



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA INGENIERÍA INDUSTRIAL

**“CONSTRUCCIÓN DE UN PROTOTIPO DE ASISTENCIA EN LA
CONDUCCIÓN DE UN AUTOMÓVIL PARA PERSONAS CON
PARAPLEJIA BASADO EN UN SISTEMA DE CONTROL
ELECTROMECAÁNICO”**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO INDUSTRIAL

AUTOR:

PAÚL GEOVANNY JUIÑA PULI

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA INGENIERÍA INDUSTRIAL

**“CONSTRUCCIÓN DE UN PROTOTIPO DE ASISTENCIA EN LA
CONDUCCIÓN DE UN AUTOMÓVIL PARA PERSONAS CON
PARAPLEJIA BASADO EN UN SISTEMA DE CONTROL
ELECTROMECAÁNICO”**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO INDUSTRIAL

AUTOR: PAÚL GEOVANNY JUIÑA PULI

DIRECTOR: Ing. EDUARDO FRANCISCO GARCÍA CABEZAS Mg.

Riobamba – Ecuador

2023

© 2023, **Paúl Geovanny Juiña Puli**

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

Yo, PAÚL GEOVANNY JUIÑA PULI, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 26 de enero del 2023



Paúl Geovanny Juiña Puli

C.I: 172413034-7

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA INGENIERÍA INDUSTRIAL

El Tribunal del Trabajo de Integración Curricular certifica que: el Trabajo de Integración Curricular; Tipo: Proyecto Técnico, “**CONSTRUCCIÓN DE UN PROTOTIPO DE ASISTENCIA EN LA CONDUCCIÓN DE UN AUTOMÓVIL PARA PERSONAS CON PARAPLEJIA BASADO EN UN SISTEMA DE CONTROL ELECTROMECAÁNICO**”, realizado por el señor: **PAÚL GEOVANNY JUIÑA PULI**, ha sido minuciosamente revisado por los miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos y legales; en tal virtud el Tribunal autoriza su presentación.

	FIRMA	FECHA
Ing. Carlos José Santillán Mariño PRESIDENTE DEL TRIBUNAL		2023 – 01 – 26
Ing. Eduardo Francisco García Cabezas DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023 – 01 – 26
Ing. Ángel Geovanny Guamán Lozano ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023 – 01 – 26

DEDICATORIA

El presente trabajo de titulación se lo dedico a mis padres Luis y Beatriz, por ser mi ejemplo por seguir, por todo su apoyo y amor, recordándome que puedo conseguir todo lo que me propongo en la vida. A mis hermanos Marisol, William, Luis, Giselle y mis sobrinos quienes me motivan a mejorar todos los días. A la familia Pérez Lozada y de manera especial a mi novia Valeria que siempre estuvo presente para brindarme su amor y apoyo incondicional.

Paúl

AGRADECIMIENTO

Mi más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo por haberme brindado los conocimientos necesarios para formarme como un gran profesional con principios éticos y morales. A los docentes de la escuela de Ingeniería Industrial por compartir sus conocimientos y ayudar en mi formación tanto académica como personal, de manera especial al Ing. Eduardo García e Ing. Ángel Guamán Lozano que con su apoyo y conocimientos han sido los guías para cumplir con éxito el trabajo de titulación. Agradecido con mi familia, la mayor motivación e inspiración para conseguir esta gran meta.

Paúl

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE ILUSTRACIONES	xiii
ÍNDICE DE ECUACIONES	xv
ÍNDICE DE ANEXOS	xvi
RESUMEN	xvi
SUMMARY	xviii
INTRODUCCIÓN	1

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA	2
1.1. Antecedentes	2
1.2. Planteamiento del problema	2
1.3. Justificación	3
1.4. Delimitación geográfica	4
1.5. Objetivos	6
1.5.1. <i>Objetivo general</i>	6
1.5.2. <i>Objetivos específicos</i>	6

CAPÍTULO II

2. Fundamentos teóricos	7
2.1. Paraplejia.....	7
2.1.1. <i>Paraplejia a nivel mundial</i>	7
2.1.2. <i>Personas con discapacidad física en Ecuador</i>	7
2.1.3. <i>Paraplejia en la ciudad de Riobamba</i>	8
2.2. Automóvil	9
2.2.1. <i>Tipos de automóviles según su carrocería</i>	9
2.2.1.1. <i>Sedán</i>	9
2.2.1.2. <i>Hatchback</i>	10
2.2.1.3. <i>SUV</i>	10
2.2.1.4. <i>Camioneta</i>	11
2.3. Adaptaciones para automóviles	12
2.3.1. <i>Volante</i>	12

2.3.1.1.	<i>Acelerador electrónico de aro concéntrico</i>	12
2.3.1.2.	<i>Acelerador mecánico de aro con freno a la mano (conjunto)</i>	13
2.3.2.	Control de pedales	14
2.3.2.1.	<i>Leva de freno vertical</i>	14
2.3.2.2.	<i>Leva vertical de acelerador y freno</i>	14
2.3.3.	Sistema total driving	15
2.4.	Pedales del automóvil	15
2.4.1.	<i>Acelerador</i>	15
2.4.2.	<i>Freno</i>	17
2.4.3.	<i>Embrague</i>	18
2.4.3.1.	<i>Posición de embrague</i>	18
2.4.3.2.	<i>Posición de desembrague</i>	18
2.4.3.3.	<i>Fase transitoria</i>	19
2.5.	Motores paso a paso	20
2.5.1.	<i>Motores bipolares y unipolares</i>	21
2.5.2.	<i>Ventajas y desventajas de los motores paso a paso</i>	22
2.6.	Software CAD	23
2.6.1.	<i>SolidWorks</i>	23
2.7.	Sistemas de control	23
2.8.	Arduino	25
2.8.1.	<i>Principales placas Arduino</i>	26
2.9.	Raspberry pi	27
2.9.1.	<i>Modelos de raspberry</i>	28
2.9.2.	<i>Raspberry pi OS</i>	30
2.10.	Sistema simulador cockpit	30
2.11.	Impresión 3D	31
2.11.1.	<i>Tipos de impresión 3D</i>	31
2.11.1.1.	<i>FDM</i>	31
2.11.1.2.	<i>Resina</i>	32
2.12.	Ángulos de confort	32

CAPÍTULO III

3.	Marco metodológico	35
3.1.	Definición de los requerimientos	35
3.2.	Consolidación del prototipo	36
3.3.	Arquitectura del sistema	37

3.3.1.	<i>Definición del algoritmo de conducción</i>	38
3.3.2.	<i>Entorno de experimentación</i>	41
3.3.3.	<i>Actuadores</i>	46
3.3.3.1.	<i>Driver digital DM556Y</i>	50
3.3.4.	<i>Driver HY-DIV268N-5A</i>	51
3.3.5.	<i>Accesorios para los motores</i>	52
3.3.6.	<i>Integración de los actuadores al diseño del cockpit</i>	54
3.3.7.	<i>Elementos para el control manual</i>	56
3.3.8.	<i>Fuentes conmutadas 12 V - 10 A y 24V - 10A</i>	57
3.3.9.	<i>Guantes para conducción</i>	57
3.3.10.	<i>Raspberry pi</i>	58
3.3.11.	<i>Arduino</i>	59
3.3.12.	<i>Distribución de los elementos de control</i>	61
3.4.	Ensamble completo del prototipo	64
3.5.	Algoritmos de control	65
3.5.1.	<i>Funcionamiento general del prototipo</i>	67
3.5.2.	<i>Algoritmo para la interfaz gráfica</i>	68
3.5.3.	<i>Algoritmo de control con arduino</i>	69

CAPÍTULO IV

4.	Resultados	71
4.1.	Ángulos de confort para un puesto de conducción	71
4.2.	Análisis estático	74
4.2.1.	<i>Análisis estructural</i>	74
4.2.2.	<i>Análisis estático – estructura cockpit</i>	74
4.2.3.	<i>Análisis estático – accesorio para los pedales</i>	77
4.3.	Interfaz gráfica	83
4.4.	Pruebas	83
4.4.1.	<i>Distancias y velocidades</i>	83
4.4.2.	<i>Embrague</i>	84
4.4.3.	<i>Freno</i>	85
4.4.4.	<i>Acelerador</i>	86
4.4.5.	<i>Consumo energético</i>	86
4.4.6.	<i>Embrague</i>	87
4.4.7.	<i>Freno</i>	87
4.4.8.	<i>Acelerador</i>	88

4.5.	Instrucciones de uso - manual de usuario	89
4.6.	Gestión del proyecto.....	90
4.6.1.	<i>Cronograma</i>	90
4.6.2.	<i>Costos</i>	90
4.6.3.	<i>Recurso humano</i>	91
4.6.4.	<i>Recurso material</i>	92
	CONCLUSIONES.....	93
	RECOMENDACIONES.....	95
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2:	Adaptaciones de acuerdo con la discapacidad.....	12
Tabla 2-2:	Placas arduino	27
Tabla 3-2:	Placas raspberry y sus características	28
Tabla 4-2:	Ángulos de confort para un puesto de conducción	33
Tabla 1-3:	Medias antropométricas para un puesto de conducción	42
Tabla 2-3:	Características del asiento	44
Tabla 3-3:	Torques	49
Tabla 4-3:	Especificaciones Eléctricas	49
Tabla 5-3:	Especificaciones Físicas.....	49
Tabla 6-3:	Especificaciones eléctricas.....	50
Tabla 7-3:	Entorno operativo y otras especificaciones	50
Tabla 8-3:	Especificaciones eléctricas y físicas.....	51
Tabla 9-3:	Accesorios para el motor	52
Tabla 10-3:	Dimensiones del piso de las principales carrocerías.....	54
Tabla 11-3:	Comparativa de los tipos de impresión 3D.....	55
Tabla 12-3:	Fuentes conmutadas.....	57
Tabla 13-3:	Comparativa entre Raspberry Pi modelo 3 B+ y 4.....	58
Tabla 14-3:	Comparativa entre la placa de Arduino UNO y MEGA.....	60
Tabla 15-3:	Tabla de Conexiones.....	62
Tabla 16-3:	Ensamble	65
Tabla 1-4:	Ángulos obtenidos de una persona de 165 cm de estatura.....	71
Tabla 2-4:	Ángulos obtenidos de una persona de 170 cm de estatura	72
Tabla 3-4:	Ángulos obtenidos de una persona de 175 cm de estatura	73
Tabla 4-4:	Tabla comparativa de resultados	73
Tabla 5-4:	Propiedades mecánicas	75
Tabla 6-4:	Puntos de sujeción	75
Tabla 7-4:	Cargas	76
Tabla 8-4:	Tensiones.....	76
Tabla 9-4:	Desplazamientos	77
Tabla 10-4:	Propiedades volumétricas	78
Tabla 11-4:	Propiedades mecánicas	78
Tabla 12-4:	Punto de sujeción.....	79
Tabla 13-4:	Cargas.....	79
Tabla 14-4:	Análisis de tensiones.....	80

Tabla 15-4:	Análisis de desplazamiento	81
Tabla 16-4:	Análisis de deformaciones unitarias.....	82
Tabla 17-4:	Distancia lineal de recorrido del embrague	84
Tabla 18-4:	Distancia lineal del recorrido del acelerador y el freno	84
Tabla 19-4:	Tiempos y velocidades del embrague	84
Tabla 20-4:	Tiempos y velocidades del freno.....	85
Tabla 21-4:	Tiempos y velocidades del acelerador.....	86
Tabla 22-4:	Consumo del motor para el embrague en stand by, avance y retroceso.....	87
Tabla 23-4:	Consumo del motor para el freno en stand by, avance y retroceso.....	87
Tabla 24-4:	Consumo del motor para el acelerador en stand by, avance y retroceso.....	88
Tabla 25-4:	Cronograma de actividades	90
Tabla 26-4:	Costos	91

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1:	Delimitación grafica.....	5
Ilustración 2-1:	Escuela Superior Politécnica de Chimborazo	5
Ilustración 1-2:	Porcentajes de personas con discapacidad física por edades.....	8
Ilustración 2-2:	Porcentajes de personas con discapacidad física en la ciudad de Riobamba.....	9
Ilustración 3-2:	Automóvil tipo sedán	10
Ilustración 4-2:	Automóvil tipo hatchback o compacto	10
Ilustración 5-2:	Automóvil tipo SUV	11
Ilustración 6-2:	Automóvil tipo camioneta	11
Ilustración 7-2:	Acelerador electrónico de aro.....	13
Ilustración 8-2:	Acelerador mecánico de aro con freno a la mano	13
Ilustración 9-2:	Leva de freno vertical.....	14
Ilustración 10-2:	Leva vertical de acelerador y freno.....	14
Ilustración 11-2:	Sistema total driving.....	15
Ilustración 12-2:	Sistema del acelerador.....	16
Ilustración 13-2:	Sistema de freno	17
Ilustración 14-2:	Posición de embrague.....	18
Ilustración 15-2:	Posición de desembrague	19
Ilustración 16-2:	Fase transitoria	19
Ilustración 17-2:	Motor paso a paso	21
Ilustración 18-2:	Tipos de motores paso a paso	22
Ilustración 19-2:	Sistema de control de lazo abierto	24
Ilustración 20-2:	Sistema de control de lazo cerrado	24
Ilustración 21-2:	Partes de la placa Arduino	26
Ilustración 22-2:	Raspberry pi 1 A+	27
Ilustración 23-2:	Cockpit simracing	30
Ilustración 24-2:	Impresora 3D FDM	31
Ilustración 25-2:	Impresora 3D DLP o resina.....	32
Ilustración 26-2:	Ángulos de confort.....	33
Ilustración 1-3:	Consolidación del prototipo	36
Ilustración 2-3:	Diagrama de bloques de la arquitectura del sistema	37
Ilustración 3-3:	Diagrama de flujo del proceso de conducción	39
Ilustración 4-3:	Proceso de conducción de un automóvil de transmisión manual	40

Ilustración 5-3:	Medidas antropométricas para un puesto de conducción.....	41
Ilustración 6-3:	Pedales	42
Ilustración 7-3:	Volante.....	43
Ilustración 8-3:	Medidas generales de “Asiento”.....	44
Ilustración 9-3:	Medidas de la estructura base	45
Ilustración 10-3:	Cockpit.....	46
Ilustración 11-3:	Dinamómetro.....	47
Ilustración 12-3:	Fuerza ejercida y distancia de los pedales.	48
Ilustración 13-3:	Motores Nema 23	50
Ilustración 14-3:	Driver Digital	51
Ilustración 15-3:	Driver HY-DIV268N	52
Ilustración 16-3:	Base de los motores.....	54
Ilustración 17-3:	Accesorio	55
Ilustración 18-3:	Distribución de los motores.....	56
Ilustración 19-3:	Guantes para conducción.....	58
Ilustración 20-3:	Raspberry pi 3 B+	59
Ilustración 21-3:	Placa de Arduino UNO.....	60
Ilustración 22-3:	Esquema de conexiones.....	61
Ilustración 23-3:	Diagrama de conexiones.....	63
Ilustración 24-3:	Distribución de los pulsadores.....	64
Ilustración 25-3:	Distribución de los dispositivos de control.....	64
Ilustración 26-3:	Prototipo ensamblado.....	65
Ilustración 27-3:	Interfaz de Arduino IDE.....	66
Ilustración 28-3:	Interfaz de Qt creator.....	66
Ilustración 29-3:	Secuencia de funcionamiento del prototipo	68
Ilustración 30-3:	Diagrama de flujo de la interfaz gráfica	69
Ilustración 31-3:	Diagrama de flujo de la programación en Arduino.....	70
Ilustración 1-4:	Persona de 165cm – A5.....	71
Ilustración 2-4:	Persona de 170cm – A5.....	72
Ilustración 3-4:	Persona de 175cm – A5.....	73
Ilustración 4-4:	Interfaz gráfica para configurar el prototipo	83
Ilustración 5-4:	Gráfico de comparación de velocidad y tiempo.....	85
Ilustración 6-4:	Gráfico de comparación de velocidad y tiempo.....	85
Ilustración 7-4:	Gráfico de comparación de velocidad y tiempo.....	86
Ilustración 8-4:	Consumo del motor para el embrague	87
Ilustración 9-4:	Consumo del motor para el freno	88
Ilustración 10-4:	Consumo del motor para el acelerador	88

ÍNDICE DE ECUACIONES

Ecuación 1-3:	Fórmula de torque	46
Ecuación 2-3:	Fuerza aplicada en los pedales	47

ÍNDICE DE ANEXOS

ANEXO A: PLANOS

ANEXO B: ENSAMBLE COMPLETO

ANEXO C: PROGRAMACIÓN DE INTERFAZ GRÁFICA

ANEXO D: PROGRAMACIÓN EN ARDUINO

RESUMEN

El presente trabajo tuvo como objetivo la construcción de un prototipo de asistencia en la conducción de un automóvil para personas con paraplejia basado en un sistema de control electromecánico. El estudio bibliográfico brindó información sobre la paraplejia y sus implicaciones físicas, datos estadísticos de las personas con discapacidad en el país y la provincia de Chimborazo, los sistemas de asistencia de la conducción existentes en el mercado, y principios de funcionamiento y elementos de control del automóvil y entornos de simulación. Se desarrolló un algoritmo de conducción para automóviles de transmisión manual y se diseñó en SolidWorks una estructura *cockpit* como entorno de experimentación de la conducción. El prototipo partió de la selección de elementos mecánicos y electromecánicos, al que se incorporó un accesorio impreso en 3D que permitirá el control automatizado. Se realizó un análisis estático de la estructura evaluando parámetros de tensiones, desplazamientos y deformaciones unitarias; se aplicó una fuerza de 925 Newtons y se deduce que la estructura de acero al carbono laminado en frío no tendrá afectación a esta fuerza, dado que, el límite elástico crítico es mayor ($3,5e+08$ N/m²). Se desarrolló una interfaz gráfica sobre la plataforma de *raspberry* para la interacción del usuario que permitió la configuración del prototipo. Se realizaron pruebas posturales midiendo ángulos de confort en personas de 165, 170 y 175 centímetros de estatura, dando como resultado que las personas de 175 cm pueden llegar a sentir incomodidad al usar el prototipo. Se concluye que el uso de elementos *opensource* es fundamental para el desarrollo del prototipo y se recomienda la revisión del procedimiento seguro de utilización del prototipo para garantizar el correcto uso y la seguridad del usuario.

Palabras clave: <PROTOTIPO> <SISTEMA DE CONTROL ELECTROMECAÁNICO> <PARAPLEJIA> <INTERFAZ GRÁFICA> <ESTRUCTURA COCKPIT> <ALGORITMO DE CONDUCCIÓN> <RASPBERRY>.

0633-DBRA-UPT-2023



SUMMARY

The aim of this work was the prototypical structure of a car driver assistance system for paraplegics based on an electromechanical control. The bibliographic study provided information on paraplegia and its physical consequences, statistical data on people with disabilities in the country and in the province of Chimborazo, on existing driver assistance systems on the market, and on operating principles and control elements of the car and simulation environments. A driving algorithm was developed for manual transmission cars and a *cockpit* structure was designed in SolidWorks as an environment for driving experiments. The prototype was based on the selection of mechanical and electromechanical elements into which a 3D-printed accessory was integrated to enable automated control. A static analysis of the structure was performed evaluating parameters of stresses, displacements, and unit deformations; A force of 925 Newtons was applied and it is concluded that the cold rolled carbon steel structure is not affected by this force as the critical elastic limit is larger ($3,5e+08 \text{ N/m}^2$). A graphical interface for user interaction was developed on the Raspberry platform, which the prototype could be configured. Finally, postures tests were conducted by measuring comfort angles in subjects 165, 170, 175 cm tall, with the result that subjects 175 cm tall may feel discomfort when using the prototype. It is concluded that the use of *opensource* elements is fundamental for the development of the prototype and it is recommended to review the safe procedure for using the prototype to ensure the correct use and the safety of the user.

Keywords: <AUTOMATED CONTROL>, <GRAPHICAL INTERFACE>, <ALGORITHM>, <PARAPLEGIA>, <RASPERRY>



Lic. Angela Cecibel Moreno Novillo
0602603938

INTRODUCCIÓN

Cada día se busca crear nuevas formas de ayudar a personas que sufren algún tipo de discapacidad a que se integren de mejor manera a la sociedad. Constantemente el país trabaja en el desarrollo de tecnologías para mejorar la calidad de vida de estas personas, han sido realizados grandes avances al incorporar el uso de elementos electrónicos y herramientas digitales.

Las personas con algún grado de paraplejia pueden requerir asistencia para realizar actividades de la vida diaria. Dependiendo de la gravedad de la condición, existen diferentes problemáticas a las que se enfrentan. En el mercado hay una gran variedad de dispositivos o equipos especiales que les permitan realizar cual actividad con mayor independencia y de las cuales se pueden beneficiar, dentro de esto se incluyen sillas de ruedas, andadores, camas articuladas, automóviles adaptados y otros equipos especiales.

Considerando que en el país no existen automóviles adecuados para que las personas con paraplejia los puedan conducir, complicando la movilidad de estas, nace la ambición de contribuir con una idea para la movilización eficaz y la conducción segura de un automóvil por parte de personas que no cuentan con un adecuado funcionamiento en sus extremidades inferiores. Con la aplicación de los conocimientos adquiridos en el transcurso de la carrera, como es el caso del CAD, estudios de ergonomía, procesos de manufactura y automatización el presente trabajo de titulación se enfoca en la construcción de un prototipo para asistencia en la conducción para personas con paraplejia, brindando la facilidad de realizar una conducción confiable y segura, el prototipo construido con un sistema de control electromecánico basado en Arduino y una interfaz gráfica diseñada en Raspberry, permite el movimiento independiente de los pedales mediante un control electrónico ubicado en sitios estratégicos que posibilita el control de dirección, cambio de velocidades, manejo de panel de instrumentos y demás accesorios del automóvil. La asistencia en la conducción es una solución para mejorar la calidad de vida de las personas con paraplejia, ya que les permite una mayor independencia y autonomía.

Dentro de este documento se pone a disposición la investigación sobre el estado del arte, los planos del diseño de la estructura, cálculos realizados para la selección de materiales, construcción del prototipo, conexiones eléctricas, después constarán las pruebas, resultados y conclusiones del desarrollo del prototipo de conducción. Todo esto puede ser usado para el perfeccionamiento del prototipo así también como para futuras investigaciones.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Antecedentes

Aproximadamente medio millón de personas al año alrededor del mundo sufren lesiones en la columna vertebral. Esas personas tienen una mayor posibilidad de morir tempranamente, siendo los países subdesarrollados quienes registran las tasas más bajas de supervivencia. En un informe de la OMS, se muestra un resumen de los datos actuales disponibles sobre las causas y las medidas de prevención de las lesiones medulares, también se puede encontrar información sobre la asistencia necesaria para estos pacientes. Los varones son más propensos a sufrir estas lesiones cuando se encuentran entre los 20 y los 29 años, y también a partir de los 70 años, mientras las mujeres tienen el riesgo entre los 15 y los 19 años, y de los 60 años en adelante (Organización mundial de la Salud, 2014, p. 5).

De acuerdo con datos del INEC, 453 ecuatorianos tuvieron lesiones medulares en 2014, el 42.38% afectó a jóvenes de 25 a 44 años, 88 personas tuvieron diagnóstico de paraplejia o tetraplejia, de estos el 66.7% son hombres y el 33.3% corresponde a mujeres. El Consejo Nacional para la igualdad de Discapacidades nos indica que, en el país en el año 2017, existieron casi medio millón de discapacitados; y se sabe que aproximadamente 65 000 son económicamente activos, en el país la población económicamente activa comprende a personas de 15 años o más que trabajaron al menos 1 hora en la semana o, aunque no trabajaron, tuvieron trabajo; y personas que no tenían empleo pero estaban disponibles para trabajar y buscan empleo (Instituto Nacional de Estadísticas y censos INEC, 2014 citado en Vivar, 2021, p. 6).

Mediante un informe el INEC también indica que el 55% de las personas discapacitadas del país que cuentan con un trabajo, tienen una discapacidad física. La inclusión en el campo de laboral no alcanza ni al 25% de las personas con capacidades especiales que se encuentran en edad para trabajar, existen dificultades y desafíos para el estado, porque tienen que cumplir con políticas públicas de inserción laboral de los discapacitados en el país.

1.2. Planteamiento del problema

El avance tecnológico en el sector automovilístico se ha visto enfocado en la optimización de los procesos de construcción y fabricación de los automóviles, así mismo en desarrollar mayor confort para el usuario, haciendo de lado a sectores de la población con impedimentos físicos que

tienen dificultades para adaptarse a la conducción tradicional. Actualmente los automóviles adaptados basan sus diseños en mecanismos pensados para personas con paraplejía en los que se hace necesario el uso palancas, estos sistemas están pensados únicamente para automóviles de transmisión automática en los que la conducción es más sencilla, estos sistemas son únicamente mecánicos y es indispensable emplear cierta fuerza para ser accionados correctamente, hasta ahora no se ha trabajado pensando en la transmisión manual.

La baja cantidad de automóviles adaptados para la conducción de personas con paraplejía, los elevados costos de fabricación, aranceles a las importaciones, limitaciones en las leyes del país, motivaron la construcción de este prototipo basado en un control electromecánico, con pulsadores ubicados en los dedos de las manos que accionaran los motores que actúan sobre los pedales de un entorno de experimentación.

Muchas personas con capacidades especiales tienen el problema, de que por su condición están excluidas de ciertas actividades, si una persona quiere postular a un puesto de trabajo que relacione alguna actividad de conducción, esta persona se ve descartada porque las empresas, por el alto costo que representa una adaptación a los automóviles no quieren incluirlos en la nómina de personal de esta parte de la cadena de producción.

En vista de la identificación de estos problemas lo que se plantea en este trabajo es la generación de un sistema basado en un sistema electromecánico, con el cual se pretende disminuir costos, que las adaptaciones en los autos sean sencillas y de esta manera contribuir a la inclusión social las personas con esta condición.

1.3. Justificación

Históricamente las personas con paraplejía han sido excluidas en la ejecución de determinadas tareas, afectando seriamente su calidad de vida. La razón de esta exclusión se ha dado principalmente porque no se tienen en cuenta como prioridad sus necesidades, haciéndolos un sector olvidado, la urgencia de trasladarnos de manera rápida y eficaz crea un impacto positivo en la sociedad.

En nuestro país, es indispensable el apoyo y la creación de nuevas formas de movilidad que planteen la implementación de un campo automotor eficiente sin dejar de lado al sector poblacional con discapacidades, en el cual se presentan diferentes patologías relacionadas con la

pérdida de movimiento de las extremidades inferiores que imposibilita la independencia de las personas.

Los automóviles adaptados para personas con discapacidad han estado siempre presentes en el desarrollo de la industria automotriz como una solución a los problemas de movilidad que tienen estas personas, sin embargo, tienen un costo muy alto y no todas las personas pueden acceder a uno, es por esto que se apuesta por la construcción de prototipos que se adapten a cualquier automóvil que sean amigables con el usuario, más asequibles, cómodos y puedan dar cumplimiento a las necesidades de personas con discapacidades en sus extremidades inferiores.

El proyecto de crear un sistema de asistencia en la conducción de automóviles para personas con paraplejía basado en el uso de un control electromecánico es con la finalidad de mejorar calidad de vida de estas personas. El uso de este prototipo puede ayudar en su inclusión laboral, puesto que tendrían más oportunidades de trabajar no solo empresas de servicios sino también en industrias formando parte de la cadena de suministro. La aplicación de los conceptos aprendidos en la carrera como lo es el CAD, el estudio de modelos ergonómicos, los procesos de manufactura y la automatización de procesos fueron fundamentales en el desarrollo del trabajo.

1.4. Delimitación geográfica

País: Ecuador

Región: Interandina

Provincia: Chimborazo

Ciudad: Riobamba

Dirección: Panamericana Sur km 1 1/2

Zona de desarrollo: Central N° 3

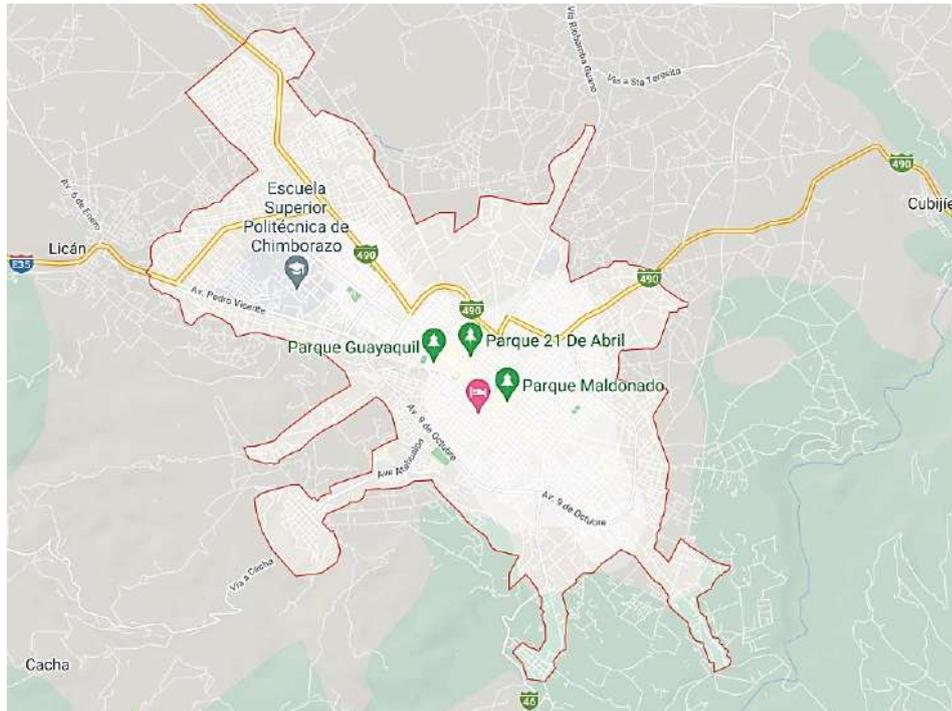


Ilustración 1-1: Delimitación grafica

Fuente: <https://goo.su/YKKdP>

El cantón Riobamba es una entidad territorial subnacional ecuatoriana, de la Provincia de Chimborazo. Su cabecera cantonal es la ciudad de Riobamba, lugar donde se agrupa la mayor parte de su población total. Se localiza al centro de la región Interandina del Ecuador, cerca del centro geográfico del país, rodeada de varios volcanes como el Chimborazo, el Tungurahua, el Altar y el Carihuairazo; en la hoya del Río Chambo, en la orilla izquierda del río Chibunga, a una altitud de 2750 metros sobre el nivel del mar.



Ilustración 2-1: Escuela Superior Politécnica de Chimborazo

Fuente: <https://goo.su/bUKal>

La Escuela Superior Politécnica de Chimborazo es una institución ecuatoriana de educación superior, con sede central en la ciudad de Riobamba, Ecuador. Desde 2012 pertenece a la Red Ecuatoriana de Universidades para Investigación y Postgrados.

1.5. Objetivos

1.5.1. Objetivo general

Construir un prototipo de asistencia en la conducción de un automóvil para personas con paraplejia basado en un sistema de control electromecánico.

1.5.2. Objetivos específicos

- Realizar un estudio bibliográfico sobre sistemas de asistencia de conducción desarrollados a través de estudios y sobre aquellos que se encuentran disponibles en el mercado para el planteamiento de requerimientos para el prototipo.
- Determinar mediante el modelado en una herramienta CAD las adecuaciones que deben vincularse, mecánicas, los elementos electromecánicos, electrónicos y software para la construcción del prototipo de asistencia de conducción.
- Determinar mediante un estudio metodológico los elementos electromecánicos y electrónicos adecuados para adaptar la conducción de un automóvil de transmisión manual.
- Definir el software adecuado para el desarrollo del algoritmo de control y una interfaz gráfica que permita interactuar al usuario con el equipo.
- Comprobar la resistencia de la estructura y el funcionamiento del sistema electromecánico.

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

2.1. Paraplejia

Es un tipo de parálisis en la cual el movimiento de la parte inferior del cuerpo se ve disminuido. Las personas con esta condición tienen imposibilitado el movimiento voluntario de los pies, las piernas e inclusive en algunos casos el abdomen. Hasta el momento la paraplejia no tiene una cura, pero se puede facilitar la vida de estas personas con tratamientos y terapias (Medical News Today, 2020). Se identifican dos tipos de paraplejia, la completa que es la de mayor gravedad. En estos casos, el paciente tiene movilidad en las extremidades inferiores e incluso ciertas partes del abdomen. Además de estar inmóviles y perder completamente el control, los pacientes no tienen sensibilidad en el área paralizada, lo que significa que no sienten roce, presión (Weatherspoon & Kandola, 2020, párr. 1-2) y la paraplejia parcial que es aquella en la que el paciente aún posee cierta capacidad para mover las extremidades aunque esta sea limitada. En este caso se pueden presentar también problemas digestivos y sexuales, debilidad en las extremidades pero conservando algo de movilidad y sensibilidad. Estas lesiones permiten algún grado de independencia; se pueden mejorar con fisioterapia y ortopedia (Moreno & Amaya, 2009, p. 185).

2.1.1. Paraplejia a nivel mundial

De acuerdo con la (Organización Mundial de la Salud, 2013, párr. 1-6) entre 250 000 y 500 000 personas aproximadamente sufren al año algún tipo de lesión en la médula espinal provocado por un traumatismo efecto de algún accidente, caída o por alguna enfermedad degenerativa como el cáncer o la esclerosis múltiple, no hay datos precisos pero se estima que la incidencia puede oscilar entre 40 y 80 personas por cada millón de habitantes, poco más del 10% de los casos se deben a experiencias traumatológicas. La mayoría de las lesiones en la médula espinal son de tipo parcial, es decir, la persona conserva alguna capacidad de movilidad.

2.1.2. Personas con discapacidad física en Ecuador

En las estadísticas presentadas por el (Consejo Nacional para la Igualdad de Discapacidades, 2022) en el país se encuentran registradas 215 156 personas con una discapacidad física, de las cuales el 49,24% tiene un porcentaje de discapacidad del 30% al 49%. De igual forma se puede evidenciar en la

ilustración de las edades que la población que se puede considerar económicamente va desde los 19 a los 64 años siendo esta la que abarca un mayor porcentaje.

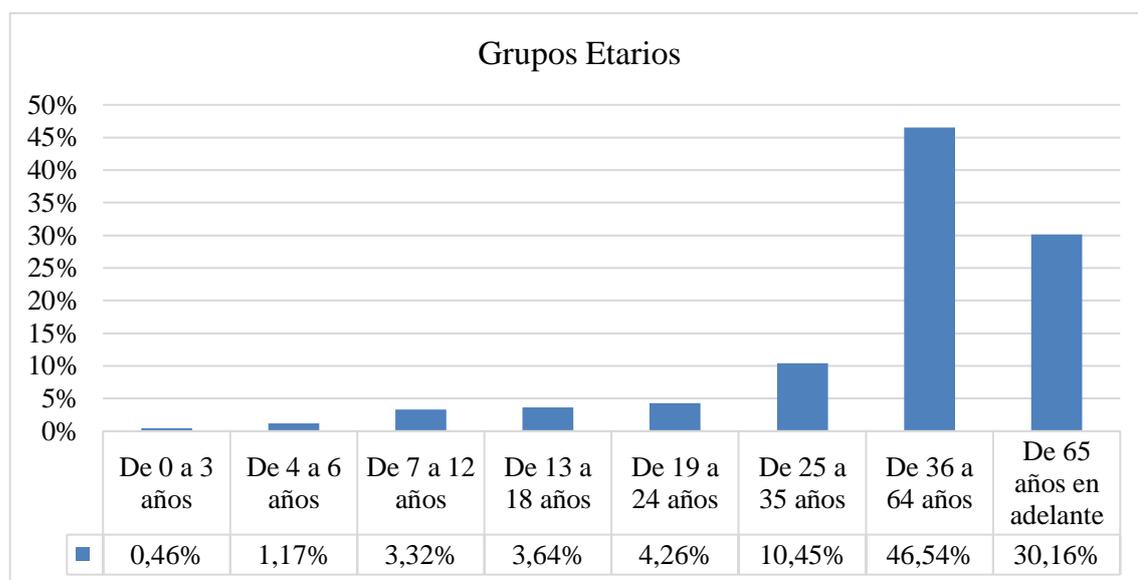


Ilustración 1-2: Porcentajes de personas con discapacidad física por edades

Fuente: Consejo Nacional para la Igualdad de Discapacidades, 2022, p. 1.

Las personas registradas en la base de datos del Consejo Nacional para la Igualdad de Discapacidades (CONADIS) y que posean el carné de discapacidad obtendrán ciertos beneficios como un descuento en el costo de los servicios básicos, descuentos o exenciones en el pago de impuestos y tributos, devolución del IVA, exoneración en tarifas de transporte, la posibilidad de importar bienes como vehículos adaptados y equipos médicos y de ayuda técnica (Reglamento a la Ley Orgánica de Discapacidades, 2017, p. 6).

2.1.3. Paraplejia en la ciudad de Riobamba

Se registran 2 942 personas con discapacidad física de acuerdo con el (Consejo Nacional para la Igualdad de Discapacidades, 2022, p. 1), en el grafico se puede observar que el mayor porcentaje de la población que se puede considerar económicamente activa tiene una tendencia muy parecida que a nivel nacional.

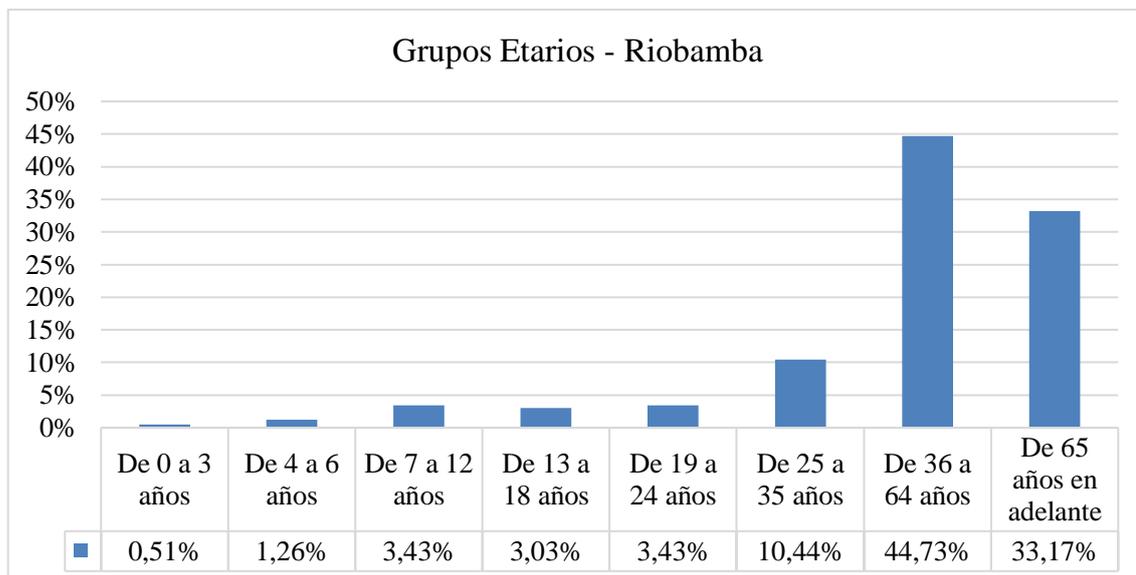


Ilustración 2-2: Porcentajes de personas con discapacidad física en la ciudad de Riobamba

Fuente: Consejo Nacional para la Igualdad de Discapacidades, 2022, p. 2.

En total un 58,6% de las personas que están entre 19 y 64 años tienen la edad para ser consideradas económicamente activas.

2.2. Automóvil

Es un medio de transporte autopropulsado que depende de una fuente de energía autónoma. En sus inicios, generalmente usa combustibles fósiles y facilita el movimiento de personas y bienes. En esencia, es un medio de transporte que no requiere tracción o propulsión externa. Por medio de un motor es capaz de mover la carrocería, esto permitió viajar largas distancias en menos tiempo, además ayudó en la carga de materiales, herramientas o mercancías de todo tipo. Su uso ha reemplazado casi por completo al uso de animales de carga (Pinilla, 2019, p. 11).

2.2.1. Tipos de automóviles según su carrocería

2.2.1.1. Sedán

Es un tipo de carrocería típica, emblemática de la industria. Se divide en 3 cuerpos: capó, habitáculo y cajuela. Es estructuralmente más fácil de fabricar, ya que al tener la separación de tres volúmenes hace que la carrocería sea más rígida (Águeda; et al., 2016, p. 356).



Ilustración 3-2: Automóvil tipo sedán

Fuente: Volvo, 2021.

En muchos países están psicológicamente asociados con automóviles de gama alta. Asimismo, los sedanes son los automóviles de representación de los dignatarios y otras personalidades públicas.

