

Desarrollo e implementación de un emulador de Teach Pendant basado en Android para el control del robot SCARAUC virtual

Development and Implementation of an Android-based Teach Pendant emulator for controlling the Virtual SCARAUC robot

Ing. Karina Chiquinquirá Aguiar Minniti 
Universidad de Carabobo, Valencia, Venezuela
<https://orcid.org/0009-0000-7006-5924>

Ing. Antonio José Torres Sarmiento 
Universidad de Carabobo, Valencia, Venezuela
<https://orcid.org/0009-0007-1497-7129>

Resumen

El estudio presenta el desarrollo e implementación de un emulador de Teach Pendant basado en el sistema operativo Android, cuyo objetivo es proporcionar una herramienta funcional para controlar el robot virtual SCARAUC del Laboratorio de Robótica y Visión Industrial de la Universidad de Carabobo. Como fundamento, se realizó una investigación documental exhaustiva para determinar tanto los requerimientos del sistema como el modelo matemático de control, abordando los problemas cinemáticos directo e inverso. Metodológicamente, la arquitectura del software fue estructurada rigurosamente utilizando el modelo C4, y se implementó en el sistema operativo Android, seleccionando el SDK 4.4 (KitKat) para garantizar una alta compatibilidad con dispositivos móviles. La comunicación entre el emulador y el autómatas se estableció mediante el protocolo TCP/IP con una arquitectura Cliente-Servidor, utilizando un programa en LabView como intermediario para interactuar con las bibliotecas RoboTalk de RoboWorks. El resultado es una herramienta práctica, económica y accesible que se posiciona como una alternativa moderna para la robótica educativa. Finalmente, se proponen futuras mejoras, como la gestión de archivos externos, el manejo óptimo de trayectorias y la adaptación del sistema a un robot físico SCARA.

Palabras clave: emulador, teach pendant, android, robot.

Abstract

The study presents the development and implementation of an Android-based Teach Pendant emulator, aimed at providing a functional tool to control the virtual SCARAUC robot from the Robotics and Industrial Vision Laboratory at the University of Carabobo. As a theoretical foundation, exhaustive documentary for determining both the system requirements and the mathematical control model, tackling the direct and inverse kinematic problems. Methodologically, the software architecture was rigorously structured using the C4 model and implemented on the Android operating system, selecting SDK 4.4 (KitKat) to ensure broad compatibility with mobile devices. Communication between the emulator and the automaton was established through the TCP/IP protocol within a Client-Server architecture, employing a LabView program as an intermediary to interact with RoboWorks' RoboTalk libraries. The outcome is a practical, economical, and accessible tool positioned as a modern alternative for educational robotics. Finally, future improvements are proposed, including external file management, optimal trajectory handling, and adaptation of the system to a physical SCARA robot.

Keywords: emulator, teach pendant, android, robot.

Recibido: 20-04-2026

Aceptado: 06-07-2026

Introducción

En la actualidad, los robots cumplen un rol importante en los entornos industriales automatizados gracias al desempeño de diversos tipos de tareas, que van desde ejecución de actividades simples hasta tareas complejas y peligrosas. Para los procesos que requieran velocidad y precisión en etapas de ensamblaje y manipulación, los robots SCARA (Selective Compliance Articulated Robot Arm) son una buena alternativa. Por otra parte, un componente fundamental en los ecosistemas de la robótica es el *Teach Pendant*, este elemento es el encargado de actuar como interfaz para recibir las indicaciones que el autómatas debe realizar de acuerdo con las órdenes dadas por un operario.

En la actualidad, el Laboratorio de Robótica y Visión Industrial de la Universidad de Carabobo cuenta con un robot de configuración SCARA denominado SCARAUC, este por falta de componentes a nivel de software y hardware se encuentra inoperativo. La falta de herramientas funcionales limita al laboratorio en el desarrollo de actividades prácticas. Es importante resaltar, que el autómatas posee un modelo virtual que simula su comportamiento. Dentro de este marco, este trabajo se fundamentó en la elaboración y la implementación de un emulador de *Teach Pendant* basado en el sistema operativo Android, con el propósito de proporcionar una herramienta funcional al laboratorio para la realización de prácticas. Un aspecto que se puede acreditar a esta investigación es la de formar parte a una línea de investigación de la cátedra de Robótica y Visión Industrial, lo que le da un valor añadido por la implementación de un sistema moderno y viable desde el punto de vista económico.

Finalmente, se estableció como objetivo desarrollar un emulador del *Teach Pendant* bajo el sistema operativo Android, que, conforme al estudio de la cadena cinemática y a los requerimientos funcionales definidos para el sistema, actúa como la interfaz responsable de procesar las órdenes del operario para controlar el robot SCARAUC. Para la comunicación entre los distintos elementos del sistema se implementó el protocolo TCP/IP, garantizando la transmisión eficiente y sincronizada de los datos.

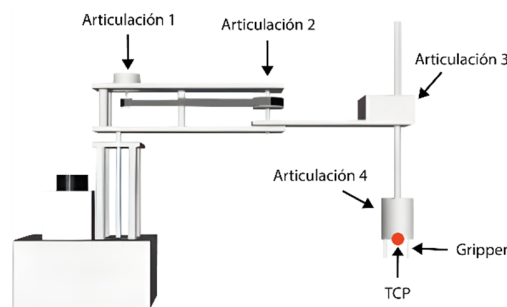
Metodología

Análisis cinemático del robot SCARAUC

Los robots industriales son herramientas funcionales presentes en diversos tipos de procesos que cumplen el objetivo de ejecutar las tareas que le sean asignadas. Tener la capacidad de controlar su accionar de acuerdo con algún método de programación es una cualidad intrínseca que deben tener los autómatas. Ahora bien, cuando se tiene la labor de desarrollar un elemento de esta naturaleza, es indispensable contar con el modelo matemático que rige su comportamiento. En este sentido, y de acuerdo con Olier Caparrosa (2020), “el análisis cinemático es uno de los primeros pasos para el diseño de un robot industrial” (p.17) por lo que, para desarrollar e implementar el emulador de Teach Pendant, fue necesario iniciar con el estudio de la cadena cinemática.

