**RESUMEN EJECUTIVO**

**Descripción breve del proyecto**

El proyecto de sistema de monitoreo y telemetría para la empresa "Buses Seguros S.A." tiene como objetivo implementar una solución tecnológica avanzada que mejore la seguridad y la eficiencia operativa de la flota de autobuses. Cada autobús estará equipado con una serie de sensores y un cluster Kubernetes que recolectará, procesará y transmitirá datos en tiempo real. Estos datos se utilizarán para monitorear el estado del autobús, generar alertas en caso de irregularidades, y almacenar información relevante para su análisis posterior.

**Objetivos principales**

1. **Monitoreo en Tiempo Real:** Capturar y procesar datos de sensores en tiempo real para proporcionar una visión actualizada de las condiciones y el rendimiento de cada autobús, permitiendo una rápida respuesta ante posibles problemas.
2. **Almacenamiento y Transmisión de Datos Confiable:** Utilizar colas de mensajería como RabbitMQ o Kafka para asegurar que todos los datos de los sensores se almacenen temporalmente y se transmitan de manera confiable a la central, minimizando la pérdida de información.
3. **Escalabilidad y Flexibilidad:** Implementar un cluster Kubernetes en cada autobús para gestionar los contenedores que procesan los datos de los sensores, permitiendo la fácil adición de nuevos sensores o la modificación de los existentes sin interrumpir el sistema.
4. **Integración Eficiente con la Central:** Transmitir los datos recopilados en formato JSON a los servicios de recepción en la central cuando los autobuses arriben a las terminales, asegurando una integración fluida y eficiente con los sistemas centrales.
5. **Análisis y Almacenamiento de Datos:** Procesar y almacenar los datos recibidos en bases de datos en la central para su posterior análisis, permitiendo la generación de informes y la toma de decisiones basadas en datos históricos y en tiempo real.
6. **Generación de Alertas:** Monitorear la velocidad del autobús en tiempo real y generar alertas automáticas si se superan las restricciones locales, mejorando la seguridad y el cumplimiento de las normativas.
7. **Optimización de Recursos:** Utilizar contenedores Docker y clusters Kubernetes para optimizar el uso de recursos en cada autobús, asegurando que el sistema sea eficiente y tenga un bajo impacto en el rendimiento del autobús.
8. **Mantenimiento y Actualización Fácil:** Facilitar el mantenimiento y la actualización del sistema mediante el uso de tecnologías de contenedorización y orquestación (Docker y Kubernetes), permitiendo desplegar nuevas versiones de software sin interrupciones significativas.
9. **Seguridad de los Datos:** Asegurar que los datos transmitidos desde los autobuses a la central estén protegidos mediante encriptación y mecanismos de autenticación, garantizando la privacidad y la integridad de la información.
10. **Cumplimiento Normativo:** Cumplir con las regulaciones locales e internacionales respecto a la operación de vehículos de transporte, especialmente en lo que se refiere a la velocidad y las condiciones de operación, ayudando a la empresa a evitar multas y mejorar la seguridad en las carreteras.

**Beneficios esperados**

* **Mejora en la seguridad y eficiencia:** El monitoreo constante y en tiempo real de las condiciones de los autobuses permitirá prevenir incidentes y optimizar el uso de recursos, asegurando que los autobuses operen en condiciones óptimas y seguras.
* **Reducción de costos operativos:** Al obtener datos precisos y en tiempo real sobre el estado de los autobuses, se podrá planificar mejor el mantenimiento y evitar reparaciones costosas no planificadas. Esto también reducirá el tiempo de inactividad de los autobuses.
* **Cumplimiento regulatorio:** El sistema ayudará a la empresa a cumplir con las normativas de seguridad y operación de los vehículos, evitando multas y sanciones. Las alertas automáticas en caso de exceso de velocidad contribuirán a mejorar la seguridad vial.
* **Mejora en la toma de decisiones:** La capacidad de almacenar y analizar datos históricos y en tiempo real permitirá a la empresa tomar decisiones informadas y estratégicas, basadas en un análisis exhaustivo de la información recopilada.
* **Optimización de la flota:** La información precisa y en tiempo real sobre el estado de cada autobús permitirá una mejor gestión de la flota, optimizando las rutas y mejorando la eficiencia operativa.

**Alcance del proyecto**

* **Implementación de sensores y clusters Kubernetes en cada autobús:** Cada autobús será equipado con sensores para monitorear diversos aspectos (combustible, presión de aire, ambiente, temperatura, velocidad y GPS) y un cluster Kubernetes para gestionar los contenedores que procesan estos datos.
* **Desarrollo e implementación de pods para cada tipo de sensor y un pod colector de datos:** Cada sensor tendrá su propio pod dedicado, y habrá un pod central que recolectará y procesará los datos de todos los sensores.
* **Configuración de colas de mensajería para el almacenamiento temporal de datos:** Se utilizarán colas de mensajería como RabbitMQ o Kafka para asegurar que los datos se almacenen temporalmente y se transmitan de manera confiable a la central.
* **Desarrollo de API Gateway y microservicios en la central para recepción, procesamiento y almacenamiento de datos:** En la central, se implementará un API Gateway para recibir los datos y varios microservicios para procesarlos y almacenarlos en bases de datos.
* **Implementación de un sistema de monitoreo en tiempo real y generación de alertas:** Se desarrollará un microservicio dedicado para monitorear la velocidad de los autobuses y generar alertas automáticas en caso de exceso de velocidad.

