

## UNIVERSIDAD POLITÉCNICA DE VICTORIA

---

### TINYML PARA LA IMPLEMENTACIÓN DE UN CUBO DE EMOCIONES

TESINA  
QUE PARA OBTENER EL GRADO DE  
INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN

PRESENTA:  
**JESÚS GUADALUPE RANGEL TURRUBIATES**

DIRECTOR  
**M.S.I. JOSÉ FIDENCIO LÓPEZ LUNA**  
CO-DIRECTOR  
**DR. SAID P. MARTAGON**  
ORGANISMO RECEPTOR  
**UNIVERSIDAD POLITÉCNICA DE VICTORIA**  
CIUDAD VICTORIA, TAMAULIPAS, DICIEMBRE DE 2023

---



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



**Tamaulipas**  
Gobierno del Estado



**Secretaría  
de Educación**



**BiS**  
UNIVERSITIES

Ciudad Victoria,  
Tamaulipas, a 8  
de agosto de  
2023

**UNIVERSIDAD POLITÉCNICA DE VICTORIA**  
**DR. SAID POLANCO MARTAGON**  
**PRESENTE.**

La Universidad Politécnica de Victoria tiene a bien presentar al alumno **RANGEL TURRUBIATES JESUS GUADALUPE**, con número de matrícula **1930156** y afiliación al seguro facultativo IMSS **04170111498**, del programa académico **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, a fin de realizar en modalidad presencial, a distancia o híbrida, según se determine de común acuerdo, sus prácticas profesionales de **ESTADÍA**, con duración de **600 horas** a partir del **4 de septiembre de 2023** al **8 de diciembre de 2023** desarrollando el proyecto **"TINYML PARA LA IMPLEMENTACIÓN DE UN CUBO DE EMOCIONES"**.

Es importante mencionar que el alumno estará comprometido al cumplimiento de los Reglamentos, Protocolos y Normas de Seguridad e Higiene que correspondan.

Hacemos de su conocimiento que previo a la conclusión del periodo de prácticas se le enviará por correo electrónico el formulario "Evaluación de desempeño del practicante"; agradecemos de antemano su valiosa respuesta, que aporta información útil para la mejora de los programas académicos que ofrece nuestra institución.

**ATENTAMENTE**

**MTRO. OTHÓN CANO GARZA**  
**DIRECTOR DE VINCULACIÓN**

**MTRO. JORGE OMAR JASSO LUNA**  
**TITULAR DEL PROGRAMA ACADÉMICO DE**  
**INGENIERÍA EN TECNOLOGÍAS DE LA**  
**INFORMACIÓN**

**UNIVERSIDAD POLITÉCNICA DE VICTORIA**

Av. Nuevas Tecnologías 5902  
Parque Científico y Tecnológico de Tamaulipas  
Carretera Victoria Soto La Marina Km. 5.5  
Cd. Victoria, Tamaulipas C.P. 87138

Tel: (834) 1711100 al 10  
Fax Ext: 5000 y 5001  
[www.upvictoria.edu.mx](http://www.upvictoria.edu.mx)



Asunto: **Carta de Aceptación**

Ciudad Victoria, Tamaulipas, a **07 Septiembre de 2023**

**M.A. OTHON CANO GARZA  
DIRECTOR DE VINCULACIÓN  
UNIVERSIDAD POLITÉCNICA DE VICTORIA  
PRESENTE.**

Por medio de la presente se hace constar que la empresa **UNIVERSIDAD POLITÉCNICA DE VICTORIA**, tiene total disponibilidad de aceptar al alumno, **Jesús Guadalupe Rangel Turrubiates** con matrícula **1930156**, del programa académico de **Ingeniería en Tecnologías de la Información**, a fin de realizar su práctica profesional de **Estadía**, colaborando con el proyecto "**TinyML para la implementación de un cubo de emociones**", para lo cual debe cumplir un total de **600** horas, en el periodo comprendido del **04 de Septiembre de 2023 al 08 de Diciembre de 2023**.

Se extiende la presente a petición del interesado, para los usos y fines que a él le convengan.

**ATENTAMENTE**



**Dr. Said Polanco Martagón  
Profesor investigador de tiempo completo**

c.c.p. Archivo.

Cd. Victoria, Tamaulipas, a 01 de Diciembre de 2023

M. A. OTHON CANO GARZA  
DIRECTOR DE VINCULACIÓN  
UNIVERSIDAD POLITÉCNICA DE VICTORIA  
PRESENTE.

Por medio del presente, se hace constar que la Universidad Politécnica de Victoria contó con el apoyo del **RANGEL TURRUBIATES JESUS GUADALUPE** con matrícula **1930156**, mediante la realización de su **Estadía** en el proyecto denominado "**TINYML PARA LA IMPLEMENTACIÓN DE UN CUBO DE EMOCIONES**" cumpliendo un total de 600 horas durante el periodo comprendido del **04 de septiembre al 08 de Diciembre 2023**.

Esperando que la presente sea de utilidad al interesado, agradezco su atención.

ATENTAMENTE  
  
Dr. Said Polanco Martagón  
Profesor - Investigador  
Universidad Politécnica de Victoria



## CARTA DE ACEPTACIÓN DEL DOCUMENTO PARA SU IMPRESIÓN

Cd. Victoria, Tamaulipas a 4 de Diciembre de 2023

Jesús Guadalupe Rangel Turrubiates  
PRESENTE

Le comunico que el Programa Académico de Ingeniería en Tecnologías de la Información le ha otorgado la autorización para la impresión de su Tesina de Estadía Práctica cuyo título es:

**TinyML para la implementación de un cubo de emociones**  
ATENTAMENTE

---

M.S.I. José Fidencio López Luna  
ASESOR INSTITUCIONAL

c.c.p Director de programa académico

## EVALUACIÓN DE ESTADÍA

### Rúbrica para evaluación de la presentación y el reporte de estadía

Nombre del alumno: JESÚS GUADALUPE RANGEL TURRUBIATES

Calificación final: \_\_\_\_\_

Periodo: AGOSTO-DICIEMBRE 2023

Ponderación	Aspecto a Evaluar	Competente 10	Independiente 9	Básico Avanzado 8	No Competente 5
40	Resultados y Actividades	Estrechamente relacionados al perfil de egreso de su programa académico	Parcialmente relacionados al perfil de egreso de su programa académico	Escasamente relacionados al perfil de egreso de su programa académico	Escasamente relacionados al perfil de egreso de su programa académico
30	Exposición de las actividades de la estadía	Detalladas y sustentadas con respecto a los resultados que se obtuvieron	Detalladas y sustentadas parcialmente con respecto a los resultados que se obtuvieron	Detalladas parcialmente con respecto a los resultados que se obtuvieron	Detalladas escasamente con respecto a los resultados que se obtuvieron
10	Material visual Lenguaje verbal	Uso el lenguaje y la terminología apropiadas; El material visual está organizado, adecuado y suficiente	Uso el lenguaje y la terminología apropiadas El material visual está parcialmente organizado y es suficiente	Uso el lenguaje y la terminología son parcialmente apropiadas; El material visual está parcialmente organizado y es suficiente	Uso el lenguaje y terminología es inapropiado; El material visual no está organizado y es insuficiente
10	Exposición en Idioma Inglés	Pronunciation is clear so language is easily understood (2.5) Uses fluent connected speech, occasionally disrupted by search for correct form of expression (2.5) Uses topic related vocabulary without problems (2.5) Responds to questions using varied and descriptive vocabulary and language structures (2.5)	Pronunciation is understandable, but there are slight errors (2.25) Speech is connected but frequently disrupted by search for correct form of expression (2.25) Uses some topic related vocabulary sufficient to communicate ideas (2.25) Responds to questions using simple but accurate vocabulary and language structures (2.25)	Pronunciation is understandable most of the time, marked native accent and many errors (2) Speaks with simple sentences, sometimes not connected, but is understood (2) Uses basic vocabulary to communicate ideas (2) Partly responds to simple questions, with limited vocabulary and language structures (2)	Pronunciation makes language very difficult to understand (1) Uses one-word/two-word utterances (1) Unable to communicate ideas due to lack of vocabulary (1) Uses isolated words or sentence fragments to respond to questions (1)
5	Respuesta a los cuestionamientos de los evaluadores	Clara y satisfactoria	Clara y parcialmente satisfactoria	Clara e insuficiente	Confusa e insuficiente
5	Autorización de tesina en tiempo y forma	Presenta en tiempo y forma	Presenta en tiempo y forma con la mayoría de requerimientos solicitados	Presenta en tiempo y con algunas limitantes de los requerimientos solicitados.	Presenta fuera de tiempo y con los mínimos requerimientos solicitados.

M.S.I. José Fidencio López Luna

ASESOR INSTITUCIONAL

Dr. Marco Aurelio Nuño Maganda

EVALUADOR

EVALUADOR DE INGLÉS



## REGISTRO DE EVALUACIÓN DE EXPOSICIÓN DE ESTADÍA

Siendo las 09:00 horas del día 11 de Diciembre de 2023, el alumno **Jesús Guadalupe Rangel Turrubiates**, del programa académico **Ingeniería en Tecnologías de la Información**, con matrícula **1930156**, presentó la exposición de la estadía realizada durante el cuatrimestre **Agosto-Diciembre 2023**, en la **Universidad Politécnica de Victoria**, con el proyecto titulado **TinyML para la implementación de un cubo de emociones**.

Una vez concluido el proceso de evaluación, y con base a la rúbrica establecida para éste propósito, se determina que la calificación de la estadía es \_\_\_\_\_.

---

M.S.I. José Fidencio López Luna  
ASESOR INSTITUCIONAL

---

Dr. Marco Aurelio Nuño Maganda  
EVALUADOR

---

EVALUADOR DE INGLÉS

## Agradecimientos

**A mis padres.** El Sr. Evodio Rangel Campos y la Sra. Maria Araceli Turrubaites López quienes juntos me compartieron sus enseñanzas, aprendizajes y valores bajo los cuales formé mi moral y modo de pensamiento, por ser los pilares del hogar, proveer para mi en la medida de lo posible y acompañarme siempre que lo necesitaba.

**A mis hermanos** Armando, Cristian Sarai y Eder Alberto por compartir su tiempo, sus recursos y su vida conmigo, permitirme aprender junto con ellos y a su vez ofrecer un consejo o experiencia si alguna vez les preguntaba, esperando yo también poderles haber aportado algo de provecho.

**A mis amigos y compañeros** por hacer más amenas las actividades del día, compartir sus opiniones y pasatiempos así como trabajar en equipo junto conmigo dentro y fuera del ámbito estudiantil.

**Al profesor** Said Polanco Martagon por la oportunidad de colaborar en este proyecto, su asesoramiento y apoyo durante el periodo de estadías que me permitió tener aprendizaje de actividades que desconocía del desarrollo de proyectos.

## Resumen

El autismo es un conjunto de alteraciones neurológicas que intervienen en el cerebro de algunos niños, provocando problemas en la comunicación e interacción social de conductas a intereses distintivos. No hay una causa concreta que sea atribuida al Trastorno de Espectro Autista, se cree que involucra factores genéticos, neurológicos, bioquímicos y ambientales que pueden dar origen al padecimiento. Alrededor de 400 mil infantes tienen autismo, es decir 1% de la población de niños en México.

Debido al desarrollo tecnológico se han desarrollado herramientas en pos de apoyar la terapia de niños autistas. Este proyecto deriva en un prototipo descrito como un cubo con el cual niños autistas puedan expresar sus emociones. Esto se logra mediante la identificación de una cara activa en el cubo la cual representa una emoción particular, de esta manera tanto el niño como sus padres pueden tener un mejor entendimiento del comportamiento presentado.

En esta etapa la tarea principal del prototipo fue hacer inferencias sobre la cara activa del cubo de emociones mediante técnicas de aprendizaje máquina, modelos de aprendizaje e internet de las cosas para la creación de un sistema embebido y posteriormente enviarlas por mensaje a una aplicación móvil perteneciente al padre o tutor del niño. Lo anterior se da usando una placa WEMOS S2 MINI donde se codifica un árbol de decisiones con datos provenientes de Switches de inclinación y un sensor MPU6050.

**Palabras clave:** Tecnología, Internet de las cosas, Arbol de decisiones, Autismo, Sistema embebido, Aprendizaje maquina.

## Summary

Autism is a group of neurological alterations that affect the brain of some children causing communication problems and social interactions of behavior on distinctive interests. There is not a definitive cause attributed to Autism Spectrum Disorder, it's believed to be related to genetic, neurobiological, biochemical and environmental factors that could cause this condition. It's estimated that around 400 million infants have autism, that is 1% of the child population in Mexico.

Due to technological development tools have been created with the purpose of supporting autistic children therapy. This project leads to creating a prototype described as a box which autistic children can use to express their emotions. Achieved by identifying an active face on the box which represents a particular emotion, this way both the child and their parents can have a better understanding of their current behaviour.

On this stage the main task of the prototype is to infer the active face of the emotion box using Machine Learning Techniques, Learning Models and Internet of Things to create an embedded system, then send it on a message to a Telegram chat on the parent's phone. This achieved by using a WEMOS S2 mini board where the decision tree is coded while Tilt Switches and a MPU6050 sensor provide the input values.

**Keywords:** Technology, Internet of things, Decision tree, Autism, Embedded system, Machine learning.

# Índice

<b>Agradecimientos</b>	<b>vii</b>
<b>Resumen</b>	<b>viii</b>
<b>Summary</b>	<b>ix</b>
<b>Índice</b>	<b>x</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Definición y justificación del proyecto . . . . .	1
1.2. Objetivo general . . . . .	1
1.3. Objetivos específicos . . . . .	2
1.4. Alcances y limitaciones . . . . .	2
1.4.1. Alcances . . . . .	2
1.4.2. Limitaciones . . . . .	2
1.5. Antecedentes de la empresa . . . . .	2
1.5.1. Misión . . . . .	2
1.5.2. Visión . . . . .	3
1.5.3. Política de calidad . . . . .	3
1.6. Antecedentes del proyecto . . . . .	3
1.7. Organización del documento . . . . .	3
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Tecnologías de Información y comunicación . . . . .	5
2.1.1. Machine Learnig . . . . .	5
2.1.2. Sistemas de aprendizaje de ML . . . . .	6
2.1.3. Clasificación en Machine learning . . . . .	7
2.1.4. Arbol de Decisiones . . . . .	8
2.1.5. Maximizando la ganancia de información . . . . .	9
2.2. Trastorno del espectro autista . . . . .	10
2.2.1. Identificando el autismo en niños . . . . .	10
2.3. Estados emocionales . . . . .	11
2.3.1. Emociones fundamentales . . . . .	12
2.4. Internet of Things (IOT) . . . . .	13
2.4.1. TinyML en recolección de datos . . . . .	14
2.5. Manofactura aditiva . . . . .	14
2.5.1. Prototipado rápido en electrónica . . . . .	15
2.5.2. Diseño Asistido por Computadora . . . . .	15
2.5.3. Fritzing . . . . .	15
2.5.4. Placa WEMOS s2 mini . . . . .	16
2.5.5. Switch de inclinación . . . . .	16
2.5.6. Sensor de aceleracion e inclinacion MPU6050 . . . . .	16
2.5.7. Arduino IDE . . . . .	16
2.5.8. Jupyter LAB . . . . .	17

2.5.9. Pandas . . . . .	17
2.5.10. NumPY . . . . .	17
2.5.11. MatplotLIB . . . . .	17
2.5.12. Scikit Learn . . . . .	18
2.5.13. Librería ESP32 Arduino . . . . .	18
<b>3. Trabajos Relacionados</b>	<b>19</b>
3.1. Reconocimiento Automático de emociones en Interacciones Robot-Niño para el tratamiento de ASD . . . . .	19
3.2. Reconocimiento facial en tiempo real de niños con autismo basado en Deep Learning y IoT . . . . .	20
3.3. Un nuevo modelo de ML basado en inducción de reglas para detección del autismo . . . . .	21
3.4. Prototipo de cubo de sentimientos en comparación con otros proyectos relacionados . . . . .	22
<b>4. Propuesta de prototipo</b>	<b>24</b>
4.1. Áreas de oportunidad del prototipo anterior . . . . .	24
4.1.1. Breve descripción del prototipo previo . . . . .	24
4.1.2. Necesidades y mejoras del prototipo previo . . . . .	26
4.2. Hardware utilizado . . . . .	26
4.2.1. Placa WEMOS S2 mini . . . . .	26
4.2.2. Swtich de inclinación SW-5200 . . . . .	27
4.2.3. Sensor MPU6050 . . . . .	28
4.2.4. Caja contenedora . . . . .	29
4.3. Tecnologías y librerías utilizadas . . . . .	30
4.3.1. FreeCad . . . . .	30
4.3.2. Arduino IDE . . . . .	31
4.3.3. Jupyter Notebook . . . . .	32
4.3.4. Pandas . . . . .	33
4.3.5. Numpy . . . . .	34
4.3.6. Matplotlib . . . . .	35
4.3.7. Scikit learn . . . . .	36
4.3.8. Arduino ESP32 . . . . .	37
4.3.9. Universtal Telegram Bot . . . . .	37
<b>5. Construcción e implementación del prototipo</b>	<b>40</b>
5.1. Diseño de caja contenedora vía software CAD . . . . .	40
5.2. Ensamblado de componentes electrónicos . . . . .	42
5.3. Método de entrada y muestra de datos . . . . .	43
5.4. Creación de Dataset a partir de salida de datos . . . . .	46
5.5. Implementación de modelo de aprendizaje de árbol y construcción del clasificador de árbol . . . . .	49
5.5.1. Transformando el modelo de clasificador de árbol . . . . .	54
5.6. Descomposición del clasificador de árbol en estructuras If-Else anidadas . . . . .	57

5.7.	Script de predicción de cara para placa WEMOS S2 mini . . . . .	59
5.7.1.	Archivo S2mini_sideclassifier_TelegramBot_Candidate.ino . . . . .	59
5.7.2.	Función setup() . . . . .	61
5.7.3.	Función calibrateGyro() . . . . .	62
5.7.4.	Función loop() . . . . .	63
5.7.5.	Función handleNewMessages() . . . . .	64
5.7.6.	Función prediction_active() . . . . .	65
5.7.7.	Función side_pred() . . . . .	68
<b>6.</b>	<b>Resultados</b>	<b>70</b>
6.1.	Funcionamiento general del cubo de emociones . . . . .	70
6.2.	Uso del cubo mediante comandos de Telegram . . . . .	71
6.3.	Entrada de datos de los sensores . . . . .	71
6.4.	Predicción y muestra de cara activa . . . . .	71
<b>7.</b>	<b>Conclusiones</b>	<b>73</b>
7.1.	Conclusión general del proyecto TinyML para la implementación de cubo de emociones . . . . .	73
7.2.	Conclusiones por objetivo establecido . . . . .	73
7.3.	Conclusiones personales . . . . .	74
7.4.	Áreas de oportunidad . . . . .	74
<b>Índice de figuras</b>		<b>76</b>
<b>Índice de tablas</b>		<b>78</b>
<b>Índice de algoritmos</b>		<b>79</b>
<b>Referencias</b>		<b>80</b>

# 1. Introducción

El aprendizaje máquina es una herramienta poderosa para la detección de señales (Carbonell et al., 1983), procesamiento de información y protección de datos mediante técnicas como las redes neuronales o modelos computacionales. Tiny ML (Schizas et al., 2022) es un término que, contrario al Machine Learning regular hace hincapié en un eficiente consumo de energía y espacio en la forma de microcontroladores (Doyu et al., 2021) y sistemas embebidos para realizar tareas particulares relacionadas a recolección de datos biométricos, entrenamiento de modelos y análisis de patrones de comportamiento.

Algunos ejemplos de áreas donde se aplica el uso de TinyML se dan en medicina especializada donde modelos matemáticos entrenados ayudan a determinar la posibilidad de que una radiografía contenga señales de cáncer en un paciente (Tsuneki, 2022), y psicología donde se analizan patrones de comportamiento y modelos estadísticos para asistir a profesionales de la salud al momento de hacer diagnósticos (Sleek, 2022), determinar posibles causas de un padecimiento mental o monitorear el estado anímico de un paciente vía remota con el fin de comprender su conducta.

Utilizando sistemas embebidos es posible llevar a cabo la recolección de datos, entrenamiento de modelos y supervisión de resultados dentro de dispositivos de bajo consumo energético con el propósito de auxiliar en el tratamiento de personas con padecimientos que afecten la capacidad de comunicación como el autismo a comunicar sus emociones y mejorar su calidad de vida.

## 1.1. Definición y justificación del proyecto

Los síntomas que presenta un niño diagnosticado como parte del espectro autista desembocan en dificultades para comunicarse, bajo rendimiento académico y una mala calidad de vida, el proyecto trae a la mesa un prototipo de sistema embebido con el cual detectar los cambios de emociones de un niño autista sobre un periodo de tiempo por medio de un modelo de Machine Learning denominado Regresión Lineal ya entrenado con valores correspondientes a una cara activa, es decir, la cara del cubo que muestra la emoción actual del usuario. Lo anterior cargado en un sistema embebido en una placa ESP32 S2 MINI en conjunto con un sensor dotado de acelerómetro y giroscopio MPU6050 e interruptores de inclinación

El uso de este sistema embebido hace posible la recolección de datos del usuario para estudiar su comportamiento y cambios de estados emocionales, dicha información es útil para futuras observaciones de un solo individuo. Todo contenido contenido en un cubo diseñado por computadora y producido por medios de tecnología aditiva considerando el espacio y rango de movimiento de los componentes utilizados para su funcionamiento.

## 1.2. Objetivo general

Implementar algoritmos de aprendizaje automático en un micro controlador ESP32 S2 MINI para detección del estado de ánimo de un niño con autismo.

## 1.3. Objetivos específicos

1. Mejorar un prototipo existente de un cubo de sentimientos.
2. Obtener un dataset para clasificación de sentimientos.
3. Implementar algoritmos de ML para clasificación de sentimientos.
4. Implementar algoritmos utilizando librerías propias de Tiny ML
5. Escritura de tesina.

## 1.4. Alcances y limitaciones

### 1.4.1. Alcances

Como un proyecto de carácter académico la principal visión para el sistema embebido es por sobre todo que produzca los datos esperados en la propia planeación de actividades, esto mediante la mejora de un prototipo ya existente de la misma línea, además de esto, que el cubo sea capaz de operar independientemente de una computadora para recolectar datos. Como resultado del periodo de trabajo debe existir una mejora de prototipo de cubo de emociones con el cual realizar pruebas con un grupo limitado de usuarios objetivo siendo estos niños autistas, con la intención de evaluar aspectos funcionales directos del prototipo, por ejemplo el desempeño de varios modelos de Machine Learning además del modelo seleccionado, la reacción de los componentes y materiales a movimientos bruscos y la efectividad del diseño físico de la caja.

### 1.4.2. Limitaciones

Algunas de las limitaciones presentadas rondan alrededor de la falta de insumos, materiales y herramientas necesarias para el proyecto, como por ejemplo la poca disponibilidad de Impresora 3D y filamento plástico así como el poco dominio de programas de modelado asistido por computadora limitando la fase de prototipado de piezas.

