DIGITAL TWIN DE BRAZOS ROBÓTICOS:
APLICACIÓN AL MANTENIMIENTO PREDICTIVO

DIGITAL TWIN OF ROBOTIC ARMS:
APPLICATION FOR PREDICTIVE MAINTENANCE

TRABAJO DE FIN DE MÁSTER EN INTERNET DE LAS COSAS
CURSO 2019-2020

AUTOR
RICHARD ALEXANDER CÓRDOVA HERRERA

DIRECTORES
LUIS PIÑUEL MORENO
SEGUNDO ESTEBAN SAN ROMAN

CONVOCATORIA: JULIO 2020 (ORDINARIA)
CALIFICACIÓN: 9 (SOBRESALIENTE)

MÁSTER EN INTERNET DE LAS COSAS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

15 DE JULIO DE 2020

DEDICATORIA

"*No importa el tiempo sino la huella que dejan*"

A mis padres, hermano y abuelita que siempre me dieron fuerzas para seguir adelante.
Especialmente a mi mascota Tito, que me acompaño muchos años, no te pude despedir, pero estas aquí.

AGRADECIMIENTOS

"*Trabaja duro en silencio y deja que tu éxito haga todo el ruido*"

A Dios, por siempre llenarme de bendiciones, salud y vida. Darme la sabiduría y fortaleza para tomar las decisiones y alcanzar las metas propuestas. Por darme la familia que tengo y mantenernos juntos a pesar de la distancia.

A mis padres, hermano, cuñada, sobrino y abuelita que estuvieron siempre pendientes de mi bienestar y dándome las fuerzas para seguir. A toda mi familia que con cada mensaje y llamada me alegran y empujan para conseguir el objetivo. Por el apoyo y las muestras de cariño de la familia que uno escoge, mis amigos, nunca faltaron. ¡Gracias!

Muchas gracias a Luis y Segundo que me ayudaron y aconsejaron en el desarrollo del trabajo. Gracias a todos los profesores del Máster que nos compartieron sus conocimientos.

¡¡Muchas gracias a todos!!

RESUMEN

DIGITAL TWIN DE BRAZOS ROBÓTICOS: APLICACIÓN AL MANTENIMIENTO PREDICTIVO

La evolución de la industria y la integración de robots a los procesos industriales ha marcado la cuarta revolución industrial denominada **Industry 4.0**. La integración de nuevas tecnologías a los procesos industriales como el *Internet of Things* (**IoT**) permite la interconexión de dispositivos, mejor monitoreo y análisis del estado actual de cada una de las partes del proceso, lo que da paso a la implementación de ***Digital Twins*** (DT). Un DT es una representación del estado actual y/o comportamiento de un dispositivo o sistema.

En los procesos industriales el mantenimiento de los equipos es una de las partes fundamentales, ya que una parada de producción no deseada representa grandes pérdidas de dinero, es por esto por lo que es de suma importancia saber cuándo un equipo está funcionando mal para realizar un mantenimiento. Esto se consigue con el **mantenimiento predictivo**, que permite tomar decisiones del proceso gracias a los datos generados del DT y aplicando técnicas de **Inteligencia Artificial**.

En el presente proyecto se realiza un DT con el fin de realizar mantenimiento predictivo para evitar paradas de planta no deseadas y pérdidas de equipos. La adquisición de datos se lo realiza por **BLE** desde un nodo sensor genérico y se envía mediante **MQTT** a un servidor, donde se almacena todos los datos en **MongoDB**. Para la visualización de la información y del estado actual del sistema se desarrolló un *dashboard* en **Ignition,** un software SCADA, que se ocupa a nivel industrial.

Palabras clave

Digital Twin, Mantenimiento Predictivo, IIoT, Industry 4.0, MQTT, BLE, Inteligencia Artificial, MongoDB, Ignition

ABSTRACT

DIGITAL TWIN OF ROBOTIC ARMS: APPLICATION TO PREDICTIVE MAINTENANCE

The evolution of the industry and the integration of robots into industrial processes has marked the fourth industrial revolution called **Industry 4.0**. The integration of new technologies to industrial processes, such as the Internet of Things (**IoT**), allows the interconnection of devices, better monitoring, and analysis of the current status of each of the parts of the process. This leads to the implementation of **Digital Twins** (DT), which is a representation of the current state and/or behavior of a device or system.

In industrial processes, equipment maintenance is one of the fundamental parts because an unwanted stop production represents large losses of money. This is the reason which why it is very important to know when equipment is malfunctioning to carry out a maintenance. This is achieved with **predictive maintenance**, which allows to make decisions made in the process thanks to the data generated from the DT and the application of **Artificial Intelligence** techniques.

In this project, a DT is carried out in order to perform predictive maintenance for avoiding unwanted plant shutdowns and equipment losses. Data acquisition is done by **BLE** from a generic sensor node and sent via **MQTT** to a server, where all the data is stored in **MongoDB**. To display the information and the current state of the system, a dashboard was developed in **Ignition**, a SCADA software, which is used at the industrial level.

Keywords

Digital Twin, Predictive Maintenance, IIoT, Industry 4.0, MQTT, BLE, Artificial Intelligence, MongoDB, Ignition.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1 - Introducción

1.1 Antecedentes

Los robots han sido una parte fundamental en el crecimiento de procesos de manufactura. Existen dos tipos de robots en la industria: robots industriales y los robots de servicio. Los robots industriales son los encargados de los procesos de producción, mientras que los robots de servicio son creados para aplicaciones más específicas, para solucionar problemas y realizar tareas de los seres humanos, excluyéndolos del sector industrial [1].

Los robots industriales han sido utilizados durante varias décadas en los procesos de manufactura realizando actividades que son repetitivas y secuenciales, como la soldadura, la clasificación, movimiento de piezas, etc. Con su inclusión en el sector industrial se ha conseguido aumentar la calidad de los productos y disminuir el tiempo de producción. Por otro lado, al ser equipos programables, existe flexibilidad al momento de delegar las tareas en los distintos puntos del proceso [2].

Cada año el uso de los robots en la industria tiene un crecimiento importante. Según la Federación Internacional de Robótica (IFR) el año 2018 tuvo un récord en ventas de 16.5 mil millones de dólares, registrando un crecimiento del 6% en las unidades existentes a nivel mundial en comparación del 2017. En el año 2019 se espera una disminución en este crecimiento, pero para los años 2020, 2021 y 2022 se estima un crecimiento promedio de 12% por año como se muestra en la Figura 1-1. Por otro lado, el sector que más unidades posee es el automotriz, seguido con un avance muy importante de las empresas fabricantes de dispositivos y piezas eléctricas/electrónicas [3].

Por lo general una línea de producción está conformada por varios robots industriales, que tienen que trabajar en conjunto para producir el producto final. Es decir, en el caso de que uno de ellos falle o tenga una avería implica que se va a parar por completo la línea, a menos de que cada robot tenga su respaldo lo que implica mayor inversión. Esto ocurre en varios de los sectores de la industria por lo que es muy importante llevar a cabo un plan de mantenimiento para evitar pérdidas en la producción y dinero.

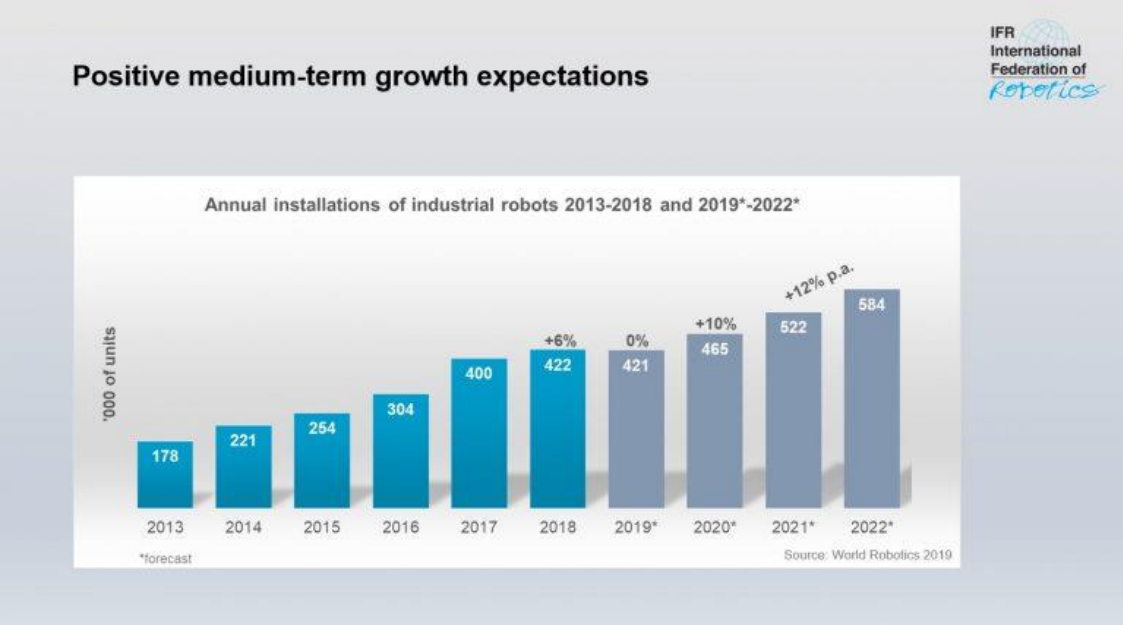


Figura 1-1. Instalaciones Anuales de Robots Industriales 2013-2018 y predicción para 2019-2020 [3].

Existen tres tipos de mantenimiento que se pueden realizar a los equipos: mantenimiento correctivo, mantenimiento preventivo y mantenimiento predictivo. El primero de ellos es el más crítico debido a que se lo realiza cuando el robot ya sufrió un daño y se detuvo para repararlo. El mantenimiento preventivo se lo realiza por lo general siguiendo las recomendaciones del fabricante del equipo y es periódico. Por último, el mantenimiento predictivo, se basa en el comportamiento del equipo, es decir se realiza un diagnóstico en tiempo real y consultando históricos del estado del equipo. Con esto se pueden detectar posibles fallos y establecer un plan de trabajo, disminuyendo al máximo la pérdida por mantenimiento de un equipo.

Los brazos robóticos tienen la capacidad de realizar auto chequeos antes de entrar en funcionamiento, es decir al momento de encender el sistema realiza una secuencia de verificación para detectar posibles fallos que puedan existir en el sistema. La implementación de un agente externo para tomar muestras y para almacenar el comportamiento del robot en todo momento, permite realizar un análisis de los posibles fallos que no han sido detectados por el sistema propio del robot. Estos sensores se los puede implementar en toda la línea de producción, facilitando el monitoreo del estado de los robots en todo momento.

La tecnología del Internet de las cosas (IoT) tiene varios campos de aplicación como: Smart Cities, Medicina, manejo de recursos, logística, industria, etc. La industria es uno de los campos

más relevantes por su importancia a nivel global. El IoT es un grupo de dispositivos interconectados que permite su manejo, acceder a la información y estado actual. Por otro lado, los nodos sensores generalmente pueden ser móviles por lo que se puede compartir un solo nodo sensor entre varios brazos robóticos. De la misma forma, un nodo sensor tiene la capacidad de realizar varias mediciones a la vez por lo que se puede adaptar al tipo de robot con el que esté trabajando. Es decir, con un mismo sensor se puede realizar las mediciones de una línea de producción de manera simultánea y distribuida sin mucha inversión.

1.2 Motivación

Como se explica en el punto anterior, estamos en la era de la digitalización industrial. La industria es uno de los principales sectores para la economía mundial y debe actualizarse empleando equipos de última tecnología para lograr alcanzar la optimización de los procesos, disminuyendo costos y aumentando la productividad. Por lo general en las plantas de producción se puede encontrar gran cantidad de datos, provenientes de sensores y actuadores, que actúan en el proceso de producción.

Las herramientas que el IoT ofrece para adquirir los datos de sensores y actuadores, interconectar los dispositivos, acceder a la información desde cualquier lugar, es de gran ayuda para tomar decisiones en el proceso, integrando un valor agregado a la empresa. Por otro lado, con el uso de inteligencia artificial se puede realizar un estudio de los datos, para predecir eventos, posibles daños y/o detectar comportamientos anómalos, disminuyendo costos de producción y tiempos muertos que existen en la industria.

Uno de los grandes problemas de la industria es el momento en que un equipo está fuera de servicio, esto implica pérdida de producción que es gran cantidad de dinero. La industria invierte mucho de dinero en planes de mantenimiento, para alargar el tiempo de vida útil de un equipo y/o evitar que la máquina se estropee. Estos mantenimientos son realizados en base a recomendaciones del fabricante o experiencia de los operadores, por esto es relevante la idea de integrar algoritmos de mantenimiento predictivo disminuyendo la inversión en repuestos que no se ocupan y paradas obligadas.

Digital Twin (DT), es una herramienta muy potente ya que integrando IoT para adquirir los datos y comunicar los dispositivos, junto con la inteligencia artificial para realizar el procesamiento de la información se puede realizar predicciones y tomar decisiones trascendentes en el proceso. Esta es la razón por la que se realizará un estudio de las últimas tendencias que tienen estos paradigmas e implementar una prueba de concepto en un brazo robótico para demostrar el potencial que puede generar a nivel industrial.

1.3 Objetivos

El objetivo principal del presente trabajo es desarrollar una infraestructura que permita mantener un Digital Twin (DT) del comportamiento de un brazo robótico, almacenando información de sensores para analizar el historial de sus mediciones con el fin de detectar comportamientos anómalos en los movimientos. Gracias a esta detección y procesamiento de los datos se puede realizar mantenimiento predictivo.

1.3.1 General

Desarrollar una infraestructura para la representación del *Digital Twin* (DT) de un brazo robótico, que permite monitorizar su comportamiento.

1.3.2 Específicos

- Monitorizar el comportamiento del brazo robótico, para realizar una detección de anomalías en su funcionamiento.
- Determinar las magnitudes necesarias que deben ser monitorizadas para realizar un correcto análisis de las posibles fallas del brazo robótico.
- Analizar la posibilidad de emplear un nodo sensor genérico para realizar la adquisición de datos de brazos robóticos.
- Desarrollar un sistema de análisis de mantenimiento predictivo, con los datos obtenidos a partir del nodo sensor.
- Almacenar la información de los sensores para tener un histórico de los datos y visualizarlos en un dashboard creado en un software SCADA industrial.

1.4 Plan de trabajo

La primera etapa del trabajo se centra en la recopilación de información relevante disponible, estudiando las herramientas y tecnologías que se han empleado para realizar proyectos similares, es decir un estudio del estado del arte. Se establecerá un conjunto de conceptos generales que se deben entender para entrar al desarrollo de la prueba de concepto. Este punto es importante ya que se define el enfoque de la investigación y de la implementación del proyecto.

La segunda etapa se basa en la definición y desarrollo de la infraestructura IoT, en esta fase se describen a detalle todas las tecnologías que se van a emplear para realizar la solución. Se describe el nodo sensor que se emplea, el tipo de comunicaciones con los que se va a trabajar y las herramientas de software que se utilizarán para alcanzar los objetivos planteados. Una vez definida la infraestructura, se realizará la aplicación con el brazo robótico, teniendo que definir las medidas que se van a tomar con los sensores disponibles. En otras palabras, se definirá la información que se va a tomar del brazo robótico para su posterior análisis y pruebas.

Por último, en la tercera etapa del proyecto se definen las pruebas que se van a realizar para evaluar el funcionamiento de la solución creada, básicamente son las pruebas de concepto seleccionadas para demostrar el uso y las capacidades del DT. En este punto se debe ejecutar el algoritmo de detección de anomalías establecido para evaluar su correcto funcionamiento y verificar los históricos que se podrán ver desde un dashboard.

Capítulo 2 - Estado del Arte

En el capítulo del estado del arte se realiza una recopilación de información relevante para desarrollo del proyecto, para dar un correcto enfoque al trabajo. Desde los conceptos generales de robots industriales, y el análisis de las distintas etapas de la industria que se han dado desde la primera revolución industrial. De la misma forma se revisan los casos de estudio que se han desarrollado en el tema del Digital Twin, las pruebas de concepto, casos de éxito que se han registrado. Por otro lado, se analizan las tecnologías disponibles que hacen posible la digitalización industrial y seleccionar las adecuadas para el proyecto.

2.1 Robots Industriales

Los robots industriales son una máquina de sistema mecánico que es programable y reprogramable, creados con el fin de realizar tareas repetitivas o riesgosas para una persona. Por otro lado, también se los conoce como robots manipuladores, ya que por lo general tienen forma de un brazo humano y son capaces de atrapar y/o mover objetos. Es muy importante entender que un robot industrial es reprogramable, lo que quiere decir que no se necesitan cambios mecánicos, para que se realice una tarea distinta [9].

Los robots industriales se basan en tres leyes fundamentales, que fueron definidas por el escritor ruso Issak Asimov en 1942. Posteriormente en 1985 se definió una ley denominada “ley cero”. Por otro lado, Los robots industriales complementan estas 4 leyes con dos que fueron añadidas por Stig Moberg de ABB Robotics; dando como resultado las siguientes leyes:

0. Un robot no puede dañar a la humanidad o, por inacción, permitir que la humanidad sufra daños [9][10].
1. Un robot no puede dañar a un ser humano o, por inacción, permitir que un ser humano sea herido [9][10].
2. Un robot debe obedecer las órdenes que le dan los seres humanos, excepto donde estas órdenes entren en conflicto con la Primera Ley [9][10].
3. Un robot debe proteger su propia existencia siempre que dicha protección no entre en conflicto con la primera o segunda ley [9][10].

4. Un robot debe seguir la trayectoria especificada por su maestro, siempre que no entre en conflicto con las primeras tres leyes [9].
5. Un robot debe seguir la velocidad y la aceleración especificadas por su maestro, siempre que nada se interpone en su camino y no entre en conflicto con las leyes anteriores [9].

Los robots se van sofisticando con el pasar de los años y el desarrollo de la tecnología, con la era de la digitalización los robots industriales deben integrar algoritmos de inteligencia para aportar a la Industria 4.0. En las empresas de manufactura se ha incrementado la necesidad de emplear robots colaborativos, es decir que exista interacción entre el robot y el humano, siendo de mucha importancia el estudio de su comportamiento para saber las reacciones que puede tener. De aquí proviene el término de robótica colaborativa, que su principal característica es la descrita anteriormente [37].

Es de suma importancia que los robots y los humanos trabajen en conjunta ya que se obtiene una optimización en el proceso y un mejoramiento en la calidad de los productos. Los robots colaborativos deben tener tres características principales [37]:

- **Accesibilidad:** capacidad de reprogramación, tener un nivel de integración alto, para que cualquier operador pueda interactuar con el robot y asignarle una nueva tarea.
- **Seguridad:** la necesidad de eliminar las fronteras entre el espacio de trabajo del robot industrial y los humanos. Es decir, ya no existen las cercas o cajas donde trabajaban los robots en la línea de producción.
- **Flexibilidad:** se pueden incorporar a espacios pequeños y su movilización es sencilla.

2.2 *Industria 4.0*

Actualmente estamos viviendo la cuarta revolución industrial, denominada Industria 4.0. Este término proviene de un programa del gobierno alemán en el 2011, que insertó el concepto "Industrial 4.0", como una estrategia de alta tecnología para el 2020 [4]. Después de dos años en el 2013, fue anunciado oficialmente, como estrategia principal para el desarrollo de la industria manufacturera, agregando nuevas tecnologías para obtener un mejor desempeño [5]. La Industria 4.0 integra nuevos conceptos en los procesos de producción, como el *Internet of Things* (IoT), la digitalización, *Cyber-Physical Systems* (CPS), *Cloud Computing (CC)*, *Big data*, etc [6].

El desarrollo de la industria se divide en cuatro fases que se pueden apreciar en la Figura 2-1, en la primera está la Industria 1.0, que es la primera revolución industrial llevándose a cabo a finales del siglo XVII y comienzo de XVIII, que se refiere a la integración de sistemas mecánicos aprovechando la energía producida por el agua y el vapor, siendo el sector de la agricultura el más importante para esta época. La industria 2.0, añadió el concepto de producción en masa, es decir se empezó a producir en grandes cantidades, introduciendo la energía eléctrica [5][6]. Por otro lado, la fase que más relación tiene con la Industria 4.0 se produce a mediados del siglo XX; la tercera revolución industrial se da por la era de la información, caracterizado por las tecnologías de comunicación empleadas, integración de control numérico por computador (CNC), robots industriales, diseño asistido por computador (CAD) y manufactura asistida por computador (CAM) [5][6].

La industria 4.0 que también significa *Smart Factory* [33], tiene un efecto muy importante en la economía global, debido a que con la integración de las diferentes tecnologías se obtiene un mejor rendimiento y mejora de la calidad del producto. Gracias a la industria inteligente se tiene mayor flexibilidad en la producción, dependiendo de la necesidad y la demanda del producto, gracias a la rápida configuración que se obtiene con los robots autónomos. De la misma forma, se logra una mejora importante en la velocidad de producción, la digitalización y modelado virtual, evita contratiempos al momento de implementar un sistema [8].

Empleando algoritmos de inteligencia artificial y análisis de programas de mantenimiento predictivo se logra mejorar la productividad. La industria puede disminuir los tiempos de parada, evitando daños prematuros de los equipos y aumentar la producción. De la misma forma se evitan los mantenimientos preventivos, que en muchos de los casos se los realiza por sugerencia del fabricante para proteger los equipos, pero no se basan en los datos y las condiciones de operación reales de los equipos.

2.2.1 *Cyber-Physical Systems (CPS)*

Un sistema CPS es la integración del mundo físico con el mundo virtual, integrado por un sistema de comunicaciones, sensorización y control. La industria 4.0 integra los CPS para obtener un sistema de producción ciber-físico (CPPS), para mejorar el desempeño y manejo de los procesos de producción, manejando la información proveniente de estos [6]. Los CPS son unos de los pilares

fundamentales de la Industria 4.0 para lograr una flexibilidad y adaptación con los sistemas. El uso combinado de dispositivos físicos y virtuales para alcanzar un objetivo, hacen de los CPS un punto importante en la revolución industrial [34].

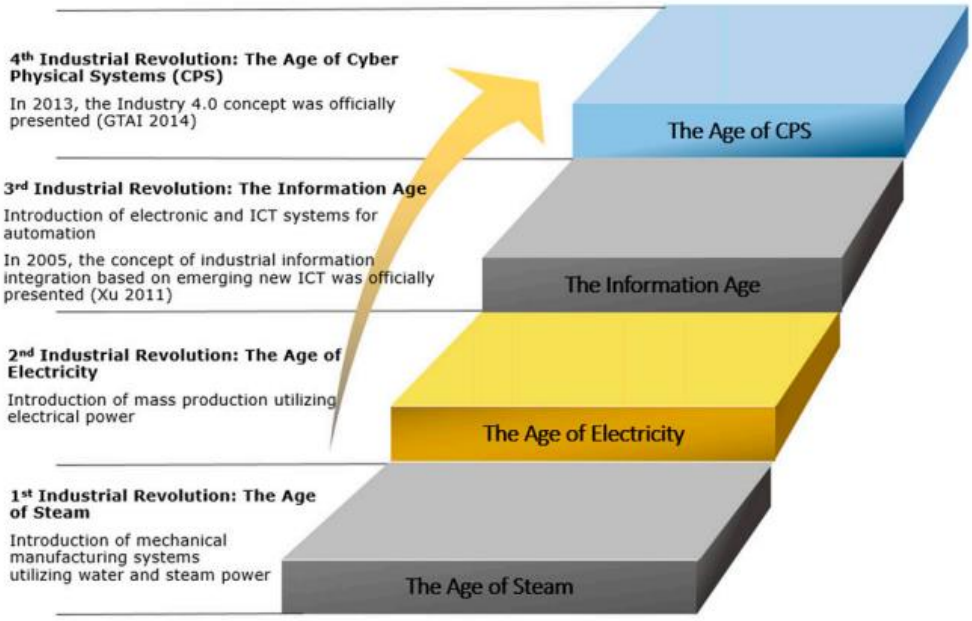


Figura 2-1. Historia de la Evolución de la Industria, desde la primera revolución hasta la Industria 4.0 [5]

2.2.2 *Cloud Computing (CC)*

Es uno de los pilares de la Industria 4.0, ya que integra los servicios de internet al sistema. Dependiendo de las necesidades del proceso, el CC nos permite trabajar con plataformas virtualizadas e infraestructuras que se construyen a medida. Es decir, el sistema es capaz de almacenar y procesar información, disminuyendo costos de inversión en infraestructura y aumentando el rendimiento del sistema. [4][7]. Gracias al CC el sistema es capaz de procesar la información y manejar los datos generados para transformarlos en información, dotando al sistema de un alto rendimiento [15].

Actualmente la industria tiene la necesidad de tomar decisiones en base a los datos, este problema es solucionado con el CC, ya que se logra almacenar gran cantidad de información y procesarla en poco tiempo, manejar esta información involucra una combinación entre el proceso de producción y negocio dando un valor agregado a la empresa [35]. Almacenar esta cantidad de

información y procesarla en servidores locales, significa un gran uso de recursos e infraestructura, por lo que el CC es la mejor solución [5][15].

2.2.3 *Big Data and Analytics*

En la industria 4.0 la digitalización cumple un papel muy importante, por lo que es primordial instalar varios sensores para recolectar información de la producción. Al tener un entorno con gran cantidad de información por procesar y analizar, es de mucha importancia trabajar los datos para convertirlos en información, para tomar decisiones en el proceso. Millones de datos en tiempo real son generados por la cantidad de dispositivos interconectados para monitorizar el estado del equipo y/o producción, almacenándolos en una base de datos local o externa que se encuentra en la nube. Para procesar esta cantidad de información es necesario emplear técnicas de ciencias de datos y análisis de datos, con ello lograr disminuir tiempos de producción, reducir costos, optimizar procesos y mejorar la eficiencia de la planta [5][8].

2.2.4 Robots Autónomos

Los Robots autónomos son una pieza clave en la Industria 4.0. Por lo general son empleados para realizar tareas repetitivas y/o que necesiten mucha precisión, también se integra un alto nivel de inteligencia artificial para que puedan tomar decisiones sin la necesidad de supervisión. Esto es lo que les diferencia de la anterior generación de robots [33]. Por otro lado, los robots son programados para trabajar en conjunto, es decir en una línea de producción existen robots a los que se les asignan diferentes tareas para terminar un procedimiento, estos robots deben ser capaces de comunicarse entre sí para lograr una sincronía en el proceso de producción. Deben ser flexibles, es decir adaptarse a las necesidades de la empresa, en el caso de que se necesite asignarle otra tarea, es sumamente sencillo reprogramarlos [6].

2.2.5 *Internet of Things (IoT)*

Existen varias definiciones propuestas por diferentes autores entorno al IoT. Cada una de las definiciones tienen un enfoque, pero términos en común que describen lo que en realidad es el IoT. Se puede decir que el IoT es la interconexión de dispositivos, sensores y actuadores que comparten información, mediante el uso de tecnologías de comunicación. Una característica fundamental es

la interoperabilidad, ya que los dispositivos deben ser capaces de hablar un mismo idioma, para facilitar el intercambio de información [22][23].

De la misma forma que aún no existe una definición oficial de IoT, existen diferentes tipos de arquitecturas que maneja el IoT. Existen autores que definen al IoT como un modelo desde cinco hasta uno de nueve capas [36]. En base a esto, como se muestra en la Figura 2-2, el autor de [36] propone un modelo de cuatro capas para el IoT aplicado a la industria que son: percepción, transmisión, computación y aplicación.

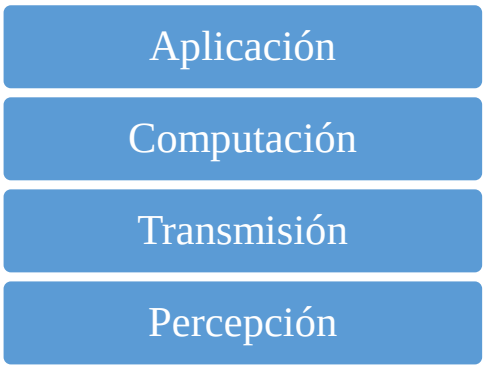


Figura 2-2. Capas IoT para la Industria 4.0 [36]

- **Capa de percepción**: abarca el nivel de campo, la adquisición de datos desde sensores, controladores, actuadores entre otros dispositivos, es decir la capa donde se puede percibir la información del objeto físico.
- **Capa Transmisión**: una vez recolectados los datos provenientes de campo, de alguna forma se los debe enviar a las capas más altas para su procesamiento. Este proceso se lo realiza tomando en cuanta varias características fundamentales, tales como: consumo ancho de banda, consumo energía, almacenamiento, etc.
- **Capa Computación**: en esta capa es donde se transforman los datos a información, es decir se da un sentido a los valores medidos. En esta etapa se integran tecnologías y herramientas capaces de procesar los datos, hardware, software, seguridad, etc.
- **Capa Aplicación:** en esta capa se consume la información obtenida en el procesamiento de los datos, es una etapa a nivel de negocio, donde se toma decisiones para el proceso o la línea de negocio.

2.2.5.1 Nodos Sensores

Son los encargados de recibir los datos del medio físico y enviarlo al gateway. Están conformado por un sensor, transductor o un actuador, una fuente de alimentación, un *System on Chip* (SoC) y memoria externa. Por lo general se diseña la infraestructura para evitar que el nodo sensor consuma mucha energía, por lo que existen varios dispositivos que se apagan para evitar el consumo de las baterías y se encienden cuando sea necesario. Por otro lado, existen nodos sensores que realizan un preprocesamiento para enviar la información, son nodos que tienen más capacidad [29].

2.2.5.2 Comunicaciones

Las comunicaciones en el IoT por lo general son inalámbricas, donde las tecnologías de comunicación juegan un papel fundamental porque se debe garantizar que la comunicación sea fiable, las principales se muestran en la Tabla 2-1. Por otro lado, como se revisó en la definición de IoT la interoperabilidad y estándares abiertos son fundamentales para lograr una conexión. Dependiendo de la necesidad de la solución se pueden realizar cuatro tipos de conexiones básicas: dispositivo a dispositivo, dispositivo a la nube, dispositivo a la nube y intercambio de datos con *backend*, este último modelo de conexión es muy útil cuando se necesita comparar la información obtenida por el sensor y compararla con información de sensores externos. Se basa en el modelo de dispositivo a la nube, ya que envía la información a la nube y se la puede comparar con los sensores externos o de terceros [29].

