



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

**IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO
FACIAL CON VISIÓN ARTIFICIAL EN EL ROBOT BIOLOID GP**

Trabajo de titulación previo a la obtención del

Título de Ingeniero Electrónico

AUTORES:

Carlos Iván Menéndez Chávez

Jorge Daniel Moncada Ramírez

TUTOR:

MSc. Mónica Miranda

Guayaquil – Ecuador

2021

CERTIFICADOS DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **CARLOS IVAN MENÉNDEZ CHÁVEZ** con documento de identificación N° 0926122250 y **JORGE DANIEL MONCADA RAMÍREZ** con documento de identificación N° 0951824143; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

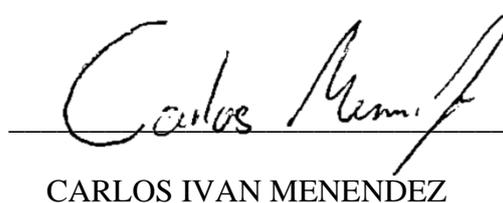
Guayaquil, 27 de enero del año 2023

Atentamente,



JORGE DANIEL MONCADA

C.I.: 0951824143



CARLOS IVAN MENENDEZ

C.I.: 0926122250

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITECNICA SALESIANA**

Nosotros, **JORGE DANIEL MONCADA RAMÍREZ**, con documento de identificación N° 0951824143, y **CARLOS IVAN MENENDEZ CHAVEZ**, con documento de identificación N° **0926122250**, expresamos nuestra voluntad y por medio del presente documento cedemos a la **UNIVERSIDAD POLITECNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado “**IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL CON VISIÓN ARTIFICIAL EN EL ROBOT BIOLOID GP**”, el cual ha sido desarrollado para optar por el título de: **INGENIERO ELECTRÓNICO**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

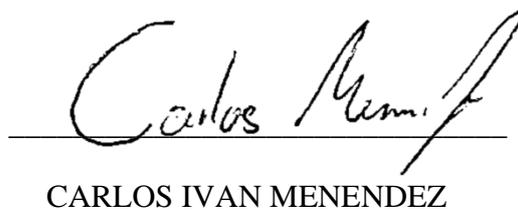
Guayaquil, 27 de enero del año 2023

Atentamente,



JORGE DANIEL MONCADA

0951824143



CARLOS IVAN MENENDEZ

0926122250

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, MONICA MARÍA MIRANDA RAMOS con documento de identificación N° , docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “**IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL CON VISIÓN ARTIFICIAL EN EL ROBOT BIOLOID GP**”, realizado **MENÉNDEZ CHÁVEZ CARLOS IVAN LUQUE** con documento de identificación N° 0926122250 y **JORGE DANIEL MONCADA RAMÍREZ** con documento de identificación N° 0951824143, obteniendo como resultado final el trabajo de titulación bajo la opción trabajo de grado que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 27 de enero del año 2023

Atentamente,

A handwritten signature in blue ink that reads "Mónica Miranda". The signature is written in a cursive style and is underlined with a blue horizontal line.

ING. MÓNICA MIRANDA RAMOS

0917271785

DEDICATORIA

Este proyecto se lo dedico a mis padres que han sido mi apoyo, impulso y motor para culminar mi carrera y ser un profesional, a mi madre MARTHA YADIRA CHÁVEZ FERNANDEZ por su amor infinito y por su apoyo incondicional; a mi padre CARLOS IVÁN MENÉNDEZ MENENSES por ser el mejor ejemplo de padre y amigo, y a mi prometida KATHERINE NAHOMY REMACHE LUQUE por estar a mi lado en todo momento y llenarme de fuerzas para lograr culminar este gran paso de mi carrera profesional.

Carlos Iván Menéndez Chávez

DEDICATORIA

Este logro está dedicado a mis padres Javier y Cecilia quienes, con su amor, paciencia y esfuerzo me han permitido cumplir con el objetivo, gracias por inculcar el ejemplo de esfuerzo, de no temer las adversidades porque Dios está conmigo siempre.

Jorge Daniel Moncada Ramírez

AGRADECIMIENTO

Quiero agradecer a Dios por darme salud, fortaleza y forjar mi camino.

A mi entrañable familia por el apoyo constante, por sus palabras de aliento, por creer en mí y en mis capacidades de seguir adelante pese a toda adversidad porque todo se puede lograr con sacrificio, amor, dedicación y esfuerzo.

Carlos Ivan Menéndez Chávez

AGRADECIMIENTO

Primero quiero agradecerle a Dios, que con su bendición me llena siempre mi vida y a toda mi familia por estar siempre presentes.

Mi profundo agradecimiento a todas las autoridades de la Universidad Politécnica Salesiana, por confiar en mí y abrirme las puertas

De igual manera, a todos mis amigos y futuros colegas que me ayudaron de una manera desinteresada y buena voluntad.

Jorge Daniel Moncada Ramírez

RESUMEN

A lo largo del periodo de duración de la carrera de Ingeniería Electrónica, los estudiantes hemos tenido la necesidad de tener instrumentos o herramientas que contribuyan a la práctica o implementación de los conocimientos adquiridos o proporcionados en la universidad. Como objetivo de este proyecto de titulación, es dar, incentivar y motivar a los alumnos en la utilización y evolución de los robots humanoides por medio de dispositivos de comunicación inalámbrica como de visión artificial. [1]

Para ello se utilizó un robot humanoide Bioloid GP, con el cual se realizó las investigaciones de uso y desarrollo de este, con el cual se logró desplegar varios algoritmos, realizando el uso del programa Roboplus y describiendo un manual con seis prácticas para brindar una guía a los futuros estudiantes que deseen implementar, avanzar, seguir y plasmar sus capacidades en la rama de la robótica y de la programación.

Parte de ello, es la iniciativa de plasmar en un robot Bioloid GP la implementación de un sistema de reconocimiento facial con visión artificial.

ABSTRACT

Throughout the duration of the Electronic Engineering career, students have had the need to have instruments or tools that allow them to put into practice or implement the knowledge acquired, as the purpose of this degree project is to contribute, encourage and motivate students in the use and evolution of humanoid robots through wireless communication devices such as artificial vision.

For this, a humanoid Bioloid GP robot was used, with which it was possible to deploy several algorithms, making the use of the Roboplus program and describing a manual with six practices to provide a guide to future students who wish to implement, advance, follow and capture their capabilities in the field of robotics and technolog programming.

Part of this is the initiative to translate the implementation of a facial recognition system with artificial vision into a Bioloid GP robot.

ÍNDICE

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	IV
DEDICATORIA	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
AGRADECIMIENTO	VIII
RESUMEN	IX
INTRODUCCIÓN	5
1.1. PROBLEMA	7
1.2. ANTECEDENTES	7
1.3. IMPORTANCIA Y ALCANCES	8
1.4. DELIMITACIÓN	8
1.4.1. Delimitación Temporal:	8
1.4.2. Delimitación Espacial:	8
1.4.3. Delimitación Académica:	8
1.5. EXPLICACIÓN DEL PROBLEMA	9
1.6. OBJETIVOS	9
1.5.1. Objetivo General:	9
1.5.2. Objetivo Específico:	9
ESTADO DEL ARTE.....	11
2.1. ESTADO DEL ARTE	11
2.2.1. Robot Bioloid GP	11

2.2.2.	Estructura de los Robots Humanoides	13
2.2.3.	Servomotores	14
2.2.4.	Características del servo AX-18^a	15
2.2.5.	Asignación de pines	15
2.2.6.	Alambrado	16
2.2.7.	Enlace de pc	16
2.2.8.	Características del servo AX-12A	17
2.2.9.	Sensores	17
2.2.10.	Sensor Gyro GS-12	18
2.2.11.	BT-410 Dongle converts USB 2.0	19
2.2.12.	Lithium polymer battery 1000mah	19
2.2.13.	Controlador CM-530	20
2.2.14.	Medio de Comunicación del Bioloid GP	21
2.2.15.	Raspberry Pi 4	23
2.2.16.	Raspberry Pi Camera Module V2 8MP	24
2.2.	SOFTWARES UTILITARIOS	24
2.2.1.	Software Roblopus	25
2.2.2.	Software VNC viewer	27
MARCO METODOLÓGICO		28
3.1.	FASE INICIAL	28
3.1.1.	Estructura	30
3.2.	ALGORITMOS DE RECONOCIMIENTO FACIAL	45

3.2.1 Librería OpenCV.....	45
3.2.2 Reconocimiento facial con OpenCV	46
3.2.3 Librería Face Recognition	46
3.2.4 Eigenfaces (PCA).....	47
3.2.5 Fisherfaces (LDA).....	47
3.2.6 Histogramas de patrones binarios locales (LBPH).....	48
3.2.7 Cascadas de Haar	50
3.3. INTERFAZ GRÁFICA.....	51
VERIFICACIÓN Y FUNCIONAMIENTO	53
4.1. RESULTADOS.....	53
CONCLUSIONES	68
RECOMENDACIONES.....	69
REFERENCIAS BIBLIOGRÁFICAS.....	70
ANEXOS A: MANUAL DE PRÁCTICAS DE ROBOT HUMANOIDE	
BIOLOID GP	73

ÍNDICE DE FIGURAS

Fig. 1 Componentes del Robot Bioloid GP	12
Fig. 2 Robot Bioloid GP (armado).....	13
Fig. 3 Diagrama del servomotor	14
Fig. 4 Servomotores dynamixel AX-18A y AX-12 ^a	15
Fig. 5 Asignaciones de pines	16
Fig. 6 Alambrado de conexión.....	16
Fig. 7 Enlace a un computador	17
Fig. 8 Descripción del giroscopio	18
Fig. 9 BT-410 Dongle converts USB 2.0.....	19
Fig. 10 Batería de Litio	19
Fig. 11 Placa controlador CM-530	20
Fig. 12 Controlador CM-530 y sus partes.....	21
Fig. 13 Modulo Zig-100/110-A	21
Fig. 14 Funcionamiento de ZigBee.....	22
Fig. 15 Placa Raspberry Pi 4.....	23
Fig. 16 Raspberry Pi Camera Module V2 8MP	24
Fig. 17 Interfaz de Roboplus.....	25
Fig. 18 Imagen principal del Software Roboplus	25
Fig. 19 Imagen de Roboplus Task	26
Fig. 20 Imagen de Roboplus Manager.....	26
Fig. 21 Imagen de Roboplus Motion	27
Fig. 22 Imagen del software VNC	27
Fig. 23 Tipos de configuraciones del robot Bioloid	28
Fig. 24 Kit del robot Bioloid.....	29

Fig. 25 Interfaz principal del programa roboplus	30
Fig. 26 Pasos para sujeción de marco metálico en los servos.....	31
Fig. 27 Resultado del ensamblaje	31
Fig. 28 Colocación de tuercas en los servomotores ID 1 e ID2.....	32
Fig. 29 Colocación servomotores en la estructura metálica	32
Fig. 30 Fijación de marcos metálicos en los ejes de los servomotores.....	33
Fig. 31 Preparación de servomotores para brazos	34
Fig. 32 Fijación de servomotores que conformaran los brazos del robot	34
Fig. 33 Estructura metálica para las manos	35
Fig. 34 Unión de brazos con el pecho.....	35
Fig. 35 Colocación del giroscopio a la estructura.....	36
Fig. 36 Partes metálicas correspondientes a los pies	36
Fig. 37 Procedimiento para acoplar servos correspondientes a los pies.....	37
Fig. 38 Procedimiento para ensamblaje de los tobillos	37
Fig. 39 Procedimiento para juntar tobillo con el pie.....	38
Fig. 40 Procedimiento para juntar tobillo con el pie.....	38
Fig. 41 Procedimiento para acoplar las rodillas con las piernas	39
Fig. 42 Unión de rodillas, pies y piernas	39
Fig. 43 Acoplar piernas juntas al cuerpo	40
Fig. 44 Separador y base de la batería	40
Fig. 45 Sujeción del controlador CM-530 en la estructura.....	41
Fig. 46 Conexión de los cables 10 y 18 en los servos correspondientes	42
Fig. 47 Conexiones de los cables de servomotores del robot	42
Fig. 48 Organización de los cables del robot.....	43
Fig. 49 Organización de los cables del robot entre sus diferentes partes	43

Fig. 50 Acople de la cabeza al cuerpo del robot.....	44
Fig. 51 Colocación de la batería del robot.....	44
Fig. 52 Esquema de Algoritmos Open CV	46
Fig. 53 Muestra del procesamiento de imágenes.....	49
Fig. 54 Procesamiento de imágenes.....	51
Fig. 55 GUI de la Implementación del sistema	51
Fig. 56 Ventana del sistema.....	52
Fig. 57 Encendido del sistema y captura del rostro	52
Fig. 58 Aviso de Almacenamiento	53
Fig. 59 Vista de ventana de entrenamiento.....	53
Fig. 60 Ensamblaje del robot humanoide.	54
Fig. 61 Esquema del Algoritmo Open CV.....	54
Fig. 62 Despligue del software	55
Fig. 63 Robot en funcionamiento	55

ÍNDICE DE TABLAS

Tabla I.....	49
--------------	----

INTRODUCCIÓN

El ser humano con el paso del tiempo ha ido evolucionando y con ello su entorno, al punto de crear objetos que satisfagan sus necesidades. [2]

La idea de poder replicar las funcionalidades del movimiento del ser humano en una máquina ha capturado la mente y la imaginación de los individuos por décadas. [3]

Los robots juegan un papel bastante fundamental en la enseñanza ya hace varios años. Siendo empleados como herramientas de apoyo en la educación, debido a que su presencia estimula a los alumnos. Los usos de la robótica resultan muy amplios y variados. Se muestra que, la iniciativa de incorporar la robótica en la enseñanza como un apoyo, donde se implementó el primer lenguaje de programación educativo denominado logos, en el laboratorio del Instituto tecnológico de Massachusetts. [1]

Los robots humanoides están firmemente arraigados en el imaginario de la sociedad actual y su estudio presenta una amplia gama de retos para investigadores y desarrolladores, pero esta dificultad se ve recompensada por el acercamiento cada vez mayor hacia esa situación de ciencia ficción de humanos conviviendo con robots. [3]

En el estado actual de la tecnología los robots humanoides se ubican principalmente en ámbitos académicos para estudios y desarrollo de nuevas tecnologías, de aquí la importancia de la presente elaboración de la Implementación de un sistema de reconocimiento facial con visión artificial en el robot Bioloid GP, el cual nos permite implementar los conocimientos que se van adquiriendo teóricamente a tal punto de visualizar y ser más interactiva la educación que se está recibiendo.

La motivación a solucionar la necesidad de este proyecto será descrita estructuralmente en el Capítulo 1, a través del planteamiento del problema, justificación, técnicas empleadas para el desarrollo y beneficio e impacto que generaría el sistema.

Seguido del Capítulo 2, que describe teóricamente los componentes del robot y de los lineamientos considerados para el desarrollo de este.

En el Capítulo 3, se detalla el uso y proceso programable tanto de las placas controladoras como del aplicativo para cumplir con nuestro objetivo.

En el Capítulo 4, se detallan teórica y gráficamente las pruebas realizadas con respecto a las prácticas propuestas y establecidas en el anteproyecto. [2]

PLANTEAMIENTO DEL PROBLEMA

1.1. PROBLEMA

En la actualidad la carrera de ingeniería electrónica de la universidad Politécnica Salesiana - Sede Guayaquil cuenta con 6 robots humanoides; sin embargo, esta cantidad es pequeña y no cuenta con la adecuada capacidad de robots humanoides, esto incide en que no todos los estudiantes puedan poner en práctica los conocimientos adquiridos en su proceso de estudio con respecto a la robótica, uso, programación y manipulación del robot.

Además, es muy importante en las universidades abordar temas de investigación en el aula que permiten proponer actividades lúdicas con robots que sean educativos y que permitan desarrollar conceptualizaciones que se imparten en las diferentes materias y hasta que abordan problemas cotidianos relacionados con el adecuado uso de la tecnología.

1.2. ANTECEDENTES

En el ámbito académico e industrial las instituciones se esfuerzan por explicar las diferentes tecnologías y medios en el cual una persona puede crear o innovar cosas haciendo uso de nuevos conocimientos; sin embargo, en el área de la robótica si bien es cierto que es de gran utilidad tener en mente la parte teórica, los fundamentos y el porqué de las cosas también es importante el contar con los robots u herramientas necesarias el cual permita visualizar, implementar y desarrollar esos conocimientos adquiridos tales como reconocimiento facial mediante visión artificial.

Al utilizar un robot Bioloid Gp se busca beneficiar en un inicio a los estudiantes de Ingeniería Electrónica y Automatización relacionados al campo de Automatización Industrial, redes de telecomunicaciones y grupos investigativos para el Club de Robótica.

1.3. IMPORTANCIA Y ALCANCES

El desarrollo del proyecto tiene como fundamento principal solventar la necesidad de los estudiantes de la Universidad Politécnica Salesiana para que efectúen prácticas de programación de rutinas con los robots humanoides, mejorando la calidad y eficiencia de la educación y conocimientos impartidos en nuestra institución. Este proyecto demuestra y pone en práctica los conocimientos adquiridos como el esquema y programación de la interfaz gráfica, la misma que realizará la identificación de rostro haciendo uso del lenguaje de programación de Python. Así mismo se desarrollará un manual de prácticas para uso didáctico dentro de la universidad por el cual tendrá técnicas biométricas que incorporan en la adquisición de datos por medio de la cámara para incorporar la información de los aspectos faciales del sujeto y etapas de reconocimiento.

1.4. DELIMITACIÓN

1.4.1. Delimitación Temporal:

Este proyecto se consideró en realizarse en un tiempo de 6 meses a partir de la aprobación de este.

1.4.2. Delimitación Espacial:

La investigación y desarrollo del proyecto de titulación se llevó a cabo en la Universidad Politécnica Salesiana, que se encuentra en el campus centenario Robles 104 y Chambers para el Club de robótica y prácticas que se impartan en la carrera de ingeniería de la UPS.

1.4.3. Delimitación Académica:

EL proyecto planteado cumplió con los parámetros solicitados por la Universidad Politécnica Salesiana en su grado investigativo y modelo de presentación para proyectos

de titulación, utilizando los conocimientos adquiridos en materias como: Programación, Circuitos Eléctricos.

1.5. EXPLICACIÓN DEL PROBLEMA

A pesar de que contamos con algunas herramientas y robots con la tecnología para ayudarnos a implementar los conocimientos adquiridos en las diferentes materias de la carrera de ingeniería electrónica, fue demostrado que al tener un crecimiento en el interés de la robótica y demanda de estudiantes hace falta tener más prototipos y robots humanoides que sean de carácter educativo y programable para que los futuros estudiantes puedan visualizar y desarrollar sus capacidades en la robótica, programación y electrónica.

El proyecto realizado es una apertura a poder contar los prototipos necesarios para realizar esas prácticas en los laboratorios, investigaciones en las aulas o competencias de clubes de robótica.

1.6. OBJETIVOS

1.5.1. Objetivo General:

Ensamblar el robot Bioloid GP e implementar un sistema de reconocimiento facial con visión artificial.

1.5.2. Objetivo Específico:

1. Ensamblar la estructura del robot Bioloid GP.
2. Analizar los diferentes algoritmos de reconocimiento facial con la finalidad de obtener el más conveniente.

3. Seleccionar e implementar el algoritmo de reconocimiento del rostro para su uso en el sistema de reconocimiento.
4. Desarrollar la interfaz gráfica de usuario que facilite la comprensión de la visión computacional.
5. Planificar un manual de 6 practicas.

ESTADO DEL ARTE

2.1. ESTADO DEL ARTE

2.2.1. Robot Bioloid GP

El Bioloid GP es un robot humanoide creado para la educación de la robótica. Tiene motores de alta calidad. Viene en compañía con 18 servos dynamixel, sensores, un controlador, batería de lipo y composición de aluminio.

El primordial beneficio de Bioloid GP como robot humanoide programable, es que no resulta tan complejo de ordenar con su controlador CM-530. Sus manos de agarre además permiten que logre ser capaz de manipular objetos.

El robot tiene diversos sensores, como por ejemplo sensor de distancia por infrarrojos para evadir los muros y el sensor giroscópico que le ayuda a conservar su equilibrio.

El robot humanoide Bioloid GP, podría ser controlado con un mando a distancia combinado con el módulo ZigBee, tal responde a sus comandos.

La programación se hace por medio del programa RoboPlus, integrado con el kit del robot. Este programa, posibilita registrar los movimientos y de esta forma, automatizar ocupaciones de una forma fácil e intuitiva.

Este programa se hace en el controlador CM-530 y el robot puede actuar de manera autónoma. En la Fig. 1, se puede ver el robot Bioloid GP junto con sus elementos.



Fig. 1 Componentes del Robot Bioloid GP [1]

El Bioloid GP es un robot en especial elaborado para destacar en toda clase de competencias, como: carreras, acrobacias, equilibrio, toma de posesión: sus potentes servomotores, su sistema de corrección instantánea de la postura, su equilibrio particularmente impresionante y su estructura muy resistente lo convierten en un campeón de cualquier clase. Como se muestra en la Fig. 2 [1]

Características:

- El mejor robot humanoide con DYNAMIXEL AX-18F (piernas) de alta calidad
- Marcos de aluminio fuertes y livianos
- Excelente movilidad, como girar al caminar o caminar a alta velocidad

- Se proporcionan movimientos humanoides básicos, como los modos de combate y fútbol.
- Autocorrección de posición mediante sensor Gyro
- Juego de pinzas y sensores incluidos para varias misiones.
- Control remoto inalámbrico incluido (BT-410 instalado)
- Versión actualizada del software de programación RoboPlus incluido
- Comunicación de paquetes digitales y disposición sencilla de cables en cadena tipo margarita. [4]



Fig. 2 Robot Bioid GP (armado) [4]

2.2.2. Estructura de los Robots Humanoides

Los robots poseen una composición mecánica, que, pese a ser complejo, la mayor parte de los robots humanoides poseen un elevado número de grados de independencia incrementando la dificultad de las construcciones, de accionamiento y los requisitos de control.

En la actualidad hay algunas maneras para diseñar un robot humanoide. Ejemplificando, se puede diseñar un robot como una agrupación de enlaces firmes, que se hallan empalmados, donde se recibe la funcionalidad de los elementos primordiales del humanoide, como las extremidades, el tronco, los hombros, los brazos y las manos con el objetivo de diseñar las propiedades mecánicas simples.

Un segundo diseño, es donde usan los conocimientos adquiridos en la biomecánica. En éste, la línea de robots humanoides es una afinidad compleja del esqueleto humano. [1]

2.2.3. Servomotores

El servomotor es un diminuto dispositivo electrónico que tiene una línea de actuación controlada. Con él mando de una señal codificada podría ser trasladadas a exactas posiciones angulares. Hace falta que exista una señal codificada en la línea de ingreso, tal servo conservara la postura angular del engranaje. Una vez que existan cambios en la señal codificada, la postura angular de los engranajes cambiara (Quintanilla, 2012). Después, en la Fig. 3 , muestra el diagrama de un servomotor.

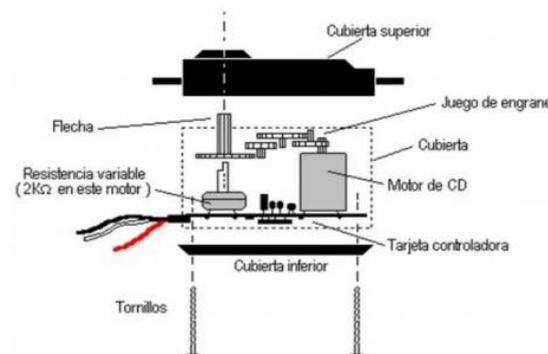


Fig. 3 Diagrama del servomotor [1]

Según Sawant & Ginoya (2010) nos expone que, los servomotores de corriente continua (CC), se usan extensamente en el campo de equipo automatizado, como robot industrial, máquinas de construcción de control numérico, impresoras y trazadores gráficos capaces. En estas aplicaciones, es preciso que la postura predeterminada debería ser adquirida de la postura antecedente en un corto lapso. Por lo consiguiente, se hace primordial mantener el control de la proporción de tensión eléctrica suministrada al servomotor por medio de la detección continua de la postura y rapidez del eje. Los

servomotores que se usan el robot Bioloid GP en el desarrollo de este trabajo, son dieciocho servos dynamixel que se presentan en la Fig. 4, 8 motores AX-12 para preeminente corporal y 10 motores AX-18 para la parte inferior corporal.



Fig. 4 Servomotores dynamixel AX-18A y AX-12^a [1]

2.2.4. Características del servo AX-18^a

Las propiedades más relevantes del dynamixel Ax-18A son:

- Peso: 54.5 gr.
- Magnitud: 32mm x 50mm x 40mm
- Ratio de reducción: 254: 1
- Torque: 1.83 N.m (a 12.0 V, 2.2 A)
- Rapidez sin carga: 97 rpm (a 12 V)
- Grados de giro: 0° ~ 300°
- Tensión de operación: 9 ~ 12 V
- Señal de comandos: paquete digital
- Tipo de protocolo: comunicación serie asíncrona half duplex

2.2.5. Asignación de pines

Las asignaciones de pines del conector son las siguientes. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 puede funcionar con un solo

conector conectado [5]. Podemos diferenciar los pines de conexión del servomotor en la Fig. 5 de abajo.