## 1.5. Antecedentes de la empresa

La Universidad Politécnica de Victoria es una universidad que fue fundada en el año de 2006 por iniciativa del gobierno del estado de Tamaulipas, por lo tanto se trata de una institución aun en crecimiento que tiene como objetivo ofrecer a los jóvenes del estado educación de la más alta calidad que ayude a la mejora de la sociedad a través de la formación de profesionistas comprometidos con su entorno. Su lema es “Construyendo conciencia” (Mextudia, 2019).

### 1.5.1. Misión

Contribuir al progreso de México mediante la formación de profesionales altamente calificados, con sentido humano y reconocida capacidad científico – tecnológica para participar en la solución de problemas de alto impacto social.

### 1.5.2. Visión

Ser una institución de educación superior reconocida nacional e internacionalmente por el excelente desempeño profesional de sus egresados, la capacidad y compromiso de su capital humano, la calidad de sus programas académicos y su aportación científico – tecnológica al progreso de Tamaulipas y de México.

### 1.5.3. Política de calidad

Ser una institución de educación superior que forma profesionistas competentes para el desarrollo científico-tecnológico de la región, mediante programas académicos de calidad.

## 1.6. Antecedentes del proyecto

El desarrollo del proyecto actualmente presentado funge como continuación de otro prototipo anteriormente trabajado, en palabras más exactas, una mejora de los principios ya manejados. Esta se conforma de varias tapas de madera de aglomerado contrachapada MDF cortada mediante maquinaria de corte láser en patrón dentado de tal manera que al embonar dichas tapas se forme una caja que alberga los componentes, una de estas se mantiene suelta haciendo la función de dar acceso al interior del cubo.

El propósito del prototipo trabajado en periodos de desarrollo pasados fue ser una prueba de los conceptos bajo los cuales la idea de la caja de emociones trabaja además de fungir como modelo rustico para la colocación aproximada de los componentes, estimaciones de tamaño, resistencia de materiales y viabilidad del diseño general.

## 1.7. Organización del documento

Esta tesina se estructura de la siguiente manera:

- **Capítulo 1 - Introducción:** En este capítulo se especifican los objetivos del proyecto, antecedentes, alcances, limitaciones e información de la organización para la cual se desarrolla el prototipo.
- **Capítulo 2 - Marco teórico** En este capítulo se incluyen conceptos e ideas generales para contextualizar y comprender el proyecto.
- **Capítulo 3 - Trabajos relacionados** En este capítulo se muestran documentos y fuentes de información con puntos similares al proyecto.
- **Capítulo 4 - Propuesta de prototipo** En este capítulo se muestra de manera específica las técnicas, procesos, y tecnologías utilizadas durante la elaboración del prototipo
- **Capítulo 5 - Resultados** En este capítulo se muestran los resultados obtenidos durante la realización del prototipo y evaluación de los alcances obtenidos.

- **Capítulo 6 - Conclusiones** En este capítulo se habla de las conclusiones a las que se llegó durante la realización del proyecto, así como conclusiones personales. Además de esto se habla de áreas de oportunidad e implementaciones futuras para el proyecto.

## 2. Marco Teórico

En este capítulo se incluyen conceptos e ideas generales para contextualizar y comprender el proyecto.

### 2.1. Tecnologías de Información y comunicación

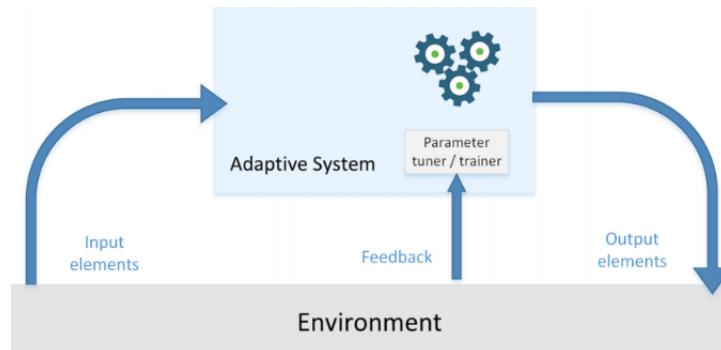
Las tecnologías de la información y comunicación, abreviadas como TICs derivan con base en los avances científicos que se den en los campos de la informática, tecnología y telecomunicaciones (Ayala y Gonzales Sánchez, 2015). Se trata del conglomerado de metodologías y dispositivos que permite a un usuario acceder, producir y tratar y comunicar piezas de información en varias presentaciones como lo son texto, imagen, sonido o video.

De acuerdo a Ayala y Gonzales Sánchez (2015), Hoy en día las TICs forman parte de todos los niveles de la sociedad, desde corporaciones de gran tamaño y cientos de trabajadores, organizaciones gubernamentales y sus administraciones, centros educativos, organismos de investigación hasta organizaciones más hogareñas.

#### 2.1.1. Machine Learning

Partiendo de la definición de “aprendizaje” de (Bonaccorso, 2017) como la habilidad de realizar ajustes dependiendo de la reacción al estímulo externo del ambiente dado la capacidad de recordar experiencias pasadas. Se puede decir que el Machine Learning o Aprendizaje Máquina es el conjunto de metodologías y técnicas para incrementar dicha capacidad de un sistema de información para adaptarse al cambio, esto desde el punto de vista de la ingeniería. Acorde a Bonaccorso (2017). En la figura 1 se presenta una abstracción de un sistema adaptativo en Machine Learning.

Figura 1: Representación de sistema adaptativo



Adaptado de Bonaccorso (2017)

El objetivo principal del ML ronda alrededor de estudiar, ingeniar y mejorar modelos matemáticos (Bonaccorso, 2017) que puedan ser entrenados con datos relacionados al contexto en el que se encuentra con el propósito de inferir un resultado futuro y tomar decisiones sin necesidad de conocer la totalidad de los datos en el ambiente.

Bajo estas definiciones un agente (entidad que recibe información externa y toma la mejor decisión para alcanzar un objetivo específico y observa los resultados de dicha decisión) trata de determinar la mejor decisión basándose en métodos de probabilidad y distribución (ya sea como un número o decisión). En base a esto, cada que el agente se equivoca mas sera probable que tenga éxito en iteraciones posteriores tomando como pivote los errores previos Bonaccorso (2017).

### 2.1.2. Sistemas de aprendizaje de ML

Existe una diversa cantidad de modelos matemáticos para el aprendizaje automático dentro del Machine Learning (Ayodele, 2010), los tipos de sistemas de aprendizaje más comunes son descritos en la tabla 1:

Tabla 1: Tipos de algoritmos de aprendizaje.

Nombre	Descripción
Aprendizaje supervisado	El algoritmo genera una función que mapea entradas a salidas deseadas. Una formulación estándar de la tarea de aprendizaje supervisado es la clasificación donde es necesario que la gente aprenda según aproximación del comportamiento , esta función relaciona un vector hacia diferentes clases al observar a distintos ejemplos de entrada y salida de la función.
Aprendizaje no supervisado	En este se modela un set de entradas, ejemplos con clasificaciones ya asignadas no están disponibles.
Aprendizaje semi-supervisado	En este se combinan elementos del dataset con y sin clase asignada para generar clasificador apropiado.
Aprendizaje por refuerzo	Donde el algoritmo aprende una mecánica de cómo actuar cada que se hace una observación de los datos. Toda acción tiene un impacto en el ambiente externo, a su vez que el ambiente provee retroalimentación que guían el aprendizaje del algoritmo.
Aprendizaje en Batch	El Batch Learning es un sistema incapaz de aprender de manera incremental, este debe ser entrenado usando todos los datos disponibles. Primero el modelo es entrenado y después puesto en producción, mismo que se llama offline learning.
Aprendizaje Online	En el Online Learning, el sistema es entrenado de manera incremental, esto se da al alimentarlo de instancias de datos secuenciales, ya sea de manera individual o en conjuntos pequeños llamados mini-batches.

Adaptado de Ayodele (2010)

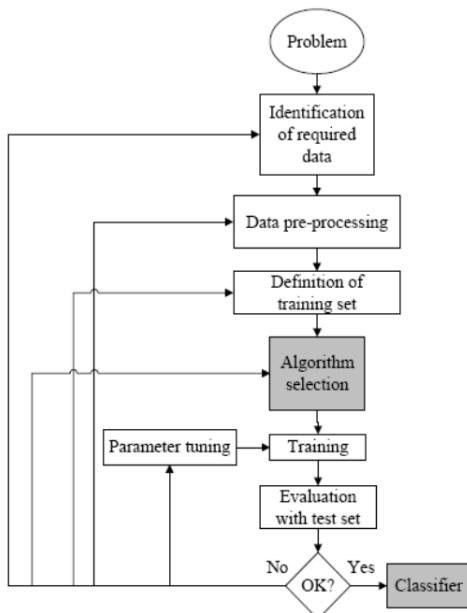
### 2.1.3. Clasificación en Machine learning

Un modelo de aprendizaje necesita aprender o aproximar un comportamiento (Osisanwo et al., 2017), es decir, debe conformar una función donde relacione un vector particular a una de

entre varias clases mediante la observación de entradas y salidas de una función. El aprendizaje máquina inductivo es el proceso donde se aprenden una serie de reglas provenientes de un grupo de instancias, dicho de otra manera, crear un clasificador que sea capaz de generalizar nuevas instancias.

En la figura 2 se muestra el proceso mediante el cual se aplica el aprendizaje supervisado a problemas del mundo real:

Figura 2: Proceso del aprendizaje supervisado en Machine Learning

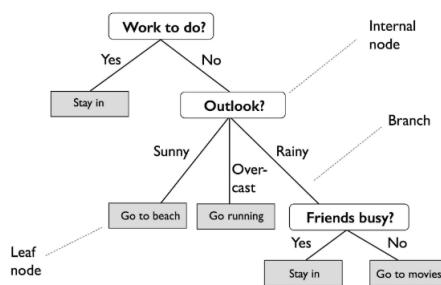


Adaptado de Osisanwo et al. (2017)

#### 2.1.4. Árbol de Decisiones

Los clasificadores de árbol de decisiones son modelos atractivos a utilizar cuando la interpretabilidad es un factor de interés(Sen et al., 2020), en el ejemplo mostrado en la figura 3 se usa un árbol de decisión para decidir una actividad particular durante el día:

Figura 3: Árbol de decisiones de una actividad común.



Adaptado de Sen et al. (2020)

Basado en las características del set de entrenamiento, el árbol de decisiones aprende una serie de preguntas para inferir la etiqueta de la clase de las muestras.

Usando el algoritmo de decisión, se comienza con la raíz del árbol y se dividen los datos (Sen et al., 2020) en la característica que tenga una mayor ganancia de información (IG). En un proceso iterativo, se puede repetir este proceso de división en cada nodo hijo hasta que las hojas sean puras. Esto quiere decir que todas las muestras de cada nodo pertenecen a la misma clase. En práctica, esto puede resultar en un árbol bastante profundo , en donde fácilmente se puede llegar al overfitting. Acorde a esto, se busca limitar el árbol al indicar un límite para la profundidad máxima del árbol.

### 2.1.5. Maximizando la ganancia de información

En esta técnica, la función objetivo busca maximizar la ganancia de información por cada split, la cual se define a continuación:

$$IG(D_p, F) = I(D_p) - \sum_{j=1}^m \frac{N_j}{N_p} I(D_j)$$

Donde:  $f$  : Característica donde se hace el split

$D_p$  : Dataset del nodo padre.

$D_j$  : Dataset del j-esimo nodo hijo.

$I$  : Medida de impuridad.

$N_p$  : número total de muestras en el nodo padre.

$N_j$  : número de muestras del j-ésimo nodo hijo.

Como se puede ver, la ganancia de información es la diferencia entre la impuridad del nodo padre y la suma de las impuridades del nodo hijo Sen et al. (2020). Mientras menor sea la imputada de los nodos hijos, mayor será la ganancia de información. Sin embargo, para simplificarlo y reducir el espacio de búsqueda, muchas librerías (Scikit-learn por ejemplo) implementan árboles de decisión binarios. Esto quiere decir que cada nodo padre se divide en dos nodos hijos,  $D_{left}$  y  $D_{right}$ :

$$IG(D_p, F) = I(D_p) - \frac{N_j}{N_p} I(D_{left}) - \frac{N_{right}}{N_{left}} I(D_{right})$$

Las tres medidas de impuridad o criterios de división que son mas comúnmente utilizados son **Gini impurity**  $I_g$ , **entropía**  $I_H$  y el **error de clasificación**  $i_E$ , A continuación se muestra una definición de entropía para todos los casos no vacíos de  $p(i|t) \neq 0$ :

$$I_H(t) = - \sum_{i=1}^e p(i|t) \log_2 p(i|t)$$

Aquí  $p(i|t)$  es la proporción de las muestras que pertenecen a la clase  $C$  para cada nodo particular  $t$ . Por ende, la entropía es 0 si todas las muestras de un nodo pertenecen a la misma clase, y máxima si se tiene una distribución uniforme de la clase Sen et al. (2020).

Por ejemplo, para un escenario binario, la entropía sera 0 cuando  $p(i = 1|t)$  o  $p(i = 0|t)$ , si las clases son distribuidas uniformemente cuando  $p(i = 1|t) = 0,5$  y  $p(i = 0|t) = 0,5$ , la entropía será 1. Por ende, se puede decir que el criterio de entropía intenta maximizar la ganancia de información mutua en el árbol.

De manera intuitiva, el Gini Impurity puede ser extendido como un criterio para minimizar la probabilidad de fallar la clasificación:

$$I_G(t) = \sum_{i=1}^e p(i|t)(1 - p(i|t)) = 1 - \sum_{i=1}^e p(i|t)^2$$

Parecido a la entropía, el Gini Impurity es maxima si las clases estan perfectamente mezcladas, por ejemplo, en un caso binario con una clase tal que ( $c = 2$ ):

$$I_G(t) = 1 - \sum_{i=1}^e 0,5^2 = 0,5$$

Otra medida de impuridad es el Error de Clasificación:

$$I_G(t) = 1 - \max[p(i|t)]$$

Este es un criterio útil para recortar mas no para crear árboles de decisiones, ya que es menos sensible a cambios en las probabilidades del nodo de la clase.

## 2.2. Trastorno del espectro autista

El Trastorno de Espectro Autista es un diagnóstico que abarca un déficit significativo de comunicación social además de patrones de interés y comportamientos restrictivos (DeFilippis y Wagner, 2016). Los tratamientos actuales para los síntomas centrales del autismo están limitados a terapias psicosociales como lo son el análisis del comportamiento además de medicaciones particulares para los comportamientos repetitivos denotados en los diagnósticos.

### 2.2.1. Identificando el autismo en niños

Acorde a DeFilippis y Wagner (2016), para descartar otras categorías de diagnósticos como el síndrome de Asperger, desorden desintegrativo o algún otro desorden no especificado el individuo debe cumplir con varios criterios de comunicación, que incluyen:

- Problemas siendo recíproco en interacciones sociales o emocionales.
- Problemas severos para mantener relaciones sociales.
- problemas con la comunicación no verbal.

Además de esto, deben también cumplir con por lo menos dos de cuatro criterios de comportamiento repetitivo DeFilippis y Wagner (2016), los cuales incluyen:

- Habla repetitiva o estereotipada
- Movimientos motores o uso de objetos.
- Apego excesivo a rutinas
- Comportamiento ritualizado o resistencia excesiva al cambio.
- Intereses altamente restringidos, anormales en intensidad e interés
- Hiper o hipo reactividad a estimulación sensorial o interés inusual a sensaciones del ambiente.

Estos síntomas deben causar dificultades en el funcionamiento del infante para poder dar un diagnóstico DeFilippis y Wagner (2016). Algunos síntomas asociados al desorden del espectro autista incluyen irritabilidad, hiperactividad, comportamientos agresivos, ansiedad, insomnio y cambios de

### 2.3. Estados emocionales

Según (Frijda, 1986) las emociones surgen por eventos que son vividos por los humanos y a su vez percibidos de manera favorable o dañina para sus propios intereses, las emociones son vistas como resultados de los muchos sucesos que se llevan a cabo en el mundo en relación con las preocupaciones de uno mismo, resultando en el cambio del comportamiento para prepararse a reaccionar ante una situación.

Darwin propone tres principales expresiones, los que se asocian a los hábitos, los de antítesis y los de acción directa del sistema nervioso Frijda (1986). Estos principios sirven para explicar las propiedades formales de las expresiones y sus efectos bajo condiciones que los resaltan. De esto surgen cuatro principios similares a los de Darwin u otros autores pero diferentes en el énfasis teórico, que se muestran en la tabla 2:

Tabla 2: Principios del comportamiento expresivo

Principio de actividad relacional	Algunas expresiones de comportamiento pueden ser entendidas como actividad relacionada, en donde la actividad se establece, debilita y eventualmente niega las relaciones físicas y cognitivas del sujeto con el ambiente, esto mediante modificaciones en la exposición sensorial.
Principio de efectividad interactiva	Algunos Comportamientos expresivos pueden entenderse como acciones que apuntan a modificar la relación del sujeto con el ambiente, esto al influenciar el comportamiento de otros individuos.
Principio de activación	Algunos comportamientos expresivos pueden ser entendidos como manifestaciones de activación o disminución de reacciones debido a una activación, por activación se hace referencia direcciones intencionales, estar listo para la atención, reaccionar y responder.
Principio de inhibición	Algunos comportamientos expresivos se pueden entender como el resultado de inhibición del propio comportamiento bajo las anteriores definiciones.

Adaptado de Frijda (1986)

Juntos, estos principios deberían explicar la naturaleza del comportamiento expresivo, su forma y ocurrencia.

### 2.3.1. Emociones fundamentales

En (Plutchik, 1984) se menciona que existen dos principales perspectivas sobre el origen de las emociones, acorde a una estas últimas son el productor de la selección natural, es decir son adaptaciones evolutivas que son entendidas usando explicaciones de psicología evolutiva. Acorde al segundo punto de vista, las emociones son construidas socialmente y varían a lo largo y ancho del espectro cultural, existe evidencia que valida ambas escuelas de pensamiento, incluso se argumenta que ambas explicaciones son correctas. La solución usual a este dilema es decir que algunas emociones son parte de la evolución mientras que otras son construidas en base al contexto cultural. Plutchik (1984) agrega que así como un estornudo, las emociones tienen mucho que ver con el estado del cuerpo, estas son asociadas a patrones de cambios físicos y también en las estructuras del cerebro que se encargan de las emociones.

Es razonable concluir que una emoción es un constructo hipotético o una inferencia basada

en varios tipos de evidencia. Esta evidencia puede incluir reportes verbales de sentimientos interiorizados, así como comportamientos expresivos y reacciones grupales entre otros. Una importante implicación sigue el concepto de las emociones como constructos hipotéticos.

En la tabla 3 se describen emociones en términos de diferentes tipos de lenguajes según Plutchik (1984), por ejemplo, uno podría identificarlas en términos subjetivos comunes que incluyen palabras como feliz, triste, enojado o disgustado.

Tabla 3: Lenguajes para describir estados emocionales.

Lenguaje subjetivo	Lenguaje funcional	Lenguaje funcional
Miedo, Terror	Retirando, Escapando	Protección
Enojo, Ira	Agrediendo, Mordiendo	Destrucción
Gozo, Éxtasis	Copulando, Poseyendo	Reproducción
Tristeza, Luto	Llorando	Reintegración
Aceptación, Confianza	Unificando, Apreciando	Incorporación
Disgusto, Aversión	Vomitando, Defecando	Rechazo
Espectativa, Anticipación	Examinando, Planeando	Exploración
Sorpresa, Asombro	Detendiéndose, Congelándose	Orientación

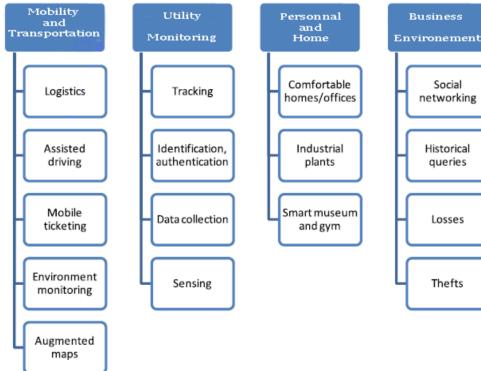
Adaptado de Plutchik (1984)

## 2.4. Internet of Things (IOT)

IoT hace referencia a las redes interconectadas de las que forman parte los objetos de uso cotidiano, mismos que actualmente se encuentran equipados con algún nivel de inteligencia (Xia et al., 2012). El IoT busca facilitar el uso de las tecnologías relacionadas al internet al integrar la mayor cantidad posible de objetos con la capacidad de interactuar entre ellos a través de sistemas embebidos, dichas comunicaciones dirigidas tanto a usuarios humanos como otros dispositivos inteligentes.

En la figura 4 se muestra un diagrama con algunos ejemplos de áreas donde se utilizan tecnologías IoT:

Figura 4: Areas del IoT



Adaptado de Abdmeziem y Tandjaoui (2014)

Una parte importante del IoT es la habilidad de los sistemas embebidos para obtener datos del ambiente en el que se encuentran y de manera autónoma organizarse para responder a cambios presentados en sensores (Abdmeziem y Tandjaoui, 2014).

Uno de los principales inconvenientes ronda alrededor del medio por el cual proveer de energía a dichos dispositivos, para mantener un nivel de funcionamiento óptimo frecuentemente es necesario el uso de baterías, además de que los avances en el campo de TI permiten un uso cada vez más eficiente de energía por parte de los componentes utilizados.

Además de esto, el IoT es aplicado en gran medida en espacios personales y sociales, negocios, industria, servicios de monitoreo y movilidad o transporte, esto dependiendo del tipo de conexión utilizada, distancia de uso, escala, repetición, el grado en el que el usuario está involucrado en los procesos e impacto del proyecto. Abdmeziem y Tandjaoui (2014)

#### 2.4.1. TinyML en recolección de datos

Uno de los extensos usos que se le da a los sistemas embebidos es el registro de datos por periodos de tiempo (Schizas et al., 2022), los pertinentes para el proyecto se les denomina datos biométricos debido a que son valores directamente producidos por el cuerpo humano o en este caso por un movimiento que es producido de manera analógica por el control del usuario, en este caso la emoción se determina por medio del movimiento al que el cubo de sentimientos se someta, para llegar a esto es necesaria la construcción tanto de un dataset como de un efectivo algoritmo de Machine Learning. Segun Dantcheva et al. (2015), La idea de este funcionamiento ronda alrededor de utilizar aprendizaje supervisado para el análisis de conjuntos de datos y así asignarles los valores que corresponden a una de las caras en el cubo de sentimientos.

### 2.5. Manofactura aditiva

De acuerdo con (Chang, 2014) el proceso de desarrollo de productos convencional es un proceso cíclico de “diseñar-construir-probar”, el desempeño y confiabilidad de un producto

depende en una gran medida de los resultados de pruebas físicas, las cuales implican la fabricación de prototipos funcionales del producto acompañado de largas y costosas pruebas físicas. Fabricar prototipos usualmente implica la planeación de procesos de manufactura y una gran cantidad de ajustes pequeños para una producción reducida, por no mencionar los altos costes o posibles correcciones al diseño resultado de pruebas de estrés.

