



**UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**

**“Sistema de detección y seguimiento de la posición de personas para  
ponentes en la UNPRG”**

**TESIS**

**Para optar el título profesional de  
INGENIERO ELECTRÓNICO**

**Autor:**

Bach. Acedo Ten, Juan Pablo

**Asesor:**

Mg. Ing. Romero Cortez, Oscar Uchelly

ORCID: 0000-0002-7727-7900

**Lambayeque, 2023**

**UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**

**“Sistema de detección y seguimiento de la posición de personas para  
ponentes en la UNPRG”**

**TESIS**

**Para optar el título profesional de  
INGENIERO ELECTRÓNICO**

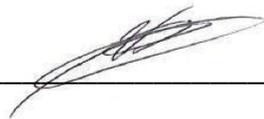
**Elaborada por:**



---

Bach. Acedo Ten, Juan Pablo

**Asesor:**



---

Mg. Ing. Romero Cortez, Oscar Uchelly

ORCID: 0000-0002-7727-7900

**Lambayeque, 2023**

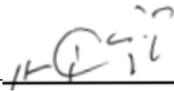
**UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**

**“Sistema de detección y seguimiento de la posición de personas para  
ponentes en la UNPRG”**

**TESIS**

**Para optar el título profesional de  
INGENIERO ELECTRÓNICO**

Aprobada por el siguiente jurado:



---

M. Sc. Ing. CHICLAYO PADILLA, HUGO JAVIER  
PRESIDENTE



---

Mg. Ing. NOMBERA LOSSIO, MARTÍN AUGUSTO  
SECRETARIO



---

Dra. Ing. CHAMÁN CABRERA, LUCIA ISABEL  
VOCAL

**Lambayeque, 2023**

## DECLARACIÓN JURADA DE ORIGINALIDAD

Yo, ACEDO TEN JUAN PABLO, principal investigador del presente proyecto de tesis, en conjunto al MG. ING. OSCAR UCHELLY ROMERO CORTEZ, asesor del trabajo de investigación denominado “SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE PERSONAS PARA PONENTES EN LA UNPRG”, declaramos bajo juramento que el presente trabajo no ha sido plagiado ni contiene datos falsos. En caso se demostrará lo contrario, asumimos responsablemente la anulación de este informe y, por ende, el proceso administrativo correspondiente, pudiendo conducir en la anulación del título o grado emitido como consecuencia de este informe.

Lambayeque, mayo del 2022

AUTOR:



---

BACH. ACEDO TEN JUAN PABLO

ASESOR:



---

MG. ING. OSCAR UCHELLY ROMERO CORTEZ

## **DEDICATORIA**

A mis padres Juan Pablo Acedo Rivas y Mery Elvira Ten Mariños, por su apoyo brindado a pesar de las limitaciones que pudieran existir, siempre confiando en mí, criándome en valores y mostrando siempre su afecto.

A mis hermanas que Yuyin Acedo Ten, Xiany Acedo Ten y Yuxiamy Naomi Acedo Ten, por su confianza y que a pesar de todo permanecemos unidos.

## **AGRADECIMIENTOS**

Agradezco profundamente a mis padres que me permitieron desempeñar mis estudios, por cuidarme desde pequeño, por su guía y esfuerzo constate durante todos estos años.

A los docentes que durante mi recorrido como alumno me acompañaron, compartiendo conocimientos y evocando el sentido de la investigación.

## RESUMEN

El presente proyecto de tesis presenta como objetivo el diseño e implementación del prototipo de un sistema de detección y seguimiento de la posición de personas para ponentes en la UNPRG, el cual controlará una cámara, mientras se realiza la transmisión o grabación de la charla.

Para la etapa del procesamiento de imágenes, se hizo uso de la librería OpenCV y el lenguaje Python, ya que presentan una comunidad de soporte amplia y la robustez suficiente para el desarrollo de proyectos de esta índole; además, es fácil el integrar otras librerías como las de protocolos de comunicación, útil para comunicar nuestro software con el circuito electrónico.

El programa determina la existencia y posición del sujeto en la imagen, este dato es enviado al circuito electrónico, el cual, por medio de motores, produce el movimiento de la cámara, teniendo como fin el encuadre adecuado de la persona. El circuito cuenta con una etapa de potencia para activar los motores y una etapa de control que procesa los datos recibidos y controla el movimiento de los motores.

Finalmente, se calculó de los gastos aproximados de la implementación del proyecto.

***Palabras clave:*** *Procesamiento de imágenes, Seguimiento, Algoritmo de control.*

## **ABSTRACT**

The objective of this thesis project is to design and implement the prototype of a system for detecting and monitoring the position of people for speakers at the UNPRG, which will control a camera, while the transmission or recording of the talk is being carried out.

For the image processing stage, the OpenCV library and the Python language were used, since they present a wide support community and sufficient robustness for the development of projects of this nature; In addition, it is easy to integrate other libraries such as communication protocols, useful for communicating our software with the electronic circuit.

The program determines the existence and position of the subject in the image, this data is sent to the electronic circuit, which, by means of motors, produces the movement of the camera, aiming at the proper framing of the person. The circuit has a power stage to activate the motors and a control stage that processes the received data and controls the movement of the motors.

Finally, the approximate costs of implementing the project were calculated.

***Keywords:*** *Image processing, Tracking, Control algorithm.*

# ÍNDICE GENERAL

<b>DEDICATORIA .....</b>	<b>viii</b>
<b>AGRADECIMIENTOS.....</b>	<b>ix</b>
<b>RESUMEN .....</b>	<b>X</b>
<b>ABSTRACT .....</b>	<b>xi</b>
<b>ÍNDICE GENERAL .....</b>	<b>xii</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPITULO I. DISEÑO TEÓRICO .....</b>	<b>4</b>
1.1. Planteamiento de la investigación científica .....	5
1.1.1. Situación Problemática .....	5
1.1.2. Antecedentes .....	5
1.1.3. Formulación del problema científico .....	7
1.2. Bases Teóricas .....	7
1.2.1. Algoritmo.....	7
1.2.2. Software .....	8
1.2.3. Python .....	8
1.2.4. Imagen Digital.....	8
1.2.5. Espacio RGB.....	9
1.2.6. Histograma de una imagen.....	9
1.2.7. Visión por computadora .....	10
1.2.8. Detección de objetos .....	10
1.2.9. Seguimiento.....	11
1.2.10. MOSSE tracking .....	11
1.2.11. Convolución de imágenes.....	11
1.2.12. Haar feature-based cascade classifiers.....	12
1.2.13. OpenCV.....	13
1.2.14. Protocolo de comunicación .....	13

1.2.15.	USART/SCI.....	13
1.2.16.	Circuitos Electrónicos .....	13
1.2.17.	Circuito de control .....	14
1.2.18.	Microcontrolador .....	14
1.2.19.	Arduino.....	15
1.2.20.	Circuito de potencia.....	15
1.2.21.	Puente H.....	15
1.2.22.	Actuador .....	16
1.2.23.	Motor eléctrico.....	16
1.2.24.	PWM (MODULACIÓN POR ANCHO DE PULSOS) .....	16
<b>CAPITULO II. MÉTODOS Y MATERIALES.....</b>		<b>17</b>
2.1.	Consideraciones para el Diseño.....	18
2.1.1.	Consideraciones para componentes.....	18
2.2.	Selección de componentes.....	19
2.2.1.	Teléfono celular con cámara .....	19
2.2.2.	Interfaz de Usuario .....	19
2.2.3.	Circuito Electrónico .....	20
2.2.4.	Driver de Puente H .....	21
2.2.5.	Motor DC.....	22
2.2.6.	Fuente de alimentación .....	23
2.2.7.	Módulo Bluetooth .....	24
2.2.8.	Sensor infrarrojo.....	25
2.2.9.	Estructura plástica de dispositivo .....	26
2.3.	Diagrama de flujo del algoritmo de control general.....	26
2.4.	Diseño e Implementación.....	28
2.4.1.	Captura de Video y transmisión a PC.....	28
2.4.2.	Software de procesamiento de video.....	31
2.4.3.	Software Arduino Nano .....	33

2.4.4.	Diseño del almacén en FreeCad .....	34
2.4.5.	Emparejamiento con el módulo Bluetooth.....	37
2.4.6.	Implementación física.....	40
<b>CAPITULO III. RESULTADOS Y DISCUSIÓN .....</b>		<b>43</b>
3.1.	Software implementado.....	44
3.1.1.	Comunicación .....	44
3.1.2.	Detección y Seguimiento .....	44
3.1.3.	FPS .....	45
3.1.4.	Limitantes .....	46
3.2.	Circuito de Control .....	46
3.2.1.	Tiempo de ejecución del movimiento .....	46
3.2.2.	Ángulo y velocidad de desplazamiento .....	48
3.2.3.	Consumo de energía.....	48
3.3.	Presupuesto.....	50
3.3.1.	Presupuesto del equipo .....	50
3.3.2.	Presupuesto de elementos complementarios .....	50
<b>CAPITULO IV. CONCLUSIONES.....</b>		<b>51</b>
<b>CONCLUSIONES .....</b>		<b>52</b>
<b>CAPITULO V. RECOMENDACIONES .....</b>		<b>53</b>
<b>RECOMENDACIONES .....</b>		<b>54</b>
<b>BIBLIOGRAFÍA .....</b>		<b>55</b>
<b>ANEXOS.....</b>		<b>59</b>
	ANEXO 1. CÓDIGO DEL SOFTWARE EN PC.....	60
	ANEXO 2. CÓDIGO DEL ARDUINO NANO .....	62
	ANEXO 3. ESPECIFICACIONES TÉCNICAS.....	66

## LISTA DE FIGURAS

Ilustración 1	Píxeles de una imagen .....	9
Ilustración 2	Histograma de la imagen de la izquierda .....	10
Ilustración 3	Convolución con una máscara .....	12
Ilustración 4	Características de Haar .....	12
Ilustración 5	Un circuito electrónico muy simple: pila, interruptor y bombilla. ....	14
Ilustración 6	Puente H .....	15
Ilustración 7	PWM con diferentes ciclos .....	16
Ilustración 8	Conexión General del Sistema .....	18
Ilustración 9	Arduino Nano .....	20
Ilustración 10	Pinout Arduino Nano .....	21
Ilustración 11	Módulo L298N pinout.....	22
Ilustración 12	Motor DC .....	23
Ilustración 13	Módulo HC-06.....	25
Ilustración 14	Pinout del módulo HC-06.....	25
Ilustración 15	Sensor infrarrojo.....	26
Ilustración 16	Diagrama de flujo del proceso .....	27
Ilustración 17	DroidCam en PlayStore .....	28
Ilustración 18	Página oficial de DroidCam .....	29
Ilustración 19	Instalación de DroidCam en PC.....	29
Ilustración 20	Pantalla de inicio de DroidCam en celular .....	30
Ilustración 21	Selección de cámara en DroidCam .....	30
Ilustración 22	Ventana de inicio DroidCAM en PC.....	31
Ilustración 23	Imagen mostrada en el teléfono y PC.....	31
Ilustración 24	Diagrama de flujo del software de reconocimiento y seguimiento .....	33
Ilustración 25	Diagrama de Flujo para el microcontrolador .....	34
Ilustración 26	Base para el motor en contacto con el piso.....	35
Ilustración 27	Pieza de unión entre los motores .....	35
Ilustración 28	Diseño de eje que soporta al teléfono.....	36
Ilustración 29	Vista del modelo general .....	36
Ilustración 30	Conexión Módulo Bluetooth.....	37
Ilustración 31	Activación de Bluetooth en PC.....	38
Ilustración 32	Añadir nuevo dispositivo Bluetooth en PC.....	38
Ilustración 33	Contraseña para vinculación del HC-06 .....	39
Ilustración 34	Verificación del puerto serie asociado al HC-06 .....	40
Ilustración 35	Conexión de los elementos eléctricos .....	40
Ilustración 36	conexión de elementos eléctricos .....	41
Ilustración 37	Estructura interna.....	42
Ilustración 38	Equipo implementado .....	42
Ilustración 39	Inicio del software .....	44
Ilustración 40	Software detectando .....	44
Ilustración 41	Software siguiendo al individuo.....	45
Ilustración 42	Software buscando objetivo .....	45

Ilustración 43 Fallo en la detección .....	46
Ilustración 44 Inicio del movimiento .....	47
Ilustración 45 Punto máximo de movimiento con cámara estática.....	47
Ilustración 46 Objetivo centrado nuevamente.....	48
Ilustración 47 Corriente con sistema en reposo .....	49
Ilustración 48 Corriente con sistema en movimiento.....	49

## LISTA DE TABLAS

Tabla 1 Características Arduino Nano .....	20
Tabla 2 Características Módulo L298N .....	21
Tabla 3 Descripción del pinout del módulo L298N .....	22
Tabla 4 Características motor DC seleccionado.....	23
Tabla 5 Características Módulo HC-06 .....	25

## **INTRODUCCIÓN**

El desarrollo de nuevas tecnologías ha permitido facilitar el trabajo de las personas, pues se crean herramientas que le permiten optimizar tiempos e incluso desatenderse de realizar alguna labor. Esto se ha visto en aumento con el progreso en el campo de la electrónica y las ciencias en computación, tales como robótica y machine learning.

### **I. SÍNTESIS DE LA SITUACIÓN PROBLEMÁTICA**

Existe un gran número de casos donde una persona trabaja frente a una cámara estacionaria; sin embargo, existen ocasiones donde el expositor se debe trasladar, mas al no tener apoyo de otro individuo, debe acercarse a la cámara con el fin de cambiar su dirección, ocasionando molestias y pérdidas de tiempo.

El presente proyecto tiene como finalidad, elaborar un sistema que permita al ponente desplazarse alrededor de una cámara, sin que esta lo pierda de la escena.

El sistema consta de dos partes principales. La primera, el software, que se encarga de la detección de la persona por medio de las imágenes procedentes de la cámara, así como también establece los movimientos que deberán realizarse según el desplazamiento del individuo. La segunda parte, corresponde al circuito electrónico, el cual controla los motores según la información que reciba del software.

### **II. FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN.**

¿Cómo desarrollar un sistema de detección y seguimiento de la posición de personas para ponentes en la UNPRG?

### **III. JUSTIFICACIÓN E IMPORTANCIA**

El poder apoyar a las personas por medio de la tecnología, representa un avance para la sociedad que permite liberar la carga de trabajo al usuario, dando lugar a que pueda centrarse en otras actividades, así mismo como es en este caso, evita el depender de terceros representándoles un ahorro en algunos casos.

Cuando un expositor u docente desea dar su charla, no siempre cuenta con el apoyo de alguien en cámara o les dificulta conseguir uno; con este proyecto pueden liberarse de esa preocupación y centrarse en otras actividades.

### **IV. HIPÓTESIS**

El diseño consiste en:

Implementar un programa capaz de identificar a una persona en una escena captada por una cámara.

Según la posición del individuo con respecto a la escena, calcular la dirección y distancia necesaria para ubicarlo en el punto deseado.

Un circuito electrónico activará los motores de acuerdo al cálculo del programa; luego, se actualizarán los datos e imagen, verificando que la nueva orientación de la cámara es la adecuada.

Al implementar un sistema de detección y seguimiento de la posición de personas para ponentes en la UNPRG, se podrá ayudar al expositor a realizar desplazamientos de manera continua, retirando la necesidad de tener un asistente en cámara o los inconvenientes en caso no pueda disponer de uno.

### **V. OBJETIVOS**

#### **A. Objetivo general**

Diseñar un sistema de detección y seguimiento de la posición de personas para ponentes en la UNPRG.

## **B. Objetivos específicos**

- Aplicar el procesamiento de imágenes para identificar a una persona utilizando visión por computadora.
- Calcular los movimientos que se deben realizar sobre la cámara según la posición de la persona.
- Diseñar un circuito capaz de controlar la posición de los motores.
- Integrar software y hardware por medio de protocolos de comunicación.
- Realizar pruebas de funcionamiento.

## **CAPITULO I. DISEÑO TEÓRICO**

## **1.1. Planteamiento de la investigación científica**

### **1.1.1. Situación Problemática**

Existe un gran número de casos donde una persona trabaja frente a una cámara estacionaria; sin embargo, existen ocasiones donde el expositor se debe trasladar, mas al no tener apoyo de otro individuo, debe acercarse a la cámara con el fin de cambiar su dirección, ocasionando molestias y pérdidas de tiempo.

El presente proyecto tiene como finalidad, elaborar un sistema que permita al ponente desplazarse alrededor de una cámara, sin que esta lo pierda de la escena.

