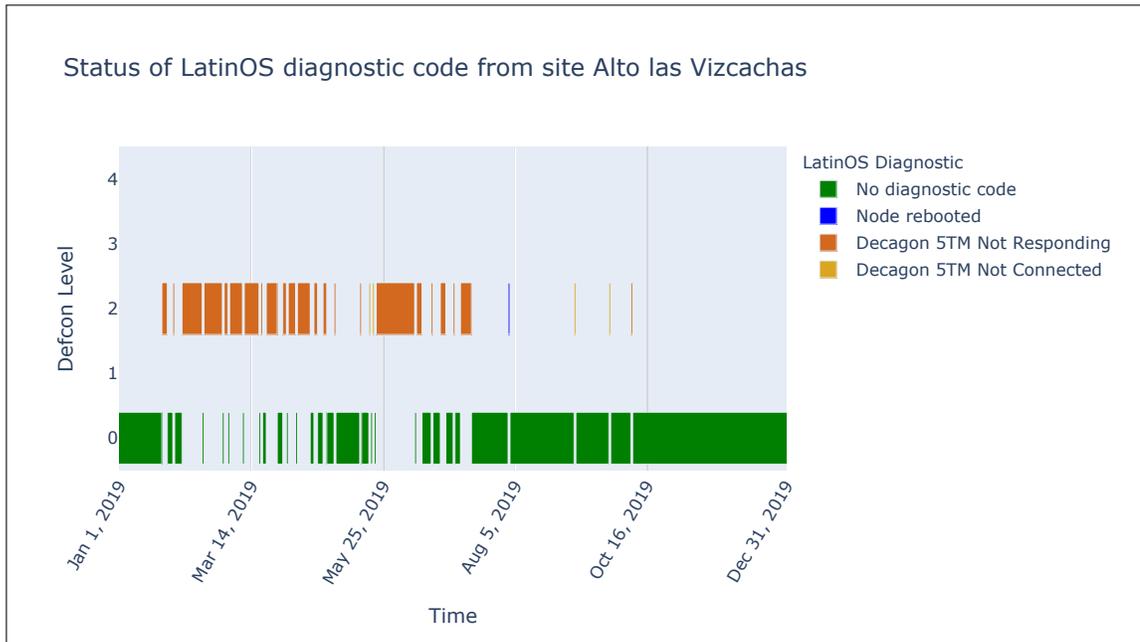


FIGURA 5.9. Códigos de diagnóstico en estación Alto Las Vizcachas.

5.2.3 Caso estación Repetidor Confluencia

La estación Repetidor Confluencia, se encuentra ubicada a 1345 m.s.n.m. y es una de las estaciones que se comunica con la estación Alto Las Vizcachas. Esta comunicación puede observarse en la topología mostrada en la Figura 5.10. Aunque también hay comunicación con la estación QR07 solo se analizará la estación Repetidor Confluencia.

Para analizar su comportamiento se analizó la batería, obteniendo lo mostrado en la Figura 5.11. Es posible notar que la estación se encuentra en una situación constante de muestreos irregulares, que al igual que en la estación Alto Las Vizcachas, podría estar causado por constantes problemas con el sensor de temperatura y humedad de suelo. Lo anterior, debido a que al analizar los códigos de diagnóstico del sistema operativo mostrados en la Figura 5.12, indican un funcionamiento incorrecto.

Aquí es posible notar que hay una mayor ocurrencia de códigos de diagnóstico con el sensor de humedad y temperatura de suelo. Este tipo de situación podría causar además

