

Instituto Tecnológico de Costa Rica

Carrera de Ingeniería Mecatrónica



Diseño de un sistema para la medición del tiempo desde el estado reposo profundo hasta la pantalla de inicio de una laptop de manera automatizada.

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura

Mario Alberto Araya Alvarado

Entrega: 03 de junio, 2022



Esta obra está bajo una licencia de
Creative Commons
ReconocimientoCompartirIgual 4.0
Internacional.

Declaración firmada del estudiante

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Costa Rica, 03/junio/2022



Mario Alberto Araya Alvarado
Céd: 117120132

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Mario Alberto Araya Alvarado

Proyecto: Diseño de un sistema para la medición del tiempo desde el estado reposo profundo hasta la pantalla de inicio de una laptop de manera automatizada.

PAULA ELENA	Firmado digitalmente por
MONGE CHANTO	PAULA ELENA MONGE
(FIRMA)	CHANTO (FIRMA)
	Fecha: 2022.06.10
	16:41:38 -06'00'

Ing. Paula Elena Monge Chanto

Asesora

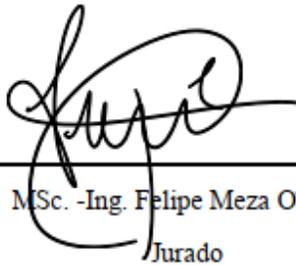
INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

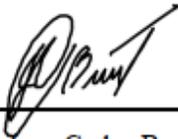
Estudiante: Mario Alberto Araya Alvarado

Proyecto: Diseño de un sistema para la medición del tiempo desde el estado reposo profundo hasta la pantalla de inicio de una laptop de manera automatizada.

Miembros del jurado evaluador



MSc. -Ing. Felipe Meza Obando
Jurado



MSc. -Ing. Juan Carlos Brenes Torres
Jurado



MSc. -Fis. Marta Vilchez Monge
Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 13 de junio 2022

Resumen

En el presente informe se desarrolla una propuesta de solución para medir el tiempo que le toma a una computadora cambiar del estado reposo profundo a inicio de pantalla, esto debido a que es un parámetro que Intel requiere para otorgar la certificación a las empresas que utilizan sus productos, además la medición también funciona como criterio para la toma de decisiones con respecto a futuras mejoras. El proceso actual no es aplicable en todos los dispositivos debido a que utiliza el wifi para generar el cambio de estado de la computadora, función que no poseen ciertas laptops en el modo reposo profundo.

Para llegar a encontrar la mejor solución posible se realiza una investigación previa, se analizan las necesidades y requerimientos que se requieren según las indicaciones del cliente, a partir de esto se realiza una generación de conceptos para definir el más apropiado.

Como resultado de este diseño se logra presentar una solución que cumple con las necesidades del cliente, fácil de replicar y a la vez validado, lo cual brinda una seguridad al momento de tomar decisiones basados en el valor de la medición.

Palabras clave: Cámara, Reposo profundo. Escritorio, Raspberry

Abstract

This report develops a solution proposal to measure the time it takes for a computer to change from Deep sleep to desktop ready because it is a parameter that Intel requires to grant certification to companies that use its products. In addition, measurement also functions as a criterion for decision-making regarding future improvements. The current process does not apply to all devices because it uses Wi-Fi to generate the change of state of the computer, a function that certain laptops do not have in Deep-sleep mode.

To find the best possible solution, a preliminary investigation is carried out, the needs and requirements that are required are analyzed according to the client's indications, from a generation of concepts is carried out to define the most appropriate.

As a result of this design, it is possible to present a solution that meets the customer's needs, is easy to replicate, and at the same time validated, which provides security when making decisions based on the value of the measurement.

Keywords: Camera, Deep sleep. Desktop ready, Raspberry

Dedicatoria

El presente trabajo es dedicado a mis padres y a mi hermana, quienes con su amor y esfuerzo me han permitido llegar a este punto, retándome y guiándome a mejorar cada día más como persona.

Así mismo, también lo quiero dedicar a mis tres mejores amigos, quienes con su compañía en los momentos felices como los no tanto han mostrado su apoyo incondicional, permitiendo poder disfrutar toda esta etapa y generar recuerdos que voy a llevar por siempre.

Agradecimiento

A pesar de la gran cantidad de personas con las cuales estoy agradecido por ser parte fundamental del proceso hasta este momento, quiero agradecer principalmente a Dios quien es el que ha colocado a todas ellas, quien ha guiado cada momento, me ha dado las capacidades y ha estado todo el proceso a mi lado. Sin Él nada hubiese sido posible.

Índice

Lista de figuras.....	iv
Lista de tablas	vii
Tabla de abreviaciones y términos.....	viii
1 Introducción	1
1.1 Antecedentes y contexto.....	1
1.2 Descripción del problema.....	4
1.3 Síntesis del problema	5
1.4 Objetivos	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos	5
1.5 Estructura del documento.....	6
1.6 Aporte de ingeniería.....	6
2 Marco teórico	7
2.1 Certificación Evo.....	7
2.2 Método conectado en reposo.....	9
2.3 Microprocesadores	10
2.3.1 Raspberry	11
2.3.2 NVIDIA Jetson Nano.....	11
2.3.3 Asus TinkerBoard	11
2.4 Lenguajes de programación	12
2.4.1 Python	12
2.4.2 C++	12
2.4.3 JAVA	13
2.5 Visión Artificial.....	13
2.6 Actuadores.....	14
2.6.1 Servo Motor	14
2.6.2 Motor a pasos	15
2.7 Herramientas de control remoto.....	16
2.7.1 TeamViewer.....	16

2.7.2	VNC Viewer	17
2.8	Diseño de experimentos	17
2.8.1	Estudio R&R del sistema de medición	17
3	Metodología	20
3.1	Determinación de necesidades	21
3.1.1	Entrevista e interpretación de necesidades	21
3.1.2	Jerarquización de necesidades	22
3.2	Determinación de métricas y especificaciones.....	24
3.3	Descomposición funcional del problema	25
3.3.1	Entradas y salidas.....	25
3.3.2	Definición de subsistemas.....	26
3.4	Generación de conceptos.....	27
3.4.1	Concepto 1	28
3.4.2	Concepto 2	29
3.4.3	Concepto 3	30
3.5	Evaluación de conceptos	31
3.5.1	Criterios de selección de conceptos	31
3.5.2	Evaluación de conceptos.....	33
3.6	Propuesta de concepto de solución.....	35
4	Propuesta de diseño.....	36
4.1	Selección del procesador	36
4.2	Configuración del procesador	38
4.2.1	Verificación de estados	41
4.2.2	Asignar tiempo de espera.....	48
4.2.3	Envío de señal al actuador	49
4.3	Selección de la fuente de alimentación del procesador.....	50
4.4	Selección del software de control.....	51
4.5	Selección de la cámara	53
4.6	Selección del servomotor	59
4.7	Diseño del soporte de la cámara.....	62
4.8	Diseño del soporte del servomotor.....	72
5	Resultados y análisis	81

5.1	Validación del sistema	81
5.1.1	Prueba Estructura mecánica	85
5.1.2	Prueba Verificación de estados	110
5.1.3	Prueba capacidad de medición del tiempo	115
5.1.4	Prueba verificación de la precisión de la medición y replicabilidad	120
5.1.5	Prueba de repetibilidad	123
5.2	Análisis financiero.....	125
6	Conclusiones y recomendaciones	128
6.1	Conclusiones	128
6.2	Recomendaciones.....	128
7	Referencias bibliográficas.....	130
A	Apéndices.....	135
A.1	Encuesta de jerarquización.....	135
A.2	Código utilizado.....	136
B	Anexos.....	145
B.1	Planos piezas mecánicas.....	145

Lista de figuras

Figura 1.1. Microprocesadores fabricados por la compañía componentes electrónicos Intel.	1
Figura 1.2. Ciertos parámetros que brinda la certificación EVO.....	2
Figura 2.1. Calcomanía que posee las laptops que cuentan con la certificación Evo.....	7
Figura 2.2. Requerimientos que poseen las laptops certificadas por Intel.....	8
Figura 2.3. Partes de un servomotor.	15
Figura 2.4. Motor a pasos.	16
Figura 2.5. Ejemplo estudio tipo1 mediante software Minitab.....	19
Figura 3.2. Diagrama de bloques de entradas y salidas.	25
Figura 3.3. Diagrama de subsistemas del sistema de control y monitorización	26
Figura 3.4. Plantilla para el proceso de combinación de los conceptos a evaluar.	28
Figura 3.5. Concepto 1.....	29
Figura 3.6. Concepto 2.....	30
Figura 3.7. Concepto 3.....	31
Figura 3.8. Diagrama del concepto de solución seleccionado.	35
Figura 4.1. Diagrama de flujo del proceso de medición del tiempo.	40
Figura 4.2. Imagen utilizada como prueba para demostración del proceso de verificación de estado de la computadora (modo encendido).....	41
Figura 4.3. Imagen utilizada como prueba para demostración del proceso de verificación de estado de la computadora (modo reposo).....	42
Figura 4.4. Imagen de ejemplo convertida en blanco y negro.	43
Figura 4.5. Imagen de ejemplo con sus bordes extraídos.	44
Figura 4.6. Imagen de ejemplo luego de ser dilatada.	45
Figura 4.7. Pantalla extraída de la imagen original (modo encendido).	46
Figura 4.8. Pantalla extraída de la imagen original (modo reposo).	46
Figura 4.9. Histograma de la imagen original (modo encendido).	47
Figura 4.10. Histograma de la imagen original (modo reposo).	48
Figura 4.11. Interfaz mostrada al usuario para la asignación del tiempo de espera.	49
Figura 4.12. Fuente de alimentación seleccionado.	51
Figura 4.13. Módulo de cámara raspberry seleccionado.	54
Figura 4.14. Cuadro inicial tomado por cámara con una configuración de 30fps.....	55
Figura 4.15. Cuadro número 33 tomado por cámara con una configuración de 30fps.....	56
Figura 4.16. Cuadro número 38 tomado por cámara con una configuración de 30fps.....	57
Figura 4.17. Cuadro número 44 tomado por cámara con una configuración de 30fps.....	58
Figura 4.18. Servomotor modelo MG90.....	61
Figura 4.20. Esquema ajuste por deslizamiento ordinario.....	65
Figura 4.21. Base del soporte para la cámara	66
Figura 4.22. Unión entre la base del soporte y el acople para la cámara.....	67
Figura 4.23. Acople para la cámara	68
Figura 4.24. Vista isométrica de la estructura completa del soporte de la cámara	69
Figura 4.25. Dimensiones del módulo de la cámara Raspberry Pi	70
Figura 4.26. Soporte para servomotor.....	73

Figura 4.28. Dimensiones del Motor MG90.....	74
Figura 4.28. Eje de sujeción para el soporte del Servo Motor.....	75
Figura 4.29. Soporte para el eje de sujeción.....	76
Figura 4.30. Esquema ajuste forzado ligero.....	77
Figura 4.31. Ensamble soporte servomotor.....	77
Figura 4.32. Punta para acople del servomotor.....	78
Figura 4.33. Tabla de calidades IT.....	79
Figura 4.34. Proceso de mecanizado según el IT.....	80
Figura 5.1. Prototipo utilizado para realizar las mediciones de validación y confirmar funcionamiento de los distintos sistemas integrados.....	83
Figura 5.1. Dimensiones generales del brazo medio en mm.....	86
Figura 5.2. Dimensiones generales de la pieza final del soporte en mm. Fuente propia.....	87
Figura 5.3. Dimensiones generales del soporte del Servomotor en mm.....	87
Figura 5.4. Dimensiones generales del eje de la estructura del servomotor.....	88
Figura 5.5. Selección del material en el software inventor.....	89
Figura 5.6. Asignación de la restricción de movimiento.....	90
Figura 5.7. Asignación de la carga debido al peso de la cámara.....	91
Figura 5.8. Asignación de la carga debido al peso de la propia pieza de la estructura.....	92
Figura 5.9. Factor de seguridad obtenido del análisis estático mediante software Inventor de la pieza que soporta la cámara.....	93
Figura 5.10. Asignación del material a la parte media de la estructura.....	94
Figura 5.11. Asignación de las restricciones de movimiento a la parte media de la estructura.....	95
Figura 5.12. Asignación de la primera carga correspondiente al peso de la cámara y al de la pieza acoplada.....	96
Figura 5.13. Asignación de la carga correspondiente al peso de la propia pieza.....	97
Figura 5.14. Resultados de la simulación del análisis estático por medio del software Inventor de la parte media de la estructura.....	98
Figura 5.15. Asignación del material a la base de la estructura.....	99
Figura 5.16. Asignación de las restricciones de movimiento a la base de la estructura.....	100
Figura 5.17. Asignación de la carga a la base de la estructura.....	101
Figura 5.18. Resultados de la simulación del análisis estático por medio del software Inventor de la base de la estructura.....	102
Figura 5.19. Asignación del material al eje del soporte del servomotor.....	103
Figura 5.20. Asignación de las restricciones de movimiento al eje del soporte del servomotor.....	103
Figura 5.21. Asignación de la carga al eje del soporte del servomotor.....	104
Figura 5.22. Resultados de la simulación del análisis estático por medio del software Inventor de al eje del soporte del servomotor para el cado de la deflexión.....	105
Figura 5.23. Resultados de la simulación del análisis estático por medio del software Inventor al eje del soporte del servomotor correspondiente al factor de seguridad.....	106
Figura 5.24. Asignación del material a la base del soporte del servomotor.....	106
Figura 5.25. Asignación de las restricciones de movimiento la base del soporte del servomotor.....	107
Figura 5.26. Asignación de la carga a la base del soporte del servomotor.....	108

Figura 5.27. Resultados de la simulación del análisis estático por medio del software Inventor de la base del soporte del servomotor.....	109
Figura 5.28. Estado modo reposo, tomada desde la cámara del sistema diseñado.	111
Figura 5.29. Estado modo encendido, tomada desde la cámara del sistema diseñado.	112
Figura 5.30. Mensaje que despliega el sistema cuando detecta la computadora en el estado de reposo.....	112
Figura 5.31. Mensaje que despliega el sistema cuando detecta la computadora encendida.	113
Figura 5.32. Pantalla que se despliega al inicializar el código del sistema diseñado.	116
Figura 5.32. Selección del tiempo a esperar para enviar la señal de activación al servomotor. .	117
Figura 5.33. Mensajes desplegados en la consola al enviar señales de activación.	117
Figura 5.34. Resultado desplegado por el programa luego de realizar la comparación de imágenes.	118
Figura 5.35. Parámetros utilizados para realizar el estudio tipo 1 mediante minitab.	121
Figura 5.36. Opciones escogidas para realizar el estudio tipo 1 mediante minitab.	121
Figura 5.37. Resultados del estudio tipo 1 de las medidas tomas por el instrumento diseñado.	122
Figura 5.38. Cálculo de costos y beneficios totales del proyecto.	126
Figura B2.1. Código utilizado para el funcionamiento del sistema diseñado (1/).	136
Figura B2.2. Código utilizado para el funcionamiento del sistema diseñado (2/10).	137
Figura B2.3. Código utilizado para el funcionamiento del sistema diseñado (3/10).	138
Figura B2.4. Código utilizado para el funcionamiento del sistema diseñado (4/10).	139
Figura B2.5. Código utilizado para el funcionamiento del sistema diseñado (5/10).	139
Figura B2.6. Código utilizado para el funcionamiento del sistema diseñado (6/10).	140
Figura B2.7. Código utilizado para el funcionamiento del sistema diseñado (7/10).	141
Figura B2.8. Código utilizado para el funcionamiento del sistema diseñado (8/10).	142
Figura B2.9. Código utilizado para el funcionamiento del sistema diseñado (9/10).	143
Figura B2.10. Código utilizado para el funcionamiento del sistema diseñado (10/10).	144

Lista de tablas

Tabla 2.1. Requerimientos necesarios para obtener la certificación Evo. Fuente: [3].....	8
Tabla 2.2. Estados de energía una computadora con sistema operativo Windows. Fuente: [8]...	10
Tabla 2.3. Procesos de la visión artificial. Fuente: [21].....	13
Tabla 3.1. Interpretación de necesidades a partir de la entrevista con el cliente	22
Tabla 3.2 Enumeración de las necesidades. Elaboración propia.	23
Tabla 3.3 Jerarquización de las necesidades basado en las indicaciones del cliente. Elaboración propia	24
Tabla 3.4. Métricas definidas para la ejecución del proyecto. Elaboración propia.	24
Tabla 3.5. Valores marginales e ideales para las métricas definidas. Elaboración propia.	25
Tabla 3.6. Criterios para la selección de conceptos. Elaboración propia.	32
Tabla 3.7. Calificación de cumplimiento de criterios y evaluación de conceptos. Elaboración propia	34
Tabla 4.1. Características de los modelos raspberry pi más comunes [27].	37
Tabla 4.2. Matriz de selección del microprocesador. Fuente propia.	37
Tabla 4.3. Matriz de selección del método del control remoto. Fuente propia.....	52
Tabla 4.4. Propiedades del módulo de cámara v1 para Raspberry pi. [34]	54
Tabla 4.5. Características de los servomotores más comunes.	59
Tabla 4.6. Matriz de selección del servomotor. Fuente propia.	60
Tabla 4.7. Características de materiales comunes utilizados para impresión 3D. [39]	62
Tabla 4.8. Matriz de selección del material. Fuente propia.	63
Tabla 4.9. Características de materiales comunes utilizados para manufactura. Fuente propia...	70
Tabla 4.10. Matriz de selección del material. Fuente propia.	71
Tabla 5.2. Cálculo de la desviación estándar según la cantidad de medidas realizadas. Fuente propia	84
Tabla 5.3. Resumen de resultados obtenidos en la simulación de estrés estático. Fuente propia	110
Tabla 5.4. Pruebas y resultados de la capacidad de identificación de estados de la herramienta diseñada. Fuente propia	114
Tabla 5.5. Resumen de resultados obtenidos en la prueba de identificación del estado de la computadora por medio de la herramienta diseñada. Fuente propia	115
Tabla 5.6. Mediciones de tiempo obtenidas con el sistema diseñado. Fuente propia.....	119
Tabla 5.7. Valor promedio obtenido de las 30 mediciones realizadas. Fuente propia	120
Tabla 5.7. Mediciones de tiempo aceptadas para la prueba de repetibilidad. Fuente propia.....	124
Tabla 5.8. Cuadro resumen de los valores aceptados dentro del rango estipulado. Fuente propia	125
Tabla 5.9. Cálculo del costo del proyecto. Fuente propia.....	126
Tabla 5.10. Resumen del análisis económico. Fuente propia	127
Tabla B.1 Encuesta para la jerarquización de necesidades.....	135

Tabla de abreviaciones y términos

Abreviación	Término	Significado
CCG	Client Computing group	Equipo dentro de Intel
OEM	Original Equipment manufacturer	Diseñadores con productos de una segunda empresa
PROFINET	Process Field Network	Red de Campo de Proceso: protocolo de red
SoC	System on Chip	Chip que integra todas o la mayor parte de componentes necesarios para el funcionamiento de un ordenador

1 Introducción

1.1 Antecedentes y contexto

Intel fue fundada en 1968 en Santa Clara, ahora cuenta con distintas sedes alrededor del mundo. Intel diseña, fabrica y vende circuitos integrados para la industria de la computación y las comunicaciones a nivel mundial. Los principales productos incluyen microprocesadores, chipsets y placas madre [1] como el que observa en la Figura 1.1. El proyecto se realizará específicamente en la sede de Intel ubicada en Costa Rica (Intel Corporation).

Intel Costa Rica abrió sus operaciones en el año 1997, muchas de las labores que se realizan corresponden a las de diseño, validación e integración de circuitos, soluciones a partir de software, ventas y equipo de marketing. Esta empresa representa el 60% de las exportaciones de investigación y desarrollo del país. [5]



Figura 1.1. Microprocesadores fabricados por la compañía componentes electrónicos Intel. [1]

Dentro de Componentes Intel Costa Rica existe un grupo llamado CCG (Client Computing Group), el cual es el equipo dentro de Intel dedicado a diseñar, producir y proveer los componentes, parámetros y especificaciones de diseño recomendados a los socios de la empresa que se dedican a fabricar computadoras con tecnologías realizadas por Intel. Esto debido a que uno de los retos que posee Intel es ofrecer productos de calidad y competitivos en un mercado donde los

competidores poseen un ecosistema cerrado, es decir, fabrican sus propios componentes. Por otra parte, Intel labora con ecosistemas abiertos por lo que requiere establecer los parámetros necesarios para ser competitivos. El problema que se pretende solventar será uno específicamente dentro de este grupo mencionado.

El grupo de CCG, como se mencionó, es el encargado de definir los parámetros de diseño que deben realizar las OEMs “Original Equipment manufacturer”, los cuales son los que se dedican a fabricar productos o componentes que son adquiridos por una segunda empresa y vendidos con la marca de la segunda empresa, estos parámetros son utilizados como requisitos para contar con la certificación de calidad que proporciona Intel, la cual posee el nombre de Evo [3], esta certificación tiene como propósito brindar datos cuantificables sobre la experiencia de usuario, brindando valores de referencia en distintos parámetros como la duración de batería, velocidad de respuesta y otros que se pueden observar en la Figura 1.2.



Figura 1.2. Ciertos parámetros que brinda la certificación EVO. [3]

Por último, tienen la responsabilidad de definir las mejoras que se desean realizar en los futuros productos que se van a ofrecer en el mercado, siendo el paso intermedio entre las propuestas que ofrece el equipo de marketing y los equipos de ingeniería que van a trabajar directamente con los componentes para lograr alcanzar las especificaciones requeridas.

De los retos que posee Intel es ofrecer productos de calidad, competitivos en el mercado, un mercado donde sus principales retos es la existencia de competidores que poseen un ecosistema cerrado, es decir, fabrican y utilizan sus propios componentes. Por lo tanto, como Intel labora con ecosistemas abiertos, para lograr competir con estos productos se proporciona a los OEMs parámetros diseños que deben alcanzar, como una duración de batería de más de 9 horas o un tiempo de carga de inicio de pantalla menor a un segundo.

Históricamente para poder definir un criterio sobre cual computadora es mejor en el mercado, se medía el rendimiento por medio de pruebas que sirven como puntos de referencia, entre ellas la especificación CPU, el cual es uno muy utilizado, el problema con esas metodologías es que ocurre lo que se conoce como puntos de referencia sintéticos, esto quiere decir que da un parámetro que no es importante en el ámbito de la experiencia de usuario, por ejemplo, se puede obtener la cantidad de operaciones con punto flotante que puede realizar en un segundo, lo que podría ser importante para una persona con conocimiento técnico pero el cual resulta en un dato que no posee una relevancia para un usuario que utiliza la computadora para entretenimiento.

Por este motivo Intel ha estado desarrollando un software propio que cuantifica la experiencia de usuario. Para lograrlo miden distintas acciones como el tiempo de carga de la pantalla inicial. Aquí se presenta un reto, ya que no se pretende enviar un procedimiento de medición a los OEMs que desean certificar sus productos, debido a que no existe manera de verificación de si realizaron el proceso de la manera correcta, lo que puede provocar resultados que no son consistentes debido a la falta de una metodología robusta.

Por lo tanto, se busca una manera de realizar esa medición de manera consistente, automatizada y que se necesita la menor cantidad posible de la interacción de un técnico para reducir el error de usuario. La solución que se posee actualmente es un proceso que se llama “Conectado en reposo” con el cual se lleva la computadora a modo de reposo y se activa por medio del envío de un paquete de datos por su interfaz wifi, lo cual provoca otra problemática, la computadora nunca llega al estado de reposo profundo debido a que en ese estado no se podría hacer uso del wifi y no podría enviar el paquete de datos, entonces el tiempo que se ha estado midiendo no es realmente el que se desea.

Otras deficiencias existentes en el método actual son la imprecisión con la que se realiza la medición y la imposibilidad de realizar este método en ciertas computadoras que no poseen la implementación del estado “C3” un estado de reposo que mantiene la función del wifi activa [4], ya que sin el interfaz wifi no es posible enviar el paquete de datos y por la tanto, no es posible provocar el cambio de estado.

Para poder ofrecer la certificación EVO sobre el parámetro de tiempo de carga de la pantalla de inicio es necesario calibrar los datos obtenidos por el método anterior, y para esto se necesita un método más preciso, que permita medir el tiempo entre el estado de reposo profundo y el inicio de pantalla completamente cargado, para así poseer un valor de referencia que permita comparar los métodos de medición y obtener un mayor criterio en la toma decisiones.

Este valor de referencia además de utilizarse para decidir si se brinda la certificación, también permite indicar a las OEMs un valor esperado de tiempo de carga de inicio de pantalla con un criterio lógico y respaldado de manera confiable, también influye en la decisión de propuestas sobre las mejoras de parámetros que se espera que posean futuros componentes, por lo tanto, el contar con un criterio sólido, permite planificar las especificaciones que los ingenieros dentro de Intel deben lograr en sus diseños y justificar de manera razonable al equipo de marketing el porqué de las decisiones tomadas, como lo sería el porcentaje de mejora que se espera en un futuro producto.

1.2 Descripción del problema

No existe un valor de referencia que sea preciso ni una metodología confiable de medición sobre el tiempo que transcurre entre el cambio del estado de reposo profundo hasta que el inicio de pantalla de una computadora se haya cargado completamente, que permita poder definir estándares de calidad, proveer especificaciones de diseño y proyecciones de futuros componentes que se deseen fabricar.

La razón por la que se presentan estos problemas mencionados es debido a que las metodologías utilizadas en la actualidad no cumplen con ciertos estándares necesarios, como una manera de confirmar la replicabilidad de la prueba. Además, la prueba actual no se realiza en las

condiciones deseadas ya que el computador no se lleva al estado reposo profundo, el cual es el requerido.

El cálculo del tiempo en que tarda este proceso se realiza con herramientas como el sistema de estados de poder que posee Windows o grabando un video de manera empírica y analizando cuadro por cuadro, lo que requiere mucho tiempo del personal o de un conocimiento técnico para utilizar las herramientas mencionadas.

1.3 Síntesis del problema

No existe un valor de referencia sobre el tiempo de carga de la pantalla de inicio lo suficientemente preciso, realizado en las condiciones deseadas y a la vez que sea medido con una metodología que brinde la confiabilidad suficiente para poseer un valor que pueda ser utilizado para permitir tomar decisiones de relevancia como el diseño, el establecimiento de parámetros, la calidad de experiencia de usuario y la aprobación de certificaciones, así tanto como para los clientes que posee Intel como para los procesos que se realizan dentro de la compañía

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un sistema capaz de interactuar con la computadora y que posea la habilidad de medir el tiempo que dura la laptop en hacer la transición del estado reposo profundo hasta que la pantalla de inicio se haya cargado completamente, esto de manera automatizada.

1.4.2 Objetivos específicos

- 1 Diagnosticar las etapas necesarias que el sistema ocupa para cumplir con todos los requerimientos y ser capaz de realizar la medición de tiempo en las condiciones deseadas.
- 2 Diseñar el sistema capaz de detectar el cambio en el computador entre el estado de reposo profundo y la pantalla de inicio, además del envío y recepción de señales.
- 3 Diseñar los componentes mecánicos que compondrán el sistema.
- 4 Validar la integración de los sistemas diseñados mediante pruebas de concepto.

1.5 Estructura del documento

El presente documento se estructura según los siguientes apartados:

- Marco teórico: En este apartado se muestran las bases teóricas del proyecto, mostrando las fórmulas y diagramas utilizados en el diseño de la solución.
- Metodología: Esta sección trata con el proceso de diseño análogo al mostrado en el libro “Diseño y Desarrollo de Productos” de Ulrich y Eppinger para así generar una propuesta de solución que solvete las necesidades del proyecto.
- Propuesta de diseño: En esta sección se genera el diseño de detalle de los diferentes aspectos establecidos del proyecto, basándose así en conocimiento técnico para abordar los aspectos de la solución propuesta en la etapa de metodología.
- Resultados y análisis: En esta etapa se muestra los resultados obtenidos de las distintas pruebas para validación de la idea propuesta.
- Conclusiones y Recomendaciones: Se concluye extrayendo la información más importante de manera resumida e indicando mejoras que se le podrían hacer en un futuro.

1.6 Aporte de ingeniería

El aporte de ingeniería que pretende otorgar este informe recae en el desarrollo y explicación de una solución que considera diferentes componentes de la Ingeniería Mecatrónica y las áreas que la conforman. Así generando un diseño íntegro, basado en criterio técnico para llegar a una propuesta funcional que tome en cuenta las características electrónicas, software y mecánicas.

2 Marco teórico

2.1 Certificación Evo

La plataforma Evo fue anunciada por primera vez en el año 2020, a la vez que fue anunciado la 11 generación de chips para laptops. Esta plataforma funciona como una nueva categoría de laptops de gama alta. Es una certificación difícil de lograr para los fabricantes debido a que los requisitos para ser aceptada son estrictos. Se puede conocer si una Laptop ha sido certificada si posee una calcomanía como la que se observa en la Figura 2.1. [5]



Figura 2.1. Calcomanía que posee las laptops que cuentan con la certificación Evo. [6]

Como primer requisito la laptop debe utilizar un procesador con la estructura llamada “Alder Lake” creada por Intel, además deben ofrecer al menos 9 horas de uso con una sola carga, así como un peso ligero, menor a 1.5 Kg [5] entre otros requisitos los cuales se pueden observar en la Figura 2.2 pero además se mencionan también en la Tabla 2.1

The Intel® Evo™ Platform

Project Athena Second Edition Specification Highlights

Instant Wake

- ✓ Adaptive Connected Standby / Lucid Sleep
- ✓ Biometric login: fingerprint sensor, face recognition or Bluetooth phone proximity

Incredible Performance and Responsiveness

- ✓ 11th Gen Intel® Core™ i5 or i7 processors with Intel® Iris® X* graphics
- ✓ Windows 10 or Chrome OS
- ✓ Intel® Adaptix technology
- ✓ 128b, ≥8GB Dual Channel Memory and ≥256GB PCIe/NVMe SSD

Intelligence Built Across Platform Levels

- ✓ Intel® DL Boost+ Intel® GNA 2.0
- ✓ Far-field voice services
- ✓ OpenVINO™ AI on PC; Win ML support
- ✓ Optional adaptive usages including Wake-on-Approach, Gaze-Based Dimming and Walk-Away Lock



Best of 11th Gen Intel® Core™ Processors with Intel® Iris® X* Graphics Based Laptops

Co-engineered Designs

Verified to Deliver Responsiveness on Battery

Real World Battery Life

Instant Wake

Fast Charging

FOCUS. ALWAYS READY. ADAPTIVE.

Battery Life For The Real World

- ✓ Fast charging capabilities over USB Type C
- ✓ Integration and optimization of low-power components and co-engineering support for power efficiency

Lightning-Fast Connectivity

- ✓ Fast and persistent connection with Intel® Wi-Fi 6 (Gig+)
- ✓ Universal cable connectivity over USB Type C with Thunderbolt™ 4
- ✓ Optional Intel® 5G LTE or 5G

Innovative and Engaging Form Factor

- ✓ >12"-15.6" ≥ FHD touch display, precision touchpads and more
- ✓ ≤15mm fanless / fanless designs
- ✓ Sleek, thin-and-light and 2-in-1 designs with narrow bezels for a more immersive experience
- ✓ Immersive audio with premium high-fidelity audio codec/speaker tuning
- ✓ User-facing camera ≥ HD/720p @ 30 fps

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. For more information visit intel.com/Evo.

Figura 2.2. Requerimientos que poseen las laptops certificadas por Intel. [6]

Tabla 2.1. Requerimientos necesarios para obtener la certificación Evo. Fuente: [3].

Requerimiento	Descripción
Más desempeño	Los procesadores Intel Core de 11 generación y los gráficos Intel Iris X ofrecen edición de fotografías y video 3 veces más rápida.
Más libertad	Tiene que poder contar con más de 9 horas de autonomía en pantallas FHD.
Más flujo	Debe iniciarse en menos de un segundo y tener una capacidad de respuesta un 40% mayor que una portátil con 2 años de antigüedad.
Más conexiones	Internet casi 3 veces más rápido con la tecnología Intel Wifi 6 incorporada, en comparación con el WiFi estándar.
Más inmersión	Los gráficos Intel Iris X y los biselados ultradelgados permite que se pueda disfrutar mejor la pantalla.
Más simplicidad	Ofrecer un ancho de banda de 40 Gb/s para datos y video a través de una conexión con un solo cable

El requerimiento resaltado en la tabla 2.1 es el de mayor importancia para el proyecto, ya que la herramienta a diseñar es realizada con el propósito de poder medir ese tiempo de respuesta.

2.2 Método conectado en reposo

“Conectado en reposo” es el método utilizado actualmente para calcular el tiempo de recuperación de la computadora. El proceso de medición es llevado a cabo de la siguiente manera: para realizar la transición de manera automática entre el estado de reposo profundo a la pantalla de inicio crean una tarea agendada, para esto se necesitan varias configuraciones y especificar una acción en la tarea, usualmente se le solicita solamente abrir el CMD, asegurarse que los temporizadores de activación de la computadora se encuentran habilitados y de esta manera es como logran conocer en el momento que inicia la transición de estados en la portátil. [7]

Luego, para conocer cuando se alcanzó el inicio de pantalla cargado completamente se realiza por medio de dos métodos, el primero consiste en tomar un video y luego analizarlo cuadro por cuadro y de esta manera poseer un aproximado de la hora en que la computadora alcanzó el estado esperado. La segunda, el cual depende de si el sistema operativo lo permite, es el de conocer el momento en que alcanzó el estado S0 o de trabajo [8] dichos estados se explican en la Tabla 2.2.

Es importante aclarar que entre las innovaciones que se desarrollan se encuentra la implementación del cálculo del tiempo de manera automatizada sin la necesidad de un técnico que analice los cuadros del video o haga uso de la herramienta de sistemas de estados que posee Windows y de esta manera poder llevar a cabo las mediciones sin el uso de la interfaz WI FI.

Tabla 2.2. Estados de energía una computadora con sistema operativo Windows. Fuente: [8].

Estado de energía	Descripción
S0	El sistema está funcionando a pleno rendimiento. Los componentes de hardware que no estén en uso pueden entrar en el modo de ahorro de energía
S0 baja energía	Los sistemas que tengan un SoC pueden entrar en este estado, En este estado el sistema puede cambiar rápidamente desde el estado de ahorro de energía al estado activo
S1, S2 y S3	En estos estados la memoria volátil (RAM) se mantiene activa, y los periféricos como el teclado se mantienen en modo de ahorro de energía USB si lo soportan. Como la RAM está activa, "despertar" de este estado es bastante rápido.
S4	El sistema parece estar apagado. El consumo de energía se reduce a la mínima expresión. El sistema guarda el contenido de la memoria RAM en un archivo de hibernación para preservar el estado del sistema. Algunos componentes como el teclado, la tarjeta de red u otros dispositivos USB permanecen activos para poder "despertar" el sistema.
S5	El Sistema está apagado.

2.3 Microprocesadores

Para el diseño de la herramienta es necesario un dispositivo que funcione como el cerebro del instrumento, el cual es el encargado de enviar las señales de activación al actuador, poseer los temporizadores necesarios para el control del proceso y que posee el código de funcionamiento.

Para cumplir los requerimientos mencionados se suele utilizar microprocesadores, por este motivo se investiga las características generales que poseen estos dispositivos y se mencionan 3, los cuales son comunes y pueden ser hallados en el mercado actual.

Un microprocesador es un componente de integración en gran escala, controlado por un código de programación, capaz de ejecutar operaciones aritméticas y lógicas sobre datos. La forma en la que se utiliza un microprocesador es llevando a cabo un sistema donde las funciones lógicas no son realizadas mediante interconexión de una serie de circuitos, sino mediante una serie de instrucciones, que conforman el programa, almacenadas en una memoria.[12]

A estos procesadores se les suele colocar una memoria RAM, entradas y salidas en un solo circuito. Cuando se realiza este cambio al dispositivo se le llama placa computadora, existen muchos tipos de placas, se mencionan algunas de las más conocidas.

2.3.1 Raspberry

La Raspberry es de los microprocesadores más utilizados ya que se puede encontrar fácilmente en el mercado y cuenta con mucha variedad entre sus productos. Es posible utilizar lenguajes de alto nivel reconocidos para hacer uso de ellos, tales como Python y C++, por lo cual es posible hallar documentación con facilidad.

Aunque existen muchos variades de este producto lo más usual es que utilice un procesador de tipo ARM1 el cual tiene una velocidad de 700 MHz, dato importante por la naturaleza del proyecto donde se necesita realizar mediciones en menos de un segundo.

Se pueden cargar sistemas operativos desde Mac, Windows y Linux lo que podría ser muy útil por temas de compatibilidad o facilidad de uso dependiendo del sistema que se desee utilizar o del que haya necesidad de escoger. [13]

2.3.2 NVIDIA Jetson Nano

Este microprocesador permite la ejecución de múltiples redes neuronales en paralelo para aplicaciones tales como clasificación de imágenes, detección de objetos, segmentación y procesamiento de voz [14], por lo cual este podría ser una buena opción si se decide utilizar la inteligencia artificial como método para la solución del proyecto. Además, posee una velocidad de 1600 MHz.

2.3.3 Asus TinkerBoard

Se ha incluido esta opción debido a que esta placa funciona con frecuencias de hasta 1.8 GHz [15], posiblemente de las placas más veloces que se pueda encontrar en el mercado actualmente, es por este motivo que se analiza y se busca información al respecto ya que la velocidad de funcionamiento es un factor decisivo para este proyecto.

Esta placa permite utilizar Python y C como lenguajes de programación, lenguajes de alto nivel y de uso común.

2.4 Lenguajes de programación

Como se mencionó es necesario tener un dispositivo el cual sea el cerebro y lleve control de todo el proceso que sea necesario, pero el proceso es definido por medio de un código y para realizar estas líneas de instrucciones es necesario utilizar un lenguaje de programación. Existen muchos tipos de lenguajes, en este documento se mencionan tres los cuales son muy conocidos, cuya documentación es fácil de hallar y además poseen compatibilidad en la mayoría de los microprocesadores.

2.4.1 Python

Este lenguaje es tomado en cuenta ya que es de los más utilizados y comunes, además posee librerías que permite el uso de Visión e Inteligencia artificial los cuales son dos métodos posibles de solución para el problema de la identificación del estado de la computadora.

Python cuenta con facilidades para la programación orientada a objetos. Es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar tareas complejas en pocas líneas de código y de manera sencilla.

A través de los años Python ha contado con distintas versiones, como por ejemplo la 2.0. Pero una de las ventajas que posee este lenguaje es que las versiones han sido compatibilizadas entre si [16] lo que implica que, aunque se utilice una versión no tan reciente de Python por limitaciones que pueda poseer el procesador, de igual manera se puede consultar documentación reciente.

2.4.2 C++

C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Este lenguaje funciona para realizar programación orientada a objetos. [18]

Es un lenguaje de más bajo nivel que Python por lo tanto su tiempo de compilación es menor.

2.4.3 JAVA

La mayoría de los microprocesadores son compatibles con este lenguaje, además posee las librerías necesarias para utilizar visión artificial e Inteligencia al igual que Python. La velocidad de compilación es menor a C y Python lo que podría llegar a influenciar en los procesamientos necesarios para realizar las mediciones.

2.5 Visión Artificial

Una de las propuestas para la solución al problema de identificar el estado del computador es por medio del uso de la visión artificial. Por este motivo resulta importante comprender en que consiste y cuáles son los pasos a comunes a desarrollar para encontrar una solución por este medio.

La visión artificial o visión por computador intenta emular la capacidad de algunos seres vivos para ver una escena (imagen), entenderla y actuar en consecuencia. Existe un crecimiento en el número y tipo de aplicaciones industriales que demandan el uso de técnicas de visión artificial. El continuo desarrollo de nuevos algoritmos y aplicaciones hacen de esta disciplina una tecnología en constante evolución y ha experimentado un rápido avance en las últimas décadas.

Otra manera de definirlo es como: Procesos de obtención, caracterización e interpretación de información de imágenes tomadas desde un mundo tridimensional a partir de imágenes bidimensionales. Estos procesos pueden ser subdivididos en seis procesos como se muestra en la Tabla 2.3. [21]

Tabla 2.3. Procesos de la visión artificial. Fuente: [21].

Proceso	Salida
Captura	Imagen.
Preprocesamiento	Imagen.
Segmentación	Grupo de pixeles (objetos o regiones).
Descripción	Información cuantitativa de los objetos o regiones.
Reconocimiento (clasificación)	Objetos clasificados en categorías
Interpretación	Compresión de la escena

2.6 Actuadores

Una de las innovaciones en la herramienta desarrollada es la capacidad de realizar la transición de estados sin el uso de la interfaz WIFI, método utilizado en la actualidad. Se pretende realizar este cambio de estados por medio de la interacción física con el computador presionando un botón del teclado, pero como es necesario realizar este proceso de manera automática, es decir, sin que se requiera de un técnico que presione el botón, es fundamental diseñar un mecanismo que realice esta acción. Por lo tanto, se vuelve indispensable el uso de un dispositivo externo que genere movimiento y por eso se concluye que es necesario el uso de un actuador.

Un actuador es un dispositivo con la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía. Por lo general se clasifican según la energía que utilicen o el según el movimiento que generan. [23] A continuación se presentan dos actuadores utilizados muy frecuentemente y de bajo costo.

2.6.1 Servo Motor

Este tipo de actuador es muy utilizado para transformar energía en movimiento, debido a su bajo costo y facilidad de uso.

Un servomotor es un actuador rotativo o motor que permite un control preciso en términos de posición angular, aceleración y velocidad, capacidades que un motor normal no tiene, lo que permite el control por medio de un procesador o circuito.

Los servomotores se controlan enviando un pulso eléctrico de ancho variable, o modulación de ancho de pulso (PWM), a través del cable de control. Existe un pulso mínimo, un pulso máximo y una frecuencia de repetición. Por lo general, un servomotor sólo puede girar 90° en cualquier dirección para un movimiento total de 180° [24]. En la Figura 2.3 se puede observar un ejemplo de un servomotor.

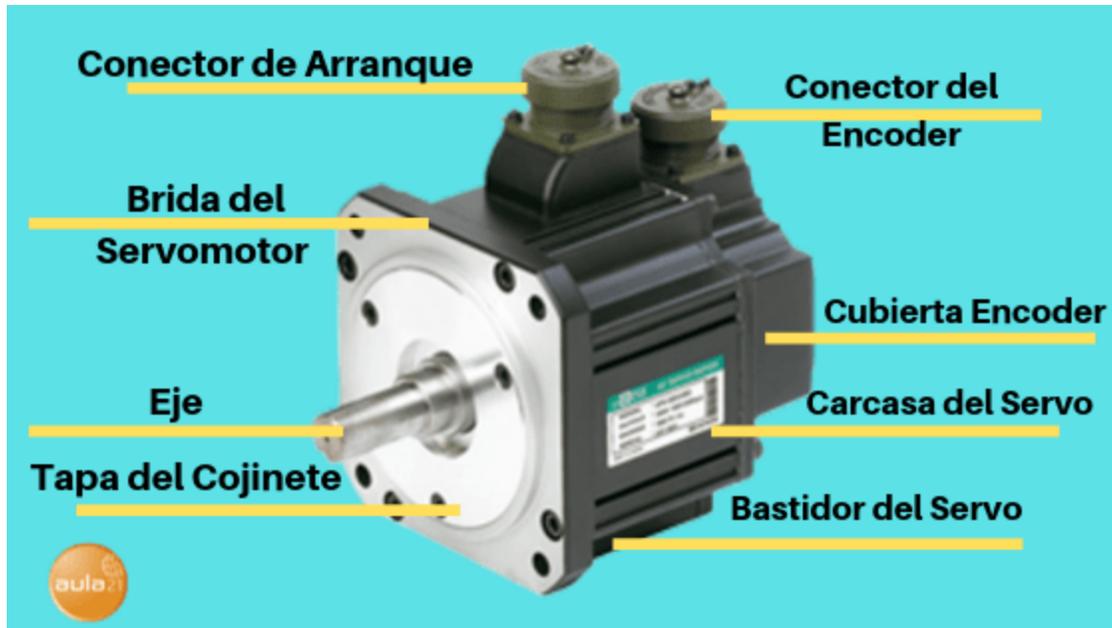


Figura 2.3. Partes de un servomotor. [24]

2.6.2 Motor a pasos

Al igual que el actuador mencionado anteriormente el motor a pasos es muy utilizado, por lo tanto, resulta sencillo encontrarlo en el mercado, se utiliza especialmente cuando se necesita mayor precisión que el servomotor con respecto a posición y también es utilizado en operaciones donde se requieren mayor torque.

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 (paso o medio paso) dependiendo de sus entradas de control. Esto permite tener un mayor control de la posición que la que se posee con el servomotor, pero a cambio se pierde velocidad de rotación, aunque se aumenta el torque.

El principio de funcionamiento está basado en un estator construido por varios bobinados en un material ferromagnético y un rotor que puede girar libremente en el estator. Estos diferentes bobinados son alimentados uno a continuación del otro y causan un determinado desplazamiento angular que se denomina “paso angular” y es la principal característica del motor. [25] En la Figura 2.4 se muestra un motor a pasos.

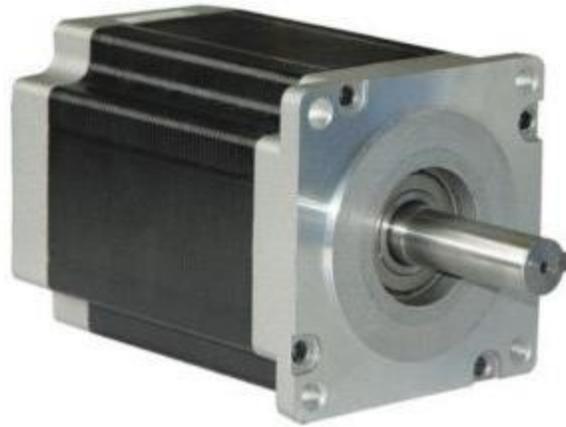


Figura 2.4. Motor a pasos. [25]

2.7 Herramientas de control remoto

Una de las mejoras implementadas en la nueva herramienta de medición es disminuir la complejidad de uso, por eso se agrega un software de control remoto que permite controlar el dispositivo diseñado desde la computadora personal sin la necesidad de instalar periféricos, también con este software se elimina la necesidad de estar a una distancia cercana del instrumento de medición para realizar las tomas de tiempo.

Un software de control remoto es una herramienta que permite el control e interacción de otro dispositivo sin la necesidad de utilizar una conexión por medio de cables, para el caso del proyecto se utiliza para manejar el microprocesador sin el uso de periféricos. Existen muchos tipos de este software, en esta sección se mencionan dos comunes y compatibles con la mayoría de los microprocesadores y sistemas operativos existentes en el mercado.

2.7.1 TeamViewer

Team Viewer es una aplicación que se utiliza para controlar una computadora personal, desde distancia. También se puede utilizar para compartir el escritorio, conferencias en línea y para transferencia de archivos entre computadoras. El programa es compatible con los sistemas operativos Windows, Mac OS, Linux, iOS, Android y Windows Phone. La función principal de Team Viewer es un control remoto de las computadoras, además de permitir brindar soporte técnico de manera remota. [19]

Con el uso de este software es posible controlar el sistema sin necesidad de periféricos.

2.7.2 VNC Viewer

Este software es competencia del mencionado anteriormente ya que posee las mismas funcionalidades, permitiendo realizar mejoras o modificaciones a la herramienta de medición sin la necesidad de utilizar periféricos o permanecer al lado de ella.

VNC es un programa que permite tomar el control del ordenador servidor remotamente a través de un cliente multiplataforma. Una vez instalado VNC en el ordenador, es posible acceder a él desde cualquier parte del mundo a través de Internet y desde cualquier dispositivo, como ordenadores o smartphones. Disponible para-Windows, Mac, Linux, Android y iOS[20].

2.8 Diseño de experimentos

Como parte del proyecto es necesario la validación del instrumento de medición por ese motivo se realiza la investigación de procesos o mecanismos utilizados para verificar la funcionalidad de la herramienta de medición diseñada.