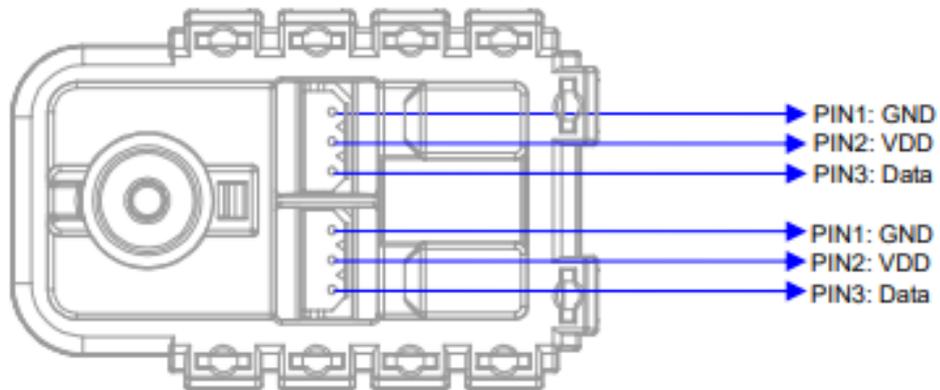


Fig. 5 Asignaciones de pines [6]

2.2.6. Alambrado

Conectar el pasador a pasador de los actuadores AX-2 como se muestra en la Fig. 6 a continuación. Muchos actuadores AX-12 se pueden controlar con un solo dispositivo de esta manera.

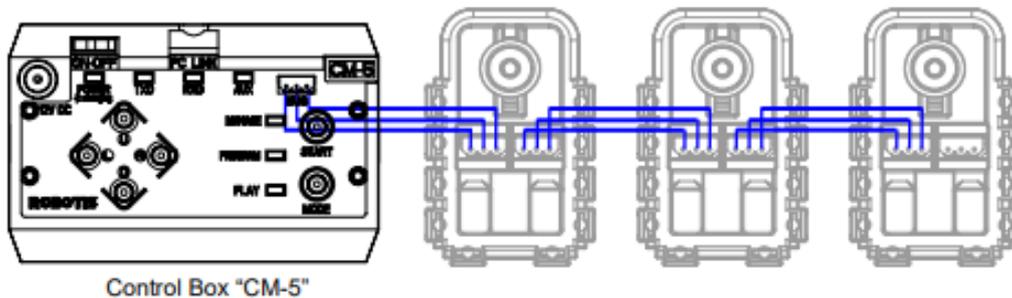


Fig. 6 Alambrado de conexión [6]

2.2.7. Enlace de pc

Se puede usar una PC para controlar el Dynamixel a través del controlador CM-5. Se logra apreciar el enlace a un pc con los servomotores en la Fig. 7 de abajo.

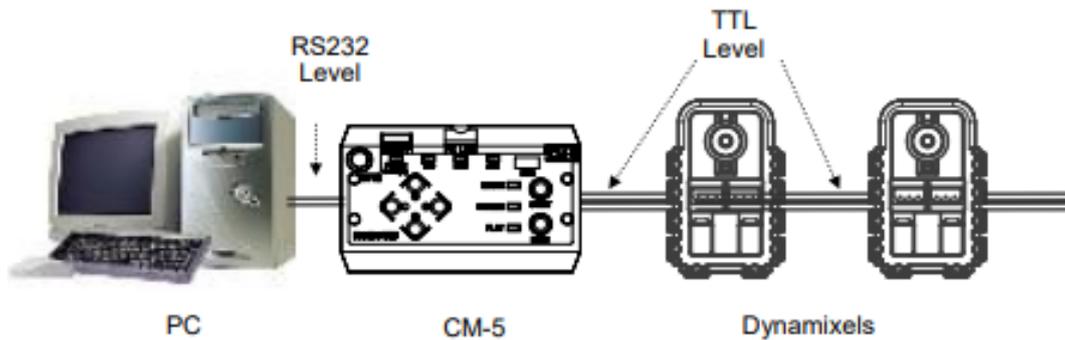


Fig. 7 Enlace a un computador [6]

2.2.8. Propiedades del servo AX-12A

Las propiedades más importantes del dynamixel Ax-12A tenemos:

- Peso: 54.6 gr
- Magnitud: 32 mm x 50 mm x 40 mm
- Resolución: 0.29°
- Ratio de reducción: 254: 1
- Torque: 1.52 N.m (a 12.0 V, 1.5 A)
- Rapidez sin carga: 59 rpm (a 12 V)
- Grados de giro: 0° ~ 300°
- Rotación continua
- Señal de comandos: paquete digital
- Tipo de protocolo: comunicación serie asíncrona half duplex

2.2.9. Sensor

Los sensores son dispositivos hechos desde elementos pasivos y activos. Dichos artefactos permaneces hechos para notar información de una intensidad del exterior y convertirla en otra, comúnmente eléctrica, simple de cuantificar y manipular.

2.2.10. Sensores Gyro GS-12

El giroscopio es un dispositivo que sirve para medir, fijar o modificar la orientación de cualquier artefacto, el giroscopio “gyro sensor (gs-12)”, es bastante diminuto, de bajo consumo y una contestación bastante instantánea. Esto ya que, es una herramienta netamente electrónica, sin piezas móviles.

En la Fig. 8 de debajo, tenemos la posibilidad de ver la especificación descripción del giroscopio. Una vez que hablamos de robots, la salida se enlaza a un microcontrolador, que es capaz de leer la holgura del 49 pulso del giroscopio, y tal cual saber en qué momento se genera un giro.

En los robots balancín y en los sistemas de exactitud, resultan muy primordiales los sensores giroscópicos. Debido que hacen una medición y compensan el instante de giro.

Las propiedades del gyro-sensor GS-12 son:

- Peso: 2,8 g
- Magnitud: 23mm * 23mm * 10mm
- Temp.: -40° ~ 85°
- Rango angular: 300° / s ~ 300° / s
- Tensión sugerida: 4,5 ~ 5,5 V

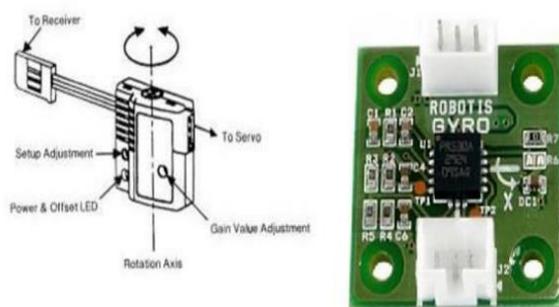


Figura 2. 21: Descripción del giroscopio.

Fuente:(Peralta, 2009).

Fig. 8 Descripción del giroscopio [1]

2.2.11. BT-410 Dongle converts USB 2.0

BT-410 Dongle convierte la interfaz USB 2.0 y la interfaz de comunicación estándar Bluetooth 4.0 Low Energy (Master), como se muestra en la Fig. 9.

Si el Dongle BT-410 está conectado a la PC, se puede emparejar con un módulo esclavo BT-410 para intercambiar datos. [7]



Fig. 9 BT-410 Dongle converts USB 2.0 [7]

2.2.12. Lithium polymer battery 1000mah

Las baterías LiPo son baterías recargables que tiene algunas veces de diversas celdas utilizadas en aplicaciones que necesitan corrientes mejores a 1A con bajo peso y tamaño limitado, ejemplificando, el método de radio control, tales como aviones, helicópteros, entre otros. Como se muestra en la Fig. 10. [8]



Fig. 10 Batería de Litio [9]

2.2.13. Controlador CM-530

Es un controlador que tiene una CPU, tarjeta TTL, LED de estado, botón de ingreso y puerto de GP I/O.

El robot es comandado por el controlador CM-530 tal como mostramos en la Fig. 11, en él tiene un microcontrolador ARM Cortex STM32F103RE. Este microcontrolador está compuesto por 64 pines que comandan independientemente cada parte del robot, como entrada/salida y cantidad de tensión que es utilizada por el robot. Cada uno de estos pines contiene una serie de aletas, nombre del pin y un nombre de red que la persona conoce para así lograr programarlos.

Cualidades:

- Similar y compatibilidad con el AX Dynamixel y la serie MX
- Mantiene la comunicación Bluetooth y Zigbee
- Se enlaza a un PC por medio del puerto USB [10]



Fig. 11 Placa controlador CM-530 [11]

El controlador tiene LED's de señal y botones para la comunicación primordial con el cliente, puertos para la conexión de dispositivos periféricos extras y diversos canales para enlaces de red Dynamixel. Un minipuerto USB con un conversor serial a USB, posibilita

la comunicación en serie para la instalación del firmware y la depuración. En la Fig. 12 posemos el controlador con las piezas.

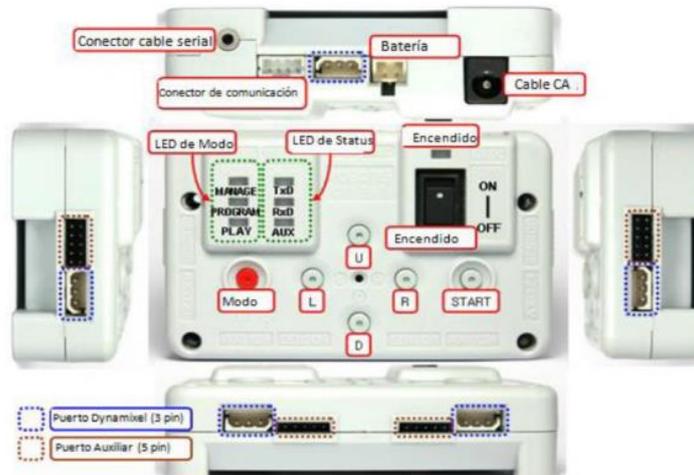


Fig. 12 Controlador CM-530 y sus partes [1]

2.2.14. Medio de Comunicación del Bioloid GP

La comunicación entre el robot humanoide Bioloid GP y el mando de lejos, se consigue por medio de la tecnología ZigBee. Esta tecnología usa los módulos zig-100/110-A, que se muestra en la Fig. 13 de debajo.



Fig. 13 Modulo Zig-100/110-A [1]

Como sugiere el creador Quazi (2012) este medio de comunicación, es una tecnología inalámbrica desarrollado como un estándar global abierto, para atender las necesidades de bajo costo y equipos inalámbricos de bajo consumo. Los módulos de ZigBee, conforme el creador Furlán (2016) trabajan en la frecuencia de 2.4 GHz, y son utilizados como una red de área personal (PAN), dejando a un lado la comunicación por cable, que tenemos la posibilidad de mirar en la Fig. 14 de debajo. Entre más especificaciones relevantes de los módulos tenemos:

- Ancho de banda más alto de 250 kbps en transferencia de datos.
- Interfaz mediante un Transmisor-Receptor Mundial (UART), que podría ser configurada para mandar y recibir información.
- Voltaje de ingesta de alimentos de 3.3 V con un consumo de corriente de 30mAh.
- Magnitudes: 26.5 mm de ancho, 19 mm de extenso y 12 mm de elevación.

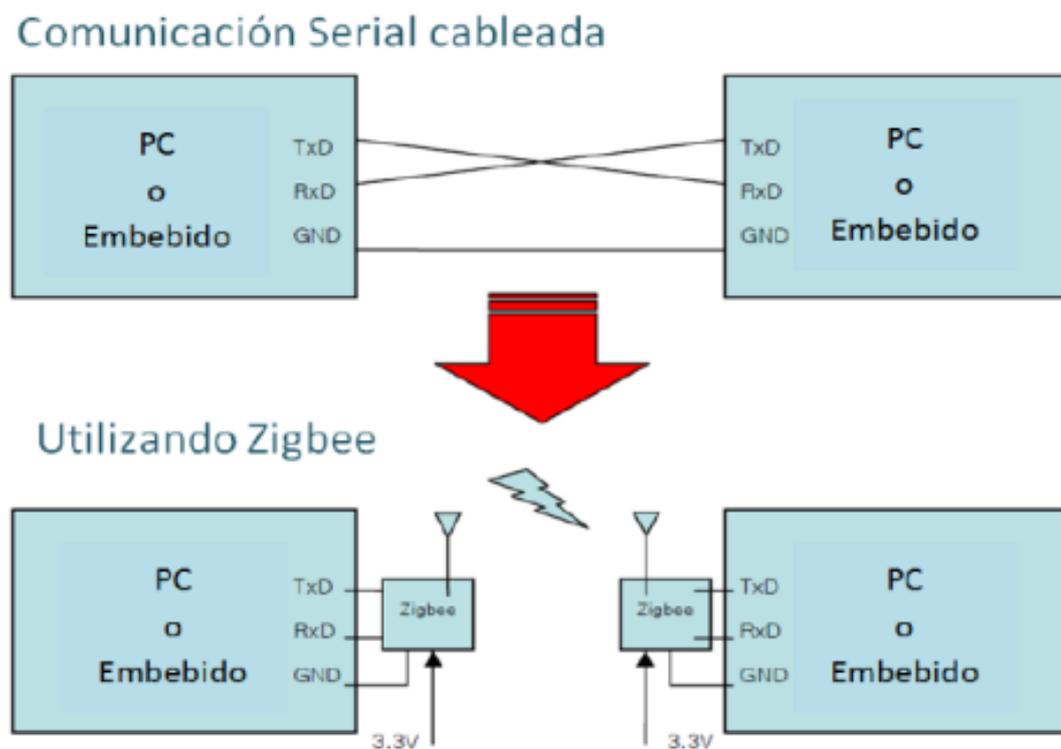


Fig. 14 Funcionamiento de ZigBee [1]

2.2.15. Raspberry Pi 4

La Raspberry es pequeño ordenador de escritorio del tamaño de una tarjeta de crédito. Se debe conectar un monitor, teclado, ratón y cargar la última imagen Raspbian en una tarjeta MicroSD. Con esto se obtendrá un diminuto ordenador listo para ser utilizar. Dada su enorme potencial, es la placa perfecta para trastear con proyectos como Tensorflow, PiHol o Kodi. Como se muestra en la Fig. 15.

Cualidades:

- Raspberry Modelo B
- Procesador: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Memoria: 4GB LPDDR4-2400 SDRAM
- Conectividad: 2.4 GHz y 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE
- Conectividad ethernet: True Gigabit Ethernet
- 2 USB 3.0
- 2 USB 2.0 [12]



Fig. 15 Placa Raspberry Pi 4 [12]

2.2.16. Raspberry Pi Camera Module V2 8MP

La Camera v2 es un sensor de imagen de 8 megapíxeles Sony IMX219 de alta calidad a medida para Raspberry Pi, con un objetivo de enfoque fijo. Es capaz de imágenes estáticas, y además aguanta vídeo. Se conecta a Pi mediante uno de los pequeños zócalos en el área preeminente de la placa y usa la interfaz CSI dedicada, diseñada en especial para interactuar con cámaras. Como se muestra en la Fig. 16. [13]



Fig. 16 Raspberry Pi Camera Module V2 8MP [13]

2.2. SOFTWARES UTILITARIOS

Hay diferentes posibilidades para laborar con el robot humanoide Bioloid GP, una de ellas es por medio del programa RoboPlus que da la vivienda de robotis. RoboPlus es un ámbito incluido que encapsula 3 elementos, RoboPlus Task, RoboPlus Manager y RoboPlus Motion. Dichos recursos implementan las funcionalidades del robot como, hacer programación y otros accesorios, en una interfaz no tan compleja de utilizar. En la Fig. 17debaajo tenemos la posibilidad de ver la interfaz del Roboplus.



Fig. 17 Interfaz de Roboplus [1]

2.2.1. Software Roblopus

RoboPlus es un paquete de software que le permite interactuar fácilmente con todo el hardware de ROBOTIS, incluidos los controladores de ROBOTIS, Dynamixel, sensores y otros componentes de hardware. Como demostramos en la Fig. 18.



Fig. 18 Imagen principal del Software Roboplus [14]

Consta de los siguientes programas:

RoboPlus Task: Task es una colorida herramienta de programación basada en iconos basada en C++, como se demuestra en la Fig. 19.

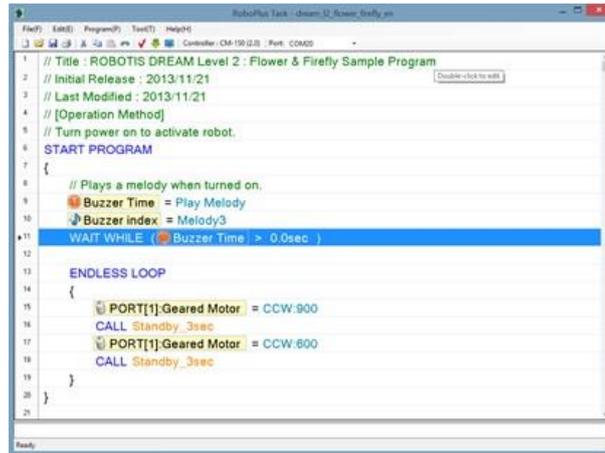


Fig. 19 Imagen de Roboplus Task [14]

RoboPlus Manager: Manager proporciona una manera fácil de configurar y administrar controladores y componentes, incluida la actualización/recuperación del firmware del controlador y la configuración Zig2Serial para comunicaciones inalámbricas ZigBee, como se demuestra en la Fig. 20.

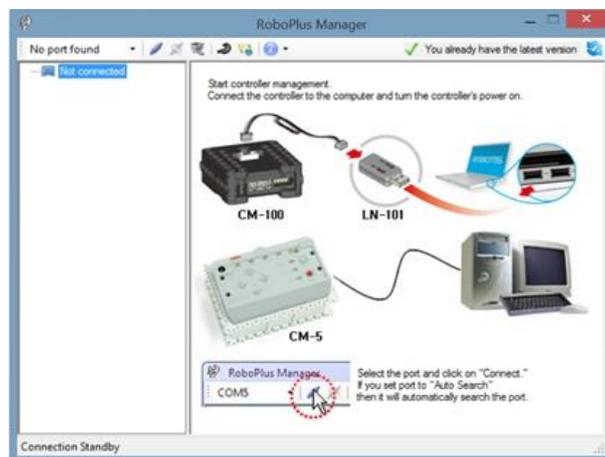


Fig. 20 Imagen de Roboplus Manager [1]

RoboPlus Motion: Motion es una herramienta de programación animada que se utiliza para crear movimientos de robots (coreografías) editando la velocidad y la posición de Dynamixel, como demuestra en la Fig. 21. [14]

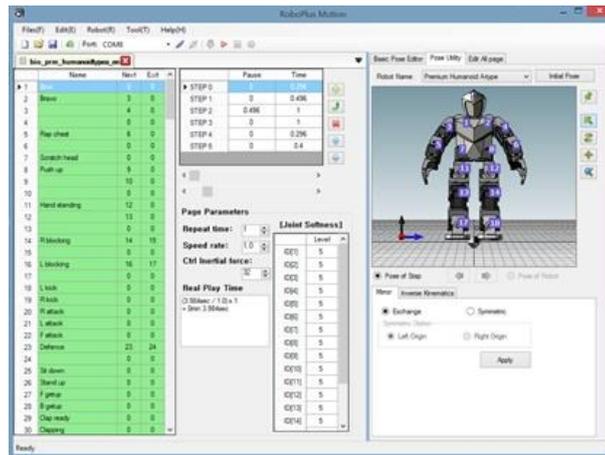


Fig. 21 Imagen de Roboplus Motion [14]

2.2.2. Software VNC viewer

VNC posibilita tomar el control del ordenador servidor remotamente por medio de un comprador multiplataforma. Una vez instalado VNC en el ordenador, es viable entrar a partir de cualquier parte por medio de Internet y a partir de otro dispositivo. VNC da consumidores en especial adaptados a cada una de las plataformas, como se observa en la Fig. 22. [15]

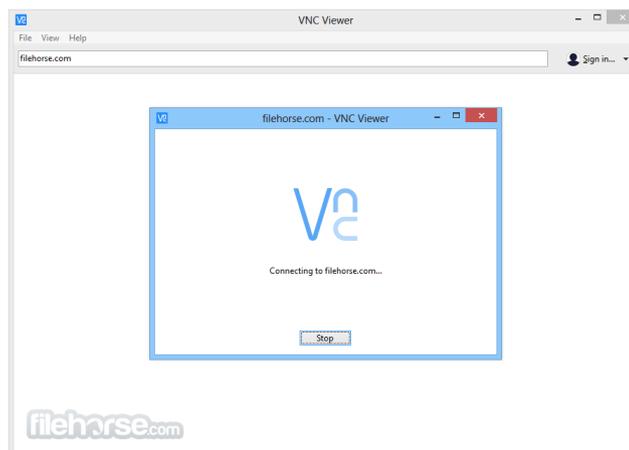


Fig. 22 Imagen del software VNC [15]

MARCO METODOLÓGICO

3.1. FASE INICIAL

En el siguiente mapa conceptual se presenta el Kit de Robótica Bioloid GP de la marca RobotiS, como se muestra en la Fig. 23.

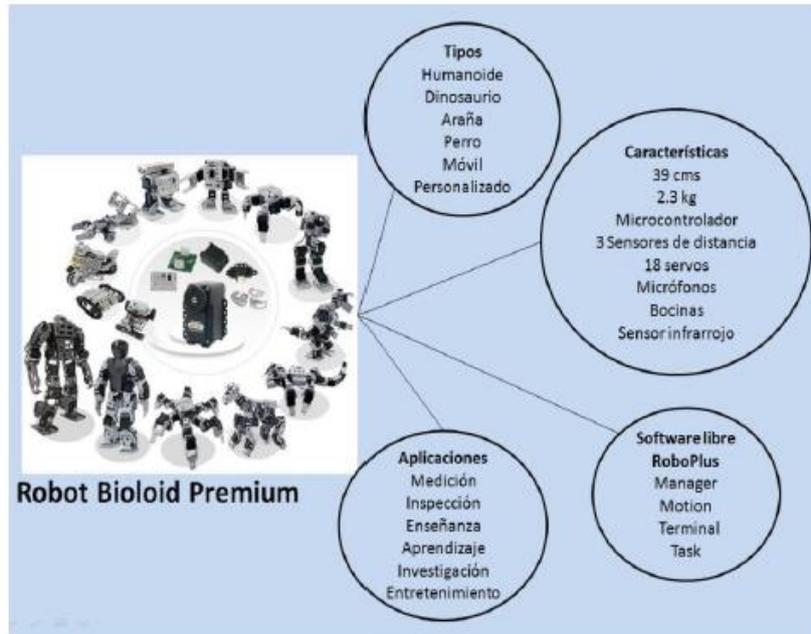


Fig. 23 Tipos de configuraciones del robot Bioloid [16]

Este robot tiene una sección mecánica, donde permanecen integrados los servomotores dynamixel, su composición metálica, y su fase de control que está conformada de sensores, controlador CM-530, un transmisor, un receptor de radiofrecuencia y para finalizar una batería de lipo de 1000mAh con 3 celdas con su respectivo cargador.

El kit adquirido, cuenta con las herramientas elementales para el conveniente ensamblaje del robot humanoide que va de la mano como un programa que es requerido para llevar a cabo los algoritmos para su correcto manejo, en la siguiente Fig. 24 se muestra de forma gráfica los recursos del kit que conforman el robot humanoide.



Fig. 24 Kit del robot Bioloid [17]

El programa roboplus, que se muestra en la Fig. 25, dispone de diversos programas que realizan viable el manejo según sus las necesidades y aplicaciones que se la deseé ofrecer al robot, entre estas poseemos las próximas:

- Roboplus Task: Esta elección explica la manera de código las secuencias de los movimientos a hacer conforme con la lógica de programación
- Roboplus Manager: Se ocupa de registrar el manejo de los servomotores y sensores del robot.
- Roboplus Motion: Este programa posibilita crear y registrar los movimientos de forma secuencial.
- Guía de Usuario: Otorga una explicación de la implementación desde la lógica de programación hasta los movimientos al ser ejecutados.



Fig. 25 Interfaz principal del programa roboplus [17]

Los servomotores poseen cualidades especiales ya que tiene de una comunicación full dúplex que permite enviar un estado y recibir instrucciones para realizar movimientos.

3.1.1. Estructura

Esta parte se describe los pasos para la construcción del robot Bioloid GP. A continuación, se detalla el proceso que era su construcción.

En lo primero se debe separar las herramientas, servos, y las partes metálicas de la estructura.

Luego, en la Fig. 26 se enseña la forma gráfica del método a seguir para ubicar cada marco metálico con tornillo y aro en los servomotores.

