

Sistema de supervisión aplicado al robot de agricultura de precisión CERES para el cuidado de los cultivos agrícolas

Supervision System Applied to the Precision Agriculture Robot CERES for Agricultural Crops Care

David S. Moreno C.¹
Alexandra Velasco V.²
Leonardo E. Solaque G.³

¹ Universidad Militar Nueva Granada, Colombia.
Email: davids.moreno@unimilitar.edu.co

² Universidad Militar Nueva Granada, Colombia.
Email: alexandra.velasco@unimilitar.edu.co

³ Universidad Militar Nueva Granada, Colombia.
Email: leonardo.solaque@unimilitar.edu.co

 OPEN ACCESS



Copyright:

©2024. La revista *Ingenierías USBmed* proporciona acceso abierto a todos sus contenidos bajo los términos de la licencia [creative commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) Atribución no comercial SinDerivar 4.0 Internacional (CC BY-NC-ND 4.0)

Tipo de artículo: Investigación.

Recibido: 23-01-2023.

Revisado: 31-03-2023.

Aprobado: 05-04-2024.

Doi: 10.21500/20275846.6216

Referenciar así:

D. S. Moreno C., A. Velasco V. y L. E. Solaque G., “Sistema de supervisión aplicado al robot de agricultura de precisión CERES para el cuidado de los cultivos agrícolas,” *Ingenierías USBMed*, vol. 15, n.º 2, pp. 26–35, 2024.

Disponibilidad de datos:

todos los datos relevantes están dentro del artículo, así como los archivos de soporte de información.

Conflicto de intereses:

los autores han declarado que no hay conflicto de intereses.

Editor: Andrés Felipe Hernández.
Universidad de San Buenaventura,
Medellín, Colombia.

Resumen. La reforma significativa en los procesos de producción que actualmente emplean los agricultores para garantizar la cadena alimenticia en Colombia requiere desarrollos que aporten a la optimización de recursos, utilización adecuada de insumos y un menor uso de mano de obra. Para esto la robótica juega un papel fundamental que puede ayudar a afrontar estos retos por medio de sistemas supervisores capaces de gestionar tareas coordinadas en beneficio del agro. Este artículo presenta el desarrollo de un sistema de supervisión para un robot que brinda cuidado y protección a la salud de los cultivos de papa, orientado a prevenir problemas fitosanitarios —como la gota (tizón temprano y tardío) y *epitrix*— mediante herramientas que se ajustan a tareas como deshierbe, aplicación de nutrientes, insecticidas o hidratación, y garantizando la salud de la planta con la nutrición necesaria. La estructura del sistema supervisor implementada en el robot agrícola CERES se basa en la teoría de toma de decisiones adaptativa que funciona en tiempo real. Por medio del lenguaje Python se codificó el algoritmo de supervisión y se integró a la arquitectura de programación general con la que cuenta el robot agrícola basada en ROS. Como resultado de la integración del supervisor se obtuvo una jerarquización de las labores para el cuidado de los cultivos y su correcta ejecución según la priorización realizada para las diferentes tareas en beneficio del cuidado fitosanitario del cultivo de papa.

Palabras Clave. Robótica móvil, inteligencia artificial, ROS, toma de decisiones, programación, robótica para agricultura.

Abstract. The significant reform in the production processes that farmers currently use to guarantee the food chain in Colombia, requires developments that contribute to the optimization of resources, adequate use of supplies, and less use of labor. For this, robotics plays a fundamental role that can help meet these challenges through supervisory systems capable of managing coordinated tasks for the benefit of agriculture. This article presents the development of a supervision system for a robot that provides care and protection to the health of potato crops, aimed at preventing phytosanitary problems through tools that adjust to tasks such as weeding, application of nutrients, insecticides or hydration, thus preventing some phytosanitary problems such as gout (early blight and late blight) and *epitrix*, and can also guarantee the health of the plant through the necessary nutrition. The structure of the supervisory system implemented in the CERES robot was based on the adaptive decision-making theory that works in real-time. The supervision algorithm was integrated into the general programming architecture of the CERES agricultural robot using the Python language, which allowed the supervision algorithm to be codified and integrated into the general programming architecture of the agricultural robot, which in this case is based on ROS. Because of the integration of the supervisor, a hierarchy of the tasks for the care of the crops and their correct execution was obtained according to the prioritization made for the different tasks for the benefit of phytosanitary care of the potato crop.

Keywords. Mobile Robotics, Artificial Intelligence, ROS, Decision- Making, Programming, Agriculture Robotics.

I. Introducción

En Colombia la agricultura es uno de los principales sectores económicos del país al brindar fuentes para el abastecimiento alimentario en Colombia y exportación de algunos productos a otras naciones [1]. Tanto es así que, según la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura), a futuro Colombia puede convertirse en una de las principales fuentes de productos agrícolas del mundo, dado que tiene un gran potencial para el desarrollo de áreas cultivables. Actualmente, de los 22 millones de hectáreas disponibles para labores de cultivo solo se están empleando 4.8 millones [2].

Para implementar cultivos y obtener productos agrícolas en las áreas cultivables en el sector colombiano es necesario entender los problemas presentes en las técnicas de agricultura tradicional empleadas en Colombia. Según [3], la agricultura colombiana conserva algunas formas y técnicas de cultivo tradicionales con un grado de tecnificación muy bajo o nulo, es decir que existe una falta de integración efectiva entre la innovación tecnológica y el campo agrícola. Por otro lado, los cultivos pueden presentar problemas fitosanitarios como plagas, hongos, entre otros. En algunos casos, para su erradicación se utilizan plaguicidas químicos que pueden causar problemas de salud a los campesinos debido a la exposición constante con este tipo de productos. En [4] se presenta un estudio en el que se evaluaron los efectos de los plaguicidas en un grupo de 132 trabajadores del sector agrícola colombiano que habían tenido un tiempo de 9 años de exposición directa a dichos productos. Se evidencia que el 95.5% de los trabajadores presentaban problemas en el sistema nervioso central. Estos problemas se podrían mitigar si se implementan soluciones tecnológicas integradas a las labores tradicionales en el agro que eviten la manipulación directa y constante de este tipo de productos.

