



IoT^x: Plataforma tecnológica para la gestión de bahías de carga/descarga de mercancías en espacios públicos

Bernardo Hernández Sánchez
Miguel Gastón Cedillo Campos
Marisol Barrón Bastida
Cesar Jaime Montiel Moctezuma

Publicación Técnica No. 694
Sanfandila, Qro.
2022

ISSN 0188-7297

Esta investigación fue realizada en la Coordinación de Transporte Integrado y Logística del Instituto Mexicano del Transporte, por el Mtro. Bernardo Hernández Sánchez, la Mtra. Marisol Barrón Bastida, el Dr. Miguel Gastón Cedillo Campos y el Dr. Cesar Jaime Montiel Moctezuma.

Esta publicación es el producto final del proyecto de investigación interna TI 08/21: “Plataforma tecnológica para la gestión de bahías de carga y descarga de mercancías en espacios públicos, etapa 1”. El término “IoT” utilizado en el título de esta publicación alude al Programa de Investigación en Tecnología de Internet de las Cosas (IoT), impulsado por un grupo de investigadores del Laboratorio Nacional Conacyt en Sistemas de Transporte y Logística (SiT LOG Lab), cuyo objetivo es su aplicación en temas de Transporte y Logística para la toma de decisiones.

Este trabajo toma como base las experiencias obtenidas durante el desarrollo del Primer Laboratorio de Logística Urbana en Tiempo Real de América Latina (LogistiX-Lab), diseñado e implementado por el Dr. Miguel Gastón Cedillo Campos (IMT), y el Dr. Jan Fransoo (Tilburg School of Economics and Management).

Se agradece la revisión y aportaciones del Dr. Carlos Martner Peyrelongue, coordinador de Transporte Integrado y Logística del IMT, cuyas observaciones y contribuciones mejoraron la calidad de este documento.

Se agradece también el apoyo brindado por el Laboratorio Nacional de Sistemas de Transporte y Logística (SiT-LOG Lab), cuyas capacidades tecnológicas facilitaron el buen desarrollo de este proyecto.

Contenido

	Página
Sinopsis.....	vii
Abstract	ix
Resumen ejecutivo.....	xi
Introducción.....	1
1. Panorama general del desarrollo de la tecnología IoT	3
2. El modelo de referencia.....	7
2.1 La capa sensorial	8
2.2 La red de comunicaciones	9
3. Arquitectura del servidor.....	11
3.1 El servidor	11
3.2 Tecnologías utilizadas.....	12
3.3 Lógica del negocio	13
3.4 Acceso a microservicios.....	15
3.5 API RESTful	15
3.6 Códigos de estado	16
4. Desarrollo de la aplicación	19
4.1 El modelo funcional.....	19
5. Resultados obtenidos	23
Conclusiones.....	27
Bibliografía	29

Índice de figuras

Figura 1.1 Proporción de proyectos IoT en cuatro etapas.....	4
Tabla 1.1 Adopción y valor de IoT por país.....	5
Figura 1.2 Distribución total de proyectos IoT en el Mundo, por número de dispositivos conectados.	6
Figura 2.1 Arquitectura general.	8
Figura 2.2 Dispositivo IoT con sensor ferromagnético.	8
Figura 2.3 Baliza con tecnología Bluetooth.	9
Figura 2.4 Antena CXL900-3LW.	9
Figura 2.5 Estación base SBT-T3902.	10
Figura 3.1 Diagrama de conectividad IoT.....	11
Figura 3.2 Arquitectura del servidor	14
Tabla 3.1 Operaciones RESTful comunes	15
Figura 4.1 Logotipos de las tecnologías de desarrollo.	19
Figura 4.3 Codificación del componente de consulta de datos históricos.....	21
Figura 5.1 Vista inicial de la aplicación web.	23
Figura 5.2 Vista de todas las bahías.	24
Figura 5.3 Vista de la sección por bahía	25
Figura 5.4 Vista de la sección “Analítica”	26

Sinopsis

En este documento se presentan los primeros resultados del desarrollo e implementación de una plataforma tecnológica (sistema web) para la gestión de espacios públicos en zonas urbanas (bahías de carga/descarga para vehículos). Se basa en uso de software libre y dispositivos con tecnología IoT (Internet de las Cosas).

Particularmente se aborda un caso de estudio, el cual toma como base lo desarrollado previamente por (Cedillo-Campos y Fransoo, 2019; Fransoo y Cedillo-Campos, 2022), cuyos resultados revelaron que la disponibilidad de espacios de carga/descarga en áreas de alta densidad de puntos de entrega de mercancías, impulsa una mayor eficiencia en las rutas, así como un menor tiempo de estacionamiento y un menor impacto en ruido y contaminación para las ciudades. Se propone un enfoque técnico y centralizado para organizar elementos tecnológicos en una red de área amplia de baja potencia con flujo de datos en tiempo real.

Con base en el proyecto interno TI 08/21, primero se desarrolló un análisis de requerimientos funcionales para la identificación de la lógica del negocio. Segundo, se estableció un protocolo para el desarrollo del sistema web y para la interacción con los sensores IoT. Tercero, se realizaron pruebas operacionales para la implementación final del sistema.

Abstract

This document presents the first results of the development and implementation of a technological platform (web system) for the management of public spaces in urban areas (loading/unloading bays for vehicles). It is based on the use of free software and devices with IoT (Internet of Things) technology.

In particular, a case study is addressed, which is based on what was previously developed by (Cedillo-Campos and Fransoo, 2019; Fransoo and Cedillo-Campos, 2022), whose results revealed that the availability of loading/unloading spaces in areas of The high density of merchandise delivery points drives greater efficiency on the routes, as well as less parking time and a lower impact on noise and pollution for cities. A technical and centralized approach is proposed to organize technological elements in a low-power wide-area network with real-time data flow.

Based on the internal IT project 08/21, a functional requirements analysis was first developed to identify the business logic. Second, a protocol for the development of the web system and for the interaction with the IoT sensors was established. Third, operational tests were carried out for the final implementation of the system.

Resumen ejecutivo

En este documento se presentan los primeros resultados del desarrollo de una plataforma tecnológica (sistema web) basada en software libre y uso de dispositivos con tecnología de baja potencia y largo alcance IoT (Internet de las Cosas). Este sistema fue desarrollado en el Laboratorio Nacional en Sistemas de Transporte y Logística (SiT-LOG) adscrito a la Coordinación de Transporte Integrado y Logística, del Instituto Mexicano del Transporte (IMT).

En el capítulo 1 “Panorama general del desarrollo de la tecnología IoT”, se abordan los resultados del reporte Hypothesis Group “*IoT Signals Edition 2021*”. Hypothesis es una agencia de creatividad, diseño, estrategia y de información. Para este reporte, tomaron las opiniones de más de 3000 responsables de la toma de decisiones en empresas en todo el mundo. Para el caso de México, se documentaron 14 proyectos de IoT y en este capítulo se mencionan dos de ellos.

El capítulo 2 “El modelo de referencia”, se abordan los detalles del modelo de referencia utilizado, y se especifican los componentes físicos y su organización funcional con una breve explicación de la arquitectura tecnológica.

En el capítulo 3 “Arquitectura del servidor”, se presentan detalles técnicos del lado del servidor, su configuración base y la arquitectura general para la implementación del proyecto. Se aborda la lógica del negocio y la tecnología de microservicios, así mismo, se describe el uso de interfaces de aplicación (API’s) sobre las cuales se definen las reglas que se deben seguir para comunicarse con otros sistemas de *software* para la manipulación de la información en el sistema.

El capítulo 4 “Desarrollo de la aplicación”, en este capítulo se detalla el diseño de un modelo funcional, con la descripción de sus componentes para la implementación y aplicación de pruebas piloto, el cómo se relacionan y comunican los elementos del modelo, su escalabilidad, modularidad y la configuración de su implementación.

El capítulo 5 “Los resultados obtenidos”, aquí se describen los resultados mediante la presentación de una serie de pantallas “*Front End*”, *esta es la parte de una aplicación que interactúa con los usuarios, mejor conocida como “del lado del cliente”*. Brevemente se explica la lógica de su estructura, desde la interfaz principal, hasta la sección de analítica del sistema.

Finalmente, en el capítulo 6 se presentan las conclusiones y perspectivas de investigación futuras.

Introducción

El Internet de las Cosas o “Internet of Things” (IoT) es uno de los pilares fundamentales de la Revolución 4.0 y la Transformación Digital. En este documento se presentan los primeros resultados del desarrollo de una plataforma tecnológica (sistema web) basada en software libre y uso de dispositivos con tecnología de baja potencia y largo alcance IoT (Internet de las Cosas). La propuesta incluye la implementación de un sistema web para gestión de espacios públicos en zonas urbanas (bahías de carga/descarga para vehículos). Dicha plataforma, pretende ser una base inicial que favorezca el desarrollo de aplicaciones en IoT a favor de la gestión de la infraestructura urbana en los países de América Latina.

