



GRUPO TECNOLÓGICO UNIVERSITARIO

CAMPUS TOLUCA

PROYECTO INTEGRADOR:

CONTROL DE DISPOSITIVOS POR GESTOS ARTICULADOS.

INGENIERÍA EN SISTEMAS COMPUTACIONALES

INTEGRANTES:

Bruno Carbajal Bladimir Santiago

Enriquez González Luis Antonio

Gómez De La Cruz Israel

Jiménez Dado Efraín

Martínez Hernández Jonathan Eduardo

Morales Diaz Geraldine Michelle

Pontón Hernández Alejandro

Santamaría Juárez Kevin

Torres Flores José María

PROFESORES:

Mtro. Alan Ismael Montes de Oca Arcos (Programación Orientada a Objetos).

L.I.A. Jorge Luis Cruz Becerril (Estructura de Datos).

Ing. Arturo Morales Cuadros (Cálculo Vectorial).

D.E. Claudia Galindo Carmona (Bases Metodológicas de la Investigación y Estadística Aplicada).

FECHA DE PRESENTACIÓN: 06/08/2025.

Control de dispositivos por gestos articulados.

ÍNDICE

INTRODUCCIÓN	4
FASE I: IDEOLÓGICA	6
1.1 DESCRIPCIÓN DE LA IDEA.....	6
1.2 IDENTIFICACIÓN DE LA NECESIDAD.....	6
1.3 NOMBRE DEL PROYECTO	7
1.4 ANTECEDENTES.....	7
FASE II: CONTEXTUAL.....	8
2.1 PROBLEMÁTICA	8
2.2 JUSTIFICACIÓN	8
2.3 OBJETIVOS.....	9
2.3.1 OBJETIVO GENERAL	9
2.3.2 OBJETIVOS ESPECÍFICOS	9
FASE III: MARCO TEORICO	10
3.1. ANTECEDENTES.....	10
3.2. ¿QUÉ SON LOS GESTOS ARTICULADOS?.....	10
3.3. FUNCIONAMIENTO	11
3.4. ÁREAS DONDE SE UTILIZA.....	11
FASE IV: DESARROLLO	13
4.1. ¿PARA QUIÉN ESTÁ DIRIGIDO?	13

Control de dispositivos por gestos articulados.

4.2. DESCRIPCIÓN.....	13
4.3. RECURSOS Y MATERIALES NECESARIOS.....	15
4.4. SIMULACIÓN	16
FASE V: CONCLUSIÓN.....	17
5.3.1 CONCLUSIONES.....	17
5.3.2 RECOMENDACIONES	17
5.3.3 ANEXOS.....	19
5.3.1 ANEXO 1	19
5.3.2 ANEXO 2	19
5.3.3 ANEXO 3	19
5.3.4 ANEXO 4	19
5.3.5 ANEXO 5	19
5.3.6 ANEXO 6	19
5.3.7 ANEXO 7	19
5.3.8 ANEXO 8	19
5.3.9 ANEXO 9	19
5.3.4 FUENTES DE CONSULTA.....	47

Control de dispositivos por gestos articulados.

INTRODUCCIÓN

Los gestos articulados son movimientos del cuerpo humano que forman parte del habla, ayudando a enfatizar o estructurar un discurso comunicarnos. Hoy en día, también representan una forma de comunicación no verbal que permite transmitir mensajes sin necesidad de palabras. Sin embargo, su uso no se limita a la interacción entre personas; esta forma de expresión también ha comenzado a implementarse en el ámbito tecnológico, abriendo nuevas posibilidades para mejorar la accesibilidad y el control de dispositivos.

La tecnología ha avanzado de forma considerable y ha sido clave para facilitar muchas tareas específicas en distintos ámbitos. Actualmente, los gestos articulados ya se utilizan en equipos de cómputo, permitiendo una interacción más natural e intuitiva con dispositivos como teclados, mouse, pantallas táctiles, entre otros. Esta funcionalidad se le conoce como control por gestos o reconocimiento de movimientos, y permite que los usuarios interactúen con los sistemas a través de sus movimientos físicos.

El reconocimiento de gestos no se limita únicamente al control de computadoras; su potencial abarca campos como la medicina, el desarrollo automotriz, la automatización industrial e incluso la educación. Este proyecto surge precisamente con la intención de explorar ese potencial, aplicándolo al entorno del hogar.

Es un proyecto desarrollado con fines meramente académicos, en el que se busca experimentar con esta tecnología y explorar nuevas formas de facilitar tareas domésticas. El objetivo principal es controlar diferentes dispositivos a partir de gestos de las manos, ofreciendo mayor comodidad en la vida diaria y aprovechando las herramientas tecnológicas actuales para hacer más sencillas las actividades del hogar.

Control de dispositivos por gestos articulados.

El documento de este proyecto se estructura en varias fases con el objetivo de organizar el proceso de creación del control de dispositivos por gestos articulados, de una manera lógica y eficiente. Cada fase cumple una función específica dentro del desarrollo, permitiendo un enfoque ordenado y metodológico. De esta manera, se abordan desde la concepción de la idea hasta su implementación y evaluación final. La división en fases facilita la identificación de problemas, la asignación de recursos y la optimización del tiempo de trabajo. Además, permite asegurar que se cumplan los objetivos planteados en cada etapa. Gracias a esta organización, el proyecto puede avanzar de manera estructurada y con una base teórica bien definida.

Control de dispositivos por gestos articulados.

FASE I: IDEOLÓGICA

1.1 DESCRIPCIÓN DE LA IDEA

Los gestos articulados o sistemas de reconocimiento de gestos son una innovación dentro del área de la tecnología e innovación, lo que significa que gestos humanos como mover un brazo, abrir o cerrar los dedos, ya se pueden utilizar para manipular algún dispositivo electrónico, específicamente un dispositivo externo como lo es un mouse, teclado, etc., utilizando una combinación de cámaras, sensores, algoritmos de aprendizaje, entre otros elementos, a fin de ofrecer una interacción más intuitiva y natural.

Este proyecto consiste en realizar un sistema que permita controlar diferentes dispositivos electrónicos. Sin embargo, esta implementación estará dirigida a casas habitación, de tal manera que nos facilite realizar diferentes tareas dentro de ellas. Su principal uso será en la apertura de puertas e incluso ventanas en las que se registrará un gesto en específico para que, al ser visualizado desde un equipo de cómputo, este permita abrir fácilmente la puerta e incluso ventanas.

