

# FACULTAD DE INGENIERÍA



Carrera de Ingeniería Mecatrónica

“IMPLEMENTACIÓN DE UN ROBOT EXPLORADOR  
CON REALIDAD VIRTUAL PARA INCREMENTAR LA  
SEGURIDAD DE LA INSTITUCIÓN EDUCATIVA 7213  
PERUANO JAPONES EN EL DISTRITO DE VILLA EL  
SALVADOR – 2020.”

Tesis para optar el título profesional de:

Ingeniero Mecatrónico

Autor:

Bach. Anthony Sotelo Gomez

Asesor:

Ing. Jorge Luis Contreras Cossío

Lima - Perú

2020

## ACTA DE AUTORIZACIÓN PARA SUSTENTACIÓN DE TESIS

El asesor Ing. Jorge Luis Contreras Cossio, docente de la Universidad Privada del Norte, Facultad de Ingeniería, Carrera profesional de INGENIERÍA MECATRÓNICA, ha realizado el seguimiento del proceso de formulación y desarrollo de la tesis del estudiante:

- Anthony Sotelo Gomez

Por cuanto, **CONSIDERA** que la tesis titulada: “Implementación de un robot explorador con realidad virtual para incrementar la seguridad de la Institución Educativa 7213 Peruano Japones en el distrito de Villa el Salvador – 2020” para aspirar el título profesional de: Ingeniero Mecatrónico por la Universidad Privada del Norte, reúne las condiciones adecuadas, por lo cual, **AUTORIZA** al o a los interesados para su presentación.

---

Ing. Jorge Luis Contreras Cossío

Asesor

## ACTA DE APROBACIÓN DE LA TESIS

Los miembros del jurado evaluador asignados han procedido a realizar la evaluación de la tesis del estudiante: Anthony Sotelo Gomez para aspirar al título profesional con la tesis denominada: “Implementación de un robot explorador con realidad virtual para incrementar la seguridad de la Institución Educativa 7213 Peruano Japones en el distrito de Villa el Salvador – 2020”

Luego de la revisión del trabajo, en forma y contenido, los miembros del jurado concuerdan:

### **( ) Aprobación por unanimidad**

Calificativo:

( ) Excelente [20 - 18]

( ) Sobresaliente [17 - 15]

( ) Bueno [14 - 13]

( ) Desaprobado

Firman en señal de conformidad:

### **( ) Aprobación por mayoría**

Calificativo:

( ) Excelente [20 - 18]

( ) Sobresaliente [17 - 15]

( ) Bueno [14 - 13]

## DEDICATORIA

A mis padres por ser parte fundamental en mi vida e inspirarme a siempre exigirme al máximo y nunca darme por vencido.

A mi esposa por darme su amor y él apoyó con el cuidado de mis hijos en esta última parte de mi carrera.

A mis hijos, fuente de mi deseo de superación, que con su ternura lograron cambiarme la vida.



## AGRADECIMIENTO

En primer lugar, a los grandes maestros dentro y fuera de la universidad que me han acompañado en este crecimiento profesional a lo largo de este tiempo; en segundo lugar, a mi PADRE y MADRE que me demostraron que con mucho esfuerzo y perseverancia se puede lograr lo que uno anhela; Por haberme dado fuerza y apoyo incondicional que me ha ayudado y llevado hasta donde estoy ahora. Por último, a mis grandes amigos en WSP PERÚ.

## TABLA DE CONTENIDOS

<b>ACTA DE AUTORIZACIÓN PARA SUSTENTACIÓN DE TESIS .....</b>	<b>ii</b>
<b>ACTA DE APROBACIÓN DE LA TESIS .....</b>	<b>iii</b>
DEDICATORIA .....	iv
AGRADECIMIENTO.....	v
TABLA DE CONTENIDOS.....	vi
INDICE DE TABLAS .....	ix
INDICE DE FIGURAS.....	x
RESUMEN.....	xiv
SUMMARY .....	xvi
<b>CAPÍTULO I. INTRODUCCIÓN .....</b>	<b>1</b>
1.1.    Realidad problemática .....	1
1.2.    Formulación del problema .....	9
1.3.    Objetivos.....	9
1.3.1.  Objetivo General .....	9
1.3.2.  Objetivos específicos.....	9
1.4.    Antecedentes .....	10
1.4.1.  Antecedentes Nacionales.....	10
1.4.2.  Antecedentes Internacionales.....	11

1.5.	Hipótesis .....	14
1.5.1.	Hipótesis general .....	14
1.5.2.	Hipótesis específicas .....	14
CAPÍTULO II. METODOLOGÍA.....		15
2.1.	Tipo de investigación.....	15
2.1.1.	Según el propósito.....	15
2.1.2.	Según el diseño de investigación .....	16
2.1.3.	Línea de investigación.....	16
2.1.4.	Tema / Eje temático.....	17
2.1.5.	Institución donde se desarrollará el proyecto.....	18
2.1.6.	Distrito, Provincia, Región.....	20
2.2.	Materiales, instrumentos y métodos .....	21
2.2.1.	Materiales .....	21
2.2.2.	Instrumentos .....	22
2.2.3.	Métodos.....	22
2.3.	Desarrollo de la aplicación.....	23
2.3.1.	Diseño mecánico del robot explorador.....	23
2.3.2.	Diseño del Sistema electrónico .....	33
2.3.3.	Programación del sistema de control y la arquitectura de comunicación .....	53

CAPÍTULO III. RESULTADOS .....	76
CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES .....	80
4.1.    Discusión.....	80
4.2.    Conclusiones .....	81
REFERENCIAS .....	82
ANEXOS.....	84

## INDICE DE TABLAS

Tabla 1 Población de 15 años de edad a más, con percepción de inseguridad en los próximos doce meses, 2017-2019(%) .....	1
Tabla 2 Principales problemas del país – Semestre: julio – diciembre 2018 / julio – diciembre 2019 (porcentajes).....	2
Tabla 3 Percepción de inseguridad por tipo de hecho delictivo.....	3
Tabla 4 Tasa de victimización.....	3
Tabla 5 Especificaciones Técnicas del controlador NodeMCU-V3. ....	34
Tabla 6. Especificaciones Técnicas - Motor XD-37GB520.....	35
Tabla 7 Tabla de verdad para giro horario .....	36
Tabla 8 Tabla de verdad para giro antihorario .....	37
Tabla 9 Especificaciones Técnicas - Servomotor TowerPro MG90S.....	39

## INDICE DE FIGURAS

Figura 1. Denuncias realizadas en cada distrito entre enero y septiembre del 2019.....	4
Figura 2. Tasa de homicidios por distritos. ....	5
Figura 3. Mapa de criminalidad por robo en flagrancia alrededor del colegio Peruano Japonés .....	6
Figura 4. Institución Educativa N° 7213 Peruano Japonés. ....	19
Figura 5. Perímetro Institución Educativa 7213 Peruano Japonés.....	19
Figura 6. Croquis del I.E. Peruano Japonés 7213. ....	20
Figura 7. Plano en explosión isométrico del robot explorador. ....	23
Figura 8. En azul las ruedas de transmisión y en rojo las ruedas intermedias. ....	24
Figura 9. Isométrico de robot explorador.....	24
Figura 10. Dimensionamiento de ruedas de acuerdo con el diámetro. ....	25
Figura 11. Diagrama de cuerpo libre en pendiente. ....	26
Figura 12. Análisis de Fuerzas en una rueda.....	27
Figura 13. Coeficientes de fricción. ....	29
Figura 14. Descomposición del peso. ....	29
Figura 15. Sistema roll y pitch. ....	31
Figura 16. Stress Analysis de Inventor. ....	32
Figura 17. Controlador NodeMCU-V3 .....	33
Figura 18. Motor DC con motor reductor XD-37GB520 .....	36
Figura 19. Diagrama de bloques de datasheet L298. ....	37
Figura 20. Diagrama de bloques de datasheet L298. ....	38
Figura 21. Servomotor TowerPro MG90S.....	39

Figura 22. Ancho de pulso vs ángulo.....	40
Figura 23. Ancho de pulso vs ángulo.....	41
Figura 24. Ancho de pulso vs ángulo.....	41
Figura 25. Esquemático electrónico del robot explorador. ....	42
Figura 26. Diseño y ubicación de componentes electrónicos en PCB.....	43
Figura 27. Vista final del PCB a fabricar en fibra de vidrio. ....	43
Figura 28. Vista final de PCB - Robot Explorador .....	44
Figura 29. Vista Superior de PCB - Robot Explorador.....	44
Figura 30. Lentes de realidad Virtual - VRBOX .....	45
Figura 31. Giroscopio MPU6050.....	45
Figura 32. Esquema de conexiones de electrónica en lentes de Realidad Virtual. ....	46
Figura 33. Ubicación de controlador NodeMCU y MPU6050 al interior del visor VR. ....	47
Figura 34. MPU6050 con plano paralelo y plano perpendicular al suelo. ....	47
Figura 35. Diagrama para hallar el desplazamiento angular.....	48
Figura 36. Mediciones obtenidas con el MPU6050 y aplicando el filtro complementario.....	49
Figura 37. Diagrama electrónico de conexiones – Joystick.....	50
Figura 38. Esquema de conexiones de Joystick de operador.....	51
Figura 39. Divisor de Voltaje de los ejes "x" e "y".....	51
Figura 40. Niveles de ruido y umbral de dolor. ....	52
Figura 41. Sirena tipo turbina.....	53
Figura 42. Prueba de conexión de Cámara IP con protocolo de transporte en tiempo real (RTP).....	54
Figura 43. Arquitectura de comunicación de Protocolo MQTT .....	55

Figura 44. Modelo de estructuración de mensaje JSON.....	56
Figura 45. Diagrama de Flujo de control de los Lentes VR.....	57
Figura 46. Programación Lentes VR - Parte 1 de 9. ....	58
Figura 47. Programación Lentes VR - Parte 2 de 9. ....	59
Figura 48. Programación Lentes VR - Parte 3 de 9. ....	59
Figura 49. Programación Lentes VR - Parte 4 de 9. ....	60
Figura 50. Programación Lentes VR - Parte 5 de 9. ....	60
Figura 51. Programación Lentes VR - Parte 6 de 9. ....	61
Figura 52. Programación Lentes VR - Parte 7 de 9. ....	62
Figura 53. Programación Lentes VR - Parte 8 de 9. ....	63
Figura 54. Programación Lentes VR - Parte 9 de 9. ....	63
Figura 55. Diagrama de Flujo de control del Joystick.....	64
Figura 56. Programación de Joystick - Parte 1 de 5.....	65
Figura 57. Programación de Joystick - Parte 2 de 5.....	65
Figura 58. Programación de Joystick - Parte 3 de 5.....	66
Figura 59. Programación de Joystick - Parte 4 de 5.....	66
Figura 60. Programación de Joystick - Parte 5 de 5.....	67
Figura 61. Diagrama de control Robot explorador VR.....	68
Figura 62. Programación Robot explorador VR - Parte 1 de 8.....	69
Figura 63. Programación Robot explorador VR - Parte 2 de 8.....	70
Figura 64. Programación Robot explorador VR - Parte 3 de 8.....	70
Figura 65. Programación Robot explorador VR - Parte 4 de 8.....	71
Figura 66. Programación Robot explorador VR - Parte 5 de 8.....	72



Figura 67. Programación Robot explorador VR - Parte 6 de 8. ....	73
Figura 68. Programación Robot explorador VR - Parte 7 de 8. ....	74
Figura 69. Programación Robot explorador VR - Parte 8 de 8. ....	75
Figura 70. Prueba realizada en campo con una profesora del centro educacional. ....	77
Figura 71. Prueba realizada sobre terreno no afirmado con pendiente. ....	78
Figura 72. Prueba realizada sobre superficie de concreto. ....	78
Figura 73. Prueba realizada sobre áreas verdes con pendiente. ....	79

## RESUMEN

La presente investigación trata sobre la implementación de un robot explorador con sistema inmersivo de realidad virtual para un centro educativo estatal, con el fin de incrementar la seguridad de los bienes y la eliminación del riesgo de atentado contra la vida del personal de vigilancia, debido a que este ha venido siendo asaltado en reiteradas veces, perdiendo equipos de gran valor que forman parte de las herramientas educacionales del alumnado por la carencia de dispositivos de seguridad al interior del recinto estudiantil.

Una vez identificado y analizado las razones del porqué, como y cuando se dan este tipo de situaciones. Se buscó dotar de una nueva herramienta de seguridad que sea eficaz en el patrullaje y evite la exposición física del personal de seguridad. Es por ello, que se busca implementar un robot capaz de entregar un sistema inmersivo que suplante físicamente a la persona, entregando una experiencia de entorno 100% real durante el patrullaje nocturno.

Teniendo los objetivos claros se pasó a investigar y buscar referencias de información relevante sobre sistemas robóticos que ayuden a las personas en situaciones de difícil acceso o riesgosas. Para ello, se realizó la exploración y revisión de fuentes confiables en repositorios y webs de índole científico estudiantil como lo son: Google Académico, Renati, Alice, Microsoft Academic, entre otros.

Finalmente, se ideó una solución con base en los fundamentos de la ingeniería mecatrónica aplicando los conocimientos adquiridos de mecánica, electrónica e informática a lo largo de la

carrera y de la información recopilada durante la investigación. Por un lado, se realizaron los cálculos mecánicos para el dimensionamiento de las ruedas, motores DC y servomotores, de acuerdo con el requerimiento de torques y velocidades. Una vez definidos los equipos mecánicos, se elige al NODEMCU, como unidad de control, por su versatilidad en el manejo de componentes electrónicos y protocolos de comunicación basados en TCP/IP. Además, se integra un regulador de voltaje y el driver L298 como parte de la electrónica. Por otro lado, se diseñó un visor VR y Joystick de desplazamiento teniendo siempre como controlador central al NODEMCU. Por último, se hace uso del protocolo de comunicación MQTT para la integración de los 3 dispositivos electrónicos mediante un servidor en la nube. Como conclusión, se logró incrementar la eficacia de la detección de intrusos además de que eliminó el riesgo de exposición de la persona a un evento que pueda atentar contra su vida.

**Palabras clave:** Robot explorador, Realidad Virtual, Sistema inmersivo, Mecatrónica, Seguridad.

## SUMMARY

This research deals with the implementation of an explorer robot with a virtual reality system for a state educational center, in order to increase the security of the goods and eliminate the risk of an attempt on the life of the surveillance personnel, because This has been repeatedly assaulted, losing valuable equipment that are part of the educational tools of the students due to the lack of security devices inside the student campus.

Once identified and analyzed the reasons why, how and when this type of situation occurs. It was sought to provide a new security tool that is effective in patrolling and avoids physical exposure of security personnel. For this reason, the aim is to implement a robot capable of delivering an immersive system that physically impersonates the person, delivering a 100% real environment experience during night patrols.

Having clear objectives, we began to investigate and search for references of relevant information on robotic systems that help people in difficult or risky situations. To do this, the exploration and review of reliable sources in repositories and websites of a student scientific nature were carried out, such as: Google Academic, Renati, Alice, Microsoft Academic, among others.

Finally, a solution was devised based on the fundamentals of mechatronics engineering by applying the knowledge acquired in mechanics, electronics and computing throughout the career and the information collected during the research. On the one hand, mechanical calculations were carried out for the dimensioning of the wheels, DC motors and servomotors, according to the requirement of torques and speeds. Once the mechanical equipment has been defined, the

NODEMCU is chosen as the control unit, due to its versatility in handling electronic components and TCP / IP-based communication protocols. In addition, a voltage regulator and the L298 driver are integrated as part of the electronics. On the other hand, a VR viewer and scrolling joystick were designed with the NODEMCU as the central controller. Finally, the MQTT communication protocol is used for the integration of the 3 electronic devices through a server in the cloud. In conclusion, it was possible to increase the effectiveness of intruder detection in addition to eliminating the risk of exposure of the person to an event that threatens his life.

**Keywords:** Robot Explorer, Virtual Reality, Immersive System, Mechatronics, Security.

## CAPÍTULO I. INTRODUCCIÓN

### 1.1. Realidad problemática

En el ámbito Nacional, el Instituto Nacional de Estadística e Informática (INEI) (2019), *Seguridad Ciudadana - Informe Anual 2019 – Desafíos Pendientes*, nos indica que en las ciudades del Perú el 85.6% de la muestra de la población tuvo una percepción de inseguridad. De modo similar, el año anterior entre julio y diciembre del 2018 con un porcentaje de 86,6% de ciudadanos que se sienten en un ambiente altamente vulnerable a cualquier acto delictivo (p.29).

Tabla 1

*Población de 15 años de edad a más, con percepción de inseguridad en los próximos doce meses, 2017-2019(%)*

Ámbito de estudio	Julio- diciembre 2017	Enero- junio 2018	Julio- diciembre 2018	Enero- junio 2019	Julio- diciembre 2019	Variación 2018-2019
Nacional urbano	85.5	85.3	86.6	85.8	85.6	-1.0
Ciudades de 20 mil habitantes a más	87.9	87.2	88.7	87.8	88.0	-0.7
Centros poblados urbanos entre 2 mil y menos de 20 mil habitantes	79.2	80.3	81.1	80.2	79.1	-2.0

Fuente: INEI (2020)

Según la encuesta, la población señala a la corrupción como el problema principal del país. Sin embargo, la delincuencia sumada a la carencia de seguridad ciudadana, que deberían ser consideradas como un único ítem, superan el 50.0% de las personas que sienten que la dificultad principal del país es la inseguridad ciudadana (tabla 2).

El índice de percepción de inseguridad va variando de acuerdo con el tipo de delito. En la tabla de la tabla 3 muestra que la población siente, en su mayoría, que podría ser víctima de un asalto o que puedan atentar contra su vivienda.

Tabla 2

*Principales problemas del país – Semestre: julio – diciembre 2018 / julio – diciembre 2019 (porcentajes)*

Prioridad del problema	Semestre móvil		Variación (%)
	Julio 2018 - diciembre	Julio 2019 - diciembre	
	2018	2019	
Corrupción	65.1	62.1	-3.0
Delincuencia	38.4	40.9	2.5
Pobreza	20.3	20.0	-0.3
Falta de seguridad ciudadana	13.6	16.0	2.4
Falta de empleo	12.9	13.3	0.4
Mala calidad de la educación estatal	8.7	10.3	1.5

Fuente: INEI (2020)

Tabla 3

*Percepción de inseguridad por tipo de hecho delictivo.*

Tipo de hecho delictivo	Julio- diciembre 2017	Enero-junio 2018	Julio- diciembre 2018	Enero-junio 2019	Julio- diciembre 2019	Variación 2018-2019
Robo de dinero, cartera, celular	76.2	75.6	77.4	76.7	77.0	-0.4
Robo a su vivienda	70.5	69.7	71.0	69.9	70.2	-0.8
Amenazas e intimidaciones	37.9	35.7	36	36.2	37.3	1.3
Robo de vehículo	31.3	30.6	32.0	30.4	30.5	-1.5
Extorsión	23.9	21.4	23.4	22.0	22.7	-0.7

Fuente: INEI (2020)

Lo más preocupante es la tasa de victimización, estas estadísticas son el porcentaje de las personas que fueron víctimas de un acto delictivo, alcanzando el 26,7% en el 2019, lo cual indica que 1 de cada 4 peruanos afirma haber sufrido un acto delincuencia. De igual manera, el año anterior con un porcentaje del 25.7%.

Tabla 4

*Tasa de victimización*

Ámbito de estudio	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Nacional urbano	33.1	40.0	36.6	35.9	30.5	30.8	28.8	26.4	25.7	26.7
Ciudades de 20 mil habitantes a más	35.1	43.4	39.4	38.8	33.1	33.5	31.0	28.8	27.9	29.6
Centros poblados urbano entre 2 mil y menos de 20 mil habitantes	27.6	30.7	29.3	27.7	23.4	23.4	22.7	19.6	19.6	19.1

Fuente: INEI (2020)



De forma específica, en el distrito de Villa el Salvador donde se implementa este proyecto, se reportaron 2811 denuncias (fig.5) entre enero y septiembre del 2019. Sin embargo, esta cifra no es totalmente cierta, ya que el INEI indica que solo el 17.2% de la población, que ha sido víctima de algún acto delictivo, registra una denuncia. Esto quiere decir que de una muestra de 10 personas victimizadas menos de 2 terminan en denuncias registradas. Esto a causa de la pérdida de confianza en la institución policial, ya que se indica que solo el 19.2% de la población confía en la policía.



*Figura 1.* Denuncias realizadas en cada distrito entre enero y septiembre del 2019.

Fuente: INEI (2020)

Los actos delincuenciales se van tornando cada vez más violentos por la escasa seguridad y las leyes no tan rigurosas. Por ejemplo, se tienen estadísticas alarmantes de robo a mano armada que lamentablemente terminaron en un desenlace fatal para el violentado, acabando con su vida. La Figura 5 nos entrega el número de homicidios por violencia delincencial, solo entre enero y septiembre del 2019 se registraron 29 decesos en el distrito de Villa el Salvador.

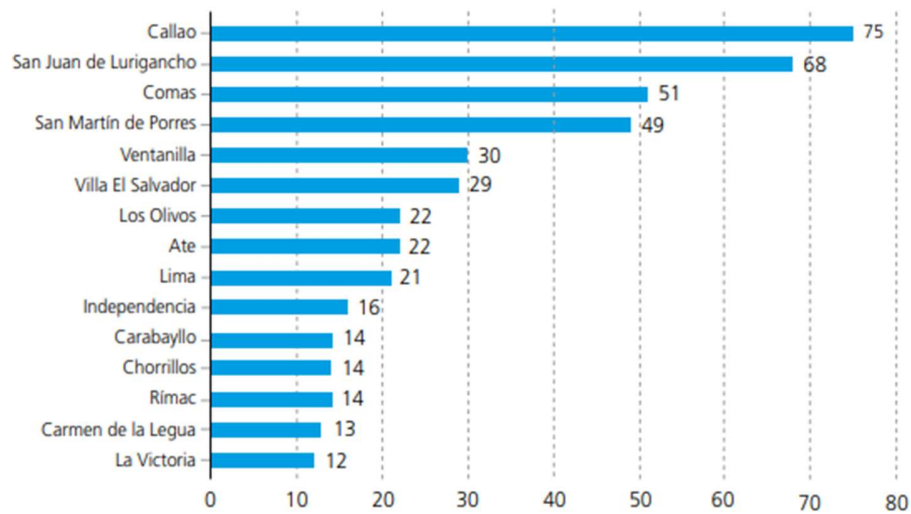


Figura 2. Tasa de homicidios por distritos.

Fuente: INEI (2020)

Una de las funciones del estado es proveer de seguridad a su población, sin embargo, esta no siempre llega. Es por ello, que se buscan alternativas de seguridad privadas. Cárdenas (2015) afirma: “Los métodos de vigilancia brindados por la Policía Nacional o el Serenazgo no son del todo efectivos. Además, existe la necesidad de implementar mecanismos que refuercen la seguridad y permitan disminuir la cantidad de robos” (p. 6).



*Figura 3.* Mapa de criminalidad por robo en flagrancia alrededor del colegio Peruano Japonés

Fuente: Google Maps & INEI

En el lugar elegido para el estudio, la Institución Educativa Peruano Japonés 7213, dedicada a la educación pública desde el año 1989, se ha ido equipando de nuevas herramientas potenciadoras de aptitudes técnicas, que han demandado, con el paso del tiempo, altos esfuerzos económicos, pero lamentablemente se ha venido descuidando en el control de la seguridad del recinto fuera del horario estudiantil, desencadenando múltiples robos a la fecha, de acuerdo con lo registrado por el Sistema integrado de estadísticas de la criminalidad y seguridad ciudadana (DATA CRIM), los cuales afectan directamente al desarrollo y a la reinversión educacional.

Entre las diferentes causas que pueden estar originando este problema se ha detectado las siguientes: Falta de iluminación adecuada durante las noches de patrullaje, facilidad de reducción al único personal de seguridad, pared colindante a zona desolada, no contar con sistema de videovigilancia en zonas críticas, falta de barreras de intrusión perimetral, grandes áreas abiertas y espacios donde ocultarse.

Los efectos que pueden traer la inseguridad son muchos, entre los que se ha detectado están la pérdida de equipos de gran valor, atentado contra la vida del personal de vigilancia, imposibilidad de sostener en el tiempo una estrategia de reinversión en desarrollo y tecnología educacional.

Por ello, la presente investigación pretende incrementar la seguridad de la Institución educativa Peruano Japonés 7213 mediante una solución dentro de los conocimientos en electrónica, mecánica e informática. La cual entregue al centro educativo un práctico, seguro y eficiente sistema de monitoreo de su territorio. El sistema se basa en un robot explorador con realidad virtual que servirá como interfaz entre hombre y máquina. Este podrá solo reemplazar físicamente al conserje entregándole la misma experiencia de patrullaje del espacio recorrido. Adicionalmente, se integra nuevas herramientas para el escaneo rápido y eficiente de ambientes pudiendo llegar a cubrir mayor área de la que normalmente puede una persona. Mediante la implementación de una cámara con lentes infrarrojos se perimirá la visualización en ambientes oscuros o de poca luz, para reconocer rápidamente la silueta de una persona e informar a las autoridades y/o proceder con la activación de alarmas de Centro Educativo.

Finalmente, este proyecto beneficiará al colegio y los más de 30 000 alumnos, ya que se podrá volver a invertir en el desarrollo y tecnología educacional, que es el objetivo principal del colegio, en vez de reponer nuevamente las herramientas hurtadas.

## **1.2. Formulación del problema**

¿De qué manera la implementación de un robot explorador con realidad virtual incrementaría la seguridad de la Institución Educativa 7213 Peruano Japonés en el distrito de Villa el Salvador – 2020?

## **1.3. Objetivos**

### **1.3.1. Objetivo General**

Implementar un robot explorador con realidad virtual para incrementar la seguridad en la Institución Educativa 7213 Peruano Japonés del distrito de Villa el Salvador – 2020

### **1.3.2. Objetivos específicos**

- Incrementar la posibilidad de detección temprana de un intento de robo.
- Aumentar la eficiencia de patrullaje con el uso de realidad virtual y sensores de movimiento inalámbricos.
- Evitar poner en riesgo la salud e integridad del personal de seguridad frente a un atentado.

## 1.4. Antecedentes

### 1.4.1. Antecedentes Nacionales

**Morzan, A & Zerga B. (2018)** en su tesis denominada: “*Diseño e implementación de robot terrestre teleoperado aplicando imágenes térmicas para mitigar incendios en residenciales operado por bomberos victoria N°08*”, de la Universidad Ricardo Palma, en su conclusión indica que: “Fue posible determinar las condiciones que permitan al robot terrestre teleoperado, expulsar agua con precisión, la cual fue demostrada en la prueba tres. Gracias a la ayuda de la calidad de imagen térmica obtenida, fue posible distinguir focos de alta temperatura, direccionando en tiempo real el flujo de agua para una mitigación de fuego eficiente” (p.122).

La investigación se desarrolla en creación y puesta en marcha de un robot terrestre teleoperado capaz manejar agua para la mitigación de fuego. Este posee una cámara infrarroja para la captación de imágenes térmicas, un diseño robusto aprueba de sobrecalentamiento y una interfaz amigable para el operador, con el cual el operario logra interactuar con ambientes hostiles, sin poner en riesgo su vida, enfocando su capacidad de extensión del fuego eficientemente de manera oportuna y rápida. Dicho robot sirve de reemplazo físico al operario. Sin embargo, el usuario es el que toma las decisiones finales mediante la evaluación de imágenes térmicas. La investigación concluye afirmando que dicho diseño es una nueva herramienta, eficaz y poderosa, para las labores arriesgadas de los bomberos. Es por ello, que el uso del robot para la extinción total del fuego será determinante.

**Lovon, P. (2019).** En su tesis denominada: “Desarrollo de una Plataforma en Android para Teleoperación de un Sistema Robótico”, de la Universidad Nacional de San Agustín, en su conclusión indica que: “Se concluye que la plataforma de teleoperación propuesta, basado en los experimentos realizados, puede ser usada en diferentes tipos de robot. Esto debido a que el Robot Operating System permite dicha compatibilidad realizando pequeños cambios” (p.52).

La tesis se desarrolla en el diseño de una interfaz en Android haciendo uso del lenguaje de programación Java y del entorno de desarrollo integrado Android Studio como solución a las interfaces hombre-maquina tradicionales. Estos funcionan con múltiples dispositivos vinculados a la maquina repartiendo sus funcionalidades entre ellos, causando que el sistema no sea nada amigable con el usuario final u operador, además de poseer una arquitectura de control no muy confiable. Para el cual, en esta tesis se logra la fusión sensorial unificando las señales de los sensores y actuadores en una sola interfaz amigable que sirva de ayuda al operador para que se pueda así ejecutar comando o funciones de manera correcta.

#### **1.4.2. Antecedentes Internacionales**

**González, G. (2011).** En su tesis doctoral denominada “*Mecanismos de cooperación en robots como agentes móviles*”, de la Universidad Autónoma de Madrid, en su conclusión indica que: “El usuario final tan sólo debe decidir, en función de las necesidades de la aplicación y los recursos de los que dispone, qué partes son las que necesita adquirir y cuáles las aporta él mismo”. (p.174).



La tesis se desarrolla en la utilización de los diversos conocimientos que implican el desarrollo de un robot de colaboración, como es la aplicación de la electrónica, software, física y mecánica. Este estudio ahonda las grandes dificultades que tienen diversos especialistas frente al campo de la robótica, pero lo más resaltante es la dificultad que se tiene en el uso de redes y protocolos TCP/IP, para el cual el autor da uso de lo que él denomina “Servidor sensorial”. Este servidor se encuentra instalado en el propio robot. El cual se encarga de recopilar datos de los sensores y gestionarlos sobre los actuadores para lograr cumplir una tarea de manera efectiva.

**Borja, R. & Bravo, H. (2016).** En su tesis titulada: “*Diseño de un robot móvil tipo oruga para exploración en terrenos irregulares*”, de la Escuela Superior Politécnica de Chimborazo de Ecuador, en su conclusión indica que: “El diseño e implementación del robot móvil tipo oruga para exploración en terrenos irregulares cumple con todas las pruebas de funcionamiento que demanda este tipo de dispositivos tecnológicos como: el desplazamiento en terrenos irregulares en diferentes superficies demostrando estabilidad” (p.91).