Para definir a las ecuaciones que rigen al autómat, se debe comenzar con la mención de los elementos fundamentales que componen su estructura mecánica, las cuales son, las articulaciones y eslabones, las primeras son las encargadas de permitir la movilidad y determinar la forma volumétrica de trabajo, y los segundos cumple con el rol de unir a estas articulaciones, dando la estructura física definitiva. Ahora bien, de acuerdo con la cantidad de articulaciones, sus características y como están dispuestas en el robot, además de los parámetros de los eslabones, podemos definir su respectivo modelo matemático. En la figura 1, se aprecia la representación virtual del autómat indicando las articulaciones que lo componen.

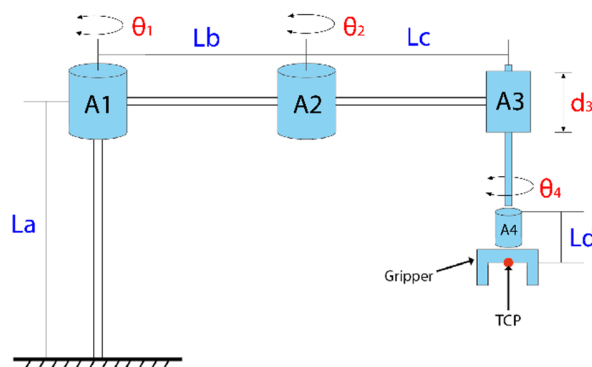
Figura 1
Representación virtual del robot SCARAUC



Fuente: Aguiar y Torres (2026)

Cabe señalar que las articulaciones rotativas 1 y 2 junto con la articulación prismática 3 son las encargadas de posicionar al punto central de herramienta denominado TCP (Tool Center Point), mientras que la articulación 4 es la encargada de dar la orientación definitiva del mismo. Como elemento final, el autómat tiene incorporado un *gripper* (pinza) para brindar al robot la posibilidad de manipular objetos.

Figura 2
Elementos y variables del robot SCARAUC



Fuente: Aguiar y Torres (2026)

Nuestro principal objetivo cuando se trata de controlar un brazo robótico es siempre saber dónde está ubicado el TCP, debido a que este representa a la posición del elemento final de control, que, a su vez, es el encargado de realizar la acción de interés final. De ahí que, para abordar el tema cinemático, se tuvo presente que los robots industriales persiguen dos propósitos generales, en primer lugar, se quiere saber cuál es la posición final del TCP luego de manipular sus articulaciones, a esta situación se le denomina problema cinemático directo, en referencia a Sanz (2013) se conoce con esta denominación al dilema de determinar la posición final del elemento terminal de un Robot Manipulador en relación de un sistema de referencia fijo.

Por otra parte, cuando se quiere colocar al TCP en un punto y con una orientación determinada en el espacio, surge la necesidad de determinar cuánto debe moverse cada articulación para lograr esa posición, esta situación es denominada problema cinemático inverso. Según también por el autor de la referencia antes citada, Sanz (2013) determinar el conjunto de valores de los parámetros θ y d de un robot, que satisfacen la necesidad de orientar y posicionar su herramienta en un punto y de un modo específico, es el planteamiento conocido como Problema Cinemático Inverso. Cabe mencionar, que los parámetros θ y d mencionadas en la cita anterior, se refieren a las variables de las articulaciones relacionadas a sus posibles trayectorias de las articulaciones rotativas y prismáticas respectivamente.

En la Figura 2, se ilustran los elementos que conforman al autómatas con las respectivas variables y parámetros implicados en el modelado geométrico. En relación con lo previamente mencionado, el robot cuenta con tres articulaciones rotativas A1, A2 y A4 y una articulación prismática A3. Las variables angulares θ_1 y θ_2 definen la posición en el plano horizontal de los primeros dos eslabones articulados, mientras que la variable θ_4 corresponde al giro del elemento final (gripper) sobre su propio eje.

Cabe destacar que, las variables de las articulaciones rotativas A1, A2 y A4, no son estrictamente secuenciales en el espacio tridimensional debido a la articulación prismática responsable del desplazamiento vertical d_3 , que se encuentra interpuesta físicamente en medio de la cadena cinemática, separando las articulaciones θ_1 y θ_2 encargadas del posicionamiento horizontal de la articulación θ_4 encargada de la orientación final del gripper.

Partiendo de esta disposición geométrica, y tal como lo establecen Adrián y Castañeda (2011), las ecuaciones 1, 2 y 3 corresponden a la cinemática directa, permitiendo determinar el punto en el espacio en coordenadas cartesianas del TCP:

$$Px = Lc \cdot \cos(\theta_1 + \theta_2) + Lb \cdot \cos(\theta_1) \quad (1)$$

$$Py = Lc \cdot \sin(\theta_1 + \theta_2) + Lb \cdot \sin(\theta_1) \quad (2)$$

$$Pz = La - Ld - d_3 \quad (3)$$

Asimismo, referente a la cinemática inversa se tienen las ecuaciones 4, 5 y 6:

$$\theta_2 = \pm \left(\frac{(px)^2 + (py)^2 - (Lc)^2 - (Lb)^2}{2 \cdot Lc \cdot Lb} \right) \quad (4)$$

$$\theta_1 = \pm \left(\frac{Lc \cdot (px \cdot \cos(\theta_2) + py \cdot \sin(\theta_2)) + px \cdot Lb}{(px)^2 + (py)^2} \right) \quad (5)$$

$$d_3 = -Ld - pz + La \quad (6)$$

Donde px , py y pz figuran como la coordenada cartesiana a la que se desea desplazar el TCP. Los valores La , Lb , Lc y Ld empleados en ambos casos, son siempre constantes y están relacionados a los parámetros de los eslabones.