1.2 IDENTIFICACIÓN DE LA NECESIDAD

La tecnología está cada vez más presente en muchas áreas de nuestra vida. Permite diversas aplicaciones en diferentes sectores gracias a su interacción sin contacto, facilidad de uso y ventajas de higiene. A medida que esta tecnología se vuelve más común, la experiencia del usuario también mejora significativamente. Esta amplia gama de tecnología ofrece soluciones a las necesidades de empresas y usuarios de diferentes sectores. Con el desarrollo de esta tecnología, se hace posible percibir movimientos más complejos y sensibles, lo que abre el camino para la aparición de nuevas áreas de aplicación.

Control de dispositivos por gestos articulados.

1.3 NOMBRE DEL PROYECTO

Control de Dispositivos por Gestos Articulados

1.4 ANTECEDENTES

Es un proyecto para fines académicos con el fin de innovar el sistema de control de dispositivos, permitiendo que la mayoría de las personas logren entender lo que se pretende desarrollar e innovar en el área de la tecnología, involucrando cada una de las áreas de la ingeniería en sistemas computacionales con módulos del tercer cuatrimestre.

Control de dispositivos por gestos articulados.

FASE II: CONTEXTUAL

2.1 PROBLEMÁTICA

Actualmente, todavía es complejo poder adquirir un sistema de control por los costos elevados que estos tienen en el mercado, debido a que esta tecnología requiere sensores, cámaras y software avanzados. Además, la precisión y exactitud en la producción conllevan altos costos para el usuario, resultando en dificultades para su compra.

Por otro lado, los factores ambientales, las condiciones de iluminación y las características físicas tanto de los usuarios como del lugar de instalación pueden afectar la precisión en la detección de movimientos, provocando errores e interrupciones en la experiencia del usuario.

2.2 JUSTIFICACIÓN

El control de dispositivos por gestos es una nueva tecnología que busca facilitar algunas tareas para las personas en diferentes áreas. Sin embargo, actualmente únicamente se centra en la parte de la computación, es decir, solo está desarrollada para controlar dispositivos relacionados con la informática, por ejemplo: teclados, mouse, cámaras e incluso dispositivos más independientes como las impresoras.

Dada la necesidad de cada usuario, se busca innovar lo que ya existe pero en otra área poco estudiada, como lo son las casas. Se pretende crear una variante de lo que se conoce como control por gestos, experimentando en una casa a escala, que permita realizar diferentes actividades relacionadas con las tareas domésticas.

Cabe señalar que, al ser un proyecto con fines académicos, será utilizado como medio de aprendizaje en el cual, se realizará un prototipo en el que se llevará a cabo la experimentación previa.

Control de dispositivos por gestos articulados.

2.3 OBJETIVOS

2.3.1 OBJETIVO GENERAL

Controlar diferentes dispositivos a partir de gestos de las manos, ofreciendo mayor comodidad en diversas situaciones de la vida diaria, y facilitando las tareas del hogar mediante las tecnologías disponibles por el reconocimiento de gestos.

2.3.2 OBJETIVOS ESPECÍFICOS

- Diseñar un sistema de control de gestos que permita detectar y traducir movimientos manuales a comandos digitales.
- Lograr la vinculación entre los lenguajes de programación Python y la variante de C++, Arduino, para así activar y controlar los diferentes dispositivos.
- Experimentar con pruebas funcionales que demuestren la efectividad del sistema.

Control de dispositivos por gestos articulados.

FASE III: MARCO TEORICO

3.1.ANTECEDENTES

Las tecnologías de reconocimiento de gestos tienen sus raíces en los primeros desarrollos de la informática y la ingeniería, impulsadas por el objetivo de hacer que la interacción entre humanos y máquinas sea cada vez más natural e intuitiva. A lo largo del tiempo, investigadores y desarrolladores han explorado diversos métodos que permitan a los usuarios comunicarse con los dispositivos a través del movimiento.

En sus inicios, este tipo de tecnologías requerían equipos complejos y costosos. Sin embargo, con el avance de disciplinas como la visión artificial, el aprendizaje automático, las tecnologías de sensores y la inteligencia artificial, los sistemas de reconocimiento de gestos han evolucionado significativamente. Estos avances han permitido una mayor precisión en la percepción y el análisis de los movimientos humanos, abriendo el camino a aplicaciones prácticas en sectores como el entretenimiento, la salud, la automoción, la industria y más.

(Limited, H. G. 2025)

3.2.¿QUÉ SON LOS GESTOS ARTICULADOS?

Los gestos articulados son movimientos corporales principalmente de manos, brazos e incluso de la cabeza, que ayudan a complementar al lenguaje hablado. Este tipo de gestos pueden transmitir un mensaje por sí solo, como sucede con el lenguaje de señas.

Dentro del área de la tecnología, los gestos articulados son un campo innovador que permiten controlar el funcionamiento de un dispositivo electrónico interpretando los movimientos físicos del usuario. Estos gestos forman parte de lo que se conoce como sistemas de reconocimiento de gestos, capaces de revolucionar no solo a la industria del entretenimiento sino también áreas de la salud, la automotriz, la educación y la manipulación automática industrial.

Control de dispositivos por gestos articulados.

3.3.FUNCIONAMIENTO

Esta tecnología funciona detectando e interpretando los movimientos humanos como resultado de un proceso de diseño de algoritmo y programación, con el fin de interactuar y controlar un sistema sin contacto físico directo. Básicamente, el sistema de reconocimiento de gestos captura los movimientos del usuario, procesa los datos y luego los traduce en comandos predefinidos. De esta forma, los usuarios pueden controlar dispositivos o sistemas con solo gestos de la mano o movimientos del cuerpo.

3.4.ÁREAS DONDE SE UTILIZA

Hoy en día la tecnología está cada vez más presente en muchas áreas de nuestra vida innovando y mejorando lo que ya existe a fin de que todo resulte ser más fácil. Estos sistemas permiten diversas aplicaciones en diferentes sectores por su interacción sin contacto y la facilidad de uso principalmente. A medida que esta tecnología se vuelve más común, la experiencia del usuario también mejora significativamente. Actualmente son aplicados en áreas como:

- **Industria del juego y el entretenimiento:** Los usuarios interactúan con los juegos mediante movimientos corporales. Dispositivos como Microsoft Kinect, PlayStation detentan los movimientos en tiempo real. Sus aplicaciones varían desde controlar avatares hasta simular acciones deportivas, bailes o combates (García Cortés 2014).
- **Servicios de salud:** Utilizados en el diagnóstico de enfermedades y rehabilitación de pacientes con afectaciones en extremidades, cirugías asistidas y apoyo a personas con discapacidades (Reynoso Martínez 2019).

Control de dispositivos por gestos articulados.

- **Industria automotriz:** Integrados en vehículos modernos para mejorar la seguridad y comodidad de los conductores, en el control de funciones del automóvil y sistema de asistencia inteligente (Rodríguez 2024).
- **Automatización industrial:** Empleados en máquinas industriales y paneles operativos en control remoto de brazos robóticos, supervisión de procesos, entre otros (Rodríguez 2024).

