



PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO



Oswaldo Andrés Piña Bustamante

Brazo y mano robótica para manipulación de objetos

Informe Proyecto de Título de Ingeniero Civil Electrónico



**Escuela de Ingeniería Eléctrica
Facultad de Ingeniería**

Valparaíso, 30 de mayo de 2019



Brazo y mano robótica para manipulación de objetos

Oswaldo Andrés Piña Bustamante

Informe Final para optar al título de Ingeniero Civil Electrónico,
aprobada por la comisión de la
Escuela de Ingeniería Eléctrica de la
Facultad de Ingeniería de la
Pontificia Universidad Católica de Valparaíso
conformada por

Sr. Gabriel Enrique Hermosilla Vigneau
Profesor Guía

Sr. Francisco Guillermo Pizarro Torres
Segundo Revisor

Sr. Sebastián Fingerhuth Massmann
Secretario Académico

Valparaíso, 30 de mayo de 2019

*Para mis padres Maritza y Pedro.
Para mis hermanas Maritza y Paola.
Para mi padrino Nelson*

*Para mis amigos y hermanos de vida,
Esteban Hermosilla, Felipe Moya y Rodrigo Jilbert.*

*Para dos personas que llevo siempre en mi corazón,
mi madrina Loretto (QEPD) y mi abuelo Kurt (QEPD).*

Agradecimientos

En primer lugar, deseo expresar mi agradecimiento a los profesores Gabriel Hermosilla y Francisco Pizarro, por brindarme su apoyo, confianza y consejos a lo largo del desarrollo y elaboración de esta tesis, animándome siempre a hacer lo mejor.

Quiero agradecer profundamente a mis padres, quienes siempre se han esforzado por brindarme las herramientas y los valores que han permitido mi desarrollo, tanto a nivel personal como profesional. Éste logro también es de ellos. Agradecer igualmente a mi hermana, que siempre ha estado ahí para aconsejarme y levantarme cuando he sentido que las cosas se han puesto difíciles.

Agradezco también a Miguel Rojas, Enrique Torres y Sebastián Vergara, por su amistad y apoyo a lo largo de los años de estudios. Y también a todos los que han sido parte de este proceso como, Matías Cuevas, Daniela Pérez, Francisco Torres, Lisa Soto, Cindy Escalona, Daniel Maldonado, Flavio Torti, Rebeca Parra, Ohbrayan Ávila, Franco Gnecco, entre otros.

Me gustaría agradecer, además, a Daniela Birchmeier, por entenderme, acompañarme, apoyarme en los momentos más complejos estando lejos de casa, y por ser una persona especial en mi vida. Igualmente agradecer a mi amigo y hermano Esteban Hermosilla, al cual conozco desde el colegio, y es quien me aconsejó y acompañó durante los primeros años de universidad, incluso hasta el día de hoy.

Por último, agradecer a todas las personas que, de una u otra manera, han sido parte de este proceso, soportando y comprendiendo con estoica paciencia la dedicación que requiere la realización de un proyecto como este.

Resumen

El presente documento tiene como finalidad guiar al lector a través del trabajo elaborado en el proyecto “Diseño y construcción de un brazo robótico para manipulación de objetos”. El motivo de su realización es proporcionar capacidad de manipulación a un robot de servicio creado por alumnos de la Facultad de Ingeniería Eléctrica de la Pontificia Universidad Católica de Valparaíso.

Para dar solución al problema planteado, se realizó una revisión de antecedentes para identificar y comprender los principios de funcionamiento, diseño y aplicación de manipuladores en la robótica de servicio. También se presentó un marco teórico donde se expusieron los conceptos a considerar para el desarrollo del proyecto.

El diseño mecánico del brazo y mano robótica se realizó aplicando herramientas CAD. El diseño de los sensores de presión se efectuó tomando como base las recomendaciones del documento [1]. En la construcción del brazo robótico se utilizaron herramientas de impresión 3D para la materialización de las piezas que conforman las articulaciones y tubos de aluminio para la unión entre ellas. Además, se realizó la instalación del cableado utilizando principalmente cables de par trenzado (UTP).

Para validar la construcción y operación del brazo robótico se generaron pruebas de manipulación, estas requirieron la utilización de un sistema de visión por computador a través de *Deep Learning* que permitiera la detección de objetos de interés. Para determinar el funcionamiento de los sensores de presión, se realizaron pruebas de sujeción de objetos y se capturaron los datos entregados por los sensores para determinar su comportamiento.

Palabras claves: diseño, sensores, brazo, mano, efector, Python, Raspberry, 3D, presión, robótica, construcción, ROS, nodos, tópicos, Deep Learning, visión por computador.

Abstract

The purpose of this document is to guide the reader through the work developed in the project "design and construction of a robotic arm for manipulation of objects". The reason for its was built to provide handling capacity to a service robot created by students of the Faculty of Electrical Engineering of the Pontifical Catholic University of Valparaíso.

To solve the problem, a background check was conducted to identify and understand the principles of operation, design and application of manipulators in service robotics. A theoretical framework was also presented where the concepts to be considered for the development of the project were presented.

The mechanical design of the arm and robotic hand was made applying CAD tools. The design of the pressure sensors was made based on the recommendations of the document [1]. In the construction of the robotic arm 3D printing tools were used to materialize the parts that make up the joints, and aluminum tubes for the union between them. In addition, the wiring was installed using mainly twisted pair (UTP) cables.

To validate the construction and operation of the robotic arm, manipulation tests were generated, these required the use of a computer vision system through Deep Learning that allowed the detection of objects of interest. To determine the operation of the pressure sensors, object holding tests were performed and the data delivered by the sensors was captured to determine their behavior.

Keywords: design, sensors, arm, hand, effector, Python, Raspberry, 3D, pressure, robotics, construction, ROS, nodes, topics, Deep Learning, computer vision.

Índice general

Introducción.....	1
1 Antecedentes	4
1.1 Problemática	4
1.2 Estado del arte	5
1.2.1 Estado del arte de brazos robóticos	5
1.2.2 Estado del arte de sensores de presión.....	16
1.3 Solución propuesta	17
1.3.1 Objetivo general.....	17
1.3.2 Objetivos específicos.....	17
2 Marco teórico	18
2.1 Algoritmo	18
2.2 Sensor.....	18
2.3 Presión	21
2.4 Eslabón	21
2.5 Articulación	21
2.6 Cadena Cinemática.....	22
2.7 Grados de libertad (DoF).....	22
2.8 Robot de servicio.....	22
2.8.1 Robot de servicio doméstico/personal.....	23
2.8.2 Robot de servicio profesional	23
2.8.3 Robot móvil.....	23
2.9 Clasificación de los robots	23
2.9.1 Clasificación por área de aplicación	23
2.9.2 Clasificación por tipo de actuadores	24
2.9.3 Clasificación por número de ejes.....	25
2.9.4 Clasificación por morfología o configuración	25
2.10 Actuadores	25
2.10.1 Servomotor.....	25
2.10.2 Actuador lineal eléctrico	26
2.11 Electrónica.....	28

2.11.1 Picaro MEGA (Arduino compatible)	28
2.11.2 Encoder	29
2.12 Impresión 3D.....	30
2.13 Visión por computador	30
2.13.1 Imagen digital	30
2.13.2 Procesamiento digital de imágenes	31
2.13.3 Deep Learning	32
2.14 Localización y detección de objetos con Deep Learning	34
2.15 Software para adquisición de datos por conexión serial (PLX-DAQ).....	38
3 Desarrollo	39
3.1 Diseño mecánico de brazo y mano robótica	39
3.1.1 Gripper o actuador final.....	40
3.1.2 Muñeca.....	42
3.1.3 Codo.....	44
3.1.4 Hombro	45
3.1.5 Análisis de torque	47
3.2 Diseño sensor de presión textil de tipo resistivo	48
3.2.1 Materiales y herramientas	50
3.3 Construcción de brazo y mano robótica.....	51
3.3.1 Entorno de trabajo	51
3.3.2 Ensamble.....	52
3.4 Construcción de sensores de presión textil de tipo resistivo	55
3.5 Caracterización de sensor de presión textil de tipo resistivo	56
3.6 Ensamble de brazo terminado en plataforma móvil de robot de servicio.....	58
3.7 Diagrama de flujo para realización de pruebas	59
3.8 Creación de tópicos y nodos para visión por computador	60
4 Resultados y análisis	61
4.1.1 Pruebas de sensores de presión	61
4.1.2 Pruebas de visión por computador.....	68
4.1.3 Pruebas de manipulación brazo y mano robótica	69
Discusión y conclusiones.....	74
Bibliografía	76
A Morfología de los robots	84
B Circuito sensores y actuadores	85
B.1 Esquema Simplificado circuito sensores y actuadores	85
B.2 Conexiones de actuadores y sensores	86
C Componentes y diagrama de brazo robótico.....	87

D Presupuesto	92
E Scripts para prueba de sensores y manipulación	94
F Instalación de roserial arduino.....	100

Introducción

La robótica se ha desarrollado de forma acelerada, tal como lo indica la *International Federation of Robotics* (IFR) [2] [3]. Las tareas cotidianas que se llevan a cabo, por ejemplo, en los hogares, son muy difíciles de lograr con un sistema robótico porque el entorno es complejo, dinámico y estructurado para las necesidades humanas. Por ello, es de interés el desarrollo enfocado en la robótica de servicio. Los robots de servicio autónomos requieren una manipulación móvil versátil y habilidades de interacción humano-robot para realmente ser útiles [4].

Un robot de servicio debe ser capaz de sobrellevar la complejidad presente en el entorno donde se desenvuelve, entonces, la manipulación de objetos es una habilidad crucial para enfrentar ese desafío. El sistema de agarre de un robot de servicio debe ser adaptable, pues, en comparación con un robot industrial (diseñado para una tarea específica), éste debe manipular objetos con distintas formas. Por lo anterior, es necesario aplicar otro tipo de soluciones como, por ejemplo, pinzas envolventes o de múltiples dedos, y estructuras que admitan mayores grados de libertad, proporcionando múltiples puntos de contacto potenciales que se pueden posicionar de acuerdo al objeto, utilizando algoritmos de control y planificación de agarre.

Al fabricar objetos o utensilios, los creadores se inspiran en aspectos ergonómicos del cuerpo humano, por lo tanto, gran parte de los enfoques de diseño de brazos y manos robóticas se basan en este aspecto, con el objetivo de dar soluciones a las problemáticas de agarre y optimización de trayectorias [5]. El uso de software de diseño 3D ha sido relevante en el desarrollo de dichos diseños, y ha experimentado un importante crecimiento en los últimos años a causa de su incremento en potencia y facilidad de uso. El proceso de diseño de cualquier pieza o dispositivo debe ser planificado con el fin de determinar posibles inconvenientes en su implementación real. Para ello existen un gran abanico de herramientas de diseño y simulación, permitiendo materializar una idea abstracta en un objeto concreto.

Por otra parte, el diseño 3D, en este caso, puede ser complementado con *Robot operating system* (ROS). Ésta es una plataforma de software desarrollada el año 2007 específicamente para aplicaciones de robótica, permitiendo generar simulaciones y configuraciones de un modelo 3D y de un modelo real.

Debido a que es un software gratuito, se ha utilizado tanto para investigación como para fines comerciales. Proporciona herramientas y librerías para obtener, compilar, escribir y ejecutar

código a través de múltiples equipos. Éste se compone de un número de nodos independientes, cada nodo se comunica con el resto de los nodos utilizando el modelo *publicador/suscriptor*. Los nodos de ROS no tienen que estar necesariamente en el mismo sistema ni en la misma arquitectura, además, es mantenido, utilizado y desarrollado por un gran número de usuarios, que se traduce en una infraestructura flexible y adaptable según las necesidades. Además, ROS promueve la reutilización de código, es decir, los usuarios del sistema operativo pueden utilizar código de repositorios, realizar mejoras y compartir éstas con la comunidad [6].

La distribución *Kinetic*, lanzada el 23 de mayo del 2016, es apropiada para la aplicación del proyecto debido a que su desarrollo y soporte otorgan la estabilidad y compatibilidad requerida [7]. Ello asegura que la integración a los demás sistemas que conformarán el hardware y software del robot de servicio, que se desarrolla en el laboratorio de robótica de la Pontificia Universidad Católica de Valparaíso, se simplifique y limite a una versión. Si bien se encuentra disponible la distribución *Melodic*, liberada durante el año 2018, no todas sus librerías son compatibles con los componentes del robot de servicio, dado que posee una corta trayectoria y requiere mayor desarrollo.

Como se comentó anteriormente, ROS utiliza un modelo *publicador/suscriptor* por medio de un sistema de nodos, los cuales pueden ser utilizados para controlar distintos tipos de actuadores, como también para obtener información del medio a través de sensores. Si se estudia el caso del ser humano, la piel, las manos y en general todo el cuerpo, contienen un gran número de terminaciones nerviosas que entregan información a cada instante sobre el entorno en que las personas se encuentran. Existen sensores de temperaturas, sonido, luz, etc. Entre todos estos sensores, se encuentran también los sensores de presión, que adquieren gran relevancia si el objetivo es manipular ciertos elementos.

Según la R.A.E. un sensor es un “dispositivo que detecta una determinada acción externa, temperatura, presión, etc., y la transmite adecuadamente” [8] o los transductores, que la RAE define como “dispositivo que transforma el efecto de una causa física, como la presión, la temperatura, la dilatación, la humedad, etc., en otro tipo de señal, normalmente eléctrica” [9], es decir, es un dispositivo que es capaz de convertir un estímulo externo en información útil que luego se podrá procesar.

En el caso de la visión, un sensor óptico permite la orientación espacial, percepción de profundidad, como también la detección de objetos en el espacio y su ubicación. Esto es importante, dado que el brazo robótico debe ser capaz de ubicar objetos en un radio de acción y en conjunto con los sensores de presión, lograr su manipulación.

Para lograr la detección y reconocimiento de diferentes objetos, la visión artificial es trascendental. En [10] se utiliza un conjunto de procesos (preprocesamiento de la imagen, descripción y extracción de características), complementándose con una fase de reconocimiento por medio de una red neuronal *backpropagation*. Igualmente, en [11] se utiliza visión artificial mediante el uso la librería “OpenCV 3.0”, “Python” y una librería denominada FLANN, la cual permite realizar la búsqueda de posibles similitudes entre los objetos de un banco de imágenes mediante comparación de descriptores. Actualmente existe un campo de la Inteligencia Artificial

que realiza esta tarea de manera más eficiente. Este campo se denomina *Deep Learning* o Aprendizaje Profundo.

Para entender el *Deep Learning*, se debe comprender de manera general qué es la Inteligencia Artificial (IA). La IA es una disciplina que permite a las máquinas imitar el comportamiento humano. Dentro de ella existe un subgrupo denominado *Machine Learning*, que utiliza métodos estadísticos para permitir que las máquinas mejoren con la experiencia. Dentro del *Machine Learning* está contenido un subconjunto denominado *Deep Learning*, tipo particular de aprendizaje automático que logra un gran poder y flexibilidad al aprender a representar el mundo como una jerarquía de conceptos o abstracción anidadas. Este campo es un tipo de aprendizaje automático que se inspira en la funcionalidad de las células cerebrales, denominada “red neuronal artificial”. Tiene capacidad de aprendizaje en múltiples niveles de abstracción, lo que permite a un sistema aprender funciones complejas de mapeo sin depender de ningún algoritmo específico.

Por las consideraciones anteriores, el diseño del proyecto “Brazo y mano robótica para manipulación de objetos”, se debe realizar tomando en cuenta diferentes factores que afectan a la estructura real, como la fuerza de gravedad, pesos, torques y esfuerzos generados en los materiales a utilizar. Es conveniente, entonces, considerar la elección de materiales adecuados y un diseño electrónico que permita un buen desempeño.

En ese contexto, se propone realizar el diseño y construcción de un brazo robótico inspirado en desarrollos actuales, mediante herramientas de modelado 3D y sistemas de control mediante ROS (*Robot Operating System*), con el fin de conseguir un sistema funcional, de bajo coste, aplicable a la investigación y educación, además de ser replicable.

1 Antecedentes

1.1 Problemática

A nivel regional, el Laboratorio de Robótica de la Facultad de Ingeniería Eléctrica perteneciente a la Pontificia Universidad Católica de Valparaíso cuenta con algunos dispositivos: *Legó Mindstorms NTX 2*, que es un set programable fabricado por LEGO utilizado como herramienta de aprendizaje de programación. Cuatro robots móviles *Khepera IV* de la empresa *K-Team Corporation*, dotados de acelerómetro, giroscopio, cámara a color, Wifi, bluetooth y un núcleo Linux, a los cuales se les puede acoplar módulos para complementar sus capacidades. Estos dispositivos son utilizados principalmente para aplicaciones en el área de la educación e investigación. Además, en este laboratorio, los estudiantes pueden desarrollar sus proyectos personales como, por ejemplo, el diseño de robots para competencias (sumo y velocistas), como también la realización de proyectos de titulación.

Durante el último año se han desarrollado tres proyectos, los dos primeros denominados “Inteligencia computacional para una cabeza robótica” y “Navegación autónoma de una base robótica usando SLAM” con el fin de generar el diseño y construcción de un robot de servicio para el laboratorio de robótica. El tercero se titula “Robot de servicio EIE PUCV” y tiene como propósito integrar todos sus componentes.

La problemática que presenta este proyecto en relación a la integración mencionada en el párrafo anterior es que el robot de servicio requiere tener la capacidad de interactuar con el medio, no solo a través del habla o la visión, sino que también debe tener la capacidad de manipular objetos presentes en su entorno y, al mismo tiempo, entregar retroalimentación del objeto al resto del sistema de robot.

El brazo tendrá como principal función permitir al robot de servicio trasladar elementos que sean solicitados por los usuarios, como por ejemplo herramientas de uso cotidiano. También tiene como objetivo ser un instrumento de aprendizaje para los estudiantes relacionados con el mundo de la robótica, incentivándolos a la creación de nuevos proyectos o aplicando sus conocimientos para generar mejoras al prototipo expuesto en este informe.

Hasta la fecha no existe un proyecto documentado a nivel institucional que comprenda la creación de un brazo robótico, ni tampoco existe documentación que haga referencia a la

creación de un robot de servicio. Es por lo anterior, que la relación existente entre los proyectos antes mencionados, en conjunto con el proyecto que se presenta por medio de este informe, toma gran relevancia en el desarrollo de la robótica en la Facultad de Ingeniería de la Pontificia Universidad Católica de Valparaíso, e incluso, a nivel regional.

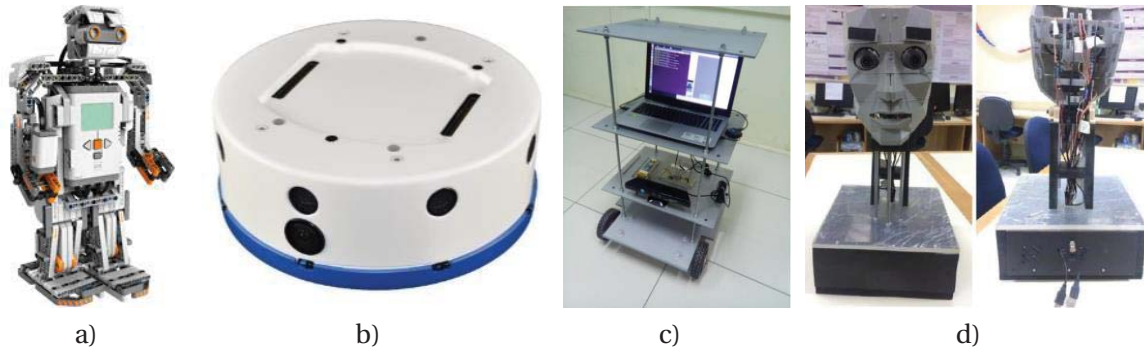


Figura 1-1: a) Lego Mindstorm NTX2 (Fuente: <http://lego.com>), b) Khepera IV (fuente: <https://www.generationrobots.com>), c) Base robótica móvil para navegación basada en SLAM [12] d) Cabeza robótica con inteligencia computacional [13]

1.2 Estado del arte

Dada la problemática que se ha presentado en el apartado 1.1, se ha realizado el estudio de cinco brazos antropomórficos enfocados a la robótica de servicio, además de una breve revisión de las últimas tecnologías aplicadas en el área. A continuación, se detallan sus principales características con el objetivo de determinar un modelo de brazo acorde a las necesidades del proyecto, iniciando con una breve contextualización histórica.

1.2.1 Estado del arte de brazos robóticos

Los primeros dispositivos que se conocen hoy en día como brazos robóticos, recibían la denominación de *manipuladores teleoperados* o *telemanipuladores*. Sus inicios se remontan al año 1948 cuando R.C. Goertz desarrolló el primer dispositivo de telemanipulación con el fin de maniobrar elementos radiactivos sin que el operario se expusiera a la radiación [14]. Luego, en 1954, Goertz aplicó tecnología utilizando electrónica y servocontrol para sustituir la transmisión mecánica, creando el primer *telemanipulador servocontrolado bilateral* [15], como se muestra en la Figura 1-2. Más adelante, en 1958 Ralph Mosher, ingeniero de *General Electric*, desarrolla un dispositivo denominado *Handy-man*, el cual consistía en dos brazos teleoperados mediante un exoesqueleto [16].

George C. Devol, en 1954 presenta un dispositivo denominado *Programmed Article Transfer* o dispositivo de transferencia de artículos programados [17], estableciendo las bases del robot industrial moderno. Luego, en 1956 Joseph F. Engelberger y Devol comienzan a trabajar en aplicaciones industriales fundando *Consolidated Controls Corporation* que posteriormente se transformó en la *Unimation Corp.* En 1959, la misma compañía instalaría un prototipo de *Unimate* en “General Motors” de Trenton, en Nueva Jersey, en una aplicación de fundición por inyección. El primer *Unimate* fue impulsado hidráulicamente. Su sistema de control se basaba en el control digital, una memoria de tambor magnético y componentes discretos de control de

estado sólido Un año después, *Unimate* es traspasado a *Condec Corporation* para, en 1961, entregar el primer robot de línea de producción del mundo a la fábrica de General Motors (Figura 1-3), el cual tenía como tarea remover y apilar partes metálicas que se encontraban a una alta temperatura desde una máquina de fundición [18].

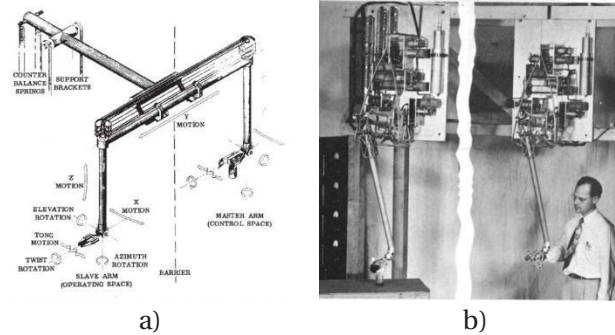


Figura 1-2: a) “Remote-Control Manipulator” primer telemanipulador de transmisión mecánica [14]
b) “Electronic Master Slave Manipulator” primer manipulador servocontrolado bilateral [15]



Figura 1-3: Robot Industrial Unimate 1961 [19]

FZI arm - DISEÑO NO COMERCIAL

En un entorno fuera del ámbito comercial, a mediados del año 1998, en la *international conference on advanced mechatronics (ICAM 98)* se presentó el artículo *Design and Control Architecture of an Anthropomorphic Robot Arm* [20], el cual describe el diseño de un brazo robótico para tareas de servicio. Su objetivo principal es el lograr diseñar un robot de servicio que pueda ayudar a las personas en tareas cotidianas. Para ello consideran que el componente principal del robot para el manejo de objetos es la construcción del brazo. Por lo anteriormente dicho, establecen realizar el diseño e implementación del brazo *FZI arm* (Figura 1-4). Sus principales características se muestran en la Tabla 1-1.

Tabla 1-1: Especificaciones técnicas de brazo robótico FZI arm [20]

	Unidad	FZI arm
Grados de libertad	-	7
Alcance máximo	cm	37
Carga máxima (temporal)	kg	5.7
Material de unión	-	Aluminio
Peso Total	kg	3.5
Tensión de alimentación	VCD	24
Motores	-	Motor DC / Poleas y cuerdas
ROS Compatible	-	No



Figura 1-4: Brazo robótico 7 grados de libertad modelo FZI arm [20]

Los autores del artículo plantean que el diseño del brazo y sus características permiten un movimiento similar al del brazo humano. La estructura utilizada corresponde a una pata del robot caminante *Lauron II*, con un sistema de cuerdas y poleas, transmisión que permite ejercer mayor fuerza.

BioRob - DISEÑO NO COMERCIAL

Durante el *International Symposium on Robotics (ISR)* en el 2010 se presentó el artículo *BioRob-Arm: A Quickly Deployable and Intrinsically Safe, Lightweight Robot Arm for Service Robotics Applications*. En el cual se le da prioridad al diseño liviano, con el fin de obtener flexibilidad, movilidad y propiedades de seguridad acorde a las aplicaciones de robótica de servicio, sus características se muestran en la Tabla 1-2.

Los autores describen ciertos requisitos claves para el establecimiento práctico y exitoso de un robot de servicio:

- Seguridad: la seguridad inherente a altas velocidades y el diseño humano fomentan la eficiencia y la aceptación.
- Flexibilidad: la movilidad y los tiempos de instalación e implementación cortos permiten cambiar rápidamente la ubicación del robot y reaccionar de manera flexible ante las condiciones de producción cambiantes y las necesidades actuales.

- Usabilidad: programación simple e intuitiva que puede ser realizada por personal no capacitado.
- Rendimiento: ejecución de tareas con velocidad y precisión comparable a un brazo humano.

Si bien los puntos anteriores están enfocados a la pequeña y mediana empresa con entornos no estructurados, es importante tomarlos en cuenta y enfocarlos al entorno de laboratorio de robótica de la casa de estudios. Considerando estructuralmente al robot *BioRob* (Figura 1-5), este se encuentra orientado a operaciones de recogida y colocación programadas mediante enseñanza guiada. Gracias a su sistema de poleas, la estructura permite cargas transitorias de hasta 0.5 [kg] y cargas máximas de 2 [kg] [21].

Tabla 1-2: Especificaciones técnicas de brazo robótico BioRob [21]

	Unidad	BioRob
Grados de libertad	-	4
Alcance máximo	cm	98
Carga continua	kg	0.5
Carga máxima (temporal)	kg	2
Material de unión	-	Aluminio
Peso Total	kg	3.75
Velocidad máxima lineal	cm/s	25
Tensión de alimentación	VCD	24
Potencia máxima	W	80
Motores	-	Motor DC / Poleas y cuerdas
ROS Compatible	-	No

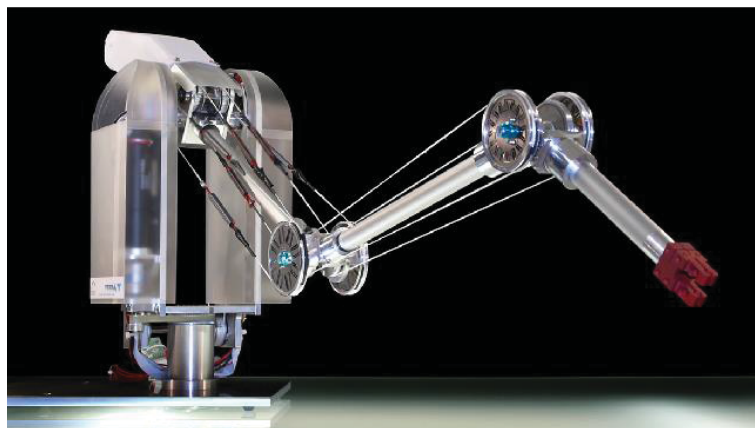


Figura 1-5: Brazo robótico 4 grados de libertad modelo BioRob [21]

LWA 4D - SCHUNK

El brazo robótico LWA 4D (Figura 1-6), es considerado un brazo ligero y compacto, es capaz de mover cargas de hasta 10 [kg]. Es utilizado como un componente estándar de los robots de servicio. Puede funcionar con baterías y consume 24 Volts continuos. Otras características se detallan en la Tabla 1-3.

Tabla 1-3: Especificaciones técnicas de brazo robótico LWA 4D [22]

	Unidad	LWA 4D
Grados de libertad	-	7
Alcance máximo	cm	1100
Carga máxima (temporal)	kg	10
Peso Total	kg	18
Tensión de alimentación	VCD	24
Motores	-	Servomotores brushless con tecnología Harmonic Drive
ROS Compatible	-	Si



Figura 1-6: Brazo robótico 6 grados de libertad modelo LWA 4D de Schunk
(fuente: <https://schunk.com>)

SRA 1000 HD - COMMONPLACE ROBOTICS GMBH

La empresa alemana *Commonplace Robotics GmbH*, dentro de sus productos cuenta con un brazo robótico de servicio (Figura 1-7), su alcance y la carga útil se orientan en el brazo humano y los componentes electrónicos integrados permiten un diseño compacto. Sus aplicaciones están orientadas a la robótica de servicio, manipulación móvil, aplicaciones educativas, entre otras [23].

Sus especificaciones se muestran en la Tabla 1-4.

Tabla 1-4: Especificaciones técnicas de brazo robótico SRA 1000 HD [23].

	Unidad	SRA 1000 HD
Grados de libertad	-	6
Alcance máximo	cm	80
Carga continua	kg	0.75
Carga máxima (temporal)	kg	1
Material de unión	-	Plástico reforzado
Peso Total	kg	6
Tensión de alimentación	VCD	24
Motores	-	Brushless DC con tecnología Harmonic Drive
ROS Compatible	-	Si



Figura 1-7: Brazo robótico 4 grados de libertad modelo SRA 1000 HD
(fuente: <http://shop.cpr-roboter.de>)

MICO² - KINOVA

En la revista internacional de Aplicaciones y Tecnologías de Robótica en el 2017 se presenta en el artículo *Kinova Modular Robot Arms for Service Robotics Applications* los sistemas robóticos de la compañía *Kinova*, dedicada a diseñar y fabricar plataformas y componentes robóticos. Entre los principales productos destaca un brazo robótico liviano denominado MICO² (Figura 1-8).

Las especificaciones técnicas del brazo MICO² se detallan en la Tabla 1-5.

Tabla 1-5: Especificaciones técnicas de brazo robótico MICO² (Los valores son para robots sin pinza) [24]

	Unidad	MICO ² 6 DOF
Grados de libertad	-	6
Alcance máximo	cm	70
Carga continua	kg	0.8
Carga máxima (temporal)	kg	1.3
Material de unión	-	Plástico reforzado
Peso Total	kg	5.2
Velocidad máxima lineal	cm/s	20
Tensión de alimentación	VCD	18 a 29
Potencia promedio	W	25
Energía en reposo	W	5
Potencia máxima	W	100
Motores	-	Servomotores brushless con tecnología Harmonic Drive
ROS Compatible	-	Si



Figura 1-8: Brazo robótico 6 grados de libertad modelo MICO² de Kinova
(fuente: <http://www.kinovarobotics.com>)

Además, en su estructura posee sensores en la base del brazo, tensión de alimentación, temperatura, acelerómetro sensores XYZ en cada actuador, *encoder* de posición óptica, posición absoluta, par, corriente y temperatura.

Un punto importante es la implementación de ROS (*Robot Operating System*), permitiendo el uso de bibliotecas abiertas y herramientas que contribuyen a la creación de aplicaciones y a la estandarización del desarrollo [24].

Para establecer el modelo de referencia, se debe realizar la comparación de las características que cada brazo otorga, para ello se genera un cuadro comparativo en el cual se observa que el brazo robótico MICO² cumple con los parámetros ideales, pero no es fiable por su elevado precio (Figura 1-9).






Nombre brazo robótico	FZI arm	BioRob	LWA 4D	SRA 1000 HD	MICO ²
Imagen					
6 grados de libertad (DoF)	✗	✗	✗	✓	✓
Alcance mínimo 65 [cm]	✗	✓	✓	✓	✓
Alcance máximo 80 [cm]	✗	✗	✗	✓	✓
Carga continua mínima 0.5 [kg]	✓	✓	✓	✓	✓
Peso total < 5.5 [kg]	✓	✓	✗	✗	✓
Motores Brushless - Armonic Drive	✗	✗	✓	✓	✓
ROS Compatible	✗	✓	✓	✓	✓
Precio	-	\$ 23.380.500.- [25]	-	\$ 3.721.628.- [23]	\$35.450.000.- [25]

Figura 1-9: Cuadro comparativo de características brazos robóticos de servicio

Efector final

Una sección importante de un brazo robótico manipulador es su efector final. A continuación, se repasarán los tipos de efectores utilizados actualmente.

Pinza multidedo

Una forma de maximizar la flexibilidad de la pinza paralela es utilizar múltiples dedos, una de las primeras manos de tres dedos desarrolladas se describe en el artículo *Design for a three-fingered hand* [26]. Algunos ejemplos de pinzas de dedos múltiples incluyen la mano *Robotiq* de 3 dedos [27], la mano *IH2 Azzura* [28] y la mano *Shadow* [29] (Figura 1-10).



