



Estructuración de datos de sensores para análisis de salud estructural

Carlos Patricio Landero Ambiado
Departamento de Ingeniería Informática y Ciencias de la Computación
Facultad de Ingeniería
Universidad de Concepción

Profesor guía: Gonzalo Rojas Durán

Concepción, 31 de marzo de 2020

Resumen

El monitoreo de salud estructural consiste en el uso de estrategias para detectar anomalías en las condiciones de una estructura, y prevenir, en lo posible, daños de gran magnitud a su salud. Estas estrategias dependen en gran medida de software que permita gestionar y monitorear la información proveniente de una estructura, con el fin de permitir a las autoridades responsables tomar decisiones con respecto a su mantenimiento.

En la presente memoria de título se propone un modelo de estructuración de datos para una plataforma de monitoreo estructural, estableciendo los tipos de fuentes de datos a manejar, cómo estas se almacenan dentro de la plataforma, y recomendaciones de herramientas a utilizar para la capa de almacenamiento de la plataforma. Como resultado final se obtiene una primera versión de la plataforma que valida la correctitud del modelo propuesto, y sobre la cual se pretende integrar nuevos módulos en próximas iteraciones.

Índice

| | | |
|---------|--|---------|
| 1. | Introducción | Pág. 3 |
| 1.1 | Antecedentes | Pág. 3 |
| 1.2 | Justificación | Pág. 4 |
| 1.3 | Objetivo general | Pág. 4 |
| 1.4 | Objetivos específicos | Pág. 5 |
| 1.5 | Alcances y limitaciones | Pág. 5 |
| 1.6 | Descripción del informe | Pág. 5 |
| 2. | Trabajos previos | Pág. 6 |
| 2.1 | Monitoreo de salud estructural | Pág. 6 |
| 2.2 | Modelado de información de construcción | Pág. 7 |
| 2.3 | Aspectos de software en monitoreo de salud estructural | Pág. 8 |
| 2.3.1 | Adquisición de datos | Pág. 8 |
| 2.3.2 | Almacenamiento | Pág. 9 |
| 2.3.2.1 | Apache Cassandra | Pág. 9 |
| 2.3.2.2 | TimescaleDB | Pág. 10 |
| 2.4 | Discusión | Pág. 10 |
| 3. | Requerimientos | Pág. 11 |
| 3.1 | Requerimientos de almacenamiento | Pág. 11 |
| 3.2 | Requerimientos de procesamiento | Pág. 11 |
| 3.3 | Requerimientos de análisis | Pág. 12 |
| 3.4 | Requerimientos de visualización | Pág. 12 |
| 4. | Modelo de datos propuesto | Pág. 13 |
| 4.1 | Modelo conceptual | Pág. 13 |
| 4.1.1 | Infraestructura física | Pág. 14 |
| 4.1.2 | Infraestructura de monitoreo | Pág. 15 |
| 4.1.3 | Gestión de monitoreo visual | Pág. 17 |
| 5. | Implementación | Pág. 20 |
| 5.1 | Arquitectura de hardware de la plataforma | Pág. 20 |
| 5.2 | Arquitectura del prototipo | Pág. 22 |
| 5.2 | Funcionalidades | Pág. 23 |
| 5.3 | Consideraciones adicionales de implementación | Pág. 30 |
| 6. | Evaluación de plataforma para modelo | Pág. 32 |
| 6.1 | Escalabilidad | Pág. 32 |
| 6.2 | Experiencia previa | Pág. 32 |
| 6.3 | Flexibilidad y consistencia | Pág. 33 |
| 5.4 | Desempeño | Pág. 33 |
| 7. | Conclusiones | Pág. 35 |
| 8. | Referencias | Pág. 36 |
| 9. | Anexos | Pág. 38 |

1. Introducción

La mayoría de industrias privadas y gubernamentales desean detectar daños no sólo en sus cadenas de producción, sino también en la infraestructura que utilizan a diario. De esta necesidad, surge el monitoreo de salud estructural (o SHM, por sus siglas en inglés), que hace referencia al proceso de implementación de estrategias para la detección y caracterización de daños en infraestructuras, como lo son puentes, edificios, entre otros.

1.1 Antecedentes

Muchos procesos industriales requieren sistemas de monitoreo estructural, motivados principalmente por el potencial impacto económico y de seguridad que conllevan. Algunos ejemplos se encuentran tanto en la industria de semiconductores, para reducir el número de maquinaria redundante, como en la industria aeroespacial, para la detección de daños en transbordadores espaciales. En cambio, en la ingeniería civil actualmente no hay métodos cuantificables para determinar si los edificios son seguros para su reocupación después de un desastre natural de gran magnitud. Es aquí donde un sistema SHM tiene el potencial de minimizar la incertidumbre asociada con las evaluaciones visuales de daños, ya sea después de un desastre natural, o en estructuras cuya vida útil se acerca a su fin.

Para el mantenimiento de estructuras se pueden utilizar diferentes enfoques. El más sencillo consiste en la “ejecución frente a fallas”, con este método el sistema funciona hasta que algún componente crítico de la estructura falle, para luego ser reemplazado. Dicho enfoque no requiere inversión en sistemas de monitoreo, pero puede ser extremadamente costoso en caso de una falla sin previo aviso, siendo inaceptable cuando la seguridad es una prioridad. Un enfoque más sofisticado que se usa ampliamente hoy en día es el mantenimiento “basado en el tiempo”, este método requiere que los componentes críticos sean reparados o reemplazados cada cierto periodo de tiempo, independientemente de su condición. El método en cuestión resulta más proactivo que la ejecución frente a fallas y ha permitido el desarrollo de sistemas más complejos y seguros.

Los sistemas SHM permiten que este enfoque evolucione hacia un mantenimiento “basado en condición”, en donde el sistema monitorea de forma continua y notifica al operador que se ha detectado daño o degradación en la estructura. Los beneficios asociados con este enfoque se perciben al entregar las alertas con antelación, de tal manera que se puedan tomar medidas correctivas antes de que la degradación evolucione a un nivel crítico. Ahora bien, estos beneficios requieren de la implementación de hardware de monitoreo y software de análisis de datos más sofisticados para consultar los datos medidos [8].

1.2 Justificación

Actualmente, el monitoreo de salud estructural en Chile (en particular, el monitoreo de puentes), ha seguido un enfoque basado en fallas, dependido únicamente de la inspección visual en terreno, la cual puede detectar los problemas que ya posee el puente, pero no puede hacer mucho a la hora de predecir defectos invisibles al ojo humano.

Es en este contexto donde surge el proyecto FONDEF titulado “Plataforma de Monitoreo Estructural de Puentes” (código IT18I0112), adjudicado por la Facultad de Ingeniería de la Universidad de Concepción. Uno de sus principales objetivos es el desarrollo de una plataforma de monitoreo de salud estructural para puentes utilizando un enfoque basado en condición, en la cual se requiere que el sistema capture y transmita datos provenientes desde diferentes tipos de sensores instalados en una estructura. La plataforma debe permitir la gestión y monitoreo en tiempo real de las estructuras, con el fin de detectar y/o anticipar posibles daños estructurales antes de que su situación empeore.

El sistema de monitoreo estructural propuesto se basa en una infraestructura de sensores instalada en un puente a monitorear. Esto implica un gran flujo de datos entrando y saliendo del sistema sin un formato estándar que facilite su uso ulterior en la plataforma. La presente memoria de título tiene como objetivo proponer un modelo estructural de datos para un sistema de monitoreo estructural, que facilite los procesos de almacenamiento, consulta, análisis y visualización de datos.

1.3 Objetivo General

Proponer un modelo de datos que permita estructurar tanto los datos recibidos por una red de sensores instalados en una estructura, como también los datos relacionados con la gestión y monitoreo de un puente. Este modelo debe soportar los requerimientos de almacenamiento, procesamiento, análisis y visualización de la plataforma, siendo compatible con formatos de captura de datos de sensores y de especificación de estructura de monitoreo.

1.4 Objetivos Específicos

Entre los objetivos específicos para esta memoria de título, se considera lo siguiente:

1. Establecer los requerimientos de almacenamiento, procesamiento, análisis y visualización de la plataforma en cuanto a estructuración de datos.
2. Definir el modelo para almacenar la información capturada por los sensores, por medio de un diagrama de clases sujeto a lo requerido para la plataforma.
3. Implementar el modelo de estructuración de los datos en motores de bases de datos compatibles con la arquitectura a utilizar para la plataforma.
4. Evaluar los resultados de la implementación para obtener resultados acerca de cuál de los motores de base de datos utilizados es más conveniente.

1.5 Alcances y limitaciones

1. Para la validación del modelo de estructuración de datos, se implementó un producto mínimo viable utilizando una arquitectura web, montada en una máquina portátil de recursos limitados.
2. Como fuentes de datos, se utilizaron conjuntos de datos representativos de lo que se espera obtener de una red de sensores instalados en una estructura, dejando los detalles de conexión del producto mínimo viable con el hardware de monitoreo fuera del contexto del informe.

1.6 Descripción del Informe

En resumen, el presente informe consta de cuatro secciones principales. Primero se centra en el establecimiento de los requerimientos de la plataforma para realizar la estructuración de datos, en la siguiente sección se da a conocer la propuesta de modelo de datos, seguido de los detalles de implementación del prototipo software, para finalmente realizar una evaluación de la plataforma y del gestor de bases de datos a utilizar.

2. Trabajos previos

2.1 Monitoreo de salud estructural

El monitoreo de salud estructural aplicado al monitoreo de puentes, tiene como principal objetivo el identificar anomalías invisibles a la inspección visual, y que, de no ser tratadas, pueden provocar daños severos a esta. En el desarrollo de esta memoria se analizaron antecedentes de trabajos realizados previamente.

El primer caso de estudio es de un SHM desarrollado en el año 2014, para monitorear el puente de Yeongjong, en la República de Corea [2]. Aquí se utilizó un “framework” para el monitoreo de la información recibida desde una red de sensores instalada (ver Figura 1). Esta información era almacenada en una base de datos NoSQL, utilizando tanto MongoDB como Apache Cassandra, principalmente por sus capacidades para la adquisición de datos, manejo de consultas por rango y manejo de transacciones de grandes volúmenes de datos, en especial a la hora de realizar operaciones de escritura. Este framework se destaca por estructurar la información de la plataforma en capas (desde la adquisición de los datos hasta su procesamiento) y el uso de diferentes “datastores” (o almacenes de datos) en un hardware distribuido, dado que estas herramientas cumplen labores distintas (MongoDB se ocupaba del almacenamiento persistente, mientras que Cassandra del almacenamiento para consultas).

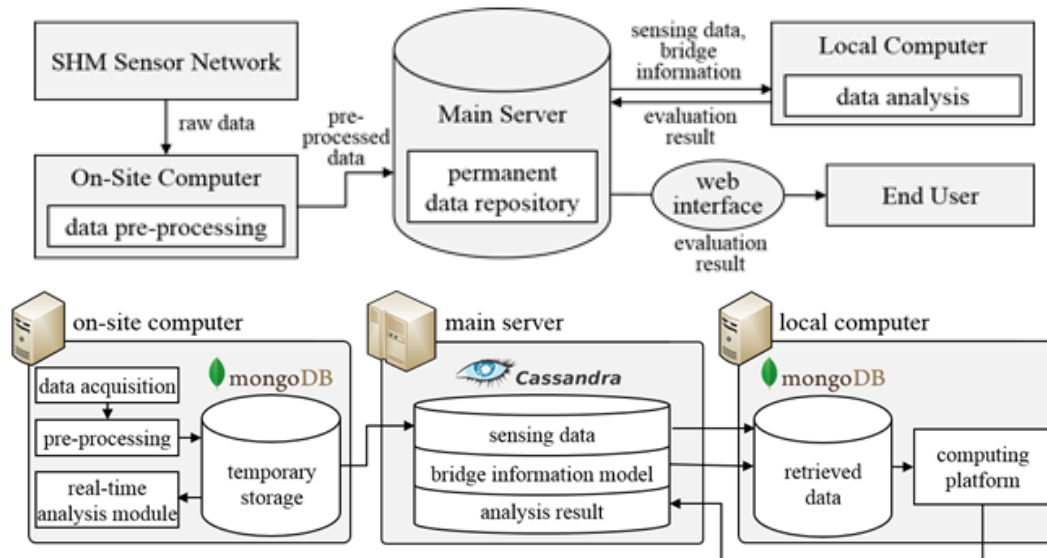


Figura 1: Framework utilizado para el manejo de los datos de monitoreo (arriba), y detalle de infraestructura de las herramientas de almacenamiento (abajo), Fuente: [2].

El segundo caso de estudio analizado tuvo lugar en el año 2008, en un puente colgante en Harbin, República Popular China [4]. Aquí se presenta un problema de “benchmark” con datos obtenidos del sistema SHM implementado en esta estructura. Los datos y el estudio sobre la detección de daños, tanto en vigas como en cables de suspensión durante un periodo de tiempo (desde que el puente es “sano” hasta que presenta daño estructural visible), sirven como guía para el estudio y evaluación de las condiciones de un puente tras sufrir daño.

2.2 Modelado de información de construcción

Más conocido como “Building Information Modelling” (“BIM” por sus siglas en inglés) consiste en una tecnología usada para la interoperabilidad a la hora de crear modelos digitales de estructuras, a través del intercambio de información por medio de estándares abiertos en la industria de la construcción y la ingeniería. De esta forma, tanto arquitectos como ingenieros trabajan sobre el mismo modelo digital, ya sea en 2D o en 3D, y en cualquier fase de la obra [9]. Del mismo contexto surge una variante especializada de BIM, centrada en ser el estándar para el modelado de información de puentes, llamada “Bridge Information Modelling” (BrIM).

Un enfoque utilizado para estandarizar y extraer la información en los archivos de esta metodología es utilizando el lenguaje de marcado XML, con el objetivo de reducir la carga de trabajo y los errores humanos producidos al manipular la información, además de facilitar la construcción y descomposición de datos en una base de datos NoSQL [2].

Estas metodologías tienen el potencial de otorgar funcionalidades distintivas a la plataforma, pero esto dependerá de los casos de uso del contexto donde se aplique, pues estos pueden ir desde un sistema que identifique los componentes y secciones de una estructura, a sólo generar una visualización de la estructura.