Control de dispositivos por gestos articulados.

FASE IV: DESARROLLO

4.1.¿PARA QUIÉN ESTÁ DIRIGIDO?

- **Usuarios residenciales:** Personas que buscan facilitar la ejecución de tareas en el hogar (FasterCapital 2025).
- **Personas con discapacidades:** Facilita el manejo de luces, puertas, electrodomésticos, etc. (ElectroHogar 2024).
- **Usuarios de tecnología Smart home:** Interesados en domótica y automatización del hogar que busquen soluciones innovadoras (NOU GRUP 2022).
- **Arquitectos y diseñadores:** Profesionales que diseñan espacios inteligentes y desean integrar tecnologías accesibles (NOU GRUP 2022).
- **Sector educación:** Estudiantes interesados en innovar en tecnología, desarrollando interfaces hombre-maquina (SER C 2025).
- **Sector salud:** Personas temporalmente imposibilitadas y requieran rehabilitación física (SER C 2025).

4.2.DESCRIPCIÓN

Mediante la programación se busca innovar los sistemas de control de dispositivos, implementándolos en hogares para facilitar las tareas del hogar. Desde un equipo de cómputo se podrán precargar diferentes gestos corporales de las extremidades superiores, los cuales el usuario realizará frente a la cámara. Este identificará el gesto y ejecutará la acción correspondiente.

Las acciones disponibles son:

- Encender un foco LED.
- Activar un ventilador a escala.

Control de dispositivos por gestos articulados.

- Encender un zumbador que funcionará como alarma.
- Operar un motor que permitirá abrir una puerta.

Para el correcto funcionamiento del prototipo, se trabajó en dos lenguajes de programación; Arduino y Python cada uno con funciones específicas.

La variante de C++, Arduino, se encarga del funcionamiento de la placa Arduino UNO donde se carga cada una de las funciones, la apertura de la puerta y el encendido de un foco LED, entre otras. Python, realiza la lectura de los del usuario capturados por la cámara, los cuales harán posible el funcionamiento de los dispositivos.

Gestos Soportados

Gestos (Dedos extendidos)	Comando Enviado	Acción en Arduino
00000 (Puño)	ALL_OFF	Apaga todos los dispositivos.
11111 (Mano abierta)	LED_ON	Enciende el LED.
01100 (Índice + medio)	FAN_ON	Activa el ventilador.
10000 (Solo pulgar)	BUZZER_ON	Enciende el zumbador.
00111 (Anular + meñique + medio)	DOOR_OPEN	Servo a 180° (abierto).
00001 (Solo meñique)	DOOR_CLOSE	Servo a 0° (cerrado).
01000 (Solo índice)	DOOR_SET_ANGLE=90	Servo a 90° (mitad).

-  (Puño) → Apaga todos los dispositivos.

Control de dispositivos por gestos articulados.

-  (Mano abierta) → Enciende el LED.
-  (Índice + medio) → Activa el ventilador.
-  (Pulgar arriba) → Enciende el zumbador.
-  (Anular + meñique + medio) → Abre el servo (180°).
-  (Solo meñique) → Cierra el servo (0°).

4.3.RECURSOS Y MATERIALES NECESARIOS

Para lograr el correcto funcionamiento del prototipo de este prototipo se utilizaron diversos materiales los cuales, se enlistan a continuación: (Para consultar sus características ver Anexo del 1 al 7)

- 1 equipo de cómputo.
- Arduino UNO.
- 1 Servomotor.
- 1 Buzzer (Zumbador).
- 1 Motor Reductor con aspas.
- 1 Batería.
- Cable estañado.
- Leds.

Control de dispositivos por gestos articulados.

4.4. SIMULACIÓN

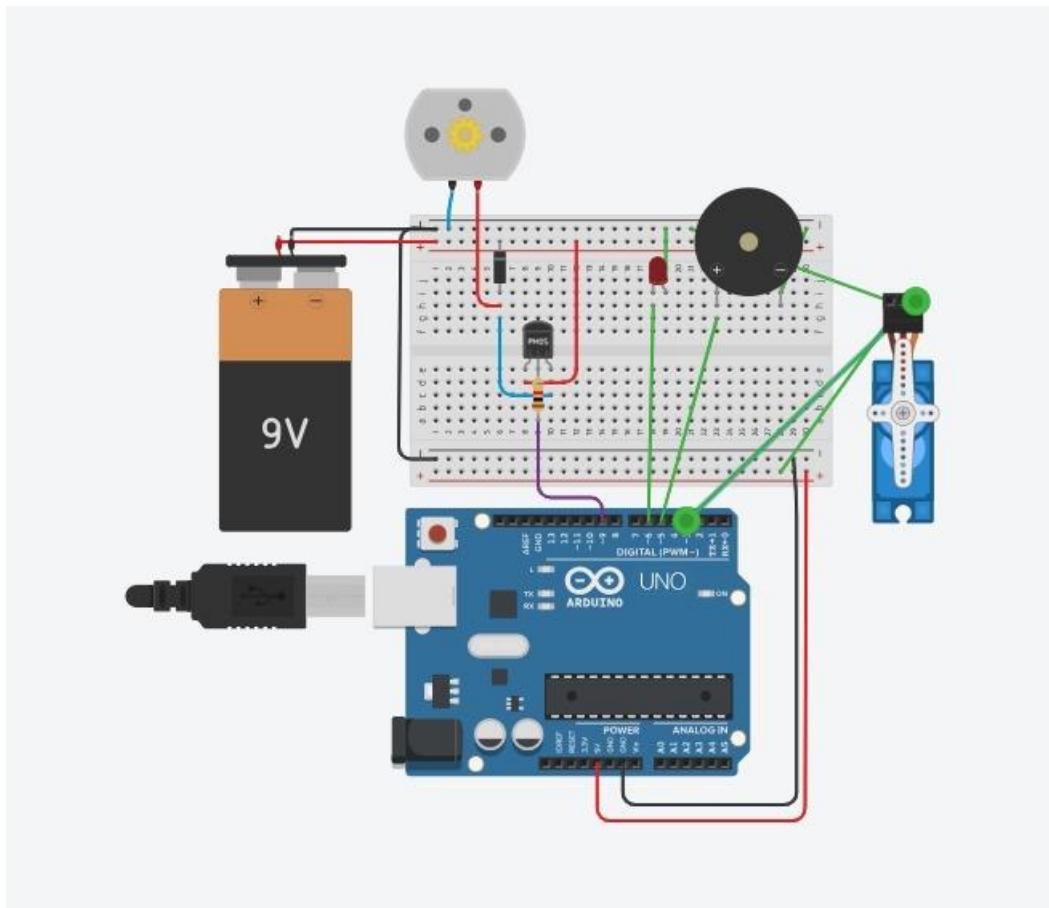


Figura 1 Simulación del Sistema

Control de dispositivos por gestos articulados.