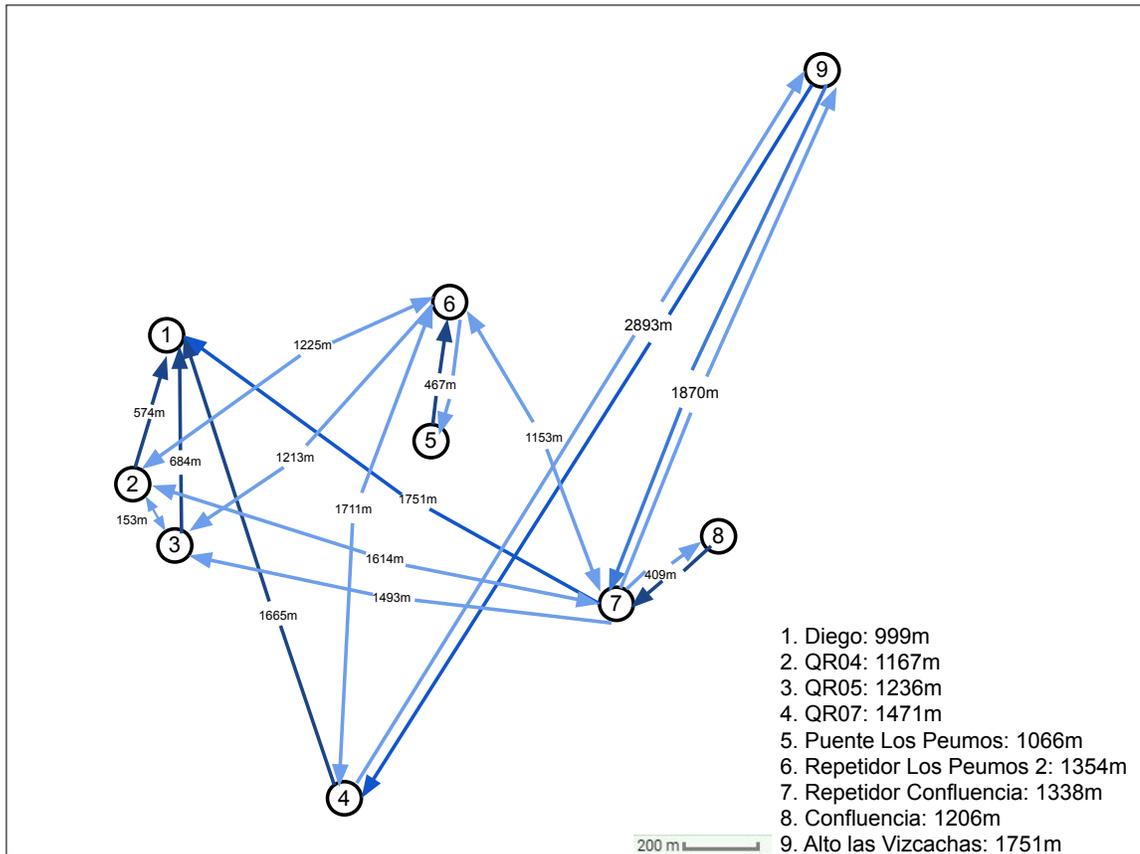


FIGURA 5.10. Topología de la red de la Quebrada de Ramón. Los colores más oscuros indican una mayor conexión con dicho vecino. Figura obtenida de Guerrero (2020).

problemas en la comunicación de la estación con otras estaciones, ya que al verse afectado el procesamiento del sistema operativo, la estación podría perder sincronismo con la red.

Al encontrarse en red los problemas de sincronismo son graves para mantener un buen funcionamiento, y así poder dar garantías de resiliencia a un grupo de estaciones. Esto, debido al hecho que la ocurrencia de problemas de comunicación con una estación puede hacer que los datos medidos no lleguen a destino en el tiempo esperado, ya que la ruta que deben seguir no está operando de forma correcta. Un ejemplo de esta situación es lo que se observó con las estaciones Alto Las Vizcachas y Repetidor Confluencia, en que la falla de un sensor puede producir problemas que afectan a otros sensores.

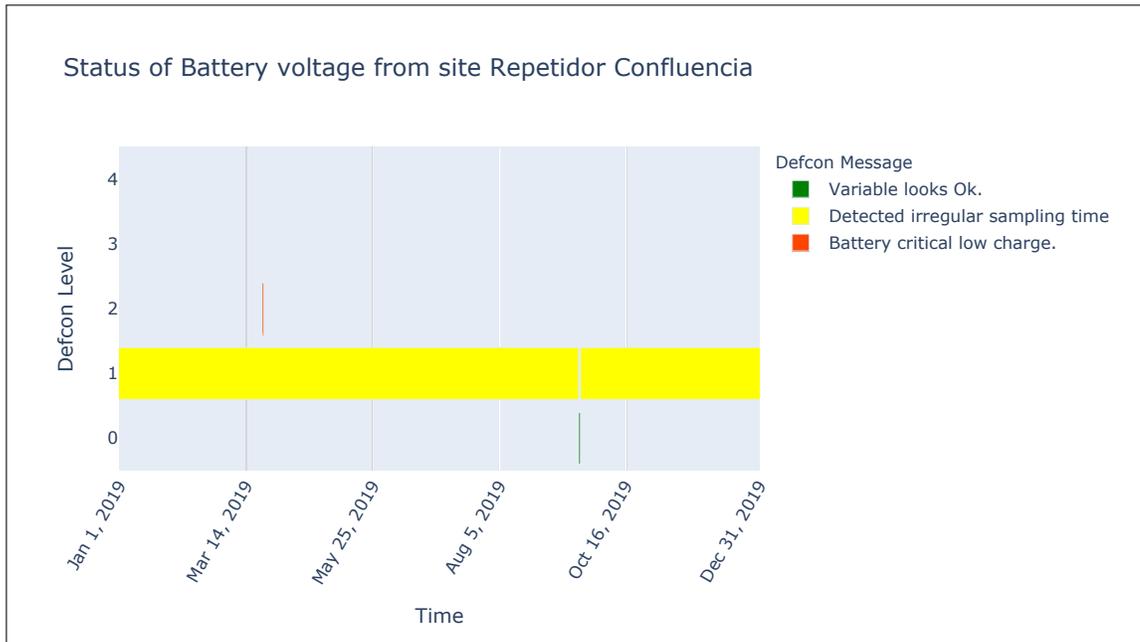


FIGURA 5.11. Estado irregular en estación Repetidor Confluencia.