2.2.1.2. *Hatchback*

Conocido también como automóvil compacto es un tipo de carrocería que consiste en una cabina o área para pasajeros con un espacio para el maletero integrado, al cual se tiene acceso mediante un portón trasero. Su principal ventaja es el ahorro de combustible (Águeda; et al., 2016, p. 356). Una de las ventajas es la buena relación calidad – precio que tiene, además debido a que la forma del automóvil es más compacta es más fácil de estacionar y maniobrar.



Ilustración 4-2: Automóvil tipo hatchback o compacto

Fuente: Hyundai, 2021.

2.2.1.3. *SUV*

Corresponden a las siglas en inglés de *Sport Utility Vehicle* que significa vehículo utilitario deportivo, son automóviles todoterreno ligeros, suelen consumir menos combustible que un todoterreno, son más fáciles de conducir. Su función principal es la de un auto familiar (Águeda; et

al., 2016, p. 356). Son automóviles que apuestan por una estética todoterreno, pero que por sus cualidades está más orientados al recorrido en la ciudad.



Ilustración 5-2: Automóvil tipo SUV

Fuente: Stellantis, 2021.

2.2.1.4. *Camioneta*

Es una carrocería usada generalmente para el transporte de mercancías debido a que cuentan con una parte trasera descubierta (denominada caja, balde) que sirve como zona de carga para productos u objetos (Águeda; et al., 2016, p. 356).



Ilustración 6-2: Automóvil tipo camioneta

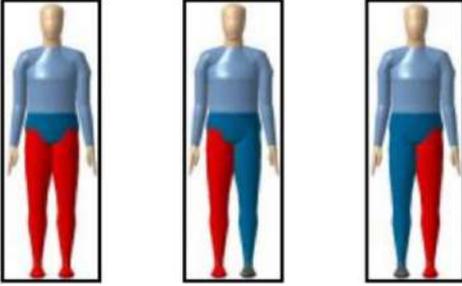
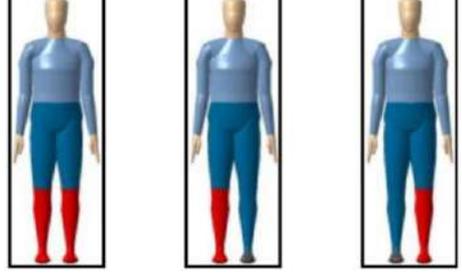
Fuente: Volkswagen, 2021.

Pueden variar según su tamaño, configuración de cabina y área de carga, tracción, motor y chasis. En América y Asia por lo general suelen ser de tracción clásica: motor delantero y tracción trasera o bien poseer sistema de doble tracción, en tanto que las de tipo europeo por estar basadas en vehículos subcompactos poseen tracción delantera.

2.3. Adaptaciones para automóviles

De acuerdo con Pérez (2001, p. 25), las adaptaciones propuestas para que personas con discapacidades puedan conducir se basan básicamente en cambios en los controles del automóvil como una forma de compensar las discapacidades físicas de los conductores. Estas modificaciones requieren un nivel mínimo de comodidad para no interferir en la conducción, pero requieren un equipamiento que permita una conducción segura y ágil, al mismo tiempo que garantiza que el conductor se adapte rápidamente a estas modificaciones, deben ser comprensibles y fáciles de aprender, no deben interferir con el proceso de conducción normal, así mismo deben permitir un adecuado funcionamiento de los diferentes sistemas de seguridad que posea el automóvil.

Tabla 1-2: Adaptaciones de acuerdo con la discapacidad

Tipo de discapacidad	Adaptaciones
	<ul style="list-style-type: none"> - Caja de cambio automática o servoembrague con control manual, semiautomático, o automatizado. - Adaptación del asiento en caso de artrodesis o anquilosis - Acelerador en el volante - Freno de servicio manual - Acelerador y embrague al algún pie específico.
	<ul style="list-style-type: none"> - Caja de cambio automática o servoembrague con control manual, semiautomático o automatizado. - Acelerador al pie izquierdo. - Acelerador en el volante - Servofreno o freno de servicio manual, preferentemente.

Fuente: (Corral, y otros, 2006)

Realizado por: Juiña, P., 2021.

2.3.1. Volante

2.3.1.1. Acelerador electrónico de aro concéntrico

Consigue una conducción más relajada al permitir acelerar indistintamente del punto en donde se presione sin importar si es con ambas manos o sólo con una. Es una opción más discreta que puede estar tanto al frente como en la parte trasera del volante (Álvarez et al., 2015, p. 58). En la

ilustración 7-2 se observa una desventaja, y es que interrumpe el funcionamiento del sistema de airbag del volante, además no es compatible con todos los modelos de vehículo.



Ilustración 7-2: Acelerador electrónico de aro

Fuente: AdaptAuto, 2018.

2.3.1.2. Acelerador mecánico de aro con freno a la mano (conjunto)

Garantiza la transformación de automóviles en los que el centro del volante no gira junto con el aro. Es una opción más completa y económica que los aceleradores electrónicos (Guerrero et al., 2022, pp. 1591-1592). La ilustración 8-2 muestra la adaptación en el automóvil. Permite acelerar mediante la tracción del aro hacia el volante, su principal desventaja es la complejidad en su adaptación por lo que hace que su mantenimiento sea más difícil de realizar.



Ilustración 8-2: Acelerador mecánico de aro con freno a la mano

Fuente: AdaptAuto, 2018.

2.3.2. Control de pedales

2.3.2.1. Leva de freno vertical

Se sujeta al piso. Diseñada para salvar la salida del airbag en la parte de las rodillas. En los pulsadores se puede integrar claxon o luces intermitentes (Guerrero et al., 2022, pp. 1591-1592). La ilustración 9-2 muestra un acercamiento de la adaptación. El frenado es manual y la intensidad es directamente proporcional a la fuerza ejercida, por lo tanto el frenado no será causado por acciones involuntarias que puedan suceder de manera natural durante esta acción.



Ilustración 9-2: Leva de freno vertical

Fuente: AdaptAuto, 2018.

2.3.2.2. Leva vertical de acelerador y freno

Anclado al piso, este conjunto se convierte en una alternativa a los mandos convencionales de acelerador y freno (Guerrero et al., 2022, pp. 1591-1592). En la ilustración 10-2 se muestra cómo queda la adaptación al automóvil. De esta forma se consigue dos puntos principales que son: el mantener las configuraciones del volante como la altura y la profundidad, además de ser compatible con el airbag de rodilla.



Ilustración 10-2: Leva vertical de acelerador y freno

Fuente: AdaptAuto, 2018.

2.3.3. Sistema total driving

Permite la conducción del vehículo sin necesidad de bajarse de la silla de ruedas, se puede ingresar con ella hasta la posición del conductor (Guerrero et al., 2022, pp. 1591-1592). En la ilustración 11-2 como el sistema *total driving* se adapta a la carrocería del automóvil.



Ilustración 11-2: Sistema total driving

Fuente: AdaptAuto, 2018.

La silla se fija mediante un sistema de anclaje directamente en el suelo del automóvil. Esta es su principal desventaja puesto que hace más cara la adaptación, además, no todos los automóviles están en la capacidad de ser adaptados por cuestiones de tamaño y anchura de la carrocería.

2.4. Pedales del automóvil

2.4.1. Acelerador

Es un mecanismo usado en los motores de combustión interna mediante el cual se regula el flujo de la mezcla aire/combustible en el caso de motores de carburador o del aire en los motores de inyección de combustible por medio de constricción y obstrucción del conducto de admisión. La variación en el llenado del cilindro que se obtiene de esta manera permite regular la masa de mezcla introducida en el cilindro en el proceso de renovación de la carga de este. Esta variable determina, por tanto, la fuerza de la carrera de expansión y, finalmente, el par motor en cada ciclo

útil. Es decir, esta regulación de la cantidad de gas combustible admitido en cada ciclo nos permite ajustar el par motor a la carga motor. Esto quiere decir que, teniendo diferentes condiciones de carga y el mismo régimen, las diferentes posiciones de la mariposa nos permite ajustar el llenado del cilindro y, por tanto, la fuerza de la combustión, con el mismo régimen, manteniendo siempre la proporción aire combustible. Como la potencia de un motor es igual al par multiplicado por el régimen motor, se puede incrementar o disminuir ésta, mediante la restricción de gases para la mezcla de aire y combustible.

El término acelerador hace referencia, de forma incorrecta e informal, a cualquier mecanismo mediante el cual se regula la potencia o velocidad de un motor. En el motor diésel no existe este mecanismo de restricción; se dosifica en su lugar el caudal de combustible mediante una bomba inyectora o inyectores de mando electrónico (Otero, 2017, p. 36). El funcionamiento del acelerador tiene dos señales que son incorporadas y provenientes de dos potenciómetros que cambian el voltaje dependiendo si está abierto o cerrado el pedal. En la ilustración 12-2 se muestra como es el funcionamiento del sistema acelerador del automóvil.

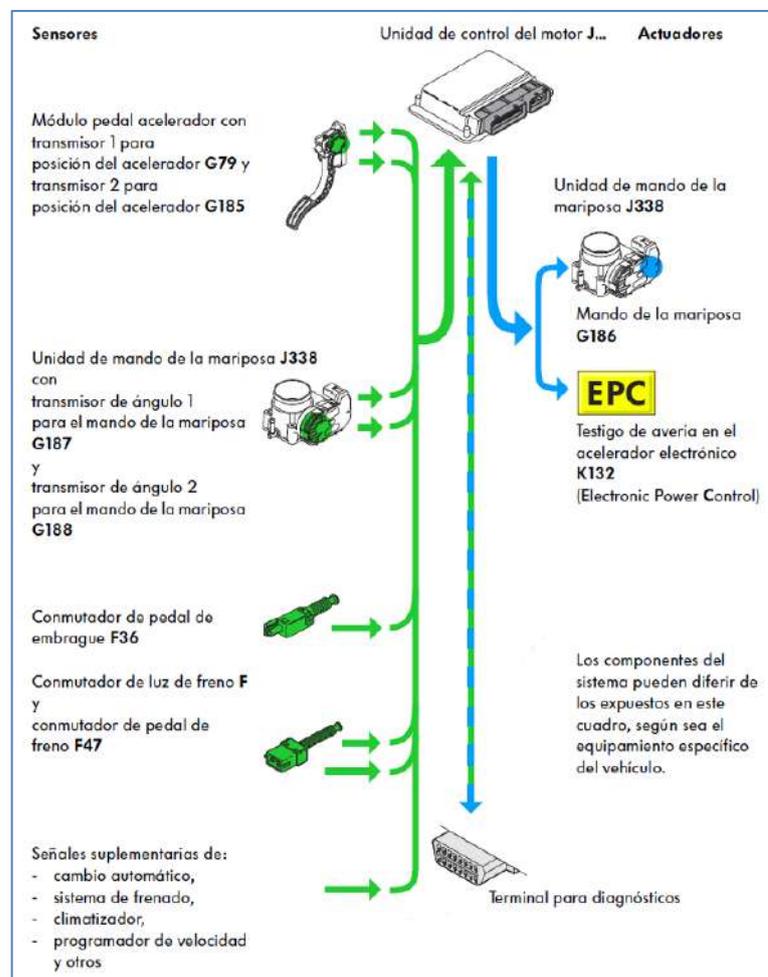


Ilustración 12-2: Sistema del acelerador

Fuente: (e-auto, 2015)

Cuando se trata de mantener una velocidad constante, el pedal del acelerador debe pisarse ligeramente, para evitar la fatiga del pie y reducir el riesgo de una aceleración repentina. Por otro lado, hay que empujar con fuerza cuando se quiere acelerar en la carretera o adelantar a otro vehículo (Dols, 2015, p. 6).

2.4.2. Freno

Es un dispositivo cuyo propósito es detener o disminuir la velocidad del automóvil, se pueden considerar transformadores de energía. A pesar de que los frenos son también máquinas, generalmente se les encuentra en la literatura del diseño como un “elemento de máquina”. En la ilustración 13-2 se muestra el principio de funcionamiento del sistema de freno.

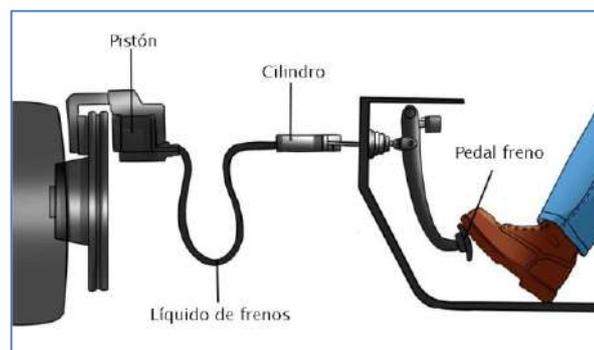


Ilustración 13-2: Sistema de freno

Fuente: Get Auto, 2017.

Es el sistema que se considera que es el más importante en cuestión de seguridad vial, los accidentes pueden ser evitados dándole un uso correcto y recordando siempre darle mantenimiento. Este pedal se encuentra ubicado en el medio de la línea de pedales, ligeramente más alto que el acelerador, se compone de un sistema hidráulico que se acopla a cada una de las ruedas del auto, y a través de un engranaje genera una presión que provoca por el líquido de frenos se distribuya uniformemente en el sistema.

Cualquier fuga o derrame del líquido de frenos es motivo de asistencia inmediata ya que afecta de manera directa la funcionalidad y pone en peligro al conductor, cuando hay presión en el pedal por leve que esta sea, se envía una señal de advertencia externa a través de una luz activa en la parte trasera del automóvil. Es importante tener en cuenta que no se debe abusar del freno y mantenerlo en buenas condiciones (Alonso, 2001, p. 170).

2.4.3. Embrague

Este pedal es necesario al momento de conducir un automóvil de transmisión manual. Una buena comprensión de su función nos permite usarlo adecuadamente y así evitar el desgaste prematuro de sus partes. Es un sistema que interrumpe la transmisión de potencia del motor y, por lo tanto, controla la tracción creada por las ruedas (Trifonov, 2019). En otras palabras, es lo que permite el cambio de marcha, este sistema se activa mediante el pedal ubicado en la parte izquierda de la línea de pedales, después de pisarlo el motor se desconecta y la transmisión deja de recibir potencia (Camarena Lillo, 2011, p. 3).

2.4.3.1. Posición de embrague

El disco está conectado y permite la transmisión de la potencia del motor a las ruedas. Los propios resortes del plato de presión son los que nos van a acoplar el disco de embrague al volante motor (Otero, 2017, p. 37). En la ilustración 14-2 se aprecia la posición de embragado, el pie no presiona el pedal por lo que el embrague está acoplado al motor.

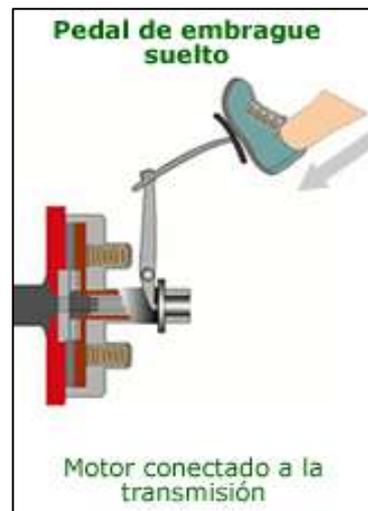


Ilustración 14-2: Posición de embrague

Fuente: Coches.com, 2018.

2.4.3.2. Posición de desembrague

El pedal está pisado, desconectando al motor de la transmisión. Esta posición permite realizar el cambio de marcha. Las ruedas están libres y dependiendo de la inercia del automóvil continuará su marcha o se detendrá (Otero, 2017, p. 38).

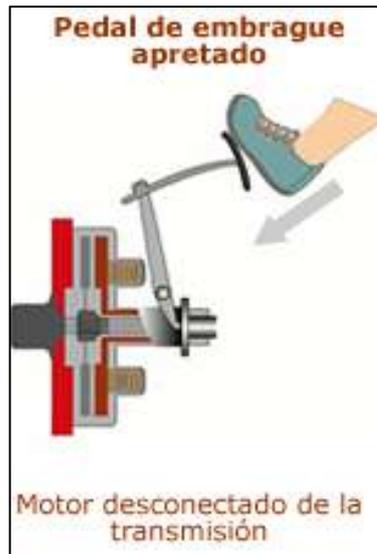


Ilustración 15-2: Posición de desembrague

Fuente: Coches.com, 2018.

En la ilustración 15-2 se observa el momento en el que pisamos el pedal, esto hace desplazar el collarín hacia el interior presionando las pastillas, éstas, al girar sobre la articulación, vencen la resistencia de los muelles separando el disco de embrague del volante motor.

2.4.3.3. Fase transitoria

El pedal se encuentra entre ambas posiciones, en este punto puede funcionar igual, aligerando la carga mecánica del cambio y permitiendo que no se haga de manera brusca evitando los daños en el motor (Otero, 2017, p. 38).

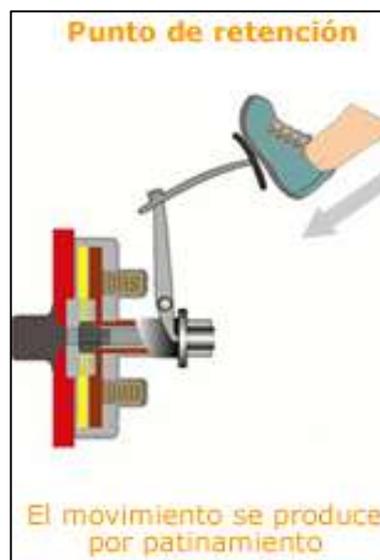


Ilustración 16-2: Fase transitoria

Fuente: Coches.com, 2018.

En la ilustración 16-2 se muestra el momento en el que se está presionando el pedal, realizar el cambio de marcha en esta posición no es lo recomendable puesto que puede generar un desgaste mayor en el embrague.

2.5. Motores paso a paso

Un motor paso a paso es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamientos angulares, lo que significa que es capaz de girar una cantidad de grados dependiendo de sus entradas de control. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Es por eso por lo que ese tipo de motores son muy utilizados, ya que pueden moverse a deseo del usuario según la secuencia que se les indique a través de un microcontrolador. El motor paso a paso es un motor DC sin escobillas que se divide en series de pasos dándole así su nombre por la estructura del motor, como son 200 series en las que se divide el eje de 360° , por lo que cada $1,8^\circ$ se realiza un avance o paso (Fernández, 2011, p. 17).

Están diseñados para girar un cierto ángulo, el movimiento se basa en una señal eléctrica aplicada a sus terminales de control. El tamaño o la precisión de los pasos que puede tomar el motor dependerán de sus características estructurales, incluidos ángulos de menos de 10° a más de 150° según el modelo (Redondo, 2016, pp. 3-4). Estos tipos de motores se usan comúnmente en sistemas de control digital donde el motor es impulsado en un bucle abierto por un grupo de pulsos para girar el eje en un ángulo completamente definido. Las aplicaciones típicas incluyen, entre otras, motores de alimentación en impresoras y máquinas de etiquetadoras, ubicación de cabezales de impresión o bolígrafos en trazadores, cabezales de lectura/escritura en unidades de computadora, ubicación de herramientas y vehículos en equipos de mecanizado CNC, aplicaciones robóticas, etc. (Jennings, 2002, p. 50).

Tiene dos estados, arranque y parada. Este movimiento puede ser suave o desigual según la frecuencia y la magnitud de los pasos involucrados en la inercia del rotor. Como todos los motores, los motores paso a paso son convertidores electromecánicos, pero debido a su aplicación específica, forman una categoría distinta. Estos tipos de motores responden de una manera muy obvia (es decir, el eje gira en uno o más pasos) a señales digitales específicas que se alimentan a su mecanismo de control. Por lo tanto, los motores paso a paso se pueden utilizar para el control en sistemas abiertos sin retroalimentación. Esto evita los problemas que a menudo causan los sistemas receptivos, como la inestabilidad y el exceso de velocidad. Sin embargo, el motor paso

a paso puede ser reemplazado por un el motor de corriente continua tradicional en el sistema de retroalimentación servo (Muniozguren, 2018, p. 20).

La característica principal de un motor paso a paso es que la rotación es controlada, por lo que cada paso representa un desplazamiento angular constante del eje del motor. Este tipo de motores son impulsados por bobinas que crean una variación de flujo (Conti, 2005, pp. 2-4). En la ilustración 17-2 se muestra un motor paso a paso nema 17.

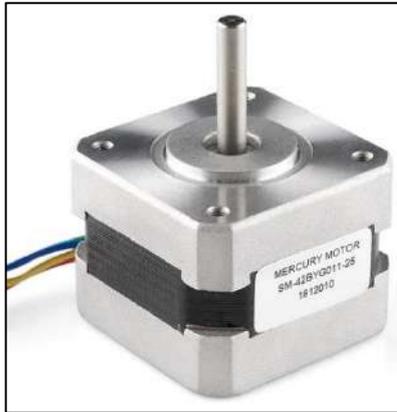


Ilustración 17-2: Motor paso a paso

Fuente: Control automático educación, 2010.

En el caso de los motores paso a paso, controlaremos el desplazamiento del rotor en función del voltaje aplicado a las bobinas. Esto significa que podemos controlar el avance y el retroceso, así como un número específico de pasos para cada revolución.

2.5.1. Motores bipolares y unipolares

Una sección especial de los motores paso a paso se extrae del método utilizado para el bobinado de motor bifásicos. De acuerdo con esto se subdividen en unipolares y bipolares. Se diferencian en que los motores unipolares trabajan con un polo de corriente, mientras que los motores bipolares trabajan con ambos polos, lo que significa que la dirección de la corriente en los devanados es cambiante (Conti, 2005).

Los devanados del motor deberán conectarse de tal forma que permitan la transmisión de energía entre bobinas. Este método permite el uso de corriente unipolar.

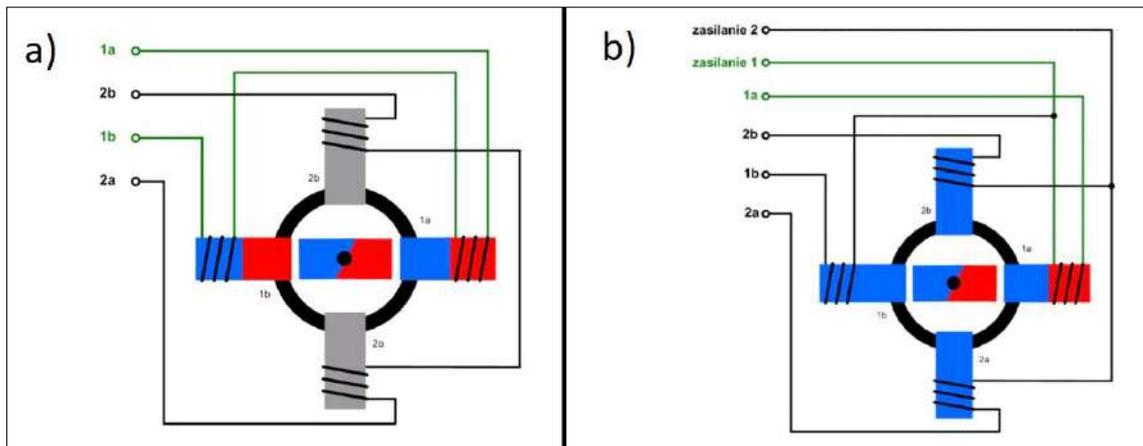


Ilustración 18-2: Tipos de motores paso a paso

Fuente: TME, 2020.

Las diferencias se muestran en la ilustración 18-2 donde el literal “a” corresponde a un motor bipolar y el “b” es un motor unipolar. Es recomendable hacer las mediciones precisas al motor a pasos, esto permite conectarlo apropiadamente y aprovechar al máximo su funcionamiento.

2.5.2. Ventajas y desventajas de los motores paso a paso

La principal ventaja es la facilidad en el control de los giros, la velocidad y su precisa maniobrabilidad. Todo esto se debe a su diseño simple que también hace que su costo sea bajo en comparación con otras opciones. Se pueden obtener torques altos a bajas velocidades. En su construcción no hay escobillas lo que provee de una alta resistencia mecánica y confiabilidad. Otro punto para tener en cuenta su fácil control: con arranque inmediato, una parada efectiva debido al alto torque y tener la posibilidad de cambiar el sentido de rotación inmediatamente. Su mayor inconveniente es su consumo de energía, necesita estar energizado cuando está en movimiento y cuando está en reposo. Tiene mayor torque a velocidades bajas pero a mayores velocidades este tiende a bajar.

Está estrechamente relacionado con la corriente que fluye a través de los devanados, su impedancia aumenta al incrementarse la frecuencia de conmutación. Esto hace que sea imposible conseguir altas revoluciones manteniendo el torque del motor y la capacidad de carga. Si el torque no es suficiente, se producirá deslizamientos o pasos en falso. Por lo que se necesario un dispositivo de retroalimentación para mejorar control, por ejemplo, un motor que está basado en un codificador permite que la unidad de control del motor asegure el número de pasos ordenado (Jennings, 2002, p. 25).

2.6. Software CAD

El diseño o dibujo asistido por computadora (CAD por sus siglas en inglés) es una tecnología utilizada para el diseño técnico que reemplaza los dibujos manuales, automatizando y digitalizando el proceso. Estos programas son generalmente usados en las áreas de ingeniería, arquitectura, diseño gráfico, ayudan mucho en la redacción de documentación de construcción, estudiar conceptos de diseño, renderizaciones y simulación del comportamiento de estructuras en la vida real (Autodesk, 2021, párr. 1).

2.6.1. SolidWorks

Es un software CAD usado principalmente para diseñar piezas y ensamblajes en 3D, también para la creación de planos en 2D. Tiene una amplia gama de soluciones para ofrecer, que cubren todos los aspectos de desarrollo de productos. Sus productos brindan la capacidad de crear, trazar, producir, administrar datos de procesos de diseño y realizar simulaciones (SolidBi, 2015, párr. 1).

2.7. Sistemas de control

El inicio de los sistemas de control dentro de la civilización humana data de hace mucho tiempo atrás. Se tiene conocimiento que los primeros sistemas de control se registran en la época de la Grecia antigua. Durante este tiempo, se construyeron mecanismos para una plataforma que flotaba en el agua. Este suceso se considera históricamente como el primer sistema de control automático hecho por el ser humano. Es un sistema formado por un conjunto de elementos destinados a controlar otros sistemas. Su objetivo es realizar eficientemente las tareas y trabajos programados para él. Para ello, deben comportarse de forma coherente frente a los errores. Estos sistemas son muy útiles actualmente dado que tienen una gran capacidad de analizar la información de forma automática. Los recientes avances tecnológicos de los últimos años han propiciado el desarrollo de sistemas de control automatizado, esto a su vez genera nuevas teorías, crea procesos y retroalimenta el desarrollo de más tecnología (Agudelo et al., 2020, pp. 2-3).

Se clasifican en dos tipos dependiendo del manejo que le dan a la variable que se necesita controlar. De lazo abierto, en esta categoría no hay información ni retroalimentación sobre la variable que se requiere controlar. Esto significa que la salida es independiente de la entrada. Es ampliamente aplicada en procesos y sistemas donde las variables se pueden predecir y márgenes de error bastante grandes (Reyes; et al., 2013, p. 35).

Como por ejemplo los semáforos. Son circuito de control abiertos, no existen datos sobre el tráfico en tiempo real, son sincronizados pero cada semáforo se programa de forma independiente. Los equipos de amplificación modulares son otro claro ejemplo de sistemas de control de lazo abierto, puesto que, cuando se realiza un cambio en la potencia del volumen, el equipo no reconoce esta que se haya realizado esta acción. En la ilustración 19-2 se describe de forma general un sistema de control de lazo abierto.

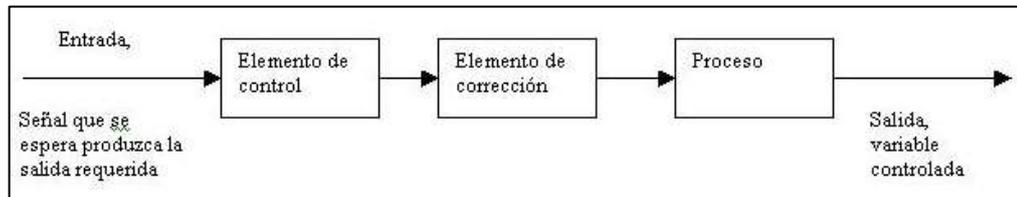


Ilustración 19-2: Sistema de control de lazo abierto

Fuente: Sites.google, 2016.

Y de lazo cerrado, a diferencia de la categoría anterior, en este si existe información acerca de la variable, inclusive los estados en los que se encuentra envían retroalimentación al sistema, esto se logra mediante sensores que se distribuyen estratégicamente lo largo del sistema para que recolecten la información de la variable. Son los sensores los que hacen que el sistema sea completamente autónomo (Reyes et al., 2013, pp. 48-49). Los aires acondicionados son un ejemplo bastante extendido sobre este sistema. Los sensores que tienen miden la temperatura ambiental (variable) y este decide si debe hacer trabajar al compresor para enfriar el ambiente o no. Los componentes elementales para el control de procesos dentro de la ingeniería pueden parecer simples pero son totalmente todo lo opuesto. La automatización la podemos encontrar en casi todos los dispositivos que utilizamos diariamente, este recurso es la base para la ingeniería industrial.

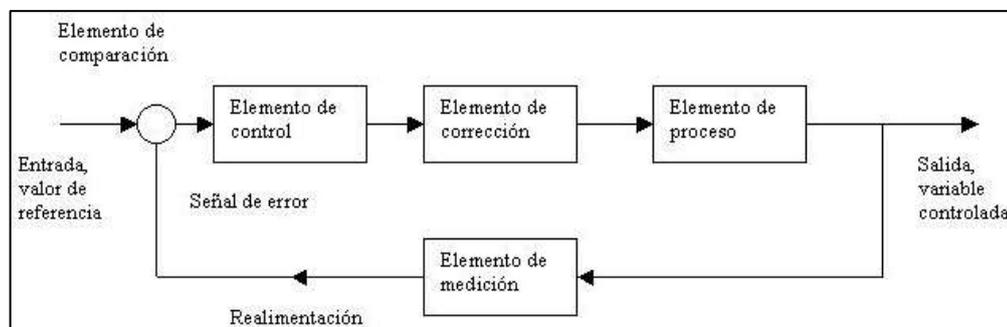


Ilustración 20-2: Sistema de control de lazo cerrado

Fuente: Sites.google, 2016.

En la ilustración 20-2 se describe de manera general un sistema de control de lazo cerrado, se trata de sistemas complejos, ya que poseen gran cantidad de parámetros, su salida se compara con la entrada y esto afecta al control del sistema, es más estable ante variaciones internas.

2.8. Arduino

Es una plataforma de hardware y software open-source, utilizada en pequeños proyectos de electrónica, automatización, además de ser un elemento didáctico muy extendido en, debido a su bajo costo y simplicidad de arquitectura es ideal para aquellas personas que desean aprender a programar. En lo correspondiente al hardware es una placa con circuito integrado que cuenta con un microprocesador, que realizara tareas lógicas, posee varias entradas y salidas tanto analógicas como digitales en las que se pueden conectar diferentes dispositivos, ya sean estos desarrollados por la propia organización o por terceros. En lo referente al software, su nombre es Arduino IDE, es una interfaz creada en Java que permite la programación de las placas compatibles con la arquitectura Arduino, puesto que es software libre, terceras personas pueden usar su código para desarrollar otro tipo placas compatibles. Acepta lenguajes de programación como C y C ++ siempre y cuando se cumplan con ciertas normas en la estructura del código. Es multi plataforma lo que quiere decir que podrá ser instalado en MacOS, Windows y Linux (Peña, 2020, p. 60).

Tabla 2-2: Componentes de la placa Arduino

COMPONENTE	DESCRIPCIÓN
Pines digitales	Son 14 en total, se los cuenta desde el 0 hasta el 13, corresponden a las conexiones de los elementos acoplados a la placa. Solo comprenden lenguaje binario, eso quiere decir, 0 y 1, o pulsos eléctricos de 0V(low) o 5V(high). Se los puede utilizar como pines de entrada o salida.
Pines analógicos	Estos pueden reconocer valores continuos que van desde 0V hasta 5V. A ciertos pines digitales también se los puede utilizar como pines analógicos, siempre y cuando tengan la simbología “~” presente en la placa.
Pines de alimentación	Permiten energizar los diferentes dispositivos que se vayan a conectar a la placa. Específicamente pueden alimentar 5V; 3,3V, además de esto también están presentes los pines GND o tierra.
Microcontrolador de comunicaciones	Es el responsable de administrar las comunicaciones de la placa con todos los dispositivos que se conecten a ella.
Microcontrolador de programación	Se programa con la interfaz de desarrollo Arduino IDE, es la parte más importante de la placa, es el cerebro de todo, aquí se guardan las instrucciones que van a ejecutar los dispositivos conectados la placa.
Botón reset	Interrumpe el programa que se esté ejecutando en el microcontrolador, reinicia la placa mas no borra el programa que ya está cargado.

Puerto USB	Puede ser considerado como un puerto de alimentación, aunque su principal función es la de conectar la placa con una computadora, así se puede cargar la programación y recibir información.
Conector de alimentación	La placa permite la alimentación externa mediante este conector, de esta forma se asegura el funcionamiento independiente si la necesidad de conectarla a una computadora. La alimentación debe estar entre los rangos de 7V y 12V, si se llega a sobre pasar es posible que la placa se vea afectada.

Fuente: Vázquez, 2015, pp. 4-5.

Realizado por: Juiña, P., 2021.

Los componentes son lo que generalmente se encuentran en el Arduino, esto dependerá mucho del tipo de placa a utilizarse, es posible que muchos de los componentes enlistados no se encuentren o sean cambiados por otros (Peña, 2020).

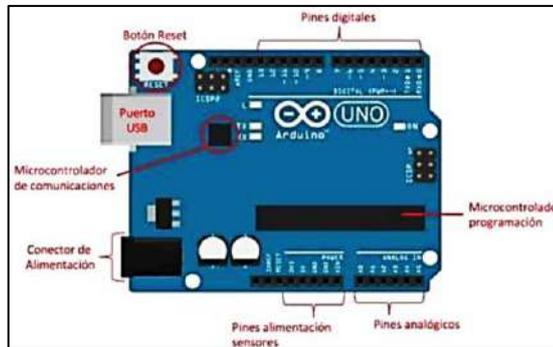


Ilustración 21-2: Partes de la placa Arduino

Fuente: Vázquez, 2015.

La figura 20-2 nos muestra la manera en la que los diferentes componentes se distribuyen en la placa Arduino. La disposición de estos puede cambiar dependiendo del tipo de placa que se usará.

2.8.1. Principales placas Arduino

Arduino dispone de una amplia variedad de placas para usar dependiendo de las necesidades ya sea esta educación o automatización básica. Con el tiempo se hace necesario que surja nuevo hardware que cumpla ciertos parámetros y además este actualizado, una muestra de la diversidad de placas Arduino se presenta en la siguiente tabla.

Tabla 2-2: Placas arduino

A continuación se enlistan los tipos de placas que se encuentran disponibles en el mercado.

Placas básicas	Placas avanzadas	Placas para internet de las cosas	Placas para wearables
Leonardo	Mega	YÚN	Gemma
Arduino UNO R3	ADK	Ethernet	Lilypad USB
Genuino 101	DUE	TIAN	Lilypad Main Board
Esplora	Zero	Industrial 101	Lilypad Simple
Micro	MO	Leonardo ETH	Lilypad Simple Snap
Nano	MO PRO	MKR 1000 WIFI	
Mini	MR ZERO	MKR FOX 1200	
		MKR WAN 1300	
		MKR GSM 1400	
		YUM MINI	

Fuente: Solectro, 2021.

Realizado por: Juiña, P., 2021.

Las placas poco convencionales son las dedicadas para wearables “Lilypad”, estas están pensadas para el desarrollo de aplicaciones electrónicas textiles, por lo que tiene conectores grandes con la finalidad de ser cocidos.

2.9. Raspberry pi

Es una placa electrónica diseñada por la fundación Raspberry, funciona como una minicomputadora, debido a su diseño y estructura su precio es asequible a todos los usuarios. Es ampliamente utilizada en el área de computación y robótica, siendo su aplicación más extendida la automatización de procesos, y el desarrollo de prototipos. Por su bajo costo es muy popular como material didáctico en colegios y universidades (Raspberry Pi Foundation, 2017, párr. 1-2).



Ilustración 22-2: Raspberry pi 1 A+

Fuente: Raspberry Pi Foundation, 2017.

En la figura 20-2 se muestra la primera placa que la fundación lanzó en febrero de 2012. Con el propósito de fomentar la enseñanza computacional, teniendo como requisitos primordiales ofrecer potencia en un tamaño reducido y ser lo más económica posible.

2.9.1. Modelos de raspberry

En la página web oficial de Raspberry se encuentran los modelos de placa doméstica, además de sus variaciones para uso industrial y una colección de accesorios, pero también existen placas no oficiales, haciendo que las posibilidades crezcan. En la siguiente tabla se resumen las distintas opciones.

Tabla 3-2: Placas raspberry y sus características

Especificaciones	Raspberry Pi 3 modelo B+	Raspberry Pi 3 modelo A+	Raspberry Pi 4 modelo B
SoC (CPU, GPU, DSP, RAM y puertos USB)	Broadcom BCM2837	Broadcom BCM2837	Broadcom BCM2711
CPU	1.4GHz 64-bit quad-core ARMv8	1.4GHz 64-bit quad-core ARMv8	1.5GHz 64-bit quad-core Cortex-A72
Juego de instrucciones	RISC de 64 bits	RISC de 64 bits	RISC de 64 bits
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC	Broadcom VideoCore VI, OpenGL ES 3.0, 1080p30 H.264/MPEG-4 AVC, 4kp60 H.265
Memoria	1 GB (compartidos con la GPU)	512 MiB (compartidos con la GPU)	1 GB, 2 GB, 4 GB u 8 GB (compartidos con la GPU)
Puertos USB 2.0	4	1	2
Puertos USB 3.0	Ninguno	Ninguno	2
Entradas de vídeo	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la Raspberry Pi Foundation	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la Raspberry Pi Foundation	Conector MIPI CSI que permite instalar un módulo de cámara

			desarrollado por la Raspberry Pi Foundation
Salidas de vídeo	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD	Conector RCA (PAL y NTSC), microHDMI rev. 2.0, Interfaz DSI para panel LCD
Salidas de audio	Jack de 3.5 mm, HDMI	Jack de 3.5 mm, HDMI	Jack de 3.5 mm, 2 puertos microHDMI
Almacenamiento	MicroSD	MicroSD	MicroSD
Conectividad de red	Puerto RJ-45 (Ethernet) de 10/100/1000Mbps vía hub USB limitado a 300Mbit/s Wi-Fi 802.11ac de doble banda Bluetooth 4.2 BLE	Wifi 802.11ac de doble banda Bluetooth 4.2 BLE	Puerto RJ-45 10/100/1000Mbps vía hub USB 3.0 Wi-Fi 802.11ac de doble banda Bluetooth 5.0 BLE
Periféricos de bajo nivel	17 x GPIO y un bus HAT ID	17 x GPIO y un bus HAT ID	17 x GPIO y un bus HAT ID
Consumo energético	800 mA (4.0 W)	800 mA (4.0 W)	Máximo 3A (15.3 W)
Fuente de alimentación	5 V vía Micro USB o puerto GPIO	5 V vía Micro USB o puerto GPIO	5 V vía USB-C o puerto GPIO
Dimensiones	85mm x 53mm	85mm x 53mm	85mm x 53mm
Sistemas operativos soportados	GNU/Linux: Raspbian, Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM, RISC OS	GNU/Linux: Raspbian, Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM, RISC OS	GNU/Linux: Raspbian

Fuente: Raspberry, 2021.

Realizado por: Juiña, P., 2021.

Como se puede observar, existe una gran cantidad de versiones de las Raspberry Pi pensadas para cualquier tipo de uso que imaginemos. Los proyectos que se pueden desarrollar con una de estas placas de formato compacto están limitados por los conocimientos y la imaginación del usuario. En internet es posible encontrar una gran variedad de proyectos y documentación sobre los que construir cualquier cosa prácticamente. El poder de las Raspberry Pi reside en su bajo precio y la enorme versatilidad que ofrece, además de complementar su uso con accesorios útiles como cámaras, teclados, ratones, medidores de temperatura, gas, humedad, viento y un sinfín de otros elementos para el desarrollo de proyectos (González, 2015, p. 7). Teniendo esto en cuenta se tiene una

visión más amplia y clara sobre la placa, así es posible orientarse de mejor manera a la hora de escoger aquella que mejor se ajuste a las necesidades.

2.9.2. Raspberry pi OS

El sistema operativo denominado Raspberry OS, antes llamado Raspbian, que monta las placas está basado en una distribución de GNU/Linux, un software *open-source* que permite la interacción del usuario con la placa. A pesar de que es una computadora y tiene funciones especiales, su alcance es limitado. Los desarrolladores sugieren el uso de Raspberry OS dado que es una versión estable y amigable pero el entorno en el que se vaya a usar la placa depende enteramente de las aplicaciones que se le quiera dar (Raspberry Pi Foundation, 2017, párr. 2).