Para que Colombia se proyecte como país agricultor con terrenos cultivables y llegue a ser una de las principales despensas a nivel mundial, es necesario suplir la demanda de alimento en respuesta al aumento poblacional del mundo. Esto hace que a futuro la necesidad de producción crezca de manera progresiva, lo que conlleva a que se requieran implementar nuevas tecnologías que garanticen seguridad y aumento de producción, como lo evidencian los estudios en [5], donde se reporta aumento en la rentabilidad al emplear sistemas robóticos agrícolas en un porcentaje que puede oscilar entre el 20% y 72%, según los cultivos.

El uso de sistemas robóticos permite contribuir a la tecnificación del campo, mejorar la producción y reducir los impactos negativos del uso de productos como plaguicidas. Además, la realización de tareas relacionadas con el cuidado y manejo de cultivos agrícolas son complejas y variables, ya que se deben ejecutar

actividades como el riego y tratamiento de enfermedades. El desarrollo de estas requiere de un sistema de supervisión que ayude a dirigir cada una de forma ordenada y precisa, de acuerdo con las necesidades de cada planta.

CERES es un sistema robótico enfocado en soportar el cuidado de los cultivos de hortalizas, entre otros, por medio de sensores y actuadores. El robot realiza actividades agrícolas como el deshierbe, la aplicación de nutrientes, plaguicidas e hidratación, las cuales se determinan según el estado de salud de la planta, solucionando el problema fitosanitario presentado.

Para que dichas labores puedan ser realizadas de forma autónoma por el robot se crea un algoritmo de toma de decisiones por medio de la definición de las variables necesarias, y la organización y estructuración de las prioridades que se tienen para proveer una solución efectiva tal como se presenta en [6]. Dicho algoritmo funciona con un mapa que permite clasificar las diferentes acciones que debe realizar el robot para intervenir la planta de forma adecuada. Para esto se establece un protocolo automatizado que integra los diferentes subsistemas involucrados para el cuidado del cultivo. Este proceso y su validación se realiza sobre la estructura real mediante ROS, el meta sistema operativo del robot.



Figura 1. Robot eléctrico Ceres

De acuerdo con el contexto anterior, este artículo tiene como fin presentar un sistema supervisor integrado al robot agrícola CERES [7] (ver Figura 1), que permita la articulación de todas las tareas requeridas para su funcionamiento para que sea un sistema orientado a aportar en la industria agrícola colombiana. De esta manera, se busca obtener un impacto positivo en la producción de los cultivos para promover el uso de máquinas autónomas en Colombia y evidenciar la efectividad de tecnificar el agro en el país.

II. Fundamentación teórica

A. Estado del Arte

En el área de la industria agrícola la implementación de robots que ayuden a mejorar el desempeño de los cultivos es cada día más frecuente, debido a las dificultades que se presentan dadas las diferentes características

de los terrenos, el manejo constante de químicos, la manipulación de cargas y el aumento constante de la demanda de productos [8]. Por ello, diversas compañías buscan implementar robots que faciliten, agilicen y hagan más segura la producción, por lo que su uso en agricultura se ha incrementado en la última década.

En un contexto global se puede encontrar diversidad de prototipos de vehículos autónomos ejecutando diferentes técnicas y herramientas destinadas a la agricultura. Uno de estos prototipos es el desarrollado en la India por Sharma y Borse [9], quienes presentan un prototipo de robot de supervisión agrícola que emplea un algoritmo de toma de decisiones basado en la estructura de árbol de decisiones para la detección de enfermedades en plantas, control de crecimiento, rociado de pesticida, fertilizante y agua por medio de sensores inalámbricos, y análisis de imágenes que permitan el seguimiento de la planta controlando su salud y necesidades. De igual manera, Beloev [10] describe el diseño y creación de un algoritmo de toma de decisiones para un prototipo de robot agrícola enfocado a labores simples de cultivos en terrenos de Bulgaria. Dicho prototipo cuenta con un sistema móvil autónomo, con autolocalización y la posibilidad de realizar un mapeo o inspección en los terrenos, proporcionando la capacidad de tomar decisiones dependiendo de la situación y el entorno en el que se encuentre. Igualmente, Rubio y Rovira [11] presentan la creación de un robot autónomo agrícola llamado Vineyard, enfocado en cultivos de viñedos de New Orleans, Louisiana. Este realiza un mapeo en tiempo real por medio de GPS para obtener la posición y sensores no invasivos con el fin de registrar las características de los frutos, el contenido de nitrógeno en las hojas y el nivel de antocianinas de las uvas, para así generar un recorrido a partir de la toma de decisiones según el estado de maduración de la fruta y la preparación para su cosecha.

Los prototipos de vehículos autónomos de supervisión enfocados en la agricultura cuentan, por lo general, con diversas herramientas y técnicas para realizar rutinas de desplazamiento por medio de sistemas de locomoción. Así mismo, se implementan rutinas de detección de obstáculos por medio de sensores, cámaras, centrales inerciales o rutinas para generar mapeo del terreno por medio de GPS. Adicionalmente pueden incluir sistemas para intervenir en el campo agrícola, tales como un manipulador, una bomba de rocío o un taladro para el deshierbe. Por ejemplo, Barbosa [12] presenta la navegación autónoma de un robot en los campos agrícolas sujeta a una visión con cámara monocular, con lo cual condiciona la velocidad de conducción y los desniveles del terreno. De igual manera, Durand [13] expone un protocolo de evasión de obstáculos que emplea la captura de objetos por medio de cámaras o sensores dedicados a identificar elementos de interés y sobre esa base determinar la mejor ruta segura con evasión, siendo estas dos opciones referentes para la jerarquización

de las tareas de una correcta ejecución de los procesos y lograr la navegación autónoma supervisada.

