Pontificia Universidad JAVERIANA Cali | Facultad de Ingeniería y Ciencias

Acta de Correcciones al Proyecto de Grado
Ingeniería

Fecha: 17/mayo/2023

Autores:
Juan David Jamioy Cabrera
Gabriel Alejandro Rodríguez Téllez

Nombre del Proyecto de Grado:
Arquitectura cognitiva aplicada en procesos de clasificación para un robot colaborativo

Director: Alexánder Martínez Álvarez
Co-Director: José Hernando Mosquera De La Cruz

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Alexander Martinez A.

Firma de Director del Proyecto de Grado

Firma Co-Director del proyecto de grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero Electrónico.

Camilo Rocha

Dr. Hernán Camilo Rocha Niño
Decano de la Facultad de Ingeniería y Ciencias

Dr. Luis Eduardo Tobón Llano
Director Carrera Ingeniería Electrónica.

Alexander Martinez A.

Dr. Alexánder Martínez Álvarez
Director del Trabajo de grado

MSc. José Hernando Mosquera De La Cruz
Codirector del Trabajo de grado

Sandra E. Nope R

Dra. Sandra Esperanza Nope Rodríguez
Jurado 1

Dr. Humberto Loaiza Correa
Jurado 2

Santiago de Cali, abril 26 de 2023

Doctor
Luis Eduardo Tobón Llano
Director
Carrera de Ingeniería Electrónica

Asunto: Entrega final trabajo de grado "Sistema de clasificación para un robot colaborativo utilizando una arquitectura cognitiva" y "Sistema de clasificación para un robot colaborativo utilizando técnicas de aprendizaje de máquina"

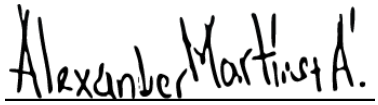
Cordial saludo.

Por medio de la presente nos permitimos presentar los trabajos de grado terminados bajo nuestra dirección: "SISTEMA DE CLASIFICACIÓN PARA UN ROBOT COLABORATIVO UTILIZANDO UNA ARQUITECTURA COGNITIVA" el cual fue desarrollado por los estudiantes de Ingeniería Electrónica Juan David Jamioy Cabrera (cod: 8943644) y Gabriel Alejandro Rodríguez Téllez (cod: 8943090) y "SISTEMA DE CLASIFICACIÓN PARA UN ROBOT COLABORATIVO UTILIZANDO TÉCNICAS DE APRENDIZAJE DE MÁQUINA" el cual fue desarrollado por los estudiantes de Ingeniería Electrónica María de los Ángeles Delgado Giraldo (cod: 8943648) y Juan Felipe Penagos Angrino (cod: 8945827).

Estos trabajos de grado fueron desarrollados bajo el marco del proyecto de investigación "APROXIMACIÓN A UNA ARQUITECTURA COGNITIVA PARA EL APRENDIZAJE Y GENERALIZACIÓN DE UN PROCESO DE CLASIFICACIÓN APLICADO EN UN ROBOT COLABORATIVO" el cual pertenece a la Convocatoria interna de proyectos "Por una universidad transformadora: Horizonte 2021-2025". Estos trabajos de grado comparten un núcleo común en lo referente a: 1) La tarea de identificación de objetos cúbicos por color; 2) La forma de adquirir la información de entrada (comandos gestuales y verbales) y 3) Las salidas del sistema (Realimentación auditiva, gráfica y control del robot UR3). Todos estos desarrollos fueron realizados de manera conjunta por los cuatro estudiantes trabajando en equipo.

Consideramos importante aclarar que la principal diferencia de los trabajos, la cual visibiliza el aporte individual de cada pareja de estudiantes, radica en el método de aprendizaje utilizado, ya que un trabajo utilizó una aproximación de la Arquitectura Cognitiva SOAR y el otro utilizó Redes Neuronales Artificiales (Perceptrón Multicapa MLP), para realizar el aprendizaje y la solución al problema de clasificación de objetos cúbicos propuesto.

Atentamente,



Alexander Martínez Álvarez, Ph.D.
Director

José Hernando Mosquera de la Cruz, M.Sc.
Codirector

Santiago de Cali, 25 de abril del 2023

Doctor
Luis Eduardo Tobón Llano
Director
Carrera de Ingeniería Electrónica

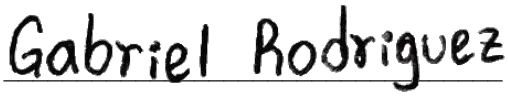
Cordial saludo.

Por medio de la presente nos permitimos presentarle el Trabajo de Grado titulado “Arquitectura cognitiva aplicada en procesos de clasificación para un robot colaborativo”.

Esperamos que este trabajo reúna todos los requisitos académicos, cumpla el propósito para el cual fue creado y sirva de apoyo para futuros proyectos relacionados con la materia.

Atentamente,

Juan David Jamioy Cabrera



Gabriel Alejandro Rodríguez Téllez

Arquitectura cognitiva aplicada en procesos de clasificación para un robot colaborativo

Juan David Jamioy-Cabrera, Gabriel Alejandro Rodríguez-Téllez

***Resumen*— Se desarrolló una arquitectura basada en el modelo de arquitectura cognitiva SOAR, la cual permite a los usuarios enseñarle tareas de clasificación de objetos cúbicos por color al robot colaborativo UR3 mediante interacción multimodal comandada por gestos y voz.

La evaluación de la arquitectura fue realizada por siete sujetos de prueba, a quienes se les indicaron los comandos gestuales y de voz para enseñarle al robot la tarea de clasificación por color de objetos cúbicos. A partir de estas interacciones se evaluó el desempeño, ejecutando tanto pruebas cuantitativas como cualitativas.

En las pruebas cuantitativas se ideó un protocolo que permitió evaluar un total de 588 interacciones verbales, 252 interacciones gestuales, y 63 interacciones verbales y gestuales. El porcentaje de reconocimiento de las interacciones fue del 98.41% para los comandos de voz, 81.35 % de las gestuales, y 80.95% de las multimodales. Además, el robot ejecutó la clasificación por color de objetos cúbicos el 100%.

En las pruebas cualitativas se les realizaron a los usuarios cinco preguntas sobre su percepción sobre las interacciones verbales, gestuales, y multimodales, utilizando como opciones de respuesta una escala *Likert*. Las encuestas mostraron una alta satisfacción sobre la arquitectura propuesta durante la interacción del usuario con el robot.

***Palabras Clave*— Arquitectura cognitiva, aprendizaje en robots, clasificación y robótica colaborativa.**

I. INTRODUCCIÓN

Cada vez es más común encontrar entornos de trabajo en donde colaboran humanos y robots para realizar una tarea, creando la necesidad de encontrar maneras más eficientes de interactuar y comunicarse, así como la forma en las que los robots puedes aprender a ejecutar una tarea a partir de la demostración de un humano[1].

Las arquitecturas cognitivas (AC) a diferencia de otras técnicas de aprendizaje, al tener la capacidad de incorporar reglas en la memoria de trabajo y conocimientos previos en las producciones pertenecientes a la memoria a largo plazo, permite un procesamiento más sofisticado y contextualizado de la información, lo que facilita su adaptabilidad y flexibilidad en diversas tareas de inteligencia artificial[2].

En la actualidad, existen diferentes arquitecturas cognitivas que han sido implementadas de forma satisfactoria, entre las cuales se encuentran arquitecturas basadas en el aprendizaje multimedia [3], [4], que se fundamenta en que el cerebro humano aprende de una forma más profunda si se integran palabras e imágenes, que cuando se usan solo palabras o imágenes por sí solas. Otras arquitecturas están basadas en aproximaciones a procesos biológicos del cerebro humano como ACT-R [5], procesos de cognición humana que estudian la relación percepción-acción como ICARUS [6] y arquitecturas basadas en la distribución de las memorias humanas como SOAR [7], la que destaca al ser la única que explícitamente considera los mecanismos de aprendizaje de nuevas tareas.

La implementación de arquitecturas cognitivas aplicadas en robots colaborativos permite enseñar y ejecutar tareas mediante la transmisión de información, utilizando diferentes modalidades como gestos corporales y comandos de voz, simplificando la interacción para personas no experimentadas y avanzando hacia el camino de una interacción natural [8], [9].

En la Tabla 1 se presenta una comparativa con otros desarrollos de aplicaciones robóticas reales o simuladas, mediante interacciones comandadas por gestos, voz o sensores, las cuales utilizan arquitecturas cognitivas como ICARUS, SOAR o ACT-R.

Tabla 1. Tabla comparativa de desarrollos multimodales con arquitecturas cognitivas.

Referencia	Aplicación robótica		Tipo de interacción			Arquitectura cognitiva	Realimentación	
	Real	Simulada	Voz	Gestos	Sensores		Audio	Visual
[10]	x			x	x	SOAR		x
[11]	x			x		SOAR		x
[20]		x			x	SOAR		x
[21]	x			x	x	ACT-R		x
[22]	x		x	x		ACT-R		x
[23]	x		x			ICARUS	x	
Arquitectura propuesto en este trabajo	**x**		**x**	**x**		**SOAR**	**x**	**x**

La revisión bibliográfica realizada muestra que la mayoría de sistemas fueron probados sobre aplicaciones robóticas reales, utilizando gestos como principal medio de interacción y arquitectura cognitiva SOAR. En los tipos de interacción se presentan casos de multimodalidad basada en gestos y sensores como [10] y [11], mientras que en [12] se presenta un sistema comandado por gestos y voz como el propuesto en este trabajo.

En [10], se utilizó la arquitectura SOAR para controlar la selección de marcha en un robot hexápodo, y el aprendizaje multimedia involucrando datos multimodales provenientes de una cámara en espectro visible, sensores de ultrasonido, sensores de fuerza en la base de las patas, un receptor de GPS y una brújula eléctrica. Del mismo modo, en [13] se utilizó esta misma arquitectura cognitiva en un sistema robótico automatizado que retira diferentes tipos de tornillos de las cajas en los computadores portátiles, a partir de la información provista por dos cámaras web y un sensor Kinect.