FASE V: CONCLUSIÓN

5.3.1 CONCLUSIONES

La evolución de la tecnología ha demostrado que la interacción entre el ser humano y los dispositivos electrónicos puede ser cada vez más natural y accesible. Este proyecto "Control de Dispositivos por Gestos Articulados" es un claro ejemplo de cómo una idea innovadora puede pasar de sectores tecnológicos, como la computación o el entretenimiento, a entornos más cotidianos como el hogar. Esta propuesta más que poner en práctica los conocimientos de programación, también responde a una necesidad: facilitar tareas comunes mediante una interfaz intuitiva y sin contacto.

Durante el desarrollo, se combinó programación con sensores y cámaras para que una computadora pueda leer ciertos movimientos y, con eso, mandar órdenes a distintos dispositivos. Aunque suene complicado, la idea es que el usuario solo tenga que hacer un gesto sencillo para que el sistema lo reconozca y actúe por él. Además, nos permitió a nosotros como estudiantes interactuar con diferentes Softwares de aplicación como lo fue Python y Arduino que como tal no es un lenguaje de programación sino que es una variante de C++.

Con este proyecto una vez más nos pudimos dar cuenta que nada es tan sencillo como parece esto debido a que tuvimos que hacer que ambos lenguajes de programación trabajaran en conjunto, sin mencionar las fallas que fueron surgiendo tanto en la programación como en el circuito.

5.3.2 RECOMENDACIONES

No hay actividad y/o proyecto que sea sencillo de hacer, sin embargo, hay habilidades que ya las tenemos desarrolladas y que por supuesto, no es nada difícil sacar adelante lo que se pretende hacer. La principal recomendación que podemos dar con la experiencia que tuvimos

Control de dispositivos por gestos articulados.

durante este proyecto es que por muy difícil que parezca tiene que salir a flote, sin importar cada obstáculo que vaya suscitando porque al final del día se tienen que superar.

Y, por último, trabajar en equipo facilita el desarrollo y por ende la ejecución de un proyecto, sin embargo, si una o varias partes de las involucradas no tienen iniciativa de continuar, no hay ningún motivo para detener el proyecto porque como se dice así sea uno contra el mundo entero, una mentalidad fuerte podrá con eso y más.

Control de dispositivos por gestos articulados.

5.3.3 ANEXOS

5.3.1 ANEXO 1

5.3.2 ANEXO 2

5.3.3 ANEXO 3

5.3.4 ANEXO 4

5.3.5 ANEXO 5

5.3.6 ANEXO 6

5.3.7 ANEXO 7

5.3.8 ANEXO 8

5.3.9 ANEXO 9

CÓDIGO EN ARDUINO

```
import cv2
import mediapipe as mp
import serial
import time
import numpy as np
from matplotlib.backends.backend_agg import FigureCanvasAgg
from matplotlib import pyplot as plt
import pygame
from pygame.locals import *

# Configuración serial - cambiar COM según tu sistema
SERIAL_PORT = 'COM3'
```

Control de dispositivos por gestos articulados.

BAUD_RATE = 9600

try:

```
arduino = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)  
time.sleep(2) # Espera para inicialización  
print(f'Conexión establecida con Arduino en {SERIAL_PORT}')
```

except serial.SerialException as e:

```
print(f'Error de conexión: {e}')
```

```
arduino = None
```

Configuración de MediaPipe Hands

```
mp_hands = mp.solutions.hands  
mp_drawing = mp.solutions.drawing_utils  
mp_drawing_styles = mp.solutions.drawing_styles
```

hands = mp_hands.Hands(

```
    max_num_hands=1,  
    min_detection_confidence=0.8,  
    min_tracking_confidence=0.8
```

)

Inicializar cámara

```
cap = cv2.VideoCapture(0)
```

Control de dispositivos por gestos articulados.

```
if not cap.isOpened():

    print("Error al abrir la cámara")

    exit()

# Inicializar pygame para interfaz gráfica

pygame.init()

info = pygame.display.Info()

screen_width, screen_height = info.current_w - 100, info.current_h - 100

screen = pygame.display.set_mode((screen_width, screen_height))

pygame.display.set_caption("Control por Gestos - Sistema de Domótica")
```

Colores

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

GREEN = (0, 255, 0)

RED = (255, 0, 0)

BLUE = (0, 0, 255)

YELLOW = (255, 255, 0)

ORANGE = (255, 165, 0)

Fuentes

font_large = pygame.font.SysFont('Arial', 30)

font_medium = pygame.font.SysFont('Arial', 24)

Control de dispositivos por gestos articulados.

```
font_small = pygame.font.SysFont('Arial', 18)

# Mapeo de gestos a comandos

GESTURE_COMMANDS = {

    "00000": {"cmd": "ALL_OFF", "desc": "Puño cerrado - Apagar todo", "color": RED},

    "11111": {"cmd": "LED_ON", "desc": "Mano abierta - Encender luces", "color": GREEN},

    "01100": {"cmd": "FAN_ON", "desc": "Paz y amor - Ventilador ON", "color": BLUE},

    "01111": {"cmd": "FAN_OFF", "desc": "Cuatro dedos - Ventilador OFF", "color": RED},

    "10000": {"cmd": "BUZZER_ON", "desc": "Solo pulgar - Alarma sonora", "color": ORANGE},

    "00111": {"cmd": "DOOR_OPEN", "desc": "Tres dedos - Abrir puerta", "color": GREEN},

    "00001": {"cmd": "DOOR_CLOSE", "desc": "Solo meñique - Cerrar puerta", "color": RED},

    "01000": {"cmd": "DOOR_SET_ANGLE=90", "desc": "Solo índice - Ángulo 90°", "color": YELLOW},

    "11000": {"cmd": "FAN_REVERSE", "desc": "Pulgar + índice - Reversa ventilador", "color": ORANGE}

}

# Estados adicionales para control progresivo
```

