



*Control de un módulo robótico mediante el uso del cloud computing y una app móvil*

*Control de un módulo robótico mediante el uso del cloud computing y una app móvil*

*Controle de um módulo robótico através do uso de computação em nuvem e um aplicativo móvel*

Juan David Chimarro-Amaguaña <sup>I</sup>  
[juan.chimarro@intsuperior.edu.ec](mailto:juan.chimarro@intsuperior.edu.ec)  
<https://orcid.org/0000-0001-9454-8357>

Delia Jazmín Vilatuña-Catagña <sup>II</sup>  
[delia.vilatuna@intsuperior.edu.ec](mailto:delia.vilatuna@intsuperior.edu.ec)  
<https://orcid.org/0009-0005-9158-5052>

Jackson Martín Luzón-Maldonado <sup>III</sup>  
[martin.luzon@intsuperior.edu.ec](mailto:martin.luzon@intsuperior.edu.ec)  
<https://orcid.org/0000-0003-2349-2733>

Fidel David Parra-Balza <sup>IV</sup>  
[fideldavidparrabalza@gmail.com](mailto:fideldavidparrabalza@gmail.com)  
<https://orcid.org/0000-0003-2283-4551>

**Correspondencia:** [juan.chimarro@intsuperior.edu.ec](mailto:juan.chimarro@intsuperior.edu.ec)

Ciencias Técnicas y Aplicadas  
Artículo de Investigación

\* **Recibido:** 08 de agosto de 2023 \* **Aceptado:** 17 de agosto de 2023 \* **Publicado:** 28 de septiembre de 2023

- I. Instituto Superior Tecnológico Nelson Torres, Ecuador.
- II. Instituto Superior Tecnológico Nelson Torres, Ecuador.
- III. Instituto Superior Tecnológico Nelson Torres, Ecuador.
- IV. Investigador Independiente, Ecuador.



## Resumen

La presente investigación llevó a cabo el control de un módulo robótico mediante el uso del Cloud Computing y una App Móvil y se simplificó la integración de aplicaciones de control remoto por medio de tecnologías de la Industria 4.0. Para ello, se realizó una revisión de la literatura, informes y sitios web especializados en el área de investigación, y se utilizó las APIs y los servicios de Google Cloud Platform. Como resultado, se logró la interacción directa entre el módulo robótico y la App móvil mediante el uso de una tarjeta Wi-Fi con el estándar 802.11g y protocolos de telemetría como el MQTT, lo que permitió tener control sobre todas las variables del robot, transmisión de información a la base de datos, seguridad de la información y reporte de estadísticas sobre el uso de la aplicación. La conclusión del proyecto involucró la evaluación del funcionamiento del control del módulo robótico, analizando información como el tráfico tele-transmitido, los tiempos de latencia, los errores y los eventos, lo que permitió verificar el rendimiento de la aplicación.

**Palabras Clave:** Cloud Computing; App móvil; Módulo robótico; APIs; Google Cloud Platform.

## Abstract

The present research carried out the control of a robotic module through the use of Cloud Computing and a Mobile App and the integration of remote control applications was simplified through Industry 4.0 technologies. To do this, a review of the literature, reports and websites specialized in the research area was carried out, and the APIs and services of Google Cloud Platform were used. As a result, direct interaction between the robotic module and the mobile App was achieved through the use of a Wi-Fi card with the 802.11g standard and telemetry protocols such as MQTT, which allowed control over all the variables of the robot, transmission of information to the database, information security and reporting of statistics on the use of the application. The conclusion of the project involved the evaluation of the control operation of the robotic module, analyzing information such as tele-transmitted traffic, latency times, errors and events, which allowed verifying the performance of the application.

**Keywords:** Cloud Computing; Mobile App; robotic module; APIs; Google Cloud Platform.

## Resumo

A presente pesquisa realizou o controle de um módulo robótico através do uso de Cloud Computing e um aplicativo móvel e a integração de aplicativos de controle remoto foi simplificada através de tecnologias da Indústria 4.0. Para isso, foi realizada uma revisão de literatura, relatórios e sites especializados na área de pesquisa, sendo utilizadas as APIs e serviços do Google Cloud Platform. Como resultado, a interação direta entre o módulo robótico e o aplicativo móvel foi conseguida através da utilização de uma placa Wi-Fi com padrão 802.11g e protocolos de telemetria como MQTT, que permitiram o controle de todas as variáveis do robô, transmissão de informações para o banco de dados, segurança da informação e reporte de estatísticas sobre o uso do aplicativo. A conclusão do projeto envolveu a avaliação do funcionamento do controle do módulo robótico, analisando informações como tráfego teletransmitido, tempos de latência, erros e eventos, o que permitiu verificar o desempenho da aplicação.

**Palavras-chave:** Computação em Nuvem; Aplicativo móvel; módulo robótico; APIs; Plataforma Google Cloud.

## Introducción

En muchos casos, la integración de sistemas remotos en tiempo real puede resultar complicada y costosa, por lo que el uso de estas tecnologías puede ayudar a mejorar la eficiencia y la accesibilidad de los sistemas de control remoto. Además, esto puede beneficiar a una amplia variedad de industrias donde el uso de robots y sistemas remotos es cada vez más común.

El control de un módulo robótico mediante el uso del Cloud Computing y una App Móvil, se realiza para optimizar recursos, minimizar procesos de integración y cumplir con los estándares de calidad y servicio. Para lo cual se realizó la revisión de trabajos relacionados con la presente investigación. En ese sentido, el trabajo de Guzmán Luna et al. (2014), explora alternativas de servicios web como el SOAP (*Simple Object Access Protocol*) que es para el control del dispositivo mediante un servidor y el Restfull, que se resalta la simplicidad, la capacidad de transmitir datos a través de protocolos HTTP, a través del cual los servicios web habilitan la integración de usuario-servidor-robot y el modelo de teleoperación, que está fundamentado en los criterios de escalabilidad y versatilidad del software.