La tesis se desarrolla en el diseño de un robot de desplazamiento tipo oruga para la exploración de terrenos irregulares haciendo uso de múltiples sistemas de control para lograr mayor estabilidad del robot. Este cuenta con una tracción tipo oruga, comunicación por radiofrecuencia y una base de teleoperación para la visualización en tiempo real mediante monitores. Además, cuenta con un brazo robot programable de cuatro grados de libertad (DOF, de sus siglas en inglés) que le permite manipular objetos. Dicho prototipo demostró un nivel de eficiencia muy alto de aceptabilidad tras pruebas realizadas a la tracción y comunicación. La

investigación concluye afirmando que los robots con transmisión tipo oruga son superiores a los de tracción por llantas, ya que el primero cuenta con una mejor adaptabilidad en superficies irregulares o no afirmados, lo cual lo convierte en un sistema confiable permitiéndole una mejor maniobrabilidad al momento de desplazarse.

## **1.5. Hipótesis**

### **1.5.1. Hipótesis general**

La implementación de un robot explorador con realidad virtual incrementará la seguridad de la Institución Educativa 7213 Peruano Japonés en el distrito de Villa el Salvador – 2020.

### **1.5.2. Hipótesis específicas**

- Al incrementar la posibilidad de detección temprana de un intento de robo se podrá evitar que dicho acto delincuencia se perpetre.
- Al aumentar la eficiencia de patrullaje con el uso de realidad virtual y sensores de movimiento inalámbricos se podrá identificar un atentado de robo tempranamente.
- Al implementar un robot explorador con realidad virtual integrada se evita poner en riesgo la salud e integridad del personal de seguridad frente a un atentado.

## CAPÍTULO II. METODOLOGÍA

### 2.1. Tipo de investigación

#### 2.1.1. Según el propósito

La investigación es aplicada, según Murillo señala que este tipo de investigación es nombrada también como “investigación práctica o empírica” (Vargas,2009). Este estudio es caracterizado por utilizar el conocimiento conseguido, el cual a su vez se complementa de nuevo conocimiento, respondiendo al porqué de las cosas y generando nuevo conocimiento práctico para resolver problemas sociales o industriales a partir del conocimiento teórico.

Para esta investigación se hará uso del conocimiento consolidado durante la carrera. Así como también, del conocimiento que se adquirirá en el transcurso de la elaboración e implementación de este proyecto.

Finalmente, se tendrán en cuenta las observaciones y consejos brindados por compañeros y profesores de la universidad para así llegar al objetivo deseado.

### **2.1.2. Según el diseño de investigación**

La Investigación no experimental muestra el reconocimiento de la realidad tal como es presentada en un intervalo de tiempo y en el lugar donde ocurre el hecho. Aquí se observa y registra, o se pregunta y se registra. Describe el fenómeno sin introducir modificaciones: tal cual. Las preguntas de rigor son: ¿Qué es?, ¿Cómo es?, ¿Dónde está?, ¿Cuándo ocurre?, ¿Cuántos individuos o casos se observan?, ¿Cuáles se observan? (Rojas, 2015, p. 7).

En esta investigación se exhibirá la realidad problemática del centro educacional, el cual se busca reducir, mediante la implementación del proyecto, al finalizar con la investigación. A demás, se harán una serie de preguntas dentro del colegio, tales como: ¿Qué es?, ¿Cómo es?, ¿Dónde está?, ¿Cuándo ocurre?, entre otras, que ayude a tener una lluvia de ideas y posibilidades de mejora, para así encontrar una solución óptima y económica con respecto a la seguridad actual que presenta, buscando reducir las posibilidades de algún robo.

### **2.1.3. Línea de investigación**

La investigación de la carrera a la cual se adscribe el presente proyecto es la robótica. Según Ollero (2001), la robótica en nuestro siglo ha estado fuertemente influida por el progreso tecnológico. De esta forma se pasa de máquinas que tienen como objetivo exclusivo la amplificación de la potencia muscular del hombre, sustituyéndolo en su

trabajo físico, a maquinas o instrumentos que son también capaces de procesar información, complementando, o incluso sustituyendo, al hombre en algunas actividades intelectuales y físicas (p.1).

La robótica juntamente con la informática ha ido evolucionando al paso de los años con nuevos dispositivos tecnológicos, que ayudan al hombre en sus tareas diarias. En este proyecto, se busca reemplazar parcialmente al hombre y equiparlo con nuevas herramientas para un mejor desempeño laboral y evitar el grado de exposición al peligro que se tiene como parte de su trabajo.

#### **2.1.4. Tema / Eje temático**

Este proyecto, está basado en la implementación de un robot explorador con realidad virtual para incrementar la seguridad del centro educacional Peruano Japonés en Villa el Salvador – 2020. De esta manera, se busca reducir o eliminar atentados de robos con completa seguridad y sin arriesgar la integridad física del personal a cargo. La realidad virtual permite al usuario una completa inmersión del ambiente. Esta característica puede entregar al usuario la misma sensación de un entorno real sin la necesidad de estar presente físicamente. Es por esta razón, que la realidad virtual se hace de gran importancia en la integración de un robot explorador, para así mejorar la tarea de patrullaje por parte del usuario, además de brindarle nuevas herramientas para la detección de incidentes. En resumen, con la fusión de conocimientos de mecánica, electrónica e informática se buscará lograr los objetivos trazados para este proyecto.

### **2.1.5. Institución donde se desarrollará el proyecto.**

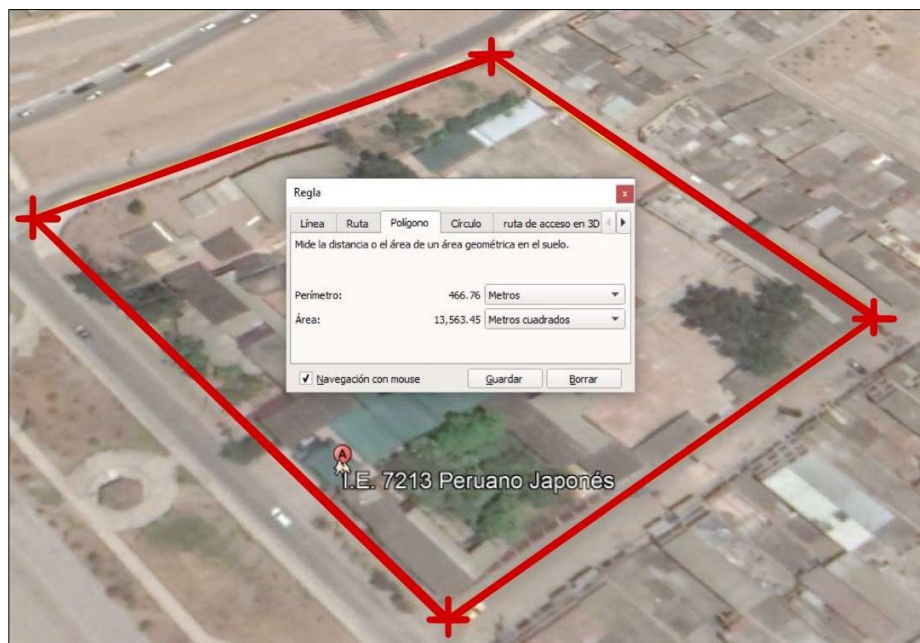
Colegio Peruano Japonés – 7213

Este centro educacional fue fundado en el año 1989 en el cruce de las avenidas 200 millas y Pachacútec y construida gracias a donaciones de la asociación APEKAI en Villa el Salvador. Dicho recinto busca constantemente la excelencia educacional, cuenta con laboratorios de cómputo y robótica, talleres de carpintería, talleres de costura, laboratorio de química, biblioteca y salón de instrumentos musicales. La mayor parte de equipos para la formación de los talleres se consiguieron gracias a aportes de los padres de familia y ONG's. Cuenta con terreno aproximadamente de 12 805 m<sup>2</sup> y lamentablemente, en varias oportunidades, ha sido víctima de robos durante la madrugada, ya que por poseer escaso equipamiento de seguridad y ser de territorio amplio, lo convierte en una localidad difícil de patrullar para un solo vigilante y no muy difícil de vulnerar para los delincuentes que terminan llevándose computadoras, maquinaria de los talleres e instrumentos musicales de la banda que difícilmente les costó conseguir.



*Figura 4.* Institución Educativa N° 7213 Peruano Japonés.

Fuente: I.E 7213



*Figura 5.* Perímetro Institución Educativa 7213 Peruano Japonés

Fuente: Google Earth Pro

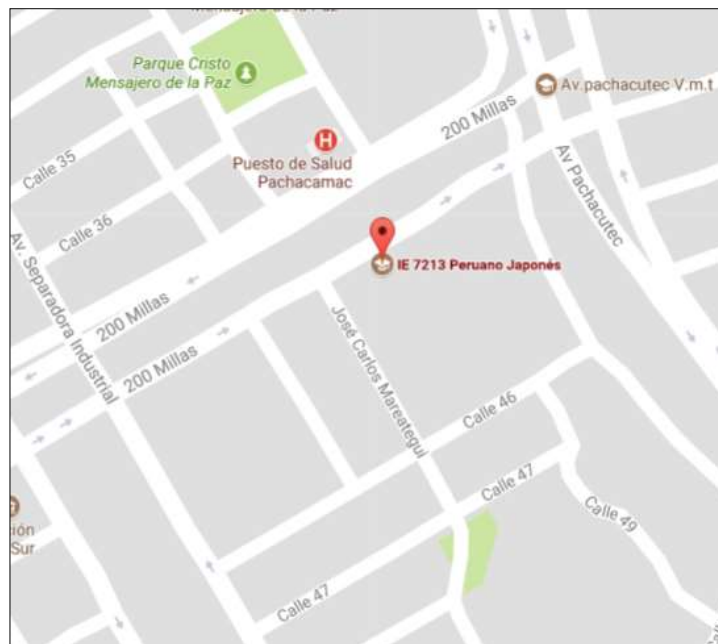


En la Figura 8 se observa que por ser un terrero grande es de difícil patrullaje para un solo encargado de seguridad.

### 2.1.6. Distrito, Provincia, Región.

Institución Educacional Peruano Japonés

Distrito de Villa el Salvador, en la provincia de Lima, departamento de Lima.



*Figura 6.* Croquis del I.E. Peruano Japonés 7213.

Fuente: Google Maps

## 2.2. Materiales, instrumentos y métodos

### 2.2.1. Materiales

En esta investigación de la implementación de un robot explorador con realidad virtual para incrementar la seguridad se utilizaron:

Para la parte mecánica, se seleccionaron 2 motores DC de 12V con sus respectivos motoredutores, utilizados para obtener torque y velocidad necesarios en el sistema de tracción tipo oruga; también planchas de aluminio y acrílico de 2mm para la fabricación de las piezas que conforman el chasis. Además, se dio uso de la técnica de impresión 3d para el modelado de piezas mecánicas diseñadas en inventor para el manejo de la cámara de 2 grados de libertad, mediante 2 servomotores, que replicará los movimientos del cuello del operario.

Para la parte electrónica, se da uso de la plataforma de IoT NodeMCU. El cual será el encargado de gestionar y controlar las señales recibidas por los sensores y actuadores. Además de diversos elementos electrónicos como condensadores, regulador de voltaje, driver L298, diodos y un banco de baterías para el control de la parte de potencia de los motores y la alimentación del controlador. También el uso de un giroscopio modelo MPU6050 para el monitoreo de posición de la cabeza del operario. Por último, un Joystick que servirá como control remoto para el desplazamiento del robot.

Para la parte de programación el IDE de PLATFORMIO capaz de comunicarse con el NodeMCU y la integración de la compilación de la programación en C++.

### **2.2.2. Instrumentos**

Para lograr replicar los movimientos del cuello del operario se necesitó el uso del Giroscopio MPU6050. El cual se encuentra instalado dentro del visor de realidad virtual y va reportando su ubicación en el espacio al servidor en la nube. Es allí donde recoge el controlador del robot los parámetros del giroscopio y del joystick de desplazamiento.

### **2.2.3. Métodos**

El robot explorador con realidad virtual enfocado en incrementar la seguridad finge de suplantador físico del operario más siempre debe de contar con su asistencia. Por tal motivo, los sensores que irán montados en el visor de realidad virtual como en el control remoto, ambas herramientas del operario de seguridad serán escaneados con un refresco de 150ms y replicados en los actuadores instalados en el robot explorador.

## 2.3. Desarrollo de la aplicación

### 2.3.1. Diseño mecánico del robot explorador

El diseño se encuentra inspirado en un modelo Skid Steer con tracción de oruga trapezoidal similar a la de un tanque de guerra. Se transmite el torque de las ruedas posteriores a las delanteras mediante el sistema de cadena de oruga que se enganchan a los engranes dentados de la rueda de transmisión.

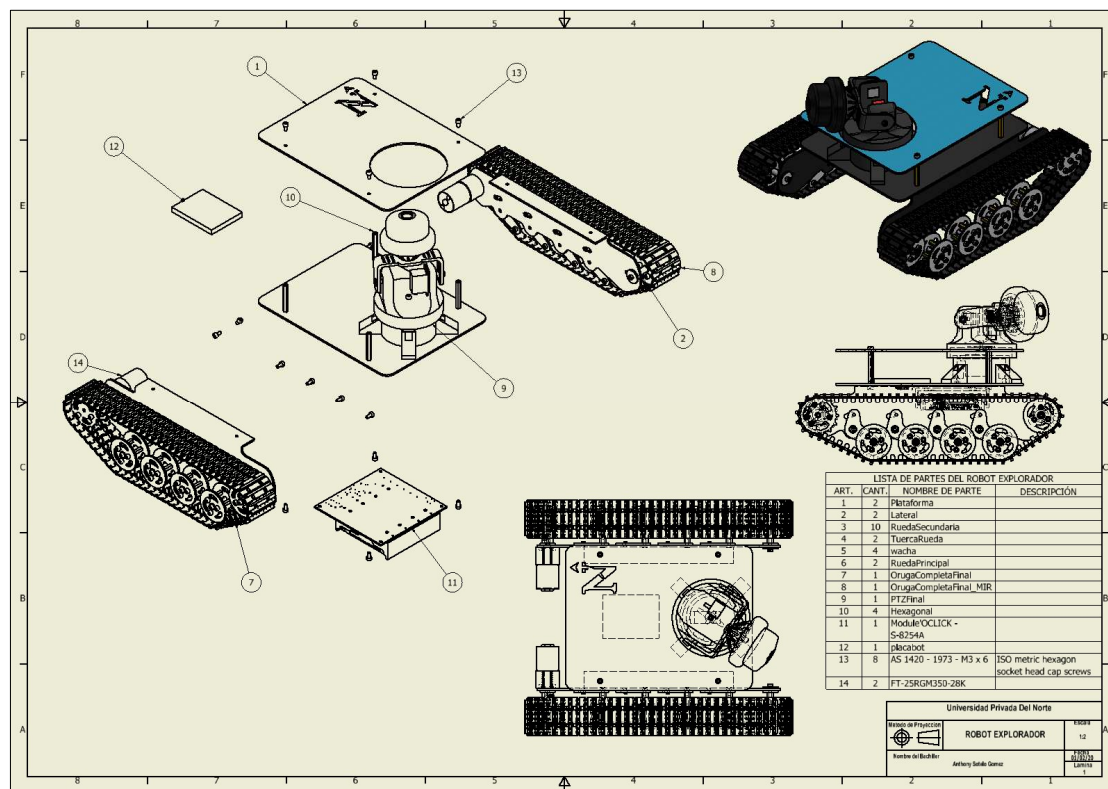
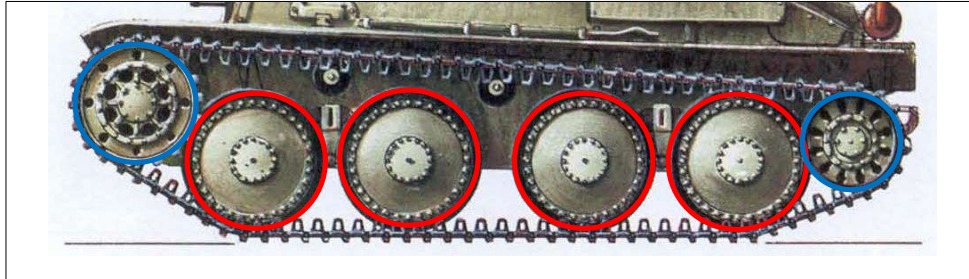


Figura 7. Plano en explosión isométrico del robot explorador.

Los mecanismos fueron diseñados en el software Autodesk Inventor y fabricados en diversos materiales como el Aluminio, Acrílico y plástico ABS. La información completa de los planos desarrollados se encuentra en los Anexos 1 al 17. Fuente: Elaboración Propia.

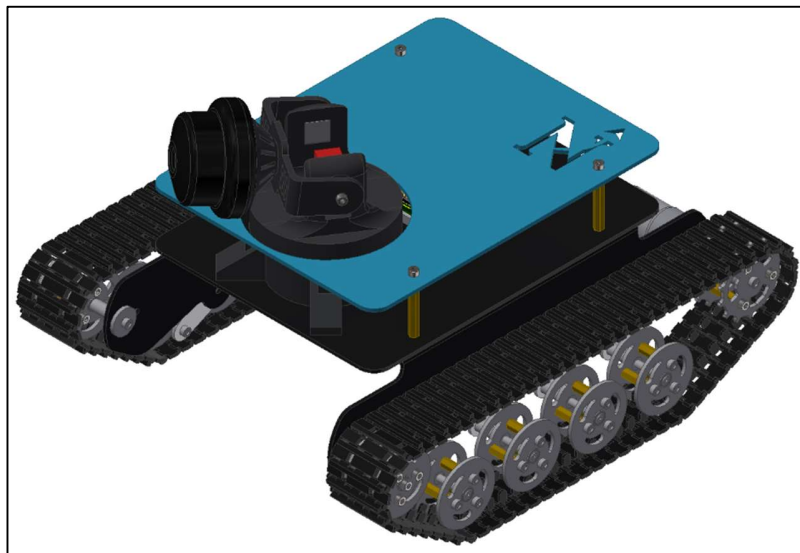
Las ruedas intermedias son las encargadas de amortiguar las irregularidades del suelo y dar estabilidad al robot al desplazamiento.



*Figura 8.* En azul las ruedas de transmisión y en rojo las ruedas intermedias.

Fuente: Elaboración Propia.

El software facilitó el diseño de cada una de las piezas y su integración con los dispositivos electrónicos montados en él. Cada pieza mecánica fue diseñada gracias al apoyo que te entrega el software al poder generar nuevos planos sobre las piezas prediseñadas con anterioridad, y así previsualizar el encaje entre piezas.



*Figura 9.* Isométrico de robot explorador

Fuente: Elaboración Propia.

La estructura del robot explorador fue diseñada en la plataforma Autodesk Inventor Professional 2018 tomando en cuenta el ambiente y el tipo de suelo a explorar, además de basarse en modelos ya existentes que sirven para otro tipo de aplicaciones. La locomoción skid-steer es ampliamente usada en muchos vehículos terrestres como cargadores frontales, maquinaria agrícola, de minería y otros. A continuación, se explica y justifica físicamente el diseño.

- Dimensionamiento de las ruedas.
- Dimensionamiento de los motores DC.
- Dimensionamiento de los Servomotores.
- Eficiencia – Potencia mecánica vs eléctrica.

#### 1. Dimensionamiento de las ruedas.

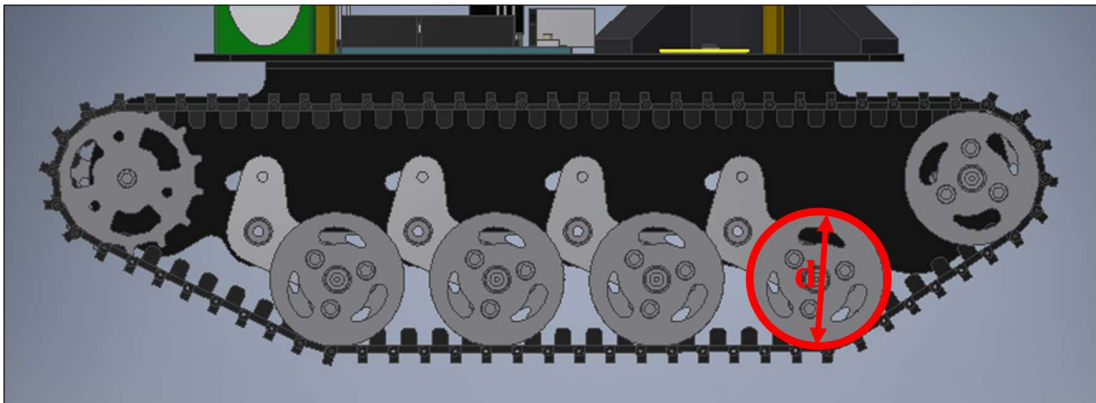


Figura 10. Dimensionamiento de ruedas de acuerdo con el diámetro.

Fuente: Elaboración Propia.

Como se desea reemplazar al operador se toma como base la velocidad promedio que tiene

una persona al caminar que es:  $Vt = 2 \frac{km}{h} \dots (1)$

$$\Rightarrow Vt = w * r \dots (2)$$

Se decide elegir un motor de 300.0 rpm a 12 V.

$$w = 300 \frac{\text{rev}}{\text{min}} * \frac{1 \text{ min}}{60 \text{ s}} * \frac{2\pi \text{ rad}}{1 \text{ rev}} = 31.42 \text{ rad/s}$$

De (1):

$$Vt = 2 \frac{\text{km}}{\text{h}} * \frac{1000 \text{ m}}{1 \text{ km}} * \frac{1 \text{ h}}{3600 \text{ s}} = 0.56 \text{ m/s}$$

De (2):

$$r = \frac{Vt}{w} = \frac{0.56}{31.41} = 0.0185 \text{ m} = 19 \text{ mm}$$

$$\Rightarrow d = 38 \text{ mm}$$

## 2. Dimensionamiento de los motores DC

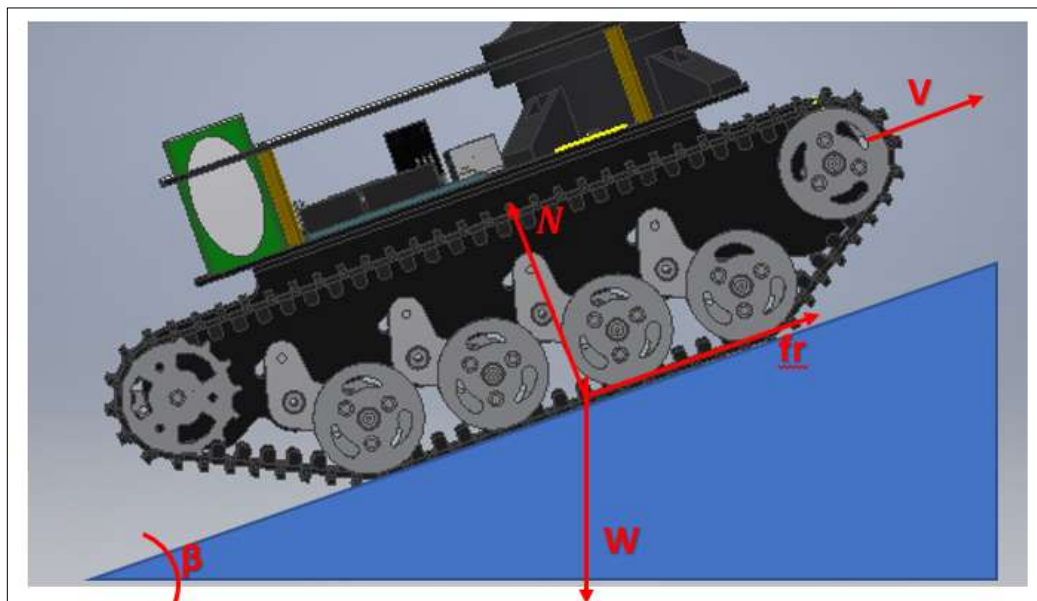
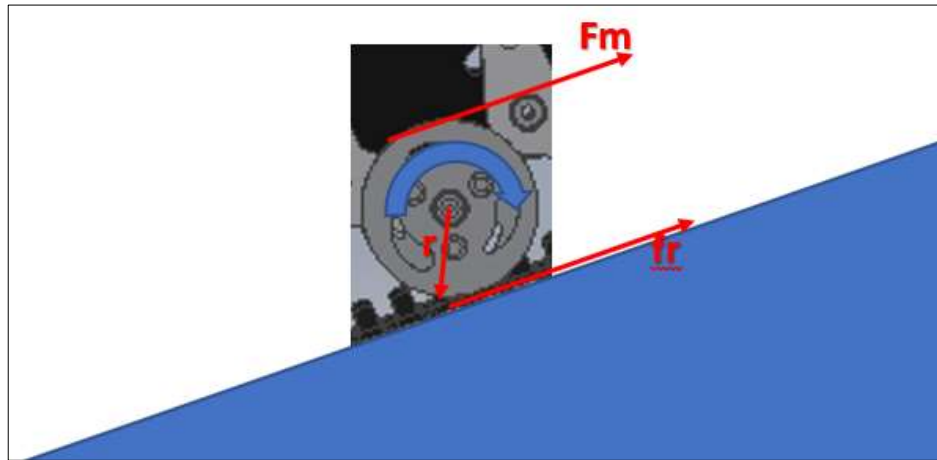


Figura 11. Diagrama de cuerpo libre en pendiente.

Fuente: Elaboración Propia.

Para facilitar el cálculo se analiza puntualmente lo que sucede en una rueda.



*Figura 12. Análisis de Fuerzas en una rueda.*

Fuente: Elaboración Propia.

*Fm: Fuerza motriz que entrega el motor.*

*Fr: Fuerza de fricción (contraria al giro de la rueda).*

Como se sabe que: “un torque neto sobre un objeto que tiene un punto de rotación fijo ocasiona una aceleración angular sobre él, la cual es inversamente proporcional a cierta cantidad de Inercia” (Sears, 2009, p.316).

Esto es análogo a la conocida 2da ley de Newton para cuerpos rotacionales. Entonces:

$$\sum \tau = I * \alpha \dots (a)$$

Donde:

$\sum t$  : Sumatoria de torques

$I$  : Inercia del cuerpo

$\alpha$ : aceleración angular



De la figura 28 aplicando de la formula “a” se obtiene:

$$Fm * r - Ff * r = I * \alpha$$

Como el objetivo es hallar el torque del motor, se sabe que:

$$Tm = Fm * r$$

Y se despeja

$$Tm = I * \alpha + Ff * r$$

Se sabe:

La inercia para un cuerpo circular

$$I = \frac{1}{2} M * R^2$$

La aceleración angular por condiciones iniciales y como se desea que alcance su velocidad máxima en 2 segundos se toma 15 rad/s<sup>2</sup> y el radio propio del cuerpo solido que es 19 mm por los cálculos.

Para obtener finalmente el torque del motor se necesita saber cuánto es la Fuerza de fricción, para lo cual se sabe que la fuerza de Fricción es:

$$Fr = N * \mu$$

Donde:

*N: Fuerza normal perpendicular y opuesta al plano*

*μ: coeficiente de fricción*

Para que robot explorador pueda desplazarse sin ningún problema se toma el peor de los casos y se elige un coeficiente de fricción alto de la siguiente figura:

Materiales	Coefficiente de fricción estática, $\mu_s$	Coefficiente de fricción cinética, $\mu_k$
Acero sobre acero	0.74	0.57
Aluminio sobre acero	0.61	0.47
Cobre sobre acero	0.53	0.36
Latón sobre acero	0.51	0.44
Zinc sobre hierro colado	0.85	0.21
Cobre sobre hierro colado	1.05	0.29
Vidrio sobre vidrio	0.94	0.40
Cobre sobre vidrio	0.68	0.53
Teflón sobre teflón	0.04	0.04
Teflón sobre acero	0.04	0.04
Hule sobre concreto (seco)	1.0	0.8
Hule en concreto (húmedo)	0.30	0.25

Figura 13. Coeficientes de fricción.

Fuente: Sears, Física Universitaria.

La Normal se saca con un ángulo de  $25^\circ$  que es lo máximo en pendiente dentro del complejo educativo. Al descomponer el peso del robot explorador se obtiene que la componente y del peso es igual a la Normal.

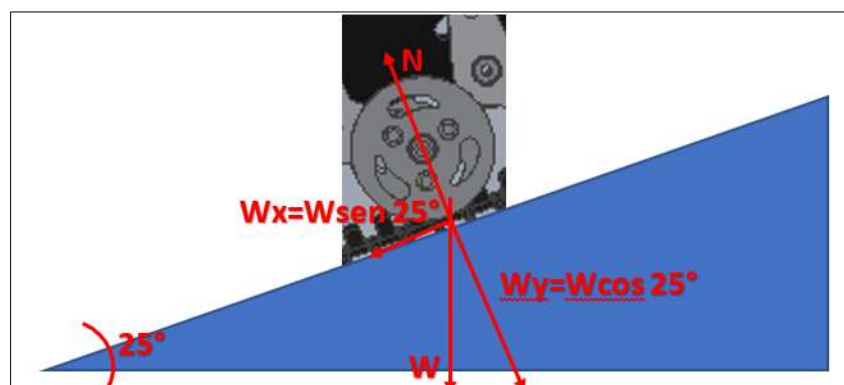


Figura 14. Descomposición del peso.

Fuente: Elaboración Propia.

Entonces:

$$N = W * \cos(25^\circ)$$

$$N = m * g * \cos(25^\circ)$$

$$N = 3 * 9.81 * \cos(25^\circ)$$

$$N = 26.7 \text{ N}$$

Reemplazando en la ecuación de  $F_r$  y multiplicándolo por un factor de seguridad de 2:

$$F_r = 26.7 * 1 * 2 = 53.3 \text{ N}$$

Finalmente, reemplazando se halla el torque necesario para desplazar el robot explorador.

$$T_m = I * \alpha + F_r * r$$

$$T_m = \frac{1}{2}(3)(0.019)^2 * 15 + 53.3(0.019)$$

$$T_m = 1.02 \text{ N.m}$$

Los cuales se dividen en 2 motores de 0.51 N.m.

