



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT RECOLECTOR DE PELOTAS DE TENIS DE CAMPO MEDIANTE VISIÓN ARTIFICIAL”

TRABAJO DE TITULACIÓN TIPO: PROYECTO TÉCNICO

Presentado para optar al grado académico de:

INGENIERA EN ELECTRÓNICA CONTROL Y REDES INDUSTRIALES

AUTOR: CAREN GABRIELA GUERRERO PILCO
DIRECTOR: ING. JOSÉ LUIS TINAJERO LEÓN

Riobamba – Ecuador

2020

©2020, Caren Gabriela Guerrero Pilco

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Caren Gabriela Guerrero Pilco, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autora asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 21 de enero de 2020.



Caren Gabriela Guerrero Pilco

0604963025

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES**

El tribunal del trabajo de titulación certifica que: El trabajo de titulación: Tipo: Proyecto Técnico, “**DISEÑO E IMPLEMENTACIÓN DE UN ROBOT RECOLECTOR DE PELOTAS DE TENIS DE CAMPO MEDIANTE VISIÓN ARTIFICIAL.**”, de responsabilidad de la señorita **CAREN GABRIELA GUERRERO PILCO**, ha sido minuciosamente revisado por los miembros del tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

FIRMA

FECHA

Ing. José Luis Morales Gordón

PRESIDENTE DEL TRIBUNAL

Ing. José Luis Tinajero León

**DIRECTOR DEL TRABAJO DE
TITULACIÓN**

Ing. Pablo Eduardo Lozada Yáñez

MIEMBRO DEL TRIBUNAL

DEDICATORIA

El presente trabajo de titulación se lo dedico en primer lugar a mis padres Roberto Guerrero y Yolanda Pilco quienes con su lucha y sacrificio han permitido cumplir un sueño más, por ser la motivación y el apoyo constante en cada paso de mi vida, por su amor y cariño incondicional. A mi hermana Fátima Guerrero, que a pesar de su corta edad ha sabido brindarme su apoyo y su amistad. A mis abuelitos Rosa y Manuel, que siempre estuvieron para brindarme palabras de aliento. A mis familiares que han sabido brindarme su apoyo y consejo.

Dedicado también a Mariuxi, Luz María y Alfredo que, aunque físicamente no los pueda ver sé que desde el cielo comparten la felicidad de este logro.

Caren

AGRADECIMIENTO

Agradezco a Dios por haberme dado la vida y el entendimiento para poder aceptar sus designios. A mi madre Yolanda que ha estado presente en cada paso de mi vida quien con su amor ha sabido guiarme por el buen camino. A mis amigas Viviana, María Gabriela y Gabriela, por su apoyo incondicional, por sus palabras de ánimo para seguir adelante aun cuando los tiempos sean difíciles. A la persona con quien compartí buenos y malos momentos durante varios años de mi vida. A los ingenieros José Luis Tinajero, Pablo Lozada y Henry Vallejo, quienes con su apoyo incondicional han hecho posible la culminación del presente trabajo de titulación.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xiii
ÍNDICE DE ANEXOS.....	xvi
ÍNDICE DE ABREVIATURAS.....	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
INTRODUCCIÓN	1

CAPITULO I

1.	MARCO TEÓRICO	6
1.1.	METODOLOGÍA IMPLEMENTADA EN EL DISEÑO.	6
1.1.1.	<i>Sistemas odométricos</i>	6
1.1.1.1.	<i>Análisis odométrico</i>	7
1.1.2.	<i>Visión artificial</i>	12
1.1.2.1.	<i>Preprocesamiento por segmentación</i>	13
1.2.	ANÁLISIS DEL SOFTWARE PRINCIPAL IMPLEMENTADO EN EL DISEÑO.	16
1.2.1.	<i>MATLAB</i>	16
1.2.2.	<i>LabView</i>	17
1.2.3.	<i>OPEN CV</i>	17
1.3.	ANÁLISIS DEL HARDWARE PRINCIPAL DEL SISTEMA DE RECOLECCIÓN.	19
1.3.1.	<i>Configuración de los Robots Móviles</i>	19
1.3.1.1.	<i>Configuración de Rueda convencional y giratoria</i>	19
1.3.1.2.	<i>Configuración Direccional y Ackerman</i>	20
1.3.1.3.	<i>Configuración Omni Wheel universal</i>	21
1.3.1.4.	<i>Configuración Rueda Mecanum</i>	22
1.3.2.	<i>Aplicaciones de robots móviles</i>	23
1.3.2.1.	<i>SR1 Robot Móvil Multifuncional</i>	23
1.3.2.2.	<i>Robot móvil basado en Arduino</i>	24
1.3.2.3.	<i>TI robotics systems</i>	25
1.3.3.	<i>Codificadores (Encoders)</i>	26
1.3.3.1.	<i>Lectura de encoders</i>	27

1.3.3.2.	<i>Tipos de codificadores.</i>	28
1.3.4.	<i>Cámaras</i>	29
1.3.4.1.	<i>Cámaras Monocromáticas</i>	29
1.3.4.2.	<i>Cámara estereoscópica.</i>	30
1.3.5.	<i>Tarjetas de desarrollo</i>	30
1.3.5.1.	<i>Tarjeta microcontroladora Arduino</i>	31
1.3.5.2.	<i>Raspberry Pi</i>	31
1.3.5.3.	<i>Intel Galileo</i>	32

CAPITULO II

2.	MARCO METODOLÓGICO	34
2.1.	CONCEPCIÓN DEL PROTOTIPO DEL SISTEMA DEL ROBOT RECOGEDOR	34
2.1.1.	<i>Selección del hardware que conforma el prototipo</i>	34
2.1.1.1.	<i>Arduino Nano</i>	35
2.1.1.2.	<i>Módulo L298N puente H L298</i>	36
2.1.1.3.	<i>Motor reductor</i>	37
2.1.1.4.	<i>Módulo NRF24l01</i>	38
2.1.1.5.	<i>Kinetic X-Box</i>	39
2.1.2.	<i>Programación del sistema</i>	40
2.1.2.1.	<i>Etapas de Programación en Matlab</i>	41
2.1.2.2.	<i>Obtención de imágenes</i>	41
2.1.2.3.	<i>Tratamiento de imágenes</i>	43
2.1.2.4.	<i>Algoritmo de Detección de Pelotas y localización</i>	44
2.1.2.5.	<i>Algoritmo de Navegación</i>	48
2.1.2.6.	<i>Programación Arduino</i>	49
2.1.2.7.	<i>Librería Arduino</i>	52
2.1.3.	<i>Interfaz del usuario en Matlab</i>	53
2.2.	ESQUEMA DE CONEXIÓN DEL SISTEMA	57
2.2.1.	<i>Pines de conexión Robot y Receptor (Arduino Nano)</i>	58
2.2.2.	<i>Pines de conexión Emisor (Arduino Uno)</i>	59
2.3.	DISEÑO EN DEL PROTOTIPO EN SOLIDWORKS	59

CAPITULO III

3.	VALIDACIÓN DEL PROTOTIPO	62
3.1.	PRUEBAS DE DETECCIÓN DE PELOTAS	63

3.2.	PRUEBAS DE POSICIONAMIENTO DE PELOTAS	64
3.3.	PRUEBAS DE RECOLECCIÓN.....	67
3.4.	PRUEBAS DE TIEMPO DE RECOLECCIÓN.....	68
3.5.	ANÁLISIS DE COSTOS	69
CONCLUSIONES		71
RECOMENDACIONES		72
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-1.	Factores de consideración entre errores sistemáticos y no sistemáticos.....	7
Tabla 2-1.	Ventajas y Desventajas en los softwares de desarrollo.....	18
Tabla 3-1.	Tabla comparativa de las diferentes configuraciones.....	22
Tabla 4-1.	Parámetros generales de las ruedas.....	23
Tabla 5-1.	Secuencia de bits generado por los encoders.....	27
Tabla 6-1.	Disposición de conexión del encoder.....	27
Tabla 7-1.	Comparación de los sensores.....	28
Tabla 1-2.	Ventajas y desventajas de Arduino nano.....	36
Tabla 2-2.	Ventajas y Desventajas del módulo L298N.....	37
Tabla 3-2.	Ventajas y desventajas del motor-reductor.....	38
Tabla 4-2.	Ventajas y desventajas del Módulo NRF24l01.....	39
Tabla 5-2.	Ventajas y desventajas del Kinetic Xbox 360.....	40
Tabla 6-2.	Tabla de objetos circulares y su área.....	45
Tabla 7-2.	Tabla conexiones robot.....	58
Tabla 8-2.	Tabla Conexiones Emisor.....	59
Tabla 9-2.	Medidas de la parte superior.....	60
Tabla 10-2.	Medidas de la parte lateral.....	61
Tabla 1-3.	Valores obtenidos por las pruebas realizadas en campo.....	63
Tabla 2-3.	Errores en las posiciones del vector en coordenadas X e Y.....	65
Tabla 3-3.	Datos obtenidos de eficiencia en cada escenario.....	68
Tabla 4-3.	Costos y descripción de materiales.....	69

ÍNDICE DE FIGURAS

Figura 1-1.	Parámetros de ubicación del robot	7
Figura 2-1.	Triangulación para el cálculo de	8
Figura 3-1.	Diagrama de flujo del principio odométrico	11
Figura 4-1.	Diagrama del cálculo de velocidad.....	12
Figura 5-1.	Algoritmos de detección de rostros	13
Figura 6-1.	Segmentación de una imagen.....	14
Figura 7-1.	Escalamiento de algoritmo SIFT y SURF.....	15
Figura 8-1.	Detección de variaciones en una imagen tomada en tiempos distintos.	15
Figura 9-1.	Ventana de inicio de Matlab.	16
Figura 10-1.	Ventana de inicio de LabView.....	17
Figura 11-1.	Ventana de OpenCV.....	18
Figura 12-1.	Tipos de ruedas: (a) un volante convencional /, (b1) una rueda giratoria, (b2) una rueda giratoria, (c) una rueda universal, (d) una rueda Mecanum.....	20
Figura 13-1.	(a) una vista genérica de la configuración, (b) una configuración de manejo de Ackerman, (c) una configuración casi omni drive, (d) Configuración direccional.	21
Figura 14-1.	Configuraciones de rueda universal: (a) base triangular; (b) base cuadrada, las ruedas están a los lados; (c) base cuadrada, las ruedas están en las esquinas.....	21
Figura 15-1.	Configuración de ruedas Mecanum	22
Figura 16-1.	Robot SR1 móvil multifuncional.....	24
Figura 17-1.	Robot móvil de la plataforma ARDUINO.25	
Figura 18-1.	Robot Stellaris from Texas Instruments.....	25
Figura 19-1.	Partes principales de un encoder.....	26
Figura 20-1.	Tipos de discos de los codificadores.....	26
Figura 21-1.	Esquema de conexión de un encoder.....	28
Figura 22-1.	Cámara monocromática.....	29
Figura 23-1.	Cámara estereoscópica.....	30
Figura 24-1.	Arduino Mega, Leonardo, Uno y Nano respectivamente.....	31
Figura 25-1.	Microcontrolador Raspberry Pi B.....	32
Figura 26-1.	Tarjeta Intel Galileo.....	33
Figura 1-2.	Arduino Nano.....	35
Figura 2-2.	Puente H L298N.....	37

Figura 3-2.	Motorreductor.....	38
Figura 4-2.	Módulo -nRF14l01.....	39
Figura 5-2.	Estructura exterior de un Kinetic.....	40
Figura 6-2.	Esquema final del funcionamiento.....	41
Figura 7-2.	Imágenes preliminares obtenidas en por el Kinetic.....	42
Figura 8-2.	Conexión Kinect con Matlab.....	42
Figura 9-2.	Tratamiento de las imágenes (Imagen Real, Imagen HSV, Imagen Segmentada).....	43
Figura 10-2.	Filtrado y selección de imágenes.....	43
Figura 11-2.	Diagrama de reconocimiento y localización de pelotas.....	45
Figura 12-2.	Esquema para encontrar las coordenadas del objeto.....	46
Figura 13-2.	Diagrama de flujo sistema de navegación.....	49
Figura 14-2.	Diagrama de control Arduino Receptor (Robot).....	51
Figura 15-2.	Diagrama del Emisor.....	52
Figura 16-2.	Monitor Serial Tarjeta desarrollada del robot.....	53
Figura 17-2.	Vista principal de la pantalla de usuario.....	54
Figura 18-2.	Vista 2 botón inicio.....	54
Figura 19-2.	Vista 3 botón de paro.....	55
Figura 20-2.	Vista de la cámara en tiempo real.....	55
Figura 21-2.	Vista de ruta que se debe trazar.....	56
Figura 22-2.	Vista de descripción de coordenadas.....	56
Figura 23-2.	Vista 7 de la pantalla de usuario.....	57
Figura 24-2.	Esquema Electrónico del Robot +Receptor.....	57
Figura 25-2.	Esquema Electrónico Emisor.....	58
Figura 26-2.	Vista del prototipo en SolidWorks.....	59
Figura 27-2.	Vista Superior.....	60
Figura 28-2.	Vista Lateral Izq.....	60
Figura 29-2.	Vista General Isométrica.....	61
Figura 30-2.	Robot Ensamblado.....	61
Figura 1-3.	Mediciones de la distancia del objeto con respecto al robot.....	67

ÍNDICE DE GRÁFICOS

Gráfico 1-3.	Escala de tiempo vs costo del desarrollo del equipo.....	62
Gráfico 2-3.	Eficiencia del robot en distintos tiempos	64
Gráfico 3-3.	Comparación de posiciones real vs medida en el eje X.....	66
Gráfico 4-3.	Comparación de posiciones real vs medida en el eje Y.....	66
Gráfico 5-3.	Valores porcentuales promedios de eficiencia en cada escenario.....	69

ÍNDICE DE ANEXOS

ANEXO A: Especificaciones en Arduino Nano

ANEXO B: Puente H L298N

ANEXO C: Características del motorreductor

ANEXO D: Módulo nRF21L01

ANEXO E: Características del Kinetic

ANEXO F: Pruebas del Robot

ÍNDICE DE ABREVIATURAS

COI:	Comité Olímpico Internacional
SIFT:	Scale-invariant feature transform
SURF:	Speeded Up Robust Features
DoG:	Difference of Gaussians
BSD:	Berkeley Software Distribution
API:	Application Programming Interface
SO:	Sistema Operativo
DW:	Deutsche Welle
DC:	Direct Current
DoF:	Degree of Freedom
RF:	Radio Frecuencia
IC:	Circuito Integrado
E/S:	Entrada/Salida
SO:	Sistema Operativo
SPI:	Serial Peripheral Interface
IR:	Infrarojo
IMU:	Inertial Measurement Unit
I2C:	Inter Integrated Circuits
cm:	Centímetros

RESUMEN

Este trabajo de titulación tuvo como objetivo la implementación de un prototipo de robot para la recolección de pelotas de tenis aplicando métodos de visión artificial. El robot fue construido a partir de un microcontrolador Arduino Nano, una antena de radiofrecuencia NRF24L01 el cual se encarga de receptor datos de la etapa de control y es sistema de locomoción integrado por motores y un sistema de odometría que consta de un giroscopio MPU6050 y encoders, además de un módulo emisor conformado por un Arduino Uno y una antena de las mismas características. El sistema de control consta de tres partes, la primera parte dedicada al algoritmo de detección de pelotas mediante segmentación por color y forma, las imágenes se obtuvieron mediante una cámara Kinect 360° y el entorno de desarrollo Matlab. La segunda etapa se encarga de la localización de las pelotas en coordenadas rectangulares dadas en centímetros, con estos datos y mediante pruebas realizadas dieron como resultado una precisión del 99.088% en el eje X y 98.264% en el eje Y las cuales indican un rango aceptable del funcionamiento. La última etapa consta del envío de datos desde Matlab hacia el robot, al recibir dicha información el robot se dirige hacia el objeto describiendo trayectorias polares, es decir, gira un ángulo dependiente de las coordenadas y avanza una magnitud, cabe recalcar que el robot es monitorizado todo el tiempo por medio del sistema odométrico y la cámara. Para la evaluación del sistema se obtuvo una precisión de recolección del 96,9% considerando que el sistema se puso a prueba en un campo real y con varios distractores en cada prueba logrando así crear un sistema de alta eficiencia en la recolección de pelotas de tenis de campo.

PALABRAS CLAVE: <VISIÓN ARTIFICIAL>, <RECOLECCION DE PELOTAS DE TENIS>, <ALGORITMO DE DETECCION >, <ADQUISICIÓN DE IMÁGENES>, <PROCESAMIENTO DE IMÁGENES>, <SEGMENTACION POR COLOR>, <SEGMENTACION POR FORMA>, <MATLAB (SOFTWARE)>, <CÁMARA KINECT (HARDWARE)>.



ABSTRACT

This degree work is aimed at implementing a robot prototype for tennis balls collection by using artificial vision methods. The robot was built from an Arduino Nano microcontroller, a radio frequency antenna NRF24L01, which is responsible for receiving data from the control stage, a locomotion system integrated by engines and an odometer system consisting of an MPU6050 gyroscope and encoders, in addition to a transmitter module consisting of an Arduino Uno and an antenna with the same characteristics. The control system consists of three parts, the first part dedicated to the balls detection algorithm through color and shape segmentation, the images were obtained by means of a Kinect 360⁰ camera and the Matlab environmental development. The second stage is responsible for locating the balls in rectangular coordinates given in centimeters, with these data and through tests performed the results were provided with an accuracy of 99.088% on the X axis and 98.264% on the Y axis which indicate an acceptable operation range. The last stage consists of sending data from the Matlab program to the robot, upon receiving such information the robot tries to look for the object describing polar trajectories, that is, it rotates an angle dependent on the coordinates and moves around a magnitude, it should be noted that the robot is monitored all the time by means of the odometer system and the camera. For the system assessment, a collection accuracy of 96.9% was obtained taking into consideration that the system was tested in a real bonnet and with several distractors in each test, thus creating a high efficiency system in the collection of tennis balls out on the field.

KEY WORDS: <ARTIFICIAL VISION>, <TENNIS BALL COLLECTION>, <DETECTION ALGORITHM>, <IMAGE ACQUISITION>, <IMAGE PROCESSING>, <COLOR SEGMENTATION>, <FORM SEGMENTATION>, <MATLAB (SOFTWARE)>, <KINECT CAMERA (HARDWARE)>.



INTRODUCCIÓN

Con el desarrollo continuo de la electrónica e inserción de herramientas como sensores y actuadores en muchas máquinas ya creadas se pretende probar y buscar diferentes métodos para poder facilitar las actividades cotidianas que el hombre realiza en diferentes campos dependiendo de su actividad. El uso de máquinas para realizar tareas, es un medio que poco a poco va teniendo aceptación en la sociedad dentro de áreas donde las cualidades del hombre pueden llegar a fallar como en áreas donde la precisión o repetición son de vital importancia, las equivocaciones mejor conocidas como “error humano” son habituales en estos campos muchas veces creadas por agotamiento. Por lo tanto, en este prototipo tecnológico del recolector de pelotas en un campo de tenis utilizando visión artificial es el proceso que se ejecutará. Con la utilización de un robot inteligente reutilizable que sorteará la cancha en busca de pelotas cuando sea necesario es como se planea brindar ayuda al encargado de efectuar esta actividad repetitiva.

ANTECEDENTES

Inventado en 1873 por Walter Wingfield el tenis es un deporte que se desarrolla al aire libre de manera individual (1 versus 1) o en parejas (2 vs 2) en una cancha rectangular de 23.77 x 8.23 m en individual y de 23.77 x 10.97 m en dobles. (BENAVIDES, 2016, p.2) Consiste en golpear una pelota de cuero de aproximadamente 6.7 cm de diámetro sobre una red situada en el centro de la cancha asegurando que rebote en un cuadrado que se encuentra en diagonal en el lado rival y sumando puntos cada vez que no se consiga esta condición. (International Tennis Federation, Reglas del tenis, 2017, p.5)

Estuvo en el programa olímpico de los Juegos de la I Olimpiada en Atenas en 1896, hasta los Juegos de la VIII Olimpiada en París en 1924. Luego se retiró del programa debido a una diferencia de opinión entre el COI y la Federación Internacional de Tenis e hizo su regreso olímpico como deporte de demostración en 1968 y 1984 y ha aparecido en el programa olímpico desde los Juegos de la XXIV Olimpiada en Seúl en 1988. (TENNIS, History of Tennis at the Olympic Games, 2017, p.2)

Además, practicar un deporte es el mejor método para llevar una vida sana y activa. Actualmente, el incentivo a los mejores deportistas ha llevado a que la competencia en un área específica se convierta en una actividad cotidiana para poder crear una carrera en ello y por su puesto la práctica

de Tenis se encuentra en la categoría de mayor exigencia con respecto a dedicación, técnica, resistencia y disciplina en el mundo.

Adicional a los jugadores en la cancha, existen personas cumpliendo una determinada tarea que en conjunto hacen que la práctica de este deporte se lleve a cabo de la mejor manera posible, tal es el caso de la posición de los jueces de silla, de red, de servicio, de línea, central de línea, de fondo de campo y el jugador de balón. (SportReglas, “TODO SOBRE TENIS”, 2018, p.1)

El jugador de balón es aquella persona encargada de recoger toda aquella pelota de tenis que se encuentre dispersa en la cancha, esta persona debe recorrer la cancha recogiendo una a una las pelotas y entregárselas al jugador que le corresponda, finalmente esta actividad además de ser repetitiva resulta ser tediosa para aquella persona encargada de ello. Se han desarrollado equipos que brindan ayuda al encargado ya sea de manera total o parcial. (Tabares, 2015, p.12)

Actualmente la tecnología ha incursionado en muchos campos de la sociedad y el empleo de máquinas electrónicas para realizar actividades repetitivas y el campo del deporte en el área de tenis, ha sido tomado en cuenta en áreas donde el trabajo se torna tedioso y repetitivo, tal es el caso del empleo de máquinas que ayudan al jugador del balón a recoger las pelotas de un modo más sencillo y rápido

Desde máquinas manuales como Roll in usado de manera eficiente en la recolección de objetos sin la necesidad de inclinarse y asegurando una postura correcta en el cuerpo del encargado de las pelotas, (ROLL-IN RECOGEDOR MULTIUSOS, 2017, p.1) hasta máquinas totalmente autónomas como Tennibot que es un recolector de pelotas totalmente robótico y autónomo (Tennibot, 2018, p.1). Han sido utilizadas en campos de tenis para facilitar las tareas que el jugador del balón debe realizar con respecto al control de pelotas dispersas en la cancha.

JUSTIFICACIÓN

Justificación Teórica

Los recogedores de pelotas son máquinas que actualmente se encuentran en desarrollo, pero debido a su exclusivo uso y área de desempeño, son aparatos costosos y de difícil acceso, debido a los inconvenientes mencionados en este proyecto, lo que se propone realizar es un prototipo de recogedor automático de pelotas de tenis de campo utilizando equipos y elementos con

características de acceso sencillo y costos accesibles, en comparación con la marca más reconocida dentro del campo como es TENNIBOT y su robot recogedor basado en roomba.

Para la elaboración del dispositivo se tomaron en cuenta tres parámetros, el primero es el dispositivo encargado de sortear la cancha de tenis, el segundo es la visión artificial por medio de una cámara que cubrirá la cancha con la finalidad de encontrar el pixel en una coordenada específica en la imagen que determinará la presencia de pelotas en el campo y finalmente el accionador que recogerá cada una de las pelotas sin llegar a soltarla.

Dentro del campo de juego, existen varios elementos contenedores de pelotas como: Bolsa de tenis, Cesta de tenis de mano, Cesta de tenis con ruedas, Carro de tenis, Cesta recogedora, Cubo de tenis, Tubos recogedores, Rodillos recogedores. (El foro del Tenis, “Contenedor de pelotas de tenis para entrenamientos”, 2016, p.1)

Actualmente los tenistas amateurs o profesionales utilizan uno de estos elementos mencionados, los cuales requieren de una intervención constante del hombre convirtiendo a cada tipo de contenedor en un elemento totalmente dependiente, haciendo que la tarea de recolectar las pelotas siga siendo rutinaria cansada y repetitiva además de mantener al encargado en posiciones no recomendadas para la salud del mismo. (Trujillo, 2015, p.8-9)

Lo que se consiguió con este proyecto fue implementar un sistema robótico de tracción por ruedas capaz de sortear una cancha de tenis con la finalidad de ubicar y recolectar pelotas dispersas en el campo, dichos puntos de ubicación fueron enviados por medio de un sistema de monitoreo encargado de reconocer un pixel establecido por el color de la pelota de tenis por medio de segmentación de imagen y video.