### 2.5.1. Prototipado rápido en electrónica

La manufactura aditiva se utiliza para la producción de modelos y prototipos a nivel industrial, en tiempos modernos existen técnicas como la impresión 3D (Gardan, 2017) en donde se han desarrollado técnicas como el prototipado rápido, sin necesidad de plantas de gran tamaño, debido a esto es que se vuelve más fácil el acceso a estas tecnologías permitiendo iterar y evaluar con mayor velocidad propuestas de diseño. (Kruth et al., 1998)

Esto compete al proyecto dado que se propone un diseño para la construcción de la caja de emociones que toma en cuenta tanto el espacio que necesitan los componentes electrónicos como la disposición de espacios para tornillos, medida y calce de la tapa e incluso el aspecto estético del prototipo sin dejar de lado la resistencia del material

### 2.5.2. Diseño Asistido por Computadora

El paradigma de e-Design se entiende como la realización de que las necesidades actuales en la ingeniería pueden ser asistidas por medio de prototipado virtual y físico a través de métodos sistemáticos y cuantitativos en la toma de decisiones relacionada al diseño (Chang, 2014).

Un ambiente típico de software de e-Design puede ser construido usando una combinación de herramientas CAD, CAE y CAM como la base, e integraciones de software de las disciplinas específicas disponibles del mercado.

En las investigaciones de Chang (2014) los modelos de CAD permiten a los ingenieros explorar de manera conveniente alternativas en el diseño de productos. El modelado CAD define las dimensiones que maneja la geometría de las partes a través de características geométricas y estableciendo relaciones entre las dimensiones a lo largo y ancho de todas las piezas. A través de las dimensiones y relaciones, es posible hacer cambios al hacer el cambio de algunas medidas dimensionales particulares. Los cambios son propagados automáticamente a través de todo el producto mecánico de tal manera que siga dichas dimensiones y relaciones establecidas.

### 2.5.3. Fritzing

Fritzing es una iniciativa de código abierto que provee accesibilidad a componentes electrónicos como material creativo para todos, es un software y servicios para las plataformas Processing y Arduino (Knörig et al., 2009), siendo hogar de extensos ecosistemas de componentes que ayudan a sus usuarios a documentar sus prototipos, compartirlos en comunidades y

apartar al aprendizaje de electrónica, así como esquemáticos, organización, manufactura y creación de PCBs profesionales.

Fritzing es un software que tiene soporte de una gran variedad de plataformas arduino como lo es Uno, GAlileo, Yun. BT, MEGA 2560, Due; Nano entre otros, además, Knörig et al. (2009) indica que no solo esto sino que también contiene una larga selección de componentes robóticos como lo son motores, medidores de rango, bocinas, silbatos, motores, pantallas LCD y indicadores digitales.

#### **2.5.4. Placa WEMOS s2 mini**

La placa principal donde se lleva a cabo la predicción de las caras, además de recibir los datos de los sensores de inclinación y sensor MPU 6050, algunas de las características técnicas de este componente son: 3.3V de consumo energético, 27 pines digitales I/O, 140MHz de velocidad de reloj, 4MB de memoria flash, 2MB de memoria PSRAM (wemos.cc, 2021).

#### **2.5.5. Switch de inclinación**

Según las investigaciones de (Villanueva, 2023) los switches de inclinación (tilt switch en inglés) usualmente detectan solo posiciones horizontales dentro de cierto rango angular, como lo puede ser la vecindad de la posición horizontal, usualmente de algunos ángulos particulares, usualmente no operan con la totalidad de los ángulos y no generan contacto eléctrico entre los electrodos, mientras estén orientados más allá del rango angular asignados como válidos para la posición horizontal y viceversa, es decir, no tienen contacto eléctrico en posiciones horizontales.

#### **2.5.6. Sensor de aceleración e inclinación MPU6050**

Los dispositivos MEMS están hechos de sustrato de silicón de la misma manera que los circuitos integrados de un solo chip, lo que significa que sus tamaños pueden variar entre algunos micrones hasta milímetros (Fedorov et al., 2015). Existen diversos tipos de dispositivos MEMS de giroscopio que difieren en sus estructuras internas, pero todos comparten una característica la cual es que se basan en la fuerza Coriolis. En estos hay un cuerpo activo haciendo movimientos recíprocos, es decir que al rotar el sustrato donde este cuerpo está localizado la fuerza Coriolis que está dirigida perpendicularmente al eje de rotación comenzará a actuar en él.

Las oscilaciones de la masa del dispositivo ocurren a lo largo del eje X y son generadas por fuerzas electrostáticas, además, la oscilación de la estructura interna es posible solamente en el eje Y Fedorov et al. (2015). Capas de capacitores planos se localizan entre la estructura interna y la estructura falsa, haciendo que se haga la medida de la capacitancia, a través de esto es posible encontrar el rango de movimiento relativo a la sustancia

#### **2.5.7. Arduino IDE**

IDE es un acrónimo para “Integrated Development Environment” (Entorno de Desarrollo Integrado), este es un software oficial introducido por Arduino.cc (Fezari y Al Dahoud, 2018)

que es mayormente utilizado para editar, compilar y subir código a dispositivos de la marca Arduino y similares. Casi todos los módulos Arduino son compatibles con este software de código abierto y esta siempre listo para instalar y comenzar a compilar código en el momento.

Software oficial de Arduino que es capaz de hacer la compilación sencilla incluso para el usuario común sin tanto conocimiento técnico previo con el propósito de que sea sencillo de aprender y rápido de comenzar a usar. Cuenta con un rango de módulos disponibles como lo son la familia UNO, arduino MEGA, Arduino Leonardo, Arduino Micro además de múltiples opciones de terceros. Al hacer la escritura de código la cual se hace por medio de archivos llamados sketches creados en el IDE, se genera un archivo hexadecimal que después es transferido y cargado al controlador de la placa Fezari y Al Dahoud (2018).

### 2.5.8. Jupyter LAB

Acorde a (Broker y Schroeder, 2023) Este es un entorno computacional interactivo basado en web para crear documentos denominados como notebooks, Jupyter Notebook está construido usando variedad de librerías de código abierto como lo son IPython, ZeroMQ, Tornado, jQuery, Bootstrap y MathJax. Una aplicación hecha en un Jupyter Notebook es un REPL basado en navegador, mismo que contiene una lista ordenada de celdas de entrada y salida que puede contener texto, código, computación matemática, gráficas así como otros elementos multimedia.

### 2.5.9. Pandas

Pandas es una librería de Python especialmente diseñada para análisis de datos, Pandas está construido sobre dos librerías fundamentales de Python (McKinney et al., 2011), las cuales son matplotlib para visualización de datos y Numpy para operaciones matemáticas, esta librería actúa como Wrapper de las librerías antes mencionadas, permitiendo que se puedan acceder a variedad de métodos de Matplotlib y Numpy con menos código.

### 2.5.10. NumPY

Numpy es una librería parte de un paquete de computación científica para python, es una librería que provee objetos de arreglos multidimensionales, además de varios objetos derivados como lo son los arreglos enmascarados o matrices (Bressert, 2012), cuenta también con algunas rutinas para operaciones rápidas dentro de arreglos, incluyendo manipulación lógica y matemática, agrupación, selección, transformaciones de fourier discretas, álgebra lineal básica, operaciones estadística básicas y simulación aleatoria por poner algunos ejemplos.

### 2.5.11. MatplotLIB

Acorde a (Ari y Ustazhanov, 2014) un script de matplotlib está estructurado de tal manera que con pocas líneas de código se puedan crear la mayor cantidad posible de instancias para visualizar gráficas de datos. La construcción de scripts de matplotlib trabaja con dos APIs, las cuales son: API de pyplot. Una jerarquía de objetos para python escrita como

matplotlib.pyplot. API orientada a objetos, una colección de objetos que puede ser incluso más flexible que pyplot, esta provee acceso directo al backend de matplotlib.

### 2.5.12. Scikit Learn

Scikit learn provee algoritmos para trabajos de machine learning incluyendo clasificación, regresión, reducción de dimensionalidad y agrupamiento lo que lo provee módulos para el pre procesamiento de datos, extracción de características, optimización de hiper parametros así como evaluación de modelos (Hackeling, 2017). Scikit learn está construido con las bases de populares librerías de python como lo son numpy, scipy, estos últimos extienden el soporte para operaciones eficientes en arreglos largos y matrices multidimensionales. Scipy provee módulos para computación científica. La librería de visualización de matplotlib es usada a menudo con la librería scikit learn.

Operando Bajo la licencia permisiva BSD, scikit learn puede ser usada en aplicaciones comerciales sin restricciones, muchos algoritmos de esta librería son rápidos y escalables para toda situación salvo por datasets demasiado grandes. Según Hackeling (2017) scikit learn es notable por su fiabilidad dado que mucho de esta librería es compatible con evaluaciones automáticas.

### 2.5.13. Librería ESP32 Arduino

Por defecto Arduino IDE está diseñado para trabajar con una limitada cantidad de componentes de la familia arduino como lo son la placa UNO por ejemplo, a pesar de esto el compilador cuenta con la opción de descargar librerías que expanden la compatibilidad hacia otros tipos de placas, como lo son la utilizada durante el proyecto de caja de emociones (Babiuch et al., 2019).

### 3. Trabajos Relacionados

En este capítulo se muestran documentos y fuentes de información con características similares a las mostradas en este proyecto.

#### 3.1. Reconocimiento Automático de emociones en Interacciones Robot-Niño para el tratamiento de ASD

Acorde a las investigaciones de (Leo et al., 2015) se propone un sistema orientado manejo automático de un protocolo médico dirigido a mejorar las capacidades de niños afectados por ASD para asociar emociones particulares a expresiones faciales específicas. El protocolo implementa la idea en la que el robot muestra una expresión facial, posteriormente pide al niño que imite la expresión para evaluar su capacidad de imitar emociones, posteriormente midiendo el tiempo que pasa entre que el robot pide la expresión y la reacción del niño.

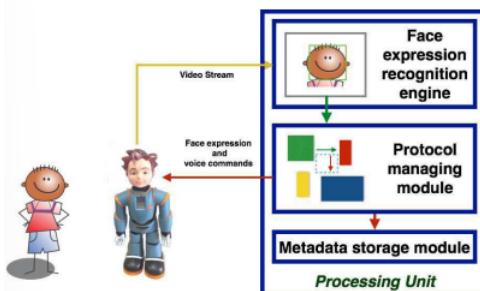
Leo et al. (2015) continua su propuesta explicando que la implementación de este protocolo le asigna al operador humano dos roles: por un lado el manejo del robot y por el otro la evaluación del nivel de interacción del niño. La introducción de técnicas de Machine Learning permiten liberar al operador del sistema haciendo que la información de la terapia progrese a través de objetivos durante las sesiones.

La unidad de procesamiento implementa una secuencia de acciones derivadas del protocolo utilizado al usar tres módulos diferentes:

- Módulo de manejador de protocolos. Dedicado a la implementación de las acciones del controlador acorde a la especificación del protocolo.
- Módulo de almacenamiento de rostros. realiza el reconocimiento de rostros del niño.
- Módulo de almacenamiento de metadatos. Se enfoca en almacenar, presentar y análisis de metadata recolectada por medios estadísticos y gráficos para evaluar el comportamiento del niño a lo largo del tiempo.

En la figura 5 se muestra una abstracción que maneja el sistema:

Figura 5: Sistema de interacción robot-niño.



Adaptado de Leo et al. (2015)

La investigación de (Leo et al., 2015) indica que aquí el protocolo médico es automáticamente administrado por la unidad de procesamiento, el protocolo de control modular le da al robot las instrucciones con el objetivo de estimular la reacción del niño mientras la el módulo de identificación de expresiones lo envía como señal de vídeo para hacer el análisis de la expresión del niño. Finalmente la expresión del niño y el tiempo de respuesta son almacenados en el módulo de almacenamiento de metadatos.

### 3.2. Reconocimiento facial en tiempo real de niños con autismo basado en Deep Learning y IoT

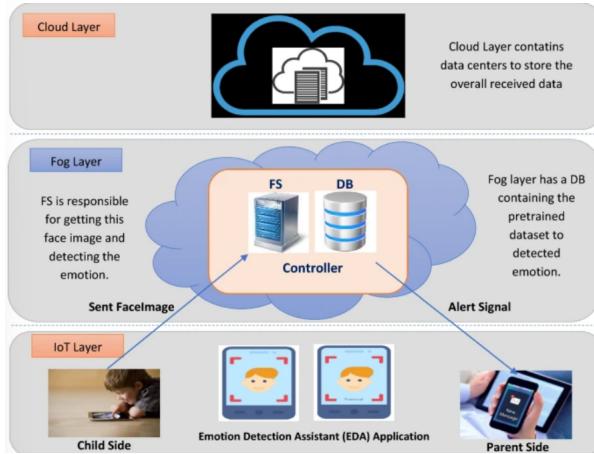
(Talaat, 2023) hace la propuesta de un framework de detección de emociones para niños autistas. Reconocimiento facial, extracción de características de rostros y categorización de características son las tres etapas del reconocimiento de emociones. Se manejan un total de 6 emociones que pueden ser detectadas por el sistema, siendo estas: enojo, miedo, alegría natural, tristeza y sorpresa.

El sistema consta en 3 capas: Capa de nube, Capa de niebla y Capa de IoT.

Talaat (2023) continúa su explicación con que en las primeras dos se propone una aplicación móvil que se usa para capturar una imagen del niño mientras esta usando un dispositivo inteligente, posteriormente envía la imagen a la capa de niebla, en esta capa existe un controlador denominado como server de niebla (Fog Server) que es responsable de obtener esta imagen y detectar la emoción dependiendo la técnica propuesta. Aquí se alberga también la base de datos con el dataset pre-entrenado. Luego de detectar la emoción, fog server envía una alerta de mensaje al dispositivo del parent cuando las emociones detectadas son enojo, miedo, tristeza o sorpresa.

En la figura 6 se muestra una abstracción del sistema, el manejo y control principales está implementado en el controlador de la capa de niebla para reducir la latencia de la detección en tiempo real para una respuesta rápida.

Figura 6: Abstracción de las capas del framework



Adaptado de Talaat (2023)

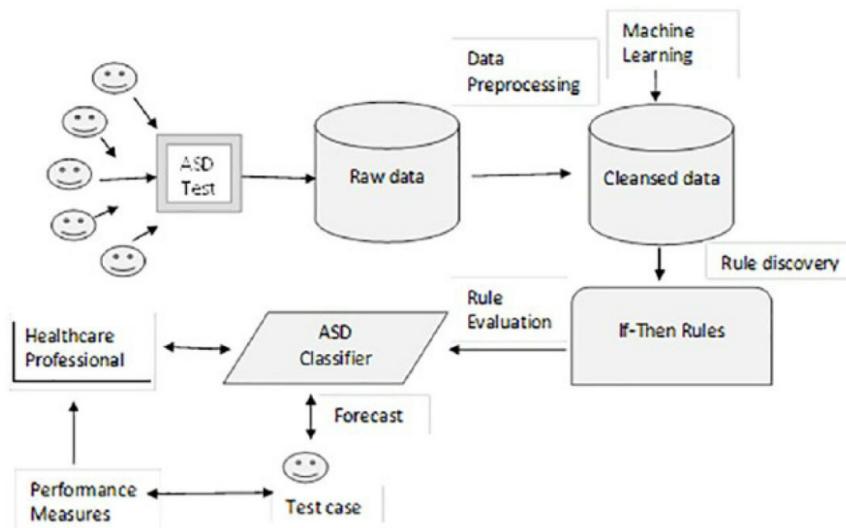
### 3.3. Un nuevo modelo de ML basado en inducción de reglas para detección del autismo

Uno de los modelos de clasificación de ML es el cubrimiento (Covering). La investigación de (Thabtah y Peebles, 2020) indaga en las técnicas de cubrimiento donde se descubren piezas simples de información proveniente de datasets históricos estructurados en el formato de Si-Entonces, lo que hace que sean fáciles de implementar. Thabtah y Peebles (2020) continúa proponiendo un nuevo método de detección del autismo basado en la arquitectura de cubrimiento.

El método está basado en la clasificación donde se utiliza un método de búsqueda por regla de descubrimiento, el RML evalúa la regla descubierta y desecha las redundancias, por ende, solo las reglas que han sido clasificadas como instancias de entrenamiento se mantienen. La fase de evaluación no solo reduce el número de reglas descubiertas sino que también reduce el espacio de búsqueda de los datos individuales, lo cual mejora la eficiencia del proceso de entrenamiento.

En la figura 7 se muestra una abstracción general de la arquitectura de ML que se utiliza en el proyecto.

Figura 7: Abstracción de clasificador por método de cubrimiento.



Adaptado de Thabtah y Peebles (2020)

Los datos son recolectados por una aplicación móvil llamada ASDTests que implementa varios métodos de muestreo. Las investigaciones se centran en tener como usuarios a niños, adolescentes y adultos con 3 datasets recolectados entre septiembre del 2017 y enero del 2018, una vez que los datos crudos fueron obtenidos, varias operaciones de pre procesamiento fueron aplicadas, incluyendo reposición de valores faltantes, discretización de ciertos atributos continuos como lo son la edad de los individuos.

La extracción de características se realizó para remover las características redundantes, dos siendo eliminadas incluyendo puntuación final obtenida por el método de muestreo y el tipo de método de puntuación. El resultado de la fase de reglas de evaluación es el sistema de clasificación (clasificador) que será usado para predecir el valor de la clase para clases desconocidas, es decir, individuos que no han sido clasificados Thabtah y Peebles (2020). Cuando el clasificador es puesto a prueba, varias métricas de evaluación se muestran para revelar la efectividad de las reglas para predecir casos.

### 3.4. Prototipo de cubo de sentimientos en comparación con otros proyectos relacionados

La gran mayoría de proyectos relacionados que mezclan tecnología de aprendizaje máquina, visión por computadora con psicología y registro de emociones de pacientes autistas utilizan métodos que implican dos características principales:

- Reconocimiento de rostros, donde el principal dato de ingreso o posible generador de datos tomados en cuenta es el rostro.
- Pruebas con supervisión de expertos, donde para que el sistema y método sean funcio-

nales es necesaria la presencia de un experto durante sesiones de evaluación.

A pesar de los métodos y técnicas utilizados la constante supervisión de expertos en psicología derrotan de cierta manera el propósito de la toma automática de datos, es decir, el sistema se vuelve menos relevante con la operación directa del usuario. El prototipo de caja de emociones apunta a mostrar dichas deducciones sin la influencia directa de personal especializado, enviando los resultados relevantes a la mano del tutor legal del niño autista.

## 4. Propuesta de prototipo

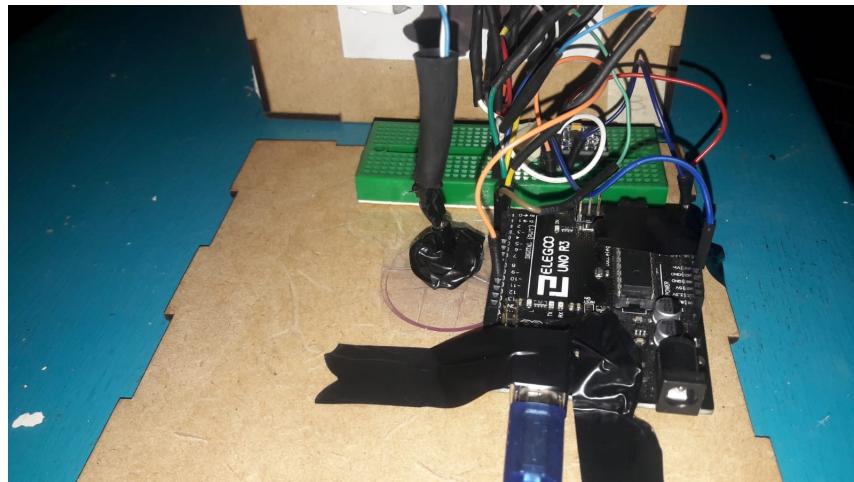
En este capítulo se muestra la concepción de la abstracción bajo la cual se forma la propuesta así como descripción específica de las partes que lo componen.

### 4.1. Áreas de oportunidad del prototipo anterior

#### 4.1.1. Breve descripción del prototipo previo

En la figura 8 se puede apreciar la construcción física de la caja de emociones resultado de la etapa de desarrollo anterior inmediata.

Figura 8: Caja de emociones prototipo previo



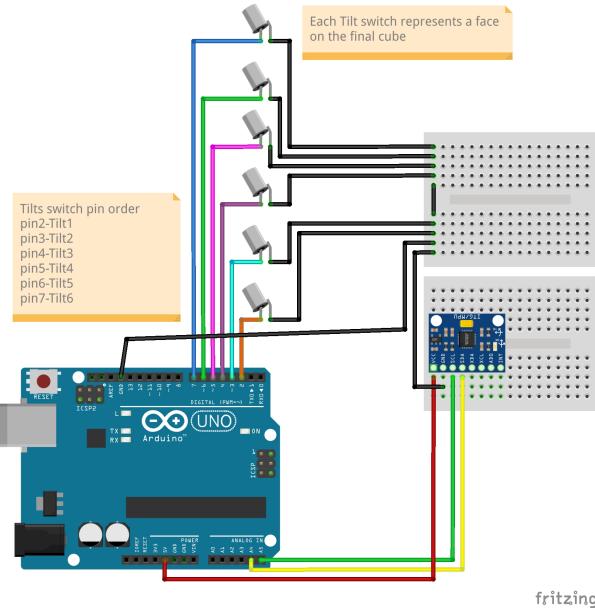
Elaboración propia

Los componentes utilizados son: Placa Arduino UNO, Switches de inclinación (uno para cada cara), Sensor de inclinación MPU 6050 además de otros materiales como un mini protoboard de 170 puntos para la organización de los componentes y varias líneas de cable trenzado para la comunicación de componentes. Cuando todos los componentes se encuentran conectados y listos la caja obtiene las señales numéricas producidas tanto por el giroscopio como de los switches de inclinación, mediante un algoritmo que toma en cuenta los datos de dichos componentes se evalúa de manera directa e infiere la cara de la caja en base a comparaciones de los datos, mismos que envía por medio de cable USB a la computadora.

En esta última un script de python obtiene la inferencia de la cara y la muestra en terminal a una velocidad de 0.5 segundos por muestra.

En la figura 9 se aprecia un esquema gráfico de las conexiones del sistema:

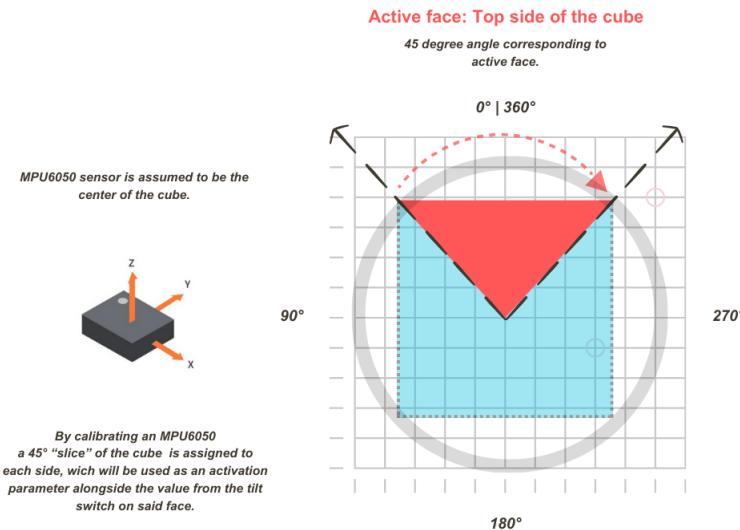
Figura 9: Gráfico de los componentes del prototipo previo



Elaboración propia.