### 3. Dimensionamiento de los Servomotores.

El mecanismo del robot cuenta con un mini brazo robot de 2 ejes rotatorios. En su plataforma superior, se encuentra posada la cámara IP que va registrando en tiempo real el ambiente que lo rodea. Este arreglo de servomotores con piezas mecánicas le permite a la cámara tener un ángulo de visión de 180 grados tanto vertical como horizontal e imita el movimiento natural del cuello.

Lo cual es fundamental para una inmersión absoluta en el ambiente a monitorear. A la vez que cuenta con sensores infrarrojos pasivos para detectar movimiento sin ser detectado.

A continuación, se halla el torque necesario para realizar dicho objetivo.

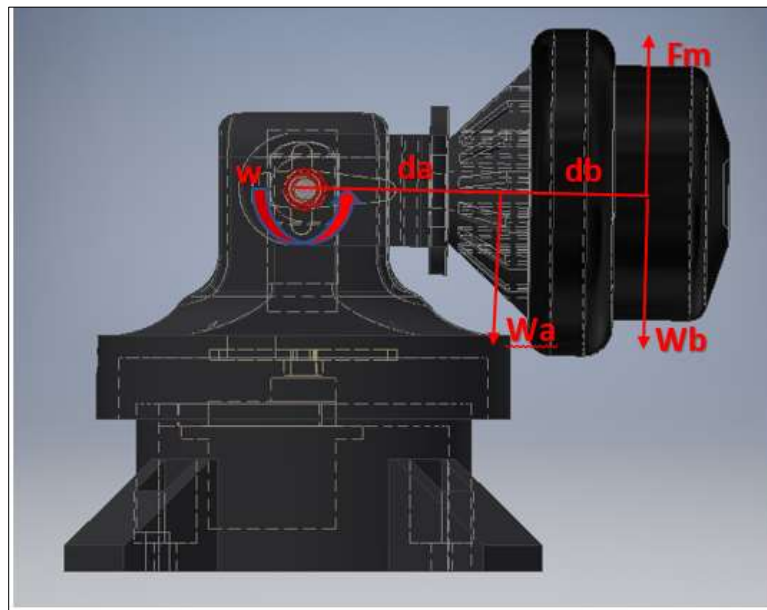


Figura 15. Sistema roll y pitch.

Fuente: Elaboración Propia.

Por segunda Ley del movimiento rotacional:

$$\sum \tau = I * \alpha \dots (x)$$

Sabiendo que la Inercia es:

$$I = \frac{1}{3} M * a^2$$

Siendo “a” la distancia perpendicular del punto final del brazo hasta el eje de rotación.

De la figura 19 se reemplaza en la ecuación x:

$$Tm - Wa * da - Wb(da + db) = \frac{1}{3}M(da + db)^2 * \alpha$$

Despejando Torque del Servomotor:

$$Tm = \frac{1}{3}M(da + db)^2 * \alpha + Wa * da + Wb(da + db)$$

$$Tm = \frac{1}{3}(0.25)(0.05)^2 * (6) + 0.1(9.81)(0.03) + 0.15(9.81)(0.05)$$

$$Tm = 0.104 \text{ N.m}$$

Entonces el Servomotor que más se ajusta es el MG90S que entrega un torque de 0.2157 N.m. y porque las fuerzas de torsión son perpendiculares al eje de la base se decide utilizar el mismo tipo de servomotor para la rotación en el plano XY.

Finalmente se valida por medio de la simulación realizada en Inventor Professional 2018, que muestra, aunque mínimo, el estrés concentrado en los ejes de giro (Ver Anexo 20).

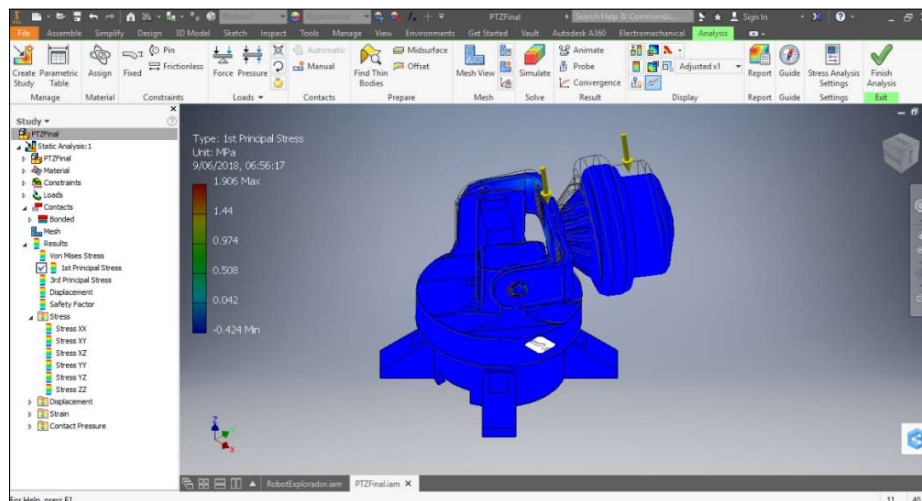


Figura 16. Stress Analysis de Inventor.

Fuente: Elaboración Propia.

### 2.3.2. Diseño del Sistema electrónico

Los dispositivos electrónicos que conforman del diseño del robot explorador se basan en el Kit de desarrollado IoT NodeMCU-V3 que cuenta con el chip Wifi ESP8266. Este chip, a su vez, cuenta con un microcontrolador Tensilica Xtensa LX106 que es el encargado de manejar el protocolo de comunicación inalámbrico basado en el estándar IEEE 802.11b/g/n. Además de contar con un gran número de entradas y salidas digitales de propósito general (GPIO), 16 en concreto y 1 entrada analógica de resolución de 10 bits. Asimismo, posee el puerto de comunicación serial I2C y salidas PWM que son importantes en este desarrollo.



*Figura 17.* Controlador NodeMCU-V3

Fuente: Nodemcu.com.

Tabla 5

*Especificaciones Técnicas del controlador NodeMCU-V3.*

NodeMCU-V3	
Microcontrolador	Tensilica 32-bit RISC CPU Xtensa LX106
Voltaje de Operación	3.3 V
Voltaje de Entrada	7-12 V
Pines Digitales I/O	16
Pin Analógico	1
UARTs	1
SPIs	1
I2Cs	1
Flash Memory	4MB
SRAM	64 KB
Clock Speed	80 MHz
Protocolos de Red	IPv4, TCP/UDP/HTTP/FTP
Rango de Frecuencia	2.4GHz -2.5GHz
Seguridad/Encriptación	WPA/WPA2/THIP/AES

Fuente: Elaboración Propia.

El diseño electrónico del proyecto de tesis se encuentra dividido en 3 dispositivos que funcionan conjuntamente mediante el uso del NodeMCU y la red de redes conocida como internet. Estos son los siguientes;

- Diseño electrónico del sistema locomotor del robot explorador.
- Diseño electrónico del Visor de Realidad Virtual.
- Diseño electrónico de la palanca de mando de desplazamiento.

### 2.3.2.1. Diseño electrónico – Robot explorador

De acuerdo con el diseño mecánico, se necesitan 2 motores que superen los 0.51 N.m y entreguen una velocidad angular de 300 rpm/min para igualar la velocidad de desplazamiento de una persona al caminar. Es por ello, que se elige el Motor XD-37GB520 que entrega un torque máximo de 0.981 N.m.

Tabla 6.

*Especificaciones Técnicas - Motor XD-37GB520.*

Especificaciones técnicas – Motor XD-37GB520	
Voltaje	12V
Velocidad sin carga	300 rpm/min
Velocidad de carga	9 rpm/min
Corriente sin carga	0.17 A
Corriente de carga	0.59 A
Corriente máxima	2.19 A
Torque nominal	0.196 N.m.
Torque máximo	0.981 N.m.
Relación de reducción	12.5
Potencia	5 W
Peso	195 mg

Fuente: Elaboración Propia.





*Figura 18.* Motor DC con motor reductor XD-37GB520

Fuente: Gearbest.com.

De acuerdo a lo anterior, cada motor consumirá 0.59 A. En consecuencia, se elige el chip integrado L298 capaz de energizar un motor por debajo de los 2A. Para poder integrar y utilizar este driver en la PCB base es necesario entender su funcionamiento.

- Si  $A=0$ ,  $B=1$  y  $C=1$ .

La salida del AND  $\overline{A}C$  será un 1 lógico el cual se traduce en un nivel de voltaje igual al de la referencia. Este excitará la base del transistor y unirá el colector con el emisor. Lo mismo sucede con el AND  $BC$  obteniendo como resultados:

Tabla 7

*Tabla de verdad para giro horario*

A	B	C	OUT1	OUT2
0	1	1	0V	12V

Fuente: Elaboración Propia.

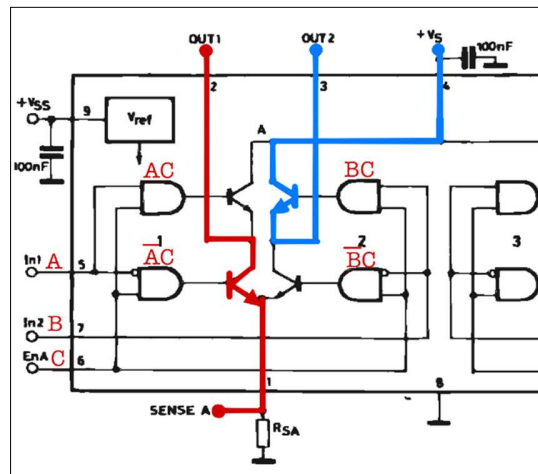


Figura 19. Diagrama de bloques de datasheet L298.

Seguimiento de respuesta del L298 de acuerdo con los estados de sus pines de entrada para un giro horario.  
 Fuente: Elaboración Propia.

- Si  $A=1$ ,  $B=0$  y  $C=1$ .

La salida del AND  $AC$  será un 1 lógico el cual se traduce en un nivel voltaje igual al de la referencia. Este excitará la base del transistor y unirá el colector con el emisor. Lo mismo sucede con el AND  $\overline{B}C$  obteniendo como resultados:

Tabla 8

Tabla de verdad para giro antihorario

A	B	C	OUT1	OUT2
1	0	1	12V	0V

Fuente: Elaboración Propia.

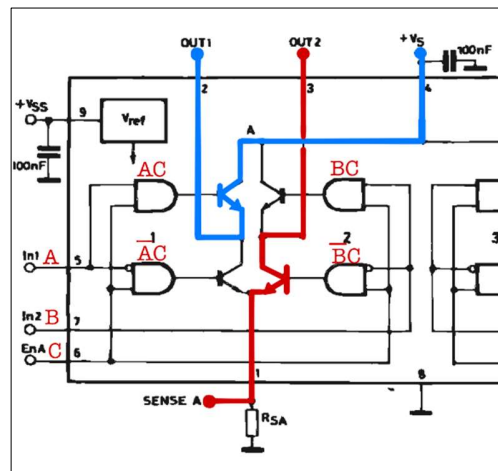


Figura 20. Diagrama de bloques de datasheet L298.

Seguimiento de respuesta del L298 de acuerdo con los estados de sus pines de entrada para un giro antihorario. Fuente: Elaboración Propia.

Una vez obtenido el resultado de ambas tablas de verdad, se puede pasar a la programación del controlador que replicará las señales de entrada para el giro horario y antihorario de los motores, el cual se explica en el desarrollo de la aplicación de programación.

Por otro lado, de acuerdo con el diseño mecánico se elige los servomotores TowerPro MG90S, ya que estos superan el torque necesario para realizar los movimientos PAN y TILT fundamentales para replicar los movimientos naturales del cuello.

Tabla 9

*Especificaciones Técnicas - Servomotor TowerPro MG90S*

Especificaciones técnicas – Servomotores TowerPro MG90S	
Peso	13.4g
Torque máximo	1.8kg/cm (4.8V); 2.2kg (6.6V)
Velocidad de operación	0.1s/60° (4.8V); 0.08s/60° (6.0V)
Voltaje de operación	4.8V – 6.6V
Ancho de banda muerta	5 $\mu$ s

Fuente: Elaboración Propia.



*Figura 21.* Servomotor TowerPro MG90S.

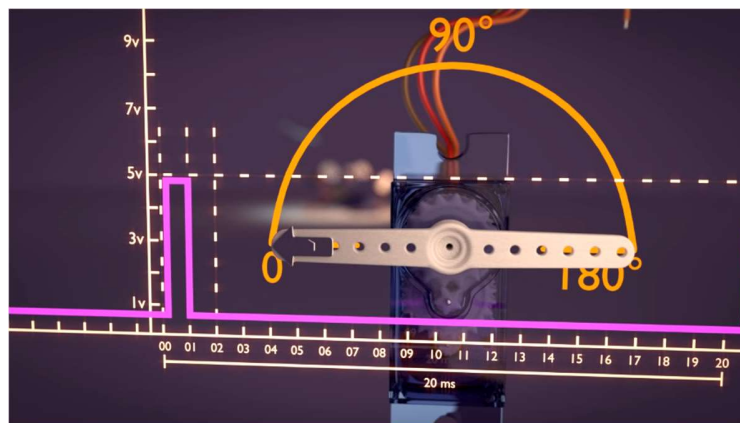
Este servomotor cuenta con engranes metálicos y un torque superior a su antecesor el SG90 TowerPro. Fuente: Gearbest.

Estos servomotores cuentan con una pequeña placa controladora en su interior que necesita una alimentación de 5 V y señal PWM que sirve de setpoint para el arreglo de control de lazo cerrado al interior, entre motor DC y potenciómetro. Estos últimos se encuentran unidos mediante engranes

lo cual permite controlar la posición exacta del servomotor que siempre sigue al setpoint ingresado por la entrada PWM.

Para el diseño electrónico se reservan salidas PWM del controlador central del robot. El periodo de la onda cuadrada es de 20 ms y de acuerdo como se valla modulando el ancho del pulso se obtendrán los ángulos requeridos entre 0 y 180°.

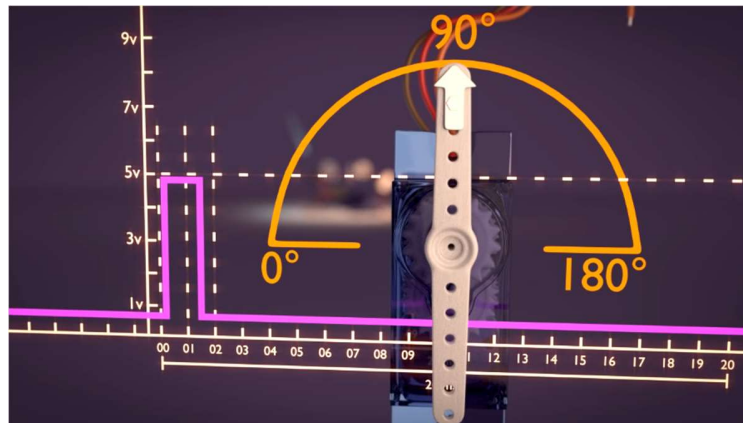
Si se genera un ancho de pulso de 1ms en un periodo de 20 ms se obtendrá un ángulo de 0°.



*Figura 22.* Ancho de pulso vs ángulo.

Para un ancho pulso de 1ms el servomotor se ubica en la posición angular de 0°. Fuente: Vt en linea.

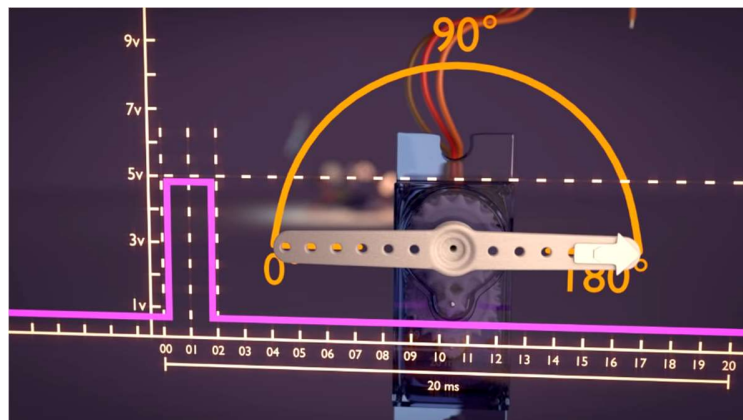
Si se genera un ancho de pulso de 1.5ms en un periodo de 20 ms se obtendrá un ángulo de 90°.



*Figura 23. Ancho de pulso vs ángulo.*

Para un ancho pulso de 1.5ms el servomotor se ubica en la posición angular de 90°. Fuente: Vt en línea.

Si se genera un ancho de pulso de 2 ms en un periodo de 20 ms se obtendrá un ángulo de 180°.



*Figura 24. Ancho de pulso vs ángulo*

Para un ancho pulso de 2ms el servomotor se ubica en la posición angular de 180°. Fuente: Vt en línea.

Es por ello, que se deberá modificar el ancho del pulso de entre 1ms a 2ms para alcanzar la ubicación exacta de la rotación de cuello reportado por el giroscopio ubicado en el visor de realidad virtual, el cual se explica en el desarrollo de la aplicación de programación.

Para controlar los actuadores montados en el robot explorador, se vio necesario diseñar previamente un PCB que integre todos los dispositivos electrónicos de manera ordenada y que su vez simplifique posteriormente las actualizaciones o modificaciones que se puedan hacer. Es por ello, que se utilizó el programa Proteus 8 Professional, el cual ayudó a la simulación y al diseño de este prototipo.

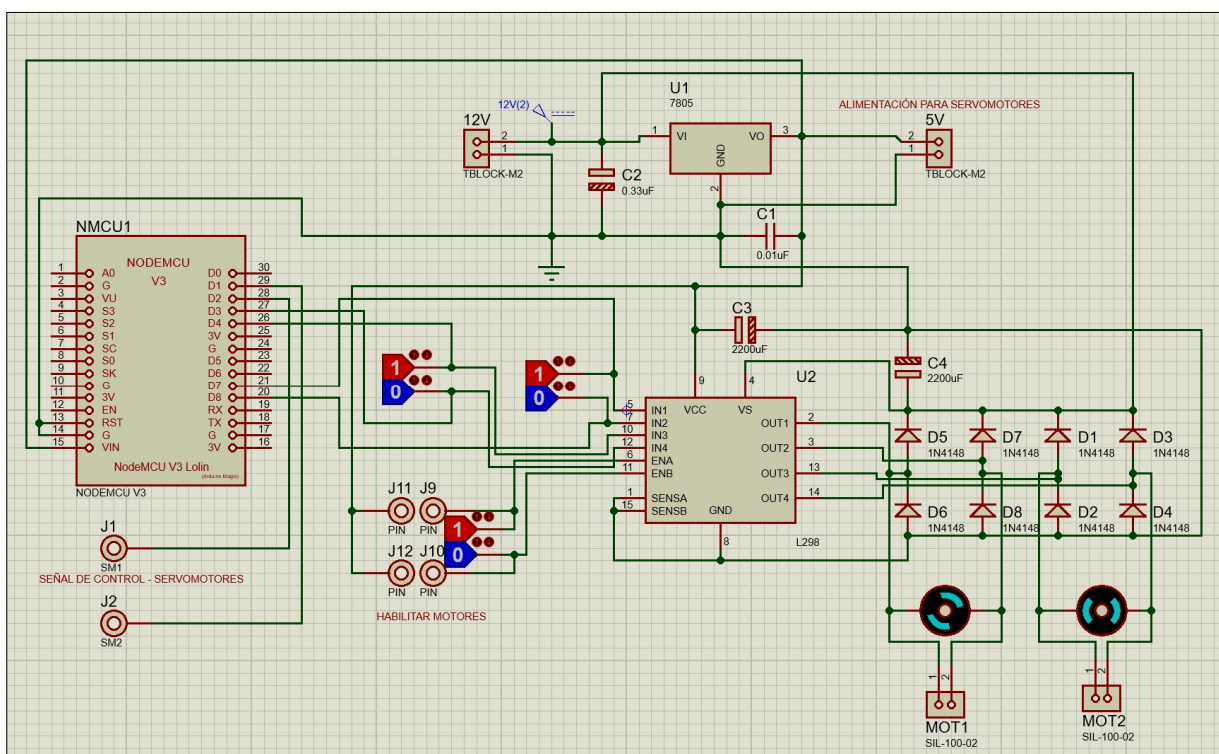


Figura 25. Esquemático electrónico del robot explorador.

El esquemático electrónico fue desarrollado y simulado en el software Proteus 8. Fuente: Propia.

Una vez comprobado el funcionamiento del sistema mediante la simulación se pasó a diseñar la placa de doble capa en el PCB Layout de Proteus para lograr así unas dimensiones menores y eliminar las interferencias entre pistas en una sola capa.

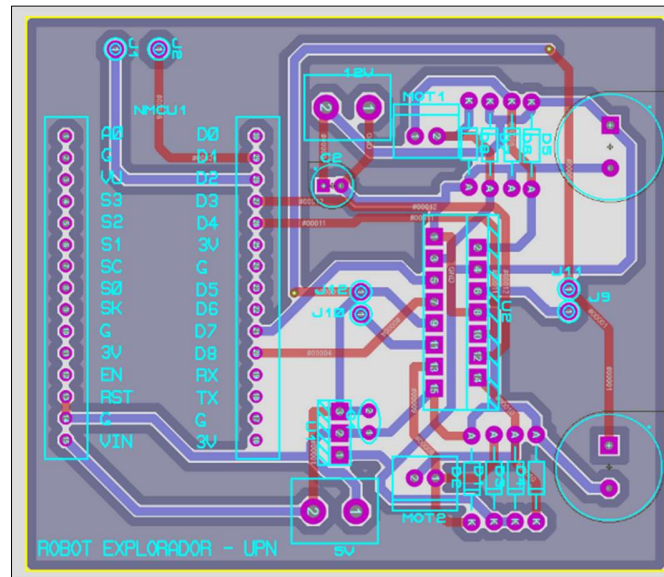


Figura 26. Diseño y ubicación de componentes electrónicos en PCB.

El software Proteus 8 tiene una extensión que sirve para el diseño de la PCB de acuerdo con el esquemático final del diseño. Fuente: Propia.

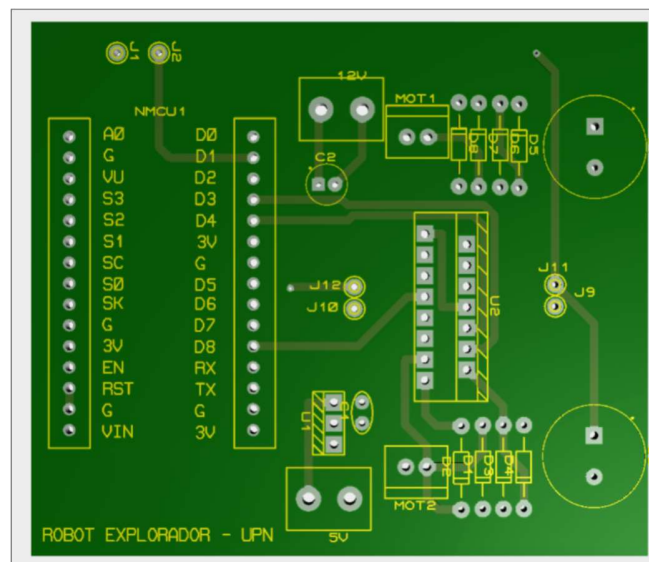
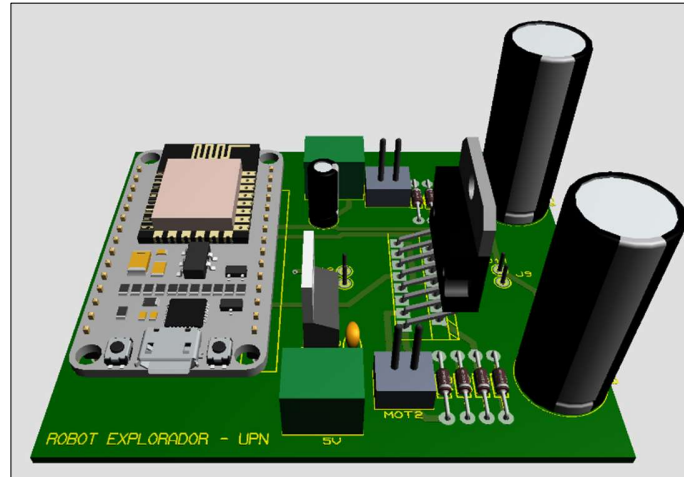


Figura 27. Vista final del PCB a fabricar en fibra de vidrio.

Fuente: Propia.

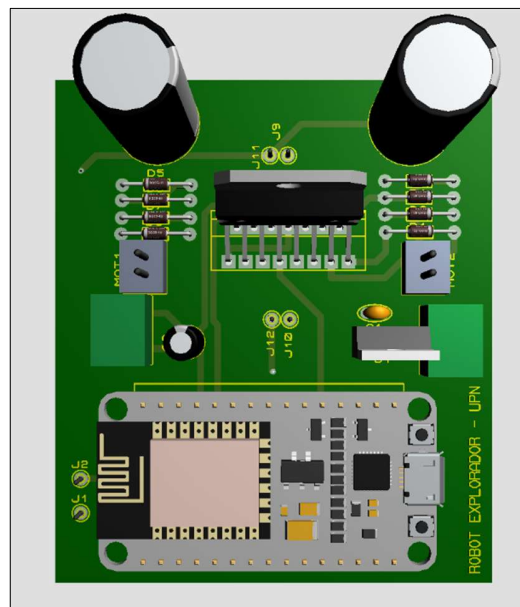


Una vez fabricado el PCB se pasan a soldar los componentes electrónicos por la parte inferior de la placa, obteniendo como resultado lo siguiente:



*Figura 28.* Vista final de PCB - Robot Explorador

El software Proteus 8 permite tener una vista 3d final con los componentes electrónicos soldados al PCB.  
Fuente: Propia.



*Figura 29.* Vista Superior de PCB - Robot Explorador

Fuente: Propia.

### 2.3.2.2. *Diseño electrónico – Lentes de realidad virtual*

El problema de los lentes de realidad virtual es que son dispositivos no inteligentes. Lo más resaltante es que cuentan con mecanismos que ajustan los lentes interiores para cubrir el desfase de imagen que le brinda normalmente un smartphone montado en él. Para lograr el propósito de que el robot explorador pueda replicar los movimientos del cuello del operario, se vio con la necesidad de integrar un dispositivo de monitoreo de rotación de la cabeza. Es por ello, que se decidió montar un giroscopio al interior de estos lentes.



Figura 30. Lentes de realidad Virtual - VRBOX

Fuente: Propia.

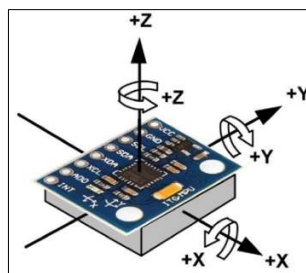
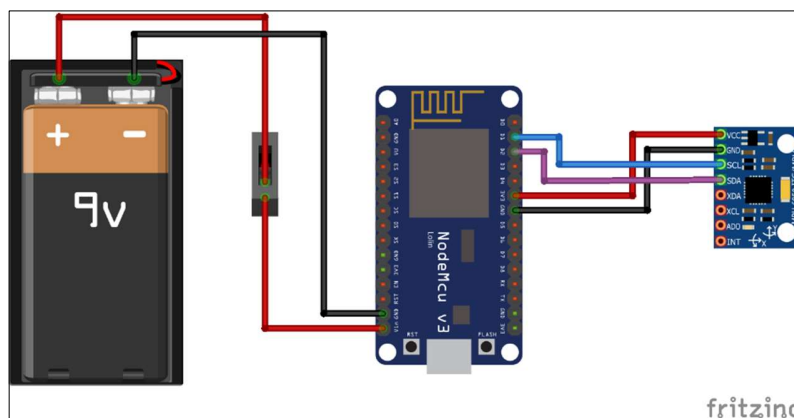


Figura 31. Giroscopio MPU6050

Fuente: Gearbest.

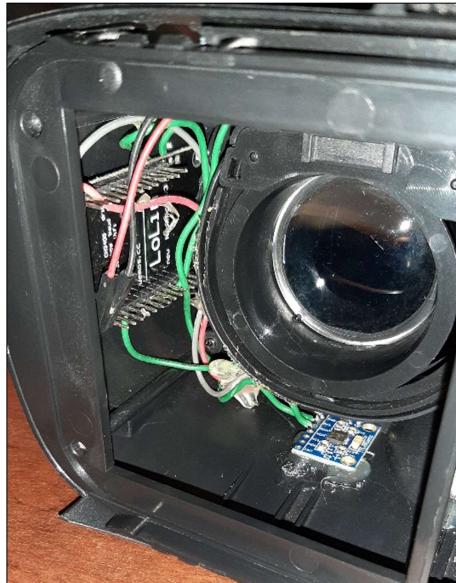
El MPU6050 es un IMU (Unidad de medición inercial) que posee 3 ejes para medir la tasa de cambio del desplazamiento angular por unidad de tiempo. Este módulo utiliza un MEMS, sistema microelectromecánico para medir la velocidad angular gracias al efecto Coriolis. Para el proyecto se ve la necesidad de establecer un punto de origen, ya que sabiendo el punto de inicio del giro y el tiempo que demora el cambio, se puede calcular su posición angular final.

Es síntesis, el módulo MPU6050 entregará la velocidad angular de los ejes x, y, z al controlador esclavo NodeMCU. Para ello, se deberán comunicar mediante protocolo I2C (Inter integrated circuits), a través de dos pines, el pin SCL (Serial Clock) y el SDA (Serial Data). Estos puertos son necesarios para la sincronización de envío y recepción de datos entre ambos controladores. Una vez recepcionada la velocidad angular se convertirá mediante un algoritmo de programación a posición angular actual en los 3 ejes.

