

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA

ROBOT MOVIL OPERADO A DISTANCIA

Trabajo de Titulación para optar al Título de
TÉCNICO UNIVERSITARIO EN ELECTRÓNICA

Alumno:

Miguel Andrés González Vergara

Profesor Guía:

Mag. Guelis Darío Montenegro Zamora

Profesor Correferente:

Mag. Loreto Elisa Marín Carcey

2021

RESUMEN

KEYWORDS: ROBÓTICA, COVID-19, TELEOPERADO, ARDUINO, VR.

En el presente informe de proyecto de título, se describirá el contexto actual de desarrollo de la tecnología robótica, y junto con esto, se mencionarán algunos problemas actuales surgidos por el COVID-19. La exposición, plantea que hay ciertos tipos de trabajo que podrían eventualmente ser realizados de forma remota con la tecnología robótica adecuada. En este caso, se busca traspasar a modo remoto con esta tecnología, cualquier tipo de trabajo que implique inspeccionar o caminar por un espacio, y explorarlo con la vista.

Posteriormente se abarcará el desarrollo de un prototipo de robot móvil teleoperado, a través de internet y con un control intuitivo de lo que el robot percibe, además que sea capaz de desplazarse por espacios de superficie plana, según lo comande el usuario.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: ROBOTS, SEGURIDAD Y TRABAJO REMOTO	2
1. ROBOTS, SEGURIDAD Y TRABAJO REMOTO	3
1.1. COVID-19	4
1.2. PROPUESTA DE SOLUCIÓN	5
1.3. COMPONENTES DEL ROBOT	6
1.3.1. LattePanda	6
1.3.2. Arduino Mega 2560	7
1.3.3. Arduino Nano	8
1.3.4. Placas Shields.	9
1.3.4.1 Cervical Shield	9
1.3.4.2 Lumbar Shield	10
1.3.5. Webcam Logitech C270	10
1.3.6. Motor 12 V Alto Torque	11
1.3.7. Servomotor DS3218	12
1.3.8. Batería	13
1.4. COMPONENTES DEL CONTROL	14
1.5. DIAGRAMA DE BLOQUES	15
1.5.1. Diagrama Control Remoto	16
1.5.2. Diagrama Robot	17
1.6. OBJETIVOS DE PROYECTO	18
1.6.1. Objetivo General	18
1.6.2. Objetivos Específicos	18
CAPÍTULO 2: DESARROLLO, CONEXIÓN Y PROGRAMACIÓN DEL PROYECTO	18
2. DESARROLLO, CONEXIÓN Y PROGRAMACIÓN DEL PROYECTO	19
2.1. DISEÑO FÍSICO Y MECÁNICO DEL ROBOT	19
2.1.1. Base Móvil	19

2.1.1.1.	Modelo de cálculo del torque	20
2.1.1.2.	Diseño estructural de la Base Móvil	24
2.1.2.	Mecanismo de movimiento Webcam	25
2.2.	HARDWARE	27
2.2.1.	Control Remoto	28
2.2.2.	Arquitectura electrónica del Robot	30
2.2.2.1.	Cervical Shield	31
2.2.2.2.	Lumbar Shield	32
2.3.	SOFTWARE	33
2.3.1.	Software en Control Remoto	33
2.3.2.	Software en Smartphone	34
2.3.3.	Software en el Robot	35
2.3.4.	Diagrama Completo	39
3.	RESULTADOS DE IMPLEMENTACIÓN Y COSTOS DEL PROYECTO	41
3.1.	RESULTADOS DE IMPLEMENTACIÓN	41
3.1.1.	Resultados en Base Móvil	41
3.1.2.	Resultados mecanismo de movimiento webcam	42
3.1.3.	Resultados en Hardware	44
3.1.3.1	Método de Fabricación	44
3.1.3.1	Control Shield	45
3.1.3.2	Cervical Shield	45
3.1.3.2	Lumbar Shield	46
3.1.4.	Resultados en Software	47
3.1.4.1	App Smartphone	47
3.1.4.2	Python Server	48
3.1.4.3	Shields	49
3.1.5.	Funcionamiento del Prototipo	49
3.1.5.1	Movimiento del robot	50
3.1.5.1	Movimiento de la webcam	51

3.2. COSTOS DEL PROYECTO	52
3.2.1. Costos	52
3.2.2. Costos por mano de obra	53
3.2.3. Costo Total	54
CONCLUSIONES	55
BIBLIOGRARFÍA	56
GLOSARIO	57
ANEXOS	58

ÍNDICE DE FIGURAS

Figura 1-1. Lattepanda	7
Figura 1-2. Placa arduino mega	8
Figura 1-3. Placa arduino nano	8
Figura 1-4. Webcam logitech c270.	11
Figura 1-5. Motorreductor dc alto torque.	11
Figura 1-6. Servomotor ds3218.	12
Figura 1-10. Diagrama robot.	17
Figura 2-1. Fuerzas normales en plano inclinado.	20
Figura 2-2. Torque en plano inclinado.	21
Figura 2-3. Torque en plano inclinado considerando aceleración.	22
Figura 2-4. Diseño 2D base móvil	24
Figura 2-5. Tipos de movimiento de la base móvil	25
Figura 2-6. Diseño 2D mecanismo movimiento webcam en autocad	26
Figura 2-7. Diseño 3D mecanismo movimiento webcam en fusion 360.	27
Figura 2-8. Esquema conexionado control remoto.	29
Figura 2-9. Diseño pcb control shield.	29
Figura 2-10. Arquitectura de hardware electrónico.	30
Figura 2-11. Diseño pcb cervical shield.	31

Figura 2-12. Esquema conexionado de cervical shield.	31
Figura 2-13. Diseño pcb lumbar shield.	32
Figura 2-14. Esquema conexionado de lumbar shield.	32
Figura 2-15. Print screen yawcam.	36
Figura 2-16. Cadena de bytes de cervical shield.	37
Figura 2-17. Cadena de bytes de lumbar shield.	38
Figura 2-18. Diagrama completo.	39
Figura 3-1. Fotografía base móvil	42
Figura 3-2. Fotografía mecanismo de movimiento de webcam	43
Figura 3-3. Imagen comparativa de circuito y diseño control shield	45
Figura 3-4. Imagen comparativa de circuito y diseño cervical shield	46
Figura 3-5. Imagen comparativa de circuito y diseño lumbar shield	47
Figura 3-6. Interfaz gráfica aplicación smartphone.	48
Figura 3-7. Robot completo	50
Figura 3-8. Pruebas de movimiento webcam	51

ÍNDICE DE TABLAS

Tabla 3-1. Proveedores por componente y/o material.	53
Tabla 3-2. Costos por mano de obra del proyecto.	54
Tabla 3-3. Costo total del proyecto.	54

ÍNDICE DE ECUACIONES

Ecuación 2-1. Fórmulas fuerzas.	21
Ecuación 2-2. Fórmula torque.	21
Ecuación 2-3. Fórmula equilibrio de fuerzas.	22
Ecuación 2-4. Fórmula relación fuerzas y torque.	22
Ecuación 2-5. Fórmula torque en plano inclinado.	22
Ecuación 2-6. fórmula torque de motores.	23

Ecuación 2-7. fórmula torque de motores considerando perdidas.

23

ÍNDICE DE GRÁFICOS

Gráfico 2-1. Gráfico comparativo de filtrado de señal por Average.

34

SIGLAS Y SIMBOLOGÍAS

A. SIGLAS:

R.U.R	: Robots Universales Rossum (Robots Universales Rossum)
VR	: Virtual Reality (Realidad Virtual)
HDMI	: High-Definition Multimedia Interface (Interfaz Multimedia de Alta Definición)
MIPI-DSI	: MIPI Display Serial Interface (Interfaz serie de pantalla MIPI)
ICSP	: In-Circuit Serial Programming (Programación serie en circuito)
CPU	: Central Processing Unit (Unidad Central de Procesamiento)
URL	: Uniform Resource Locator (Localizador Uniforme de Recursos)
IP	: Internet Protocol (Protocolo de Internet)
HH	: Hora-Hombre
PLA	: Polylactic Acid (Ácido Poliláctico)
LAN	: Local Area Network (Red de área local)
UART	: Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal).
DOF	: Degrees Of Freedom (Grados de Libertad).
TTL	: Transistor-Transistor Logic (Lógica transistor a transistor).
PWM	: Pulse Width Modulation (Modulación por ancho de pulso).
USB	: Universal Serial Bus (Bus Serie Universal).
+VCC	: Voltaje en corriente continua.
GND	: Tierra (Ground).
SBC	: Single Board Computer (Computador de placa única)
UF	: Unidad de Fomento
CLP	: Chilean Peso (Peso Chileno)

B. SIMBOLOGÍAS:

g : Gramos.

m : Metros.

cm : Centímetro.

mm : Mili-metro.

ms : Mili-segundo.

Hz : Hertz, unidad de medida de frecuencia.

KHz : Kilo-Hertz.

MHz : Mega-Hertz.

GHz : Giga-Hertz.

GB : Gigabyte.

Kg : Kilogramo.

Kg/cm : Kilogramos por centímetro.

V : Volt, unidad de medida de Potencial eléctrico.

A : Amper, unidad de medida de Corriente eléctrica.

m/s : Metros por segundo

INTRODUCCIÓN

La robótica es una tecnología que día a día se abre paso en la vida cotidiana, teniendo desde hace años gamas de robots industriales que hacen la diferencia en la manufactura de productos, hoy en día hacen presencia en los hogares, ya sea en contexto de robots de limpieza como también robots de entretención.

Si bien, actualmente se considera que los robots reemplazarán a los humanos en todas las tareas que se realizan, existen variados tipos de robots y no todos implican el reemplazo por completo del ser humano.

Desde hace algunos años, en la década del 90, se acuñó el término cobot, que son robots colaborativos. Estos robots, a diferencia de los robots industriales, están diseñados para trabajar a la par físicamente con seres humanos, lo cual implica un cambio de paradigma en términos de diseño y seguridad.

Por otro lado se tiene robots tele operados, los cuales actualmente se usan para fines de exploración, investigaciones y también para telepresencia.

La pandemia actual, ha forzado a que variados trabajos se realicen en forma remota, principalmente por dos fines:

- Conservar y mantener la producción normal de productos y servicios.
- Proteger los recursos humanos de las empresas.

Frente a este desafío, se presenta una propuesta de robot que permita realizar en forma remota trabajos que de momento se deben realizar en forma presencial. El robot es controlado por un ser humano y éste no se utiliza como un reemplazo, sino que más bien como una herramienta complementaria que protege el capital humano de las empresas.

CAPÍTULO 1: ROBOTS, SEGURIDAD Y TRABAJO REMOTO

1. **ROBOTS, SEGURIDAD Y TRABAJO REMOTO**

El término robot proviene del vocablo checo Robota, que significa servidumbre, se comenzó a usar en una obra de teatro llamada R.U.R (Robots Universales Rossum), escrita por el autor checo Karel Čapek en 1920, desde ese uso en adelante, se ha utilizado esta palabra en forma transversal en el mundo para denominarlos.

Si bien desde la antigüedad existían ciertos tipos de autómatas que podían realizar algunas operaciones básicas, desde los avances de la electrónica digital en el siglo XXI, tanto lo que significa el concepto de robot, como también sus capacidades para realizar tareas han crecido en forma exponencial.

En la evolución de los robots, se han creado distintos tipos de ellos, si bien no existe acuerdo aún sobre todas las reglas que permitirían crear ramas taxonómicas como se da en el caso, de los organismos vivos, se diferencian por lo general en términos de:

- Morfología
- Aplicación
- Tecnología
- Fuentes de energía
- Procesamiento

En la sociedad actual, la primera incursión masiva de robots se dio en el área industrial, donde múltiples compañías fabrican brazos robóticos, que en términos de producción en serie pueden realizar tareas repetitivas en forma más rápida, exacta y menor riesgo que un ser humano. Este tipo de robots se utilizan mayormente en la industria de fabricación automotriz, pero independiente de eso, hoy están presentes en una importante cantidad de otras industrias.

Con respecto a modelos más particulares de robots, y que mayoritariamente son más relevantes para este proyecto, se tiene a los robots teleoperados, donde existen 3 tipos que son más comunes:

- **Robot de exploración:** Son Robots que sirven para explorar espacios recónditos o a veces difíciles de acceder físicamente por un humano normal o bien peligrosos para el ser humano, existen algunos robots de exploración que sirven en caso de catástrofes como derrumbes o terremotos, y otros que se usan más para aplicaciones de índole industrial, ya sea, por ejemplo, como la exploración de conductos bloqueados por material en minería.
- **Robot de telepresencia:** Robots móviles por lo general de dos ruedas móviles, con una de ellas libre, que sirven para que una persona haga presencia en un determinado sitio en forma remota, por lo general se utiliza en reuniones o en ceremonias donde la presencia simbólica es importante.
- **Robot de teleoperaciones:** Robots muy específicos del área Médica, que sirven para realizar operaciones de emergencia por un doctor que se encuentra a distancia, tiene un alto grado de exactitud y seguridad, pues cualquier desperfecto, podría terminar con la vida de un paciente.

1.1. COVID-19

En diciembre del año 2019 se produjo un brote epidémico de neumonía de causa desconocida en Wuhan, provincia de Hubei, China. Este brote, el cual fue denominado Covid-19 por la OMS el 11 de Febrero del 2020, en el lapso aproximado de un año se expandió a casi todos los países del planeta, generando diversas consecuencias negativas tanto en aspectos sociales como económicos.

Bajo los efectos de la pandemia, la mayoría de los trabajos que pertenecen al área de las TICS, pasaron a ser remotos debido a las facilidades que permite internet para este tipo de conexiones, sin embargo, varios otros trabajos se deben realizar aún en forma presencial.