2.2.5.3 Aplicaciones

Existe un sin número de aplicaciones en las que se puede emplear el IoT, entre las principales y más desarrolladas están: casas inteligentes, edificios inteligentes, en la medicina, transporte y logística, redes inteligentes, industria inteligente, etc [28]. En la Industria 4.0 es de mucha utilidad ya que integra la posibilidad de obtener los datos en tiempo real de los diferentes procesos de la empresa. Se integra la capacidad de almacenar los datos tomados de los sensores, para su posterior análisis y tomar decisiones del proceso [7].

Tabla 2-1. Tecnologías Comunicación disponibles IoT [28].

Tecnología	Velocidad	Rango	Consumo energético	Aplicaciones
Bluetooth	700 kb/s	30 m	Bajo	Smart Home, intercambio datos
Bluetooth LE	25 Mb/s	10 m	Bajo	Smart Home, salud, entrenamiento
ZigBee	250 kb/s	1000 m	Bajo	Smart homes, Smart health, redes sensors, edificios, automatización industrial
Wifi	54 Mb/s, 6.75 Gb/s	140 m, 100 m	Medio	Smart home, Smart cities, multimedia, video.
NFC	400 kbps	10 cm	Bajo	Aplicaciones de pago, intercambio de datos.
RFID	400 kbps	3 m	Bajo	Accesos, seguridad
6LoWPAN	250 kbps	800 m	Medio	Redes sensores, edificios, automatización industrial

2.3 *Digital Twin (DT)*

Como se pudo apreciar con el desarrollo de la Industria 4.0 nos encontramos en la era digital, junto con las nuevas tecnologías de comunicación y las técnicas existentes es posible desarrollar varias soluciones a nivel industrial. El término *Digital Twin* (DT), se dio a conocer en un proyecto en la industria aeroespacial, que pretendía realizar varios experimentos para realizar mantenimiento y para predecir la vida útil de vehículos aeroespaciales [11]. Existen varias definiciones de DT que aparecieron a partir del año 2010 aproximadamente, Negri (2017) lo define como [12][13]:

"Un Gemelo digital consiste en una representación virtual de un sistema de producción que es capaz de correr en diferentes disciplinas de simulación que es caracterizado por la sincronización entre el sistema real y virtual, gracias a los datos sensorizados y los dispositivos inteligentes conectados, modelos matemáticos y elaboración de datos en tiempo real"

Por otro lado, un DT está conformado por tres partes principales: el dispositivo físico, su representación virtual y datos sensorizados que son recolectados para comunicar el mundo físico con el virtual [14][15]. Es decir, un DT, es una representación virtual, de un dispositivo físico, que se lo emplea para optimizar los tiempos de desarrollo y pruebas de los equipos, siendo capaces de realizar un análisis del comportamiento, predicciones de posibles fallas y optimización del proceso de producción [16][17].

La necesidad de interconectar los dispositivos, sensores y actuadores existentes en una línea de producción hace que el *Industrial Internet of Things* (IIoT) sea la solución ideal para la conectividad. El IIoT es una clasificación del IoT que se diferencia principalmente por la infraestructura empleada, por los protocolos de comunicación y los equipos que intervienen. La diferencia entre los dispositivos que intervienen el IoT y el IIoT básicamente es que son más robustos y seguros para resistir un entorno industrial. Gracias al IIoT, la información proveniente de los distintos sensores se la envía para poder procesarla. Cada gemelo digital debe almacenar su información, este almacenamiento se lo puede realizar de manera local y externa específicamente en la nube, para poder acceder a ella desde cualquier lugar [19][20].

Existen varias arquitecturas que pretenden definir un flujo de trabajo con los DT, en el trabajo desarrollado en [25] se muestra un resumen de las partes fundamentales que debe tener la arquitectura de un DT de un proceso de producción que se muestra en la Figura 2-3. Donde se muestra la parte física que consta de sensores y actuadores, sistema de comunicaciones y el empleo del IoT. Por su parte el DT es una mezcla de las herramientas expuestas de la industria 4.0 el *Big Data*, inteligencia artificial y el *Cloud Computing*, para transformar esos datos tomados del modelo físico y transformarlos en información útil para tomar decisiones. Existen varias aplicaciones de los DT, en las que se va a enfocar el presente proyecto es realizar predicciones, para la toma de decisiones y realizar mantenimiento predictivo [25].

Por otro lado, en el trabajo de [26] se muestra un caso de estudio de la metodología empleada para crear un DT. Se basa en la integración de cuatro partes:

- Modelo físico: que es el encargado de generar los datos, en el caso del presente proyecto se los va a obtener con un nodo sensor.
- Creación del modelo virtual del sistema. Es decir, con los datos obtenidos de los sensores crear un modelo virtual para su representación.
- Herramienta que permita acceder a los datos desde un agente externo.
- Los sistemas externos que van a consumir la información almacenada del DT, el usuario final que va a visualizar la información generada.

En la prueba de concepto realizada en [24] se crea el DT de un banco de pruebas de flexión en la que mediante dos celdas de carga realizan la medición de la fuerza aplicada resultante. En

esta prueba se ocupa MQTT como protocolo de comunicación, la principal razón en la escalabilidad que tiene, debido a que fácilmente se puede integrar un nuevo cliente al sistema de publicación/suscripción. En la arquitectura del sistema uno de los clientes MQTT es una plataforma IoT que sirve para controlar el modelo físico y el DT que es la vista de todas las mediciones resultantes en un *dashboard*.

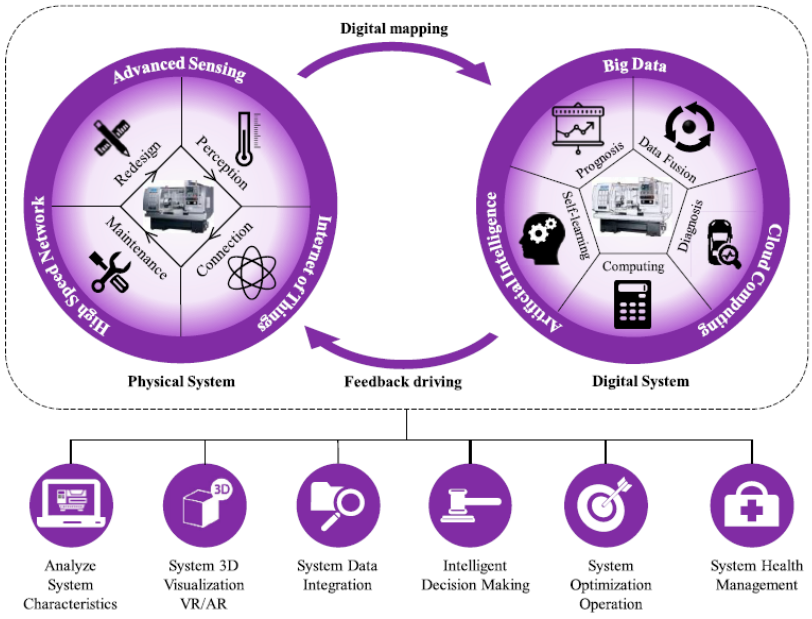


Figura 2-3. Arquitectura de DT [25]

2.3.1 Aplicaciones del *Digital Twin*

El sector que más acogida ha tenido a los DT es el industrial, por la cantidad de equipos que intervienen en una línea de producción y la necesidad de monitorear en tiempo real los datos provenientes del proceso. La necesidad conocer el estado de la producción ha hecho que los DT se desarrollen rápidamente. Se debe adquirir gran cantidad de datos y procesarlos con técnicas avanzadas de *Big Data*, para tener en cualquier momento acceso a los datos y poder tomar decisiones.

Los gemelos digitales juegan un papel primordial en el desarrollo de la Industria 4.0, por su capacidad de procesamiento de datos y modelado del sistema para realizar pruebas. Se realizan simulaciones para exponer al sistema a condiciones de trabajo reales y saber cuál es su comportamiento, para tomar decisiones en la optimización del comportamiento. Por otro lado, se

pueden disminuir tiempo de producción, detectando tiempo muertos y/o alternativas a la forma de realizar las actividades.

Con respecto al mantenimiento de los equipos los DT juegan un papel fundamental, debido a que tienen los datos reales del dispositivo. Se puede disminuir los costos invertidos en mantenimiento preventivo o programado que por lo general se lo realiza por recomendaciones del fabricante. Por otro lado, se evitan las paradas obligadas de las líneas de producción por un fallo. Para evitar interrumpir el proceso continuo de producción, con los datos provenientes del equipo se puede realizar un análisis para realizar mantenimiento predictivo, obteniendo la información necesaria para determinar la vida útil de cada uno de los componentes del sistema [12][14][16][18].

En [16] se muestra un ejemplo de aplicación de DT, donde se logró disminuir el tiempo de producción de un producto en un 5,2%. El proyecto se basó en tomar los datos de tiempos de producción de cada estación de la línea de producción. La información proveniente del entorno físico fue procesada por un algoritmo virtual y como resultado se determinó la función para alcanzar el mejor rendimiento del proceso.

Existen estudios realizados en entornos en los cuales los humanos interactúan con los robots para realizar una tarea en específico. En el trabajo de [27] se emplean un DT para generar un entorno virtual y estudiar las reacciones que se van a producir cuando el humano interactúa con el robot ante movimientos predecibles y movimientos no predecibles generados por el brazo robótico, estos últimos suelen ser muy peligrosos y comunes en estos entornos. Gracias al DT se pueden recopilar la información y estudiar las reacciones que tienen los humanos.

Muchas de las aplicaciones de los DT se basan en las predicciones que se logran realizar a partir de la información recolectada, específicamente mantenimiento predictivo lo que incrementa la productividad y eficiencia de la planta, por otro lado, maximiza el ciclo de vida de los componentes y reduce el costo de mantenimiento [30]. En el trabajo realizado en [32] se muestra el modelo de un DT basado en simulación, de un sistema de frenos de un automóvil, para monitoreo de calor y mantenimiento predictivo. Los modelos basados en DT son de mucha ayuda para agregar situaciones anormales al sistema y estudiar el comportamiento del modelo. Por otro lado, se pueden visualizar los fallos de las lecturas provenientes de sensores que en modelo físico no se las logra apreciar para llevar un históricos de la información y tomar decisiones.

Capítulo 3 - Infraestructura IoT

En el presente capítulo se va a realizar un estudio de las tecnologías que se han empleado para levantar la infraestructura IoT. La infraestructura se centra en la adquisición de datos por medio de un nodo sensor, que mediante protocolos de comunicación son enviados al servidor para su almacenamiento y posterior análisis. Es por esta razón que se va a realizar un análisis de cada uno de los componentes del sistema: nodo sensor, protocolos de comunicación, software y hardware empleado.

3.1 Nodo Sensor

Un nodo sensor no es más que un nodo de una red de sensores. Un nodo es un dispositivo generalmente pequeño que está conformado por cinco partes que se muestran en la Figura 3-1: sensor encargado de adquirir los datos del entorno en el que está funcionando, la unidad Central por lo general es un microprocesador que realiza el manejo de las tareas, módulo de comunicaciones, memoria y batería. Es de suma importancia el control del manejo del consumo de energía, ya que la mayoría de los sensores funcionan con baterías tradicionales, por lo que se debe integrar la capacidad de apagar el sensor en entornos que no sea necesario realizar las mediciones continuas. [38] [40]

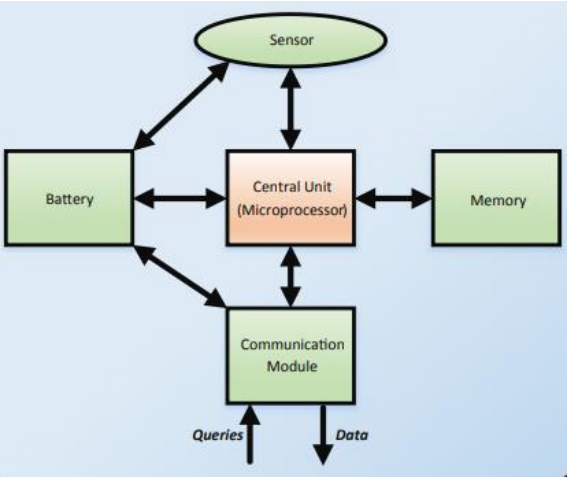


Figura 3-1. Arquitectura de un Nodo Sensor [40]

3.1.1 *SensorTag CC2650*

El SensorTag es kit de evaluación desarrollado por la empresa *Texas Instruments* destinado específicamente para soluciones en sistemas IoT. Integra comunicación inalámbrica *Bluetooth Low Energy* (BLE) para enviar y recibir información. Es un nodo sensor que tiene un bajo consumo de energía garantizando años de funcionamiento con un solo recambio. Entre las características mas importantes estan: [38] [39]

- Integración hasta 10 sensores de bajo consumo.
- Consumo de energía sumamente bajo, años de duración con una sola batería.
- Es un nodo reprogrammable y permite la integración de nuevos sensores y/o actuadores.
- Múltiples estándares de red: BLE, 6LowPAN, Zigbee.
- Acceso seguro a la nube.

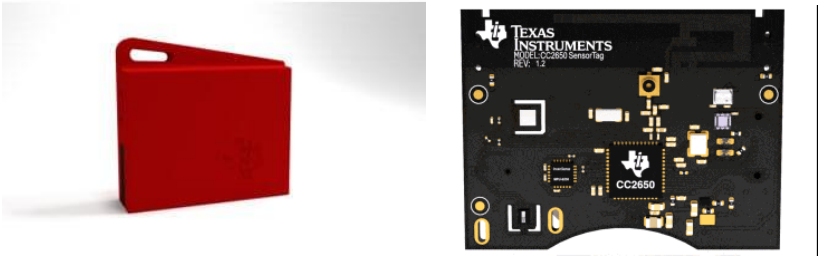


Figura 3-2. CC2650 SensorTag. a) Parte posterior con estuche. b) Placa PCB [39]

Para el desarrollo de la solución se ha seleccionado el *SensorTag*, ya que con el mismo nodo se pueden tomar varias medidas. En este caso se va a trabajar con el acelerómetro, sensor de temperatura ambiente y del objeto. Sim embargo este dispositivo tiene ya integrados los siguientes sensores: temperatura, movimiento, humedad, presión y luminosidad. [41] En el caso de que se quiera integrar un nuevo sensor, se lo puede realizar conectándolo al *SensorTag* de ser compatible o integrándolo a la infraestructura como un nuevo nodo sensor.

3.1.1.1 Sensor Movimiento

El sensor integrado en el dispositivo es en MPU9250 desarrollado por *InvenSense*. Es un dispositivo que realiza 9 mediciones: acelerómetro, giroscopio y magnetómetro en sus tres ejes cada uno. Este sensor es muy utilizado en teléfonos inteligentes, tablets, etc. Su resolución es de

10 ms, pero el rango de trabajo del programa que viene cargado es de 100ms a 2.55 s. Por otro lado, en la configuración de este sensor se puede habilitar la opción de *Wake-On-Motion*, que permite desactivar el sensor si no se detecta movimiento por el tiempo establecido. [41]

- **Configuración:** en la configuración del dispositivo se habilitan los sensores que se van a emplear y se defines las características necesarias. Es un registro de 16 bits que se detalla en la Tabla 2-1.

Tabla 3-1. Configuración Sensor Movimiento [41]

Bits	Uso
0	Habilitar Giroscopio Eje Z
1	Habilitar Giroscopio Eje Y
2	Habilitar Giroscopio Eje X
3	Habilitar Acelerómetro Eje X
4	Habilitar Acelerómetro Eje Y
5	Habilitar Acelerómetro Eje Z
6	Habilitar Magnetómetro (Todos los ejes)
7	Habilitar Wake-On-Motion
8:9	Rango Acelerómtro (0=2G, 1=4G, 2=8G, 3=16G)
10:15	No usado

- **Datos:** el valor para cada una de las mediciones consiste en valores de 16 bits sin signo en el siguiente orden: giroscopio, acelerómetro y magnetómetro. Otro de los parámetros importantes es el valor de la gravedad: 2G, 4G, 8G y 16G; el valor por defecto es de 2G que es con el que se va a trabajar. En la Ec. 3-1, Ec. 3-2 y Ec. 3-3 y se describen las fórmulas para obtener el valor de la rotación, aceleración y magnetismo en deg/s, G y uT respectivamente.

$$Rotación\ (x,y,z) = \frac{(\ data(x,y,z) * 1.0\ )}{\frac{65536}{500}} \qquad \text{Ec. 3-1 [41]}$$

$$Aceleración\ (x,y,z) = \frac{(\ data(x,y,z) * 1.0\ )}{\frac{32768}{2}} \qquad \text{Ec. 3-2 [41]}$$

$$Magnetismo\ (x,y,z) = \frac{(\ data(x,y,z) * 1.0\ )}{\frac{32768}{2}}$$ Ec. 3-3 [41]

3.1.1.2 Sensor Humedad

El sensor integrado en el dispositivo es el HDC1000 desarrollado por *Texas Instruments*. Es un sensor digital de humedad y de temperatura de baja potencia y una precisión del 3%. Es un sensor capacitivo que viene calibrado de fábrica. El rango de operación del sensor de temperatura integrado es de -40 °C a 125°C. Su resolución es de 10 ms, pero el rango de trabajo del programa que viene cargado es de 100ms a 2.55 s. [41]

- **Configuración:** para empezar a recibir las lecturas del sensor se debe habilitarlo, por defecto realizará las mediciones cada segundo, dependiendo de la aplicación se pueden cambiar estos tiempos. Una vez que se deshabilita el sensor, se borrar todos los datos y se deja de reportar.

- **Datos:** el sensor de humedad envía dos valores enteros de 16 bits sin signo, la temperatura en el bit 0 y la humedad en el bit 1. En la Ec. 3-4 y Ec. 3-5 se describen las fórmulas para obtener el valor de la temperatura y humedad °C y %RH respectivamente.

$$Temperatura\ (°C) = \left(\frac{rawTemp}{65536}\right) * 165 - 40$$ Ec. 3-4 [41]

$$Humedad\ (\%\ RH) = \left(\frac{rawHum}{65536}\right) * 100$$ Ec. 3-5 [41]

3.1.1.3 Sensor Temperatura

El sensor integrado en el dispositivo es el TMP007 desarrollado por *Texas Instruments*. Es un sensor infrarrojo que internamente realiza una conversión digital analógica dando el resultado final al *SensotTag.* Se puede realizar la calibración ya que contienen una memoria EPROM integrada para almacenar los parámetros y lograr una mayor exactitud en las medidas. Tiene comunicación I2C y SMBus, resolución de 14 bits, 0.031 °C. [41]

- **Configuración:** para empezar a recibir las lecturas del sensor se debe habilitarlo, por defecto realizará las mediciones cada segundo, dependiendo de la aplicación se pueden

cambiar estos tiempos. Esta medida en un promedio de 4 medidas realizadas. Una vez que se deshabilita el sensor, se borrar todos los datos y se deja de reportar.

- **Datos:** el sensor de temperatura envía dos valores enteros de 16 bits sin signo, la temperatura ambiente en el bit 0 y la temperatura del objeto en el bit 1. En la Ec. 3-6 y Ec. 3-7 se describen las fórmulas para obtener el valor de la temperatura ambiente y del objeto en °C.

$$\text{it} = (\text{int})\ ((\text{rawAmbTemp}) >> 2)$$
$$T.Ambiente\ (°C) = \ ((\text{float})\text{it}) \ * \ \text{SCALE\_LSB} \qquad \text{Ec. 3-6 [41]}$$

$$\text{it} = (\text{int})\ ((\text{rawObjTemp}) >> 2)$$
$$T.Objeto\ (°C) = \ ((\text{float})(\text{it})) \ * \ \text{SCALE\_LSB} \qquad \text{Ec. 3-7 [41]}$$

- SCALE_LSB = 0.03125

3.2 Gateway

Uno de los principales problemas a los que se enfrenta el IoT es a la integración de múltiples dispositivos y protocolos. Por lo que es necesario emplear dispositivos que aporten soluciones ante estos escenarios. Un dispositivo de puerta de enlace o más conocido como *gateway*, es un puente para que diferentes tecnologías y protocolos se puedan comunicar. El *gateway* IoT toma los datos de los sensores, los traduce y se encarga de enviarlos tanto al internet como a otros dispositivos dependiendo de la necesidad del sistema. Existen algunos casos en los que los *gateways* realizan un preprocesamiento de los datos, dependiendo de la cantidad de información y de la aplicación.

3.2.1 *Raspberry Pi*

La *Raspberry Pi* [1]es un ordenador de placa simple de bajo costo diseñada *Raspberry Pi Foundation.* Es un microordenador de alto rendimiento y bajo consumo, que permiten el desarrollo de proyectos de varias índoles entre ellas robótica, automatización, IoT, etc. El modelo que se va

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

a emplear para el presente proyecto es la *Raspberry Pi 3*, ya que integra las principales tecnologías de red como son: WiFi, Bluetooth y Bluetooth Low Energy y ethernet 10/100. Tiene el sistema operativo *Raspbian*, de código abierto basado en *Debian*, tiene 1gb de memoria ram y procesador de 4 núcleos, además posee conector HDMI para poder conectarlo a un monitor.

Figura 3-3. Raspberry Pi 3[2]

3.3 Comunicaciones de Red

Para la conexión de las diferentes partes del sistema es necesario definir las tecnologías de red con las que se va a trabajar. Por un lado, el nodo sensor como se especificó anteriormente tiene comunicación BLE que se va a conectar con el *gateway* directamente. Los protocolos de red para la comunicación entre el servidor y el *gateway* son TCP en la capa de transporte y MQTT que es un protocolo a nivel de la capa de aplicación.

3.3.1 Protocolo *TCP*

Transmission Control Protocol (TCP) es un protocolo de transporte orientado a la conexión, que permite el intercambio de mensajes entre dispositivos dentro de una red. Para enviar los mensajes el protocolo se garantiza que existe la conexión entre los dos puntos, este inicio de comunicación se lo conoce como negociado en 3 pasos. El envío de los mensajes se basa en la segmentación de la información, es decir los divide en diferentes paquetes, se asigna un número

[2] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

de secuencia y son enviados a su destino. Para saber si los mensajes son recibidos, se debe responder con un ACK, el número de paquete y la cantidad de bytes recibida. En caso de ser necesario se realizará un reenvío del paquete para garantizar que llegue a su destino, con ello garantiza una de sus principales características la fiabilidad.

Este protocolo garantiza la entrega de los mensajes completos, para lo que establece una serie de mecanismos para saber si se ha entregado el mensaje y si es necesario reenviar los mensajes. Entre ellos están en *checksum*, con el cual el destinatario comprueba la validez del paquete recibido. De la misma forma al enviar un paquete se establece un temporizador, si pasado este tiempo no retorna el acuse de recibo, se da como perdido el paquete y se lo vuelve a enviar. Una vez terminada la comunicación, se finaliza la conexión en este caso se la realiza en 4 pasos por ser *full duplex*, lo que obliga a que en cada sentido se debe finalizar la comunicación.

3.3.2 Protocolo *MQTT*

MQTT por sus siglas en inglés *Message Queing Telemetry Transport* es uno de los protocolos más empleados para las soluciones IoT por ser muy ligero. Es un protocolo máquina a máquina (*M2M*) de transporte de mensajes bajo un modelo productor/suscriptor. Existen dos tipos de clientes que se muestran en la Figura 3-4: publicadores y suscriptores. Los publicadores envían la información mediante un canal determinado a un *broker* que es el encargado de distribuir el mensaje a todos los clientes suscriptores que estén escuchando dicho canal.

En la red pueden existir múltiples clientes de cada tipo. El suscriptor se puede registrar a más de un tópico o canales para recibir sus mensajes. Por otro lado, MQTT trabaja sobre TCP/IP y tiene la posibilidad de agregar autenticación mediante un usuario y contraseña. De la misma forma el intercambio de mensajes se lo puede configurar como SSL/TLS para tener comunicación cifrada. Posee tres tipos de calidad de servicio:

- 0: como máximo una vez (no ACK). Por lo general se emplea cuando la pérdida de información o las medidas de los sensores no son críticos y se puede esperar hasta que llegue el mensaje.
- 1: al menos una (ACK). Llegaran los mensajes, pero pueden existir duplicidad.

- 2: solo una vez (*handshake* 4 vías). Los mensajes deben llegar una única vez, se lo emplea en aplicaciones que no pueden existir duplicados.

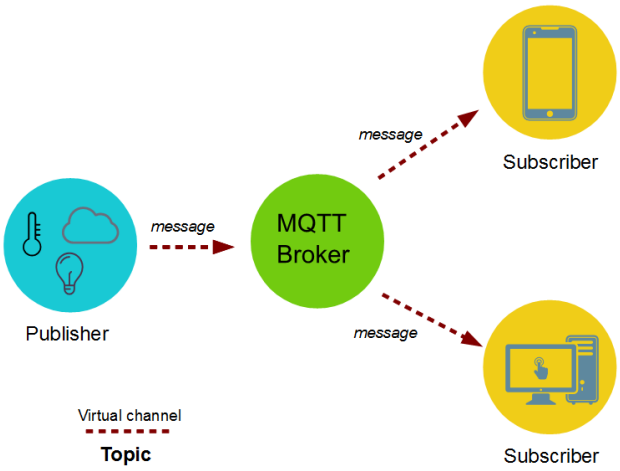


Figura 3-4. Protocolo MQTT[3]

3.3.3 *Bluetooth Low Energy (BLE)*

Es una tecnología inalámbrica muy bajo consume y de corto alcance. BLE también conocido como *Bluetooth Smart* es una parte del desarrollo de *Bluetooth* 4.0, no está diseñado para garantizar alta velocidad, pero si un bajo consumo, con una batería de botón puede funcionar por años.[4] BLE originalmente fue un proyecto desarrollado por Nokia, pero fue adoptado por *Bluetooth SIG* y es compatible con un gran número de dispositivos móviles incluyendo el sistema operativo de iOS. El protocolo BLE funciona en una arquitectura de 3 capas como se muestra en la Figura *3-5* : controlador, host y aplicación.

En la capa de aplicación se describe el funcionamiento y el uso que se le va a dar al desarrollo. Depende directamente de *Generic Access Profile* (GAP) y el *Generic Attribute Profile (GATT)*, que se refiere a como la aplicación recibe y envía los datos.

- Capa Física: BLE opera en la banda 2.4 Hz, que esta segmentada en 40 canales. Existen dos tipos de canales físicos: anuncio (*advertising*) y de datos. Tres de los canales de son de

[3] https://www.survivingwithandroid.com/mqtt-protocol-tutorial/

[4] https://medium.com/@zpcat/how-bluetooth-le-works-link-layer-b18475250259

tipo anuncio que están pensados para evitar las interferencias con el protocolo 802.11. Mientras que se dispone de 37 canales de datos que tiene una selección adaptativa.

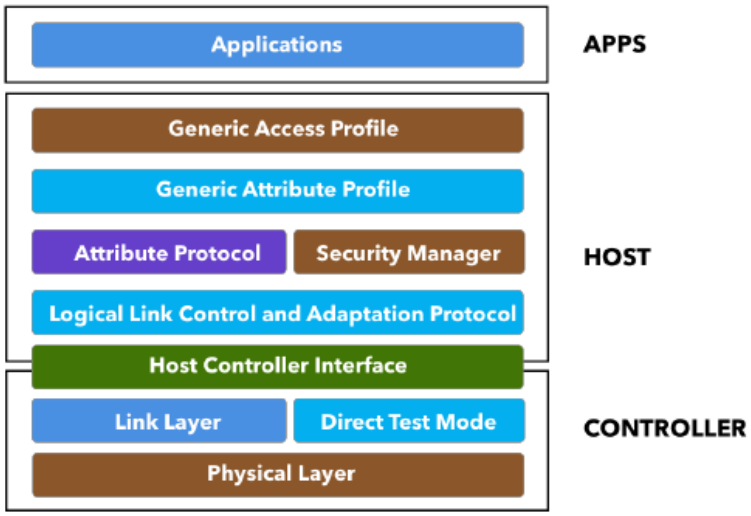


Figura 3-5. Arquitectura BLE [42]

- Capa de enlace: es la capa que interactúa entre la capa inferior, capa física y la HCI. Es la responsable de transmitir el estado del enlace que pueden ser 3: *advertising*, *scanning* y *connected*. Cuando el dispositivo se encuentra en estado de anuncio, permite que otros dispositivos que se encuentran en estado *scanning* puedan verlo y establecer conexión.
- Direct Test Mode: esta capa se la emplea durante la fabricación para realizar pruebas de operación y de certificación.

3.3.3.1 *Generic Acces Profile (GAP)*

Es un *framework* que permite a los dispositivos BLE interactuar entre ellos, es decir descubrir y conectarse. Los dispositivos pueden tener diferentes roles que son: *broadcaster*, observador, periférico y central. Hay que tomar en cuenta que un dispositivo puede operar con varios roles a la vez, por ejemplo, cuando un dispositivo móvil se comunica con un reloj inteligente actúa como dispositivo central, pero se puede conectar con un portátil en este caso es periférico.

- ***Broadcaster:*** dispositivo encargado de enviar paquetes de tipo *advertising*, no recibe paquetes ni acepta conexiones de otros elementos.

- **Observador:** dispositivo que está a la escucha de paquetes *advertising* para emplearlos en la aplicación, pero no establece conexión con el dispositivo.
- **Periférico:** dispositivo que envía paquetes de tipo *advertising* y permite conexión de dispositivos centrales. Cuando se conecta al dispositivo este se transforma en esclavo.
- **Central:** dispositivo que descubre y escucha a los dispositivos periféricos. Una vez conectado se convierte en maestro, que solicita la información.