En cuanto a los manipuladores integrados a los robots agrícolas, se pueden observar técnicas empleadas en

los sistemas de supervisión como el empleado por Liu y Chen [14] con el desarrollo de un sistema *offline* de planificación de trayectorias para el efector final del manipulador, generando recorridos de movimiento continuo y sin colisión para la eliminación de malezas, técnica que complementa el desarrollo de la precisión del *gripper* y brinda al robot una capacidad de respuesta superior al momento de realizar una vigilancia de los estados de salud de los campos agrícolas. Desde una perspectiva para mejorar los tiempos de respuesta de los sistemas para los procesos de supervisión agrícola, Kapić [15] plantea una rutina de conexión a internet para cargar los datos a la nube, enviar al usuario alarmas o datos importantes por medio de una aplicación en el celular y realizar los cálculos de los algoritmos de toma de decisiones, liberando el procesamiento del computador principal con el fin de disminuir la carga del sistema para enfocarse en sus protocolos de supervisión y de toma de decisiones como lo son el análisis de las plantas y el sistema de evasión de obstáculos al momento de realizar su operación. Asimismo, Smirnov y Teslya [16] presentan una técnica de supervisión por medio de un mapa de decisiones con varios sistemas robóticos supervisores agrícolas, los cuales permiten realizar una coalición por medio del funcionamiento unificado de labores con el fin de realizar múltiples tareas con mejor precisión, reduciendo los tiempos de respuesta y abarcando una mayor área de trabajo.

El robot de supervisión agrícola CERES [7] cuenta con un sistema de locomoción en configuración diferencial, empleando dos motores eléctricos de 5 kw con un control de bajo nivel que asegura velocidad de traslación y rotación mediante controladores PI. Tiene, a su vez, un control de alto basado en técnicas de Lyapunov, que asegura el seguimiento de trayectorias. Además, implementa un manipulador cartesiano controlado por un sistema de servo-visión que evita colisiones con las plantas, lo que permite que el efector final logre su objetivo. Este último posee un sistema de remoción de maleza y un rociador para emplearlo en las labores de nutrición y fumigación focalizado que garantiza la salud de los cultivos [17].

Para que el robot sea autónomo en la actividad agrícola es necesario implementar un sistema de supervisión, con este fin se desarrolló la estructura de un sistema supervisor basado en la teoría de toma de decisiones adaptativa [18], este se puede adecuar a cualquier cambio interno en tiempo real. Por ejemplo, de acuerdo con la limitación de la carga de la batería del robot o los niveles de los tanques de nutrición o fumigación, el

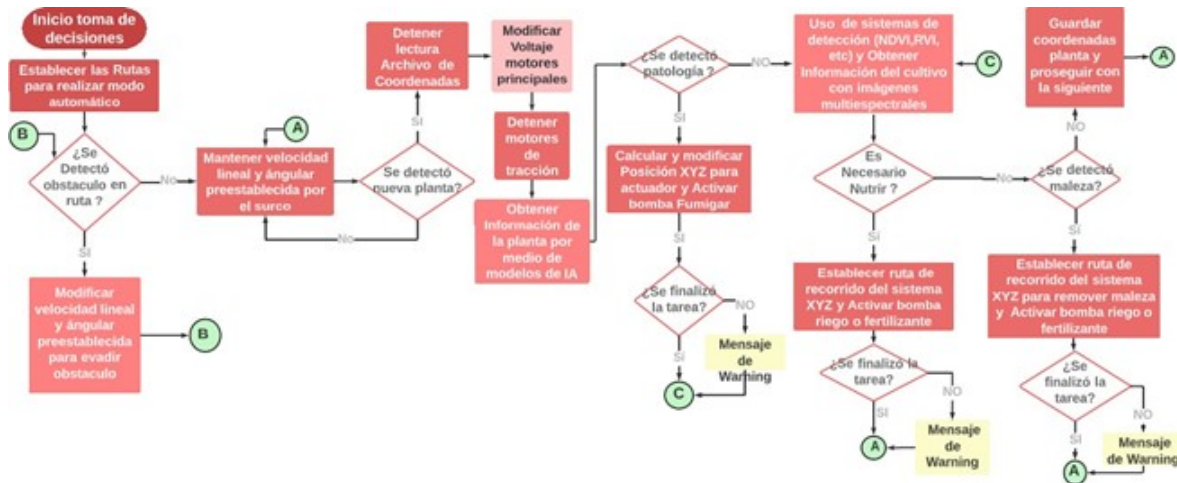


Figura 2. Proceso de soporte técnico

sistema de toma de decisiones se adapta estableciendo un cambio en el sistema y convirtiendo en prioridad la recarga de productos o batería del robot, o cualquier cambio externo como las condiciones climáticas, la evasión de obstáculos o los cambios en el terreno que requieren la toma de decisiones para el funcionamiento adecuado del robot. La teoría de toma de decisiones adaptativa proporciona una solución acorde a las necesidades que se presentan durante los cuidados de la salud en los cultivos [18].

B. Sistema de toma de decisiones

Un sistema de toma de decisiones se refiere a un conjunto de técnicas y herramientas generadas por medio de un programa que planifica y permite al robot realizar los movimientos necesarios para cumplir una tarea, ello para funcionar de forma autónoma sin necesidad de tener un supervisor permanente [19]. Para crear un sistema de toma de decisiones de un robot agrícola es necesario establecer las técnicas que abordarán los problemas de salud de los cultivos y las herramientas con las que debe contar o cuenta el robot. Sobre esta base se realiza la planificación de acciones que permitan el cumplimiento de las tareas u objetivos propuestos. Con base en lo anterior, en esta sección se establece un diagrama de toma de decisiones según la estrategia que se identificó para la plataforma robótica donde se consideran los elementos de percepción, el sistema de locomoción, el sistema del manipulador cartesiano con su efector final, que puede hacer riego o aplicación de plaguicidas, y el sistema de navegación autónoma.