Este artículo describe cómo se ajustó una arquitectura cognitiva SOAR para que un robot colaborativo UR3 aprendiera una tarea de clasificación de objetos, a partir de la interacción multimodal (gestos y voz) con humano. Para ello, el artículo inicia con la descripción de la metodología seguida, continúa con la descripción de cómo se ajustó la arquitectura para el aprendizaje mediante interacción multimodal. El

apartado 4 se presentan los resultados alcanzados, y se discuten en el apartado 5. Finalmente, en el apartado 6 se exponen las conclusiones y trabajos futuros.

II. METODOLOGÍA

Esta sección inicia con la especificación de la tarea de clasificación, asociándola con el espacio de trabajo y los objetos utilizados. Posteriormente se describen las interacciones gestuales, verbales y multimodales, requeridos para enseñar la tarea de clasificación y que el robot logre ejecutar la tarea. El apartado finaliza con la descripción de las pruebas que se plantearon para validar la arquitectura, incluyendo información sobre la población.

Tarea de clasificación, Espacio de trabajo, y Objetos: se eligió la tarea de clasificación de objetos cúbicos por color, problema similar al que ha sido abordado por [14] en el que mediante la implementación de un sistema que clasifica objetos por color en tiempo real para la selección de granos de café en una FPGA, proporciona una solución al desafío real de seleccionar los granos de café de acuerdo a su etapa de madurez. Este enfoque destaca la importancia de este tipo de tareas y su contribución al aspecto económico de una industria específica.

La tarea de clasificación se ejecutó sobre una mesa de trabajo metálica de 90.0 cm por 50.0 cm, sobre la que se demarcó con cinta roja una zona de trabajo de 33.2 cm por 23.2 cm. La zona de trabajo se subdividió en tres filas y cuatro columnas, conformando una retícula de doce zonas, cada una de aproximadamente 8.0 cm x 7.4 cm, esta característica no se evidenció en los trabajos consultados. La ubicación de la zona de trabajo, el robot manipulador y la cámara, garantizan la observación ininterrumpida del centro de la cara superior de los cubos durante todo el movimiento del robot, y además, el que la pinza realice un agarre cómodo de los cubos dentro de la zona de trabajo, dado que esta se mantiene en todo momento orientada perpendicularmente a la mesa de trabajo y con la apertura suficiente para tomar el cubo independiente de la orientación que este tenga como se puede observar en la Figura 1. La ubicación de la cámara es a 50.0 cm por encima de la mesa, alejada horizontalmente del espacio de trabajo 10.0 cm, y frente al robot colaborativo a una distancia horizontal de aproximadamente 33.2 cm cuando este se encuentra en su posición inicial.

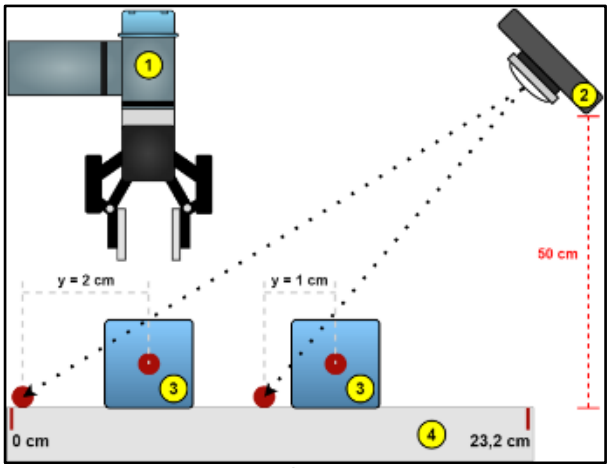


Figura 1. Ilustración del espacio de trabajo.

Debido a la orientación constante de la pinza que el brazo mantiene durante la ejecución de los movimientos, al llegar a su máxima extensión sin alterar dicha orientación, no logra entrar en contacto con la cámara. Esto no solo evita colisiones, sino que también permite que la cámara se encuentre más cerca del espacio de trabajo en comparación con su ubicación directamente encima de este.

Los objetos utilizados fueron cubos impresos en 3D con material PLA negro, de 2 cm de lado, a los que se les pegó en una de sus caras stickers azules (3) y verdes (3). En la disposición inicial de los objetos, se puede elegir emplear entre 1 y 3 cubos de cada color, y se ubica cada uno de ellos dentro de una de las celdas de la retícula, en cualquier posición y orientación. Como resultado se espera que el robot manipulador organice, ya sea en columnas o filas, los cubos de cada color, cada uno de ellos dentro de celdas diferentes de la retícula. La Figura 1 presenta un ejemplo, donde la imagen de la izquierda corresponde a la distribución inicial de 6 cubos, 3 de cada color, note que efectivamente el cubo puede tener cualquier ubicación dentro de una celda de la retícula; la imagen a la derecha corresponde a la clasificación en columnas realizada por el robot, que por lo general ubica los cubos centrados en las celdas. A las imágenes de la Figura 2 se les adicionó unas líneas que permiten detallar la ubicación relativa de los cubos, pero estas divisiones no existen físicamente en la zona de trabajo. Así mismo, en la parte superior izquierda de cada celda hay un número en gris claro, que se relaciona con la identificación de la zona usada en los comandos verbales y gestuales.

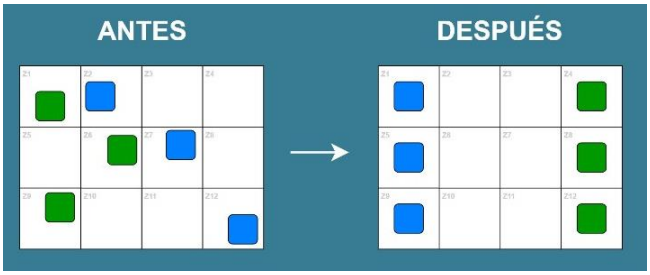


Figura 2.Descripción de la tarea de clasificación.

El reconocimiento de la zona de trabajo, la retícula y los cubos, se realizó un sistema de visión por computador, que empleó una cámara en espectro visible Logitech C920, con una resolución

estándar de 640x480 pixeles y corrección de iluminación automática, en condiciones de ruido e iluminación tipo oficina. Así mismo, permite la identificación de los gestos establecidos para la tarea de clasificación por color de cubos, los cuales se explican a continuación. En todos los casos, participaron siete jóvenes sin entrenamiento previo, quienes ejecutaron cuantitativamente un total de 588 expresiones verbales, 63 veces el comando de voz, 63 veces el comando multimodal y 27 clasificaciones por color utilizando el sistema completo.

Interacción gestual: Se definió un gesto denominado "seleccionar", el cual está compuesto por la transición entre dos posturas. La primer postura corresponde a la mano abierta, manteniendo una separación de al menos 1.5 cm entre la punta del dedo índice y la punta del dedo pulgar. En la segunda postura, los dedos índice y pulgar se unen formando una especie de circunferencia, mientras la mano permanece abierta. Esta acción se asemeja a la forma en que una persona intenta agarrar un objeto pequeño utilizando únicamente estos dos dedos.

Al ejecutar este gesto sobre o cerca de un cubo por primera vez, se identifica el objeto con el cual se desea realizar el desplazamiento. Posteriormente, al realizar una ejecución adicional de este gesto dentro de una zona específica, se determina la ubicación a la cual se desea desplazar dicho objeto.

Interacción verbal: Se definió un diccionario en el que fueron incluidas 27 palabras comunes, asociadas a las acciones de "Tomar" y "Dejar", así como la descripción del color del cubo y su "Ubicación", o la descripción de la zona. El diccionario recopila diferentes expresiones comunes en español, que podrían ser usadas durante la interacción humano robot como saludos, enseñanza y ejecución de una nueva tarea de clasificación. A partir del diccionario, se pueden reconocer expresiones como: "*toma el cubo verde dos y ponlo en la zona once".*

Interacción multimodal: se presenta cuando para entender una acción se debe ejecutar simultáneamente un comando de voz y un gesto. Se definió un comando multimodal, que necesita que se replique la oración "Este cubo ponlo aquí" mientras se ejecuta el gesto "seleccionar" dos veces. La primera debe ser lo más cerca posible al cubo que se desea mover, y la segunda dentro de la zona a la que se desea mover el cubo, la expresión verbal en este caso es la que le indica a la arquitectura que tendrá que tomar los valores que se almacenaron al ejecutar el gesto "seleccionar". Un ejemplo es: *"Este cubo (*gesto seleccionar sobre cubo azul uno*) ponlo aquí (*gesto seleccionar sobre la zona cinco*)".*

Validación del sistema: Como resultado, se evaluaron un total de 588 expresiones verbales, 63 veces el comando de voz, 63 veces el comando multimodal.

La validación del sistema se realizó en dos partes. La primera es una prueba cuantitativa en la que se verificó si el sistema pudo reconocer las interacciones gestuales, verbales y multimodales. La segunda es una prueba cualitativa que pretende medir, mediante encuestas, el nivel de satisfacción de los usuarios frente a las interacciones. La encuesta recoge 35 preguntas cuyas opciones de respuesta corresponden a la escala *Likert de cinco niveles* [15].

III. DESCRIPCIÓN DE LA ARQUITECTURA

La arquitectura propuesta se basa en la arquitectura cognitiva SOAR [7], ya que como se explicó anteriormente, es una arquitectura que puede adaptarse a la realización de interacciones multimodales, y además, es altamente escalable. La Figura 3 presenta la arquitectura SOAR, mientras que la Figura 4 la adaptación realizada para realizar interacciones multimodales, según [10], las principales limitaciones de este sistema provienen de las entradas. La razón principal de esta elección es su naturaleza modular y su capacidad considerar los mecanismos de aprendizaje de nuevas tareas, lo que le permite adaptarse a diferentes entornos y la inclusión de nuevas funcionalidades, haciéndola.

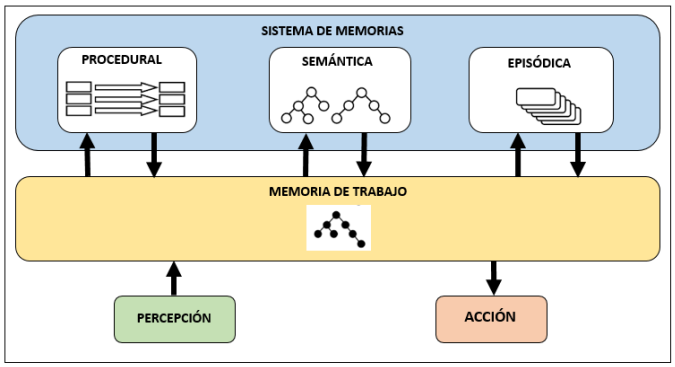


Figura 3. Arquitectura SOAR.