En otro estudio realizado por Manzi et al. (2016), se describe una robótica de nube genérica sistema de teleoperación que permite controlar en tiempo real un robot conectado con una red 4G que tiene

su transmisión de video como retroalimentación. El sistema propuesto se basa en la nube de Plataforma Azure. Los resultados que obtuvieron mostraron que la tecnología elegida permite tener actuaciones en tiempo real en términos de video y comandos de velocidad de transmisión.

De la misma forma en el trabajo de Alhomsi et al. (2017), propone implementar un sistema de simulación para médicos en tiempo real, el procedimiento requirió robustez y coordinación en la arquitectura de red por el cual se hizo uso del IoT (*Internet of Things*), el servicio Firebase como Hub y es el método de comunicación que permite que se mantenga en operación incluso si alguna unidad en la red tiene una mala conexión y además pueda ser fácilmente integrado con el sistema actual y establecer condiciones de la vida real.

Se revisó el estudio de Toquica et al. (2019), presenta los desarrollos recientes de los conceptos de la Industria 4.0, como sistemas ciberfísicos (CPS), internet de las cosas (IoT) y Cloud Computing, existe la necesidad de robustez y eficiencia en arquitecturas compatibles con cualquier proveedor de robótica industrial. Los enfoques actuales que generalmente se utilizan en sistemas robóticos basados en la nube, toma el concepto de software como servicio (SaaS) para difundir servicios en orden y teleoperar un robot, estos servicios se realiza por el usuario remoto a través de una programación de aplicaciones de interfaz (API) desarrollada en una programación de lenguaje de código abierto.

De igual manera Gavilán (2019a) afirma que los sistemas de información son un conjunto de hardware y software que proporcionan una funcionalidad, una solución, pero en el fondo, se tiende a centrar en el software porque es el elemento diferencial, lo que hace a un sistema realmente distinto de otro. La integración de sistemas, si solo se tratase de relacionar dos sistemas el problema sería de moderadas dimensiones: se establece esa interfaz y punto. Pero cuando se habla de decenas o centenas de sistemas diferentes se genera una explosión combinatoria de interfaces que reclama una solución más global.

Para contrastar la descripción de los sistemas integrados en tiempo real Garrell and Guilera (2018a), mencionan que, si se dispone de un dispositivo con conexión a internet se puede saber, desde cualquier punto del planeta donde se encuentre, en qué horario tiene un transporte público, reservar una plaza en un hotel, ver las obras maestras de un museo prestigioso, escuchar un concierto de música, leer una tesis doctoral de un científico, teleoperar un dispositivo o máquina desde la otra punta del mundo y todo eso en tiempo real, ya que la telefonía inalámbrica ya lleva cuatro generaciones y va por la quinta (5G).

Las aplicaciones de estos sistemas integrados en tiempo real en la actualidad marcan tendencias en el mundo digital, y así se plantea Garrell and Guilera (2018b) y menciona que los pilares del progreso tecnológico, muchos de los avances tecnológicos que constituyen la base de la industria 4.0, transformarán dramáticamente la producción: las celdas aisladas y poco optimizadas se convertirán en un flujo de producción totalmente integrado, automatizado y llevarán a la fábrica a una mayor eficiencia y productividad. Las interrelaciones tradicionales entre proveedores, productores y clientes experimentarán cambios importantes, así como las relaciones entre humanos y máquinas.

Los sistemas robóticos teleoperados actualmente están aplicados en varias áreas tales como la industria, salud, investigación y así lo describe en Barrientos et al. (2007), plantea que las tecnologías de teleoperación tiene una apertura a nuevos campos de aplicación, como el espacial, la cirugía, nuclear y submarino, con la ayuda de nuevos desarrollos en computadores digitales y la integración de tecnologías propias de la robótica y la inteligencia artificial. Se desarrolla, además, nuevos elementos que mejoran la eficiencia de la manipulación, como pueden ser novedosos diseños de dispositivos de entrada y nuevas técnicas de control, como puede ser el control supervisado.

Para este trabajo de investigación se describió las tecnologías que se utilizaron para el desarrollo de la teleoperación de un brazo robótico, tales como el Cloud Computing, Internet de las cosas (IoT), internet industrial de las cosas (IIoT), robots industriales. Por ello Cerón (2005) describe que el Cloud Computing (computación en la nube), no es más que trasladar los servidores con las aplicaciones corporativas desde máquinas aisladas en las empresas o en centros de proceso de datos corporativos a unos centros de procesos de datos ofrecidos por terceros y a los cuales se accede a través de Internet, a través de la nube.

Gavilán (2019b) explica que los componentes de los Sistemas robóticos teleoperados se componen principalmente de una estación de teleoperación, un sistema de comunicación y esclavo, el esclavo puede ser un manipulador o un robot móvil equipado con un manipulador ubicado en un entorno remoto. La estación de teleoperación permite controlar al esclavo a distancia por medio del sistema de comunicación, el cual permite transmitir las señales de control hacia el esclavo y, a su vez, recibir señales de información sobre el estado de éste en la estación de teleoperación a través de un canal de comunicación que puede ser una red de computadores, un enlace de radio frecuencia o microondas.