El Laboratorio Nacional en Sistemas de Transporte y Logística, ha impulsado proyectos de desarrollo de software con una arquitectura monolítica, la cual, es un modelo tradicional de un programa de software que se compila como una unidad unificada y que es autónoma e independiente de otras aplicaciones. La palabra "monolito" suele evocar algo grande y glacial, una imagen que no está alejada de la realidad de las arquitecturas monolíticas para el diseño de software. Para hacer cambios en este tipo de aplicación se debe actualizar toda la pila, lo que requiere acceder a la base de código y compilar e implementar una versión actualizada de la interfaz del lado del servicio.

Esta publicación hace uso de una arquitectura orientada a microservicios, la cual se basa en una serie de servicios que se pueden implementar de forma independiente. Estos servicios tienen su propia lógica y base de datos con un objetivo específico. Los microservicios no reducen la complejidad, pero hacen que cualquier complejidad sea visible y más gestionable, ya que separan las tareas en procesos más pequeños que funcionan de manera independiente entre sí y contribuyen al conjunto global.

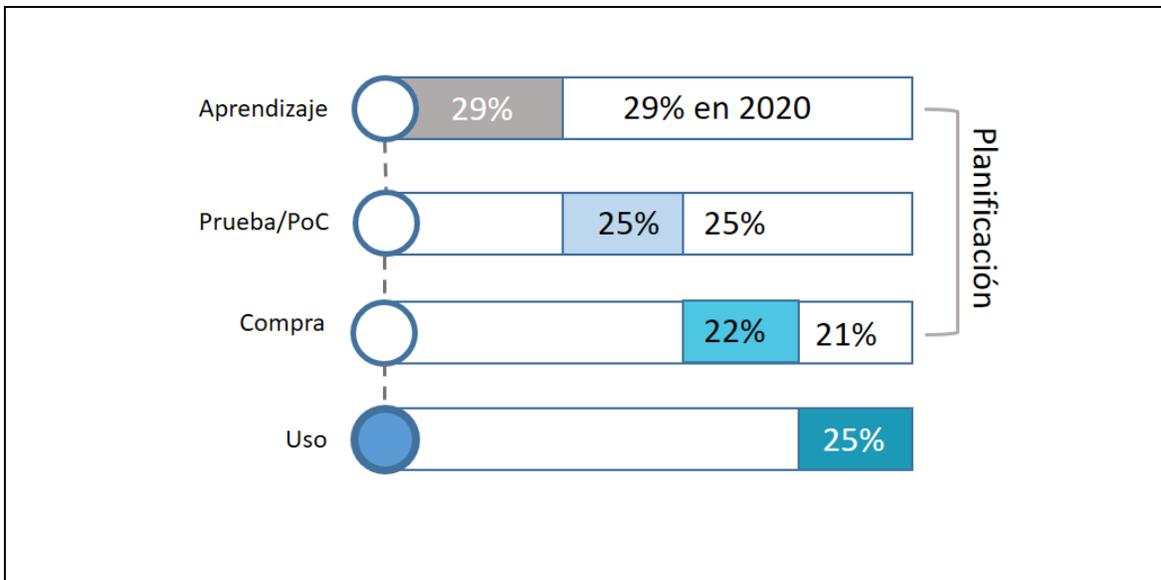
Finalmente, este proyecto continuará en constante desarrollo creando nuevas funcionalidades y módulos de software que permitan aplicarlo a diversos proyectos, por ejemplo, en el sistema ferroviario.

1. Panorama general del desarrollo de la tecnología IoT

Como consecuencia de la pandemia de COVID-19, el Internet de las Cosas se ha ido consolidado en la mayoría de los sectores del mercado, permitiendo a las organizaciones crear aplicaciones adaptables a todo tipo de entornos y alcanzar sus objetivos. El enfoque principal es crear una experiencia entre el mundo físico y el digital y ofrecer soluciones de IoT más seguras y escalables, reduciendo la complejidad y creando oportunidades en nuevas áreas de negocio.

Desde esta perspectiva, resulta particularmente importante comprender el panorama general del desarrollo de la tecnología IoT. Sobre esto, la empresa tecnológica Microsoft hizo el encargo a Hypothesis Group (agencia de conocimientos, diseño y estrategia, que ejecutó una investigación sobre IoT en 2021), elaborar un reporte, el cual se titula “*IoT Signals Edition*”. Éste se basó en la opinión de más de 3000 responsables de la toma de decisiones en empresas de EE. UU., Reino Unido, Alemania, Francia, China y Japón, que participaban en proyectos dentro del área de Internet de las Cosas. Esta investigación incluyó a responsables de la toma de decisiones empresariales, responsables de la toma de decisiones de tecnologías de información y desarrolladores de una variedad de industrias como: fabricación, venta minorista/mayorista, petróleo y gas, atención médica, entre otros.

Los resultados de dicho estudio, se presentan en la Figura 1.1. Se observa que, en 2021, el 29% de los proyectos de IoT están en la etapa de aprendizaje, el mismo porcentaje que en 2020. El porcentaje de proyectos en la etapa de prueba de concepto (PoC) también permanece igual, 25% en 2020 y 2021. Los proyectos en la fase de compra han aumentado en un 1%, pasando del 21% en 2020 al 22% en 2021.



Fuente: Reporte IoT Signals Edición 3 con datos de octubre 2021. Microsoft Hypothesis Group.

Figura 1.1 Proporción de proyectos IoT en cuatro etapas.

En cuanto al porcentaje de proyectos en fase de uso, EE. UU., lidera el grupo, con un 27%, en comparación con países como España (22%) y Australia (18%). EE. UU., también está por delante del juego global en términos de aumento del uso de IoT. Así, el 78% de las organizaciones de EE. UU. planea usar más IoT en los próximos dos años. En contraste, solo el 53% de las empresas alemanas encuestadas planearon aumentar el uso para 2023 y con el 51% las organizaciones en Japón. Esto puede deberse a que el mercado alemán y japonés podrían ser más cautelosos en su enfoque de la tecnología.

Es probable que los proyectos en la región de Asia-Pacífico lleven más tiempo porque se necesita más educación y conocimiento sobre IoT para poder usarlo. Las organizaciones en Asia-Pacífico también enfrentan desafíos más frecuentes al no tener los recursos para implementar y administrar soluciones de IoT. En la Tabla 1.1, se muestran los porcentajes de adopción del IoT de algunos países a nivel mundial.

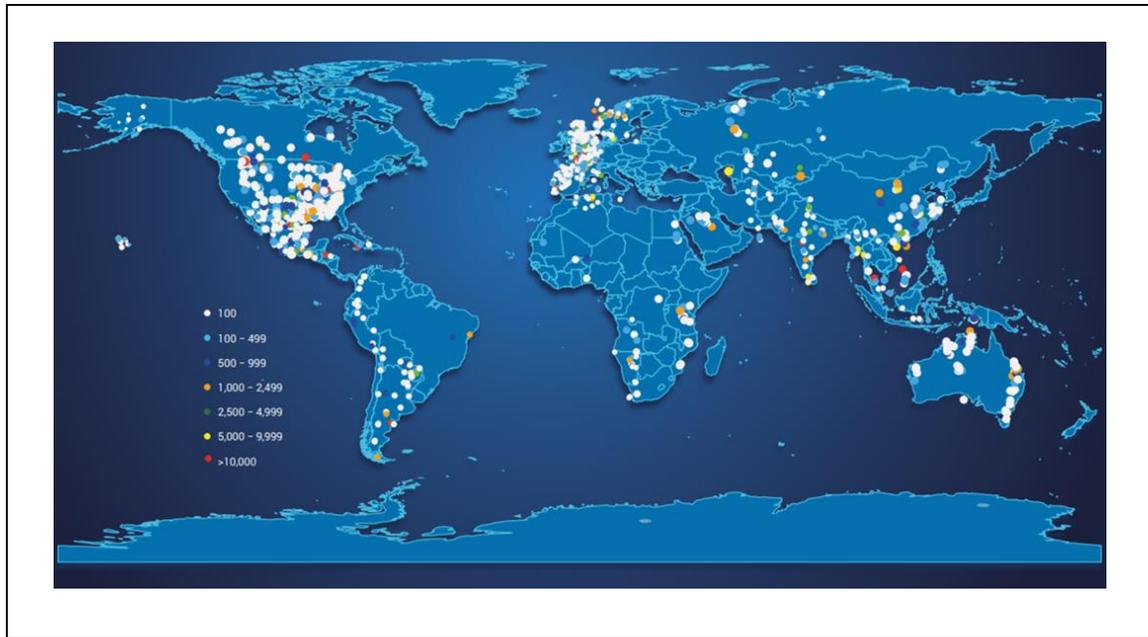
Tabla 1.1 Adopción y valor de IoT por país.