3.3.3.2 *Generic Attribute Profile (GATT)*

El intercambio de información entre dispositivos es la razón fundamental por la que los dispositivos BLE establecen conexión. GATT define el formato de exposición de los datos a través de este protocolo. Trabaja bajo una arquitectura Cliente/Servidor. El servidor expone la información y realiza el control de su contenido, mientras que el cliente solicita el acceso a estos datos. Estos datos se organizan en servicios y características. En la Bluetooth SIG[5], se definen atributos genéricos para definir estas tablas GATT e integrarlas al sistema.

3.3.3.3 Características y Servicios

El primer término para referirnos a las características y servicios son los atributos que organizan servicios y características. Los atributos no son más que cualquier tipo de datos que es expuesto por un dispositivo BLE. Los servicios pueden contener una o varias características, por lo general se agrupan características que están relacionadas. Mientras que las características siempre forman parte de un servicio contienen un valor único y cualquier número de descriptores para el mismo.

[5] https://www.bluetooth.com/specifications/gatt/

3.4 Software Empleado

3.4.1 Base de Datos

En la implementación de un sistema IoT es de suma importancia el almacenamiento de datos, para trabajar con los datos. En el presente trabajo se propone el desarrollo de un gemelo digital por lo que es necesario llevar un registro del comportamiento del brazo robótico para su posterior análisis. Por otro lado, al trabajar en un entorno IoT en el que se integran diferentes dispositivos, que no tiene las mismas características se debe encontrar una alternativa para almacenar los datos.

Las bases de datos no relacionales (NoSQL), se caracteriza por no tener una estructura de almacenamiento de los datos, por lo que es la mejor solución para este tipo de sistemas. Otra característica fundamental de una base de datos NoSQL es que no usa el lenguaje SQL como base para sus consultas. Tomando en cuenta estos dos puntos se ha seleccionado MongoDB para el desarrollo de la solución.

3.4.1.1 *MongoDB*

Es un base de datos NoSQL, que guarda documentos en formato JSON. Tiene la capacidad de almacenar varios documentos, sin la necesidad de tener el mismo formato. Es una base de datos gratuita que trabaja en un entorno distribuida lo que quiere decir que garantiza alta disponibilidad y un alto nivel de escalabilidad horizontal, se pueden distribuir *clusters* a nivel mundial para mantener la información respaldada. Entre las principales características de este gestor de base de datos se destaca: [43]

- **Modelo de datos intuitivo:** los datos con los que trabajan están en formato JSON, por lo que es mucho más sencillo manejarlo a nivel de código, no es necesario realizar una búsqueda compleja ya que todos los datos relacionados se los puede almacenar en un único documento.
- **Esquema flexible:** no tiene un esquema definido, es decir en cada almacenamiento de los documentos se puede trabajar con un esquema diferente dependiendo de la necesidad del proyecto.

- **Universal:** al trabajar con documentos tipo JSON, se acopla fácilmente a varios lenguajes de programación y su interpretación es sencilla.
- **Potente:** si bien no se basa en el lenguaje de consulta estructurada (SQL), el lenguaje de consulta es completo, soporta índices, agregaciones, agrupaciones, etc. La diferencia que no son las mismas estructuras de datos se trabaja con varios documentos a la vez.
- **Sistema distribuido:** una de las características principales de esta base de datos es su resiliencia y escalabilidad a nivel mundial. Al manejar documentos es sencillo su movimiento de un servidor a otro. Se pueden establecer configuraciones de replicación dependiendo de la necesidad de la aplicación garantizando el acceso a los datos en cualquier momento.

3.4.2 Sistema SCADA

Un SCADA (Supervisory Control And Data Acquisition) es un sistema de supervisión, control y adquisición de datos. Es un software mediante el cual se puede realizar monitorización y ejecutar acciones de control. Se debe tener claro que un SCADA no es un sistema de control, simplemente se comunica con los dispositivos de control de la planta, para mostrarlos en pantallas de operación.

3.4.2.1 *Ignition: The New SCADA*

Es un software SCADA integra las características de un modelo de licenciamiento ilimitado con un desarrollo instantáneo basado en la web. Es una plataforma escalable y modular. Ignition permite de una manera sencilla el control de los procesos, controlar, visualizar y analizar toda la información sin límites. Las principales características del sistema son: [44]

- Control y análisis en tiempo real.
- Potentes herramientas de visualización.
- Desarrollo sencillo y rápido.
- Posibilidad de crear aplicaciones totalmente *responsive.*
- Arquitectura modular, adaptable a las necesidades de cada aplicación.

Figura 3-6. Ignition, Inductive Automation [44]

Ignition permite levantar una arquitectura distribuida. Por un lado, el servidor tiene acceso a todos los dispositivos de medición, sensores, actuadores, controladores lógicos, etc. De la misma forma se conecta con diferentes bases de datos y visualizar pantallas desde cualquier dispositivo. Las aplicaciones de *Ignition* se basan en acceso web, en el entorno de desarrollo de aplicaciones esto es una característica básica, pero a nivel de software industrial no todos cumplen con ese requisito. Con esto se garantiza el acceso a la aplicación desde un sin número de dispositivos que tengan acceso al servidor, sin la necesidad de instalar software extra.

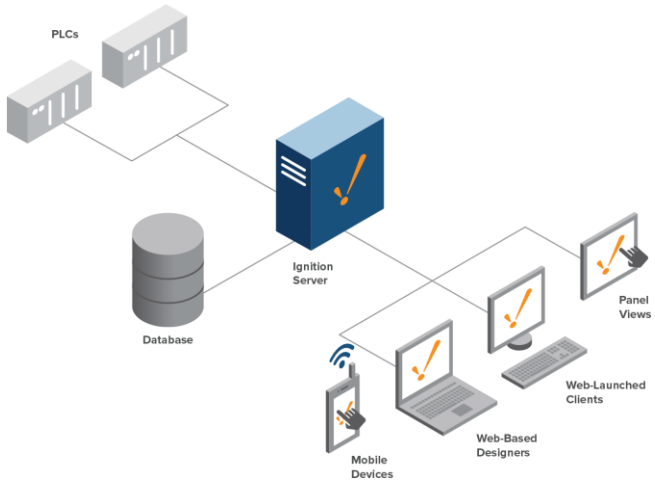


Figura 3-7. Arquitectura básica de sistema de control con Ignition [44]

Capítulo 4 - Aplicación a Brazo Robótico

En el presente capítulo se detallará la aplicación para realizar las pruebas de concepto del *Digital Twin*. El escenario seleccionado fue un brazo robótico desarrollado por la empresa *Dobot*, del cual se va a medir las aceleraciones en cada uno de sus movimientos y almacenarlos en la base de datos para su posterior análisis. Por otro lado, en la segunda parte se propone un entorno simulado, con la ayuda de Matlab, se simulará el comportamiento de un robot y se enviaran los datos al sistema para evaluar su comportamiento. Por último, se describirá la infraestructura final y la interfaz de usuario con la que se pueden visualizar los resultados.

4.1 *Dobot Magician*

El *Dobot Magician* es una de las alternativas que ha creado *Dobot*, que es un robot de escritorio multifuncional para trabajos académicos y de investigación. Soporta varios tipos de programación, guiada, visual por bloques y código en el entorno propio de *DobotStudio* o en *Python*[6]. Se pueden instalar diferentes actuadores finales: succionador, pinza, laser y porta lápiz. Por otro lado, soporta la integración de sensores externos directamente, lo que convierte al *Dobot Magician* en una herramienta completa para el desarrollo de investigaciones. Las partes principales del brazo robótico se las puede ver en la Figura 4-1: base, brazo, antebrazo, actuador final, etc.

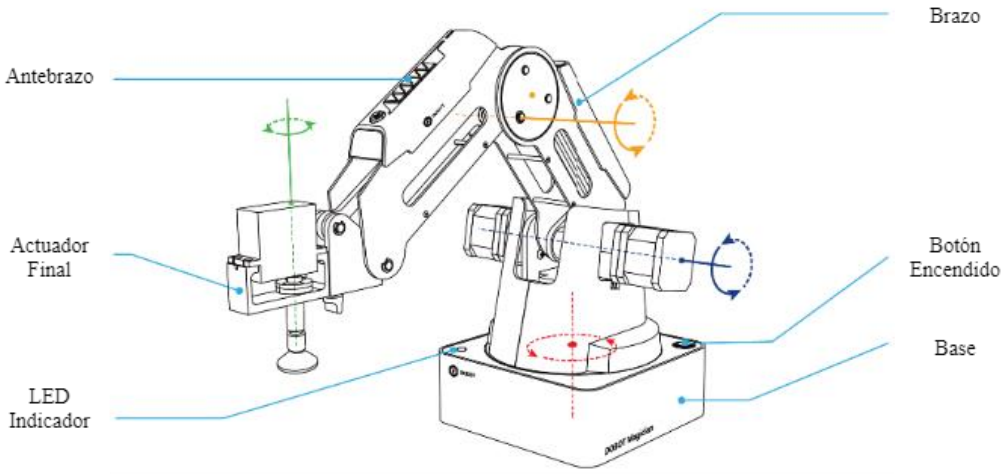


Figura 4-1. Esquema Dobot Magician [46]

[6] https://www.python.org/

El robot tiene dos tipos de sistemas de coordenadas: el articulado y el cartesiano, que se muestran en las Figura *4-2* y Figura *4-3* respectivamente. En el primer sistema el número de coordenadas viene dado por el número de articulaciones del robot, en el caso de que no esté instalado el actuador final funciona con J1, J2 y J3. El sentido positivo de estas coordenadas es en sentido antihorario. [46]

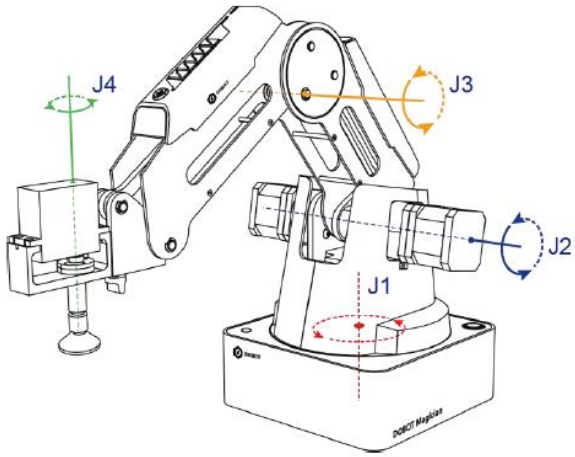


Figura 4-2. Sistema de Coordenadas Articulado [46]

En el sistema cartesiano, el punto de referencia es la base como se muestran en la Figura 4-3, que es el origen de los tres motores instalados en la base, brazo y antebrazo. El eje de las X es perpendicular a la base hacia adelante. El eje Y es perpendicular a la base, hacia la izquierda y la Z es perpendicular en sentido hacia arriba. [46]

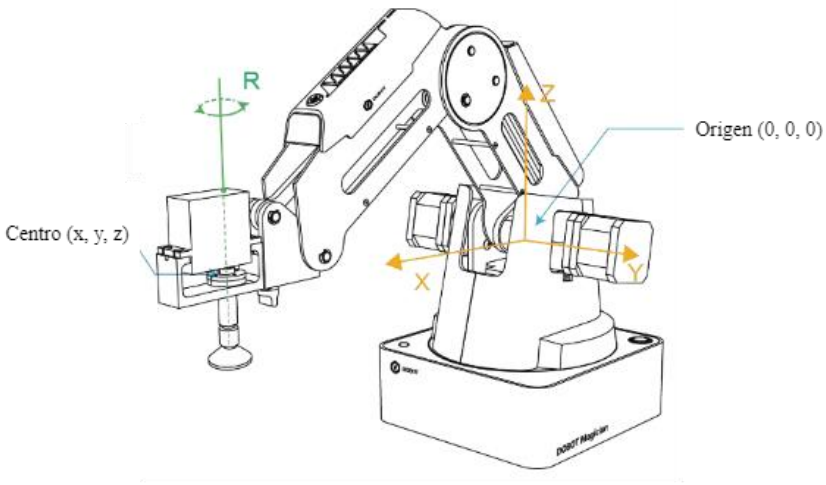


Figura 4-3. Sistema de Coordenadas Cartesiano [46]

Por otro lado, una función muy útil para realizar pruebas en el presente trabajo son los tipos de movimiento que puede realizar el brazo robótico que son tres: *jogging*, punto a punto y ARC. En el presente trabajo nos vamos a basar en el modo de funcionamiento punto a punto que se subdivide en tres: movimiento articulado (*MOVJ*), movimiento rectilíneo (*MOVL*) y salto (*JUMP*).

Para el desplazarse desde el punto A hasta el punto B con el MOVJ, cada articulación va a iniciar desde el ángulo de inicio hacia el objetivo independientemente de la trayectoria. El movimiento MOVL, realizará un movimiento lineal como se muestra en la Figura *4-4*.

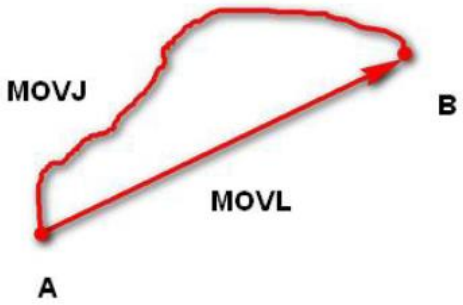


Figura 4-4. MOVJ y MOVL [46]

Por su parte para realizar el movimiento tipo salto emplea el movimiento MOVJ, en el punto A se eleva, se mueve horizontalmente hasta llegar por encima del punto B y bajar hasta llegar al objetivo. El movimiento se lo puede visualizar en la Figura 4-5.

Figura 4-5. Movimiento tipo Salto (JUMP) [46]

De la misma forma que se puede seleccionar el tipo de movimiento con el que se va a trabajar, el robot permite especificar la velocidad y aceleraciones con las que se va a trabajar. Las unidades de trabajo son mm/s y el rango recomendable es de 0 mm/s hasta 500 mm/s y la aceleración de 0 mm/s^2 hasta 500 mm/s^2. [46] Con estos parámetros de configuración se puede establecer un entorno adecuando para realizar diferentes pruebas a la infraestructura que se propone en el presente trabajo.

4.2 Simulación Brazo Robótico Matlab

Una de las alternativas que se tienen al momento de realizar modelos es el emplear entornos simulados. Existen un sin número de herramientas que permiten realizar simulaciones de brazos robóticos, algunas de estas son de pago y otras no. En el presente trabajo se decidió trabajar con Matlab, por el potencial que tiene, por el conocimiento del manejo que se tiene con la herramienta y porque la universidad provee licencias a sus estudiantes y profesores para las distintas actividades académicas.

Matlab es una herramienta muy potente que su principal función es análisis de datos y diseño de modelos, mediante la utilización de programación sencilla. Matlab es una herramienta que continuamente está en desarrollo, se actualiza e integra las tecnologías más importantes, entre ellas están: aprendizaje profundo, *machine learning* (ML), visión artificial, procesamiento de señales, robótica, sistemas de control, etc. De la misma forma ofrece un gran abanico de *toolbox* para disminuir el tiempo de desarrollo de los modelos. Un *toolbox* es un conjunto de herramientas y funciones integradas en el sistema.

Otra de las herramientas que hace de Matlab potente es *Simulink*, que permite el desarrollo de modelos y realización de pruebas. Ofrece varias opciones para modelado de sistemas, establecer entornos de simulación y estudiar el comportamiento del sistema.

En el presente trabajo se emplea el *Robotics System Toolbox*, que integra la posibilidad de simular comportamiento de robots industriales. Se puede crear nuevas definiciones o emplear robots que están en las librerías de Matlab, que son de las marcas más utilizadas en nuestro medio. Realizado el modelo se necesita visualizar el comportamiento del robot y su movimiento, para lo que se emplea *Simulink 3D Animation*, que integra aplicaciones de ejemplo, funciones y bloques que hacen posible la visualización del comportamiento de los modelos creados.

El modelo desarrollado para el trabajo se basa en un ejemplo de Matlab, del *Robotics System Toolbox*, denominado *Visualize Manipulator Trajectory Tracking with Simulink 3D Animation.*[47] Para la creación del modelo Figura *4-7* se emplearon los bloques: *Polynomial Trajectory*, *Joint Space Motion Model*.

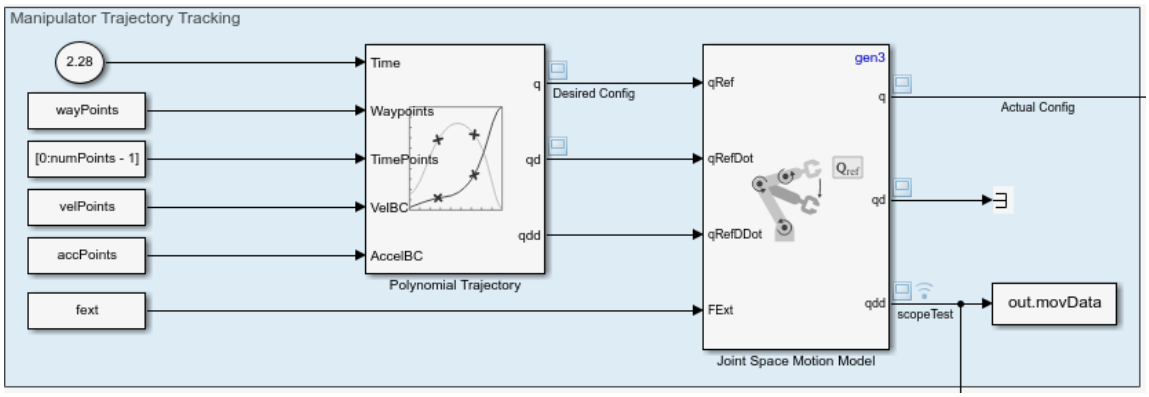


Figura 4-6. Modelo Brazo Robótico

- ***Polynomial Trajectory:*** generación de trayectorias para viajar a través de los puntos definidos. Se establecen los puntos de referencia, la velocidad, aceleración y los puntos de tiempo. Con esto se genera la trayectoria polinómica que pueden ser: cúbica, quíntuple o *B-Spline*. [48]

- ***Joint Space Motion Model:*** se encarga de modelar el comportamiento de un brazo robótico, dada una trayectoria. Se deben definir las propiedades del robot con el que se desea trabajar, como se mencionó anteriormente existen manipuladores definidos, en este caso se emplea el *kinovaGen3*[7]. En este bloque se puede definir el tipo de movimiento con el que se quiere trabajar: controlado por torque, movimiento independiente por articulaciones o un control PD. [48]

Para la visualización del modelo Figura *4-7* se emplearon los bloques: *VR RigidBodyTree, Get Transform, Coordinate Transformation Conversion* y *VR Sink.*

- ***VR RigidBodyTree:*** permite la visualización del manipulador especificado en la entrada de configuración. En la configuración por lo general se especifica el número de articulaciones, el actuador final, etc. La configuración es un vector de las diferentes posiciones de las articulaciones. [49]

- ***Get Transform:*** a partir de un grupo de puntos los transforma en puntos del manipulador, empleando la configuración del robot que se indique. [48]

[7] https://www.kinovarobotics.com/en/products/gen3-robot

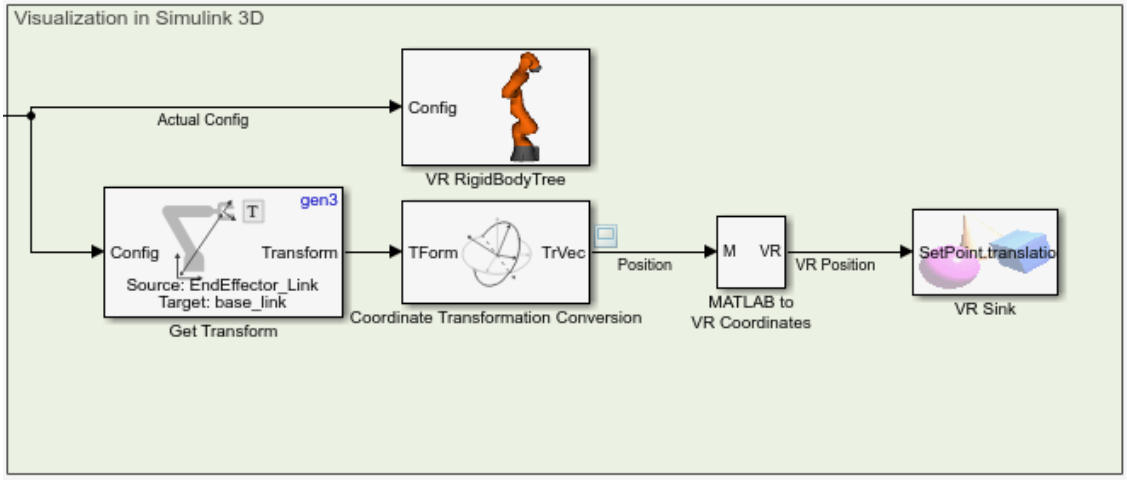


Figura 4-7. Visualización 3D Brazo Robótico. [47]

- ***Coordinate Transformation Conversion:*** es un bloque de transformación de coordenadas, ante una configuración de entrada se transforma a la especificada en la salida. [48]
- ***VR Sink:*** este es el bloque final para poder simular el modelo. Es decir, es la salida del modelo para poder controlarlo en un entorno virtual. Traduce los valores que recibe en su entrada a puntos en el mundo virtual. [49]

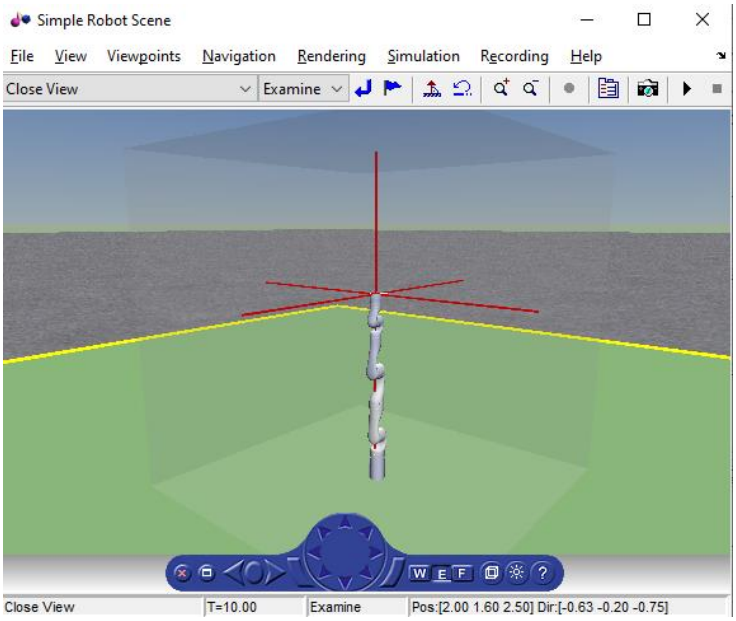


Figura 4-8. Simulación Brazo Robótico

La simulación final al ejecutar el programa se muestra en la Figura *4-8*, el robot manipulador *kinovaGen3*, que realiza los movimientos que se definen para las diferentes pruebas que se van a

analizar en el capítulo 5 el presente trabajo. De la misma forma se han insertado *scopes* para la visualización de las señales que entrega cada uno de los bloques, verificando el correcto comportamiento del sistema.

4.3 Descripción de la Solución

Una vez conocidas las diferentes herramientas y tecnologías que se van a emplear en el desarrollo de la infraestructura a implementar, se explicará a detalle cada parte del sistema. En la Figura *4-9*, se muestra el sistema levantado. Por un lado, se adquieren los datos del brazo robótico por medio del *SensorTag* que realizará mediciones de temperatura del objeto, temperatura ambiente, aceleración en los tres ejes y rotación en los tres ejes. Las medidas serán enviadas por BLE, que es un medio de comunicación de bajo consumo. Cabe mencionar que desde el SensorTag no podremos enviar directamente los datos a la base de datos, por lo que es necesario implementar un *gateway*, que sea capaz de traducir los datos recibidos por BLE y enviarlos al servidor.

El *gateway* se encarga de recibir los datos provenientes del *SensorTag* y enviarlos al servidor, este envío se lo realiza por MQTT por sus características y prestaciones que se detallaron en el capítulo anterior. Una vez que el servidor recibe estos datos por un lado los almacena en la base de datos MongoDB para tener un registro del comportamiento del *Dobot Magician*, mientras que en tiempo real los muestra en el *dashboard* desarrollado en *Ignition*, que puede ser visualizado por cualquier usuario que tenga acceso.

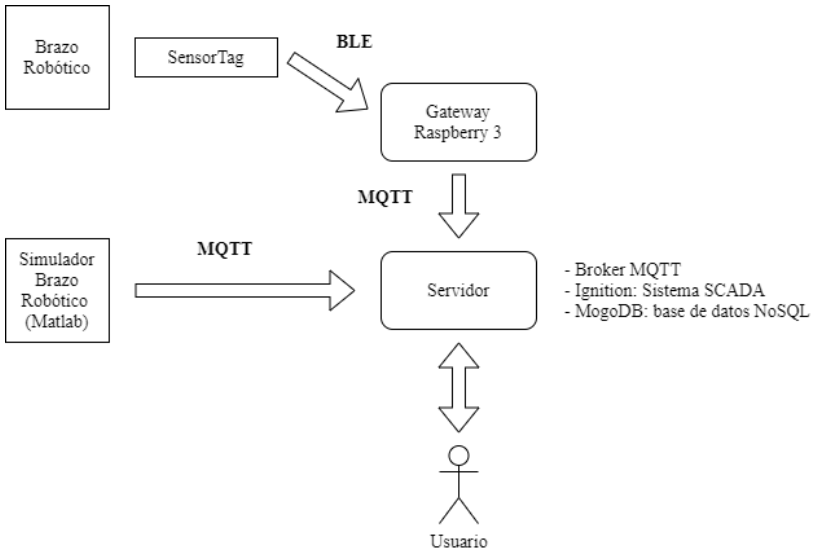


Figura 4-9. Infraestructura IoT implementada

La otra alternativa para evidenciar el potencial de la infraestructura es trabajar con un entorno simulado, para esto se emplea Matlab que simulará el comportamiento de un brazo robótico. Los movimientos serán enviados al servidor por MQTT y se realizará el mismo procedimiento que en el brazo robótico físico. Por un lado, se almacenan para tener un registro del comportamiento y por otro lado se los visualiza en el *dashboard* en tiempo real.

Se ha mencionado que los datos visualizados en el *dashboard* serán en tiempo real, pero al tener toda la información en la base de datos se puede acceder a los registros de los movimientos y del estado del entorno de funcionamiento del robot (temperatura ambiente y del objeto). Con esto se puede visualizar el histórico del comportamiento, se podrá realizar consultas en el rango de fechas deseado.

Por otro lado, otro objetivo del trabajo es emplear los datos para realizar mantenimiento predictivo. Se decidió realizar una detección de anomalías en cada uno de los movimientos del robot, por lo que después de cada registro que tenga el manipulador, se expondrán los datos a un algoritmo de detección de anomalías y se activará una alarma en el caso de que se detecte datos sospechosos.

4.3.1 Adquisición de datos

Se va a trabajar con las aceleraciones y rotaciones del brazo robótico por lo que es necesario definir el tiempo de muestreo para la adquisición de los datos. El *SensorTag* tiene una resolución por defecto configurable de 100ms a 2.5seg. Al tratarse de aceleraciones de un brazo robótico industrial se deben detectar los mínimos picos que puede tener el movimiento y se estableció una frecuencia de muestreo de 100Hz (10ms). Para obtener este desempeño del sensor se realizó un cambio en la programación del *firmware* del dispositivo y se logró disminuir la resolución, cabe recalcar que los manuales de funcionamiento del sensor especifican que la resolución mínima trabajo es de 10ms, por lo que nos aseguramos de que va a funcionar en este rango de trabajo. [41]

Los parámetros del ambiente en el que el robot está trabajando son relevantes al momento de revisar el desempeño del robot y tomar decisiones de funcionamiento, por lo que se va a almacenar la información de temperatura del ambiente del robot y del sensor. Estas medidas son relativamente lentas, por lo que no es necesario una frecuencia de muestreo rápida. El sensor permite tomar

medidas entre 100ms a 2.5seg, en este caso se ha seleccionado 2.5seg que es suficiente para estas características.

Como se explicó en el Capítulo 3, el protocolo BLE trabaja con servicios y características. Para poder acceder a ellos se debe verificar la documentación del sensor. En la Tabla 4-1 y Tabla *4-2*, se muestran las configuraciones necesarias para activar y recibir notificaciones del sensor de movimiento y de temperatura respectivamente.

Tabla 4-1. Configuraciones Sensor Movimiento *SensorTag* [41]

Tipo	UUID	Acceso	Tamaño (bytes)	Descripción
Data	AA81*	R/N	18	GyroX[0:7], GyroX[8:15], GyroY[0:7], GyroY[8:15], GyroZ[0:7], GyroZ[8:15], AccX[0:7], AccX[8:15], AccY[0:7], AccY[8:15], AccZ[0:7], AccZ[8:15], MagX[0:7], MagX[8:15], MagY[0:7], MagY[8:15], MagZ[0:7], MagZ[8:15]
Notificación	2902	R/W	2	Escribir 0x001 para habilitar notificaciones, 0x0000 para deshabilitarlas
Configuración	AA82*	R/W	2	Un bit para cada eje rotación y aceleración (6), magnetómetro (1), habilitar wake-on-motion (1), rango acelerómetro (2). Si se escribe 0x0000 se deshabilita el sensor
Periodo	AA83*	R/W	1	Resolución 10ms. Rango 100ms (0x0A) hasta 2.55 seg (0xFF). Por defecto (0x64)

Una vez realizado los cambios en el programa del *SensorTag* se descargó el firmware con la ayuda del programa *UniFlash*[8] *de Texas Instruments*. Para leer los datos de las mediciones del *SensorTag* desde el *gateway,* el desarrollo se lo hizo en *Python* empleando el módulo *bluepy*[9], que permite conectarse y realizar configuraciones a dispositivos BLE.

Para realizar la conexión con el dispositivo en el programa se tiene que crear el dispositivo periférico, definir las características y enviar los códigos de configuración para habilitar los sensores y las notificaciones de cada uno. El Gateway va a estar a la escucha de las notificaciones provenientes del *SensorTag*. En el sensor de movimiento se habilitó la configuración de *wake-on-motion* con un tiempo de cinco segundos, lo que significa que al no detectar movimiento en un

[8] http://www.ti.com/tool/UNIFLASH

[9] https://github.com/IanHarvey/bluepy

bucle de 5 segundos pasará al modo de baja potencia. Por defecto este tiempo está en 10 segundos, de la misma forma se modificó al momento de actualizar el firmware.