Definición de requerimientos

A partir del análisis desarrollado, los requerimientos del sistema fueron definidos considerando tres aspectos principales. En primer lugar, aquellos relacionados con el modelo cinemático del robot, de donde se derivan funcionalidades como el movimiento independiente de cada articulación y el movimiento simultáneo a partir de posiciones articulares finales, asociados a la cinemática directa, así como el movimiento simultáneo a partir de coordenadas cartesianas, correspondiente a la cinemática inversa.

En segundo lugar, se establecieron requerimientos vinculados al esquema de comunicación implementado. En este sentido, el robot virtual SCARAUC fue modelado y simulado en el software RoboWorks, el cual dispone de librerías RoboTalk que permiten la interacción con sistemas externos. A partir de estas capacidades, se optó por integrar el software LabVIEW como intermediario, actuando como receptor de los comandos generados por el emulador mediante el protocolo TCP/IP, lo que dio origen al requerimiento de establecer comunicación bajo este esquema.

Finalmente, se incorporaron requerimientos orientados a la interacción con el usuario, tales como el almacenamiento y gestión de posiciones, la ejecución de tareas, el control de movimientos (inicio, pausa y detención), la visualización en tiempo real del estado del robot y la operación del sistema en ausencia de comunicación con el autómatas. Estas funcionalidades fueron definidas tomando como referencia interfaces reales de control robótico, particularmente teach pendants industriales, con el fin de garantizar una interacción adecuada, robusta y flexible. De este modo, se definieron los requerimientos mostrados en la Tabla 1.

Tabla 1

Requerimientos para el desarrollo de la App SCARAUC

Requerimientos funcionales
Mover de forma independiente cada articulación
Mover las articulaciones de forma simultánea a partir de las posiciones finales asignadas a cada articulación
Mover las articulaciones de forma simultánea a partir de las coordenadas cartesianas dadas
Establecer comunicación con el robot virtual SCARA-UC a través LabView mediante el protocolo TCP/IP
Posibilidad de almacenar posiciones en una lista para la ejecución de tareas del robot.
Posibilidad de guardar listas de posiciones en una base de datos interno del dispositivo, para luego ser reutilizado
Iniciar, pausar y detener los movimientos
Mostrar en tiempo real la posición del robot
Opción de actualizar y eliminar posiciones en listas almacenadas
Posibilidad de usar el emulador a pesar de la falta de comunicación con el autómat

Fuente: Aguiar y Torres (2026)

Diseño de la arquitectura software

Se implementó el modelo C4 como herramienta para estructurar la arquitectura del sistema, debido a su capacidad para representar de manera clara y jerárquica los diferentes niveles de abstracción de un sistema software. Este enfoque, creado por Brown (2015) en "The C4 model for visualising software architecture", permite organizar el sistema desde una visión general hasta los detalles técnicos mediante sus capas de contexto, contenedor, componente y código, facilitando la comprensión tanto del funcionamiento global como de la estructura interna del sistema desarrollado.

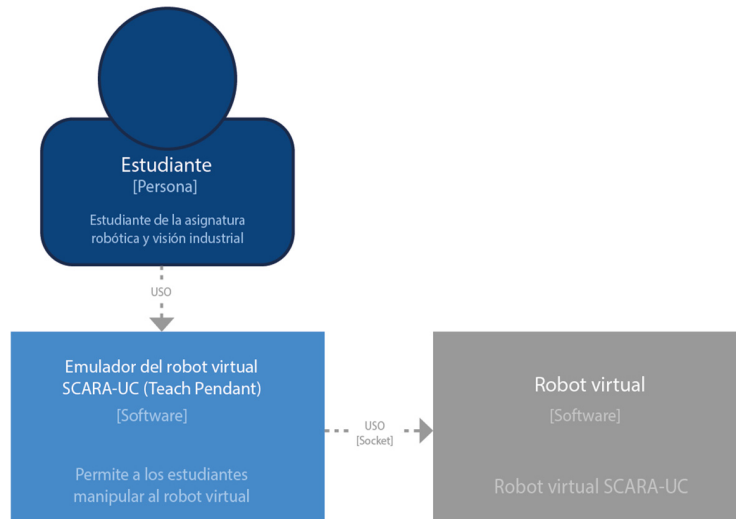
En la figura 3 se muestra el nivel de contexto, donde se representa de forma general el sistema y su interacción con el entorno. En este nivel se identifica al estudiante como el actor principal, quien manipula el emulador del Teach Pendant para enviar instrucciones al robot virtual. Este diagrama permite delimitar el alcance del sistema y comprender cómo se integra dentro del entorno de trabajo, sirviendo como punto de partida para la organización de los elementos internos.

Una vez definido el contexto, el sistema se organizó a nivel de contenedores, como se muestra en la figura 4, con el objetivo de identificar las principales unidades funcionales y tecnológicas. En este nivel se distinguen la aplicación móvil desarrollada en Android, encargada de gestionar la interacción con el usuario, y el programa en LabVIEW, que actúa como intermediario en la comunicación con el robot virtual en RoboWorks. La interacción entre estos elementos se estableció mediante el protocolo TCP/IP utilizando

el método de Sockets, lo que permite estructurar claramente la distribución de responsabilidades dentro del sistema.