En la figura 10 se describe la abstracción bajo la cual se genera la lógica para deducir la cara activa:

Figura 10: Abstracción del prototipo previo



Elaboración propia.

#### 4.1.2. Necesidades y mejoras del prototipo previo

De entre las áreas de oportunidad presentes en el trabajo anteriormente mencionado se pueden destacar:

- Ajustar o cambiar componentes para disminuir el tamaño físico del proyecto.
- Organizar el espacio al interior de la caja para que cada pieza tenga su lugar indicado y esté bien asegurada a la estructura.
- Cambiar el algoritmo a base de decisiones para implementar un clasificador de mayor precisión y escalabilidad.
- Reforzar la construcción de la caja para evitar ruido en la toma de datos.
- Evitar que el sistema dependa de una conexión USB para tener energía.
- Hacer que los datos sean enviados mediante conexión WIFI y posteriormente mostrados en un dispositivo móvil.
- Crear una sola función que prediga la cara activa a base de enviarle una cadena con los datos de los sensores.

### 4.2. Hardware utilizado

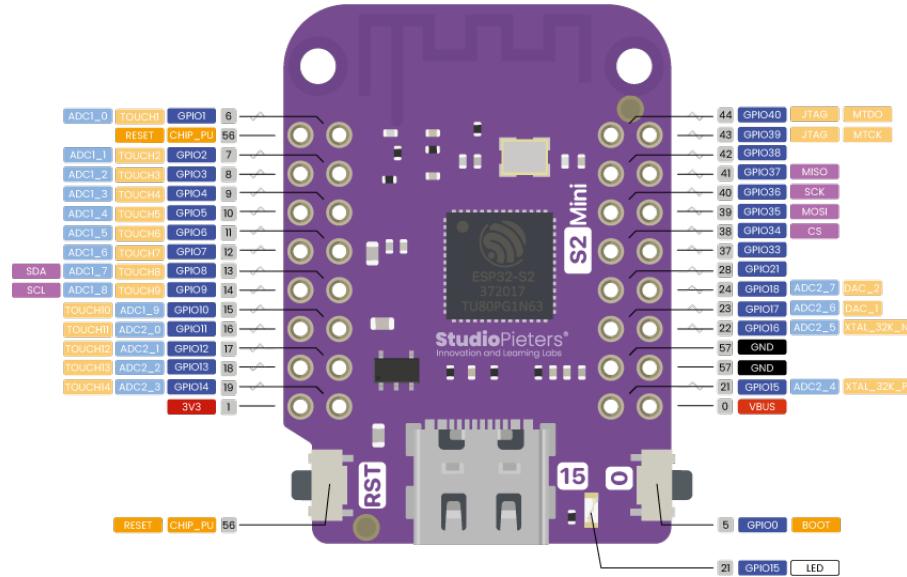
#### 4.2.1. Placa WEMOS S2 mini

Otra característica importante para la realización del proyecto es su capacidad de conexión wifi integrada a una velocidad de 2.4 GHz, mismo que integra una dirección MAC, radio wi-fi, switch RF balun, amplificador de energía, amplificador de sonido bajo. Wemos s2 mini cumple con los estándares y protocolos IEEE 802.11b/g mismo que ofrece las soluciones de Wi-fi. El núcleo de esta placa es hogar de un chip Xtensa de 32 bits LX7 CPU que opera a 240MHz, lo cual facilita llevar a cabo desarrollo de aplicaciones sin necesidad de un host MCU.

Esta placa es altamente recomendada para escenarios de desarrollo IoT en donde se necesiten de componentes pequeños o sistemas embebidos usados en el cuerpo, esto en conjunto con el módulo WiFi integrado y la cantidad de pines disponibles para uso vuelven esta placa de alto valor para el proyecto Cubo de Emociones.

En la figura 11 se aprecia una representación gráfica de la placa utilizada en el proyecto.

Figura 11: Distribución de pines de placa ESP WEMOS S2 MINI



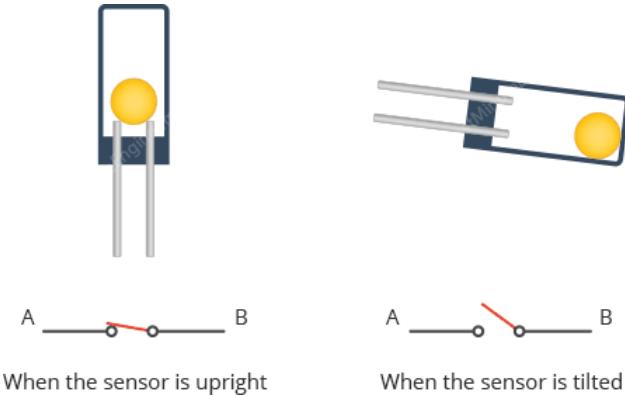
Adaptado de wemos.cc (2021)

#### 4.2.2. Swtich de inclinación SW-5200

Uno de los componentes principales para la detección de datos de la caja es el sensor de inclinación, también conocido como sensor de inclinación o switch de vibración, este es un dispositivo que proporciona una señal digital cuando presenta una inclinación mayor a un determinado umbral (Components101, 2023), esta pieza no permite saber el grado de inclinación, solamente deja de permitir el paso de la corriente en cierto ángulo.

En la figura 12 se muestra un gráfico de su operación, esta consiste un cilindro cuya pared constituye un contacto eléctrico, mientras que el otro contacto se encuentra en el centro de la base, al inclinarse lo suficiente una esfera metálica crea un puente entre ambos contactos cerrando así el circuito y por consiguiente emitiendo una señal.

Figura 12: Switch de inclinación



Adaptado de Villanueva (2023)

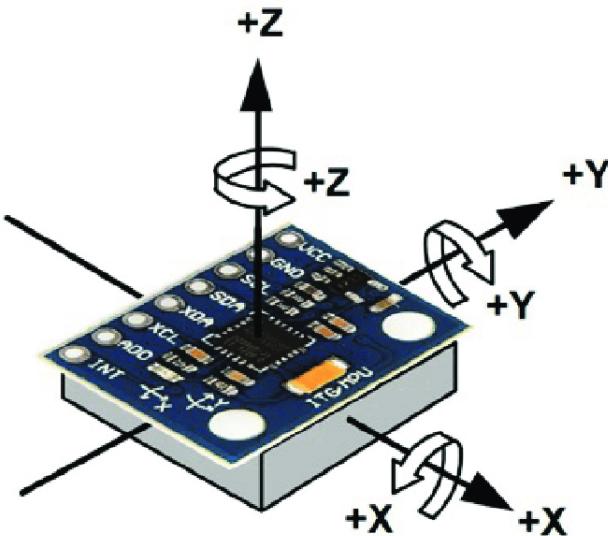
#### 4.2.3. Sensor MPU6050

El módulo MPU6050 es un sistema micro electro-mecánico (MEMS) que consiste en un acelerómetro y un giroscopio de 3 ejes cada uno, con esto es capaz de medir aceleración, velocidad y orientación, así como posicionamiento y otros parámetros relacionados con el movimiento de un sistema u objeto, algunas características importantes de este dispositivo son: acelerómetro y giroscopio de 3 ejes, consumo de 3-5V, comunicación I2C, ADC de 16 bit para alta precisión, DMP integrado para un considerable poder de computación, dirección IIC configurable. Este módulo permite leer datos a través del bus IIC, cualquier cambio en el movimiento será reflejado en el sistema mecánico, lo que sea visible con el cambio de salida de voltaje, mismos datos que se almacenan en el buffer FIFO. Esto quiere decir que los datos están constantemente listos para ser leídos, por lo que se usa un MCU para leer los datos del buffer FIFO por medio de comunicación I2C.

El principal inconveniente con este módulo se presenta al momento de dar sentido a los datos en bruto que obtiene el sensor con las funciones por defecto, en el caso de la caja de emociones se implementa una función que transforma dichos valores para que correspondan a coordenadas de una circunferencia, es decir, los correspondientes al eje de rotación X y Y tanto del acelerómetro como del giroscopio (mencionados como Roll y Pitch respectivamente en la documentación del componente).

En la figura 13 se muestra una representación visual del sensor y los ejes en los que opera:

Figura 13: Sensor acelerómetro y giroscopio MPU6050



Adaptado de Fedorov et al. (2015)

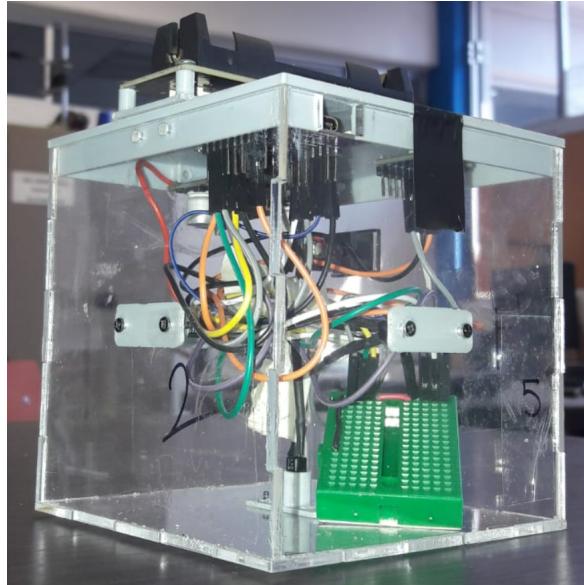
#### 4.2.4. Caja contenedora

La caja contenedora de prototipo se constituye de dos materiales:

- Paneles de acrílico de 10x10 cm de () de grosor que fungen como las caras del cubo, mismos que han sido cortados mediante maquinaria láser, en los extremos del panel se realizó un patrón dentado rectangular cuyo objetivo es hacer embonar las piezas en forma de cubo, al centro de cada uno de estos se encuentran dos agujeros para tornillos.
- Piezas de filamento PLA, las cuales son:
  - Una tapa de dimensiones similares a las de acrílico que alberga espacios para tornillos donde se aseguran la placa WEMOS S2 mini, el sensor MPU6050 y un mini protoboard de 170 puntos ARD 335 para conexiones a tierra.
  - Sockets diseñados para albergar switches de inclinación, mismos que son asegurados a cada tapa de acrílico por medio de tornillos.

En la figura 14 se muestra las características de la caja contenedora y el espacio asignado para cada componente. Muchas de las piezas presentadas son aseguradas con tornillos con la intención de que sean fáciles de mover cuando necesiten ser reemplazadas o manipuladas para hacer espacio al interior de la caja.

Figura 14: Construcción física de la caja de emociones



Elaboración propia.

### 4.3. Tecnologías y librerías utilizadas

#### 4.3.1. FreeCad

Programa de modelado paramétrico 3D de código abierto hecho principalmente para hacer el diseño de objetos del mundo real en cualquier tamaño, el modelado paramétrico permite modificar fácilmente el diseño y controlar las versiones por medio de un historial de cambios. Las principales características de FreeCAD incluyen kernels de geometría, renderizado, boceador, arquitectura modular, simulación de robots, formatos de salida de archivos estándar.

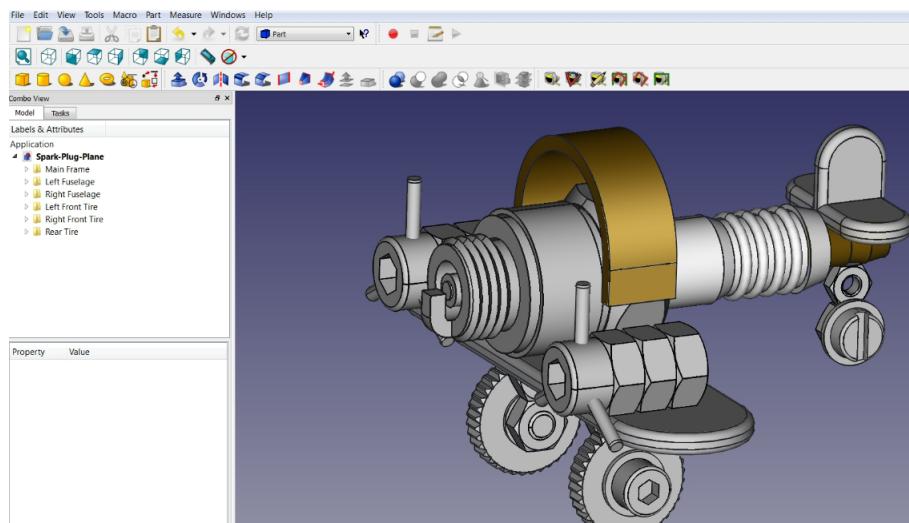
FreeCAD no está diseñado para trabajo especializados o para hacer cierto tipo de piezas, por esto mismo es capaz de ser usado para una amplia gama de usos, pues permite al usuario producir modelos de todos los tamaños desde componentes electrónicos hasta piezas completas para ser impresas. Algunas características de este software son:

- Modelo paramétrico completo, es decir su forma puede ser basada en propiedades establecidas o depender de otros objetos al calcular los cambios según se necesiten.
- Arquitectura modular que permite extensiones de plugins para añadir funcionalidades a la aplicación, existe control de un intérprete de python nativo, macro y scripts, representación de geometría 3D y 2D así como transformaciones.
- Importación y exportación de archivos en formato estándar como lo son STEP, IGES, OBJ, STL, DXL, SVG entre otros además de su propio formato (.FCStd) lo que lo provee de una gran compatibilidad.

- Módulo de dibujo técnico con opciones para vistas detalladas, visualización de secciones, dimensionado, entre otros, permitiendo generar vistas 2D de modelos 3D.

Entre otras herramientas útiles. En la figura 15 se muestra un ejemplo de su interfaz de usuario.

Figura 15: Interfaz de usuario y modelo de ejemplo en FreeCAD



Adaptado de Riegel et al. (2016)

#### 4.3.2. Arduino IDE

el Entorno Integrado de Desarrollo de Arduino (Arduino IDE) contiene un editor de texto para escribir código, área de mensajes, consola de texto, barra de herramientas con botones para funciones comunes entre otras opciones, conecta con hardware tanto de la propia marca Arduino como de otras plataformas gracias a su facilidad de carga de librerías.

Las librerías proporcionan funcionalidades extra para los sketches producidos por el IDE, por ejemplo trabajar con dispositivos manejando datos, cabe decir que el uso de librerías aumenta el espacio consumido por los dispositivos a los que se haga la carga de código.

El IDE de arduino permite también agregar reconocimiento para hardware de terceros por medio del diccionario de sketch. Las plataformas instaladas incluyen definiciones de placas, librerías centrales, bootloaders y programadores definidos, además de contar con herramientas útiles como monitor y graficador serial para revisar la actividad de las señales eléctricas y los valores de las variables.

A pesar de que Arduino IDE esta hecho para ser utilizado con productos de la marca Arduino, es posible la descarga de librerías que permiten el uso de otras placas como lo es el caso de la WEMOS S2 MINI (o cualquiera de la familia ESP32), la cual es utilizada en el proyecto de la caja de emociones.

Dada su facilidad de uso es posible cargar código a placas de manera rápida y efectiva (siempre y cuando se importan las librerías correctas para el uso de dicha placa) lo que permite hacer modificaciones con facilidad, En la figura 16 se muestra un fragmento de la interfaz gráfica del IDE de Arduino.

Figura 16: Logotipo de la interfaz de Arduino



Adaptado de Villanueva (2023)

#### 4.3.3. Jupyter Notebook

Un Jupyter Notebook es un archivo de formato JSON, siguiendo esquema de versiones, usualmente en extensión .ipynb. Las partes principales del Notebook son la metadata, el formato y la lista de celdas. la metadata son los datos de diccionarios de definiciones para preparar y mostrar el notebook, el formato de notebook es el número de versión del software, mientras que la lista de celdas se dividen en: Celdas de Markdown para muestra de texto, celdas de Código para ejecutar lenguaje python y celdas de salida para las celdas de código.

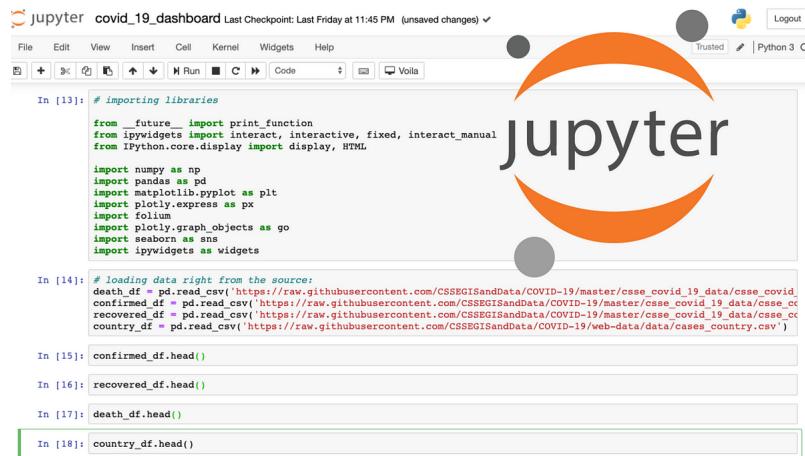
El principal uso de esta herramienta radica no solo en la presentación visual del código y del texto para generar reportes sino también dada la manera en la que está distribuida permite fácilmente controlar archivos y mezclar celdas de código con celdas de texto plano. Aunado a esto el propio entorno de desarrollo del notebook permite la ejecución de código y muestra de resultados sin necesidad de generar archivos independientes por lo que el proceso de desarrollo y debugging se facilita de sobremanera.

En el caso del prototipo Júpiter Notebook ofrece un espacio donde desarrollar algoritmos que permiten desde la visualización de datasets como el tratado y ajuste de los datos hasta la creación del propio modelo de árbol de decisiones y graficación, es decir, es el entorno de

desarrollo idea no solo para ejecucion de código sino para importación de librerías que se necesiten durante el desarrollo.

En la figura 17 se muestra un extracto de código relacionado con métodos estadísticos del Covid 19 como ejemplo de las celdas manejadas en Jupyter Notebook.

Figura 17: Ejemplo de la interfaz presentada por Jupyter Notebook



The screenshot shows a Jupyter Notebook interface with several code cells. Cell In [13] imports various libraries including numpy, pandas, matplotlib.pyplot, plotly.express, folium, plotly.graph\_objects, seaborn, and ipywidgets. Cells In [14] through In [18] load data from CSV files: death\_df, confirmed\_df, recovered\_df, country\_df, confirmed\_df.head(), recovered\_df.head(), death\_df.head(), and country\_df.head(). The notebook has tabs for File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a Voila button. A large orange smiley face watermark is overlaid on the right side of the interface.

```

In [13]: # importing libraries
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets

In [14]: # loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_deaths.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_confirmed.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_recovered.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()

```

Adaptado de Brocker y Schroeder (2023)

#### 4.3.4. Pandas

La librería de pandas tiene a sus disposición diversas estructuras de datos útiles para la muestra de largas listas de datos, los cuales se muestran en la tabla 4

Tabla 4: Estructuras de la librería Pandas.

Nombre	Descripción
Series	Una sola columna de datos, donde cada valor tiene una etiqueta, estas etiquetas son referidas como índices.
Dataframes	Los DataFrames guardan datos en el formato de tabla familiar de filas y columnas, algo similar a las hojas de cálculo, los DataFrames hacen las tareas de análisis más fácil.

Adaptado de McKinney et al. (2011)

En la tabla 5 se muestran algunas funciones útiles de la librería Pandas que fueron utilizadas para la recolección, manipulación y visualización del dataset.

Tabla 5: Algunas funciones de Pandas.

Función	Uso
read_csv()	Obtener datos de archivos .csv en el formato de dataframe.
head()	Mostrar los primeros 5 resultados de un dataframe.
dtypes()	Muestra una serie de datos de cada tipo de dato por columna.
shape()	Muestra una tupla que representa la dimensionalidad del dataframe.
describe()	Muestra estadísticas del data frame como el mínimo, máximo, desviación estándar, etc.
fillna()	recorta valores bajo ciertos parámetros.
groupby()	Agrupación de datos bajo ciertos criterios.
corr()	Usada para encontrar correlaciones entre las columnas del dataframe.

Adaptado de McKinney et al. (2011)

#### 4.3.5. Numpy

La base de la paquetería Numpy es el arreglo ndarray, es te encapsula arreglos dimensionales de n-ésimos elementos de tipos de datos homogéneos con muchas operaciones realizadas en código para mantener el rendimiento. algunas diferencias importante entre arreglos NumPy y arreglos convencionales de python son:

- Los arreglos Numpy tienen tamaños definidos al ser creados, no como las listas de python que pueden cambiar de tamaño dinámicamente.
- Se requiere que todos los elementos de un arreglo numpy sean del mismo tipo de datos, de esta manera serán del mismo tamaño en memoria.
- Los arreglos Numpy facilitan matemática avanzada así como otros tipos de operaciones en grandes cantidades de datos, es decir, las operaciones se hacen más eficientemente.

En la tabla 6 se muestra algunas funciones útiles usadas para el prototipo de entre la extensa variedad de Numpy:

Tabla 6: Funciones de arreglos de la librería Numpy.

Operaciones de arreglos	Funciones
Funciones de creacion de arreglos	np.array(), np.zeros(), np.ones(), np.empty(), etc.
Funciones de manipulacion de arreglos	np.reshape(), np.transpose(), etc.
Funciones matemáticas para arreglos	np.add(), np.subtract(), np.sqrt(), np.power(), etc.
Funciones estadísticas para arreglos	np.median(), np.mean(), np.std(), and np.var().
Funciones de entrada y salida en arreglos	np.save(), np.load(), np.loadtxt(), etc.

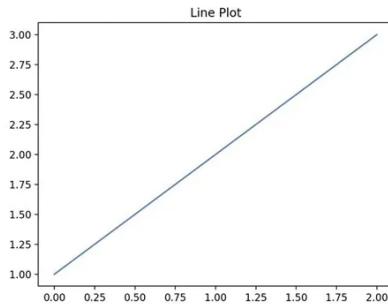
Adaptado de (Bressert, 2012)

Esta librería es fundamental para el desarrollo del prototipo dado que permite una fácil manipulación de arreglos que representan la división del dataset el dataset de pruebas y dataset de evaluación, además de funciones útiles para mezclar el orden de los datos durante el entrenamiento.

#### 4.3.6. Matplotlib

Esta es una librería multiplataforma comprensiva para la creación de elementos gráficos y visualizaciones estadísticas, animadas e interactivas en python, algunas de las estructuras gráficas que es capaz de mostrar son: histogramas, gráficas de puntos, gráficas de barras, etc. además de trabajar en conjunto con numpy con alternativas viables para MATLAB. En la figura 18 se muestra un ejemplo del tipo de graficas que se pueden realizar, esta ne particular siendo una linea simple que pasa por valores en un arreglo:

Figura 18: Ejemplo de gráfica de datos



Adaptado de (Ari y Ustazhanov, 2014)

La librería permite además la exportación hacia formatos de imagen populares como lo son PDF, SVG, JPG, PNG, BMP e incluso GIF así como otros estilos de visualización e incluso graficado en 3D. En la tabla 7 se muestran algunas funciones utilizadas dado que operan en arreglos de dos dimensiones, mismos que fueron los trabajados en el proyecto:

Tabla 7: Funciones de arreglos de la librería Matplotlib.