El crecimiento tecnológico en la industria 4.0 ha permitido, que gran parte de los dispositivos electrónicos puedan conectarse a la red de Internet e intercomunicarse entre sí; y de igual manera se ha incrementado nuevos servicios en áreas de la teleoperación, que forman parte de los sistemas controlados de forma remota en tiempo real. Sin embargo, este proceso requiere la utilización de diferentes plataformas, cada una de ellas con una gran cantidad de librerías, las cuales requieren un trabajo exhaustivo y complejo para poder integrarlas. Esto causa limitantes de compatibilidad, escalabilidad y versatilidad entre el software y el hardware.

La actual investigación tiene como propósito desarrollar un sistema teleoperado para un brazo robótico mediante la integración de diferentes librerías, protocolos de comunicación, en conjunto con dispositivos móviles ya que es importante abordar desafíos exigentes de tal forma que puedan recopilar información sobre el manipulador robótico, verificar el rendimiento y optimizar costos de implementación. Por tal razón es de gran interés la investigación y el desarrollo tecnológico de estos sistemas actuales.

Finalmente se verificó que los sistemas robóticos teleoperados en tiempo real mediante la plataforma Firebase y protocolos de comunicación IoT, son viables ya que se optimizan los recursos económicos para su desarrollo e implementación, y cumple además con criterios como la interacción con dispositivos móviles, escalabilidad, versatilidad, baja latencia, robustez, control remoto, compatibilidad de hardware, autenticación, historial del número de usuarios, registro sobre la cantidad de eventos que son los mandos de teleoperación que se envía hacia la nube, la recopilación de datos se obtuvieron a través de Google Cloud Platform.

### **Metodología**

En este estudio se utilizó el enfoque cuantitativo ya que permitió analizar las variables del control del módulo robótico, el diseño de investigación fue experimental por que se evaluó el control del sistema robótico y como método se usó la observación indirecta, pues se analizaron los datos recopilados por otros investigadores, esto se hizo mediante la lectura de trabajos. Es decir, esta recolección se apoyó en libros, revistas, informes, grabaciones, fotografías, etc., relacionadas con lo que se investiga. Se toma en cuenta esta técnica como levantamiento de información, ya que se realiza un análisis de la situación actual sobre el impacto tecnológico de la industria 4.0 y su relación con el control de módulos robóticos (Valencia, 2005).

Por otra parte, en la metodología de desarrollo de aplicaciones de software de Drake (2008), se ajustó de manera apropiada al objetivo de esta investigación. Y se describe a continuación: primeramente, se definieron las especificaciones del control del módulo robótico. Luego, se estructuró el esquema general del hardware y software que se utilizó en el proyecto de investigación. Después se conformó el organigrama general del control. Se realizó la adaptación entre el hardware y el software. Se constituyeron los ordinogramas modulares y se seleccionó la plataforma en el cual se desarrolló la codificación del módulo robótico para el desarrollo del software. Así también, se llevó a cabo la implementación del hardware con el software, se detalla las hojas de especificaciones de los componentes y protocolos de comunicación que permitieron la implementación del sistema integrado en tiempo real. Finalmente, se integran todos los dispositivos con sus respectivos protocolos de comunicación y los programas que se compilan, para llevar a cabo el funcionamiento general del módulo, y realizar las pruebas de rendimiento, latencia e integridad de la información.

## Resultados

### Esquema general del hardware y especificaciones del control del módulo robótico

Las interconexiones se realizan de tal manera que, cada una de las partes elementales del sistema como es el módulo robótico y las interfaces de control del dispositivo móvil, se conectan al servidor en la nube implementado en Firebase, ver Figura. 1.

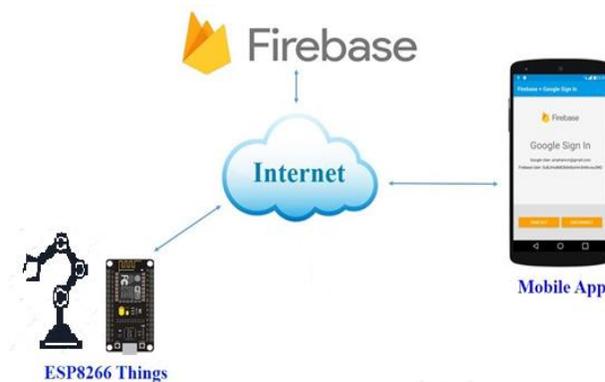


Figura 1. Esquema general del hardware

El organigrama general plantea la arquitectura de red para la comunicación entre hardware y software, lo cual garantiza que la información se envía y recibe de forma correcta, se detalla en la Figura. 2.

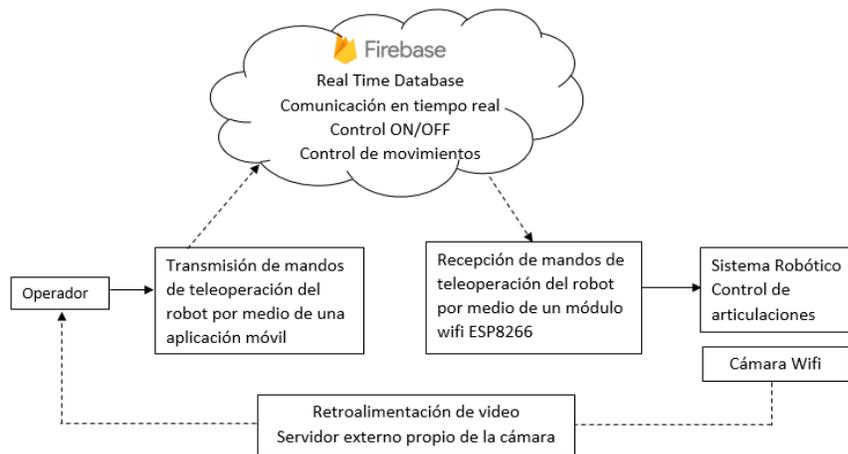


Figura 2. Ordinograma general del sistema teleoperado

### Arquitectura de red para la comunicación

La arquitectura de red está conformada por el brazo robótico con cinco grados de libertad, controlado por un módulo WiFi ESP8266 necesario para al servicio Firebase el cual gestiona la información del estado del robot en tiempo real, también cuenta con una aplicación móvil compatible con el sistema operativo Android para la teleoperación y manejo de la información sobre el estado del robot en tiempo real. Adicional a esto se incorporó una cámara WiFi para obtener una retroalimentación del entorno de trabajo. Ver la Figura. 3.