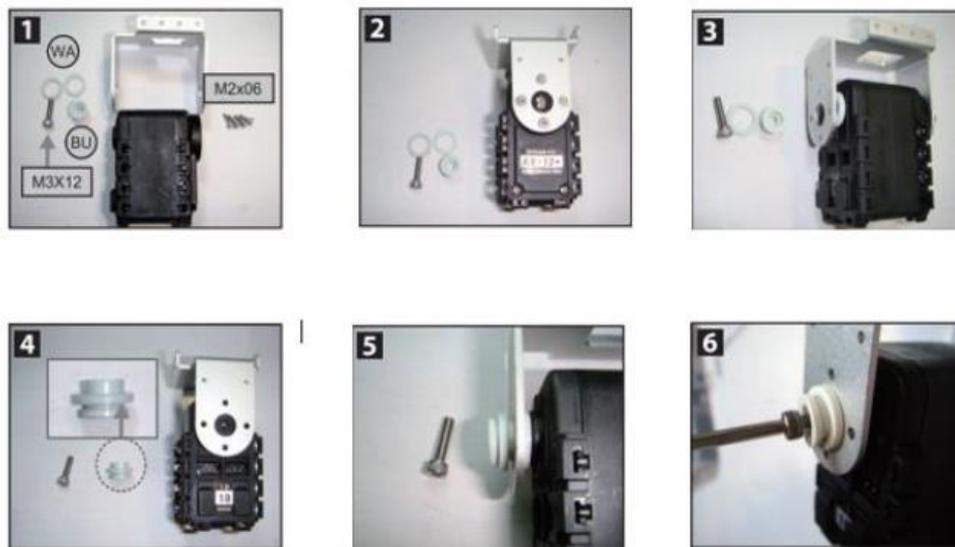


Fig. 26 Pasos para sujeción de marco metálico en los servos. [17]

Con este proceso en la siguiente Fig. 27 se demuestra la construcción de los servomotores con los brazos metálicos.



Fig. 27 Resultado del ensamblaje. [17]

Posteriormente para seguir con la obra del robot se necesita empezar ensamblando el pecho y después los brazos, para ello se debe insertar tuercas en los servos ID1 e ID2 como se mira en la Fig. 28

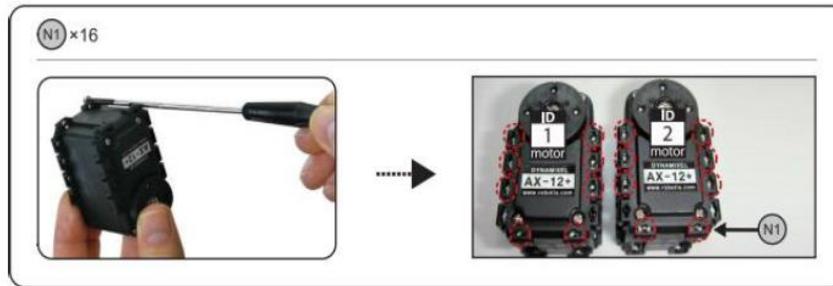


Fig. 28 Colocación de tuercas en los servomotores ID 1 e ID2 [17]

Posteriormente se debe conectar los servomotores ID1 e ID2 con los cables 10, para ser ubicados en la estructura metálica que se encuentra al pecho, en la Fig. 29 se puede ver el método a seguir.

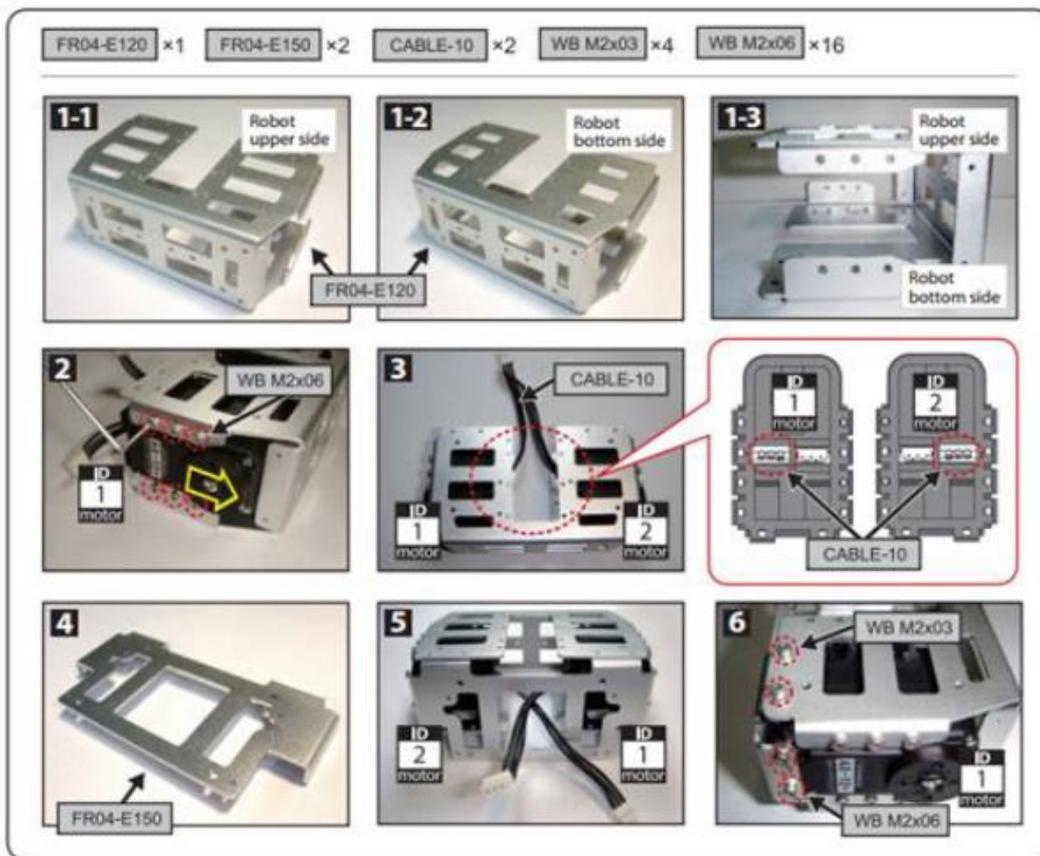


Fig. 29 Colocación servomotores en la estructura metálica. [17]

Una vez realizado lo anterior se debe ensamblar el pecho del robot, para lo cual, se deben juntar los marcos metálicos de los hombros para agarrar los servomotores de los brazos.

Luego, en la Fig. 30, se realiza una descripción de los pasos avanzar del cómo poner dichos marcos en el disco de movilización que se encuentran localizados en la caja realizada anteriormente.

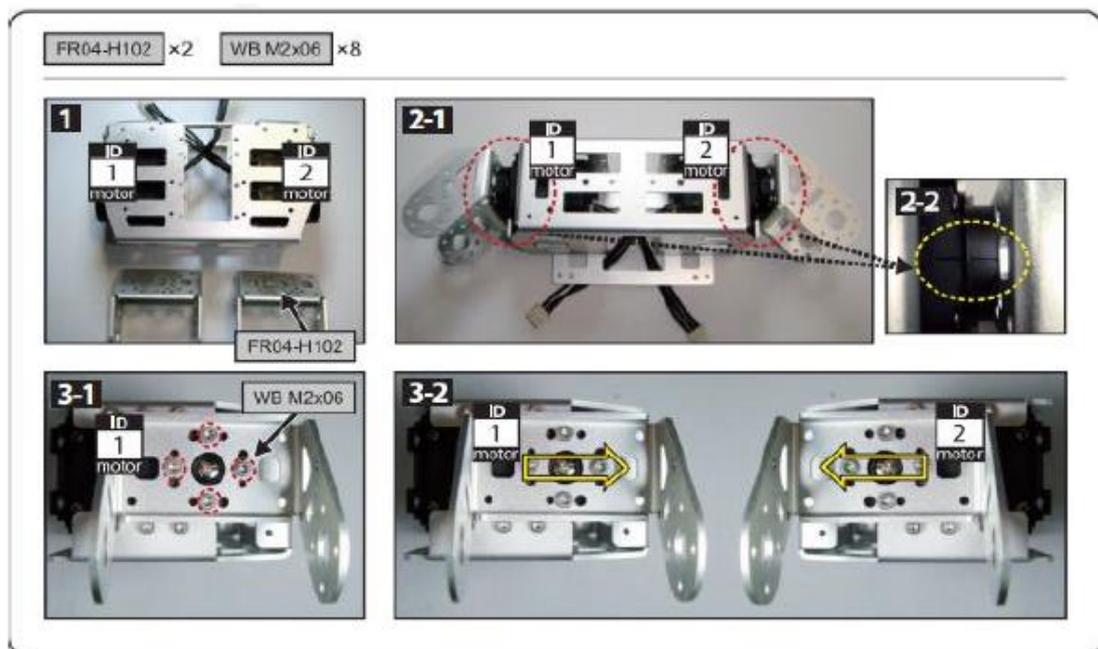


Fig. 30 Fijación de marcos metálicos en los ejes de los servomotores. [17]

Luego en la Fig. 31, se describe cómo ejecutar los servomotores que serán parte de los brazos. Por esto, se debería de ingresar las tuercas de los servos ID3 e ID4 de AX-12A; posteriormente se demuestra la composición FR04-SC101. Al final, se ingresan las tuercas de los servos ID5 e ID6 de AX-12A.

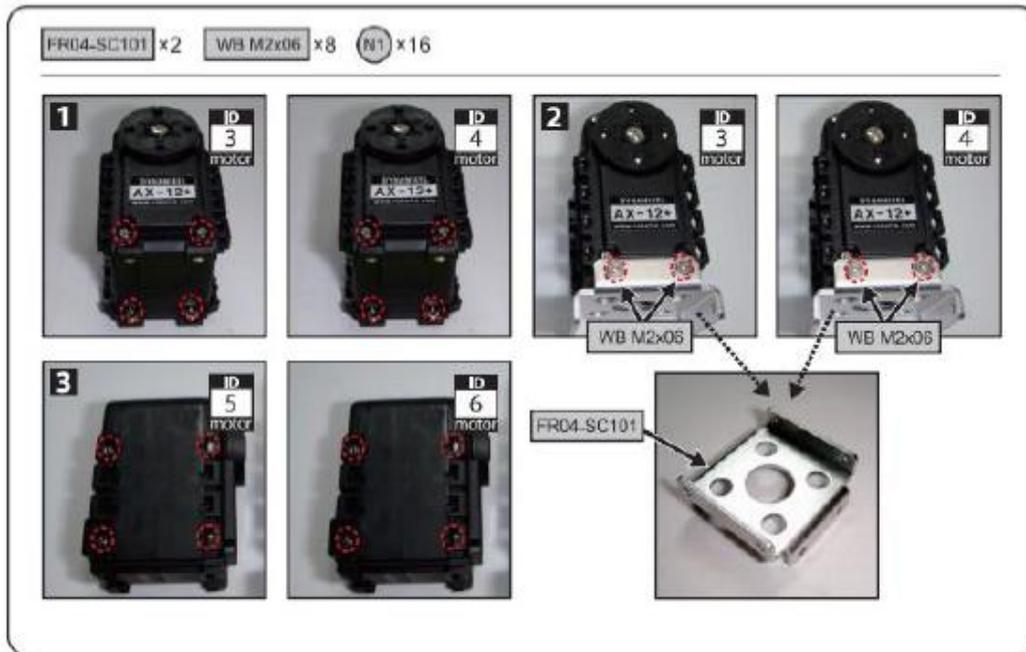


Fig. 31 Preparación de servomotores para brazos. [17]

Luego, se ensambla los servomotores, se estima unirlos como en la Fig. 32, en el cual los brazos se instalan arriba de los marcos mencionados en pasos atrás.



Fig. 32 Fijación de servomotores que conformaran los brazos del robot [17]

En este caso para armar los brazos, es necesario poner los tornillos de la composición (FR04-E180) para luego acoplar con la parte FR04-H102 como detallamos en la Fig. 33

del método. En este caso, se debe tener en cuenta que el kit incluye 2 servomotores extra que son de ayuda a las pinzas de agarre del robot.

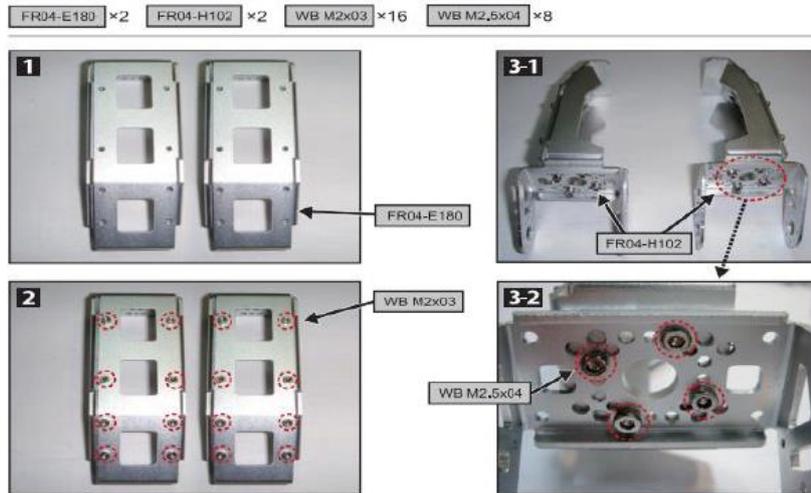


Fig. 33 Estructura metálica para las manos [17]

Como siguiente paso se debe realizar la construcción y conjuntamente los brazos al pecho del robot, con esto es para los respectivos servos que detallamos en la Fig. 34.

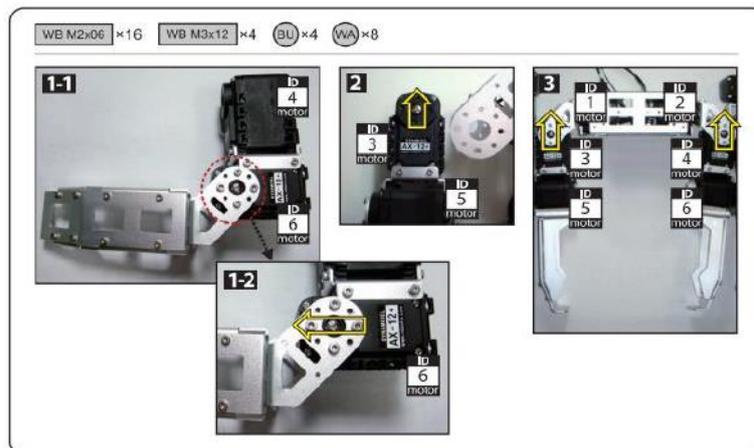


Fig. 34 Unión de brazos con el pecho. [17]

Luego, Fig. 35 se enseña el ingreso del giroscopio al soporte que viene para el dispositivo, FR04-E191, con esto se sitúa todo lo descrito en la composición del tórax del robot humanoide FR04-E131.

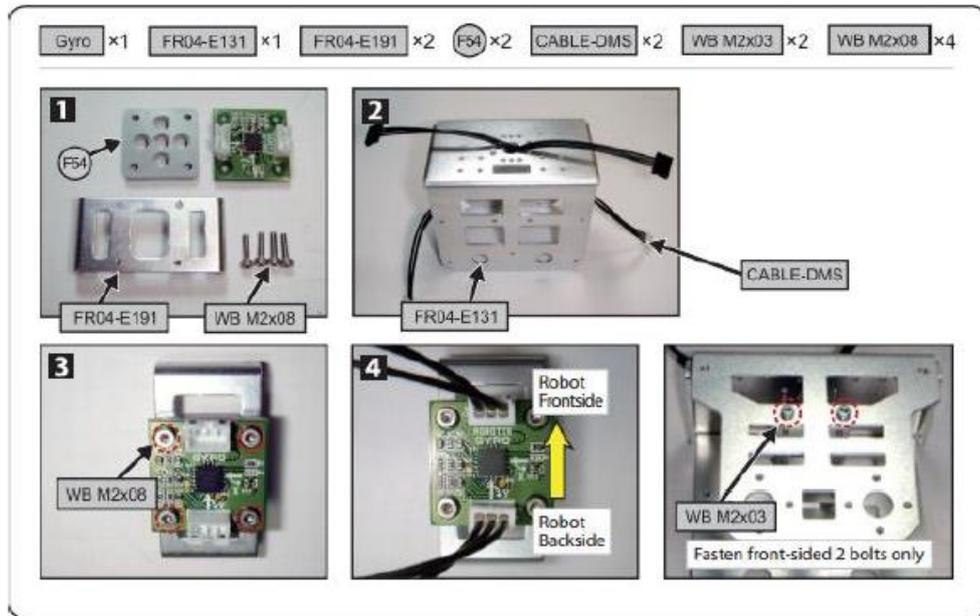


Fig. 35 Colocación del giroscopio a la estructura. [17]

Las partes individuales se usan para ubicar el lugar donde ensamblar cada una. Esto se detalla en la Fig. 36, que nos permite ver la estructura metálica que corresponde a la pata adosada.

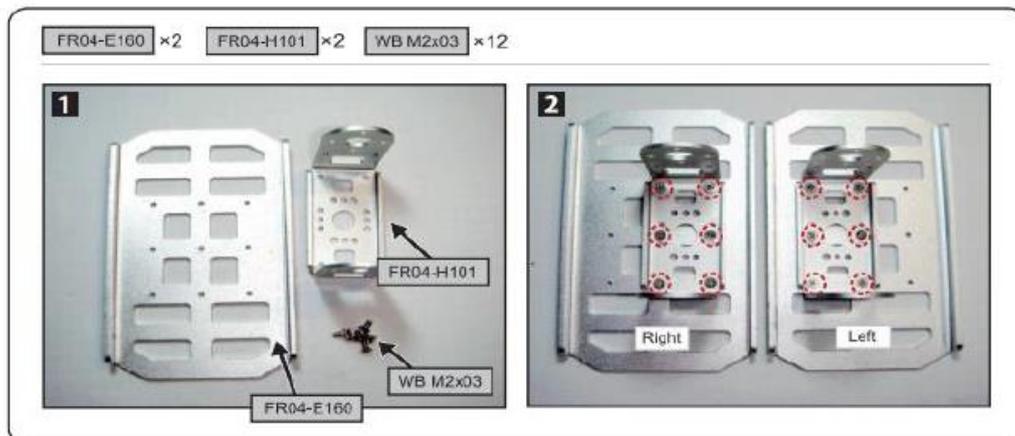


Fig. 36 Partes metálicas correspondientes a los pies. [17]

Una vez realizado esto, se debe ingresar las tuercas en los servos ID15 e ID16, con esto la tuerca N2 a la parte F7, se unifica a la F7 del motor como se describe en la Fig. 37.

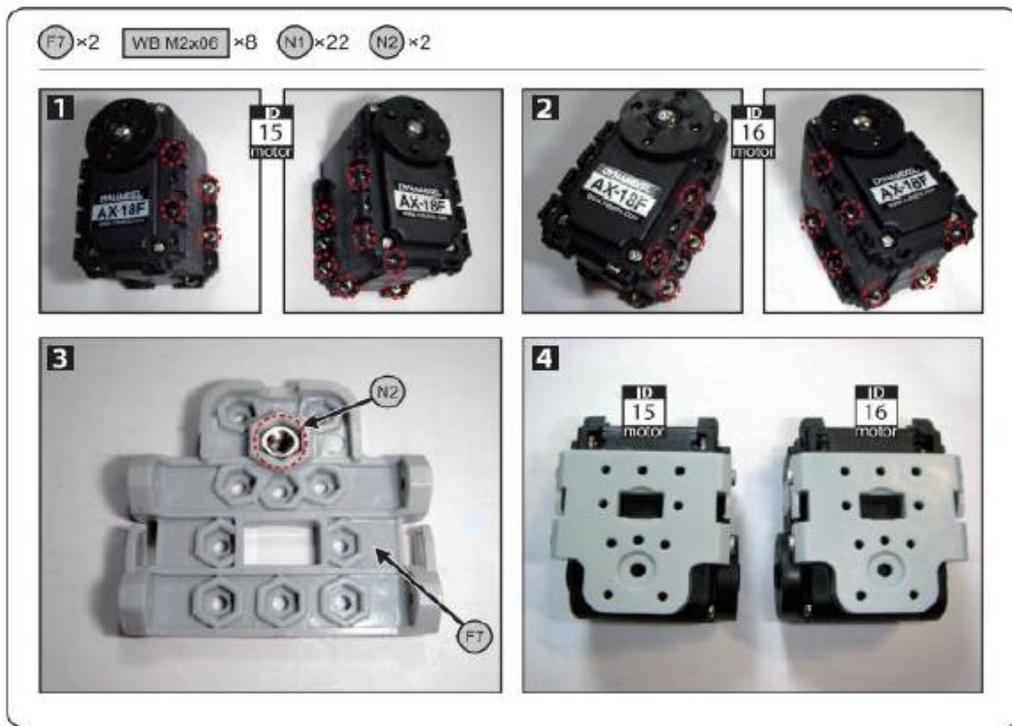


Fig. 37 Procedimiento para acoplar servos correspondientes a los pies [17]

Con las patas del robot se debe instalar tuercas en los servos ID17 e ID18, y se une a la forma horizontal del FR04X10 como se detalla en la Fig. 38.

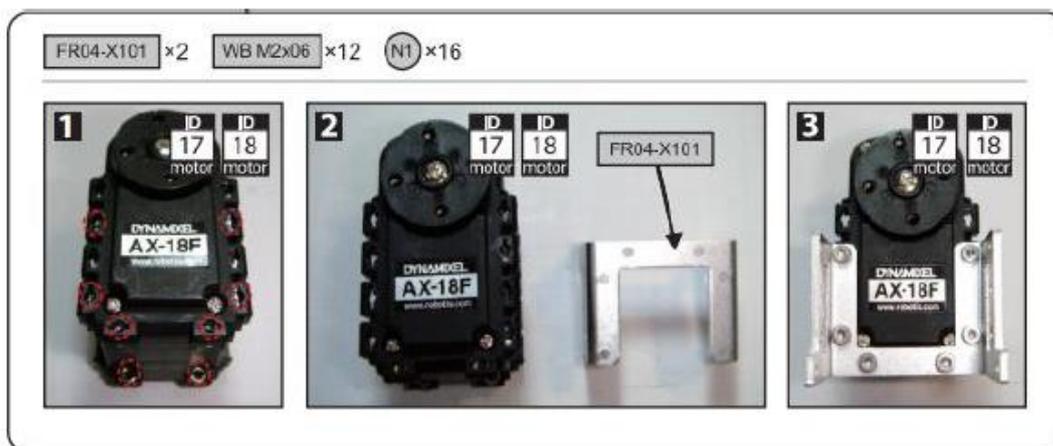


Fig. 38 Procedimiento para ensamblaje de los tobillos. [17]

Luego, en la Fig. 39 se describe el seguimiento donde se describe la postura de la estructura sobre los servos ID15 e ID16 para unir la formación del pin y a esto se unifica al pin anteriormente ensamblado.

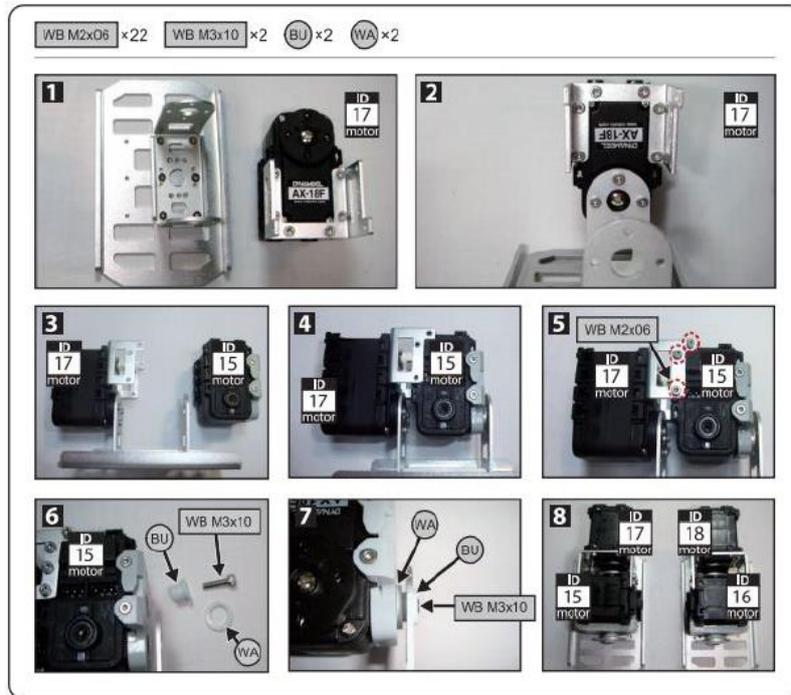


Fig. 39 Procedimiento para juntar tobillo con el pie. [17]

Con esto se pone el tobillo y el pie encima de la estructura FR04H120 que conforma la pierna. La figura 3.18 detalla el proceso de manera ilustrativa.

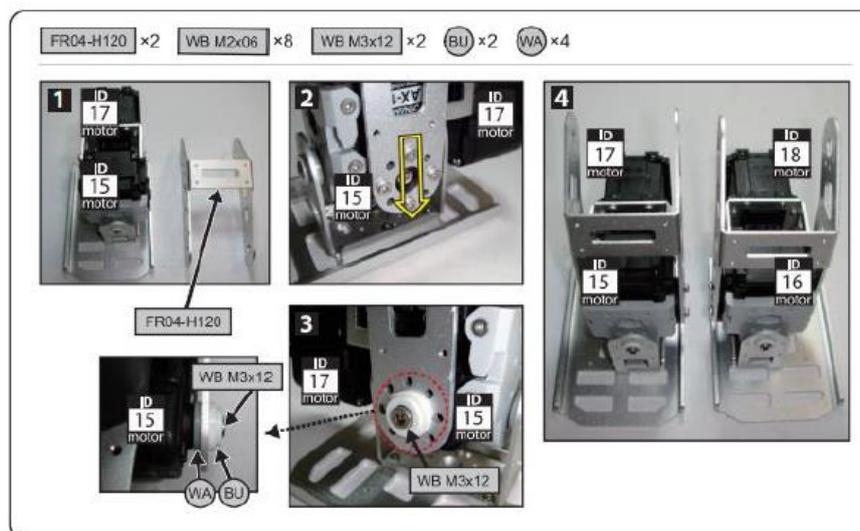


Fig. 40 Procedimiento para juntar tobillo con el pie. [17]

Posteriormente, la Fig. 41 nos enseña cómo los pines unidos y la construcción FR04SC110 se elabora la rodilla del robot humanoide.