**INTRODUCCIÓN**

**Contexto y antecedentes**

"Buses Seguros S.A." es una empresa de transporte de pasajeros que opera una flota de autobuses en diversas rutas. La empresa busca mejorar la seguridad y la eficiencia operativa de su flota mediante la implementación de un sistema avanzado de monitoreo y telemetría. Actualmente, la empresa enfrenta desafíos en la gestión y mantenimiento de su flota debido a la falta de información en tiempo real sobre el estado de los autobuses. Implementar un sistema de monitoreo y telemetría permitirá a la empresa obtener datos precisos y actualizados, mejorar la seguridad vial, reducir costos operativos y cumplir con las normativas de transporte.

**Propósito del documento**

El propósito de este documento es proporcionar una descripción detallada de la arquitectura del sistema de monitoreo y telemetría para "Buses Seguros S.A.". Este documento está diseñado para servir como una guía completa para el diseño, desarrollo e implementación del sistema. Incluye la estructura de los componentes del sistema, los procesos de recopilación y transmisión de datos, y las tecnologías utilizadas para asegurar la confiabilidad y eficiencia del sistema. Además, el documento aborda los requisitos de seguridad, mantenimiento y cumplimiento normativo.

**Público objetivo**

Este documento está dirigido a:

* **Ingenieros de software:** Que participarán en el diseño, desarrollo e implementación del sistema. Este grupo necesita entender la arquitectura técnica y los componentes específicos del sistema para asegurar una implementación exitosa.
* **Gerentes de proyecto:** Que supervisarán el progreso del proyecto y asegurarán que se cumplan los objetivos. Este grupo necesita una comprensión clara de los objetivos del proyecto, los beneficios esperados y los hitos clave para gestionar el proyecto de manera efectiva.
* **Stakeholders y directivos de la empresa:** Que necesitan entender los beneficios y el impacto del sistema en la operación de la flota. Este grupo requiere información detallada sobre cómo el sistema mejorará la seguridad y la eficiencia operativa, así como los beneficios financieros y de cumplimiento regulatorio.
* **Especialistas en seguridad y cumplimiento:** Que garantizarán que el sistema cumpla con las normativas y regulaciones pertinentes. Este grupo necesita asegurarse de que el sistema implemente todas las medidas de seguridad necesarias y cumpla con las regulaciones locales e internacionales.

**Stakeholders**

|  |  |  |
| --- | --- | --- |
| Stakeholders | Descripción | Expectativas |
| Equipo de TI | Responsable de la infraestructura tecnológica y del soporte técnico. | Sistema confiable y fácil de mantener, con alta disponibilidad y un rendimiento óptimo. |
| Equipo de Seguridad de TI | Encargado de la seguridad de la información y de las comunicaciones. | Datos protegidos mediante encriptación y autenticación, sin brechas de seguridad. |
| Equipo de Desarrollo de Software | Desarrolladores y arquitectos de software encargados de la implementación del sistema. | Código eficiente, escalable y fácil de actualizar, con mínima interrupción del servicio durante despliegues. |
| Equipo de Operaciones | Monitorea la operación diaria de la flota de autobuses. | Sistema de monitoreo en tiempo real preciso y confiable, con generación rápida de alertas. |
| Equipo de Mantenimiento | Responsable del mantenimiento del sistema y de los autobuses. | Procedimientos de mantenimiento y actualización del software que no interfieran con la operación de los autobuses. |
| Departamento Legal y de Cumplimiento | Asegura que la empresa cumpla con todas las regulaciones locales e internacionales. | Cumplimiento de todas las normativas, especialmente en cuanto a velocidad y condiciones de operación de los autobuses. |
| Equipo de Análisis de Datos | Analistas encargados de procesar y analizar los datos recolectados. | Acceso a datos históricos y en tiempo real de manera confiable para la generación de informes y toma de decisiones. |
| Operadores de la Central | Personal que gestiona el sistema de recepción y monitoreo de datos en la central. | Interfaz de usuario intuitiva y eficiente, que permita responder rápidamente a las alertas y gestionar datos recibidos. |
| Clientes/Pasajeros | Usuarios finales que utilizan el servicio de transporte. | Mayor seguridad y eficiencia en el servicio de transporte, con una experiencia de viaje cómoda y sin interrupciones. |
| Alta Gerencia | Directivos responsables de la toma de decisiones estratégicas en la empresa. | Sistema que mejore la seguridad y eficiencia operativa, proporcionando datos fiables para la toma de decisiones estratégicas. |

**REQUISITOS DEL SISTEMA**

**Requisitos funcionales**

|  |  |  |
| --- | --- | --- |
|  | **Requisitos Funcionales** | |
| **#** | **Nombre** | **Items** |
| **1** | **Monitoreo en Tiempo Real** | * Capturar y procesar datos de múltiples sensores en el autobús en tiempo real. * Proporcionar una visión actualizada de las condiciones y el rendimiento del autobús. |
| **2** | **Almacenamiento y Transmisión de Datos** | * Utilizar colas de mensajería (RabbitMQ o Kafka) para asegurar la captura y transmisión de todos los datos de los sensores. * Transmitir los datos recopilados en formato JSON a los servicios de recepción en la central cuando los autobuses lleguen a las terminales. |
| **3** | **Procesamiento de Datos** | * Centralizar la información de los sensores en el pod colector de datos. * Preparar los datos para transmisión cuando el autobús llega a una terminal. |
| **4** | **Generación de Alertas** | * Monitorear la velocidad del autobús en tiempo real. * Generar alertas automáticas si se superan las restricciones locales de velocidad. |
| **5** | **Análisis y Almacenamiento de Datos** | * Procesar y almacenar los datos recibidos en bases de datos en la central. * Permitir la generación de informes y la toma de decisiones basadas en datos históricos y en tiempo real. |