Funcion	Descripcion
imshow()	Muestra datos en una imagen, por ejemplo en graficas 2D simples.
matshow()	Mostrar un arreglo como una matriz en unanueva figura.
pcolor()	Crear una grafica semi coloreada en una cuadricula retctangular no regular
pcolormesh()	Crear una grafica semi coloreada en una cuadricula retctangular no regular
figimage()	Agregar una imagen sin muestrear a la figura.

Adaptado de (Ari y Ustazhanov, 2014)

#### 4.3.7. Scikit learn

Librería de código abierto para análisis de datos, pero además de esto especializada en Machine Learning dentro del ecosistema de python, algunos conceptos básicos y características que maneja son:

- Métodos algoritmos para toma de decisiones:
  - Clasificación, donde se identifican y categorizan datos basados en patrones.
  - Regresion, donde se predicen o proyectan valores de datos basados en promedios de datos existentes previamente planeados.
  - Agrupamiento en donde se debe realizar agrupaciones automáticas de datos con características similares hacia datasets.
- Algoritmos que dan soporte de análisis predictivo que van desde regresión lineal simple a reconocimiento de patrones mediante redes neuronales.
- interoperabilidad con las librerías Numpy, pandas y matplotlib.

Cabe destacar que Scikit learn (también abreviado como Sklearn) se utiliza solamente para crear los modelos de machine learning utilizados, no para lectura manipulación, cálculos o reportes de los datos, para esto es que se usan las otras librerías especializados en ello como lo son Numpy y Pandas. Otras características presentes en la librería y que son utilizadas durante la etapa de experimentación son:

- Algoritmos de aprendizaje supervisado.
- Algoritmos de aprendizaje no supervisado

- Validación cruzada. Métodos con los cuales revisar diversas métricas de precisión de modelos ya sean supervisados o no para tener una idea sobre el comportamiento de los datos durante los entrenamientos.

En la tabla 8 se muestran funciones usadas durante la construcción y comprensión del modelo de árbol de decisiones durante el entrenamiento:

Tabla 8: Funciones de Sklearn para árbol de decisiones.

Función	Descripción
tree.DecisionTreeClassifier(*[, criterion, ...])	Crear arbol de decisiones \\
tree.DecisionTreeRegressor(*[, criterion, ...])	Crear regresor de arbol de decisiones. \\
tree.ExtraTreeClassifier(*[, criterion, ...])	Crear arbol de decisiones con valores aleatorios
tree.ExtraTreeRegressor(*[, criterion, ...])	Crear regresor de arbol con valores aleatorios \\
tree.export_graphviz(decision_tree[, ...])	Exportar un arbol de decisiones en formato DOT. \\
tree.export_text(decision_tree, *[, ...])	Construir un reporte mostrando las reglas generadas del arbol de decisiones.
tree.plot_tree(decision_tree, *[, ...])	Graficar un arbol de decisiones.

Adaptado de Hackeling (2017)

#### 4.3.8. Arduino ESP32

La librería ESP32 contiene funciones y APIs compatibles con dicha familia de placas y componentes, entre estas se encuentran diversas versiones de la WEMOS S2 mini. En la tabla 9 se muestran algunas de las APIs particulares que se incluyen dentro de esta librería:

Tabla 9: Tecnologías compatibles con librería ESP 32

ADC	BLE	Bluetooth
DAC	Deep Sleep	ESP-NOW
Ethernet	GPIO	I2C
I2S	ESP Insights	LED Control (LEDC)
Preferences	Pulse Counter	ESP Rainmaker
Reset Reason	RMT	SDIO
SD MMC	Timer	Wi-Fi API

Adaptado de Babiuch et al. (2019)

Esta es fundamental para poder cargar código a la placa WEMOS S2 usando el IDE de arduino, además del necesario protocolo de comunicación I2C y el módulo integrador de wifi, por medio de los cuales se hace posible la conexión a redes cercanas y la comunicación entre la placa y el sensor de inclinación.

#### 4.3.9. Universtral Telegram Bot

Esta librería cuenta con varias características las cuales son listadas en las tablas 10 y 11:

Tabla 10: Algunas características relevantes de la librería parte 1

Característica	Descripción	Uso
Recibir mensajes	El bot puede enviar mensajes que le sean enviados, esto es útil para enviar comandos a la placa.	int getUpdates(long offset) obtiene cualquier mensaje pendiente y los almacena en bot.messages. la holgura debe ser asignada a bot.last-message-received + 1. lo cual devuelve los nuevos mensajes recibidos.
Enviar mensajes	El bot puede enviar mensajes a cualquier chat de Telegram. Esto puede ser útil para enviar mensajes desde la placa como notificaciones de eventos.	bool sendMessage(String chat-id, String text, String parse-mode = ). Envía un mensaje a chat-id. Devuelve cuando el mensaje sea enviado o no.
Teclados de respuestas	El bot puede enviar teclados de respuestas que pueden ser usados como menús.	chat-id, String text, String parse-mode, String keyboard, bool resize = false, bool oneTime = false, bool selective = false). Envía un teclado con respuestas pre establecidas a chat-id. parse-mode puede ser dejado en blanco. La función es verdadera si el mensaje se envía correctamente.
Teclados de linea	El bot puede enviar teclados en línea como URLs y respuestas de llamada a función.	bool sendMessageWithInlineKeyboard(String chat-id, String text, String parse-mode, String keyboard). Envía un teclado con respuestas pre establecidas a chat-id. parse-mode puede ser dejado en blanco.

Adaptado de Hasyim, Pramono et al. (2021)

Tabla 11: Algunas características relevantes de la librería parte 2

Enviar fotos	Es posible enviar fotos con el bot, se pueden mandar imágenes de internet o desde arduino directamente desde un sistema de almacenamiento.	-
Acciones de chat	El bot puede enviar acciones de chat como lo son escribir o enviar imágenes para hacer que el usuario sepa que el bot está haciendo algo.	bool sendChatAction(String chat-id, String chat-action). Envía una acción de chat hacia el chat-id especificado. Existen una lista de acciones de chat para el soporte de telegram, regresa respuesta si el mensaje se envia correctamente.

Adaptado de Hasyim, Pramono et al. (2021)

Universal Telegram Bot es una librería para el Arduino Ide para hacer uso de la API de telegram BOT, misma que está diseñada para hacer uso de diversas arquitectura de arduino. Esta hace uso de la red social telegram, un servicio de mensajería instantánea para enviar mensajes, esto es útil en el contexto de proyectos de arduino ya sea para recibir notificaciones de un proyecto o enviar comandos por medio de la aplicación de Telegram.

Para hacer uso de esto es necesaria también la librería complementaria de Arduino Json, además de un cliente de SSL y un token de bot, generado por medio del bot de chat que se encuentra en Telegram.

## 5. Construcción e implementación del prototipo

En este capítulo se describe de manera específica las distintas fases de elaboración del prototipo tanto en construcción física como lógica de programación.

### 5.1. Diseño de caja contenedora vía software CAD

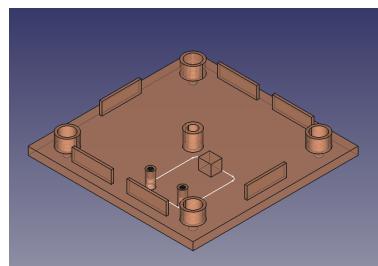
Al retomar la lista de áreas de oportunidad del prototipo previo el primero de los puntos a considerar es la construcción física de la caja. El nuevo diseño del contenedor se realiza mediante diseño asistido por computadora en el software FreeCAD. En primera instancia fue necesario idear dimensiones que al mismo tiempo fueran de menor tamaño y a su vez con espacio suficiente para albergar los componentes manejados, tras una lluvia de ideas y comparación de medidas se opta por un cubo con medida de 10 cm x 10 cm x 10 cm, lo que se traduce en 1000 cm cúbicos de espacio total utilizable donde ubicar los componentes.

Cinco de las tapas de acrílico actúan como caras y a la vez estructura del prototipo, siendo la sexta la tapa de acceso al interior, dada la facilidad con la que se puede realizar la manipulación de las piezas desde este ángulo se decide que tanto la placa como el sensor de inclinación y el protoboard de 170 puntos se ubiquen en este lado. Asimismo dado que los switches de inclinación solo presentan una corriente positiva en inclinación de 45° es necesario que cada uno de esos se ubique en el centro de cada cara, de esta manera se identifica con facilidad cuál es la cara activa, especialmente cuando se toma en cuenta que al ser un componente que trabaja de manera analógica (la bolita del interior del switch permite el paso de corriente cuando se completa el circuito) existe una posibilidad de que en un dado momento este activo más de un solo switch dado el movimiento de la propia caja.

Una situación particular se presenta al utilizar el sensor MPU6050, mismo que se ubica en la tapa junto con el resto de componentes, este debe estar posicionado lo más perpendicular posible a la tapa para dar las mejores mediciones acorde a la posición de reposo en la que se encuentre, para ello es asegurado con tornillos M2 en una posición que siga la cuadratura de la caja, de esta manera las componentes X e Y para la toma de valores se alinean con el sentido en que la caja cambia de dirección.

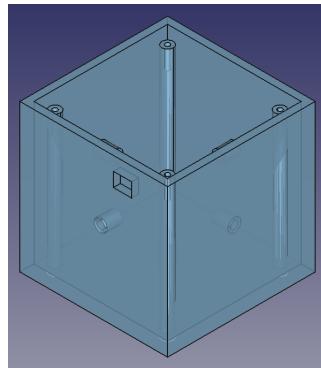
Las figuras 19 y 20 muestran distintos ángulos del diseño realizado en freecad:

Figura 19: Diseño conceptual de la tapa hecho en FreeCAD



Elaboración propia.

Figura 20: Diseño conceptual del contenedor hecho en FreeCAD

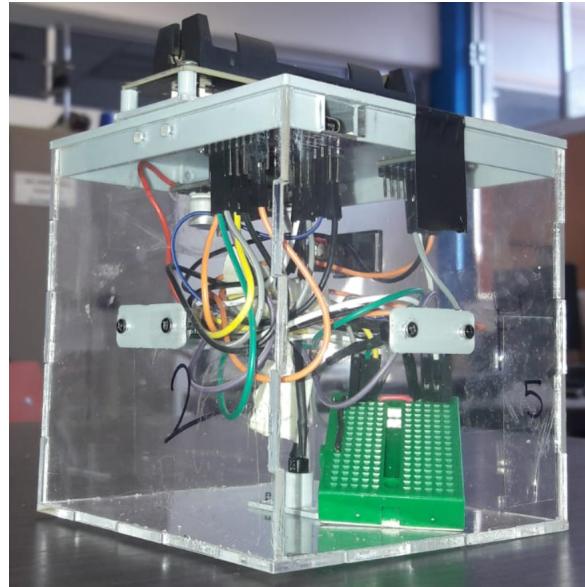


Elaboración propia.

En la etapa final del diseño ocurre una importante toma de decisiones partiendo del sketch realizado en software, debido a los costes, complejidad y accesibilidad a tecnologías de impresión 3D se decide que solo la tapa de acceso que alberga los componentes esté hecha del material de filamento mientras que el resto de caras sean de material de acrílico dada la poca complejidad de su diseño, el resultado de estas decisiones permite un ahorro monetario y de insumos considerable al usar menos filamento para impresión.

En la figura 21 puede ser apreciada la construcción final del nuevo prototipo propuesto:

Figura 21: Construcción final de la caja usando acrílico e impresión 3D

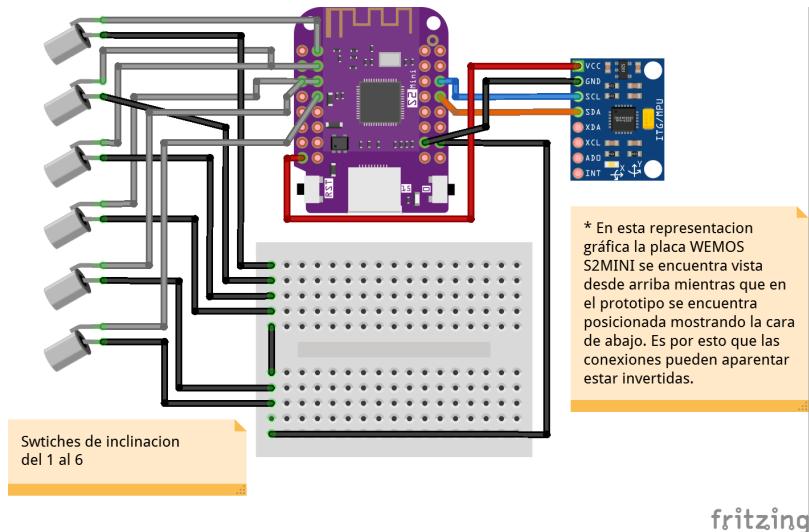


Elaboración propia.

## 5.2. Ensamblado de componentes electrónicos

En la figura 22 se puede apreciar un esquema realizado en el programa de diseño de sistemas electrónicos Fritzing donde se hace una muestra gráfica de los componentes y la manera en la que están conectados , esta no contempla la construcción completa incluyendo la distribución de las partes en la caja, solo el como deben estar comunicadas las piezas.

Figura 22: Esquema del prototipo usando la placa WEMOS S2 mini



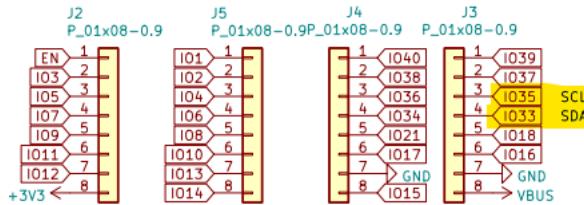
Elaboración propia.

Cada uno de los switches de inclinación se conecta a un pin del número correspondiente (1 al 6) en la placa WEMOS S2, de manera que se puedan identificar y diferenciar fácilmente los unos de los otros, además de esto cada switch se conecta a su propia tierra física, que a su vez es compartida con las tierras de la misma placa a través del mini protoboard de 170 puntos.

Del otro lado puede ser apreciado el sensor de inclinación MPU6050, de este son utilizados 4 pines: VCC por el cual se reciben hasta 5V para que funcione, GND como pin de tierra, SCL para sincronización de relojes entre dos controladores mediante el protocolo I2C junto con el pin de serial SDA para hacer el envío de datos hacia la placa WEMOS.

Un caso particular ocurre con los pines SDA y SCL de la placa WEMOS S2, estos no están explícitamente indicados en la nomenclatura mostrada, aun así, un vistazo rápido al esquemático nos indica que estos existen y están asignados a los pines IO35 e IO33 respectivamente, lo cual se puede ver en la figura 23.

Figura 23: Ubicación de SDA y SCL



Adaptado de wemos.cc (2021)

### 5.3. Método de entrada y muestra de datos

Una vez el prototipo se encuentra listo con todos los componentes en su lugar, se conecta a una corriente mediante cable de datos USB. Cuando esto sucede entra en acción el algoritmo programado en la placa. La función del código es la de tomar el estado actual proveniente de todos los componentes, es decir, los 6 switches de inclinación y 6 valores del sensor MPU6050.

En el caso de los switches la toma de datos es directa, obtenerlos de la señal analogia y almacenarlos en una variable, se vuelve más complejo al pasar con el sensor MPU6050 dado que este necesita ser previamente calibrado antes de hacer la toma de datos, de no hacerlo los datos que muestre serán casi aleatorios y no podrían ser usados para mediciones. El algoritmo 1 muestra como hacer la calibracion del giroscopio, esto es lo primero que se hace en el arduino.

---

#### Algoritmo 1 Calibración de sensor mpu6050

---

```

1: procedure CALIBRATEGYRO
2:   print Calibrando giroscopio...
3:   for (int i = 0; i < calibrationSamples; i++)
4:     gyroX, gyroY, gyroZ
5:     getRotation(gyroX, gyroY, gyroZ)
6:     totalAngularVelocity += gyroZ / sensitivity
7:     delay(10)
8:
9:   averageAngularVelocity = totalAngularVelocity / calibrationSamples
10:  sensitivity = averageAngularVelocity
11:
12:  print Calibración completa. Sensibilidad:
13:  print(sensitivity)
14: end procedure

```

Elaboración propia.

---

En la figura 24 se muestra el algoritmo de calibración implementado en el Arduino IDE.

Figura 24: Función de calibración de MPU6050 en C++

```

193
194 //Funcion de calibracion del giroscopio
195 void calibrategyro() {
196     Serial.println("Calibrando giroscopio...");
197
198     for (int i = 0; i < calibrationSamples; i++) {
199         int16_t gyroX, gyroY, gyroZ;
200         mpu.getRotation(&gyroX, &gyroY, &gyroZ);
201         totalAngularVelocity += gyroZ / sensitivity;
202         delay(10);
203     }
204
205     float averageAngularVelocity = totalAngularVelocity / calibrationSamples;
206     sensitivity = averageAngularVelocity;
207
208     Serial.print("Calibración completa. Sensibilidad: ");
209     Serial.println(sensitivity);
210 }
```

Elaboración propia.

Al momento de llamar la función esta no recibe ni envia ningun parametro por lo que se usa el prefijo void, el ciclo mostrado opera las veces establecidas por la variable calibrationSamples, lo que la función hace es tomar las rotaciones de los componentes x, y, z del giroscopio y almacenarlos en la variable totalAngularVelocity, posteriormente estos son utilizados para obtener el promedio de la velocidad angular promedio, usada como referencia para el posicionamiento del sensor, finalmente este promedio es asignado y referido como la variable sensitivity, misma que es mostrada en serial y establecida para el resto de operaciones.

en la figura 25 se muestra la toma de datos de todos los switches, sensores de inclinación y giroscopio:

Figura 25: Fragmento de código donde se da la toma de datos

```

64 //Obtencion de señal analoga del tilt switch a valores digitales.
65 tval1 = digitalRead(tp1);
66 tval2 = digitalRead(tp2);
67 tval3 = digitalRead(tp3);
68 tval4 = digitalRead(tp4);
69 tval5 = digitalRead(tp5);
70 tval6 = digitalRead(tp6);
71
72 //-----Comienza código de funcionamiento del giroscopio
73
74 int16_t accX, accY, accZ; //Componentes XYZ de Acelerómetro
75 int16_t gyroX, gyroY, gyroZ; //Componentes XYZ de Giroscopio
76
77 mpu.getAcceleration(&accX, &accY, &accZ); //Obtención de valores
78 mpu.getRotation(&gyroX, &gyroY, &gyroZ);
79
80 // Convertir los valores de aceleración y velocidad angular a unidades reales
81 float ax = (float)accX / 16384.0; // Dividir por 16384 para obtener g (gravedad, 9.81 m/s^2)
82 float ay = (float)accY / 16384.0;
83 float az = (float)accZ / 16384.0;
84
85 float gx = (float)gyroX / sensitivity; // Convertir a grados por segundo
86 float gy = (float)gyroY / sensitivity;
87 float gz = (float)gyroZ / sensitivity;
88
89 // Calcular los ángulos de inclinación en grados
90 float pitch = atan2(-ax, sqrt(ay * ay + az * az)) * 180.0 / PI;
91 float roll = atan2(ay, sqrt(ax * ax + az * az)) * 180.0 / PI;
92
93 // Asegurarnos de que los ángulos estén entre 0 y 360 grados
94 pitch = (pitch < 0) ? (pitch + 360) : pitch;
95 roll = (roll < 0) ? (roll + 360) : roll;
96

```

Elaboración propia.

Se hace lectura digital de los pines IO utilizados, es decir del 1 al 6 correspondientes a los switches, posteriormente se inicializan las componentes x, y, z del acelerómetro y del giroscopio, una vez establecidos como enteros de 16 bits se usan las funciones getAcceleration y getRotaton de la librería mpu, finalmente se obtienen los valores flotantes finales de los 6 valores del mpu, los valores ax, ay y az se dividen entre 16354 para que el resultado sea en relación a la gravedad. del mismo modo los valores gx, gy y gz se dividen entre el valor sensitivity anteriormente establecido para obtener grados sobre segundo.

Los valores pitch y roll mostrados aunque no son forzosamente necesarios para cálculos son buenos puntos de referencia para comprobar que los valores de las componentes sean lógicos y reproducibles.

En la figura 26 se muestra un fragmento de código cuya única función es hacer muestra en serial de los datos obtenidos, debido a que se encuentra dentro de la función loop() este código se estará ejecutando constantemente a 9600 bits por segundo con un delay de 200 milisegundos, por lo que habrá una actualización constante de los valores acorde a los cambios de posición del cubo.

Figura 26: Fragmento de código para muestra de datos.

```

136 //Muestra valores de clicks en el monitor Serial, en orden como
137 //Y por consecuente a la cara del cubo
138 Serial.print(tval1); //tilt 1
139 Serial.print(",");
140 Serial.print(tval2); //tilt 2
141 Serial.print(",");
142 Serial.print(tval3); //tilt 3
143 Serial.print(",");
144 Serial.print(tval4); //tilt 4
145 Serial.print(",");
146 Serial.print(tval5); //tilt 5
147 Serial.print(",");
148 Serial.print(tval6); //tilt 6
149
150
151
152
153 //Muestra X, Y, Z tanto del giroscopio como del acelerometro
154 Serial.print(",");
155
156 Serial.print(ax);
157 Serial.print(",");
158 Serial.print(ay);
159 Serial.print(",");
160 Serial.print(az);
161
162 Serial.print(",");
163
164 Serial.print(gx);
165 Serial.print(",");
166 Serial.print(gy);
167 Serial.print(",");
168 Serial.print(gz);
169

```

Elaboración propia.

Cabe recalcar que el resultado de la carga de este código en particular solamente es una fuente constante de datos separados por comas que representan el cambio de valores en el movimiento del cubo en tiempo real, lo que será utilizado para la posterior creación del dataset.

Estos datos se conforman de 12 valores en total, 6 valores binarios de los switches de inclinación y 6 valores flotantes del acelerómetro y giroscopio.

#### 5.4. Creación de Dataset a partir de salida de datos

A este punto del desarrollo la placa WEMOS S2 ha sido cargada con un programa en C++ la cual toma los datos de los sensores y los muestra en salida serial a lo largo del paso del tiempo, un ejemplo de esta salida se puede apreciar en la figura 27

Figura 27: Salida serial de datos segun componentes del cubo.

```

1,1,1,1,1,0,0.03,-0.02,0.90,1033.37,-39.70,12.41,6
1,1,1,1,1,0,0.04,-0.02,0.90,1011.04,-22.33,0.00,6
1,1,1,1,1,0,0.03,-0.02,0.90,1034.61,-4.96,-21.09,6
1,1,1,1,1,0,0.04,-0.02,0.90,1033.37,-39.70,12.41,6
1,1,1,1,1,0,0.03,-0.02,0.90,1034.61,-4.96,-21.09,6
1,1,1,1,1,0,0.04,-0.02,0.90,1011.04,-22.33,0.00,6
1,1,1,1,1,0,0.03,-0.02,0.90,1034.61,-4.96,-21.09,6
1,1,1,1,1,0,0.04,-0.02,0.90,1027.17,-9.92,-11.16,6
1,1,1,1,1,0,0.03,-0.02,0.90,1034.61,-4.96,-21.09,6
1,1,1,1,1,0,0.04,-0.02,0.90,1011.04,-22.33,0.00,6
1,1,1,1,1,0,0.03,-0.02,0.90,1033.37,-39.70,12.41,6
1,1,1,1,1,0,0.04,-0.02,0.90,1011.04,-22.33,0.00,6
1,1,1,1,1,0,0.03,-0.02,0.90,1034.61,-4.96,-21.09,6
1,1,1,1,1,0,0.04,-0.02,0.90,1027.17,-9.92,-11.16,6

```

Elaboración propia.