El sistema consta de dos partes principales. La primera, el software, que se encarga de la detección de la persona por medio de las imágenes procedentes de la cámara, así como también establece los movimientos que deberán realizarse según el desplazamiento del individuo. La segunda parte, corresponde al circuito electrónico, el cual controla los motores según la información que reciba del software.

### **1.1.2. Antecedentes**

#### **A. Internacionales**

**“Detección, seguimiento y análisis de personas en streams de vídeo”** (Bolaños, 2013)

El trabajo consistió en una aplicación capaz de detectar de personas de cuerpo entero. Se aplicó un detector a las imágenes, el cual está escrito en el lenguaje de C++. Se utilizó herramientas como OpenCV, aplicando el método de histograma de gradientes orientados, así mismo, la librería ENCARA, también posee un detector facial, lo que sirve de comprobación. Por persona encontrada los datos se almacenan en una lista, luego se comparan con los obtenidos anteriormente, e inicia nuevamente el proceso. Visualmente cada persona es enmarcada con un cuadro de diferente color. Junto a lo anterior se guarda la hora y frame de aparición y salida del individuo.

### **“Seguimiento autónomo de una persona con marcador visualmente identificada por un dron” (Rodríguez, 2018)**

“En este proyecto se diseña y desarrolla un dron capaz de seguir a una persona con un marcador visual.” (Rodríguez, 2018, pág. 2) El dron o cuadricóptero, posee una cámara que permitirá seguir al individuo.

El software se encarga de detectar el color que el usuario defina y además de seguirlo. Además, el software controla los movimientos y altitud del dron, según lo asignado.

## **B. Nacionales**

### **“Histograma de orientación de gradientes aplicado al seguimiento múltiple de personas basado en video” (Tolentino, 2017)**

En la tesis, para el seguimiento múltiple de individuos utilizando histograma de orientación de gradientes se tomaron las siguientes etapas: Procesamiento de Video, aquí se obtienen los frames que forma el video, para ello utilizaron la librería OpenCV; la siguiente parte fue la Clasificación de Candidatos, aquí hacen uso del algoritmo de HOG para la detección de personas y así mismo la elección de candidatos; finalmente, la etapa de Seguimiento y Asociación, donde el algoritmo de Kalman Filter, permitió establecer las relaciones entre las secuencias de objetos anteriormente identificados. (Tolentino, 2017, pág. 2)

### **“Diseño de un algoritmo de estabilización de video orientado a la detección de personas.” (Inafuku, 2015)**

El proyecto tenía como finalidad el poder crear un algoritmo de estabilización de video eficaz y robusto cara a permutaciones en la escala, iluminación y rotación. La etapa de estabilización es importante, ya que es un pre-procesado del video, ayudando a suprimir o reducir el ruido producido por los movimientos involuntarios, lo que ayuda en las etapas de procesamiento de imágenes posteriores, puesto que necesitan datos más precisos, libres de distorsión. (Inafuku, 2015, pág. 1)

## C. Locales

**“Sistema de visión artificial para apoyar en la identificación de plagas y enfermedades del cultivo de sandía en el distrito de Ferreñafe”** (Piscoya, 2019)

Este proyecto busca ayudar a los agricultores de sandías de la ciudad de Ferreñafe, a poder identificar las plagas que ataquen sus cultivos y además indicarles el agroquímico correcto. Para ello compara la imagen tomada con un repositorio, donde identifica los patrones propios de cada clasificador y retorna un resultado. (Piscoya, 2019, pág. 1)

**“Visión artificial aplicada al monitoreo automatizado del proceso de cloración para mejorar la calidad del agua”** (Palacín, 2011)

El proyecto trata de solventar los problemas existentes como, el índice de error, la demora o grandes costos del proceso de monitorización. Debido a ello se propuso el desarrollo de un sistema de visión artificial procesado por una red neuronal, para las mediciones de nivel de cloro durante el proceso de cloración. (Palacín, 2011, pág. 8)

### 1.1.3. Formulación del problema científico

¿Cómo desarrollar un sistema de detección y seguimiento de la posición de personas para ponentes en la UNPRG?

## 1.2. Bases Teóricas

### 1.2.1. Algoritmo

De manera general hace referencia a un conjunto determinado de procesos los cuales a su vez son finitos y estar adecuadamente definidos, los cuales llevan a un resultado. (Martínez & Martín, 2003, pág. 41). En otras palabras, algoritmo nos permite poder definir los pasos a seguir para la ejecución de una tarea, el cual podremos reproducir las veces que sea necesario, siempre que se presenten las condiciones para su funcionamiento.

### **1.2.2. Software**

“Término genérico que se aplica a los componentes no físicos de un sistema informático, como p. ej. los programas, sistemas operativos, etc., que permiten a este ejecutar sus tareas.” (Wordreference, s.f., pág. definición 1).

En este se implementarán los algoritmos establecidos para el procesamiento de imágenes que permitan proceder con la identificación y rastreo del elemento establecido, para este caso una persona; para ello se hace uso de un lenguaje de programación, el cual permite que crear las órdenes que la computadora interpretará y ejecutará.

### **1.2.3. Python**

Es un lenguaje de programación de interpretado, el cual, ha tomado gran popularidad en el campo del software libre, resaltando en áreas como el procesamiento de datos, machine learning, inteligencia artificial, entre otros.

Este lenguaje posee una amplia variedad de librerías para diferentes trabajos, entre ellos el de procesamiento de señales, operaciones matemáticas, diseño de interfaces, manipulación de imágenes, entre otros; así también, una comunidad sólida que constantemente desarrolla nuevas herramientas.

### **1.2.4. Imagen Digital**

“Es la representación bidimensional de una imagen empleando bits, unidad mínima de información compuesta por dígitos binarios (1 y 0), que son los que regulan los sistemas informáticos y cualquier dispositivo de tipo digital.” (Gómez, 2017)

La unidad mínima de una imagen digital es un punto el cual recibe por nombre “pixel”, este posee un valor el cual representa la intensidad del color asignado; además, es utilizado para definir la resolución de la imagen (total de pixeles), es decir mientras mayor número de pixeles posea una imagen, mayor resolución tendrá; lo que permite ofrecer una mejor calidad de la imagen.

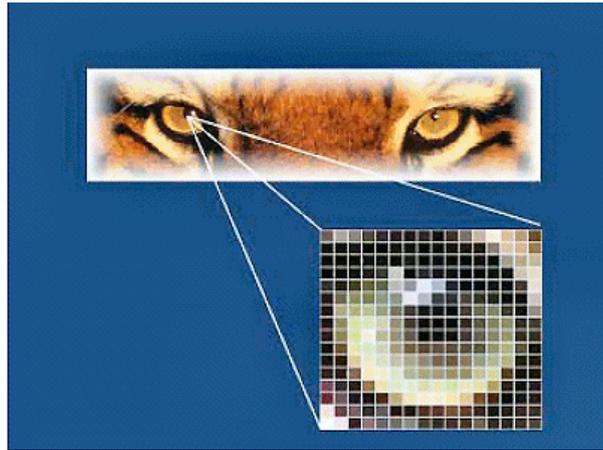


Ilustración 1 Pixeles de una imagen  
(Miranda, 2009)

### 1.2.5. Espacio RGB

Este concepto está basado en la combinación de la luminancia cromática entre los colores rojo, verde y azul; la cual menciona que se puede obtener un color deseado asignando una intensidad a cada uno de estos colores primarios. (Esqueda & Palafox, 2005, pág. 13)

Una imagen digital a color básicamente se puede representar como la unión de tres imágenes, las cuales están determinadas por la intensidad de cada color; es decir una imagen donde solo están las intensidades de rojo, otra de azul y la última de verde; cabe decir que actualmente existen muchos otros formatos para representar imágenes, por ejemplo algunos tienen una capa adicional que representa si el píxel es transparente o no, sin embargo este modelo es el más usado y es el referente para muchos otros.

### 1.2.6. Histograma de una imagen

“Un histograma representa la distribución de las intensidades de los píxeles (color o escala de grises) en una imagen” (Taquía, 2017, pág. 19)

Este al ser una representación gráfica, nos permite poder observar la intensidad que poseen los píxeles según la capa de color que se encuentren, este gráfico puede ser una curva, barras u otro que puede expresar los valores.

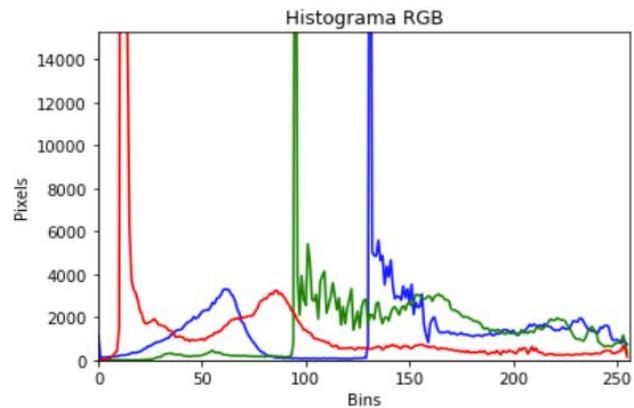


Ilustración 2 Histograma de la imagen de la izquierda  
(Taquiá, 2017, pág. 19)

Taquia (2017) nos explica sobre la Ilustración 2 como en el histograma se puede observar los picos verde y azul en el centro del gráfico puesto que son los colores más resaltantes de la imagen, en contraposición el rojo posee mayor presencia en la parte más oscura de la imagen (lo que sería la sombra del árbol), por ello sus valores son más cercanos a cero. (pág. 19)

### 1.2.7. Visión por computadora

“La visión por computador, también conocida como visión artificial o interpretación de imágenes, es una disciplina que persigue la deducción automática de la estructura y propiedad de un escenario o mundo tridimensional, posiblemente cambiante, a partir de una o varias imágenes captadas del mismo.” (Somolinos, 2002, págs. 29-30)

### 1.2.8. Detección de objetos

La detección de objetos es uno de los resultados de aplicar técnicas de procesamiento de imágenes y algoritmos especializados, a una imagen donde está presente un elemento del que se desea obtener principalmente su posición y tamaño con respecto a la imagen; en otras palabras, el ser capaz de poder identificarlo de manera automática.

### 1.2.9. Seguimiento

Mayormente conocido como “tracking” (palabra en inglés), hace referencia a la capacidad del algoritmo de poder identificar el movimiento que realiza un elemento designado. Se diferencia de la detección ya que no busca encontrar un elemento como tal, sino que se basa en poder darle seguimiento a un área asignada, corresponda a un objeto, persona, color u otro.

### 1.2.10. MOSSE tracking

“El MOSSE filter es un filtro de correlación estable que se puede inicializar en un solo cuadro de un video. El filtro MOSSE se adapta a los cambios en la apariencia del objeto durante el seguimiento.” (Kaur, 2016, pág. ii)

El filtro MOSSE es uno de los más rápidos, aunque sacrifica un poco la precisión para lograr esto.

### 1.2.11. Convolución de imágenes

La convolución es una operación entre dos funciones, obteniendo otra función como resultado. Para el caso de las imágenes, implica la extensión del caso unidimensional, donde una señal cualquiera se procesa con un filtro arbitrario, el cual tiene una respuesta conocida. (Passariello & Mora, 1995, pág. 75)

Considerando  $x(i,j)$  como la matriz de una imagen y  $h(i,j)$  como la matriz de convolución (también llamada núcleo o filtro de convolución), la convolución de estos obtiene una respuesta impulsiva una imagen  $y(i,j)$ , determina la siguiente ecuación, donde  $m,n$  determinan la vecindad según el tamaño del núcleo de convolución  $h(i,j)$

$$y(i,j) = \sum_{m=K_1}^{K_2} \sum_{n=L_1}^{L_2} h(m,n)x(i-m,j-n) \quad \text{Ecuación 1}$$

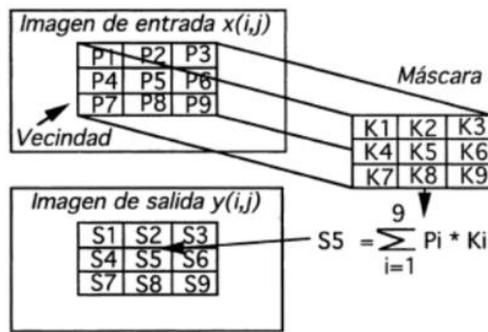


Ilustración 3 Convolución con una máscara  
(Passariello & Mora, 1995, pág. 76)

### 1.2.12. Haar feature-based cascade classifiers

Es un enfoque basado en el aprendizaje automático propuesto por Paul Viola y Michael Jones, en el cual una función en cascada se entrena en base a una gran cantidad de imágenes positivas y negativas. Para el caso de la detección de rostros, el algoritmo necesita primero, una gran cantidad de imágenes positivas (imágenes de rostros) e imágenes negativas (imágenes sin rostros), las cuales se utilizan para entrenar al clasificador. Luego, se procede a extraer las características de cada una utilizando las características de Haar como muestra la Ilustración 4; **Error! No se encuentra el origen de la referencia.**; lo que vendría a ser el núcleo convolucional. Cada entidad es un valor único obtenido restando la suma de píxeles debajo del rectángulo blanco de la suma de píxeles debajo del rectángulo negro. (OpenCV, 2021)

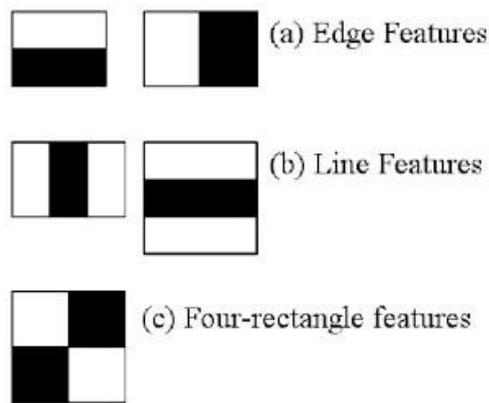


Ilustración 4 Características de Haar  
(OpenCV, 2021)

### 1.2.13. OpenCV

“OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión artificial y aprendizaje automático de código abierto. Se creó para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales.” (OpenCV, 2020)

### 1.2.14. Protocolo de comunicación

Los protocolos de comunicación son normas que existen con el fin de poder comunicar un mensaje entre dos o más entidades. Cada protocolo tendrá características que lo especialicen o sea preferido para una aplicación. En el campo tecnológico, para compartir información entre equipos, existen diferentes protocolos, tales como el Serial, UTP, TCP, I2C, IP, HTTP, etc.

### 1.2.15. USART/SCI

Siglas del inglés Universal Synchronous Asynchronous Receiver Transmitter/Serial Communication Interface. Esta interfaz tiene dos modos de comunicación: asíncrona y síncrona. La primera es full duplex con una vía RX (Recepción) y otra TX (Transmisión), la segunda interfaz es half duplex, utiliza una vía de CLK (reloj) y otra de DT (Transmisión de datos), con sus configuraciones por dispositivo como maestro o esclavo. En los

microcontroladores también pueden existir otros módulos de comunicación serie como el SPI (Serial Peripheral Interface) o I2C (Inter. -Integrated Circuit) que son de tipo síncrono. (Herrera, Tlízaliturri, & Morales, 2006, pág. 3)

### 1.2.16. Circuitos Electrónicos

La electrónica se encarga de estudiar la conducta de los electrones en diferentes medios, con el fin de aplicar estos conocimientos y así conseguir que “los electrones hagan lo que nosotros queramos”. Así, por ejemplo, un circuito electrónico elaborado por una bombilla, una pila y un interruptor (Ilustración 5) interconectados, se logrará que los electrones circulen a través de este y que al pasar por la bombilla parte de ellos se transformen en luz. (González, 2014, pág. 5)

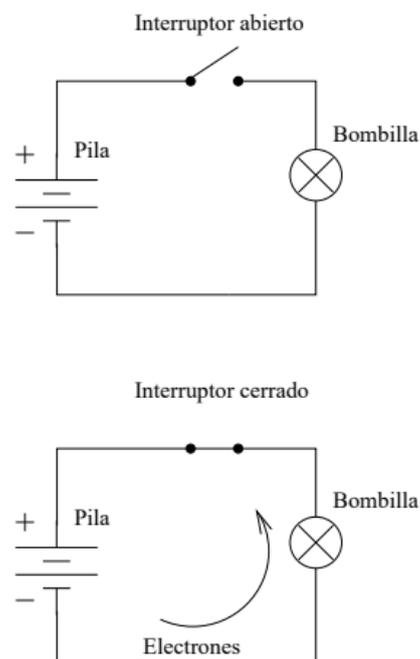


Ilustración 5 Un circuito electrónico muy simple: pila, interruptor y bombilla. (González, 2014, pág. 5)

### 1.2.17. Circuito de control

Es la etapa donde, de acuerdo a los parámetros establecidos o al programa aplicado, obtiene datos de diferentes fuentes y genera órdenes de salida, las cuales producen respuestas en los actuadores.