Tabla 4-2. Configuraciones Sensor Temperatura *SensorTag* [41]

Tipo	UUID	Acceso	Tamaño (bytes)	Descripción
Data	AA21*	R/N	4	Object [0:7], Object [8:15], Ambience[0:7], Ambience [8:15]
Notificación	2902	R/W	2	Escribir 0x001 para habilitar notificaciones, 0x0000 para deshabilitarlas
Configuración	AA22*	R/W	1	Escribir 0x01para habilitar el sensor y 0x00 para apagarlo
Periodo	AA23*	R/W	1	Resolución 10ms. Rango 100ms (0x0A) hasta 2.55 seg (0xFF). Por defecto (0x64)

4.3.2 *Broker MQTT*

La infraestructura define la integración de un *broker MQTT* que se lo instaló en el servidor. El *broker* es una de las partes fundamentales de la infraestructura, ya que ayuda en el intercambio de mensajes entre la etapa de adquisición de datos y su almacenamiento. La instalación se la realizó en *Ubuntu 18.04*, se integraron las librerías de *mosquitto* [10]de *Eclipse Foundation.*

4.3.3 Envío de datos del Gateway al Servidor

Una vez que los datos están en el *gateway*, se los debe enviar al servidor para poder trabajar con ellos, esta comunicación se la realiza con el protocolo *MQTT.* El módulo *paho*[11], de *Python* integra la capacidad de crear clientes *MQTT* publicadores y suscriptores en nuestros programas. El esquema de tópicos con el que se trabajará es ***idDigitalTwin/idSensor/#***, donde:

- **idDigitalTwin:** nombre o identificador del DigitalTwin.
- **idSensor:** nombre o identificador del sensor.

[10] https://mosquitto.org/

[11] https://github.com/eclipse/paho.mqtt.python

- **#:** en casos específicos se puede necesitar más o menos subsecciones por lo que de aquí para adelante se define dependiendo del sensor. Por lo general si no se tiene más información, será el valor medido.

Los valores recibidos del *SensorTag*, deben ser procesados como se detalló en el Capítulo 3, los datos se reciben de diferentes formas dependiendo el sensor. Para esto se implementó otro programa en *Python*, que es llamado desde el programa principal para realizar las diferentes conversiones dependiendo del sensor, es decir aplicando las ecuaciones Ec. 3-1, Ec. 3-2, Ec. 3-6 y Ec. 3-7 para aceleración, rotación, temperatura ambiente y temperatura del objeto respectivamente.

Para establecer la conexión con el *broker MQTT* que se encuentra en el servidor, en el programa se debe definir la dirección y puerto del servicio. La primera configuración de seguridad que se realizó para esta comunicación fue cambiar el puerto con el que *MQTT* trabaja por defecto 1883 y 8883 sobre TLS. De la misma forma, se estableció usuario y contraseña para poder establecer la conexión y evitar intrusos en la red. Una vez realizada la conexión con el *broker*, se publica el mensaje en el tópico dependiendo de la medida recibida.

4.3.4 Envío de datos de Matlab al Servidor

La simulación del brazo robótico es idónea para realizar pruebas de funcionamiento de la infraestructura. Por lo que los datos generados en los movimientos del robot son enviados directamente al servidor, para su procesamiento. En este caso no es necesario el *gateway* para traducir los datos, ya que Matlab, permite crear un cliente *MQTT* y publicar los mensajes en el *broker*. Con esto nuevamente se realiza el procedimiento de establecer la conexión con el *broker MQTT* y conforme se desarrolla el movimiento se envían los datos en tiempo real para simulación real del comportamiento del robot.

Las funciones de implementación *MQTT* en Matlab trabajan en scripting y no en el entorno de *Simulink*, lo que hace necesario tomar los datos desde el *workspace*, pero se lo debe realizar en tiempo real para tener el modelo idéntico al del funcionamiento con el *SensorTag* y el brazo robótico físico. Para esta implementación se integró un bloque *Matlab Function* que llama a funciones implementadas de Matlab, en esta función se realiza la publicación de los datos

obtenidos desde la simulación. Con esto se logró enviar los datos en tiempo real del movimiento del robot. Para evitar problemas de diferencias de tiempo, al mensaje se agrega una estampa de tiempo, con ello se asegura que se almacene el dato con la hora exacta.

4.3.5 Almacenamiento de los datos en MongoDB

Posterior a la etapa de adquisición, preprocesamiento y envío de los datos, se deben visualizarlos en tiempo real y almacenarlos para tener un registro del comportamiento del *Digital Twin.* En el servidor se implementó un programa en *Python* que está a la escucha de todos los mensajes *MQTT* que pasan por la red. Cabe recalcar que dependiendo del tópico proveniente se arma un documento para almacenarlo en la base de datos. Para la implementación del código se empleó el módulo *pymongo*[12] que permite interactuar con MongoDB.

La propuesta es definir una colección para el *Digital Twin*, en este caso se llama *Dobot*. Dentro de la colección se disponen de todos los documentos con los datos provenientes de los datos obtenidos mediante el simulador y el robot físico. Al trabajar con una base de datos NoSQL, no es necesario definir una estructura específica, pero se debe establecer las claves de los documentos para organizar la información y poder trabajar con los datos, es por esto que se definieron cuatro tipos de documentos que se resumen en la Tabla *4-3*: movimiento, detalles de movimiento, sensores y alarmas.

- **Movimiento:** este documento tiene la información de los sensores de aceleración y rotación en cada uno de los tres ejes, la estampa de tiempo y un identificador del movimiento realizado.
    - **movCode:** identificador del movimiento. 0 = general, 1 = secuencia 1, 2 = secuencia 2, 3 = secuencia 3.
- **Detalle Movimiento:** para verificar el comportamiento del brazo robótico se deben establecen tipos de movimientos y se lleva un registro del inicio y fin de cada uno de ellos.

[12] https://pymongo.readthedocs.io/en/stable/

- **sensorMode:** modo de funcionamiento del *Digital Twin. "Train", "Normal", "PredML" y "TrainML".*
- **movStart:** estampa de tiempo inicio movimiento.
- **movEnd:** estampa de tiempo fin movimiento.
- **messValue:** registro de la velocidad o fuerza externa para trabajar con modelos de inteligencia artificial.

- **Sensores:** este documento está destinado a almacenar los distintos sensores que se pueden integrar al sistema, se puede ver que se pueden agregar cualquier dispositivo, simplemente se define un identificador de sensor y se crea el documento.
- **Alarmas:** en el caso de que se detecte un problema en el movimiento realizado se debe registrar para poder consultar y verificar que efectivamente existe un problema en el comportamiento del robot.
  - **sensorCode:** código del sensor y/o eje que se activó la alarma.
  - **almValue:** valor que activó la alarma.
  - **almTime:** registro estampa tiempo activación de la alarma.
  - **almMovCode:** identificador del movimiento. 0 = general, 1 = secuencia 1, 2 = secuencia 2, 3 = secuencia 3.
  - **almMovStart:** estampa de tiempo inicio movimiento.
  - **almMovEnd:** estampa de tiempo fin movimiento.

Tabla 4-3. Documentos definidos en MongoDB

Movimiento	Detalles Movimiento	Sensores	Alarmas
time	movCode	time	sensorCode
sensor	sensorMode	sensor	almValue
movCode	movStart	value	almTime
sensorAccX	movEnd		almMovCode
sensorAccY	messValue		almMovStart
sensorAccZ			almMovEnd

sensorGyrX			
sensorGyrY			
sensorGyrZ			

Con esta explicación de la definición de los documentos se puede evidenciar que el sistema es altamente escalable, ya que se pueden agregar colecciones de nuevos *Digital Twin* o simplemente agregar sensores al existente, simplemente se definen los identificadores en el caso de sensores nuevos. De la misma forma, si se desea agregar un sensor o datos con alguna estructura específica se lo puede integrar sin la necesidad de realizar configuraciones en la base de datos.

4.3.6 Algoritmo de detección de anomalías

Al tener un registro del comportamiento del brazo robótico se puede procesar los datos y realizar un sin número de análisis relevantes para el *Digital Twin.* En el presente proyecto se realizará un algoritmo para detectar comportamientos anormales, con el fin de tomar decisiones y realizar mantenimiento predictivo, disminuyendo el tiempo de parada del equipo o del sistema como se revisó en el Capítulo 2.

En el documento “detalles movimiento” almacenado en la colección de MongoDB, se detalla el tipo de movimiento y el modo de funcionamiento del *Digital Twin*, es con estos registros que se trabaja para aplicar el algoritmo de detección. En el caso de que está en modo ***“Train”***, solo se realizará el registro en la base de datos y este algoritmo no funcionará porque se está almacenando los registros del patrón del movimiento, es decir las aceleraciones en condiciones normales de funcionamiento. En el modo de funcionamiento es ***“Normal”***, es cuando se ejecuta el algoritmo, con la ayuda del identificador del movimiento, es decir dependiendo del tipo de movimiento se selecciona el respectivo patrón de la base de datos.

Es muy importante tomar en cuenta que la señal que proviene del simulador o del *SensorTag*, puede contener ruido lo que se debería eliminar para hacer un análisis. Por esta razón se debe implementar un filtro en las señales, ya que en las pruebas que se han realizado se encontraron valores que perjudican al correcto funcionamiento del algoritmo. Para la implementación del filtro

se emplea *signal.filtfilt* [13]del módulo *scipy*, que recibe los datos de configuración y la señal, la salida va a ser el nuevo conjunto de datos para analizar.

Para realizar el análisis se emplea el principio de correlación entre dos señales. Lo que muestra el coeficiente de correlación es el grado de similitud entre dos señales. Un problema en este punto es que no siempre las señales tendrán en mismo número de puntos, si bien es cierto el muestreo se lo realiza con la misma frecuencia, pero el movimiento puede sufrir varios cambios, en duración, trayecto, etc. Esta es la razón por la que se debe implementar un proceso para igualar el número de puntos que se tienen en las dos señales.

Para igualar el número de muestras de las señales se realiza una interpolación, que no es más que la generación de un número de puntos deseado, partiendo de un conjunto de datos. En este caso *Python* posee la clase *interpolate.interpld* [14] del módulo *scipy*. Partiendo de dos grupos de datos, devuelve una función aproximada que satisface a los puntos dados. Con esto se puede evaluar la función en el rango de puntos deseado. Una vez igualadas las señales se calcula la correlación de las señales con la ayuda de la función *numpy.corrcoef*[15].

Una vez obtenido el coeficiente de correlación se realizan dos acciones: se publica el valor en un tópico *MQTT*, que está a la escucha desde *Ignition* y en el caso de que este por debajo del umbral definido de correlación se almacena en la base de datos en el documento de tipo “alarmas” para llevar un registro de los eventos anormales que se han producido.

4.3.7 Inteligencia Artificial aplicada al *Digital Twin*

Al tener almacenados los datos del comportamiento del brazo robótico se pueden realizar varias pruebas y estudios. Una posibilidad es aplicar inteligencia artificial (AI) para validar el comportamiento y detectar movimientos fuera de lo normal. Es una alternativa al algoritmo que se presentó anteriormente. La inteligencia artificial trata de representar el comportamiento del

[13] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html

[14] https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html

[15] https://numpy.org/doc/1.18/reference/generated/numpy.corrcoef.html

cerebro humano ante diferentes experiencias, en otras palabras, es el comportamiento y las tareas que va a realiza una máquina ante ciertas condiciones. Dentro de la inteligencia artificial está el *Machine Learning* (ML) que es un subconjunto de la AI que aplica técnicas estadísticas para medición de rendimiento y mejorar el desempeño de los modelos.

Dentro del aprendizaje existen dos grandes grupos, el supervisado y el no supervisado. En el primero se conoce el objetivo o valor que se quiere conseguir, mientras que en el aprendizaje no supervisado se basa en agrupaciones de conjuntos que tengan las mismas características.

Con esto lo que se propone es aplicar técnicas de aprendizaje supervisado, para determinar si el *Digital Twin*, está trabajando en condiciones normales o necesita un ajuste. Para generar un modelo se necesitan datos de entrenamiento, estos datos se los va a obtener del comportamiento normal del robot, entre más información se obtenga se podrá construir un mejor modelo. En el presente trabajo se aplican dos técnicas, para mostrar que obtenidos y almacenados los datos se pueden aplicar técnicas de inteligencia artificial. Los modelos que se van a implementar son: regresión lineal y redes neuronales.

El desarrollo de esta parte del proyecto se empleará nuevamente *Python*, con la ayuda del módulo *sklearn.linear_model*, que nos permite crear modelos de regresión lineal con las configuraciones necesarias. Para implementar la red neuronal se ocupa el módulo *tensorflow* y *keras*. Cabe destacar que las últimas versiones de *keras* vienen embebidas en el módulo de *tensorflow*, por lo que no es necesario realizar la instalación del paquete *keras*.

Para la creación de los modelos se va a trabajar con dos tipos de datos en base a una fuerza externa y a la velocidad del movimiento. Se procederá a entrenar los modelos y comparar su desempeño. Cuando el modelo detecte que los datos de entrada están fuera de los rangos permitidos se realizaran las dos acciones anteriores: se publica el valor en un tópico *MQTT*, que está a la escucha desde *Ignition* y se almacena en la base de datos en el documento de tipo “alarmas” para llevar un registro de los eventos anormales que se han producido.

4.4 Interfaz Ignition

La idea del proyecto es generar un *Digital Twin* de un brazo robótico con la ayuda del IoT en un entorno industrial (IIoT), por lo que la mejor alternativa para realizar la interfaz gráfica y probar

el desempeño del sistema es una herramienta que se emplee en procesos de producción y la industria en general. La interfaz tiene tres objetivos principales: visualización en tiempo real del comportamiento del *Digital Twin*, consulta de históricos y visualización de alarmas del sistema. El Apéndice B contiene el manual de usuario de la aplicación desarrollada.

4.4.1 Pantalla Resumen

Para visualizar los datos en tiempo real, en *Ignition* se crea un cliente *MQTT* que está suscrito a todos los valores de interés para su monitoreo, con ello se puede visualizar los valores actuales de cada uno de los sensores. De la misma forma, se generaron gráficas con los últimos registros de los sensores en el caso de las temperaturas los últimos 30 minutos y para el movimiento los últimos 5 segundos, como se muestra en la Figura 4-10.

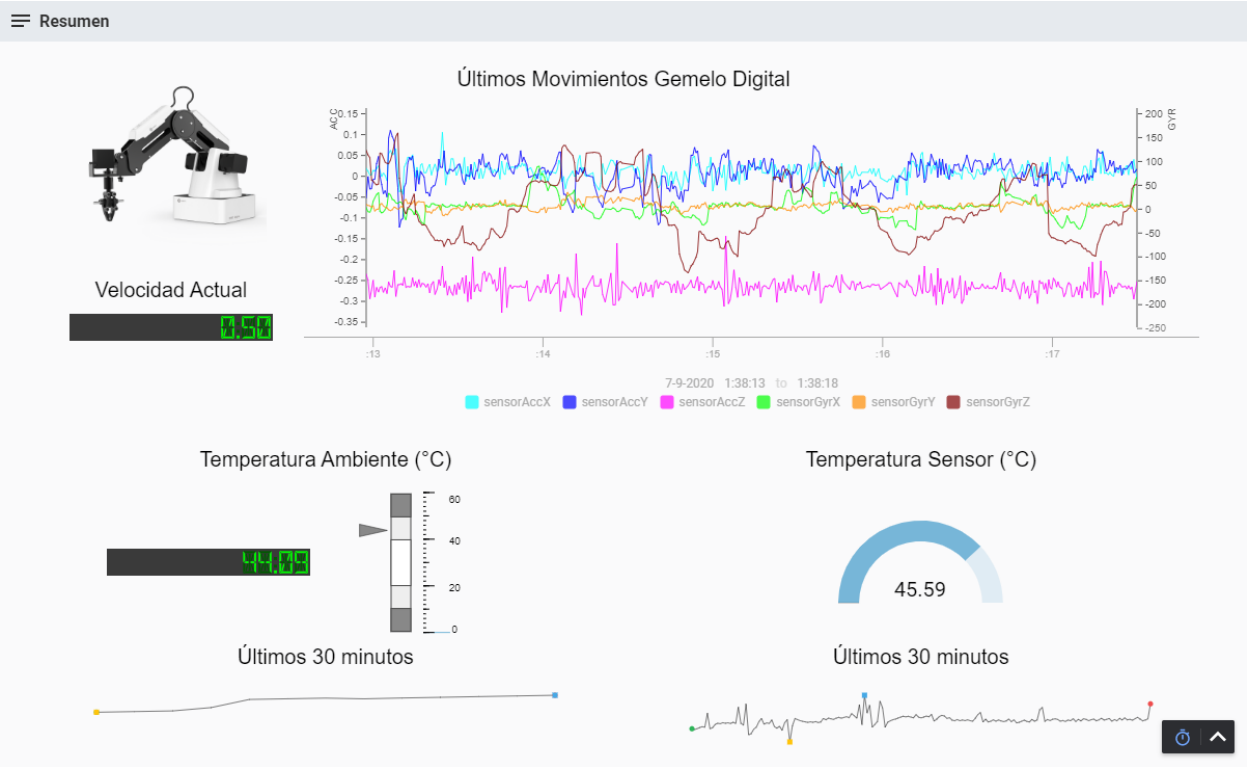


Figura 4-10. Pantalla Resumen

Como se mencionó en el Capítulo 3, *Ignition* tiene la ventaja de ser modular. *Ignition* tiene sus propios módulos y existen módulos de terceros que están disponibles en la página oficial de *Inductive Automation*, lo que garantiza su funcionamiento. Para lograr que el SCADA pueda

establecer comunicación con el *broker MQTT*, se emplearon los siguientes módulos desarrollados por *Cirrus Link Solutions*[16]:

- ***MQTT Engine Module**:* establece la comunicación bidireccional entre el sistema y el *broker MQTT*. En el presente proyecto se trabaja con el *broker mosquitto*. Cirrus Link tiene su propio módulo para crear un servidor *MQTT*. *(MQTT Distributor Module)*. En la Figura *4-11* y Figura *4-12*, se muestra la conexión entre Ignition y broker MQTT y la configuración de los tópicos respectivamente. El proceso para establecer la conexión y crear tópicos se los explicará en el Apéndice C.

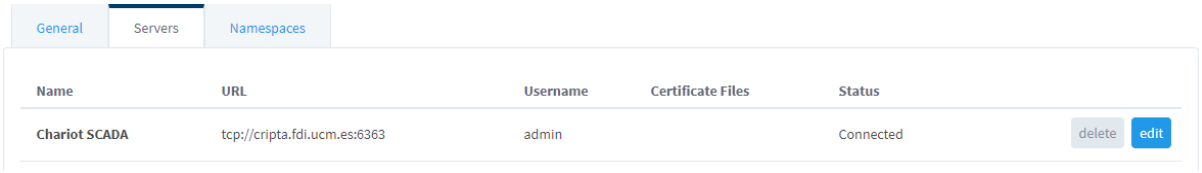


Figura 4-11. Estado Conexión *Ignition* y *broker MQTT*

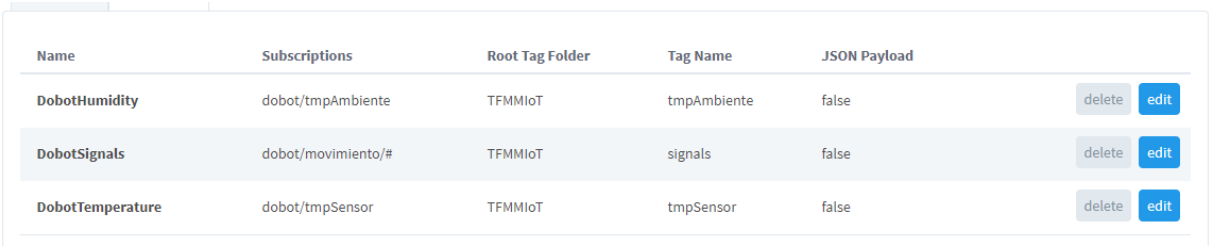


Figura 4-12. Configuración de los tópicos *MQTT* a escucha desde *Ignition*

- ***MQTT Transmission Module:*** permite la conexión con un broker MQTT y publicar mensajes en un tópico.

4.4.2 Pantalla Histórico Digital Twin

Para realizar las consultas de los diferentes movimientos y sensores del sistema Ignition debe comunicarse con MongoDB. *Ignition* a diferencia de otros sistemas SCADA, integra la posibilidad de conectarse con un sin número de las principales bases de datos SQL disponibles, entre ellas: *MySQL*, *SQLServer*, MariaDB, *PostgresSQL*, etc, sin embargo, no posee una conexión directa con

[16] https://www.cirrus-link.com/mqtt-software-for-iiot-scada/#

bases NoSQL. Las bases de datos NoSQL, en los últimos años han venido tomando fuerza, en diferentes aplicaciones, pero no en entornos industriales.

Ignition, permite realizar programación empleando lenguaje *Jython 2.7.0*[17], que básicamente es una implementación de *Python* con *Java*. Como se mencionó anteriormente se empleó el módulo *pymongo* para interactuar con MongoDB instalándolo directamente por consola con el comando *pip*[18]. *Ignition* no emplea las librerías que se encuentran directamente en el sistema operativo y no tiene un módulo directo para interactuar con MongoDB. La solución fue copiar todas las carpetas y dependencias que se crean al momento de ejecutar la instalación en el sistema con el comando *pip*. Cabe recalcar que se trabaja con la versión *Jython 2.7.0*, por lo que los módulos y carpetas que se pueden emplear son de *Python 2.7* y no de *Python 3.XX*, es sumamente importante hacer hincapié en este aspecto ya que si se instancia la nueva versión no va a funcionar correctamente. Una vez copiados todos los archivos en el entorno de *Ignition*, se puede importar los módulos como se lo realizaría en cualquier programa de *Python.*

Figura 4-13. Pantalla Históricos Digital Twin No.1

[17] https://www.jython.org/

[18] Comando para instalar paquetes de Python. https://pypi.org/project/pip/

Existen dos pestañas en la pantalla de históricos, pantalla de históricos generales Figura 4-13 y los históricos específicos de cada uno de los movimientos Figura 4-15. En la pantalla de la Figura 4-13, se realiza una consulta de los movimientos que existen en un rango de fechas seleccionado. Se puede realizar filtro por el tipo de movimiento realizado: movimiento libre, secuencia 1, secuencia 2 y secuencia 3. De la misma forma se puede realizar consultas del resto de sensores disponible en un gráfico diferente.

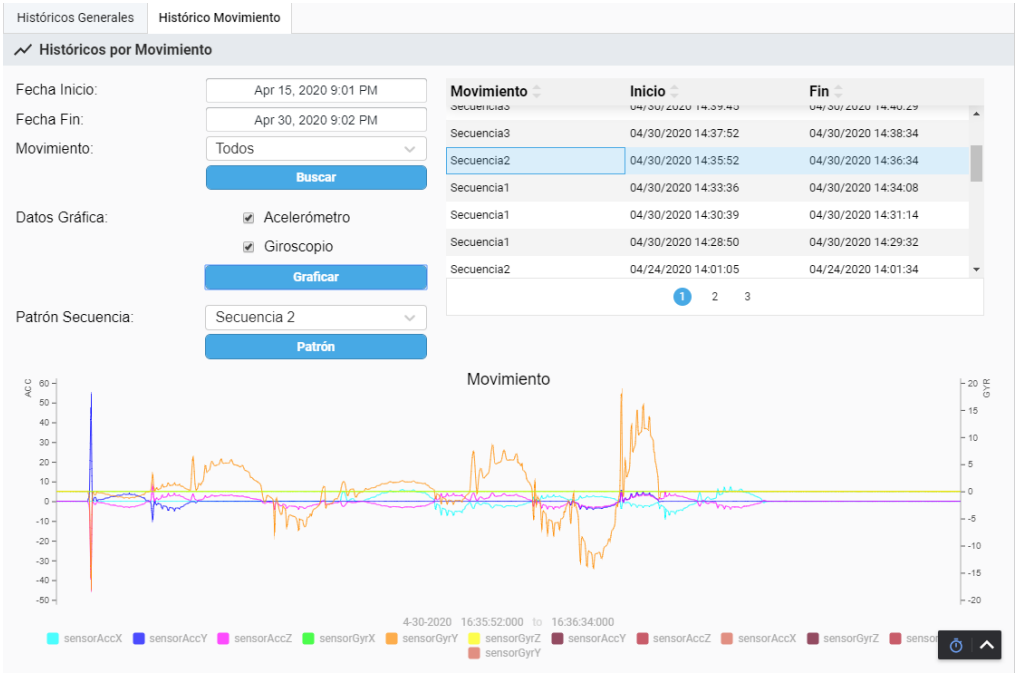


Figura 4-14. *Pantalla Históricos Digital Twin No.2*

La segunda vista de históricos de la Figura 4-15, despliega todas las secuencias de movimientos que se han producido en un rango de fechas seleccionado, de la misma forma se puede filtrar por el tipo de movimiento. La diferencia está en que se puede visualizar en detalle el movimiento, el momento exacto de inicio y fin. Con estos datos se selecciona un periodo en específico y se lo agrega al gráfico, incluyendo la posibilidad de realizar un contraste con los datos del patrón de la señal. Es decir, se puede realizar una comparativa del movimiento seleccionado con las aceleraciones nominales del sistema.

4.4.3 Pantalla Alarmas

En cada uno de los movimientos realizados se aplica el criterio explicado en el punto 4.3.6 y se almacenan los datos en la base de datos y se envían por *MQTT* para ser monitoreados en el sistema. La mayor parte de software SCADA permite la configuración de alarmas de variables del sistema, en este caso se configuró que si el coeficiente de correlación es inferior al umbral configurado se produzca una alarma. Esta alarma se la puede ver en la barra superior del *dashboard* Figura *4-16*, al hacer clic en ella se muestra un resumen de todas las alarmas activas o no reconocidas. En la Figura *4-16*, se muestra el *banner* de alarmas propio de *Ignition*, con esta herramienta se puede realizar el manejo de todas las alarmas del sistema como su reconocimiento, ignorar, visualización, etc.

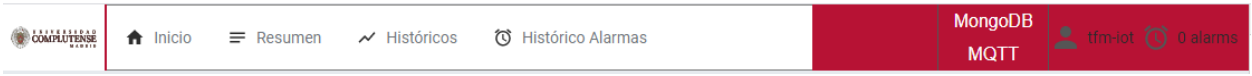


Figura 4-15. Barra superior del *dashboard*

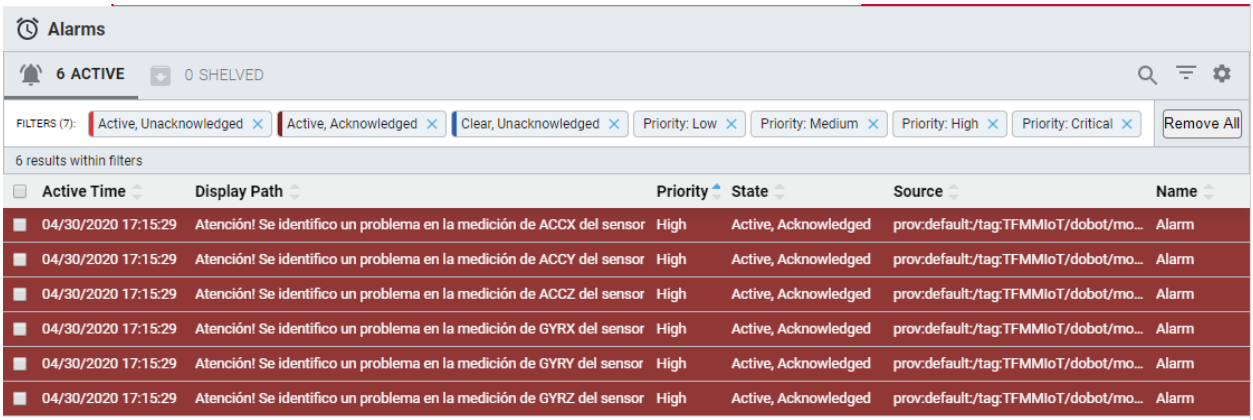


Figura 4-16. Banner de alarmas propio de *Ignition*

Para el manejo del registro de alarmas y verificar el momento en que se detectó algún comportamiento anormal, se creó otra pantalla que se muestra en la Figura 4-17, donde se puede realizar consultas dentro del rango de fechas seleccionado y verificar lo sucedido. De la misma forma se realiza la consulta, se selecciona el dato de interés y se puede comparar con el patrón de la señal.

