

# DISEÑO DE UN ROBOT CARTESIANO PARA LA MEDICIÓN DE TEMPERATURA CORPORAL Y DETECCIÓN DE MASCARILLA FACIAL MEDIANTE SENSORES INFRARROJOS Y VISIÓN ARTIFICIAL PARA PREVENIR EL CONTAGIO DEL COVID -19

## *DESIGN OF A CARTESIAN ROBOT FOR BODY TEMPERATURE MEASUREMENT AND FACIAL MASK DETECTION USING INFRARED SENSORS AND ARTIFICIAL VISION TO PREVENT THE SPREAD OF COVID-19*

Sergio A. Gutiérrez Sánchez<sup>1</sup>, Mario R. Hernández Valladares<sup>2</sup>, Diego A. Magallanes Chauca<sup>3</sup>

Dr. Ing. Ricardo J. Palomares Orihuela<sup>4</sup>

### RESUMEN

Este proyecto tuvo por finalidad diseñar un robot cartesiano de un eje para la medición de la temperatura corporal y detección de mascarilla facial como parte del protocolo de control de acceso de personas en los centros comerciales de Lima, Perú. El robot cartesiano fue diseñado utilizando el software SolidWork y simulado mediante CoppeliaSim, mide la temperatura corporal mediante un sensor infrarrojo y detecta la presencia de mascarilla mediante una cámara GOPRO Hero 7 Black y una red neuronal convolucional. Mediante la prueba 1 se comprobó el funcionamiento del sistema de detección de mascarillas, al obtener la etiqueta "Con\_Mascarilla" cuando la persona tiene puesta una. En la prueba 2 se simuló el desplazamiento del sensor infrarrojo y la medición de la temperatura corporal, obteniendo una luz verde para una medición en el rango normal de 36° a 37.5°. En la prueba 3, se simuló el funcionamiento del robot al ingreso del centro comercial. La etapa final del proyecto, incluye su implementación y validación mediante un sistema de alarma mediante voz.

**Palabras claves:** Robot cartesiano, visión artificial, red neuronal, Covid-19

### ABSTRACT

The objective of this project was to design a one-axis Cartesian robot for body temperature measurement and face mask detection as part of the access control protocol for people in shopping centers in Lima, Peru. The Cartesian robot was designed using SolidWork software and simulated using CoppeliaSim, it measures body temperature using an infrared sensor and detects the presence of a mask using a GOPRO Hero 7 Black camera and a convolutional neural network. Through test 1, the operation of the mask detection system was verified, obtaining the label "With\_Mascarilla" when the person is wearing one. In test 2, the displacement of the infrared sensor and the body temperature measurement were simulated, obtaining a green light for a measurement in the normal range of 36 ° to 37.5 °. In test 3, the operation of the robot at the entrance of the shopping center was simulated. The final stage of the project includes its implementation and validation through a voice alarm system.

**Keywords:** Cartesian robot, computer vision, neural network, Covid-19

---

<sup>1</sup> Estudiante del décimo ciclo de Ingeniería Mecatrónica de la Universidad Ricardo Palma. Conocimientos y certificaciones en temas relacionados a robótica, inteligencia artificial y ciencia de datos. Practicante de Inteligencia Artificial en Lucky. Past President del Capítulo Estudiantil Robotics and Automation Society RAS-IEEE 2019 de la URP. Email: sergio.gutierrez@urp.edu.pe

<sup>2</sup> Estudiante del décimo ciclo de Ingeniería Mecatrónica de la Universidad Ricardo Palma. Conocimientos en robótica e inteligencia artificial. Expositor en Student Poster Competition en la Conferencia Virtual LACCEI 2021. Email: mario.hernandez@urp.edu.pe

<sup>3</sup> Estudiante del décimo ciclo de Ingeniería Mecatrónica de la Universidad Ricardo Palma. Conocimientos en robótica, diseño gráfico y CAD. Email: diego.magallanes@urp.edu.pe

<sup>4</sup> Ingeniero Mecatrónico de la Universidad Nacional de Ingeniería, Doctor en Ciencias de la Educación. Magister en Ciencias de la Educación con mención en Docencia Universitaria, Maestría en Ciencias de la Electrónica con mención en Ingeniería Biomédica. Docente de la Escuela Profesional de Ingeniería Mecatrónica de la Universidad Ricardo Palma. Senior Member IEEE. Miembro profesional RAS, EMBS, ComSoc y EduSoc - IEEE. Investigador RENACYT – Grupo Carlos Monge IV. Email: rpalomares@ieee.org

## 1. INTRODUCCIÓN

Debido a la pandemia por el virus Covid-19, las empresas se han visto obligadas a contratar personal adicional para el control en los ingresos de las personas en los establecimientos comerciales con el objetivo de medir la temperatura corporal y la verificación de la mascarilla facial. La labor del personal de control genera un gasto adicional mensual en la empresa y por otro lado, y pone en riesgo su salud al interactuar constantemente con otras personas, ocasionando un incremento indirecto en la cantidad de contagiados por Covid-19, hecho que, se vuelve crítico en épocas festivas, como el incremento de casos confirmados a finales del año 2020 e inicios del año 2021 [1], relacionados a la festividad de la Navidad y el Año Nuevo, lo que significó el inicio de la segunda ola con un incremento de 10,398 detecciones en la semana 51 del año 2020 a 29,744 detecciones en la semana 2 del año 2021.