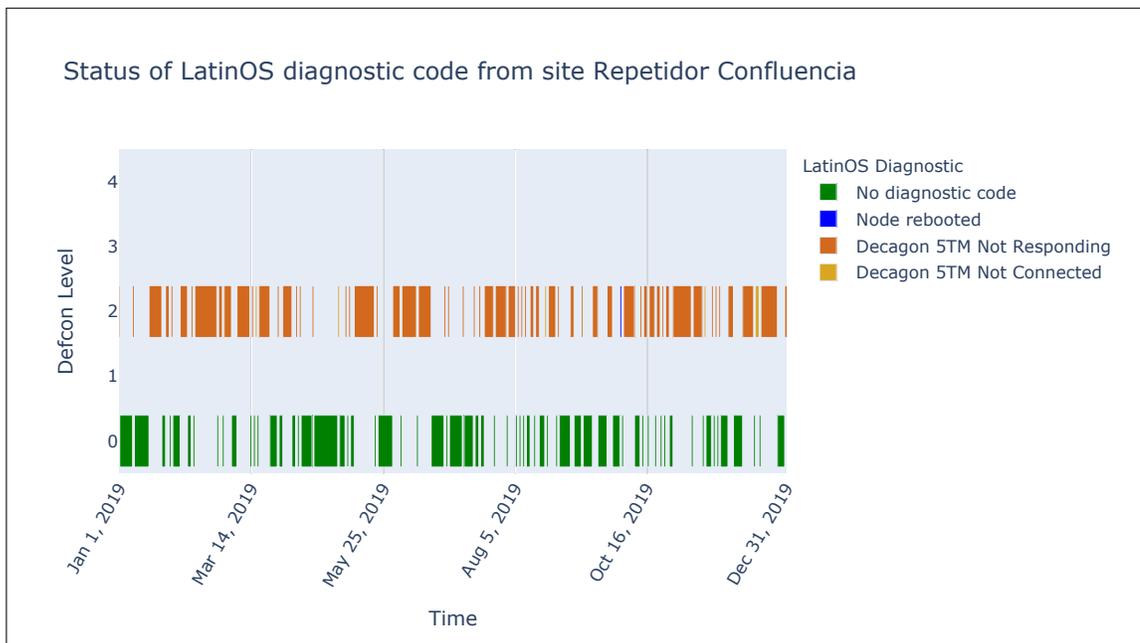


FIGURA 5.12. Códigos de diagnóstico en estación Repetidor Confluencia.

Este tipo de situación es un buen ejemplo en que es necesario conocer el estado de una estación en periodo de tiempos mayores, para así determinar qué está causando problemas a la red y poder tomar acciones para solucionar la situación.

6. CONCLUSIONES, DISCUSIONES Y TRABAJO FUTURO

6.1 Conclusiones de los resultados obtenidos

Los ejemplos mencionados en la sección anterior muestran un uso que se le puede dar a la plataforma de gestión de estado de estaciones, tanto para aquellas estaciones que funcionan de forma independiente como las que se encuentran interconectadas en red.

En el caso de las situaciones mostradas con las estaciones que se encuentran en alta montaña y funcionan de forma independiente, es posible notar que, mediante una visualización de su estado en una ventana de tiempo mucho mayor que analíticas diarias, se puede generar una hipótesis de lo que puede haber sucedido al detectarse un comportamiento anormal.

Por otra parte, en estaciones que han tenido un comportamiento esperado, se puede visualizar el tiempo en que ha estado operativa y de esta forma poder validar su funcionamiento. Un ejemplo de esto es la estación Diego, la que presentó un comportamiento sin problemas durante todo el periodo analizado, esto se conoce como el *uptime* o tiempo en línea que un sistema ha estado funcionando.

Sumado a lo anterior, en aquellas estaciones que están teniendo un comportamiento no esperado, como lo es la estación Alto Las Vizcachas, se puede visualizar de mejor forma las posibles causas de esta situación. En este caso la aparición de un sensor defectuoso puede producir inconvenientes a la hora de estar en sincronía con la red y de medir valores de otros sensores.

