



CARRERA DE ESPECIALIZACIÓN EN INTERNET DE LAS COSAS

MEMORIA DEL TRABAJO FINAL

Diseño e implementación de una central operativa para el control y monitoreo en el material rodante

Autor:

Fernando Julio Iglesias

Director:

Ing. Fernando Lichtschein (FIUBA)

Jurados:

Nombre del jurado 1 (pertenencia)

Nombre del jurado 2 (pertenencia)

Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires,
entre marzo de 2022 y agosto de 2023.*

Resumen

En la presente memoria se describe el desarrollo, para Trenes Argentinos, de una arquitectura modular basada en el paradigma de Internet de las Cosas (IoT) que permite visualizar y gestionar, por un supervisor u operario, las formaciones ferroviarias de forma remota desde una central operativa. De esta manera, se logra mejorar la seguridad, la eficiencia y aumentar la flexibilidad del sistema ferroviario.

El sistema se trata de una solución fullstack embebida en donde se expone cada uno de los componentes de hardware y software involucrados. Entre los conocimientos aplicados se destacan el diseño de una arquitectura cliente-servidor, el diseño de una API GraphQL, la interacción con un broker de mensajes y la seguridad en las comunicaciones.

Índice general

Resumen	I
1. Introducción general	1
1.1. La Internet de las Cosas y las formaciones ferroviarias	1
1.2. Motivación	2
1.3. Estado del arte	2
1.4. Alcance y objetivos	4
2. Introducción específica	5
2.1. Protocolos de comunicación utilizados	5
2.1.1. Protocolos de comunicación	5
2.2. Tecnologías <i>full-stack</i>	5
2.2.1. Tecnologías del <i>front-end</i>	6
TypeScript	6
React	6
Parcel	6
Apollo GraphQL	7
Material UI	7
JWT Decode	7
2.2.2. Tecnologías del <i>backend</i>	7
Kotlin	7
Ktor	7
Kafka	7
Hasura	8
PostgreSQL	8
Mosquitto Broker	8
2.2.3. Tecnologías del <i>firmware</i>	8
C lang	8
FreeRTOS	8
Paho MQTT <i>client</i>	8
2.2.4. Herramientas utilizadas	8
Docker	8
Docker Compose	9
Redpanda	9
Git	9
MQTT.fx	9
CLion	9
Wireshark	9
Herramientas del navegador de internet	9
JTAG	10
ST-Link	10
STM32CubeMX	10

STM32CubeProgrammer	10
Bibliografía	11

Índice de figuras

2.1. Arquitectura de la pila de protocolos <i>TCP/IP</i>	6
--	---

Índice de tablas

2.1. Protocolos de comunicación utilizados en cada capa del <i>stack TCP/IP</i>	5
---	---

Capítulo 1

Introducción general

En este capítulo se presenta una breve descripción sobre la inserción de la Internet de las cosas (*IoT*) [1] en el ámbito ferroviario, abarcando las motivaciones que llevaron a la realización de este trabajo, el estado del arte en cuanto a esta temática y además, los alcances y objetivos del trabajo.

1.1. La Internet de las Cosas y las formaciones ferroviarias

La Internet de las Cosas (*IoT*, por sus siglas en inglés) es una tecnología que se ha convertido en una tendencia en la actualidad. Este concepto se estableció en el mercado en la década de 1990, aunque su verdadero auge comenzó en los últimos años gracias a la aparición de dispositivos interconectados y al aumento en el uso de internet. La *IoT* se enfoca en la interconexión de objetos cotidianos con la red, lo que permite recopilar datos en tiempo real, optimizar procesos y tomar decisiones basadas en la información recolectada. Algunos beneficios que la *IoT* ofrece son una mayor eficiencia, una reducción de costos, un mejor monitoreo y control, y una mejor calidad de vida para las personas.

En el sector ferroviario, la *IoT* ha demostrado ser una tecnología clave para mejorar la eficiencia y la seguridad de las formaciones ferroviarias. Las formaciones ferroviarias pueden ser equipadas con sensores y dispositivos interconectados que permiten recopilar información sobre la ubicación, velocidad, consumo de energía, estado mecánico y otros aspectos importantes. Esta información se puede enviar en tiempo real a la red, lo que permite una mejor gestión del tráfico ferroviario y una toma de decisiones más rápida y eficaz. Además, la *IoT* también puede ayudar a prevenir accidentes ferroviarios al detectar problemas mecánicos antes de que se conviertan en un riesgo para la seguridad.