2.3 Aspectos de software en monitoreo de salud estructural

Muchos de los desafíos y exigencias que enfrenta el monitoreo de salud estructural a nivel de software, van de la mano con los principios del “Big Data”, como lo son el volumen (cantidad de datos a almacenar), variedad (formato de los datos), velocidad (cuán rápido se generan y procesan los datos), variabilidad (nivel de inconsistencia de los datos) y veracidad (cuán útiles son los datos) [16]. En esta subsección se destacan aquellos aspectos de software más relevantes en el monitoreo de salud estructural y el manejo de Big Data, que son aplicables a la presente memoria, como lo son estrategias de adquisición y almacenamiento de datos.

2.3.1 Adquisición de datos

En el contexto de la ingesta de datos heterogéneos, como puede observarse en la Figura 2, se propone un modelo que facilita la adquisición y fusión de datos desde múltiples fuentes [1]. Este incluye cuatro estrategias para la sincronización, segmentación, división e indexación de datos, con el fin de recopilar datos desde múltiples fuentes. Estas estrategias fueron aplicadas en un contexto de “Internet of Things” (o IoT) para tratar un gran volumen de datos a nivel industrial. Aunque no hay evidencia de que se haya utilizado en algún sistema SHM, resulta una estrategia factible para la sincronización de ingesta de fuentes de datos, en la cual aquella información que se requiere con mayor inmediatez sea procesada en un formato de “streaming”, para luego ser almacenada en un formato de mayor persistencia.

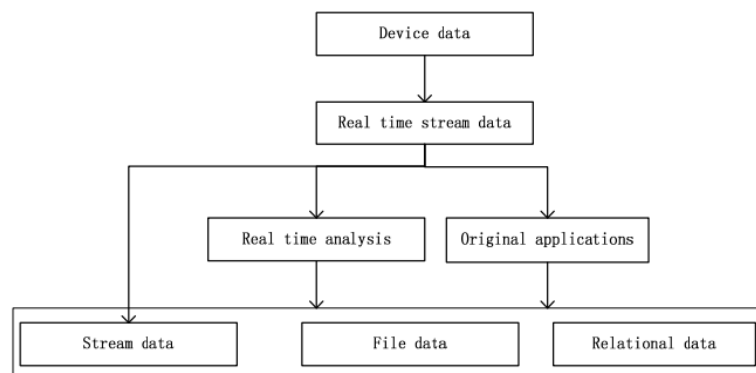


Figura 2: Múltiples fuentes de datos usadas como datos de los dispositivos, Fuente: [1].

2.3.2 Almacenamiento

La capa de almacenamiento representa una pieza clave dentro de la construcción de una plataforma de monitoreo estructural, pues de esta depende el asegurar la persistencia y rapidez de acceso a la información recopilada. En el contexto de este proyecto se destacan dos herramientas, una de ellas es Apache Cassandra, un gestor de bases de datos distribuidas, y TimescaleDB, una extensión de PostgreSQL especializada para datos en series de tiempo.

2.3.2.1 Apache Cassandra

Apache Cassandra es un DDBMS (Distributed Database Management System) utilizado para almacenar grandes cantidades de datos y realizar consultas optimizadas. Es un sistema NoSQL que usa un modelo de datos “Wide-Column”, que permite la consistencia eventual de los datos, en la cual se asume que, al repetirse una misma tupla, el último valor ingresado es el correcto [4].

El modelo conceptual en bases de datos NoSQL es importante para sacar provecho a sus características y rendimiento. En [3] se describe un conjunto de etapas, principios y patrones para utilizar Cassandra como gestor de bases de datos. En resumen, tal como se muestra en la Figura 3, se llevan a cabo las mismas etapas que en el modelado relacional (Modelado conceptual, optimización física y el desarrollo de un esquema de bases de datos), con la excepción de que en el modelado relacional se lleva a cabo un proceso de normalización sobre el modelo de datos, el cual se omite en el modelado no relacional, dando prioridad a las consultas pertinentes del sistema, al flujo de trabajo de la aplicación y los patrones de acceso a los datos.

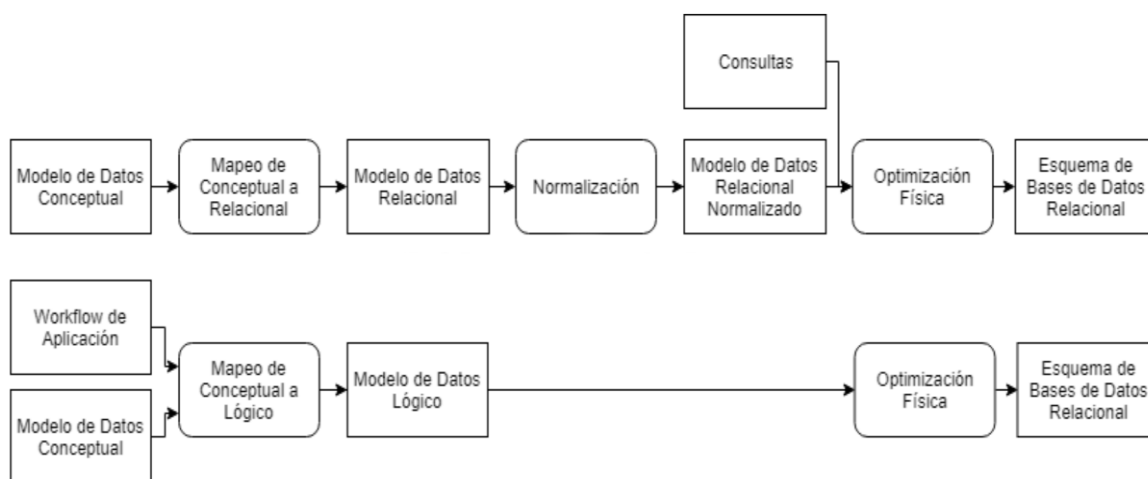


Figura 3: Modelamiento de datos relacional (arriba) versus la metodología no relacional (abajo) [12].

2.3.2.2 TimescaleDB

TimescaleDB es una extensión del administrador de bases de datos relacionales (RDBMS) PostgreSQL, optimizada para bases de datos basadas en series de tiempo. Permite una rápida ingesta de datos y la realización de consultas complejas, es compatible con todas las instrucciones de SQL y con todas las funcionalidades de PostgreSQL, además de escalar de formas previamente reservadas para bases de datos NoSQL.

La información en TimescaleDB es manejada a través de “hypertables”, abstracciones de una vista (o “view”) compuesta de múltiples tablas individuales (llamadas “chunks”), las cuales almacenan los datos. A la fecha, la opción de distribuir el contenido de una “hypertable” de TimescaleDB en un clúster está en periodo de beta.

Al usar esta extensión, se deben tener en cuenta ciertos aspectos. Uno de estos es la necesidad de escoger previamente un “schema” e indicar explícitamente los índices a usar. Otro aspecto es definir un “chunk_time_interval”, para mantener los datos del intervalo más reciente en memoria principal, la recomendación es que el tamaño de estos intervalos no exceda un 25% de la memoria del sistema.

2.4 Discusión

De lo expuesto en esta sección, se pueden obtener las siguientes reflexiones:

- El monitoreo de salud estructural lleva años siendo implementando en diversas partes del mundo. Aspectos como la arquitectura y la experiencia en detección de daños inspiran el desarrollo de la presente memoria.
- Los datos en tiempo real pueden requerir un preprocesamiento en la plataforma, con tal de que la visualización de datos tenga menor latencia. Luego estos datos pasarán a la capa de almacenamiento, donde deben mantenerse persistentes durante un periodo de tiempo a definir.
- Ante las características que presentan Cassandra y TimescaleDB, es necesario realizar una evaluación de ambos candidatos, con tal de establecer el gestor de bases de datos adecuado al contexto de la plataforma.
- La velocidad, almacenamiento y estructuración de datos surgen como las principales exigencias en cuanto a Big Data en la presente memoria. Y son estos los que influirán en las demás secciones de la plataforma.

En la siguiente sección del presente informe, se expondrá cómo estos aspectos en trabajos previos influyen a los requisitos de almacenamiento, visualización, análisis y procesamiento del modelo de datos, y por consiguiente, a la implementación de la plataforma.

3. Requerimientos

En esta sección se abordan los requerimientos de estructuración de datos para la plataforma de monitoreo estructural. Se tomaron en cuenta tanto los requerimientos funcionales por parte de los “stakeholders” del proyecto (Profesionales del MOP y de la Facultad de Ingeniería, ver Anexo 9.1), además de los resultados obtenidos de una simulación de “streaming” de datos, realizada con Python y Apache Kafka. En cuanto a los requisitos de estructuración, estos se clasifican en cuatro tipos: procesamiento, visualización, almacenamiento y análisis.

3.1 Requerimientos de procesamiento

Tomando en cuenta las características de otras herramientas SHM existentes, es necesario que el modelo de estructuración soporte los siguientes tipos de consultas:

1. Dado un sensor o grupos de sensores, obtener sus datos filtrando entre rangos de fechas y/o valores específicos.
2. Dado un usuario autorizado, acceder a los informes de monitoreo visual de una estructura, y a los hallazgos asociados a este.
3. Dado un usuario autorizado, obtener información general acerca de una estructura en particular, o de un grupo de estructuras dentro del inventario (Ejemplo: Ubicación, longitud, materiales, etc.).
4. Dada una estructura, obtener todos los sensores instalados, los detalles acerca del estado de monitoreo, además de los recursos audiovisuales que contribuyan al monitoreo.

3.2 Requerimientos de Visualización

Un usuario de una plataforma SHM interactúa principalmente con dos formas de visualización, por medio de gráficas (Histogramas, OHLC, entre otras) y con “Logs” para visualizar el registro histórico de la estructura. En vista de esto, el modelo de estructuración debe considerar lo siguiente:

1. Poseer métodos de acceso que faciliten la ingesta de datos a las herramientas cuya labor es generar las gráficas.
2. Registrar eventos (anomalías, lecturas fuera de rango, etc.) como parte del “estado de monitoreo” de una estructura, por medio de un registro histórico.

3.3 Requerimientos de Almacenamiento

Para definir estos requerimientos, se llevó a cabo una simulación de “streaming” con datos de aceleración provenientes del dataset del puente en Harbin [2], en la cual se utilizaron dos máquinas ejecutando Apache Kafka, la primera cumplía el rol de “Productor de datos”, mientras el segundo cumplía el rol de “Consumidor de Datos”. La Figura 4 muestra un resumen de la simulación, la cual consistió de las siguientes etapas:

1. Leer los archivos presentes en el conjunto de datos del puente situado en Harbin, utilizando el lenguaje Python 3.6.
2. Transmitir estos datos al servidor de Kafka, ubicado en el productor de datos (por medio de la librería “KafkaProducer”)
3. Establecer una conexión entre los nodos “Consumidor” y “Productor”, con tal de recibir los datos en la segunda máquina.
4. Procesar y almacenar las lecturas recibidas por medio de Python y sus librerías para Kafka, Cassandra y PostgreSQL.

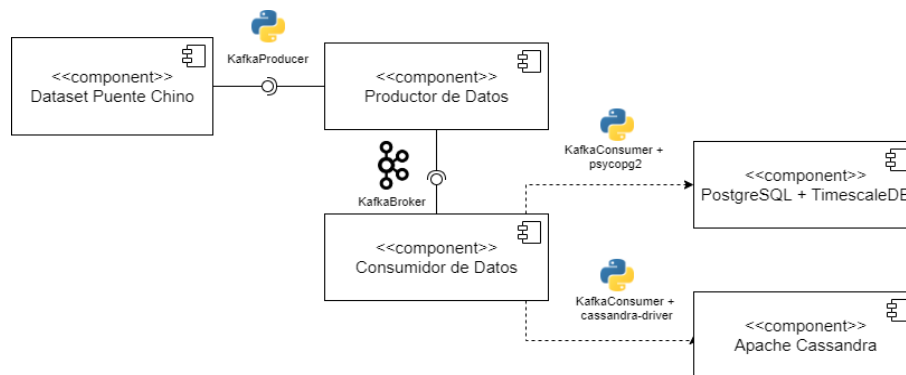


Figura 4: Diagrama de la simulación de streaming realizada.

En resumen, se transmitieron 295 MB en aproximadamente 46 segundos, a un volumen de 6,4 MB/s. De esta se dedujo que los atributos más importantes para el monitoreo estructural son fecha, frecuencia del sensor y lectura registrada, ya que están presentes en todas las fuentes de datos (de forma explícita, o implícita al realizar un preprocesamiento) y permiten caracterizar en el tiempo el estado de una estructura.

3.4 Requerimientos de Análisis

Para que la plataforma pueda analizar la situación estructural de un puente, y pueda “predecir” futuras fallas, se requiere que el modelo de estructuración de los datos relacione tanto las lecturas de los sensores, los registros históricos de los componentes de la estructura, como también la información proveniente de los informes de monitoreo visual. Al hacer esto, se espera que a futuro la plataforma detecte los daños antes que se produzcan, pues ya habrá aprendido a detectar patrones de otros puentes de similares características.

4.1.1 Infraestructura física de la plataforma

Esta sección del modelo conceptual agrupa a las clases que representan las estructuras a monitorear (ver Figura 6, primera imagen). La clase “Estructura” representa a los puentes, mientras que la clase “Zona de estructura” representa a las partes que lo componen, siendo posible clasificarlas según tipos (Ejemplos: Tablero, Viga, Estribo, Estructura portante, etc.). Si observamos la Figura 6 (segunda imagen), podemos ver que un objeto de la clase “Estructura” posee varias instancias de la clase “Zona estructura”, y a su vez, estas zonas se relacionan con un objeto del tipo “Tipo de Zona”. Dado que estas clases representan al puente en terreno, los datos de estas entidades son actualizados con menor frecuencia.