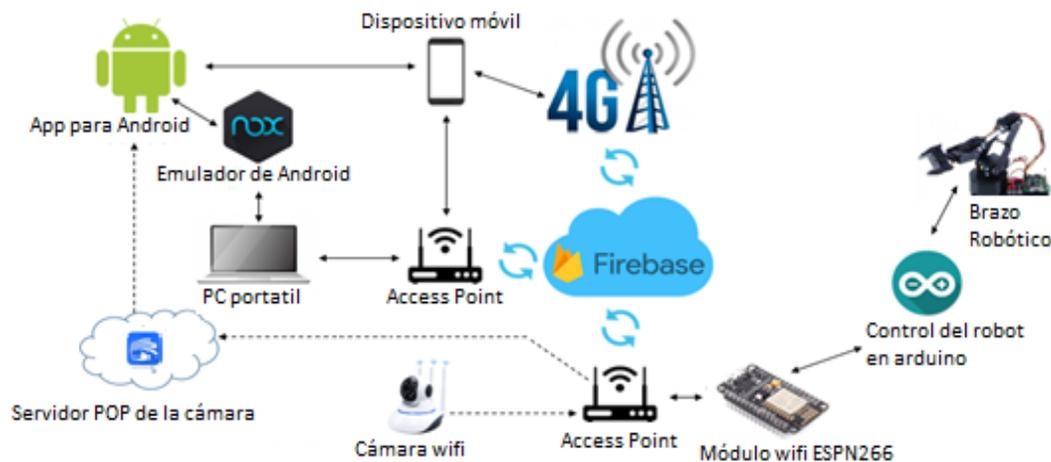


Figura 3. Arquitectura de comunicación entre hardware y software

### Ordinogramas modulares y codificación del control del módulo robótico

En esta fase cada uno de los siguientes bloques del diagrama de flujo se codifican individualmente con el software seleccionado, para asegurar que cada parte realice el trabajo en forma eficiente y segura. Se agrega la lógica *Backend* a través de Firebase el cual contiene varios servicios como

son: *Authentication*, *Realtime Database*, *Google Analytics*, *Cloud Functions* y *Google App Engine*. Se implementa una aplicación, una base de datos, se procesa la información y se envía al usuario. Los datos se transmiten y almacenan en tiempo real. La sincronización es automática entre las herramientas de interfaz de usuario en los dispositivos, que son los *Frontend*. Ver la Figura. 4.

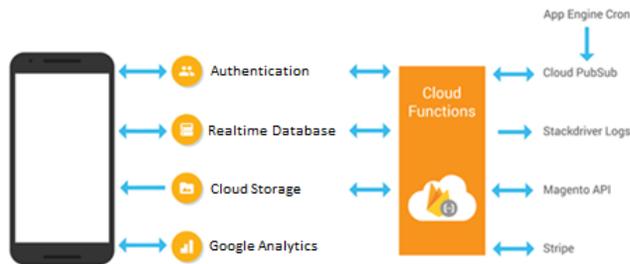


Figura 4. Lógica de comunicación Firebase Android

Fuente: Adaptado por DeRemer (2017): Hamilton App Takes the Stage, (<https://www.googblogs.com/hamilton-app-takes-the-stage/>)

Las herramientas que se usan para realizar las aplicaciones de *Cloud Computing* en el entorno flexible de App Engine es una plataforma de aplicaciones que supervisa, actualiza y escala el entorno de hosting; donde solo se tiene que escribir el código del servicio de Backend móvil. El entorno ejecuta el servicio de Backend dentro de contenedores de Docker que se pueden configurar. Esto significa que se puede llamar a binarios nativos, escribir en el sistema de archivos y hacer otras llamadas al sistema. Se puede procesar datos del usuario u organizar eventos, ampliar Firebase con el entorno flexible de App Engine, además se consigue el beneficio de poder sincronizar de forma automática datos en tiempo real hacia diferentes multiplataforma como es Android, iOS y web, sin la necesidad de ejecutar el código dentro de la zona de pruebas de App Engine, (Firebase, 2019) ver Figura. 5.

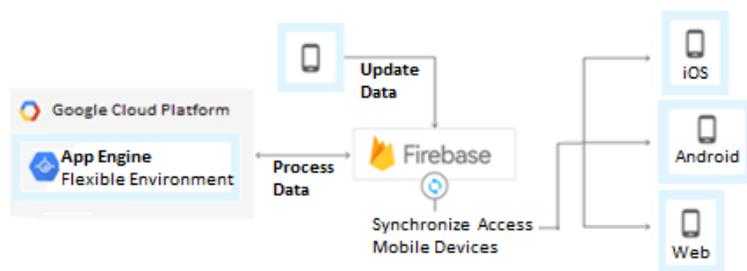


Figura 5. Entorno flexible App Engine

Cabe destacar que se usó el Sistema Operativo Android para instalar la aplicación móvil que realiza el control de mandos teleoperados en tiempo real y se desarrolló en la plataforma de Android Studio

por su compatibilidad con la herramienta Firebase, así también se usa el *Realtime Database* de Firebase porque la base de datos es alojada en la nube. Otra herramienta que se usa de Firebase es el *Authentication* porque permite añadir seguridad sobre la información en la aplicación al guardar los datos en la nube. Ver la Figura. 6.