Esta situación se podría ver acrecentada si es que la estación con la que se debe comunicar mantiene un problema similar, dificultando la comunicación entre las estaciones al no estar sincronizadas, lo que, es algo crucial para el sistema operativo LatinOS (Barros (2013)). Esta falta de sincronismo en las estaciones puede traer que los datos no lleguen en los tiempos esperados, lo que dependiendo de la aplicación, podría traer problemas para validar la robustez y resiliencia de la red (Guerrero (2020)).

6.2 Aporte y discusiones

Al comienzo de este texto se mencionó que el sistema con el que contaba LatinaUC al momento del desarrollo de la tesis poseía una serie de limitaciones. Estas fueron detectadas y solucionadas, lo que fue discutido y mostrado en el capítulo 4. En particular, la plataforma desarrollada y el aporte logrado en esta tesis se puede resumir en los siguientes puntos:

1. La plataforma desarrollada optimizó el procesamiento y adquisición de los datos mediante la incorporación de una base de datos local. Esta mejora permite agregar la capacidad de unir la información localmente y realizar análisis en ventanas de tiempo no necesariamente condicionadas a los que fue sincronizado en dicha ejecución, debido a que ahora son procesos independientes.
2. Mediante los casos analizados, fue posible observar estados de fallas a las que las estaciones de la Quebrada de Ramón se han visto expuestas. Realizando un análisis de su comportamiento y relacionando la información adquirida, se pudo llegar a las causas que influyeron en su funcionamiento, permitiendo validar el uso de la plataforma mediante situaciones reales en las que ocurrieron fallas.
3. Se incorporó la capacidad de tener un registro de las analíticas realizadas, permitiendo visualizar el comportamiento global de una estación y no solamente mediante analíticas diarias.
4. La reestructuración de la plataforma *SNAPY* le entrega a LatinaUC un nuevo set de herramientas que permite la incorporación de nuevos tipos de analíticas a futuro.

Recordando la visión de largo plazo de LatinaUC, es decir, poder dar garantías de que las redes de sensores desarrolladas sean resilientes, la plataforma desarrollada sienta las bases para llegar hasta este objetivo. La razón es que ahora es posible incorporar analíticas para continuar con el desarrollo de los futuros proyectos de LatinaUC al incorporar este conocimiento a nuevas versiones.

Estas nuevas capacidades, que es posible incorporar a los futuros desarrollos de LatinaUC, no solo se conectan con la posibilidad de agregar análisis a los datos, sino que puede convertirse en una herramienta de desarrollo que permita un mejor uso de la información generada a potenciales clientes y proyectos que sean desarrollados.

Lo anterior, debido a las nuevas funcionalidades, tiene como objetivo alinearse a un marco de continuo desarrollo y mejoras futuras, que no solo sean implementados en aplicaciones de redes para hidrometeorología, sino que para cualquier proyecto de monitoreo remoto y que haga uso de la Internet de las Cosas.

6.3 Trabajo futuro

En esta tesis se sentaron las bases necesarias para proseguir con estudios futuros de LatinaUC que busquen lograr redes de sensores inalámbricas resilientes, ante distintas adversidades que se puedan encontrar. Esta plataforma forma las bases necesarias para el camino que se quiere lograr a largo plazo, es por esto que a partir de aquí surgen diversos trabajos futuros que permiten continuar con el desarrollo de SNAPY como una plataforma de gestión de estaciones:

1. Al sistema es posible agregarle una serie de análisis y alertas que permiten la gestión del estado de las estaciones, alguno de ellos son:
 - a) Incorporación de alertas de tendencia a la baja de batería. Esto enfocado en aquellos casos en que el sistema de carga pueda estar fallando, que el panel solar haya sufrido algún problema o que en estaciones que funcionen sin alimentación suplementaria se pueda planificar a tiempo un recambio de batería. En algunas estaciones la duración de las baterías es de 21 días desde el momento en que se desconecta el panel solar, por lo que al analizar la tendencia de los últimos 7 días podría ser útil para identificar si el rango es el esperable.
 - b) Incorporación de alertas que indiquen una demora en la llegada de los datos desde el momento en que fueron medidos. Esto para identificar aquellas estaciones que poseen un tiempo sensor a nube superior al deseado, como lo que se necesita para sistemas de alerta temprana.