**Requisitos no funcionales**

|  |  |  |
| --- | --- | --- |
|  | **Requisitos No Funcionales** | |
| **#** | **Nombre** | **Items** |
| **1** | **Rendimiento** | * Procesamiento de datos en tiempo real con baja latencia. * Transmisión confiable de datos sin pérdida de información. |
| **2** | **Escalabilidad** | * Implementar un cluster Kubernetes en cada autobús para gestionar los contenedores. * Facilitar la adición de nuevos sensores o la modificación de los existentes sin interrumpir el sistema. |
| **3** | **Seguridad** | * Proteger los datos transmitidos desde los autobuses a la central mediante encriptación y mecanismos de autenticación. * Garantizar la privacidad y la integridad de la información. |
| **4** | **Mantenimiento y Actualización** | * Facilitar el mantenimiento y la actualización del sistema mediante el uso de tecnologías de contenedorización y orquestación (Docker y Kubernetes). * Desplegar nuevas versiones de software sin interrupciones significativas. |

**Restricciones y supuestos**

|  |  |  |
| --- | --- | --- |
|  | **Restricciones y supuestos** | |
| **#** | **Nombre** | **Items** |
| **1** | **Restricciones** | * Los autobuses deben estar equipados con un cluster Kubernetes y sensores específicos (combustible, presión de aire, ambiente, temperatura, velocidad, GPS). * La transmisión de datos a la central debe ocurrir únicamente cuando el autobús llegue a una terminal. |
| **2** | **Supuestos** | * Los autobuses tienen conectividad suficiente para transmitir los datos a la central en las terminales. * Los datos de los sensores se pueden capturar y procesar en tiempo real dentro del entorno del cluster Kubernetes en el autobús. * Los sistemas centrales están preparados para recibir, procesar y almacenar los datos transmitidos en formato JSON. |

**Requerimientos de calidad**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Eficiencia | |
| Tiempo | ID | Descripción | Prioridad |
| 1 Semana | 001 | Implementación de sensores en los autobuses | Alta |
| 2 Semanas | 002 | Configuración del cluster Kubernetes en cada autobús | Alta |
| 1 Semana | 003 | Integración de colas de mensajería (RabbitMQ/Kafka) | Media |
| 3 días | 004 | Transmisión de datos a la central en formato JSON | Alta |
| 1 Semana | 005 | Desarrollo del sistema de monitoreo en tiempo real | Alta |
| 2 Semanas | 006 | Implementación de microservicios en la central | Media |
| 3 Días | 007 | Configuración del API Gateway | Media |
| 1 Semana | 008 | Pruebas y validación del sistema completo | Alta |
| Recursos | ID | Descripción | Prioridad |
| Tiempo de respuesta | 1 | Medición del tiempo que tarda el sistema en responder a una solicitud. | Alta |
| Utilización de recursos | 2 | Evaluación de cómo el sistema utiliza los recursos disponibles (CPU, memoria). | Media |
| Capacidad | 3 | Capacidad del sistema para manejar grandes volúmenes de datos y usuarios. | Alta |
| Volumen de carga | 4 | Evaluación del rendimiento bajo condiciones de alta carga y estrés. | Media |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Fiabilidad | |
| Tolerancia a Fallas | ID | Descripción | Prioridad |
| Recuperabilidad | 001 | Capacidad del sistema para recuperarse de fallas y continuar operando. | Alta |
| Disponibilidad | 002 | Capacidad del sistema para estar operativo y accesible cuando se necesita. | Media |
| Redundancia | 003 | Implementación de componentes de respaldo que pueden tomar el lugar de los componentes fallidos. | Alta |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Mantenimiento | |
| Modificación | ID | Descripción | Prioridad |
| Facilidad de reparación | 1 | Capacidad del sistema para ser reparado rápidamente y con facilidad. | Alta |
| Disponibilidad de repuestos | 2 | Disponibilidad de piezas y componentes necesarios para el mantenimiento. | Media |
| Documentación técnica | 3 | Calidad y accesibilidad de la documentación técnica para el mantenimiento. | Alta |
| Capacitación del personal | 4 | Nivel de formación y capacitación del personal encargado del mantenimiento. | Media |
| Facilidad de Pruebas | ID | Descripción | Prioridad |
| Modularidad | 001 | La capacidad del sistema para ser dividido en módulos independientes. | Alta |
| Reusabilidad | 002 | La capacidad de utilizar componentes del sistema en otros contextos. | Media |
| Analizabilidad | 003 | La facilidad con la que se pueden identificar las causas de los defectos. | Alta |
| Modificabilidad | 004 | La facilidad con la que se pueden realizar cambios en el sistema. | Alta |
| Estabilidad | 005 | La capacidad del sistema para evitar efectos adversos debido a cambios. | Media |
| Testabilidad | 006 | La facilidad con la que se pueden realizar pruebas al sistema. | Alta |
| Estabilidad | ID | Descripción | Prioridad |
| Alta | 001 | Control de variación de materiales | Alta |
| Media | 002 | Eliminación de problemas en el proceso | Media |
| Alta | 003 | Capacidad del personal de mantenimiento | Alta |
| Baja | 004 | Reducción de errores y desperdicios | Baja |
| Flexibilidad | ID | Descripción | Prioridad |
| Alta | 001 | Capacidad de adaptarse a cambios en los requisitos de mantenimiento. | Alta |
| Media | 002 | Implementación de mejoras continuas en los procesos de mantenimiento. | Media |
| Baja | 003 | Estandarización de procedimientos para minimizar errores. | Baja |
| Escalabilidad | ID | Descripción | Prioridad |
| Alta | 001 | Capacidad del sistema para manejar un aumento en la carga de trabajo sin afectar el rendimiento. | Alta |
| Media | 002 | Adaptabilidad del sistema para incorporar nuevas funcionalidades sin necesidad de rediseño significativo. | Media |
| Alta | 003 | Facilidad de mantenimiento y actualización del sistema con mínimo tiempo de inactividad. | Alta |
| Baja | 004 | Capacidad del sistema para integrarse con otros sistemas y tecnologías existentes. | Baja |