La arquitectura SOAR está compuesta por cuatro módulos; un módulo de *sistema de memorias de largo plazo* (procedural, semántica y episódica) que es el encargado de almacenar hechos, reglas y principios que representan el conocimiento de la arquitectura, un módulo de *memoria de trabajo* que integra la información proveniente de las diferentes fuentes (percepción y sistema de memorias) para realizar comparaciones y evaluaciones que dan lugar a las acciones que se deben realizar para cumplir con un objetivo dado, un módulo de *percepción* que es el encargado de recibir y procesar la información sensorial del entorno y un módulo de *acción* que se encarga de seleccionar y ejecutar las acciones para garantizar que se cumpla con la tarea solicitada.

En la figura 4 se presenta la arquitectura propuesta, que está constituida por cuatro módulos: *Entrada, Administrador del diálogo, Sistema de memorias y Salida.*

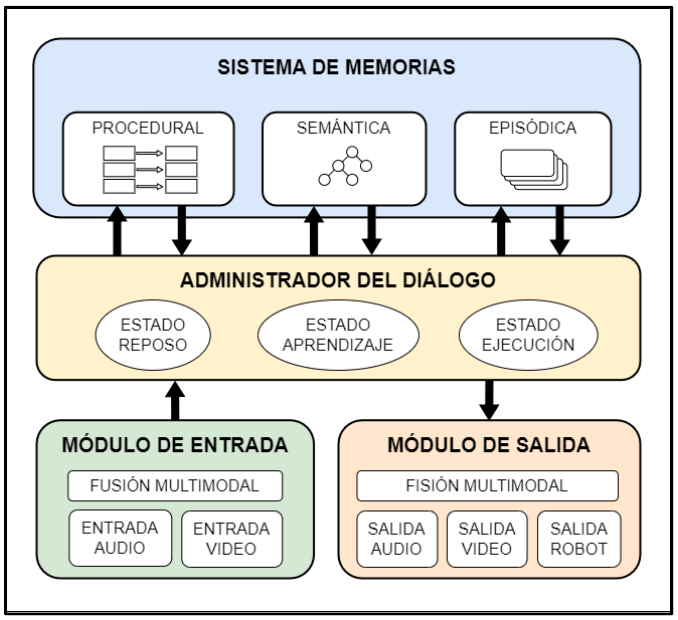


Figura 4. Arquitectura propuesta.

Esta arquitectura propone las siguientes modificaciones respecto a la arquitectura SOAR: 1) Integración de dos modalidades de entrada (audio y video) lo cual se refleja al cambiar el módulo de *percepción* por un *módulo de entrada* para capturar en simultaneo y fusionar las señales de audio y video. 2) Integración de tres modalidades de salida (audio, video y robot) lo cual se refleja al cambiar el módulo de *acción* por un *módulo de salida* para realimentar al usuario de forma multimodal y accionar el robot. 3) La integración de tres estados de interacción; reposo, aprendizaje y ejecución los cuales controlan el flujo de la interacción humano-robot, esto se refleja al modificar la *memoria de trabajo* por el *administrador del dialogo*.

La arquitectura fue implementada utilizando como IDE *Visual Studio Code* y lenguaje de programación *Python 3.9.12*, corriendo sobre un sistema operativo *Windows 10 Pro* de 64 bits. A continuación, se presenta una descripción detallada de cada uno de los módulos implementados.

A. Módulo de Entrada

Este módulo está compuesto por el sub-módulo de *entrada de audio* encargado de realizar el reconocimiento de comandos verbales, el sub-módulo de *entrada de video* encargado de identificar el color, la ubicación de los cubos en la retícula y el gesto "seleccionar", y el sub-módulo de *fusión multimodal*, encargado de priorizar e integrar las señales verbales y gestuales.

1) Entrada de Audio

Para la captura de audio, se utilizó una diadema inalámbrica de la marca y modelo Logitech G935, la cual está equipada con un micrófono de patrón de captación cardiode (unidireccional) y una respuesta en frecuencia que abarca el rango de 100 Hz a 10 kHz. Este módulo permite reconocer diversas estructuras de comandos de voz para la tarea de clasificación por colores. Las estructuras se conforman mezclando seis componentes principales, los cuales se detallan en la Tabla 2. Los cuatro primeros resaltados en verde son necesarios para identificar el objeto (cubo) asociado a la acción "Tomar"; mientras que los tres últimos son necesarios para determinar la ubicación asociada a la acción "Dejar". Un ejemplo de estructura que puede usarse es: "coge *el cubo azul tres y muévelo a la zona cuatro*". Con estos 27 componentes se podrían armar 1152 estructuras diferentes

Componentes de la estructura del comando de voz implementado						
Tomar	**Objeto**	**Color**	**Índice**	**Dejar**	**Ubicación**	
['toma', 'sujeta', 'coge', 'agarra']	['cubo']	['azul', 'verde']	['uno', 'dos', 'tres']	['ponlo', 'colócalo', 'déjalo', 'muévelo']	['zona']	['uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis', 'siete', 'ocho', 'nueve', 'diez', 'once', 'doce']

Tabla 2. Estructura del comando de voz para la instrucción verbal.

Cuando se identifica la presencia de una entrada de audio, inicialmente se convierte la estructura recibida en una cadena de texto, se separa cada palabra en el texto y se compara cada una con las componentes de la Tabla 2. De este modo, no son tenidos en cuenta artículos y preposiciones que son usados en español para conformar una oración, y se preservan aquellas con información relevante. Así mismo, debe encontrarse al menos una componente de cada una de las seis categorías para tener la información mínima requerida para entender un comando verbal, de lo contrario será identificado como comando no válido, y el usuario deberá repetirlo, incidiendo negativamente en la satisfacción del usuario con el sistema.

Para el reconocimiento de los comandos de voz se utilizaron las librerías *PyAudio 0.2.13* [16] y *Speech Recognition 3.9.0* [17], que a su vez emplean el motor de reconocimiento de voz Google Speech Recognition [18].

2) Entrada de Video

A través de la entrada de video se reconoce el área de trabajo, los objetos y los gestos ejecutados por la mano, para ello, se usaron las siguientes estrategias:

Dado que el espacio de trabajo se delimitó usando una cinta de color rojo, se usó la información de color para encontrarlo; para ello, la imagen RGB capturada por la cámara Logitech C920 se convierte al espacio de color HSV utilizando funciones proporcionadas por la librería OpenCV. Este proceso es realizado por la misma librería aplicando las ecuaciones (1), (2) y (3).

$$H \leftarrow \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)} & if\ V = R \\ 120 + \frac{60(B-R)}{V-\min(R,G,B)} & if\ V = G \\ 240 + \frac{60(R-G)}{V-\min(R,G,B)} & if\ V = B \\ 0 & if\ R = G = B \end{cases} \quad (1)$$

$$S \leftarrow \begin{cases} \frac{V-\min(R,G,B)}{V} & if\ V \neq 0 \\ 0 & otherwise \end{cases} \quad (2)$$

$$V \leftarrow \max(R,G,B) \quad (3)$$

Según [19] los valores se convierten dependiendo del tipo de dato de destino, para el caso de imágenes de 8 bits se realiza como se presenta en (4), (5) y (6).

$$H \leftarrow \frac{H}{2} \ (para\ ajustar\ de\ 0\ a\ 255) \tag{4}$$

$$S \leftarrow 255S \tag{5}$$

$$V \leftarrow 255V \tag{6}$$

Uno de los desafíos de los espacios de color lineales, como RGB, es que sus coordenadas individuales no capturan de manera intuitiva las percepciones humanas del color. No existe una relación local directa entre las coordenadas que permita representar adecuadamente el matiz. Para abordar esta limitación, se utiliza comúnmente el espacio de color HSV (Tono, Saturación y Valor). El espacio de color HSV proporciona una representación más apropiada para definir filtros, ya que se pueden mantener constantes el valor (o luminosidad) y la saturación, mientras se varía únicamente el tono. Esto simplifica la definición de filtros, ya que solo se requiere una componente en lugar de tres [20].

Teniendo en cuenta lo anterior, se escogió el espacio de color HSV, debido a que es posible relacionar de manera más intuitiva las tonalidades, lo que lo convierte en una elección estándar para aplicaciones donde la manipulación y selección de colores son importantes. Es entonces que gracias al contraste entre la zona metálica y la cinta roja, los pixeles que delimitan el área de trabajo pueden encontrarse usando la regla que se muestra en (7).

$$\begin{aligned} \text{sí } ((0 \leq H \leq 5) o (175 \leq H \leq 179)) \& (100 \leq S \\ \leq 255) \& (20 \leq V \leq 255) \rightarrow \\ \in \text{pixeles\_cinta} \end{aligned} \tag{7}$$

Con el filtro resultante de aplicar la regla presentada en (7) se logra identificar únicamente el rectángulo rojo correspondiente al espacio de trabajo como se presenta en la figura 5 (Derecha).

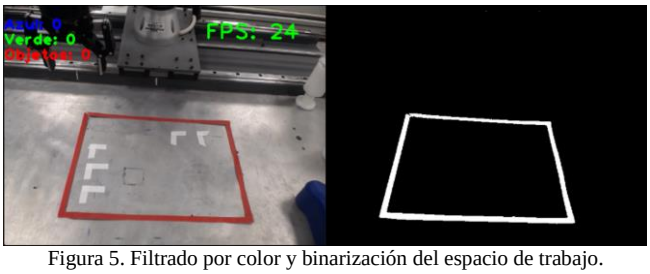


Figura 5. Filtrado por color y binarización del espacio de trabajo.

Dado que se conoce la forma y tamaño del espacio de trabajo, se puede realizar a la imagen una transformación en perspectiva, de manera que el área de trabajo se observe como si el campo de visión de la cámara fuese paralelo a la mesa. Lo anterior se logró aplicando a la imagen de entrada la transformación dada por la matriz M en la ecuación 8.

$$M = \begin{bmatrix} -2{,}6163 * 10^{0} & -1{,}8080 * 10^{-1} & 1{,}4692 * 10^{3} \\ -5{,}3953 * 10^{-1} & 2{,}5701 * 10^{0} & 5{,}5277 * 10^{1} \\ 2{,}0502 * 10^{-4} & 6{,}6274 * 10^{-4} & 1{,}0000 * 10^{0} \end{bmatrix} \tag{8}$$

Para esto, se partió de la imagen binarizada, a la cual se le realizó una extracción de contornos, acompañada de una aproximación poligonal de cuatro lados que da como resultado dos rectángulos, uno correspondiente al borde externo y otro correspondiente al borde interno. Luego se selecciona el rectángulo con menor área y en este con la ayuda de la opción que ofrece openCV en las funciones de búsqueda de contornos para encontrar puntos no redundantes, se encuentran las esquinas de la zona de trabajo, con las cuales se realiza la transformación de perspectiva.