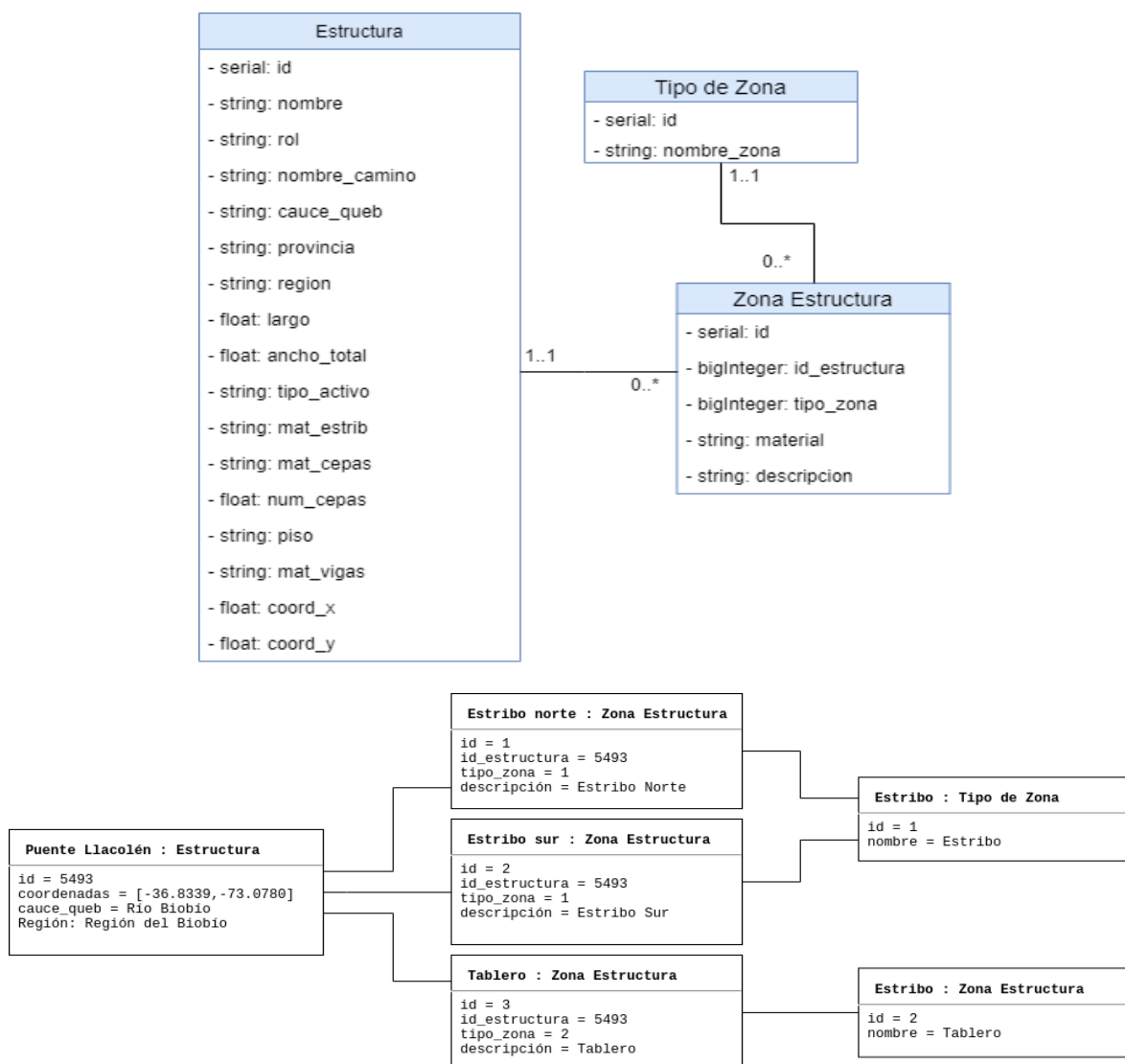


Figura 6: Diagrama de clases y diagrama de objeto que ejemplifican la infraestructura física en la plataforma.

4.1.2 Infraestructura de monitoreo

El grupo de clases presente en la Figura 7, hacen referencia al equipamiento de monitoreo a instalar en una estructura (Sensores, DAQ), sus características (estados, calibraciones, conexiones, canales, etc.), lecturas y alertas registradas durante el monitoreo. De esta forma, la plataforma permite acceder a todo el detalle de la infraestructura de monitoreo, pues a partir de un sensor y un instante de tiempo, se puede saber en qué fecha fue instalado, cuál es su configuración en dicho instante (umbrales de sensibilidad), a que DAQ está o estuvo conectado (y a cuáles estuvo conectado) e identificar si pertenece o pertenecía a un conjunto o “clúster” de sensores en una intersección de zonas en la estructura.

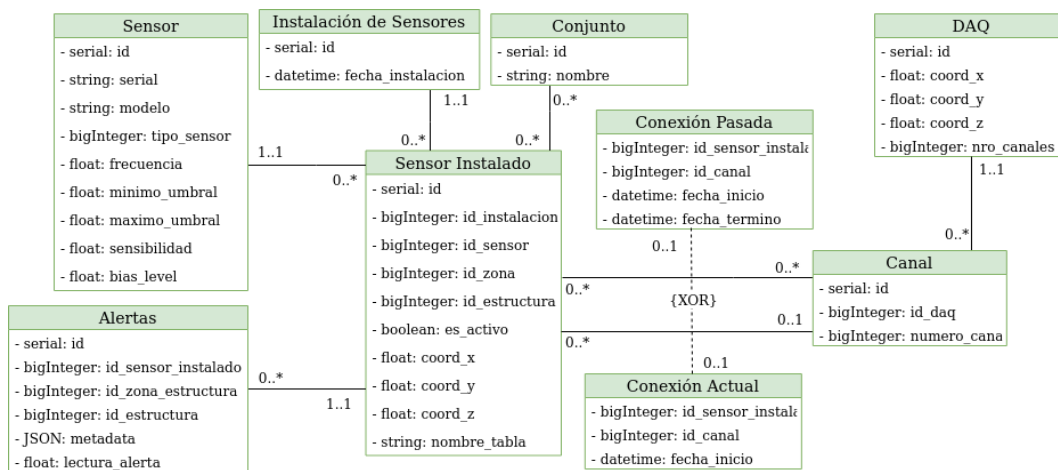


Figura 7: Sección del diagrama de clases dedicada a la infraestructura de monitoreo

Como se puede observar en la Figura anterior, existen diferencias entre las clases “Sensor”, “Sensor Instalado” e “Instalación de sensor”, la clase “Sensor” representa al dispositivo físico, el cual a su vez se relaciona con la clase “Tipo de Sensor”, la clase “Sensor Instalado” representa la ubicación actual en donde está instalado, pudiendo así representarlo cuando es cambiado de ubicación, y la clase “Instalación de sensor” representa el instante de tiempo en el que fueron instalados uno o más sensores.

En cuanto a las relaciones entre las clases, tenemos el siguiente ejemplo en la Figura 8: las instancias de “Sensor Instalado” se relacionan con una o más instancias de “Conexión Actual”, según las necesidades que tenga el sensor (si el sensor es de un eje, sólo requiere de una conexión actual), al mismo tiempo, cada conexión está asociada a un canal, que a su vez está asociado con un DAQ. Mientras que “sensor instalado” y “DAQ” poseen varias instancias de las clases “Estado Sensor” y “Estado DAQ” para representar los cambios de estado que han experimentado en el tiempo.

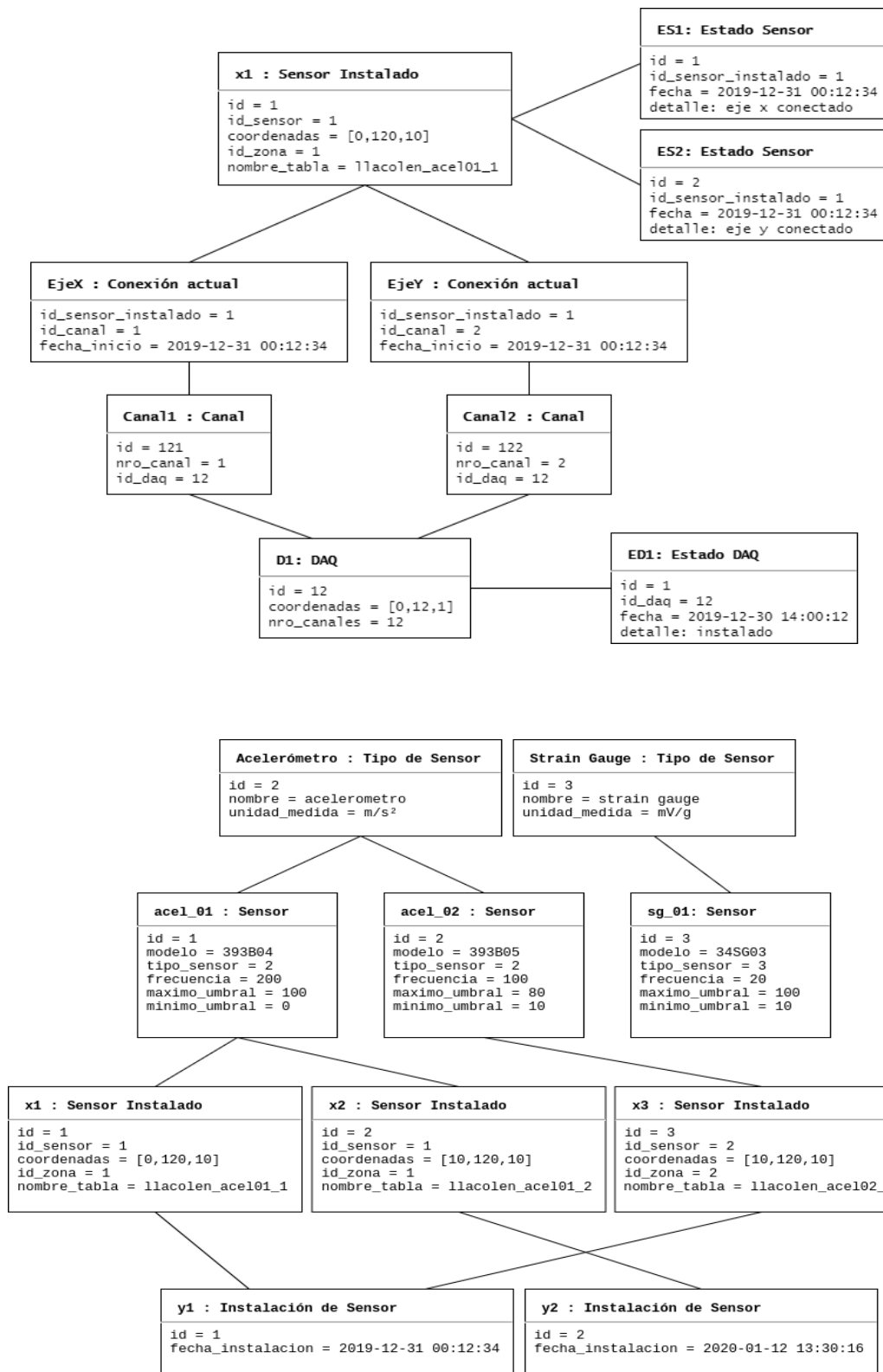


Figura 8: Diagramas de objeto que ejemplifican las clases de la sección “infraestructura de monitoreo”.

4.1.3 Elementos de software y de gestión del monitoreo

Esta sección (ver Figura 9) se compone de las clases relacionadas con aspectos de software (Usuario, Archivos, Grupos definidos por el usuario, etc.), gestión del monitoreo visual (Informes de monitoreo visual, Hallazgos, etc.) y generación de reportes (Reportes de monitoreo).

El objetivo de esta sección es permitir a los inspectores en terreno gestionar los informes de monitoreo estructural y hallazgos detectados asociados. De esta forma, al integrar esta sección al resto del modelo, es posible reportar que hay daño visible en una zona de la estructura, con tal de enfocar el monitoreo y hacer un seguimiento del estado actual. O bien el caso opuesto, donde a partir de las alertas generadas a partir las lecturas, se genere un reporte del estado de monitoreo, que sirva de apoyo a una futura inspección en terreno.

Como se muestra en la Figura 10 (primera imagen), cada instancia de “Informe Monitoreo Visual” está relacionada con una instancia de “Estructura” y con la instancia de “Usuario” que creó dicho informe. A su vez, cada informe de monitoreo puede tener asociadas varias instancias de la clase “Hallazgo Visual”, y viceversa, cada hallazgo puede estar asociado a varios informes de monitoreo visual.

Por otra parte, tenemos las clases que modelan grupos de sensores en la plataforma (ver Figura 10, segunda imagen). La clase “Conjunto” fue pensada con el fin de agrupar la información de sensores ubicados en un área de interés en la estructura (Ejemplo: la intersección de dos zonas de la estructura), mientras que la clase “Grupo definido por usuario” tiene por propósito permitir al usuario seleccionar sensores de diferentes tipos y estructuras, y agruparlos para facilitar su acceso en la plataforma.

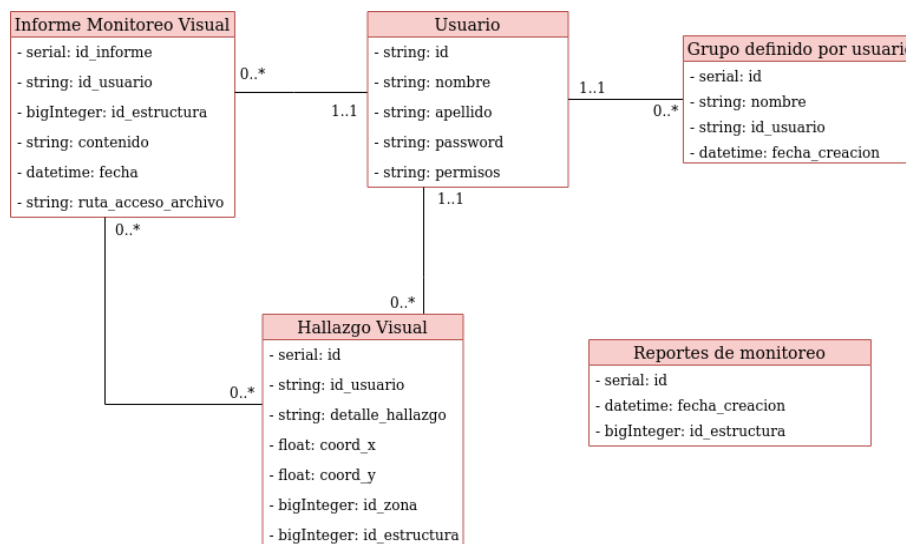


Figura 9: Sección del diagrama de clases dedicada a la inspección visual de estructuras.