Entre estos trabajos existen varios que implican que una persona deba explorar un lugar, y en estos casos los típicos robots de telepresencia no tienen la libertad de visión de exploración necesaria para una óptima exploración, pero por otro lado los robots exploradores normales no tienen las dimensiones necesarias para explorar el lugar como lo haría un humano, pues su diseño en esencia busca llegar a los espacios que normalmente para un ser humano resulta imposible.

1.2. PROPUESTA DE SOLUCIÓN

Un robot móvil teleoperado, que permita al usuario explorar edificaciones. Para facilitar y reducir la curva de aprendizaje de control del robot, el desplazamiento de éste, podrá ser controlado por un pequeño control con una palanca tipo Joystick que quepa en la palma de la mano, y para el movimiento del campo de visión del robot, se utilizarán lentes de realidad virtual que utiliza un Smartphone como pantalla. Esta última característica, hace bastante más intuitivo el control del robot, pues el usuario que lleve los lentes de realidad virtual solo deberá cambiar la dirección hacia donde observa con sus ojos, y en consecuencia el robot remoto generará los mismos cambios angulares en los ejes de su cámara.

En términos simples el control de mano se conectará vía Bluetooth al Smartphone en los lentes VR, entregándole las señales de dirección de desplazamiento, el Smartphone tomará estos datos y le sumará sus datos del acelerómetro y giroscopio para los ángulos de la cámara, esta información la enviará por internet al robot, el cual enviará su respuesta también por medio de internet de la imagen que capta su cámara, la cual se verá reflejada en el Smartphone que está en los lentes VR del usuario.

1.3. COMPONENTES DEL ROBOT

Para la correcta estructura y diseño del robot se utilizarán componentes electrónicos. Estos se describen a continuación detalladamente para tomar conocimiento adecuado de sus funciones específicas en el proyecto desarrollado.

1.3.1. LattePanda

LattePanda es un computador de placa única (SBC) diseñado por un equipo de ingenieros en Shanghái, China. Fue lanzado al mercado en el año 2015, a través de una campaña de crowdfunding en la plataforma Kickstarter. Es conceptualmente similar a la Raspberry Pi, pero a diferencia de ésta, tiene un costo mayor y funciona con procesadores Intel en lugar de ARM. Además, tiene la capacidad de ejecutar Windows 10 o Linux. En la figura 1-1, que se encuentra posterior a las especificaciones, se muestra una imagen de la Placa LattePanda.

En este proyecto, se usará la LattePanda V1 2GB/32GB, versión que cuenta con las siguientes especificaciones:

- Procesador: Intel Cherry Trail Z8350
- Frecuencia base: 1.44GHz (frecuencia de la ráfaga 1.92GHz)
- Sistema Operativo: Windows 10 Home Edition (Desactivado)
- RAM: 2GB DDR3L
- Capacidad de almacenamiento: 32 GB
- GPU: Intel HD Graphics, 12 EUs @ 200-500Mhz, memoria de un solo canal
- USB 3.0 x 1, USB 2.0 x 2
- Wi-Fi 802.11n 2.4G
- Bluetooth 4.0
- Salida de vídeo: HDMI y MIPI-DSI
- Conector de superposición de panel táctil incorporado
- Soporta Ethernet de 100Mbps

- Procesador Intel GPIO x 6
- Procesador ATmega GPIO x 20
- Conectores de interfaz "Gravity Series" x 6
- Alimentación: 5V / 2A
- Dimensiones del tablero: 88 x 70mm / 3.46 x 2.76 "
- Dimensiones del paquete: 110 x 94 x 30 mm / 4.33 x 3.70 x 1.18 "
- NET Peso: 55g



Fuente: <https://bit.ly/3IMqOp0>

Figura 1-1. LattePanda

1.3.2. Arduino Mega 2560

Arduino Mega 2560 es una placa basada en el microcontrolador ATmega2560. Tiene 54 pines de entrada/salida digital (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un cabezal ICSP y un botón de reinicio.

Esta tarjeta Arduino, estará conectada al Lumbar Shield. Se usa esta versión de Arduino debido a la cantidad de señales digitales que interactúan con los motores de desplazamiento, que son dos señales de control para el motor. Como esta versión tiene una elevada cantidad de entradas y salidas digitales, quedará en reserva un número considerable de estas, para agregar otros tipos de sensores complementarios en el futuro. En la figura 1-2 se puede apreciar la placa Arduino Mega.



Fuente: <https://bit.ly/3oDHb6K>.

Figura 1-2. Placa Arduino Mega

1.3.3. Arduino Nano

Arduino Nano es una placa compacta, completa y compatible con placas de pruebas basada en el microcontrolador ATmega328P, lanzada al mercado en el año 2008. Ofrece la misma conectividad y especificaciones de la placa Arduino Uno, en un tamaño más pequeño.

Un Arduino Nano estará ubicado en la placa Cervical Shield, y esta permitirá al CPU coordinar los movimientos de los servomotores que controlan el ángulo de visión de la webcam. Se utilizará también como microcontrolador para la lectura y transmisión de señales de control del joystick en el control remoto. A continuación, en la figura 1-3 se puede apreciar la placa Arduino Nano.



Fuente: <https://arduino.cl/arduino-nano/>

Figura 1-3. Placa Arduino Nano

1.3.4. Placas Shields.

Los Shields son PCBs que se conectan a las placas Arduino a través de sus pines de acoplamiento. La principal función de estas placas, es la de actuar como un módulo complementario que permite ampliar las capacidades de las tarjetas Arduino.

Existen Múltiples tipos de Shields para diferentes aplicaciones, como por ejemplo: Ethernet, bluetooth, relays, wifi, etc. En el caso de este proyecto, se diseñarán 3 tipos de Shields específicos.

Se justifica la creación y uso de estos Shields, puesto que, si todas las conexiones se hacen con cables Dupont o similares en este tipo de proyectos, que por lo general están expuestos a vibraciones o movimientos bruscos, generan entre las conexiones de cables y la tarjeta, un falso contacto eléctrico, lo cual perjudica el adecuado funcionamiento del circuito.

Por otra parte, este tipo de estructura modular con el uso de Shields, facilita la implementación del circuito, además de crear mejores condiciones para el mantenimiento del mismo, pues, si en determinado momento alguna de las tarjetas Arduino falla, simplemente se desacopla del Shield y se conecta una nueva. Por el contrario, si el Shield falla, se reemplaza simplemente por otro módulo en buen estado.

A continuación se describirán con mayor detalle los Shields diseñados para este robot y cuáles son sus características y funciones principales.

1.3.4.1 Cervical Shield

Esta tarjeta de circuito impreso estará habilitada para recibir y enviar datos a través del conector USB que tiene integrado, además de recibir energía a través del mismo. Posee también un borne adicional de entrada para la conexión de energía que alimentará a los servomotores, tiene habilitados tres pines por servo motor, para que éstos se puedan conectar en forma fácil y

directa en la placa. Estará encargada de transmitir en forma mecánica a través del cuello del robot, movimientos angulares hacia la webcam.

1.3.4.2 Lumbar Shield

Esta tarjeta de circuito impreso estará habilitada para recibir y enviar datos a través del conector USB que tiene integrado, además de recibir energía a través del mismo. Posee un tren de pines, que envía las señales de control al driver que energiza a los dos motores con ruedas en la base del robot.

1.3.5. Webcam Logitech C270

Una Webcam es una cámara de vídeo que transmite una imagen o un vídeo en tiempo real a una red, como Internet, o a través de ella. Las Webcam suelen ser pequeñas cámaras que se ubican en un escritorio, se fijan al monitor del usuario o se encuentran integradas en el hardware.

Este modelo en particular (Logitech C270), se escogió por el comprobado uso que se ha logrado obtener en otros proyectos de robótica y visión por computadora, además de tener una excelente relación entre calidad de imagen y costo., tiene una resolución de 1280 x 720 pixeles.

La Webcam se conectará vía USB a la LattePanda y tiene como función capturar la visión frontal del entorno del robot, para que estas imágenes se envíen a través de internet hacia el smartphone que tiene el usuario que controla el robot. A continuación, en la figura 1-4 se encuentra una imagen de la Webcam.



Fuente: <https://www.logitech.com/es-roam/products/webcams/c270-hd-webcam.html>

Figura 1-4. Webcam Logitech C270.

1.3.6. Motor 12 V Alto Torque

Este tipo de motor es usado por lo general para producir movilidad en diversos tipos de robots. Dentro de sus características destacables, se tiene, que está compuesto por un motor de corriente continua, que viene embebido con una caja reductora, la cual reduce la velocidad de giro, pero adicionalmente aumenta el torque del eje. Estos equipos poseen una amplia gama de torques y velocidades en revoluciones por minuto (rpm). En su parte trasera posee un encoder, el cual es un sensor que a través de dos trenes de pulsos, permite determinar el número de giros que ha realizado el eje. A continuación, en la figura 1-5 se encuentra una imagen que permite apreciar el Motor.



Fuente: <https://bit.ly/3u8uIOX>

Figura 1-5. Motorreductor DC Alto Torque.

1.3.7. Servomotor DS3218

Los Servomotores son actuadores con la capacidad de mantener o desplazarse hasta un ángulo determinado, este ángulo se le envía por medio de una señal de pulso modulada (PWM).

El robot utilizará dos servomotores, en la parte superior, lugar donde se encuentra la webcam, uno de ellos se utilizará para el movimiento horizontal y el otro para movimiento vertical, de esta forma los actuadores, según se les indique, cambiarán la dirección en que la cámara recibe la imagen.

Sus dimensiones constan de un largo de 40 mm, un ancho de 20 mm y un alto de 40.4 mm, posee tres terminales, dos de alimentación y uno para PWM. El torque de este modelo es de 20 Kg/cm. A continuación, en la figura 1-6 se encuentra una imagen del Servomotor.



Fuente: <https://bit.ly/3vnIUR6>

Figura 1-6. Servomotor DS3218.

1.3.8. Batería

Investigando la experiencia de múltiples desarrolladores de proyectos robóticos con placas SBC (LattePanda, Raspberry Pi, etc.) , estos han indicado problemas de estabilidad, que han tenido como causa, alimentar estas CPU en conjunto con motores desde la misma batería, estas complicaciones se han producido a pesar de tener reguladores de voltaje o conversores DC/DC. En varios casos el ruido EM generado por los motores u otros equipos de alto consumo, terminan afectando las funciones normales de una SBC. En base a los argumentos expuesto en los hilos de discusión, ya sea tanto por explicaciones teóricas, como también por experiencias prácticas, la arquitectura de conexión que ha logrado mayor estabilidad, ha sido el uso de una batería que alimente la SBC y por separado otra que energice los equipos de mayor consumo.

Es por estos antecedentes que el diseño del robot considerará dos baterías para la alimentación de las diferentes etapas.

- Batería Recargable 12V (Control):

Esta batería alimentará a la SBC (LattePanda) y a las tarjetas de desarrollo con microcontroladores (Arduino Mega, Nano), también entregará energía solo a los componentes con funciones lógicas de los drivers que controlan a los dos motores de la base.

- Batería Recargable 12V (Fuerza):

Esta batería alimentará los drivers de control de los dos motores que otorgan movilidad al robot en su base, y entregará energía a los servomotores encargados de proporcionar movimiento a la Webcam.

1.4. COMPONENTES DEL CONTROL

Para la correcta estructura y diseño del control se utilizarán variados componentes electrónicos descritos a continuación.

1.4.1. Módulo Bluetooth HC-06

Este módulo que se conecta a los pines de comunicación serial del Arduino (Rx/Tx), permite la recepción y envío de datos de forma inalámbrica por medio de tecnología Bluetooth.

Este componente se encuentra ubicado en el control remoto y envía los datos del joystick al smartphone para que éste los transmita vía internet al robot. A continuación, en la figura 1-7 se encuentra una imagen del Módulo Bluetooth.



Fuente: <https://bit.ly/3vbClux>

Figura 1-7. Módulo Bluetooth HC-06.

1.4.3. Módulo Joystick Análogo de 2 Ejes

Este módulo consta de un mecanismo Gimbal que tiene conectado a cada uno de sus dos ejes, potenciómetros, los cuales con su variación resistiva indican el cambio de ángulo del

Joystick. Posee también un botón central que se activa al presionar la palanca. Este componente se encontrará instalado en el control remoto del robot, de esta forma con los movimientos del dedo pulgar en la palanca se indicará al robot la dirección a la que debe desplazarse. A continuación, en la figura 1-8 se encuentra una imagen del Módulo Joystick.



Fuente: <https://bit.ly/3ucepuW>

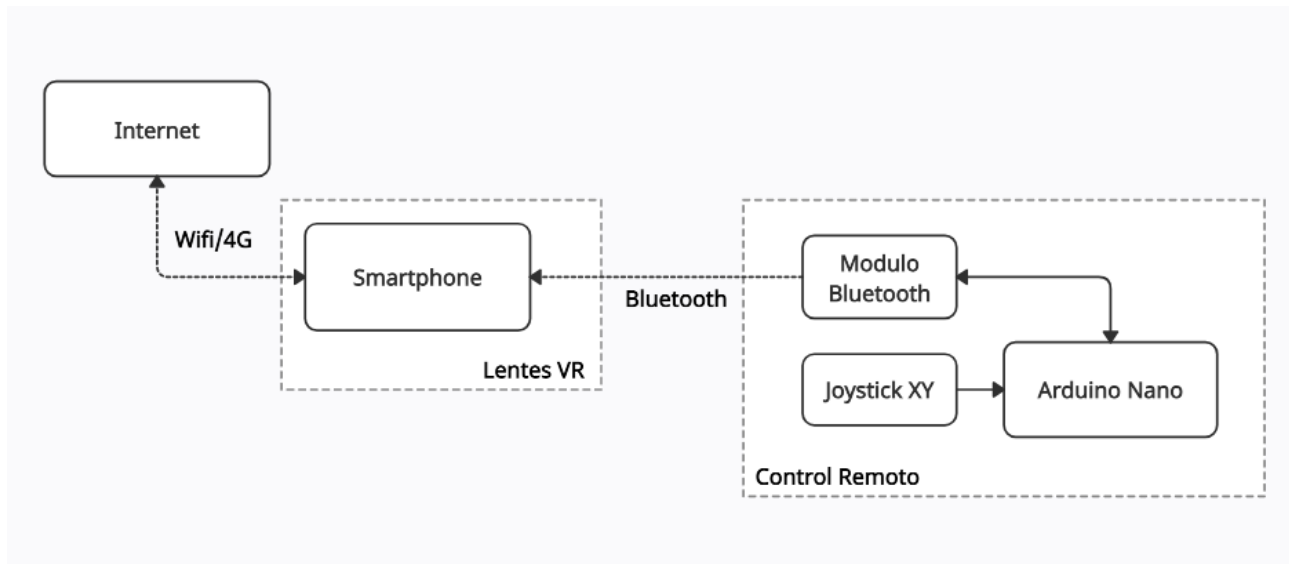
Figura 1-8. Módulo Joystick Análogo 2 Ejes.