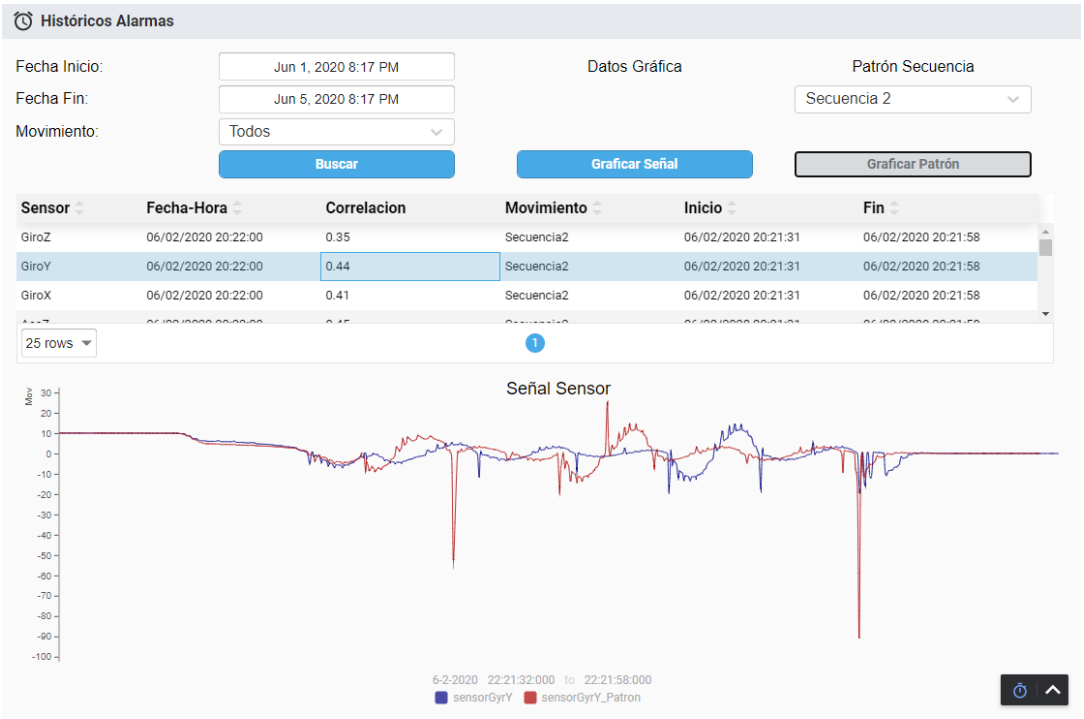


Figura 4-17. Pantalla Histórico Alarmas

Como en cualquier sistema es necesario definir los usuarios que pueden acceder a la aplicación. Ignition integra la capacidad de crear roles y el número de usuarios que sea necesario. Para el presente trabajo se crearon 3 roles como se muestra en la Tabla 4-4: administrador, operador y espectador. El administrador es el que mayor privilegio tiene, puede modificar la aplicación y acceder a todas las pantallas de la aplicación. Operador puede moverse por toda la aplicación, pero no puede editar ninguna pantalla y por último el espectador únicamente podrá ver la pantalla de resumen y consultar los movimientos realizados.

Tabla 4-4. Roles creados en la aplicación

Rol	**Detalles**
Espectador	Solo puede visualizar y realizar consultas básicas de los movimientos realizados. Acceso a la pantalla resumen, históricos por movimiento y resumen de alarmas
Operador	Privilegios de rol espectador y consulta detallada de los movimientos y reconocimiento de alarmas. Acceso a pantallas de espectador, histórico general y banner de alarmas.
Administrador	Privilegios de rol operador, control total de la aplicación y acceso para realizar cambios en el diseño de las pantallas.

Mientras el dashboard está funcionando es necesario informar al usuario el estado de las conexiones establecidas tanto para base de datos *MongoDB* como para el *broker MQTT*, para ello se han creado indicadores que monitorean estas conexiones. En el caso de que se pierda el enlace o no esté disponible el servidor de cualquiera de ellas, saltará una alarma en la aplicación y el indicador de conexión cambiará de color, como se muestra en la Figura 4-18.

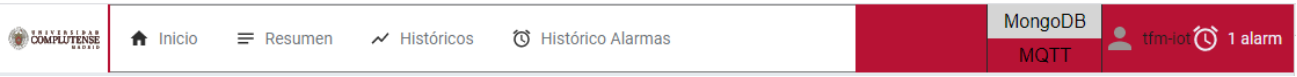


Figura 4-18. Barra superior, indicadores estado conexiones *MongoDB y Broker MQTT*

Capítulo 5 - Pruebas y análisis de Resultados

En el presente capítulo se explicarán a detalle las pruebas que se realizaron para verificar el correcto funcionamiento del sistema. Por un lado, se analizarán los datos recogidos en la simulación de Matlab, si bien es un entorno virtual, pero nos sirve para comprobar la implementación del sistema. Verificados los resultados se aplicará al *Dobot Magician* y se analizarán los resultados. Se mostrarán las gráficas y los datos más relevantes que se sacaron de la infraestructura.

Los códigos implementados se encuentran en un repositorio de *GitHub*, en el Apéndice A se indica la dirección del repositorio y su contenido.

5.1 Mecanismo de pruebas

Las pruebas se las llevará a cabo en dos entornos que se explicaron en el Capítulo 4: Simulación brazo robótico en Matlab y el *Dobot Magician.* En los dos entornos se definirán tres secuencias de movimiento, se realizará la captura de datos y almacenamiento en *MongoDB.* Una vez almacenados los datos patrones del correcto funcionamiento de cada una de las secuencias se aplicará fuerzas externas o variaciones en las secuencias para verificar que se logra detectar un comportamiento anormal.

Al momento de desarrollar las secuencias se tendrá un identificador de movimiento y una bandera para indicar que se está entrenando el sistema, es decir almacenando el patrón del movimiento.

5.2 Simulación Brazo Robótico Matlab

Como se explicó anteriormente, Matlab nos permite exponer al sistema a diferentes situaciones para verificar el desempeño del sistema. En este caso vamos a variar los parámetros de configuración de velocidad, aceleración y agregar una fuerza externa a los distintos tipos de movimientos configurados, con ello podremos comprobar que el sistema esté funcionando correctamente. En la primera parte se va a enviar los movimientos en condiciones normales de las tres secuencias que se tienen preconfiguradas. Para ello se creó una función, donde se debe especificar la secuencia del movimiento, tipo de movimiento y la fuerza externa que se va a aplicar.

- TFMMIoTMove(*fnMovCode*, *fnMovCodeTrain*, *wrench*)
- *fnMovCode:* secuencia de movimiento. 1, 2 o 3.
- *fnMovCodeTrain:* tipo de movimiento. 0 = normal, 1 = *train*, *2 = predML y 3 = trainML*.
- *wrench:* vector de seis elementos, especificando las fuerzas a aplicar**.**

Dentro de la función se realizarán las asignaciones, cálculos previos y se inicia la simulación. En este punto se tiene que especificar que los parámetros normales de funcionamiento son velocidad 0.5, aceleración 0 y ninguna fuerza externa. Los movimientos de los últimos 5 segundos del *Digital Twin* se los puede ver en tiempo real, en la pantalla Resumen del *dashboard* creado como se muestra en la Figura *5-1*, con esto podemos verificar que la comunicación *MQTT* entre el simulador del *Digital Twin* y el servidor está funcionando.

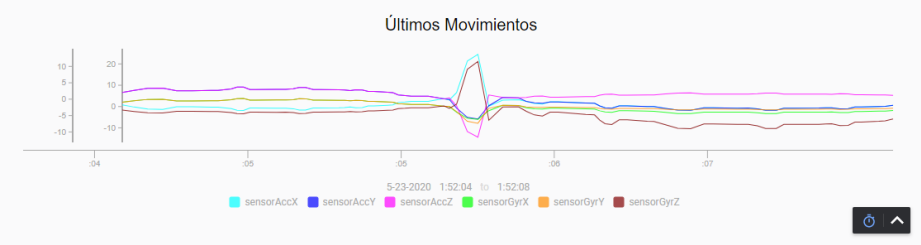


Figura 5-1. Movimientos en tiempo real, pantalla Resumen

Por otro lado, Figura 5-2 se puede verificar que los datos del movimiento se están guardando en la base de datos, se puede realizar la consulta y se deben desplegar en la tabla. Con el identificador del movimiento y las fechas de inicio y de fin.

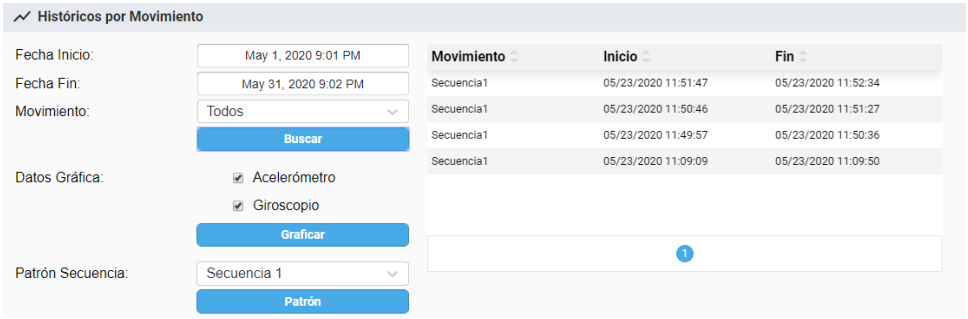


Figura 5-2. Consulta de históricos de movimiento

5.2.1 Algoritmo de detección de anomalías

Para probar el algoritmo de verificación de comportamientos anormales, se debe cambiar el *movCodeTrain* de la función a 0, para poner a funcionar el algoritmo. La primera prueba que se va a realizar es un movimiento en condiciones normales de una de las secuencias, posterior a esto se van aplicando variantes en velocidad, aceleración y fuerzas para verificar los resultados.

Tabla 5-1. Parámetros Prueba No. 1

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
1	Secuencia 1	0.5	0	[0 0 0 0 0 0]

Al terminar el movimiento se aplica el algoritmo de detección de anomalías, como se esperaba en este caso la señal patrón y la señal actual son idénticas por lo que el coeficiente de correlación es 1 y no genera ninguna alarma. Los resultados se muestran en la Figura *5-3* y Figura 5-4.

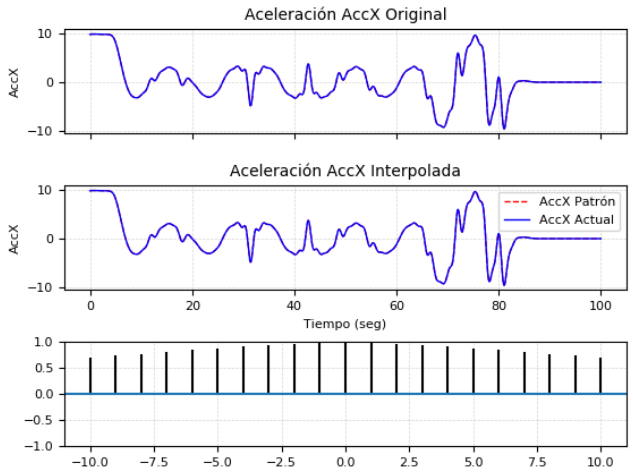


Figura 5-3. Gráfica AccX Señal Original vs Señal Actual, Prueba No.1

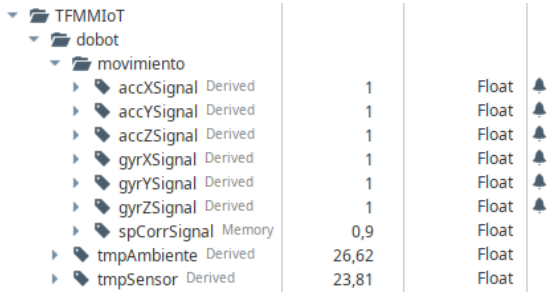


Figura 5-4. Valores Coeficiente correlación Prueba No.1

Tabla 5-2. Parámetros Prueba No. 2

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
2	Secuencia 1	0.52	0	[0 0 0 0 0 0]

En este caso al aplicar una velocidad ligeramente mayor a la de la señal original se puede ver pequeñas variaciones en la señal final, pero no existe tanta diferencia. Es este caso no salta ninguna de las alarmas ya que los valores de correlación siguen siendo altos. En la Figura 5-5 y Figura 5-6, se pueden ver las gráficas y los coeficientes de correlación respectivamente.

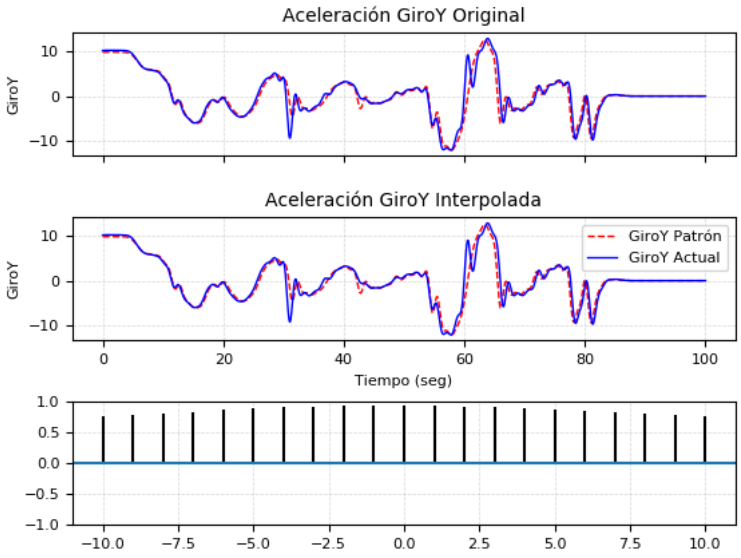


Figura 5-5. Gráfica GiroY Señal Original vs Señal Actual, Prueba No.2

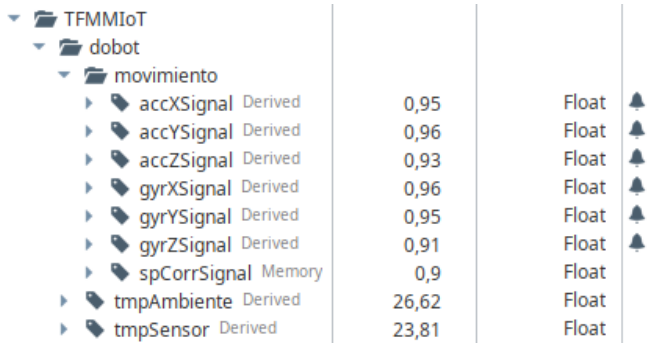


Figura 5-6. Valores Coeficiente correlación Prueba No.2

Tabla 5-3. Parámetros Prueba No. 3

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
3	Secuencia 1	0.54	0	[0 0 0 0 0 0]

Ahora se aumenta un poco la velocidad y se puede apreciar variaciones más acentuadas en los seis ejes. Con esto se puede verificar en la Figura 5-7 que existen cambios bruscos que hacen que la señal ya no tenga una correlación alta con el patrón. De la misma forma, en la Figura 5-8 se verifican el valor del coeficiente de correlación.

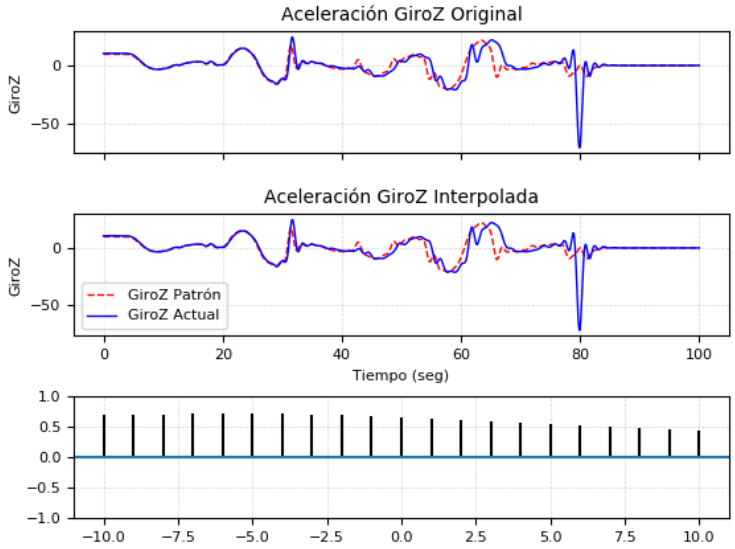


Figura 5-7. Gráfica GiroZ Señal Original vs Señal Actual, Prueba No.3

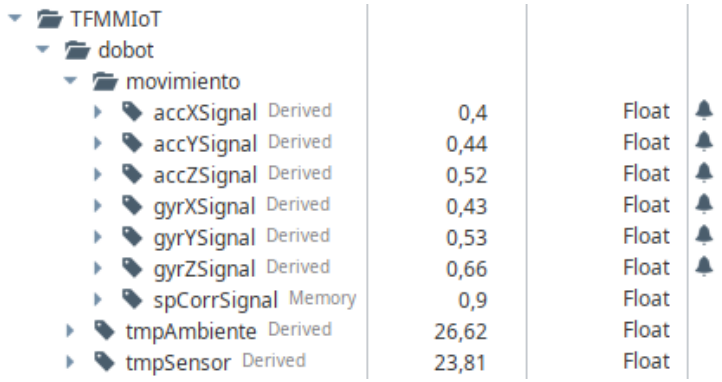


Figura 5-8. Valores Coeficiente correlación Prueba No.3

Por otro lado, cuando el sistema detecta una alarma se visualiza en la barra superior de la aplicación como se muestra en la Figura 5-9, indicando con un color blanco que existen alertas y el número de alarmas actual. Al dar clic sobre el número de alarmas, nos dirigimos a la pantalla del banner de alarmas de la Figura 5-10 donde se ven las alarmas activas no reconocidas, activas reconocidas y limpias no reconocidas, de colores rojo fuerte, rojo oscuro y azul respectivamente.

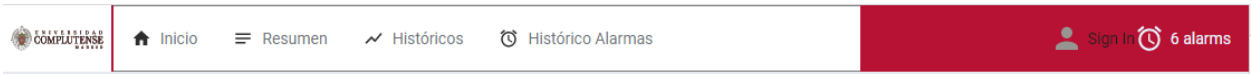


Figura 5-9. Barra superior de la aplicación, Prueba No.2

Active Time	Display Path	Priority	State	Source	Name
05/25/2020 16:45:06	Atención! Se identifico un problema...	High	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/...	Alarm
05/25/2020 16:45:06	Atención! Se identifico un problema...	High	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/...	Alarm
05/25/2020 16:45:06	Atención! Se identifico un problema...	High	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/...	Alarm
05/25/2020 16:45:06	Atención! Se identifico un problema...	High	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/...	Alarm
05/25/2020 16:45:06	Atención! Se identifico un problema ...	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/...	Alarm
05/25/2020 16:45:06	Atención! Se identifico un problema ...	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/...	Alarm

Figura 5-10. Banner de alarmas activas reconocidas y activas no reconocidas

En la pantalla de resumen de alarmas, se puede realizar la consulta por rango de fechas y graficar la señal para observar los cambios existentes en las señales. En la Figura 5-11, se muestra las alarmas guardadas en la base de datos.

Figura 5-11. Consulta de los valores de alarma y datos del movimiento

Una vez realizada la consulta se puede contrastar la señal con el patrón de la secuencia actual. En la Figura 5-12 se muestran las señales del giroscopio en los tres ejes y en la Figura 5-13 las del acelerómetro en los tres ejes. Con esto se puede ver claramente que existen variaciones para poder tomar decisiones con respecto a un mantenimiento en alguna de lar articulaciones y sustituir alguna de las piezas.

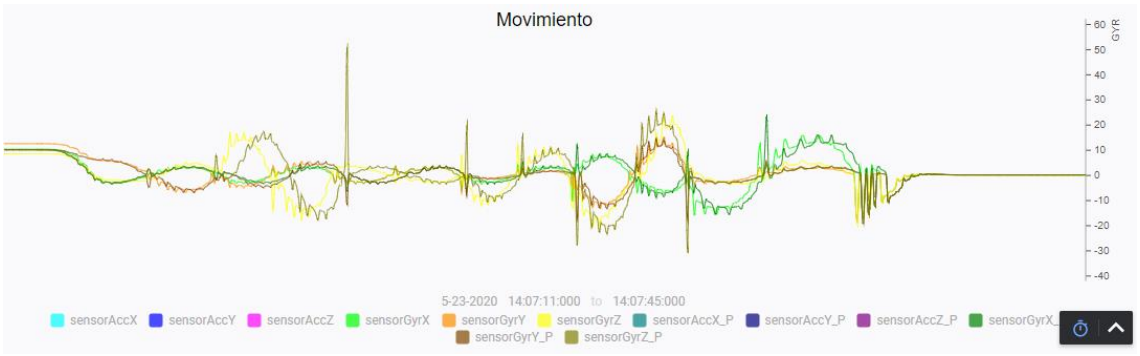


Figura 5-12. Gráfica Giroscopio Patrón y Actual Prueba No.3

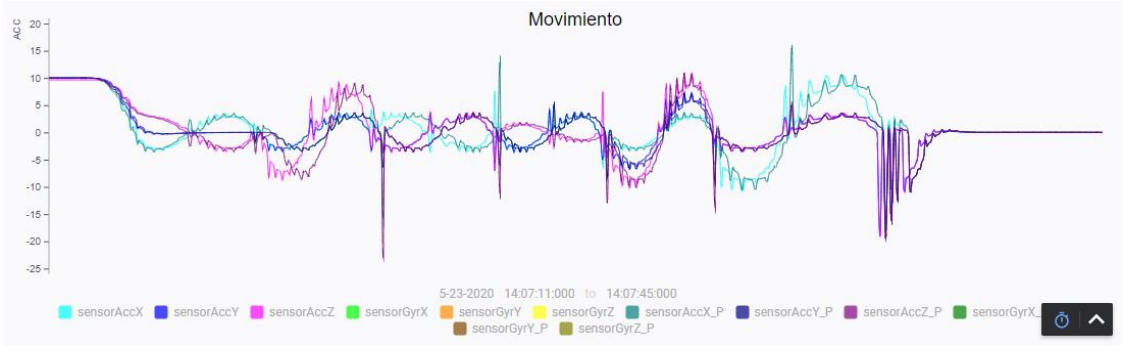


Figura 5-13. Gráfica Acelerómetro Patrón y Actual Prueba No.3

Tabla 5-4. Parámetros Prueba No. 4

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
4	Secuencia 1	0.50	0	[0 0 0.01 0 0 0]

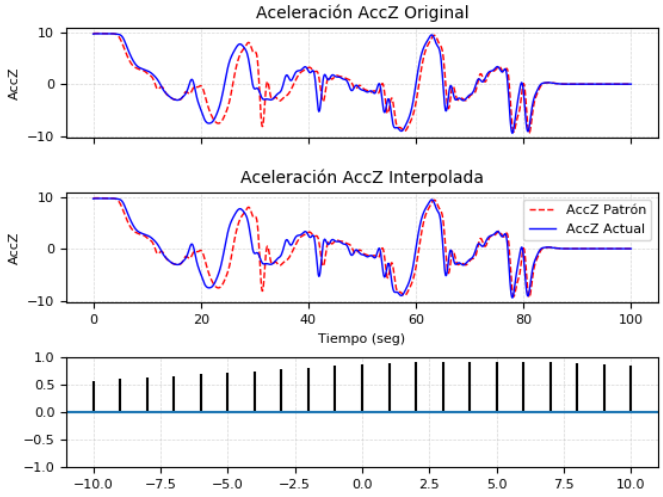


Figura 5-14. Gráfica AccX Señal Original vs Señal Actual, Prueba No.4

En las primeras pruebas se pudo apreciar que el algoritmo es capaz de detectar movimientos anormales causados por la velocidad, ahora se va a aplicar una fuerza externa para ver cómo responde. Aplicando una única fuerza, el resultado en la aceleración del eje Z se lo aprecia en la Figura 5-14, donde se ve una variación significativa en la señal y como resultado dos componentes generan una alarma aceleración y rotación en el eje Z como se muestra en la Figura 5-15.

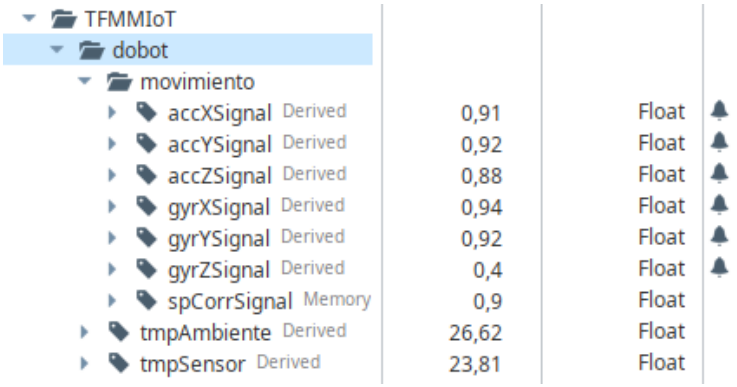


Figura 5-15. Valores Coeficiente correlación Prueba No.4

De la misma forma en el banner de alarmas se puede verificar la existencia de las dos alarmas generadas con el estado "*Active Unacknowledge*" color rojo y las alarmas que ya no están activas como "*Cleared Unacknowledge*" de color azul que una vez que se las reconozca desaparecerán de esta pantalla.

Active Time	Display Path	Priority	State	Source	Nan
05/25/2020 16:54:44	Atención! Se identifico un problema en la medición de GYRZ del sensor	High	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:54:44	Atención! Se identifico un problema en la medición de ACCZ del sensor	High	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:45:06	Atención! Se identifico un problema en la medición de ACCX del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:54:44	Atención! Se identifico un problema en la medición de GYRY del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:54:44	Atención! Se identifico un problema en la medición de GYRX del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:54:44	Atención! Se identifico un problema en la medición de ACCY del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:54:44	Atención! Se identifico un problema en la medición de ACCX del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:45:06	Atención! Se identifico un problema en la medición de GYRZ del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:45:06	Atención! Se identifico un problema en la medición de ACCZ del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar
05/25/2020 16:45:06	Atención! Se identifico un problema en la medición de ACCY del sensor	High	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/mo...	Alar

Figura 5-16. Banner de alarmas activas y limpias no reconocidas

Tabla 5-5. Parámetros Prueba No. 5

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
5	Secuencia 2	0.50	0	[0 0 0.01 0 0 0]

El algoritmo debe funcionar para cualquier tipo de movimiento, por lo que la siguiente prueba se la va a aplicar con la secuencia 2, se aplica una única fuerza externa, velocidad y aceleración en condiciones normales. El resultado se muestra en la Figura 5-17 que muestra que la aceleración en el eje Y tiene una desviación significativa del valor normal. En la Figura 5-18 se muestran los coeficientes de correlación confirmando que existen dos componentes fuera de rango aceptable.

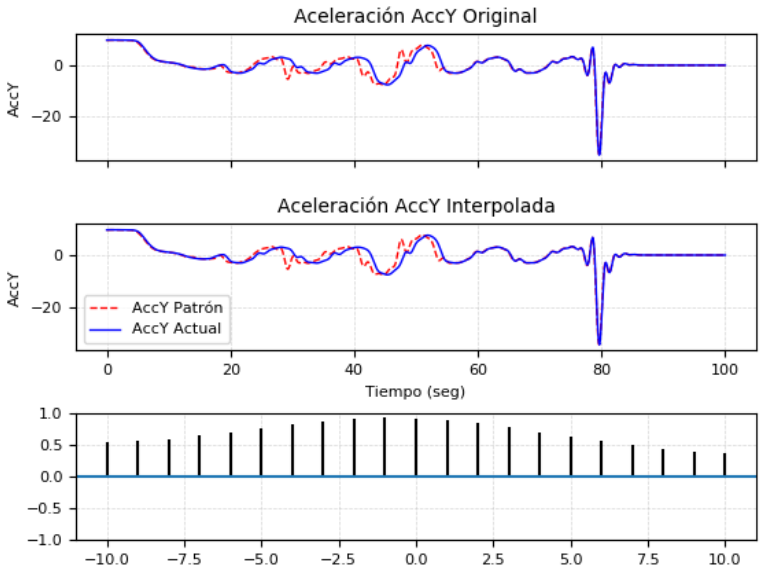


Figura 5-17. Gráfica AccY Señal Original vs Señal Actual, Prueba No.5

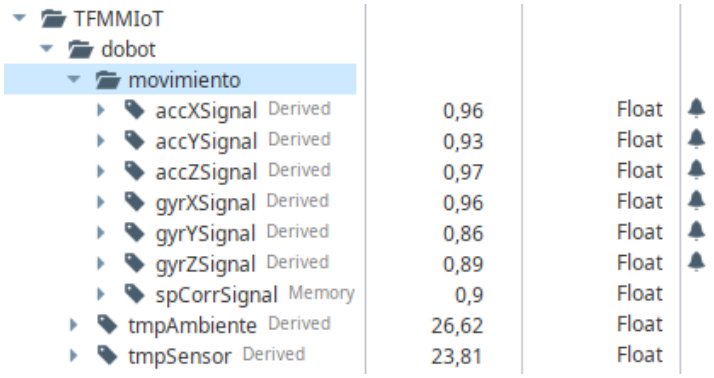


Figura 5-18. Valores Coeficiente correlación Prueba No.5

En la Figura 5-19, se muestra las alertas de las dos alarmas que se han activado después del movimiento.

Figura 5-19. Banner superior de la aplicación, Prueba No. 5

Tabla 5-6. Parámetros Prueba No. 6

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
6	Secuencia 3	0.50	0	[0 0.01 0 0.01 0 0]

En las pruebas con la secuencia 3 se aumentó una fuerza con velocidad y aceleraciones en condiciones normales. Los resultados fueron los esperados, se ve una señal más lejana del comportamiento normal. En la Figura 5-20 se muestra la desviación de la señal actual en la rotación del eje Y y en la Figura 5-21 se muestran que los coeficientes de correlación son menores a 0.90, lo que genera las alarmas.

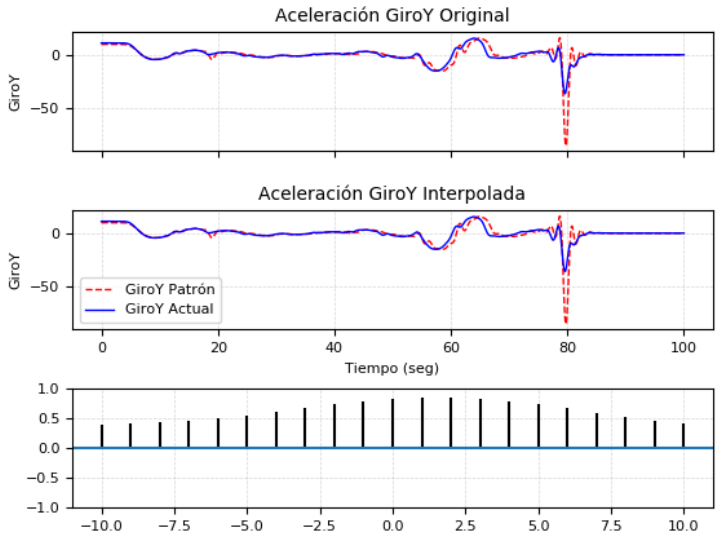


Figura 5-20. Gráfica GiroY Señal Original vs Señal Actual, Prueba No.6

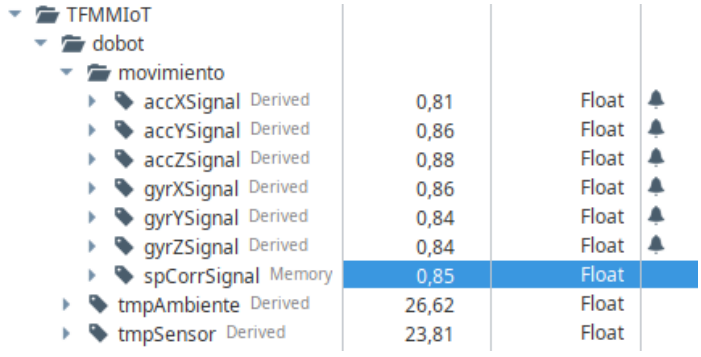


Figura 5-21. Valores Coeficiente correlación Prueba No.6

Tabla 5-7. Parámetros Prueba No. 7

No. Prueba	Movimiento	Velocidad	Aceleración	Fuerza
7	Secuencia 3	0.50	0.1	[0 0 0 0 0 0]

En la última prueba se aplicó una variación en la aceleración de 0.1, velocidad y fuerzas en condiciones normales. En la Figura 5-22 se comprueba que al registrar una aceleración diferente en la señal existe una desviación con el patrón, por su parte los coeficientes de correlación de Figura 5-23 son mucho menores a 0.5 lo que efectivamente existen movimientos anormales.