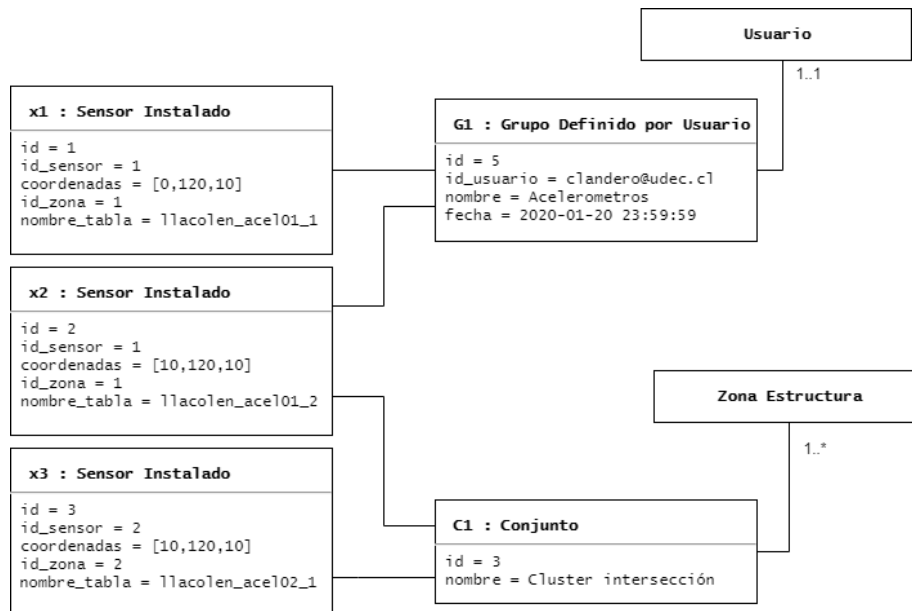
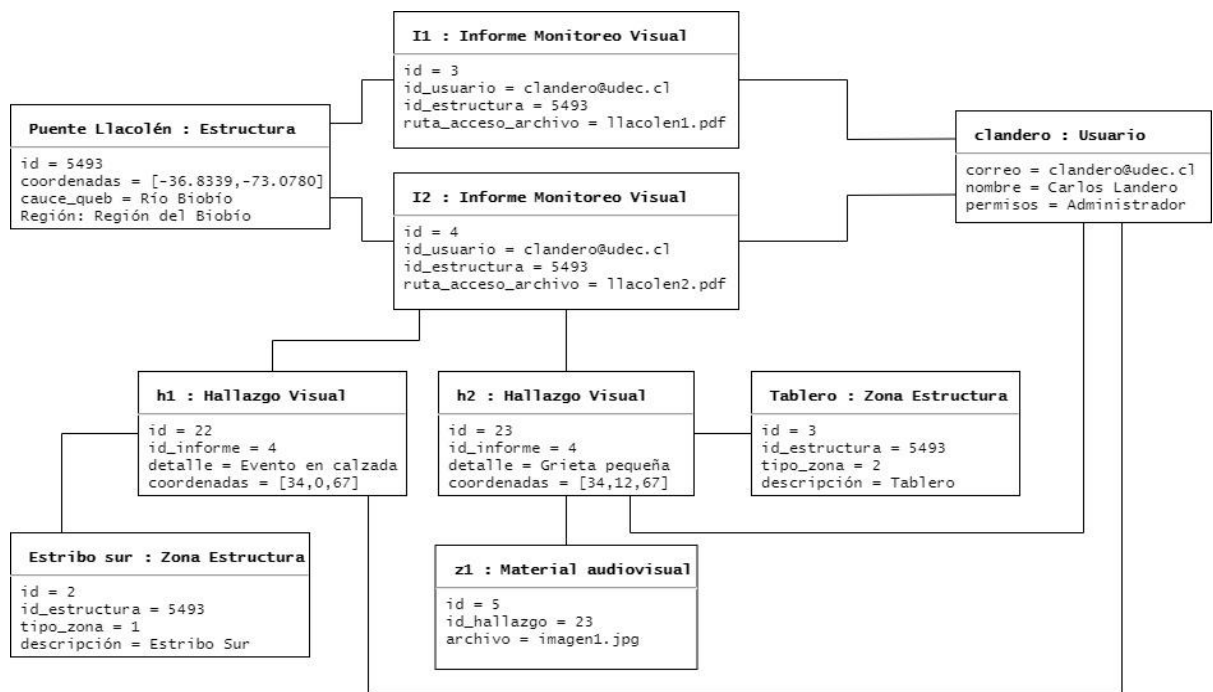


Figura 10: Diagramas de objeto relacionados con la gestión de monitoreo.

Finalmente, para la generación de reportes de monitoreo se tiene la clase “Reporte de monitoreo”, la Figura 11 muestra cómo las instancias de esta clase se relacionan con una sola instancia de “Estructura”, donde a partir de esto, obtendrán las instancias de “Estado de Sensor”, “Estado de Zona”, “Estado de Monitoreo” y “Alerta” desde la última vez que se generó un reporte.

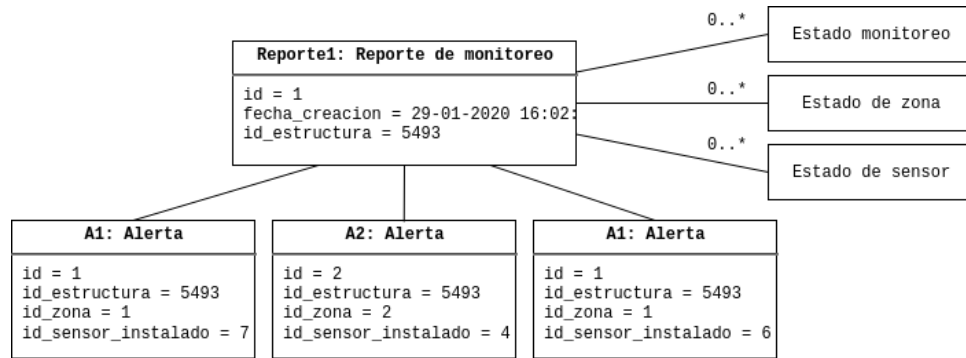


Figura 11: Diagrama de objeto que muestra los objetos asociados con una instancia de “Reporte de monitoreo”.

5. Implementación

En esta sección se exponen los detalles de implementación acerca del prototipo software implementado para validar el modelo de datos propuesto. Esta implementación considera aspectos tanto de la instalación en terreno, las herramientas a utilizar y cómo estos componentes se relacionan en la arquitectura de la plataforma.

5.1 Arquitectura de hardware de la plataforma

En concreto, la operación en terreno de la plataforma SHM (ver Figura 12) consistirá de una red de sensores coordinados a un “data adquisitor” (DAQ), el cual estará conectado a un microcomputador (ya sea USB, WLAN o Ethernet) para transmitir los datos, luego este microcomputador retransmite a la plataforma SHM por medio del protocolo MQTT [17]. Esta memoria de título se centra en establecer un formato para los datos recibidos desde Apache Kafka, con tal de facilitar su almacenamiento, visualización, análisis y procesamiento, además de gestionar la información existente en el ministerio de obras públicas y la dirección de vialidad.

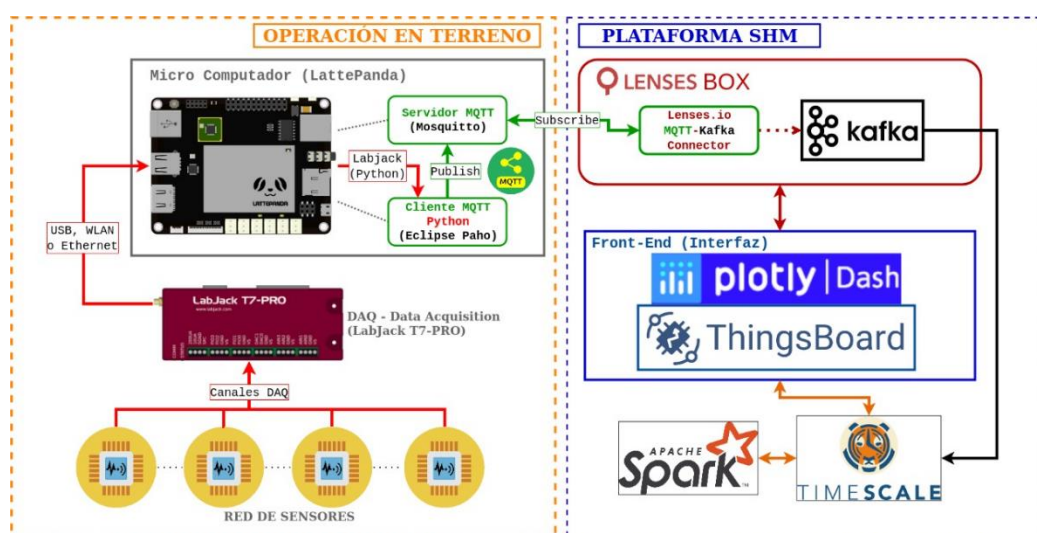


Figura 12: Arquitectura de software para plataforma SHM.

La plataforma SHM de este proyecto se compone de diferentes herramientas y módulos (Visualización de datos en tiempo real, visualización de datos históricos y Visualización de estructura en 2D/3D), para entender cómo interactúan entre sí, tenemos en la Figura 13 el siguiente diagrama de componentes.

En el componente “Almacenamiento y Procesamiento”, es Apache Kafka quien recibe los datos desde la instalación en terreno, redirigiendo estas lecturas hacia la base de datos. Mientras que la inspección de hallazgos se almacena directamente.

En cuanto a las interfaces que prestan servicios a los componentes de visualización, el componente “Visualización en tiempo real” recibe desde Spark un “log de alertas” y las lecturas desde Kafka, los componentes “Gestión de puentes y sensores” y “Visualización de estructura en 3D” realizan consultas a la base de datos sobre los sensores y hallazgos, mientras que “Visualización de datos históricos” recibe desde Spark los datos procesados por medio de una API. Finalmente, cada uno de estos módulos otorga acceso al componente “Vista resumen por sensor”, quien a su vez presta servicios al componente “Vista general de puente”.

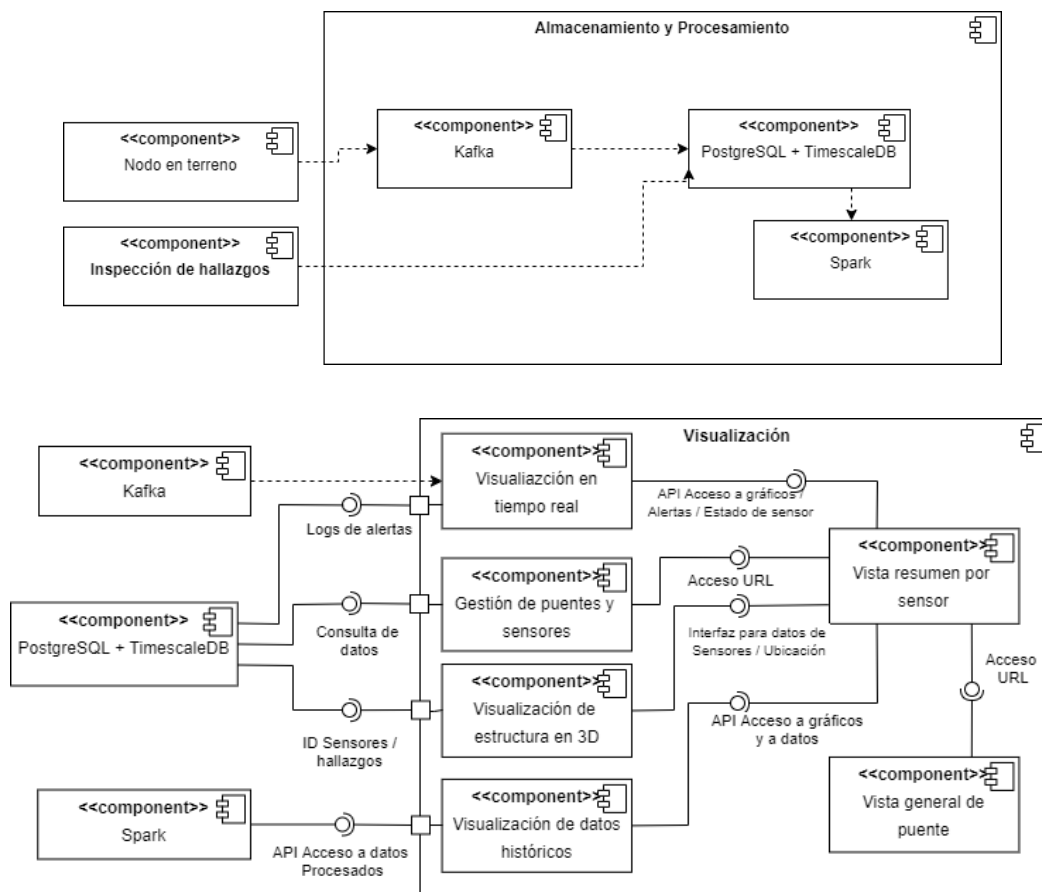


Figura 13: Diagrama de componentes de la plataforma SHM

5.2 Arquitectura del prototipo

El prototipo desarrollado en la presente memoria utiliza el patrón de diseño “Model-View-Template”, en el cual un usuario interactúa con el front-end de la plataforma (“Template”), esta interacción es recibida por el back-end de la plataforma (“View”), quien a su vez realiza una solicitud a la base de datos (“Model”). Esta solicitud genera una respuesta, la cual es redirigida por el back-end al usuario, para ser desplegada en la interfaz gráfica.

Como puede observarse en la Figura 14, la capa “Model” fue desarrollada tanto en Cassandra como en TimescaleDB, la capa “View” fue creada usando el “microframework” Flask, basado en Python (versión 3.6.9) y el “Template” consiste de una interfaz web desarrollada con el framework de CSS “Bootstrap”.