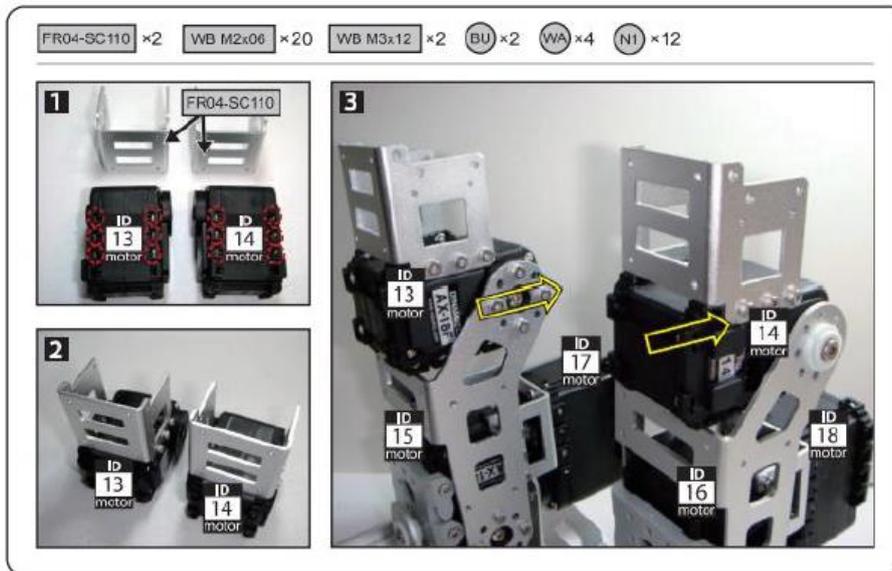


Fig. 41 Procedimiento para acoplar las rodillas con las piernas. [17]

Luego nos lleva a elaborar el paso es unir los servos ID11 e ID12 a los pines para que luego de manera conjunta se una a los ID9 e ID10 a la estructura del FR04HC110, una vez que se tenga listo esto se unen a los pines. A continuación, en la Fig. 42 se detalla cómo se ve la parte de la conexión de los cables de control.



Fig. 42 Unión de rodillas, pies y piernas. [17]

Con esto el desarrollado que está bajo la unión del pie y el cuerpo, se detalla en la Fig. 43. Para este paso se debe a unir los accesorios plásticos que agarran a la batería, como se detalla en la Fig. 44, para esto de manera inicial es un espaciador y posterior a ello la base que nos ayuda el mantenimiento del marco que posee la permanentemente una lipo stack.

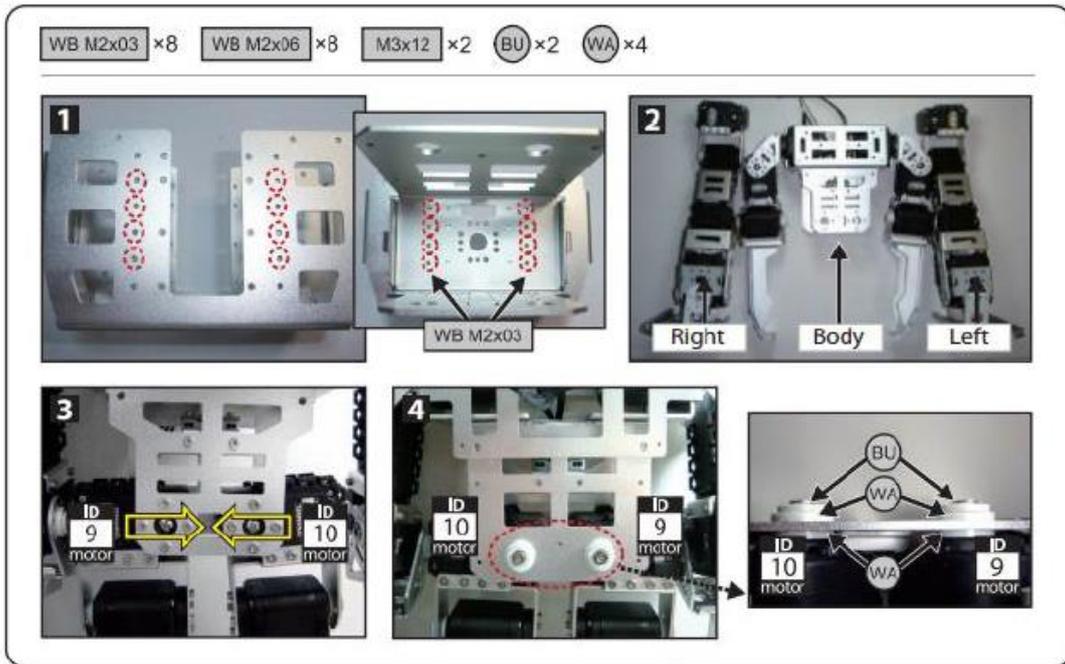


Fig. 43 Acoplar piernas juntas al cuerpo. [17]

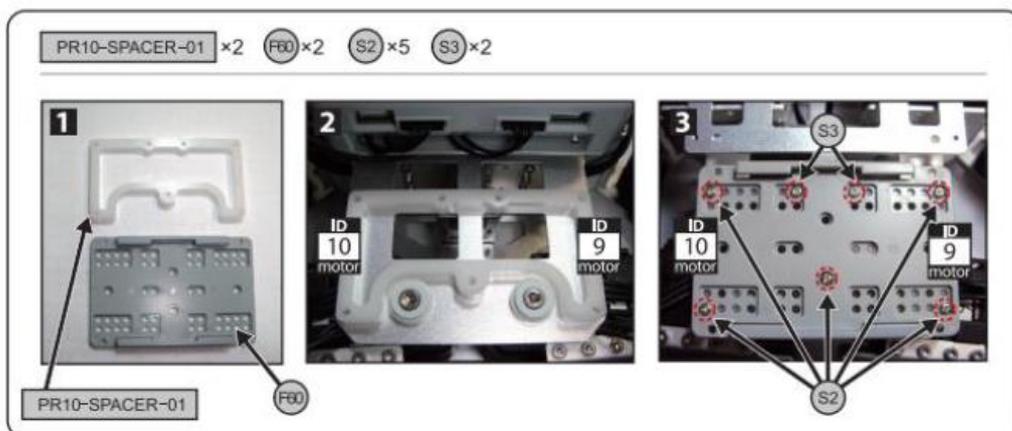


Fig. 44 Separador y base de la batería. [17]

Es de suma importancia tener presente que los tornillos se deben colocar de manera correcta como lo muestra la figura de atrás, el cual debe ponerse correctamente para fijar el soporte de la batería lipo.

Luego, se debe poner el controlador CM530 en arriba de la parte superior trasera de la estructura, es decir, la parte trasera del robot, donde se conecta los cables del giroscopio como se detalla en la Fig. 45 y se aprietan los tornillos.

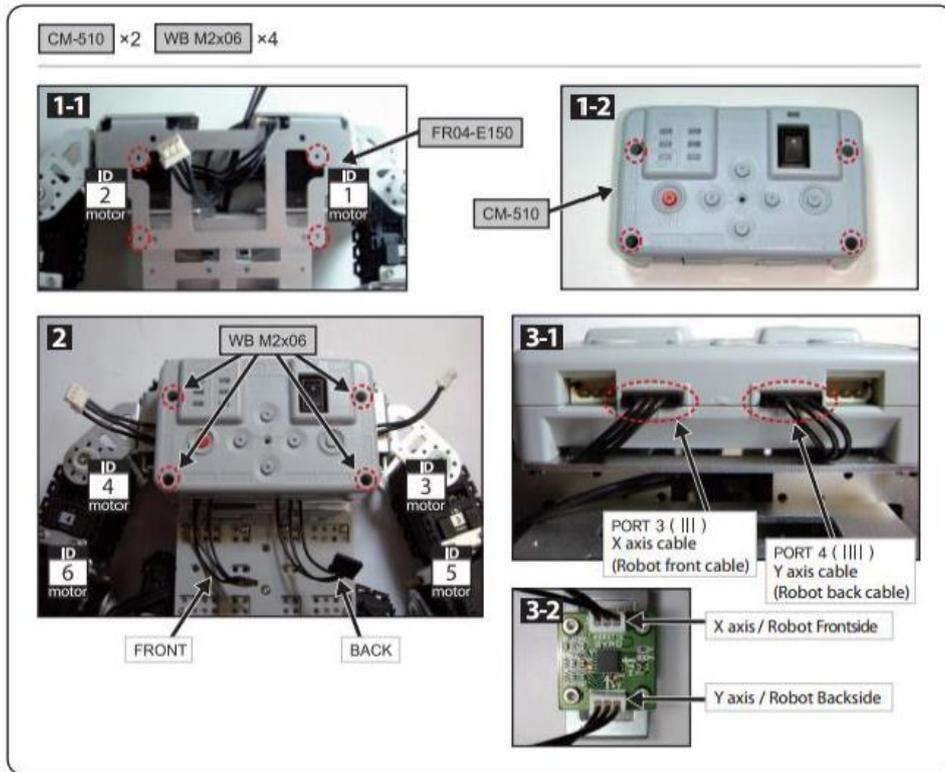


Fig. 45 Sujeción del controlador CM-530 en la estructura. [17]

El siguiente paso que se debe realizar es la conexión de los cables a los servos en orden como se describe en la Fig. 46, que nos enseña la manera de cómo deben conectarse. Luego, se ponen los servos ID2 e ID4; ID1 e ID3 con cable número 18.

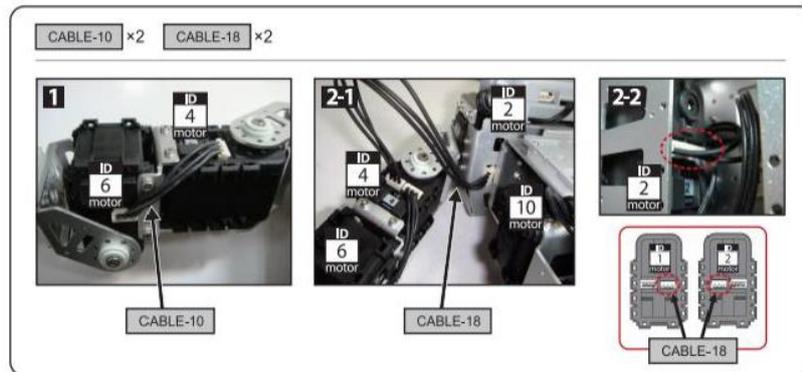


Fig. 46 Conexión de los cables 10 y 18 en los servos correspondientes. [17]

Después de hacer los pasos anteriores se debe conectar al servomotor como se detalla en la siguiente lista Fig. 47.

- Colocar los servomotores ID9, ID10 al CM-530 utilizando el CABLE-10.
- Colocar los servomotores ID9 e ID11; ID10 e ID12 con un CABLE-14.
- Colocar los servomotores ID11 e ID13; ID12 e ID14 con un CABLE-14.
- Colocar los servomotores ID13 e ID15; ID14 e ID16 con un CABLE-18.
- Colocar los servomotores ID15 e ID17; ID16 e ID18 con un CABLE-14.

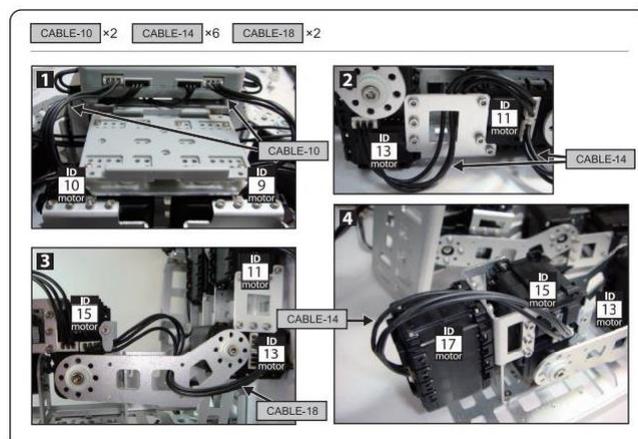


Fig. 47 Conexiones de los cables de servomotores del robot. [17]

Con lo que se demuestra en la Fig. 48 y Fig. 49, se continua con la organización del cableado de todo el robot con el porta cables F55 para que los cables estén en su posición, siendo esto necesario para reducir errores en el proceso de presentación.

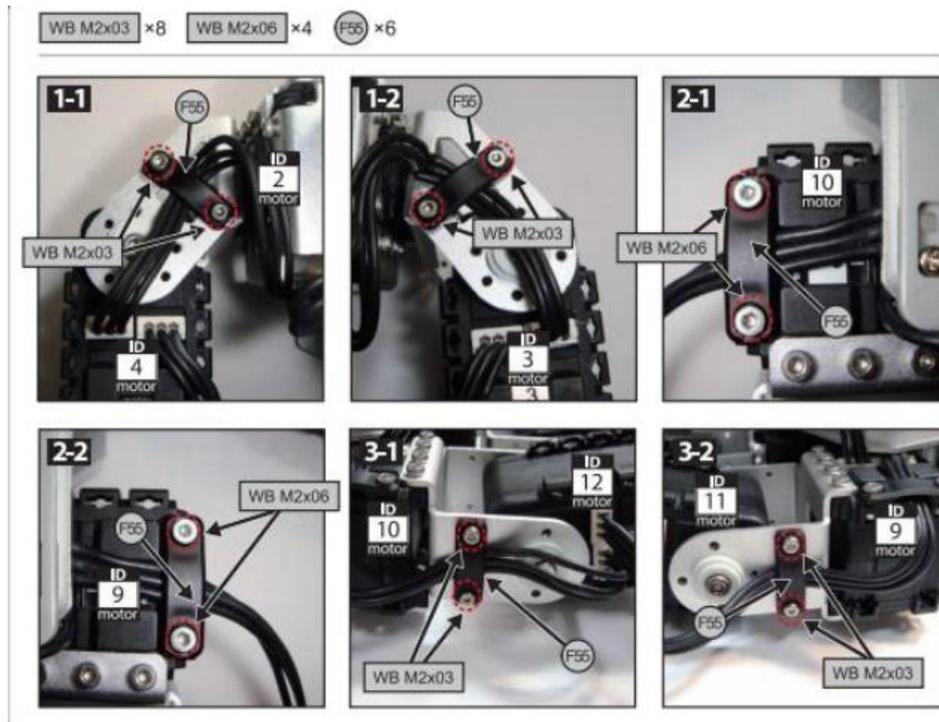


Fig. 48 Organización de los cables del robot. [17]

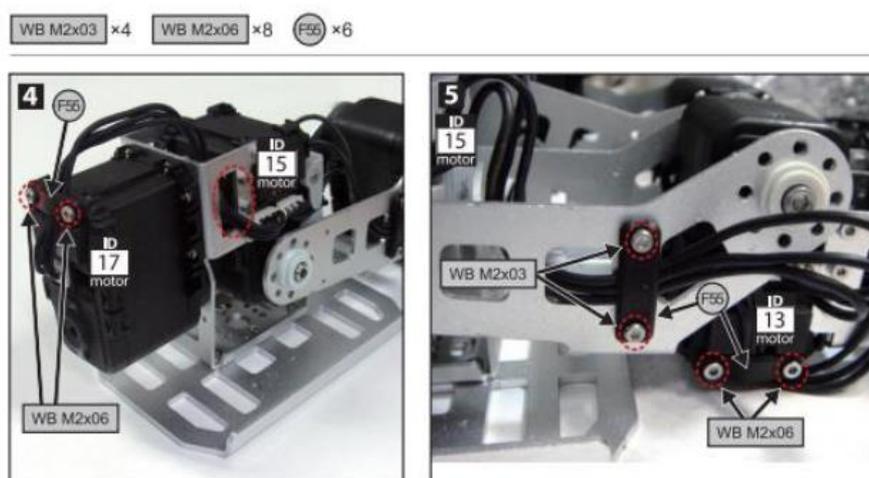


Fig. 49 Organización de los cables del robot entre sus diferentes partes. [17]

Luego se debe unir la cabeza al cuerpo del robot que tiene un marco de metal con luces LED como se detalla en La Fig. 50

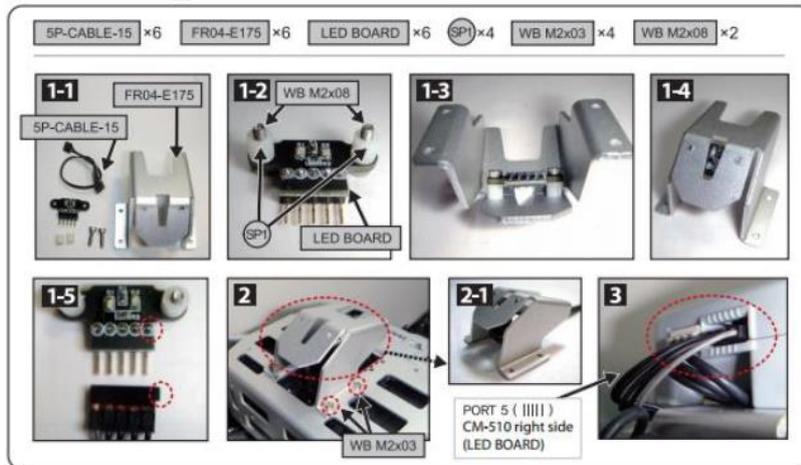


Fig. 50 Acople de la cabeza al cuerpo del robot. [17]

Una vez realizado esto se conecta un cable al controlador CM530, que hace la función de indicador permitiendo controlar los flashes por código. Y por último se realiza el montaje del robot humanoide completo, para comprobar su funcionamiento, no se debe olvidar que se debe conectar la batería lipo como se describe en la Fig. 51.

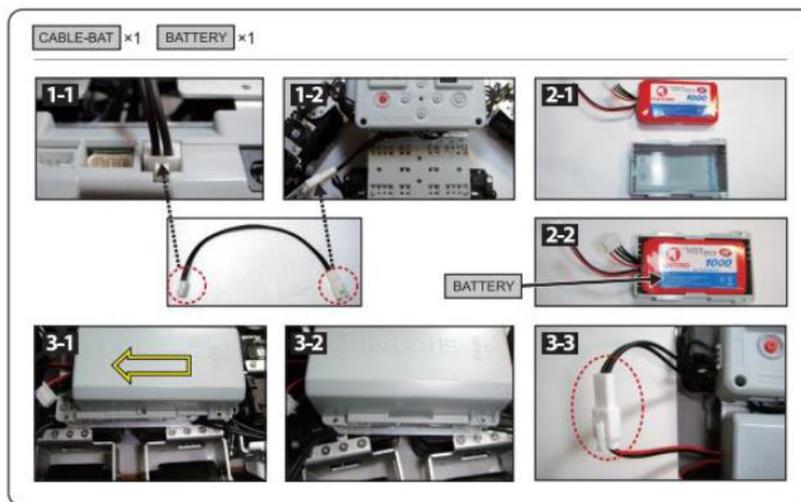


Fig. 51 Colocación de la batería del robot. [17]

3.2. ALGORITMOS DE RECONOCIMIENTO FACIAL

Los principios del uso del reconocimiento de la cara se brindaron a partir de los años 1960 con un sistema semiautomático. La técnica usada era el hacer marcas en las fotografías para limitar las propiedades primordiales; empleaba propiedades como ojos, oídos, narices y boca, este sistema necesitaba de un “administrador” para ubicar dichos aspectos. [18]

3.2.1 Librería OpenCV

Para este objetivo, eligió la librería OpenCV debido a que es una biblioteca abierta, que tiene algoritmos optimizados y contiene algoritmos de enseñanza automático y perspectiva por PC tradicionales.

“La librería OpenCV o en inglés (Open Source Computer Vision Library) fue desarrollada inicialmente por Intel en 1999, es una librería multiplataforma y usada de manera libre con propósitos investigativos o comerciales con los términos y condiciones establecidos. Contiene más de 500 funciones que abarcan varias áreas en el proceso de visión, como reconocimiento de objetos o reconocimiento facial, calibración de cámaras, visión estereoscópica, visión robótica, entre otras.

Usa su programación en código C y C++ optimizados, aprovechando también las capacidades que proveen los procesadores multi núcleo”.

“El componente CV incluye el procesamiento de imágenes y los algoritmos de visión por computadora de alto nivel. MLL es la librería de Machine learning, la cual incluye varios clasificadores estadísticos y herramientas de clustering.

HighGUI contiene rutinas de E/S y funciones para almacenar y cargar video e imágenes, y finalmente, CXCore contiene la estructura básica y contenida”. [18]

3.2.2 Reconocimiento facial con OpenCV

OpenCV contiene una clase abstracta que se llama `face_recognition` que se utiliza únicamente al reconocimiento facial y utiliza estos algoritmos:

- Eigenfaces (PCA), `EigenFaceRecognizer ()`
- Fisherfaces (LDA), `createFisherFaceRecognizer ()`
- Histogramas de patrones binarios locales (LBPH), `createLBPHFaceRecognizer ()`
- `Haarcascade_frontalface_default.xml`

Para esto se necesita dos bibliotecas extras que son: `dlib` y `face_recognition`.

3.2.3 Librería Face Recognition

La librería `face_recognition` contiene dentro de ella la funcionalidad del `dlib`. Es una librería de Python elaborada con deep learning que trae herramientas que nos ayuda al reconocimiento y manipulación de carass en una imagen o video de una manera sencilla permitiendo medir que el parecido de una persona con respecto a otra.

Está red neuronal convolucional tipo ResNet con 27 capas y el descriptor HOG tiene un margen del 99.38% según lo descrito en página oficial.

La librería se puede copiar desde Github o utilizar vía Git en el proyecto. “Dlib utiliza el detector facial basado en histograma de gradientes orientados (HOG)”.[18]

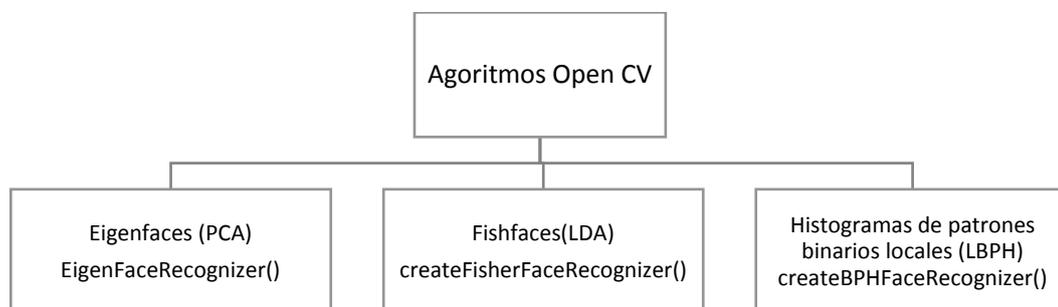


Fig. 52 Esquema de Algoritmos Open CV

3.2.4 Eigenfaces (PCA)

“Es una técnica que permite determinar, mediante la ortogonalidad dimensional, qué vectores ofrecen más información a un conjunto de datos de dimensión N . No obstante, la información N dimensional obtenida con Eigenfaces contiene datos redundantes que solo ocasionan que un sistema de clasificación tenga un alto costo computacional.

Para minimizar esto, se aplica análisis de componentes principales (PCA) que consiste en tomar una cantidad menor de los vectores entregados por las imágenes de la base de datos, pero con información necesaria para la reconstrucción de los rostros de las imágenes ingresadas para lograr con esta técnica disminuir el costo computacional al realizarse el procesamiento de datos”.

Eigenfaces, es una unión de vectores que se representan de manera gráfica y se convierten de manera de mapa para mencionar las modificaciones que se tiene entre imágenes en un espacio multidimensional. Esto se realiza a través de una comparación entre: los pesos requeridos para elaborar la imagen de la cara que no se conoce y los pesos requeridos para la elaboración de las imágenes de las caras que son conocidas, y como resultado se obtiene el reconocimiento.

3.2.5 Fisherfaces (LDA)

“Fisherface es una técnica de reconocimiento de rostros, que tiene en cuenta la luz y expresiones faciales, se encarga de clasificar y reducir las dimensiones de los rostros usando el método FLD (Discriminant Lineal Fisher).

En este sentido, el análisis discriminante de Fisher intenta proyectar los datos de manera que su nueva dispersión sea óptima para la clasificación.

Mientras PCA busca los vectores que mejor describen los datos, LDA (Discriminant Lineal Analysis) busca los vectores que proporcionan mejor discriminación entre clases después de la proyección”.