La figura 6 (izquierda) corresponde a la captura original de la cámara, en donde se observa sobre la mesa el recuadro rojo que delimita el espacio de trabajo, mientras que en la Figura 7 (derecha), se observa el espacio de trabajo después de aplicarle la transformación de perspectiva, al que se le sobreponen líneas que permiten identificar las 12 zonas de la retícula y su número de identificación.

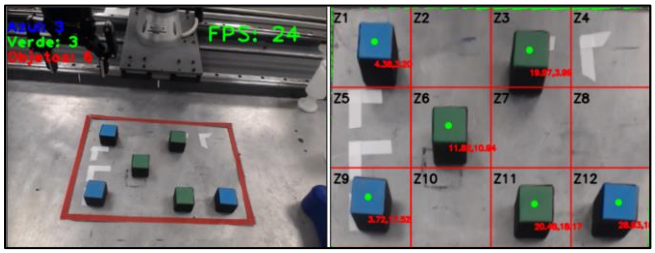


Figura 6. Transformación de perspectiva.

De forma similar a la identificación por color del área de trabajo, en cada celda de la retícula se busca la presencia o no de un cubo de acuerdo con su color, así, se usaron las siguientes reglas para identificar el color azul (9) y el verde (10):

$$\begin{aligned} \text{sí } (100 \leq H \leq 107) \& (150 \leq S \\ \leq 255) \& (20 \leq V \leq 255) \\ \rightarrow \text{ color azul} \end{aligned} \tag{9}$$

$$\begin{aligned} \text{sí } (47 \leq H \leq 99) \& (50 \leq S \\ \leq 230) \& (20 \leq V \leq 230) \\ \rightarrow \text{ color verde} \end{aligned} \tag{10}$$

La Figura 7 (izquierda) se puede observar la visión de la cámara con la transformación de perspectiva, mientras que en la imagen de la Figura 7 (derecha) se puede apreciar la imagen con la segmentación de los objetos (cubos) de color azul.

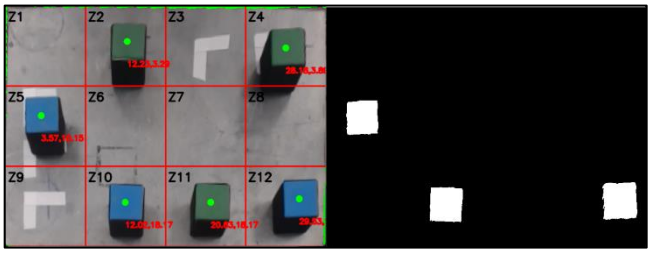


Figura 7. Cubos en espacio de trabajo (izquierda) y binarización (Derecha).

Utilizando las imágenes binarizadas, se llevó a cabo un proceso para identificar los contornos mediante el uso de funciones proporcionadas por la biblioteca OpenCV. Esto permitió la numeración de los objetos cúbicos de color x (según el filtro aplicado) presentes en la escena. A partir de los contornos encontrados, se procedió a calcular los centroides de los cubos aprovechando la capacidad de OpenCV para calcular momentos invariantes de una imagen. Para el cálculo de las coordenadas (x, y) del centro de los objetos cúbicos, se realizó la siguiente operación:

$$x = \frac{n(1,0)}{n(0,0)} \tag{11}$$

$$y = \frac{n(0,1)}{n(0,0)} \tag{12}$$

Donde n(1,0) y n(0,1) son los momentos de primer orden y n(0,0) es el momento de área.

Cabe destacar que las coordenadas obtenidas de los centroides de los cubos están en formato de píxeles. Para convertirlas a centímetros, se realizó una conversión utilizando una regla de tres simple. Se consideraron los valores del ancho y largo del espacio de trabajo tanto en la escala física como en la digital, teniendo en cuenta que la imagen digital tenía una resolución de 640x480 píxeles. Como resultado, se obtuvieron los siguientes factores de conversión presentados en (13) y (14), donde los subíndices (cm) y (px) indican las unidades en las que se expresa la componente de la coordenada correspondiente.

$$x_{(cm)} = \frac{x_{(px)} * 1\ cm}{19.87\ px} \tag{13}$$

$$y_{(cm)} = \frac{y_{(px)} * 1\ cm}{21.57\ px} \tag{14}$$

Experimentalmente se observó una variación entre la ubicación real en el espacio de trabajo para los centros de los cubos y la calculada por el algoritmo propuesto, esto es debido a las líneas de proyección visual de la cámara respecto al espacio de trabajo las cuales se presentan como *líneas punteadas* en la figura 1. Para corregir este problema se tomaron mediciones milimétricas sobre la ubicación real y calculada, para realizar un ajuste correctivo como se presenta en la figura 8, en la que se puede observar en una vista superior. el espacio de trabajo de color gris oscuro en (4) y el robot UR3 en (1), sobre este espacio de trabajo se pueden observar unas particiones con líneas rojas en las que internamente se muestran los ajustes que se realizaron en las coordenadas para cada una.

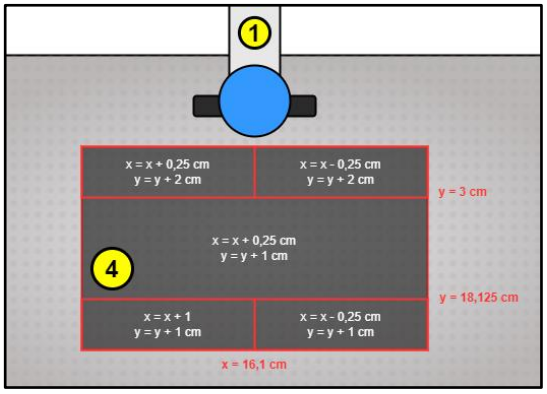


Figura 8. Ilustración de las particiones del espacio de trabajo sobre las que se realizaron ajustes a las coordenadas identificadas.

Una vez se establecen la cantidad de cubos por color y su ubicación inicial en la zona de trabajo, el usuario humano puede empezar a interactuar con los cubos. Cuando la interacción corresponde a la gestual, el primer proceso que se realiza es la segmentación de la mano con base en los datos de la imagen utilizando tanto modelos de aprendizaje automático como datos estáticos para la generación de puntos de referencia en la mano Una vez la mano es segmentada, se aplica una operación morfológica de esqueletización, la cual se implementó haciendo uso de la librería *Mediapipe Hands* [21]. Como resultado, se obtienen 20 puntos (ver Figura 9). Note que los dedos índice y pulgar asociados a los gestos corresponden a los puntos 5-8 y 1-4, respectivamente. En particular, se estableció que los dedos están cerrados (realizando un agarre) cuando la distancia euclidiana entre los puntos 4 y 8 es de 30 pixeles o menos, de lo contrario, se considera que los dedos están abiertos. La Figura 10 presenta un ejemplo de detección de los dedos abiertos (imágenes superiores) y de dedos cerrados (imágenes inferiores). Se muestra al lado izquierdo sobre la imagen capturada por la cámara, y al lado derecho sobre la zona de trabajo con corrección de perspectiva.

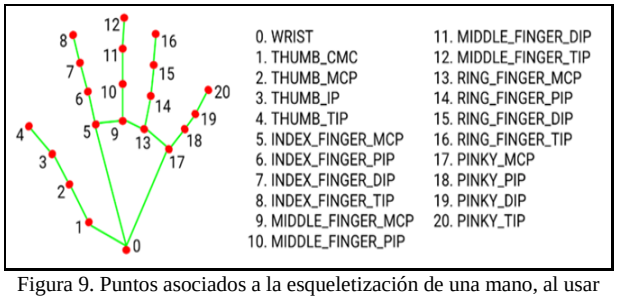


Figura 9. Puntos asociados a la esqueletización de una mano, al usar Mediapipe Hands.

A partir del esqueleto de la mano, se establece la distancia entre las puntas de los dedos índice y pulgar para identificar si la postura de la mano es abierta o cerrada, estableciendo empíricamente un umbral de 30 pixeles para identificar la postura como "cerrado" (Figura 10, abajo), o en caso contrario como “abierto” (Figura 10, arriba). Así mismo, se dibuja en pantalla una línea entre ambas puntas, en la cual se muestra centralmente un punto denominado el “cursor digital” que sirve para guiar al usuario como si fuese un puntero para seleccionar el cubo o la zona, se puede ver ejemplo de esto en la Figura 10 costado derecho. En el recuadro superior se pueden apreciar las

dos vistas ofrecidas al usuario en las que la postura de la mano es “abierta”, en la parte izquierda se puede observar solo el “cursor” y la línea sobre la que se ubica, mientras que en la parte derecha se pueden observar la retícula superpuesta y tres líneas que salen del cursor hacia los cubos que se encuentran en el espacio de trabajo, dos de estas son verdes y una es roja, esta última indica cuál de los cubos está más cerca al cursor y de esta manera al ejecutar el gesto “seleccionar” este cubo será el elegido. En el recuadro inferior se presenta la postura de la mano “cerrada” en la que se puede observar lo que se explicó anteriormente.

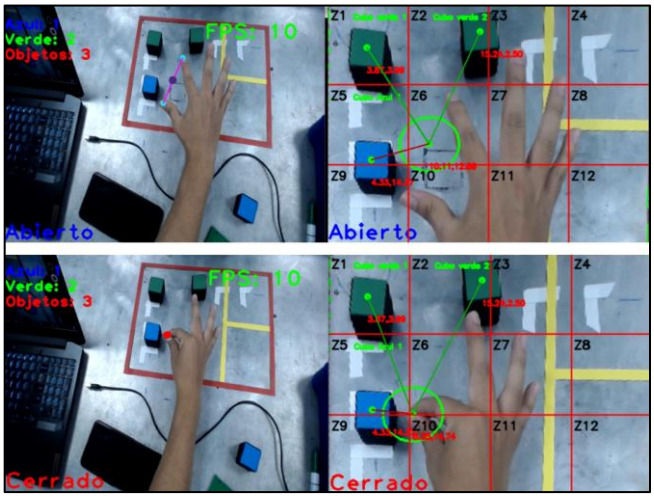


Figura 10. Identificación de la postura “abierto” y “cerrado” implementada.