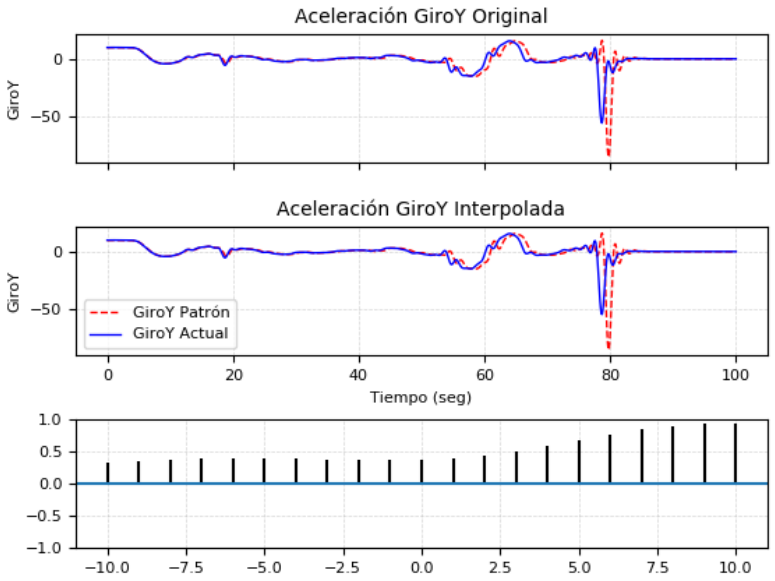


Figura 5-22. Gráfica GiroY Señal Original vs Señal Actual, Prueba No.7

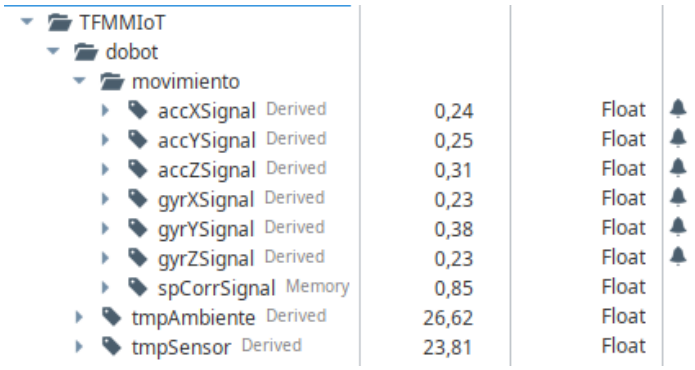


Figura 5-23. Valores Coeficiente correlación Prueba No.7

Finalmente se verifica que existen las 6 alarmas activas en el sistema, como se muestra en la Figura 5-24.

Figura 5-24. Banner de alarmas de aplicación, Prueba No. 7

5.2.2 Inteligencia Artificial aplicada al *Digital Twin*

Para realizar los modelos de predicción se van a emplear dos variables la velocidad y una fuerza externa. Para cada una de las variables se aplicaron dos técnicas: regresión lineal y redes neuronales. Por un lado, el conjunto de entrenamiento de la velocidad consta de 800 secuencias de movimientos que se recolectaron con ayuda del simulador *Matlab*, existen valores de velocidad que están en el rango $0 < x < 1$, ya que el movimiento en condiciones normales se lo definió en 0.5. Por su parte, para entrenar la fuerza externa se emplearon 200 movimientos con valores de fuerza externa de $0 < x < 0.5$.

Se empleó una red neuronal profunda, que no es más que una red neuronal multicapa. La estructura final de la red neuronal se muestra en la Figura 5-25: una capa de entrada de la dimensión del número de datos adquiridos, cuatro capas ocultas con 200, 100, 64 y 32 nodos respectivamente y la capa de salida que es el valor para predecir.

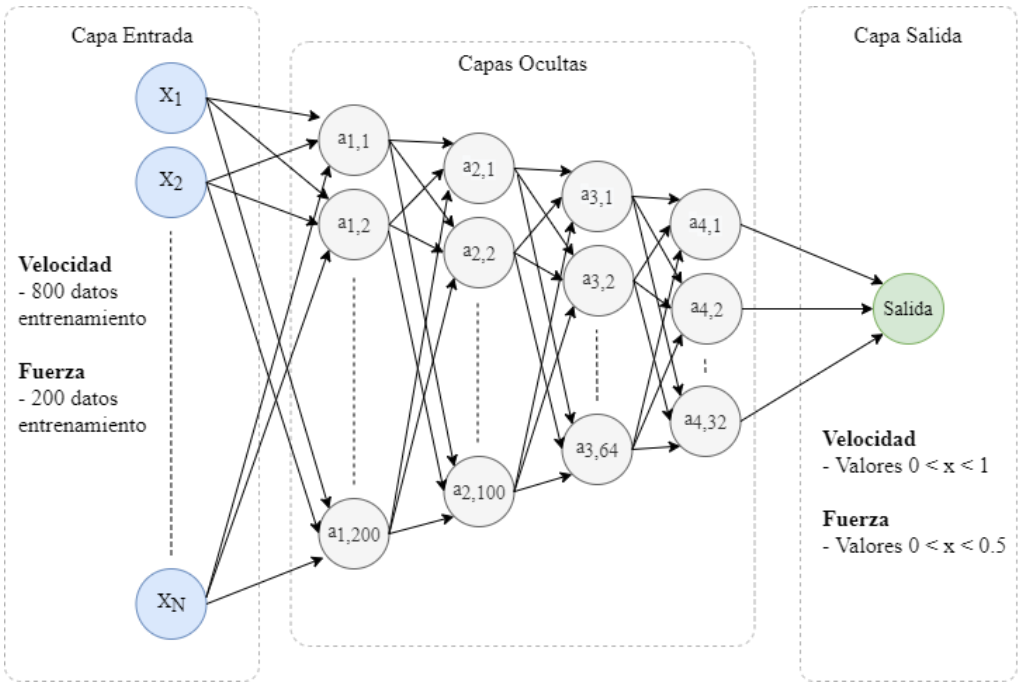


Figura 5-25. Estructura Red Neuronal Creada

Una vez entrenado el sistema, *keras* nos permite revisar el resumen del modelo creado, en la Figura 5-26 se muestran los resultados.

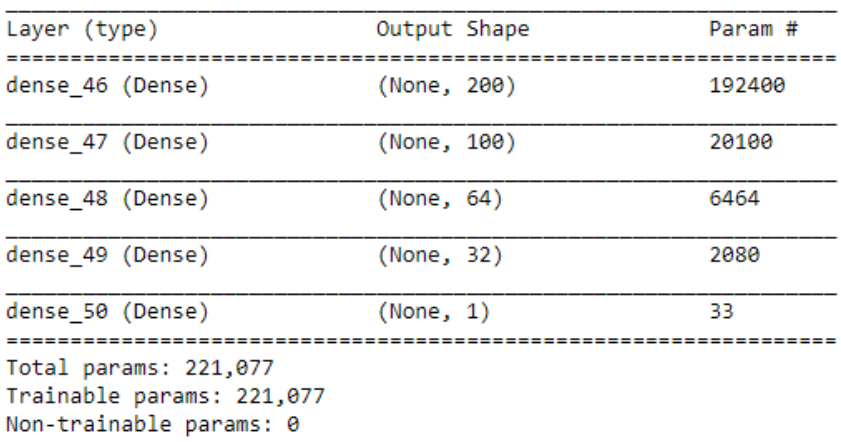


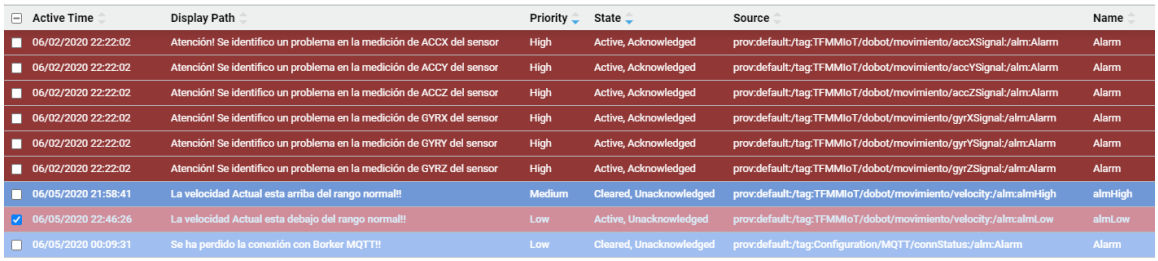
Figura 5-26. Resumen Red Neuronal Creada con Keras

Para realizar las pruebas se generaron varios movimientos con diferentes velocidades y fuerzas. Para la visualización del valor de la velocidad se agregó un indicador en la pantalla Resumen. De la misma forma, se establecieron los rangos de tolerancia gracias a las pruebas que se realizaron con el factor de correlación. El límite inferior de la velocidad en 0.45 y el límite superior en 0.55. Se realizaron 5 pruebas que se muestran en la Tabla 5-8.

Tabla 5-8. Parámetros Pruebas Variación Velocidad

No. Prueba	**Velocidad**	**Predicción Velocidad**	**Fuerza**
1	0.50	0.4999	0
2	0.55	0.5478	0
3	0.43	0.4307	0
4	0.68	0.6844	0
5	0.73	0.7304	0

Cuando el algoritmo detectó un error fuera de los límites configurados, se activó la alarma y se mostró el error en la pantalla del resumen de alarmas. Con esto se muestra el correcto funcionamiento de los modelos creados.



Active Time	Display Path	Priority	State	Source	Name
06/02/2020 22:22:02	Atención! Se identifico un problema en la medición de ACCX del sensor	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/accXSignal:/alm:Alarm	Alarm
06/02/2020 22:22:02	Atención! Se identifico un problema en la medición de ACCY del sensor	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/accYSignal:/alm:Alarm	Alarm
06/02/2020 22:22:02	Atención! Se identifico un problema en la medición de ACCZ del sensor	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/accZSignal:/alm:Alarm	Alarm
06/02/2020 22:22:02	Atención! Se identifico un problema en la medición de GYRX del sensor	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/gyrXSignal:/alm:Alarm	Alarm
06/02/2020 22:22:02	Atención! Se identifico un problema en la medición de GYRY del sensor	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/gyrYSignal:/alm:Alarm	Alarm
06/02/2020 22:22:02	Atención! Se identifico un problema en la medición de GYRZ del sensor	High	Active, Acknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/gyrZSignal:/alm:Alarm	Alarm
06/05/2020 21:58:41	La velocidad Actual esta arriba del rango normal!!	Medium	Cleared, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/velocity:/alm:almHigh	almHigh
06/05/2020 22:46:26	La velocidad Actual esta debajo del rango normal!!	Low	Active, Unacknowledged	prov:default:/tag:TFMMIoT/dobot/movimiento/velocity:/alm:almLow	almLow
06/05/2020 00:09:31	Se ha perdido la conexión con Borker MQTT!!	Low	Cleared, Unacknowledged	prov:default:/tag:Configuration/MQTT/connStatus:/alm:Alarm	Alarm

Figura 5-27. Banner de Alarmas pruebas Velocidad

Por otro lado, se hicieron las pruebas aplicando una fuerza externa. El límite configurado para activar la alarma fue de 0.1. El desempeño del algoritmo se muestra en la Tabla 5-9. En comparación con el modelo de velocidad conforme aumenta el valor de la fuerza el error aumenta, esto se debe a que el modelo creado para la fuerza externa tiene un menor número de elementos de entrenamiento.

Tabla 5-9. Parámetros Pruebas Variación Fuerza Externa

No. Prueba	**Velocidad**	**Fuerza**	**Predicción Fuerza**
1	0.50	0	0.0002
2	0.50	0.1	0.1016
3	0.50	0.35	0.3504
4	0.50	0.27	0.2530
5	0.50	0.43	0.4015

Con las pruebas realizadas se demuestra que con la información almacenada del *Digital Twin* se pueden aplicar diferentes técnicas para evaluar el desempeño del robot y tomar decisiones de varias índoles.

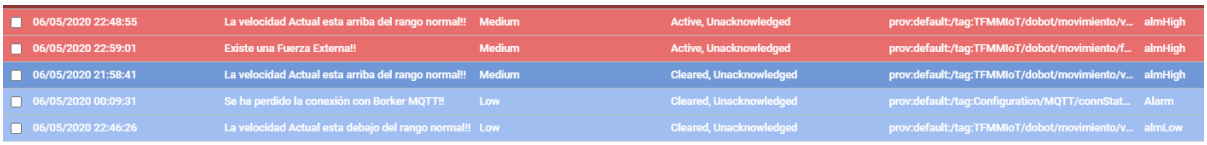


Figura 5-28. Banner de Alarmas pruebas Velocidad

5.3 *Dobot Magician*

Para realizar las pruebas del *Digital Twin* del brazo robótico se debe verificar el correcto funcionamiento del *SensorTag*, para esto se conectó el sensor los datos obtenidos se los puede monitorear en tiempo real en la pantalla de resumen como se muestra en la Figura 5-29. Se pueden ver los datos de aceleración y rotación en los tres ejes, de temperatura ambiente y temperatura del objeto. Con esto se confirma el correcto funcionamiento del sensor y los parámetros que se configuraron.

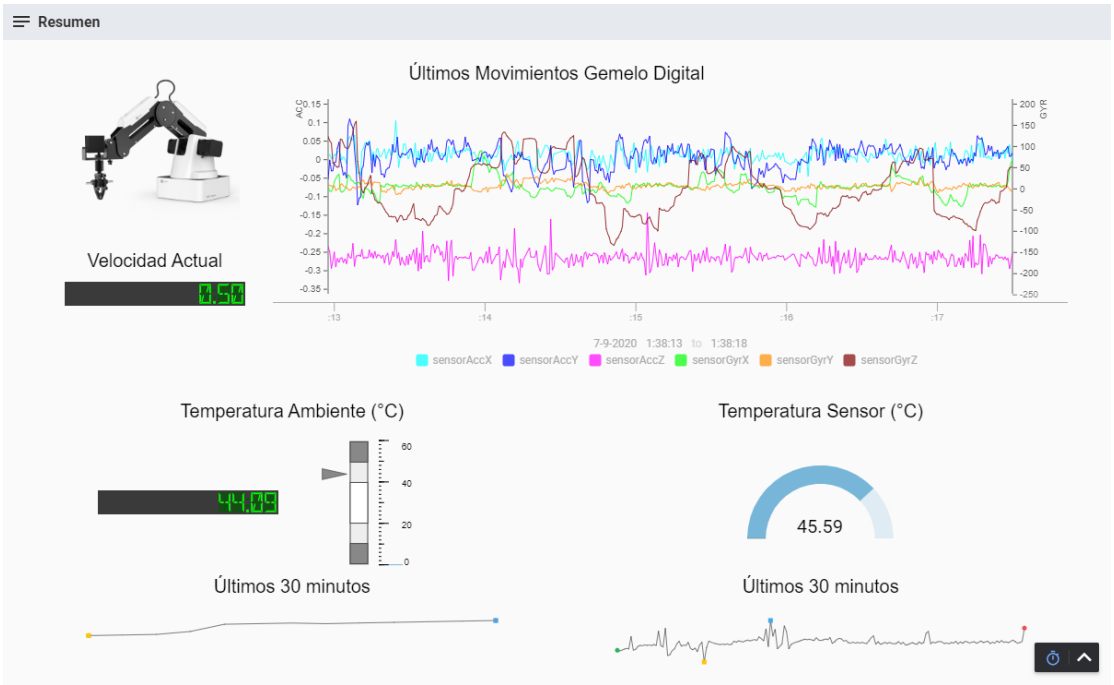


Figura 5-29. Pantalla Resumen *Digital Twin Dobot Magician*

Las pruebas de las secuencias de movimiento, registro de los datos y del algoritmo de detección de anomalías no se las pudo aplicar al *Dobot Magician* por la situación actual del virus COVID-19. Toda la infraestructura y la programación de las secuencias están listas para realizar las pruebas, pero por el estado de alarma decretado por la Pandemia COVID-19 se cerró el acceso a la universidad y es imposible acceder al robot.

5.4 Evaluación *Dashboard*

Para la evaluación de la interpretación y de usabilidad del dashboard se empleó la Puntuación Media de Opinión (MOS), que en la actualidad es uno de los indicadores más utilizados en varios sectores. En un principio fue pensado para aplicaciones de telecomunicaciones, específicamente para medir la calidad de sistemas de voz y evaluar el entendimiento que tienen los usuarios. Esta medida se basa en la retroalimentación del usuario después de emplear una herramienta, es una medida cuantificable.[42] La MOS es un número real, por lo general entre 1 y 5 siendo 1 la peor ponderación, es por ello por lo que por lo general se emplea la Clasificación Absoluta de Categorías (ACR) que son las escalas cuantificables que se emplean en las pruebas de calidad. En la Tabla 5-10, se muestran los niveles de escala detallados.

Tabla 5-10. Niveles Escala ACR

Ponderación	Detalle
5	Excelente
4	Buena
3	Justa
2	Pobre
1	Malo

Para desarrollar esta métrica se empleará una encuesta corta enfocada a los puntos relevantes de la aplicación. Un punto importante es el grupo al que va destinada la aplicación, esto se debe tomar en cuenta para sacar las conclusiones del desempeño del *dashboard*, es por lo que la primera pregunta de la encuesta será cualitativa ya que deseamos conocer la ocupación de la persona que está llenando el formulario. Con esto se plantearon las siguientes preguntas:

1. ¿Cuál es su profesión y/o sector de trabajo?
    a. Ingeniería Industrial
    b. Ingeniería Sistemas
    c. Mantenimiento
    d. Procesos

e. Otros

2. La interpretación de los datos de la pantalla de "Resumen", muestran claramente el estado actual del brazo robótico.
3. En la pantalla de "Históricos" la información es suficiente para tomar decisiones de funcionamiento del brazo robótico.
4. En la pantalla "Histórico Alarmas" se puede interpretar claramente los valores presentados.
5. Los colores y textos de las pantallas permiten la correcta interpretación, visualización y navegación en la aplicación.
6. La navegación entre pantallas del dashboard es intuitiva y natural.
7. Con los datos mostrados en las pantallas, estaría en la capacidad de tomar decisiones de mantenimiento del Gemelo Digital.

5.4.1 Resultados de la evaluación

La encuesta se la realizó mediante el uso de la herramienta de “Formularios” de Google, restringiendo a una respuesta por usuario. En total se recogieron 25 formularios y los resultados de cada pregunta del número 2 a la 7 se los muestra en la Tabla 5-11.

Tabla 5-11. Resultados encuesta Evaluación Dashboard

Número de Pregunta	Malo	Pobre	Justa	Buena	Excelente	Total
2	0	1	2	10	12	25
3	0	3	3	6	13	25
4	0	2	4	5	14	25
5	0	1	2	13	9	25
6	0	1	2	8	14	25
7	0	0	7	11	7	25

Por otro lado, el objetivo de la primera pregunta es saber el sector de trabajo o conocimiento que tienen las personas que llenaron el formulario. Los resultados se pueden ver en la Figura 5-30. En su gran mayoría con el 60% fueron personas dedicadas a la Automatización Industrial que es el sector al que va enfocada la solución.

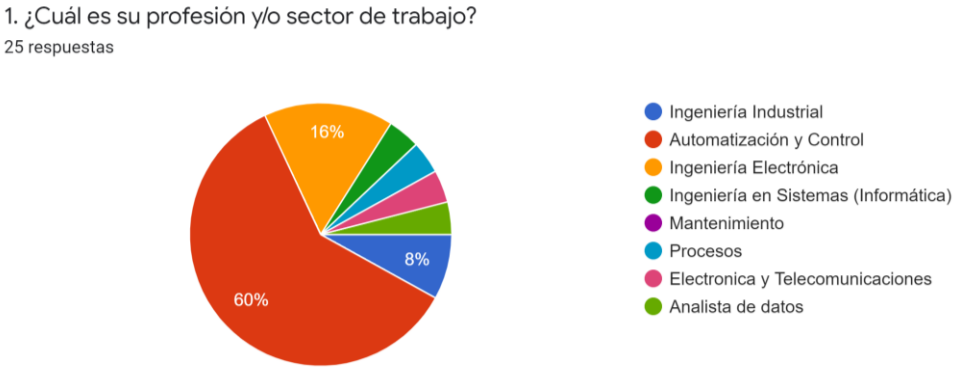


Figura 5-30. Resultados Pregunta No. 1

Las preguntas número 5 y 6 están enfocadas a la aceptación del diseño del *dashboard.* En la Figura 5-31, se muestran los resultados de la pregunta 5, más de la mitad de las personas logran visualizar de una forma adecuada los textos y gráficos de las pantallas. El 52% califica como bueno, el 36% excelente y tan solo un 4% que la forma de los gráficos es pobre.

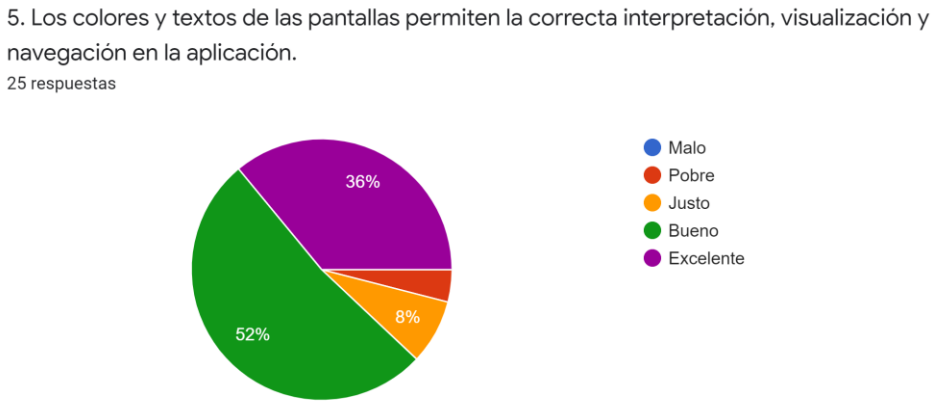


Figura 5-31. Resultados Pregunta No. 5

En la Figura 5-32, se muestran los resultados de la pregunta 6, la navegación entre pantallas tiene una aceptación del 90%. El 56% califica como excelente, el 32% excelente y tan solo un 4% que la navegación es pobre.

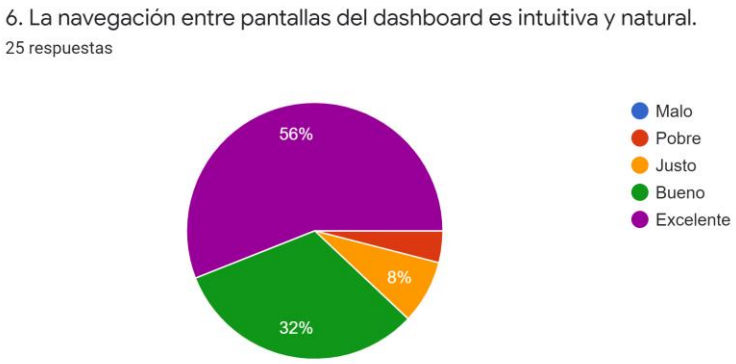


Figura 5-32. Resultados Pregunta No. 6

En la Figura 5-33Figura 5-32, se muestran los resultados de la pregunta 7, que valida la utilidad y en enfoque de la aplicación. El 44% considera como bueno, el 28% excelente y el 28 restante como justo. En este caso no se tiene ningún registro menor a 3. Lo que indica que se tiene una correcta interpretación de los datos para tomar decisiones en el proceso.

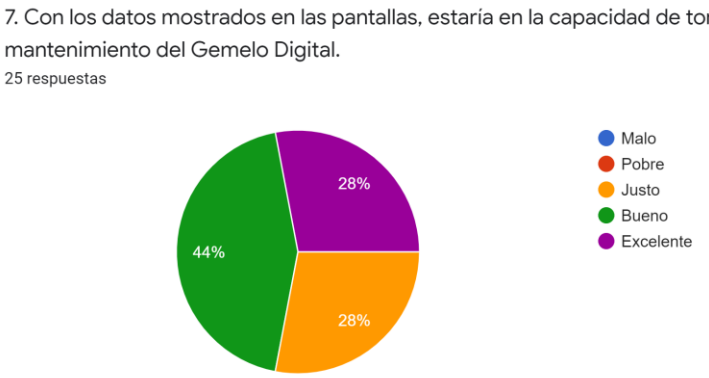


Figura 5-33. Resultados Pregunta No. 7

De los resultados obtenidos al realizar las encuestas se puede apreciar que un *dashboard* con la información necesaria es de gran ayuda para tomar decisiones de mantenimiento en el proceso. La información que se mostró en la aplicación desarrollada es suficiente y clara al momento de

tomar decisiones. Relacionando la primera pregunta y la última del formulario se determina que la solución desarrollada tiene un impacto positivo en el ámbito industrial y va a ser de gran ayuda para realizar una planificación de mantenimiento predictivo en las líneas de producción. Por otro lado, cabe destacar que la mayoría de las respuestas están entre los valores de Buena y Excelente lo que significa una aceptación del diseño y el objetivo del *dashboard.*

Capítulo 6 - Conclusiones y trabajo futuro

En el presente trabajo se ha explicado detalladamente la implementación de la infraestructura y las pruebas realizadas para alcanzar los objetivos. Este capítulo pretende sintetizar todos los logros obtenidos, los problemas encontrados y las posibles líneas de investigación que se pueden abarcar en este sector del IIoT. Esta información podrá ser tomada en cuenta para futuros desarrollos.

6.1 Conclusiones

A lo largo del trabajo se estudiaron las diferentes alternativas y despliegues existentes de *Digital Twin*, que fueron la base para el desarrollo de la infraestructura levantada. Se construyó un *Digital Twin* del comportamiento de un brazo robótico simulado y físico, almacenando la información del nodo sensor desplegado. De esta manera, se obtuvo un historial de los datos relevantes con el fin de tomar decisiones en su funcionamiento.

Por lo general los brazos robóticos tienen un elevado grado de exactitud y precisión por lo que medir la aceleración y rotación fue la opción ideal para crear el *Digital Twin*. Las características del entorno en el que se está trabajando influyen directamente en la vida útil del brazo robótico, por lo que son necesarias las mediciones y registros de temperatura, humedad, etc. Estos registros no son solo útiles para el cliente final, si no para la empresa que ha creado el brazo robótico ya que puede estudiar el desempeño del robot a diferentes condiciones.

Para la adquisición de datos es de suma importancia la correcta selección del Nodo Sensor con el que se va a trabajar. Se deben definir las medidas que se pretenden tomar y las características de trabajo como: resolución, frecuencia, etc. En la solución implementada el *SensorTag* cumplió con el objetivo de emplear un solo tipo de Nodo genérico y versátil que se pueda adaptar a cualquier brazo robótico, ya que un solo dispositivo permite la medición de varias magnitudes que aportan información relevante del *Digital Twin.*

La integración de varios dispositivos y diferentes protocolos de comunicación empleados en el desarrollo de la solución confirman la interoperabilidad de los dispositivos en un despliegue IoT e IIoT. El protocolo *MQTT*, uno de los más conocidos y empleados en despliegues IoT, es

altamente escalable, sencillo de implementar y de bajo consumo de energía en las comunicaciones. Además, la opción de integrar seguridades lo hace sumamente completo e integrable a las necesidades del entorno industrial.

Las bases de datos NoSQL aportan flexibilidad al sistema ya que permiten integrar múltiples dispositivos, que generan diferentes estructuras de información sin la necesidad de volver a diseñar la base de datos. En consecuencia, MongoDB es capaz de almacenar la información de las diferentes tecnologías y sensores desplegados en los sistemas IIoT.

La detección de eventos anormales en un sistema posibilita la toma de decisiones del mantenimiento predictivo, preventivo y en el peor de los casos correctivo. Conocer de manera prematura que un robot va a sufrir una posible falla permite aumentar la vida útil de los equipos y con ello disminuir las pérdidas de dinero por las paradas de producción. Los registros de las condiciones normales de funcionamiento del *Digital Twin* permiten aplicar diferentes técnicas para el mantenimiento predictivo.

En este estudio se aplicaron dos técnicas para el mantenimiento predictivo: el análisis de coeficiente de correlación y la elaboración de modelos con Inteligencia Artificial. Los resultados obtenidos indican que el coeficiente de correlación es una opción adecuada para hacer detección de anomalías del sistema, dado el gran desempeño que presentó. Así mismo, los modelos creados con Inteligencia Artificial respondieron dentro de los rangos esperados. Además, la base de datos implementada facilitó el manejo de la gran cantidad de información necesaria para la creación de los modelos.

Por último, existen gran cantidad de software para desarrollar sistemas SCADA. En el presente trabajo se empleó *Ignition,* por su versatilidad y características que presenta. Se creo el *dashboard,* para visualizar el resumen del *Digital Twin* y los datos históricos. *Ignition* es una gran herramienta, ya que integra la capacidad de manejo de usuarios, gestión de alarmas, envío de notificaciones por mensaje, correo electrónico, etc.