Figura 6. Seguridad con Firebase

Fuente: Adaptado por Stevenson (2019): Patterns for security with Firebase: offload client work to Cloud Functions, Medium, (<https://medium.com/firebase-developers/patterns-for-security-with-firebase-offload-client-work-to-cloud-functions-7c420710f07>)

Para esta investigación se usó el Node MCU V3 el cual se basa en el chip WiFi™ ESP8266, por lo que se puede usar para aplicaciones IoT de manera muy fácil, mediante unas dependencias y librerías para conectarse directamente a la plataforma de Firebase y acceder a todas sus funciones, ver Figura. 7.

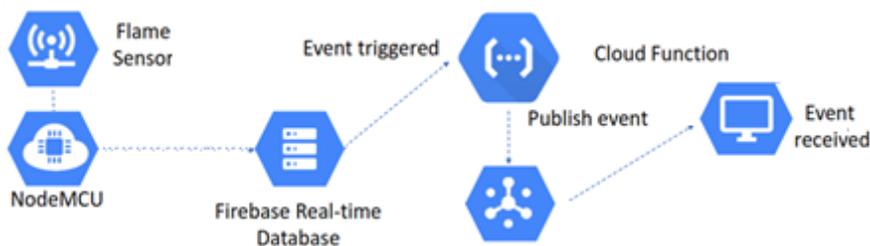


Figura 7. NodeMCU y Firebase

Fuente: Adaptado por Yi Lim (2017): IoT Google Cloud Functions with Firebase, SlideShare, (<https://www.slideshare.net/LimYi1/iot-google-cloud-functions-with-firebase/27>)

### Especificaciones técnicas del hardware del control del módulo robótico

El módulo robótico en el que se implementó el control tiene cinco grados de libertad y brinda un movimiento rápido y preciso, su estructura está compuesta por hombro plano, codo, movimiento de muñeca y una pinza de 2 grados para agarre funcional, ver figura 8.



Figura 8. Módulo robótico con cinco grados de libertad

Se usó la placa WiFi Node MCU V3-ESP8266 compatible con el IDE de Arduino, que puede conectarse con relativa facilidad al ordenador. Se utilizó un SmartPhone con S.O. Android ya que permiten la instalación de programas para incrementar sus posibilidades, como el procesamiento de datos y la conectividad a internet. Finalmente se usó una cámara WiFi para un feedback visual en la teleoperación, ya que permite observar en tiempo real al brazo robótico de forma remota a través de una aplicación móvil.

### **Desarrollo del software de la aplicación móvil y controlador del módulo robótico**

En esta fase se desarrolló, se realizaron las pruebas de funcionamiento y se depuró tanto el programa de la aplicación móvil que transmite los mandos de teleoperación como el programa del controlador del módulo robótico que receipta los mandos de teleoperación hasta que el funcionamiento sea adecuado. La base de datos *Realtime Database* es sin duda uno de los servicios más populares de la plataforma Firebase, ya que cuenta con la capacidad de almacenar datos en la nube, sin necesidad de preocuparse por toda la infraestructura de servidor. No obstante, es necesario crear y configurar el nuevo proyecto y asociarlo a la nueva aplicación.

Para el control de los movimientos del robot se crea un nodo y sus elementos. Se asigna la clave Teleoperación y añaden sus elementos como son: la base, codo, hombro, muñeca pitch, muñeca yaw, las cuales representan el valor del ángulo en grados de cada articulación, el gripper con el que se va a controlar la herramienta de trabajo del robot sería el abierto o cerrado, el estado del robot es encendido o apagado y finalmente se asigna un nombre. Así mismo, se inicia con el desarrollo de la interfaz de la aplicación móvil, se crea un layout para la pantalla principal, donde se mostrará los mandos de control del módulo robótico para la teleoperación.

También se realizan las configuraciones de Firebase en la clase principal del proyecto, se crea en primer lugar una referencia a la base de datos, la cual se podría entender como un indicador que apunta a un lugar concreto de la base de datos. Aunque es posible definir una referencia al elemento raíz de la base de datos, casi nunca se necesita acceder a la totalidad de los datos, sino que basta con un fragmento o sección concreta.

Para iniciar la aplicación móvil, se implementó un *login* para ingresar a los mandos de control para la teleoperación a través de la herramienta *Firebase Authentication*, de esta manera se tendrá un alto nivel de seguridad. A continuación, se muestra los métodos de acceso hacia la aplicación, los cuales pueden ser: correo/contraseña, *Google* o *Facebook*, finalmente se empieza el desarrollo de la actividad dentro de la aplicación móvil, y se agregan los elementos dentro del *login* para el diseño.

### Integración del hardware con el software según los protocolos de comunicación

Para continuar se prueba la comunicación entre el dispositivo móvil y la plataforma Firebase y se realiza la transmisión entre el controlador del robot y la plataforma Firebase de tal forma que se pueda confirmar una correcta correspondencia entre ambos para que el sistema interactúe de manera eficiente. Para integrar la teleoperación se usó los protocolos de comunicación IoT, por su versatilidad en los sistemas de comunicación, ya que es soporte central del IoT. Sobre los protocolos de esta arquitectura en la nube el *Cloud IoT Core* admite dos protocolos para la conexión y comunicación del dispositivo: MQTT y HTTP. Los dispositivos se comunican con *Cloud IoT Core* a través de un puente, ya sea el por el MQTT o el HTTP, ver figura 9.