#### **1.2.18. Microcontrolador**

“Un microcontrolador es un microcomputador realizado en un circuito integrado (chip). Están concebidos fundamentalmente para ser utilizados en aplicaciones puntuales” (Valdes & Pallás, 2007, pág. 12). Suele ser el encargado de procesar las órdenes dentro de un circuito de control.

Al ser programables se pueden implementar en diferentes aplicaciones, pudiendo ser actualizables en el tiempo, según el diseño del circuito.

#### **1.2.19. Arduino**

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino son capaces de leer entradas, como la de un sensor, botón, o un mensaje de internet, para luego convertirlo en una salida ya sea la activación un motor, encendido de un LED o la publicación de datos en internet; además, las placas más comunes están basadas en microcontroladores de la marca ATMEL. Para programar las instrucciones utilizan el lenguaje de programación Arduino (basado en Wiring) y el software interface Arduino IDE (basado en Processing). (Arduino, 2018)

#### **1.2.20. Circuito de potencia**

Es circuito es el encargado de poder activar de manera efectiva los elementos que necesitan un mayor consumo de corriente; esto, puesto que la etapa de control no suele manejar una salida de corriente suficiente para los actuadores, tales como motores, relés, etc.

Este circuito posee la entrada de bajo nivel, la cual activa los componentes que permiten o niegan el paso de la fuente de energía del actuador.

#### **1.2.21. Puente H**

Es un circuito que permite manipular la polaridad del voltaje aplicado a una carga, por lo general un motor. Este suele ser una composición de transistores y diodos, mas también existen encapsulados.

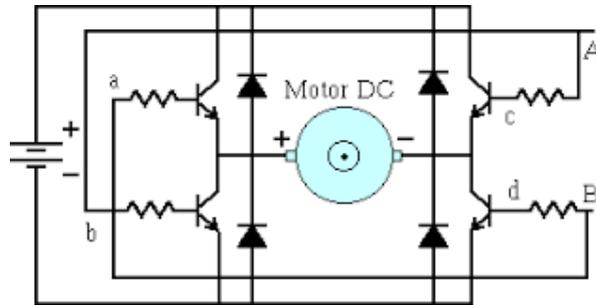


Ilustración 6 Puente H  
(Tecno&Logía, 2021)

### 1.2.22. Actuador

“Es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). Dependiendo de el origen de la fuerza el actuador se denomina neumático, hidráulico o eléctrico”. (Vildósola, 2021, pág. 1)

### 1.2.23. Motor eléctrico

“Es un elemento que, al ser alimentado mediante una corriente eléctrica, produce un movimiento giratorio de su eje” (Martín & García, 2016, pág. 106). Este, como actuador, vendría a ser el elemento final o de salida a controlar, en este caso el que genera el movimiento.

### 1.2.24. PWM (MODULACIÓN POR ANCHO DE PULSOS)

“En la modulación por ancho de pulso, la frecuencia se controla con pulsos positivos durante medio período y pulsos negativos durante el siguiente medio período. El circuito de control establece los tiempos de activación y

desactivación de los dispositivos de potencia” (Díaz, García, & Ríos, 2009, pág. 58)

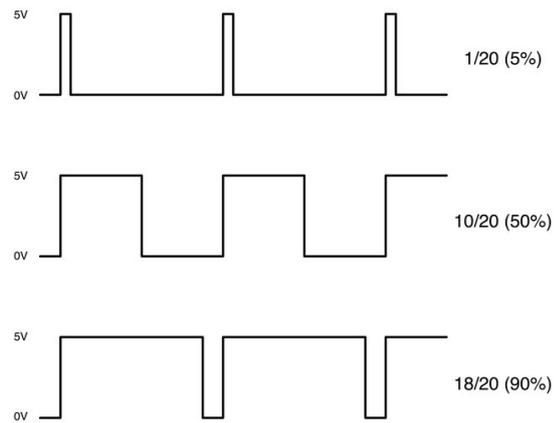


Ilustración 7 PWM con diferentes ciclos  
(Programo Ergo Sum, s.f.)

## **CAPITULO II. MÉTODOS Y MATERIALES**

## 2.1. Consideraciones para el Diseño

Para el desarrollo del proyecto se plantea dividir el sistema en dos partes principales. La primera parte consiste en desarrollar un software capaz de detectar a la persona en una imagen, determinar su posición relativa en la imagen y poder comunicar este dato al circuito. La segunda parte, es la del circuito electrónico que controla al motor y será capaz de mover a la cámara anclada a este.

De manera general las principales partes involucradas y su conexión es la siguiente:

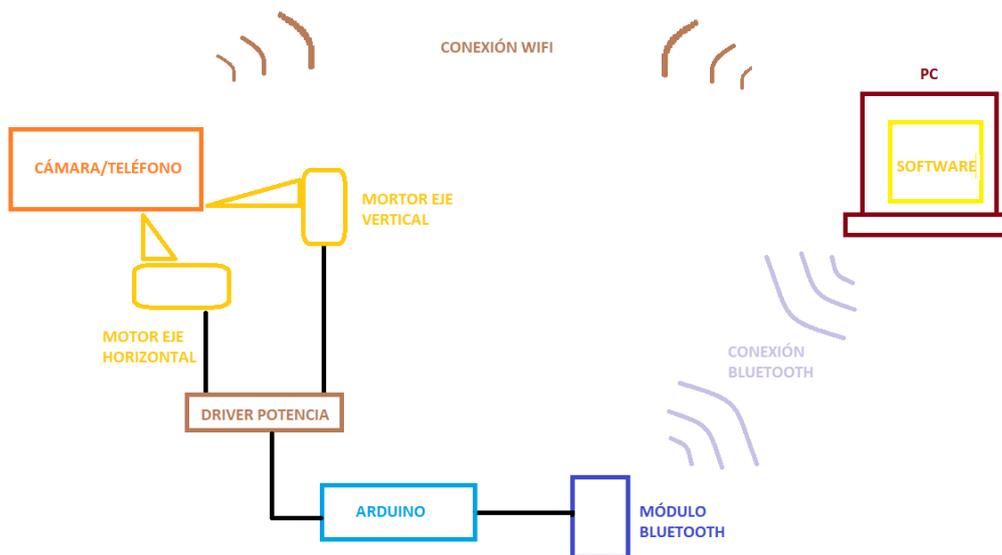


Ilustración 8 Conexión General del Sistema  
(fuente propia)

### 2.1.1. Consideraciones para componentes

- Cámara del teléfono celular: Para este proyecto se utilizará la cámara de un teléfono celular, puesto que es un elemento del cual la mayoría dispone; este deberá tener una aplicación de transmisión de video vía wifi o similar.
- Software de Reconocimiento e Interfaz Gráfica: Este elemento será el mismo ya que se puede desarrollar como un solo ente en la programación; debe poder recibir y procesar la imagen que captura la cámara, también la mostrarla al usuario indicando al sujeto detectado.

- **Circuito Electrónico:** Se encargará de mandar la señal de control al driver de potencia para el movimiento de los motores, para el control de la velocidad se utilizará PWM; así mismo se conectará a la PC, de la que recibirá los datos sobre la dirección que debe seguir el movimiento y procesará el error.
- **Drivers de potencia:** Un puente H capaz de entregar la corriente necesaria para el movimiento del motor de al menos 0.5A, esto porque el microcontrolador directamente no genera la corriente suficiente en sus pines.
- **Motores:** Encargados del movimiento de la cámara, la cual deberá ser anexada a estos por medio de un adaptador. Se utilizarán dos motores para mover la cámara tanto en el eje vertical como horizontal.
- **Fuente de voltaje:** Servirá para alimentar a los motores y el circuito electrónico, está será de 5V ya que es el estándar de trabajo de los elementos a utilizar.
- **Adicionales:** cables, estructura, conectores, reguladores de tensión u otros necesarios para el funcionamiento o apoyo a los componentes principales.

## **2.2. Selección de componentes**

### **2.2.1. Teléfono celular con cámara**

Se sugiere un teléfono con sistema operativo Android ya que posee una tienda de aplicaciones con muchas opciones para el streaming de video entre este y la pc; para el desarrollo de este sistema se utilizará la aplicación DroidCam.

Se recomienda que la cámara posea una resolución mínima de 2Mpx, para obtener detalles suficientes en la imagen.

### **2.2.2. Interfaz de Usuario**

Para poder obtener una interfaz a nuestras necesidades se decidió elaborarla utilizando el lenguaje Python ya que posee módulos para crear interfaces gráficas, comunicación serial, procesamiento de imágenes y data science.

Para el procesamiento de imágenes se aplicará la librería de OpenCV y PySerial para la comunicación del puerto serial.

### 2.2.3. Circuito Electrónico

Este circuito tendrá como base Arduino ya que es fácil de comunicar con diferentes elementos y posee un diseño de fácil conexión, además de una amplia librería para trabajar comunicación serial, PWM, señales analógicas y digitales, junto a una gran comunidad de respaldo.

Tomaremos al arduino nano para mostrar algunas características:

Parámetro	Descripción
Micro controlador	Atmel ATmega328
Voltaje de Operación	5 V
Voltaje de entrada	7-12 V
Rango máximo y mínimo de voltaje	6-20 V
Entradas/Salidas Digitales I/O	14
Canales PWM	6
Entradas analógicas	8
Memoria Flash	32 KB (ATmega328) 2 KB used by bootloader

Tabla 1 Características Arduino Nano  
(Arduino, s.f.)

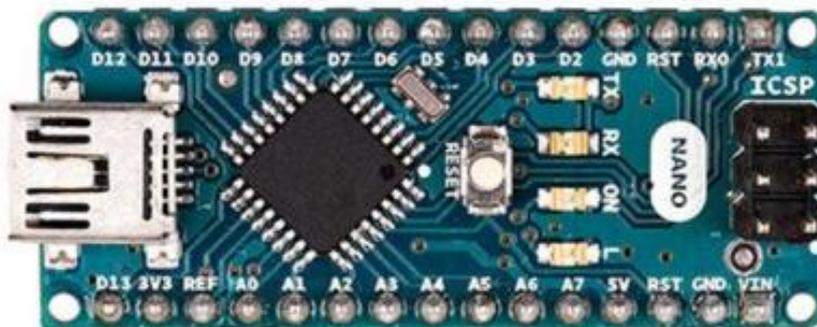


Ilustración 9 Arduino Nano  
(Arduino, s.f.)

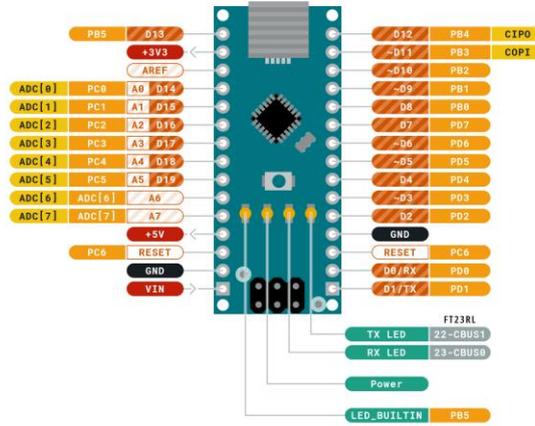


Ilustración 10 Pinout Arduino Nano  
(Arduino, s.f.)

### 2.2.4. Driver de Puente H

El driver propuesto es el L298N, el cual presenta las siguientes características:

Parámetro	Descripción
Chip	L298N
Canales	2 (soporta 2 motores DC o 1 motor PAP)
Voltaje lógico	5V
Voltaje de potencia (V motor)	5V - 35V DC
Consumo de corriente (lógico)	0 a 36mA
Capacidad de corriente	2A (picos de hasta 3A)
Potencia máxima	25W

Tabla 2 Características Módulo L298N  
(Naylamp Mechatronics SAC, s.f.)

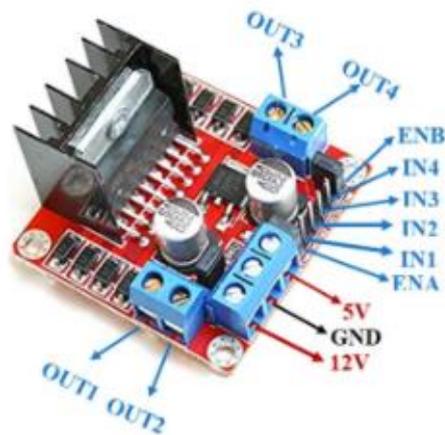


Ilustración 11 Módulo L298N pinout  
(Components101, 2020)

Pin Name	Description
IN1 & IN2	Pines de entrada del motor A. Se utiliza para controlar la dirección de giro del motor A
IN3 & IN4	Pines de entrada del motor B. Se utiliza para controlar la dirección de giro del motor B
ENA	Habilita la señal PWM para el motor A
ENB	Habilita la señal PWM para el motor B
OUT1 & OUT2	Pines de salida del Motor A
OUT3 & OUT4	Pines de salida del motor B
12V	Entrada de 12V de la fuente de alimentación de CC
5V	Suministra energía para los circuitos lógicos de conmutación dentro del CI L298N
GND	Pasador de tierra

Tabla 3 Descripción del pinout del módulo L298N  
(Components101, 2020)

### 2.2.5. Motor DC

Debido a que un grado en el giro del motor puede representar una distancia considerable entre las coordenadas de una imagen, la cual se ve aumentada a medida el objetivo se ubique más lejanamente a la cámara, el movimiento del motor no debe ser muy rápido; por ello se escogió un motor reductor, el cual

además de moverse lento, también puede mover cargas más pesadas, lo que ahorra en el diseño de engranajes.

Sus características son:

Parámetro	Descripción
Voltaje nominal	DC 12V
Velocidad en RPM	16RPM
Diámetro del eje	6mm
Tamaño de la caja reductora	37mmx26mm
Tamaño del motor	32mmx16mm

Tabla 4 Características motor DC seleccionado  
(ElectroPro, s.f.)



Ilustración 12 Motor DC  
(ElectroPro, s.f.)

#### 2.2.6. Fuente de alimentación

La fuente de alimentación debe proveer 5V con una capacidad de corriente de al menos 0.5A, ya que se considera que el circuito tendrá a consumo no superior a esa cantidad, esta puede ser una batería recargable o un cargador de celular ya que estos suelen ser de este voltaje.

### 2.2.7. Módulo Bluetooth

Con el fin de reducir el número de cables externos al dispositivo y así evitar enredos o inconvenientes, se plantea usar un módulo bluetooth para comunicar el Arduino con la computadora, cabe decir que el pc posee esta opción de comunicación preinstalada, caso contrario se puede reemplazar por un cable de comunicación, para el caso del Arduino nano US-micro USB, utilizando el protocolo serial.

El módulo utilizado es el HC-06 y presenta las siguientes características:

Parámetro	Descripción
Voltaje de operación	3.3V - 5VDC
Corriente de operación	< 40mA
Corriente modo sleep	< 1mA
Chip	BC417143
Bluetooth	V2.0+EDR
Frecuencia	Banda ISM de 2,4 GHz
Modulación	GFSK (Gaussian Frequency Shift Keying)
Potencia de emisión	4 dBm, clase 2
Sensibilidad	-84dBm a 0.1% VER
Alcance	10 metros
Interfaz de comunicación	Serial UART TTL
Velocidad de transmisión	1200bps hasta 1.3Mbps
Baudrate por defecto	9600,8,1,n.
Velocidad asíncrona	2.1Mbps (máx.) / 160 kbps.
Velocidad síncrona	1Mbps/1Mbps
Seguridad	Autenticación y encriptación

Dimensiones	37*16 mm
Peso	3.2 gramos

Tabla 5 Características Módulo HC-06  
(Naylamp Mechatronics SAC, s.f.)

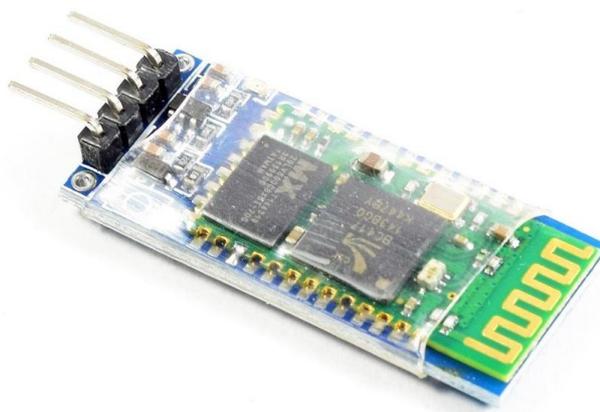


Ilustración 13 Módulo HC-06  
(Naylamp Mechatronics SAC, s.f.)