1.5. DIAGRAMA DE BLOQUES

Para facilitar la comprensión de cómo van a interactuar los distintos componentes en el proyecto, a continuación, se expondrán los diagramas de bloques del control remoto y del robot respectivamente.

1.5.1. Diagrama Control Remoto

Este Diagrama muestra el hardware del control remoto y la conexión entre sus partes, también se muestra como éste se comunica con el smartphone para posteriormente enviar los datos por internet. A continuación en la figura 1-9 se puede apreciar el Diagrama de bloques.

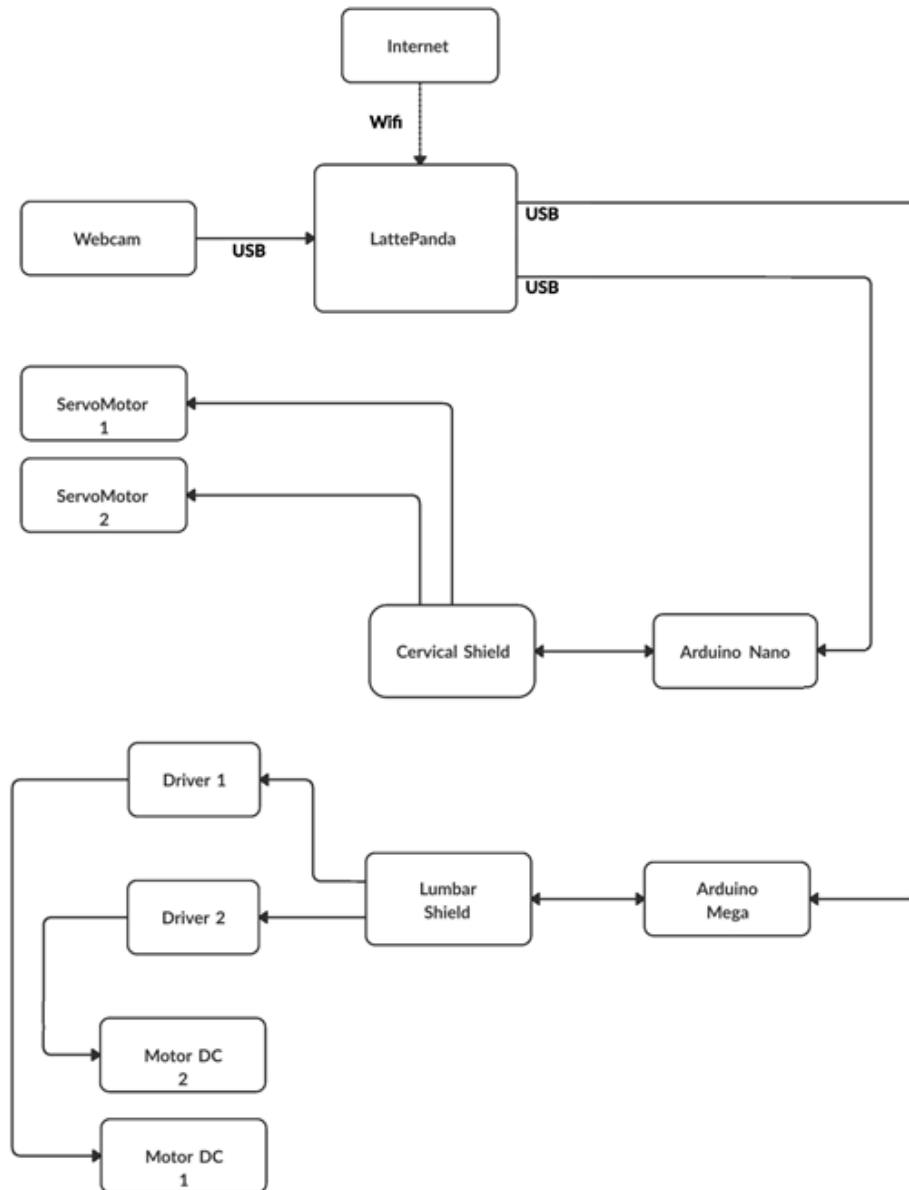


Fuente: Creación Propia

Figura 1-9. Diagrama en bloques Control Remoto.

1.5.2. Diagrama Robot

Este diagrama de bloques muestra la relación de conexiones de los distintos tipos de hardware anteriormente expuestos y que se encuentran dentro del robot, donde se muestran también los tipos de comunicación utilizados. A continuación en la figura 1-9 se puede apreciar el diagrama de bloques.



Fuente: Creación Propia

Figura 1-10. Diagrama Robot.

1.6. OBJETIVOS DE PROYECTO

A continuación se procede a exponer el objetivo general de este proyecto de título como también los objetivos específicos del mismo.

1.6.1. Objetivo General

Diseñar e implementar un robot de telepresencia que permita explorar edificaciones, brindándole seguridad al usuario.

1.6.2. Objetivos Específicos

- Manejo intuitivo del robot.
- Crear un sistema de movilidad con amplio rango de grados de libertad del robot.
- Diseñar y fabricar circuitos electrónicos que permitan controlar de forma modular las distintas partes del robot.
- Diseñar un sistema de control remoto vía internet que permita manejar el robot.

CAPÍTULO 2: DESARROLLO, CONEXIÓN Y PROGRAMACIÓN DEL PROYECTO

2. DESARROLLO, CONEXIÓN Y PROGRAMACIÓN DEL PROYECTO

En este capítulo podrá apreciar el proceso de desarrollo del prototipo de robot, que se propuso como solución al problema expuesto en el capítulo anterior. Se comenzará por explicar los diseños, estimaciones y cálculos requeridos para la construcción de la parte física y también en específico, mecánica del robot. Desde esta perspectiva, se abarcarán dos partes críticas del diseño del robot; la primera, es la base móvil, la parte baja del robot que tiene contacto con la superficie del suelo y que debe tener las características necesarias para permitir que el robot se pueda desplazar en una superficie bidimensional preferentemente plana., y la segunda, es el mecanismo que genera cambios angulares en la dirección a la que se dirige la webcam.

Una vez se ha concluido la explicación y exposición del proceso del desarrollo físico y mecánico del prototipo, se proseguirá con el diseño e implementación del hardware del robot, para continuar, con la exposición del funcionamiento de los códigos implementados en software para el correcto funcionamiento del robot.

2.1. DISEÑO FÍSICO Y MECÁNICO DEL ROBOT

En esta sección se describirán con mayor detalle los cálculos y etapas de desarrollo de la Base Móvil y el Mecanismo de movimiento de webcam.

2.1.1. Base Móvil

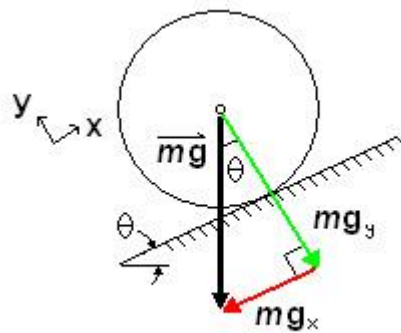
Una de las características del prototipo de robot propuesto es que éste sea móvil, y posea ruedas. El prototipo poseerá dos Ruedas, por lo tanto, se debe determinar qué actuador se utilizará para dar movimiento a éstas. En el caso de éste diseño, se determinó que se utilizarán motores de corriente continua (DC), y esta decisión se justifica, porque este tipo de motor se puede controlar de manera simple, lo que implicaría solamente encender, apagar, y determinar el sentido de giro. Además, si se requiere en el futuro, con la adición de sensores en el motor DC, se pueden lograr lazos de control más complejos. En el mercado se encuentra una amplia gama de este tipo de motores y hardware de control asociado a cada uno de ellos.

Para determinar el motor DC adecuado, se debe escoger el torque mínimo necesario, para movilizar al robot. El Torque se define como una magnitud vectorial, obtenida como producto vectorial del vector de posición del punto de aplicación de la fuerza por el vector fuerza., en términos simples, indica la capacidad de fuerza desde el eje del motor para desplazar una carga determinada. Si se escoge de forma arbitraria un motor, se corre el riesgo que el robot no tenga la capacidad suficiente como para desplazarse.

A continuación, se describirá un modelo físico, donde se consideran todas las fuerzas generales que se aplican en el desplazamiento del robot, a partir de este modelo y de la explicación de sus partes, se irán reordenando los valores en la ecuación, para determinar una fórmula que permita calcular el torque requerido.

2.1.1.1. Modelo de cálculo del torque

En una superficie inclinada (con un ángulo θ), sólo una componente de su peso (mg_x paralela a la superficie) hace que el robot se mueva hacia abajo. La otra componente, mg_y , se equilibra con la fuerza normal que la superficie ejerce sobre las ruedas.



Fuente: <https://bit.ly/3itg38Z>

Figura 2-2. Fuerzas normales en plano inclinado.

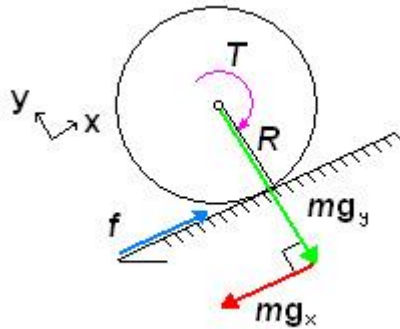
$$m g_x = m g * \sin(\theta)$$

$$m g_y = m g * \cos(\theta)$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-1. Fórmulas Fuerzas.

Para que el robot no se deslice por la pendiente, debe existir fricción entre la rueda y la superficie. Se puede tener un motor con un torque adecuado, sin embargo si no existe una buena fricción, el robot puede aun así deslizarse en la dirección contraria al avance. Es la fricción (f) la que sustenta al Torque.



Fuente: <https://bit.ly/3itg38Z>

Figura 2-2. Torque en plano inclinado.

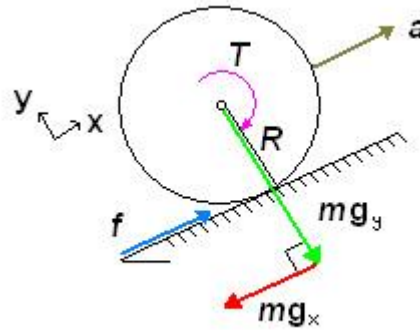
El Torque (T) necesario del motor, donde R es el radio de la rueda, es:

$$T = f * R$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-2. Fórmula Torque.

Para seleccionar el motor adecuado, se debe tener presente el "peor escenario", en el que el robot no sólo está en una pendiente, sino que acelera hacia arriba.



Fuente: <https://bit.ly/3itg38Z>

Figura 2-3. Torque en plano inclinado considerando aceleración.

Todas las fuerzas (F) están en los ejes x e y. Se equilibran las fuerzas en la dirección x:

$$\sum F_x = M * a = f - m * g_x$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-3. Fórmula equilibrio de Fuerzas.

Insertando la ecuación del par anterior, y la ecuación de mg_x , se obtiene:

$$M * a = T/R - M * g * \sin(\Theta)$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-4. Fórmula relación fuerzas y Torque.

Reordenando la ecuación para despejar T, se obtiene:

$$T = R * M * (a + g * \sin(\Theta))$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-5. Fórmula Torque en plano inclinado.

Este valor de Torque representa el Torque total necesario para acelerar el robot en una pendiente. Sin embargo, este valor debe dividirse por el número total (N) de ruedas motrices para obtener el Torque necesario para cada motor. Se debe tener presente que no se considera el número total de ruedas libres, debido a que no tienen ningún efecto sobre el Torque necesario para mover el objeto, ya que solo añaden peso.

$$T = \frac{(a + g * \sin(\theta)) * M * R}{N}$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-6. Fórmula Torque de motores.

El último punto a considerar es la eficiencia (e) en el motor, el engranaje y la rueda (deslizamiento).

$$T = (100/e) * \frac{(a + g * \sin(\theta)) * M * R}{N}$$

Fuente: <https://bit.ly/3itg38Z>

Ecuación 2-7. Fórmula torque de motores considerando perdidas.

Habiendo establecido la fórmula para calcular el torque de los motores del robot, se reemplazarán en la fórmula las siguientes variables.