1.2. Motivación

El sistema ferroviario de la República Argentina cuenta con una gran cantidad de formaciones ferroviarias en las que se encuentran diferentes sistemas de seguridad a bordo. Estos equipos se encargan de supervisar el correcto funcionamiento de los subsistemas críticos. Ante una falla en uno de los subsistemas, una formación ferroviaria se detiene inmediatamente por la activación automática de las señales de corte de tracción (CT) y freno de emergencia (FE). En esta situación, el conductor debe llevar la formación a un lugar seguro para que los pasajeros puedan descender y, posteriormente, trasladarla a un taller para que pueda ser reparada.

El SAL/T, según sus siglas, Sistema de Aislamiento Limitado y Total [1], es un dispositivo del cual se cuenta con una primera versión prototipada; que se presenta como solución a las contingencias descritas anteriormente. De esta manera, el maquinista de una formación ferroviaria cuenta con la posibilidad de activar y desactivar el modo aislado limitado. En este modo, el equipo permite la circulación de la formación al desactivar las señales de corte de tracción y freno de emergencia generadas por los subsistemas críticos. Para que esta operación se complete de forma segura, se debe monitorear la velocidad de la formación tal que seaposible garantizar que no se supere cierto valor máximo.

A partir del desarrollo previo del prototipo, se presentan los avances en la implementación de una central operativa que centraliza los dispositivos SAL/T para su respectiva administración, configuración y monitoreo en tiempo real de la información recibida y transmitida desde una plataforma digital. El proyecto se desarrolla por CONICET-GICSAFe [2] para la empresa Trenes Argentinos [3].

Los subsistemas asociados al SAL/T, como la seguridad de puertas, el sistema de hombre vivo y la protección de coche a la deriva, son críticos debido a que, en caso de fallar, pueden ocasionar lesiones o muertes de personas e incluso generar pérdidas materiales.

La central operativa permite la administración y configuración en forma remota de los dispositivos de supervisión de seguridad de cada formación ferroviaria, la visualización de los diferentes parámetros de interés involucrados por las personas asignadas dentro de una entidad y de este modo es posible optimizar la toma de decisiones.

1.3. Estado del arte

A partir del análisis en las últimas tendencias referidas a la central operativa SAL/T, se listan aquellas herramientas, de gestión y control de dispositivos de seguridad en el sector ferroviario, que presentan características similares al producto propuesto:

1. *Indra*[4]: ofrece una solución integral de seguridad para el sector ferroviario, que incluye una plataforma de gestión centralizada y un sistema de supervisión remota de dispositivos de seguridad.
2. *Thales Group*[5]: desarrolla soluciones de seguridad para el sector ferroviario, que incluyen una aplicación de gestión y control centralizada para supervisar y configurar dispositivos de seguridad de forma remota.

3. *Alstom*[\[6\]](#): ofrece una plataforma de gestión y control para el sector ferroviario, que permite la supervisión y configuración de dispositivos de seguridad de forma remota, así como el análisis en tiempo real de los datos recopilados por estos dispositivos.
4. *Siemens Mobility*[\[7\]](#): ofrece una solución de gestión y control para el sector ferroviario, que incluye una aplicación web de central operativa para supervisar y configurar dispositivos de seguridad de forma remota, así como un sistema de análisis de datos en tiempo real.

1.4. Alcance y objetivos

El sistema resultante de este trabajo se encuentra destinado al desarrollo del software para una central operativa que permite administrar y configurar de forma remota dispositivos de supervisión de seguridad de formaciones ferroviarias denominados *SAL/T* (Sistema de Aislamiento Limitado/Total).

El alcance del trabajo comprende los siguientes puntos:

- Servicio de monitoreo y control: visualizar en tiempo real los datos recibidos y enviar comandos de control a los dispositivos activos.
- Gestión de configuración de dispositivos: visualizar y modificar los parámetros configurables de los dispositivos.
- Base de datos: registrar la información recibida desde todos los dispositivos.
- Componentes de seguridad: brindar seguridad a las comunicaciones y a la información almacenada.
- Gestión de usuarios y perfiles: administrar roles y permisos de acceso.
- Realizar la migración de un microcontrolador de la familia *Cortex* tipo M [8] a uno de la misma familia que disponga de características destinadas a la seguridad funcional del sistema.