2.10. Sistema simulador cockpit

Cockpit es un anglicismo proveniente del campo de la aviación. En lo que respecta al automovilismo, este hace referencia a la cabina de un automóvil, también se designa con este nombre a una estación que simula ser una cabina de automóvil, esta cabina simulada puede tener diferentes aditamentos para hacer más realista la simulación de la conducción, puede tener pedales, volante, palanca de cambios, en algunos casos si se quiere llevar la simulación a un ambiente virtual también puede poseer pantallas o en su defecto un tablero de control en que se muestren los diferentes medidores, relojes y alertas que tienen los paneles en un automóvil real. Además también es muy utilizado en el ámbito de los videojuegos donde se pueden adaptar los controles a los simuladores de conducción (Juarros Izquierdo, 2022). En la ilustración 23-2 se muestra un cockpit dedicado a entornos virtuales, el modelo hace referencia a una cabina simulada para jugar a videojuegos relacionados con automóviles o simuladores de vuelo.



Ilustración 23-2: Cockpit simracing

Fuente: González, 2015, p. 8.

2.11. Impresión 3D

La impresión 3D es una alternativa rápida y económica a las tecnologías más avanzadas de manufactura tradicional, como por ejemplo, el mecanizado, fundición, control numérico, etc. Esta tecnología permite la fabricación de herramientas, instrumentos y estructuras complejas diseñadas en un programa CAD, con diferentes tipos de materiales (Ortiz, 2019, p. 3).

2.11.1. Tipos de impresión 3D

2.11.1.1. FDM

Impresión por deposición o filamento fundidos, es el tipo de impresión 3D más extendido debido a que la fabricación de las piezas es rápida y económica además es la más adecuada para la creación de prototipos conceptuales o básicos, como aquellas que pueden someterse a esfuerzos simples. Comparándola con otros tipos de impresión 3D es la menos precisa en cuanto a detalles puesto que está limitada por las configuraciones y el tamaño de las boquillas de impresión. No se recomienda usarla en piezas complejas con detalles extremadamente pequeños, cuando las capas se funden puede haber la posibilidad de que no se compacten por completo lo que generará piezas deficientes, lo que hay que tomar en cuenta si estas serán sometidas a esfuerzos o cargas (Rúa et al., 2017, p. 73). En la ilustración 24-2 se muestra una impresora FMD de tipo cartesiana.



Ilustración 24-2: Impresora 3D FDM

Fuente: Rúa et al., 2017, p. 73

Se puede conseguir mejores resultados y acabados mediante el uso de químicos y pulidos mecánicos. El tipo de resultados que se obtenga dependerá mucho de la impresora 3D que se use, por ejemplo las máquinas industriales de impresión usan termoplásticos de ingeniería muchos

más sofisticados para aminorar los posibles errores, esto con un precio mucho más elevado para el consumidor (Jorquera, 2016, p. 44).

2.11.1.2. Resina

La impresión 3D mediante resina se clasifica en 2 tipos de tecnología: SLA o DLP, y estas se basan en la estereolitografía, esta fue el primer tipo de impresión 3D del mundo, desarrollada en los años 80 y es la más usada de manera profesional. Las piezas impresas poseen mayor precisión, detalles mucho más cuidados y mejores acabados que cualquier otro tipo de impresión 3D (Rúa et al., 2017, p. 73). En la ilustración 25-2 se muestra una impresora de resina que utiliza el proceso DLP.



Ilustración 25-2: Impresora 3D DLP o resina

Fuente: <https://goo.su/ZaePZ>

Es una gran elección cuando se necesita fabricar prototipos muy detallados que necesitan altas tolerancias, como por ejemplo moldes o piezas funcionales. Estas también pueden ser alisadas o adornadas para convertirse en productos finales para su comercialización. Presentan una resistencia más homogénea independientemente de la orientación de impresión, debido a que surgen enlaces químicos en cada capa. El resultado son piezas con un rendimiento mecánico previsible, lo cual es importante si el fin de estas es ser accesorios de fijación o sujeción (Rúa et al., 2017, p. 73).

2.12. Ángulos de confort

Son los ángulos que tienen las articulaciones, fuera de los cuales el trabajo a realizar es más difícil, penoso o incluso peligroso para las personas. Los ángulos de confort son los movimientos que podemos realizar con las diversas articulaciones. Los ángulos de confort nos dan los entornos de los límites mínimos y máximos de apertura de las articulaciones en las que estas no están

sometidas a tensiones mayores de lo normal. Cabe destacar que los ajustes de los ángulos de confort dependerán de la edad, la condición física, diferencias anatómico-funcionales, del sexo (Soto, 2009).

En la ilustración 26-2 se muestran los ángulos de confort recomendados para manos, pies y extremidades.

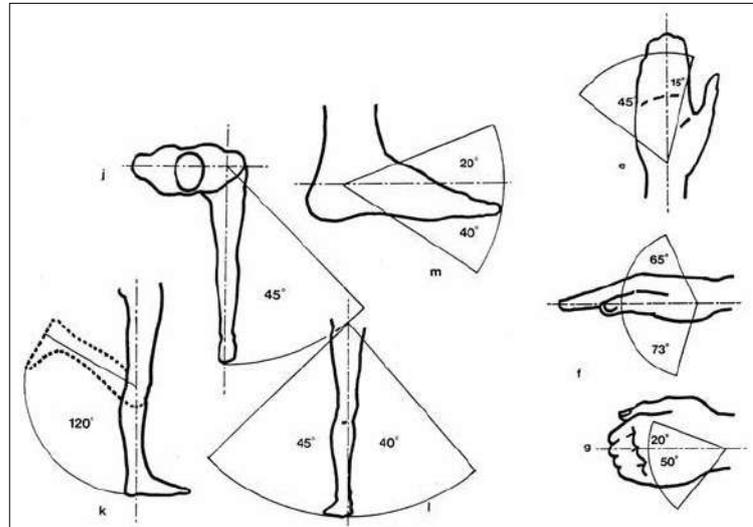


Ilustración 26-2: Ángulos de confort

Fuente: Soto, 2009.

El estudio y la determinación de los valores otorgados a los ángulos de confort se obtienen primordialmente por medio de la información suministrada por muestras de sujetos encuestados, a través de técnicas de observación directa, fotografías, grabaciones de vídeo o por medio de la medición del desplazamiento de los segmentos, utilizando instrumentos como goniómetros o flexómetros. Por otra parte, existen otros procedimientos, como por el análisis de datos anatomopatológicos obtenidos de observaciones clínicas y constataciones radiológicas, o bien, por los valores de la intensidad de la contracción de los músculos posturales obtenidos por métodos electromiográficos.

Tabla 4-2: Ángulos de confort para un puesto de conducción

Postura	Zona Corporal	Arco
	Espalda	$10^\circ < A1 < 20^\circ$
	Piernas	$90^\circ < A2 < 110^\circ$
	Rodilla	$80^\circ < A3 < 120^\circ$
	Pie	$90^\circ < A4 < 110^\circ$
	Brazo	$80^\circ < A5 < 160^\circ$

Fuente: Soto, 2009.

Realizado por: Juiña, P., 2021.

En la tabla 4-2 se muestra los ángulos de confort para un puesto de conducción, donde a cada zona corporal se ha asignado un código y también se muestran los rangos en los que se puede mover respectivamente.

CAPÍTULO III

3. MARCO METODOLÓGICO

Para comenzar con el desarrollo de este proyecto se realizó un estudio descriptivo, que partió con la fundamentación teórica, revisando aquellos elementos que van a ser necesarios en el sistema de control, sus diferentes características con la finalidad de crear una metodología comparativa, para establecer los requerimientos en el diseño mecánico, hardware y software.

3.1. Definición de los requerimientos

Para que el objetivo de construir un prototipo de asistencia en la conducción se lleve a cabo, como primer requisito se realizó una investigación sobre los tipos de adaptaciones o equipamientos a automóviles, para comprender de mejor manera la disponibilidad del software, hardware y materiales a usarse en el desarrollo del prototipo.

Para el funcionamiento es necesario que el sistema cuente con elementos mecánicos debidamente manufacturados, elementos electromecánicos y electrónicos funcionales como componentes vitales, estos deben ser elegidos cuando cumplan con los siguientes objetivos:

- Definir el proceso de conducción de automóviles de transmisión manual mediante un algoritmo o la descripción de una secuencia de pasos.
- Construir una estructura que sirva como entorno de experimentación y soporte de equipos y dispositivos considerados para la implementación del prototipo.
- Con el enfoque a la generación de un prototipo para asistencia a personas con restricción motora en las piernas, vincular actuadores que generen movimiento para el accionamiento de los pedales del auto a la hora de conducir.
- En base a la revisión teórica realizada en el capítulo 2 se aprecia que los modelos de sistemas de asistencia en la conducción para personas con paraplejia se apoyan en la interacción con los miembros superiores, por lo tanto para el desarrollo del prototipo, proponer un tipo de control de los actuadores a través de señales generadas por medio de las manos.
- Integrar al prototipo una interfaz gráfica que permita la interacción del usuario con la máquina para su manipulación y calibración.
- Seleccionar hardware y software *opensource* con el objetivo de reducir costos por licencias.

3.2. Consolidación del prototipo

La consolidación del prototipo se muestra en la ilustración 1-3 con los equipos y dispositivos que lo conforman, se describe un sistema de asistencia en la conducción de un automóvil para personas con paraplejía.

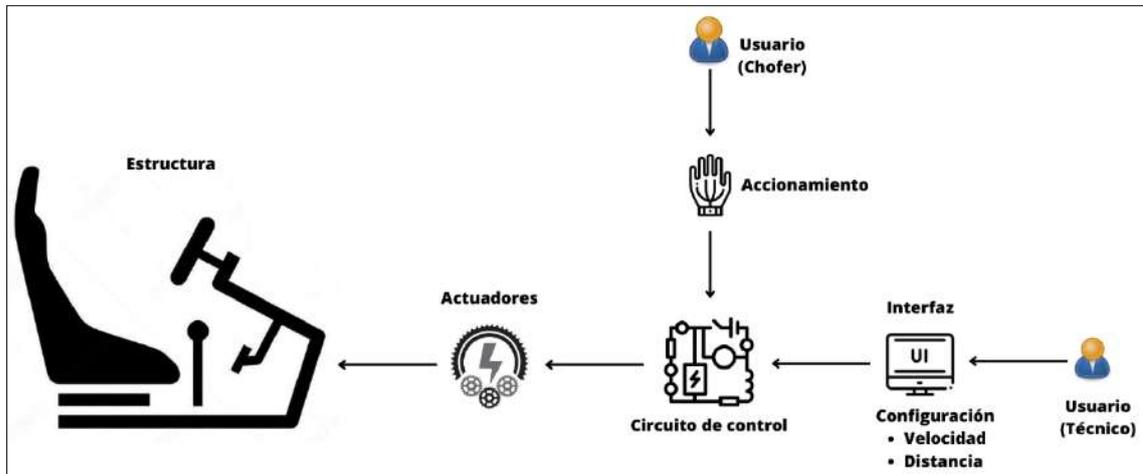


Ilustración 1-3: Consolidación del prototipo

Realizado por: Juiña, P., 2021

El funcionamiento inicia con la interfaz gráfica para la configuración y calibración de los actuadores, una vez que se hayan establecido se envían a los microcontroladores. El usuario enviará señales mediante los controles en sus manos, estas serán interpretadas mediante programación y así se gobernará el movimiento de los pedales.

Cada pedal tendrá su propio microcontrolador, esto permitirá independencia de maniobra, cumplirán con tres funciones principales:

- Recibir y almacenar las configuraciones de velocidad y distancia de desplazamiento por parte de la interfaz gráfica.
- Recibir las señales que el usuario enviará por medio de los controles en las manos.
- Mover los actuadores de acuerdo con los parámetros previamente fijados.
- Permitiendo:
 - Independencia en el movimiento y maniobrabilidad de los pedales.
 - Ingresar diferentes parámetros de velocidad y distancia para que cada pedal se ajuste a las necesidades del usuario.
 - Visualizar en tiempo real las configuraciones que se van realizando en la interfaz gráfica.

3.3. Arquitectura del sistema

En la ilustración 2-3 se representa el diagrama de bloques de la arquitectura general del sistema. Se marcan diferentes secciones en las que se divide el prototipo, dentro de las cuales se utiliza equipos y dispositivos específicos con tareas y propósitos definidos.

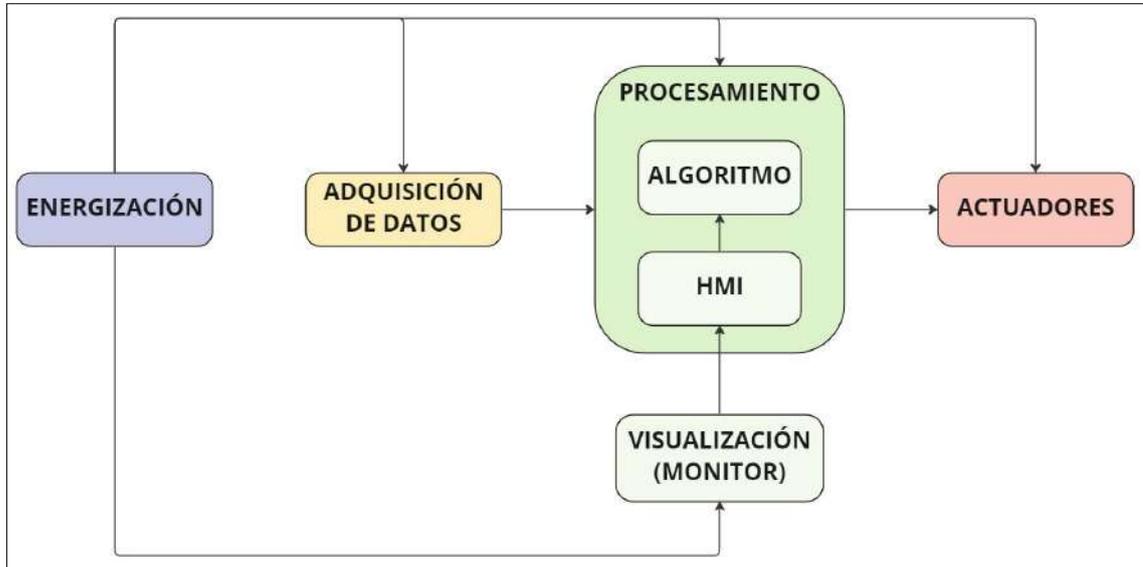


Ilustración 2-3: Diagrama de bloques de la arquitectura del sistema

Realizado por: Juiña, P., 2021

La arquitectura relaciona los recursos presentes en el prototipo, la función que cumple cada uno de ellos se detalla a continuación.

- **Bloque de energización:** Esta sección se refiere a la fuente de suministro de energía que requieren los elementos eléctricos y electrónicos que conforman el prototipo. Los elementos principales se alimentan de corriente continua, lo que hace necesario el uso de fuentes conmutadoras que se conecten al suministro eléctrico de 110V de corriente alterna.
- **Bloque de adquisición de datos:** Esta etapa es destinada a la toma de variables de velocidad y distancia ingresadas mediante una interfaz gráfica y las señales enviadas mediante el control en las manos del usuario. Además, hay que recalcar que esta etapa se define también por los algoritmos que conforman la programación.
- **Bloque de procesamiento:** En esta sección se gestionan las variables que ingresan mediante la interfaz gráfica, estas se envían a los microprocesadores y se almacenan para ser utilizados en un bucle, aquí también se gestionan las señales del control en las manos del usuario para los actuadores del prototipo, acorde a la configuración previamente guardada.
- **Actuadores:** Luego de la gestión de los datos obtenidos por parte de la interfaz gráfica y del usuario se ejecutan los algoritmos codificados en los microprocesadores, resultando así en la sección de salida donde las señales enviadas generan un movimiento controlado en los

actuadores. En esta etapa se genera un bucle con las señales enviadas por el control en las manos del usuario que permite el uso del prototipo de manera continua porque las configuraciones se encuentran almacenadas.

3.3.1. Definición del algoritmo de conducción

Conducir un automóvil es una acción compleja que no sólo necesita de habilidades y técnica sino que también juega un papel importante el estado de ánimo de la persona. No es suficiente con saber arrancar el automóvil, se hace necesario incorporar conciencia sobre el cuidado propio y hacia terceras personas, porque la vía pública es una zona de armonía donde se tiene que ser respetuoso. Antes, durante y después del proceso de conducción se deben tener presentes los conceptos de valor y comprensión hacia las leyes, comprender para qué y por qué existen (Herrera & Pueblas, 2021, p. 4).

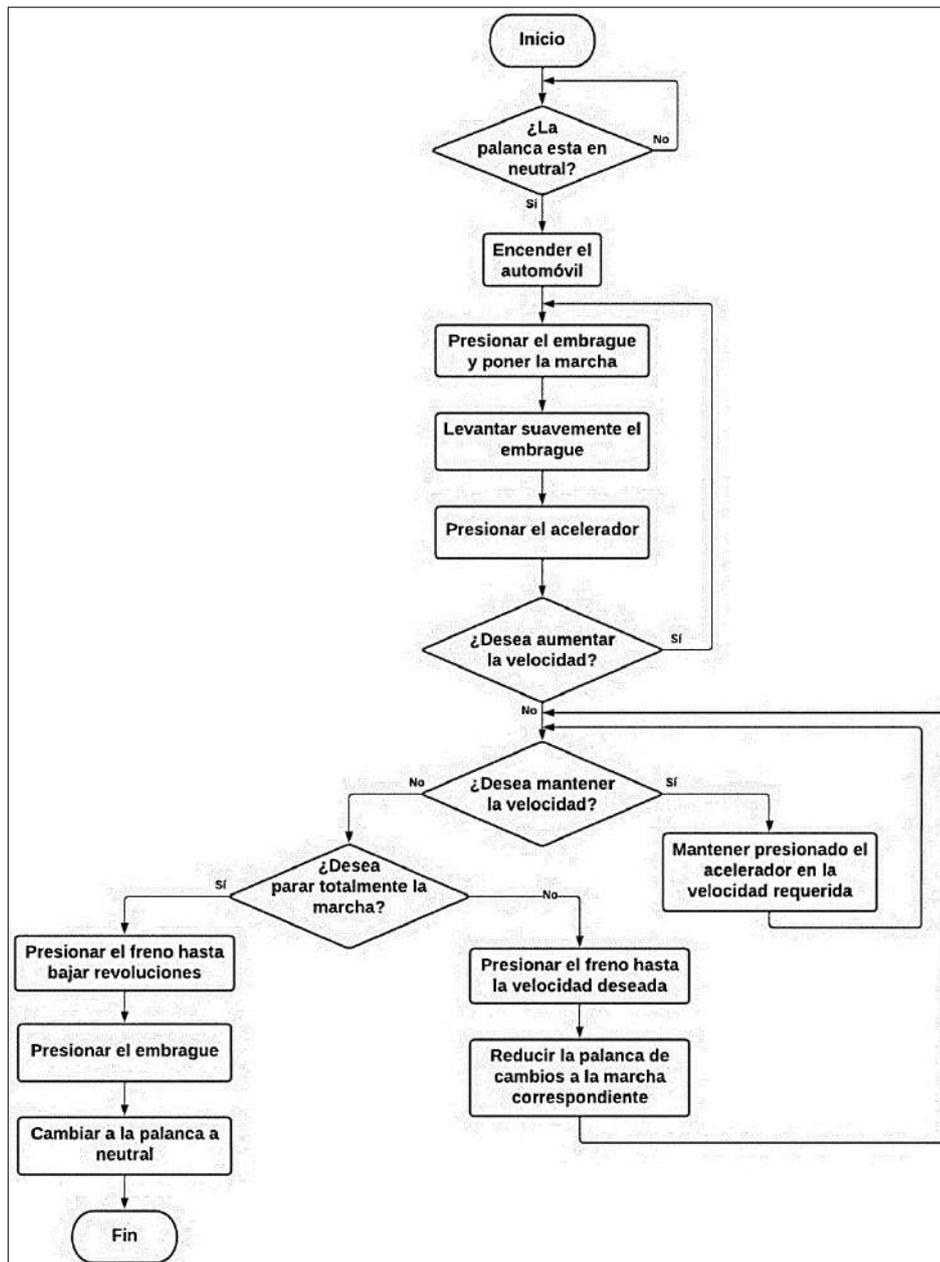


Ilustración 3-3: Diagrama de flujo del proceso de conducción

Fuente: Fajardo, 2018.

En el ilustración 2-3 se muestra el diagrama de flujo del proceso de conducción para un automóvil de transmisión manual. Para tener una idea más clara sobre el proceso de conducción en la ilustración 3-3 se muestra mediante imágenes el proceso antes descrito.

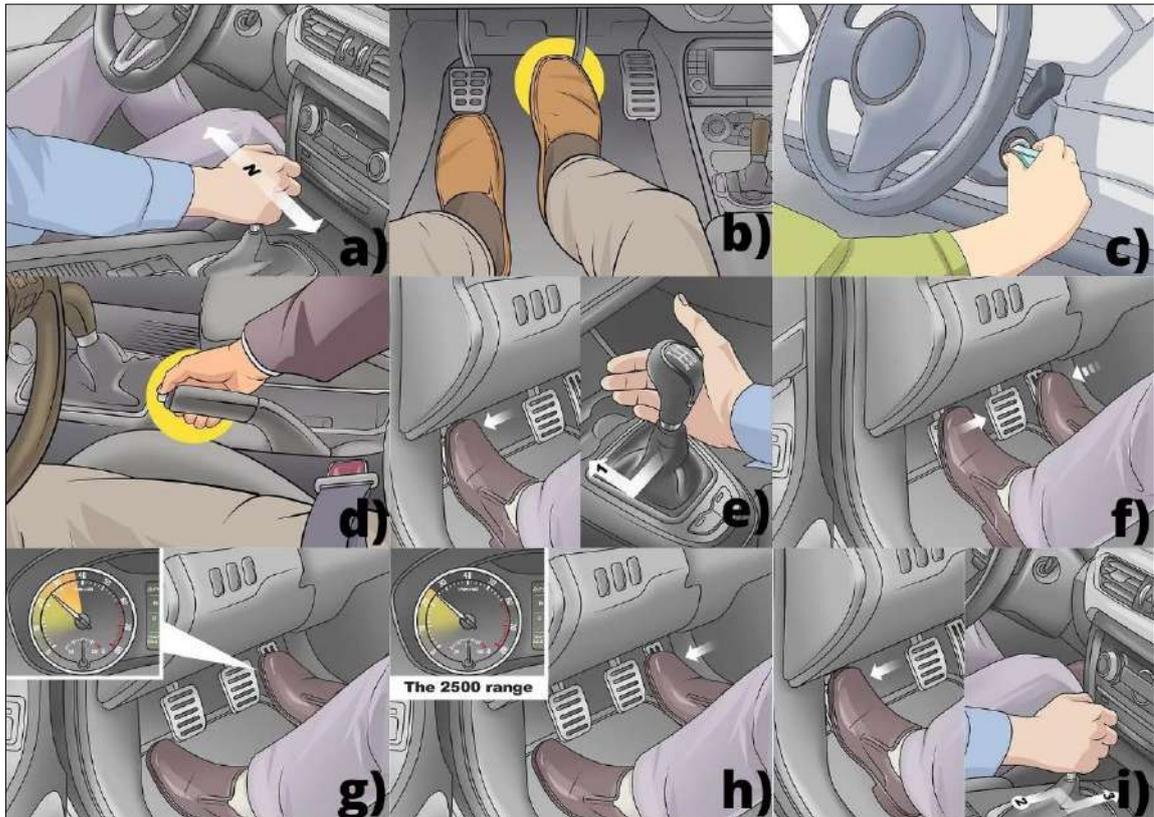


Ilustración 4-3: Proceso de conducción de un automóvil de transmisión manual

Fuente: WikiHow, 2019.

Antes de conducir el automóvil hay que asegurarse de que su encendido es seguro. En la ilustración 4-3 en el literal “a” se comprueba que la palanca de cambios debe estar en posición neutral, luego en el literal “b” el conductor debe mantener el pedal de freno presionado para evitar que el carro pueda hacer un recorrido no deseado, en el literal “c” se procede con el encendido al girar la llave, se desactiva el freno de mano como se indica en el literal “d”.

Una vez encendido el auto se procede con el proceso de conducción, presionando el pedal de embrague a fondo y posicionando la primera marcha con la palanca de cambios como se muestra en el literal “e”. Para que el automóvil empiece a moverse se debe subir el pedal del embrague poco a poco, esta maniobra evitará que el motor se apague, de la misma manera se presiona el acelerador como se indica en el literal “f”, con esto la velocidad aumentará como se muestra en el literal “g”.

En el tablero se observarán las revoluciones por minuto del motor (RPM) como indica el literal “h”, así al alcanzar ciertas revoluciones se hace necesario el cambio de marcha. Para hacerlo hay que pisa nuevamente el embrague, colocar la marcha correspondiente con la palanca de cambios y nuevamente dejar de levantar el embrague como se muestra en el literal “i”.

Este ciclo se repetirá para las demás velocidades. Para disminuir la velocidad, se tiene que hacer la misma maniobra de embrague y posicionar la palanca en la marcha correspondiente. Para detener totalmente el automóvil se debe pisar el freno y al llegar a bajas revoluciones se pisa el embrague para evitar el apagado del motor. Para reanudar el recorrido se debe iniciar nuevamente con los pasos.

3.3.2. Entorno de experimentación

En esta parte se detalla el diseño estructural en software CAD del entorno de experimentación que brinda soporte a todos los equipos y dispositivos que conforman la sección mecánica y electromecánica del prototipo. Se consideró como ideal para simular el entorno de conducción de un automóvil de transmisión manual realizar el diseño de un cockpit. Se partió de la revisión de las medidas antropométricas consideradas como ideales en el proceso de conducción, la ilustración 5-3 representa estas medidas para un puesto de conducción.

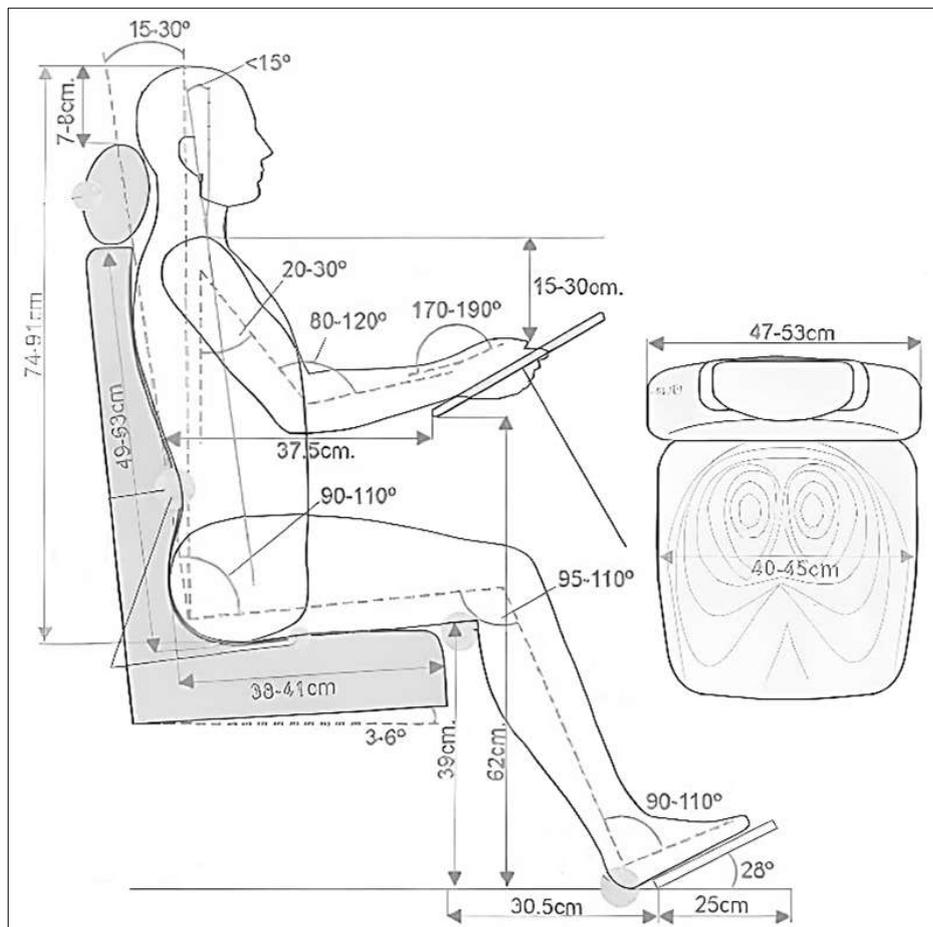


Ilustración 5-3: Medidas antropométricas para un puesto de conducción

Fuente: Quintana, 2017.

Para una mejor visualización se desarrolló la tabla 1-3 donde se resaltan cada uno de los rangos en los que se pueden mover las diferentes partes del cuerpo que intervienen en el proceso de conducción.

Tabla 1-3: Medias antropométricas para un puesto de conducción

Zona Corporal	Rango
Espalda	15° - 30°
Brazo	20° - 30°
Codo	80° - 120°
Muñeca	170° - 190°
Cadera	90° - 110°
Rodilla	95° - 110°
Tobillo	90° - 110°

Fuente: Quintana, 2017.

Realizado por: Juiña, P., 2021

Con estas referencias se comenzó con el diseño del cockpit, además se añadieron los diseños de un asiento, volante y pedales para simular la cabina de un automóvil de transmisión manual, consta de embrague, freno y acelerador, estos se accionarán mediante actuadores. Se usaron pedales de un automóvil Chevrolet Aveo estos son de acero y la parte que tiene contacto con el pie tiene una cobertura de caucho. Para ser montados en la estructura base, cada uno cuenta con un soporte metálico que será atornillado en la estructura.

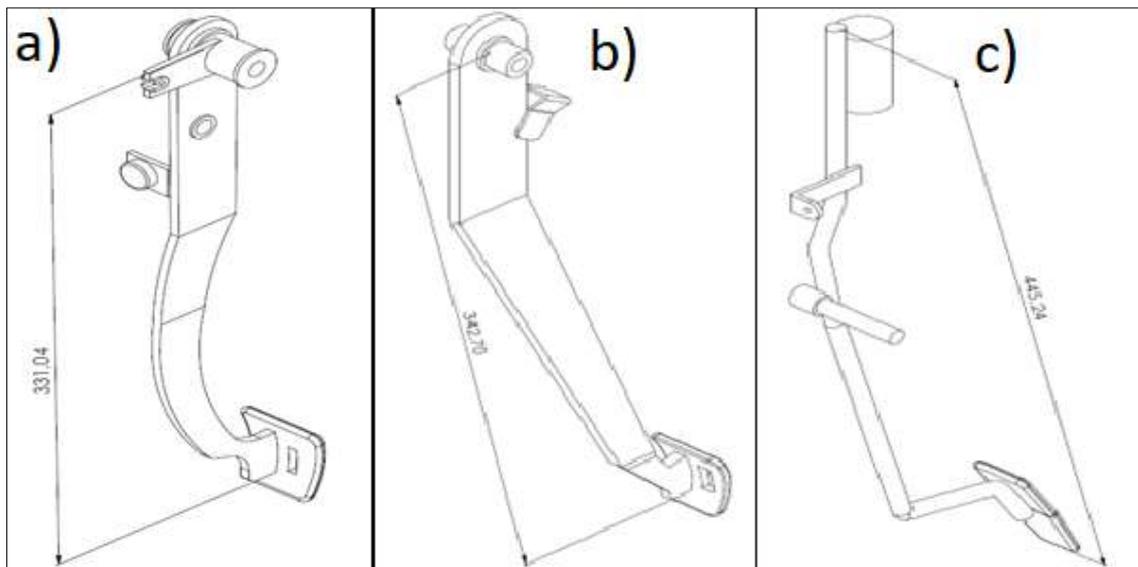


Ilustración 6-3: Pedales

Realizado por: Juiña, P., 2021.

En la ilustración 6-3 se muestran las medidas de los pedales: a) pedal de embrague tiene un total de 331,04 mm de largo, b) freno tiene 324,70 mm y c) acelerador con 445,24 mm.

Aunque existen diferentes tipos de volante para automóviles, para el entorno de experimentación se tomaron en cuenta las medidas de un volante convencional de forma circular, en la ilustración 7-3 se representan las dimensiones del accesorio, este posee un brazo metálico que se ajustará a la estructura del cockpit mediante pernos.

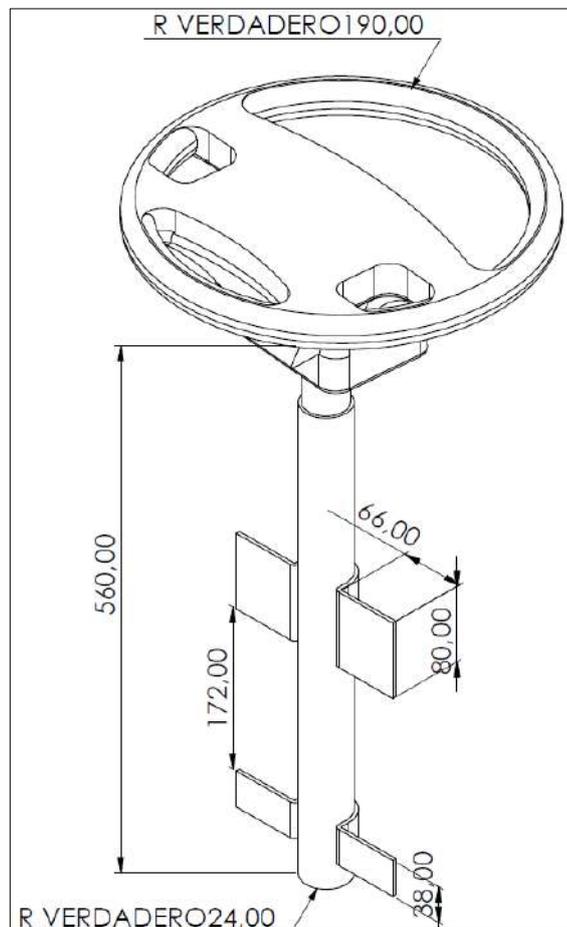


Ilustración 7-3: Volante

Realizado por: Juiña, P., 2021.

Para el confort se añadió un asiento como el que se muestra en la ilustración 8-3. El asiento es de esponja cubierta con tela de microfibra, la altura del asiento es ajustable al igual que la posición del espaldar, la cabecera es extraíble y puede rotar su posición, la distancia del asiento con los pedales se puede ajustar con el sistema de rieles que este posee.

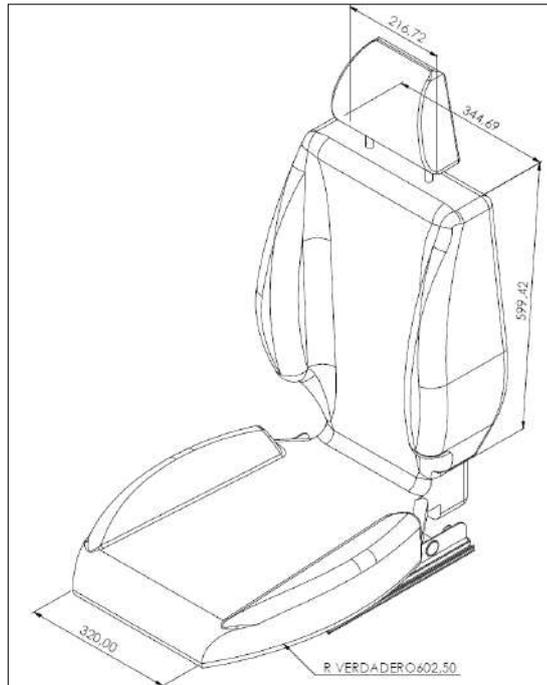


Ilustración 8-3: Medidas generales de “Asiento”

Realizado por: Juiña. P., 2021.

El asiento se ubicará en una parte elevada de la estructura que ha sido destinada para este fin, sus medidas generales son 320 mm de ancho, 599,42 mm de alto, así mismo la cabecera tiene 216,72 mm de ancho. En la siguiente tabla se muestran más características.

Tabla 2-3: Características del asiento

Fabricante	GO Car
Modelo	JRS-01-WT
Peso	20 kg
Dimensiones	320 x 599,42 x 216.72
País de origen	China
Posición	Conductor

Realizado por: Juiña, P., 2021.

Con todas las dimensiones obtenidas se diseñó la estructura que se muestra en la ilustración 9-3 mediante el programa SolidWorks, en esta estructura se montaron los elementos para el entorno experimentación, tiene 1500 mm de largo por 430 mm de ancho, 200 mm de altura, armada con tubos metálicos de 30 x 30 mm en acero laminado al frío AISI 1020.

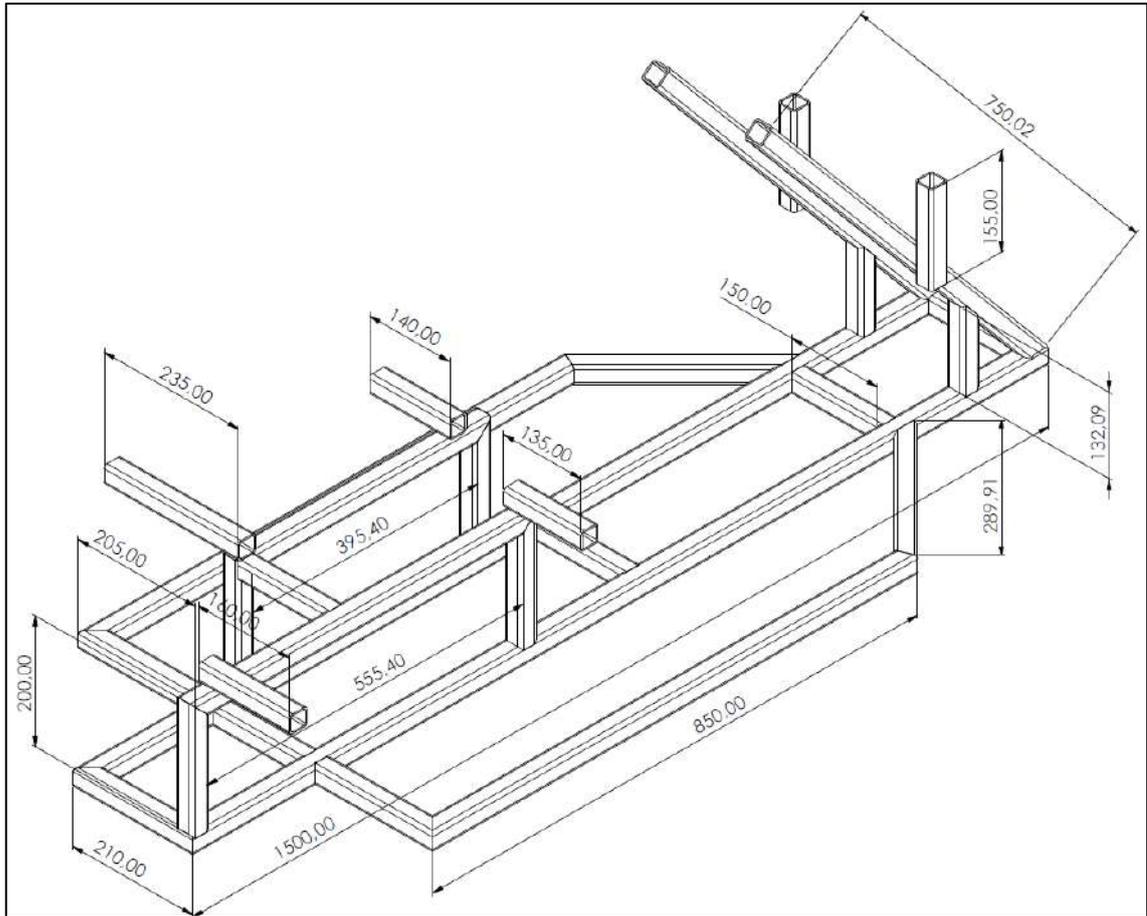


Ilustración 9-3: Medidas de la estructura base

Realizado por: Juiña, P., 2021.

La estructura se diseñó para limitar su uso en personas de estatura promedio en el país que de acuerdo con (Datos Mundial, 2020) es de 167 cm, además de soportar un peso de 74,2 kg se debe sumar el peso de la silla de automóvil que se va a colocar en el cockpit que de acuerdo con (Renting Finders, 2020) es de 20 kg en promedio.

Se procedió con la construcción de la estructura del cockpit, esta se compone de por tubo cuadrado de acero de 30x30 mm, que en sus juntas está soldada por la técnica de arco eléctrico, y recubierta por una capa de pintura para evitar la oxidación por la interacción con el ambiente. Una vez acabada la estructura base del cockpit, se colocaron los demás elementos como el asiento, el volante y los pedales.



Ilustración 10-3: Cockpit

Realizado por: Juiña, P., 2021.