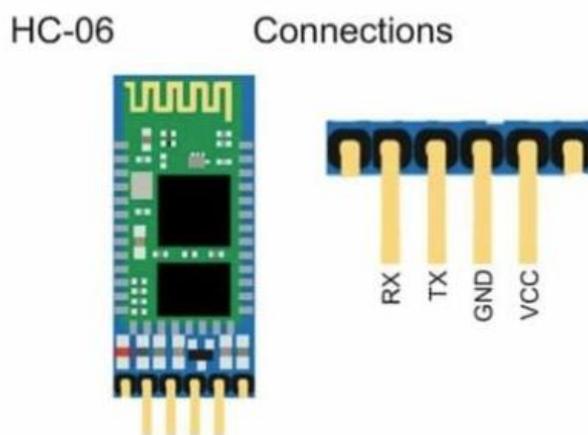


Ilustración 14 Pinout del módulo HC-06  
(UNIT Electronics, s.f.)

### 2.2.8. Sensor infrarrojo

Con el fin de poder determinar el tope en el movimiento vertical del dispositivo, se considera el uso de un sensor infrarrojo que pueda activarse cuando el brazo esté cerca a este; este sensor puede ser reemplazado según

convenga ya sea por un final de carrera o incluso el giroscopio del celular, en cas se utilice este como cámara, detentando el grado de inclinación.

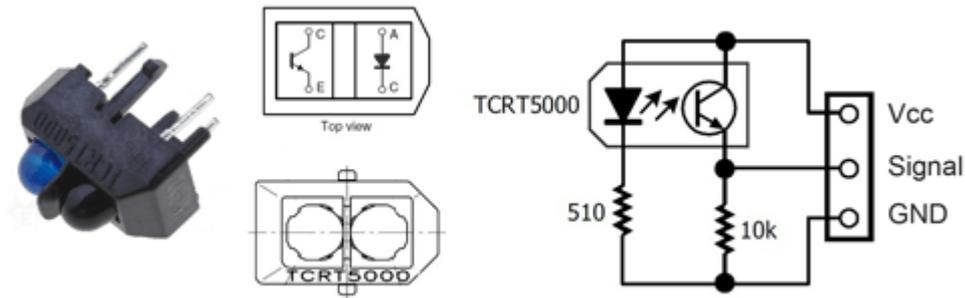


Ilustración 15 Sensor infrarrojo  
( (Llamas, 2016))

### 2.2.9. Estructura plástica de dispositivo

Una vez ya conectada todas las partes es necesario poder ubicarlas en un chasis que las proteja y ordene, además servirá de soporte y facilitará el transporte. Haciendo uso de una impresora 3D, se puede modelar el diseño deseado e imprimirlo, así mismo se puede realizar algún cambio de ser necesario, esto ayuda a no limitar mucho los componentes en tamaño o disposición.

### 2.3. Diagrama de flujo del algoritmo de control general

Para poder desarrollar el sistema, se analiza los principales puntos de trabajo y como se espera funcionen; empezando desde la captura del video y terminando en el movimiento de los motores, los cuales centrarán la cámara en el expositor.

De manera general se presenta el siguiente diagrama de flujo para el sistema planteado.

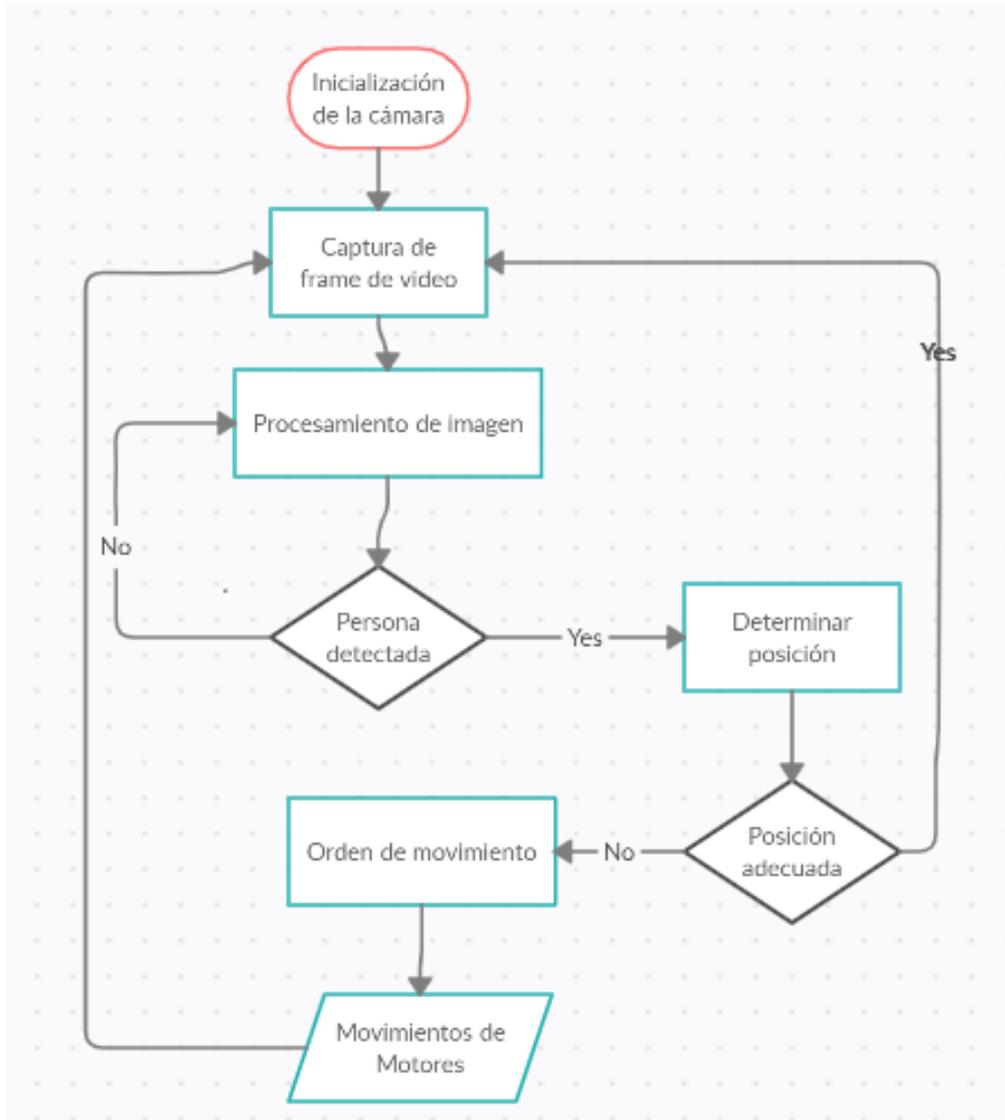


Ilustración 16 Diagrama de flujo del proceso (fuente propia)

Antes de iniciar el sistema se debe verificar la conexión de comunicación entre el circuito y la PC, para ello se debe seleccionar el puerto serial correspondiente al circuito.

Como se puede apreciar en el diagrama de flujo, al iniciar el programa solicita que la cámara esté activa y enviando el video para que así se pueda procesar estos datos.

Una vez procesada la el frame de video recibido, en caso encontrara a un individuo, determinará su posición relativa en la imagen, luego calculará el error con respecto a la posición que deba estar y luego comunicará este dato al circuito para poder mover los

motores de ser necesario. En el caso que no detecte al individuo, continuará con su búsqueda, repitiendo el proceso anterior hasta hallar a uno.

Finalmente, el proceso se repite con el fin de ubicar nuevamente a la persona en caso se moviera, realizando así un seguimiento de su posición en tiempo real.

## 2.4. Diseño e Implementación

### 2.4.1. Captura de Video y transmisión a PC

Podemos utilizar cualquier cámara que se conecte a nuestra PC, puesto que el software que se desarrolle más adelante será el encargado de realizar el procesamiento de la imagen; para este caso, con el fin de representar un ahorro, además que es un elemento del cual la mayoría dispone, utilizará un teléfono Android, al cual se le descargará e instalará la aplicación DroidCam (también existe una versión para iPhone), esta presenta una versión gratuita, disponible actualmente en la PlayStore; este programa permite utilizar la cámara nuestro celular como una webcam en la PC, lo que nos permite tomar dicha imagen y trabajarla.

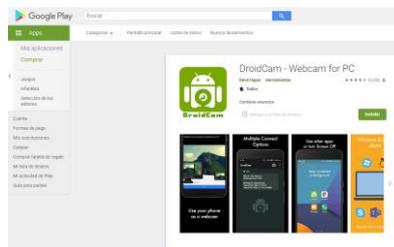


Ilustración 17 DroidCam en PlayStore  
(fuente propia)

También es necesario instalar la versión para a PC, ya que sirve de receptor y enlace; para ello nos dirigimos a la página oficial <http://www.dev47apps.com/> y procedemos a seleccionar nuestro sistema operativo, en este caso Windows.

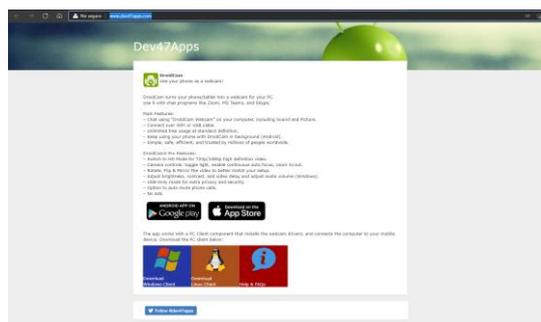


Ilustración 18 Página oficial de DroidCam  
(fuente propia)

Ya descargada procedemos a instalarla dando doble click el archivo obtenido y seleccionado install en la ventana que aparece y luego en finish.

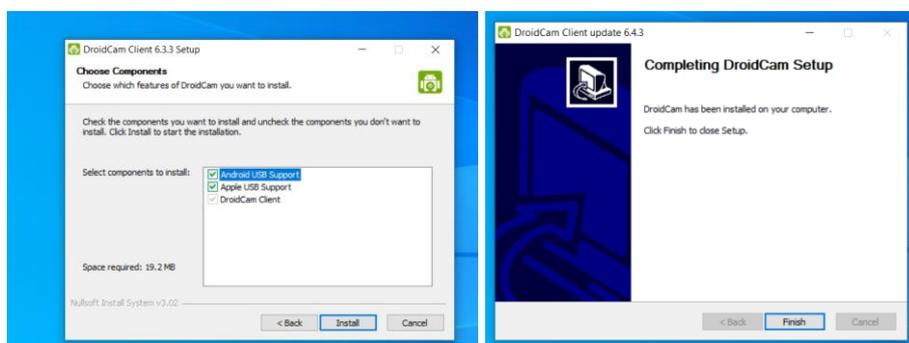


Ilustración 19 Instalación de DroidCam en PC  
(fuente propia)

Ya teniendo el software instalado en ambos dispositivos, el siguiente punto importante es fijarse que se encuentren en la misma red, es decir el teléfono debe estar conectado al WIFI del mismo equipo al que está la computadora, esto ya que el programa necesita comunicarse con ambos elementos, en este caso utilizando la misma red LAN, aunque existe la posibilidad de realizar la conexión vía USB, para evitar las limitaciones de distancia y enredo dejamos de lado esta opción.

Al abrir la aplicación el teléfono celular nos aparecerá una pantalla similar a la siguiente.

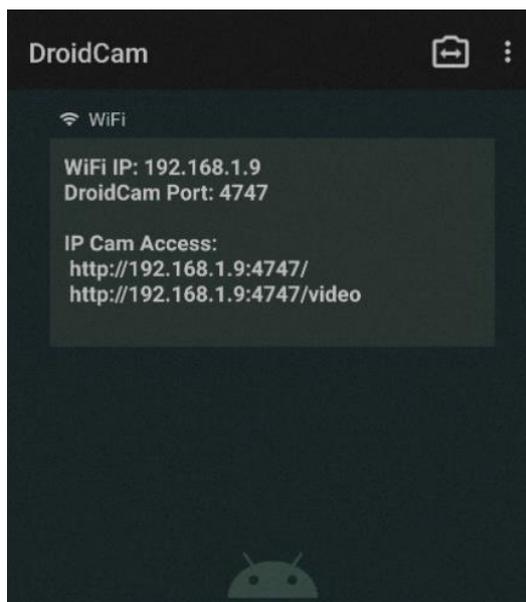


Ilustración 20 Pantalla de inicio de DroidCam en celular  
(fuente propia)

En esta pantalla los datos que debemos tener en cuenta son los de WiFi IP el cual en este caso es 192.168.1.9 y DroidCam Port dado por 4747.

Para escoger entre la cámara frontal y trancera, se presiona el ícono de la cámara con las flechas, de la parte superior derecha, aparecerá un pequeño menú donde seleccionamos la cámara a utilizar.

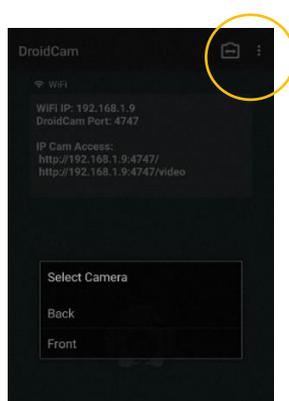


Ilustración 21 Selección de cámara en DroidCam  
(fuente propia)

Por su parte al iniciar el programa en la PC nos parece una ventana como la que se muestra a continuación.

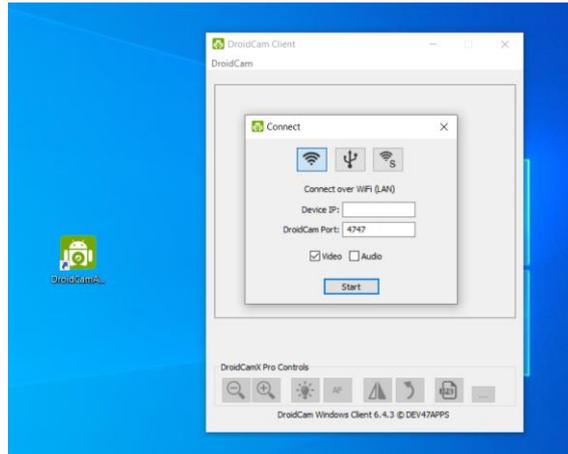


Ilustración 22 Ventana de inicio DroidCAM en PC  
(fuente propia)

La cual presenta dos recuadros editable; en el primero (Device IP) colocaremos el valor obtenido de campo WiFi IP, mientras que en el campo de DroidCam Port el segundo valor mencionado anteriormente, en este caso aparece predefinido el valor 4747, que resulta ser el mismo del mostrado en el celular.

Ya colocados los valores, se inicia dando click en el boton de Start, de estar todo correcto podremos ver la imagen de lo que capture nuestra cámara tanto en el celular como en el PC.

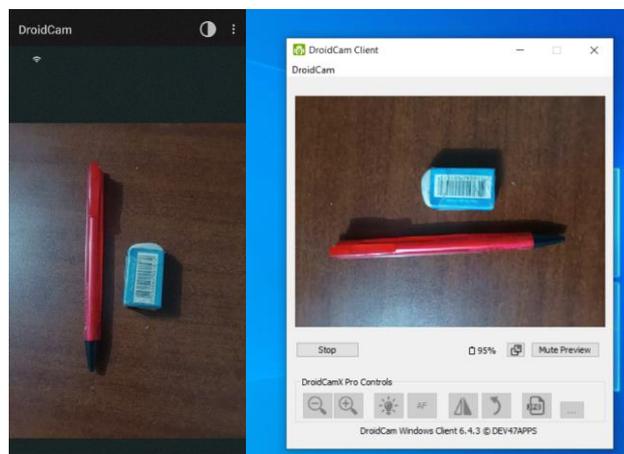


Ilustración 23 Imagen mostrada en el teléfono y PC  
(fuente propia)

#### 2.4.2. Software de procesamiento de video

Para el desarrollo del software se decidió utilizar el lenguaje Python, en conjunto con las librerías OpenCV para el tratamiento de las imágenes, puesto que presenta módulos para detectar rostros y tracking de elementos; también está PySerial para la comunicación con el circuito electrónico.