											
	Global	Estados Unidos	Reino Unido	Francia	Alemania	España	Italia	Benelux	China	Japón	Australia
% de adopción IoT	90%	94%	91%	91%	88%	89%	95%	91%	85%	88%	96%
% Proyectos en fase de uso	25%	27%	25%	23%	25%	22%	26%	25%	25%	23%	18%
Tiempo de espera para llegar a la fase de uso (meses)	12	11	13	12	14	11	10	12	16	12	16
Planean usar IoT en proyectos de más de dos años	66%	78%	69%	67%	53%	76%	69%	59%	65%	51%	56%

Fuente: Elaboración propia con base en el Reporte IoT Signals Edición 3 con datos de octubre 2021. Microsoft Hypothesis Group.

Otro importante estudio a mencionar es el desarrollado en México por el Instituto Federal de Telecomunicaciones (IFT), titulado: *“Análisis exploratorio de la comercialización de servicios de conectividad para IoT”* (2019). Este informe fue elaborado con base en información estadística obtenida mediante técnicas de ciencia de datos, que van desde el análisis exploratorio, hasta la automatización de consultas a sitios web para la sistematización y elaboración de bases de datos, permitiendo identificar patrones de comercialización entre los proveedores de Internet de las Cosas (IoT, por sus siglas en inglés) a nivel mundial.

En la Figura 1.2 se muestra la distribución total de proyectos IoT en el mundo, por número de dispositivos conectados. Se trata de 102 proyectos, que se encuentran en 31 países, entre los cuales destaca Estados Unidos que contabiliza 33 proyectos; Inglaterra 15; Canadá, Francia y España con 5.



Fuente: IFT con datos de Global Data (2019), IoT Deployment Database. Disponible en: <https://www.globaldata.com/>.

Figura 1.2 Distribución total de proyectos IoT en el Mundo, por número de dispositivos conectados.

Para el caso de México, se encuentran documentados 14 proyectos de IoT. El proyecto más grande se refiere al despliegue de un sistema SCADA (Supervisory Control and Data Acquisition) para el control y monitoreo de agua en Saltillo Coahuila, proyecto que implica conectar entre 5,000 y 9,999 dispositivos.

El segundo proyecto con el mayor número de dispositivos conectados corresponde a la Comisión Federal de Electricidad (CFE). Este tiene como objetivo proporcionar comunicación M2M para aplicaciones de Smart Grid con un intervalo de dispositivos conectados de entre 1,000 y 2,499; este rango representa el 4.97% de los sensores que se están desplegando a nivel mundial.

2. El modelo de referencia

Para el desarrollo del proyecto se usó como base un modelo de referencia de cuatro capas. En la capa de percepción los sensores se encargan de obtener los datos, posteriormente se aplican diversas técnicas para la analítica de datos y finalmente, se implementa una capa de visualización, la cual, permite presentar de manera gráfica la información histórica y en tiempo real.

A continuación, se presenta una breve descripción de cada una de las capas:

Capa de percepción: Es la capa sensorial de IoT donde “*las cosas*” identifican sus alrededores, recaban información del mundo físico. Los sensores, actuadores y otros dispositivos inteligentes, capturan datos (sensores) y con estos, toman medidas (actuadores).

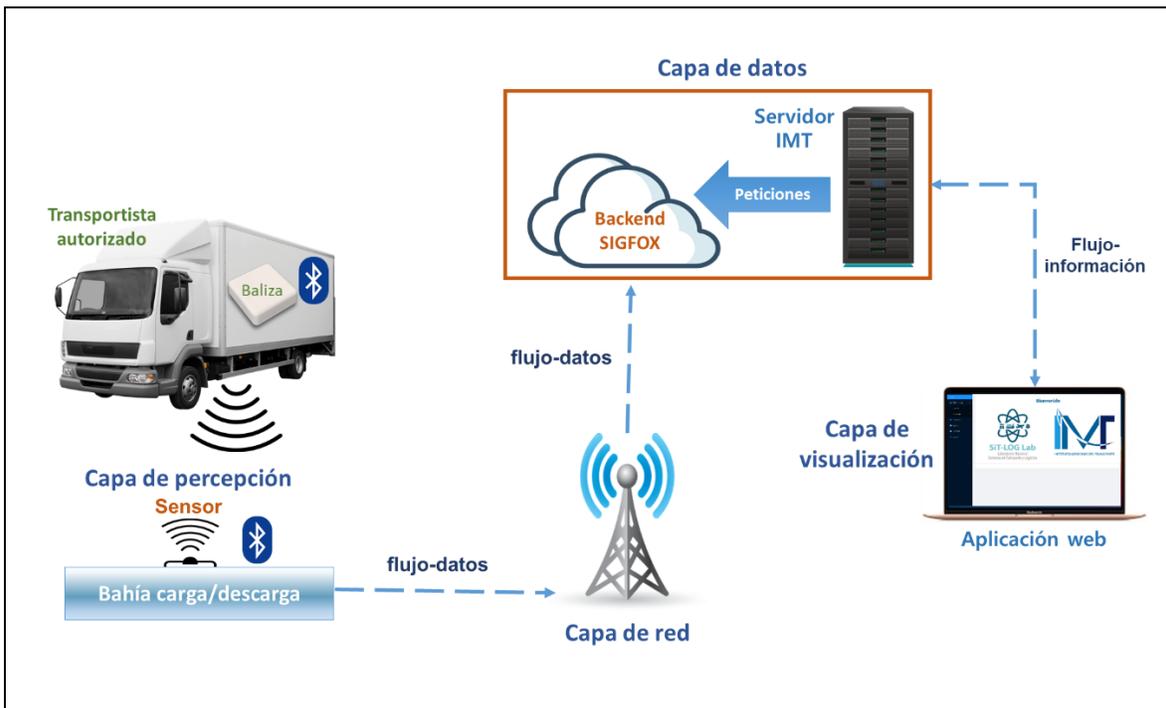
Capa de red: Su función principal es transmitir y procesar la información obtenida por la capa de percepción. Así mismo, es responsable de interconectar otras redes de dispositivos inteligentes, elementos de red y servidores. Comprende los dispositivos de red y los protocolos de comunicación.

Capa de datos: Capa responsable de entregar servicios y aplicaciones específicas al usuario final. Proporciona una gestión global de la aplicación basada en la información de los objetos procesados en la capa de *Middleware*¹.

Capa de visualización: Esta capa comprende el análisis y visualización. El enfoque de ésta es analizar los datos abastecidos por el IoT y proporcionarlos a los usuarios y aplicaciones para que adquieran sentido.

¹ El middleware es un software con el que las diferentes aplicaciones se comunican entre sí. El middleware actúa como un puente entre tecnologías, herramientas y bases de datos diversas para que pueda integrarlas sin dificultad en un único sistema.

En la figura 2.1 se muestra el modelo de la arquitectura general descrito anteriormente.



Fuente: Elaboración propia, basada en la arquitectura de cuatro capas IoT

Figura 2.1 Arquitectura general.

2.1 La capa sensorial

Uno de los dispositivos utilizado en la instrumentación de la bahía, que tiene como función principal la de monitorización del estado de ocupación de la bahía, es el **IoT Park** (de *IoT Solutions, Ltd*). Este dispositivo realiza la detección de los estados (libre u ocupado) haciendo uso de un sensor ferromagnético integrado en el dispositivo. Permite detectar la presencia de masa metálica y transmite el dato en tiempo real al sistema.

En la figura 2.2 se muestra el sensor utilizado en el proyecto.



Fuente: Fotografía propia, basada en la tecnología de IoT Solutions 2021.

Figura 2.2 Dispositivo IoT con sensor ferromagnético.

Otro dispositivo que complementa la capa de percepción del sistema, es una baliza con la característica de tecnología *Bluetooth* de baja energía. Este dispositivo se usó como elemento de identificación en el sistema, cada dispositivo tiene un ID único y se parametriza de acuerdo a la empresa que lo porta.

En la figura 2.3 se muestra el aspecto físico de la baliza.



Fuente: Obtenida del sitio web blueupbeacons.com.

Figura 2.3 Baliza con tecnología Bluetooth.

2.2 La red de comunicaciones

Para lograr la transmisión de datos y la telemetría de los sensores, se utilizó una antena omnidireccional, marca **PROCOM**, modelo **CXL900-3LW**. Esta antena forma parte de la capa de red del modelo, con un alcance de recepción de hasta 20 km.