El diseño de experimentos permite realizar una división de un proceso en estudios de carácter de prototipo, exploratorios y estudios exhaustivos, abordando de manera adecuada los diferentes tipos de variabilidad que se pueden encontrar, con esto se analizan de manera provechosa los resultados obtenidos de manera que se puedan tomar decisiones argumentadas. [22]

Para realizar el diseño de experimentos no se recomienda el uso de datos históricos, ya que puede no conocerse las condiciones en las que estos resultados fueron obtenidos.

2.8.1 Estudio R&R del sistema de medición

Los estudios R&R permite evaluar simultáneamente la repetibilidad y la reproducibilidad, estos estudios, se realizan de modo experimental y se evalúa que parte de la variabilidad total observada en el producto es atribuible al error de medición; además, permite cuantificar si este error es mucho o poco en comparación con la variabilidad del producto y con las tolerancias de la característica de calidad que se mide.[9]

Es importante tener claro el concepto de repetibilidad y reproducibilidad, el primero se refiere a la precisión en condiciones lo más controladas posibles, con diferencias pequeñas de tiempo. Se define como la variación alrededor de la media. El segundo, es la precisión de las mediciones en distintas condiciones, como distinto operador, instrumento de medida o cambio de laboratorio.

Con el diseño de experimentos y el análisis de varianza es posible realizar la prueba de repetibilidad y reproducibilidad del sistema de medición.

Según [8] los pasos recomendados para realizar un estudio R&R son los siguientes:

1. Realizar las pruebas al menos con dos operadores distintos.
2. Seleccionar un conjunto de piezas que serán medidas en varias ocasiones por cada operador
3. Decidir el número de veces que cada operador medirá la pieza.
4. Realizar el análisis estadístico de los datos, ANOVA es el más efectivo para este análisis.

2.8.1.1 Estudio tipo 1

Un estudio tipo 1 del sistema de medición evalúa únicamente la variación que proviene del sistema de medición. Específicamente, este estudio evalúa los efectos del sesgo y la repetibilidad sobre las mediciones de un operador y una parte de referencia.[41]

Se debe utilizar este método al principio del análisis de sus sistemas de medición para concentrarse únicamente en el sistema de medición, y no en cualquier otra fuente de variación. De hecho, muchas organizaciones requieren un estudio tipo 1 del sistema de medición como el primer paso en un análisis de un sistema de medición completo.[41]

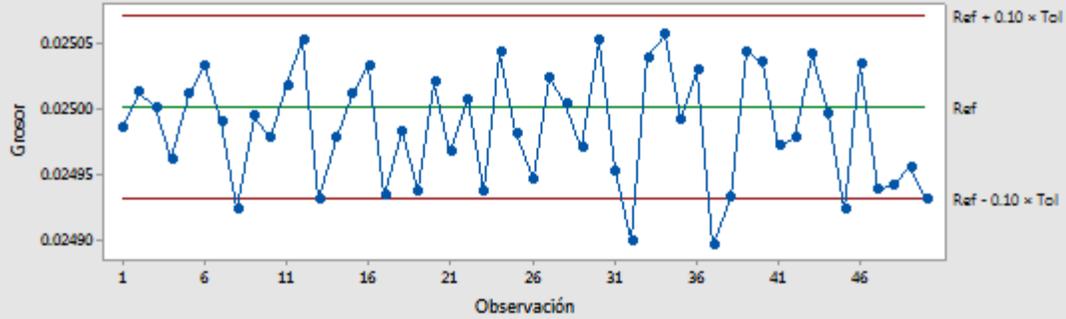
Para evaluar la repetibilidad de un sistema de medición, Minitab calcula la métrica Cg para comparar la variación del estudio (la dispersión de las mediciones del sistema de medición) con un porcentaje de la tolerancia. Los valores de Cg mayores que 1.33 indican que la dispersión de las mediciones del sistema de medición es adecuadamente estrecha en relación con su rango de tolerancia.[41]. Un ejemplo de cómo se puede observar los resultados de esta prueba se observa en la Figura 2.5 donde se obtiene el valor de sesgo así también como el de la variable Cg.

Estudio tipo 1 del sistema de medición para Grosor

Nombre del sistema de medición :
Fecha del estudio:

Notificado por:
Tolerancia: 0.0007
Misc:

Gráfica de corridas de Grosor



Estadísticas básicas		Sesgo		Capacidad	
Referencia	0.025	Sesgo	-0.000015	Cg	0.53
Media	0.024985	T	2.3847019	Cgk	0.42
Desv.Est.	0.0000438	ValorP	0.021		
6.0 × Desv.Est. (VE)	0.0002625	(sesgo de la prueba=0)		%Var(Repetibilidad)	37.50%
Tolerancia (Tol)	0.0007			%Var(Repetibilidad y sesgo)	47.52%

Figura 2.5. Ejemplo estudio tipo1 mediante software Minitab. [42]

3 Metodología

Como el proyecto consiste en el diseño de un producto, es importante utilizar una metodología de diseño, ya que es una herramienta útil que permite encaminar a la solución del problema evitando que se desvíe en conceptos externos que no deban intervenir en el proceso creativo, facilita la aplicación de conocimientos multidisciplinarios y obtener soluciones como respuesta a un análisis previo. Uno de los métodos más completos y descriptivos es el de Ulrich. En la Figura 3.1 se muestra un diagrama que contiene las etapas de la metodología mencionada.

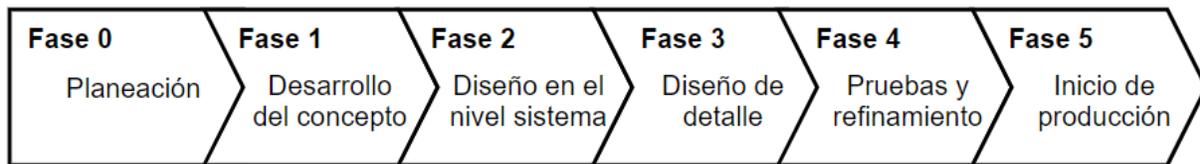


Figura 3.1. Proceso de desarrollo de solución. Fuente [26].

Para esta sección referente a la metodología se desarrollan las etapas 1 y 2 las cuales contemplan la toma de decisiones para generar una solución. Para este desarrollo se realizan las siguientes etapas:

1. Determinación de Necesidades.
2. Determinación de Especificaciones y Métricas.
3. Descomposición Funcional del Problema.
4. Búsqueda de Información Interna y Externa.
5. Generación de Conceptos.
6. Evaluación de Conceptos.
7. Propuesta de Solución.

3.1 Determinación de necesidades

3.1.1 Entrevista e interpretación de necesidades

Para lograr identificar las necesidades por parte del grupo CCG se realizó una entrevista con el Ingeniero Luis Carlos Rosales Alpizar, miembro del grupo antes mencionado. Luis Carlos también es el encargado de definir los requerimientos y dar la aprobación de si la solución propuesta es la buscada por la empresa. La entrevista consiste en una explicación por parte del cliente donde define el problema que presentan actualmente y los requerimientos del nuevo sistema a diseñar.

En la tabla 3.1 se transcriben fragmentos de la entrevista y se coloca al lado la necesidad interpretada a dicho comentario.

Tabla 3.1. Interpretación de necesidades a partir de la entrevista con el cliente. Elaboración propia.

Cliente:	Luis Carlos Rosales	Entrevistador: Mario Araya Alvarado
Profesión:	Ingeniero Electrónico	
Dirección:	Virtual	Fecha: 30/09/2021
Mención del cliente		Necesidad Interpretada
“... Estamos buscando una forma distinta en la que se mide el rendimiento de la compu”		El nuevo sistema debe innovar el método
“Debemos buscar la manera que automatice y sea consistente dependientemente de la plataforma”		El sistema debe funcionar independientemente de la computadora a la cual se le realice la prueba
“Que dependa de la menor cantidad posible de un usuario humano”		Sistema debe llevar a cabo la medición con la menor interacción humana posible
“El problema actual es que no lleva la computadora a reposo profundo”		El sistema debe poseer contadores de tiempo que indique cuando ejecutar las acciones
“Algunas computadoras no retornan debido...”		El sistema debe funcionar independientemente de la computadora a la cual se le realice la prueba
“Necesito crear el evento en que empieza la recuperación del reposo profundo”		El sistema debe provocar el cambio de estado por medio de una acción mecánica
“Tiene que detectar el momento en que la pantalla se apagó”		El sistema debe reconocer el estado de la computadora
“Debe medir el tiempo desde que envía el comando para iniciar la computadora hasta el momento como se encontraba antes”		El sistema debe poder realizar una medición de tiempo
“El tiempo obtenido debe ser calibrado”		Diseñar un método de validación
“Debe contar con cierta precisión”		El sistema debe realizar mediciones con al menos 200 ms de precisión

3.1.2 Jerarquización de necesidades

Para la jerarquización de las necesidades se procede a enviarle un documento, véase en el apéndice B.1, al cliente donde se le solicita enumerar la importancia de cada necesidad basándose en la escala de 1 a 5 que se explica más adelante. Como resultado de las respuestas del cliente se

obtuvo la información mostrada en la Tabla 3.2 y en la tabla 3.3. Se realizó una enumeración de cada necesidad para facilitar futuras menciones, la escala de 1 a 5 se interpreta de la siguiente manera:

1. Función indeseable.
2. Función no importante.
3. Función innecesaria pero deseada.
4. Función muy deseable.
5. Función indispensable.

Tabla 3.2 Enumeración de las necesidades. Elaboración propia.

Enumeración	Necesidad
1	El sistema debe funcionar independientemente de la computadora a la cual se le realice la prueba.
2	Sistema debe llevar a cabo la medición con la menor interacción humana posible
3	El sistema debe poseer contadores de tiempo que indique cuando ejecutar las acciones
4	El sistema debe provocar el cambio de estado por medio de una acción mecánica
5	El sistema debe reconocer el estado de la computadora
6	El sistema debe poder realizar una medición de tiempo
7	El sistema debe realizar mediciones con al menos 100 ms de precisión

Tabla 3.3 Jerarquización de las necesidades basado en las indicaciones del cliente. Elaboración propia

Necesidad	Importancia	Importancia Relativa
1	4	El sistema debe funcionar independientemente de la computadora a la cual se le realice la prueba.
2	5	Sistema debe llevar a cabo la medición con la menor interacción humana posible
3	4	El sistema debe poseer contadores de tiempo que indique cuando ejecutar las acciones
4	4	El sistema debe provocar el cambio de estado por medio de una acción mecánica
5	5	El sistema debe reconocer el estado de la computadora
6	4	El sistema debe poder realizar una medición de tiempo
7	5	El sistema debe realizar mediciones con al menos 200 ms de precisión

3.2 Determinación de métricas y especificaciones

Posteriormente, con las necesidades identificadas se procede a determinar las métricas para cada una de ellas, en la Tabla 3.4 se puede observar cada una de las métricas a utilizar.

Tabla 3.4. Métricas definidas para la ejecución del proyecto. Elaboración propia.

Número	Métrica	Importancia	Unidades
Parámetros medibles			
1	Precisión	5	ms
2	Interacción humana	5	Acciones
Parámetros no medibles			
3	Replicable	4	Binario
4	Eventos controlados por tiempo	4	Binario
5	Acción mecánica	4	Binario
6	Reconocimiento de estado	5	Binario
7	Medición del tiempo	5	Binario

Con las métricas definidas se procede con la asignación de los valores marginales e ideales de cada una de ellas, véanse en la Tabla 3.5, la función de estos valores es para determinar el cumplimiento de los objetivos.

Tabla 3.5. Valores marginales e ideales para las métricas definidas. Elaboración propia.

Número	Métrica	Importancia	Unidades	Valor Marginal	Valor Ideal
Parámetros medibles					
1	Precisión	5	ms	<220	<200
2	Interacción humana	5	Acciones	<6	<4
Capacidades de control					
3	Replicable	4	Binario	Cumple	Cumple
4	Eventos controlados por tiempo	4	Binario	Cumple	Cumple
5	Acción mecánica	4	Binario	Cumple	Cumple
6	Reconocimiento de estado	5	Binario	Cumple	Cumple
7	Medición del tiempo	5	Binario	Cumple	Cumple

3.3 Descomposición funcional del problema

3.3.1 Entradas y salidas

Para el sistema a desarrollar, se posee como entradas la fuente de energía que alimentará a la herramienta y las señales de control, de esta manera se obtiene como salida la medición del tiempo en las condiciones necesarias. La Figura 3.2 muestra un bloque con las entradas y salidas del sistema de forma general.

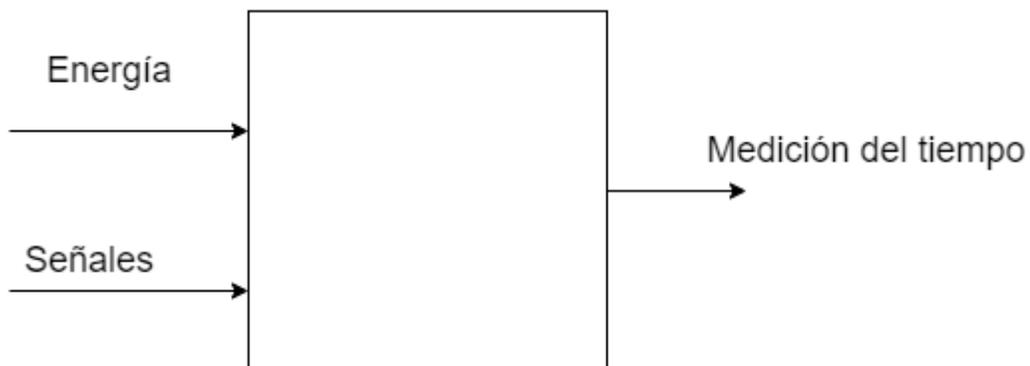


Figura 3.2. Diagrama de bloques de entradas y salidas. Elaboración propia

3.3.2 Definición de subsistemas

Con el objetivo de disminuir la dificultad y de aumentar la comprensión del sistema que se realizó, este se ha dividido en varios subsistemas, con el objetivo de identificar las diferentes partes y funciones que posee cada uno de estos. También se descompuso el problema general en varios problemas específicos más sencillos de resolver.

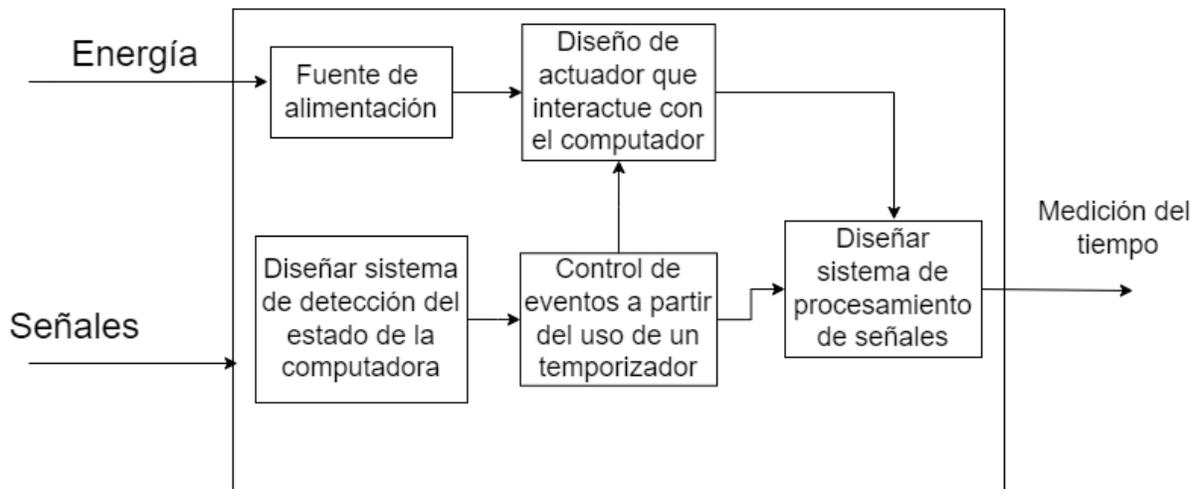


Figura 3.3. Diagrama de subsistemas del sistema de control y monitorización. Elaboración propia

3.3.2.1 Fuente de alimentación

Este componente cumple la función de brindar energía al sistema y a los circuitos que se utilicen, así también como al actuador que tendrá interacción con el computador.

3.3.2.2 Diseño de sistema de detección del estado de la computadora

Como parte de los requerimientos se solicita que el sistema pueda diferenciar el estado que se encuentra la computadora para poder realizar la medición del tiempo de manera precisa, se diseñó un método que compara el estado inicial de la computadora y que realice la medición hasta que el computador vuelva a ese estado inicial luego de haber pasado por el modo reposo profundo.

3.3.2.3 Control de eventos a partir del uso de un temporizador

Se estima necesario este bloque debido a que la medición del tiempo es desde el estado reposo profundo y para asegurarse que la computadora se encuentre en ese estado debe transcurrir un aproximado de 15 min desde que se activa el reposo, además el inicio del contador del tiempo también debe ser preciso para ocurrir en el momento justo que se envía la señal al actuador.

3.3.2.4 Diseño de actuador que interactúe con el computador

Esta parte es fundamental para el funcionamiento del sistema, es necesario de un mecanismo que a partir de una señal realice una interacción con la laptop, de manera más específica, esta interacción corresponde a la acción de presionar una tecla del teclado.

3.3.2.5 Diseño de sistema de procesamiento de señales

Este viene a ser el cerebro del sistema, donde se controla el manejo de las señales, desde la señal de accionamiento del actuador, hasta la señal de los temporizadores y estados de la computadora, administra la información recibida de todos los demás subsistemas y envía señales a partir de esta información.

3.4 Generación de conceptos

Para esta sección del proyecto se generan múltiples conceptos de solución con base en la investigación interna y externa de la metodología [26]. Se ha consultado documentación académica y se han tenido conversaciones con el cliente, además de asesoramiento de otros ingenieros de la empresa.

Para presentar un panorama general de los subproblemas y las diversas soluciones que puede brindarse a cada uno de estos se presenta el esquema de la Figura 3.4, con una síntesis de las opciones.

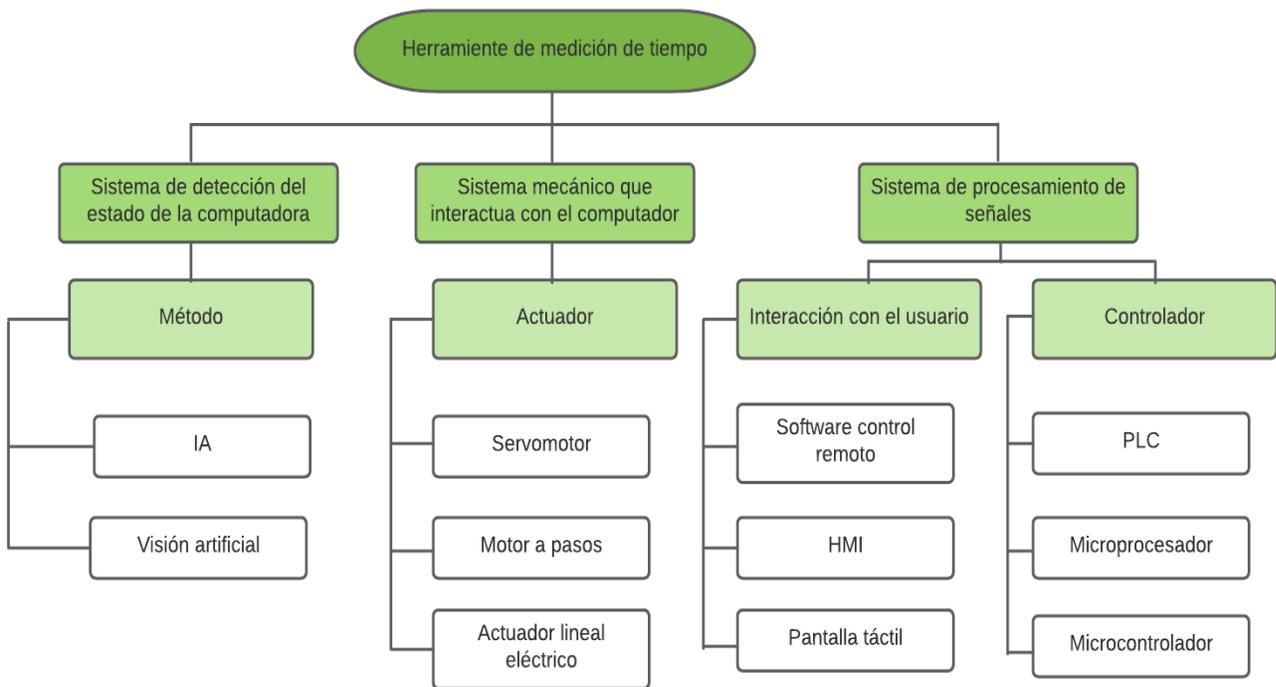


Figura 3.4. Plantilla para el proceso de combinación de los conceptos a evaluar. Elaboración propia

Se pueden generar varios conceptos a partir de la Figura 3.4, se escogen solamente los tres preferidos para realizar la comparación y escoger un favorito que cumpla lo más posible de manera satisfactoria los criterios de selección.

3.4.1 Concepto 1

El primer concepto generado consiste en utilizar Inteligencia artificial para identificar el estado de la computadora, utilizando redes neuronales o algún algoritmo evolutivo que pueda recibir de entrada una imagen y sea capaz de clasificar o identificar cual es la imagen del escritorio cargada completamente o si se encuentra en el estado de reposo.

Como controlador se plantea el uso de un PLC, el cual se activa por medio de una interfaz HMI, este dispositivo es el encargado de los distintos temporizadores que se vayan a utilizar y también de enviar la señal de activación al actuador lineal eléctrico, este último al ser activado y extenderse realiza un contacto con alguna tecla del computador lo que inicia el proceso de finalización del estado de reposo.

La Figura 3.5 muestra la combinación seleccionada para la generación del concepto 1.

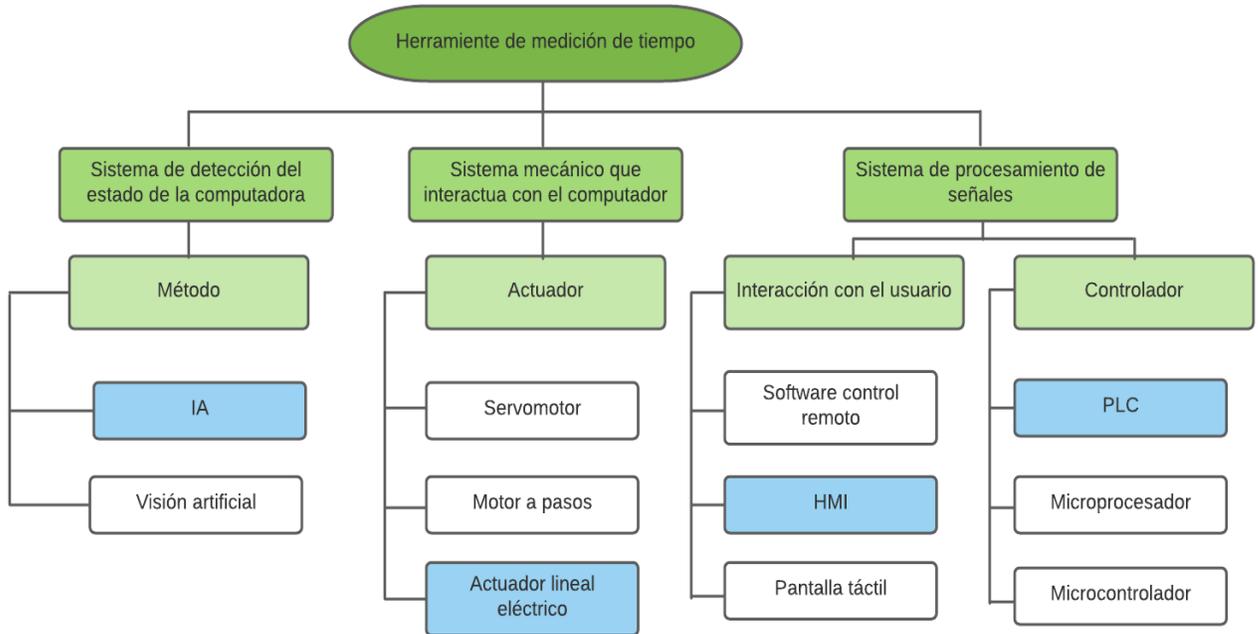


Figura 3.5. Concepto 1. Elaboración propia.

3.4.2 Concepto 2

El segundo concepto consiste en la utilización de un microcontrolador para el procesamiento de señales, este dispositivo es el encargado del manejo de los temporizadores, así como del envío y la recepción de señales.

Para este concepto se propone utilizar el método de Visión Artificial para reconocer el estado de la computadora, por medio del procesamiento de imágenes, haciendo uso de filtros o binarizaciones, para mencionar algunos ejemplos.

El sistema físico que interactúa con el computador corresponde al diseño de una pieza que se acople a un motor de pasos de tal manera que al accionar el motor provoque que la pieza diseñada posea contacto con la computadora portátil. El usuario puede iniciar todo este proceso descrito al interactuar con una pantalla táctil.

La Figura 3.6 muestra la combinación seleccionada para la generación del concepto 2.

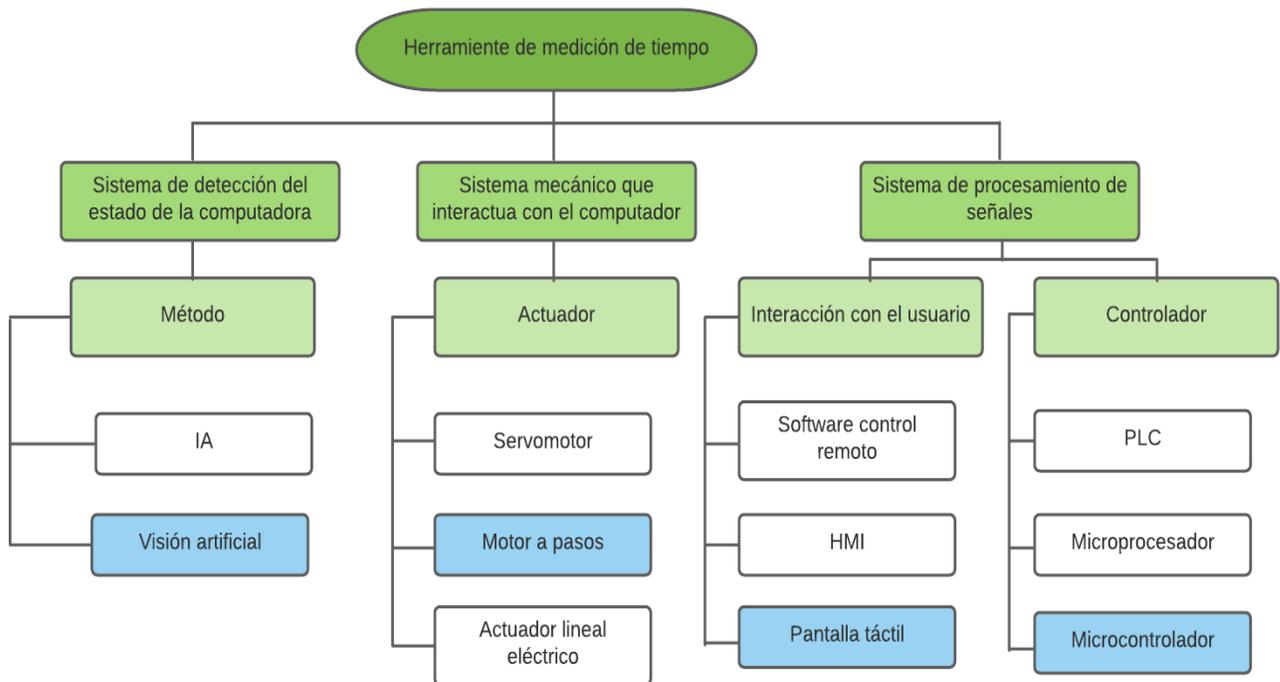


Figura 3.6. Concepto 2. Elaboración propia.

3.4.3 Concepto 3

El tercer concepto de solución planteado consiste en la utilización de un microprocesador para el manejo de las señales y temporizadores. Se hace uso también de la Visión artificial para indicar el estado de la computadora. En lugar del motor a pasos se utiliza un servomotor e igual es necesario el diseño de una pieza que encaje en este, la manera de activación y costo son ejemplos de las diferencias que posee con el motor a pasos.

El usuario utiliza un software de control remoto para manejar el microprocesador, lo que quiere decir que puede activar el proceso de medición de tiempo por medio de otro computador o un teléfono. VNC viewer o Team Viewer son ejemplos de posibles softwares de este tipo.

La Figura 3.7 muestra la combinación seleccionada para la generación del concepto 3.

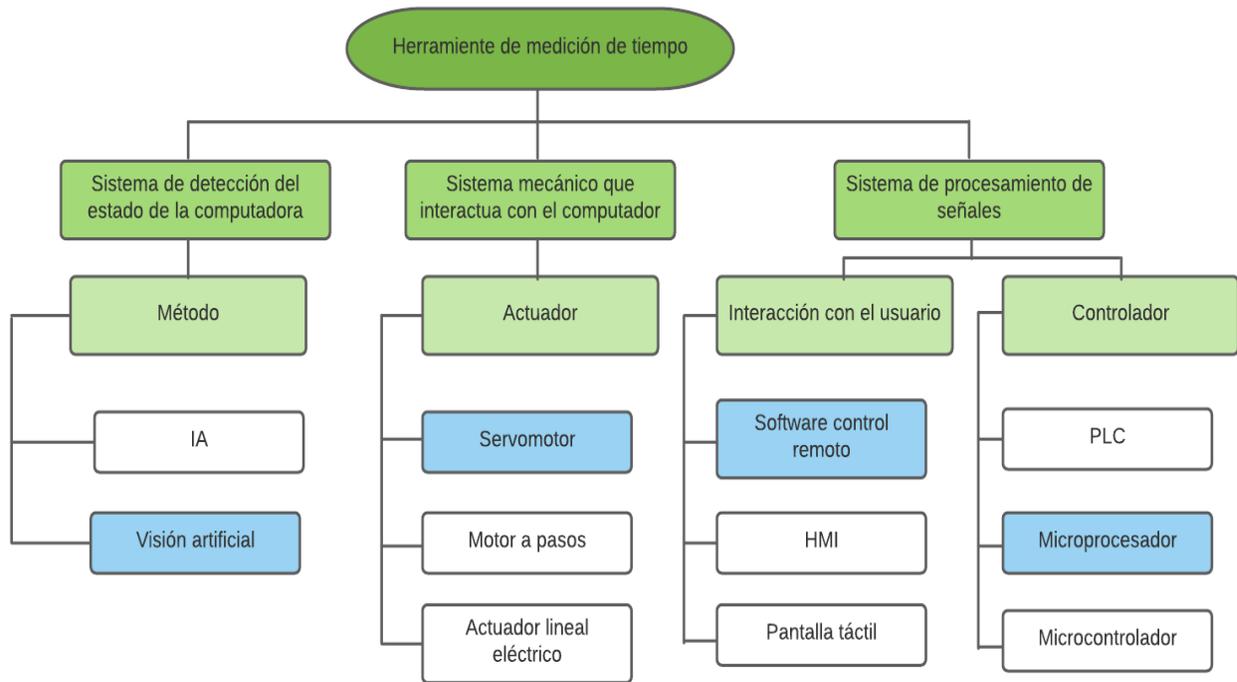


Figura 3.7. Concepto 3. Elaboración propia

3.5 Evaluación de conceptos

3.5.1 Criterios de selección de conceptos

En esta sección se presenta los criterios de selección a utilizar para realizar la escogencia del concepto ganador. La Tabla 3.6 muestra los criterios de selección identificados con base en la jerarquización de necesidades realizada en la Tabla 3.3 y Tabla 3.4 así como otras funciones o características consideradas de importancia al momento de escoger la solución a realizar.

Tabla 3.6. Criterios para la selección de conceptos. Elaboración propia.

#	Criterio de selección	Descripción	Peso
1	Replicabilidad	Se refiere a la poca interacción humana necesaria para su uso.	20%
2	Herramientas de control	Se refiere a las herramientas que brinda el sistema de control para cumplir sus funciones, como cantidad de temporizadores, posibilidad de uso de algoritmos de programación, manejo de actuadores.	20%
3	Costo	Costo general de la implementación del proyecto.	10%
4	Disponibilidad	Indica si se cuenta con la facilidad de conseguirlo dentro de la empresa o si es posible conseguirlo dentro del país de manera sencilla y cuanto tomaría obtenerlo.	20%
5	Facilidad de implementación	Referente a la facilidad de instalar el sistema posterior a la etapa de diseño.	15%
6	Cantidad de elementos para instalación	Se refiere a los elementos que deben instalarse para que el dispositivo de control pueda cumplir sus tareas.	15%

Se puede observar que los criterios con mayores pesos corresponden al de la replicabilidad, herramientas de control y disponibilidad. Este primero debido a que es una indicación del cliente que fue categorizada como función indispensable.

El segundo, herramientas de control es necesario para cumplir con el objetivo que se requiere de controlar todo el sistema por medio de temporizadores y posibilidad de usar algoritmos que definan la lógica necesaria para el funcionamiento de la herramienta.

Luego seguiría el de disponibilidad, aunque el cliente no haga mención directa con respecto a este criterio debido a la naturaleza del proyecto y las limitantes que se posee con respecto a las fechas de entrega es necesario analizar la opción más veloz de poder realizar.

Por último, se posee los criterios de facilidad de implementación y cantidad de elementos instalación, los cuales están directamente relacionados con el parámetro interacción humana y el

desarrollo de la herramienta lo más intuitivamente posible, por medio de menos objetos necesarios para su uso, menor es la complejidad.

3.5.2 Evaluación de conceptos

En esta sección se muestra la selección del concepto ganador, se realiza la escogencia asignándole un valor a cada criterio definido en la sección anterior, el primer valor indicado con la letra C posee valores del rango del 1 al 5, cada valor se interpreta de la siguiente manera:

- 1: No cumple con el criterio.
- 2: Cumplimiento mínimo del criterio.
- 3: Cumplimiento regular del criterio.
- 4: Cumplimiento adecuado del criterio.
- 5: Cumplimiento a cabalidad del criterio.

Además del valor de calificación mencionado, también se le asigna un valor relativo indicado con la letra E, este valor varía dependiendo el peso que posea el criterio, siendo el resultado de la multiplicación del peso por el valor C otorgado. Estos valores se muestran en la Tabla 3.7.

Al final se suma el porcentaje de cada concepto, de esta manera se procede a escoger el concepto cuyo valor sea mayor en comparación al resto.

Tabla 3.7. Calificación de cumplimiento de criterios y evaluación de conceptos. Elaboración propia

		Conceptos					
		Concepto 1		Concepto 2		Concepto 3	
		Control por PLC		Control por microcontrolador		Control por Microprocesador	
Criterio de selección	Peso	C	E	C	E	C	E
Replicabilidad	20%	4	0.8	4	0.8	4	0.8
Herramientas de control	20%	4	0.8	3	0.6	4	0.8
Costo	10%	2	0.2	5	0.5	4	0.4
Disponibilidad	20%	3	0.6	4	0.8	5	1
Facilidad de implementación	15%	3	0.45	3	0.45	4	0.6
Cantidad de elementos para instalación	15%	3	0.45	3	0.45	4	0.6
	Total		3.30		3.6		4.20
	Lugar		3		2		1
	Continuar	No		No		Desarrollar	

Luego de aplicar la escala de calificación a cada criterio de selección se llega a la conclusión que el concepto a implementar para la solución del problema corresponde al concepto 3, el cual es el que posee mayor puntaje, como se observa en la Tabla 3.7.

Se puede ver que el concepto 1 y el 2 poseen menores valores en criterios como la facilidad de implementación y cantidad de elementos para instalación. Ambos necesitan su propio lenguaje de programación, lo que puede aumentar la complejidad ya que podría existir la posibilidad de requerir mezclar más de un lenguaje para implementar ya sea el IA o el método de Visión Artificial, además que sería necesario de otros elementos para su funcionamiento, como la pantalla táctil o el dispositivo donde se vaya a mostrar el HMI.

Otro factor importante por el cual gana el concepto 3 es el de disponibilidad, debido a que en Intel se posee microprocesadores que pueden ser utilizados por este mismo motivo es que se le asigna un mejor valor en costo, ya que en el concepto 2 aunque un microcontrolador puede poseer un menor precio a un microprocesador, por supuesto esto puede depender de los modelos que se seleccionen más adelante, como ya se posee el microprocesador el costo no es un factor importante

en esta opción. Esto sin contar que las otras dos opciones poseen más dispositivos necesarios para que el usuario pueda interactuar con el dispositivo.

Una ventaja del concepto 3 es que toda la configuración y programación necesaria se puede realizar con el microprocesador sin la necesidad de utilizar otros dispositivos, esto sin contar los actuadores o sensores que se requieran. Aunque debido a la selección de la visión artificial como método de reconocimiento del estado de la pantalla es probable que no se requiera ningún sensor lo que implicaría menor cantidad de componentes facilitando la implementación.

3.6 Propuesta de concepto de solución

Finalmente, para esta se muestra el concepto escogido, como se observa en la Figura 3.8. Consta de las siguientes partes:

- **Sistema de detección del estado de la computadora:** Visión Artificial.
- **Actuador:** Servomotor.
- **Interacción con el usuario:** Software de control remoto.
- **Controlador:** Microprocesador.
- Cámara.
- Dispositivo interacción con el usuario.

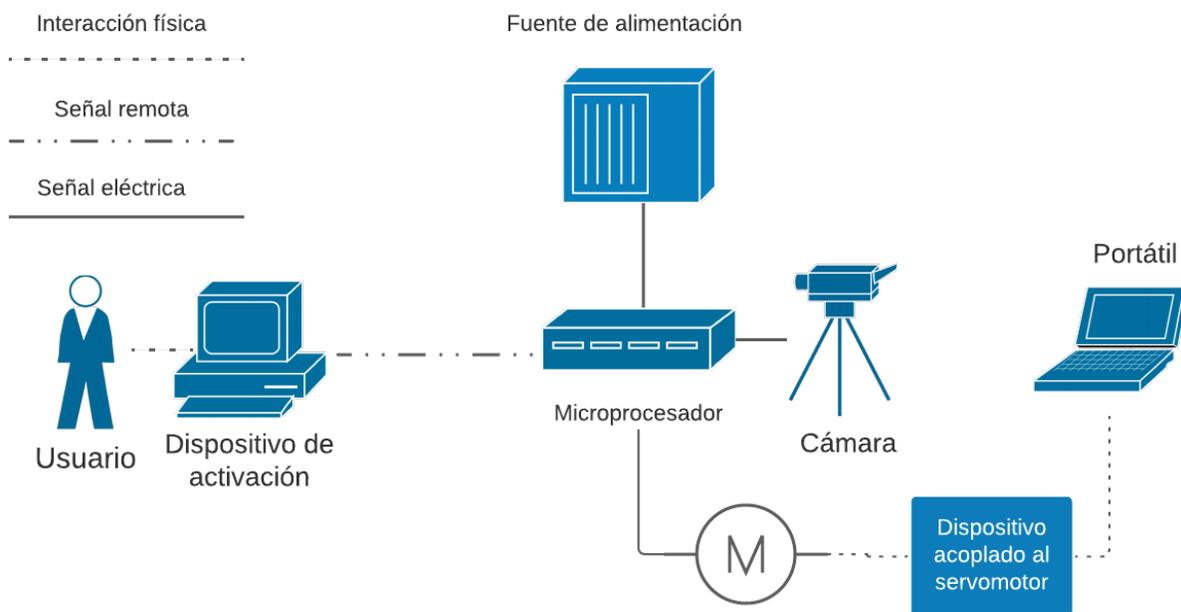


Figura 3.8. Diagrama del concepto de solución seleccionado. Elaboración propia

4 Propuesta de diseño

En esta sección se procede con la toma de decisiones en la selección de componentes específicos a utilizar para la creación del concepto de solución mencionado en la sección anterior, específicamente el de la Figura 3.8. Además, se presenta la configuración necesaria, como los diagramas de flujo de los procesos. Los dispositivos y tecnologías contempladas son:

- El procesador.
- La fuente de alimentación del procesador.
- El software de control remoto.
- La cámara.
- El servomotor.
- Dispositivo que realiza el contacto con la computadora.
- Soporte ajustable para la cámara.

4.1 Selección del procesador

Para iniciar es importante definir qué características como mínimo son necesarias que posea el microprocesador. El actuador del sistema es controlado por medio de este microprocesador, por lo tanto, se requiere que posea pines de salida, lo cual es importante pero no un factor decisivo ya que todos los microprocesadores cuentan con al menos un pin que puede ser utilizado como salida.

Un factor más importante es que posea interfaz de cámara o algún puerto que se le pueda agregar este periférico, como el caso de las cámaras con puerto USB 3.0, las cuales son muy comunes, este factor es determinante ya que sin el uso de la cámara no sería posible realizar el proceso de detección por medio de visión artificial.

Por último, también resulta importante que sea posible la conexión de este dispositivo por medio de manera inalámbrica, ya sea por bluetooth o preferiblemente WI-FI, esto debido a que es necesario la comunicación y el control del procesador de manera remota para cumplir con las especificaciones del concepto de solución propuesto.

Como se puede observar, es sencillo encontrar en el mercado microprocesadores que cumplan con estos requerimientos, existen muchas marcas disponibles, pero por cuestiones de facilidad al momento de tener acceso a ellos y de sencillez en su uso, así también como gran

disponibilidad de módulos o productos compatibles en el mercado se utiliza un microprocesador de la familia Raspberry pi. En la tabla 4.1 se menciona algunos de los modelos más conocidos y utilizados de esta familia y sus respectivas características.

Tabla 4.1. Características de los modelos raspberry pi más comunes [27].

Característica	PI 4	PI 3 B	PI Zero
Bluetooth	4.0	2.0	Sí
WI-FI	Sí	Sí	No
CSI (Interfaz de la cámara)	Sí	Sí	Sí
GPIO	40 pines	40 pines	40 pines
DSI (Interfaz de display)	Sí	Sí	No
Sistema Operativo	Raspbian	Raspbian	Raspbian
Alimentación	5V/3A USB-C	5.1V-2.5A micro USB	5V/1A micro USB
Velocidad	2 GHz	1.4 GhZ	1000 MHz
Puertos USB	2xUSB 3.0 y 2x USB 2.0	4x USB 2.0	2x Micro USB

Con las características de cada modelo identificados se procede a realizar la matriz de selección.

Tabla 4.2. Matriz de selección del microprocesador. Fuente propia.

		Modelos					
		Raspberry pi 4		Raspberry pi 3 B		Raspberry pi zero	
Criterio de selección	Peso	C	E	C	E	C	E
Entradas y salidas	5%	5	0.25	5	0.25	5	0.25
Costo	20%	2	0.4	4	0.8	5	1
Disponibilidad	25%	5	1.25	5	1.25	4	1
Wifi	15%	5	0.75	5	0.75	0	0
Interfaz de la cámara	25%	5	1.25	5	1.25	5	1.25
Cantidad de puertos USB	10%	5	0.5	4	0.4	3	0.3
	Total		4.4		4.7		3.8
	Lugar		2		1		3
	Continuar	no		Seleccionado		no	

Como se puede observar en la Tabla 4.2 el modelo seleccionado es el Raspberry Pi 3, aunque posee características muy similares a la versión mejorada que corresponde al Raspberry Pi 4, el modelo seleccionado posee un precio menor en el mercado y cumple con las especificaciones necesarias para realizar el sistema de medición de tiempo, por lo tanto, el invertir más dinero en un modelo más reciente resulta en un desperdicio de recursos. Todos los modelos poseen el mismo sistema operativo, misma cantidad de entradas y salidas y capacidad de conexión bluetooth por ese motivo no se consideraron como factores determinantes para realizar la selección. En cuanto a los puertos USB se le asigna mayor puntaje al modelo PI 4 ya que posee más variedad, lo que implica que es compatible con una gama mayor de periféricos, que en este caso el único de interés es la cámara.

Aunque la velocidad del procesador es fundamental debido a lo que se requiere realizar es una medición de tiempo, no fue incluido entre los factores para la matriz de selección debido a que todos poseen una frecuencia mayor por mucho a la requerida, teniendo en mente que el rango que se requiere en precisión es de 200 ms.

4.2 Configuración del procesador

Con el procesador seleccionado se procede a realizar la configuración necesaria, en esta sección se presenta el diagrama de flujo del código necesario para realizar la toma de la medición del tiempo. El código completo se adjunta en el Apéndice B.2. Para iniciar se aclara que se debe realizar la instalación del sistema operativo, no se realiza mucho énfasis en esto debido a que es un paso necesario para todos los modelos Raspberry, no solamente para el caso específico de este proyecto, pero se menciona porque, a pesar de esto, es un paso necesario.

Para poder utilizar el microprocesador se debe descargar el sistema operativo deseado desde la página oficial [28], se seleccionó la opción Raspbian versión completa con interfaz gráfica, debido a la que es la más utilizada, posee mayor documentación y se facilita mucho su uso. La versión completa contiene los programas necesarios para poder escribir el código y compilarlos.

En resumen, se recuerda que el objetivo del proyecto es lograr realizar la medición de tiempo de la transición del estado reposo profundo a la pantalla de inicio por medio del procesamiento de imágenes, es necesario enviar una señal de activación al actuador para que este sea el que interactúe con la computadora portátil y provoque el cambio de estado. La señal de activación del actuador

es enviada luego de un tiempo escogido por el usuario. Teniendo el proceso requerido claro se muestra el diagrama de flujo en la Figura 4.1.

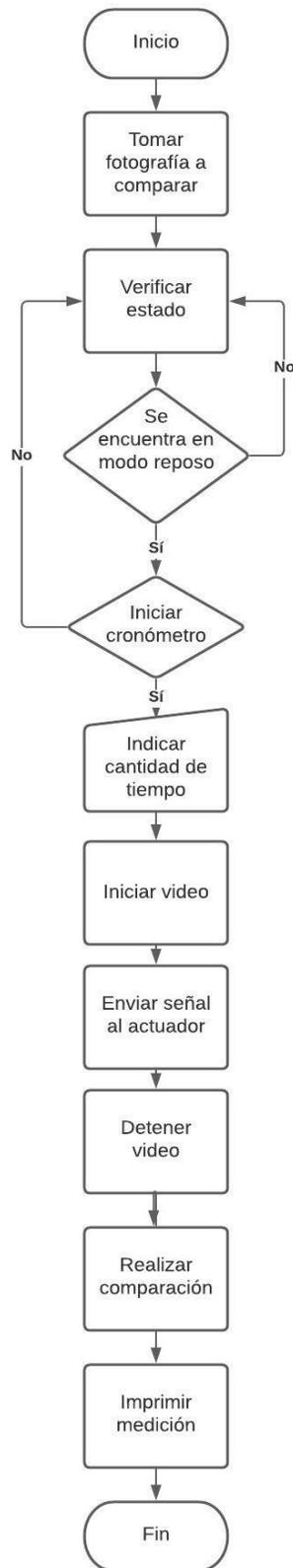


Figura 4.1. Diagrama de flujo del proceso de medición del tiempo. Elaboración propia

4.2.1 Verificación de estados

En esta sección se explica el procedimiento utilizado para obtener el estado del computador portátil, este procedimiento se realizó por medio del procesamiento de imágenes. El primer paso necesario fue el de aplicar un método de segmentación para la detección y extracción de la pantalla de la computadora, el segundo el de seleccionar un criterio que diferenciara si la computadora se encontraba “apagada” o “encendida”. Los pasos realizados se muestran a continuación, utilizando como ejemplo para mostrar el proceso las imágenes presentadas la Figura 4.2 y Figura 4.3. El código utilizado completo se encuentra en el Apéndice B2.