- c) Estadísticas e informes semanales, mensuales y anuales de tiempo de conexión o *uptime* para cada variable, estación y/o grupo.
 - d) Detección de datos no válidos. Esto puede ser analizando la tendencia de sensores en los que se conoce su comportamiento. Un ejemplo es la medición de nieve, debido a que, por los procesos hidrometeorológicos propios de este fenómeno, hay cambios que no pueden ser muy drásticos.
 - e) Detección y validación de datos con estaciones vecinas, esto para poder validar los datos medidos con estaciones similares en zonas equivalentes. Un ejemplo son aquellas estaciones meteorológicas en las que se conoce y mantiene una cierta correlación en los datos, ya que una alza de estos valores podría indicar que existe un problema en la estación.
 - f) Correlación entre variables internas, lo que podría traducirse en una calidad de la medición de la estación que puede ser informada junto con los datos.
2. En las estaciones estudiadas existe una información adicional que solo es guardada en ellas, que son las Tablas de Vecinos. Estos datos, una vez recuperados de las estaciones, podrían incorporarse a la base de datos local que se posee, para así poder identificar el estado de las estaciones y posiblemente instruir cambios en el protocolo para mejorar el enrutamiento que siguen los datos.
 3. Al conocer el comportamiento de las estaciones, ante distintas perturbaciones que hayan sido detectadas utilizando la plataforma, es posible incorporar este conocimiento en el sistema operativo propio de las estaciones. Estas nuevas versiones podrían incorporar la capacidad de realizar el procesamiento en la misma estación y pasando de un sistema centralizado a un sistema distribuido de gestión de fallas, como los mencionados en la sección 2.2.1.
 4. La generación de nuevos protocolos de ruteo que tomen como base SensorScope y LatinOS, incorporando lo logrado en los análisis de la plataforma desarrollada. Este conocimiento permitiría aumentar las capacidades de las estaciones a nuevos niveles.

5. Si bien, actualmente, la capacidad de algunas podría ser limitada, se espera que con el pasar de los años la capacidad de procesamiento y eficiencia energética siga aumentando, lo que permitiría a LatinaUC estar explorando nuevos sistemas inalámbricos y procesamiento de datos que estén a niveles actuales.
6. Incorporar análisis que hagan uso de aprendizaje de máquina e inteligencia artificial. Actualmente hay una gran cantidad de datos que han sido recolectados en el último tiempo, lo que permitiría tener un nuevo enfoque a la hora de realizar estudios que hagan uso de estas herramientas.

BIBLIOGRAFÍA

Alshammari, K., y Ahmed, A. E. S. (2017). An efficient approach for detecting nodes failures in wireless sensor network based on clustering. En *2017 International Symposium on Networks, Computers and Communications (ISNCC)* (p. 1-6). doi: 10.1109/ISNCC.2017.8071988

APIs de geolocalización — Google Maps Platform — Google Cloud. (2020). Google. Descargado de <https://cloud.google.com/maps-platform?hl=es>

Atat, R., Liu, L., Wu, J., Li, G., Ye, C., y Yang, Y. (2018). Big Data Meet Cyber-Physical Systems: A Panoramic Survey. *IEEE Access*, 6, 73603-73636. doi: 10.1109/ACCESS.2018.2878681

Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O., Parlange, M., ... Parlange, M. (2008, April). Sensorscope: Out-of-the-Box Environmental Monitoring. En *2008 International Conference on Information Processing in Sensor Networks (IPSN 2008)* (p. 332-343). doi: 10.1109/IPSN.2008.28

Barron, J. W., Moustapha, A. I., y Selmic, R. R. (2008). Real-Time Implementation of Fault Detection in Wireless Sensor Networks Using Neural Networks. En *Fifth International Conference on Information Technology: New Generations (ITNG 2008)* (p. 378-383). doi: 10.1109/ITNG.2008.187

Barros, S. (2013). *Diseño de protocolos para redes inalámbricas de sensores sobre latinos: metodología y ejemplos* (Tesis de Master no publicada). Pontificia Universidad Católica de Chile.

Bishop, M., Carvalho, M., Ford, R., y Mayron, L. M. (2011). Resilience is More than Availability. En *Proceedings of the 2011 New Security Paradigms Workshop* (p. 95–104).

New York, NY, USA: Association for Computing Machinery. Descargado de <https://doi.org/10.1145/2073276.2073286> doi: 10.1145/2073276.2073286

Campamá, S. (2012). *Sistema operativo para redes inalámbricas de sensores* (Tesis de Master no publicada). Pontificia Universidad Católica de Chile.

Decagon 5TM VWC + Temp. (2014, May). Decagon Devices, Inc. Descargado de <https://www.ictinternational.com/products/5tm/decagon-5tm-vwc-temp/>

DICTUC. (2017). *Estudio Básico: Investigación modelo hidrológico para inundaciones de crecidas en la región Metropolitana. Informe Final para Instituto Nacional de Hidráulica* (Inf. Téc.). Autor.