Figura 3

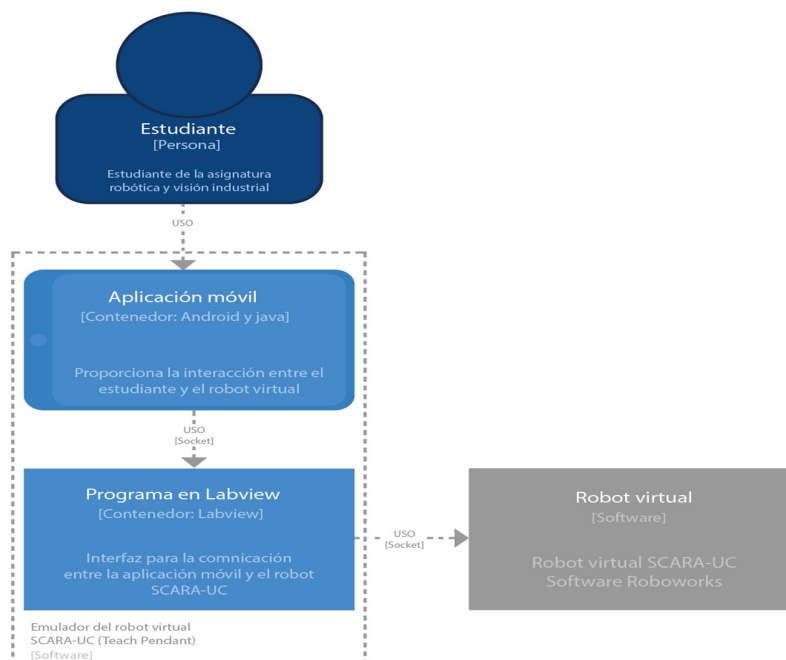
Elementos y variables del robot SCARAUC



Fuente: Aguiar y Torres (2026)

Figura 4

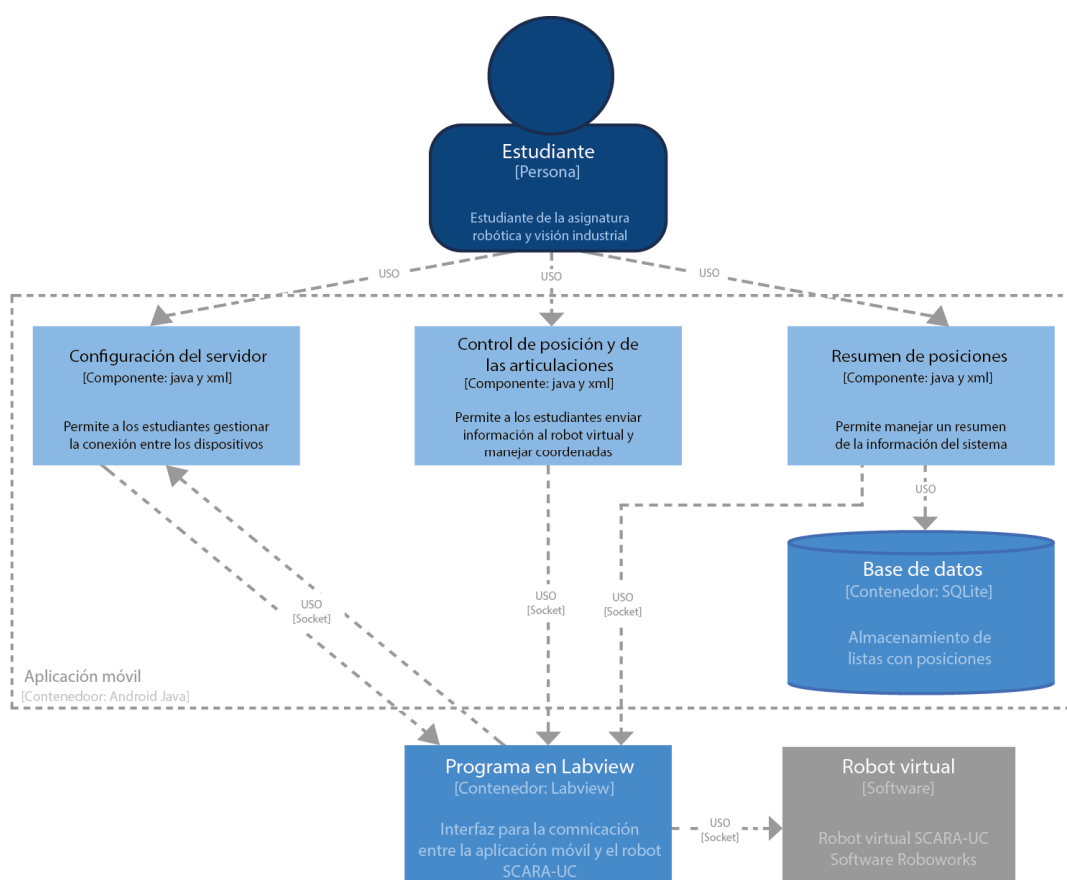
Elementos del sistema de emulación



Fuente: Aguiar y Torres (2026)

Finalmente, en la figura 5 se presenta el diagrama de componentes, el cual describe la organización interna de la aplicación móvil en función de sus requerimientos. En este nivel se detallan las interfaces y módulos que conforman el sistema, tales como la configuración del servidor, el control de posición y articulaciones, y el resumen de posiciones, así como el componente de base de datos encargado de la persistencia de la información. Esta descomposición permite evidenciar cómo se estructuró el sistema para cumplir con las funcionalidades definidas, manteniendo coherencia con los niveles superiores del modelo C4.

Figura 5
Componentes del sistema de emulación



Fuente: Aguiar y Torres (2026)

Implementación del sistema

Para el desarrollo de la aplicación se seleccionó Android como sistema operativo debido a su amplia adopción en dispositivos móviles y a las facilidades que ofrece para la creación de interfaces intuitivas y personalizables. Tal como manifiesta Meier (2015), Android proporciona un entorno de desarrollo sólido y flexible basado en Java y XML, lo

que favorece la construcción de aplicaciones modulares y escalables. Asimismo, su naturaleza abierta y su compatibilidad con una amplia gama de dispositivos lo convierten en una opción adecuada para aplicaciones de control remoto en entornos educativos y experimentales.