Justificación Aplicativa

El procedimiento de verificar que una cancha de tenis se encuentre despejada sin ningún tipo de pelotas en el campo se vuelve cansado, además de convertirse en una tarea incomoda y aburrida para los jugadores por lo que deben contratar personas encargadas de realizar esta acción y en las prácticas manteniendo un límite de pelotas que se pueden permitir en la cancha debido a que recogerlas conforme vayan apareciendo, llegaría a producir un mal entrenamiento debido a los continuos paros del juego mientras recogen cada pelota.

Por este motivo se desarrolló un robot capaz de recoger de manera automática las pelotas mediante un accionador en su parte frontal el cuál consigue retirar las pelotas del campo, así también llegando a recoger un número indeterminado de pelotas, es decir, el robot es capaz de recoger las todas las pelotas que se encuentren en el campo, mientras su único inconveniente llegaría a ser su capacidad de almacenamiento para cada tarea de recolección.

Dependiendo de la sección de la cancha el robot se moverá de manera automática a los puntos donde el pixel establecido por el programa y adquirido por la cámara se lo indique, será capaz de realizar la tarea de manera repetitiva las veces que el jugador lo requiera, garantizando así que la cancha sea apta para el juego o entrenamiento del deportista.

El monitoreo del campo y el uso de visión artificial permiten que el robot sea autónomo, al enviar los datos de ubicación de cada pelota y controlando la posición del robot en la cancha de tenis.

Actualmente el interés por el desarrollo se basa en recolectar pelotas, pero con la orientación adecuada puede ser un punto de partida para el desarrollo de sistemas robóticos que se encarguen de la recolección de distintos tipos de objetos, fomentando el desarrollo continuo en el prototipo y volviéndolo versátil para su uso en diferentes áreas donde las tareas aparte de tediosas resulten incluso peligrosas para los humanos. (Villegas & López, 2016, p.1)

OBJETIVOS

Objetivo General

Desarrollar un prototipo capaz de recolectar pelotas de tenis de campo mediante visión artificial luego de un entrenamiento.

Objetivos Específicos

- Realizar la investigación del método y cantidad de recolección de pelotas de tenis.
- Diseñar el prototipo más adecuado que cumpla con la recolección de pelotas de tenis.
- Implementar el prototipo encargado de recoger las pelotas de tenis fuera del área de la cancha.
- Evaluar el prototipo recolector de pelotas de tenis.

CAPÍTULO I

1 MARCO TEÓRICO

Dentro de las muchas actividades que deben ser tomadas en cuenta en el tenis de campo, la tarea que debe realizar el jugador del balón consiste en retirar todas las pelotas que se encuentren dentro del campo de tenis para poder llevar a cabo un juego ordenado y eficaz.

El uso de un robot autónomo como ayudante en el deporte, es un método que ha tenido una buena aceptación al momento de fusionar el deporte con la robótica. Por lo tanto, en este capítulo se mostrará una introducción respecto a los conceptos que fueron englobados en este trabajo tales como son robots móviles de desarrollo, tipos más comunes, visión artificial, sistemas de odometría, tarjetas de desarrollo, envío y recepción de información para la toma de decisiones, entre otros puntos que se deben considerar.

1.1 Metodología implementada en el diseño.

Para poder llevar a cabo el proyecto de manera eficiente es necesario analizar la metodología de funcionamiento que se llevara a cabo en el prototipo y al tener claro esto se puede proceder a analizar los requerimientos necesarios para efectuar la tarea que el robot debe realizar.

1.1.1 Sistemas odométricos

Al hablar de un robot técnicamente autónomo, es de vital importancia conocer el comportamiento que tendrá en el momento de realizar su desplazamiento a través de una trayectoria, por lo que el posicionamiento para determinar la localización del vehículo es una parte vital en el proyecto. Uno de los puntos que se debe tratar en este tipo de sistemas es la posición y orientación del robot cuyos datos serán adquiridos mediante las ruedas integradas en el robot, no obstante, el inconveniente que existe en este método es la continua acumulación de error cuando realiza la tarea de desplazamiento y que se va incrementando si la ruta de desplazamiento llega a ser demasiado grande. (Vaca, 2017, p.17)

Los errores en cuanto al desplazamiento pueden encontrarse y se los puede dividir en: errores sistemáticos y errores no sistemáticos, los mismos que se detallan en la Tabla 1-1.

Tabla 1-1. Factores de consideración entre errores sistemáticos y no sistemáticos.

Errores Sistemáticos	Errores no Sistemáticos
<ul style="list-style-type: none"> • Distinto tamaño en las ruedas • Diferencias en el radio estimado de las ruedas y su valor real • Resolución y precisión del encoder 	<ul style="list-style-type: none"> • Existe con coeficiente de fricción entre la rueda y el suelo donde se pretenda realizar la tarea • Separación de la rueda con la superficie (en casos donde la superficie no es uniforme)

Fuente: (Ramírez, 2017)

Realizado por: Guerrero, Caren, 2020

1.1.1.1 Análisis odométrico

El robot realiza movimientos de dos dimensiones X, Y basados en su orientación angular por lo tanto su ubicación tendrá un principio mostrado en la Figura 1-1.

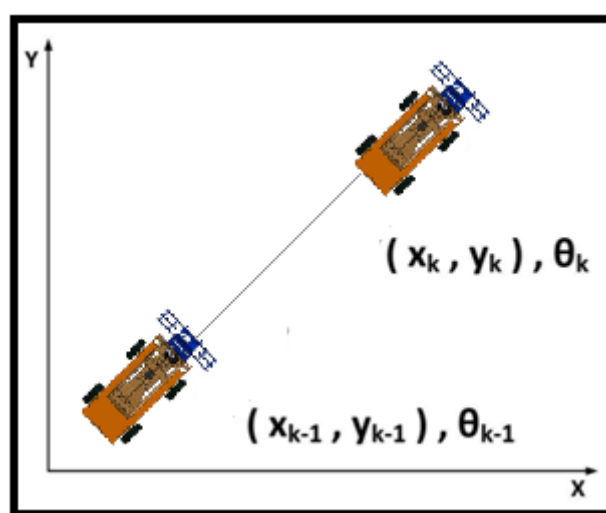


Figura 1-1. Parámetros de ubicación del robot.

Realizado por: Guerrero, Caren, 2020

Con la triangulación se puede determinar la escala de la imagen como se observa en la Figura 2-1, utilizando la Ecuación 1-1 se puede determinar el escalonamiento de imágenes.

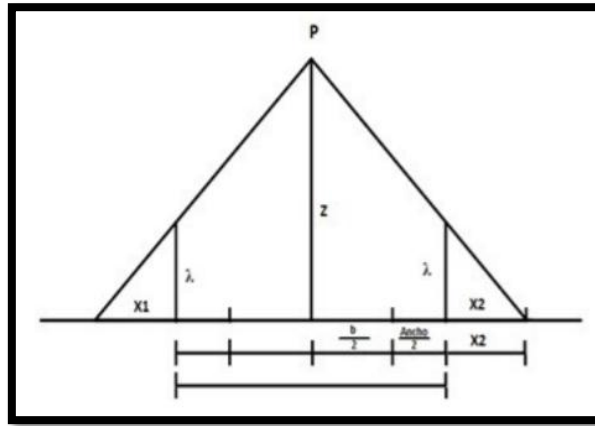


Figura 2-1. Triangulación para el cálculo de profundidad de la imagen

Realizado por: Guerrero, Caren, 2020

Ecuación 1-1. Ecuación para el escalamiento de imágenes

$$Escala = \frac{D \times F}{X_i \times X_d}$$

Donde,

D = distancia que existe entre los centros ópticos.

F = Foco.

Xi = posición izquierda.

Xd = posición derecha.

Entonces con esta información lo que se procedió a realizar fue obtener una nube de puntos con la finalidad de leer las coordenadas generadas en el procesamiento de imágenes y guardar estos datos en un archivo de Excel para poder realizar su manipulación desde Matlab. (Cruzado Muñoz, 2017, p.28-30)

Finalmente, una vez obtenida la imagen y enviada hacia el robot el movimiento se basa en los principios de la odometría mencionada para lo cual se basó en las ecuaciones deducidas para el uso de robots móviles odométricos en la Ecuación 2-1 la cual nos indica el diferencial de ángulo que existe entre las llantas. (Cruzado Muñoz, 2017, p.30)

Ecuación 2-1. Ecuación para hallar el diferencial del ángulo del robot

$$\Delta\theta = \frac{2\pi \times R_w}{T_R \times D} \times (T_i - T_d)$$

Donde,

Rw = Radio de la llanta del encoder.

D = Distancia entre las llantas.

Ti = Cantidad de pulsos registrados por el encoder ubicado en la izquierda.

Td = Cantidad de pulsos registrados por el encoder ubicado en la derecha.

Tr = Cantidad total de registros en una revolución.

Con la variación del ángulo ya obtenida se procede a calcular la variación de la posición en el eje X dado por la Ecuación 3-1.

Ecuación 3-1. Ecuación para hallar el diferencial de la posición X del robot.

$$\Delta x = \frac{\pi \times R_w}{T_R} \times (T_i - T_d) \times \cos(\theta)$$

Donde,

Rw = Radio de la llanta del encoder.

D = Distancia entre las llantas.

Ti = Cantidad de pulsos registrados por el encoder ubicado en la izquierda.

Td = Cantidad de pulsos registrados por el encoder ubicado en la derecha.

Tr = Cantidad total de registros en una revolución.

Para obtener la variación de la siguiente coordenada en eje Y esta viene dominada por la Ecuación 4-1.

Ecuación 4-1. Ecuación para hallar el diferencial de la posición Y del robot.

$$\Delta y = \frac{\pi \times R_w}{T_R} \times (T_i - T_d) \times \sin(\theta)$$

Donde,

Rw = Radio de la llanta del encoder.

D = Distancia entre las llantas.

Ti = Cantidad de pulsos registrados por el encoder ubicado en la izquierda.

Td = Cantidad de pulsos registrados por el encoder ubicado en la derecha.

Tr = Cantidad total de registros en una revolución.

Con la variación de las posiciones obtenidas se definen los valores de ángulo actual que es la suma del ángulo anterior más la variación de ángulo existente por las llantas en un instante de tiempo la cual se expresa mediante la Ecuación 5-1.

Ecuación 5-1. Ecuación para hallar el ángulo actual del robot.

$$\theta_k = \theta_k + \Delta\theta$$

Donde,

θ_k = Angulo anterior del robot.

$\Delta\theta$ = Diferencial de Angulo.

De la misma manera que sucede con el ángulo actual para hallar la posición actual del robot en el espacio se aplican la Ecuación 6-1 y la Ecuación 7-1 para las coordenadas X, Y respectivamente.

Ecuación 6-1. Ecuación para hallar la posición actual del robot en el eje X

$$x_k = x_{k-1} + \Delta x$$

Donde,

x_k = Posición X anterior del robot.

Δx = Diferencial de posición X.

Ecuación 7-1. Ecuación para hallar la posición actual del robot en el eje Y.

$$y_k = y_{k-1} + \Delta y$$

Donde,

y_k = Posición Y anterior del robot.

Δy = Diferencial de posición Y.

Este listado de ecuaciones fue implementado mediante el diagrama de flujo que se muestran en la Figura 3-1.

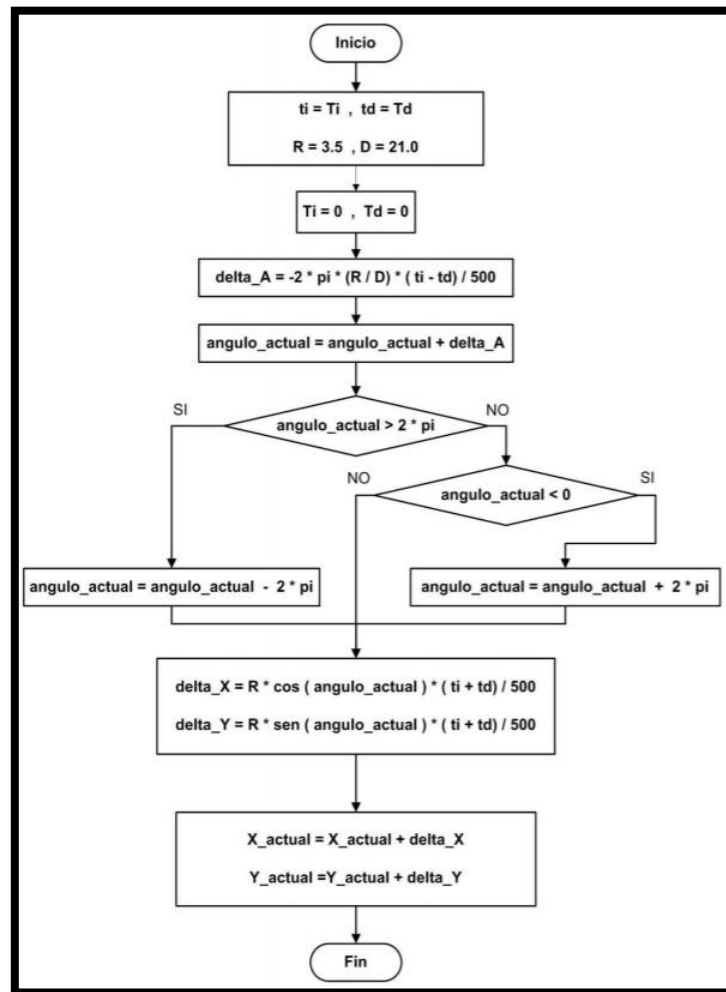


Figura 3-1. Diagrama de flujo del principio odométrico.

Fuente: (Zea & Alberto, 2015, p.64)

Con la lectura de “ti” y “td” se evaluarán los encoders derecho e izquierdo, estas variables son iniciadas con el valor de las variables globales “Td” y “Ti” cuya función tiene actualizado los ticks de los encoders. (Arellano, 2015, p.65)

Continuando con la variable R cuyo valor será del radio de las ruedas acopladas y “D” que será la distancia entre las ruedas, posteriormente se reinicia el contador de “Ti” y “Td” para poder contar los ticks y evitar la acumulación de datos. (Arellano, 2015, p.65)

Al obtener la variación angular por medio de la Ecuación 5-1 se obtiene el ángulo actual el cual se asigna a delta A, después para operar dentro de 0° a 360° se ejecutará la calibración de vuelta, (Por lo que si el valor actual es mayor a 360° se restará una vuelta y si el ángulo actual es inferior a 0° se le sumara una vuelta es decir 360), consiguiendo que la toma de datos sea desde la primera revolución. (Arellano, 2015, p.65)

Posteriormente se calculan las variaciones de X, Y respectivamente asignándoles en delta X e Y los ángulos actuales, siendo incrementados con los diferenciales obtenidos.

La velocidad se puede definir mediante la división de las posiciones delta X y delta Y con respecto al tiempo para cada eje y después con estos valores se obtiene la velocidad instantánea por medio de la raíz cuadrada de la sumatoria estos valores al cuadrado. (Cruzado, 2017, p.25)

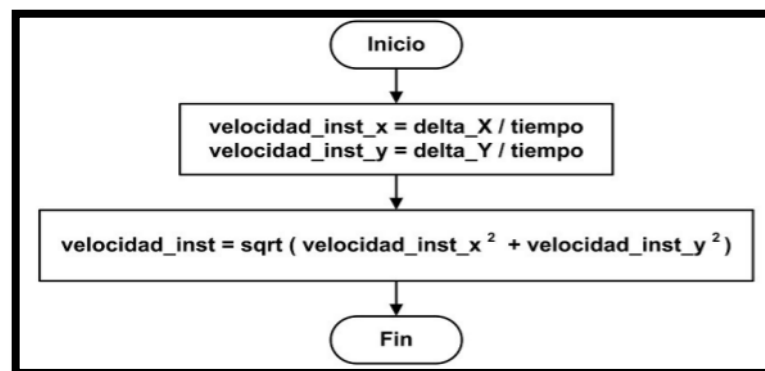


Figura 4-1. Diagrama del cálculo de velocidad.

Fuente: (Cruzado Muñoz, 2017, p.22)

1.1.2 Visión artificial

Como humanos, percibimos la estructura tridimensional del mundo que nos rodea con aparente facilidad, al mirar un retrato grupal enmarcado, puede contar fácilmente a todas las personas en la imagen e incluso adivinar sus emociones por su apariencia facial y aunque pueden idear ilusiones ópticas, para descifrar algunos de sus principios, una solución completa a este rompecabezas sigue siendo difícil de alcanzar. (Szeliski, 2015, p.25)

En la Figura 5-1 se muestra una aplicación de visión artificial en este caso detección de facial.

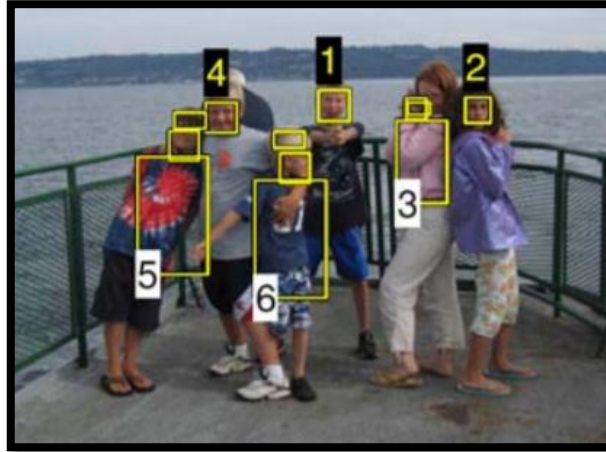


Figura 5-1. Algoritmos de detección de rostros

Fuente: (Szeliski, 2015)

Los investigadores en visión artificial han desarrollado, en paralelo, técnicas matemáticas para recuperar la forma tridimensional y la apariencia de los objetos en las imágenes. Sin embargo, a pesar de todos estos avances, el sueño de que una computadora interprete una imagen al mismo nivel que un niño de dos años (por ejemplo, contando a todos los animales en una imagen) sigue siendo difícil. (Szeliski, 2015, p.26)

1.1.2.1 *Preprocesamiento por segmentación*

Significa que al manipular la información que cada imagen posee, con la ayuda de un ordenador y con esta información adquirida de cada imagen se debe realizar un procedimiento específico, de acuerdo al objetivo que se quiera alcanzar, esto puede ser: segmentación, compresión, identificación y detección. (Procesamiento digital de imágenes, *Capítulo 1.pdf*, 2018, p.1)

➤ Segmentación de imágenes

La segmentación de imagen es la división de una imagen en regiones o categorías, que corresponden a diferentes objetos o partes de objetos. Cada píxel en una imagen se asigna a una de varias de estas categorías. (Chris, 2017, p.2). Una buena segmentación es típicamente una en la que:

- ✓ Los píxeles en la misma categoría tienen una escala de grises similar de valores multivariados y forman una región conectada,

- ✓ Los píxeles vecinos que se encuentran en diferentes categorías tienen valores diferentes.

Se muestra en la Figura 6-1, un ejemplo de segmentación de píxeles comunes.

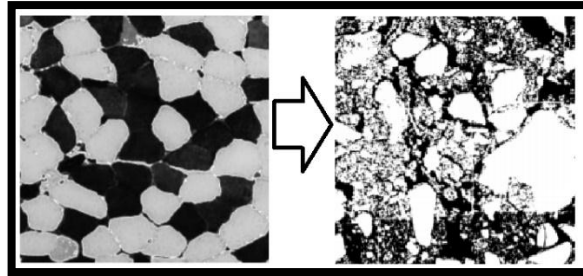


Figura 6-1. Segmentación de una imagen

Fuente: (Szeliski, 2015)

➤ Algoritmo de detección SIFT y SURF

ES la transformación de características invariables a escala, SIFT es un enfoque exitoso para la detección de características introducido por Lowe. El algoritmo SURF es basado en los mismos principios y pasos, pero utiliza un esquema diferente y debería proporcionar mejores resultados, al tener la ventaja de ser más rápido en costos computacionales. (Pedersen, 2016, p.1)

Con el fin de detectar puntos de características de manera invariable a escala, SIFT utiliza un enfoque de filtrado en cascada. Donde la diferencia de Gauss, DoG, se calcula en imágenes progresivamente reducidas, en general, la técnica para lograr la invariancia de escala es examinar la imagen a diferentes escalas, espacio de escala, utilizando núcleos gaussianos. Tanto SIFT como SURF divide el espacio de la escala en niveles y octavas. Una octava corresponde a una duplicación de σ , y la octava se divide en niveles uniformemente espaciados. (Pedersen, 2016, p.2)

En la Figura 7-1 se muestra las formas de escalamiento del algoritmo SIFT (reducción) y algoritmo surf (ampliación).

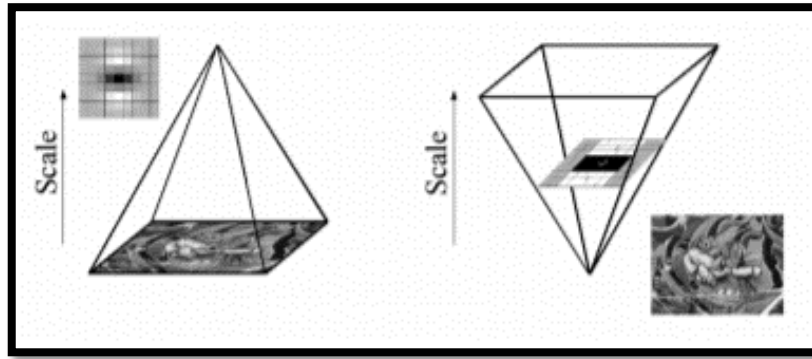


Figura 7-1. Escalamiento de algoritmo SIFT y SURF.

Fuente: (Pedersen, 2016)

➤ Identificación

El objetivo es identificar el conjunto de píxeles que son "significativamente diferentes" entre la última imagen de la secuencia y las imágenes anteriores; estos píxeles comprenden la máscara de cambio. La máscara de cambio puede ser el resultado de una combinación de factores subyacentes, incluida la aparición o desaparición de objetos, el movimiento de objetos en relación con el fondo o los cambios de forma de los objetos. (Radke et al, 2005, p.2)

Se puede diferenciar los cambios que hubo de imagen en distintos tiempos como da entender la Figura 8-1.



Figura 8-1. Detección de variaciones en una imagen tomada en tiempos distintos.

Fuente: (Radke, Andra, Al-Kofahi, & Roysam, 2005)

1.2 Análisis del Software principal implementado en el diseño.

Para realizar el proceso de visión artificial y control de posición del prototipo, son tomados en cuenta los programas de desarrollo dentro de los cuales Matlab, LabView y Python son los más indicados, debido a que se encuentran orientados en el manejo de objetos y procesamiento tanto de los datos suministrados por los actuadores y sensores como el procesamiento de imágenes y se posee un conocimiento previo sobre este tipo de programas.