6.2 Problemas encontrados

Uno de los primeros problemas encontrados fue la placa seleccionada como gateway en primera instancia se planteó emplear la *Creator Ci40* de *Imagination,* pero la imposibilidad de

instalar paquetes *Python* con *pip* dificultó su implementación. Por lo que se cambió de placa y se optó por la *Raspberry Pi 3* que soporta BLE.

El estado de alarma decretado a nivel mundial debido la Pandemia COVID-19 se cerró el acceso a la universidad y fue imposible acceder al robot, para hacer las pruebas del algoritmo de detección de anomalías. En este sentido, solo fue posible la creación de varias secuencias tras una primera interacción con el robot, pero no se logró probar la infraestructura completa.

A nivel de software *SCADA* la mayoría trabaja solo en sistema operativo Windows y no tienen las características que un software de desarrollo actual debe tener, como es el desarrollo basado en web, interfaces *responsive*, integración de lenguajes de programación, multiplataforma, etc. *Ignition* es uno de los pocos que cumple con estas características propuestas, por lo que fue empleado para el desarrollo de la solución. A pesar de ser un software propietario, al igual que la mayoría de software SCADA, *Ignition* permite emplear el software sin licencia para desarrollar las aplicaciones y probarlas antes de ponerlas en producción.

Ignition a pesar de su versatilidad y poder integrar *Python*, tiene varias limitaciones en el manejo de gráficos. Si bien es cierto se maneja gran cantidad de datos aproximadamente 6000 en cada movimiento del *Digital Twin*, las pantallas de históricos en muchas ocasiones se vuelven muy pesadas.

6.3 Trabajo Futuro

A partir de los datos obtenidos en el presente trabajo, la línea de trabajo futura inmediata debería ser la implementación y la aplicación de las pruebas de la infraestructura en el brazo robótico físico. Si bien se ha probado y verificado su correcto funcionamiento en un ambiente de simulación, es de suma importancia verificar su comportamiento en un entorno real.

Las tecnologías empleadas permiten la escalabilidad del sistema, por lo que este trabajo sirve de base para la futura implementación de varios *Digital Twins* en el sistema y la verificación de su desempeño. La estructura de los tópicos MQTT y de las colecciones de la base de datos MongoDB permiten su integración, definiendo un identificador único para cada uno de ellos.

La infraestructura actual presenta un solo servidor que consta de: broker MQTT, servidor de base de datos (MongoDB), servidor *Ignition* y ejecución de algoritmos de detección de anomalías.

Un mejor desempeño se lograría separando cada uno de ellos, y con ello lograr una mayor capacidad de almacenamiento y de procesamiento.

Como alternativa a la integración de inteligencia artificial se pueden emplear servicios en la nube de Azure o AWS y probar su desempeño. De la misma forma, se puede integrar el servicio de gestión de identidad tanto para el manejo de dispositivos IoT como para el acceso a la aplicación. Además, un análisis costo-beneficio de las características que pudieran aportar al sistema contribuiría a la elección de la mejor solución.

Para finalizar, en el desarrollo de la solución se encontraron varias herramientas existentes para el análisis de datos a nivel industrial. Con la información almacenada en la base de datos se podrá integrar diferentes protocolos de comunicación para trabajar con dichas plataformas. Una alternativa es *Seeq*[19], una herramienta de analítica avanzada para procesos de manufactura.

[19] https://www.seeq.com/

Chapter 7 - Introduction

7.1 Background

Robots have been a critical part of the growth of manufacturing processes. There are two types of robots in the industry: industrial robots and service robots. Industrial robots are in charge of production processes, while service robots are created for more specific applications, to solve problems and perform human tasks, excluding them from the industrial sector [1].

Industrial robots have been used for several decades in manufacturing processes performing activities that are repetitive and sequential, such as welding, sorting, moving parts, etc. With its inclusion in the industrial sector, it has been possible to increase the quality of products and decrease production time. On the other hand, as they are programmable teams, there is flexibility when delegating tasks at different points in the process [2].

Every year the use of robots in the industry has an important growth. According to the International Federation of Robotics (IFR), 2018 had a sales record of 16.5 billion dollars, registering a growth of 6% in existing units worldwide compared to 2017. In 2019, a decrease is expected. in this growth, but for the years 2020, 2021 and 2022 an average growth of 12% per year is estimated as shown in Figure 1 1. On the other hand, the sector that owns the most units is the automotive sector, followed by progress very important of the manufacturers of electrical/electronic parts and devices [3].

In general, a production line is made up of several industrial robots, which must work together to produce the final product. That is, if one of them fails or has a fault, it implies that the line will completely stop unless each robot has its backup, which implies greater investment. This occurs in several sectors of the industry, so it is very important to carry out a maintenance plan to avoid losses in production and money.

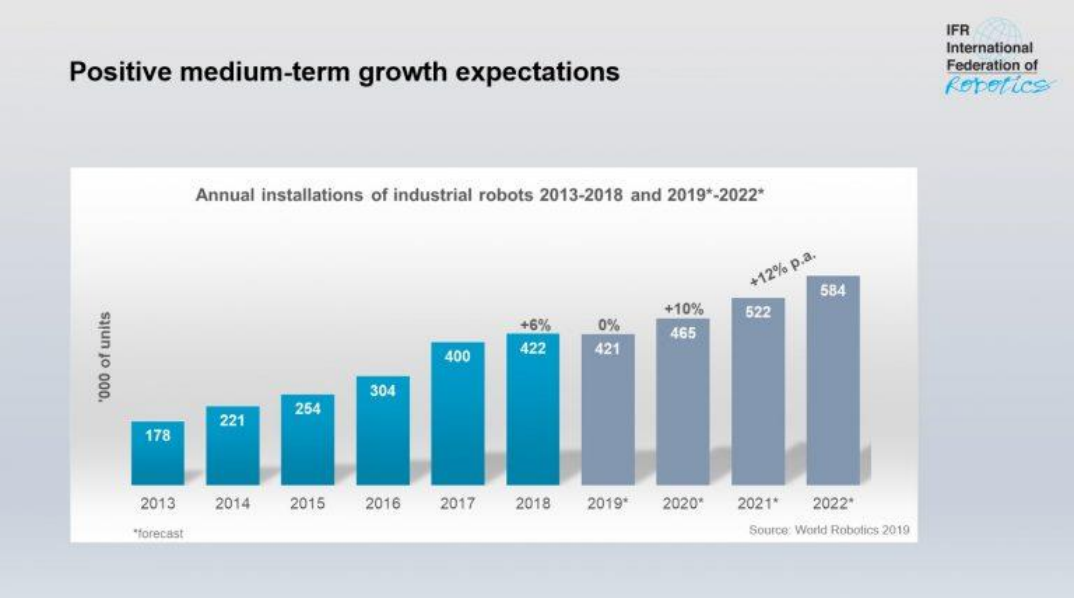


Figure 7-1. Annual Industrial Robot Installations 2013-2018 and prediction for 2019-2020 [3].

There are three types of maintenance that can be performed on equipment: corrective maintenance, preventive maintenance, and predictive maintenance. The first of them is the most critical because it is done when the robot has already suffered damage and stopped to repair it. Preventive maintenance is generally performed following the equipment manufacturer's recommendations and is periodic. Finally, predictive maintenance is based on the behavior of the equipment, that is, a diagnosis is made in real-time and by consulting historical data on the state of the equipment. With this, possible failures can be detected, and a work plan can be established, minimizing the loss due to equipment maintenance.

The robotic arms have the ability to carry out self-checks before starting operation, that is, when the system is turned on, it performs a verification sequence to detect possible faults that may exist in the system. The implementation of an external agent to take samples and to store the behavior of the robot at all times allows an analysis of possible failures that have not been detected by the robot's system. These sensors can be implemented throughout the production line, facilitating the monitoring of the state of the robots at all times.

The technology of the Internet of things (IoT) has several fields of application such as: Smart Cities, Medicine, resource management, logistics, industry, etc. The industry is one of the most relevant fields due to its importance at a global level. The IoT is a group of interconnected devices that allows its management, access to information, and current status. On the other hand, the sensor

nodes can generally be mobile, so a single sensor node can be shared between several robotic arms. In the same way, a sensor node has the capacity to carry out several measurements at the same time, so it can be adapted to the type of robot with which it is working. In other words, measurements from a production line can be made simultaneously and distributed with the same sensor without much investment.

7.2 Motivation

As explained before, industry is in the new era of digitalization. Industry is one of the main sectors of global economy and it must keep using the latest technology for achieving processes optimization, reducing costs and increasing productivity. In general, large amounts of data can be found in production plants, coming from sensors and actuators, which act in the production process.

The tools that the IoT offers to acquire data from sensors and actuators, interconnect devices, access information from anywhere, are of great help to make decisions in the process, integrating added value to the company. On the other hand, Artificial Intelligence allows companies to analyze the data and predict events, future damages or abnormal behavior of the machinery in order to reduce production costs and decrease downtimes of the manufacturing process.

One of the main issues in industry is when equipment or machinery are out of service which implies a loss in production and therefore a loss of money. The industry invests a lot of money in maintenance plans, to extend the life of the equipment and/or prevent the machine from breaking down. These maintenances are carried out based on the manufacturer's recommendations or the experience of the operators. To improve maintenance planning, predictive maintenance algorithms can be integrated to reduce the investment on spare parts and reduced unplanned downtimes.

Digital Twin (DT), is a very powerful tool since integrating IoT to acquire data and communicate devices, together with artificial intelligence to perform information processing, predictions can be made, and important decisions can be made in the process. This is the reason why a study of the latest trends in these paradigms will be carried out and a proof of concept will be implemented in a robotic arm to demonstrate the potential that it can generate at an industrial level.

7.3 Goals

The main objective of this work is to develop an infrastructure that allows maintaining a Digital Twin (DT) of the behavior of a robotic arm. For this purpose, the information from sensors to analyze the history of their measurements must be stored to detect abnormal behaviors in the movements. So, thanks to this detection and data processing, predictive maintenance can be performed.

7.3.1 General

To develop an infrastructure for the representation of the Digital Twin (DT) of a robotic arm, which allows monitoring its behavior.

7.3.2 Specific

- To monitor the behavior of the robotic arm, to detect anomalies in its operation.
- To determine the necessary quantities that must be monitored to carry out a correct analysis of possible failures of the robotic arm.
- To analyze the possibility of using a generic sensor node to perform data acquisition on robotic arms.
- To develop a predictive maintenance analysis system, with the data obtained from the sensor node.
- To store the information from the sensors to have a history of the data and view them on a dashboard created in industrial SCADA software.

7.4 Work Plan

The first stage of the work focuses on gathering the relevant information available, studying the tools and technologies that have been used to carry out similar projects, that is, a study of the state of the art. A set of general concepts will be established that must be understood to enter the development of the proof of concept. This point is important since the focus of the research and the implementation of the project is defined.

The second stage is based on the definition and development of the IoT infrastructure, in this phase, all the technologies that will be used to carry out the solution are described in detail. It

describes the sensor node that is used, the type of communications with which it is going to work, and the software tools that will be used to achieve the stated objectives. Once the infrastructure is defined, the application will be carried out with the robotic arm, having to define the measures to be taken with the available sensors. In other words, the information to be taken from the robotic arm will be defined for subsequent analysis and testing.

Finally, in the third stage of the project, the tests to be carried out to evaluate the operation of the created solution are defined, basically they are the proofs of concept selected to demonstrate the use and capabilities of DT. At this point, the established anomaly detection algorithm must be executed to evaluate its correct operation and verify the historical data that can be seen from a dashboard.

Chapter 8 - Conclusions and future work

In this work, the infrastructure implementation and the tests carried out to reach the objectives have been explained in detail. This chapter aims to synthesize all the achievements obtained, the problems found and the possible lines of research that can be covered at IIoT sector. This information may be considered for future developments.

8.1 Conclusions

Throughout the work, different alternatives and existing deployments of Digital Twin were studied, which were the basis for the development of the raised infrastructure. A Digital Twin of the behaviour of a simulated and physical robotic arm was constructed, storing the information of the deployed sensor node. A history of the relevant data was obtained in order to make decisions on its operation.

Robotic arms generally have a high degree of accuracy and precision, so measuring acceleration and rotation was the ideal option to create the Digital Twin. The characteristics of the environment in which you are working directly influence the useful life of the robotic arm, so measurements and records of temperature, humidity, etc. are necessary. These records are not only useful for the end customer, but also for the company that has created the robotic arm since they allow studying the performance of the robot under different conditions.

For data acquisition, the correct selection of the Sensor Node which you are going to work with is very important. The measures to be taken and the work characteristics must be defined, such as: resolution, frequency, etc. In the implemented solution the SensorTag fulfilled the objective of using a single type of generic and versatile Node that can be adapted to any robotic arm, insomuch as a single device allows the measurement of various quantities that provide relevant information on the Digital Twin.

The integration of various devices and different communication protocols used in the development of the solution confirm the interoperability of the devices of an IoT and IIoT deployment. The MQTT protocol, one of the best known and used in IoT deployments, is a highly scalable, simple to implement and low power consumption system in communications.

Furthermore, the option of integrating securities makes it highly complete and integrable to the needs of the industrial environment.

NoSQL databases provide flexibility to the system since they allow integrating multiple devices that generate different information structures without the need to redesign the database. Thus, MongoDB is capable of storing the information of the technologies and sensors grouped in IIoT systems.

Detecting abnormal events in a system enables to make decisions about predictive, preventive and, in the worst case, corrective maintenance. Knowing prematurely that a robot is going to suffer a possible failure allows to increase the useful life of the equipment and thereby decrease the loss of money due to production stops. The records of the normal operating conditions of the Digital Twin allow the application of different techniques for predictive maintenance.

In this study, two techniques were applied for predictive maintenance: the correlation coefficient analysis and the elaboration of models with artificial intelligence. The results obtained indicate that the correlation coefficient is an appropriate option to detect anomalies in the system, due to the great performance it presented. Likewise, the models created with Artificial Intelligence responded within the expected ranges. Furthermore, the implemented database facilitated the handling of the large amount of information necessary to create a model that works correctly.

Finally, there is a large amount of software to develop SCADA systems. Ignition was used in the present work due to its versatility and characteristics. The dashboard was created to visualize the Digital Twin summary and historical data. Ignition is a great tool, since it integrates the ability to manage users, manage alarms, send notifications by message, email, etc.

8.2 Problems Found

One of the first problems encountered was the board selected as the Gateway. In the first instance, the idea was to use the Creator Ci40 from Imagination, but the impossibility of installing Python packages with pip made its implementation a problem. Therefore, the board was changed and the Raspberry Pi 3 that supports BLE was chosen.

The state of alarm decreed due to the CoVid-19 pandemic closed Access to the university and made impossible the access to the robot to test the anomaly detection algorithm. The creation of

several sequences was only possible after a first interaction with the robot, but it was not possible to test the complete infrastructure.

Most of SCADA software work only on Windows operating systems and do not have the characteristics that current development software should have, such as: web-based development, responsive interfaces, integration of programming languages, multiplatform, etc. Ignition is one of the few that meets these proposed characteristics, which is why it was used to develop the solution. Despite being closed-source software, like most SCADA software, Ignition allows to use unlicensed software to develop applications and test them before applying in production.

Despite its versatility and being able to integrate Python, Ignition has several limitations in graphics management. Although it is true that a large amount of data is handled, approximately 6000 in each movement of the Digital Twin, the historical screens often become very heavy.

8.3 Future Work

From the data obtained in this work, the future line of work should be the implementation and application of infrastructure tests in the physical robotic arm. Although its correct operation has been tested and verified in a simulation environment, it is extremely important to verify its behaviour in a real environment.

The technologies used allow the scalability of the system, so this work can be used as the basis for the future implementation of various Digital Twins in the system and the verification of their performance. The structure of the MQTT topics and the collections of the MongoDB database allow their integration, defining a unique identifier for each of them.

The current infrastructure presents a single server consisting of: MQTT broker, database server (MongoDB), Ignition server and execution of anomaly detection algorithms. Better performance would be achieved by separating each of them, and thereby achieve greater storage and processing capacity.

As an alternative to Artificial Intelligence integration, it is possible to use Azure or AWS cloud services and test their performance. In the same way, the identity management service can be integrated for both IoT device management and application access. In addition, a cost-benefit analysis of the system characteristics would help to choose the best algorithm.

Finally, in the development of the solution, several tools existing for the analysis of data at an industrial level were found. With the information stored in the database, different communication methods can be integrated to work with these platforms. An alternative is Seeq, an advanced analytics tool for manufacturing processes.

Capítulo 9 - Bibliografía

[1] Shukla, M., & Shukla, A. N. (2012). Growth of robotics industry early in 21st century. *International Journal of Computational Engineering Research*, *2*(5), 1554-1558.

[2] Fryman, J., & Matthias, B. (2012, May). Safety of industrial robots: From conventional to collaborative applications. In ROBOTIK 2012; 7th German Conference on Robotics (pp. 1-5). VDE.

[3] IFR, I. F. (2019). World Robotics Industrial Robots. Kawasaki Heavy Industries, Ltd.

[4] Zhou, K., Liu, T., & Zhou, L. (2015, August). Industry 4.0: Towards future industrial opportunities and challenges. In 2015 12th International conference on fuzzy systems and knowledge discovery (FSKD) (pp. 2147-2152). IEEE.

[5] Xu, L. D., Xu, E. L., & Li, L. (2018). Industry 4.0: state of the art and future trends. International Journal of Production Research, 56(8), 2941-2962.

[6] Tay, S. I., Lee, T. C., Hamid, N. A. A., & Ahmad, A. N. A. (2018). An Overview of Industry 4.0: Definition, Components, and Government Initiatives. Journal of Advanced Research in Dynamical and Control Systems, 10(Special Issue), 1379.

[7] Erboz, G. (2017). How to define industry 4.0: main pillars of industry 4.0. Szent Istvan University, Gödöllő, 1-9.

[8] Davies, R. (2015). Industry 4.0: Digitalisation for productivity and growth (Briefing). European Parlamentary Research Service, PE 568.337.

[9] Wallén, J. (2008). The history of the industrial robot. Linköping University Electronic Press.

[10] Bizony, P. (2015). Focus: THE 3 Laws of Robotics. Engineering & Technology, 10(6), 44-45.

[11] Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: correlation and comparison. Engineering, 5(4), 653-661.

[12] Aivaliotis, P., Georgoulias, K., Arkouli, Z., & Makris, S. (2019). Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance. Procedia CIRP, 81, 417-422.

[13] Negri, E., Fumagalli, L., & Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. Procedia Manufacturing, 11, 939-948.

[14] Qi, Q., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. Ieee Access, 6, 3585-3593.

[15] Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. The International Journal of Advanced Manufacturing Technology, 94(9-12), 3563-3576.

[16] Vachálek, J., Bartalský, L., Rovný, O., Šišmišová, D., Morháč, M., & Lokšík, M. (2017, June). The digital twin of an industrial production line within the industry 4.0 concept. In 2017 21st International Conference on Process Control (PC) (pp. 258-262). IEEE.

[17] Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine, 51(11), 1016-1022.

[18] Borgi, T., Hidri, A., Neef, B., & Naceur, M. S. (2017, January). Data analytics for predictive maintenance of industrial robots. In 2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET) (pp. 412-417). IEEE.

[19] Rosen, R., Von Wichert, G., Lo, G., & Bettenhausen, K. D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine, 48(3), 567-572.

[20] Dahmen, U., & Rossmann, J. (2018, October). Experimentable digital twins for a modeling and simulation-based engineering approach. In 2018 IEEE International Systems Engineering Symposium (ISSE) (pp. 1-8). IEEE.

[21] Gilchrist, A. (2016). Industry 4.0: the industrial internet of things. Apress.

[22] Witkowski, K. (2017). Internet of things, big data, industry 4.0–innovative solutions in logistics and supply chains management. Procedia Engineering, 182, 763-769.

[23] Vargas, D. C. Y., & Salvador, C. E. P. (2016). Smart IoT gateway for heterogeneous devices interoperability. IEEE Latin America Transactions, 14(8), 3900-3906.

[24] Haag, S., & Anderl, R. (2018). Digital twin–Proof of concept. Manufacturing Letters, 15, 64-66.

[25] Wang, J., Ye, L., Gao, R. X., Li, C., & Zhang, L. (2019). Digital Twin for rotating machinery fault diagnosis in smart manufacturing. International Journal of Production Research, 57(12), 3920-3934.

[26] Schroeder, G. N., Steinmetz, C., Pereira, C. E., & Espindola, D. B. (2016). Digital twin data modeling with automationml and a communication methodology for data exchange. IFAC-PapersOnLine, 49(30), 12-17.

[27] Oyekan, J. O., Hutabarat, W., Tiwari, A., Grech, R., Aung, M. H., Mariani, M. P., ... & Dupuis, C. (2019). The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans. Robotics and Computer-Integrated Manufacturing, 55, 41-54.

[28] Ahmed, E., Yaqoob, I., Gani, A., Imran, M., & Guizani, M. (2016). Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. IEEE Wireless Communications, 23(5), 10-16.

[29] Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. The Internet Society (ISOC), 80.

[30] Liu, Z., Meyendorf, N., & Mrad, N. (2018, April). The role of data fusion in predictive maintenance using digital twin. In AIP Conference Proceedings (Vol. 1949, No. 1, p. 020023). AIP Publishing LLC.

[31] Glaessgen, E., & Stargel, D. (2012, April). The digital twin paradigm for future NASA and US Air Force vehicles. In 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA (p. 1818).

[32] Magargle, R., Johnson, L., Mandloi, P., Davoudabadi, P., Kesarkar, O., Krishnaswamy, S., ... & Pitchaikani, A. (2017, July). A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system. In Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017 (No. 132, pp. 35-46). Linköping University Electronic Press.

[33] Korčok, D. (2016). Industry 4.0: The future concepts and new visions of factory of the future development. In Sinteza 2016-International Scientific Conference on ICT and E-Business Related Research (pp. 293-298). Singidunum University.

[34] Queiroz, J., Leitão, P., Barbosa, J., & Oliveira, E. (2019, August). Agent-Based Approach for Decentralized Data Analysis in Industrial Cyber-Physical Systems. In International Conference on Industrial Applications of Holonic and Multi-Agent Systems (pp. 130-144). Springer, Cham.

[35] Santos, B. P., Charrua-Santos, F., & Lima, T. M. (2018, July). Industry 4.0: an overview. In Proceedings of the World Congress on Engineering (Vol. 2, pp. 4-6).

[36] Trappey, A. J., Trappey, C. V., Govindarajan, U. H., Chuang, A. C., & Sun, J. J. (2017). A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. Advanced Engineering Informatics, 33, 208-229.

[37] Pelegrí, J. (10 de enero de 2019). Universal-Robots. Obtenido de LA CUARTA REVOLUCIÓN INDUSTRIAL: COBOTS Y AUTOMATIZACIÓN: https://blog.universal-robots.com/es/revolucion-industrial-cobots. Última visita: 10/03/2020.

[38] Pozhilova, A. N. (2016). Estudio de la plataforma Sensortag de Texas Instruments para aplicaciones en redes inalambricas de sensores.

[39] Texas Instruments SimpleLink Multi-Standard CC2650 SensorTag. Obtenido de http://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG. Última visita: 11/05/2020.

[40] Galar, D., & Kumar, U. (2017). EMaintenance: Essential electronic tools for efficiency (pp. 1-72). Academic Press.

[41] Texas Instruments. CC2650 SensorTag User's Guide. Obtenido de https://processors.wiki.ti.com/index.php/CC2650_SensorTag_User's_Guide. Última visita: 11/05/2020.

[42] Streijl, R. C., Winkler, S., & Hands, D. S. (2016). Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. Multimedia Systems, 22(2), 213-227.

[43] MongoDB. https://www.mongodb.com/what-is-mongodb. Última Visita: 11/05/2020

[44] Inductive Automation. Ignition SCADA Software. Obtenido de https://inductiveautomation.com/scada-software/. Última Visita: 11/05/2020

[45] Mohammad A. (2018). Novel Bits. INTRO TO BLUETOOTH LOW ENERGY (pp. 11-26).

[46] Dobot (2019). Shenzhen Yuejiang Technology Co., Ltd. Dobot Magician User Guide.

[47] The MathWorks, Inc. Visualize Manipulator Trajectory Tracking with Simulink 3D Animation. Obtenido de https://es.mathworks.com/help/robotics/examples/visualize-manipulator-trajectory-tracking-with-simulink-3d.html?lang=en#VisualizeManipulatorTrajectoriesWithSimulink3DAnimationExample-1. Última Visita: 15/05/2020

[48] The MathWorks, Inc (2020). Robotics System Toolbox™ Reference.

[49] The MathWorks, Inc (2020). Simulink® 3D Animation™ User's Guide.

APÉNDICES

Apéndice A - Códigos de los programas realizados

Los códigos implementados se los expondrá en el siguiente repositorio público de *GitHub:* https://github.com/Richard0493/TFMMIoT. En la Tabla 1, se detalla el contenido del repositorio.

Tabla A-1. Detalle Contenido Repositorio Códigos TFM

Carpeta	Nombre Archivo	Detalle
Matlab	TFMMIoTMain.m	Programa principal, inicializaciones variables.
	TFMMIoTMqttCon.m	Programa establecer conexión con broker MQTT.
	TFMMIoTSigSave.m	Programa para publicar datos en tópico MQTT de la simulación del brazo robótico.
	TFMMIoTSimAR.slx	Programa en *Simulink* de la simulación del brazo robótico.
Gateway	TFMMIoTGtwMqttPublish.py	Programa adquisición de datos del *SensorTag* por BLE y publicar en los tópicos broker MQTT.
	TFMMIoTSensorTag.py	Programa transformación de valores del *SensorTag.*
Server	TFMMIoTIgnHist.py	Programa para realizar la consulta en la base de datos de la pantalla Históricos Generales del *dashboard*.
	TFMMIoTMlPredict.py	Programa para realizar predicciones de velocidad y fuerza, de los modelos de Machine Learning creados.

	TFMMIoTMlTrain.py	Programa para crear y entrenar los modelos de Machine Learning.
	TFMMIoTMongoDb.py	Programa que maneja todas las funciones de conexión, lectura y escritura con la base de datos MongoDB.
	TFMMIoTSignalCorr.py	Programa para calcular el coeficiente de correlación de las secuencias.
	TFMMIoTSvrMqttSubscribe.py	Programa a la escucha de mensajes MQTT para almacenar los datos que provienen del *SensorTag* y de la simulación de Matlab.
	TFMMIoTIgnBack.gwbk	Respaldo del *dashboard* creado. Al restaurar en Ignition se crea una imagen del que esta funcionando.

Apéndice B - Manual Usuario Dashboard

El presente apéndice, tiene como objetivo detallar el manejo de la aplicación para revisar el estado actual del sistema y realizar las consultas de los datos y alarmas históricas del Digital Twin.

B.1 Acceso a la aplicación

B.1.1 Para acceder a la aplicación se debe acceder desde cualquier navegador al siguiente enlace:

http://cripta.fdi.ucm.es:8088/data/perspective/client/TFMDashBoard

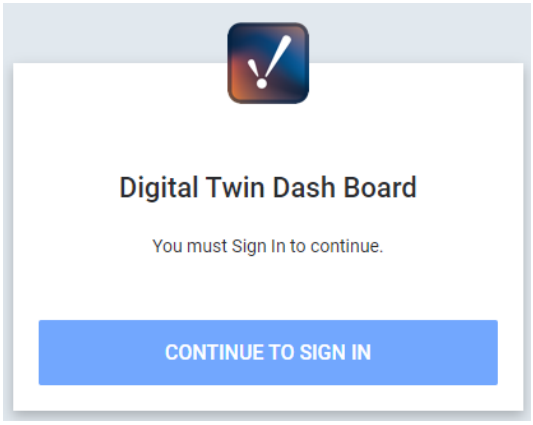


Figura B-1. Broker MQTT

B.1.2 Dependiendo del usuario con el que se ingrese a la aplicación se van a poder realizar las acciones. Los tipos de usuario que existen la aplicación se los muestra en la Tabla A-1.

Tabla B-1. Roles creados en la aplicación

Rol	Detalles
Espectador	Solo puede visualizar y realizar consultas básicas de los movimientos realizados. Acceso a la pantalla resumen, históricos por movimiento y resumen de alarmas
Operador	Privilegios de rol espectador y consulta detallada de los movimientos y reconocimiento de alarmas. Acceso a pantallas de espectador, histórico general y banner de alarmas.
Administrador	Privilegios de rol operador, control total de la aplicación y acceso para realizar cambios en el diseño de las pantallas.

B.2 Distribución de la aplicación

La aplicación está conformada por cinco pantallas: inicio, resumen, históricos generales, históricos por movimiento, banner de alarmas e histórico de alarmas. En este manual se va a explicar el funcionamiento de cada una de ellas. Desde todas las pantallas se puede visualizar la barra superior y el contenido de cada una de las pantallas como se muestra en la Figura A-2.