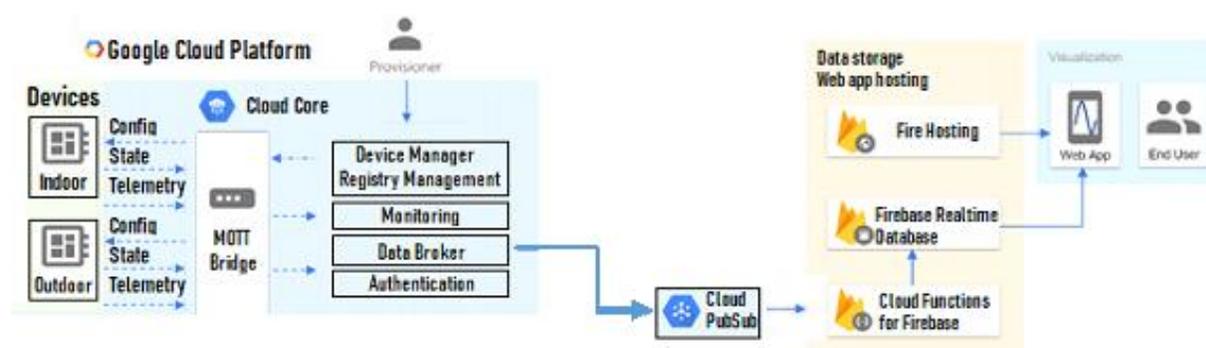


Figura 9. Arquitectura del proyecto

Fuente: Adaptado por Lourme (2021): How to check the weather using GCP-Cloud IoT Core with ESP32 and Mongoose OS, Medium, (<https://medium.com/free-code-camp/gcp-clouddotcore-esp32-mongooseos-1st-5c88d8134ac7>)

En la tabla 1 se detalla las características generales de cada protocolo como se muestra a continuación:

Tabla 1. Características del protocolo MQTT y HTTP

MQTT	HTTP
Menor uso de ancho de banda	Peso ligero (fácil de comenzar; comandos de curvatura simples)
Baja latencia	Menos problemas de firewall
Mayor rendimiento	Los datos binarios deben estar codificados en base64, lo que requiere más recursos de red y CPU
Admite datos binarios sin procesar	

Fuente: Adaptado por Google Cloud Platform (2020): Google.: IoT Protocols, Google Cloud, (<https://cloud.google.com/iot/docs/concepts/protocols>)

Para este proyecto se usó el protocolo MQTT de *Google Cloud IoT Core* como *broker*, ya que interactúa con dispositivos ESP8266 y el *Cloud Functions* de la plataforma Firebase que a su vez es compatible con aplicaciones móviles.

Una vez que se culminó con el desarrollo de un sistema distribuido de baja latencia, la teleoperación del módulo robótico fue mediante la plataforma Firebase con el uso de dispositivos móviles. Se realizó las pruebas de rendimiento, latencia e integridad de la información y se obtuvo los resultados finales. El resultado del sistema fue óptimo. Se obtuvo un historial del número de usuarios, se registró satisfactoriamente el número de eventos que fueron los mandos de teleoperación que se enviaron hacia Firebase. Los datos se obtuvieron a través de *Google Analytics* y *Google Cloud Platform*.

Con respecto a la información sobre el uso de la base de datos en *Realtime Database* de Firebase se obtuvo los siguientes resultados: el número de conexiones simultáneas, almacenamiento de la base de datos, carga y descarga de datos. Esta información se recopiló a través de la consola de Firebase. En la tabla 2 se detalla cada parámetro del uso de la base de datos.

Tabla 2. Resultados del Realtime Database

Parámetro	Descripción	Resultado
Conexiones	Conexiones simultáneas en modo prueba máx. 100 dispositivos.	3 conexiones

Almacenamiento	La cantidad de datos que se almacenan, sin incluir el hosting de Firebase ni los datos almacenados mediante sus otras funciones.	Almacenamiento
Descarga	Cuenta de descargas para todos los bytes descargados de tu base de datos, incluidos los gastos del protocolo de conexión y la encriptación SSL en intervalos de 1 minuto.	Descarga
Parámetro	Descripción	Resultado
Carga	Porcentaje de la base de datos que está ocupado en el procesamiento de solicitudes (ya sea conexiones en tiempo real o solicitudes de la API de RESTful) durante intervalos de 1 minuto. Es posible que ocurran problemas de rendimiento al acercarse al 100%.	51%

En cuanto a la información del estado y entorno de trabajo del brazo robótico se usó una cámara WiFi como retroalimentación de video en tiempo real, a través del servidor propio de la cámara.

En la tabla 3 se presenta los resultados del tiempo promedio de retardo que existe entre atributos con el protocolo MQTT. Se consideró: nombre del robot, base, hombro, codo, pitch, yaw, gripper y estado del robot.

Tabla 3. Retardo que existe entre atributos con el protocolo MQTT.

Iteración de datos por nodo	Promedio del tiempo de retardo entre cada atributo (ms)
1	712,286
2	660,375
3	563,000
4	601,750

5	460,375
Promedio	599,557

Con respecto a la tarjeta ESP8266 se usó el protocolo MQTT y se analizó el tiempo promedio de retardo que existe entre atributos. El tiempo promedio de retardo de cada atributo al momento de generar un evento teleoperado, fue un promedio de 599,557 milisegundos como retardo entre cada atributo lo cual representó un tiempo de latencia aceptable ya que la red WiFi de casa tenía un ancho de banda de 4Mbps, y este tiene un tiempo promedio de carga y descarga de 85 ms.