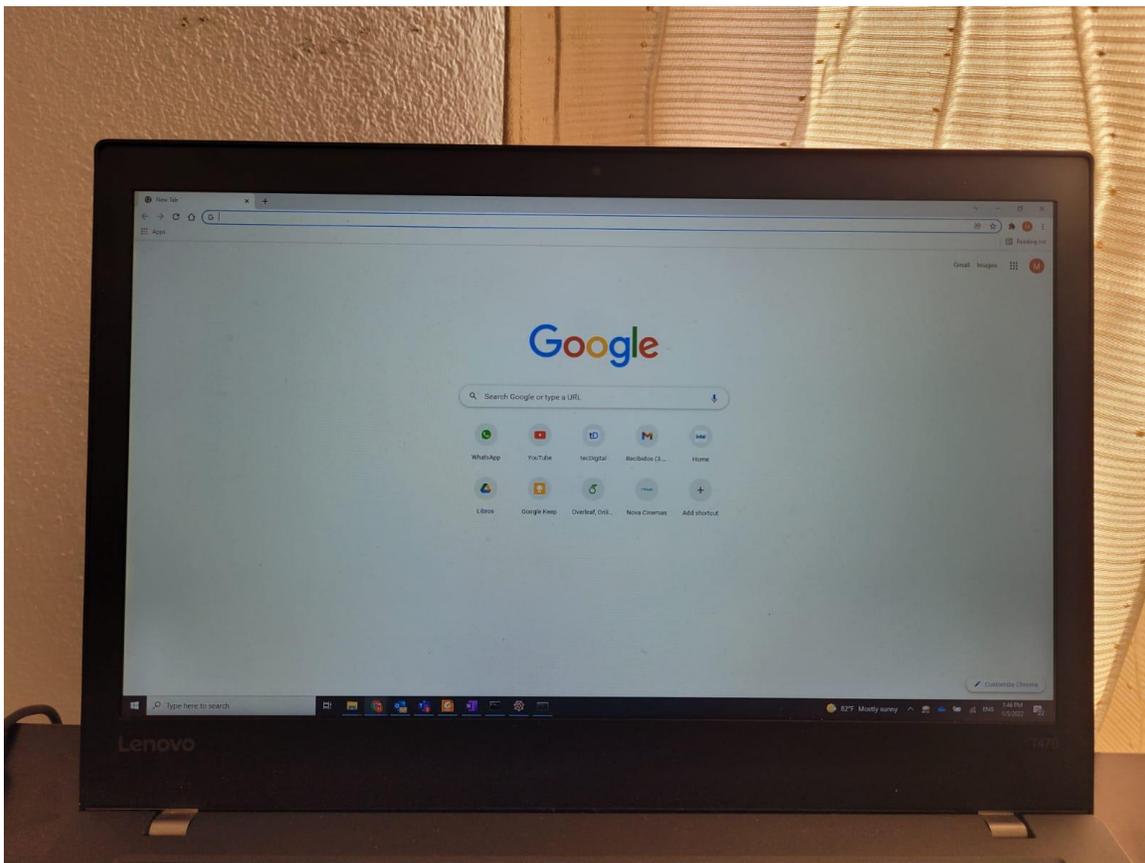


Figura 4.2. Imagen utilizada como prueba para demostración del proceso de verificación de estado de la computadora (modo encendido). Elaboración propia.



Figura 4.3. Imagen utilizada como prueba para demostración del proceso de verificación de estado de la computadora (modo reposo). Elaboración propia.

La extracción de la pantalla se realiza por medio del uso de distintos operadores, el primero que se utiliza es el de la conversión de la imagen en blanco y negro, ya que inicialmente los colores no son un factor importante en la extracción de la pantalla, con lo cual se obtiene un resultado como el que se muestra en la Figura 4.4. Luego se procede a realizar e implementar un filtro con el fin de eliminar cualquier ruido que pueda existir en la imagen, para el ejemplo utilizado la diferencia es insignificante por lo cual no se agrega un ejemplo del resultado.

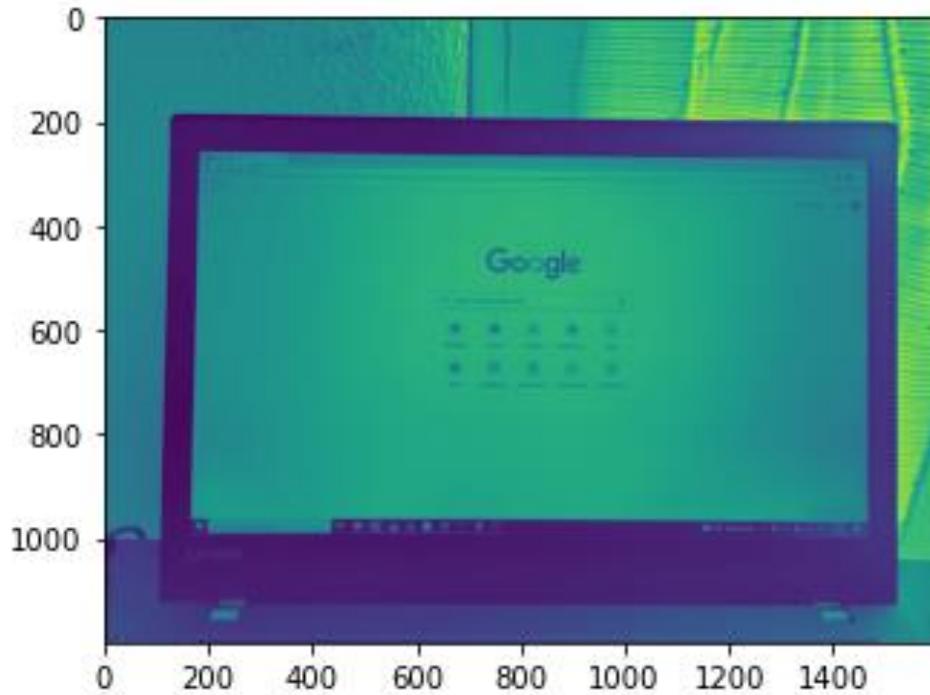


Figura 4.4. Imagen de ejemplo convertida en blanco y negro. Elaboración propia.

El paso que corresponde es el de extraer los bordes de la imagen. Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos, de ahí también la importancia de haber convertido la imagen en blanco y negro, ya que para aplicar esta función era necesario. El resultado de la extracción de bordes se puede observar en la Figura 4.5.

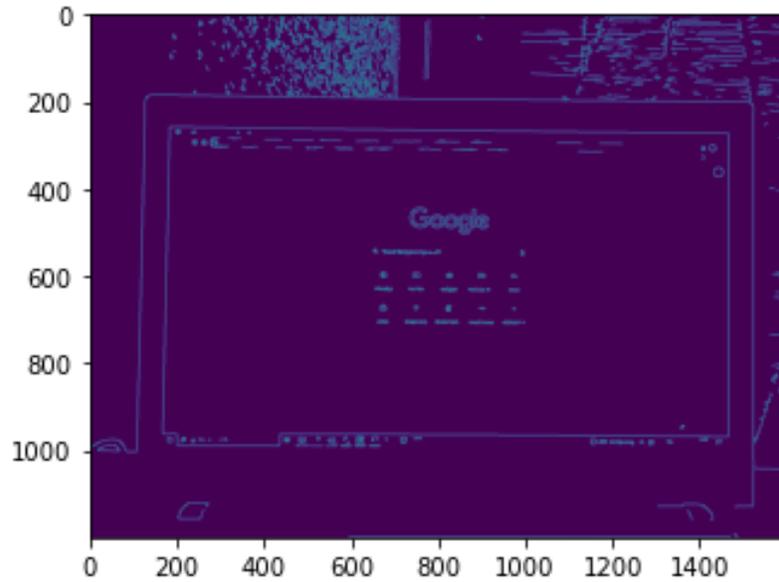


Figura 4.5. Imagen de ejemplo con sus bordes extraídos. Elaboración propia.

Con el objetivo de facilitar los procesos que se realizarán en los siguientes pasos, además de cubrir ciertas discontinuaciones en las líneas de los bordes que son posibles que se encuentren en ciertas imágenes, se realiza el proceso de dilatación, el cual consiste en agregar píxeles a los límites de los objetos de una imagen, lo que se traduce como un ensanchamiento de los bordes, rellenando vacíos no deseados o simplemente haciendo más notorios los bordes. Para el caso de la imagen que se utiliza como ejemplo se puede observar en la Figura 4.6 como se visualiza este operando sobre la imagen.

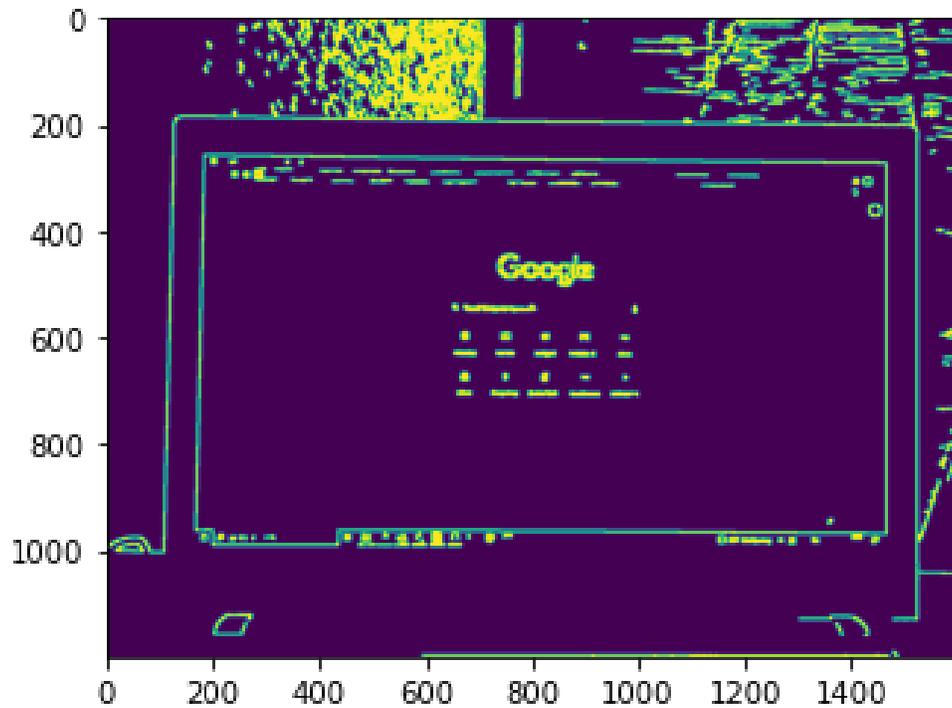


Figura 4.6. Imagen de ejemplo luego de ser dilatada. Elaboración propia.

Una vez que se obtiene la imagen procesada se realizan ciertas operaciones sobre ella. El primero es el de obtener todos los contornos que posea la imagen, para extraer los contornos se realiza una medición del perímetro de todos los hallados y se utilizan valores umbrales para eliminar elementos que sean muy grandes o pequeños al valor umbral escogido. De esa manera con el contorno deseado, el cual sería el de la pantalla, se localiza las coordenadas de los vértices y se realiza un recorte en la imagen original, lo que da como resultado la extracción de la pantalla de la computadora de las imágenes originales, tal como se observa en las Figura 4.7 y la Figura 4.8.



Figura 4.7. Pantalla extraída de la imagen original (modo encendido). Elaboración propia.

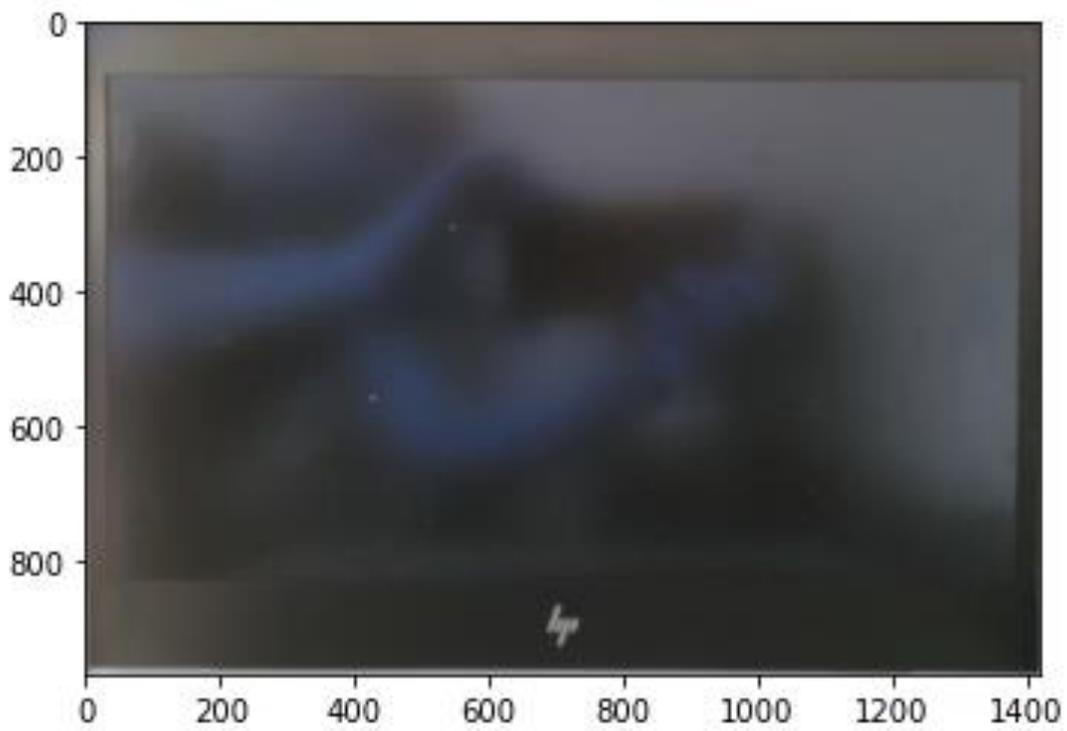


Figura 4.8. Pantalla extraída de la imagen original (modo reposo). Elaboración propia.

Con la extracción de la pantalla realizada, el paso que corresponde es reconocer la característica que pueda ser utilizada como identificadora del estado de la computadora, no se puede utilizar el color como característica, debido a que la pantalla reflectiva que poseen las computadoras siempre presenta algún color en ellas, por lo tanto, se planea diferenciar por medio del brillo de la imagen, para esto se transforma la imagen al espacio HSV. Se sabe que el brillo es una característica capaz de ser utilizada debido a que se extrae el histograma de ambas imágenes y se observa el comportamiento, en modo encendido la gráfica siempre tiende a valores desplazados hacia la derecha con respecto al eje x, véase la Figura 4.9 y la Figura 4.10.

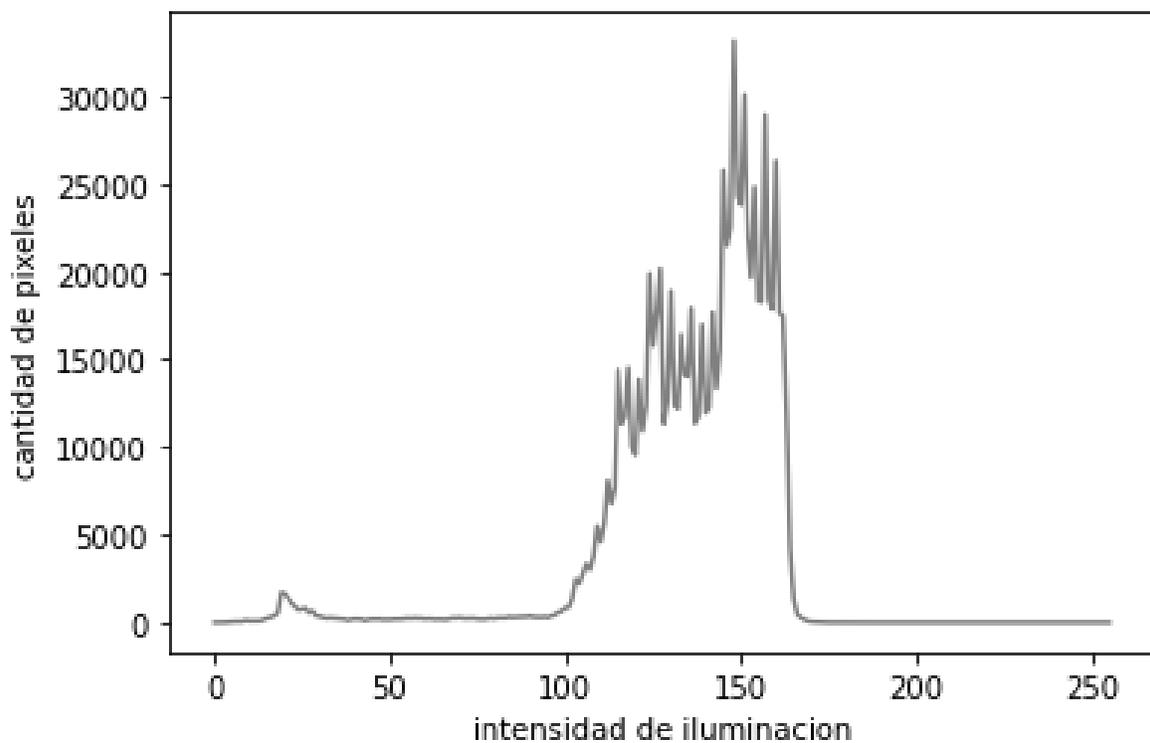


Figura 4.9. Histograma de la imagen original (modo encendido). Elaboración propia.

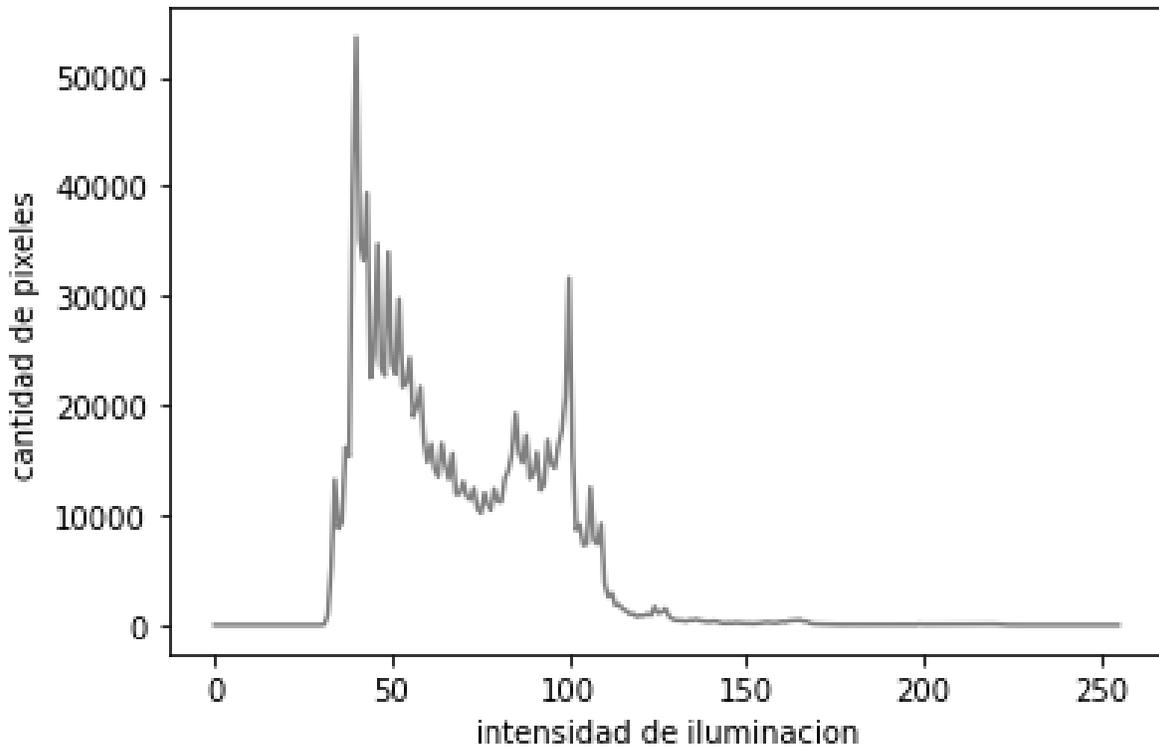


Figura 4.10. Histograma de la imagen original (modo reposo). Elaboración propia.

Por lo tanto, al ver este comportamiento típico solamente es necesario extraer el valor promedio y seleccionar umbrales que permita diferenciar entre un estado y el otro.

4.2.2 Asignar tiempo de espera

Una función fundamental que debe poseer la aplicación es tener un tiempo de espera antes de enviar la señal al actuador con el objetivo de garantizar que la computadora alcance el estado reposo profundo, el tiempo aproximado para lograr que alcance el estado es de 12 minutos, aun así, por cuestiones de versatilidad se decidió permitirle al usuario asignar el tiempo deseado de manera manual, sin olvidar que uno de los objetivos de la aplicación es que sea lo más intuitiva posible.

El código completo para asignar el tiempo se puede observar en el Apéndice B2. Para lograr realizar esta función se hizo uso de dos librerías, la primera corresponde a la librería `time` de Python [29]. Con esta librería es posible realizar la conversión en horas, minutos y segundos.

Además, se decidió agregar una interfaz gráfica para facilitar la implementación de la herramienta al hacerla más intuitiva, para lograr esto último se utilizó la librería de python llamada

Tkinter [30]. En la Figura 4.11 se puede observar la interfaz que es mostrada al usuario y donde debe ser ingresada la cantidad de tiempo necesario para enviar la señal al actuador, la aplicación es capaz de recibir horas, minutos o solo segundos.

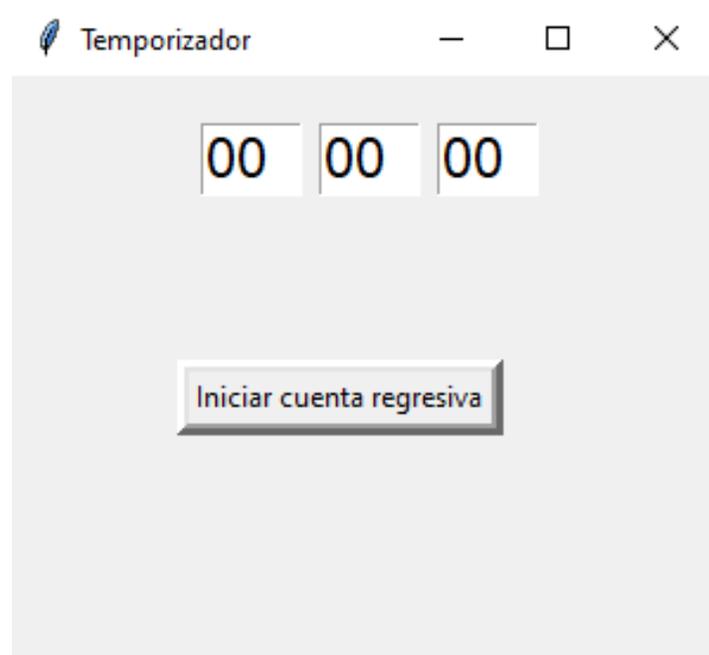


Figura 4.11. Interfaz mostrada al usuario para la asignación del tiempo de espera. Elaboración propia.

4.2.3 Envío de señal al actuador

Hay que recordar que la importancia de este paso es garantizar que la prueba pueda ser aplicada a cualquier computadora que se encuentre en modo reposo profundo, ya que anteriormente se presentaba el problema que al desactivarse la funcionalidad de Wi-Fi no había posibilidad de realizar la transición al estado inicio de pantalla encendido.

Esta señal como se mencionó es enviada, al finalizar el tiempo de espera, al actuador que interactúa presionando una tecla de la computadora portátil y de esta manera se logra el cambio de transición.

Para facilitar la implementación se hizo uso de la librería GPIO zero [31] que posee las raspberrys para manejo de los pines de entradas y las salidas, se le asigna la posición deseada del servo por medio de un valor numérico.

4.3 Selección de la fuente de alimentación del procesador

Como se observa en la Tabla 4.1 para lograr el funcionamiento del microprocesador es requerido la implementación de 5.1 V y 3 A. Existen muchas fuentes de energías posibles como la eólica, solar, neumática entre otras, pero, por cuestiones de facilidad, recordando que el dispositivo puede ser transportado de manera sencilla, además que frecuentemente será utilizado en un laboratorio o en casa de habitación, se alimenta por medio de energía eléctrica.

Cabe recordar que el modelo del procesador utilizado, el cual corresponde al raspberry pi 3B posee una entrada micro USB para conectar la fuente de alimentación, por lo que tampoco se utiliza baterías con el objetivo de evitar el costo de reemplazarlas, así como el de diseñar o adquirir una estructura que permita la transferencia de energía desde una batería a un puerto micro USB.

Con esto definido se busca en el mercado fuentes de alimentación disponibles, los cuales existe una gran variedad que fueron diseñados con el propósito específico de proveer energía a este modelo en particular, a un bajo costo y fácil acceso. En la Figura 4.12 se muestra un ejemplo del tipo de alimentación seleccionado.



Figura 4.12. Fuente de alimentación seleccionado. [32]

4.4 Selección del software de control

Existe la posibilidad de conectar periféricos al microprocesador como el monitor y el teclado para poder hacer uso de este, sin embargo, de acuerdo con la solución seleccionada en la sección 3 se decide por poder controlar la herramienta de manera remota por lo que este método de conectar los periféricos no es utilizado.

Los tres métodos más utilizados para controlar este microprocesador de manera remota se encuentran la Tabla 4.3.

Tabla 4.3. Matriz de selección del método del control remoto. Fuente propia.

		Métodos					
		SSH		VNC		RDP	
Criterio de selección	Peso	C	E	C	E	C	E
Facilidad de uso	25%	3	0.75	5	1.25	4	1
Funcionalidad	30%	4	1.2	5	1.5	4	1.2
Velocidad	20%	5	1	3	0.6	4	0.8
No necesidad de otros programas	10%	4	0.4	5	0.5	4	0.4
Compatibilidad	15%	4	0.6	5	0.75	5	0.75
	Total		3.95		4.6		4.15
	Lugar		3		1		2
	Continuar	No		Seleccionado		no	

El software seleccionado es el VNC, en la Tabla 4.3 se muestran factores que llevaron a tomar esta decisión, pero aun así se da una breve explicación del por qué estos factores. Con respecto a la facilidad de uso, el segundo factor de mayor peso es importante considerar que entre menor cantidad de pasos o menor complejidad para poder hacer uso del software es mejor para no incluir dificultad al uso del sistema.

El software VNC ya viene incluido en el sistema operativo de la Raspberry, por lo que solo es necesario tener el programa en la computadora o celular a utilizar, no es necesario utilizar líneas de comando y tampoco es necesario la descarga de más softwares, como si lo es para el caso del protocolo SSH.

Con respecto a la velocidad el software VNC es el que posee menor rendimiento, pero aun así posee la velocidad suficiente para lograr el objetivo, es importante recordar que VNC y RDP se puede observar la interfaz gráfica al contrario de SSH. RDP puede comprimir el flujo de datos y VNC solo envía pixeles sin procesar por este motivo posee menor velocidad.

La compatibilidad es otro factor que se tomó en cuenta, aunque con poco peso, ya que la herramienta es utilizada usualmente en Windows, el cual es soportado por los 3 métodos, pero el caso de RDP es exclusivo para este último sistema operativo mencionado, en cambio, VNC puede ser utilizado también en Android y en iOS.

4.5 Selección de la cámara

La raspberry pi 3B, el cual fue el escogido, posee puertos USB como se mencionó en la Tabla 4.1, lo que permite poder hacer uso de cámaras con este mismo puerto de entrada y no solo limitarse a la interfaz CSI del dispositivo, por lo que se presenta más opciones al momento de realizar la selección de este componente.

Aunque existe la posibilidad, el hacer uso de cámaras no oficiales por la raspberry puede traer ciertas complicaciones, es necesario instalarle los drivers necesarios lo cual puede resultar en un procedimiento más complejo que simplemente utilizar la interfaz creada con ese propósito que posee el microprocesador.

Un detalle importante por tomar en cuenta es que no todas las cámaras son compatibles con el sistema operativo Linux, el cual es en el que está basado Raspbian que como se había mencionado en la sección de configuración es el utilizado para la creación de la herramienta. La lista de cámaras compatibles se puede encontrar en la referencia número [33].

Una ventaja de las cámaras con puerto USB es la movilidad que permite, en el caso del módulo de cámara de la raspberry, el “cable” de conexión posee una longitud de 15 cm lo que dependiendo de la aplicación puede ser una desventaja, para el caso de este proyecto no es así por lo que no se toma como un factor de peso importante al momento de realizar la selección.

Por último, en cuanto a precio, disponibilidad y rendimiento ambas poseen características similares, por supuesto que estas características pueden variar según el modelo de cámara que se analice, pero en general, las opciones más comunes que se encuentran en el mercado poseen un rango de precio similar que corresponde en promedio aproximadamente a 30 dólares y con una resolución de 5 Megapíxeles.

Teniendo todo esto en cuenta, se decide utilizar el módulo de la cámara para raspberry, ya que posee las características necesarias para garantizar la funcionalidad del sistema y disminuye

la complejidad de su uso. En la Tabla 4.4 se pueden observar las características de la cámara y en la Figura 4.13 se observa el módulo seleccionado.

Tabla 4.4. Propiedades del módulo de cámara v1 para Raspberry pi. [34]

Característica	Requerimiento
Tamaño	25 x24 x 9 mm
Peso	3 g
Resolución	5 MP.
Modos de video	1080p30fps, 720p60fps and 640 × 480p60/90fps
Integración Linux	V4L2 driver available
Sensor	OmniVision OV5647
Resolución del sensor	2592 × 1944 pixeles
Área de imagen del sensor	3.76 x 2.74 mm
Tamaño del pixel	1,4 x 1.4 μm
Tamaño óptico	¼ “
Radio focal	2.9
Distancia focal	3.60mm +/- 0.01



Figura 4.13. Módulo de cámara raspberry seleccionado. [35]

Un detalle fundamental al momento de seleccionar la cámara es tomar en cuenta la cantidad de cuadros por segundo que es capaz de tomar (fps) debido a la manera que se plantea tomar la medición.

Se observa que la cantidad mínima de fps que puede tomar este módulo es de 30 cuadros por segundo, lo que implica que es capaz de recolectar imágenes cada 33ms, lo cual es un valor aceptable dentro del rango de 200 ms que es el requerido, aun así, existe la posibilidad de aumentar la cantidad a 60 o hasta 90 cuadros por segundo. Con estos datos se logra confirmar que el módulo de cámara escogido es funcional para el diseño de la herramienta.

En las Figuras 4.14, 4.15, 4.16 y 4.17 se puede observar el cambio que hay entre distintos cuadros con una configuración de 30fps.

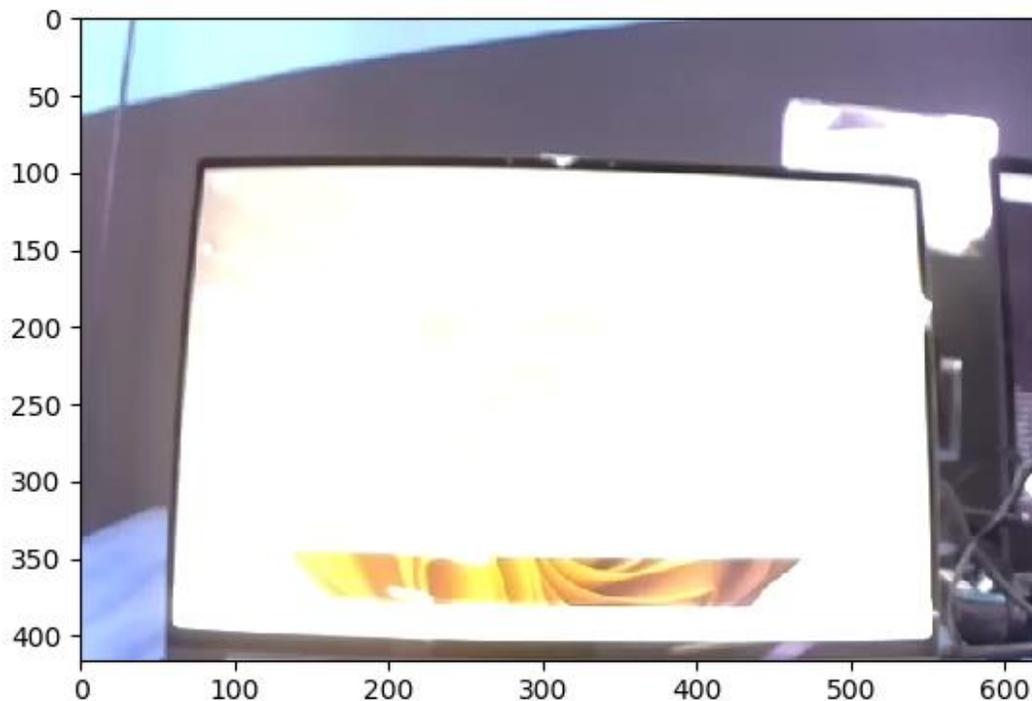


Figura 4.14. Cuadro inicial tomado por cámara con una configuración de 30fps. Fuente propia

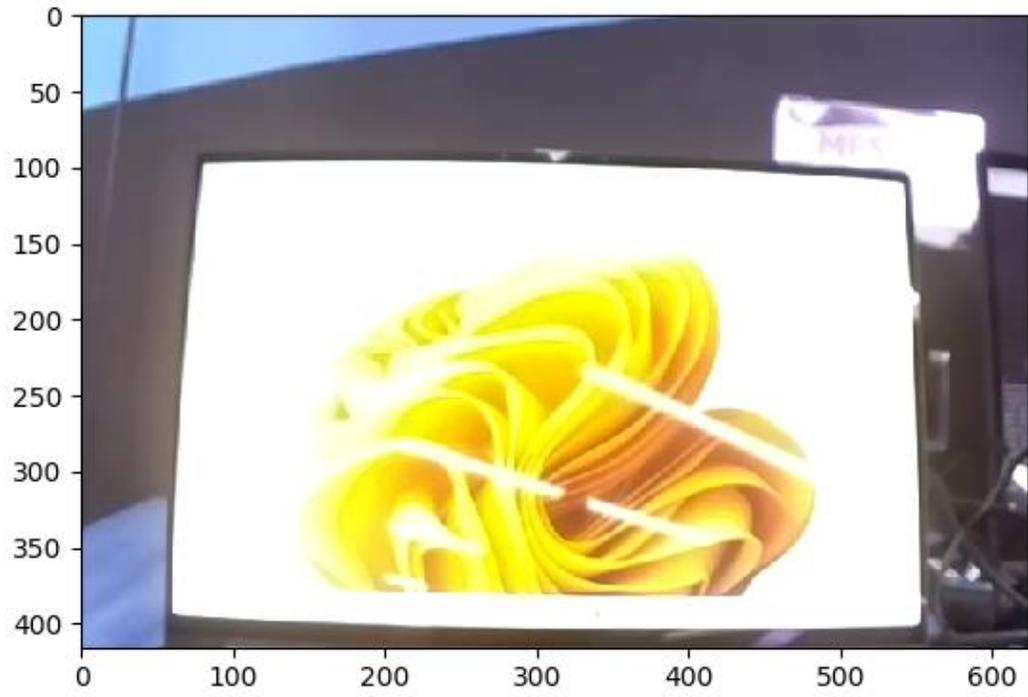


Figura 4.15. Cuadro número 33 tomado por cámara con una configuración de 30fps. Fuente propia

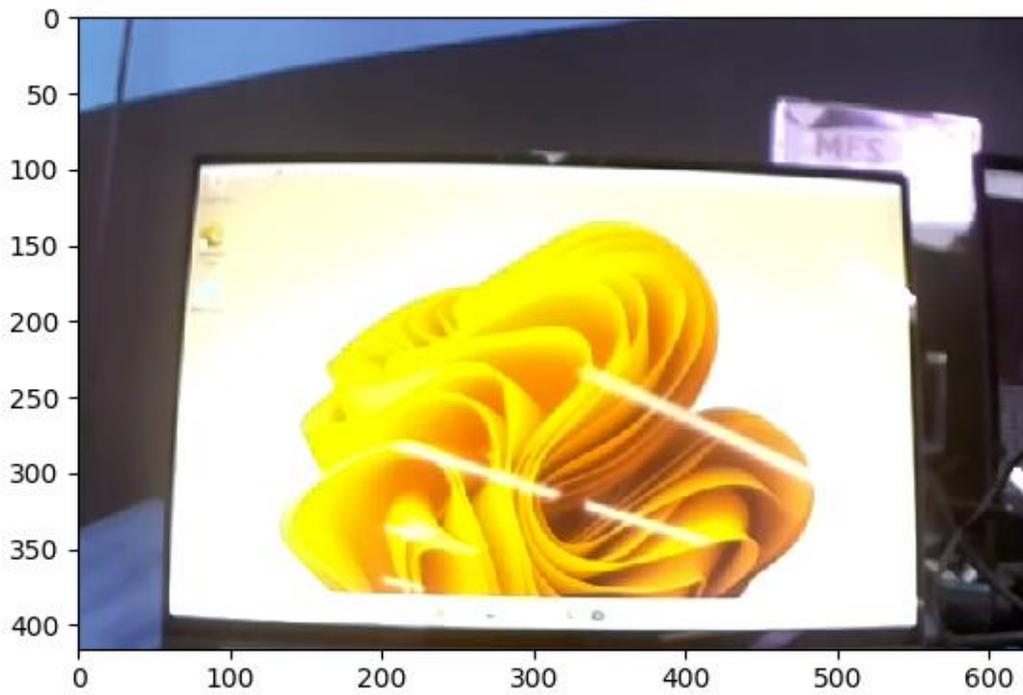


Figura 4.16. Cuadro número 38 tomado por cámara con una configuración de 30fps. Fuente propia

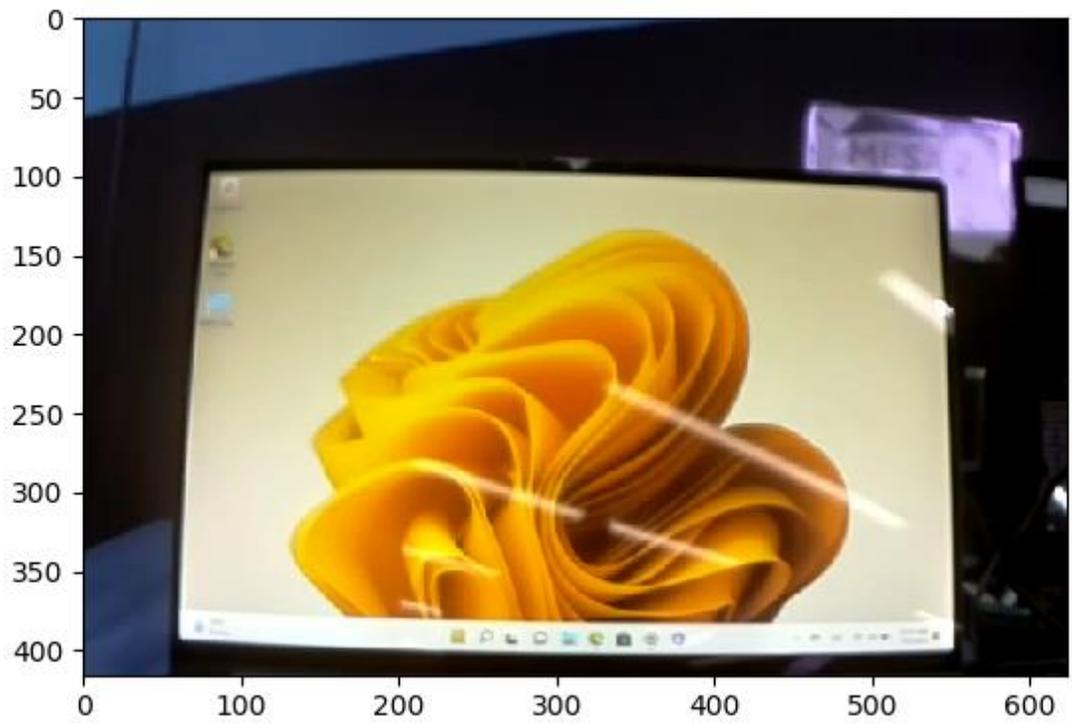


Figura 4.17. Cuadro número 44 tomado por cámara con una configuración de 30fps. Fuente propia

4.6 Selección del servomotor

La propuesta de solución seleccionada en la sección de generación de conceptos se decidió utilizar un servomotor como actuador con el propósito de generar el movimiento que por medio de un mecanismo va a provocar la interacción con la computadora portátil provocando la pulsación de una tecla.

Debido a esto en la Tabla 4.5 se muestran ciertas características de los servomotores más comunes en el mercado con el propósito de poder seleccionar la mejor opción para la herramienta.

Tabla 4.5. Características de los servomotores más comunes.

Característica	Sg90	MG90	MG996
Tamaño	23x 12x 32 mm	22,8 x 12,2 x 28,5 mm	40.7 x 19.7 x 42.9mm
Tensión	4.8 V	4.8V a 6V DC	3.0V - 7.2V DC
Peso	9 g	13,4 g	62.41 g
Velocidad	-	0,1 seg / 60grados (a 4.8V), 0.08 seg / 60 grados (a 6.0V)	60 grados en 0,20 seg (@ 4.8V), 60 grados en 0,16 seg (@ 6.0V)
Torque	-	En 4.8V: 1,8 kg-cm / 25 oz-in. En 6.0V: 2,2 kg-cm / 30,5 oz-in	a 7.2V: 13 kg-cm

Con las características claras de cada opción que se encuentra en el mercado se procede a realizar una matriz de selección para poder escoger cual motor se va a utilizar. La matriz se puede observar en la Tabla 4.6.

Tabla 4.6. Matriz de selección del servomotor. Fuente propia.

		Servomotor					
		MG90		MG996		SG90	
Criterio de selección	Peso	C	E	C	E	C	E
Velocidad	25%	5	1.25	4	1	3	0.75
Precio	25%	4	1	3	0.75	5	1.25
Torque	15%	5	0.75	5	0.75	3	0.45
Disponibilidad	20%	5	1	5	1	5	1
Dimensiones	15%	5	0.75	3	0.45	4	0.6
	Total		4.75		3.95		4.05
	Lugar		1		3		2
	Continuar	Seleccionado		No		No	

La opción ganadora es el motor MG90, en la Figura 4.18 se puede observar este dispositivo, los factores más influyentes fueron el precio y la velocidad. El factor disponibilidad para este caso no fue fundamental debido a que es sencillo conseguir cualquier de los tres modelos, con respecto a facilidad de uso los tres se controlan de la misma manera.

La diferencia principal entre los tres es el torque que poseen, pero debido a la naturaleza del proyecto no es necesario un torque muy elevado ya que solo se utiliza para la presión de una tecla. En cuanto las dimensiones entre menores fueran se consideran una ventaja, entre más compacto sea la herramienta más fácil su traslado.

Con respecto a la tensión necesaria para la alimentación, la raspberry es capaz de suplir 5 V, por lo tanto, es capaz de alimentar cualquiera de las tres opciones. Aunque la SG90 posee las características necesarias y un menor precio se decide la MG90 debido a que la diferencia en el precio es menor de 4 dólares por lo que no se considera como una pérdida de recursos significativa. Por lo tanto, se obtiene un producto de mayor calidad, mayor velocidad y con más documentación por un precio muy similar.

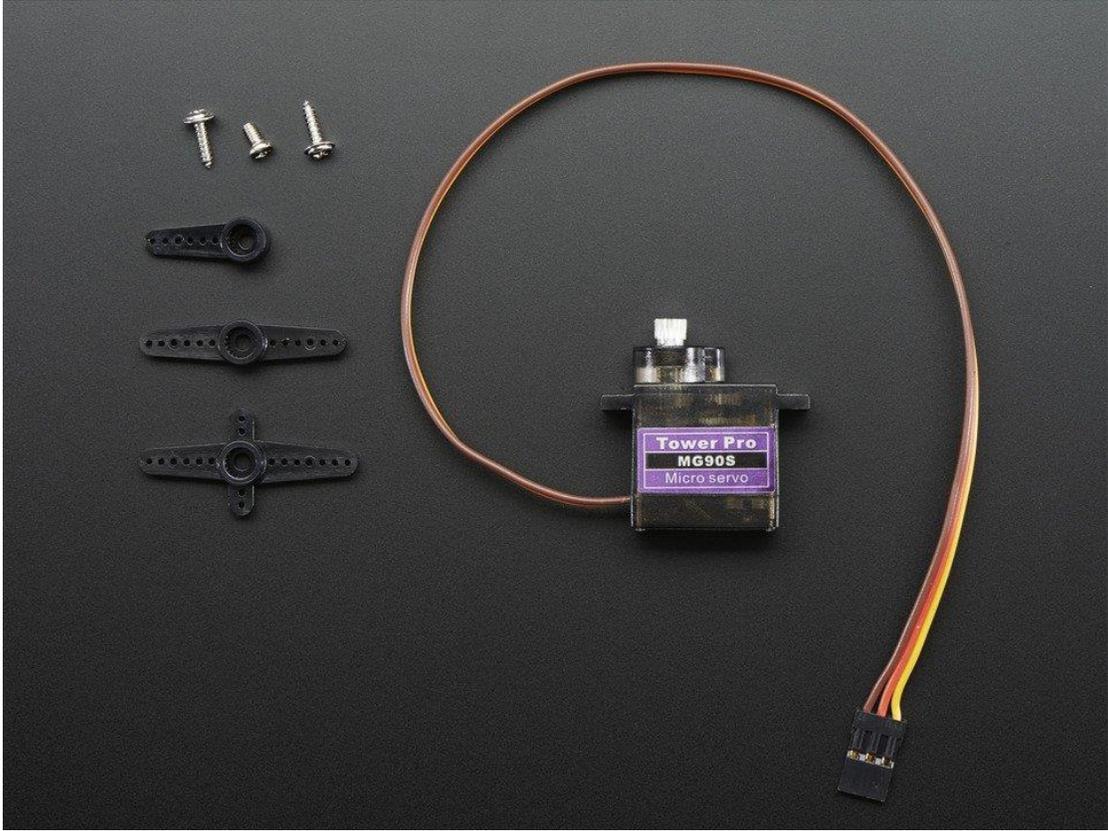


Figura 4.18. Servomotor modelo MG90. [36]

4.7 Diseño del soporte de la cámara

Para el diseño del soporte se toma en cuenta que la empresa posee una impresora 3D, lo que facilita mucho la fabricación y la reducción de los costos por estos motivos se decide fabricar por este medio el prototipo. Por lo tanto, el primer punto a analizar para el diseño del soporte es la selección del material.

En la tabla 4.7 se muestran características de materiales comunes y de fácil acceso con los cuales se puede realizar la manufactura por medio de la impresora y en la Tabla 4.8 se observa la matriz de selección utilizada para escoger el material. Todos los materiales de la tabla son del tipo modelado por deposición fundida, el tipo más común y de acuerdo con los requerimientos del proyecto cumple con la calidad necesaria.

Tabla 4.7. Características de materiales comunes utilizados para impresión 3D. [39]

	Ventajas	Desventajas
PLA	<ul style="list-style-type: none">• Bajo costo• Buena rigidez• Calidad visual	<ul style="list-style-type: none">• Frágil• Baja resistencia a la tensión
ABS	<ul style="list-style-type: none">• Bajo costo• Acabado suave• Resistencia al impacto y al desgaste	<ul style="list-style-type: none">• Con el tiempo presenta deformaciones• Mala precisión dimensional
Nailon	<ul style="list-style-type: none">• Gran resistencia a la fatiga• Gran resistencia a la tensión• Gran resistencia al calor	<ul style="list-style-type: none">• Alto costo• Baja resistencia a la humedad

Tabla 4.8. Matriz de selección del material. Fuente propia.

		Materiales					
		ABS		PLA		NAILON	
Criterio de selección	Peso	C	E	C	E	C	E
Costo	30%	5	1.5	5	1.5	3	0.9
Acabado	10%	3	0.3	3	0.3	4	0.4
Disponibilidad	30%	4	1.2	5	1.5	3	0.9
Facilidad de impresión	20%	5	1	5	1	4	0.8
Durabilidad	10%	3	0.3	4	0.4	5	0.5
	Total		4.3		4.7		3.5
	Lugar		2		1		3
	Continuar	No		Seleccionado		No	

Como se observa en la Tabla 4.8 el material seleccionado es el PLA(ácido poli láctico), aunque en la matriz se observa que el Nailon posee algunas propiedad mejores que el PLA como el acabado y durabilidad se sigue poseyendo como preferencia este segundo material, ya que estas propiedades no se consideran razones de peso para realizar la escogencia, por la naturaleza del proyecto el acabado no requiere que posee una calidad muy alta, ya que el soporte no influye en la medición del tiempo de manera directa, es decir, mientras el soporte brinde su función de mantener la cámara en una posición estable, la calidad no influye en el sistema de medición.