La lógica aplicada consiste en:

1. Verificar que exista una línea video transmitiéndose y un puerto serie que comunique con el circuito a los cuales conectarse, de ser así, enlazarse a estos. Para el caso del bluetooth, se deberá haber emparejado con anterioridad el módulo a la PC (se mostrará los pasos más adelante), ya teniéndolo hecho, este se comporta como un puerto serie convencional.
2. Recibir el frame captado por la cámara, este lo tomamos de DroidCam ya que funciona como una cámara web convencional en nuestra PC.
3. Aplicar la función de detección de rostros en la imagen, la cual nos arroja las coordenadas de los elementos que detecte en la imagen.
4. Tomar las coordenadas obtenidas y enmarcar visualmente el objetivo.
5. Enviar la coordenada al circuito mediante el protocolo serial por el puerto anteriormente seleccionado.
6. Aplicar la función de tracking a las coordenadas obtenidas. Esta función es más rápida que la de detección, por lo que permite una mayor fluidez en la reproducción de video.
7. Obtener los nuevos valores generados por la función trabajarlos como en los puntos 4 y 5.
8. Para el caso en que el tracking presente errores, se vuelve a aplicar la detección para hallar al ponente en la escena.

El diagrama de flujo para el software es el siguiente:

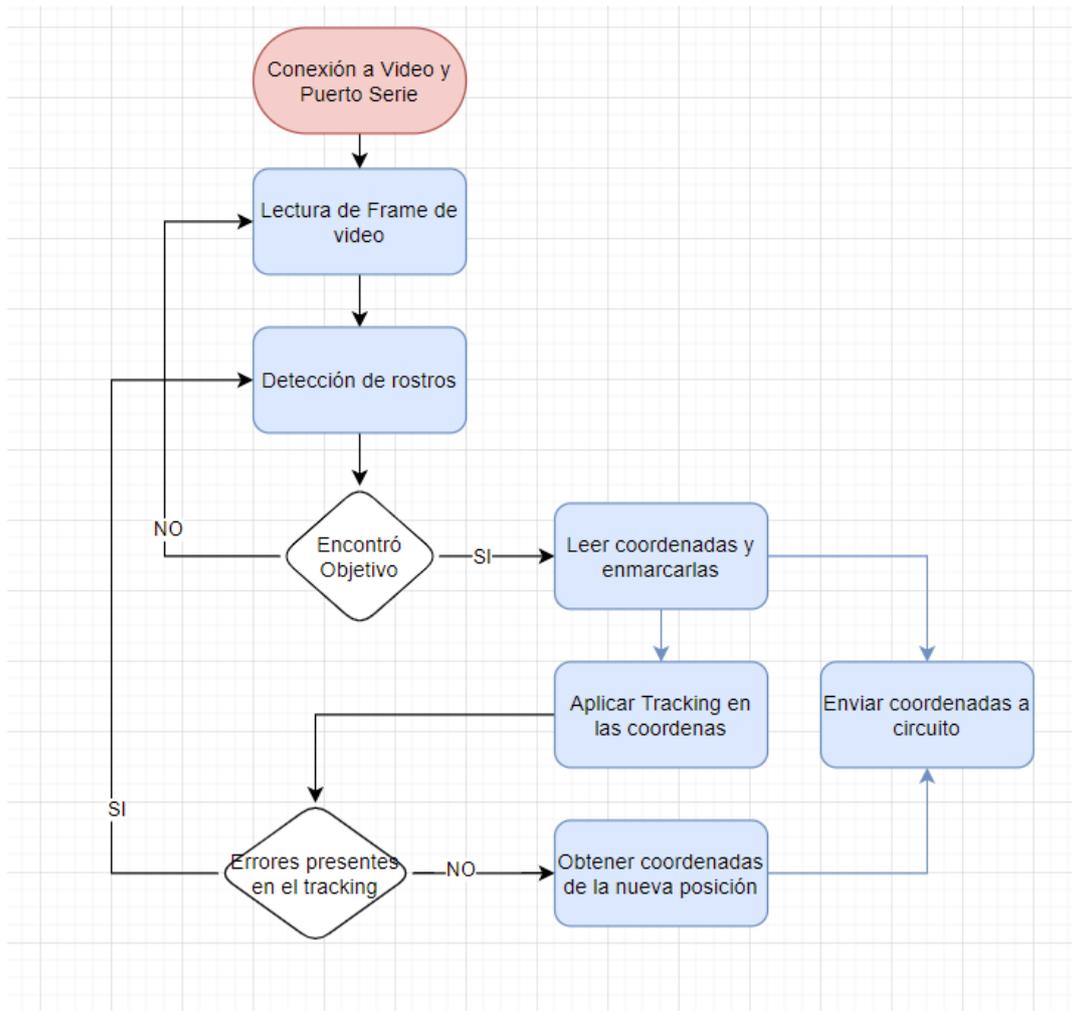


Ilustración 24 Diagrama de flujo del software de reconocimiento y seguimiento (fuente propia)

### 2.4.3. Software Arduino Nano

Las principales funciones a tener en cuenta son las de recibir las coordenadas de la persona encontrada, calcular el error para el desplazamiento y controlar los motores.

La secuencia a seguir es la siguiente:

1. Inicializar puertos de entradas y salidas, tanto para el control del motor y la recepción de datos seriales.
2. Espera la recepción de las coordenadas enviadas por el software en PC.
3. Calcular el error con respecto al centro de la imagen

4. Aplica la función de control y generar la señal de salida al driver de potencia para el control de los motores.

El diagrama de la ruta es el siguiente:

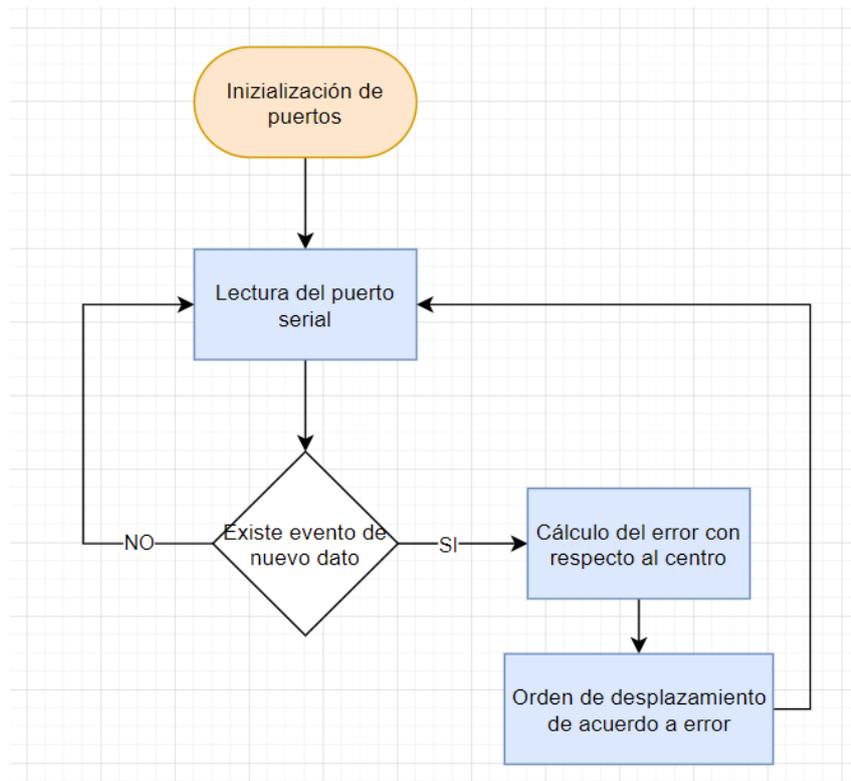


Ilustración 25 Diagrama de Flujo para el microcontrolador (fuente propia)

Es importante señalar que para el control de los motores se utilizará PWM, ya que esto permite regular la velocidad de giro de estos. Los pines con capacidad de generar la señal PWM en el Arduino Nano son 3,5,6,9,10,11; en la Ilustración 10 Pinout Arduino Nano, están identificados con el símbolo “~”.

#### 2.4.4. Diseño del armazón en FreeCad

La estructura del equipo fue diseñada en el software FreeCad, ya que este es software libre y presenta muchas herramientas para el modelado 3D.

Se planteó un diseño en el cual, para evitar enredamiento de cables internos, el eje del motor horizontal girase apuntando hacia abajo, así si consideramos que

la base está fija al suelo, toda la parte superior gira junto con el chasis del motor, incluido el microcontrolador y el teléfono. Es importante que el disco sea amplio ya que permitirá frenar un poco el peso del teléfono celular.

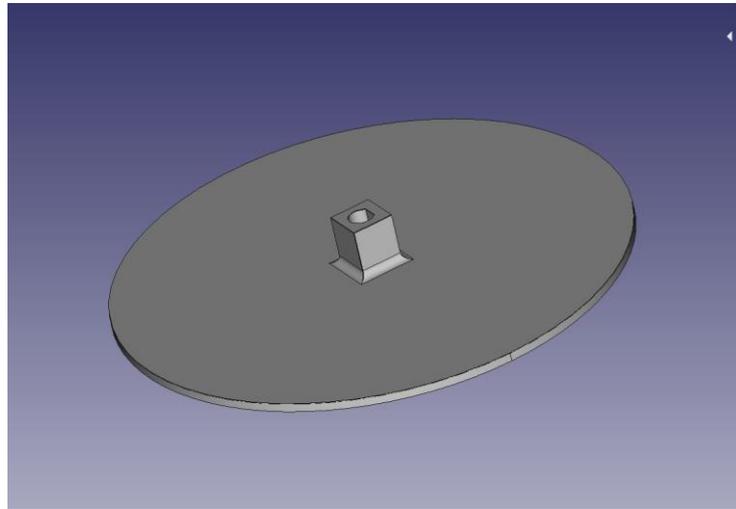


Ilustración 26 Base para el motor en contacto con el piso  
(fuente propia)

La parte superior (motor que sujeta al teléfono) e inferior (motor que gira sobre su eje), están unidos por medio de un sistema en “L”.

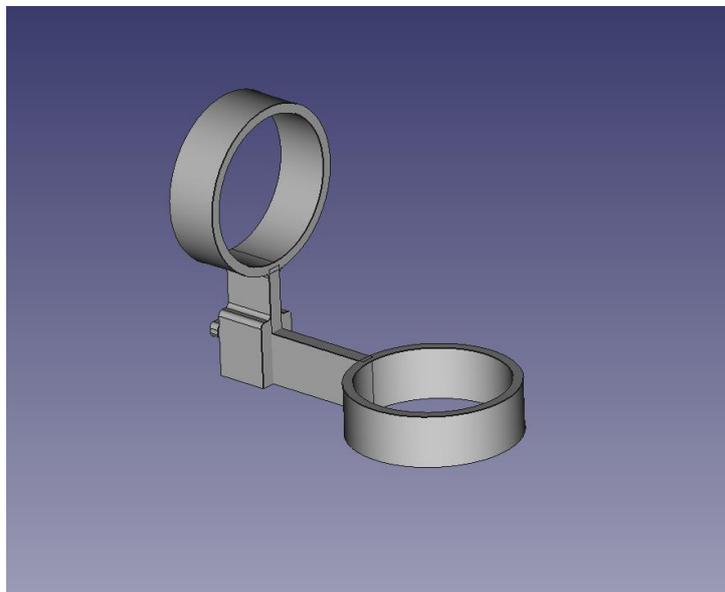


Ilustración 27 Pieza de unión entre los motores  
(fuente propia)

Se calculó el ancho y alto interno teniendo en cuenta los elementos que se colocarían internamente.

Como para este caso se utilizará un teléfono celular, se optó por usar el soporte basado en un selfie stick, al eje que une el celular con el equipo, a este se le puede agregar algún implemento para mejorar la adherencia, como una cinta doble cara o algún sujetador externo.

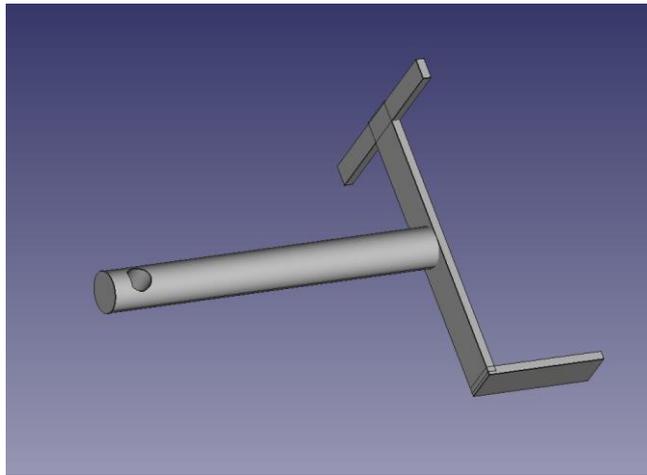


Ilustración 28 Diseño de eje que soporta al teléfono  
(fuente propia)

Las piezas en conjunto se complementan de la siguiente forma:

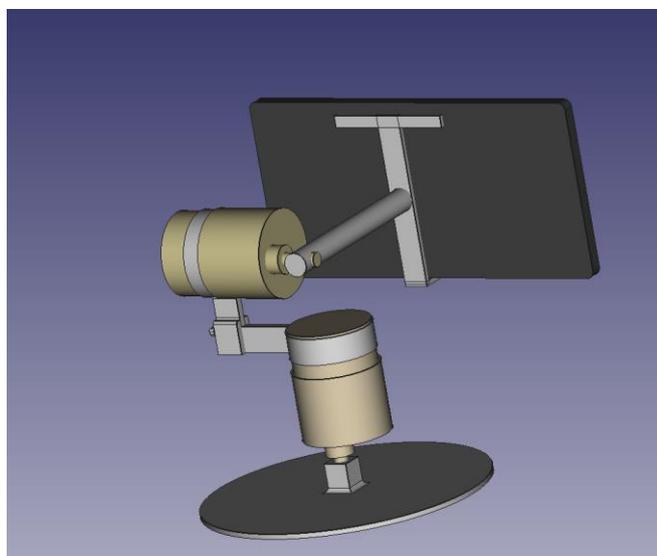


Ilustración 29 Vista del modelo general  
(fuente propia)

Una vez ya modelado todos los elementos, se procedió a imprimirlo en una impresora 3D.

#### 2.4.5. Emparejamiento con el módulo Bluetooth

Iniciaremos por la vinculación entre el módulo bluetooth y la PC. Para ello primero encenderemos el módulo bluetooth, para ello conectaremos una fuente de voltaje en los pines VCC y GND, también, para hacer algunas pruebas, podemos conectarlo directamente a un Arduino como se muestra en la imagen, tanto los pines de alimentación como los de comunicación serial, para recibir los datos.

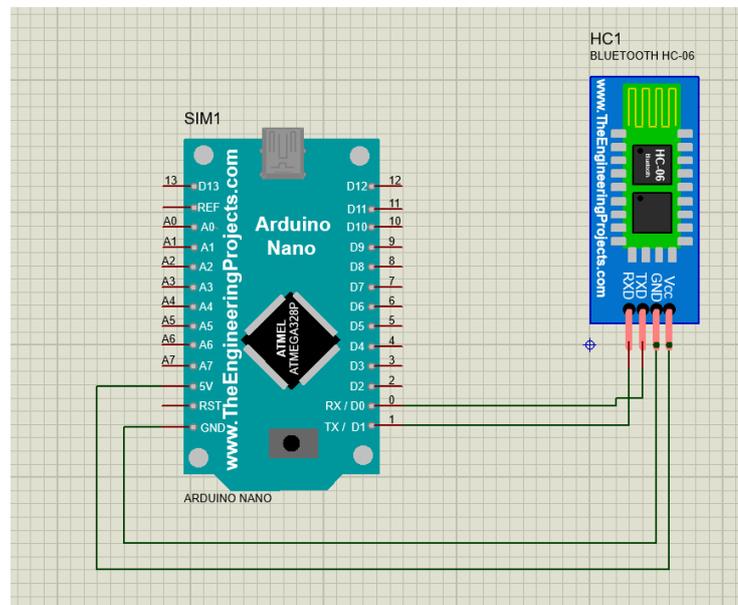


Ilustración 30 Conexión Módulo Bluetooth  
(fuente propia)

Nos percataremos que está encendido porque el led incorporado al módulo comienza a parpadear.

Ahora verificaremos que el bluetooth del PC se encuentra encendido, tomando como base el sistema operativo Windows 10, daremos click en el ícono de notificaciones ubicado en la parte inferior derecha de la pantalla, este desplegará una ventana, en la parte inferior de esta se encuentran diferentes opciones, nos centraremos en la bluetooth (en caso no aparezca, dar click en

expandir, esto mostrará algunas opciones ocultas, de lo contrario podría ser problemas de drivers o la pc no tiene un chip bluetooth conectado), para encenderlo simplemente se le dará click y cambiará a un color azul, del mismo modo para apagarlo, cambiando un color negro; sin embargo para este caso, con el fin de vincular un nuevo dispositivo, le daremos anticlik y escogeremos la opción de Ir a Configuración.