“Fisherfaces realiza un LDA, donde se busca aprovechar la información disponible, sobre la clasificación de las imágenes de entrenamiento, para buscar una proyección que maximice la separación entre imágenes de diferentes personas (o clases) y minimice la distancia entre imágenes de una misma clase. Así logra concentrar las imágenes mejorando, en forma importante, la tasa de reconocimiento”.

3.2.6 Histogramas de patrones binarios locales (LBPH)

“El método de patrones binarios locales fue diseñado para la descripción de texturas”.

“El uso de descripciones locales en algunas regiones del rostro aporta más información que otras, por lo que los descriptores de texturas tienden a promediar la información que describen, lo cual no es conveniente al describir rostros puesto que mantener la información de las relaciones espaciales es importante”.

“Para formar la descripción global, la imagen del rostro es dividida en diferentes regiones, a las que se les aplica un histograma con el que se obtiene el operador LBPH que describe información independiente por región, estas descripciones locales son entonces concatenadas para construir una descripción global del rostro”.

Como primero se debe realizar la parte computacional del LBPH que es básicamente la creación de una imagen intermedia que detalle la imagen original, resaltando las características faciales. Para ello, el algoritmo utiliza un concepto de ventana deslizante, basado en los parámetros radio y vecinos.

La siguiente imagen **¡Error! No se encuentra el origen de la referencia.** muestra este procedimiento:

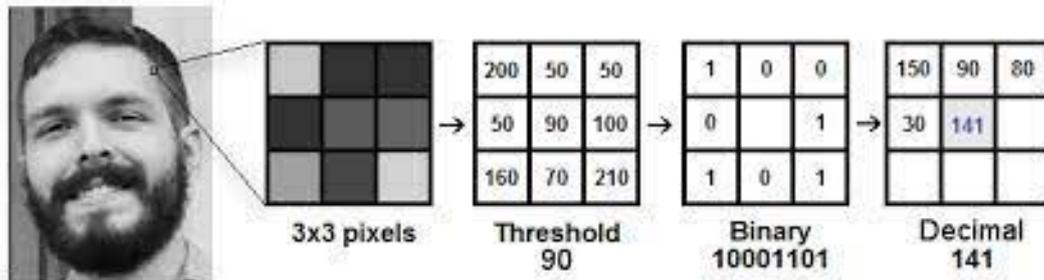


Fig. 53 Muestra del procesamiento de imágenes

Tabla I

TABLA INFORMATIVO DE VENTAJAS Y DEVENTAJAS

Algoritmo	Ventajas	Desventajas
Eigenface(PCA)	<ul style="list-style-type: none"> • La implementación es sencilla, ya que el algoritmo que utiliza es utilizado desde el inicio para técnicas de reconocimiento del rostro. • El tiempo del proceso es rápido. 	<ul style="list-style-type: none"> • La sensibilidad de iluminación. • Se debe buscar parecidos en tamaño o localización entre imágenes, con esto podría necesitar realizar un pre procesamiento a las capturas antes de aplicar este método.
Fisherfaces(LDA)	<ul style="list-style-type: none"> • La variación de la luz de las imágenes no es problema en la ubicación de clases, gracias a la distancia que se tiene entre imágenes de rostros iguales. 	<ul style="list-style-type: none"> • En esto es necesario varias imágenes del mismo rostro.
Histogramas de patrones binarios locales (LBPH)	<ul style="list-style-type: none"> • La implementación de este método es eficiente. • Se recomienda trabajar con las expresiones de la cara y las posturas que puedan tener la cabeza, ya que está basado en las características faciales. 	<ul style="list-style-type: none"> • Cuando se tiene cambios con la luz de manera radical.

3.2.7 Cascadas de Haar

El barrido de objetos con la utilización de la clasificación en cascada utilizando funcionalidades de Haar es utilizada para el barrido de objetos que ha sido postulado por Paul Viola y Michael en 2001.

“Es un enfoque basado en el aprendizaje automático en el que se entrena una función en cascada a partir de una gran cantidad de imágenes. Luego se usa este archivo de entrenamiento para detectar objetos en otras imágenes”.

En el proyecto, vamos a trabajar con la detección de rostros. Es decir, “este clasificador funciona a partir de un entrenamiento previo realizado con unos cientos de vistas de ejemplos de un objeto en particular (por ejemplo, una cara), llamados ejemplos positivos, y también entrenado con ejemplos negativos.

Luego que el clasificador es entrenado, puede ser aplicado a regiones de interés en una imagen de entrada. La salida del clasificador marca 1 si la región es congruente con el objeto (por ejemplo, una cara), y 0 en el caso contrario.

Cada característica es un valor único obtenido restando la suma de píxeles debajo del rectángulo blanco de la suma de píxeles debajo del rectángulo negro. A la hora de entrenar al clasificador, se buscan combinaciones de estas características en las imágenes de ejemplo y se eligen las más significativas para que formen parte del clasificador”.

OpenCV nos otorga muchos clasificadores de Haar en formato XML, que depende del uso que se desee dar, esto va desde el enfoque de caras de gatitos hasta la de las personas, emociones, siluetas, etc. Como guía, estos barridos se encuentran en la librería de GitHub. Para este proyecto se eligió el barrido `haarcascade_frontalface_default.xml` debido a las pruebas que se detallarán en el siguiente capítulo, basados en los falsos positivos que arrojan cada clasificador. Como se detalla la Fig. 54

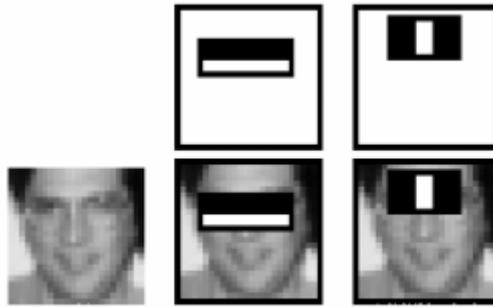


Fig. 54 Procesamiento de imágenes

3.3. INTERFAZ GRÁFICA

Algoritmo Implementado en código Python, con se detalla la Fig. 55

En el ANEXO

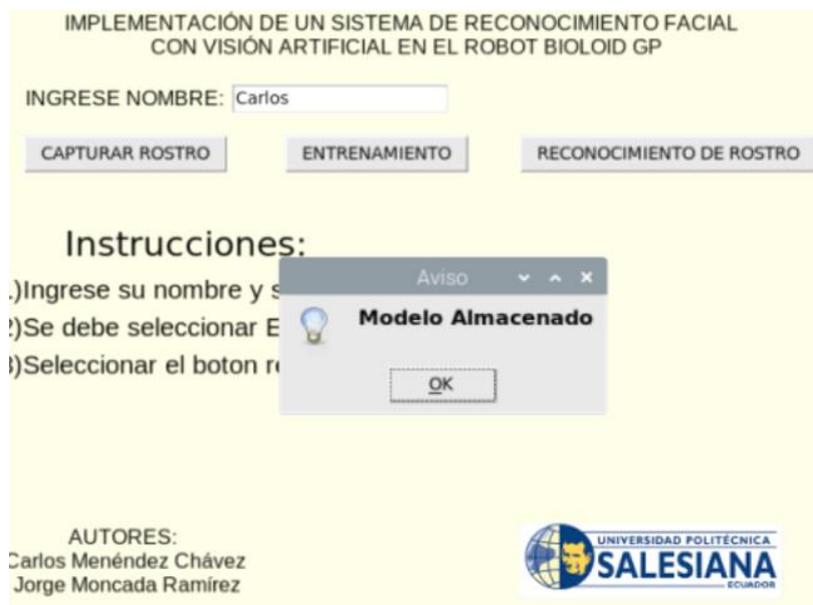


Fig. 55 GUI de la Implementación del sistema

Es por ello que podemos decir que el proceso para desarrollar la interfaz gráfica se divide de la siguiente manera:

1. Inicio de Interfaz
2. Captura de rostros
3. Entrenamiento del algoritmo

4. Reconocimiento de rostro

Como primer punto se tiene el inicio de la interfaz gráfica en donde se muestra el título del proyecto, se debe ingresar un nombre de la persona a quién se solicita capturar rostro, realizar su entrenamiento y seleccionar el reconocimiento del rostro. Fig. 56

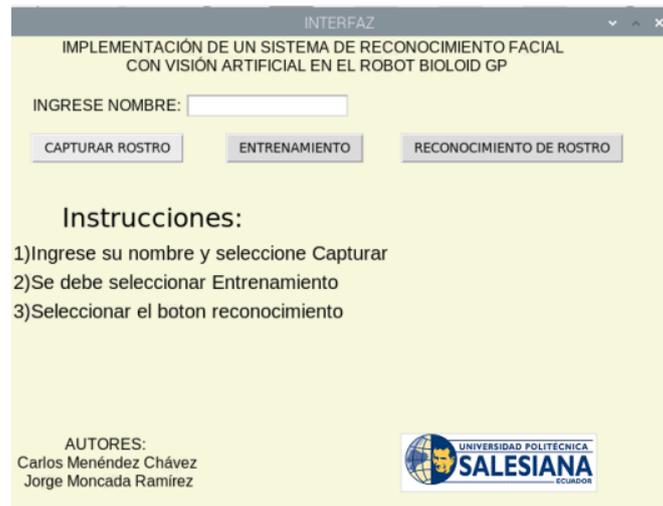


Fig. 56 Ventana del sistema

Dentro de este paso es el ingreso del nombre y la captura del rostro, como se muestra en la Fig. 57

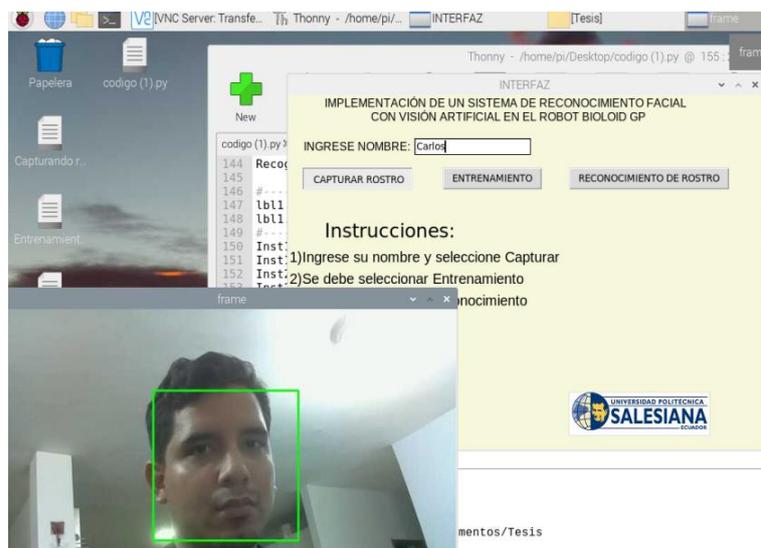


Fig. 57 Encendido del sistema y captura del rostro

Se realiza el entrenamiento, como se detalla en la Fig. 58.

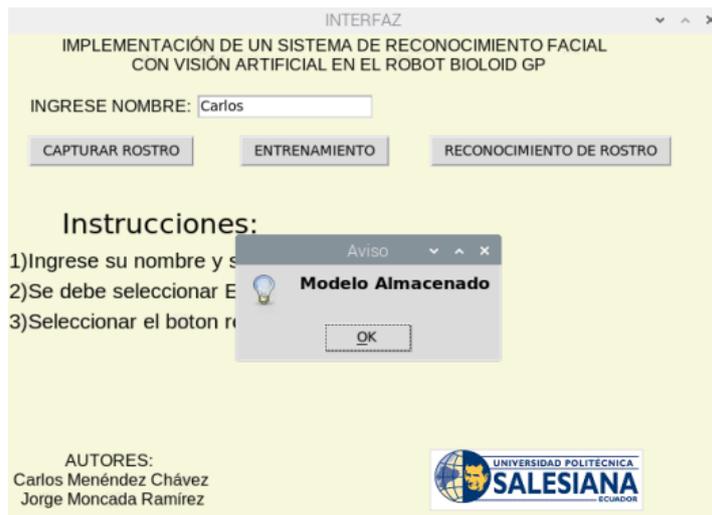


Fig. 58 Aviso de Almacenamiento

Se reconoce el rostro capturarado y entrenado, como se detalla en la Fig. 59

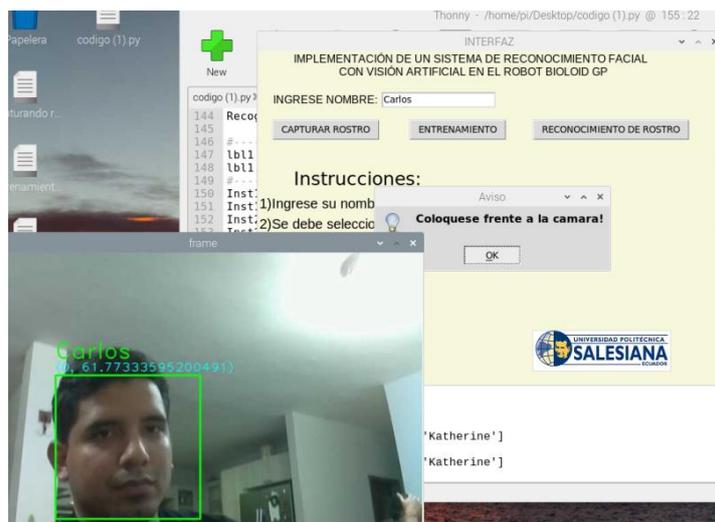


Fig. 59 Vista de ventana de entrenamiento

VERIFICACIÓN Y FUNCIONAMIENTO

4.1. RESULTADOS

4.2. En esta parte se demuestran los resultados logrados en este plan. Como primer punto se tiene el ensamble de la composición del robot Bioloid GP (ver imagen de abajo) y el resultado de la implementación, como se observa en la Fig. 60.



Fig. 60 Ensamblaje del robot humanoide.

Posteriormente se procedió con el análisis de los diferentes algoritmos de reconocimiento facial con la finalidad de obtener el más conveniente para la realización de este proyecto.

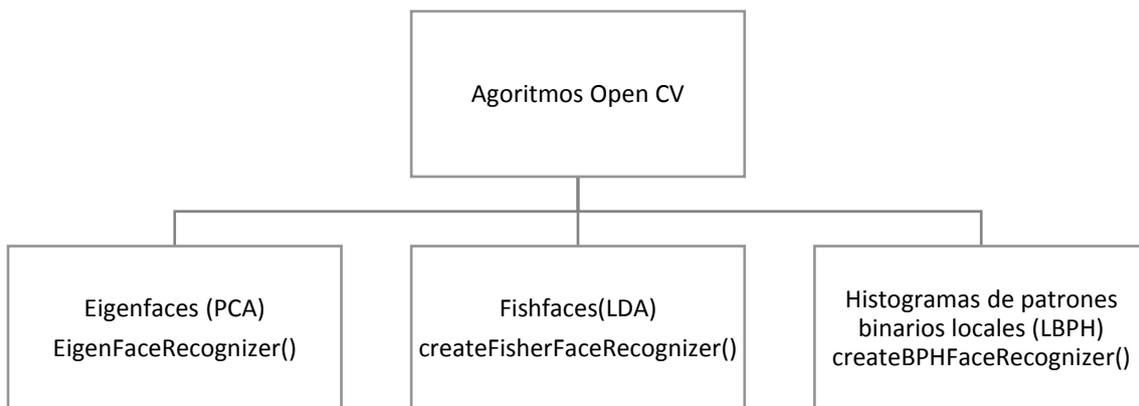


Fig. 61 Esquema del Algoritmo Open CV

Luego se seleccionó e implementó el algoritmo de reconocimiento facial para su uso de su reconocimiento el cuál fue el algoritmo LBPH que nos permitió producir una imagen intermedia que describa mejor la imagen original, destacando las propiedades faciales como se observa en la Fig. 62 y Fig. 63.

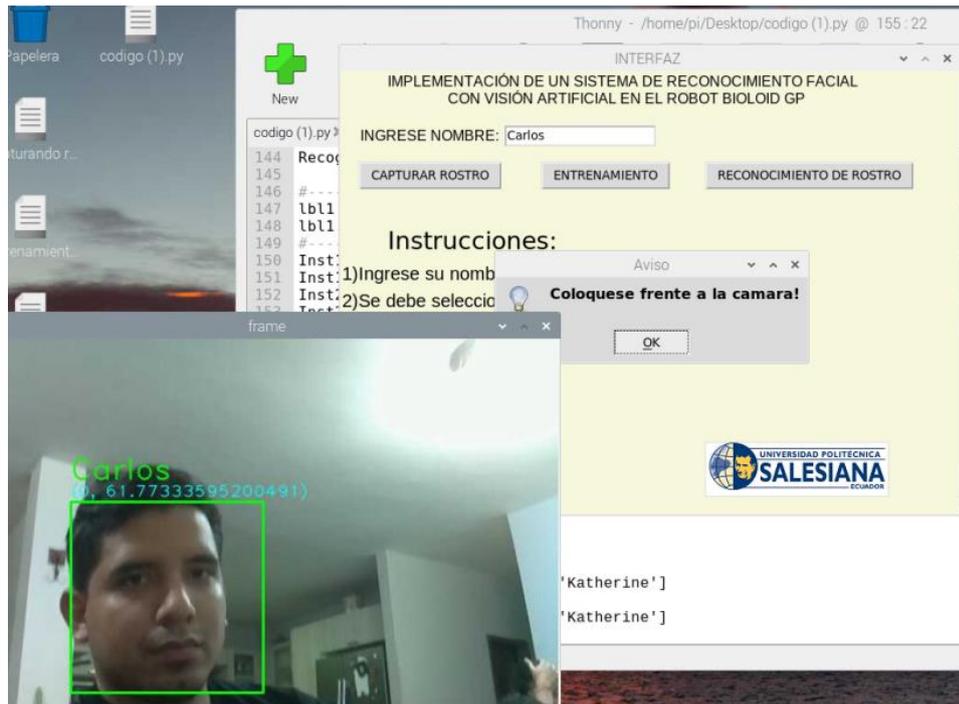


Fig. 62 Despligue del software



Fig. 63 Robot en funcionamiento

Al final, se muestra el resultado obtenido el desplazamiento del robot al instante de reconocer un cara anteriormente guardado por nuestra interfaz gráfica.

Práctica #1: Operación del robot Bioloid GP con control remoto en modo estándar.

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA: Ing. Electrónica
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Operación del robot Bioloid GP con control remoto en modo estándar.
OBJETIVO: Objetivo General: <ul style="list-style-type: none"> - Manejar el Robot Bioloid Gp con el control remoto en el modo estándar Objetivo Específico: <ul style="list-style-type: none"> - Subir la programación del modo estándar al controlador CM-530. - Comprobar el modo estándar con el control remoto. - Llevar a cabo conjunción de botones, y comprobar el cumplimiento de los movimientos del robot. 		
INSTRUCCIONES:		<ol style="list-style-type: none"> 1. Verificar el detalle de la práctica #1 en los Anexos del libro de titulación. 2. Continuar el marco procedimental de la práctica #1 de los Anexos del libro de titulación 3. Comprobar el nivel de carga en la batería del robot antes de operar el mismo.
ACTIVIDADES POR DESARROLLAR		
<ol style="list-style-type: none"> 1. Subir la programación del robot al controlador CM-530 usando el programa RobotPlus Task. 2. Comenzar a realizar las combinaciones de botones y de esta forma el robot ejecute los movimientos anteriormente definidos y programados en el robot. 		
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> - La sincronización del controlador CM-530 y el control remoto. - Realización de movimientos y operación idónea del robot. 		
CONCLUSIONES: <ul style="list-style-type: none"> - Se observa el conveniente manejo del robot. - Se consigue operar el robot con el control remoto y hacer las diversas combinaciones de movimientos. 		
RECOMENDACIONES: <ul style="list-style-type: none"> - Revisar la adecuada carga de la programación del robot. - Revisar el voltaje de las baterías del control como del robot. 		

BOTONES (CONTROL REMOTO)	ACCIÓN	BOTONES	BOTONES (CONTROL REMOTO)
U	ADELANTE	D	ATRÁS
L	GIRO (IZQUIERDA)	R	GIRO (DERECHA)
L+5	Paso Lateral izquierda lento	L+5+6	Paso lateral izquierda rápido
L+U+5	Paso lateral izquierdo + Paso adelante	L+D+5	Paso lateral izquierdo + paso atrás
R+5	Paso lateral derecho (lento)	R+5+6	Paso lateral derecho (rápido)
R+U+5	Paso lateral derecho + paso adelante	R+D+5	Paso lateral derecho + paso atrás
1+U	Se eleva boca abajo	1+D	Se eleva boca arriba

Práctica #2: Operación del robot Bioloid GP con control remoto en modo soccer.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Ing. Electrónica	
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Operación del robot Bioloid GP con control remoto en modo soccer.	
OBJETIVO: Objetivo General: <ul style="list-style-type: none"> - Manejar el Robot Bioloid Gp con el control remoto en el modo soccer. Objetivo Específico: <ul style="list-style-type: none"> - Subir la programación del modo soccer al controlador CM-530. - Comprobar el modo soccer con el control remoto. - Llevar a cabo conjunción de botones, y comprobar el cumplimiento de los movimientos del robot. 			
INSTRUCCIONES:		1. Verificar el detalle de la práctica #2 en los Anexos del libro de titulación.	
		2. Continuar el marco procedimental de la práctica #2 de los Anexos del libro de titulación	
		3. Comprobar el nivel de carga en la batería del robot antes de operar el mismo.	
ACTIVIDADES POR DESARROLLAR			
1. Subir la programación del robot al controlador CM-530 usando el programa RobotPlus Task			
2. Comenzar a realizar las combinaciones de botones y de esta forma el robot ejecute los movimientos anteriormente definidos y programados en el robot.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> - La sincronización del controlador CM-530 y el control remoto. - Realización de movimientos y operación idónea del robot. 			
CONCLUSIONES: <ul style="list-style-type: none"> - Se observa el conveniente manejo del robot en modo soccer. - Se consigue operar el robot con el control remoto y hacer las diversas combinaciones de movimientos. 			
RECOMENDACIONES: <ul style="list-style-type: none"> - Revisar la adecuada carga de la programación del robot. - Revisar el voltaje de las baterías del control como del robot. 			

BOTONES (CONTROL REMOTO)	ACCIÓN	BOTONES (CONTROL REMOTO)	ACCIÓN
U	ADELANTE	D	ATRÁS
L	GIRO (IZQUIERDA)	R	GIRO (DERCHA)
2+U	Patada Izquierda. para adelante	4+U	Patada derecha. para adelante
2+D	Patada Izquierda. para atrás	4+D	Patada Derecha. para atrás
2+L	Patada lateral Izquierda.	4+L	Patada lateral derecha.
2+R	Patada diagonal Izquierda.	4+R	Patada diagonal derecha.
3	Modo defensa	3+L	Modo defensa izquierda.
3+U	Modo defensa ambas manos	3+R	Modo defensa derecha.

Práctica #3: Programación de captura, detección, entrenamiento y reconocimiento de rostros en Python.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Ing. Electrónica	
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Programación de captura, detección, entrenamiento y reconocimiento de rostros en Python.	
OBJETIVO: Objetivo General: - Realizar programación de captura, detección, entrenamiento y reconocimiento de rostros en Python. Objetivo Específico: - Cargar la programación de la captura, detección, entrenamiento y reconocimiento de rostros en la Raspberry PI. - Probar el funcionamiento de las programaciones. - Ejecutar programación en Python usando módulo de cámara Raspberry.			
INSTRUCCIONES:		<ol style="list-style-type: none"> 1. Verificar el detalle de la práctica #3 en los Anexos del libro de titulación. 2. Continuar el marco procedimental de la práctica #3 de los Anexos del libro de titulación 3. Comprobar el nivel de carga en la batería del robot antes de operar el mismo. 	
ACTIVIDADES POR DESARROLLAR			
<ol style="list-style-type: none"> 1. Conectar la Raspberry Pi a la red. 2. Conectar remotamente a la Raspberry. 3. Realizar el código para la captura de rostros. 4. Realizar el código para el entrenamiento del rostro mediante el modelo LPBH. 5. Realizar la programación del reconocimiento de rostros, usando un módulo de cámara. 			
RESULTADO(S) OBTENIDO(S): - La sincronización con la Raspberry y conexión mediante el VNC VIEWER. - Ejecución de la captura de rostro. - Ejecución del entrenamiento del rostro mediante el modelo LBPH. - Reconocimiento del rostro del usuario.			
CONCLUSIONES: - Se observa el correcto funcionamiento de las programaciones realizadas en Python. - Se logra la captura, entrenamiento y reconocimiento del rostro del usuario.			
RECOMENDACIONES: - Verificar que la programación este bien escrita. - Verificar que la programación funcione de manera correcta.			

192.168.100.129 (raspberrypi): VNC Viewer

frame

w/Desktop/Capturandorostro.py @ 42:24

Assistent ✕

Warnings
May be ignored if you are happy with your program.