De acuerdo con trabajos previos que emplearon Android como plataforma para la manipulación de autómatas, como las propuestas de Lovon (2019) y Ramírez (2016), se ha demostrado la versatilidad de este sistema operativo en aplicaciones de control robótico, lo que respalda su selección en este trabajo. La implementación se realizó utilizando el Android SDK versión 4.4 (KitKat), garantizando compatibilidad con la mayoría de los dispositivos disponibles al momento del desarrollo. La lógica de la aplicación se programó en Java, mientras que las interfaces gráficas se diseñaron empleando XML, tal como se recomienda en la documentación oficial de Android (Android Developers, 2024), lo que permitió una clara separación entre la presentación y la funcionalidad del sistema, facilitando su mantenimiento y escalabilidad. Asimismo, se utilizó SQLite como motor de base de datos, dado que, según su documentación oficial (Hipp, 2023), proporciona un almacenamiento interno ligero, eficiente e integrado con el entorno Android, permitiendo gestionar de forma local la información generada por el emulador. Esta combinación de tecnologías aseguró un desarrollo eficiente, estable y adaptable a distintas resoluciones de pantalla.

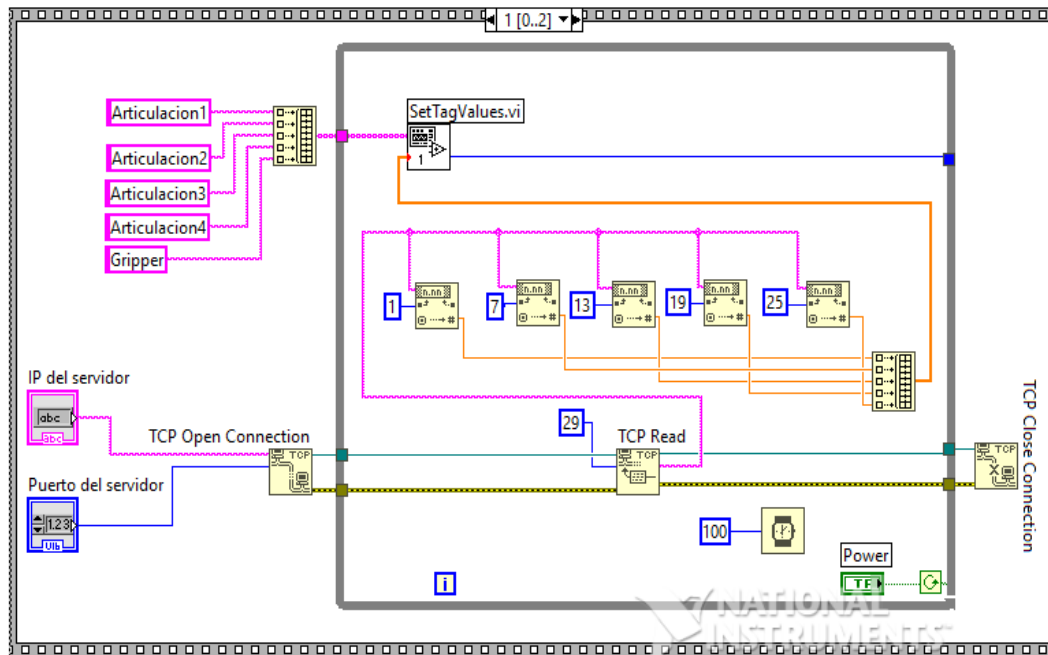
Arquitectura de comunicaciones

La comunicación entre el emulador *Teach Pendant* y el robot virtual SCARAUC constituye un elemento indispensable en el desarrollo del presente trabajo. Para cumplir este objetivo, se elaboró un programa en LabVIEW encargado de recibir la información procedente del emulador y transmitirla al robot virtual. Como se deduce, el sistema de comunicación cuenta con dos etapas para la transmisión de información. En ambos casos se empleó el conjunto de protocolos TCP/IP, ampliamente adoptado como base de comunicación en sistemas industriales modernos y ciber-físicos de acuerdo con Xu, He y Li (2014).

La primera conexión fue establecida entre el *Teach Pendant* y LabView de acuerdo con las consideraciones descritas en los requerimientos del emulador, cuya arquitectura establecida fue (Cliente-Servidor), donde LabView funciona como cliente mientras que el *Tech Pendant* funciona como servidor. Para la segunda etapa, se implementó las bibliotecas RoboTalk proporcionadas por RoboWorks, la cual emplea los protocolos TCP/IP de forma predeterminada. El programa en LabView siguió un esquema secuencial, iniciando con la biblioteca "Connect.vi", responsable de solicitar la conexión con el robot virtual (donde LabVIEW actúa como cliente y RoboWorks como servidor), siguiendo las prácticas fundamentales de programación en LabVIEW descritas por Bishop (2015). Por otra parte, para la recepción y transmisión de datos desde y hacia el emulador, se incorporaron bloques funcionales mostrados en la figura 6, destacando "TCP Open Connection", que recibe los parámetros de IP y puerto del servidor proporcionados por el usuario a través de la interfaz gráfica mostrada en la figura 7. El