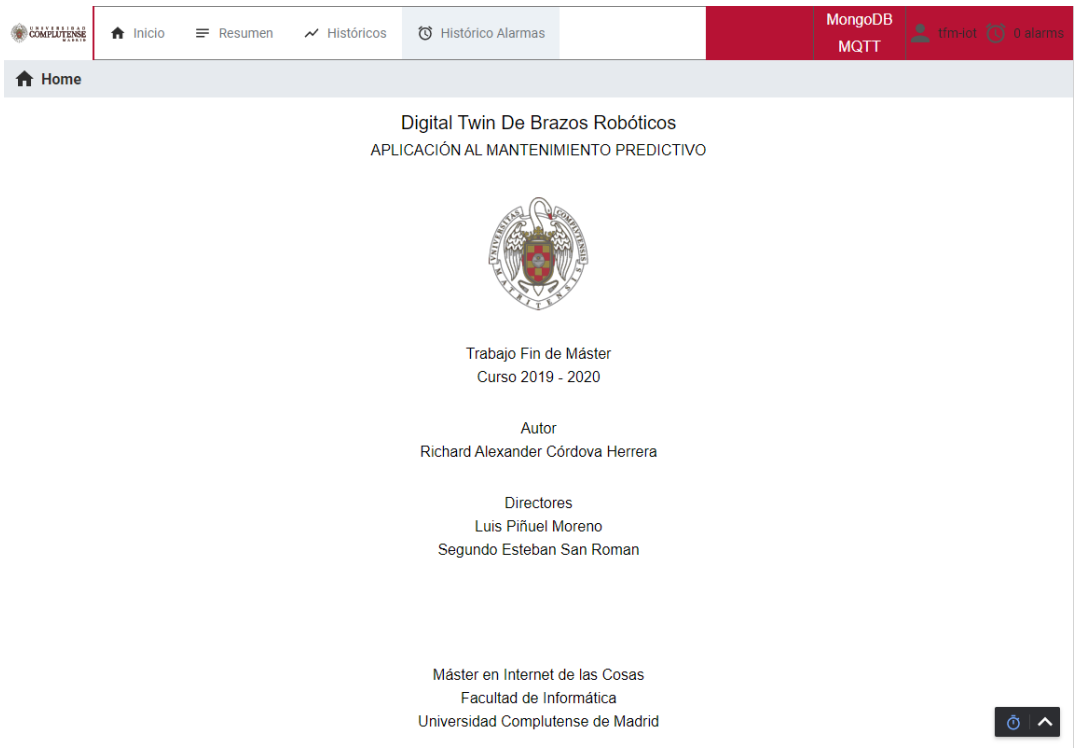


Figura B-2. Esquema pantalla dashoard

B.2.1 Barra superior

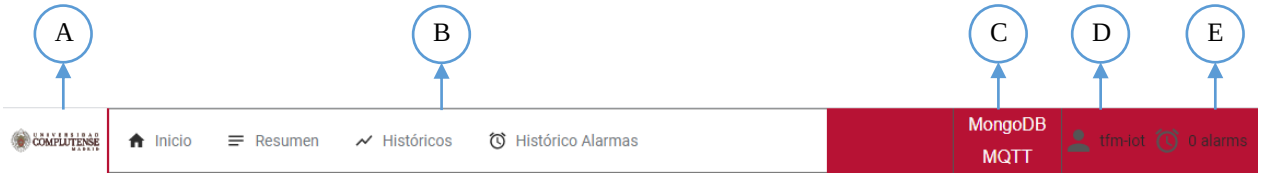


Figura B-3. Barra Superior Dashboard

A. Logo de la universidad. En caso de que se llegue a implementar en una compañía se cambiará por el logo de la empresa.
B. Barra de navegación, para moverse entre las pantallas del *dashboard*.

C. Indicadores de estado de las comunicaciones. Tienen dos estados que se muestran en la Tabla A-2.

Tabla B-2. Significado Estados Comunicaciones

Estado	Descripción
MongoDB	Conexión válida
MQTT	No existe conexión

D. Información usuario conectado. Se puede visualizar el usuario conectado y cerrar la sesión para cambiar de usuario.
E. Indicador alarmas activas. En el caso de que existan alarmas cambiará de color el texto para alertar al operador.

B.3 Pantallas principales del Dashboard

B.3.1 Pantalla Inicio

La pantalla de inicio se muestra en la Figura A-4, es una pantalla de presentación del Dashboard. En este caso se muestra información del tema del proyecto y las personas que desarrollaron y colaboraron en su implementación.

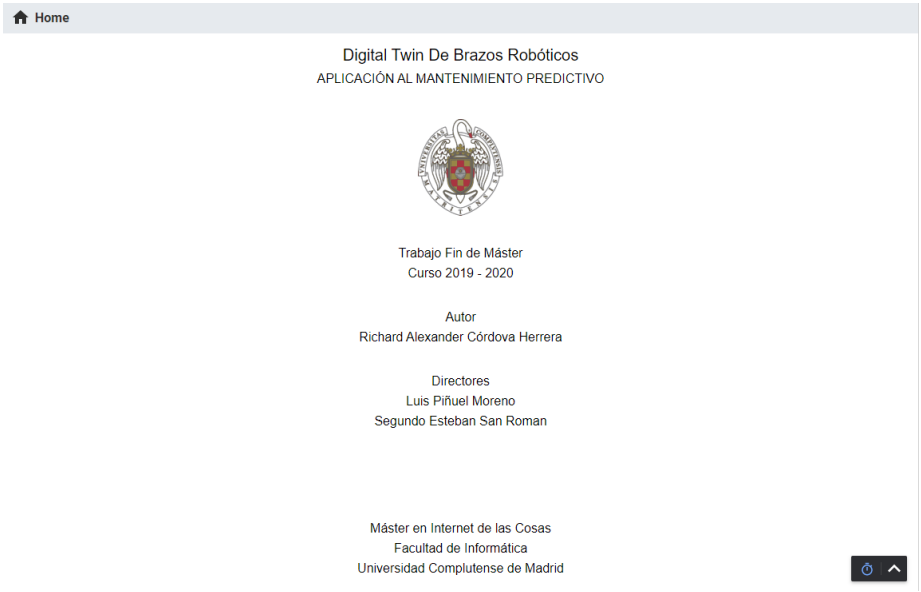


Figura B-4. Pantalla Inicio

B.3.2 Pantalla Resumen

En la pantalla de resumen se puede ver el estado actual del *Digital Twin*. En la parte superior se aprecia una gráfica con los movimientos actuales del brazo robótico, exactamente los últimos 5 segundos y su velocidad. Por otro lado, en la parte inferior se muestra un resumen de las temperaturas ambiente y del sensor de los últimos 30 minutos.

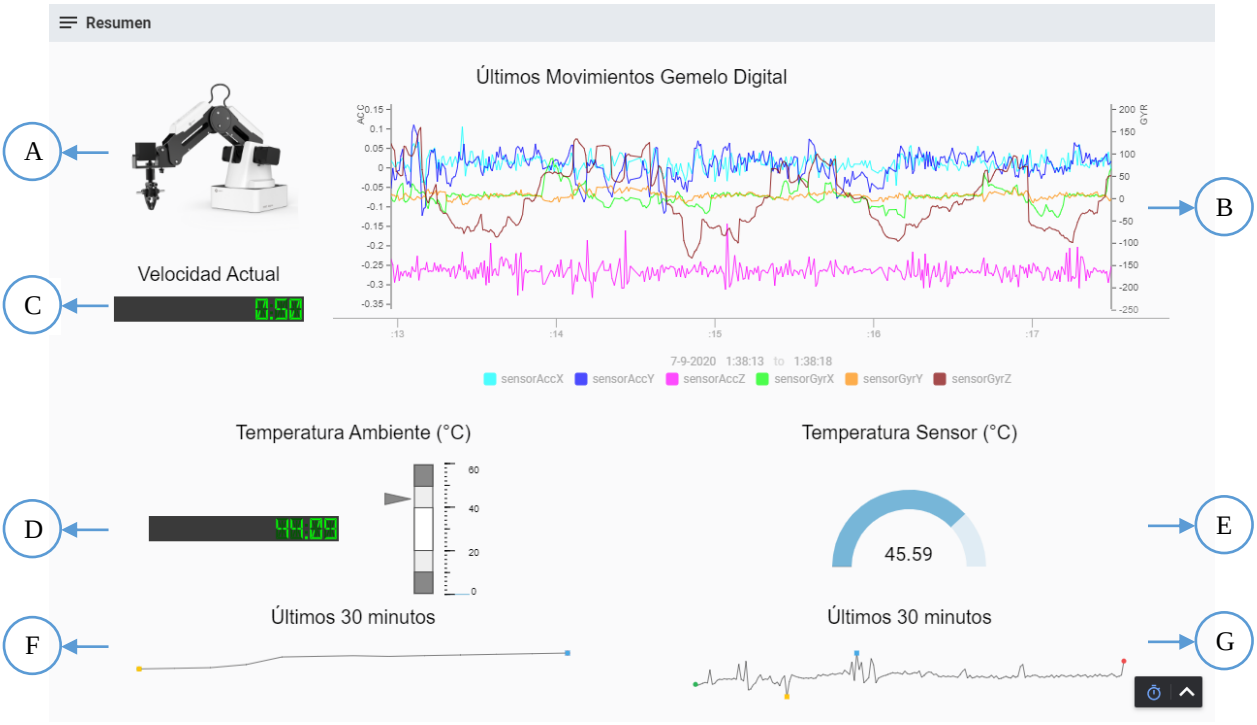


Figura B-5. Pantalla Resumen

A. Representación del brazo robótico.
B. Gráfica de últimos movimientos aceleraciones y rotaciones en los ejes X, Y y Z cada una.
C. Velocidad actual del movimiento, se lo calcula mediante redes neuronales.
D. Valor temperatura ambiente actual, representada en °C.
E. Valor temperatura del sensor actual, representada en °C.
F. Tendencia del comportamiento de la temperatura ambiente en los últimos 30 minutos.
G. Tendencia del comportamiento de la temperatura del sensor en los últimos 30 minutos.

B.3.3 Pantalla Históricos Generales

La consulta de históricos se la puede realizar de dos formas: general que se cargan todos los movimientos dentro de un rango de fechas o por movimiento. En la Figura A-6, se muestra la pantalla históricos generales. En esta pantalla se puede consultar y visualizar los históricos de movimientos y de los sensores registrados, en este caso la temperatura ambiente y del dispositivo. Para realizar las consultas se debe seleccionar un rango de fecha válido, seleccionar el tipo de movimiento y dar clic en el botón **Buscar y Graficar.** En el caso del movimiento se puede filtrar solo el acelerómetro, el giroscopio o los dos, seleccionando la casilla de selección de cada uno de ellos.

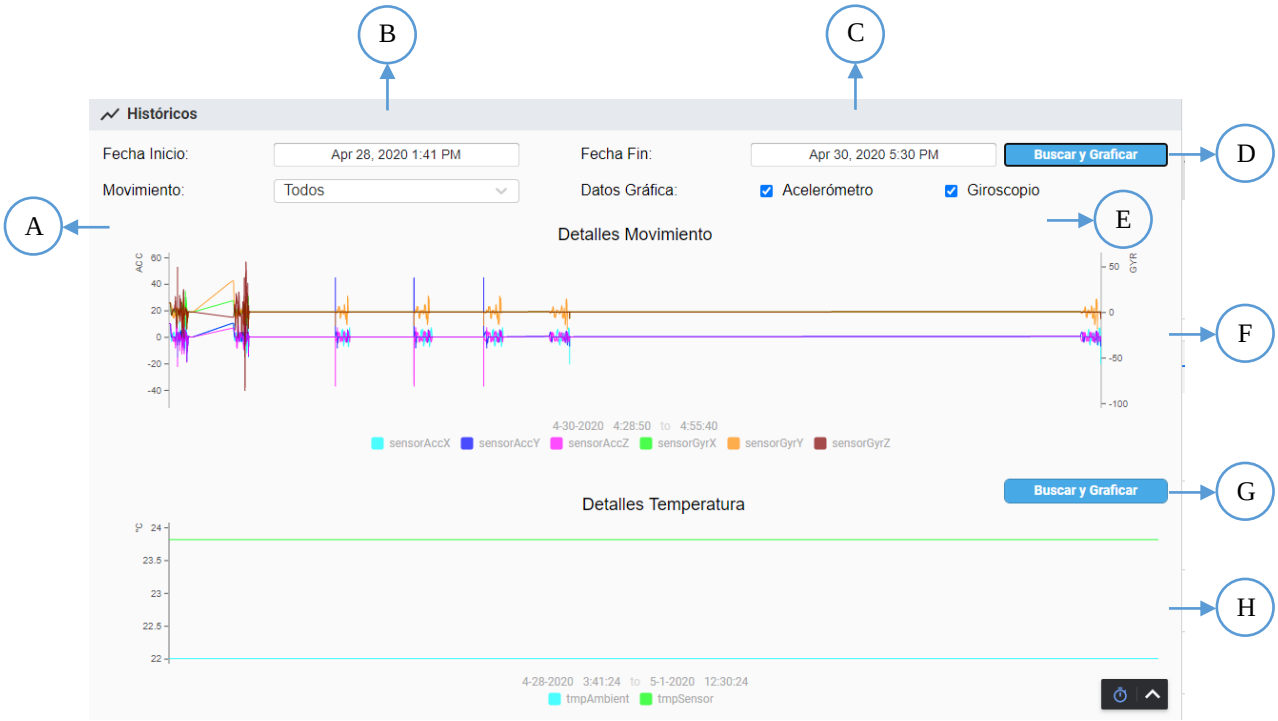


Figura B-6. Pantalla Históricos Generales

A. Seleccionar el tipo de movimiento: todos, secuencia 1, secuencia 2 y secuencia 3.
B. Seleccionar fecha inicio para la búsqueda.
C. Seleccionar fecha fin para la búsqueda.
D. Buscar y graficar los movimientos existentes dentro del rango seleccionado del *Digital Twin*, aceleración y rotación.
E. Casillas de selección para mostrar información de aceleración, rotación, ambos o ninguna.

F. Tendencia de los movimientos del *Digital Twin*.

G. Buscar y graficar los valores de temperatura existentes dentro del rango seleccionado del *Digital Twin.*

H. Tendencia de la temperatura ambiente y del sensor del *Digital Twin*.

B.3.4 Pantalla Histórico por Movimiento

La pantalla de históricos por movimiento únicamente muestra un movimiento a la vez, para poder tener una mayor interpretación de los datos. Para cargar un movimiento se deben seguir los siguientes pasos:

1. Seleccionar un rango de fecha válido y buscar los movimientos.
2. Si existen movimientos se deben cargar en la tabla, seleccionar un movimiento.
3. Se habilita el botón Graficar, y se despliega el gráfico de dicho movimiento.
4. En esta pantalla se puede comparar el movimiento actual con el movimiento ideal. Para ello se da clic en el botón Patrón y se podrán ver las señales ideales de cada uno de los movimientos.

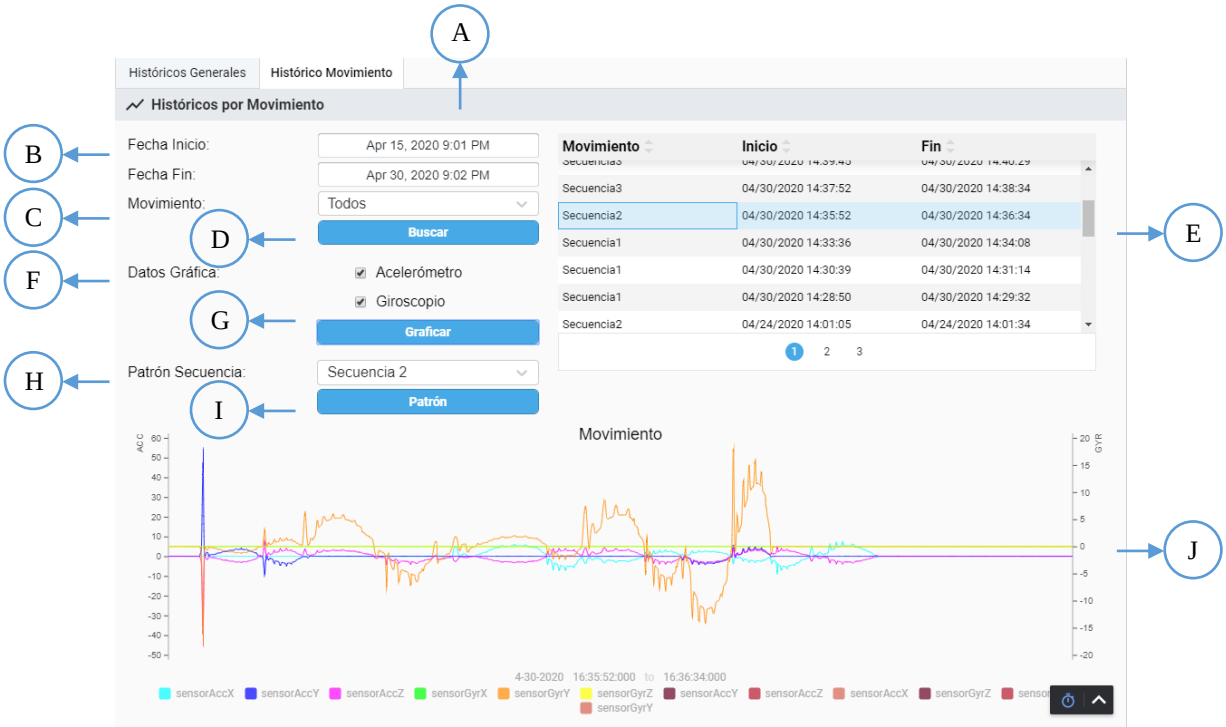


Figura B-7. Pantalla Históricos Por Movimiento

A. Seleccionar fecha inicio para la búsqueda.
B. Seleccionar fecha fin para la búsqueda.
C. Seleccionar el tipo de movimiento: todos, secuencia 1, secuencia 2 y secuencia 3.
D. Buscar movimientos registrados que cumplan el criterio de la búsqueda.
E. Tabla donde se carga el resultado de la búsqueda realizada, se puede seleccionar un movimiento para visualizarlo en la tendencia.
F. Casillas de selección para mostrar información de aceleración, rotación, ambos o ninguna.
G. Buscar y graficar los movimientos existentes del movimiento seleccionado del *Digital Twin,* aceleración y rotación.
H. Seleccionar el tipo de movimiento para graficar el patrón del movimiento: secuencia 1, secuencia 2 y secuencia 3.
I. Buscar y graficar los movimientos existentes del patrón seleccionado del *Digital Twin,* aceleración y rotación.
J. Tendencia de los movimientos del *Digital Twin*.

B.3.5 Pantalla Banner Alarmas

La pantalla del resumen de alarmas o banner de alarmas se muestra en la Figura A-8, este es un componente propio de Ignition que muestran las últimas alarmas que se han registrado en el sistema. Las alarmas pueden tener 4 estados que se muestran en la Tabla A-3.

Tabla B-3. Significado Estados Banner de Alarmas

Estado	Descripción
Activa, no reconocida	La alarma esta activa y no ha sido reconocida por ningún usuario.
Activa, reconocida	La alarma sigue activa y ya ha sido reconocida por un usuario.
Limpia, no reconocida	La alarma ya se arregló, pero no ha sido reconocida por ningún usuario
Limpia, reconocida	La alarma ya se arregló y fue reconocida. **Esta alarma ya no se muestra en el banner.**

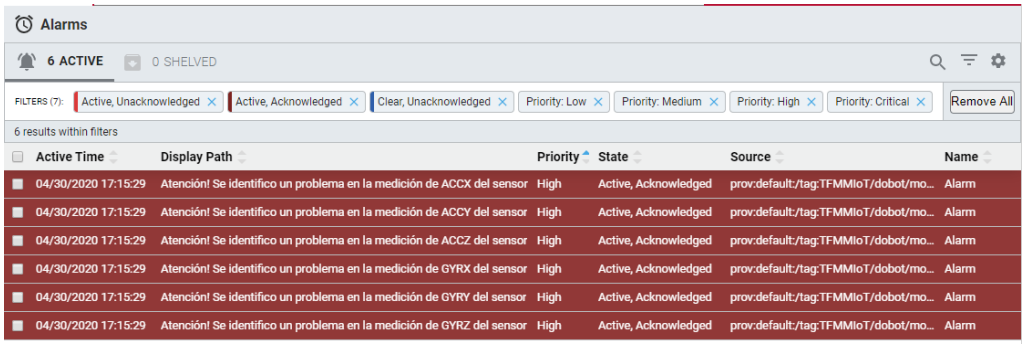


Figura B-8. Pantalla Banner Alarmas

B.3.6 Pantalla Históricos Alarmas

Por último, está la pantalla de histórico de alarmas, donde se van a mostrar todos los eventos que han registrado en la base de datos. En la Figura A-9 se muestra la distribución de esta pantalla. Para graficas las señales y verificar el problema se debe seguir los siguientes pasos:

1. Seleccionar un rango de fecha válido y buscar los movimientos.
2. Si existen alarmas en el criterio de búsqueda seleccionado se deben cargar en la tabla.
3. Seleccionar el movimiento de interés.
4. Se habilita el botón **Graficar**, y se despliega el gráfico de la señal seleccionada.
5. En esta pantalla se puede comparar el movimiento actual con el movimiento ideal. Para ello se da clic en el botón **Graficar Patrón** y se podrán ver la señal.

A. Seleccionar fecha inicio para la búsqueda.
B. Seleccionar fecha fin para la búsqueda.
C. Seleccionar el tipo de movimiento: todos, secuencia 1, secuencia 2 y secuencia 3.
D. Buscar alarmas registradas que cumplan el criterio de la búsqueda.
E. Tabla donde se carga el resultado de la búsqueda realizada, se puede seleccionar una alarma para visualizar la señal.
F. Buscar y graficar la señal por la que se registró la señal de alarma.
G. Seleccionar el tipo de movimiento para graficar el patrón del movimiento: secuencia 1, secuencia 2 y secuencia 3.
H. Buscar y graficar la señal patrón para realizar la comparación con la señal actual y evidenciar la variación.

I. Gráfica de la señal que activó la alarma y del movimiento ideal, en este caso como se muestra en la Figura A-9, se visualiza claramente que existe gran diferencia entre las dos señales.

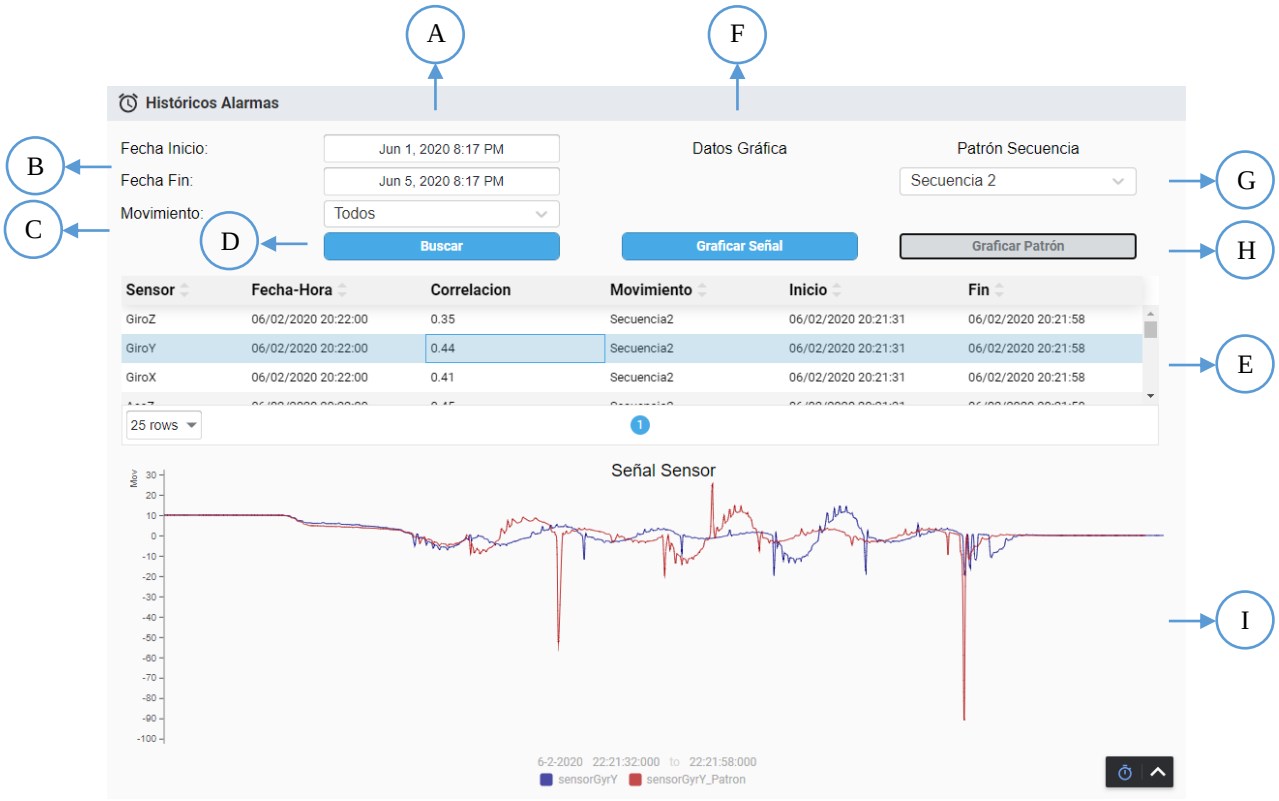


Figura B-9. Pantalla Histórico Alarmas

Apéndice C - Configuración MQTT en Ignition

C.1 Configurar conexión con broker MQTT

1. Lo primero que se debe realizar es instalar el módulo MQTT ***Engine***, disponible en la página oficial de descargas de Inductive Automation. Una vez instalado el módulo en la pantalla de configuración, nos dirigimos a la sección MQTT ***Engine*** y se selecciona la pestaña ***Servers***.

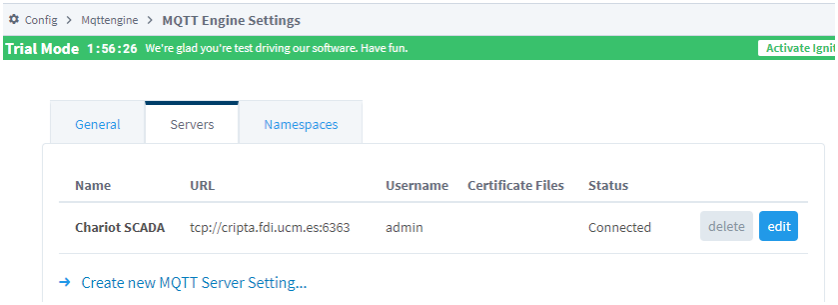


Figura C-1. Broker MQTT

2. Editar conexión disponible, para ello debemos tener la dirección del broker y las credenciales en el caso de ser necesario.

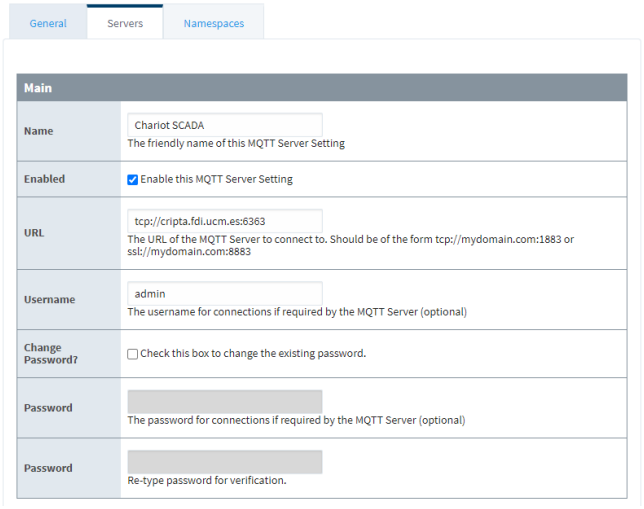


Figura C-2. Configuraciones Broker MQTT

3. Se guarda la configuración y el estado de la conexión debe cambiar a ***connected***.

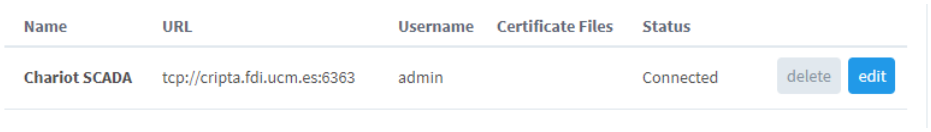


Figura C-3. Configuraciones Broker MQTT

C.2 Suscripción a Tópicos

1. Una vez conectado el broker MQTT, se pueden crear suscripciones a tópicos deseados. Para ellos nos dirigimos a la pestaña ***Namespaces*** y dentro de ella seleccionar ***Custom***.

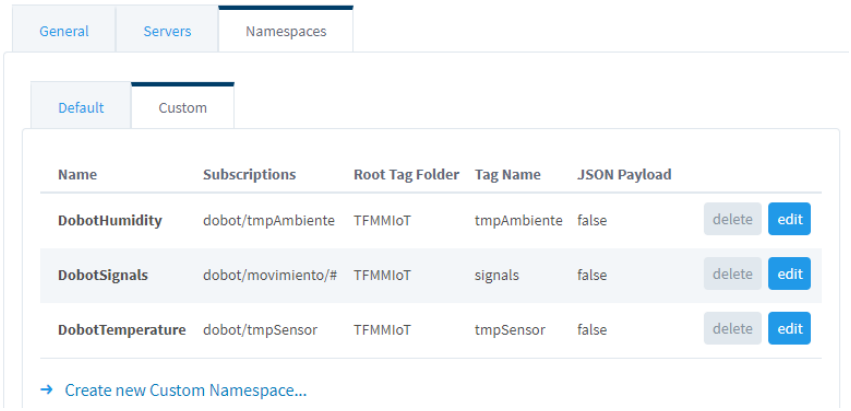


Figura B-4. Creación un nuevo Namespace

2. Seleccionar ***Create new Custom Namespace.***
3. Realizar la configuración como se muestra en la Figura B-5.

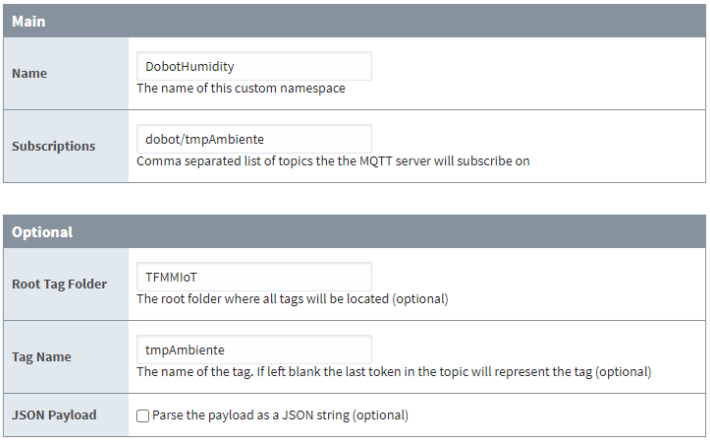


Figura B-5. Configuración Namespace

4. Las suscripciones se deben reflejar en el diseñador de Ignition.