Esta muestra de datos en serie se hace como retroalimentación visual y además para mostrar datos necesarios para recolección, ahora es necesario generar un script con la capacidad de leer esa salida del serial que hace por cada momento la placa para posteriormente almacenar todos los datos producidos en un archivo de formato de tablas .csv.

En las figuras 28 y 29 se muestra lo que el script debe ser capaz de hacer. A grandes rasgos el propósito de este programa es: leer las salidas en serial de la placa, reconocer el formato separado por comas y colocar los 12 valores en celdas separadas a manera de lista y producir esta lista a un documento .csv

Figura 28: Programa para obtener y almacenar la salida serial de datos. Parte 1.

```

1 import serial
2 import time
3 import numpy as np
4 from collections import deque
5 import warnings
6 import itertools
7 import csv
8
9
10 hw_sensor = serial.Serial(port='/dev/ttyACM0', baudrate=9600)
11 hw_sensor.setDTR(False)
12 time.sleep(.1)
13 hw_sensor.flushInput()
14 hw_sensor.setDTR(True)
15
16 label = 'side_pred_test'
17 count_file = 1
18
19 M = []
20 data = []
21
22 f = open(f'{label}_{count_file}.csv', 'a')
23 w = csv.writer(f)
24

```

Elaboración propia.

Figura 29: Programa para obtener y almacenar la salida serial de datos. Parte 2

```

25 # write the header
26 w.writerow(['tVal2','tVal3','tVal4','tVal5','tVal6','tVal7','ax','ay','az','gx','gy','gz'])
27 with hw_sensor:
28     while True:
29         try:
30             line = hw_sensor.readline()
31             if not line:
32                 # HACK: Descartamos líneas vacías porque fromstring produce
33                 # resultados erróneos, ver https://github.com/numpy/numpy/issues/1714
34                 continue
35             # M.append(np.fromstring(Line.decode('ascii', errors='replace'), sep=','))
36             # data.append(yy)
37             tmp = line.decode('ascii', errors='replace').rstrip().split(',')
38             if (len(tmp) >= 5):
39                 data.append(tmp)
40
41             if (len(data) == 119):
42                 r = input('¿Guardar? ')
43                 if (r == 's'):
44                     for d in data:
45                         w.writerow(d)
46                     w.writerow([])
47                     data = []
48             except KeyboardInterrupt:
49                 print("Exiting")
50                 break
51

```

Elaboración propia.

El programa funciona tomando mediciones de la caja hasta alcanzar las 120 muestras, luego de esto preguntará al usuario si desea guardarlas, momento en el que serán almacenadas y el ciclo seguirá para almacenar otras 120 muestras, así por 5 ciclos hasta tener 410 muestras por cada cara, en las figuras 30 y 31 se muestra un ejemplo de la salida que da el programa al terminar de ejecutarse y el formato en el que se presentan los datos.

Figura 30: Salida de programa a archivo .csv.



Elaboración propia.

Figura 31: Formato de datos almacenados.

tVal2	tVal3	tVal4	tVal5	tVal6	tVal7	ax	ay	az	gx	gy	gz
0	1	1	1	1	1	-0.04	-0.01	-1.05	195.66	-3495.59	
0	1	1	1	1	1	-0.05	-0.01	-1.15	-2192.8	-2324.03	
0	1	1	1	1	1	-0.25	-0.01	-1.12	10680.04	30816.07	
0	1	1	1	1	1	-0.06	0.05	-0.97	8692.45	-48919.17	
0	1	1	1	1	1	-0.15	-0.1	-1.21	-22128.39	3245.05	
0	1	1	1	1	1	-0.11	-0.37	-1.1	-3571.94	16428.08	
0	1	1	1	1	1	-0.18	0.04	-0.98	54800.82	-29895.05	

Elaboración propia.

Cabe mencionar que dichas 410 muestras corresponden solo al movimiento de una sola cara, todo este proceso se repite 6 veces, una por cada cara donde el cubo es rotado manualmente, ajustes manuales se hacen para eliminar celdas vacías o mal colocadas, de tal manera que el archivo final cuenta con 2869 muestras correspondientes a todas las caras y diferenciadas por su clase dado que se trata de aprendizaje supervisado, es decir, una clase extra indica a que clase pertenece cada una de las muestras.

## 5.5. Implementación de modelo de aprendizaje de árbol y construcción del clasificador de árbol

De entre todos los tipos de clasificadores y modelos se utilizó un modelo de clasificación de árboles bajo el cual se entrenaría el dataset, esto por varias razones:

- Los árboles de decisión son fáciles de visualizar, generar e interpretar
- Son capaces de generar patrones no lineales
- Requieren pocos datos de pre procesamiento del usuario, por ejemplo no es completamente necesario normalizar columnas.
- Puede ser usado para características de ingeniería como lo son la predicción de valores faltantes y especialmente selección de variables.
- El árbol de decisiones no hace suposiciones de la distribución dado que es un algoritmo de naturaleza no paramétrica.

A grandes rasgos un algoritmo de árbol de decisión funciona de la siguiente manera:

1. Selecciona el mejor atributo usando un método de selección de atributos para dividir los valores.
2. Volver ese atributo a nodos de decisión que dividan el dataset en subsets mas pequeños.
3. El árbol comienza a construirse al repetir estos procesos de manera recursiva para cada nodo hijo hasta que una de estas condiciones se consiga:

- a) Que todas las tuplas pertenezcan al mismo valor de atributo.
- b) Que no haya atributos restantes.
- c) Que no haya más instancias las cuales recorrer.

Estos puntos son particularmente importantes dado que este es un problema con salida multiclass, es decir, la salida de la predicción no es binaria sino que hay mas de dos posibles clases (del 1 al 6 correspondiente a cada cara).

Utilizando el entorno web de jupyter lab en un entorno virtual de linux WSL se crea un notebook sobre el cual se ejecuta código de python para tomar el dataset de lados previamente creado, extraer los datos y crear el modelo.

En la figura 32 se muestra la importación de librerías necesarias para la toma y manipulación de datos a partir del dataset en formato .csv. Pandas, Numpy y Sklearn son librerías separadas pero complementarias para métodos estadísticos y matemáticos así como la carga del propio clasificador de árbol, matplotlib lo es también, salvo que este se utiliza para la graficación del modelo.

Figura 32: Notebook de clasificador Parte 1. Importe de librerias.

```
[1]: import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import datasets
import os
np.random.seed(42)

%matplotlib inline
```

Elaboración propia.

La primera parte del modelo es la carga del dataset, la función de este fragmento de código es leer el archivo .csv el dataset llamado “Side-Face-Dataset-Clean” y quitar las filas que contengan celdas vacías, finalmente mostrando los primeros cinco elementos del archivo. En este punto el dataset está almacenado en la variable cubeSides-Dataset, Algo notable del fragmento mostrado es que los nombres de las características se encuentran en la primera fila lo cual recorre el tamaño del dataset. Todo lo anterior mencionado puede ser visto en la figura 33:

Figura 33: Notebook de clasificador Parte 2. Carga de dataset a variable.

```
[2]: cubeSides_Dataset = pd.read_csv("Side_Face_Dataset_Clean.csv", header=None)
cubeSides_Dataset.dropna(inplace = True)
cubeSides_Dataset.head()

[2]:   0   1   2   3   4   5   6   7   8   9   10  11  12
  0 tVal2 tVal3 tVal4 tVal5 tVal6 tVal7    ax    ay    az      gx      gy      gz  face
  1  0     1     1     1     1     1 -0.04 -0.01 -1.05  195.66 -3495.59 3536.15    1
  2  0     1     1     1     1     1 -0.05 -0.01 -1.15 -2192.8 -2324.03 2901.46    1
  3  0     1     1     1     1     1 -0.25 -0.01 -1.12 10680.04 30816.07 -25010.77    1
  4  0     1     1     1     1     1 -0.06  0.05 -0.97  8692.45 -48919.17 21092.84    1
```

Elaboración propia.

Una vez se tienen los datos almacenados en una variable es importante tener información general acerca de este para lo cual se usa el comando `describe()`. La salida mostrada de este aplicado a la variable del datafram muestra un corto resumen de los datos, Esto puede ser visto en la figura 34:

Figura 34: Notebook de clasificador Parte 3. Descripción de dataset.

```
[3]: cubeSides_Dataset.describe

[3]: <bound method NDFrame.describe of
      0   1   2   3
      0 tVal2 tVal3 tVal4 tVal5 tVal6 tVal7    ax    ay    az
      1  0     1     1     1     1     1 -0.04 -0.01 -1.05
      2  0     1     1     1     1     1 -0.05 -0.01 -1.15
      3  0     1     1     1     1     1 -0.25 -0.01 -1.12
      4  0     1     1     1     1     1 -0.06  0.05 -0.97
      ...
      ...   ...   ...   ...   ...   ...   ...   ...   ...
      2864  1     1     1     1     1     1    0     0   -0.57  0.54
      2865  1     0     1     1     1     1    0   -0.02  -0.18  0.86
      2866  1     1     1     1     1     1    0   -0.42   0.09  0.86
      2867  1     1     1     1     1     1    0   -0.06   0.28  0.84
      2868  1     1     1     1     1     1    0   0.66   0.56  0.81

      9   10   11   12
      0      gx      gy      gz  face
      1  195.66 -3495.59 3536.15    1
      2 -2192.8 -2324.03 2901.46    1
      3 10680.04 30816.07 -25010.77    1
      4  8692.45 -48919.17 21092.84    1
      ...
      ...   ...   ...   ...
      2864 29453.62 3879.75 -21720.38    6
      2865 -31560.52 -10119.31 -29813.92    6
      2866 -11231.22 -2612.74 -38709.18    6
      2867 -5485.57 10756.39 -28597.03    6
      2868 23731.83 33197.36 45268.48    6
```

Elaboración propia.

Los datos mostrados tienen un error en particular el cual es que los encabezados de cada columna están posicionados como la primera fila mientras que el dataset se mantiene sin encabezados aun cuando el archivo .csv se guardó con ellos. Las líneas de código mostradas en la figura 35 solamente eliminan la primera fila del dataset.

Figura 35: Notebook de clasificador Parte 4. Corrección de características.

```
[4]: cubeSides_Dataset.columns = cubeSides_Dataset.iloc[0]
cubeSides_Dataset = cubeSides_Dataset[1:]
```

Elaboración propia.

Hasta este punto el dataset está completo salvo por los encabezados, los cuales son establecidos con la línea mostrada en la figura 36, una vez aplicada la función se hace una muestra de datos para comprobar que los cambios han sido realizados. Cabe destacar que dado que se está trabajando con aprendizaje supervisado en todo momento se mantiene relevante la columna f, que corresponde a la cara a las que pertenecen dichas filas, por ende son 13 columnas, 12 características + 1 de la clase.

Figura 36: Notebook de clasificador Parte 5. Asignación de características a encabezado.

```
[5]: cubeSides_Dataset.columns = ['t1','t2','t3','t4','t5','t6','ax','ay','az','gx','gy','gz','f']
cubeSides_Dataset.head()
```

	t1	t2	t3	t4	t5	t6	ax	ay	az	gx	gy	gz	f
1	0	1	1	1	1	1	-0.04	-0.01	-1.05	195.66	-3495.59	3536.15	1
2	0	1	1	1	1	1	-0.05	-0.01	-1.15	-2192.8	-2324.03	2901.46	1
3	0	1	1	1	1	1	-0.25	-0.01	-1.12	10680.04	30816.07	-25010.77	1
4	0	1	1	1	1	1	-0.06	0.05	-0.97	8692.45	-48919.17	21092.84	1
5	0	1	1	1	1	1	-0.15	-0.1	-1.21	-22128.39	3245.05	8255.79	1

Elaboración propia.

Con el dataset listo se separan las filas de datos de las características para luego separar las filas restantes en set de entrenamiento y set de prueba, las librerías importadas facilitan esto utilizando la función train\_test\_split(), que automáticamente almacena las divisiones de datasets en valores separados, esta función recibe los datos X que corresponde a todas las filas, y que corresponde a todas las columnas, test\_size que corresponde al porcentaje de los datos que serán asignados como datos de prueba, en este caso, se usa un 20% del total de datos para pruebas y random\_state siendo una variable semilla para la mezcla aleatoria de las filas, el código que aplica esto se muestra en la figura 37:

Figura 37: Notebook de clasificador Parte 6. Separación de dataset y características en entrenamiento y prueba.

```
[6]: X = cubeSides_Dataset.iloc[:, :-1]
y = cubeSides_Dataset.iloc[:, -1]

# split the dataset
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42)
```

Elaboración propia.

Tras hacer el pre-procesamiento de los datos llega el momento de crear el clasificador de árbol, al notebook se importan las funciones de DecisionTreeClassifier, parte de la librería sklearn. La función DecisionTreeClassifier() crea el clasificador de árbol que será usado y utiliza las funciones criterion la cual indica el uso de alguno de los criterios del modelo, ya sea gini, log\_loss o entropy, para este caso se indica específicamente el criterio gini, max\_depth para establecer la profundidad de nodos del árbol, en este caso profundidad de 5 y random\_state como valor semilla de la generación aleatoria. Posteriormente de la creación del modelo se entrena con los datos en la función fit(), la cual utiliza X\_train como los datos designados para entrenamiento y luego contraponerse con las características de la variable y\_train, lo que se ve representado en la figura 38:

Figura 38: Notebook de clasificador Parte 7. Creación y entrenamiento de clasificador de árbol.

```
[7]: from sklearn.tree import DecisionTreeClassifier

tree_Pred_Side = DecisionTreeClassifier(criterion='gini', max_depth=5, random_state=1)
tree_Pred_Side.fit(X_train, y_train)

[7]: ▾ DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5, random_state=1)
```

Elaboración propia.

El modelo ya ha sido creado y entrenado con los datos designados para ello, el siguiente paso consiste en evaluar usando los sets de prueba para medir su precisión. La precisión es una métrica sencilla que describe la razón de predicciones correctas sobre predicciones hechas, es decir el porcentaje de acierto directo acerca del set de entrenamiento en comparación con el de pruebas. Se utiliza la función format() con el modelo creado y entrenado para que evalúe tanto los sets de entrenamiento como los sets de prueba, esto sería el modelo comparando con datos previamente vistos y además datos nuevos desde el punto de vista del modelo. El código y resultado de esto puede ser visto en la figura 39, ambas pruebas con precisión mayor

al 0.95, lo que indica que hace predicciones correctas el 96 % de las veces, lo cual es ideal dado que presenta un excelente porcentaje de aciertos pero sin llegar al overfitting.

Figura 39: Notebook de clasificador Parte 8. Precisión de set de prueba y set de entrenamiento del modelo.

```
[8]: print('Accuracy of Decision Tree classifier on training set: {:.2f}'
       .format(tree_Pred_Side.score(X_train, y_train)))
print('Accuracy of Decision Tree classifier on test set: {:.2f}'
       .format(tree_Pred_Side.score(X_test, y_test)))

Accuracy of Decision Tree classifier on training set: 0.97
Accuracy of Decision Tree classifier on test set: 0.96
```

Elaboración propia.

En la figura 40 se muestra el uso de la función classification\_report, el objetivo de esta es utilizar el set de entrenamiento y el set de pruebas para aplicar diversas métricas de evaluación para observar el comportamiento del modelo, la salida de esta función siendo una tabla con porcentajes de evaluación no solo por métrica sino también por característica, en este caso particular todas superan el 90 % de acierto, por ende se puede decir que por cada característica el modelo hace buenas predicciones, lo que lo vuelve un candidato importante para su uso en el prototipo.

Figura 40: Notebook de clasificador Parte 9. Reporte de clasificación del clasificador de arbol.

```
[9]: predictions = tree_Pred_Side.predict(X_test)

from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))

precision    recall   f1-score   support
      1        1.00     0.92      0.96     104
      2        0.97     0.97      0.97      98
      3        0.98     0.98      0.98     107
      4        0.90     0.98      0.94      89
      5        0.93     0.96      0.94      77
      6        0.99     0.96      0.97      99

accuracy                           0.96      574
macro avg       0.96     0.96      0.96      574
weighted avg    0.96     0.96      0.96      574
```

Elaboración propia.

### 5.5.1. Transformando el modelo de clasificador de árbol

A este punto del desarrollo se puede decir que existe ya un clasificador de árbol entrenado con datos provenientes de la caja de emociones y que además sus resultados demuestran que

tiene buena efectividad, el siguiente paso es adaptar este modelo a la placa WEMOS S2 mini, de tal manera que sea la placa la que haga las inferencias según se haga el movimiento, esto genera algunos problemas:

- Capacidad de memoria y procesamiento de la placa.
- Escasez de métodos accesibles para cargar el modelo entrenado.
- Ajuste y transformación de código .py a .cpp.

Existen diversos proyectos para cargar modelos en dispositivos embebidos, el problema de estos suele ser la complejidad y la necesidad de componentes extra en el diseño como lo son lectores de memoria micro SD, así como directamente compatibilidad limitada para usar ciertos modelos en placas particulares.

La solución de estos problemas se da al analizar el modelo utilizado, en concepto un clasificador de árbol se denomina como un árbol de decisiones binarias, el modelo puede abarcar tantos nodos como sea posible y tener una profundidad alta, más aún al final del día las unidades básicas de este serán decisiones binarias que llegan a las clases posibles.

Siguiendo este concepto, una gran cantidad de estructuras if-else son más fáciles de procesar por la placa, al final de la construcción de estructura anidadas se llega a la inferencia de la clase basado en comparaciones establecidas.

Se hace importe de las librerías pydot plus y sklearn para poder extraer las características del modelo, estructurarlas en formato de árbol y convertirlo a imagen en formato .png. Posteriormente se utiliza la función export\_graphviz() para exportar el modelo de árbol específico a su estructura en imagen, parámetros importantes de la función son class\_names donde se define los nombres de las clases, en este caso del 1 al 6 según la cara, y feature\_names donde se nombran todas las características usadas para el entrenamiento, importante recalcar que ambas funciones deben contener los mismos parámetros con los cuales se realizó el entrenamiento, lo anterior se puede ver en la figura 41:

Figura 41: Graficando el clasificador.

```
from pydotplus import graph_from_dot_data
from sklearn.tree import export_graphviz

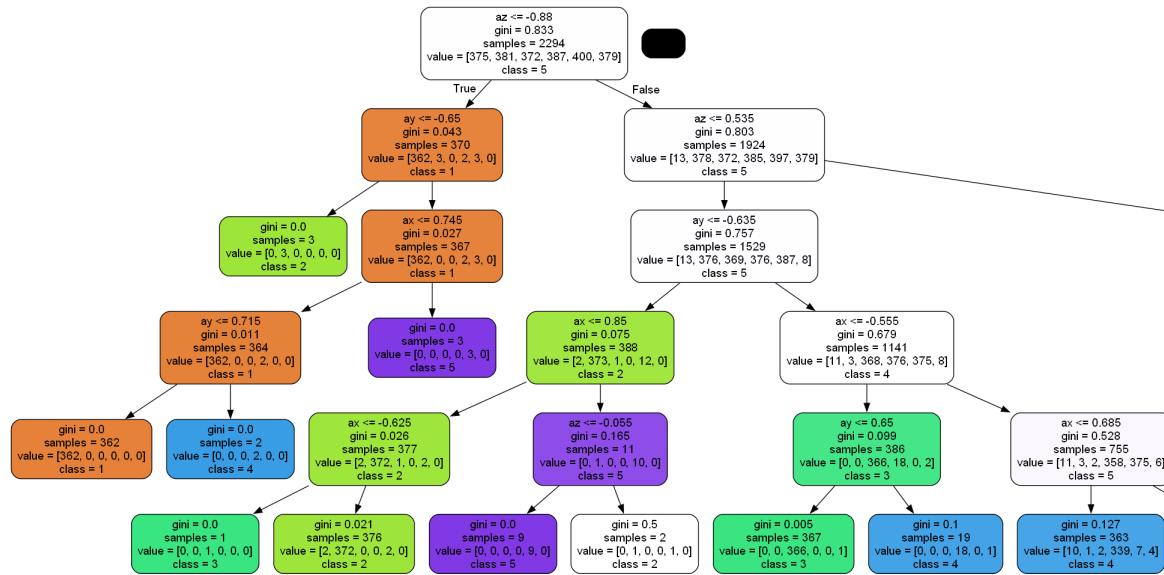
dot_data = export_graphviz(tree_Pred_Side, filled=True, rounded=True, class_names=['1', '2', '3', '4','5','6'], feature_names=['t1','t2','t3','t4','t5'])
graph = graph_from_dot_data(dot_data)
graph.write_png('Side_Pred_Tree.png')
```

Elaboración propia.

Como resultado de la ejecución del código se crea el archivo Side\_pred\_Tree.png, como la extensión lo indica es una imagen la cual contiene la construcción gráfica del clasificador de árbol. Como puede ser observado en las figuras 42 y 43 el árbol consiste de una diversa

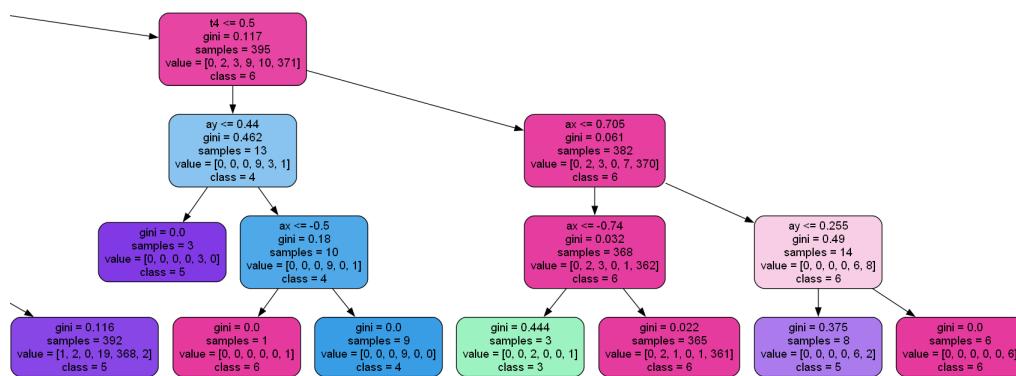
cantidad de nodos con una profundidad máxima de 5, cada clase se ha atribuido una serie de especificaciones representadas en estos árboles binarios dentro de la estructura mayor.

Figura 42: Clasificador de arbol para caja de emociones. Parte 1.



Elaboración propia.

Figura 43: Clasificador de árbol para caja de emociones. Parte 2.



Elaboración propia.

Cabe recalcar que el dato de entrada para este árbol debe consistir en una cadena de 12 valores separados por comas, basado en la magnitud de los valores que se presenten el modelo es

capaz de definir la clase a la que dicha cadena pertenece, es decir, en qué posición se encuentra la caja en ese momento particular.