También se prefirió el PLA ante al ABS debido a su rigidez, lo que provoca que el material presente una mayor duración lo que a futuro podría significar una menor inversión en impresiones reduciendo de esta manera los costos. El PLA posee una baja resistencia a la tensión, pero este no es un factor considerado ya que el soporte no va a presentar situaciones donde sea sometido a este esfuerzo de manera que no pueda ser despreciable.

Con el material seleccionado se procede a diseñar la estructura del soporte, se realiza de manera modular para así poseer la capacidad de ajustar la altura lo que podría facilitar el posicionamiento de la cámara ante la computadora portátil. El sistema consta de tres partes los cuales se puede observar en las Figuras 4.20,4.21 y 4.22, las dimensiones se pueden hallar en los

planos que se encuentran en el Anexo B1. La unión de las 3 partes, lo que vendría a ser el ensamble se puede observar en la Figura 4.24.

Para unir cada parte del ensamble general se realizó por medio de uniones insertadas por presión, para colocar la tolerancia de las medidas se utilizó como guía la tabla de ajustes recomendados por ISO, la cual se puede observar en la Figura 4.19.

AGUJERO ÚNICO		EJE ÚNICO		CARACTERÍSTICAS DE ASIENTO	EJEMPLOS
H8	x8			Prensado duro. Montaje a prensa. No necesita seguro.	Coronas de bronce, ruedas.
H8	u8				
H7	s6			Prensado. Montaje a prensa.	Piñón motor.
H7	r6			Prensado ligero. Necesita seguro.	Engranajes de máquinas.
H7	n6			Muy forzado. Montaje a martillo.	Casquillos especiales.
H7	k6			Forzado. Montaje a martillo.	Rodamientos a bolas.
H7	j6			Forzado ligero. Montaje a mazo.	Rodamientos a bolas.
H7	h6			Deslizante con lubricación.	Ejes de lira.
H8	h9			Deslizante sin lubricación.	Ejes de contrapunto.
H11	h9			Deslizante. Ajuste corriente.	Ejes de colocaciones.
H11	h11			Deslizante. Ajuste ordinario.	Ejes-guías atados.
H7	g6	G7	h6	Giratorios sin juego apreciable.	Émbolos de freno.
H7	f7	F8	h6	Giratorios con poco juego.	Bielas, cojinetes.
H8	f7	F8	h9	Giratorios con poco juego.	Bielas, cojinetes.
H8	e8	E9	h9	Giratorios con gran juego.	Cojinetes corrientes.
H8	d9	D10	h9	Giratorios con mucho juego.	Soportes múltiples.
H11	c11	C11	h9	Libre, con holgura.	Cojinetes de máquinas agrícolas.
H11	a11	A11	h11	Muy libre	Avellanados. Taladros de tornillos.

Figura 4.19. Tabla ajustes recomendados por ISO. [43].

Se decide utilizar el ajuste deslizante tipo ajuste corriente, donde el sistema es de agujero único y la nomenclatura corresponde a H11-h9, este tipo de ajuste fue escogido debido a facilidad de ensamble, es posible realizar las uniones sin necesidad de crear agujeros con rosca. Además, las uniones son utilizadas como ejes lo cual es un caso recomendado según la Figura 4.15, con esta selección de ajuste, en conclusión, permite el fácil ensamblaje y la movilidad de las piezas lo que mantiene la característica de altura variable. Se puede observar un diagrama de este ajuste en la

Figura 4.20. Las medidas correspondientes se pueden observar en el anexo B1, en el plano correspondiente.

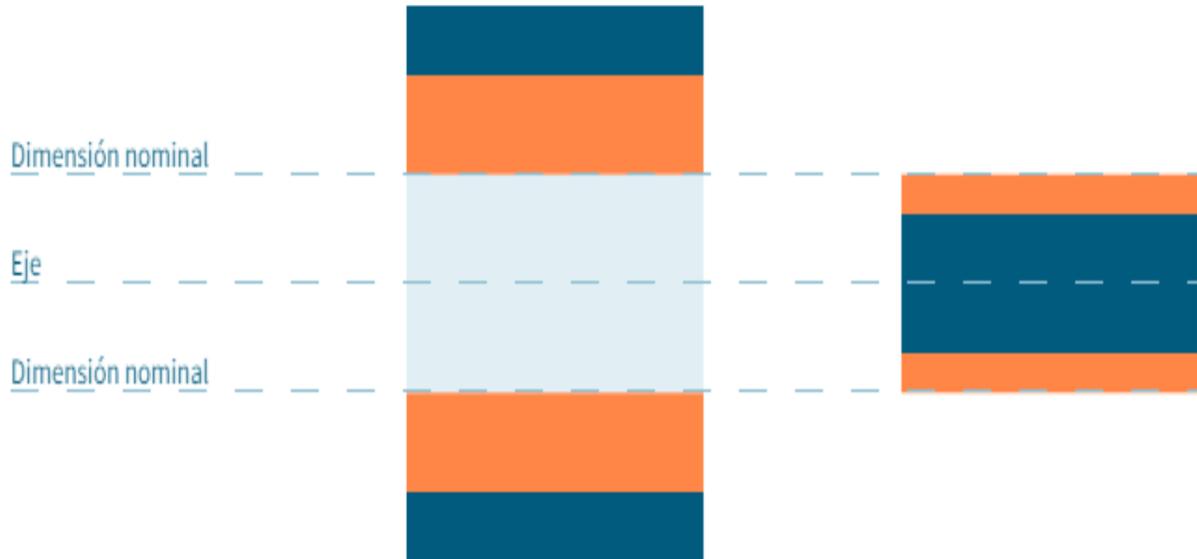


Figura 4.20. Esquema ajuste por deslizamiento ordinario. [45]



Figura 4.21. Base del soporte para la cámara. Fuente propia.

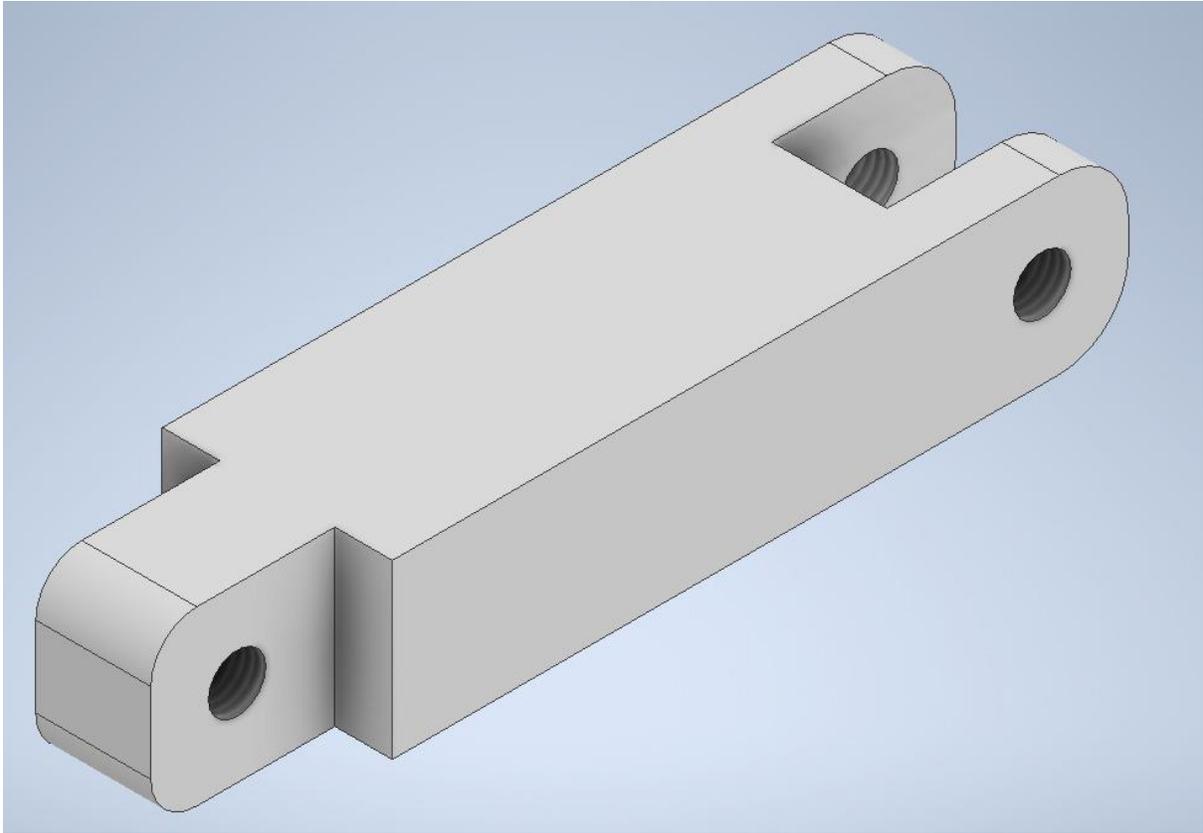


Figura 4.22. Unión entre la base del soporte y el acople para la cámara. Fuente propia.

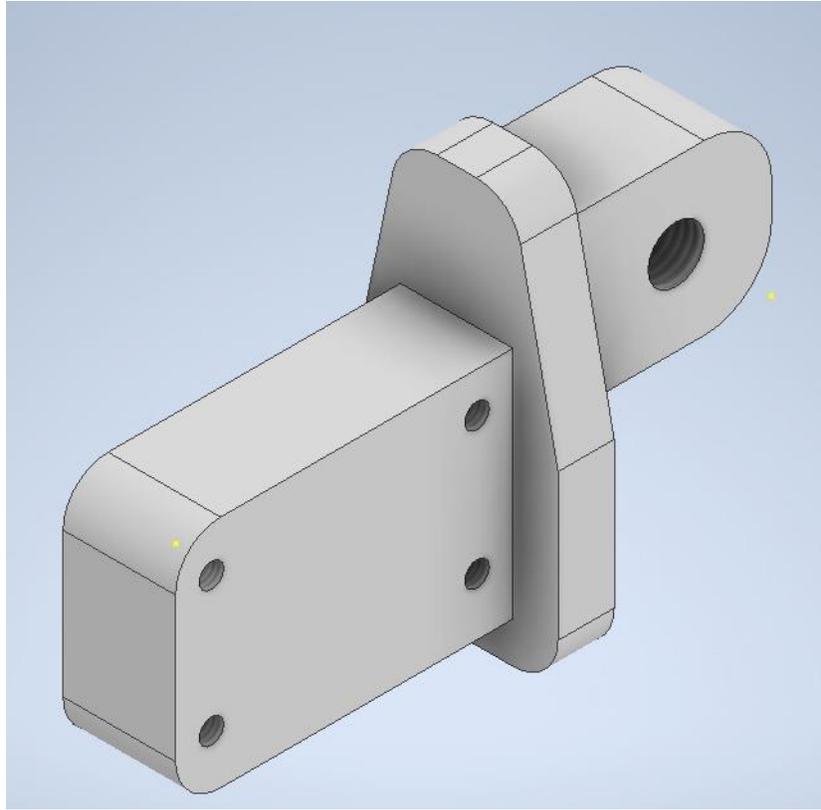


Figura 4.23. Acople para la cámara. Fuente propia

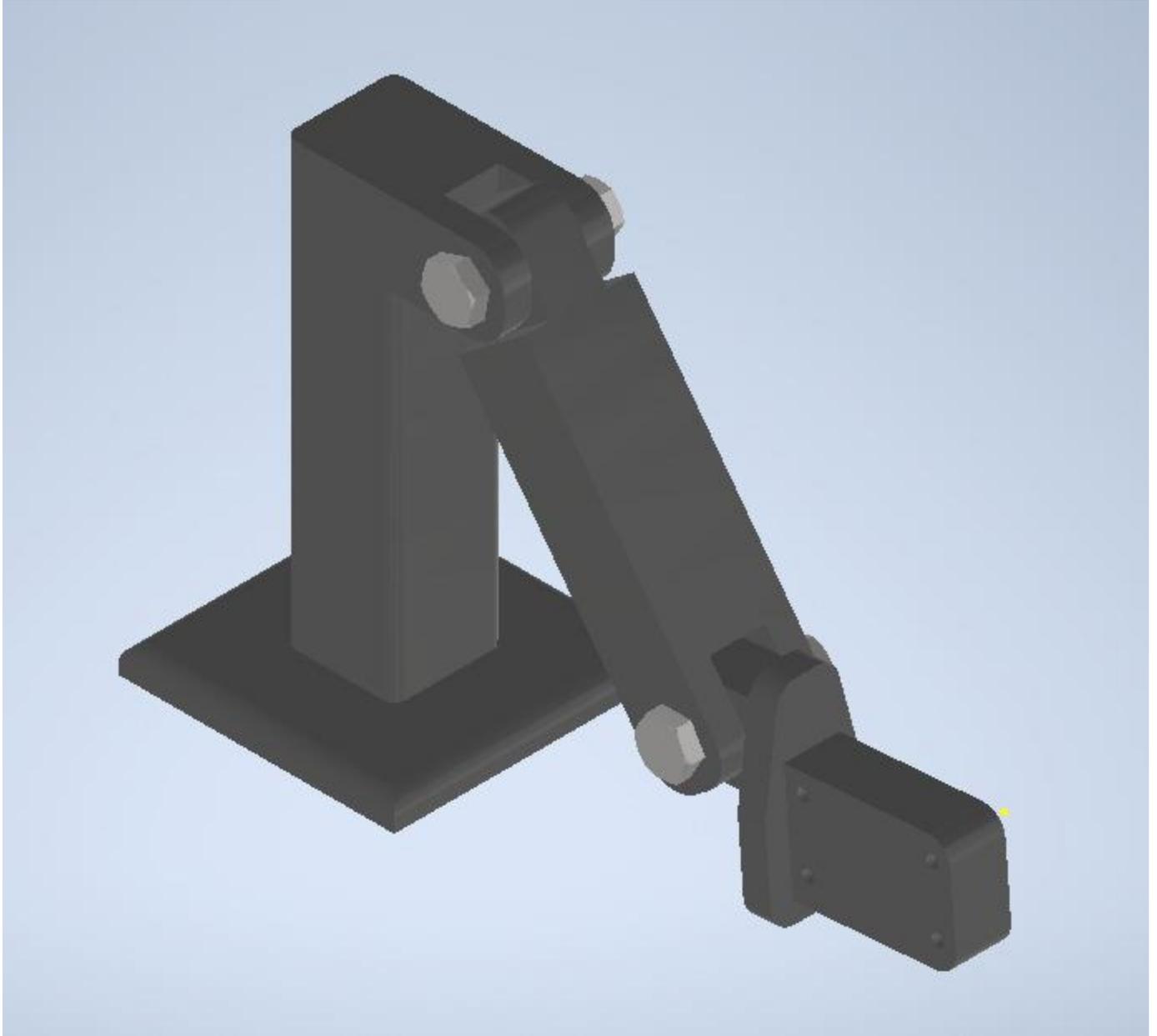
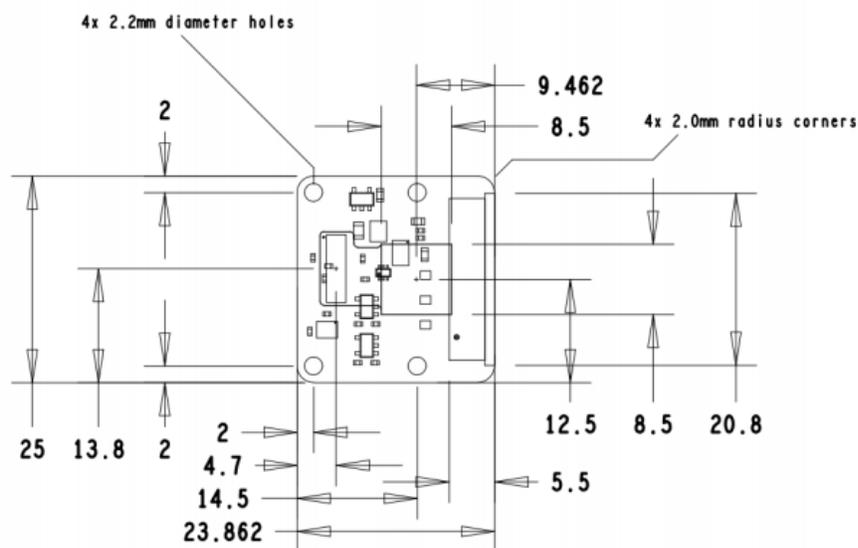


Figura 4.24. Vista isométrica de la estructura completa del soporte de la cámara. Fuente propia

Las medidas para los agujeros donde se coloca la cámara se hicieron basados en la hoja de datos del módulo de cámara para Raspberry Pi, las medidas de dicho modulo se pueden observar en la Figura 4.25.



 Raspberry Pi www.raspberrypi.org © Raspberry Pi 2015			
TITLE	RASPBERRY PI CAMERA MODULE V2.1		
DATE	12/11/2015	REF	RPI-CAM-V2_1
DRAWN	Mike Stimson	APVD	James Adams

Figura 4.25. Dimensiones del módulo de la cámara Raspberry Pi. [38]

En cuanto el diseño final se selecciona otro material con el motivo de alcanzar mayor durabilidad y firmeza en la estructura. Se tomaron en cuenta los materiales más utilizados en el mercado y cuya manufactura es sencilla y fácil acceso en el país. En la Tabla 4.9 se puede observar estos materiales y algunas características básicas de ellas tomadas en cuenta al momento de realizar la escogencia de uno de ellos.

Tabla 4.9. Características de materiales comunes utilizados para manufactura. Fuente propia

	Resistente a corrosión	Forma comercial	Peso	Precio
Acero Inoxidable	Sí	Láminas, barras o perfiles	Alto	Alto
Acrílico	Sí	Láminas	Bajo	Bajo
Aluminio	Sí	Láminas o barras	Medio	Medio

Con los materiales más comunes contemplados se procede a realizar la matriz de selección, la cual se encuentra en la Tabla 4.10.

Tabla 4.10. Matriz de selección del material. Fuente propia.

		Materiales					
		Aluminio		Acero inoxidable		Acrílico	
Criterio de selección	Peso	C	E	C	E	C	E
Costo	25%	4	1	3	0.75	5	1.25
Resistencia a la corrosión	15%	4	0.6	5	0.75	5	0.75
Facilidad de manufactura	30%	5	1.5	4	1.2	3	0.9
Peso	30%	4	1.2	3	0.9	4	1.2
	Total		4.3		3.6		4.1
	Lugar		1		3		2
	Continuar	Seleccionado		No		No	

Basado en la matriz de selección, el material seleccionado corresponde al aluminio para el diseño final de la estructura, los criterios de mayor peso fueron la facilidad de manufactura y peso, el material acrílico fue el peor calificado en facilidad de manufactura debido al grosor de las láminas que se encuentran comúnmente los cuales son de 3 y 6 mm, ninguna posee el grosor que se utiliza para la estructura lo cual implicaría la necesidad de unir varias laminas lo que incrementa la dificultad de manufactura, en cambio el aluminio es sencillo de trabajar y existen barras con el grosor necesario.

El peso fue considerado un factor de importancia debido a que es el causante de los esfuerzos cortantes y de aplastamiento en la estructura, entre menor sean estos pesos, a menor esfuerzos es sometido cada unión que encaja un soporte al otro y mayor son los factores de seguridad que se obtienen.

Luego tenemos el costo, el cual es un criterio importante, pero se le asignó menor valor debido a que la estructura posee dimensiones pequeñas y no se pretende fabricar en masa por lo tanto la inversión, aunque se hubiese escogido el material de mayor costo, no representa una

inversión significativa y como los tres materiales poseen larga durabilidad tampoco es una inversión que se requiera realizar frecuentemente.

Por último, se tiene el criterio de corrosión el cual va ligado con la durabilidad que se mencionó en el párrafo anterior, se le asignó menor peso debido a que la estructura no se planea utilizar en lugares donde haya líquidos o componentes que provoquen la corrosión en las partes diseñadas.

4.8 Diseño del soporte del servomotor

Tomando en cuenta que el servomotor incluye varios tipos de acoples para su uso como se observa en la Figura 4.18, no se presenta la necesidad de diseñar un dispositivo que pueda interactuar con la computadora. La problemática que presente el motor resulta en el corto alcance de estos acoples y la inestabilidad de este cuando es utilizado.

Por ese motivo se origina la necesidad de diseñar un soporte que brinde estabilidad y además logre posicionar el motor en una posición donde el acople posee el alcance suficiente para interactuar con el teclado de la computadora portátil.

Con respecto al material a utilizar y método de manufactura se decide utilizar los mismos a los seleccionados en la sección anterior, con esto en consideración se planea un diseño que utilice la menor cantidad posible de material para reducir los costos y el tiempo de fabricación. En la Figura 4.26 se puede observar el diseño realizado.

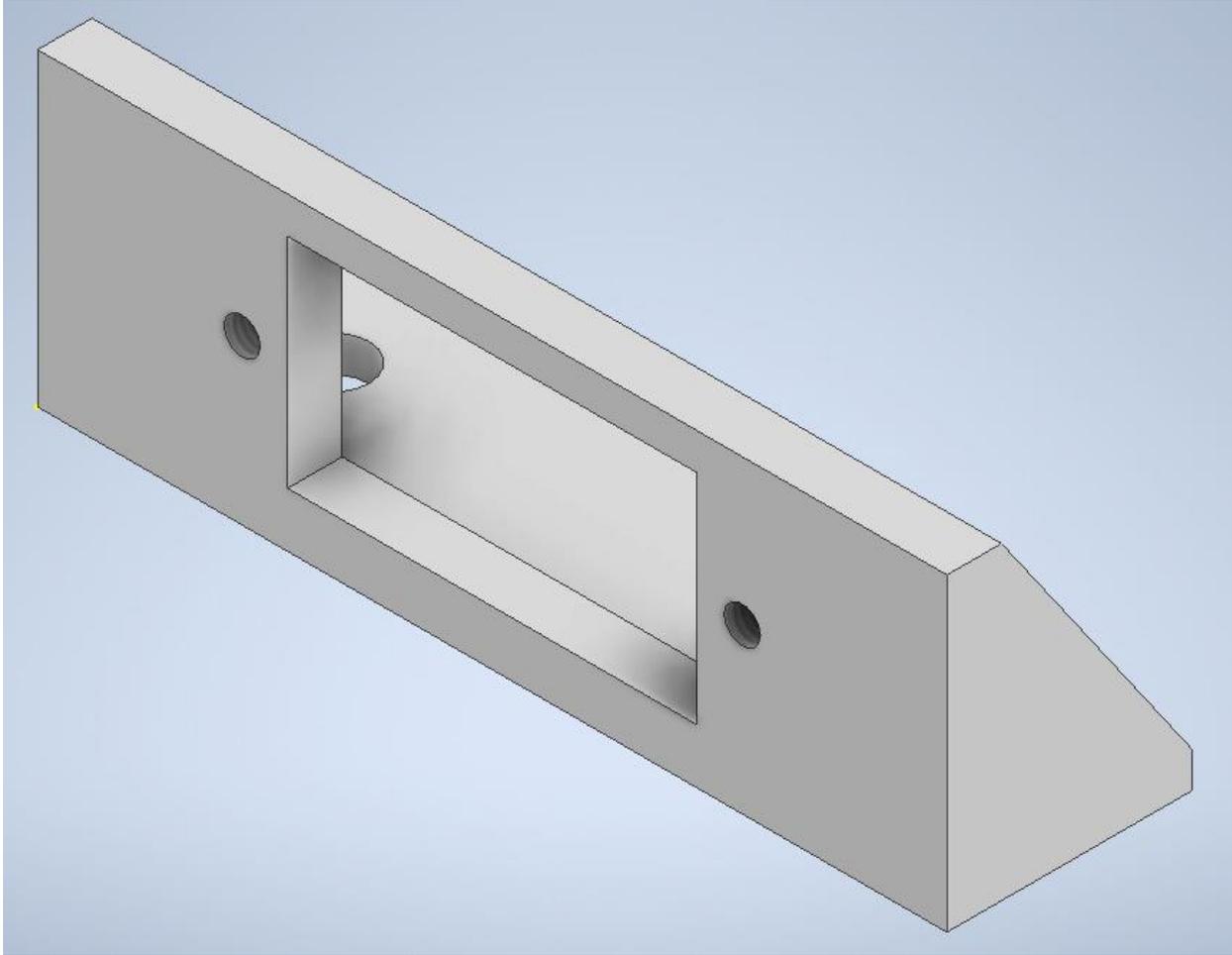


Figura 4.26. Soporte para servomotor MG90. Fuente propia

El diseño posee dos nervios para brindar estabilidad y resistencia al momento de colocar el servomotor, el cual se debe introducir la parte angosta dentro del agujero en una posición horizontal para lograr que el acople tenga el alcance suficiente para tener la capacidad de presionar una tecla al recibir una señal de parte del microprocesador.

Las dimensiones del agujero donde se coloca el servomotor y el de los tornillos se colocaron basados en la hoja de datos del MG90, donde se presenta todas las medidas de este dispositivo como se observa en la Figura 4.27.

MG90S
Metal Gear Servo

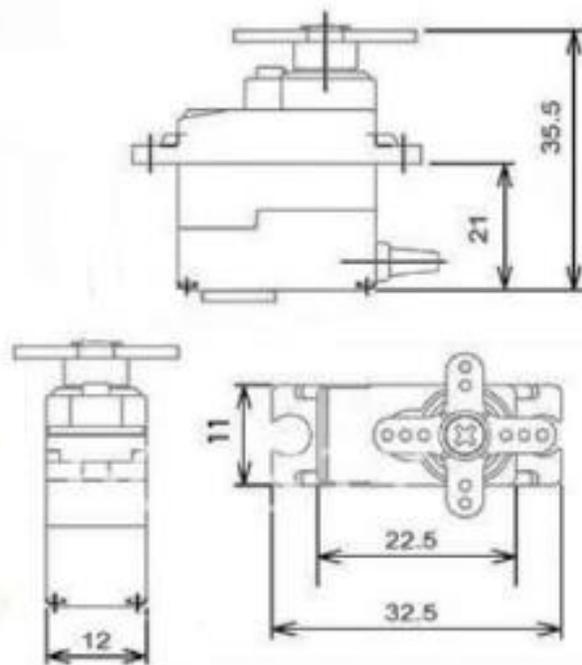


Figura 4.28. Dimensiones del Motor MG90. [40]

Con el soporte diseñado es necesario realizar una estructura para brindar estabilidad al momento de hacer uso del servo, además que permita la sujeción en una locación específica deseada para el usuario, no es parte de los requerimientos ya que la necesidad es la capacidad de presionar cualquier tecla, aun así, se decidió añadir esta posibilidad. Por este motivo se diseñó un eje que posee un agujero de oblongo en el cual se puede acoplar manualmente el soporte para el servomotor en la posición que se desee por medio de dos tornillos, en la Figura 4.28 se logra visualizar el diseño realizado del eje.

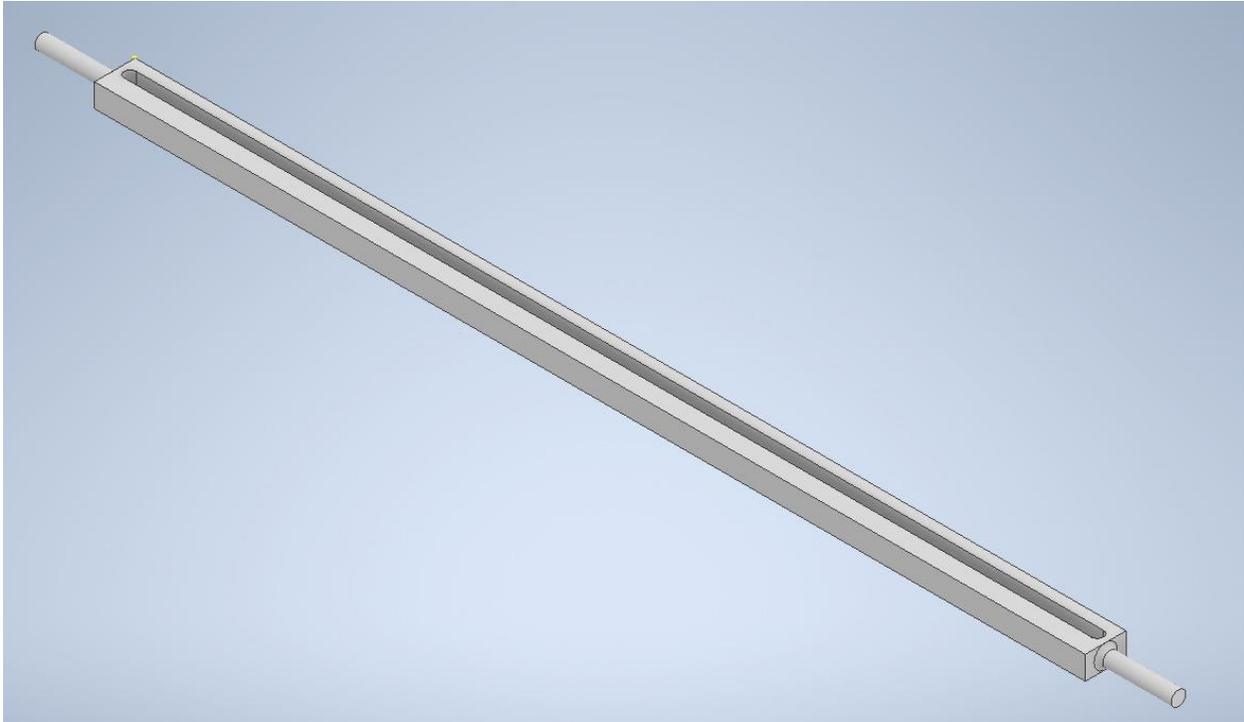


Figura 4.28. Eje de sujeción para el soporte del Servo Motor. Fuente Propia

También se tomó en cuenta que las computadoras a las que se le requiere realizar la medición pueden poseer distintas alturas, por este motivo a los soportes del eje se le agregaron distintas ranuras con el objetivo de poder colocar el eje en la ranura que se requiera y de esta manera controlar la altura en la que se va a encontrar el servomotor, además estos soportes poseen una base ancha que agrega peso y estabilidad al sistema. Véase Figura 4.29.

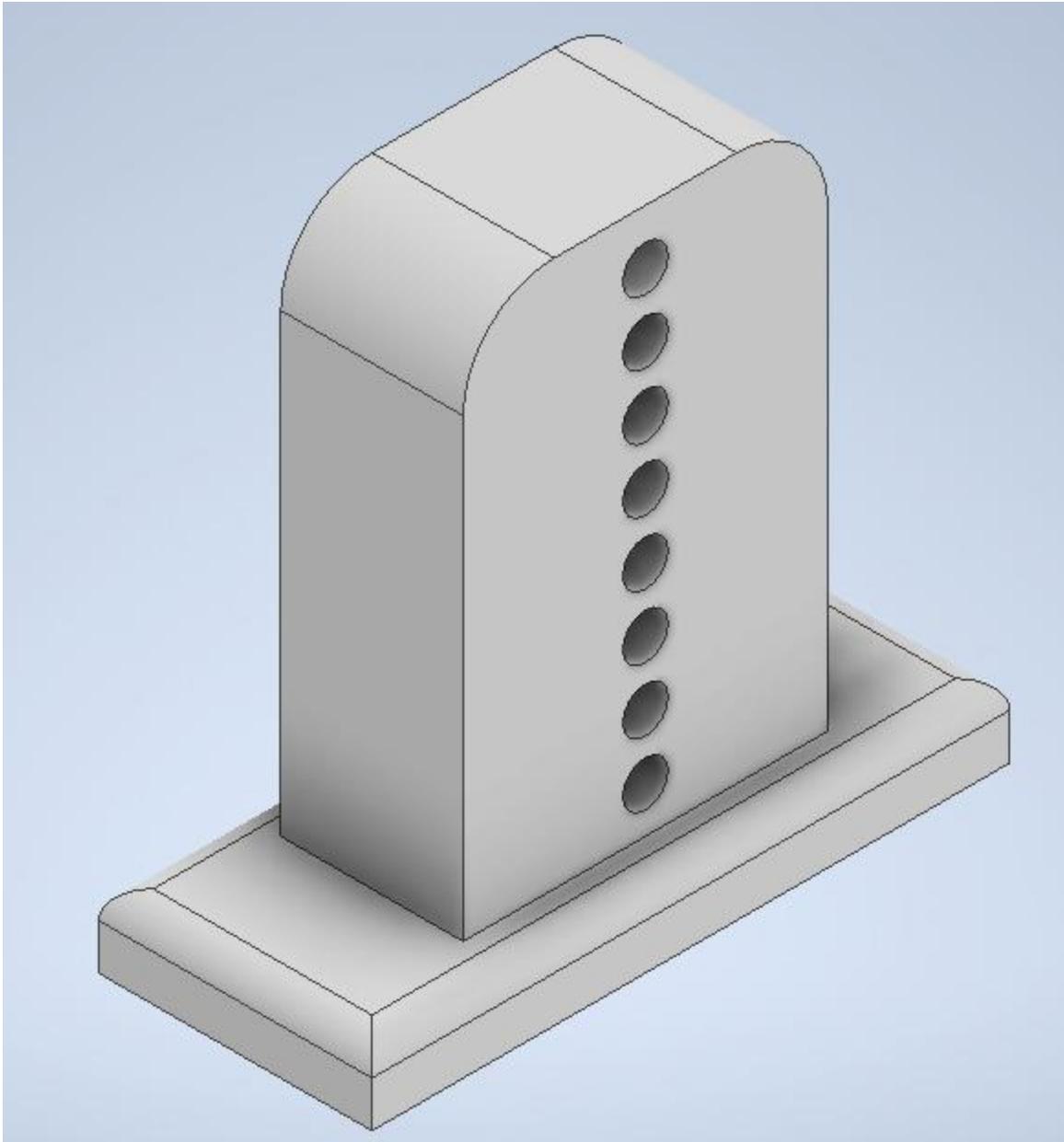


Figura 4.29. Soporte para el eje de sujeción. Fuente Propia

Con respecto al acople entre el eje de sujeción y el soporte de este se decidió hacerlo por medio de presión haciendo uso del ajuste de tolerancias basados en la tabla de la Figura 4.19 escogiendo el sistema forzado ligero debido a que esta manera se restringe cualquier movimiento de rotación que pueda presentar el eje, usando el sistema de agujero único las tolerancias corresponden a H7-j6, en la Figura 4.30 se muestra un diagrama de estas tolerancias.

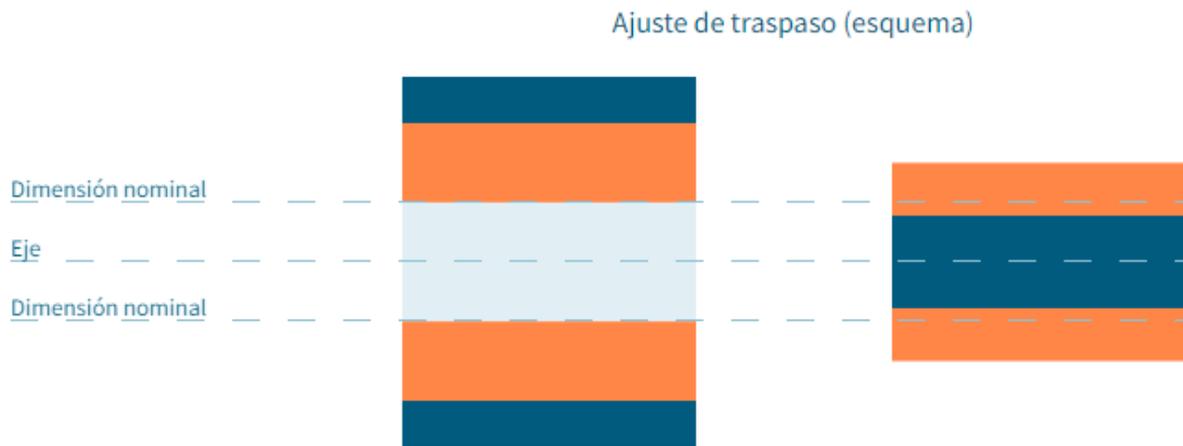


Figura 4.30. Esquema ajuste forzado ligero. [45]

En la Figura 4.31 se puede apreciar el sistema de soporte del servomotor ensamblado.

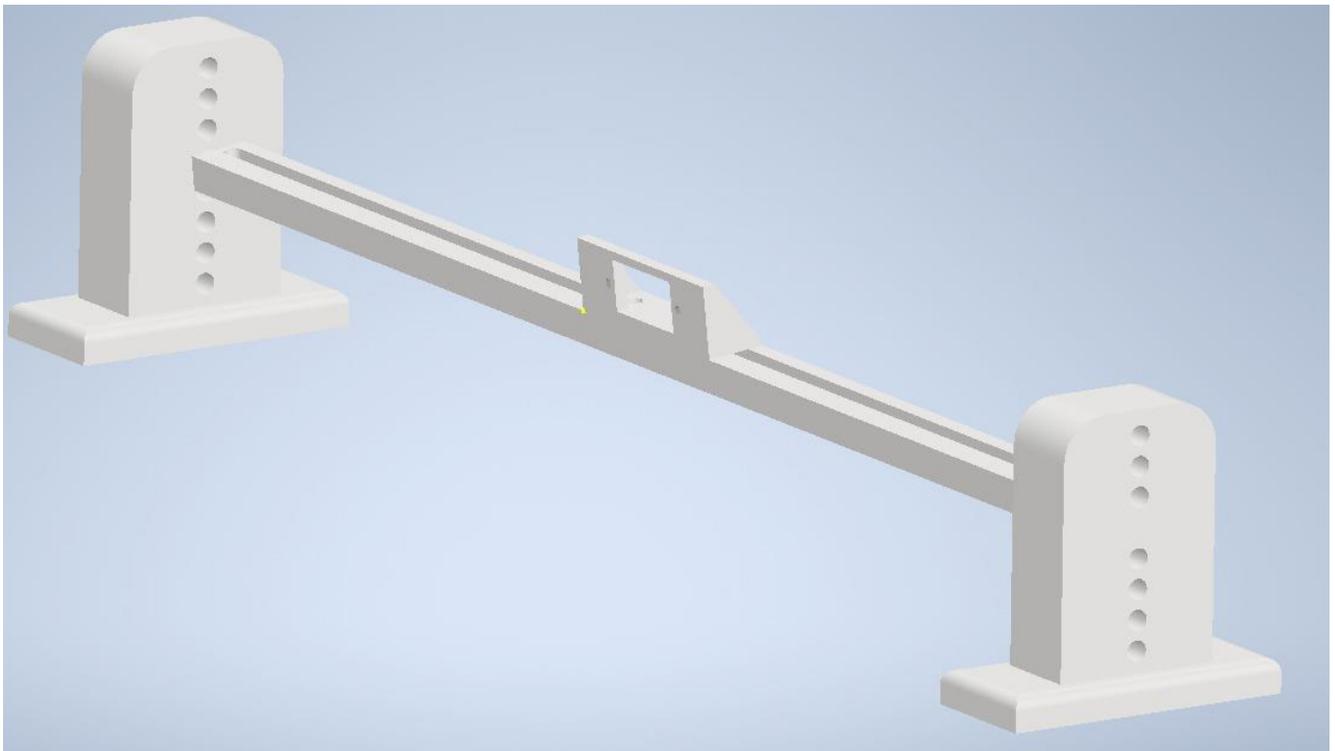


Figura 4.31. Ensamble soporte servomotor. Fuente Propia

Para efecto de prevención, en casos donde el alcance del acople que posee el servo no sea suficiente o la superficie de contacto debe ser menor para mayor precisión, se diseñó la parte que

se muestra en la Figura 4.32, la cual posee un orificio donde se puede introducir el acople que ya posee el Servo.

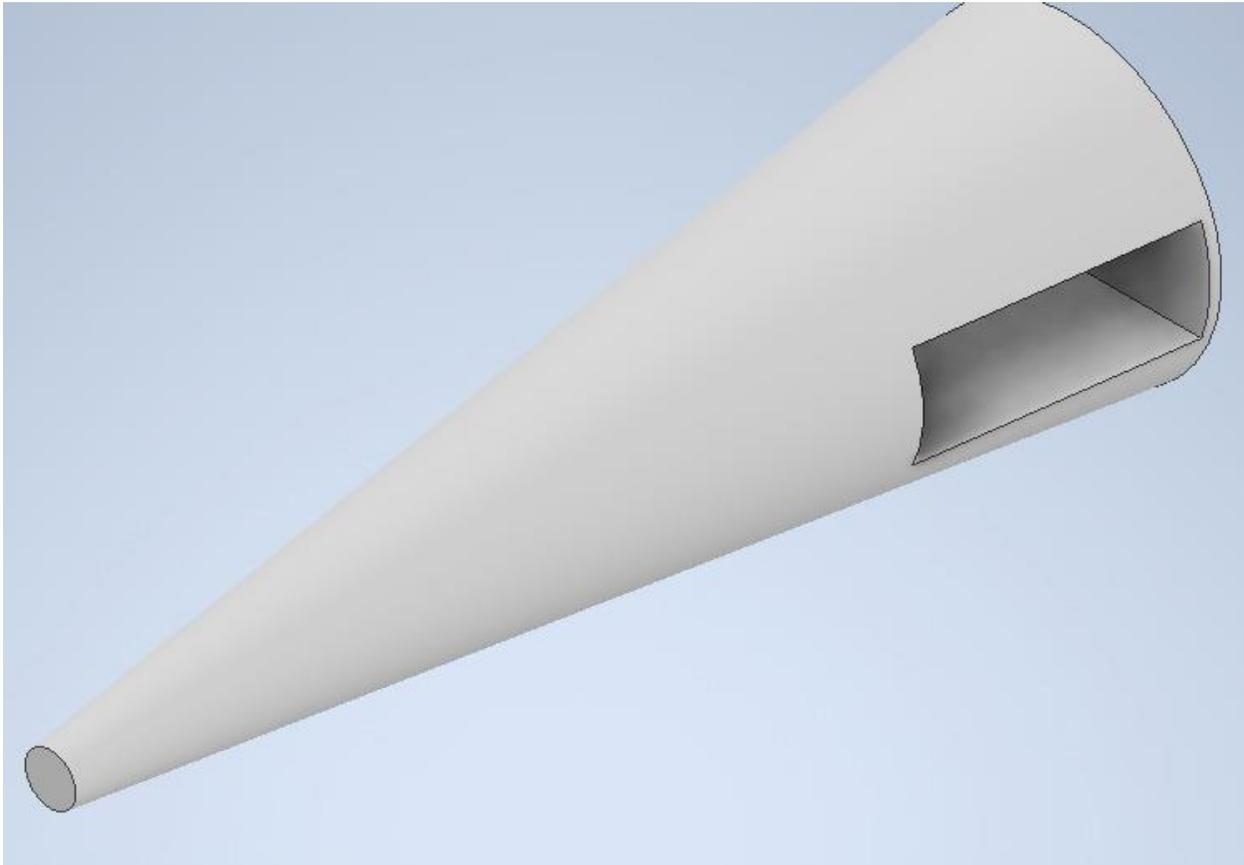


Figura 4.32. Punta para acople del servomotor. Fuente Propia

Cabe recalcar que para este proyecto se pretende realizar solo un prototipo como ya se ha mencionado, aun así, se establece el tipo de mecanizado necesario a utilizar para realizar las piezas al sistema final, esta decisión se realizó basado en las tablas de tolerancias IT, como en la que se observa en la Figura 4.33.

Grupos de Diámetros (mm)	CALIDADES																	
	IT 01	IT 0	IT 1	IT 2	IT 3	IT 4	IT 5	IT 6	IT 7	IT 8	IT 9	IT 10	IT 11	IT 12	IT 13	IT 14	IT 15	IT 16
$d \leq 3$	0.3	0.5	0.8	1.2	2	3	4	6	10	14	25	40	60	100	140	250	400	600
$3 < d \leq 6$	0.4	0.6	1	1.5	2.5	4	5	8	12	18	30	48	75	120	180	300	480	750
$6 < d \leq 10$	0.4	0.6	1	1.5	2.5	4	6	9	15	22	36	58	90	150	220	360	580	900
$10 < d \leq 18$	0.5	0.8	1.2	2	3	5	8	11	18	27	43	70	110	180	270	430	700	1100
$18 < d \leq 30$	0.6	1	1.5	2.5	4	6	9	13	21	33	52	84	130	210	330	520	840	1300
$30 < d \leq 50$	0.6	1	1.5	2.5	4	7	11	16	25	39	62	100	160	250	390	620	1000	1600
$50 < d \leq 80$	0.8	1.2	2	3	5	8	13	19	30	46	74	120	190	300	460	740	1200	1900
$80 < d \leq 120$	1	1.5	2.5	4	6	10	15	22	35	54	87	140	220	350	540	870	1400	2200
$120 < d \leq 180$	1.2	2	3.5	5	8	12	18	25	40	63	100	160	250	400	630	1000	1600	2500
$180 < d \leq 250$	2	3	4.5	7	10	14	20	29	46	72	115	185	290	460	720	1150	1850	2900
$250 < d \leq 315$	2.5	4	6	8	12	16	23	32	52	81	130	210	320	520	810	1300	2100	3200
$315 < d \leq 400$	3	5	7	9	13	18	25	36	57	89	140	230	360	570	890	1400	2300	3600
$400 < d \leq 500$	4	6	8	10	15	20	27	40	63	97	155	250	400	630	970	1550	2500	4000
	Ultrapre- -cisión		Calibre y piezas de gran precisión			Piezas o elementos destinados a ajustar						Piezas o elementos que no han de ajustar						

Figura 4.33. Tabla de calidades IT. [46]

Por medio de esta tabla también se verifica que la selección de ajuste es correcta, ya que si tomamos como ejemplo el acople entre el eje y soporte del eje la tolerancia es de 12 mm y el diámetro es de 5mm, lo que indica que se obtiene un IT7, lo cual entra dentro de la calificación de piezas o elementos destinados a ajustar, las medidas de las piezas y sus respectivos ajustes se pueden observar en Anexos, en la sección B1.

También se puede hacer uso de la tabla que muestra en la Figura 4.34 donde muestra el tipo de mecanizado que es posible usar según el IT y la rugosidad, por este motivo se decide utilizar el torno y la fresa, es decir, mecanizado por desprendimiento de viruta para realizar las piezas, la tolerancia utilizada es un valor conservador por lo que se podría reducir, importante mencionar que el laboratorio de Intel posee estas dos herramientas por lo que no sería necesario realizar una inversión.

CALIDADES SUPERFICIALES

Grupos de valores adoptados		Tolerancia	Superficie	Posible ejecución																			
				Superacabado	Lapidadada	Rectificado	Brochado	Mandrilado	Taladrado	Fresado	Cepillado	Torneado											
h	m	Designación convencional		Fino	Fino	Común	Acabado	Desbastado	Fino	Común	Fino	Común	Medio	Basto	Acabado	Desbastado	Acabado	Desbastado	Acabado finísimo	Acabado común	Desbastado		
0	a			0,04	A determinar según los casos		█	█															
0,04	a	0,06	█	█																			
0,06	a	0,1	█	█																			
0,1	a	0,16	█	█																			
0,16	a	0,25	IT5			█	█																
0,25	a	0,4				█	█																
0,4	a	0,6				█	█																
0,6	a	1	IT7				█	█															
1	a	1,6					█	█															
1,6	a	2,5	IT8					█	█														
2,5	a	4							█	█													
4	a	6	IT9						█	█													
6	a	10									█	█											
10	a	25	IT10											█	█								
25	a	63	-																				

Figura 4.34. Proceso de mecanizado según el IT. [46]

5 Resultados y análisis

5.1 Validación del sistema

En esta sección se muestra las estrategias de validación utilizadas para comprobar el funcionamiento del sistema con sus respectivos resultados. En la tabla 5.1 se encuentra en resumen de las pruebas realizadas.