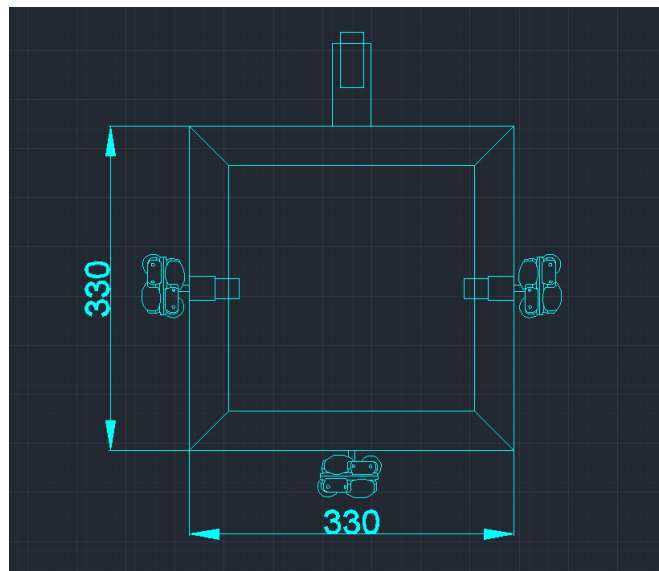
- Número de Ruedas motorizadas(N) = 2
- Ángulo de inclinación (Theta) = 10°
- Estimación Peso máximo del robot (M) = 5Kg
- Radio de la ruedas (R) = 0.03 m
- Eficiencia estimada (100/e) = 65%
- Aceleración esperada (a) = 0.2 m/s

Con estos valores en la fórmula y posteriormente haciendo una conversión a unidades en Kg/cm, que es el estándar comercial con el que se indica esta capacidad en los motores DC, el resultado es de 2,2 Kg/cm, por lo tanto, se implementarán motores que posean este torque o un aproximado superior.

2.1.1.2. Diseño estructural de la Base Móvil

Para que el robot pueda desplazarse en un plano bidimensional, la base está compuesta por dos motores ubicados en los costados del robot, conectados cada uno a una rueda y además, dos ruedas libres, una ubicada al frente y otra en la parte posterior. Estas últimas tienen por fin, sólo dar estabilidad a la base, para evitar así cualquier tipo de inclinación extrema, o que por otra parte, la zona baja del chasis del robot genere roce con el suelo.

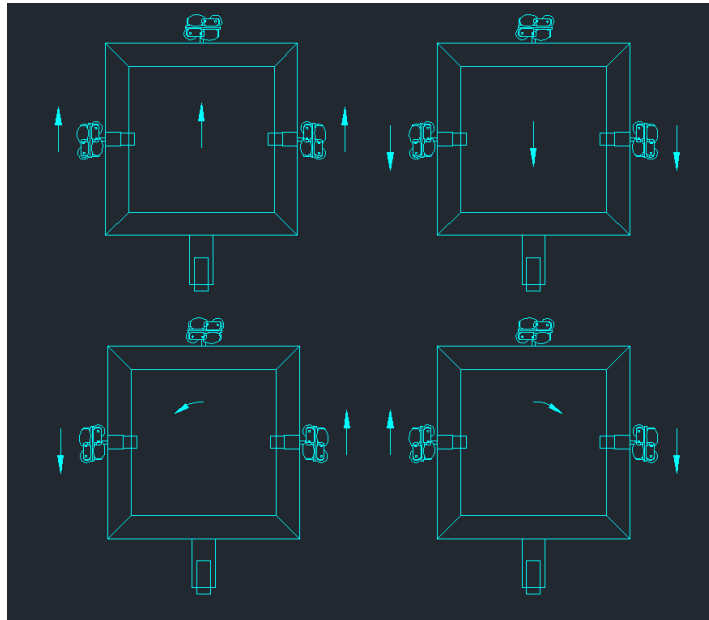
La estructura de la base es cuadrada, con dimensiones de 33 x 33 cm y fue diseñada para implementarse con aluminio tubular de 40 x 40 cm. En la figura 2-4 se puede apreciar el diseño en dos dimensiones de la Base.



Fuente: Creación propia

Figura 2-4. Diseño 2D Base Móvil

Este tipo de diseño de robot móvil se conoce como configuración diferencial. Ésta permite al robot realizar cuatro tipos de movimientos, avanzar, retroceder, girar a izquierda y girar a derecha. Estos movimientos se pueden apreciar con mayor facilidad en la figura 2-5.



Fuente: Creación propia

Figura 2-5. Tipos de movimiento de la Base Móvil

2.1.2. Mecanismo de movimiento Webcam

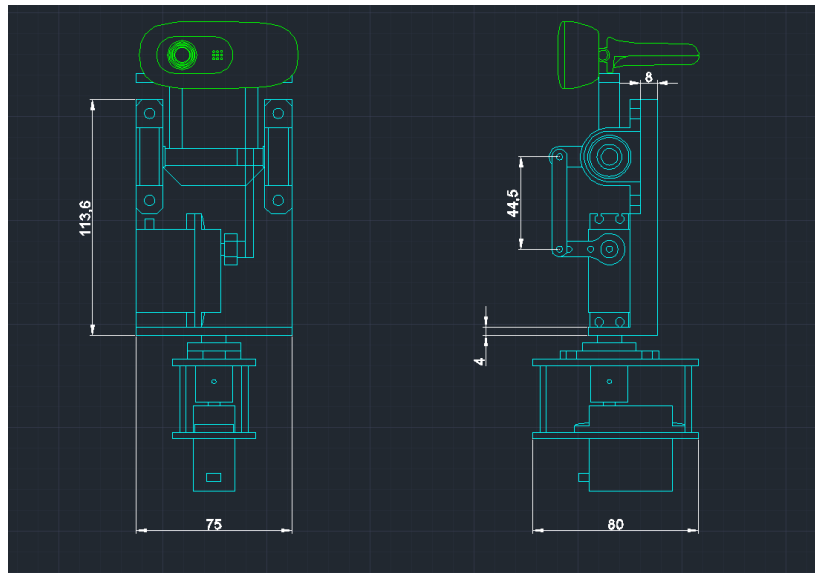
Para el mecanismo de movimiento del ángulo de la webcam, se deben diseñar y crear piezas mecánicas, que permitan que la webcam pueda realizar un movimiento de giro horizontal, y un movimiento de giro vertical. De esta forma, el punto de visión de la webcam, podrá explorar en forma visual y efectiva el entorno en donde se encuentre el robot.

Se debe elegir un tipo de actuador que permita los movimientos antes descritos para la webcam, las variables relevantes en este mecanismo, son los ángulos, por lo tanto lo importante es tener un tipo de actuador en el cual se pueda controlar la posición angular de

éste. Los actuadores que presentan este tipo de características y además poseen métodos de control de rápida implementación, son los servomotores. Este tipo de actuadores poseen una caja de engranaje en su interior que les permite tener mayor torque, y una tarjeta electrónica con un lazo de control cerrado para controlar la posición de su eje.

Los servomotores de más bajo costo a nivel comercial poseen un torque aproximado de 1.3Kg/cm (Servos SG90), los cuales en teoría podrían mover la webcam, sin embargo, este tipo de servomotor en particular, en ocasiones presenta inestabilidad en la posición y su placa interna además tiende a sobrecalentarse con cargas altas. Por lo anterior, en esta implementación se utilizarán los servomotores modelo DS3218, los cuales poseen un torque de 20Kg/cm, lo que permitirá mover la webcam y el mecanismo de forma efectiva.

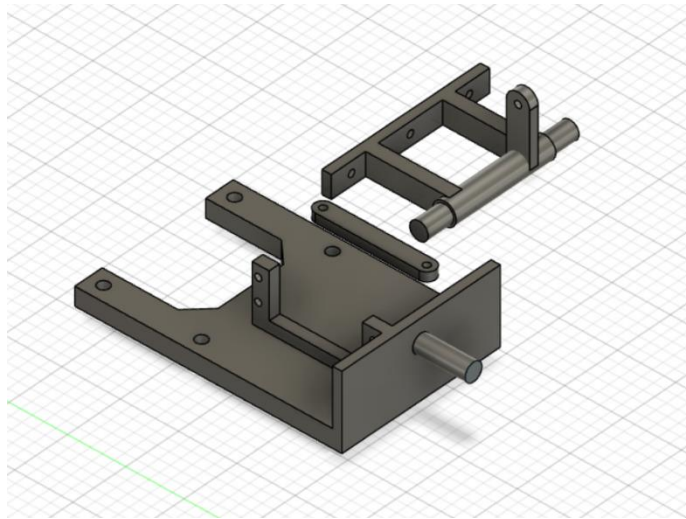
Se realizó en primera instancia un bosquejo en lápiz y papel, el cual posteriormente se traspasó y digitalizó en el software Autocad, donde se definieron las medidas finales. Para poder realizar este diseño, previamente se tuvo que hacer un modelamiento 2D de los elementos involucrados en el mecanismo, que en este caso fueron los servomotores y los rodamientos. Se puede apreciar el diseño en la figura 2-6.



Fuente: Creación Propia

Figura 2-6. Diseño 2D Mecanismo movimiento webcam en AutoCAD

Una vez concluido el diseño bidimensional del mecanismo, y habiendo comprobado que ,a lo menos en este nivel, no había conflicto entre el movimiento y posicionamiento de las piezas, posteriormente el diseño se modeló en tres dimensiones, en este caso, se hizo uso del software Fusion 360, como se puede apreciar en la figura 2-7. En la etapa de modelado 3D, también se realizaron pruebas que permitieran dilucidar si había conflictos de movimiento o sobreposición entre las piezas.



Fuente: Creación Propia

Figura 2-7. Diseño 3D Mecanismo movimiento webcam en Fusion 360.

Con el mecanismo modelado en tres dimensiones, éste puede ser exportado a un formato que permita imprimir las piezas en una impresora 3D.

2.2. HARDWARE

En esta sección se describirán con mayor detalle las etapas de diseño asociado a los circuitos del control remoto y al sistema de tarjetas Shield del robot.

2.2.1. Control Remoto

El control remoto, está principalmente compuesto por tres módulos. Los cuales se describen a continuación.

2.2.1.1. Joystick

En términos básicos está compuesto por dos resistencias variables o potenciómetros, los cuales a través de un mecanismo de Gimbal en que se mueve una palanca, generan distintos desplazamientos en ambos potenciómetros, permitiendo en cierta medida que los movimientos mecánicos de dirección en el plano X e Y se transformen en dos variaciones de voltaje, las cuales son leídas por un Arduino Nano.

2.2.1.2. Arduino Nano

El microcontrolador de Arduino, recibe los voltajes de las resistencias variables del módulo joystick. Arduino Nano posee entradas análogas, las cuales admiten la lectura de voltajes variables, gracias a que en su interior se encuentra un conversor análogo digital (ADC). Las señales ya digitalizadas son procesadas por el microcontrolador, para posteriormente ser enviadas vía comunicación serie UART, hacia el módulo de comunicación inalámbrica Bluetooth.

2.2.1.3. Módulo Bluetooth HC-06

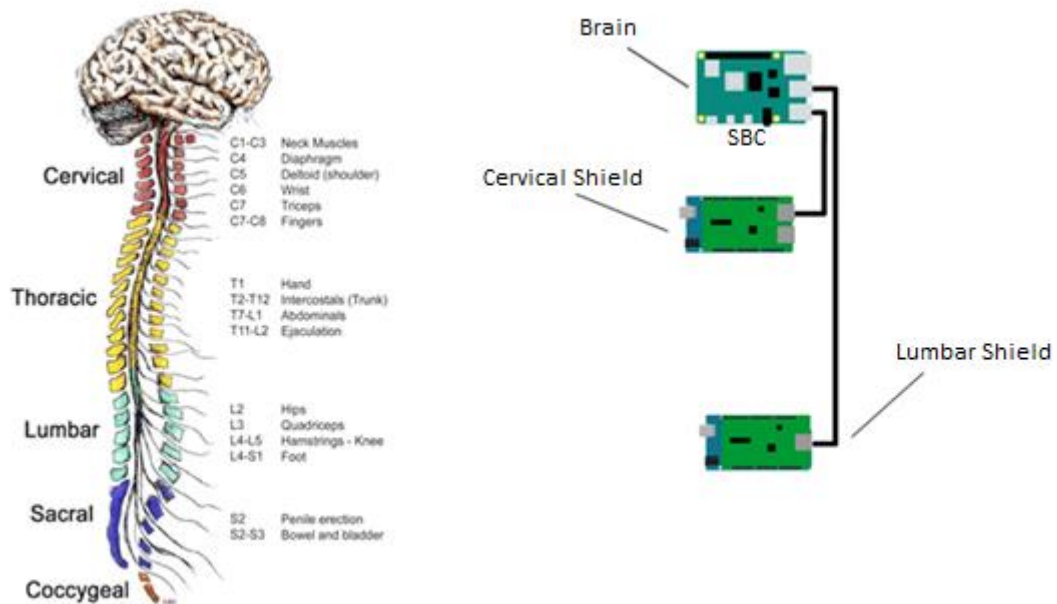
Este módulo recibe la señal procesadas de dirección del joystick desde el microcontrolador por su pin RX, para posteriormente enviar esta información vía Bluetooth, hacia un Smartphone que se encuentra en los lentes de realidad virtual que posee el usuario.

A continuación, en la figura 2-8 se presenta un esquema simple de conexión entre los módulos que componen el control remoto.

2.2.2. Arquitectura electrónica del Robot

Para la arquitectura electrónica del robot, se tomó como base la estructura del sistema nervioso de un ser humano. Desde esta perspectiva en términos análogos, se tiene una computadora de placa única LattePanda (SBC), dispositivo que actúa como un cerebro. Desde éste se distribuye una conexión vía USB, hacia el microcontrolador que se encuentra con el Cervical Shield, este hardware está encargado de controlar los servomotores que mueven a la webcam. En paralelo desde la LattePanda, hay otra conexión vía USB, que se conecta el microcontrolador con el Lumbar Shield, este último hardware está encargado de controlar los motores de corriente continua que están conectados a ruedas en la base móvil.

A continuación se adjunta una imagen esquemática y comparativa de la arquitectura anteriormente expuesta.