Figura 1-10: Pinzas de agarre Kinova KG-3 [24] y 3-Finger Adaptive Robot Gripper [27]

Pinzas envolvente

El agarre de objetos de formas extrañas siempre ha sido un desafío, es por ello por lo que en un artículo llamado *Stable Precision Grasps by Underactuated Grippers* publicado el 2011, menciona la construcción de un par de manos no operadas. Este tipo de *gripper* es valioso debido a la adaptabilidad de la pinza para adaptarse a la forma del objeto que está sosteniendo [30]. Las pinzas envolventes nacen de la inspiración biológica como, por ejemplo, el diseño de un mecanismo de efector final basado en la boca de una serpiente [31], permitiendo la fijación continua de objetos con bordes lisos. Desarrollando aún más esta tecnología, es posible obtener pinzas que requieran menos control en comparación con las pinzas de dedos múltiples. Un ejemplo de ellas es la Pinza de agarre *Kinova KG-2* (Figura 1-11).

Pinzas maleables

Este tipo de pinzas están construidas con materiales que cambian su viscosidad, por ejemplo, con piel extra flexible y en su interior algún material granular. Su funcionamiento se basa principalmente en la adaptación del efector final a la forma del objeto, luego el interior se endurece para lograr el agarre y finalmente la consistencia del efector vuelve a un estado de mayor fluidez para soltar el objeto (Figura 1-12). Una versión comercializada de este diseño se llama *Versaball* [32]. En definitiva, las pinzas maleables son aptas para tomar objetos con diferentes formas; sin embargo, la destreza de estas pinzas sigue siendo un desafío de diseño en la actualidad.

Además, se encuentran disponibles otro tipo de dispositivos llamados Pinzas piezoeléctricas, pero su uso se limita a objetos muy pequeños.

A continuación, se muestra una tabla con distintos diseños de pinzas, sus ventajas e inconvenientes, aplicación específica y ejemplos de materiales utilizados (Tabla 1-6a y Tabla 1-6b).



Figura 1-11: Pinzas de agarre Kinova KG-2 [24]

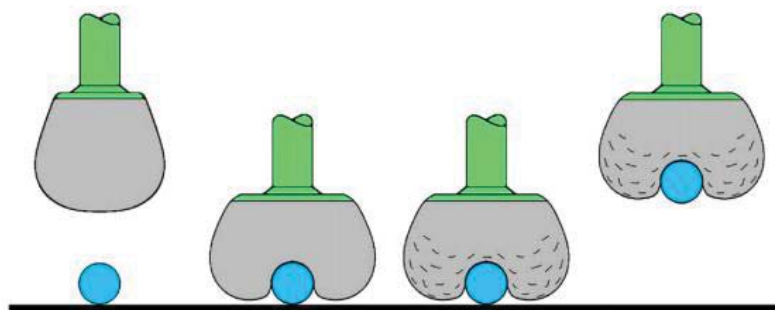


Figura 1-12: Funcionamiento de una pinza maleable [33]

Tabla 1-6a: Tabla de la capacidad de las pinzas para recoger diferentes tipos de objetos

Diseño	Ventajas	Inconvenientes	Aplicación específica	Ejemplo de material usado y actuadores
Pinzas piezoeléctricas	Simplicidad, facilidad de uso, agarre de objetos pequeños (hasta $50[\mu\text{m}]$)	Baja precisión de agarre	Agarre micro y nano	fluoruro de polivinilideno polimérico (PVDF)
Pinza de dedos múltiples	Agarre flexible para diferentes formas de objetos, agarre con fuerza de retroalimentación	Complejo de controlar	Agarre de objetos con distintas formas, control de fuerza	Materiales blandos, microactuadores flexibles (FMA), sistemas de accionamiento de lazo de alambre

Tabla 1-6b: Tabla de la capacidad de las pinzas para recoger diferentes tipos de objetos

Diseño	Ventajas	Inconvenientes	Aplicación específica	Ejemplo de material usado y actuadores
Pinzas envolventes	Adaptabilidad de la forma del objeto	Baja capacidad de control de la fuerza	Agarre de objetos con forma extraña o desconocidas.	Actuadores neumáticos, mecanismos accionados por cable
Pinzas maleables	Adaptable a diferentes formas, agarre confiable	Baja destreza de agarre	Agarrar objetos desconocidos y especialmente deformantes	Fluido MR, fluido ER, material granular

Otros avances

Hoy en día, los investigadores buscan nuevos actuadores que permitan aumentar la capacidad de carga de los brazos robóticos y brazos robóticos blandos, como por ejemplo el caso de estos últimos, que pueden ser accionados mediante cables de tensión variable, sistemas neumáticos o hidráulicos [34]. FESTO se han enfocado también en la investigación de sistemas neumáticos, como uno de sus proyectos entre los años 2006 y 2009 denominado *Airic's arm* [35] (Figura 1-13a), y durante el período 2010-2012 con el proyecto *ExoHand* [36]. Por su lado, *Shadow Robot Company* también ha realizado contribuciones al tema por medio de su proyecto *Shadow Dexterous Hand* [29].

Un desarrollo importante en termino de actuadores es la realizada por Schunk es el *Powerball* (Figura 1-13b). Según lo que el fabricante indica, cada unidad *Powerball* tiene dos ejes de servo posicionados ortogonalmente, que se alimentan a través de un engranaje de precisión *Harmonic Drive* y un servo-accionamiento DC sin escobillas. Cada eje tiene un codificador propio, una unidad de control y un freno de parada propio.

Siguiendo la línea mostrada por FESTO, el artículo [37] presentado el año 2017 ha desarrollado un brazo robótico bio-inspirado suave y liviano utilizando músculos artificiales y mangas inflables (Figura 1-13c), combinado con elementos de robótica rígido, otorgando a la estructura de precisión y bajo peso. Los músculos son accionados neumáticamente utilizando músculos artificiales tipo *McKibben*.

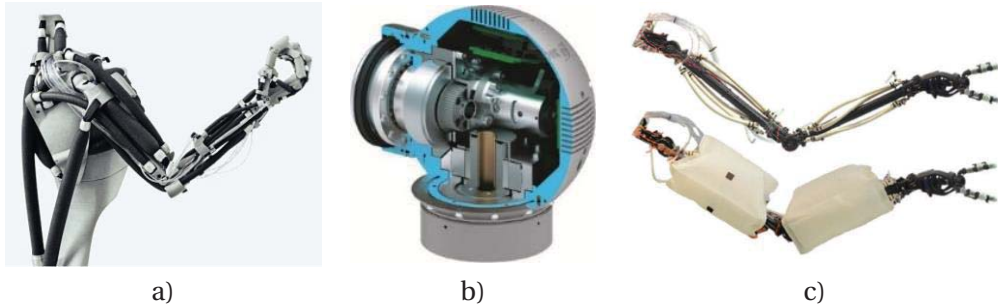


Figura 1-13: a) Airic's arm de FESTO [35] b) actuador Powerball de la compañía Schunk. (fuente: <https://schunk.com>), c) prototipo de brazo presentado en artículo [37]

1.2.2 Estado del arte de sensores de presión

Sensores de principio capacitivo

Existen un gran número de sensores con distintos principios de funcionamiento, por ejemplo, los transductores de tipo capacitivo. Entre las ventajas que poseen este tipo de transductores destacan una resolución infinita, baja dependencia a la temperatura, bajo consumo de energía, entre otros. Un ejemplo de su aplicación se describe en el artículo [38], donde la utilización de transductores capacitivos permite sensibilidad y rango dinámico lineal más altos y anchos que los dispositivos con electrodos paralelos. Sin embargo, pese a sus cualidades, posee una desventaja que lo hace poco recomendable para la aplicación que se desea implementar, requiere circuitos de lectura de datos muy especializado, el cual debe ser implementado en el proceso de fabricación, evitando así la presencia de capacitancias parásitas [39].

Sensores de principio piezoeléctrico

Existe otro tipo de transductores denominados piezoeléctricos, los cuales tienen la capacidad de generar carga eléctrica cuando una presión externa es aplicada sobre ellos [39]. Esta característica es utilizada en el artículo [40], utilizando nanocompuestos avanzados para el desarrollo de sensores de presión y deformación como, por ejemplo, piel artificial, dispositivos deformables implantados e instrumentos quirúrgicos flexibles multifuncionales. Sin embargo, el proceso de fabricación es complejo considerando las herramientas y los materiales disponibles.

Los sensores que utilizan materiales textiles piezo-resistivos y conductores son más blandos, delgados y livianos, lo cual permite su integración en tejidos usados regularmente en la fabricación de prendas [41]. Existen un variado espectro de aplicaciones para este tipo de sensores, como es, por ejemplo, el desarrollo de sensores para clasificación de postura corporal sentada [42], desarrollo de aplicación portátil para monitorización de señales cardiorrespiratorias en el sueño mediante tejido conductor [43]. El desarrollo de prendas inteligentes permite mejorar la postura o controlar otras afecciones, como se muestra en el artículo [41], donde son utilizados sensores de presión textil para la monitorización de la presión plantar anormal, ente otras.

Recientemente en el campo de la seguridad en el lugar de trabajo, se han desarrollado nuevos sensores para detección y monitorización del estrés corporal debido a movimientos repetitivos que eventualmente podrían causar traumas o accidentes de mayor complejidad [44], así como

también existen desarrollos orientados a la investigación y educación, generando experiencias de bajo costo y rápida implementación utilizando herramientas domésticas [1].

1.3 Solución propuesta

Es importante mencionar que, a nivel de facultad, no existen indicios del diseño y/o desarrollo de un brazo robótico para manipulación de objetos cuyo enfoque haya apuntado a aplicaciones de robótica de servicio. Como base, se han considerado los proyectos estudiados en el estado del arte de los brazos robóticos, con el fin de establecer un conjunto de características que mejor se adapten al entorno de trabajo del robot de servicio en el cual se ensamblará.

Inicialmente se estimó la posibilidad de realizar la adquisición de un brazo robótico comercial, sin embargo, debido a su alto valor se resolvió realizar el diseño y construcción del brazo robótico con elementos de bajo costo y fácil adquisición. Además, la retroalimentación para la mano robótica que manipulará distintos objetos se realizará a través de sensores de presión fabricados en base a material textil conductor y láminas antiestáticas convencionales de bajo costo.

Para cumplir con la propuesta de solución se han establecido los siguientes objetivos:

1.3.1 Objetivo general

- Diseñar y crear un brazo robótico que permita manipular objetos mediante sensores de presión ubicados en una mano robótica.

1.3.2 Objetivos específicos

- Estudiar el estado del arte de sensores de presión y brazos robóticos.
- Estudiar y utilizar herramientas de diseño e impresión 3D.
- Integrar sensores de presión en el robot.
- Implementar visión por computador para reconocimiento de objetos.
- Implementar comunicación mediante ROS (brazo y PC).
- Generar pruebas para la manipulación de objetos.
- Validación del brazo robótico mediante pruebas de precisión.

2 Marco teórico

Para llevar a cabo el proyecto, se deben entender conceptos que serán aplicados a lo largo de su desarrollo. Es por ello por lo que en esta sección se expondrán definiciones e información técnica relevante para su comprensión y ejecución.

2.1 Algoritmo

Conjunto definido de reglas o procesos que llevan a la solución de un problema en un número determinado de pasos [45]. La programación del sistema posee una serie de algoritmos que llevan a cabo su funcionamiento.

2.2 Sensor

De acuerdo con lo señalado en el libro *“Expanding the Vision of Sensors Materials”*, se adoptan las siguientes definiciones:

- Elemento sensor: “Mecanismo de transducción fundamental, que convierte una forma de energía en otra”, un sensor puede estar compuesto por más de un elemento sensor.
- Sensor: “Un elemento sensor que incluye su embalaje físico y conexiones externas” [46].
- Sistema de sensores: “Sensor y su hardware de procesamiento (analógico o digital)”, integrado en el mismo embalaje físico, o interconectado con el sensor [46].

Para poder describir y caracterizar el rendimiento de un sensor se requiere comprender algunos términos que definen las principales características del sensor [47], como se muestra en la Tabla 2-1.

Tabla 2-1: Características principales que definen un sensor

Característica	Definición
Sensibilidad	: Entrada mínima de un parámetro físico que creará un cambio de salida detectable. El error de sensibilidad es una desviación de la pendiente ideal de la curva característica.
Rango	: Valor máximo y mínimo del parámetro aplicado que se puede medir. El rango dinámico es el rango total del sensor de mínimo a máximo.
Precisión	: Grado de reproducibilidad de una medida, es decir, rango de valores distribuidos en relación a un valor real correcto.
Resolución	: Cambio incremental detectable más pequeño del parámetro de entrada que se puede detectar en la señal de salida.
Exactitud	: Diferencia máxima que existirá entre el valor real y el valor indicado en la salida del sensor.
Desplazamiento	: Error de desplazamiento se define como la diferencia entre el valor de salida real y el valor de salida especificado bajo algún conjunto particular de condiciones.
Linealidad	: Expresión de la medida en que la curva de medición real de un sensor se aparta de la curva ideal. Su expresión usualmente se mide en porcentaje de no linealidad, como muestra en la ecuación (2-1).
Histéresis	: Medida de la capacidad que tiene un transductor de seguir los cambios del parámetro de entrada.
Tiempo de respuesta	: Tiempo requerido para que la salida de un sensor cambie de su estado anterior a un valor final establecido dentro de una banda de tolerancia del nuevo valor correcto.
Linealidad dinámica	: Medida de la capacidad de un sensor para seguir cambios rápidos en el parámetro de entrada.

$$\text{No Linealidad (\%)} = \frac{D_{in(max)}}{IN_{f.s.}} \times 100 \quad (2-1)$$

Donde:

- No Linealidad (%): es el porcentaje de no linealidad.
- $D_{in(max)}$: es la desviación máxima de entrada.
- $IN_{f.s.}$: es la entrada máxima a escala completa.

Sensor de estiramiento

Un sensor de estiramiento es un componente que genera un cambio de resistencia cuando se estira. Para entender este fenómeno es conveniente revisar el funcionamiento de las galgas extensométricas, basadas en el efecto piezo-resistivo y perteneciente al grupo de los sensores resistivos.

Cuando las galgas extensométricas son sometidas a un esfuerzo mecánico, varían la resistencia del conductor o semiconductor que las conforman. Según lo descrito por Kelvin en el año 1856 [48], para un hilo metálico de longitud l , sección A y resistividad ρ , su resistencia eléctrica estará dada por (2-2).

$$R = \rho \frac{l}{A} \quad (2-2)$$

Si se realiza un esfuerzo longitudinal, cada uno de los elementos que actúan en el valor de R experimentan un cambio, y, por consiguiente, R también presenta una variación en su valor [48]. Esto se muestra en la expresión (2-3).

$$\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dl}{l} - \frac{dA}{A} \quad (2-3)$$

Cuando se aplica una fuerza a un material, éste se estira o comprime como resultado. Este fenómeno se encuentra modelado mediante la Ley de Hooke (2-4).

$$\sigma = \frac{F}{A} = E\varepsilon = E \frac{dl}{l_0} \quad (2-4)$$

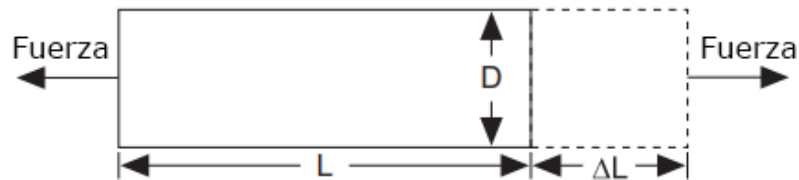
Donde σ representa el esfuerzo, y se define como fuerza aplicada por unidad de área, ε se le denomina a la deformación producida por estiramiento o compresión mientras el material responde al esfuerzo. El esfuerzo es medido como el cociente entre la variación de la longitud l del material y la longitud inicial l_0 [48]. E representa el Módulo de Young y mide la resistencia de un material a ser deformado elásticamente [49].

$$E = \frac{\sigma}{\varepsilon} = \frac{\text{esfuerzo}}{\text{deformación}} \quad (2-5)$$

Un parámetro fundamental es la sensibilidad de deformación, expresada cuantitativamente como Factor de Gauge (2-6). Este factor se define como la relación entre el cambio fraccional en la resistencia eléctrica y el cambio fraccional en la longitud [50].

$$\frac{dR/R}{dl/l_0} = \frac{dR/R}{\varepsilon} = K \rightarrow \frac{\Delta R}{R} = K \cdot \varepsilon \quad (2-6)$$

Donde K es una constante conocida como Factor de Gauge [51].



$$\varepsilon = \frac{\Delta L}{L}$$

Figura 2-1: Definición de estiramiento [50]

2.3 Presión

La presión se define como fuerza por unidad de área aplicada en una dirección perpendicular a la superficie de un objeto [52], descrita en la ecuación (2-7).

$$P = \frac{F}{A} \quad (2-7)$$

Donde P es la presión, F es la fuerza normal y A es la superficie de contacto

2.4 Eslabón

Los eslabones se definen como un conjunto de piezas unidas rígidamente entre sí, sin movimiento posible entre ellas. Al unirse a otros eslabones presentan movimiento relativo entre ellos y se clasifican en dos tipos [53]:

- Eslabones rígidos: Están diseñados para transmitir fuerza, jalar o empujar.
- Eslabones flexibles: Construidos para ofrecer resistencia.
 - Eslabones que actúan a tensión.
 - Eslabones que actúan a presión.

2.5 Articulación

Los eslabones deben estar unidos entre sí, permitiendo un movimiento relativo entre ellos. A este tipo de articulaciones móviles se les denomina también pares cinemáticos [53]. Se pueden clasificar en dos grupos:

- Pares superiores: El contacto es lineal o puntual como, por ejemplo:
 - Contacto de dos dientes engranando.
 - El seguidor con la leva.
 - Una rueda sobre un riel.

- Pares inferiores: El contacto es superficial.
 - Par giratorio.
 - Par prismático.
 - Par de tornillo o par helicoidal.
 - Par cilíndrico.
 - Par esférico.
 - Par plano.

2.6 Cadena Cinemática

“Cuando cada eslabón de la cadena cinemática se conecta al menos con otros dos, esta forma uno o más bucles cerrados, definiéndose una (o varias) cadena cinemática cerrada, en caso contrario se tiene una cadena cinemática abierta” [54].

2.7 Grados de libertad (DoF)

Se denomina grados de libertad al número de parámetros de entrada que deben ser controlados independientemente para llevar un mecanismo a una posición en particular [54].

Un mecanismo puede poseer n eslabones, cada uno poseerá tres grados de libertad antes de conectarse, a excepción de los eslabones que se encuentren fijos. Estos grados de libertad se pueden determinar mediante la siguiente expresión (2-8).

$$3 \cdot (n - 1) \tag{2-8}$$

A medida que éstos se conectan se limita el movimiento relativo entre ellos. Al conectar todos los eslabones, los grados de libertad del mecanismo son representados por la expresión (2-9):

$$m = 3(n - 1) - 2j_1 - j_2 \tag{2-9}$$

Donde m representa la movilidad, n corresponde al número de elementos (eslabones), j_1 el número de uniones de 1 grado de libertad y j_2 el número de uniones de 2 grados de libertad.

2.8 Robot de servicio

Un robot de servicio es “un robot que realiza tareas útiles para humanos o equipos, excluida la aplicación de automatización industrial”.

Las aplicaciones de automatización industrial incluyen, entre otras, fabricación, inspección, embalaje y ensamblaje [55]. La clasificación de un robot en robot industrial o robot de servicio se realiza de acuerdo con su aplicación prevista [56]. Por ejemplo, los robots articulados utilizados en las líneas de producción son robots industriales y los robots articulados similares utilizados para servir alimentos son robots de servicio [55].

Dentro de la categoría de los robots de servicio existen subcategorías a considerar en el desarrollo del proyecto, las cuales se describen a continuación:

2.8.1 Robot de servicio doméstico/personal

Según la *International Organization for Standardization* es un “robot utilizado para una tarea no comercial, generalmente operado por personas no expertas. Algunos ejemplos son los robots de servicio doméstico, la silla de ruedas automatizada y el robot de asistencia de movilidad personal [55]”.

2.8.2 Robot de servicio profesional

“Robot de servicio utilizado para una tarea comercial, generalmente operado por un profesional debidamente entrenado [55]”, En este contexto, un operador es una persona designada para iniciar, supervisar y detener el funcionamiento previsto de un robot.

2.8.3 Robot móvil

Según la ISO, un robot móvil es aquel “que contiene todo lo necesario para su pilotaje y movimiento (potencia, control y sistema de navegación)”. Un robot móvil no tiene por qué ser un robot de servicio propiamente tal, pues existen robots industriales trabajando en almacenaje y logística [55].

2.9 Clasificación de los robots

Los robots pueden ser clasificados según su generación, área de aplicación, tipo de actuadores y número de ejes o morfología. Las categorías que tienen relevancia en el desarrollo del proyecto son:

2.9.1 Clasificación por área de aplicación

Dependiendo del uso que se le proporcione a un robot, este puede ser clasificado de acuerdo con el área de aplicación. Es así como existen dos grandes grupos: clasificación de las aplicaciones de los robots industriales y clasificación de las aplicaciones de los robots de servicio. Como el objetivo de este proyecto se basa en la implementación de un brazo robótico en un robot de servicio, solo se abordará este último tipo de clasificación, tal como se muestra en la Tabla 2-2a y Tabla 2-2b.

Tabla 2-2a: Clasificación de los robots de servicio por áreas de aplicación según la IFR [16]

Sección I	ROBOTS PERSONALES Y DOMÉSTICOS
1-5	Robots para tareas domésticas
6-10	Robots de entretenimiento
11-14	Asistenciales, ayuda a discapacitados
15	Transporte personal
16	Seguridad y vigilancia de la vivienda
17	Otros usos personales y domésticos

Tabla 2-2b: Clasificación de los robots de servicio por áreas de aplicación según la IFR [16]

Sección II	ROBOTS DE SERVICIOS PROFESIONALES
18-23	Asistenciales, ayuda a discapacitados
25-28	Limpieza profesional
29-31	Sistemas de inspección
32-36	Construcción y demolición
37-40	Sistemas logísticos
41-44	Medicina
45-50	Defensa, rescate y seguridad
51	Submarinos
52	Plataformas móviles de uso general
53-55	Robots de laboratorio
56-59	Relaciones públicas
60-61	Propósito especial
62	Humanoides
63	Robots a medida
64	Otros no especificados
Sección III	I+D EN ROBÓTICA
64	Percepción
65-67	Actuación
68	Micro y nano robots
69	Arquitecturas e integración
70	Navegación y control
71	Interfases con usuario y otras
72	Otras actividades de I+D no especificadas
73	Investigación básica

2.9.2 Clasificación por tipo de actuadores

Existen tres categorías para el tipo de actuadores utilizados en un robot. En este caso se utilizan actuadores pertenecientes al último grupo mostrado en Tabla 2-3.

Tabla 2-3: Clasificación de los robots por tipo de actuadores utilizados [16]

Tipo de robot	Ejemplo de aplicación
Robot neumático	En casos que sea necesario contar con aislamiento eléctrico.
Robot hidráulico	En casos que sea necesaria una alta relación carga-peso.
Robot eléctrico	En los casos más comunes se utiliza este tipo de actuador.

2.9.3 Clasificación por número de ejes

De acuerdo con la definición ISO, un robot manipulador industrial debe tener al menos tres ejes (realizando movimientos independientes). Esta definición se extiende también a los robots de servicio que posean la característica de manipulación [55]. Por lo general es posible encontrar robots con 6 ejes o grados de libertad. Robots con un número superior de ejes son escasos [16].

2.9.4 Clasificación por morfología o configuración

Configuración aplicable a robots o telerobots con cadena cinemática. Dado que un robot es considerado como una cadena cinemática formada por eslabones unidos entre sí por articulaciones [57], su movimiento está determinado por el tipo de movimiento permitido entre dos eslabones consecutivos [16]. En la Tabla A-1 del Apéndice A, es posible observar los tipos de configuraciones mecánicas existentes.

2.10 Actuadores

“Un actuador es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o actuar otro dispositivo mecánico [58]“. La fuerza ejercida por el actuador puede ser hidráulica, neumática o eléctrica.

Los actuadores eléctricos tienen ciertas características, por ejemplo, son precisos, de fácil control, instalación sencilla, pero tienen potencia limitada en comparación con los otros tipos de actuadores [59].

Para el desarrollo del proyecto se estableció el uso de servomotores debido a su alto torque y bajo costo. La posibilidad de utilizar motores paso a paso se descartó principalmente por su costo, peso y poco torque, el cual debe ser compensado con una caja de engranajes adicional. Ésta última se encuentra integrada de fábrica en el hardware del servomotor. También se utilizará un actuador lineal, el cual otorga altas fuerzas de tracción y empuje. Las principales características de los actuadores utilizados se pueden consultar en la Tabla 2-4.

2.10.1 Servomotor

El servomotor es un dispositivo que está compuesto por un motor DC que tiene la capacidad de ser controlado en posición. Puede ubicarse en cualquier posición dentro de un rango de operación. Dicho rango, por lo general es de 180°, pudiéndose encontrar servos que operan en 360° o incluso de rotación continua [60].

Su estructura se basa principalmente en 4 elementos (Figura 2-2a):

- Motor de corriente continua o DC: Este elemento permite generar el movimiento del servomotor. Al alimentar los terminales con cierto valor de tensión, se genera movimiento en un sentido o en otro sentido según la polarización aplicada [60].

- Sensor de desplazamiento: Suele tratarse de un potenciómetro que se encuentra ubicado en el eje del servomotor, con el fin de conocer la posición angular del motor [60].
- Circuito de control: Circuito electrónico en placa impresa que realiza control de posición por realimentación. El circuito compara la señal de referencia (entrada) con la posición actual medida por el potenciómetro. Según sea necesario el motor se desplazará en una dirección reduciendo el error [60].
- Engranajes: Existe un juego de engranajes que relaciona el eje del motor con el eje del servomotor, su función es reducir la velocidad del motor y asimismo incrementar su capacidad de torque [60].

La alimentación y el control del servomotor se realiza por medio de tres cables, por lo general negro para la tierra o GND, rojo para la señal de alimentación, y blanco (amarillo o naranja) para la señal de control. La alimentación puede ir de los 4.8 [V] a 6[V], y el control se realiza por medio de pulsos PWM o “*Pulse Width Modulation*”.

PWM (Pulse Width Modulation)

Las señales PWM se forman mediante la comparación de una señal de referencia a modular y una señal portadora de forma triangular o diente de sierra; la comparación genera un tren de pulsos con anchos de pulso específicos [61]. Las señales PWM que controlan los servos se forman por pulsos positivos y su duración determinará la posición donde se debe ubicar el eje del servomotor, estos pulsos se repiten cada 20 [ms] o 50 [Hz]. Para un rango de movimiento de 90°, los pulsos tienen una duración de 0.9 [ms] a 2.1 [ms], sin embargo, servomotores que trabajan en un rango de 180° utilizan pulsos que van de 0.5 [ms] a 2.5 [ms] como se observa en la Figura 2-2b. Si se desea mantener un ángulo en particular se deberá enviar periódicamente el mismo pulso, de lo contrario el eje del servomotor no generará fuerza y quedará inactivo.

2.10.2 Actuador lineal eléctrico

Dispositivo que tiene la capacidad de transformar el movimiento rotatorio de un motor DC en movimiento lineal. Esto permite al motor movimientos de empuje y hale. Se utiliza para realizar acciones como elevar, inclinar, empujar o halar objetos.

Son actuadores de movimiento relativamente silencioso, precisos, de larga vida útil, por lo que no necesita mayor mantenimiento como los sistemas hidráulicos o neumáticos. Además, su estructura permite ejercer grandes fuerzas, la cual se compone de un motor DC, una caja de engranajes y una rosca sin fin de eje que incluye una tuerca [62], como se observa en la Figura 2-3. Para realizar el control del actuador lineal, se utiliza un *driver* L298n, el cual opera entre 5 y 12 volts, y soporta una corriente máxima por puente de 2 [A].

Tabla 2-4: Actuadores utilizados y sus características

Característica			
Tipo de actuador	Servomotor	Servomotor	Actuador lineal
Modelo	MG996r	DS3218	02PZ
Tensión de operación	4.8 [V] a 7.2 [V]	5 [V] a 6.8 [V]	12 [V]
Torque	11 [kg/cm] @6 [V]	21.5 [kg/cm] @6.8 [V]	-
Empuje máximo	-	-	90 [kg]
Velocidad	0.15 seg/60 grados (sin carga)	0.14 sec/60° (sin carga)	5.7 [mm]
Recorrido	180°	180°	50 [mm]
Dimensiones	Largo: 40.7 [mm] Ancho: 18.7 [mm] Alto: 42.9 [mm]	Largo: 40 [mm] Ancho: 20 [mm] Alto: 40.5 [mm]	Extensión máxma: 200 [mm] Extensión mínima: 150 [mm]

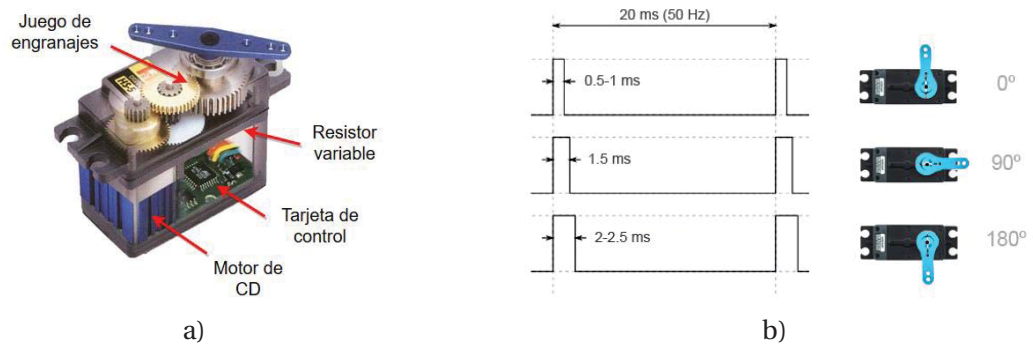


Figura 2-2: a) Componentes del servomotor [63], b) Pulsos PWM y posición de Servomotor [64]

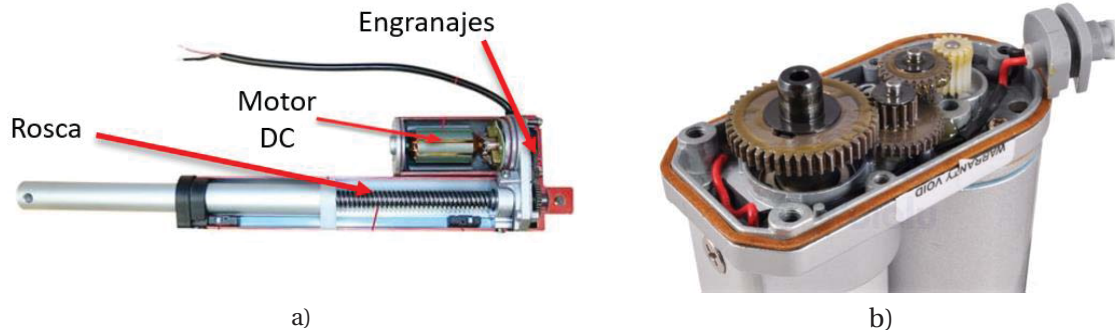


Figura 2-3: a) Principales componentes del actuador lineal, b) Juego de engranajes [65]

2.11 Electrónica

2.11.1 Picaro MEGA (Arduino compatible)

El corazón de la electrónica del brazo robótico es la tarjeta de desarrollo *Picaro MEGA* (Figura 2-4) basada en la *Arduino Mega*, la cual posee un microcontrolador Atmega2560 con 16 entradas analógicas, y 54 entradas y salidas digitales, de las cuales 15 son PWM.

Fue seleccionada debido a la mayor cantidad de entradas y salidas PWM en comparación a otras tarjetas de desarrollo de valor similar y porque su valor es inferior a la tarjeta Arduino Mega.

Algunas de sus características se indican en la Tabla 2-5.

Tabla 2-5: Características de la tarjeta de desarrollo Picaro Mega [66]

Características	
Microcontrolador	ATmega2560
Voltaje de operación	5 [V]
Voltaje de alimentación	7 [V]-12 [V]
Pines digitales I/O	54 (de los cuales 15 proporcionan salida PWM)
Entradas análogas	16
Corriente por pin I/O	20 [mA]
Corriente de 3.3 [V]	300 [mA]
Memoria Flash	256 [kB] de los cuales 8 [kB] son utilizados por el bootloader
SRAM	8 [kB]
EEPROM	4 [kB]
Frecuencia del cristal	16 [MHz]
LED_BUILTIN	13
Largo	101.52 [mm]
Ancho	53.3 [mm]
Peso	37 [g]



Figura 2-4: Tarjeta de desarrollo “Picaro MEGA” [66]

2.11.2 Encoder

Un *encoder* es un transductor rotativo que permite determinar posición angular de un eje, velocidad y aceleración del rotor de un motor. Se componen por un disco conectado a un eje giratorio. El disco se encuentra codificado, presentando secciones transparentes y otras opacas.