## 5.6. Descomposición del clasificador de árbol en estructuras If-Else anidadas

El gráfico del clasificador de árbol es de gran utilidad para atender la problemática de implementar el modelo en la placa WEMOS S2, este modelo se conforma de una serie de sentencias If-else anidadas para llegar a la inferencia de la cara, la función del graficado del árbol no solo sirve como representación visual sino que es la guía para hacer la estructura.

Se hace uso de una función que es capaz de tomar la estructura y describirla directamente como es necesario, esto se puede ver en la figura 44, el resultado es una descripción escrita de las decisiones ya estructuradas.

Figura 44: Desglose de clasificador de arbol.

```
from sklearn import tree as arbol

text_rep = arbol.export_text(tree_Pred_Side)
print(text_rep)

|--- feature_8 <= -0.88
|   |--- feature_7 <= -0.65
|   |   |--- class: 2
|   |--- feature_7 >  -0.65
|   |   |--- feature_6 <= 0.75
|   |       |--- feature_7 <= 0.72
|   |           |--- class: 1
|   |           |--- feature_7 >  0.72
|   |               |--- class: 4
|   |       |--- feature_6 >  0.75
|   |           |--- class: 5
|--- feature_8 >  -0.88
```

Elaboración propia.

El resultado completo de este código se muestra en el siguiente listado:

Listing 1: Python example

```
|--- feature_8 <= -0.88
|   |--- feature_7 <= -0.65
|   |   |--- class: 2
|   |--- feature_7 >  -0.65
|   |   |--- feature_6 <= 0.75
|   |       |--- feature_7 <= 0.72
```

```

    |   |   |   |--- class: 1
    |   |   |   |--- feature_7 >  0.72
    |   |   |   |--- class: 4
    |   |   |--- feature_6 >  0.75
    |   |   |--- class: 5
    |   |--- feature_8 > -0.88
    |   |--- feature_8 <= 0.53
    |   |   |--- feature_7 <= -0.63
    |   |   |   |--- feature_6 <= 0.85
    |   |   |   |   |--- feature_6 <= -0.62
    |   |   |   |   |--- class: 3
    |   |   |   |   |--- feature_6 > -0.62
    |   |   |   |   |--- class: 2
    |   |   |   |--- feature_6 >  0.85
    |   |   |   |   |--- feature_8 <= -0.06
    |   |   |   |   |--- class: 5
    |   |   |   |   |--- feature_8 > -0.06
    |   |   |   |   |--- class: 2
    |   |   |--- feature_7 > -0.63
    |   |   |   |--- feature_6 <= -0.56
    |   |   |   |   |--- feature_7 <= 0.65
    |   |   |   |   |--- class: 3
    |   |   |   |   |--- feature_7 >  0.65
    |   |   |   |   |--- class: 4
    |   |   |   |--- feature_6 > -0.56
    |   |   |   |   |--- feature_6 <= 0.69
    |   |   |   |   |--- class: 4
    |   |   |   |   |--- feature_6 >  0.69
    |   |   |   |   |--- class: 5
    |--- feature_8 >  0.53
    |--- feature_3 <= 0.50
    |   |--- feature_7 <= 0.44
    |   |   |--- class: 5
    |   |--- feature_7 >  0.44
    |   |   |--- feature_6 <= -0.50
    |   |   |   |--- class: 6
    |   |   |--- feature_6 > -0.50
    |   |   |   |--- class: 4
    |--- feature_3 >  0.50
    |--- feature_6 <= 0.70
    |   |--- feature_6 <= -0.74
    |   |   |--- class: 3
    |   |--- feature_6 > -0.74
    |   |   |--- class: 6
    |--- feature_6 >  0.70

```

```

|   |   |   |
|   |   |   |--- feature_7 <= 0.26
|   |   |   |   |--- class: 5
|   |   |   |--- feature_7 > 0.26
|   |   |   |   |--- class: 6

```

Con esta guía se construye una función que sera llamada "prediction", el uso directo de esta es tomar todos los valores de los sensores y regresar la clase inferida como el valor de la cara.

## 5.7. Script de predicción de cara para placa WEMOS S2 mini

El resultado del proceso de desarrollo es un archivo de extension .ino escrito en lenguaje C++ que puede ser cargado desde el IDE de Arduino hacia una placa WEMOS S2 mini, una carga exitosa de este script permitirá que la caja de emociones haga conexión con un bot de chat de telegram, de haber una red disponible la placa utiliza las funcionalidades de conexión wifi comenzara a enviar mensajes al chat con la inferencia de la cara que este activa en la caja.

### 5.7.1. Archivo S2mini\_sideclassifier\_TelegramBot\_Candidate.ino

El código comienza al importar varias librerías necesarias para el funcionamiento, en particular Wifi.h que trae consigo funciones para el uso del modulo wifi de la placa, WiFiClienteSecure.h, universalTelegramBot.h y ArduinoJson.h son complementarios para que la placa sea capaz no solo de utilizar credenciales y red wifi sino comunicarse con el bot de telegram. En la figura 45 se establecen los valores ssid y password, los cuales corresponden a nombre de red para conectar y su respectiva contraseña, así como BOTtoken y CHAT\_ID que son las credenciales de identificación para enviar mensajes al bot de telegram.

Figura 45: Importe de librerías.

```

20
21 #ifdef ESP32
22   #include <WiFi.h>
23 #else
24   #include <ESP8266WiFi.h>
25 #endif
26 #include <WiFiClientSecure.h>
27 #include <UniversalTelegramBot.h> // Universal Telegram Bot Library written
28 #include <ArduinoJson.h>
29
30 //const char* ssid = "AP-I120"; // Credenciales Lab
31 //const char* password = "TrGhFdIX";
32
33 const char* ssid = "INFINITUM6059"; // Credenciales casa
34 const char* password = "1CEF568FA7";
35
36 //Credenciales 687
37 //const char* ssid = "los odio a todos";
38 //const char* password = "noseprogramarmevoyapymes";
39
40 #define BOTtoken "6338955865:AAExN5ySiig3nJDalMdnB7XnC7Sw9QgJJ0" // your Bot
41 #define CHAT_ID "6563493359"
42 #ifdef ESP8266
43   X509List cert(TELEGRAM_CERTIFICATE_ROOT);
44 #endif
45
46 WiFiClientSecure client;
47 UniversalTelegramBot bot(BOTtoken, client);
48

```

Elaboración propia.

En la figura 46 se muestra el establecimiento de los valores botRequestDelay y lastTimeBotRan que permiten inferir la velocidad en la que ciclan los mensajes hacia el bot, ledPin y ledState que definen el led integrado en la placa. Finalmente se incluyen las librerías para que operen los componentes MPU6050 con el protocolo I2Cdev.

Figura 46: Inicialización de variables. Parte 1

```

// Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

const int ledPin = 15;
bool ledState = HIGH;
//-----

#include "MPU6050.h"
#include "I2Cdev.h"
#include "Wire.h"
MPU6050 mpu;

```

Elaboración propia.

Tras los importes necesarios se inicializan otras variables necesarias, calibrationSamples, sensitivity y totalAngularVelocity se utilizan para la calibración del giroscopio, pinled así como tp1 hasta tp6 indican los pines I/O de la placa, tval1 hasta tval6 variables para almacenar

los valores de los switches de inclinacion, last\_pred lleva la cuenta de la predicción inmediata anterior, pred\_mode define si el comando de predicción ha sido usado, esto puede ser visto en la figura 47.

Figura 47: Inicialización de variables. Parte 2

```

64 const int calibrationSamples = 200;
65 float sensitivity = 50.0;
66 float totalAngularVelocity = 0.0;
67
68 // tp. tiltpin. numero de pin acorde a construccion fisica.
69 const int pinled = 15; //Led integrado de la placa
70 const byte tp1 = 1;
71 const byte tp2 = 2;
72 const byte tp3 = 3;
73 const byte tp4 = 4;
74 const byte tp5 = 5;
75 const byte tp6 = 6;
76
77 int tval1, tval2, tval3, tval4, tval5, tval6; // tval. valor del tilt. puede ser
78 int Shown_Face; // Valor de la cara mostrada. 1 a 6 segun la cara activa hacia ar
79
80 int last_pred = -1;
81
82 int pred_mode = -1; //Valor bandera para predicción activada o desactivada
83

```

Elaboración propia.

### 5.7.2. Función setup()

La función setup() solo se ejecuta una sola en vez en el programa y se utiliza para establecer variables que se usaran en la ejecución. en la figura 48 se inicializan los pines I/O que serán usados para la transmisión de datos, un dato importante es que en esta sección se llama a la función CalibrateGyro(), la cual hará calibración del giroscopio para dar operaciones precisas.

Figura 48: Función setup 1. Inicio de pines en placa.

```

131 void setup() {
132   // put your setup code here, to run once:
133
134   Serial.begin(9600);
135
136   Wire.begin();      //Conexion con el MPU
137   mpu.initialize(); //Inisializacion del mpu
138   calibrateGyro(); //Funcion de calibracion de giroscopio
139
140   pinMode(pinled, OUTPUT);
141   digitalWrite(pinled, LOW);
142
143   pinMode(tp1, INPUT_PULLUP);    //Declaracion de pinesára señals de los tilts
144   pinMode(tp2, INPUT_PULLUP);
145   pinMode(tp3, INPUT_PULLUP);
146   pinMode(tp4, INPUT_PULLUP);
147   pinMode(tp5, INPUT_PULLUP);
148   pinMode(tp6, INPUT_PULLUP);
149
150   //Setup telegramBOT----WIFI-----
151 #ifdef ESP8266
152   configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
153   client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
154 #endif
155
156   pinMode(ledPin, OUTPUT);
157   digitalWrite(ledPin, ledState);
158

```

Elaboración propia.

En la figura 49 se muestra la sección de código de la función setup() donde se toman las credenciales tanto de la librería Wifi como de universal telegram bot, un ciclo simple trata de hacer conexión mientras muestra el mensaje “Connecting to WiFi...”, solo cuando la conexión se establece se envia un único mensaje de confirmación al chat de Telegram.

Figura 49: Función setup 2. Conexión Wifi y Telegram

```

159   // Connect to Wi-Fi
160   WiFi.mode(WIFI_STA);
161   WiFi.begin(ssid, password);
162   #ifdef ESP32
163     client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
164   #endif
165   while (WiFi.status() != WL_CONNECTED) {
166     delay(1000);
167     Serial.println("Connecting to WiFi..");
168   }
169   // Print ESP32 Local IP Address
170   Serial.println(WiFi.localIP());
171
172   bot.sendMessage(CHAT_ID, "Setup done - Emotion Box Connected - use /start to begin.", "");
173 }
174

```

Elaboración propia.

### 5.7.3. Función calibrateGyro()

Los datos sin filtrar que son obtenidos del sensor giroscopio carecen de un sentido o patrón, por lo que es necesario utilizar una calibración de estos valores. En la figura 50 se muestra la

funcion de calibracion en donde una cantidad arbitaria de muestras definidas por la variable calibrationSamples iterara un ciclo for que toma los valores de rotacion gyroX, gyroY y gyroZ para promediarlo almacenarlo en la variable totalAngularVelocity, esta ultima para generar el valor sensitivity que sera usado para cálculos posteriores.

Figura 50: Función de calibración de giroscopio MPU6050.

```

377 void calibrateGyro() {
378     Serial.println("Calibrando giroscopio...");
379
380     for (int i = 0; i < calibrationSamples; i++) {
381         int16_t gyroX, gyroY, gyroZ;
382         mpu.getRotation(&gyroX, &gyroY, &gyroZ);
383         totalAngularVelocity += gyroZ / sensitivity;
384         delay(10);
385     }
386
387     float averageAngularVelocity = totalAngularVelocity / calibrationSamples;
388     sensitivity = averageAngularVelocity;
389
390     Serial.print("Calibración completa. Sensibilidad: ");
391     Serial.println(sensitivity);
392 }
393

```

Elaboración propia.

#### 5.7.4. Función loop()

Una característica particular del IDE de arduino es la función loop(), esta es una función que estará ciclando constantemente y constituye el funcionamiento del programa. En la figura 51 el fragmento de código presentado se utiliza para constantemente verificar si hay mensajes nuevos enviados por el usuario, notifica en serial en el momento que se recibe un nuevo mensaje.

Figura 51: Chequeo de mensajes nuevos.

```

void loop() {
    //Serial.println("Loop cycled"); //tilt 1

    if (millis() > lastTimeBotRan + botRequestDelay) {
        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        while(numNewMessages) {
            Serial.println("got response");
            handleNewMessages(numNewMessages);
            numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        }
        lastTimeBotRan = millis();
    }
}

```

Elaboración propia.

En al figura 52 se muestra la siguiente parte de la función, en esta solo se revisa si la variable pred\_mode tiene un valor de 1, este valor es dado cuando se utiliza el comando de iniciar

predicción, al hacerlo constantemente se llamará a la función prediction\_active(), provocando que se haga la predicción de la cara, finalmente se establece un retraso de 1000 milisegundos entre cada iteración.

Figura 52: Modo de predicción activo.

```
//-----Finaliza código de funcionamiento del giroscopio

//Muestra valores de tilts en el monitor serial. el orden corresponde al tilt switch físico
//Y por consecuente a la cara del cubo

//String cube_values = tval1 + ",";
if (pred_mode == 1){
| prediction_active();
}

// bot.sendMessage(chat_id, predToShow, "");

//parpadeo de led integrado para confirmación visual física.
delay(1000);

}
```

Elaboración propia.

### 5.7.5. Función handleNewMessages()

Esta función se utiliza para manejar los mensajes que se generan en el chat de Telegram. Esta función recibe mensajes nuevos y los muestra en serial como confirmación, después establece el mensaje de bienvenida para el comando /start, esto se puede ver en la figura 53;

Figura 53: Manejador de mensajes y mensaje de bienvenida.

```
84 void handleNewMessages(int numNewMessages) {
85     Serial.println("handleNewMessages");
86     Serial.println(String(numNewMessages));
87
88     for (int i=0; i<numNewMessages; i++) {
89         // Chat id of the requester
90         String chat_id = String(bot.messages[i].chat_id);
91         if (chat_id != CHAT_ID){
92             bot.sendMessage(chat_id, "Unauthorized user", "");
93             continue;
94         }
95
96         // Print the received message
97         String text = bot.messages[i].text;
98         Serial.println(text);
99
100        String from_name = bot.messages[i].from_name;
101
102        if (text == "/start") {
103            String welcome = "Welcome, " + from_name + ".\n";
104            welcome += "Use the following commands to control your outputs.\n\n";
105            welcome += "/pred_start to start receiving the face prediction \n";
106            welcome += "/pred_stop to stop receiving the face prediction \n";
107            welcome += "/box_state to show current box side \n";
108            bot.sendMessage(chat_id, welcome, "");
109            Serial.println("/start used");
110        }
111    }
```

Elaboración propia.

En esta función también se definen los comandos disponibles y sus acciones mediante coincidencia de caracteres, ambos comandos muestran un mensaje de confirmación en chat y en monitor serial como se ve en la figura 54. El resultado son dos comandos, activación y desactivación de la función para predecir la cara del cubo a través de la variable pred\_mode, 1 para activar, -1 para desactivar.

Figura 54: Comandos de chat para usuario.

```

112
113     if (text == "/pred_start"){
114         Serial.println("/pred_start used");
115         bot.sendMessage(chat_id, "Side Prediction Started - Box will output side when rotated.", "");
116         pred_mode = 1;
117     }
118
119     if (text == "/pred_stop"){
120         Serial.println("/pred_stop used");
121         bot.sendMessage(chat_id, "Side Prediction Stopped - Box won't output anything.", "");
122         pred_mode = -1;
123     }
124
125     if (text == "/curr_state"){
126         bot.sendMessage(chat_id, "Show current active face.", "");
127     }
128 }
129
130 }
```

Elaboración propia.

### 5.7.6. Función prediction\_active()

Esta función se ejecuta mientras el valor de pred\_mode sea 1, primero se toman los datos de los switches de inclinación como tval junto con su respectivo numero, posteriormente se toman los valores x, y y z del acelerómetro y giroscopio, de estos últimos los datos del acelerómetro se convierten a  $\frac{m}{s^2}$  y los datos del giroscopio en grados por segundo, esto se puede ver en la figura 55:

Figura 55: Toma de datos para predicción activa.

```

205
206 void prediction_active(){
207     // put your main code here, to run repeatedly:
208
209     //Obtencion de señal analogica del tilt switch a valores digitales.
210     tval1 = digitalRead(tp1);
211     tval2 = digitalRead(tp2);
212     tval3 = digitalRead(tp3);
213     tval4 = digitalRead(tp4);
214     tval5 = digitalRead(tp5);
215     tval6 = digitalRead(tp6);
216
217     //-----Comienza codigo de funcionamiento del giroscopio
218
219     int16_t accX, accY, accZ;    //Componentes XYZ de Acelerometro
220     int16_t gyroX, gyroY, gyroZ;    //Componentes XYZ de Giroscopio
221
222     mpu.getAcceleration(&accX, &accY, &accZ);    //Obtencion de valores
223     mpu.getRotation(&gyroX, &gyroY, &gyroZ);
224
225     // Convertir los valores de aceleración y velocidad angular a unidades reales
226     float ax = (float)accX / 16384.0; // Dividir por 16384 para obtener g (gravedad, 9.81 m/s^2)
227     float ay = (float)accY / 16384.0;
228     float az = (float)accZ / 16384.0;
229
230     float gx = (float)gyroX / sensitivity; // Convertir a grados por segundo
231     float gy = (float)gyroY / sensitivity;
232     float gz = (float)gyroZ / sensitivity;
233

```

Elaboración propia.

En la figura 56 se muestra un fragmento de código cuya función es enviar a monitor serial todas las muestras separadas por comas tomadas en ese momento por los sensores, no es visible para el usuario pero es de utilidad para conocer el comportamiento de la caja en un determinado momento.

Figura 56: Salida de datos en monitor serial.

```

233
234     Serial.print(tval1); //tilt 1
235     Serial.print(",");
236     Serial.print(tval2); //tilt 2
237     Serial.print(",");
238     Serial.print(tval3); //tilt 3
239     Serial.print(",");
240     Serial.print(tval4); //tilt 4
241     Serial.print(",");
242     Serial.print(tval5); //tilt 5
243     Serial.print(",");
244     Serial.print(tval6); //tilt 6
245
246 //Muestra X , Y, Z tanto del giroscopio como del acelerometro
247 Serial.print(",");
248
249     Serial.print(ax);
250     Serial.print(",");
251     Serial.print(ay);
252     Serial.print(",");
253     Serial.print(az);
254
255     Serial.print(",");
256
257     Serial.print(gx);
258     Serial.print(",");
259     Serial.print(gy);
260     Serial.print(",");
261     Serial.print(gz);
262

```

Elaboración propia.

Posteriormente se utiliza la función side\_pred, a la cual se le envían los datos de todos los componentes para que realice la predicción, el resultado de esta es un valor entero entre el 1 y el 6 correspondiente la cara inferida que es almacenado y convertido en cadena. El mensaje de Telegram con la inferencia solo es enviado cuando existe un cambio de cara en el cubo, mientras last\_pred sea igual a la predicción actual no se hace nada, al momento de existir un cambio se envía un mensaje al chat de telegram con la predicción, así como puede ser visto en la figura 57:

Figura 57: Predicción de cara y envío a chat.

```

262
263     int prediction = side_pred(tval1, tval2, tval3, tval4, tval5, tval6, ax, ay, az, gx, gy, gz);
264     Serial.print(" | Prediccion: ");
265     Serial.println(prediction);
266
267     String predToshow = String(prediction);
268
269 //Mandar mensajes al bot de telegram
270
271     if (last_pred != prediction){
272         String chat_id = CHAT_ID;
273         bot.sendMessage(chat_id, "(predicted) Current side is: " + predToshow, "");
274         last_pred = prediction;
275     }
276
277 }

```

Elaboración propia.

### 5.7.7. Función side\_pred()

Esta función alberga las decisiones que conforman el clasificador de árbol, mediante esta estructura anidada es como se evalúan características para encontrar la clase a la que pertenece una sola muestra. En las figuras 58 y 59 se muestra que la función recibe todos los parámetros de los sensores.

Figura 58: Función de predicción Parte 1.

```

278 int side_pred(int t1, int t2,int t3,int t4,int t5,int t6,int ax,int ay,int az,int gx,int gy,int gz){
279     //Serial.print("flag: side_pred reached | ");
280     int prediction;
281
282     if(az < -0.88){
283         if(ay < -0.65){
284             prediction = 2;
285         }
286         if(ay > -0.65){
287             if(ax < 0.745){
288                 if(ay <= 0.715){
289                     prediction = 1;
290                 }
291                 if(ay > 0.715){
292                     prediction = 4;
293                 }
294             }
295             if(ax > 0.745){
296                 prediction = 5;
297             }
298         }
299     }
300     if(az > -0.88){
301         if(az < 0.535){
302             if(ay < -0.635){
303                 if(ax < -0.625){
304                     prediction = 3;
305                 }
306                 if(ax > -0.625){
307                     prediction = 2;
308                 }
309             }
310             if(ax > 0.85){
311                 if(ay < -0.055){
312                     prediction = 5;
313                 }
314                 if(ay > -0.055){
315                     prediction = 2;
316                 }
317             }
318         }
319     }
320 }
```

Elaboración propia.

Figura 59: Función de predicción Parte 2.

```

320 }  
321 if(ay > -0.635){  
322 if(ax < -0.555){  
323 if(ay < 0.05){  
324 prediction = 3;  
325 }  
326 if(ay > 0.65){  
327 prediction = 4;  
328 }  
329 }  
330 if(ax > -0.555){  
331 if(ax < 0.685){  
332 prediction = 4;  
333 }  
334 if(ax > 0.685){  
335 prediction = 5;  
336 }  
337 }  
338 }  
339 if(ax > 0.535){  
340 if(c4 < 0.5){  
341 if(ay < 0.44){  
342 prediction = 5;  
343 }  
344 if(ay > 0.44){  
345 if(ax < -0.5){  
346 prediction = 6;  
347 }  
348 if(ax > -0.5){  
349 prediction = 4;  
350 }  
351 }  
352 }  
353 if(ax < -0.705){  
354 if(ax < -0.74){  
355 prediction = 3;  
356 }  
357 if(ax > -0.74){  
358 prediction = 6;  
359 }  
360 }  
361 }  
362 if(ax > 0.705){  
363 if(ay <= 0.255){  
364 prediction = 5;  
365 }  
366 if(ay > 0.255){  
367 prediction = 6;  
368 }  
369 }  
370 }  
371 }  
372 return prediction;  
373 }  
374 }  
375 }

```

Tras haber inferido la cara conforme la muestra ingresada la función regresa como resultado un valor entero correspondiente a la predicción, esto se puede ver en la figura 60:

Figura 60: Función de predicción Parte 3.

```

362 if(ax > 0.705){  
363 if(ay <= 0.255){  
364 prediction = 5;  
365 }  
366 if(ay > 0.255){  
367 prediction = 6;  
368 }  
369 }  
370 }  
371 }  
372 return prediction;  
373 }  
374 }

```

Elaboración propia.