Eyada, M. M., Saber, W., El Genidy, M. M., y Amer, F. (2020). Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments. *IEEE Access*, 8, 110656-110668. doi: 10.1109/ACCESS.2020.3002164

Garreaud, R. (2013). Warm Winter Storms in Central Chile. *Journal of Hydrometeorology*, 14(5), 1515 - 1534. Descargado de https://journals.ametsoc.org/view/journals/hydr/14/5/jhm-d-12-0135_1.xml doi: 10.1175/JHM-D-12-0135.1

Guerrero, R. (2020). *Field evaluation of sensor to cloud communication metrics for assessing the robustness of a wireless sensor network used in early warning systems* (Tesis de Master no publicada). Pontificia Universidad Católica de Chile.

Hosseini, S., Barker, K., y Ramirez-Marquez, J. E. (2016). A review of definitions and measures of system resilience. *Reliability Engineering & System Safety*, 145, 47–61.

Huang, Y., Martínez, J.-F., Sendra, J., y López, L. (2015). Resilient Wireless Sensor Networks Using Topology Control: A Review. En (Vol. 15, pp. 24735–24770). Multidisciplinary Digital Publishing Institute.

Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O., y Parlange, M. (2010). SensorScope: Application-Specific Sensor Network for Environmental Monitoring. *ACM Transactions on Sensor Networks*, 6(2), 1-32.

Kang, Y., Park, I., Rhee, J., y Lee, Y. (2016). MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data. *IEEE Sensors Journal*, 16(2), 485-497. doi: 10.1109/JSEN.2015.2483499

King, K., y Nittel, S. (2010). Efficient Data Collection and Event Boundary Detection in Wireless Sensor Networks Using Tiny Models. En *Geographic Information Science* (pp. 100–114). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-15300-6_8

Klau, G. W., y Weiskircher, R. (2005). Robustness and resilience. En *Network analysis* (pp. 417–437). Springer.

Kott, A., y Abdelzaher, T. F. (2014). Adaptive, dynamic, and resilient systems. En N. Suri y G. Cabri (Eds.), (cap. Resiliency and Robustness of Complex Systems and Networks). CRC Press.

Laprie, J.-C. (2008). From dependability to resilience. En *38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks* (pp. G8–G9).

MB7574 SCXL-MaxSonar-WRST7. (2020, Nov). Descargado de https://www.maxbotix.com/ultrasonic_sensors/mb7574.htm

Miao, X., Liu, K., He, Y., Papadias, D., Ma, Q., y Liu, Y. (2013). Agnostic Diagnosis: Discovering Silent Failures in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 12(12), 6067-6075.

Moridi, E., Haghparast, M., Hosseinzadeh, M., y Jassbi, S. J. (2020). Fault management frameworks in wireless sensor networks: A survey. *Computer Communications*, 155, 205-226. Descargado de <https://www.sciencedirect.com/science/>

article/pii/S0140366419312150 doi: <https://doi.org/10.1016/j.comcom.2020.03.011>

The most popular database for modern apps. (s.f.). Descargado de <https://www.mongodb.com/>

Panda, M., y Khilar, P. (2015). Distributed Byzantine fault detection technique in wireless sensor networks based on hypothesis testing. *Computers & Electrical Engineering*, 48, 270-285. Descargado de <https://www.sciencedirect.com/science/article/pii/S0045790615002293> doi: <https://doi.org/10.1016/j.compeleceng.2015.06.024>

Pappas, V., Verma, D. C., Swami, A., y Ko, B. J. (2008, apr). On the resiliency of wireless sensor networks with circulation graphs. En R. Suresh (Ed.), *Defense Transformation and Net-Centric Systems 2008*. SPIE. doi: 10.1117/12.783460

R.A.E. (2014). Resiliencia. En *Diccionario Real Academia Española*. Descargado 2020-07-08, de <https://dle.rae.es/resiliencia>

Rafiei, A. (2017). *Resilient wireless sensor networks* (Tesis Doctoral, University of Technology Sydney. Faculty of Engineering and Information Technology.) Descargado de <https://opus.lib.uts.edu.au/handle/10453/123199>

Rautmare, S., y Bhalerao, D. M. (2016). MySQL and NoSQL database comparison for IoT application. En *2016 IEEE International Conference on Advances in Computer Applications (ICACA)* (p. 235-238). doi: 10.1109/ICACA.2016.7887957

Roy, A., Misra, S., Kar, P., y Mondal, A. (2017, jan). Topology Control for Self-Adaptation in Wireless Sensor Networks with Temporary Connection Impairment. *ACM Transactions on Autonomous and Adaptive Systems*, 11(4), 1-34. doi: 10.1145/2979680

Selmic, R. R., Phoha, V. V., y Serwadda, A. (2016). *Wireless Sensor Networks*. Springer International Publishing. doi: 10.1007/978-3-319-46769-6

Sterbenz, J. P., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., y Smith, P. (2010, jun). Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8), 1245–1265. doi: 10.1016/j.comnet.2010.03.005

Yu, C., Hu, J., Li, R., Deng, S., y Yang, R. (2014). Node Fault Diagnosis in WSN Based on RS and SVM. En *2014 International Conference on Wireless Communication and Sensor Network* (p. 153-156).