**VISIÓN GENERAL DE LA ARQUITECTURA**

**Descripción general del sistema**

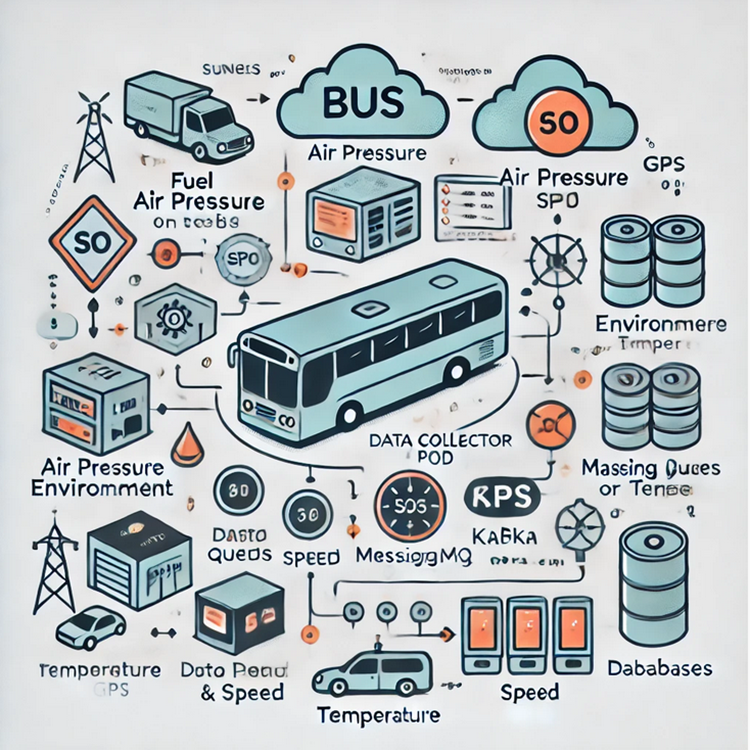
El sistema de monitoreo y telemetría para "Buses Seguros S.A." está diseñado para recopilar, procesar, almacenar y analizar datos en tiempo real provenientes de una variedad de sensores instalados en cada autobús de la flota. Cada autobús estará equipado con un cluster Kubernetes que gestionará múltiples pods, cada uno dedicado a un sensor específico, como combustible, presión de aire en las llantas, ambiente, temperatura, velocidad y GPS. Los datos capturados por estos sensores se procesarán localmente en el autobús y se almacenarán temporalmente en colas de mensajería (RabbitMQ o Kafka). Cuando el autobús llega a una terminal, los datos se transmitirán a la central de la empresa, donde se recibirán, procesarán y almacenarán para análisis posterior y generación de alertas.

El sistema incluye los siguientes componentes principales:

* **Sensores:** Dispositivos físicos instalados en cada autobús que recopilan datos específicos (combustible, presión de aire, ambiente, temperatura, velocidad y GPS).
* **Cluster Kubernetes en el Autobús:** Un cluster Kubernetes que ejecuta pods Docker para cada sensor, así como un pod colector de datos que centraliza la información de los sensores.
* **Colas de Mensajería:** Utilizadas para almacenar temporalmente los datos antes de su transmisión a la central.
* **API Gateway:** En la central, recibe las solicitudes de datos desde los autobuses.
* **Microservicios:** Procesan y almacenan los datos recibidos en bases de datos en la central.
* **Sistema de Monitoreo y Alertas:** Monitorea la velocidad del autobús en tiempo real y genera alertas automáticas si se superan las restricciones locales.

**Diagrama de Alto Nivel**

El siguiente diagrama de alto nivel ilustra la arquitectura propuesta para el sistema de monitoreo y telemetría de "Buses Seguros S.A.". Este diagrama destaca los componentes principales y su interacción desde la recolección de datos en el autobús hasta su procesamiento y almacenamiento en la central. La arquitectura está diseñada para garantizar la recolección confiable de datos, su procesamiento en tiempo real, y la generación de alertas automáticas para mejorar la seguridad y eficiencia operativa de la flota de autobuses. Los componentes clave incluyen sensores en el autobús, un cluster Kubernetes para la gestión de contenedores, colas de mensajería para el almacenamiento temporal de datos, un API Gateway para la recepción de datos, microservicios para el procesamiento y almacenamiento de información, y un sistema de monitoreo y alertas en tiempo real.