Control de dispositivos por gestos articulados.

```
GESTURE_CONTROL = {  
    "11111": {"action": "INCREASE", "target": "SERVO", "step": 5},  
    "00000": {"action": "DECREASE", "target": "SERVO", "step": 5},  
    "10101": {"action": "INCREASE", "target": "FAN", "step": 25},  
    "01010": {"action": "DECREASE", "target": "FAN", "step": 25}  
}
```

```
def count_fingers(hand_landmarks):  
  
    tips_ids = [4, 8, 12, 16, 20] # Pulgar, índice, medio, anular, meñique  
    fingers = []  
  
    # Detección pulgar (comparación eje X)  
    thumb_tip = hand_landmarks.landmark[tips_ids[0]]  
    thumb_dip = hand_landmarks.landmark[tips_ids[0]-1]  
    fingers.append(1 if thumb_tip.x < thumb_dip.x else 0)  
  
    # Detección otros dedos (comparación eje Y)  
    for id in range(1, 5):  
        tip = hand_landmarks.landmark[tips_ids[id]]  
        dip = hand_landmarks.landmark[tips_ids[id]-2]  
        fingers.append(1 if tip.y < dip.y else 0)  
  
    return "".join(map(str, fingers))
```

Control de dispositivos por gestos articulados.

```
def send_command(cmd):
    if arduino:
        try:
            arduino.write(f'{cmd}\n'.encode())
            print(f'Comando enviado: {cmd}')
            return True
        except Exception as e:
            print(f'Error enviando comando: {e}')
            return False
    else:
        print(f'Simulando comando: {cmd}')
        return True

def draw_hand_graph(hand_landmarks, size=(300, 300)):
    """Crea una gráfica de los puntos de la mano usando matplotlib"""
    fig, ax = plt.subplots(figsize=(4, 4), facecolor='black')
    fig.subplots_adjust(left=0, right=1, bottom=0, top=1)
    ax.set_xlim(0, 1)
    ax.set_ylim(1, 0) # Invertir eje Y para coincidir con imagen
    ax.axis('off')
    # Dibujar conexiones
```

Control de dispositivos por gestos articulados.

```
connections = mp_hands.HAND_CONNECTIONS

for connection in connections:

    start_idx = connection[0]

    end_idx = connection[1]

    start = hand_landmarks.landmark[start_idx]

    end = hand_landmarks.landmark[end_idx]

    ax.plot([start.x, end.x], [start.y, end.y], 'w-', linewidth=2)

# Dibujar puntos

for landmark in hand_landmarks.landmark:

    ax.plot(landmark.x, landmark.y, 'ro', markersize=5)

# Convertir a superficie pygame

canvas = FigureCanvasAgg(fig)

canvas.draw()

renderer = canvas.get_renderer()

raw_data = renderer.tostring_argb()

plt.close(fig)

# Cambiado de "RGB" a "ARGB" y ajustado el tamaño

size_pixels = (int(renderer.width), int(renderer.height))

surf = pygame.image.fromstring(raw_data, size_pixels, "ARGB")

return pygame.transform.scale(surf, size)
```

Control de dispositivos por gestos articulados.

```
def draw_gesture_info(surface, finger_state, command, servo_angle, fan_speed):
    """Dibuja información sobre el gesto detectado"""

    # Fondo del panel de información
    pygame.draw.rect(surface, (30, 30, 40), (0, 0, surface.get_width(), 120))

    # Texto de estado de dedos
    gesture_data = GESTURE_COMMANDS.get(finger_state, {"desc": "Gestos no
reconocido", "color": WHITE})

    fingers_text = font_large.render(f'Estado de dedos: {finger_state}', True,
gesture_data["color"])

    surface.blit(fingers_text, (20, 20))

    # Descripción del gesto
    gesture_text = font_medium.render(f'Gesto: {gesture_data["desc"]}', True, WHITE)

    surface.blit(gesture_text, (20, 55))

    # Comando actual
    cmd_text = font_medium.render(f'Comando: {command if command else "Ninguno"}',
True, GREEN if command else RED)

    surface.blit(cmd_text, (surface.get_width() - 400, 20))

    # Ángulo del servo
```

Control de dispositivos por gestos articulados.

```
servo_text = font_medium.render(f"Ángulo puerta: {servo_angle}°", True, WHITE)
surface.blit(servo_text, (surface.get_width() - 400, 55))
```

```
# Velocidad ventilador
```

```
fan_text = font_medium.render(f"Veloc. ventilador: {fan_speed}/255", True, WHITE)
surface.blit(fan_text, (surface.get_width() - 400, 85))
```

```
def draw_device_status(surface, devices):
```

```
    """Dibuja el estado de los dispositivos"""

```

```
    # Fondo del panel de estado
```

```
    pygame.draw.rect(surface, (40, 40, 50), (0, surface.get_height() - 150,
surface.get_width(), 150))
```

```
# Título
```

```
status_title = font_large.render("Estado de Dispositivos:", True, YELLOW)
surface.blit(status_title, (20, surface.get_height() - 140))
```

```
# Dispositivos
```

```
for i, device in enumerate(devices):
```

```
    x_pos = 20 + (i * 250)
```

```
    if x_pos < surface.get_width() - 200:
```

```
        # Icono representativo
```

```
        icon = font_large.render(device["icon"], True, device["color"])
```

Control de dispositivos por gestos articulados.

```
surface.blit(icon, (x_pos, surface.get_height() - 100))

# Texto del estado

state_text = font_medium.render(device["state"], True, device["color"])

surface.blit(state_text, (x_pos + 40, surface.get_height() - 100))

# Barra de progreso para elementos con valores

if "value" in device:

    pygame.draw.rect(surface, (70, 70, 80), (x_pos, surface.get_height() - 60, 200,
20))

    pygame.draw.rect(surface, device["color"], (x_pos, surface.get_height() - 60,
int(200 * (device["value"]/device["max"])), 20))

    value_text = font_small.render(f"{'device['value']} / {'device['max']}", True,
WHITE)

    surface.blit(value_text, (x_pos + 80, surface.get_height() - 55))

def parse_status_message(message):

    """Parsea el mensaje de estado del Arduino"""

    if not message.startswith("status:"):

        return None

    status_data = {}

    parts = message[7:].strip().split(",")

    for part in parts:
```

Control de dispositivos por gestos articulados.

for part in parts:

```
    key, value = part.split("=")
```

```
    status_data[key] = value
```

```
return status_data
```

```
prev_command = None
```

```
servo_angle = 90 # Ángulo inicial del servo
```

```
fan_speed = 0 # Velocidad inicial del ventilador
```

```
clock = pygame.time.Clock()
```

```
running = True
```

```
last_status = {}
```

```
gesture_active_time = 0
```

```
current_gesture = None
```

```
# Dispositivos iniciales
```

```
devices = [
```

```
    {"name": "Luces", "state": "OFF", "color": RED, "icon": "💡", "cmd_on": "LED_ON", "cmd_off": "LED_OFF"},  
  
    {"name": "Ventilador", "state": "OFF", "color": RED, "icon": "🌀", "value": 0, "max": 255, "cmd_on": "FAN_ON", "cmd_off": "FAN_OFF", "cmd_reverse": "FAN_REVERSE"},
```