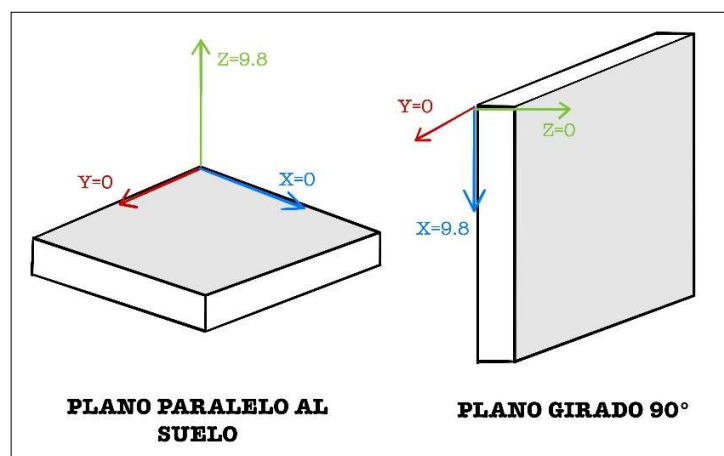


*Figura 32.* Esquema de conexiones de electrónica en lentes de Realidad Virtual.

Este esquema de conexiones fue realizado en el fritzing, software libre de automatización de diseño electrónico que ayuda a pasar prototipos a diseños finales. Fuente: Propia.



*Figura 33.* Ubicación de controlador NodeMCU y MPU6050 al interior del visor VR.  
Fuente: Propia.



*Figura 34.* MPU6050 con plano paralelo y plano perpendicular al suelo.  
Fuente: Propia.

En el caso del acelerómetro, el IMU mide constantemente la aceleración perpendicular al suelo que existe por la gravedad. Es decir, en el supuesto de que el módulo estuviese perfectamente alineado con el suelo. El eje z debería marcar  $9.81 \text{ m/s}^2$  y los ejes restantes 0. Es por ello, que una

vez obtenido la medición de los valores de aceleración de los tres ejes se pueden obtener los desplazamientos angulares mediante trigonometría.

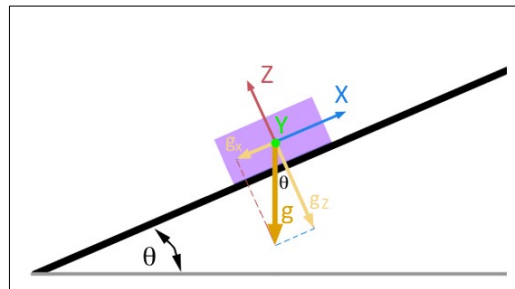


Figura 35. Diagrama para hallar el desplazamiento angular.

Fuente: Propia.

$$\theta = \text{atan}\left(\frac{A_x}{A_z}\right)$$

De forma similar para un espacio en 3 dimensiones:

$$\text{Angulo } Y = \text{atan}\left(\frac{x}{\sqrt{y^2 + z^2}}\right)$$

$$\text{Angulo } X = \text{atan}\left(\frac{y}{\sqrt{x^2 + z^2}}\right)$$

Ya que, el cálculo se realiza por la gravedad no es posible calcular el ángulo z. Sin embargo, para este proyecto solo son necesarios los ángulos X e Y.

Por el giroscopio, el IMU entrega las velocidades angulares en los 3 ejes en rad/s. Para obtener el desplazamiento angular final en todo momento es necesario conocer el ángulo inicial en el que empieza cada rotación dentro de un intervalo de tiempo.

$$\theta_f = \theta_0 + \omega * \Delta t$$

Donde:

$\theta_f =$  Posición angular final

$\theta_0 =$  Posición angular inicial

$\omega =$  Velocidad angular obtenida por el giroscópio

$\Delta t =$  Intervalo de tiempo en el que sucede el cambio

Una vez obtenidos los resultados de las posiciones angulares finales en los tres ejes mediante los 2 métodos, se opta por aplicar un filtro complementario que sirve para reducir los ruidos e imprecisiones. Este filtro sirve como un filtro pasa alto para los valores que entrega el giroscopio y uno pasa bajo para los del acelerómetro. De esta manera, se obtiene un posicionamiento angular más preciso.

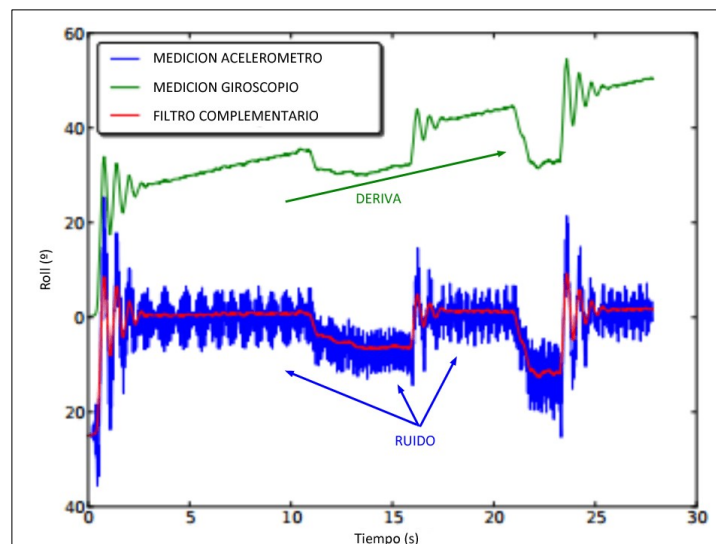


Figura 36. Mediciones obtenidas con el MPU6050 y aplicando el filtro complementario.

Fuente: Luisllamas.es.

$$\theta_{fc} = 0.98 * (\theta_{fc} + \theta_{giro} * \Delta t) + 0.02 * \theta_{acc}$$

Donde:

$\theta_{fc}$  = Posición angular final obtenida aplicando el filtro complementario

$\theta_{giro}$  = Posición angular obtenida por el metodo del giroscópio

$\theta_{acc}$  = Posición angular obtenida por el metodo del acelerómetro

$\Delta t$  = Intervalo de tiempo en el que sucede el cambio

La integración de las fórmulas mencionadas se realiza en el desarrollo de la aplicación – diseño de la programación 2.3.3.

### 2.3.2.3. Diseño electrónico – Joystick

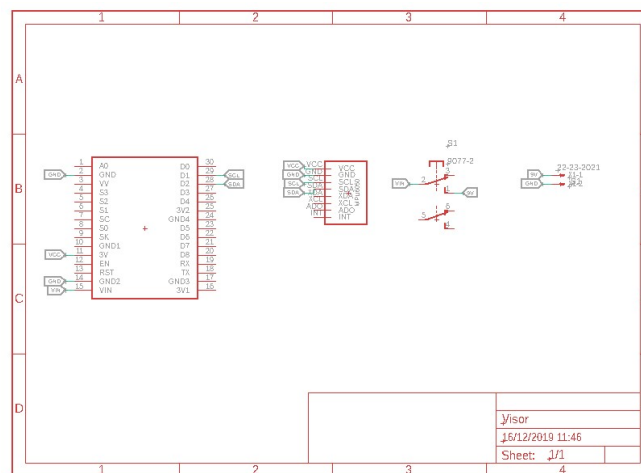


Figura 37. Diagrama electrónico de conexiones – Joystick.

Fuente: Propia.

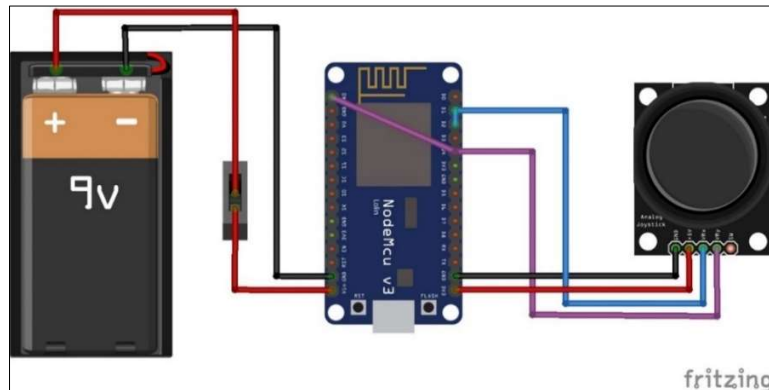


Figura 38. Esquema de conexiones de Joystick de operador.

Fuente: Propia.

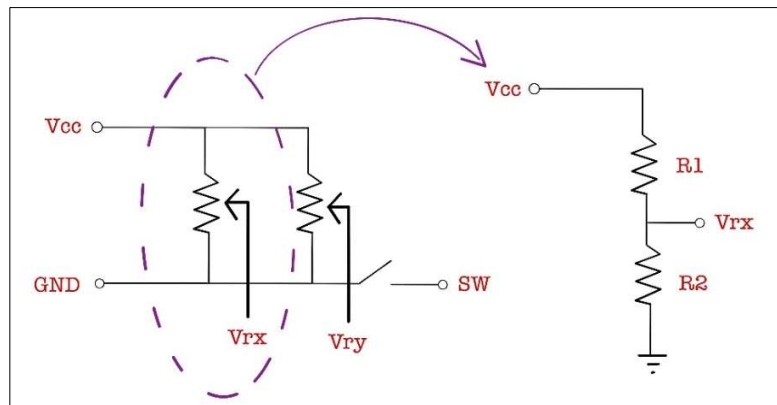


Figura 39. Divisor de Voltaje de los ejes "x" e "y".

Fuente: Elaboración propia.

Para hallar el voltaje de salida que lee el controlador se aplica la fórmula de división de voltaje:

Para la coordenada x: 
$$Vr_x = \frac{R_2}{R_1 + R_2} \times V_{CC}$$

Para la coordenada y: 
$$Vr_y = \frac{R_4}{R_3 + R_4} \times V_{CC}$$

El controlador posee un convertido Analógico/Digital de resolución de 10 bits lo cual permite tener 1024 combinaciones, valores del 0 al 1023. Cuando el Joystick se encuentre en estado de reposo



las resistencias serán iguales, entonces el voltaje medido teórico será de 2.5V. Sin embargo, este valor no será tan cierto en forma experimental, ya que los mecanismos propios del Joystick no están perfectamente regulados para que cuando se esté en reposo, la palanca se encuentre exactamente al medio. Es por ello, que mediante programación se define una zona de reposo para que cuando el joystick permanezca intacto el robot no empiece a desplazarse por sí solo.

Por otro lado, este módulo posee un pulsador en la palanca representado gráficamente mediante el contacto abierto. Este tendrá una configuración de resistencia pull-down para indicarle al controlador el cambio de estado y mandar a activar la sirena de alarma. Adicionalmente, se plantea agregar un mecanismo de defensa para el robot en caso corra peligro de ser robado.

De acuerdo con el estudio realizado por el Instituto Auditivo Salesa de España. Existe un nivel de ruido que es insoportable por el oído humano. Todo ruido que es igual o mayor a 120 dBs se encuentra en el nivel de umbral de dolor para el ser humano. Este repercute en el cuerpo mayormente en forma de dolor de cabeza y estrés.

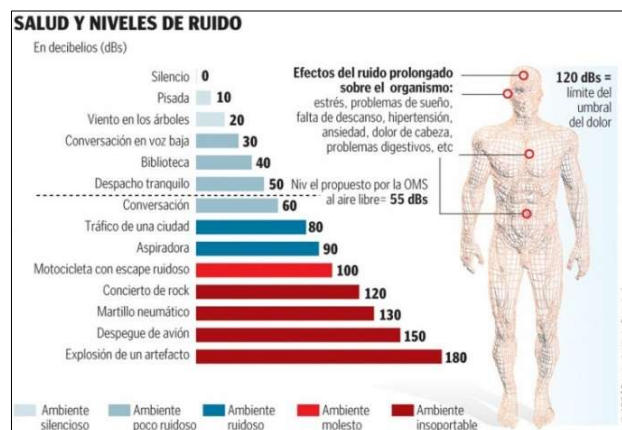


Figura 40. Niveles de ruido y umbral de dolor.

Fuente: Instituto Auditivo Salesa.

El robot explorador aprovechará el umbral de dolor y lo infringirá solo en situación de defensa mediante una sirena tipo turbina de 120dBs, el cual será accionado remota o automáticamente al ser levantado del piso. Dicho sistema de defensa será accionado por un relé que es controlado por el controlador NodeMCU montado en el robot.



*Figura 41. Sirena tipo turbina*

La sirena tipo turbina tiene en su interior un motor DC que gira a alta velocidad. Su diseño mecánico permite generar ruido a partir de la rotación de sus hélices al interior. Fuente: Instituto Auditivo Salesa.

### **2.3.3. Programación del sistema de control y la arquitectura de comunicación**

Para desarrollar este proyecto se utiliza Platformio IDE como entorno de desarrollo integrado, el cual maneja una serie de librerías que permite facilitar la tarea de programación.

Uno de los principales problemas era recuperar las imágenes registradas y mostrarlas en la interfaz visual de un dispositivo portátil. Por lo cual se decidió utilizar el dispositivo de mayor uso en la actualidad, por parte de las personas, que es un smartphone. Luego se optó por aprovechar el poder gráfico que posee este gadget, por lo cual se usa una cámara IP con protocolo de transporte en tiempo real (RTP, por sus siglas en inglés). De esta manera, no se satura el Microcontrolador del NodeMCU y se deja el procesamiento de video, en su totalidad, al smartphone.

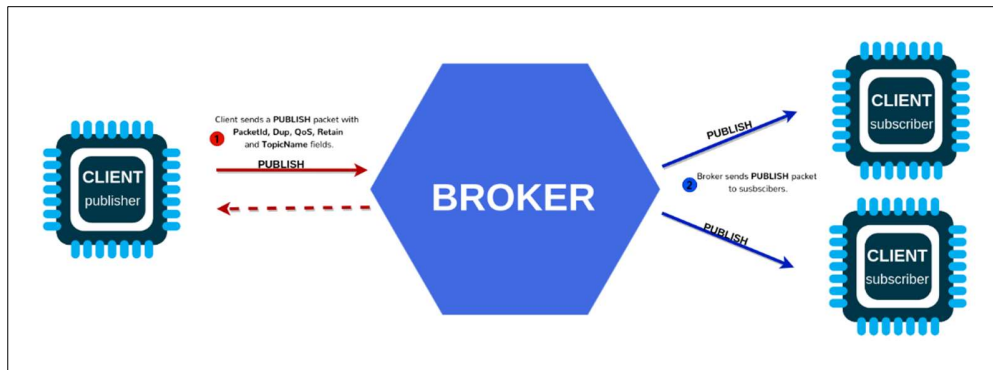


*Figura 42.* Prueba de conexión de Cámara IP con protocolo de transporte en tiempo real (RTP).

Fuente: Propia

Por otro lado, de acuerdo con la infraestructura de la red del centro educacional se vio con la necesidad de encontrar una manera de intercambiar datos que sea rápida y no necesite de un gran ancho de banda. Por esta razón, se decidió integrar el protocolo comunicación MQTT a este proyecto.

El protocolo de comunicación MQTT fue inventado en el año 1999 por Andy Stanford de IBM y Arlen Nipper de Arcom para solucionar el problema que tenían a la hora de hacer radiocomunicación entre sensores instalados en oleoductos y módems satelitales, ya que este tipo de enlaces eran costosos y lentos.



*Figura 43. Arquitectura de comunicación de Protocolo MQTT*

Fuente: MQTT.org

MQTT funciona sobre el paradigma SUSCRIPCIÓN/PUBLICACIÓN, para ello utiliza un BROKER MQTT, en la nube, el cual recibe un mensaje de un publicador (Joystick, MPU6050) bajo un tópico o tema determinado. Una vez ingresado este mensaje, el servidor se encarga de remitir esta información a todos los clientes que se encuentren suscritos (Robot explorador) a este tópico en particular (ver Anexo 19).

La razón por la cual se decide utilizar este protocolo en vez de otro basado en TCP/IP como puede ser HTTP, es por la topología que maneja. Lo que lo hace diferente de otras es de que no existe lo que comúnmente se conoce como Maestro/Esclavo, ya que el Broker es el encargado de retransmitir la información a los suscriptores de un determinado tema cada vez que se actualiza. Esto permite liberar de tráfico la red, ya que el cliente no se mantendría preguntando, a cada momento, si la información se actualizó. Con ello se logra un uso rápido y eficiente de la red, lo cual lo convierte de uso pertinente en una red de escaso ancho de banda.

Por otro lado, se vio con la necesidad de transmitir más de un valor para múltiples variables a la vez. Es allí, donde se dio uso de JSON (JavaScript Object Notation), el cual es un tipo de notación para el intercambio de datos de formato ligero. Este es independiente del tipo de lenguaje de programación y se encuentra estructurado de una cadena de caracteres en el cual se puede indicar múltiples valores para múltiples variables en un solo mensaje. Para ello, el suscriptor debe entender que el mensaje que está recibiendo, de determinado tópico, se encuentra estructurado en formato JSON y deberá disgregar los valores independientes para cada variable aprovechándose de la notación estandarizada.

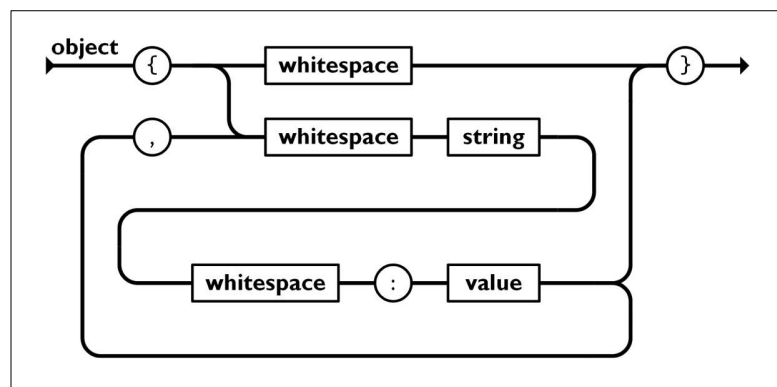


Figura 44. Modelo de estructuración de mensaje JSON.

Fuente: JSON.org

Para lograr este objetivo se desarrolló el siguiente código:

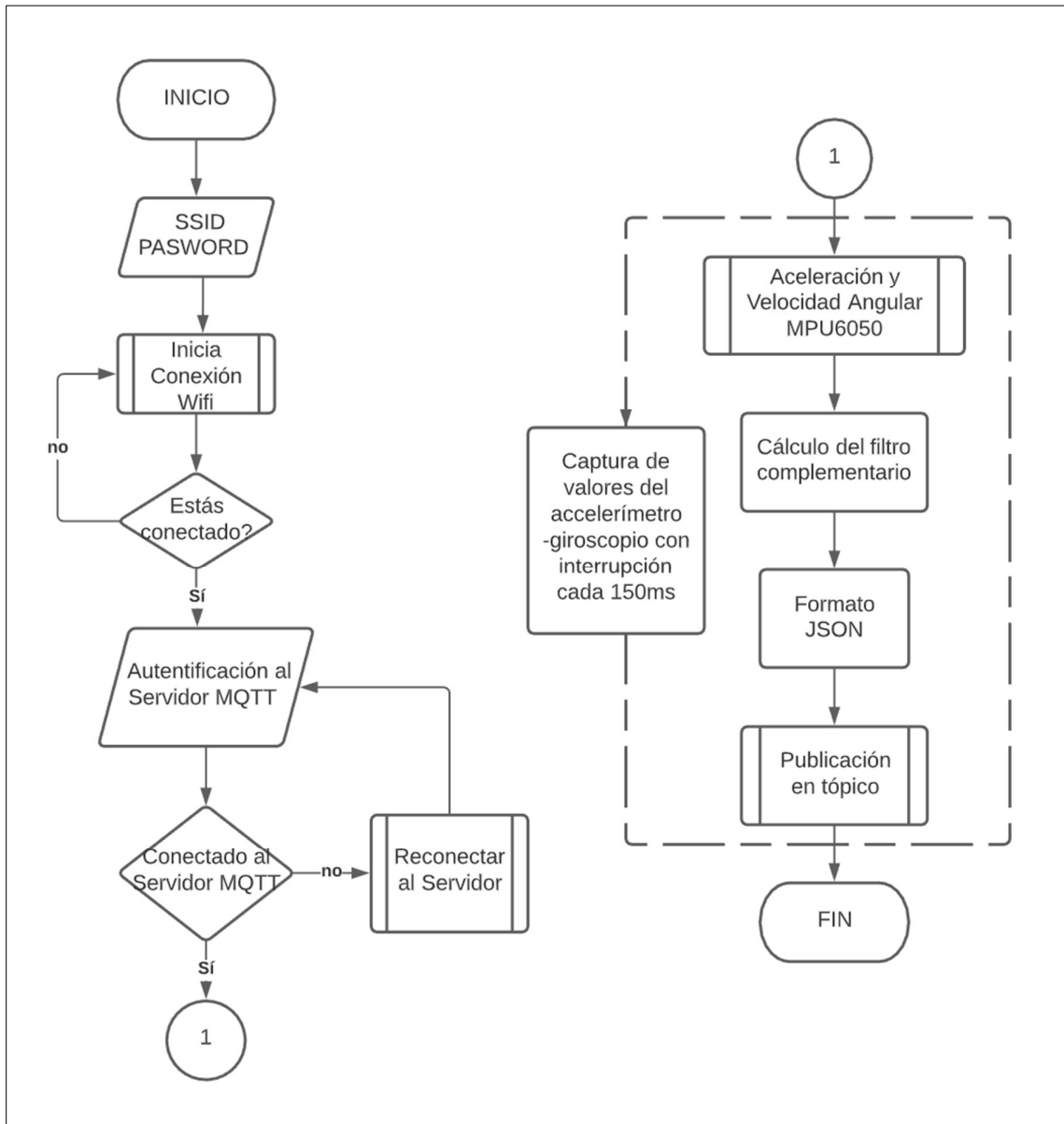


Figura 45. Diagrama de Flujo de control de los Lentes VR

Fuente: Propia.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>

//*****VARIABLES-MPU6050*****
//Direccion I2C de la IMU
#define MPU 0x68

//Ratios de conversion
#define A_R 16384.0 // 32768/2
#define G_R 131.0 // 32768/250

//Conversion de radianes a grados 180/PI
#define RAD_A_DEG = 57.295779

//MPU-6050 da los valores en enteros de 16 bits

int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

//Angulos
float Acc[2]; //Metodo de aceleración
float Gy[3]; //Metodo del giroscopio
float Angle[3]; //Angulo final obtenido

String valores;

long tiempo_prev;
float dt;

int x,y,z;
```

Figura 46. Programación Lentes VR - Parte 1 de 9.

Fuente: Propia.

```
/*******CONFIGURACIÓN WIFI*****  
  
const char* ssid = "HUAWAI10";  
const char* password = "tony12345";  
  
/*******CONFIGURACIÓN MQTT*****  
  
const char* mqtt_server = "broker.shiftr.io"; //Host o servidor a donde me conecto  
const char clientID[]="NODEMCU_ENVIO_MPU"; //identificador unico de para cada dispositivo iot  
const char username[]="robotvr-iot"; //usuario configurado en broker  
const char passwords[]="robotvr-iot"; //contraseña usuario broker  
const char willTopic[]= "";  
int willQoS=0; //0-1-2  
int willRetain=0 ; //0-1 //si se activa o no la retencion de data  
const char willMessage[]=""; //mensaje cuando device este desconectado de broker  
  
/*******VARIABLES GLOBALES*****  
  
WiFiClient espClient;  
PubSubClient client(espClient);  
long lastMsg = 0;  
char msg[50]; //Cadena de caracteres de mensaje a enviar  
int value = 0;
```

Figura 47. Programación Lentes VR - Parte 2 de 9.

Fuente: Propia.

```
void setup() {  
  
  Wire.begin(4,5); // D2(GPIO4)=SDA / D1(GPIO5)=SCL  
  Wire.beginTransmission(MPU);  
  Wire.write(0x6B); //Apuntamos al registro  
  Wire.write(0); //Encendemos el MPU  
  Wire.endTransmission(true); //cierro comunicación para liberar el bus de comunicación I2C  
  
  pinMode(BUILTIN_LED, OUTPUT); // Initialize the BUILTIN_LED pin as an output  
  Serial.begin(115200);  
  setup_wifi();  
  client.setServer(mqtt_server, 1883); //Seteo el cliente MQTT  
  client.setCallback(callback); //Cuando llegue mensaje ejecutar callback  
}
```

Figura 48. Programación Lentes VR - Parte 3 de 9.

Fuente: Propia.



```
void setup_wifi() {  
  
    delay(10);  
    // Empieza a conectarse a una red Wifi  
    Serial.println();  
    Serial.print("Conectando a ");  
    Serial.println(ssid);  
  
    WiFi.begin(ssid, password);  
  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
  
    Serial.println("");  
    Serial.println("WiFi conectado");  
    Serial.println("Dirección IP: ");  
    Serial.println(WiFi.localIP());  
}
```

Figura 49. Programación Lentes VR - Parte 4 de 9.

Fuente: Propia.

```
void callback(char* topic, byte* payload, unsigned int length) {  
    Serial.print("Llegó el mensaje [");  
    Serial.print(topic); //imprimo el topico  
    Serial.print("] ");  
    for (int i = 0; i < length; i++) {  
        Serial.print((char)payload[i]); //imprimo el mensaje  
    }  
    Serial.println();  
  
    // Enciende el LED si se recibio un 1 como primer caracter  
    if ((char)payload[0] == '1') {  
        digitalWrite(BUILTIN_LED, LOW);  
    } else {  
        digitalWrite(BUILTIN_LED, HIGH);  
    }  
}
```

Figura 50. Programación Lentes VR - Parte 5 de 9.

Fuente: Propia.

```
void reconnect() {  
  // Bucle de re-conexión  
  while (!client.connected()) {  
    Serial.print("Intentando conexión MQTT...");  
  
    if(client.connect(clientID,username,passwords,willTopic,willQoS,willRetain,willMessage)){  
      Serial.println("Conectado!");  
      client.subscribe("inTopic");  
    } else {  
      Serial.print("Conexión fallida, CÓDIGO=");  
      Serial.print(client.state());  
      Serial.println(" Vuelve a intentar en 5 segundos");  
      delay(5000);  
    }  
  }  
}
```

*Figura 51.* Programación Lentes VR - Parte 6 de 9.

Fuente: Propia.

```

void mpu_6050(){
  //Leer los valores del Acelerometro de la IMU
  Wire.beginTransmission(MPU);
  Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true); //A partir del 0x3B, se piden 6 registros
  AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();

  //A partir de los valores del acelerometro, se calculan los angulos Y, X
  //respectivamente, con la formula de las tangentes.
  Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
  Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;

  //Leer los valores del Giroscopio
  Wire.beginTransmission(MPU);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true); //A partir del 0x43, se piden 6 registros
  GyX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
  GyY=Wire.read()<<8|Wire.read();
  GyZ=Wire.read()<<8|Wire.read();

  //Calculo del angulo del Giroscopio
  Gy[0] = GyX/G_R;
  Gy[1] = GyY/G_R;
  Gy[2] = GyZ/G_R;

  dt = (millis() - tiempo_prev) / 1000.0;
  tiempo_prev = millis();

  //Aplicar el Filtro Complementario
  Angle[0] = 0.98 *(Angle[0]+Gy[0]*dt) + 0.02*Acc[0];
  Angle[1] = 0.98 *(Angle[1]+Gy[1]*dt) + 0.02*Acc[1];

  //Integración respecto del tiempo paras calcular el YAW
  Angle[2] = Angle[2]+Gy[2]*dt;

  //Mostrar los valores por consola

  //Valores calibrados para pasarselos a los servomotores
  x =((-1*Angle[0])/1.5)-80;
  y= ((-1*Angle[1])*4.5)-30;
  z =Angle[2];
  valores = "90, " +String(x) + ", " + String(y) + ", " + String(z) + ", -90";
  //Serial.println(valores);

  delay(10);
}
  
```

Figura 52. Programación Lentes VR - Parte 7 de 9.

Fuente: Propia.

```
//FUNCION PARA CONVERTIR A FORMATO JSON
String json() {

    String value = "\"X\":" + String(x);
    String value2 = ",\"Y\":" + String(y) ;
    String value3 = ",\"Z\":" + String(z) ;

    value = value + value2 + value3;
    String payload = "{" + value + "}";

    return payload;
}
```

Figura 53. Programación Lentes VR - Parte 8 de 9.

Fuente: Propia.