Al rotar el eje, un emisor infrarrojo emite luz que es recibida por un fototransistor, generando pulsos digitales a medida que el disco bloquea o permite el paso de este, produciendo una secuencia que puede ser utilizada para controlar giro, velocidad o dirección de giro.

Existen varios tipos de *encoder*: tipo incremental, de cuadratura, absoluto, monovuelta y multivuelta. En este caso se definirá el *encoder* de tipo incremental, que es el que se utilizará en el diseño [67].

Encoder incremental

En la Figura 2-5 se puede ver este tipo de *encoder*, el cual determina el ángulo de posición por medio de cuentas incrementales, donde cada posición es completamente única. El *encoder* seleccionado es denominado “*encoder* incremental de cuadratura” y utiliza dos sensores ópticos con un desplazamiento de $\frac{1}{4}$ de ranura el uno del otro, generando dos señales de pulsos digitales desfasados en 90° o en cuadratura. Mediante las señales de salida es posible suministrar posición, velocidad y dirección de rotación del eje [67]. Sus características se indican en la Tabla 2-6.



Figura 2-5: Encoder incremental de cuadratura marca Grove [68]

Tabla 2-6: Características de la encoder incremental de cuadratura Grove [68]

Características	
Codificación	Incremental
Rotación	360°
Tensión de operación	4.5 [V] a 5.5 [V]
Corriente de operación	10 [mA] hasta 30 [mA]
Dimensiones	20 [mm] x 20 [mm]
Peso neto	12 [g]

2.12 Impresión 3D

La impresión en tres dimensiones (3D) es el proceso de generar un objeto real a partir de un modelo digital mediante la unión de materiales, usualmente mediante la unión de capas sucesivas. Se inició en el año 1984 con la denominada “estereolitografía”, creada por Chuck Hull, el cual utiliza el principio de fotopolimerización para generar modelos 3D a partir de resinas sensibles a la radiación UV [69].

La impresora *RepRap* hizo su aparición en el año 2004 y significó una revolución en la industria debido a su licencia “*OpenSource*” y por ser autorrepliable, disminuyendo los costos de fabricación y adquisición; acercando esta tecnología al usuario común.

El material utilizado por este tipo de impresoras puede ser plástico (ABS, PLA, PCL, policarbonato, entre otros) o metal (Bronzefill, Copperfill, entre otros.), y se le provee a la máquina mediante rollos. El filamento se introduce en una boquilla que se encuentra a una temperatura superior a la de fusión del material. La boquilla puede desplazarse en uno, dos o tres ejes dependiendo del modelo de impresora, generando la estructura mediante finos hilos de material que se solidifican al contacto con el aire y la base donde se deposita.

En particular, para el proyecto fue utilizada la impresora 3D *MakerBot Replicator 2*, filamento de PLA (Ácido poliláctico) con diámetro de 1.75 [mm] de color negro y el software de impresión *MakerBot Desktop* versión 3.10.1.1389.

2.13 Visión por computador

La visión por computador es la ciencia que permite dotar a las computadoras u otras máquinas, de visión o la capacidad de ver [70]. A continuación, se señalan algunos conceptos necesarios para entender la composición de una imagen y el proceso de reconocimiento de objetos y personas.

2.13.1 Imagen digital

Una imagen digital se define matemáticamente como una función bidimensional descrita por (2-10).

$$0 < f(x, y) < \infty \quad (2-10)$$

donde x e y son coordenadas espaciales en un plano, y la función f representa la intensidad o nivel de gris de la imagen en ese punto coordenado. Este tipo de imágenes están compuestas por un número finito de elementos, los cuales se denominan píxel [71] (Figura 2-6).

Un píxel es la unidad mínima de visualización de una imagen digital. Al realizar un acercamiento o *zoom* a una imagen digital, se pueden observar cada uno de estos componentes, los cuales forman una matriz o cuadrícula, por tanto, las imágenes se forman como una sucesión de píxeles [72].

La resolución de una imagen está directamente ligada a la calidad o grado de detalle que se le pueda atribuir. El valor es expresado en “dpi” (dots per inch) o “ppp” (píxeles por pulgada). Mientras sea mayor el número de píxeles por pulgada lineal, la calidad de imagen será mejor [72].

Profundidad de color se refiere al número de bits necesarios para realizar la codificación de la información de color de cada píxel de la imagen. Si la profundidad de color en bits es mayor, la paleta de colores a utilizar será más amplia. Por ejemplo, si se utiliza 1 bit, se obtendrán imágenes en blanco y negro, representando el bit = 0 como negro y el bit = 1 como blanco. Si se utilizan 256 bits, permitirá el uso de 16,7 millones de colores.

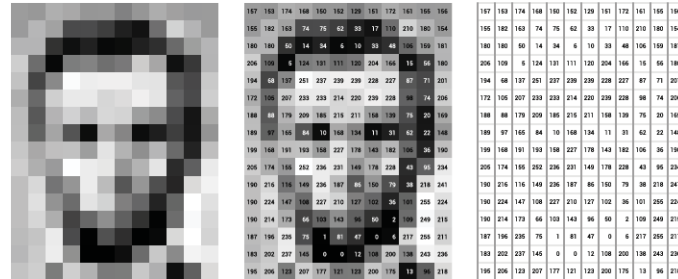


Figura 2-6: Representación de imagen como matriz de 12x16 píxeles y 256 bits de profundidad de color [73]

2.13.2 Procesamiento digital de imágenes

El procesamiento digital de imágenes tiene como objetivo la aplicación de distintas técnicas a imágenes para generar una mejora en su calidad o facilitar el reconocimiento de determinados patrones o información presentes en ellas [74]. Se caracteriza por fases de procesamiento muy marcadas y bien delimitadas, entre ellas se encuentran la digitalización, procesamiento, segmentación, representación y descripción, y el reconocimiento e interpretación (Figura 2-7). Si bien, estos pasos tienden a seguir esta secuencia, no necesariamente estarán presentes todas ellas.

En la actualidad, el procesamiento digital de imágenes se encuentra inmerso en numerosos campos, como, por ejemplo, la medicina nuclear, astronomía, angiografía, toma axial computarizada (CAT), aplicaciones en control de calidad, microscopía, detección remota, policía, entre otros [74].

Dentro de la visión por computador existen tres niveles, estos son:

- Visión de bajo nivel : se asocia a aquellos procesos que podrían ser catalogados como acciones automáticas y no requieren ningún tipo de inteligencia, como la adquisición y el procesado [75].
- Visión de nivel intermedio : tiene asociados todos aquellos procesos que tienen como función extraer, caracterizar y etiquetar componentes de la imagen obtenida en la etapa anterior. Es decir, en esta etapa se involucran las etapas de segmentación, descripción y reconocimiento [75].

Visión de alto nivel : está relacionada con procesos que intentan emular la cognición. Mientras los niveles más bajos se encuentran en un rango de procesos bien definidas, los procesos de alto nivel son complejos y especulativos [75].

En la visión de alto nivel se incluye el reconocimiento e interpretación. El reconocimiento tiene como función asignar etiquetas a un objeto basándose en la información que le entregan sus descriptores. Si los objetos detectados presentan descriptores semejantes se agruparán en una misma clase. La interpretación implica la asignación de un significado a un conjunto de objetos reconocidos [75]. Es importante recalcar que en esta etapa se utilizan modelos o conocimiento previo relacionado al problema.

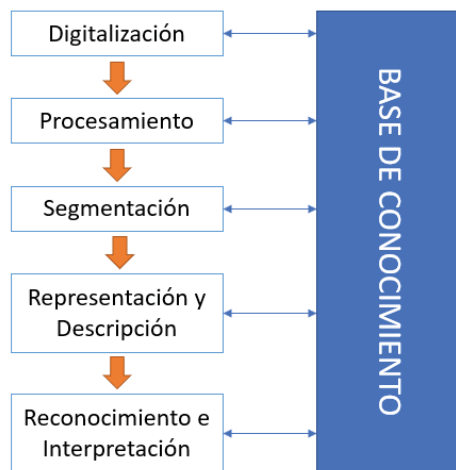


Figura 2-7: Fases del procesamiento de una imagen digital [76]

2.13.3 Deep Learning

El comienzo de las redes neuronales se produce alrededor de la década de 1940 [77] con el propósito de simular el sistema del cerebro humano para resolución de problemas generales de aprendizaje. Entre los años 1980 y 1990 el uso de redes neuronales se popularizó debido a la propuesta de propagación inversa de Geoffrey E. Hinton [78], que es un algoritmo que opera mediante la estimación de la pérdida o error en la salida, propagándolo nuevamente hacia atrás de la red. El objetivo de este algoritmo es actualizar los pesos para minimizar el error resultante de cada neurona, permitiendo el aprendizaje de las redes neuronales [79]. Sin embargo, por falta de datos, computación limitada, y su bajo rendimiento en comparación con otras herramientas de aprendizaje automático, es que fue descartada a inicios de los años 2000.

Ya en el año 2006 el *Deep Learning* reaparece [80], potenciando, además, el desarrollo en el reconocimiento de voz [81]. Ésta nueva alza podría ser atribuible a la existencia de bases de datos de entrenamiento a gran escala, el rápido desarrollo en sistemas computacionales en paralelo de alto rendimiento, permitiendo uso, por ejemplo, de GPU, desarrollo de nuevas arquitecturas, plataformas de software, entre otros.

Las redes neuronales son un modelo computacional basado en un gran número de neuronas artificiales simples, que intentan emular el comportamiento de las neuronas reales, en particular, en la transmisión de la información a través de los axones [79].

Una neurona tiene una estructura similar a la mostrada en la Figura 2-8, Donde X_1, X_2, \dots, X_n son los datos de entrada a la neurona, que podrían corresponder a la salida de otra neurona. X_0 representa el sesgo, un valor constante que es sumado a la entrada de la función de activación de la neurona. Por lo general tiene valor 1. $W_0, W_1, W_2, \dots, W_n$ representan los pesos relativos de cada entrada, incluso la del sesgo o *bias*. Cabe señalar que la función f simboliza la función de activación de la neurona, y puede ser tanto una función lineal como no lineal.

La interconexión entre las neuronas genera redes que son capaces de aprender y modificarse sin necesidad de ser programados explícitamente, debido a que cada neurona tiene la capacidad de potenciar o inhibir a las neuronas adyacentes (Figura 2-9).

Existen distintas arquitecturas dependiendo del tipo de información en la entrada, entre ellas se encuentran las redes neuronales prealimentadas, redes neuronales recurrentes y las redes neuronales convolucionales o CNN.

Las redes neuronales convolucionales tienen gran similitud con las redes neuronales prealimentadas o clásicas. Las neuronas que la conforman poseen sesgos y pesos, permitiendo su aprendizaje. Existe una función de pérdida sobre la última capa, la cual se encuentra completamente conectada. En las CNN se da por supuesto que las entradas son imágenes, permitiendo optimizar la configuración de la red, aumentando su eficiencia.

Existen tres tipos de capas en las redes neuronales convolucionales: capa convolucional (*convolutional layer*), capas de reducción (*pooling layer*) y, por último, una capa clasificadora totalmente conectada (*fully connected layer*).

Las capas convolucionales tienen como objetivo extraer características de la imagen que se está aplicando en la entrada, esto se realiza aplicando un *kernel* o filtro, devolviendo un mapa de características de la imagen original. Este procedimiento es sumamente conveniente, pues se logra la reducción del tamaño de los parámetros y cálculo.

La capa de *pooling* permite la reducción espacial de la entrada, adecuándola a la siguiente capa convolucional. Este proceso produce pérdida de información, sin embargo, la disminución en el tamaño reduce la cantidad de cálculos a realizar en las siguientes capas, además de bajar el sobreajuste. Usualmente la operación utilizada es de *max-pooling*, la cual divide a la imagen de entrada en regiones, donde permanece solo el valor máximo de cada una (Figura 2-10).

Las capas de clasificación o *fully connected layer* se encuentran completamente conectadas, en estas capas, cada píxel se considera una neurona como en las redes neuronales tradicionales, y existirán tantas capas de salida como número de clases a predecir [79].

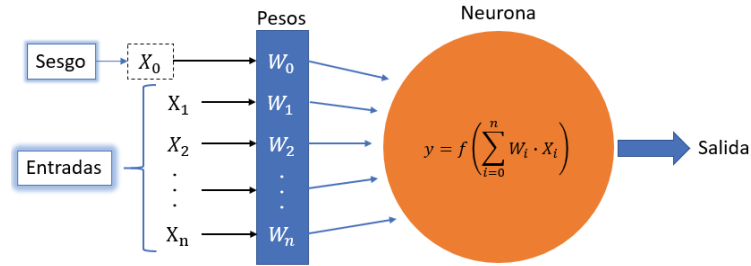


Figura 2-8: Representación típica de una neurona simple

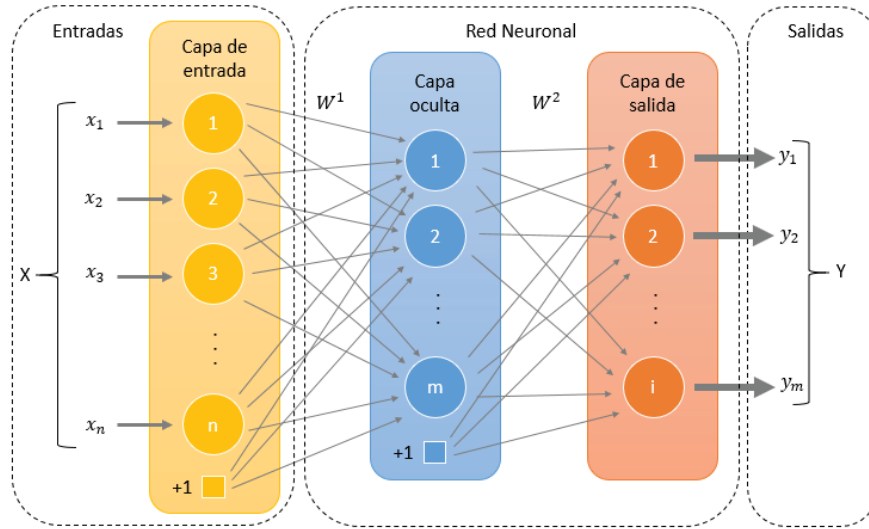


Figura 2-9: Representación de una red neuronal

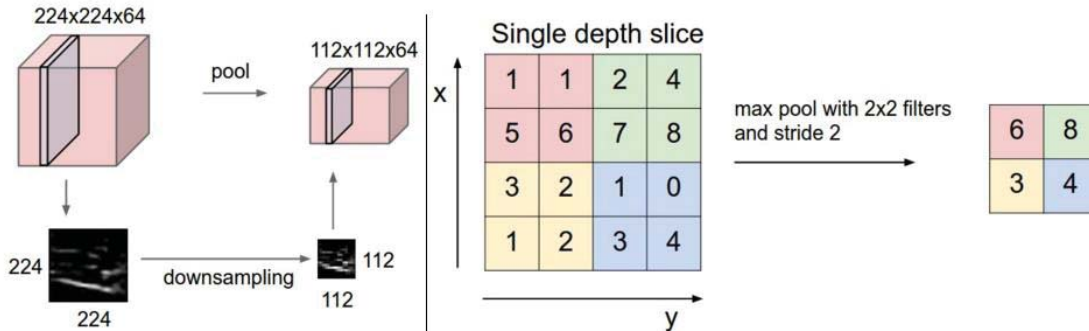


Figura 2-10: Proceso de *Max-pooling* con un filtro de 2x2 y un paso de 2 píxeles [82]

2.14 Localización y detección de objetos con Deep Learning

Para este proyecto, la red neuronal seleccionada consiste en una CNN o Red Neuronal Convolutiva, debido a que se trabajará con imágenes para realizar la detección y reconocimiento de objetos.

Los algoritmos de detección intentan dibujar un cuadro delimitador alrededor del objeto de interés para ubicarlo dentro de la imagen. Podrían también existir un elevado número de cuadros

delimitadores representando diferentes objetos de interés. Al construir una red convolucional estándar seguida de una capa totalmente conectada se presenta un problema, pues la longitud de la capa de salida es variable, esto se debe a que el número de ocurrencias de los objetos de interés no está determinado. Una manera de dar solución a este problema es tomar distintas regiones de interés, para luego utilizar una CNN para clasificar la presencia de un objeto en esa región. El inconveniente que surge entonces es que los objetos podrían presentar distintas ubicaciones y escalas dentro de la imagen, lo que conllevaría a aumentar el número de regiones y, por consiguiente, incrementar el consumo de recursos computacionales [83].

Como solución al problema planteado, se han generado algoritmos y modelos como por ejemplo R-CNN, Fast R-CNN, Faster R-CNN y SSD, que permiten encontrar estas ocurrencias rápidamente.

Los primeros dos métodos realizan una propuesta de región y clasificación en dos fases distintas. Si bien, estos métodos son muy precisos, conllevan un sacrificio en la velocidad de fotogramas por el alto costo computacional que requieren [82]. En la Figura 2-11 se muestra la comparación de su desempeño en el tiempo.

Existen otros métodos denominados *Single Shot Detectors*, que combina los dos procesos en un solo paso, proporcionando los cuadros de delimitación y la clase al momento que la red examina la imagen [84].

SSD (Multibox Single Shot Detector): A partir de una imagen de entrada y un conjunto de etiquetas para el *dataset*, se realizan los siguientes procesos:

- La imagen pasa a través de una serie de capas convolucionales, produciendo mapas de características a diferentes escalas [84].
- Por cada posición en cada uno de los mapas de características se aplica un filtro convolucional de 3x3 para evaluar un pequeño grupo de recuadros delimitadores predefinidos, que serían un equivalente a los cuadros de anclaje (anchor boxes) del modelo Faster R-CNN como se muestra en la Figura 2-12 [84].
- Por cada recuadro se establece un recuadro del *bounding box* o cuadro delimitador y la probabilidad para cada clase [84].

En la Figura 2-13 se observan los ejemplos de detección realizados por los autores de [85] con un modelo SSD512 y el conjunto de pruebas de imágenes COCO con una puntuación de detección es superior a 0.6.

La estructura del modelo SSD por lo general comienza con una VGG, en el modelo pre-entrenado de Resnet, siendo una red completamente de convolución. Posteriormente se adjuntan algunas capas adicionales de convolución, con el fin de ayudar a manejar objetos de mayor tamaño. Una vez que la imagen pasa por la red VGG, se producen mapas de características de tamaños 19x19, 10x10, 5x5, 3x3 y 1x1 que, en conjunto con el mapa de características de 38x38 producido por el Conv4_3 de la VGG, serán los mapas de características utilizados para predecir los cuadros delimitadores (

Figura 2-14). Es importante recalcar, además, que la capa Conv4_3 se encarga de detectar los objetos más pequeños, mientras que la capa Conv11_2 detecta los objetos más grandes [82].

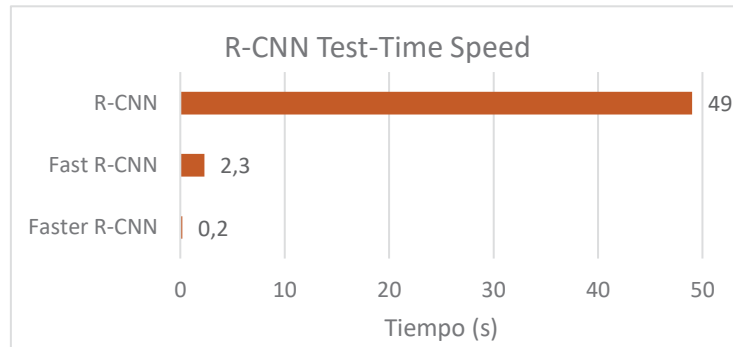


Figura 2-11: Comparación de tiempo de los algoritmos R-CNN, Fast R-CNN y Faster R-CNN [82]

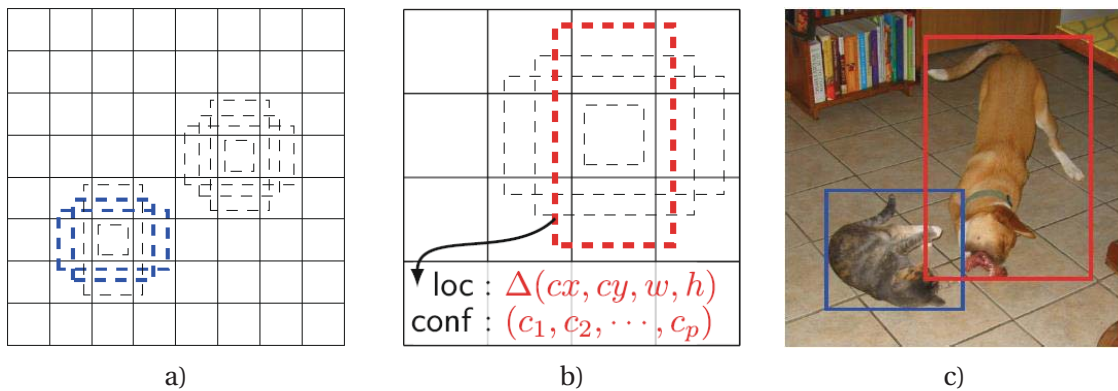


Figura 2-12: a) Mapa de características de 8x8, b) Mapa de características de 4x4, c) Imagen de entrada con Ground True Boxes (GT) [85]



Figura 2-13: Ejemplos de detección en conjunto de pruebas COCO con modelo SSD512 donde la puntuación de detecciones son superiores a 0.6 [85]

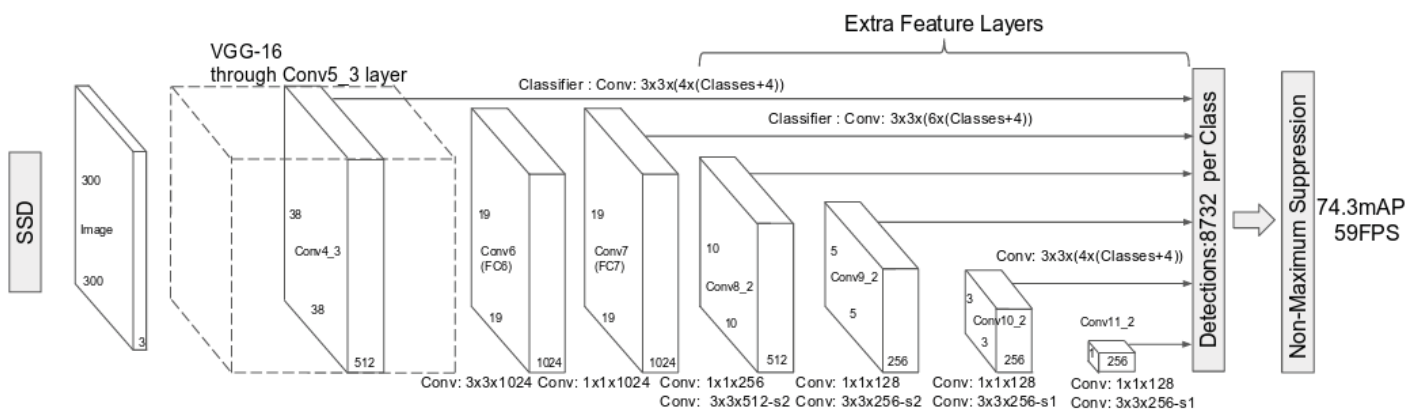
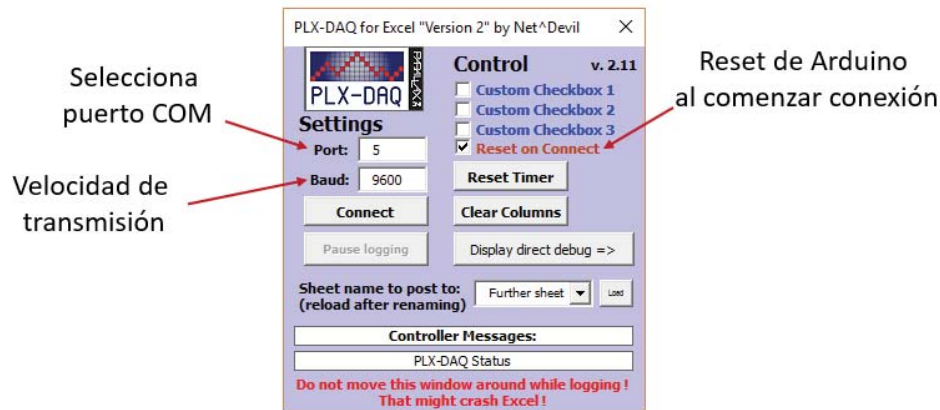


Figura 2-14: Esquema de modelo SSD [86]

2.15 Software para adquisición de datos por conexión serial (PLX-DAQ)

PLX-DAQ es un software programado en *Visual Basic* que permite la adquisición de datos mediante conexión serial, escribiéndolos en una plantilla Excel para su posterior procesamiento. En la interfaz del programa se puede seleccionar el puerto *COM* de conexión, la velocidad de transmisión en *baudios*, entre otras opciones que el usuario puede manipular según su necesidad (Figura 2-15). Sin embargo, para que los mensajes sean coherentes para el programa, deben tener una estructura determinada como se muestra en el Listado 2-1, donde se deben enviar cadenas de caracteres que el programa en Microsoft Excel ejecutará como comandos, por ejemplo limpiar la hoja de cálculo y etiquetar los encabezados de las columnas con el nombre que se asigne en el script. Luego, al presionar *Connect* en la interfaz gráfica y con el programa ejecutándose en el microcontrolador, los datos se irán grabando en la plantilla.

Figura 2-15: Interfaz de programa PLX-DAQ para realizar la adquisición de datos



Listado 2-1: Script Arduino base para envío de datos a través de puerto serial

```

1 int voltaje = 0;
2 float tiempo;
3 i = 0;
4
5 void setup() {
6     // inicializar puerto serie
7     Serial.begin(9600);
8     Serial.println("CLEARDATA");
9     Serial.println("LABEL,Muestra,Segundos,Voltaje");
10
11 }
12
13 void loop() {
14     // leer pines
15     voltaje= analogRead(A0);
16     // enviar
17     tiempo = millis()/1000.0;
18     Serial.print("DATA,");
19     Serial.print(i++);
20     Serial.print(",");
21     Serial.print(muestra);
22     Serial.print(",");
23     Serial.print(tiempo);
24     Serial.print(",");
25     Serial.println(voltaje);
26     // esperar
27     delay(20);
28 }

```


3 Desarrollo

En este apartado se expondrá el proceso de diseño del brazo y mano robótica, considerando cada uno de sus componentes. Del mismo modo, se presentará el diseño del sensor de presión textil de tipo resistivo, el cual está basado en el artículo [1], para luego dar paso al proceso de construcción tanto del brazo como del sensor.

3.1 Diseño mecánico de brazo y mano robótica

Debido a los altos costos que implica adquirir un brazo robótico comercial, se determinó realizar el diseño de este. Para ello, se debieron estudiar ciertas funciones de las herramientas CAD, principalmente cómo realizar el croquizado de una pieza y posteriormente su transformación en modelo 3D. También fue necesario realizar un modelo funcional, considerando características como: dimensiones, peso, tipo de material, grados de libertad, limitaciones del área de trabajo, entre otros.

“Un manipulador debe tener al menos seis grados de libertad para poder ubicar su efector final en un punto arbitrario con una orientación arbitraria en el espacio (Figura 3-1). Manipuladores con menos de seis grados de libertad no cumplen este criterio. Por otro lado, manipuladores con más de seis grados de libertad tienen un número infinito de soluciones para su ecuación cinemática y se conocen como manipuladores redundantes” [87], sin embargo, dadas las condiciones en donde se desenvolverá el robot de servicio, éste se ha limitado solo a 4 grados de libertad (DoF), siendo un quinto grado de libertad el movimiento de la plataforma.



Figura 3-1: Representación de brazo humano con seis grados de libertad

3.1.1 Gripper o actuador final

La primera versión de *gripper* que fue impresa en 3D corresponde al modelo *STL* disponible en la página de la compañía *Roboteus, Inc.* La compañía se encuentra enfocada en las necesidades de la robótica aplicada a la educación para estudiantes de todos los niveles [88].

En el diseño del fabricante, la sección de la base del *gripper* presenta una perforación para la inserción del eje de un motor paso a paso, el cual soporta toda la pieza (Figura 3-2). Para hacer uso de la mano robótica se debió diseñar una pieza que pudiese acoplarse a ésta.

Se utilizó la separación existente entre las dos piezas que conforman la base del efector final (superior e inferior), además de dos muescas en los extremos de su base como se muestra en la Figura 3-3a. Mediante los orificios para los dos tornillos se fijó la mano, los mismos que fueron incorporados en la pieza de la muñeca de unión con la mano para su fijación.

Luego de realizar el montaje de la pieza se determinó que su configuración es frágil considerando las cargas que teóricamente se desean manipular, pues no existe un área de apoyo estable ni suficientemente grande, por lo tanto, fue descartada.

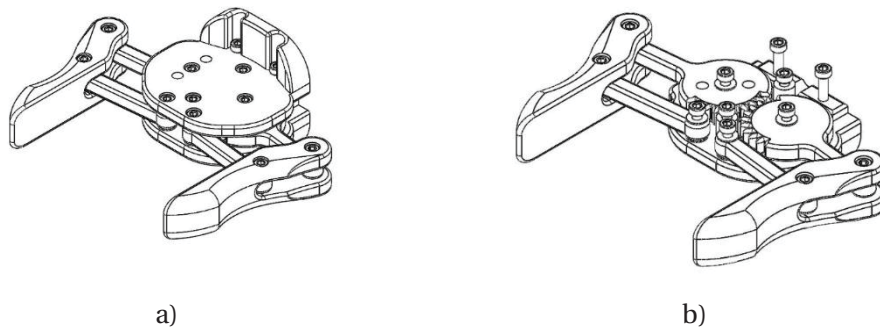


Figura 3-2: a) *Gripper* de brazo robótico RBX1 de la compañía Roboteus
b) Mecanismo de acción de *gripper* [89]

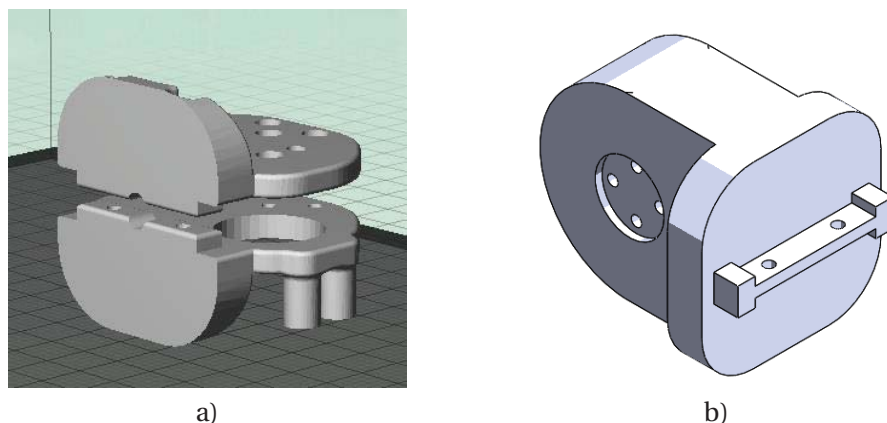


Figura 3-3: a) Base de mano robótica, agujero central para eje de actuador, y muescas que fueron aprovechadas para la sujeción de la pieza a la muñeca. b) pieza de muñeca diseñada para sujeción

Utilizando como base el diseño de la compañía *Roboteus*, se generó una nueva pieza, considerando un sistema de sujeción de mayor robustez, y pilares de descanso entre las dos piezas para minimizar el roce de los componentes móviles (Figura 3-4). También se alargaron los soportes de los dedos con el objetivo de aumentar la apertura y capacidad manipular objetos de mayores dimensiones.

En la Figura 3-5 se muestran dos tipos de soportes diseñados para accionar los dedos, los que poseen engranajes en su parte inferior corresponden a los soportes que se encontraran directamente relacionados con el actuador de la mano que realizará el movimiento de apertura y cierre de los dedos. Los soportes lisos tienen función de guía para realizar un accionamiento controlado de los dedos como se observa en la Figura 3-6.