Por otra parte, el tiempo promedio de retardo del evento que se produce por cada variable del brazo robótico es de 3503,429 milisegundos. Sin embargo, para el sistema teleoperado también se puso a prueba con una red móvil 4G con un ancho de banda de 10Mbps y un tiempo promedio de carga y descarga de 38 milisegundos para tener la comunicación de la manera óptima posible.

Finalmente se obtuvo los resultados de la API sobre el tráfico tele-transmitido, el porcentaje de errores de ingreso a los controles de teleoperación y los tiempos de latencia mediante el reporte general de *Google Cloud Platform*. Observar la tabla 4, los tiempos de latencia aproximados que se midieron en la interfaz serial de Arduino, con respecto al tráfico de datos del *Cloud Functions API* con Firebase, fueron tiempos de latencia entre cada atributo de 208,896 milisegundo. El tiempo de latencia por evento de cada atributo fue de 35232,154 milisegundos. También se presentó un 20,635% de errores de ingreso con el *Authentication* de Firebase.

Tabla 4. Reporte de resultados de API y servicios en Google Cloud Platform

Nombre	Solicitudes	Errores (%)	Latencia mediana (ms)	95% de latencia (ms)
Identity Toolkit API	107	0		
Token Service API	63	20,635	47,982	117,497
Cloud Functions API	19	0	208,896	35232,154
Firebase Extensions API	1	0	196,608	255,59
Firebase Moods API	1	0	98,304	127,795

La arquitectura de *Cloud Computing* aportó una reducción de recursos y costos ya que toda la infraestructura de la teleoperación se implementó a través de las *APIs* y *Services* de *Google Cloud Platform*. Solo se necesitó tener una cuenta de Gmail, dicha plataforma actuó como *broker* lo que permitió una interacción directa entre las herramientas de Firebase para el control de los mandos

teleoperados por medio de dispositivos móviles. Todo este proceso también permitió conectarse de forma remota a tarjetas de control que se conectaron a la red de Internet. En este caso se usó una ESP8266 ya que, permitió controlar las acciones de teleoperación del módulo robótico a través del protocolo MQTT, los cuales son muy utilizados en aplicaciones IoT. Para este sistema distribuido de baja latencia se obtuvo un menor uso de ancho de banda, baja latencia y mayor rendimiento. En cuanto a la información del estado y entorno del brazo robótico se usó una cámara WiFi como retroalimentación de video a través de un servidor propio de la cámara.

Tabla 5. Comparación del control en módulos robóticos mediante el uso del Cloud Computing

Parámetros	P1	P2	P3	P4	P5	P6	P7
Esta investigación	✓	✓	✓	✓	✓	✓	✓
Toquica et al. (2019) (1)	✓		✓		✓		
Dawarka and Bekaroo (2018) (2)	✓		✓		✓		
Wang et al. (2015)	✓	✓		✓	✓	✓	
Liu et al. (2018) (3)	✓	✓		✓	✓	✓	
Chen and Tsai (2017) (4)	✓			✓	✓	✓	
Smirnov and Kashevnik (2018) (5)			✓		✓	✓	

P1 Seguridad

P2 Software libre

P3 Compatibilidad del Sistema robótico

P4 Conectividad

P5 Aplicación de robots industriales

P6 Validación real con robots

P7 La solución no depende de: RSNP(1); ROS(2); MATLAB (3); Microsoft Azure (4);

Cloudminds Cross-Border (5)

La aplicación móvil que se desarrolló tuvo la capacidad de conectarse a la red de Internet lo cual permitió enlazar los controles y mandos del brazo robótico desde casi cualquier punto del planeta en tiempo real. La simplicidad de la aplicación permitió una gran usabilidad, y contó con un potente sistema de seguridad ya que permitió autenticar a los usuarios.

Para la gestión del proceso de integración de la teleoperación como gestor se usó Firebase, ya que contiene herramientas como: *Cloud Functions API*, *Realtime Database*, *Authentication*. Esto

permitió tener el control de todas las variables del robot de forma remota con tiempos de baja latencia, carga y descarga de información de la base de datos, seguridad de la información y estadísticas sobre el uso de la aplicación. Además, que se tuvo compatibilidad de hardware.

Se valoró el rendimiento del sistema robótico teleoperado a través de la información estadística que ofrece *Firebase Analytics*, *Google Analytics*, y *Google Cloud Platform*. Se midió el tráfico tele-transmitido, tiempos de latencia, errores producidos, número de eventos. Se verificó el cumplimiento de confiabilidad y versatilidad del sistema, ya que toda esta infraestructura está Dockerizada en Google.

Como recomendación se debe desarrollar una red de telecomunicaciones ya que va a depender de una WLAN y una red de movilidad 3G/4G, y se necesita que los tiempos de latencia sean estables, y tanto el operador como el sistema remoto no pierdan comunicación. Actualmente se puede probar este tipo de investigaciones sobre la tecnología 5G.

## Conclusiones

La integración de aplicaciones de teleoperación a través de tecnologías de la industria 4.0 como, por ejemplo, la arquitectura del *Cloud Computing*, las *APIs*, Servicios de *Google Cloud Platform* y los protocolos de comunicación IoT (*Internet of Things*), demostraron, según las pruebas de rendimiento e integridad de la información, un trabajo óptimo del sistema. Además, se realizó la comparación del control en módulos robóticos mediante el uso del *Cloud Computing*, se contrastaron los parámetros y los resultados con otras investigaciones y se obtuvo información sobre el tráfico tele-transmitido, un historial del número de usuarios, la cantidad de eventos que son los mandos de teleoperación que se envía hacia *Firebase* con un porcentaje mínimo de error. Se evidenció que la latencia depende en gran medida del tipo de tecnología de conexión a internet (3G/4G). Los datos obtenidos a través de *Google Analytics*, permitieron que los sistemas teleoperados sean viables mediante la utilización de plataformas que integraron los sistemas robóticos teleoperados con tiempos de baja latencia. Todo esto con un mínimo gasto de recursos económicos para su desarrollo. Además, se cumplió con criterios como la interacción con dispositivos móviles, escalabilidad, versatilidad, baja latencia, robustez, control remoto, compatibilidad de hardware para la verificación del rendimiento de la aplicación.