Por ello, se buscó como diseñar un robot cartesiano [2], [3], de 1 GDL que usa un sensor ultrasónico para obtener la altura de la persona que desea ingresar al centro comercial y en consecuencia controla el desplazamiento de un sensor infrarrojo hacia la muñeca de las personas para su posterior medición de la temperatura corporal. Asimismo, determina la existencia de la mascarilla facial mediante el análisis fotográfico de las personas mediante una cámara y una red neuronal convolucional. De esta manera, en cuanto la temperatura corporal se encuentre en el rango permitido y la persona tenga colocada la mascarilla facial, el sistema autoriza su ingreso al centro comercial, abriendo la puerta de ingreso de manera automática; en caso contrario, se notifica a la persona que podría tener un problema de salud. Por otro lado, si la persona no colabora con las indicaciones del robot cartesiano, se procede a avisar por medio de una alarma al personal de seguridad del centro comercial, con la finalidad de evitar riesgos de contagio. Según el MINSa, en el mes de agosto, el Perú se encuentra saliendo de la segunda ola y si no se fortalece los mecanismos de prevención de contagios se estaría ingresando en la tercera ola, tal como se muestra en la Figura 1.

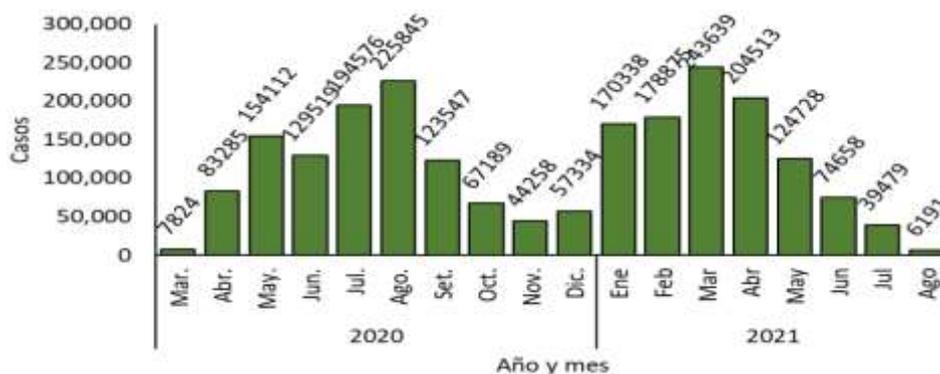


Figura 1. Casos confirmados de Covid-19, según fecha de inicio de síntomas

Fuente: Sala Situacional Covid-19 Perú del MINSa

## 2. DISEÑO DEL SISTEMA DEL ROBOT CARTESIANO

### 2.1 Diseño Mecánico del Robot Cartesiano

El robot realiza un desplazamiento en el eje Z, como se observa en la figura 2, con la finalidad de mover el módulo de medición en función al 80% de la altura de la persona medida, lo cual es controlado por una Raspberry Pi, con un sensor de ultrasonido [4] ubicado en el techo de la entrada del centro comercial. Después se realiza el desplazamiento, utilizando el módulo de medición, el cual toma la temperatura corporal mediante un sensor infrarrojo [5] y mediante una cámara GOPRO Hero 7 Black se detecta si el cliente porta o no una mascarilla facial; si cumple los requisitos se procede a abrir automáticamente las puertas del centro comercial.

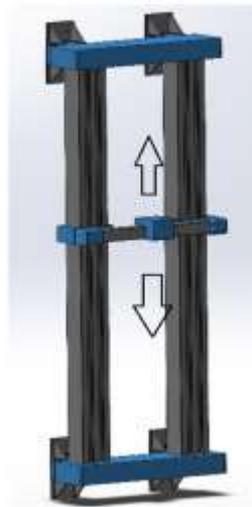


Figura 2. Estructura mecánica del robot cartesiano

Fuente: elaborado en SolidWork 2020 SP3.0

Con la señal que el sensor de proximidad mida, se activa los motores paso a paso, instalados como se aprecia en la Figura 3, conectados a los tornillos sin fin mediante un acoplador, y giran hasta que el sensor infrarrojo esté a la altura requerida.

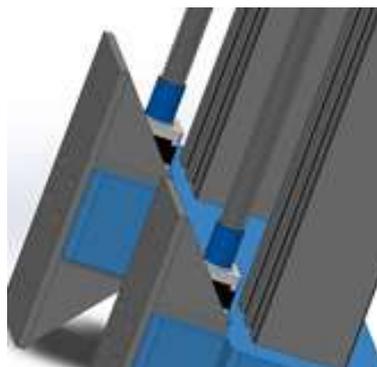


Fig. 3. Motores paso a paso activados por sensor de proximidad

Fuente: elaborado en SolidWork 2020 SP3.0

El robot diseñado en SolidWorks, cuenta con una altura aproximada de 2 metros, con la finalidad de ser utilizado con personas con estatura por encima del promedio. Las medidas del robot cartesiano se aprecian en la figura 4.