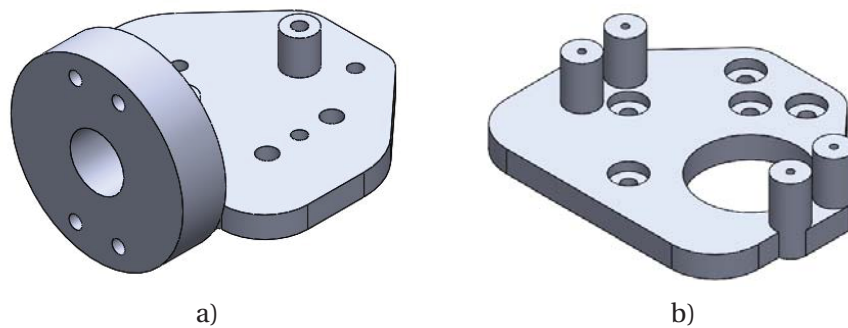


Figura 3-4: a) Base circular y sección inferior de mano robótica, b) sección superior de mano robótica y soporte para servomotor

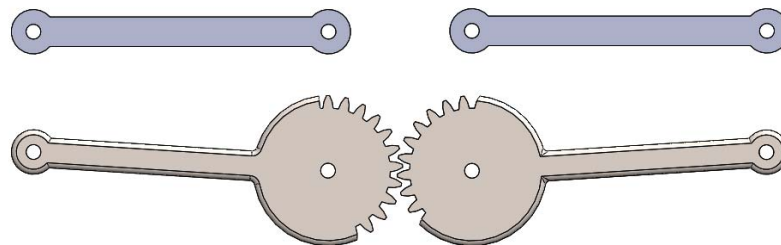


Figura 3-5: Soporte con engranajes para permitir movimiento simultaneo de los dos dedos y soportes lisos con función de guía en el movimiento de apertura y cierre

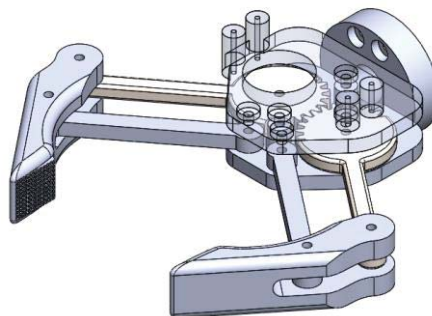


Figura 3-6: Ensamble de mano robótica

3.1.2 Muñeca

La articulación que une el antebrazo con la mano o *gripper* se compone de cuatro elementos. Soporte de gripper, Primera pieza de la articulación de la muñeca, Segunda pieza de la articulación de la muñeca y Soporte de muñeca – antebrazo, cuyas especificaciones se detallan a continuación:

Soporte de gripper

El diseño del soporte para la mano del brazo robótico tiene como propósito generar un sistema de anclaje que permita un mayor contacto entre las superficies, un acople seguro y estable. Está pensado además para que su estructura sea modular, es decir, que permita el cambio del efector final dependiendo del uso que el usuario desee dar a la estructura.

En la parte frontal de la pieza se puede anclar al actuador final con cuatro pernos de 3 [mm] y existe un ducto en la parte central que desemboca en la parte inferior para pasar cables de conexión (Figura 3-7a).

A los costados posee dos extrusiones de corte para albergar un rodamiento de 5 [mm] de radio interno, 16 [mm] de radio externo y 6[mm] de altura para evitar el roce del eje con la pieza soporte. El otro extremo posee cuatro perforaciones para unir un disco de acople de servomotor, actuador que permitirá el movimiento de rotación de 180 grados en relación con el eje de unión (Figura 3-7b).

Primera pieza de la articulación de la muñeca

La siguiente unión se produce entre el soporte del *gripper* y la articulación de la muñeca, la cual supone uno de los cuatro grados de libertad de la estructura. En uno de sus lados presenta una extrusión de corte para insertar el actuador, que en este caso se trata de un servomotor con giro de 180 grados (Figura 3-8a). En la vista posterior se generaron cuatro orificios para acople de un segundo servomotor que generará giro en torno al mismo eje del antebrazo (Figura 3-8b). La porción sobresaliente se adaptará a un rodamiento que hará de unión con la tercera pieza de la muñeca.

Segunda pieza de la articulación de la muñeca

La segunda pieza de la articulación de la muñeca representa un segundo grado de libertad del brazo robótico, se acopla mediante un rodamiento de 20 [mm] de diámetro interno, 52 [mm] de diámetro externo y 15 [mm] de altura, permitiendo al servomotor producir el giro de la primera pieza de la articulación de la muñeca sin sobrecargar el eje del motor. Su estructura permite albergar el servomotor en su interior. Inicialmente se diseñó la pieza de sujeción del servomotor y antebrazo en una sola estructura, pero debido a la complejidad en la instalación del actuador fue desechada. Un segundo motivo de su descarte es porque el peso de la estructura de la muñeca y mano recae en su totalidad en el eje del servomotor. El diseño final que se puede observar en la Figura 3-9.

Soporte de muñeca-antebrazo

La última pieza que conforma la muñeca es el soporte que une la articulación con el antebrazo. Su diseño permite que sea acoplada a la tercera pieza de la articulación de la muñeca mediante tres tornillos y en su otro extremo posee un sistema de abrazadera como se puede visualizar en la Figura 3-10.

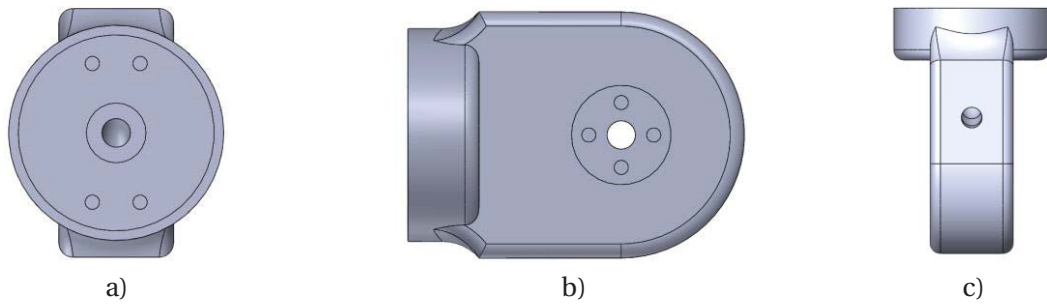


Figura 3-7: a) Vista frontal, b) vista lateral de pieza, c) vista inferior de la pieza soportadora para mano robótica

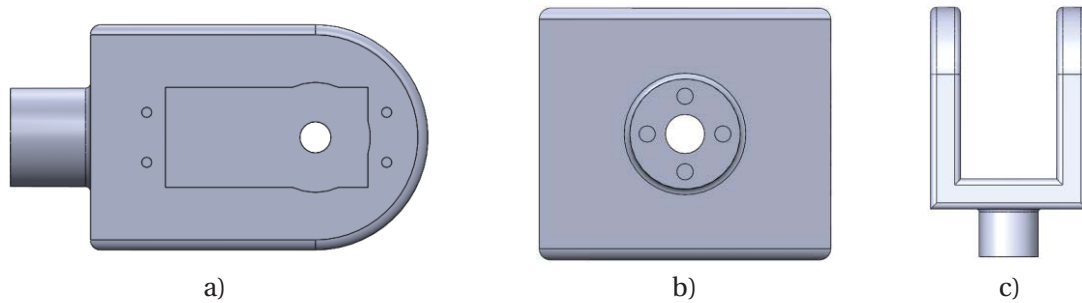


Figura 3-8: a) Vista lateral, b) Vista posterior, c) vista superior de la articulación de la muñeca

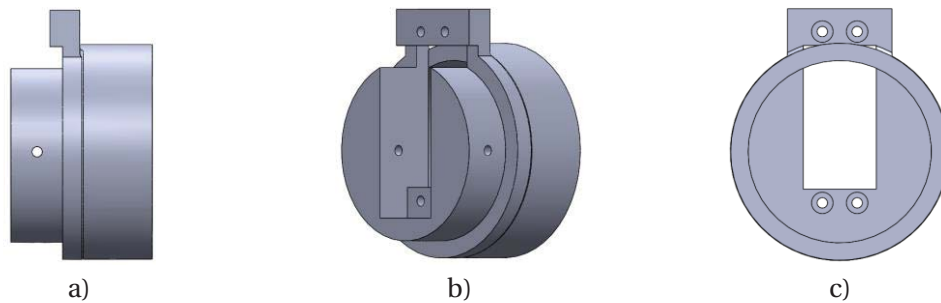


Figura 3-9: a) Vista lateral, b) vista en perspectiva oblicua posterior, c) vista frontal de tercera pieza de articulación de la muñeca

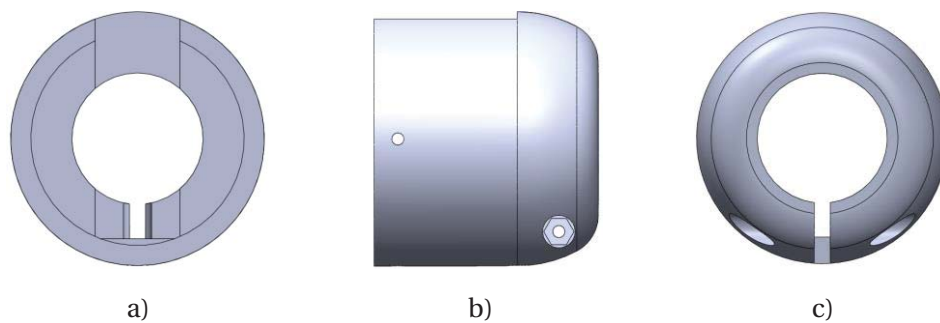


Figura 3-10: a) Vista frontal, b) Vista lateral, c) vista posterior del soporte muñeca antebrazo

3.1.3 Codo

La articulación del codo se compone de dos piezas principales que tienen como función unir el antebrazo con el brazo. Su diseño actúa como pivote para el actuador que genera el movimiento del codo, ya que éste no se encuentra situado en el mismo eje de la articulación.

Pieza antebrazo-codo

El tubo del antebrazo se ajusta a la articulación del codo por medio de tornillos que son introducidos en orificios presentes en la pieza. En su parte central e inferior cuenta con un ducto que tiene como propósito el paso de los cables de los actuadores de la mano y muñeca, además de los cables propios de los sensores que serán instalados en la mano robótica en las siguientes etapas del proyecto tal como se observa en la Figura 3-11a y Figura 3-11b.

La Figura 3-11c muestra un orificio en la parte lateral en la zona posterior de la pieza, el cual realiza funciones de soporte y pivote del antebrazo. ahí será encajado el eje de la articulación del codo.

Pieza codo-brazo

La pieza que une el codo con el brazo presenta mayor complejidad en su construcción debido a que se diseñó para albergar un *encoder* que permitirá determinar el ángulo en el que se encuentra posicionado el antebrazo respecto del brazo, debido a que el actuador utilizado para realizar el movimiento no tiene realimentación propia.

En la Figura 3-12a se muestra el orificio del eje que unirá la pieza antebrazo-codo y una sección para la instalación del *encoder*. Un juego de engranajes será adaptado tanto al eje del *encoder* como al eje de unión de las piezas del codo y se posicionará en las perforaciones que se encuentran en la parte interna de la pieza como se muestra en la Figura 3-12b.

Los dos engranajes fueron diseñados con 20 dientes considerando la resolución y calidad de impresión de la impresora 3D MakerBot Replicator 2, y se distinguen por la forma de su eje, adaptados al eje de rotación de la articulación y eje del *encoder* respectivamente (Figura 3-13).

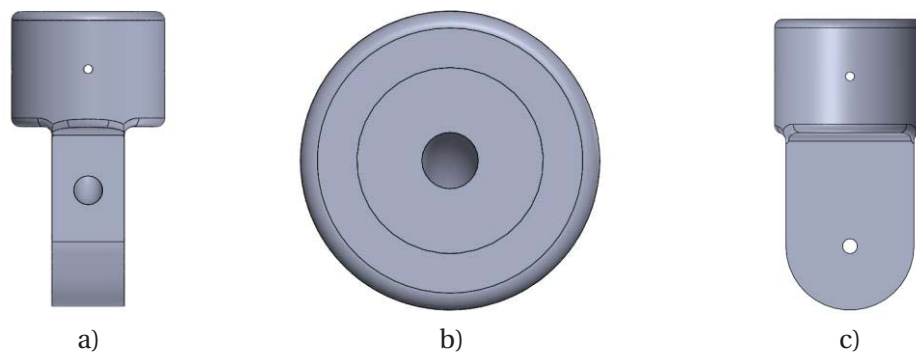


Figura 3-11: a) Vista inferior, b) vista frontal, c) vista lateral de pieza unión antebrazo-codo

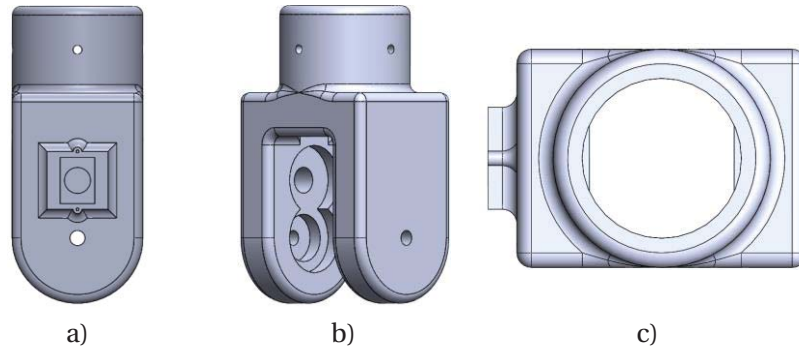


Figura 3-12: a) Vista lateral, b) vista en perspectiva oblicua, c) vista superior de la pieza codo-brazo

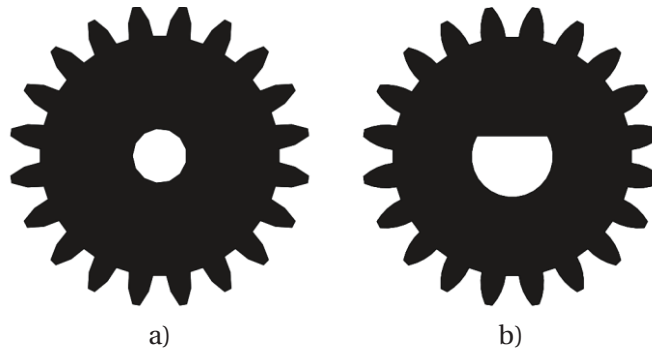


Figura 3-13: a) Engranajes de 20 dientes con eje circular, b) engranaje de 20 dientes adaptado a eje de *encoder*

3.1.4 Hombro

El hombro se conforma de 2 piezas principales, la primera ha sido diseñada para anclarse a la plataforma y ser el soporte de la segunda pieza que contendrá la estructura del brazo robótico, además de actuar como cuarto grado de libertad.

Soporte de plataforma

El soporte de plataforma se diseñó para acoplar el brazo robótico a la plataforma móvil del robot de servicio que se encuentra en desarrollo en el Laboratorio de Robótica. En su parte frontal cuenta con una sección rectangular que calza con la pieza de soporte para el brazo (Figura 3-14a). En su zona media se observa un surco donde conecta la plataforma (Figura 3-14b) y en la parte superior presenta un orificio central donde cruza una de las barras de acero que conforman la plataforma (Figura 3-14c). De esta forma, la estructura se encontrará fuertemente asegurada a la base.

Soporte de Brazo

La pieza está conformada por una estructura cilíndrica que se adhiere al soporte de plataforma mediante dos pernos hexagonales (Figura 3-15), y otra pieza que encaja en ella soportando la estructura del brazo. En su interior alberga un rodamiento axial que minimiza el roce que produce la fricción entre las dos piezas.

En la parte superior del cilindro que contiene el tubo del brazo se acopla un servomotor (Figura 3-16a). Luego el actuador es fijado mediante una pieza denominada soporte de servomotor, permitiendo la rotación del hombro (Figura 3-16b).

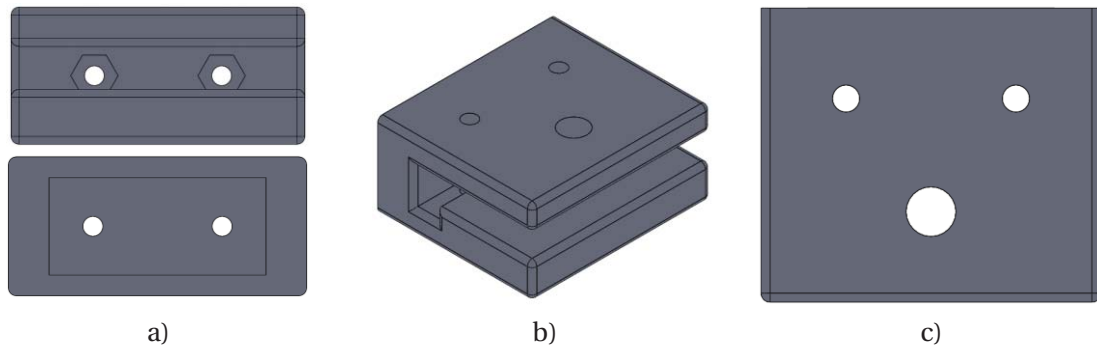


Figura 3-14: a) vista posterior y frontal, b) vista isométrica, vista superior de soporte de plataforma

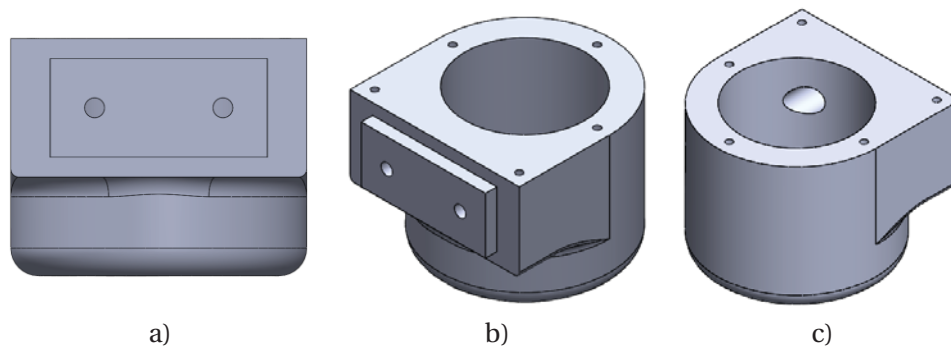


Figura 3-15: a) Vista posterior, b) vista isométrica posterior, c) Vista isométrica de soporte de brazo

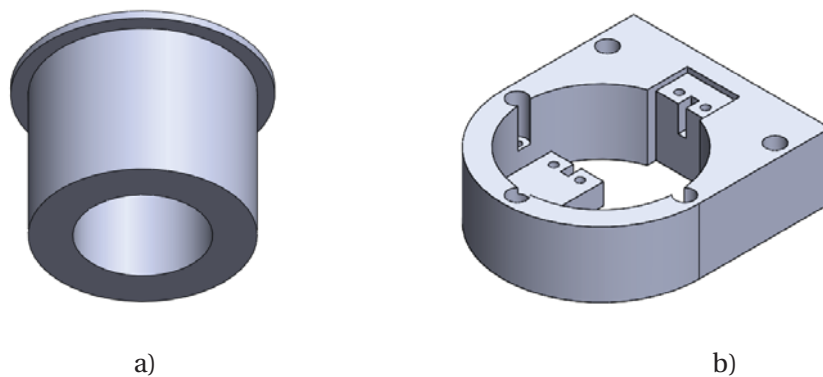


Figura 3-16: a) Soporte de tubo de brazo robótico, b) Soporte para servomotor

3.1.5 Análisis de torque

Antes de realizar la impresión 3D de los componentes estructurales del brazo, se debe realizar una estimación de los torques que ejercerán los motores para permitir el movimiento de los objetos de interés. Además, es importante considerar el torque ejercido en los mecanismos que son sometidos a cargas importantes como, por ejemplo, el hombro.

El análisis de torque desarrollado a continuación ayuda a determinar los actuadores que mejor se adaptan a las fuerzas ejercidas en la estructura. El desarrollo del análisis de torques es realizado con una carga externa de 0.5 [kg] ubicado en la mano del brazo robótico, asumiendo que el soporte del hombro se encuentra fija a una estructura rígida (ideal), es decir, no presenta deformación:

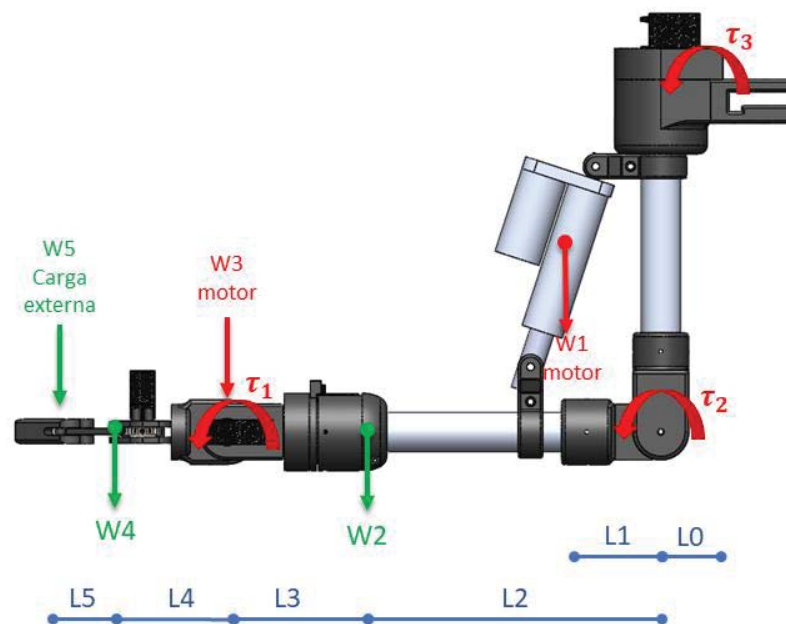


Figura 3-17: Representación de las fuerzas ejercidas en el brazo robótico, las longitudes de sus eslabones y los torques ejercidos en cada articulación

$$\tau_1 = W5 \cdot (L5 + L4) + W4 \cdot L4 = 0.5 \text{ [kg]} \cdot (6.79 \text{ [cm]} + 8.11 \text{ [cm]}) + 0.214 \text{ [kg]} \cdot 8.11 \text{ [cm]}$$

$$\tau_1 = 9.19 \text{ [kg} \cdot \text{cm]}$$

$$\begin{aligned} \tau_2 &= W5 \cdot (L5 + L4 + L3 + L2) + W4 \cdot (L4 + L3 + L2) + W3 \cdot (L3 + L2) + W2 \cdot L2 = \\ \tau_2 &= 0.5 \text{ [kg]} \cdot (6.79 \text{ [cm]} + 8.11 \text{ [cm]} + 7.87 \text{ [cm]} + 27.05 \text{ [cm]}) + 0.21 \text{ [kg]} \cdot (8.12 \text{ [cm]} + 7.87 \text{ [cm]} \\ &\quad + 27.05 \text{ [cm]}) + 0.06 \text{ [kg]} \cdot (7.87 \text{ [cm]} + 27.05 \text{ [cm]}) + (0.57 \text{ [kg]} \cdot 27.05 \text{ [cm]}) \\ \tau_2 &= 24.91 \text{ [kg} \cdot \text{cm]} + 9.04 \text{ [kg} \cdot \text{cm]} + 2.10 \text{ [kg} \cdot \text{cm]} + 15.42 \text{ [kg} \cdot \text{cm]} \end{aligned}$$

$$\tau_2 = 51.47 \text{ [kg} \cdot \text{cm]}$$

$$\begin{aligned} \tau_3 &= W5 \cdot (L5 + L4 + L3 + L2 + L0) + W4 \cdot (L4 + L3 + L2 + L0) + W3 \cdot (L3 + L2 + L0) + W2 \\ &\quad \cdot (L2 + L0) + W1 \cdot (L1 + L0) \\ \tau_3 &= 0.5 [kg] \cdot (6.79 [cm] + 8.11 [cm] + 7.87 [cm] + 27.05 [cm] + 5.30 [cm]) + 0.21 [kg] \\ &\quad \cdot (8.12 [cm] + 7.87 [cm] + 27.05 [cm] + 5.30 [cm]) + 0.06 [kg] \\ &\quad \cdot (7.87 [cm] + 27.05 [cm] + 5.30 [cm]) + 0.57 [kg] \cdot (27.05 [cm] + 5.30 [cm]) \\ &\quad + 0.9 [kg] \cdot (7.70 [cm] + 5.30 [cm]) \\ \tau_3 &= 27.56 [kg \cdot cm] + 10.15 [kg \cdot cm] + 2.41 [kg \cdot cm] + 18.44 [kg \cdot cm] + 11.7 [kg \cdot cm] \\ \tau_3 &= 70.26 [kg \cdot cm] \end{aligned}$$

El torque generado en la muñeca es de 9.19 [kg · cm], por ese motivo se ha seleccionado un servomotor con un torque de 11 [kg · cm]. El torque que es generado en el codo corresponde a 51.47 [kg · cm]. El actuador lineal tiene la capacidad de halar cerca de 90 [kg] de forma lineal, por ello no se omiten los cálculos asociados a este dado que está sobredimensionado para las cargas que se pueden manejar. Además, el peso del motor y el antebrazo descansa en la abrazadera y tubo de aluminio del brazo.

En el caso de τ_3 , se debe prestar atención al efecto que produce en el hombro, dado que genera un efecto palanca que se debe considerar a la hora de la construcción, donde el torque ejercido por la estructura es de 70.26 [kg · cm].

Se omite el análisis para el servomotor que genera la rotación de toda la muñeca y mano debido a que posee un mayor torque, es decir, se encuentra dentro de los rangos necesarios para operar sin problemas. Igualmente, el servomotor que rota el hombro posee torque suficiente para cumplir su función.

3.2 Diseño sensor de presión textil de tipo resistivo

Como este tipo de sensor debe ser diseñado para su instalación en los dedos de una mano robótica, deben ser fabricados de un material flexible, permitiendo su adaptación a superficies redondeadas. Es por ello por lo que se determinó el uso de sensores de presión de materiales maleables como son la tela conductora y las bolsas antiestáticas.

Para realizar el diseño del sensor de presión de tipo resistivo fabricado con material textil, se utilizó como referencia el artículo *Easy-to-Build Textile Pressure Sensor* [1], donde se describe paso a paso el desarrollo de este tipo de elementos.

La configuración propuesta en la literatura [1], muestra que solo es necesario cortar dos porciones de tela conductora. Estas deben ser separadas por un material semiconductor con un margen que sobrepase las dimensiones de la tela conductora para que proporcione aislación entre las capas exteriores (Figura 3-18).

La combinación de las láminas textiles conductoras con la lámina de polímero produce un efecto piezo-resistivo [1], gracias a este efecto la presión que es ejercida en el sensor es transformada en variaciones de resistencia. La resistencia eléctrica de la capa intermedia disminuye cuando se

aplica presión, y se produce por el cambio de distancias entre las partículas de carbono conductoras que componen el material [90], como se muestra en la Figura 3-19.

Las dimensiones de los sensores se establecieron de acuerdo con el tamaño de los dedos de la mano robótica. Se respeta la simbología utilizada en el artículo [1], y se define W_{cond} como el ancho y l_{cond} como el largo del material conductor. Tanto W_{cond} como l_{cond} tienen 1.5 [cm] de longitud. La sección donde se instalará el terminal conductor se posiciona en la sección inferior del sensor, y sus dimensiones se han establecido como W_{term} y l_{term} , con un valor de 0.375 [cm]. El tamaño de la sección del polímero se identificó como W_{LDPE} y l_{LDPE} , que forman un rectángulo de 2 [cm] y 2.5 [cm] respectivamente, como se observa en la Figura 3-20.

Para la sección interna de los dedos, la forma del sensor es rectangular, con un ancho de la tela conductora de $W_{cond} = 0.75$ [cm] y un largo $l_{cond} = 2,625$ [cm], en tanto el ancho y largo del polímero sobrepasa alrededor de un 5% el material conductor.

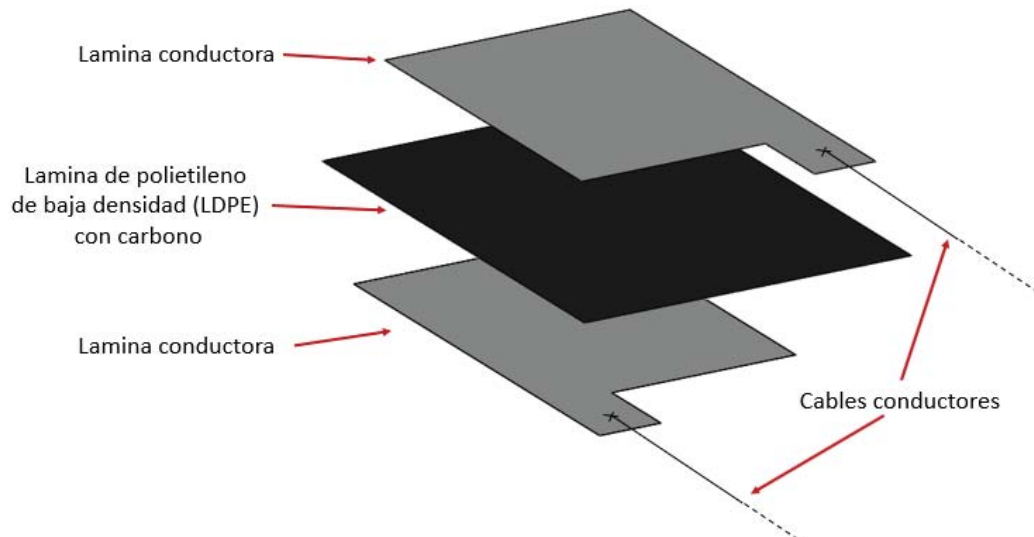


Figura 3-18: Representación del diseño propuesto por [1], donde se observan las láminas conductoras en los extremos del sensor separadas por la lámina de polietileno de baja densidad con partículas de carbono

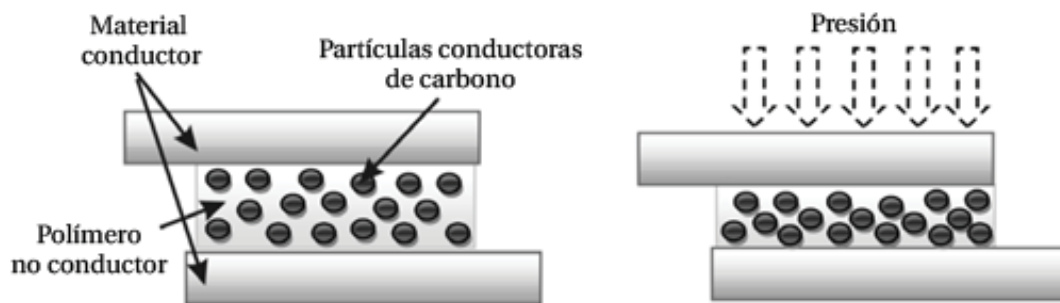


Figura 3-19: Comportamiento de sensor al aplicar presión en la superficie del material conductor. Las partículas de carbono se acercan generando una disminución de la resistencia eléctrica [90]

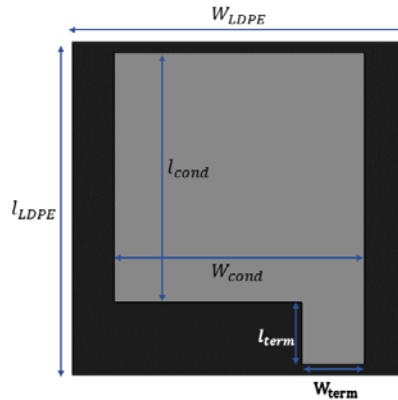


Figura 3-20: Vista superior y dimensiones del sensor de presión textil resistivo [1]

3.2.1 Materiales y herramientas

Los principales materiales utilizados para la construcción de los sensores de presión textiles resistivos son los siguientes:

Shieldex NoraDell: Tela de alta conducción de uso general, compuesta por níquel, cobre, plata y nylon ripstop, que es un tejido antidesgarro. Su resistencia superficial es menor a $0.009 [\Omega]$. Resiste temperaturas entre $40 [^{\circ}C]$ y $100 [^{\circ}C]$. Su grosor ronda los $0.125 [mm]$, con un error de $\pm 15\%$. El peso del material es de $100 \left[\frac{g}{m^2} \right]$ con un error de $\pm 10\%$ [91].

Black Conductive Bag ANT006BCB: Bolsa de polietileno de baja densidad (LDPE) con una capa de carbono de $0.1016 [mm]$ utilizado principalmente en aplicaciones antiestáticas, evitando la acumulación de cargas eléctricas [1]. Además, posee una resistencia superficial de $103 [\Omega]$ a $105 [\Omega]$, un período de atenuación de tensión estática menor o igual a $2 [s]$ y densidad de $0.92 [g/cm]$ [92].

Además, se utilizó una balanza de $5 [kg]$, distintos adhesivos, cables, cautín y pesos de metal de aproximadamente 12 gramos cada uno para aplicar presión en las pruebas realizadas al sensor.



Figura 3-21: a) Bolsa conductiva ANT006BCB [92], b) Tela conductora Shieldex NoraDell

3.3 Construcción de brazo y mano robótica

En esta sección se describe el entorno de trabajo para realizar la construcción del brazo y mano robótica, la configuración de impresión utilizada y el ensamble final de sus partes.

3.3.1 Entorno de trabajo

El software MakerBot (Figura 3-22) permite exportar los archivos STL, que define la geometría de la figura, a un archivo de tipo x3g. Este proceso permite entregar a la impresora 3D toda la configuración de impresión de la pieza.

La configuración utilizada para la impresión de las piezas es la opción predefinida, pero se editaron algunos parámetros para generar piezas de mayor densidad y calidad como se muestra en la Tabla 3-1.