Para la realización del mapa de decisiones se definieron las prioridades y los tratamientos necesarios para intervenir en los cultivos de papa. Las tareas definidas son: el deshierbe, la nutrición y la fumigación. Posteriormente se establecieron las etapas y los recorridos necesarios para realizar una ruta de supervisión con sus respectivas banderas de activación con base en los sistemas con los que cuenta el robot.

En el mapa de decisiones de la Figura 2 se evidencian las acciones necesarias para el cumplimiento de las tareas de supervisión y las acciones a implementar en un recorrido automático, donde por medio de servo-visión se pueden detectar los obstáculos presentes en la ruta establecida y las plantas para verificar su estado de salud. Dicho mapa cuenta con acciones como la activación de motores de riego de nutrición o fumigación, activación de *gripper* para el deshierbe y de alarmas para informar acerca de las acciones que se están ejecutando.

C. Solución planteada

El diagrama de estados de la Figura 3 define el funcionamiento del sistema de toma de decisiones planteado. En total se presentan 19 estados para realizar un funcionamiento de supervisión autónomo para el robot CERES.

Para la activación de cada estado se utilizaron 23 señales, como se muestran en la Tabla 1. Las señales implementadas fueron: “be”, que controla el encendido y apagado de todo el robot; “pe” para el apagado de emergencia; “gp” para la ejecución de los programas que activan y corren la GUI principal (interfaz gráfica); “sP” es la bandera de funcionamiento de los periféricos (motores, actuadores, cámaras, etc.); “mov” es la bandera de movimiento y traslación; “jb” es la señal de inicio de rutinas de trabajo; “sF” activa las tareas de fumigación; “sN” es la bandera de activación de tareas de nutrición; “sD” activa las tareas del deshierbe; “obp” es la señal captada por cámara de detección de planta; “obs” es la señal captada por cámara de detección de obstáculos y gripper; “prp” finaliza el recorrido programado; “ie” es la bandera de finalización de la tarea ejecutada con éxito; “Sv” almacena los datos de estado de la planta y su posicionamiento; “pV” es la bandera de provisiones de los tanques de trabajo; “cH” es la bandera de la ruta de posición HOME; “fR” es la señal de la ruta de recarga; “fM” es la señal de la ruta de desplazamiento del recorrido de trabajo; “fS” permite

retomar el trabajo en la posición guardada; “fSM” detiene el desplazamiento; “fSP” detiene el recorrido de trabajo; “fSB” es el paro de emergencia; y “cT” es la señal de la continuación del trabajo.

Tabla 1. 23 señales empleadas para el diagrama de estados

Señales	Acciones
be	Encendido
pe	Paro de emergencia
gp	Run GUI python
sP	Switch Periféricos
mov	Movimiento
jb	Job
sN	Nutrición
sD	Deshierbe
obs	Detección de obstáculos
obp	Detección de plantas
prp	Parada de recorrido programado
ie	Intervención ejecutada
Sv	Guardar dato
pV	Chequeo de provisiones
cH	Coordenadas HOME
fR	Bandera Recarga
fM	Bandera Movimiento
fS	Bandera Estado Guardado
fSM	Bandera Stop Motion
fSP	Bandera Stop Job
fSB	Bandera Stand By
cT	Continuar trabajo

La Figura 3 presenta de forma compacta la estructura del supervisor. Se tiene un total de 34 posibles combinaciones de señales debido a que hay estados, por ejemplo el “Stop Motion”, que se activan por medio de diferentes combinaciones de señales. Cabe mencionar que de ninguna forma es posible activar dos o más estados al mismo tiempo, con el fin de garantizar un funcionamiento cíclico del algoritmo de toma de decisiones impidiendo que un estado entre en conflicto con otro. Todas las señales empiezan desde 0 como se observa en la Figura 4.

Cada estado del diagrama de toma de decisiones se diseña por separado con su respectiva subcapa de decisiones donde se evidencia la tarea que se realizará posteriormente. Por ejemplo, el estado “Start motion” de la Figura 5 permite la ruta de desplazamiento del CERES en su recorrido de trabajo. Para que este estado inicie su funcionamiento debió haber sido activado anteriormente por medio de la señal “fM” como se observa en el diagrama de la Figura 3.

Cada uno de los estados presentados en la Tabla 2 cumple una función en especial para poder generar el algoritmo de toma de decisiones.

Posterior a su activación se observa que el estado tiene como primer paso detectar si existe o no algún obstáculo, en caso de detectarlo se activarán las señales “obs” y

Tabla 2. 19 estados definidos para el sistema de supervisión

No. estado	Estados
1	El estado de reposo
2	GUI ROS para dar inicio a los códigos de programación
3	Activación de periféricos
4	Robot Stand By para iniciar estado de alerta por paro de emergencia desactivando el funcionamiento
5	Robot On para establecer coordenadas de movimiento
6	Recarga CERES para dirigir al punto de recarga HOME
7	Redirección Estado Guardado para retornar a las coordenadas guardadas para continuar el trabajo
8	Start Job para dar inicio al trabajo del manipulador cartesiano
9	Información Planta para detectar especificaciones de la planta a tratar
10	Fumigación
11	Nutrición
12	Deshierbe
13	Verificación Provisión para comprobar niveles de tanques o batería
14	Start Motion para dar inicio al desplazamiento del robot
15	Stop Motion para detener el desplazamiento del robot
16	Stop Job el cual detiene el manipulador cartesiano
17	Save posición y coordenadas
18	Coordenadas HOME redirige al robot a las coordenadas para abastecimiento
19	Reemplazar variables originales

“fSM”, que activan los estados necesarios para empezar a realizar su evasión. En el caso de no detectar obstáculos, se procederá a identificar la presencia de alguna planta en el recorrido, y en el caso contrario de detectar alguna se activará la señal “jb”, la cual empieza el estado de trabajo sobre la planta y se desactiva la señal “fM” responsable de mantenerse en el estado “Start motion” para que funcione de forma cíclica. Finalmente, en el caso de no detectar ninguna planta, se verifica si la bandera de finalización de la tarea ejecutada con éxito “ie” está activada. Si la tarea fue ejecutada con éxito se procede a seguir con la trayectoria programada y continuar con la ruta de supervisión, en el caso de no haber sido realizada con éxito se identificará que anteriormente se presentó un problema que activó el paro de emergencia, lo que significa que el sistema va a reanudar el trabajo desde el último punto registrado por el sistema.