1.2.1 MATLAB

Es una plataforma de programación diseñada específicamente para ingenieros y científicos. El corazón de MATLAB es el lenguaje MATLAB, un lenguaje basado en matrices que permite la expresión más natural de las matemáticas computacionales permitiendo con facilidad controlar los estados necesarios que necesitamos para poder realizar los cálculos en la odometría. (¿MathWorks, What is MATLAB?, 2018, p.1)

Con MATLAB, puedes: Analizar datos, Desarrollar algoritmos, Crear modelos y aplicaciones. El lenguaje, las aplicaciones y las funciones matemáticas integradas le permiten explorar rápidamente múltiples enfoques para llegar a una solución. MATLAB le permite llevar sus ideas de la investigación a la producción mediante la implementación en aplicaciones empresariales y dispositivos integrados, así como la integración con Simulink® y el diseño basado en modelos. El entorno se observa en la Figura 9-1 la cual posee diferentes ventanas donde se trabaja tanto en la programación en sí como en las variables de entorno. (¿What is MATLAB?, 2018, p.1)

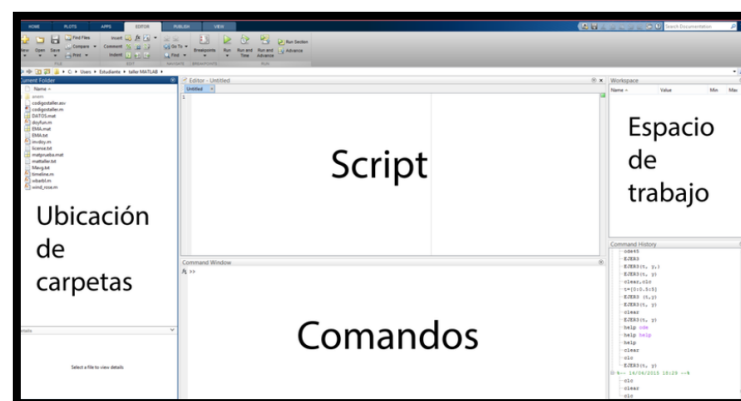


Figura 9-1. Ventana de inicio de Matlab.

Fuente: (MATLAB, 2018)

1.2.2 LabView

LabView es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con tipo dinámico y enlace dinámico, lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para usarlo como un lenguaje de scripting o pegamento para conectar componentes existentes. (Programmierung mit NI LabVIEW, 2015, p.1)

LabVIEW es esencialmente un entorno que permite la programación en G: este es un lenguaje de programación gráfico creado por National Instruments que se desarrolló inicialmente para comunicarse a través de GPIB, pero desde entonces se ha actualizado considerablemente. Hoy en día, G se puede utilizar para aplicaciones de prueba automatizadas, adquisición de datos generales, programación de FPGA, etc. (What is LabVIEW, 2019, p.2)

En la Figura 10-1 se muestra la pantalla principal de la plataforma LabView 2018.

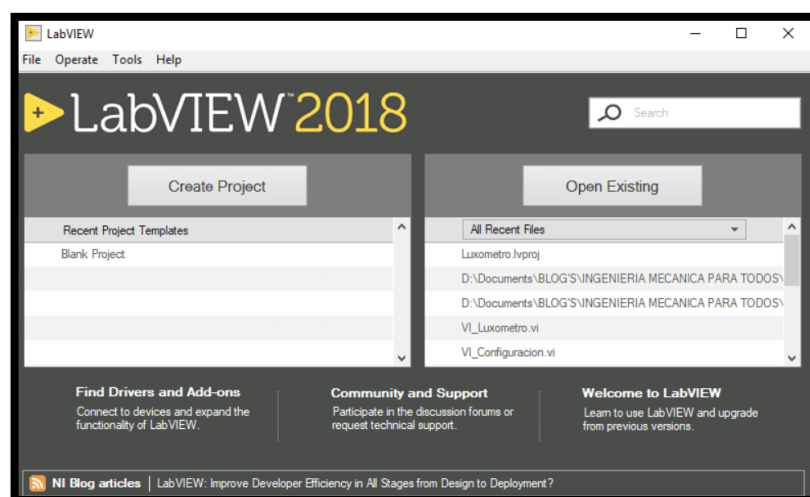


Figura 10-1. Ventana de inicio de LabView.

Fuente: (LABVIEW, 2019)

1.2.3 OPEN CV

Es una biblioteca de código abierto con licencia BSD que incluye varios cientos de algoritmos de visión por ordenador, el documento de visión describe la API de OpenCV 2.x que es esencialmente una API de C++, en oposición de a la API de OpenCV 1.x basada en C. tiene una estructura modular, lo que significa que el paquete incluye varias bibliotecas compartidas o estáticas. (The OpenCV Manual, 2019, p1)

En la Figura 11-1 se muestra el entorno gráfico de la plataforma OpenCV.

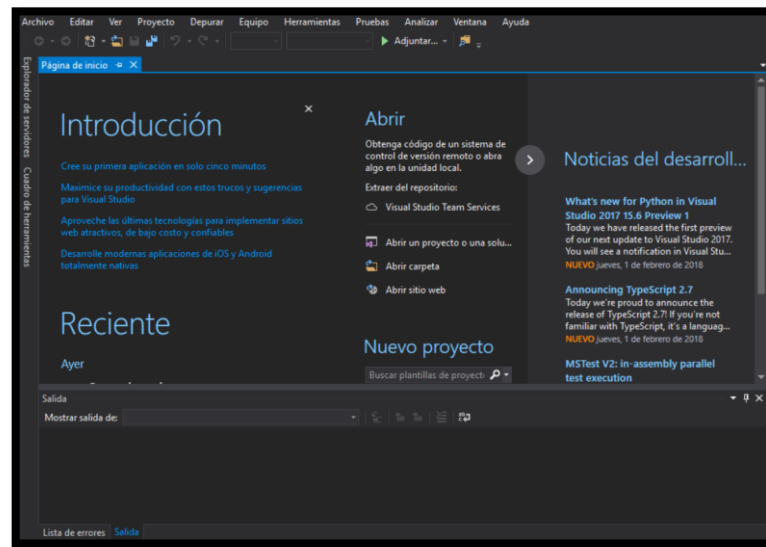


Figura 11-1. Ventana de OpenCV.

Fuente: (OpenCV, 2018)

En la tabla 2-1 se muestra una breve comparación entre ventajas y desventajas de los softwares clásicos utilizados para el desarrollo de prototipos.

Tabla 2-1. Ventajas y Desventajas en los softwares de desarrollo.

	LabView	MatLab	OpenCV
Ventajas	<p>Compatibilidad con SO.</p> <p>Programación gráfica, lenguaje G.</p> <p>Compatible con tarjetas controladoras.</p> <p>Flexibilidad al sistema para realizar cambios.</p>	<p>Compatibilidad con SO.</p> <p>Cuenta con herramientas o toolboxes.</p> <p>Interfaz amigable.</p> <p>Compatible con tarjetas controladoras.</p> <p>Tiempo bajo de procesamiento.</p>	<p>Compatibilidad con SO</p> <p>Programación lenguaje C o C++.</p> <p>Específicamente para proyectos de visión artificial.</p> <p>Programación en lenguaje de alto y bajo nivel.</p>
Desventajas	<p>Menor capacidad de procesamiento.</p> <p>Alto costo.</p> <p>Interfaz algo complicado.</p>	<p>En ocasiones errores de ejecución de software.</p> <p>Compatibilidad de ejecutables en otros ordenadores.</p>	<p>Interfaz algo complicada</p> <p>Tiempo elevado de procesamiento.</p>

Fuente: (Tapia & Manzano, 2013); (Veintimilla & Siguencia, 2014); (Muñoz Manso, 2014)

Realizado por: Guerrero, Caren, 2020

1.3 Análisis del Hardware principal del sistema de recolección.

1.3.1 Configuración de los Robots Móviles

De acuerdo a los avances de la robótica este tipo de robot es capaz de moverse de manera autónoma dentro de los rangos establecidos por el usuario y el su uso ha abierto campos para diversos aspectos en la comunidad, no obstante, mientras más sofisticado sea el dispositivo, más costoso llegará a ser. (Vaca, 2017, p.23)

Las configuraciones de los robots móviles son las siguientes:

1.3.1.1 Configuración de Rueda convencional y giratoria

En la Figura 12-1 se observa ruedas giratorias que se usan como ruedas pasivas y se pueden aplicar junto con otras ruedas para alcanzar la movilidad omnidireccional. Una de las configuraciones populares para usar tales ruedas es un robot de accionamiento diferencial. Las ruedas convencionales se usan como ruedas activas y una rueda giratoria agrega estabilidad al robot, proporcionando un polígono de soporte triangular, y se usa para permitir una rotación libre. (Shabalina, Sagitov & Magid, 2018, p.2)



Figura 12-1. Tipos de ruedas: (a) un volante convencional /, (b1) una rueda giratoria, (b2) una rueda giratoria, (c) una rueda universal, (d) una rueda Mecanum.

Fuente: (Shabalina, 2018)

1.3.1.2 Configuración Direccional y Ackerman

En la Figura 13-1 muestras las distintas configuraciones de volantes, las cuales proviene de su mecanismo de dirección, que actúa a través del mecanismo de transmisión de un robot. Así, un motor de dirección controla una dirección de dirección (a la izquierda o la derecha) de las ruedas y de esta manera una dirección de movimiento, mientras que un motor de transmisión transfiere un par a las ruedas para proporcionar un movimiento hacia adelante y hacia atrás. (Shabalina et al., 2018, p.3)

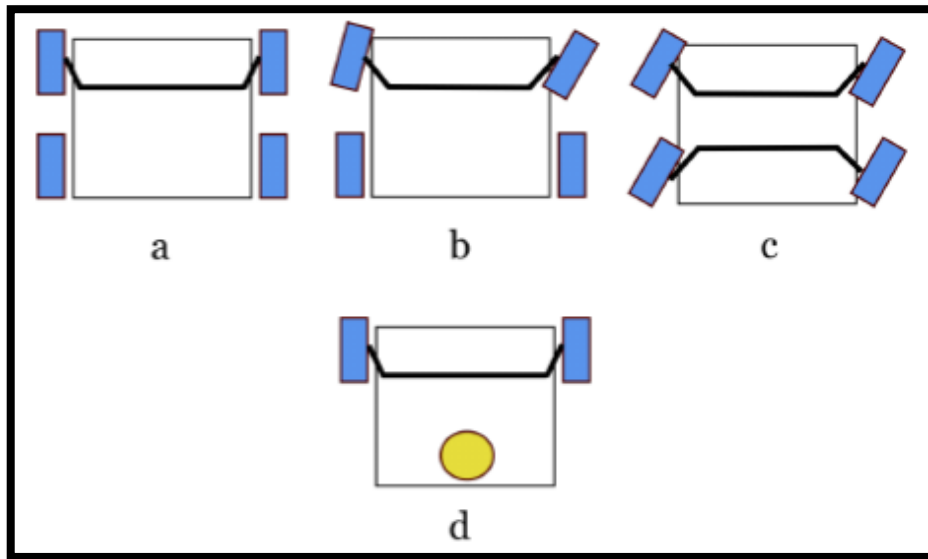


Figura 13-1. (a) una vista genérica de la configuración, (b) una configuración de manejo de Ackerman, (c) una configuración casi omni drive, (d) Configuración direccional.

Fuente: (Shabalina, 2018)

1.3.1.3 Configuración Omni Wheel universal

Una configuración se puede componer de manera óptima utilizando tres ruedas externas. Se pueden montar tres ruedas universales en una plataforma triangular con sus ejes inclinados 120 grados entre sí como se observa en la Figura 14-1. (Shabalina et al., 2018, p.3)

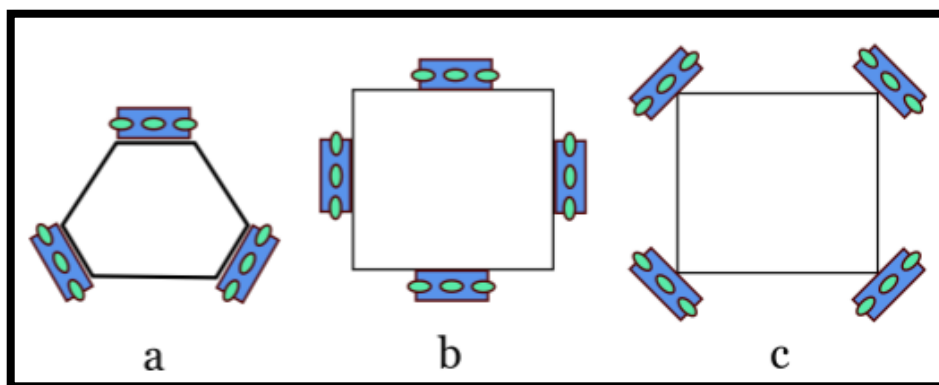


Figura 14-1. Configuraciones de rueda universal: (a) base triangular; (b) base cuadrada, las ruedas están a los lados; (c) base cuadrada, las ruedas están en las esquinas.

Fuente: (Shabalina, 2018)

1.3.1.4 Configuración Rueda Mecanum

Con las ruedas omnidireccionales Mecanum, son una plataforma típica que tiene una forma rectangular o cuadrada y las ruedas están ubicadas en un estilo de automóvil, están montadas en línea entre sí. Generalmente los robots usan cuatro Mecanumwheels de acuerdo a la Figura 15-1. (Shabalina et al., 2018, p.4)

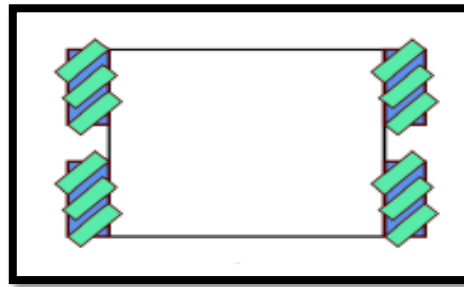


Figura 15-1. Configuración de ruedas Mecanum

Fuente: (Shabalina, 2018)

Se compara los parámetros mecánicos de las ruedas y sus configuraciones. En la segunda y tercera columnas se indican otras ruedas DW (diferentes), mientras que PC significa que una rueda se considera a través de un prisma de ser parte de una configuración complicada con otras ruedas (diferentes).

La Tabla 3-1 presenta los resultados de la comparación de las diferentes configuraciones para los parámetros, Posible configuración de ruedas, número mínimo de ruedas y los grados de libertad.

Tabla 3-1. Tabla comparativa de las diferentes configuraciones.

Tipo de rueda	Posible configuración de ruedas	Número mínimo de ruedas	DoF
Convencional	2 o más ruedas	2	1
Direccional	2 (+1 DW) 3, 4 ruedas	2 (+1 DW)	2
Giratoria	1 (es PC)	1 (es PC)	1, 2, 3
Universal	3 o 4 ruedas	3	3
Mecanum	4 o más ruedas	4	3

Fuente: (Shabalina, 2018)

Realizado por: Guerrero, Caren, 2020

La Tabla 4-1 presenta los resultados de la comparación de parámetros generales de diferentes tipos de ruedas. Para tales parámetros, la sensibilidad a las condiciones del piso, la complejidad de fabricación y los tipos de ruedas simples (convencional, direccional y rueda giratoria) proporcionan la mejor solución según las necesidades del operador.

Tabla 4-1. Parámetros generales de las ruedas.

Tipo de Rueda.	Complejidad de armado.	Sensibilidad a la superficie.	Sensibilidad a objetos en la superficie.
Convencional	Baja	Baja	Baja
Direccional	Baja	Baja	Baja
Giratoria	Baja	Baja	Baja
Universal	Media	Alta	Alta
Mecanum	Alta	Alta	Alta

Fuente: (Shabalina et al., 2018)

Realizado por: Guerrero, Caren, 2020

1.3.2 Aplicaciones de robots móviles

1.3.2.1 SR1 Robot Móvil Multifuncional

El prototipo posee todos los componentes necesarios para cumplir las más complejas exigencias que requiera el usuario, posee elementos de posicionamiento, sensores, actuadores e incluso puede transmitir audio y video para un entorno de desarrollo en tiempo real como se observa en la Figura 16-1, todo ello reunido en una tarjeta controladora programable desde cualquier ordenador con el software distribuido por la misma compañía. (SuperRobotica, “Robot SR1”, 2017, p.1)



Figura 16-1. Robot SR1 móvil multifuncional.

Fuente: http://www.superrobotica.com/sr1_Robot.htm

1.3.2.2 Robot móvil basado en Arduino

Arduino posee librerías de ejemplo para casos de robots móviles ya sea en configuración diferencial o Ackermann las instrucciones que debe poseer cada motor es intuitiva por lo tanto elaborar un robot móvil en conjunto con módulos creados por Arduino para su comunicación en tiempo real ya sea con bluetooth o antenas de RF como se aprecia en la Figura 17-1, son opciones que pueden dar una solución al problema que ha sido planteado el que consiste en enviar información sobre la posición del robot en el campo de juego mientras realiza la mejor trayectoria para recoger las pelotas.

Arduino nano con sus entradas y salidas compensa los requerimientos en manejo de sensores y actuadores tanto como en espacio para satisfacer las ambiciones del robot.

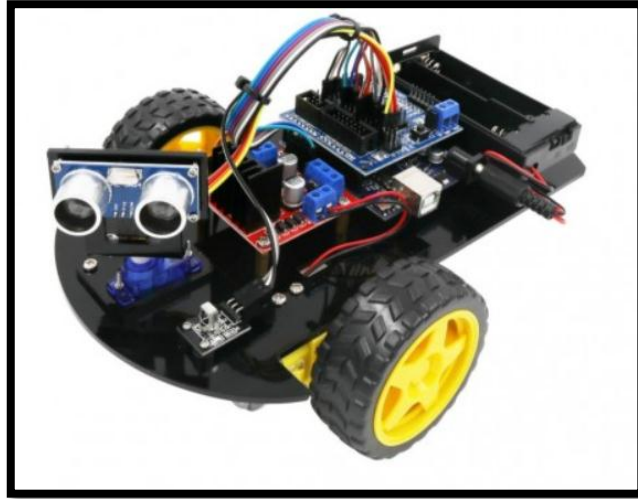


Figura 17-1. Robot móvil de la plataforma ARDUINO.

Fuente: (Kit Arduino Rw, 2018)

1.3.2.3 TI robotics systems

TI-RSLK Maze es un robot de bajo costo que proporciona al usuario la información necesaria sobre el funcionamiento de los diseños electrónicos. TI-RSLK Maze Edition viene con 20 módulos de aprendizaje y está diseñado para enseñar sistemas y aplicaciones integrados pudiéndose ampliar y utilizar en una variedad de clases de ingeniería. (TI-Robotics-System-Learning-Kit, TI-RSLK kits, 2017, p.1)

En la Figura 11-1 se muestra el robot diseñado por TI robotics conjuntamente el kit de sensores propuestos por Texas Instruments.

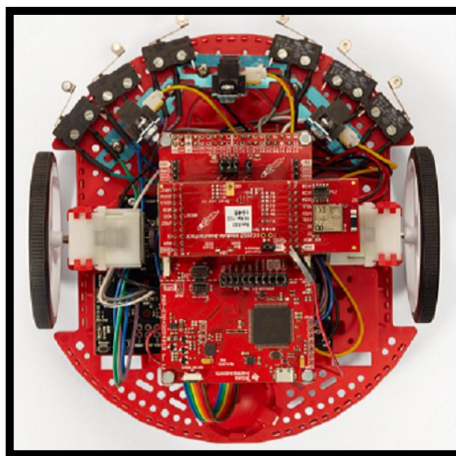


Figura 18-1. Robot Stellaris from Texas Instruments.

Fuente: (TI-RSLK Advanced Kit, 2018)

1.3.3 Codificadores (Encoders)

Los codificadores de posición son uno de los componentes clave para la posición, orientación y control de movimiento del robot. (RLS, Magnetic encoders for industrial robotics, 2019, p.1)

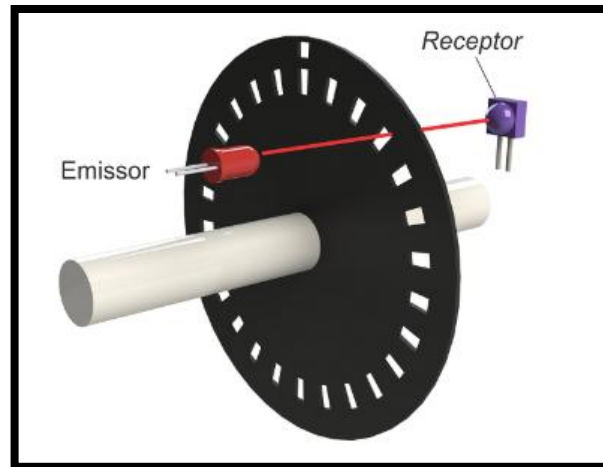


Figura 19-1. Partes principales de un encoder.

Fuente: (Post_encoder, 2019)

Las marcas de disco tienen la funcionalidad de bloqueo y desbloqueo del haz de luz led al fotodetector como se observa en la Figura 19-1, por lo que a medida que el disco gira el fotodetector junto con un circuito electrónico, envía una señal de forma de onda a las salidas del codificador cuadrado, proporcional al número de marcas del codificador de acuerdo con la resolución del codificador. (HI Tecnologia, O que é Encoder?, 2018, p.1)

En la Figura 20-1 se detallan los diferentes tipos de disco para encoders infrarrojos por el número de pulsos por vuelta.

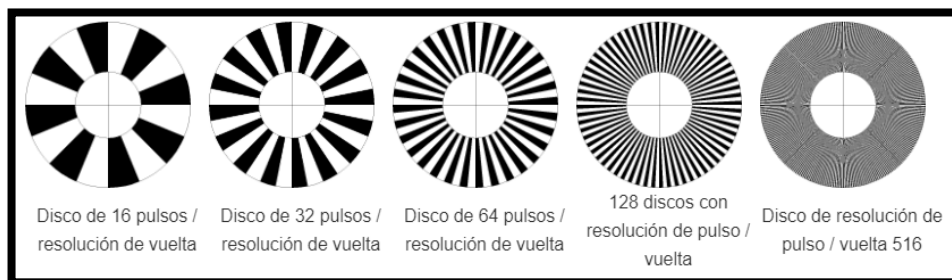


Figura 20-1. Tipos de discos de los codificadores.

Fuente: (O que é Encoder, 2019)

1.3.3.1 Lectura de encoders

Son fundamentales en el robot debido a que de estos encoders ubicados en las partes laterales posteriores del robot tendrán la tarea de registrar el movimiento real generado por el movimiento de los motores

La secuencia de bits generada por el encoder se muestra en la Tabla 5-1 obteniendo un sentido horario y uno antihorario para poder tener una idea del movimiento y acumulación de ticks en los encoders.

Tabla 5-1. Secuencia de bits generado por los encoders

Sentido Anti horario	Decimal	Intercambio de 2 por 3	Sentido Horario	Decimal	Intercambio de 2 por3
01	1	1	11	3	2
00	0	0	10	2	3
10	2	3	00	0	0
11	3	2	01	1	1
01	1	1	11	3	2
00	0	0	10	2	3
10	2	3	00	0	0
11	3	2	01	1	1

Fuente: (Arellano, 2015)

Realizado por: Guerrero, Caren, 2020

En resumen, la Tabla 5-1 indica que cuando exista un giro en sentido horario se generará un incremento y viceversa en sentido antihorario por lo que podrá realizar un conteo de los estados de manera secuencial. Mientras que el Tabla 6-1 se muestra la disposición de conexión de los encoders.

Tabla 6-1. Disposición de conexión del encoder.

Entrada	Encoder	Contador Rápido	Entradas Simples
I0	Señal (A)	No	Si
I1	Señal (B)	No	Si
I2	Señal (O)	Si	Si
I3	-----	Si	Si

Fuente: (Conexión encoder, 2018)

Realizado por: Guerrero, Caren, 2020

El diagrama de conexión del encoder se aprecia en la Figura 21-1.

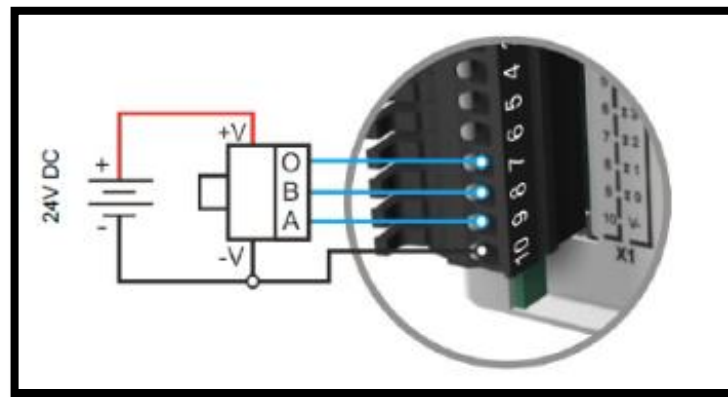


Figura 21-1. Esquema de conexión de un encoder.