Tabla 5.1. Resumen pruebas realizadas con el objetivo de verificar funcionalidad de la herramienta diseñada. Fuente propia

	Descripción de la Prueba
Prueba Estructura mecánica	Para esta prueba se pretende comprobar la rigidez y durabilidad de cada una de las partes que conforman la estructura diseñada, esta prueba se realiza por medio del software Inventor donde se colocan las cargas a las cuales está sometido el soporte y se verifica que el factor de seguridad sea mayor a 3 como mínimo.
Prueba verificación de estados	Con el objetivo de comprobar la buena funcionalidad de la capacidad del instrumento de verificar los estados se realiza 30 mediciones y se verifica el porcentaje de error el cual debe ser menor a 5%.
Prueba capacidad de medición del tiempo	Utilizando el prototipo se realiza 30 medidas a una computadora la cual haya sido medida por el método anterior, lo cual es el valor de referencia utilizado, el porcentaje de error con el instrumento diseñado deber ser menor a 5%.
Prueba verificación de la precisión en la medición	Con los mismos valores utilizados para la prueba anterior se realiza un estudio tipo 1 mediante el software minitab donde se extrae el valor Cg el cual indica la replicabilidad de la herramienta, además la máquina debe poseer una precisión de al menos 200 ms, por medio de esta prueba mencionada también se comprueba si el instrumento es útil al momento de tomar medidas con la precisión requerida.
Prueba de la repetibilidad	El objetivo de esta prueba es validar la repetibilidad de las medidas dentro del rango requerido el cual era de 200ms, para este prueba se asigna un valor binario entre aceptado y no aceptado a todos los valores, siendo aceptado si la medición se encuentra dentro de la incertidumbre esperada, para caso de validación el rango será de 0.03 que es la capacidad de la herramienta en lugar de 200ms que es el requerido, ya que se quiere validar la funcionalidad de la herramienta y este valor se encuentra dentro del rango requerido. Al final se verifica la cantidad de datos aceptados y se calcula el porcentaje de ellos, se pretende que este valor sea mayor a 90%

Se realizó un prototipo del sistema electromecánico mostrado en la sección anterior para realizar las pruebas de validación, este prototipo se puede observar en la Figura 5.1



Figura 5.1. Prototipo utilizado para realizar las mediciones de validación y confirmar funcionamiento de los distintos sistemas integrados. Fuente propia.

El criterio para decidir la cantidad de mediciones a realizar se basó en el resultado de la desviación estándar, inicialmente se tomó una muestra de 10 mediciones, luego de 20 y finalmente de 30, en la Tabla 5.6 se pueden observar estos datos utilizados, el criterio de parada consistió en tomar medidas hasta que variación en el cálculo de una desviación estándar a otra no fuera significativa, consideran no significativa como una diferencia menor a 5%, en la Tabla 5.2 se puede observar los valores de desviación estándar obtenidos.

Tabla 5.2. Cálculo de la desviación estándar según la cantidad de medidas realizadas. Fuente propia

Número de mediciones	Desviación estándar obtenida	Porcentaje de diferencia con el antecesor
5	0.268	---
10	0,0212	21,37%
20	0,0179	15,57%
30	0,0187	4.47%

Otra guía utilizada para definir que el valor obtenido de desviación estándar con 30 medidas es válido corresponde a la regla empírica, la cual indica que el 99,7% de los valores se deben ubicar a no más de tres desviaciones estándar. Las mediciones realizadas se encuentran en la Tabla 5.6 y el promedio de todas ellas en la tabla 5.7, el promedio de los valores y el cálculo de la sumatoria y resta de las 3 desviaciones estándar sobre este valor se coloca a continuación, así también como el valor máximo y mínimo que se pueden encontrar dentro de los resultados de la Tabla 5.6.:

$$0,0187 * 3 = 0.0561$$

$$\text{Promedio de las mediciones} = 1,02s$$

$$\text{Promedio} + 3\sigma = 1,0761s$$

$$\text{Promedio} - 3\sigma = 0,963s$$

$$\text{Valor máximo obtenido} = 1,03s$$

$$\text{Valor mínimo obtenido} = 0.97s$$

Con los cálculos colocados anteriormente se comprueba que el 100% de los datos obtenidos se encuentra dentro del rango de las 3 desviaciones estándar, debido a que el resultado de la sumatoria es mayor al valor máximo que se obtuvo en las mediciones y la resta del promedio es un valor menor al mínimo de los 30 resultados obtenidos.

5.1.1 Prueba Estructura mecánica

El valor requerido de factor de seguridad escogido corresponde a mayor de 3, esto basándose en la Directiva 98/37/CE [48] la cual determina los requisitos esenciales que debe cumplir la fabricación y comercialización de las máquinas y los componentes de seguridad. La norma indica que debe ser mayor a 1.25 pero para ser conservadores y garantizar la durabilidad y calidad de la herramienta a diseñar se aumenta este valor a 3.

Para realizar las simulaciones por software es necesario obtener el valor de la carga a la cual es sometido la estructura, esta carga corresponde al peso de las distintas partes que componen la estructura además de la cámara. El peso de la cámara se sabe que es de 3g como mencionaba la Tabla 4.4, el siguiente paso corresponde a identificar el peso de cada módulo.

Para esto se utiliza la densidad del aluminio, la cual corresponde a $2.7 \frac{g}{cm^3}$, se asume que cada parte de la estructura es un bloque sin agujeros y de esta manera se realizan los cálculos del peso de las piezas, este es un valor conservador ya que se asume que existe más material del que verdaderamente hay, por lo tanto, se le agrega un mayor valor de masa lo que implica un mayor peso.

La primera pieza corresponde a la del brazo intermedio la cual posee unas dimensiones generales de 80 x 20 x 15 mm como se observa en la Figura 5.1.

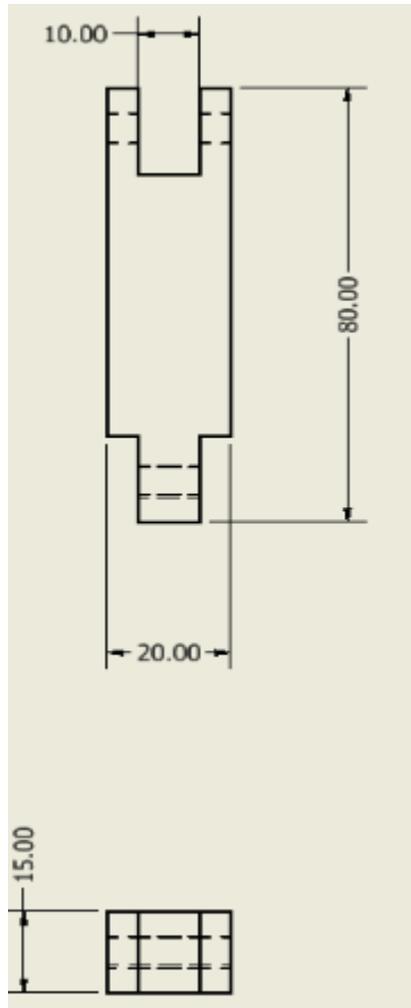


Figura 5.1. Dimensiones generales del brazo medio en mm. Fuente propia.

Con estas medidas se calcula el volumen y el peso del bloque de la siguiente manera:

$$80 * 15 * 20 = 24000mm^3$$

$$= 24cm^3$$

$$2.7 \frac{g}{cm^3} * 24 cm^3 = 64.8 g$$

$$0.0648Kg * 9,81 \frac{m}{s^2} = 0.6357 N$$

Se procede a realizar el mismo procedimiento con la pieza final del soporte, cuyas medidas se observan en la Figura 5.2.

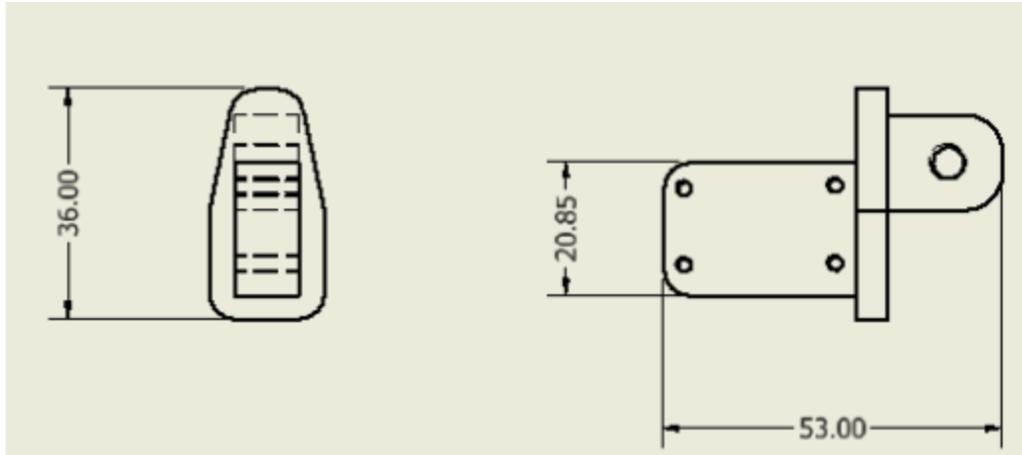


Figura 5.2. Dimensiones generales de la pieza final del soporte en mm. Fuente propia.

$$36 * 53 * 18 = 34344 \text{mm}^3$$

$$= 34.344 \text{cm}^3$$

$$2.7 \frac{\text{g}}{\text{cm}^3} * 34.344 \text{cm}^3 = 92.729 \text{g}$$

$$0.0927 \text{Kg} * 9.81 \frac{\text{m}}{\text{s}^2} = 0.9094 \text{N}$$

Así sucesivamente se repite el procedimiento para las piezas que conforman la estructura de soporte del servomotor, las dimensiones generales se aprecian en las Figuras 5.3 y 5.4.

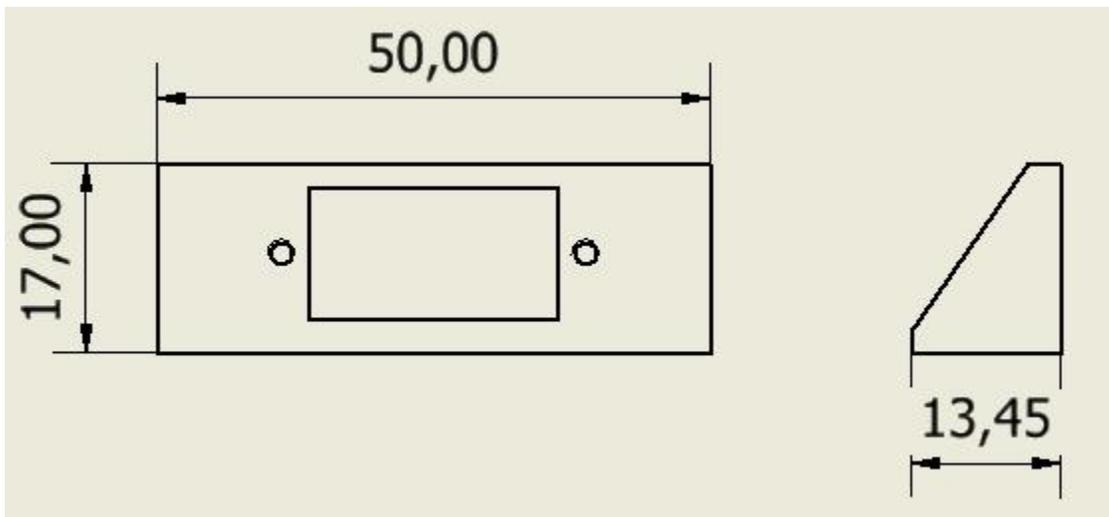


Figura 5.3. Dimensiones generales del soporte del Servomotor en mm. Fuente propia.

$$50 * 17 * 13,45 = 11390mm^3$$

$$= 11,39cm^3$$

$$2.7 \frac{g}{cm^3} * 11,39 cm^3 = 30,75 g$$

$$0.03075Kg * 9,81 \frac{m}{s^2} = 0.3017 N$$

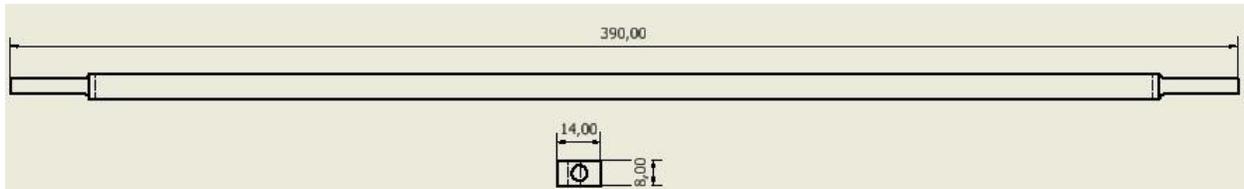


Figura 5.4. Dimensiones generales del eje de la estructura del servomotor. Fuente propia.

$$14 * 8 * 390 = 43680mm^3$$

$$= 43,68cm^3$$

$$2.7 \frac{g}{cm^3} * 43,68 cm^3 = 117,94 g$$

$$0.11794Kg * 9,81 \frac{m}{s^2} = 1,157 N$$

Con los pesos definidos es posible realizar las simulaciones con las cargas correspondientes, el primer paso es elegir el material, como ya se mencionó se selecciona Aluminio 6061, de esta manera el simulador posee todas las propiedades necesarias para los cálculos como el módulo de elasticidad, límite elástico y resistencia a la tensión. En la Figura 5.5 se observa la escogencia del material en el software Inventor.

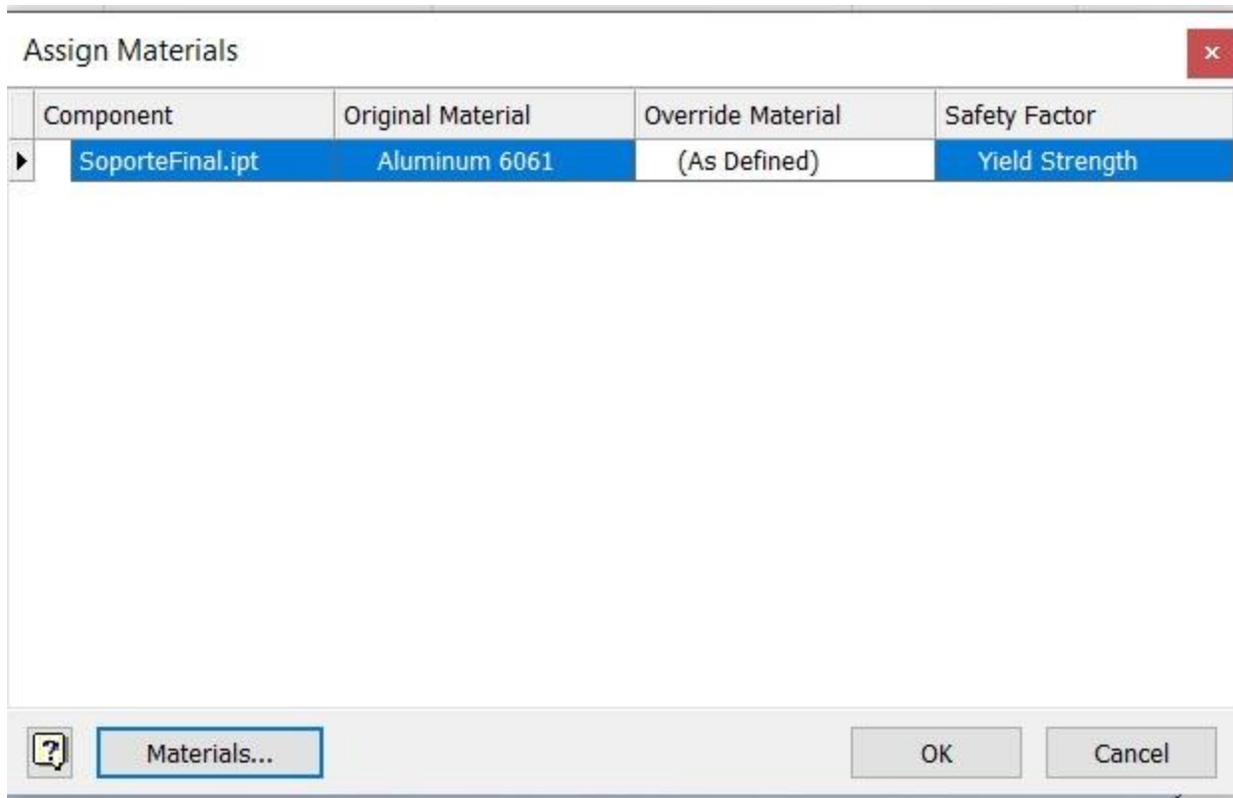


Figura 5.5. Selección del material en el software inventor. Fuente propia.

El siguiente paso corresponde a seleccionar la restricción de movimientos que va a poseer la pieza, para este caso como se quiere verificar que no haya ningún fallo en el material y asumiendo un caso estático, ya que el soporte luego de haber definido a la altura con la cual es utilizado no se encuentra en movimiento al momento de soportar el motor. Se asume una restricción de movimiento en todos los sentidos desde la posición donde se encontraría la unión. En la Figura 5.6 se demuestra la escogencia y posición de esta restricción.

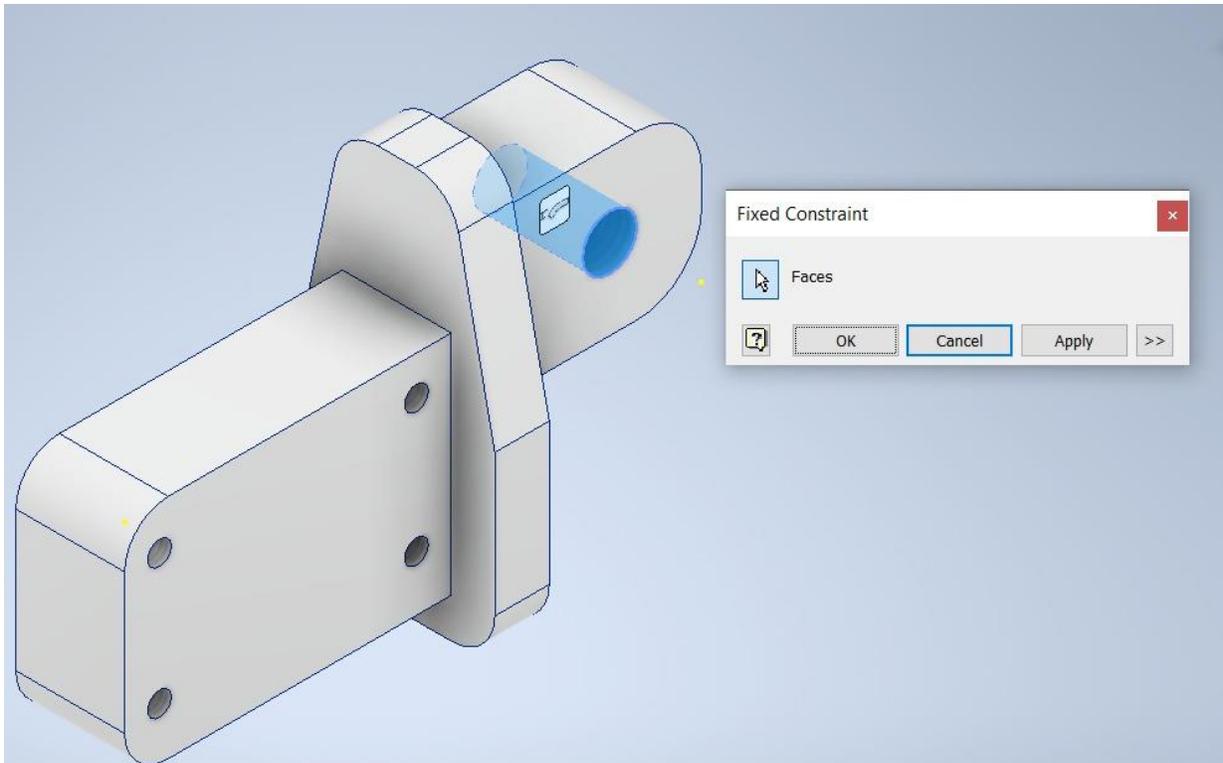


Figura 5.6. Asignación de la restricción de movimiento. Fuente propia.

Por último, se agregan las cargas que generan los esfuerzos de aplastamiento, por este motivo era importante el cálculo de los pesos de cada pieza. En las Figuras 5.7 y 5.8 se puede visualizar el valor dado a cada carga en N, una carga corresponde al peso del módulo de la cámara y la otro al propio peso de la pieza.

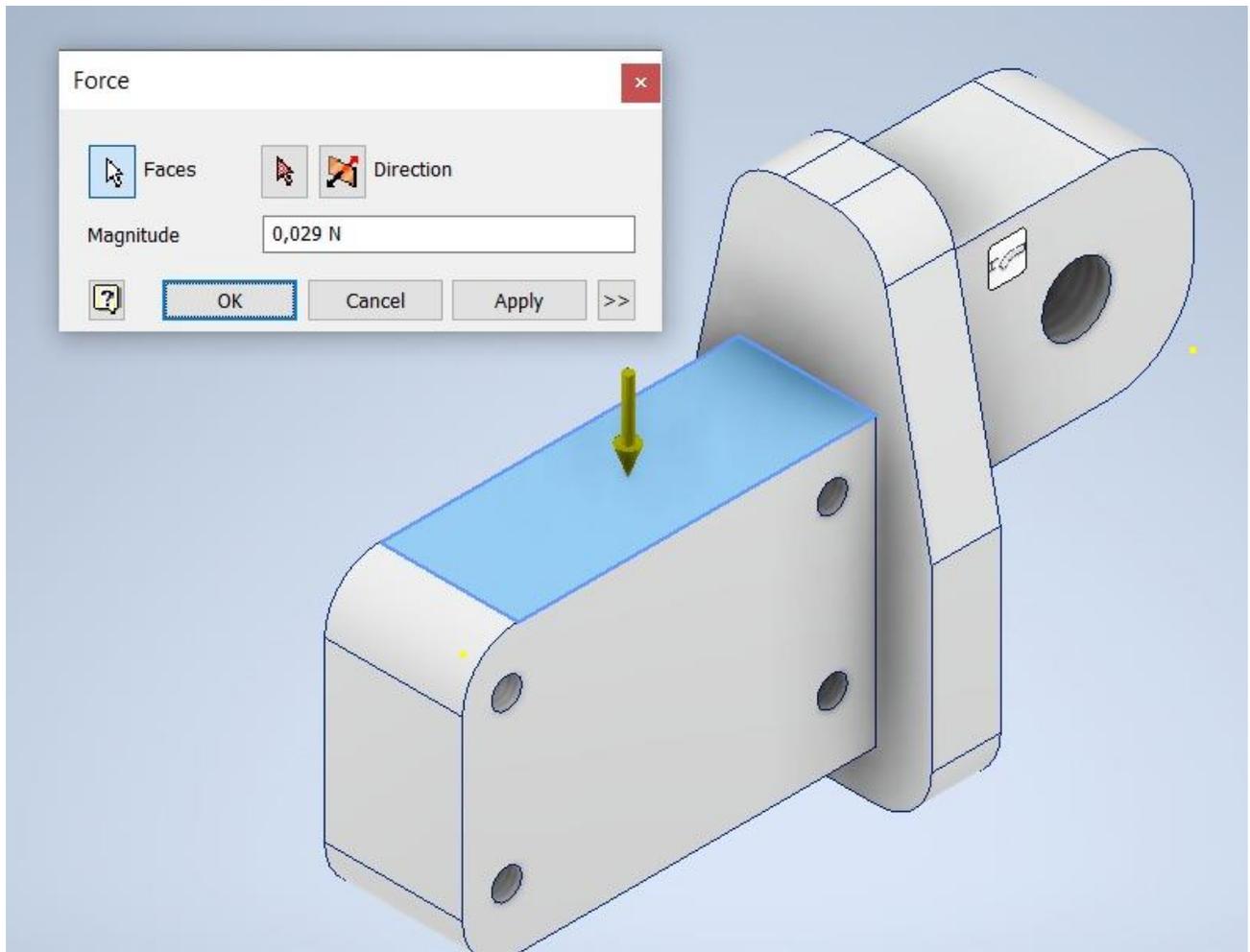


Figura 5.7. Asignación de la carga debido al peso de la cámara. Fuente propia.

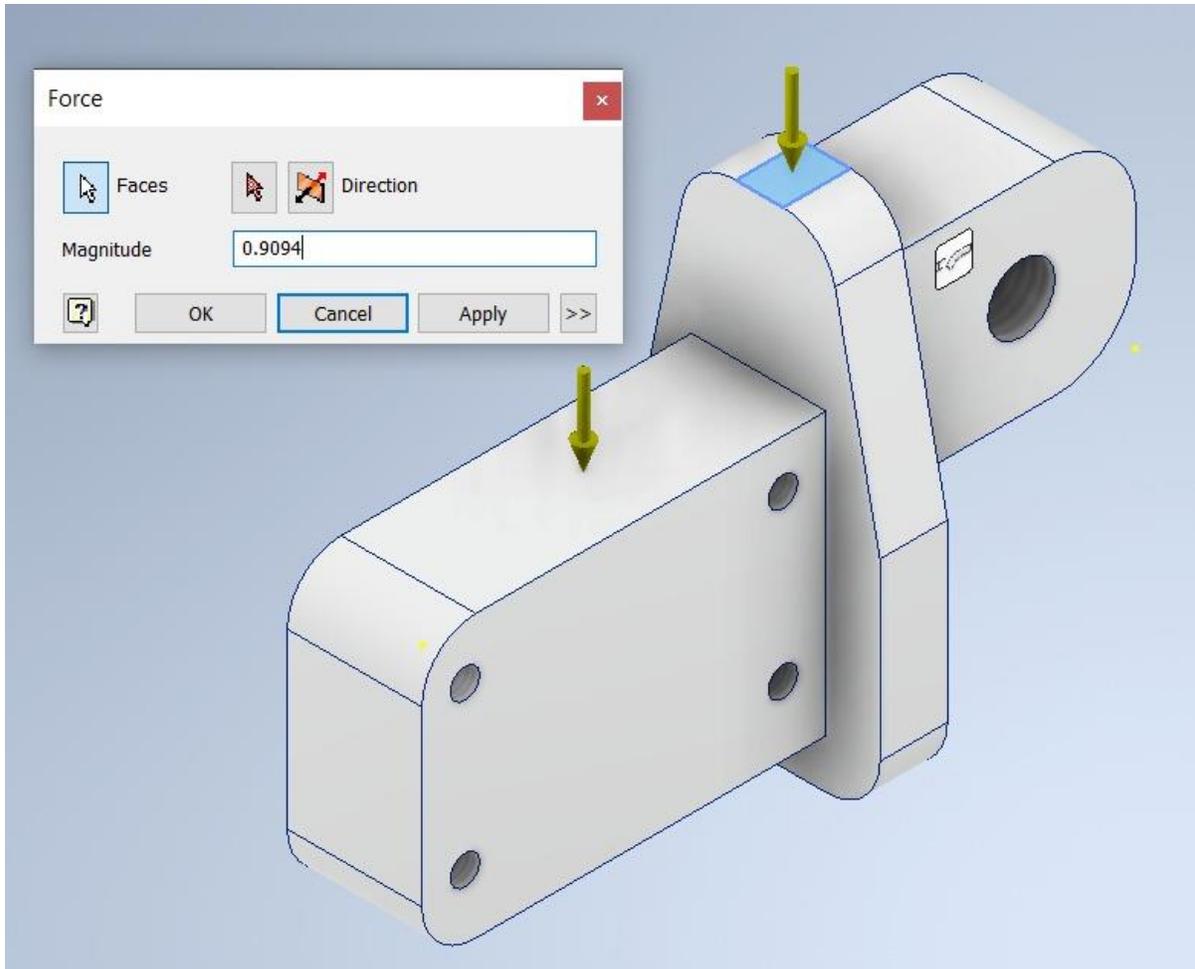


Figura 5.8. Asignación de la carga debido al peso de la propia pieza de la estructura. Fuente propia.

Con todos los pasos anteriores realizados es posible ejecutar la simulación y verificar el valor de seguridad que posee la estructura, en la Figura 5.9 se muestra el resultado de la simulación para la pieza que sostiene el motor.

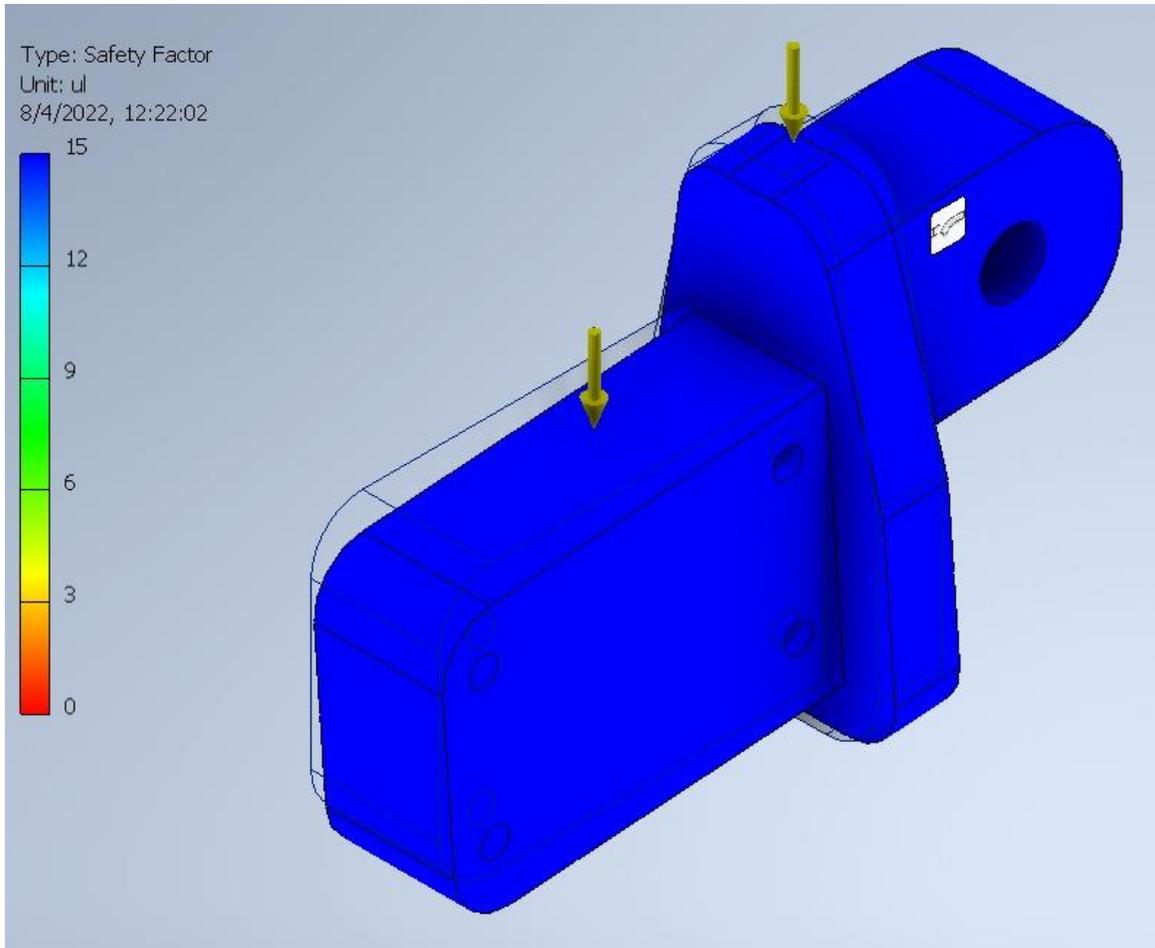


Figura 5.9. Factor de seguridad obtenido del análisis estático mediante software Inventor de la pieza que soporta la cámara. Fuente propia.

Como la estructura diseñada es modular y consta de distintas piezas se procede a realizar los mismos pasos para cada una de sus partes, véase las Figuras 5.10,5.11,5.12,5.13,5.14,5.15,5.16,5.17,5.18,5.19,5.10,5.21,5.22,5.23,5.24,5.25,5.26,5.27.

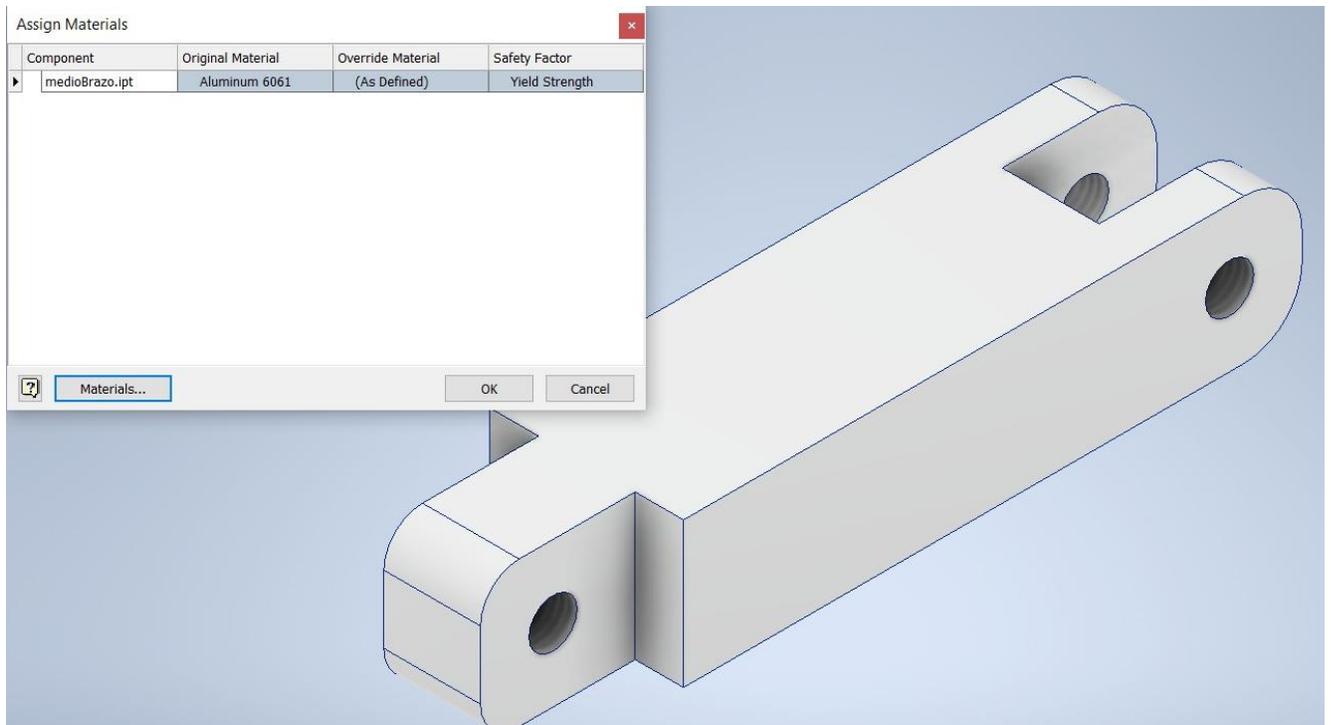


Figura 5.10. Asignación del material a la parte media de la estructura. Fuente propia.

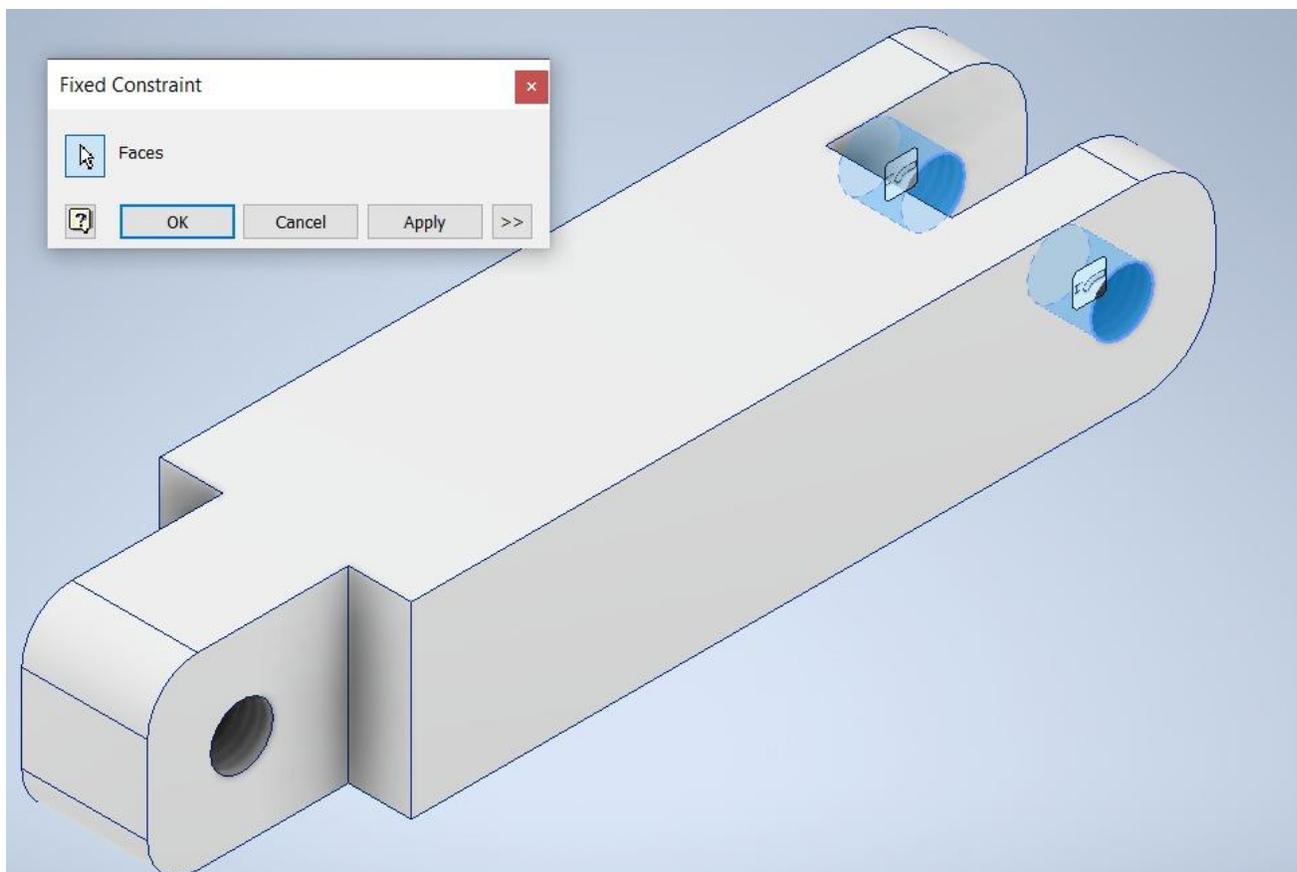


Figura 5.11. Asignación de las restricciones de movimiento a la parte media de la estructura.
Fuente propia.

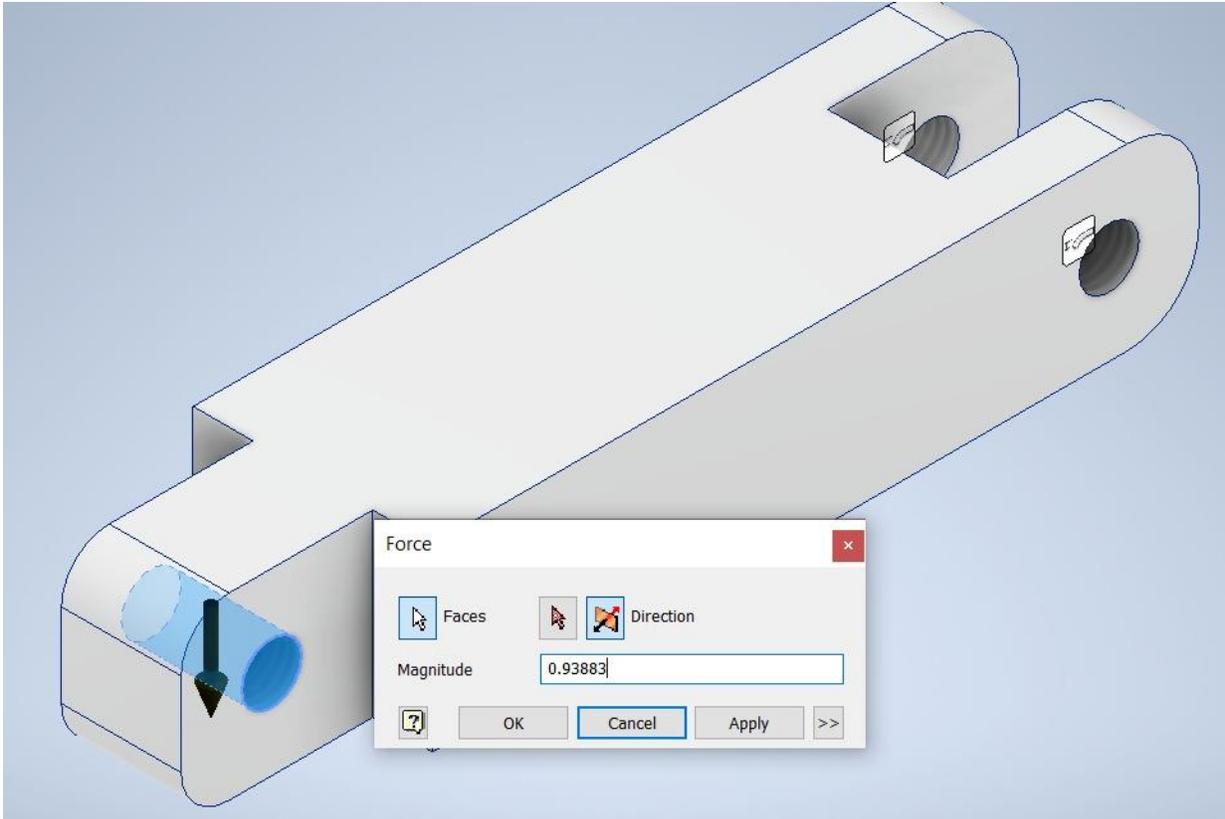


Figura 5.12. Asignación de la primera carga correspondiente al peso de la cámara y al de la pieza acoplada. Fuente propia.

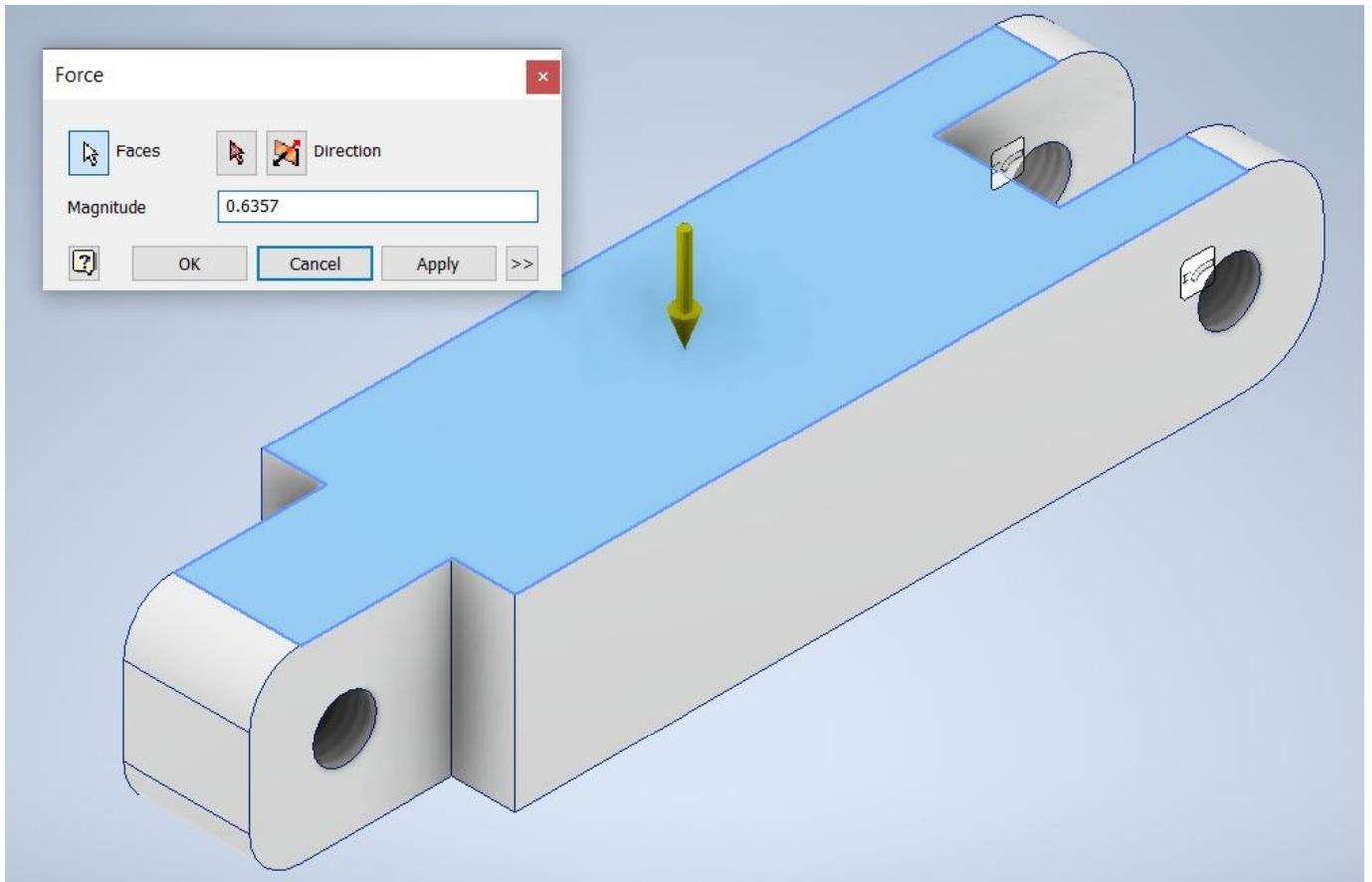


Figura 5.13. Asignación de la carga correspondiente al peso de la propia pieza. Fuente propia.

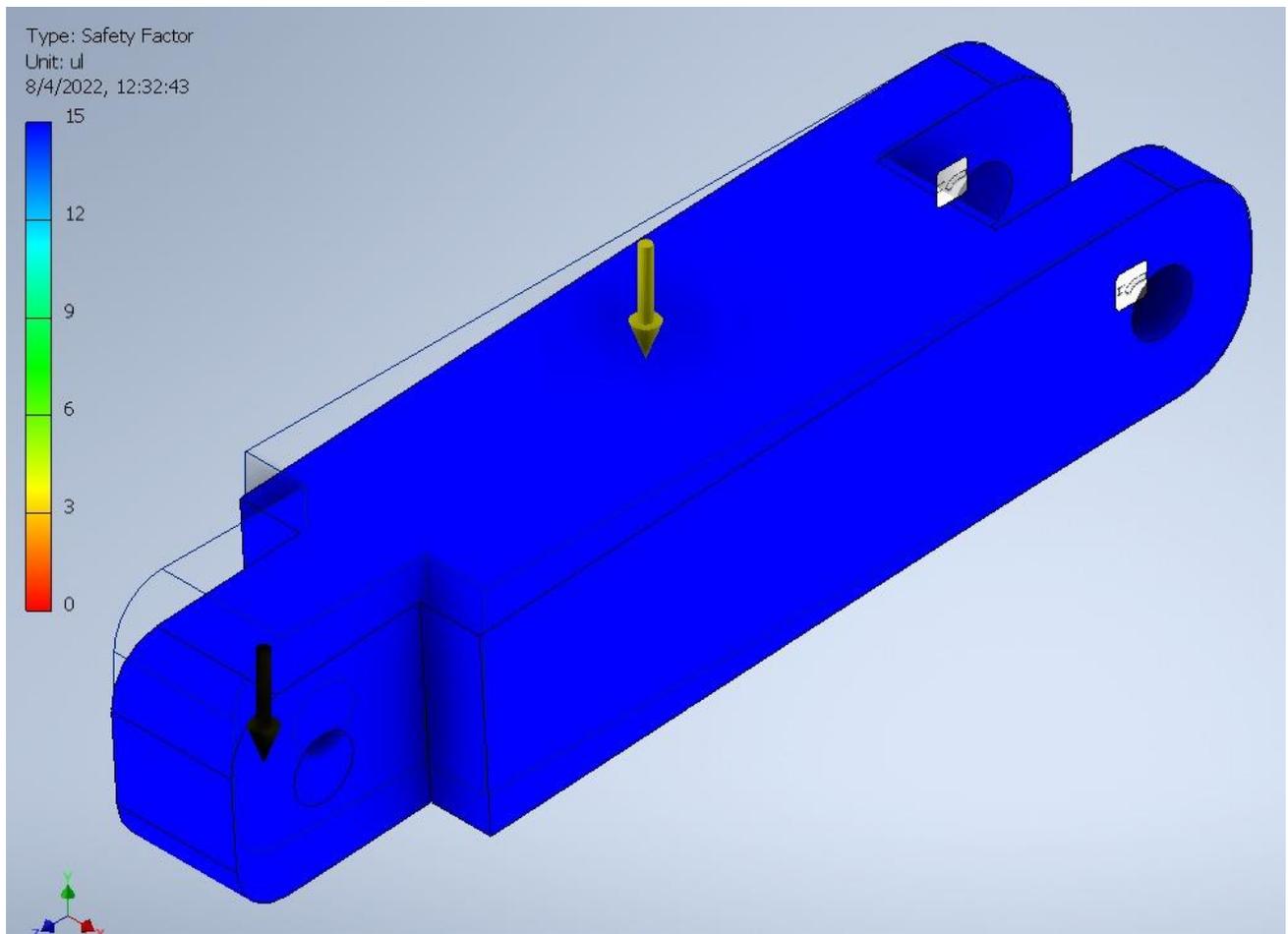


Figura 5.14. Resultados de la simulación del análisis estático por medio del software Inventor de la parte media de la estructura. Fuente propia.