**Principios y patrones arquitectónicos utilizados**

1. **Desacoplamiento y Microservicios:**

* El sistema utiliza una arquitectura de microservicios para garantizar que cada función del sistema (recolección de datos, procesamiento, almacenamiento y monitoreo) esté desacoplada de las demás. Esto permite una mayor flexibilidad y escalabilidad, ya que los componentes pueden desarrollarse, desplegarse y mantenerse de manera independiente.

1. **Orquestación de Contenedores:**

* La utilización de Kubernetes para orquestar contenedores Docker en cada autobús permite una gestión eficiente de los recursos, facilita la implementación de actualizaciones y mejora la resiliencia del sistema ante fallos individuales.

1. **Patrón de Colas de Mensajería:**

* RabbitMQ o Kafka se utilizan para implementar colas de mensajería, lo que asegura que los datos se almacenen de manera temporal y se transmitan de forma confiable. Este patrón garantiza que ningún dato se pierda, incluso si hay problemas temporales de conectividad.

1. **Seguridad y Cumplimiento:**

* La arquitectura del sistema incorpora medidas de seguridad como encriptación y autenticación para proteger los datos durante la transmisión y almacenamiento. Además, se asegura el cumplimiento de las normativas locales e internacionales de transporte.

1. **Monitoreo y Alertas en Tiempo Real:**

* Un microservicio dedicado se encarga del monitoreo en tiempo real de la velocidad del autobús, utilizando los datos del GPS y del sensor de velocidad. En caso de que se superen las restricciones de velocidad, se generan alertas automáticas para notificar a los operadores.

1. **Escalabilidad y Flexibilidad:**

* La arquitectura basada en Kubernetes permite escalar el sistema fácilmente mediante la adición de nuevos pods para sensores adicionales o la modificación de los existentes sin interrumpir el servicio.

1. **Alta Disponibilidad y Resiliencia:**

* Al utilizar Kubernetes y microservicios, el sistema está diseñado para ser altamente disponible y resiliente. Los fallos en componentes individuales no afectan al sistema en su conjunto, y los contenedores pueden reiniciarse automáticamente en caso de fallos.

Esta arquitectura asegura que el sistema de monitoreo y telemetría para "Buses Seguros S.A." sea robusto, escalable, y capaz de proporcionar datos precisos y en tiempo real para mejorar la seguridad y eficiencia de la flota de autobuses.

**Discusiones de Arquitectura**

En el diseño del sistema de monitoreo y telemetría para la empresa de transporte "Buses Seguros S.A.", se han tomado una serie de decisiones arquitectónicas clave con el objetivo de asegurar la eficiencia, escalabilidad, fiabilidad y seguridad del sistema. Cada una de estas decisiones ha sido cuidadosamente considerada para cumplir con los objetivos específicos del proyecto y para enfrentar los desafíos únicos presentados por la operación de una flota de autobuses en tiempo real. A continuación, se detallan las decisiones arquitectónicas más relevantes, junto con sus respectivas justificaciones:

1. **Uso de Kubernetes en Cada Autobús**

**Descripción:** Cada autobús cuenta con un cluster Kubernetes para gestionar y orquestar los contenedores que ejecutan las aplicaciones de monitoreo y procesamiento de datos. Este cluster incluye varios pods, que son las unidades básicas de ejecución en Kubernetes, y gestionan la ejecución de contenedores Docker.

**Justificación:**

* **Escalabilidad y Flexibilidad:** Kubernetes permite gestionar múltiples instancias de aplicaciones (contenedores) en un solo nodo. Esto facilita la escalabilidad horizontal, permitiendo añadir más contenedores (por ejemplo, para nuevos tipos de sensores) sin necesidad de reconfigurar el sistema completo. Además, Kubernetes maneja la redistribución de contenedores en caso de fallos, asegurando que el sistema se mantenga operativo.
* **Orquestación y Gestión de Recursos:** Kubernetes proporciona un sistema de gestión de recursos eficiente, donde los contenedores se distribuyen según la carga y los recursos disponibles, evitando sobrecargas y asegurando un uso óptimo de los recursos del autobús. Esto es crucial para mantener el rendimiento del autobús mientras se ejecutan las aplicaciones de monitoreo.
* **Automatización de Despliegues:** Kubernetes soporta actualizaciones continuas y despliegues automatizados, lo que facilita la implementación de nuevas versiones de software sin interrumpir el funcionamiento del sistema.

1. **Contenedores Docker para los Sensores y el Colector de Datos**

**Descripción**: Se utilizan contenedores Docker para ejecutar las aplicaciones que procesan los datos de los sensores y el pod colector de datos. Docker proporciona un entorno de ejecución estandarizado para estas aplicaciones.

