

Uso de un contenedor inteligente en la logística del traslado de mercadería

Ciro Edgardo Romero

Departamento de Investigación, Desarrollo e Innovación

C & S Informática S. A

Buenos Aires, Argentina

cromero@cys.com.ar

Abstract—En varios sistemas de seguimiento de envíos, se utilizan diferentes dispositivos para agregar precisión. Es común utilizar GPS integrados a vehículos o aplicaciones de celular, donde se envíe información relevante. En este esquema de funcionamiento, se realiza el seguimiento de medio de transporte, o de las personas ocupadas en realizar los traslados. El objetivo del presente artículo, es describir un sistema de seguimiento, que realice la supervisión de bienes, de informa independiente del medio por el cual es trasladado.

Palabras Clave—Internet de las Cosas, IOT, Logística

I. INTRODUCCION

En un sentido clásico, el propósito de la trazabilidad es el seguimiento activo desde el lugar de producción hasta el destino. El objetivo de este seguimiento es identificar los orígenes y estudiar las distintas etapas de todo el proceso de distribución hasta llegar al consumidor final [3]. Los avances en la tecnología IoT han ayudado a las empresas involucradas en actividades logísticas a comenzar a utilizar dispositivos inteligentes para transportar mercancías. Estos dispositivos permiten recopilar, procesar y enviar datos relacionados con el traslado de mercancías a un software especializado, creando así un sistema de seguimiento inteligente. Gracias a estos avances tecnológicos, el prototipo presentado en este artículo, logra ser un sistema de seguimiento de productos y no del vehículo que lo traslada. El mismo, está basado en el proyecto final de posgrado del autor y el departamento de I+D+i de C&S [14].

A. Posibles aplicaciones

Se pueden monitorear las variables ambientales internas que afectan el estado sensorial de los productos de envío biodegradables. Condiciones similares fuera del contenedor pueden afectar negativamente al propio contenedor y dañar su contenido [11]. Además, la exposición a la luz puede cambiar ciertos productos, como los medicamentos. La luz es una forma de energía, por lo que se convierte en calor, lo que puede afectar negativamente al recipiente y/o a su contenido [5].

Otra variable que se puede monitorear es la posición del contenedor con respecto a un plano de referencia que

representa el suelo sobre el cual se mueve el contenedor. Recordar las diferentes posiciones del envase puede ayudarte a detectar errores a la hora de manipular el envase, ya que una inclinación excesiva o movimientos muy bruscos pueden dañar el producto [16]. Finalmente, se debe conocer la ubicación geográfica del contenedor para garantizar un seguimiento detallado y preciso del seguimiento. La seguridad y la confianza se crean al poder encontrar productos en sistemas de referencia de forma transparente e identificable [3].

II. ESTRUCTURA GENERAL DEL PROTOTIPO

El prototipo cuenta con contenedor térmico que envía constantemente información de las condiciones internas y externas, recopiladas por sensores. El envío de esta información se realiza a través de Internet y es recibida por un servidor que persiste todo en una base de datos. Esto se suma a una aplicación, donde se visualizan los detalles de las mediciones realizadas durante cada etapa de cada viaje.

A. Contenedor térmico

El contenedor del sistema, posee como componente principal la placa de desarrollo TTGO T-Call. Este permite implementar conectividad Wifi e Internet móvil desde una tarjeta SIM, por medio de tecnología GSM [10]. La placa como base el SoC (*System on Chip*) ESP32-wroom-32 de la empresa

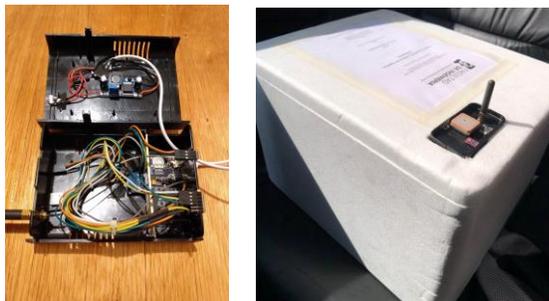
Espressif [19]. Se seleccionaron diferentes sensores y módulos para obtener las variables que serán monitorizadas durante el traslado del contenedor:

- Temperatura, presión y humedad (interna y externa): las variables en cuestión son obtenidas por medio de los módulos BMP180 y BME280, basado en los sensores homónimos, ubicados de forma externa e interna respectivamente. Por un lado, el sensor BMP180 permite medir presión barométrica y temperatura [1]. por otro lado, el sensor BME280 mide presión barométrica, humedad relativa y temperatura [2]. En ambos casos, se utiliza el protocolo de comunicación I2C para conectar los módulos con la placa de desarrollo.
- Luz (interna y externa): la medición de esta variable se realiza por medio del módulo-sensor TGMT6000

[15]. Este mismo, transmite la medición por medio de una señal analógica que la placa procesa posteriormente.

- Posición relativa: la posición se mide utilizando el módulo MPU6050, el cual es una unidad de medición inercial o IMU (del inglés *inertial measurement unit*) [6] de seis grados de libertad, que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes, fabricado por la empresa Invensense [7]. Este módulo, también se encuentra comunicado por medio de protocolo I2C.
- Ubicación: para ello se utilizó el módulo NEO-6M, parte de una familia de receptores GPS con alto rendimiento y bajo costo de la marca Ublox [18]. Este módulo, se comunica con el microcontrolador por medio de protocolo UART.

La conexión de los componentes se realiza mediante cableado simple utilizando cables prototipo de 0,5 mm de espesor y el hardware está protegido en un gabinete de plástico. Este último, se puede modificar para agregar antenas de comunicación, botones y luces indicadoras al dispositivo. Lleva incorporadas dos baterías recargables, lo que garantiza la autonomía del prototipo. Al mismo tiempo, el gabinete esta equipado con una caja aislada para mantener la temperatura interna cercana a la de un prototipo comercial. Esto puede ilustrarse en la Fig. 1 donde se puede observar el prototipo terminado, con las antenas GSM y GPS sobresaliendo de la tapa de la caja térmica.



(a) Gabinete (b) Contenedor

Fig. 1. Prototipo de contenedor inteligente

El microcontrolador incluido en el contenedor es el encargado de orquestar el funcionamiento de los diferentes módulos del sistema. El código del programa para ejecutar el funcionamiento fue escrito en MicroPython, una implementación sencilla y eficiente de Python 3 [9]. Este lenguaje incluye un pequeño subconjunto de la biblioteca estándar, reimplementada y optimizada para ejecutarse en microcontroladores.

El programa implementa las librerías de los módulos antes mencionados, utilizando dos archivos importantes

que MicroPython busca en la raíz de su sistema de archivos: *boot.py* y *main.py*. El primero se ejecuta al encender, o reiniciar, el microprocesador y contiene la configuración de la placa en el arranque. El segundo, contiene el programa de los servicios que implementan las librerías de los sensores, la inicialización de los puertos, la conexión a Internet y la lógica de envío de los datos. En la Fig. 2 se muestra el diagrama de flujo que describe el comportamiento del programa que se ejecuta dentro del contenedor.