Ilustración 31 Activación de Bluetooth en PC (fuente propia)

En esta nueva ventana también podremos activar el bluetooth en caso estuviese apagado; además de una lista de los equipos ya vinculados. Para emparejar el módulo, debemos seleccionar en “Agregar Bluetooth u otro dispositivo”, luego en la primera opción “Bluetooth”.

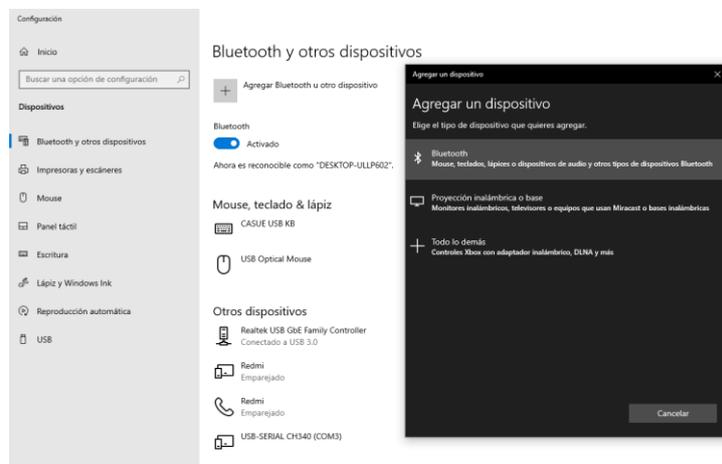


Ilustración 32 Añadir nuevo dispositivo Bluetooth en PC (fuente propia)

Aparecerá una lista con los dispositivos disponibles, el que utilizamos es el HC-06, por lo que lo seleccionamos. Aparecerá un cuadro solicitando una contraseña de vinculación, esta es “1234”; habiéndola ya colocada, daremos click en conectar, finalmente aparecerá un mensaje indicando que el dispositivo está listo, adicionalmente aparecerá en la lista de dispositivos ya vinculados.

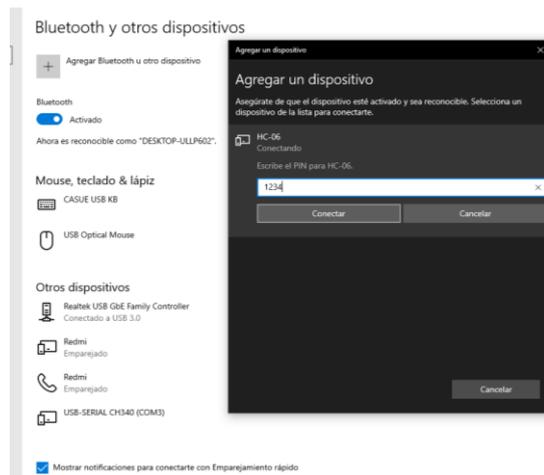
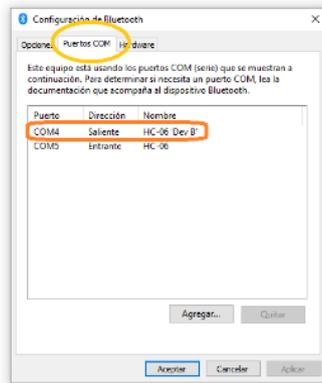


Ilustración 33 Contraseña para vinculación del HC-06  
(fuente propia)

Lo que sigue ahora es verificar el puerto serie que enlaza la comunicación entre el módulo bluetooth y la PC, para ello nos dirigimos a la parte derecha de la ventana de configuración en la que nos encontramos y ubicaremos la opción de “Más opciones de Bluetooth”; emergerá una nueva ventana, de esta última, escogemos la pestaña “Puertos COM”, aquí se muestra los puertos series vinculados, para este caso el que utilizaremos será el que indica “Saliente” para el “HC-06”, que vendría a ser el COM4.



#### Activar Bluetooth aún más rápido

Para activar o desactivar Bluetooth sin abrir Configuración, abre el centro de actividades y selecciona el icono Bluetooth.

#### Configuración relacionada

[Dispositivos e impresoras](#)

[Configuración de sonido](#)

[Configuración de pantalla](#)

[Más opciones de Bluetooth](#)

[Enviar o recibir archivos a través de Bluetooth](#)

[Ayuda de la web](#)

[Corrección de conexiones Bluetooth](#)

[Reinstalar controladores Bluetooth](#)

[Compartir archivos a través de Bluetooth](#)

[Obtener ayuda](#)

[Enviar comentarios](#)

Ilustración 34 Verificación del puerto serie asociado al HC-06 (fuente propia)

## 2.4.6. Implementación física

Los componentes electrónicos llevan la siguiente conexión.

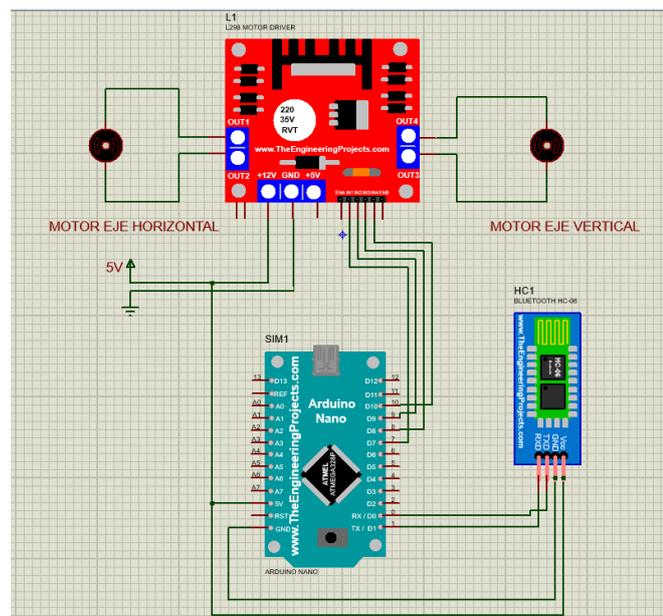


Ilustración 35 Conexión de los elementos eléctricos (fuente propia)

En el esquema se observa al módulo bluetooth unido al Arduino por su puerto serie, esto permite la entrega de datos, así mismo el Arduino controla los motores por medio de sus 4 pines digitales (7,8,9,10) conectándolos a las entradas del módulo L298N. De las salidas del L298N, se conectan los motores correspondientes al control de cada lado, izquierda el motor del eje horizontal y derecha del vertical. Finalmente se observa que el sistema comparte la misma fuente de alimentación, esto porque todos pueden trabajar a los 5V; para el caso de los motores, aunque sus especificaciones indican un voltaje nominal de 12V, realizando pruebas se verificó que pueden operar a 5V, limitándolos un poco en el torque máximo que puedan generar.

Se opta por una conexión cableada, con el fin de poder ubicar de mejor manera cada componente dentro del chasis.



Ilustración 36 conexión de elementos eléctricos  
(fuente propia)

Del mismo modo la estructura interna que soporta a los motores, así como el plato inferior y el brazo para el celular, quedan de la siguiente forma:



Ilustración 37 Estructura interna  
(fuente propia)

Todo el sistema implementado se ve de la siguiente manera:



Ilustración 38 Equipo implementado  
(fuente propia)

## **CAPITULO III. RESULTADOS Y DISCUSIÓN**

### 3.1. Software implementado

#### 3.1.1. Comunicación

En el software recibe correctamente la imagen y envía los datos por el puerto serial. Al iniciar comprueba si existe una señal de video con la que trabajar, de ser así procede a preguntar el puerto al que conectarse, en nuestro caso el COM4.

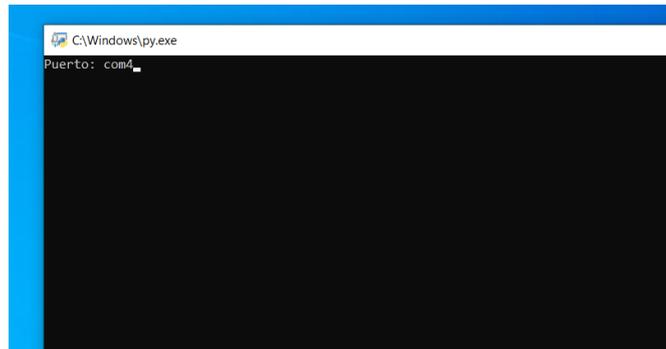


Ilustración 39 Inicio del software  
(fuente propia)

#### 3.1.2. Detección y Seguimiento

Las imágenes son procesadas por la librería de OpencCV, pudiendo reconocer el rostro y devolver las coordenadas de este; así, se puede enviar este dato al microcontrolador, además de enmarcarlo en la imagen para una mejor visualización del usuario.

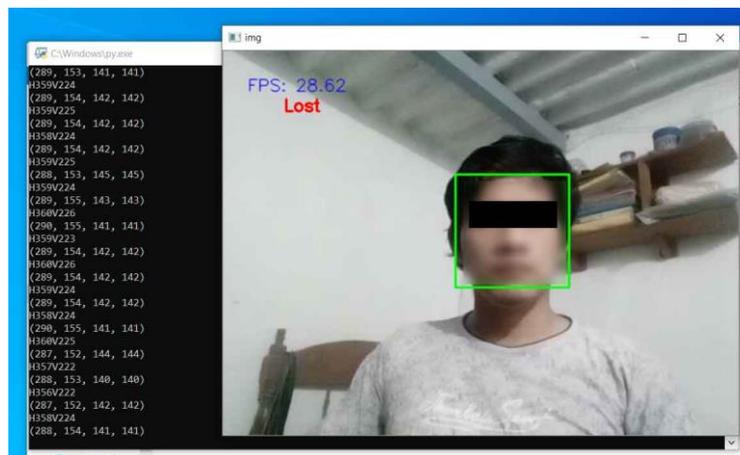


Ilustración 40 Software detectando  
(fuente propia)

Se puede observar que cuando el objetivo está perdido aparece mensaje de “Lost”, pero cuando lo detecta nuevamente, lo enmarca primero en verde, indicando que está detectando, y luego pasa a un modo de “Tracking”, cambiando a un cuadro más pequeño de color rosa, esto se hizo con el fin de poder observar mejor el cambio entre ambos estados.

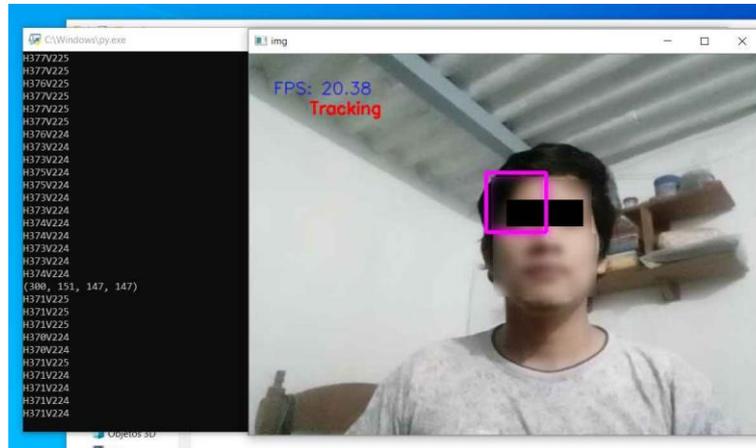


Ilustración 41 Software siguiendo al individuo  
(fuente propia)

### 3.1.3. FPS

El promedio de FPS en el video resultante es alrededor de 20 FPS, lo que permite un video lo suficientemente fluido para el espectador, esto gracias a que se implementó la función de tracking y no solo la de detección, por ejemplo, en la siguiente imagen donde el programa intenta buscar a una persona, pero no es capaz de hallarla, solo esta efectuando la detección y no el tracking, obteniendo un promedio de 11 FPS.

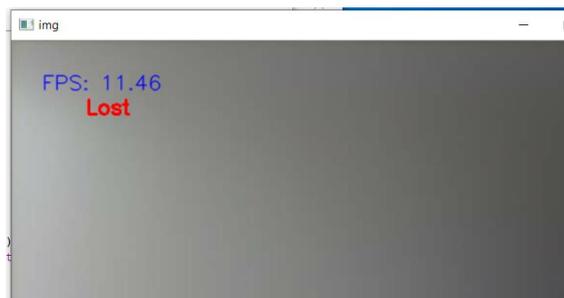


Ilustración 42 Software buscando objetivo  
(fuente propia)

### 3.1.4. Limitantes

El software puede perder a la persona cuando esta se desplaza muy rápido o cuando existe algún fallo en la detección.

Con respecto a los fallos, principalmente ocurrían cuando la iluminación era débil o el fondo irregular, la que el algoritmo no es capaz de identificar correctamente las formas para con las que fue entrenado su filtro, ocasionando que no encuentre al objetivo o errando en su predicción señalando un elemento distinto.

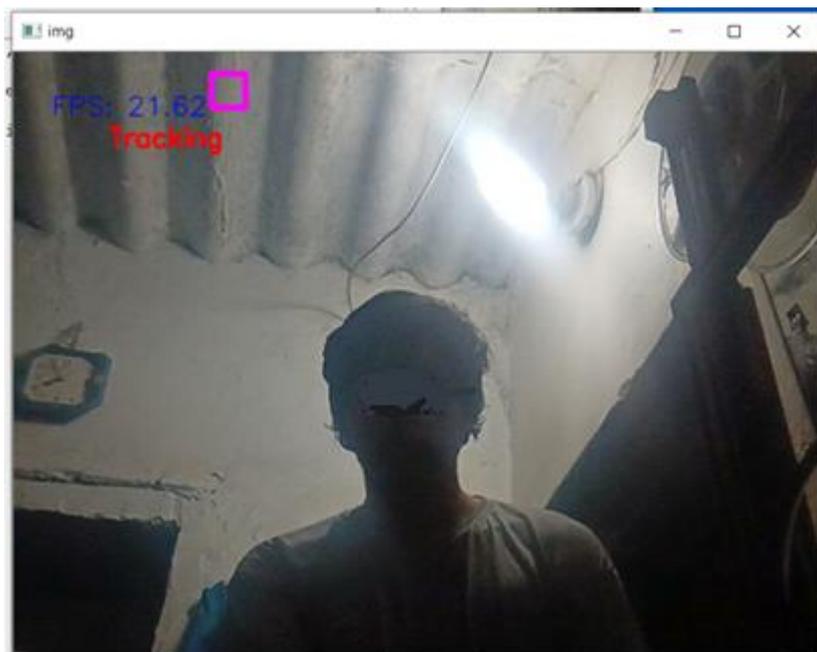


Ilustración 43 Fallo en la detección  
(fuente propia)

## 3.2. Circuito de Control

La parte física del proyecto cumple con el propósito de seguir al individuo, los ejes de los motores giran en las direcciones esperadas y se detienen cuando las coordenadas recibidas indican que el individuo está centrado.

### 3.2.1. Tiempo de ejecución del movimiento

Haciendo uso de un cronometro se pudo calcular el retardo desde que la persona inicia el movimiento hasta cuando el equipo logra centrar nuevamente al sujeto.

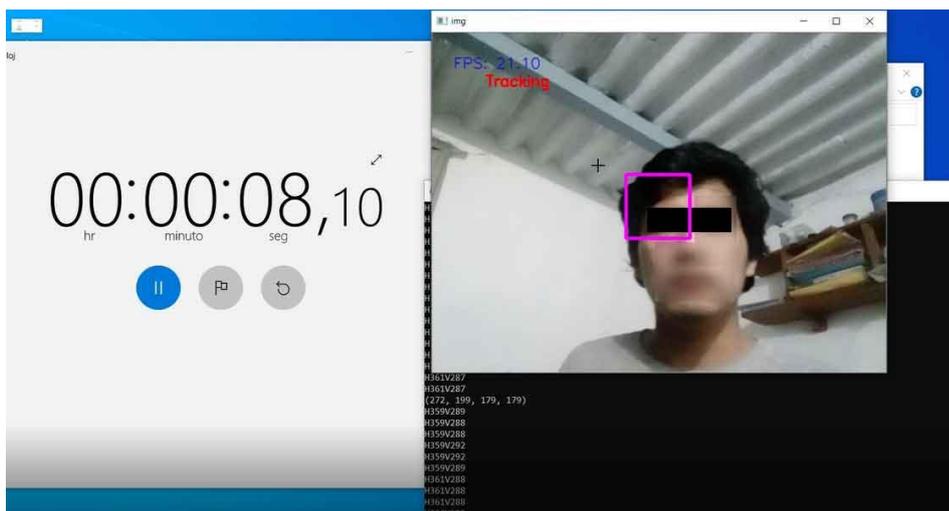


Ilustración 44 Inicio del movimiento  
(fuente propia)

Para este caso el movimiento ha iniciado cerca de los 8 segundos. Aquí el usuario se comenzará a mover hasta un lado de la pantalla, tratando de alejarse lo mayor posible del centro.