**Justificación:**

* **Aislamiento y Portabilidad:** Docker asegura que las aplicaciones se ejecuten de manera aislada, evitando conflictos entre diferentes aplicaciones o entre aplicaciones y el sistema operativo del autobús. Esto también facilita la portabilidad de las aplicaciones, permitiendo que se ejecuten en diferentes entornos (desarrollo, prueba, producción) sin necesidad de modificar el código.
* **Facilidad de Despliegue:** Docker simplifica el proceso de despliegue de aplicaciones, ya que los contenedores incluyen todas las dependencias necesarias para ejecutar la aplicación. Esto reduce el tiempo y el esfuerzo necesarios para desplegar y actualizar las aplicaciones en el autobús.

1. **Pods para Cada Tipo de Sensor**

**Descripción:** Se asigna un pod separado para cada tipo de sensor en el autobús (combustible, presión de aire, ambiente, temperatura, velocidad, GPS). Cada pod está encargado de la recolección y procesamiento de datos específicos de su sensor.

**Justificación:**

* **Modularidad:** La separación en pods permite que cada tipo de sensor funcione de manera independiente. Esto facilita el mantenimiento y la actualización del sistema, ya que las modificaciones en un pod no afectan a los demás. También permite una gestión más eficiente de los recursos, ya que cada pod puede ser escalado de manera independiente según la demanda.
* **Optimización del Rendimiento:** La carga de trabajo está distribuida entre múltiples pods, lo que evita cuellos de botella en el procesamiento de datos. Cada pod puede centrarse en procesar los datos de un tipo específico de sensor, lo que mejora la eficiencia del sistema en general.

1. **Colas de Mensajería (RabbitMQ o Kafka)**

**Descripción:** Los datos recolectados por los sensores se envían a colas de mensajería, como RabbitMQ o Kafka. Estas colas actúan como intermediarios para almacenar los datos temporalmente antes de su transmisión a la central.

**Justificación:**

* **Garantía de Entrega:** RabbitMQ y Kafka son sistemas de mensajería robustos que aseguran que los datos sean entregados incluso en caso de fallos en el sistema. Esto es crucial para evitar la pérdida de datos durante la operación del autobús y garantizar que toda la información recopilada sea transmitida a la central.
* **Desacoplamiento:** Las colas de mensajería permiten desacoplar los componentes del sistema. Los pods de sensores pueden enviar datos a las colas sin necesidad de preocuparse por el estado de los servicios de recepción en la central, y los sistemas de recepción pueden procesar los datos cuando estén disponibles, mejorando la robustez y la flexibilidad del sistema.

1. **Transmisión de Datos en Formato JSON**

**Descripción:** Los datos recopilados en el autobús se transmiten en formato JSON cuando el autobús llega a la terminal. JSON (JavaScript Object Notation) es un formato ligero y fácil de leer.

**Justificación:**

* **Estandarización y Facilidad de Integración:** JSON es un formato ampliamente utilizado en la transmisión de datos y es compatible con una gran cantidad de herramientas y servicios. Utilizar JSON facilita la integración con otros sistemas y servicios en la central, ya que es un formato estandarizado que muchos sistemas pueden procesar fácilmente.
* **Legibilidad y Manipulación de Datos:** JSON es fácil de leer y entender tanto para humanos como para sistemas automatizados. Esto facilita la depuración y el análisis de datos, así como la generación de informes y la toma de decisiones.

1. **API Gateway en la Central**

**Descripción**: Un API Gateway en la central recibe las solicitudes de datos desde los autobuses y actúa como el punto de entrada para todas las solicitudes de datos.

**Justificación:**

* **Gestión Centralizada de API:** El API Gateway centraliza la gestión de las solicitudes de datos, facilitando el enrutamiento, la autenticación, y la autorización. Esto permite una administración más eficiente de las solicitudes y facilita la implementación de políticas de seguridad y control de acceso.
* **Escalabilidad:** El API Gateway puede manejar grandes volúmenes de tráfico de datos y distribuir las solicitudes a los servicios adecuados en la central. Esto permite escalar el sistema para manejar un número creciente de autobuses sin comprometer el rendimiento.

1. **Microservicios para Procesamiento y Almacenamiento de Datos**

**Descripción**: En la central, se utilizan microservicios para procesar y almacenar los datos recibidos de los autobuses. Cada microservicio tiene una responsabilidad específica, como almacenamiento, análisis, o generación de informes.

**Justificación:**

* **Desacoplamiento:** La arquitectura de microservicios permite que diferentes componentes del sistema se desarrollen, desplieguen y escalen de manera independiente. Esto facilita la evolución y el mantenimiento del sistema, ya que los cambios en un microservicio no afectan a los demás.
* **Mantenimiento y Actualización:** Los microservicios pueden ser actualizados o reemplazados de manera independiente, lo que simplifica el proceso de mantenimiento y permite la implementación de nuevas funcionalidades sin interrumpir el funcionamiento del sistema.

1. **Microservicio de Monitoreo de Velocidad y Sistema de Alertas**

**Descripción**: Un microservicio específico se encarga del monitoreo en tiempo real de la velocidad del autobús. Este microservicio verifica los datos de velocidad y genera alertas si se superan las restricciones locales.

**Justificación:**

* **Monitoreo en Tiempo Real:** El microservicio de monitoreo permite la supervisión continua de la velocidad del autobús, lo que es crucial para detectar y responder rápidamente a violaciones de las restricciones de velocidad. Esto mejora la seguridad del autobús y asegura el cumplimiento de las normativas.
* **Automatización de Alertas:** El sistema de alertas automatiza la notificación de problemas, lo que permite a los operadores actuar rápidamente en caso de exceso de velocidad. Esto reduce el tiempo de respuesta y mejora la eficiencia operativa.