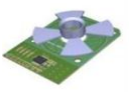

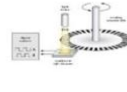
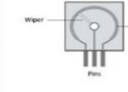
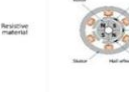

Fuente: (O que é Encoder, 2019)

1.3.3.2 Tipos de codificadores.

El tipo más básico de sensor de posición magnético utilizado para la retroalimentación de posición en motores es el de efecto Hall. Es un dispositivo de circuito integrado (IC) con sensor magnético muy simple que cambia su estado de salida cuando un imán pasa por él de alguna polaridad magnética particular y fuerza de campo. El dispositivo permanece en ese estado hasta que pasa un campo magnético de igual intensidad, pero de polaridad opuesta. (O que é Encoder, 2019, p.1)

La Tabla 7-1 presenta los resultados de la comparación de parámetros generales de diferentes tipos encoder. Para tales tipos, Inductivo, Resolver, Óptico, Resistivo, Efecto Hall, Magnético.

Tabla 7-1. Comparación de los sensores.

						
Tipo de Sensor	Inductivo	Resolver	Óptico	Resistivo	Efecto Hall	Magnético
Tamaño y peso	Bajo	Alto	Medio	Muy bajo	Muy Bajo	Bajo
Costos	Medio	Alto	Alto	Bajo	Bajo	Medio
Potencia	Bajo	Alto	Medio	Bajo	Bajo	Bajo
Precisión	Muy alto	Muy alto	Muy alto	Bajo	Medio	Alto
Redundancia	Posible	No	No	Posible	No	Posible

Fuente: (RB, How Inductive Position Sensors, 2018)

Realizado por: Guerrero, Caren, 2020

1.3.4 Cámaras

Una cámara es una caja que es negra por dentro y permite que caiga la cantidad correcta de luz en la superficie que grabará la fotografía. Esta superficie varía en los dos tipos de cámaras. Si se trata de una cámara de película, la luz cae sobre la superficie de una película y, si se trata de una cámara digital, es un sensor de imagen que registra la luz. Si la cantidad de luz es mayor que la requerida, la imagen aparecerá en blanco y faltarán detalles debido a la sobreexposición, y si la luz es inferior a la requerida, la foto aparecerá oscura, quitando nuevamente los detalles de la imagen. (Photojournalism, 2016, p.1)

1.3.4.1 Cámaras Monocromáticas

Contiene una cantidad significativa de información sobre los objetos. Con el desarrollo de cámaras de alta resolución, es práctico cambiar la resolución espacial para obtener información espectral en una técnica de imagen de disparo único (multiplexación espacial). (Sahoo, Tang, & Dang, 2017, p.1)

Una versión de los dispositivos de imágenes multiespectrales es la cámara a color donde los múltiples filtros espectrales se distribuyen espacialmente en una matriz de detectores 2D. Uno puede incluso deshacerse de la resolución espacial para lograr una mayor resolución espectral con una gran cantidad de filtros espectrales para usar como un espectrómetro compacto como se puede apreciar en la Figura 22-1. (Sahoo Tang & Dang, 2017, p.2).



Figura 22-1. Cámara monocromática.

Fuente: (macchina fotografica, 2017)

1.3.4.2 Cámara estereoscópica.

Una cámara estereoscópica permite a los usuarios capturar imágenes que se renderizan instantáneamente en 3D. Ya sea una cámara fotográfica o de video, las cámaras estereoscópicas contienen más de una lente como se puede ver en la Figura 23-1, cada una con su propio sensor de imagen o marco. Para crear la profundidad adecuada en las imágenes estereoscópicas en 3D, hay tres elementos básicos que debe determinar: el primer plano, el fondo y la paralaje cero. El primer plano es el elemento en su imagen que aparecerá más cerca del espectador, mientras que el fondo es la imagen que aparecerá más abajo. (Knowing How a Stereoscopic Camera Works, Steve's Digicams, 2018, p.1)



Figura 23-1. Cámara estereoscópica.

Fuente: (Merino, 2017)

1.3.5 Tarjetas de desarrollo

Es un componente "pequeño" y con "inteligencia" programable, utilizada para controlar procesos lógicos. El control del proceso debe entenderse como el control periférico como LED, botones, pantallas LCD. (Microcontrolador PIC, 2018, p.1)

Un microcontrolador difiere de un microprocesador de varias maneras. Lo primero y más importante es su funcionalidad. Entonces se puede usar microprocesador, deben usar otros

componentes tales como memoria y componentes para recibir y enviar datos. (*Microcontrolador PIC*, 2018, p.2)

1.3.5.1 Tarjeta microcontroladora Arduino

Es una tarjeta de código abierto programable, al ser una tarjeta de bajo costo es ideal para realizar procesos de automatización y proyectos de electrónica y robótica. Puede recibir información de sensores y manejar accionadores, motores, etc., Además que existe una gran gama de ellos como se observa en la Figura 24-1. (¿MECATRÓNICA LATAM, What is Arduino?, 2018, p.1)

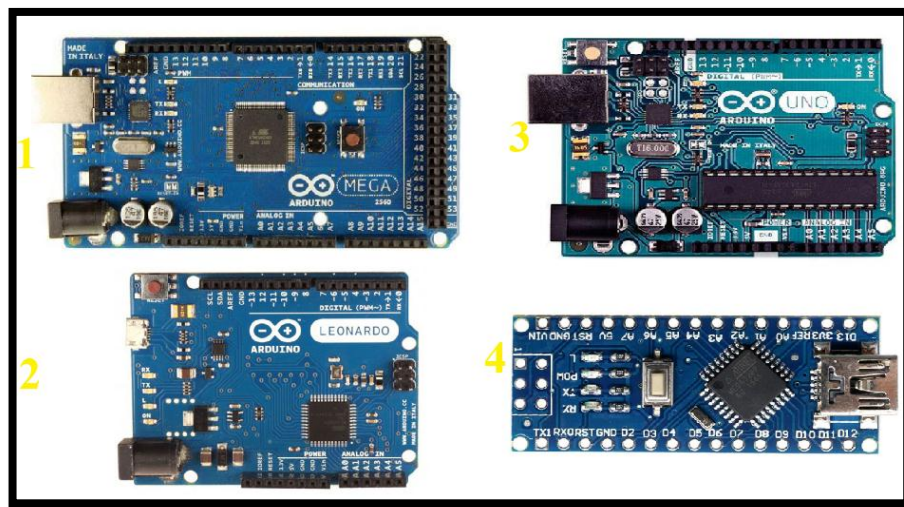


Figura 24-1. Arduino Mega, Leonardo, Uno y Nano respectivamente

Fuente: (Arduino Official Store, 2019)

Todas las tarjetas de estos procesadores son similares sus pines permiten entrada y salida de información además de poder enviar señales de control para activar o desconectar los actuadores a los que se encuentren conectados.

1.3.5.2 Raspberry Pi

Al poseer una gran capacidad de procesamiento de información, esta una tarjeta de programación brinda una alta gama de opciones en la elaboración de proyectos de electrónica, teniendo dos modelos para su elección

- El modelo A debido a que posee un bajo consumo de energía es una buena elección en proyectos de energía solar, eólica o batería. Además de poseer una memoria interna de

256MB haciendo un poco limitada la ejecución de algoritmos y aplicaciones robustas. (Halfacree, 2018, p.12)

- El modelo B tiene un precio algo superior al modelo A, debido a que viene con más características, comenzando con una memoria de 512MB, además de tener puertos USB adicionales permitiéndonos conectar un teclado y un ratón como se puede ver en la Figura 25-1. (Halfacree, 2018, p.14)



Figura 25-1. Microcontrolador Raspberry Pi B

Fuente: (Raspberry Pi, 2019)

1.3.5.3 Intel Galileo

Galileo es una placa de microcontrolador basada en el procesador de aplicaciones Intel® Quark SoC X1000, un sistema Intel Pentium de 32 bits en un chip. Los pines digitales 0 a 13 (y los pines adyacentes AREF y GND), las entradas analógicas 0 a 5, el encabezado de alimentación, el encabezado ICSP y los pines del puerto UART (0 y 1) como se observa en la Figura 26-1. (Arduino, IntelGalileo, 2018, p.1)

Galileo está diseñado para soportar escudos que funcionan a 3.3V o 5V. El voltaje de funcionamiento central de Galileo es 3.3V. Sin embargo, un puente en el tablero permite la traducción de voltaje a 5V en los pines de E / S. Esto proporciona soporte para escudos Uno de 5V y es el comportamiento predeterminado. Al cambiar la posición del puente, la traducción de voltaje se puede desactivar para proporcionar una operación de 3.3V en los pines de E / S. (Arduino, IntelGalileo, 2018, p.1)

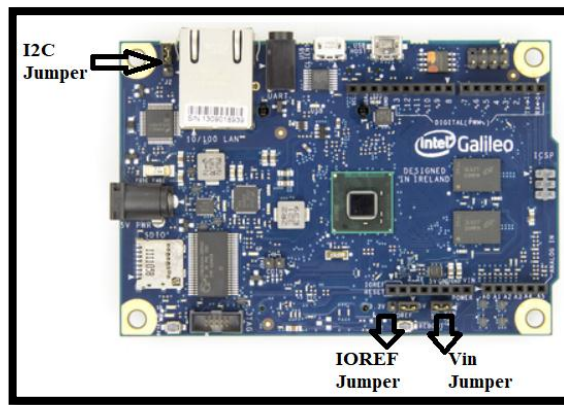


Figura 26-1. Tarjeta Intel Galileo

Fuente: (Arduino Official, 2019)

En la Tabla 8-1 se analiza los distintos tipos de tarjeta de desarrollo dependiendo de sus características en Hardware.

Tabla 8-1. Principales características de las tarjetas de desarrollo.

Características	Raspberry pi	Intel Galileo (Gen 1)	Arduino
Procesador	ARM11	SoC Quark X100	ATMega2560
Voltaje de operación	3.3V/5V	3.3V/5V	5V
Voltaje de entrada	5V	5V	7-12V
RAM	512MB	512KB	256KB
Velocidad de reloj	700Mhz	400MHz	16MHz
E/s Analógicas	-	6	16
E/S digitales	8	14	54

Fuente: (Arduino, 2018)

Realizado por: Guerrero, Caren, 2020

CAPÍTULO II

2 MARCO METODOLÓGICO

El presente capítulo se enfocará en los requerimientos del sistema de visión artificial a implementar el dispositivo móvil. Se presentará el diseño y armado de la estructura mecánica necesaria para el robot y la cámara para la detección de imágenes. Posteriormente, se revisará el diseño y montaje electrónico para el control mencionado y la comunicación del sistema de visión artificial. Además, se presentará en general características de los elementos usados para la implementación de este proyecto.

2.1 Concepción del prototipo del sistema del robot recogedor

- Implementar un sistema móvil seguro y con la capacidad de poder levantar una pelota de tenis con la ayuda de una herramienta incorporada en el mismo.
- Configurar los parámetros necesarios en la kinetic y en el receptor móvil para tomar la mejor decisión en el movimiento odométrico
- Calibrar el sistema para poder disminuir el error que presentan los sistemas odométrico cuando la ruta de desplazamiento es considerablemente grande.
- La etapa de adquisición de imagen, así como la de envío y recepción de información tendrá la facilidad de informar continuamente al usuario.

2.1.1 Selección del hardware que conforma el prototipo

Para el correcto funcionamiento del sistema de recolección de pelotas el uso de microcontroladores es necesario para poder tener un control sobre los sensores y actuadores implementados en el dispositivo.

Los elementos utilizados en el dispositivo son:

- 1 Arduino Nano
- 1 Arduino Uno
- 1 Servomoto 180°
- 4 MicroServo 180°

- 1 Lm2596
- 2 nrf24l01
- 2 encoders comparador LM393
- 1 mpu6050 giroscopio acelerómetro
- Llantas goma
- 1 batería lipo de 1ª 7.4v marca tuning
- 1 boton
- 1switch
- Cortes laser

2.1.1.1 Arduino Nano

El Arduino Nano como se puede observar en la Figura 1-2, es un tablero pequeño, completo y amigable basado en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x).

Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero en un paquete diferente. Carece solo de un conector de alimentación de CC y funciona con un cable USB Mini-B en lugar de uno estándar. El Nano fue diseñado y producido por Gravitech. (Arduino Nano, 2016, p.1)

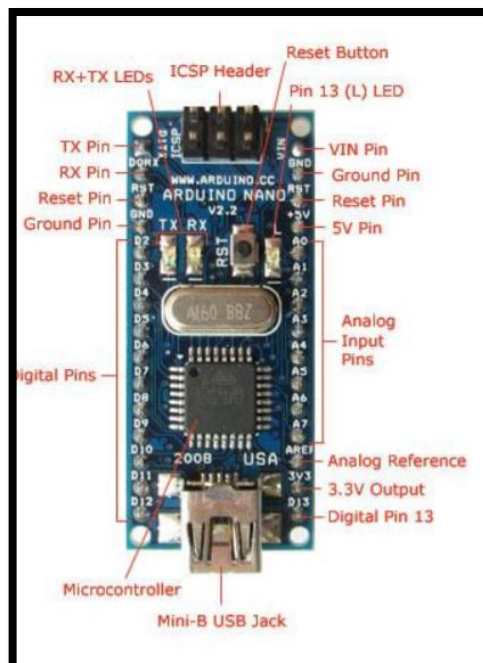


Figura 1-2. Arduino Nano.

Fuente: (Arduino Nano, 2016)

Las características principales se observan en el anexo A

- Voltaje de operación: 5V.
- Digital E/S: 14.
- Analógica E/S: 8.
- Velocidad de reloj: 16 Mhz.

A continuación, en la Tabla 1-2 se describe brevemente las ventajas y desventajas del uso del Arduino Nano

Tabla 1-2. Ventajas y desventajas de Arduino nano.

Ventajas	<ul style="list-style-type: none">• Es mucho más pequeño que el Arduino Mega.• Está basado en el microcontrolador ATmega328.• Tiene una entrada mini-usb a través de la cual se puede subir el código fuente para la ejecución de los comandos.
Desventajas	<ul style="list-style-type: none">• Limitado número de pines de entrada y salida, en comparación con otras versiones.

Fuente: (Arduino Nano, 2016)

Realizado por: Guerrero, Caren, 2020

2.1.1.2 Módulo L298N puente H L298

Como se muestra en la Figura 2-2, es una tarjeta que sirve para el control de motores de corriente directa y todo dispositivo que necesita una carga inductiva, la tarjeta es ideal para controlar los motorreductores que son los encargados del descenso de cada uno de los huevos al finalizar el periodo de incubación. (*L298N Motor Driver.pdf*, 2017, p.1)

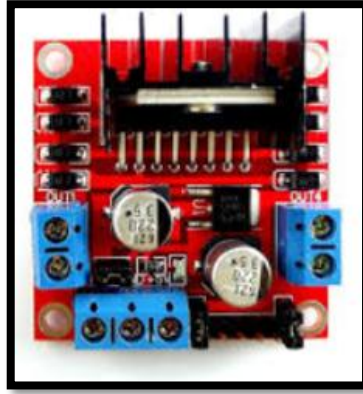


Figura 2-2. Puente H L298N.

Fuente: (Driver L298, 2016, p.1)

Las características de este dispositivo se pueden revisar en el anexo B

- Integrado L298N.
- Corriente de funcionamiento de 2A.

A continuación, en la Tabla 2-2 se describe brevemente las ventajas y desventajas del uso del módulo L298N.

Tabla 2-2. Ventajas y Desventajas del módulo L298N.

Ventajas	<ul style="list-style-type: none"> • Tiene gran capacidad de filtrado de ruido. • Puede llegar a trabajar hasta con un nivel de tensión de entrada de 46V. • Posee un rendimiento anti-interferencia excepcional.
Desventajas	<ul style="list-style-type: none"> • En el caso de tensiones mayores a 12v se debe conectar el jumper, caso contrario se puede llegar a quemar el dispositivo.

Fuente: (Driver L298 , 2016)

Realizado por: Guerrero, Caren, 2020

2.1.1.3 Motor reductor

Como se muestra en la Figura 3-2, es el encargado del movimiento del robot móvil, es ideal para el desplazamiento controlado por un encoder ubicado en cada uno de las ruedas posteriores del robot. (RB04M022.pdf, 2016, p.1)

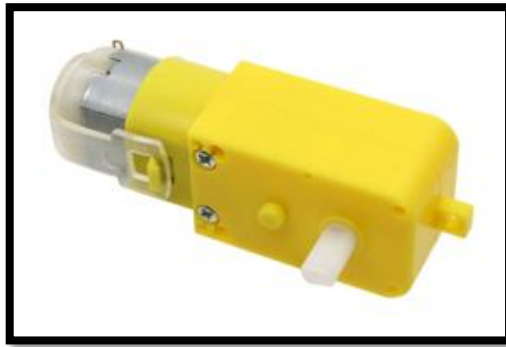


Figura 3-2. Motorreductor

Fuente: (MOTO REDUCTOR, 2017)

Las características de este motor que se deben considerar están en el anexo C

- Voltaje de operación: 3-6V.
- Velocidad máx.: 220 RPM.

A continuación, en la Tabla 3-2 se describe brevemente las ventajas y desventajas del motor-reductor.

Tabla 3-2. Ventajas y desventajas del motor-reductor.

Ventajas	<ul style="list-style-type: none"> • Consigue una adaptación perfecta tanto en la velocidad como en la potencia. • Adquiere mayor eficacia en la transmisión de la potencia que suministra al motor.
Desventajas	<ul style="list-style-type: none"> • Al ser un diseño mecánico, todo tren de engranajes, agrega pérdidas importantes al robot.

Fuente: (MOTO REDUCTOR, 2017)

Realizado por: Guerrero, Caren, 2020

2.1.1.4 Módulo NRF24l01

nRF24L01 es un transceptor de radio de un solo chip como se muestra en la Figura 4-2, sirve para una comunicación de 2.4 - 2.5 GHz de banda. El transceptor consta de un sintetizador de frecuencia totalmente integrado, una potencia amplificadora, un oscilador de cristal, un demodulador, modulador y ShockBurst. (NRF24L01 2.4 GHz, 2017, p.1)

La potencia de salida, los canales de frecuencia y la configuración del protocolo son fácilmente programable a través de una interfaz SPI. El consumo de corriente es muy bajo, solo 9.0mA a una

potencia de salida de -6dBm y 12.3mA en modo RX. Apagado incorporado y los modos de espera hacen que el ahorro de energía sea fácilmente realizable. (Módulo NRF24L01 2.4 GHz, 2017)

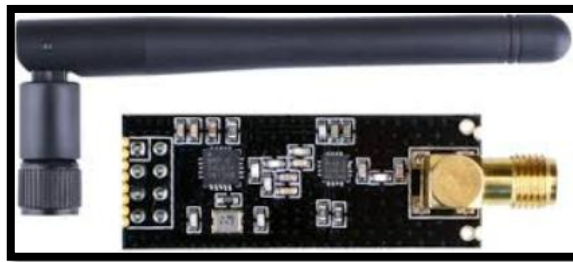


Figura 4-2. Módulo -nRF14l01.

Fuente: (NRF24L01 2.4 GHz, 2017)

Las características de estas antenas se pueden observar en el anexo D

- Voltaje de operación: 1.9V.
- Distancia máxima de detección: 100m.
- Corriente máxima de funcionamiento 12mA

En la tabla 4-2 se detallan las ventajas y desventajas de usar el módulo inalámbrico NRF24L01.

Tabla 4-2. Ventajas y desventajas del Módulo NRF24l01.

Ventajas	<ul style="list-style-type: none"> • Posee una distancia máxima de conexión de hasta 1km. • Mantiene una comunicación constante con el receptor
Desventajas	<ul style="list-style-type: none"> • La distancia máxima puede variar dependiendo de la presencia de obstáculos. • Al perder la comunicación es necesario reiniciar la comunicación.

Fuente: (NRF24L01 2.4 GHz, 2017)

Realizado por: Guerrero, Caren, 2020

2.1.1.5 *Kinetic X-Box*

Kinect es más inteligente que una cámara web promedio. Como se observa en la Figura 5-2, primero, tiene una vista excelente gracias a su truco secreto: destacar la habitación con luz infrarroja invisible. La cámara te ve maravillosamente gracias a este infrarrojo. Y, junto con un software avanzado adecuado puede rastrear 48 puntos de un cuerpo en tiempo real.

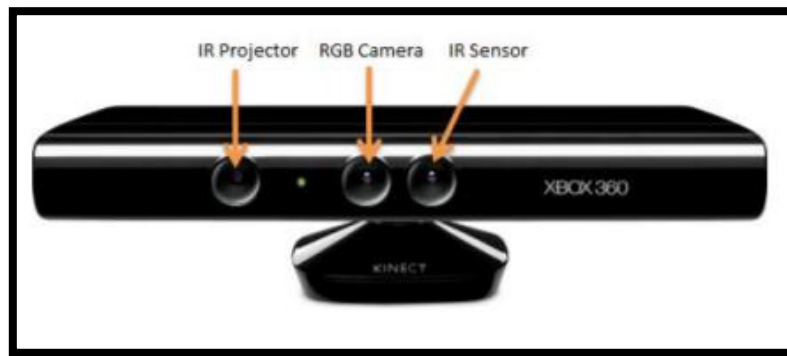


Figura 5-2. Estructura exterior de un Kinetic.

Fuente: (Top5 Kinetic, 2016)

Los detalles de especificaciones de la kinetic están en el anexo E

- Sensor de cámara común para adquirir imágenes RGB.
- El emisor IR + sensor IR utiliza luz infrarroja para adquirir imágenes de profundidad
- V1: luz estructurada / luz codificada
- V2: tiempo de vuelo
- Conjunto de micrófonos para capturar la señal de sonido y posición de sonido

En la Tabla 5-2 se describe brevemente las ventajas y desventajas del uso de la cámara Kinect Xbox 360.

Tabla 5-2. Ventajas y desventajas del Kinetic Xbox 360.

Ventajas	<ul style="list-style-type: none"> • Adquisición de imágenes con descripción de profundidad. • Facilidad de instalación y compatibilidad con un ordenador.
Desventajas	<ul style="list-style-type: none"> • Distancia de detección limitada hasta 3m • Depende de condiciones ideales para realizar su mejor desempeño.

Fuente: (Top5 Kinetic, 2016)

Realizado por: Guerrero, Caren, 2020

2.1.2 Programación del sistema.

Cabe recalcar que el software usado para el reconocimiento de pelotas y a su vez una parte del sistema de localización u odometria es Matlab. Para ello se aplicaron varios algoritmos tanto de reconocimiento de objetos basados en su color y su forma además usando un microntrolador para controlar en todo momento el prototipo. El sistema tiene un principio de funcionamiento basado

en 4 etapas en las cuales cada etapa tiene una programación propia como se mencionó anteriormente haciendo que el sistema tenga algoritmos de forma estructurada haciendo más visible el sistema de procesamiento y funcionalidad del sistema como se observa en el ciclo de función generalizado en la Figura 6-2.

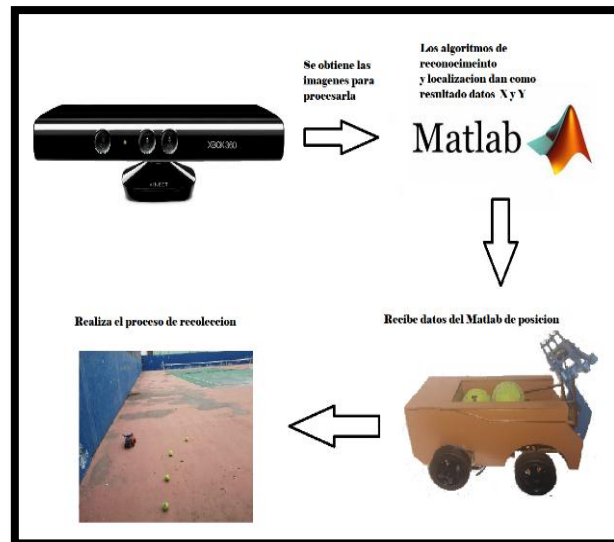


Figura 6-2. Esquema final del funcionamiento.