En la figura 2.4 se ilustra la antena en las instalaciones del SiT LOG Lab-IMT en Querétaro.



Fuente: Fotografía propia.

Figura 2.4 Antena CXL900-3LW.

Para complementar la red, se usó una estación base **Sigfox SBT-T3902 v3.0-1**, con rango de frecuencia **RCZ 902.104 - 902.292MHz** y micro canal de **100Khz**. En la siguiente figura se ilustra la estación base.

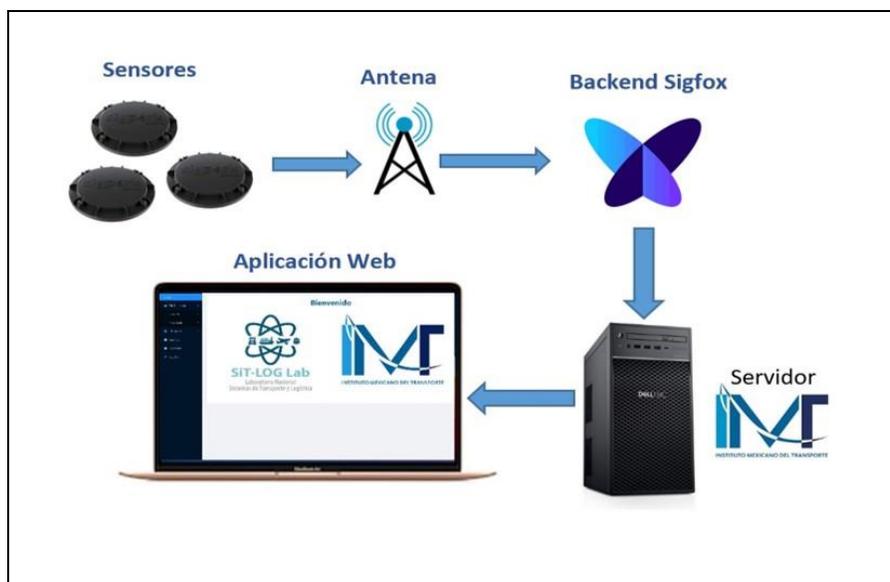


Fuente: Fotografía propia.

Figura 2.5 Estación base SBT-T3902.

3. Arquitectura del servidor

El desarrollo de la aplicación para la gestión de las bahías de carga y descarga, se dividió en dos partes. En primera instancia se planteó la lógica de negocio, es decir, las funcionalidades elementales del sistema para la gestión de la información. Y segundo, se enfocó en el desarrollo de la aplicación web donde el usuario final podría interactuar con los sensores, sus configuraciones y un panel de control. En éste último se puede visualizar información relevante del sistema, desde indicadores de desempeño de los sensores, incluyendo alertas y errores en la infraestructura, hasta datos representativos de las bahías correspondientes, históricos, estado actual, fecha y hora de eventos, así como otros indicadores que describen las operaciones habituales. En la figura 3.1 se muestra el diagrama de conectividad IoT.



Fuente: Elaboración propia.

Figura 3.1 Diagrama de conectividad IoT.

3.1 El servidor

Para dar soporte al proyecto se montó un servidor con la capacidad para ejecutar módulos de software, así como, para procesar y visualizar información para la toma de decisiones. Sus funciones son: servidor web, gestión de datos, administración del sistema, almacenamiento de la telemetría de los sensores, analítica de datos, etc.

Algunas de las características técnicas del servidor son:

- CPU: Intel(R) Xeon(R) E5-2630 v3 @2.40GHZ (dos procesadores)
- RAM: 128GB
- Almacenamiento: 350GB
- Windows Server R 2016 Std

3.2 Tecnologías utilizadas

Para la definición y desarrollo del proyecto, se utilizaron distintas tecnologías para la comunicación, monitoreo, re-direccionamiento, bases de datos, gestión de cargas, entre otras tareas necesarias para la arquitectura. A continuación, se presentan sus definiciones:

- **Tomcat.** Servidor que funciona como contenedor de *servlets*² desarrollado bajo el proyecto Jakarta en la Fundación de Apache Software (The Apache Software Foundation, 2022).
- **Node.JS.** Entorno en tiempo de ejecución multiplataforma, *OpenSource*, para la capa de servidor basado en el lenguaje de programación JavaScript, asíncrono, con una arquitectura orientada a eventos y basado en el motor V8 de Google (OpenJS Foundation, 2022).
- **React.** Es una biblioteca de Javascript para la creación de interfaces de usuario interactivas de forma sencilla, con la capacidad de actualizar y renderizar de manera eficiente los componentes de las vistas (Meta Platforms, 2022).
- **Microsoft IIS.** Servidor web que contiene un conjunto de servicios para los sistemas operativos de Microsoft (Microsoft, 2022).
- **Spring Cloud Config Server.** Proporciona el soporte para la configuración externalizada de un sistema distribuido del lado servidor y del lado cliente (VMware, 2022c).
- **Yaml.** Es un formato de serialización de datos legible por humanos, inspirado en XML, C, Python, Perl y otros formatos de correos electrónicos (RedHat, 2021).
- **Maven.** Herramienta para la gestión y construcción de proyectos. Se basa en el concepto de un modelo de objetos de proyecto (Project Object Model POM), y gestiona la construcción, los informes y la documentación de un proyecto a partir de una pieza central de información (The Apache Software Foundation, 2022).

² Un servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. La palabra servlet deriva de otra anterior, applet, que se refiere a pequeños programas que se ejecutan en el contexto de un navegador web.

- **GitHub.** Es una plataforma de integración de código para desarrolladores que utiliza el sistema de control de versiones Git (GitHub, 2022).
- **Spring Gateway.** Es un proyecto dentro de Spring que se encarga de proporcionar una biblioteca que define el punto de entrada al ecosistema de microservicios, propiciando enrutamiento dinámico, seguridad y monitorización de las llamadas que se realicen (VMware, 2022).
- **Spring Load Balancer.** Biblioteca dentro de Spring que ofrece la capacidad de distribuir el tráfico de los procesos en un conjunto de instancias diferentes de la misma aplicación (VMware, 2022b).
- **Spring Eureka Server.** Es un servicio REST³, ofrecido por Spring, que se comporta como un servidor que registra y localiza microservicios en un ecosistema, proporciona información de su localización, su estado y datos relevantes de cada uno de estos servicios (Tarnum Java SRL, 2022).
- **PostgreSQL.** Es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto (The PostgreSQL Global Development Group, 2022).

3.3 Lógica del negocio

La arquitectura de desarrollo de la lógica de negocio se basó en el paradigma de microservicios. Esta arquitectura se utilizó debido a la sencillez de implementación, escalabilidad, independencia en las funcionalidades y mejora del rendimiento para la respuesta del servidor a las peticiones externas.

Es una solución eficiente para realizar cambios de manera ágil al sistema y poder personalizarlo de acuerdo a las necesidades de cada proyecto que desee trabajar con este sistema, además ofrece una estrategia de compatibilidad para la comunicación entre diferentes tecnologías, lo cual abre las oportunidades de implementación y colaboración con otros proyectos similares. En la figura 3.2 se puede observar el diagrama de la comunicación interna de los microservicios para este proyecto.