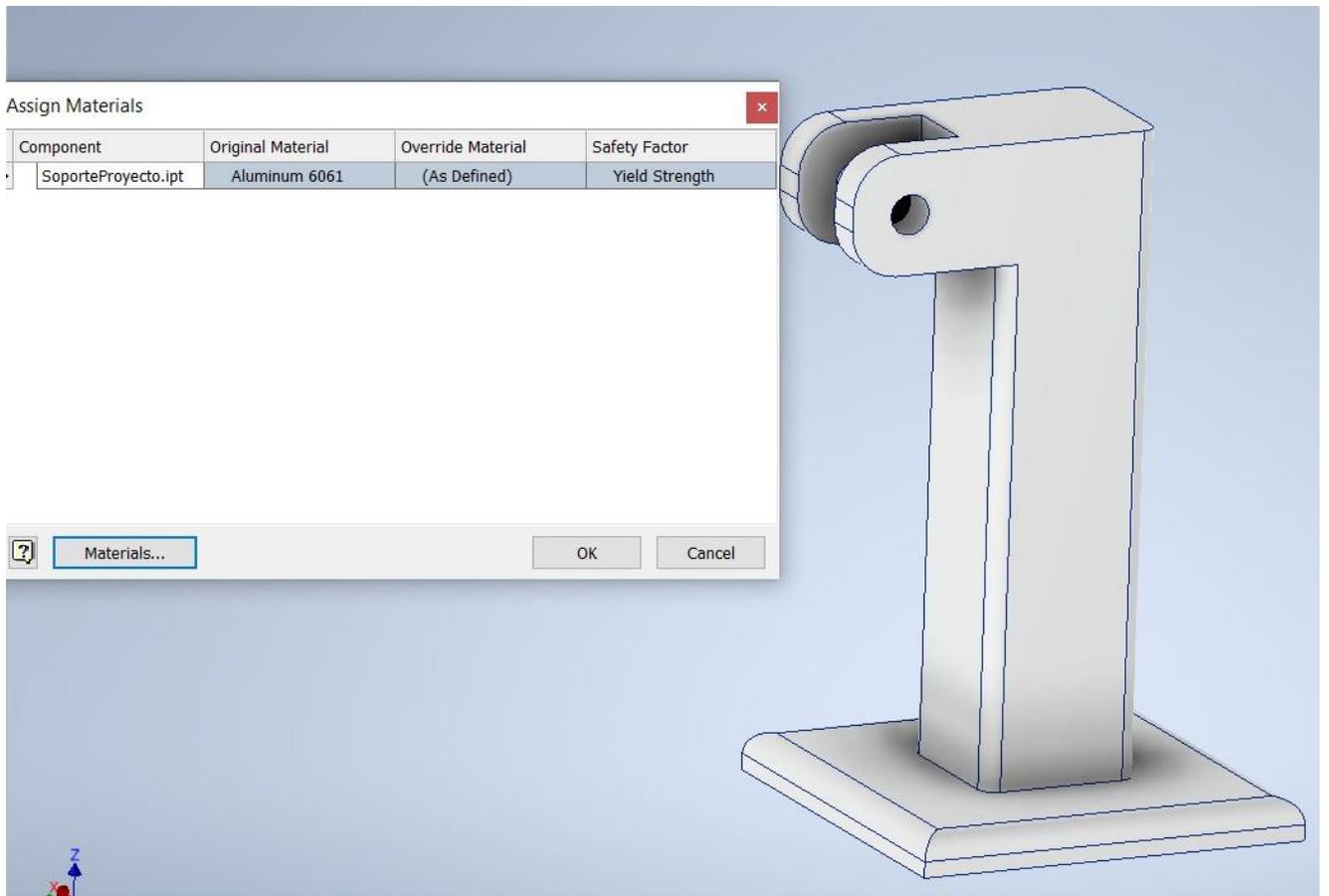


Figura 5.15. Asignación del material a la base de la estructura. Fuente propia.

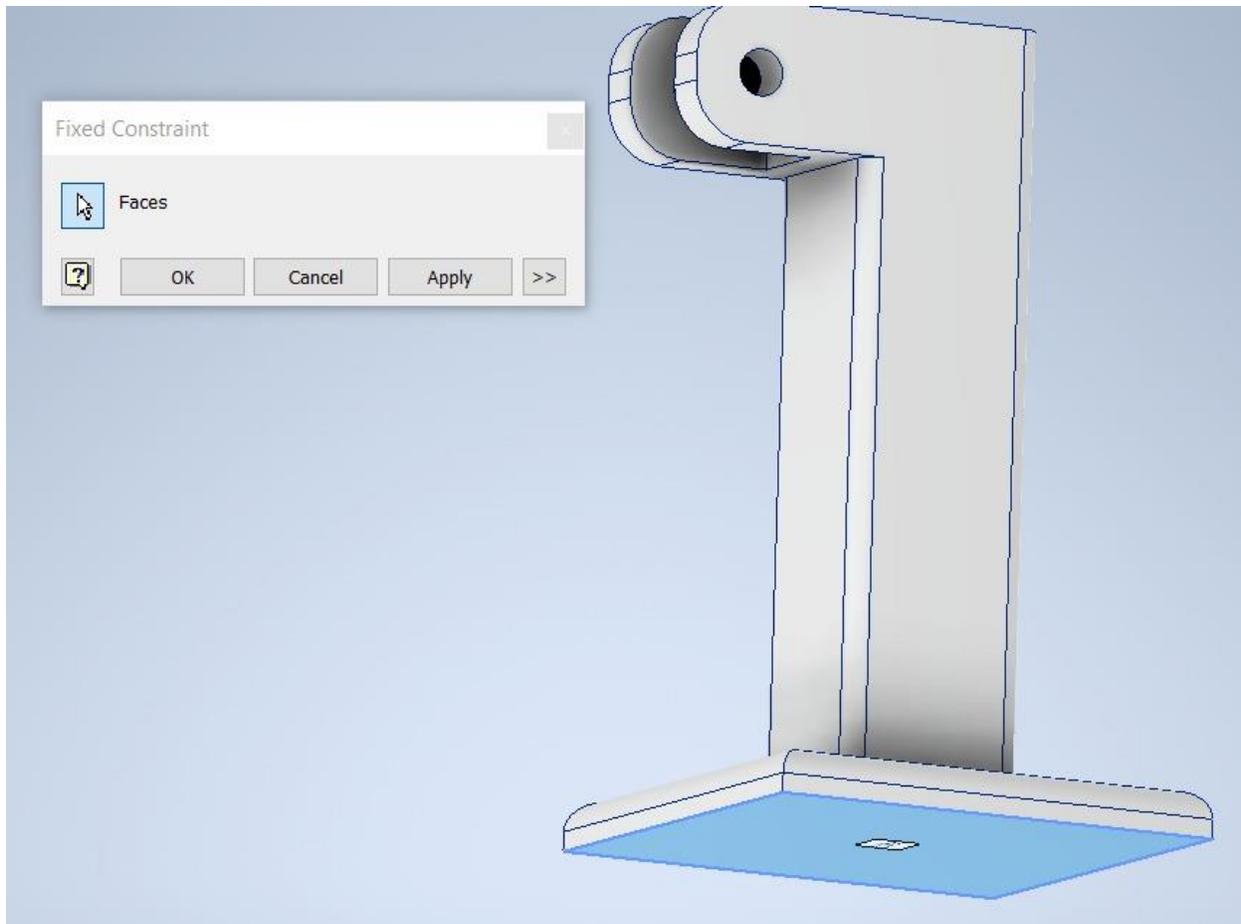


Figura 5.16. Asignación de las restricciones de movimiento a la base de la estructura. Fuente propia.

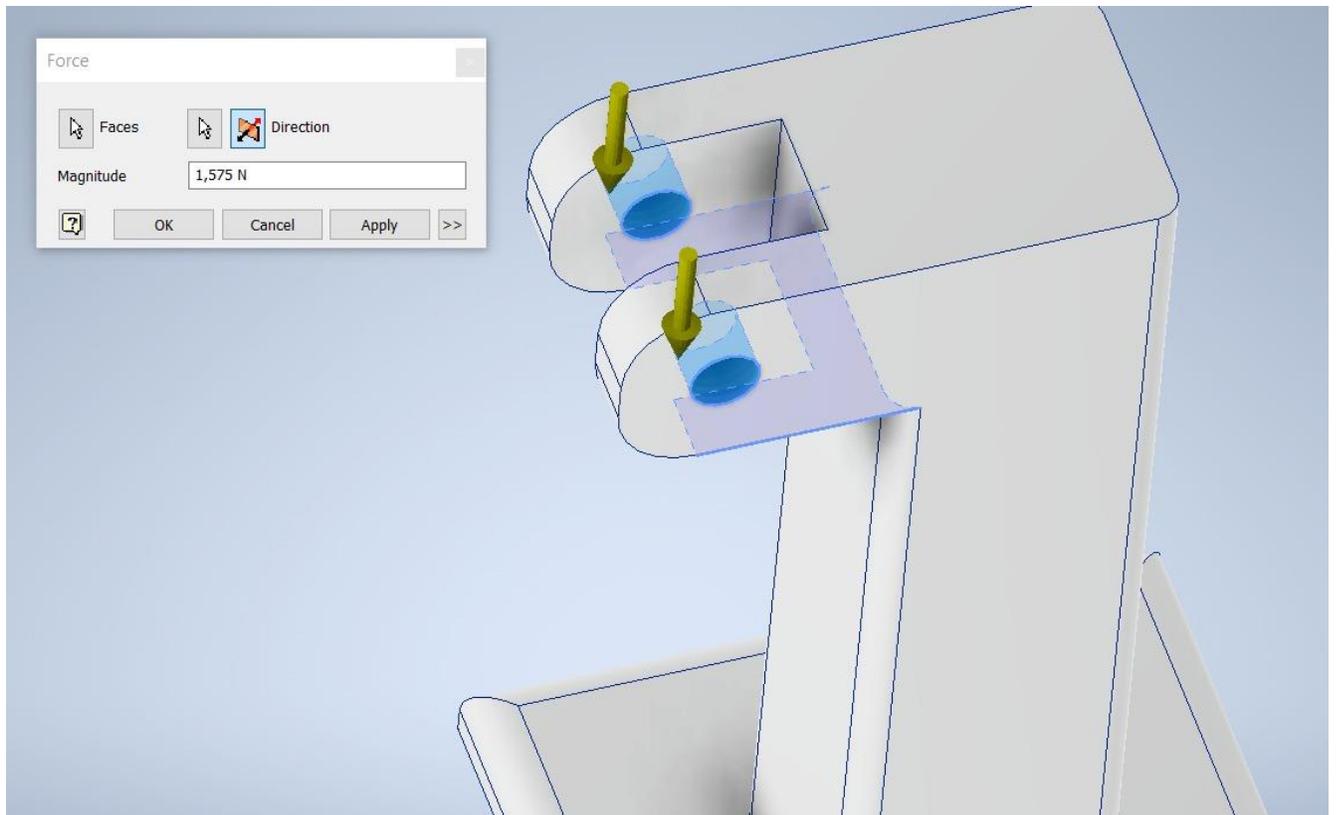


Figura 5.17. Asignación de la carga a la base de la estructura. Fuente propia.

Es importante aclarar que el valor utilizado para la carga que se observa en la Figura 5.15 es la sumatoria de todas las cargas anteriores.

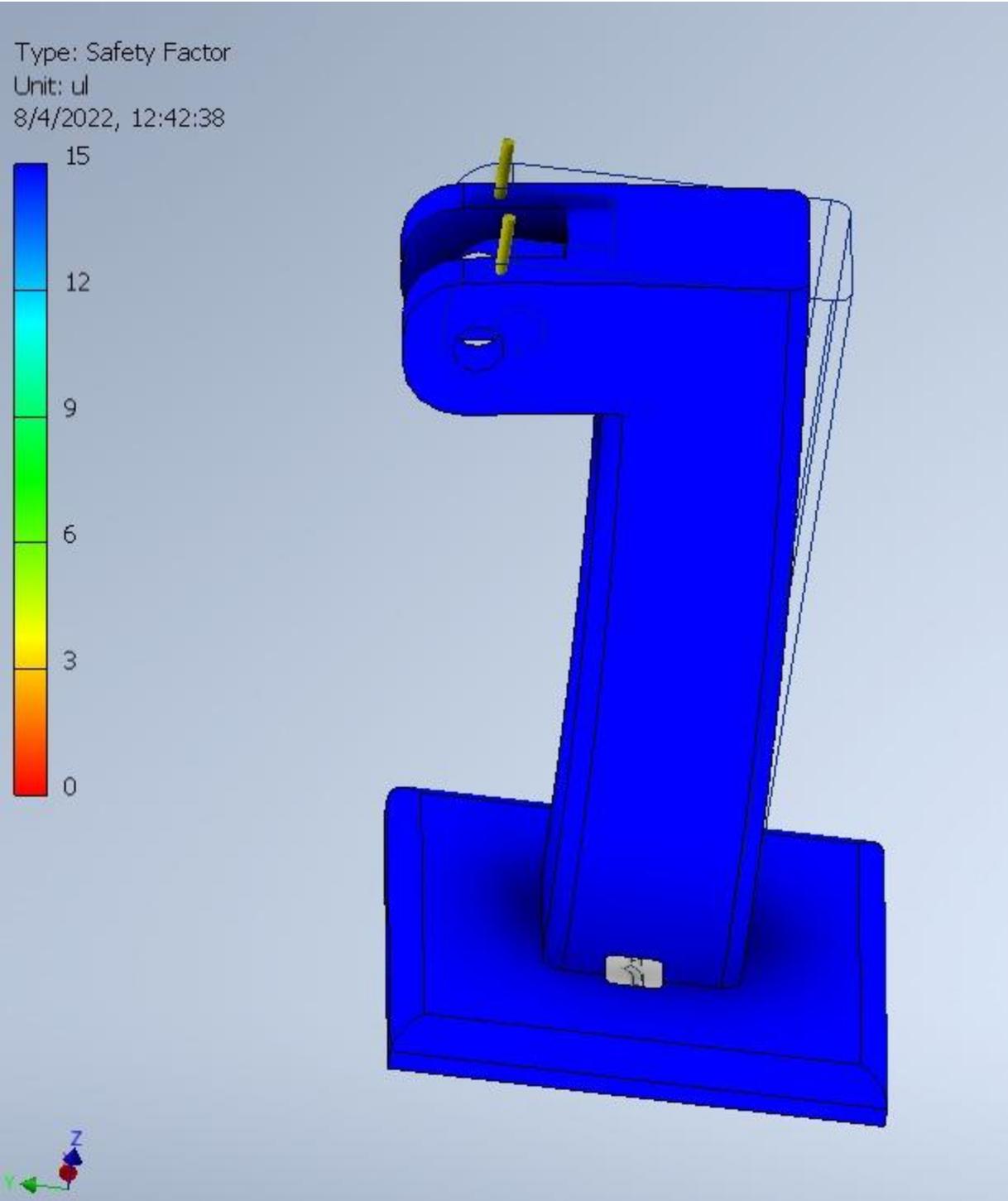


Figura 5.18. Resultados de la simulación del análisis estático por medio del software Inventor de la base de la estructura. Fuente propia.

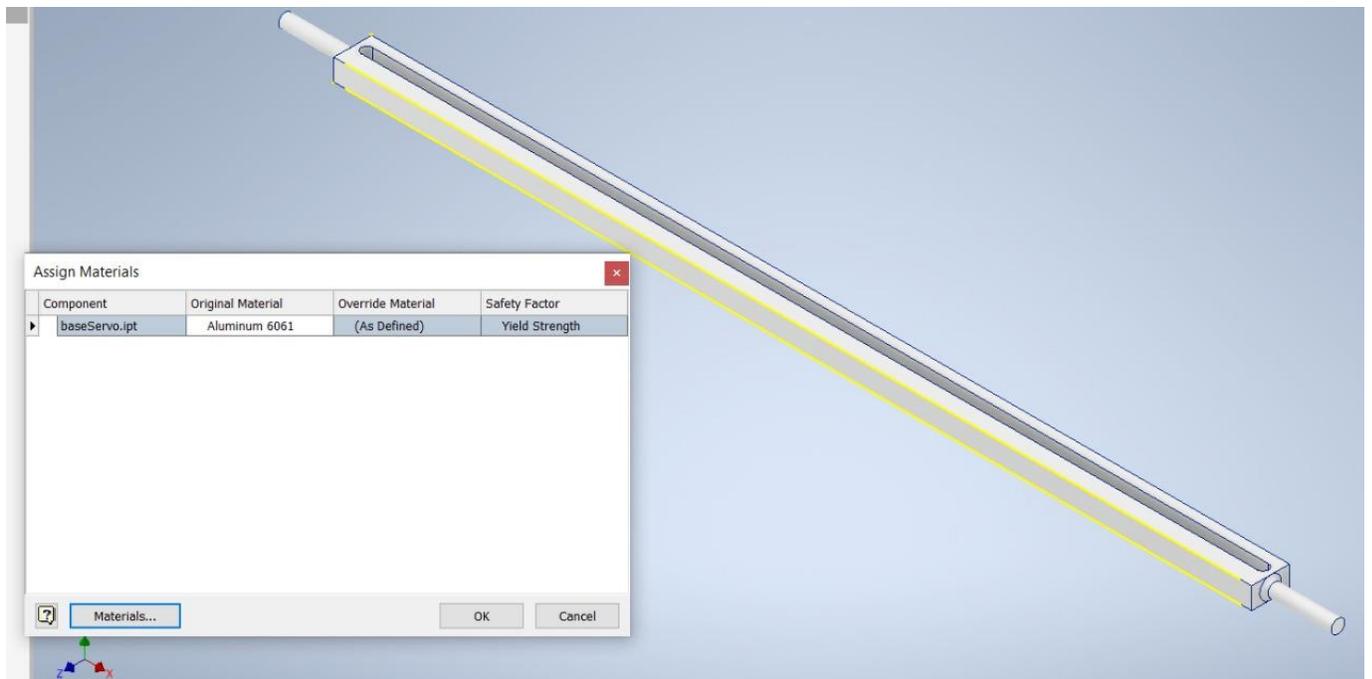


Figura 5.19. Asignación del material al eje del soporte del servomotor. Fuente propia.

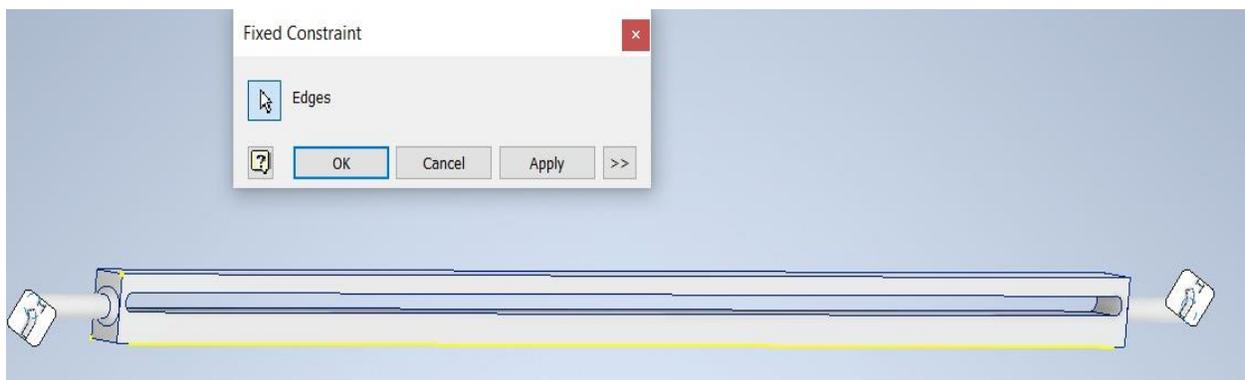


Figura 5.20. Asignación de las restricciones de movimiento al eje del soporte del servomotor. Fuente propia.

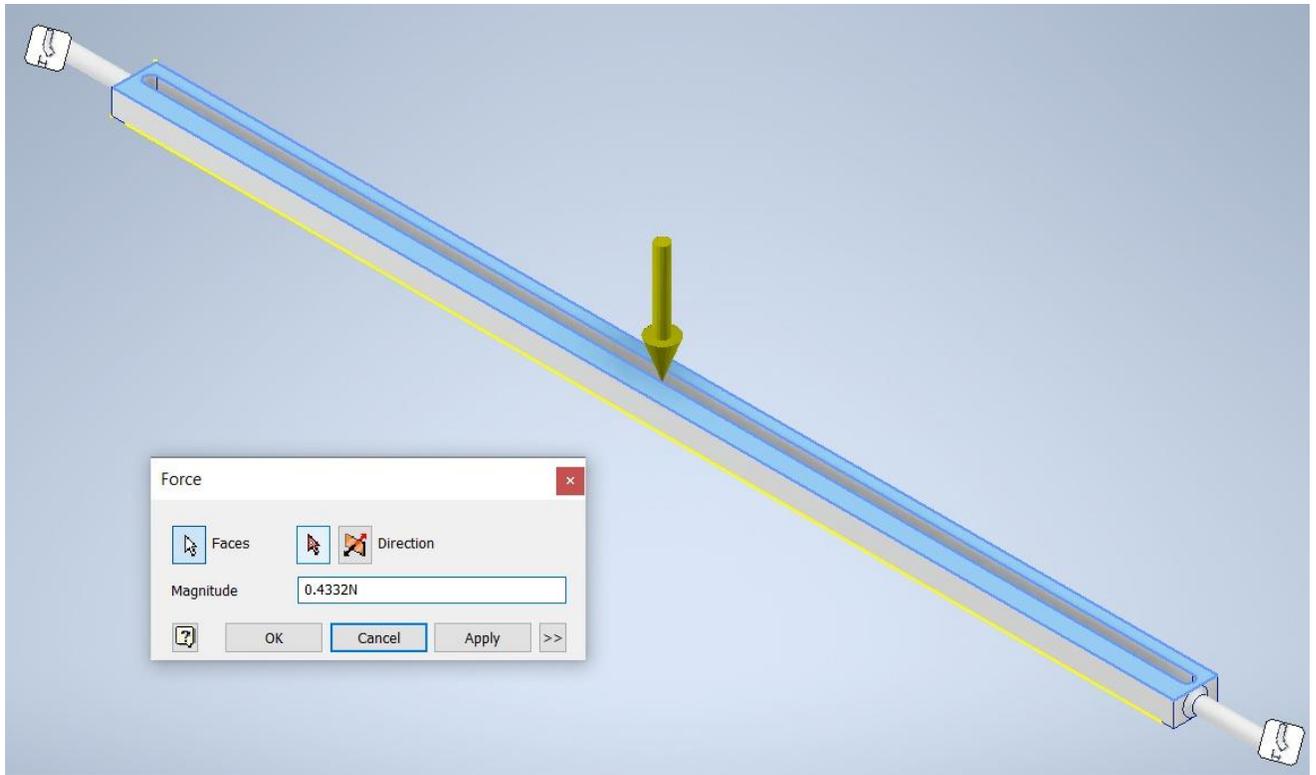


Figura 5.21. Asignación de la carga al eje del soporte del servomotor. Fuente propia.

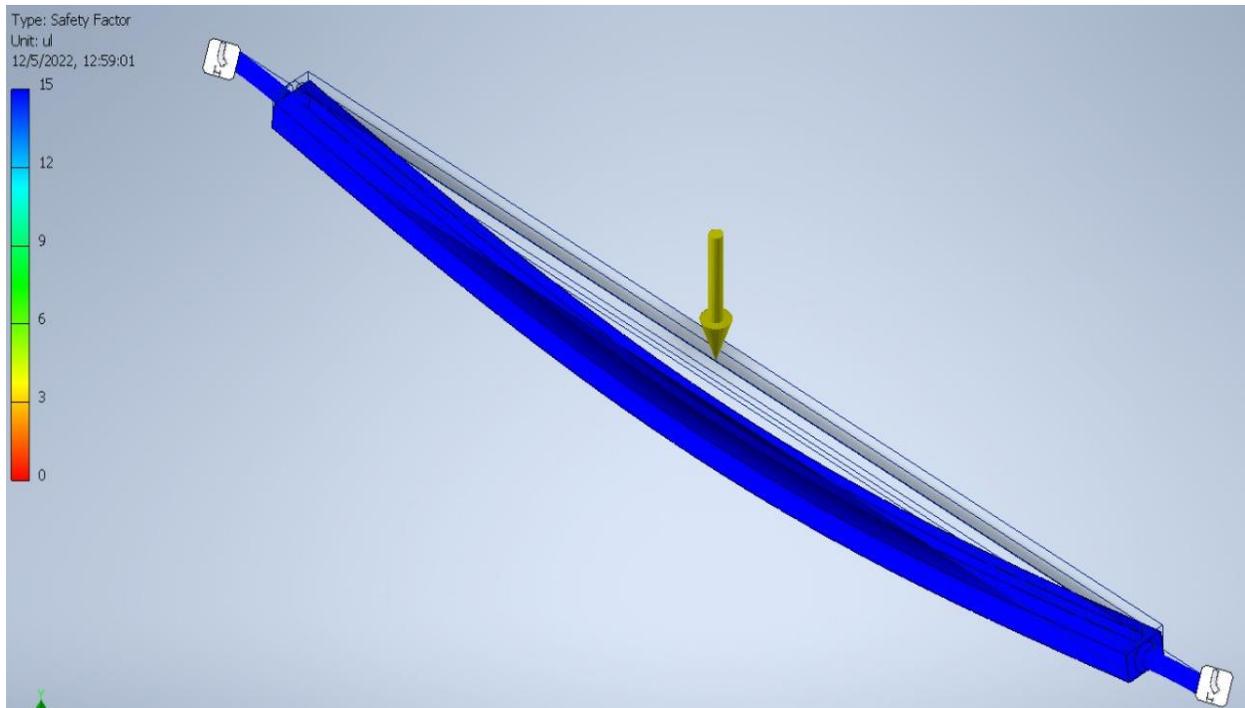


Figura 5.22. Resultados de la simulación del análisis estático por medio del software Inventor de al eje del soporte del servomotor para el caso de la deflexión. Fuente propia.

Para el caso de este eje también se realizó una prueba de deflexión debido a que la deformación que presente es importante, ya que el sistema está diseñado con la idea de que la distancia del servomotor y el teclado sea muy pequeña. En la Figura 5.23 se evidencia que la deformación es mínima, se podría decir imperceptible ya que en la parte de mayor deformación alcanza un valor de solo 0,0075mm por lo cual no presenta problemas con respecto a la funcionalidad que se espera.

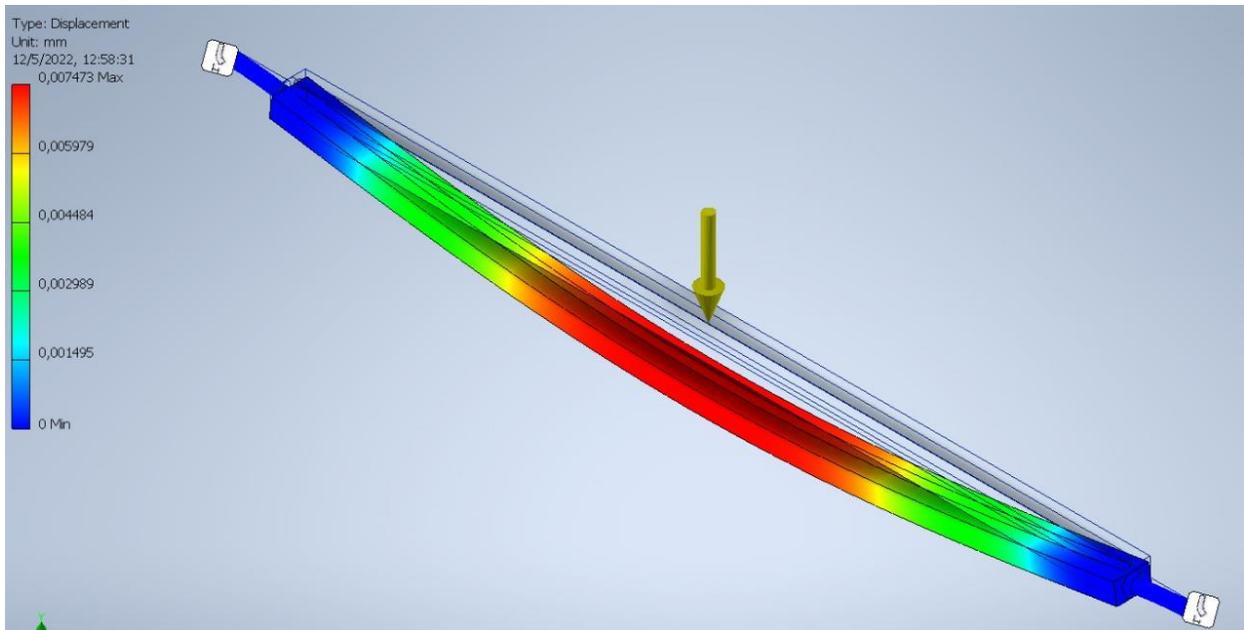


Figura 5.23. Resultados de la simulación del análisis estático por medio del software Inventor al eje del soporte del servomotor correspondiente al factor de seguridad. Fuente propia.

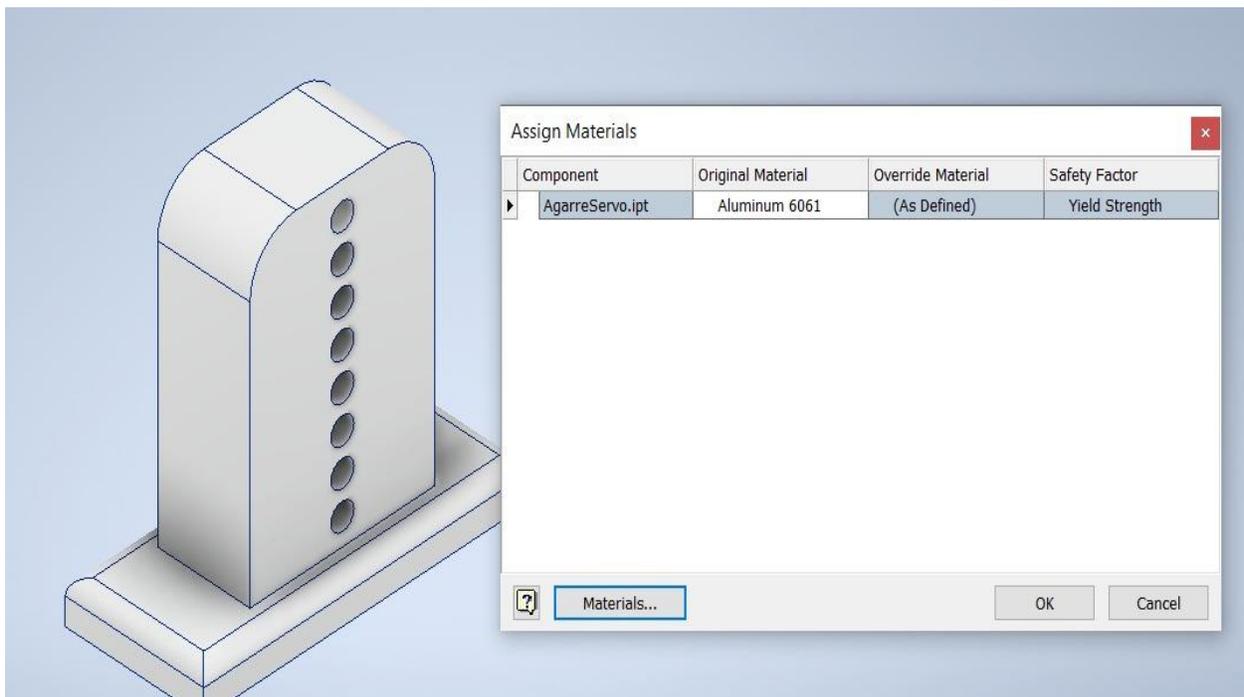


Figura 5.24. Asignación del material a la base del soporte del servomotor. Fuente propia.

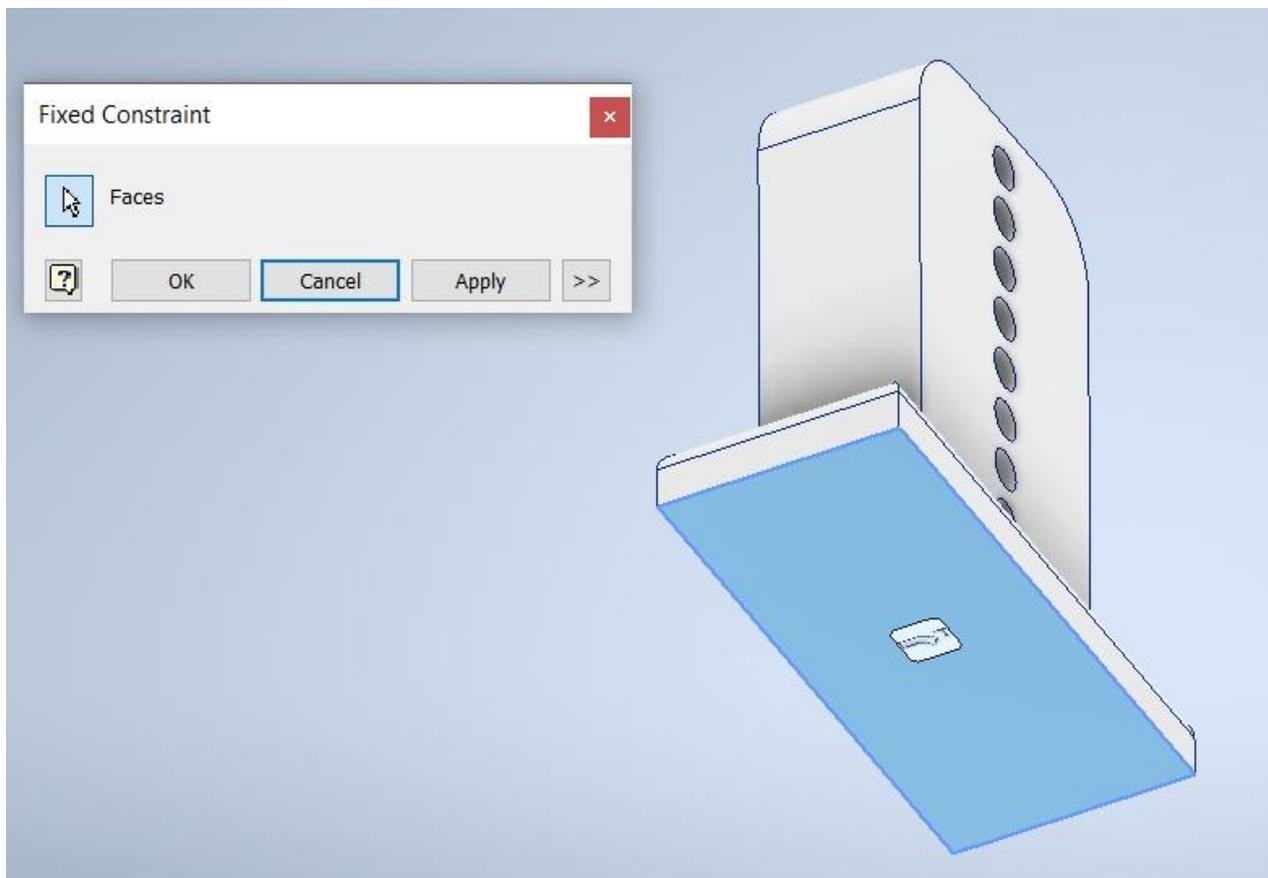


Figura 5.25. Asignación de las restricciones de movimiento la base del soporte del servomotor.
Fuente propia.

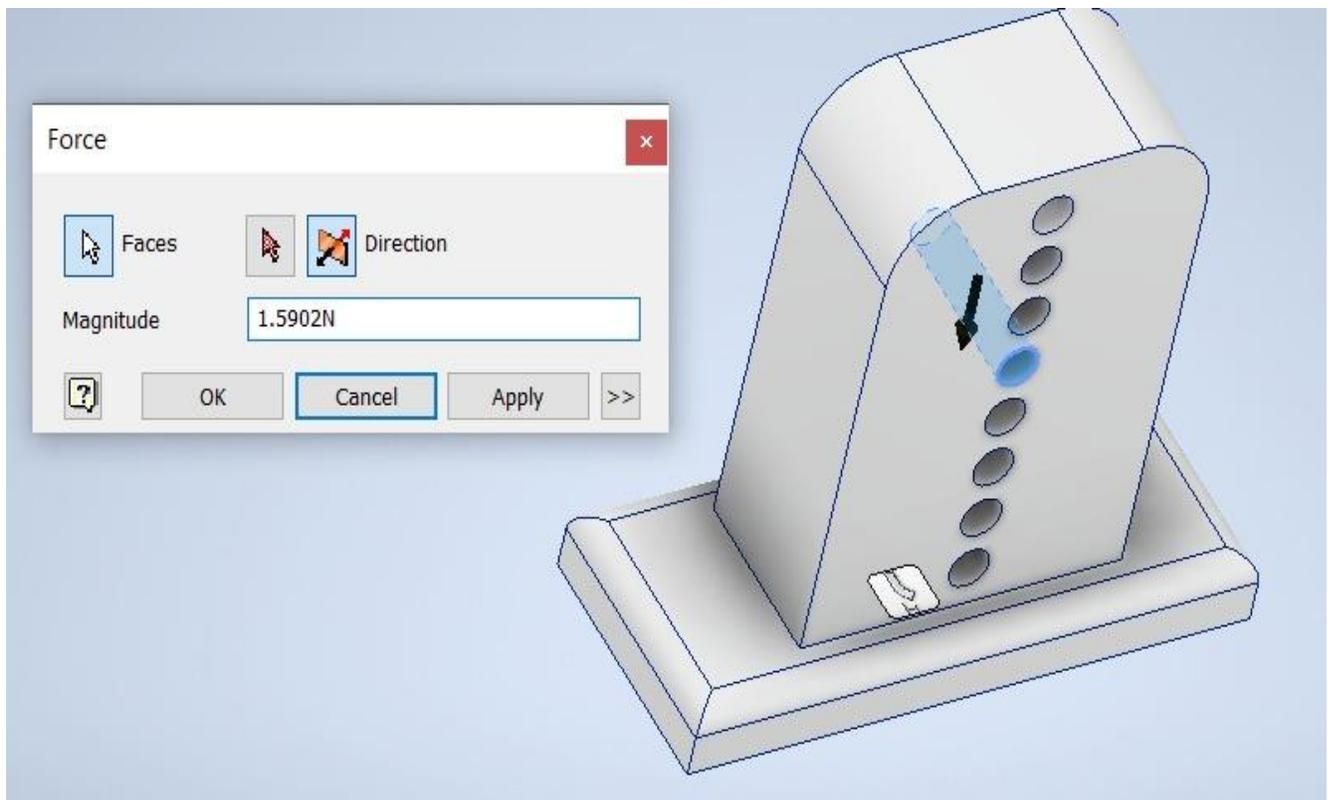


Figura 5.26. Asignación de la carga a la base del soporte del servomotor. Fuente propia.

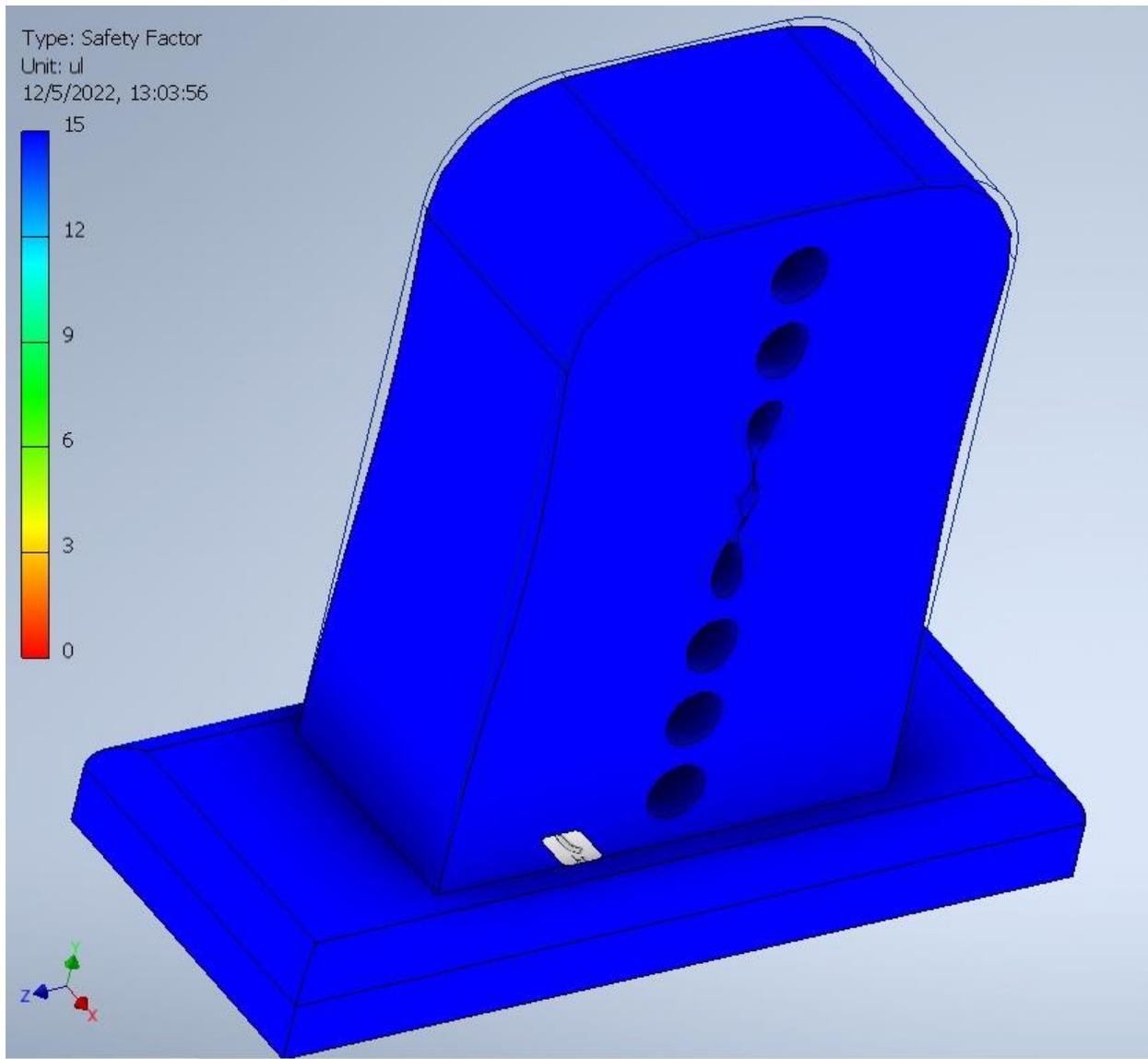


Figura 5.27. Resultados de la simulación del análisis estático por medio del software Inventor de la base del soporte del servomotor. Fuente propia.

Se le brinda especial atención a las Figuras 5.9, 5.14, 5.18, 5.23 y 5.27 debido a que ellas muestran los resultados obtenidos de las simulaciones realizadas, en los tres casos se puede notar como las diferentes piezas de la estructura poseen un color azul en su totalidad, utilizando la barra de colores al lado de cada pieza se concluye que todas las piezas poseen un factor de seguridad mayor que 12, como requerimiento se buscaba que este valor fuera mayor que 3. En la tabla 5.3 se puede observar un resumen de los resultados que se obtuvo en comparación con los requerimientos que se poseen.

Tabla 5.3. Resumen de resultados obtenidos en la simulación de estrés estático. Fuente propia

Pieza	Factor de seguridad obtenido	Factor de seguridad requerido	Resultado
Soporte de la cámara	>12	>3	Cumple
Parte media	>12	>3	Cumple
Base de la estructura	>12	>3	Cumple
Eje de la estructura de soporte del servomotor	>12	>3	Cumple
Base de la estructura de soporte del servomotor	>12	>3	Cumple

Este gran valor en el factor de seguridad indica que la estructura no va a presentar fallas cuando se utilice como soporte de la cámara para lo cual fue diseñado, por lo tanto, es una estructura con gran durabilidad que puede cumplir su propósito.

5.1.2 Prueba Verificación de estados

Para esta prueba se coloca la herramienta frente a la computadora y se presenta dos casos al azar, modo encendido y el modo reposo, véase las Figuras 5.28 y 5.29. Al presionar la tecla “V” del teclado la herramienta inicia el proceso de identificación de estados, al finalizar se despliega un mensaje en la computadora que controla la herramienta indicando cual estado fue el identificado según el código utilizado.

En el caso que la pantalla detecta la computadora en el estado de reposo despliega el mensaje “Se ha detectado la computadora en sleep mode ¿Desea iniciar el cronómetro?” como se muestra en la Figura 5.30, en el caso que se encuentre la pantalla con una luminosidad alta se

imprime en la consola el mensaje “Si ya posee la imagen a comparar proceda a activar el sleep mode” véase Figura 5.31.

La prueba se realiza 30 veces debido a que fue el valor seleccionado para las mediciones de tiempo como se especifica en la subsección 5.1. En la Tabla 5.3 se muestra el orden de las pruebas y el resultado obtenido.



Figura 5.28. Estado modo reposo, tomada desde la cámara del sistema diseñado. Fuente propia.

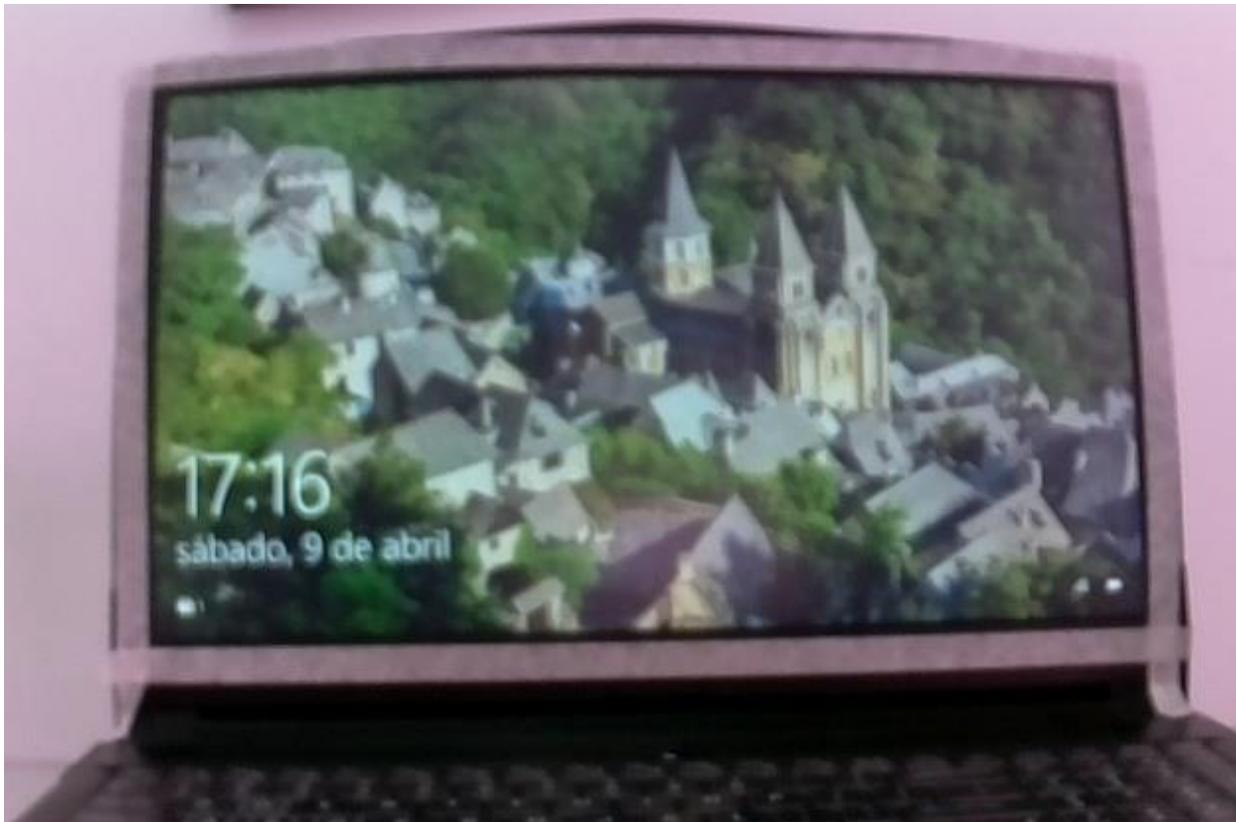


Figura 5.29. Estado modo encendido, tomada desde la cámara del sistema diseñado. Fuente propia.