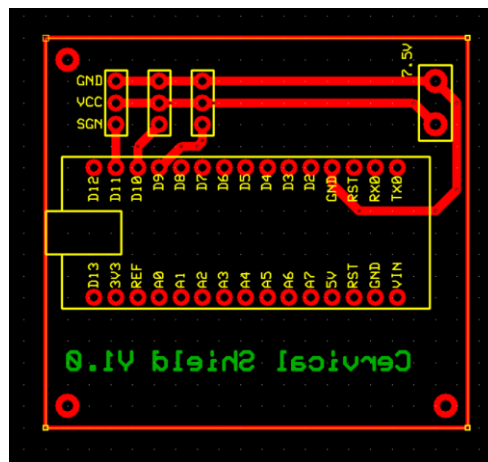


Fuente: Creación Propia

Figura 2-10. Arquitectura de Hardware electrónico.

2.2.2.1. Cervical Shield

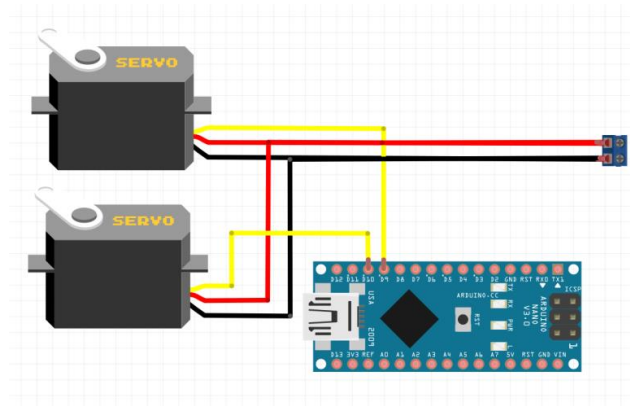
La PCB Cervical Shield, está conectada a un Arduino Nano. La función primordial de esta tarjeta es facilitar la conexión física entre la tarjeta de desarrollo, los servomotores que controlan el mecanismo de movimiento de la Webcam y una alimentación de voltaje externa que permite alimentar de forma adecuada en términos de corriente a los servomotores. Se realizó el diseño de esta tarjeta en la aplicación ExpressPCB, como se puede apreciar en la figura 2-11.



Fuente: Creación Propia

Figura 2-11. Diseño PCB Cervical Shield.

A continuación, en la figura 2-12 se expone también un esquema de conexionado de los elementos electrónicos que interactúan en la Cervical Shield.

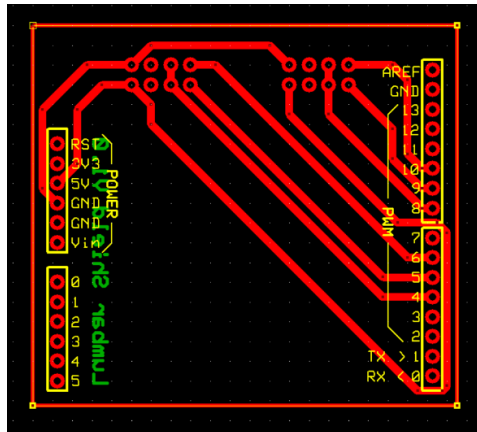


Fuente: Creación Propia

Figura 2-12. Esquema conexionado de Cervical Shield.

2.2.2.2. Lumbar Shield

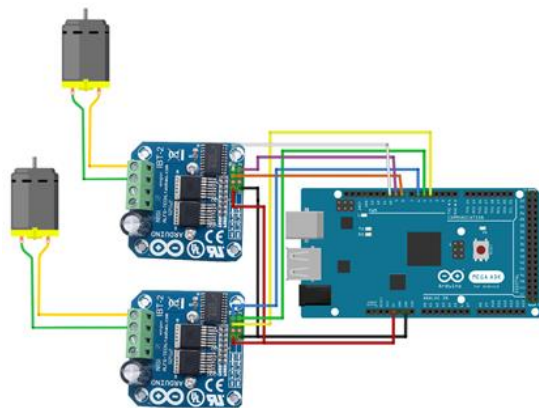
La PCB Lumbar Shield, está conectada a un Arduino Mega. La función primordial de esta tarjeta es facilitar la conexión física entre la tarjeta de desarrollo, los motores que controlan la base móvil y una alimentación de voltaje externa que permite alimentar de forma adecuada en términos de corriente a los motores. Se realizó el diseño de esta tarjeta en la aplicación ExpressPCB. El diseño de la placa se puede observar en la figura 2-13.



Fuente: Creación Propia

Figura 2-13. Diseño PCB Lumbar Shield.

A continuación, en la figura 2-14, se expone también un esquema de conexionado de los elementos electrónicos que interactúan en la Cervical Shield.



Fuente: Creación Propia

Figura 2-14. Esquema conexionado de Lumbar Shield.

2.3. SOFTWARE

En esta sección se describirán con mayor detalle las etapas de diseño asociado al código del control remoto y tarjetas Shield del robot.

2.3.1. Software en Control Remoto

Como se mencionó en la sección anterior, el control remoto tiene un Arduino Nano el cual hace lectura de las resistencias variables que se encuentran en el módulo Joystick, de ésta forma el usuario le indica al robot en qué dirección desea que se desplace. En las pruebas de funcionamiento, la lectura del voltaje del módulo joystick, presentaba ruido considerable.

Se realizaron una serie de pruebas prácticas, para determinar cuál era la mejor técnica que permitiera tratar la señal en términos matemáticos, y así reducir el ruido inherente en su lectura.

Para los términos específicos de este hardware y los requerimientos de este proyecto en particular, los mejores resultados se dieron utilizando una función de promedio, llamada en inglés "Average". Por tanto se utiliza esta última técnica para el tratamiento de la señal de los potenciómetros del Joystick. El Arduino toma alrededor de 20 muestras de la señal, y obtiene un promedio de estas. Si bien, otras técnicas generaban un suavizado considerable en la señal, estas ocasionaban un retardo pronunciado en variaciones de mayor magnitud, y fue debido a esta razón que se descartaron para su uso.

En el Gráfico 2-1, a continuación, se aprecia de mejor manera como se filtra la señal con la técnica de promedio.

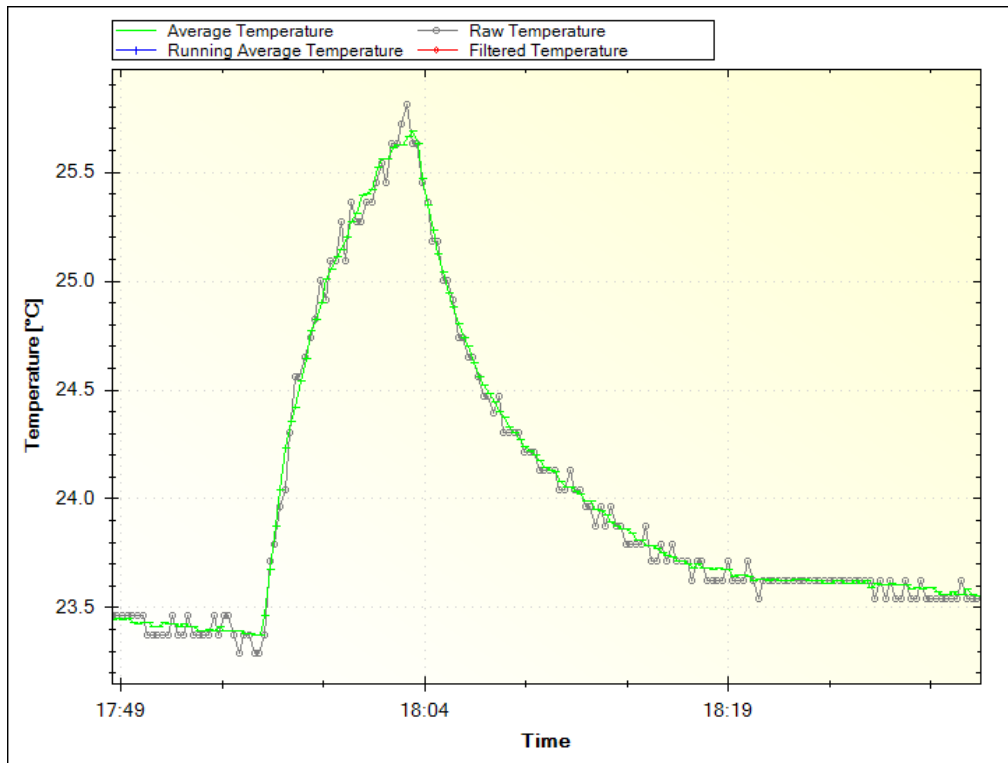


Gráfico 2-1. Gráfico comparativo de filtrado de señal por Average.

El código presente en el Arduino Nano del control remoto, luego de hacer un promedio de la señal, escala los valores desde un rango de 0- 1024 a 0-100. Esto, con el fin de reducir aún más las variaciones de la señal. Posteriormente, los valores ya tratados de ambos potenciómetros del Joystick se envían en forma de String, a través de los pines serie del Arduino Nano, los cuales están conectados al módulo Bluetooth, de esta forma se transmite la información en forma inalámbrica hacia el Smartphone que se encuentra en los lentes de realidad virtual del usuario.

2.3.2. Software en Smartphone

En el caso del Smartphone, se desarrolló una aplicación a través de “App Inventor”., Este es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android.

Esta aplicación, hace lectura de dos sensores específicos dentro del teléfono, uno de ellos es el sensor de orientación "Azimuth", este sensor indica el ángulo horizontal medido en el sentido de las agujas del reloj a partir de una línea base norte. Esto permite estimar variaciones angulares en el plano horizontal que realiza la cabeza del usuario que posee los lentes, por otro lado la aplicación también hace lectura del acelerómetro "Z", el cual mide el ángulo de inclinación hacia el suelo, esto permite estimar variaciones angulares en el plano vertical que realiza la cabeza del usuario.

La aplicación genera en forma automática una conexión Bluetooth, con el control remoto. A través de esta conexión, recibe los valores del Joystick, posteriormente hace un tratamiento de las señales del joystick donde hace un cambio en los caracteres de control y separación, para luego unir los valores de lectura de los sensores internos junto con los valores que recibió de la palanca del joystick. La aplicación construye una dirección URL, que está compuesta por la IP del robot a controlar, y por parámetros que llevan los valores de los sensores y mediciones del joystick.

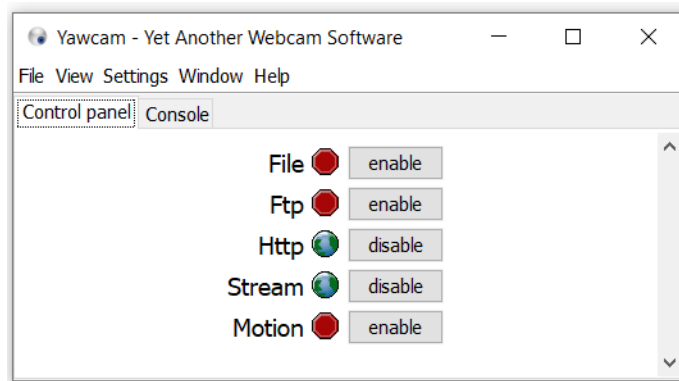
En lo que respecta a la visualización de lo que el robot percibe a través de su cámara por parte del usuario, la aplicación presente en el Smartphone, genera una conexión a través de una URL, que posee la IP del robot y un puerto específico (8081), con esta conexión, recibe la imagen de la Webcam que es suministrada por el software presente en el computador del robot que se llama YawCam.

2.3.3. Software en el Robot

El robot contiene software principalmente en tres sectores, gran parte se encuentra concentrado en el computador de placa simple LattePanda, y lo restante en los microcontroladores que se encuentran en los Shield Cervical y Lumbar.

2.3.3.1. Software en LattePanda

LattePanda es un mini computador que ejecuta en su interior el sistema operativo Windows 10 Home Edition. En lo que respecta al uso de software para las aplicaciones referentes al funcionamiento del robot, se tiene por un lado el software YawCam, el cual hace lectura de la webcam que está conectada vía USB al computador, y publica la imagen con una IP y un puerto específico para poder acceder. La interfaz gráfica de Yawcam se puede apreciar en la figura2-16. Éste software entrega también la posibilidad de proteger la vista de la cámara con un usuario y contraseña, el cual puede ser configurado a elección.



Fuente: Creación Propia

Figura 2-15. Print Screen Yawcam.

Por otro lado en este equipo también se tiene instalado Python. Se construyó una aplicación en este lenguaje que permite levantar un servidor HTTP. Este servidor recibe los datos que son enviados por el Smartphone, que incluyen los datos de los sensores de orientación e inclinación más las mediciones del Joystick del control remoto. Cuando la aplicación recibe estos datos, posee una lógica en su interior que separa los datos a través de un arreglo. Luego toma los valores que pertenecen a las mediciones del joystick, y las pasa por un comparador, esto con el fin de transformar esas mediciones finalmente en un número entero en un rango variable entre 0 y 4. Este número indicará al robot qué tipo de movimiento debe realizar.

Una vez se ha transformado la información del joystick en un número entero, la aplicación genera una cadena de datos, con los valores de los sensores internos del teléfono, y ésta es enviada en forma específica al Cervical Shield. El número entero resultado de la operación de comparaciones de las señales del joystick es adicionado a una cadena de caracteres, la cual es enviada de forma específica al Lumbar Shield.

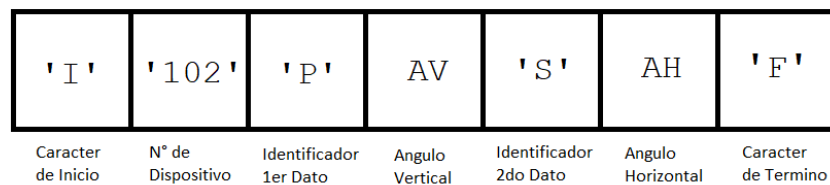
La estructura de las cadenas de datos que se envían a los Shields, sigue un protocolo de comunicación simple, para asegurar que la trama completa y los datos lleguen a las tarjetas de desarrollo en forma correcta.