Una vez que se ha exportado el archivo, se debe transferir a la impresora. Para el caso en particular se realiza la transferencia a través de una tarjeta SD, desde donde la impresora leerá el archivo .x3g y procederá a imprimir la pieza a producir capa por capa.

La impresora se puede observar en la Figura 3-23a, y en la Figura 3-23b se visualiza la impresión de una de las piezas del brazo robótico.

Esta operación se realizó para cada una de las piezas, pero se debe tener en cuenta que el proceso no está exento de inconvenientes, pues se debe tomar en consideración, por ejemplo, un margen de tolerancia debido al grosor del material de impresión, tener en cuenta el error en el desplazamiento de la boquilla o extrusor, su correcta calibración, niveles de filamento, entre otros parámetros.

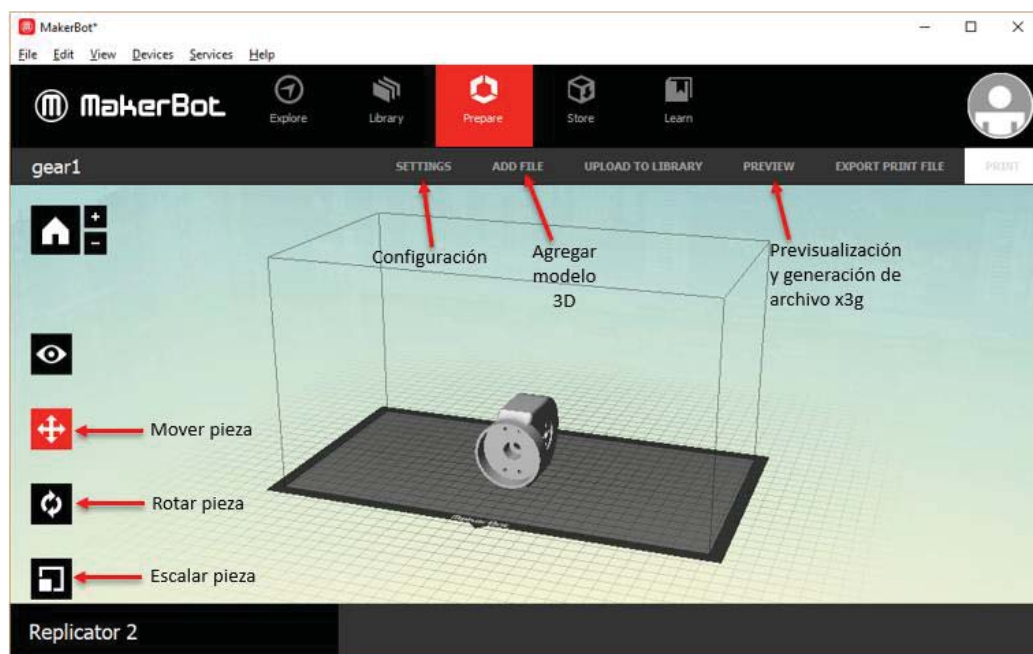
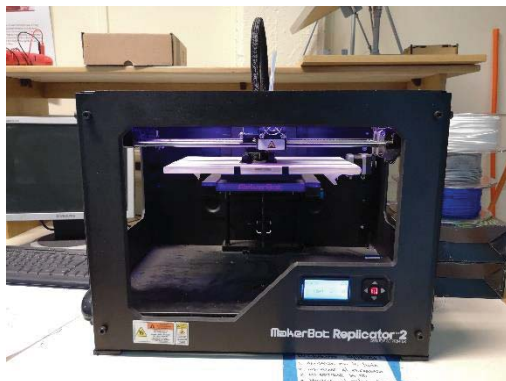


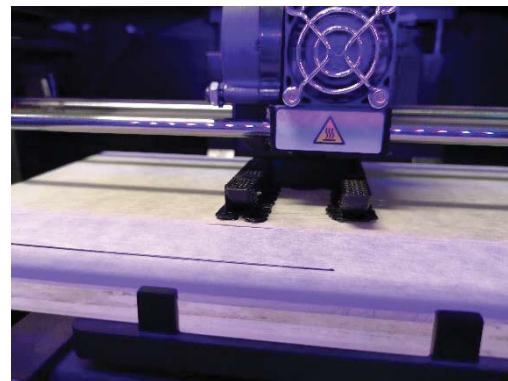
Figura 3-22: Entorno de trabajo de software *MakerBot*

Tabla 3-1: Configuración establecida en software *MakerBot* para impresión

Parámetro	Valor
Infill Density	40%
Infill Layer Height	0.2 [mm]
Support	Habilitado



a)



b)

Figura 3-23: a) Impresora MakerBot Replicator 2, b) impresión de abrazadera para actuador lineal

3.3.2 Ensamble

En este proceso se solicitó asistencia a personal del Taller de Electromecánica de la Facultad de Ingeniería Eléctrica con el fin de realizar mejoras de diseño, realizar la instalación y/o fabricación de piezas que no son posibles de crear en el Laboratorio de Robótica.

En primer lugar, se realizó el ensamble de la mano robótica (Figura 3-24). Para disminuir el roce producido por las uniones y articulaciones de los dedos, se tornearon los tornillos dejando su superficie lisa y dejando una sección con hilo para la fijación de las tuercas.

El ensamble de la muñeca y mano de la primera versión se muestra en la Figura 3-25. Se destacan algunos puntos importantes de diseño. Por ejemplo, si es necesario realizar el reemplazo del servomotor que se une al brazo y a la siguiente articulación, la extracción o instalación podría tomar varios minutos; además, esta estructura no posee un descanso de la pieza que produce la rotación de la mano, es decir, el peso de las tres piezas siguientes incluida el objeto que soporte la mano será la carga que deberá soportar el servomotor (rojo), acortando significativamente su vida útil.

Por lo antes descrito se determina realizar un rediseño e impresión de las nuevas piezas, las cuales se muestran junto al ensamble de la nueva versión de la muñeca y mano robótica en la Figura 3-26 y Figura 3-27.

La articulación del codo (Figura 3-28) muestra la instalación del *encoder* junto a los engranajes que informarán la posición del brazo robótico. Por otro lado, la articulación del hombro (Figura 3-29), como se mostró en el diseño inicial, se compone de varias piezas. Para su diseño se requirió la asistencia de personal del Taller Mecánico.



Figura 3-24: Ensamble de primera versión mano robótica



Figura 3-25: Ensamble de muñeca y mano primer diseño

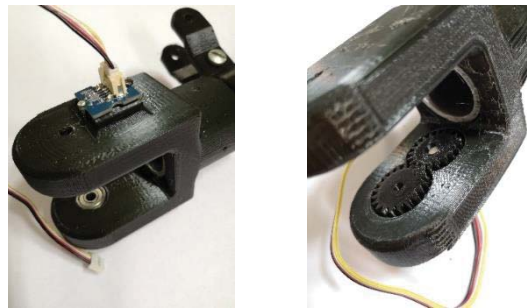


Figura 3-26: a) Mejora en la articulación montando un rodamiento para disminuir carga en el eje de servomotor, b) ensamble de la segunda versión de muñeca y mano



b)

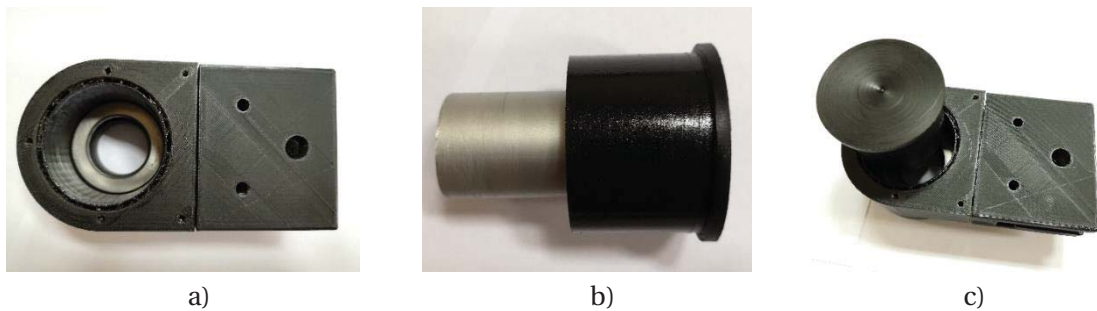
Figura 3-27: a) Mejora en la articulación montando un rodamiento para disminuir carga en el eje de servomotor, b) ensamble de la segunda versión de muñeca y mano



a)

b)

Figura 3-28: a) Instalación de *encoder* en articulación del codo, b) Sistema de engranajes para generar movimiento del *encoder* al mover el eje



a)

b)

c)

Figura 3-29: a) Vista superior de soporte brazo-plataforma, b) pieza de rotación de brazo, c) presentación de pieza de rotación y soporte

Los cortes de los tubos de aluminio se realizaron en el Taller Mecánico como se muestra en la Figura 3-30a con dimensiones de 25 [cm] cada uno y 1 pulgada $\frac{1}{4}$ de diámetro. Luego se ejecuta el ensamble del brazo y se presenta e instala en la plataforma (Figura 3-30b).

Como última actividad de la fase de construcción se realiza el cableado interno y programación de prueba de funcionamiento para verificar la operatividad de la estructura (Figura 3-30c y Figura 3-31).

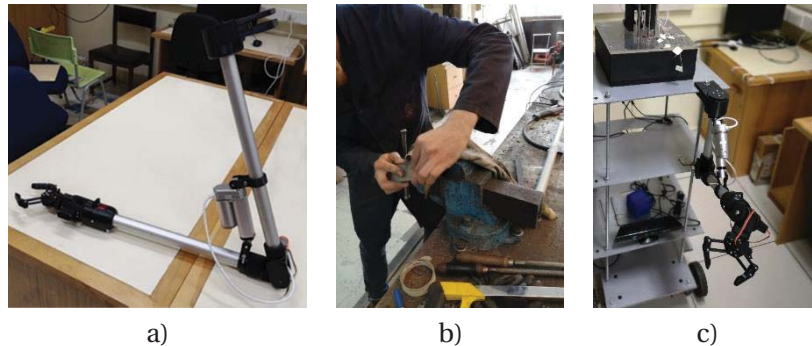


Figura 3-30: a) Presentación de piezas en tubos de aluminio para determinar dimensiones finales, b) Corte de los tubos de aluminio en Taller Electromecánico, c) presentación e instalación de brazo robótico en plataforma

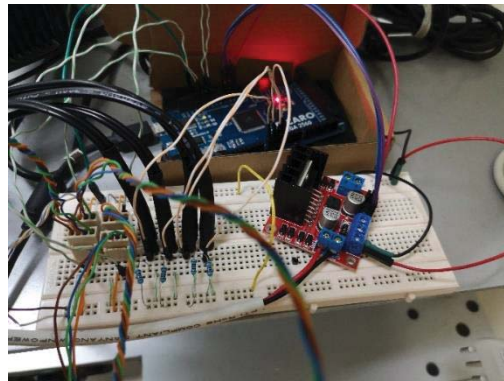


Figura 3-31: Circuito de control y lectura de sensores de brazo y mano robótica

3.4 Construcción de sensores de presión textil de tipo resistivo

La construcción de los sensores de presión se llevó a cabo realizando las siguientes acciones:

Se debe recortar la tela conductora con el tamaño y forma necesaria para su instalación en los dedos de la mano robótica, ésta misma acción se llevó a cabo en la bolsa semiconductor, procurando utilizar un excedente de $0.25[cm]$ de material por lado, con el fin de evitar cortocircuitos entre las dos capas conductoras (Figura 3-32). Es importante considerar el movimiento que realiza el manipulador del brazo robótico, de esta manera se puede determinar el lugar idóneo para la instalación de los sensores y obtener valores útiles a la hora de manipular objetos.

Se deben unir los cables conductores, para ello se habilitó previamente una sección en la tela conductora para este fin, soldándolos con estaño y un cautín. Cabe destacar que también pueden ser utilizados hilos de material conductor para realizar este paso (Figura 3-33a).

El siguiente proceso involucra la utilización de una capa de material piezo-resistivo, el cual es intercalado entre las dos capas conductoras como se muestra en la Figura 3-33b. En esta ocasión se realizaron algunas pruebas utilizando distintos tipos de adhesivo, con el fin de determinar posibles mejoras en la respuesta del sensor. Dentro de los adhesivos utilizados se encuentran: Scotch, Agorex, silicona, y cinta doble faz, dando mejores resultados la cinta adhesiva Scotch.

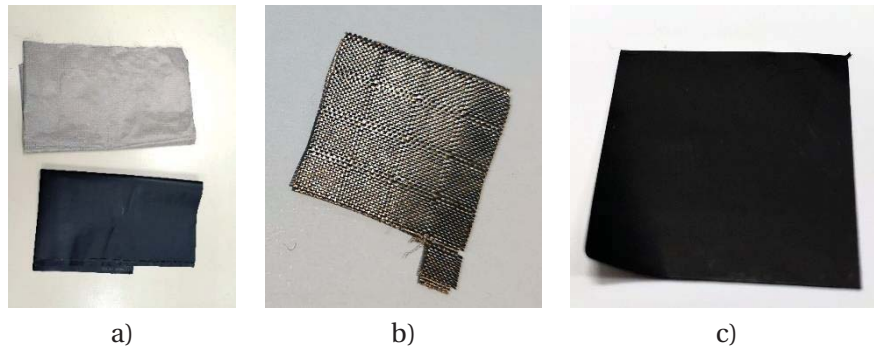


Figura 3-32: a) Material conductor textil en la sección superior y bolsa semiconductor en la sección inferior, b) Material conductor recortado en las dimensiones y forma deseada, c) Bolsa de polietileno de baja densidad cortada en las dimensiones deseadas

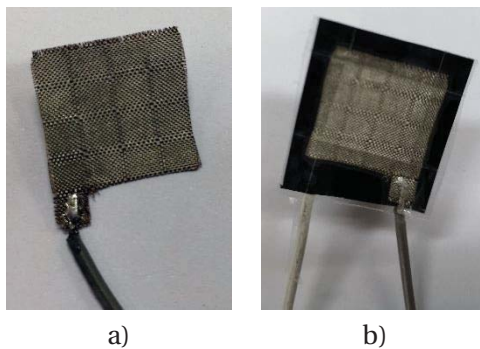


Figura 3-33: a) Unión mediante soldadura con estaño a tela conductora, b) Sensor terminado, con unión de cinta adhesiva

3.5 Caracterización de sensor de presión textil de tipo resistivo

Una vez construido el sensor de presión, es necesario evaluar su comportamiento. Para ello se siguen los pasos realizados en el artículo *Easy-to-Build Textile Pressure Sensor* [1].

Como ya se han determinado las dimensiones de los sensores, se realiza por medio de un software de diseño 3D un recipiente con una porción adicional en la parte inferior con la misma área de la zona activa del sensor (Figura 3-34).

Posteriormente, se realiza un circuito para leer la información que proporciona el sensor en el microcontrolador Arduino. Como los valores de resistencia en el sensor podrían ser muy bajos, podría ocurrir un cortocircuito, por ese motivo se recomienda utilizar una resistencia que limite la corriente a través del sensor. En esta ocasión se ha seleccionado una resistencia de $1[k\Omega]$ teóricos (Figura 3-35).

Para adquirir los datos desde la placa arduino, se utilizó un programa generado en lenguaje VisualBasic denominado PLX-DAQ en su versión 2.11, la cual consiste en una planilla de Microsoft Excel habilitada para macros que, a través de conexión serial, recibe los mensajes enviados por el microcontrolador (Figura 3-36).

Para caracterizar el sensor, se conectan los terminales de este a un multímetro para leer las variaciones de resistencia. Se llena el contenedor con el peso deseado y se ubica sobre el sensor, cubriendo la totalidad del área activa como se muestra en Figura 3-37. Luego, se aumenta el peso y se vuelve a tomar la medición. Para el caso en particular se tomaron intervalos de 30 segundos por cada una de ellas.

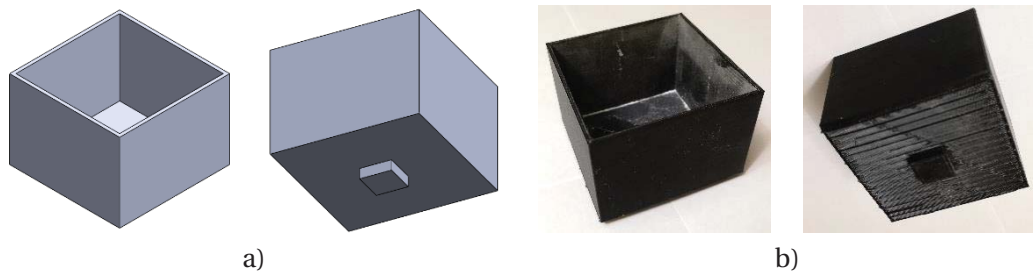


Figura 3-34: a) Diseño 3D de recipiente (vista isométrica), b) Recipiente impreso (vista superior)

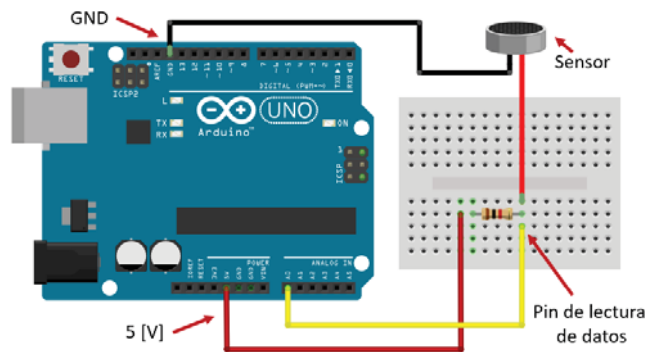


Figura 3-35: Esquema del circuito generado para lectura de datos del sensor

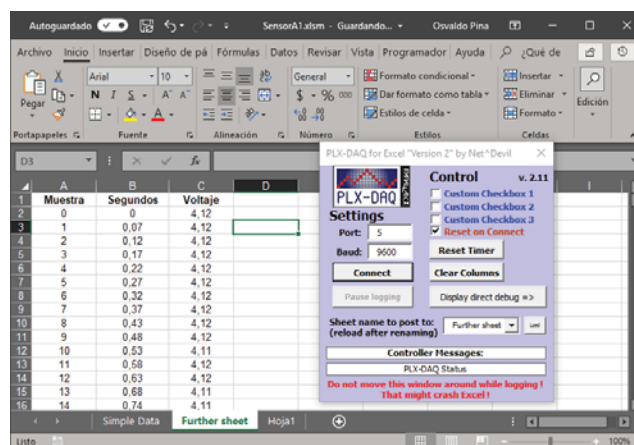


Figura 3-36: Interfaz gráfica PLX-DAQ y plantilla Microsoft Excel

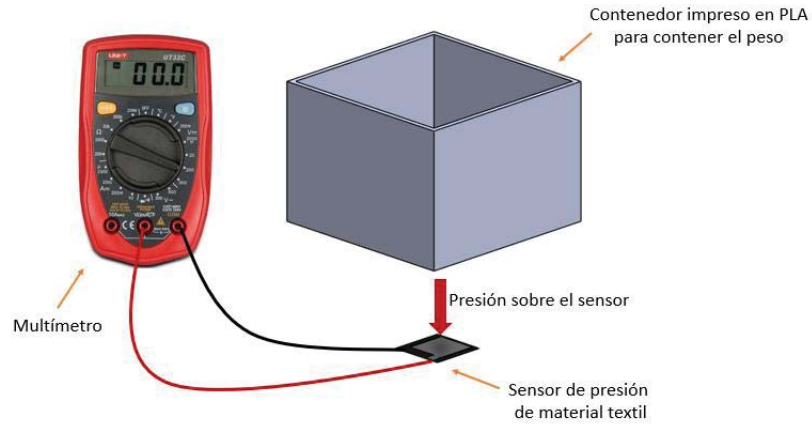


Figura 3-37: Configuración utilizada para la medición de la variación de resistencia del sensor a medida que cambia la presión

3.6 Ensamble de brazo terminado en plataforma móvil de robot de servicio

La Figura 3-38 muestra el acople de los componentes del robot de servicio, el cual se compone de una plataforma móvil, una cabeza robótica con IA y el brazo robótico con sensores de presión instalados en los dedos de la mano.

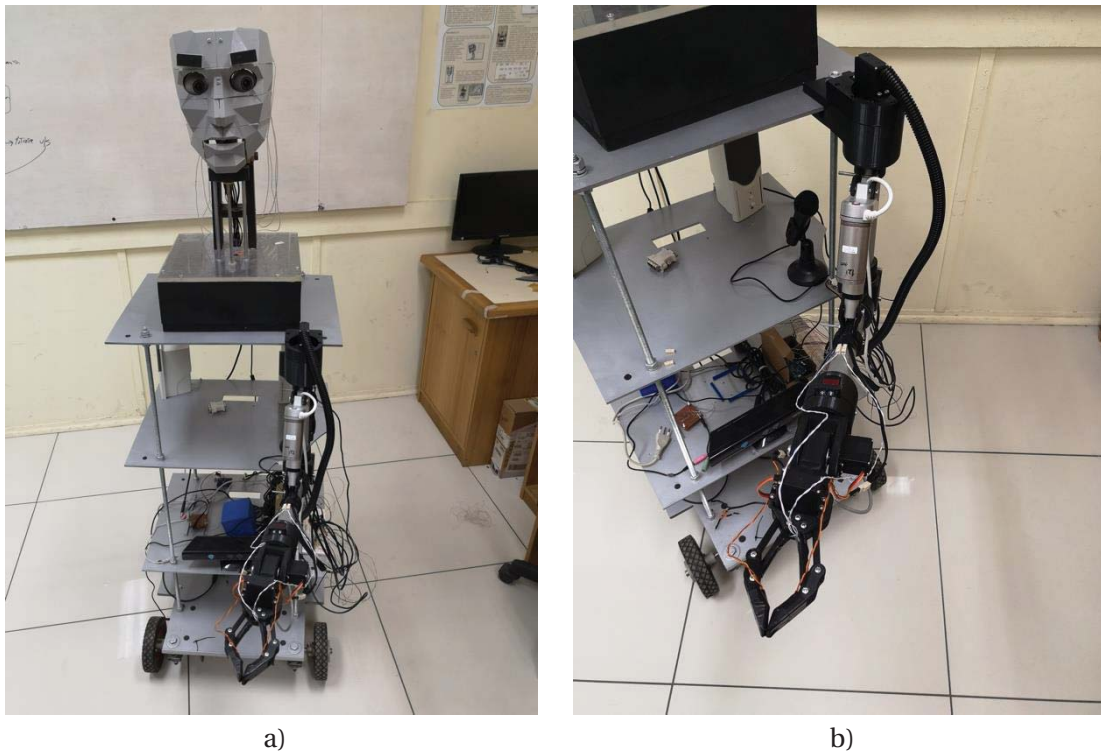


Figura 3-38: a) Robot de servicio ensamblado, b) Brazo robótico finalizado (con sensores de presión incorporados) instalado en plataforma móvil

3.7 Diagrama de flujo para realización de pruebas

Para realizar pruebas con el brazo robótico, se generó un script con movimientos preestablecidos, con el objetivo de determinar el comportamiento en un ambiente real y determinar su correcta operación.

El script comienza definiendo todas las variables y configuraciones a utilizar, luego ingresa a la subrutina *Pos_inicial()*, que tiene como objetivo posicionar el brazo robótico en su estado de reposo. Cuando el usuario selecciona la rutina (dependiendo del objeto a manipular) en la subrutina *Pos_accion()*, el brazo se moverá al punto definido en dicha rutina. Al momento de realizar la manipulación del objeto, se consulta por una condición dentro de la subrutina *sensores()*, el cual compara la lectura del sensor con un umbral establecido, si la tensión en los terminales del sensor supera dicho umbral, el *gripper* se cierra en 10 grados, esto se repetirá hasta que no se cumpla la condición. Luego, la rutina que se había seleccionado finalizará posicionando el objeto en el punto de destino y luego se ejecutará nuevamente la subrutina *Pos_inicial()*, cerrando el *loop*, como se muestra en la Figura 3-39.

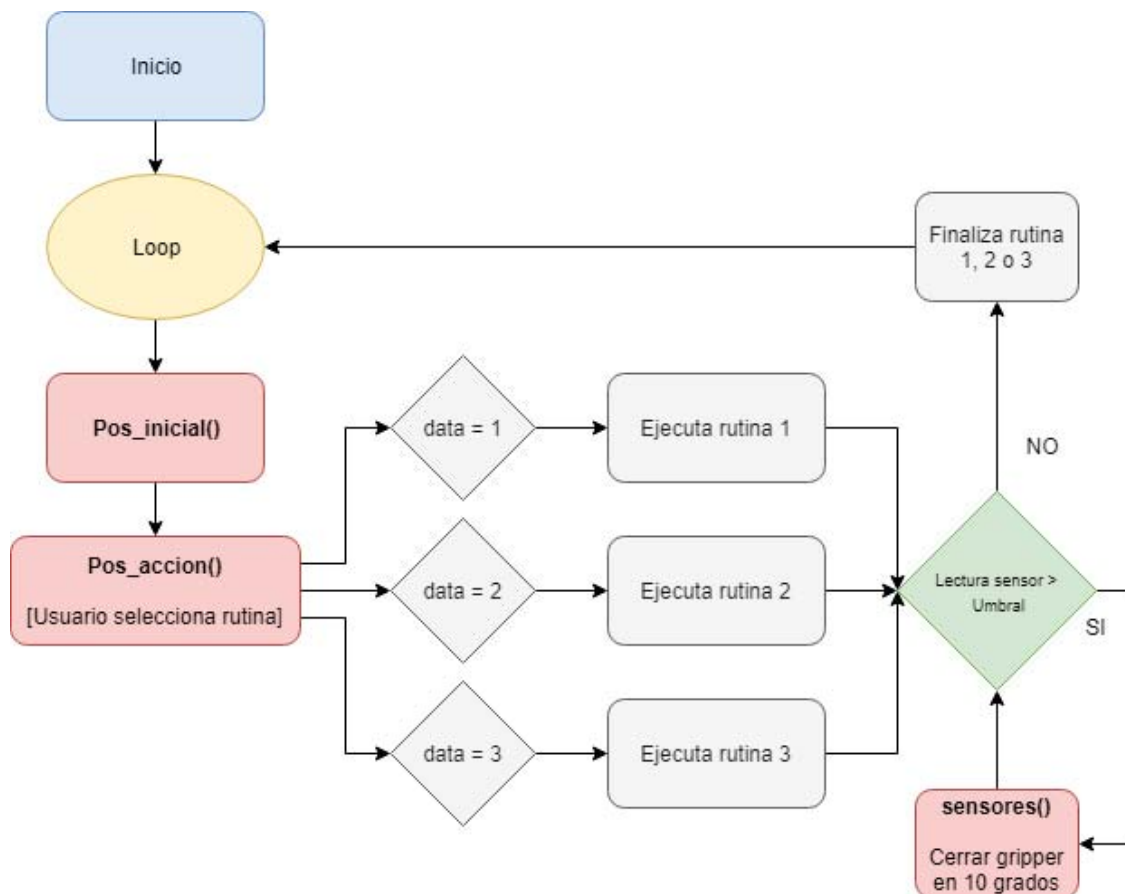


Figura 3-39: Diagrama de flujo de script controlador del brazo

3.8 Creación de tópicos y nodos para visión por computador

Para realizar las pruebas se implementó una red SSD Mobilenet pre-entrenada. Utilizando una de las cámaras web de la cabeza robótica, se obtiene imagen mediante un nodo ROS denominado *cv_camera*, el cual ocupa la librería *OpenCV* y *Python*, permitiendo la configuración del dispositivo a utilizar.

Una vez el nodo publica la imagen mediante el *topic cv_camera/Image_raw*, un *script* de Python se suscribe a dicho *topic*, adquiriendo la imagen y procesándola a través de la red neuronal utilizada. como los pesos de la red neuronal convolucional SSD ya se encuentran pre-entrenados, solo se cargan los datos y se ejecuta el nodo. Éste, a su vez, publica mediante el *topic /coordenada* el centro del rectángulo donde se encuentra etiquetado el objeto a detectar. Para el caso en particular solo se realizaron pruebas con objetos de clase *bottle*.



Figura 3-40: Nodos activos para visualización de datos.

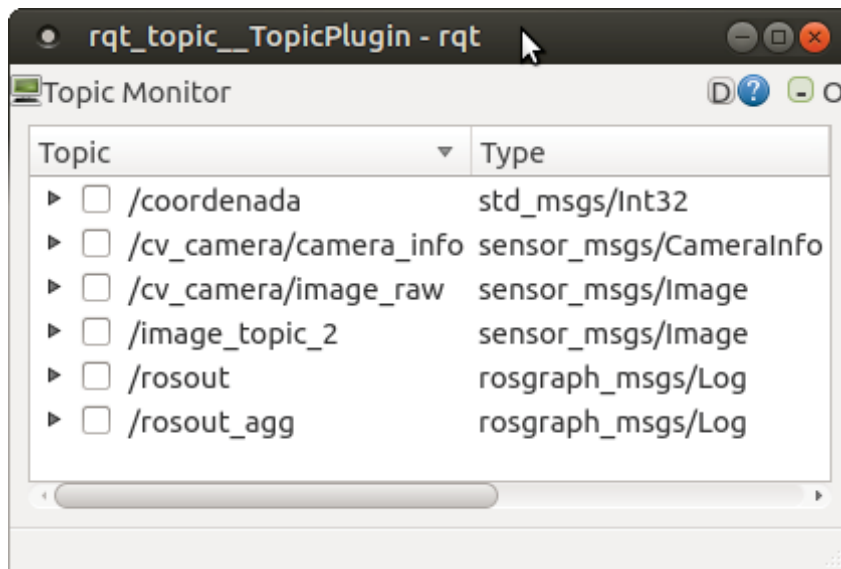


Figura 3-41: *Topics* activos al momento de ejecutar el nodo *cv_camera* y el script de procesamiento, detección y reconocimiento de objetos

4 Resultados y análisis

En esta sección se muestran los resultados obtenidos de las pruebas realizadas para validar el funcionamiento de los sensores de presión, visión por computador y la generación de pruebas de manipulación.

4.1.1 Pruebas de sensores de presión

Para determinar el comportamiento de los sensores se realizó la operación mencionada en la sección 3.5. Ésta consiste en obtener la variación de la resistencia eléctrica a medida que se aplica mayor presión en el área útil de los sensores. Las presiones a las que fueron sometidos los sensores de sección cuadrada utilizados en la yema de los dedos de la mano robótica varían entre 1.39 [kPa] y 95.60 [kPa] para el Sensor 1 y entre 1.96 [kPa] y 66.41 [kPa] para el Sensor 2. La variación de resistencia para los valores de presión antes mencionados se encuentran entre 1320 [Ω] y 32 [Ω] para el *Sensor 1* y entre 1060 [Ω] y 27 [Ω] para el *Sensor 2*. Lo anterior se observa en las curvas presentes en la Figura 4-1, donde los sensores de sección cuadrada muestran comportamientos similares. De acuerdo a los resultados obtenidos, los sensores, al presentar morfologías y dimensiones similares se podría estimar su respuesta.

Al observar la Figura 4-2, correspondiente a las curvas características de los sensores de sección rectangular, se observa que las curvas difieren levemente, obteniéndose resultados de presión y resistencias que van desde los 3.04 [kPa] y 448 [Ω] hasta 78.92 [kPa] y 30 [Ω] para el *Sensor 3*, y desde 2.99 [kPa] y 498 [Ω] hasta 73.04 [kPa] y 26 [Ω] para el *Sensor 4*. Al igual que en el caso anterior, su respuesta tiene directa relación con la morfología y dimensiones de los sensores.

En [1] se señala la importancia de comparar y modelar los sensores construidos, con el fin de determinar su comportamiento y principales diferencias entre sí. Para ello, se utiliza la expresión (4-1), donde R representa la resistencia resultante en [$k\Omega$], P es la presión aplicada en [kPa], m es la pendiente y b es la intersección de la función lineal.

$$R = b + m \cdot \log(P) \quad (4-1)$$

Al analizar las gráficas de forma general, se observa un cambio en la pendiente al llegar a una presión aproximada de 10 [kPa]. Es así como se puede establecer una función por partes, que se establecen para los valores de presión que van desde 1 [kPa] a 10 [kPa] y los valores de presión

sobre 10 [kPa] hasta 100 [kPa]. Los valores para cada uno de los sensores se muestran en la Tabla 4-1 y Tabla 4-2 de acuerdo a la región en que se encuentren (menores o iguales a 10 [kPa] y mayores a 10 [kPa]).