[Capturandorostro.py](#)

- Line 10: Bad indentation. Found 1 spaces, expected 4
- Line 11: Bad indentation. Found 1 spaces, expected 4
- Line 21: Bad indentation. Found 1 spaces, expected 4
- Line 22: Bad indentation. Found 1 spaces, expected 4
- Line 23: Bad indentation. Found 1 spaces, expected 4
- Line 24: Bad indentation. Found 1 spaces, expected 4
- Line 25: Bad indentation. Found 1 spaces, expected 4

```
14 cap = cv2.VideoCapture(0)
15
16 faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
17 count = 0
18
19 while True:
```

Shell ✕

```
Python 3.9.2 (/usr/bin/python3)
>>> %Run Capturandorostro.py
>>> %Run Capturandorostro.py
```

Python 3.9.2

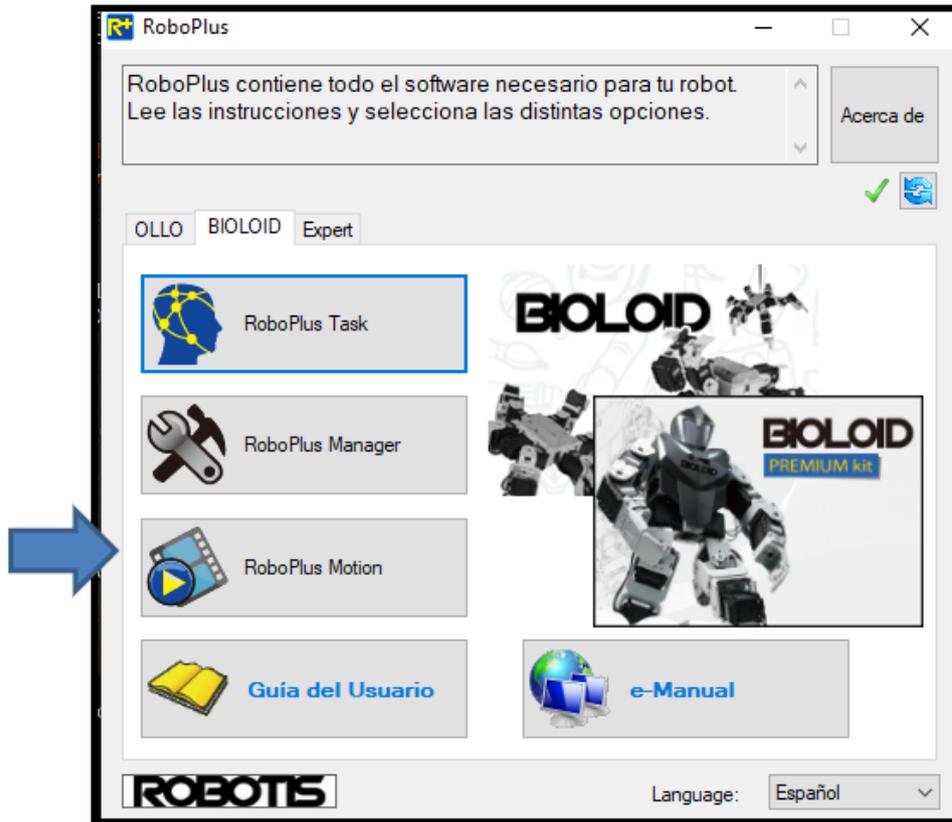
Práctica #4: Control de movimientos del robot Bioloid GP usando raspberry Pi mediante lenguaje de programación de Python.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Ing. Electrónica	
NRO. PRÁCTICA:	4	4.1. TÍTULO PRÁCTICA: Control de movimientos del robot Bioloid GP usando raspberry Pi mediante lenguaje de programación de Python.	
OBJETIVO: Objetivo General: <ul style="list-style-type: none"> - Control de Robot Bioloid GP usando Raspberry PI. Objetivo Específico: <ul style="list-style-type: none"> - Desarrollar programación para la comunicación entre el controlador CM-530 y la Raspberry pi. - Comprobar la comunicación entre el controlador CM-530 y raspberry pi, mediante lenguaje Python. 			
INSTRUCCIONES:		<ol style="list-style-type: none"> 1. Verificar el detalle de la práctica #4 en los Anexos del libro de titulación. 2. Continuar el marco procedimental de la práctica #4 de los Anexos del libro de titulación 3. Comprobar el nivel de carga en la batería del robot antes de operar el mismo. 	
ACTIVIDADES POR DESARROLLAR			
<ol style="list-style-type: none"> 1. Conectar la Raspberry Pi a una red cableada o inalámbrica para conectarse mediante VNC VIEWER. 2. Realizar la comunicación con el controlador CM-530. 3. Desarrollar el código en el editor Visual Studio Code o similar. 			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> - Sincronización con la Raspberry y conexión mediante el VNC VIEWER. - Comunicación con el controlador CM-530 			
CONCLUSIONES: <ul style="list-style-type: none"> - Se logro desarrollar la programación mediante Python. - Comunicación con el controlador CM-530 			
RECOMENDACIONES: <ul style="list-style-type: none"> - Verificar que la programación este bien escrita. - Verificar que la programación funcione de manera correcta. 			



Práctica #5: Crear movimientos del robot Bioloid GP en software ROBOPLUS.

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA: Ing. Electrónica
NRO. PRÁCTICA:	5	TÍTULO PRÁCTICA: Crear movimientos del robot Bioloid GP en software ROBOPLUS.
OBJETIVO: Objetivo General: - Crear movimientos para el Robot Bioloid GP mediante software Roboplus Objetivo Específico: - Utilizar software Roboplus motion para crear los movimientos en el robot. - Utilizar software Roboplus task para cargar movimientos en el robot.		
INSTRUCCIONES:		<ol style="list-style-type: none"> 1. Verificar el detalle de la práctica #5 en los Anexos del libro de titulación. 2. Continuar el marco procedimental de la práctica #5 de los Anexos del libro de titulación 3. Comprobar el nivel de carga en la batería del robot antes de operar el mismo.
ACTIVIDADES POR DESARROLLAR		
<ol style="list-style-type: none"> 1. Abrir la aplicación “robotplus motion”, 2. Conectar el controlador CM-530 mediante cable USB al ordenador que se va a utilizar. 		
<ol style="list-style-type: none"> 2. Realizar los pasos que se especifican en los anexos adjuntos en esta práctica. 		
RESULTADO(S) OBTENIDO(S): - La creación de movimientos en el software “roboplus motion”. - Subir el nuevo movimiento o nueva programación creada “Roboplus Task”.		
CONCLUSIONES: - Se logro crear un nuevo movimiento en “roboplus motion”. - Se logra subir una nueva programación o movimiento con “roboplus task”		
RECOMENDACIONES: - Verificar que la programación este bien escrita. - Verificar que la programación funcione de manera correcta.		

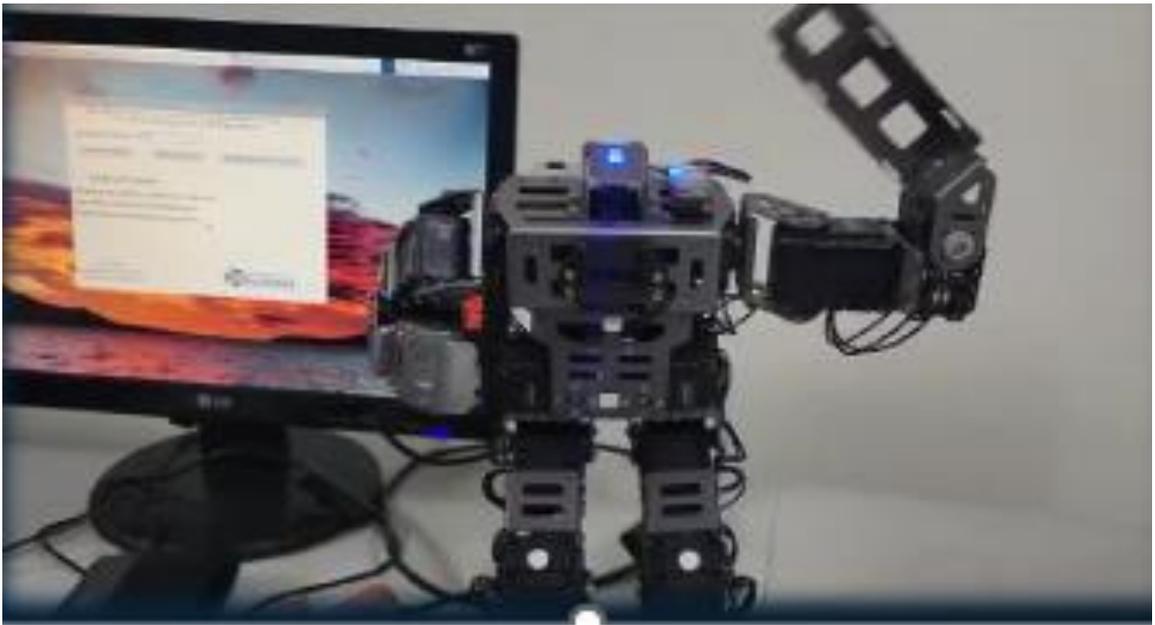


Nombre	Siguiente	Salir	Pausa	Tiempo
64	66	0	0	1
65	66	0	0	0
66	0	0	0	0
67 SideDumbling	68	0	0	0
68	69	0	0	0
69	70	0	0	0
70	0	0	0	0
71 Push up	72	0	0	0
72	73	0	0	0
73	74	0	0	0
74	75	0	0	0
75	76	0	0	0
76	0	0	0	0
77	0	0	0	0
▶ 78 SALUDAR	0	0	0	0
79	0	0	0	0
80	0	0	0	0
81 Sentadillas	0	0	0	0
82	0	0	0	0
83 Abdominales	30	0	0	0
84	0	0	0	0
85 Levantarse	0	0	0	0
86	0	0	0	0
87 Espera	0	0	0	0
88	0	0	0	0
89 R_5_Pausa	0	0	0	0
90 L_5_Pausa	0	0	0	0
91	0	0	0	0
92 Doble bow	0	0	0	0

Parámetros de página		Dureza de las	
Tiempo de	1	ID[1]	5
Velocidad:	1,5	ID[2]	5
Control de la fuerza	32	ID[3]	5
Tiempo de jecución	(1,000seg / 1,5) x 1 = 0min 0,656seg	ID[4]	5
		ID[5]	5
		ID[6]	5
		ID[7]	5
		ID[8]	5
		ID[9]	5
		ID[10]	5
		ID[11]	5
		ID[12]	5
		ID[13]	5
		ID[14]	5

4.2. Práctica #6: Realizar movimiento del robot bioloid Gp mediante el reconocimiento facial usando Raspberry PI (cámara Raspberry Pi).

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Ing. Electrónica	
NRO. PRÁCTICA:	6	TÍTULO PRÁCTICA: Realizar movimiento del robot bioloid Gp mediante el reconocimiento facial usando Raspberry PI (cámara Raspberry Pi).	
OBJETIVO: Objetivo General: <ul style="list-style-type: none"> - Efectuar el movimiento de un saludo en el robot Bioloid Gp al reconocer un rostro guardado en una base de datos. - Objetivo Específico: <ul style="list-style-type: none"> - Desarrollar interfaz gráfica para facilitar la comprensión del Reconocimiento Facial. - Realizar la captura de rostros mediante la cámara Raspberry PI - Entrenar el Algoritmo LBPH con los rostros capturados. - Reconocimiento del rostro y movimiento de saludo del Robot. 			
INSTRUCCIONES:		1. Verificar el detalle de la práctica #6 en los Anexos del libro de titulación.	
		2. Continuar el marco procedimental de la práctica #6 de los Anexos del libro de titulación	
		3. Comprobar el nivel de carga en la batería del robot antes de operar el mismo.	
ACTIVIDADES POR DESARROLLAR			
1. Crear la interfaz gráfica en Python.			
2. Realizar y desarrollar la programación de captura, entrenamiento y reconocimiento de rostros.			
3. Comprobar el movimiento de saludo al reconocer un rostro.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> - El Reconocimiento de rostros mediante la interfaz y la programación realizada en el Raspberry. - Ejecución de movimiento de saludo del robot Bioloid Gp al momento de reconocer un rostro. 			
CONCLUSIONES: <ul style="list-style-type: none"> - Se observa el correcto funcionamiento del robot. - Se logra el movimiento del robot mediante el reconocimiento de rostro. 			
RECOMENDACIONES: <ul style="list-style-type: none"> - Verificar que la programación este bien escrita. - Verificar que la programación funcione de manera correcta 			



CONCLUSIONES

En base a los resultados conseguidos del desarrollo de este presente trabajo de titulación nombrado, “IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL CON VISIÓN ARTIFICIAL EN EL ROBOT BIOLOID GP” se muestran las conclusiones conseguidas que surgieron desde el estudio llevado a cabo a las metas que fueron definidos en secciones anteriores:

- La estructura y composición del robot Bioloid GP es muy resistente a golpes y caídas, debido a que es un conjunto de materiales de alta calidad y con servomotores de excelente calidad, al igual que las matemáticas de programación de última generación han permitido que el robot se desempeñe bien en competiciones como acrobacias y ataques.
- La interfaz gráfica desarrollada en este proyecto está orientada para facilitar al usuario la captura, entrenamiento y reconocimiento de los rostros.
- El modelo LPBH Histogramas de patrones binarios locales (LBPH), en este proyecto fue el de mejor resultados al momento de realizar el reconocimiento de rostros.
- Con la preparación del manual de 6 prácticas se hizo documentar de forma organizada y descriptiva los métodos llevados a cabo para el ensamblaje de robot, brindando de esta forma una guía práctica para futuros trabajos.

RECOMENDACIONES

- El presente proyecto pretendió obtener el mayor alcance posible teniendo en cuenta los objetivos planteados, sin embargo, durante el proceso de desarrollo del mismo, fue posible observar aspectos que pueden ser mejorados en caso de mejorar el sistema.
- Se puede manifestar la necesidad de utilizar una cámara de mejor calidad que permita obtener imágenes de mejor calidad con un tamaño reducido.
- Se propone un conveniente ámbito tomando como elemento fundamental la localización de la cámara para eludir las variaciones luminosas desmesuradas que podrían provocar siluetas en las imágenes faciales al instante de hacer captura de las imágenes, para de esta forma afirmar el conveniente manejo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. C. Cabrera Bernal y J. B. Zambrano Ostaiza, «UNIVERSIDAD CATÓLICA,» 12 Septiembre 2016. [En línea]. Available: <http://201.159.223.180/bitstream/3317/6605/1/T-UCSG-PRE-TEC-ITEL-163.pdf>. [Último acceso: 7 marzo 2020].
- [2] H. A. Matovelle Rentería y K. N. Remache Luque, «Universidad Politecnica Salesiana,» 9 febrero 2021. [En línea]. Available: <http://dspace.ups.edu.ec/handle/123456789/21827>. [Último acceso: 7 marzo 2022].
- [3] J. C. Brenes Torres, «UNIVERSITAT POLITÈCNICA DE VALÈNCIA,» Septiembre 2016. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/77757/BRENES%20-%20API%20de%20generaci%C3%B3n%20de%20trayectorias%20para%20robot%20Bioloid%20sobre%20Beaglebone%20Black.pdf?sequence=1>. [Último acceso: 7 marzo 2022].
- [4] «Emanual Robotis,» [En línea]. Available: <https://emmanual.robotis.com/docs/en/edu/bioloid/gp/>. [Último acceso: 02 Mayo 2022].
- [5] G. Jeronimo, P. Bertolucci, R. L. Fialho, W. Tanaka y M. Bruno, «Humanoid soccer player and educational robotic: Development an architecture to connect the dynamixel ax-12a servo motor from robotis to the raspberry pi 3b,» *16th International Conference on Information Technology-New Generations (ITNG 2019)*, 2019.

- [6] C. G. Navarrete Valencia y S. D. Vallejo Santana, «Universidad Politecnica Salesiana,» 2022. [En línea]. [Último acceso: 24 Julio 2022].
- [7] «Emanual Robotis,» [En línea]. Available: <https://emmanual.robotis.com/docs/en/parts/communication/bt-410-dongle/#certifications>. [Último acceso: 02 Mayo 2022].
- [8] «Dynamo Electronics,» [En línea]. Available: <https://dynamoelectronics.com/baterias-lipo-caracteristicas-y-cuidados/>. [Último acceso: 02 Mayo 2022].
- [9] «TINYCIRCUITS,» [En línea]. Available: <https://tinycircuits.com/products/lithium-ion-polymer-battery-3-7v-1000mah>. [Último acceso: 24 Julio 2022].
- [10] «Sandorobotics,» [En línea]. Available: <https://sandorobotics.com/producto/902-0062-000/>. [Último acceso: 02 Mayo 2022].
- [11] «Emanual Robotis,» [En línea]. Available: <https://emmanual.robotis.com/docs/en/parts/controller/cm-530/>. [Último acceso: 02 Mayo 2022].
- [12] «Tienda Bricogeek,» [En línea]. Available: <https://tienda.bricogeek.com/placas-raspberry-pi/1330-raspberry-pi-4-model-b-4-gb.html>. [Último acceso: 02 Mayo 2022].
- [13] «Amazon,» [En línea]. Available: <https://www.amazon.com/-/es/Raspberry-m%C3%B3dulo-c%C3%A1mara-megapixeles-1080p/dp/B01ER2SKFS>. [Último acceso: 02 Mayo 2022].

- [14] «robotis,» [En línea]. Available: <https://robotis.co.uk/software/roboplus-1-0.html>. [Último acceso: 02 Mayo 2022].
- [15] «OSI,» [En línea]. Available: <https://www.osi.es/es/herramientas-gratuitas/vnc-viewer>. [Último acceso: 02 Mayo 2022].
- [16] N. Florián, «Aplicación del programa "Ciencia Divertida" basado en el método experimental para mejorar la actitud científica en el componente mundo físico y conservación del medio ambiente del área ciencia y ambiente en los alumnos del quinto grado de educación,» Universidad Privada Antenor Orrego - UPAO, Trujillo, 2016.
- [17] Robotis, Bioloid Gp Assembly Start, http://www.robotis.com/download/doc/BIO_GP_Humanoid_ASM_EN.pdf.
- [18] D. M. GUTIÉRREZ MENESES, «Dspace UPS,» 03 2021. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/19956/1/UPS-CT008985.pdf>. [Último acceso: 08 09 2022].
- [19] F. Mangkusasmito, «Visual servo strategies using linear quadratic Gaussian (LQG) for Yaw-Pitch camera platform,» *2018 International Conference on Signals and Systems*, 2018.

**ANEXOS A: MANUAL DE PRÁCTICAS DE ROBOT HUMANOIDE
BIOLOID GP**



UNIVERSIDAD POLITECNICA SALESIANA

SEDE GUAYAQUIL

INGENIERÍA ELECTRONICA

AUTORES:

MENENDEZ CHAVEZ CARLOS IVAN

MONCADA RAMIREZ JORGE DANIEL

ECUADOR – GUAYAQUIL

PRESENTACIÓN

Las presentes prácticas fueron desarrolladas en apego al programa de la Unidad de Aprendizaje de Electiva, con la finalidad de que los alumnos puedan poner en práctica los conocimientos teóricos que se van adquiriendo a lo largo del curso. Esta Unidad de Aprendizaje consta de la siguiente estructura:

1. Comprender los conceptos básicos de robótica móvil identificando los conceptos importantes.
2. Identificar los tipos de sensores utilizados en el ROBOT HUMANOIDE BIOLOID GP.
3. Conocer los programas a utilizar y sus funciones Cada Práctica consta de Título, Objetivos, Introducción, Material y Equipo a Utilizar, Duración, Desarrollo y Evaluación. El alumno debe replicar de forma práctica lo que se encuentra en el apartado de Desarrollo y resolver lo que se pide en el apartado de Evaluación.

INTRODUCCIÓN A LA ROBÓTICA

Actualmente la robótica ha cobrado gran importancia en la vida cotidiana y en procesos industriales. Se ha enfocado en las líneas de producción, donde el monitoreo de la producción desempeña un papel sumamente importante. Por lo que, desarrollar un sistema automatizado genera grandes beneficios.

El termino robot, viene de la palabra checa robota, que significa "labor forzada", servicio o esclavo. La robótica educativa es el conjunto de actividades pedagógicas que apoyan y fortalecen áreas específicas del conocimiento y desarrollan habilidades y competencias en el alumno, a través de los siguientes procesos: concepción, creación, ensamble y puesta en funcionamiento de robots.

Mediante la manipulación del Kit de Robótica, se lleva a cabo el ensamble de hasta 3 diferentes tipos de robots móviles (todo terreno, montacargas y pinzas), los cuales permiten ilustrar los principios y fundamentos básicos de la robótica móvil. La robótica móvil es una ciencia o rama de la tecnología que estudia el diseño y creación de máquinas que pueden desplazarse de un lugar a otro, cuya finalidad es desempeñar tareas repetitivas o peligrosas por el ser humano o que requieren el uso del pensamiento e inteligencia.

INTRODUCCIÓN A ROBOT HUMANOIDE BIOLOID GP

Los robots se utilizan en la educación para enseñar y desarrollar en el estudiante habilidades y destrezas que les sirvan para resolver problemas de la vida real.

El proceso de programación de un robot consiste en introducir en su sistema de control las instrucciones necesarias para que desempeñe las tareas para las que ha sido diseñado. Para programar un robot se sigue un proceso semejante al de la elaboración de un programa computacional destinado a cualquier otra aplicación (celular, tablet, sitio web, etc.).

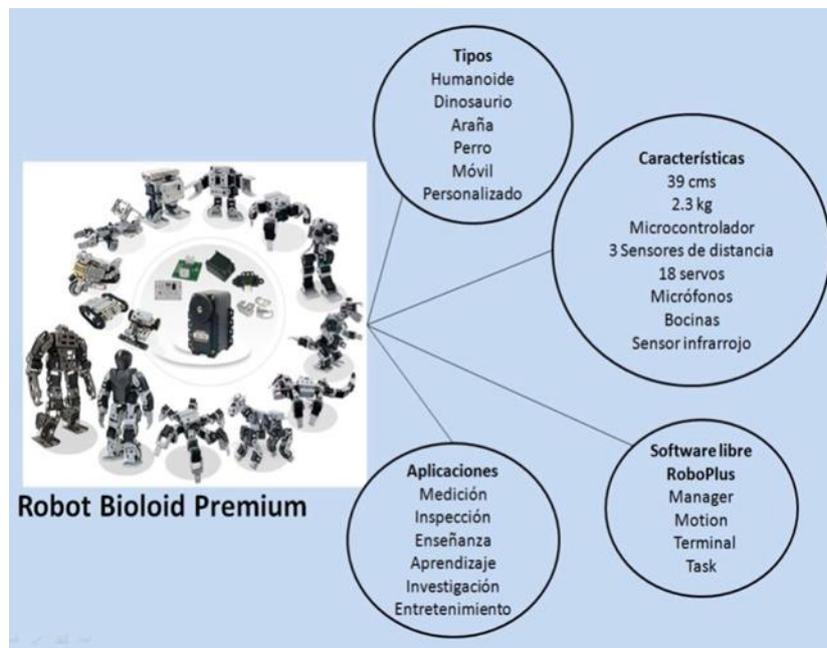
Primero será necesario establecer el algoritmo idóneo que permita al robot llevar a cabo las tareas para las que ha sido diseñado, tras lo cual se traducirá dicho algoritmo en un lenguaje de programación por el sistema de control del robot. Dicho lenguaje debe permitir especificar de forma clara y sencilla las tareas que debe realizar el robot.

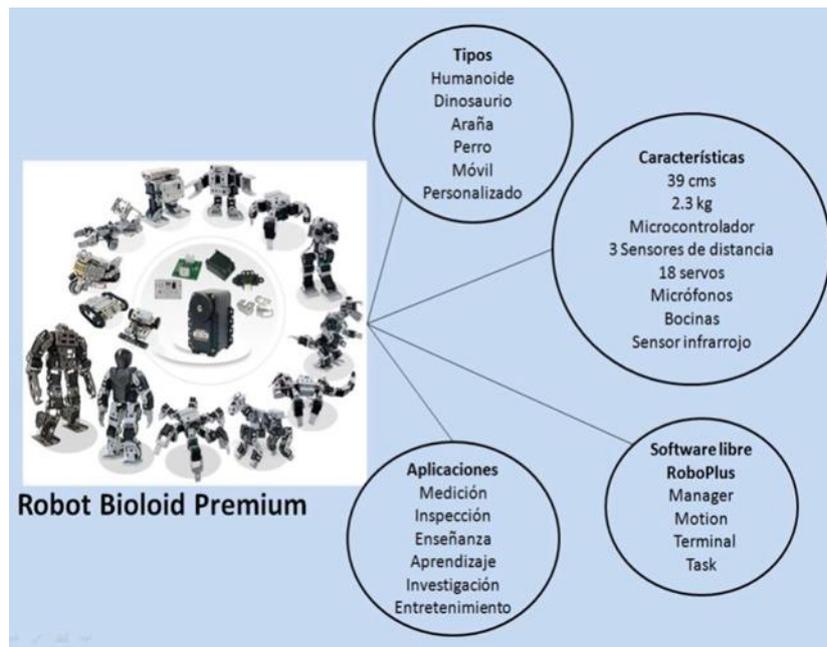
A continuación, se presentan diferentes tipos de procedimientos de programación de robots:

- a) Programación guiada o directa: El operario interviene guiando manualmente el brazo del robot, y hace que este vaya describiendo los movimientos y trace las trayectorias necesarias para cumplir su función. Cada uno de los movimientos realizados se va almacenando en la memoria del robot, de forma que podrán ser repetidos posteriormente, ya sin intervención humana. En este tipo de programación es necesario disponer del propio robot para la elaboración del programa.
- b) Programación textual o indirecta: En este caso no es necesaria la presencia del robot para realizar el programa, puesto que este se lleva a cabo en un lenguaje de programación. El programa consiste en un conjunto de instrucciones; cuando el

programa sea grabado en la memoria del robot, este realizará las acciones indicadas en el mismo. Este tipo de programación permite realizar operaciones más complejas y con mayor grado de precisión. Además, presenta la ventaja de que es posible establecer relaciones entre el robot y su entorno. Para ello basta con introducir en el programa los datos procedentes de los sensores de forma que el robot actúe en consonancia con los mismos, tal y como ocurre en los robots inteligentes. Este tipo de programación puede dividirse en 2 tipos: explícita (lenguajes de programación estructurados) y especificativa (lenguajes de programación orientados a objetos).