2.3.3.1.1. Cadena de caracteres Cervical Shield

En el caso del microcontrolador con el Cervical Shield, se requiere que lleguen a este dos variables, el de orientación (ángulo de variación horizontal) y el de inclinación (ángulo de variación vertical).

Para facilitar la comprensión de la estructura de la cadena de datos, al ángulo de variación horizontal se le llamará "AH", y al ángulo de variación vertical se le llamará "AV".

En la figura 2-17, se expresa de manera gráfica la distribución de los bytes en la cadena.



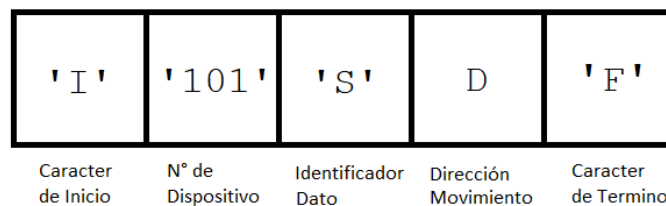
Fuente: Creación Propia

Figura 2-16. Cadena de Bytes de Cervical Shield.

2.3.3.1.2. Cadena de caracteres Lumbar Shield

En el caso del Arduino con el Lumbar Shield, se requiere que llegue a este solo una variable, la cual es el número que indica la dirección de movimiento a la base móvil.

Para facilitar la comprensión de la estructura de la cadena de datos, el número que indica la dirección de movimiento se le llamara "D". En la figura 2-18, se expresa de manera gráfica la distribución de los bytes en la cadena.



Fuente: Creación Propia

Figura 2-17. Cadena de Bytes de Lumbar Shield.

2.3.3.2. Código en Cervical Shield

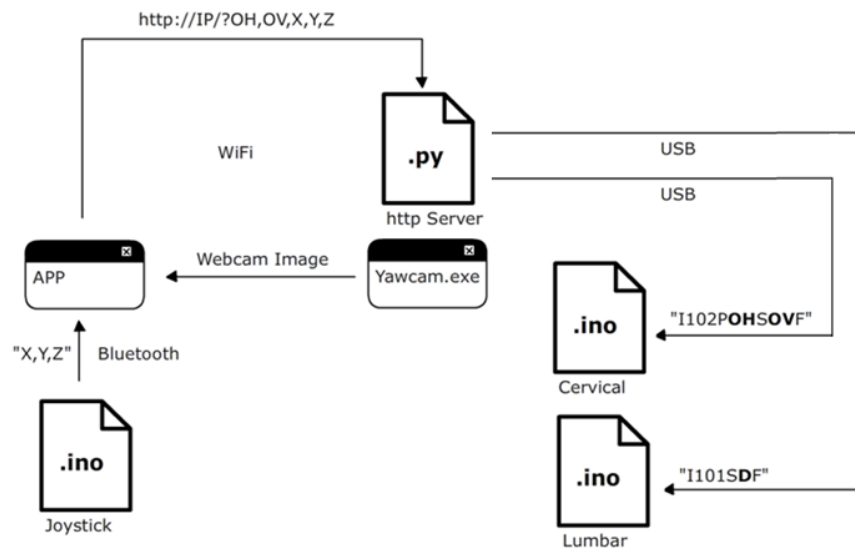
El código presente en esta tarjeta se encarga en una primera parte de verificar que la trama de bytes de datos que llegan desde la SBC LattePanda cumpla con la estructura esperada, esto es, comprobar la llegada del carácter de inicio, que el número de dispositivo coincida con su número de dispositivo designado, que los caracteres identificadores de datos sean los correctos, y por último que se encuentre el carácter de termino. Una vez comprobada la integridad de la trama, los valores de los ángulos tanto horizontal como vertical de la trama, son almacenados, para pasar por un escalamiento que ajusta la magnitud y sentido de giro. Luego estos valores angulares son enviados hacia el servomotor respectivo encargado de cada movimiento.

2.3.3.3. Código en Microcontrolador Lumbar Shield

El código presente en esta tarjeta se encarga en una primera etapa de verificar que la trama de bytes de datos que llegan desde la SBC LattePanda cumpla con la estructura esperada, esto es, comprobar la llegada del carácter de inicio, que el número de dispositivo coincida con su número de dispositivo designado, que el carácter identificador de datos sea el correcto, y por último que se encuentre el carácter de termino. Una vez comprobada la integridad de la trama, el valor de la dirección de movimiento de la trama es almacenado, y se pasa a un switch Case, donde se determinan los motores que deben activarse y su sentido de giro respectivo, para que el robot se desplace.

2.3.4. Diagrama Completo

En base a las descripciones previas, a continuación en la figura 2-19, se expone en un diagrama completo, la interrelación entre los distintos software y códigos. Donde además, se puede identificar tanto el tipo de comunicación como las tramas de datos utilizadas.



Fuente: Creación Propia

Figura 2-18. Diagrama Completo.

CAPÍTULO 3: RESULTADOS DE IMPLEMENTACIÓN Y COSTOS DEL PROYECTO

3. RESULTADOS DE IMPLEMENTACIÓN Y COSTOS DEL PROYECTO

En la exposición temática previa a este capítulo, se definieron los cálculos, estructuras y circuitos electrónicos necesarios para desarrollar la creación del prototipo.

En este capítulo se expondrán los resultados del desarrollo y construcción del prototipo de robot antes mencionado. Se expondrán registros visuales de la implementación de los diseños y se describirá en forma específica si las partes claves y también el prototipo completo, operan según lo esperado.

En la segunda parte de este capítulo, se describirán los costos asociados a la realización de este prototipo, teniendo en consideración los valores asociados a los materiales necesarios para su construcción y las horas hombre (HH) dedicadas en su diseño e implementación.

3.1. RESULTADOS DE IMPLEMENTACIÓN

Con el prototipo completo, la evaluación de resultados debe estar centrada principalmente en la capacidad de movilidad y la capacidad de explorar visualmente el espacio, a continuación se expondrán con mayor detalle los resultados prácticos de las etapas involucradas de forma crítica en estas funciones.

3.1.1. Resultados en Base Móvil

Esta parte del robot es la que le permite al robot desplazarse en distintas direcciones, se construyó en forma manual y con el uso de algunas herramientas tales como sierra , taladro y remachadora, entre otras.

Primero se construyó el marco metálico de aluminio. Posteriormente se adicionaron en sus cuatro caras inferiores, ángulos de aluminio, a los cuales se les realizaron las perforaciones necesarias para ensamblar los motores de alto torque.

Se agregaron además en la sección central inferior otras dos tiras de ángulos de aluminio, para que la base pudiese sostener las baterías que energizan al robot, y una platina de aluminio en la parte central superior, para adosar la columna de aluminio rectangular que sostendrá al mecanismo de movimiento de la webcam. Finalmente, se atornillaron los motores en las perforaciones de los ángulos de aluminio en la parte inferior de la estructura, quedando en una posición geométrica correcta y con conexión resistente.

La Base Móvil en términos de implementación física y mecánica tuvo un resultado aceptable. A continuación se aprecia una fotografía en la figura 3-1.



Fuente: Fotografía tomada por el alumno.

Figura 3-1. Fotografía Base Móvil

3.1.2. Resultados mecanismo de movimiento webcam

Este mecanismo que permite los movimientos específicos de la webcam, en base al movimiento de dos servomotores, pasó por las etapas de desarrollo previas, que implican el diseño 2D y posteriormente el 3D, lo cual se expuso con mayor detalle en el capítulo 2.

El diseño 3D que fue realizado en Fusion 360, se exportó a formato .STL para luego importarlo hacia el software Creality Slicer, donde se configuraron múltiples parámetros que permitieran la impresión de la pieza en tres dimensiones, entre los parámetros destacables, se encuentran, grosor de filamento, velocidad de impresión y porcentaje de relleno de las piezas.

La impresión de las piezas se realizó en una máquina Creality CR-10, donde el material para la impresión fue un filamento de plástico PLA de color blanco con un grosor estándar de 1.75mm. Es menester mencionar también que a la impresora tardó alrededor de aproximadamente 5 Horas, imprimir todas las piezas.

Una vez impresas, se realizó una limpieza de las piezas, lo que implicó retirar algunos residuos de filamento. A las piezas que tenían conexiones críticas dentro del mecanismo, se les realizó un leve desgaste para ajustar su ensamblaje con las otras piezas. Posterior al ensamblado mecánico de las piezas plásticas de PLA, se instalaron los servomotores y la webcam. Finalmente, se ajustó la base del mecanismo al pilar de aluminio rectangular que lo une con la base móvil.

El mecanismo de movimiento de webcam en términos de implementación física y mecánica tuvo un resultado aceptable. A continuación se aprecia una fotografía en la figura 3-2.



Fuente: Fotografía tomada por el alumno.

Figura 3-2. Fotografía mecanismo de movimiento de Webcam

3.1.3. Resultados en Hardware

En esta sección, se expondrán sobre los resultados del desarrollo de las tarjetas de circuito, en este caso llamadas Shields, se explicará a través de qué técnica fueron fabricadas, para finalmente mostrar fotografías o registros gráficos del resultado final.

3.1.3.1 Método de Fabricación

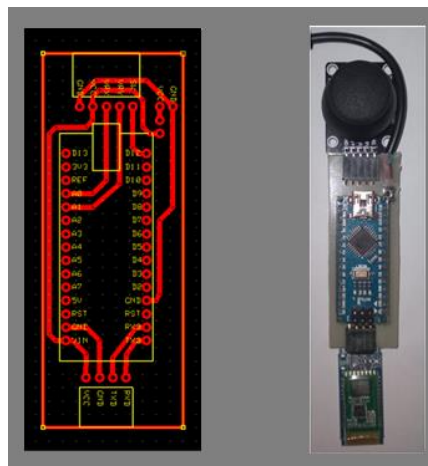
Las Placas Shields fueron fabricadas usando tarjetas PCB vírgenes de cobre con fibra de vidrio. El diseño de los circuitos se realizó en ExpressPCB, posteriormente los diseños se imprimieron en hojas de transferencia de tóner, con una impresora láser. Cada hoja con los circuitos ya impresos fue planchada con calor sobre su placa PCB virgen respectiva. Una vez que las placas ya tenían el circuito en tóner completamente transferido a la parte de cobre de las placas, estas se sumergieron por un periodo de 20-30 min en Cloruro Ferrico. Al sumergir la placa en este líquido, el Cloruro Ferrico produce una reacción química con el cobre de la placa, generando por consecuencia que toda la parte de cobre que no se encuentre cubierta por tóner, sea eliminada de la placa, quedando solo las vías del circuito.

Con el proceso químico ya completo, las placas fueron, lavadas, y secadas, para posteriormente pasar por una breve limpieza con acetona, la cual permitió retirar el tóner que se encontraba adherido aun a la parte de cobre. Una vez limpias y secas las placas otra vez, se realizó la perforación de los agujeros necesarios para la implementación de los componentes electrónicos. Los componentes en su gran mayoría fueron en primera instancia adheridos a la placa con un pegamento de tipo cinoacrilato, esto con el fin de asegurar que los componentes quedaran con la totalidad de su superficie en contacto con la placa. Con los componentes electrónicos ya ubicados, se procedió a soldar cada uno de ellos con alambre de estaño y cautín.

3.1.3.1 Control Shield

Tarjeta de circuito construida para que interactúen un joystick, Arduino Nano y el módulo Bluetooth. Con la tarjeta ya terminada se realizaron pruebas de conductividad, y posteriormente, energizando la tarjeta se realizaron pruebas con códigos de testeo, los cuales fueron exitosos.

Control Shield en términos de implementación física y electrónica tuvo un resultado aceptable. A continuación en la figura 3-3 se adjunta una Imagen comparativa del resultado y el diseño.



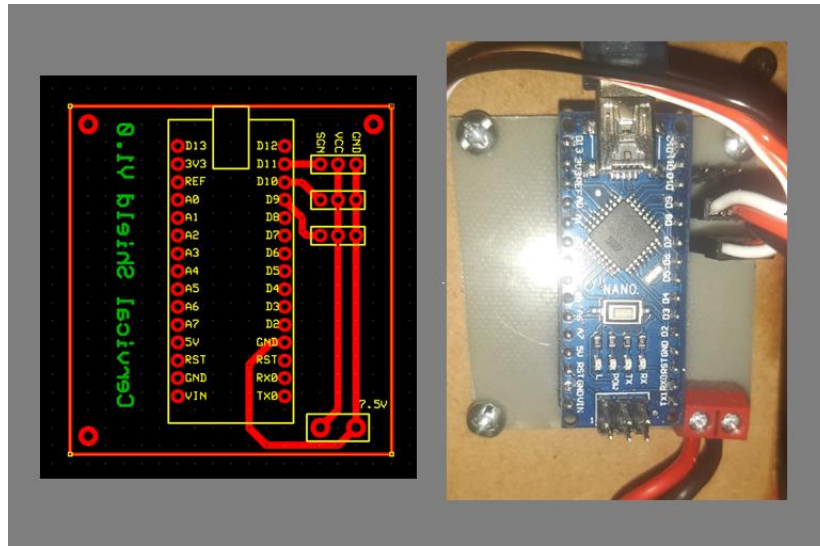
Fuente: Composición y fotografía creada por el alumno.

Figura 3-3. Imagen comparativa de circuito y diseño Control Shield

3.1.3.2 Cervical Shield

Tarjeta de circuito construida para que interactúen un Arduino Nano, 2 servomotores y una fuente de energía externa para alimentación. Con la tarjeta ya terminada se realizaron pruebas de conductividad, y posteriormente, energizando la tarjeta se realizaron pruebas con códigos de testeo, los cuales fueron exitosos.

Cervical Shield en términos de implementación física y electrónica tuvo un resultado aceptable. A continuación en la figura 3-4 se adjunta una Imagen comparativa del resultado y el diseño.