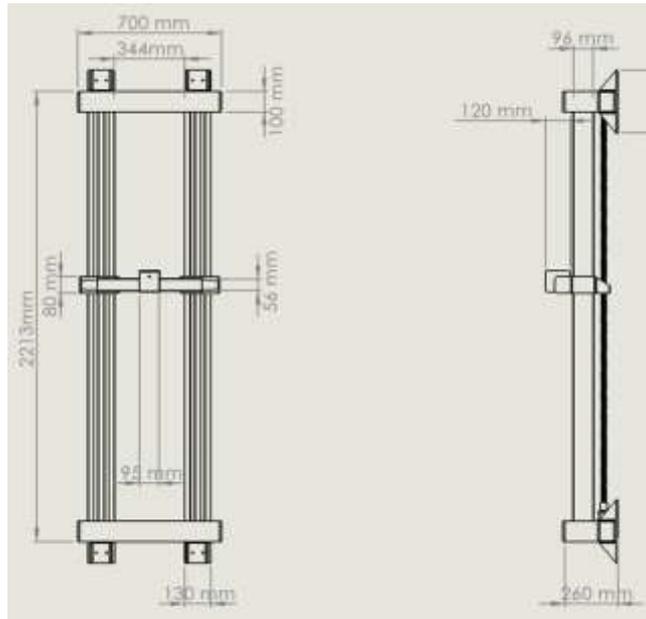


Fig. 4. Dimensionado del Robot Cartesiano  
Fuente: elaborado en SolidWork 2020 SP3.0

### Análisis Cinemático y Dinámico del robot cartesiano

Se utilizó el algoritmo de Denavit-Hartenberg para calcular la cinemática directa del robot, tal como se observa en la figura 5.

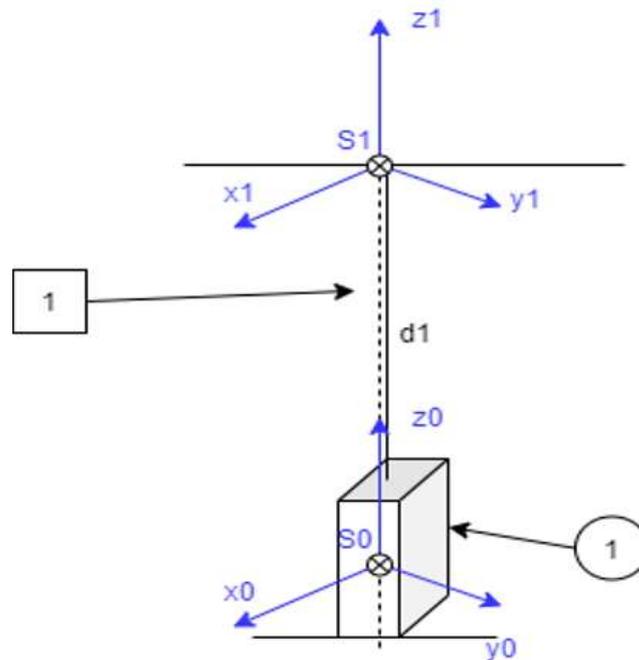


Fig. 5. Esquema del robot cartesiano

En la Tabla 1, se presentan los parámetros cinemáticos del robot cartesiano.

**Tabla 1** Parámetros Cinemáticos del robot

Articulación	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	0	$d_1$	0	0

Donde:

$d_1$  = desplazamiento variable del robot cartesiano

Luego se realiza el cálculo de la dinámica utilizando el método de Lagrange-Euler y se obtiene la ecuación 1:

$$F_1 = m_1 * (\ddot{d}_1 + 9.8) \dots \dots \dots (1)$$

Donde:

$F_1$  = fuerza ejercida por el motor

$m_1$  = masa del módulo electrónico

$\ddot{d}_1$  = segunda derivada del desplazamiento variable del robot cartesiano

## 2.2 Diseño Electrónico del Robot Cartesiano

Para el diseño electrónico, se describe el algoritmo que se implementó para la programación del sistema. El ciclo de funcionamiento del robot cartesiano inicia detectando la altura de la persona parada frente al robot para luego posicionarse al 80% de su altura (de abajo hacia arriba). Después, mediante un algoritmo de Deep Learning, utilizando una red neuronal convolucional, se identifica si la persona porta o no mascarilla facial, prohibiendo el ingreso en caso de ser negativo y continúa con la medición de la temperatura corporal en caso sea positivo. Si la temperatura corporal de la persona se encuentra en el rango de 36° a 37.5° C se apertura automáticamente la puerta de ingreso y se permite su acceso, de ser negativo, se muestra una señal indicando que no se le permitirá el ingreso, como se indica en el diagrama de flujo de la figura 6.