Realizado por: Guerrero, Caren, 2020

2.1.2.1 Etapas de Programación en Matlab

Dentro del lenguaje de programación Matlab se realizó varias etapas que consta de:

- Obtención de imágenes.
- Tratamiento de Imágenes.
- Algoritmo de Detección de Pelotas y localización.
- Algoritmo de Navegación.

2.1.2.2 Obtención de imágenes

Para el proceso se utiliza una Kinect Xbox360 para ellos en Matlab se instala un complemento llamado “Kinect for Windows Sensor” cuya librería permite la detección y utilización de la Kinect como elemento de entrada de imágenes y de distancia para el algoritmo.

Para verificar que el sensor Kinect funciona adecuadamente se realizó una prueba de conexión de imagen y del sensor de profundidad infrarrojo obteniendo las siguientes imágenes, como se

observa en la Figura 7-2 la cámara funciona como cualquier otra cámara USB pero el sensor de profundidad e da como respuesta una matriz de 640x480 datos los cuales son la distancia en mm de cada pixel que para la calcular la distancia real se debe hacer otro tipo de conversión.

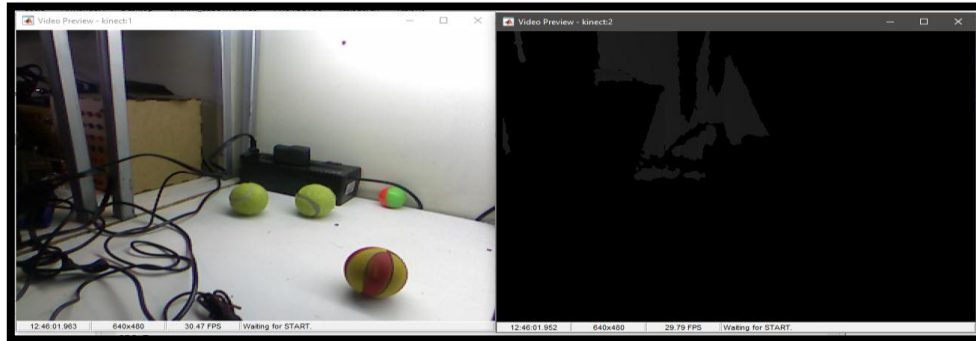


Figura 7-2. Imágenes preliminares obtenidas en por el Kinetic.

Realizado por: Guerrero, Caren, 2020

El código en Matlab para la conexión de la Kinect con el mismo se muestra en la Figura 8-2 a continuación se describirá la función de comando más importante:

- Imaqreset: este comando limpia cualquier otro tipo de conexión con otra cámara así asegurando que no haya problemas de conexión
- Imaqhwinfo: entrega toda la información de todas cámaras conectadas al ordenador
- Video input: inicia la conexión de la Kinect con Matlab
- confingColor: Obtener información para la configuración del disparador
- triggerconfig: Establecer activadores en manual para que sea posible trabajar con ellos
- FramesPerTrigger: Establecer cuadros por disparador. Un número mayor significa más cuadros cuando se activa.

```
ans =

    InstalledAdaptors: {'dcam'  'kinect'  'winvideo'}
    MATLABVersion: '8.5 (R2015a)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '4.9 (R2015a)'

Kinect Conectada
```

Figura 8-2. Conexión Kinect con Matlab.

Realizado por: Guerrero, Caren, 2020

2.1.2.3 Tratamiento de imágenes

Con las imágenes ya obtenidas gracias a la Kinect se empieza a adecuarlas para su ingreso al algoritmo de detección por lo cual se realiza una conversión de imagen de RGB a HSV como se observa en la Figura 9-2 para tener un mejor control sobre el color que se desea detectar, para la detección de pelotas de tenis las cuales son de tonalidad verde se asignó un rango de color de 200 a 230, ya obtenido una imagen solo de objetos de color verde se procede a la etapa de filtrado para la cual se aplica métodos morfológicos dentro del campo de visión artificial (erosión y dilatación) la cual se observa en la Figura 9-2, como resultado se consigue una imagen relativamente limpia como se muestra en la Figura 10-2 y apta para el ingreso al algoritmo de detección.

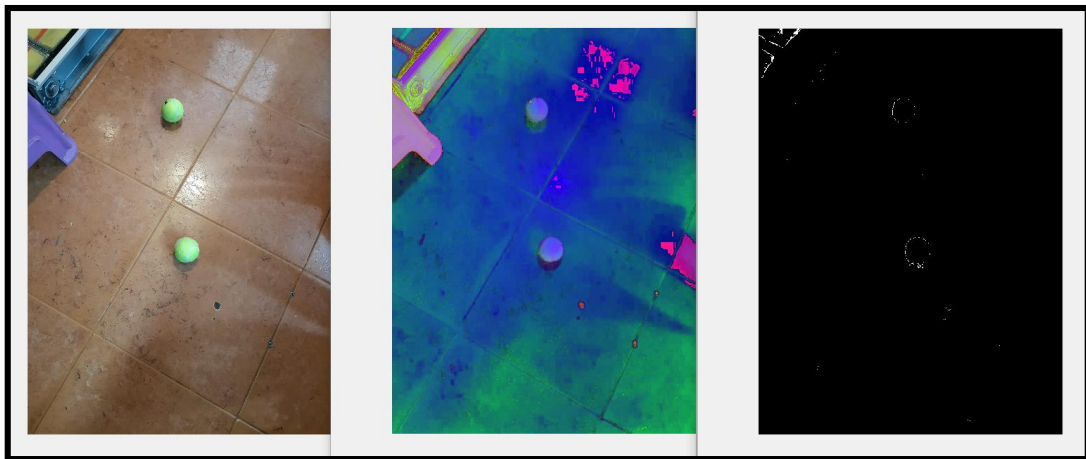


Figura 9-2. Tratamiento de las imágenes (Imagen Real, Imagen HSV, Imagen Segmentada).

Realizado por: Guerrero, Caren, 2020

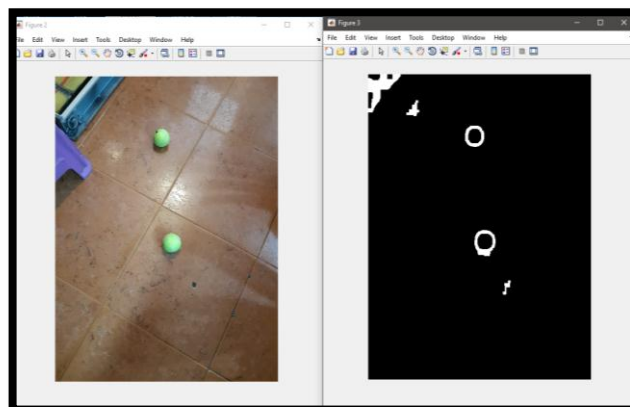


Figura 10-2. Filtrado y selección de imágenes

Realizado por: Guerrero, Caren, 2020

2.1.2.4 Algoritmo de Detección de Pelotas y localización.

Para comenzar con la detección de pelotas las imágenes antes procesadas o filtradas en la parte de tratamiento llegan a la entrada del algoritmo y al mismo tiempo toma la lectura de sensor de profundidad como se observa en la Figura 11-2, de la imagen se toma todas la características de las figuras o formas cerradas tanto su valor de área y su perímetro después de ello con los datos mediante un arreglo de vectores se obtiene un valor numérico el cual indica si la figura encerrada de la imagen tiene forma circular.

El valor numérico que en la Figura 11-2 toma el nombre de “IsCircle” no debe de superar el valor de 1.4, en la Tabla 6-2 se observa algunos valores con diferentes tipos de objetos, por ello se compara el valor de variable “IsCircle” y si es el objeto tiene una forma circular entra a otro ciclo de comparación de su área esto es para discriminar a los objetos que no tengan el tamaño de una pelota de tenis cabe recalcar que los objetos que sean distintos al color verde de las pelotas ya fueron descartados en la etapa de tratamiento.

Con el resultado de que el objeto es circular del tamaño adecuado se tiene que es una pelota de tenis y obtenemos las características del objeto para ser más específico se obtiene las coordenadas en pixeles del eje X, Y además también se tiene el valor de profundidad de dichos pixeles como se muestra en el diagrama se le nombra como Distancia Kinect el cual es un valor de suma importancia ya que con este dato se obtiene las coordenadas reales en centímetros del objeto.

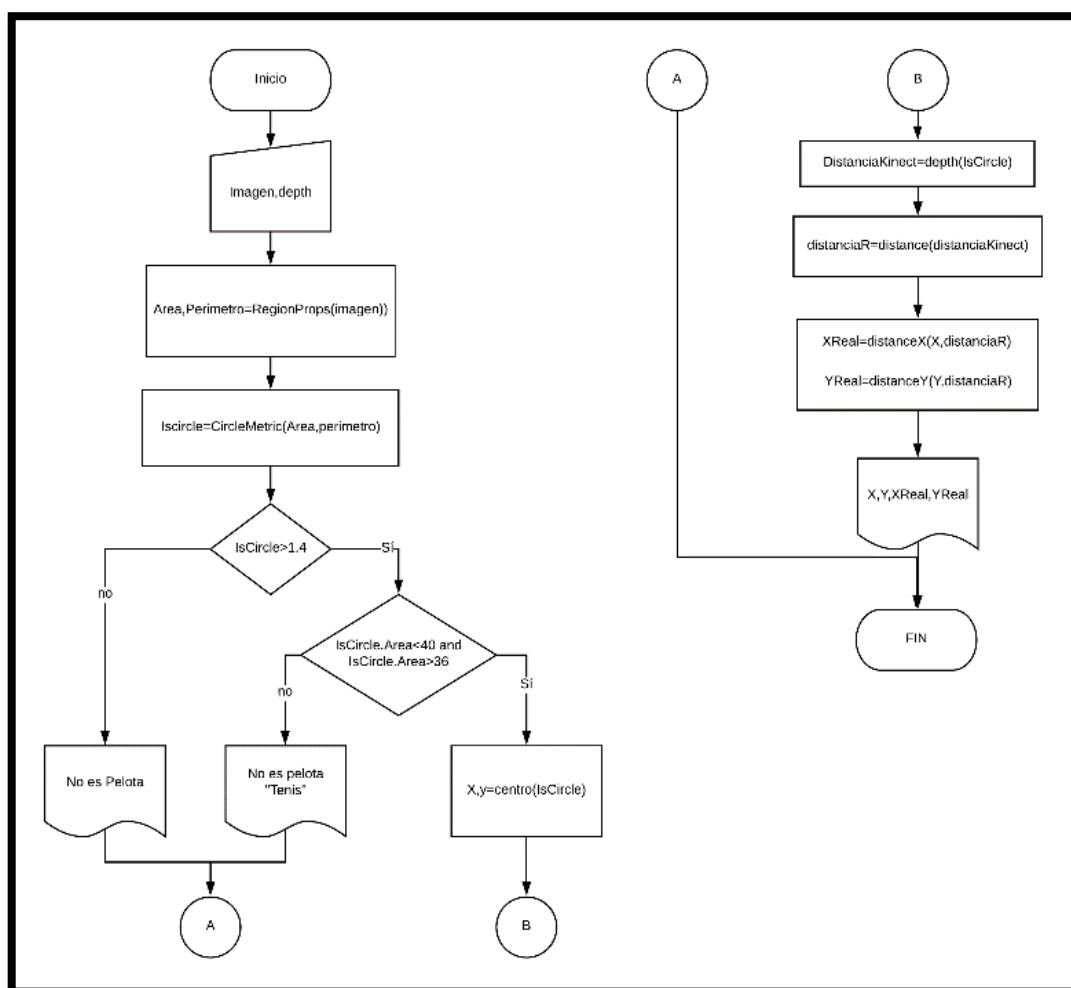


Figura 11-2. Diagrama de reconocimiento y localización de pelotas

Realizado por: Guerrero, Caren, 2020

Tabla 6-2. Tabla de objetos circulares y su área.

Objeto	IsCircle	Área
Pelota de tenis	1,3	40
Pelota de futbol	1,24	500
Objeto circular	1,09	200
Objeto cuadrado	3,6	600
Objeto Rectangular	2.63	200
Objeto sin forma definida	10,63	300

Realizado por: Guerrero, Caren, 2020

Como se muestra en la Figura 12-2, δ es la distancia real del objeto hacia el punto (0,0) de la Kinect y se obtiene mediante la aplicación de la Ecuación 1-2, ya adquiriendo dicho valor pueden

calcular las coordenadas (x, y) utilizando trigonometría básica y conociendo la resolución de la imagen que es de (640×480). Lo primero en tener en cuenta es el valor del ángulo máximo horizontal cuyo valor es de 57°, por lo tanto, se calcula la variación de ángulo por pixel $57^\circ/640px = 0,08901^\circ/px$, y verticalmente $43^\circ/480px = 0,08959^\circ/px$, teniendo el centro de la imagen en el punto (0, 0), se puede determinar el ángulo vertical y horizontal respecto al punto de la imagen mediante la Ecuación 2-2 y la Ecuación 3-2 respectivamente. Un valor adicional se calcula mediante la Ecuación 4-2 el cual me indica la distancia real en cm del objeto hacia el punto central de la cámara como se observa en la Figura 12-2. Con los valores obtenidos de α , β y d a continuación se procede a calcular las posiciones en X, Y ocupando la Ecuación 5-2 para el eje X y la Ecuación 6-2 para hallar las coordenadas en el eje Y logrando así la posición real del punto al origen. (Preciado & Forero, 2016, p.4-10)

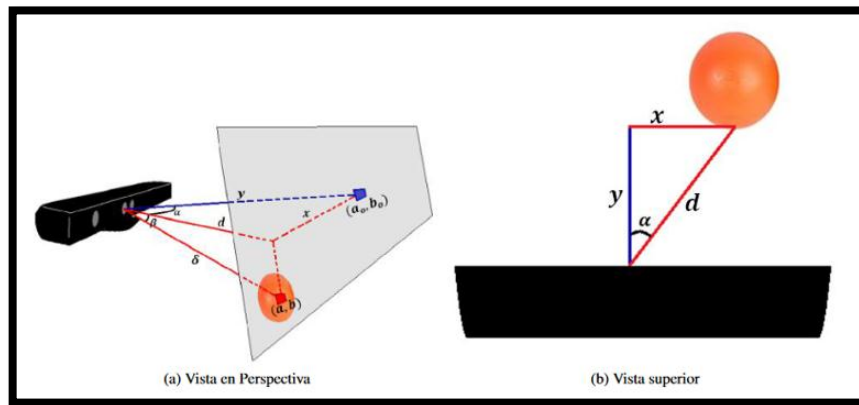


Figura 12-2. Esquema para encontrar las coordenadas del objeto

Realizado por: Guerrero, Caren, 2020

Ecuación 1-2. Ecuación para calcular distancia real del objeto hacia el plano de la cámara.

$$distanciaReal = \frac{1}{-2.8667 \times 10^{-3} * DistanciaKinect + 3.1547}$$

Donde,

DistanciaKinect = Distancia obtenida del sensor del Kinect dada en [mm].

Ecuación 1-2. Ecuación para calcular el ángulo con respecto al eje Y.

$$\beta = (b_0 - b) * 0.08906^\circ/px$$

Donde,

$b0$ = Posición del punto 0,0 en la coordenada Y dado en pixeles.

b = Posición del objeto en la coordenada Y dado en pixeles.

Ecuación 3-2. Ecuación para calcular el ángulo con respecto al eje X.

$$\alpha = (a0 - a) * 0.08959^\circ / px$$

Donde,

$a0$ = Posición del punto (0,0). en la coordenada X dado en pixeles.

a = Posición del objeto en la coordenada X dado en pixeles.

Ecuación 4-2. Ecuación para calcular la distancia real del objeto hacia el punto de origen de la cámara.

$$d = distanciaReal * \cos(\beta)$$

Donde,

$distanciaReal$ = Distancia real del objeto hacia el plano de la cámara.

β = Angulo con respecto al eje Y.

Ecuación 5-2. Ecuación para calcular la distancia real del objeto en el eje X.

$$XReal = d * \sin(\alpha)$$

Donde,

d = Distancia real del objeto hacia el punto de origen de la cámara.

α = Angulo con respecto al eje X.

Ecuación 6-2. Ecuación para calcular la distancia real del objeto en el eje Y.

$$Y_{Real} = d * \cos(\alpha)$$

Donde,

d = Distancia real del objeto hacia el punto de origen de la cámara.

α = Angulo con respecto al eje X.

2.1.2.5 Algoritmo de Navegación.

En el diagrama de la Figura 13-2, se puede observar los pasos que sigue el sistema una vez ya tenga los datos de posición real, comienza con la lectura de dichos datos después nuevamente lee una imagen de la cámara para verificar si el robot se encuadra en el área de trabajo para ello estratégicamente el robot tiene colores contrastantes con el ambiente por lo cual el robot se detecta por una segmentación por color ya obteniendo la posición en píxeles del robot y del objeto y a su vez detectando si este existe o está en la zona de trabajo pasa a un bucle infinito donde estará constantemente monitorizando la trayectoria del robot además envía los valores de ángulo del robot y del objeto hacia el robot por medio del sistema físico implementado que se explicará después.

Cuando la distancia del objeto y del robot son bastantes cercanas nuevamente el Matlab envía la orden de paro y a su vez la orden de recoger la pelota por medio de la herramienta. Una vez realizada la opción el robot empieza su regreso hacia el punto de origen para que el proceso si existiese un nuevo objeto pueda realizarse nuevamente. Cabe recalcar el sistema de movimiento con encoders y el imu MPU6050 será explicado más adelante en la programación interna del robot, además se debe tomar en cuenta el área de trabajo limitada del prototipo que es de 2.10 x 1.60 m.

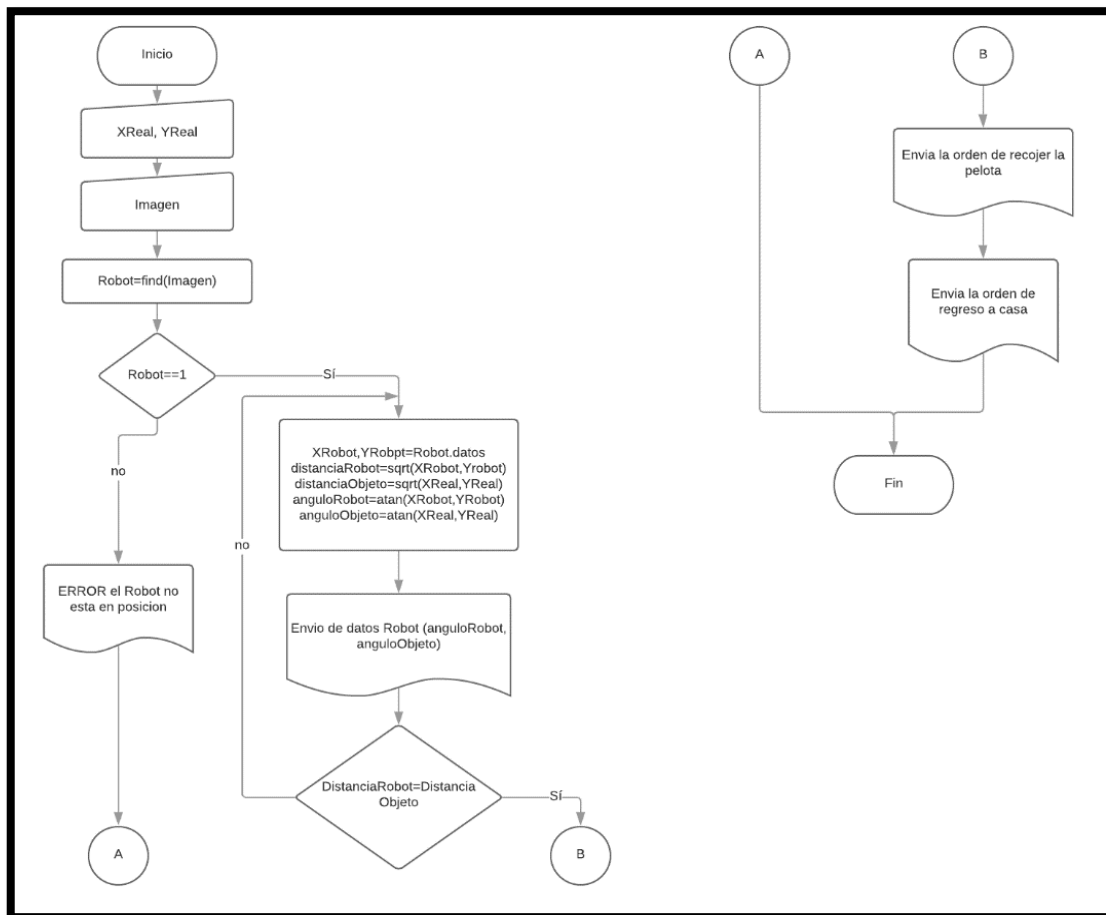


Figura 13-2. Diagrama de flujo sistema de navegación

Realizado por: Guerrero, Caren, 2020

2.1.2.6 Programación Arduino

El proceso que ejecuta Arduino es la inicialización de las variables, sensores y además de la antena para la comunicación inalámbrica, después pregunta si se desea calibrar el robot cabe recalcar que la calibración del robot se realiza solo una vez que se enciende este proceso dura aproximadamente 5 min.

Todo este proceso de calibración es básicamente colocar en el punto (0,0,0) el MPU es sus respectivos movimientos angulares y así tener una medida más precisa.

A continuación de la calibración entra en la etapa de control que se muestra en la Figura 14-2, en si donde el robot siempre se pregunta si existe una orden emitida por el emisor. Se tiene 5 órdenes básicas para el robot que se explicaran a continuación:

- Giro Derecha: en esta función se lee el valor de ángulo físico mediante el MPU6050 si el ángulo es menor los motores se enciende de forma tal que el robot gira hasta alcanzar el ángulo de control enviado por el Matlab.
- Giro Izquierda: en esta función se lee el valor de ángulo físico mediante el MPU6050 si el ángulo es menor los motores se enciende de forma tal que el robot gira hasta alcanzar el ángulo de control enviado por el Matlab.
- Avanzar: de tal manera que los anteriores movimientos el robot avanza una distancia X enviada por el Matlab, pero esta medida también es compara tanto por el sistema de visión como del robot en si gracias a sus encoders y se detiene cuando llega al punto destinado.
- Retro: El robot hace un movimiento hacia atrás una distancia X enviada por el Matlab, pero esta medida también es compara tanto por el sistema de visión como del robot en si gracias a sus encoders y se detiene cuando llega al punto destinado-
- Recoger: básicamente son secuencia de los 2 servos de la pinza para realizar el proceso de recolección.