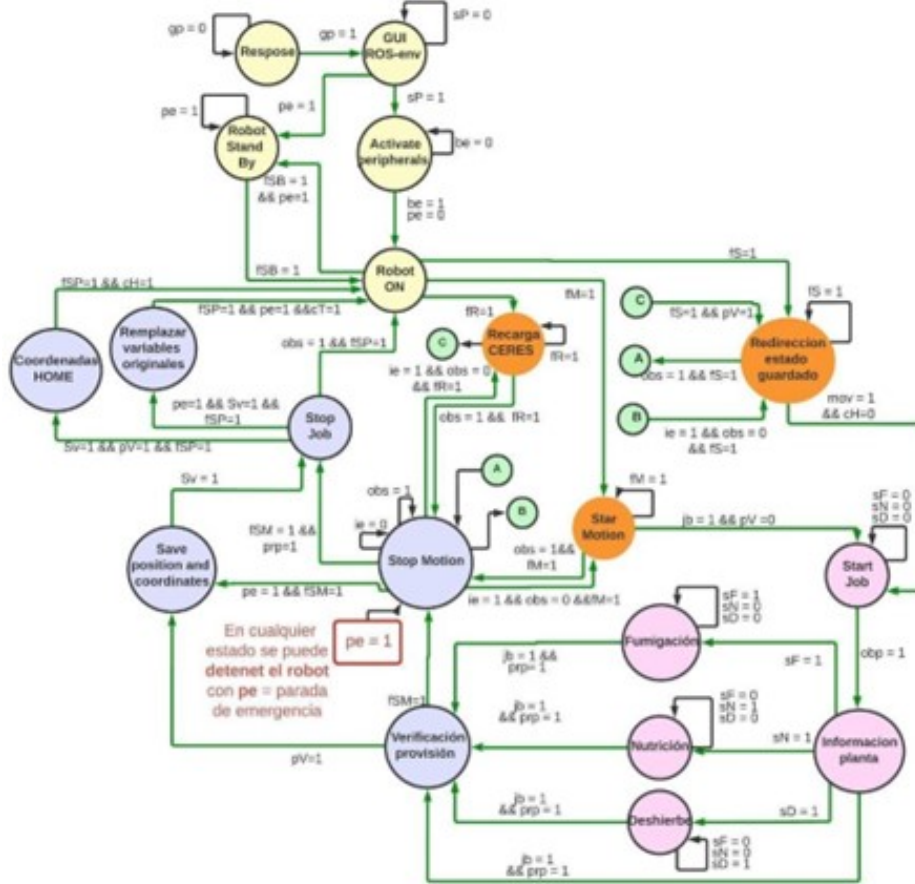


Figura 3. Diagrama de estados final

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
	ts	fm	fr	ct	ch	pv	sv	ie	prp	obp	obs	sd	sn	sf	jb	mov	sp	gp	pe	be	rsm	rsp	rsh	Estado Activado
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Response
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	GUI ROS-ENV
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ACTIVATE PERIPHERALS
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	ROBOT STAND BY
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	ROBOT ON
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Start Motion
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Stop Motion
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Start Motion
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Start Job
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Información planta
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Fumigación
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nutrición
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Deshierbe
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Verificación provisión
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Save position and coordinates
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Stop Motion
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Stop Job
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Stop Job
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Save position and coordinates
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROBOT ON
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Remplazar variables originales
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Coordenadas HOME
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROBOT ON
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROBOT ON
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROBOT ON
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROBOT STAND BY
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROBOT ON
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Redireccion estado guardado
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Stop Motion
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Redireccion estado guardado
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Start Job
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Recarga CERES
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Stop Motion
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Recarga CERES
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Redireccion estado guardado

Figura 4. Combinación de las 23 señales con su respectivo estado activo

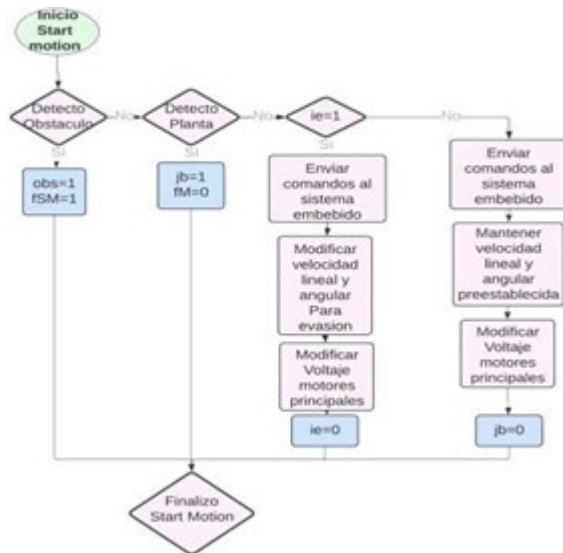


Figura 5. Ejemplos estados “Start motion”

III. Integración

En esta sección se desarrolla el algoritmo de toma de decisiones y se prueba en la estructura del robot con base en los sistemas y técnicas que actualmente presenta. Es importante aclarar que la detección, identificación y clasificación de problemas fitosanitarios están fuera del alcance de este artículo. Este se enfoca exclusivamente en la integración de todos los programas que presenta el CERES para realizar un algoritmo únicamente para el desarrollo del sistema de toma de decisiones. Sin embargo, fue necesario entender y priorizar las acciones que se deben tomar sobre los problemas fitosanitarios presentes en los cultivos de papa por medio de un proceso de investigación que tuvo como fin identificar e implementar las actividades necesarias y sus especificaciones en la toma de decisiones del robot de manera efectiva [7].