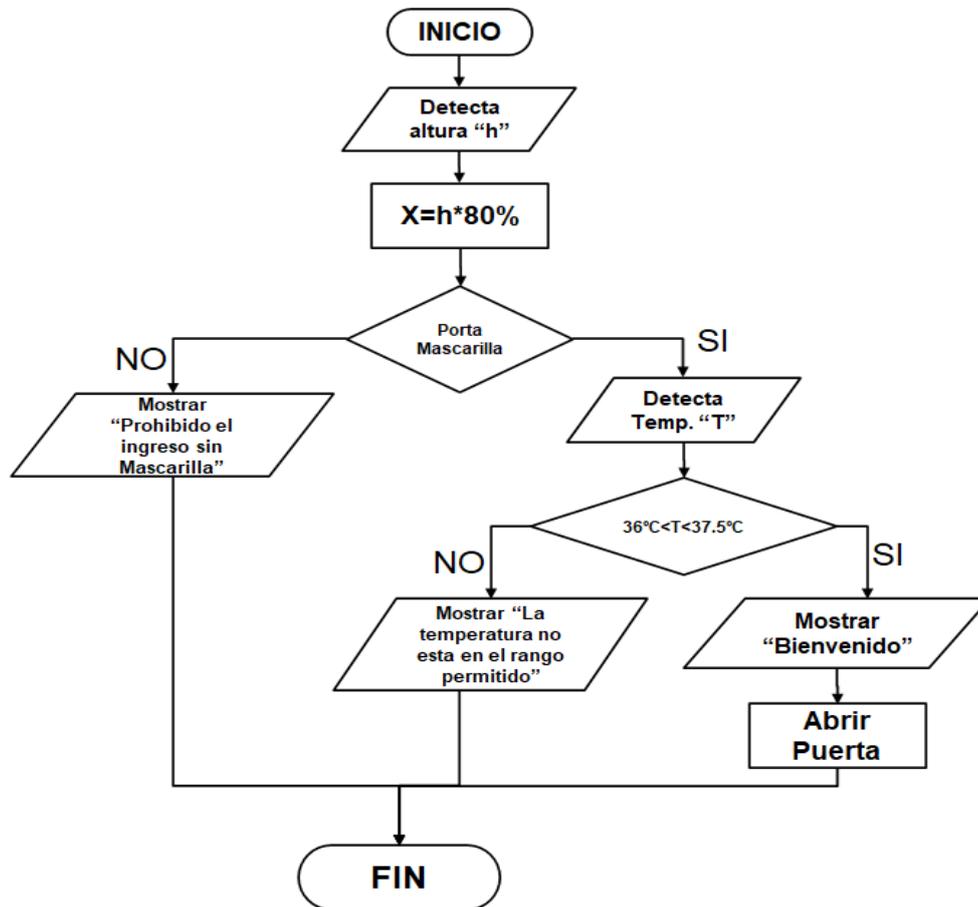


Fig. 6. Diagrama de Flujo del Funcionamiento del Robot Cartesiano

Para la simulación del circuito electrónico se utilizó Tinkercad, una plataforma de diseño para modelado 3D y circuitos electrónicos perteneciente a Autodesk. Se ejecuta en la web y es reconocido por su simplicidad y facilidad de uso.

Como se observa en la figura 7, el circuito se simuló en la web para que se implemente en el robot cartesiano.

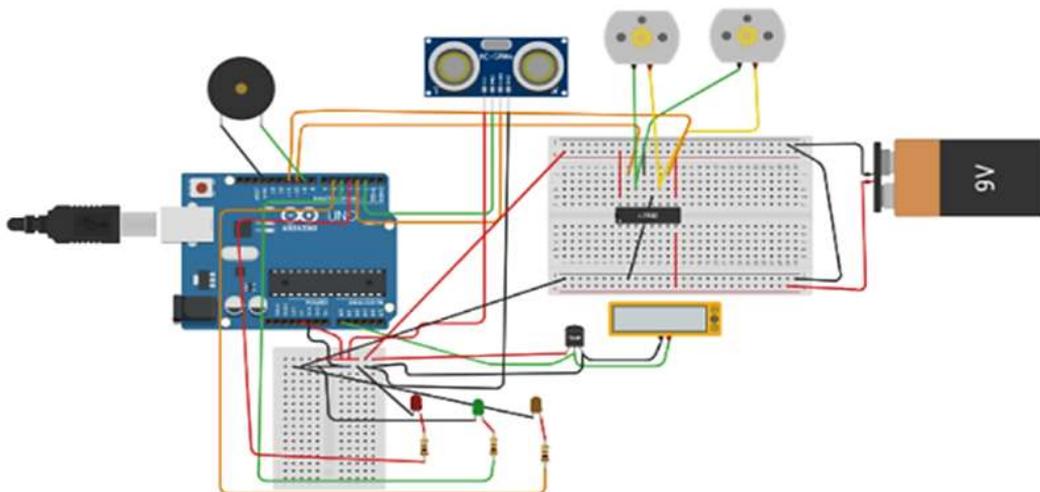


Fig. 7. Partes del ensamblado electrónico

Fuente: elaborado en Tinkercad

En la figura 8, se observa el fragmento del código que permite determinar si hay presencia de alguien. En la línea 52 se calcula el tiempo en el cambio de estado mediante la función “*pulseIn ( )*”. En la línea 57 se determina la altura de la persona y se guarda en la variable “*distancia*”. Si la altura es cero, el sistema no continúa, de lo contrario, el sistema detecta que hay una persona y sigue con el proceso.

```

36 void loop() {
37   Serial.print("\nNuevo Ciclo");
38   delay(2000);
39   digitalWrite(led_rojo,LOW);
40   digitalWrite(led_verde, LOW);
41   digitalWrite(led_naranja, LOW);
42   // SI HAY PRESENCIA SE ACTIVA EL MECANISMO, DE LO CONTRARIO
43   // APAGADO
44   // ENVIAR PULSO DE DISPARO EN EL PIN "TRIGGER"
45   digitalWrite(trigger, LOW);
46   delayMicroseconds(2);
47   digitalWrite(trigger, HIGH);
48   // EL PULSO DURA AL MENOS 10 uS EN ESTADO ALTO
49   delayMicroseconds(10);
50   digitalWrite(trigger, LOW);
51   // MEDIR EL TIEMPO EN ESTADO ALTO DEL PIN "ECHO" EL PULSO ES PROPORCIONAL A LA DISTANCIA MEDIDA
52   tiempo = pulseIn(echo, HIGH);
53
54   // LA VELOCIDAD DEL SONIDO ES DE 348 M/S O 29 MICROSEGUNDOS POR CENTIMETRO
55   // DIVIDIMOS EL TIEMPO DEL PULSO ENTRE 58, TIEMPO QUE TARDA RECORRER IDA
56   // Y VUELTA UN CENTIMETRO LA ONDA SONORA