En el siguiente mapa conceptual se presenta un mapa conceptual sobre el Kit de Robótica Bioloid GP de la marca Robotis; el cual hace referencia a los diferentes tipos de configuraciones, características y en qué áreas se pueden utilizar.





Software RoboPlus: Es un software libre que sirve para trabajar con los diferentes modelos (Ollo, Bioid y Darwin Op) de la marca Robotis. Contiene las siguientes herramientas:



- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion
- e-Manual

PRÁCTICA # 1

TEMA:

“OPERACIÓN DEL ROBOT BIOLOID GP CON CONTROL REMOTO EN MODO ESTÁNDAR ”

OBJETIVO GENERAL:

- Operar el Robot Bioloid Gp con el control remoto en el modo estándar

OBJETIVOS ESPECIFICOS:

- Cargar la programación del modo estándar al controlador CM-530.
- Probar el modo estándar con el control remoto.
- Ejecutar combinación de botones, y verificar el cumplimiento de los movimientos del robot.

MARCO TEÓRICO:

El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado, el robot deberá medir la distancia evitando que este colisione con algún objeto usando el sensor DMS-80 y a la vez, cuando el prototipo este en marcha el giroscopio GS-12 deberá darle un control de estabilidad recibiendo una señal y enviándolo al controlador CM530 para poder dirigir la velocidad y fuerza de los servomotores, evitando que el robot pueda derrumbarse.

Robot Humanoide:

La robótica humanoide es una rama de la robótica que trata de que los robots imiten tanto en apariencia como en movimiento, ya sea locomoción, movimientos gestuales o movimientos en estático a los seres humanos. Son diseñados reproduciendo la forma de los brazos, de la cabeza y de los pies que los seres humanos. Pueden ser bípedos o pueden desplazarse por medio de una plataforma con ruedas en la parte inferior. Los robots más sofisticados buscan el hiperrealismo e intentan simular las expresiones de la cara, nuestros gestos y la piel. El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines. Por otro lado, el intento de simular el cuerpo humano conduce a una mejor comprensión de él. Además de la investigación, se están desarrollando robots humanoides para realizar tareas humanas como asistencia personal, a través de las cuales deberían poder ayudar a los enfermos y ancianos, y trabajos sucios o peligrosos. Los humanoides también son adecuados para algunas vocaciones basadas en procedimientos, como los administradores de mostradores de recepción y los trabajadores de la línea de fabricación de automóviles. En esencia, dado que pueden usar herramientas y operar equipos diseñados para la forma humana, los humanoides podrían, en teoría, realizar cualquier tarea que pueda hacer un ser humano, siempre que cuenten con el software adecuado. Sin embargo, la complejidad de hacerlo es inmensa

Algunos ejemplos de robots humanoides que encontramos en la actualidad son:

- Robot Sophia
- Robot Pepper
- Robot NAO
- Robot Atlas
- Robot ASIMO

Servomotores Dynamixel AX-12A:

El servomotor Dynamixel Ax-12A es un nuevo actuador robótico del sistema ROBOTIS Bioloid sustituye al antiguo Dynamixel Ax12+ con el mismo rendimiento, pero un diseño externo más avanzado. Cada actuador tiene un ID único lo cual le va a permitir ser identificado por el controlador, se puede usar un rango de 0 a 252. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 se puede operar con un solo conector adjunto.

RoboPlus:

Es un software libre que sirve para trabajar con los diferentes modelos de robots tales como: Ollio, Bioloid y Darwin Op, pertenecientes a la marca Robotis. Además, es compatible con el controlador CM-530, los servomotores Dynamixel y los sensores de Robotis los cuales se usarán en nuestro proyecto. Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

Receptor BT-410:

El receptor BT-410 tiene una comunicación Bluetooth 4.0 que será compatible con nuestro control 'Rc-100B' que nos ayudará para que nuestro prototipo pueda ser operado de una manera remota. Nos permite la comunicación en serie (UART) a través de Bluetooth y es montable a equipos de robótica, controlable a través de Smartphones, tablet PC, Laptop, etc.

Control RC-100B:

El control remoto Rc – 100B nos ayudará a comunicarnos con el robot de manera remota ya que admite su comunicación con los módulos Bluetooth, ZIGbee o infrarrojo en caso de que no se adapte modulo Bluetooth o ZIGbee su comunicación predeterminada será por infrarrojo.

Sensor gyro GS-12

El sensor Gyro GS-12 nos permitirá mantener el equilibrio de nuestro prototipo, ya que mide la velocidad angular y a su vez se puede utilizar para otras aplicaciones de movimiento. También, calcula que lado está inclinado el robot o está recibiendo más potencia.

Controlador CM-530

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido. Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

MARCO PROCEDIMENTAL

Cargar la programación del robot al controlador CM-530 usando el programa RobotPlus Task, el nombre del archivo es: “Control Remoto”, donde estará predefinido la programación del robot con los comandos establecidos en él, programa dichos comandos se utilizarán para poner en práctica los algoritmos previamente programados usando el control remoto, daremos marcha al robot y probaremos dicha programación.

Luego, se procede a cargar la programación del robot al controlador CM-530 usando el programa RobotPlus Task. Una vez que el programa se encuentre cargado en el controlador, pulsaremos en botón “U” en el controlador CM-530, para que ejecute su caminata estándar. Al presionar el botón, el controlador emitirá una melodía, que nos indicará que función vamos a ejecutar.

Para el uso del control utilizaremos los siguientes comandos previamente definidas, en la programación que hemos cargado a nuestro robot, que son los siguientes:

BOTONES	MOVIMIENTO	BOTONES	MOVIMIENTO
U	ADELANTE	D	ATRÁS
L	GIRO IZQ	R	GIRO DER
L+5	Paso Lateral izq lento	L+5+6	Paso lateral izq rápido
L+U+5	Paso lateral izq + Paso adelante	L+D+5	Paso lateral izq + paso atrás
R+5	Paso lateral derecho lento	R+5+6	Paso lateral derecho rapido
R+U+5	Paso lateral derecho + paso hacia adelante	R+D+5	Paso lateral derecho + paso hacia atrás
1+U	Se levanta boca abajo	1+D	Se levanta boca arriba

RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Robot Bioloid GP
- Controlador CM-530
- Programa Robot Plus

- Modulos BT 4.10 (Bluetooth).
- RC-100 (Control remoto)

ANEXOS:

Material de apoyo “eManual Robotis – Robot Bioloid Gp” de Robotis

PRÁCTICA # 2

TEMA:

“OPERACIÓN DEL ROBOT BIOLOID GP CON CONTROL REMOTO EN MODO SOCCER ”

OBJETIVO GENERAL:

- Operar el Robot Bioloid Gp con el control remoto en el modo estándar.

OBJETIVOS ESPECIFICOS:

- Cargar la programación del modo estándar al controlador CM-530.
- Probar el modo soccer con el control remoto.
- Ejecutar combinación de botones, y verificar el cumplimiento de los movimientos del robot.

MARCO TEÓRICO:

El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado, el robot deberá medir la distancia evitando que este colisione con algún objeto usando el sensor DMS-80 y a la vez, cuando el prototipo este en marcha el giroscopio GS-12 deberá darle un control de estabilidad recibiendo una señal y enviándolo al controlador CM530 para poder dirigir la velocidad y fuerza de los servomotores, evitando que el robot pueda derrumbarse.

Robot Humanoide:

La robótica humanoide es una rama de la robótica que trata de que los robots imiten tanto en apariencia como en movimiento, ya sea locomoción, movimientos gestuales o movimientos en estático a los seres humanos. Son diseñados reproduciendo la forma de los brazos, de la cabeza y de los pies que los seres humanos. Pueden ser bípedos o pueden desplazarse por medio de una plataforma con ruedas en la parte inferior. Los robots más sofisticados buscan el hiperrealismo e intentan simular las expresiones de la cara, nuestros gestos y la piel. El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines. Por otro lado, el intento de simular el cuerpo humano conduce a una mejor comprensión de él. Además de la investigación, se están desarrollando robots humanoides para realizar tareas humanas como asistencia personal, a través de las cuales deberían poder ayudar a los enfermos y ancianos, y trabajos sucios o peligrosos. Los humanoides también son adecuados para algunas vocaciones basadas en procedimientos, como los administradores de mostradores de recepción y los trabajadores de la línea de fabricación de automóviles. En esencia, dado que pueden usar herramientas y operar equipos diseñados para la forma humana, los humanoides podrían, en teoría, realizar cualquier tarea que pueda hacer un ser humano, siempre que cuenten con el software adecuado. Sin embargo, la complejidad de hacerlo es inmensa

Algunos ejemplos de robots humanoides que encontramos en la actualidad son:

- Robot Sophia
- Robot Pepper
- Robot NAO
- Robot Atlas
- Robot ASIMO

Servomotores Dynamixel AX-12A:

El servomotor Dynamixel Ax-12A es un nuevo actuador robótico del sistema ROBOTIS Bioloid sustituye al antiguo Dynamixel Ax12+ con el mismo rendimiento, pero un diseño externo más avanzado. Cada actuador tiene un ID único lo cual le va a permitir ser identificado por el controlador, se puede usar un rango de 0 a 252. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 se puede operar con un solo conector adjunto

RoboPlus:

Es un software libre que sirve para trabajar con los diferentes modelos de robots tales como: Ollio, Bioloid y Darwin Op, pertenecientes a la marca Robotis. Además, es compatible con el controlador CM-530, los servomotores Dynamixel y los sensores de Robotis los cuales se usarán en nuestro proyecto. Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

Receptor BT-410:

El receptor BT-410 tiene una comunicación Bluetooth 4.0 que será compatible con nuestro control 'Rc-100B' que nos ayudará para que nuestro prototipo pueda ser operado de una manera remota. Nos permite la comunicación en serie (UART) a través de Bluetooth y es montable a equipos de robótica, controlable a través de Smartphones, tablet PC, Laptop, etc.

Control RC-100B:

El control remoto Rc – 100B nos ayudará a comunicarnos con el robot de manera remota ya que admite su comunicación con los módulos Bluetooth, ZIGbee o infrarrojo en caso de que no se adapte modulo Bluetooth o ZIGbee su comunicación predeterminada será por infrarrojo.

Sensor Gyro GS-12:

El sensor Gyro GS-12 nos permitirá mantener el equilibrio de nuestro prototipo, ya que mide la velocidad angular y a su vez se puede utilizar para otras aplicaciones de movimiento. También, calcula que lado está inclinado el robot o está recibiendo más potencia.

Controlador CM-530:

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido.

Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

MARCO PROCEDIMENTAL:

Cargar la programación del robot al controlador CM-530 usando el programa RobotPlus Task, el nombre del archivo es: “Control Remoto”, donde estará predefinido la programación del robot con los comandos establecidos en él, programa dichos comandos se utilizarán para poner en práctica los algoritmos previamente programados usando el control remoto, daremos marcha al robot y probaremos dicha programación.

Una vez que el programa se encuentre cargado en el controlador, pulsaremos en botón “D” en el controlador CM-530, para que el modo soccer. Al presionar dicho botón, el controlador emitirá una melodía, que nos indicará que función vamos a ejecutar.

Para el uso del control utilizaremos los siguientes comandos previamente definidas, en la programación que hemos cargado a nuestro robot, que son los siguientes:

BOTONES	MOVIMIENTO	BOTONES	MOVIMIENTO
U	ADELANTE	D	ATRÁS
L	GIRO IZQ	R	GIRO DER
2+U	Patada Izq. Hacia adelante	4+U	Patada Izq. Hacia adelante
2+D	Patada Izq. Hacia atrás	4+D	Patada Der. Hacia atrás
2+L	Patada lateral Izq.	4+L	Patada lateral der.
2+R	Patada diagonal Izq.	4+R	Patada lateral der.
3	Modo defensa	3+L	Modo defensa izq.
3+U	Modo defensa ambas manos	3+R	Modo defensa der.

RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Robot Bioloid GP
- Controlador CM-530
- Programa Robot Plus
- Módulos BT 4.10 (Bluetooth).
- RC-100 (Control remoto)

ANEXOS:

Material de apoyo “eManual Robotis – Robot Bioloid Gp” de Robotis

PRÁCTICA # 3

TEMA:

“PROGRAMACIÓN DE CAPTURA, DETECCIÓN, ENTRENAMIENTO Y RECONOCIMIENTO DE ROSTROS EN PYTHON.”

OBJETIVO GENERAL:

- Realizar programación de captura, detección, entrenamiento y reconocimiento de rostros en Python.

OBJETIVOS ESPECIFICOS:

- Cargar la programación de la captura, detección, entrenamiento y reconocimiento de rostros en la Raspberry PI.
- Probar el funcionamiento de las programaciones.
- Ejecutar programación en Python usando módulo de cámara Raspberry.

MARCO TEÓRICO:

El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado, nuestra programación se implementa en la Raspberry pi empleando el lenguaje de Python, y mediante el código realizado se busca la captura, entrenamiento y reconocimiento de rostros.

Con esto buscamos el control del Robot Bioloid GP mediante la raspberry pi, con la finalidad de que el robot pueda realizar acciones al momento de reconocer un rostro almacenado en la base de datos.

Robot Humanoide:

La robótica humanoide es una rama de la robótica que trata de que los robots imiten tanto en apariencia como en movimiento, ya sea locomoción, movimientos gestuales o movimientos en estático a los seres humanos. Son diseñados reproduciendo la forma de los brazos, de la cabeza y de los pies que los seres humanos. Pueden ser bípedos o pueden desplazarse por medio de una plataforma con ruedas en la parte inferior. Los robots más sofisticados buscan el hiperrealismo e intentan simular las expresiones de la cara, nuestros gestos y la piel. El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines. Por otro lado, el intento de simular el cuerpo humano conduce a una mejor comprensión de él. Además de la investigación, se están desarrollando robots humanoides para realizar tareas humanas como asistencia personal, a través de las cuales deberían poder ayudar a los enfermos y ancianos, y trabajos sucios o peligrosos. Los humanoides también son adecuados para algunas vocaciones basadas en procedimientos, como los administradores de mostradores de recepción y los trabajadores de la línea de fabricación de automóviles. En esencia, dado que pueden usar herramientas y operar equipos diseñados para la forma humana, los humanoides podrían, en teoría, realizar cualquier tarea que pueda hacer un ser humano, siempre que cuenten con el software adecuado. Sin embargo, la complejidad de hacerlo es inmensa

Algunos ejemplos de robots humanoides que encontramos en la actualidad son:

- Robot Sophia

- Robot Pepper
- Robot NAO
- Robot Atlas
- Robot ASIMO

Servomotores Dynamixel AX-12A:

El servomotor Dynamixel Ax-12A es un nuevo actuador robótico del sistema ROBOTIS Bioloid sustituye al antiguo Dynamixel Ax12+ con el mismo rendimiento, pero un diseño externo más avanzado. Cada actuador tiene un ID único lo cual le va a permitir ser identificado por el controlador, se puede usar un rango de 0 a 252. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 se puede operar con un solo conector adjunto.

Controlador CM-530:

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido.

Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

RoboPlus:

Es un software libre que sirve para trabajar con los diferentes modelos de robots tales como: Ollio, Bioloid y Darwin Op, pertenecientes a la marca Robotis. Además, es compatible con el controlador CM-530, los servomotores Dynamixel y los sensores de Robotis los cuales se usarán en nuestro proyecto. Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

Receptor BT-410:

El receptor BT-410 tiene una comunicación Bluetooth 4.0 que será compatible con nuestro control 'Rc-100B' que nos ayudará para que nuestro prototipo pueda ser operado de una manera remota. Nos permite la comunicación en serie (UART) a través de Bluetooth y es montable a equipos de robótica, controlable a través de Smartphones, tablet PC, Laptop, etc.

Control RC-100B:

El control remoto Rc – 100B nos ayudará a comunicarnos con el robot de manera remota ya que admite su comunicación con los módulos Bluetooth, ZIGbee o infrarrojo en caso de que no se adapte modulo Bluetooth o ZIGbee su comunicación predeterminada será por infrarrojo.

Sensor Gyro GS-12:

El sensor Gyro GS-12 nos permitirá mantener el equilibrio de nuestro prototipo, ya que mide la velocidad angular y a su vez se puede utilizar para otras aplicaciones de movimiento. También, calcula que lado está inclinado el robot o está recibiendo más potencia.

Raspberry Pi 4:

Raspberry Pi es un miniordenador que se utiliza desde centro multimedia hasta de cerebro en robots.

Software VNC Server-Viewer:

VNC por sus siglas en inglés Virtual Network Computing o en español, Computación Virtual en Red, es un software de código libre de tipo cliente servidor que permite ver la pantalla del ordenador servidor y controlarlo en uno o varios ordenadores clientes sin importar que sistema operativo pueda ejecutar el cliente o el servidor, podemos ver la pantalla y controlar el equipo del que ejecuta el servidor desde el cliente.

Módulo de cámara Raspberry Pi:

Módulo con Cámara diseñado especialmente para Raspberry Pi. Se conecta en uno de los dos pequeños sockets que tiene la tarjeta Raspberry en su cara superior. Esta conexión, usa la interfaz dedicada CSI, la cual es óptima para conectar una cámara gracias a su alta capacidad de transmisión de datos.

Receptor BT-410 Dongle:

BT-410 Dongle convierte la interfaz USB 2.0 y la interfaz de comunicación estándar Bluetooth 4.0 Low Energy (Master).

Si el Dongle BT-410 está conectado a la PC, se puede emparejar con un módulo esclavo BT-410 para intercambiar datos.

Lenguaje Python:

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

MARCO PROCEDIMENTAL:

Conectamos la Raspberry Pi a nuestra red, creada con el router xxx para poder conectarnos mediante VNC VIEWER y realizar nuestra programación en Python de forma remota.

Luego de conectarnos remotamente a nuestra Raspberry empezamos realizar el código para la captura de rostros, que se almacenara como data en una carpeta con el nombre (Ej.: USUARIO).

Seguido de esto con la data ya guardado, se debe realizar el código para el entrenamiento del rostro mediante el modelo LPBH (Local Binary Pattern Histograms) que es el más eficiente de los modelos investigados, al correr la programación se crea un archivo “modeloLBPH.xml” que nuestro modelo ya entrenado.

Por último, se realiza la programación del reconocimiento de rostros, usando nuestro módulo de cámara y obtenemos como resultado el reconocimiento del rostro del USUARIO.

PROGRAMACIÓN DE CAPTURA DE ROSTRO:

```
import cv2      #librerias
import os       #Libreria
import imutils  #Libreria

personName = 'Usuario'
```

```

dataPath = '/home/pi/Documents/Interfaz/prueba/DATA' #Cambia a la ruta
donde hayas almacenado Data
personPath = dataPath + '/' + personName

if not os.path.exists(personPath):
    print('Carpeta creada: ',personPath)
    os.makedirs(personPath)

cap = cv2.VideoCapture(0)
#cap = cv2.VideoCapture('video.mp4')

faceClassif =
cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_d
efault.xml')
count = 0

while True:

    ret, frame = cap.read()
    if ret == False: break
    frame = imutils.resize(frame, width=640)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = frame.copy()

    faces = faceClassif.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        rostro = auxFrame[y:y+h,x:x+w]
        rostro =
cv2.resize(rostro,(150,150),interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(personPath +
'/rotro_{}.jpg'.format(count),rostro)
        count = count + 1
        cv2.imshow('frame',frame)

    k = cv2.waitKey(1)
    if k == 27 or count >= 30:

```

PROGRAMACIÓN DE ENTRENAMIENTO:

```

import cv2
import os
import numpy as np

dataPath = '/home/pi/Documents/Interfaz/prueba/DATA' #Cambia a la ruta
donde hayas almacenado Data
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

labels = []
facesData = []
label = 0

for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName,0))
        #image = cv2.imread(personPath+'/'+fileName,0)
        #cv2.imshow('image',image)
        #cv2.waitKey(10)
    label = label + 1

```

```

#print('labels= ',labels)
#print('Número de etiquetas 0:
',np.count_nonzero(np.array(labels)==0))
#print('Número de etiquetas 1:
',np.count_nonzero(np.array(labels)==1))

# Métodos para entrenar el reconocedor
#face_recognizer = cv2.face.EigenFaceRecognizer_create()
#face_recognizer = cv2.face.FisherFaceRecognizer_create()
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Entrenando el reconocedor de rostros
print("Entrenando...")
face_recognizer.train(facesData, np.array(labels))

# Almacenando el modelo obtenido
#face_recognizer.write('modeloEigenFace.xml')
#face_recognizer.write('modeloFisherFace.xml')
face_recognizer.write('modeloLBPHFace.xml')
print("Modelo almacenado...")

PROGRAMACIÓN DE RECONOCIMIENTO FACIAL
import cv2
import os
import time
dataPath = '/home/pi/Documents/Interfaz/prueba/DATA' #Cambia a la ruta
donde hayas almacenado Data
imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)

#face_recognizer = cv2.face.EigenFaceRecognizer_create()
#face_recognizer = cv2.face.FisherFaceRecognizer_create()
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Leyendo el modelo
#face_recognizer.read('modeloEigenFace.xml')
#face_recognizer.read('modeloFisherFace.xml')
face_recognizer.read('modeloLBPHFace.xml')

cap = cv2.VideoCapture(0)
#cap = cv2.VideoCapture('path_video')

faceClassif =
cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_d
efault.xml')

while True:
    ret,frame = cap.read()
    if ret == False: break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = gray.copy()

    faces = faceClassif.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro,(150,150),interpolation=
cv2.INTER_CUBIC)
        result = face_recognizer.predict(rostro)

        cv2.putText(frame,'{}'.format(result),(x,y-
5),1,1.3,(255,255,0),1,cv2.LINE_AA)

        # EigenFaces
        if result[1] < 5700:

cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-
25),2,1.1,(0,255,0),1,cv2.LINE_AA)
            cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        else:

```

```

        cv2.putText(frame, 'Desconocido', (x, y-
20), 2, 0.8, (0, 0, 255), 1, cv2.LINE_AA)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)

        # FisherFace
        if result[1] < 500:
cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x, y-
25), 2, 1.1, (0, 255, 0), 1, cv2.LINE_AA)
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
        else:
cv2.putText(frame, 'Desconocido', (x, y-
20), 2, 0.8, (0, 0, 255), 1, cv2.LINE_AA)
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
            , , ,
        # LBPHFace
        if result[1] <= 70:
cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x, y-
25), 2, 1.1, (0, 255, 0), 1, cv2.LINE_AA)
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
        else:
cv2.putText(frame, 'Desconocido', (x, y-
20), 2, 0.8, (0, 0, 255), 1, cv2.LINE_AA)
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)

        cv2.imshow('frame', frame)
        k = cv2.waitKey(1)
        if k == 27:
            break

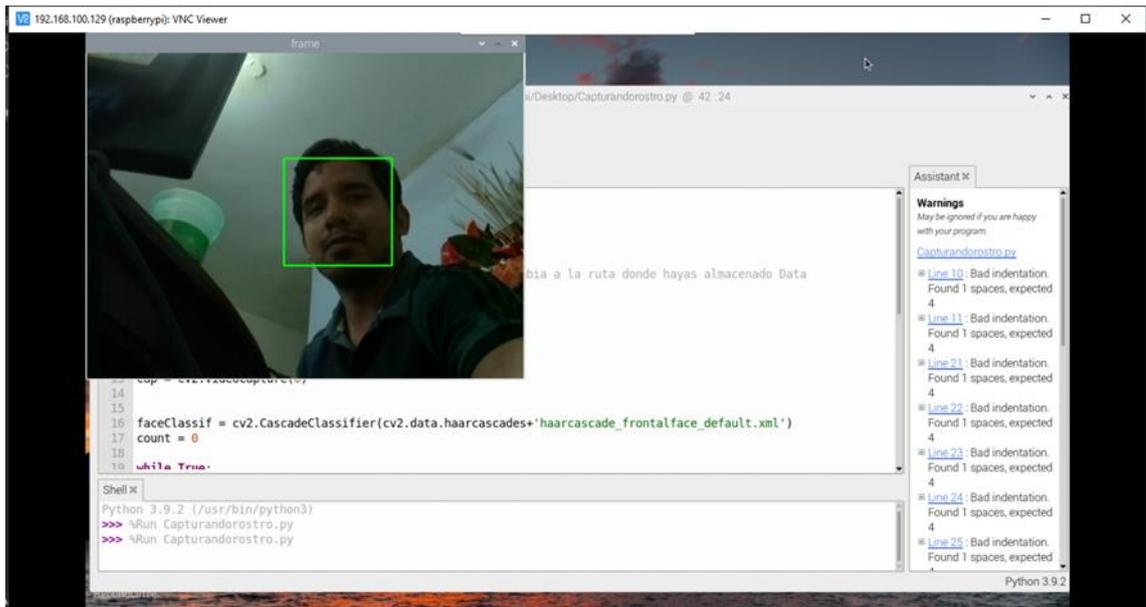
cap.release()
cv2.destroyAllWindows()

```

RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Raspberry Pi 4
- Módulo de cámara Raspberry
- VNC Viewer
- Visual Studio Code

ANEXOS



ANEXOS:

Material de apoyo “eManual Robotis – Robot Bioloid Gp” de Robotis

PRÁCTICA # 4

TEMA:

“CONTROL DE MOVIMIENTOS DEL ROBOT BIOLOID GP USANDO RASPBERRY PI MEDIANTE LENGUAJE DE PROGRAMACIÓN PYTHON”

OBJETIVO GENERAL:

- Control de Robot Bioloid GP usando Rapsberry PI

OBJETIVOS ESPECIFICOS:

- Realizar programación para la comunicación entre el controlador CM-530 y la Raspberry pi.
- Comprobar la comunicación entre el controlador CM-530 y raspberry pi, mediante lenguaje Python.