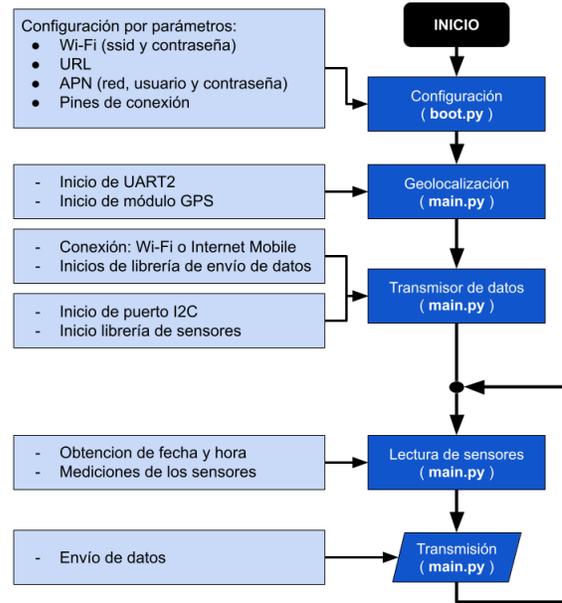


Fig. 2. Diagrama de flujo que describe el programa del contenedor

B. Recepción de datos

La lógica para enviar mediciones se construye a partir de un contenedor. Las especificaciones de esta placa son para el funcionamiento del módulo y la conexión a Internet. Utiliza métodos para la lectura de cada sensor, los guarda y agrupa en un objeto JSON declarado en el programa. Este objeto se envía como una *request* al servidor web que procesa y almacena la información. A partir de ahí, se expone una interfaz llamada API REST para recibir y almacenar datos [8]. Esta interfaz verifica los contenedores habilitados en el sistema para limitar los dispositivos que pueden interactuar con la aplicación.

C. Plataforma Mobile

La plataforma *mobile* proporciona a los usuarios la capacidad de visualizar gráficamente la información brindada por los contenedores. Dicha plataforma se desarrolló utilizando la arquitectura basada en *Context API* en React Native[13] es una opción poderosa y simplificada para gestionar el estado global de una aplicación sin necesidad de instalar librerías externas como Redux. La *Context API* de React permite compartir

datos a través de componentes sin necesidad de pasar datos manualmente en cada nivel del árbol de componentes generado en React. Esto resulta especialmente útil en aplicaciones de tamaño pequeño a mediano.

Al mismo tiempo, se implementó una base de datos para persistir la información. Por último, se utilizó la plataforma abierta conocida como Docker para facilitar la ejecución de todos los componentes [4].

- *Back-end*: desarrollado en NestJS [12], un *framework* moderno para aplicaciones *backend* de Node.js que permite a los desarrolladores crear aplicaciones robustas, eficientes y escalables. Fue diseñado para aprovechar las características de TypeScript [17] y proporcionar una arquitectura modular y fácilmente extensible.
- *Mobile*: desarrollado en React Native, *framework* diseñado para construir aplicaciones compatibles con dispositivos móviles.
- Base de datos: se utilizó una base de datos relacional *Postgres*.

En la Fig. 3 se observa un diagrama con las diferentes tecnologías utilizadas en la plataforma.

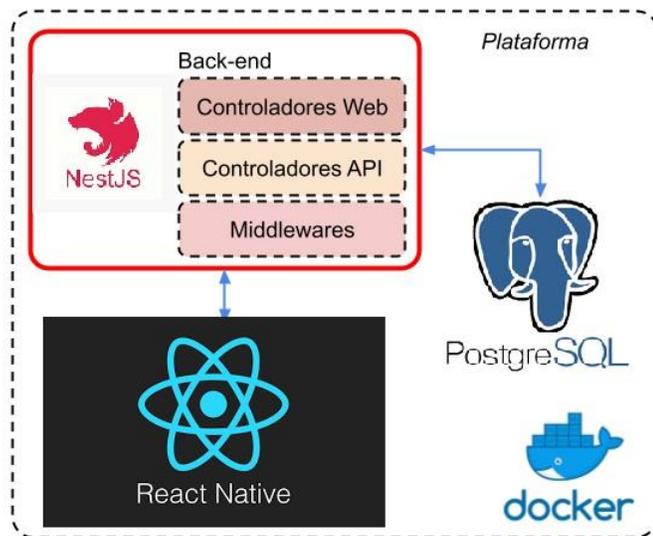


Fig. 3. Componentes de la plataforma.

Existe una capa de servicios, desarrollada para manejar la lógica de la plataforma. Se utiliza para el almacenamiento de datos y proporcionar canales de comunicación (conocidos como *endpoints*), que la aplicación *mobile* utiliza tanto para crear un nuevo envío como para consultar el estado del mismo. La comunicación entre la aplicación *mobile* y los servicios, se realizan a través de solicitudes HTTP.

III. CONCLUSIONES Y TRABAJO FUTURO

El conocimiento adquirido durante el desarrollo e implementación del presente prototipo, aportó valor a todos los *stakeholders*. Se ha alcanzado un cierto nivel de innovación en relación con los sistemas de logística y trazabilidad. Todos los requisitos se analizan deliberadamente para que el proyecto pueda ser útil a la empresa y sus colaboradores. La propuesta sirve como prueba de conceptos y, utilizando las lecciones aprendidas de este trabajo, se obtuvo suficiente conocimiento para comenzar a refinar varios aspectos del prototipo para garantizar que madure.