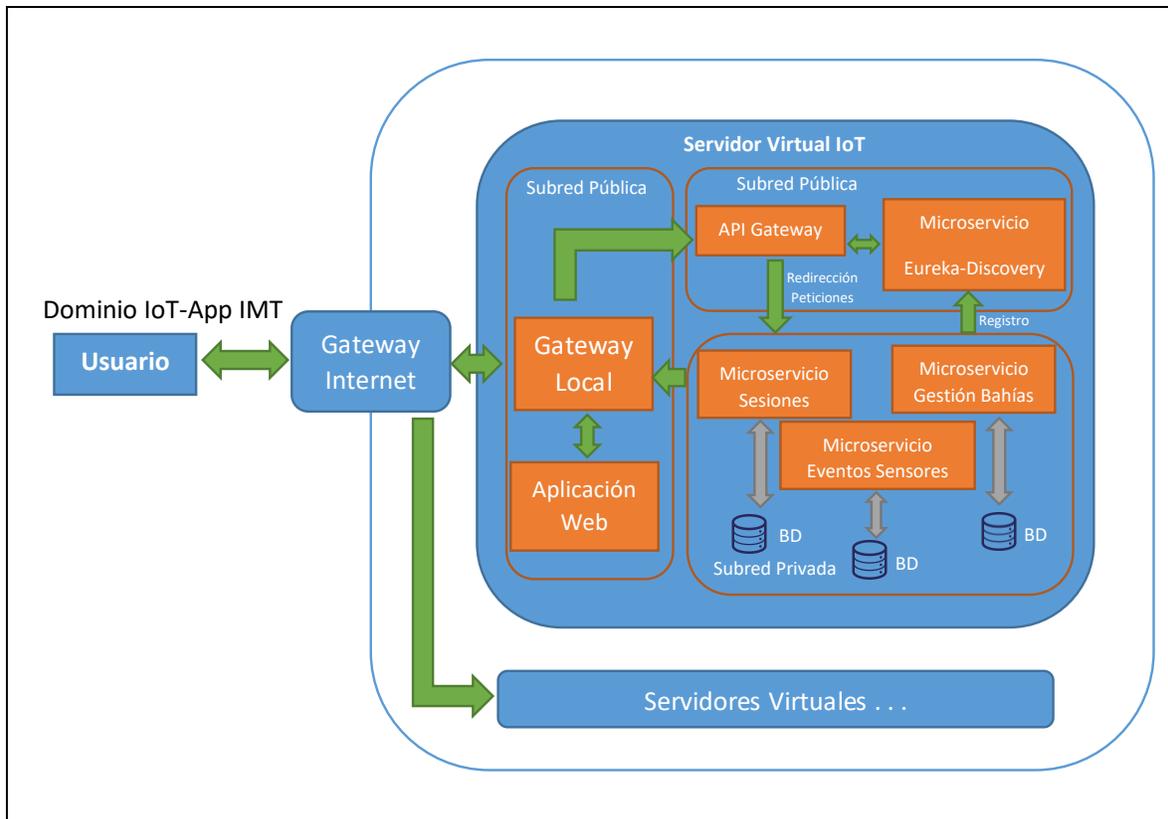
Por otro lado, dentro del servidor se separó el sistema en una red privada y pública. En la primera se encuentra cada uno de los microservicios desarrollados. Para los resultados de esta investigación solo se incluyen tres microservicios y una base de datos asociada a cada uno de estos.

A continuación, se da una breve explicación sobre la comunicación interna a nivel servidor:

- a. La gestión de los usuarios que se conectarán a la aplicación, contemplando la autenticación y la seguridad para los diferentes tipos de usuarios.

³ Servicios REST (REpresentational State Transfer). Estos servicios reciben las peticiones por el protocolo HTTP el cual se usa para ver cualquier página en un navegador web.

- b. La recolección y tratamiento de los datos obtenidos de los sensores de IoT, así como las balizas que interactúan con el sistema. Existen un gran número de eventos generados por cada tipo de sensor dependiendo de la actividad que se ejecuta.
- c. Un tercer microservicio se encarga del análisis de los datos y la generación de indicadores, reportes e históricos de cada dispositivo que interactúa con una bahía. También se incluyen las configuraciones específicas por bahía, usuario (empresa) y estado en el que se encuentran.



Fuente: Elaboración propia.

Figura 3.2 Arquitectura del servidor

En la red pública podemos encontrar la aplicación web que se mostrará a los usuarios finales, y con la cual, interactuarán para lectura-escritura del sistema; esta aplicación se explicará con más detalle en la siguiente sección. También se encuentran dos microservicios que se encargan de gestionar la lógica de negocio: el primero es el encargado de redirigir las peticiones al microservicio correspondiente, y el otro se encarga de mantener una lista activa de los microservicios existentes en el sistema. En el caso de que se actualicen los servicios de la red privada, se desarrollarán nuevos o se eliminarán los existentes. Este microservicio se encargará de mantener actualizada la información para el uso correspondiente de los usuarios finales.

Para acceder como usuario final a la información, se puede realizar de dos maneras: i) la primera es a través del consumo de las API's de cada uno de los microservicios; y ii) la segunda opción es con el uso de la aplicación web.

3.4 Acceso a microservicios

Para la manipulación de la información en el sistema se usa un conjunto de API's desarrolladas con el estilo RESTful⁴. Una API (Interfaz de Programación de Aplicaciones) es un conjunto de definiciones y protocolos que se utilizan para el diseño e integración de las aplicaciones. Por su parte, los servicios web RESTful son servicios web basados en la arquitectura REST. Dicha arquitectura es una metodología de solicitud de recursos que se entrega a través de peticiones HTTP con formatos definidos: JSON, XML o texto sin formato. El formato utilizado para este proyecto fue JSON.

Los servicios web RESTful tienen características particulares que los diferencian de otras arquitecturas de aplicaciones web, estas son: a) contienen operaciones básicas para listar, crear, leer, actualizar y borrar; b) cada operación requiere de los métodos URI y HTTP; c) en el URI se incluye un sustantivo con el nombre del recurso, y d) el método HTTP identifica el verbo a realizar de acuerdo a la siguiente tabla:

Tabla 3.1 Operaciones RESTful comunes

Operación	Método http	URI	Parámetros	Resultado
Listar	GET	/recurso	No aplica	Lista el tipo de recurso
Crear	POST	/recurso	Body	Se crea un nuevo recurso
Leer	GET	/recurso/{id}	No aplica	Recurso dependiendo del id
Actualizar	PUT	/recurso/{id}	A través de una cadena de consulta	Se actualiza el recurso
Borrar	DELETE	/recurso/{id}	No aplica	Se elimina el recurso

Fuente: Elaboración propia

3.5 API RESTful

Las API REST proporcionan una forma flexible y ligera de integrar aplicaciones. Es el método más común para conectar componentes en la arquitectura de microservicios. La selección del estándar RESTful API (también conocido como API REST) se debió a la facilidad de implementación, y a que es uno de los protocolos más utilizados por la comunidad de desarrolladores; se basa en el uso de comandos

⁴ La API RESTful es una interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet.

HTTP para realizar la mayoría de sus operaciones. Sus características principales son la definición del modelo REST para la elaboración de las API's, utiliza encriptación SSL⁵, el lenguaje de comunicación es independiente al lenguaje de desarrollo de los microservicios y este tipo de API's permiten crear una aplicación web con operaciones CRUD (Create, Retrieve, Update, Delete).

Algunos términos importantes para comprender como trabajan las API's:

- **Recursos.** Son objetos que tienen una funcionalidad en el sistema, los cuales, en la mayoría de los casos están relacionados con información de la base de datos. Dichos objetos pueden ser manipulados a través de métodos que operan con estos.
- **Colecciones.** Representan un conjunto de objetos, por ejemplo, si en la base de datos se maneja información de diferentes personas, el objeto "Usuario" representaría la información de una sola persona, y la colección sería "Usuarios" representando un conjunto de personas.
- **URL (Uniform Resource Locator).** Representa una ruta por la cual se pueden ubicar los recursos y también puede indicar las operaciones a realizar con dichos recursos.

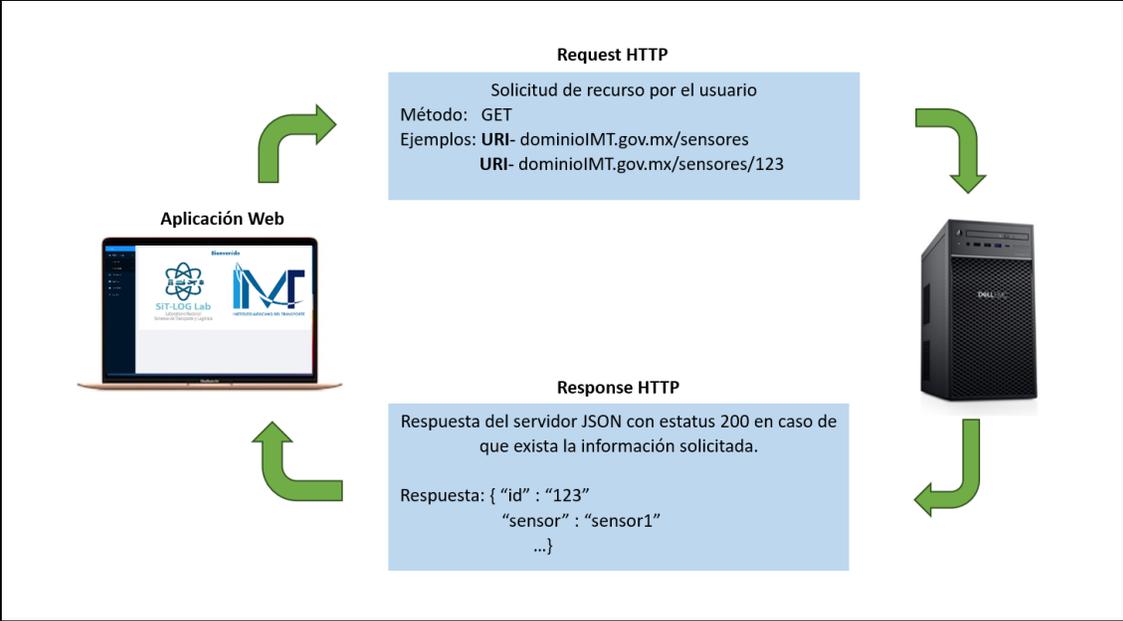
3.6 Códigos de estado

Para el desarrollo de las API's de cada microservicio, fue necesario seguir algunas normativas dependiendo del estado de respuesta de las peticiones. El Grupo de Trabajo de Ingeniería de Internet⁶ (Internet Engineering Task Force; IETF, 2022) definió un conjunto de códigos que explican la respuesta que los sistemas web deberían devolver en caso de ciertas situaciones.