En la figura 11 se presenta el robot tomando un cubo de color verde ubicado en la zona 10, de igual forma, se presenta un computador portátil ejecutando el procesamiento de video; al lado derecho de la pantalla del computador se presenta la imagen original capturada por la cámara y al izquierdo el espacio de trabajo con la realimentación gráfica de la retícula. Se puede observar que, incluso cuando la pinza del robot está justo encima del cubo y a punto de tomarlo, todavía es posible reconocer la cara superior del mismo.


Figura 11. Espacio de trabajo y procesamiento de video.

En la implementación de este módulo “Entrada de Video” se utilizó la librería *OpenCV-Python 4.7.0.68* [21].

3) Fusión multimodal

Este sub-módulo tiene dos funciones. Por un lado, identifica cuando se está realizando una interacción multimodal, y por otro. prioriza e integra las interacciones verbales y gestuales que provienen de los sub-módulos de *Entrada de Audio* y *Entrada de Video*.

Se implementó el comando multimodal; *“Mira este cubo (gesto seleccionar sobre un cubo) ponlo aquí (gesto seleccionar sobre una zona de la retícula)”*, el cual permite realizar la selección y desplazamiento de un cubo hacia una zona específica. Se realizó una implementación por hilos para que los sub-módulos de entrada siempre se estén ejecutando al mismo tiempo, permitiendo recolectar y actualizar la información visual que se comparte a los demás módulos de manera constante, y al mismo tiempo mostrar al usuario la realimentación visual sin interrupción. Finalmente, la fusión multimodal calcula en paralelo la interacción verbal y gestual, pero le da prioridad a la interacción verbal, de manera que solo al identificar el comando; *“Mira, este cubo, ponlo aquí”*, es cuando se extrae la información visual correspondiente al cubo y zona de trabajo indicados por el gesto “seleccionar” completando el comando multimodal.

B. Administrador del diálogo

Este módulo envía y recibe información al sistema de memorias que está compuesto por las memorias *semántica*, *episódica* y *procedural*. De igual forma recibe la información multimodal proveniente de la etapa de *fusión multimodal* y, con base en esta información, se encarga de definir las acciones a realizar dependiendo del estado de interacción: *estado de reposo*, *estado de aprendizaje* y *estado de ejecución*.

1) Estado de reposo

En el estado de reposo el algoritmo se encuentra constantemente en escucha a la espera de reconocer la palabra *“hola”* como señal de activación; como respuesta, el algoritmo pregunta al usuario por el estado que desea inciar, reproduciendo un sonido con la oración: *“hola, ¿deseas que aprenda o ejecute una tarea?”*. En usuario debe responder con la palabra *“Aprender”* o *“Ejecutar”*, dando inicio al estado correspondiente.

2) Estado de aprendizaje

Inicia si la respuesta del usuario a la pregunta fue *“Aprender”*. Lo primero que hace la arquitectura, es identificar los cubos presentes en la escena, sus colores y las zonas de la retícula en la que se ubican. A continuación, se debe identificar una instrucción verbal o multimodal del usuario para cada cubo presente en el espacio de trabajo. Cuando se completan las instrucciones para cada cubo, se determinan las posiciones finales en las que quedaron cada uno de los cubos, y se almacena en la *memoria episódica; así mismo,* se realimenta auditivamente al usuario con la frase *"Tarea aprendida"*, y la arquitectura vuelve al estado de reposo.

3) Estado de ejecución

Inicia si la respuesta del usuario a la pregunta en el estado de reposo fue "Ejecutar". En este caso, este módulo solicita el *vector de posición inicial* a la *memoria Semántica* para saber cuántos cubos de cada color están presentes en el espacio de trabajo, una vez obtenida esta información, solicita a la *memoria episódica* una solución con la disposición final para cada cubo. A continuación entra en acción la *memoria procedural*, que verifica que las zonas en las que deben ubicarse los cubos estén libres; cuando no es así, se adiciona a la secuencia de movimientos dada por la memoria episódica movimientos que desplacen los cubos a zonas diferentes a las requeridas por la solución. . Finalmente, se envían al *módulo de salida* las acciones de desplazamiento para cada cubo de manera secuencial.

C. Sistema de Memorias

El sistema de memorias está compuesto por tres memorias: *memoria semántica*, *memoria episódica* y *memoria procedural*.

1) Memoria Semántica

Es la memoria empleada en los estados de aprendizaje y ejecución para interpretar y organizar el espacio de trabajo tomando como base la información multimodal proveniente del *administrador del dialogo* y arroja como salida un *vector de posiciones* que contiene la cantidad de cubos identificados, color y posición.

2) Memoria Episódica

Esta memoria debe contener al menos una solución para las posibles combinaciones del número de cubos por colores en el espacio de trabajo (9), las cuales se almacenan en un archivo de texto plano (.csv). Posterior al *estado de aprendizaje* se adiciona una nueva solución al archivo, si esta no existe previamente. Durante el *estado de ejecución*, se selecciona aleatoriamente una solución acorde con la cantidad y color de los cubos presentes en el espacio de trabajo, y la envía al *administrador del dialogo*.

3) Memoria Procedural

Esta memoria es utilizada sólo en el *estado de ejecución* para establecer la secuencia de movimientos que debe realizar el manipulador para clasificar por colores los cubos presentes en el área de trabajo. Seleccionada una solución de la memoria episódica, la memoria procedural debe ajustar la secuencia de movimientos para ejecutar de manera correcta la solución. Para ello, se desarrolló una funcionalidad que valida que las zonas en las que se van a ubicar los cubos se encuentren libres. En caso de encontrar cubos en las zonas correspondientes a las posiciones finales de la solución, esta memoria adiciona movimientos que garanticen que previamente se desplaza el cubo que ocupa la zona del siguiente movimiento, en caso de ser necesario. La secuencia de movimientos a realizarse se envía al módulo de salida.

Durante el *estado de aprendizaje*, esta memoria se encarga de procesar las instrucciones dadas por el usuario en cada demostración y dar las órdenes de movimiento correspondientes al robot, para que este pueda desplazar los cubos siguiendo las indicaciones del usuario.

D. Módulo de salida

Este módulo está compuesto por cuatro sub-módulos.

1) Salida de audio

Es el sub-módulo encargado de realizar una realimentación auditiva al usuario, reproduciendo alguna de las oraciones en la Tabla 3, en momentos específicos. El diccionario de síntesis de voz se presenta en la Tabla 3, y está compuesto por 15 expresiones. Para realizar la síntesis de voz se utilizó la librería de conversión de texto a voz *Pyttsx3 2.90* [17], se seleccionó el intérprete de voz en español y para establecer una forma ordenada de reproducir las oraciones se crearon dos tipos de expresiones para reproducir, una es *solicitudes* y la otra es *respuestas*.

Tabla 3. Diccionario de síntesis de voz.

Función	Síntesis de Voz
Solicitudes	'¿Quieres que aprenda una tarea o que la ejecute?'
	'Error, comando de voz no identificado'
Respuestas	'Hola', 'Hola como estás', 'Hola, que tal', 'Es bueno volver a verte', 'Regresaste', '¿Qué tal?'
	'La tarea se ejecutó con éxito'
	'Enséñame'
	'Tarea aprendida'
	'La tarea no fue reconocida, por favor repita nuevamente'
	'Siguiente instrucción'
	'No se ha encontrado una tarea para ejecutar. Por favor revisa que la cantidad de cubos sea correcta, ¿Quieres que aprenda o ejecute una tarea?'
	'Error, comando de voz no identificado'

2) Salida de video

Este sub-módulo es el encargado de brindar una realimentación gráfica al usuario en la pantalla del equipo de cómputo utilizado; ahí se visualizan dos imágenes a las que se les sobrepone información. En el caso de la imagen de la cámara a la izquierda, se le sobreponen mensajes con el número de cubos identificados y el estado de los dedos. En la imagen con el área de trabajo con corrección de perspectiva, se sobreponen líneas para ver la retícula, en la parte superior izquierda aparece el identificador de las zonas, el centro de cada cubo en el área de trabajo y un mensaje con el color de cubo reconocido, el estado reconocido de los dedos en la parte inferior izquierda.

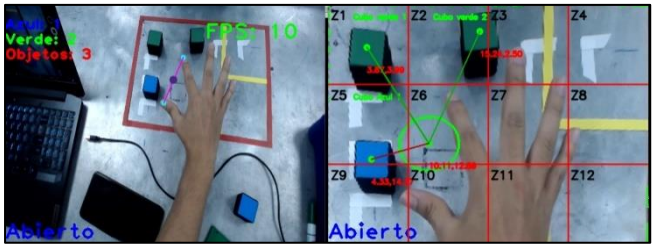


Figura 12. Realimentación visual del sistema.

3) Salida del Robot

El sub-módulo de *salida del robot* convierte los movimientos establecidos por la memoria procedural en instrucciones para el manipulador. El control del movimiento y accionamiento de la pinza para el robot colaborativo UR3 se

realiza mediante una comunicación TCP/IP en una red LAN a la cual debe estar conectado también el equipo de cómputo que corre los algoritmos, estableciendo una comunicación cliente/servidor tipo socket con IP y puerto estáticos.

Este trabajo se apoyó en los desarrollos presentados en [18], [19] y proporciona las librerías *comunicacionUR3.py* y las funciones *Gripper.activate()*, *Gripper.close()*, *Gripper.half()*, *Gripper.open()* para establecer comunicación con el robot UR3 y accionar la pinza. Adicionalmente, contiene la librería *controlUR3*, en la cual se encuentra la función *move()* para el desplazamiento del brazo robótico, la cual utiliza como parámetros las coordenadas (*x*,*y*,*z*) de la posición final en centímetros. Todas estas librerías fueron creadas con base en el lenguaje de programación *URScript*, desarrollado por el fabricante *Universal Robots*.