```
void loop() {

    if (!client.connected()) {
        reconnect();
    }
    client.loop(); // para que los mensajes lleguen cuando se ejecute
    mpu_6050();

    long now = millis();
    if (now - lastMsg > 150) {
        lastMsg = now;

        char sense[50]=""; //almacenar dato en char array

        String my_payload=json();
        my_payload.toCharArray(sense, (my_payload.length() + 1)); //luego convierto a char array
        client.publish("proyecto_servo/mpu6050", sense); //siempre recibe un char array
        Serial.println(sense);
    }
}
```

Figura 54. Programación Lentes VR - Parte 9 de 9.

Fuente: Propia.

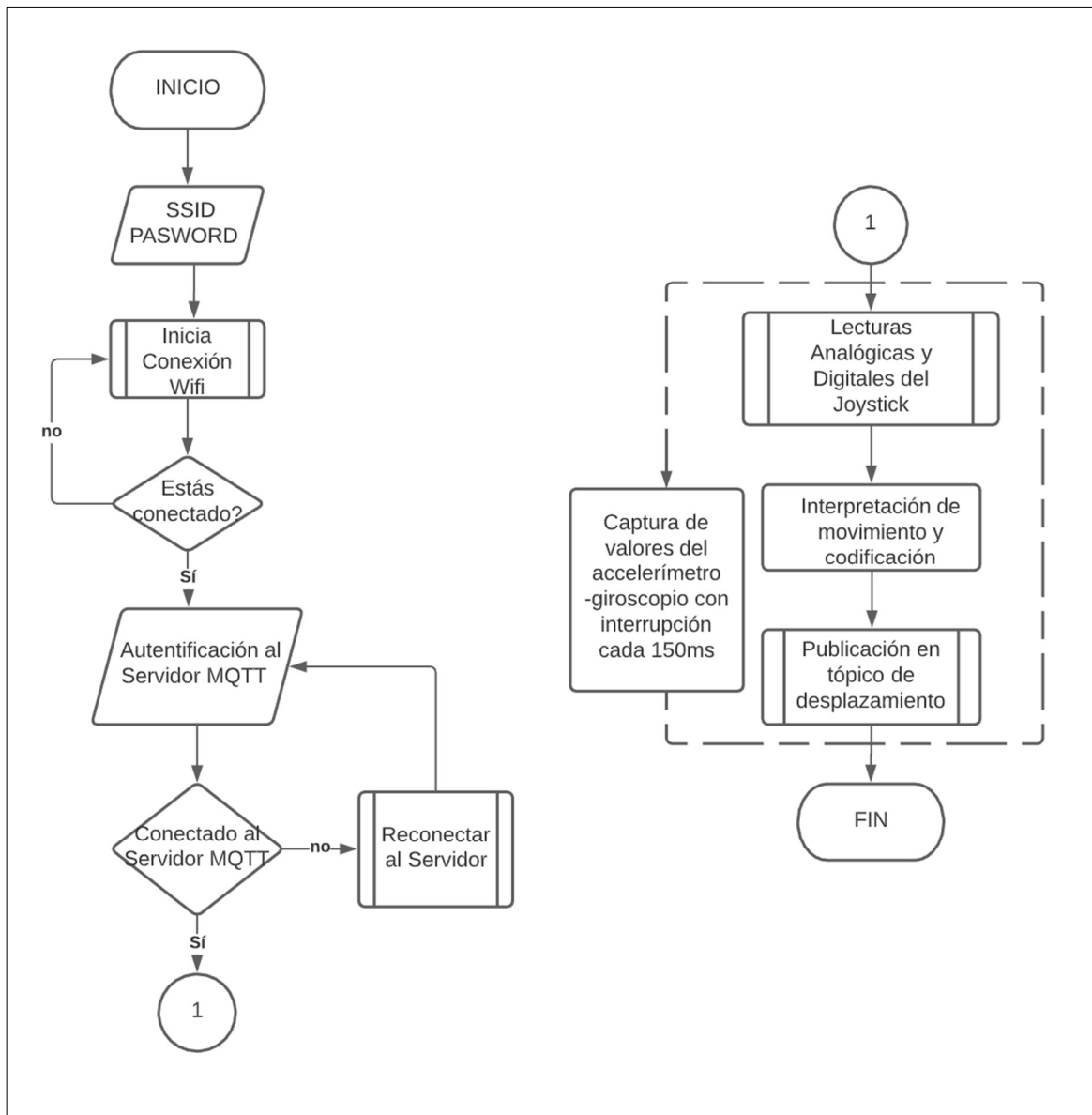


Figura 55. Diagrama de Flujo de control del Joystick

Fuente: Propia.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

//*****CONFIGURACIÓN WIFI*****

const char* ssid = "HUAWEI10";
const char* password = "tony12345";

//*****CONFIGURACIÓN MQTT*****

const char* mqtt_server = "broker.shiftr.io";
const char clientID[]="NODEMCU-RECEPCION-SERVOS"; //identificador unico de para cada dispositivo iot
const char username[]="robotvr-iot"; //usuario configurado en broker
const char passwords[]="robotvr-iot"; //contraseña usuario broker
const char willTopic[]= "";
int willQoS=0; //0-1-2
int willRetain=0; //0-1 //si se activa o no la retencion de data
const char willMessage[]=""; //mensaje cuando device este desconectado de broker

//*****VARIABLES GLOBALES*****

char c; //variable para contener caracteres del payload
String readString="";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
```

Figura 56. Programación de Joystick - Parte 1 de 5.

Fuente: Propia.

```
void setup() {

  pinMode(BUILTIN_LED, OUTPUT); // Inicializa el pin led incorporado como una salida
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);

  pinMode(D1, INPUT);
  pinMode(D2, INPUT);
  pinMode(D5, INPUT);
  pinMode(D4, INPUT);

}
```

Figura 57. Programación de Joystick - Parte 2 de 5.

Fuente: Propia.

```
void setup_wifi() {  
  
    delay(10);  
    // Empieza a conectarse a una red Wifi  
    Serial.println();  
    Serial.print("Connecting to ");  
    Serial.println(ssid);  
  
    WiFi.begin(ssid, password);  
  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
  
    Serial.println("");  
    Serial.println("WiFi connected");  
    Serial.println("IP address: ");  
    Serial.println(WiFi.localIP());  
}
```

Figura 58. Programación de Joystick - Parte 3 de 5.

Fuente: Propia.

```
void reconnect() {  
    // Bucle de re-conexión  
    while (!client.connected()) {  
        Serial.print("Intentando conexión MQTT...");  
  
        if(client.connect(clientID,username,passwords,willTopic,willQoS,willRetain,willMessage)){  
            Serial.println("Conectado!");  
  
            client.subscribe("proyecto_servo/mpu6050");  
            client.subscribe("proyecto_servo/mpu6050_motor");  
        } else {  
            Serial.print("Conexión fallida, CÓDIGO=");  
            Serial.print(client.state());  
            Serial.println(" Vuelve a intentar en 5 segundos");  
            delay(5000);  
        }  
    }  
}
```

Figura 59. Programación de Joystick - Parte 4 de 5.

Fuente: Propia.



```
void loop() {  
  
  if (!client.connected()) {  
    reconnect();  
  }  
  
  client.loop(); // para que los mensajes lleguen cuando se ejecute  
  
  long now = millis();  
  if (now - lastMsg > 160) {  
    lastMsg = now;  
  
    int x=digitalRead(D1);  
    int y=digitalRead(D2);  
    int a=analogRead(A0);  
  
    Serial.print(x);  
    Serial.print(y);  
    Serial.print(a);  
  
    if(x==1 && y==0 && (a<700 && a>500)){  
      char v1[3]="0";  
      client.publish("proyecto_servo/mpu6050_motor",v1);  
      Serial.println("stop");}  
    if(x==0 && y==0 && (a<700 && a>500)){  
      char v1[3]="A";  
      client.publish("proyecto_servo/mpu6050_motor",v1);  
      Serial.println("avanzar");}  
    if(x==1 && y==1 && (a<700 && a>500)){  
      char v1[3]="D";  
      client.publish("proyecto_servo/mpu6050_motor",v1);  
      Serial.println("retroceder");}  
      if(x==1 && y==0 && a>800){  
        char v1[3]="B";  
        client.publish("proyecto_servo/mpu6050_motor",v1);  
        Serial.println("izquierda");}  
      if(x==1 && y==0 && a<200){  
        char v1[3]="C";  
        client.publish("proyecto_servo/mpu6050_motor",v1);  
        Serial.println("derecha");}  
    }  
  }  
}
```

Figura 60. Programación de Joystick - Parte 5 de 5.

Fuente: Propia.



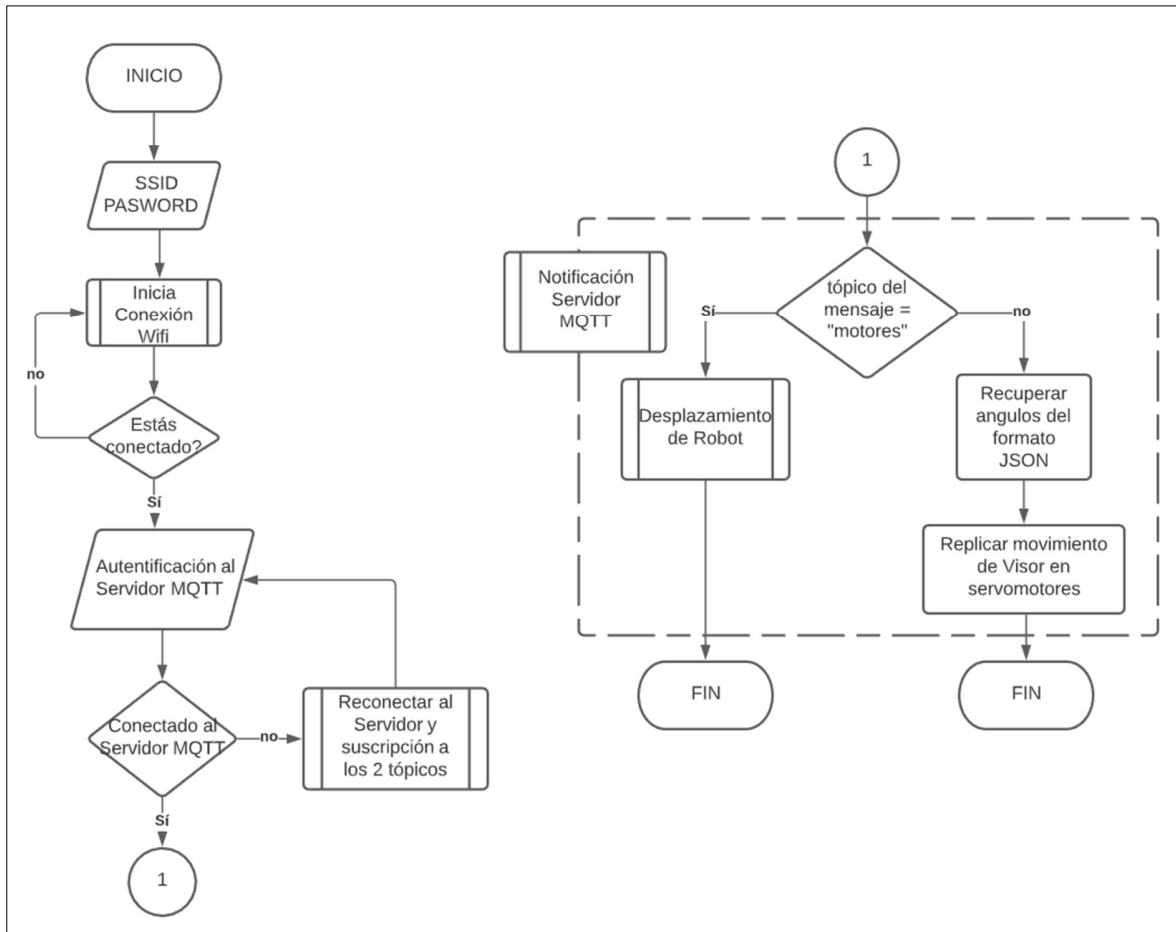


Figura 61. Diagrama de control Robot explorador VR.

Fuente: Propia.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Servo.h>
#include <ArduinoJson.h>

#define motor_1_a    D3
#define motor_1_r    D4
#define motor_2_a    D7
#define motor_2_r    D8
#define pin_pwm1     D5
#define pin_pwm2     D6

//////////////////////CREACION DEL OBJETO SERVOMOTOR//////////////////////

Servo myservo;      // Creamos los objetos Servo para controlar los servomotores
Servo myservo2;

int pin_servo=5;    ///eje x, pin gpio (D1)del la nodemcu
int pos=0;
int pin_servo2=4;  ///eje y, pin gpio (D2)del la nodemcu
int pos2=0;
```

*Figura 62.* Programación Robot explorador VR - Parte 1 de 8.

Fuente: Propia.

```

//*****CONFIGURACIÓN WIFI*****

const char* ssid = "HUAWEI10";
const char* password = "tony12345";

//*****CONFIGURACIÓN MQTT*****
const char* mqtt_server = "broker.shiftr.io";
const char clientID[]="NODEMCU-RECEPCION-SERVOS"; //identificador unico de para cada dispositivo iot
const char username[]="robotvr-iot"; //usuario configurado en broker
const char passwords[]="robotvr-iot"; //contraseña usuario broker
const char willTopic[]="";
int willQoS=0; //0-1-2
int willRetain=0; //0-1 //si se activa o no la retencion de data
const char willMessage[]=""; //mensaje cuando device este desconectado de broker
///////////////////////////////////////////////////////////////////

//*****VARIABLES GLOBALES*****

char c; //variable para contener caracteres del payload
String readString="";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
  
```

Figura 63. Programación Robot explorador VR - Parte 2 de 8.

Fuente: Propia.

```

void setup() {

  pinMode(motor_1_a, OUTPUT);
  pinMode(motor_1_r, OUTPUT);
  pinMode(motor_2_a, OUTPUT);
  pinMode(motor_2_r, OUTPUT);
  pinMode(pin_pwm1, OUTPUT);
  pinMode(pin_pwm2, OUTPUT);
  myservo.attach(pin_servo); // conecta el servo en GIO2 al objeto servo
  myservo.write(0); // Indica a que posición ir al servomotor
  myservo2.attach(pin_servo2);
  myservo2.write(0);

  pinMode(D6,OUTPUT);
  pinMode(BUILTIN_LED, OUTPUT); // Inicializa el pin BUILTIN_LED como una salida
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
  
```

Figura 64. Programación Robot explorador VR - Parte 3 de 8.

Fuente: Propia.

```
//RECEPCION DE DATA
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Llegó el mensaje [");
  Serial.print(topic); //imprimo el topico
  String string_topic;
  string_topic = String(topic);
  Serial.print("] ");

  String readStrings="";
  char c;
  char json_data_in[50];
  for (int i = 0; i < length; i++) {

    Serial.print((char)payload[i]); //imprimo el mensaje caracter x caracter
    c=payload[i];
    readStrings +=c; //concatena datos que llegan
  }

  if(string_topic=="proyecto_servo/mpu6050_motor")
  {
    if(readStrings=="A"){
      Serial.println("adelante");
      motores(100,100);
    }
    if(readStrings=="B"){
      Serial.println("derecha");
      motores(0,100);
    }
    if(readStrings=="C"){
      Serial.println("izquierda");
      motores(100,0);
    }
    if(readStrings=="D"){
      Serial.println("atras");
      motores(-100,-100);
    }
    if(readStrings=="0"){
      Serial.println("parar");
      motores(0,0);
    }
  }
}
```

Figura 65. Programación Robot explorador VR - Parte 4 de 8.

Fuente: Propia.

```
}else
{
    Serial.println();

    readStrings.toCharArray(json_data_in, (readStrings.length() + 1));

    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.parseObject(json_data_in);
    if (!root.success()) {
        Serial.println("parseObject() failed");
        return;
    }
    int pos = root["X"];
    int pos2 = root["Y"];
    int pos3 = root["Z"];

    int serv1=90-pos+5;
    myservo.write(serv1);
    int serv2=90-pos2;
    serv2=180-serv2;
    myservo2.write(serv2);

    Serial.print(pos);Serial.print(" ");Serial.print(pos2);Serial.print(" ");Serial.println(pos3);
}
}
```

Figura 66. Programación Robot explorador VR - Parte 5 de 8.

Fuente: Propia.

```

void reconnect() {
  // Bucle de re-conexión
  while (!client.connected()) {
    Serial.print("Intentando conexión MQTT...");

    if(client.connect(clientID,username,passwords,willTopic,willQoS,willRetain,willMessage)){
      Serial.println("Conectado!");
      // Una vez conectado me suscribo a los dos topicos
      client.subscribe("proyecto_servo/mpu6050");
      client.subscribe("proyecto_servo/mpu6050_motor");

      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      digitalWrite(BUILTIN_LED,!digitalRead(BUILTIN_LED));
      delay(200);
      motores(0,0);

    } else {
      Serial.print("Conexión fallida, CÓDIGO=");
      Serial.print(client.state());
      Serial.println(" Vuelve a intentar en 5 segundos");
      delay(5000);
    }
  }
}

```

Figura 67. Programación Robot explorador VR - Parte 6 de 8.

Fuente: Propia.

```
void motores(float pwm1, float pwm2)
{
    int v1, v2;

    //para el motor 1
    if(pwm1 > 0)
    {
        digitalWrite(motor_1_a, HIGH);
        digitalWrite(motor_1_r, LOW);
        v1 = pwm1;
    }
    if(pwm1 < 0)
    {
        digitalWrite(motor_1_a, LOW);
        digitalWrite(motor_1_r, HIGH);
        v1 = pwm1 * (-1);
    }
    if(pwm1 == 0)
    {
        digitalWrite(motor_1_a, LOW);
        digitalWrite(motor_1_r, LOW);
        v1 = pwm1;
    }

    //para el motor 2
    if(pwm2 > 0)
    {
        digitalWrite(motor_2_a, HIGH);
        digitalWrite(motor_2_r, LOW);
        v2 = pwm2;
    }
    if(pwm2 < 0)
    {
        digitalWrite(motor_2_a, LOW);
        digitalWrite(motor_2_r, HIGH);
        v2 = pwm2 * (-1);
    }
    if(pwm2 == 0)
    {
        digitalWrite(motor_2_a, LOW);
        digitalWrite(motor_2_r, LOW);
        v2 = pwm2 * (-1);
    }

    analogWrite(pin_pwm1, v1);
    analogWrite(pin_pwm2, v2);
}
```

Figura 68. Programación Robot explorador VR - Parte 7 de 8.

Fuente: Propia.

```
void loop() {  
  
    if (!client.connected()) {  
        reconnect();  
    }  
    client.loop();  
  