Para este proyecto se utilizaron solo los métodos GET y POST, con ajustes para que el método POST pudiera realizar las operaciones de actualizar y borrar. En las Figuras 3.3 y 3.4 se puede observar la interacción de estas operaciones.

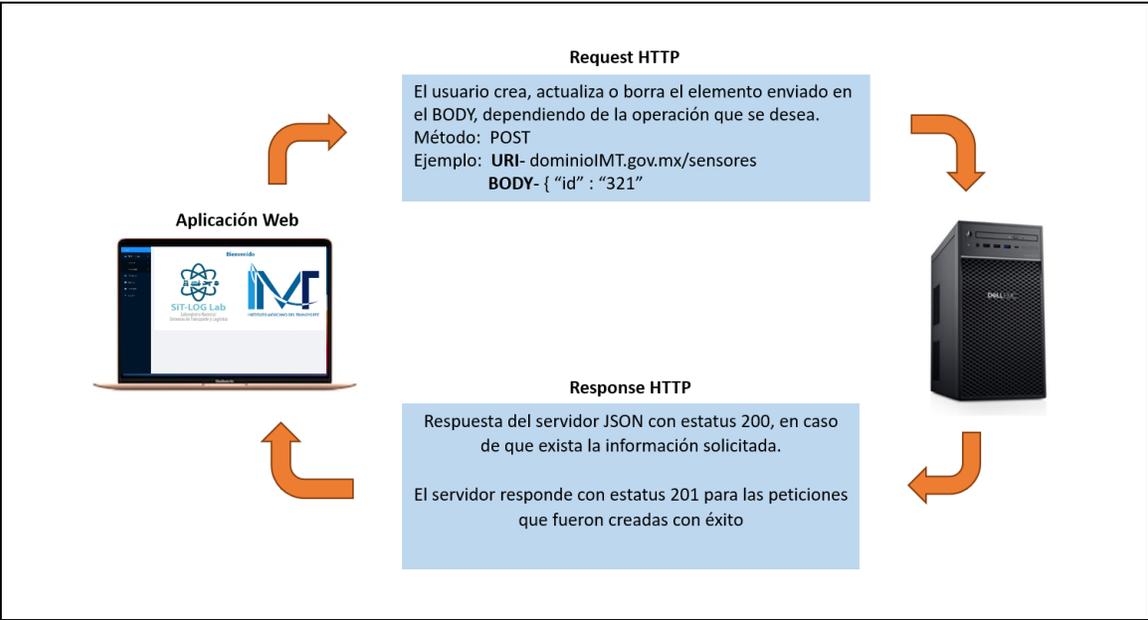
⁵ La Capa de Sockets Seguros (*Secure Sockets Layer*) es un protocolo para navegadores web y servidores que permite la autenticación y encriptación de datos enviados a través de Internet (The OpenSSL Project Authors, 2021).

⁶ Es el organismo internacional encargado de desarrollar los estándares de Internet.



Fuente: Elaboración propia.

Figura 3.3 Servicio web para una solicitud GET



Fuente: Elaboración propia.

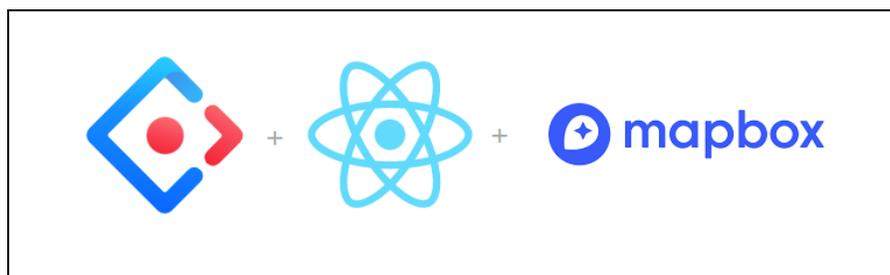
Figura 3.4 Servicio web para una solicitud POST

4. Desarrollo de la aplicación

Tomando como base las arquitecturas IoT expuestas en el capítulo anterior, se presenta el diseño de un modelo funcional con la descripción de sus componentes para la implementación y aplicación de pruebas piloto. Dichos componentes incluyen:

- La biblioteca basada en Javascript: **ReactJS**, la cual, al estar enfocada a la visualización, facilitó la creación de componentes interactivos y reutilizables durante el desarrollo de las interfaces.
- La biblioteca “**React UI antd**” para crear de manera interactiva las interfaces de usuario se eligieron diversos componentes como tablas, gráficas, tarjetas, entre otros.
- Uso del proveedor de mapas “**Mapbox**” para mostrar la ubicación y comportamiento de las bahías en la aplicación web.

En la figura 4.1 se muestran los logotipos referentes a dichas tecnologías.



Fuente: extraídas de su sitio web.

Figura 4.1 Logotipos de las tecnologías de desarrollo.

4.1 El modelo funcional

A continuación, una breve explicación de los componentes que integran el modelo funcional. El código de la aplicación web consiste en una serie de solicitudes tipo **GET** hacia la API usando el método **Fetch**⁷ escrito en **React hooks**⁸. Esto permitió separar el código de programación en varios componentes para construir la aplicación, y con ello, agilizar las solicitudes a la API.

⁷ Método que proporciona una forma fácil y lógica de obtener recursos de forma asíncrona por la red.

⁸ El uso de las funciones “*Hooks*” permite reutilizar la lógica de un componente de una manera más práctica.

En la Figura 4.2 se muestra el código de un componente basado en la lógica de React⁹ para la obtención de los eventos de cada sensor. El resto de los componentes desarrollados son similares al mostrado en la imagen, solo varía el conjunto de información obtenida desde la API.

```
useEffect(() => {
  async function component() {
    await fetch('http://localhost/IoT/evento?operacion=distintostatus')
      .then(response => {
        if (!response.ok) {
          throw new Error(`HTTP error: ${response.status}`);
        }
        return response.json();
      })
      .then(json => {
        setDataOperacion(json);
      })
      .catch(error => {
        console.error(`El servidor no responde: ${error}`);
      });
  }
}, [])
```

Fuente: Estudios de campo realizados por el IMT.

Figura 4.2 Codificación de un componente en React.

Una vez obtenidos los eventos de los sensores, se dividieron en dos grupos: operación y alarmas. El primero hace referencia a los eventos: libre y ocupado de un usuario válido. Caso contrario sucede en las notificaciones de alarmas, donde la bahía está siendo ocupada por alguien que no se reconoce como válido.

Los datos relevantes a mostrar de los dispositivos fueron la fecha y hora en que se detonó el evento, así como el nombre de la bahía y su evento en cuestión.

Otra petición que resultó importante obtener desde la API fue, la consulta de los datos históricos de los dispositivos (acumulados desde el día en que se dieron de alta los sensores en la base de datos). Esto incluye: el nombre de la bahía, la fecha y hora (de inicio y fin) del incidente ocurrido.

A partir de esta información se programó un componente que permitió agrupar las bahías existentes. Además, se calculó un conteo por tipo de incidente y su proporción. En la figura 4.3 se muestra parte del código de este proceso.

⁹ React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

```
async function fetchDataJSON() {
  await fetch('http://localhost/IoT/historico', { method: 'GET' })
    .then(response => {
      if (!response.ok) {
        throw new Error(`HTTP error: ${response.status}`);
      }
      return response.json();
    })
    .then(json => {
      dataGrafica.push(json);
      function groupBy(list, keyGetter) {
        const map = new Map();
        list.forEach((item) => {
          const key = keyGetter(item);
          if (!map.has(key)) {
            map.set(key, [item]);
          } else {
            map.get(key).push(item);
          }
        });
        return map;
      }
      const agruparPorBahias = groupBy(json, data => data.bahia);
      const Bahia = {
        nombre: "",
        incidentes: [],
        init: function (nombre, listaIncidentes) {
          this.nombre = nombre;
          this.incidentes = listaIncidentes;
        },
        setNombre: function (nombre) {
          this.nombre = nombre;
        },
        getPromedioPorIncidente: function (incidente) {
          var sum = 0;
          this.incidentes.get(incidente).forEach(value => {
            sum = sum + value.tiempo;
          });
          return sum / this.getConteoTotal(incidente);
        },
        getConteoPorIncidente: function (incidente) {
          return this.incidentes.get(incidente).length;
        },
        getConteoTotal: function () {
```

Fuente: Elaboración propia.