4) Fisión Multimodal

El sub-módulo de *fisión multimodal* se encarga de repartir las solicitudes provenientes de la memoria de trabajo a los demás sub-módulos (salida de audio, salida de video y salida del robot) correspondientes, conforme llegan. Ya sea movimientos del robot, síntesis de oraciones verbales, o presentar en pantalla la información que entrega la cámara.

IV. PRUEBAS Y RESULTADOS

Se realizaron pruebas con un grupo de siete personas, cuyas edades oscilaban entre los 22 y los 25 años. Todos los participantes contaban con conocimientos previos en computación y experiencia en robótica. Además, todos los usuarios manifestaron tener experiencia en la interacción con interfaces gestuales o verbales, adquirida previamente mediante la interacción con robots, sistemas a control remoto, consolas de videojuegos y asistentes virtuales. Todos los miembros del grupo también contaban con experiencia previa en la manipulación del robot UR3, y conocimiento mínimo sobre interacciones kinestésicas y tele-operadas.

En cuanto a la experiencia con sistemas multimodales, se encontró que cinco de los participantes tenían experiencia en la interacción con sistemas que utilizaban dos o más modalidades simultáneamente.

El desempeño del sistema fue evaluado en dos etapas: una prueba cuantitativa en la que se evaluó el desempeño del reconocimiento durante la interacción de los siete usuarios con la arquitectura. Así, se evaluó el reconocimiento de las 28 expresiones verbales del diccionario, el gesto "seleccionar" ejecutado en cada una de las doce zonas del espacio de trabajo, el comando multimodal, y la solución dada por la arquitectura en el problema de clasificación de objetos cúbicos por color. En la segunda etapa, se realizaron encuestas para conocer la percepción de los usuarios frente al desempeño de la arquitectura propuesta, evaluando las interacciones verbales, gestuales y multimodales, así como la realimentación auditiva y visual.

A. Pruebas cuantitativas

A continuación se presentan los resultados de cada una de las cuatro pruebas cuantitativas:

1) Reconocimiento de Componentes verbales del Diccionario

Cada uno de los siete usuarios pronunció tres veces cada una de las 28 palabras del diccionario; por lo tanto, cada expresión fue evaluada 21 veces. Se obtuvo un desempeño global promedio del 94.56% con una desviación estándar del 7.41, siendo los Usuarios 3 y 5 con quienes el sistema presentó el mayor y el menor desempeño en el reconocimiento con (96.43% ± 10.50%) y (91.67% ± 17.27%) respectivamente, la razón por la cual no se logró alcanzar un porcentaje de acierto del 100% se debe a que la tecnología no ha sido capaz de reconocer consistentemente palabras como "coge", "cubo", "dos", "doce" y "déjalo" en la totalidad de los intentos realizados. Esta limitación se debe a la variabilidad en el ritmo, volumen y pronunciación de los usuarios, lo cual representa un desafío para el sistema de reconocimiento.

La figura 13 resume el porcentaje de reconocimiento de cada una de ellas. Se puede notar que 16 de las 28 expresiones alcanzaron un reconocimiento del 100%. La palabra con menor reconocimiento fue "coge" con un 76.19%, y siendo "toma" la opción mejor reconocida dentro de este tipo. La palabra "cubo" que se usa tanto en interacciones verbales como multimodales alcanzó un 80.95% de reconocimiento siendo confundida con la palabra "culo". Los colores de los cubos fueron reconocidos el 100 % de las veces, así como los índices uno y tres que se asocian al cubo. El índice "dos" se reconoció el 80.95% de las veces, confundiéndolo con la palabra "los". Dentro de las opciones de palabras para la acción "Dejar", la mejor opción es usar la palabra "ponlo" con un 95.24% de acierto; mientras que la opción con reconocimiento más bajo (85.71%) fue usar la palabra "déjalo". Para obtener más detalles sobre los resultados obtenidos, se recomienda consultar la tabla 1 en el anexo A.

Figura 13. Resultados del reconocimiento del diccionario de palabras/componentes.

También se evaluaron tres expresiones que podrían conformarse con los compontes del diccionario, cada una pronunciada tres veces por cada uno de los siete usuarios. Se obtuvo un reconocimiento del 98.41% con una desviación estándar del 4.19% en las 63 ocasiones; de hecho, sólo se presentó una falla en el reconocimiento de la expresión 2 pronunciada por el Usuario 5, identificándose que este usuario presentaba una pronunciación inusual de las letras "S" y "C", además, note que la ubicación en la que se debe dejar el cubo (doce) obtuvo un reconocimiento del 85.71 % individualmente. La figura 14 presenta gráficamente el porcentaje de reconocimiento alcanzado por las tres expresiones evaluadas. Para un análisis más detallado de los resultados, se puede consultar la tabla 2 en el Anexo A.

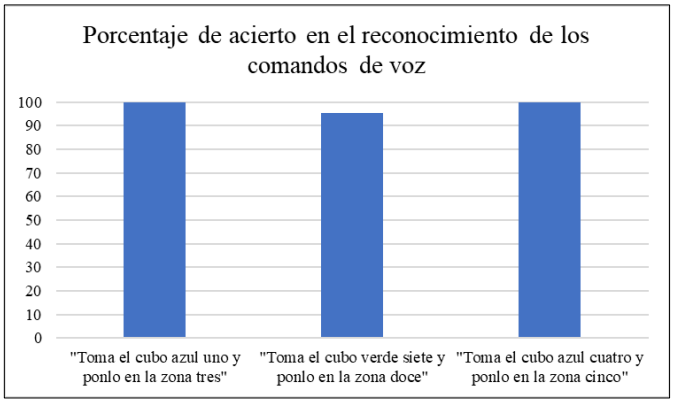


Figura 14. Resultados del reconocimiento de comandos de voz.

2) Interacción gestual

Se solicitó a cada uno de los siete usuarios ejecutar el gesto "seleccionar" tres veces, sobre cada una de las 12 zonas definidas en el espacio de trabajo. De este modo, en las 63 repeticiones se obtuvo un reconocimiento promedio del 81.35% con una desviación estándar del 7.98%. Los menores desempeños del sistema ocurrieron al reconocer los gestos realizados por los Usuarios 1 y 7 con (72.22% ± 19.24%) y (69.45% ± 22.28%) respectivamente. En cuanto a las zonas, la Figura 15 presenta los porcentajes promedio en el reconocimiento del gesto "seleccionar" en cada zona. Las zonas 6, 7, 10, 11 y 12 fueron las mejores con (85.72% ± 17.81%), (90.48 ± 16.26%), (90.48% ± 16.26%), (95.24% ± 12.60%) y (85.72 ± 17.81%), y sus ubicaciones se encuentran en la parte inferior central de la imagen de la cámara. El porcentaje de menor de reconocimiento ocurrió en las zonas 5 y 8 con (71.43% ± 23.00%) y (71.43% ± 12.60%), respectivamente, que se ubican en los bordes laterales centrales de la zona de trabajo. Para obtener más detalles sobre los resultados obtenidos, se puede consultar la tabla 3 en el anexo A.

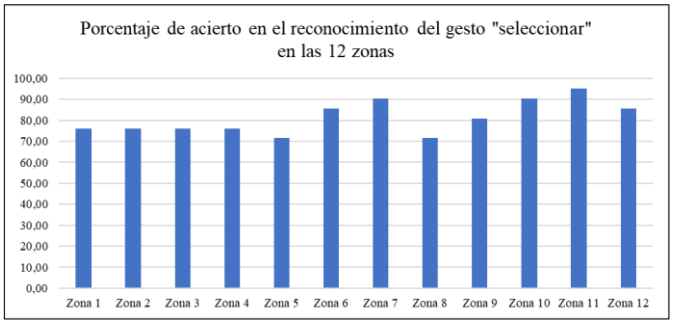


Figura 15. Resultados de interacciones gestuales.

Se observó que al ejecutar el gesto de "seleccionar", los usuarios no cerraban los dedos de manera uniforme. En su lugar, se encontró que en la mayoría de los casos acercaban el dedo índice hacia el pulgar. Esto ocasionalmente resultaba en un desplazamiento del cursor, lo que conllevaba a la identificación errónea de un cubo o una zona diferente a la que se pretendía señalar.

3) Interacción multimodal

Cada uno de los 7 usuarios repitieron 3 veces la interacción multimodal que implica decir "Mira este cubo, ponlo aquí", y al mismo tiempo, realizar el gesto "seleccionar" durante la pronunciación de la parte "mira este cubo", y posteriormente mover su mano hasta la zona a la que se desea llevar el cubo. Se eligieron 3 zonas como ubicación final del cubo, la 1, 5 y 9, que no fueron las de mejor reconocimiento en la ejecución del gesto "seleccionar" en la interacción gestual. Se logró un reconocimiento del 80.95% con una desviación estándar del 12.36% en las 63 interacciones. De las 63 interacciones, se observa que el menor desempeño del sistema se obtuvo al reconocer las interacciones realizadas por los Usuarios 1, 4 y 7 con (77,78% ± 19.24%) para los dos primeros, y (55,55% ± 19.24%) para el último. En la figura 16 se muestran los resultados promedio obtenidos en el reconocimiento de las interacciones multimodales. Para más detalles sobre los resultados obtenidos, se recomienda consultar la tabla 3 en el anexo A.

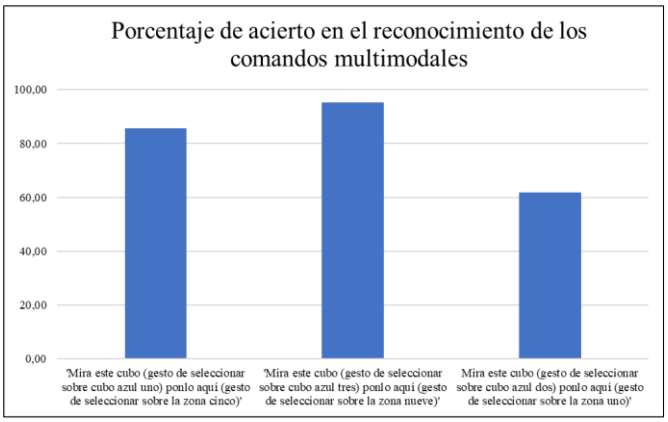


Figura 16. Resultados de interacciones multimodales.