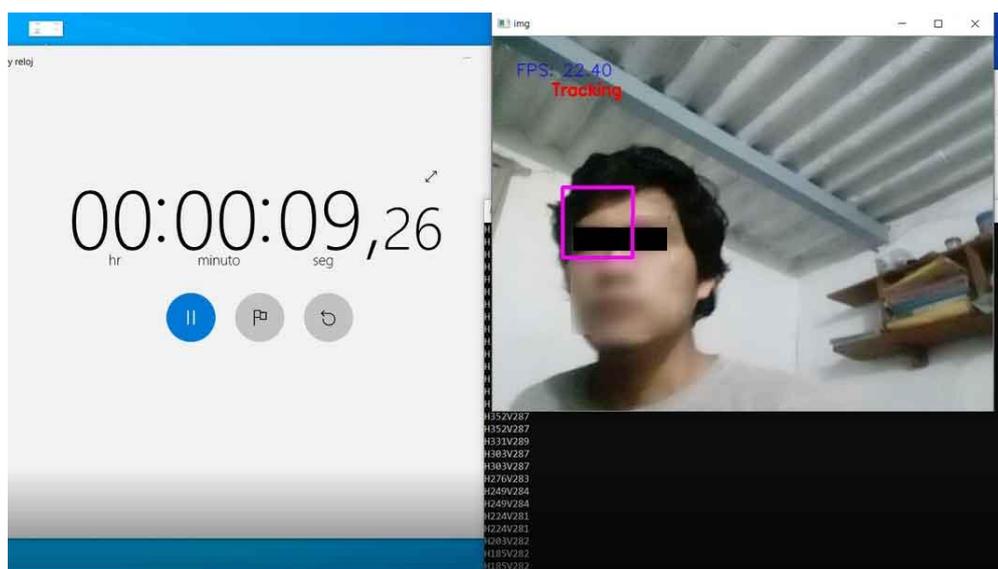


Ilustración 45 Punto máximo de movimiento con cámara estática  
(fuente propia)

En la imagen de arriba observamos el punto máximo que logra desplazarse la persona sin que el equipo físico genere movimientos en la cámara, este vendría a ser el tiempo de reacción de nuestro equipo, teniendo un tiempo aproximado de 1.2 segundos; este tiempo viene dado por los retardos en el envío de dato,

recepción y procesamiento de las coordenadas, además de tiempo de arranque de los motores.

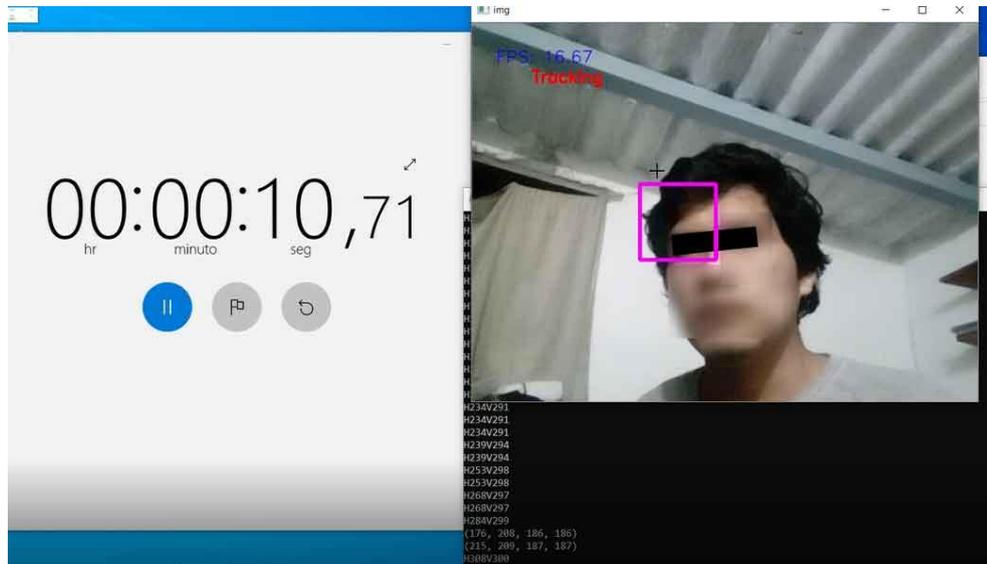


Ilustración 46 Objetivo centrado nuevamente (fuente propia)

Finalmente, el objetivo es centrado luego de iniciado el movimiento, este tiempo en promedio es de 1.5 segundos, desde el inicio del movimiento en su punto máximo hasta su detención. Considerando que el ángulo de desplazamiento fue aproximadamente  $31^\circ$ , tendríamos una respuesta a 20.66 grados/s.

### 3.2.2. Ángulo y velocidad de desplazamiento

Nuestro equipo puede girar horizontalmente  $360^\circ$  y verticalmente, por temas de chasis, está limitado a  $240^\circ$ . Por su parte, la velocidad de desplazamiento estará limitada por las características del motor, en este caso, según indica su etiqueta es de 16 rpm.

### 3.2.3. Consumo de energía

En cuanto a la corriente necesaria para funcionar, se realizaron dos pruebas, una cuando el equipo se encuentra en reposo y la otra cuando activa los motores. En la primera se tiene que el consumo del sistema es de 0.08A, mientras que en la segunda el consumo llega a 0.13A.



Ilustración 47 Corriente con sistema en reposo  
(fuente propia)



Ilustración 48 Corriente con sistema en movimiento  
(fuente propia)

### 3.3. Presupuesto

#### 3.3.1. Presupuesto del equipo

Ítem	Cantidad	Precio unitario (S/)	Precio total (S/)
Arduino Nano	1	25	25
L298N	1	15	15
Módulo bluetooth	1	25	25
Motor	2	40	80
Batería	1	30	30
Sensor infrarrojo	1	2	2
Cable	10	0.3	3
Armazón	1	50	50
TOTAL			230

#### 3.3.2. Presupuesto de elementos complementarios

Ítem	Cantidad	Precio unitario (S/)	Precio total (S/)
Laptop/PC	1	2000	2000
Teléfono/Cámara	1	300	300
TOTAL			2300

En total ambos presupuestos suman S/ 2530.

## **CAPITULO IV. CONCLUSIONES**

## CONCLUSIONES

- 1) Se logró de diseñar un prototipo de detección y seguimiento de la posición de personas para ponentes en la UNPRG.
- 2) Se logró implementar un software que unifica detección, seguimiento y envío de datos. Se verificó la eficiencia de procesamiento al determinar los FPS que entrega cuando está trabajando, observando que, si sólo se utilizara el método de detección, la tasa de FPS sería menor, ya que este método es más lento en comparación al de tracking, ya que este último emplea algoritmos de mayor velocidad. También se trataron de identificar las principales limitantes en diferentes entornos.
- 3) El circuito trabajó correctamente, pudiendo recibir y procesar los datos recibidos; así mismo controló de manera efectiva el movimiento según la posición del usuario. El movimiento resultó fluido, de manera general, a pesar de existir retardos tanto para el inicio del movimiento, como el del desplazamiento en sí.
- 4) Para evitar sistemas complicados de engranajes se optó por utilizar motores reductores, ya que estos poseen un sistema interno de engranajes; así mismo esto facilitó el control del movimiento ya que un pequeño ángulo de giro puede significar un gran desplazamiento en el objetivo, en especial si este se encuentra lejos, ya que se forma un cono con respecto al origen.

## **CAPITULO V. RECOMENDACIONES**

## RECOMENDACIONES

- 1) Se debe verificar que el software reciba y envíe la información correctamente, para ello es necesario inicializar estos elementos externos primero, dígase la fuente de video, para este caso la proveniente del teléfono celular y el puerto serie, que corresponde al enlace con el módulo bluetooth.
- 2) Considerar el uso de terminales que mantengan su posición a pesar de los movimientos que se puedan presentar durante la manipulación, así, se puede soldar los que sean necesarios.
- 3) Realizar las pruebas en entornos con la iluminación y fondos adecuados, para mejorar la precisión de los algoritmos de detección y seguimiento.
- 4) Se puede mejorar el software con nuevas técnicas de detección y seguimientos que aparezcan, como las de machine learning.
- 5) Diseñar un equipo más compacto, utilizando componentes más reducidos o quizás con una mejor disposición de estos, así mismo el chasis se puede optimizar reduciendo espacios libres.
- 6) Para Mantener el celular fijo al soporte, es importante colocar un adhesivo o un gancho que permita la sujeción de este.

## BIBLIOGRAFÍA

Arduino. (05 de 02 de 2018). *What is Arduino?* Recuperado el 01 de 02 de 2021, de Arduino:

<https://www.arduino.cc/en/Guide/Introduction>

Arduino. (s.f.). *Arduino Nano*. Recuperado el 02 de 07 de 2021, de Arduino:

<https://store.arduino.cc/usa/arduino-nano>

Bolaños, R. (2013). *Detección, seguimiento y análisis de personas en streams de vídeo. (Tesis de pregrado)*. Universidad de las Palmas de Gran Canaria, Las Palmas de Gran Canaria, España. Recuperado el 13 de 11 de 2020, de

<https://accedacris.ulpgc.es/handle/10553/11234>

Components101. (13 de 04 de 2020). *L298N Motor Driver Module*. Recuperado el 05 de 07 de 2021, de Components101: <https://components101.com/modules/l293n-motor-driver-module>

Díaz, J., García, A., & Ríos, E. (2009). IMPLEMENTACION DE UNA MODULACION PWM OPTIMIZADA PARA EL CONTROL DE UN MOTOR TRIFASICO DE INDUCCION USANDO LA TARJETA DSP - TMS320C6416 DSK -. *Revista Colombiana de Tecnologías de Avanzada*, 1(13), 57-65. Recuperado el 02 de 07 de 2021, de [http://www.unipamplona.edu.co/unipamplona/portallG/home\\_40/recursos/03\\_v13\\_18/revista\\_13/04112011/08.pdf](http://www.unipamplona.edu.co/unipamplona/portallG/home_40/recursos/03_v13_18/revista_13/04112011/08.pdf)

ElectroPro. (s.f.). *Motoreductor 16RPM 12VDC 37GB/12V/16RPM*. Recuperado el 02 de 07 de 2021, de Electro Pro:

[https://electropro.pe/index.php?route=product/product&product\\_id=740](https://electropro.pe/index.php?route=product/product&product_id=740)

- Esqueda, J., & Palafox, L. (2005). *Fundamentos para el procesamiento de imágenes*. México: Universidad Autónoma de Baja California.
- Gómez, M. (2017). *Historia(s) de la Imagen Digital*. Recuperado el 19 de 01 de 2020, de Interartive: <https://interartive.org/2017/04/historias-de-la-imagen-digital-marisa-gomez>
- González, J. (2014). *Curso Completo de Electrónica Digital*. Madrid: Universidad Pontifica de Salamanca en Madrid. Recuperado el 01 de 02 de 2021, de [http://www.edudevices.com.ar/download/articulos/digitales/Cur\\_dig\\_01.pdf](http://www.edudevices.com.ar/download/articulos/digitales/Cur_dig_01.pdf)
- Herrera, J., Tlízaliturri, I., & Morales, M. (2006). Unidad Básica de Comunicación Serial en un Microcontrolador. (I. P. Nacional, Ed.) *Polibits*(33), 3-7. Recuperado el 01 de 02 de 2021, de <https://www.redalyc.org/articulo.oa?id=402640446001>
- Inafuku, A. (2015). Diseño de un algoritmo de estabilización de video orientado a la detección de personas. (*Tesis de fin de pregrado*). Pontificia Universidad Católica Del Perú, Lima, Perú. Recuperado el 13 de 11 de 2020, de <http://hdl.handle.net/20.500.12404/6036>
- Kaur, R. (2016). Tutorial on Minimum Output Sum of Squared Error Filter. *Tesis de Maestría*. Colorado State University, Fort Collins, Colorado, USA. Recuperado el 07 de 07 de 2021, de [https://mountainscholar.org/bitstream/handle/10217/173486/Sidhu\\_colostate\\_0053N\\_13486.pdf](https://mountainscholar.org/bitstream/handle/10217/173486/Sidhu_colostate_0053N_13486.pdf)
- Llamas, L. (09 de 06 de 2016). *DETECTOR DE LÍNEAS CON ARDUINO Y SENSOR TCRT5000L*. Recuperado el 06 de 04 de 2022, de Luis Llamas: <https://www.luisllamas.es/arduino-detector-lineas-tcrt5000l/>
- Martín, J., & García, M. (2016). *Motores eléctricos (Automatismos industriales)*. Madrid: Editex.
- Martínez, F., & Martín, G. (2003). *Introducción a la programación estructurada en C*. Valencia: Maite Simon.

- Miranda, M. (06 de 2009). *La imagen digital*. Recuperado el 19 de 01 de 2021, de Scielo:  
[http://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S0016-35032009000200016](http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0016-35032009000200016)
- Naylamp Mechatronics SAC. (s.f.). *DRIVER PUENTE H L298N 2A*. Recuperado el 02 de 07 de 2021, de Naylamp Mechatronics: <https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html>
- Naylamp Mechatronics SAC. (s.f.). *MÓDULO BLUETOOTH HC06*. Recuperado el 02 de 07 de 2021, de Naylamp Mechatronics: <https://naylampmechatronics.com/inalambrico/24-modulo-bluetooth-hc06.html>
- OpenCV. (2020). *About OpenCV*. Recuperado el 13 de 11 de 2020, de OpenCV:  
<https://opencv.org/about/>
- OpenCV. (2021). *Cascade Classifier*. Recuperado el 01 de 02 de 2021, de OpenCV:  
[https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)
- Palacín, M. (2011). Visión artificial aplicada al monitoreo automatizado del proceso de cloración para mejorar la calidad del agua. (*Tesis de pregrado*). Universidad Católica Santo Toribio de Mogrovejo, Chiclayo. Recuperado el 23 de 11 de 2020, de <http://hdl.handle.net/20.500.12423/536>
- Passariello, G., & Mora, F. (1995). *Imágenes médicas. Adquisición, Análisis* (1 ed.). Venezuela: Universidad Simón Bolívar.
- Piscoya, J. (2019). Sistema de visión artificial para apoyar en la identificación de plagas y enfermedades del cultivo de sandía en el distrito de Ferreñafe. (*Tesis de pregrado*). Universidad Católica Santo Toribio de Mogrovejo, Chiclayo. Recuperado el 23 de 11 de 2020, de <http://hdl.handle.net/20.500.12423/2356>
- Programo Ergo Sum. (s.f.). *Salidas analógicas PWM con Arduino*. Recuperado el 02 de 07 de 2021, de Programo Ergo Sum: <https://www.programoergosum.com/cursos-online/arduino/255-salidas-analogicas-pwm-con-arduino/salidas-analogicas-pwm>

- Rodríguez, J. (2018). Seguimiento autónomo de una persona con marcador visualmente identificada por un dron. (*Tesis de pregrado*). Universidad del País Vasco, País Vasco, España. Recuperado el 13 de 11 de 2020, de <http://hdl.handle.net/10810/29491>
- Somolinos, J. (2002). *Avances en robótica y visión por computador*. Univ de Castilla La Mancha.
- Taquía, J. (22 de 12 de 2017). El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital. *INTERFASES*(10), 11-29.  
doi:<https://doi.org/10.26439/interfases2017.n10>
- Tecno&Logía. (02 de 07 de 2021). *Montaje de un puente en H para el control de motores*.  
Obtenido de Tecno&Logía: <http://tecnoandlogia.blogspot.com/2012/06/montaje-de-un-puente-en-h-para-el.html>
- Tolentino, Á. (2017). Histograma de orientación de gradientes aplicado al seguimiento múltiple de personas basado en video. (*Tesis de Maestría*). Pontificia Universidad Católica del Perú, Lima, Perú. Recuperado el 13 de 11 de 2020, de <http://hdl.handle.net/20.500.12404/8323>
- UNIT Electronics. (s.f.). *Modulo Bluetooth HC-06*. Recuperado el 05 de 07 de 2021, de UNIT Electronics: <https://uelectronics.com/producto/modulo-bluetooth-hc-06/>
- Valdes, F., & Pallás, R. (2007). *Microcontroladores Fundamentos y Aplicaciones con PIC*. España: Marcombo.
- Vildósola, E. (2021). Actuadores. *Actuadores*. Soltex Chile S.A., Chile. Recuperado el 09 de 02 de 2021, de <http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>
- Wordreference. (s.f.). *software*. Recuperado el 13 de 01 de 2020, de Wordreference: <https://www.wordreference.com/definicion/software>