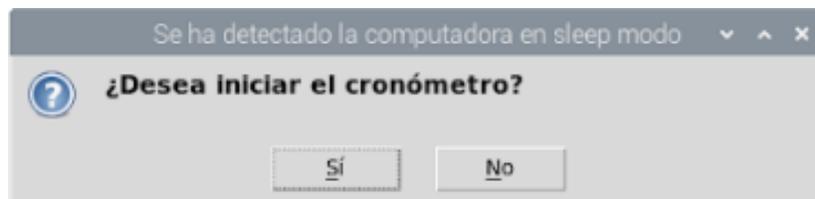


Figura 5.30. Mensaje que despliega el sistema cuando detecta la computadora en el estado de reposo. Fuente propia.

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run borrar.py
Si ya posee la imagen a comparar proceda a activar el sleep mode
```

Figura 5.31. Mensaje que despliega el sistema cuando detecta la computadora encendida.
Fuente propia.

Tabla 5.4. Pruebas y resultados de la capacidad de identificación de estados de la herramienta diseñada. Fuente propia

Número de prueba	Estado del computador	Estado detectado	Resultado
Prueba 1	Reposo	Reposo	Correcto
Prueba 2	Reposo	Reposo	Correcto
Prueba 3	Encendido	Encendido	Correcto
Prueba 4	Reposo	Reposo	Correcto
Prueba 5	Encendido	Encendido	Correcto
Prueba 6	Reposo	Reposo	Correcto
Prueba 7	Encendido	Encendido	Correcto
Prueba 8	Reposo	Reposo	Correcto
Prueba 9	Reposo	Reposo	Correcto
Prueba 10	Encendido	Encendido	Correcto
Prueba 11	Reposo	Reposo	Correcto
Prueba 12	Encendido	Encendido	Correcto
Prueba 13	Encendido	Encendido	Correcto
Prueba 14	Encendido	Encendido	Correcto
Prueba 15	Reposo	Reposo	Correcto
Prueba 16	Reposo	Reposo	Correcto
Prueba 17	Reposo	Reposo	Correcto
Prueba 18	Reposo	Reposo	Correcto
Prueba 19	Reposo	Reposo	Correcto
Prueba 20	Encendido	Encendido	Correcto
Prueba 21	Reposo	Reposo	Correcto
Prueba 22	Reposo	Reposo	Correcto
Prueba 23	Encendido	Encendido	Correcto
Prueba 24	Reposo	Reposo	Correcto
Prueba 25	Reposo	Reposo	Correcto
Prueba 26	Encendido	Encendido	Correcto
Prueba 27	Reposo	Reposo	Correcto
Prueba 28	Reposo	Reposo	Correcto
Prueba 29	Reposo	Reposo	Correcto
Prueba 30	Encendido	Encendido	Correcto

Basado en la Tabla 5.4 se genera una tabla resumen, véase Tabla 5.5, con la cantidad de veces que se realizó la prueba en cada estado y cuál fue el porcentaje de detecciones correctas.

Tabla 5.5. Resumen de resultados obtenidos en la prueba de identificación del estado de la computadora por medio de la herramienta diseñada. Fuente propia

Estado	Cantidad de pruebas	Cantidad de detecciones correctas	Porcentaje de detecciones correctas	Porcentaje de error
Reposo	19	19	100%	0%
Encendido	11	11	100%	0%

Como se muestra en los resultados de la tabla 5.5, el porcentaje de error es de 0% lo cual satisface el requerimiento que se había indicado al inicio que correspondía a un porcentaje de error menor a 5%, por lo tanto, se puede concluir que el sistema realiza la detección de estados de manera correcta.

5.1.3 Prueba capacidad de medición del tiempo

Como se mencionó en la sección 2 del documento, el método de medición existente calcula el tiempo que tarde la computadora en llegar a al escritorio desde modo reposo por medio de un evento programado a una hora predefinida.

Este método posee una variabilidad significativa, se observaron 8 resultados con un valor mínimo de 0.7s hasta un valor máximo de 2.22 s, considerando lo anterior y utilizando una computadora de pruebas con procesador de generación 11, se estableció una expectativa de este tiempo a medir dentro de un rango de 1s a 1.10s, esto por medio del criterio de experto obtenido por el cliente y basándose en las características de la portátil, debido a que se espera un tiempo menor de 1 segundo para la generación 12, y como fue mencionado la utilizada es generación 11.

El programa diseñado realiza la toma del tiempo por medio de comparación de imágenes, al iniciar a ejecutar el código se despliega una ventana donde se puede observar lo que la cámara se encuentra visualizando a tiempo real, con una leyenda que brinda información, indicando al usuario presionar la tecla “barra espaciadora” para tomar la fotografía con la que se compara más adelante, véase Figura 5.32.

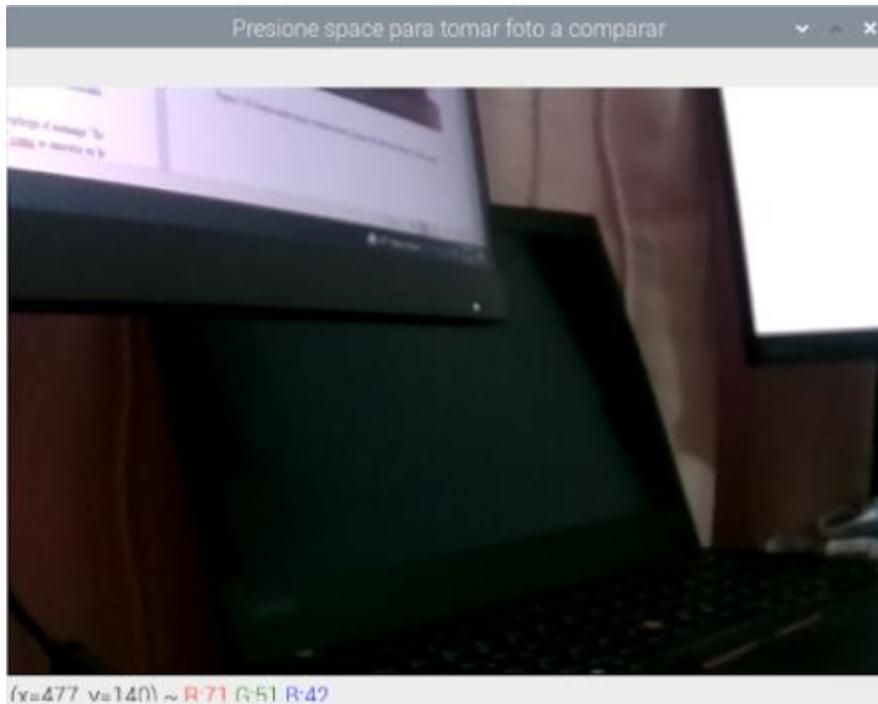


Figura 5.32. Pantalla que se despliega al inicializar el código del sistema diseñado. Fuente propia.

Cuando se posee la imagen a comparar, la cual no es necesaria tomar cada vez que se realice la prueba, pero es lo recomendable, se procede a activar el temporizador, seleccionando “sí” cuando el programa realice la pregunta como se muestra en la Figura 5.30. Antes de que el temporizador se inicialice es necesario colocar el tiempo que se desea esperar antes de enviar la señal de activación al servomotor y presionar la tecla “Iniciar la cuenta regresiva”, este botón se muestra en la Figura 5.33. Para efectos de las pruebas realizadas se colocaba un tiempo de 15 minutos.



Figura 5.32. Selección del tiempo a esperar para enviar la señal de activación al servomotor.
Fuente propia.

Al finalizar la cuenta regresiva se inicia la grabación del video y se envía la señal de activación al servomotor, el programa despliega un mensaje en la consola cuando realiza todos esos pasos, como se muestra en la Figura 5.33.



Figura 5.33. Mensajes desplegados en la consola al enviar señales de activación. Fuente propia.

Por último, lo que el algoritmo realiza es una comparación de la imagen tomada al inicio con todas las extraídas del video, calculando el error de cada comparación, de esta manera se extrae el cuadro que posee el error de menor valor y se realiza el cálculo del tiempo, el programa despliega el valor del error, el cuadro donde se encuentra y el valor de la medición total. En la Figura 5.34 se muestra un ejemplo de este mensaje, vale recalcar que los datos que despliega no fueron valores obtenidos de una medición real.

```
El valor minimo de error es: 21.63521666666667  
Es el frame numero 55  
Cantidad de frames 165  
tiempo medicion: 0.6977205117543632
```

Figura 5.34. Resultado desplegado por el programa luego de realizar la comparación de imágenes. Fuente propia.

De manera resumida este fue el proceso realizado para tomar cada una de las mediciones, los datos tomados para la Tabla 5.6 fueron obtenidos a partir de una computadora portátil marca DELL modelo XPS 13, fabricado con Intel Core i7 1165G7.

Tabla 5.6. Mediciones de tiempo obtenidas con el sistema diseñado. Fuente propia

Número de prueba	Medición del tiempo (± 0.03) s
Prueba 1	1.03
Prueba 2	1.03
Prueba 3	0.97
Prueba 4	1.00
Prueba 5	1.03
Prueba 6	1.03
Prueba 7	1.00
Prueba 8	1.03
Prueba 9	1.00
Prueba 10	1.03
Prueba 11	1.03
Prueba 12	1.03
Prueba 13	1.03
Prueba 14	1.03
Prueba 15	1.00
Prueba 16	1.00
Prueba 17	1.03
Prueba 18	1.00
Prueba 19	1.03
Prueba 20	1.03
Prueba 21	1.03
Prueba 22	1.00
Prueba 23	1.00
Prueba 24	1.03
Prueba 25	1.00
Prueba 26	1.03
Prueba 27	1.03
Prueba 28	1.03
Prueba 29	1.03
Prueba 30	0.97

Con estos datos obtenidos se procede a realizar el cálculo para obtener el valor promedio de todas las mediciones con el fin de comparar con el valor de referencia y calcular su porcentaje de error.

Tabla 5.7. Valor promedio obtenido de las 30 mediciones realizadas. Fuente propia

Valor de referencia	Valor promedio de las 30 mediciones	Porcentaje de error	Porcentaje de error esperado
1±1s	1.02±0.03	2%	<5%

Luego de observar la Tabla resumen 5.7 se infiere que el porcentaje de error obtenido es menor al requerido, además que el sistema electromecánico propuesto es más preciso que el método anterior reduciendo la variabilidad de 1 s a 0.03 s.

5.1.4 Prueba verificación de la precisión de la medición y replicabilidad

Se realiza la prueba tipo 1 por medio del software Minitab, para esto es necesario establecer ciertos parámetros, los cuales se muestran en las Figuras 5.35 y 5.36, se utiliza 1s como valor de referencia según lo indicado anteriormente, se selecciona un rango de tolerancia de 0.4 esto debido a que el requerimiento del proyecto es 0.2, en el programa se coloca el rango de distancia del valor mínimo y máximo posible, es decir, la resta de 1,2 entre 0,8.

Otros parámetros utilizados son el porcentaje de tolerancia Cf y la variación del estudio, los cuales se asignaron como 20,0 y 3,0 respectivamente, el primer valor es el recomendado por el programa y viene predeterminado, el segundo es la cantidad de desviaciones estándar necesarias para abarcar todos los datos. Por último, se coloca el valor de 0,03 en la casilla de resolución debido a que este es la distancia entre valores que es capaz de medir la herramienta diseñada.

Estudio tipo 1 del sistema de medición

Datos de medición:

Referencia:

Tolerancia

Espec. superior - espec. inferior:

Espec. inferior solamente:

Espec. superior solamente:

Figura 5.35. Parámetros utilizados para realizar el estudio tipo 1 mediante minitab. Fuente propia.

Estudio tipo 1 del sistema de medición: Opciones

Porcentaje de tolerancia para calcular Cg:

Variación del estudio: (número de desviaciones estándar)

Resolución:

Sin prueba de sesgo

Título:

Figura 5.36. Opciones escogidas para realizar el estudio tipo 1 mediante minitab. Fuente propia.

Estudio tipo 1 del sistema de medición para medidas

Nombre del sistema de medición : Verificación proyecto de graduación
 Fecha del estudio: 9 de mayo

Notificado por:
 Tolerancia: 0,4
 Misc:



Estadísticas básicas		Sesgo		Capacidad	
Referencia	1	Sesgo	0,017	Cg	1,42
Media	1,017	T	4,9576	Cgk	0,82
Desv.Est.	0,0188	ValorP	0,000		
3 × Desv.Est. (VE)	0,0563	(sesgo de la prueba=0)			
Tolerancia (Tol)	0,4			%Var(Repetibilidad)	14,09%
Resolución	0,03 > 5% de Tol			%Var(Repetibilidad y sesgo)	24,50%

Figura 5.37. Resultados del estudio tipo 1 de las medidas tomadas por el instrumento diseñado. Fuente propia.

La Figura 5.37 muestra los resultados obtenidos del estudio tipo 1, se puede observar la variación de los resultados con respecto al valor de referencia, así como los límites de tolerancia establecidos, los cuales son $0,2s$.

Antes de mencionar el resultado C_g obtenido, el cual es el valor de interés, es importante mencionar que las medidas poseen un sesgo de $0,017$, esto con respecto al valor de referencia y el valor de P es menor a $0,05$ lo que indica que este sesgo, aunque luzca pequeño es estadísticamente significativo. Para motivos del proyecto este valor no es influyente debido a que el valor de referencia utilizado es una estimación obtenida por el criterio de un experto, la herramienta es la que define con más precisión la verdadera duración de la transición de estado de la computadora. Por lo tanto, que los resultados presenten un sesgo no indica que la herramienta posee errores.

Con respecto a los valores C_g y C_{gk} , se toma solo en cuenta el parámetro de C_g ya que este compara la tolerancia con la variación de las mediciones, en cambio el parámetro C_{gk} compara la

tolerancia con el sesgo el cual se mencionó no es de relevancia. Para esta prueba se espera que el valor de C_g sea mayor a 1.33 el cual es el recomendado, como se observa cumple con el requerimiento ya que el valor obtenido es de 1.42, lo que indica que la dispersión de las mediciones del sistema de medición es adecuadamente estrecha en relación con el rango de tolerancia del producto.

El último dato de relevancia obtenido en esta prueba es el porcentaje de repetibilidad de las 30 mediciones tomadas, el cual corresponde a 14,09%, según la documentación de minitab el resultado obtenido deber ser menor a 15%, lo cual significa que la variación que se debe al sistema de medición no es significativa.

5.1.5 Prueba de repetibilidad

Para esta prueba se utilizan los mismos datos utilizados en las subsecciones anteriores y se clasifican en aceptado si el valor está dentro de los límites inferiores y superiores, siendo el valor de referencia 1,00s y conociendo que la incertidumbre diseñada corresponde a 0.03s. Se pretende medir el porcentaje de repetibilidad del sistema diseñado y no de la medición en sí, es decir, el porcentaje dirá que tan probable es que el sistema brinde un valor dentro del rango de incertidumbre para el cual fue diseñado para un tiempo conocido, verificando así también la precisión.

Tabla 5.7. Mediciones de tiempo aceptadas para la prueba de repetibilidad. Fuente propia

Número de prueba	Medición del tiempo (± 0.03) s	Condición
Prueba 1	1.03	Aceptado
Prueba 2	1.03	Aceptado
Prueba 3	0.97	Aceptado
Prueba 4	1.00	Aceptado
Prueba 5	1.03	Aceptado
Prueba 6	1.03	Aceptado
Prueba 7	1.00	Aceptado
Prueba 8	1.03	Aceptado
Prueba 9	1.00	Aceptado
Prueba 10	1.03	Aceptado
Prueba 11	1.03	Aceptado
Prueba 12	1.03	Aceptado
Prueba 13	1.03	Aceptado
Prueba 14	1.03	Aceptado
Prueba 15	1.00	Aceptado
Prueba 16	1.00	Aceptado
Prueba 17	1.03	Aceptado
Prueba 18	1.00	Aceptado
Prueba 19	1.03	Aceptado
Prueba 20	1.03	Aceptado
Prueba 21	1.03	Aceptado
Prueba 22	1.00	Aceptado
Prueba 23	1.00	Aceptado
Prueba 24	1.03	Aceptado
Prueba 25	1.00	Aceptado
Prueba 26	1.03	Aceptado
Prueba 27	1.03	Aceptado
Prueba 28	1.03	Aceptado
Prueba 29	1.03	Aceptado
Prueba 30	0.97	Aceptado

Tabla 5.8. Cuadro resumen de los valores aceptados dentro del rango estipulado. Fuente propia

Número de mediciones	Cantidad de mediciones aceptadas	Porcentaje de mediciones aceptadas
30	30	100%

Basado en la Tabla 5.8 existe un 100% de valores que cumplen estar dentro del rango propuesto, lo cual indica que el instrumento diseñado mejora la precisión de la toma del tiempo en comparación con el método anterior, reduciendo la incertidumbre a 0.03s.

5.2 Análisis financiero

Para esta sección se realiza un cálculo de la inversión que se requiere para implementar el sistema, tomando en cuenta los materiales, imprevistos y el tiempo. Es importante aclarar que la empresa posee la mayoría de los materiales a utilizar por lo que la inversión correspondería a solo el tiempo de diseño, aun así, se realiza la Tabla 5.9 especificando el costo de todos los materiales.

Para los imprevistos se toma un 10% de los costos de los materiales y para el cálculo del tiempo se toma el salario mínimo que indica el ministerio de trabajo en el cual la hora para un pasante universitario corresponde a \$4,46. Los ingenieros actualmente tardan aproximadamente 5 minutos en iniciar la prueba y unos 10 minutos observando el resultado del tiempo, con la nueva herramienta les tomaría solamente unos 3 minutos para realizar la prueba obtener el resultado, por supuesto sin contar el tiempo que la computadora debe estar en modo reposo.

Asumiendo para efectos de cálculo que se realicen 3 pruebas al día esto quiere decir que se ahorran 36 minutos por día, al mes 1080 min o lo que es lo mismo 18 horas. El salario promedio de un ingeniero es de \$6.88 por hora, con lo cual se posee una ganancia de \$124 al mes, \$1486 al año.

El tiempo de diseño se asume un total de 20 horas semanales por 16 semanas.

Tabla 5.9. Cálculo del costo del proyecto. Fuente propia

Componente	Cantidad	Precio unitario	TOTAL
Servomotor MG90	1	\$8.00	\$8.00
Raspberry 3b	1	\$99.00	\$99.00
Modulo de Cámara	1	\$32.00	\$32.00
Aluminio 6061	2	\$32.00	\$64.00
Subtotal de materiales			\$203
Imprevistos	10%	\$20.30	
Subtotal de imprevistos			\$20.30
Tiempo de diseño	320	\$4.46	\$1.427.20
Subtotal de tiempo de diseño			\$1.427.20
Total			\$1.650.50

Con la inversión total necesaria para realizar el proyecto se procede a realizar el análisis de costos y beneficios para la empresa en un lapso de 5 años, se asume una tasa de descuento del 15%. Este análisis se puede observar en la Figura 5.38.

ROI (Basado en valor actual neto, \$)		2022	2023	2024	2025	2026	2027	Total	VA
Beneficios	Tipo de Ahorro	\$0	\$1,486	\$1,560	\$1,638	\$1,720	\$1,806	\$8,211	\$5,431
Horas ahorradas para reinversión	Horas	\$0	\$1,486	\$1,560	\$1,638	\$1,720	\$1,806	\$8,211	\$5,431
Item 2								\$0	\$0
Costo del Proyecto	Tipo de Inversión	\$1,650	\$20	\$20	\$20	\$20	\$20	\$1,752	\$1,718
Inversión inicial y Mantenimiento Anual	Capital Inicial	\$203	\$20	\$20	\$20	\$20	\$20	\$305	\$271
Imprevistos	Capital Inicial	\$20						\$20	\$20
Costo de Tiempo de Diseño	Horas Trabajadas	\$1,427						\$1,427	\$1,427
IMPUESTOS		\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
Cambio de capital de trabajo								\$0	\$0
Flujo de Caja Neto		(\$1,650)	\$1,466	\$1,540	\$1,618	\$1,700	\$1,786	\$6,460	\$3,713
Flujo de Caja Neto Descontado		(\$1,650)	\$1,275	\$1,164	\$1,064	\$972	\$888		\$3,713
Valor Actual Neto (\$)		\$3,713							

Figura 5.38. Cálculo de costos y beneficios totales del proyecto. Fuente propia.

En forma de resumen se realiza un cuadro con los datos más importantes del análisis realizado en la Figura 5.38, véase Tabla 5.10. En esta tabla se observa que el VAN es positivo por lo tanto el proyecto se puede catalogar como viable, además se obtiene un 89.2% de TIR con respecto a la inversión inicial. La inversión inicial se recupera en un lapso de 2 años

Tabla 5.10. Resumen del análisis económico. Fuente propia

Beneficio Total	\$8,211
Costo total	\$1,752
Retorno total	\$6,460
VAN (Después de impuestos)	\$3,713
TIR	89.2%
ROI	368.8%
Recuperación de Inversión (Años)	2

6 Conclusiones y recomendaciones

6.1 Conclusiones

- Se logra diagnosticar las etapas necesarias para cumplir los requerimientos mostrados en la Tabla 3.2 dividiéndolo en subsistemas como los mostrados en la Figura 3.3.
- Se diseñó un sistema capaz de realizar la detección del estado del computador con un porcentaje de aceptación del 100%, lo que indica que el porcentaje de rechazo es de 0%, este es menor a 5%, por lo tanto, cumple el requisito. Además, este sistema posee la capacidad de realizar la medición de tiempo de lo que tarda la computadora en realizar la transición de estados con una precisión de 0.03s, mucho menor a la requerida de 0.2s.
- Se diseñaron los distintos componentes mecánicos que forman el sistema, todas ellos con un factor de seguridad calculado por simulaciones realizadas en programas, mayor de 12, superando el valor meta de 3, los planos de las distintas piezas se adjuntan en los Anexos B1.
- Se validó la integración de todos los sistemas mediante pruebas de concepto realizadas a un prototipo donde se verificó la precisión y la repetibilidad con la prueba tipo 1 donde se obtuvo un valor de Cg de 1.44, superando el valor propuesto de 1.33, y un porcentaje de repetibilidad de 14% siendo menor al 15% el cual era el límite mayor permitido.

6.2 Recomendaciones

- Debido a que el reconocimiento del estado del computador se basa en la luminosidad reflejada en la pantalla se podría diseñar una carcasa que proteja el computador de fuentes de luces externas que podrían alterar el resultado en caso de que se desee realizar pruebas en otras localidades y no en un ambiente conocido como el laboratorio, donde fueron tomadas las muestras.
- Los cálculos estadísticos realizados, por cuestiones de tiempo y material, fueron solo las pruebas iniciales recomendadas para los instrumentos de medición, luego se podría aplicar más estudios para verificar la confiabilidad de la herramienta como un estudio tipo R&R.
- El proceso de enviar la computadora a modo de reposo no está automatizado debido a que no era requerimiento, se podría implementar un método de comunicación entre el microprocesador y la computadora que por medio de señales active el modo reposo.

- Actualmente solo el temporizador posee una interfaz gráfica, esto podría mejorarse implementado una interfaz gráfica que permita el uso de la herramienta por medio de solo interacciones con el ratón, en lugar de poseer la necesidad de presionar teclas del teclado.
- Por ahora es necesario el control de la herramienta por medio de una computadora externa, en un futuro se puede implementar los cambios necesarios para convertir el instrumento en un sistema completamente embebido, donde sea solo necesario conectarlo a una fuente de alimentación y posea una pantalla que permita el control de esta.

7 Referencias bibliográficas

- [1] Intel, *Company overview*. Obtenido en línea en: <https://www.intel.la/content/www/xl/es/company-overview/company-overview.html>
- [2] H. Sarvetnick, *Plastisols and Organosols*. New York: Van Nostrand Reinhold, 1972
- [3] Introducing the Intel Evo platform. Obtenido en línea: <https://www.intel.com/content/www/us/en/products/systems-devices/laptops/evo.html>
- [4] Difference Between Deep and Deeper Sleep States for Processors. Obtenido en línea: <https://www.intel.com/content/www/us/en/support/articles/000006619/processors/intel-core-processors.html>
- [5] Copeman, A (2021). What is Intel Evo? Obtenido en línea: <https://www.techadvisor.com/buying-advice/laptop/what-is-intel-evo-3801973/>
- [6] Gómez, B (2021). Intel Evo y Project Athena: Qué es y cómo se obtiene este certificado. Obtenido en línea: <https://www.profesionalreview.com/2020/09/06/intel-evo-y-project-athena/>
- [7] Hoffman, C (2019). How to make your PC wake from sleep automatically. Obtenido en línea en: <https://www.howtogeek.com/119028/how-to-make-your-pc-wake-from-sleep-automatically/>
- [8] “System Power States” Accesado: Dic. 7, 2021. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/windows/win32/power/system-power-states>
- [9] Vinazco, L, “Estudio de Repetibilidad y Reproducibilidad R&R,” *CircuitDigest*, Obtenido en línea: <https://cicalidad.com/articulos/PAPER%20R&R.pdf> (Accesado Dic. 07, 2021).
- [10] “Estudio R&R del systema de medición” *Soporte Minitab*. https://support.minitab.com/es-mx/minitab/18/Asistente_Estudio_R_R_del_sistema_de_medici%C3%B3n.pdf (Accesado dic. 07, 2021).
- [11] “Anova de un factor” Portal de formación estadística. Obtenido en línea: https://www.jmp.com/es_pe/statistics-knowledge-portal/one-way-anova.html. (Accesado dic. 20, 2021).

- [12] D'Inca, C. O. F. (1982). Los microprocesadores. *Revista de la Universidad de Mendoza*. (Accesado dic. 20, 2021).
- [13] Casco, S. (2014). Raspberry Pi, Arduino y Beaglebone Black Comparación y Aplicaciones. *vol., 1*, 4-8. (Accesado dic. 20, 2021).
- [14] “NVIDIA developer” *Obtenido en Línea:* <https://developer.nvidia.com/embedded/jetson-nano-developer-kit> (Accesado dic. 20, 2021).
- [15] “Asus TinkerBoard,” Accesado: dic. 20, 2021. [En línea]. Disponible: <https://www.asus.com/latin/Networking-IoT-Servers/AIoT-Industrial-Solutions/All-series/Tinker-Board/>. (Accesado dic. 20, 2021).
- [16] Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 20(2), 1-13. (Accesado dic. 20, 2021).
- [17] Fernández, O. B. (2005). Introducción al lenguaje de programación Java. Una guía básica, 9. (Accesado dic. 20, 2021).
- [18] SI Didact. (2005). *Manual de programación lenguaje C++*. MAD-Eduforma. (Accesado dic. 20, 2021).
- [19] Drugarin, C. A., Draghici, S., & Raduca, E. (2016). Team Viewer Technology for Remote Control of a Computer. *Analele Universitatii'Eftimie Murgu'*, 23(1), 61-66. (Accesado dic. 20, 2021).
- [20] “Oficina de seguridad del internauta, “Recursos, VNC viewer. *Obtenido en Línea:* <https://www.osi.es/es/herramientas-gratuitas/vnc-viewer#:~:text=VNC%20es%20un%20programa%20que,dispositivo%2C%20como%20ordenadores%20o%20smartphones>. (Accesado dic. 20, 2021).
- [21] García, I., & Caranqui, V. (2015). La visión artificial y los campos de aplicación. *Tierra infinita*, 1(1), 94-103. (Accesado dic. 21, 2021).
- [22] J. Mariño, “Módulo 1. Sesión 8: Elementos fundamentales de Diseño de Experimentos (I),” *Tecnológico de Costa Rica*. Cartago, 2021. (Accesado dic. 21, 2021).

- [23] Ramírez, L. G. C., Jiménez, G. S. A., & Carreño, J. M. (2014). *Sensores y actuadores*. Grupo Editorial Patria. (Accesado dic. 21, 2021).
- [24] “Que es un servomotor y para qué sirve” *centro de formación técnica para la Industria*. <https://www.cursosaula21.com/que-es-un-servomotor/> (Accesado dic. 22, 2021).
- [25] Conti, F. (2005). *Motores paso a paso*. TECNIBOOK EDICIONES.
- [26] K. T. Ulrich and S. D. Eppinger, *Diseño y desarrollo de productos*, vol. 5, no. 1. 2017
- [27] “Comparativa entre todas las raspberry pi del mercado” 2020. [En línea]. Disponible en: <https://yourgeekweb.com/es/2020/05/05/comparativa-entre-todas-las-raspberry-pi-del-mercado/>
- [28] Raspberry, “Página oficial de la raspberry pi” Obtenido en línea: <https://www.raspberrypi.org/>
- [29] Python docs, “time-tiempo de acceso y conversiones.” [En línea]. Disponible en: <https://docs.python.org/es/3/library/time.html>
- [30] Python docs, “Interfaz gráfica de usuario con TK.” [En línea]. Obtenido en: <https://docs.python.org/es/3/library/tk.html>
- [31] GPIO zero docs, “servo.” [En línea]. Obtenido en: https://gpiozero.readthedocs.io/en/stable/api_output.html?highlight=servo#gpiozero.Servo
- [32] Amazon, “Fuentes de alimentación Raspberrys pi 3B” 2007. [En línea]. Disponible en: <https://www.amazon.com/PiSeries/dp/B08523DFR4>
- [33] “RPi USB webcams,” [En línea]. Obtenido en: https://elinux.org/RPi_USB_Webcams
- [34] Raspberry pi documentation. “Camera” [En línea]. Obtenido en: <https://www.raspberrypi.com/documentation/accessories/camera.html#raspberry-pi-hq-camera-filter-removal>
- [35] MicroJPM. “Raspberry PiCamera 5MP Night Vision” [En línea]. Obtenido en: <https://www.microjpm.com/products/ad53484/>

- [36] MicroJPM. “MG90S Micro servo” [En línea]. Obtenido en: <https://www.microjpm.com/products/micro-servo-high-torque-metal-gear-mg90s/>.
- [37] Siemens, “ES_6ES5710-8MA11,” Hoja de datos, ago. 2020. https://www.electricalautomationnetwork.com/PDF/PIM/SIE/ES_6ES5710-8MA11.pdf
- [38] The Raspberry Pi Foundation. Documentation and datasheets for Raspberry Pi products.2020. [En línea]. Disponible en: <https://datasheets.raspberrypi.org/>
- [39] Guía de materiales de impresión 3D: Tipos, aplicaciones y propiedades. [En línea]. Disponible en: <https://formlabs.com/es/blog/materiales-impresion-3d/>
- [40] MG90s Datasheet. [En línea]. Disponible en: <https://datasheetspdf.com/pdf/1106582/ETC/MG90S/1>
- [41] Soporte de Minitab. “Un estudio tipo1 del sistema de medición evalúa la capacidad de un proceso de medición” [En línea]. Obtenido en: <https://support.minitab.com/es-mx/minitab/19/help-and-how-to/quality-and-process-improvement/measurement-system-analysis/supporting-topics/other-gage-studies-and-measures/type-1-gage-study/>
- [42] Soporte de Minitab. “Ejemplo de Estudio tipo1 del sistema de medición” [En línea]. Obtenido en: <https://support.minitab.com/es-mx/minitab/19/help-and-how-to/quality-and-process-improvement/measurement-system-analysis/how-to/gage-study/type-1-gage-study/before-you-start/example/>
- [43] Hernández-Castellano, P. M., Taboada-Margalejo, E. A., Rodríguez-Perdomo, J. I., Marrero-Alemán, M. D., Ortega García, F., Socorro-Perdomo, P. P., ... & Suárez-García, L. A. (2016). Introducción al sistema ISO GPS. Tolerancias dimensionales.
- [44] Gasque Silva, L. Aluminio. *Educación Química*, 14(1), 52-53.
- [45] “Calculadora de ajustes” [En línea]. Obtenido en: <https://www.facturee.de/es/herramientas-online/calculadora-de-ajuste/>
- [46] Grasso, J. M. R., Valhuerdi, A. T., & González, A. A. TOLERANCIAS DIMENSIONALES Y GEOMÉTRICAS.

- [47] “Signos superficiales” [En línea]. Obtenido en:
<https://ibiguridt.wordpress.com/temas/conjunto/signos-superficiales/#arriba>
- [48] Monllor Miralles, J. (2013). *Norma UNE-EN ISO 9001: 2000 Directiva 98/37/CE (Marcado CE)* (Doctoral dissertation, Universitat Politècnica de València).

A Apéndices

A.1 Encuesta de jerarquización

En la Tabla B.1 se puede observar el documento que se le envió al cliente Luis Carlos Rosales para realizar la jerarquización y la añadidura de nuevas necesidades.

Tabla B.1 Encuesta para la jerarquización de necesidades.

Diseño de un sistema embebido para la medición del tiempo desde el estado Deep sleep hasta el estado Wake up	
Jerarquización de las necesidades	
Para cada una de las siguientes necesidades identificadas, por favor indique en una escala del 1 al 5 que tan importante es esa necesidad, se puede repetir el valor las veces que se desee. Si considera que es importante agregar otra necesidad favor de agregarlo en los espacios en blanco.	
1-La función es indeseable. No consideraría un producto con esta función. 2-La función no es importante, pero no importaría tenerla 3- Sería bueno tener esa función, pero no es necesaria. 4-La función es altamente deseable, pero consideraría un producto sin esta función. 5-La función es de importancia crítica, no consideraría un producto sin esta función.	
Importancia en escala del 1 al 5	Necesidad
	El sistema debe funcionar independientemente de la computadora a la cual se realice la prueba
	El sistema debe llevar a cabo la medición con la menor interacción humana posible
	El sistema debe poseer contadores de tiempo que indique cuando ejecutar las acciones
	El sistema debe provocar el cambio de estado por medio de una acción mecánica
	El sistema debe reconocer el estado de la computadora
	El sistema debe poder realizar mediciones de tiempo

A.2 Código utilizado

```
1 |
2 | """
3 | Created on Wed Jan  5 10:59:10 2022
4 |
5 | @author: maarayaa
6 | """
7 |
8 | import cv2
9 | import matplotlib.pyplot as plt
10 | import imutils
11 | from imutils.perspective import four_point_transform
12 | from PIL import Image, ImageStat
13 | import numpy as np
14 | import math
15 | import numpy
16 | import RPi.GPIO as GPIO
17 | import time
18 | from time import sleep
19 | from gpiozero import Servo
20 | from picamera import PiCamera
21 | from picamera.array import PiRGBArray
22 | from tkinter import *
23 | from tkinter import messagebox as MessageBox
24 | import tkinter as tk
25 | import sys
26 | import datetime as dt
27 |
28 |
29 | #Todos los comentarios que posee el codigo son lineas
30 | #utilizadas para verificar funcionalidad de cada funcion
31 |
32 |
33 |
34 | def ifSleep(ima):
35 |     #image = cv2.imread(ima)
36 |     gray = cv2.cvtColor(ima, cv2.COLOR_BGR2GRAY)
37 |     blurred = cv2.medianBlur(gray,3)
38 |     #median = cv2.medianBlur(dilatada,5)
39 |     edged = cv2.Canny(blurred, 100, 200, 255)
40 |     kernel = np.ones((5,5), np.uint8)
41 |     dilatada = cv2.dilate(edged, kernel, iterations=1)
42 |
43 |
44 |     #plt.imshow(image)
45 |     #plt.show()
46 |     #plt.imshow(gray)
47 |     #plt.show()
48 |     #plt.imshow(blurred)
49 |     #plt.show()
50 |     #plt.imshow(edged)
51 |     #plt.show()
52 |     #plt.imshow(edged2)
53 |     #plt.show()
54 |     #plt.imshow(dilatada)
55 |     #plt.show()
56 |
```

Figura B2.1. Código utilizado para el funcionamiento del sistema diseñado (1/). Fuente propia.

```

56
57     # Encuentra los contornos en la imagen y los acomoda
58     # en orden descendente
59     #PARA PRIMERA IMAGEN
60     cnts = cv2.findContours(dilatada.copy(), cv2.RETR_EXTERNAL,
61                             cv2.CHAIN_APPROX_SIMPLE)
62     cnts = imutils.grab_contours(cnts)
63     cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
64     displayCnt = None
65     # ciclo sobre los contornos
66     for c in cnts:
67         #aproxima el valor del contorno
68         peri = cv2.arcLength(c, True)
69         approx = cv2.approxPolyDP(c, 0.02 * peri, True)
70         # Extrae el contorno que posee 4 vertices
71         if len(approx) == 4:
72             displayCnt = approx
73             break
74         # Extrae el contorno y le aplica una transformacion de perspectiva
75         warped = four_point_transform(gray, displayCnt.reshape(4, 2))
76         output = four_point_transform(image, displayCnt.reshape(4, 2))
77         #plt.imshow(output)
78         #plt.show()
79         #frameHSV = cv2.cvtColor(output, cv2.COLOR_BGR2HSV)
80         gray = cv2.cvtColor(output, cv2.COLOR_BGR2GRAY)
81         #plt.imshow(gray)
82         #plt.show()
83         #hist = cv2.calcHist([gray], [0], None, [256], [0, 256])
84         #plt.plot(hist, color='gray' )
85
86         #plt.xlabel('intensidad de iluminacion')
87         #plt.ylabel('cantidad de pixeles')
88         #plt.show()
89         #cv2.destroyAllWindows()
90         #im = Image.open(output)
91         #stat = ImageStat.Stat(gray.copy())
92         #r,g,b = stat.mean
93         #brillo= math.sqrt(0.241*(r**2) + 0.691*(g**2) + 0.068*(b**2))
94         #print (brillo)
95         #Transforma el espacio de RGB a HSV
96         hsv = cv2.cvtColor(output, cv2.COLOR_BGR2HSV)
97         h, s, v = cv2.split(hsv)
98         #brillo= numpy.mean( v)
99         #print("el brillo es de", brillo)
100        #print("el h es de", numpy.mean( h))
101        #print("el s es de", numpy.mean( s))
102        #Calcula el promedio y selecciona valor umbral
103        if (numpy.mean( s)<70) :
104            return True
105        else:
106            return False
107

```

Figura B2.2. Código utilizado para el funcionamiento del sistema diseñado (2/10). Fuente propia.

```

108
109 def compIma(im1,im2):
110     # Transforma las imagenes a escalas de grises
111     gray = cv2.cvtColor(im1, cv2.COLOR_BGR2GRAY)
112     gray2 = cv2.cvtColor(im2, cv2.COLOR_BGR2GRAY)
113     #blurred = cv2.medianBlur(gray2,3)
114     #median = cv2.medianBlur(dilatada,5)
115     #engruesa contornos
116     edged = cv2.Canny(gray, 100, 200, 255)
117     edged2 = cv2.Canny(gray2, 100, 200, 255)
118     kernel = np.ones((5,5), np.uint8)
119     dilatada = cv2.dilate(edged, kernel, iterations=1)
120     dilatada2 = cv2.dilate(edged2, kernel, iterations=1)
121
122
123     #plt.imshow(image)
124     #plt.show()
125     #plt.imshow(image2)
126     #plt.show()
127     #plt.imshow(gray2)
128     #plt.show()
129     #plt.imshow(blurred)
130     #plt.show()
131     #plt.imshow(edged)
132     #plt.show()
133     #plt.imshow(edged2)
134     #plt.show()
135     #plt.imshow(dilatada)
136     #plt.show()
137
138     # Encuentra los contornos en la imagen
139     # Acomoda en orden descendente
140     #PARA PRIMERA IMAGEN
141     cnts = cv2.findContours(dilatada.copy(), cv2.RETR_EXTERNAL,
142         cv2.CHAIN_APPROX_SIMPLE)
143     cnts = imutils.grab_contours(cnts)
144     cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
145     displayCnt = None
146     # ciclo sobre los contornos
147     for c in cnts:
148         # peimetro de contorno
149         peri = cv2.arcLength(c, True)
150         approx = cv2.approxPolyDP(c, 0.02 * peri, True)
151         # extrae el contorno con 4 vertices
152         if len(approx) == 4:
153             displayCnt = approx
154             break
155     # aplica transformada de perspectiva
156     try:
157         warped = four_point_transform(dilatada, displayCnt.reshape(4, 2))
158         output = four_point_transform(im1, displayCnt.reshape(4, 2))
159     except:
160         output=im1
161

```

Figura B2.3. Código utilizado para el funcionamiento del sistema diseñado (3/10). Fuente propia.

```

161
162     #plt.imshow(output)
163     #plt.show()
164     #PARA SEGUNDA IMAGEN
165     cnts = cv2.findContours(dilatada2.copy(), cv2.RETR_EXTERNAL,
166         cv2.CHAIN_APPROX_SIMPLE)
167     cnts = imutils.grab_contours(cnts)
168     cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
169     displayCnt = None
170     # ciclo sobre los contornos
171     for c in cnts:
172         # peimetro de contorno
173         peri = cv2.arcLength(c, True)
174         approx = cv2.approxPolyDP(c, 0.02 * peri, True)
175         # extrae el contorno con 4 vertices
176         if len(approx) == 4:
177             displayCnt = approx
178             break
179     # aplica transformada de perspectiva
180     try:
181         warped2 = four_point_transform(dilatada2, displayCnt.reshape(4, 2))
182         output2 = four_point_transform(im2, displayCnt.reshape(4, 2))
183     except:
184         output2=im2
185     #plt.imshow(output2)
186     #plt.show()
187     output= cv2.resize(output,(600,300))
188     output2=cv2.resize(output2,(600,300))
189
190
191     err = np.sum((output.astype("float") - output2.astype("float"))**2)
192     err /= float(output.shape[0] * output2.shape[1])
193     return err

```

Figura B2.4. Código utilizado para el funcionamiento del sistema diseñado (4/10). Fuente propia.