La ilustración 10-3 representa la implementación del cockpit se puede apreciar su estructura básica y el montaje de los elementos tales como el volante, asiento y pedales. Sobre esta estructura se colocará todo el sistema mecánico y electromecánico que servirá para accionar los pedales.

3.3.3. Actuadores

Para el prototipo es recomendable el uso de actuadores eléctricos, puesto que en un automóvil tenemos alimentación eléctrica constante, son dispositivos de menor costo y de fácil acceso. El tipo de actuador eléctrico que se escogió son los motores paso a paso, que aportarán con el torque necesario para el movimiento de los pedales, su control es sencillo, además su reducido tamaño hace que su montaje en el cockpit sea mucho más fácil, se hace necesario colocar una base en la estructura para que los motores se puedan fijar. Para la adquisición de los motores se tomó en cuenta el torque necesario para el accionamiento de los distintos pedales. Haciendo uso de las siguientes fórmulas:

Ecuación 1-3: Fórmula de torque

$$T = F * d \quad (\text{Ec. 1-3})$$

Donde: T = Torque

F = Fuerza

d = distancia

Ecuación 2-3: Fuerza aplicada en los pedales

$$F = m * g \quad (\text{Ec. 2-3})$$

Donde: F = Fuerza

m = masa

g = gravedad

Reemplazando la ecuación 2-3 en la ecuación 1-3 se obtiene que la fórmula del torque es:

$$T = m * g * d$$

Con esta fórmula se va a calcular el torque que va a necesitar cada pedal, debido a esto se debe obtener los datos de la masa que se requiere para moverlos y la distancia donde se aplicará la fuerza hasta su punto de apoyo.



Ilustración 11-3: Dinamómetro

Realizado por: Juiña, P., 2021.

Los valores de la masa se conseguirán usando un dinamómetro como el que se muestra en la ilustración 11-3, es un artefacto destinado a la medición de la fuerza, el peso o masa de los objetos a partir de la elasticidad de un resorte o muelle elástico.

La masa necesaria para mover cada pedal obtenida mediante medición se multiplica por la gravedad ($9,81 \text{ m/s}^2$) y por la distancia entre el punto de aplicación de la fuerza en el pedal y el eje de apoyo de este cada uno, obteniendo así el torque requerido.

En la ilustración 12-3 se muestra las mediciones realizadas a cada pedal, de esta manera se obtuvieron los siguientes resultados para a) que corresponde al pedal de embrague la masa necesaria para moverlo es de 1 kg, para b) que corresponde al pedal de freno la masa necesaria para moverlo es de 0,4 kg, para c) que corresponde al pedal de aceleración la masa necesaria para moverlo es de 0,45 kg. Además se muestra las distancias de aplicación de la fuerza en cada pedal, de esta manera se obtuvieron los siguientes resultados para a) que corresponde al pedal de embrague la distancia es de 0,3 m, para b) que corresponde al pedal de freno la distancia es de 0,32 m, para c) que corresponde al pedal de aceleración la distancia es de 0,25 m.

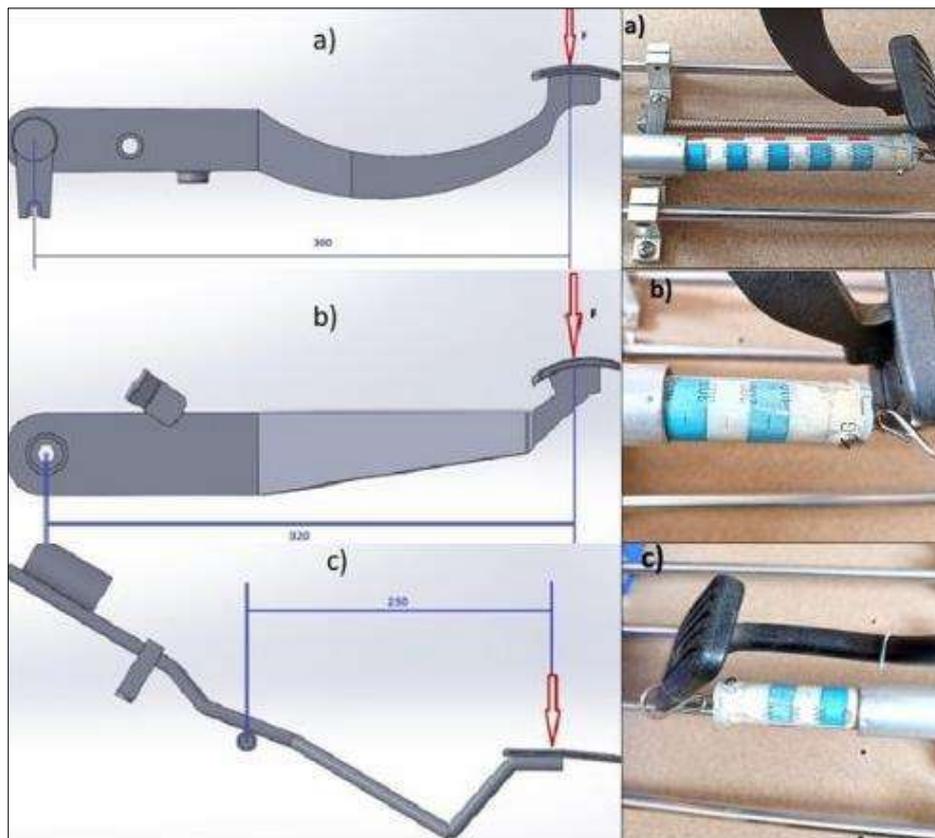


Ilustración 12-3: Fuerza ejercida y distancia de los pedales.

Realizado por: Juiña, P, 2021.

Una vez obtenidos estos valores se reemplazan en la fórmula del torque, en la siguiente tabla se muestran los resultados obtenidos.

Tabla 3-3: Torques

Pedal	Formula	Resultado
Embrague	$T = 1kg * 9,81 \frac{m}{s^2} * 0,30m$	$T = 2,94 Nm$
Freno	$T = 0,4kg * 9,81 \frac{m}{s^2} * 0,32m$	$T = 1,25 Nm$
Acelerador	$T = 0,45kg * 9,81 \frac{m}{s^2} * 0,25m$	$T = 1,13 Nm$

Realizado por: Juiña, P., 2021.

Con los valores obtenidos, para el embrague el motor no debe tener un torque menor a 2,94 Nm, por lo que se optó por un motor de 3 Nm, para el freno el motor no debe tener un torque menor a 1,25 Nm, por lo que se optó por un motor de 2 Nm, para el acelerador el motor no debe tener un torque menor a 1,13 Nm, por lo que se optó por un motor de 2 Nm.

Para el proyecto se seleccionó motores Nema 23, que cuentan con el torque requerido para mover los pedales del sistema cockpit. A lassiguientes tablas se muestra las especificaciones eléctricas y físicas.

Tabla 4-3: Especificaciones Eléctricas

Especificaciones Eléctricas	
Número de pieza del fabricante	23HE45-4204S
Número de fase	2
Ángulo de paso	1,8°
Torque	3 Nm – 2Nm
Corriente Nominal	4,2 A
Resistencia de fase	0,9 $\Omega \pm 10\%$
Inductancia	3,8 mH $\pm 20\%$

Fuente: OMC StepperOnline, 2018.

Realizado por: Juiña, P., 2021.

Tabla 5-3: Especificaciones Físicas

Especificaciones Físicas	
Tamaño del marco	57 x 57 mm
Longitud del cuerpo	113 mm
Diámetro del eje	10 mm
Longitud del eje	24 mm
Número de hilos	4
Peso	1,8 kg

Fuente: OMC StepperOnline, 2018.

En la ilustración 13-3 se muestra un motor paso a paso nema 23, estos son utilizados en CNC y máquinas con mayores necesidades de torque.

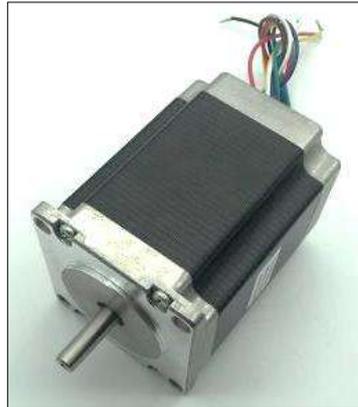


Ilustración 13-3: Motores Nema 23

Fuente: Juiña, P., 2021

3.3.3.1. Driver digital DM556Y

Este driver será usado en el motor paso a paso de 3Nm. Con este elemento se espera baja temperatura, bajo nivel de ruido, baja vibración, alta estabilidad y precisión. Las especificaciones del driver se describen en las siguientes tablas:

Tabla 6-3: Especificaciones eléctricas

Especificaciones Eléctricas	
Corriente pico de salida	1.7 - 5.6 A(1.2~4.0RMS)
Voltaje de entrada	DC 20V~50V
Corriente de señal lógica	7-15 mA (típica 10 mA)
Frecuencia de entrada de pulsos	0 - 200 kHz
Ancho de pulso	2.5μs
Resistencia de aislamiento	500 MΩ

Fuente: OMC StepperOnline, 2018.

Tabla 7-3: Entorno operativo y otras especificaciones

Entorno operativo y otras especificaciones	
Enfriamiento	Refrigeración natural o refrigeración forzada
Temperatura ambiente	0 - 65 °C
Humedad	40% - 90% HR
Temperatura de funcionamiento	50 °C MÁX.

Vibración	10 - 50 Hz/0,15 mm
Temperatura de almacenamiento	(-20) - 65 °C
Peso aproximado	300 g (10.58 oz)

Fuente: OMC StepperOnline, 2018.

En la ilustración 14-3 se muestra el driver para el motor con 3 Nm de torque, puesto que este necesita 24 V para funcionar correctamente.

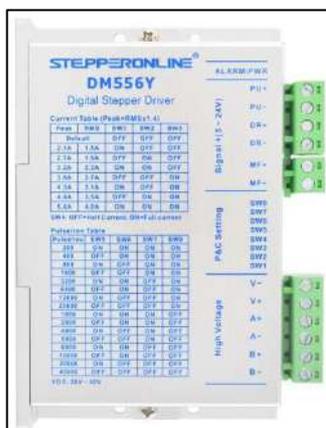


Ilustración 14-3: Driver Digital

Fuente: OMC StepperOnline, 2018.

3.3.4. Driver HY-DIV268N-5A

Este driver se usó en los motores paso a paso de 2Nm, está basado en el sistema TB6600, tiene protecciones contra alto voltaje, bajo voltaje, picos de corriente y cortocircuito. A continuación se detallan sus especificaciones eléctricas y físicas.

Tabla 8-3: Especificaciones eléctricas y físicas.

Especificaciones	
Voltaje de alimentación	12V - 48V
Corriente en la bobina	0.2A – 5A (Regulable)
Temperatura de operación	(-10)°C a 45 °C
Dimensiones	116 x 86 x 33 mm
Frecuencia de entrada de pulsos	0 - 200 kHz
Ancho de pulso	2.5µs
Resistencia de aislamiento	500 MΩ
Peso	220g

Realizado por: Juiña, P., 2021.

En la ilustración 15-3 se muestra el driver para los motores de 2 Nm de torque, puesto que estos necesitan 12 V para funcionar correctamente,



Ilustración 15-3: Driver HY-DIV268N

Fuente: BotLand, 2017.

3.3.5. Accesorios para los motores.

Estos elementos se encargarán de convertir el movimiento circular de los motores en movimiento lineal para accionar los pedales, estos se montarán en una base en conjunto con los motores. Sus características se presentan en la siguiente tabla.

Tabla 9-3: Accesorios para el motor

Elemento	Descripción	Características	Imagen
Acople flexible	Es la manera más fácil de transferir la potencia de los motores. Para que no se presenten complicaciones en la alineación del eje del motor y el eje de transmisión, se escogieron los acoplamientos flexibles.	Dimensiones interiores: 10 mm a 8 mm. Dimensiones: Largo 25mm y Diámetro exterior 18mm Tornillo de fijación (DIN913): M4 (Llave Allen #2) Peso: 15 gramos Material: Aluminio.	
Tornillos sin fin	Mecanizados de 8mm, garantiza un alto grado de precisión en el movimiento, ideal para impresoras 3D y CNC.	Recorrido por vuelta: 8mm. # de hilos: 4. Diámetro: 8mm. Diámetro exterior: Largo: 300mm	

Tuerca para tornillo sin fin	Su mecanizado de alta precisión permite un movimiento fluido y fácil con el tornillo sin fin, esta tuerca cuenta con un diámetro de 8mm.	Material: Latón. Color: Oro. Diámetro del tornillo de avance: 8 mm, 0,31". Paso: 2 mm, 0,08". Dimensiones: 1.5 × 2.2 cm, 0.59 × 0.87 "(Aprox.)	
Chumacera balero	Soportes tipo balero que ayudan a ajustar barras lineales. Diseñadas para ejes con 8mm de diámetro, tolera bastante bien vibraciones y movimiento sin perjudicar al eje.	Fabricante: Haitronic	
Soporte para eje	Proporcionará orientación y soporte, funciona con cualquier eje lineal de 8 mm de diámetro con facilidad, se debe insertar el eje y apretar el tornillo de fijación superior para bloquearlo en su lugar.	Diámetro: 8mm Dimensiones de la base: 14,4 x 41,8 mm Altura: 32,8 mm Material: Aluminio Tornillo prisionero incluido	
Rodamientos	Los rodamientos 608ZZ serán utilizados para el desplazamiento en conjunto con la tuerca y los ejes de soporte laterales.	Diámetro interior: 8mm Diámetro exterior: 22 mm Tamaño: 7 mm Peso: 0,01 kg	
Eje liso	Este eje de acero cuenta con un tratamiento de cromado, gracias al tratamiento superficial tiene una protección adicional a los golpes y rayaduras provocados por la suciedad o polvo y al contacto con piedras	Diámetro: 8mm Longitud: 300mm Material: Aleación de acero GCr15 Acabado: Cromo endurecido 68-72RC	

Realizado por: Juiña, P., 2021.

Una vez seleccionados los motores y los elementos que realizaran el movimiento lineal, se procedió a implementarlos en la estructura cockpit.

3.3.6. Integración de los actuadores al diseño del cockpit

En el mercado existen diferentes tipos de carrocería a los cuales el prototipo podría ser adaptado es por esto por lo que se tomaron medidas del piso del interior de la cabina del conductor de varias carrocerías y de diferentes marcas, con esto se obtuvo dimensiones promedio de ancho y largo.

Tabla 10-3: Dimensiones del piso de las principales carrocerías

N°	Tipo de carrocería	Marca	Modelo	Ancho (cm)	Largo (cm)
1	HB	Renault	Sandero	56	64
2	HB	Hyundai	i10	50	60
4	SUV	Hyundai	Tucson	52	63
5	SUV	Kia	Sportage	54	64
6	Camioneta	Toyota	Hilux	47	56
7	Camioneta	Volkswagen	Amarok	50	62
8	Sedán	Chevrolet	Sail	51	56
9	Sedán	Hyundai	Grand i10	56	62
Promedio				52	61

Realizado por: Juiña, P., 2021.

En la tabla 8-3 se muestra las medidas de los pisos de las carrocerías, de las cuales se obtuvo un promedio con las cuales se dimensionó una base para los motores.

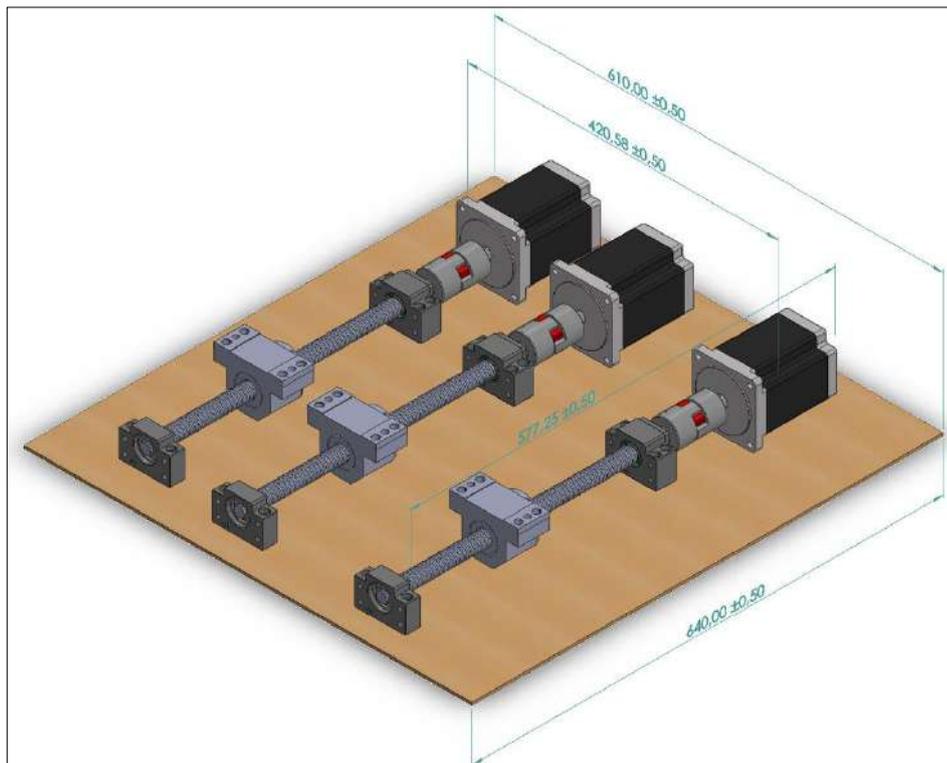


Ilustración 16-3: Base de los motores

Fuente: Juiña, P.,2021.

En la ilustración 16-3 se muestra el diseño de la base en la que los motores descansaran. Una vez distribuidos los motores se hizo necesario la fabricación de un accesorio que permita alcanzar el pedal y moverlo.

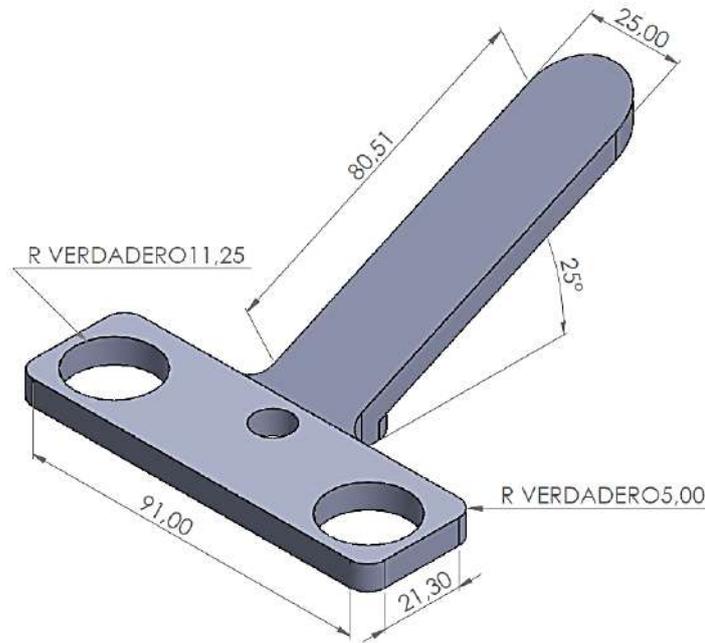


Ilustración 17-3: Accesorio

Realizado por: Juiña, Puli., 2021.

En la ilustración 17-3 se muestra el accesorio que se diseñó en SolidWorks, y gracias a la tecnología de impresión 3D, se pudo obtener dicha pieza bajo las medidas que el prototipo requería, una vez impreso se procedió a añadir los rodamientos que fueron adaptados a los ejes lisos permitiendo así el movimiento guiado como si se tratase de un carrusel, además se incluyó la tuerca para el tornillo sin fin la cual transforma el movimiento circular de los motores y lo convierte en movimiento lineal. Para escoger el tipo de impresión 3D que más se ajustaba a los requerimientos de la pieza diseñada se comparó las dos tecnologías más usadas en el mercado.

Tabla 11-3: Comparativa de los tipos de impresión 3D

	Impresión 3D por FDM	Impresión 3D por SLA
Ventajas	Disponibilidad de máquinas y materiales a bajo costo.	Alta precisión. Superficies con mejores acabados.
Desventajas	Precisión reducida. Baja calidad en los detalles.	Piezas terminadas requieren de curado. Sensibles antes exposiciones largas a luz UV.

Aplicaciones	Fabricación de modelos y piezas de bajo costo.	Fabricación de piezas y prototipos funcionales. Creación de modelos y moldes para fundición.
Tipos de material	Termoplásticos de uso industrial y en ingeniería como: PLA, PETG, ABS, TPU.	Resinas termoestables, estándar, para ingeniería parecida al polipropileno, industriales, biocompatibles.

Fuente: Rúa, y otros, 2017.

Realizado por: Juiña, P., 2021

La impresión por FDM tiene las características suficientes para la fabricación del accesorio, puesto que no requiere de una precisión tan alta ni de acabados superficiales con tantos detalles, el tipo de material utilizado en la pieza es ABS, porque es un material que ofrece alta resistencia y dureza.

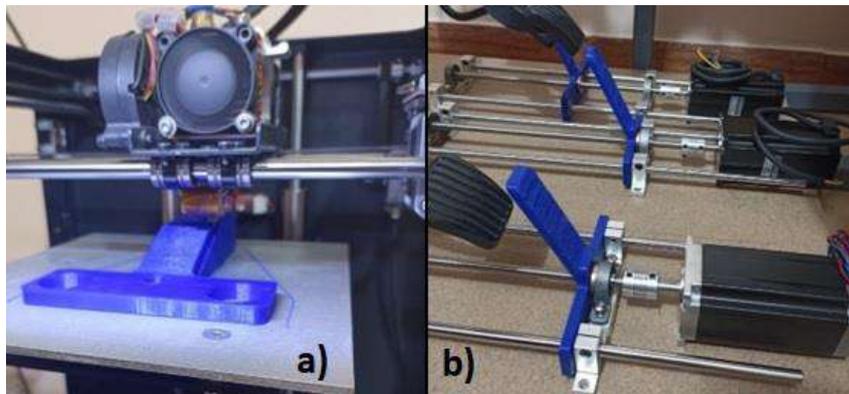


Ilustración 18-3: Distribución de los motores.

Realizado por: Juiña, P., 2021

En la ilustración 18-3 el literal “a” muestra el proceso de impresión 3D, mientras que el literal “b” muestra a los accesorios ya acoplados a los motores.

3.3.7. Elementos para el control manual

Estos componentes hacen posible el control de los actuadores y su energización, han sido seleccionados en base al tipo de alimentación que se requiere y que se va a distribuir en el prototipo.

3.3.8. Fuentes conmutadas 12 V - 10 A y 24V - 10A

Estas fuentes de alimentación energizan los motores nema 23 y demás componentes que necesiten energizarse, transforman la corriente alterna en corriente directa. En las siguientes imágenes se pueden visualizar las fuentes de poder, se detallan sus especificaciones a continuación:

Tabla 12-3: Fuentes conmutadas

Características	Fuente 12V	Fuente 24V
Marca	EVL	STV
Modelo	TPS-2410	S-120-12
Entrada	110 - 220 V AC / 50-60 Hz	110 – 220V AC / 50 – 60 Hz
Salida	24 V DC / 10 A	12 V DC / 10 A
Borneras de conexión	6	4
Carcasa	Metálica con ventilación	Metálica con ventilación
Regulador de voltaje	Si	Si
Imagen		

Realizado por: Juiña, P., 2021.

3.3.9. Guantes para conducción

Son un accesorio que garantiza la seguridad al momento de conducir un automóvil durante mucho tiempo. Son excelentes protegiendo las manos del conductor en caso de algún accidente.

Características:

- Material: poliéster
- Superficie: Antideslizante
- Dimensiones: 22cm x 12cm.
- Color: Azul
- Otras características: Cierre velcro en las muñecas

Para el prototipo también cumplirá la función de ser un soporte en el que se colocaron los pulsadores, facilitando así la adaptación y la conducción, cuenta con una superficie antideslizante que brinda mayor comodidad y seguridad al momento de maniobrar el volante y caja de cambios.



Ilustración 19-3: Guantes para conducción

Fuente: Rivera, 2023.

3.3.10. Raspberry pi

El uso de este hardware se hace necesario porque al ser una minicomputadora permite el despliegue de una interfaz gráfica que será el medio por el cual el usuario pueda interactuar con la máquina, escogiendo mediante esta las configuraciones que necesite.

En las siguientes ilustraciones se pueden visualizar dos modelos de Raspberry Pi específicamente los modelos Pi 3b+ y Pi 4, con estos dos microcontroladores se realiza un análisis comparativo de la especificación técnicas que brindan cada una de ellas, y posteriormente se elige el modelo que mejor se ajuste a los requerimientos del prototipo.

Se consideraron dos modelos de Raspberry Pi para el desarrollo de este prototipo. Con el análisis comparativo se busca determinar el modelo de Raspberry Pi adecuado. Se tuvo en cuenta las especificaciones de cada placa Raspberry Pi para el análisis.

Tabla 13-3: Comparativa entre Raspberry Pi modelo 3 B+ y 4

Especificaciones	Raspberry Pi	
	Modelo 3 B+	Modelo 4 B
Procesador	Quad-Core 1.2GHz Broadcom BCM2837	Quad-Core 1.5 GHz Broadcom BCM2711
RAM	1 GB	Hasta 8 GB
Bluetooth	4.0	5.0
Puertos USB	4 x USB 2.0	2 x USB 2.0 2 x USB 3.0

Conexión Wireless	Si	Banda 2.4 GHz Banda 5 GHz
Puertos Display	1 x HDMI 1 x DSI	2 x mico HDMI 1 x DSI
Fuente de Alimentación	Micro USB y GPIO hasta 2.5A	5V DC por USB-C y GPIO (3A)
Puerto Micro SD	Si	Si
Precio	\$35	\$35 (1 GB RAM) \$45 (2 GB RAM) \$55 (4 GB RAM) \$75 (8 GB RAM)

Fuente: Raspberry Pi, 2016. & Raspberry Pi, 2020.

Realizador por: Juiña, P., 2021.

En la tabla anterior se enumera las especificaciones de cada modelo de Raspberry: Pi 3 B+ y Pi 4 B. Con el análisis comparativo, se determina que el diseño de la tecnología de la placa Raspberry Pi 3 B+ cumple con los requisitos del prototipo, ya que sus características son suficientes para controlar los componentes individuales del sistema. Con esta podemos procesar información de entrada que coloque el usuario.



Ilustración 20-3: Raspberry pi 3 B+

Fuente: Raspberry Pi, 2016.

En la ilustración 20-3 se muestra la placa por la que finalmente se optó para la creación de la interfaz gráfica que permitirá la configuración del prototipo .

3.3.11. Arduino

Este hardware ha sido seleccionado puesto que permite un control sencillo de los actuadores, su diseño robusto sirve además como una medida de protección para las entradas de la Raspberry en caso de algún inconveniente con el entorno físico puesto que su reemplazo es más barato, es necesario utilizar una placa por cada motor para permitir el control independiente de estos.

En las siguientes imágenes se puede ver dos modelos de placas de Arduino: UNO y MEGA, con estos dos microcontroladores y las especificaciones que brindan se realiza un análisis de sus características.

Se consideraron dos modelos de Arduino para desarrollar el prototipo. A través de un análisis comparativo se pretende identificar el microcontrolador ideal que se adecue al trabajo propuesto. Se tomaron en cuenta las especificaciones de cada modelo de Arduino para su correcto análisis.

Tabla 14-3: Comparativa entre la placa de Arduino UNO y MEGA.

Característica	Arduino UNO	Arduino MEGA
Microcontrolador	ATmega328	ATmega 2560
Memoria Flash	32 KB	256 KB
Memoria SRAM	2 KB	8 KB
Memoria EEPROM	1 KB	4 KB
Pines Digitales	14	54
Pines Analógicos	12	31
Puertos	1	4
Voltaje de Alimentación	5V	5V
Velocidad de reloj	16 MHz	16 MHz
Tamaño	2, 7" x 2, 1"	4" x 2,09"
Precio	\$19,99 – \$23,00	\$36,61 – \$39,99

Fuente: Raspberry Pi, 2016. & Raspberry Pi, 2020.

Realizador por: Juiña, P., 2022.

En la tabla anterior, se detallan las especificaciones de cada placa Arduino. Con el análisis comparativo se determinó que la placa Arduino UNO es el microcontrolador que cumple con los requisitos técnicos del prototipo, ya que las características proporcionadas son suficientes para interpretar las señales que la placa Raspberry Pi enviará para el control de los motores.

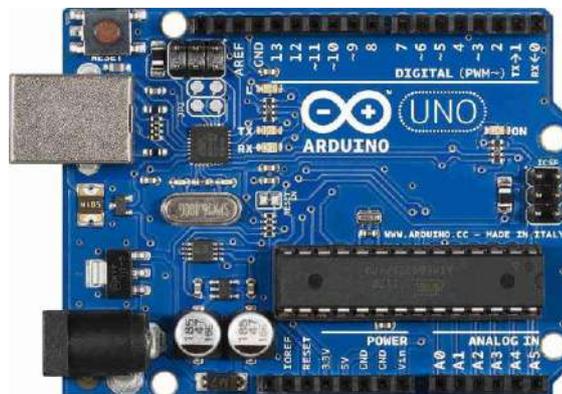


Ilustración 21-3: Placa de Arduino UNO

Fuente: Arduino, 2022.

En la ilustración 21-3 se muestra la placa de Arduino Uno seleccionada para su uso en el prototipo, esta se enlaza con la raspberry y sirve como un sistema de conexión de los motores con la raspberry, además de servir como sistema de protección para la misma.

3.3.12. Distribución de los elementos de control

En este apartado se realizó la distribución de los elementos en una caja de control, esto sirvió para proteger la electrónica implementada en el prototipo, estos actuaran sobre los pedales del cockpit. En la ilustración 22-3 se representa las conexiones que se realizaron entre todos los elementos electrónicos y electromecánicos.

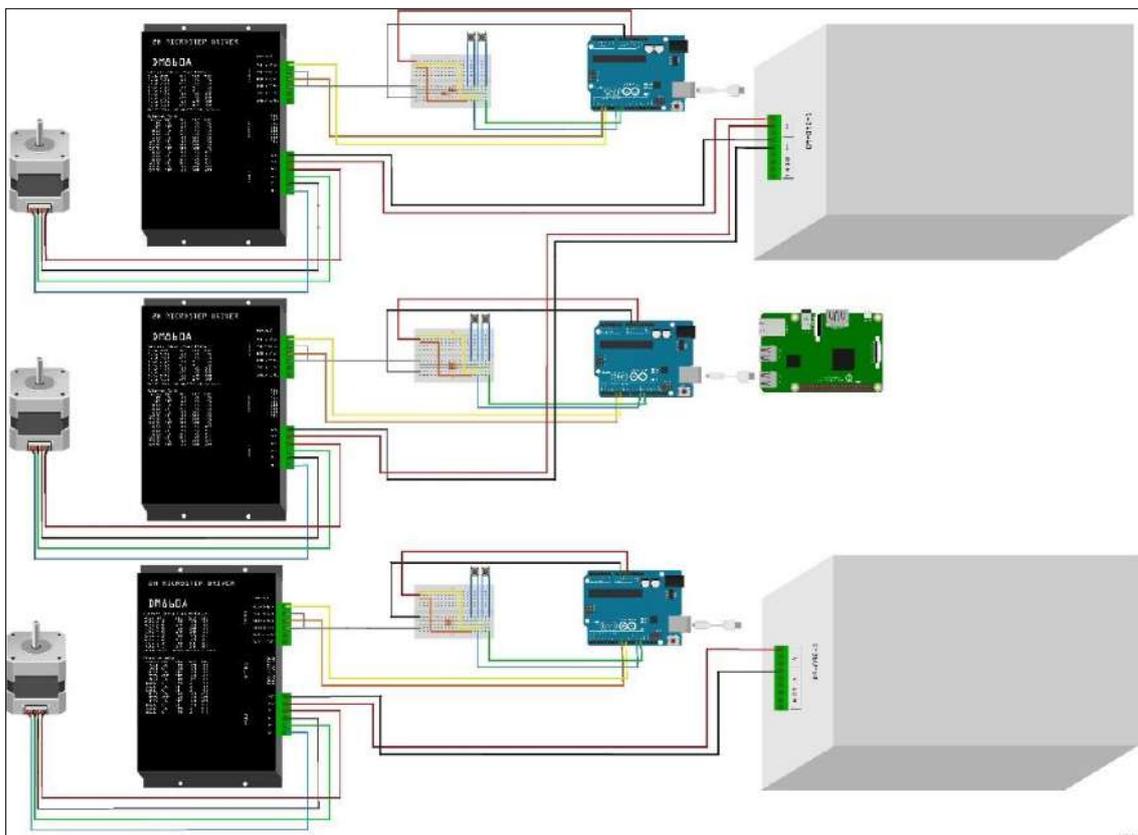


Ilustración 22-3: Esquema de conexiones.

Realizado por: Juiña, P., 2021.

Esto se complementa con la siguiente tabla en la que se describen las conexiones de las placas Arduino con cada uno de los drivers de los motores que corresponden a cada pedal.

Tabla 15-3: Tabla de Conexiones

Conexiones Placa Arduino	
Acelerador	
Pin 2	Dir (+) del driver DIV 268N
Pin 3	Pul (+) del driver DIV 268N
Pin 6	Pulsador Avance
Pin 7	Pulsador Retroceso
Pin GND	Pul (-) del driver
Pin 5V	Pulsadores para avance y retroceso
Freno	
Pin 4	Dir (+) del driver DIV 268N
Pin 5	Pul (+) del driver DIV 268N
Pin 10	Pulsador Avance
Pin 11	Pulsador Retroceso
Pin GND	Pul (-) del driver
Pin 5V	Pulsadores para avance y retroceso
Embrague	
Pin 8	Dir (+) del driver DM 556Y
Pin 9	Pul (+) del driver DM 556Y
Pin 12	Pulsador Avance
Pin 13	Pulsador Retroceso
Pin GND	Pul (-) del driver
Pin 5V	Pulsadores para avance y retroceso

Realizado por: Juiña, P., 2021

En la ilustración 23-3 se muestra el diagrama de conexiones eléctricas del prototipo, donde los motores y la raspberry están conectados a la red eléctrica, en cambio el Arduino se alimenta de la raspberry mediante el puerto USB.

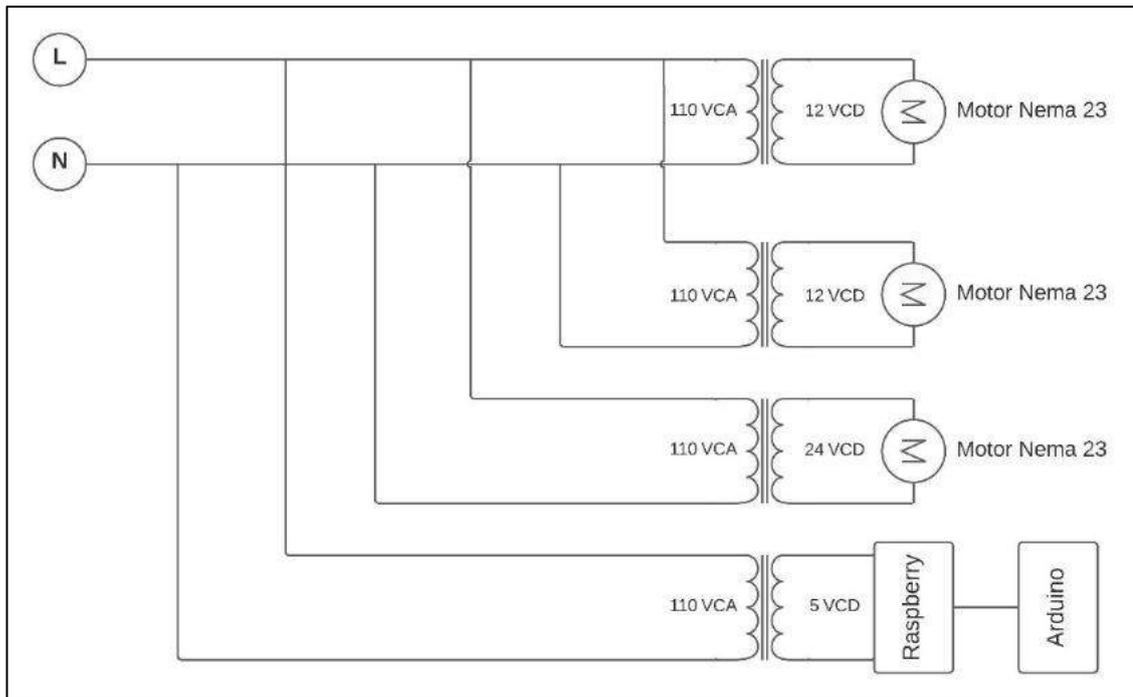


Ilustración 23-3: Diagrama de conexiones

Fuente: Juiña, P., 2021.

Microswitch fueron colocados en la yema de los dedos de los guantes para facilitar su acceso, se distribuyeron de la siguiente manera:

Mano derecha: pulsos de avance de motores.

- Dedo Índice: Pedal Acelerador
- Dedo Medio: Pedal Freno
- Dedo Anular: Pedal de Embrague

Mano Izquierda: pulsos de retroceso de motores.

- Dedo Índice: Pedal Acelerador
- Dedo Medio: Pedal Freno
- Dedo Anular: Pedal de Embrague



Ilustración 24-3: Distribución de los pulsadores

Fuente: Juiña, P., 2021

En la ilustración 24-3 se muestra cómo se distribuyen los pulsadores, estos se conectan a una protoboard mediante cable UTP con una longitud de 2,5 m, asegurándonos así movimientos libres al momento de maniobrar el volante.



Ilustración 25-3: Distribución de los dispositivos de control.

Realizado por: Juiña, P., 2021.

En la ilustración 25-3 se muestra el proceso de armado de la caja de control con la electrónica de control distribuida.

3.4. Ensamble completo del prototipo

Este proceso termina con el ensamblaje completo del prototipo, uniendo así el cockpit con la caja de control como se muestra en la siguiente ilustración. En la ilustración 26-3 se muestra el

ensamble completo del prototipo que une el cockpit con los motores y la caja de control. Los elementos se describen en la siguiente tabla.



Ilustración 26-3: Prototipo ensamblado

Fuente: Juiña, P., 2021.

Tabla 16-3: Ensamble

N°	Elemento
1	Asiento
2	Guantes
3	Caja de control
4	Base con los motores
5	Pedales
6	Volante
7	Soporte

Realizado por: Juiña, P., 2021.

3.5. Algoritmos de control

En este apartado se desarrollaron los algoritmos que controlan el prototipo, primero se procedió con una breve descripción de los entornos de programación disponibles, para después proceder con la programación de las placas.

Comunicación Raspberry-Arduino

La comunicación serial, es un protocolo que se usa para el intercambio de datos, se realiza mediante 2 señales digitales, por un lado el pin “Tx” que se encarga de la transmisión de datos y el pin “Rx” que se encarga de la recepción, estos están presentes tanto en la placa de Arduino como la raspberry, otra alternativa para la comunicación en serie es simular la conexión mediante el puerto USB que se encuentra en ambas placas, esta alternativa tiene que ser programada tanto en Arduino como en Python.

El primer paso para que a un programa le sea posible intercambiar datos mediante la conexión serial, esta debe ser habilitada en la raspberry mediante raspi-config en la línea de comandos terminal de la placa.

Después se procede a identificar el nombre del puerto serial de la placa Arduino que ha sido que determinado por la raspberry mediante su sistema operativo Raspbian, estos aparecen en conjunto con los demás elementos del sistema, haciendo uso del comando “dev” se podrá conocer el nombre de los puertos seriales del Arduino.

3.5.1. Funcionamiento general del prototipo

El prototipo está conformado por dispositivos que se han conectado de tal forma que se comunican entre sí de manera serial, están dispuestos de esta manera para lograr su propósito de ser una asistencia en la conducción para personas con paraplejia.