La integración del sistema de toma de decisiones sobre la plataforma CERES se realizó por medio del sistema operativo Ubuntu, base de los computadores embarcados en la plataforma, el cual aloja ROS encargado de gestionar y permitir la conexión entre todos los módulos que controlan las herramientas. Así se generó la interacción mediante publicaciones o suscripciones que realizan los diferentes nodos asociados a las tareas que la plataforma puede realizar. A su vez Python fue usado para codificar diferentes aplicaciones de los nodos con el fin de realizar las habilitaciones necesarias de las diferentes tareas.

Para integrar el supervisor se recopilaron los códigos con los diferentes procesos que tiene el robot. Se incluyen los elementos de percepción, el sistema de locomoción, el sistema del manipulador cartesiano y el sistema de navegación. Luego se realizó un diagrama de toma de decisiones con base en los procesos con los que

cuenta el robot. Por último se unificaron y redujeron los números de nodos y tópicos (forma de comunicación por medio de la exportación de información para realizar la interacción entre los nodos) empleados por los diferentes módulos asociados a los procesos.

El robot CERES tiene implementados un total de 3 computadores encargados de gestionar los procesos y la información. Se encuentran organizados categóricamente: un computador principal (*master*) y dos esclavos (*slave*). El computador *master* implementa la GUI principal (interfaz gráfica de usuario para la visualización de un entorno mediante el que un usuario puede interactuar con una máquina), la ejecución del sistema ROSCORE para la recepción de las comunicaciones de ROS, los códigos que activan los sensores (GPS, IMU) y actuadores (motores de locomoción y actuadores del sistema cartesiano), y el sistema de navegación autónoma. Uno de los computadores *slave* es la tarjeta JETSON TX1, la cual es un sistema embebido de bajo consumo y enfocado, en este caso, a tareas de inteligencia artificial [20], empleada en los procesos de detección de enfermedades mediante visión artificial. Por último, el otro segundo *slave* es dedicado al proceso del sistema cartesiano y cuidado fitosanitario mediante el efector final que aplica lo requerido por la planta. La comunicación de estos elementos es asegurada mediante un HUB-Ethernet. Un diagrama de estas conexiones se presenta en la Figura 6.

Para realizar el sistema de supervisión del robot, inicialmente se establece una estrategia de navegación del supervisor mediante su desplazamiento a una distancia de 1 metro (área de trabajo del sistema cartesiano). Posteriormente, a través de la cámara *primesense* se capturaron 4 coordenadas localizadas en los puntos de intersección de los lados de un cuadrado, en los cuales se realiza un recorrido consecutivo por medio del sistema cartesiano, coordenadas que son necesarias para poder abarcar el área de trabajo del *gripper*.

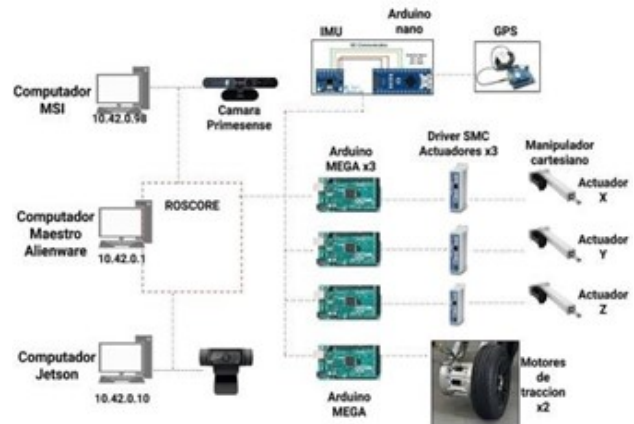


Figura 6. Mapa de conexiones de periféricos con sus respectivos computadores

Al encontrar una planta se activa el sistema de revisión de estado fitosanitario que por medio de inteligencia artificial detecta y activa el tratamiento a emplear para resolver el problema que se presente en la planta. Posterior a la identificación del problema fitosanitario se detiene el *gripper* encima de la planta a tratar y se ejecuta la solución al problema detectado, ya sea nutrición, fumigación o deshierbe. Si no se detecta ninguna otra planta, el robot CERES se desplazará un metro más y reiniciará la ejecución de las tareas como se puede observar en la Figura 7.

IV. Discusión y conclusiones

Se evidencia que la metodología seguida para el planteamiento del supervisor, donde se integró previamente el conocimiento de los subprocesos y su código, permitió un apropiado dimensionamiento de las tareas que el supervisor debe ir activando en pro de lograr la misión del robot como lo es el cuidado fitosanitario.

De acuerdo con esto, el diagrama de estados realizado para el CERES abordó todas las necesidades del robot para cumplir los objetivos planteados para su funcionamiento como lo son el desplazamiento, la detección de plantas y la determinación de su estado, y la solución de los problemas fitosanitarios que se pueden presentar, garantizando la salud de los cultivos.

Aunque los componentes tecnológicos que se integran a cualquier robot varíen tanto física como funcionalmente, el mapa de toma de decisiones podría ser integrado a robots diseñados para la supervisión agrícola, siempre y cuando el dimensionamiento del supervisor enmarque potencialidades, limitaciones de cada robot en particular y sean asociados al diagrama de estados final.

Se corroboró por medio de la práctica que el supervisor codificado e integrado al robot, cumplió con los objetivos de cuidado fitosanitario de los cultivos. De acuerdo con lo anterior, la estructura del supervisor y su integración a la plataforma CERES, fue probada mediante la realización de un recorrido en un surco de plantación de cultivos de papa, logrando detectar las plantas que pudieran tener alguna afección en sus hojas mediante la ejecución de tareas programadas por el supervisor como lo son la activación de desplazamiento, la detección de plantas en su recorrido, la detención del robot, la determinación de la afectación, el tratamiento por aplicación de fungicida y, posteriormente, la reactivación de la marcha del robot para realizar el ciclo nuevamente, todo esto por medio de las secuencias de activación presentadas en los diagramas de estados con las señales transmitidas por los tópicos creados en ROS e interpretadas por Python, lo anterior para lograr realizar una toma de decisiones adaptada a los cambios e imprevistos que se pueden presentar al momento de realizar una supervisión en un cultivo.