MARCO TEÓRICO:

El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado, nuestra programación se implementa en la Raspberry pi empleando el lenguaje de Python, y mediante el código realizado se busca la captura, entrenamiento y reconocimiento de rostros.

Con esto buscamos el control del Robot Bioloid GP mediante la raspberry pi, con la finalidad de que el robot pueda realizar acciones al momento de reconocer un rostro almacenado en la base de datos.

Robot Humanoide:

La robótica humanoide es una rama de la robótica que trata de que los robots imiten tanto en apariencia como en movimiento, ya sea locomoción, movimientos gestuales o movimientos en estático a los seres humanos. Son diseñados reproduciendo la forma de los brazos, de la cabeza y de los pies que los seres humanos. Pueden ser bípedos o pueden desplazarse por medio de una plataforma con ruedas en la parte inferior. Los robots más sofisticados buscan el hiperrealismo e intentan simular las expresiones de la cara, nuestros gestos y la piel. El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines. Por otro lado, el intento de simular el cuerpo humano conduce a una mejor comprensión de él. Además de la investigación, se están desarrollando robots humanoides para realizar tareas humanas como asistencia personal, a través de las cuales deberían poder ayudar a los enfermos y ancianos, y trabajos sucios o peligrosos. Los humanoides también son adecuados para algunas vocaciones basadas en procedimientos, como los administradores de mostradores de recepción y los trabajadores de la línea de fabricación de automóviles. En esencia, dado que pueden usar herramientas y operar equipos diseñados para la forma humana, los humanoides podrían, en teoría, realizar cualquier tarea que pueda hacer un ser humano, siempre que cuenten con el software adecuado. Sin embargo, la complejidad de hacerlo es inmensa

Algunos ejemplos de robots humanoides que encontramos en la actualidad son:

- Robot Sophia
- Robot Pepper
- Robot NAO
- Robot Atlas

- Robot ASIMO

Servomotores Dynamixel AX-12A:

El servomotor Dynamixel Ax-12A es un nuevo actuador robótico del sistema ROBOTIS Bioloid sustituye al antiguo Dynamixel Ax12+ con el mismo rendimiento, pero un diseño externo más avanzado. Cada actuador tiene un ID único lo cual le va a permitir ser identificado por el controlador, se puede usar un rango de 0 a 252. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 se puede operar con un solo conector adjunto.

RoboPlus:

Es un software libre que sirve para trabajar con los diferentes modelos de robots tales como: Ollo, Bioloid y Darwin Op, pertenecientes a la marca Robotis. Además, es compatible con el controlador CM-530, los servomotores Dynamixel y los sensores de Robotis los cuales se usarán en nuestro proyecto. [10] Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

Sensor Gyro GS-12:

El sensor Gyro GS-12 nos permitirá mantener el equilibrio de nuestro prototipo, ya que mide la velocidad angular y a su vez se puede utilizar para otras aplicaciones de movimiento. También, calcula que lado está inclinado el robot o está recibiendo más potencia.

Controlador CM-530:

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido.

Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

Raspberry Pi 4

Raspberry Pi es un miniordenador que se utiliza desde centro multimedia hasta de cerebro en robots.

Software VNC Server-Viewer

VNC por sus siglas en inglés Virtual Network Computing o en español, Computación Virtual en Red, es un software de código libre de tipo cliente servidor que permite ver la pantalla del ordenador servidor y controlarlo en uno o varios ordenadores clientes sin importar que sistema operativo pueda ejecutar el cliente o el servidor, podemos ver la pantalla y controlar el equipo del que ejecuta el servidor desde el cliente.

Módulo de cámara Raspberry Pi

Módulo con Cámara diseñado especialmente para Raspberry Pi. Se conecta en uno de los dos pequeños sockets que tiene la tarjeta Raspberry en su cara superior. Esta conexión, usa la interfaz dedicada CSI, la cual es óptima para conectar una cámara gracias a su alta capacidad de transmisión de datos.

Receptor BT-410 Dongle

BT-410 Dongle convierte la interfaz USB 2.0 y la interfaz de comunicación estándar Bluetooth 4.0 Low Energy (Master).

Si el Dongle BT-410 está conectado a la PC, se puede emparejar con un módulo esclavo BT-410 para intercambiar datos.

Lenguaje Python

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

MARCO PROCEDIMENTAL

Conectar la Raspberry Pi a una red cableada o inalámbrica, para poder conectarnos mediante VNC VIEWER y realizar nuestra programación en Python de forma remota.

Luego de conectarnos remotamente a nuestra Raspberry se empezará a realizar el código en el editor Visual Studio Code, dicho código nos permitirá establecer la comunicación con el controlador CM-530 y el raspberry pi el cual será encargado de dictar las ordenes de movimiento al robot mediante el lenguaje de programación Python.

PROGRAMACION PARA LA COMUNICACIÓN ENTRE

CONTROLADORES:

```
import time
import sys
import cv2
```

```

import serial

#Variables
#PySerial section
highbyte = 0
lowbyte = 0
TxD_packet = bytearray(6)
TxD_packet[0] = 0xff
TxD_packet[1] = 0x55
TxD_packet[2] = lowbyte
TxD_packet[3] = ~lowbyte & 0xff #asegura el contenido de ~lowbyte
TxD_packet[4] = highbyte
TxD_packet[5] = ~highbyte & 0xff #asegura el contenido de ~highbyte
ser = serial.Serial("/dev/ttyUSB0",57600, timeout=0.1,
write_timeout=0.1)
ser.reset_output_buffer() #Limpia el buffer usado para el envio de
dato al CM-530

#Seccion de interfaz de teclado
#ESC=27
#FLECHA_DERECHA= 65361
#FLECHA_IZQUIERDA= 65361
#No_1 = 49

#Funcion
def send_data():
    global TxData, TxD_packet, ser
    lowbyte = TxData & 0xff
    TxD_packet[2] = lowbyte & 0xff
    TxD_packet[3] = ~lowbyte & 0xff
    highbyte = (TxData >> 8) & 0xff
    TxD_packet[4] = highbyte & 0xff
    TxD_packet[5] = ~highbyte & 0xff
    if(ser.write (TxD_packet)!= 6):
        print("Error de transmision")
    time.sleep(0.100)
    #asegura el contenido de ~highbyte
    ser= serial.Serial("/dev/ttyUSB0",57600, timeout=0.1,
write_timeout=0.1)

while True:

    TxData = 0
    #if(key == FLECHA_DERECHA):
    TxData+= 216 #Valor derecha de TxData

    send_data()
    print("SALUDO",TxData)
    #TxData = 0
    #time.sleep(2.0)

    break

```

RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL

CONSUMIBLE):

- Robot Bioloid GP
- Controlador CM-530
- Programa Robot Plus
- Raspberry Pi

- Visual Studio Code

ANEXOS:

Material de apoyo “eManual Robotis – Robot Bioloid GP” de Robotis

PRÁCTICA # 5

TEMA:

“Crear movimientos del Robot Bioloid GP en software RoboPlus”

OBJETIVO GENERAL:

- Crear movimientos para el Robot Bioloid GP mediante software Roboplus

OBJETIVOS ESPECIFICOS:

- Utilizar software Roboplus motion para crear los movimientos en el robot.
- Utilizar software Roboplus task para cargar movimientos en el robot.

MARCO TEÓRICO

El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado, nuestra programación se implementa en la Raspberry pi empleando el lenguaje de Python, y mediante el código realizado se busca la captura, entrenamiento y reconocimiento de rostros.

Con esto buscamos el control del Robot Bioloid GP mediante la raspberry pi, con la finalidad de que el robot pueda realizar acciones al momento de reconocer un rostro almacenado en la base de datos.

Robot Humanoide:

La robótica humanoide es una rama de la robótica que trata de que los robots imiten tanto en apariencia como en movimiento, ya sea locomoción, movimientos gestuales o movimientos en estático a los seres humanos. Son diseñados reproduciendo la forma de los brazos, de la cabeza y de los pies que los seres humanos. Pueden ser bípedos o pueden desplazarse por medio de una plataforma con ruedas en la parte inferior. Los robots más sofisticados buscan el hiperrealismo e intentan simular las expresiones de la cara, nuestros gestos y la piel. El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines. Por otro lado, el intento de simular el cuerpo humano conduce a una mejor comprensión de él. Además de la investigación, se están desarrollando robots humanoides para realizar tareas humanas como asistencia personal, a través de las cuales deberían poder ayudar a los enfermos y ancianos, y trabajos sucios o peligrosos. Los humanoides también son adecuados para algunas vocaciones basadas en procedimientos, como los administradores de mostradores de recepción y los trabajadores de la línea de fabricación de automóviles. En esencia, dado que pueden usar herramientas y operar equipos diseñados para la forma humana, los humanoides podrían, en teoría, realizar cualquier tarea que pueda hacer un ser humano, siempre que cuenten con el software adecuado. Sin embargo, la complejidad de hacerlo es inmensa

Algunos ejemplos de robots humanoides que encontramos en la actualidad son:

- Robot Sophia
- Robot Pepper
- Robot NAO
- Robot Atlas
- Robot ASIMO

Servomotores Dynamixel AX-12A:

El servomotor Dynamixel Ax-12A es un nuevo actuador robótico del sistema ROBOTIS Bioloid sustituye al antiguo Dynamixel Ax12+ con el mismo rendimiento, pero un diseño externo más avanzado. Cada actuador tiene un ID único lo cual le va a permitir ser identificado por el controlador, se puede usar un rango de 0 a 252. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 se puede operar con un solo conector adjunto

RoboPlus:

Es un software libre que sirve para trabajar con los diferentes modelos de robots tales como: Ollio, Bioloid y Darwin Op, pertenecientes a la marca Robotis. Además, es compatible con el controlador CM-530, los servomotores Dynamixel y los sensores de Robotis los cuales se usarán en nuestro proyecto. [10] Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

Sensor Gyro GS-12:

El sensor Gyro GS-12 nos permitirá mantener el equilibrio de nuestro prototipo, ya que mide la velocidad angular y a su vez se puede utilizar para otras aplicaciones de movimiento. También, calcula que lado está inclinado el robot o está recibiendo más potencia.

CONTROLADOR CM-530:

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido.

Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

Raspberry Pi 4:

Raspberry Pi es un miniordenador que se utiliza desde centro multimedia hasta de cerebro en robots.

Software VNC Server-Viewer:

VNC por sus siglas en inglés Virtual Network Computing o en español, Computación Virtual en Red, es un software de código libre de tipo cliente servidor que permite ver la pantalla del ordenador servidor y controlarlo en uno o varios ordenadores clientes sin importar que sistema operativo pueda ejecutar el cliente o el servidor, podemos ver la pantalla y controlar el equipo del que ejecuta el servidor desde el cliente.

Módulo de cámara Raspberry Pi:

Módulo con Cámara diseñado especialmente para Raspberry Pi. Se conecta en uno de los dos pequeños sockets que tiene la tarjeta Raspberry en su cara superior. Esta conexión, usa la interfaz dedicada CSI, la cual es óptima para conectar una cámara gracias a su alta capacidad de transmisión de datos.

Receptor BT-410 Dongle:

BT-410 Dongle convierte la interfaz USB 2.0 y la interfaz de comunicación estándar Bluetooth 4.0 Low Energy (Master).

Si el Dongle BT-410 está conectado a la PC, se puede emparejar con un módulo esclavo BT-410 para intercambiar datos.

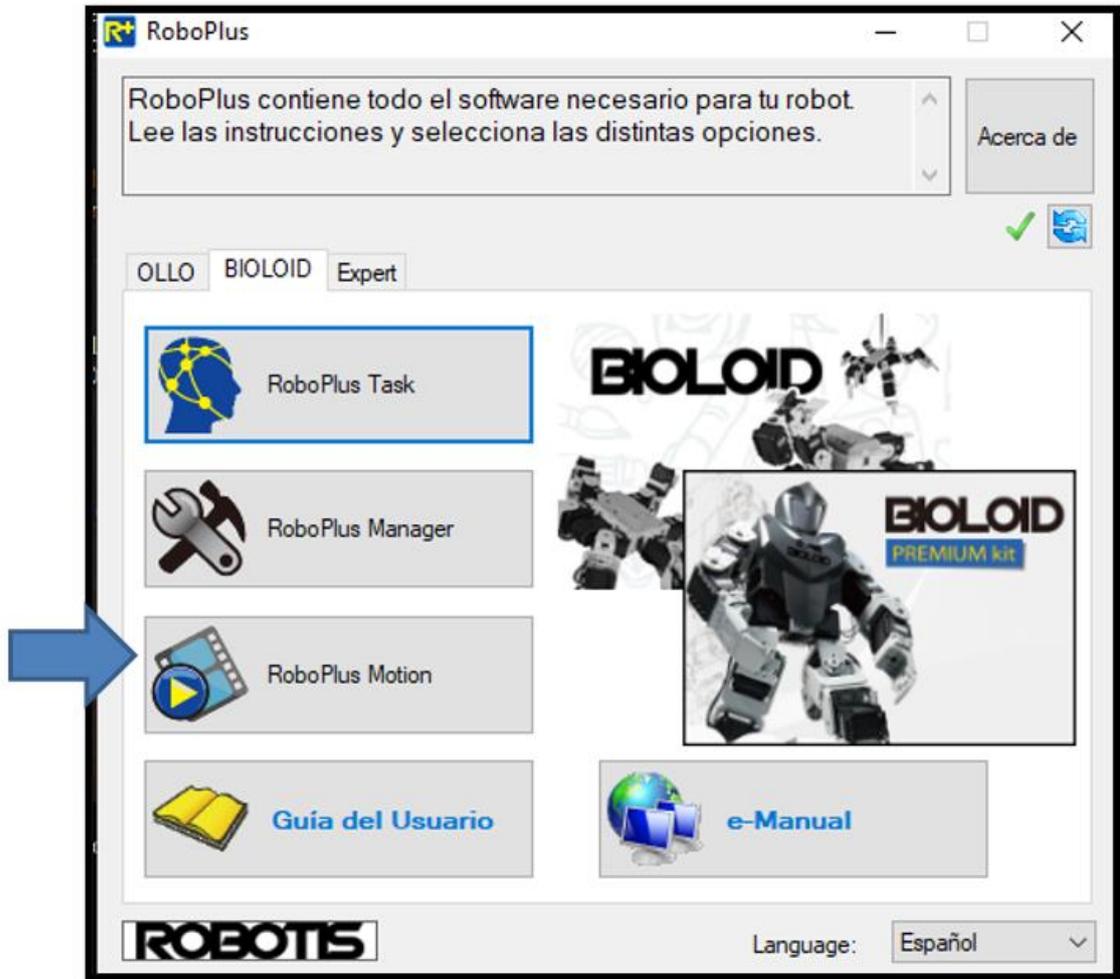
Lenguaje Python:

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

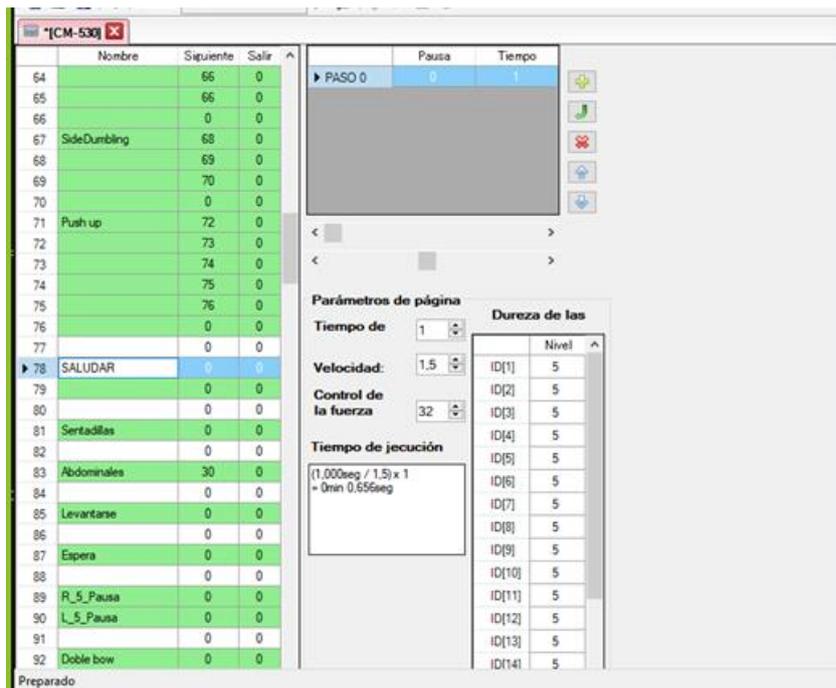
Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

MARCO PROCEDIMENTAL:

Para poder crear un movimiento para el robot, debemos abrir la aplicación “robotplus motion”, debemos conectar el robot mediante cable USB, la laptop con el controlador CM-530.



Al conectar el controlador CM-530, se cargan todos los movimientos predefinidos.



En una celda en blanco debemos poner el nombre del movimiento que vamos a crear.

Creamos el movimiento que se necesita, en este caso es un movimiento de saludo.

RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Robot Bioloid GP
- Controlador CM-530
- Programa Robot Plus

ANEXOS

Material de apoyo “eManual Robotis – Robot Bioloid Gp” de Robotis

PRÁCTICA # 6

TEMA:

“Realizar movimiento del robot bioloid gp mediante el reconocimiento facial usando Raspberry Pi (Cámara Raspberry Pi)”

OBJETIVO GENERAL:

- Efectuar el movimiento de un saludo en el robot Bioloid Gp al reconocer un rostro guardado en una base de datos.

OBJETIVOS ESPECIFICOS:

- Crear Interfaz gráfica para facilitar la comprensión del Reconocimiento Facial.
- Realizar la captura de rostros mediante la cámara Raspberry PI
- Entrenar el Algoritmo LBPH con los rostros capturados.
- Reconocimiento del rostro y movimiento de saludo del Robot.

MARCO TEÓRICO

El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado, nuestra programación se implementa en la Raspberry pi empleando el lenguaje de Python, y mediante el código realizado se busca la captura, entrenamiento y reconocimiento de rostros.

Con esto buscamos el control del Robot Bioloid GP mediante la raspberry pi, con la finalidad de que el robot pueda realizar acciones al momento de reconocer un rostro almacenado en la base de datos.

Robot Humanoide:

La robótica humanoide es una rama de la robótica que trata de que los robots imiten tanto en apariencia como en movimiento, ya sea locomoción, movimientos gestuales o movimientos en estático a los seres humanos. Son diseñados reproduciendo la forma de los brazos, de la cabeza y de los pies que los seres humanos. Pueden ser bípedos o pueden desplazarse por medio de una plataforma con ruedas en la parte inferior. Los robots más sofisticados buscan el hiperrealismo e intentan simular las expresiones de la cara, nuestros gestos y la piel. El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines. Por otro lado, el intento de simular el cuerpo humano conduce a una mejor comprensión de él. Además de la investigación, se están desarrollando robots humanoides para realizar tareas humanas como asistencia personal, a través de las cuales deberían poder ayudar a los enfermos y ancianos, y trabajos sucios o peligrosos. Los humanoides también son adecuados para algunas vocaciones basadas en procedimientos, como los administradores de mostradores de recepción y los trabajadores de la línea de fabricación de automóviles. En esencia, dado que pueden usar herramientas y operar equipos diseñados para la forma humana, los humanoides podrían, en teoría, realizar cualquier tarea que pueda hacer un ser humano, siempre que cuenten con el software adecuado. Sin embargo, la complejidad de hacerlo es inmensa

Algunos ejemplos de robots humanoides que encontramos en la actualidad son:

- Robot Sophia

- Robot Pepper
- Robot NAO
- Robot Atlas
- Robot ASIMO

Servomotores Dynamixel AX-12A:

El servomotor Dynamixel Ax-12A es un nuevo actuador robótico del sistema ROBOTIS Bioloid sustituye al antiguo Dynamixel Ax12+ con el mismo rendimiento, pero un diseño externo más avanzado. Cada actuador tiene un ID único lo cual le va a permitir ser identificado por el controlador, se puede usar un rango de 0 a 252. Los dos conectores del Dynamixel están conectados pin a pin, por lo que el AX-12 se puede operar con un solo conector adjunto

RoboPlus:

Es un software libre que sirve para trabajar con los diferentes modelos de robots tales como: Ollio, Bioloid y Darwin Op, pertenecientes a la marca Robotis. Además, es compatible con el controlador CM-530, los servomotores Dynamixel y los sensores de Robotis los cuales se usarán en nuestro proyecto. [10] Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

Sensor Gyro GS-12:

El sensor Gyro GS-12 nos permitirá mantener el equilibrio de nuestro prototipo, ya que mide la velocidad angular y a su vez se puede utilizar para otras aplicaciones de movimiento. También, calcula que lado está inclinado el robot o está recibiendo más potencia.

Controlador CM-530:

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido.

Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

Receptor BT-410:

El receptor BT-410 tiene una comunicación Bluetooth 4.0 que será compatible con nuestro control 'Rc-100B' que nos ayudará para que nuestro prototipo pueda ser operado de una manera remota. Nos permite la comunicación en serie (UART) a través de Bluetooth y es montable a equipos de robótica, controlable a través de Smartphones, tablet PC, Laptop, etc.

Control RC-100B:

El control remoto Rc – 100B nos ayudará a comunicarnos con el robot de manera remota ya que admite su comunicación con los módulos Bluetooth, ZIGbee o infrarrojo en caso de que no se adapte modulo Bluetooth o ZIGbee su comunicación predeterminada será por infrarrojo.

CONTROLADOR CM-530:

El controlador CM-530 cuenta con un procesador Arm Cortex M3 de 32 bits y combina una CPU, una placa de comunicación TTL (bus Dynamixel de 3 pines), LED de estado, botones de entrada, puertos de E/S GP, zumbador y sensor de sonido.

Es compatible con las series AX y MX, y la comunicación se maneja mediante la conexión USB nativa a la PC e IR, Zigbee o Bluetooth para el control inalámbrico.

Raspberry Pi 4:

Raspberry Pi es un miniordenador que se utiliza desde centro multimedia hasta de cerebro en robots.

Software VNC Server-Viewer:

VNC por sus siglas en inglés Virtual Network Computing o en español, Computación Virtual en Red, es un software de código libre de tipo cliente servidor que permite ver la pantalla del ordenador servidor y controlarlo en uno o varios ordenadores clientes sin importar que sistema operativo pueda ejecutar el cliente o el servidor, podemos ver la pantalla y controlar el equipo del que ejecuta el servidor desde el cliente.

Módulo de cámara Raspberry Pi:

Módulo con Cámara diseñado especialmente para Raspberry Pi. Se conecta en uno de los dos pequeños sockets que tiene la tarjeta Raspberry en su cara superior. Esta conexión, usa la interfaz dedicada CSI, la cual es óptima para conectar una cámara gracias a su alta capacidad de transmisión de datos.

Receptor BT-410 Dongle

BT-410 Dongle convierte la interfaz USB 2.0 y la interfaz de comunicación estándar Bluetooth 4.0 Low Energy (Master).

Si el Dongle BT-410 está conectado a la PC, se puede emparejar con un módulo esclavo BT-410 para intercambiar datos.

Lenguaje Python:

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

MARCO PROCEDIMENTAL:

Se procede a crear la interfaz gráfica en Python, para facilitar la comprensión de la práctica, en esta interfaz gráfica se encuentran las programaciones desde la Captura de rostro, entrenamiento del algoritmo, reconocimiento de rostro y movimiento de saludo del robot Bioloid GP.

RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Robot Bioloid GP
- Controlador CM-530
- Programa Robot Plus
- Sensor DMS-80

- Módulos BT 4.10 (Bluetooth).
- RC-100 (Control remoto)

BIBLIOGRAFÍA UTILIZADA:

Material de apoyo “eManual Robotis – Robot Bioloid Gp ” de Robotis