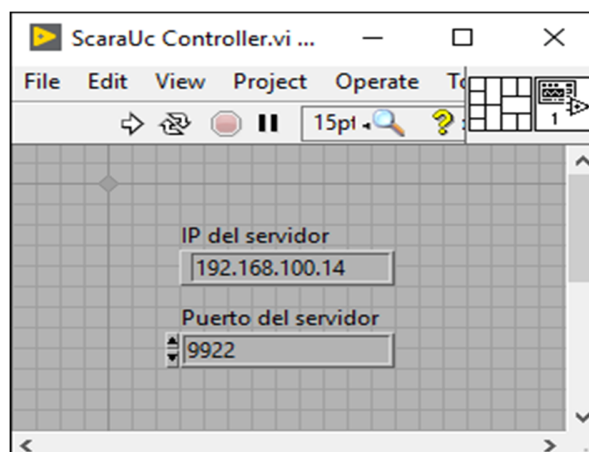
“TCP Read”, es el encargado de recibir los valores enviados desde el emulador, y “SetTagValues.vi”, perteneciente a RoboTalk, es la responsable de enviar finalmente la información al robot virtual. Para finalizar el proceso de comunicación, se empleó el bloque “TCP Close Connection”, destinado a cerrar la conexión.

Figura 6
Bloques funcionales en entorno LabView



Fuente: Aguiar y Torres (2026)

Figura 7
Interfaz del SCARAUC Controller



Fuente: Aguiar y Torres (2026)

Protocolo de validación

Para la validación del sistema, se llevó a cabo un procedimiento experimental utilizando la aplicación desarrollada en Android en conjunto con la interfaz de comunicación implementada en LabVIEW y el robot virtual SCARAUC. En una primera etapa, se estableció la conexión entre el emulador y el robot virtual mediante el protocolo TCP/IP, con el propósito de verificar la correcta transmisión de comandos entre ambos sistemas.

Posteriormente, se realizaron pruebas de movimiento asignando coordenadas cartesianas de forma aleatoria, con el objetivo de evaluar la respuesta del robot a partir de la cinemática inversa. De manera complementaria, se ejecutaron movimientos mediante el control independiente de cada articulación, con el fin de comprobar el comportamiento del sistema bajo condiciones de cinemática directa.

Finalmente, se almacenaron múltiples posiciones generadas durante las pruebas para la creación de una lista de ejecución. Esta lista fue utilizada para ejecutar secuencialmente las posiciones definidas, permitiendo validar el funcionamiento conjunto de las funcionalidades de almacenamiento, gestión y ejecución de trayectorias del sistema.

Resultados

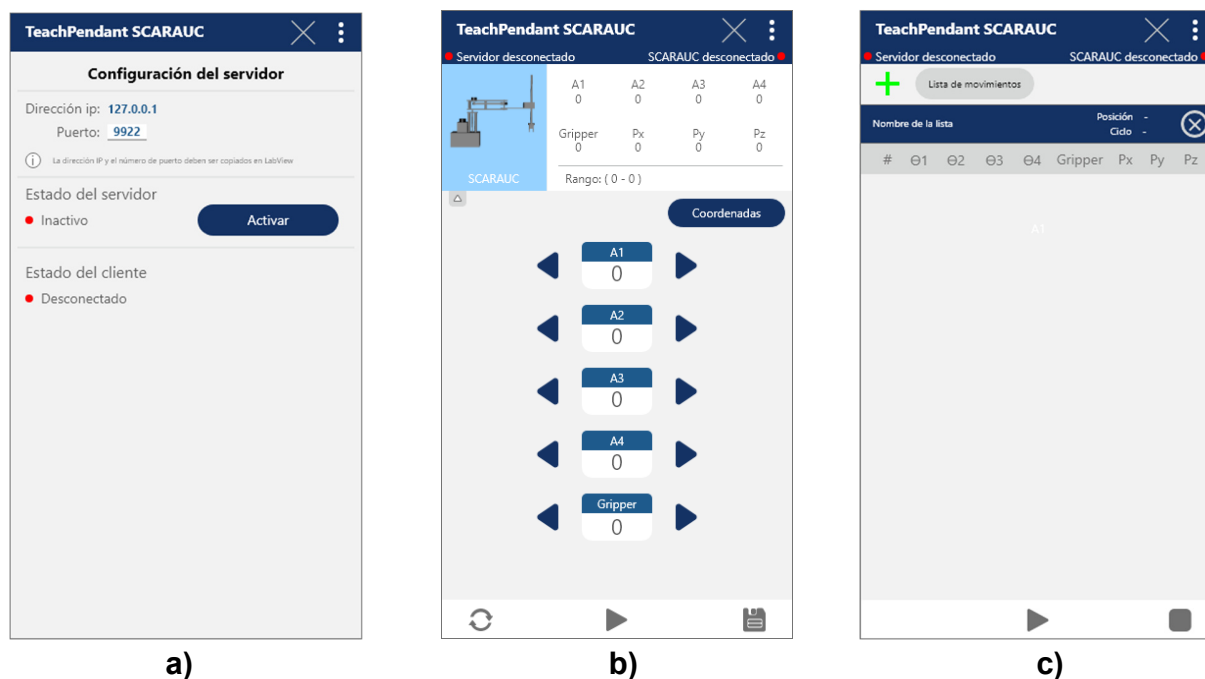
Emulador desarrollado

La aplicación desarrollada cuenta con tres pantallas principales, las cuales se presentan en la figura 8 y están orientadas a funciones específicas del sistema. En la figura 8a se muestra la pantalla “Configuración Servidor”, en la que el usuario puede establecer los parámetros necesarios para la conexión con LabVIEW, tales como la dirección IP y el puerto de comunicación. Esta interfaz permite iniciar y gestionar la comunicación con el robot virtual. En la figura 8b se presenta la pantalla “Interfaz de control”, la cual permite manipular el movimiento del robot.