Se comprobó la efectividad del sistema de supervisión del robot CERES a partir de una comparación de tiempos de ejecución de una secuencia de trabajo preestablecida por medio del sistema de toma de decisiones y de la ejecución manual de cada labor, donde se realizó una rutina que consiste en un desplazamiento del CERES a una distancia de 1 metro, movimiento del sistema cartesiano, detección de planta y tratamiento de la misma, esto realizado en un laboratorio con condiciones controladas. Se obtuvieron tiempos de 250 segundos (ejecución manual) y de 140 segundos (ejecución autónoma), obteniendo así una reducción en los tiempos de 50% con la implementación de un sistema de toma de decisiones.

V. Trabajos futuros

Este proyecto puede servir como base para aumentar el desarrollo e implementación de tecnologías en el sector agrícola al promover la interacción de los avances tecnológicos, mecanismos y sistemas que se puedan emplear de manera efectiva, ya que se evidenció que la implementación de un código de toma de decisiones en las plataformas de supervisión agrícola permite organizar la disposición de herramientas y procesos aplicados a las tareas en este ámbito, pues al emplear dicho sistema se pueden obtener resultados seguros por medio de la aplicación de las especificaciones necesarias para preservar la salud de los cultivos y potencializar su producción.

Los resultados de este artículo abren la posibilidad de desarrollar e implementar supervisores que permitan complementar la autonomía de robots agrícolas como el CERES, sujetos a las necesidades y restricciones propias que de cada sistema puedan surgir, teniendo la posibilidad de disminuir y mejorar los tiempos de ejecución de cada tarea que puede realizar el robot en función de cumplir los objetivos sobre la base de un sistema robótico con restricciones o recursos limitados.

VI. Agradecimientos

Este trabajo es soportado por el proyecto INV-ING-3482, titulado “Identificación y manejo de problemas fitosanitarios en un cultivo de papa mediante el uso de un algoritmo integrado a la plataforma robótica CERES”, financiado por la Vicerrectoría de Investigaciones de la Universidad Militar Nueva Granada en Bogotá, Colombia.

Referencias

- [1] DVA, “Importancia de la agricultura colombiana,” 16 de marzo de 2021 [Online]. Available: <https://dva.com.co/importancia-de-la-agricultura-colombiana/>. [Last access: 28-09-2022].

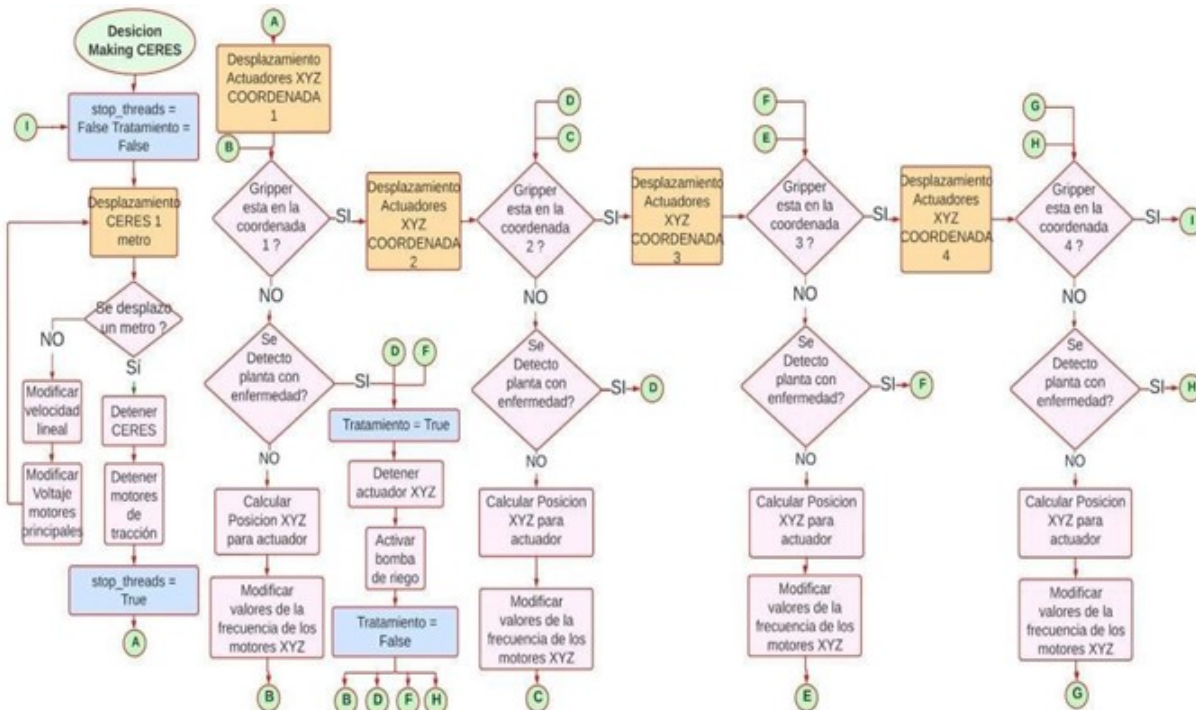


Figura 7. Prueba de algoritmo para toma de decisiones para recorrido programado por medio de coordenadas