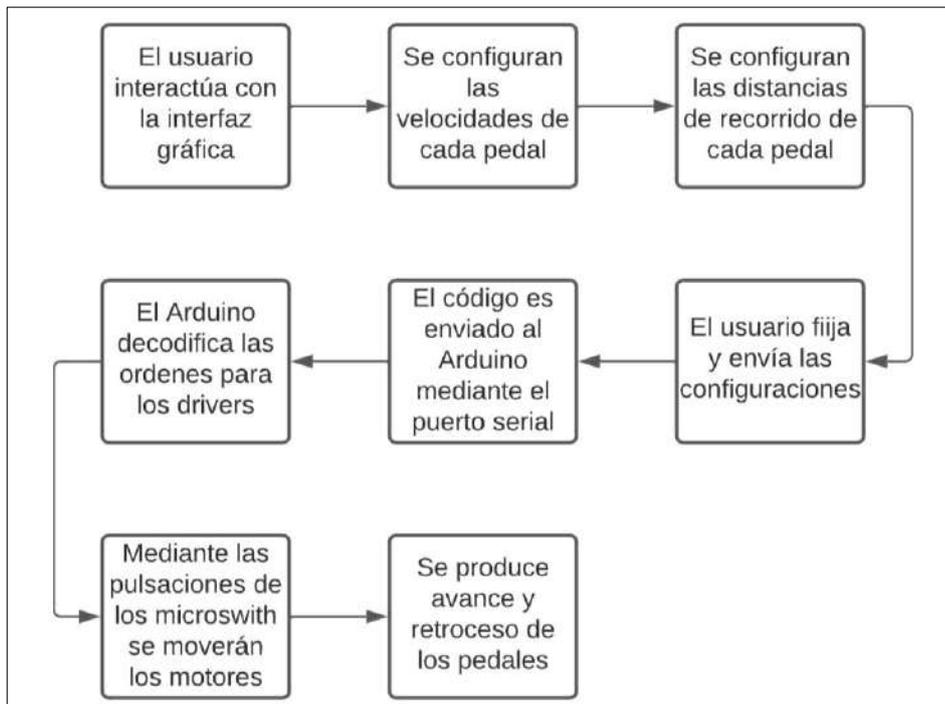


Ilustración 29-3: Secuencia de funcionamiento del prototipo

Realizado por: Juiña, P., 2021.

3.5.2. Algoritmo para la interfaz gráfica

Primero la raspberry va a comprobar si los puertos seriales están disponibles y les va a asignar un nombre, esto es importante porque al momento de conectar los Arduino, deben hacerse en orden porque cada puerto está programado para un pedal en específico, el orden en que se los debe conectar está explicado en el manual de usuario.

Después se procede a configurar las velocidades de cada pedal, esta se realiza en una escala de 0 a 100 con intervalos seleccionables múltiplos de 10, una vez se haya escogido la velocidad deseada se debe presionar el botón “setear”, seguido se procede a configurar las distancias de cada pedal de la misma forma esto se realiza con una escala de 0 a 100 con intervalos seleccionables múltiplos de 5 lo que asegura una mayor precisión en el movimiento, como las velocidades ya están fijadas, los motores se moverán acorde con lo configurado, se debe pulsar el botón “setear”, con esto se guardan las distancias. Para que estos parámetros queden fijos en los Arduino se debe pulsar el botón “empezar programa”. En caso de que se haya cometido algún error al momento de configurar se debe reiniciar el programa.

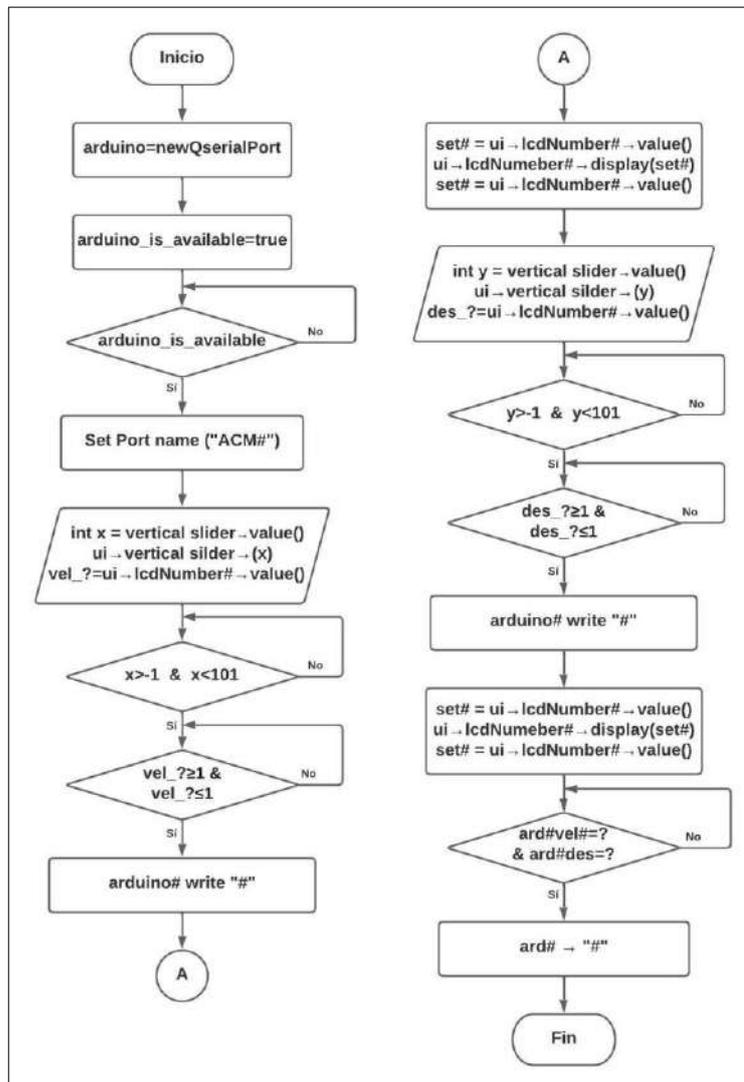


Ilustración 30-3: Diagrama de flujo de la interfaz gráfica

Realizado por: Juiña, P., 2021.

En la ilustración 30-3 se representa mediante un diagrama de flujo, el algoritmo desarrollado para la interfaz gráfica prototipo con las partes que requiere el programa para su correcto funcionamiento. Este código se programó en una Raspberry Pi 3 mediante el programa de desarrollo Qt Creator.

Mediante este software es posible la programación del código con la que se creó la interfaz gráfica y que permite la configuración del prototipo

3.5.3. Algoritmo de control con arduino

El programa va a leer los valores de velocidad que se ingresan y los va a comparar con las que tiene guardadas en su programación, va a realizar esto hasta encontrar el mismo valor, una vez que lo haya encontrado procederá a leer los valores de las distancias que se ingresan y de la misma

manera los comparará con las variables que tiene guardadas en su programación, cuando las dos configuraciones son enviadas estas entran en un bucle haciendo que con cada pulso que se realice con los microswitch va a ser leído constantemente, tanto si es para avance o retroceso de los motores.

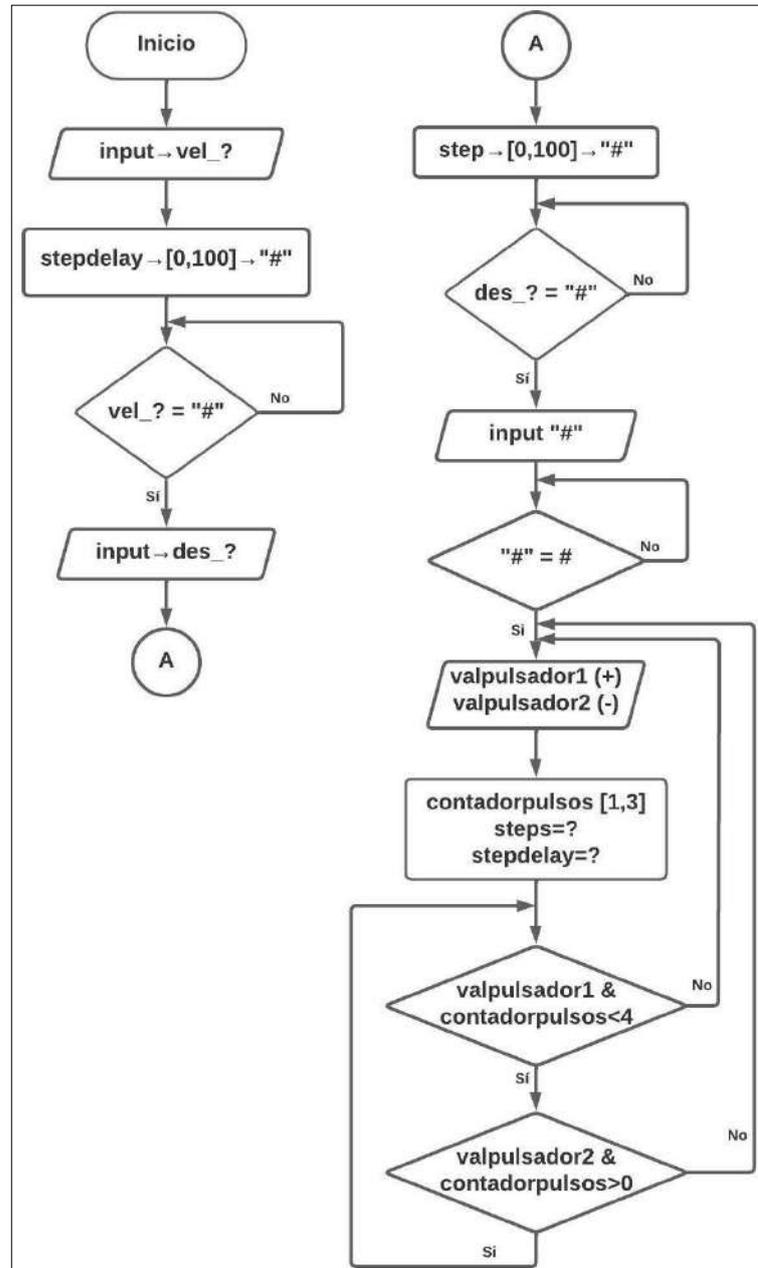


Ilustración 31-3: Diagrama de flujo de la programación en Arduino

Fuente: Juiña, P., 2021

En la ilustración 31-3 se muestra el código programado en las placas Arduino, este se integra a la interfaz gráfica, complementándose y trabajando a la par.

CAPÍTULO IV

4. RESULTADOS

4.1. Ángulos de confort para un puesto de conducción

Se tomó como base la tabla 4 del capítulo 2 sobre ángulos de confort, se hizo una medición de los arcos correspondientes en las posturas de tres personas, con estaturas de 165 cm, 170 cm y 175 cm, obteniendo los resultados y comparándolos al final en una tabla, para poder determinar si la estructura cockpit con un soporte añadido para los pies, cumplen con estos ángulos de confort.

Procedimiento:

Cada persona adoptó una postura de conducción en la que se sintió cómoda, se procedió entonces con la captura fotográfica que se analizó mediante la herramienta *RULER* de Ergonautas.

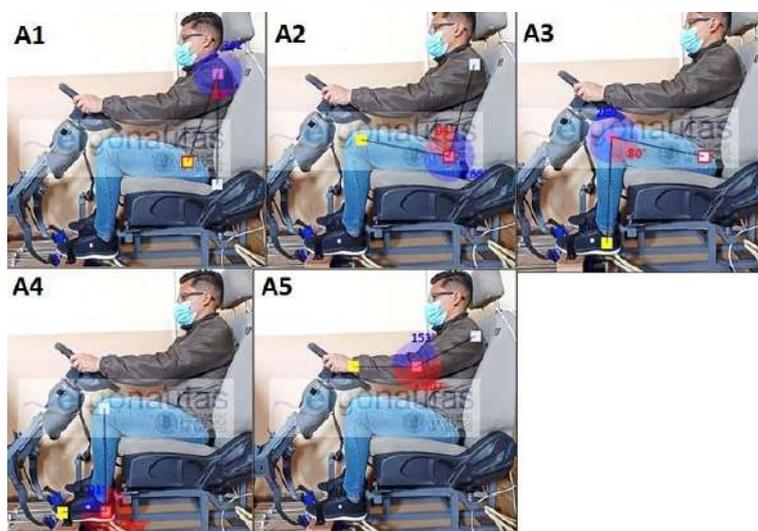


Ilustración 1-4: Persona de 165cm – A5

Realizado por: Juiña, P., 2021

Tabla 1-4: Ángulos obtenidos de una persona de 165 cm de estatura.

Zona corporal	Persona 165 cm.
Espalda	18°
Piernas	94°
Rodilla	80°
Pie	91°
Brazo	153°

Realizado por: Juiña, P., 2021.

En la tabla 1-4 se detallan los resultados obtenidos, estos ángulos se midieron en diferentes zonas corporales correspondientes a una persona de 165 cm de estatura.

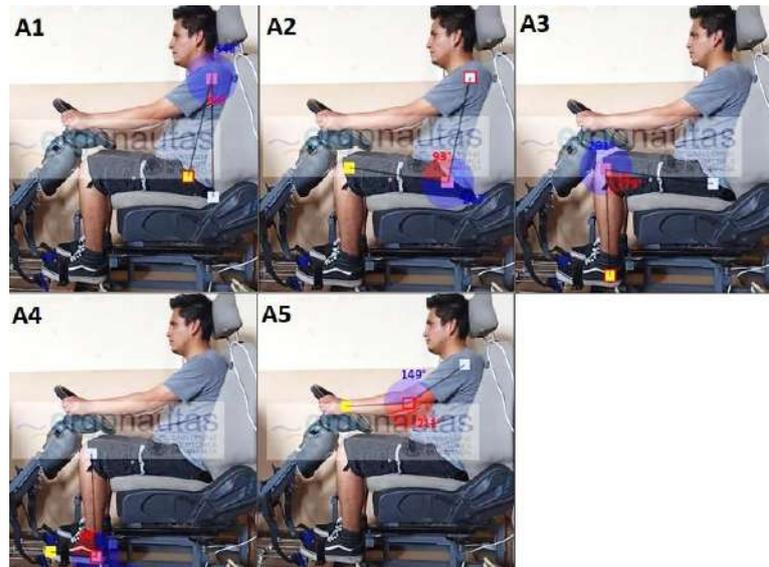


Ilustración 2-4: Persona de 170cm – A5

Fuente: Juiña, P., 2021.

Tabla 2-4: Ángulos obtenidos de una persona de 170 cm de estatura

Arco	Persona 165 cm.
Espalda	14°
Piernas	93°
Rodilla	79°
Pie	80°
Brazo	149°

Realizado por: Juiña, P., 2021.

En la tabla 2-4 se detallan los resultados obtenidos, estos ángulos se midieron en diferentes zonas corporales correspondientes a una persona de 170 cm de estatura

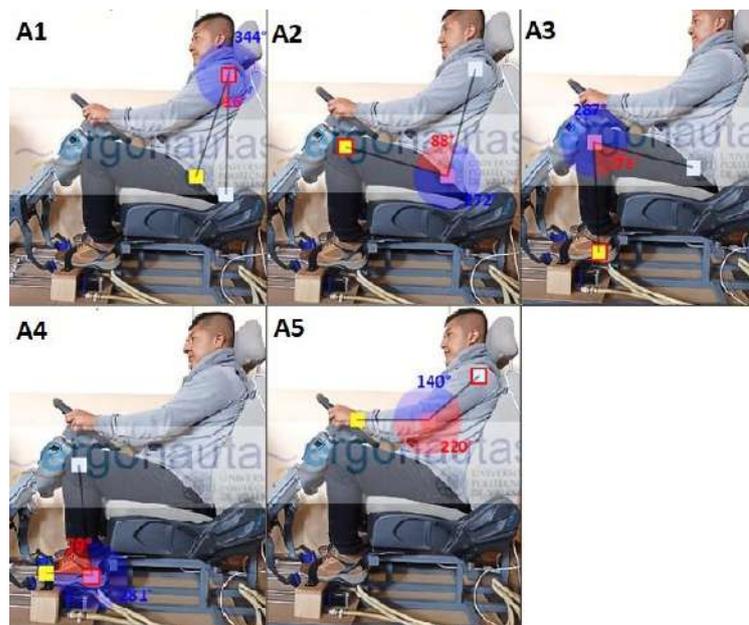


Ilustración 3-4: Persona de 175cm – A5

Fuente: Juiña, P., 2021.

Tabla 3-4 Ángulos obtenidos de una persona de 175 cm de estatura

Arco	Persona 165 cm.
Espalda	16°
Piernas	88°
Rodilla	73°
Pie	79°
Brazo	140°

Realizado por: Juiña, P., 2021.

En la tabla 1-4 se detallan los resultados obtenidos, estos ángulos se midieron en diferentes zonas corporales correspondientes a una persona de 175 cm de estatura.

Tabla 4-4: Tabla comparativa de resultados

Postura	Zona Corporal	Arco	P1 (165cm)	P2 (170cm)	P3 (175cm)
	Espalda	10° < A1 < 20°	18°	14°	16°
	Piernas	90° < A2 < 110°	94°	93°	88°
	Rodilla	80° < A3 < 120°	80°	79°	73°

	Pie	90° <	91°	80°	79°
	Brazo	A4 < 110°			
		80° <	153°	149°	140°
		A5 < 160°			

Realizado por: Juiña, P., 2021.

Con esta medición se puede ver de mejor manera que el cockpit diseñado es funcional con una persona de 165 cm de estatura, los ángulos medidos muestran una postura aceptable para el usuario donde las tensiones van a ser mínimas, con la persona de 170 cm de estatura se observa que ciertos ángulos están dentro de los rangos permitidos pero que en los ángulos A3 y A4 se empieza a tomar postura no adecuadas que pueden generar mayores tensiones, con la persona de 175 cm, los ángulos A2, A3 y A4 están fuera de los rangos establecidos la postura que se adopta puede llegar a resultar incómoda, además que la maniobrabilidad con el cockpit va a resultar entorpecida.

4.2. Análisis estático

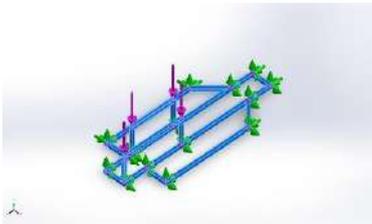
4.2.1. Análisis estructural

Con el estudio estático de las piezas que van a ser sometidas a esfuerzos mecánicos se pueden conocer las propiedades físicas de la estructura cockpit y del accesorio diseñado para los pedales, este estudio se realizó mediante la herramienta de *SolidWorks: Simulation*. Esta extensión permite el cálculo de cómo reaccionan ante estas cargas los volúmenes, la densidad de la estructura y la elasticidad de los materiales que previamente se asignaron con el software. Con el estudio se puede definir las cargas máximas y el tipo de deformaciones que se generan en el material.

4.2.2. Análisis estático – estructura cockpit

Mediante el análisis estático se comprobó la resistencia de la estructura del cockpit que se encarga de soportar a la silla (20kg) y a una persona promedio (74,2kg). Esto supone un peso de 925 N que se va a ejercer en la estructura.

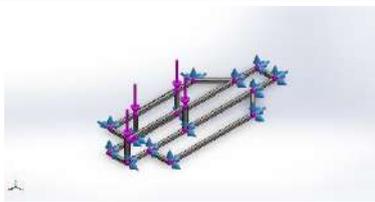
Tabla 5-4: Propiedades mecánicas

Referencia de modelo	Propiedades	Componentes
	Nombre: AISI 1020 Acero laminado en frío	Sólido 1
	Tipo de modelo: Isotrópico elástico lineal	Sólido 2
	Criterio de error predeterminado: Desconocido	Sólido 3
	Límite elástico: 3,5e+08 N/m²	Sólido 4
	Límite de tracción: 4,2e+08 N/m²	Sólido 5
	Módulo elástico: 2,05e+11 N/m²	Sólido 6
	Coefficiente de Poisson: 0,29	Sólido 7
	Densidad: 7.870 kg/m³	Sólido 8
	Módulo cortante: 8e+10 N/m²	Sólido 9
	Coefficiente de dilatación térmica: 1,2e-05 /Kelvin	Sólido 10
		Sólido 11
		Sólido 12
		Sólido 13
	Sólido 14	
	Sólido 15	
	Sólido 16	

Realizado por: Juiña, P., 2022.

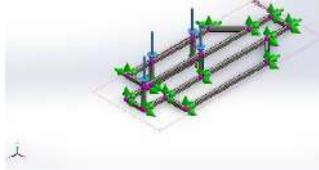
Antes de realizar el estudio se deben determinar los puntos donde se van a hacer las sujeciones que van a soportar al accesorio. Además de definir la parte que va a recibir la carga.

Tabla 6-4: Puntos de sujeción

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción
Inamovible-1		Entidades: 15 Juntas Tipo: Inamovible (sin traslación)

Realizado por: Juiña, P., 2022.

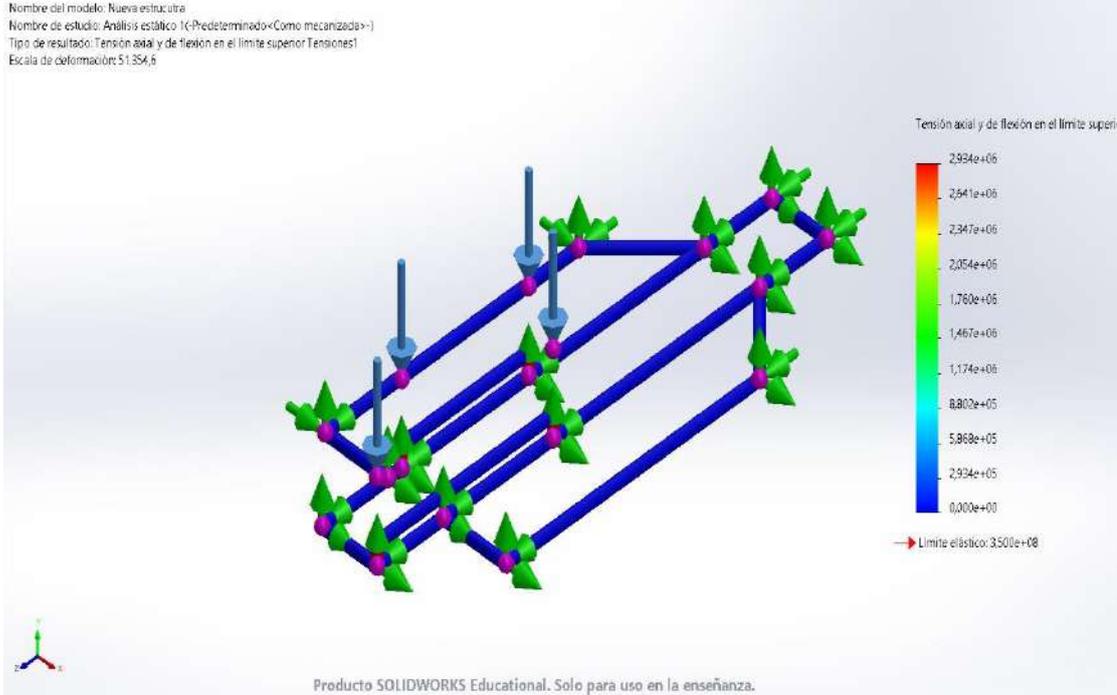
Tabla 7-4: Cargas

Nombre de carga	Cargar imagen	Detalles de carga	
Fuerza-1		Entidades: Referencia: Tipo: Valores: Momentos:	1 plano(s), 4 Juntas Planta Aplicar fuerza ---; ---; -.925 N ---; ---; --- N.m

Realizado por: Juiña, P., 2022.

El análisis estático se realizó usando una fuerza de 785 N y se consiguió la siguiente información:

Tabla 8-4: Tensiones

Nombre	Tipo	Mín.	Máx.
Tensiones1	Tensión axial y de flexión en el límite superior	0,000e+00N/m ² Elemento: 1	3,457e+06N/m ² Elemento: 11
 <p>Nombre del modelo: Nueva estructura Nombre de estudio: Análisis estático 1-(Predeterminado<Como mecanizada>-) Tipo de resultado: Tensión axial y de flexión en el límite superior Tensiones1 Escala de deformación: 51.354,8</p> <p>Tensión axial y de flexión en el límite superior</p> <p>3,457e+06 2,941e+06 2,347e+06 2,054e+06 1,760e+06 1,467e+06 1,174e+06 8,802e+05 5,868e+05 2,934e+05 0,000e+00</p> <p>→ Límite elástico: 3,500e+08</p> <p>Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.</p> <p>Nueva estructura-Análisis estático 1-Tensiones-Tensiones1</p>			

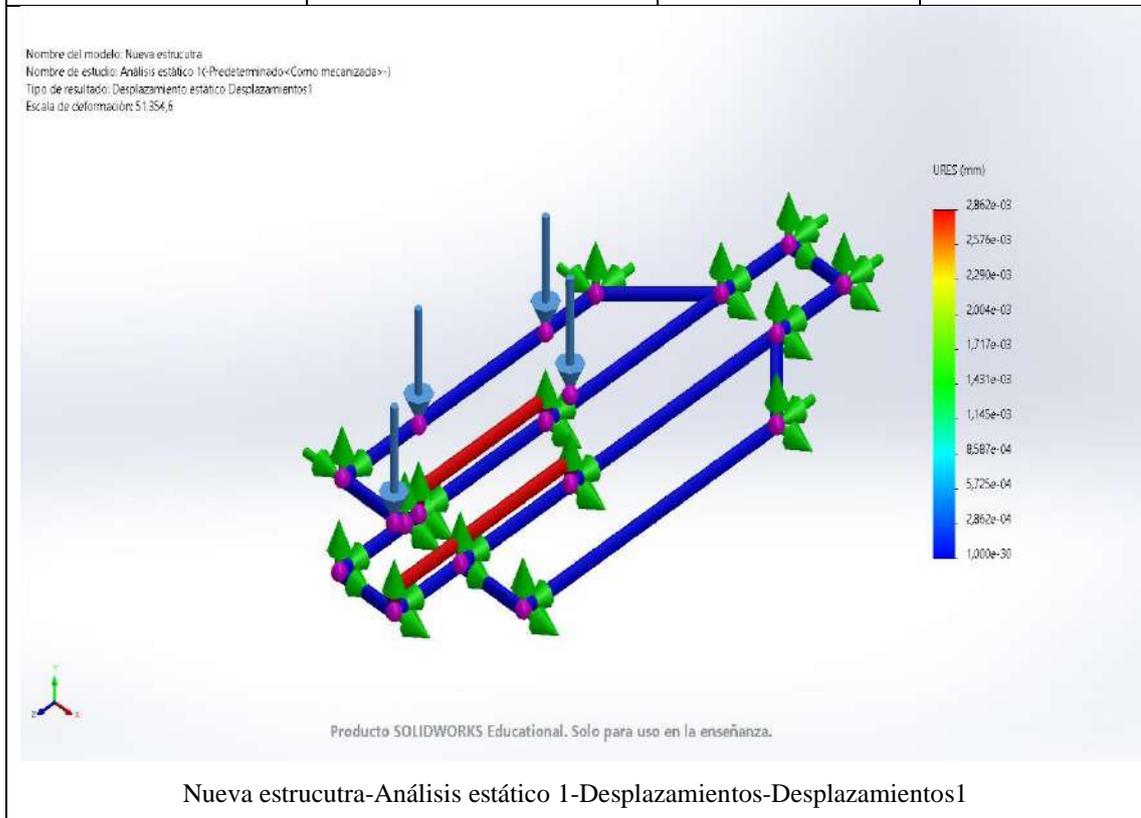
Realizado por: Juiña, P., 2022.

El análisis nos indica que la estructura fue sometida una tensión máxima de $3,457e+06 \text{ N/m}^2$, y esta es menor que el límite elástico de $3,5e+08 \text{ N/m}^2$. Se puede concluir que la estructura y el

material van a resistir la fuerza a la que va a estar sometida sin que esto vaya a representar una afectación a su forma o características.

Tabla 9-4 Desplazamientos

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0,000e+00mm Nodo: 1	3,373e-03mm Nodo: 6



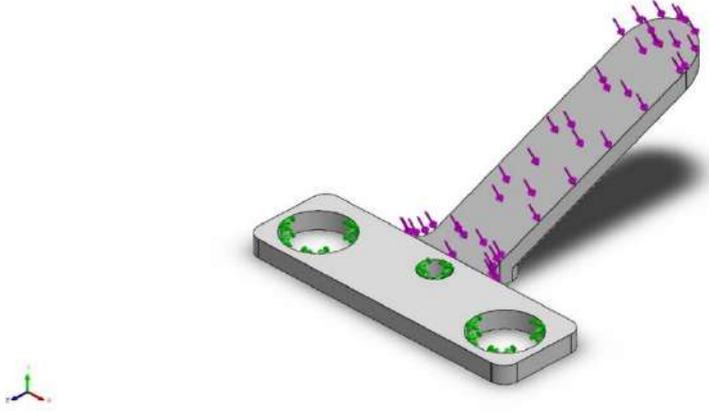
Realizado por: Juiña, P., 2022.

De acuerdo con el análisis la estructura va a lograr su mayor deformación en la parte superior donde se aplica la fuerza, su desplazamiento máximo será de 3,373e-03mm, este valor es muy bajo y representa un desplazamiento que puede ser despreciado considerando los esfuerzos a los que será sometida la estructura.

4.2.3. Análisis estático – accesorio para los pedales

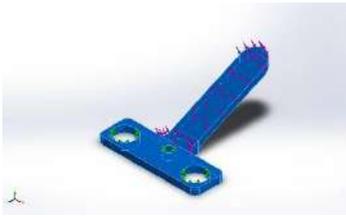
Con el análisis se enumeran las diferentes propiedades mecánicas y volumétricas que tiene el accesorio para los pedales, el material está definido como ABS.

Tabla 10-4: Propiedades volumétricas

 <p>Nombre del modelo: PisaPedal Configuración actual: Predeterminado</p>			
Sólidos			
Nombre de documento y referencia	Tratado como	Propiedades volumétricas	Ruta al documento/Fecha de modificación
 <p>Redondeo2</p>	Sólido	Masa:0,0331034 kg Volumen:3,24545e-05 m ³ Densidad:1.019,99 kg/m ³ Peso:0,324413 N	C:\Users\jumpe\Desktop\PisaPedal.SLDPRT Nov 23 21:43:03 2021

Realizador por: Juiña, P., 2021.

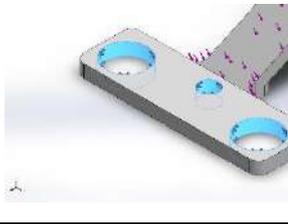
Tabla 11-4: Propiedades mecánicas

Referencia de modelo	Propiedades	Componentes
	Nombre: ABS Tipo de modelo: Isotrópico elástico lineal Criterio de error Desconocido predeterminado: Límite de tracción: 3e+07 N/m² Módulo elástico: 2e+09 N/m² Coeficiente de Poisson: 0,394 Densidad: 1.020 kg/m³ Módulo cortante: 3,189e+08 N/m²	Sólido (Redondeo2)(PisaPedal)
Datos de curva:N/A		

Realizador por: Juiña, P., 2021.

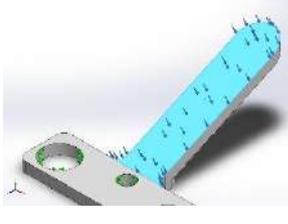
Antes de realizar el estudio se determinaron los puntos donde se van a hacer las sujeciones que van a soportar al accesorio. Además de definir la parte que va a recibir la carga.

Tabla 12-4: Punto de sujeción

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción		
Fijo-1		Entidades:	3 cara(s)	
		Tipo:	Geometría fija	
Fuerzas resultantes				
Componentes	X	Y	Z	
Fuerza de reacción(N)	0,00075439	9,0638	4,22367	
Momento de reacción(N.m)	0	0	0	

Realizador por: Juiña, P., 2021.

Tabla 13-4: Cargas

Nombre de carga	Cargar imagen	Detalles de carga		
Fuerza-1		Entidades:	1 cara(s)	
		Tipo:	Aplicar fuerza normal	
		Valor:	10 N	

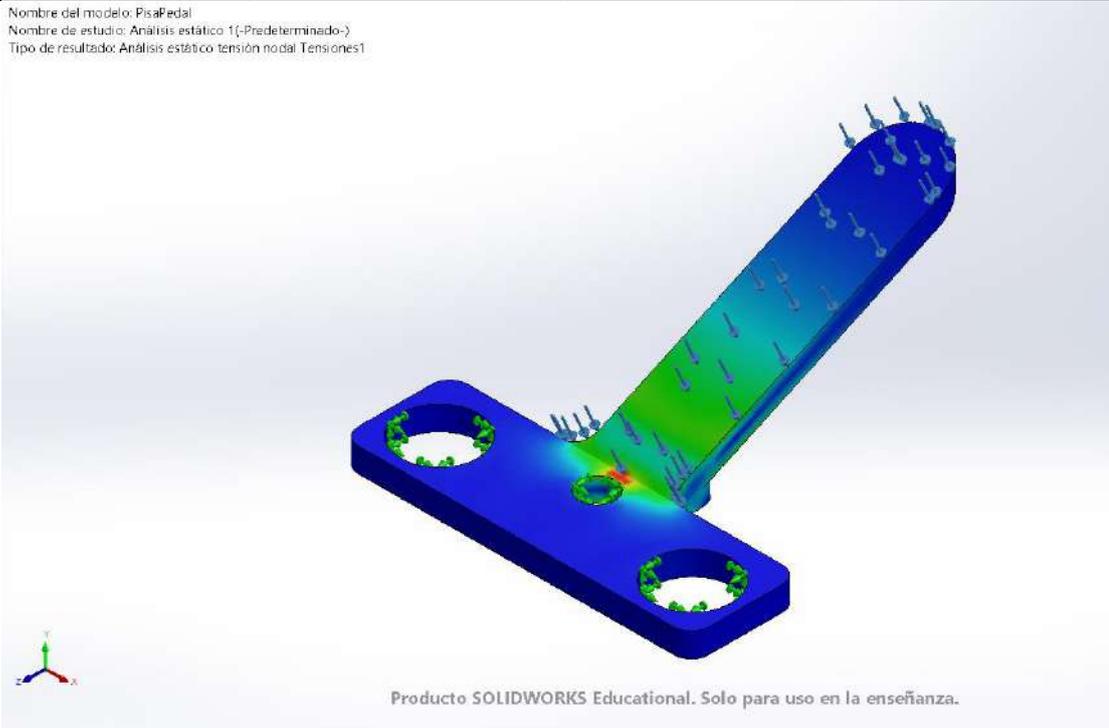
Realizador por: Juiña, P., 2021.

El análisis estático se realizó usando una fuerza de 10 N y se consiguió la siguiente información:

Tabla 14-4: Análisis de tensiones

Nombre	Tipo	Mín.	Máx.
Tensiones1	VON: Tensión de von Mises	2,263e+00N/m ² Nodo: 708	3,252e+06N/m ² Nodo: 11647

Nombre del modelo: PisaPedal
 Nombre de estudio: Análisis estático 1-(Predeterminado-)
 Tipo de resultado: Análisis estático tensión nodal Tensiones1



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

PisaPedal-Análisis estático 1-Tensiones-Tensiones1

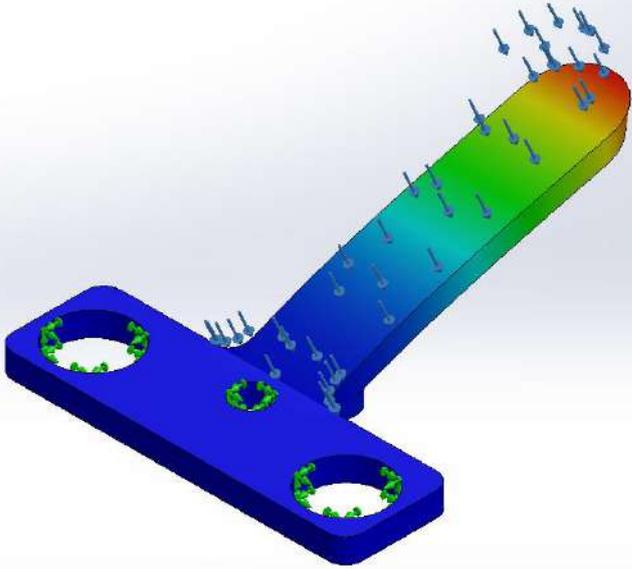
Realizador por: Juiña, P., 2021.

El análisis nos indica que el pedal fue sometido a una tensión mínima de $2,263e+00\text{N/m}^2$ y una tensión máxima de $3,252e+06\text{N/m}^2$, y resistiendo un límite elástico de $2e+09\text{ N/m}^2$. Se puede concluir que el material del accesorio va a resistir la fuerza a la que va a estar sometida sin que esto vaya a representar una afectación a su forma o características.

Tabla 15-4: Análisis de desplazamiento

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0,000e+00mm Nodo: 1	9,746e-01mm Nodo: 791

Nombre del modelo: PisaPedal
 Nombre de estudio: Análisis estático 1(-Predeterminado-)
 Tipo de resultado: Desplazamiento estático Desplazamientos1
 Escala de deformación: 14,0189



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

PisaPedal-Análisis estático 1-Desplazamientos-Desplazamientos1

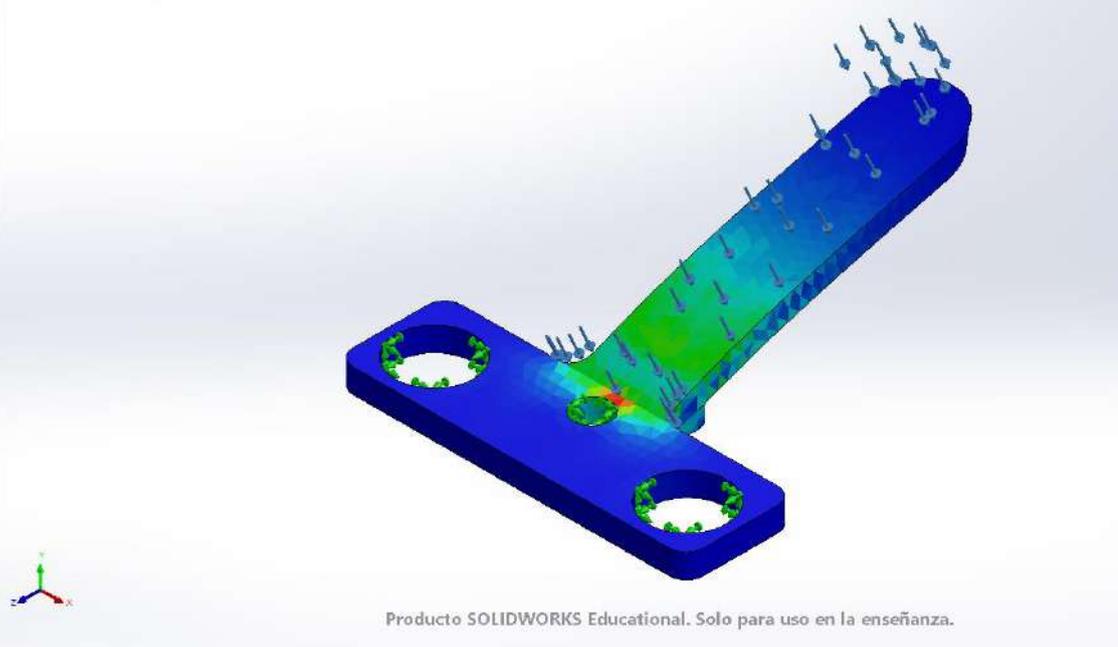
Realizador por: Juiña, P., 2021.

De acuerdo con el análisis el accesorio va a lograr su mayor deformación en la parte superior del punto donde se aplica la fuerza, su desplazamiento máximo será de 9,746e-01mm, si bien este valor representa un desplazamiento, este puede ser despreciado considerando los esfuerzos a los que será sometida la pieza.

Tabla 16-4: Análisis de deformaciones unitarias

Nombre	Tipo	Mín.	Máx.
Deformaciones unitarias1	ESTRN: Deformación unitaria equivalente	2,340e-09 Elemento: 3270	1,310e-03 Elemento: 5024

Nombre del modelo: PisaPedal
 Nombre de estudio: Análisis estático 1 (-Predeterminado-)
 Tipo de resultado: Deformación unitaria estática Deformaciones unitarias1
 Escala de deformación: 14,0189



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

PisaPedal-Análisis estático 1-Deformaciones unitarias-Deformaciones unitarias1

Realizador por: Juiña, P., 2021.

El estudio resulta en una pieza en su mayoría azul, indicando que el accesorio no corre el riesgo de deformarse, se puede asegurar esto al ver el valor mínimo de 2,340e-09 y el valor máximo de 1,310e-03, estos son los valores del coeficiente entre la deformación total y la longitud inicial de la pieza.

4.3. Interfaz gráfica

En la siguiente ilustración se muestra la interfaz gráfica con la que calibra los motores, en esta interfaz el usuario establecerá el máximo recorrido que cada motor podrá hacer, además se podrá establecer una velocidad distinta para cada pedal, la interfaz es amistosa puesto que la simplicidad de su diseño hace que escoger el desplazamiento y la velocidad sea muy fácil.



Ilustración 4-4: Interfaz gráfica para configurar el prototipo

Fuente: Juiña, P., 2021.

4.4. Pruebas

La interfaz permite la configuración de distintos desplazamientos para cada motor, en una escala de 0% a 100% con saltos de 5%, con lo que conseguiremos mayor precisión en la distancia recorrida. El máximo recorrido permitido que se programó en el Arduino y que realizan los motores, previo a la configuración en la interfaz gráfica es de 8 cm para el pedal acelerador, 8 cm para el pedal de freno y 12 cm para el pedal de embrague.

4.4.1. Distancias y velocidades

En las siguientes tablas se puede observar a la distancia de recorrido que corresponde cada porcentaje que se puede seleccionar mediante la interfaz gráfica, dentro de la programación se estableció un recorrido máximo que puede ser modificado.