Capítulo 2

Introducción específica

AGREGAR TEXTO DESCRIPTIVO SOBRE LA INTRO ESPECIFICA

2.1. Protocolos de comunicación utilizados

Comenzar escribiendo que aqui se detallaran los protocolos utilizados segun el modelo TCP/IP y hablar sobre el diagrama

AGREGAR UNA IMAGEN CON LOS 3 ENTES INTERCONECTADOS BAJO EL MODELO TCP/IP Nucleo F429ZI Board <—>Dedicated Server <—>Web Client

2.1.1. Protocolos de comunicación

De los modelos *stack IoT* y *lwIP* [9], basados en el stack *TCP/IP* [10], se emplean los siguientes protocolos de comunicación tal como se puede observar en la tabla 2.1.

TABLA 2.1. Protocolos de comunicación utilizados en cada capa del *stack TCP/IP*

Capa	Protocolo
Capa de Aplicación	HTTP [11], DHCP [12] y MQTT [13]
Capa de Transporte	UDP [14]
Capa de Red	IPv4 [15], ARP [16]
Capa de Enlace de Datos	Ethernet [17]

Cabe destacar que ambos modelos se encuentran diseñados para sistemas con recursos limitados y buscan garantizar una correcta comunicación entre los dispositivos conectados en la red.

2.2. Tecnologías *full-stack*

A partir de los modelos más relevantes que se encuentran establecidos para el desarrollo e implementación de aplicaciones de software se consideran el *stack web* y el *stack IoT*, tal como se puede ver en la figura 2.1. A pesar de las similitudes entre ambos, en la aplicación el *stack web* emplea el protocolo *HTTP* mientras que el *stack IoT* utiliza el protocolo *MQTT*; siendo este último el más adecuado para escenarios donde son limitados los recursos como el ancho de banda y el consumo de energía.

En especial, el protocolo *MQTT* dispone de mensajes más livianos y además es posible transmitir y/o recibir datos en formato binario sin la necesidad de una codificación previa. También, este protocolo permite asignar niveles de calidad de servicio (*QoS* [*ibm-qos*]) a los mensajes transmitidos, resultando una característica primordial en aplicaciones donde la probabilidad de pérdidas de paquetes es considerable.

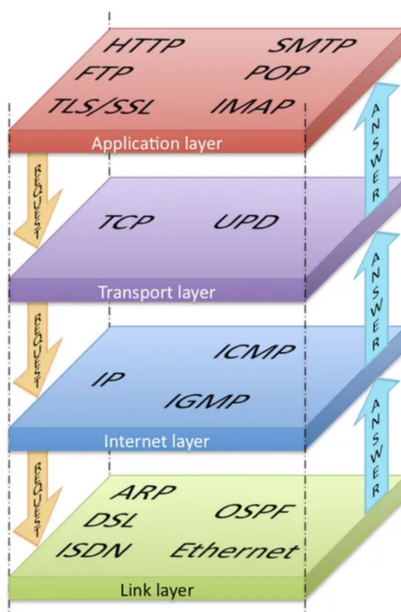


FIGURA 2.1. Arquitectura de la pila de protocolos TCP/IP.

2.2.1. Tecnologías del *front-end*

TypeScript

TypeScript [18] es una extensión de código abierto de *JavaScript* que agrega tipos estáticos opcionales y características de programación orientada a objetos avanzadas, lo que permite una mayor seguridad y mantenibilidad en el código.

React

React [19] es una biblioteca de *JavaScript* para construir una interfaz de usuario reutilizable e interactiva, desarrollada por *Facebook* y ampliamente utilizada en el desarrollo web. Usa un enfoque basado en componentes y el *DOM* virtual para mejorar el rendimiento y es altamente integrable con otras bibliotecas y frameworks.

Parcel

Parcel [20] es una herramienta de construcción de código abierto con una estrategia de cero configuración que optimiza y empaqueta módulos de *JavaScript* y otros archivos. En particular, mejora significativamente el flujo de trabajo del desarrollador al ser rápido, fácil de usar y altamente personalizable.