    long now = millis();  
    if (now - lastMsg > 1000) {  
        lastMsg = now;  
    }  
}
```

*Figura 69.* Programación Robot explorador VR - Parte 8 de 8.

Fuente: Propia.



### CAPÍTULO III. RESULTADOS

Los resultados obtenidos gracias a la aplicación de la investigación fueron los esperados, ya que se pudo eliminar la exposición del encargado de vigilancia, mediante la creación de un interfaz intuitivo hombre-robot como un sistema inmersivo capaz de recrear el ambiente explorado. De acuerdo con las pruebas realizadas, los operadores experimentaban una sensación semejante a la del patrullaje físico.

La implementación del protocolo de comunicación MQTT permitió un enlace rápido y estable entre Visor VR-Joystick y Robot Explorador a pesar del carente ancho de banda del internet. Además, el desarrollo de la aplicación termina por proveer un servidor IoT escalable de lenguaje abierto para la futura integración de mejoras o complementos al proyecto, como pueden ser detectores de movimiento inalámbricos (publicadores) en puntos estratégicos e integración con el sistema de pánico (suscriptor).

El escaso ancho de banda no fue un limitante, ya que cada valor registrado por el visor VR o el Joystick en el servidor MQTT fue inmediatamente replicado por el robot, volviendo casi imperceptible el tiempo de latencia.



*Figura 70.* Prueba realizada en campo con una profesora del centro educacional.

Fuente: Propia.

Se redujo los tiempos de patrullaje y se hicieron más efectivos los reconocimientos de eventos, ya que no pone en riesgo a la persona encargada de la seguridad del recinto. Este puede mantenerse oculto durante las noches e ir monitoreando sin ser detectado mediante su lente de visión nocturna.

Los cálculos para el dimensionamiento de motores fueron los adecuados. Esto permitió sortear los diversos tipos de suelo y pendientes con los que cuenta el centro educacional Peruano Japonés.



*Figura 71.* Prueba realizada sobre terreno no afirmado con pendiente.

Fuente: Propia.



*Figura 72.* Prueba realizada sobre superficie de concreto.

Fuente: Propia.



*Figura 73.* Prueba realizada sobre áreas verdes con pendiente.

Fuente: Propia.

## CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES

### 4.1. Discusión

El objetivo general que plantea esta investigación es la implementación de un robot explorador con realidad virtual para incrementar la seguridad del Centro Educativo Peruano Japonés en el distrito de Villa el Salvador, evitando de esta manera, la exposición al peligro de los integrantes de seguridad y maximizar la eficacia del control áreas a patrullar mediante implementación de tecnología. Para ello, se contaron con antecedentes de desarrollo tecnológico en otras áreas como comercial, bibliotecas, museos e implementación en seguridad contra atentados. La mayoría de estos proyectos son controlados mediante radiofrecuencia, ya que es altamente fiable y su tiempo de respuesta a comandos es menor a los 10ms. Sin embargo, para que estos puedan ser controlados se necesita que el operario se encuentre dentro del perímetro de cobertura de la señal que suele ser de poco alcance.

En contraste con ello, el proyecto de esta investigación cuenta con tecnología Wifi 2.4 GHz llegando a cubrir mayores distancias. Con lo cual el operario, puede encontrarse al extremo de la cobertura local de la señal o inclusive fuera de ella, ya que con la tecnología 4G LTE le permite acceder a la gran nube del internet desde cualquier parte del mundo. Sin embargo, la desventaja que presenta es el aumento del retardo de respuesta, ya que este puede alcanzar los 160 ms en áreas donde la cobertura no es muy buena.



## 4.2. Conclusiones

En esta tesis se incrementó la posibilidad de detección temprana a un intento de robo, esto por los beneficios que entrega el robot explorador con realidad virtual integrada como lo son la salvaguarda del operario, respuesta rápida, tolerante al peligro, facilidad de desplazamiento, monitoreo sin luz, sistema inmersivo, capacidad de defensa, activación remota de alarmas. Como consecuencia, se tiene mayor tiempo de respuesta a un intento de robo en el cual el operario puede dar la alerta oportunamente.

En segundo lugar, se eliminó el riesgo de violentización del personal de seguridad, ya que con el uso de conocimiento en electrónica, mecánica e informática se ideó una solución capaz entregar al operario un control remoto del patrullaje, salvaguardando su integridad mediante el reemplazo físico por el robot, sin la reducción de sus capacidades perceptivas y la dotación de nuevas capacidades como el camuflaje y la visión nocturna.

Por último, se aumentó la eficiencia de patrullaje con el uso de realidad virtual, ya que, con esta técnica unida con protocolos ligeros de comunicación, además del uso de la tracción oruga y los motores DC, con torque y velocidad adecuados, se logró recrear una experiencia vivencial real. Es por esta razón, que el operario de seguridad ahora puede realizar sus labores de patrullaje con tranquilidad y sobre todo con mayor eficacia. Las pruebas realizadas en campo con el personal de seguridad y profesores del Instituto Educacional fueron satisfactorias de acuerdo con lo esperado.

## REFERENCIAS

- Arellano, L. (2015). *Diseño e implementación de un robot móvil con control de trayectoria mediante principios odométricos*. (Tesis licenciatura). Universidad Nacional Mayor de San Marcos, Perú.
- Carrera, I. (2012). *Diseño, construcción y control de un robot doméstico para asistencia y entrenamiento de la movilidad*. (Tesis doctoral). Universidad Politécnica de Madrid, España.
- Cook, J. (2012). *Diseño e implementación de un sistema de generación de trayectorias para un robot móvil utilizando control odométrico*. (Tesis de licenciatura). Pontificia Universidad Católica del Perú, Perú.
- Gracia, L. (2015). *Modelado Cinemático y Control de Robots Móviles con Ruedas*. (Tesis doctoral). Universidad Politécnica de Valencia, España.
- Herrera, A. (2015). *Diseño e implementación del sistema de vuelo autónomo de un vehículo aéreo no tripulado para el reconocimiento en zonas hostiles azotadas por el narcoterrorismo en la Amazonía peruana*. (Tesis de licenciatura). Universidad Ricardo Palma, Perú.