Tabla 17-4: Distancia lineal de recorrido del embrague

Porcentaje de recorrido lineal (%)	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Recorrido (cm)	1	1	2	2	3	4	4	5	5	6	7	7	8	8	9	10	10	11	11	12

Realizado por: Juiña, P., 2021.

La distancia de recorrido del embrague es mayor puesto que en la realidad al presionar este pedal se lo debe hacer en su totalidad porque de no ser así puede generar desgastes indebidos.

Tabla 18-4: Distancia lineal del recorrido del acelerador y el freno

Porcentaje de recorrido lineal (%)	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Recorrido (cm)	0	1	1	2	2	2	3	3	4	4	4	5	5	6	6	6	6,8	7,2	7,6	8

Realizado por: Juiña, P., 2021.

La velocidad se selecciona en una escala de 0 a 100 con intervalos de 10%, se lo realizó de esta manera puesto que al usar intervalos más pequeños la variación en la velocidad no sería apreciable. Los valores de las velocidades y tiempos son obtenidos con la siguiente configuración.

4.4.2. Embrague

Distancia = 70% (8,4cm)

Tabla 19-4: Tiempos y velocidades del embrague

Velocidad (%)	Tiempo (s)	Velocidad (cm/s)
25	5,9	1,4
50	4,9	1,7
75	4,2	2,0
100	3,5	2,4

Realizado por: Juiña, P., 2021.

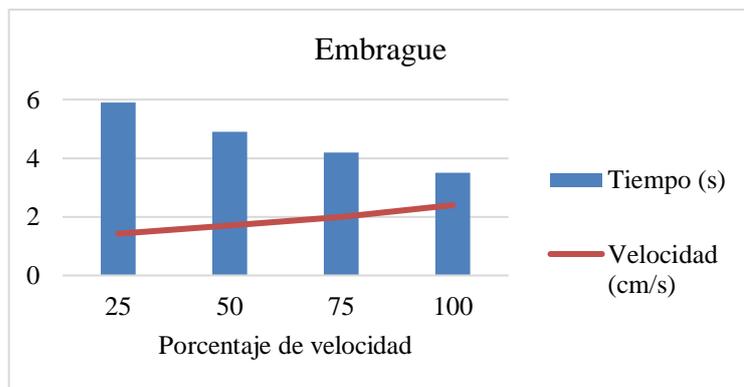


Ilustración 5-4: Gráfico de comparación de velocidad y tiempo

Realizado por: Juiña, P., 2021.

4.4.3. Freno

Distancia = 75% (6,0 cm)

Tabla 20-4: Tiempos y velocidades del freno

Velocidad (%)	Tiempo (s)	Velocidad (cm/s)
25	4,8	1,3
50	4	1,5
75	3,3	1,8
100	2,5	2,4

Realizado por: Juiña, P., 2021.

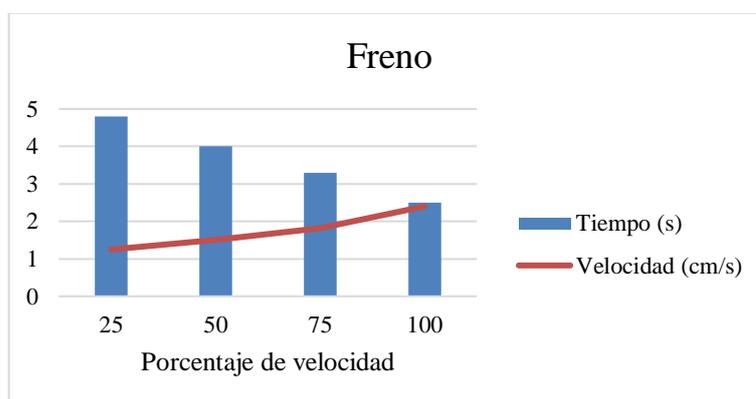


Ilustración 6-4: Gráfico de comparación de velocidad y tiempo

Realizado por: Juiña, P., 2021.

4.4.4. Acelerador

Distancia = 75% (6,0 cm)

Tabla 21-4: Tiempos y velocidades del acelerador

Velocidad (%)	Tiempo (s)	Velocidad (cm/s)
25	4,8	1,3
50	3,9	1,5
75	3,2	1,9
100	2,5	2,4

Realizado por: Juiña, P., 2021.

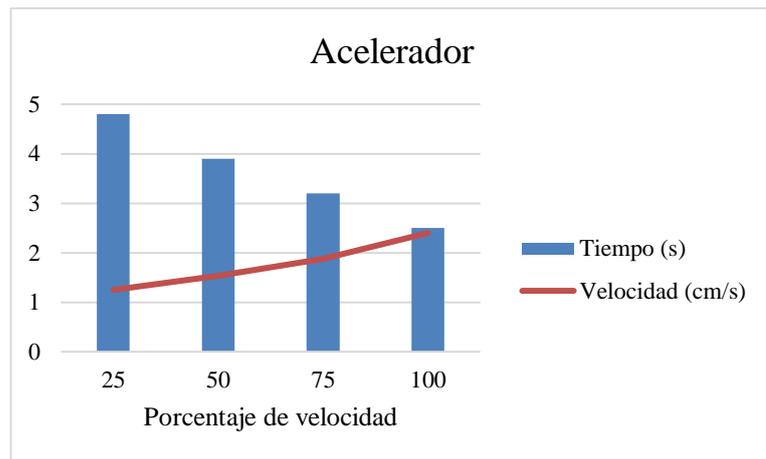


Ilustración 7-4: Gráfico de comparación de velocidad y tiempo

Realizado por: Juiña, P., 2021.

Como se muestra en las ilustraciones a medida que se incrementa la velocidad, el tiempo del recorrido del pedal disminuye. Se utilizó un recorrido similar en los pedales de freno y acelerador porque en un automóvil real, estos poseen características similares, no así el pedal del embrague que es el que más recorrido tiene y debe ser presionado en su totalidad.

4.4.5. Consumo energético

Los siguientes valores se obtuvieron a partir de mediciones realizadas a los drivers de los motores, de acuerdo con las siguientes configuraciones:

4.4.6. Embrague

Distancia = 70% (8,4cm)

Tabla 22-4: Consumo del motor para el embrague en stand by, avance y retroceso.

Velocidad %	Stand by	Avance	Retroceso
25	0,12	0,49	0,47
50	0,12	0,49	0,47
75	0,12	0,5	0,47
100	0,13	0,5	0,47

Realizado por: Juiña, P., 2021.

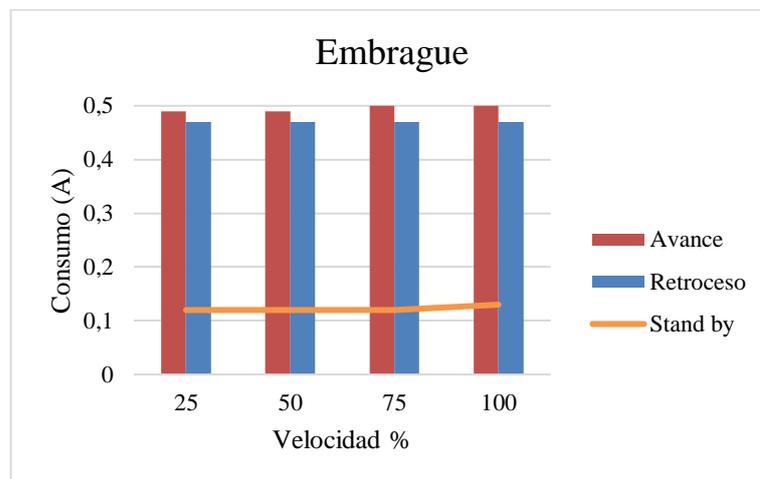


Ilustración 8-4: Consumo del motor para el embrague

Realizado por: Juiña, P., 2021.

4.4.7. Freno

Distancia = 75% (6,0 cm)

Tabla 23-4: Consumo del motor para el freno en stand by, avance y retroceso

Velocidad %	Stand by	Avance	Retroceso
25	0,6	0,64	0,64
50	0,6	0,61	0,61
75	0,59	0,5	0,49
100	0,59	0,4	0,38

Realizado por: Juiña, P., 2021.

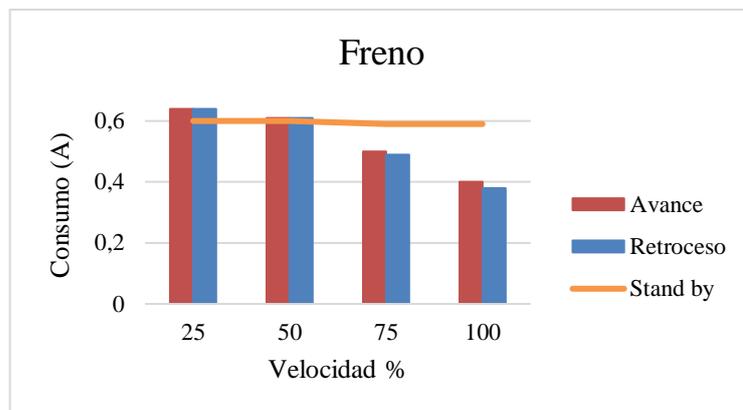


Ilustración 9-4: Consumo del motor para el freno

Realizado por: Juiña, P., 2021.

4.4.8. Acelerador

Distancia = 70% (6,0 cm)

Tabla 24-4: Consumo del motor para el acelerador en stand by, avance y retroceso

Velocidad %	Stand by	Avance	Retroceso
25	0,55	0,61	0,6
50	0,55	0,58	0,58
75	0,57	0,48	0,47
100	0,57	0,38	0,38

Realizado por: Juiña, P., 2021.

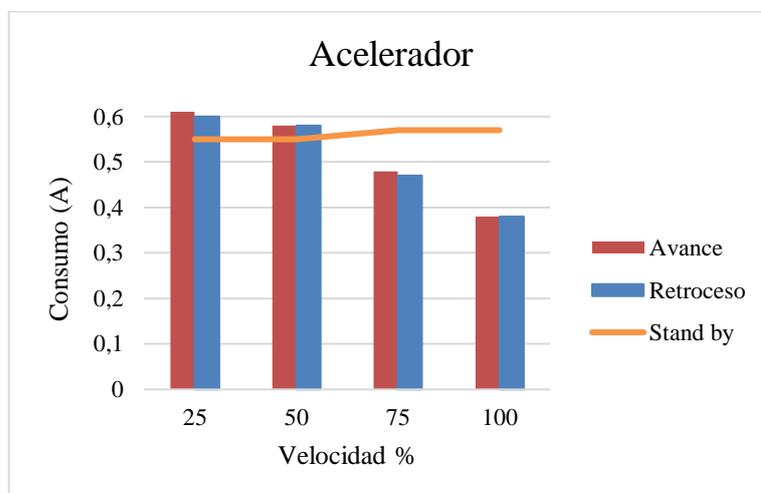


Ilustración 10-4: Consumo del motor para el acelerador

Realizado por: Juiña, P., 2021.

Como se muestra en las tablas anteriores, los motores de los pedales de freno y aceleración tienen consumos muy parecidos, esto se debe a la configuración establecida y al tipo de driver que usan ya que este se alimenta de una fuente de 12V, en cambio el motor para el embrague utiliza un driver que se alimenta de una fuente de 24V, es por esto que se nota una disminución en el consumo del freno y el acelerador al aumentar la velocidad ya que se sacrifica ligeramente el torque para aumentar la velocidad de estos, lo que no ocurre con el consumo del embrague en el que se nota un ligero aumento puesto que mantiene el torque y aumenta la velocidad.

4.5. Instrucciones de uso - manual de usuario

Antes de usar el prototipo, es necesario leer las siguientes instrucciones de uso.

Para la parte mecánica:

- El prototipo debe reposar sobre un plano para asegurar estabilidad, no deben estar objetos en su base que puedan impedir su instalación.
- La base de los motores debe estar libre de cualquier obstáculo en su recorrido porque podría dañar los ejes o chumaceras.
- Asegurarse de que los tornillos sin fin y sus respectivas tuercas están limpios y libres de cualquier tipo de suciedad u óxido.
- Asegurarse de que los pedales puedan hacer su recorrido normal, verificando su estado y limpieza.
- Tener cuidado de la manipulación durante el recorrido de las tuercas, puesto que pueden ocurrir accidentes de atrapamiento y esto podría ocasionar lesiones.
- Ubicar los pies detrás de los motores para evitar cualquier tipo de atrapamiento.
- Para la parte electrónica:
- No utilizarlos en ambientes con demasiada humedad porque podría ocasionar óxido en los terminales y provocar el mal funcionamiento del sistema.
- Verificar las conexiones de las fuentes conmutadas, tanto para red eléctrica doméstica como las conexiones que se dirigen al sistema de control.
- Evitar que lo manipulen personas que no están capacitadas porque podrían provocar corto circuitos.
- Reiniciar los microcontroladores en caso de que el prototipo no quiera encender..
- Manipular con cuidado los pulsadores ubicados en los guantes..
- Tener especial cuidado con el cableado de la caja de control, una tensión indebida podría provocar que uno de los cables salga de su posición.
- No manipular ningún dispositivo de la caja de control con las manos mojadas, esto podría provocar corto circuito y accidentes eléctricos, recordar que el prototipo utiliza la red eléctrica doméstica para funcionar.

4.6. Gestión del proyecto

Para el desarrollo del trabajo se consideraron cuatro puntos esenciales para cumplir con los objetivos trazados.

4.6.1. Cronograma

El proyecto tuvo una duración de cuatro meses, tiempo que fue distribuido por actividades, las cuales se detallan en la tabla.

Tabla 25-4: Cronograma de actividades

Actividades	Meses															
	Mes 1				Mes 2				Mes 3				Mes 4			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Análisis de modificaciones de los automóviles adaptados para personas con discapacidad y sistemas cockpit	■	■														
Análisis de elementos de control electromecánico			■	■												
Diseño del prototipo con software CAD.					■											
Construcción de la estructura cockpit.						■	■	■	■							
Construcción del sistema de control electromecánico.										■	■					
Programación del sistema de control.													■	■		
Adaptación de la estructura cockpit con el sistema de control.															■	
Pruebas y análisis de resultados.																■
Elaboración del trabajo escrito	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Realizado por: Juiña, P., 2021.

4.6.2. Costos

En la tabla se detallan los costos directos e indirectos realizados para la vinculación de un sistema automatizado de control a una máquina elíptica.

Tabla 26-4: Costos

N°	Detalle	Cantidad	Valor Unitario (\$)	Total (\$)
1	Tubo metálico cuadrado 30x30 mm	3	15,00	45,00
2	Asiento de conductor de automóvil	1	110,00	110,00
3	Sistema de pedales	1	80,00	80,00
4	Volante	1	80,00	80,00
5	Palanca de cambio	1	20,00	20,00
6	Placa Arduino	3	25,00	75,00
7	Raspberry pi 3 Model b	1	85,00	85,00
8	Motores Nema 3Nm	3	75,00	225,00
9	Driver para motor nema 23 a 5A	3	27,00	81,00
10	Tornillo sin fin de 30 cm	3	17,50	52,50
11	Eje de acero de 40 cm	6	5,00	30,00
12	Acople flexible 8 a 8mm	3	4,00	12,00
13	Chumacera piso 8mm	6	3,00	18,00
14	Soporte vertical 8mm	12	4,50	54,00
15	Rodamiento 10mm	6	2,00	12,00
16	Fuente conmutada 12 V – 10A	1	15,00	15,00
17	Fuente Conmutada de 24V – 10A	1	38,00	38,00
18	Cable flexible 18 AWG	20	0,27	5,40
19	Cable gemelo 16 AWG	2	0,63	1,26
20	Cable UTP cat5 8 hilos	10	0,47	4,70
21	Caja de madera 70x35x30 cm	1	20,00	20,00
22	Broca plana 3/8	1	1,50	1,50
33	Mano de obra	1	200,00	200,00
			Total (USD)	1 265,36

Realizado por: Juiña, P., 2021.

4.6.3. Recurso humano

El encargado de que los objetivos planteados se logre es el autor del informe Paul Juiña, quien estuvo a cargo del diseño, construcción y programación del prototipo.

4.6.4. Recurso material

Todo lo comprendido a materiales está detallado en las secciones dos y tres de este trabajo, el software y el hardware son *open-source*, permitiendo que este tema sea desarrollado con fines educativos y sociales, la adquisición de los materiales no conlleva un pago adicional por uso de licencias, todos los costos de los materiales fueron asumidos por el autor.

CONCLUSIONES

Mediante la investigación realizada se pudo tener una idea clara de lo que son las adaptaciones de automóviles para personas con capacidades especiales, su principio de funcionamiento y las directrices que se deben seguir para que una persona con estas condiciones pueda utilizarla sin que se correr ningún riesgo.

Con la herramienta *RULER* de Ergonautas, se realizó una medición de los ángulos posturales de tres personas de diferente estatura, para la primera de 165 cm se obtuvieron los siguientes ángulos $A_1=18^\circ$, $A_2=94^\circ$, $A_3=80^\circ$, $A_4=91^\circ$, $A_5=153^\circ$; para la persona de 170 cm se obtuvieron los siguientes ángulos $A_1=14^\circ$, $A_2=93^\circ$, $A_3=79^\circ$, $A_4=80^\circ$, $A_5=149^\circ$; para la persona de 175 cm se obtuvieron los siguientes ángulos $A_1=16^\circ$, $A_2=88^\circ$, $A_3=73^\circ$, $A_4=79^\circ$, $A_5=140^\circ$; que se compararon con ángulos de confort para un puesto de conducción que tienen los siguientes rangos $10^\circ < A_1 < 20^\circ$; $90^\circ < A_2 < 110^\circ$; $80^\circ < A_3 < 120^\circ$; $90^\circ < A_4 < 110^\circ$; $80^\circ < A_5 < 160^\circ$; con estas observaciones la estructura diseñada puede generar ciertas posiciones incómodas para personas con una estatura de 170 cm, para personas con una estatura superior a 175 cm la estructura ya no sería funcional puesto que generaría más tensiones e incomodidad, además de impedir la correcta maniobrabilidad del entorno de experimentación.

Con el software de diseño SolidWorks y su herramienta Simulation, se realizó un estudio de análisis estático en la estructura, aplicando una fuerza de 925 N se obtuvieron como resultados: un límite elástico de $3,5e+08$ N/m², un desplazamiento máximo de $2,862e-03$ mm. Lo que nos indica que la estructura construida soporta los esfuerzos a los que fue sometida.

Con el software de diseño SolidWorks y su herramienta Simulation, se realizó un estudio de análisis estático en el accesorio de los pedales, aplicando una fuerza de 10 N se obtuvieron como resultados: un límite elástico de $3,5e+08$ N/m², un desplazamiento máximo de $9,746e-01$ mm y una deformación unitaria de $1,310e-03$. Lo que nos indica que la pieza diseñada soporta los esfuerzos a los que fue sometida.

Definidos los requerimientos de hardware y software del prototipo se seleccionaron aquellos que son *open-source* por su costo accesible y al ser libre de licencias es posible utilizarlos sin el riesgo demandas por copyright, tanto las tecnologías Arduino como Raspberry están libres de licencias.

El consumo energético del control electromecánico en las máximas configuraciones posibles se dio de la siguiente manera, el embrague con un consumo de 0,5 A; el freno con 0,4 A; y el

acelerador con un consumo de 0,34 A, esto permite que el dispositivo se pueda conectar al sistema eléctrico de un automóvil por su bajo consumo.

RECOMENDACIONES

Para futuros proyectos de desarrollo se deben considerar el uso de tecnologías *open-source* ya que al estar libre de licencias se evita demandas por copyright.

En lo que respecta al uso de la máquina hay que considerar que al momento de apagarla todos los pedales estén su posición inicial debido a que el software solo reconoce como punto inicial aquel que está configurado al momento de encender el sistema.

Para personas con una estatura superior a 170 cm es necesario realizar otro análisis y diseño de cockpit para facilitar maniobrabilidad y evitar posturas que generen tensiones e incomodidad.

Se podría obviar el uso de protoboard y reemplazarla por una placa de circuito integrado que mejor el uso del espacio en la caja de control.

Leer antes las indicaciones para el uso del prototipo en el que se detallan de mejor manera ciertos puntos a considerar para configurarlo de mejor manera.

Para utilizar el *software* de programación *Qt Creator* en la raspberry, verificar que es compatible con las actualizaciones del sistema operativo de esta, porque puede generar conflictos al momento de configurar el puerto serial que sirve de comunicación entre la raspberry y el Arduino.

BIBLIOGRAFÍA

AGUDELO, Néstor; TANO, Giovanni & VARGAS, Carlos. “Historia de la automatización”. Universidad ECCI [en línea]. 2020. [Consulta: 25 mayo 2022]. Disponible en: <http://ingenierovizcaino.com/ecci/aut1/corte1/articulos/Historia%20de%20la%20Automatizacion.pdf>

ÁGUEDA CASADO, Eduardo; GARCÍA JIMÉNEZ, José; GÓMEZ MORALES, Tomás & MARTÍN NAVARRO, José. *Estructuras del vehículo* [en línea]. España:Ediciones Paraninfo, 2016. [Consulta: 12 enero 2022]. ISBN 978-84-283-3555-3. Disponible en: https://books.google.es/books?id=Rh6rCwAAQBAJ&dq=tipos+de+carrocerias&lr=&hl=es&source=gbs_navlinks_s

ALONSO PÉREZ, José Manuel. *Sistemas de transmisión y frenado*. España: Ediciones Paraninfo, 2014. ISBN 978-84-283-9529-8.

AUTODESK. *Software de diseño CAD* [blog]. Autodesk, 2022. [Consulta: 25 junio 2022]. Disponible en: <https://latinoamerica.autodesk.com/solutions/cad-software>

CAMARENA LILLO, Francisco. Diseño del embrague de un vehículo (Proyecto de fin de carrera) (Pregrado) [en línea]. Escuela Politécnica Superior Universidad Carlos III de Madrid, Departamento de Ingeniería Mecánica, 2018. [Consulta: 25 marzo 2022]. Disponible en: https://e-archivo.uc3m.es/bitstream/handle/10016/13717/PFC_Francisco_Camarena_Lillo.pdf?sequence=1&isAllowed=y

CONSEJO NACIONAL PARA LA IGUALDAD DE DISCAPACIDADES. *Total de personas con discapacidad registradas en el registro nacional de discapacidad* [en línea]. Quito – Ecuador, 2022. [Consulta: 21 febrero 2022]. Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>

CONTI, Francisco. *Motores paso a paso* [en línea]. Buenos Aires – Argentina: Tencibook Ediciones, 2005. [Consulta: 30 mayo 2022]. ISBN 9789505531943. Disponible en: https://books.google.es/books?id=6yiXDwAAQBAJ&dq=motor+paso+a+paso&lr=&hl=es&source=gbs_navlinks_s

DOLS RUIZ, Juan. “Adaptaciones y ayudas técnicas en los vehículos, sistema de clasificación de adaptaciones”. Universidad Politécnica de Valencia [en línea]. [Consulta: 12 febrero 2022]. Disponible en: https://www.researchgate.net/profile/Juan-Dols/publication/301953123_ADAPTACIONES_Y_AYUDAS_TECNICAS_EN_LOS_VEHICULOS_SISTEMA_DE_CLASIFICACION_DE_ADAPTACIONES/links/572ca6f608ae7a42e13346fe/ADAPTACIONES-Y-AYUDAS-TECNICAS-EN-LOS-VEHICULOS-SISTEMA-DE-CLASIFICACION-DE-ADAPTACIONES.pdf

FERNÁNDEZ ARAGÓN, Iñigo. Control de un motor paso a paso: PIC, USB, C# (Proyecto fin de carrera) (Pregrado) [en línea]. Universidad Pública de Navarra, 2011. [Consulta: 10 mayo 2022]. Disponible en: <https://academica-e.unavarra.es/xmlui/handle/2454/3547>

GONZÁLEZ DOMÍNGUEZ, Claudio. Aplicaciones orientadas a la domótica con Raspberry Pi (Trabajo de fin de grado) (Pregrado) [en línea]. Universidad de Sevilla, Escuela Técnica Superior de Ingeniería, Departamento de Ingeniería Electrónica. Sevilla – España, 2015. [Consulta: 14 mayo 2021]. Disponible en: <https://biblus.us.es/bibing/proyectos/abreproy/90330/fichero/Memoria+TFG.pdf>

GUERRERO NEJER, Ronald Alexis; CAIZA QUISHPE, Luis & PINTO AYALA, Silvia. “Evolución de vehículos adaptados con controles manuales para discapacitados”. Polo del conocimiento [en línea]. 2022, 7 (9), pp. 1591-1608. [Consulta: 15 marzo 2022]. Disponible en: <https://polodelconocimiento.com/ojs/index.php/es/article/view/4651/11167>

HERRERA, Rociño & PUEBLAS, Walter. *Manual de conducción vehicular práctica: AUTOMÓVIL – Nivel básico* [en línea]. Buenos Aires – Argentina, 2021. [Consulta: 12 mayo 2021]. Disponible en: <https://autoescuelacity.com.ar/wp-content/uploads/2022/10/Manual-de-conduccion-vehicular-automovil.pdf>

JENNINGS, Steve. “Motores paso a paso”. Informador Técnico [en línea]. 2002, 65, pp. 47-58. [Consulta: 21 mayo 2022]. Disponible en: https://revistas.sena.edu.co/index.php/inf_tec/article/view/899/988

JORQUERA ORTEGA, Adam. *Fabricación Digital: Introducción al modelo de impresión 3D* [en línea]. España, Catálogo de publicaciones del Ministerio de Educación, Cultura y Deportes, 2016. [Consulta: 15 marzo 2021]. ISBN 9788436957457. Disponible en: https://books.google.es/books?hl=es&lr=lang_es&id=9XmbDQAAQBAJ&oi=fnd&pg=PA5&d

q=impresion+3d&ots=7e1F_6HYeq&sig=VAawSGWuqF_-
EAo0sbr3203sukI#v=onepage&q&f=false

MUNIOSGUREN GARCÍA, Ignacio. Implementación de control de motor paso a paso con técnicas de microstepping (Trabajo Fin de Grado) (Pregrado) [en línea]. Universidad Pontificia Comillas, 2018. [Consulta: 21 marzo 2022]. Disponible en: <https://repositorio.comillas.edu/xmlui/handle/11531/22754>

MORENO FERGUSSON, María Elisa & AMAYA REY, Pilar. “Paraplejia: pasado y futuro del ser”. *Index de Enfermería* [en línea]. 2009, 18 (3), pp. 185-189. [Consulta: 15 febrero 2022]. Disponible en: https://scielo.isciii.es/scielo.php?script=sci_abstract&pid=S1132-12962009000300010

ORGANIZACIÓN MUNDIAL DE LA SALUD. *Lesiones medulares* [blog]. Organización Mundial de la Salud, 19 noviembre 2013. [Consulta: 12 febrero 2022]. Disponible en: <https://www.who.int/es/news-room/fact-sheets/detail/spinal-cord-injury>

ORGANIZACIÓN MUNDIAL DE LA SALUD. *Proyecto de acción mundial de la OMS sobre discapacidad 2014-2021: Mejor salud para todas las personas con discapacidad* [en línea]. Ginebra: OMS, 2014. [Consulta: 14 febrero 2022]. Disponible en: https://apps.who.int/iris/bitstream/handle/10665/175064/B134_16-sp.pdf?sequence=1&isAllowed=y

ORTIZ GIL, Amelia. “Las impresoras 3D como herramientas científicas”. Observatorio Astronómico de la Universidad de Valencia [en línea]. 2019. [Consulta: 25 mayo 2021]. Disponible en: https://repositorio.uam.es/bitstream/handle/10486/687514/EM_61_6.pdf?sequence=1&isAllowed=y

OTERO, Santiago. *Manual básico acerca del automóvil y su motor* [en línea]. Ibarra – Ecuador, 2017. [Consulta: 12 marzo 2022]. ISBN 978-9942-28-548-5. Disponible en: https://www.researchgate.net/profile/Santiago-Otero-Potosi/publication/324210587_MANUAL_BASICO_ACERCA_DEL_AUTOMOVIL_Y_SU_MOTOR/links/5ac4f47ea6fdcc26f7e1795f/MANUAL-BASICO-ACERCA-DEL-AUTOMOVIL-Y-SU-MOTOR.pdf

PEÑA MILLAHUAL, Claudio. *Descubriendo Arduino* [en línea]. Buenos – Aires: Argentina: Six Ediciones, 2020. [Consulta: 12 mayo 2022]. ISBN 978-987-4958-24-2. Disponible en: https://books.google.es/books?hl=es&lr=&id=bL7PDwAAQBAJ&oi=fnd&pg=PA1&dq=arduino&ots=fH_TSGDafe&sig=mxawqt0Bu2XXy6vmk_Jxkc2Bw-4#v=onepage&q=arduino&f=false

PÉREZ TORRALBA, Fernando. Adaptaciones de vehículos a motor para conductores con procesos invalidantes del sistema músculo-esquelético (Tesis) (Doctorado) [en línea]. Universidad Complutense de Madrid, Facultad de Medicina, Escuela de Medicina del Trabajo, 2001. [Consulta: 12 marzo 2022]. Disponible en: <http://webs.ucm.es/BUCM/tesis//19911996/D/0/D0052001.pdf>

PINILLA ALONSO, Teresa. El automóvil 4.0 (Trabajo de fin de grado) (Pregrado) [en línea]. Universidad de Sevilla, Escuela Técnica Superior de Ingeniería, Departamento de Organización Industrial y Gestión de Empresas II. Sevilla – España, 2019. [Consulta: 25 enero 2022]. Disponible en: <https://idus.us.es/bitstream/handle/11441/94482/TFG-2732-PINILLA%20ALONSO.pdf?sequence=1&isAllowed=y>

RASPBERRY PI. *Raspberry Pi 1 Modelo A+* [blog]. 2021. [Consulta: 14 marzo 2021]. Disponible en: <https://www.raspberrypi.com/products/raspberry-pi-1-model-a-plus/>

REGLAMENTO A LA LEY ORGÁNICA DE DISCAPACIDADES. *Decreto Ejecutivo 194 – Reglamento a la ley orgánica de discapacidades* [en línea]. Quito – Ecuador, 2017. [Consulta: 15 marzo 2022]. Disponible en: https://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2017/11/Reglamento-lod-decre_-194.pdf

REDONDO MELCHOR, Norberto. *Control de motor paso a paso* [en línea]. Universidad de Salamanca, 2016. [Consulta: 25 marzo 2022]. Disponible en: <http://stsproyectos.com/U/M/MotorPasoAPaso.pdf>

REYES CORTÉS, Fernando; MONJARAZ, Jaime & VARGAS SOTO, Emilio. *Mecatrónica, control y automatización* [en línea]. México: Alfaomega Editor S.A., 2013. [Consulta: 20 junio 2022]. Disponible en: <https://books.google.es/books?hl=es&lr=&id=cOt1EAAAQBAJ&oi=fnd&pg=PR2&dq=sistemas+de+control+automatizacion&ots=ebGMYfAroD&sig=k-usbs1bO3->

dQF7LQfMFghAjWCE#v=onpage&q=sistemas%20de%20control%20automatizacion&f=fals
e

REGLAMENTO A LA LEY ORGÁNICA DE DISCAPACIDADES. *Decreto Ejecutivo 194 – Reglamento a la ley orgánica de discapacidades* [en línea]. Quito – Ecuador, 2017. [Consulta: 15 marzo 2022]. Disponible en: https://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2017/11/Reglamento-lod-decre_-194.pdf

RÚA, Edwin; JIMÉNEZ, Fernando; GUTIÉRREZ, Germán & VILLAMIZAR, Nelson. “3D Printing as a Didactic Toll for Teaching come Engineering and Desing Concepts”. *Revista Ingeniería* [en línea]. 2017, 23 (1),pp. 70-73. [Consulta: 15 enero 2021]. Disponible en: <http://www.scielo.org.co/pdf/inge/v23n1/0121-750X-inge-23-01-00070.pdf>

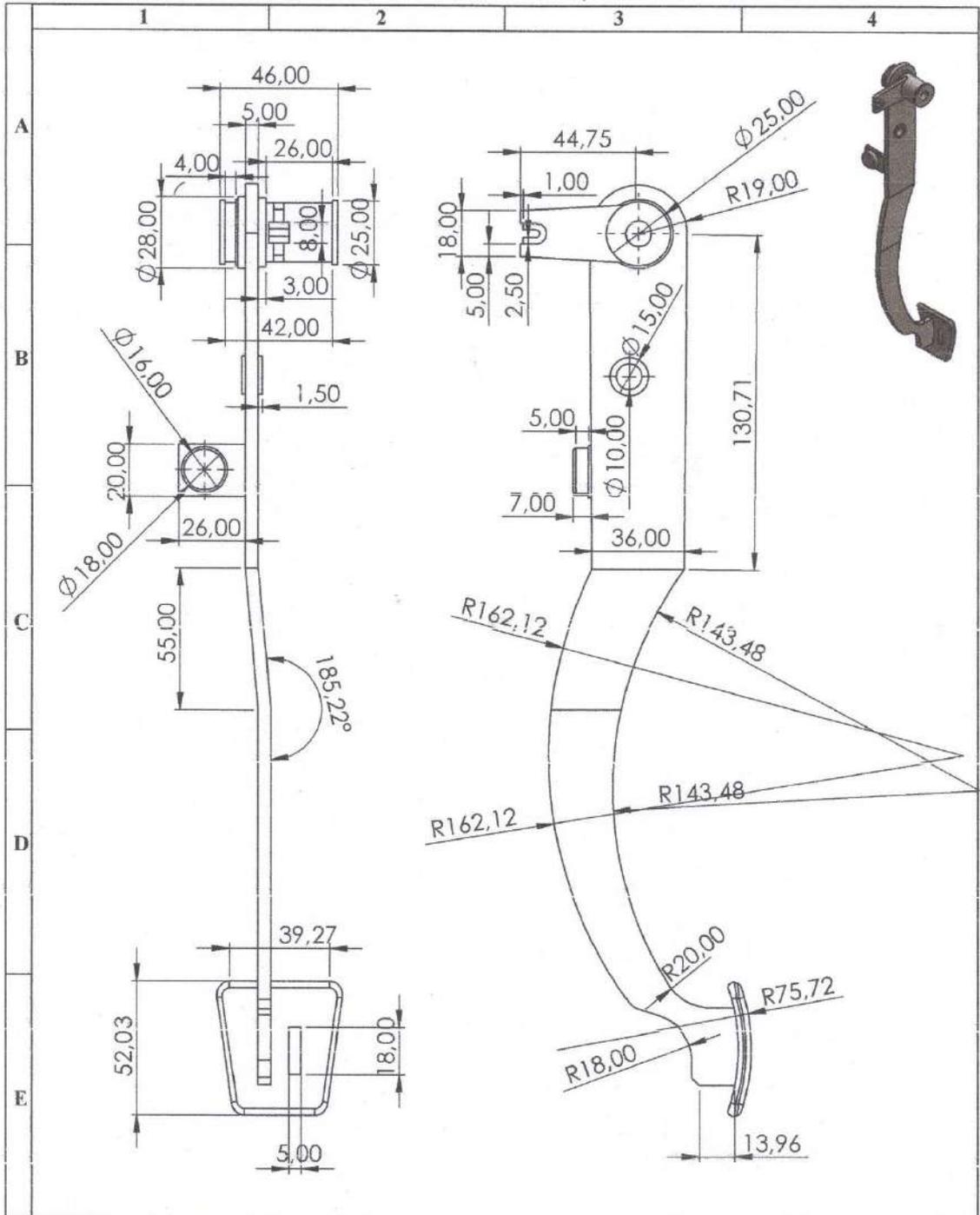
SOLIDBI. SOLINWORKS. *¿Qué es y para qué sirve?* [blog]. 2022. [Consulta: 15 junio 2022]. Disponible en: <https://solid-bi.es/solidworks/>

VIVAR BIMOS, Diego. *Análisis sobre la lesión medular en Ecuador: datos epidemiológicos y planificación sanitaria* [en línea]. Barcelona – España, 2020. [Consulta: 10 febrero 2022]. Disponible en: https://siidon.guttmann.com/files/tfm_diego_vivar.pdf

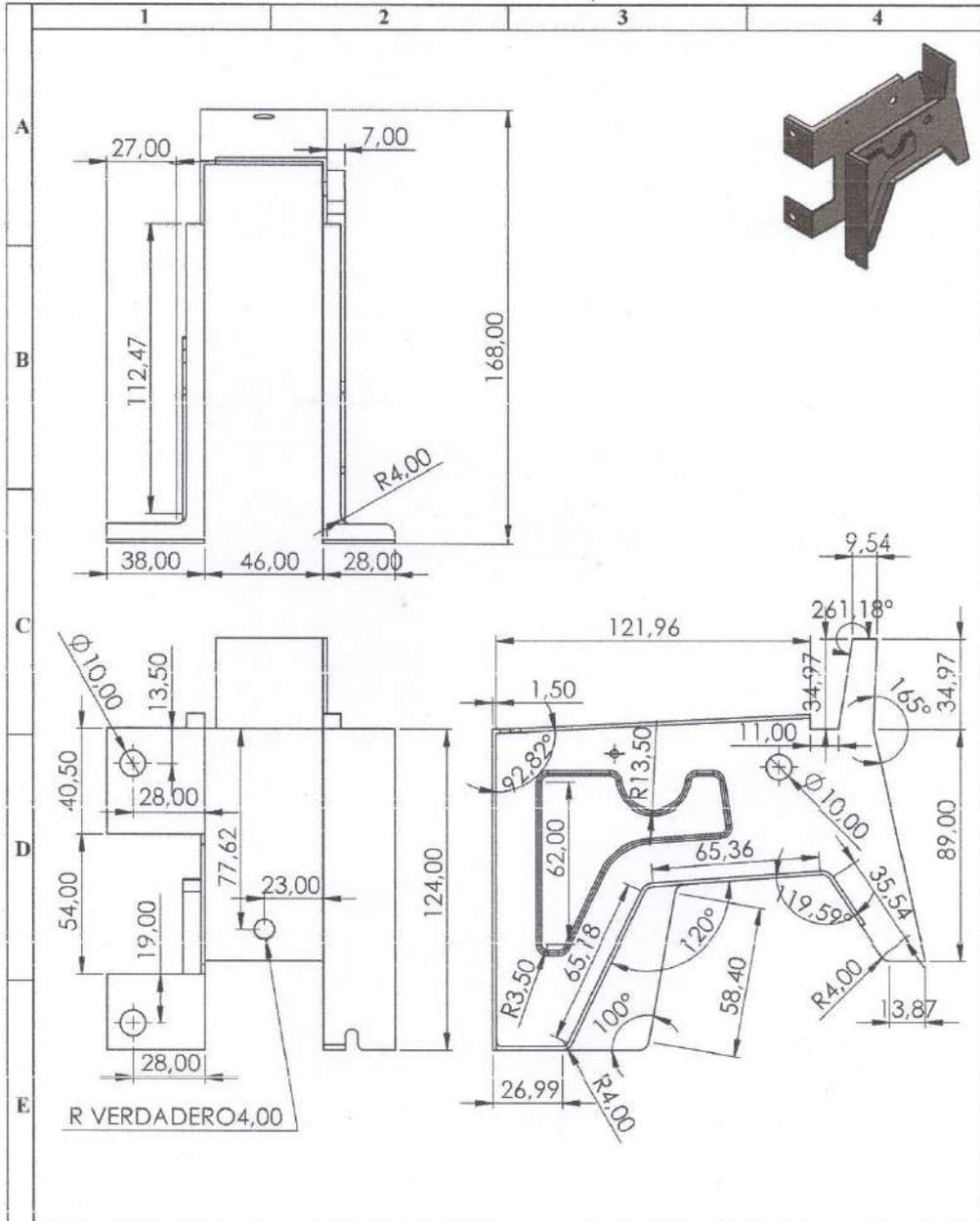
WEATHERSPOON, Deborah & KANDOLA, Aaron. *Todo lo que necesitas saber sobre la paraplejia* [blog]. *Medical News Today*, 06 mayo 2020. [Consulta: 25 febrero 2022]. Disponible en: <https://www.medicalnewstoday.com/articles/what-is-paraplegia>