Control de dispositivos por gestos articulados.

```
{"name": "Alarma", "state": "OFF", "color": RED, "icon": "⚠️", "cmd_on":  
"BUZZER_ON"},  
  
{ "name": "Puerta", "state": "90°", "color": BLUE, "icon": "🚪", "value": 90, "max":  
180,  
"cmd_open": "DOOR_OPEN", "cmd_close": "DOOR_CLOSE"}  
]
```

while running:

```
for event in pygame.event.get():  
  
    if event.type == QUIT or (event.type == KEYDOWN and event.key ==  
K_ESCAPE):  
  
        running = False  
  
    elif event.type == KEYDOWN:  
  
        if event.key == K_1:  
  
            send_command("LED_ON")  
  
        elif event.key == K_2:  
  
            send_command("LED_OFF")  
  
        elif event.key == K_3:  
  
            send_command("FAN_ON")  
  
        elif event.key == K_4:  
  
            send_command("FAN_OFF")  
  
        elif event.key == K_5:  
  
            send_command("BUZZER_ON")
```

Control de dispositivos por gestos articulados.

```
elif event.key == K_6:  
    send_command("DOOR_OPEN")  
  
elif event.key == K_7:  
    send_command("DOOR_CLOSE")  
  
elif event.key == K_0:  
    send_command("ALL_OFF")  
  
elif event.key == K_r:  
    send_command("FAN_REVERSE")  
  
  
# Leer estado del Arduino si hay datos disponibles  
  
if arduino and arduino.in_waiting > 0:  
  
    try:  
  
        status_message = arduino.readline().decode().strip()  
  
        last_status = parse_status_message(status_message) or last_status  
  
        print(f"Estado recibido: {status_message}")  
  
    except Exception as e:  
  
        print(f"Error leyendo estado: {e}")  
  
  
# Actualizar dispositivos basado en el último estado  
  
if last_status:  
  
    for device in devices:  
  
        if device["name"] == "Luces":  
  
            device["state"] = "ON" if last_status.get("led", "") == "on" else "OFF"
```

Control de dispositivos por gestos articulados.

```
device["color"] = GREEN if device["state"] == "ON" else RED
elif device["name"] == "Ventilador":
    speed = int(last_status.get("fan", "0"))
    device["value"] = speed
    device["state"] = f"{speed}/255"
    device["color"] = GREEN if speed > 0 else RED
    if "fan_dir" in last_status:
        if last_status["fan_dir"] == "reverse":
            device["state"] += " (R)"
    elif device["name"] == "Alarma":
        device["state"] = "ON" if last_status.get("buzzer", "") == "on" else "OFF"
        device["color"] = ORANGE if device["state"] == "ON" else RED
    elif device["name"] == "Puerta":
        angle = int(last_status.get("door", "90"))
        device["value"] = angle
        device["state"] = f"{angle}°"
        servo_angle = angle
# Capturar frame de la cámara
ret, frame = cap.read()
if not ret:
    continue
```

Control de dispositivos por gestos articulados.

```
frame = cv2.flip(frame, 1)

rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

results = hands.process(rgb_frame)

command = None

finger_state = "----"

hand_graph_surf = None


if results.multi_hand_landmarks:

    for hand_landmarks in results.multi_hand_landmarks:

        # Procesar gestos

        finger_state = count_fingers(hand_landmarks)

        gesture_info = GESTURE_COMMANDS.get(finger_state, {})

        command = gesture_info.get("cmd", None)

        # Crear gráfico de la mano

        try:

            hand_graph_surf = draw_hand_graph(hand_landmarks)

        except Exception as e:

            print(f"Error al dibujar gráfico de mano: {e}")

            hand_graph_surf = None


        # Control progresivo con gestos especiales

        control_action = GESTURE_CONTROL.get(finger_state, {})
```

Control de dispositivos por gestos articulados.

```
if control_action:  
    if current_gesture != finger_state:  
        current_gesture = finger_state  
        gesture_active_time = time.time()  
  
    # Aplicar acción continua si el gesto se mantiene  
    if time.time() - gesture_active_time > 0.5: # Retardo antes de acción continua  
        if control_action["target"] == "SERVO":  
            step = control_action["step"] * (1 if control_action["action"] ==  
                "INCREASE" else -1)  
            servo_angle = max(0, min(180, servo_angle + step))  
            command = f"DOOR_SET_ANGLE={servo_angle}"  
        elif control_action["target"] == "FAN":  
            step = control_action["step"] * (1 if control_action["action"] ==  
                "INCREASE" else -1)  
            fan_speed = max(0, min(255, fan_speed + step))  
            command = f"FAN_SPEED={fan_speed}"  
        else:  
            current_gesture = None  
  
    # Envío de comandos con deduplicación  
    if command and command != prev_command:  
        if send_command(command):
```

Control de dispositivos por gestos articulados.

```
prev_command = command

# Actualizar estado local inmediatamente para mejor feedback

if command == "LED_ON":

    for device in devices:

        if device["name"] == "Luces":

            device["state"] = "ON"

            device["color"] = GREEN

elif command == "LED_OFF":

    for device in devices:

        if device["name"] == "Luces":

            device["state"] = "OFF"

            device["color"] = RED

elif command == "FAN_ON":

    for device in devices:

        if device["name"] == "Ventilador":

            device["state"] = "255/255"

            device["color"] = GREEN

            device["value"] = 255

elif command == "FAN_OFF":

    for device in devices:

        if device["name"] == "Ventilador":

            device["state"] = "0/255"

            device["color"] = RED
```

Control de dispositivos por gestos articulados.

```
device["value"] = 0

elif command.startswith("FAN_SPEED="):

    speed = int(command.split("=")[1])

    for device in devices:

        if device["name"] == "Ventilador":

            device["state"] = f"{speed}/255"

            device["color"] = GREEN if speed > 0 else RED

            device["value"] = speed

elif command == "FAN_REVERSE":

    for device in devices:

        if device["name"] == "Ventilador":

            device["state"] = "255/255 (R)"

            device["color"] = ORANGE

            device["value"] = 255

elif command.startswith("DOOR_SET_ANGLE"):

    angle = int(command.split("=")[1])

    for device in devices:

        if device["name"] == "Puerta":

            device["state"] = f"{angle}°"

            device["value"] = angle

            servo_angle = angle

# Convertir frame de OpenCV a Pygame
```