Los ensayos se realizaron utilizando presiones desde 0 [kPa] a 95.6 [kPa]

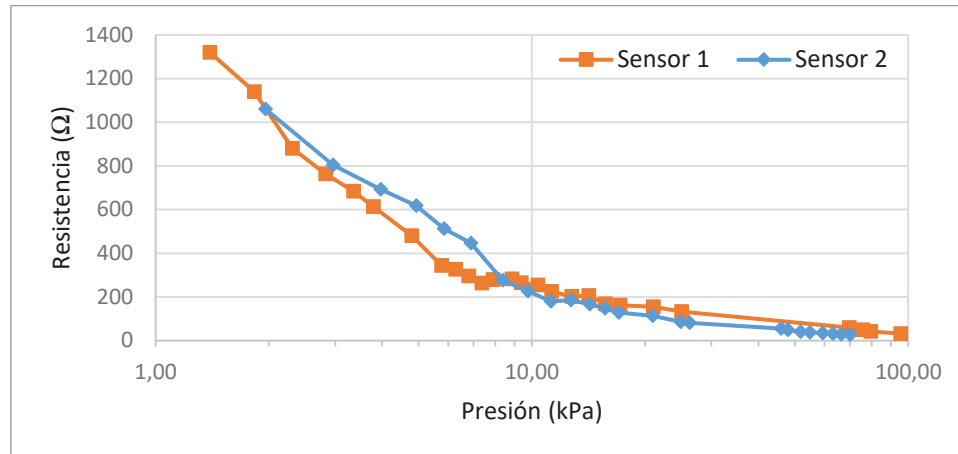


Figura 4-1: Gráfica del comportamiento de los sensores de sección cuadrada al ser sometidos a presión

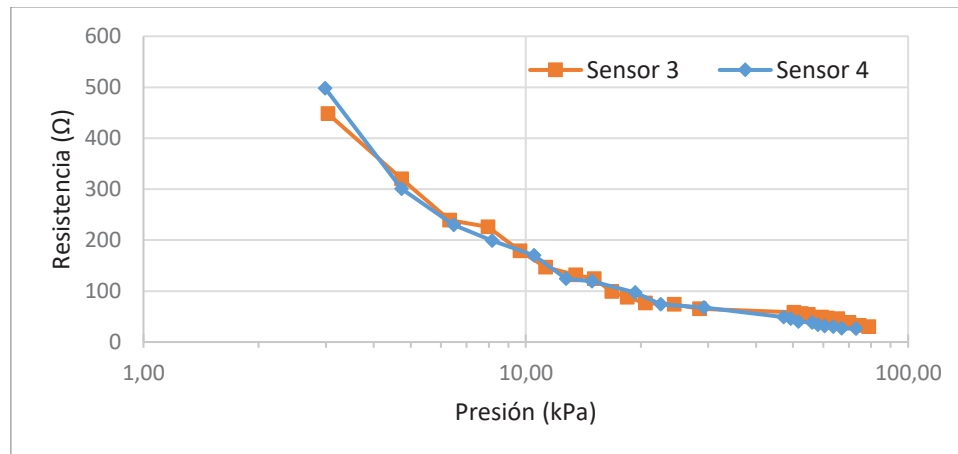


Figura 4-2: Gráfica del comportamiento de los sensores de sección rectangular al ser sometidos a presión

Tabla 4-1: Parámetros de ajuste de cada sensor para valores de presión menores o iguales a 10 [kPa]

	Intercepto	Pendiente
Sensor 1	1377,8	-536,7
Sensor 2	1395	-508,4
Sensor 3	687,1	-228,6
Sensor 4	743,08	-258,4

Tabla 4-2: Parámetros de ajuste de cada sensor para valores de presión superiores a 10 [kPa]

	Intercepto	Pendiente
Sensor 1	1377,8	-536,7
Sensor 2	1395	-508,4
Sensor 3	687,1	-228,6
Sensor 4	743,08	-258,4

Posteriormente se instalaron los sensores en la mano robótica con cinta adhesiva, con el objetivo de realizar pruebas de agarre de diferentes objetos y determinar el comportamiento de los cuatro sensores tal como se muestra en la Figura 4-3.

En la Figura 4-4, se observan las seis pruebas realizadas para medir la variación de la tensión al tomar distintos objetos. En la Figura 4-4a, se cierran los dedos de la mano robótica sin que ésta tome un objeto. Los sensores de la yema de los dedos son los que presentan mayor perturbación, como se muestra en la Figura 4-5. En la prueba que se muestra en la Figura 4-4b, el comportamiento mostrado en la gráfica (Figura 4-6), es similar a la anterior debido a que la libreta no interactúa con los sensores que se encuentran en la parte interna de los dedos. Para la prueba realizada en la Figura 4-4c existe perturbación solo en el *Sensor 1* y *Sensor 2*. Su respuesta se observa en la Figura 4-7, donde el tarro de café solo es manipulado por la yema de los dedos. En la siguiente prueba (Figura 4-4d), se observa que el cilindro de film plástico es manipulado por la parte interna de los dedos, esto produce una disminución de la resistencia del *Sensor 3* y *Sensor 4* debido al efecto piezo-resistivo, producto de la naturaleza de los materiales con los que fueron fabricados, dando como resultado la respuesta que se muestra en la Figura 4-8. La Figura 4-4e y su gráfica en la Figura 4-9, tienen un comportamiento similar al mostrado en Figura 4-8, pero con una mayor participación del *Sensor 2*. Finalmente, para el experimento Figura 4-4f, se repite el mismo comportamiento que en Figura 4-4b, dado que solo son accionados el *Sensor 1* y el *Sensor 2*.

Si se analiza el comportamiento de los sensores cuando no se aplica ningún tipo de presión, se observa que sus valores son relativamente estables, manteniendo el voltaje en valores constantes. Considerando la Figura 4-11, existe una perturbación alrededor de los 5 segundos debido a la apertura de los dedos de la mano. A los 11 segundos se genera el cierre de los dedos, donde ocurre el agarre de un objeto. En este caso no existe manipulación de objetos, y solo existe contacto entre las yemas de los dedos, por ese motivo los sensores 1 y 2 sufren una perturbación mayor. Finalmente, entre los 25 y 30 segundos los dedos de la mano se abren, y el sensor regresa a su estado de reposo. Se debe considerar, además, que existe un tiempo de recuperación de aproximadamente 15 segundos para los sensores involucrados en la acción, pero que no impactaría en el desempeño del brazo robótico ya que el tiempo que demora en completar una rutina es mayor a dicho tiempo de recuperación.

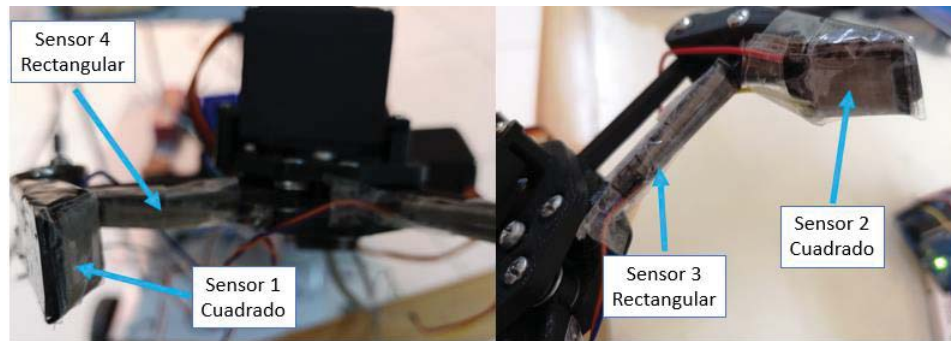


Figura 4-3: Posición de los sensores en los dedos de la mano robótica

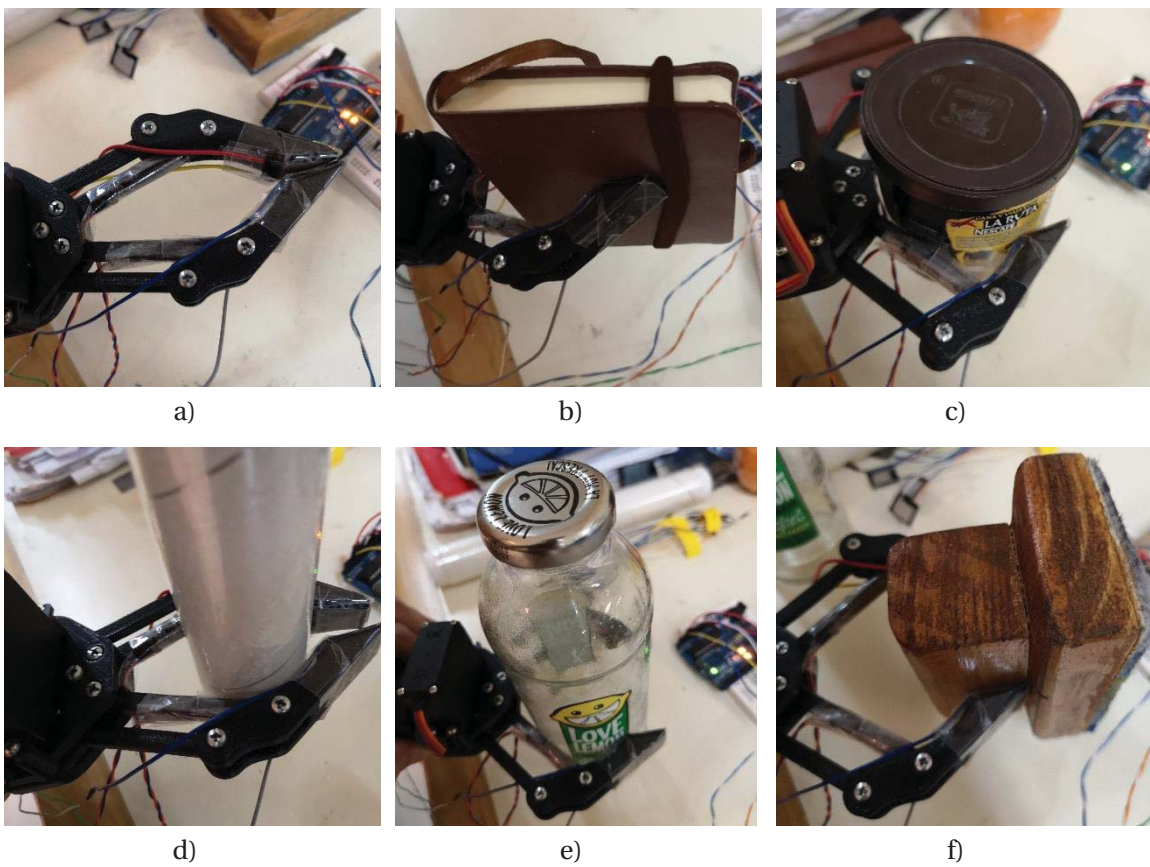


Figura 4-4: Presión de los dedos de la mano robótica dedo contra a) dedo contrario, b) libreta de notas, c) tarro de café, d) Tubo de film plástico, e) botella de vidrio, f) borrador de madera

A continuación, se muestran las gráficas de las seis pruebas realizadas:

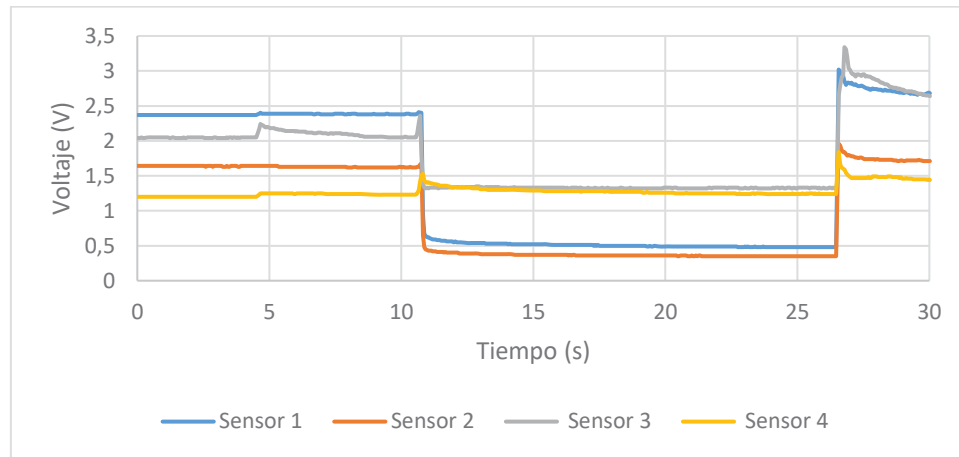


Figura 4-5: Gráfica de la variación de tensión en el tiempo cuando la mano se encuentra vacía

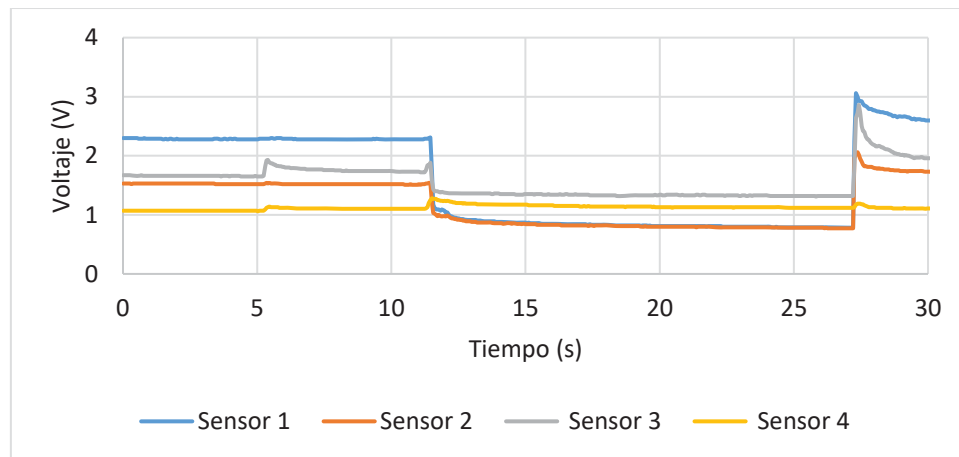


Figura 4-6: Gráfica de la variación de tensión en el tiempo cuando la mano sostiene una libreta de notas

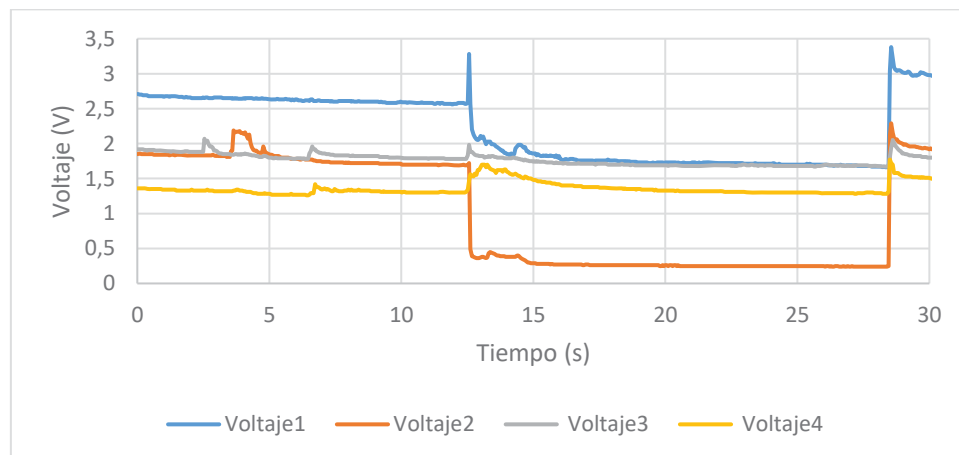


Figura 4-7: Gráfica de la variación de tensión en el tiempo cuando la mano sostiene un tarro de café

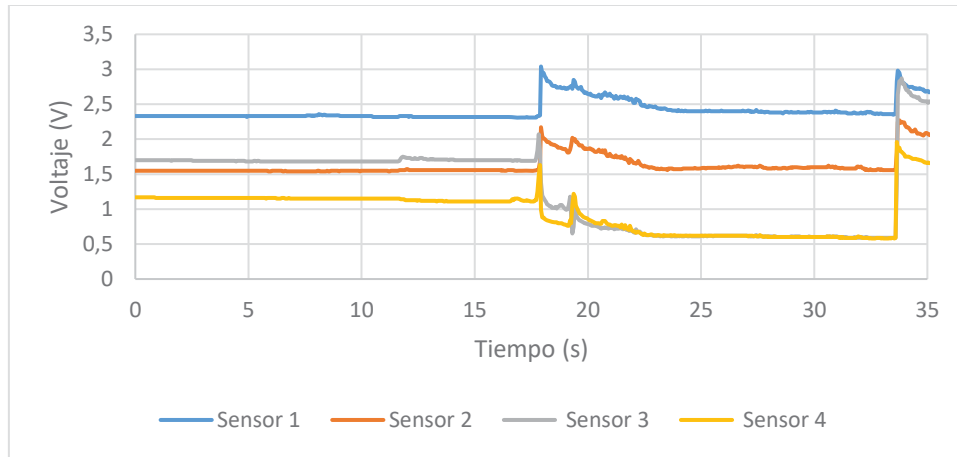


Figura 4-8: Gráfica de la variación de tensión en el tiempo cuando la mano sostiene un tubo de film plástico

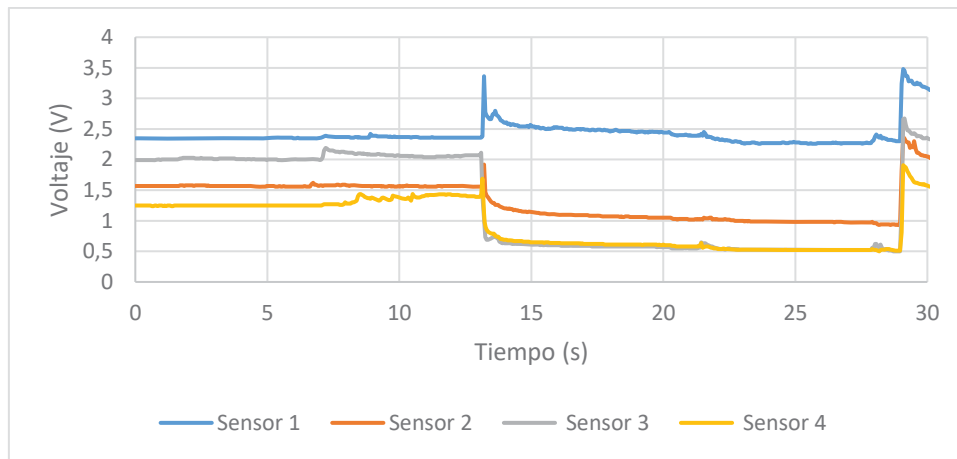


Figura 4-9: Gráfica de la variación de tensión en el tiempo cuando la mano sostiene una botella de vidrio

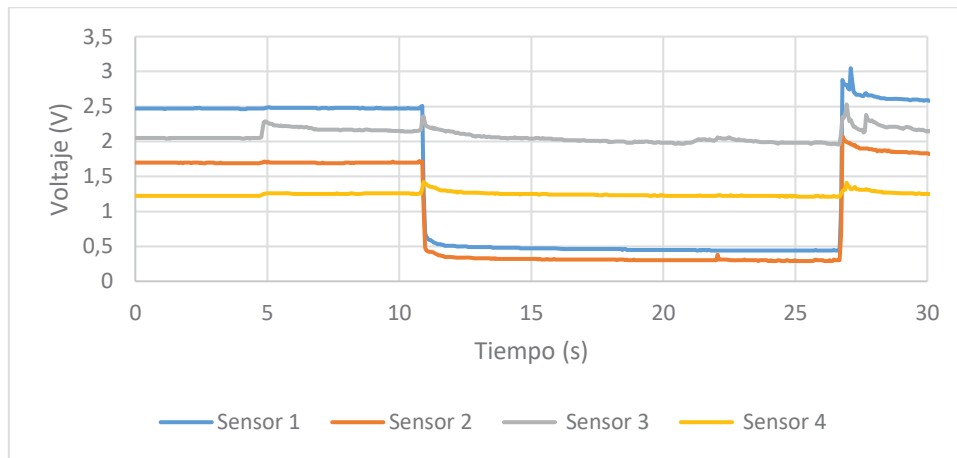


Figura 4-10: Gráfica de la variación de tensión en el tiempo cuando la mano sostiene un borrador de madera

En la gráfica de la Figura 4-11 se indican las zonas de interés en el comportamiento de los sensores de presión cuando existe movimiento de la mano y brazo robótica.

La secuencia de funcionamiento de los sensores mostrados en la Figura 4-12, el cual se basa en el script que se muestra en el Listado 2-1, y que también es explicado en la Figura 3-39 realiza la siguiente tarea: Se muestra un mensaje inicial en el momento en que se inicia el movimiento del brazo y mano, hasta que se ha ubicado en la posición deseada. De manera predeterminada, los dedos de la mano robótica se encuentran abiertos y se procede a realizar la lectura inicial de los sensores. Si el sensor no ha llegado al umbral deseado, los dedos se cerrarán 10 grados y se vuelve a medir el valor del sensor. Una vez alcanzado el valor establecido para el objeto que se desea mover, el brazo robótico comenzará a moverse y posicionará al objeto en el lugar deseado, mostrando el mensaje *Acción finalizada* para luego volver a su posición de reposo.

En el ejemplo (Figura 4-12), se observa que los sensores oscilan entre 4.02 Volts y 5,00 Volts al no existir presión entre los dedos de la mano robótica con algún objeto. Por tratarse solo de una prueba de funcionamiento, se considera solo la variación de voltaje del *Sensor 1*, el cual es perturbado al momento de manipular un objeto. En las 4 primeras mediciones realizadas se observa una oscilación en torno a los 4.50 Volts. Cuando se incrementa la presión, es decir, cuando la apertura de los dedos de la mano robótica disminuye, el voltaje del sensor comienza a mostrar una variación significativa en torno a 1.99 y 1.33 Volts, en ese momento existe un contacto importante entre los dedos y el objeto. Esto continua hasta alcanzar el umbral previamente configurado, lo cual permite un agarre firme y, por consiguiente, el traslado estable del objeto de un punto a otro del espacio de trabajo.

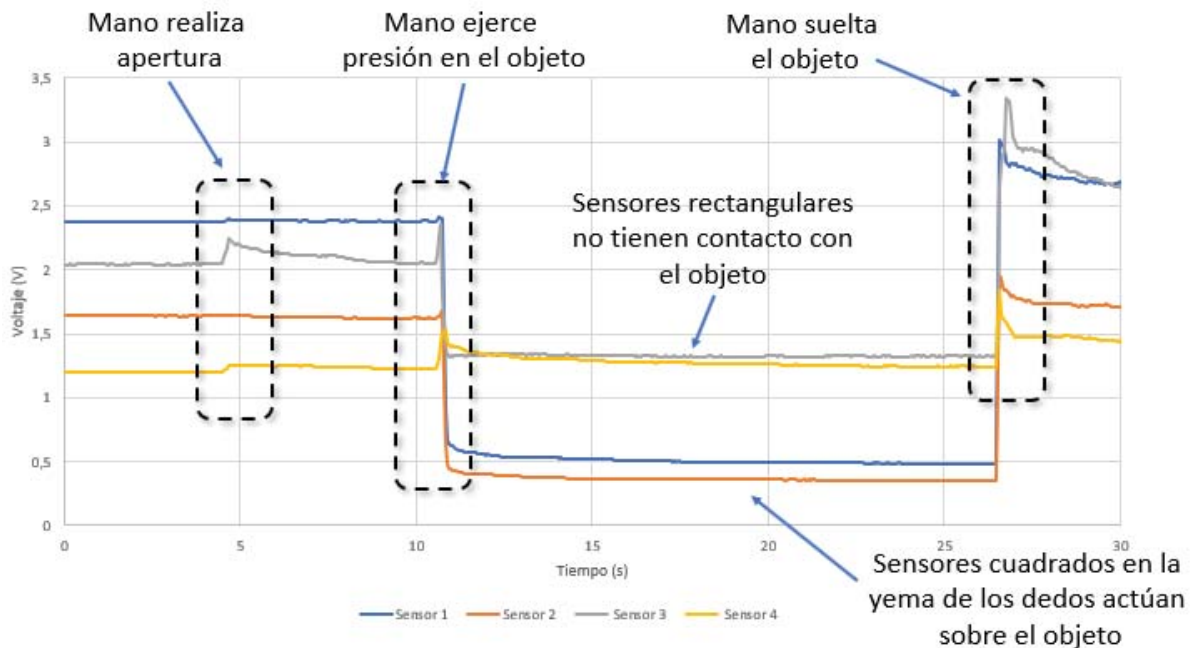


Figura 4-11: Análisis de comportamiento de la señal del sensor al manipular un objeto

```

COM4
...Iniciando Acción...
...Leyendo sensores...
Sensor1 : 4.80
Sensor2 : 5.00
Sensor3 : 5.00
Sensor4 : 4.02
...Cerrando gripper...
Sensor1 : 4.78Volts
Sensor1 : 4.40Volts
Sensor1 : 4.67Volts
Sensor1 : 4.44Volts
Sensor1 : 1.91Volts
Sensor1 : 1.33Volts
...Umbral alcanzado...
...Moviendo...
...Acción finalizada...

```

Autoscroll Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

Figura 4-12: Ejecución de script y lectura de tensión de Sensor 1

4.1.2 Pruebas de visión por computador

Como se mencionó en el apartado 2.14, para disminuir el consumo de recursos computacionales se seleccionó una red pre-entrenada denominada *Mobilenet SSD (Single Shot Multibox Detector)*. Una vez que ha sido instalada y configurada en Linux (Ubuntu 18.04 LTS.), y utilizando una de las cámaras de la cabeza robótica, se ejecuta el script de reconocimiento utilizando ROS.

Se realiza la primera prueba, la cual se enfoca en reconocer objetos no estáticos. La precisión en el reconocimiento del objeto de interés se encuentra en torno a un 48% y, en ocasiones, no se logra detectar el objeto (Figura 4-13). Sin embargo, existe un alto índice de reconocimiento de personas, lo cual puede ser útil para que el robot de servicio reconozca la presencia de individuos en el laboratorio y asimismo complementar el sistema de reconocimiento de rostros que posee la cabeza robótica en la actualidad.

En la siguiente prueba, la cabeza robótica se encuentra mirando el espacio de trabajo en donde se ubican dos objetos de interés. Dichos objetos se encuentran estáticos en una ubicación determinada, logrando una precisión de reconocimiento de 92.37% y 98.26% respectivamente (Figura 4-14), esto resulta relevante dado que la manipulación de objetos en la puesta en marcha del robot de servicio será con objetos estáticos en los muebles del laboratorio. Se debe considerar también, que las pruebas solo fueron realizadas con dos clases (Botella y Persona). Al tratarse de una red pre-entrenada, los objetos a manipular en el laboratorio de Robótica, en su mayoría, no coinciden con los objetos por defecto de la red *Mobilenet SSD*. No obstante, la red mostró buenos resultados, por lo que se puede evaluar realizar nuevas pruebas, entrenando previamente la red con una base de datos propia.

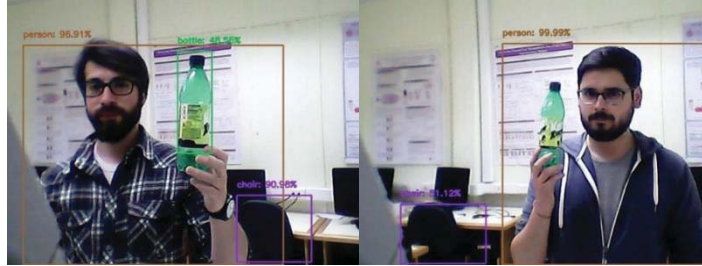


Figura 4-13: Pruebas de reconocimiento de objetos y personas



Figura 4-14: Robot “CJ” visualizando y reconociendo dos objetos de interés

4.1.3 Pruebas de manipulación brazo y mano robótica

Los objetos de interés para las pruebas de manipulación fueron una botella de bebida vacía, una botella de bebida con líquido, un control remoto y un envase de forma cilíndrica. Las pruebas consistieron en mover los objetos mencionados de un punto a otro dentro de un área de trabajo, el cual corresponde a la superficie de una de las mesas del laboratorio de robótica, lugar donde se desenvolverá el robot de servicio.

La plataforma móvil fue ubicada a 33 [cm] de la mesa. El brazo robótico cuenta con 4 grados de libertad (DoF), lo cual restringe el área de trabajo a un radio de alcance fijo de 49 [cm]. Lo anterior se debe a que la plataforma móvil le da la posibilidad al brazo robótico de moverse en dirección frontal y posterior, función que al momento de realizar las pruebas no se encontraban operativa. El detalle del brazo robótico y sus movimientos se pueden visualizar en la Figura C-5 del apéndice **Error! No se encuentra el origen de la referencia.C.**

En la Figura 4-15, se observa la manipulación de una botella de bebida vacía (PET), cuya masa no supera los 9 gramos. Si bien el peso ejercido en el brazo robótico no es significativo, una botella de estas características tiende a ser inestable al momento de posicionarla en algún punto del área de trabajo, acción que el brazo robótico logra llevar a cabo correctamente. En la prueba siguiente (Figura 4-16) se utiliza la misma botella, pero se le agrega líquido (agua), sumando una masa de 300 gramos. Al realizar la manipulación de ésta, se obtiene el mismo resultado anterior. Para la

prueba de la Figura 4-17 se utiliza un control remoto de masa similar a la botella con líquido, pero con una morfología más compleja de manipular. La acción se lleva a cabo sin inconvenientes. Por último, en la Figura 4-18 se manipula un envase que en su interior tiene elementos que otorgan una mayor masa al objeto (500 gramos), logrando realizar la acción correctamente, por lo tanto los cálculos de torque realizado en el apartado 3.1.5 en relación a la carga teórica es coherente con las pruebas y pesos reales utilizados.

Prueba de manipulación brazo robótico (botella vacía)

Como se indicó anteriormente, la plataforma se ubica a una distancia de 33 [cm]. En la siguiente secuencia se observa el proceso de manipulación de una botella vacía. La mano se adapta correctamente a la forma del objeto y el posicionamiento en el lugar deseado es estable.

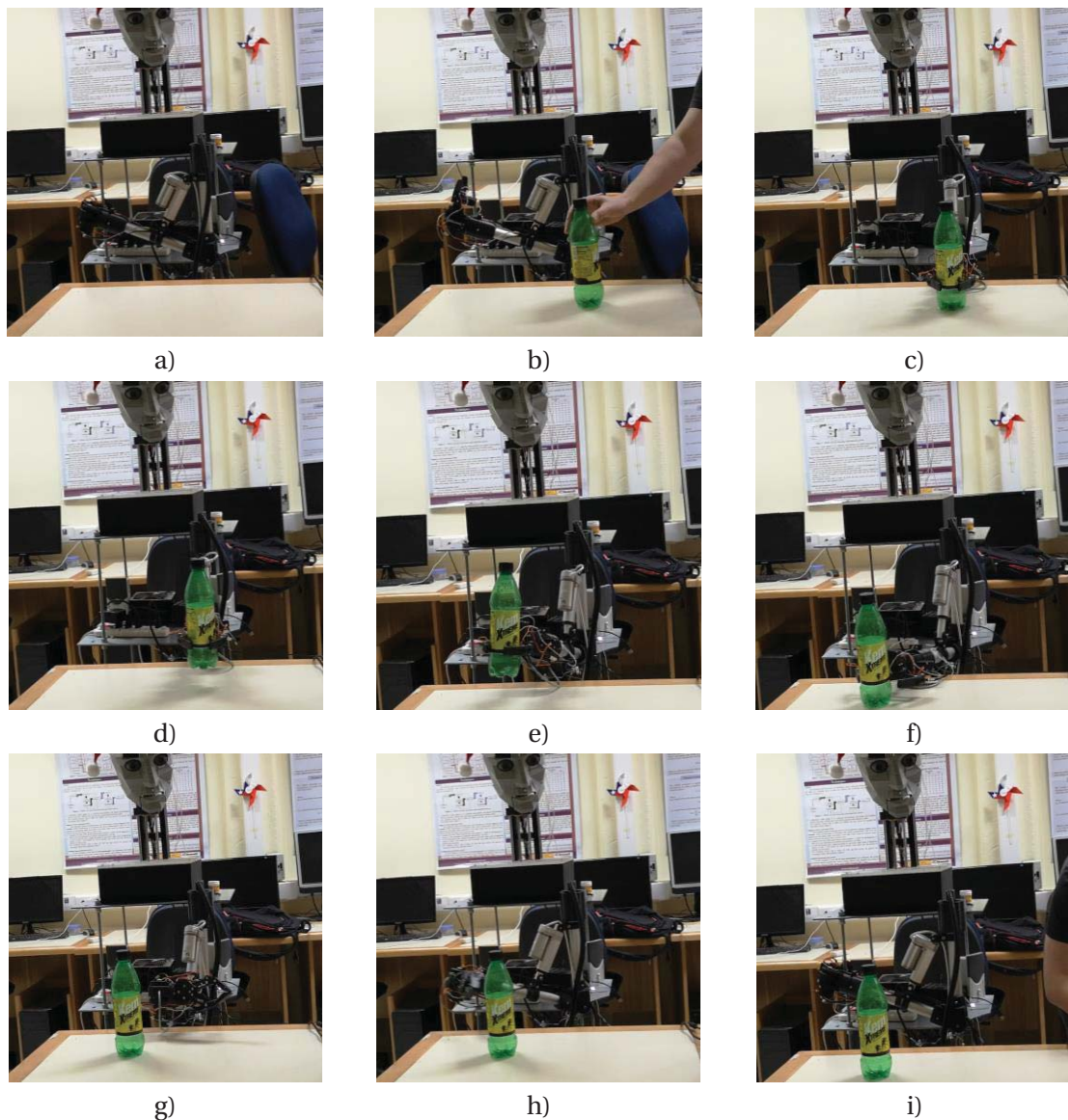


Figura 4-15: Secuencia de prueba de agarre y desplazamiento de brazo robótico

Prueba de manipulación brazo robótico (botella con liquido)

Al igual que en el caso anterior, se utiliza una botella de bebida a la cual se le introdujo agua para aumentar su peso. El agarre del objeto es correcto, demorando 59.2 segundos en completar toda la acción, desde que es activado el brazo hasta que vuelve a su posición de reposo.