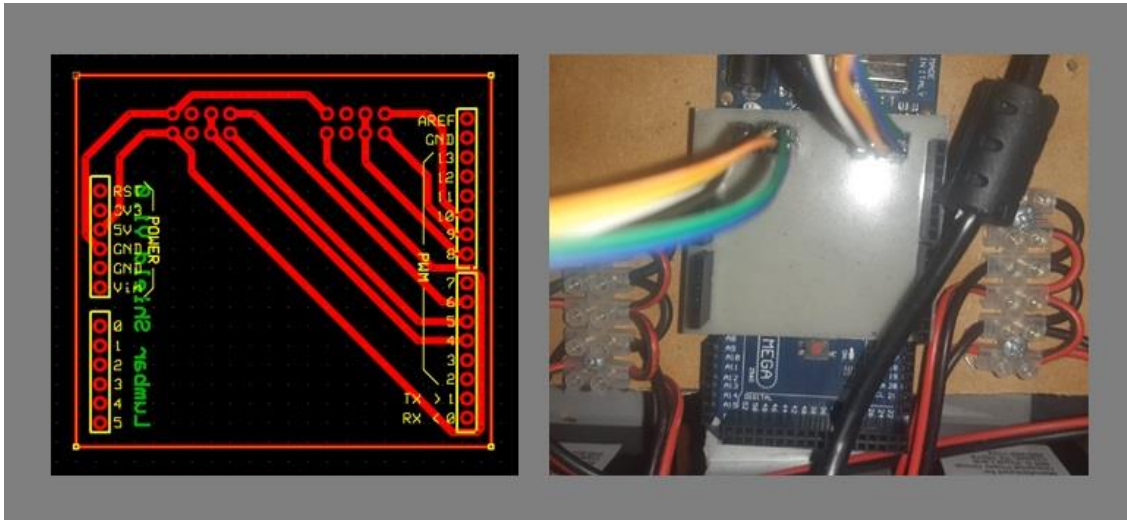
Fuente: Composición y fotografía creada por el alumno.

Figura 3-4. Imagen comparativa de circuito y diseño Cervical Shield

3.1.3.2 Lumbar Shield

Tarjeta de circuito construida para que interactúen un Arduino Mega y 2 Puentes H que controlan y alimentan a los motores de alto torque de la base móvil. Con la tarjeta ya terminada se realizaron pruebas de conductividad, y posteriormente, energizando la tarjeta se realizaron pruebas con códigos de testeo, los cuales fueron exitosos.

Lumbar Shield en términos de implementación física y electrónica tuvo un resultado aceptable. A continuación en la figura 3-5 se adjunta una Imagen comparativa del resultado y el diseño.



Fuente: Composición y fotografía creada por el alumno.

Figura 3-5. Imagen comparativa de circuito y diseño Lumbar Shield

3.1.4. Resultados en Software

En esta sección se describirá en forma breve los resultados de la creación del software y los códigos necesarios para el correcto funcionamiento del prototipo.

3.1.4.1 App Smartphone

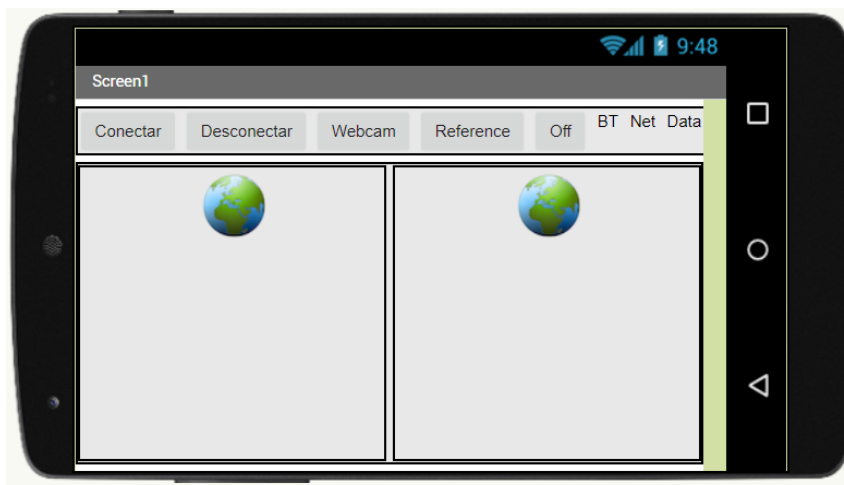
El software se puede desarrollar en forma efectiva a través de la plataforma de App Inventor, además del código generado para su funcionamiento, se creó una interfaz gráfica de usuario, que permitiera manejar con mayor facilidad al robot.

La aplicación en su interfaz gráfica cuenta con dos zonas relevantes, la primera es la zona superior, en ella se encuentran los botones:

- Conectar: Permite la conexión vía LAN o Internet, y comienza el envío de datos periódicos al servidor que se encuentra en el robot.
- Desconectar: Interrumpe la conexión y envío periódico de datos.
- Webcam: Permite la conexión a la webcam del robot.

- Reference: Como el sensor "Azimuth", entrega sus valores angulares en función del Norte, el botón Reference se debe presionar para que la aplicación sepa que en ese ángulo específico con respecto al Norte, el usuario está mirando hacia el frente con los lentes VR.
- Off: Envía una cadena de datos específico que hace que el robot inicie el proceso de apagado.

La segunda zona es la sección donde se ve la imagen de la webcam, la cual consiste en dos recuadros para entregar la imagen a cada uno de los ojos. En la figura 3-6 se puede observar cómo está compuesta la interfaz gráfica de la aplicación.



Fuente: Print Screen creada por el alumno.

Figura 3-6. Interfaz Gráfica Aplicación Smartphone.

La aplicación logró conexión exitosa con el control remoto, Python server y con la webcam.

3.1.4.2 Python Server

El servidor HTTP básico fue desarrollado en Python, se escribió y testeó con Pycharm. A continuación se listarán las librerías usadas en la versión final, junto a una breve descripción que justifica su uso.

- **HTTPServer:** Se usa para la creación del servidor básico HTTP.

- **OS:** Usada para realizar operaciones dependientes del Sistema Operativo como crear una carpeta, listar contenidos de una carpeta, etc.
- **Serial:** Utilizada para hacer uso de los puertos COM. En este caso específico, se usa para la conexión con los Arduino en el Cervical y Lumbar Shield.
- **Time:** Se usa para funciones de tiempo, en el código se utiliza para el control de retardos en ciertos procesos que lo requieren.
- **Playsound:** Librería usada para la reproducción de archivos de audio, se utiliza al principio del programa para reproducir sonidos específicos si la conexión a internet y al hardware es efectiva.

Las pruebas de conexión de esta aplicación, tanto con el Smartphone como para los Shields desarrollados en el proyecto fue exitosa, además, con extensas pruebas de tiempo demostró ser estable.

3.1.4.3 Shields

Los códigos desarrollados para los Arduinos que se encuentran conectados al Control Shield, Cervical Shield y Lumbar Shield, lograron de forma efectiva su operación, por lo que se pudo determinar después de variados test de operación y conexión, que el funcionamiento de los códigos en los microcontroladores funcionan en forma adecuada.

3.1.5. Funcionamiento del Prototipo

En esta sección se describirán los resultados alcanzados por el prototipo, cabe destacar que entre las posibilidades de funcionamiento que tiene el robot existen dos funciones críticas que se deben comprobar, la primera es la capacidad del robot de poder desplazarse y que este desplazamiento sea controlado por el control remoto que se conecta al Smartphone. En segundo lugar debe comprobarse que la webcam en la parte superior del robot es capaz de

moverse y seguir los movimientos del Smartphone que se encuentra en los lentes VR que lleva puesto el usuario.

3.1.5.1 Movimiento del robot

Para las pruebas de movimiento del robot, se testaron primero todas las conexiones, una vez validada la conexión, se procedió a probar que las direcciones ejecutadas en el joystick del control remoto se vieran reflejadas en los movimientos del robot. Se puede observar el robot completo en la figura 3-7.

Tanto en las pruebas de movimiento de avance y retroceso como también las de giro, el prototipo fue capaz de realizar los movimientos requeridos.



Fuente: Fotografías tomadas por el alumno.

Figura 3-7. Robot Completo

Dado los antecedentes expuestos, se puede aseverar que las pruebas son exitosas y el robot es capaz de mover la webcam siguiendo los movimientos del usuario.

3.1.5.1 Movimiento de la webcam

Para las pruebas de movimiento de la webcam, se testearon primero todas las conexiones, una vez validada la conexión, un usuario se puso los lentes de VR con el Smartphone en su interior, y procedió a realizar un movimiento de cabeza generando una variación en el plano horizontal, posteriormente se probó con movimientos de variación en el plano vertical, para terminar con un movimiento circular de la vista y por lo tanto de la cabeza. En la figura 3-8 se pueden apreciar fotografías de la prueba de movimiento.

Tanto en las pruebas de movimiento horizontal como vertical, el prototipo fue capaz de dar seguimiento al movimiento. Incluso también, en el movimiento de cabeza circular, que implica variaciones angulares simultáneas tanto en el plano horizontal como vertical.



Fuente: Fotografías tomadas por el alumno.

Figura 3-8. Pruebas de movimiento webcam

Se debe mencionar una breve observación. En algunos momentos, si bien la webcam sigue el movimiento, éste no es igual de fluido que el movimiento del usuario y presenta leves aceleraciones o irregularidades en la velocidad, la hipótesis para este efecto, es que la latencia

de envío de datos, que varía entre 110 a 143 ms, puede generar este efecto. Una mayor fluidez se puede alcanzar con latencias más bajas y con una mayor cantidad de muestras por segundo que lleguen al robot.

Dado los antecedentes expuesto, se puede aseverar que las pruebas son exitosas y el robot es capaz de mover la webcam siguiendo los movimientos del usuario.

3.2. COSTOS DEL PROYECTO

En este apartado se detallarán los costos asociados a los materiales y a los componentes utilizados en el desarrollo e implementación del prototipo, además de los recursos humanos empleados.

3.2.1. Costos

Para este proyecto no se realizaron mayores cotizaciones de componentes, puesto que si bien solo algunos componentes se cotizaron en distintas tiendas, las variaciones de precio de los productos no tenían una magnitud suficiente como para justificar la generación de múltiples cotizaciones con la lista de materiales. A continuación, se expone en la tabla 3-1, la lista de materiales y precios, junto con el valor total de los mismos.

Tabla 3-1. Proveedores por componente y/o material.

Materiales	Cantidad	Costo unitario (CLP)	Costo unitario (UF)	Total (CLP)	Total (UF)
Webcam Logitech C270	1	\$34.990	1,17	\$34.990	1,17
Servo Motor MG996	2	\$19.450	0,65	\$38.900	1,31
Motor 12 V Alto Torque	2	\$15.000	0,50	\$30.000	1,01
Batería 12V	2	\$14.000	0,47	\$28.000	0,94
Pack cables Dupont	1	\$6.500	0,22	\$6.500	0,22
Arduino Nano alternativo	2	\$6.500	0,22	\$13.000	0,44
Arduino Mega alternativo	1	\$14.990	0,50	\$14.990	0,50
H bridge BTS7960	2	\$11.500	0,39	\$23.000	0,77
LattePanda V1	1	\$129.100	4,33	\$129.100	4,33
Piezas impresas 3D	1	\$220.000	7,39	\$220.000	7,39
Conversor DC/DC	1	\$15.990	0,54	\$15.990	0,54
Ruedas	2	\$9.900	0,33	\$19.800	0,66
Modulo Bluetooth HC-06	1	\$5.990	0,20	\$5.990	0,20
Modulo Joystick	1	\$2.490	0,08	\$2.490	0,08
Pilas AAA	3	\$1.000	0,03	\$3.000	0,10
PCB Virgen	1	\$6.000	0,20	\$6.000	0,20
Total Materiales				\$591.750	19,86

Fuente: Tabla creada por el alumno.

3.2.2. Costos por mano de obra

Para analizar los costos por mano de obra se realizó una búsqueda de la media salarial de un Técnico en electrónica, en su primer año de trabajo y considerando que está trabajando en territorio Chileno. Se encontró este valor en la página web www.computrabajo.cl, desde donde se calculó el valor por hora, el cual es \$3.069. Por tanto para determinar el costo por mano de obra se utilizará esta suma. Con relación al tiempo dedicado en el desarrollo y construcción del prototipo se estima que se hizo una dedicación de tiempo de alrededor de 160 Horas. Considerando estas variables, el cálculo del costo por mano de obra queda en \$491.040. El valor de la UF utilizado es de \$29.789,24 y corresponde al día jueves 12 de Agosto del 2021.

Tabla 3-2. Costos por mano de obra del proyecto.

Horas dedicadas	Costo hora-hombre (CLP)	Costo hora-hombre (UF)	Costo total por mano de obra (CLP)	Costo total por mano de obra (UF)
160	\$3.069	0,1	\$491.040	16,49

Fuente: Tabla creada por el alumno.

3.2.3. Costo Total

En base a los costos anteriormente detallados, tanto de materiales, como también de horas hombre invertidas en el desarrollo del proyecto, se adjunta a continuación en la tabla 3-3, los costos totales del proyecto, considerando los costos tanto en CLP como UF.

Tabla 3-3. Costo total del proyecto.

Costo Total Materiales(CLP)	Costo Total Materiales(UF)	Costo Total mano de obra (CLP)	Costo total por mano de obra (UF)	Costo Total Proyecto(CLP)	Costo Total Proyecto(UF)
\$591.750	19,86	\$491.040	16,49	\$1.082.790	36,35

Fuente: Tabla creada por el alumno.

CONCLUSIONES

Los resultados prácticos lograron cumplir con los objetivos generales y específicos con los que se inició este proyecto.