Control de dispositivos por gestos articulados.

```
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

frame = np.rot90(frame)

frame = pygame.surfarray.make_surface(frame)

frame = pygame.transform.scale(frame, (screen_width // 2, screen_height - 150))

# Limpiar pantalla

screen.fill(BLACK)

# Dibujar frame de la cámara

screen.blit(frame, (20, 130))

# Dibujar gráfico de la mano si está disponible

if hand_graph_surf:

    screen.blit(hand_graph_surf, (screen_width // 2 + 40, 130))

else:

    no_hand_text = font_large.render("Muestra tu mano a la cámara", True, WHITE)

    screen.blit(no_hand_text, (screen_width // 2 + 100, screen_height // 2))

# Dibujar información de gestos

draw_gesture_info(screen, finger_state, command, servo_angle, fan_speed)

# Dibujar estado de dispositivos

draw_device_status(screen, devices)
```

Control de dispositivos por gestos articulados.

```
# Dibujar ayuda de teclado

help_text = font_small.render("Teclas: 1=LED ON, 2=LED OFF, 3=FAN ON, 4=FAN
OFF, 5=BUZZER, 6=OPEN, 7=CLOSE, 0=ALL OFF, R=REVERSE", True, WHITE)

screen.blit(help_text, (20, screen_height - 30))

# Actualizar pantalla

pygame.display.flip()

clock.tick(30)

# Liberar recursos

hands.close()

cap.release()

pygame.quit()

if arduino:
```

CÓDIGO EN PYTHON

```
#include <Servo.h>

// Pines de los dispositivos

const int LED_PIN = 13;

const int BUZZER_PIN = 11;
```

Control de dispositivos por gestos articulados.

```
const int SERVO_PIN = 10;

// Pines para el puente H L298N (control del ventilador)

const int FAN_IN1 = 8; // Control dirección 1

const int FAN_IN2 = 9; // Control dirección 2

const int FAN_ENA = 6; // Control PWM de velocidad

const int FAN_TACH = 5; // Pin para lectura de RPM (opcional)

// Variables de estado

struct DeviceStatus {

    int fanSpeed = 0;

    bool fanDirection = false; // false = adelante, true = reversa

    int servoAngle = 90;

    bool ledState = false;

    bool buzzerActive = false;

};

DeviceStatus status;

unsigned long lastUpdate = 0;

const unsigned long STATUS_UPDATE_INTERVAL = 1000; // Enviar estado cada 1s

Servo doorServo;
```

Control de dispositivos por gestos articulados.

```
void setup() {  
    Serial.begin(9600);  
  
    // Configurar pines  
    pinMode(LED_PIN, OUTPUT);  
    pinMode(BUZZER_PIN, OUTPUT);  
  
    // Configurar pines del ventilador  
    pinMode(FAN_IN1, OUTPUT);  
    pinMode(FAN_IN2, OUTPUT);  
    pinMode(FAN_ENA, OUTPUT);  
    pinMode(FAN_TACH, INPUT_PULLUP);  
  
    doorServo.attach(SERVO_PIN);  
    doorServo.write(status.servoAngle);  
  
    allOff();  
  
    Serial.println("Sistema_gestual:ready"); // Mensaje de inicio  
}  
  
void loop() {  
    // Procesar comandos seriales
```

Control de dispositivos por gestos articulados.

```
if (Serial.available() > 0) {  
  
    String command = Serial.readStringUntil('\n');  
  
    command.trim();  
  
    processCommand(command);  
  
}  
  
  
  
// Enviar estado periódicamente  
  
if (millis() - lastUpdate > STATUS_UPDATE_INTERVAL) {  
  
    sendStatus();  
  
    lastUpdate = millis();  
  
}  
  
  
  
// Apagar buzzer después de timeout  
  
if (status.buzzerActive && millis() - lastUpdate > 1000) {  
  
    noTone(BUZZER_PIN);  
  
    status.buzzerActive = false;  
  
}  
  
  
  
void processCommand(String command) {  
  
    Serial.print("Comando_recibido:");  
  
    Serial.println(command);
```

Control de dispositivos por gestos articulados.

```
if(command == "ALL_OFF") {  
    allOff();  
}  
  
else if(command == "LED_ON") {  
    digitalWrite(LED_PIN, HIGH);  
  
    status.ledState = true;  
}  
  
else if(command == "LED_OFF") {  
    digitalWrite(LED_PIN, LOW);  
  
    status.ledState = false;  
}  
  
else if(command == "FAN_ON") {  
    setFanSpeed(255, false);  
}  
  
else if(command == "FAN_OFF") {  
    setFanSpeed(0, status.fanDirection);  
}  
  
else if(command == "FAN_REVERSE") {  
    setFanSpeed(255, true);  
}  
  
else if(command.startsWith("FAN_SPEED=")) {  
    int speed = command.substring(10).toInt();  
  
    setFanSpeed(speed, false);
```

Control de dispositivos por gestos articulados.

```
}

else if (command.startsWith("FAN_REVERSE_SPEED=")) {

    int speed = command.substring(18).toInt();

    setFanSpeed(speed, true);

}

else if (command == "BUZZER_ON") {

    tone(BUZZER_PIN, 1000, 1000);

    status.buzzerActive = true;

}

else if (command == "DOOR_OPEN") {

    setServoAngle(180);

}

else if (command == "DOOR_CLOSE") {

    setServoAngle(0);

}

else if (command.startsWith("DOOR_SET_ANGLE=")) {

    int angle = command.substring(15).toInt();

    setServoAngle(angle);

}

else if (command == "GET_STATUS") {

   .sendStatus();

}

}
```

Control de dispositivos por gestos articulados.

```
void allOff() {  
    digitalWrite(LED_PIN, LOW);  
    setFanSpeed(0, status.fanDirection);  
    noTone(BUZZER_PIN);  
    status.ledState = false;  
    status.buzzerActive = false;  
}  
  
void setServoAngle(int angle) {  
    angle = constrain(angle, 0, 180);  
    status.servoAngle = angle;  
    doorServo.write(status.servoAngle);  
    delay(15);  
}  
  
void setFanSpeed(int speed, bool reverse) {  
    speed = constrain(speed, 0, 255);  
    status.fanSpeed = speed;  
    status.fanDirection = reverse;  
  
    if (speed > 0) {  
        digitalWrite(FAN_IN1, reverse ? LOW : HIGH);  
    }  
}
```

Control de dispositivos por gestos articulados.

```
digitalWrite(FAN_IN2, reverse ? HIGH : LOW);

analogWrite(FAN_ENA, speed);

} else {

    digitalWrite(FAN_IN1, LOW);

    digitalWrite(FAN_IN2, LOW);

    analogWrite(FAN_ENA, 0);

}

}

void sendStatus() {

    Serial.print("status:");

    Serial.print("led="); Serial.print(status.ledState ? "on" : "off"); Serial.print(",");

    Serial.print("fan="); Serial.print(status.fanSpeed); Serial.print(",");

    Serial.print("fan_dir="); Serial.print(status.fanDirection ? "reverse" : "forward");

    Serial.print(",");

    Serial.print("door="); Serial.print(status.servoAngle); Serial.print(",");

    Serial.print("buzzer="); Serial.println(status.buzzerActive ? "on" : "off");

}

// Función opcional para lectura de RPM

float readFanRPM() {

    unsigned long pulseCount = 0;

    unsigned long startTime = millis();
```