Figura 4.3 Codificación del componente de consulta de datos históricos.

5. Resultados obtenidos

El resultado obtenido fue el desarrollo de una aplicación web con las funcionalidades descritas a continuación.

En la figura 5.1 se muestra la vista de inicio de la aplicación web. Se observa un menú en la parte lateral izquierda con la zona de estudio ubicada en el Instituto Mexicano del Transporte en la ciudad de Querétaro. Esta zona incluye las opciones de operación de todas las bahías y su analítica.



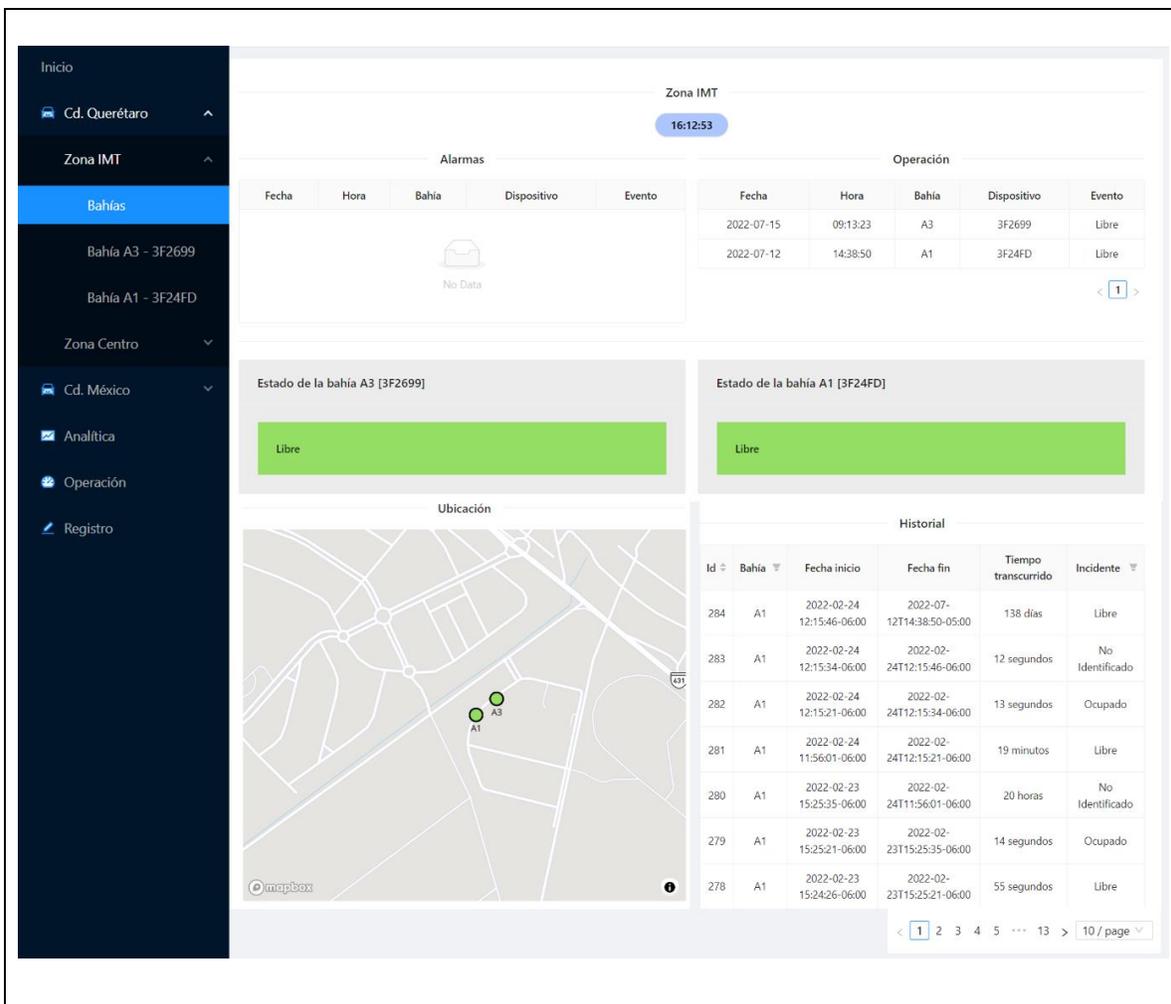
Fuente: Elaboración propia.

Figura 5.1 Vista inicial de la aplicación web.

En la figura 5.2 se muestra la operación de las dos bahías de prueba. En la parte superior izquierda se encuentra una tabla de “Alarmas”, donde se cargarán los eventos que requieren una pronta atención, es decir, se podrá visualizar aquellos usuarios “no identificados” que estacionen su vehículo en alguna de las bahías.

En la tabla identificada como “Operación” se muestran los eventos que son considerados como acciones normales, estas son el evento “libre” y usuario “validado”. Posteriormente, en la parte media se muestran ambas bahías. Dependiendo del evento que se encuentre activo, se mostrará con un color diferente, por ejemplo, libre en verde, ocupado en naranja, validado amarillo y no identificado como rojo.

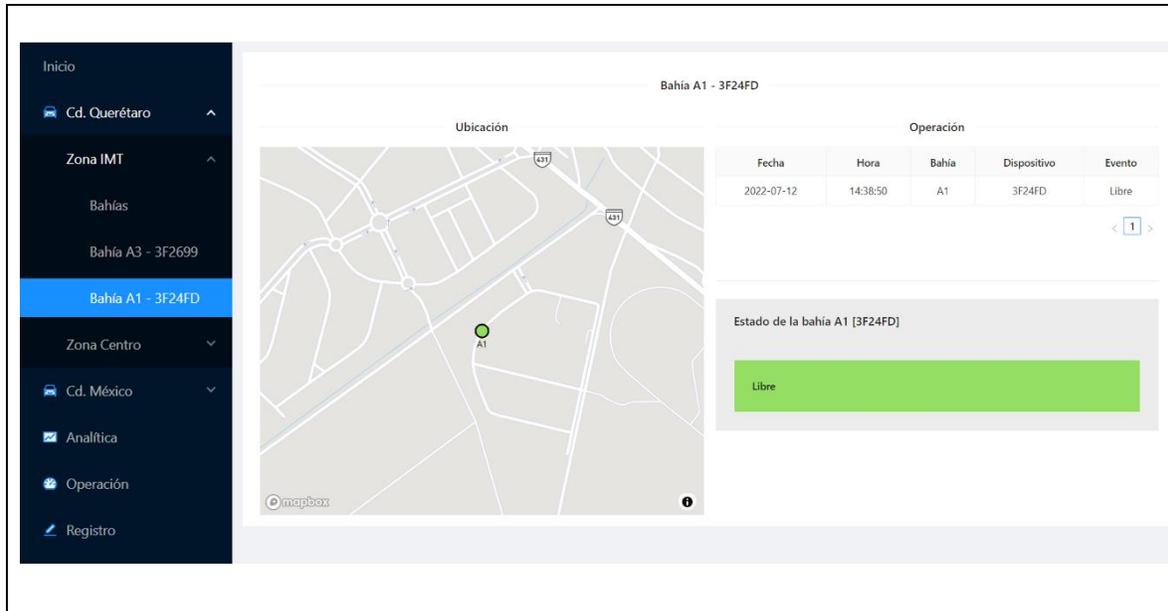
También se aprecia un mapa con la ubicación de las bahías y su color que cambia dependiendo del evento. La tabla de “Historial” carga los registros por evento de las bahías, la fecha inicio y fin en que se detonó el evento y su duración.



Fuente: Elaboración propia.

Figura 5.2 Vista de todas las bahías.