1. **Almacenamiento de Datos en Bases de Datos**

**Descripción**: Los datos transmitidos desde los autobuses se almacenan en bases de datos en la central para su posterior análisis y generación de informes.

**Justificación:**

* **Análisis y Generación de Informes:** El almacenamiento de datos permite realizar análisis a largo plazo y generar informes detallados sobre el rendimiento de la flota. Esto ayuda a identificar patrones, prever mantenimientos y tomar decisiones informadas basadas en datos históricos.
* **Histórico de Datos:** Mantener un registro histórico de los datos permite consultar y comparar el rendimiento de los autobuses a lo largo del tiempo, lo que es útil para la gestión de la flota y la planificación estratégica.

1. **Seguridad de los Datos**

**Descripción:** La transmisión de datos entre los autobuses y la central está protegida mediante encriptación y mecanismos de autenticación.

**Justificación:**

* **Privacidad e Integridad:** La encriptación asegura que los datos sean accesibles solo para usuarios autorizados y evita que sean alterados durante la transmisión. Esto protege la privacidad de la información y garantiza la integridad de los datos.
* **Cumplimiento Normativo:** La implementación de mecanismos de seguridad ayuda a cumplir con las regulaciones locales e internacionales sobre protección de datos y privacidad, evitando posibles sanciones y mejorando la confianza en el sistema.

1. **Cumplimiento Normativo**

**Descripción:** El sistema está diseñado para cumplir con las regulaciones locales e internacionales sobre la operación de vehículos de transporte, incluyendo restricciones de velocidad y otras normativas.

**Justificación:**

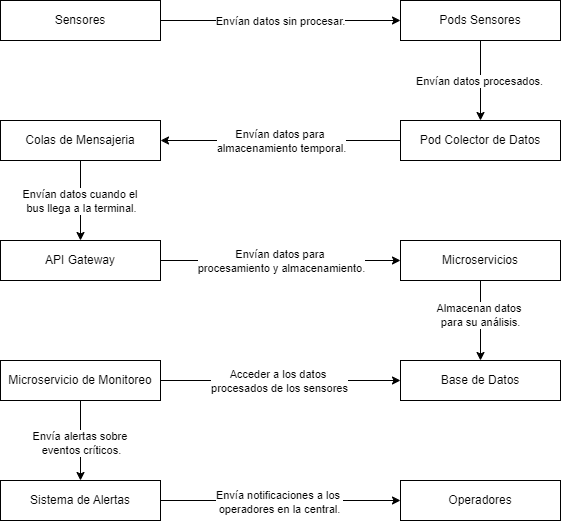
* **Evitación de Multas:** El cumplimiento de las normativas asegura que la empresa evite multas y sanciones asociadas con el incumplimiento de las regulaciones de transporte.
* **Seguridad en las Carreteras:** Al cumplir con las restricciones de velocidad y otras normativas.

**DISEÑO DE LA ARQUITECTURA**

**Diagrama de Contexto General**

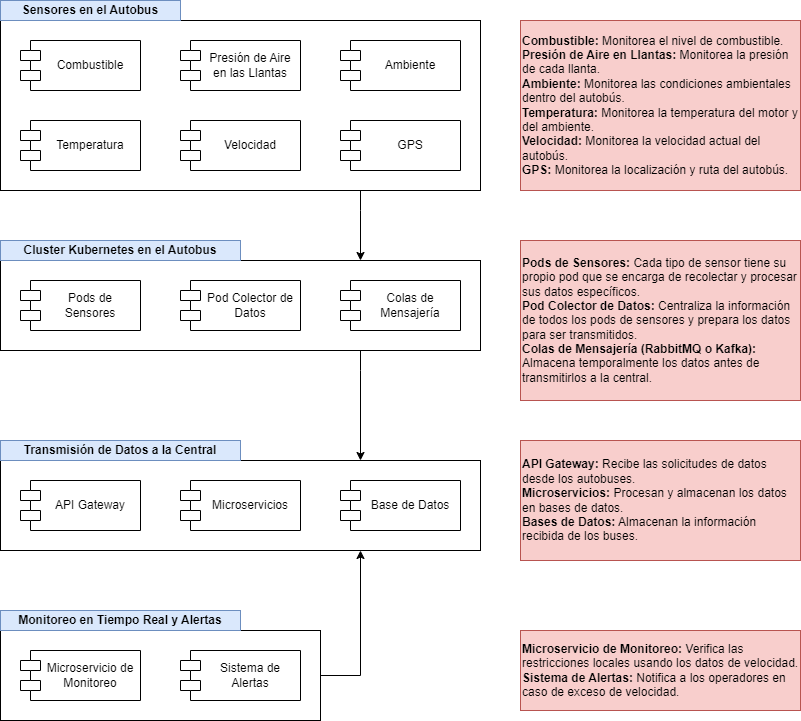
El diagrama de contexto general proporciona una visión macro de los componentes y la interacción entre los distintos elementos del sistema de monitoreo y telemetría para "Buses Seguros S.A.". En este diagrama, se destacan los flujos de datos y las relaciones principales entre los sensores en los autobuses, los componentes del sistema en cada autobús, y los sistemas centrales en la empresa.