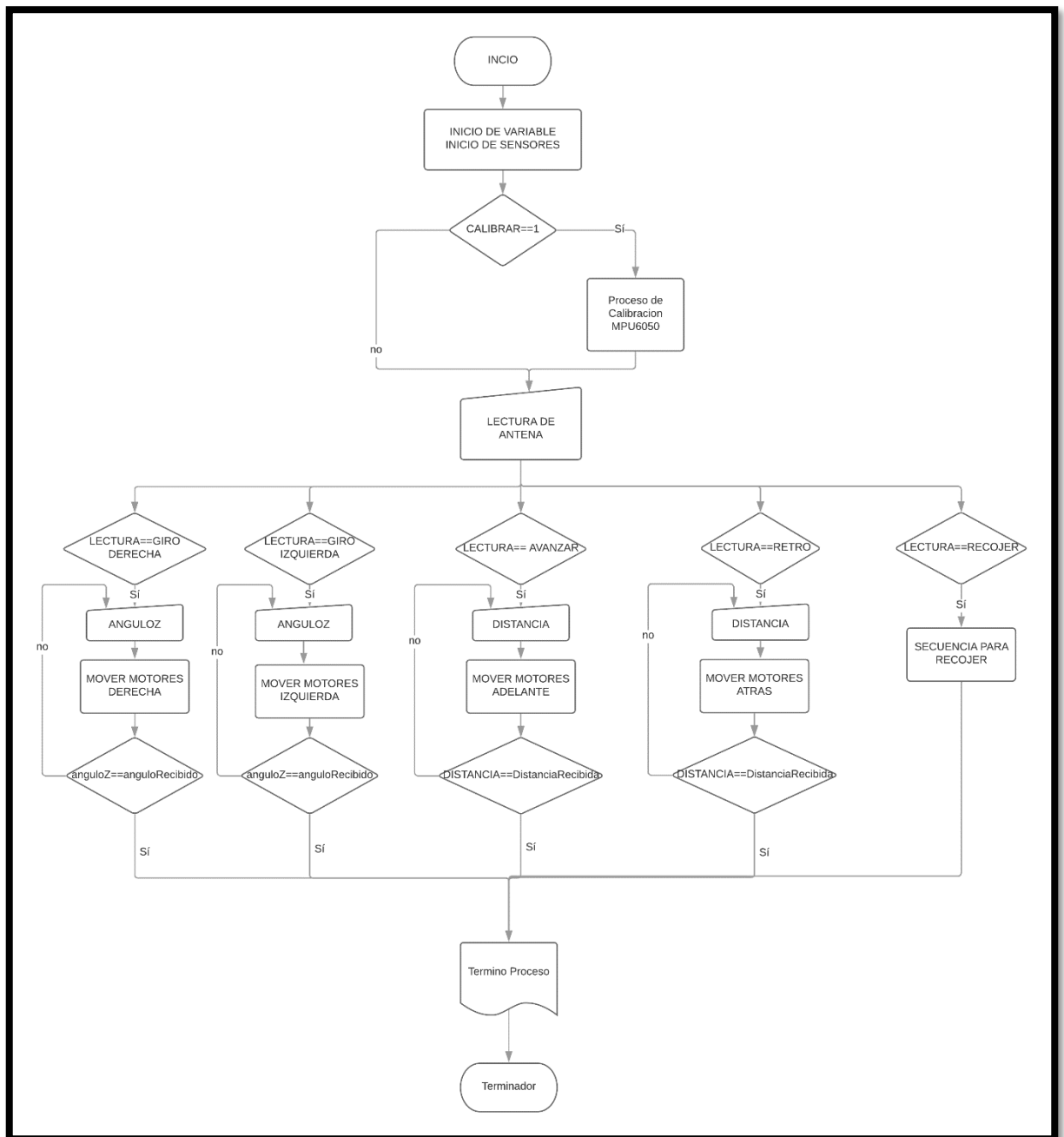


Figura 14-2. Diagrama de control Arduino Receptor (Robot)

Realizado por: Guerrero, Caren, 2020

Para el dispositivo emisor básicamente se comunica serial con Matlab y todos los datos que transmite el Matlab son enviados por medio de la antena NRF24L01 de tal manera que el dispositivo se transforma en un paso para la información como se observa en el diagrama de la Figura 15-2.

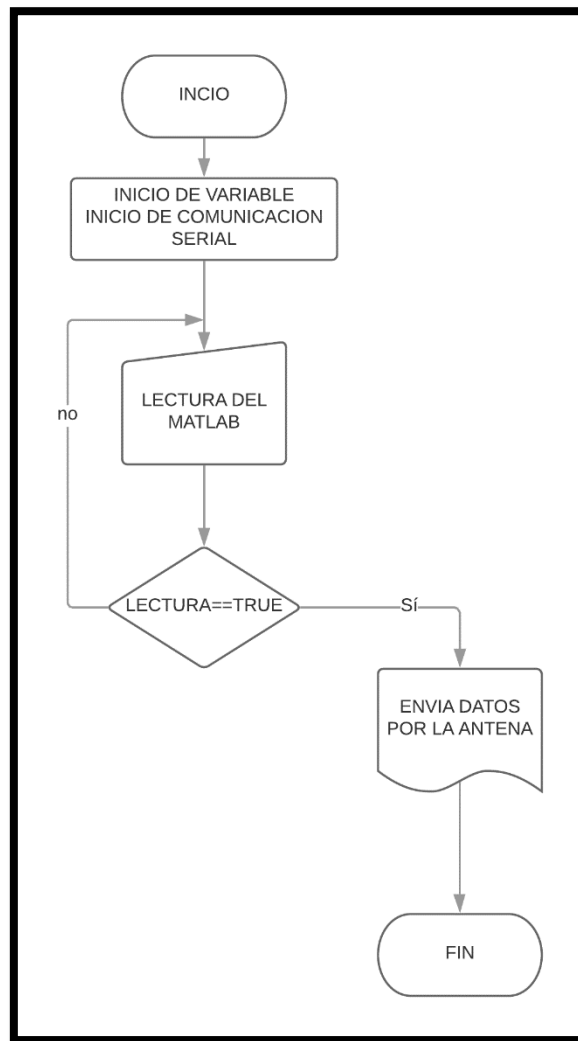


Figura 15-2. Diagrama del Emisor

Realizado por: Guerrero, Caren, 2020

2.1.2.7 Librería Arduino

Dentro de la programación en Arduino, se utiliza algunas librerías con el fin de facilitar la programación del prototipo, dichas librerías se describirán a continuación:

- *Servo.h*: esta librería se utiliza para el control de servomotores y como se describió anteriormente el robot utiliza estos motores como actuadores de traslación
- *I2Cdev.h*: esta librería inicia y sirve para la configuración de sensores que se comunique por medio de i2c al microcontrolador
- *MPU6050.h*: librería para usar el sensor MPU6050

- *Wire.h*: Esta librería se utiliza para comunicar la placa Arduino con dispositivos que trabajan mediante el protocolo I2C/TWI.
- *Math.h*: Mediante esta librería el Micronrolador puede usar funciones matemáticas como cos, sin, Atan, etc.
- *SPI.h*: contiene las funciones necesarias para controlar el hardware integrado de SPI
- *nRF24L01.h*: contiene funciones para controlar y configurar la antena NRF24L01
- *RF24.h*: librería adicional de ayuda para la librería principal de control de las antenas

En la Figura 16-2 se muestra el monitor serial con los datos al iniciar la placa desarrollada del robot.

```

COM3 (Arduino/Genuino Uno)

Enciendo Robot
Comprobando Direcciones I2C
Done.....
NRF24L01..... Activada
Esperando Calibracion MPU6050
MPU6050 connection successful
PID tuning Each Dot = 100 readings
>.....>.....
at 600 Readings

//          X Accel  Y Accel  Z Accel   X Gyro   Y Gyro   Z Gyro
//OFFSETS  -800,    -1825,   1376,    16,      111,    48

>.>.700 Total Readings

```

Figura 16-2. Monitor Serial Tarjeta desarrollada del robot

Realizado por: Guerrero, Caren, 2020

2.1.3 Interfaz del usuario en Matlab

Para que el usuario final no tenga problemas al momento de interactuar con el sistema se ha planteado una guide o aplicación básica que consta básicamente de un botón de inicio, un botón de stop y la pantalla de visualización para ello se describe la funcionalidad de cada uno de ellos, cabe recalcar que el sistema al momento de arrancar inicia automáticamente el sensor Kinect y el

placa emisora de datos del sistema, si no existe una de ella el programa o interfaz no inicia hasta que los 2 elemento estén conectados al ordenador. En la Figura 17-2 se muestra la pantalla principal del sistema.



Figura 17-2. Vista principal de la pantalla de usuario

Realizado por: Guerrero, Caren, 2020

El botón de inicio mostrado en la Figura 18-2, se encarga de iniciar todo el sistema de detección, localización y navegación y únicamente termina su proceso cuando ya no existe pelotas en el área de trabajo



Figura 18-2. Vista 2 botón inicio.

Realizado por: Guerrero, Caren, 2020

El botón de Paro mostrado en la Figura 19-2, se encarga de interrumpir todo el proceso tanto el de reconocimiento como el de navegación ya que en el botón se encuentra una instrucción de envío de paro también al robot



Figura 19-2. Vista 3 botón de paro.

Realizado por: Guerrero, Caren, 2020

La Pantalla VideoStream mostrada en la Figura 20-2, como ya se mencionó antes el sistema antes de arrancar la interfaz inicia el sensor Kinect esa toma de datos de imágenes se puede visualizar en esta pantalla que dentro del sistema sería una vista previa del lugar de trabajo en el cual podemos ver y calibrar el punto de origen del robot.



Figura 20-2. Vista de la cámara en tiempo real.

Realizado por: Guerrero, Caren, 2020

En la pantalla de visualización de los algoritmos de la Figura 21-2, es donde se apreciará visualmente la funcionalidad de todos los algoritmos antes diseñados e implementados y así saber el comportamiento del prototipo en tiempo real.



Figura 21-2. Vista de ruta que se debe trazar.

Realizado por: Guerrero, Caren, 2020

En la figura 22-2 se despliega Pantalla de grafico de coordenadas: en esta pantalla se graficará en tiempo real las coordenadas en centímetros que se encuentra el robot con respecto al objeto a recolectar.



Figura 22-2. Vista de descripción de coordenadas.

Realizado por: Guerrero, Caren, 2020

La Tabla de Datos de la Figura 23-2, alojara todos los datos sobresalientes de los objetos detectado como son número de objetos, localización tanto como en pixeles y centímetros reales, profundidad con respecto a la Kinect, distancia Real y en el caso de tiempo de recolección de cada objeto o pelota dichos datos se utilizarán para su respectivo análisis.



Figura 23-2. Vista 7 de la pantalla de usuario

Realizado por: Guerrero, Caren, 2020

2.2 Esquema de conexión del sistema

Se realizó un modelo esquemático mostrado en la Figura 24-2, de la conexión de los sensores y actuadores del robot con la finalidad de poder detectar las etapas y dividirlas para así poder realizar la calibración de los mismos y la reparación de componentes con la menor intervención global del prototipo.

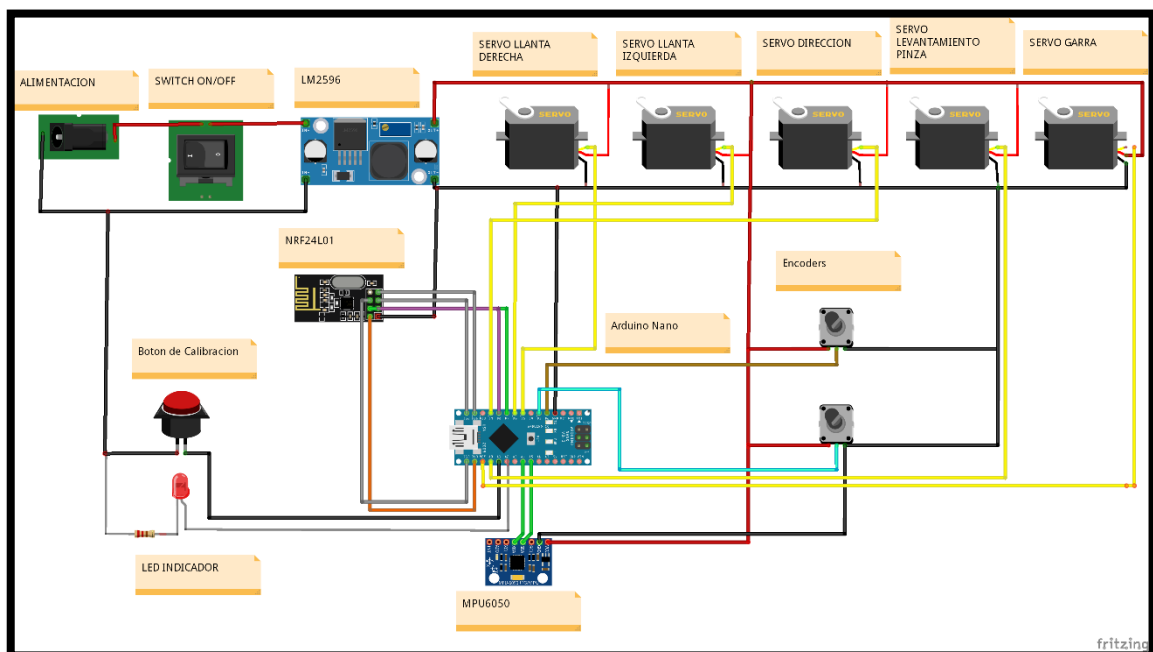


Figura 24-2. Esquema Electrónico del Robot +Receptor

Realizado por: Guerrero, Caren, 2020

Cuando se refiere al circuito emisor de datos mostrado en la Figura 25-2, se debe tomar en cuenta que el dispositivo está conectado en todo momento de ejecución del sistema con la finalidad de mantener al robot con la información más actual posible.

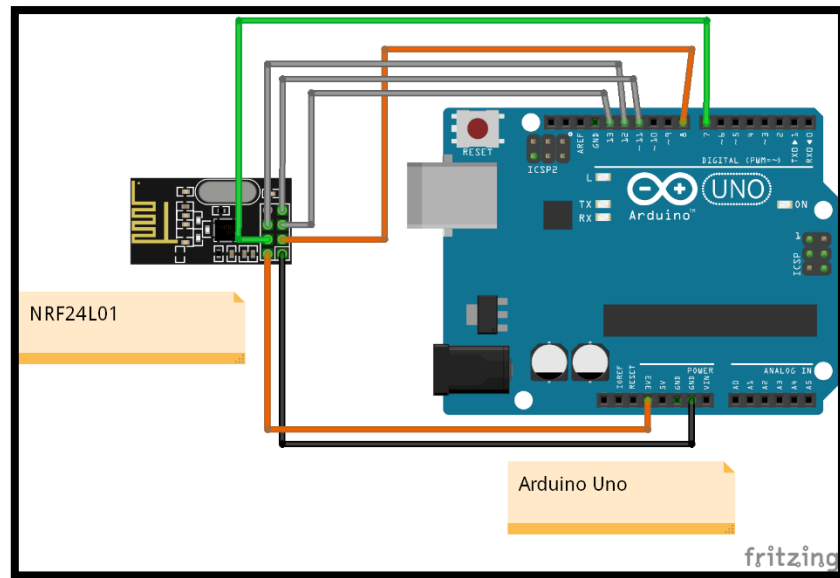


Figura 25-2. Esquema Electrónico Emisor

Realizado por: Guerrero, Caren, 2020

2.2.1 Pines de conexión Robot y Receptor (Arduino Nano)

En la Tabla 7-2 se describe la conexión de los elementos con respecto a los pines del Arduino Nano ubicado en el Robot.

Tabla 7-2. Tabla conexiones robot

PIN DEL MICROCONTROLADOR	ELEMENTO CONECTADO
2	Entrada encoder lado Izquierdo
3	Entrada encoder lado Derecho
4	Salida Motor Derecho
5	Salida Motor Izquierdo
7	CE NRF24L01 comunicación SPI
8	CNS NRF24L01 comunicación SPI
9	Salida Motor Dirección
11	MISO NRF24L01 comunicación SPI
12	MOSI NRF24L01 comunicación SPI

13	CLK NRF24L01 comunicación SPI
14 o A0	Salida Motor Elevador
15 o A1	Salida Motor Pinza
16 o A2	Entrada Pulsador Calibración
17 o A3	Salida Led indicador
18 o A4	Entrada SDA sensor MPU6050
19 o A5	Entrada SCL sensor MPU6050

Realizado por: Guerrero, Caren, 2020

2.2.2 Pines de conexión Emisor (Arduino Uno)

En la Tabla 8-2 se describe la conexión de los elementos con respecto a los pines del Arduino Uno ubicado en modulo Emisor.

Tabla 8-2. Tabla Conexiones Emisor

PIN DEL MICROCONTROLADOR	ELEMENTO CONECTADO
7	CE NRF24L01 comunicación SPI
8	CNS NRF24L01 comunicación SPI
11	MISO NRF24L01 comunicación SPI
12	MOSI NRF24L01 comunicación SPI
13	CLK NRF24L01 comunicación SPI

Realizado por: Guerrero, Caren, 2020

2.3 Diseño en del prototipo en SolidWorks

En la Figura 26-2 se puede observar el acabado final del robot mediante le software de diseño y simulación SolidWorks.

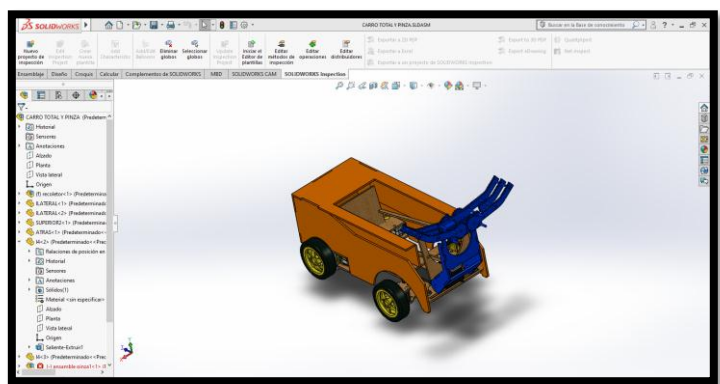


Figura 26-2. Vista del prototipo en SolidWorks

Realizado por: Guerrero, Caren, 2020

En la Figura 27-2 se observa la vista Superior del Robot y conjuntamente en la Tabla 9-2 se describe las medidas de largo y ancho.

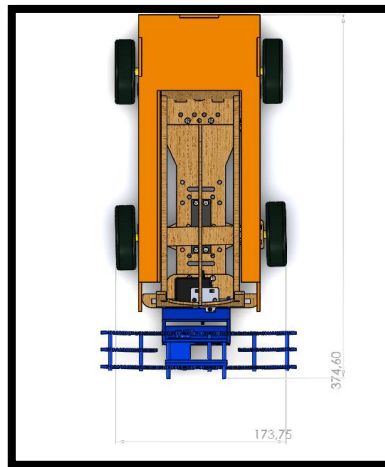


Figura 27-2. Vista Superior

Realizado por: Guerrero, Caren, 2020

Tabla 9-2. Medidas de la parte superior

MEDIDAS	mm
LARGO	374.60
ANCHO	173.75

Realizado por: Guerrero, Caren, 2020

En la Figura 28-2 se observa la vista lateral Izquierda del Robot y conjuntamente en la Tabla 10-2 se describe las medidas de largo y alto.

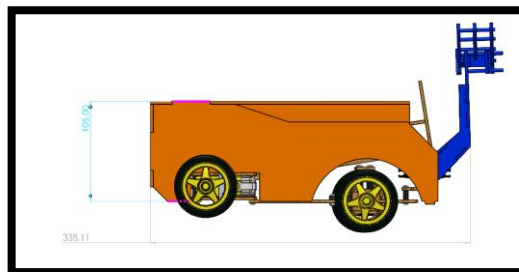


Figura 28-2. Vista Lateral Izq.

Realizado por: Guerrero, Caren, 2020

Tabla 10-2. Medidas de la parte lateral

MEDIDAS	mm
LARGO	338.11
ALTO	105.90

Realizado por: Guerrero, Caren, 2020

En la Figura 29-2 se observa la vista Isométrica del Robot

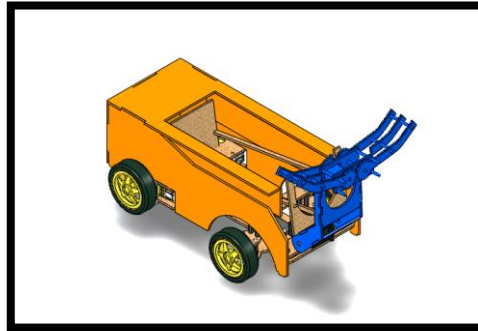


Figura 29-2. Vista General Isométrica.

Realizado por: Guerrero, Caren, 2020

Se muestra en la Figura 30-2, el robot ensamblado en su totalidad para ponerlo en funcionamiento.



Figura 30-2. Robot Ensamblado

Realizado por: Guerrero, Caren, 2020

CAPÍTULO III

3 VALIDACIÓN DEL PROTOTIPO

En este capítulo se detallará los valores obtenidos de las pruebas realizadas tanto en hardware como en software implementados para el funcionamiento del prototipo de robot móvil, mostrando la calibración de los sensores y actuadores además de la actualización y monitoreo de las imágenes e información enviada hacia la interfaz de usuario, finalmente se realizaron tablas comparativas de las diferentes pruebas realizadas.

Al tomar en cuenta el análisis tanto económico como de diseño se puede observar en la Gráfica 1-3.

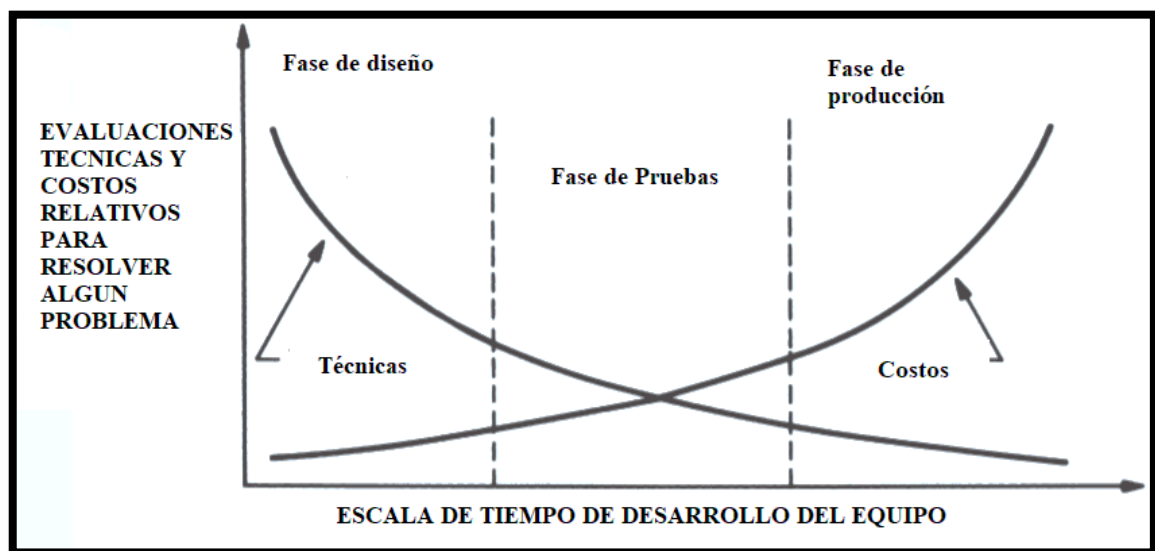


Gráfico 1-3. Escala de tiempo vs costo del desarrollo del equipo

Realizado por: Guerrero, Caren, 2020

En base a la información del Capítulo I se puede determinar los requerimientos necesarios para obtener una maquina recogedora de pelotas óptima los cuales son:

- Implementar un sistema móvil seguro y con la capacidad de poder levantar una pelota de tenis con la ayuda de una herramienta incorporada en el mismo.
- Configurar los parámetros necesarios en la kinetic y en el receptor móvil para tomar la mejor decisión en el movimiento odométrico

- Calibrar el sistema para poder disminuir el error que presentan los sistemas odométrico cuando la ruta de desplazamiento es considerablemente grande.
- La etapa de adquisición de imagen, así como la de envío y recepción de información tendrá la facilidad de informar continuamente al usuario sobre el estado actual del robot móvil.

3.1 Pruebas de detección de pelotas

En la Tabla 1-3, se verifica la eficiencia que tiene el algoritmo de detección con respecto a las pelotas y sus distractores para ello se realizó las pruebas en diferentes horas del día para tener una diferente tonalidad de luz además es todas las pruebas se colocaron 5 pelotas y 5 distractores en cada prueba obtenido un resultado del 95.238% de efectividad al encontrar las pelotas.

Además, se debe tener en cuenta que en la hora de la tarde es la hora donde el sistema tiene más problemas en detectar. De la misma forma se tiene que los distractores tienen un porcentaje del 3.810% de ser detectados que se puede decir que es un porcentaje bajo de detección lo que asegura una buena detección de los objetos como se puede apreciar en la Tabla 1-3.

Tabla 1-3. Valores obtenidos por las pruebas realizadas en campo.