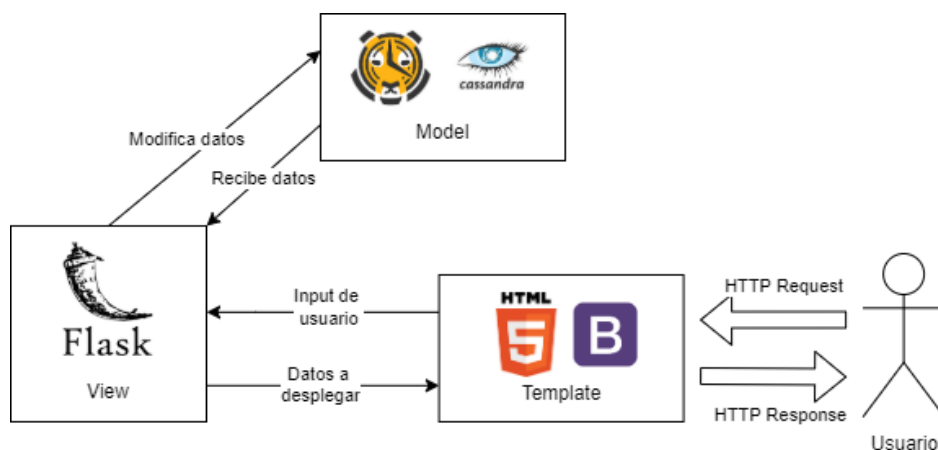


Figura 14: Patrón de diseño “Model-View-Template” en la plataforma SHM

5.3 Funcionalidades

La siguiente sección describe el prototipo de software desarrollado durante la presente memoria, este prototipo consiste principalmente en las primeras versiones de los componentes “Gestión de puentes y sensores”, “Vista resumen por sensor” y “Vista general de puente” descritos en la arquitectura. Este prototipo sienta las bases para una primera versión de la plataforma SHM, la cual fue validada tanto por el equipo de desarrollo del proyecto, como por los “stakeholders”. A continuación, se describen las funcionalidades y características implementadas exitosamente, con referencia al requerimiento asociado en el Anexo 9.1.

5.3.1 Exploración del inventario de puentes

La primera funcionalidad a la que un usuario de la plataforma accede es a la exploración del inventario de puentes (ver Figura 15), esta vista permite buscar una estructura (ya sea escribiendo una de sus características o buscando en el mapa) y acceder a su información. El listado de estructuras consiste en la información disponible desde el inventario de puentes de la dirección de vialidad [14]. Esta funcionalidad atiende al requerimiento RV2.

5.3.2 Visualización del resumen de una estructura

Cuando un usuario ha seleccionado una estructura en la vista de exploración, la plataforma le redirige a la visualización del resumen de la estructura seleccionada, la cual podemos observar en la Figura 16 (primera imagen). Desde aquí, el usuario puede acceder a un resumen con la ubicación, tipo de activo y composición de la estructura. A su vez, es posible acceder desde esta vista a la información relacionada con las zonas que le componen (ver Figura 16, segunda imagen), el historial de monitoreo, los inventarios de sensores y DAQs instalados, la visualización en 3D y los informes de monitoreo visual asociados (ver Figura 17). Esta funcionalidad está asociada a los requerimientos RV2, RV3, RV4, RV5 y RP2.

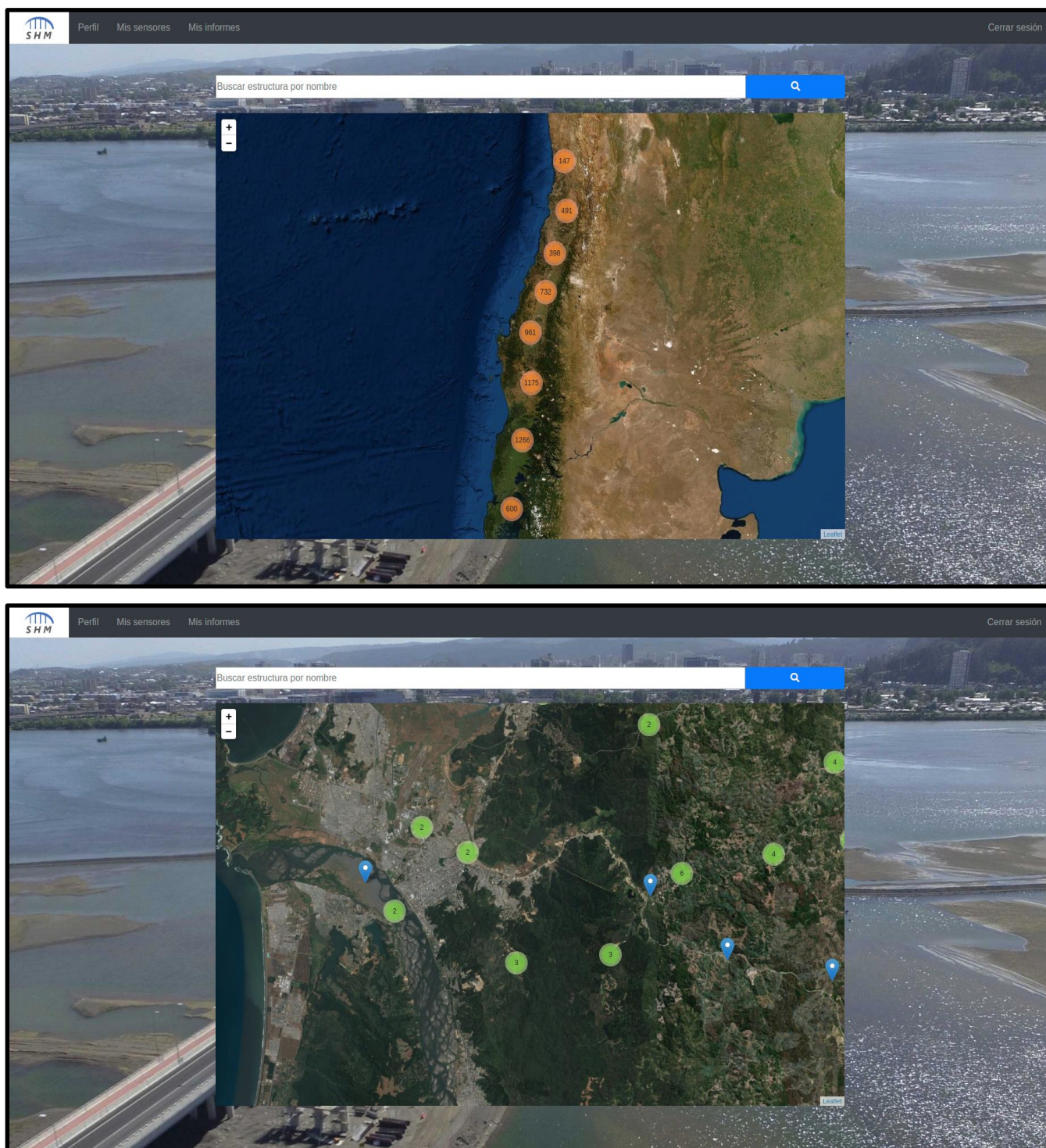


Figura 15: Capturas de pantalla con la exploración de puentes disponibles.

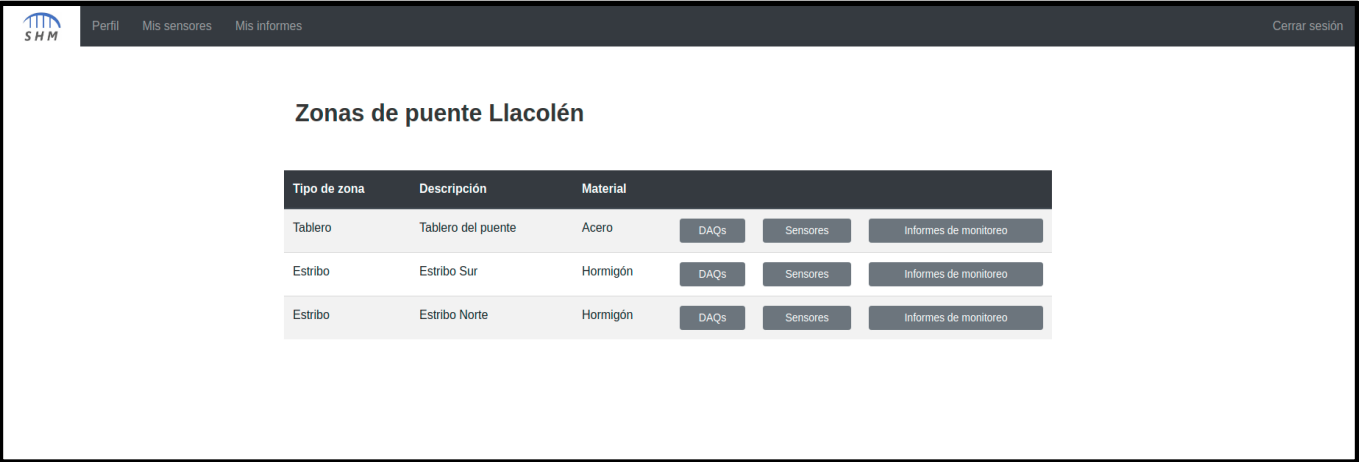
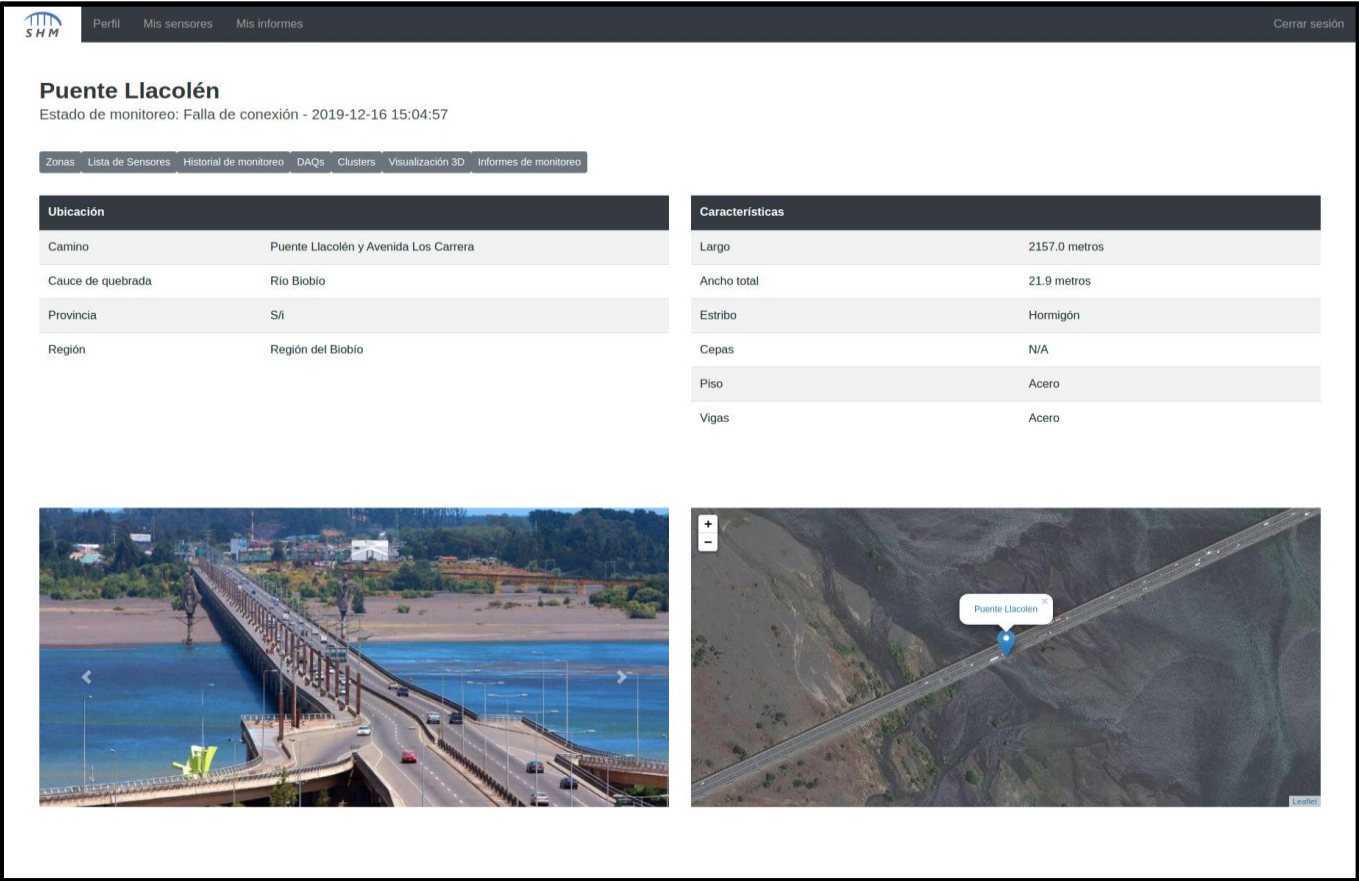


Figura 16: Capturas de pantalla del resumen del Puente Llacolén y sus zonas que lo componen.