Yuvaraja, M., y Sabrigiriraj, M. (2015, dec). Fault detection and recovery scheme for routing and lifetime enhancement in WSN. *Wireless Networks*, 23(1), 267–277. doi: 10.1007/s11276-015-1141-7

Zhang, Z., Mehmood, A., Shu, L., Huo, Z., Zhang, Y., y Mukherjee, M. (2018). A Survey on Fault Diagnosis in Wireless Sensor Networks. *IEEE Access*, 6, 11349-11364. doi: 10.1109/ACCESS.2018.2794519

Zidi, S., Moulahi, T., y Alaya, B. (2018). Fault Detection in Wireless Sensor Networks Through SVM Classifier. *IEEE Sensors Journal*, 18(1), 340-347.

APÉNDICES

A. DIAGRAMAS DE CLASES DETALLADOS

A.1 Diagrama de clases de versión previa de SNAPY

La Figura A.1 muestra el diagrama de clases detallado que corresponde a SNAPY.

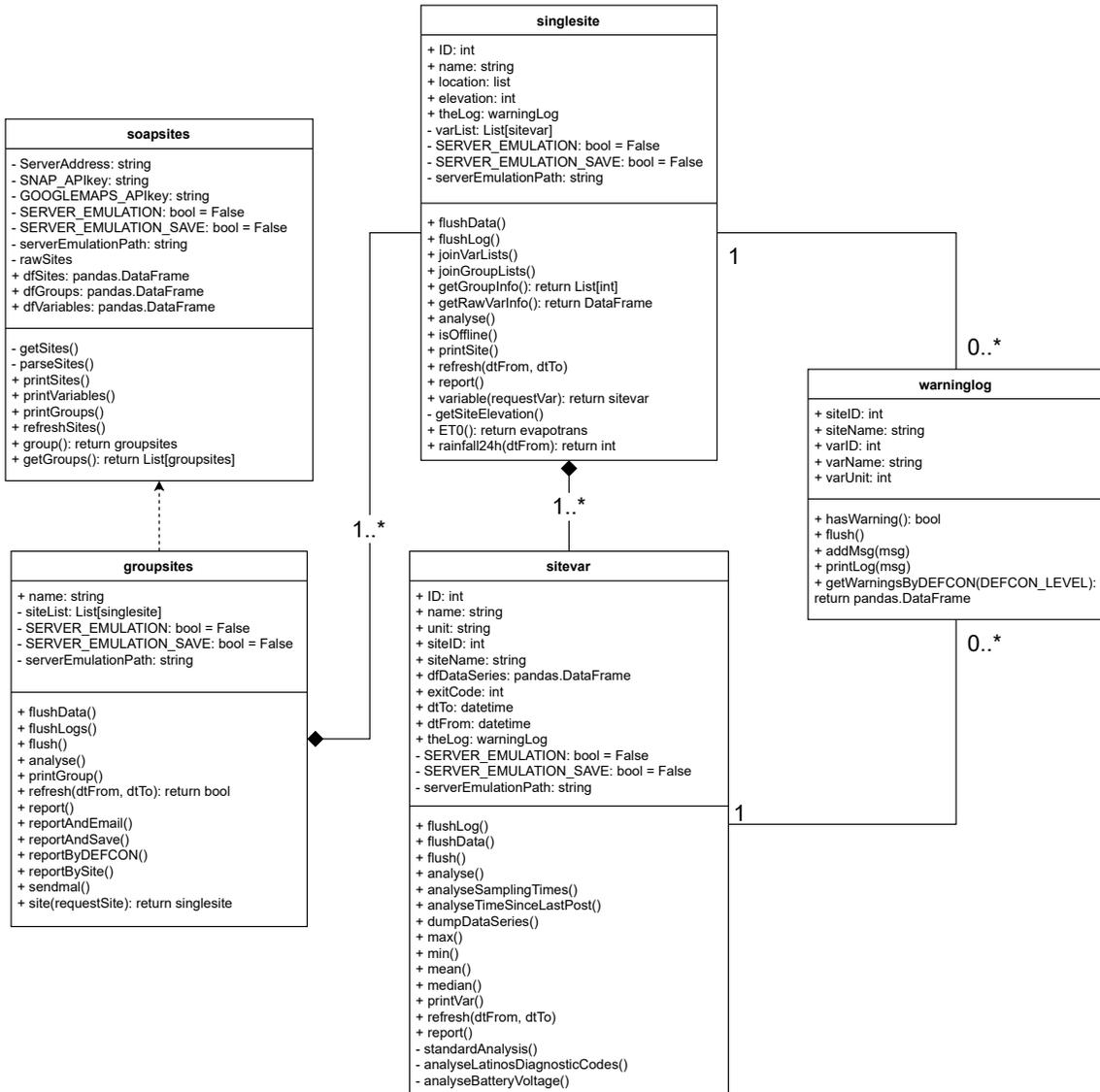


FIGURA A.1. Diagrama de clases versión previa SNAPY.