* **Sensores:** Recopilan datos sobre el estado del autobús (relación de medición).
* Cluster Kubernetes: Ejecuta pods para procesar los datos de los sensores (relación de procesamiento).
* **Colas de Mensajería:** Almacenan temporalmente los datos (relación de almacenamiento temporal).
* **API Gateway:** Recibe las solicitudes de datos desde los autobuses (relación de recepción).
* Microservicios: Procesan, almacenan y analizan los datos (relación de procesamiento y análisis).
* **Base de Datos:** Almacena los datos históricos y en tiempo real (relación de almacenamiento).
* **Sistema de Alertas:** Notifica a los operadores sobre eventos críticos (relación de notificación).



**Diagrama de Componentes**

El diagrama de componentes ilustra la estructura del sistema de monitoreo y telemetría para la empresa de transporte "Buses Seguros S.A.". Este sistema se despliega tanto en los autobuses como en la central de la empresa, permitiendo la captura, procesamiento, transmisión y análisis de datos en tiempo real.



**Diagrama de Despliegue**

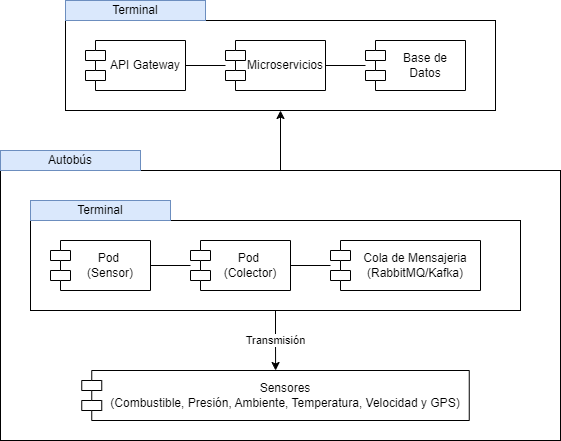
El diagrama de despliegue ilustra la arquitectura física del sistema de monitoreo y telemetría para la empresa de transporte "Buses Seguros S.A.". Los principales componentes y su interacción se dividen en dos entornos: el entorno del autobús y el entorno central de la empresa.

1. **Terminal**

* **API Gateway:** Recibe datos desde los autobuses.
* **Microservicios:** Procesan, almacenan y monitorean los datos.
* **Bases de Datos:** Almacenan la información procesada.

1. **Autobús**

* **Cluster Kubernetes:** Gestiona los pods en el autobús.
* **Pods (Sensor):** Recogen y procesan datos de diferentes sensores.
* **Pod Colector de Datos:** Centraliza y prepara los datos para la transmisión.
* **Cola de Mensaje (RabbitMQ/Kafka):** Asegura que los datos se almacenen temporalmente y se transmitan de manera confiable.
* **Sensores:** Dispositivos físicos que recogen datos sobre el autobús (combustible, presión, ambiente, temperatura, velocidad, GPS).



**Diagrama de Secuencia**

El diagrama de secuencia muestra las interacciones entre los componentes clave del sistema en una serie de pasos que reflejan el flujo de datos a través del sistema. Cada paso en el diagrama representa un componente o una acción específica en el proceso de monitoreo y telemetría:

1. **Recolección y Procesamiento de Datos en el Autobús:**

* **Sensores:** Los sensores instalados en el autobús recolectan datos sobre diversas métricas como combustible, presión de aire en las llantas, condiciones ambientales, temperatura, velocidad y ubicación GPS.
* **Pods de Sensores:** Los datos de los sensores se envían a los pods correspondientes dentro del cluster Kubernetes en el autobús. Cada pod está especializado en manejar datos de un tipo específico de sensor.
* **Pod Colector de Datos:** Este pod centraliza los datos provenientes de los diferentes pods de sensores y los prepara para su transmisión a través de las colas de mensajería.

1. **Transmisión de Datos a la Central:**

* **Colas de Mensajería:** Los datos preparados por el pod colector se envían a las colas de mensajería (RabbitMQ o Kafka), donde se almacenan temporalmente. Esto garantiza que los datos sean almacenados de manera confiable antes de su transmisión a la central.
* **API Gateway:** Cuando el autobús llega a una terminal, los datos almacenados en las colas de mensajería se transmiten a la central a través de un API Gateway. Este componente recibe los datos y los dirige a los microservicios adecuados para su procesamiento.

1. **Procesamiento y Almacenamiento en la Central:**

* **Microservicios:** En la central, los microservicios procesan y almacenan los datos recibidos en las bases de datos de la empresa. Este procesamiento incluye la preparación de los datos para su análisis y almacenamiento a largo plazo.
* **Bases de Datos:** Los datos se almacenan en bases de datos para su análisis posterior y la generación de informes.

1. **Análisis y Generación de Alertas:**

* **Sistema de Monitoreo y Alertas:** Los microservicios analizan los datos almacenados, especialmente la velocidad del autobús, para detectar cualquier violación de las restricciones locales. Si se detecta una infracción, el sistema de alertas genera notificaciones para los operadores en la central.

El diagrama de secuencia proporciona una visión clara de cómo los datos fluyen desde su origen en los sensores hasta su análisis y la generación de alertas, destacando la interdependencia entre los diferentes componentes del sistema y asegurando una operación eficiente y efectiva.