REFERENCES

- [1] Componentes 101. *BMP180 - Atmospheric Pressure Sensor*. <https://components101.com/sensors/bmp180atmospheric-pressure-sensor>. Mar. 2021. (Visited on 03/02/2021).
- [2] Bosch. *BME280*. <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors/bme280/>. Mar. 2021. (Visited on 03/02/2021).
- [3] Miguel Gaston Cedillo Campos y Cuauhtemoc Sanchez Ramirez. *Análisis dinámico de sistemas industriales*. Editorial Trillas, Mexico, 2008.
- [4] Docker. *Our Company*. Visitado: 2021-03-24. Mar. 2021. URL: <https://www.docker.com/company>.
- [5] L Dominguez and M Parzanese. "Luz ultravioleta en la conservación de alimentos". In: *Alimentos argentinos* 52.2 (2011), pp. 70–76.
- [6] DS Fedorov et al. "Using of measuring system MPU6050 for the determination of the angular velocities and linear accelerations". In: *Automatics & Software Engineering* 11.1 (2015), pp. 75–80.
- [7] InvenSense. *MPU-6000 and MPU-6050 Product Specification - Revision 3.4*. <https://invensense.tdk.com/wpcontent/uploads/2015/02/MPU-6000-Datasheet1.pdf>. Mar. 2021. (Visited on 03/02/2021).
- [8] Rafael Luis Granados La Paz. *Desarrollo de aplicaciones web en el entorno servidor. IFCD0210*. IC Editorial, 2015.
- [9] Micropython. *Micropython (Home)*. Visitado: 2021-03-10. Mar. 2021. URL: <https://micropython.org/>.
- [10] Michael Mouly and Marie-Bernadette Pautet. "GSM System for Mobile Communications, The". In: *Revista Zona Logística* (1992).
- [11] HECTOR Navarro. "Logística en la cadena de frío". In: *ISBN 2950719007* (2013), pp. 34–37.
- [12] NestJS. *NestJS*. Visitado: 2024-07-24. July 2024. URL: <https://nestjs.com/>.
- [13] React. *React*. Visitado: 2024-03-24. Apr. 2024. URL: <https://react.dev/reference/react/apis>.

- [14] TS. Ciro Edgardo Romero. "Smart traceability, sistema de logística inteligente". Carrera de Especialización en Internet de las Cosas, Feb. 2021. URL: [http : / / laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-TrabajoFinal-CEIoT-Ciro-Edgardo-Romero-2021.pdf](http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-TrabajoFinal-CEIoT-Ciro-Edgardo-Romero-2021.pdf).
- [15] Vishay Semiconductors. *Ambient Light Sensor*. <https://www.vishay.com/docs/81579/temt6000.pdf>. Mar. 2021. (Visited on 03/02/2021).
- [16] GS1 Global Traceability Standard. *Business Process and System Requirements for Full Supply Chain Traceability*. [https : / / www . gs1 . org / sites / default / files / docs / traceability / Global Traceability Standard . pdf](https://www.gs1.org/sites/default/files/docs/traceability/GlobalTraceabilityStandard.pdf). Mar. 2021. (Visited on 03/01/2021).
- [17] TypeScript. *TypeScript*. Visitado: 2024-07-24. July 2024. URL: <https://www.typescriptlang.org/>.
- [18] U-Blox. *NEO-6 series. Versatile u-blox 6 GPS modules*. <https://www.u-blox.com/en/product/neo-6-series>. Mar. 2021. (Visited on 03/17/2021).
- [19] Xinyuan-LilyGO. *LilyGo T-Call SIM800 Series*. Visitado: 2021-03-10. Mar. 2021. URL: [https://github.com/ Xinyuan-LilyGO/LilyGo-T-Call-SIM800](https://github.com/Xinyuan-LilyGO/LilyGo-T-Call-SIM800).