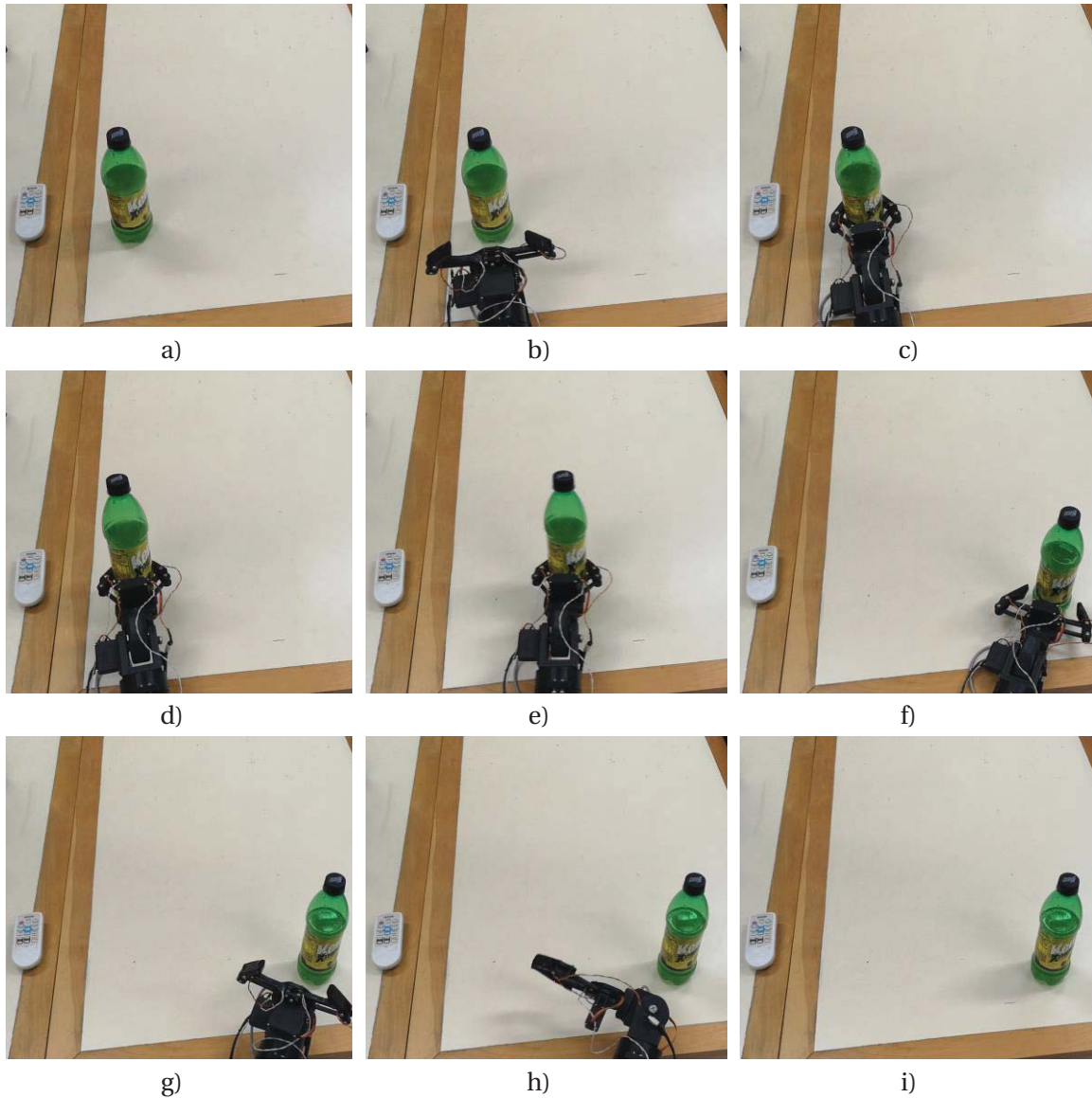


Figura 4-16: Secuencia de prueba de agarre y desplazamiento de brazo robótico (vista desde cabeza robótica)

Prueba de manipulación brazo robótico (control remoto)

En este caso, resulta más complejo realizar la manipulación de este objeto debido a su morfología, además de presentar bordes lisos. Para lograr un buen agarre, las yemas de los dedos deben tener mayor fricción y se debe ejercer una mayor presión en las paredes del objeto. El tiempo de ejecución del script que mueve el control remoto o elementos que poseen similares dimensiones es de 45 segundos en total, desde el momento de la activación hasta el estado de reposo del brazo robótico.

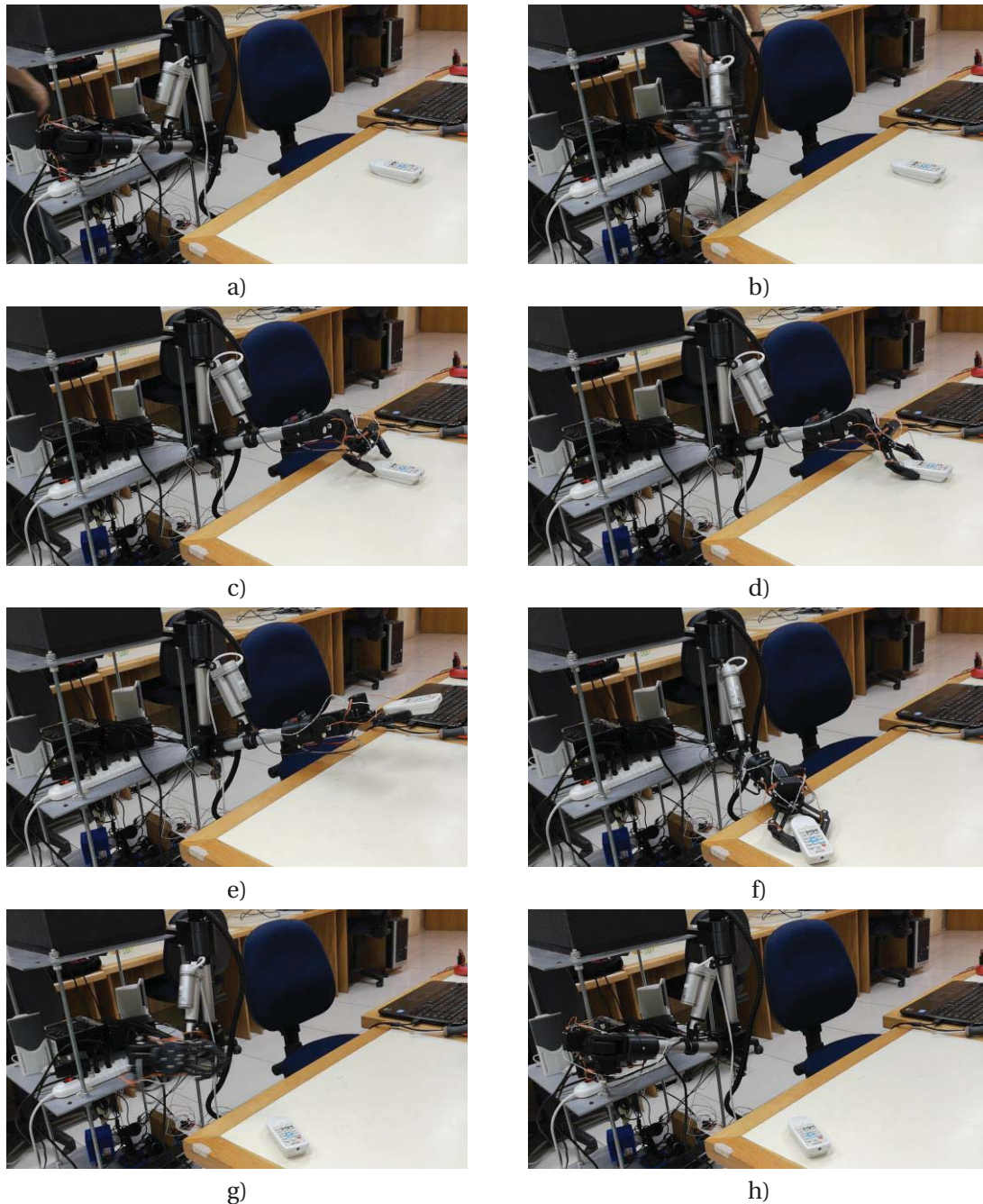


Figura 4-17: Secuencia de prueba de agarre y desplazamiento de brazo robótico de objeto control remoto

Prueba de manipulación brazo robótico (envase cilíndrico)

En la última prueba se manipula un objeto de mayor diámetro y peso. Los dedos se adaptaron correctamente a la forma del objeto y los motores ejercieron el torque necesario para moverlo de un punto a otro sin inconvenientes. Se debe considerar que ésta y todas las pruebas anteriores se realizaron a la misma distancia.

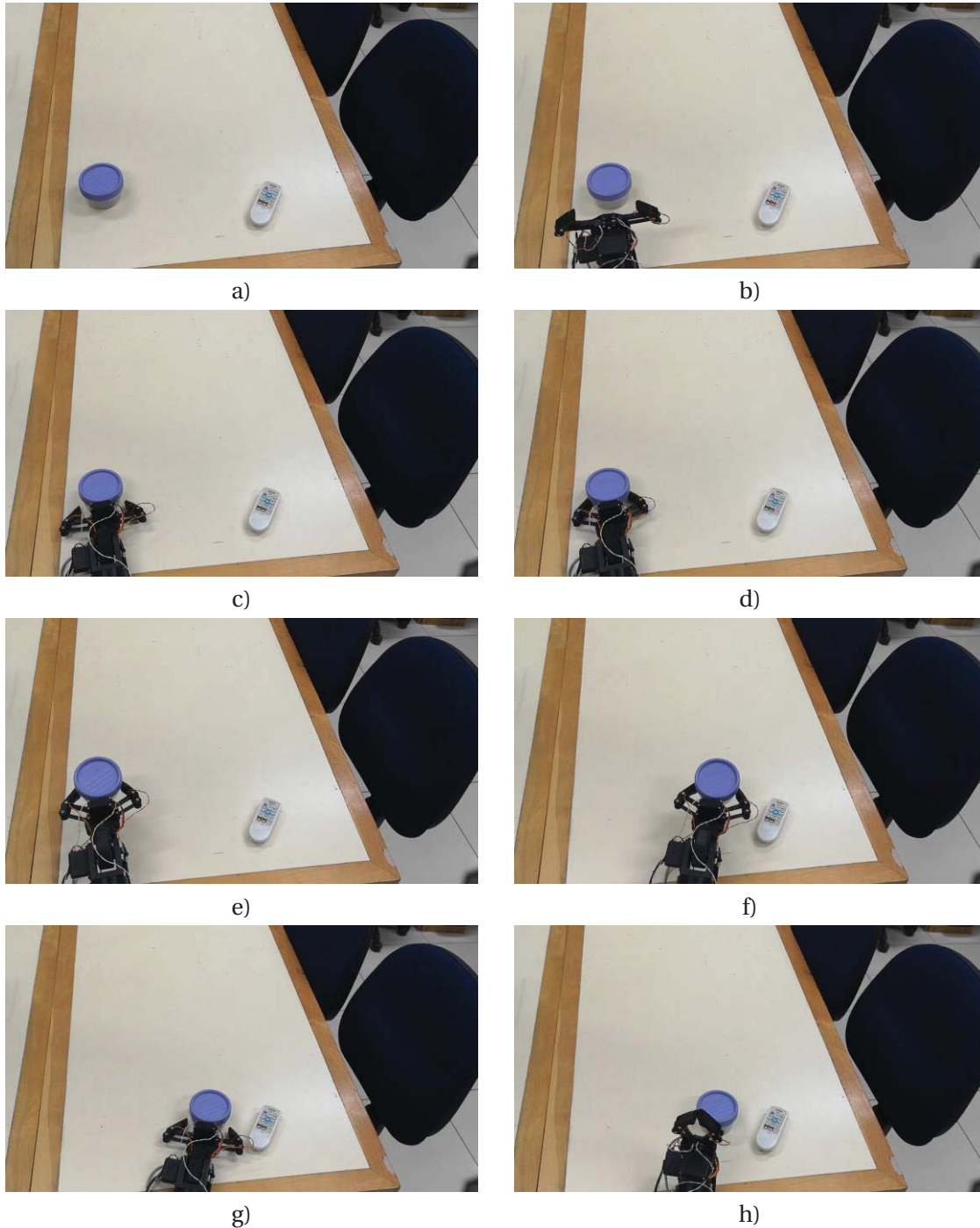


Figura 4-18: Secuencia de prueba de agarre y desplazamiento de brazo robótico de objeto cilíndrico (vista desde cabeza robótica)

Discusión y conclusiones

Para realizar la construcción del brazo robótico fue necesario generar un diseño que se ajustara a las necesidades y presupuestos del Laboratorio de Robótica de la Escuela de Ingeniería Eléctrica. Es así como se determinó disminuir el número de grados de libertad, considerando que la plataforma móvil donde se instalará permite movimientos que reemplazan en cierta medida estos grados de libertad.

Al poseer una representación virtual del modelo confeccionado, es posible simular y estimar su comportamiento sin la necesidad de realizar la construcción de éste, evitando una inversión innecesaria. Otra ventaja de realizar una representación virtual en un software de diseño es que permite realizar adaptaciones, cambios en el modelo y almacenar distintas versiones en base a una estructura común.

Una vez construido el brazo robótico y ensamblado en la plataforma para realizar pruebas de funcionamiento se detectó un problema que podría significar realizar modificaciones a la estructura del hombro, pues lo que se indicó inicialmente en el análisis de torque ocurre en la práctica, es decir, la sección del hombro se tiende a inclinar produciendo mayor roce entre la pieza de soporte y la pieza que debe rotar en su interior. Como solución se reforzó la estructura y se lubricó mejor el contacto entre piezas, mitigando en parte este inconveniente y obteniendo un correcto funcionamiento.

La textura de los dedos al instalar los sensores de presión con cinta aislante aumenta, permitiendo un buen agarre. Los actuadores de la muñeca funcionaron sin problemas, manipulando objetos pequeños y dentro del peso de operación para el que fue diseñado (0.5 [kg]).

La implementación de sensores de presión en la mano robótica permite un mayor control y precisión a la hora de maniobrar distintos objetos. Hay que tomar en consideración que existen objetos que podrían ser más frágiles y requieren un mayor control de las fuerzas que se le aplican.

Durante la construcción de los sensores se utilizaron diferentes materiales adhesivos con el fin de buscar alternativas que pudiesen mejorar el desempeño del diseño propuesto en [1], sin embargo, no hubo mejoras. La aplicación del adhesivo Agorex causó una drástica disminución en la resistencia del sensor, limitando el rango dinámico. También se utilizó cinta adhesiva doble faz, aumentando la resistencia del sensor en orden del megaohm, superando la escala del multímetro

y haciendo imposible su caracterización. Vale decir que, para este caso, al ejercer presiones muy grandes si existe una disminución de la resistencia, pero no es acorde a las necesidades de la mano robótica. Por último, se utilizó cinta adhesiva Scotch, sellando exteriormente el sensor, sin intervenir en el contacto existente entre las capas conductoras y el polímero, entregando un rango dinámico amplio, y buena respuesta a las presiones ejercidas.

Los sensores de presión textiles de tipo resistivos, en la práctica demuestran que pueden adaptarse a un sinnúmero de tareas, y no solo se limitan a aplicaciones de ropa, sino que también, a la robótica y electrónica general. Por otro lado, los sensores de presión tienden a variar demasiado en sus valores al momento de realizar movimientos con el brazo robótico, esto se puede deber a que los cables utilizados (cable UTP), tienden a ser rígidos, produciendo movimiento en el sensor, desprendimiento del material textil, o su rotura. Como solución se pueden utilizar hilos conductores que otorguen mayor libertad de movimiento de la mano, sin afectar la posición del sensor.

La sujeción de los objetos es aceptable si se utiliza dentro de los márgenes de diseño, sin embargo, los movimientos de desplazamiento tienden a ser bruscos debido a que se ha programado un *script* de prueba indicando a los servomotores solo los ángulos a los que deben llegar, lo que implica que el servomotor se mueva a la velocidad establecida por defecto por el fabricante.

Respecto a la visión por computador, se ha logrado generar nodos relacionados con la visión para reconocimiento de objetos y publicación de las coordenadas del objeto a manipular, no así la integración de ROS en su totalidad. La tarjeta de desarrollo Arduino aún no se encuentra asociada a ROS, pero, para dar solución a esta problemática se debe conectar la tarjeta de desarrollo por medio de un puerto USB al computador que tiene la función de controlador o *ROS Master*. Luego, es necesario configurar el nodo en ROS para que tome control del brazo y para que la cabeza robótica envíe instrucciones al nodo correspondiente a la tarjeta de desarrollo *Pícaro Mega* para accionar el brazo robótico.

El modelo pre-entrenado implementado en la visión por computador es deficiente en cantidad de clases, por ello es recomendable utilizar modelos pre-entrenados con un mayor número de clases, acorde a los elementos cotidianos del laboratorio. Otra posibilidad es reentrenar la red, utilizando una base de datos propia.

El desempeño del brazo robótico resulta satisfactorio en un entorno controlado, ejecutando rutinas preestablecidas. Como trabajos futuros se debe integrar el brazo a ROS y generar el enlace con la visión, permitiendo la automatización del proceso de manipulación de objetos. También se deben mejorar los scripts utilizados en el control de los actuadores del brazo robótico. Se recomienda reemplazar los cables UTP de los sensores por hilos conductores de material textil. Finalmente se debe diseñar y construir una placa con la circuitería de control y conexiones que actualmente se encuentra en una *protoboard*.

Bibliografía

- [1] F. Pizarro, P. Villavicencio, D. Yunge, M. Rodríguez, G. Hermosilla y A. Leiva, «Easy-to-Build Textile Pressure Sensor,» *Sensors*, vol. 18, nº 4, pp. 1-13, 2018.
- [2] International Federation of Robotics, «Why service robots boom worldwide,» de *Executive Summary World Robotics 2017 Service Robots*, Brussels, 2017.
- [3] International Federation of Robotics, «How robots conquer industry worldwide,» de *Executive Summary World Robotics 2017 Industrial Robots*, Frankfurt, 2017.
- [4] J. Stückler, M. Schwarz y S. Behnke, «Mobile Manipulation, Tool Use, and Intuitive Interaction for Cognitive Service Robot Cosero,» *Frontiers in Robotics and AI*, vol. 3, nº 58, pp. 1-20, 2016.
- [5] S. W. Ruehl, C. Parlitz, G. Heppner, A. Hermann, A. Roennau y R. Dillmann, «Experimental Evaluation of the Schunk 5-Finger Gripping Hand for Grasping Tasks,» de *International Conference on Robotics and Biomimetics*, Indonesia, 2014.
- [6] Erle Robotics, «ROS Introduction ES,» Acutronic Robotics company., 2018. [En línea]. Available: <https://erlerobotics.com/blog/ros-introduction-es/>. [Último acceso: 17 07 2018].
- [7] ROS, «ROS.org | Powering the world's robots,» Creative Commons Attribution 3.0, [En línea]. Available: <http://www.ros.org/>. [Último acceso: 17 07 2018].
- [8] Real Academia Española, «Diccionario de la lengua española - Sensor,» 2018. [En línea]. Available: <http://dle.rae.es/?w=sensor>. [Último acceso: 26 Noviembre 2018].
- [9] Real Academia Española, «Diccionario de la lengua Española - Transductor,» [En línea]. Available: <http://dle.rae.es/srv/search?m=30&w=transductor>.

-
- [10] E. Sobrado Malpartida y J. C. Tafur Sotelo, «Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot,» PUCP, Lima, 2003.
- [11] G. J. González Osorio, «Reconocimiento de objetos utilizando OpenCV y Python en una Raspberry pi 2 en una tlapalería,» Universidad Autónoma del Estado de México, Texcoco, 2017.
- [12] G. L. Nail Valenzuela, «Navegación autónoma de una base robótica usando SLAM».
- [13] N. S. Carpio Aravena, «Inteligencia Computacional para una cabeza robótica,» Valparaíso, 2018.
- [14] R. C. Goertz, «REMOTE-CONTROL MANIPULATOR». United States Patente US2774488, 18 December 1956.
- [15] R. C. Goertz, «ELECTRONIC MASTER SLAVE MANIPULATOR». Chicago Patente US2846048, 5 August 1958.
- [16] A. Barrientos Cruz, L. F. Peñín, C. Balaguer y R. Aracil, FUNDAMENTOS DE ROBOTICA, Madrid: S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, 2007.
- [17] B. Malone, «IEEE Spectrum,» 26 Septiembre 2011. [En línea]. Available: <https://spectrum.ieee.org/automaton/robotics/industrial-robots/george-devol-a-life-devoted-to-invention-and-robots>. [Último acceso: 19 Abril 2018].
- [18] H. P. Moravec, «Encyclopædia Britannica,» 15 Noviembre 2018. [En línea]. Available: <https://www.britannica.com/technology/robot-technology#ref849969>. [Último acceso: 25 Diciembre 2018].
- [19] IEEE Spectrum, «Robots, Your Guide to the World of Robotics,» 2019. [En línea]. Available: <https://robots.ieee.org/robots/unimate/>. [Último acceso: 13 05 2019].
- [20] K. Berns, H. Vogt, T. Asfour y R. Dillmann, «Design and Control Architecture of an Anthropomorphic Robot,» de *Intern. Conf. on Advanced Mechatronics (ICAM'98)*, Okayama, 1998.
- [21] T. Lens, J. Kunz, O. von Stryk, C. Trommer y A. Karguth, «BioRob-Arm: A Quickly Deployable and Intrinsically Safe, LightWeight Robot Arm for Service Robotics Applications,» de *41st International Symposium on and 2010 6th German Conference on Robotics*, Berlin, 2010.
- [22] SCHUNK GmbH & Co. KG, «SCHUNK,» 2018. [En línea]. Available: [http://www.schunk-](http://www.schunk-modular-)

- robotics.com/fileadmin/user_upload/PDF_Downloads/SCHUNK_Technical_data_LWA4 D.pdf. [Último acceso: 21 abril 2018].
- [23] Commonplace Robotics GmbH, «SRA Service Robot Arm,» 2016. [En línea]. Available: http://www.cpr-robots.com/ressourcen/ProductSheet_SRA.pdf. [Último acceso: 22 abril 2018].
- [24] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire y L.-J. Caron L'Ecuyer, «Kinova Modular Robot Arms for Service Robotics Applications,» *International Journal of Robotics Applications and Technologies*, vol. 5, nº 2, pp. 49-71, 2017.
- [25] D. Mihai, «Smashing Robotics,» 16 Mayo 2014. [En línea]. Available: <https://www.smashingrobotics.com/overview-of-commercially-available-professional-robotic-manipulators-part-2/>. [Último acceso: 15 Abril 2018].
- [26] F. Erskine Crossley y F. Umholtz, «Design for a three-fingered hand,» *Mechanism and Machine Theory*, vol. 12, nº 1, pp. 85-93, 1977.
- [27] ROBOTIQ, «3-Finger Adaptive Robot Gripper,» 2015. [En línea]. Available: <https://robotiq.com/products/3-finger-adaptive-robot-gripper?301=%2Fproducts%2Findustrial-robot-hand%2F>. [Último acceso: 22 abril 2018].
- [28] Prensilia S.r.l., «Prensilia,» 2014. [En línea]. Available: <https://www.prensilia.com/portfolio/ih2-azzurra/>. [Último acceso: 22 abril 2018].
- [29] Shadow Robot Company, «Shadow Dexterous Hand E Series (E2M3R, E2M3L, E2P1R, E2P1L),» 17 agosto 2015. [En línea]. Available: https://www.shadowrobot.com/wp-content/uploads/shadow_dexterous_hand_technical_specification_E_20150827.pdf. [Último acceso: 21 abril 2018].
- [30] G. A. Kragten, M. Baril, C. Gosselin y J. L. Herder, «Stable Precision Grasps by Underactuated Grippers,» *IEEE TRANSACTIONS ON ROBOTICS*, vol. 27, nº 6, pp. 1056-1066, 2011.
- [31] F. Zhuang, Z. Hangfei, L. Zijuan, F. Jian, Y. Weixin y Z. Yanzheng, «Design of a Robot End-Effector Grabbing Mechanism Based on a Bionic Snake Mouth,» *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1-7, 2012.
- [32] Empire Robotics, Inc., «Verseball Gripper,» 2015. [En línea]. [Último acceso: 22 abril 2018].
- [33] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson y H. M. Jaeger, «Universal robotic gripper based on the jamming of granular material,» *PNAS*, vol. 107, nº 44, pp. 18809-18814, 2010.

- [34] Z. Gong, J. Cheng, X. Chen, W. Sun, X. Fang, K. Hu, Z. Xie, T. Wang y L. Wen, «A Bio-inspired Soft Robotic Arm: Kinematic Modeling and Hydrodynamic Experiments,» *Journal of Bionic Engineering*, vol. 15, n° 2, pp. 204-219, 2018.
- [35] FESTO, «FESTO,» 2006-2009. [En línea]. Available: https://www.festo.com/net/SupportPortal/Files/42058/Airics_arm_en.pdf. [Último acceso: 21 abril 2018].
- [36] FESTO, «FESTO,» 2010-2012. [En línea]. Available: https://www.festo.com/PDF_Flip/corp/Festo_ExoHand/en/files/assets/basic-html/page-1.html#. [Último acceso: 21 abril 2018].
- [37] P. Ohta, L. Valle, J. King, K. Low, J. Yi, C. G. Atkeson y Y.-L. Park, «Design of a Lightweight Soft Robotic Arm Using Pneumatic Artificial Muscles and Inflatable Sleeves,» *Soft Robotics*, vol. 00, n° 00, pp. 1-12, 2017.
- [38] R. B. McIntosh, P. E. Mauger y S. R. Patterson, «Capacitive Transducers With Curved Electrodes,» *IEEE SENSORS JOURNAL*, vol. 6, n° 1, pp. 125-138, 2006.
- [39] L. G. Corona Ramírez, G. S. Abarca Jiménez y J. M. Carreño, *Sensores y Actuadores*, Grupo Editorial Patria, 2014.
- [40] L. Ricotti, T. Ranzani, V. Calarota y A. Menciassi, «Thin and flexible pressure/deformation sensors based on piezoelectric nanocomposites (Piezoelectric nanoparticles embedded in tunable polymeric matrices),» de *SENSORS, 2014 IEEE*, Valencia, 2014.
- [41] X. Lin y B.-C. Seet, «Battery-Free Smart Sock for Abnormal Relative Plantar Pressure Monitoring,» *IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS*, vol. 11, n° 2, p. 464-473, 2017.
- [42] J. Meyer, B. Arnrich, J. Schumm y G. Tröster, «Design and Modeling of a Textile Pressure Sensor for Sitting Posture Classification,» *IEEE SENSORS JOURNAL*, vol. 10, n° 8, pp. 1391-1398, 2010.
- [43] S. Choi y Z. Jiang, «A novel wearable sensor device with conductive fabric and PVDF film for monitoring cardiorespiratory signals,» *Sensors and Actuators A: Physical*, vol. 128, n° 2, pp. 317-326, 2006.
- [44] M. Tessarolo, L. Possanzini, E. G. Campari, R. Bonfiglioli, F. Saverio Violante, A. Bonfiglio y B. Fraboni, «Adaptable pressure textile sensors based on a conductive polymer,» *Flexible and Printed Electronics*, vol. 3, pp. 1-11, 2018.
- [45] moocintef educaLAB, «Glosario de términos y conceptos robóticos,» 2017. [En línea]. Available:

- <http://el.uy/doc/robotica/mooc.educalab.es.%202017.%20Glosario%20de%20t%C3%A9rminos%20y%20conceptos%20rob%C3%B3ticos.pdf>. [Último acceso: 26 Diciembre 2018].
- [46] National Research Council, *Expanding the Vision of Sensors Materials*, Washington: National Academy Press, 1995.
- [47] J. Carr y J. Brown, *Introduction to Biomedical Equipment Technology*, New Jersey: Prentice Hall, 2001.
- [48] R. Pallás Areny, *Sensores y acondicionadores de señal*, Barcelona: Marcombo, 2014.
- [49] Khan Academy, «Khan Academy,» 01 Enero 2018. [En línea]. Available: <https://es.khanacademy.org/science/physics/work-and-energy/hookes-law/a/what-is-hookes-law>. [Último acceso: 24 Noviembre 2018].
- [50] National Instruments Corporation, «Strain Gauge Measurement – A Tutorial,» Texas, 1998.
- [51] R. diMauro, «Mediciones Eléctricas II,» 2017. [En línea]. Available: <https://docplayer.es/48088553-Mediciones-electricas-ii.html>. [Último acceso: 23 Noviembre 2018].
- [52] CIDEAD, «Fuerza y Presión en los fluidos,» [En línea]. Available: <http://recursostic.educacion.es/secundaria/edad/4esofisicaquimica/impresos/quincena4.pdf>. [Último acceso: 20 04 2018].
- [53] A. Castillo Ramírez, «Cinémática de las Máquinas,» San Luís Potisí, 2005.
- [54] J. Edward Shigley y J. Uicker, *Teoría de máquinas y mecanismos*, McGraw-Hill, 1983.
- [55] International Organization for Standardization, «ISO 8373:2012(en), Robots and robotic devices,» 2018. [En línea]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en:term:4.3>. [Último acceso: 20 abril 2018].
- [56] International Federation of Robotics, «IFR,» 2 Octubre 2016. [En línea]. Available: https://ifr.org/img/office/Service_Robots_2016_Chapter_1_2.pdf. [Último acceso: 20 Abril 2018].
- [57] D. A. Moya Pinta, «Modelo y análisis cinemático de un robot manipulador esférico industrial aplicando MATLAB,» Quito, 2010.
- [58] L. Aponte, «Contenido 14 - Máquinas de Desplazamiento Positivo,» 1 Lulio 2014. [En línea]. Available: <https://sites.google.com/site/desplazamientopositivo/tema-4/clase-13>. [Último acceso: 8 Octubre 2018].

-
- [59] J. A. Somolinos Sánchez, Avances en robótica y visión por computador, Castilla-La Mancha: Universidad de Castilla La Mancha, 2002.
- [60] F. A. Candelas Herías y J. A. Corrales Ramón, «Servomotores,» 2007. [En línea]. Available: <http://www.aurova.ua.es/previo/dpi2005/docs/publicaciones/pub09-ServoMotores/servos.pdf>. [Último acceso: 7 Octubre 2018].
- [61] J. Posada Contreras, El Hombre y la Máquina, Colombia: Universidad Autónoma de Occidente, 2005, p. 72.
- [62] LINAK Actuadores, S.L.u, «¿Qué es un actuador lineal?,» 2018. [En línea]. Available: <http://www.linak.es/about/?id3=4283>. [Último acceso: 7 Octubre 2018].
- [63] Mecatrónica Latam, «SERVOMOTOR ¿Qué es y cómo funciona?,» 8 Abril 2018. [En línea]. Available: <https://www.mecatronicalatam.com/motores/servomotor>. [Último acceso: 7 Octubre 2018].
- [64] L. Llamas, «Controlar un servo con arduino,» 6 Junio 2016. [En línea]. Available: <https://www.luisllamas.es/controlar-un-servo-con-arduino/>. [Último acceso: 7 Octubre 2018].
- [65] Fircelli Automations Team, «Fircelli Automations,» 17 Noviembre 2014. [En línea]. Available: <https://www.fircelliauto.com/blogs/news/how-does-a-linear-actuator-work>. [Último acceso: 7 Octubre 2018].
- [66] MCI Electronics, «PICARO Mega (Arduino Mega Compatible),» 2018, [En línea]. Available: <https://www.mcielectronics.cl/shop/product/picaro-mega-arduino-mega-compatible-25601>. [Último acceso: 7 Octubre 2018].
- [67] Ingeniería mecafenix, «Encoder ¿cómo funciona? y sus tipos,» 2018. [En línea]. Available: <http://www.ingmecafenix.com/automatizacion/encoder/>. [Último acceso: 7 Octubre 2018].
- [68] MCI Electronics, «Grove Encoder,» 2018. [En línea]. Available: <https://www.mcielectronics.cl/shop/product/grove-encoder-22851>. [Último acceso: 7 Octubre 2018].
- [69] J. Fontrodona Francolí y R. Blanco Díaz, Estado actual y perspectivas de la impresión en 3D, Barcelona: Generalitat de Catalunya, 2014.
- [70] E. G. Learned-Miller, Introduction to Computer Vision, Amherst, 2011.
- [71] J. R. Mejía Vilet, «Apuntes de Procesamiento digital de imágenes,» San Luis Potosí, 2005.