ANEXOS

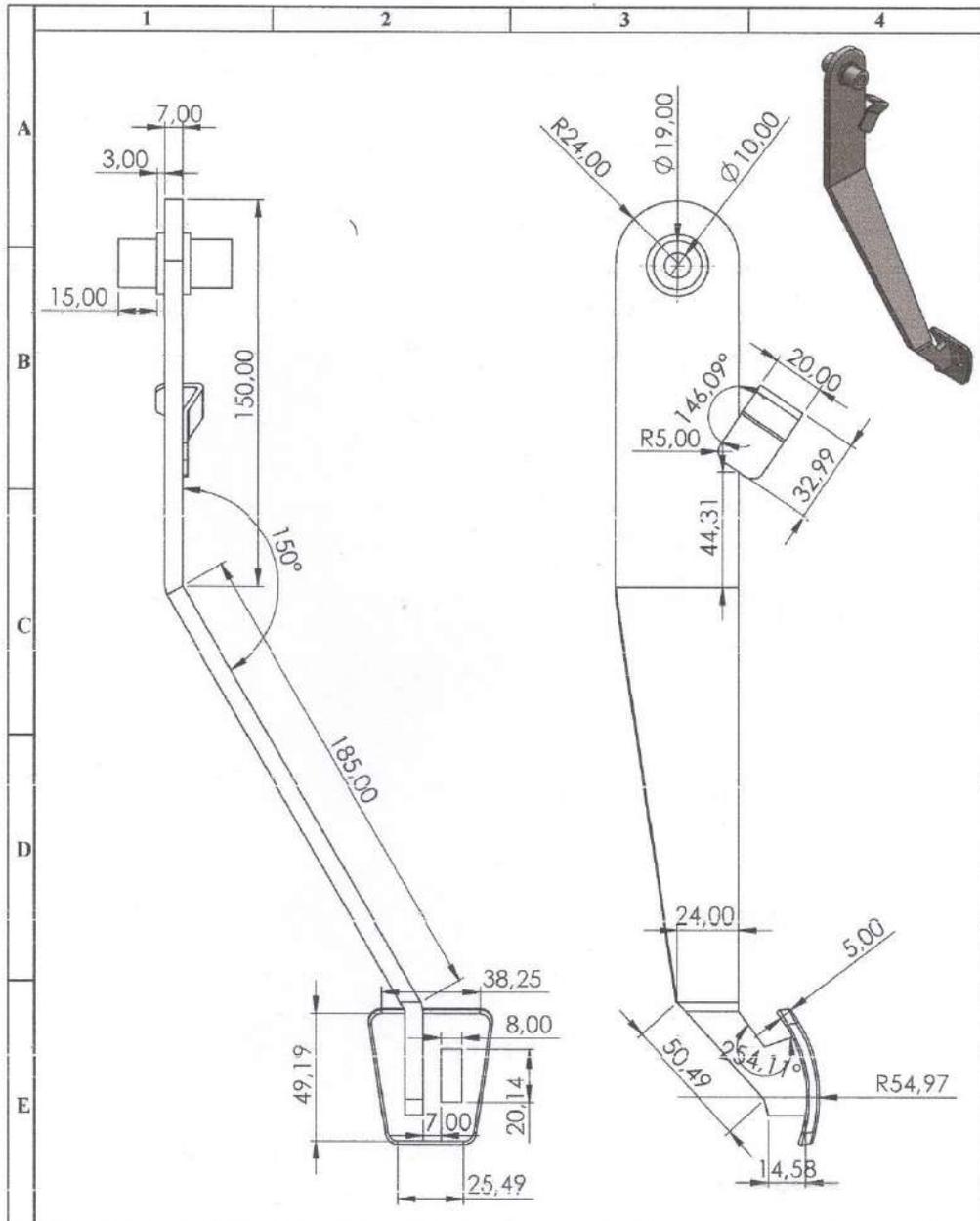
ANEXO A: PLANOS



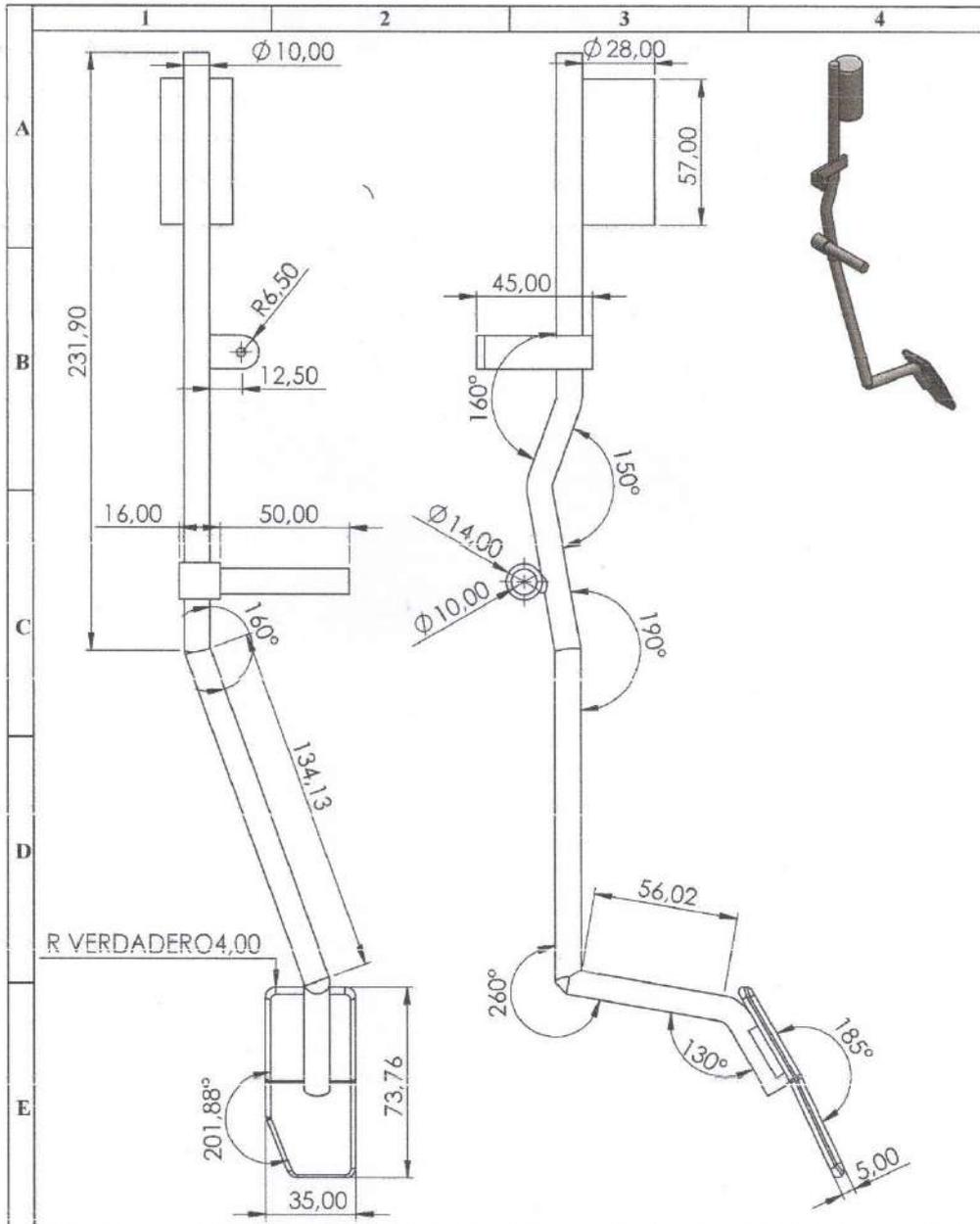
N° Lámina:		N° Hojas: 1 de 9		Denominación:		ESPOCH			
Sustitución:				Pedal de embrague		INGENIERÍA INDUSTRIAL			
Datos	Nombre:	Firma:	Fecha:			Peso (Kg)	Tolerancia	Escala	Registro
Proyectó	Julia P.							1:2	
Dibujó	Julia P.								
Revisó	Ing. Garcia E.								
Aprovó	Ing. Garcia E.			Materiales:		Codificación:			
				Acero		CII-2023			



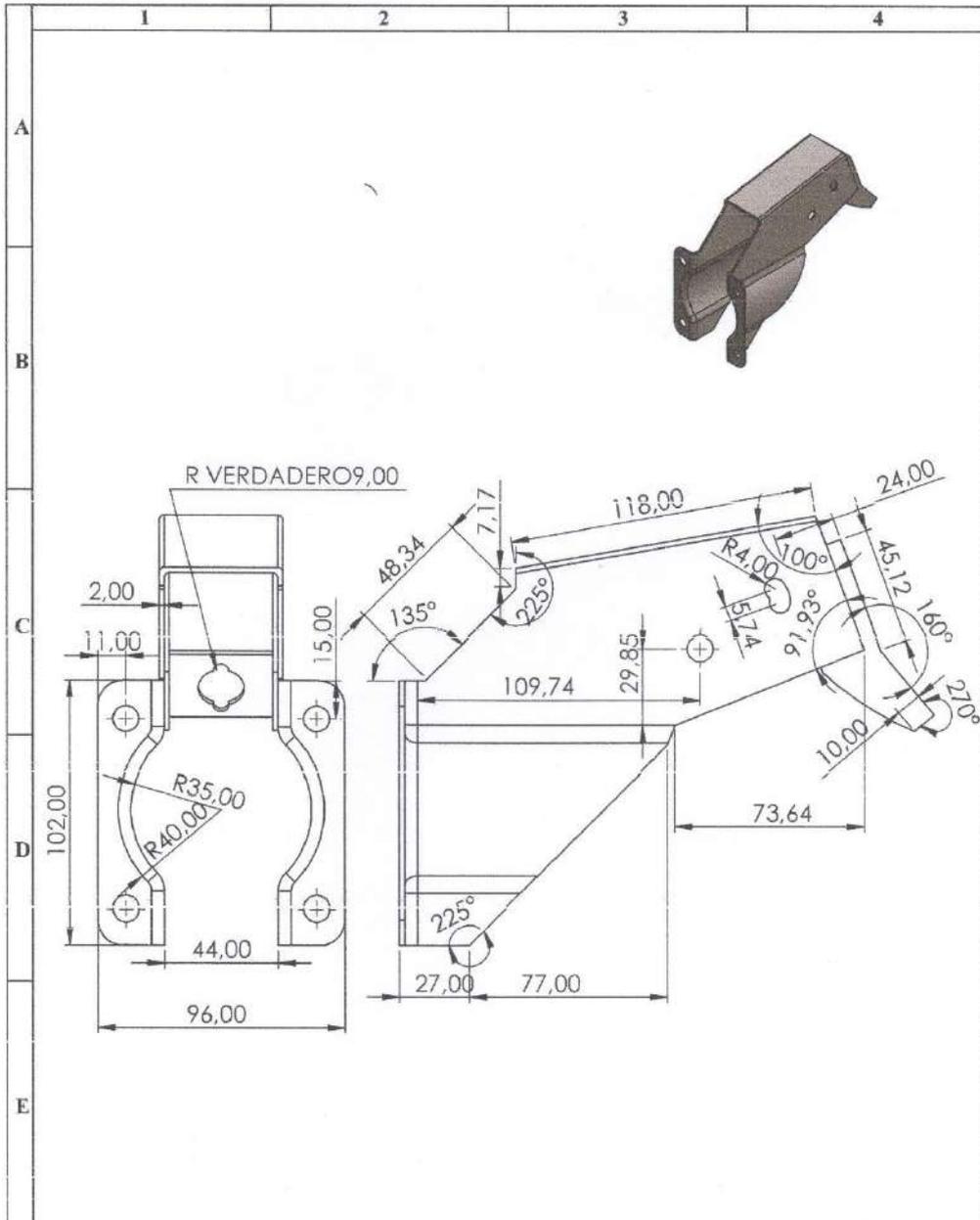
N° Lámina:		N° Hojas:		Denominación:		ESPOCH			
		2 de 9				INGENIERÍA INDUSTRIAL			
Sustitución:						Soporte de embrague			
Datos	Nombre:	Firma:	Fecha:			Peso (Kg)	Tolerancia	Escala	Registro
Proyectó	Juñta P.							1:2	
Dibujó	Juñta P.								
Revisó	Ing. García E.					Codificación:			
Aprobó	Ing. García E.			Acero		CII-2023			



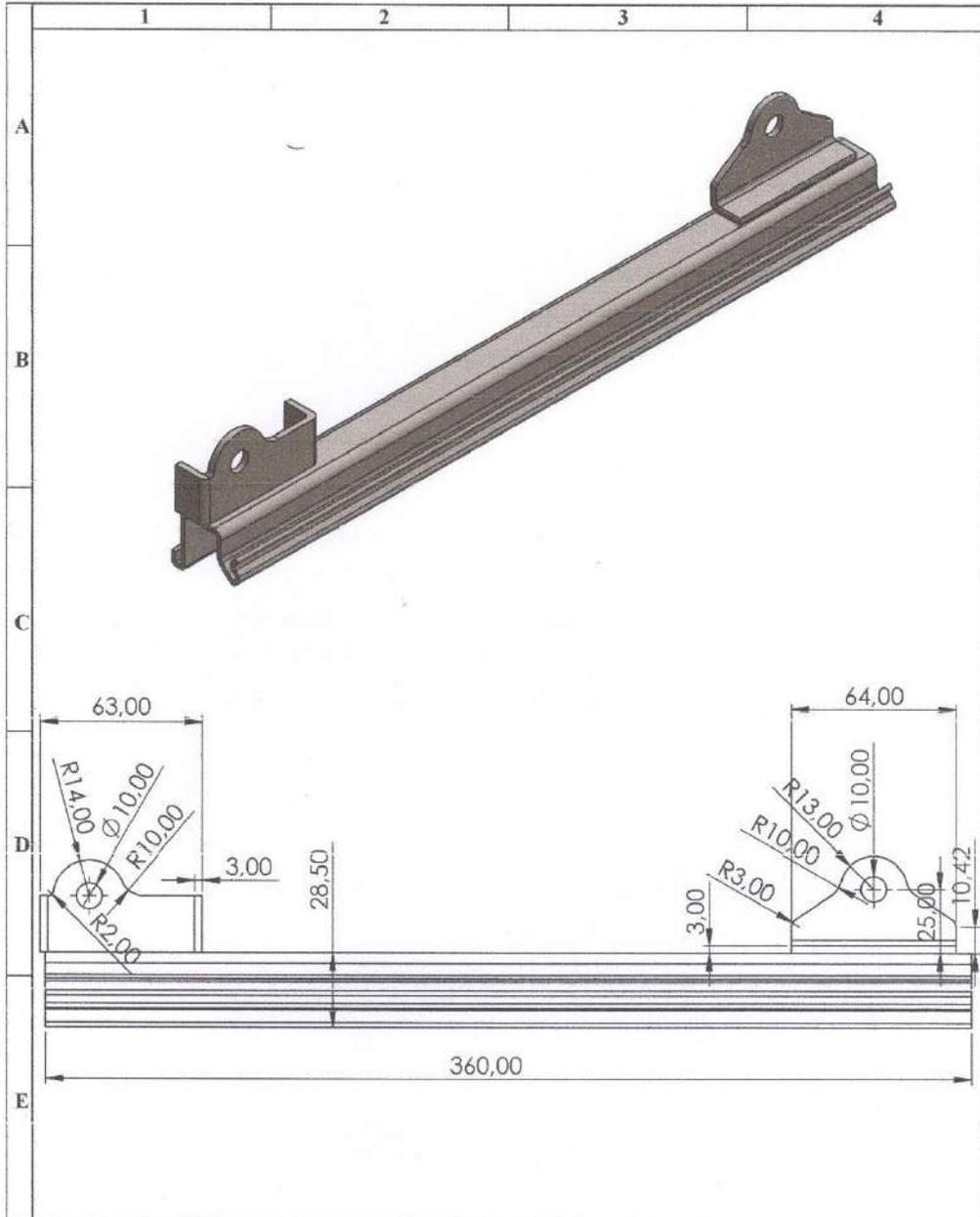
N° Lámina:		N° Hojas:		Denominación:		ESPOCH			
		3 de 9		Pedal de freno		INGENIERÍA INDUSTRIAL			
Sustitución:						Peso (Kg)	Tolerancia	Escala	Registro
Datos	Nombre:	Firma:	Fecha:					1:2	
Proyectó	Juliá P.								
Dibujó	Juliá P.								
Revisó	Ing. García E.			Materiales:		Codificación:			
Aprovó	Ing. García E.			Acero		CH-2023			



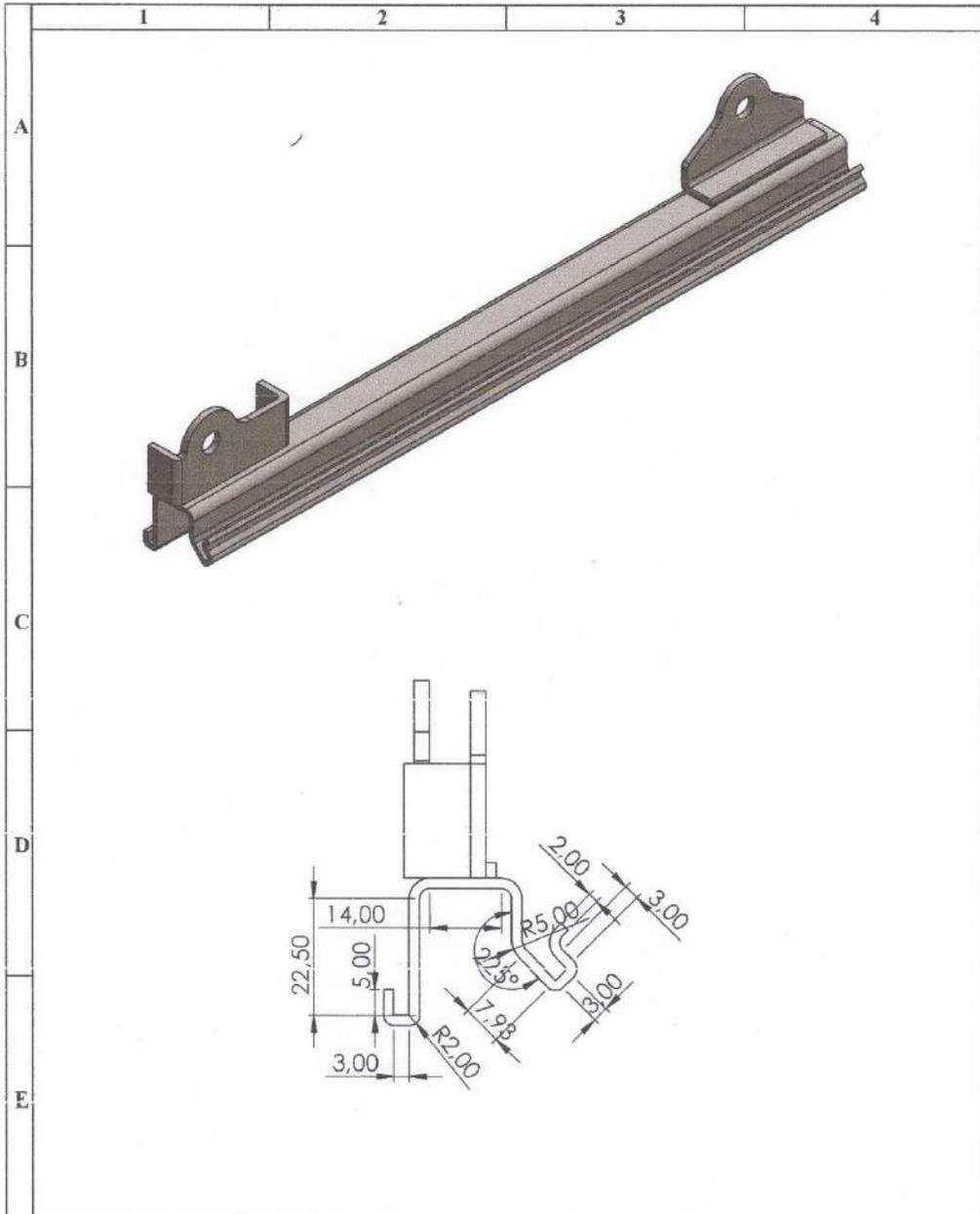
N° Lámina:		N° Hojas:		Denominación:				ESPOCH			
		4 de 9		Pedal acelerador				INGENIERÍA INDUSTRIAL			
Sustitución:								Peso (Kg)		Tolerancia	
Datos	Nombre:	Firma:	Fecha:					1:2		☐ ⊕	
Proyectó	Juina P.			Materiales:				Codificación:			
Dibujó	Juina P.										
Revisó	Ing. García E.										
Aprovó	Ing. García E.			Acero				CII-2023			



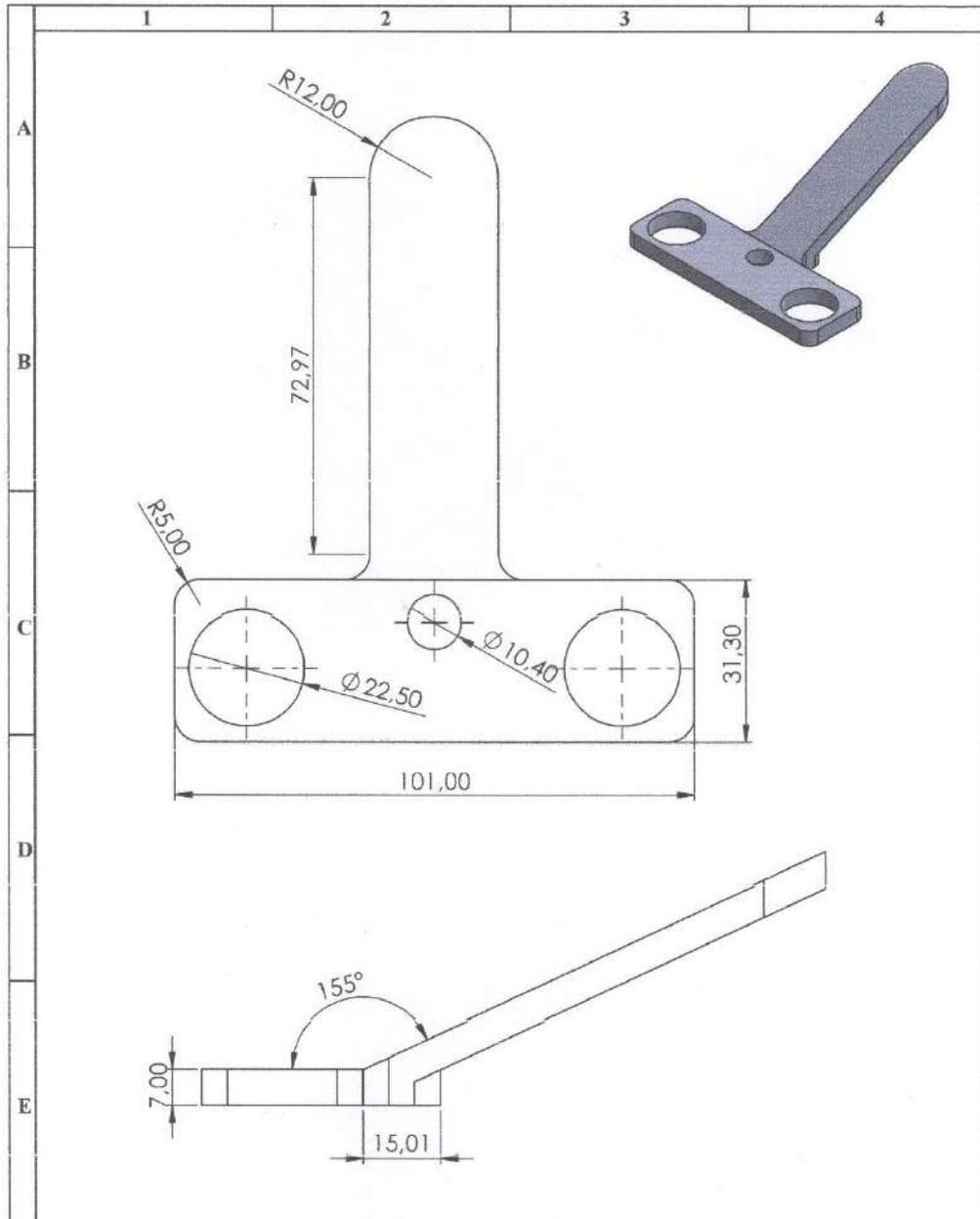
N° Lámina:		N° Hojas:		Denominación:	ESPOCH INGENIERÍA INDUSTRIAL				
		5 de 9							Soporte freno- acelerador
Sustitución:					Peso (Kg)	Tolerancia	Escala	Registro	
Datos	Nombre:	Firma:	Fecha:	Materiales:	Codificación:	1:2			
Proyectó	Juñta P.							Acero	CII-2023
Dibujó	Juñta P.								
Revisó	Ing. García E.								
Aprovó	Ing. García E.								



N° Lámina:		N° Hojas:		Denominación:		ESPOCH					
		6 de 9		Riel del asiento - alzado		INGENIERÍA INDUSTRIAL					
Sustitución:						Peso (Kg)	Tolerancia	Escala	Registro		
Datos	Nombre:	Firma:	Fecha:					1:2			
Proyectó	Juñta P.										
Dibujó	Juñta P.			Materiales:		Codificación:					
Revisó	Ing. García E.			Acero al carbono laminado en frío		CII-2023					
Aprobó	Ing. García E.										

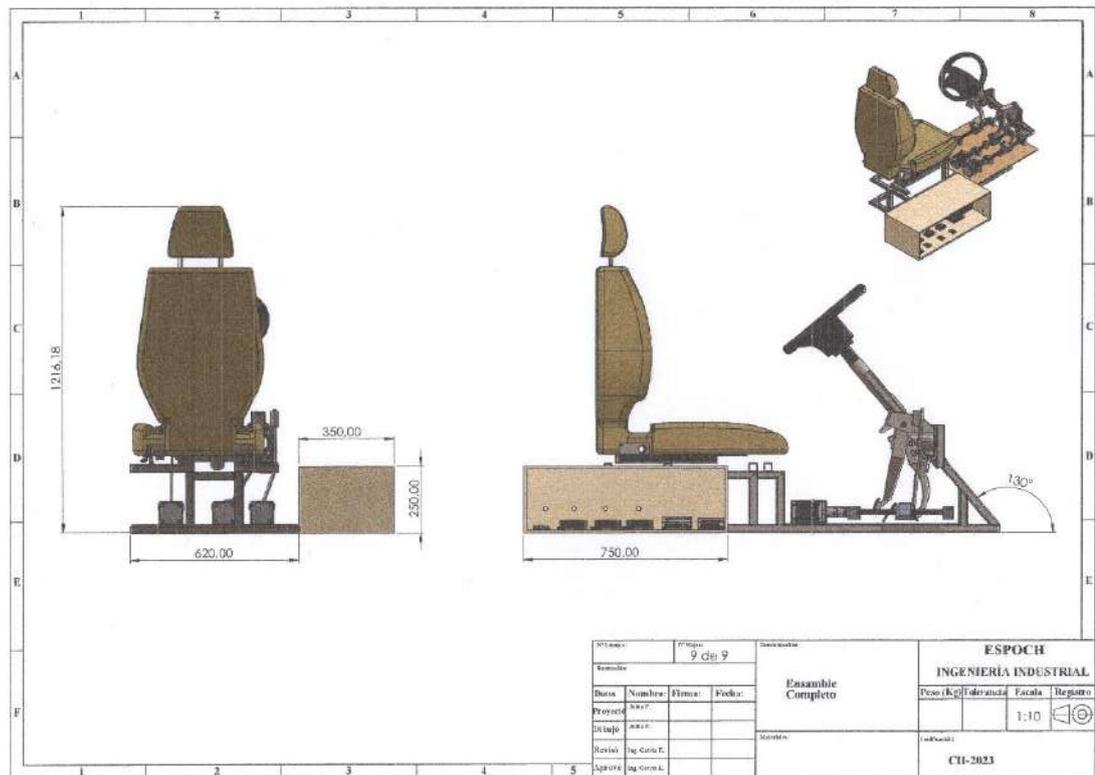


N° Lámina:		N° Hojas: 7 de 9		Denominación:		ESPOCH			
Sustitución:				Riel del asiento - Lateral		INGENIERÍA INDUSTRIAL			
Datos	Nombre:	Firma:	Fecha:			Peso (Kg)	Tolerancia	Escala	Registro
Proyectó	Juñta P.					1:1			
Dibujó	Juñta P.			Acero al carbono laminado en frío		CII-2023			
Revisó	Ing. García E.								
Aprovó	Ing. García E.								



N° Lámina:		N° Hojas:		Denominación:		ESPOCH INGENIERÍA INDUSTRIAL				
		8 de 9		Accesorio						
Sustitución:						Peso (Kg)		Tolerancia	Escala	Registro
Datos	Nombre:	Firma:	Fecha:					1:1		
Proyectó	Juñfa P.									
Dibujó	Juñfa P.					Materiales:		Codificación:		
Revisó	Ing. García E.					ABS		CII-2023		
Aprovó	Ing. García E.									

ANEXO B: ENSAMBLE COMPLETO



ANEXO C: PROGRAMACIÓN DE INTERFAZ GRÁFICA

```
Projects | mainwindow.cpp | #include <QtWidgets>
interface_Conduccion.pro
Headers
mainwindow.h
Sources
main.cpp
mainwindow.cpp
Forms
mainwindow.ui
Resources

1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3 #include <QSerialPortInfo>
4 #include <QMessageBox>
5 #include <QThread>
6 #include <QDebug>
7
8 MainWindow::MainWindow(QWidget *parent) :
9     QMainWindow(parent),
10    ui(new Ui::MainWindow)
11 {
12    ui->setupUi(this);
13    char data;
14
15    port_name = "";
16    arduino = new QSerialPort();
17    arduino2 = new QSerialPort();
18    arduino3 = new QSerialPort();
19    arduino->isAvailable() = true;
20    arduino2->isAvailable() = true;
21    arduino3->isAvailable() = true;
22    serialBuffer = "";
23
24
25
26
27
28
29
30
31 foreach (const QSerialPortInfo &serial_info, QSerialPortInfo::availablePorts()) {
32     qDebug() << "Puerto: " << serial_info.portName();
33 }
34
35 if(arduino->isAvailable()){
36     arduino->setPortName("/dev/ttyACM0");
37     arduino->open(QSerialPort::ReadWrite);
38     arduino->setBaudRate(QSerialPort::Baud9600);
39     arduino->setDataBits(QSerialPort::Data8);
40     arduino->setParity(QSerialPort::NoParity);
41     arduino->setStopBits(QSerialPort::OneStop);
42     arduino->setFlowControl(QSerialPort::NoFlowControl);
43 }else {
44     QMessageBox::warning(this, "Error en el Puerto", "No se puede conectar a arduino");
45 }
46
47 if(arduino2->isAvailable){
48     arduino2->setPortName("/dev/ttyACM0");
49     arduino2->open(QSerialPort::ReadWrite);
50     arduino2->setBaudRate(QSerialPort::Baud9600);
51     arduino2->setDataBits(QSerialPort::Data8);
52     arduino2->setParity(QSerialPort::NoParity);
53     arduino2->setStopBits(QSerialPort::OneStop);
54     arduino2->setFlowControl(QSerialPort::NoFlowControl);
55 }
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```

137 void MainWindow::on_enviar_des_emb_8_clicked()
138 {
139     int x = ui->verticalSlider->value();
140     int y = 5;
141     ui->verticalSlider_2->setValue(8);
142     vel_emb=ui->lineEdit_4->text();
143     //explicit conversion turns floating-point number into integer: 'double' to 'int'
144     if (R > -1 & R < 100){
145         if (vel_emb >= 1 & vel_emb <=10){
146             arduino->write("11");
147         }
148         if (vel_emb >= 11 & vel_emb <=20){
149             arduino->write("12");
150         }
151         if (vel_emb >= 21 & vel_emb <=30){
152             arduino->write("13");
153         }
154         if (vel_emb >= 31 & vel_emb <=40){
155             arduino->write("14");
156         }
157         if (vel_emb >= 41 & vel_emb <=50){
158             arduino->write("15");
159         }
160         if (vel_emb >= 51 & vel_emb <=60){
161             arduino->write("16");
162         }
163         if (vel_emb >= 61 & vel_emb <=70){
164             arduino->write("17");
165         }
166         if (vel_emb >= 71 & vel_emb <=80){
167             arduino->write("18");
168         }
169         if (vel_emb >= 81 & vel_emb <=90){
170             arduino->write("19");
171         }
172         if (vel_emb >= 91 & vel_emb <=100){
173             arduino->write("20");
174         }
175     }
176 }

```

```

137 }
138 }
139 void MainWindow::on_enviar_des_emb_9_clicked()
140 {
141     int x = ui->verticalSlider_2->value();
142     int y = 5;
143     ui->verticalSlider_2->setValue(1);
144     vel_fren=ui->lineEdit_5->text();
145     //explicit conversion turns floating-point number into integer: 'double' to 'int'
146     if (T > -1 & T < 100){
147         if (vel_fren >= 1 & vel_fren <=10){
148             arduino2->write("21");
149         }
150         if (vel_fren >= 11 & vel_fren <=20){
151             arduino2->write("22");
152         }
153         if (vel_fren >= 21 & vel_fren <=30){
154             arduino2->write("23");
155         }
156         if (vel_fren >= 31 & vel_fren <=40){
157             arduino2->write("24");
158         }
159         if (vel_fren >= 41 & vel_fren <=50){
160             arduino2->write("25");
161         }
162         if (vel_fren >= 51 & vel_fren <=60){
163             arduino2->write("26");
164         }
165         if (vel_fren >= 61 & vel_fren <=70){
166             arduino2->write("27");
167         }
168         if (vel_fren >= 71 & vel_fren <=80){
169             arduino2->write("28");
170         }
171         if (vel_fren >= 81 & vel_fren <=90){
172             arduino2->write("29");
173         }
174     }
175 }

```

```

137 }
138 if (vel_fren >= 91 & vel_fren <=100){
139     arduino2->write("30");
140 }
141 }
142 void MainWindow::on_enviar_des_emb_10_clicked()
143 {
144     int x = ui->verticalSlider_2->value();
145     int y = 5;
146     ui->verticalSlider_2->setValue(0);
147     vel_fren=ui->lineEdit_5->text();
148     //explicit conversion turns floating-point number into integer: 'double' to 'int'
149     if (U > -1 & U < 100){
150         if (vel_fren >= 1 & vel_fren <=10){
151             arduino2->write("31");
152         }
153         if (vel_fren >= 11 & vel_fren <=20){
154             arduino2->write("32");
155         }
156         if (vel_fren >= 21 & vel_fren <=30){
157             arduino2->write("33");
158         }
159         if (vel_fren >= 31 & vel_fren <=40){
160             arduino2->write("34");
161         }
162         if (vel_fren >= 41 & vel_fren <=50){
163             arduino2->write("35");
164         }
165         if (vel_fren >= 51 & vel_fren <=60){
166             arduino2->write("36");
167         }
168         if (vel_fren >= 61 & vel_fren <=70){
169             arduino2->write("37");
170         }
171         if (vel_fren >= 71 & vel_fren <=80){
172             arduino2->write("38");
173         }
174     }
175 }

```

```
Projects | mainwindow.cpp | MainWindow.on_enviar_des_emb_9_clicked() = void | Line 282, Col 6
interface_Conduccion
  interface_Conduccion.pro
  Headers
    mainwindow.h
  Sources
    main.cpp
    mainwindow.cpp
  Forms
    mainwindow.ui
  Resources

Open Documents
mainwindow.cpp

282 |         if (vel_fren >= 81 & vel_fren <=90) {
283 |             arduino2->write("13");
284 |         }
285 |
286 |         if (vel_fren >= 91 & vel_fren <=100) {
287 |             arduino2->write("40");
288 |         }
289 |     }
290 |
291 | void MainWindow::on_enviar_des_emb_11_clicked()
292 | {
293 |     int x = ui->verticalSlider_3->value();
294 |     Wx=5;
295 |     ui->verticalSlider_3->setValue(Wx);
296 |     vel_acel1=ui->lineEdit_6->text();
297 |
298 |     if (W > -1 & W < 101) {
299 |         if (vel_acel >= 1 & vel_acel <=10) {
300 |             arduino3->write("42");
301 |         }
302 |
303 |         if (vel_acel >= 11 & vel_acel <=20) {
304 |             arduino3->write("42");
305 |         }
306 |
307 |         if (vel_acel >= 21 & vel_acel <=30) {
308 |             arduino3->write("43");
309 |         }
310 |
311 |         if (vel_acel >= 31 & vel_acel <=40) {
312 |             arduino3->write("44");
313 |         }
314 |
315 |         if (vel_acel >= 41 & vel_acel <=50) {
316 |             arduino3->write("45");
317 |         }
318 |
319 |         if (vel_acel >= 51 & vel_acel <=60) {
320 |             arduino3->write("46");
321 |         }
322 |
323 |         if (vel_acel >= 61 & vel_acel <=70) {
324 |             arduino3->write("47");
325 |         }
326 |     }
327 | }
```

```
Projects | mainwindow.cpp | MainWindow.on_enviar_des_emb_11_clicked() = void | Line 369, Col 27
interface_Conduccion
  interface_Conduccion.pro
  Headers
    mainwindow.h
  Sources
    main.cpp
    mainwindow.cpp
  Forms
    mainwindow.ui
  Resources

Open Documents
mainwindow.cpp

369 |         if (vel_acel >= 71 & vel_acel <=80) {
370 |             arduino3->write("48");
371 |         }
372 |
373 |         if (vel_acel >= 81 & vel_acel <=90) {
374 |             arduino3->write("49");
375 |         }
376 |
377 |         if (vel_acel >= 91 & vel_acel <=100) {
378 |             arduino3->write("50");
379 |         }
380 |     }
381 |
382 | void MainWindow::on_enviar_des_emb_11_clicked()
383 | {
384 |     int x = ui->verticalSlider_3->value();
385 |     Wx=5;
386 |     ui->verticalSlider_3->setValue(Wx);
387 |     vel_acel1=ui->lineEdit_6->text();
388 |
389 |     if (W > -1 & W < 101) {
390 |         if (vel_acel >= 1 & vel_acel <=10) {
391 |             arduino3->write("51");
392 |         }
393 |
394 |         if (vel_acel >= 11 & vel_acel <=20) {
395 |             arduino3->write("52");
396 |         }
397 |
398 |         if (vel_acel >= 21 & vel_acel <=30) {
399 |             arduino3->write("53");
400 |         }
401 |
402 |         if (vel_acel >= 31 & vel_acel <=40) {
403 |             arduino3->write("54");
404 |         }
405 |
406 |         if (vel_acel >= 41 & vel_acel <=50) {
407 |             arduino3->write("55");
408 |         }
409 |
410 |         if (vel_acel >= 51 & vel_acel <=60) {
411 |             arduino3->write("56");
412 |         }
413 |     }
414 | }
```

```
Projects | mainwindow.cpp | MainWindow.on_enviar_des_emb_5_clicked() = void | Line 429, Col 47
interface_Conduccion
  interface_Conduccion.pro
  Headers
    mainwindow.h
  Sources
    main.cpp
    mainwindow.cpp
  Forms
    mainwindow.ui
  Resources

Open Documents
mainwindow.cpp

429 |     }
430 |
431 | void MainWindow::on_enviar_des_emb_2_clicked()
432 | {
433 |     set1=ui->lineEdit_7->text();
434 |     ui->lineEdit_7->setDisplay(set1);
435 |     set2 = ui->lineEdit_7->text();
436 |     //arduino->write(QString::number(set2));
437 |     //arduino->write(set2);
438 | }
439 |
440 | void MainWindow::on_enviar_des_emb_3_clicked()
441 | {
442 |     set3=ui->lineEdit_8->text();
443 |     ui->lineEdit_8->setDisplay(set3);
444 |     set4 = ui->lineEdit_8->text();
445 | }
446 |
447 | void MainWindow::on_enviar_des_emb_4_clicked()
448 | {
449 |     set5=ui->lineEdit_9->text();
450 |     ui->lineEdit_9->setDisplay(set5);
451 |     web5 = ui->lineEdit_9->text();
452 | }
453 |
454 | void MainWindow::on_enviar_des_emb_5_clicked()
455 | {
456 | }
```

```

429 void MainWindow::on_enviar_des_5_clicked()
430 {
431     setNum->lcdNumber_4->setValue(); //explicit conversion turns floating-point number into integer: 'double' to 'int'
432     ui->lcdNumber_10->setDisplay(set7); //explicit conversion turns floating-point number into integer: 'double' to 'int'
433     set8 = ui->lcdNumber_10->value(); //explicit conversion turns floating-point number into integer: 'double' to 'int'
434 }
435
436 void MainWindow::on_enviar_des_6_clicked()
437 {
438     setNum->lcdNumber_5->setValue(); //explicit conversion turns floating-point number into integer: 'double' to 'int'
439     ui->lcdNumber_11->setDisplay(set8); //explicit conversion turns floating-point number into integer: 'double' to 'int'
440     set9 = ui->lcdNumber_11->value(); //explicit conversion turns floating-point number into integer: 'double' to 'int'
441 }
442
443 void MainWindow::on_enviar_des_7_clicked()
444 {
445     setNum->lcdNumber_6->setValue(); //explicit conversion turns floating-point number into integer: 'double' to 'int'
446     ui->lcdNumber_12->setDisplay(set9); //explicit conversion turns floating-point number into integer: 'double' to 'int'
447     set10 = ui->lcdNumber_12->value(); //explicit conversion turns floating-point number into integer: 'double' to 'int'
448 }
449
450 void MainWindow::on_pushButton_clicked()
451 {
452     //arduino->write("567");
453
454     if (set2 >= 1 & set2 <=5 & set8 >= 1 & set8 <=25){
455         arduino->write("61");
456     }
457     if (set2 >= 1 & set2 <=5 & set8 >= 26 & set8 <=50){
458         arduino->write("62");
459     }
460     if (set2 >= 1 & set2 <=5 & set8 >= 51 & set8 <=75){
461         arduino->write("63");
462     }
463     if (set2 >= 1 & set2 <=5 & set8 >= 76 & set8 <=100){
464         arduino->write("64");
465     }
466
467     if (set2 >= 6 & set2 <=10 & set8 >= 1 & set8 <=25){
468         arduino->write("65");
469     }
470     if (set2 >= 6 & set2 <=10 & set8 >= 26 & set8 <=50){
471         arduino->write("66");
472     }
473     if (set2 >= 6 & set2 <=10 & set8 >= 51 & set8 <=75){
474         arduino->write("67");
475     }
476     if (set2 >= 6 & set2 <=10 & set8 >= 76 & set8 <=100){
477         arduino->write("68");
478     }
479 }