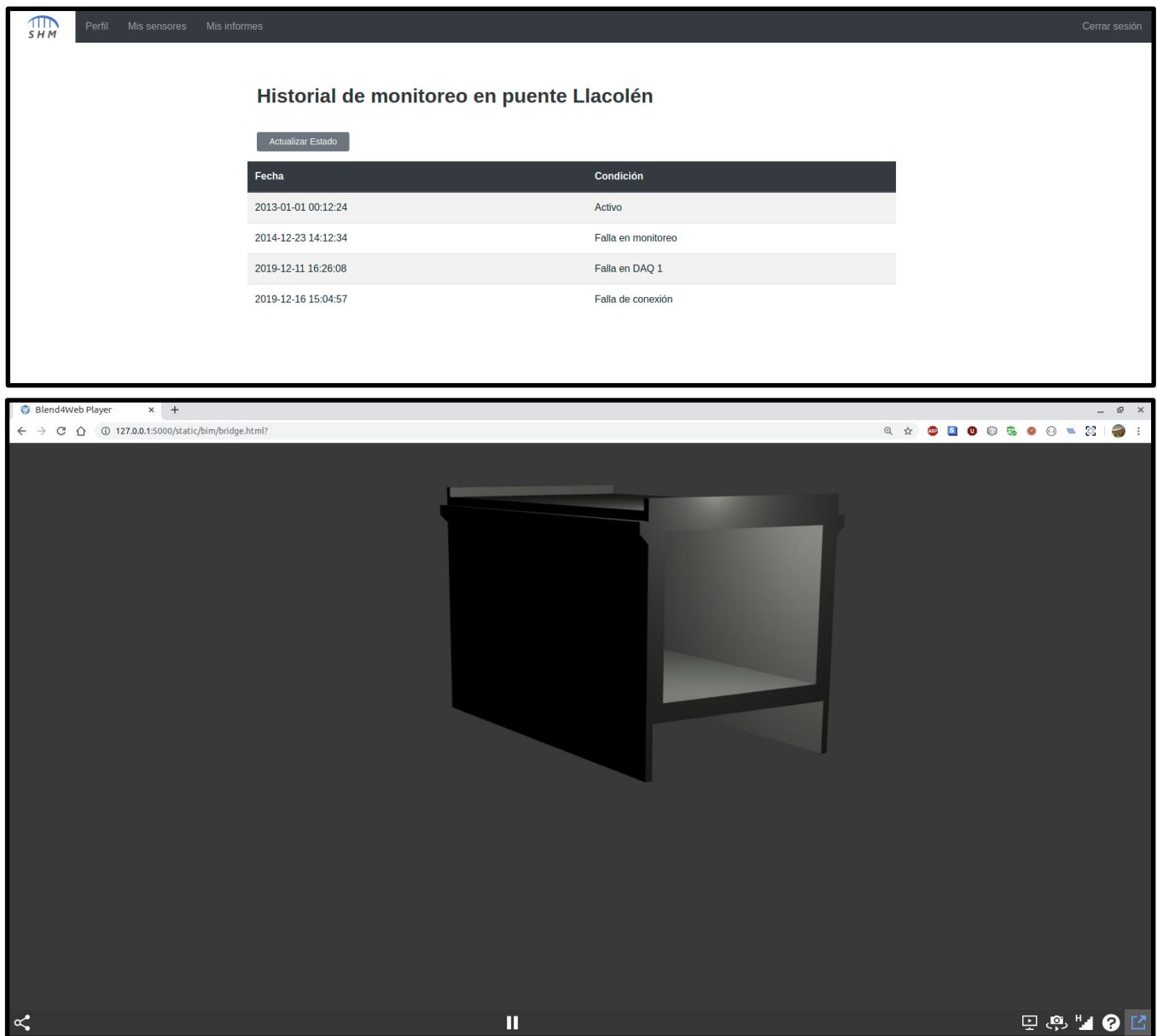


Figura 17: Capturas de pantalla del historial de monitoreo de la estructura (primera imagen) y la visualización 3D del archivo BIM asociado (segunda imagen).

5.3.3 Inventario de equipamiento de la estructura

Esta funcionalidad se compone de múltiples vistas (ver Figura 18), las principales están relacionadas con la gestión de los sensores instalados en la estructura. Desde esta vista puede accederse con mayor detalle a la ubicación, configuración, lecturas registradas, historial de estado e historial de calibraciones de cada sensor instalado en algún instante de tiempo (primera imagen). La plataforma además permite a un usuario con privilegios de administrador el añadir sensores a la plataforma (segunda imagen). También está disponible la vista resumen de los DAQs instalados sobre la estructura (tercera imagen), en particular su ubicación, estado más reciente y número de canales disponibles. La funcionalidad se encuentra asociada a los requerimientos RV6, RV7 y RP1.

The figure consists of three screenshots of a web application interface for managing sensors and DAQs on the Llacolén bridge. Each screenshot has a top navigation bar with the SHM logo, 'Perfil', 'Mis sensores', 'Mis informes', and 'Cerrar sesión'.

Top Screenshot: Sensores de puente Llacolén

This screen shows a table of installed sensors with buttons for 'Añadir nuevo sensor' and 'Definir grupo de sensores'. The table lists sensors with their names, zones, frequencies, and links to 'Lecturas', 'Historial', and 'Calibraciones'.

| Nombre | Zona | Frecuencia | Lecturas | Historial | Calibraciones |
|--------------|--------------------|------------|----------|-----------|---------------|
| Acelerometro | Tablero del puente | 100 Hz | | | |
| Acelerometro | Estribo Norte | 200 Hz | | | |
| LDVT | Estribo Sur | 10 Hz | | | |
| Strain Gauge | Estribo Norte | 20 Hz | | | |

Middle Screenshot: Añadir sensor en puente Llacolén

This screen shows a form to add a new sensor. It includes dropdown menus for 'Zona a instalar', 'Conexiones disponibles (DAQ, Canal)', and 'Tipo de sensor', a text input for 'Frecuencia', and a 'Crear conexión' button.

Bottom Screenshot: DAQs de puente Llacolén

This screen shows a table of installed DAQs with a button for 'Añadir nuevo DAQ'. The table lists DAQs with their IDs, latest states, dates, number of channels, and characteristics.

| ID | Estado más reciente | Fecha | Nº de canales | Características |
|----|------------------------|---------------------|---------------|-----------------|
| 1 | Conectado | 2013-01-01 00:00:00 | 12 | DAQ N°1 |
| 2 | Conexión reestablecida | 2019-12-12 11:39:13 | 13 | DAQ N°2 |
| 3 | Conectado | 2014-02-03 00:00:00 | 14 | DAQ Final |
| 10 | Conectado | 2020-01-03 14:26:30 | 2 | Prueba DAQ |

Figura 18: Capturas de pantalla de la gestión de sensores instalados y DAQs en una estructura.

5.3.4 Gestión de monitoreo visual y grupos de sensores

La funcionalidad que podemos observar en la Figura 19, tiene como principal objetivo el otorgar acceso a los informes de monitoreo visual (creados por inspectores en terreno) y sus hallazgos asociados. Cualquier usuario puede acceder a los informes de una estructura, pero la gestión de estos (editar o eliminar un informe) sólo es accesible para el usuario que subió dicho informe a la plataforma. Esta funcionalidad se encuentra asociada a los requerimientos RV8, RV9 y RP5.

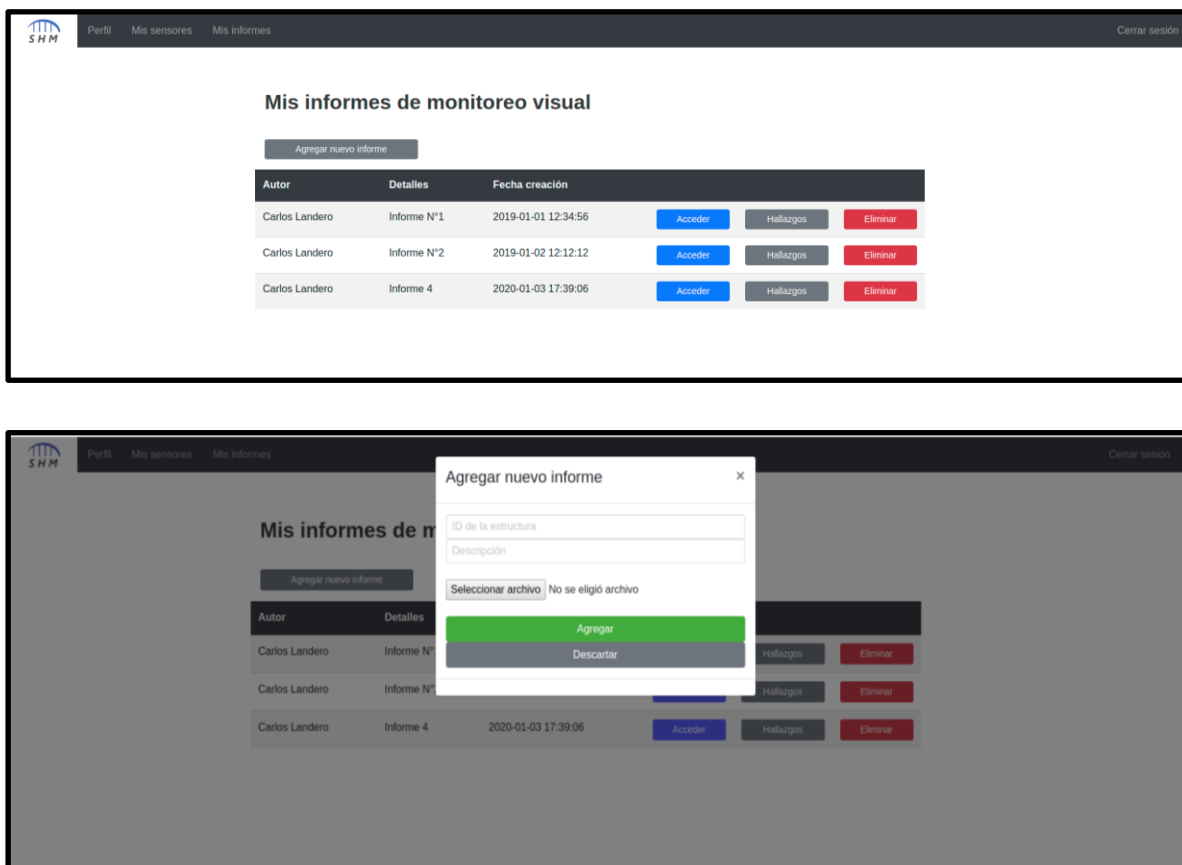


Figura 19: Capturas de pantalla de la gestión de informes de monitoreo por parte de un usuario (primera imagen), y la subida de un nuevo informe (segunda imagen).

5.3.5 Gestión de grupos de sensores definidos por el usuario

Como se observa en la Figura 20, en caso de que un usuario desee agrupar la información de sensores, la plataforma le permite desde su perfil acceder al listado de grupos que ha creado (primera imagen), desde donde puede crear, modificar o eliminar grupos (segunda y tercera imagen). De esta forma, estos listados de sensores quedan disponibles para que el usuario acceda a estos en cualquier momento. Esta funcionalidad se relaciona con el requerimiento RP6.

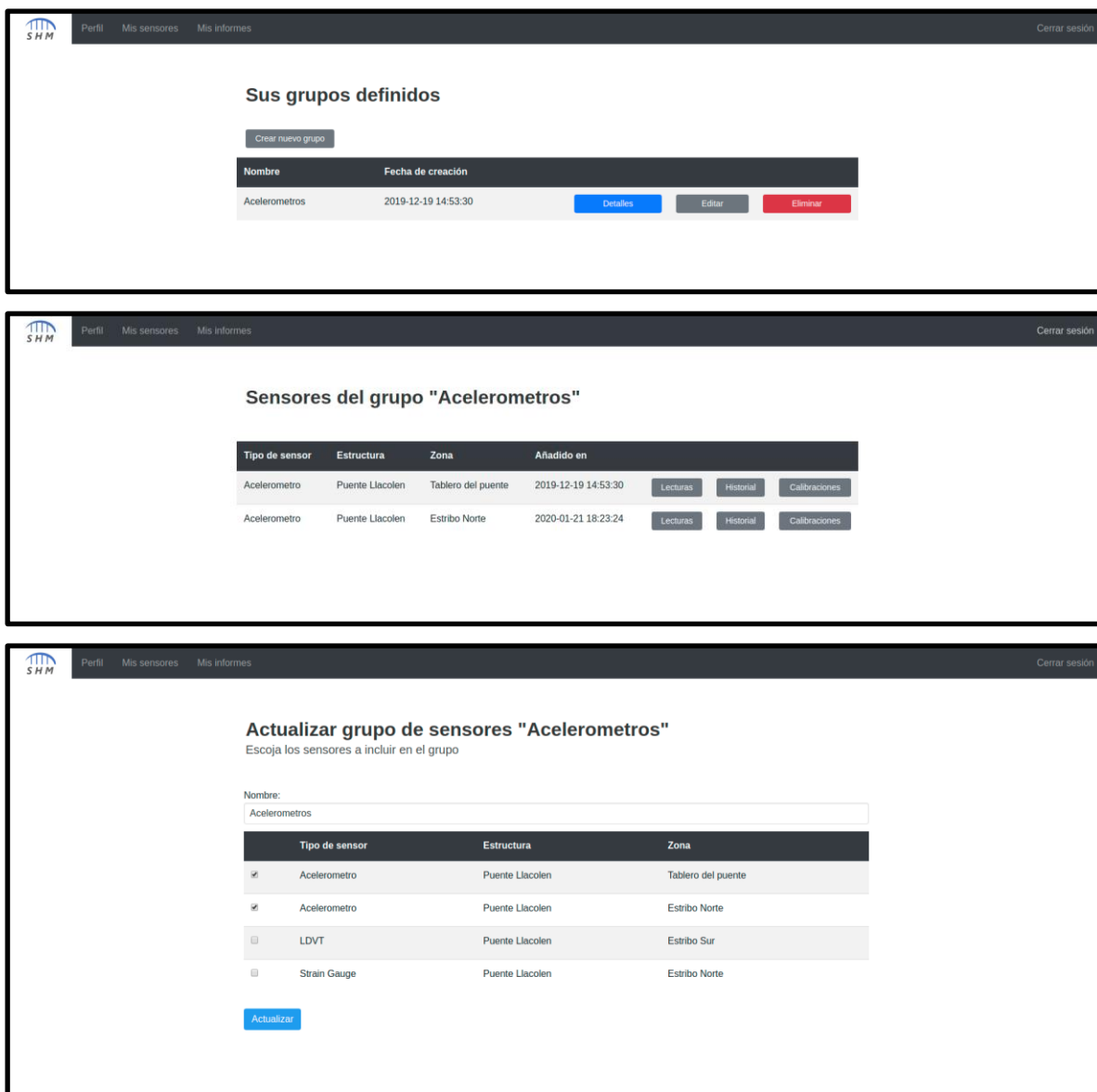


Figura 20: Capturas de pantalla de un usuario accediendo al listado de grupos que él ha creado, el listado de sensores en el grupo, y la vista para editarlo.

5.4 Consideraciones adicionales de implementación

5.4.1 Autogeneración de tablas y esquemas

Una de las características que posee la plataforma para adaptarse a un incremento del número de estructuras monitoreadas, es la generación automática de “schemas” y tablas necesarias para su funcionamiento.