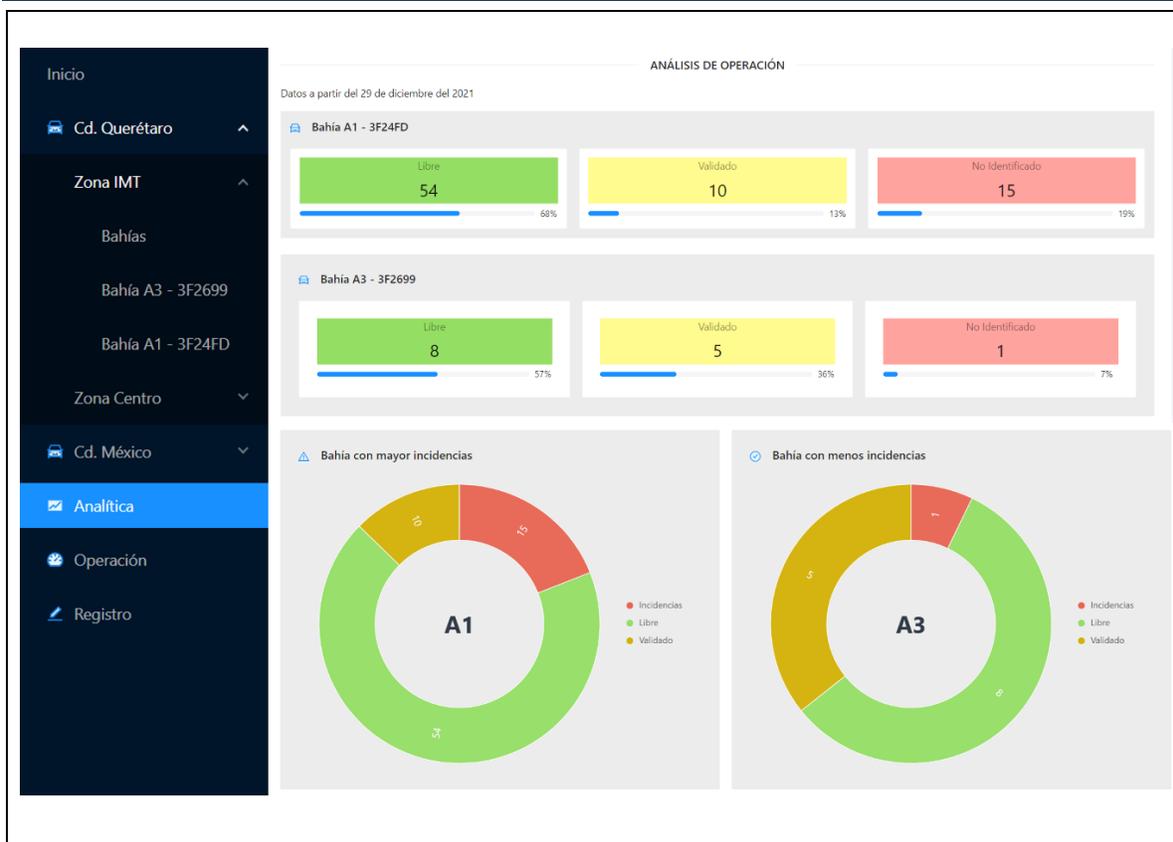
En la figura 5.3 se muestra de manera individual cómo opera una de las bahías. En este caso el estado de la bahía se encuentra libre y se observa la fecha y hora en que sucedió el evento. De igual forma, se apoya su visualización con la identificación de colores por evento.



Fuente: Elaboración propia.

Figura 5.3 Vista de la sección por bahía

Para finalizar, en la figura 5.4 se muestra la sección de "Análisis" de la operación de las bahías. Se considera que esta vista puede ser útil para los tomadores de decisiones, ya que con la información que se presenta, pueden conocer el conteo de eventos y su proporción. Por ejemplo, en la bahía A1 han sucedido 54 eventos como "Libre" 10 como "Validado" y 15 "Incidencias". Con el apoyo de los gráficos que se muestran en la parte de abajo, se deduce que la bahía A1 es la que al momento ha tenido más incidencias.



Fuente: Elaboración propia.

Figura 5.4 Vista de la sección “Analítica”

Es importante mencionar que este tipo de tecnología favoreció el desarrollo ágil y rápido del producto final. Por ello, se pretende agregar más zonas de estudio y con ello, seguir alimentando su operación. Los datos recabados y otras variables quizás no consideradas continuarán enriqueciendo el análisis, interpretación y presentación de la información.

Conclusiones

Se observó que el uso de la tecnología IoT aportó diversas ventajas para la elaboración de la solución aquí expuesta. Por un lado, las métricas obtenidas al ser utilizadas en tiempo real, favorecen la mejora de la operación logística, pero también la gestión de la infraestructura. Por otro lado, la adecuada programación de los dispositivos permite que estos tengan un ahorro energético, por lo tanto, los dispositivos pueden operar por varios años y, en consecuencia, su costo se prorratea, dando como resultado una solución viable.

Asimismo, al aprovechar el gran potencial que tiene el uso de software libre, se redujeron los costos de desarrollo. Desde la base datos, el API y la aplicación web fueron desarrollados bajo el esquema de bibliotecas de código abierto, que son mantenidas por grandes compañías y la comunidad de software libre.

Finalmente, se comprobó que la automatización en la operación de la infraestructura es posible de una forma efectiva y confiable. Contar con este tipo de herramientas para ordenar el uso de bahías de carga y descarga, permitirá en un futuro, controlar los tiempos de operación. Además, se puede obtener una visión global de lo que sucede, desde que llega un vehículo (al espacio de estacionamiento), el tiempo que dura en él y cuándo se va.

Debido a que no existe intervención humana, se reduce la posibilidad de falla, pero, además, provee en tiempo real, información tanto a la dirección logística de las empresas, como a los gobiernos, con lo cual, estos últimos tienen información para ejercer el cumplimiento de los reglamentos de tránsito, gestionar mejor la infraestructura y finalmente, poner en práctica políticas de movilidad más sostenibles en donde el transporte de carga también es tomado en cuenta.

Uno de los retos a futuro es implementar una prueba piloto dentro de la zona urbana definida por el LogistiX-Lab, usando la tecnología IoT, aquí descrita. Se pretende instrumentar y monitorear un conjunto de diez bahías de carga/descarga.

Bibliografía

Microsoft, Hypothesis Group, (2021). IoT Signals Edition 3.

Instituto federal de Telecomunicaciones (IFT). (2019). Análisis exploratorio de la comercialización de servicios de conectividad para IoT (Internet of Things). [Artículo en línea]. Disponible en <http://www.ift.org.mx/sites/default/files/contenidogeneral/estadisticas/analisisexploratorioiot.pdf>

Hernández Sánchez, B., Barrón, M. y Cedillo, M. (2021). *IoTx Arquitectura Tecnológica integrada (Fase 1)*. Publicación Técnica N° 664. México: Instituto Mexicano del Transporte.

Cedillo-Campos, M. Fransoo, J. (2019). Distribución urbana inteligente de mercancías. *Revista IC*, Julio, Colegio de Ingenieros Civiles de México, México [En línea]: https://issuu.com/cicm_oficial/docs/ic599-final

Fransoo, J., and Cedillo-Campos, M. (2022). Estimating the benefits of dedicated unloading bays by field experimentation. *Transportation Research Part A: Policy and Practice*. Vol. 160, June 2022, pp. 348-354.

Lora Alliance (2021), plataforma de conectividad para aplicaciones de Smart City. [Artículo en línea]. Disponible en <https://lora-alliance.org/>.

Kamal R. (2017). *Internet of Things Architecture and Design Principles*. India: McGraw Hill.

Savy, M. (2016). European Urban Freight: a comprehensive approach. In: Wolmar, C. (Ed.) *Urban Freight for Livable Cities*, The Volvo Research and Educational Foundations, VREF, Göteborg.

SigFox (2022). SIGFOX TECHNOLOGY. [Artículo en línea]. Disponible en <https://www.sigfox.com/en/what-sigfox/technology>.

Shanzhi, C. Hui, X., Dake L. (2014). A Vision of IoT: Applications, challenges, and opportunities with China perspective. *IEEE Internet of Things Journal*.

UN Habitat (2020). Reporte anual de las Naciones Unidas, programa de asentamientos humanos: Por un mejor futuro urbano.

Sánchez Ramos, I. (2017). *Las "Smart Cities": un nuevo paradigma. Aspectos éticos*. Tesis Doctoral, Universidad Rey Juan Carlos, Madrid España.

Boyes, H., Hallaq, B., Cunningham, J., Watson, T. (2018). *The industrial internet of things (IIOT): An analysis framework*. Computer Industry.

Gaur, A., Scotney, B., Parr, G., McClean, S. (2015). *Smart City architecture and its applications based IoT*. Science Direct.

Harris, Ch. (2022). *Desarrollo de software. Comparación entre la arquitectura monolítica y la arquitectura de microservicios*. Atlassian [En línea]: <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>.



COMUNICACIONES

SECRETARÍA DE INFRAESTRUCTURA, COMUNICACIONES Y TRANSPORTES



Km 12+000 Carretera Estatal 431 “El Colorado Galindo”
Parque Tecnológico San Fandila, Mpio. Pedro Escobedo,
Querétaro, México. C.P. 76703
Tel: +52 (442) 216 97 77 ext. 2610
Fax: +52 (442) 216 9671

publicaciones@imt.mx

<http://www.imt.mx/>