```

194
195     #Pregunta al usuario
196     def question():
197         return MessageBox.askquestion("Se ha detectado la computadora en sleep modo", "¿Desea iniciar el cronómetro?")
198     def cerrar():
199         root.destroy()
200     #funcion que controla el movimiento del servo
201     def moverServo():
202         global motor
203         servo = Servo(17)
204         while True:
205             servo.min()
206             sleep(1)
207             end=time.time()
208             servo.mid()
209             sleep(1)
210             break
211     motor=0

```

Figura B2.5. Código utilizado para el funcionamiento del sistema diseñado (5/10). Fuente propia

```

213     #Funcion que controla la cuenta regresiva
214     def submit():
215         global motor
216         try:
217
218
219             temp = int(hour.get())*3600 + int(minute.get())*60 + int(second.get())
220         except:
221             print("xxxxxxx")
222         while temp >-1:
223
224
225             mins,secs = divmod(temp,60)
226
227
228
229
230             hours=0
231             if mins >60:
232
233
234
235                 hours, mins = divmod(mins, 60)
236
237
238
239                 hour.set("{0:2d}".format(hours))
240                 minute.set("{0:2d}".format(mins))
241                 second.set("{0:2d}".format(secs))
242
243
244
245                 root.update()
246                 time.sleep(1)
247
248
249
250                 if (temp == 0):
251                     root.destroy()
252                     motor=1
253
254
255
256
257                 temp -= 1
258     #Otro metodo utilizado para comparar las imagenes
259

```

Figura B2.6. Código utilizado para el funcionamiento del sistema diseñado (6/10). Fuente propia

```

258     #Otro metodo utilizado para comparar las imagenes
259
260     ▼ def otro(im1,im2):
261         output=im1
262         output2=im2
263         output= cv2.resize(output,(600,300))
264         output2=cv2.resize(output2,(600,300))
265
266
267         err = np.sum((output.astype("float") - output2.astype("float"))**2)
268         err /= float(output.shape[0] * output2.shape[1])
269         return err
270
271     #Configuracion de la camara
272     cam = PiCamera()
273     cam.resolution = (624, 416)
274     cam.framerate = 30
275     rawCapture = PiRGBArray(cam, size=(624, 416))
276     img_counter = 0
277     estado=True
278     #Inicio de la interfaz
279     root = Tk()
280     root.mainloop

```

Figura B2.7. Código utilizado para el funcionamiento del sistema diseñado (7/10). Fuente propia

```

282 #Condiciones para utilizar la herramienta
283 while True:
284     for frame in cam.capture_continuous(rawCapture, format="bgr", use_video_port=True):
285         image = frame.array
286         cv2.imshow("Presione space para tomar foto a comparar", image)
287         rawCapture.truncate(0)
288         k = cv2.waitKey(1)
289         rawCapture.truncate(0)
290         if k%256 == 27: # Presionar la tecla esc
291             break
292         elif k%256 == 32:# Presionar la tecla space
293             img_name = "desktop.jpg".format(img_counter)
294             cv2.imwrite(img_name, image)
295             print("{} written!".format(img_name))
296         elif k == ord('v'): #letra V
297             continuar=0
298             try:
299                 estado=ifSleep(image)
300             except:
301                 print("No se detectó la pantalla,intente de nuevo")
302
303             if estado==False:
304                 resultado= question()
305                 if resultado=="yes":
306                     continuar=1
307                     break
308                 else:
309                     cerrar()
310                     root = Tk()
311                     break
312
313             if estado==True:
314                 print("Si ya posee la imagen a comparar proceda a activar el sleep mode")
315                 break
316             break
317
318
319
320         if k%256 == 27:
321             print("Escape hit, closing...")
322             break
323         if continuar==1:
324             break
325         continuar==0
326
327     cv2.destroyAllWindows()
328     cerrar()
329     global motor
330     motor=0
331     #Crea la interfaz del temporizador
332     root = Tk()
333     root.geometry("300x250")
334
335     root.title("Temporizador")

```

Figura B2.8. Código utilizado para el funcionamiento del sistema diseñado (8/10). Fuente propia

```

335 root.title("Temporizador")
336
337 hour=StringVar()
338 minute=StringVar()
339 second=StringVar()
340
341 hour.set("00")
342 minute.set("00")
343 second.set("00")
344
345 hourEntry= Entry(root, width=3, font=("Arial",18,""),
346                 textvariable=hour)
347 hourEntry.place(x=80,y=20)
348
349 minuteEntry= Entry(root, width=3, font=("Arial",18,""),
350                  textvariable=minute)
351 minuteEntry.place(x=130,y=20)
352
353 secondEntry= Entry(root, width=3, font=("Arial",18,""),
354                  textvariable=second)
355 secondEntry.place(x=180,y=20)
356
357
358
359 btn = Button(root, text='Iniciar cuenta regresiva', bd='5',
360             command= submit)
361 btn.place(x = 70,y = 120)
362
363 root.mainloop()
364
365 cam.framerate=30
366 #cam.annotate_text = dt.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
367 cam.start_preview()
368 #start=time.time()
369 sleep(2)
370 start=time.time()
371 cam.start_recording('/home/pi/Desktop/video.h264')
372 print("Inicio video")
373 if motor==1:
374     print("mover motor")
375     moverServo()
376     sleep(0.5)
377     print("Inicio video")
378     sleep(3)
379     motor=0
380 cam.stop_recording()
381 cam.stop_preview()
382 print("se detuvo el video")
383
384 #guarda la imagen a comparar y el video
385 desktop = cv2.imread("desktop.jpg")
386 capture = cv2.VideoCapture('video.h264')
387 cont = 0
388 frames=[]

```

Figura B2.9. Código utilizado para el funcionamiento del sistema diseñado (9/10). Fuente propia

```

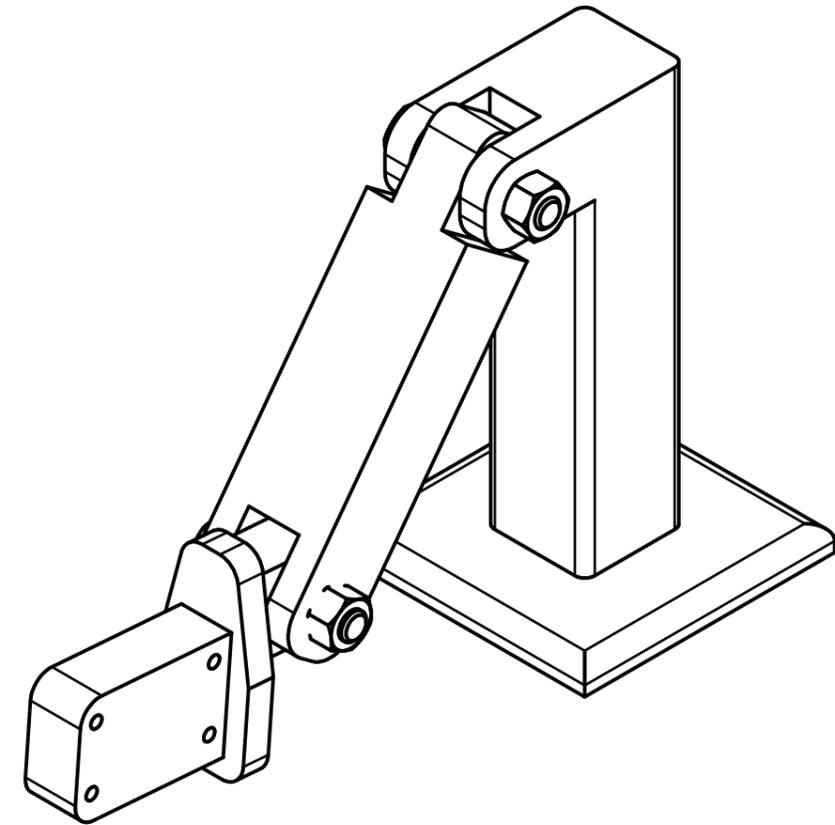
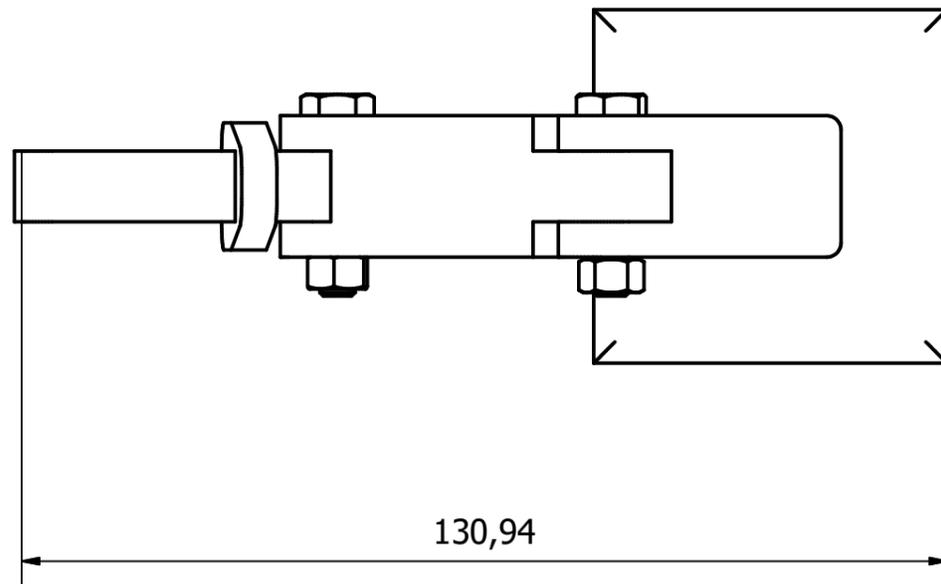
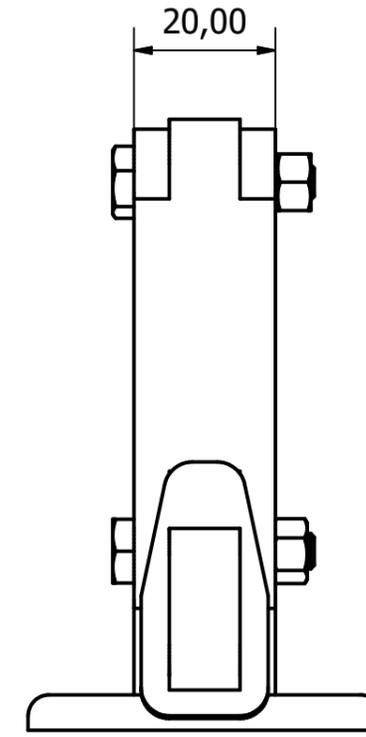
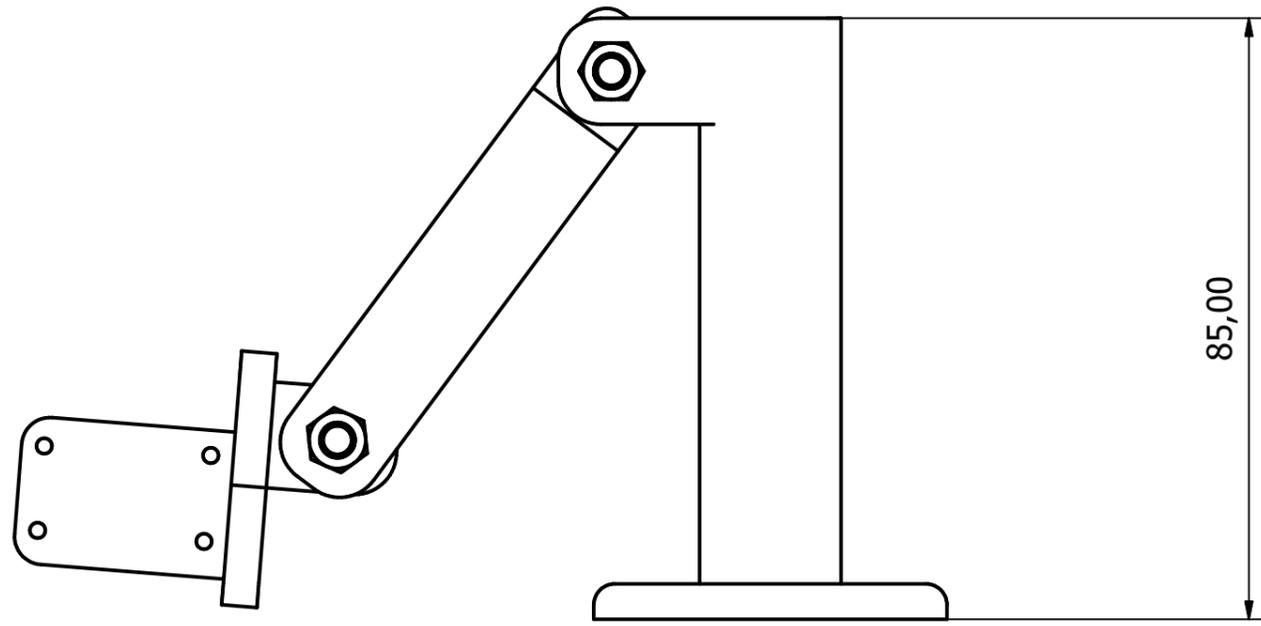
389
390 while (capture.isOpened()):
391     ret, frame = capture.read()
392     if (ret == True):
393         frames.append(frame)
394         cont=cont+1
395     else:
396         break
397
398 capture.release()
399 error=[]
400 error2=[]
401
402 cantidadFrames=0
403 for i in range(len(frames)):
404     #valor=compIma(desktop,frames[i])
405     #error.append(valor)
406     valor=otro(desktop,frames[i])
407     error2.append(valor)
408     cantidadFrames=cantidadFrames+1
409
410 #print("El valor minimo de error es:", min(error))
411 #print("Es el frame numero", np.argmin(error))
412 #print("Cantidad de frames",cont)
413 #print("Con el otro metodo")
414 print("El valor minimo de error es:", min(error2))
415 print("Es el frame numero", np.argmin(error2))
416 print("Cantidad de frames",cont)
417
418 extra=end-start
419 factor=1/30 #e lugar del 30 debe ir la cantidad de frames por segundo
420 #tiempo= factor*np.argmin(error)
421 tiempo2= factor*np.argmin(error2)
422 #print("tiempo medicion:", tiempo-0.8)
423 #print("tiempo extra:", extra)
424 #print("segundo metodo")
425 print("tiempo medicion:", tiempo2-0.8)
426 print("tiempo extra:", extra)
427

```

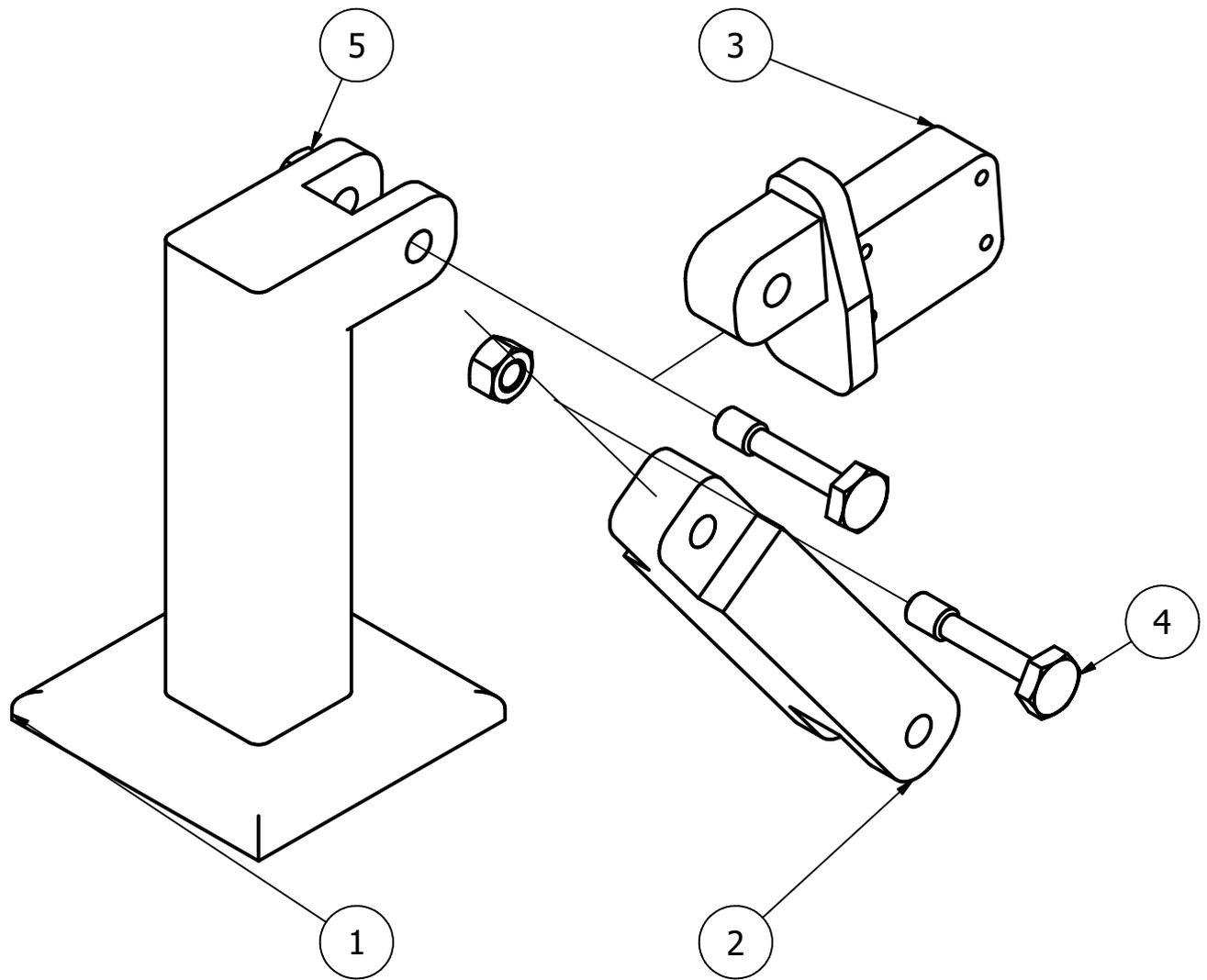
Figura B2.10. Código utilizado para el funcionamiento del sistema diseñado (10/10). Fuente propia

B Anexos

B.1 Planos piezas mecánicas



 Tecnológico de Costa Rica				ÁREA ACADÉMICA INGENIERÍA MECATRÓNICA		REVISIÓN:			
				CURSO: PROYECTO DE GRADUACIÓN		A			
DIBUJ.	M.Araya	FIRMA	FECHA	ACABADO GENERAL	DESIGNACIÓN: Ensamble soporte de la cámara		DENOMINACIÓN:		
VERIF.	P.Monge		17/05/22	N7					
APROB.			17/05/22	MATERIAL:	PROYECTO: Sistema de medición del tiempo		ESCALA: 1:1		
FABR.				Aluminio 6061					
SISTEMA:				TOLERANCIAS:	TOLERANCIA EN:	PESO: gramos		UNIDADES: mm	
	LINEAL: .X ±0.25 .X ±0.1 .XX ±0.05 .XXX ±0.010 ANGULAR: ±0.1°			mm	FORMATO: A4				
						FIRMA			



PARTS LIST			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Base estructura cámara	
2	1	Pieza media del soporte de la cámara	
3	1	Acople de la cámara	
4	2	DIN 7964 - M5 x 25 - KD2 - K - 4.8	
5	2	AS 1112 - M5 Type 5	

TEC | Tecnológico de Costa Rica

AREA ACADÉMICA INGENIERÍA MECATRÓNICA

REVISIÓN:

CURSO:
PROYECTO DE GRADUACIÓN

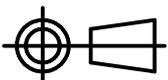
A

	NOMBRE	FIRMA	FECHA	ACABADO GENERAL
DIBUJ.	M.Araya		17/05/22	N7
VERIF.	P.Monge		17/05/22	
APROB.				MATERIAL:
FABR.				Aluminio 6061

DESIGNACIÓN:
Explosionado Soporte de la cámara

DENOMINACIÓN:

SISTEMA:



TOLERANCIAS:

LINEAL: X ±0.25
X ±0.1
.XX ±0.05
.XXX ±0.010
ANGULAR: ±0.1°

TOLERANCIA EN:

mm

PROYECTO:

Sistema de medición del tiempo

ESCALA:

1:1

FORMATO:

A4

PESO: 99 gramos

UNIDADES: mm

FIRMA

4

3

2

1

F

F

E

E

D

D

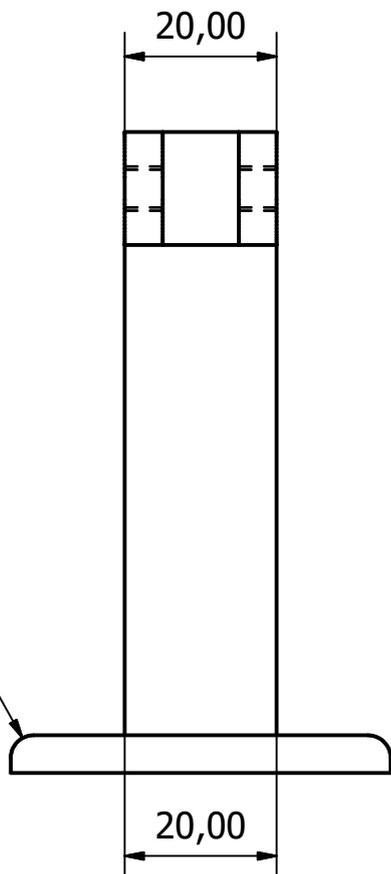
C

C

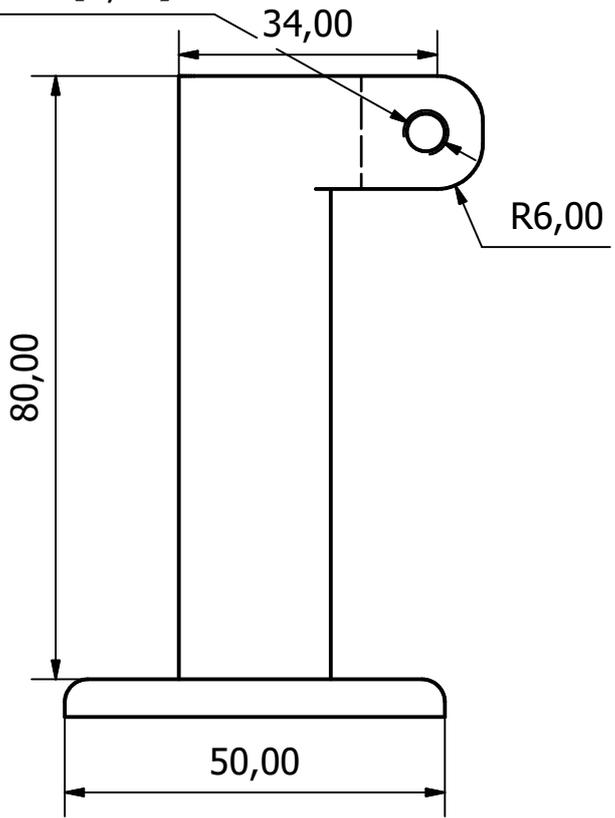
B

B

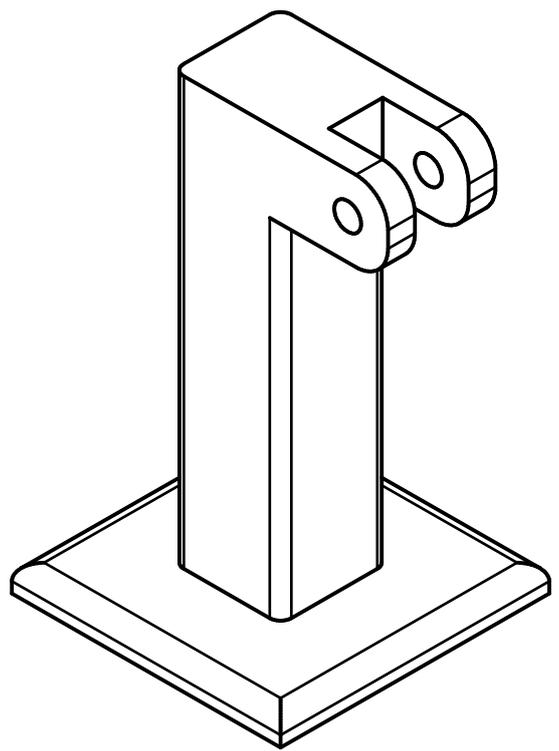
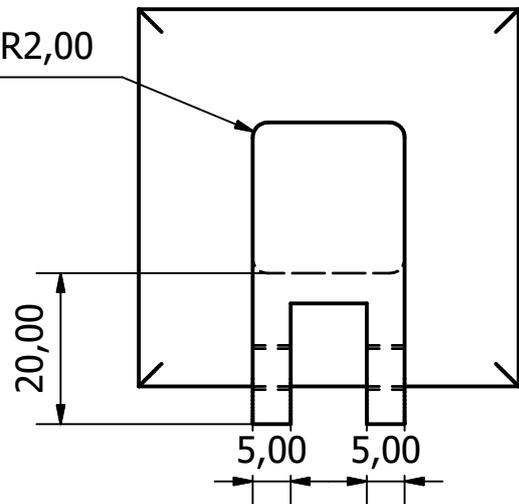
R3,00



Ø5,00 H11 (5,08 / 5,00)



R2,00



AREA ACADÉMICA INGENIERÍA MECATRÓNICA
 CURSO: PROYECTO DE GRADUACIÓN

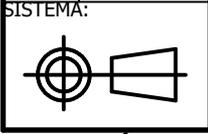
REVISIÓN:
A

	NOMBRE	FIRMA	FECHA	ACABADO GENERAL
DIBUJ.	M.Araya		17/05/22	N7
VERIF.	P.Monge		17/05/22	
APROB.				MATERIAL:
FABR.				Aluminio 6061

TOLERANCIAS:
 LINEAL: X ±0.25
 X ±0.1
 .XX ±0.05
 .XXX ±0.010
 ANGULAR: ±0.1°

DESIGNACIÓN:
Base del soporte de la cámara

DENOMINACIÓN:
 ESCALA: **1:1**
 FORMATO: **A4**



TOLERANCIA EN: mm
 PESO: 99 gramos

PROYECTO: Sistema de medición del tiempo
 UNIDADES: mm

FIRMA

4

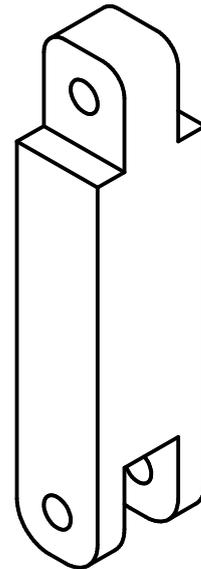
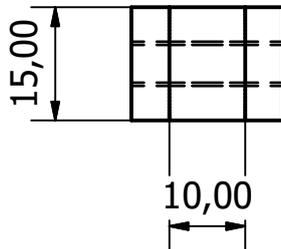
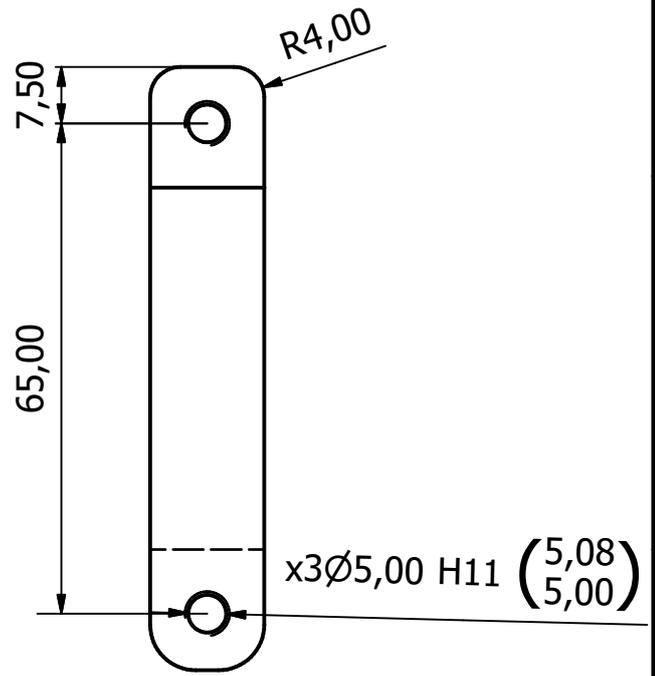
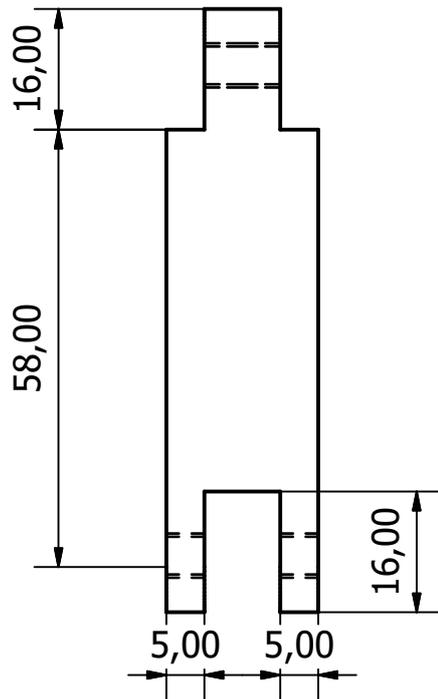
3

2

1

A

A



TEC | Tecnológico de Costa Rica

AREA ACADÉMICA INGENIERÍA MECATRÓNICA

REVISIÓN:

CURSO:
PROYECTO DE GRADUACIÓN

A

DESIGNACIÓN:
**Parte media
del soporte de
la cámara**

DENOMINACIÓN:

PROYECTO:
Sistema de
medición del tiempo

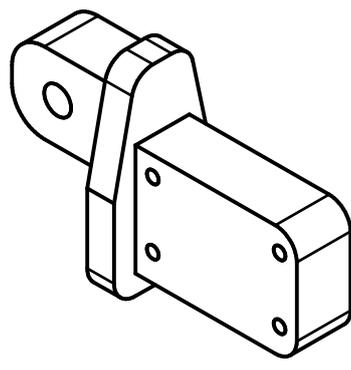
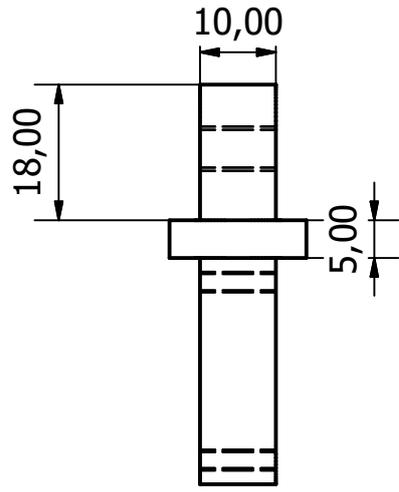
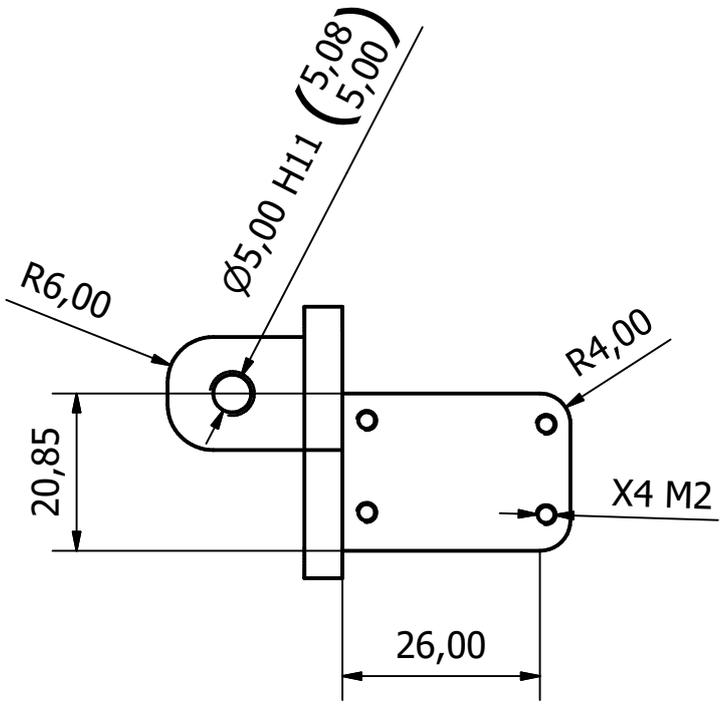
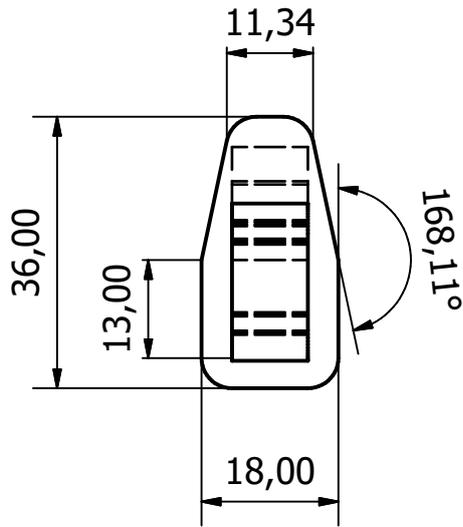
ESCALA:
1:1

FORMATO:
A4

UNIDADES: mm

FIRMA

	NOMBRE	FIRMA	FECHA	ACABADO GENERAL
DIBUJ.	M.Araya		17/05/22	N7
VERIF.	P.Monge		17/05/22	
APROB.				MATERIAL:
FABR.				Aluminio 6061
SISTEMA:	TOLERANCIAS:			TOLERANCIA EN:
	LINEAL: X ±0.25 X ±0.1 .XX ±0.05 .XXX ±0.010 ANGULAR: ±0.1°			mm
	PESO: 99			gramos



TEC | Tecnológico de Costa Rica

AREA ACADÉMICA INGENIERÍA MECATRÓNICA
CURSO: PROYECTO DE GRADUACIÓN

REVISIÓN:
A

	NOMBRE	FIRMA	FECHA	ACABADO GENERAL
DIBUJ.	M.Araya		17/05/22	N7
VERIF.	P.Monge		17/05/22	
APROB.				MATERIAL: Aluminio 6061
FABR.				
SISTEMA:	TOLERANCIAS:			TOLERANCIA EN:
	LINEAL: X ±0,25 X ±0,1 .XX ±0,05 .XXX ±0,010 ANGULAR: ±0,1°			mm
	PESO:			UNIDADES:
	gramos			mm

DESIGNACIÓN:
Acople de la cámara

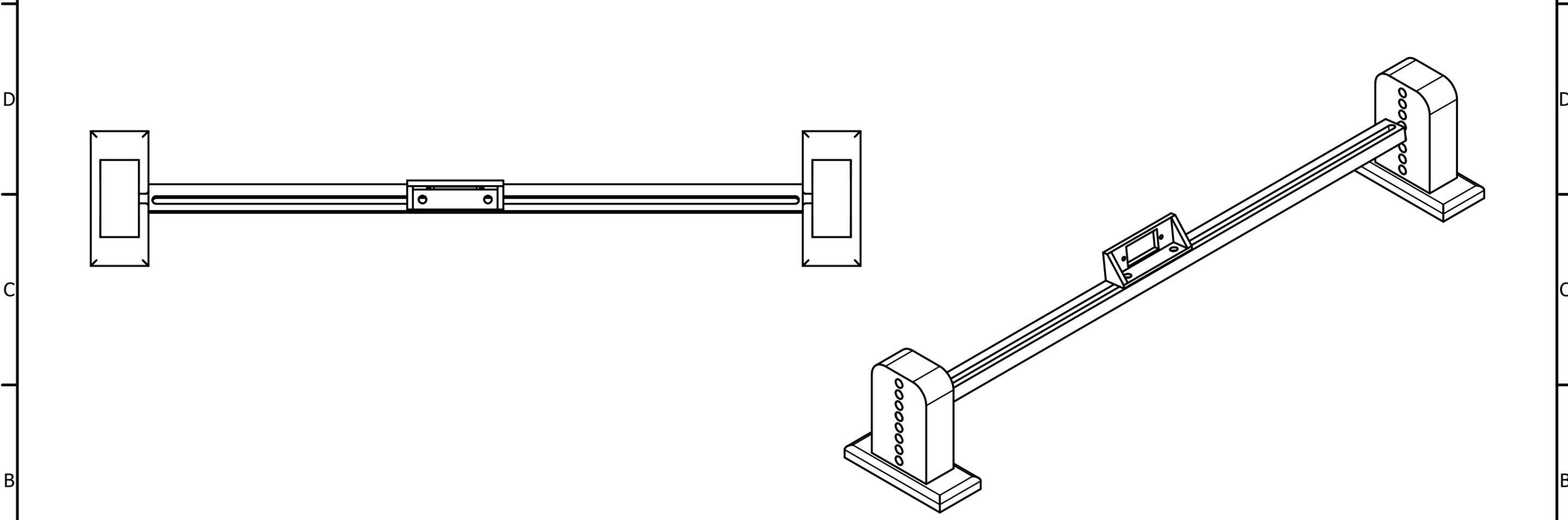
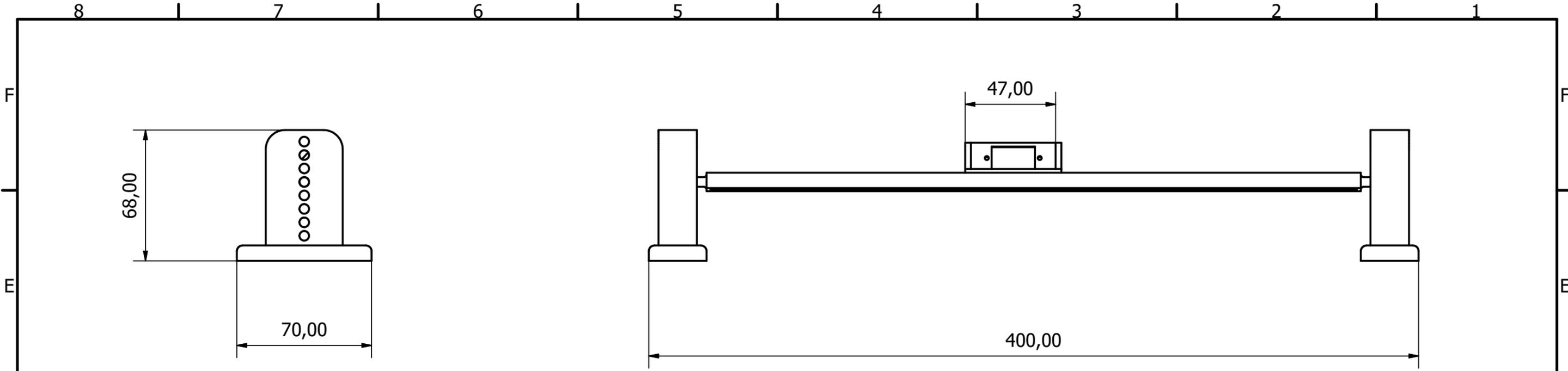
PROYECTO:
Sistema de medición del tiempo

DENOMINACIÓN:

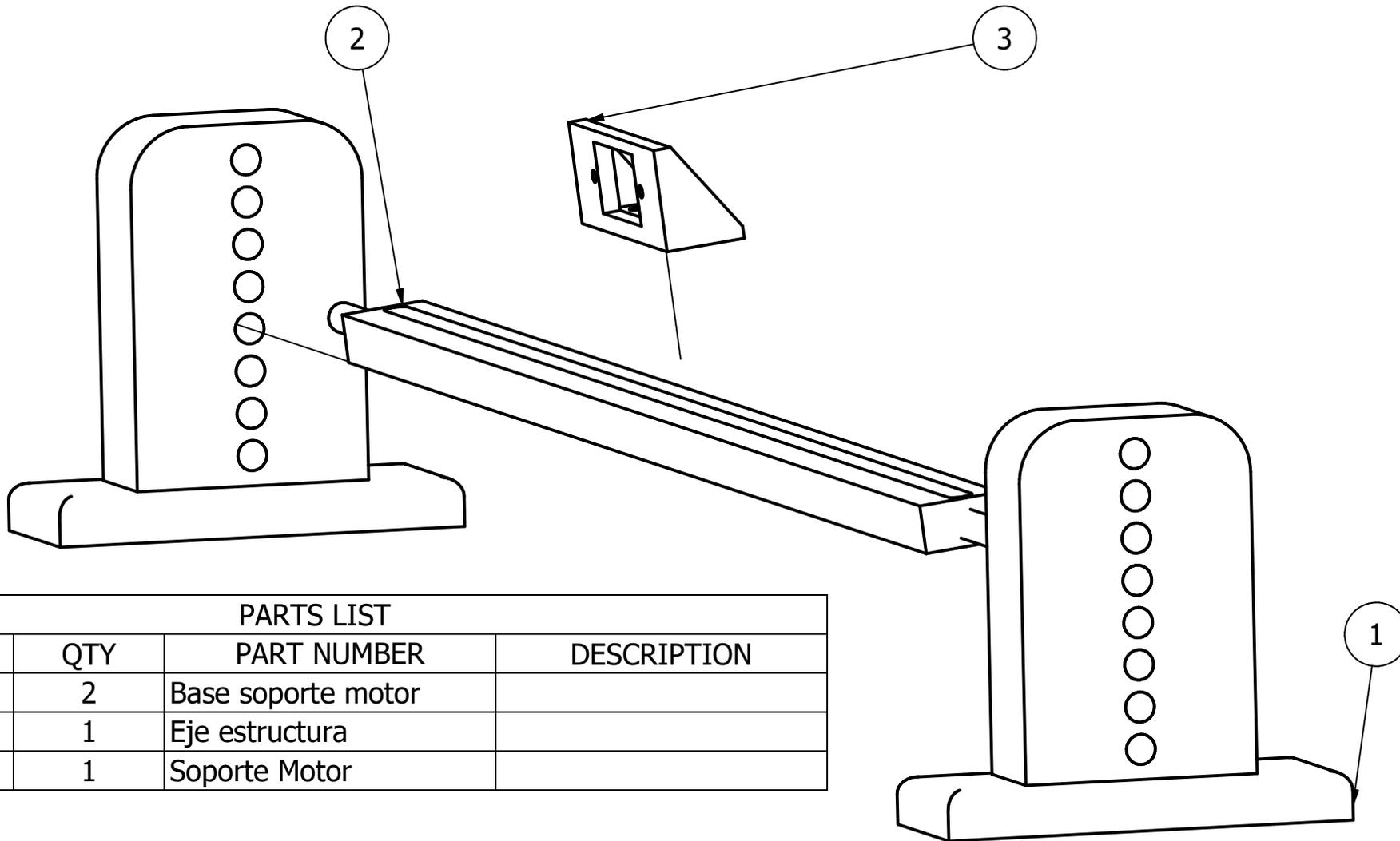
ESCALA:
1:1

FORMATO:
A4

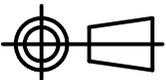
FIRMA

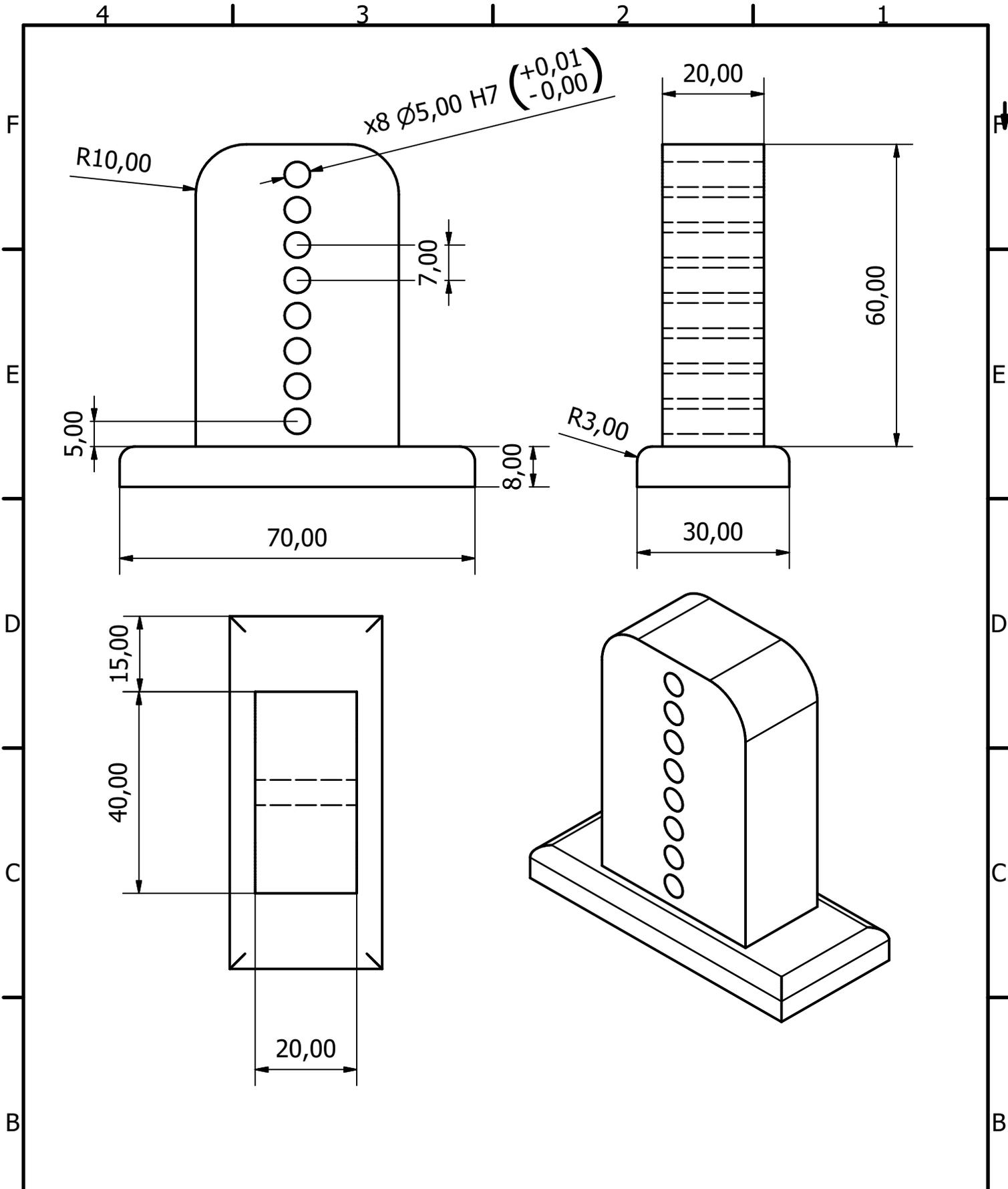


 Tecnológico de Costa Rica				AREA ACADÉMICA INGENIERÍA MECATRÓNICA		REVISIÓN:	
				CURSO: PROYECTO DE GRADUACIÓN		A	
DIBUJ. M.Araya		FIRMA		FECHA: 17/05/22		ACABADO GENERAL: N7	
VERIF. P.Monge				17/05/22		DESIGNACIÓN: Ensamble soporte servomotor	
APROB.						DENOMINACIÓN:	
FABR.						MATERIAL: Aluminio 6061	
SISTEMA:		TOLERANCIAS:		TOLERANCIA EN: mm		PROYECTO: Sistema de medición del tiempo	
		LINEAL: .X ±0.25 .X ±0.1 .XX ±0.05 .XXX ±0.010 ANGULAR: ±0.1°		PESO: gramos		UNIDADES: mm	
						ESCALA: 1:2	
						FORMATO: A4	
						FIRMA	



PARTS LIST			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	2	Base soporte motor	
2	1	Eje estructura	
3	1	Soporte Motor	

 Tecnológico de Costa Rica				AREA ACADÉMICA INGENIERÍA MECATRÓNICA		REVISIÓN:			
				CURSO: PROYECTO DE GRADUACIÓN		A			
DIBUJ. M.Araya		FIRMA		ACABADO GENERAL N7		DESIGNACIÓN: Explosionado soporte del servomotor		DENOMINACIÓN:	
VERIF. P.Monge		FECHA 17/05/22		MATERIAL: Aluminio 6061		PROYECTO: Sistema de medición del tiempo		ESCALA: 1:1	
APROB.		TOLERANCIAS: LINEAL: X ±0.25 X ±0.1 XX ±0.05 XXX ±0.010 ANGULAR: ±0.1°		TOLERANCIA EN: mm		UNIDADES: mm		FORMATO: A4	
FABR.		SISTEMA: 		PESO: 99 gramos		FIRMA			



TEC | Tecnológico de Costa Rica

AREA ACADÉMICA INGENIERÍA MECATRÓNICA
CURSO: PROYECTO DE GRADUACIÓN

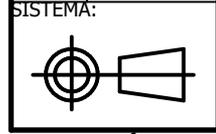
REVISIÓN: **A**

	NOMBRE	FIRMA	FECHA
DIBUJ.	M.Araya		17/05/22
VERIF.	P.Monge		17/05/22
APROB.			
FABR.			

ACABADO GENERAL: N7
MATERIAL: Aluminio 6061

DESIGNACIÓN: **Base soporte servomotor**

DENOMINACIÓN:

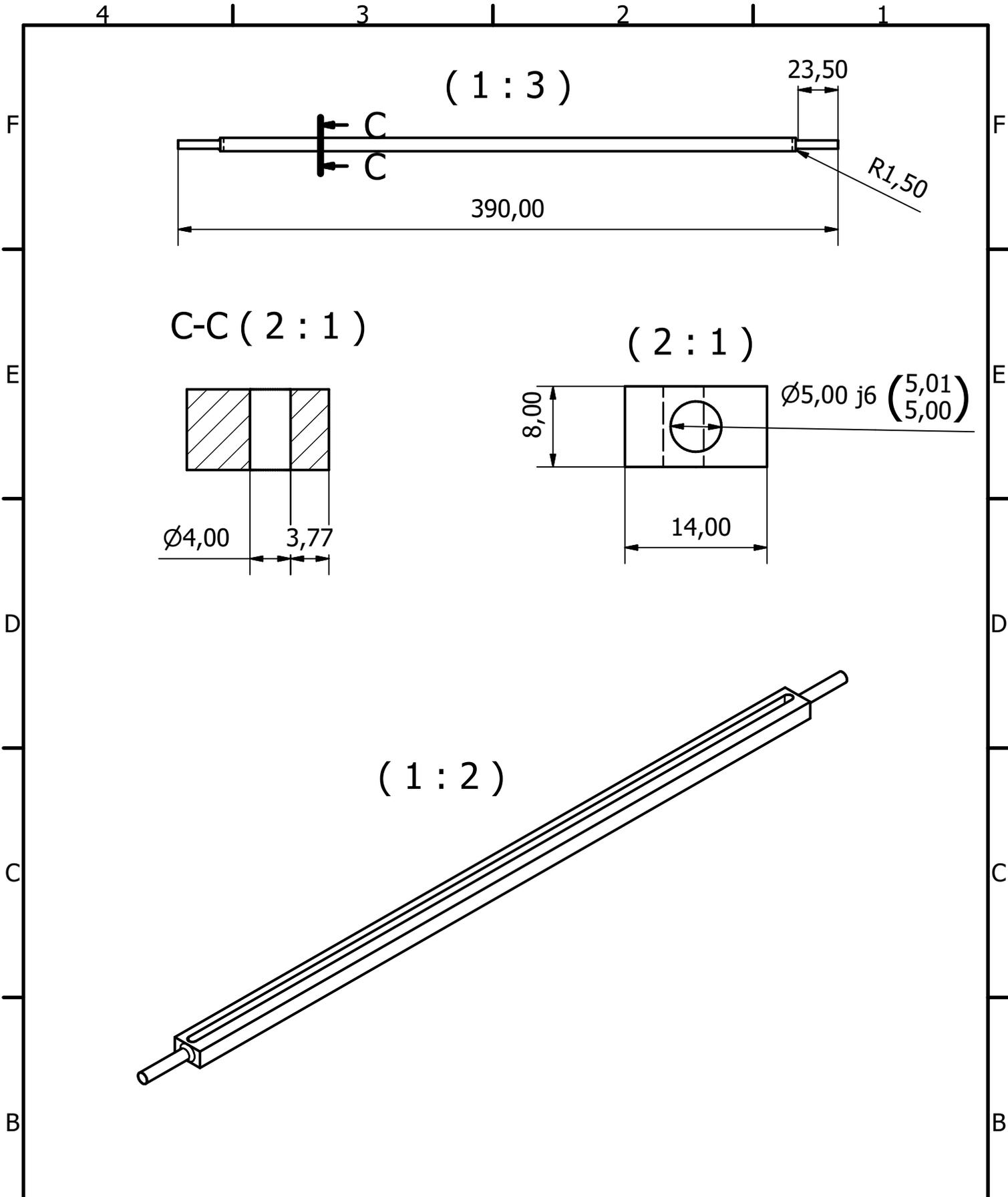


TOLERANCIAS:
LINEAL: X ±0,25
 X ±0,1
 XX ±0,05
 XXX ±0,010
ANGULAR: ±0,1°

TOLERANCIA EN: mm
PESO: gramos

PROYECTO: Sistema de medición del tiempo
UNIDADES: mm

ESCALA: **1:1**
FORMATO: **A4**
FIRMA



TEC | Tecnológico de Costa Rica

AREA ACADÉMICA INGENIERÍA MECATRÓNICA
 CURSO: PROYECTO DE GRADUACIÓN
 REVISIÓN: **A**

DIBUJ.	NOMBRE M.Araya	FIRMA	FECHA 17/05/22	ACABADO GENERAL N7	DESIGNACIÓN: Soporte servomotor	DENOMINACIÓN:	
VERIF.	P.Monge	17/05/22	MATERIAL: Aluminio 6061				
APROB.							
FABR.					PROYECTO: Sistema de medición del tiempo	ESCALA: Varios	FORMATO: A4
SISTEMA:	TOLERANCIAS: LINEAL: X ±0,25 X ±0,1 XX ±0,05 XXX ±0,010 ANGULAR: ±0,1°		TOLERANCIA EN: mm	PESO: gramos	UNIDADES: mm	FIRMA	