Control de dispositivos por gestos articulados.

```
while (millis() - startTime < 100) { // Medir durante 100ms  
    if (digitalRead(FAN_TACH)) { // Se corrigió aquí - faltaba el paréntesis de cierre  
        pulseCount++;  
        while(digitalRead(FAN_TACH)); // Esperar a que baje  
    }  
}  
  
// Suponiendo 2 pulsos por revolución (común en ventiladores PC)  
return (pulseCount * 600.0) / 2.0; // Convertir a RPM  
}
```

5.3.4 FUENTES DE CONSULTA

- Arduino Docs. (2025, marzo 28). “Datasheets”. [pdf]. Consultado en: <https://docs.arduino.cc/resources/Datasheets/A000066-Datasheet.pdf>
- Arrow. (2023, abril 19). “Uno: Una visión general de la computadora de placa única más popular de Arduino”. Consultado en: <https://www.arrow.com/es-mx/research-and-events/articles/arduino-uno-product-overview>
- ElectroHogar. (2024, 21 enero). Domótica inclusiva: Soluciones para personas con discapacidad y movilidad reducida. https://electrohogar.info/domotica-y-electrodomesticos-conectados/domotica-inclusiva-soluciones-personas-discapacidad-movilidad-reducida/?utm_source
- Electronic Board (2022, 25 enero). ¿Qué es un servomotor? ¿Cómo funciona? <https://www.electronicboard.es/que-es-un-servomotor-como-funciona/#:~:text=Un%20servomotor%20es%20un%20motor,y%20posici%C3%B3n%20exacta%20al%20servo.>
- FasterCapital. (2025, 19 mayo). "Interacción del usuario Tecnología de reconocimiento de gestos". <https://fastercapital.com/es/contenido/Interaccion-del-usuario--Tecnologia-de-reconocimiento-de-gestos--Mas-alla-de-la-pantalla--Explorando-la-tecnologia-de-reconocimiento-de-gestos.html>
- García Cortés, D. C. (2014). Reconocimiento de gestos de manos como mecanismo de interacción humano-computador [Tesis de maestría, Universidad Nacional de Colombia]. Repositorio UNAL.

Control de dispositivos por gestos articulados.

- Limited, H. G. (2025, 17 marzo). Control de gestos: tecnologías y aplicaciones de reconocimiento de gestos. Hostragons.

<https://www.hostragons.com/es/blog/tecnologias-de-reconocimiento-de-movimiento-por-control-de-gestos/>

- NOU GRUP. (2022, 20 septiembre). ¿Qué tipos de sistemas domóticos hay en la actualidad? NOTICIAS MATERIAL ELECTRICO. NOU GRUP.

https://www.nougrup-aie.com/que-tipos-de-sistemas-domoticos-hay-en-la-actualidad/?utm_source

- Reynoso Martínez, F. O., Abud Figueroa, M. A., & Peláez Camarena, S. G. (2019). Prototipo de interfaz humano-computadora controlada a través de gestos faciales. ReCIBE, 8(1), 1–15. Universidad de Guadalajara.

- Rodríguez Copa, G., Castro, S., & Guerrero, R. (2024). Un módulo integrado de interfaz humano-computadora basada en gestos. Revista Electrónica del Instituto de Tecnología Aplicada de la UNPA, 2(1).

- SER C. (2025, 29 enero). Cadena SER. Cadena SER.

https://cadenaser.com/castillayleon/2025/01/29/habilitan-en-palencia-un-piso-inteligente-para-personas-que-sufren-ictus-radio-palencia/?utm_source

- UNIT Electronics. (2025, 28 julio). Servomotor SG90 RC 9G - UNIT Electronics Arduino Micro Servo. <https://uelectronics.com/producto/servomotor-sg90-rc-9g/?srsltid=AfmBOoqvlp7VoqQnx7mCKPMu6xClkKS9wStlVQgP5r2dvhpOD6CDXa0>

Control de dispositivos por gestos articulados.

- Hardware Libre. (2024, octubre 18). “Buzzer: todo sobre este dispositivo para emitir sonido”. Consultado en: <https://www.hwlible.com/buzzer/>
- OSAKA Electronics. (2024, abril 18). “Todo sobre Buzzers: Cómo Conectarlos y Para Qué Sirven”. Consultado en:
<https://osakaelectronicsltda.com/blog/biblioteca/que-es-un-buzzer#:~:text=%C2%BFQu%C3%A9%20es%20un%20Buzzer?,de%20notificaci%C3%B3n%20pueden%20ser%20ineficas>
- Estévez, A. (2020, 29 enero). Motorreductores: Cómo funcionan y de qué elementos están compuestos -. <https://www.roydisa.es/archivos/5419>
- SANDOROBOTICS. (2025, 14 mayo). 20.4:1 Motorreductor de Metal 25Dx50L mm HP 12V - <https://sandorobotics.com.mx/producto/3203/>
- ClaroShop. (2025). Pila-alcalina-9v-blister-volteck-46319.
<https://www.claroshop.com/producto/17202765/pila-alcalina-9v-blister-volteck-46319>
- Mercado Libre. (2025). “Rollo Cable Estañado Unipolar Calibre #22 Color Rojo 100m”. JPEG. Consultado en: https://articulo.mercadolibre.com.mx/MLM-2249293578-rollo-cable-estanado-unipolar-calibre-22-color-rojo-100m-_JM
- ZW Cables. (2024). “Cable Estañado”. Consultado en:
<https://zwcables.com/es/tinned-cable/#:~:text=El%20cable%20esta%C3%B1ado%20es%20un,resistentes%20a%20la%20corrosi%C3%B3n%20atmosf%C3%A9rica>

Control de dispositivos por gestos articulados.

- Visual Led. (2022, 8 noviembre). ¿Qué es un led? Tipos de Led, aplicaciones y usos. [https://visualled.com/glosario/que-es-un-led/#:~:text=Un%20LED%20\(acr%C3%B3nimo%20del%20concepto,que%20se%20conoce%20como%20electroluminiscencia](https://visualled.com/glosario/que-es-un-led/#:~:text=Un%20LED%20(acr%C3%B3nimo%20del%20concepto,que%20se%20conoce%20como%20electroluminiscencia).