Los resultados globales de esta prueba son muy similares a los presentados en las pruebas de la interacción gestual, lo que puede estar relacionado con el hecho de que la interacción gestual tiene mayor fuerza sobre los resultados de acierto en el reconocimiento del comando, dado que esta es la que permite definir el cubo y la zona, además, para que el comando se cumpla correctamente ambas interacciones (verbal y gestual) deben ser bien reconocidas por la arquitectura, pero como se observó en los resultados de la interacción verbal, el porcentaje de acierto es bastante alto (casi seguro), dejando casi toda la responsabilidad en el reconocimiento a la interacción gestual.

4) Resultados de la Clasificación de los Objetos

Para evaluar el desempeño de la arquitectura se realizó una prueba con cada uno de los siete usuarios, en la que se tienen tres pasos: en primer lugar se restableció la memoria episódica cada vez que pasaba un usuario a realizar el ejercicio, para eliminar cualquier recuerdo anterior; el segundo paso consistió en pasar al estado de aprendizaje y enseñarle una tarea por cada una de las posibles combinaciones que pueden tener los cubos en el espacio de trabajo, por ejemplo, un cubo azul y un cubo verde, un cubo azul y dos cubos verdes, un cubo azul y tres cubos verdes, hasta llegar al escenario en el que estén presentes en el espacio de trabajo los tres cubos azules y los tres cubos

verdes. Finalmente, en el último paso, el usuario tendría que indicarle al robot que debía ejecutar una tarea con cada una de las posibles combinaciones de cantidad y color de los cubos que se enseñó, pero con la posibilidad de variar las posiciones iniciales de dichos cubos. Con esto se validó que la arquitectura posicionase cada uno de los cubos exactamente en las zonas indicadas por el usuario en el estado de aprendizaje.

Los resultados de esta prueba con un total de 63 interacciones en las que cada usuario ejecutó el ejercicio una sola vez por cada una de las nueve posibles combinaciones, la arquitectura ejecutó correctamente el posicionamiento de todos los cubos de acuerdo a los recuerdos que almacenó en el 100% de los casos.

B. Pruebas cualitativas

Posterior a los tres tipos de interacciones que ejecutaron los siete usuarios, se les pidió diligenciar una encuesta de 5 afirmaciones, a través de las cuales indicaban su nivel de acuerdo o desacuerdo con el desempeño de la arquitectura, usando para ello la escala *Likert de cinco niveles* [12]. Las afirmaciones se detallan en la Tabla 4.

	Afirmación
1	Según su criterio, el sistema multimodal desarrollado en la arquitectura mejora la experiencia de usuario comparado con una interacción kinestésica o tele operada.
2	Según su percepción, los comandos de voz fueron identificados correctamente por la arquitectura.
3	En su opinión, la arquitectura identificó correctamente los comandos gestuales.
4	Considera que la arquitectura identificó correctamente los comandos multimodales.
5	Desde su punto de vista, considera que la realimentación gráfica/auditiva contribuyó a tener una mejor experiencia.

Tabla 4. Cuestionario usado en la prueba cualitativa de la arquitectura.

Los resultados obtenidos se presentan gráficamente en la Figura 17, donde sobresale la satisfacción de los usuarios con las interacciones verbales, pues le asignaron la respuesta máxima. La segunda afirmación mejor evaluada fue la 5, sobre el aporte en la experiencia de las realimentaciones auditivas y visuales. Las afirmaciones 4 y 5 tuvieron un buen desempeño, siendo similar que ambas afirmaciones indagan sobre el reconocimiento en las interacciones gestuales y multimodales. La afirmación que tuvo más disenso fue en la 1 sobre el aporte de la arquitectura en la interacción con el manipulador, de hecho, aunque tres usuarios estuvieron totalmente de acuerdo en que el sistema multimodal mejora la experiencia de usuario, los demás usuarios manifestaron que es necesario mejorar la identificación de gestos para evitar errores y tareas mal enseñadas.

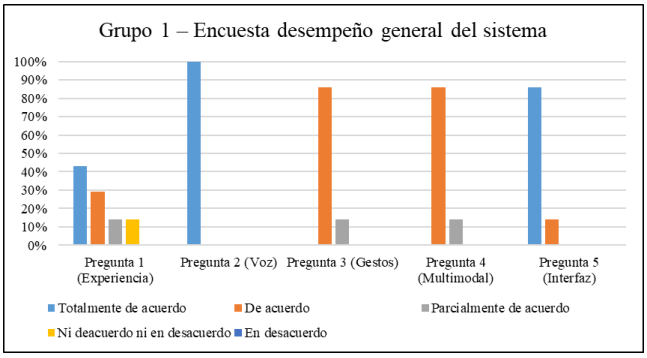


Figura 17. Encuesta del desempeño cualitativo de la arquitectura.

Algunos usuarios destacaron la necesidad de mayor precisión en la identificación de las zonas o cubos utilizando el gesto "seleccionar". Además, algunos usuarios sugirieron que el uso de una interfaz común podría ser una opción más efectiva.

Para acceder al código de programación utilizado en este trabajo, se puede encontrar una copia en el siguiente enlace de Google Drive: https://drive.google.com/drive/folders/1-nznC0VvSoU-y-Y0TupVwaXjjUKSpzHr?usp=sharing. En este repositorio se encuentra el código en su totalidad, junto con los archivos necesarios para reproducir los resultados presentados en este artículo.

V. DISCUSIÓN

Se puede afirmar que el sistema implementado cumple con el objetivo propuesto, ya que permite a los usuarios enseñar, mediante comandos multimodales, una tarea de clasificación de objetos cúbicos, almacenando estas demostraciones en la memoria episódica en la etapa de aprendizaje y posteriormente ejecutándolas por la memoria procedural en la etapa de ejecución.

El sistema propuesto tiene como principales características: (i) capacidad de interpretar comandos gestuales y verbales simultáneamente; (ii) aproximación de una arquitectura cognitiva SOAR para el aprendizaje y ejecución de tareas de clasificación; (iii) experimentación en un ambiente real utilizando el robot UR3; (iv) realimentación gráfica y auditiva para mejorar la experiencia de usuario.

Los resultados de las pruebas cuantitativas indican que el sistema tiene un alto desempeño en las interacciones verbales, con una tasa de acierto promedio del 98.41% y una baja desviación estándar del 4.19%. Sin embargo, en las interacciones multimodales, la tasa de acierto promedio fue del 80.95%, con una desviación estándar del 12.36%, lo que sugiere que aún hay margen de mejora en este tipo de interacción.

El análisis realizado reveló que la distorsión de la lente y la inclinación de la cámara son dos factores importantes que contribuyen a la introducción de errores en las interacciones multimodales. La inclinación de la cámara puede causar una deformación en la perspectiva del espacio de trabajo, lo que resulta en errores al identificar las coordenadas de los cubos. Además, la distorsión de la lente también puede afectar la calidad y precisión de la imagen capturada, lo que contribuye a la introducción de errores.

Al abordar estos problemas, se espera que el sistema alcance un mayor nivel de precisión y mejore su desempeño en las interacciones multimodales.

En las pruebas cualitativas, los usuarios mostraron su satisfacción en cuanto a la capacidad del sistema para identificar correctamente los comandos de voz, sin embargo, los comandos gestuales y multimodales presentan oportunidades de mejora para identificar correctamente las tareas enseñadas por los usuarios. Los resultados sugieren que los usuarios se sienten más cómodos con los comandos verbales, aunque están dispuestos a utilizar interacciones multimodales si se logra mejorar la capacidad del sistema para identificarlas.

Finalmente, los usuarios de prueba están en general de acuerdo en que la implementación de comandos verbales y multimodales mejora la experiencia de interacción con el robot. Esto puede deberse a la experiencia previa que manifestaron tener con interfaces kinestésicas o tele-operadas, las cuales requerían conocimientos previos y experiencia en programación. De esta manera, se puede concluir que la arquitectura propuesta logra mejorar la experiencia de usuario en la interacción con el robot.

VI. CONCLUSIONES

En este trabajo, se desarrolló y validó el desempeño de una arquitectura que utiliza interacción multimodal comandada por gestos y voz, para permitir a los usuarios enseñar y ejecutar tareas de clasificación de objetos cúbicos por color al robot colaborativo UR3. El enfoque utilizado fue una aproximación a la arquitectura cognitiva SOAR.

Se ha encontrado que la complejidad en el aprendizaje no aumenta con la disminución del tamaño de las zonas o la eliminación de estas para utilizar únicamente coordenadas. En cambio, la complejidad se traslada a los módulos encargados de la memoria procedural y el desplazamiento de los cubos para evitar colisiones. Este hallazgo sugiere que la arquitectura propuesta, en el estado de ejecución, puede superar estos desafíos al enfocarse en las posiciones finales de los cubos en lugar de sus posiciones iniciales, y al validar y retirar los cubos que bloquean la zona objetivo. Este hallazgo tiene implicaciones importantes para futuros sistemas de inteligencia artificial diseñados para tareas similares.

Como trabajos futuros se propone la adición de un nuevo gesto para contrastar e identificar si existe otro gesto que permita aumentar el porcentaje de acierto en la identificación, la exploración de otros métodos para el despeje de las zonas objetivo cuando se encuentran obstruidas por otros cubos y la aplicación de técnicas que permitan la selección de recuerdos adecuados de acuerdo a alguna condición que pueda definir el usuario, así como implementar una técnica de control de la pinza que le permita ser más precisa en la toma de los cubos y evaluar que tanto se podría disminuir el tamaño de las zonas sin tener colisiones. También se debe contemplar la adición del giro de la pinza para evitar obstrucciones con cubos que puedan estar muy cercanos al objetivo. Para el sistema de visión se propone implementar una técnica de calibración de cámara que permita ajustar las coordenadas de los cubos en tiempo real, para compensar la deformación causada por la inclinación de la cámara.

VII. AGRADECIMIENTOS

Este trabajo fue financiado con recursos del proyecto de investigación interna *Aproximación a una arquitectura cognitiva para el aprendizaje y generalización de un proceso de clasificación aplicado en un robot colaborativo* el cual pertenece a la Convocatoria interna de proyectos *Por una universidad transformadora: Horizonte 2021-2025*, de la Pontificia Universidad Javeriana Cali.

VIII. REFERENCIAS

[1] D. Tabuenca, “Implantación de robots colaborativos en línea de producción.” pp. 17–24, 2017. [Online]. Available: https://uvadoc.uva.es/bitstream/handle/10324/23076/TFG-I-584.pdf?sequence=1&isAllowed=y

[2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 2016.