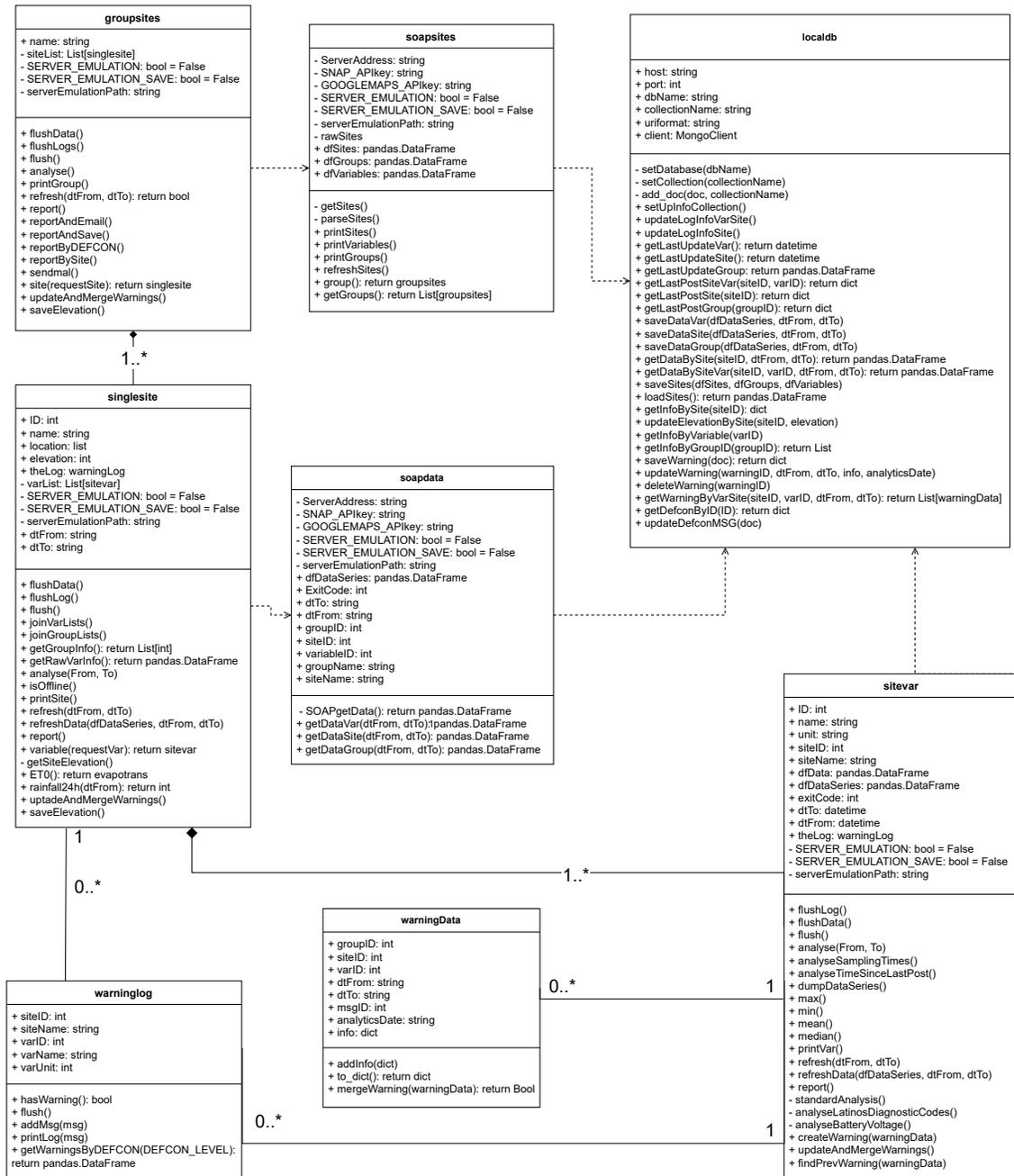


FIGURA A.2. Diagrama de clases versión nueva de SNAPY.

A.2 Diagrama de clases de versión nueva de SNAPY

La Figura A.2 muestra el diagrama de clases detallado que corresponde a las nuevas capacidades de SNAPY.