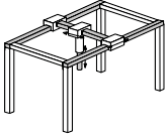
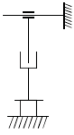
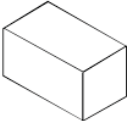
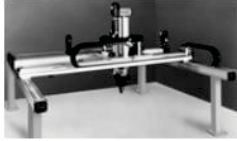
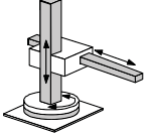
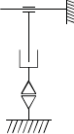


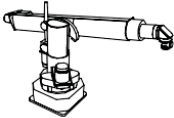
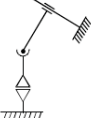


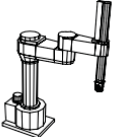
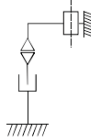
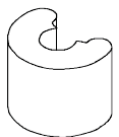

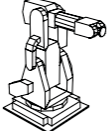
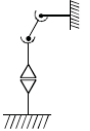


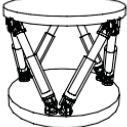
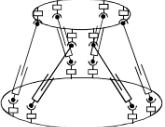


- [72] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, «Conceptos básicos de imagen digital,» 2018. [En línea]. Available: <http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/pdf/imagen01.pdf>. [Último acceso: 25 Noviembre 2018].
- [73] S. Malireddi, «COMPUTER VISION IN IOS – SWIFT+OPENCV,» 4 Junio 2017. [En línea]. Available: <https://sriraghu.com/2017/06/04/computer-vision-in-ios-swiftopencv/>. [Último acceso: 24 Noviembre 2018].
- [74] P. Querejeta Simbeni, «Procesamiento digital de imágenes,» Bahía Blanca, 2015.
- [75] Facultad de Ingeniería - Universidad Nacional de San Juan, «Capitulo 1. Procesamiento digital de imágenes,» 2018. [En línea]. Available: <http://dea.unsj.edu.ar/imagenes/recursos/Capitulo1.pdf>. [Último acceso: 24 Noviembre 2018].
- [76] N. L. Fernández García, «Tema 1.- Introducción a la Visión Artificial,» Córdoba, 2012.
- [77] W. Pitts y W. S. McCulloch, «How we know universals the perception of auditory and visual forms,» *Bulletin of Mathematical Biophysics*, vol. 9, pp. 127-147, 1947.
- [78] D. E. Rumelhart, G. E. Hinton y R. J. Williams, «Learning representation by back-propagating error,» *Nature*, vol. 323, n° 6088, pp. 533-536, 1986.
- [79] R. E. López Briega, «IAAR Capacitación - Introducción al Deep Learning,» 13 Junio 2017. [En línea]. Available: <https://iaarhub.github.io/capacitacion/2017/06/13/introduccion-al-deep-learning/>. [Último acceso: 24 Noviembre 2018].
- [80] G. E. Hinton y R. R. Salakhutdinov, «Reducing the Dimensionality of Data with Neuronal Networks,» *Science*, vol. 313, n° 5786, pp. 504-507, 2006.
- [81] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath y B. Kingsbury, «Deep Neural Networks for Acoustic Modeling in Speech Recognition,» *IEEE SIGNAL PROCESSING MAGAZINE*, vol. 29, n° 6, pp. 82-97, 2012.
- [82] L. Araujo dos Santos, «Artificial Inteligence - GitBook,» 2017. [En línea]. Available: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/pooling_layer.html. [Último acceso: 25 Noviembre 2018].
- [83] R. Gandhi, «Towards Data Science,» 9 Junio 2018. [En línea]. Available: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. [Último acceso: 25 Noviembre 2018].

-
- [84] J. Xu, «Uso del aprendizaje profundo para el reconocimiento de objetos.,» 8 Junio 2018. [En línea]. Available: <https://www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos/>. [Último acceso: 25 Noviembre 2018].
- [85] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu y A. C. Berg, «SSD: Single Shot MultiBox Detector,» de *Computer Vision – ECCV 2016*, Amsterdam, 2016.
- [86] Towards Data Science, «Single shot multibox detector with pytorch,» 2018. [En línea]. Available: <https://towardsdatascience.com/learning-note-single-shot-multibox-detector-with-pytorch-part-1-38185e84bd79>. [Último acceso: 24 Noviembre 2018].
- [87] A. Jaramillo Botero, «Cinemática de Manipuladores Robóticos,» de *CINEMÁTICA DE MANIPULADORES*, pp. 1-68.
- [88] Roboteus, Inc., «Roboteus - Education Robots,» 2018. [En línea]. Available: <https://roboteurs.com/>. [Último acceso: 2018 Octubre 04].
- [89] Roboteus, Inc., «RBX1 - Remix Design Files - Roboteus,» 2018. [En línea]. Available: <https://roboteurs.com/pages/rbx1-remix-design-files>. [Último acceso: 04 Octubre 2018].
- [90] C. B. Goy, V. Menichetti, L. M. Yanicelli, J. B. Lucero, M. A. Gómez López, N. F. Parodi y M. C. Herrera, «Design, fabrication and metrological evaluation of wearable pressure sensors,» *Journal of Medical Engineering & Technology*, vol. 39, n° 3, pp. 208-215, 2015.
- [91] Technical Textiles Inc., «Shieldex trading,» 10 04 2018. [En línea]. Available: <https://www.shieldextrading.net/wp-content/uploads/2018/08/1401101S80-NoraDell.pdf>. [Último acceso: 23 Noviembre 2018].
- [92] ANTISTAT, «Black Conductive Bag ANT006BCB,» 8 Enero 2018. [En línea]. Available: https://www.antistat.co.uk/wp-content/uploads/2018/01/ANT006BCB_Black-Conductive-Bag.pdf. [Último acceso: 23 Noviembre 2018].

A Morfología de los robots

La Tabla A-1 muestra los tipos de robots de acuerdo a su estructura mecánica, para este proyecto se utilizó la morfología de tipo angular.

Tabla A-1: Robots desglosados por estructura mecánica (fuente: <http://www.asimo.pl>).

Principio	Estructura cinemática	Espacio de trabajo	Ejemplo
 <p>Cartesiano</p>			
 <p>Cilíndrico</p>			
 <p>Polar o esférico</p>			
 <p>SCARA</p>			
 <p>Angular</p>			
 <p>Paralelo</p>			

B Circuito sensores y actuadores

B.1 Esquema Simplificado circuito sensores y actuadores

En la Figura B-1 se muestra el esquema simplificado de las conexiones de los sensores a la placa de desarrollo *Pícaro Mega (Arduino)* como entrada, y la salida a los distintos actuadores. El Actuator Lineal requiere un *Driver*, el cual permite que pueda moverse en dos sentidos.

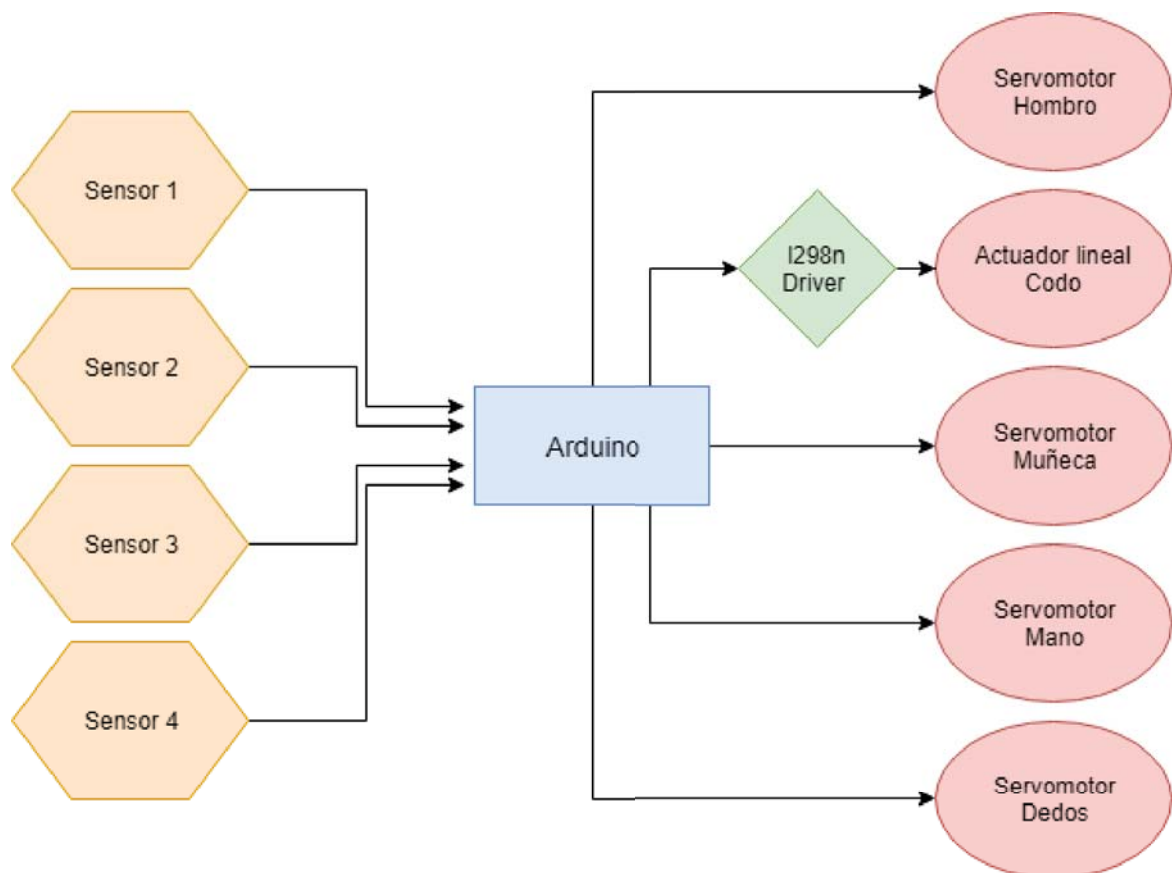


Figura B-1: Esquema simplificado de conexiones de sensores y actuadores a tarjeta de desarrollo Arduino

B.2 Conexiones de actuadores y sensores

En la Figura B-2 se muestra la conexión de todos los actuadores presentes en el brazo, como se observa, los servomotores del Hombro, Muñeca, Mano y Dedos requieren la utilización de una fuente de 6 volts como máximo, en cambio el motor DC que conforma el actuador lineal requiere una fuente de 12 volts conectada directamente en el *driver*. Además, las señales de control son conectadas a las salidas digitales PWM configuradas en la placa.

Por otro lado, los sensores de presión producen señales analógicas, por lo tanto, dichos sensores deben ser conectados en las entradas analógicas como se muestra en la Figura B-3, donde son transformadas en señales digitales mediante un convertor A/D.

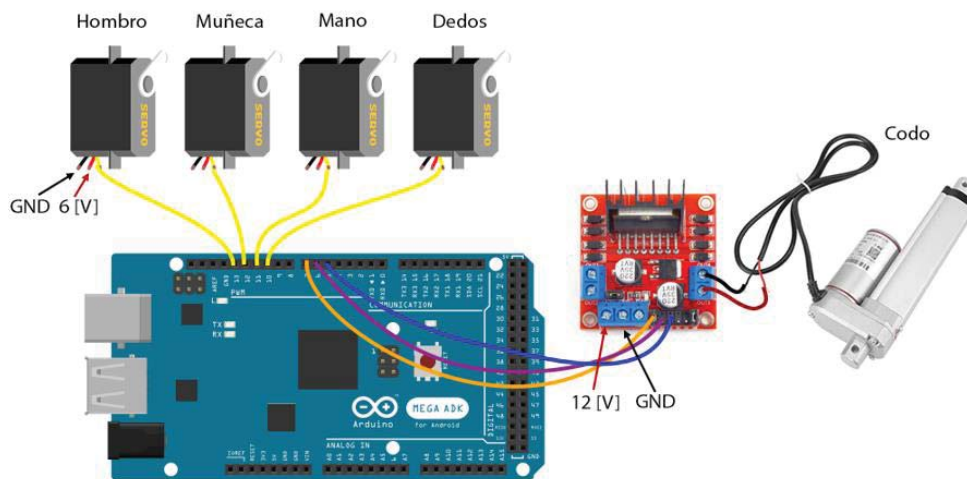


Figura B-2: Esquema de conexión de los actuadores a placa de desarrollo

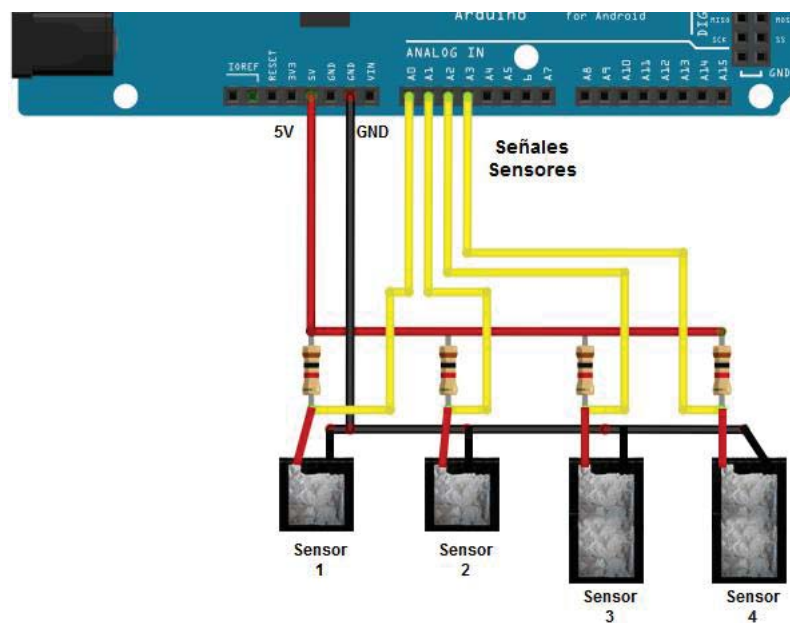


Figura B-3: Esquema de conexión de sensores a placa de desarrollo

C Componentes y diagrama de brazo robótico

En las siguientes figuras se muestran los componentes utilizados para la elaboración del brazo robótico. En la Figura C-1 se muestran las piezas 3D que forman parte de la mano robótica, inspirada inicialmente en el diseño realizado por la empresa *Roboteus, Inc.*



Figura C-1: Componentes de la mano de brazo robótico para manipulación de objetos

La Figura C-2 muestra los componentes que han sido impresos en 3D que forman la estructura de la muñeca y parte del codo. También se presentan piezas como el rodamiento 63042RS, que da soporte al eje del servomotor (que gira la muñeca), los tubos de aluminio que forman la estructura del antebrazo y brazo, además de las abrazaderas que soportan el motor lineal como se muestra en la Figura C-5.



Figura C-2: Componentes de muñeca y codo de brazo robótico para manipulación de objetos

En el caso de la Figura C-3, se presentan los componentes que componen el codo y el hombro. El rodamiento 6252Z es utilizado en la muñeca para soportar el eje que se conecta con la mano y el rodamiento axial 51107RLW es utilizado para facilitar el giro de la articulación del hombro.



Figura C-3: Componentes de codo y hombro de brazo robótico para manipulación de objetos

La Figura C-4 muestra los componentes electrónicos que permiten el movimiento de la estructura del brazo robótico. Entre ellos se encuentran los servomotores para las articulaciones de los dedos, muñeca y hombro. Además, existe un actuador lineal y un *driver* que permite el movimiento de la articulación del codo en dos sentidos. Otro componente importante es la placa de desarrollo, la cual que actúa como centro de control de los componentes antes mencionados y se encarga, además, de recibir señales provenientes de los sensores de las manos y del *encoder*, de procesar dicha información y transformarla en acciones coherentes.



Figura C-4: Componentes electrónicos de Brazo y mano robótica para manipulación de objetos

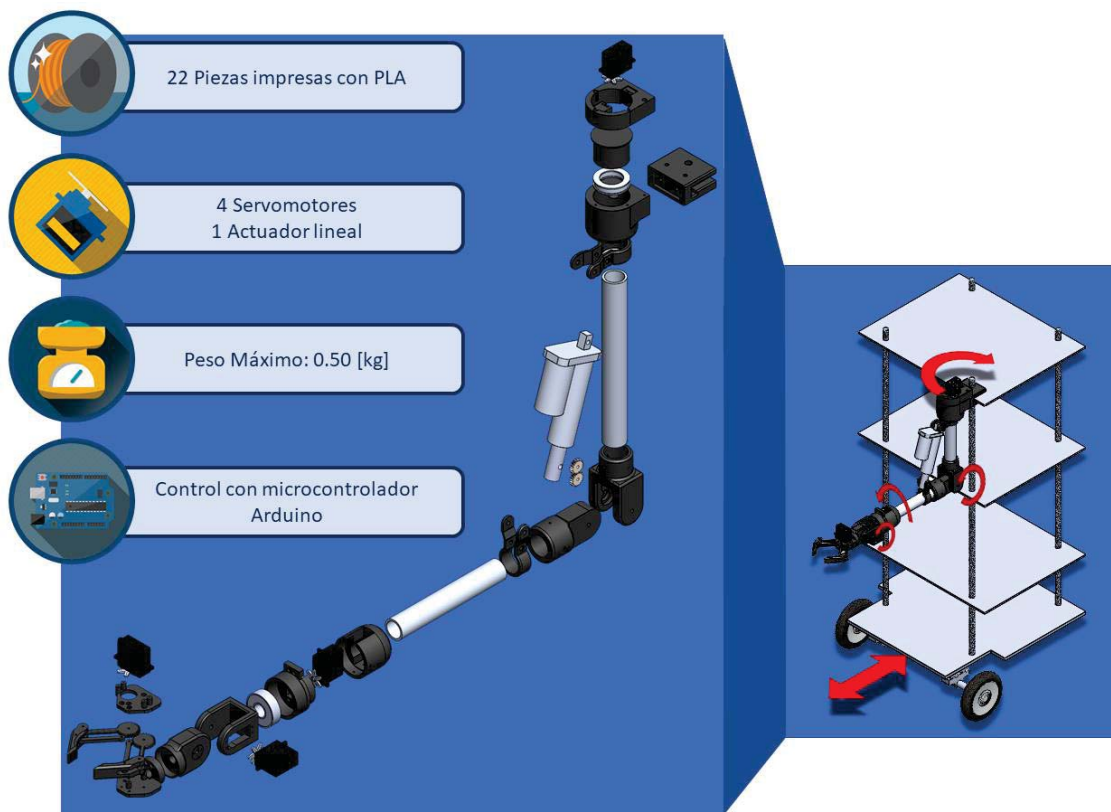


Figura C-5: Esquema Brazo Robótico indicando principales características, movimientos del brazo en relación a la plataforma y movimiento de plataforma al realizar manipulación de objetos

D Presupuesto

El presupuesto inicial consideraba un total de \$199.445.- debido a que se realizaron compras adicionales como por ejemplo rollos de PLA utilizados en piezas de prueba previas, entre otros artículos. Sin embargo, el presupuesto se puede reducir considerablemente utilizando el mínimo de recursos para replicar el brazo robótico construido en el presente informe, el cual entrega un total de \$157.766.-, como se muestra en la Tabla D-1.

Producto	Cantidad	Valor unitario (\$)	Valor total (\$)	Proveedor
Tubo aluminio 1[m]	1	6.530	6.530	ACEROSVAL
Servo MG996R	2	5.450	10.900	ALTRONICS
Servo MG995R	1	5.250	5.250	ALTRONICS
Disco 25T para Servo	3	979	2.937	ALTRONICS
Servomotor DS3218	1	17.950	17.950	ALTRONICS
Pernos M3	20	17	340	DIPERVAL
Tuercas	20	12	240	DIPERVAL
Rodamiento 6252Z	1	1.107	1.107	DUCASSE
Rodamiento 63042RS	1	1.499	1.499	DUCASSE
Rodamiento 51107RLW	1	5.248	5.248	DUCASSE
Perno hexagonal G2	1	1.990	1.990	EASY
Picaro Mega (Arduino)	1	28.990	28.990	MCI ELECTRONICS
Grove Encoder	1	8.970	8.970	MCI ELECTRONICS

Tabla D-1a: Presupuesto de componentes mínimo para llevar a cabo la construcción del brazo robótico

Producto	Cantidad	Valor unitario (\$)	Valor total (\$)	Proveedor
Actuador lineal	1	45.835	45.835	RAMBAL
Filamento PLA 1 [kg]	1	15.990	15.990	VOLTA
Cable UTP 5 [m]	1	3.990	3.990	CASA ROYAL
Total			157.766	

Tabla D-2b: Presupuesto de componentes mínimo para llevar a cabo la construcción del brazo robó

E Scripts para prueba de sensores y manipulación

El *script* que se muestra en Listado E-1 se utiliza en la adquisición de datos de los sensores. Para llevar a cabo dicha adquisición, el script Arduino se complementa con el programa PLX-DAQ, el cual recibe la información a través de conexión serial. El código envía un comando que limpia el archivo Excel donde se escriben los datos, luego etiqueta cada columna donde se realizará la importación de éstos. Una vez que se leen las cuatro entradas análogas de la tarjeta de desarrollo *Picaro MEGA*, se guardan las variables para calcular mediante un divisor de voltaje, la tensión en los sensores, se calcula la resistencia de los sensores y el tiempo. Finalmente, se envía la información y es almacenado en el documento Excel.

Listado E-1: Script Arduino para realizar la adquisición de datos de los sensores de presión por medio de programa PLX-DAQ

```
1 int pin_medicion = A0;
2 int R1 = 994;
3 long lectura1 = 0;
4 long lectura2 = 0;
5 long lectura3 = 0;
6 long lectura4 = 0;
7 float tension_de_trabajo = 5.0;
8 float voltaje1;
9 float voltaje2;
10 float voltaje3;
11 float voltaje4;
12 float resistencial;
13 float resistencia2;
14 float resistencia3;
15 float resistencia4;
16
17 float tiempo;
18
19 int i = 0;
20 int v = 0;
21
22 void setup()
23 {
24
25     // Abriendo conexión serial
26     Serial.begin(9600);
27
28     //Eliminar Data en archivo excel e imprime las etiquetas de las columnas
29     Serial.println("CLEARDATA");
30     Serial.println("LABEL,Muestra,Segundos,Voltaje1,Voltaje2,Voltaje3,Voltaje4");
```

```

31 }
32
33 void loop()
34 {
35     //Lectura de datos y envío por serial
36     lectura1 = analogRead(A0);
37     lectura2 = analogRead(A1);
38     lectura3 = analogRead(A2);
39     lectura4 = analogRead(A3);
40     lectura1 = trunc(lectura1);
41     lectura2 = trunc(lectura2);
42     lectura3 = trunc(lectura3);
43     lectura4 = trunc(lectura4);
44
45     voltaje1 = (tension_de_trabajo/1023.0)*lectura1;
46     voltaje2 = (tension_de_trabajo/1023.0)*lectura2;
47     voltaje3 = (tension_de_trabajo/1023.0)*lectura3;
48     voltaje4 = (tension_de_trabajo/1023.0)*lectura4;
49
50     resistencial = R1*lectura1/(tension_de_trabajo-voltaje1);
51     resistencia2 = R1*lectura2/(tension_de_trabajo-voltaje2);
52     resistencia3 = R1*lectura3/(tension_de_trabajo-voltaje3);
53     resistencia4 = R1*lectura4/(tension_de_trabajo-voltaje4);
54
55     tiempo = millis()/1000.0;
56
57     Serial.print("DATA,");
58     Serial.print(i++); Serial.print(",");
59     Serial.print(tiempo);
60     Serial.print(",");
61     Serial.print(voltaje1);
62     Serial.print(",");
63     Serial.print(voltaje2);
64     Serial.print(",");
65     Serial.print(voltaje3);
66     Serial.print(",");
67     Serial.print(voltaje4);
68     Serial.print(",");
69     Serial.println(resistencia);
70     delay(50);
71 }

```

El Listado E-2 muestra el código correspondiente al utilizado en las pruebas de manipulación de objetos y lectura de sensores. Al inicio del código mostrado en el *script* se importa la librería *Servo.h*, la cual permite utilizar los servomotores con funciones predefinidas, facilitando su implementación. La segunda librería utilizada se denomina *L298N.h*, y corresponde a la librería que permite la utilización del *driver* controlador del actuador lineal. Posteriormente se definen las distintas constantes y variables, se habilita la comunicación serial y la modalidad de funcionamiento de los pines. También se definen subrutinas, entre ellas la denominada *pos_accion()*, la cual posee tres opciones, cada una de ellas se define para un objeto y posición en particular. Una vez que se ha ingresado en una de las rutinas, el brazo se moverá hasta una posición determinada y comenzará la lectura de los sensores de presión gracias a la función *sensores()*. Si la tensión del sensor es superior al umbral, los dedos de la mano se cerrarán 10 grados, dicha acción se repetirá hasta que la tensión disminuya por la presión y no se cumpla la condición. Luego el brazo moverá el objeto hasta el destino programado, dejará el objeto y volverá a su estado de reposo definida en la función *pos_inicial()*.

Listado E-2: Script Arduino para realizar la adquisición de datos de los sensores de presión por medio de programa PLX-DAQ

```

1  #include <Servo.h>
2  #include <L298N.h>
3  #define EN 7
4  #define IN1 6
5  #define IN2 5
6
7  //Creando Actuadores
8  Servo dedos, mano, muneca, hombro;
9
10
11 //Posición inicial de los actuadores
12 int pos_dedos = 50;
13 int pos_mano = 10;
14 int pos_muneca = 0;
15 int pos_hombro = 180;
16
17 //Pines de cada actuador
18 int pin_dedos = 10;
19 int pin_mano = 11;
20 int pin_muneca = 12;
21 int pin_hombro = 13;
22 int pin_activ_cod = 7;
23 int pin_sub_cod = 6;
24 int pin_baj_cod = 5;
25
26 //Pines de cada sensor
27 const int pin_sensor1 = A0;
28 const int pin_sensor2 = A1;
29 const int pin_sensor3 = A2;
30 const int pin_sensor4 = A3;
31
32 float Sensor1 = 0, Sensor2 = 0, Sensor3 = 0, Sensor4 = 0;
33
34 unsigned short velocidad = 0;
35 String a = "";
36
37 void setup()
38 {
39   Serial.begin(9600);
40   pinMode(EN, OUTPUT);
41   pinMode(IN1, OUTPUT);
42   pinMode(IN2, OUTPUT);
43 }
44
45 void loop()
46 {
47   pos_inicial();
48   pos_accion();
49 }
50
51 void pos_inicial()
52 {
53   parametros();
54   setDir(2);
55   slide(10);
56   dedos.write(pos_dedos);
57   delay(1000);
58   muneca.write(pos_muneca);
59   delay(1000);
60   hombro.write(pos_hombro);
61   delay(1000);
62   mano.write(pos_mano);
63   delay(1000);
64   dedos.detach();
65   mano.detach();
66   muneca.detach();
67   hombro.detach();

```

```
68 }
69
70 void pos_accion()
71 {
72     while(1)
73     {
74         if(Serial.available())
75         {
76             int data = Serial.read();
77             if (data == '1')
78             {
79                 Serial.println("...Iniciando Acción...");
80                 parametros();
81                 mano.write(180);
82                 muneca.write(90);
83                 delay(1000);
84                 dedos.write(180);
85                 hombro.write(90);
86                 delay(2000);
87                 setDir(1);
88                 slide(8);
89                 delay(2000);
90                 mano.write(90);
91                 delay(3000);
92                 sensores();
93                 delay(1000);
94                 mano.write(80);
95                 setDir(2);
96                 slide(16);
97                 hombro.write(130);
98                 delay(2000);
99                 mano.write(90);
100                setDir(1);
101                slide(11);
102                dedos.write(180);
103                delay(2000);
104                mano.write(180);
105                Serial.println("...Acción finalizada...");
106                break;
107            }
108            if (data == '2')
109            {
110                parametros();
111                mano.write(180);
112                muneca.write(90);
113                delay(1000);
114                dedos.write(180);
115                hombro.write(90);
116                delay(5000);
117                mano.write(45);
118                delay(2000);
119                dedos.write(20);
120                delay(2000);
121                mano.write(100);
122                setDir(2);
123                slide(10);
124                hombro.write(125);
125                delay(2000);
126                mano.write(70);
127                setDir(1);
128                slide(8);
129                dedos.write(180);
130                delay(2000);
131                mano.write(180);
132                break;
133            }
134            if (data == '3')
135            {
136                parametros();
137                mano.write(180);
138                muneca.write(90);
```

```
139     delay(1000);
140     dedos.write(180);
141     hombro.write(90);
142     delay(2000);
143     setDir(1);
144     slide(12);
145     delay(2000);
146     mano.write(70);
147     delay(3000);
148     dedos.write(20);
149     delay(2000);
150     mano.write(80);
151     setDir(2);
152     slide(20);
153     hombro.write(115);
154     delay(2000);
155     mano.write(70);
156     setDir(1);
157     slide(11);
158     dedos.write(180);
159     delay(2000);
160     mano.write(180);
161     break;
162   }
163 }
164 }
165 }
166
167 void setDir(int d)
168 {
169   switch(d)
170   {
171     case 0: //APAGADO
172       digitalWrite(IN1, LOW);
173       digitalWrite(IN2, LOW);
174       break;
175     case 1: //SUBIR
176       digitalWrite(IN1, HIGH);
177       digitalWrite(IN2, LOW);
178       break;
179     case 2: //BAJAR
180       digitalWrite(IN1, LOW);
181       digitalWrite(IN2, HIGH);
182       break;
183     case 3: //BLOQUEADO
184       digitalWrite(IN1, HIGH);
185       digitalWrite(IN2, HIGH);
186       break;
187   }
188 }
189
190 void slide(int d)
191 {
192   for(int v=0; v<256; ++v)
193   {
194     analogWrite(EN,v);
195     delay(d);
196   }
197   for(int v=255; v>=0;--v)
198   {
199     analogWrite(EN,v);
200     delay(d);
201   }
202 }
203
204 void parametros()
205 {
206   dedos.attach(pin_dedos);
207   mano.attach(pin_mano);
208   muneca.attach(pin_muneca);
209   hombro.attach(pin_hombro);
```

```
300 }
301
302 void sensores()
303 {
304   Serial.println("...Leyendo sensores...");
305   int a=100;
306   delay(100);
307   Sensor1=analogRead(A0);
308   Serial.print("Sensor1 : ");
309   Serial.println((Sensor1*5)/1024);
310
311   Sensor2=analogRead(A1);
312   Serial.print("Sensor2 : ");
313   Serial.println((Sensor2*5)/1024);
314
315   Sensor3=analogRead(A3);
316   Serial.print("Sensor3 : ");
317   Serial.println((Sensor3*5)/1024);
318
319   Sensor4=analogRead(A4);
320   Serial.print("Sensor4 : ");
321   Serial.println((Sensor4*5)/1024);
322
323   Serial.println("...Cerrando gripper...");
324   while(Sensor1 > 1.5)
325   {
326     Sensor1=analogRead(A0);
327     Sensor1=Sensor1*5/1024;
328     Serial.print("Sensor1 : ");
329     Serial.print(Sensor1);
330     Serial.println("Volts");
331     dedos.write(a);
332     a = a -10;
333     delay(1000);
334   }
335   Serial.println("...Umbral alcanzado...");
336   delay(100);
337   Serial.println("...Moviendo...");
338 }
```

F Instalación de roserial arduino

Rosserial es un protocolo que permite extender las capacidades de mensajería de ROS través de una interfaz serial. Antes de realizar la instalación de *roserial* ya debe tener instalada alguna versión de ROS. Para este proyecto se recomienda el uso de *ROS Kinetic*. Las instrucciones de instalación las puede encontrar en la dirección <http://wiki.ros.org/>, donde se explica paso a paso dicho proceso. Para realizar la instalación de *roserial* a través de Git debe tener instalado previamente el IDE de Arduino,

En una ventana terminal debe ir al directorio local de trabajo y ejecutar los siguientes comandos que se muestran a continuación:

```
$ sudo apt-get
```

```
$ sudo apt-get upgrade
```

```
$ sudo apt-get install ros-kinetic-roserial-arduino
```

```
$ sudo apt-get install ros-kinetic-roserial
```

Posteriormente debe instalar las librerías *ros_lib* de la siguiente forma:

```
$ cd Arduino/libraries/
```

```
$ rosruntime roserial_arduino make_libraries.py.
```

Es muy importante respetar el espacio y el punto final. Luego de realizar esta acción debería tener una carpeta llamada *ros_lib* en el directorio. También puede validarlo navegando a través del IDE de Arduino ingresando al menú “Archivo”, luego seleccionar “Ejemplos” donde encontrará “*ros_lib*”.

Si realizó correctamente todos los pasos su placa *Arduino* o *Picaro MEGA* ya está en condiciones para ser configurada como un *Nodo*. Para tener mayor claridad de este tema es recomendable que realice los tutoriales disponibles en la dirección http://wiki.ros.org/roserial_arduino/Tutorials/.