Dado que en la plataforma pueden visualizarse los resúmenes de estructuras tanto monitoreadas como no monitoreadas, cuando un administrador requiera “iniciar” el monitoreo de una estructura, se ejecuta un proceso en el back-end que crea el “schema” o “keyspace” dedicado al monitoreo de esa estructura (ver Figura 21, primer diagrama).

Al mismo tiempo, cuando se instala un nuevo sensor en la estructura, la plataforma verifica que el “schema” o “keyspace” del puente ya exista (ver Figura 21, segundo diagrama), para entonces proceder a crear la tabla que almacene las lecturas de ese sensor. Esto también aplica para los cambios en la instalación de un sensor, dado que, al ser ubicado en una sección diferente del puente, las lecturas que registre no estarán bajo el mismo contexto que las lecturas anteriores, la plataforma genera una nueva tabla, para almacenar las lecturas del sensor en su nueva ubicación.

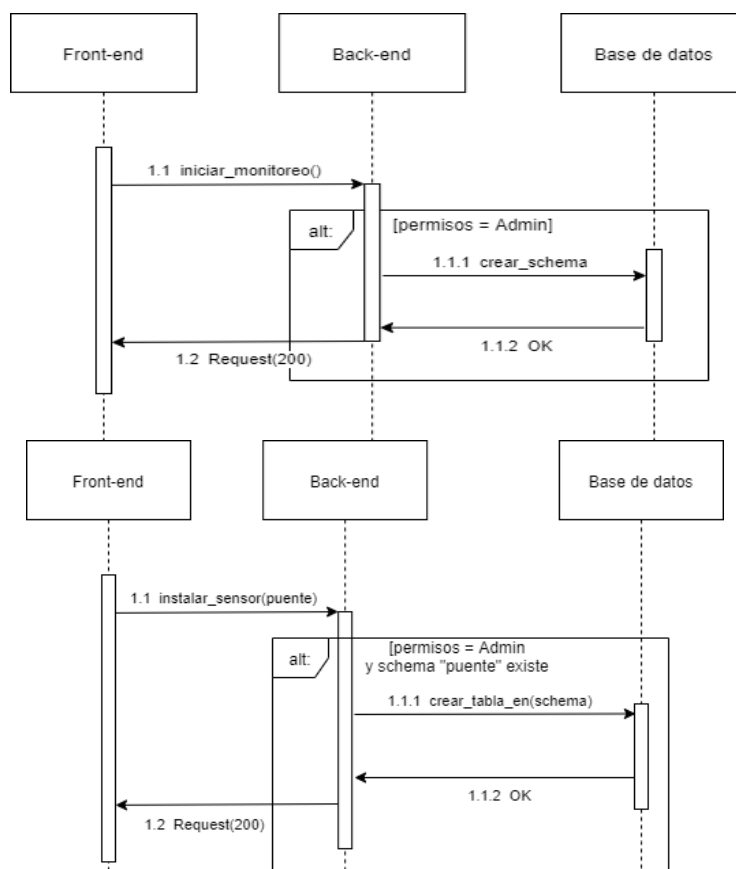


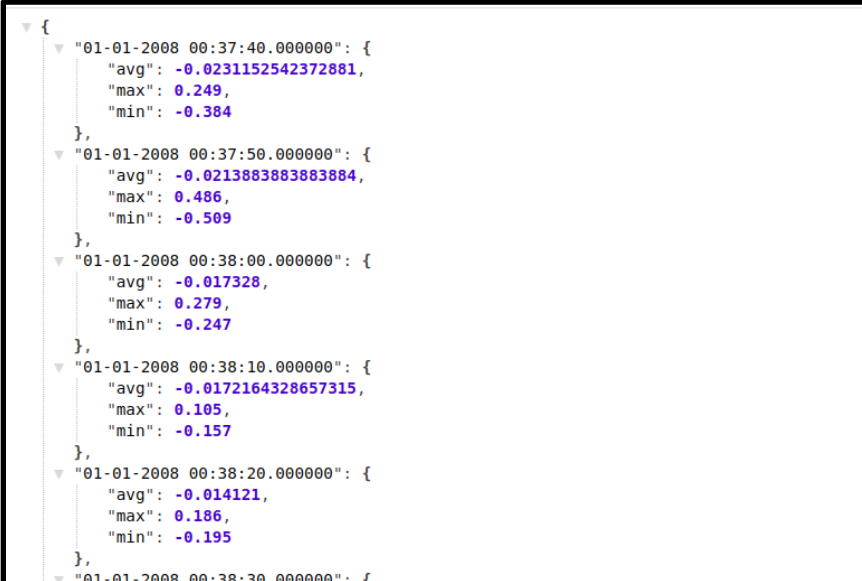
Figura 21: Diagramas de secuencia con los casos de éxito, tanto para iniciar el monitoreo de una estructura, como para instalar un sensor en esta respectivamente.

5.4.2 Visualización de lecturas en base de datos

Para prestar servicios a los componentes de visualización, la plataforma posee una API que entrega las lecturas de un sensor en particular, usando el formato JSON como estándar en todos los demás módulos de la plataforma. Esta característica se asocia con el requerimiento RS1.

Para utilizar la API en este prototipo, se deben otorgar dos parámetros, el primero corresponde al id del sensor instalado en la base de datos, y la longitud del periodo de tiempo para las lecturas (ya sea una hora, un día o una semana). Cabe destacar que el segundo parámetro es opcional, si se requiere de todas las lecturas del sensor, sólo se otorga el primer parámetro.

Para facilitar la generación de vistas y gráficas en los demás módulos, los datos son retornados utilizando “time buckets”, ventanas de tiempo donde se obtienen los datos más representativos, como, por ejemplo, en la Figura 22 tenemos las lecturas máximas, mínimas y promedio cada 10 segundos. Esto se lleva a cabo tanto desde el back-end de la plataforma al usar Cassandra, o bien de forma nativa en TimescaleDB [15].



```
{
  "01-01-2008 00:37:40.000000": {
    "avg": -0.0231152542372881,
    "max": 0.249,
    "min": -0.384
  },
  "01-01-2008 00:37:50.000000": {
    "avg": -0.0213883883883884,
    "max": 0.486,
    "min": -0.509
  },
  "01-01-2008 00:38:00.000000": {
    "avg": -0.017328,
    "max": 0.279,
    "min": -0.247
  },
  "01-01-2008 00:38:10.000000": {
    "avg": -0.0172164328657315,
    "max": 0.105,
    "min": -0.157
  },
  "01-01-2008 00:38:20.000000": {
    "avg": -0.014121,
    "max": 0.186,
    "min": -0.195
  },
  "01-01-2008 00:38:30.000000": {
    "avg": -0.014121,
    "max": 0.186,
    "min": -0.195
  }
}
```

Figura 22: Captura de pantalla con los resultados obtenidos desde la API, a ser utilizados por otros módulos de la plataforma.

6. Evaluación de plataforma para modelo

Para decidir qué motor de base de datos utilizar en la plataforma SHM, se llevó a cabo una evaluación de Cassandra y TimescaleDB en el contexto de la plataforma. Se definió una serie de criterios a aplicar sobre ambos gestores de bases de datos, con el objetivo de llegar a una conclusión sobre cuál gestor va a ser utilizado, los criterios a aplicar son los siguientes:

1. Escalabilidad a medida que crezcan las necesidades de almacenamiento.
2. Experiencia previa de usuarios y “stakeholders” con ambos gestores.
3. Flexibilidad del lenguaje de consulta.
4. Capacidad para ejercer criterios de consistencia sobre los datos.
5. Desempeño acorde a los requerimientos de lectura y escritura de datos.

6.1 Escalabilidad

En cuanto a qué tan escalables son los gestores de bases de datos candidatos, tenemos que Cassandra, por medio de su particionamiento de datos, abstracción de nodos virtuales, y replicación entre nodos, ofrece grandes posibilidades para escalar con tan sólo añadir otro nodo al clúster, aumentando la capacidad de escritura de forma sub lineal. Mientras que TimescaleDB, sólo es capaz de escalar el flujo de lecturas mediante el uso del “streaming replication” en PostgreSQL, donde cada nodo réplica se puede usar para lecturas y así aumentar el rendimiento.

6.2 Experiencia previa

Respecto a la experiencia previa del equipo de desarrollo de la plataforma, stakeholders del proyecto y personal del MOP y la dirección de vialidad, TimescaleDB posee mayor soporte que Cassandra. Dado que sus sistemas actuales ya utilizan este gestor de bases de datos, y su personal posee mayor experiencia en esta herramienta, utilizar TimescaleDB facilita una futura integración de la plataforma a los demás sistemas del Ministerio, y su uso conlleva una curva de aprendizaje menos pronunciada que la de Cassandra.

6.3 Flexibilidad y consistencia

Para evaluar la flexibilidad de los candidatos, se tiene en consideración que el lenguaje de consulta para Cassandra (dada su naturaleza NoSQL) posee las siguientes limitaciones: no permite realizar “JOIN” entre tablas, sólo permite aplicar “WHERE” a las Partition y/o Clustering Keys (para evitar realizar un escaneo de toda la base de datos distribuida) y no posee integridad referencial (no se pueden aplicar “foreign keys”). A diferencia de TimescaleDB, que posee soporte completo de SQL. Estas diferencias conllevan a que realizar la misma consulta de formas distinta para cada candidato, teniendo que “desnormalizar” el modelo para aplicar las consultas.

En el aspecto de consistencia de los datos, mientras TimescaleDB escribe todos los datos a un sólo nodo primario, el cual luego replica los datos a otros nodos (“Streaming replication”), Cassandra fragmenta las escrituras a través del clúster sin replicar, a menos que el usuario lo defina (“replication factor”), por lo que ambos poseen herramientas para controlar la consistencia [7].

6.4 Desempeño

Las lecturas de los sensores representan la fuente de datos más significativa en la plataforma, por lo tanto, es imperativo obtener el mejor desempeño posible al manejar esta información. Para probar el rendimiento de ambos candidatos, se plantea realizar lecturas y escrituras de las lecturas disponibles en el dataset del puente chino.

Para estas pruebas se utilizó la simulación de streaming de datos utilizando un “broker” de Apache Kafka, con el fin de simular las condiciones de latencia que pueden darse en un ambiente real, donde estos nodos están físicamente distantes entre sí. Para las consultas se utilizaron las siguientes tablas, descritas en la Tabla 1.

| DBMS | Keyspace /Schema | Tablas |
|--------------------------|------------------|---|
| Cassandra | harbin | acelerometro_uniaxial(id integer, fecha timestamp, lectura float) PK: id CK: fecha, lectura |
| PostgreSQL / TimescaleDB | harbin | acelerometro_uniaxial(<u>fecha</u> timestamp, lectura float) PK: fecha |

Tabla 1: Tablas para almacenar y consultar las lecturas de un acelerómetro, tanto en Cassandra como TimescaleDB. En Cassandra, PK representa la “Partition Key” y CK es “Clustering Key”, mientras que en TimescaleDB PK es “Primary Key”.

La primera consulta involucra la inserción de una hora de monitoreo de un acelerómetro uniaxial con una frecuencia de 100 Hz (equivalente a 360.000 filas aproximadamente). Se aplicaron dos estrategias para insertar los datos, insertando de a un dato a la vez y por “batches” equivalentes a diez minutos de monitoreo cada uno, los resultados en promedio se muestran en la Tabla 2.

| | Sin Batch | | Con Batch | |
|-------------|------------------|----------------------|------------------|----------------------|
| | Tiempo total (s) | Tiempo por fila (ms) | Tiempo total (s) | Tiempo por fila (ms) |
| Cassandra | 28.220 | 0.43 | 2.547 | 0.04 |
| TimescaleDB | 6.613 | 0.10 | 1.188 | 0.02 |

Tabla 2: Tiempos promedio de ejecución para la primera consulta.

Como puede observarse en la Tabla 2, los tiempos de inserción (con y sin batch) son menores en TimescaleDB que en Cassandra. En efecto, no pudo realizarse la consulta en su plan original (utilizar un batch de 1 hora de monitoreo) debido a que el límite de batch para Cassandra es de 65535 filas, debido a que, al insertar de esta forma, el nodo coordinador debe acceder a varias particiones, generando congestión en dicho nodo [6].

De la misma forma, dado que los tiempos de inserción en Batch, son muy inferiores a los tiempos de inserción sin este, se opta por insertar las lecturas de sensores en batch cuyo tamaño sea la frecuencia del sensor. Así pues, se reduce la latencia de los datos al tiempo de transmisión desde la instalación en terreno a la plataforma.

La segunda consulta consiste en obtener todas las lecturas del mismo acelerómetro de la primera consulta. Se consultaron dos ventanas de tiempo, una hora y un día de monitoreo respectivamente, los resultados en promedio se muestran en la Tabla 3.

| | 1 hora de monitoreo | | 1 día de monitoreo | |
|-------------|---------------------|----------------------|--------------------|----------------------|
| | Tiempo total (s) | Tiempo por fila (ms) | Tiempo total (s) | Tiempo por fila (ms) |
| Cassandra | 0.046 | 0.0007 | 26.864 | 3.554 |
| TimescaleDB | 0.510 | 0.0078 | 15.833 | 1.832 |

Tabla 3: Tiempos promedio de ejecución para la segunda consulta.

Puede observarse en la tabla 3 que, aunque para una hora de monitoreo, el tiempo de ejecución de las consultas en Cassandra es menor que en TimescaleDB, esta tendencia se revierte al consultar por una mayor cantidad de datos.

6.5 Resultados de la evaluación

Tras realizar el análisis en base a los criterios determinados, resulta factible utilizar PostgreSQL junto con la extensión TimescaleDB como el gestor de bases de datos para la plataforma SHM, puesto que no sólo tiene mejor desempeño que Cassandra, sino que además es más flexible, y fácil de integrar a los sistemas ya existentes en el MOP.