Apollo GraphQL

Apollo GraphQL [21] es una plataforma de código abierto para el desarrollo de aplicaciones *GraphQL*, con una amplia gama de herramientas y servicios para construir y mantener aplicaciones *GraphQL* de manera efectiva, ofreciendo una administración de caché y una amplia compatibilidad con diferentes *frameworks* y tecnologías.

Material UI

Material UI [22] es una biblioteca de componentes de interfaz de usuario de código abierto basada en Material Design de Google, ofrece componentes preconstruidos altamente personalizados y una amplia compatibilidad con diferentes tecnologías y *frameworks* para construir aplicaciones web modernas y atractivas.

JWT Decode

JWT Decode [23] es una biblioteca *JavaScript* que decodifica *tokens JWT*, con características útiles como validación de tokens y verificación de firma. Se integra fácilmente en diferentes *frameworks* de *JavaScript* como *React* y *Angular*.

2.2.2. Tecnologías del backend

Kotlin

Kotlin [24] es un lenguaje de programación de tipado estático que puede correr sobre la máquina virtual de *Java* y ser compilado en *JavaScript* y *LLVM*. También ofrece interoperabilidad con *Java*, programación funcional, orientación a objetos y manejo de nulos seguro, lo que ha llevado a su popularidad debido a su facilidad de uso y legibilidad.

Ktor

Ktor [25] es un *framework* web de *Kotlin* que permite crear aplicaciones web y *API REST* de manera fácil y eficiente, con un enfoque en la programación funcional. *Ktor* es altamente modular y personalizable, lo que significa que los desarrolladores pueden seleccionar solo los componentes que necesitan para su aplicación, reduciendo así la complejidad y el tamaño del código.

Kafka

Kafka [26] es una plataforma de *streaming* de datos que permite a las aplicaciones enviar y recibir datos en tiempo real. Se encuentra basada en el modelo de publicación/suscripción, se caracteriza por su alta velocidad y rendimiento, haciéndola ideal para aplicaciones que necesitan procesar grandes cantidades de datos en tiempo real.

Hasura

Hasura [27] es una plataforma sin servidor que permite a los desarrolladores crear y escalar aplicaciones con *GraphQL* rápidamente. Ofrece seguridad de extremo a extremo y se integra fácilmente con diferentes bases de datos para generar automáticamente una *API* de *GraphQL*.

PostgreSQL

PostgreSQL [28] es un sistema de gestión de bases de datos de código abierto y gratuito, conocido por su fiabilidad, escalabilidad y seguridad. Ofrece soporte para transacciones *ACID*, integración con lenguajes de programación populares y una amplia variedad de herramientas y extensiones.

Mosquitto Broker

Mosquitto [29] es un *broker MQTT* de código abierto utilizado en *IoT* para transmitir mensajes entre dispositivos. Es popular por su escalabilidad, facilidad de uso y características de seguridad.

2.2.3. Tecnologías del *firmware*

C lang

C [30] es un lenguaje de programación estructurado y de propósito general, ampliamente utilizado en sistemas operativos, aplicaciones de bajo nivel y dispositivos embebidos. Su sintaxis simple y directa permite escribir código claro y legible, y ofrece gran control sobre la memoria y acceso directo al hardware.

FreeRTOS

FreeRTOS [31] es un *RTOS* de código abierto y gratuito que controla sistemas embebidos y microcontroladores, utilizado en control industrial, automoción y electrónica de consumo, con capacidad de rendimiento confiable y predecible. Es altamente portátil y ofrece gestión de tareas, semáforos, colas y temporizadores para sistemas complejos y eficientes en recursos.

Paho MQTT client

Paho [32] es una biblioteca que desarrolla un cliente *MQTT* de código abierto que se utiliza para conectar aplicaciones a un *broker MQTT*. *Paho* es compatible con una amplia variedad de plataformas y lenguajes de programación, brinda una *API* sencilla para publicar y suscribir mensajes. Es ampliamente utilizado en aplicaciones de *IoT* y *M2M* para la transmisión eficiente de datos.

2.2.4. Herramientas utilizadas

Docker

Docker [33] es una plataforma de software que permite a los desarrolladores crear, desplegar y ejecutar aplicaciones en contenedores. Estos contenedores permiten que las aplicaciones se ejecuten de manera aislada del sistema operativo y otras aplicaciones, lo que facilita su portabilidad y escalabilidad.