```

Fig. 8. Fragmento del código utilizado para la simulación

Fuente: elaborado en Tinkercad

En la figura 9, se observa en la línea 70, como se calcula la distancia de la persona en metros. En la línea 87 se guarda la señal analógica que retorna el sensor infrarrojo, simulado con un sensor *tmp 36*, en la variable “*tmpPin*” para luego determinar la temperatura corporal de la persona mediante la fórmula de la línea 88. Posteriormente, con una condicional se prohíbe el ingreso de la persona si su temperatura corporal es superior a 37.5 °C, como se observa en la línea 92, caso contrario se permite el acceso de la persona.

```

69   Serial.print("\nLa persona mide (m):");
70   Serial.println(distancia/100);
71   delay(3000);
72   Serial.print("Altura a posicionarse del sensor de temperatura (m):");
73   Serial.println((distancia/100)*0.8);
74   delay(3000);
75   Serial.print("Motores en movimiento ... \n");
76
77   digitalWrite(motorizq, HIGH);
78   digitalWrite(motorder,LOW);
79   delay(5000);
80
81   Serial.print("Sensores en posicion ... \n");
82   delay(2000);
83
84   // temperatura
85   Serial.print("Tomando temperatura... \n");
86   delay(2000);
87   tmpPin = analogRead(A0);
88   temperatura = ((tmpPin)*(500.0 / 1023.0)) - 50.0;
89   Serial.print("Temperatura de la persona (celcius):");
90   Serial.println(temperatura);
91   delay(2000);
92   if (temperatura > 37.5){
93     Serial.print("La persona presenta fiebre");
94     delay(2000);
95     digitalWrite(led_naranja, LOW);

```

Fig. 9. Fragmento del código utilizado para la simulación

Fuente: elaborado en Tinkercad

### 2.3. Diseño del Sistema de Detección de Mascarillas

El desarrollo del sistema de detección de mascarilla facial se realizó en dos etapas: la detección de rostro y la detección de mascarilla facial. En la primera etapa, utilizando el software OpenCV se captura en tiempo real el video proveniente de la cámara y mediante el módulo de Face Detection de MediaPipe [6] se procede a obtener el rostro de las personas que se encuentren en el campo de visión para luego procesarlos en la etapa de detección de mascarillas.

En la segunda etapa, se procedió a reentrenar la red neuronal convolucional MobileNet [7] debido a que es un modelo eficiente y ligero por lo que se puede ejecutar con pocos recursos en la Raspberry Pi. Para la detección de mascarillas se modificó la capa de entrada para que pueda procesar una imagen de 128x128 píxeles en formato RGB y la capa de salida para indicar si cuenta o no con mascarilla utilizando la función de activación Sigmoidal. Para el entrenamiento se utilizaron 12,800 imágenes y para la validación se utilizaron 790 imágenes, las cuales fueron obtenidas de Kaggle [8], [9] y se pre-procesan al tamaño y formato concordante a la capa de entrada de la red neuronal convolucional. Se obtuvo un 99.82% de precisión como máximo con los datos de entrenamiento y un 100% de precisión como máximo con los datos de validación, los cuales se observan en la figura 10.

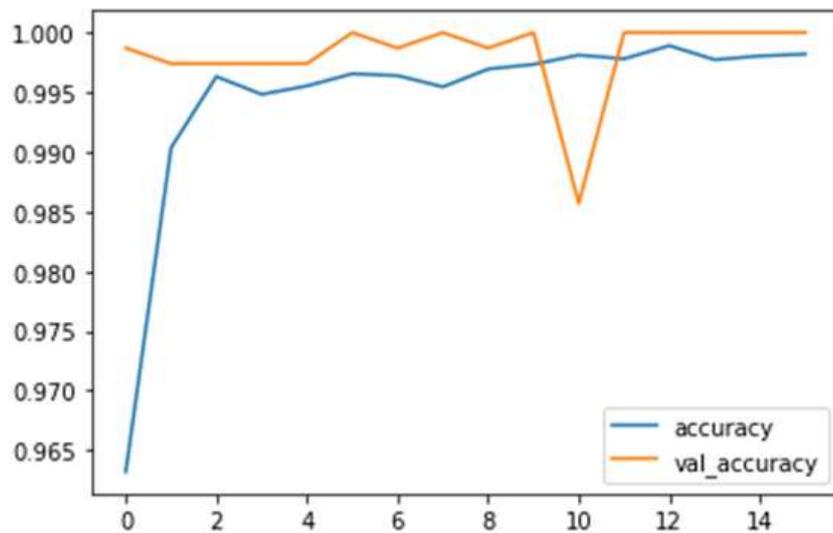


Fig. 10. Gráfico de precisión vs épocas del entrenamiento

Luego del entrenamiento, se procedió a implementar el modelo obtenido en el Sistema de Detección de Mascarillas. Las imágenes de los rostros obtenidos con MediaPipe son pre procesadas para cumplir con las condiciones de la capa de entrada de la red neuronal convolucional para identificar la presencia o no de las mascarillas, lo cual se indica mediante un mensaje en la visualización de la imagen que cambia entre el color rojo y verde, el cual

indica la ausencia o presencia de la mascarilla facial respectivamente, además, envía el resultado al sistema principal del Robot Cartesiano para permitir o no el acceso dependiendo de la temperatura corporal y si cuenta o no con mascarilla facial. El diagrama de flujo de datos del sistema de detección de mascarilla facial se observa en la figura 11.