Hora del día	No de pruebas	No Pelotas	Numero de distractores	Pelotas Detectadas	distractores detectados	Eficiencias pelotas detectadas (%)	Porcentaje de pelotas detectadas (%)
Mañana	1	5	5	5	0	100	0
	2			5	0	100	0
	3			5	0	100	0
	4			5	0	100	0
	5			5	0	100	0
	6			5	0	100	0
	7			5	0	100	0
Medio Día	8	5	5	5	0	100	0
	9			5	0	100	0
	10			5	0	100	0
	11			5	0	100	0
	12			5	0	100	0
	13			5	0	100	0
	14			5	0	100	0
	15	5	5	4	1	80	20

Tarde	16			5	0	100	0
	17			3	1	60	20
	18			5	1	100	20
	19			3	0	60	0
	20			5	0	100	0
	21			5	1	100	20
					Promedio	95,238 %	3,810 %

Realizado por: Guerrero, Caren, 2020

Al obtener estos resultados se realizó una gráfica con la finalidad de poder visualizar con mas exactitud la eficiencia del robot recogedor como se muestra en la Grafica 2-3.

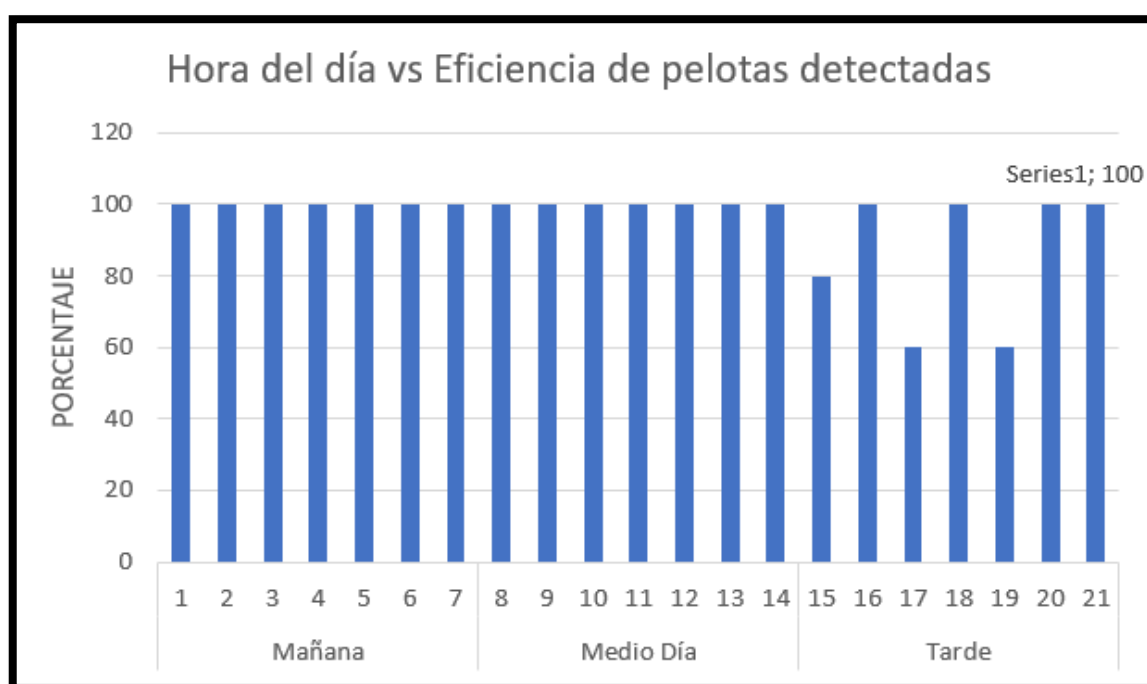


Gráfico 2-3. Eficiencia del robot en distintos tiempos

Realizado por: Guerrero, Caren, 2020

3.2 Pruebas de posicionamiento de pelotas

En la Tabla 2-3, los valores tomados en distintas horas del día para validar el algoritmo de visión artificial para la detección de pelotas se consigue obtener una columna de error porcentual definida tanto en el eje X como en el eje Y.

El error entre el valor dado por el sistema gracias a la cámara y la medición física que se realizó mediante un fluxómetro se llega a determinar un error en cada toma de datos que proporciona un

error promedio del 0.912% en el eje X y 1.736% promedio en el eje Y que da a conocer la precisión de sistema que sería del 99.088% del eje X y del 98.264% en el eje Y.

Tabla 2-3. Errores en las posiciones del vector en coordenadas X e Y

Hora del día	X Real (cm)	Y real (cm)	X medido (cm)	Y medido (cm)	Error X (%)	Error Y (%)
En la mañana	156	50	157,2	50,8	0,769	1,600
	122	60	121,5	61,4	0,410	2,333
	89	73	88,3	73,9	0,787	1,233
	87	103	88	103,8	1,149	0,777
	119	80	121,5	80,3	2,101	0,375
	111	31	110,7	33,5	0,270	8,065
Medio día	69	96	67,5	98,2	2,174	2,292
	48	82	48,6	82,3	1,250	0,366
	150	109	152,3	107,6	1,533	1,284
	139	63	139,4	64,6	0,288	2,540
	127	32	126,8	32,7	0,157	2,188
	118	106	116,8	106,6	1,017	0,566
	135	118	135,6	117,6	0,444	0,339
Tarde	122	81	120	81,4	1,639	0,494
	92	19	92,4	19,4	0,435	2,105
	61	103	60	103,7	1,639	0,680
	167	22	167,4	22,5	0,240	2,273
	166	112	166,8	112	0,482	0,000
	91	69	91,5	71,4	0,549	3,478
	43	99	45	99,6	4,651	0,606
				Promedio	0,912 %	1,736 %
				Eficiencia	99,088 %	98,264 %

Realizado por: Guerrero, Caren, 2020

Al obtener valores de posición real en el plano X y Y podemos estimar la desviación (error) que el robot puede llegar a tener, es necesario aclarar que el sistema de odometría presentará errores cada vez mayores cuando la distancia de recorrido es elevada, por lo tanto, como se muestra en el Gráfico 3-3 y el Gráfico 4-3, se puede ver claramente los puntos que el robot alcanzaron respecto a los valores reales y medidos en campo.

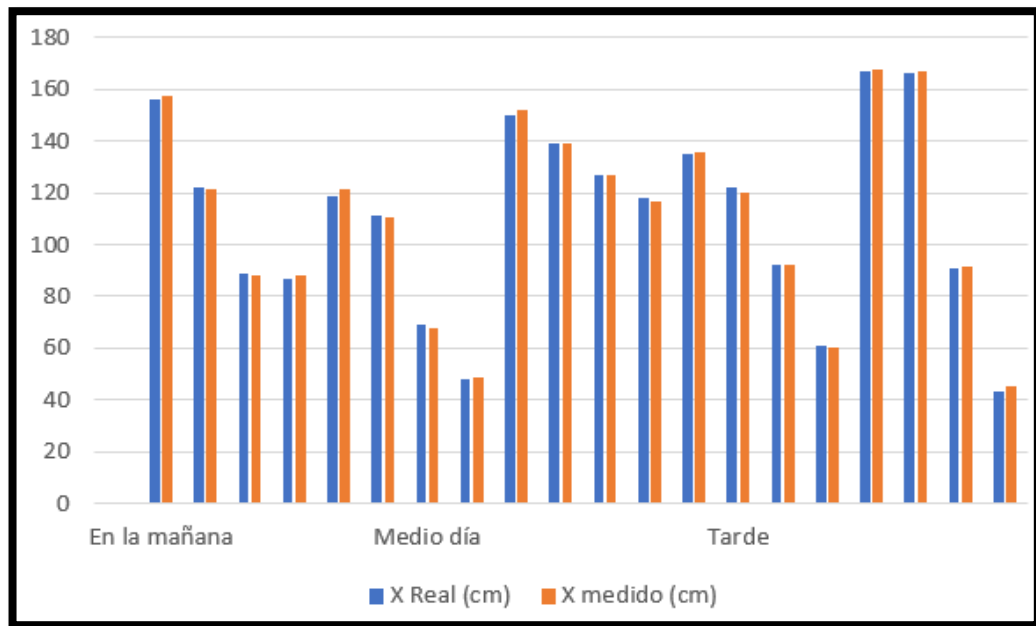


Gráfico 3-3. Comparación de posiciones real vs medida en el eje X

Realizado por: Guerrero, Caren, 2020

Y de igual manera se puede observar analíticamente la posición en el eje Y como muestra la Gráfica 4-3.

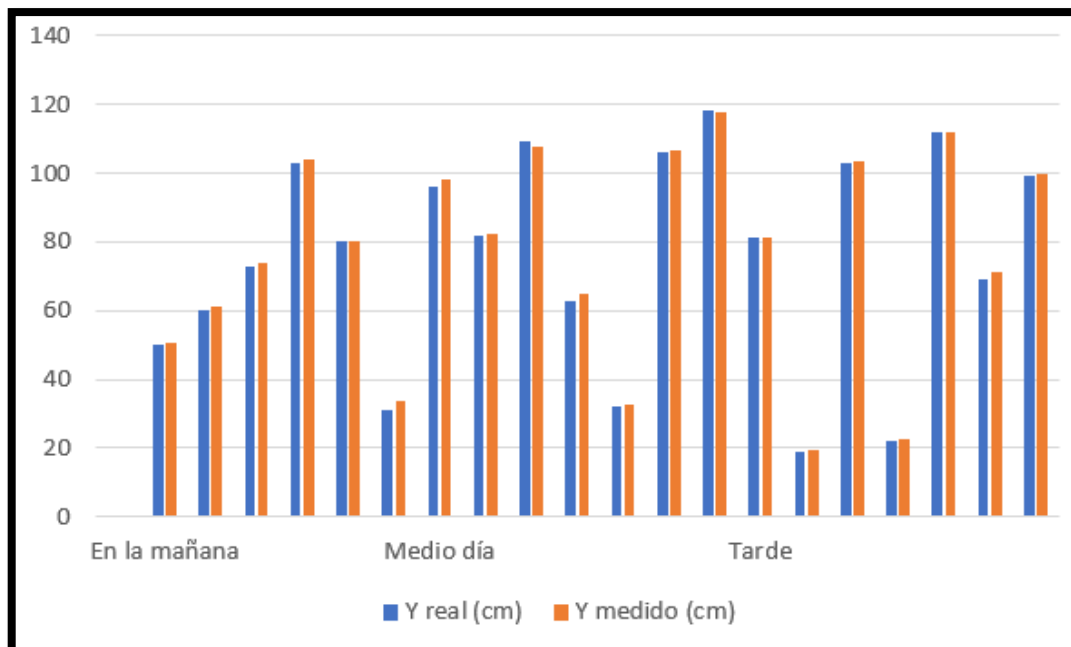


Gráfico 4-3. Comparación de posiciones real vs medida en el eje Y

Realizado por: Guerrero, Caren, 2020

En cada evaluación se tomaron las respectivas mediciones como muestra la Figura 1-3, con el fin de obtener una mayor cantidad de información para que sea analizada.

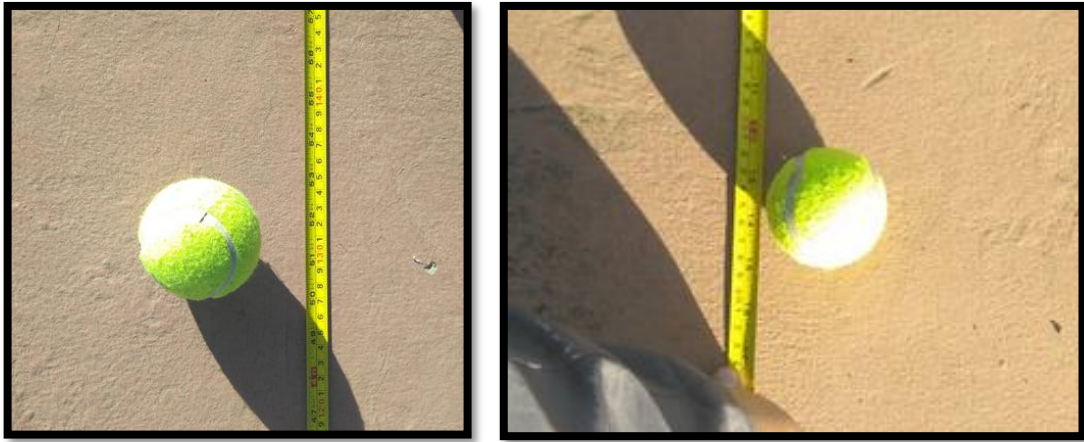


Figura 1-3. Mediciones de la distancia del objeto con respecto al robot.

Realizado por: Guerrero, Caren, 2020

3.3 Pruebas de Recolección

Esta es la prueba más importante que se realizó en el sistema para ello se desglosó en 5 escenarios diferentes, cada escenario tiene 6 pruebas realizadas, en cada prueba varía el número de distractores y se mantiene fijo el número de pelotas.

Por ejemplo, en el escenario 1 se tiene una pelota para la recolección y de esta manera se va aumentando el número de distractores hasta llegar a 6 distractores por cada escenario.

En cada escenario se obtuvo un promedio de efectividad, por lo general superior al 80% este porcentaje son de las pelotas recolectadas, además en la Tabla 3-3, se puede observar una eficiencia promedio total de todos los escenarios incluyendo el escenario ideal con 0 distractores dando como resultado una eficiencia del 96.9% de pelotas recolectadas por el robot. Además se realizaron pruebas adicionales las cuales se encuentran en el Anexo F de igual manera las pruebas se realizaron mediante escenarios y variando el número de distractores.

Tabla 3-3. Datos obtenidos de eficiencia en cada escenario.

ESCENARIO 1	No.	Número de Pelotas	Número de distractores	Pelotas Recolectadas	Eficiencia
	1	1	1	1	100
	2		2	1	100
	3		3	1	100
	4		4	1	100
	5		5	1	100
	6		6	1	100
					Eficiencia promedio
ESCENARIO 2	7	2	1	2	100
	8		2	2	100
	9		3	2	100
	10		4	2	100
	11		5	2	100
	12		6	2	100
					Eficiencia promedio
ESCENARIO 3	13	3	1	3	100
	14		2	3	100
	15		3	3	100
	16		4	2	66
	17		5	3	100
	18		6	2	66
					Eficiencia promedio
ESCENARIO 4	19	4	1	4	100
	20		2	4	100
	21		3	4	100
	22		4	4	100
	23		5	3	75
	24		6	4	100
					Eficiencia promedio
ESCENARIO 5 O IDEAL	25	4	0	4	100
	26			4	100
	27			4	100
	28			4	100
	29			4	100
	30			4	100
					Eficiencia promedio
				Eficiencia promedio Total	96,9%

Realizado por: Guerrero, Caren, 2020

Por lo que se puede ver que el sistema es claramente efectivo al tener un margen bajo de error la eficiencia siempre estará por los niveles óptimos de funcionamiento siempre y cuando no existan agentes que puedan dañar la integridad física del robot y de la cámara como se observa en la Grafica 5-3.

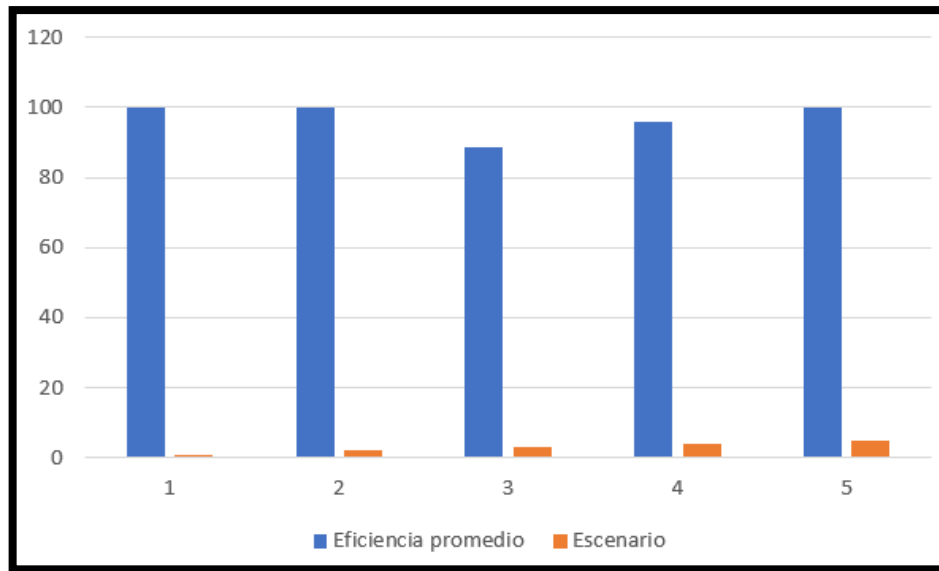


Gráfico 5-3. Valores porcentuales promedios de eficiencia en cada escenario

Realizado por: Guerrero, Caren, 2020

3.4 Análisis de costos

En la Tabla 4-3 se puede establecer el costo total estimado para la elaboración del proyecto.

Tabla 4-3. Costos y descripción de materiales.

Total	DESCRIPCIÓN	Valor UNITARIO	Valor TOTAL(USDS)
	Chasis del robot	150	150
	Impresiones 3D	150	220
5	servomotores SG90	5	5
10	pinzas sujetadoras (Impresas en filamento de 1,75mm)	5	50
1	Batería LIPO	80	80
1	microprocesador Arduino MEGA2560	40	40
10	motores reductores de 6v a 12v	5	50
1	cámara Kinetic XBOX 360	80	80

2	reguladores DC-DC 5v de 5A	10	20
1	PCA9685 de 16 canales (controlador de servomotores)	12	12
6	puentes H L298n de 12v	5	30
	Materiales de montaje y conexión	34	34
		TOTAL	\$ 771

Realizado por: Guerrero, Caren, 2020

CONCLUSIONES

- Se analizaron los distintos métodos de visión artificial para la detección de objetos basándose principalmente en segmentación por color y forma enfocándose en la detección de pelotas de tenis.
- Se determinó que el robot por ser un prototipo experimental tendrá que recoger 4 pelotas teniendo en cuenta el área de recolección es limitada debido a que la Kinect se encuentra en una posición fija.
- Se estableció los requerimientos más importantes para el diseño estructural del robot basándonos en las configuraciones de estos y la capacidad de pelotas que podrá contener el robot al momento de la recolección.
- Se implementó prototipo o robot cuyo funcionamiento se basa en la identificación de pelotas mediante visión artificial y aplicando sistemas de navegación mediante motores, encoders y giroscopio para la recolección de pelotas afuera del campo de tenis.
- La evaluación del sistema de visión artificial para el cálculo de las coordenadas reales en los ejes dando como resultado una precisión del 99.088% y el 98.264% en el eje X e Y respectivamente.
- Se determinó mediante las pruebas realizadas que el prototipo debido al hardware implementado específicamente el sistema de locomoción el robot en si es un poco lento al trasladarse de un lugar a otro, pero eso no es impedimento para que realice su tarea.
- Se determinó la eficiencia del robot con un valor promedio del 96.9% el cual se obtuvo mediante pruebas en distintos escenarios los cuales varían dependiendo el número de pelotas a recoger y el número de distractores.
- Finalmente, se consiguió el objetivo principal realizar un robot recolector con una alta precisión y eficiencia al recolectar pelotas aplicando diversos ámbitos de investigación que con lleva el amplio campo de la electrónica y visión artificial.

RECOMENDACIONES

- Uno de las principales limitantes en el prototipo es su limitada área de trabajo debido a que la cámara Kinect se encuentra en un lugar fijo para ello en un futuro se puede implementar un sistema de movimiento incluido a la cámara para cubrir más espacio de recolección en el campo.
- Mediante la prueba de medición y precisión del prototipo se llega que en la hora de la tarde es donde el sistema de visión puede a llegar a fallar por lo cual se recomienda utilizar el prototipo en la mañana o al medio día.
- Al fin de mejorar el número de pelotas recolectadas se puede modifica el sistema de almacenamiento del robot para que abarque un mayor número de pelotas en el proceso de recolección.
- Como un complemento de la recomendación anterior se puede añadir que se puede mejorar el sistema de traslación (Motores) para que el tiempo de recolección se a un más rápido y por ello más eficiente.
- Para tener un mayor control en la parte de visión artificial se puede ocupar otros métodos que con llevan entrenamiento como son algoritmos clasificadores e inclusive llegando a la implementación de neuronas artificiales.
- Para llegar al campo industrial con el prototipo se debe tener en cuenta los estándares que debe ser sometidos el mismo como son material, protecciones para el usuario y otros campos que no se han tomado en cuenta en la implementación del robot.

BIBLIOGRAFÍA

2ITF. *International Tennis Federation*. [en línea], 2019. [Consulta 09 de octubre de 2019]
Disponible en: <https://www.itftennis.com/media/221229/221229.pdf>

ARDUINO. *Arduino Nano: Más versátil que una navaja Suiza*. [en línea], 2018. [Consulta 12 de agosto de 2019]. Disponible en: <https://dronesdecarreras.com/arduino-nano-mas-versatil-que-una-navaja-suiza/>

ARDUINO. *Arduino Official Store / Boards Shields Kits Accessories*. [en línea], 2019. [Consulta 29 de noviembre de 2019]. Disponible en: <https://store.arduino.cc/>

ARDUINO. *Arduino—IntelGalileo*. [en línea], 2019. [Consulta 7 de noviembre de 2019]
Disponible en: <https://www.arduino.cc/en/ArduinoCertified/IntelGalileo>

BEGINNER RASPBERRY. *Beginners_Guide_v1*. [en línea], 2019. [Consulta 10 de noviembre de 2019]. Disponible en: https://www.raspberrypi.org/magpi-issues/Beginners_Guide_v1.pdf

BENAVIDES PINA. GCAFD. [en línea]. 2016. Disponible en:
https://buleria.unileon.es/bitstream/handle/10612/6635/BENAVIDES%20PINA_SAMUEL_2016_GCAFD.pdf?sequence=1

CAPITULO1. *Procesamiento Digital de imágenes*. [en línea]. 2015. Disponible en:
<http://dea.unsj.edu.ar/imagenes/recursos/Capitulo1.pdf>

CAPTACIÓN ENCODERS. *cas. qxpCaptacion09*. [en línea]. 2016. Disponible en:
http://www.fagorautomation.com/downloads/otros/es/conexion_estandar.pdf

CHAPTER4. *Segmentation*. [en línea]. 2015. Disponible en:
<http://www.bioss.ac.uk/people/chris/ch4.pdf>

CROBERTO FABRIS. *Come scegliere la macchina fotografica*. [en línea], 2017, [Consulta 10 de noviembre de 2019]. Disponible en:
<https://www.robbyfabrisdesign.it/come-scegliere-la-macchina-fotografica/>

CONTENEDORES DEL FORO DEL TENIS. *Contenedor de pelotas de tenis para entrenamientos.* [en línea]. 2017. [Consulta 26 de julio de 2019]. Disponible en: <https://www.forodeltenis.com/materiales-y-equipamiento/contenedor-de-pelotas-de-tenis-para-entrenamientos/?PHPSESSID=0048f22049f8987e0ed9d47b7f79bc59>

CRUZADO MUÑOZ. *Control y guiado de un robot móvil.* (tesis) (pregrado). [en línea]. 2017. Disponible en: <https://upcommons.upc.edu/handle/2117/108460>

DIE VORTEILE. *Programmierung mit NI LabVIEW - National Instruments.* [en línea]. 2018 [Consulta 10 de diciembre de 2019]. Disponible en: <https://www.ni.com/de-de/innovations/white-papers/13/benefits-of-programming-graphically-in-ni-labview.html>

FIRST STEPS IN MATLAB. *Ventana de inicio de MATLAB.* [En línea]. 2017. [Consulta 6 de diciembre de 2019]. Disponible en: https://www.researchgate.net/figure/Figura-15-Ventana-de-inicio-MATLAB_fig7_284070945

HOW DOES IT WORK *Inductive Position Sensors Offer Alternative for Mobile Robot Design.* [en línea]. 2018. [Consulta 2 de diciembre de 2019]. Disponible en: <https://www.roboticsbusinessreview.com/sponsored-content/rf-inductive-position-sensors-alternative-mobile-robot-design/>

INGENIERIA MECANICA. *Hola mundo en LABVIEW.* [en línea]. 2019. [Consulta 6 de diciembre de 2019]. Disponible en: <http://ingmecpt.blogspot.com/2019/07/hola-mundo-en-labview.html>

INSTALACION SOFTWARE. *Instalar OpenCV con Visual Studio 2017.* [en línea]. 2018. [Consulta 6 de diciembre de 2019]. Disponible en: <http://aprendiendoingenieria.es/instalar-opencv-con-visual-studio-2017/>

KNOWING HOW IT WORKS. *Stereoscopic Camera Works—Steve's Digicams.* [en línea]. 2017. [Consulta 19 de noviembre de 2019] Disponible en: <http://www.steves-digicams.com/knowledge-center/how-tos/photo-accessories/knowning-how-a-stereoscopic-camera-works.html#b>

L298N. *Motor Driver.* [en línea]. 2017. [Consulta 20 de octubre de 2019]. Disponible en: <http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf>

MAGNETIC ENCODERS *encoders for industrial robotics*. [en línea]. 2016. [Consulta 2 de noviembre de 2019] Disponible en: https://www.rls.si/en/magnetic-encoders-for-industrial-robotics?gclid=EAIaIQobChMI8vv8z9DK5QIVBOd3Ch3IRwlcEAAYASAAEgJFgfD_BwE

MERINO, *Two Eyes VR, una cámara estereoscópica con calidad 4K*. (tesis) (pregrado). [en línea]. 2018. Disponible en: <https://www.ticbeat.com/tecnologias/twoeyes-vr-una-camara-estereoscopica-con-calidad-4k/>

MICROCONTROLLER. *Microcontrolador PIC*. [en línea]. 2016. [Consulta 21 de octubre de 2019]. Disponible en: <http://www.conexaoinformatica.net.br/apostilas/Eletronica/Microcontroladores/Microcontrolador%20PIC.pdf>

MODULOS NRF. *Módulo NRF24L01 2.4 GHz*. [en línea]. 2017. [Consulta 6 de diciembre de 2019]. Disponible en: <https://naylampmechatronics.com/inalambrico/38-modulo-nrf24l01.html>

MROBOTICA COMPETITIVA. *Moto Reductor*. [en línea]. 2018. [Consulta 6 de diciembre de 2019]. Disponible en: <http://robotica-competitiva.jimdo.com/motores/>

MUÑOZ MANSO, *Sistema de visión artificial para la detección y lectura de matrículas*. (tesis) (pregrado). [en línea]. 2014. Disponible en: <https://uvadoc.uva.es/bitstream/handle/10324/11848/TFG-P-165.pdf;jsessionid=938F191A8FD19545A9B0459BE155EA30?sequence=1>

O QUE É ENCODER. *Para que serve? Como escolher? Como interfacear?* [en línea]. 2018. [Consulta 2 de noviembre de 2019] Disponible en: <https://www.hitecnologia.com.br/blog/o-que-%C3%A9-encoder-para-que-serve-como-escolher-como-interfacear/>

ODOMETRIA SYSTEM. *Odometria.pdf*. [en línea]. 2017. [Consulta 22 de octubre de 2019]. Disponible en: <http://www.kramirez.net/Robotica/Material/Presentaciones/Odometria.pdf>

OPEN CV. *Opencv2refman*. [en línea]. 2018. [Consulta 25 de octubre de 2019]. Disponible en: <https://docs.opencv.org/2.4/opencv2refman.pdf>
Pedersen, J. T. (2011). *Study group SURF: Feature detection & description*.