Docker Compose

Docker Compose [34] es una herramienta que permite definir y ejecutar aplicaciones de múltiples contenedores *Docker*. Permite a los desarrolladores especificar los servicios y la configuración de cada contenedor en un archivo *YAML* [35] para simplificar la creación, ejecución y gestión de aplicaciones complejas.

Redpanda

Redpanda [36] es una plataforma de *streaming* de datos distribuida y de alto rendimiento que combina las funcionalidades de *Kafka* y *Redis*. Es una solución escalable y confiable para el procesamiento de datos en tiempo real en entornos empresariales, y es compatible con una amplia variedad de casos de uso, desde el análisis de datos hasta la inteligencia artificial y el aprendizaje automático.

Git

Git [37] es un sistema de control de versiones distribuido que se utiliza para rastrear los cambios en el código fuente de un proyecto de software. Permite a los desarrolladores trabajar en colaboración en el mismo código fuente y hacer un seguimiento de las diferentes versiones y ramas del proyecto.

MQTT.fx

MQTT.fx [38] es una herramienta de escritorio de código abierto que se utiliza para probar y depurar conexiones *MQTT*. Ofrece una interfaz gráfica de usuario fácil de usar para interactuar con *brokers MQTT* y suscribirse a temas y mensajes. *MQTT.fx* es compatible con una variedad de características de seguridad, como *TLS/SSL* y autenticación, lo que lo hace adecuado para su uso en entornos de producción.

CLion

CLion [39] es un entorno de desarrollo integrado (*IDE*) para programar en C y C++ que proporciona herramientas para la edición de código, depuración, refactorización y gestión de proyectos. Es conocido por su alta capacidad de análisis estático de código, lo que permite a los desarrolladores encontrar errores de forma eficiente.

Wireshark

Wireshark [40] es una herramienta de análisis de redes de código abierto y gratuita, utilizada para capturar y analizar paquetes de datos en tiempo real. Permite examinar el tráfico de red para identificar problemas de rendimiento, seguridad y configuración; siendo compatible con una amplia variedad de protocolos de red.

Herramientas del navegador de internet

Las herramientas del navegador de internet son características incorporadas que permiten a los usuarios analizar y modificar elementos de una página web, como su estructura *HTML*, *CSS* y *JavaScript*. Algunas herramientas comunes incluyen la consola de desarrollador, el inspector de elementos, el depurador de *JavaScript*

y el analizador de red, que ayudan a los desarrolladores a depurar problemas y mejorar la calidad de sus sitios web.

JTAG

JTAG (Joint Test Action Group) [41] es un estándar para la depuración y programación de dispositivos electrónicos. Permite acceder a los pines de prueba de un dispositivo para realizar pruebas de hardware y depuración a nivel de circuito. *JTAG* se ha convertido en un estándar común para la depuración y programación de dispositivos, y muchas herramientas de desarrollo de software y hardware lo admiten.

ST-Link

ST-Link [42] es una herramienta de programación y depuración de microcontroladores fabricada por *STMicroelectronics*. Se utiliza para programar y depurar microcontroladores *STM32* y otros dispositivos compatibles con *JTAG* o *SWD*. La herramienta se conecta al ordenador mediante *USB* y se puede usar con una variedad de entornos de desarrollo integrados (*IDE*) y herramientas de depuración.

STM32CubeMX

STM32CubeMX [43] es una herramienta de configuración gráfica para dispositivos *STM32* que permite a los desarrolladores generar automáticamente código inicial para su proyecto. Ofrece una interfaz de usuario intuitiva que simplifica la configuración de periféricos y pines, además, admite una variedad de opciones de generación de código.

STM32CubeProgrammer

STM32CubeProgrammer [44] es una herramienta de programación y actualización de firmware para dispositivos *STM32* que permite programar y depurar dispositivos *STM32*, así como actualizar su firmware en campo. Es compatible con una variedad de interfaces de programación, como *JTAG*, *SWD* y *UART*, también, es fácil de usar gracias a su interfaz gráfica de usuario.