## Referencias

- Alhomsy, Y., Alsalemi, A., Disi, M., Ahmed, I., Bensaali, F., Amira, A., Alinier, G., 2017. Real-Time Communication Network Using Firebase Cloud IoT Platform for ECMO Simulation. <https://doi.org/10.1109/iThings-GreenCom-CPSSCom-SmartData.2017.31>
- Barrientos, A., Balaguer, C., Peñin, L., Aracil, R., 2007. FUNDAMENTOS DE ROBÓTICA, 2DA ed. Ingebook.
- Cerón, A., 2005. Sistemas robóticos teleoperados. Ciencia e Ingeniería Neogranadina, ISSN 0124-8170, No. 15, 20058 pags. 15. <https://doi.org/10.18359/rcin.1255>
- Chen, T., Tsai, H.-R., 2017. Ubiquitous manufacturing: Current practices, challenges, and opportunities. Robotics and Computer-Integrated Manufacturing, Special Issue on Ubiquitous Manufacturing (UbiM) 45, 126–132. <https://doi.org/10.1016/j.rcim.2016.01.001>
- Dawarka, V., Bekaroo, G., 2018. Cloud Robotics Platforms: Review and Comparative Analysis. <https://doi.org/10.1109/ICONIC.2018.8601220>
- DeRemer, D., 2017. Hamilton App Takes the Stage | [googblogs.com](http://googblogs.com). URL <https://www.googblogs.com/hamilton-app-takes-the-stage/> (accessed 6.19.23).
- Drake, J.M., 2008. Proceso de desarrollo de aplicaciones software.
- Firebase, 2019. Descripción general de la arquitectura de FCM | Firebase Cloud Messaging [WWW Document]. Firebase. URL <https://firebase.google.com/docs/cloud-messaging/fcm-architecture?hl=es-419> (accessed 6.19.23).
- Garrell, A., Guilera, L., 2018a. La Industria 4.0 en la sociedad digital - Marge Books.
- Garrell, A., Guilera, L., 2018b. La Industria 4.0 en la sociedad digital - Marge Books.
- Gavilán, I., 2019a. La Carrera Digital. ExLibric.
- Gavilán, I., 2019b. La Carrera Digital. ExLibric.
- Google Cloud Platform, 2020. Protocolos | Documentación de Cloud IoT Core [WWW Document]. Google Cloud. URL <https://cloud.google.com/iot/docs/concepts/protocols?hl=es-419> (accessed 6.20.23).
- Guzmán Luna, J.A., Torres Pardo, Í.D., Galeano Hincapié, P.A., 2014. Propuesta de un Modelo de Gestión de Servicios en la Nube para la Manipulación de Sistemas Robóticos con el uso de Dispositivos Móviles. Lámpsakos 43–51. <https://doi.org/10.21501/21454086.1373>

- Liu, Y., Wang, L., Vincent Wang, X., 2018. Cloud manufacturing: latest advancements and future trends. *Procedia Manufacturing*, Proceedings of the 8th Swedish Production Symposium (SPS 2018) 25, 62–73. <https://doi.org/10.1016/j.promfg.2018.06.058>
- Lourme, O., 2021. How to check the weather using GCP-Cloud IoT Core with ESP32 and Mongoose OS. We've moved to freeCodeCamp.org/news. URL <https://medium.com/free-code-camp/gcp-cloudiotcore-esp32-mongooseos-1st-5c88d8134ac7> (accessed 6.20.23).
- Manzi, A., Fiorini, L., Limosani, R., Sincak, P., Dario, P., Cavallo, F., 2016. Use Case Evaluation of a Cloud Robotics Teleoperation System (Short Paper). pp. 208–211. <https://doi.org/10.1109/CloudNet.2016.49>
- Smirnov, A., Kashevnik, A., 2018. Semantic Interoperability for Coalition Creation by Mobile Robots and Humans: an Approach and Case Study. *IFAC-PapersOnLine*, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018 51, 1409–1414. <https://doi.org/10.1016/j.ifacol.2018.08.319>
- Stevenson, D., 2019. Patterns for security with Firebase: offload client work to Cloud Functions. *Firebase Developers*. URL <https://medium.com/firebase-developers/patterns-for-security-with-firebase-offload-client-work-to-cloud-functions-7c420710f07> (accessed 6.20.23).
- Toquica, J.S., Benavides, D., Motta, J.M.S.T., 2019. Web Compliant Open Architecture For Teleoperation of Industrial Robots, in: 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE). IEEE Press, Vancouver, BC, Canada, pp. 1408–1414. <https://doi.org/10.1109/COASE.2019.8842847>
- Valencia, H.G., 2005. *MANUAL DE TECNICAS DE INVESTIGACION Conceptos y Aplicaciones*. Hector Guillermo Huaman Valencia.
- Wang, X., Alhusin Alkhdur, A., Wang, L., 2015. Cloud-based Robotic System: Architecture Framework and Deployment Models.
- Yi Lim, S., 2017. IoT Google Cloud Functions with Firebase [WWW Document]. URL <https://www.slideshare.net/LimYi1/iot-google-cloud-functions-with-firebase> (accessed 6.20.23).

© 2023 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)

(<https://creativecommons.org/licenses/by-nc-sa/4.0/>).