- Herrera, G. (2014). *Navegación Autónoma de un robot móvil*. (Tesis licenciatura). Universidad Nacional Autónoma de México, México D.F.

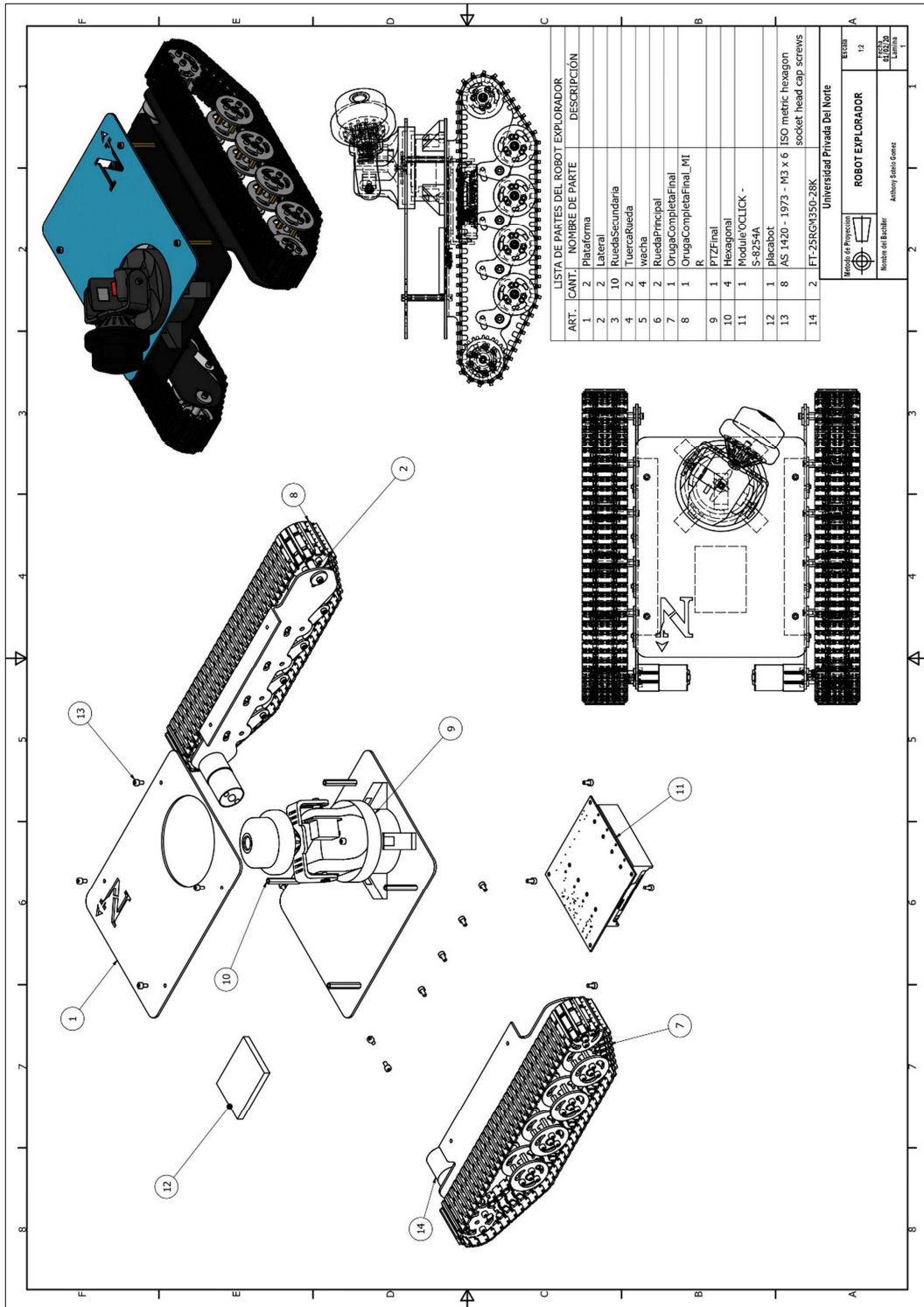
Naranjo, K. (2013). *Implementación de un prototipo de robot móvil con visión nocturna para vigilancia y seguridad*. (Tesis licenciatura). Universidad estatal península de Santa Elena, Ecuador.

Reyes Cortés, F. (2011). *Robótica, Control de robots manipuladores*. México D.F.: Alfaomega.

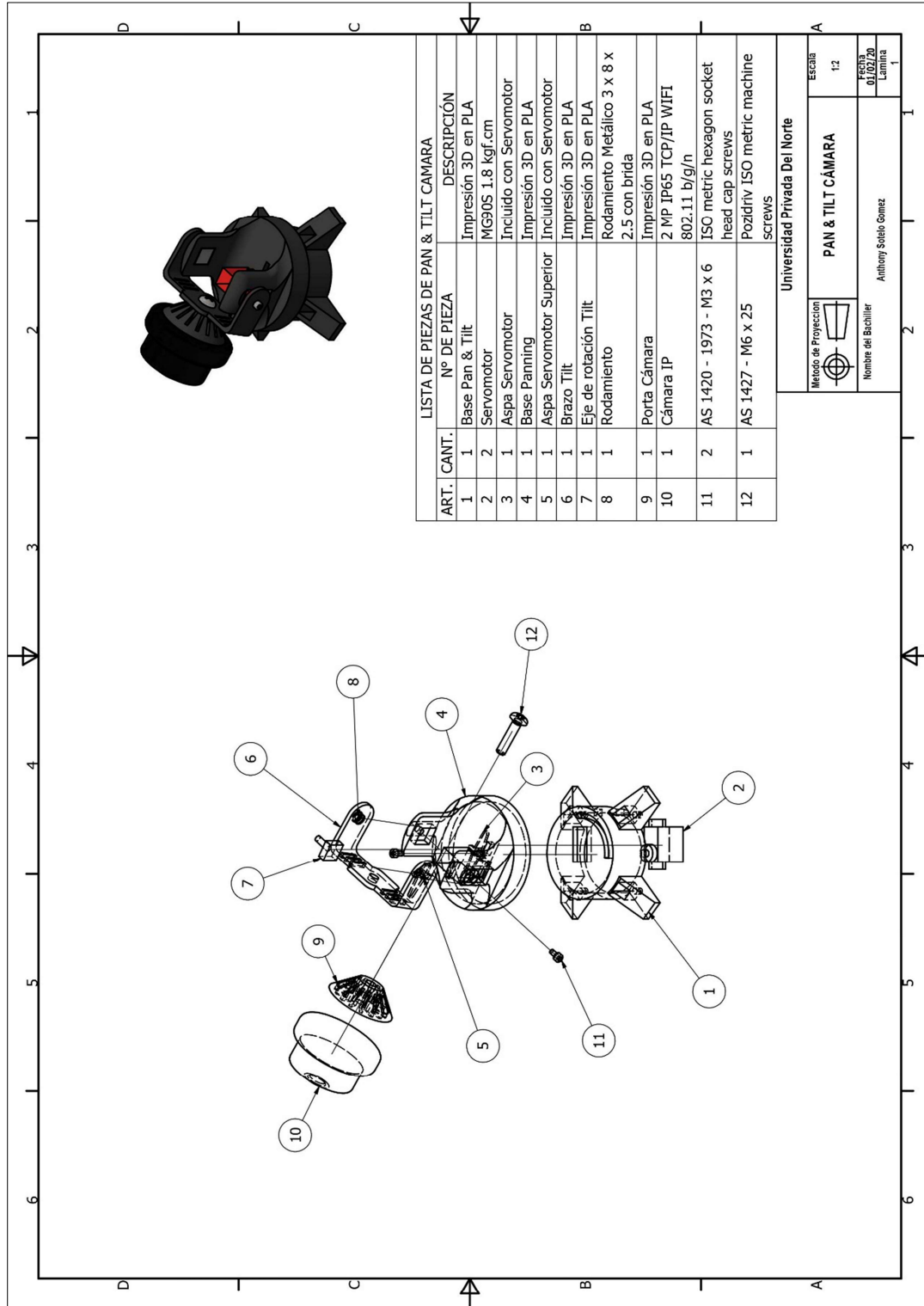


**ANEXOS**

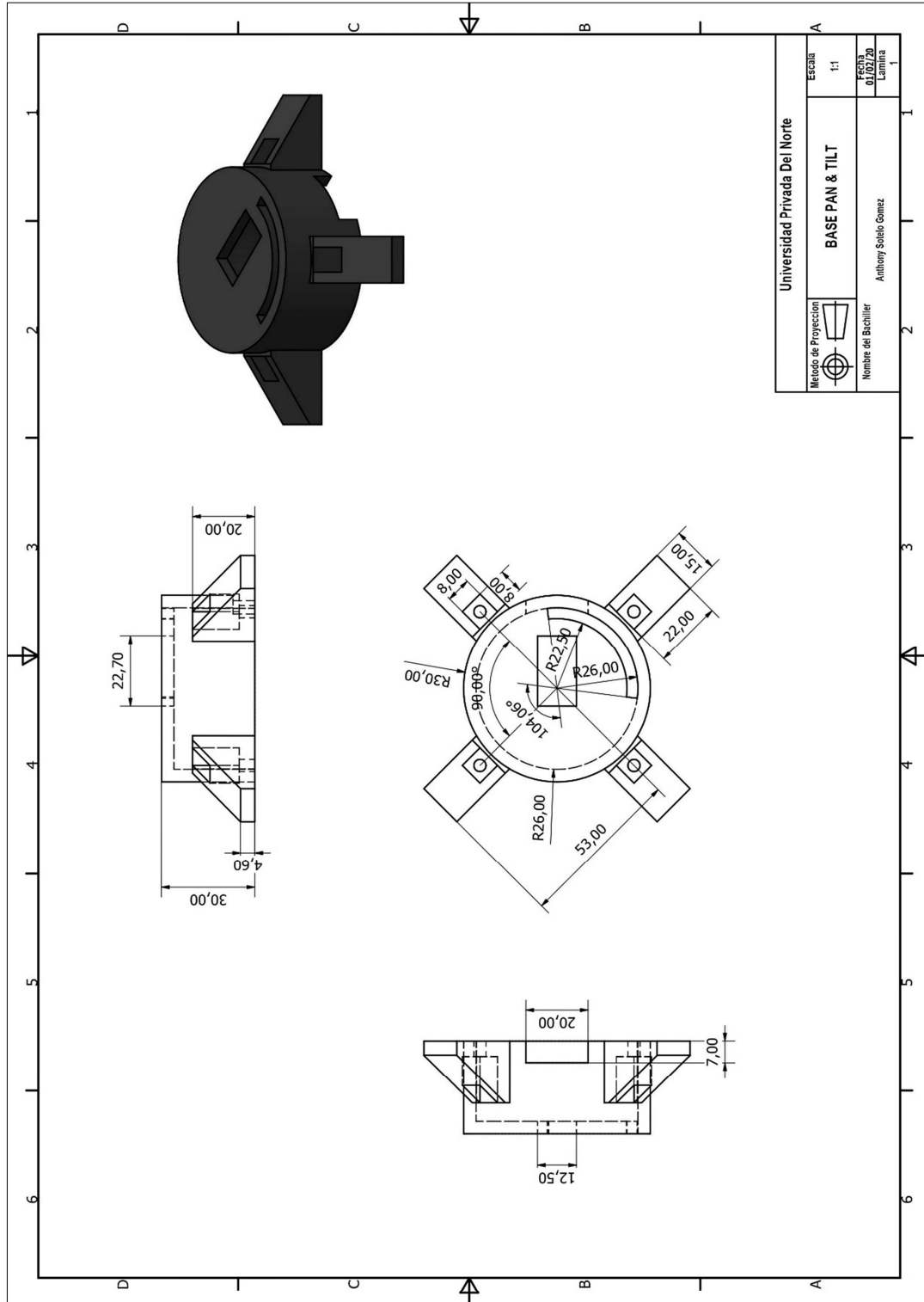
Anexo N° 1. Plano en explosión de robot explorador.



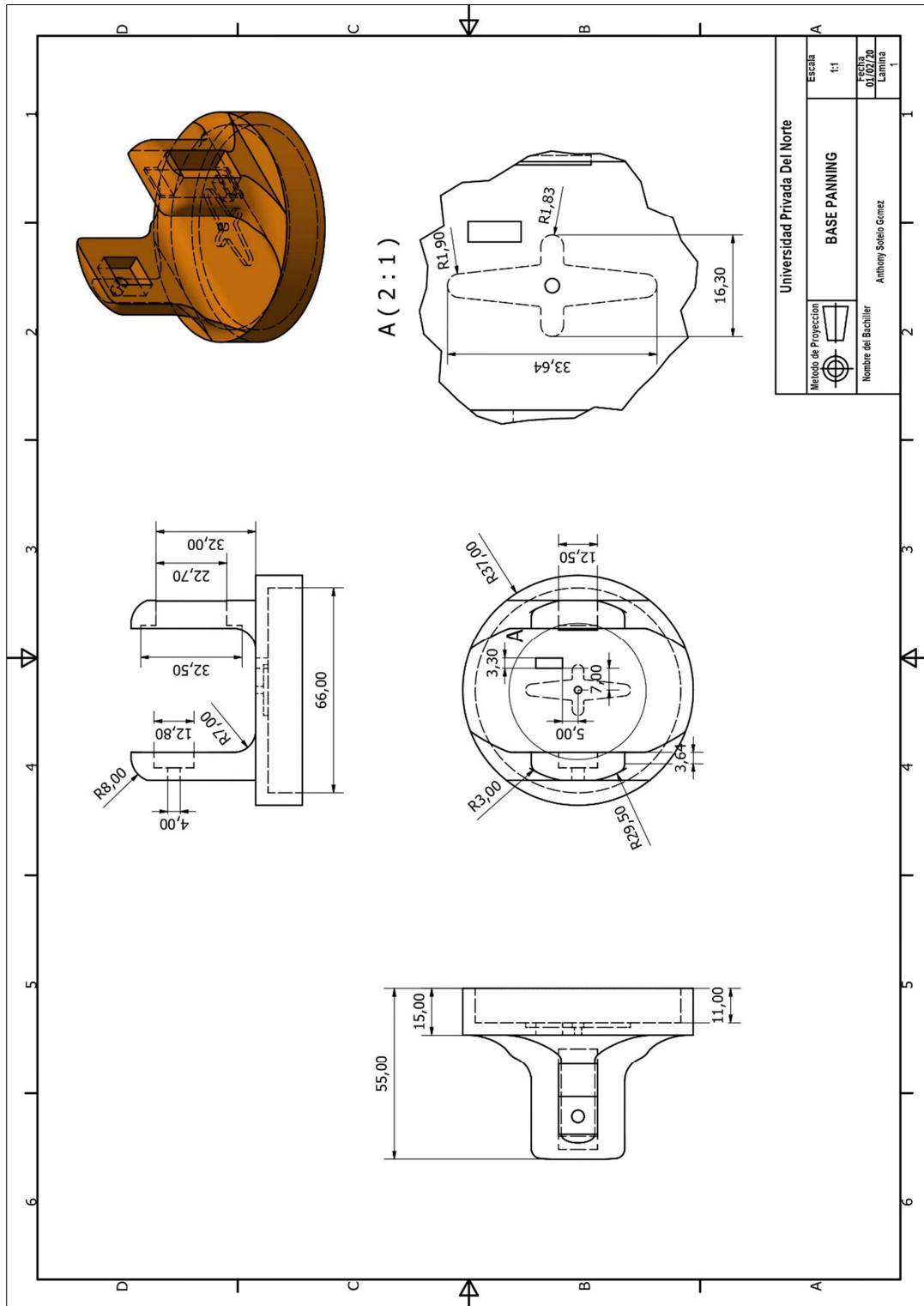
Anexo N° 2. Plano en explosión de robot explorador.



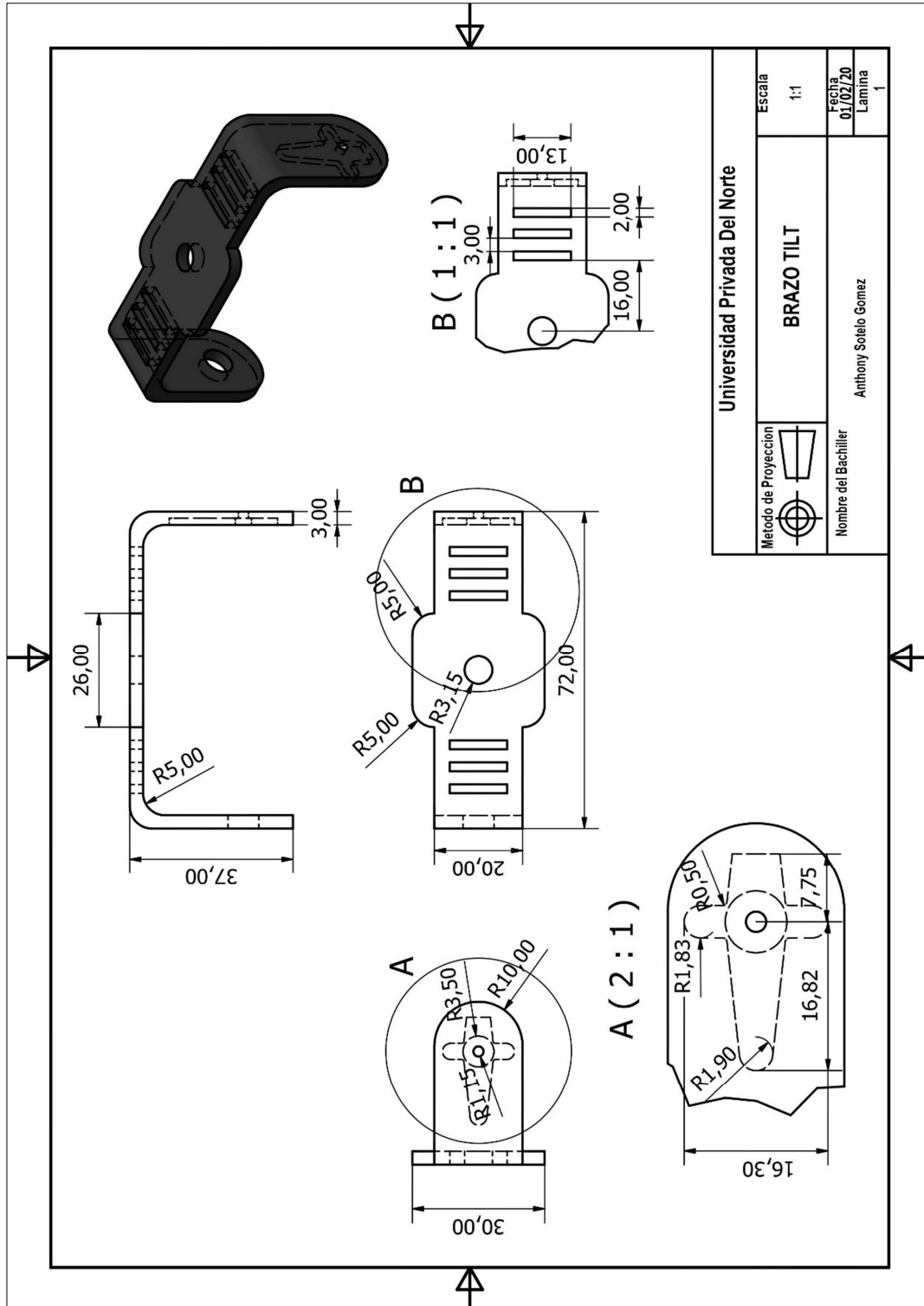
Anexo N° 3. Plano en explosión de robot explorador.



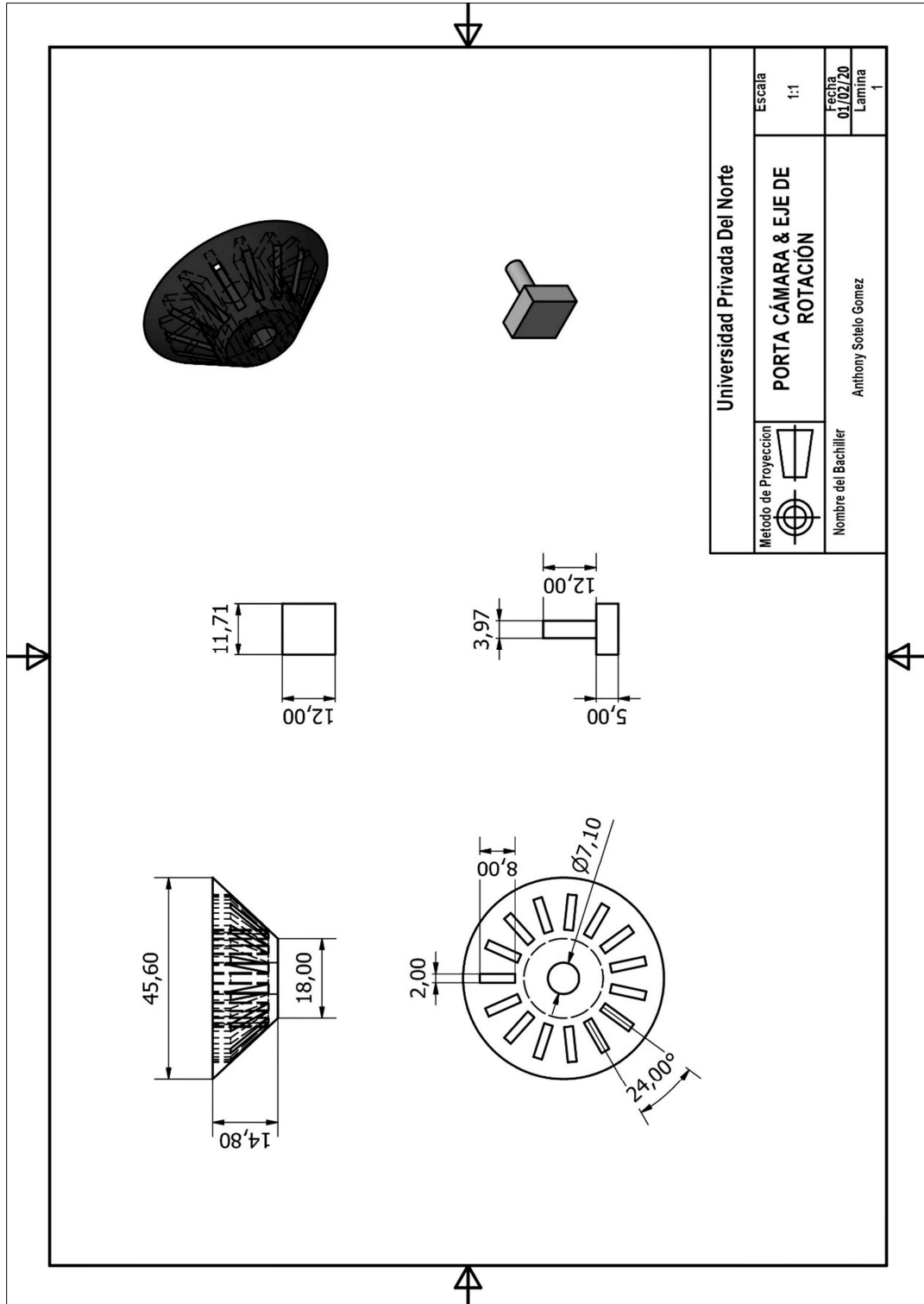
Anexo N° 4. Plano en explosión de robot explorador.



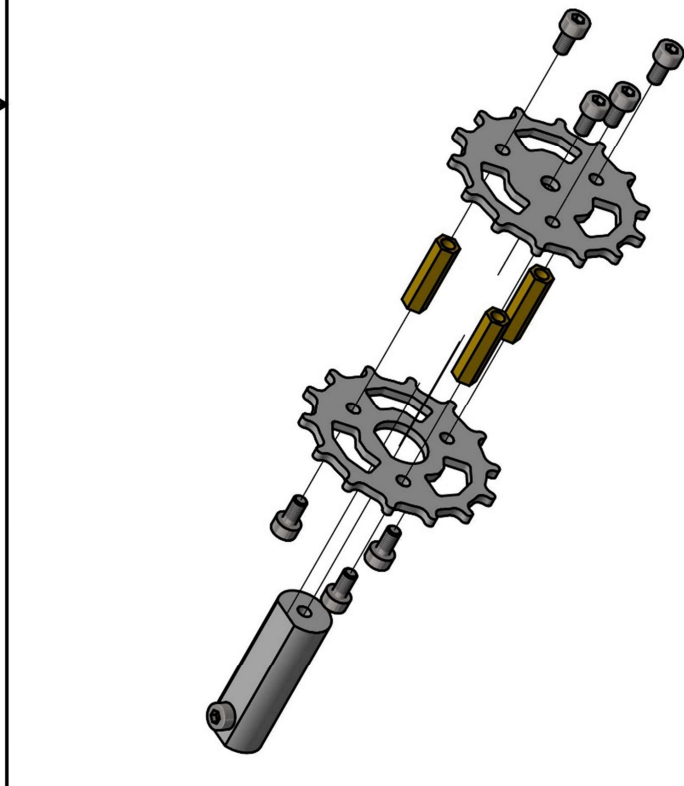
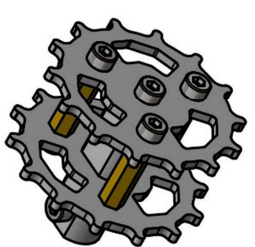
Anexo N° 5. Plano en explosión de robot explorador.



Anexo N° 6. Plano en explosión de robot explorador.




Anexo N° 7. Plano en explosión de robot explorador.

LISTA DE PIEZAS DE RUEDA DE TRANSMISIÓN			
ART.	CANT.	N° DE PIEZA	DESCRIPCIÓN
1	2	Disco de Transmisión	Disco dentado de 2 mm de espesor
2	1	Eje de acople	Acero 1018
3	3	Separador Hexagonal	Metálico Hembra/Hembra M3 18 mm
4	8	AS 1420 - 1973 - M3 x 6	Tornillos métricos ISO con cabeza cilíndrica con hueco hexagonal

Metodo de Proyección



**RUEDA DE TRANSMISIÓN**

Nombre del Bachiller  
Anthony Sotelo Gomez

Universidad Privada Del Norte

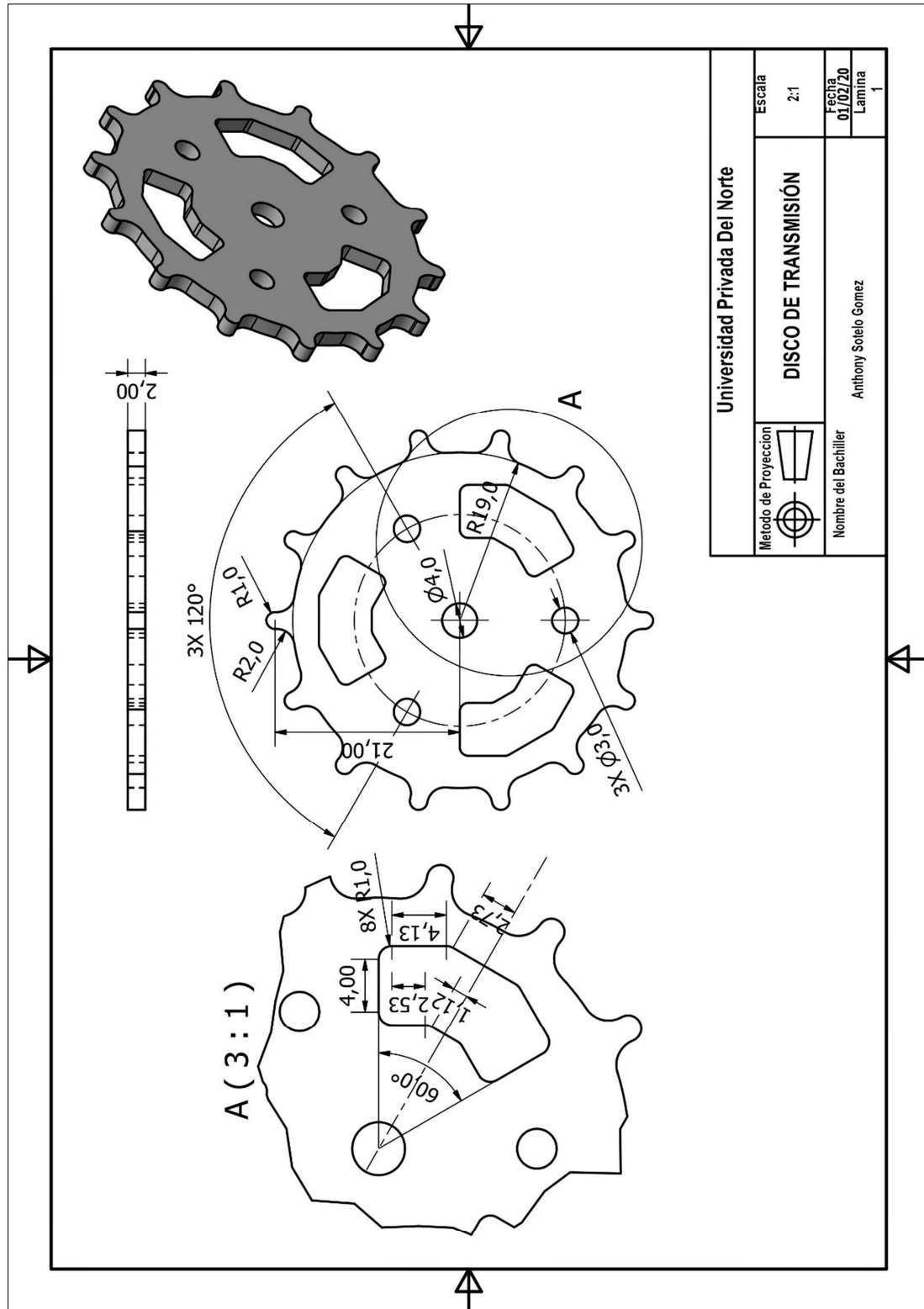
Escala  
1:1

Fecha  
01/02/20

Lamina  
1

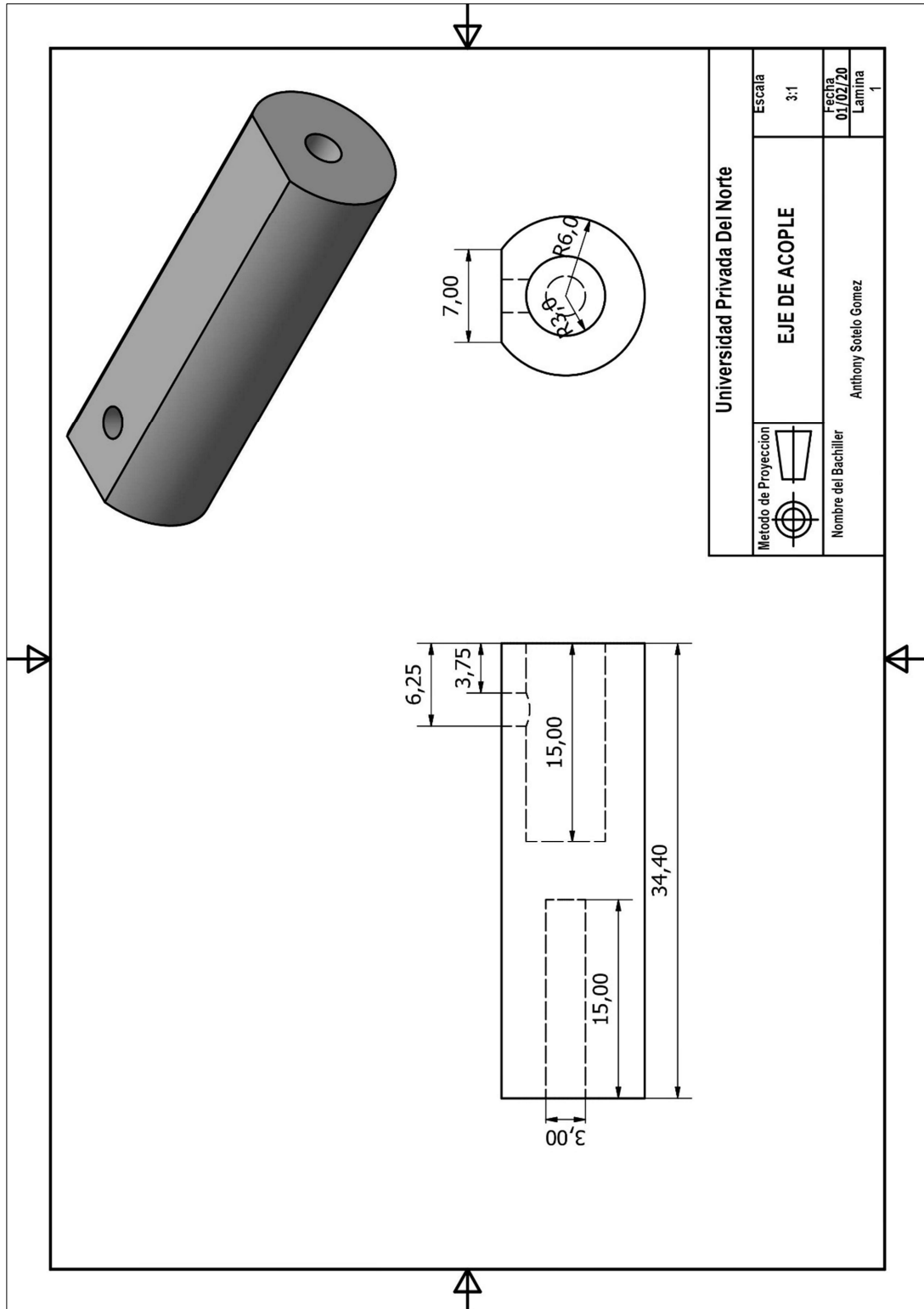


Anexo N° 8. Plano en explosión de robot explorador.



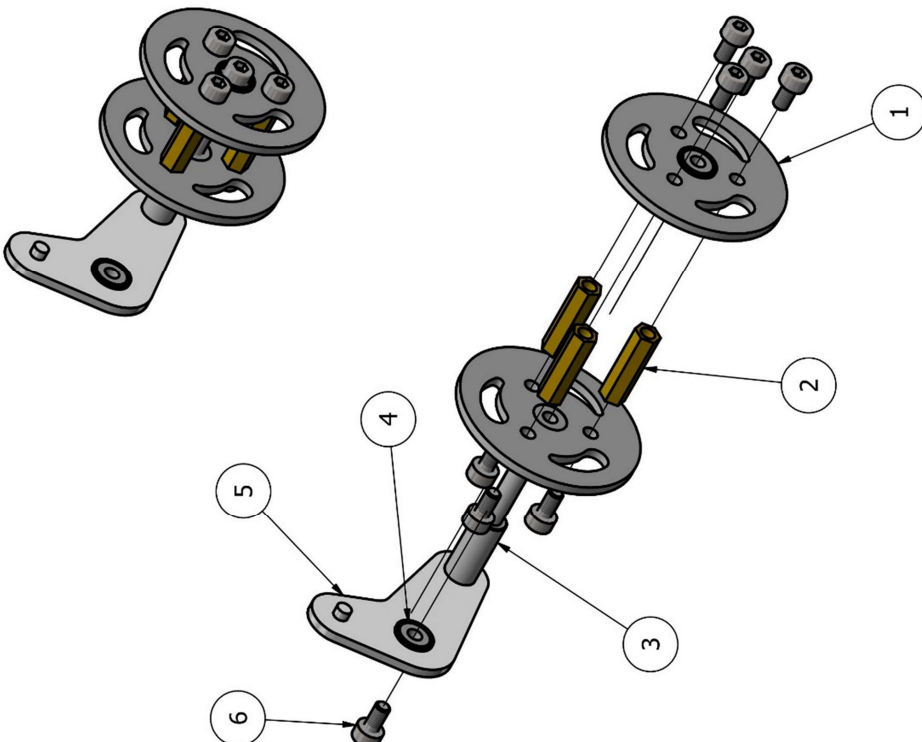



Anexo N° 9. Plano en explosión de robot explorador.



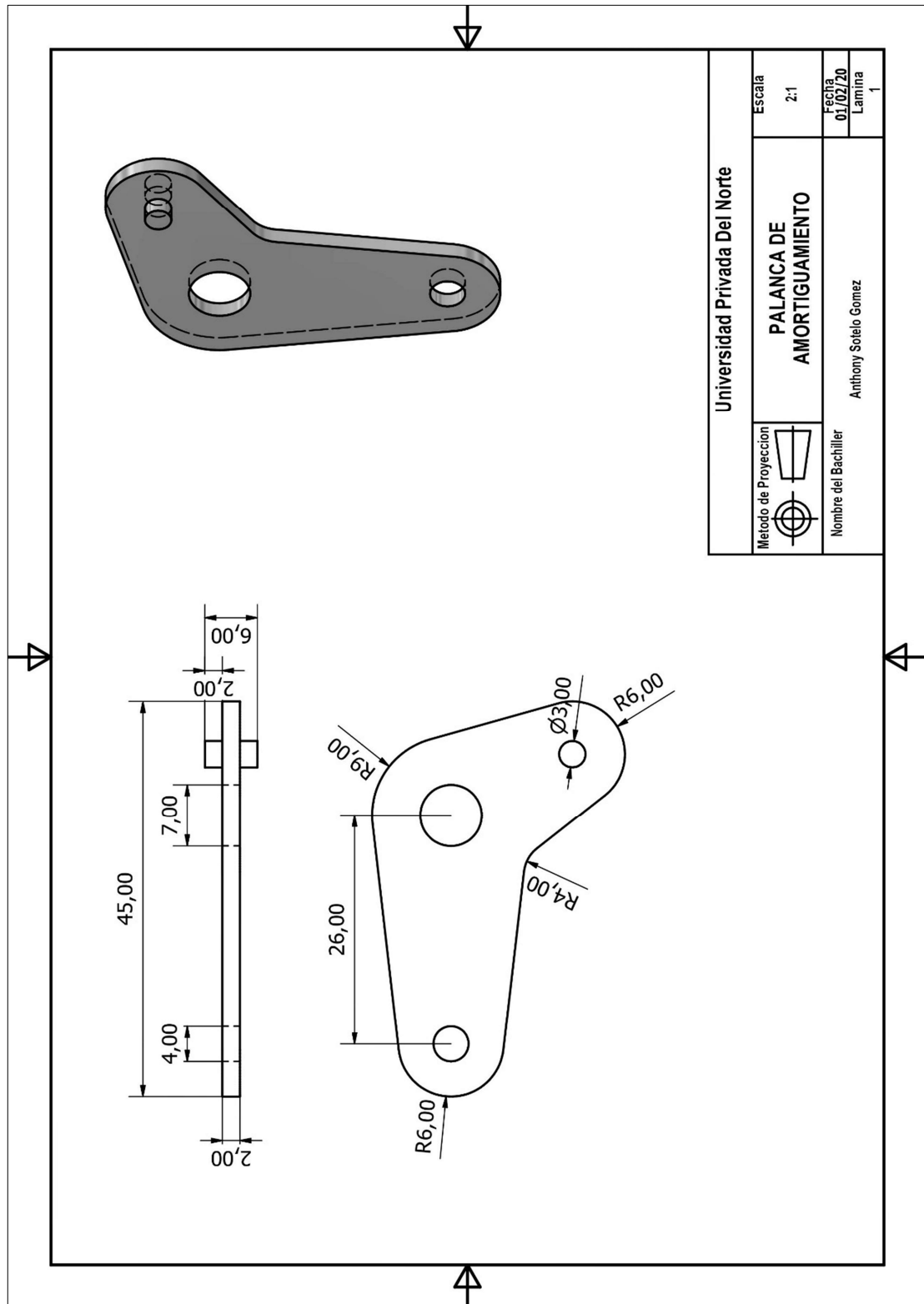
Anexo N° 10. Rueda de amortiguamiento.

LISTA DE PIEZAS DE RUEDA DE AMORTIGUAMIENTO		ART.	CANT.	N° DE PIEZA	DESCRIPCIÓN
1	2	Rueda	Rueda de Aluminio de 2 mm de espesor		
2	3	Separador Hexagonal	Metálico Hembra/Hembra a M3 18mm		
3	1	Eje Rueda	Acero 1018		
4	3	Rodamiento	Rodamiento Metálico 3 x 8 x 2.5 con brida		
5	1	Palanca de amortiguamiento	Aluminio 2 mm de espesor		
6	8	AS 1420 - 1973 - M3 x 6	Tornillos métricos ISO con cabeza cilíndrica con hueco hexagonal		

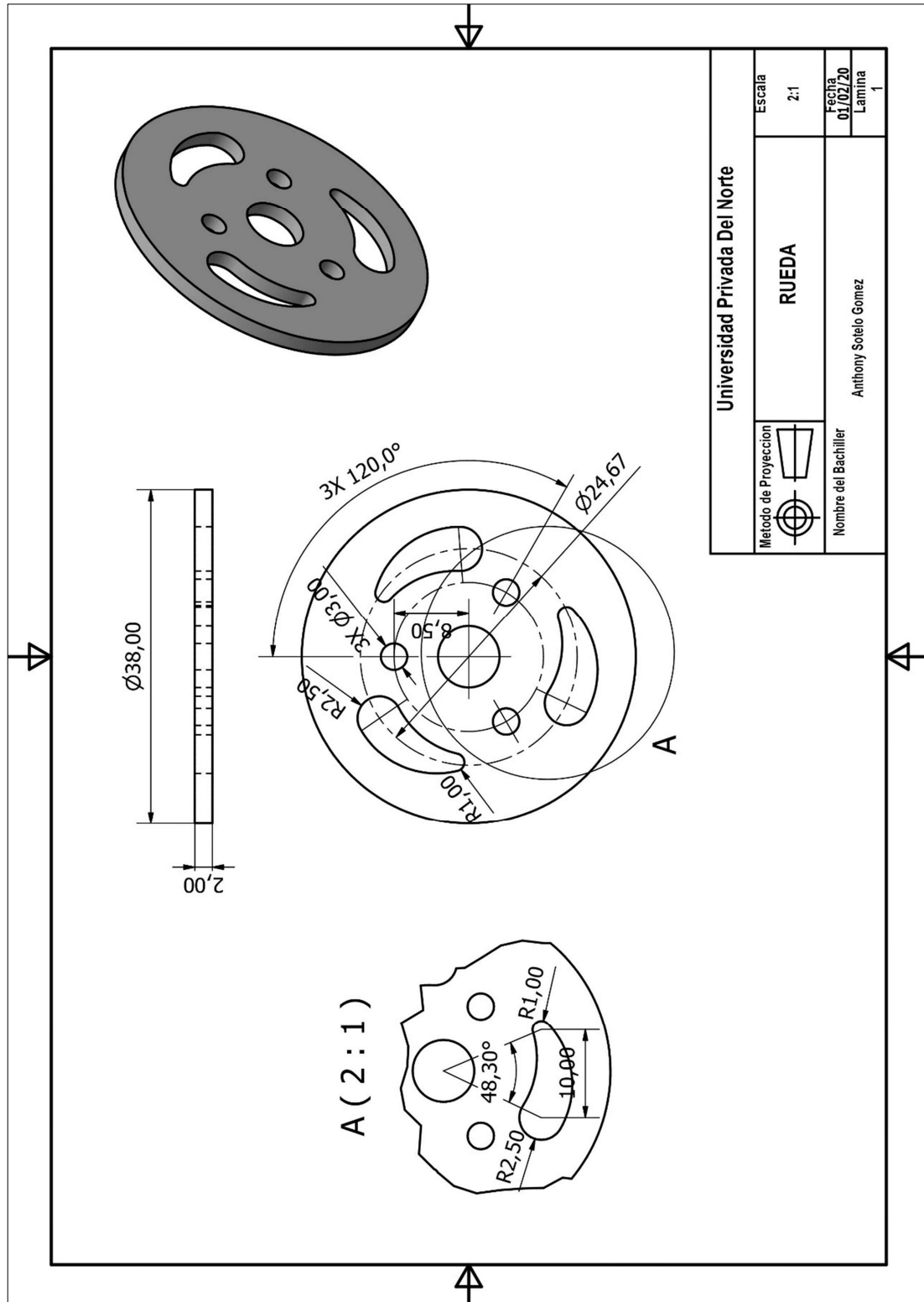
  

  

Universidad Privada Del Norte	
Metodo de Proyeccion 	Escala 1:1
<b>RUEDA DE AMORTIGUAMIENTO</b>	
Nombre del Bachiller Anthony Sotelo Gomez	Fecha 01/02/20 Lámina 1

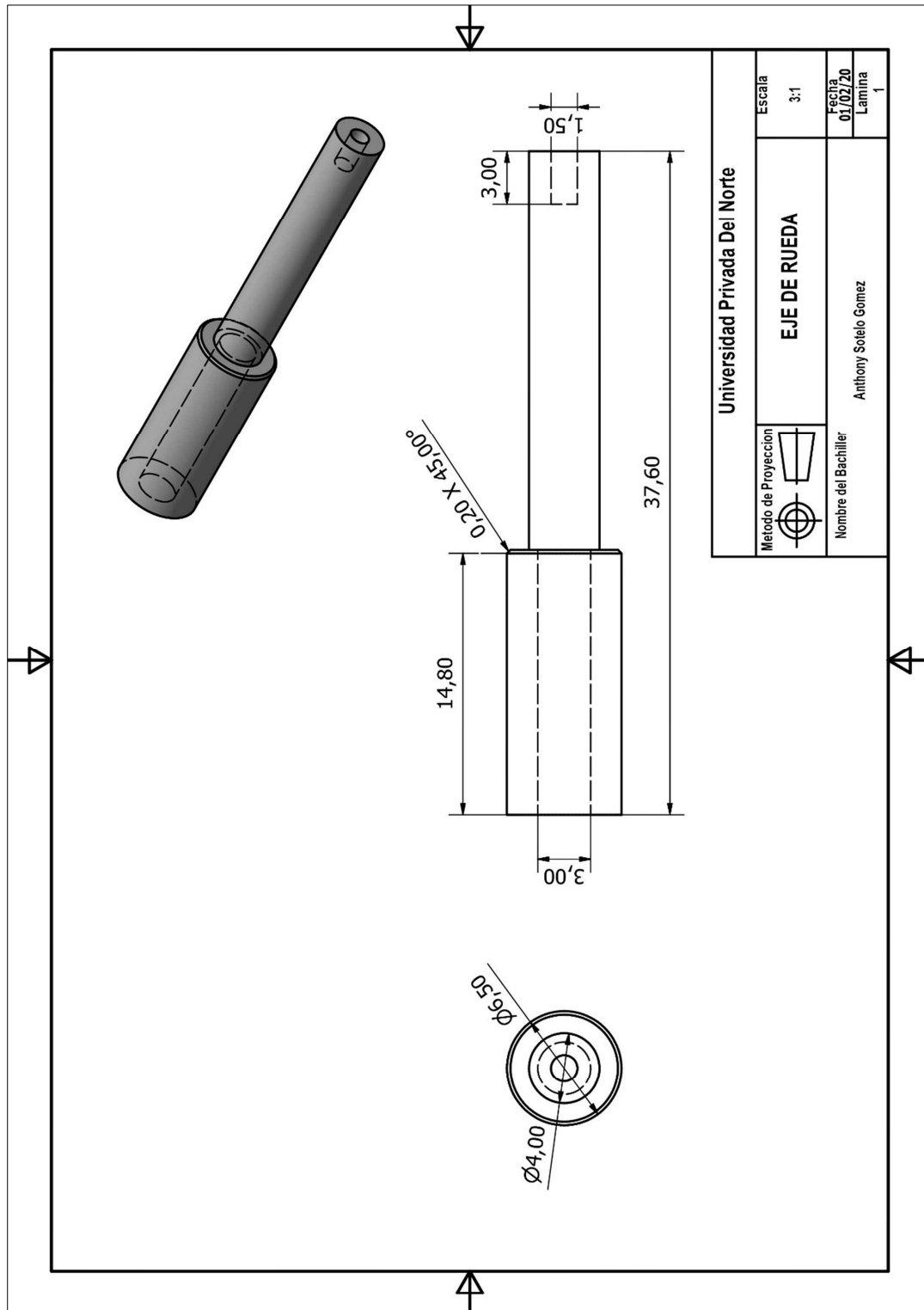
Anexo N° 11. Palanca de amortiguamiento.



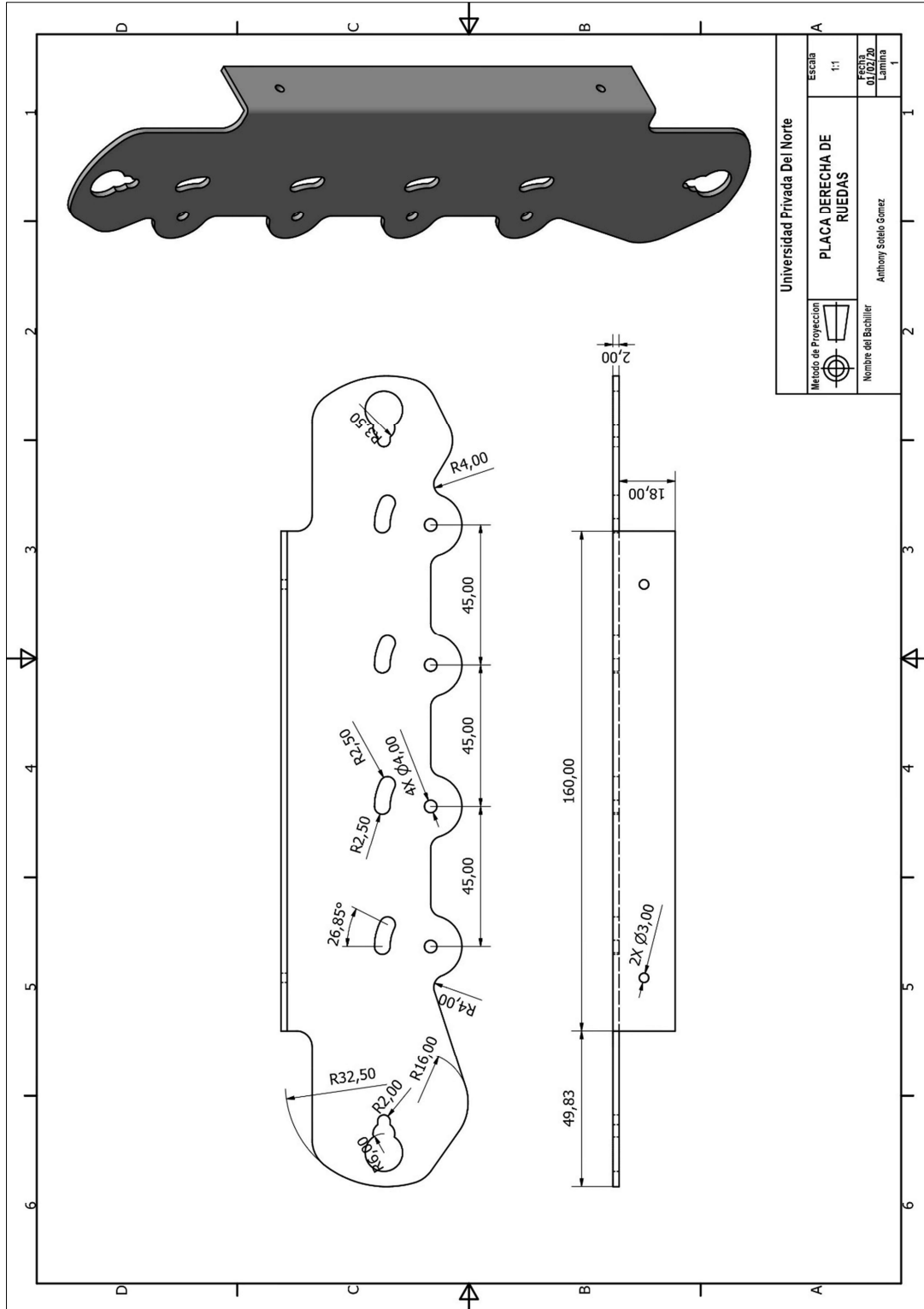
Anexo N° 12. Rueda.



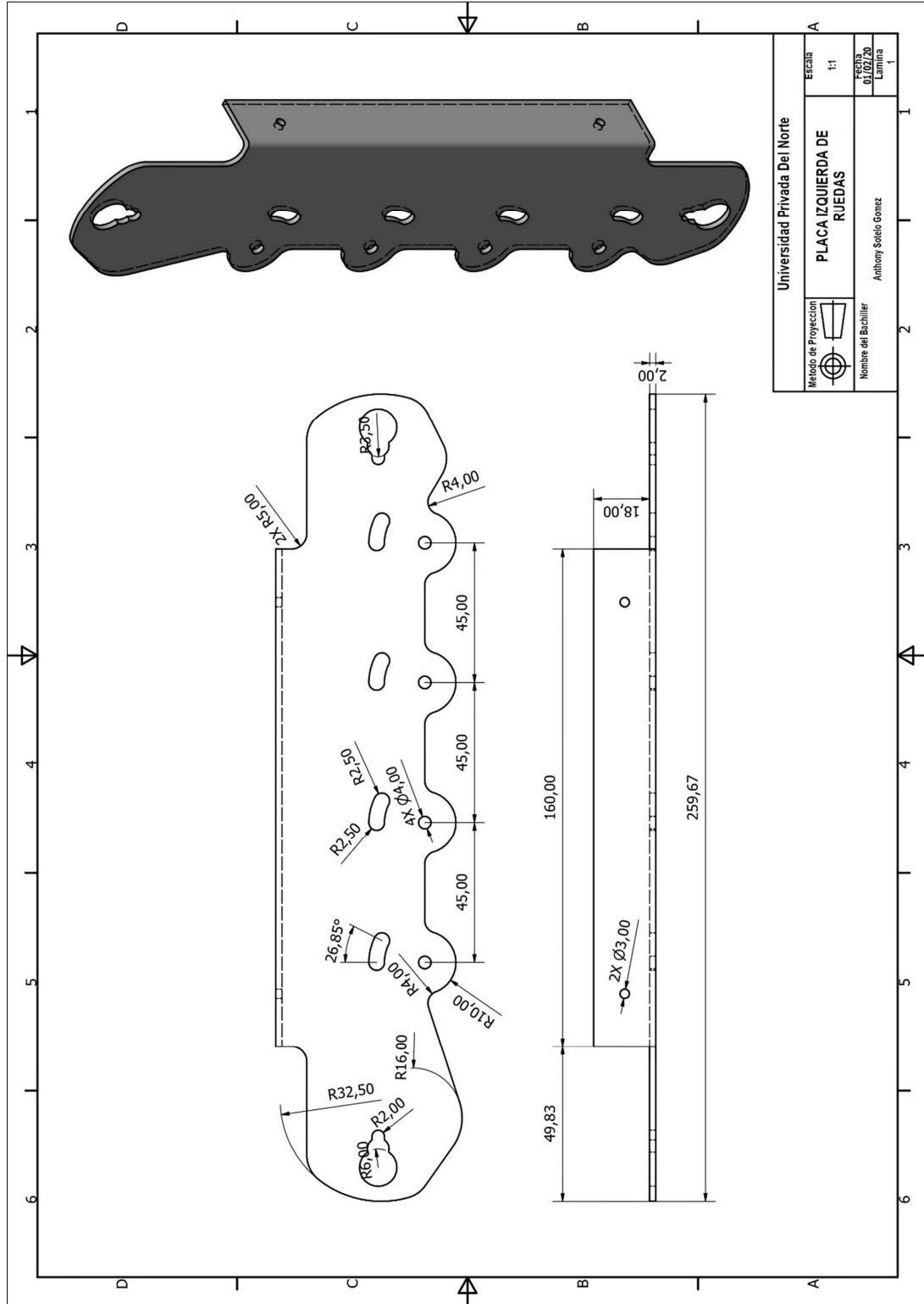
Anexo N° 13. Eje de rueda.



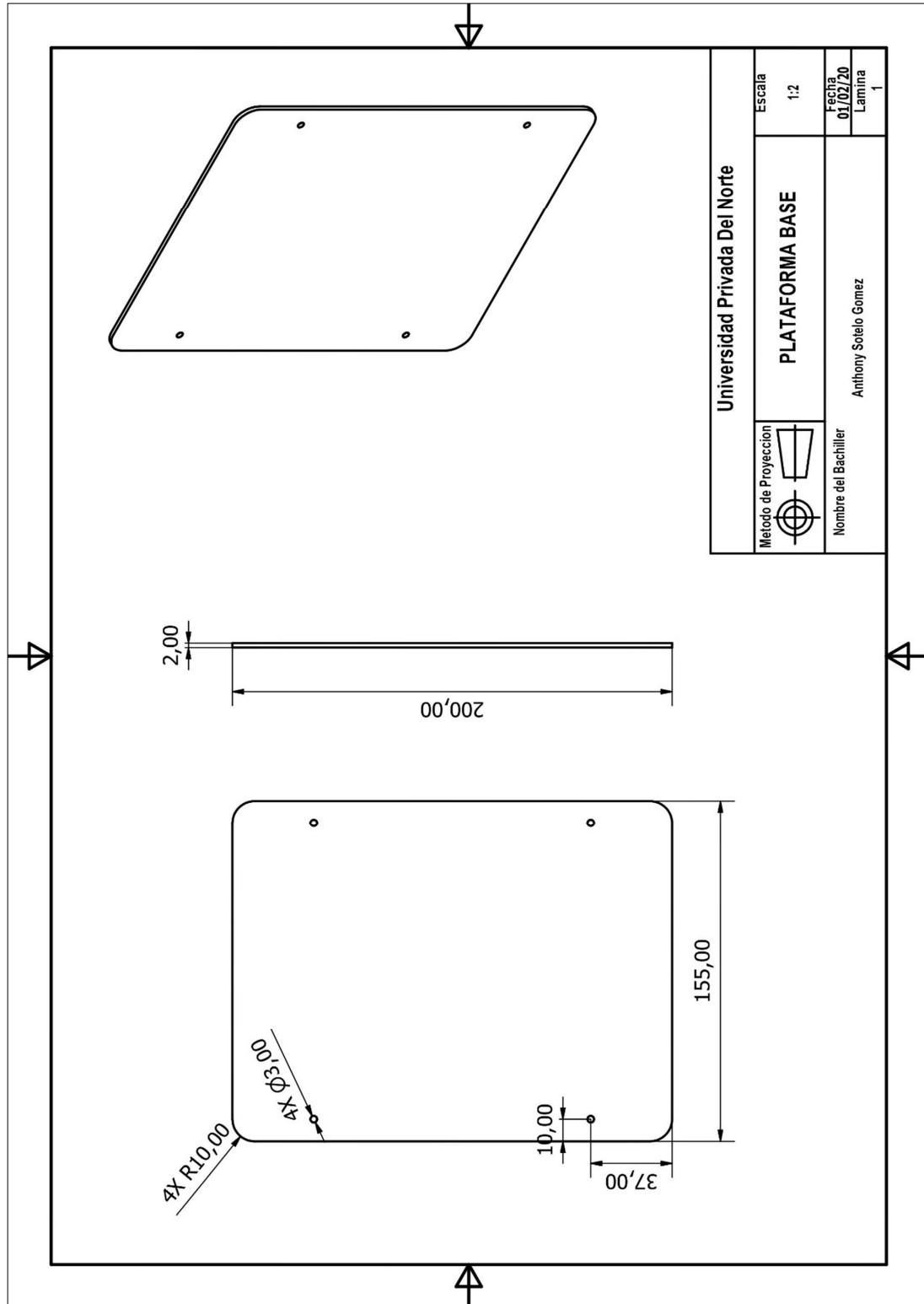
Anexo N° 14. Placa derecha de ruedas.



Anexo N° 15. Placa izquierda de ruedas.

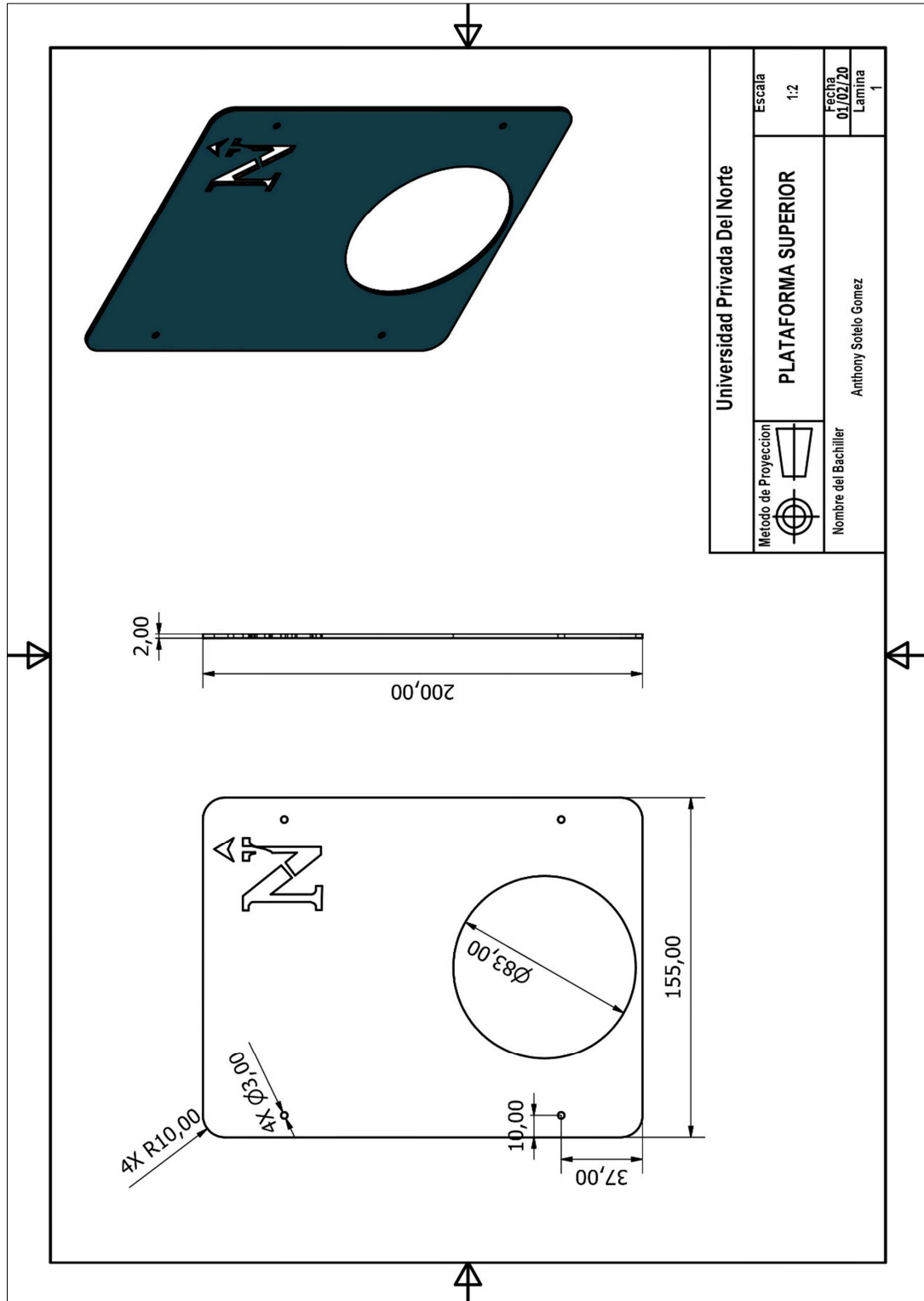


Anexo N° 16. Plataforma Base.

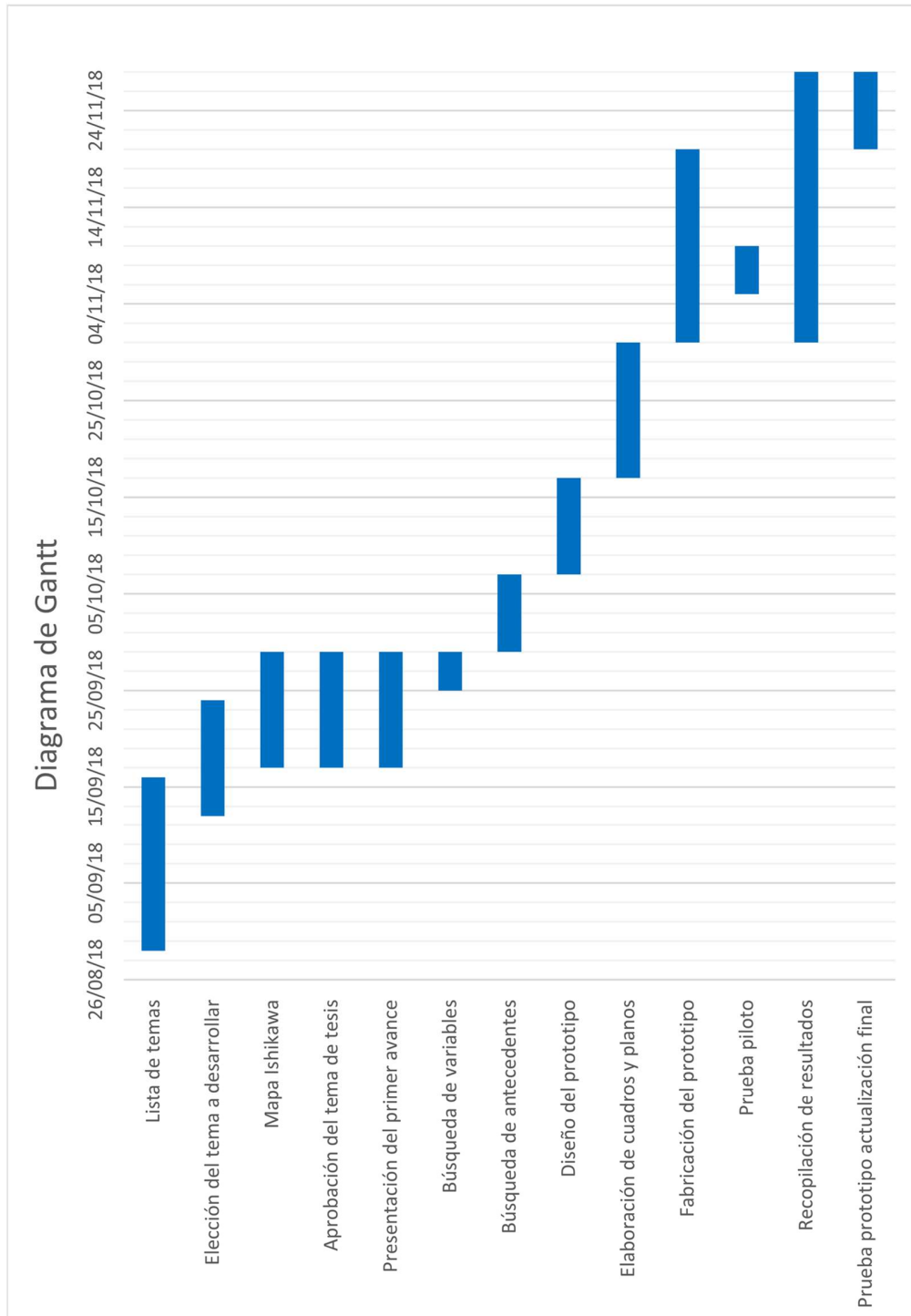




Anexo N° 17. Plataforma Superior.



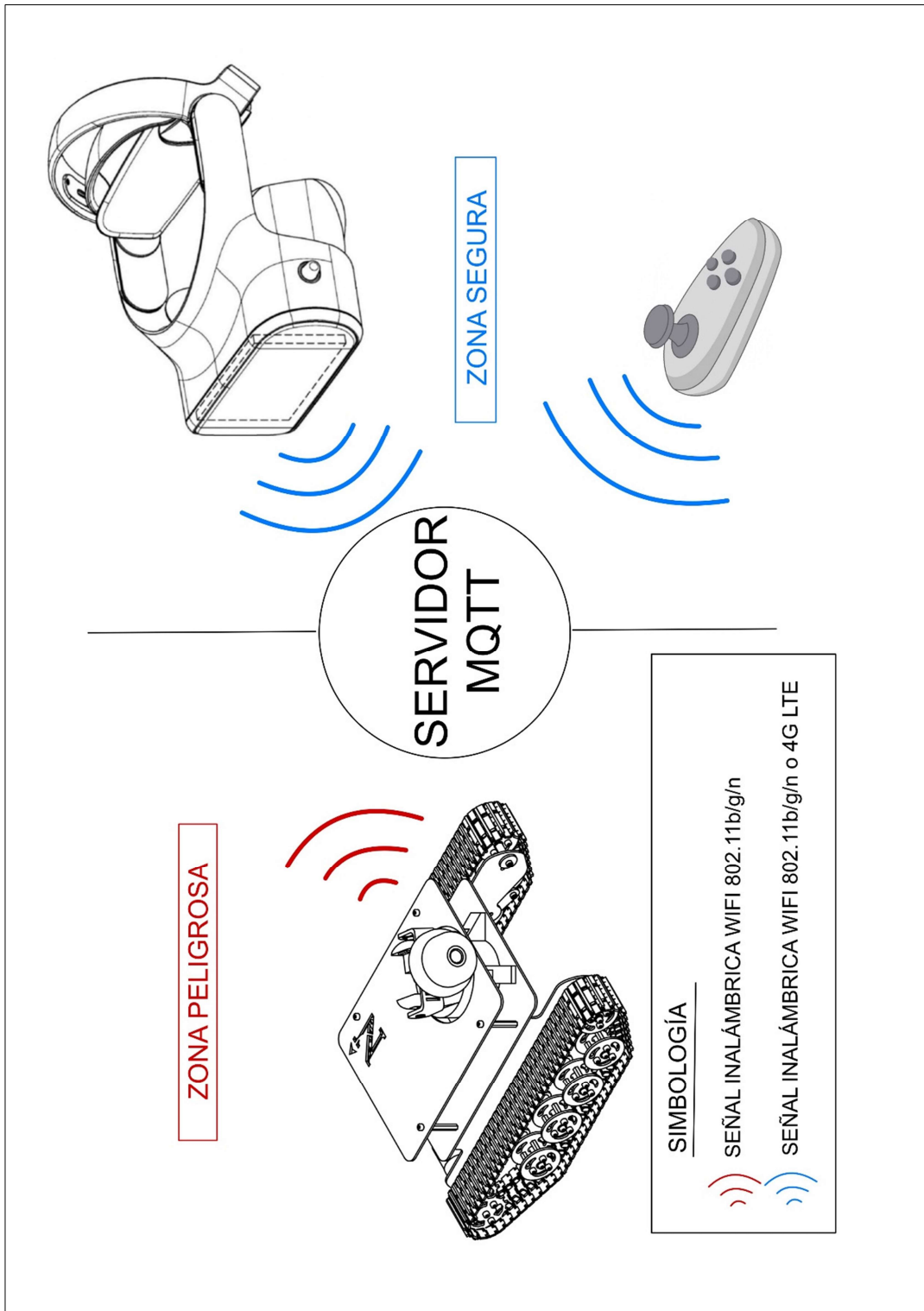
Anexo N° 17. Diagrama de Gantt.



Anexo N° 18. Presupuesto del proyecto de desarrollo

Componentes Mecánicos			
Cantidad	Ítem	Costo Unitario	Costo total
1	Minicámara IP	S/. 86.40	S/. 86.40
2	Importación de faja Oruga	S/. 55.35	S/. 110.70
1	Piezas Acrílicas + Corte laser	S/. 150.00	S/. 150.00
2	Impresión 3D	S/. 35.00	S/. 70.00
2	Motor DC XD-37GB520	S/. 33.49	S/. 66.98
2	Servomotor	S/. 20.00	S/. 40.00
2	Baterías de Litio	S/. 26.00	S/. 52.00
1	Modulo Cargador de baterías	S/. 4.00	S/. 4.00
3	Importación de Controlador NodeMCU	S/. 15.00	S/. 45.00
1	Fabricación de PCB	S/. 50.00	S/. 50.00
1	Visor VR	S/. 60.00	S/. 60.00
1	Joystick Analógico	S/. 12.00	S/. 12.00
1	Importación módulo MPU6050	S/. 4.00	S/. 4.00
1	Sirena tipo turbina	S/. 45.00	S/. 45.00
1	Módulo Relevador	S/. 8.00	S/. 8.00
1	Driver L298	S/. 8.00	S/. 8.00
	Otros Gastos		S/. 25.00
	<b>Total</b>		<b>S/. 837.08</b>

Anexo N° 19. Arquitectura de comunicación y control.



Anexo N° 20. Reporte de Análisis de Estrés Autodesk Inventor.

## Stress Analysis Report

Analyzed File:	PTZFinal.iam
Autodesk Inventor Version:	2018 (Build 220112000, 112)
Creation Date:	3/12/2020, 10:07 p. m.
Study Author:	ANTHONY SOTELO
Summary:	



### Project Info (iProperties)

#### Summary

Author ANTHONY SOTELO

#### Project

Part Number	PTZFinal
Designer	TONY
Cost	S/,0,00
Date Created	13/05/2018

#### Status

Design Status WorkInProgress

#### Physical

Mass	0,198249 kg
Area	73075,4 mm <sup>2</sup>
Volume	189967 mm <sup>3</sup>
Center of Gravity	x=5,14479 mm y=55,7092 mm z=52,4755 mm

Note: Physical values could be different from Physical values used by FEA reported below.

### Static Analysis:1

#### General objective and settings:

Design Objective	Single Point
Study Type	Static Analysis
Last Modification Date	3/12/2020, 9:45 p. m.
Detect and Eliminate Rigid Body Modes	No
Separate Stresses Across Contact Surfaces	No
Motion Loads Analysis	No

#### Mesh settings:

Avg. Element Size (fraction of model diameter)	0,1
Min. Element Size (fraction of avg. size)	0,2
Grading Factor	1,5
Max. Turn Angle	60 deg
Create Curved Mesh Elements	No
Use part based measure for Assembly mesh	Yes

#### Material(s)

Name	ABS Plastic	
General	Mass Density	1,06 g/cm <sup>3</sup>

	Yield Strength	20 MPa
	Ultimate Tensile Strength	29,6 MPa
Stress	Young's Modulus	2,24 GPa
	Poisson's Ratio	0,38 ul
	Shear Modulus	0,811594 GPa
Part Name(s)	BaseTilt Servomotorsg90 AspaServo SuperiorTilt Servomotorsg90 AspaServodos Brazo EjeTilt3 Rodaje PortaCamara Camara	
Name	Steel, Mild	
General	Mass Density	7.85 g/cm <sup>3</sup>
	Yield Strength	207 MPa
	Ultimate Tensile Strength	345 MPa
Stress	Young's Modulus	220 GPa
	Poisson's Ratio	0.275 ul
	Shear Modulus	86.2745 GPa
Part Name(s)	AS 1420 - Metric M3 x 6 AS 1420 - Metric M3 x 6 AS 1427 H - Metric M6 x 25	

**Operating conditions**

**Force:1**

Load Type	Force
Magnitude	1,470 N
Vector X	0,093 N
Vector Y	0,143 N
Vector Z	-1,460 N