## **ANEXOS**

## ANEXO 1. CÓDIGO DEL SOFTWARE EN PC

```
import cv2
import serial
import sys

# Carga del filtro para deetcción
face_cascade =
cv2.CascadeClassifier('data/haarcascades/haarcascade_frontalface_alt2.xml')

# recepción del video
cap = cv2.VideoCapture(0)

frame_rate_calc = 1
freq = cv2.getTickFrequency()
aux = 0

#Solicitud del puerto serial
puertocom = (input("Puerto: "))
ser = serial.Serial(puertocom, 9600, timeout=3)

def detecta(img):
    # Conversión a escalas grises para el procesado
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Detección de rostros
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)
    # Enmarcar el rostro en un rectángulo
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
        return (x, y, w, h)

#función que enmarca en un recuadro el elemento a seguir
def drawBox(img, bbox):
    x,y,w,h = int(bbox[0]),int(bbox[1]),int(bbox[2]),int(bbox[3])
    cv2.rectangle(img, (x,y), ((x+int(w/2)), (y+int(h/2))),
(255,0,255),3,1)
    cv2.putText(img, "Tracking", (75,75), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0,0,255),2)

#aplica la función MOSSE para el tracking
tracker = cv2.TrackerMOSSE_create()

while True:
    # Lectura del fram de video
    t1 = cv2.getTickCount()
    _, img = cap.read()

    #actualiza la posición del tracking
    success, bbox = tracker.update(img)

    if success and (aux < 50):
        drawBox(img, bbox)
```

```

        x1,y1,w1,h1 =
int(bbox[0]),int(bbox[1]),int(bbox[2]),int(bbox[3])
        valor= "H"+ str(x1+(w1//2))+ "V"+str(int(y1+(h1/2)))
        #envío de la coordenada del tracking
        ser.write(valor.encode())
        ser.write(b'\n')

    else:
        #en caso se pierde el objetivo, se imprime un mensaje y
        comienza a buscar
        cv2.putText(img, "Lost", (75,75), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0,0,255), 2)
        bbox = detecta(img)
        aux = 0

    try:
        x1,y1,w1,h1 =
int(bbox[0]),int(bbox[1]),int(bbox[2]),int(bbox[3])
        valor= "H"+ str(x1+(w1//2))+ "V"+str(int(y1+(h1/2)))
        #envío de la coordenada de la detección
        ser.write(valor.encode())
        ser.write(b'\n')
        tracker = cv2.TrackerMOSSE_create()
        tracker.init(img,bbox)

    except:
        pass

    #muestra los FPS calculados
    cv2.putText(img, 'FPS:
{0:.2f}'.format(frame_rate_calc), (30,50), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255,0,0), 1, cv2.LINE_AA)

    # Muestra ventanade video
    cv2.imshow('img', img)
    #función de salida del software
    k = cv2.waitKey(30) & 0xff
    t2 = cv2.getTickCount()
    time1 = (t2-t1)/freq
    frame_rate_calc= 1/time1
    if k==27:
        exit()
        ser.close()
        break
    aux += 1

cap.release()

```

## ANEXO 2. CÓDIGO DEL ARDUINO NANO

```
#define mot_h      7
#define mot_v      8
#define pin_pwm_h  9
#define pin_pwm_v 10

//variables para almacenar valores de sensores y posicion

unsigned int pos_h=320, pos_v=160;
unsigned int set_point_h=360, set_point_v=300; //centro de la imagen

/// variables para el pid
long derivativoh=0, proporcionalh=0, integralh=0; //errores
int salida_pwmh=0, proporcional_pasadoh=0;
long derivativov=0, proporcionalv=0, integralv=0; //errores
int salida_pwmv=0, proporcional_pasadov=0;

int velocidad=150; //variable para la velocidad, el
maximo es 255
float Kp=0.7, Ki=0.001, Kd=0.5; //constantes

String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete

int va=0;
String sva = ""; //tiempo lado A
String svb = ""; //timepo lado B
////////////////////////////////////
void setup()
{
  pinMode(mot_h, OUTPUT); //pin de direccion motor izquierdo
  pinMode(mot_v, OUTPUT); //pin de direccion motor derecho
  pinMode(pin_pwm_h, OUTPUT); //pin de direccion motor izquierdo
  pinMode(pin_pwm_v, OUTPUT); //pin de direccion motor derecho
  // initialize serial:
  Serial.begin(9600);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
}

void loop()
{
  if (stringComplete) {
    pos_h= sva.toInt();
    pos_v= svb.toInt();
    stringComplete = false;
  }
  if(Serial.available()){
    ////VERIFICA EL MOTOR DE LA BASE, DESPLAZAMIENTO HORIZONTAL
```

```

        pid(pos_h, velocidad, Kp, Ki, Kd, 1, set_point_h); //funcion para
algoritmo pid
/////VERIFICA EL MOTOR DE LA BASE, DESPLAZAMIENTO VERTICAL
        pid(pos_v, velocidad, Kp, Ki, Kd, 0, set_point_v);

    delay(500);
}else{
    motores(0,1);
    motores(0,0);
}

}

//////////funciones para el control/////
void pid(int pos, int velocidad, float Kp, float Ki, float Kd, bool
motor, int set_point)
{

    if(motor){
        proporcionalh = (pos) - set_point;
        integralh=integralh + proporcional_pasadoh; //obteniendo
integral
        derivativoh = (proporcionalh - proporcional_pasadoh);
//obteniedo el derivativo
        if (integralh>10000) integralh=10000; //limitamos la integral
para no causar problemas
        else if (integralh<-10000) integralh=-10000;
        salida_pwmh = ( proporcionalh * Kp ) + ( derivativoh * Kd
)+(integralh*Ki);

        if ( salida_pwmh > velocidad ) salida_pwmh = velocidad;
//limitamos la salida de pwm
        if ( salida_pwmh < -velocidad ) salida_pwmh = -velocidad;

        motores(salida_pwmh, motor);

        proporcional_pasadoh = proporcionalh;
    }else{
        proporcionalv = (pos) - set_point;
        integralv=integralv + proporcional_pasadov; //obteniendo
integral
        derivativov = (proporcionalv - proporcional_pasadov);
//obteniedo el derivativo
        if (integralv>10000){
            integralv=1000; //limitamos la integral para no causar
problemas
        }else if (integralv<-10000){
            integralv=-10000;
        }
        salida_pwmv = ( proporcionalv * Kp ) + ( derivativov * Kd
)+(integralh*Ki);

        if ( salida_pwmv > velocidad ) salida_pwmv = velocidad;
//limitamos la salida de pwm
        if ( salida_pwmv < -velocidad ) salida_pwmv = -velocidad;

```

```

        motores(salida_pwm_v, motor);

        proporcional_pasado_v = proporcional_v;
    }
}

//función para control de motores
void motores(int vel, bool motor)
{
    if(motor){    ///activar motor 1 -h - base
        if ( vel >= 0 )
        {
            digitalWrite(mot_h,HIGH); // con high avanza
            if(vel < 150 && vel>0){
                analogWrite(pin_pwm_h,0);
                delay(10);
            }

            analogWrite(pin_pwm_h,255-vel); //se controla de manera
                                                //inversa para mayor
control
        }
        else
        {
            digitalWrite(mot_h,LOW); //con low retrocede
            vel = vel*(-1); //cambio de signo
            if(vel < 150 && vel>0){
                analogWrite(pin_pwm_h,255);
                delay(10);
            }
            analogWrite(pin_pwm_h,vel);
        }
    }else{ ///activar motor 0 -v - celular
        if ( vel >= 0 )
        {
            digitalWrite(mot_v,HIGH);
            if(vel < 180 && vel>0){
                analogWrite(pin_pwm_v,0);
                delay(20);
            }
            analogWrite(pin_pwm_v,255-vel);
        }
        else
        {
            digitalWrite(mot_v,LOW);
            vel= vel*(-1);
            if(vel < 180 && vel>0){
                analogWrite(pin_pwm_v,255);
                delay(20);
            }
            analogWrite(pin_pwm_v,vel);
        }
    }
}

```

```

    }
}

void serialEvent() {
    while (Serial.available()) {

        char inChar = (char)Serial.read();

        if (inChar == '\n') {
            stringComplete = true;
            va=0;
        }
        if (!(stringComplete)) {
            if(inChar=='H'){
                va=1;
                sva="";
                inChar='0';    //"0";
            }else if(inChar=='V'){
                va=2;
                svb="";
                inChar='0';    //"0";
            }

            if (va==1){
                sva += inChar;
            }else if(va==2){
                svb += inChar;
            }

        }
    }
}

```

### ANEXO 3. ESPECIFICACIONES TÉCNICAS

CARACTERÍSTICA	VALOR
POTENCIA REPOSO	0.264 W
POTENCIA MOVIMIENTO	0.430W
VOLTAJE NOMINAL	3.3V
VOLTAJE CARGA	5V
CORRIENTE MÁXIMA DE TRABAJO	0.15A
PROTOCOLO DE COMUNICACIÓN	Bluetooth v2.0 / Serial UART
ALCANCE RECOMENDADO	10m
DIMENSIONES	12mm x 18mm x 27.5mm



## ACTA DE SUSTENTACIÓN VIRTUAL N° 002-2023-D/FACFyM

Siendo las 10:00 am del día 10 de enero del 2023, se reunieron vía plataforma virtual, meet.google.com/oao-zrei-muc, los miembros del jurado evaluador de la Tesis titulada:

“SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE PERSONAS PARA PONENTES EN LA UNPRG”

Designados por Resolución N° 031-2020-VIRTUAL-UI/FACFyM de fecha 05 de diciembre de 2020

Con la finalidad de evaluar y calificar la sustentación de la tesis antes mencionada, conformada por los siguientes docentes:

M.Sc, Ing. Hugo Javier Chiclayo Padilla	Presidente
Mg. Ing. Martín Augusto Nombera Lossio	Secretario
Dra. Ing. Lucía Isabel Chamán Cabrera	Vocal

La tesis fue asesorada por el Mg. Ing. Oscar Uchelly Romero Cortez, nombrado por Resolución N° 031-2020-VIRTUAL-UI/FACFyM de fecha 05 de diciembre de 2020.

El Acto de Sustentación fue autorizado por Resolución N° 1043-2022-VIRTUAL-D/FACFyM de fecha 27 de diciembre 2022.

La Tesis fue presentada y sustentada por el Bachiller en Ingeniería Electrónica, ACEDO TEN JUAN PABLO, y tuvo una duración de 30 minutos.

Después de la sustentación, y absueltas las preguntas y observaciones de los miembros del jurado se procedió a la calificación respectiva, otorgándole el Calificativo de 18 (Dieciocho) en la escala vigesimal, mención Muy Bueno.

Por lo que queda apto para obtener el Título Profesional de **Ingeniero Electrónico**, de acuerdo con la Ley Universitaria 30220 y la normatividad vigente de la Facultad de Ciencias Físicas y Matemáticas y la Universidad Nacional Pedro Ruiz Gallo.

Siendo las 10:45 am se dio por concluido el presente acto académico, dándose conformidad al presente acto con la firma de los miembros del jurado.

M.Sc, Ing. Hugo Javier Chiclayo Padilla  
Presidente

Mg. Ing. Martín Augusto Nombera Lossio  
Secretario

Dra. Ing. Lucía Isabel Chamán Cabrera  
Vocal

Mg. Ing. Oscar Uchelly Romero Cortez  
Asesor



## CONSTANCIA DE VERIFICACIÓN DE ORIGINALIDAD

Yo, OSCAR UCHELLY ROMERO CORTEZ, usuario revisor del documento titulado: **SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE PERSONAS PARA PONENTES EN LA UNPRG.**

Cuyo autor es, Bach. Acedo Ten Juan Pablo. Identificado con Documento de Identidad 70031226; declaro que la evaluación realizada por el Programa Informático, ha arrojado un porcentaje de similitud de 7%, verificable en el Resumen de Reporte automatizado de similitudes que se acompaña.

El suscrito analizó dicho reporte y concluyó que cada una de las coincidencias detectadas dentro del porcentaje de similitud permitido no constituyen plagio y que el documento cumple con la integridad científica y con las normas para el uso de citas y referencias establecidas en los protocolos respectivos.

Se cumple con adjuntar el Recibo Digital a efectos de la trazabilidad respectiva del proceso.

Lambayeque, 13 de setiembre de 2022.



---

Mg. Oscar Uchelly, Romero Cortez  
DNI: 41812294  
ASESOR

Adj.

1. Recibo Digital Turnitin firmado.
2. Vista actual con Informe de Originalidad Turnitin firmados



## CONSTANCIA DE SIMILITUD N° 19-2023- VIRTUAL-UI-FACFyM

El que suscribe, director de la Unidad de Investigación de la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruiz Gallo, hace constar:

Que, el Bachiller **ACEDO TEN JUAN PABLO**, de la Escuela Profesional de **INGENIERÍA ELECTRÓNICA**, ha cumplido con presentar la **SIMILITUD DE ORIGINALIDAD DE LA TESIS (TURNITIN)**, como requisito indispensable para la sustentación de la tesis, según detalle:

- **TÍTULO DE LA TESIS:** "SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE PERSONAS PARA PONENTES EN LA UNPRG"

- **ÍNDICE DE SIMILITUD:** 07 %

- **ASESOR:** Mg. Ing. OSCAR UCHELLY ROMERO CORTEZ.

Se expide la presente constancia, para la tramitación del Título Profesional, dispuesto en la Directiva para la evaluación de originalidad de los documentos académicos, de investigación formativa y para la obtención de Grados y Títulos de la UNPRG.

Lambayeque, 16 de febrero de 2023

**Dr. WALTER ARRIAGA DELGADO**  
**DIRECTOR - UNIDAD DE INVESTIGACIÓN**




## Recibo digital

Este recibo confirma que su trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Juan Pablo Acedo Ten  
Título del ejercicio: SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE...  
Título de la entrega: SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE...  
Nombre del archivo: Tesis-Completa\_2.docx  
Tamaño del archivo: 7.32M  
Total páginas: 81  
Total de palabras: 10,852  
Total de caracteres: 60,448  
Fecha de entrega: 13-sept.-2022 09:12p. m. (UTC-0500)  
Identificador de la entre... 1899302603



# SISTEMA DE DETECCIÓN Y SEGUIMIENTO DE LA POSICIÓN DE PERSONAS PARA PONENTES EN LA UNPRG

## INFORME DE ORIGINALIDAD



## FUENTES PRIMARIAS

<b>1</b>	<b>hdl.handle.net</b> Fuente de Internet	<b>2%</b>
<b>2</b>	<b>repositorio.unan.edu.ni</b> Fuente de Internet	<b>1%</b>
<b>3</b>	<b>docplayer.es</b> Fuente de Internet	<b>1%</b>
<b>4</b>	<b>www.akizi.com</b> Fuente de Internet	<b>&lt;1%</b>
<b>5</b>	<b>Submitted to Politécnico Colombiano Jaime Isaza Cadavid</b> Trabajo del estudiante	<b>&lt;1%</b>
<b>6</b>	<b>Submitted to Universidad Nacional de Frontera</b> Trabajo del estudiante	<b>&lt;1%</b>
<b>7</b>	<b>www.learobotics.com</b> Fuente de Internet	<b>&lt;1%</b>
<b>8</b>	<b>electropro.pe</b> Fuente de Internet	<b>&lt;1%</b>

9	Submitted to Universidad Nacional Pedro Ruiz Gallo Trabajo del estudiante	<1 %
10	creativecommons.org Fuente de Internet	<1 %
11	Submitted to Universidad Manuela Beltrán Virtual Trabajo del estudiante	<1 %
12	repositorio.espe.edu.ec Fuente de Internet	<1 %
13	repositorio.ug.edu.ec Fuente de Internet	<1 %
14	repositorio.unprg.edu.pe Fuente de Internet	<1 %
15	rraae.cedia.edu.ec Fuente de Internet	<1 %

Excluir citas      Activo  
Excluir bibliografía      Activo

Excluir coincidencias < 15 words