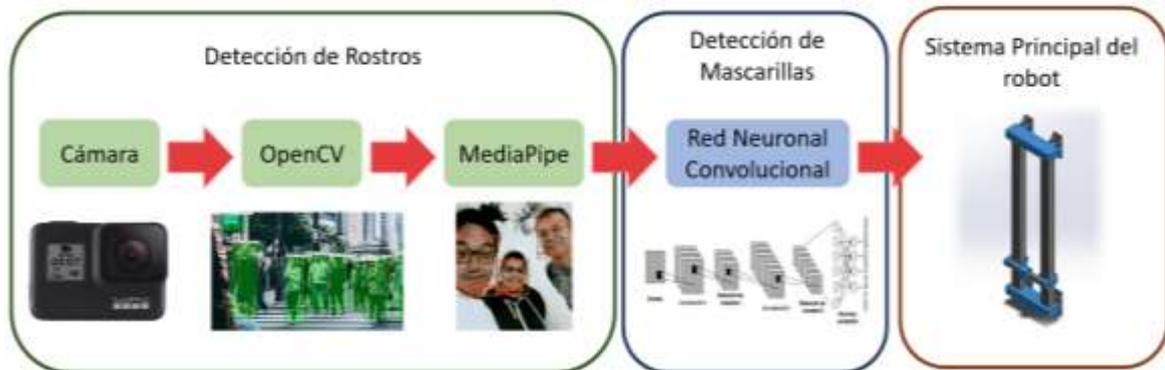


Fig. 11. Diagrama de flujo de datos del Sistema de Detección de Mascarillas

### 3. PRUEBAS Y RESULTADOS

#### Prueba 1

Se realizaron las pruebas del sistema de detección de mascarilla facial utilizando la IDE Spyder para su ejecución. Cuando la persona no posee la mascarilla facial, se observa un recuadro de color rojo y la etiqueta “Sin\_Mascarilla” encima del rostro como se muestra en la figura 12.



Fig. 12. Detección de persona sin mascarilla

En caso contrario, el recuadro es de color verde y presenta la etiqueta “Con\_Mascarilla”, tal como se muestra en la figura 13.



Fig. 13. Detección de persona con mascarilla

## Prueba 2

En esta prueba se simuló el posicionamiento del sensor infrarrojo y el ingreso mediante luces, siendo verde el óptimo cuando está en el rango de  $36^{\circ}$  a  $37.5^{\circ}\text{C}$ , de color naranja si está en proceso y rojo si está fuera de los parámetros normales, tal como se muestra, en las figuras 14, 15 y 16 respectivamente.