7. Conclusiones

Tras examinar lo expuesto por el presente informe, se llega a la conclusión de que se cumplieron satisfactoriamente los objetivos propuestos, tanto generales como específicos. Los resultados más significativos son el modelo conceptual que estructura la información relevante al monitoreo de salud estructural de un puente, y el prototipo software que valida la estructuración de los datos, desde la representación de la estructura misma, la red de sensores, hasta la gestión del monitoreo visual. En cuanto a la comparativa entre los motores de bases de datos candidatos, se muestra que TimescaleDB es la opción a recomendar al equipo de desarrollo del proyecto, dado que presenta un mayor rendimiento, menor costo y mayor usabilidad que Cassandra.

Durante el transcurso de esta memoria, se aprendieron a usar herramientas y técnicas para el manejo de grandes volúmenes de información dentro de ambos candidatos, definiendo parámetros de evaluación que pueden ser aplicables a otros tipos de proyectos con requisitos de estructuración de datos similares. A esto se agrega el hecho de que el prototipo software que valida al modelo conceptual puede utilizarse como producto mínimo viable del proyecto, y sentar las bases al desarrollo de la siguiente iteración de la plataforma de monitoreo objetivo del proyecto FONDEF.

Como trabajo futuro, se plantea la implementación de los requerimientos que, aunque presentes en el modelo, no llegaron al prototipo (como la generación de reportes, etc.), la integración de los resultados de las demás memorias de título involucradas en el proyecto FONDEF (los cuales a su vez tributan a los mismos requerimientos) y la automatización de procesos de captura de información a través de la metodología BIM. Todo esto siguiendo el objetivo inicial, de que la estructuración de los datos facilite los procesos de visualización, exploración de la estructura y análisis en la plataforma de monitoreo estructural.

8. Bibliografía

- [1] Ji, Cun & Shao, Qingshi & Sun, Jiao & Liu, Shijun & Li, Pan & Wu, Lei & Yang, Chenglei. (2016). Device Data Ingestion for Industrial Big Data Platforms with a Case Study. Sensors (Basel, Switzerland). 16. 10.3390/s16030279.
- [2] Jeong, Seongwoon & Byun, Jaewook & Kim, Doh Kwan & Bae, In & Law, Kincho. (2015). A data management infrastructure for bridge monitoring. Proceedings of SPIE - The International Society for Optical Engineering. 9435. 10.1117/12.2177109.
- [3] A. Chebotko, A. Kashlev & S. Lu, "A Big Data Modeling Methodology for Apache Cassandra," 2015 IEEE International Congress on Big Data, New York, NY, 2015, pp. 238-245.
- [4] Li, Shunlong & Li, Hui & Yang, Liu & Lan, Chengming & Zhou, Wensong & Ou, Jinping. (2014). SMC structural health monitoring benchmark problem using monitored data from an actual cable-stayed bridge. Structural Control and Health Monitoring. 21. 10.1002/stc.1559.
- [5] Confluence, Cassandra Limitations.
<https://cwiki.apache.org/confluence/display/CASSANDRA2/CassandraLimitations>
- [6] Datastax, CQL limits.
https://docs.datastax.com/en/archived/cql/3.3/cql/cql_reference/refLimits.html
- [7] Lee Hampton (2018), Eye or the Tiger: Benchmarking Cassandra vs. TimescaleDB for time-series data.
<https://blog.timescale.com/blog/time-series-data-cassandra-vs-timescaledb-postgresql-7c2cc50a89ce/>
- [8] Farrar, Charles & Worden, Keith. (2013). Structural Health Monitoring A Machine Learning Perspective. 10.1002/9781118443118.
- [9] Sacks, Rafael & Eastman, Charles & Lee, Ghang & Teicholz, Paul. (2018). BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers. 10.1002/9781119287568.
- [10] Tim Horgas, Benchmarking of Big Data Architecture Trade-Offs
- [11] Raúl Estrada. 2016. Fast Data Processing Systems with SMACK Stack. Packt Publishing.

- [12] Johann Llanos (2018), Implementación de un sistema de almacenamiento de datos masivos para monitoreo estructural.
- [13] DB-Engines, System Properties Comparison Cassandra vs. TimescaleDB <https://db-engines.com/en/system/Cassandra%3BTimescaleDB>
- [14] Ministerio de Obras Públicas, Red Caminera de Chile <http://www.mapas.mop.cl/>
- [15] TimescaleDB API Reference <https://docs.timescale.com/latest/api>
- [16] Cremona, Christian. (2016). Big Data and Structural Health Monitoring.
- [17] MQTT Protocol. <https://mqtt.org/>

9. Anexos

9.1 Listado requerimientos funcionales de la plataforma SHM.

| Código Req. | Significado |
|-------------|--|
| RM | Requerimiento de administración de la plataforma |
| RV | Requerimiento asociado a la visualización |
| RP | Requerimiento asociado al procesamiento |
| RS | Requerimiento asociado al almacenamiento |
| RA | Requerimiento asociado al análisis |

| Proceso | Funcionalidad | Código Req. | Descripción |
|--|--|-------------|--|
| Autenticación | Acceso con cuenta (usuario y contraseña) | RM1 | El usuario puede acceder sólo con cuenta registrada. |
| | Administración de usuarios | RM2 | Crear/Ver/Actualizar/Borrar cuentas de usuario y sus datos asociados (Nombre, Apellido, email, teléfono, cargo, descripción cargo, otros) y perfil de gestión de la cuenta. |
| Autorización | Definición de perfiles de usuario | RM3 | Preliminarmente los perfiles serán: Administrador, Analista, Dueño y Visita. |
| | Definición roles de gestión por perfil | | Administración rol, responsabilidad y actividades asociadas a cada perfil de usuario. |
| Asignación de funcionalidades y vistas | Permisos y vistas diferenciadas según perfil de usuario. | RM4 | Administrador: Acceso general, configuración y administración de todos los componentes del sistema. |
| | | | Analista: Administración de categorías globales, hallazgos. Configuración de dispositivos y señales. Descarga de datos. |
| | | | Dueño: Acceso de visualización a resultados diagnóstico, reportes monitoreo, inventario y condición de componentes. |
| | | | Visita: Acceso sólo lectura de información general de estructuras. Enfoque a difusión de trabajo y resultados públicos. |
| Gestión de puentes | Mostrar resumen de estructuras | RV1 | Desplegar información general asociada a cada puente, con los datos: nombre de estructura, mandante/propietario, condición global, estado de monitoreo e indicadores de gestión. |

| | | | |
|---|---|-----|--|
| | Mostrar información de inventario de las estructuras | RV2 | Desplegar información de inventario de puentes: nombre puente, rol, nombre camino, cauce de agua, provincia, región, largo, ancho, total, tipo activo, material estribo, material cepas, número de cepas, material piso, material vigas. |
| | Administrar inventario de estructuras. | RP1 | Crear/Ver/Actualizar/Borrar estructuras del conjunto manejado por la plataforma y sus datos asociados. |
| Monitoreo de salud estructural de puentes | Mostrar información general de monitoreo para las estructuras | RV3 | Listar las estructuras monitoreadas por nombre, condición global de monitoreo (estable, anormal, crítico, etc), coordenadas y descripción. |
| | Administrar condiciones estructurales | RP2 | Permite gestionar los estados de cada estructura y administrar los tipos de estado estructural global existentes, con su respectiva descripción, datos y condiciones asociadas. |
| | Desplegar información de una estructura seleccionada | RV4 | Para una estructura seleccionada, mostrar estado global, stream de video, diagramas descriptivos (sistemas global y local), número de sensores, tipos de sensores e imágenes complementarias con descripción. |
| | Visualización 2D/3D interactiva de la estructura | RV5 | Los usuarios de la plataforma podrán interactuar con una visualización de la estructura sensorizada en formatos 2D y 3D, para poder conocer la localización geoespacial de cada sensor, y favorecer el análisis de anomalías locales y globales. |
| | Transmisión de datos de los sensores | RS1 | Permite al usuario seleccionar grupo de sensores para añadir a una lista de visualización de sus respectivos stream de datos, estados de operación (encendido o apagado / mal funcionamiento o mucho ruido presente). |
| | Desplegar información de sensor detallada | RV6 | Para un sensor seleccionado, mostrar coordenadas en estructura, umbrales de operación correctos, dispositivo adquirente asociado y respectivos sensores vecinos. |
| | Configurar bandas de normalidad para sensor | RP3 | El usuario define y administra los rangos de operación normal para los sensores. |
| | Descarga de datos para sensor(es) seleccionado(s) | RP4 | Se permite al usuario descargar un paquete de datos en formato estándar para uno o múltiples sensores, en un rango de tiempo seleccionado. Puede ser de datos crudos o datos derivados/procesados. |
| | Análisis modal de la estructura | RA1 | Se deben procesar datos estadísticos, extraer características modales e identificar modos de vibrar. |

| | | | |
|------------------------------------|---|-----|---|
| | Repositorio de informes de estudio especiales | RP5 | Se permite a los usuarios especializados subir a la plataforma informes del monitoreo sobre la estructura y la descarga de éstos por usuarios admitidos. |
| Gestión de dispositivos | Administración de inventario de equipos | RP6 | Para una estructura seleccionada, se permite crear/ver/actualizar/borrar la información de los dispositivos (sensores, DAQ, etc.) presentes, sus dispositivos asociados, modelo, marca, descripción, especificaciones técnicas, etc. |
| | Visualización y descarga de fichas técnicas de sensores | RV7 | Los usuarios podrán descargar y/o acceder a la información de la ficha técnica de cada sensor, con su respectiva marca, modelo, canal DAQ, unidad de medida, frecuencia de muestreo, etc. |
| Gestión de hallazgos estructurales | Generar reporte de hallazgos estructurales | RA2 | Permite al inspector en terreno realizar un reporte de un hallazgo estructural detectado, con los datos de identificación de usuario (nombre, cargo, identificación, etc.) y del reporte (fecha, hora, estructura, etc.) información geoespacial, fotografías y la opción de audio o texto descriptivo. |
| | Visualizar hallazgos en la estructura | RV8 | En la vista de visualización 2D/3D para la estructura, permite ver la ubicación y severidad de los hallazgos presentes, con la posibilidad de despliegue de información a pedido. |
| | Ver seguimiento de un hallazgo | RV9 | Desplegar una línea de tiempo del hallazgo seleccionado, con la detección, seguimiento, mantención/repación y características detectadas en cada nueva inspección. |

9.2 Resumen de las entidades presentes en la plataforma SHM.

| Nombre | Descripción |
|-----------------------|--|
| Estructura | Representa la obra civil a monitorear, en particular, a los puentes. |
| Zona estructura | Hace referencia a las partes que juntas componen una estructura en concreto, se clasifican según tipos. |
| Tipo de zona | Son las clasificaciones de las zonas que componen a una estructura, en el caso de los puentes tenemos: Tablero, Viga, Armadura, Fundación, Estribo, Estructura portante, Pilar, etc. |
| Sensor | Dispositivo que mide una o varias características de la salud en una estructura, se clasifican en tipos de sensores. |
| Tipo de sensor | Clasificaciones de los sensores, en el caso de la plataforma se consideran los siguientes tipos: Acelerómetro, Strain Gauge, Weather Station, LVDT, etc. |
| Sensor Instalado | Representa cuando un sensor está instalado en una sección de un puente, desde una fecha determinada. Está pensado para modelar que un sensor podría cambiar de ubicación, y por tanto, el contexto de sus datos es distinto a la instancia previa. |
| Instalación de Sensor | Representa el instante cuando se instala un nuevo sensor, o se cambia de ubicación a un sensor instalado previamente. |
| Estado de sensor | Indica el estado actual del monitoreo de un sensor instalado, el cual puede ser: Activo, Presenta fallas, Apagado, etc. |
| Calibración de sensor | Registra los instantes cuando un sensor instalado es calibrado, con el fin de mantener un registro histórico. |
| Descripción de sensor | Detalla las características de un sensor instalado. |
| Conjunto | Modela un “clúster” o grupo de sensores instalados, cercanos entre sí en lugares estratégicos, por ejemplo, en las intersecciones de dos o más zonas de la estructura. |
| DAQ | O “Data Acquisitor”, es un dispositivo usado para tomar las muestras analógicas de los sensores instalados, y los transforma a señales digitales, cada uno posee varios canales desde los cuales los sensores se conectan. |
| Descripción de DAQ | Detalla las características de un DAQ. |

| | |
|---|---|
| Revisión de DAQ | Registra los instantes cuando un DAQ es revisado manualmente por personal autorizado, con el fin de mantener un registro histórico. |
| Estado de DAQ | Indica el estado actual del DAQ, el cual puede ser: Activo, Presenta fallas, Apagado, etc. |
| Canal | Representa a los canales que un DAQ posee para recibir datos desde sensores. |
| Conexión actual | Representa el/los canales de un DAQ que están siendo usados por un sensor instalado actualmente, puede ser un solo canal (sensor de un eje, o uniaxial) o múltiples ejes (sensores de múltiples ejes). |
| Conexión pasada | Mantiene un registro histórico de que canales fueron usados por qué sensor instalado, desde una fecha de inicio hasta una fecha de término. |
| Cámara de monitoreo | Representa las cámaras de monitoreo a instalar sobre una estructura. |
| Usuario | Corresponden a los usuarios de la plataforma SHM, con sus diferentes niveles de permisos: Administrador, Analista, Dueño y Visita. |
| Informe monitoreo visual | Consiste en los informes de monitoreo visual generados por los inspectores en terreno, los cuales pueden ser subidos a la plataforma como evidencia de inspección visual. |
| Hallazgo visual | Anomalías a la salud de la estructura, visibles al ojo del inspector en terreno y que respaldan el contenido de un informe de monitoreo visual. |
| Grupo definido por usuario | Grupo de sensores asociados por un usuario autorizado de la plataforma con fines de visualización y/o análisis. Estos sensores pueden estar instalados en distintos puentes, distintas zonas de un puente, o haber sido instalados en distintos periodos de tiempo. |
| Imagen estructura, Material audiovisual y Visualización BIM | Clases que representan los archivos cuyo fin es el apoyo audiovisual de la plataforma. |
| Alertas | Clase que representa las alertas detectadas por la plataforma, ya sea respecto a una zona, sensor, o la estructura. |
| Reportes de monitoreo | Clase que representa el “estado de vida” de la estructura desde que comenzó a ser monitoreada, esto incluye el historial de estado de la estructura, las alertas y estados de sus componentes durante un periodo de tiempo. |

9.3 Diagrama de clases plataforma SHM