- [2] E. Press, “Colombia, la despensa de alimentos del mundo para 2030,” 19 de octubre de 2018 [Online]. Available: <https://www.europapress.es/comunicados/internacional-00907/noticia-comunicado-colombia-despensa-alimentos-mundo-2030-20181019080139.html>. [Last access: 28-09-2022].
- [3] D. J. Pérez-Ortega, F. A. Bolaños-Alomia y A. M. Silva, “Variables que influyen en la aplicación de la agricultura de precisión en Colombia: revisión de estudios,” *Ciencia y Tecnología Agropecuaria*, vol. 23, n.º 1, 2022. DOI: 10.21930/rcta.vol23_num1_art:2298..
- [4] M. V. Uribe et al., “Impacto en la salud y el medio ambiente por exposición a plaguicidas e implementación de buenas prácticas agrícolas en el cultivo de tomate,” *Revista Chilena de Salud Pública*, vol. 16, n.º 2, págs. 96-106, jun. de 2011.
- [5] T. W. Griffin y J. Lowenberg-DeBoer, “Worldwide adoption and profitability of precision agriculture Implications for Brazil,” *Revista de Política Agrícola*, vol. 14, n.º 4, págs. 20-37, 2005.
- [6] B. (Colombia) Instituto Colombiano Agropecuario, *Instituto Colombiano Agropecuario, Manejo fitosanitario del cultivo de la papa (Solanum tuberosum subsp. andigena y S. phureja): medidas para la temporada invernal*. ICA, 2011 [Online]. Available: <https://repository.agrosavia.co/handle/20500.12324/2281>. Last access: 07-10-2022.
- [7] L. Solaque, G. Sánchez y A. Riveros, “Controlador PI2 para las velocidades de un robot agrícola evaluado usando Hardware en el lazo (HIL),” *Ingenierías USBMed*, 2022 [Online]. Available: <https://revistas.usb.edu.co/index.php/IngUSBMed/article/view/4662>. Last access: 1-10-2022.
- [8] M. Hussain, S. H. A. Naqvi, S. H. Khan y M. Farhan, “An Intelligent Autonomous Robotic System for Precision Farming,” en *2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS)*, feb. de 2020, págs. 133-139. DOI: 10.1109/ICoIAS49312.2020.9081844.
- [9] S. Sharma y R. Borse, “Automatic Agriculture Spraying Robot with Smart Decision Making,” en *Intelligent Systems Technologies and Applications 2016*, Cham, 2016, págs. 743-758. DOI: 10.1007/978-3-319-47952-1_60.
- [10] I. Beloev, D. Kinaneva, G. Georgiev, G. Hristov y P. Zahariev, “Artificial Intelligence-Driven Autonomous Robot for Precision Agriculture,” *Acta Technologica Agriculturae*, vol. 24, n.º 1, págs. 48-54, feb. de 2021. DOI: 10.2478/ata-2021-0008.
- [11] V. Saiz-Rubio, F. Rovira-Más, P. Broseta-Sancho y R. A. Aguilera-Hernández, “Robot-generated crop maps for decision-making in Vineyards,” en *2015 ASABE Annual International Meeting*, American Society of Agricultural y Biological Engineers, 2015, pág. 1.
- [12] G. B. P. Barbosa, E. C. Silva y A. C. Leite, “Robust Image-based Visual Servoing for Autonomous Row Crop Following with Wheeled Mobile Robots,” en *2021 IEEE 17th International Conference on Automation Science and Engineering*

- ring (CASE), ago. de 2021, págs. 1047-1053. DOI: [10.1109/CASE49439.2021.9551667](https://doi.org/10.1109/CASE49439.2021.9551667).
- [13] A. Durand-Petiteville, E. Le Flecher, V. Cadenat, T. Sentenac y S. Vougioukas, "Tree Detection With Low-Cost Three-Dimensional Sensors for Autonomous Navigation in Orchards»,” *IEEE Robotics and Automation Letters*, vol. 3, n.º 4, págs. 3876-3883, oct. de 2018. DOI: [10.1109/LRA.2018.2857005](https://doi.org/10.1109/LRA.2018.2857005).
- [14] Z. Liu, J. Chen, Z. Mei y C. Li, "ROS-based robot offline planning simulation system,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 711, n.º 1, pág. 012002, ene. de 2020. DOI: [10.1088/1757-899X/711/1/012002](https://doi.org/10.1088/1757-899X/711/1/012002).
- [15] Z. Kapić, A. Crnkić, E. Mujčić y J. Hamzabegović, "A web application for remote control of ROS robot based on WebSocket protocol and Django development environment,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1208, n.º 1, pág. 012035, nov. de 2021. DOI: [10.1088/1757-899X/1208/1/012035](https://doi.org/10.1088/1757-899X/1208/1/012035).
- [16] A. Smirnov y N. Teslya, "Robot Coalition Coordination in Precision Agriculture by Smart Contracts in Blockchain,” en *Agriculture Digitalization and Organic Production*, Singapore, 2022, págs. 271-283. DOI: [10.1007/978-981-16-3349-2_23](https://doi.org/10.1007/978-981-16-3349-2_23).
- [17] L. Solaque, A. Velasco y A. Riveros, "Planificación de trayectorias por técnica de A* y suavizado por curvas de Bezier para la herramienta del sistema de remoción de maleza de un robot dedicado a labores de agricultura de precisión,” *Entre Ciencia e Ingeniería*, vol. 12, n.º 24, págs. 43-51, 2018. DOI: [10.31908/19098367.3814](https://doi.org/10.31908/19098367.3814).
- [18] O. Tsymbal, A. Bronnikov y A. Yerokhin, "Adaptive Decision - making for Robotic tasks,” en *2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL)*, sep. de 2019, págs. 594-597. DOI: [10.1109/CAOL46282.2019.9019488](https://doi.org/10.1109/CAOL46282.2019.9019488).
- [19] M. A. Salichs, M. Malfaz y J. F. Gorostiza, "Toma de Decisiones en Robótica,” *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 7, n.º 4, págs. 5-16, oct. de 2010. DOI: [10.1016/S1697-7912\(10\)70055-8](https://doi.org/10.1016/S1697-7912(10)70055-8).
- [20] NVIDIA, "Jetson TX1 Module,” NVIDIA Developer, 1 de ago. de 2016 [Online]. Available: <https://developer.nvidia.com/embedded/jetson-tx1>.