```

```

479 }
480
481 if (set2 >= 11 & set2 <=15 & set8 >= 1 & set8 <=25){
482     arduino->write("69");
483 }
484 if (set2 >= 11 & set2 <=15 & set8 >= 26 & set8 <=50){
485     arduino->write("70");
486 }
487 if (set2 >= 11 & set2 <=15 & set8 >= 51 & set8 <=75){
488     arduino->write("71");
489 }
490 if (set2 >= 11 & set2 <=15 & set8 >= 76 & set8 <=100){
491     arduino->write("72");
492 }
493
494 if (set2 >= 16 & set2 <=20 & set8 >= 1 & set8 <=25){
495     arduino->write("73");
496 }
497 if (set2 >= 16 & set2 <=20 & set8 >= 26 & set8 <=50){
498     arduino->write("74");
499 }
500 if (set2 >= 16 & set2 <=20 & set8 >= 51 & set8 <=75){
501     arduino->write("75");
502 }
503 if (set2 >= 16 & set2 <=20 & set8 >= 76 & set8 <=100){
504     arduino->write("76");
505 }
506
507 if (set2 >= 21 & set2 <=25 & set8 >= 1 & set8 <=25){
508     arduino->write("77");
509 }
510 if (set2 >= 21 & set2 <=25 & set8 >= 26 & set8 <=50){
511     arduino->write("78");
512 }
513 if (set2 >= 21 & set2 <=25 & set8 >= 51 & set8 <=75){
514     arduino->write("79");
515 }
516 if (set2 >= 21 & set2 <=25 & set8 >= 76 & set8 <=100){
517     arduino->write("80");
518 }
519
520 if (set2 >= 26 & set2 <=30 & set8 >= 1 & set8 <=25){
521     arduino->write("81");
522 }
523 if (set2 >= 26 & set2 <=30 & set8 >= 26 & set8 <=50){
524     arduino->write("82");
525 }
526 if (set2 >= 26 & set2 <=30 & set8 >= 51 & set8 <=75){
527     arduino->write("83");
528 }

```

```

529 }
530
531 if (set2 >= 31 & set2 <=35 & set8 >= 76 & set8 <=100){
532     arduino->write("84");
533 }
534
535 if (set2 >= 31 & set2 <=35 & set8 >= 1 & set8 <=25){
536     arduino->write("85");
537 }
538 if (set2 >= 31 & set2 <=35 & set8 >= 26 & set8 <=50){
539     arduino->write("86");
540 }
541 if (set2 >= 31 & set2 <=35 & set8 >= 51 & set8 <=75){
542     arduino->write("87");
543 }
544 if (set2 >= 31 & set2 <=35 & set8 >= 76 & set8 <=100){
545     arduino->write("88");
546 }
547
548 if (set2 >= 36 & set2 <=40 & set8 >= 1 & set8 <=25){
549     arduino->write("89");
550 }
551 if (set2 >= 36 & set2 <=40 & set8 >= 26 & set8 <=50){
552     arduino->write("90");
553 }
554 if (set2 >= 36 & set2 <=40 & set8 >= 51 & set8 <=75){
555     arduino->write("91");
556 }
557 if (set2 >= 36 & set2 <=40 & set8 >= 76 & set8 <=100){
558     arduino->write("92");
559 }
560
561 if (set2 >= 41 & set2 <=45 & set8 >= 1 & set8 <=25){
562     arduino->write("93");
563 }
564 if (set2 >= 41 & set2 <=45 & set8 >= 26 & set8 <=50){
565     arduino->write("94");
566 }
567 if (set2 >= 41 & set2 <=45 & set8 >= 51 & set8 <=75){
568     arduino->write("95");
569 }
570 if (set2 >= 41 & set2 <=45 & set8 >= 76 & set8 <=100){
571     arduino->write("96");
572 }
573
574 if (set2 >= 46 & set2 <=50 & set8 >= 1 & set8 <=25){
575     arduino->write("97");
576 }
577 if (set2 >= 46 & set2 <=50 & set8 >= 26 & set8 <=50){
578     arduino->write("98");
579 }

```

```

146     }
147     if (set2 >= 46 & set3 <=58 & set8 >= 51 & set8 <=75){
148         arduino->write("997");
149     }
150     if (set2 >= 46 & set3 <=59 & set8 >= 76 & set8 <=100){
151         arduino->write("1007");
152     }
153     }
154     if (set2 >= 51 & set3 <=55 & set8 >= 1 & set8 <=25){
155         arduino->write("1017");
156     }
157     if (set2 >= 51 & set3 <=55 & set8 >= 26 & set8 <=50){
158         arduino->write("1027");
159     }
160     if (set2 >= 51 & set3 <=55 & set8 >= 51 & set8 <=75){
161         arduino->write("1037");
162     }
163     if (set2 >= 51 & set3 <=55 & set8 >= 76 & set8 <=100){
164         arduino->write("1047");
165     }
166     }
167     if (set2 >= 56 & set3 <=68 & set8 >= 1 & set8 <=25){
168         arduino->write("1057");
169     }
170     if (set2 >= 56 & set3 <=68 & set8 >= 26 & set8 <=50){
171         arduino->write("1067");
172     }
173     if (set2 >= 56 & set3 <=68 & set8 >= 51 & set8 <=75){
174         arduino->write("1077");
175     }
176     if (set2 >= 56 & set3 <=68 & set8 >= 76 & set8 <=100){
177         arduino->write("1087");
178     }
179     }
180     if (set2 >= 61 & set3 <=65 & set8 >= 1 & set8 <=25){
181         arduino->write("1097");
182     }
183     if (set2 >= 61 & set3 <=65 & set8 >= 26 & set8 <=50){
184         arduino->write("1107");
185     }
186     if (set2 >= 61 & set3 <=65 & set8 >= 51 & set8 <=75){
187         arduino->write("1117");
188     }
189     if (set2 >= 61 & set3 <=65 & set8 >= 76 & set8 <=100){
190         arduino->write("1127");
191     }
192     }
193     if (set2 >= 66 & set3 <=78 & set8 >= 1 & set8 <=25){
194         arduino->write("1137");
195     }

```

```

196     }
197     if (set2 >= 66 & set3 <=79 & set8 >= 26 & set8 <=50){
198         arduino->write("1147");
199     }
200     if (set2 >= 66 & set3 <=79 & set8 >= 51 & set8 <=75){
201         arduino->write("1157");
202     }
203     if (set2 >= 66 & set3 <=79 & set8 >= 76 & set8 <=100){
204         arduino->write("1167");
205     }
206     }
207     if (set2 >= 71 & set3 <=75 & set8 >= 1 & set8 <=25){
208         arduino->write("1177");
209     }
210     if (set2 >= 71 & set3 <=75 & set8 >= 26 & set8 <=50){
211         arduino->write("1187");
212     }
213     if (set2 >= 71 & set3 <=75 & set8 >= 51 & set8 <=75){
214         arduino->write("1197");
215     }
216     if (set2 >= 71 & set3 <=75 & set8 >= 76 & set8 <=100){
217         arduino->write("1207");
218     }
219     }
220     if (set2 >= 76 & set3 <=80 & set8 >= 1 & set8 <=25){
221         arduino->write("1217");
222     }
223     if (set2 >= 76 & set3 <=80 & set8 >= 26 & set8 <=50){
224         arduino->write("1227");
225     }
226     if (set2 >= 76 & set3 <=80 & set8 >= 51 & set8 <=75){
227         arduino->write("1237");
228     }
229     if (set2 >= 76 & set3 <=80 & set8 >= 76 & set8 <=100){
230         arduino->write("1247");
231     }
232     }
233     if (set2 >= 81 & set3 <=85 & set8 >= 1 & set8 <=25){
234         arduino->write("1257");
235     }
236     if (set2 >= 81 & set3 <=85 & set8 >= 26 & set8 <=50){
237         arduino->write("1267");
238     }
239     if (set2 >= 81 & set3 <=85 & set8 >= 51 & set8 <=75){
240         arduino->write("1277");
241     }
242     if (set2 >= 81 & set3 <=85 & set8 >= 76 & set8 <=100){
243         arduino->write("1287");
244     }
245     }

```

```

246     }
247     if (set2 >= 86 & set3 <=98 & set8 >= 1 & set8 <=25){
248         arduino->write("1297");
249     }
250     if (set2 >= 86 & set3 <=98 & set8 >= 26 & set8 <=50){
251         arduino->write("1307");
252     }
253     if (set2 >= 86 & set3 <=98 & set8 >= 51 & set8 <=75){
254         arduino->write("1317");
255     }
256     if (set2 >= 86 & set3 <=98 & set8 >= 76 & set8 <=100){
257         arduino->write("1327");
258     }
259     }
260     if (set2 >= 91 & set3 <=95 & set8 >= 1 & set8 <=25){
261         arduino->write("1337");
262     }
263     if (set2 >= 91 & set3 <=95 & set8 >= 26 & set8 <=50){
264         arduino->write("1347");
265     }
266     if (set2 >= 91 & set3 <=95 & set8 >= 51 & set8 <=75){
267         arduino->write("1357");
268     }
269     if (set2 >= 91 & set3 <=95 & set8 >= 76 & set8 <=100){
270         arduino->write("1367");
271     }
272     }
273     if (set2 >= 96 & set3 <=100 & set8 >= 1 & set8 <=25){
274         arduino->write("1377");
275     }
276     if (set2 >= 96 & set3 <=100 & set8 >= 26 & set8 <=50){
277         arduino->write("1387");
278     }
279     if (set2 >= 96 & set3 <=100 & set8 >= 51 & set8 <=75){
280         arduino->write("1397");
281     }
282     if (set2 >= 96 & set3 <=100 & set8 >= 76 & set8 <=100){
283         arduino->write("1407");
284     }
285     }
286     if (set2 >= 1 & set3 <=5 & set8 >= 1 & set8 <=25){
287         arduino->write("1417");
288     }
289     if (set2 >= 1 & set3 <=5 & set8 >= 26 & set8 <=50){
290         arduino->write("1427");
291     }
292     }

```

```
mainwindow.cpp 1 x MainWindow.on_pushButton_clicked() -> void Line: 771, Col: 6
interface_Conduccion
  interface_Conduccion.pro
  Headers
    mainwindow.h
  Sources
    main.cpp
    mainwindow.cpp
  Forms
  mainwindow.ui
  Resources

Open Documents
mainwindow.cpp

770     if (set4 >= 1 & set4 <= 5 & set10 >= 51 & set18 <= 75){
771         arduino2->write("143");
772     }
773     if (set4 >= 1 & set4 <= 5 & set10 >= 76 & set18 <= 100){
774         arduino2->write("144");
775     }
776     if (set4 >= 6 & set4 <= 10 & set10 >= 1 & set18 <= 25){
777         arduino2->write("145");
778     }
779     if (set4 >= 6 & set4 <= 10 & set10 >= 26 & set18 <= 50){
780         arduino2->write("146");
781     }
782     if (set4 >= 6 & set4 <= 10 & set10 >= 51 & set18 <= 75){
783         arduino2->write("147");
784     }
785     if (set4 >= 6 & set4 <= 10 & set10 >= 76 & set18 <= 100){
786         arduino2->write("148");
787     }
788     if (set4 >= 11 & set4 <= 15 & set10 >= 1 & set18 <= 25){
789         arduino2->write("149");
790     }
791     if (set4 >= 11 & set4 <= 15 & set10 >= 26 & set18 <= 50){
792         arduino2->write("150");
793     }
794     if (set4 >= 11 & set4 <= 15 & set10 >= 51 & set18 <= 75){
795         arduino2->write("151");
796     }
797     if (set4 >= 11 & set4 <= 15 & set10 >= 76 & set18 <= 100){
798         arduino2->write("152");
799     }
800     if (set4 >= 16 & set4 <= 20 & set10 >= 1 & set18 <= 25){
801         arduino2->write("153");
802     }
803     if (set4 >= 16 & set4 <= 20 & set10 >= 26 & set18 <= 50){
804         arduino2->write("154");
805     }
806     if (set4 >= 16 & set4 <= 20 & set10 >= 51 & set18 <= 75){
807         arduino2->write("155");
808     }
809     if (set4 >= 16 & set4 <= 20 & set10 >= 76 & set18 <= 100){
810         arduino2->write("156");
811     }
812     if (set4 >= 21 & set4 <= 25 & set10 >= 1 & set18 <= 25){
813         arduino2->write("157");
814     }
815 }
```

```
mainwindow.cpp 1 x MainWindow.on_pushButton_clicked() -> void Line: 820, Col: 1
interface_Conduccion
  interface_Conduccion.pro
  Headers
    mainwindow.h
  Sources
    main.cpp
    mainwindow.cpp
  Forms
  mainwindow.ui
  Resources

Open Documents
mainwindow.cpp

816     if (set4 >= 21 & set4 <= 25 & set10 >= 26 & set18 <= 50){
817         arduino2->write("158");
818     }
819     if (set4 >= 21 & set4 <= 25 & set10 >= 51 & set18 <= 75){
820         arduino2->write("159");
821     }
822     if (set4 >= 21 & set4 <= 25 & set10 >= 76 & set18 <= 100){
823         arduino2->write("160");
824     }
825     if (set4 >= 26 & set4 <= 30 & set10 >= 1 & set18 <= 25){
826         arduino2->write("161");
827     }
828     if (set4 >= 26 & set4 <= 30 & set10 >= 26 & set18 <= 50){
829         arduino2->write("162");
830     }
831     if (set4 >= 26 & set4 <= 30 & set10 >= 51 & set18 <= 75){
832         arduino2->write("163");
833     }
834     if (set4 >= 26 & set4 <= 30 & set10 >= 76 & set18 <= 100){
835         arduino2->write("164");
836     }
837     if (set4 >= 31 & set4 <= 35 & set10 >= 1 & set18 <= 25){
838         arduino2->write("165");
839     }
840     if (set4 >= 31 & set4 <= 35 & set10 >= 26 & set18 <= 50){
841         arduino2->write("166");
842     }
843     if (set4 >= 31 & set4 <= 35 & set10 >= 51 & set18 <= 75){
844         arduino2->write("167");
845     }
846     if (set4 >= 31 & set4 <= 35 & set10 >= 76 & set18 <= 100){
847         arduino2->write("168");
848     }
849     if (set4 >= 36 & set4 <= 40 & set10 >= 1 & set18 <= 25){
850         arduino2->write("169");
851     }
852     if (set4 >= 36 & set4 <= 40 & set10 >= 26 & set18 <= 50){
853         arduino2->write("170");
854     }
855     if (set4 >= 36 & set4 <= 40 & set10 >= 51 & set18 <= 75){
856         arduino2->write("171");
857     }
858     if (set4 >= 36 & set4 <= 40 & set10 >= 76 & set18 <= 100){
859         arduino2->write("172");
860     }
861 }
```

```
mainwindow.cpp 1 x MainWindow.on_pushButton_clicked() -> void Line: 869, Col: 61
interface_Conduccion
  interface_Conduccion.pro
  Headers
    mainwindow.h
  Sources
    main.cpp
    mainwindow.cpp
  Forms
  mainwindow.ui
  Resources

Open Documents
mainwindow.cpp

862     if (set4 >= 41 & set4 <= 45 & set10 >= 1 & set18 <= 25){
863         arduino2->write("173");
864     }
865     if (set4 >= 41 & set4 <= 45 & set10 >= 26 & set18 <= 50){
866         arduino2->write("174");
867     }
868     if (set4 >= 41 & set4 <= 45 & set10 >= 51 & set18 <= 75){
869         arduino2->write("175");
870     }
871     if (set4 >= 41 & set4 <= 45 & set10 >= 76 & set18 <= 100){
872         arduino2->write("176");
873     }
874     if (set4 >= 46 & set4 <= 50 & set10 >= 1 & set18 <= 25){
875         arduino2->write("177");
876     }
877     if (set4 >= 46 & set4 <= 50 & set10 >= 26 & set18 <= 50){
878         arduino2->write("178");
879     }
880     if (set4 >= 46 & set4 <= 50 & set10 >= 51 & set18 <= 75){
881         arduino2->write("179");
882     }
883     if (set4 >= 46 & set4 <= 50 & set10 >= 76 & set18 <= 100){
884         arduino2->write("180");
885     }
886     if (set4 >= 51 & set4 <= 55 & set10 >= 1 & set18 <= 25){
887         arduino2->write("181");
888     }
889     if (set4 >= 51 & set4 <= 55 & set10 >= 26 & set18 <= 50){
890         arduino2->write("182");
891     }
892     if (set4 >= 51 & set4 <= 55 & set10 >= 51 & set18 <= 75){
893         arduino2->write("183");
894     }
895     if (set4 >= 51 & set4 <= 55 & set10 >= 76 & set18 <= 100){
896         arduino2->write("184");
897     }
898     if (set4 >= 56 & set4 <= 60 & set10 >= 1 & set18 <= 25){
899         arduino2->write("185");
900     }
901     if (set4 >= 56 & set4 <= 60 & set10 >= 26 & set18 <= 50){
902         arduino2->write("186");
903     }
904     if (set4 >= 56 & set4 <= 60 & set10 >= 51 & set18 <= 75){
905         arduino2->write("187");
906     }
907     if (set4 >= 56 & set4 <= 60 & set10 >= 76 & set18 <= 100){
908         arduino2->write("188");
909     }
910 }
```



```
mainwindow.cpp | Main Window on_pushButton_clicked() | void | Line: 1065, Col: 6
interface_Conduccion
interface_Conduccion.pro
Headers
mainwindow.h
Sources
main.cpp
mainwindow.cpp
Forms
mainwindow.ui
Resources
1065 | if (set6 >= 16 & set6 <= 20 & set12 >= 1 & set12 <= 25){
1066 |     arduino3->write("133");
1067 | }
1068 | if (set6 >= 16 & set6 <= 20 & set12 >= 26 & set12 <= 50){
1069 |     arduino3->write("134");
1070 | }
1071 | if (set6 >= 16 & set6 <= 20 & set12 >= 51 & set12 <= 75){
1072 |     arduino3->write("135");
1073 | }
1074 | if (set6 >= 16 & set6 <= 20 & set12 >= 76 & set12 <= 100){
1075 |     arduino3->write("136");
1076 | }
1077 | }
1078 | if (set6 >= 21 & set6 <= 25 & set12 >= 1 & set12 <= 25){
1079 |     arduino3->write("137");
1080 | }
1081 | if (set6 >= 21 & set6 <= 25 & set12 >= 26 & set12 <= 50){
1082 |     arduino3->write("138");
1083 | }
1084 | if (set6 >= 21 & set6 <= 25 & set12 >= 51 & set12 <= 75){
1085 |     arduino3->write("139");
1086 | }
1087 | if (set6 >= 21 & set6 <= 25 & set12 >= 76 & set12 <= 100){
1088 |     arduino3->write("140");
1089 | }
1090 | }
1091 | if (set6 >= 26 & set6 <= 30 & set12 >= 1 & set12 <= 25){
1092 |     arduino3->write("141");
1093 | }
1094 | if (set6 >= 26 & set6 <= 30 & set12 >= 26 & set12 <= 50){
1095 |     arduino3->write("142");
1096 | }
1097 | if (set6 >= 26 & set6 <= 30 & set12 >= 51 & set12 <= 75){
1098 |     arduino3->write("143");
1099 | }
1100 | if (set6 >= 26 & set6 <= 30 & set12 >= 76 & set12 <= 100){
1101 |     arduino3->write("144");
1102 | }
1103 | }
1104 | if (set6 >= 31 & set6 <= 35 & set12 >= 1 & set12 <= 25){
1105 |     arduino3->write("145");
1106 | }
1107 | if (set6 >= 31 & set6 <= 35 & set12 >= 26 & set12 <= 50){
1108 |     arduino3->write("146");
1109 | }
1110 | if (set6 >= 31 & set6 <= 35 & set12 >= 51 & set12 <= 75){
1111 |     arduino3->write("147");
1112 | }
1113 | }
```

```
mainwindow.cpp | Main Window on_pushButton_clicked() | void | Line: 1114, Col: 6
1114 | if (set6 >= 21 & set6 <= 25 & set12 >= 76 & set12 <= 100){
1115 |     arduino3->write("148");
1116 | }
1117 | if (set6 >= 26 & set6 <= 30 & set12 >= 1 & set12 <= 25){
1118 |     arduino3->write("149");
1119 | }
1120 | if (set6 >= 26 & set6 <= 30 & set12 >= 26 & set12 <= 50){
1121 |     arduino3->write("150");
1122 | }
1123 | if (set6 >= 26 & set6 <= 30 & set12 >= 51 & set12 <= 75){
1124 |     arduino3->write("151");
1125 | }
1126 | if (set6 >= 26 & set6 <= 30 & set12 >= 76 & set12 <= 100){
1127 |     arduino3->write("152");
1128 | }
1129 | }
1130 | if (set6 >= 41 & set6 <= 45 & set12 >= 1 & set12 <= 25){
1131 |     arduino3->write("153");
1132 | }
1133 | if (set6 >= 41 & set6 <= 45 & set12 >= 26 & set12 <= 50){
1134 |     arduino3->write("154");
1135 | }
1136 | if (set6 >= 41 & set6 <= 45 & set12 >= 51 & set12 <= 75){
1137 |     arduino3->write("155");
1138 | }
1139 | if (set6 >= 41 & set6 <= 45 & set12 >= 76 & set12 <= 100){
1140 |     arduino3->write("156");
1141 | }
1142 | }
1143 | if (set6 >= 46 & set6 <= 50 & set12 >= 1 & set12 <= 25){
1144 |     arduino3->write("157");
1145 | }
1146 | if (set6 >= 46 & set6 <= 50 & set12 >= 26 & set12 <= 50){
1147 |     arduino3->write("158");
1148 | }
1149 | if (set6 >= 46 & set6 <= 50 & set12 >= 51 & set12 <= 75){
1150 |     arduino3->write("159");
1151 | }
1152 | if (set6 >= 46 & set6 <= 50 & set12 >= 76 & set12 <= 100){
1153 |     arduino3->write("160");
1154 | }
1155 | }
1156 | if (set6 >= 51 & set6 <= 55 & set12 >= 1 & set12 <= 25){
1157 |     arduino3->write("161");
1158 | }
1159 | if (set6 >= 51 & set6 <= 55 & set12 >= 26 & set12 <= 50){
1160 |     arduino3->write("162");
1161 | }
1162 | }
```

```
mainwindow.cpp | Main Window on_pushButton_clicked() | void | Line: 1163, Col: 6
1163 | if (set6 >= 51 & set6 <= 55 & set12 >= 51 & set12 <= 75){
1164 |     arduino3->write("163");
1165 | }
1166 | if (set6 >= 51 & set6 <= 55 & set12 >= 76 & set12 <= 100){
1167 |     arduino3->write("164");
1168 | }
1169 | }
1170 | if (set6 >= 56 & set6 <= 60 & set12 >= 1 & set12 <= 25){
1171 |     arduino3->write("165");
1172 | }
1173 | if (set6 >= 56 & set6 <= 60 & set12 >= 26 & set12 <= 50){
1174 |     arduino3->write("166");
1175 | }
1176 | if (set6 >= 56 & set6 <= 60 & set12 >= 51 & set12 <= 75){
1177 |     arduino3->write("167");
1178 | }
1179 | if (set6 >= 56 & set6 <= 60 & set12 >= 76 & set12 <= 100){
1180 |     arduino3->write("168");
1181 | }
1182 | }
1183 | if (set6 >= 61 & set6 <= 65 & set12 >= 1 & set12 <= 25){
1184 |     arduino3->write("169");
1185 | }
1186 | if (set6 >= 61 & set6 <= 65 & set12 >= 26 & set12 <= 50){
1187 |     arduino3->write("170");
1188 | }
1189 | if (set6 >= 61 & set6 <= 65 & set12 >= 51 & set12 <= 75){
1190 |     arduino3->write("171");
1191 | }
1192 | if (set6 >= 61 & set6 <= 65 & set12 >= 76 & set12 <= 100){
1193 |     arduino3->write("172");
1194 | }
1195 | }
1196 | if (set6 >= 66 & set6 <= 70 & set12 >= 1 & set12 <= 25){
1197 |     arduino3->write("173");
1198 | }
1199 | if (set6 >= 66 & set6 <= 70 & set12 >= 26 & set12 <= 50){
1200 |     arduino3->write("174");
1201 | }
1202 | if (set6 >= 66 & set6 <= 70 & set12 >= 51 & set12 <= 75){
1203 |     arduino3->write("175");
1204 | }
1205 | if (set6 >= 66 & set6 <= 70 & set12 >= 76 & set12 <= 100){
1206 |     arduino3->write("176");
1207 | }
1208 | }
1209 | if (set6 >= 71 & set6 <= 75 & set12 >= 1 & set12 <= 25){
1210 |     arduino3->write("177");
1211 | }
1212 | }
```


ANEXO D: PROGRAMACIÓN EN ARDUINO

```
Programacion_pedales
//#include <Wire.h>
const int dirPinEmbr = 8;
const int stepPinEmbr = 9;

const int LED=7;

int cond=1;
const int pulEmbrMas = 12;
const int pulEmbrMenos = 13;

int buttonPushCounter1 = 0; // Contador de pulsaciones
int valorPulsador1 = 0; // Estado actual del pulsador
int lastButtonState1 = 0; // Estado anterior del pulsador

int buttonPushCounter2 = 0; // Contador de pulsaciones
int valorPulsador2 = 0; // Estado actual del pulsador
int lastButtonState2 = 0; // Estado anterior del pulsador

unsigned int pedido=0;
int stepDelay;
int val;
char env1;

/*void respuestaEvento() //Funcion que se ejecuta al recibir una peticion.
{
}
void recibidoEvento(int recepcion) //Funcion que se ejecuta al recibir datos.
```

```
Programacion_pedales

while( Wire.available())
{
    pedido = Wire.read();
}
}+//

void setup() {
    //Wire.begin(1);
    Serial.begin(9600);
    Serial.setTimeout(100);
    pinMode(pulEmbrMas, INPUT);
    pinMode(pulEmbrMenos, INPUT);

    pinMode(dirPinEmbr, OUTPUT);
    pinMode(stepPinEmbr, OUTPUT);

    digitalWrite(LED,LOW);

    /*Wire.onRequest(respuestaEvento);
    Wire.onReceive(recibidoEvento);*/
}

void loop() {
    valorPulsador1 = digitalRead(pulEmbrMas);
    valorPulsador2 = digitalRead(pulEmbrMenos);

    //env1 =Serial.read();
    //val = String(env1).toInt();
    //env1 = map(val,0,100,0,3000);

    val = Serial.parseInt();
```

Programacion_pedales

```
if (val == 1){
  giro1(stepPinEmbr, dirPinEmbr);
}
if (val == 2){
  giro2(stepPinEmbr, dirPinEmbr);
}
if (val == 3){
  giro3(stepPinEmbr, dirPinEmbr);
}
if (val == 4){
  giro4(stepPinEmbr, dirPinEmbr);
}
if (val == 5){
  giro5(stepPinEmbr, dirPinEmbr);
}
if (val == 6){
  giro6(stepPinEmbr, dirPinEmbr);
}
if (val == 7){
  giro7(stepPinEmbr, dirPinEmbr);
}
if (val == 8){
  giro8(stepPinEmbr, dirPinEmbr);
}
if (val == 9){
  giro9(stepPinEmbr, dirPinEmbr);
}
if (val == 10){
  giro10(stepPinEmbr, dirPinEmbr);
}
if (val == 11){
  giro11(stepPinEmbr, dirPinEmbr);
}
if (val == 12){
  giro12(stepPinEmbr, dirPinEmbr);
```

Programacion_pedales

```

}
if (val == 13){
  giro13(stepPinEmbr, dirPinEmbr);
}
if (val == 14){
  giro14(stepPinEmbr, dirPinEmbr);
}
if (val == 15){
  giro15(stepPinEmbr, dirPinEmbr);
}
if (val == 16){
  giro16(stepPinEmbr, dirPinEmbr);
}
if (val == 17){
  giro17(stepPinEmbr, dirPinEmbr);
}
if (val == 18){
  giro18(stepPinEmbr, dirPinEmbr);
}
if (val == 19){
  giro19(stepPinEmbr, dirPinEmbr);
}
if (val == 20){
  giro20(stepPinEmbr, dirPinEmbr);
}

if (val == 61){
  do{
    valorPulsador1 = digitalRead(pulEmbrMas);
    valorPulsador2 = digitalRead(pulEmbrMenos);
    Serial.setTimeout(100);
    const int steps = 150;
    int stepDelay = 600;
```

Programacion_pedales

```
if (valorPulsador2 != lastButtonState2) {
  if (buttonPushCounter2 <= -1){
    if (valorPulsador2 == HIGH) {
      buttonPushCounter2++;
      buttonPushCounter1--;
      //Serial.println("Pulsaste EM-");
      digitalWrite(dirPinEmbr, HIGH);
      for (int x = 0; x < steps + 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB2:");
      Serial.println(buttonPushCounter2);*/
    }
  }
}
```

```
if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0){
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB1:");
      Serial.println(buttonPushCounter1);*/
    }
  }
}
```

Programacion_pedales

```
}

}

}

}while(cond==1);
}

if (val == 62){

  do{
    valorPulsador1 = digitalRead(pulEmbrMas);
    valorPulsador2 = digitalRead(pulEmbrMenos);
    Serial.setTimeout(100);
    const int steps = 150;
    int stepDelay = 400;

    if (valorPulsador2 != lastButtonState2) {
      if (buttonPushCounter2 <= -1){
        if (valorPulsador2 == HIGH) {
          buttonPushCounter2++;
          buttonPushCounter1--;
          //Serial.println("Pulsaste EM-");
          digitalWrite(dirPinEmbr, HIGH);
          for (int x = 0; x < steps + 1; x++) {
            digitalWrite(stepPinEmbr, HIGH);
            delayMicroseconds(stepDelay);
            digitalWrite(stepPinEmbr, LOW);
            delayMicroseconds(stepDelay);
          }
          /*Serial.println("EMB2:");
          Serial.println(buttonPushCounter2);*/
        }
      }
    }
  }
}
```

Programacion_pedales

```

}
}

if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0){
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB1:");
      Serial.println(buttonPushCounter1);*/
    }
  }
}

}while(cond==1);
}

if (val == 63){
  do{
    valorPulsador1 = digitalRead(pulEmbrMas);
    valorPulsador2 = digitalRead(pulEmbrMenos);
    Serial.setTimeout(100);
    const int steps = 150;
    int stepDelay = 250;

```

Programacion_pedales

```

if (valorPulsador2 != lastButtonState2) {
  if (buttonPushCounter2 <= -1){
    if (valorPulsador2 == HIGH) {
      buttonPushCounter2++;
      buttonPushCounter1--;
      //Serial.println("Pulsaste EM-");
      digitalWrite(dirPinEmbr, HIGH);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB2:");
      Serial.println(buttonPushCounter2);*/
    }
  }
}

if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0){
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB1:");

```

Programacion_pedales

```
Serial.println(buttonPushCounter1);*/
}

}

}while(cond==1);
}

if (val == 64){

do{
valorPulsador1 = digitalRead(pulEmbrMas);
valorPulsador2 = digitalRead(pulEmbrMenos);
Serial.setTimeout(100);
const int steps = 150;
int stepDelay = 50;

if (valorPulsador2 != lastButtonState2) {
if (buttonPushCounter2 <= -1){
if (valorPulsador2 == HIGH) {
buttonPushCounter2++;
buttonPushCounter1--;
//Serial.println("Pulsaste EM-");
digitalWrite(dirPinEmbr, HIGH);
for (int x = 0; x < steps * 1; x++) {
digitalWrite(stepPinEmbr, HIGH);
delayMicroseconds(stepDelay);
digitalWrite(stepPinEmbr, LOW);
delayMicroseconds(stepDelay);
}
/*Serial.println("EMB2:");
Serial.println(buttonPushCounter2);*/
}
}


```

Programacion_pedales

```

}
}

if (valorPulsador1 != lastButtonState1) {
if (buttonPushCounter1 <= 0){
if (valorPulsador1 == HIGH) {
buttonPushCounter1++;
buttonPushCounter2--;
//Serial.println("Pulsaste EM+");
digitalWrite(dirPinEmbr, LOW);
for (int x = 0; x < steps * 1; x++) {
digitalWrite(stepPinEmbr, HIGH);
delayMicroseconds(stepDelay);
digitalWrite(stepPinEmbr, LOW);
delayMicroseconds(stepDelay);
}
/*Serial.println("EMB1:");
Serial.println(buttonPushCounter1);*/
}
}
}

}while(cond==1);
}

if (val == 65){

do{
valorPulsador1 = digitalRead(pulEmbrMas);
valorPulsador2 = digitalRead(pulEmbrMenos);
Serial.setTimeout(100);
}


```

Programacion_pedales

```
const int steps = 300;
int stepDelay = 600;

if (valorPulsador2 != lastButtonState2) {
  if (buttonPushCounter2 <= -1){
    if (valorPulsador2 == HIGH) {
      buttonPushCounter2++;
      buttonPushCounter1--;
      //Serial.println("Pulsaste EM-");
      digitalWrite(dirPinEmbr, HIGH);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EM2:");
      Serial.println(buttonPushCounter2);*/
    }
  }
}

if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0){
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
```

Programacion_pedales

```
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EM1:");
      Serial.println(buttonPushCounter1);*/
    }
  }
}

}while(cond==1);
}

if (val == 66){
  do{
    valorPulsador1 = digitalRead(pulEmbrMas);
    valorPulsador2 = digitalRead(pulEmbrMence);
    Serial.setTimeout(100);
    const int steps = 300;
    int stepDelay = 400;

    if (valorPulsador2 != lastButtonState2) {
      if (buttonPushCounter2 <= -1){
        if (valorPulsador2 == HIGH) {
          buttonPushCounter2++;
          buttonPushCounter1--;
          //Serial.println("Pulsaste EM-");
          digitalWrite(dirPinEmbr, HIGH);
          for (int x = 0; x < steps * 1; x++) {
            digitalWrite(stepPinEmbr, HIGH);
            delayMicroseconds(stepDelay);
            digitalWrite(stepPinEmbr, LOW);
            delayMicroseconds(stepDelay);
```

Programacion_pedales

```
    }
    /*Serial.println("EMB2:");
    Serial.println(buttonPushCounter2);*/
  }
}

if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0){
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB1:");
      Serial.println(buttonPushCounter1);*/
    }
  }
}

}while(cond==1);
}

if (val == 67){
  do{
    valorPulsador1 = digitalRead(pulEmbrMas);
```

Programacion_pedales

```
valorPulsador2 = digitalRead(pulEmbrMenos);
Serial.setTimeout(100);
const int steps = 300;
int stepDelay = 250;

if (valorPulsador2 != lastButtonState2) {
  if (buttonPushCounter2 <= -1){
    if (valorPulsador2 == HIGH) {
      buttonPushCounter2++;
      buttonPushCounter1--;
      //Serial.println("Pulsaste EM-");
      digitalWrite(dirPinEmbr, HIGH);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB2:");
      Serial.println(buttonPushCounter2);*/
    }
  }
}

if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0){
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
```

Programacion_pedales

```

        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
    }
    /*Serial.println("EME1:");
    Serial.println(buttonPushCounter1);*/
}

}

}while(cond==1);
}

if (val == 68) {

    do{
        valorPulsador1 = digitalRead(pulEmbrMas);
        valorPulsador2 = digitalRead(pulEmbrMenos);
        Serial.setTimeout(100);
        const int steps = 300;
        int stepDelay = 50;

        if (valorPulsador2 != lastButtonState2) {
            if (buttonPushCounter2 <= -1){
                if (valorPulsador2 == HIGH) {
                    buttonPushCounter2++;
                    buttonPushCounter1--;
                    //Serial.println("Pulsaste EM-");
                    digitalWrite(dirPinEmbr, HIGH);
                    for (int x = 0; x < steps * 1; x++) {
                        digitalWrite(stepPinEmbr, HIGH);
                        delayMicroseconds(stepDelay);
                        digitalWrite(stepPinEmbr, LOW);

```

Programacion_pedales

```

        delayMicroseconds(stepDelay);
    }
    /*Serial.println("EME2:");
    Serial.println(buttonPushCounter2);*/
}

}

if (valorPulsador1 != lastButtonState1) {
    if (buttonPushCounter1 <= 0){
        if (valorPulsador1 == HIGH) {
            buttonPushCounter1++;
            buttonPushCounter2--;
            //Serial.println("Pulsaste EM+");
            digitalWrite(dirPinEmbr, LOW);
            for (int x = 0; x < steps * 1; x++) {
                digitalWrite(stepPinEmbr, HIGH);
                delayMicroseconds(stepDelay);
                digitalWrite(stepPinEmbr, LOW);
                delayMicroseconds(stepDelay);
            }
            /*Serial.println("EME1:");
            Serial.println(buttonPushCounter1);*/
        }
    }
}

}while(cond==1);
}

if (val == 69) {

```

Programacion_pedales

```
do{
valorPulsador1 = digitalRead(pulEmbrMas);
valorPulsador2 = digitalRead(pulEmbrMenos);
Serial.setTimeout(100);
const int steps = 450;
int stepDelay = 600;

if (valorPulsador2 != lastButtonState2) {
if (buttonPushCounter2 <= -1){
if (valorPulsador2 == HIGH) {
buttonPushCounter2++;
buttonPushCounter1--;
//Serial.println("Pulsaste EM-");
digitalWrite(dirPinEmbr, HIGH);
for (int x = 0; x < steps * 1; x++) {
digitalWrite(stepPinEmbr, HIGH);
delayMicroseconds(stepDelay);
digitalWrite(stepPinEmbr, LOW);
delayMicroseconds(stepDelay);
}
/*Serial.println("EMB2:");
Serial.println(buttonPushCounter2);*/
}
}
}

if (valorPulsador1 != lastButtonState1) {
if (buttonPushCounter1 <= 0){
if (valorPulsador1 == HIGH) {
buttonPushCounter1++;
buttonPushCounter2--;
//Serial.println("Pulsaste EM+");
digitalWrite(dirPinEmbr, LOW);
```

Programacion_pedales

```
for (int x = 0; x < steps * 1; x++) {
digitalWrite(stepPinEmbr, HIGH);
delayMicroseconds(stepDelay);
digitalWrite(stepPinEmbr, LOW);
delayMicroseconds(stepDelay);
}
/*Serial.println("EMB1:");
Serial.println(buttonPushCounter1);*/
}
}
}while(cond==1);
}

if (val == 70){

do{
valorPulsador1 = digitalRead(pulEmbrMas);
valorPulsador2 = digitalRead(pulEmbrMenos);
Serial.setTimeout(100);
const int steps = 450;
int stepDelay = 400;

if (valorPulsador2 != lastButtonState2) {
if (buttonPushCounter2 <= -1){
if (valorPulsador2 == HIGH) {
buttonPushCounter2++;
buttonPushCounter1--;
//Serial.println("Pulsaste EM-");
digitalWrite(dirPinEmbr, HIGH);
for (int x = 0; x < steps * 1; x++) {
```

Programacion_pedales

```
    digitalWrite(stepPinEmbr, HIGH);
    delayMicroseconds(stepDelay);
    digitalWrite(stepPinEmbr, LOW);
    delayMicroseconds(stepDelay);
}
/*Serial.println("EMB2:");
Serial.println(buttonPushCounter2);*/
}
}

if (valorPulsador1 != lastButtonState1) {
  if (buttonPushCounter1 <= 0) {
    if (valorPulsador1 == HIGH) {
      buttonPushCounter1++;
      buttonPushCounter2--;
      //Serial.println("Pulsaste EM+");
      digitalWrite(dirPinEmbr, LOW);
      for (int x = 0; x < steps * 1; x++) {
        digitalWrite(stepPinEmbr, HIGH);
        delayMicroseconds(stepDelay);
        digitalWrite(stepPinEmbr, LOW);
        delayMicroseconds(stepDelay);
      }
      /*Serial.println("EMB1:");
      Serial.println(buttonPushCounter1);*/
    }
  }
}

}while (cond==1);
}
```