En esta sección, el usuario puede ingresar coordenadas cartesianas o valores de posición en los ejes p_x , p_y y p_z para controlar la posición del efector final, así como realizar movimientos directos de las articulaciones. Finalmente, en la figura 8c se muestra la pantalla “Lista”, destinada a la gestión de posiciones almacenadas. En esta interfaz, el usuario puede visualizar, ejecutar y administrar secuencias de puntos previamente definidos, permitiendo la ejecución automática de trayectorias del robot.

Figura 8

Pantallas principales del Emulador Teach Pendant. a) Pantalla “Configuración Servidor”. b) Pantalla “Interfaz de control”. c) Pantalla “Lista”.



Fuente: Aguiar y Torres (2026)

Funcionalidades implementadas

En la Tabla 2 se presentan las principales funcionalidades implementadas en la aplicación SCARAUC, incluyendo su descripción, los pasos para su ejecución y las consideraciones relevantes durante su uso.

Tabla 2
Principales funciones de la aplicación SCARAUC

Funcionalidad	Descripción	Pasos principales	Particularidades / Observaciones
Conexión con el robot virtual	Permite que el usuario establezca conexión entre el dispositivo móvil y el robot virtual SCARAUC	1. El Teach Pendant muestra la dirección IP del dispositivo. 2. El usuario activa el servidor.	1. El emulador no tiene IP asignada. 2. El robot virtual no está conectado a la red. 3. Error en el intento de conexión.
Envío de información	Una vez establecida la conexión, el usuario puede enviar datos de control al robot.	1. El usuario elige entre manipular articulaciones individualmente o en conjunto. 2. Indica la acción de enviar la información.	1. Error en el envío de información.
Mostrar posición	Permite visualizar en tiempo real las coordenadas del robot virtual.	1. El emulador muestra por defecto las coordenadas en la interfaz.	
Lista de posiciones	Permite programar tareas mediante listas de posiciones almacenadas.	1. Agregar posiciones a la lista. 2. Guardar en la base de datos. 3. Ejecutar la lista. 4. Indicar número de ciclos. 5. Pausar, reanudar o detener la ejecución.	1. No es necesario guardar la lista para ejecutarla. 2. Error en la transmisión de datos.

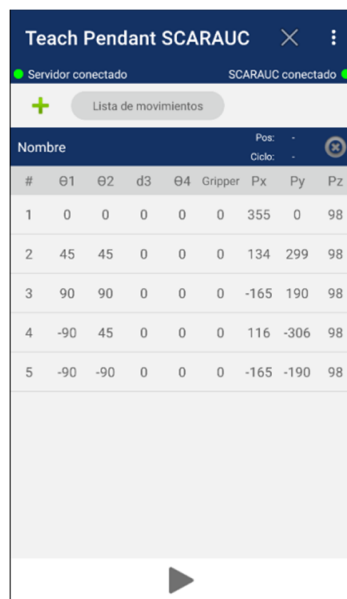
Fuente: Aguiar y Torres (2026)

Validación funcional

En la figura 9 se presenta la ejecución de una lista de posiciones previamente definida en la aplicación. En la figura 9a se muestra la interfaz con las posiciones almacenadas, mientras que de 9c a 9g se evidencian las distintas configuraciones alcanzadas por el robot virtual durante la ejecución de la secuencia. A partir de estas evidencias, se verifica que el sistema ejecuta de manera secuencial cada una de las posiciones definidas por el usuario, respetando el orden establecido en la lista. Asimismo, se observa que el robot alcanza correctamente cada una de las posiciones programadas, lo que demuestra el adecuado funcionamiento de la comunicación entre el emulador y el entorno de simulación, así como la correcta implementación de las funcionalidades de control y ejecución de trayectorias.

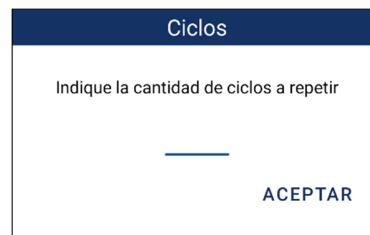
Figura 9

Ejecución del listado de posiciones. a) Pantalla “Lista” con varias posiciones almacenadas, b) Cantidad de ciclos a repetir, c) Posición 1, d) Posición 2, e) Posición 3, f) Posición 4, g) Posición 5.



#	Θ1	Θ2	d3	Θ4	Gripper	Px	Py	Pz
1	0	0	0	0	0	355	0	98
2	45	45	0	0	0	134	299	98
3	90	90	0	0	0	-165	190	98
4	-90	45	0	0	0	116	-306	98
5	-90	-90	0	0	0	-165	-190	98

a)

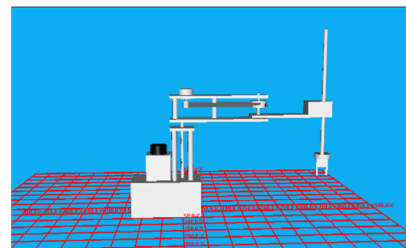


Ciclos

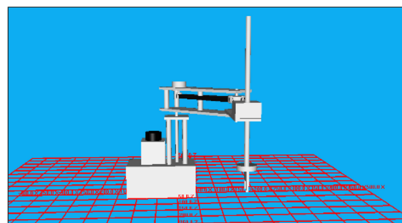
Indique la cantidad de ciclos a repetir

ACEPTAR

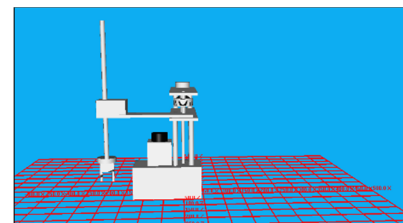
b)



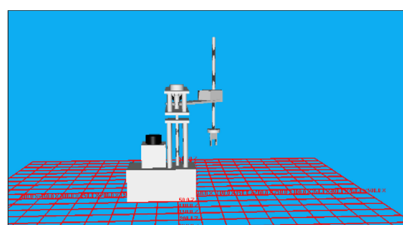
c)



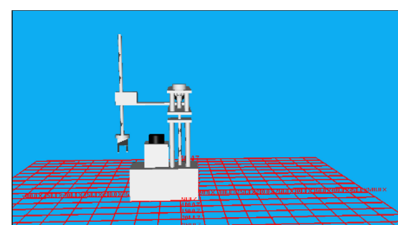
d)



e)



f)



g)

Fuente: Aguiar y Torres (2026)

Discusión

El desarrollo del emulador Teach Pendant permitió cumplir con el objetivo general de la investigación, el cual consistía en implementar una aplicación Android para el control del robot SCARAUC virtual.