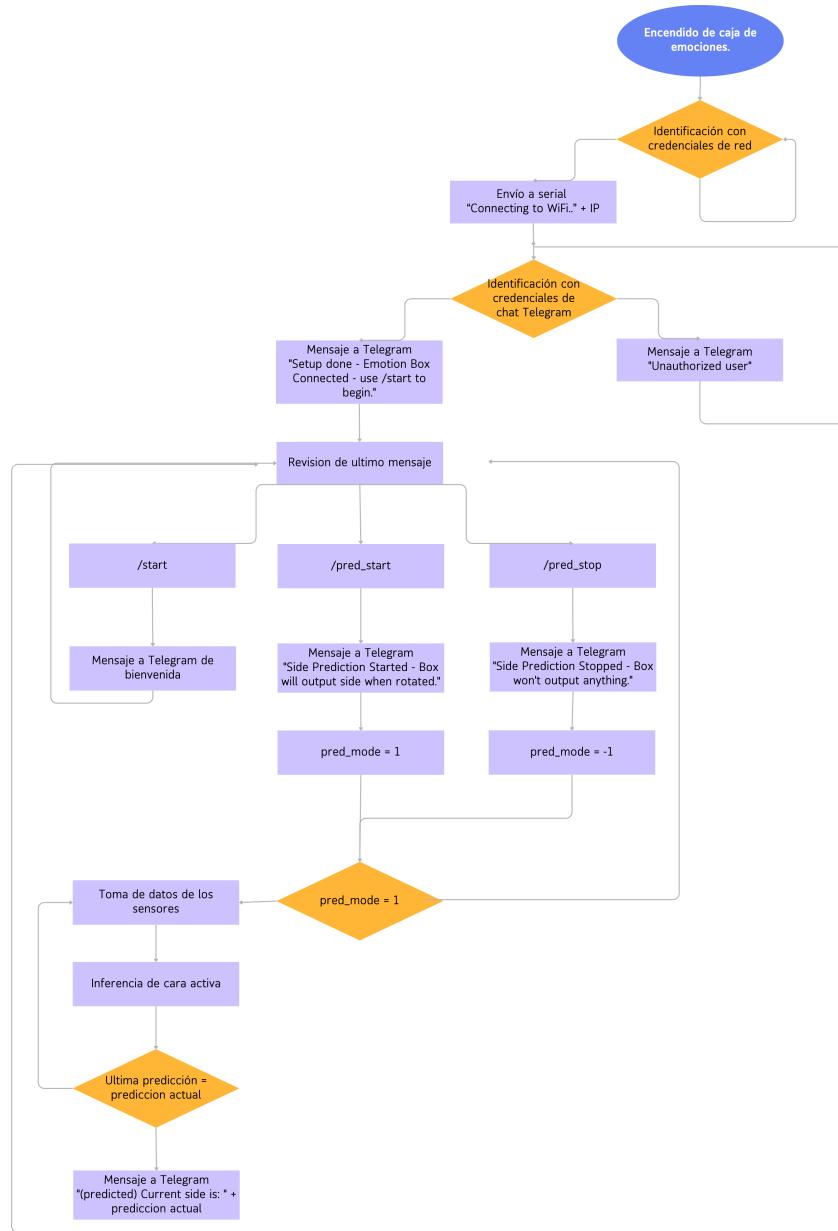
## 6. Resultados

En este capítulo se muestran los resultados obtenidos durante la realización del prototipo.

### 6.1. Funcionamiento general del cubo de emociones

En la figura 61 se muestra un diagrama con el uso general que tiene la caja desde el punto de vista del usuario.

Figura 61: Diagrama para uso de caja de emociones.



Un punto importante a considerar es que la caja solo hace la predicción de la cara activa en el momento que se hace un cambio de cara, esto con el propósito de no saturar de mensajes al usuario, por lo que no se notificara al usuario mientras la misma cara se mantenga activa. Del mismo modo la predicción puede ponerse en acción o detenerse en cualquier momento.

## 6.2. Uso del cubo mediante comandos de Telegram

Los comandos que el usuario puede ingresar por el chat se muestran en la tabla 12 con una descripción corta de lo que hacen.

Tabla 12: Comandos para el uso del cubo de emociones.

Comando	Uso
/start	Comando de inicio, muestra una corta bienvenida y muestra los comandos disponibles.
/pred_start	Pemite el comienzo de la predicción de caras del cubo, esto solo ocurre cuando se hace un cambio de cara, no notifica constantemente.
/pred_stop	Una vez activa la predicción, con este comando se puede detener, al mover el cubo no se notifica en el chat de Telegram.

## 6.3. Entrada de datos de los sensores

En la figura 62 se muestra como se da la obtención de los datos y también la inferencia de la cara, esta información se envía por serial por lo que no es visible para el usuario en el chat, esta información se conforma de una cadena con todos los valores obtenidos en ese momento acompañado de la inferencia hecha. Una vez se usa el comando /pred\_start se hacen predicciones cada segundo acerca de la posición pero solo se le es mostrada al usuario en cuanto mueve la caja.

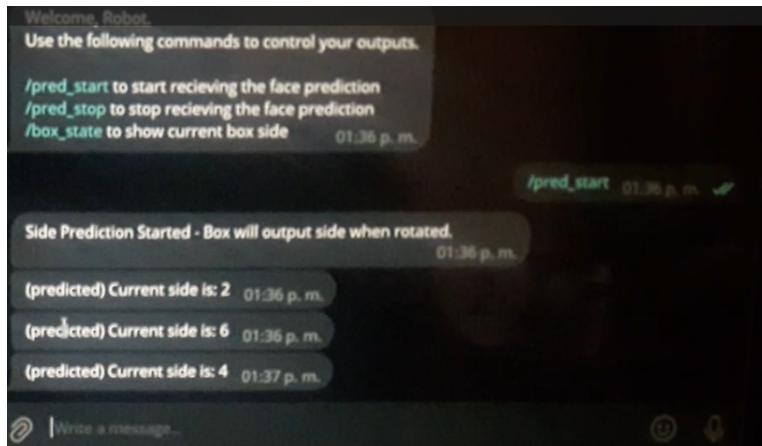
Figura 62: Salida de valores en monitor serial.

```
got response
handleNewMessages
1
/pred_start
/pred_start used
1.0,1,1,1,-0.02,-1.02,-0.12,1152.86,-70.00,-38.57 | Prediccion: 2
,0,1,1,1,1,-0.02,-1.02,-0.12,1115.71,-34.29,-35.71 | Prediccion: 2
1,0,1,1,1,1,-0.01,-1.02,-0.11,1164.29,-37.14,-35.71 | Prediccion: 2
1,0,1,1,1,1,-0.01,-1.02,-0.11,1142.86,-28.57,-5.71 | Prediccion: 2
1,1,1,1,1,0,0.17,0.01,0.98,471.43,1484.29,-10588.57 | Prediccion: 6
1,1,1,1,1,0,0.05,-0.03,0.89,1118.57,-40.00,-38.57 | Prediccion: 6
1,1,1,1,1,0,0.05,-0.03,0.90,1132.86,-10.00,-21.43 | Prediccion: 6
1,1,1,1,1,0,0.04,-0.02,0.88,1051.43,231.43,-2042.86 | Prediccion: 6
1,1,1,1,1,0,0.07,0.98,-0.10,1138.57,-61.43,-21.43 | Prediccion: 4
1,1,1,1,1,0,0.06,0.98,-0.10,1141.43,-67.14,-17.14 | Prediccion: 4
```

## 6.4. Predicción y muestra de cara activa

En la figura 63 se muestra un ejemplo del uso indicado de la caja controlada por comandos del chat de Telegram, para que esto se lleve a cabo es necesario que exista una red estable cargada en la placa. Dado el estado de prototipo las inferencias se hacen solamente como números bajo el entendido de que cada uno de estos representaría una expresión que en un futuro puede ser asignada.

Figura 63: Prediccion de caja en chat de Telegram.



## 7. Conclusiones

En este capítulo se habla de las conclusiones a las que se llegó durante la realización del proyecto, así como conclusiones personales. Además de esto se habla de áreas de oportunidad e implementaciones futuras para el proyecto.

### 7.1. Conclusión general del proyecto TinyML para la implementación de cubo de emociones

Tomando como guía los objetivos propuestos a inicio del desarrollo de las diferentes partes de este proyecto puede ser dicho que la gran mayoría de estas son cumplidas con un aceptable grado de efectividad. Una situación particular que ocurre al hablar de la parte de Machine Learning en las tecnologías de información es el como la creación de modelos matemáticos, clasificadores supervisados, entrenamiento, evaluación etc. suele ser de las ultimas etapas relacionadas a un proyecto propio de la disciplina, mas de la primera mitad del flujo de trabajo suele darse en el análisis, obtención, limpieza y partición de datos para que sean usados por determinados modelos de clasificación.

El autismo como todos los padecimientos relacionados con la mente son situaciones complejas de tratar especialmente en los niños, no solo por la naturaleza propia de los síntomas sino por el ambiente social en la que ellos se desarrollan, por lo mismo es que hay un gran interés en la investigación de tecnologías que ayuden a mejorar su calidad de vida como por ejemplo el proyecto aquí presente. Es importante que dichas tecnologías tomen en consideración no solo la parte electrónica y matemática sino también física, psicológica, demográfica y hasta económica para la retroalimentación de prototipos como la caja de emociones

### 7.2. Conclusiones por objetivo establecido

1. Objetivo general
  - a) Implementar algoritmos de aprendizaje automático en un micro controlador ESP32 S2 MINI para detección del estado de ánimo de un niño con autismo.
    - 1) Usando la placa Wemos y técnicas de ML fue posible inferir una cara de la caja, lo que funge como la base para indicar las emociones de un niño.
2. Objetivos específicos
  - a) Mejorar un prototipo existente de un cubo de sentimientos.
    - 1) El nuevo prototipo desarrollado mejoró con respecto a su antecesor en tamaño, construcción física, resistencia, programación y potencial de uso.
    - b) Obtener un dataset para clasificación de sentimientos.

- 1) Se desarrollo una metodología practica y eficiente para obtener datasets, gracias a esto fue posible crear varios de estos durante el desarrollo del prototipo.
- c) Implementar algoritmos de ML para clasificación de sentimientos.
  - 1) Se implemento con efectividad un algoritmo de clasificador de árbol que identifica caras del cubo activas correspondientes a emociones con un 97 % de precisión.
- d) Implementar algoritmos utilizando librerías propias de Tiny ML.
  - 1) Se utilizo con efectividad una gran parte del abanico de librerías necesarias para la aplicación de Tiny ML.
- e) Escritura de tesina.
  - 1) El desarrollo completo del prototipo desde su justificación, piezas, ensamblado, notas y algoritmos ha sido documentado no solo como tesina sino también en repositorio en Github para su conservación y futuras mejoras.

### 7.3. Conclusiones personales

El desarrollo de tecnologías de información normalmente requieren conocimiento y manejo de otras áreas de la industria además de la que se este trabajando, por ejemplo, aunque el eje de este mismo proyecto sean modelos de aprendizaje, se hace indispensable el conocimiento de electrónica para figurar el ensamblado, método de obtención de datos, manejo de cableado e incluso diseño en 3D para construir un sistema embebido. La ventaja de estas situaciones es que mediante requerimientos reales es que se impulsa el aprendizaje de técnicas desconocidas con el objetivo de resolver una tarea particular, esto tanto en las tareas manuales como en las técnicas de investigación, escritura de reportes, consulta de expertos, etc.

Otra lección importante que se rescata es que mediante la compresión de conceptos de modelos de aprendizaje es posible implementar soluciones prácticas y efectivas, así sacándole un provecho a las herramientas que se tengan a la mano, por ejemplo la descomposición del clasificador en decisiones if-else ahorra la necesidad de instalaciones o librerías más complejas pero más exigentes, este principio puede ser aplicado a modelos de aprendizaje de mayor complejidad.

### 7.4. Áreas de oportunidad

Luego la fase de desarrollo y pruebas de prototipo se hacen notables algunos puntos con los cuales dar retroalimentación para el diseño, entre ellos se encuentran:

- Reducir el tamaño del prototipo y utilizar un método de construcción más resistente.

Aunque el material de la caja es firme para uso en ambientes controlados, los escenarios de uso real posiblemente demanden de una construcción mas resistente y aislada.

- Definir requerimientos y comportamientos de red wifi para el funcionamiento de la caja. El modulo wifi realiza su función, pero se desconocen los parámetros exactos bajo los cuales opera.
- Cambiar el uso de UniversalTelegramBot por una aplicación complementaria propia del prototipo. El uso de Telegram o alguna otra plataforma es útil para probar funcionalidades mas aun el proyecto debe apuntar a la independencia con el desarrollo de una aplicación móvil particular.
- Mejorar el modelo de árbol para considerar una red neuronal. Basado en las técnicas aprendidas en este desarrollo adaptar otros tipos de modelos de aprendizaje de mayor complejidad, no solo para este prototipo sino para proximos.

## Índice de figuras

1. Representación de sistema adaptativo . . . . .	5
2. Proceso del apredizaje supervisado en Machine Learning . . . . .	8
3. Árbol de decisiones de una actividad común. . . . .	8
4. Areas del IoT . . . . .	14
5. Sistema de interacción robot-niño. . . . .	19
6. Abstracción de las capas del framework . . . . .	21
7. Abstracción de clasificador por método de cubrimiento. . . . .	22
8. Caja de emociones prototipo previo . . . . .	24
9. Gráfico de los componentes del prototipo previo . . . . .	25
10. Abstracción del prototipo previo . . . . .	25
11. Distribución de pines de placa ESP WEMOS S2 MINI . . . . .	27
12. Switch de inclinación . . . . .	28
13. Sensor acelerómetro y giroscopio MPU6050 . . . . .	29
14. Construcción física de la caja de emociones . . . . .	30
15. Interfaz de usuario y modelo de ejemplo en FreeCAD . . . . .	31
16. Logotipo de la interfaz de Arduino . . . . .	32
17. Ejemplo de la interfaz presentada por Jupyter Notebook . . . . .	33
18. Ejemplo de gráfica de datos . . . . .	35
19. Diseño conceptual de la tapa hecho en FreeCAD . . . . .	40
20. Diseño conceptual del contenedor hecho en FreeCAD . . . . .	41
21. Construcción final de la caja usando acrílico e impresión 3D . . . . .	41
22. Esquema del prototipo usando la placa WEMOS S2 mini . . . . .	42
23. Ubicación de SDA y SCL . . . . .	43
24. Función de calibración de MPU6050 en C++ . . . . .	44
25. Fragmento de código donde se da la toma de datos . . . . .	45
26. Fragmento de código para muestra de datos. . . . .	46
27. Salida serial de datos según componentes del cubo. . . . .	46
28. Programa para obtener y almacenar la salida serial de datos. Parte 1. . . . .	47
29. Programa para obtener y almacenar la salida serial de datos. Parte 2 . . . . .	48
30. Salida de programa a archivo .csv. . . . .	48
31. Formato de datos almacenados. . . . .	49
32. Notebook de clasificador Parte 1. Importe de librerías. . . . .	50
33. Notebook de clasificador Parte 2. Carga de dataset a variable. . . . .	51
34. Notebook de clasificador Parte 3. Descripción de dataset. . . . .	51
35. Notebook de clasificador Parte 4. Corrección de características. . . . .	52
36. Notebook de clasificador Parte 5. Asignación de características a encabezado. . . . .	52
37. Notebook de clasificador Parte 6. Separación de dataset y características en entrenamiento y prueba. . . . .	53
38. Notebook de clasificador Parte 7. Creación y entrenamiento de clasificador de árbol. . . . .	53
39. Notebook de clasificador Parte 8. Precisión de set de prueba y set de entrenamiento del modelo. . . . .	54

40. Notebook de clasificador Parte 9. Reporte de clasificación del clasificador de arbol. . . . .	54
41. Graficando el clasificador. . . . .	55
42. Clasificador de arbol para caja de emociones. Parte 1. . . . .	56
43. Clasificador de árbol para caja de emociones. Parte 2. . . . .	56
44. Desgloce de clasificador de arbol. . . . .	57
45. Importe de librerias. . . . .	60
46. Inicialización de variables. Parte 1 . . . . .	60
47. Inicialización de variables. Parte 2 . . . . .	61
48. Función setup 1. Inicio de pines en placa. . . . .	62
49. Función setup 2. Conexión Wifi y Telegram . . . . .	62
50. Función de calibración de giroscopio MPU6050. . . . .	63
51. Chequeo de mensajes nuevos. . . . .	63
52. Modo de predicción activo. . . . .	64
53. Manejador de mensajes y mensaje de bienvenida. . . . .	64
54. Comandos de chat para usuario. . . . .	65
55. Toma de datos para predicción activa. . . . .	66
56. Salida de datos en monitor serial. . . . .	67
57. Predicción de cara y envío a chat. . . . .	67
58. Función de predicción Parte 1. . . . .	68
59. Función de predicción Parte 2. . . . .	69
60. Función de predicción Parte 3. . . . .	69
61. Diagrama para uso de caja de emociones. . . . .	70
62. Salida de valores en monitor serial. . . . .	71
63. Predicción de caja en chat de Telegram. . . . .	72

## Índice de tablas

1. Tipos de algoritmos de aprendizaje . . . . .	7
2. Principios del comportamiento expresivo . . . . .	12
3. Lenguajes para describir estados emocionales. . . . .	13
4. Estructuras de la librería Pandas. . . . .	33
5. Algunas funciones de Pandas. . . . .	34
6. Funciones de arreglos de la librería Numpy. . . . .	34
7. Funciones de arreglos de la librería Matplotlib. . . . .	36
8. Funciones de Sklearn para árbol de decisiones. . . . .	37
9. Tecnologías compatibles con librería ESP 32 . . . . .	37
10. Algunas características relevantes de la librería parte 1 . . . . .	38
11. Algunas características relevantes de la librería parte 2 . . . . .	39
12. Comandos para el uso del cubo de emociones. . . . .	71

## Índice de algoritmos

1. Calibración de sensor mpu6050 . . . . .	43
--	----

## Referencias

- Abdmeziem, R., y Tandjaoui, D. (2014). Internet of things: Concept, building blocks, applications and challenges. *arXiv preprint arXiv:1401.6877*.
- Ari, N., y Ustazhanov, M. (2014). Matplotlib in python. *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 1-6.
- Ayala, E., y Gonzales Sánchez, S. (2015, agosto). Tecnologías de la Información y La Comunicación. <http://repositorio.uigv.edu.pe/handle/20.500.11818/1189>
- Ayodele, T. O. (2010). Machine learning overview. *New Advances in Machine Learning*, 2, 9-18.
- Babiuch, M., Foltynek, P., y Smutny, P. (2019). Using the ESP32 microcontroller for data processing. *2019 20th International Carpathian Control Conference (ICCC)*, 1-6.
- Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.
- Bressert, E. (2012). SciPy and NumPy: an overview for developers.
- Broker, A., y Schroeder, U. (2023). Investigating Feedback Types in JupyterLab for Programming Novices.
- Carbonell, J. G., Michalski, R. S., y Mitchell, T. M. (1983). An overview of machine learning. *Machine learning*, 3-23.
- Chang, K.-H. (2014). *Product design modeling using CAD/CAE: the computer aided engineering design series*. Academic Press.
- Components101. (2023). SW-520D - Tilt Sensor Module. <https://components101.com/sensors/sw-520d-tilt-sensor-module>
- Dantcheva, A., Elia, P., y Ross, A. (2015). What else does your biometric data reveal? A survey on soft biometrics. *IEEE Transactions on Information Forensics and Security*, 11(3), 441-467.
- DeFilippis, M., y Wagner, K. D. (2016). Treatment of autism spectrum disorder in children and adolescents. *Psychopharmacology bulletin*, 46(2), 18.
- Doyu, H., Morabito, R., y Brachmann, M. (2021). A TinyMLaaS Ecosystem for Machine Learning in IoT: Overview and Research Challenges. *2021 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 1-5. <https://doi.org/10.1109/VLSI-DAT52063.2021.9427352>
- Fedorov, D., Ivoilov, A. Y., Zhmud, V., y Trubin, V. (2015). Using of measuring system MPU6050 for the determination of the angular velocities and linear accelerations. *Automatics & Software Enginery*, 11(1), 75-80.
- Fezari, M., y Al Dahoud, A. (2018). Integrated development environment “IDE” for Arduino. *WSN applications*, 1-12.
- Frijda, N. H. (1986). *The emotions*. Cambridge University Press.
- Gardan, J. (2017). Additive manufacturing technologies: state of the art and trends. *Additive Manufacturing Handbook*, 149-168.
- Hackeling, G. (2017). *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd.
- Hasyim, M., Pramono, S., et al. (2021). Web-based telegram chatbot management system: Create chatbot without programming language requirements. *IOP Conference Series: Materials Science and Engineering*, 1096(1), 012075.

- Knörig, A., Wettach, R., y Cohen, J. (2009). Fritzing: a tool for advancing electronic prototyping for designers. *Proceedings of the 3rd international conference on tangible and embedded interaction*, 351-358.
- Kruth, J.-P., Leu, M.-C., y Nakagawa, T. (1998). Progress in additive manufacturing and rapid prototyping. *Cirp Annals*, 47(2), 525-540.
- Leo, M., Del Coco, M., Carcagni, P., Distante, C., Bernava, M., Pioggia, G., y Palestre, G. (2015). Automatic emotion recognition in robot-children interaction for ASD treatment. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 145-153.
- McKinney, W., et al. (2011). pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9), 1-9.
- Mextudia. (2019, abril). Universidad politécnica de victoria mextudia. <https://mextudia.com/universidades/universidad-politecnica-de-victoria/>
- Osisanwo, F., Akinsola, J., Awodele, O., Hinmikaiye, J., Olakanmi, O., Akinjobi, J., et al. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48(3), 128-138.
- Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984(197-219), 2-4.
- Riegel, J., Mayer, W., y van Havre, Y. (2016). FreeCAD. *Freecadspec2002. pdf*.
- Schizas, N., Karras, A., Karras, C., y Sioutas, S. (2022). TinyML for Ultra-Low Power AI and Large Scale IoT Deployments: A Systematic Review. *Future Internet*, 14(12), 363.
- Sen, P. C., Hajra, M., y Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, 99-111.
- Sleek, S. (2022). How Machine Learning Is Transforming Psychological Science. *APS Observer*, 36.
- Talaat, F. M. (2023). Real-time facial emotion recognition system among children with autism based on deep learning and IoT. *Neural Computing and Applications*, 35(17), 12717-12728.
- Thabtah, F., y Peebles, D. (2020). A new machine learning model based on induction of rules for autism detection. *Health informatics journal*, 26(1), 264-286.
- Tsuneki, M. (2022). Deep learning models in medical image analysis. *Journal of Oral Biosciences*, 64(3), 312-320.
- Villanueva, M. J. S. (2023). Theft alarm system using Arduino Uno's PIR motion and tilt ball switch sensors. *International Journal of Research*, 11(3), 11-20.
- wemos.cc. (2021). S2 mini. [https://www.wemos.cc/en/latest/s2/s2\\_mini.html](https://www.wemos.cc/en/latest/s2/s2_mini.html)
- Xia, F., Yang, L. T., Wang, L., Vinel, A., et al. (2012). Internet of things. *International journal of communication systems*, 25(9), 1101.