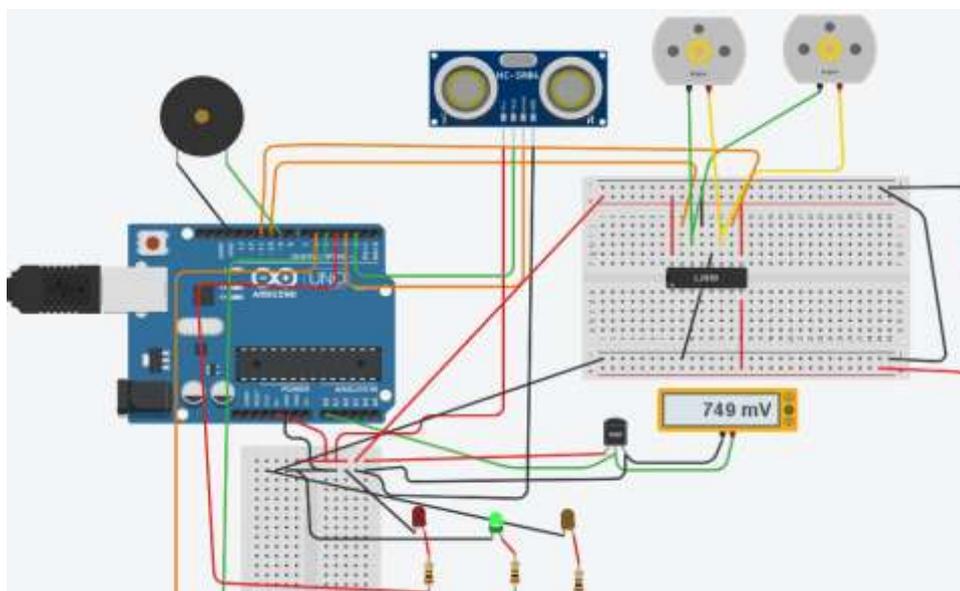


Fig. 14. Señal de acceso permitido después de medición de temperatura

Fuente: elaborado en Tinkercad

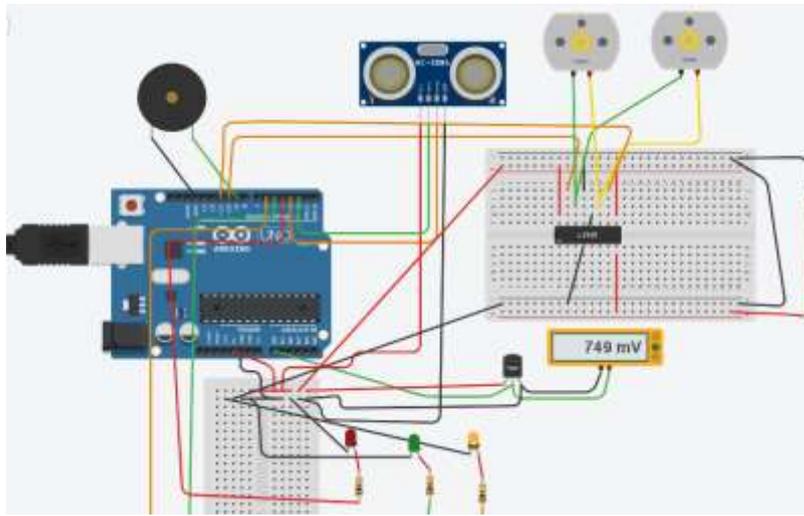


Fig. 15. Señal de medición de temperatura en proceso elaborado en Tinkercad

Fuente: Elaboración propia

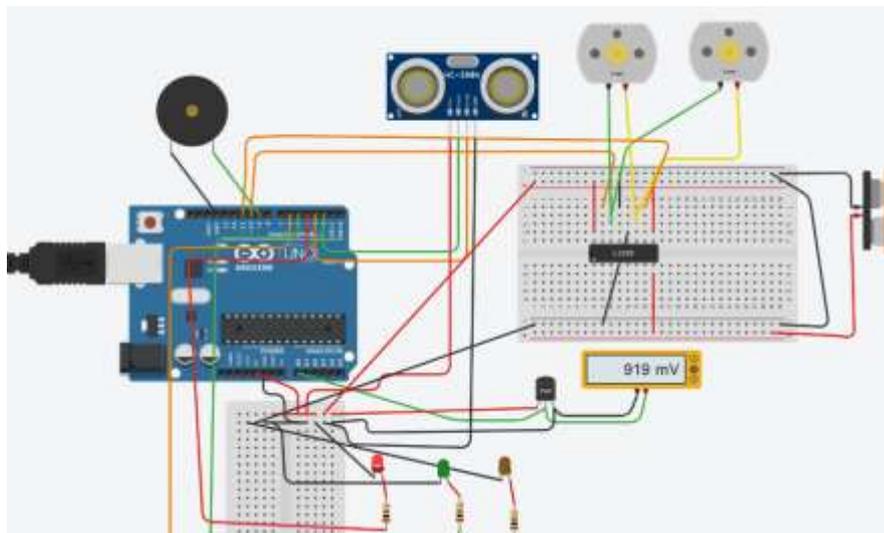


Fig. 16. Señal de acceso denegado después de medición de temperatura elaborado en Tinkercad

Fuente: Elaboración propia

### Prueba 3

En esta prueba se realizó la simulación del robot cartesiano en el software CoppeliaSim, primero se realizó la importación de los archivos *stl* del ensamblaje; luego se realizó la programación del sensor ultrasónico y de los motores para que cuando se detecte a la persona, el sensor de ultrasonido mida su altura y luego se realiza el movimiento cartesiano al 80% de la medida obtenida para que realice la medición de la temperatura. En la figura 17 se observa el instante en el que la persona aún no llega al rango de detección del sensor de ultrasonido por la cual el robot se encontró en su posición inicial, y en la figura 18 se observa el instante en el que la persona fue detectada y el robot cartesiano se desplazó para realizar la medición de la temperatura corporal.

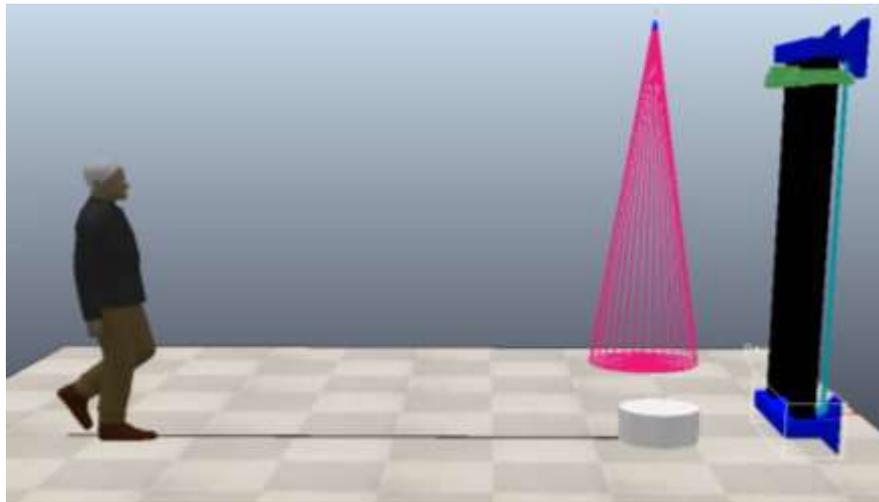


Fig. 17. Simulación inicial del robot cartesiano  
Fuente: elaborado en CoppeliaSim V4.2.0