**Selected Face(s)**





**Force:2**

Load Type	Force
Magnitude	0.074 N
Vector X	0.008 N
Vector Y	0.005 N
Vector Z	-0.073 N

**Selected Face(s)**



**Fixed Constraint:1**

Constraint Type Fixed Constraint

**Selected Face(s)**



**Contacts (Bonded)**

Name	Part Name(s)
Bonded:1	SuperiorTilt:1 EjeTilt3:1
Bonded:2	Brazo:1 Rodaje:1
Bonded:3	SuperiorTilt:1 EjeTilt3:1
Bonded:4	SuperiorTilt:1 Servomotorsg90:2
Bonded:5	SuperiorTilt:1 Servomotorsg90:2
Bonded:6	SuperiorTilt:1 Servomotorsg90:2
Bonded:7	SuperiorTilt:1 Servomotorsg90:2
Bonded:8	SuperiorTilt:1 Servomotorsg90:2
Bonded:9	SuperiorTilt:1 Servomotorsg90:2
Bonded:10	SuperiorTilt:1 Servomotorsg90:2
Bonded:11	SuperiorTilt:1



Bonded:12	SuperiorTilt:1 Servomotorsg90:2
Bonded:13	PortaCamara:1 Camara:1
Bonded:14	Brazo:1 PortaCamara:1
Bonded:15	Brazo:1 Rodaje:1
Bonded:16	BaseTilt:1 Servomotorsg90:1
Bonded:17	BaseTilt:1 Servomotorsg90:1
Bonded:18	BaseTilt:1 Servomotorsg90:1
Bonded:19	BaseTilt:1 Servomotorsg90:1
Bonded:20	BaseTilt:1 Servomotorsg90:1
Bonded:21	BaseTilt:1 Servomotorsg90:1
Bonded:22	AspaServodos:1 Brazo:1
Bonded:23	AspaServodos:1 Brazo:1
Bonded:24	AspaServodos:1 Brazo:1
Bonded:25	AspaServo:1 SuperiorTilt:1
Bonded:26	AspaServodos:1 Brazo:1
Bonded:27	AspaServo:1 SuperiorTilt:1
Bonded:28	AspaServodos:1 Brazo:1
Bonded:29	AspaServo:1 SuperiorTilt:1
Bonded:30	AspaServodos:1 Brazo:1
Bonded:31	AspaServo:1 SuperiorTilt:1
Bonded:32	AspaServodos:1 Brazo:1
Bonded:33	AspaServo:1 SuperiorTilt:1
Bonded:34	AspaServodos:1 Brazo:1
Bonded:35	AspaServo:1 SuperiorTilt:1
Bonded:36	AspaServodos:1 Brazo:1
Bonded:37	AspaServo:1 SuperiorTilt:1
Bonded:38	AspaServo:1 SuperiorTilt:1
Bonded:39	EjeTilt3:1 Rodaje:1
Bonded:40	AspaServodos:1 Brazo:1
Bonded:41	AspaServo:1 SuperiorTilt:1
Bonded:42	AspaServo:1 SuperiorTilt:1
Bonded:43	AspaServodos:1 Brazo:1
Bonded:44	AspaServo:1 SuperiorTilt:1
Bonded:45	AspaServodos:1 Brazo:1

Bonded:46	AspaServo:1 SuperiorTilt:1
Bonded:47	AspaServodos:1 Brazo:1
Bonded:48	AspaServo:1 SuperiorTilt:1
Bonded:49	AspaServodos:1 Brazo:1
Bonded:50	AspaServo:1 SuperiorTilt:1
Bonded:51	AspaServodos:1 Brazo:1
Bonded:52	AspaServo:1 SuperiorTilt:1
Bonded:53	AspaServodos:1 Brazo:1
Bonded:54	AspaServo:1 SuperiorTilt:1
Bonded:55	AspaServodos:1 Brazo:1
Bonded:56	AspaServo:1 SuperiorTilt:1
Bonded:57	Servomotorsg90:2 AspaServodos:1
Bonded:58	Servomotorsg90:1 AspaServo:1
Bonded:59	Servomotorsg90:2 AspaServodos:1
Bonded:60	Servomotorsg90:1 AspaServo:1

## Results

### Reaction Force and Moment on Constraints

Constraint Name	Reaction Force		Reaction Moment	
	Magnitude	Component (X,Y,Z)	Magnitude	Component (X,Y,Z)
Fixed Constraint:1	1.54389 N	-0.101309 N	0.020289 N m	0.0168193 N m
		-0.148295 N		-0.0113469 N m
		1.53341 N		-0.000050479 N m

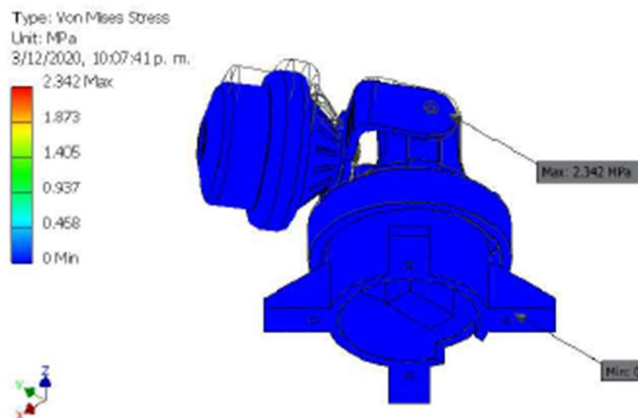
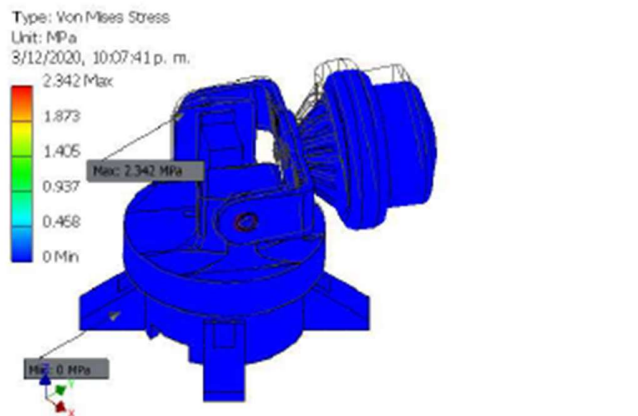
### Result Summary

Name	Minimum	Maximum
Volume	188772 mm <sup>3</sup>	
Mass	0.200098 kg	
Von Mises Stress	0.00000431095 MPa	2.34171 MPa
1st Principal Stress	-0.424434 MPa	1.90571 MPa
3rd Principal Stress	-1.71328 MPa	0.695162 MPa
Displacement	0 mm	0.243271 mm
Safety Factor	8.54076 ul	15 ul
Stress XX	-1.03047 MPa	0.96383 MPa
Stress XY	-0.333445 MPa	0.403477 MPa
Stress XZ	-0.500448 MPa	0.578077 MPa
Stress YY	-0.980934 MPa	1.3054 MPa
Stress YZ	-0.552499 MPa	1.16035 MPa
Stress ZZ	-1.17958 MPa	1.31432 MPa
X Displacement	-0.0377046 mm	0.0877243 mm
Y Displacement	-0.0217933 mm	0.105208 mm
Z Displacement	-0.234231 mm	0.0356536 mm
Equivalent Strain	0.0000000177062 ul	0.000961775 ul
1st Principal Strain	0.000000000522043 ul	0.000885058 ul
3rd Principal Strain	-0.000771416 ul	-0.00000000154787 ul
Strain XX	-0.0004605 ul	0.000443041 ul
Strain XY	-0.000205426 ul	0.000248571 ul
Strain XZ	-0.000308312 ul	0.000356137 ul

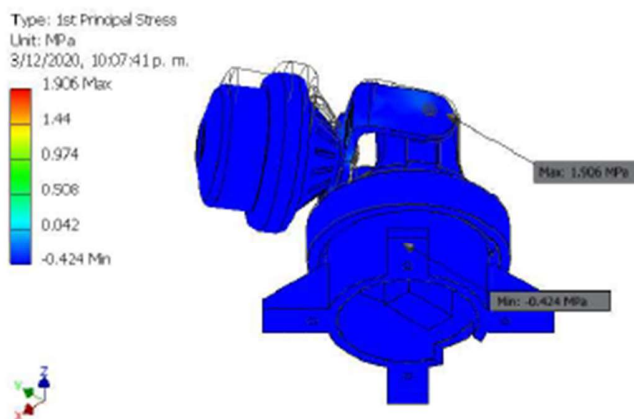
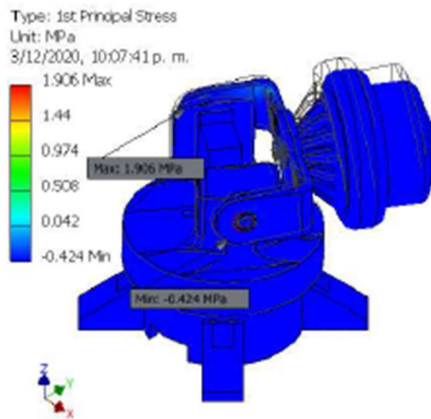
Strain YY	-0.000324392 ul	0.000317258 ul
Strain YZ	-0.000340379 ul	0.000714858 ul
Strain ZZ	-0.000358014 ul	0.000412257 ul
Contact Pressure	0 MPa	5.41936 MPa
Contact Pressure X	-2.61459 MPa	2.03072 MPa
Contact Pressure Y	-2.49096 MPa	2.44509 MPa
Contact Pressure Z	-5.37172 MPa	3.13765 MPa

## Figures

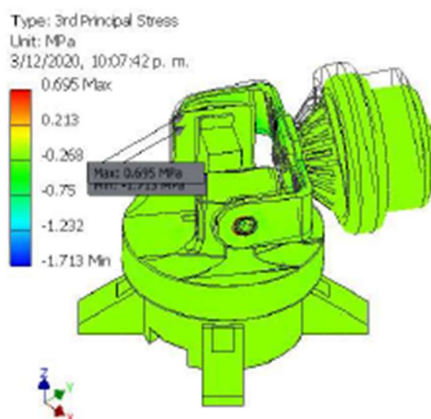
### Von Mises Stress

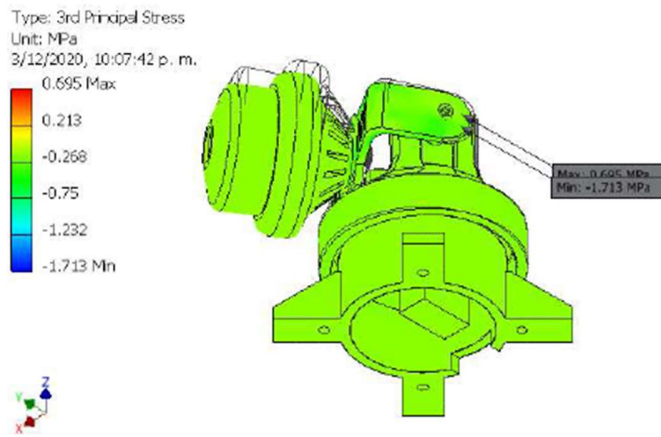


### 1st Principal Stress

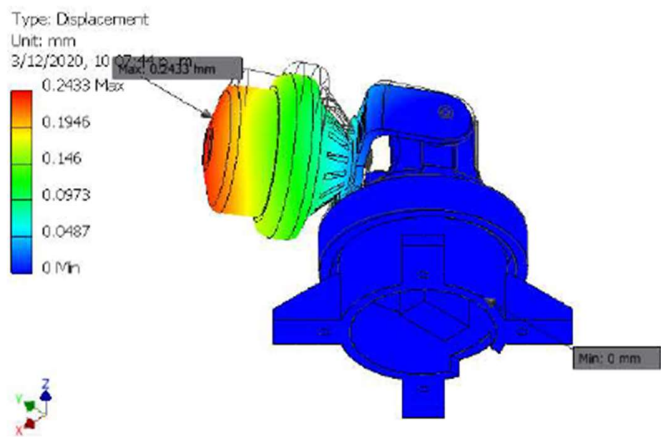
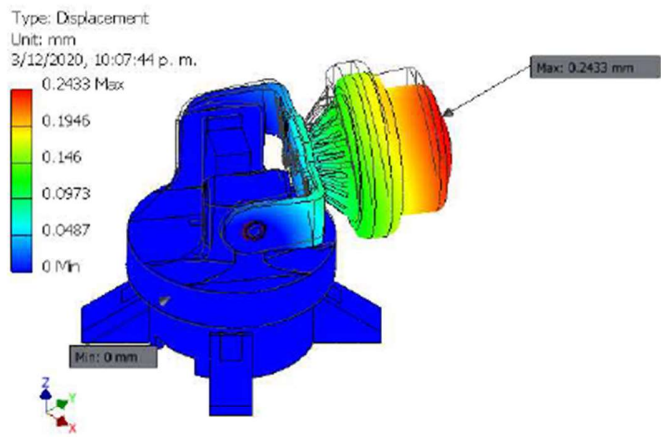


### 3rd Principal Stress

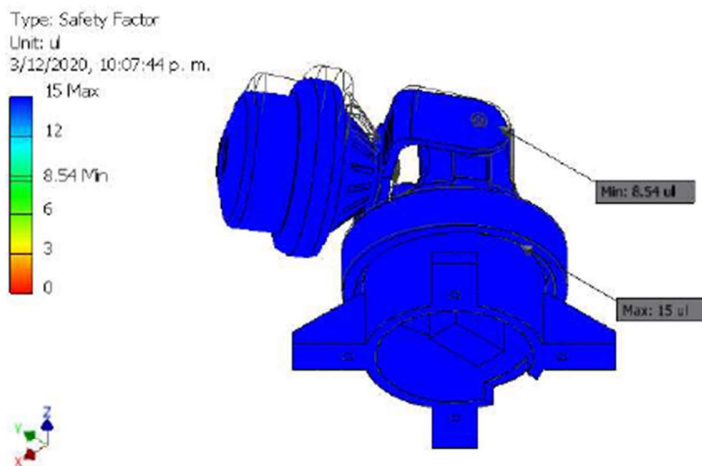
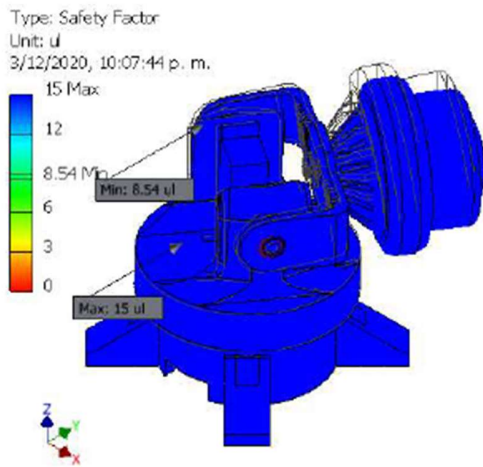




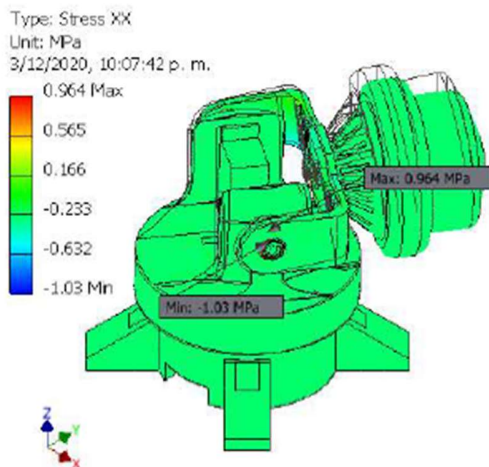
**Displacement**



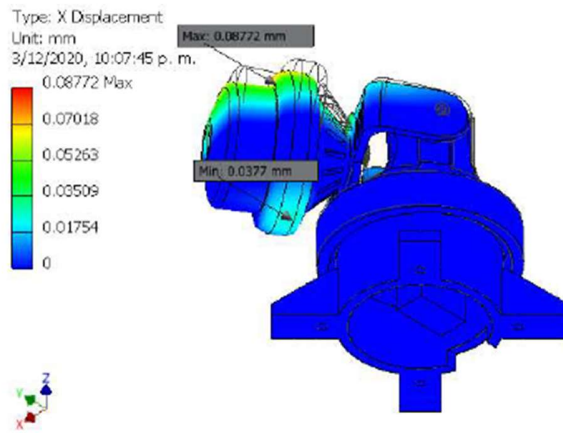
**Safety Factor**



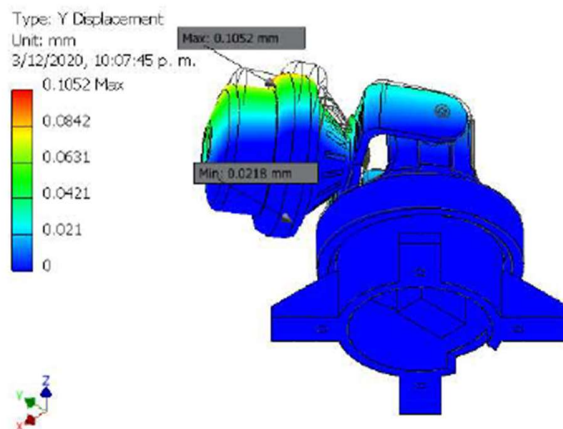
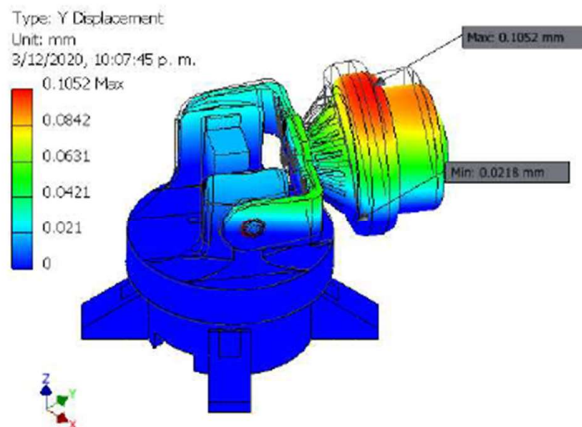
### **Stress XX**



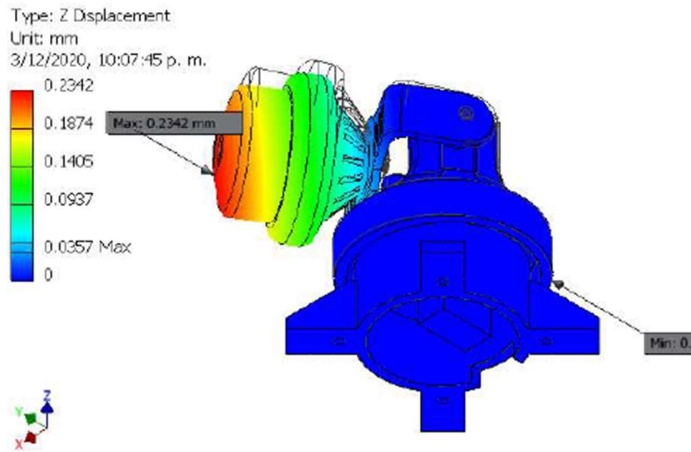
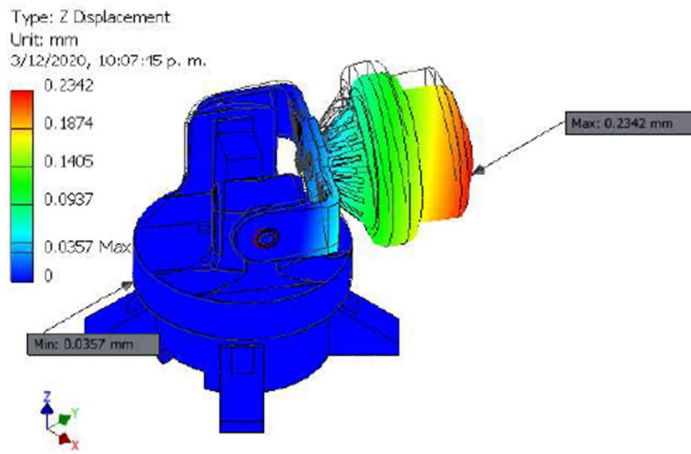




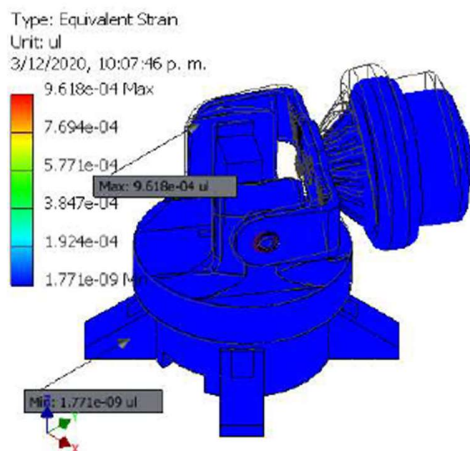
### Y Displacement



### Z Displacement



### Equivalent Strain





## ☐ Static Analysis:2

### General objective and settings:

Design Objective	Single Point
Study Type	Static Analysis
Last Modification Date	3/12/2020, 10:07 p. m.
Detect and Eliminate Rigid Body Modes	No
Separate Stresses Across Contact Surfaces	No
Motion Loads Analysis	No

### Mesh settings:

Avg. Element Size (fraction of model diameter)	0.1
Min. Element Size (fraction of avg. size)	0.2
Grading Factor	1.5
Max. Turn Angle	60 deg
Create Curved Mesh Elements	No
Use part based measure for Assembly mesh	Yes

## ☐ Material(s)

Name	ABS Plastic	
General	Mass Density	1.06 g/cm <sup>3</sup>
	Yield Strength	20 MPa
	Ultimate Tensile Strength	29.6 MPa
Stress	Young's Modulus	2.24 GPa
	Poisson's Ratio	0.38 ul
	Shear Modulus	0.811594 GPa
Part Name(s)	BaseTilt Servomotorsg90 AspaServo SuperiorTilt Servomotorsg90 AspaServodos Brazo EjeTilt3 PortaCamara Camara	
Name	Aluminum 6061	
General	Mass Density	2.7 g/cm <sup>3</sup>
	Yield Strength	275 MPa
	Ultimate Tensile Strength	310 MPa
Stress	Young's Modulus	68.9 GPa
	Poisson's Ratio	0.33 ul
	Shear Modulus	25.9023 GPa
Part Name(s)	Rodaje	

AS 1420 - Metric M3 x 6
AS 1420 - Metric M3 x 6
AS 1427 H - Metric M6 x 25

## ☐ **Operating conditions**

### ☐ **Force:1**

Load Type	Force
Magnitude	5.000 N
Vector X	0,317 N
Vector Y	0.487 N
Vector Z	-4.966 N