Bibliografía

- [1] Investopedia. Online. 2023. URL: <https://www.investopedia.com/terms/i/internet-things.asp>.
- [2] CONICET-GICSAFe. Online. 2023. URL: <https://sites.google.com/view/conicet-gicsafe/inicio>.
- [3] Trenes Argentinos. Online. 2023. URL: <https://www.argentina.gob.ar/transporte/trenes>.
- [4] Indra. Online. 2023. URL: <https://www.indracompany.com>.
- [5] Thales. Online. 2023. URL: <https://www.thalesgroup.com>.
- [6] Alstom. Online. 2023. URL: <https://www.alstom.com/>.
- [7] Siemens Mobility. Online. 2023. URL: <https://www.mobility.siemens.com/>.
- [8] arm. Online. 2023. URL: <https://developer.arm.com/Processors/Cortex-M4>.
- [9] lwIP. Online. 2023. URL: <https://savannah.nongnu.org/projects/lwip/>.
- [10] tcpip. Online. 2023. URL: <https://www.ibm.com/docs/en/aix/7.2?topic=protocol-tcpip-protocols>.
- [11] HTTP. Online. 2023. URL: <https://www.ietf.org/rfc/rfc2616.txt>.
- [12] DHCP. Online. 2023. URL: <https://www.rfc-editor.org/rfc/rfc2131>.
- [13] MQTT. Online. 2023. URL: <https://mqtt.org/>.
- [14] UDP. Online. 2023. URL: <https://www.rfc-es.org/rfc/rfc0768-es.txt>.
- [15] IPv4. Online. 2023. URL: <https://www.rfc-editor.org/rfc/rfc791>.
- [16] ARP. Online. 2023. URL: <https://tools.ietf.org/html/rfc826>.
- [17] Ethernet. Online. 2023. URL: <https://www.rfc-editor.org/rfc/rfc894>.
- [18] TypeScript. Online. 2023. URL: <https://www.typescriptlang.org/>.
- [19] React. Online. 2023. URL: <https://reactjs.org/>.
- [20] Parcel. Online. 2023. URL: <https://parceljs.org/>.
- [21] Apollo GraphQL. Online. 2023. URL: <https://www.apollographql.com/>.
- [22] Material UI. Online. 2023. URL: <https://mui.com/>.
- [23] JWT Decode. Online. 2023. URL: <https://jwt.io/>.
- [24] Kotlin. Online. 2023. URL: <https://kotlinlang.org/>.
- [25] Ktor. Online. 2023. URL: <https://ktor.io/>.
- [26] Kafka. Online. 2023. URL: <https://kafka.apache.org/>.
- [27] Hasura. Online. 2023. URL: <https://hasura.io/>.
- [28] PostgreSQL. Online. 2023. URL: <https://www.postgresql.org/>.
- [29] Eclipse Mosquitto. Online. 2023. URL: <https://mosquitto.org/>.
- [30] Lenguaje C. Online. 2023. URL: <https://www.iso.org/standard/74528.html>.
- [31] FreeRTOS. Online. 2023. URL: <https://www.freertos.org/>.
- [32] Paho MQTT Client. Online. 2023. URL: <https://www.eclipse.org/paho/>.
- [33] Docker. Online. 2023. URL: <https://www.docker.com/>.
- [34] Docker Compose. Online. 2023. URL: <https://docs.docker.com/compose/>.
- [35] yaml. Online. 2023. URL: <https://yaml.org/>.
- [36] Redpanda. Online. 2023. URL: <https://vectorized.io/>.

- [37] Git. Online. 2023. URL: <https://git-scm.com/>.
- [38] MQTT.fx. Online. 2023. URL: <https://softblade.de/en/mqtt-fx/>.
- [39] CLion. Online. 2023. URL: <https://www.jetbrains.com/es-es/clion/>.
- [40] Wireshark. Online. 2023. URL: <https://www.wireshark.org/>.
- [41] JTAG. Online. 2023. URL: <https://www.jtag.com/>.
- [42] ST-Link. Online. 2023. URL:
<https://www.st.com/en/development-tools/st-link-v2.html>.
- [43] STM32CubeMX. Online. 2023. URL:
<https://www.st.com/en/development-tools/stm32cubemx.html>.
- [44] STM32CubeProgrammer. Online. 2023. URL:
<https://www.st.com/en/development-tools/stm32cubeprog.html>.