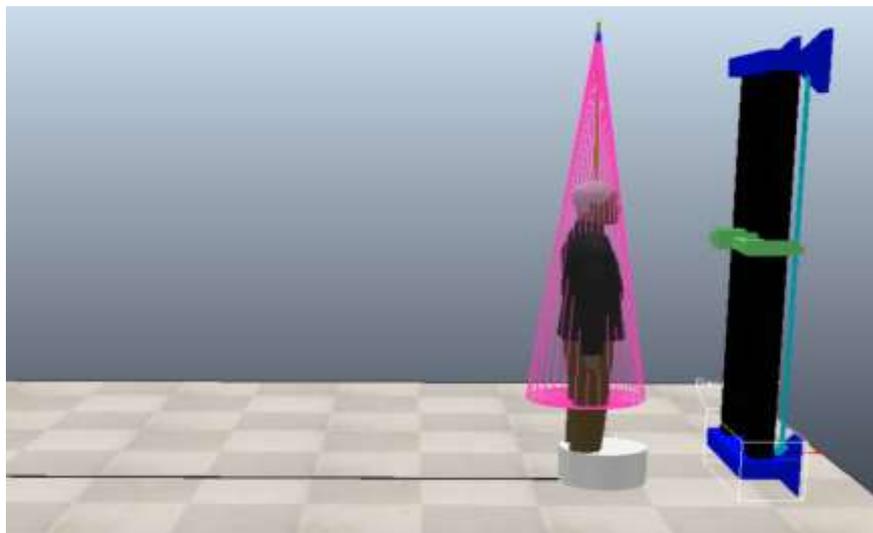


Fig. 18. Simulación de persona detectada y desplazamiento del módulo de medición de temperatura corporal  
Fuente: elaborado en CoppeliaSim V4.2.0

#### 4. CONCLUSIONES

Para el diseño del robot cartesiano, se tuvo en consideración diversos tipos de soportes estructurales. Inicialmente, se evaluó empotrarlo en el suelo, sin embargo, debido a su altura de casi 2 m y su masa de aproximada de 200kg, presentaría un alto grado de inestabilidad, con el riesgo de caída, por lo que, se optó por empotrarlo a la pared a través de 04 soportes instalados en cada esquina.

En el diseño electrónico, se evaluó el uso de una variedad de motores DC y sensores, tanto de ultrasonido como infrarrojo, determinando el uso de aquellos modelos disponibles en el

mercado local con la finalidad de contar, en el corto plazo, con un primer prototipo del robot cartesiano para su implementación y validación. [10]

Se observó durante las pruebas del sistema de detección de mascarilla facial que cuando una persona se tapa la boca con las manos o con algún objeto diferente a una mascarilla facial el sistema detecta que si tiene mascarilla, por lo que, en la segunda fase del proyecto, se añadirá fotografías de personas tapándose la cara en el *dataset* para que el sistema sea capaz de identificar estos casos adecuadamente.

Finalmente, en la simulación mediante el software CoppeliaSim, se tuvo que agregar un filtro de media exponencial a la medida obtenida con el sensor ultrasónico, de tal manera, que el movimiento del robot cartesiano sea suave.

## 5. REFERENCIAS

- [1] Sala Situacional Covid-19 Perú del MINSA (2021). Situación Actual Covid19 Perú 2020-2021. <https://www.dge.gob.pe/portal/docs/tools/coronavirus/coronavirus140821.pdf>
- [2] Godoy, R. y Rodríguez, W. (2007). Diseño y modelamiento de un robot cartesiano para el posicionamiento de piezas. [https://ciencia.lasalle.edu.co/ing\\_automatizacion/50](https://ciencia.lasalle.edu.co/ing_automatizacion/50)
- [3] Rodríguez, C. (2017). Diseño y construcción de un robot cartesiano con un control de posición de punto a punto. <https://repository.unilibre.edu.co/handle/10901/11249>
- [4] Naylamp Mechatronics. (2021). Sensor de ultrasonido HC-SR04. <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- [5] Naylamp Mechatronics. (2021). Sensor de temperatura infrarrojo. <https://naylampmechatronics.com/sensores-temperatura-y-humedad/330-sensor-de-temperatura-mlx90614.html>
- [6] MediaPipe. (2021). MediaPipe Face Detection. [https://google.github.io/mediapipe/solutions/face\\_detection](https://google.github.io/mediapipe/solutions/face_detection)
- [7] Venkateswarlu, I., Kakarla, J. y Prakash, S. (2020) Face mask detection using MobileNet and Global Pooling Block <https://ieeexplore.ieee.org/abstract/document/9312083>
- [8] Kaggle. (2021). Face Mask Detection 12k Images Dataset. <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset?select=Face+Mask+Dataset>
- [9] Kaggle. (2021). Face Mask Detection Dataset. <https://www.kaggle.com/omkargurav/face-mask-dataset>
- [10] Richard M. Ñope-Giraldo, Luis A. Illapuma-Ccallo, R. Palomares, J. Cornejo y otros "Mechatronic Systems Design of ROHNI-1: Hybrid Cyber-Human Medical Robot for COVID-19 Health Surveillance at Wholesale-Supermarket Entrances", 2021 Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE), 2021, pp. 1-7, doi: 10.1109/GMEPE/PAHCE50215.2021.9434874.