Asimismo, los objetivos específicos fueron alcanzados de manera coherente, el diagnóstico inicial permitió identificar correctamente los parámetros de entrada del robot, el diseño de la interfaz gráfica respondió a los requerimientos del sistema, la implementación en Android se realizó de forma funcional, y la validación experimental confirmó el correcto desempeño del emulador en la ejecución de tareas programadas.

En comparación con trabajos previos como los de Lovon (2019) y Ramírez (2016), donde también se emplea Android para el control de sistemas robóticos, la solución propuesta mantiene la misma línea de versatilidad y accesibilidad, pero incorpora como aporte la integración con un entorno de simulación 3D específico (RoboWorks) mediante LabVIEW, lo que amplía las posibilidades de experimentación en entornos educativos. A diferencia de estos trabajos, el presente desarrollo enfatiza la emulación de un Teach Pendant industrial, lo que acerca al usuario a una experiencia más realista de interacción con sistemas robóticos.

Entre las principales ventajas del sistema desarrollado se encuentra su bajo costo de implementación, al no requerir hardware especializado, así como su utilidad en contextos educativos, donde permite a estudiantes interactuar con un robot virtual de manera intuitiva. La estructura de la aplicación, organizada en tres pantallas claramente definidas, facilita su uso y aprendizaje, mientras que el almacenamiento local de datos mediante SQLite aporta autonomía al sistema. Asimismo, el uso de tecnologías ampliamente disponibles como Android y TCP/IP contribuye a la portabilidad y escalabilidad de la solución.

No obstante, el sistema presenta ciertas limitaciones. En primer lugar, la comunicación depende del uso de LabVIEW como intermediario, lo que introduce un componente adicional en la arquitectura. Además, el emulador no incorpora control avanzado de trayectorias, limitándose a movimientos basados en posiciones discretas. Por otro lado, su funcionamiento está restringido al robot virtual, requiriéndose adaptaciones para su implementación en un robot físico real. Finalmente, la comunicación se basa en redes LAN o WLAN, lo que condiciona su uso a entornos donde se disponga de conectividad adecuada.

En conjunto, estos resultados evidencian que la solución desarrollada es viable y funcional para aplicaciones educativas y de simulación, aunque aún presenta oportunidades de mejora orientadas a su extensión hacia sistemas físicos y a la incorporación de capacidades de control más avanzadas.

Conclusiones

El desarrollo e implementación del emulador Teach Pendant permitió cumplir con los objetivos planteados, logrando una aplicación funcional basada en Android para el control del robot SCARAUC virtual. La solución desarrollada demostró ser capaz de ejecutar movimientos mediante coordenadas cartesianas y control individual de articulaciones, así como gestionar listas de posiciones para la automatización de tareas, validando su utilidad en entornos de simulación. Asimismo, la integración mediante comunicación TCP/IP entre el emulador, LabVIEW y RoboWorks evidenció un desempeño estable y coherente, confirmando la viabilidad del enfoque propuesto para aplicaciones educativas en robótica. En este sentido, el sistema desarrollado representa una herramienta accesible y de bajo costo que facilita la interacción con robots industriales en entornos de formación. Finalmente, aunque el emulador cumple con los requerimientos establecidos, se identifican oportunidades de mejora orientadas a la ampliación de sus funcionalidades y a su posible adaptación para el control de robots físicos reales, lo que permitiría extender su alcance hacia aplicaciones más avanzadas.

Referencias

- Adrián, E., & Castañeda, A. (2011). *Implementación de un sistema de manipulación y control mediante visión artificial para un autómatas industrial SCARA* [Trabajo de Grado, Universidad de Carabobo]. Repositorio Institucional UC. <http://www.riuc.bc.uc.edu.ve/handle/123456789/7093>
- Android Developers. (2024). Android documentation. <https://developer.android.com/>
- Bishop, R. H. (2015). *Learning with LabVIEW* (8th ed.). Pearson.
- Brown, S. (2015). *The C4 model for visualising software architecture*. <https://c4model.com>
- Caparrosa, O., I. (1999). Análisis cinemático. *Revista Universidad del Norte*, 7(1), 17–26. <https://dialnet.unirioja.es/servlet/articulo?codigo=5313949>
- Hipp, D. R. (2023). *SQLite documentation*. SQLite Consortium. <https://sqlite.org/docs.html>
- Lovon, P. W. (2019). *Desarrollo de una plataforma en Android para teleoperación de un sistema robótico en seguridad de empresas y almacenes* [Tesis de grado, Universidad Nacional de San Agustín]. Repositorio UNSA. <http://repositorio.unsa.edu.pe/handle/UNSA/8352>
- Meier, R. (2015). *Professional Android™ 5 Application Development*. Wrox Press.
- Ramírez, J. (2016). *Teleoperación de un robot móvil mediante dispositivos Android* (Trabajo de fin de grado, Universidad de Málaga). Repositorio UMA. <http://hdl.handle.net/10630/11435>
- Sanz, W. S. (2013). *Cinemática de robots industriales* (2.ª ed.). Dirección de Medios y Publicaciones de la Universidad de Carabobo.
- Xu, L. D., He, W., & Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243.