[3] R. E. Mayer, *Multimedia Learning (Second Edition)*, 2nd ed. Cambridge University Press, 2009.

[4] R. E. Mayer, “Thirty years of research on online learning,” *Appl Cogn Psychol*, vol. 33, no. 2, pp. 152–159, Mar. 2019, doi: 10.1002/acp.3482.

[5] J. Sweller, J. J. G. van Merriënboer, and F. Paas, “Cognitive Architecture and Instructional Design: 20 Years Later,” *Educ Psychol Rev*, vol. 31, no. 2, pp. 261–292, Jun. 2019, doi: 10.1007/s10648-019-09465-5.

[6] D. Choi and P. Langley, “Evolution of the Icarus Cognitive Architecture,” *Cogn Syst Res*, vol. 48, pp. 25–38, May 2018, doi: 10.1016/j.cogsys.2017.05.005.

[7] J. Laird, *The Soar Cognitive Architecture*. The MIT Press, 2012.

[8] J. H. Mosquera, H. Loaiza, S. E. Nope, and A. D. Restrepo, “Disability and Rehabilitation : Assistive Technology Human-computer multimodal interface to internet navigation Human-computer multimodal interface to internet navigation,” *Disabil Rehabil Assist Technol*, vol. 0, 2020, doi: https://doi.org/10.1080/17483107.2020.179944.

[9] J. Laird, C. Lebiere, and P. S. Rosenbloom, “A Standard Model of the Mind: Toward a Common Computational Framework Across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics,” *AI magazine*, pp. 13–26, 2017.

[10] O. Janrathitikarn and L. N. Long, “Gait Control of a Six-Legged Robot on Unlevel Terrain Using a Cognitive Architecture,” in *2008 IEEE Aerospace Conference*, IEEE, Mar. 2008, pp. 1–9. doi: 10.1109/AERO.2008.4526240.

[11] A. D. Dubey and R. B. Mishra, “Cognition of a Robotic Manipulator Using the Q-Learning Based Situation-Operator Model,” *Journal of Information Technology Research*, vol. 11, no. 1, pp. 146–157, Jan. 2018, doi: 10.4018/JITR.2018010109.

[12] A. Bono, A. Augello, G. Pilato, F. Vella, and S. Gaglio, "An ACT-R Based Humanoid Social Robot to Manage Storytelling Activities," *Robotics*, vol. 9, no. 2, p. 25, Apr. 2020, doi: 10.3390/robotics9020025.

[13] N. M. DiFilippo and M. K. Jouaneh, "Using the Soar Cognitive Architecture to Remove Screws From Different Laptop Models," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 767–780, Apr. 2019, doi: 10.1109/TASE.2018.2860945.

[14] M. A. Tamayo-Monsalve, N. L. Montes-Castrillón, and G. A. Osorio-Londoño, "REAL-TIME CLASSIFICATION OF OBJECTS BY COLOR USING FIELD PROGRAMMABLE GATE ARRAY," *Revista de Investigaciones Universidad del Quindío*, vol. 23, no. 1, pp. 33–39, Aug. 2012, doi: 10.33975/riuq.vol23n1.416.

[15] S. Mcleod, "Likert Scale," 2008. www.simplypsychology.org/likert-scale.html (accessed Apr. 30, 2023).

[16] Pypi.org, "PyAudio 0.2.13," 2022. https://pypi.org/project/PyAudio/ (accessed Jan. 17, 2023).

[17] Pypi.org, "Python Speech Recognition 3.9.0," 2022. https://pypi.org/project/SpeechRecognition/ (accessed Mar. 28, 2023).

[18] Google LLC, "Language model selection for speech-to-text conversion," 2023. https://patents.google.com/patent/US9495127B2/en (accessed Mar. 28, 2023).

[19] OpenCV Development Team, "Color conversions." https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html#color_convert_rgb_hsv (accessed May 07, 2023).

[20] Ponce Jean and Forsyth David A., *Computer Vision: A Modern Approach*, 2nd ed. Pearson, 2011.

[21] Pypi.org, "OpenCV-Python 4.7.0.68," 2022. https://pypi.org/project/pyttsx3/ (accessed Jan. 17, 2023).

IX. ANEXOS

Anexo A. Tablas de resultados

En esta sección se presentan las tablas con los resultados de las pruebas realizadas, que complementan la información presentada en el artículo.

La tabla A1 presenta los resultados de las pruebas llevadas a cabo para evaluar el rendimiento de la arquitectura en el reconocimiento de expresiones verbales. Las expresiones verbales se encuentran representadas en la primera columna, mientras que los encabezados U1 a U7 corresponden a los usuarios que participaron en las pruebas.

Tabla A1 – Resultados del desempeño de reconocimiento de las componentes verbales.

Porcentaje de desempeño de las expresiones verbales								
Expresión	**U1**	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**Total**
"hola"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"aprender"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"ejecutar"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"zona"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"uno"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"dos"	66,7	100,0	66,7	100,0	66,7	100,0	66,7	**80,95**
"tres"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"cuatro"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"cinco"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"seis"	66,7	66,7	100,0	100,0	100,0	100,0	100,0	**90,48**
"siete"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"ocho"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"nueve"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"diez"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"once"	100,0	66,7	66,7	100,0	100,0	100,0	100,0	**90,48**
"doce"	66,7	100,0	66,7	100,0	100,0	66,7	100,0	**85,71**
"cubo"	100,0	100,0	100,0	66,7	66,7	66,7	66,7	**80,95**
"toma"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"sujeta"	100,0	100,0	100,0	100,0	66,7	66,7	66,7	**85,71**
"coge"	100,0	100,0	100,0	66,7	33,3	66,7	66,7	**76,19**
"agarra"	100,0	66,7	100,0	100,0	100,0	100,0	100,0	**95,24**
"aquí"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"ponlo"	100,0	100,0	100,0	66,7	100,0	100,0	100,0	**95,24**
"colócalo"	66,7	100,0	100,0	100,0	66,7	100,0	100,0	**90,48**
"déjalo"	100,0	100,0	100,0	66,7	66,7	66,7	100,0	**85,71**
"muévelo"	100,0	66,7	100,0	100,0	100,0	100,0	66,7	**90,48**
"azul"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"verde"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
Total	**95,24**	**95,24**	**96,43**	**95,24**	**91,67**	**94,05**	**94,05**	**94,56**

La tabla A2 muestra los resultados obtenidos de las pruebas realizadas para evaluar el desempeño de la arquitectura en el reconocimiento de los comandos de voz. Los comandos evaluados se encuentran representados en los encabezados de la segunda fila, mientras que los resultados de los usuarios se presentan en las filas U1 a U7. Cada celda de la tabla muestra el resultado obtenido para cada usuario en cada uno de los comandos evaluados.

Tabla A2 – Resultados del desempeño de reconocimiento de los comandos de voz.

USR	**Comando de Voz**			**Total**
	"Toma el cubo azul uno y ponlo en la zona tres"	**"Toma el cubo verde siete y ponlo en la zona doce"**	**"Toma el cubo azul cuatro y ponlo en la zona cinco"**	
U1	100	100	100	**100**
U2	100	100	100	**100**
U3	100	100	100	**100**
U4	100	100	100	**100**
U5	100	66,67	100	**88,89**
U6	100	100	100	**100**
U7	100	100	100	**100**
Total	**100**	**95,24**	**100**	**98,41**

En la tabla A3 se presentan los resultados del desempeño de la arquitectura en la elección de las zonas mediante el gesto "seleccionar". Las zonas evaluadas se presentan en las filas de la tabla, mientras que los encabezados de las columnas indican a cada uno de los usuarios participantes en las pruebas (U1, U2, U3, etc.). En cada celda de la tabla se puede observar el resultado obtenido para cada usuario en cada una de las zonas evaluadas.

Tabla A3 – Resultados del desempeño de reconocimiento de los gestos sobre las zonas.

Porcentaje de desempeño GESTO								
ZONA	**U1**	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**Total**
1	66,67	100	100	100	66,67	66,67	33,33	**76,19**
2	66,67	100	100	66,67	66,67	66,67	66,67	**76,19**
3	66,67	66,67	100	100	66,67	66,67	66,67	**76,19**
4	33,33	66,67	66,67	66,67	100	100	100	**76,19**
5	66,67	66,67	66,67	100	100	66,67	33,33	**71,43**
6	66,67	100	100	66,67	100	66,67	100	**85,72**
7	100	100	100	100	66,67	100	66,67	**90,48**
8	66,67	100	66,67	66,67	66,67	66,67	66,67	**71,43**
9	66,67	100	66,67	66,67	100	100	66,67	**80,95**
10	100	100	100	100	66,67	100	66,67	**90,48**
11	66,67	100	100	100	100	100	100	**95,24**
12	100	100	66,67	66,67	100	100	66,67	**85,72**
Total	**72,22**	**91,67**	**86,11**	**83,34**	**83,34**	**83,34**	**69,45**	**81,35**

La tabla A4 muestra los resultados de la evaluación del desempeño de la arquitectura en el reconocimiento de los comandos multimodales. En los encabezados de la segunda fila se presentan los comandos evaluados, mientras que en las demás filas se presentan los usuarios participantes en la prueba (U1, U2, U3, etc.). Cada celda de la tabla contiene el resultado obtenido para cada usuario en cada uno de los comandos evaluados.

Tabla A4 – Resultados del desempeño de reconocimiento de los comandos multimodales.

USR	Comando multimodal			Total
	'Mira este cubo (gesto de seleccionar sobre cubo azul uno) ponlo aquí (gesto de seleccionar sobre la zona cinco)'	'Mira este cubo (gesto de seleccionar sobre cubo azul tres) ponlo aquí (gesto de seleccionar sobre la zona nueve)'	Mira este cubo (gesto de seleccionar sobre cubo azul dos) ponlo aquí (gesto de seleccionar sobre la zona uno)'	
U1	66,67	100,00	66,67	**77,78**
U2	100,00	100,00	66,67	**88,89**
U3	100,00	100,00	66,67	**88,89**
U4	66,67	100,00	66,67	**77,78**
U5	100,00	100,00	66,67	**88,89**
U6	100,00	100,00	66,67	**88,89**
U7	66,67	66,67	33,33	**55,55**
Total	**85,71**	**95,24**	**61,90**	**80,95**