POSTS ENCODER. *Encoder 3b.* [en línea]. 2016. [Consulta 29 de noviembre de 2019]. Disponible en: https://www.hitecnologia.com.br/media/uploads/blog/posts/outros/encoder/post_encoder3b.png

PRECIADO, P. A. G., & FORERO. *Reconocimiento de color y posición con un sensor Kinect para aplicación de robótica móvil.* (tesis) (pregrado). [en línea]. 2016. Disponible en: <https://repository.javeriana.edu.co/handle/10554/21442>

PUENTE H ELECTRÓNICA DIGITAL. *Que es y cómo funciona el Driver L298 control con arduino.* [en línea]. 2016. [Consulta 6 de diciembre de 2019] Disponible en: <https://www.electrontools.com/Home/WP/2016/05/06/puente-h-con-driver-l298/>

PLASTICOS ALVAREZ. *Recogedora multiusos.* [en línea]. 2016. [Consulta 25 de julio de 2019]. Disponible en: http://riegosyplasticosalvarez.com/blog/24_ROLL-IN-RECOGEDOR-DE-NUECES

Radke, R., Andra, S., Al-Kofahi, O., & Roysam, B. *Image change detection algorithms: A systematic survey.* (artículo). [en línea]. 2005. Disponible en: <https://doi.org/10.1109/TIP.2004.838698>

RB04M022. *motor reductor de DC* [en línea]. 2017. [Consulta 10 de noviembre de 2019]. Disponible en: <http://www.agspecinfo.com/pdfs/R/RB04M022.PDF>

REDACCIÓN MICROCONTROLADORES. *Tipos de Arduino. Tabla de comparación de las placas más utilizadas.* [en línea]. 2018. [Consulta 6 de diciembre de 2019]. Disponible en: <https://descubrearduino.com/comparamos-varias-de-las-arduino-mas-utilizadas/>

ROBOTICA. *Robot SR1.* [en línea]. 2018. [Consulta 15 de agosto de 2019] Disponible en: http://www.superrobotica.com/sr1_Robot.htm

ROBOT-ADVANCED-KIT. *TI-RSLK Advanced Kit: TI-RSLK Maze Basic Kit with Wireless Communication Bluetooth & Wi-Fi Modules.* [en línea]. 2017. [Consulta 29 de noviembre de 2019] Disponible en: <https://www.newark.com/element14/robot-advanced-kit/advanced-robotic-kit-education/dp/43AC8494>

SAHOO, S. K., TANG, D., & DANG. *Single-shot multispectral imaging with a monochromatic camera. Optica.* (tesis) (pregrado). [en línea]. 2015. Disponible en: <https://doi.org/10.1364/OPTICA.4.001209>

SHABALINA, K., SAGITOV, A., & MAGID, E. *Comparative Analysis of Mobile Robot Wheels Design.* (tesis) (pregrado). [en línea]. 2018. Disponible en: <https://doi.org/10.1109/DeSE.2018.00041>

SZELISKI. *Computer Vision Algorithms and Applications.* (tesis) (pregrado). [en línea]. 2016. Disponible en: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf

Tabares y Arcila. *SISTEMA AUTÓNOMO PARA RECOLECCIÓN DE BOLAS DE TENIS.* (tesis) (pregrado). [en línea]. 2018. Disponible en: <http://repository.udistrital.edu.co/bitstream/11349/3422/1/Gait%C3%A1nTabaresDavidLeonardo2015.pdf>

TAPIA, C., & MANZANO, H. *Evaluación De La Plataforma Arduino E Implementación De Un Sistema De Control De Posición Horizontal.* (tesis) (pregrado). [en línea]. 2013. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/5522/1/UPS-GT000511.pdf>

TEACH, LEARN, AND MAKE WITH RASPBERRY PI. *Raspberry Pi.* [en línea]. 2016. [Consulta 29 de noviembre de 2019] Disponible en: <https://www.raspberrypi.org>

TENNIBOT. *How to use tennibot.* [en línea]. 2017. [Consulta 25 de julio de 2019] Disponible en: <http://tennibot.com>

TENNIS. *History of Tennis at the Olympic Games.* [en línea]. 2018. [Consulta 12 de noviembre de 2019]. Disponible en: <https://www.olympic.org/news/a-brief-history-of-tennis>

SANTIAGO TRUJILLO. *“La Metodología De Enseñanza En La Práctica Del Tenis De Campo En Los Estudiantes Del Centro De Educación Física De La Escuela Superior Politécnica De Chimborazo.* (tesis) (pregrado). [en línea]. 2015. Disponible en: <https://repositorio.uta.edu.ec/jspui/bitstream/123456789/17209/1/Tesis%20Final%202015%20Santiago%20Trujillo.pdf>

TI-ROBOTICS. *System-Learning-Kit.* [en línea]. 2016. [Consulta 6 de septiembre de 2019] Disponible en: <https://university.ti.com/en/faculty/ti-robotics-system-learning-kit/ti-robotics->

system-learning-kit?utm_source=google&utm_medium=cpc&utm_campaign=corp-uni-null-rslk_build-cpc-lp-google-
wwe&utm_content=rslk_build&ds_k=ti+robot+kit&DCM=yes&DCM=yes&gclid=EAIaIQobChMI2tD-s-i65AIVAxgMCh3ISwoJEAAYASAAEgJGt_D_BwE&gclsrc=aw.ds

TENNIS SPORT REGRAS. *Todo sobre Tenis [Historia, Reglas, ...]*. [en línea]. 2018. [Consulta 20 de julio de 2019] Disponible en: <https://sportsregras.com/es/tenis-historia-reglas/>

TOP5 GEEEXELS. *Ventajas y Desventajas del Xbox One Sin Kinect!*. [en línea]. 2016. [Consulta 6 de diciembre de 2019] Disponible en: <https://geexels.com/videojuegos/top5-ventajas-y-desventajas-del-xbox-one-sin-kinect/>

VACA PAREDES, “*ROBOT MÓVIL PARA INVESTIGACIÓN EN ALGORITMOS DE PLANEAMIENTO DE RUTAS: SISTEMA DE ODOMETRÍA*”. (tesis) (pregrado). [en línea]. 2017. Disponible en: <http://repositorio.utn.edu.ec/bitstream/123456789/7908/1/04%20MEC%20211%20TRABAJO%20DE%20GRADO.pdf>

VEINTIMILLA, D., & SIGUENCIA, F. *Diseño de un sistema inteligente de parqueo vehicular mediante videograbación e implementación de un prototipo de prueba para la FIEE*. (tesis) (pregrado). [en línea]. 2014. Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/7853>

VILLEGAS Y LÓPEZ. *Prototipo de robot recolector de objetos*. (tesis) (pregrado). [en línea]. 2016. Disponible en: <http://tlamati.uagro.mx/t7e1/83.pdf>

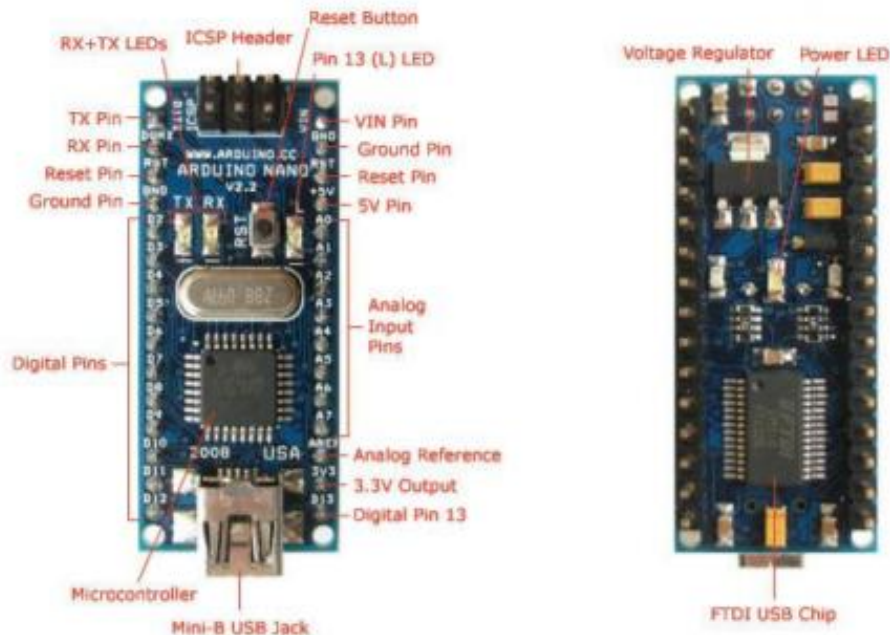
WEB MECATRONICALATAM. *What is Arduino?* [en línea]. 2016. [Consulta 10 de junio de 2019]. Disponible en: <https://www.mecatronicalatam.com/en/tutorials/arduino/que-es>

WEB ELECTRONIC NOTES. *Graphical System Design*. [en línea]. 2018. [Consulta 2 de diciembre de 2019]. Disponible en: <https://www.electronics-notes.com/articles/test-methods/labview/what-is-labview.php>

WEBSITE MATLAB. *What is MATLAB?* [en línea]. 2019. [Consulta 7 de noviembre de 2019]. Disponible en: <https://la.mathworks.com/discovery/what-is-matlab.html>

ZEA, A., & ALBERTO, L. *Diseño e implementación de un robot móvil con Control de trayectoria mediante principios odométricos. Repositorio de Tesis - UNMSM.* (tesis) (pregrado). [en línea]. 2016. Disponible en: <http://cybertesis.unmsm.edu.pe/handle/cybertesis/4440>

ANEXO A: Especificaciones en Arduino Nano



Schematic and Design

Arduino Nano 3.0 (ATmega328): [schematic](#), [Eagle files](#).

Arduino Nano 2.3 (ATmega168): [manual](#) (pdf), [Eagle files](#). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Specifications:

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

Power:

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.



SKU: MDU-1049

Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
 - Low: $-0.3V \leq V_{in} \leq 1.5V$.
 - High: $2.3V \leq V_{in} \leq V_{ss}$.
- Enable signal input voltage range :
 - Low: $-0.3 \leq V_{in} \leq 1.5V$ (control signal is invalid).
 - High: $2.3V \leq V_{in} \leq V_{ss}$ (control signal active).
- Maximum power consumption: 20W (when the temperature $T = 75\text{ }^{\circ}\text{C}$).
- Storage temperature: $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

ANEXO C: Características del motorreductor

Motor reductor de DC (6V)

Número de parte: RB-04M022



Introducción.

Motor reductor con gran sensibilidad y compatibilidad con microcontroladores (Arduino), puede emplearse PWM (control de velocidad), además de ser efectivos cuando se busca alta velocidad de respuesta en pequeñas aplicaciones.

Especificaciones:

- Engranajes internos: 48:1
- Velocidad sin carga (3V): 120 rpm
- Velocidad sin carga (6V): 240 rpm
- Corriente sin carga (3V): 40mA
- Corriente sin carga (6V): 70mA
- Fuerza de torque (3V): 3200mg*cm
- Fuerza de torque (6V): 5500mg*cm
- Dimensiones: 70.50mm x 27mm x 23mm
- Peso: 40g

Electronica S.A. de C.V.

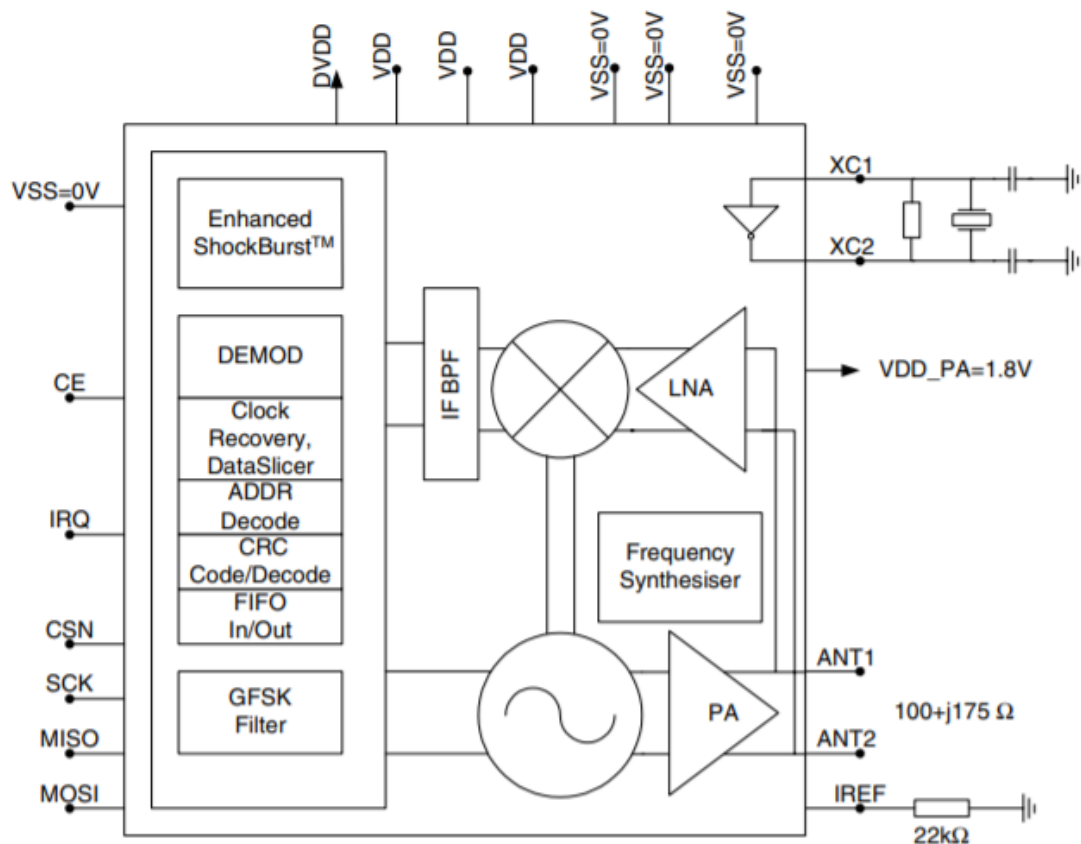
|

QUICK REFERENCE DATA

Parameter	Value	Unit
Minimum supply voltage	1.9	V
Maximum output power	0	dBm
Maximum data rate	2000	kbps
Supply current in TX mode @ 0dBm output power	11.3	mA
Supply current in RX mode @ 2000 kbps	12.3	mA
Temperature range	-40 to +85	°C
Sensitivity @ 1000 kbps	-85	dBm
Supply current in Power Down mode	900	nA

Type Number	Description	Version
nRF24L01	20 pin QFN 4x4, RoHS & SS-00259 compliant	D
nRF24L01 IC	Bare Dice	D
nRF24L01-EVKIT	Evaluation kit (2 test PCB, 2 configuration PCB, SW)	1.0

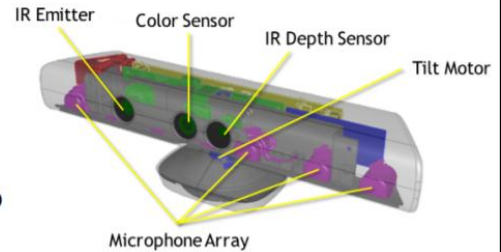
BLOCK DIAGRAM



ANEXO E: Características del Kinetic

Components

- Color sensor:
 - common camera sensor to acquire RGB images
- Depth camera:
 - IR emitter + IR sensor use infrared light to acquire depth images
 - V1: Structured Light / Coded Light
 - V2: Time of Flight
- Audio sensor:
 - Microphone array to capture sound signal and sound position



Feature	V1	V2
Color stream	640x480 at 30FPS	1920x1280 at 30FPS
Depth stream	320x240	512x424
Infrared stream	---	512x424
Depth distance	0.4m (near mode) – 4m	0.4m – 4.5m
Horizontal field of view	57°	70°
Vertical field of view	43°	60°
# defined skeleton joints	20	25
# bodies tracked	2 (+4 recognized)	6
Operating system	Windows 7 / 8	Windows 8

ANEXO F: Pruebas del Robot

ESCENARIO1 1 Pelotas				
No	1P0D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 1P0D
1	0	1	0	100,00
2	0	1	0	100,00
3	0	1	0	100,00
4	0	1	0	100,00
5	0	1	0	100,00
6	0	1	0	100,00
No	1P1D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 1P1D
1	1	1	0	100,00
2	1	1	0	100,00
3	1	1	0	100,00
4	1	1	0	100,00
5	1	1	0	100,00
6	1	1	0	100,00
No	1P2D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 1P2D
1	2	1	0	100,00
2	2	1	0	100,00
3	2	1	0	100,00
4	2	0	1	0,00
5	2	0	1	0,00
6	2	1	0	100,00
No	1P3D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 1P3D
1	2	1	0	100,00
2	2	1	0	100,00
3	2	1	0	100,00
4	2	0	1	0,00
5	2	1	0	100,00
6	2	1	0	100,00
No	1P4D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 1P4D
1	2	0	1	0,00
2	2	0	1	0,00
3	2	1	0	100,00
4	2	1	0	100,00
5	2	1	0	100,00
6	2	1	0	100,00
			Eficiencia Promedio	79,17

ESCENARIO2		2 Pelotas		
No	2P0D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 2P0D
1	0	2	0	100,00
2	0	2	0	100,00
3	0	2	0	100,00
4	0	2	0	100,00
5	0	2	0	100,00
6	0	2	0	100,00
No	2P1D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 2P1D
1	1	2	0	100,00
2	1	2	0	100,00
3	1	2	0	100,00
4	1	2	0	100,00
5	1	2	0	100,00
6	1	2	0	100,00
No	2P2D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 2P2D
1	2	2	0	100,00
2	2	2	0	100,00
3	2	1	1	50,00
4	2	1	1	50,00
5	2	1	1	50,00
6	2	2	0	100,00
No	2P3D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 2P3D
1	2	2	0	100,00
2	2	1	1	50,00
3	2	2	0	100,00
4	2	2	0	100,00
5	2	2	0	100,00
6	2	2	0	100,00
No	2P4D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 2P4D
1	2	1	1	50,00
2	2	1	1	50,00
3	2	2	0	100,00
4	2	1	1	50,00
5	2	1	1	50,00
6	2	2	0	100,00
			Eficiencia Promedio	83,33

ESCENARIO3 3 Pelotas				
No	3P0D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 3P0D
1	0	3	0	100,00
2	0	3	0	100,00
3	0	3	0	100,00
4	0	3	0	100,00
5	0	3	0	100,00
6	0	3	0	100,00
No	3P1D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 3P1D
1	1	3	0	100,00
2	1	3	0	100,00
3	1	3	0	100,00
4	1	3	0	100,00
5	1	3	0	100,00
6	1	3	0	100,00
No	3P2D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 3P2D
1	2	3	0	100,00
2	2	3	0	100,00
3	2	2	1	66,67
4	2	2	1	66,67
5	2	3	0	100,00
6	2	3	0	100,00
No	3P3D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 3P3D
1	2	3	0	100,00
2	2	2	1	66,67
3	2	3	0	100,00
4	2	2	1	66,67
5	2	2	1	66,67
6	2	3	0	100,00
No	3P4D	Pelotas Recogidas	Pelotas No recogidas	Eficiencia 3P4D
1	2	2	1	66,67
2	2	1	2	33,33
3	2	3	0	100,00
4	2	3	0	100,00
5	2	1	2	33,33
6	2	3	0	100,00
			Eficiencia Promedio	86,11

ESCENARIO4		4	Pelotas		
No	4P0D		Pelotas Recogidas	Pelotas No recogidas	Eficiencia 4P0D
1		0	4	0	100
2		0	4	0	100
3		0	4	0	100
4		0	4	0	100
5		0	4	0	100
6		0	4	0	100
No	4P1D		Pelotas Recogidas	Pelotas No recogidas	Eficiencia 4P1D
1		1	4	0	100
2		1	4	0	100
3		1	4	0	100
4		1	4	0	100
5		1	4	0	100
6		1	4	0	100
No	4P2D		Pelotas Recogidas	Pelotas No recogidas	Eficiencia 4P2D
1		2	3	1	75
2		2	4	0	100
3		2	4	0	100
4		2	3	1	75
5		2	4	0	100
6		2	4	0	100
No	4P3D		Pelotas Recogidas	Pelotas No recogidas	Eficiencia 4P3D
1		2	4	0	100
2		2	3	1	75
3		2	4	0	100
4		2	4	0	100
5		2	2	2	50
6		2	4	0	100
No	4P4D		Pelotas Recogidas	Pelotas No recogidas	Eficiencia 4P4D
1		2	3	1	75
2		2	3	1	75
3		2	4	0	100
4		2	4	0	100
5		2	2	2	50
6		2	4	0	100
				Eficiencia Promedio	90,63