Desde una perspectiva técnica, el prototipo cumple con todas las funcionalidades esperadas. Se considera también que obviamente, ya que es solo un prototipo, el robot puede ser mejorado notablemente. Esto, por supuesto, implicaría una mayor inversión de horas hombre en términos de desarrollo y búsqueda de otros componentes o protocolos de comunicación que logren el perfeccionamiento en las funciones del robot.

La tecnología aplicada en este proyecto, en conjunto con la arquitectura que posee el prototipo, abren la posibilidad de poder escalar este proyecto hacia niveles de mayor complejidad, ya que no solamente se puede transmitir la información angular de la vista o la dirección en la que se espera que se desplace el robot. Eventualmente, el robot podría tener otro tipo de extremidades, tales como un brazo por ejemplo, y si esto fuese así, se podría brindar la posibilidad para que los individuos pudieran realizar diversos tipos de actividades en forma remota, lo que permitiría abarcar aún, más tipos de trabajo que se podrían transformar a modalidad remota, gracias a esta tecnología, además, este tipo de desarrollo también podría ser utilizado por personas que tengan algún tipo de discapacidad física, facilitando así los procesos de integración en las empresas.

Esta tecnología no solamente puede ser aplicable a los problemas surgidos por la pandemia del COVID-19, sino que también puede dar solución o puede surgir como alternativa para proteger y mantener la integridad física de otros tipos de trabajo en que también los individuos se verán enfrentados a algún tipo de riesgo o peligro. Este punto es especialmente relevante, puesto que, desde una perspectiva social, se está creando una tecnología que utiliza la robótica pero no como un remplazo del ser humano sino como un complemento, el cual protege y reduce el riesgo de accidente para el trabajador. Por otro lado, presenta beneficios económicos para la industria, ya que al proteger al trabajador, evitamos que éste sufra algún tipo de perjuicio, lo que conlleva a pérdidas en los recursos humanos de la empresa.

BIBLIOGRARFÍA

Drive Motor Sizing Tutorial. [En línea].

<<https://www.robotshop.com/community/blog/show/drive-motor-sizing-tutorial>> [consulta: 2 de Julio de 2021].

Consulta el salario medio de Técnico electrónica en Chile. [En línea].

<<https://www.computrabajo.cl/salarios/tecnico-electronica>> [consulta: 3 de Agosto de 2021].

COMUNICACIÓN RS485 CON ARDUINO. [En línea].

<https://naylampmechatronics.com/blog/37_comunicacion-rs485-con-arduino.html> [consulta: 2 de Julio de 2021].

Cloruro de hierro (III).[En línea].

< [https://es.wikipedia.org/wiki/Cloruro_de_hierro_\(III\)](https://es.wikipedia.org/wiki/Cloruro_de_hierro_(III))> [consulta: 9 de Agosto de 2021].

Brote de enfermedad por coronavirus (COVID-19). [En línea].

< <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019>> [consulta: 2 de Mayo de 2021].

Consulta el salario medio de Técnico electrónica en Chile. [En línea].

<<https://www.computrabajo.cl/salarios/tecnico-electronica>> [consulta: 3 de Agosto de 2021].

Three Methods to Filter Noisy Arduino Measurements. [En línea].

<<https://www.megunolink.com/articles/coding/3-methods-filter-noisy-arduino-measurements/>> [consulta: 2 de Julio de 2021].

GLOSARIO

CPU: Del inglés, Central Processing Unit (Unidad central de proceso). Es una máquina programable que responde a un listado de instrucciones específicas pregrabadas.

HARDWARE: Corresponde al conjunto de materiales y componentes físicos que conforman un sistema informático o una computadora.

SOFTWARE: Es un conjunto de códigos de programación y rutinas que permiten a la computadora ejecutar determinadas tareas.

PCB: del inglés, Printed Circuit Board (Tablero de circuito impreso). Es una placa hecha a base de un material no conductor de electricidad, con caminos o pistas fabricadas de un material conductor, utilizadas para conectar componentes electrónicos.

Gravity: Kit Desarrollado por la tienda DFRobot, compuesto por herramientas electrónicas modulares, plug and play de alta calidad para que cualquier tipo de individuo, pueda crear proyectos de manera fácil.

ANEXOS**PROGRAMAS****ARDUINO MEGA 2560 EN LUMBAR SHIELD**

```
#include "BTS7960.h"

const uint8_t EN = 8;
const uint8_t L_PWM = 9;
const uint8_t R_PWM = 10;
const uint8_t EN2 = 4;
const uint8_t L_PWM2 = 5;
const uint8_t R_PWM2 = 6;

int speed = 255;
int Mov_Base = 0;

BTS7960 motorController(EN, L_PWM, R_PWM);
BTS7960 motorController2(EN2, L_PWM2, R_PWM2);

const int mydireccion = 101; //Direccion del esclavo

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);

  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
```

```
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000);
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000);          // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000);
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000);          // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000);
```

```
Serial.begin(9600);
Serial.setTimeout(10); //establecemos un tiempo de espera de 10ms
motorController.Enable();
motorController2.Enable();
```

```
motorController.Stop();
motorController2.Stop();
```

```
delay(2000);
```

```
motorController.TurnRight(speed);
motorController2.TurnRight(speed);
```

```
delay(2000);
```

```
motorController.TurnLeft(speed);
motorController2.TurnLeft(speed);
```

```
delay(2000);
```

```
motorController.TurnRight(speed);
```

```
motorController2.TurnLeft(speed);
```

```
delay(2000);
```

```
motorController.TurnLeft(speed);
```

```
motorController2.TurnRight(speed);
```

```
delay(2000);
```

```
motorController.Stop();
```

```
motorController2.Stop();
```

```
delay(2000);
```

```
motorController.Disable();
```

```
motorController2.Disable();
```

```
}
```

```
void loop()
```

```
{
```

```
  if(Serial.available())
```

```
  {
```

```
    if(Serial.read()=='I') //Si recibimos el inicio de trama
```

```
    {
```

```
      int direccion=Serial.parseInt(); //recibimos la direccion
```

```
      if(direccion==mydireccion) //Si direccion es la nuestra
```

```
{  
char funcion=Serial.read(); //leemos el carácter de función  
  
//---Si el carácter de función es una S entonces la trama es para mover el motor-----  
if(funcion=='S')  
{  
    Mov_Base=Serial.parseInt(); //recibimos la instruccion  
    if(Serial.read()=='F') //Si el fin de trama es el correcto  
    {  
        if(Mov_Base < 6) //verificamos que sea un valor en el rango de la instruccion  
        {  
            motorController.Enable();  
            motorController2.Enable();  
            switch (Mov_Base) {  
            case 0:  
                motorController.Stop();  
                motorController2.Stop();  
                motorController.Disable();  
                motorController2.Disable();  
                break;  
            case 1:  
                motorController.TurnRight(speed);  
                motorController2.TurnRight(speed);  
                break;  
            case 2:  
                motorController.TurnLeft(speed);  
                motorController2.TurnLeft(speed);  
                break;  
            case 3:
```


ARDUINO NANO EN CERVICAL SHIELD

```
#include <Servo.h>
```

```
int Angle_H = 0;
```

```
int Angle_V = 0;
```

```
Servo myservo; // create servo object to control a servo
```

```
Servo myservo2; // create servo object to control a servo
```

```
const int mydireccion = 102; //Direccion del esclavo
```

```
void setup()
```

```
{
```

```
  pinMode(LED_BUILTIN, OUTPUT);
```

```
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
  myservo2.attach(10); // attaches the servo on pin 10 to the servo object
```

```
  myservo.write(95);
```

```
  myservo2.write(94);
```

```
  delay(1500);
```

```
  Serial.begin(9600);
```

```
  Serial.setTimeout(10); //establecemos un tiempo de espera de 10ms
```

```
}
```

```
void loop()
```

```
{
```

```
  if(Serial.available())
```

```
  {
```

```

if(Serial.read()=='I') //Si recibimos el inicio de trama
{
  int direccion=Serial.parseInt(); //recibimos la direccion
  if(direccion==mydireccion) //Si direccion es la nuestra
  {
    char funcion=Serial.read(); //leemos el carácter de función

    //---Si el carácter de función es una P entonces a continuacion viene el angulo del 1er
Motor-----
    if(funcion=='P')
    {
      Angle_V=Serial.parseInt(); //recibimos la instruccion
      char funcion2 =Serial.read(); //leemos el carácter de función
      //---Si el carácter de función es una P entonces a continuacion viene el angulo del 1er
Motor-----
      if(funcion2=='S')
      {
        Angle_H=Serial.parseInt(); //recibimos la instruccion
        if(Serial.read()=='F') //Si el fin de trama es el correcto
        {
          Angle_H = map(Angle_H, 0, 180, 160, 30); // scale it to use it with the servo (value
between 0 and 180)
          Angle_V = map(Angle_V, 0, 180, 29, 158); // scale it to use it with the servo (value
between 0 and 180)
          myservo.write(Angle_V); // sets the servo position according to the scaled
value
          myservo2.write(Angle_H); // sets the servo position according to the
scaled value
        }
      }
    }
  }
}

```

```
    }  
  }  
}  
}  
}  
}  
delay(10);  
}
```

ARDUINO NANO EN CONTROL SHIELD

```
float AverageX = 0;  
float AverageY = 0;  
int MeasurementsToAverage = 20;  
  
const int JoyX = A0;  
const int JoyY = A1;  
const int JoyBtn = 12;  
int X = 0;  
int Y = 0;  
int Xval_New = 0;  
int Yval_New = 0;  
bool BtnVal_New = false;  
int Xval = 0;  
int Yval = 0;  
bool BtnVal = false;  
  
void setup() {  
  pinMode(JoyBtn , INPUT_PULLUP); //activar resistencia pull up  
  Serial.begin(9600);
```

```
}

```

```
void loop() {

```

```
  for(int i = 0; i < MeasurementsToAverage; ++i)

```

```
  {

```

```
    AverageX += analogRead(JoyX);

```

```
    delay(1);

```

```
    AverageY += analogRead(JoyY);

```

```
    delay(1);

```

```
  }

```

```
  AverageX /= MeasurementsToAverage;

```

```
  AverageY /= MeasurementsToAverage;

```

```
  X = int(AverageX);

```

```
  Y = int(AverageY);

```

```
  Xval_New = map(X, 0, 1023, 0, 100); // Se escala lectura analoga en X de rango 0-1023 a 0-255

```

```
  Yval_New = map(Y, 0, 1023, 0, 100); // Se escala lectura analoga en Y de rango 0-1023 a 0-255

```

```
  BtnVal_New = digitalRead(JoyBtn);

```

```
  if ((Xval_New != Xval ) || (Yval_New != Yval ) || (BtnVal_New != BtnVal )) {

```

```
    //Si hubo una variacion en X o Y o Btn, se muestran valores por serial

```

```
    Serial.print(Xval_New);

```

```
    Serial.print(":");

```

```
    Serial.print(Yval_New);

```

```
    Serial.print(":");

```

```
    Serial.println(BtnVal_New);

```

```
  }

```

//Se guardan los actuales valores de X,Y Btn, para compraralos en el futuro con los nuevos valores leidos

Xval = Xval_New;

Yval = Yval_New;

BtnVal = BtnVal_New;

delay(200)

Aplicación Smartphone en APP Inventor

```

initialize global IP to " http://[redacted].8.107/? "
initialize global IP_Cam to " http://[redacted].8.107:8081/ "
initialize global All_Data to " "
initialize global BT_Data to " 0,0,0 "
initialize global Delta to 0
initialize global Average_Counter to 0
initialize global Average_Z to 0
initialize global Average_Azimuth to 0
initialize global Center_Azimuth to -1
initialize global Horizontal to 0
initialize global Vertical to 0
initialize global Send to false

when Screen1.Initialize
do
  set AccelerometerSensor1.Enabled to true
  set OrientationSensor1.Enabled to true

when Clock5.Timer
do
  call Notifier1.ShowAlert notice " "

when Clock1.Timer
do
  set global Average_Z to (get global Average_Z + AccelerometerSensor1.ZAccel)
  set global Average_Azimuth to (get global Average_Azimuth + OrientationSensor1.Azimuth)
  set global Average_Counter to (get global Average_Counter + 1)
  if (get global Average_Counter = 15)
  then
    set global Average_Z to (get global Average_Z / 15)
    set global Average_Azimuth to (get global Average_Azimuth / 15)
    set global Vertical to (call Scale Variable get global Average_Z)
    call Ref_Angle.Angle round (get global Average_Azimuth)
    set global Average_Counter to 0
    set global Average_Z to 0
    set global Average_Azimuth to 0
  
```

```

when Clock2.Timer
do
  set global All_Data to join...
  set Label1.Text to get global All_Data
  if get global Send
  then
    set WebView1.HomeUrl to call Make_URL
    set Label4.BackgroundColor to green
  else
    set Label4.BackgroundColor to red

```

```

when Button1.Click
do
  set global Send to true

```

```

when Button2.Click
do
  set global Send to false

```

```

to Make_URL
result
  join
    get global IP
    get global All_Data

```

```

to Scale Variable
result
  round
    (get Variable + 10) * 180 / 20

```

```
to Ref_Angle Angle
do
  if get global Center_Azimuth = -1
  then set global Horizontal to get Angle
  else set global Delta to get Angle - get global Center_Azimuth
      if absolute get global Delta > 180
      then
        if get global Center_Azimuth > 270
        then set global Horizontal to 450 + get global Delta
        else set global Horizontal to -270 + get global Delta
      else set global Horizontal to 90 + get global Delta
```

```
when Clock3 .Timer
do
  if not BluetoothClient1 . IsConnected
  then
    if call BluetoothClient1 .Connect
       address " 98:DA:60:00:BE:98 "
    then
      if BluetoothClient1 . IsConnected
      then set Label3 . BackgroundColor to green
      else set Label3 . BackgroundColor to red
```

